EPFL

Thèse n° 9825

Theory of Deep Learning: Neural Tangent Kernel and Beyond

Présentée le 9 août 2022

Faculté des sciences de base
Chaire de théorie des champs statistiques
Programme doctoral en mathématiques

pour l'obtention du grade de Docteur ès Sciences

par

Arthur Ulysse JACOT-GUILLARMOD

Acceptée sur proposition du jury

Prof. F. Nobile, président du jury
Prof. C. Hongler, directeur de thèse
Prof. B. Hanin, rapporteur
Prof. M. Mondelli, rapporteur
Prof. E. Abbé, rapporteur

École
polytechnique
fédérale
de Lausanne

2022

Acknowledgements

I first want to thank my advisor Clément for all the inspiration and advice he gave me. Throughout my PhD, he always understood when to correct me and push me forward and when to trust me and give me freedom. I loved all the discussions we had and will continue to have in the future.

I also wish to thank Franck who followed me from the very start of my PhD to the end. For each new idea, I was always looking forward to coming to the EPFL to explain it all to him. Our team was always full of great discussions and exchanges of ideas, for which I have to thank Berfin, Francesco, François, Evgenii, Maxime, SC, and Marie.

I want to thank all of the other researchers I worked with: Matthieu, Mario, Stefano, Levent, Stéphane, Giulio, Johanni, Wulfram, Benjamin, Nikolai, and Stanislav. The results of some of these collaborations appear in this thesis.

I am also thankful to the members of the Jury for my thesis defense: Emmanuel Abbé, Boris Hanin, Marco Mondelli, and Fabio Nobile.

Je remercie aussi tout spécialement ma famille qui m'a suivi tout au long de mes aventures mathématiques. Ma maman qui est allée chercher mon diplôme last minute à Berlin pour que je puisse commencer mon master, et qui n'a jamais caché sa fierté d'avoir un garçon doctorant en mathématiques. Mon papa qui m'a introduit à la programmation et qui m'a surtout appris à ne respecter aucune convention et toujours suivre mon intuition. Finalement, ma grande sœur Adèle (et sa fille Inna) qui m'a appris tant de choses, et surtout qui m'a expliqué la trigonométrie quand j'en avais besoin pour construire une maquette, je croyais que le cosinus était de la magie pure!

Ma très chère Aline, je te remercie pour tout l'amour et le bonheur que tu m'a donné toutes ces années. Beaucoup des idées et preuves de cette thèse me sont venues alors que tu dormais doucement dans mes bras.

Abstract

In the recent years, Deep Neural Networks (DNNs) have managed to succeed at tasks that previously appeared impossible, such as human-level object recognition, text synthesis, translation, playing games, and many more. In spite of these major achievements, our understanding of these models, in particular of what happens during their training, remains very limited.

This PhD started with the introduction of the Neural Tangent Kernel (NTK) to describe the evolution of the function represented by the network during training. In the infinite-width limit, i.e. when the number of neurons in the layers of the network grows to infinity, the NTK converges to a deterministic and time-independent limit, leading to a simple yet complete description of the dynamics of infinitely-wide DNNs. This allowed one to give the first general proof of convergence of DNNs to a global minimum, and yielded the first description of the limiting spectrum of the Hessian of the loss surface of DNNs throughout training.

More importantly, the NTK plays a crucial role in describing the generalization abilities of DNNs, i.e. the performance of the trained network on unseen data. The NTK analysis uncovered a direct link between the function learned by infinitely wide DNNs and Kernel Ridge Regression predictors, whose generalization properties are studied in this thesis using tools of random matrix theory.

Our analysis of KRR reveals the importance of the eigendecomposition of the NTK, which is affected by a number of architectural choices. In very deep networks, an ordered regime and a chaotic regime appear, determined by the choice of non-linearity and the balance between the weights and bias parameters; these two phases are characterized by different speeds of decay of the eigenvalues of the NTK, leading to a tradeoff between convergence speed and generalization. In practical contexts such as Generative Adversarial Networks or Topology Optimization, the network architecture can be chosen to guarantee certain properties of the NTK and its spectrum.

These results give an almost complete description of infinitely-wide DNNs in the NTK regime. It is then natural to wonder how it extends to finite-width networks used in practice. In the NTK regime, the discrepancy between finite- and infinite-widths DNNs is mainly a result of the variance with respect to the sampling of the parameters, as shown empirically and mathematically, relying on the similarity between DNNs and random feature models.

In contrast to the NTK regime, where the NTK remains constant during training, there exist so-called active regimes, where the evolution of the NTK is significant, and which appear in a number of settings. We describe one such regime in Deep Linear Networks with a very small initialization, where the training dynamics approaches a sequence of saddle-points, representing linear maps of increasing rank, leading to a low-rank bias which is absent in the NTK regime.

Keywords: Machine Learning, Deep Learning, Deep Neural Network, Neural Tangent Kernel, Kernel Methods, Random Matrix Theory, Statistical Learning Theory.

Résumé

Ces dernières années, les Réseaux de Neurones Multicouches (RNMs) ont accompli des tâches qui paraissaient auparavant impossibles, telles que la reconnaissance d'objets, la synthèse et traduction de textes et bien d'autres encore. Malgré ces succès majeurs, notre compréhension de ces modèles, en particulier de leur phase d'entraînement, reste encore très limitée.

Cette thèse de doctorat a commencé avec l'introduction du Neural Tangent Kernel (NTK) qui décrit l'évolution de la fonction représentée par le réseau pendant l'entraînement. Dans la limite de largeur infinie, où le nombre de neurones dans les couches du réseau tend vers l'infini, le NTK converge vers une limite déterministe et constante dans le temps, ce qui permet une description simple mais complète de la dynamique d'entraînement des réseaux de largeur infinie dans le régime dit NTK. Cela a permis la première preuve de convergence des RNMs vers un minimum global, ainsi que la première description du spectre limite de la Hessienne de la surface de coût des RNMs pendant l'entraînement.

De plus, le NTK joue un rôle crucial dans la description des propriétés de généralisation des RNMs, c'est-à-dire les performances du réseau sur des nouvelles données. L'analyse NTK révèle un lien direct entre la fonction apprise par des RNMs de largeur infinie et le prédicteur de Régression Ridge à Noyau dont les propriétés de généralisation sont décrites dans cette thèse en utilisant la théorie des matrices aléatoires.

Cette analyse révèle l'importance de la décomposition spectrale du NTK, qui est affectée par l'architecture des RNMs. En particulier, dans les réseaux avec un très grand nombre de couches, un régime ordonné et un régime chaotique apparaissent, déterminés par le choix de non-linéarité et l'équilibre entre les poids de connections et de biais. Ces deux régimes sont caractérisés par un spectre qui décroît à des vitesses différentes, conduisant à un compromis entre la vitesse de convergence et la généralisation. Dans des contextes pratiques tels que les Réseaux Antagonistes Génératifs et l'Optimization Topologique, l'architecture du réseau peut être choisie afin de garantir certaines propriétés du NTK et de son spectre.

Ces résultats présentent une théorie presque complète des RNMs de largeur infinie dans le régime NTK. Il est donc important de comparer cette description aux réseaux de largeur finie utilisés en pratique. Dans le régime NTK, on peut montrer, grâce à la similarité entre les RNMs et les modèles aux 'features' aléatoires, que la différence entre les réseaux de largeur finie et infinie est principalement due à la variance résultant de l'aléa des paramètres à initialisation.

Contrairement au régime NTK, où le NTK reste constant durant l'entraînement, des regimes dis actifs ont été observés, où l'évolution temporelle du NTK est significative. Un tel régime actif apparaît dans les Réseaux Linéaires Multicouches (RLMs) avec une petite initialisation, où la dynamique d'entraînement approche une suite de points selles, chacun représentant une fonction linéaire de rang croissant. En conséquence, le RLM tend à apprendre des fonctions de petit rang, ce qui n'est pas le cas dans le régime NTK.

Contents

List of Figures

Chapter 1

Introduction

Artificial Neural Networks (ANNs) are a family of Machine Learning models, which represent complex functions through a network of many simple computational units: the artificial neurons. These models were at first directly inspired by similarities with biological neural networks, but with the recent success of ANNs, their design is now mostly driven by their practical performance. As a result, today's ANNs often bear little resemblance to their biological cousins.

ANNs have a long history, generally considered to have started with the Perceptron [182], introduced in 1958. In the following half century, ANNs slowly evolved to resemble those in use today:

- In 1965, the first Deep Neural Networks (DNNs) with multiple layers of neurons is implemented [100, 99]. DNNs solve the issue described in [154] with the Perceptron which cannot describe the XOR function.
- ANNs require a training phase where the connection weights between the neurons are tuned. While a range of methods existed at first, today the training is almost always done with gradient based methods. The computation of the gradient of DNNs uses the so-called backpropagation algorithm, which was popularized by [187] in 1986, though earlier implementations exist [140].
- Inspired by biological neural networks [97], K. Fukushima [64] introduced the Neocognitron, which uses weight-sharing between connections to take advantage of translation invariance in images, resembling the animal visual cortex. This architecture evolved into today's Convolutional Neural Networks (CNNs) [123, 121, 124] which are now ubiquitous in computer vision.
- Other types of architecture best suited for text and time series data appeared: Recurrent Neural Networks (RNNs) [187], Long Short-Term Memory (LSTM) [91] and more recently Transformers [215].

The 21st century has marked the start of what is sometimes called the 'deep learning revolution', where ANNs have transformed from an exciting curiosity to being a key element behind many technological tools that we use everyday. DNNs have become the standard in areas such as Computer Vision, Natural Language Processing, Speech Analysis, self-driving cars and many more, outperforming more traditional statistical methods.

1.1 Towards a Theory of Deep Learning

In spite of the impressive practical success of Deep Learning, our theoretical understanding of DNNs remains very limited. DNNs are often said to be 'black boxes' - powerful models whose inner workings are mysterious. There are a number of important questions related to the analysis of DNNs:

- DNNs are notoriously hard to train and their performances can be dramatically affected by the choices of hyper-parameters such as the learning rate, the choice of non-linearity, the architecture and size of the neural network and many more. To tune these hyper-parameters, practitioners either rely on their intuition (acquired through years of experiences or passed on as 'common practices') or on a costly hyper-parameter search where a wide range of hyper-parameters are tried out. A theory of deep learning has the potential to confirm and formalize these intuitions, and to speed up the hyper-parameter search.

- DNNs are used in a wide variety of settings, from computer vision to protein folding. A large proportion of the papers appearing every day in Machine Learning can be described as proposing a new architecture that performs well on a specific task, resulting in an ever-growing list of architectures. Some of these new techniques are sometimes motivated by unclear theoretical arguments: for example Batch Normalization - which is used in the training of most DNNs in practice - was first proposed as a solution to the 'internal covariate shift' problem, which is neither rigorously defined, nor shown to be a problem, nor shown to be solved by batch normalization in the original paper [98]. We need a theory of deep learning to compare different architectures and to explain why a certain model is adapted to a specific task.

- A better theory of deep learning could also have impacts on the way we approach statistical models. DNNs differ from traditional statistical models and go directly against some of the accepted statistical wisdoms - such as avoiding models with many parameters. However DNNs outperform these traditional models on many tasks. Better understanding DNNs would allow one to update these beliefs, opening the door to a whole new family of related models.

- DNNs are particularly efficient on tasks that humans are good at: object recognition, text and speech analysis/synthesis, car driving and more. The fact that DNNs and the human brain are successful on similar tasks suggests that DNNs are a good simplification of biological networks, which still captures most of its key features. The mathematical tools and concepts that we develop to understand DNNs today might one day play a role in understanding key mechanisms in the working of the human brain.

There are many open theoretical questions related to DNNs, this thesis will focus on and give partial answers to the two general questions of **convergence** and **generalization**:

Convergence: DNNs undergo a training phase, where the parameters of the network are optimized with a local search (typically a variant of gradient descent) to minimize a loss which measures the performance of the network on a fixed training set. There is no general guarantee of convergence of the local search to a global minimum of the loss because the latter is in general not convex. However, in practice, DNNs typically reach a global minimum during training consistently as long as the network is large enough, i.e. has enough neurons. The aim is to understand why increasing the size of the network alleviates the problem of non-convexity.

Generalization: Once the network has been trained, it is typically evaluated on a test set, a new dataset distinct from the training set, to evaluate the performance of the network on unseen data and to measure whether the network is able to generalize and not only memorize. The behavior of the test error of DNNs as the number of neurons in the network grows is however in apparent contradiction with the traditional statistical framework of bias/variance tradeoff: decomposing the expected test error into a bias and variance term, as the number of neurons increases (and hence the number of parameters), the bias term is expected to decrease and the variance term to increase, leading to a U-shaped curve with a sweet spot balancing the bias and variance terms. Instead for DNNs the test error has consistently been observed to keep decreasing as the number of neurons grows, suggesting that the best performances would be obtained as the number of neurons grows to infinity.

This phenomenon could be explained by the existence of an implicit bias effect for gradient descent: even though large networks may have many global minima, each with potentially very different test errors; it seems that the training dynamics naturally converge to global minima which generalize well. To describe this implicit bias of gradient descent, we need mathematical tools to describe the dynamics of training.

Approach

The theory of deep learning remains a very open field, and many distinct approaches have been explored. The approach underlying the results presented in this thesis can be described by three choices: we focus on 'wide' networks, we study the dynamics of training and we take a functional perspective.

Wide networks: The neurons of a neural network are usually organized into layers, from the input layer 0 to the output layer L, and $L-1$ so-called *hidden* layers in between. The number L is called the *depth*, while the number of neurons in each of the hidden layers is called the *width*. Most of the work presented in this thesis occurs in the infinite-width limit, i.e. in the limit where the width of the network grows as the depth is kept constant.

Dynamics: To understand the properties of the trained network - such as the test error - we first need to understand the training dynamics that lead to this particular trained network. We therefore focus on developing mathematical tools to describe the training dynamics of DNNs[1].

Functional approach: A lot of theoretical work on DNNs focuses on understanding the loss landscape of DNNs, in particular its critical points and Hessian, leading to a vision of DNNs 'from the parameter space'. In this thesis, the focus is shifted to the function represented by the network (the so-called network function, which maps the activation of input neurons to that of the output neurons). There are two key advantages to this perspective:

1. Since the number of parameters grows with the size of the network, the dimension of the parameter space grows when one studies the infinite-width limit. In contrast, one can fix a function space to which the network function belongs for any width, and study the infinite-width limit of this function directly.

2. Two neurons in the same hidden layer can be swapped (by swapping all their incoming and outcoming connections) without changing the outputs of the network (and hence neither the

[1] This is in contrast to another line of work [90, 81, 80] which identifies properties that are typically satisfied by networks at the end of training and prove generalization bounds conditioned on these properties (which could then be checked empirically for a specific network).

train nor test error). The parameters of the network are affected by this swapping which implies that any property of the loss surface is true up to any permutation, and that two choices of parameters which are separated by a large Euclidean distance, might be very close after applying the right permutation. In contrast, the network function is invariant under permutation symmetries, and from the functional perspective, these permutations can be essentially forgotten.

1.2 Original Papers

This thesis compiles the results of 7 papers published during my PhD and 1 preprint. In this introduction I give an overview of each of these papers and explain how they are linked. Note that the papers are not presented in chronological order, they are grouped by themes, to present a cohesive theory. The original text of the 8 papers can be found in Sections 2 to 9 and their respective appendices are reproduced in Appendices B to I.

The first 3 papers describe the convergence and generalization of infinitely wide DNNs using the Neural Tangent Kernel (NTK):

- The starting point of my PhD is the paper *Neural Tangent Kernel: Convergence and Generalization in Neural Networks* [105], written with my coauthors Franck Gabriel and Clément Hongler and published at NeurIPS 2018. The Neural Tangent Kernel is introduced and its infinite-width limit throughout training is described, leading to a simple description of the training dynamics of infinitely wide DNNs.

- As a follow up, I studied the implications of the previous result on the loss surface of DNNs around the training path of infinitely wide DNNs in the paper *Asymptotic Spectrum of the Hessian of DNN Throughout Training* [106], written with Franck Gabriel and Clément Hongler and published at ICLR 2020.

- The NTK analysis implies a direct link between the training of infinitely wide DNNs and Kernel Ridge Regression (KRR). In the paper *Kernel Alignment Risk Estimator: Risk Prediction from Training Data* [103], written with Berfin Şimşek, Francesco Spadaro, Clément Hongler and Franck Gabriel and published at NeurIPS 2020, we described the expected risk (or test error) of KRR and as an extension of infinitely wide DNNs, using tools of random matrix theory.

These three results showed the importance of the spectral decomposition of the NTK in the analysis of the convergence and generalization of DNNs. I studied how the architecture of DNNs affects the NTK and its spectrum in two papers:

- DNNs were known to feature an ordered (or freeze) and chaotic regime in very deep networks. In the paper *Freeze and Chaos: NTK views on DNN Normalization, Checkerboard and Boundary Artifacts [104]*, written with François Ged, Franck Gabriel and Clément Hongler and to appear at MSML2022, we extended this analysis to the NTK: with the ordered and chaotic regime characterized respectively by a very fast or slow spectral decay of the NTK. We also showed links with Layer Normalization and Mode Collapse in Generative Adversarial Networks.

- In the paper *DNN-Based Topology Optimisation: Spatial Invariance and Neural Tangent Kernel* [52] written with Benjamin Dupuis and published at NeurIPS 2021, we used the NTK to study the impact of using a DNN to learn optimal shapes in the task of Topology Optimization. We proposed two input embeddings to preserve the translation symmetries inherent in the underlying physical model and uncovered the role of the NTK as a low-pass filter, whose decay can be tuned to obtain shapes with different levels of detail.

The above results along with the work of many other researchers propose an almost complete theory of infinitely wide DNNs through the NTK. Since the networks used in practice have a large but finite-width, it is crucial to understand how close those finite-width networks are to their infinite-width counterparts:

- I first studied this question in the mostly empirical paper *Scaling Description of Generalization with Number of Parameters in Deep Learning* [67], with Mario Geiger, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler and Matthieu Wyart, published in the Journal of Statistical Mechanics: Theory and Experiments. We observed that the difference in test error between finite-width networks and their infinite-width counterparts is mostly due to the variance of the trained network function w.r.t. sampling of the parameters at initialization.

- This was followed by the theoretical paper *Implicit Regularization of Random Feature Models [102]* with Berfin Şimşek, Francesco Spadaro, Clément Hongler and Franck Gabriel and published at ICML 2020, in which we partially explain these observations mathematically for Random Feature models, which are good approximations of DNNs in the so-called NTK regime, where the NTK moves little during training.

Beyond the NTK regime, there are settings and limits of DNNs that lead to an NTK with a significant time evolution. These regimes are called active, in opposition to the NTK regime. I have studied one such regime in Deep Linear Networks (DLNs):

- In the paper *Saddle-to-Saddle Dynamics in Deep Linear Networks: Small Initialization Training, Symmetries and Sparsity* [107], written with François Ged, Berfin Şimşek, Clément Hongler and Franck Gabriel, we describe the training dynamics of DLNs initialized close to the saddle at the origin in parameter space. Gradient flow leaves the saddle along an optimal escape path which leads it to approach another saddle and so on and so forth. The rank of the matrix represented by the network increases by 1 at each of these saddles, approximating a greedy low rank algorithm where the network first optimizes amongst rank 1 matrices then rank 2 matrices and so on until reaching a global minimum [135].

In the rest of the introduction, I will summarize the results of these papers in a unified setup.

1.3 Setup

We will now define DNNs, describe their training procedure and introduce notations to study DNNs 'from the function space'.

Fully-Connected DNNs

Up to a few exceptions, all the results in this thesis are presented for so-called *fully-connected* DNNs. In fully-connected DNNs, the neurons are organized into $L+1$ layers, numbered from 0 (input layer) to L (output layer), with the layers 1 to $L-1$ being the hidden layers. The number of neurons in a layer $\ell = 0, \dots, L$ is denoted by n_ℓ, the number of neurons in the input layer n_ℓ equals the input dimension d_{in} and the number of output neurons n_L equals the output dimension d_{out}. We will generally consider d_{in} and d_{out} to be fixed while the number of neurons in the hidden layers $n_1, \dots, n_{L-1}$ vary.

Given an input $x \in \mathbb{R}^{n_0}$, the activations $\alpha_i^{(\ell)}(x)$ and pre-activations $\tilde{\alpha}_i^{(\ell)}(x)$ of the i-th neuron in the ℓ-th layer are defined from the vector of activations $\alpha^{(\ell-1)}(x) \in \mathbb{R}^{n_{\ell-1}}$ of the previous layers. For a fixed function $\sigma : \mathbb{R} \to \mathbb{R}$, which we call the non-linearity, and a scalar β, which we call the *bias strength*, we define inductively:

$$\begin{aligned}
\alpha^{(0)}(x) &= x \\
\tilde{\alpha}^{(\ell)}(x) &= \frac{1}{\sqrt{n_{\ell-1}}} W^{(\ell)} \alpha^{(\ell-1)}(x) + \beta b^{(\ell)} \\
\alpha^{(\ell)}(x) &= \sigma\left(\tilde{\alpha}^{(\ell)}(x)\right),
\end{aligned}$$

where $W^{(\ell)}$ is a $n_\ell \times n_{\ell-1}$ matrix called *connection weight* matrix and $b^{(\ell)}$ is a n_ℓ dimensional vector called *bias* vector.

The *parameters* of the network are all the connection weights and bias vectors of all the layers. We define the vector of parameters θ of dimension $P = \sum_{\ell=1}^{L}(n_{\ell-1}+1)n_\ell$ as the concatenation of all the parameters $\theta = (W_1, b_1, \dots, W_L, b_L)$. The parameters are to be learned during the training phase.

In contrast, the depth L and widths $n_0, \dots, n_L$ as well as the non-linearity σ and the bias strength β are called *hyper-parameters* of the network, as they remain fixed during training. Typical choices for the non-linearity include the ReLU function $\sigma(x) = \max\{0, x\}$, the arc tangent $\sigma(x) = \arctan(x)$ or the sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$.

The output of the network are the pre-activations $\tilde{\alpha}^{(\ell)}(x)$ of the last layer. The network function $f_\theta : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ is defined as $f_\theta(x) = \tilde{\alpha}^{(L)}(x)$.

Training Phase

In supervised learning, the goal is to find parameters θ of a network that implement a practical task such as classyfing cats from dogs in images. If we want e.g. to classify 64x64 RGB images of cats and dogs and train a network to discriminate between the two categories, we are looking for parameters θ such that the network function f_θ maps images x (represented as a dimension $d_{in} = 64 \times 64 \times 3$ vector) to a scalar value $f_\theta(x)$ that is positive when the image is of a cat and negative if it is of a dog.

To implement this task, we rely on a training set of N input/output pairs (x_i, y_i). In the cat and dogs example, the x_is would be images of cats and dogs and y_i would be $+1$ for cats and -1 for dogs (in this example $d_{out} = 1$). We write X and Y for the $d_{in} \times N$ and $d_{out} \times N$ matrices obtained from the concatenation of the inputs $(x_i)_{i=1,\dots,N}$ and outputs $(y_i)_{i=1,\dots,N}$ respectively.

Our goal is to find parameters θ that minimize a training loss which measures how accurate the network is on the training data. Typical choices are the Mean Squared Error (MSE) for regression tasks, where the outputs y_i can take any value in $\mathbb{R}^{d_{out}}$

$$\mathcal{L}^{MSE}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|f_\theta(x_i) - y_i\|^2$$

or the Binary Cross-Entropy loss for binary classification, with scalar outputs $y_i \in \{-1, +1\}$

$$\mathcal{L}^{BCE}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \log\left(1 + e^{-y_i f_\theta(x_i)}\right).$$

The loss $\mathcal{L}$ is then minimized with gradient descent (or a variant thereof) starting from a random initialization of the parameters as i.i.d. standard Gaussian $\mathcal{N}(0, 1)$. The results presented in this thesis are for gradient flow

$$\partial_t \theta(t) = -\nabla \mathcal{L}(\theta(t))$$

which approximates the gradient descent

$$\theta(t + \eta) = \theta(t) - \eta \nabla \mathcal{L}(\theta(t))$$

as the learning rate η goes to zero.

Our goal is to understand the evolution of the parameters $\theta(t)$, in particular at the end of training $\theta(\infty)$.

Function Space Perspective

The loss of DNNs as a function of the parameters $\mathcal{L}(\theta)$ is high-dimensional and non-convex, making the analysis of gradient flow difficult. Furthermore the parameter space $\mathbb{R}^P$ with $P = \sum_{\ell=1}^{L} (n_{\ell-1} + 1)n_\ell$ is inconvenient to work with, since the parameters $\theta \in \mathbb{R}^P$ are a concatenation of weight matrices W_ℓ and bias vectors b_ℓ each with their distinct gradients $\nabla_{W_\ell} \mathcal{L}(\theta(t))$ and $\nabla_{b_\ell} \mathcal{L}(\theta(t))$ and the dimensions of each of these matrices depends on the widths n_ℓ (which we later let grow to infinity).

In addition the loss $\mathcal{L}(\theta)$ is invariant under permutations of the neurons in the hidden layers, as a result every permutation of a critical point θ^* of the loss $\mathcal{L}$ is also a critical point [204]. Similarily any gradient flow path $\theta(t)$ can also be permuted in the same manner, yielding another valid gradient flow path. While there exists techniques to 'mod out' these symmetries for shallow networks ($L = 2$) [35, 183], it is difficult to find such symmetry-invariant representations of the parameters for deep networks ($L > 2$).

If we focus on the evolution of network function $f_{\theta(t)} : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ instead of the parameters $\theta(t)$, both of these problems are however naturally avoided:

1. As long as the non-linearity and the input and output dimensions n_0 and n_L are fixed, the network function f_θ belongs to a fixed space of functions $\mathcal{F}$, whatever the depth L or the

widths of the hidden layers $n_1, \dots, n_{L-1}$ are. Note that there are multiple reasonable choices for the function space: if σ is continuous, $\mathcal{F}$ could be taken to be the space $\mathcal{C}^0\left[\mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}\right]$ of continuous functions from $\mathbb{R}^{d_{in}}$ to $\mathbb{R}^{d_{out}}$, if σ is differentiable then one could consider the space of differentiable functions $\mathcal{C}^1\left[\mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}\right]$ instead.

2. The network function f_θ is invariant under permutations of the neurons within any given hidden layer.

From a functional perspective, the loss $\mathcal{L} : \mathbb{R}^P \to \mathbb{R}$ is to be viewed as the composition of two functions: first the *realization function* $F^{(L)} : \mathbb{R}^P \to \mathcal{F}$ which maps the parameters θ to the network function $f_\theta \in \mathcal{F}$ and the cost function $C : \mathcal{F} \to \mathbb{R}$ which maps a function f to its cost. Multiple choices for the cost function are possible, such as the already mentioned MSE $C^{MSE}(f) = \frac{1}{N}\sum_{i=1}^N \|f(x_i) - y_i\|^2$ or the BCE cost $C^{BCE}(f) = \frac{1}{N}\sum_{i=1}^N \log\left(1 + \exp\left(-y_i f(x_i)\right)\right)$. Note that both C^{MSE} and C^{BCE} are convex (C^{MSE} is even quadratic).

More generally, we will consider a general convex cost $C(f)$ which only depends on the value of the function f on the training set. Writing Y_θ for the $n_L \times N$ matrix obtained from the concatenation of the vectors $(f_\theta(x_i))_{i=1,\dots,N}$, we will often abuse notation and write the cost C as taking Y_θ as input, i.e. $C^{MSE}(Y_\theta) = \frac{1}{N}\|Y_\theta - Y\|_F^2$, where Y is the matrix of output labels and $\|\cdot\|_F$ is the Frobenius norm.

1.4 Neural Tangent Kernel

The Neural Tangent Kernel (NTK) naturally arises when one tries to describe the evolution of the network function $f_{\theta(t)}$ as the parameters $\theta(t)$ follow gradient flow on the parameters. The evolution of the parameters $\theta(t)$ trained on the MSE loss is given by the formula

$$\partial_t \theta(t) = \frac{2}{N}\sum_{i=1}^N Jf_{\theta(t)}(x_i)(y_i - f_{\theta(t)}(x_i)),$$

where the Jacobian $Jf_{\theta(t)}(x_i)$ of the outputs of the network with respect to the parameters θ is a $P \times n_L$ matrix. The evolution of the network function $f_{\theta(t)}(x)$ at any input x is given by

$$\partial_t f_{\theta(t)}(x) = \left(Jf_{\theta(t)}(x)\right)^T \partial_t \theta(t) = \frac{2}{N}\sum_{i=1}^N \left(Jf_{\theta(t)}(x)\right)^T Jf_{\theta(t)}(x_i)(y_i - f_{\theta(t)}(x_i)).$$

We define the *Neural Tangent Kernel (NTK)* $\Theta^{(L)} : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}^{n_L \times n_L}$ by

$$\Theta^{(L)}(x, y) = \left(Jf_{\theta(t)}(x)\right)^T Jf_{\theta(t)}(y).$$

The NTK is a *multidimensional kernel*: a function $K : \mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out} \times d_{out}}$ which takes two inputs $x, y \in \mathbb{R}^{n_0}$ and outputs a $d_{out} \times d_{out}$ matrix, such that for any set of N inputs $x_1, \dots, x_N$ the $d_{out}N \times d_{out}N$ *kernel Gram matrix* $K(X, X)$ with entries $(K(X, X))_{ki,k'i'} = K_{kk'}(x_i, x_{i'})$ is a positive semidefinite matrix. This is a generalization of the notion of *kernel* which is a function $K : \mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{in}} \to \mathbb{R}$ such that for any set of inputs $x_1, \dots, x_N$, the $N \times N$ kernel Gram matrix $K(X, X)$ with entries $(K(X, X))_{i,i'} = K(x_i, x_{i'})$ is positive semidefinite matrix.

Using the NTK, we can rewrite the evolution of the network function $f_{\theta(t)}$ as

$$\partial_t f_{\theta(t)}(x) = \frac{2}{N} \sum_{i=1}^{N} \Theta^{(L)}(x, x_i)(y_i - f_{\theta(t)}(x_i)).$$

This can be generalized to a cost C of the form $C(f) = \frac{1}{N} \sum_{i=1}^{N} c_i(f(x_i))$, in which case

$$\partial_t f_{\theta(t)}(x) = \frac{1}{N} \sum_{i=1}^{N} \Theta^{(L)}(x, x_i) \nabla c_i(f_{\theta(t)}(x_i)).$$

Clearly, if we can describe the evolution of the NTK for all time t then we can describe the evolution of the network function $f_{\theta(t)}$. The NTK is however a complex object: it is random at initialization due to the randomness of the parameters, and evolves in time as a result of the evolution of the parameters.

Tangent Linear Model

The Neural Tangent Kernel can be interpreted as approximating the DNNs by a 'tangent' linear model (as in tangent space for a manifold), hence the name. This tangent linear model $T_{\theta_0} F^{(L)}(\theta)$ around a fixed parameter vector θ_0 is given by the realization function

$$T_{\theta_0} F^{(L)}(\theta) = F^{(L)}(\theta_0) + JF^{(L)}(\theta_0)\theta$$

which is clearly affine in θ.

A central property of DNNs is that they are a non-linear model, in the sense that the realization function $F^{(L)}$ is non-linear. This non-linearity makes the analysis of the training of DNNs difficult: the non-linearity of $F^{(L)}$ makes the loss $\mathcal{L} = C \circ F^{(L)}$ non-convex. In comparison, linear models have a much nicer behavior: the loss $\mathcal{L} = C \circ F$ is convex, hence guaranteeing convergence of gradient flow on the parameters $\theta(t)$.

A general linear model can be defined as $f_\theta(x) = \sum_{p=1}^{P} \theta_p f_p(x)$ for a set of P function $f_p : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ called the *features* of the model. The realization function F therefore maps the parameters θ to a linear combination $\sum_{p=1}^{P} \theta_p f_p$ with coefficients given by the parameters θ. The features of the tangent linear model introduced above are the derivatives $x \mapsto \partial_{\theta_p} f_{\theta_0}(x)$ of the outputs w.r.t. to each of the parameters θ_p.

A linear model defines a kernel $K : \mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out} \times d_{out}}$ equal to the (rescaled) covariance of the features $f_1, \ldots, f_P$

$$K(x, y) = \sum_{p=1}^{P} f_p(x) \left(f_p(y)\right)^T.$$

The NTK $\Theta^{(L)}$ at θ_0 is simply the kernel of the tangent linear model at θ_0.

For a cost of the form $C(f) = \frac{1}{N} \sum_{i=1}^{N} c_i(f(x_i))$ for some differentiable functions $c_1, \ldots, c_N$, the parameters follows the differential equation

$$\partial_t \theta_p(t) = \frac{2}{N} \sum_{i=1}^{N} \left(f_p(x_i)\right)^T \nabla c_i(f_{\theta(t)}(x_i)),$$

while the function $f_{\theta(t)}$ follows the differential equation

$$\partial_t f_{\theta(t)}(x) = \frac{2}{N} \sum_{i=1}^{N} K(x, x_i) \nabla c_i(f_{\theta(t)}(x_i)). \tag{1.4.1}$$

Since the derivative only depends on $f_{\theta(t)}$ (and the kernel K) and not the parameters $\theta(t)$ themselves, the differential equation governing $f_{\theta(t)}$ can be solved independently of $\theta(t)$.

Furthermore since the dynamics of training of a linear model in function space only depends on the kernel K and not the features $f_1, \dots, f_P$, a linear model is uniquely described by its kernel K (if we abstract away the dynamics of the parameters). This is a special case of the so-called the 'Kernel trick'.

Kernel Gradient Descent

The dynamics in function space of a linear model can be interpreted as performing *kernel gradient descent* on the cost C with respect to the kernel K. kernel gradient descent can itself be interpreted as performing gradient descent in a function space $\mathcal{F}$ w.r.t. to a specific norm.

As we are in a infinite dimensional space, there is no canonical notion of gradient descent on a function space $\mathcal{F}$. The derivative $\partial_f C(f)$ of the cost $C(f)$ with respect to f is not a function in $\mathcal{F}$ but rather an element of the dual space $\mathcal{F}^*$ (the set of linear functions from $\mathcal{F}$ to $\mathbb{R}$). Given a scalar product $\langle \cdot, \cdot \rangle$ on the space $\mathcal{F}$ forming a Hilbert space, one can define the gradient $\nabla C(f)$ as the unique function $g \in \mathcal{F}$ such that $\langle g, f \rangle = \partial_f C(f)$. The choice of the scalar product can have a significant effect on the resulting gradient and one particular choice leads to the kernel gradient.

Given a multidimensional kernel K, there is a natural function space $\mathcal{F}_K$ and a scalar product $\langle \cdot, \cdot \rangle_K$ which form a Hilbert space called the Reproducing Kernel Hilbert Space (RKHS) of the kernel K: $\mathcal{F}_K$ is the completion of the set of functions $f(x) = \sum_{i=1}^{N} K(x, x_i) b_i$ for any finite N, set of inputs $x_1, \dots, x_N \in \mathbb{R}^{d_{in}}$ and coefficients $b_1, \dots, b_N \in \mathbb{R}^{d_{out}}$ and the scalar product of two functions $f(x) = \sum_{i=1}^{N} K(x, x_i) b_i$ and $g(x) = \sum_{j=1}^{N'} K(x, x'_j) b'_j$ is defined as $\langle f, g \rangle_K = \sum_{i=1}^{N} \sum_{j=1}^{N'} b_i^T K(x_i, x'_j) b'_j$ (and the definition is extended to the completion continuously).

For kernels of the form $K(x, y) = \sum_{p=1}^{P} f_p(x) \left(f_p(y)\right)^T$ (i.e. the kernel of a linear model with features $f_1, \dots, f_p$) the function space $\mathcal{F}_K$ is the set of functions $f(x) = \sum_{p=1}^{P} \theta_p f_p(x)$ for some coefficients $\theta_1, \dots, \theta_P$, i.e. $\mathcal{F}_K$ is the image of the realization function F. The scalar product of two functions $f(x) = \sum_{p=1}^{P} \theta_p f_p(x)$ and $g(x) = \sum_{p=1}^{P} \theta'_p f_p(x)$ is the scalar product $\theta^T \theta'$ of the coefficient vectors. If the features $f_1, \dots, f_P$ are linearly independent, the realization function F is invertible (when restricted to its image $\mathcal{F}_K$) and we can simply write $\langle f, g \rangle_K = \left(F^{-1}(f)\right)^T F^{-1}(g)$.

The dynamics described by equation 1.4.1 describe kernel gradient flow on the cost C.

1.5 Infinite-width Limit of the Neural Tangent Kernel

The infinite-width limit of the NTK was first described in the paper [105] which can be found in Section 2.

The infinite-width limit corresponds to letting the number of neurons in each of the hidden layers $n_1, \dots, n_{L-1}$ grow to infinity. Our description of the limiting NTK relies on some previous results describing the distribution of the network function f_θ at initialization. We will start by presenting this result and follow by a description of the limit of the NTK at initialization and during training.

Finally we will discuss what this result implies for the training dynamics of the network function $f_{\theta(t)}$ and the loss landscape around the gradient flow path.

Neural Networks as Gaussian Processes

At initialization, the network function $f_{\theta(0)}(\cdot)$ and more generally all pre-activations $\tilde{\alpha}^{(\ell)}(\cdot)$ are random functions, due to the randomness of the parameters. Conditionally on the activations of the previous layer $\alpha^{(\ell-1)}(\cdot)$ (or conditionally on the parameters up to the $(\ell-1)$-th layer $W_1, b_1, \dots, W_{\ell-1}, b_{\ell-1}$), the distribution of $\tilde{\alpha}^{(\ell-1)}(\cdot)$ is Gaussian with zero mean and covariance

$$Cov\left(\tilde{\alpha}_k^{(\ell-1)}(x), \tilde{\alpha}_m^{(\ell-1)}(y)\right) = \frac{1}{n_{\ell-1}}\left(\alpha^{(\ell-1)}(x)\right)^T \alpha^{(\ell-1)}(y)\delta_{km}$$

for any pair of inputs $x, y \in \mathbb{R}^{d_{in}}$ and neuron indices $k, m \in \{1, \dots, n_\ell\}$, where δ_{km} is the Kronecker delta.

This implies that the distribution of $\tilde{\alpha}^{(\ell-1)}(\cdot)$ is a mixture of Gaussians. Furthermore the conditioned distribution of $\tilde{\alpha}^{(\ell-1)}(\cdot)$ depends on the conditioned parameters only through the so-called *conjugate kernel*

$$\Sigma^{(\ell)}(x, y) = \frac{1}{n_{\ell-1}}\left(\alpha^{(\ell-1)}(x)\right)^T \alpha^{(\ell-1)}(y).$$

It turns out that in the infinite-width limit the conjugate kernels $\Sigma^{(\ell)}(x, y)$ converge to deterministic limits $\Sigma_\infty^{(\ell)}(x, y)$, which are independent of the parameters of the previous layer. As a result, the distribution of the pre-activations $\tilde{\alpha}^{(\ell)}(\cdot)$ becomes asymptotically Gaussian:

Proposition 1.1. *As $n_1, \dots, n_{L-1} \to \infty$, for a Lipschitz non-linearity σ, we have for any x, y*

$$\Sigma^{(\ell)}(x, y) \to \Sigma_\infty^{(\ell)}(x, y)$$

where $\Sigma_\infty^{(\ell-1)}(x, y)$ is defined recursively as

$$\Sigma_\infty^{(1)}(x, y) = x^T y + \beta^2$$
$$\Sigma_\infty^{(\ell+1)}(x, y) = \mathbb{E}_{u,v}\left[\sigma(u)\sigma(v)\right] + \beta^2$$

where the expectation is taken over pairs u, v sampled from $\mathcal{N}\left(0, \begin{pmatrix} \Sigma_\infty^{(\ell)}(x,x) & \Sigma_\infty^{(\ell)}(x,y) \\ \Sigma_\infty^{(\ell)}(y,x) & \Sigma_\infty^{(\ell)}(y,y) \end{pmatrix}\right)$.

Furthermore for all layer ℓ, the pre-activations $\tilde{\alpha}_k^{(\ell)}(\cdot)$ of each neuron k converge in law to i.i.d. centered Gaussian processes with covariance $\Sigma_\infty^{(\ell)}$.

Multiple versions of this result exist: the earliest appearance is in [159] but only for shallow networks ($L = 2$), it was then generalized in a sequence of papers [37, 42, 126, 46, 228]. The version presented here matches the one in our 2018 paper in Section 2, which only applies in the sequential limit, i.e. we first let $n_1 \to \infty$ then $n_2 \to \infty$ and so on until $n_{L-1} \to \infty$, for proofs in the simultaneous limit, i.e. when $n_1 = \cdots = n_{L-1} = w$ and $w \to \infty$, see [126, 46, 228].

In the simultaneous limit, the convergence of the conjugate kernels $\Sigma^{(\ell)}$ to their limit as a function of the width w is of order $w^{-\frac{1}{2}}$ [126, 46] which is expected as the proof consists in iterated law of large numbers for each layer.

Limit of the NTK at Initialization

In the same infinite-width limit the NTK $\Theta^{(L)}$ converges to a deterministic limit at initialization:

Theorem 1.1. *As $n_1, \ldots, n_{L-1} \to \infty$, for a Lipschitz non-linearity σ, the NTK $\Theta^{(L)}_{km}(x, y)$ converges to $\Theta^{(L)}_{\infty}(x, y)\delta_{km}$ for a deterministic kernel $\Theta^{(L)}_{\infty} : \mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{in}} \to \mathbb{R}$ defined recursively as*

$$\Theta^{(1)}_{\infty}(x, y) = \Sigma^{(1)}_{\infty}(x, y)$$
$$\Theta^{(\ell)}_{\infty}(x, y) = \Sigma^{(\ell)}_{\infty}(x, y) + \Theta^{(\ell-1)}_{\infty}(x, y)\dot{\Sigma}^{(\ell)}_{\infty}(x, y)$$

where $\dot{\Sigma}^{(\ell)}_{\infty}(x, y) = \mathbb{E}_{u,v}\left[\dot{\sigma}(u)\dot{\sigma}(v)\right]$ for u, v sampled from $\mathcal{N}\left(0, \begin{pmatrix} \Sigma^{(\ell-1)}_{\infty}(x, x) & \Sigma^{(\ell-1)}_{\infty}(x, y) \\ \Sigma^{(\ell-1)}_{\infty}(y, x) & \Sigma^{(\ell-1)}_{\infty}(y, y) \end{pmatrix}\right)$ and where $\dot{\sigma}$ is the derivative of σ.

Sketch of proof. The NTK $\Theta^{(L)}$ can be expressed in a recursive manner in terms of the NTK up to the last layer $\Theta^{(L-1)}$. The proof is by induction: the NTK up to the second layer $\Theta^{(2)}_{km}$ converges to deterministic limit $\Theta^{(2)}_{\infty}\delta_{km}$ as $n_1 \to \infty$ by a law of large number, which in turns allows one to prove the convergence of $\Theta^{(3)}$ as $n_2 \to \infty$ and so on and forth. □

Remark 1.1. Since the non-linearity σ is Lipschitz, its derivative is defined almost everywhere and therefore $\Theta^{(L)}_{km}(x, y)$ is almost surely well defined at initialization and the expectation $\dot{\Sigma}^{(\ell)}_{\infty}(x, y)$ is well defined.

The proof presented in our original paper [105] (see Section 2) is for the sequential infinite-width limit. Since then a number of generalization have been proven: for the simultaneous limit [128, 6] and for more general architectures [228] amongst others. In the simultaneous limit, the rate of convergence of the NTK to its limit as a function of w is $w^{-\frac{1}{2}}$ [128, 6, 95], like the convergence of the conjugate kernels $\Sigma^{(\ell)}$.

Theorem 1.1 shows that the tangent linear model at initialization $T_{\theta(0)}F(\theta) = f_{\theta(0)} + JF(\theta(0))\theta$ has a deterministic limiting kernel. In contrast, the features $\partial_{\theta_p} f_{\theta(0)}$ of the tangent model are random even in the infinite-width limit. This shows the advantage of working from a functional perspective: while the features $\partial_{\theta_p} f_{\theta(0)}$ remain random in the infinite-width limit, the NTK is asymptotically deterministic.

Limit of the NTK during Training

But the convergence of the NTK at initialization only describes the asymptotic derivative $\partial_t f_{\theta(t)}$ at initialization $t = 0$, i.e. it only describes the dynamics of $f_{\theta(t)}$ for very small times t. It turns out that as the width of the network grows, the rate of change of the NTK goes to zero, so that the limiting NTK is fixed in time:

Theorem 1.2. *(sketch) For a Lipschitz twice differentiable non-linearity σ and for a time T (defined in Section 2), we have, uniformly for all $t \in [0, T]$ and any $x, y \in \mathbb{R}^{d_{in}}$*

$$\lim_{n_1,\ldots,n_{L-1}\to\infty} \Theta^{(L)}_{\theta(t)}(x, y) = \Theta^{(L)}_{\infty}(x, y)I_{d_{out}}.$$

Sketch of proof. The proof relies on a recursive argument, which is formalized using Grönwall's Lemma: knowing the size of the NTK, we can bound how much the parameters move as a result of gradient flow, and knowing that the parameters have not moved much we can guarantee that the NTK is close to its initialization (and hence by Theorem 1.1 close to its limit $\Theta_\infty^{(L)}$). A surprising implication of the proof is that for large widths, the parameters move very little during training: there is a growing number of parameters but each of them moves less and less during training, resulting in a change of the vector of parameters $\|\theta(t) - \theta(0)\|$ (in Euclidean norm) of order 1, which is insufficient to affect the NTK for large widths. □

The details of how T can be chosen are in the paper [105] (see Section 2). Thanks to a number of follow-ups, this results has been generalized and improved:

- Regarding the time T, it was later shown that for the MSE T can be taken to be infinite (i.e. the convergence is uniform over all $t \in \mathbb{R}_+$) and that the result generalizes to gradient descent [6, 128]. More generally, exponential convergence (uniformly over all times $t \in \mathbb{R}_+$, with gradient flow or gradient descent) can be proven for any cost C that satisfies the Polyak-Lojasiewicz (PL) condition [142], which in particular includes all strictly convex costs.

- Our original proof was in the sequential limit, but it was later generalized to the simultaneous limit [6, 128]. While the rate of change of the NTK was at first only bounded by $w^{-\frac{1}{2}}$ [6, 128], it was observed empirically that the actual rate of change of the NTK is w^{-1} [128]. This lower rate of w^{-1} was later proven in [95] by introducing the Neural Tangent Hierarchy (NTH): a sequence of tensors, the first of which is the NTK, that each describe the derivative of the previous one.

This NTK analysis can also be interpreted as showing that the tangent linear model at initialization is good approximation of the non-linear model for large widths. In particular, in the infinite-width limit the model and its linearization match on the training path. This approach has been used to not only describe the dynamics of the network function $f_{\theta(t)}$, but also of the parameters $\theta(t)$ in wide networks [128].

Infinite-Width Dynamics

Now that we have described the infinite-width NTK, we can describe the training dynamics of the network function $f_{\theta(t)}$ for any cost C of the form $C(f) = \frac{1}{N}\sum c_i(f(x_i))$:

$$\begin{aligned} f_{\theta(0)} &\sim \mathcal{N}(0, \Sigma^{(L)}) \\ \partial_t f_{\theta(t)} &= -\frac{1}{N}\sum \Theta_\infty^{(L)}(x, x_i)\nabla c_i(f_{\theta(t)}(x_i)). \end{aligned}$$

The function $f_{\theta(0)}$ is initialized as a Gaussian process with covariance $\Sigma^{(L)}$ and then follows kernel gradient flow w.r.t. the limiting NTK $\Theta_\infty^{(L)}$ on the cost C. In other words, the dynamics on the non-linear model $F^{(L)}$ are asymptotically equivalent to the dynamics on the tangent linear model $T_{\theta(0)}F^{(L)}$ around initialization $\theta(0)$.

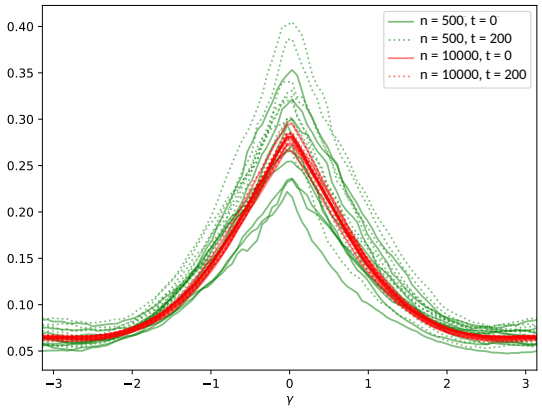


Figure 1.5.1: Convergence of the NTK to a fixed limit for two widths n and two times t.

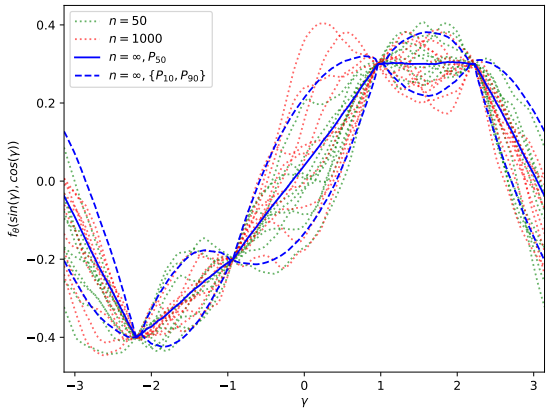


Figure 1.5.2: Networks function f_θ near convergence for two widths n and 10th, 50th and 90th percentiles of the asymptotic Gaussian distribution.

Explicit Formulas for the MSE

For the MSE loss (or any other quadratic loss) the network function $f_{\theta(t)}$ evolves according to a linear differential equation, which implies that $f_{\theta(t)}$ is Gaussian for all times t, with a mean and covariance that can be explicitely formulated (see Section 2.5).

The matrix of values on the training set $Y_{\theta(t)}$ is described by the following linear differential equation:

$$\partial_t Y_{\theta(t)} = \frac{2}{N}(Y - Y_{\theta(t)})\Theta_\infty^{(L)}(X, X)$$

where Y and $Y_{\theta(t)}$ are $d_{out} \times N$ matrices, and $\Theta_\infty^{(L)}(X, X)$ is the $N \times N$ Gram matrix with entries $\left(\Theta_\infty^{(L)}(X, X)\right)_{ij} = \Theta_\infty^{(L)}(x_i, x_j)$. Solving the linear differential equation, we obtain the evolution of $Y_{\theta(t)}$:

$$Y_{\theta(t)} = Y + (Y_{\theta(0)} - Y)e^{-t\Theta_\infty^{(L)}(X,X)},$$

whose expectation is $\mathbb{E}\left[Y_{\theta(t)}\right] = Y\left(I - e^{-t\Theta_\infty^{(L)}(X,X)}\right)$.

To describe the evolution of the whole function $f_{\theta(t)}$ we need to introduce the so-called *empirical integral operator* T_K^N for a kernel $K : \mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{in}} \to \mathbb{R}$ (though we are mostly interested in the case $K = \Theta_\infty^{(L)}$) which maps any function $f : \mathbb{R}^{d_{in}} \to \mathbb{R}$ to another function $T_K^N(f) : \mathbb{R}^{d_{in}} \to \mathbb{R}$ with values

$$T_K^N(f)(x) = \frac{1}{N}\sum_{i=1}^{N} K(x, x_i)f(x_i).$$

In the infinite-width limit, each of the outputs $f_{\theta(t),k} : \mathbb{R}^{d_{in}} \to \mathbb{R}$ for $k \in \{1, \ldots, d_{out}\}$ evolves independently according to the linear differential equation

$$\partial_t f_{\theta(t),k} = 2T_{\Theta_\infty^{(L)}}^N(f_k^* - f_{\theta(t),k})$$

where $f^* : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ is the *true function* we are trying to fit, which maps the inputs X to the outputs Y, i.e. $f^*(x_i) = y_i$ for all $i = 1, \dots, N$.

Solving the above equation, we obtain that

$$f_{\theta(t),k} = f_{\theta(0),k} + \left(I - e^{-2tT^N_{\Theta^{(L)}_\infty}}\right)(f^*_k - f_{\theta(0),k}).$$

Since the network function at initialization has zero mean, i.e. $\mathbb{E}\left[f_{\theta(0),k}(x)\right] = 0$, the expected network function throughout training is equal to

$$\mathbb{E}\left[f_{\theta(t),k}\right] = \left(I - e^{-2tT^N_{\Theta^{(L)}_\infty}}\right) f^*_k.$$

Since the different outputs evolve independently in the limit we can assume that $d_{out} = 1$ without loss of generality.

Similarity with Kernel Ridge Regression

The expected network function is similar to the so-called *Kernel Ridge Regression (KRR)* predictor which is defined by [2]

$$\hat{f}_\lambda(x) = K(x, X)\left(K(X, X) + N\lambda I\right)^{-1} Y,$$

for some positive ridge parameter λ, where $K(x, X)$ is a N-dimensional row vector with entries $K(x, x_i)$ for $i \in \{1, \dots, N\}$. The $\hat{f}_\lambda$ KRR predictor can also be expressed in terms of the integral operator T^N_K:

$$\hat{f}_\lambda = T^N_K \left(T^N_K + \lambda I\right)^{-1} f^*.$$

This formula illustrates the link between the expected network function $\mathbb{E}\left[f_{\theta(t),k}\right]$ and the KRR predictor with the NTK ($K = \Theta^{(L)}_\infty$). The operators $\left(I - e^{-2tT^N_{\Theta^{(L)}_\infty}}\right)$ and $T^N_{\Theta^{(L)}_\infty}\left(T^N_{\Theta^{(L)}_\infty} + \lambda I\right)^{-1}$ share the same eigenfunctions, they only differ in their eigenvalues. Given an eigenvalue λ_i of empirical integral operator $T^N_{\Theta^{(L)}_\infty}$, the corresponding eigenvalue of $\left(I - e^{-2tT^N_{\Theta^{(L)}_\infty}}\right)$ is $1 - e^{-2t\lambda_i}$ while the corresponding eigenvalue of $T^N_{\Theta^{(L)}_\infty}\left(T^N_{\Theta^{(L)}_\infty} + \lambda I\right)^{-1}$ is $\frac{\lambda_i}{\lambda + \lambda_i}$. Both result in a form of cutoff of the small eigenvalues: $1 - e^{-2t\lambda_i}$ is close to 1 if $\lambda_i \gg \frac{1}{2t}$ and close to 0 if $\lambda_i \ll \frac{1}{2t}$ while $\frac{\lambda_i}{\lambda + \lambda_i}$ is close to 1 when $\lambda_i \gg \lambda$ and close to 0 when $\lambda_i \ll \lambda$. In other terms $\left(I - e^{-tT^N_{\Theta^{(L)}_\infty}}\right)$ and $T^N_{\Theta^{(L)}_\infty}\left(T^N_{\Theta^{(L)}_\infty} + \lambda I\right)^{-1}$ are 'smooth' approximate projections to the space spanned by the eigenvectors of $T_{\Theta^{(L)}_\infty}$ with eigenvalues larger than $\frac{1}{2t}$ resp. λ (they are smooth in the sense that they are continuous w.r.t. to changing t or λ).

The expected network function $\mathbb{E}\left[f_{\theta(t)}\right]$ at a time t is similar to the KRR predictor $\hat{f}_\lambda$ with ridge parameters $\lambda = \frac{1}{2t}$. This similarity illustrates the regularizing effect of early stopping in DNNs.

[2] The KRR predictor is often defined without the N factor in front of the λ, this change has little impact, since λ can be chosen as any positive real number. This definition leads to a nicer theoretical analysis of the large N behavior of $\hat{f}_\lambda$, as discussed in Section 1.6.

In the limit as $t \to +\infty$ (and $\lambda \searrow 0$), this approximation becomes an equality: we have

$$\lim_{t \to +\infty} \mathbb{E}\left[f_{\theta(t)}\right] = \lim_{\lambda \searrow 0} \hat{f}_\lambda = P_{\mathrm{Im} T^N_{\Theta^{(L)}_\infty}} f^*$$

where $P_{\mathrm{Im} T^N_{\Theta^{(L)}_\infty}}$ is the projection to the image of the operator $T^N_{\Theta^{(L)}_\infty}$.

Kernel Ridge Regression is a well known method and its generalization properties are well-studied [238, 150, 181]. The expected risk (or test error) of KRR is described in 1.6 and the similarity presented in this section suggests that we can expect approximately the same risk of for the expected network function $\mathbb{E}\left[f_{\theta(t)}\right]$ at a time $t = \frac{1}{2\lambda}$.

Loss Landscape Perspective

Though the NTK analysis that we just introduced shows that the dynamics of DNNs can be studied in function space directly, it is interesting to understand what these dynamics look like in parameters space. A lot of work has been done to study the $P \times P$ Hessian $\mathcal{HL}(\theta)$ of the loss landscape $\mathcal{L}$, both empirically [189, 190] and mathematically [38, 171, 172, 112]. The aforementioned theoretical results apply to the Hessian at initialization and the complexity of the training dynamics makes it difficult to extend these results to later training times. The NTK analysis allowed us to overcome this hurdle and describe properties of the Hessian throughout training in a paper [106] which is reproduced in full in Section 3.

There is a direct link between the NTK and the Hessian of the loss $\mathcal{L}$ of DNNs. Since $\mathcal{L}(\theta) = C(Y_\theta)$, the Hessian of $\mathcal{L}$ equals the sum of two $P \times P$ matrices

$$\mathcal{HL}(\theta) = I + S = \left(JY_\theta(\theta)\right)^T \mathcal{H}C(Y_\theta) JY_\theta(\theta) + \nabla C(Y_\theta) \cdot \mathcal{H}Y_\theta$$

where the Jacobian $JY_\theta(\theta)$ is understood as a $Nd_{out} \times P$ matrix, the Hessian $\mathcal{H}C(Y_\theta)$ is a $Nd_{out} \times Nd_{out}$ matrix and $\mathcal{H}Y_\theta$ is a $Nd_{out} \times P \times P$ tensor which is multiplied with the Nd_{out} vector $\nabla C(Y_\theta)$ to obtain a $P \times P$ matrix.

The first matrix is the so-called the Fisher Information Matrix (FIM) I and it is a positive matrix (since for convex C the Hessian $\mathcal{H}C(Y_\theta)$ is positive).

The second matrix S typically has positive and negative eigenvalues and it vanishes at a global minimum since $\nabla C(Y_\theta) = 0$. The matrix S is directly related to the non-linearity of the model $F^{(L)}$ since it vanishes in linear models.

For the MSE, we have $\mathcal{H}C(Y_\theta) = \frac{2}{N} I_{Nd_{out}}$, so that the FIM $I = \frac{2}{N} \left(JY_\theta(\theta)\right)^T JY_\theta(\theta)$ is equal (up to a scaling of $\frac{2}{N}$) to the dual of NTK Gram matrix $\Theta^{(L)}(X, X) = JY_\theta(\theta) \left(JY_\theta(\theta)\right)^T$, which implies that the two matrices have matching non-zero eigenvalues. A direct consequence of our NTK analysis is that, in the infinite-width limit, at the end of training $t \to \infty$, we reach a global minimum where the Hessian equals the FIM whose spectrum is equal to the limiting NTK Gram matrix $\Theta^{(L)}_\infty(X, X)$. Furthermore this implies that the spectrum of the FIM is asymptotically fixed in time, which means that results which described the spectrum of the FIM at initialization [172, 112] could be extended to describe the FIM at the end of training.

In the paper [106], we give a full description of the asymptotic moments $\mathrm{Tr}\left[\left(\mathcal{HL}(\theta(t))\right)^k\right]$ of the spectrum of the Hessian from the following observations:

- The matrices I and S are asymptotically orthogonal in the sense that their moments asymptotically add up $\mathrm{Tr}\left(I + S\right)^k \approx \mathrm{Tr} I^k + \mathrm{Tr} S^k$, this allows us to compute the asymptotic moments of $\mathcal{HL}(\theta) = I + S$ from the moments of I and S.

- The moments of I are the moments of the NTK Gram matrix.
- We express the first two moments of S for any time t by introducing some other kernels, defined in a similar manner to the limiting NTK $\Theta_\infty^{(L)}$ (for the full formulas see Section 3). The first two asymptotic moments are non-zero at initialization and decay to zero. All higher moments of S vanish in the infinite-width limit.

For the MSE, the differential equations describing the evolution of the first two moments of S can be solved and we obtain explicit formulas for all time t.

These results suggest the following interpretation of the spectrum of I and S: as the width of the network grows the FIM I has a bounded (at most Nd_{out}) number of eigenvalues of order 1 while the matrix S has a growing number of eigenvalues whose magnitude goes to zero (S has a non-vanishing Frobenius norm but a vanishing operator norm).

All the mixed terms in the trace of the Hessian $\mathrm{Tr}\,(I+S)^k$ such as $\mathrm{Tr}\,(IS)$ or $\mathrm{Tr}\,(ISSISI)$ vanish since the mixed products (i.e. IS or $ISSISI$) have a finite number of vanishing eigenvalues, hence leading to the asymptotic orthogonality. Empirically, we observe that this orthogonality might be stronger, in the sense that I is large along directions where S is small and vice-versa, see Section 3 for more details.

These results can also be interpreted in terms of the geometry of the loss surface around the training path $\{\theta(t) : t \in \mathbb{R}_+\}$. For the MSE, the loss is almost quadratic in the region, with the Hessian being almost constant in the region I (the fixed FIM I asymptotically dominates the S matrix in operator norm). This is in line with the observation that in the infinite-width limit the DNN model becomes equivalent to its linearization, i.e. the wider the network the more it behaves as a linear model.

As usual in a quadratic cost, what determines the speed of convergence is the conditioning $\kappa = \lambda_{max}/\lambda_{min}$ of the Hessian, which in our case equals the conditioning of the limiting NTK Gram matrix $\Theta_\infty^{(L)}(X,X)$, a large conditioning implies that the loss has a narrow valley structure with large eigenvalues corresponding to the 'cliffs' of the valley, where the loss increases very fast, forcing a very small learning rate (at most $2/\lambda_{max}$) and this small learning means that training is going to be very slow along the bottom of the valley which correspond to the small eigenvalues. In contrast a small condition number allows for fast convergence. This interpretation of the loss surface is in line with other results such as [128, 142].

1.6 Generalization of Kernel Ridge Regression

As explained in Section 1.5, for infinitely-wide DNNs trained with the MSE, the expected network function $\mathbb{E}\,[f_\theta]$ at the end of training ($t \to \infty$) is the ridgeless ($\lambda \searrow 0$) KRR predictor $\hat{f}_{\lambda \searrow 0}$, and for finite times $t < +\infty$, the expected network function $\mathbb{E}\,[f_\theta]$ is similar to the KRR predictor $\hat{f}_\lambda$ with $\lambda = 1/2t$. Motivated by this similarity, we studied the test error of KRR in a NeurIPS 2020 paper [103] with Berfin Şimşek, Francesco Spadaro, Franck Gabriel and Clément Hongler. This section summarizes these results and the original paper can be found in Section 4.

Statistical Learning Setup

To study the test error of KRR, we need to describe how the training and test data are sampled. We assume that all the training inputs x_i are sampled i.i.d. from a probability measure π over a domain

$\Omega \subset \mathbb{R}^{d_{in}}$ and the outputs are of the form $y_i^\epsilon = f^*(x_i) + \epsilon e_i$ for a *true function* $f^* : \mathbb{R}^{d_{in}} \to \mathbb{R}$, a *noise intensity* ϵ and some i.i.d. standard Gaussian noise variables $e_i \sim \mathcal{N}(0,1)$. Remember that for any $\lambda > 0$, we define the Kernel Ridge Regression w.r.t. to a kernel $K : \Omega \times \Omega \to \mathbb{R}$ as the function[3]

$$\hat{f}_\lambda^\epsilon(x) = \frac{1}{N} K(x, X) \left(\frac{1}{N} K(X, X) + \lambda I_N \right)^{-1} Y^\epsilon$$

where Y^ϵ is the vector of dimension N containing all of the outputs y_i^ϵ - in this section we only consider the single output case ($d_{out} = 1$) so that Y^ϵ is simply a vector of dimension N.

The training loss or empirical risk is the MSE

$$\hat{R}^\epsilon(\hat{f}_\lambda^\epsilon) = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{f}_\lambda^\epsilon(x_i) - y_i^\epsilon \right)^2 .$$

The risk is the expected error on a new data point x sampled from the distribution π independently from the training set:

$$R^\epsilon(\hat{f}_\lambda^\epsilon) = \mathbb{E}_{x,e} \left[\left(\hat{f}_\lambda^\epsilon(x) - f^*(x) + \epsilon e \right)^2 \right]$$

where e is a standard Gaussian noise variable independent of all other variables. The risk can be simplified to $R^\epsilon(\hat{f}_\lambda^\epsilon) = \mathbb{E}_{x,e} \left[\left(\hat{f}_\lambda^\epsilon(x) - f^*(x) \right)^2 \right] + \epsilon^2$.

Our goal is to describe the typical risk $R^\epsilon(\hat{f}_\lambda^\epsilon)$ when the number of datapoints N is large. More precisely, we will study the expected risk $\mathbb{E}_{X,E} \left[R^\epsilon(\hat{f}_\lambda^\epsilon) \right]$, taking the expectation over the sampling of the training set X and the noise variables $E = (e_1, \dots, e_N)$.

The expected risk is decomposed into a bias and a variance term:

$$\mathbb{E}_{X,E} \left[R^\epsilon(\hat{f}_\lambda^\epsilon) \right] = R^\epsilon \left(\mathbb{E}_{X,E} \left[\hat{f}_\lambda^\epsilon \right] \right) + \mathbb{E}_x \left[\mathrm{Var}_{X,E} \left(\hat{f}_\lambda^\epsilon(x) \right) \right] .$$

The goal is to identify the optimal choice of ridge λ and to describe how fast the risk goes to zero as N grows. This rate depends on a notion of alignement of the true function f^* with the kernel K, hence giving an idea of what functions are 'easy' and 'hard' to learn with KRR (and as an extension with infinitely wide DNNs).

Note the contrast with the results up to this point, where the randomness was coming from the random initialization of the parameters. In this setting, there is no random initialization of parameters. Instead the randomness comes from the sampling of the inputs.

Functional Perspective

Taking a functional perspective will again be useful for this problem. Instead of thinking of the training set $x_1, \dots, x_N$ as a set of points of the input domain Ω, one can think of them as linear maps $o_i : \mathcal{C} \to \mathbb{R}$ (where $\mathcal{C}$ is the space of continuous functions from Ω to the reals), mapping a function $f : \Omega \to \mathbb{R}$ to its value $f(x_i)$ at x_i. Together they form a random linear operator $\mathcal{O} : \mathcal{C} \to \mathbb{R}^N$

[3] If one multiplies the inside and the outside of the parenthesis by N we recover the definition of the KRR predictor given in Section 1.5.

mapping a function f to the vector $(f(x_1), \ldots, f(x_N))^T$ of values of f on the training set which is called the *sampling operator* $\mathcal{O}$.

The noiseless KRR predictor $f_\lambda^{\epsilon=0}$ can be written as $A_\lambda^N f^*$ for a random operator $A_\lambda^N : \mathcal{C} \to \mathcal{C}$ of the form

$$A_\lambda^N = \frac{1}{N} K\mathcal{O}^T \left(\frac{1}{N} \mathcal{O} K \mathcal{O}^T + \lambda I_N \right)^{-1} \mathcal{O}$$

where the kernel K is understood as an operator $K : \mathcal{C}^* \to \mathcal{C}$ from the dual space $\mathcal{C}^*$ to the primal and where $\mathcal{O}^T$ is the adjoint $\mathcal{O}^T : \mathbb{R}^N \to \mathcal{C}^*$ of the sampling operator $\mathcal{O}$, i.e. for any vector z and any continuous function $f : \Omega \to \mathbb{R}$, we have $\left(\mathcal{O}^T z\right)(f) = \sum_{i=1}^N z_i o_i(f)$.

The operator A_λ^N is closely related to the so-called *integral operator* $T_K : \mathcal{C} \to \mathcal{C}$ and the *empirical integral operator* $T_K^N : \mathcal{C} \to \mathcal{C}$ which map a function $f : \Omega \to \mathbb{R}$ to a functions with respective values

$$\left(T_K f\right)(x) = \int_\Omega K(x, z) f(z) d\pi(z)$$
$$\left(T_K^N f\right)(x) = \frac{1}{N} \sum_{i=1}^N K(x, x_i) f(x_i).$$

Clearly we have $T_K^N(x) \to T_K(x)$ as $N \to \infty$.

The empirical integral operator can also be expressed in terms of the sampling operator as $T_K^N = K\mathcal{O}^T\mathcal{O}$, allowing us to rewrite A_λ^N in terms of T_K^N:

$$A_\lambda^N = T_K^N \left(T_K^N + \lambda I_{\mathcal{C}}\right)^{-1} = I_{\mathcal{C}} - \lambda \left(T_K^N + \lambda I_{\mathcal{C}}\right)^{-1}.$$

The operator $\left(T_K^N + \lambda I_{\mathcal{C}}\right)^{-1}$ is the so-called *resolvent* of T_K^N (at $-\lambda$). Amongst other uses, the resolvent is one of the central tools used in random matrix theory to understand the spectrum of random matrices. The following analysis of A_λ^N is inspired from previous work [203] which studies so-called general Wishart matrices (matrices of the form TW^TW for a deterministic $m \times m$ matrix T and a random $k \times m$ matrix W with i.i.d. entries) which are very similar to the random operator $T_K^N = K\mathcal{O}^T\mathcal{O}$.

Remark 1.2. Readers familiar with random matrix theory might wonder why we did not simply apply these previous results to our setting. Our setting is distinct from the typical random matrix theory setting in a few ways:

- We are studying random operators instead of random matrices.
- We are interested in finite-N approximations. In contrast, most results on the spectrum of general Wishart matrices are in the limit where m and k go to infinity with a fixed ratio γ. In our case m is in a sense $+\infty$ while k is finite (it is the number of datapoints N).
- In the aforementioned type of analysis, the spectrum of the deterministic operator T must converge to a continuous measure as $m \to \infty$. In our setting T_K is fixed and its spectrum does not have a continuous measure, instead the operator T_K has typically a countable number of eigenvalues which decay to zero (the rate of decay is determined by the regularity of the kernel K, e.g. if K is smooth then the decay is exponential).

- To approximate the expected risk for finite N, we need to describe fluctuations of the resolvent (i.e. approximate the rescaled variance of the resolvent). The difficulty to compute these fluctuations made a full description of the risk impossible in previous works [48, 144].

Inspired by Random Matrix Theory, we assume universality, i.e. that the large N statistics of the resolvent $\left(T_K^N + \lambda I_{\mathcal{C}}\right)^{-1}$ – and as an extension of A_λ^N – are the same if we replace the sampling operator $\mathcal{O}$ with a centered Gaussian equivalent $\tilde{\mathcal{O}}$ (a random Gaussian operator with zero mean and a covariance that matches the covariance of $\mathcal{O}$). From now on we assume that $\mathcal{O}$ is Gaussian; for more details see Section 4.

Signal Capture Threshold

For a fixed λ, the operator $A_\lambda^N = T_K^N \left(T_K^N + \lambda I_{\mathcal{C}}\right)^{-1}$ converges to the operator $A_\lambda = T_K \left(T_K + \lambda I_{\mathcal{C}}\right)^{-1}$ as $N \to \infty$, but our goal is to understand the behavior of the expected risk $\mathbb{E}\left[R^\epsilon(\hat{f}_\lambda^\epsilon)\right]$ for finite but large N, for which we need to describe the mean and variance of A_λ^N for such finite N.

The central object describing the mean and variance of A_λ^N for finite N is the the Signal Capture Threshold $\vartheta(\lambda)$, which is the unique positive solution to the equation

$$\vartheta = \lambda + \frac{\vartheta}{N} \mathrm{Tr}\left[A_\vartheta\right].$$

The trace $\mathrm{Tr}\left[A_\vartheta\right] = \mathrm{Tr}\left[T_K \left(T_K + \vartheta I_{\mathcal{C}}\right)^{-1}\right]$ is well defined since it is bounded by $\frac{1}{\lambda}\mathrm{Tr}T_K$ and $\mathrm{Tr}T_K = \mathbb{E}_{x\sim\pi}\left[K(x,x)\right] < \infty$. The SCT is bounded from below by the ridge parameter: $\vartheta \geq \lambda$.

We first show in Theorem 4.1 in Section 4.3 that the mean $\mathbb{E}_{\mathcal{O}}\left[A_\lambda^N\right]$ is $O(\frac{1}{N})$ close[4] to the operator $A_{\vartheta(\lambda)}$.

This first result motivates the name Signal Capture Threshold: assume that the true function f^* is an eigenfunction with eigenvalue d of the integral operator T_K (i.e. $T_K f^* = d f^*$), then the expected predictor $\mathbb{E}\left[\hat{f}_\lambda\right]$ is approximately equal to $\frac{d}{d+\vartheta} f^*$; if $d \ll \vartheta$ then $\frac{d}{d+\vartheta} \approx 0$ 'the signal is lost' and if $d \gg \vartheta$ then $\frac{d}{d+\vartheta} \approx 1$ 'the signal is captured'. More generally, if we write $f^* = \sum_{k=1}^\infty b_k f^{(k)}$ for some coefficients $b_k \in \mathbb{R}$ and where $f^{(k)}$ is the k-th eigenfunction (with eigenvalue λ_k) of T_K, then the expected predictor will learn along the eigenfunctions with eigenvalues above the signal capture threshold and not learn along the eigenfunctions below it.

This description of the mean predictor leads to an approximation of the bias term:

$$R^\epsilon\left(\mathbb{E}_{X,E}\left[\hat{f}_\lambda^\epsilon\right]\right) \approx \left\|\left(I_C - A_{\vartheta(\lambda)}\right) f^*\right\|_\pi^2 + \epsilon^2$$

where $\|\cdot\|_\pi^2$ is the ℓ_2-norm $\|f\|_\pi^2 = \int_\Omega f(x)^2 d\pi(x)$ over the distribution π.

We then describe the variance of the predictor $\hat{f}_\lambda$ along each of the eigenfunctions $f^{(k)}$ of T_K:

$$\mathrm{Var}_{X,E}\left(\left\langle f^{(k)}, \hat{f}_\lambda \right\rangle_\pi\right) \approx \frac{\partial_\lambda \vartheta(\lambda)}{N}\left(\left\|\left(I_C - A_{\vartheta(\lambda)}\right) f^*\right\|_\pi^2 + \epsilon^2 + \left\langle f^{(k)}, f^* \right\rangle_\pi \frac{\vartheta(\lambda)^2}{\left(d_k + \vartheta(\lambda)\right)^2}\right) \frac{d_k^2}{\left(d_k + \vartheta(\lambda)\right)^2},$$

where $\langle\cdot,\cdot\rangle_\pi$ is the scalar product $\langle f, g\rangle_\pi = \int_\Omega f(x) g(x) d\pi(x)$ over the distribution π.

[4] See Section 4.3 for more precise bounds, in terms of the ridge λ.

This description of the variance allows us to approximate the variance term in the expected risk:

$$\mathbb{E}_x\left[\mathrm{Var}_{X,E}\left(\hat{f}^\epsilon_\lambda(x)\right)\right] \approx (\partial_\lambda \vartheta(\lambda) - 1)\left(\left\|\left(I_C - A_{\vartheta(\lambda)}\right) f^*\right\|^2_\pi + \epsilon^2\right),$$

yielding an approximation of the expected risk in terms of the SCT:

$$\mathbb{E}_{X,E}\left[R^\epsilon(\hat{f}^\epsilon_\lambda)\right] \approx \partial_\lambda \vartheta(\lambda)\left(\left\|\left(I_C - A_{\vartheta(\lambda)}\right) f^*\right\|^2_\pi + \epsilon^2\right).$$

This approximation in terms of the SCT suggests a multiplicative version of the bias/variance tradeoff (instead of the traditional additive form):

- The bias term $\left\|\left(I_C - A_{\vartheta(\lambda)}\right) f^*\right\|^2_\pi + \epsilon^2$, represents how much signal is captured. As the ridge goes to zero $\lambda \searrow 0$, the SCT $\vartheta(\lambda)$ decreases and so does the bias term, since we capture more of the signal. Note however that for a fixed N it is impossible to capture all the signal: even in the limit $\lambda \searrow 0$ the SCT converges the solution of the equation $\mathrm{Tr}\left[A_{\vartheta(0)}\right] = N$ (which is generally positive). There is an interesting interpretation for this equation: let $f^{(k)}$ be an eigenfunction of T_K with eigenvalue λ_k, then $f^{(k)}$ is an eigenfunction of $A_{\vartheta(0)}$ with eigenvalue $\frac{\lambda_i}{\lambda_i + \vartheta(0)}$ which is close to 1 if $\lambda_i > \vartheta(0)$ and close to zero if $\lambda_i < \vartheta(0)$; the trace $\mathrm{Tr}\left[A_{\vartheta(0)}\right]$ measures in a sense the number of eigenfunctions along which the signal is captured and the equation $\mathrm{Tr}\left[A_{\vartheta(0)}\right] = N$ ensures that this number equals the number of datapoints N. This makes intuitive sense, if we receive information of dimension N (in the form of the labels Y) then we can only capture information of the same dimension, i.e. we only capture the signal along roughly the N largest eigenfunctions of T_K.
- The derivative $\partial_\lambda \vartheta(\lambda)$ plays the role of the variance term, which interestingly does not depend on the true function f^*. The derivative $\partial_\lambda \vartheta(\lambda)$ grows as λ becomes smaller and may explode as $\lambda \searrow 0$. The optimal ridge λ (and by extension the optimal time T at which we stop training an infinitely wide DNN) is determined by a tradeoff, where decreasing λ leads to capturing more signal but at the risk of an explosion of variance.

This result also shows that the functions that are easy to learn for KRR with a kernel K are those whose signal decays rapidly along the eigenfunctions of the integral operator T_K, so that most of the signal is contained along the first eigenfunctions. The faster the decay, the faster the term $\left\|\left(I_C - A_{\vartheta(\lambda)}\right) f^*\right\|^2_\pi$ decays, leading to a smaller test error.

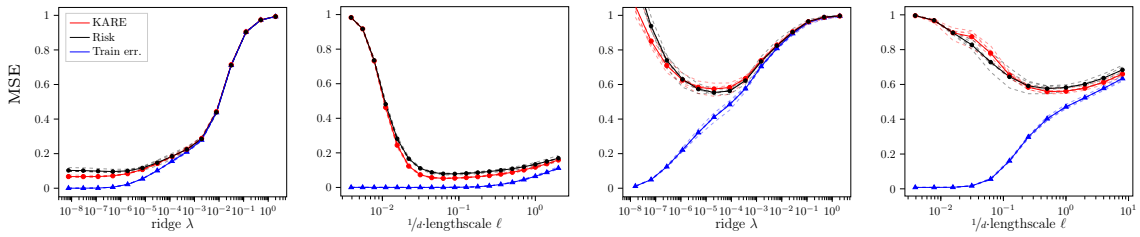
Kernel Alignement Risk Estimator

In practice, it is difficult to use the approximation of the expected risk in terms of the SCT ϑ presented in the previous section to real data, since we have typically no information on the data distribution π, and we therefore cannot compute the integral operator T_K nor the SCT ϑ. It turns out that the risk can also be approximated by the *Kernel Alignement Risk Estimator (KARE):*

$$\mathbb{E}\left[R^\epsilon(\hat{f}^\epsilon_\lambda)\right] \approx \frac{\frac{1}{N}\left(Y^\epsilon\right)^T\left(\frac{1}{N}K(X,X) + \lambda I_N\right)^{-2} Y^\epsilon}{\left(\frac{1}{N}\mathrm{Tr}\left[\left(\frac{1}{N}K(X,X) + \lambda I_N\right)^{-1}\right]\right)^2}.$$

The KARE depends only on the training labels Y^ϵ and the kernel Gram matrix $K(X,X)$ of the training inputs: it can therefore be computed from the training data.

The KARE is motivated by the following three facts:



(a) MNIST, $\ell = d$ (b) MNIST, $\lambda = 10^{-5}$ (c) Higgs, $\ell = d$ (d) Higgs, $\lambda = 10^{-4}$

Figure 1.6.1: Comparison between the KRR risk and the KARE for various choices of normalized lengthscale ℓ/d and ridge λ on the MNIST dataset (restricted to the digits 7 and 9, labeled by 1 and -1 respectively, $N = 2000$) and on the Higgs dataset (classes 'b' and 's', labeled by -1 and 1, $N = 1000$) with the RBF Kernel $K(x, x') = \exp(-\|x-x'\|_2^2/\ell)$. KRR predictor risks, and KARE curves (shown as dashed lines, 5 samples) concentrate around their respective averages (solid lines).

1. The mean risk $\mathbb{E}_{X,E}\left[R^\epsilon(\hat{f}_\lambda^\epsilon)\right]$ and empirical risk $\mathbb{E}_{X,E}\left[\hat{R}^\epsilon(\hat{f}_\lambda^\epsilon)\right]$ are related by the following formula $\mathbb{E}_{X,E}\left[R^\epsilon(\hat{f}_\lambda^\epsilon)\right] \approx \frac{\vartheta^2}{\lambda^2}\mathbb{E}_{X,E}\left[\hat{R}^\epsilon(\hat{f}_\lambda^\epsilon)\right]$.
2. The empirical risk can be written as $\hat{R}^\epsilon(\hat{f}_\lambda^\epsilon) = \frac{\lambda^2}{N}\left(Y^\epsilon\right)^T\left(\frac{1}{N}K(X,X) + \lambda I_N\right)^{-2} Y^\epsilon$.
3. The SCT $\vartheta(\lambda)$ can be approximated by[5] $\frac{1}{\frac{1}{N}\mathrm{Tr}\left[\left(\frac{1}{N}K(X,X)+\lambda I_N\right)^{-1}\right]}$.

Assuming that both the risk and the expected risk concentrate in their expectation for large N, we recover the KARE from the three above considerations.

The KARE suggest that we can have a small test error when the numerator is small and the denominator is large:

- The numerator $\frac{1}{N}\left(Y^\epsilon\right)^T\left(\frac{1}{N}K(X,X) + \lambda I_N\right)^{-2} Y^\epsilon$ is small when Y^ϵ and $K(X,X)$ are 'aligned' in the sense that $K(X,X)$ is large along the direction of Y^ϵ.
- The denominator is large when $K(X,X)$ is small.

We therefore want the kernel Gram matrix to be large along Y^ϵ and small along all the other directions. In the noiseless case $\epsilon = 0$, the optimal choice of kernel would be the kernel $K^*(x,y) = f^*(x)f^*(y)$ where f^* is the true function, so that the Gram matrix is of the form $K(X,X) = Y^\epsilon\left(Y^\epsilon\right)^T$. As $\lambda \searrow 0$ the numerator remains upper bounded while the denominator scales as λ^{-2}, so that the KARE is of order λ^2. Of course, such a choice of kernel would imply that one has knowledge of the true function f^*, which explains why one can reach zero test loss with only a finite number of datapoints.

In practice we do not know the true function f^* but we might have some prior knowledge: for example we might expect the Fourier coefficients of the true function to decay rapidly, in which case it makes sense to choose a so-called translation-invariant kernel such as the so-called Radial Basis Function (RBF) kernel $K(x,y) = \exp(-\frac{\|x-y\|^2}{2h^2})$.

[5] For those familiar, this is the reciprocal of the Stieljes transform of the kernel Gram matrix.

Convergence Speed and Generalization

The KARE also implies the existence of a tradeoff between convergence speed and generalization. As discussed in Section 1.5, the number of steps to reach a certain train error scales with the condition number $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$ of the kernel Gram matrix (where λ_{min} and λ_{max} are the smallest and largest eigenvalues of the kernel Gram matrix $K(X, X)$) and the fastest convergence is when the kernel Gram matrix equals the identity (or a scaling thereof) with a condition number of 1.

In contrast, the KARE is lower bounded by

$$\left(\frac{\lambda_{min}}{\lambda_{max}}\right)^2 \frac{1}{N} \|Y^\epsilon\|^2 = \kappa^{-2} \frac{1}{N} \|Y^\epsilon\|^2 .$$

The term $\frac{1}{N} \|Y^\epsilon\|^2$ concentrates in $\|f\|_\pi^2 + \epsilon^2$ for large N, which is the risk of the zero predictor ($\hat{f}(x) = 0$ for all x), intuitively it represents the test error when nothing is learned. As a result, for the KARE (and by extension the test error) to go to zero, the conditioning $\kappa = \frac{\lambda_{max}}{\lambda_{min}}$ must be very large.

Note however that a fast decay does not guarantee good generalization, it is only a necessary condition.

1.7 Spectral Bias of DNNs

The so-called *spectral bias* of DNNs, observed empirically in [175, 227, 226], denotes a tendency of DNNs to learn low frequencies faster than high frequencies.

In the infinite-width limit, this phenomenon is directly related to the spectrum of the NTK Gram matrix $\Theta_\infty^{(L)}(X, X)$. If $v \in \mathbb{R}^N$ is an eigenvector of $\Theta_\infty^{(L)}(X, X)$ with eigenvalue d, the error along v will go to zero at a rate of e^{-dt} (this follows from the dynamics described in Section 1.5), i.e. the larger the eigenvalue, the faster the network learns along v. What happens informally is that top eigenvectors of the NTK Gram matrix $\Theta_\infty^{(L)}(X, X)$ tends to be 'low frequency' or 'smoother' while the bottom eigenvectors are 'higher frequency' or 'rougher'.

For some data distributions and in the *population limit* $N \to \infty$, we can formalize these notions, since the eigendecomposition of the NTK Gram matrix then matches a classical spectral decomposition: the spherical harmonics.

Note that the eigenvalues of the $\Theta_\infty^{(L)}(X, X)$ are the same (up to a scaling) as those of the empirical integral operator $T^N_{\Theta_\infty^{(L)}}$: if v is an eigenvector of $\Theta_\infty^{(L)}(X, X)$ eigenvalue d, the function $x \mapsto \Theta_\infty^{(L)}(x, X)\Theta_\infty^{(L)}(X, X)^{-1}v)$ is an eigenfunction of $T^N_{\Theta_\infty^{(L)}}$ with eigenvalue $\frac{d}{N}$. In the population limit, as $N \to \infty$, the empirical integral operator $T^N_{\Theta_\infty^{(L)}}$ converges to the integral operator $T_{\Theta_\infty^{(L)}}$ whose eigendecomposition can in some cases be easier to describe.

Consider the uniform distribution π on the hyper-sphere $\mathbb{S}^{d_{in}-1}$. Since the limiting NTK $\Theta_\infty^{(L)}$ is rotationally invariant (i.e. for any orthogonal transformation $\Theta_\infty^{(L)}(Ox, Oy) = \Theta_\infty^{(L)}(x, y)$) the eigenfunctions of $T_{\Theta_\infty^{(L)}}$ are the (hyper-)spherical harmonics. The spherical harmonics have a degree k and all harmonics of the same degree have the same eigenvalue λ_k. The spherical harmonics generalize the notion of Fourier analysis from the plane to the sphere. The degree k is a notion of frequency: the only spherical harmonics of degree 0 is the constant function and higher harmonics become more oscillating (harmonics of degree k are homogeneous polynomials of degree k). In the 2D case ($d_{in} = 2$) the link between spherical harmonics and Fourier analysis is direct: the

space of spherical harmonics of degree k is spanned by the two functions $x \mapsto \cos(k \arg x)$ and $x \mapsto \sin(k \arg x)$ for any x on the circle $\mathbb{S}^1$.

In this setting, the spectral bias of DNNs can be formalized as the fact that the eigenvalues λ_k decrease for large degree k, which follows directly from the fact that the NTK is continuous. But our goal is to have a more precise description of how fast the eigenvalues decay (the 'strength' of the spectral bias) which has an impact on the convergence speed and generalization of DNNs, as discussed in the previous sections.

We will now study how hyper-parameters such as the non-linearity σ, the bias strength β and the depth of the network affect the strength of the spectral bias. There are multiple settings where one might want to tune the strength of the spectral bias: in the classical regression setting, to choose the right tradeoff between convergence speed and generalization (as discussed at the end of the last section), but also in settings such as the training of Generative Adversarial Networks (in Section 1.7) or in topology optimization with DNNs (in Section 1.7).

This section follows the two papers [104, 52] whose original text can be found in Sections 5 and 6.

Order and Chaos for Large Depths

When the depth of the network L becomes very large, two phases have been observed [173], determined by properties of the non-linearity σ and the amount of bias[6].

Since we are studying DNNs with increasing depth L, we need to choose our non-linearity carefully to ensure that the activations of the output layer $\tilde{\alpha}^{(L)}(x)$ remain of order 1. We restrict ourselves to inputs on the $\sqrt{d_{in}}$-hypersphere $\sqrt{d_{in}}\mathbb{S}^{d_{in}-1} = \left\{x \in \mathbb{R}^{d_{in}} : \|x\| = \sqrt{d_{in}}\right\}$, and restrict ourselves to nonlinearities σ of the form $\sigma(x) = \alpha\sigma_0(x)$ for some $\alpha > 0$ and a *standardized* nonlinearity σ_0, i.e. σ_0 satisfies $\mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\sigma_0(z)^2\right] = 1$ (taking the mean over a standard Gaussian variable z). For a given bias strength $0 \leq \beta \leq 1$ we need to choose $\alpha = \sqrt{1-\beta^2}$ so that $\Sigma_\infty^{(\ell)}(x,x) = 1$ for all layers ℓ and any $x \in \sqrt{d_{in}}\mathbb{S}^{d_{in}-1}$ which ensures a constant infinite-width variance of the pre-activations $\tilde{\alpha}^{(\ell)}(x)$ for all layers ℓ.

In this setting an ordered and a chaotic phase appear, determined by the characteristic value $r_{\sigma_0,\beta} = \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\dot{\sigma}(x)^2\right] = (1-\beta^2)\mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\dot{\sigma}_0(x)^2\right]$:

- **Order:** If $r_{\sigma,\beta} < 1$, the NNGP kernel $\Sigma_\infty^{(L)}$ converges to a constant kernel as $L \to \infty$, i.e. for all $x, y \in \sqrt{d_{in}}\mathbb{S}^{d_{in}-1}$

$$\lim_{L\to\infty} \Sigma_\infty^{(L)}(x,y) = 1,$$

  and the limiting rescaled NTK $\hat{\Theta}_\infty^{(L)}(x,y) = \frac{\hat{\Theta}_\infty^{(L)}(x,y)}{\sqrt{\hat{\Theta}_\infty^{(L)}(x,x)\hat{\Theta}_\infty^{(L)}(y,y)}}$ also converges to a constant kernel

$$\lim_{L\to\infty} \hat{\Theta}_\infty^{(L)}(x,y) = 1.$$

[6] In previous works studying this phenomenon, the effect of the variance of the parameters at initialization is also taken into account, which we do not. Note that scaling the non-linearity σ has a similar effect to scaling the initialization of the connection weights by the same factor.

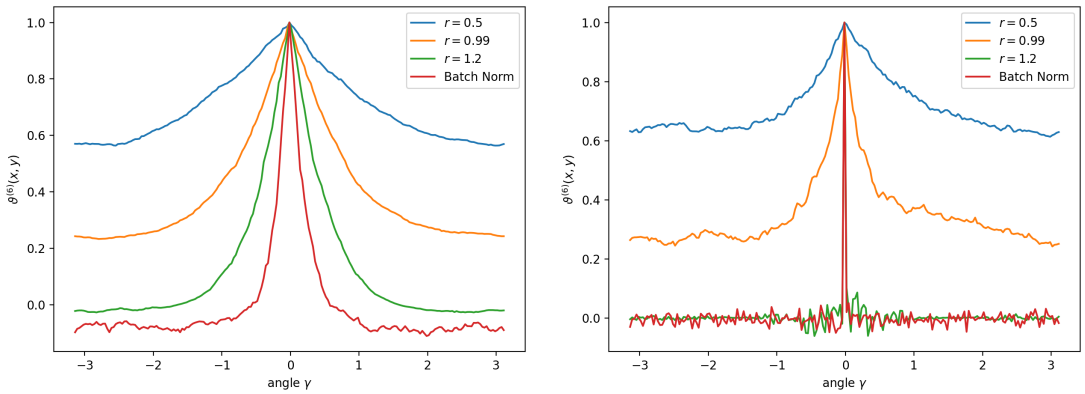


Figure 1.7.1: The NTK on the unit circle for four architectures with depth $L = 5$ (left) and $L = 25$ (right): vanilla ReLU network with $\beta = 1.0$ (blue) and $\beta = 0.1$ (orange), with a normalized ReLU / Layer norm (green) and with Batch Norm (red). Both networks have width 3000, but the deeper network is further from convergence, leading to more noise.

- **Chaos:** If $r_{\sigma,\beta} > 1$ the NNGP kernel $\Sigma_\infty^{(L)}$ converges to the sum of a constant kernel and a Kronecker delta kernel, i.e. there is a value $0 \leq h < 1$ such that for all x, y (with $x \neq -y$)

$$\lim_{L\to\infty} \Sigma_\infty^{(L)}(x,y) = \begin{cases} 1 & \text{if } x = y \\ h & \text{if } x \neq y \end{cases},$$

  and the rescaled NTK $\hat{\Theta}_\infty^{(L)}(x,y)$ converges to a Kronecker delta kernel:

$$\lim_{L\to\infty} \hat{\Theta}_\infty^{(L)}(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}.$$

This phase transition was first observed before the introduction of the NTK, and only the effect on the NNGP kernel $\Sigma_\infty^{(\ell)}$ were proven at the time [173, 42, 231]. In 2019 we studied the effect of the order/chaos transition on the NTK kernel and other related questions (see Section 5) in a paper with Franck Gabriel, François Ged and Clément Hongler. A few similar analysis of the order/chaos transition for the NTK came out independently [86, 225, 96].

Clearly, these two regimes have a significant effect on the decay of the eigenvalues of the limiting NTK and consequently on the strength of the spectral bias of DNNs:

- In the chaotic regime, the condition number κ of the limiting NTK Gram matrix $\Theta_\infty^{(L)}(X, X)$ converges to 1 as the depth grows to infinity $L \to \infty$, in other words the spectral bias vanishes. In this regime, DNNs with very large depths L can be trained very efficiently but cannot generalize.

- In contrast, in the ordered regime, the condition number κ grows to infinity as $L \to \infty$. In other terms the spectral bias becomes stronger with the depth. The training time blows up, but generalization is still possible (though it is of course not guaranteed).

This suggests that for very large depths, it is crucial to choose a non-linearity σ and bias strength β with a characteristic value $r_{\sigma,\beta}$ of 1, as this is the only regime where a reasonable balance between convergence speed and generalization can be achieved.

However, for networks that are not as deep, a wider range of characteristic values around 1 is viable. In such settings, the characteristic value can be thought of as an indicator of the strength of the spectral bias (a large characteristic value indicating a weak spectral bias and vice-versa), and one can tune the strength of the spectral bias through the characteristic value $r_{\sigma,\beta}$.

The spectral bias can always be strengthened by increasing the bias strength β (hence reducing the characteristic value $r_{\sigma,\beta}$) but for some non-linearities even removing completely the bias ($\beta = 0$) still leads to a characteristic value $r_{\sigma,\beta}$ smaller or equal to 1 – for example for the standardized ReLU $\sigma_0(x) = \frac{1}{\sqrt{2}} \max\{0, x\}$ the characteristic value without bias is $r_{\sigma_0,\beta=0} = 1$. It might therefore be useful to find techniques to increase the characteristic value (and reduce the spectral bias).

The Chaotic Effect of Layer Normalization

It seems that various normalization techniques have such a chaotic effect, i.e. they increase the characteristic value $r_{\sigma_0,\beta}$. Let us first consider layer normalization in which the definition of the activation $\alpha_k^{(\ell)}(x)$ of the k-th neuron in the ℓ-th layer is changed to

$$\alpha_k^{(\ell)}(x) = \frac{\sigma\left(\tilde{\alpha}^{(\ell)}(x)\right) - m^{(\ell)}(x)}{d^{(\ell)}(x)}$$

for the mean $m^{(\ell)}(x) = \frac{1}{n_\ell} \sum_{k=1}^{n_\ell} \sigma(\tilde{\alpha}_k^{(\ell)}(x))$ and standard deviation

$$d^{(\ell)}(x) = \sqrt{\frac{1}{n_\ell} \sum_{k=1}^{n_\ell} \left(\sigma(\tilde{\alpha}_k^{(\ell)}(x)) - m^{(\ell)}(x)\right)^2}$$

of the activations $\sigma\left(\tilde{\alpha}^{(\ell)}(x)\right)$ of the layer.

We show that in the infinite-width limit, layer normalization is equivalent to non-linearity normalization, in which the non-linearity σ_0 is translated and scaled to have zero mean and unit variance when evaluated on a standard Gaussian variable z, i.e. σ_0 is replaced by the normalized non-linearity $\bar{\sigma}_0$ defined as

$$\bar{\sigma}_0(x) = \frac{\sigma_0(x) - \mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\sigma_0(z)\right]}{\sqrt{\mathrm{Var}_{z\sim\mathcal{N}(0,1)}(\sigma_0(z))}}.$$

We then show that the normalized non-linearity $\bar{\sigma}_0$ has always a larger characteristic value than the original one $r_{\bar{\sigma}_0,\beta} \geq r_{\sigma_0,\beta}$ and that in the absence of bias ($\beta = 0$) a normalized non-linearity always lies in the chaotic regime $r_{\bar{\sigma}_0,\beta=0} > 1$, as long as the non-linearity σ_0 is not linear. This reveals the chaotic effect of layer normalization.

In practice, it is much more common to use batch normalization instead of layer normalization. While batch normalization is much more difficult to study theoretically, we show that if one applies batch normalization after the last non-linearity, it has the effect of upper bounding the eigenvalue corresponding to the 0-th order spherical harmonic (the constant function) which typically dominates. This suggests that batch normalization has a similar chaotic effect (as supported by Figure 1.7.1), reducing the spectral bias of DNNs. For more details, see Section 5.

Generative Adversarial Networks

Generative Adversarial Networks (GANs) [78] are generative models: given a set X of datapoints (for example images of human faces or handwritten digits) GANs are trained to learn to sample new datapoints x which resemble the training dataset X. GANs are made up of two DNNs competing against each other:

- The Generator $G_\theta : \mathbb{R}^k \to \mathbb{R}^{d_{in}}$ with parameters θ: the generated datapoints x are sampled as $x = G_\theta(z)$ where z is a random k-dim vector with i.i.d. standard Gaussian entries $\mathcal{N}(0, 1)$. By tuning the parameters θ, the distribution of x can be learned.
- The Discriminator $D_\phi : \mathbb{R}^{d_{in}} \to \mathbb{R}$ with parameters ϕ: during training, the discriminator receives a batch of real datapoints X and of generated data $\tilde{X} = G_\theta(Z)$ and learns to classify real from generated data. Simultaneously, the generator learns to 'fool' the discriminator, to make the generated data undistinguishable from the real data.

The two player game aspect of GANs makes training very unstable and GANs are notoriously hard to train [27, 174, 191]. A technique that helps a lot for training GANs is batch normalization [223]. While batch normalization is known to speed up training in more traditional DNN training, in GANs the absence of batch normalization often makes training almost impossible. In [104] (see Section 5) we propose an explanation for the importance of normalization in GANs: normalization moves the network outside of the ordered regime, which is characterized by problems of so-called mode collapse and checkerboard artifacts.

Mode Collapse

The most common failure state in GANs is the so-called *mode collapse*, where the generator becomes constant or almost constant, hence collapsing the generated distribution to a Dirac mass or a very concentrated distribution. Once such a state is reached, it is difficult to 'uncollapse' the generator. Therefore, we want to understand why the generator has a natural tendency to converge to a constant function and how to avoid this issue.

There is an obvious link between the ordered regime and mode collapse: in the ordered regime, the NNGP kernel and the NTK of a very deep and wide generator G_θ are almost constant, which implies that the generator will be almost constant at initialization and move along constant directions during training. More precisely, the NTK matrix will have d_{in} dominating eigenvalues whose eigenvectors roughly span the d_{in} space of constant functions from $\mathbb{R}^k$ to $\mathbb{R}^{d_{in}}$.

This suggests the following explanation for the importance of normalization in the generator: the typical choice of non-linearity (the ReLU) lies in the ordered regime when $\beta > 0$, and even in the absence of bias $\beta = 0$ we observe a dominating constant mode [104] (see Section 5). We therefore need normalization to move to the chaotic regime to weaken the bias towards constant functions and avoid mode collapse.

Checkerboard Artifacts

GANs are typically trained to generate images, in which case the generator is a so-called *deconvolutional network*, in which the neurons of each layer have a spatial structure, i.e. they are indexed by an x coordinate, a y coordinate, and a channel k. The neurons of two consecutive layer are connected in a convolutional manner, according to their coordinates, and with a stride s which

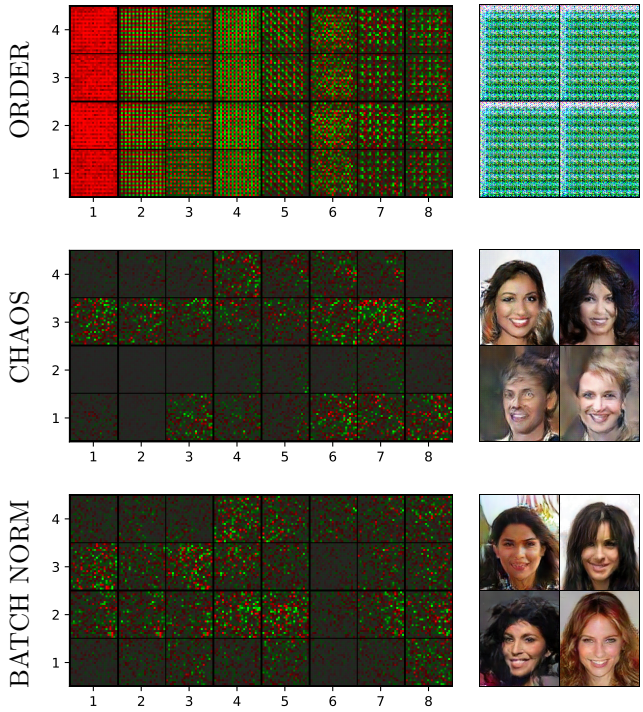


Figure 1.7.2: The left column represents the first 8 eigenvectors of the NTK Gram matrix of a DC-NN (L=3) on 4 inputs (as well as some other architecture changes, see Section 5 for more details). The right column represents the results of a GAN on CelebA. Each line correspond to a choice of nonlinearity/normalization for the generator: (top) ReLU, (middle) normalized ReLU and (bottom) ReLU with Batch Normalization.

allows to multiply by s the height and width of the spatial field at every layer to generate large images. In this setting the infinite-width limit corresponds to letting the number of channels grow to infinity.

When GANs are trained with such a generator, it is common to observe checkerboard artifacts, which are small patterns in the image which repeat every s pixels (or every s^m pixels for some integer m). These checkerboard patterns are especially visible in the image generated after mode collapse (see Figure 1.7.2 top right).

In [104] (Section 5), we study the large depth behaviour of the NTK of infinitely wide deconvolutional network. We observe the same order/chaos transition: while the chaotic regime remains mostly the same, in the ordered regime we observe a natural tendency for checkerboard artifacts. More precisely, we observe that in the infinite depth and width limit, the dominating eigenfunctions $f^{(k)} : \mathbb{R}^k \to \mathbb{R}^{d_{in}}$ of the NTK are constant in their inputs (as for the fully-connected case), but instead of all constant eigenfunctions having the same eigenvalue, they are ordered in a specific manner.

These constant functions $f^{(k)}(x) = y$ are determined by the image $y \in \mathbb{R}^{w \times h}$ that they generate (for grayscale images $d_{in} = wh$ where w and h are the width and height of the image). The

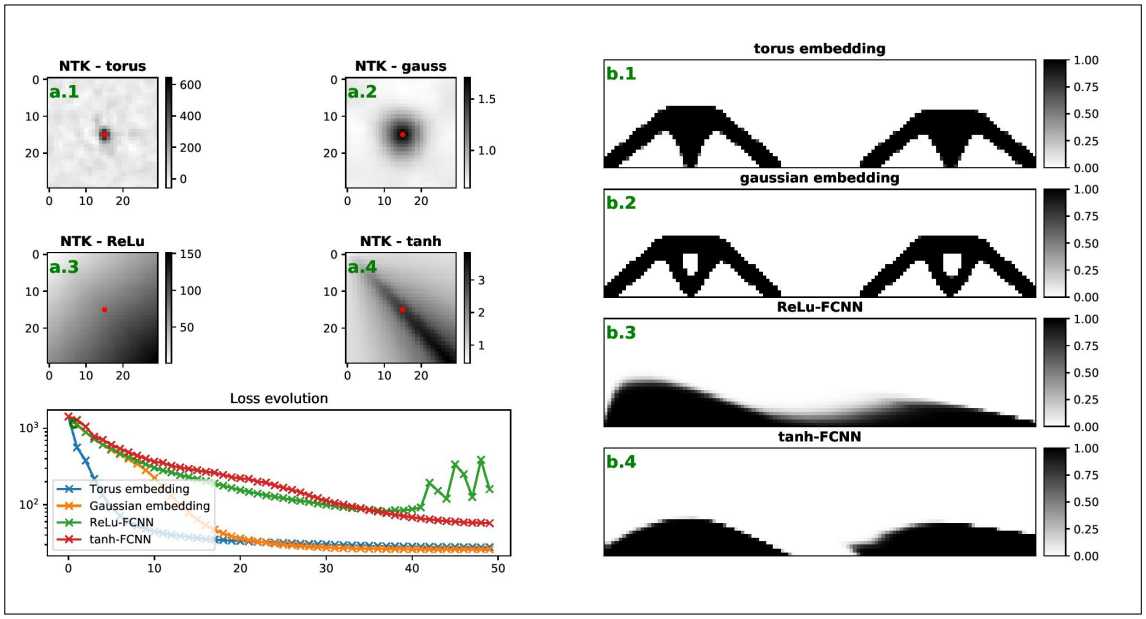


Figure 1.7.3: Left: empirical NTK of FCNNs with both embedding (a.1, a.2, see Section 6.4 for details) or without embedding (a.3 with ReLu, a.4 with tanh). Right: Corresponding shape obtained after training. Note that methods without spatial invariance particularly struggles with this symmetric load case (b.3, b.4) while both "embedded methods" respect the symmetry (b.1, b.2). We also observed that training with non-embedded methods is very unstable

dominating eigenfunction generates a constant image (which can be understood as a checkerboard pattern that repeat every $s^0 = 1$ pixels), followed by eigenfunctions whose output image y has a checkerboard pattern repeating every s pixels, followed by checkerboard patterns that repeat every s^2 pixels and so on and so forth.

This order of the eigenfunctions is visible in the top line of Figure 1.7.2, where the first 8 eigenfunctions on 4 inputs are plotted with a stride of $s = 2$, the largest eigenfunction is constant, the following 3 feature checkerboard patterns that repeat every 2 pixels and the following 4 feature checkerboard patterns that repeat every 4 pixels.

Here again these checkerboard artifacts can be avoided in the chaotic regime, hence further supporting the importance of normalization.

DNN-based Topology Optimization

Topology optimization tackles tasks such as finding the optimal shape of a bridge to be as sturdy as possible with the minimal amount of material. The underlying principle is that given a 2D or 3D object shape (represented by a 'shape image', a 2D or 3D array that indicate the presence of material at every pixel) and a set of forces acting on the object, the stability of the object under these forces can be computed. Since this computation is differentiable, it is possible to maximize the stability of the shape with gradient ascent on the 2D or 3D image. This is called the Solid Isotropic Material Penalisation (SIMP) method [21, 149].

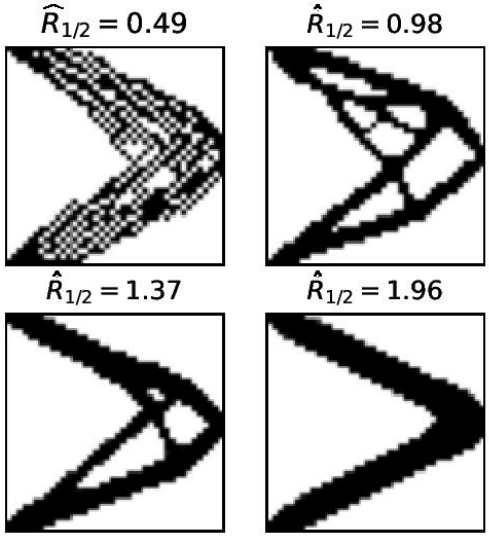


Figure 1.7.4: Shape obtained for different values of $\widehat{R}_{1/2}$ with a Gaussian embedding for different values of $\ell \in \{0.5, 1, 1.4, 2\}$.

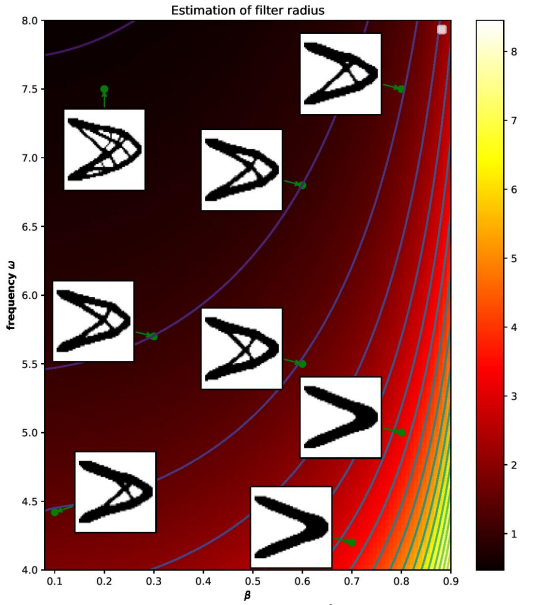


Figure 1.7.5: Colormap of $\widehat{R}_{1/2}$ in the (β, ω) plane, torus embedding. Level lines and shapes obtained for different radius are represented.

However, this optimization commonly leads to checkerboard artifacts in the optimized shape, where the program appears to consider pixels connected diagonally to be stable. A common solution to this problem is to apply a low-pass filter to the shape image before computing stability, but this technique has the drawback of leading to blurry shapes.

It was later observed that one can avoid both checkerboard artifacts and blurriness by using a DNN to represent the shape [207, 31]. The DNN takes as inputs the 2D or 3D coordinates of the pixel and outputs a value between 0 and 1 (by applying the sigmoid function to the outputs of the network) representing the presence of material.

NTK Analysis

If the network used to represent the shape is infinitely wide, the reason why DNNs avoid checkerboard patterns is quite simple: the NTK implicitly plays the role of a filter. In addition to this, the application of the sigmoid function to the outputs of the network avoids the blurriness.

However this analysis also reveals a problem: the NTK is invariant under rotations but not translations, as can be seen in Figure 1.7.3 (a.3 and a.4). This breaks the symmetry of topology optimization, where translating the force constraints should lead to a translation of the resulting optimal shape, but the lack of translation invariance of the NTK leads to this property not being respected as can be seen in Figure 1.7.3 (b.3 and b.4). We propose two solutions that lead to an (approximately) translation- and rotation-invariant NTK.

The idea behind both techniques is to first map the input coordinates (x, y, z) to a larger space $\mathbb{R}^{d_{in}}$ with a map $\phi : \mathbb{R}^3 \to \mathbb{R}^{d_{in}}$ such that the scalar product $\phi(x, y, z)^T \phi(x', y', z')$ only depends on

the Euclidean distance between (x, y, z) and (x', y', z'). Since the limiting NTK $\Theta_\infty^{(L)}(u, v)$ depends only on the norm of u and v and their scalar product $u^T v$, we have that $\Theta_\infty^{(L)}(\phi(x, y, z), \phi(x', y', z'))$ is translation and rotation invariant (since it only depends on the Euclidean distance between (x, y, z) and (x', y', z')).

We however show that any non-constant function ϕ which satisfies this property must have an infinite-dimensional image ($d_{in} = \infty$). We instead propose two maps with finite-dimensional image that approximately satisfy this property:

- **Hypertorus embedding:** The first map sends the coordinate x, y, z to pairs $(\sin(\delta x), \cos(\delta x))$, $(\sin(\delta y), \cos(\delta y))$ and respectively $(\sin(\delta z), \cos(\delta z))$ for some $\delta > 0$. With this map the NTK becomes translation-invariant but loses its rotation-invariance, however we argue that translation-invariance plays a more important role than rotation-invariance (see Section 6).
- **Gaussian embedding:** The second map uses so-called random Fourier features, the coordinates (x, y, z) are mapped to the d_{in}-dimensional vector $\phi(x, y, z)$ with entries

$$\phi_k(x, y, z) = \frac{1}{\sqrt{d_{in}}} \sin\left(\frac{a_k x + b_k y + c_k z}{\ell} + \frac{\pi}{4}\right)$$

  for some i.i.d. standard Gaussian scalars a_k, b_k, c_k and the so-called lengthscale $\ell > 0$. In the limit $d_{in} \to \infty$ the NTK becomes translation- and rotation-invariant and for finite d_{in} it is only approximately so.

With these embeddings, the NTK $\Theta_\infty^{(L)}(\phi(x, y, z), \phi(x', y', z'))$ only depends on the Euclidean distance between the coordinates (up to a small error), much like a traditional low-pass filter. We can therefore define the filter radius $\hat{R}_{1/2}$ of the NTK as the smallest distance d such that the value of the NTK at two coordinates at a distance d from one other is half the value the NTK at its center (evaluated at the same coordinate twice). This filter radius is directly related to the strength of the spectral bias of the NTK, and it has a strong impact on the convergence speed of topology optimization and on the final optimized shape. The filter radius can be tuned using the scalar δ for the hypertorus embedding and the lengthscale ℓ for the Gaussian embedding to obtain a range of shapes with different levels of detail, as shown in Figures 1.7.4 and 1.7.5.

Remark 1.3. There are many other settings where DNNs are used to represent images and the same problem of translation invariance and tuning of the spectral bias appear [155, 214]. The analysis that we did for topology optimization, as well as the solutions we proposed would also apply to these other settings.

1.8 Finite-width Analysis

The results presented up to this point have mostly focused on infinitely wide DNNs, giving a thorough description of their convergence and generalization properties. It is now natural to ask how close finite-width networks are to their infinite-width counterparts?

Double-Descent Curve

When one plots the train and test error as a function of the width of the network w (or as a function of the number of parameters P which scales with the width), a surprising phenomenon is observed [18, 69] (see Figure 1.8.1 for an example with MNIST):

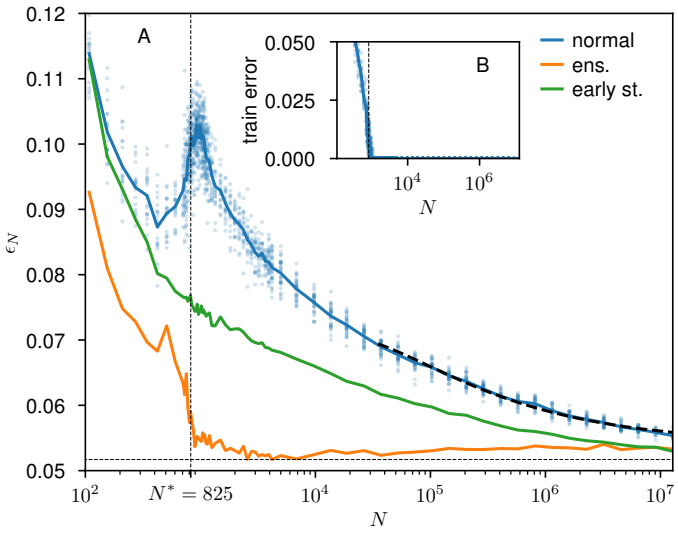


Figure 1.8.1: (A) Empirical test error *v.s.* number of parameters: average curve (blue, averaged over 20 runs); early stopping (green); ensemble average $\bar{f}_N^n$ (orange) over $n = 20$ independent runs. In all the simulations we used fully-connected networks with depth $L = 5$ and input dimension $d = 10$, trained for $t = 2 \cdot 10^6$ epochs to classify $P = 10k$ MNIST images depending on their parity, using their first 10 PCA components, and the test set includes $50K$ images (the plots are taken from the original paper where the number of parameters is denoted by N and the number of datapoints by P). The vertical dashed line corresponds to the interpolation threshold: at that point the test error peaks. Ensemble averaging leads to an essentially constant behavior when N becomes larger than N^*.

- When it comes to the training loss, a clear transition can be observed: from an *under-parameterized regime* for small widths, where the number of parameters is too small to fit the data and to achieve a zero training loss, to an *over-parameterized regime* for large widths, where the training loss is zero. We call the smallest number of parameters where gradient descent reaches a zero training loss the *interpolation threshold*. For regression tasks (e.g. the MSE loss) the interpolation threshold happens when the number of parameters P equals the number of datapoints N, however for classification tasks (e.g. the cross-entropy loss or hinge loss), this transition can happen significantly earlier [69].

- The test error on the other hand has a surprisingly non-monotonous behavior. From the traditional point of view of the bias-variance tradeoff, one would expect the test error curve to have a U-shape with the optimal width striking a balance between the bias and variance terms. We do observe such a U-shape, but only in the under-parameterized regime. At the interpolation threshold, the test error explodes and then starts to go down again as we move further in the over-parameterized regime. As the width goes to infinity, the test loss converges to some finite value which appears in some cases to be optimal (as in Figure 1.8.1).

- The explosion of the test error can be avoided with early stopping: at the ideal stopping

time (stopping at the time t where the test error is at its lowest) the test error decreases monotonically with the width w.

The NTK analysis explains why the test error converges to a finite value as $w \to \infty$, but it does not explain why in the over-parameterized regime the test error decreases as the width increases nor the explosion of the test loss at the interpolation threshold.

Effect of Ensembling

A similar double-descent curve has been observed in many different models such as random forests and random features [18]. Interestingly, all of these models are random models, in the sense that given a fixed training set $\{(x_i, y_i), i = 1, \dots, N\}$ the final estimator (the final network function $f_{\theta(T)}$ for DNNs) is random.

This observation suggests that there might be a relation between the randomness of the network function f_θ w.r.t. the sampling of the parameters and the double descent curve. We studied this relationship in the paper [67] presented in Section 7.

We used ensembling to average out the randomness due to the sampling of the parameters: ensembling consists in training K network with i.i.d. initializations $\theta^1(0), \dots, \theta^K(0)$ on the same dataset and averaging their outputs $f_t^{ens}(x) = \frac{1}{K} \sum_{k=1}^{K} f_{\theta^k(t)}(x)$. A K-fold ensembling allows one to reduce the variance of the network function by K.

As shown in Figure 1.8.1, we observed that the double descent curve disappears after ensembling. This implies that the explosion of the test loss at the interpolation threshold is due to an explosion in the variance of the network function. Furthermore, we observe that right after the interpolation threshold it is possible to attain the same test error as an infinitely-wide network after ensembling.

NTK Regime

Our strategy to explain the above observations mathematically is to extend the NTK analysis of infinite-width networks to finite-width ones. As mentioned in Section 1.5, the standard deviation of the NTK at initialization is of order $w^{-\frac{1}{2}}$ while the rate of change of the NTK during training is of order w^{-1}. This suggests that for large but finite-width there is a regime where the change of the NTK in time is negligible but its randomness at initialization is not, which we call the NTK regime (also called the lazy regime [34] or kernel regime).

Understanding the extent of this NTK regime is still an area of active research. At the moment however, we will study the double descent under the assumption that the network is in the NTK regime, i.e. we will assume that the NTK is random but fixed in time. The double descent phenomenon is well suited to this type of analysis since our numerical experiments suggest that it is related to the randomness of the network function, suggesting that the randomness of the NTK is more important than its time evolution.

Remark 1.4. There are regimes outside the NTK regime where the NTK is not constant, as described in Section 1.9, resulting in a different behavior to the one described in this section. For example, the mathematical analysis in this section implies that the test loss is always optimal in the infinite-width limit (which matches our experiments on MNIST in Figure 1.8.1 as well as those of others [18, 69] also on MNIST), however it has been observed empirically on CIFAR-10 that there are finite-width networks that have a smaller test loss than that of an infinite-width network [6]. We cannot fully explain this difference yet, but it seems that the choice of dataset plays a key role.

Random Feature Models

As mentioned above, the double-descent curve can also be observed in Random Features (RF) models [176]. In the most general interpretation, a RF model is a linear model $F(\theta) = \frac{1}{\sqrt{P}} \sum_{p=1}^{P} \theta_p f_p$ whose features $f_p : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ are random functions. With this interpretation, the tangent linear model $T_{\theta(0)} F^{(L)}(\theta)$ (see Section 1.4) of a DNN at initialization is a RF model, with features $f_p(x) = \partial_{\theta_p} f_{\theta(0)}(x)$.

We will now describe the double-descent curve theoretically for Gaussian RF models, where the random features $f_p : \mathbb{R}^{d_{in}} \to \mathbb{R}$ (for simplicity, we assume $d_{out} = 1$) are sampled as i.i.d. Gaussian processes with a fixed covariance kernel $K : \mathbb{R}^{d_{in}} \times \mathbb{R}^{d_{in}} \to \mathbb{R}$. As we will see, this model has the advantage of simplifying the theoretical analysis while keeping all of the interesting features of the double descent curve. These results were published in a paper [102] reproduced in full in Section 8.

Training a RF model corresponds to doing kernel gradient descent with a random kernel $\tilde{K}(x, y) = \frac{1}{P} \sum_{p=1}^{P} f_p(x) f_p(y)$. On the MSE cost with inputs X and outputs Y, the network function at a time t can be approximated by the kernel ridge regression estimator

$$\hat{f}^{RF}_{\lambda,P}(x) = \frac{1}{N} \tilde{K}(x, X) \left(\frac{1}{N} \tilde{K}(X, X) + \lambda I_N \right)^{-1} Y$$

for $\lambda = \frac{1}{2t}$ (this follows from the same argument as in Section 1.5).

Since as the number of features P grows to infinity the random kernel $\tilde{K}$ converges to the fixed kernel K, we know that the RF predictor $\hat{f}^{RF}_{\lambda,P}(x)$ concentrates as $P \to \infty$ around the kernel predictor

$$\hat{f}^{K}_{\lambda}(x) = \frac{1}{N} K(x, X) \left(\frac{1}{N} K(X, X) + \lambda I_N \right)^{-1} Y.$$

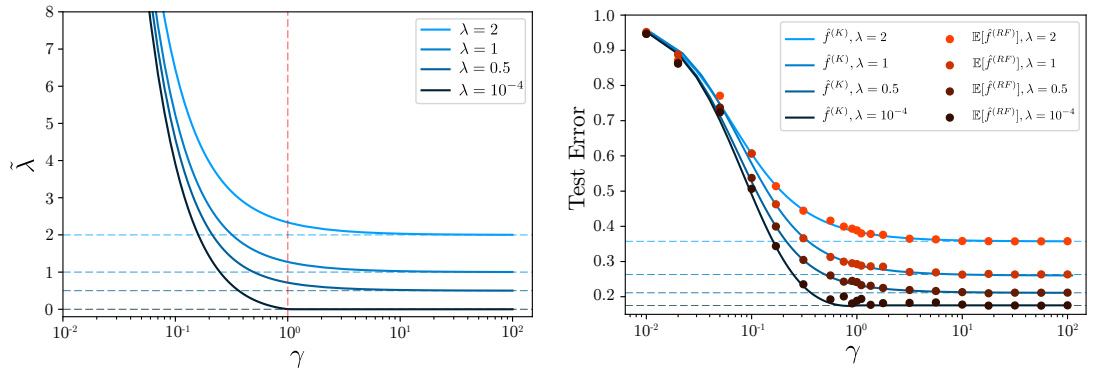
Our goal is to study the distribution of the RF predictor $\hat{f}^{K}_{\lambda,P}$ for finite but large P, in particular its expectation and variance, to describe the expected test error.

Remark 1.5. For DNNs in the NTK regime, the random kernel $\tilde{K}$ is the finite-width NTK $\Theta^{(L)}$ while the kernel K is the limiting NTK $\Theta^{(L)}_{\infty}$. While in both cases we have a random kernel approximating a deterministic kernel, there is an important distinction. In contrast to the features of a Gaussian RF model which are all independent, the features of the NTK $\partial_{\theta_p} f_{\theta(0)}$ are not independent. This leads to a different rate of convergence of the random kernel to its deterministic limit for deep networks. While for Gaussian RF the error $\tilde{K} - K$ is of order $O(P^{-\frac{1}{2}})$ for deep networks ($L > 2$) the error $\Theta^{(L)} - \Theta^{(L)}_{\infty}$ is of order $P^{-\frac{1}{4}}$ (since the error $\Theta^{(L)} - \Theta^{(L)}_{\infty}$ is $w^{-\frac{1}{2}}$ and P is of order w^2 when $L > 2$). This has an effect on the rate at which the test error of finite-width networks converges to the test error of infinite-width networks, as discussed in [67] (see Section 7).

Implicit Regularization of Random Feature Models

The analysis of the mean RF predictor reveals an implicit regularization, in the sense that the mean RF predictor $\mathbb{E}\left[\hat{f}^{RF}_{\lambda,P}\right]$ with ridge λ is close[7] to the kernel predictor $\hat{f}^{K}_{\tilde{\lambda}}$ with an *effective ridge* $\tilde{\lambda}$

[7]Explicit bounds on the distance between the two can be found in Section 8.4. Though these theoretical bounds blow up as $\lambda \searrow 0$ we observe empirically that this approximation remains accurate even for very small or zero ridges λ.



(a) Evolution of $\tilde{\lambda}$

(b) Average λ-RF predictor vs. $\tilde{\lambda}$-KRR

Figure 1.8.2: *Comparison of the test errors of the average λ-RF predictor and the $\tilde{\lambda}$-KRR predictor.* We train the RF predictors on $N = 100$ MNIST data points where K is the RBF kernel, i.e. $K(x, x') = \exp\left(-\|x - x'\|^2/\ell\right)$. We approximate the average λ-RF on 100 random test points for various ridges λ. In **(a)**, given γ and λ, the effective ridge $\tilde{\lambda}$ is computed numerically using (8.4.2). In **(b)**, the test errors of the $\tilde{\lambda}$-KRR predictor (blue lines) and the empirical average of the λ-RF predictor (red dots) agree perfectly.

which is the unique positive solution of the equation

$$\tilde{\lambda} = \lambda + \frac{\tilde{\lambda}}{P}\mathrm{Tr}\left[\frac{1}{N}K(X,X)\left(\frac{1}{N}K(X,X) + \tilde{\lambda}I_N\right)^{-1}\right].$$

The fact that the effective ridge is larger than the original ridge $\tilde{\lambda} \geq \lambda$ implies that the use of random features has an implicit regularization effect, increasing the ridge parameter.

Moreover in the over-parametrized $P \geq N$ and ridgeless $\lambda \searrow 0$ setting, one can show with a simple argument that $\mathbb{E}\left[\hat{f}^{RF}_{\lambda\searrow 0,P}\right] = \hat{f}^{K}_{\lambda\searrow 0}$. This is in line with the empirical observation made in [67] that in the overparametrized regime, the test loss at the end of training is almost constant after ensembling.

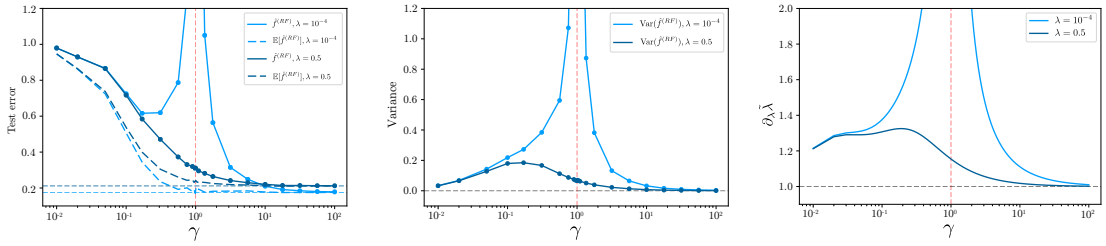
Remark 1.6. There is a direct correspondence between the effective ridge $\tilde{\lambda}$ and the Signal Capture Threshold ϑ from Section 1.6. Both theoretical analyses rely on the same tools from Random Matrix Theory.

Variance Explosion

For non-zero ridge $\lambda > 0$, we can obtain some bounds over the variance of the predictor directly:

$$\mathrm{Var}\left(\hat{f}^{RF}_{\lambda,P}(x)\right) \leq \frac{c\,\|Y\|^2}{PN^2\lambda^2},$$

where the constant c depends only on the kernel K the input data X and the point x (see Appendix A.1 for a derivation of the bound and an explicit formula for c).



(a) Ridgeless vs. Ridge (b) Variance of λ-RF (c) Evolution of $\partial_\lambda \tilde{\lambda}$

Figure 1.8.3: *Average test error of the ridgeless vs. ridge λ-RF predictors.* In **(a)**, the average test errors of the ridgeless and the ridge RF predictors (solid lines) and the effect of ensembling (dashed lines) for $N = 100$ MNIST data points. In **(b)**, the variance of the RF predictors and in **(c)**, the evolution of $\partial_\lambda \tilde{\lambda}$ in the ridgeless and ridge cases. The experimental setup is the same as in Figure 1.8.2.

If we fix the number of datapoints N, this bound illustrates how increasing the ridge helps avoid an explosion of variance. This is also in agreement with the observation that early stopping avoids the explosion of variance, since early stopping at a time t is similar to taking a ridge of $\lambda = 1/2t$.

If we instead fix a positive ridge $\lambda > 0$ and increase P, we see that the variance goes to zero at a rate of P^{-1}. However as noted in Remark 1.5, for DNNs the non-independence leads to a variance of order $P^{-\frac{1}{2}}$ instead, as observed empirically in [67] (see Section 7). Note also that for DNNs there is an extra source of variance, the randomness of the network function at initialization, and this variance does not vanish in the infinite-width limit, though it seems to be very small in Figure 1.8.1, where the difference of the test before and after ensembling seems to almost vanish as the width grows.

Our goal was to describe the explosion of variance at the interpolation threshold, but due to a number of technical issues, we were not able to precisely describe the covariance of the RF predictor, especially not in the ridgeless case $\lambda \searrow 0$. Nevertheless, our theoretical analysis suggests that the variance of the RF predictor scales with the derivative of the effective ridge[8] $\partial_\lambda \tilde{\lambda}$. This derivative explodes when $P = N$ and $\lambda \searrow 0$, which is exactly the location of the explosion of variance in the numerical experiments. A more detailed discussion of the variance of the RF predictor can be found in Section 8.5.

These theoretical results suggest that at least in the NTK regime, finite-width networks are very similar to their infinite-width counterparts in expectation (up to a slight increase in the ridge from λ to $\tilde{\lambda}(\lambda)$, which would correspond to a slight change of training time from $1/2\lambda$ to $1/2\tilde{\lambda}(\lambda)$). The change in test error between finite and infinite-width networks is mostly due to the variance, which has a complex behavior: it explodes at the ridgeless interpolation threshold ($N = P$ and $\lambda \searrow 0$), but this explosion of variance can be avoided with either ensembling, early stopping, or by increasing the width.

[8] This is reminiscent of the role the derivative of the SCT $\partial_\lambda \vartheta$ played in the description of the variance of the KRR predictor in Section 1.6.

1.9 Regimes of Training

The results presented up to this point have all been in the so-called NTK regime (see Section 1.8) where the rate of change of the NTK is negligible (so that we can assume it to be constant). But there exists another regime where the rate of change of the NTK has a significant impact on the training dynamics and the network function that is learned. While this regime is commonly called the active regime (also feature-learning regime or rich regime) there are actually multiple active regimes, each corresponding to different ways to leave the NTK regime with distinct training dynamics.

Let us remind that the proof of convergence of the NTK to a constant kernel relies on the fact that the parameters of the network move very little for large widths, which implies that there is a global minimum close to the parameters at initialization $\theta(0)$ that gradient flow converges to (see Theorem 1.2). There are a number of settings where the rate of change of the parameters is not small:

1. With a cost such as cross-entropy, which decays towards infinity, there is no finite global minimum. As a result, training never stops and the length of the training path is infinite and the NTK remains approximately constant only up to a time T (which increases with the width w of the network) after that, the change in time of the NTK is significant. For more details, see [36, 80, 81, 156, 209, 235].

2. Increasing the number of training points N at the same time as the width w can lead to an active regime since the parameters need to move more to fit the data. This setting is less studied, but it could explain why the test loss curve can be non-monotonically decreasing in the over-parameterized regime on CIFAR-10 [6] as explained in Section 1.8.

3. Adding L_2 regularization to the loss of DNNs $\mathcal{L}_\lambda(\theta) = C(f_\theta) + \lambda \|\theta\|^2$ completely changes the critical points of the loss surface. In particular there might not be any closeby global minimum, leading to active dynamics (see [80, 81, 41] for linear networks and [10, 194, 166] for shallow non-linear networks).

4. There is a ball around the origin in parameter space $\mathbb{R}^P$ with no global minimum, since the network must represent a non-zero function to fit the data. If one initializes the parameters with a small variance, then the parameters at initialization will lie with high probability inside this ball, far away from any global minimum, hence leading to another active regime, which we will discuss in the rest of this section: we will see that changing how the variance of the parameters at initialization scales with the width, one can reach three regimes: the NTK regime, a critical regime, and a saddle-to-saddle regime. The critical regime is related to the Mean-Field limit studied in [35, 183] or Maximal Update parametrization from [229].

Active regimes have also been observed for large learning rates [134], or if one lets the width and depth of the network grow at the same time [84].

In general these active regimes are much less understood than the NTK regime, especially in the deep case. While all of these active regimes have very distinct dynamics, they are almost all related to some notion of sparsity. This is especially visible for linear networks, i.e. networks with no non-linearity σ (or equivalently with $\sigma(x) = x$); which can only represent linear functions. It has been observed in different settings that the linear maps learned by linear network feature some form of low-rank bias [5, 74, 81, 135, 209, 235, 195, 196]. It remains a challenge to generalize these

results to the non-linear case, mostly because the underlying proofs often rely on some tricks that only apply to linear networks.

The rest of this section summarizes the results of the paper [107] written with François Ged, Berfin Şimşek, Clément Hongler and Franck Gabriel, which studies the training dynamics of linear networks for different scales of initialization of the parameters. The original paper can be found in Section 9.

Linear Networks

A Deep Linear Network (DLNs) is a model for linear maps from $\mathbb{R}^{d_{in}}$ to $\mathbb{R}^{d_{out}}$. The $d_{in} \times d_{out}$ network matrix A_θ which defines the linear map takes the form of a matrix product of L matrices

$$A_\theta = W_L \cdots W_1$$

where W_ℓ is a $n_\ell \times n_{\ell-1}$ dimensional matrix.

Remark 1.7. Note the absence of the $\frac{1}{\sqrt{n_\ell}}$ factors in the definition of the network matrix. If we were to take the definition of DNNs from Section 1.3 and remove the non-linearity and the bias, we would instead have the definition

$$A_\theta^{NTK} = \frac{1}{\sqrt{n_0 \cdots n_{L-1}}} W_L \cdots W_1.$$

The parametrization from in Section 1.3 (with the $\frac{1}{\sqrt{n_\ell}}$ factors) is called the NTK parameterization; it is best suited when one studies the NTK regime. We call the parameterization presented in this section (without the $\frac{1}{\sqrt{n_\ell}}$ factors) the *standard parameterization*, as it is the most common one.

Note that the two parametrizations are equivalent up to a scaling of the parameters and the learning rate: if $t \mapsto \theta(t)$ is a gradient flow path for the standard parameterization, then $t \mapsto (n_0 \cdots n_{L-1})^{\frac{1}{2L}} \theta((n_0 \cdots n_{L-1})^{-\frac{L+1}{2L}} t)$ is a gradient flow path for the NTK parameterization. Studying one or the other parameterization is therefore purely a matter of convenience, and we obtain nicer scaling factors with the standard parameterization in the active regime. Informally, it seems that the NTK parametrization is best suited to study the NTK regime while the standard parametrization is best suited for active regimes.

Symmetries of Deep Linear Networks

Two types of symmetries of DLNs will play an important role in our analysis:

Rotations: We already mentioned how in DNNs one can permute neurons without changing the outputs of the network. For DLNs this can be extended to any orthogonal transformation of the hidden layers. We define a rotation $R = (O_1, \dots, O_{L-1})$ of a DLN where O_ℓ is a $n_\ell \times n_\ell$ orthogonal transformation. A rotation can be applied to a vector of parameter $\theta = (W_1, \dots, W_L)$, yielding a new set of parameters

$$R\theta = (O_1 W_1, O_2 W_2 O_1^T \dots, W_L O_{L-1}^T).$$

Rotations preserve the network matrix ($A_{R\theta} = A_\theta$ for any parameter θ) and as well as gradient flow (if $\theta(t)$ is a gradient flow path, then so is $R\theta(t)$).

Inclusions: A network of width w can be included into a network into a wider network of width w' by adding zero connections everywhere. More precisely, given parameters $\theta = (W_1, \dots, W_L)$ of a network of width w, the inclusion $I^{(w\to w')}\theta$ of θ into a network of width w' is defined as $I^{(w\to w')} = (W_1, \dots, W_L)$ with

$$V_1 = \begin{pmatrix} W_1 \\ 0 \end{pmatrix}, V_\ell = \begin{pmatrix} W_\ell & 0 \\ 0 & 0 \end{pmatrix}, V_L = \begin{pmatrix} W_L & 0 \end{pmatrix}.$$

Again inclusions preserve the network matrix as well as gradient flow.

Matrix Completion

In practice, DLNs are commonly used for Matrix Completion (MC) where a $d_{out} \times d_{in}$ matrix A^* is reconstructed from a subset of its entries $A^*_{i_1 j_1}, \dots, A^*_{i_N j_N}$. It is of course impossible to reconstruct a general matrix from a subset of its entries, but under the assumption that A^* is low rank, the minimal rank solution $\hat{A}$ (the matrix with the smallest rank amongst matrices whose entries match the observed entries of A^*) is a good estimator. Recovering $\hat{A}$ is in general NP hard [28], so a common strategy is to select the solution with minimal nuclear norm instead, which often matches the minimal rank solution [28].

Another strategy is to use a DLN fit the matrix A^* and to train it with gradient descent on the loss $\mathcal{L}^{MC}(\theta) = C^{MC}(A_\theta)$ where C^{MC} is the Matrix Completion cost

$$C^{MC}(A) = \frac{1}{N}\sum_{k=1}^{N} \left(A^*_{i_k j_k} - A_{\theta, i_k j_k}\right)^2.$$

The matrix $A_{\theta(t\to\infty)}$ learned in this manner approximates the true function A^* well, suggesting that the linear map learned by the DLN is low rank. However it is not obvious why gradient descent is naturally biased towards low-rank solutions.

Initialization Scale and Loss Surface

Our goal is to understand the training dynamics of DLN as a function of the initialization scale $-\gamma$: we initialize the parameters $\theta_1, \dots, \theta_P$ as i.i.d. Gaussian $\mathcal{N}(0, w^{-\gamma})$ with variance $w^{-\gamma}$ where $w = n_1 = \cdots = n_{L-1}$ is the width of the network. A small γ corresponds to large initialization and a large γ corresponds to small initialization. We will only consider the case $\gamma \geq 1 - \frac{1}{L}$, as any larger initialization scale leads to an exploding variance of the network matrix A_θ at initialization.

As the initialization scale γ increases and leaves the NTK regime $\gamma \geq 1$ we observe a significant change in the loss surface $\mathcal{L}(\theta)$ around initialization, suggesting the existence of three regimes:

Theorem 1.3. *Let the parameters θ be sampled with i.i.d. $\mathcal{N}(0, w^{-\gamma})$ entries and denote d_m and d_s the distance between θ and the closest global minimum resp. saddle of the MC loss*[9] *$\mathcal{L}^{MC}(\theta) = C(A_\theta)$. We have:*

1. ***NTK regime** ($1 - \frac{1}{L} \leq \gamma < 1$): $d_m \asymp w^{-\frac{(1-\gamma)(L-1)}{2}}$ and $d_s \asymp w^{\frac{1-\gamma}{2}}$.*
2. ***Critical regime** ($\gamma = 1$): $d_m \asymp 1$ and $d_s \asymp 1$.*

[9] The result actually holds for a general cost C with a few conditions, see Section 9.

3. ***Saddle-to-Saddle regime*** ***($\gamma > 1$):*** *$d_m \asymp 1$ and $d_s \asymp w^{-\frac{\gamma-1}{2}}$.*

For any two random variables $f(w)$ and $g(w)$ which depend on w, we write $f(w) \asymp g(w)$ if both $f(w)/g(w)$ and $g(w)/f(w)$ are stochastically bounded as $w \to \infty$.

Theorem 1.3 shows a significant change between large initializations $\gamma < 1$ where the initialization is close to a global minimum but far from any saddle and small initializations $\gamma > 1$ where the initialization is close to a saddle but far from any global minima.

In the NTK regime, gradient flow converges to one of the close global minima directly. In Section 1.9, we discuss the resulting dynamics using the limiting NTK and show the absence of low-rank bias in this regime, suggesting that the NTK regime should be avoided for Matrix Completion.

The critical regime has been studied for shallow non-linear networks ($L = 2$) in [35, 183] or for the deep case in [229]. This regime interpolates between the NTK and Saddle-to-Saddle regime: by changing the variance at initialization by a constant, i.e. taking $\sigma^2 = cw^{-1}$ for a constant c, one can obtain dynamics which are either close to the NTK regime for large c or close to the Saddle-to-Saddle regime for small c. In contrast, the asymptotic dynamics in the two other regimes should be less affected by such constant changes to the variance.

The dynamics in the Saddle-to-Saddle regime are studied in Section 1.9. A difficulty in the study of this regime is that since the initial parameters converge to a saddle as $w \to 0$, the time it takes for gradient descent to escape the saddle grows with w, and up to this 'escape time' nothing happens. In Section 1.9, we will fix the width w and let the variance of the parameters at initialization go to zero, which in a sense corresponds to the case $\gamma \to +\infty$. We conjecture that the dynamics observed in this limit are a good description of the whole regime $\gamma > 1$.

NTK Regime for Deep Linear Networks

The infinite-width limit with scaling $\gamma = 1 - \frac{1}{L}$ is the same (up to a rescaling of the learning rate) as the limit we studied in Sections 1.5 and 1.5. In this limit, the rescaled NTK $w^{-(1-\frac{1}{L})}\Theta^{(L)}$ converges to the deterministic and constant limiting kernel $\Theta_\infty^{(L)}(x, y) = Lx^T y$. The limiting dynamics of the network matrix A_θ are equivalent in this limit to the gradient descent on the cost $C : \mathbb{R}^{d_{out} \times d_{in}} \to \mathbb{R}$ directly:

$$\partial_t A_{\theta(t)} = -Lw^{(1-\frac{1}{L})} \nabla C(A_\theta).$$

Recent results [229] show that the NTK regime extends to initialization scales in the range $1 - \frac{1}{L} \geq \gamma > 1$, which is in line with our description of the loss surface from Theorem 1.3.

The training dynamics in the NTK regime exhibits no low-rank bias: when trained on the MC cost C^{MC}, the entries of the network matrix $A_{\theta(t),ij}$ at indices i, j which were not observed do not change during training in the infinite-width limit. At the end of training, the network matrix $A_{\theta(t\to\infty)}$ matches the true matrix A^* on the observed entries and has i.i.d. standard Gaussian $\mathcal{N}(0, 1)$ values in the unobserved entries. In other terms, a DLN in the linear regime returns random guesses on a Matrix Completion task. We argue that this is because of the lack of bias towards low rank matrices in the NTK regime, in contrast to the Saddle-to-Saddle whose low-rank bias we now discuss.

Saddle-to-Saddle Dynamics

Given a random initialization θ_0 with i.i.d. standard Gaussian entries $\mathcal{N}(0, 1)$, our goal is to study the dynamics of gradient flow $\theta_\alpha(t)$ initialized at $\alpha\theta_0$ in the limit as $\alpha \searrow 0$, which should be

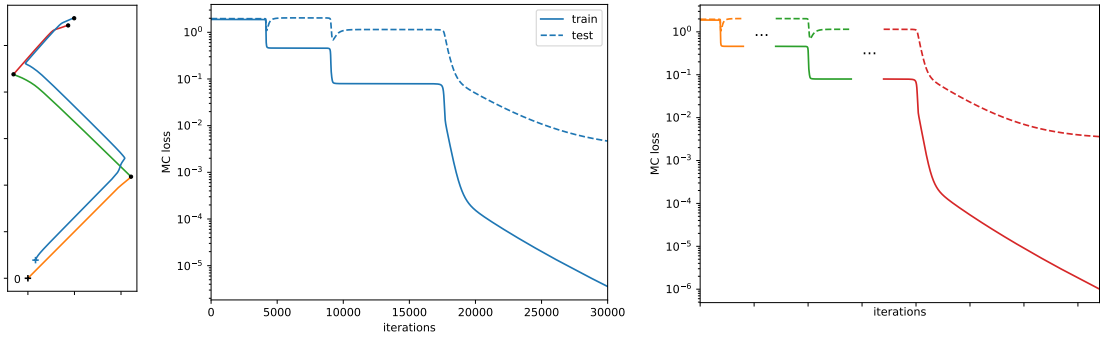


Figure 1.9.1: *Saddle-to-Saddle dynamics:* A DLN ($L = 4, w = 100$) with a small initialization ($\gamma = 2$) trained on a MC loss fitting a 10×10 matrix of rank 3. **Left:** Projection onto a plane of the gradient flow path θ_α in parameter space (in blue) and of the sequence of 3 limiting paths (in orange, green and red), starting from the origin ($+$) and passing through 2 saddles ($\cdot$) before converging. **Middle:** Train (solid) and test (dashed) MC costs through training. We observe three plateaus, corresponding to the three saddles visited. **Right:** The train (solid) and test (dashed) losses of the three paths plotted sequentially, in the saddle-to-saddle limit; the dots represent an infinite amount of steps separating these paths.

representative of the dynamics in the Saddle-to-Saddle regime ($\gamma > 1$). Since there is a saddle-point at the origin in parameter space, the limiting dynamics at any finite time t is trivial: $\lim_{\alpha \searrow 0} \theta_\alpha(t) = 0$.

However, under the assumption that gradient flow escapes the saddle at the origin (which we show happens with prob. 1 when $L \leq 3$ and with prob. at least $1/2$ otherwise), we can define an *escape time* t_α such that the limit $\lim_{\alpha \searrow 0} \theta_\alpha(t_\alpha + t)$ is non-trivial for any fixed t. The escape time scales as $-\log \alpha$ for shallow networks ($L = 2$) and as $\alpha^{-(L-2)}$ for deep networks ($L > 2$). The difference in scaling is due to the fact that for shallow networks the saddle at the origin is strict (i.e. the Hessian at the origin is non-zero) whereas for deep networks it is not strict (the first $L-1$ derivatives of the loss at the origin vanish). This limiting path $\lim_{\alpha \searrow 0} \theta_\alpha(t_\alpha + t)$ is unique up to symmetries:

Theorem 1.4. *(sketch) Under the assumption that the gradient flow path escapes the saddle, there is a gradient flow path $\underline{\theta}^1(t)$ of a width 1 network such that*

$$\lim_{\alpha \searrow 0} \theta_\alpha(t_\alpha + t) = RI^{(1 \to w)} \underline{\theta}^1(t),$$

where only the escape time t_α and the rotation R depend on the random initialization θ_0.

Sketch of proof. The proof relies on the fact that gradient flow naturally escapes along an *optimal escape path*, i.e. a gradient flow path $\theta(t)$ which escapes the saddle at the origin ($\lim_{t \to -\infty} \theta(t) = 0$) at an optimal rate. We then show a bijection between these optimal escape paths and the optimal escape paths of L-th order Taylor approximation of the flow around the origin, allowing us to show the unicity (up to symmetries) of these optimal escape paths.

The bijection between optimal escape paths at the heart of the proof can be extended to *fast escape paths* (paths that escape at a rate larger than some lower bound) and we prove its existence

for a general loss. It can be viewed as a generalization of the Hartman-Grobmann Theorem to non-strict saddles and might be of independent interest to study non-strict saddles. □

Remark 1.8. A weaker version of this result was proven in [135]. The distinctions between the two are discussed in the original paper [107], see Section 9. The main advantage of our approach is that it does not rely on tricks specific to linear networks, which is important if we want to one day generalize these results to the non-linear case.

Theorem 1.4 implies that for small α the gradient flow path $\theta_\alpha(t)$ will first get stuck at the saddle at the origin up to a time t_α, after which it will follow the inclusion of a width 1 path $\underline{\theta}^1(t)$. The path $\underline{\theta}^1(t)$ converges to a critical point $\underline{\vartheta}^1$ of the width 1 loss as $t \to \infty$, while $\underline{\vartheta}^1$ is typically a local minimum amongst width 1 network, its inclusion $RI^{(1\to w)}\underline{\vartheta}^1$ will typically be a saddle if $w > 1$. Theorem 1.4 implies that as $\alpha \searrow 0$ the gradient flow path θ_α will approach this new saddle $RI^{(1\to w)}\underline{\vartheta}^1$. At this point, we conjecture that gradient flow will escape this second saddle along the inclusion of a width 2 path $RI^{(2\to w)}\underline{\theta}^2(t)$ and then approach another saddle $RI^{(2\to w)}\underline{\vartheta}^2(t)$ and so on and so forth until reaching a global minimum.

This can be interpreted as DLNs implementing a greedy low-rank algorithm, which tries to minimize the cost C first among the matrices of rank 1, then those of rank 2, and so on until reaching a global minimum (a more detailed version of this algorithm is presented in the paper [107], see Section 9). While this algorithm might not always recover the minimal rank solution, it has a clear low-rank bias.

These Saddle-to-Saddle dynamics are visible when one plots the train and test error throughout training. Each of the saddles that is approached leads to a plateau where both test and train error remain almost constant for many gradient descent steps. As α gets smaller, these plateaus become longer.

Remark 1.9. Note that Theorem 1.4 implies that the time evolution of the NTK is significant. Indeed we know that at initialization the NTK is of order $\alpha^{2(L-1)}$. However at the escape time t_α, it is of order 1 (since as $\alpha \searrow 0$ the parameters at the escape time $\theta_\alpha(t_\alpha)$ converge to a set of parameters $RI^{(1\to w)}\underline{\theta}^1(0)$ with a non-zero NTK). In the $\alpha \searrow 0$ limit, the change in time of the NTK becomes infinitely larger than the size of the NTK at initialization.

1.10 Conclusion

This thesis started with the introduction of the NTK and a proof of its convergence to a deterministic and constant limit as the width of the network grows to infinity. This result implies the existence of a NTK regime where DNNs can be approximated by their tangent linear models, leading to surprisingly simple training dynamics.

The NTK analysis can be extended to describe the loss surface of DNNs along the training path in the infinite-width limit, revealing the fact that gradient flow remains in a region of the loss surface of DNNs where the dynamics resembles that of a convex function.

The limiting dynamics of DNNs as described by the NTK imply a direct link between DNNs and Kernel Ridge Regression (KRR). Relying and improving upon tools from Random Matrix Theory, the test error of KRR – and as an extension that of infinitely-wide DNNs – can be approximated in terms of the eigendecomposition of the kernel.

These results reveal the importance of the spectral decay of the NTK to understand both convergence and generalization of DNNs. The spectral bias of DNNs is affected by architectural choices such as the non-linearity σ, the bias strength β, the depth L as well as the use of normalization.

Analysing these effects helps better understand practical problems such as mode collapse in GANs, leading to solutions backed by theory. Likewise, the NTK analysis of DNN-based topology optimization leads to theoretically-motivated architecture choices to ensure translation invariances inherent in the problem, and to tune the level of details in the final shape.

The NTK allows for a very precise description of infinitely wide DNNs, it is natural to ask how similar finite-width networks are to their infinite-width counterparts. An empirical analysis identified a number of features of the test loss of DNNs as the width grows, related to the Double Descent curve. These features can be analyzed mathematically in the Random Features setup, which is closely related to DNNs in the NTK regime.

Thanks to these results, and thanks to the contributions of many other researchers, our understanding of the NTK regime is now almost complete. However there exists a number of active regimes, characterized by a non-constant NTK, whose dynamics remain ill-understood at the moment. We analyze one such active regime in linear networks, which appears for very small initialization, where the training path approaches a sequence of saddles, each corresponding to linear maps of increasing rank, leading to a bias towards low-rank solution.

The NTK is a first step in the development of a conceptual understanding of DNNs, with a well-understood NTK regime and a number of less understood active regimes. This suggests a strategy for the development of a theory of Deep Learning: we need to identify these regimes, understand under which condition they arise and describe the resulting dynamics and generalization properties. In spite of their differences, active regimes are linked by some common properties, such as feature learning and some form of sparsity, which are absent in the NTK regime. A lot of work remains to be done formalizing these properties and understanding how they arise.

Chapter 2

Neural Tangent Kernel: Convergence and Generalization in Neural Networks

Abstract

At initialization, artificial neural networks (ANNs) are equivalent to Gaussian processes in the infinite-width limit [159, 42, 47, 126, 46], thus connecting them to kernel methods. We prove that the evolution of an ANN during training can also be described by a kernel: during gradient descent on the parameters of an ANN, the network function f_θ (which maps input vectors to output vectors) follows the kernel gradient of the functional cost (which is convex, in contrast to the parameter cost) w.r.t. a new kernel: the Neural Tangent Kernel (NTK). This kernel is central to describe the generalization features of ANNs. While the NTK is random at initialization and varies during training, in the infinite-width limit it converges to an explicit limiting kernel and it stays constant during training. This makes it possible to study the training of ANNs in function space instead of parameter space. Convergence of the training can then be related to the positive-definiteness of the limiting NTK. We prove the positive-definiteness of the limiting NTK when the data is supported on the sphere and the non-linearity is non-polynomial.

We then focus on the setting of least-squares regression and show that in the infinite-width limit, the network function f_θ follows a linear differential equation during training. The convergence is fastest along the largest kernel principal components of the input data with respect to the NTK, hence suggesting a theoretical motivation for early stopping.

Finally we study the NTK numerically, observe its behavior for wide networks, and compare it to the infinite-width limit.

2.1 Introduction

Artificial neural networks (ANNs) have achieved impressive results in numerous areas of machine learning. While it has long been known that ANNs can approximate any function with sufficiently many hidden neurons [93, 133], it is not known what the optimization of ANNs converges to. Indeed the loss surface of neural networks optimization problems is highly non-convex: it has a high number of saddle points which may slow down the convergence [44]. A number of results [38, 170, 171] suggest that for wide enough networks, there are very few "bad" local minima, i.e. local minima with much higher cost than the global minimum. More recently, the investigation of

the geometry of the loss landscape at initialization has been the subject of a precise study [112]. The analysis of the dynamics of training in the large-width limit for shallow networks has seen recent progress as well [153]. To the best of the authors knowledge, the dynamics of deep networks has however remained an open problem until the present paper: see the contributions section below.

A particularly mysterious feature of ANNs is their good generalization properties in spite of their usual over-parametrization [190]. It seems paradoxical that a reasonably large neural network can fit random labels, while still obtaining good test accuracy when trained on real data [236]. It can be noted that in this case, kernel methods have the same properties [20].

In the infinite-width limit, ANNs have a Gaussian distribution described by a kernel [159, 42, 47, 126, 46]. These kernels are used in Bayesian inference or Support Vector Machines, yielding results comparable to ANNs trained with gradient descent [37, 126]. We will see that in the same limit, the behavior of ANNs during training is described by a related kernel, which we call the neural tangent network (NTK).

Contribution

We study the network function f_θ of an ANN, which maps an input vector to an output vector, where θ is the vector of the parameters of the ANN. In the limit as the widths of the hidden layers tend to infinity, the network function at initialization, f_θ converges to a Gaussian distribution [159, 42, 47, 126, 46].

In this paper, we investigate fully connected networks in this infinite-width limit, and describe the dynamics of the network function f_θ during training:

- During gradient descent, we show that the dynamics of f_θ follows that of the so-called *kernel gradient descent* in function space with respect to a limiting kernel, which only depends on the depth of the network, the choice of nonlinearity and the initialization variance.

- The convergence properties of ANNs during training can then be related to the positive-definiteness of the infinite-width limit NTK. In the case when the dataset is supported on a sphere, we prove this positive-definiteness using recent results on dual activation functions [42]. The values of the network function f_θ outside the training set is described by the NTK, which is crucial to understand how ANN generalize.

- For a least-squares regression loss, the network function f_θ follows a linear differential equation in the infinite-width limit, and the eigenfunctions of the Jacobian are the kernel principal components of the input data. This shows a direct connection to kernel methods and motivates the use of early stopping to reduce overfitting in the training of ANNs.

- Finally we investigate these theoretical results numerically for an artificial dataset (of points on the unit circle) and for the MNIST dataset. In particular we observe that the behavior of wide ANNs is close to the theoretical limit.

2.2 Neural networks

In this article, we consider fully-connected ANNs with layers numbered from 0 (input) to L (output), each containing $n_0, \dots, n_L$ neurons, and with a Lipschitz, twice differentiable nonlinearity function

$\sigma : \mathbb{R} \to \mathbb{R}$, with bounded second derivative [1].

This paper focuses on the ANN *realization function* $F^{(L)} : \mathbb{R}^P \to \mathcal{F}$, mapping parameters θ to functions f_θ in a space $\mathcal{F}$. The dimension of the parameter space is $P = \sum_{\ell=0}^{L-1}(n_\ell + 1)n_{\ell+1}$: the parameters consist of the connection matrices $W^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell+1}}$ and bias vectors $b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}}$ for $\ell = 0, ..., L-1$. In our setup, the parameters are initialized as iid Gaussians $\mathcal{N}(0, 1)$.

For a fixed distribution p^{in} on the input space $\mathbb{R}^{n_0}$, the function space $\mathcal{F}$ is defined as $\{f : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}\}$. On this space, we consider the seminorm $||\cdot||_{p^{in}}$, defined in terms of the bilinear form

$$\langle f, g \rangle_{p^{in}} = \mathbb{E}_{x \sim p^{in}} \left[f(x)^T g(x) \right].$$

In this paper, we assume that the input distribution p^{in} is the empirical distribution on a finite dataset $x_1, ..., x_N$, i.e the sum of Dirac measures $\frac{1}{N}\sum_{i=0}^{N} \delta_{x_i}$.

We define the network function by $f_\theta(x) := \tilde{\alpha}^{(L)}(x; \theta)$, where the functions $\tilde{\alpha}^{(\ell)}(\cdot; \theta) : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}$ (called *preactivations*) and $\alpha^{(\ell)}(\cdot; \theta) : \mathbb{R}^{n_0} \to \mathbb{R}^{n_\ell}$ (called *activations*) are defined from the 0-th to the L-th layer by:

$$\begin{aligned} \alpha^{(0)}(x; \theta) &= x \\ \tilde{\alpha}^{(\ell+1)}(x; \theta) &= \frac{1}{\sqrt{n_\ell}} W^{(\ell)} \alpha^{(\ell)}(x; \theta) + \beta b^{(\ell)} \\ \alpha^{(\ell)}(x; \theta) &= \sigma(\tilde{\alpha}^{(\ell)}(x; \theta)), \end{aligned}$$

where the nonlinearity σ is applied entrywise. The scalar $\beta > 0$ is a parameter which allows us to tune the influence of the bias on the training.

Remark 2.1. Our definition of the realization function $F^{(L)}$ slightly differs from the classical one. Usually, the factors $\frac{1}{\sqrt{n_\ell}}$ and the parameter β are absent and the parameters are initialized using what is sometimes called LeCun initialization, taking $W_{ij}^{(\ell)} \sim \mathcal{N}(0, \frac{1}{n_\ell})$ and $b_j^{(\ell)} \sim \mathcal{N}(0, 1)$ (or sometimes $b_j^{(\ell)} = 0$) to compensate. While the set of representable functions $F^{(L)}(\mathbb{R}^P)$ is the same for both parametrizations (with or without the factors $\frac{1}{\sqrt{n_\ell}}$ and β), the derivatives of the realization function with respect to the connections $\partial_{W_{ij}^{(\ell)}} F^{(L)}$ and bias $\partial_{b_j^{(\ell)}} F^{(L)}$ are scaled by $\frac{1}{\sqrt{n_\ell}}$ and β respectively in comparison to the classical parametrization.

The factors $\frac{1}{\sqrt{n_\ell}}$ are key to obtaining a consistent asymptotic behavior of neural networks as the widths of the hidden layers $n_1, ..., n_{L-1}$ grow to infinity. However a side-effect of these factors is that they reduce greatly the influence of the connection weights during training when n_ℓ is large: the factor β is introduced to balance the influence of the bias and connection weights. In our numerical experiments, we take $\beta = 0.1$ and use a learning rate of 1.0, which is larger than usual, see Section 2.6. This gives a behaviour similar to that of a classical network of width 100 with a learning rate of 0.01.

2.3 Kernel gradient

The training of an ANN consists in optimizing f_θ in the function space $\mathcal{F}$ with respect to a functional cost $C : \mathcal{F} \to \mathbb{R}$, such as a regression or cross-entropy cost. Even for a convex functional cost C, the

[1] While these smoothness assumptions greatly simplify the proofs of our results, they do not seem to be strictly needed for the results to hold true.

composite cost $C \circ F^{(L)} : \mathbb{R}^P \to \mathbb{R}$ is in general highly non-convex [38]. We will show that during training, the network function f_θ follows a descent along the kernel gradient with respect to the Neural Tangent Kernel (NTK) which we introduce in Section 2.4. This makes it possible to study the training of ANNs in the function space $\mathcal{F}$, on which the cost C is convex.

A *multi-dimensional kernel* K is a function $\mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}^{n_L \times n_L}$, which maps any pair (x, x') to an $n_L \times n_L$-matrix such that $K(x, x') = K(x', x)^T$ (equivalently K is a symmetric tensor in $\mathcal{F} \otimes \mathcal{F}$). Such a kernel defines a bilinear map on $\mathcal{F}$, taking the expectation over independent $x, x' \sim p^{in}$:

$$\langle f, g \rangle_K := \mathbb{E}_{x,x' \sim p^{in}} \left[f(x)^T K(x, x') g(x') \right].$$

The kernel K is *positive definite with respect to* $|| \cdot ||_{p^{in}}$ if $||f||_{p^{in}} > 0 \implies ||f||_K > 0$.

We denote by $\mathcal{F}^*$ the dual of $\mathcal{F}$ with respect to p^{in}, i.e. the set of linear forms $\mu : \mathcal{F} \to \mathbb{R}$ of the form $\mu = \langle d, \cdot \rangle_{p^{in}}$ for some $d \in \mathcal{F}$. Two elements of $\mathcal{F}$ define the same linear form if and only if they are equal on the data. The constructions in the paper do not depend on the element $d \in \mathcal{F}$ chosen in order to represent μ as $\langle d, \cdot \rangle_{p^{in}}$. Using the fact that the partial application of the kernel $K_{i,\cdot}(x, \cdot)$ is a function in $\mathcal{F}$, we can define a map $\Phi_K : \mathcal{F}^* \to \mathcal{F}$ mapping a dual element $\mu = \langle d, \cdot \rangle_{p^{in}}$ to the function $f_\mu = \Phi_K(\mu)$ with values:

$$f_{\mu,i}(x) = \mu K_{i,\cdot}(x, \cdot) = \langle d, K_{i,\cdot}(x, \cdot) \rangle_{p^{in}}.$$

For our setup, which is that of a finite dataset $x_1, \ldots, x_n \in \mathbb{R}^{n_0}$, the cost functional C only depends on the values of $f \in \mathcal{F}$ at the data points. As a result, the (functional) derivative of the cost C at a point $f_0 \in \mathcal{F}$ can be viewed as an element of $\mathcal{F}^*$, which we write $\partial_f^{in} C|_{f_0}$. We denote by $d|_{f_0} \in \mathcal{F}$, a corresponding dual element, such that $\partial_f^{in} C|_{f_0} = \langle d|_{f_0}, \cdot \rangle_{p^{in}}$.

The *kernel gradient* $\nabla_K C|_{f_0} \in \mathcal{F}$ is defined as $\Phi_K \left(\partial_f^{in} C|_{f_0} \right)$. In contrast to $\partial_f^{in} C$ which is only defined on the dataset, the kernel gradient generalizes to values x outside the dataset thanks to the kernel K:

$$\nabla_K C|_{f_0}(x) = \frac{1}{N} \sum_{j=1}^{N} K(x, x_j) d|_{f_0}(x_j).$$

A time-dependent function $f(t)$ follows the *kernel gradient descent with respect to* K if it satisfies the differential equation

$$\partial_t f(t) = -\nabla_K C|_{f(t)}.$$

During kernel gradient descent, the cost $C(f(t))$ evolves as

$$\partial_t C|_{f(t)} = -\left\langle d|_{f(t)}, \nabla_K C|_{f(t)} \right\rangle_{p^{in}} = -\left\| d|_{f(t)} \right\|_K^2.$$

Convergence to a critical point of C is hence guaranteed if the kernel K is positive definite with respect to $|| \cdot ||_{p^{in}}$: the cost is then strictly decreasing except at points such that $||d|_{f(t)}||_{p^{in}} = 0$. If the cost is convex and bounded from below, the function $f(t)$ therefore converges to a global minimum as $t \to \infty$.

Random functions approximation

As a starting point to understand the convergence of ANN gradient descent to kernel gradient descent in the infinite-width limit, we introduce a simple model, inspired by the approach of [176].

A kernel K can be approximated by a choice of P random functions $f^{(p)}$ sampled independently from any distribution on $\mathcal{F}$ whose (non-centered) covariance is given by the kernel K:

$$\mathbb{E}[f_k^{(p)}(x) f_{k'}^{(p)}(x')] = K_{kk'}(x, x').$$

These functions define a random linear parametrization $F^{lin} : \mathbb{R}^P \to \mathcal{F}$

$$\theta \mapsto f_\theta^{lin} = \frac{1}{\sqrt{P}} \sum_{p=1}^{P} \theta_p f^{(p)}.$$

The partial derivatives of the parametrization are given by

$$\partial_{\theta_p} F^{lin}(\theta) = \frac{1}{\sqrt{P}} f^{(p)}.$$

Optimizing the cost $C \circ F^{lin}$ through gradient descent, the parameters follow the ODE:

$$\partial_t \theta_p(t) = -\partial_{\theta_p}(C \circ F^{lin})(\theta(t)) = -\frac{1}{\sqrt{P}} \partial_f^{in} C|_{f_{\theta(t)}^{lin}} \, f^{(p)} = -\frac{1}{\sqrt{P}} \left\langle d|_{f_{\theta(t)}^{lin}}, f^{(p)} \right\rangle_{p^{in}}.$$

As a result the function $f_{\theta(t)}^{lin}$ evolves according to

$$\partial_t f_{\theta(t)}^{lin} = \frac{1}{\sqrt{P}} \sum_{p=1}^{P} \partial_t \theta_p(t) f^{(p)} = -\frac{1}{P} \sum_{p=1}^{P} \left\langle d|_{f_{\theta(t)}^{lin}}, f^{(p)} \right\rangle_{p^{in}} f^{(p)},$$

where the right-hand side is equal to the kernel gradient $-\nabla_{\tilde{K}} C$ with respect to the *tangent kernel*

$$\tilde{K} = \sum_{p=1}^{P} \partial_{\theta_p} F^{lin}(\theta) \otimes \partial_{\theta_p} F^{lin}(\theta) = \frac{1}{P} \sum_{p=1}^{P} f^{(p)} \otimes f^{(p)}.$$

This is a random n_L-dimensional kernel with values $\tilde{K}_{ii'}(x, x') = \frac{1}{P} \sum_{p=1}^{P} f_i^{(p)}(x) f_{i'}^{(p)}(x')$.

Performing gradient descent on the cost $C \circ F^{lin}$ is therefore equivalent to performing kernel gradient descent with the tangent kernel $\tilde{K}$ in the function space. In the limit as $P \to \infty$, by the law of large numbers, the (random) tangent kernel $\tilde{K}$ tends to the fixed kernel K, which makes this method an approximation of kernel gradient descent with respect to the limiting kernel K.

2.4 Neural tangent kernel

For ANNs trained using gradient descent on the composition $C \circ F^{(L)}$, the situation is very similar to that studied in the Section 2.3. During training, the network function f_θ evolves along the (negative) kernel gradient

$$\partial_t f_{\theta(t)} = -\nabla_{\Theta^{(L)}} C|_{f_{\theta(t)}}$$

with respect to the *neural tangent kernel* (NTK)

$$\Theta^{(L)}(\theta) = \sum_{p=1}^{P} \partial_{\theta_p} F^{(L)}(\theta) \otimes \partial_{\theta_p} F^{(L)}(\theta).$$

However, in contrast to F^{lin}, the realization function $F^{(L)}$ of ANNs is not linear. As a consequence, the derivatives $\partial_{\theta_p} F^{(L)}(\theta)$ and the neural tangent kernel depend on the parameters θ. The NTK is therefore random at initialization and varies during training, which makes the analysis of the convergence of f_θ more delicate.

In the next subsections, we show that, in the infinite-width limit, the NTK becomes deterministic at initialization and stays constant during training. Since f_θ at initialization is Gaussian in the limit, the asymptotic behavior of f_θ during training can be explicited in the function space $\mathcal{F}$.

Initialization

As observed in [159, 42, 47, 126, 46], the output functions $f_{\theta,i}$ for $i = 1, ..., n_L$ tend to iid Gaussian processes in the infinite-width limit (a proof in our setup is given in the appendix):

Proposition 2.1. *For a network of depth L at initialization, with a Lipschitz nonlinearity σ, and in the limit as $n_1, ..., n_{L-1} \to \infty$, the output functions $f_{\theta,k}$, for $k = 1, ..., n_L$, tend (in law) to iid centered Gaussian processes of covariance $\Sigma^{(L)}$, where $\Sigma^{(L)}$ is defined recursively by:*

$$\Sigma^{(1)}(x, x') = \frac{1}{n_0} x^T x' + \beta^2$$
$$\Sigma^{(L+1)}(x, x') = \mathbb{E}_{f \sim \mathcal{N}\left(0, \Sigma^{(L)}\right)}[\sigma(f(x))\sigma(f(x'))] + \beta^2,$$

taking the expectation with respect to a centered Gaussian process f of covariance $\Sigma^{(L)}$.

Remark 2.2. Strictly speaking, the existence of a suitable Gaussian measure with covariance $\Sigma^{(L)}$ is not needed: we only deal with the values of f at x, x' (the joint measure on $f(x), f(x')$ is simply a Gaussian vector in 2D). For the same reasons, in the proof of Proposition B.1 and Theorem 2.1, we will freely speak of Gaussian processes without discussing their existence.

The first key result of our paper (proven in the appendix) is the following: in the same limit, the Neural Tangent Kernel (NTK) converges in probability to an explicit deterministic limit.

Theorem 2.1. *For a network of depth L at initialization, with a Lipschitz nonlinearity σ, and in the limit as the layers width $n_1, ..., n_{L-1} \to \infty$, the NTK $\Theta^{(L)}$ converges in probability to a deterministic limiting kernel:*

$$\Theta^{(L)} \to \Theta_\infty^{(L)} \otimes Id_{n_L}.$$

The scalar kernel $\Theta_\infty^{(L)} : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}$ is defined recursively by

$$\Theta_\infty^{(1)}(x, x') = \Sigma^{(1)}(x, x')$$
$$\Theta_\infty^{(L+1)}(x, x') = \Theta_\infty^{(L)}(x, x')\dot{\Sigma}^{(L+1)}(x, x') + \Sigma^{(L+1)}(x, x'),$$

where

$$\dot{\Sigma}^{(L+1)}(x, x') = \mathbb{E}_{f \sim \mathcal{N}\left(0, \Sigma^{(L)}\right)}\left[\dot{\sigma}(f(x))\,\dot{\sigma}(f(x'))\right],$$

taking the expectation with respect to a centered Gaussian process f of covariance $\Sigma^{(L)}$, and where $\dot{\sigma}$ denotes the derivative of σ.

Remark 2.3. By Rademacher's theorem, $\dot{\sigma}$ is defined everywhere, except perhaps on a set of zero Lebesgue measure.

Note that the limiting $\Theta_\infty^{(L)}$ only depends on the choice of σ, the depth of the network and the variance of the parameters at initialization (which is equal to 1 in our setting).

Training

Our second key result is that the NTK stays asymptotically constant during training. This applies for a slightly more general definition of training: the parameters are updated according to a training direction $d_t \in \mathcal{F}$:

$$\partial_t \theta_p(t) = \left\langle \partial_{\theta_p} F^{(L)}(\theta(t)), d_t \right\rangle_{p^{in}}.$$

In the case of gradient descent, $d_t = -d|_{f_{\theta(t)}}$ (see Section 2.3), but the direction may depend on another network, as is the case for e.g. Generative Adversarial Networks [78]. We only assume that the integral $\int_0^T \|d_t\|_{p^{in}} dt$ stays stochastically bounded as the width tends to infinity, which is verified for e.g. least-squares regression, see Section 2.5.

Theorem 2.2. *Assume that σ is a Lipschitz, twice differentiable nonlinearity function, with bounded second derivative. For any T such that the integral $\int_0^T \|d_t\|_{p^{in}} dt$ stays stochastically bounded, as $n_1, ..., n_{L-1} \to \infty$, we have, uniformly for $t \in [0, T]$,*

$$\Theta^{(L)}(t) \to \Theta_\infty^{(L)} \otimes Id_{n_L}.$$

As a consequence, in this limit, the dynamics of f_θ is described by the differential equation

$$\partial_t f_{\theta(t)} = \Phi_{\Theta_\infty^{(L)} \otimes Id_{n_L}} \left(\langle d_t, \cdot \rangle_{p^{in}} \right).$$

Remark 2.4. As the proof of the theorem (in the appendix) shows, the variation during training of the individual activations in the hidden layers shrinks as their width grows. However their collective variation is significant, which allows the parameters of the lower layers to learn: in the formula of the limiting NTK $\Theta_\infty^{(L+1)}(x, x')$ in Theorem 2.1, the second summand $\Sigma^{(L+1)}$ represents the learning due to the last layer, while the first summand represents the learning performed by the lower layers.

As discussed in Section 2.3, the convergence of kernel gradient descent to a critical point of the cost C is guaranteed for positive definite kernels. The limiting NTK is positive definite if the span of the derivatives $\partial_{\theta_p} F^{(L)}$, $p = 1, ..., P$ becomes dense in $\mathcal{F}$ w.r.t. the p^{in}-norm as the width grows to infinity. It seems natural to postulate that the span of the preactivations of the last layer (which themselves appear in $\partial_{\theta_p} F^{(L)}$, corresponding to the connection weights of the last layer) becomes dense in $\mathcal{F}$, for a large family of measures p^{in} and nonlinearities (see e.g. [93, 133] for classical theorems about ANNs and approximation). In the case when the dataset is supported on a sphere, the positive-definiteness of the limiting NTK can be shown using Gaussian integration techniques and existing positive-definiteness criteria, as given by the following proposition, proven in Appendix B.1:

Proposition 2.2. *For a non-polynomial Lipschitz nonlinearity σ, for any input dimension n_0, the restriction of the limiting NTK $\Theta_\infty^{(L)}$ to the unit sphere $\mathbb{S}^{n_0 - 1} = \{x \in \mathbb{R}^{n_0} : x^T x = 1\}$ is positive-definite if $L \geq 2$.*

2.5 Least-squares regression

Given a goal function f^* and input distribution p^{in}, the least-squares regression cost is

$$C(f) = \frac{1}{2} ||f - f^*||_{p^{in}}^2 = \frac{1}{2} \mathbb{E}_{x \sim p^{in}} \left[\|f(x) - f^*(x)\|^2 \right].$$

Theorems 2.1 and B.2 apply to an ANN trained on such a cost. Indeed the norm of the training direction $\|d(f)\|_{p^{in}} = \|f^* - f\|_{p^{in}}$ is strictly decreasing during training, bounding the integral. We are therefore interested in the behavior of a function f_t during kernel gradient descent with a kernel K (we are of course especially interested in the case $K = \Theta_\infty^{(L)} \otimes Id_{n_L}$):

$$\partial_t f_t = \Phi_K \left(\langle f^* - f, \cdot \rangle_{p^{in}} \right).$$

The solution of this differential equation can be expressed in terms of the map $\Pi : f \mapsto \Phi_K \left(\langle f, \cdot \rangle_{p^{in}} \right)$:

$$f_t = f^* + e^{-t\Pi}(f_0 - f^*)$$

where $e^{-t\Pi} = \sum_{k=0}^{\infty} \frac{(-t)^k}{k!} \Pi^k$ is the exponential of $-t\Pi$. If Π can be diagonalized by eigenfunctions $f^{(i)}$ with eigenvalues λ_i, the exponential $e^{-t\Pi}$ has the same eigenfunctions with eigenvalues $e^{-t\lambda_i}$.

For a finite dataset $x_1, ..., x_N$ of size N, the map Π takes the form

$$\Pi(f)_k(x) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k'=1}^{n_L} f_{k'}(x_i) K_{kk'}(x_i, x).$$

The map Π has at most Nn_L positive eigenfunctions, and they are the kernel principal components $f^{(1)}, ..., f^{(Nn_L)}$ of the data with respect to to the kernel K [198, 200]. The corresponding eigenvalues λ_i is the variance captured by the component.

Decomposing the difference $(f^* - f_0) = \Delta_f^0 + \Delta_f^1 + ... + \Delta_f^{Nn_L}$ along the eigenspaces of Π, the trajectory of the function f_t reads

$$f_t = f^* + \Delta_f^0 + \sum_{i=1}^{Nn_L} e^{-t\lambda_i} \Delta_f^i,$$

where Δ_f^0 is in the kernel (null-space) of Π and $\Delta_f^i \propto f^{(i)}$.

The above decomposition can be seen as a motivation for the use of early stopping. The convergence is indeed faster along the eigenspaces corresponding to larger eigenvalues λ_i. Early stopping hence focuses the convergence on the most relevant kernel principal components, while avoiding to fit the ones in eigenspaces with lower eigenvalues (such directions are typically the 'noisier' ones: for instance, in the case of the RBF kernel, lower eigenvalues correspond to high frequency functions).

Note that by the linearity of the map $e^{-t\Pi}$, if f_0 is initialized with a Gaussian distribution (as is the case for ANNs in the infinite-width limit), then f_t is Gaussian for all times t. Assuming that the kernel is positive definite on the data (implying that the $Nn_L \times Nn_L$ Gram marix $\tilde{K} = (K_{kk'}(x_i, x_j))_{ik,jk'}$ is invertible), as $t \to \infty$ limit, we get that $f_\infty = f^* + \Delta_f^0 = f_0 - \sum_i \Delta_f^i$ takes the form

$$f_{\infty,k}(x) = \kappa_{x,k}^T \tilde{K}^{-1} y^* + \left(f_0(x) - \kappa_{x,k}^T \tilde{K}^{-1} y_0 \right),$$

with the Nn_l-vectors $\kappa_{x,k}$, y^* and y_0 given by

$$\kappa_{x,k} = (K_{kk'}(x, x_i))_{i,k'}$$
$$y^* = (f_k^*(x_i))_{i,k}$$

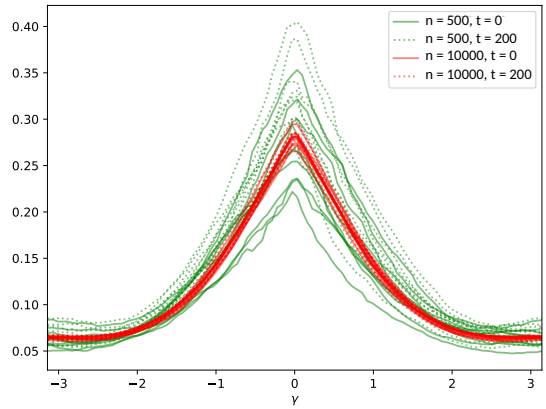


Figure 2.6.1: Convergence of the NTK to a fixed limit for two widths n and two times t.

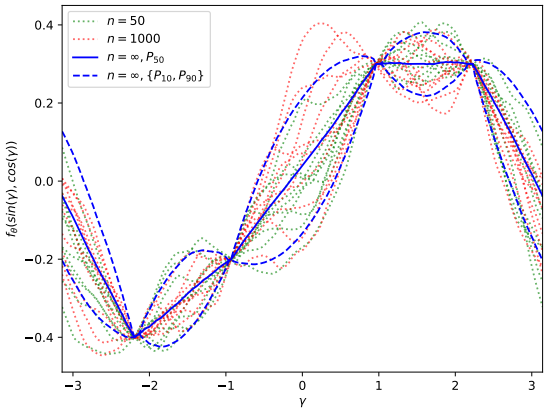


Figure 2.6.2: Networks function f_θ near convergence for two widths n and 10th, 50th and 90th percentiles of the asymptotic Gaussian distribution.

$$y_0 = (f_{0,k}(x_i))_{i,k}\,.$$

The first term, the mean, has an important statistical interpretation: it is the maximum-a-posteriori (MAP) estimate given a Gaussian prior on functions $f_k \sim \mathcal{N}(0, \Theta_\infty^{(L)})$ and the conditions $f_k(x_i) = f_k^*(x_i)$. Equivalently, it is equal to the kernel ridge regression [200] as the regularization goes to zero ($\lambda \to 0$). The second term is a centered Gaussian whose variance vanishes on the points of the dataset.

2.6 Numerical experiments

In the following numerical experiments, fully connected ANNs of various widths are compared to the theoretical infinite-width limit. We choose the size of the hidden layers to all be equal to the same value $n := n_1 = ... = n_{L-1}$ and we take the ReLU nonlinearity $\sigma(x) = \max(0, x)$.

In the first two experiments, we consider the case $n_0 = 2$. Moreover, the input elements are taken on the unit circle. This can be motivated by the structure of high-dimensional data, where the centered data points often have roughly the same norm [2].

In all experiments, we took $n_L = 1$ (note that by our results, a network with n_L outputs behaves asymptotically like n_L networks with scalar outputs trained independently). Finally, the value of the parameter β is chosen as 0.1, see Remark 2.1.

Convergence of the NTK

The first experiment illustrates the convergence of the NTK $\Theta^{(L)}$ of a network of depth $L = 4$ for two different widths $n = 500, 10000$. The function $\Theta^{(4)}(x_0, x)$ is plotted for a fixed $x_0 = (1, 0)$ and

[2]The classical example is for data following a Gaussian distribution $\mathcal{N}(0, Id_{n_0})$: as the dimension n_0 grows, all data points have approximately the same norm $\sqrt{n_0}$.

$x = (cos(\gamma), sin(\gamma))$ on the unit circle in Figure 2.6.1. To observe the distribution of the NTK, 10 independent initializations are performed for both widths. The kernels are plotted at initialization $t = 0$ and then after 200 steps of gradient descent with learning rate 1.0 (i.e. at $t = 200$). We approximate the function $f^*(x) = x_1 x_2$ with a least-squares cost on random $\mathcal{N}(0, 1)$ inputs.

For the wider network, the NTK shows less variance and is smoother. It is interesting to note that the expectation of the NTK is very close for both networks widths. After 200 steps of training, we observe that the NTK tends to "inflate". As expected, this effect is much less apparent for the wider network ($n = 10000$) where the NTK stays almost fixed, than for the smaller network ($n = 500$).

Kernel regression

For a regression cost, the infinite-width limit network function $f_{\theta(t)}$ has a Gaussian distribution for all times t and in particular at convergence $t \to \infty$ (see Section 2.5). We compared the theoretical Gaussian distribution at $t \to \infty$ to the distribution of the network function $f_{\theta(T)}$ of a finite-width network for a large time $T = 1000$. For two different widths $n = 50, 1000$ and for 10 random initializations each, a network is trained on a least-squares cost on 4 points of the unit circle for 1000 steps with learning rate 1.0 and then plotted in Figure 2.6.2.

We also approximated the kernels $\Theta_\infty^{(4)}$ and $\Sigma^{(4)}$ using a large-width network ($n = 10000$) and used them to calculate and plot the 10th, 50th and 90-th percentiles of the $t \to \infty$ limiting Gaussian distribution.

The distributions of the network functions are very similar for both widths: their mean and variance appear to be close to those of the limiting distribution $t \to \infty$. Even for relatively small widths ($n = 50$), the NTK gives a good indication of the distribution of $f_{\theta(t)}$ as $t \to \infty$.

Convergence along a principal component

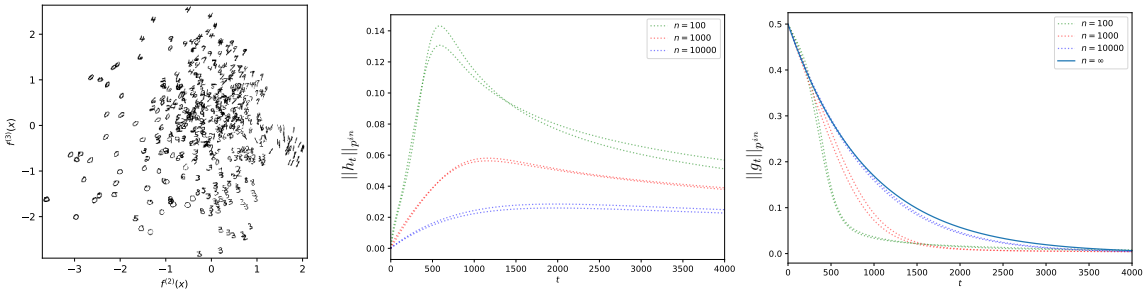
We now illustrate our result on the MNIST dataset of handwritten digits made up of grayscale images of dimension 28×28, yielding a dimension of $n_0 = 784$.

We computed the first 3 principal components of a batch of $N = 512$ digits with respect to the NTK of a high-width network $n = 10000$ (giving an approximation of the limiting kernel) using a power iteration method. The respective eigenvalues are $\lambda_1 = 0.0457$, $\lambda_2 = 0.00108$ and $\lambda_3 = 0.00078$. The kernel PCA is non-centered, the first component is therefore almost equal to the constant function, which explains the large gap between the first and second eigenvalues[3]. The next two components are much more interesting as can be seen in Figure 2.6.3a, where the batch is plotted with x and y coordinates corresponding to the 2nd and 3rd components.

We have seen in Section 2.5 how the convergence of kernel gradient descent follows the kernel principal components. If the difference at initialization $f_0 - f^*$ is equal (or proportional) to one of the principal components $f^{(i)}$, then the function will converge along a straight line (in the function space) to f^* at an exponential rate $e^{-\lambda_i t}$.

We tested whether ANNs of various widths $n = 100, 1000, 10000$ behave in a similar manner. We set the goal of the regression cost to $f^* = f_{\theta(0)} + 0.5 f^{(2)}$ and let the network converge. At each time step t, we decomposed the difference $f_{\theta(t)} - f^*$ into a component g_t proportional to $f^{(2)}$ and another one h_t orthogonal to $f^{(2)}$. In the infinite-width limit, the first component decays

[3]It can be observed numerically, that if we choose $\beta = 1.0$ instead of our recommended 0.1, the gap between the first and the second principal component is about ten times bigger, which makes training more difficult.



(a) The 2nd and 3rd principal components of MNIST.

(b) Deviation of the network function f_θ from the straight line.

(c) Convergence of f_θ along the 2nd principal component.

Figure 2.6.3: NTK PCA and convergence speed.

exponentially fast $||g_t||_{p^{in}} = 0.5e^{-\lambda_2 t}$ while the second is null ($h_t = 0$), as the function converges along a straight line.

As expected, we see in Figure 2.6.3b that the wider the network, the less it deviates from the straight line (for each width n we performed two independent trials). As the width grows, the trajectory along the 2nd principal component (shown in Figure 2.6.3c) converges to the theoretical limit shown in blue.

A surprising observation is that smaller networks appear to converge faster than wider ones. This may be explained by the inflation of the NTK observed in our first experiment. Indeed, multiplying the NTK by a factor a is equivalent to multiplying the learning rate by the same factor. However, note that since the NTK of large-width network is more stable during training, larger learning rates can in principle be taken. One must hence be careful when comparing the convergence speed in terms of the number of steps (rather than in terms of the time t): both the inflation effect and the learning rate must be taken into account.

2.7 Conclusion

This paper introduces a new tool to study ANNs, the Neural Tangent Kernel (NTK), which describes the local dynamics of an ANN during gradient descent. This leads to a new connection between ANN training and kernel methods: in the infinite-width limit, an ANN can be described in the function space directly by the limit of the NTK, an explicit constant kernel $\Theta_\infty^{(L)}$, which only depends on its depth, nonlinearity and parameter initialization variance. More precisely, in this limit, ANN gradient descent is shown to be equivalent to a kernel gradient descent with respect to $\Theta_\infty^{(L)}$. The limit of the NTK is hence a powerful tool to understand the generalization properties of ANNs, and it allows one to study the influence of the depth and nonlinearity on the learning abilities of the network. The analysis of training using NTK allows one to relate convergence of ANN training with the positive-definiteness of the limiting NTK and leads to a characterization of the directions favored by early stopping methods.

Chapter 3

The Asymptotic Spectrum of the Hessian of DNN Throughout Training

Abstract

The dynamics of DNNs during gradient descent is described by the so-called Neural Tangent Kernel (NTK). In this article, we show that the NTK allows one to gain precise insight into the Hessian of the cost of DNNs. When the NTK is fixed during training, we obtain a full characterization of the asymptotics of the spectrum of the Hessian, at initialization and during training. In the so-called mean-field limit, where the NTK is not fixed during training, we describe the first two moments of the Hessian at initialization.

3.1 Introduction

The advent of deep learning has sparked a lot of interest in the loss surface of deep neural networks (DNN), and in particular its Hessian. However to our knowledge, there is still no theoretical description of the spectrum of the Hessian. Nevertheless a number of phenomena have been observed numerically.

The loss surface of neural networks has been compared to the energy landscape of different physical models [38, 68, 153]. It appears that the loss surface of DNNs may change significantly depending on the width of the network (the number of neurons in the hidden layer), motivating the distinction between the under- and over-parametrized regimes [12, 68, 66].

The non-convexity of the loss function implies the existence of a very large number of saddle points, which could slow down training. In particular, in [170, 44], a relation between the rank of saddle points (the number of negative eigenvalues of the Hessian) and their loss has been observed.

For overparametrized DNNs, a possibly more important phenomenon is the large number of flat directions [12]. The existence of these flat minima is conjectured to be related to the generalization of DNNs and may depend on the training procedure [90, 32, 222].

In [105] it has been shown, using a functional approach, that in the infinite-width limit, DNNs behave like kernel methods with respect to the so-called Neural Tangent Kernel, which is determined by the architecture of the network. This leads to convergence guarantees for DNNs [105, 51, 2, 95] and strengthens the connections between neural networks and kernel methods [159, 37, 126].

Our approach also allows one to probe the so-called mean-field/active limit (studied in [183, 35, 153] for shallow networks), where the NTK varies during training.

This raises the question: can we use these new results to gain insight into the behavior of the Hessian of the loss of DNNs, at least in the small region explored by the parameters during training?

Contributions

Following ideas introduced in [105], we consider the training of $L+1$-layered DNNs in a functional setting. For a functional cost $\mathcal{C}$, the Hessian of the loss $\mathbb{R}^P \ni \theta \mapsto \mathcal{C}\left(F^{(L)}(\theta)\right)$ is the sum of two $P \times P$ matrices I and S. We show the following results for large P and for a fixed number of datapoints N:

- The first matrix I is positive semi-definite and its eigenvalues are given by the (weighted) kernel PCA of the dataset with respect to the NTK. The dominating eigenvalues are the principal components of the data followed by a high number of small eigenvalues. The "flat directions" are spanned by the small eigenvalues and the null-space (of dimension at least $P - N$ when there is a single output). Because the NTK is asymptotically constant [105], these results apply at initialization, during training and at convergence.

- The second matrix S can be viewed as residual contribution to H, since it vanishes as the network converges to a global minimum. We compute the limit of the first moment $\mathrm{Tr}\,(S)$ and characterize its evolution during training, of the second moment $\mathrm{Tr}\left(S^2\right)$ which stays constant during training, and show that the higher moments vanish.

- Regarding the sum $H = I + S$, we show that the matrices I and S are asymptotically orthogonal to each other at initialization and during training. In particular, the moments of the matrices I and S add up: $tr(H^k) \approx tr(I^k) + tr(S^k)$.

These results give, for any depth and a fairly general non-linearity, a complete description of the spectrum of the Hessian in terms of the NTK at initialization and throughout training. Our theoretical results are consistent with a number of observations about the Hessian [90, 170, 44, 32, 222, 171, 68], and sheds a new light on them.

Related works

The Hessian of the loss has been studied through the decomposition $I + S$ in a number of previous works [190, 171, 68].

For least-squares and cross-entropy costs, the first matrix I is equal to the Fisher matrix [219, 169], whose moments have been described for shallow networks in [172]. For deep networks, the first two moments and the operator norm of the Fisher matrix for a least squares loss were computed at initialization in [112] conditionally on a certain independence assumption; our method does not require such assumptions. Note that their approach implicitly uses the NTK.

The second matrix S has been studied in [171, 68] for shallow networks, conditionally on a number of assumptions. Note that in the setting of [171], the matrices I and S are assumed to be freely independent, which allows them to study the spectrum of the Hessian; in our setting, we show that the two matrices I and S are asymptotically orthogonal to each other.

3.2 Setup

We consider deep fully connected artificial neural networks (DNNs) using the setup and NTK parametrization of [105], taking an arbitrary nonlinearity $\sigma \in C_b^4(\mathbb{R})$ (i.e. $\sigma : \mathbb{R} \to \mathbb{R}$ that is 4 times continuously differentiable function with all four derivatives bounded). The layers are numbered from 0 (input) to L (output), each containing n_ℓ neurons for $\ell = 0, \dots, L$. The $P = \sum_{\ell=0}^{L-1} (n_\ell + 1)\, n_{\ell+1}$ parameters consist of the weight matrices $W^{(\ell)} \in \mathbb{R}^{n_{\ell+1} \times n_\ell}$ and bias vectors $b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}}$ for $\ell = 0, \dots, L-1$. We aggregate the parameters into the vector $\theta \in \mathbb{R}^P$.

The activations and pre-activations of the layers are defined recursively for an input $x \in \mathbb{R}^{n_0}$, setting $\alpha^{(0)}(x;\theta) = x$:

$$\tilde{\alpha}^{(\ell+1)}(x;\theta) = \frac{1}{\sqrt{n_\ell}} W^{(\ell)} \alpha^{(\ell)}(x;\theta) + \beta b^{(\ell)},$$
$$\alpha^{(\ell+1)}(x;\theta) = \sigma\big(\tilde{\alpha}^{(\ell+1)}(x;\theta)\big).$$

The parameter β is added to tune the influence of the bias on training[1]. All parameters are initialized as iid $\mathcal{N}(0,1)$ Gaussians.

We will in particular study the network function, which maps inputs x to the activation of the output layer (before the last non-linearity):

$$f_\theta(x) = \tilde{\alpha}^{(L)}(x;\theta).$$

In this paper, we will study the limit of various objects as $n_1, \dots, n_{L-1} \to \infty$ *sequentially*, i.e. we first take $n_1 \to \infty$, then $n_2 \to \infty$, etc. This greatly simplifies the proofs, but they could in principle be extended to the simultaneous limit, i.e. when $n_1 = ... = n_{L-1} \to \infty$. All our numerical experiments are done with 'rectangular' networks (with $n_1 = ... = n_{L-1}$) and match closely the predictions for the sequential limit.

In the limit we study in this paper, the NTK is asymptotically fixed, as in [105, 2, 51, 6, 95]. By rescaling the outputs of DNNs as the width increases, one can reach another limit where the NTK is not fixed [35, 34, 183, 151]. Some of our results can be extended to this setting, but only at initialization (see Section 3.3). The behavior during training becomes however much more complex.

Functional viewpoint

The network function lives in a function space $f_\theta \in \mathcal{F} := [\mathbb{R}^{n_0} \to \mathbb{R}^{n_L}]$ and we call the function $F^{(L)} : \mathbb{R}^P \to \mathcal{F}$ that maps the parameters θ to the network function f_θ the *realization function*. We study the differential behavior of $F^{(L)}$:

- The derivative $\mathcal{D}F^{(L)} \in \mathbb{R}^P \otimes \mathcal{F}$ is a function-valued vector of dimension P. The p-th entry $\mathcal{D}_p F^{(L)} = \partial_{\theta_p} f_\theta \in \mathcal{F}$ represents how modifying the parameter θ_p modifies the function f_θ in the space $\mathcal{F}$.
- The Hessian $\mathcal{H}F^{(L)} \in \mathbb{R}^P \otimes \mathbb{R}^P \otimes \mathcal{F}$ is a function-valued $P \times P$ matrix.

The network is trained with respect to the cost functional:

$$\mathcal{C}(f) = \frac{1}{N} \sum_{i=1}^{N} c_i\left(f(x_i)\right),$$

[1] In our experiments, we take $\beta = 0.1$.

for strictly convex c_i, summing over a finite dataset $x_1, \dots, x_N \in \mathbb{R}^{n_0}$ of size N. The parameters are then trained with gradient descent on the composition $\mathcal{C} \circ F^{(L)}$, which defines the usual loss surface of neural networks.

In this setting, we define the finite realization function $Y^{(L)} : \mathbb{R}^P \to \mathbb{R}^{Nn_L}$ mapping parameters θ to be the restriction of the network function f_θ to the training set $y_{ik} = f_{\theta,k}(x_i)$. The Jacobian $\mathcal{D}Y^{(L)}$ is hence an $Nn_L \times P$ matrix and its Hessian $\mathcal{H}Y^{(L)}$ is a $P \times P \times Nn_L$ tensor. Defining the restricted cost $C(y) = \frac{1}{N}\sum_i c_i(y_i)$, we have $\mathcal{C} \circ F^{(L)} = C \circ Y^{(L)}$.

For our analysis, we require that the gradient norm $\|\mathcal{D}C\|$ does not explode during training. The following condition is sufficient:

Definition 1. A loss $C : \mathbb{R}^{Nn_L} \to \mathbb{R}$ has bounded gradients over sublevel sets (BGOSS) if the norm of the gradient is bounded over all sets $U_a = \left\{Y \in \mathbb{R}^{Nn_L} : C(Y) \leq a\right\}$.

For example, the Mean Square Error (MSE) $C(Y) = \frac{1}{2N}\left\|Y^* - Y\right\|^2$ for the labels $Y^* \in \mathbb{R}^{Nn_L}$ has BGOSS because $\left\|\nabla C(Y)\right\|^2 = \frac{1}{N}\left\|Y^* - Y\right\|^2 = 2C(Y)$. For the binary and softmax cross-entropy the gradient is uniformly bounded, see Proposition C.1 in Appendix C.1.

Neural Tangent Kernel

The behavior during training of the network function f_θ in the function space $\mathcal{F}$ is described by a (multi-dimensional) kernel, the *Neural Tangent Kernel* (NTK)

$$\Theta^{(L)}_{k,k'}(x, x') = \sum_{p=1}^{P} \partial_{\theta_p} f_{\theta,k}(x) \partial_{\theta_p} f_{\theta,k'}(x').$$

During training, the function f_θ follows the so-called *kernel gradient descent* with respect to the NTK, which is defined as

$$\partial_t f_{\theta(t)}(x) = -\nabla_{\Theta^{(L)}} C_{|f_{\theta(t)}}(x) := -\frac{1}{N}\sum_{i=1}^{N} \Theta^{(L)}(x, x_i) \nabla c_i(f_{\theta(t)}(x_i)).$$

In the infinite-width limit (letting $n_1 \to \infty, \dots, n_{L-1} \to \infty$ sequentially) and for losses with BGOSS, the NTK converges to a deterministic limit $\Theta^{(L)} \to \Theta^{(L)}_\infty \otimes Id_{n_L}$, which is constant during training, uniformly on finite time intervals $[0, T]$ [105]. For the MSE loss, the uniform convergence of the NTK was proven for $T = \infty$ in [6].

The limiting NTK $\Theta^{(L)}_\infty : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}$ is constructed as follows:

1. For $f, g : \mathbb{R} \to \mathbb{R}$ and a kernel $K : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}$, define the kernel $\mathbb{L}^{f,g}_K : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}$ by

   $$\mathbb{L}^{f,g}_K(x_0, x_1) = \mathbb{E}_{(a_0,a_1)}\left[f(a_0)g(a_1)\right],$$

   for (a_0, a_1) a centered Gaussian vector with covariance matrix $(K(x_i, x_j))_{i,j=0,1}$. For $f = g$, we denote by $\mathbb{L}^f_K$ the kernel $\mathbb{L}^{f,f}_K$.

2. We define the kernels $\Sigma^{(\ell)}_\infty$ for each layer of the network, starting with $\Sigma^{(1)}_\infty(x_0, x_1) = {}^1\!/_{n_0}(x_0^T x_1) + \beta^2$ and then recursively by $\Sigma^{(\ell+1)}_\infty = \mathbb{L}^\sigma_{\Sigma^{(\ell)}_\infty} + \beta^2$, for $\ell = 1, \dots, L-1$, where σ is the network non-linearity.

3. The limiting NTK $\Theta_\infty^{(L)}$ is defined in terms of the kernels $\Sigma_\infty^{(\ell)}$ and the kernels $\dot{\Sigma}_\infty^{(\ell)} = \mathbb{L}_{\Sigma_\infty^{(\ell-1)}}^{\dot{\sigma}}$:

$$\Theta_\infty^{(L)} = \sum_{\ell=1}^{L} \Sigma_\infty^{(\ell)} \dot{\Sigma}_\infty^{(\ell+1)} \dots \dot{\Sigma}_\infty^{(L)}.$$

The NTK leads to convergence guarantees for DNNs in the infinite-width limit, and connect their generalization to that of kernel methods [105, 6].

Gram Matrices

For a finite dataset $x_1, \dots, x_N \in \mathbb{R}^{n_0}$ and a fixed depth $L \geq 1$, we denote by $\tilde{\Theta} \in \mathbb{R}^{Nn_L \times Nn_L}$ the Gram matrix of $x_1, \dots, x_N$ with respect to the limiting NTK, defined by

$$\tilde{\Theta}_{ik,jm} = \Theta_\infty^{(L)}(x_i, x_j)\, \delta_{km}.$$

It is block diagonal because different outputs $k \neq m$ are asymptotically uncorrelated.

Similarly, for any (scalar) kernel $\mathcal{K}^{(L)}$ (such as the limiting kernels $\Sigma_\infty^{(L)}, \Lambda_\infty^{(L)}, \Upsilon_\infty^{(L)}, \Phi_\infty^{(L)}, \Xi_\infty^{(L)}$ introduced later), we denote the Gram matrix of the datapoints by $\tilde{\mathcal{K}}$.

3.3 Main Theorems

Hessian as $I + S$

Using the above setup, the Hessian H of the loss $\mathcal{C} \circ F^{(L)}$ is the sum of two terms, with the entry $H_{p,p'}$ given by

$$H_{p,p'} = \mathcal{HC}_{|f_\theta}(\partial_{\theta_p} F, \partial_{\theta_{p'}} F) + \mathcal{DC}_{|f_\theta}(\partial_{\theta_p,\theta_{p'}} F).$$

For a finite dataset, the Hessian matrix $\mathcal{H}\left(C \circ Y^{(L)}\right)$ is equal to the sum of two matrices

$$I = \left(\mathcal{D}Y^{(L)}\right)^T \mathcal{HC}\mathcal{D}Y^{(L)} \quad \text{and} \quad S = \nabla C \cdot \mathcal{H}Y^{(L)}$$

where $\mathcal{D}Y^{(L)}$ is a $Nn_L \times P$ matrix, $\mathcal{HC}$ is a $Nn_L \times Nn_L$ matrix and $\mathcal{H}Y^{(L)}$ is a $P \times P \times Nn_L$ tensor to which we apply a scalar product (denoted by $\cdot$) in its last dimension with the Nn_L vector ∇C to obtain a $P \times P$ matrix.

Our main contribution is the following theorem, which describes the limiting moments $\mathrm{Tr}\left(H^k\right)$ in terms of the moments of I and S:

Theorem 3.1. *For any loss C with BGOSS and $\sigma \in C_b^4(\mathbb{R})$, in the sequential limit $n_1 \to \infty, \dots, n_{L-1} \to \infty$, we have for all $k \geq 1$*

$$\mathrm{Tr}\left(H(t)^k\right) \approx \mathrm{Tr}\left(I(t)^k\right) + \mathrm{Tr}\left(S(t)^k\right).$$

The limits of $\mathrm{Tr}\left(I(t)^k\right)$ and $\mathrm{Tr}\left(S(t)^k\right)$ can be expressed in terms of the NTK $\Theta_\infty^{(L)}$, the kernels $\Upsilon_\infty^{(L)}, \Xi_\infty^{(L)}$ and the non-symmetric kernels $\Phi_\infty^{(L)}$, $\Lambda_\infty^{(L)}$ defined in Appendix C.3:

- *The moments* $\mathrm{Tr}\left(I\left(t\right)^k\right)$ *converge to the following limits (with the convention that* $i_{k+1}=i_1$*):*

$$\mathrm{Tr}\left(I\left(t\right)^k\right)\to\mathrm{Tr}\left(\left(\mathcal{H}C(Y\left(t\right))\tilde{\Theta}\right)^k\right)=\frac{1}{N^k}\sum_{i_1,...,i_k=1}^{N}\prod_{m=1}^{k}c''_{i_m}(f_{\theta(t)}(x_{i_m}))\Theta_\infty^{(L)}(x_{i_m},x_{i_{m+1}}).$$

- *The first moment* $\mathrm{Tr}\left(S\left(t\right)\right)$ *converges to the limit:*

$$\mathrm{Tr}\left(S\left(t\right)\right)=\left(G(t)\right)^T\nabla C(Y(t)).$$

At initialization $(G(0),Y(0))$ *form a Gaussian pair of* Nn_L*-vectors, independent for differing output indices* $k=1,...,n_L$ *and with covariance* $\mathbb{E}\left[G_{ik}(0)G_{i'k'}(0)\right]=\delta_{kk'}\Xi_\infty^{(L)}(x_i,x_{i'})$ *and* $\mathbb{E}\left[G_{ik}(0)Y_{i'k'}(0)\right]=\delta_{kk'}\Phi_\infty^{(L)}(x_i,x_{i'})$ *for the limiting kernel* $\Xi_\infty^{(L)}(x,y)$ *and non-symmetric kernel* $\Phi_\infty^{(L)}(x,y)$*. During training, both vectors follow the differential equations*

$$\begin{aligned}\partial_t G(t)&=-\tilde{\Lambda}\nabla C(Y(t))\\ \partial_t Y(t)&=-\tilde{\Theta}\nabla C(Y(t)).\end{aligned}$$

- *The second moment* $\mathrm{Tr}\left(S\left(t\right)^2\right)$ *converges to the following limit defined in terms of the Gram matrix* $\tilde{\Upsilon}$*:*

$$\mathrm{Tr}\left(S^2\right)\to\left(\nabla C(Y(t))\right)^T\tilde{\Upsilon}\nabla C(Y(t))$$

- *The higher moments* $\mathrm{Tr}\left(S\left(t\right)^k\right)$ *for* $k\geq 3$ *vanish.*

Proof. The moments of I and S can be studied separately because the moments of their sum is asymptotically equal to the sum of their moments by Proposition C.4 below. The limiting moments of I and S are respectively described by Propositions 3.1 and C.3 below. □

In the case of a MSE loss $C(Y)=\frac{1}{2N}\left\|Y-Y^*\right\|^2$, the first and second derivatives take simple forms $\nabla C(Y)=\frac{1}{N}\left(Y-Y^*\right)$ and $\mathcal{H}C(Y)=\frac{1}{N}Id_{Nn_L}$ and the differential equations can be solved to obtain more explicit formulae:

Corollary 1. *For the MSE loss* C *and* $\sigma\in C_b^4(\mathbb{R})$*, in the limit* $n_1,...,n_{L-1}\to\infty$*, we have uniformly over* $[0,T]$

$$\mathrm{Tr}\left(H(t)^k\right)\to\frac{1}{N^k}\mathrm{Tr}\left(\tilde{\Theta}^k\right)+\mathrm{Tr}\left(S(t)^k\right)$$

where

$$\begin{aligned}\mathrm{Tr}\left(S(t)\right)\to&-\frac{1}{N}(Y^*-Y(0))^T\left(Id_{Nn_L}+e^{-t\tilde{\Theta}}\right)\tilde{\Theta}^{-1}\tilde{\Lambda}^Te^{-t\tilde{\Theta}}(Y^*-Y(0))\\&+\frac{1}{N}G(0)^Te^{-t\tilde{\Theta}}(Y^*-Y(0))\\ \mathrm{Tr}\left(S(t)^2\right)\to&\frac{1}{N^2}(Y^*-Y(0))^Te^{-t\tilde{\Theta}}\tilde{\Upsilon}e^{-t\tilde{\Theta}}(Y^*-Y(0))\\ \mathrm{Tr}\left(S(t)^k\right)\to&0\quad\textit{when } k>2.\end{aligned}$$

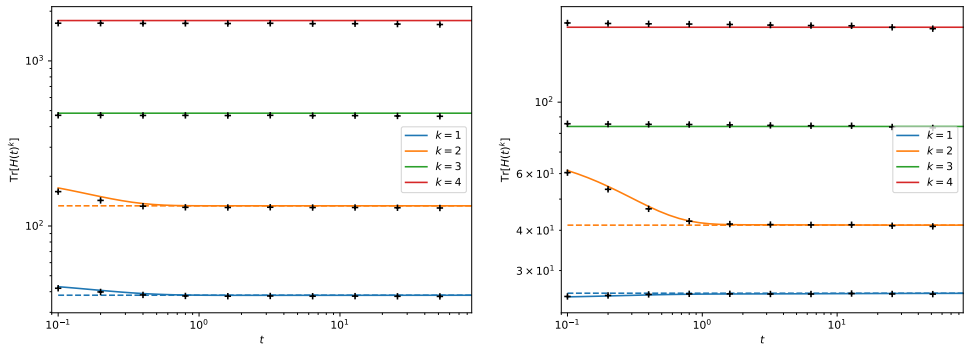


Figure 3.3.1: Comparison of the theoretical prediction of Corollary 1 for the expectation of the first 4 moments (colored lines) to the empirical average over 250 trials (black crosses) for a rectangular network with two hidden layers of finite widths $n_1 = n_2 = 5000$ ($L = 3$) with the smooth ReLU (left) and the normalized smooth ReLU (right), for the MSE loss on scaled down 14x14 MNIST with $N = 256$. Only the first two moments are affected by S at the beginning of training.

In expectation we have:

$$\mathbb{E}\left[\mathrm{Tr}\left(S(t)\right)\right] \to -\frac{1}{N}Tr\left(\left(Id_{Nn_L} + e^{-t\tilde{\Theta}}\right)\tilde{\Theta}^{-1}\tilde{\Lambda}^T e^{-t\tilde{\Theta}}\left(\tilde{\Sigma} + Y^* Y^{*T}\right)\right) + \frac{1}{N}Tr\left(e^{-t\tilde{\Theta}}\tilde{\Phi}^T\right)$$
$$\mathbb{E}\left[\mathrm{Tr}\left(S(t)^2\right)\right] \to \frac{1}{N^2}Tr\left(e^{-t\tilde{\Theta}}\tilde{\Upsilon}e^{-t\tilde{\Theta}}\left(\tilde{\Sigma} + Y^* Y^{*T}\right)\right).$$

Proof. The moments of I are constant because $\mathcal{H}C = \frac{1}{N}Id_{Nn_L}$ is constant. For the moments of S, we first solve the differential equation for $Y(t)$:

$$Y(t) = Y^* - e^{-t\tilde{\Theta}}(Y^* - Y(0)).$$

Noting $Y(t) - Y(0) = -\tilde{\Theta}\int_0^t \nabla C(s)ds$, we have

$$\begin{aligned} G(t) &= G(0) - \tilde{\Lambda}\int_0^t \nabla C(s)ds \\ &= G(0) + \tilde{\Lambda}\tilde{\Theta}^{-1}(Y(t) - Y(0)) \\ &= G(0) + \tilde{\Lambda}\tilde{\Theta}^{-1}\left(Id_{Nn_L} + e^{-t\tilde{\Theta}}\right)(Y^* - Y(0)) \end{aligned}$$

The expectation of the first moment of S then follows. □

Mutual Orthogonality of I and S

A first key ingredient to prove Theorem 3.1 is the asymptotic mutual orthogonality of the matrices I and S

Proposition (Proposition C.4 in Appendix C.4)**.** *For any loss C with BGOSS and $\sigma \in C_b^4(\mathbb{R})$, we have uniformly over $[0,T]$*

$$\lim_{n_{L-1}\to\infty} \cdots \lim_{n_1\to\infty} \|IS\|_F = 0.$$

As a consequence $\lim_{n_{L-1}\to\infty} \cdots \lim_{n_1\to\infty} \mathrm{Tr}\left([I+S]^k\right) - \left[\mathrm{Tr}\left(I^k\right) + \mathrm{Tr}\left(S^k\right)\right] = 0.$

Remark 3.1. If two matrices A and B are mutualy orthogonal (i.e. $AB = 0$) the range of A is contained in the nullspace of B and vice versa. The non-zero eigenvalues of the sum $A + B$ are therefore given by the union of the non-zero eigenvalues of A and B. Furthermore the moments of A and B add up: $\mathrm{Tr}\left([A+B]^k\right) = \mathrm{Tr}\left(A^k\right) + \mathrm{Tr}\left(B^k\right)$. Proposition C.4 shows that this is what happens asymptotically for I and S.

Note that both matrices I and S have large nullspaces: indeed assuming a constant width $w = n_1 = ... = n_{L-1}$, we have $Rank(I) \leq Nn_L$ and $Rank(S) \leq 2(L-1)wNn_L$ (see Appendix C.3), while the number of parameters P scales as w^2 (when $L > 2$).

Figure 3.3.2 illustrates the mutual orthogonality of I and S. All numerical experiments are done for rectangular networks (when the width of the hidden layers are equal) and agree well with our predictions obtained in the sequential limit.

Mean-field Limit

For a rectangular network with width w, if the output of the network is divided by $\sqrt{w}$ and the learning rate is multiplied by w (to keep similar dynamics at initialization), the training dynamics changes and the NTK varies during training when w goes to infinity. The new parametrization of the output changes the scaling of the two matrices:

$$\mathcal{H}\left[C\left(\frac{1}{\sqrt{w}}Y^{(L)}\right)\right] = \frac{1}{w}\left(\mathcal{D}Y^{(L)}\right)^T \mathcal{H}C\mathcal{D}Y^{(L)} + \frac{1}{\sqrt{w}}\nabla C \cdot \mathcal{H}Y^{(L)} = \frac{1}{w}I + \frac{1}{\sqrt{w}}S.$$

The scaling of the learning rate essentially multiplies the whole Hessian by w. In this setting, the matrix I is left unchanged while the matrix S is multiplied by $\sqrt{w}$ (the k-th moment of S is hence multiplied by $w^{k/2}$). In particular, the two moments of the Hessian are dominated by the moments of S, and the higher moments of S (and the operator norm of S) should not vanish. This suggests that the active regime may be characterised by the fact that $\|S\|_F \gg \|I\|_F$. Under the conjecture that Theorem 3.1 holds for the infinite-width limit of rectangular networks, the asymptotic of the two first moments of H is given by:

$$1/\sqrt{w}\mathrm{Tr}\left(H\right) \to \mathcal{N}(0, \nabla C^T \tilde{\Xi} \nabla C)$$
$$1/w\mathrm{Tr}\left(H^2\right) \to \nabla C^T \tilde{\Upsilon} \nabla C,$$

where for the MSE loss we have $\nabla C = -Y^*$.

The matrix S

The matrix $S = \nabla C \cdot \mathcal{H}Y^{(L)}$ is best understood as a perturbation to I, which vanishes as the network converges because $\nabla C \to 0$. To calculate its moments, we note that

$$\mathrm{Tr}\left(\nabla C \cdot \mathcal{H}Y^{(L)}\right) = \left(\sum_{p=1}^{P} \partial^2_{\theta_p^2} Y\right)^T \nabla C = G^T \nabla C,$$

where the vector $G = \sum_{k=1}^{P} \partial_{\theta_p^2}^2 Y \in \mathbb{R}^{Nn_L}$ is the evaluation of the function $g_\theta(x) = \sum_{k=1}^{P} \partial_{\theta_p^2}^2 f_\theta(x)$ on the training set.

For the second moment we have

$$\mathrm{Tr}\left(\left(\nabla C \cdot \mathcal{H} Y^{(L)}\right)^2\right) = \nabla C^T \left(\sum_{p,p'=1}^{P} \partial_{\theta_p\theta_{p'}}^2 Y \left(\partial_{\theta_p\theta_{p'}}^2 Y\right)^T\right) \nabla C = \nabla C^T \tilde{\Upsilon} \nabla C$$

for $\tilde{\Upsilon}$ the Gram matrix of the kernel $\Upsilon^{(L)}(x,y) = \sum_{p,p'=1}^{P} \partial_{\theta_p\theta_{p'}}^2 f_\theta(x) \left(\partial_{\theta_p\theta_{p'}}^2 f_\theta(y)\right)^T$.

The following proposition desribes the limit of the function g_θ and the kernel $\Upsilon^{(L)}$ and the vanishing of the higher moments:

Proposition (Proposition C.3 in Appendix C.3)**.** *For any loss C with BGOSS and $\sigma \in C_b^4(\mathbb{R})$, the first two moments of S take the form*

$$\mathrm{Tr}\left(S(t)\right) = G(t)^T \nabla C(t)$$
$$\mathrm{Tr}\left(S(t)^2\right) = \nabla C(t)^T \tilde{\Upsilon}(t) \nabla C(t)$$

- At initialization, g_θ and f_θ converge to a (centered) Gaussian pair with covariances

$$\mathbb{E}[g_{\theta,k}(x) g_{\theta,k'}(x')] = \delta_{kk'} \Xi_\infty^{(L)}(x,x')$$
$$\mathbb{E}[g_{\theta,k}(x) f_{\theta,k'}(x')] = \delta_{kk'} \Phi_\infty^{(L)}(x,x')$$
$$\mathbb{E}[f_{\theta,k}(x) f_{\theta,k'}(x')] = \delta_{kk'} \Sigma_\infty^{(L)}(x,x')$$

and during training g_θ evolves according to

$$\partial_t g_{\theta,k}(x) = \sum_{i=1}^{N} \Lambda_\infty^{(L)}(x,x_i) \partial_{ik} C(Y(t)).$$

- Uniformly over any interval $[0,T]$, the kernel $\Upsilon^{(L)}$ has a deterministic and fixed limit

$$\lim_{n_{L-1}\to\infty} \cdots \lim_{n_1\to\infty} \Upsilon_{kk'}^{(L)}(x,x') = \delta_{kk'} \Upsilon_\infty^{(L)}(x,x')$$

with limiting kernel:

$$\Upsilon_\infty^{(L)}(x,x') = \sum_{\ell=1}^{L-1} \left(\Theta_\infty^{(\ell)}(x,x')^2 \ddot{\Sigma}_\infty^{(\ell)}(x,x') + 2\Theta_\infty^{(\ell)}(x,x') \dot{\Sigma}_\infty^{(\ell)}(x,x')\right) \dot{\Sigma}_\infty^{(\ell+1)}(x,x') \cdots \dot{\Sigma}_\infty^{(L-1)}(x,x').$$

- The higher moment $k > 2$ vanish: $\lim_{n_{L-1}\to\infty} \cdots \lim_{n_1\to\infty} \mathrm{Tr}\left(S^k\right) = 0$.

This result has a number of consequences for infinitely wide networks:

1. At initialization, the matrix S has a finite Frobenius norm $\|S\|_F^2 = \mathrm{Tr}\left(S^2\right) = \nabla C^T \tilde{\Upsilon} \nabla C$, because Υ converges to a fixed limit. As the network converges, the derivative of the cost goes to zero $\nabla C(t) \to 0$ and so does the Frobenius norm of S.

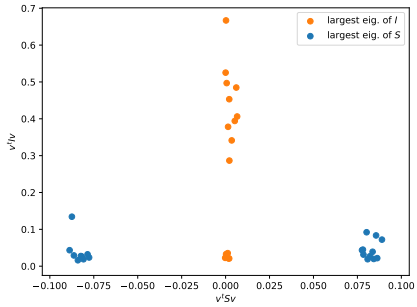


Figure 3.3.2: Illustration of the mutual orthogonality of I and S. For the 20 first eigenvectors of I (blue) and S (orange), we plot the Rayleigh quotients $v^T I v$ and $v^T S v$ (with $L = 3$, $n_1 = n_2 = 1000$ and the normalized ReLU on 14x14 MNIST with $N = 256$). We see that the directions where I is large are directions where S is small and vice versa.

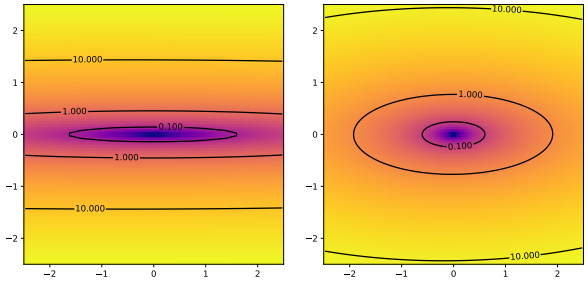


Figure 3.3.3: Plot of the loss surface around a global minimum along the first (along the y coordinate) and fourth (x coordinate) eigenvectors of I. The network has $L = 4$, width $n_1 = n_2 = n_3 = 1000$ for the smooth ReLU (left) and the normalized smooth ReLU (right). The data is uniform on the unit disk. Normalizing the non-linearity greatly reduces the narrow valley structure of the loss thus speeding up training.

2. In contrast the operator norm of S vanishes already at initialization (because for all even k, we have $\|S\|_{op} \leq \sqrt[k]{\mathrm{Tr}\left(S^k\right)} \to 0$). At initialization, the vanishing of S in operator norm but not in Frobenius norm can be explained by the matrix S having a growing number of eigenvalues of shrinking intensity as the width grows.

3. When it comes to the first moment of S, Proposition C.3 shows that the spectrum of S is in general not symmetric. For the MSE loss the expectation of the first moment at initialization is
$$\mathbb{E}\left[\mathrm{Tr}(S)\right] = \mathbb{E}\left[(Y - Y^*)^T G\right] = \mathbb{E}\left[Y^T G\right] - \left(Y^*\right)^T \mathbb{E}\left[G\right] = \mathrm{Tr}\left(\tilde{\Phi}\right) - 0$$
which may be positive or negative depending on the choice of nonlinearity: with a smooth ReLU, it is positive, while for the arc-tangent or the normalized smooth ReLU, it can be negative (see Figure 3.3.1).
This is in contrast to the result obtained in [171, 68] for the shallow ReLU networks, taking the second derivative of the ReLU to be zero. Under this assumption the spectrum of S is symmetric: if the eigenvalues are ordered from lowest to highest, $\lambda_i = -\lambda_{P-i}$ and $\mathrm{Tr}(S) = 0$.

These observations suggest that S has little influence on the shape of the surface, especially towards the end of training, the matrix I however has an interesting structure.

The matrix I

At a global minimizer θ^*, the spectrum of I describes how the loss behaves around θ^*. Along the eigenvectors of the biggest eigenvalues of I, the loss increases rapidely, while small eigenvalues correspond to flat directions. Numerically, it has been observed that the matrix I features a few

dominating eigenvalues and a bulk of small eigenvalues [189, 190, 82, 167]. This leads to a narrow valley structure of the loss around a minimum: the biggest eigenvalues are the 'cliffs' of the valley, i.e. the directions along which the loss grows fastest, while the small eigenvalues form the 'flat directions'or the bottom of the valley.

Note that the rank of I is bounded by Nn_L and in the overparametrized regime, when $Nn_L < P$, the matrix I will have a large nullspace, these are directions along which the value of the function on the training set does not change. Note that in the overparametrized regime, global minima are not isolated: they lie in a manifold of dimension at least $P - Nn_L$ and the nullspace of I is tangent to this solution manifold.

The matrix I is closely related to the NTK Gram matrix:

$$\tilde{\Theta} = \mathcal{D}Y^{(L)} \left(\mathcal{D}Y^{(L)}\right)^T \text{ and } I = \left(\mathcal{D}Y^{(L)}\right)^T \mathcal{H}C\mathcal{D}Y^{(L)}.$$

As a result, the limiting spectrum of the matrix I can be directly obtained from the NTK[2]

Proposition 3.1. *For any loss C with BGOSS and $\sigma \in C_b^4(\mathbb{R})$, uniformly over any interval $[0, T]$, the moments $\mathrm{Tr}\left(I^k\right)$ converge to the following limit (with the convention that $i_{k+1} = i_1$):*

$$\lim_{n_{L-1}\to\infty} \cdots \lim_{n_1\to\infty} \mathrm{Tr}\left(I^k\right) = \mathrm{Tr}\left(\left(\mathcal{H}C(Y_t)\tilde{\Theta}\right)^k\right) = \frac{1}{N^k} \sum_{i_1,\ldots,i_k=1}^{N} \prod_{m=1}^{k} c''_{i_m}(f_{\theta(t)}(x_{i_m}))\Theta_\infty^{(L)}(x_{i_m}, x_{i_{m+1}})$$

Proof. It follows from $\mathrm{Tr}\left(I^k\right) = \mathrm{Tr}\left(\left(\left(\mathcal{D}Y^{(L)}\right)^T \mathcal{H}C\mathcal{D}Y^{(L)}\right)^k\right) = \mathrm{Tr}\left(\left(\mathcal{H}C\tilde{\Theta}\right)^k\right)$ and the asymptotic of the NTK [105]. □

Mean-Square Error

When the loss is the MSE, $\mathcal{H}C$ is equal to $\frac{1}{N} Id_{Nn_L}$. As a result, $\tilde{\Theta}$ and I have the same non-zero eigenvalues up to a scaling of $1/N$. Because the NTK is assymptotically fixed, the spectrum of I is also fixed in the limit.

The eigenvectors of the NTK Gram matrix are the kernel principal components of the data. The biggest principal components are the directions in function space which are most favorised by the NTK. This gives a functional interpretation of the narrow valley structure in DNNs: the cliffs of the valley are the biggest principal components, while the flat directions are the smallest components.

Remark 3.2. As the depth L of the network increases, one can observe two regimes [173, 104]: Order/Freeze where the NTK converges to a constant and Chaos where the NTK converges to a Kronecker delta. In the Order/Freeze the $Nn_L \times Nn_L$ Gram matrix approaches a block diagonal matrix with n_L constant blocks, and as a result n_L eigenvalues of I dominate the other ones, corresponding to constant directions along each outputs (this is in line with the observations of [167]). This leads to a narrow valley for the loss and slows down training. In contrast, in the Chaos regime, the NTK Gram matrix approaches a scaled identity matrix, and the spectrum of I should hence concentrate around a positive value, hence speeding up training. Figure 3.3.3 illustrates this phenomenon: with the smooth ReLU we observe a narrow valley, while with the normalized smooth ReLU (which lies in the Chaos according to [104]) the narrowness of the loss is reduced. A similar phenomenon may explain why normalization helps smoothing the loss surface and speed up training [193, 73].

[2]This result was already obtained in [112], but without identifying the NTK explicitely and only at initialization.

Cross-Entropy Loss

For a binary cross-entropy loss with labels $Y^* \in \{-1, +1\}^N$

$$C(Y) = \frac{1}{N} \sum_{i=1}^{N} log\left(1 + e^{-Y_i^* Y_i}\right),$$

$\mathcal{H}C$ is a diagonal matrix whose entries depend on Y (but not on Y^*):

$$\mathcal{H}_{ii} C(Y) = \frac{1}{N} \frac{1}{1 + e^{-Y_i} + e^{Y_i}}.$$

The eigenvectors of I then correspond to the weighted kernel principal component of the data. The positive weights $\frac{1}{1+e^{-Y_i}+e^{Y_i}}$ approach $1/3$ as Y_i goes to 0, i.e. when it is close to the decision boundary from one class to the other, and as $Y_i \to \pm\infty$ the weight go to zero. The weights evolve in time through Y_i, the spectrum of I is therefore not asymptotically fixed as in the MSE case, but the functional interpretation of the spectrum in terms of the kernel principal components remains.

3.4 Conclusion

We have given an explicit formula for the limiting moments of the Hessian of DNNs throughout training. We have used the common decomposition of the Hessian in two terms I and S and have shown that the two terms are asymptotically mutually orthogonal, such that they can be studied separately.

The matrix S vanishes in Frobenius norm as the network converges and has vanishing operator norm throughout training. The matrix I is arguably the most important as it describes the narrow valley structure of the loss around a global minimum. The eigendecomposition of I is related to the (weighted) kernel principal components of the data w.r.t. the NTK.

Chapter 4

Kernel Alignment Ridge Estimator: Risk Prediction From Training Data

Abstract

We study the risk (i.e. generalization error) of Kernel Ridge Regression (KRR) for a kernel K with ridge $\lambda > 0$ and i.i.d. observations. For this, we introduce two objects: the Signal Capture Threshold (SCT) and the Kernel Alignment Risk Estimator (KARE). The SCT $\vartheta_{K,\lambda}$ is a function of the data distribution: it can be used to identify the components of the data that the KRR predictor captures, and to approximate the (expected) KRR risk. This then leads to a KRR risk approximation by the KARE $\rho_{K,\lambda}$, an explicit function of the training data, agnostic of the true data distribution. We phrase the regression problem in a functional setting. The key results then follow from a finite-size analysis of the Stieltjes transform of general Wishart random matrices. Under a natural universality assumption (that the KRR moments depend asymptotically on the first two moments of the observations) we capture the mean and variance of the KRR predictor. We numerically investigate our findings on the Higgs and MNIST datasets for various classical kernels: the KARE gives an excellent approximation of the risk, thus supporting our universality assumption. Using the KARE, one can compare choices of Kernels and hyperparameters directly from the training set. The KARE thus provides a promising data-dependent procedure to select Kernels that generalize well.

4.1 Introduction

Kernel Ridge Regression (KRR) is a widely used statistical method to learn a function from its values on a training set [198, 200]. It is a non-parametric generalization of linear regression to infinite-dimensional feature spaces. Given a positive-definite kernel function K and (noisy) observations y^ϵ of a true function f^* at a list of points $X = x_1, \dots, x_N$, the λ-KRR estimator $\hat{f}^\epsilon_\lambda$ of f^* is defined by

$$\hat{f}^\epsilon_\lambda(x) = \frac{1}{N} K(x, X) \left(\frac{1}{N} K(X, X) + \lambda I_N \right)^{-1} y^\epsilon,$$

where $K(x, X) = (K(x, x_i))_{i=1,..,N} \in \mathbb{R}^N$ and $K(X, X) = (K(x_i, x_j))_{i,j=1,..,N} \in \mathbb{R}^{N \times N}$.

Despite decades of intense mathematical progress, the rigorous analysis of the generalization of kernel methods remains a very active and challenging area of research. In recent years, many new

kernels have been introduced for both regression and classification tasks; notably, a large number of kernels have been discovered in the context of deep learning, in particular through the so-called Scattering Transform [148], and in close connection with deep neural networks [37, 105], yielding ever-improving performance for various practical tasks [6, 51, 136, 199]. Currently, theoretical tools to select the relevant kernel for a given task, i.e. to minimize the generalization error, are however lacking.

While a number of bounds for the risk of Linear Ridge Regression (LRR) or KRR [29, 72, 212, 150] exist, most focus on the rate of convergence of the risk: these estimates typically involve constant factors which are difficult to control in practice. Recently, a number of more precise estimates have been given [145, 48, 152, 144, 25]; however, these estimates typically require a priori knowledge of the data distribution. It remains a challenge to have estimates based on the training data alone, enabling one to make informed decisions on the choices of the ridge and of the kernel.

Contributions

We consider a generalization of the KRR predictor $\hat{f}^{\epsilon}_{\lambda}$: one tries to reconstruct a true function f^* in a space of continuous functions $\mathcal{C}$ from noisy observations y^{ϵ} of the form $(o_1(f^*) + \epsilon e_1, \dots, o_N(f^*) + \epsilon e_N)$, where the observations o_i are i.i.d. linear forms $\mathcal{C} \to \mathbb{R}$ sampled from a distribution π, ϵ is the level of noise, and the $e_1, \dots, e_N$ are centered of unit variance. We work under the universality assumption that, for large N, only the first two moments of π determine the behavior of the first two moments of $\hat{f}^{\epsilon}_{\lambda}$. We obtain the following results:

1. We introduce the Signal Capture Threshold (SCT) $\vartheta(\lambda, N, K, \pi)$, which is determined by the ridge λ, the size of the training set N, the kernel K, and the observations distribution π (more precisely, the dependence on π is only through its first two moments). We give approximations for the expectation and variance of the KRR predictor in terms of the SCT.

2. Decomposing f^* along the kernel principal components of the data distribution, we observe that in expectation, the predictor $\hat{f}^{\epsilon}_{\lambda}$ captures only the signal along the principal components with eigenvalues larger than the SCT. If N increases or λ decreases, the SCT ϑ shrinks, allowing the predictor to capture more signal. At the same time, the variance of $\hat{f}^{\epsilon}_{\lambda}$ scales with the derivative $\partial_{\lambda}\vartheta$, which grows as $\lambda \to 0$, supporting the classical bias-variance tradeoff picture [71].

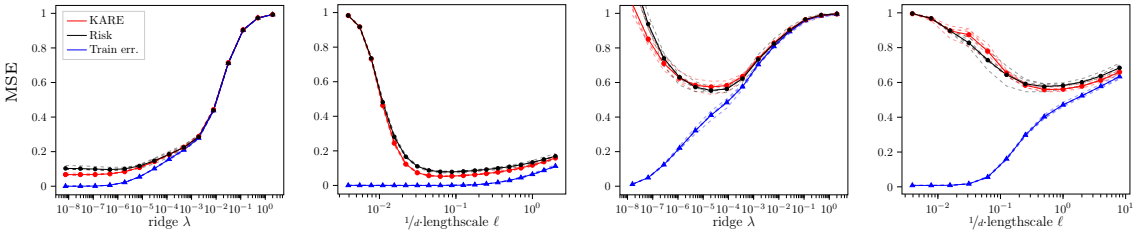
3. We give an explicit approximation for the expected MSE risk $R^{\epsilon}(\hat{f}^{\epsilon}_{\lambda})$ and empirical MSE risk $\hat{R}^{\epsilon}(\hat{f}^{\epsilon}_{\lambda})$ for an arbitrary continuous true function f^*. We find that, surprisingly, the expected risk and expected empirical risk are approximately related by

$$\mathbb{E}[R^{\epsilon}(\hat{f}^{\epsilon}_{\lambda})] \approx \frac{\vartheta(\lambda)^2}{\lambda^2}\mathbb{E}[\hat{R}^{\epsilon}(\hat{f}^{\epsilon}_{\lambda})].$$

4. We introduce the Kernel Alignment Risk Estimator (KARE) as the ratio ρ defined by

$$\rho(\lambda, N, y^{\epsilon}, G) = \frac{\frac{1}{N}\left(y^{\epsilon}\right)^T \left(\frac{1}{N}G + \lambda I_N\right)^{-2} y^{\epsilon}}{\left(\frac{1}{N}\mathrm{Tr}\left[\left(\frac{1}{N}G + \lambda I_N\right)^{-1}\right]\right)^2},$$

   where G is the Gram matrix of K on the observations. We show that the KARE approximates the expected risk; unlike the SCT, it is agnostic of the true data distribution. This result



(a) MNIST, $\ell = d$ (b) MNIST, $\lambda = 10^{-5}$ (c) Higgs, $\ell = d$ (d) Higgs, $\lambda = 10^{-4}$

Figure 4.1.1: Comparison between the KRR risk and the KARE for various choices of normalized lengthscale ℓ/d and ridge λ on the MNIST dataset (restricted to the digits 7 and 9, labeled by 1 and -1 respectively, $N = 2000$) and on the Higgs dataset (classes 'b' and 's', labeled by -1 and 1, $N = 1000$) with the RBF Kernel $K(x, x') = \exp(-\|x-x'\|_2^2/\ell)$ (see the Appendix for experiments with the Laplacian and ℓ_1-norm kernels). KRR predictor risks, and KARE curves (shown as dashed lines, 5 samples) concentrate around their respective averages (solid lines).

follows from the fact that $\vartheta(\lambda) \approx 1/m_G(-\lambda)$, where $m_G(z) = \mathrm{Tr}\left[(\frac{1}{N}G - zI_N)^{-1}\right]$ is the Stieltjes Transform of the Gram matrix.

5. Empirically, we find that the KARE predicts the risk on the Higgs and MNIST datasets. We see empirically that our results extend extremely well beyond the Gaussian observation setting, thus supporting our universality assumption (see Figure 4.1.1).

Our proofs (see the Appendix) rely on a generalized and refined version of the finite-size analysis of [102] of generalized Wishart matrices, obtaining sharper bounds and generalizing the results to operators. Our analysis relies in particular on the complex Stieltjes transform $m_G(z)$, evaluated at $z = -\lambda$, and on fixed-point arguments.

Related Works

The theoretical analysis of the risk of KRR has seen tremendous developments in the recent years. In particular, a number of upper and lower bounds for kernel risk have been obtained [29, 212, 150] in various settings: notably, convergence rates (i.e. without control of the constant factors) are obtained in general settings. This allows one to abstract away a number of details about the kernels (e.g. the lengthscale), which don't influence the asymptotic rates. However, this does not give access to the risk at finite data size (crucial to pick e.g. the correct lengthscale or the NTK depth [105]).

A number of recent results have given precise descriptions of the risk for ridge regression [48, 144], for random features [152, 102], and in relation to neural networks [145, 25]. These results rely on the analysis of the asymptotic spectrum of general Wishart random matrices, in particular through the Stieltjes transform [203, 11]. The limiting Stieltjes transform can be recovered from the formula for the product of freely independent matrices [65]. To extend these asymptotic results to finite-size settings, we generalize and adapt the results of [102].

While these techniques have given simple formulae for the KRR predictor expectation, approximating its variance has remained more challenging. For this reason the description of the expected risk in [145] is stated as a conjecture. In [144] only the bias component of the risk is approximated.

In [48] the expected risk is given only for random true functions (in a Bayesian setting) with a specific covariance. In [25], the expected risk follows from a heuristic spectral analysis combining a PDE approximation and replica tricks. In this paper, we approximate the variance of the predictor along the principal components, giving an approximation of the risk for any continuous true function.

The SCT is related to a number of objects from previous works, such as the effective dimension of [238, 29], the companion Stieltjes transform of [48, 144], and particularly the effective ridge of [102]. The SCT can actually be viewed as a direct translation to the KRR risk setting of [102].

Outline

In Section 8.2, we first introduce the Kernel Ridge Regression (KRR) predictor in functional space (Section 4.2) and formulate its train error and risk for random observations (Section 4.2).

The rest of the paper is then devoted to obtaining approximations for the KRR risk. In Section 4.3,the Signal Capture Threshold (SCT) is introduced and used to study the mean and variance of the KRR predictor (Sections 4.3 and 4.3). An approximation of the SCT in terms of the observed data is then given (Section 4.3). In Section 4.4, the expected risk and the expected empirical risk are approximated in terms of the SCT and its derivative w.r.t. the ridge λ. The SCT approximation of Section 4.3, together with the estimates of Section 4.4, leads to an approximation of the KRR risk by the Kernel Alignment Risk Estimator (KARE).

4.2 Setup

Given a compact $\Omega \subset \mathbb{R}^d$, let $\mathcal{C}$ denote the space of continuous $f : \Omega \to \mathbb{R}$, endowed with the supremum norm $\|f\|_\infty = \sup_{x\in\Omega} |f(x)|$. In the classical regression setting, we want to reconstruct a true function $f^* \in \mathcal{C}$ from its values on a training set $x_1, \dots, x_N$, i.e. from the noisy labels $y^\epsilon = (f^*(x_1) + \epsilon e_1, \dots, f^*(x_N) + \epsilon e_N)^T$ for some i.i.d. centered noise $e_1, \dots, e_N$ of unit variance and noise level $\epsilon \geq 0$.

In this paper, the observed values (without noise) of the true function f^* consist in observations $o_1, \dots, o_N \in \mathcal{C}^*$, where $\mathcal{C}^*$ is the dual space, i.e. the space of bounded linear functionals $\mathcal{C} \to \mathbb{R}$. We thus represent the training set of N observations $o_1, \dots, o_N$ by the *sampling operator* $\mathcal{O} : \mathcal{C} \to \mathbb{R}^N$ which maps a function $f \in \mathcal{C}$ to the vector of observations $\mathcal{O}(f) = (o_1(f), \dots, o_N(f))^T$.

The classical setting corresponds to the case where the observations are evaluations of f^* at points $x_1, \dots, x_N \in \Omega$, i.e. $o_i(f^*) = f^*(x_i)$ for $i = 1, \dots, N$. In time series analysis (when $\Omega \subset \mathbb{R}$), the observations can be the averages $o_i(f^*) = \frac{1}{b_i - a_i} \int_{a_i}^{b_i} f^*(t)dt$ over time intervals $[a_i, b_i] \subset \mathbb{R}$.

Kernel Ridge Regression Predictor

The regression problem is now stated as follows: given noisy observations $y_i^\epsilon = o_i(f^*) + \epsilon e_i$ with i.i.d. centered noises $e_1, \dots, e_N$ of unit variance, how can one reconstruct f^*?

Definition 1. *Consider a continuous positive kernel $K : \Omega \times \Omega \to \mathbb{R}$ and a ridge parameter $\lambda > 0$. The Kernel Ridge Regression (KRR) predictor with ridge λ is the function $\hat{f}_\lambda^\epsilon : \Omega \to \mathbb{R}$*

$$\hat{f}_\lambda^\epsilon = \frac{1}{N} K\mathcal{O}^T \left(\frac{1}{N}\mathcal{O}K\mathcal{O}^T + \lambda I_N\right)^{-1} y^\epsilon$$

where $\mathcal{O}^T : \mathbb{R}^N \to \mathcal{C}^*$ is the adjoint of $\mathcal{O}$ defined by $(\mathcal{O}^T y)(f) = y^T \mathcal{O}(f)$ and where we view K as a map $\mathcal{C}^* \to \mathcal{C}$ with $(K\mu)(x) = \mu(K(x, \cdot))$.

Remark 4.1. The KRR predictor arises naturally in the following setup: assuming a (centered) Gaussian Bayesian prior on the true function with covariance operator K and noise amplitude ϵ, the expected posterior, for observed labels y^ϵ is given by $\hat{f}^\epsilon_\lambda$ for $\lambda = \epsilon^2$.

We call the $N \times N$ matrix $G = \mathcal{O}K\mathcal{O}^T$ the *Gram matrix*: in the classical setting, when the observations are $o_i = \delta_{x_i}$ (with $\delta_x(f) = f(x)$), G is the usual Gram matrix, i.e. $G_{ij} = K(x_i, x_j)$.

Training Error and Risk

We consider the least-squares error (MSE loss) of the KRR predictor, taking into account randomness of: (1) the test point, random observation o to which is added a noise ϵe (2) the training data, made of N observations o_i plus noises $\epsilon e_i \sim \nu$, where $o, o_1, \dots, o_n \sim \pi$ and $e, e_1, \dots, e_N$ are i.i.d. The expected risk of the KRR predictor is thus taken w.r.t. the test and training observations and their noises. Unless otherwise specified, the expectations are taken w.r.t. all these sources of randomness.

For (fixed) observations $o_1, \dots, o_N$, the *empirical risk* or *training error* of the KRR predictor $\hat{f}^\epsilon_\lambda$ is

$$\hat{R}^\epsilon(\hat{f}^\epsilon_\lambda) = \frac{1}{N}\sum_{i=1}^N (o_i(\hat{f}^\epsilon_\lambda) - y^\epsilon_i)^2 = \frac{1}{N}\left\|\mathcal{O}(\hat{f}^\epsilon_\lambda) - y^\epsilon\right\|^2.$$

For a random observation o sampled from π and a noise ϵe (where $e \sim \nu$ is centered of unit variance as before), the *risk* $R^\epsilon(\hat{f}^\epsilon_\lambda)$ of the KRR predictor $\hat{f}^\epsilon_\lambda$ is defined by

$$R^\epsilon(\hat{f}^\epsilon_\lambda) = \mathbb{E}_{o\sim\pi, e\sim\nu}\left[(o(f^*) + \epsilon e - o(\hat{f}^\epsilon_\lambda))^2\right].$$

Describing the observation variance by the bilinear form $\langle f, g\rangle_S = \mathbb{E}_{o\sim\pi}[o(f)o(g)]$ and the related semi-norm $\|f\|_S = \langle f, f\rangle_S^{1/2}$, the risk can be rewritten as $R^\epsilon(\hat{f}^\epsilon_\lambda) = \|\hat{f}^\epsilon_\lambda - f^*\|_S^2 + \epsilon^2$.

From now on, we will assume that $\langle\cdot,\cdot\rangle_S$ is a scalar product; note that in the classical setting, when o is the evaluation of f^* at a point $x \in \Omega$ with $x \sim \sigma$, the S-norm is given by $\|f\|_S^2 = \int_\Omega f(x)^2\sigma(dx)$.

The following three operators $\mathcal{C} \to \mathcal{C}$ are central to our analysis:

Definition 2. *The KRR reconstruction operator $A_\lambda : \mathcal{C} \to \mathcal{C}$, the KRR Integral Operator $T_K : \mathcal{C} \to \mathcal{C}$, and its empirical version $T_K^N : \mathcal{C} \to \mathcal{C}$ are defined by*

$$\begin{aligned} A_\lambda &= \frac{1}{N}K\mathcal{O}^T(\frac{1}{N}\mathcal{O}K\mathcal{O}^T + \lambda I_N)^{-1}\mathcal{O}, \\ (T_K f)(x) &= \langle f, K(x,\cdot)\rangle_S = \mathbb{E}_{o\sim\pi}\left[o(f)o(K(x,\cdot))\right], \\ (T_K^N f)(x) &= \frac{1}{N}K\mathcal{O}^T\mathcal{O}f(x) = \frac{1}{N}\sum_{i=1}^N o_i(f)o_i(K(x,\cdot)). \end{aligned}$$

Note that in the noiseless regime (i.e. when $\epsilon = 0$), we have $\hat{f}^\epsilon_\lambda\big|_{\epsilon=0} = A_\lambda f^*$. Also note that A_λ and T_K^N are random operators, as they depend on the random observations. The operator T_K

is the natural generalization to our framework of the integration operator $f \mapsto \int K(x,\cdot)f(x)\sigma(dx)$, which is defined with random observations δ_x with $x \sim \sigma$ in the classical setting.

The reconstruction and empirical integral operators are linked by $A_\lambda = T_K^N(T_K^N + \lambda I_{\mathcal{C}})^{-1}$, which follows from the identity $\left(\frac{1}{N}\mathcal{O}K\mathcal{O}^T + \lambda I_N\right)^{-1}\mathcal{O} = \mathcal{O}\left(\frac{1}{N}K\mathcal{O}^T\mathcal{O} + \lambda I_{\mathcal{C}}\right)^{-1}$. As $N \to \infty$, we have that $T_K^N \to T_K$, and it follows that

$$A_\lambda \to \tilde{A}_\lambda := T_K(T_K + \lambda I_{\mathcal{C}})^{-1}. \tag{4.2.1}$$

Eigendecomposition of the Kernel

We will assume that the kernel K can be diagonalized by a countable family of eigenfunctions $(f^{(k)})_{k\in\mathbb{N}}$ in $\mathcal{C}$ with eigenvalues $(d_k)_{k\in\mathbb{N}}$, orthonormal with respect to the scalar product $\langle\cdot,\cdot\rangle_S$, such that we have (with uniform convergence):

$$K(x,x') = \sum_{k=1}^{\infty} d_k f^{(k)}(x) f^{(k)}(x').$$

The functions $f^{(k)}$ are also eigenfunctions of T_K: we have $T_K f^{(k)} = d_k f^{(k)}$. We will also assume that $\mathrm{Tr}\left[T_K\right] = \sum_{k=1}^{\infty}\left\langle f^{(k)}, T_K(f^{(k)})\right\rangle_S = \sum_{k=1}^{\infty} d_k$ is finite. Note that in the classical setting K can be diagonalized as above (by Mercer's theorem), and $\mathrm{Tr}\left[T_K\right] = \mathbb{E}_{x\sim\sigma}\left[K(x,x)\right]$ is finite. Computing the eigendecomposition of T_K is difficult for general kernels and data distributions, but explicit formulas exist for special cases, such as for the RBF kernel and isotropic Gaussian inputs as described in Section 1.5 of the Appendix.

Gaussianity Assumption

As seen in Equation (4.2.1) above, $\tilde{A}_\lambda$ only depends on the second moment of π (through $\langle\cdot,\cdot\rangle_S$), suggesting the following assumption, with which we will work in this paper:

Assumption A. *As far as one is concerned with the first two moments of the A_λ operator, for large but finite N, we will assume that the observations $o_1,\ldots,o_N$ are centered Gaussian, i.e. that for any tuple of functions $(f_1,\ldots,f_N)$, the vector $(o_1(f_1),\ldots,o_N(f_N))$ is a mean zero Gaussian vector.*

Though our proofs use this assumption, the ideas in [145, 23] suggest a path to extend them beyond the Gaussian case, where our numerical experiments (see Figure 4.1.1) suggest that our results remain true. See Section 2.1 of the Appendix for a more detailed discussion.

4.3 Predictor Moments and Signal Capture Threshold

A central tool in our analysis of the KRR predictor $\hat{f}_\lambda^\epsilon$ is the Signal Capture Threshold (SCT):

Definition 3. *For $\lambda > 0$, the* Signal Capture Threshold $\vartheta(\lambda) = \vartheta(\lambda, N, K, \pi)$ *is the unique positive solution (see Section 2.2 in the Appendix) to the equation:*

$$\vartheta(\lambda) = \lambda + \frac{\vartheta(\lambda)}{N}\mathrm{Tr}\left[T_K\left(T_K + \vartheta(\lambda)I_{\mathcal{C}}\right)^{-1}\right].$$

In this section, we use $\vartheta(\lambda)$ and the derivative $\partial_\lambda\vartheta(\lambda)$ for the estimation of the mean and variance of the KRR predictor $\hat{f}_\lambda^\epsilon$ upon which the Kernel Alignment Risk Estimator of Section 4.4 is based.

Mean predictor

The expected KRR predictor can be expressed in terms of the expected reconstruction operator A_λ

$$\mathbb{E}[\hat{f}^\epsilon_\lambda] = \mathbb{E}[\frac{1}{N} K\mathcal{O}^T (\frac{1}{N}\mathcal{O}K\mathcal{O}^T + \lambda I_N)^{-1} y^\epsilon] = \mathbb{E}\left[A_\lambda\right] f^*,$$

where we used the fact that $\mathbb{E}_{e_1,\dots,e_N}[y^\epsilon] = \mathcal{O}f^*$.

Theorem 4.1 (Theorem 10 in the Appendix)**.** *The expected reconstruction operator $\mathbb{E}[A_\lambda]$ is approximated by the operator $\tilde{A}_\vartheta = T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1}$ in the sense that for all $f, g \in \mathcal{C}$,*

$$\left|\left\langle f, \left(\mathbb{E}\left[A_\lambda\right] - \tilde{A}_\vartheta\right) g\right\rangle_S\right| \leq \left(\frac{1}{N} + \boldsymbol{P}_0(\frac{\mathrm{Tr}[T_K]}{\lambda N})\right) \left|\left\langle f, \tilde{A}_\vartheta (I_{\mathcal{C}} - \tilde{A}_\vartheta) g\right\rangle_S\right|,$$

for a polynomial $\boldsymbol{P}_0$ with nonnegative coefficients and $\boldsymbol{P}_0(0) = 0$.

Proof. (Sketch; see the Appendix for details) First we show that $\mathbb{E}\left[\left\langle f^{(k)}, A_\lambda f^{(m)}\right\rangle_S\right] = 0$ whenever $m \neq k$, using the invariance of the observations' distribution o_i w.r.t. reflection along a principal component $f^{(k)}$. This implies that $\mathbb{E}\left[A_\lambda\right]$ and $\tilde{A}_\vartheta$ both have the same eigenfunctions $(f^{(k)})_{k\geq 1}$. It thus only remains to show that the eigenvalues of both operators are close: $\mathbb{E}\left[\left\langle f^{(k)}, A_\lambda f^{(k)}\right\rangle_S\right] \approx \frac{d_k}{d_k + \vartheta}$.

The difficulty lies in computing the inverse of $B = \frac{1}{N}\mathcal{O}K\mathcal{O}^T + \lambda I_N$. We use the Sherman-Morrison formula to isolate the contribution along the k-th principal component $f^{(k)}$. Defining the kernel $K_{(k)}(x,y) = \sum_{\ell \neq k} d_\ell f^{(\ell)}(x) f^{(\ell)}(y)$ and the vector $\mathcal{O}_k = \mathcal{O}f^{(k)} \in \mathbb{R}^N$, we obtain

$$B^{-1} = B^{-1}_{(k)} - \frac{1}{N}\frac{d_k}{1 + d_k g_k} B^{-1}_{(k)} \mathcal{O}_k \mathcal{O}_k^T B^{-1}_{(k)}$$

for $B_{(k)} = \frac{1}{N}\mathcal{O}K_{(k)}\mathcal{O}^T + \lambda I_N$ and $g_k = \frac{1}{N}\mathcal{O}_k^T B^{-1}_{(k)} \mathcal{O}_k$. Using the above formula we obtain that

$$\left\langle f^{(k)}, A_\lambda f^{(k)}\right\rangle_S = \frac{1}{N} d_k \mathcal{O}_k^T B^{-1} \mathcal{O}_k = \frac{d_k g_k}{1 + d_k g_k}.$$

Since the vector $\mathcal{O}_k$ is independent of $B_{(k)}$ and has i.i.d. $\mathcal{N}(0, d_k)$ entries, g_k concentrates around $\frac{1}{N}\mathrm{Tr} B^{-1}_{(k)}$ which itself can be approximated by the Stieltjes transform $m(z = -\lambda) = \frac{1}{N}\mathrm{Tr} B^{-1}$ (since $B_{(k)}$ is a rank-one deformation of B). Expanding the trivial equation $\frac{1}{N}\mathrm{Tr}\left[BB^{-1}\right] = 1$, we obtain the relation

$$\frac{1}{N}\sum_{k=1}^{\infty} \frac{d_k g_k}{1 + d_k g_k} + \lambda m(-\lambda) = 1$$

which implies that both the g_k's and the Stieltjes transform $m(-\lambda)$ concentrate around the unique solution $\tilde{m}$ to the equation $\frac{1}{N}\sum_{k=1}^{\infty} \frac{d_k \tilde{m}}{1 + d_k \tilde{m}} + \lambda \tilde{m} = 1$. The SCT is then defined as the reciprocal $\vartheta = 1/\tilde{m}$ and since $g_k \approx \tilde{m}$ we obtain that $\mathbb{E}\left[\left\langle f^{(k)}, A_\lambda f^{(k)}\right\rangle_S\right] = \mathbb{E}\left[\frac{d_k g_k}{1 + d_k g_k}\right] \approx \frac{d_k}{\vartheta + d_k}$ as needed. □

This theorem gives the following motivation for the name SCT: if the true function f^* is an eigenfunction of T_K, i.e. $T_K f^* = \delta f^*$, then $\tilde{A}_\vartheta f^* = \frac{\delta}{\vartheta(\lambda) + \delta} f^*$ and we get:

- if $\delta \gg \vartheta(\lambda)$, then $\frac{\delta}{\vartheta(\lambda)+\delta} \approx 1$ and $\mathbb{E}\left[A_\lambda\right] f^* \approx f^*$, i.e. the function is learned on average,

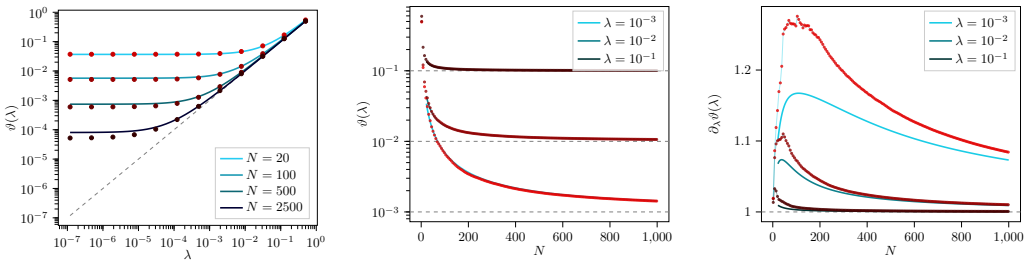


Figure 4.3.1: *Signal Capture Threshold and Derivative.* We consider the RBF Kernel on the standard d-dimensional Gaussian with $\ell = d = 20$. In blue lines, exact formulas for the SCT $\vartheta(\lambda)$ and $\partial_\lambda \vartheta(\lambda)$, computed using the explicit formula for the eigenvalues d_k of the integral operator T_K given in Section 1.5 of the Appendix; in red dots, their approximation with Proposition 4.3.

- if $\delta \ll \vartheta(\lambda)$, then $\frac{\delta}{\vartheta(\lambda)+\delta} \approx 0$ and $\mathbb{E}\left[A_\lambda\right] f^* \approx 0$, i.e. the function is not learned on average.

More generally, if we decompose a true function f^* along the principal components (i.e. eigenfunctions) of T_K, the signal along the k-th principal component $f^{(k)}$ is captured whenever the corresponding eigenvalue $d_k \gg \vartheta(\lambda)$ and lost when $d_k \ll \vartheta(\lambda)$.

Variance of the predictor

We now estimate the variance of $\hat{f}_\lambda^\epsilon$ along each principal component in terms of the SCT $\vartheta(\lambda)$ and its derivative $\partial_\lambda \vartheta(\lambda)$. Along the eigenfunction $f^{(k)}$, the variance is estimated by V_k, where

$$V_k(f^*, \lambda, N, \epsilon) = \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\left\| (I_{\mathcal{C}} - \tilde{A}_\vartheta) f^* \right\|_S^2 + \epsilon^2 + \left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta^2(\lambda)}{(\vartheta(\lambda) + d_k)^2} \right) \frac{d_k^2}{(\vartheta(\lambda) + d_k)^2}.$$

Theorem 4.2 (Theorem 15 in the Appendix)**.** *There is a constant $\boldsymbol{C}_1 > 0$ and a polynomial $\boldsymbol{P}_1$ with nonnegative coefficients and with $\boldsymbol{P}_1(0) = 0$ such that*

$$\left| \mathrm{Var}\left(\left\langle f^{(k)}, \hat{f}_\lambda^\epsilon \right\rangle_S \right) - V_k \right| \leq \left(\frac{\boldsymbol{C}_1}{N} + \boldsymbol{P}_1\left(\frac{\mathrm{Tr}[T_K]}{\lambda N^{\frac{1}{2}}} \right) \right) V_k.$$

As shown in Section 4.4, understanding the variance along the principal components (rather than the covariances between the principal components) is enough to describe the risk.

Behavior of the SCT

The behavior of the SCT can be controlled by the following (agnostic of the exact spectrum of T_K)

Proposition 4.1 (Proposition 5 in the Appendix)**.** *For any $\lambda > 0$, we have*

$$\lambda < \vartheta(\lambda, N) \leq \lambda + \frac{1}{N} \mathrm{Tr}[T_K], \qquad 1 \leq \partial_\lambda \vartheta(\lambda, N) \leq \frac{1}{\lambda} \vartheta(\lambda, N),$$

moreover $\vartheta(\lambda, N)$ is decreasing as a function of N.

Remark 4.2. As $N \to \infty$, we have $\vartheta(\lambda, N)$ decreases down to λ (see also Figure 4.3.1), in agreement with the fact that $A_\lambda \to \tilde{A}_\lambda$.

As $\lambda \to 0$, the above upper bound for $\partial_\lambda \vartheta$ becomes useless. Still, assuming that the spectrum of K has a sufficiently fast power-law decay, we get:

Proposition 4.2 (Proposition 9 in the Appendix)**.** *If $d_k = \Theta(k^{-\beta})$ for some $\beta > 1$, there exist $c_0, c_1, c_2 > 0$ such that for any $\lambda > 0$*

$$\lambda + c_0 N^{-\beta} \leq \vartheta(\lambda, N) \leq \ c_2\lambda + c_1 N^{-\beta}, \qquad 1 \leq \partial_\lambda \vartheta(\lambda, N) \leq c_2.$$

Approximation of the SCT from the training data

The SCT ϑ and its derivative $\partial_\lambda \vartheta$ are functions of λ, N, and of the spectrum of T_K. In practice, the spectrum of T_K is not known: for example, in the classical setting, one does not know the true data distribution σ. Fortunately, ϑ can be approximated by $1/m_G(-\lambda)$, where m_G is the *Stieltjes Transform* of the Gram matrix, defined by $m_G(z) = \operatorname{Tr}\left[(\frac{1}{N}G - zI_N)^{-1}\right]$. Namely, we get:

Proposition 4.3 (Proposition 3 in the Appendix)**.** *For any $\lambda > 0, s \in \mathbb{N}$, there is a $\boldsymbol{c}_s > 0$ such that*

$$\mathbb{E}\left[\left|{}^{1}\!/_{\vartheta(\lambda)} - m_G(-\lambda)\right|^{2s}\right] \leq \frac{\boldsymbol{c}_s(\operatorname{Tr}[T_K])^{2s}}{\lambda^{4s}N^{3s}}.$$

Remark 4.3. Likewise, we have $\partial_\lambda \vartheta \approx \left(\partial_z m_G(z)/m_G(z)^2\right)|_{z=-\lambda}$, as shown in the Appendix.

4.4 Risk Prediction with KARE

In this section, we show that the Expected Risk $\mathbb{E}[R^\epsilon(\hat{f}^\epsilon_\lambda)]$ can be approximated in terms of the training data by the Kernel Alignment Risk Estimator (KARE).

Definition 4. *The Kernel Alignment Risk Estimator (KARE) ρ is defined by*

$$\rho(\lambda, N, y^\epsilon, G) = \frac{\frac{1}{N}\left(y^\epsilon\right)^T\left(\frac{1}{N}G + \lambda I_N\right)^{-2} y^\epsilon}{\left(\frac{1}{N}\operatorname{Tr}\left[\left(\frac{1}{N}G + \lambda I_N\right)^{-1}\right]\right)^2}.$$

In the following, using Theorems 4.1 and D.2, we give an approximation for the expected risk and expected empirical risk in terms of the SCT and the true function f^*. This yields the important relation (4.4.1) in Section 4.4, which shows that the KARE can be used to efficiently approximate the kernel risk.

Expected Risk and Expected Empirical Risk

The expected risk is approximated, in terms of the SCT and the true function f^*, by

$$\tilde{R}^\epsilon(f^*, \lambda, N, K, \pi) = \partial_\lambda \vartheta(\lambda)(\|(I_{\mathcal{C}} - \tilde{A}_\vartheta)f^*\|_S^2 + \epsilon^2),$$

as shown by the following:

Theorem 4.3 (Theorem 16 in the Appendix)**.** *There exists a constant $\boldsymbol{C}_2 > 0$ and a polynomial $\boldsymbol{P}_2$ with nonnegative coefficients and with $\boldsymbol{P}_2(0) = 0$, such that we have*

$$\left|\mathbb{E}[R^\epsilon(\hat{f}^\epsilon_\lambda)] - \tilde{R}^\epsilon(f^*, \lambda, N, K, \pi)\right| \leq \left(\frac{\boldsymbol{C}_2}{N} + \boldsymbol{P}_2\left(\frac{\operatorname{Tr}[T_K]}{\lambda N^{\frac{1}{2}}}\right)\right)\tilde{R}^\epsilon(f^*, \lambda, N, K, \pi).$$

Proof. (Sketch; the full proof is given in the Appendix). From the bias-variance decomposition:

$$\mathbb{E}[R^\epsilon(\hat{f}_\lambda^\epsilon)] = R^\epsilon(\mathbb{E}[\hat{f}_\lambda^\epsilon]) + \sum_{k=1}^{\infty} \mathrm{Var}(\langle f^{(k)}, \hat{f}_\lambda^\epsilon \rangle_S).$$

By Theorem 4.1, and a small calculation, the bias is approximately $\|(I_{\mathcal{C}} - \tilde{A}_\vartheta) f^*\|_S^2 + \epsilon^2$. By Theorem D.2, and a calculation, the variance is approximately $(\partial_\lambda \vartheta(\lambda) - 1)(\|(I_{\mathcal{C}} - \tilde{A}_\vartheta) f^*\|_S^2 + \epsilon^2)$. □

The approximate expected risk $\tilde{R}^\epsilon(f^*, \lambda, N, K, \pi)$ is increasing in both ϑ and $\partial_\lambda \vartheta$. As λ increases, the bias increases with ϑ, while the variance decreases with $\partial_\lambda \vartheta$: this leads to the bias-variance tradeoff. On the other hand, as a function of N, ϑ is decreasing but $\partial_\lambda \vartheta$ is generally not monotone: this can lead to so-called multiple descent curves in the risk as a function of N [138].

Note also that if we decompose the true function along the principal components $f^* = \sum_{k=1}^{\infty} b_k f^{(k)}$, the risk is approximated by $\tilde{R}^\epsilon(f^*) = \partial_\lambda \vartheta(\lambda)(\sum_{k=1}^{\infty} \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2} b_k^2 + \epsilon^2)$.

Remark 4.4. For a decaying ridge $\lambda = cN^{-\gamma}$ for $0 < \gamma < \frac{1}{2}$, as $N \to \infty$, by Proposition D.3, we get $\vartheta(\lambda) \to 0$ and $\partial_\lambda \vartheta(\lambda) \to 1$: this implies that $\mathbb{E}[R^\epsilon(\hat{f}_\lambda^\epsilon)] \to \epsilon^2$. Hence the KRR can learn any continuous function f^* as $N \to \infty$ (even if f^* is not in the RKHS associated with K).

Remark 4.5. In a Bayesian setting, assuming that f^* is random with zero mean and covariance kernel Σ, the optimal choices for the KRR predictor are $K = \Sigma$ and $\lambda = \epsilon^2/N$ (see Section 2.7 in the Appendix). When $K = \Sigma$ and $\lambda = \epsilon^2/N$, the formula of Theorem 6 simplifies (see Corollary 18 in the Appendix) to

$$\mathbb{E}\left[R^\epsilon\left(\hat{f}_\lambda^\epsilon\right)\right] \approx N\vartheta\left(\frac{\epsilon^2}{N}, \Sigma\right).$$

The empirical risk (or train error) $\hat{R}^\epsilon(\hat{f}_\lambda^\epsilon) = \lambda^2 (y^\epsilon)^T (\frac{1}{N} G + \lambda I_N)^{-2} y^\epsilon$ can be analyzed with the same theoretical tools. Its approximation in terms of the SCT is given as follows:

Theorem 4.4 (Theorem 17 in the Appendix)**.** *There exists a constant $\boldsymbol{C}_3 > 0$ and a polynomial $\boldsymbol{P}_3$ with nonnegative coefficients and with $\boldsymbol{P}_3(0) = 0$ such that we have*

$$\left|\mathbb{E}[\hat{R}^\epsilon(\hat{f}_\lambda^\epsilon)] - \frac{\lambda^2}{\vartheta(\lambda)^2} \tilde{R}^\epsilon(\hat{f}_\lambda^\epsilon, \lambda, N, K, \pi)\right| \leq \left(\frac{1}{N} + \boldsymbol{P}_3\left(\frac{\mathrm{Tr}[T_K]}{\lambda N}\right)\right) \tilde{R}^\epsilon(f^*, \lambda, N, K, \pi).$$

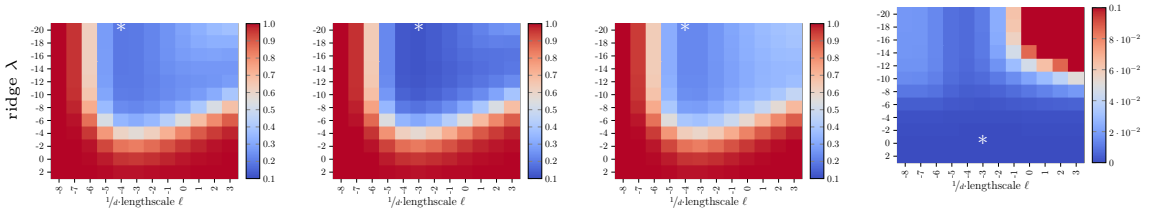
KARE: Kernel Alignment Risk Estimator

While the above approximations (Theorems D.3 and 4.4) for the expected risk and empirical risk depend on f^*, their combination yields the following relation, which is surprisingly independent of f^*:

$$\mathbb{E}\left[R^\epsilon\left(\hat{f}_\lambda^\epsilon\right)\right] \approx \frac{\vartheta^2}{\lambda^2} \mathbb{E}\left[\hat{R}^\epsilon\left(\hat{f}_\lambda^\epsilon\right)\right]. \tag{4.4.1}$$

Since ϑ can be approximated from the training set (see Proposition 4.3), so can the expected risk. Assuming that the risk and empirical risk concentrate around their expectations, we get the KARE:

$$R^\epsilon\left(\hat{f}_\lambda^\epsilon\right) \approx \rho(\lambda, N, y^\epsilon, G) = \frac{\frac{1}{N}\left(y^\epsilon\right)^T \left(\frac{1}{N} G + \lambda I_N\right)^{-2} y^\epsilon}{\left(\frac{1}{N} \mathrm{Tr}\left[\left(\frac{1}{N} G + \lambda I_N\right)^{-1}\right]\right)^2}.$$



(a) Risk (b) KARE Preds. (c) LOO Preds. (d) Log-likelihood Preds.

Figure 4.4.1: *Comparision of risk predictors.* We calculate the risk (i.e. test error) of $\hat{f}_\lambda^\epsilon$ on MNIST with the RBF Kernel for various values of ℓ and λ on $N = 200$ data points (same setup as Fig. 4.1.1). We mark the minimum MSE achieved with a star. We display the predictions of KARE and leave-one-out (LOO); both find the hyper-parameters minimizing the risk. We also show the (normalized) log-likehood estimator and observe that it favors large λ values. Axes are $\log_2$ scale.

Remark 4.6. As shown in the Appendix, estimating the risk of the expected predictor $\mathbb{E}[\hat{f}_\lambda^\epsilon]$ yields:

$$R^\epsilon(\mathbb{E}[\hat{f}_\lambda^\epsilon]) \approx \varrho(\lambda, N, y^\epsilon, G) = \frac{(y^\epsilon)^T(\frac{1}{N}G + \lambda I_N)^{-2}y^\epsilon}{\mathrm{Tr}[(\frac{1}{N}G + \lambda I_N)^{-2}]}.$$

Note that both ρ and ϱ are invariant (as is the risk) under the simultaneous rescaling $K, \lambda \rightsquigarrow \alpha K, \alpha\lambda$.

The KARE can be used to optimize the risk over the space of kernels, for instance to choose the ridge and length-scale. The most popular kernel selection techniques are (see Figure 4.4.1):

- Leave-one-out: accurate estimator of the risk on a test set, it has a closed-form formula similar yet different from the KARE [181].
- Kernel likelihood (Chapter 5 of [178]): efficient to optimize and takes into account the ridge, but not a risk estimator; unlike the risk, not invariant under the simultaneous rescaling $K, \lambda \rightsquigarrow \alpha K, \alpha\lambda$.
- Classical kernel alignment [40]: very efficient to optimize and scale invariant, but not a risk estimator, not sensitive to small eigenvalues and inadequate to select hyperparameters such as the ridge.

The KARE has the following three desirable properties:

- it can be computed efficiently on the training data, and optimized over the space of kernels;
- like the risk, it is invariant under the simultaneous rescaling $K, \lambda \rightsquigarrow \alpha K, \alpha\lambda$;
- it is sensitive to the small Gram matrix eigenvalues and to the ridge λ.

4.5 Conclusion

In this paper, we introduce new techniques to study the Kernel Ridge Regression (KRR) predictor and its risk. We obtain new precise estimates for the test and train error in terms of a new object,

the Signal Capture Threshold (SCT), which identifies the components of a true function that are being learned by the KRR: our estimates reveal a remarkable relation, which leads one to the Kernel Alignment Risk Estimator (KARE). The KARE is a new efficient way to estimate the risk of a kernel predictor based on the training data only. Numerically, we observe that the KARE gives a very accurate prediction of the risk for Higgs and MNIST datasets for a variety of classical kernels.

Chapter 5

Freeze and Chaos: NTK views on DNN Normalization, Checkerboard and Boundary Artifacts

Abstract

We analyze architectural features of Deep Neural Networks (DNNs) using the so-called Neural Tangent Kernel (NTK), which describes the training and generalization of DNNs in the infinite-width setting. In this setting, we show that for fully-connected DNNs, as the depth grows, two regimes appear: *freeze* (or *order*), where the (scaled) NTK converges to a constant, and *chaos*, where it converges to a Kronecker delta. Extreme freeze slows down training while extreme chaos hinders generalization. Using the scaled ReLU as a nonlinearity, we end up in the frozen regime. In contrast, Layer Normalization brings the network into the chaotic regime. We observe a similar effect for Batch Normalization (BN) applied after the last nonlinearity. We uncover the same freeze and chaos modes in Deep Deconvolutional Networks (DC-NNs). Our analysis explains the appearance of so-called checkerboard patterns and border artifacts. Moving the network into the chaotic regime prevents checkerboard patterns; we propose a graph-based parametrization which eliminates border artifacts; finally, we introduce a new layer-dependent learning rate to improve the convergence of DC-NNs. We illustrate our findings on DCGANs: the frozen regime leads to a collapse of the generator to a checkerboard mode, which can be avoided by tuning the nonlinearity to reach the chaotic regime. As a result, we are able to obtain good quality samples for DCGANs without BN.

5.1 Introduction

The training of Deep Neural Networks (DNN) involves a great variety of architecture choices. It is therefore crucial to find tools to understand their effects and to compare them. For example, Batch Normalization (BN) [98] has proven to be crucial in the training of DNNs but remains ill-understood. While BN was initially introduced to solve the problem of "covariate shift", recent results [193] suggest an effect on the smoothness of the loss surface. Some alternatives to BN have been proposed [132, 192, 118], yet it remains difficult to compare them theoretically. Recent theoretical results [230] suggest some relation to the transition from "order" (freeze) to "chaos" observed as the depth of the NN goes to infinity [173, 42, 231, 197, 87].

The impact of architecture is very apparent in GANs [78]: their results are heavily affected by the architecture of the generator and discriminator [174, 237, 27, 114] and the training may fail without BN [7, 223].

Recently, there has been important advances [105, 51, 2, 34, 128] in the understanding of the training of DNNs when the number of neurons in each hidden layer is very large. These results give new tools to study the asymptotic effect of BN. In particular, the Neural Tangent Kernel (NTK) [105] illustrates the effect of architecture on the training of DNNs and also describes their loss surface [112, 106]. The NTK can easily be extended to Convolutional Neural Networks (CNNs) and other architectures [228, 6], hence allowing comparison. Since the first apparition of this work on arxiv, the freeze/chaos transition for the NTK has been further studied in [88, 86, 224, 96]. To stay consistent with the literature, we will henceforth use the term *order* in place of *freeze*.

Our Contributions

In Section 5.3, we study fully-connected deep neural networks of infinite width as the depth L increases. Using a characteristic value $r_{\sigma,\beta}$ (for the non-linearity σ and the amount of bias β), we identify two regimes:

- In the **Ordered regime** (when $r_{\sigma,\beta} < 1$) the NTK approaches a constant kernel, leading to an ill-conditioned kernel Gram matrix and a very narrow valley around the global minimum, hence hurting convergence of the network.
- In the **Chaotic regime** (when $r_{\sigma,\beta} > 1$) the NTK approaches a Kronecker delta kernel, leading to an identity kernel Gram matrix and wide valley around the global minimum, leading to fast convergence but conversely hurting generalization.

For very large depths only critical networks ($r_{\sigma,\beta} = 1$) can be trained successfully [88, 86, 224]. Outside of this large depth regime, the characteristic value plays a similar role to the lengthscale parameters in traditional kernel methods, depending on the application different values of $r_{\sigma,\beta}$ may be optimal. Therefore we discuss in Section 5.4 how $r_{\sigma,\beta}$ can be changed. A network can be pushed towards the ordered regime by increasing the amount of bias β. Unfortunately even for $\beta = 0$ the network can remain in the ordered regime: to move to the chaotic regime, we show that one can use normalization. We study three types of normalizations and show their 'chaotic' properties:

- We introduce **Nonlinearity Normalization**, which modifies the non-linearity $\sigma(x) \mapsto \frac{\sigma(x)-b}{v}$ to normalize it over random Gaussian inputs. With a normalized nonlinearity, the characteristic value $r_{\sigma,\beta}$ can always reach the chaotic region for small enough β.
- We show that in the infinite width limit, **Layer Normalization** has no effect on training when applied before the nonlinearity and is equivalent to Nonlinearity Normalization when applied after the nonlinearity: in the latter case, the network can therefore reach the chaotic regime.
- We show that **Batch Normalization** at the last layer of the network controls the intensity of the constant mode of the kernel Gram matrix which otherwise dominates in the ordered regime, hence avoiding the slow convergence related to the ordered phase.

Finally in Section 5.6, we conduct a similar analysis on deconvolutional networks, to understand problems of mode collapse in Generative Adversarial Networks (GANs). Mode collapse occurs

when a GAN only generates the same image for all inputs. Typically the generated image features checkerboard patterns and border artifacts. We show that these problems can be mitigated by modifying the generator:

- To avoid **border artifacts**, we propose a Graph-based parameterization of deconvolutional networks which ensures that the intensity of the NTK is constant over the whole image, preventing the dip in intensity on the border with the traditional parametrization.

- To circumvent the collapse and the **checkerboard patterns** we show that one needs to avoid the ordered regime, where the dominating eigenvectors of the NTK Gram matrix are constant over the inputs of the generator and feature checkerboard patterns. This may explain why normalization is so crucial in practice for the training of GANs, to avoid the ordered regime in the generator.

The traditional technique to avoid Mode Collapse is to use Batch Normalization. Based on our results, we are able to train a simple DC-GAN without Batch Normalization, using a Graph-based parameterization and Nonlinearity Normalization.

Related Works

The order/chaos transition was first observed for the covariance of the activations in neural networks at initialization [173, 42, 231, 197, 87]. The frontier between the two regimes is the same as for the NTK, however the NTK analysis allows one to describe the behavior of the network during training.

Since and simultaneously with the original release of this paper on arxiv, there has been numerous works studying the order/chaos transition for the NTK: the edge of chaos ($r_{\sigma,\beta} = 1$) is studied in more details for both fully-connected and convolutional networks in [88, 86, 224] and the effect of resnet architecture in [88, 86, 96]. To our knowledge, only our paper shows the chaotic effect of normalization and the order/chaos transition in deconvolutional networks leading to checkerboard patterns. Furthermore, while the aforementioned works conclude that only the edge of chaos is viable for training of very deep networks, we show that for reasonable depths the characteristic value plays a similar role to the lengthscale parameters in traditional kernel methods, and we show that for GANs it is advantageous to have a generator in the chaotic regime.

Our work (as well as the aforementioned order/chaos literature) studies infinitely wide DNNs in the linear or lazy regime, characterized by the NTK staying constant during training, by changing the initialization and/or parametrization of DNNs, one can instead reach the so-called mean-field regime where the NTK evolves in time [183, 35, 151, 229]. To our knowledge, the order/chaos transition in the mean-field regime has not yet been studied.

Finally note that as described in [83, 84], the limiting behavior of the NTK can be very different in the limit when both width and depth go to infinity simultaneously than in the finite depth, infinite width limit of [105, 51, 2, 128]. This work (and other order/chaos literature) gives finite depth bounds for the infinite width limit, roughly speaking, our work applies to large depths and widths but with a width significantly larger than the depth, while in [83, 84] the depth and width are of the same order.

5.2 Fully-Connected Neural Networks

The first type of architecture we consider are deep Fully-Connected Neural Networks (FC-NNs). An FC-NN $\mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ with nonlinearity $\sigma : \mathbb{R} \to \mathbb{R}$ consists of $L+1$ layers ($L-1$ hidden layers), respectively containing $n_0, n_1, \ldots, n_L$ neurons. The parameters are the connection weight matrices $W^{(\ell)} \in \mathbb{R}^{n_{\ell+1} \times n_\ell}$ and bias vectors $b^{(\ell)} \in \mathbb{R}^{n_{\ell+1}}$ for $\ell = 0, 1, \ldots, L-1$. Following [105], the network parameters are aggregated into a single vector $\theta \in \mathbb{R}^P$ and initialized using iid standard Gaussians $\mathcal{N}(0,1)$. For $\theta \in \mathbb{R}^P$, the DNN network function $f_\theta : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ is defined as $f_\theta(x) = \tilde{\alpha}^{(L)}(x)$, where the activations and preactivations $\alpha^{(\ell)}, \tilde{\alpha}^{(\ell)}$ are recursively constructed using the NTK parametrization: we set $\alpha^{(0)}(x) = x$ and, for $\ell = 0, \ldots, L-1$,

$$\tilde{\alpha}^{(\ell+1)}(x) = \frac{\sqrt{1-\beta^2}}{\sqrt{n_\ell}} W^{(\ell)} \alpha^{(\ell)}(x) + \beta b^{(\ell)}$$
$$\alpha^{(\ell+1)}(x) = \sigma\left(\tilde{\alpha}^{(\ell+1)}(x)\right),$$

where σ is applied entry-wise and $\beta \geq 0$.

Remark. *The hyperparameter β allows one to balance the relative contributions of the connection weights and of the biases during training; in our numerical experiments, we set $\beta = 0.1$. Note that the variance of the normalized bias $\beta b^{(\ell)}$ at initialization can be tuned by β.*

Neural Tangent Kernel

The NTK [105] describes the evolution of $(f_{\theta_t})_{t \geq 0}$ in function space during training. In the FC-NN case, the NTK $\Theta_\theta^{(L)} : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}^{n_L \times n_L}$ is defined by

$$\Theta_{\theta,kk'}^{(L)}(x, x') = \sum_{p=1}^{P} \partial_{\theta_p} f_{\theta,k}(x)\, \partial_{\theta_p} f_{\theta,k'}(x').$$

For a dataset $x_1, \ldots, x_N \in \mathbb{R}^{n_0}$, we define the *output* vector $Y_\theta = (f_{\theta,k}(x_i))_{ik} \in \mathbb{R}^{N n_L}$. The DNN is trained by optimizing a cost $C : \mathbb{R}^{n_L N} \to \mathbb{R}$ through gradient descent, defining a flow $\partial_t \theta_t = -\nabla_\theta C(Y_\theta)\big|_{\theta_t}$. The evolution of the output vector Y_θ can be expressed in terms of the NTK Gram Matrix $\tilde{\Theta}_\theta^{(L)} = \left(\Theta_{\theta,km}^{(L)}(x_i, x_j)\right)_{ik,jm} \in \mathbb{R}^{n_L N \times n_L N}$ and gradient $\nabla_Y C(Y_{\theta_t}) \in \mathbb{R}^{n_L N}$:

$$\partial_t Y_{\theta_t} = -\tilde{\Theta}_{\theta_t}^{(L)} \nabla_Y C(Y_{\theta_t}).$$

Infinite-Width Limit

Following [159, 37, 126], in the overparametrized regime at initialization, the preactivations $\left(\tilde{\alpha}_i^{(\ell)}\right)_{i=1,\ldots,n_\ell}$ are described by iid centered Gaussian processes with covariance kernels $\Sigma^{(\ell)}$ constructed as follows. For a kernel K, set

$$\mathbb{L}_K^g(z_0, z_1) = \mathbb{E}_{(y_0,y_1) \sim \mathcal{N}\left(0, (K(z_i,z_j))_{i,j=0,1}\right)}\left[g(y_0)\, g(y_1)\right].$$

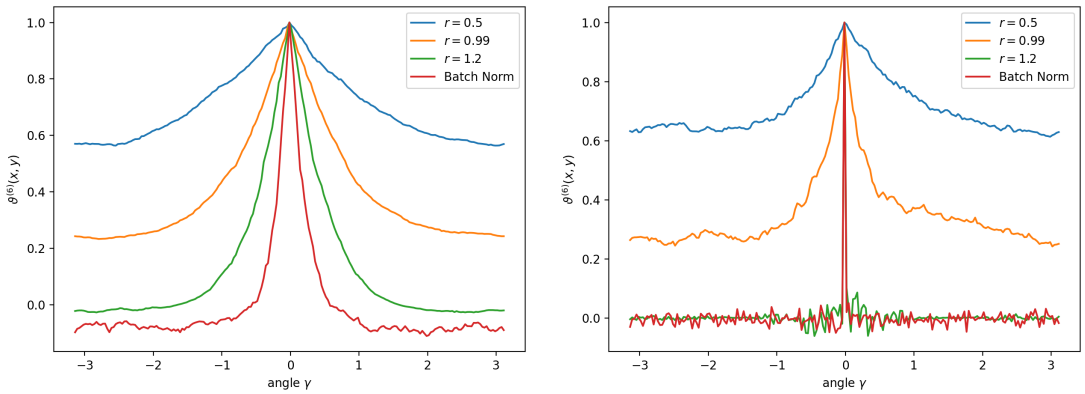


Figure 5.2.1: The NTK on the unit circle for four architectures with depth $L = 5$ (left) and $L = 25$ (right) are plotted: vanilla ReLU network with $\beta = 1.0$ (blue) and $\beta = 0.1$ (orange), with a normalized ReLU / Layer norm. (green) and with Batch Norm (red). Both networks have width 3000, but the deeper network is further from convergence, leading to more noise.

The *activation kernels* $\Sigma^{(\ell)}$ are defined recursively by

$$\Sigma^{(0)}(z_0, z_1) = \beta^2 + \frac{(1-\beta^2)}{n_0} z_0^T z_1$$
$$\Sigma^{(\ell+1)}(z_0, z_1) = \beta^2 + (1-\beta^2)\, \mathbb{L}^{\sigma}_{\Sigma^{(\ell)}}(z_0, z_1).$$

While random at initialization, in the infinite-width-limit, the NTK converges to a deterministic limit, which is moreover constant during training:

Theorem 5.2.1. *As $n_1, \ldots, n_{L-1} \to \infty$, for any $z_0, z_1 \in \mathbb{R}^{n_0}$ and any $t \geq 0$, the kernel $\Theta^{(L)}_{\theta_t}(z_0, z_1)$ converges to $\Theta^{(L)}_{\infty}(z_0, z_1) \otimes \mathrm{Id}_{n_L}$, where*

$$\Theta^{(L)}_{\infty}(z_0, z_1) = \sum_{\ell=1}^{L} \Sigma^{(\ell)}(z_0, z) \prod_{l=\ell+1}^{L} \dot{\Sigma}^{(l)}(z_0, z_1)$$

and $\dot{\Sigma}^{(l)} = (1-\beta^2)\mathbb{L}^{\dot{\sigma}}_{\Sigma^{(l-1)}}$ with $\dot{\sigma}$ denoting the derivative of σ.

We refer to [105] for a proof for the sequential limit $n_1 \to \infty, \ldots, n_{L-1} \to \infty$ and [228, 6] for the simultaneous limit $\min(n_1, \ldots, n_{L-1}) \to \infty$. As a consequence, in the infinite-width limit, the dynamics of the labels $Y_{\theta_t,k} \in \mathbb{R}^N$ for each outputs k acquires a simple form in terms of the limiting NTK Gram matrix $\tilde{\Theta}^{(L)}_{\infty} \in \mathbb{R}^{N \times N}$

$$\partial_t Y_{\theta_t,k} = -\tilde{\Theta}^{(L)}_{\infty} \nabla_{Y_k} C(Y_{\theta_t}),$$

where the Gram matrix is now fixed.

5.3 Order and Chaos in FC-NNs

We now investigate the large L behavior of the NTK (in the infinite-width limit), revealing a transition between two phases: "order" and "chaos". To ensure that the variance of the neurons is constant for all depths ($\Sigma^{(\ell)}(x,x)=1$) we consider *standardized* nonlinearity, i.e. such that

$$\mathbb{E}_{x\sim\mathcal{N}(0,1)}\left[\sigma^2(x)\right]=1$$

and inputs on the *standard* $\sqrt{n_0}$*-sphere*[1]

$$\mathbb{S}_{n_0}=\{x\in\mathbb{R}^{n_0}:\|x\|=\sqrt{n_0}\}.$$

For a standardized σ, the large-depth behavior of the *normalized NTK*

$$\vartheta^{(L)}(x,y):=\frac{\Theta_\infty^{(L)}(x,y)}{\sqrt{\Theta_\infty^{(L)}(x,x)\,\Theta_\infty^{(L)}(y,y)}}$$

is determined by the *characteristic value*

$$r_{\sigma,\beta}=(1-\beta^2)\mathbb{E}_{x\sim\mathcal{N}(0,1)}\left[\dot\sigma^2(x)\right]. \tag{5.3.1}$$

Theorem 5.3.1. *Suppose that σ is twice differentiable and standardized.*

Order: *If $r_{\sigma,\beta}<1$, there exists $C_1>0$ such that for $x,y\in\mathbb{S}_{n_0}$,*

$$1-C_1 L r_{\sigma,\beta}^L\le\vartheta^{(L)}(x,y)\le 1.$$

Chaos: *If $r_{\sigma,\beta}>1$, for $x\neq\pm y$ in $\mathbb{S}_{n_0}$, there exist $h<1$ and $C_2>0$, such that*

$$\left|\vartheta^{(L)}(x,y)\right|\le C_2 h^L.$$

Theorem 5.3.1 shows that in the ordered regime, the normalized NTK $\vartheta^{(L)}$ converges to a constant as $L\to\infty$, whereas in the chaotic regime, it converges to a Kronecker δ (taking value 1 on the diagonal, 0 elsewhere). This suggests that the training of deep FC-NN is heavily influenced by the characteristic value: when $r_{\sigma,\beta}<1$, $\Theta^{(L)}$ becomes constant, thus slowing down the training, whereas when $r_{\sigma,\beta}>1$, $\Theta^{(L)}$ is concentrates on the diagonal, ensuring fast training, but limiting generalization. To train very deep FC-NNs, it is necessary to lie "on the edge of chaos" $r_{\sigma,\beta}=1$ [173, 231].

The order/chaos transition can also be related to the "roughness" of the loss around a global minimum. As observed in [106] the eigenvalues of the Hessian at convergence are the same as those of the NTK Gram matrix. In the chaotic regime all eigenvalues are close to each other, leading to a "wide valley" around the minimum, on the other hand in the ordered regime, the dominating eigenvalue (corresponding to the constant mode) is much larger than the other eigenvalues, leading to a very "narrow valley".

[1] Note that high dimensional datasets tend to concentrate on hyperspheres: for example in GANs [78] the inputs of a generator are vectors of iid $\mathcal{N}(0,1)$ entries which concentrate around $\mathbb{S}_{n_0}$ for large dimensions.

Order and Chaos for ReLU networks

Theorem 5.3.1 does not apply directly to the standardized ReLU $\sigma(x) = \sqrt{2}\max(x, 0)$, because it is not differentiable in 0. The characteristic value for the standardized ReLU is $r_{\sigma,\beta} = 1 - \beta^2$ which lies in the ordered regime for $\beta > 0$:

Theorem 5.3.2. *With the same notation as in Theorem 5.3.1, taking σ to be the standardized ReLU and $\beta > 0$, the NTK is in the ordered regime: there exists a constant C such that $1 - C r_{\sigma,\beta}^{L/2} \leq \vartheta^{(L)}(x, y) \leq 1$.*

We observe two interesting (and potentially beneficial) properties of the standardized ReLU:

1. Its characteristic value $r_{\sigma,\beta} = 1 - \beta^2$ is very close to the 'edge of chaos' for small β and typically with LeCun initialization the variance of the bias at initialization is $\frac{1}{w}$ for w the width, which roughly corresponds to a choice of $\beta = \frac{1}{\sqrt{w}}$.

2. The rate of convergence to the limiting kernel is smaller ($r_{\sigma,\beta}^{L/2}$) for the ReLU than for differentiable nonlinearities ($r_{\sigma,\beta}^{L}$)[2].

These observations suggest that an advantage of the ReLU is that the NTK of ReLU networks converges to its constant limit at a slower rate and may naturally offer a good tradeoff between generalization and training speed.

5.4 Chaotic effect of normalization

Figure 5.2.1 shows that even on the edge of chaos, the NTK may exhibit a strong constant component (i.e. $\vartheta(x, y) > 0.2$ for all x, y) which can lead to a bad conditioning of the Gram matrix governing the infinite-width training behavior. It may be helpful to slightly 'move' the network towards the chaotic regime to reduce this effect. In Figure 5.2.1, $r_{\sigma,\beta}$ plays a similar role to that of the lengthscale parameter in classical kernel methods: increasing $r_{\sigma,\beta}$ makes the NTK 'narrower', reducing the correlation length.

From the definition 5.3.1 of the characteristic value, we see that increasing the bias pushes the network towards the ordered regime, whereas $r_{\sigma,\beta}$ reaches its highest value $\mathbb{E}\left[\dot{\sigma}^2(x)\right]$ when the bias is 0, which may still be in the ordered regime (or on the edge with the ReLU). We are therefore interested in ways to push the network further towards the chaotic regime.

In this section, we show that Layer Normalization is asymptotically equivalent to Nonlinearity Normalization which entails $r_{\sigma,\beta} > 1$ for β small enough. While Batch normalization cannot be directly interpreted in terms of $r_{\sigma,\beta}$, it is easy to show that it directly controls the constant component of the NTK, which is characteristic of the ordered regime.

Nonlinearity Normalization

Intuitively, the dominating constant component in ReLU networks is partly a consequence of the ReLU being non-negative: after the first hidden layer, all negative correlations become positive (i.e. $\Sigma^{(1)}(x, y) \geq \beta$ for all x, y, even $x = -y$). One can address this issue thanks to the following. We

[2] Of course the rates of Theorems 5.3.1 and 5.3.2 may not be tight, but from the proofs in Appendix B.1 one can observe that the rate of $r_{\sigma,\beta}^{L/2}$ appears as a result of the non-differentiability of the ReLU.

shall write Z for a random variable with standard normal distribution. We say that σ is normalized if $\mathbb{E}[\sigma(Z)] = 0$ and $\mathbb{E}[\sigma(Z)^2] = 1$. In particular, if $\sigma \neq \text{id}$, then

$$\overline{\sigma}(\cdot) := \frac{\sigma(\cdot) - \mathbb{E}[\sigma(Z)]}{\sqrt{\mathbb{E}[(\sigma(Z) - \mathbb{E}[\sigma(Z)])^2]}}$$

is normalized. By Poincaré Inequality, after nonlinearity normalization, one can always reach the chaotic regime:

Proposition 5.4.1. *If $\sigma \neq \text{id}$ is normalized, then $\mathbb{E}\left[\dot{\sigma}^2(Z)\right] > 1$ and $r_{\sigma,\beta} > 1$ for $\beta > 0$ small enough.*

Layer Normalization

Nonlinearity Normalization is closely related to Layer Normalization (LN). We define a normalization layer on any vector $v \in \mathbb{R}^d$ as

$$\text{LN}(v) = \sqrt{d}\frac{v - \bar{v}}{\|v - \bar{v}\|}.$$

for $\bar{v} = \frac{1}{d}\sum_i v_i$. We consider two types of Layer normalization depending on whether we apply the normalization layer before or after the nonlinearity: pre-nonlinearity LN where the activations are changed to $\alpha^{(\ell)}(x) = \sigma(\text{LN}(\tilde{\alpha}^{(\ell)}(x)))$ and post-nonlinearity LN where they are changed to $\alpha^{(\ell)}(x) = \text{LN}(\sigma(\tilde{\alpha}^{(\ell)}(x)))$. Depending on whether Layer Normalization is applied before or after the nonlinearity it has either no effect or is equivalent to Nonlinearity Normalization:

Proposition 5.4.2. *Suppose that the inputs belong to $\mathbb{S}_{n_0}$ and that σ is standardized. In the infinite width limit, the network function is the same at initialization and during training:*

- *with or without pre-nonlinearity LN,*
- *with Post-nonlinearity LN or with Nonlinearity Normalization.*

Proof. (sketch) At initialization, the normalization parameters $\bar{v}$ and $\|v - \bar{v}\|/\sqrt{d}$ respectively converge to 0 and 1 for pre-nonlinearity LN, and to $\mathbb{E}[\sigma(Z)]$ and $\sqrt{\mathbb{E}[(\sigma(Z) - \mathbb{E}[\sigma(Z)])^2]}$ for post-nonlinearity LN. These values stay asymptotically constant during training because the rate of change of the (pre-)activations is sufficiently small in the linear/lazy regime. □

Batch Normalization

For any $N \times d$ matrix of features X leading to a $N \times N$ Gram matrix $K = \frac{1}{d}XX^T$, the Rayleigh quotient $\frac{1}{N}\mathbf{1}^T K \mathbf{1}$ of the constant vector $\mathbf{1}$ measures how big the constant component is. Applying Batch Normalization (BN) at a layer ℓ centers (and standardizes) the activations[3] $\alpha_j^{(\ell)}(x_i)$ over a batch $x_1, ..., x_N$, thus zeroing the constant Rayleigh quotient of the $N \times N$ features Gram matrices $\tilde{\Sigma}^{(\ell)}$ with entries $\tilde{\Sigma}_{ij}^{(\ell)} = \frac{1}{n_\ell}\sum_{k=1}^{n_\ell} \alpha_k^{(\ell)}(x_i)\alpha_k^{(\ell)}(x_j)$. Adding a single BN layer after the last hidden layer controls the constant Rayleigh quotient of the NTK Gram matrix $\tilde{\Theta}^{(L)}$:

Lemma 5.4.3. *Consider FC-NN with L layers, with a ost-nonlinearity-BN after the last nonlinearity. Then $\frac{1}{N}\mathbf{1}^T\tilde{\Theta}^{(L)}\mathbf{1} = \beta^2$.*

[3] We consider here *post-nonlinearity* BN, it is common to normalize the pre-activations $\tilde{\alpha}^{(\ell)}$ instead.

In contrast, for a network in the extreme ordered regime, i.e. such that $\Theta^{(L)}(x,y) \approx c$ for some constant $c > 0$, the constant Rayleigh quotient scales as $\frac{1}{N}\mathbf{1}^T\tilde{\Theta}^{(L)}\mathbf{1} \approx cN$. The analysis of BN presented in [113] is also closely related to this phenomenon.

The chaotic effect of Batch Normalization can also be observed in Figure 5.2.1 where the NTK with Nonlinearity and Batch Normalization have a similar behavior.

5.5 Convolutional Networks

In this section, we introduce convolutional networks as a special case of a general Graph-based neural networks. We then describe the convergence of the NTK in the infinite width limit.

Graph-based Neural Networks (GB-NNs)

In GB-NNs, the neurons are indexed by their layer ℓ and their channel $i \in \{1, ..., n_\ell\}$, in convolutional networks each neuron furthermore has a location on the image (or on a downscaled image). The position p of a neuron determines its connections with the neurons of the previous and subsequent layers. Furthermore certain connections are shared, i.e. they evolve together. We abstract these concepts in the following manner:

For each layer $\ell = 0, ..., L$, the neurons are indexed by a position $p \in I_\ell$ and a channel $i = 1, ..., n_\ell$. The sets of positions I_ℓ can be any set, in particular any subset of $\mathbb{Z}^D$. Each position $p \in I_{\ell+1}$ has a set of parents $P(p) \subset I_\ell$ which are neurons of the previous layer connected to p. The connections from the parent (q, ℓ) to the position $(p, \ell+1)$ are encoded in an $n_\ell \times n_{\ell+1}$ weight matrix $W^{(\ell,q\to p)}$. Finally two connections $q \to p$ and $q' \to p'$ can be shared, setting the corresponding matrices to be equal $W^{(\ell,q\to p)} = W^{(\ell,q'\to p')}$.

The inputs of the network x are vectors in $(\mathbb{R}^{n_0})^{I_0}$, for example for colour images of width w and height h, we have $n_0 = 3$ and $I_0 = \{1, ..., w\} \times \{1, ..., h\} \subset \mathbb{Z}^2$. The activations and preactivations $\alpha^{(\ell)}, \tilde{\alpha}^{(\ell)} \in (\mathbb{R}^{n_\ell})^{I_\ell}$ are constructed recursively using the NTK parametrization: we set $\alpha^{(0,p)}(x) = x^{(p)}$ and for $\ell = 0, \dots, L-1$ and any position $p \in I_{\ell+1}$,

$$\begin{aligned} \tilde{\alpha}^{(\ell+1,p)}(x) &= \beta b^{(\ell)} + \frac{\sqrt{1-\beta^2}}{\sqrt{|P(p)|\, n_\ell}} \sum_{q\in P(p)} W^{(\ell,q\to p)}\alpha^{(\ell,q)}(x) \\ \alpha^{(\ell+1,p)}(x) &= \sigma\left(\tilde{\alpha}^{(\ell+1,p)}(x)\right) \end{aligned} \tag{5.5.1}$$

where σ is applied entry-wise, $\beta \geq 0$ and $|P(p)|$ is the cardinality of $P(p)$.

Deconvolutional networks

Deconvolutional networks (DC-NNs) in dimension D can be seen as a special case of GB-NNs. We first consider borderless DC-NNs, i.e. the set of positions are $I_\ell = \mathbb{Z}^D$ for all layers ℓ. Given window dimensions $(w_1, ..., w_D)$ and strides $(s_1, ..., s_D)$, the set of parents of $p \in I_{\ell+1}$ is the hyperrectangle $P(p) = \{\lfloor p_1/s_1 \rfloor + 1, ..., \lfloor p_1/s_1 \rfloor + w_1\} \times \cdots \times \{\lfloor p_D/s_D \rfloor + 1, ..., \lfloor p_D/s_D \rfloor + w_D\} \subset \mathbb{Z}^D$. Two connections $q \to p$ and $q' \to p'$ are shared if $s_d \mid p_d - p'_d$ (i.e. s_d is a divisor of $p_d - p'_d$) and $q_d - q'_d = \frac{p_d - p'_d}{s_d}$ for all $d = 1, ..., D$. This definition can easily be extended to any other choices of position sets $I_\ell \subset \mathbb{Z}^D$ (for example hyperrectangles) by considering $P(p) \cap I_\ell$ in place of $P(p)$ as parents of p.

Neural Tangent Kernel

As for FC-NNs , in the infinite width limit (when $n_1, ..., n_{L-1} \to \infty$) the preactivations $\tilde{\alpha}_i^{(\ell,p)}(x)$ converge to Gaussian processes with covariance

$$Cov\left(\tilde{\alpha}_i^{(\ell+1,p)}(x), \tilde{\alpha}_j^{(\ell+1,q)}(y)\right) = \delta_{ij}\Sigma^{(\ell,pq)}(x,y).$$

The behavior of the network during training is described by the NTK

$$\Theta_{ij}^{(\ell,pq)}(x,y) = \sum_{k=1}^{P} \partial_{\theta_k}\tilde{\alpha}_i^{(\ell+1,p)}(x)\partial_{\theta_k}\tilde{\alpha}_j^{(\ell+1,q)}(y).$$

In the Appendix E we prove the convergence $\Theta_{ij}^{(\ell,pq)}(x,y) \to \delta_{ij}\Theta_\infty^{(\ell,pq)}(x,y)$ of the NTK for the sequential limit $n_1, \cdots, n_{L-1} \to \infty$ and give formulas for the limiting kernels $\Sigma^{(\ell,pq)}(x,y)$ and $\Theta_\infty^{(\ell,pq)}(x,y)$. The simultaneous limit yields the same formulas.

5.6 Mode Collapse in Generative Adversarial Networks

The order/chaos transition is even more interesting for convolutional networks, in particular in the context of Generative Adversarial Networks (GANs): a common problem in GAN training is the so-called 'mode collapse', where the generator converges to a constant function, hence generating a single image instead of a variety of images. This problem is closely related to the fact that the constant mode of the NTK Gram matrix dominates, and indeed the problem of mode collapse is most prominent in the ordered regime (Figure 5.6.1), while normalization techniques (leading to a chaotic network) mitigate this problem.

In this section, we use the NTK to explain the appearance of border and checkerboard artifacts in generated images. We show that the border artifacts issue can be solved by a change of parametrization and that the checkerboard artifacts occur in the ordered regime, and can hence be avoided by adding normalization and using layer-wise learning rates. With these changes we are able to train GANs on CelebA dataset without Batch Normalization.

Border Effects

A very important element of the NTK parametrization proposed in Section 5.5 is the factors $1/\sqrt{|P(p)|\, n_\ell}$ in the definition of the preactivation (Equation 5.5.1): we scale the contribution of the previous layer according to the number of neurons $|P(p)|\, n_\ell$ (i.e. n_ℓ channels for each of the$|P(p)|$ positions) which are fed into the neuron. For inputs $x \in \mathbb{S}_{n_0}^{I_0}$ (i.e. such that $x^{(p)} \in \mathbb{S}_{n_0}$ for all p), these factors ensure that the limiting variance $\Sigma^{(\ell,pp)}(x,x)$ of $\tilde{\alpha}_i^{(\ell,p)}(x)$ at initialization is the same for all p:

Proposition 5.6.1. *For GB-NNs with the NTK parametrization, $\Sigma^{(\ell,pp)}(x,x)$ and $\Theta_\infty^{(\ell,pp)}(x,x)$ do not depend neither on $p \in I_\ell$ nor on $x \in \mathbb{S}_{n_0}^{I_0}$.*

These factors are usually not present and to compensate, the variance of the weights at initialization is reduced. In convolutional networks with LeCun initialization, the standard deviation of the weights at initialization is set to $\frac{1}{\sqrt{whn_\ell}}$ for w and h the width and height of the window of

convolution, which has roughly the effect of replacing the $\frac{1}{\sqrt{|P(p)|n_\ell}}$ factors by $\frac{1}{\sqrt{whn_\ell}}$. However whn_ℓ is the maximal number of parents that a neuron can have, it is typically attained at positions p in the middle of the image. Positions p on the border of the image have less parents hence leading to a smaller contribution of the previous layer. This leads both kernels $\Sigma^{(\ell,pp)}(x,x)$ and $\Theta^{(\ell,pp)}(x,x)$ to have lower intensity for $p \in I_\ell$ on the border (see Appendix G for an example when $I_\ell = \mathbb{N}$, i.e. when there is one border pixel), leading to border artifacts as seen in Figure 5.6.1.

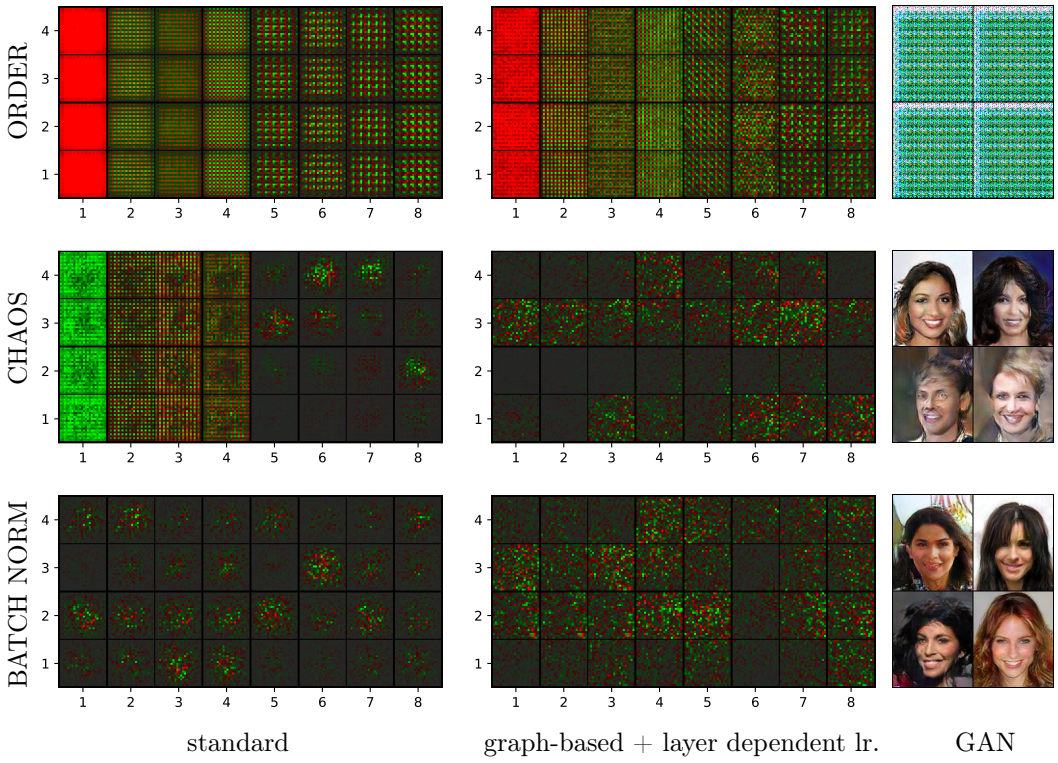


Figure 5.6.1: The left and middle columns represent the first 8 eigenvectors of the NTK Gram matrix of a DC-NN (L=3) on 4 inputs. (left) without the Graph-Based Parametrization (GBP) and the Layer-Dependent Learning Rate (LDLR); (middle) with GBP and LDLR. The right column represents the results of a GAN on CelebA with GBP and LDLR. Each line correspond to a choice of nonlinearity/normalization for the generator: (top) ReLU, (middle) normalized ReLU and (bottom) ReLU with Batch Normalization.

Order, Chaos and Checkerboard Patterns

Large depths deconvolutional networks exhibit a similar Order/Chaos transition as that of FC-NNs, the values of the limiting kernel at different positions $\Theta^{(L,pq)}$ is especially interesting.

For GB-NNs, the value of an output neuron at a position $p \in I_L$ only depends on the inputs which are ancestors of p, i.e. all positions $q \in I_0$ such that there is a chain of connections from q

to p. For the same reason , the NTK $\Theta^{(L,pp')}(x,y)$ only depends on the values $x_q, y_{q'}$ for $q, q' \in I_0$ ancestors of p and p' respectively.

For a stride $s \in \{2,3,\ldots\}^d$, we denote the s-valuation $v_s(n)$ of $n \in \mathbb{Z}^d$ as the largest $k \in \{0,1,2,\ldots\}$ such that $s_i^k \mid n_i$ for all $i = 1,...,d$. The behaviour of the NTK $\Theta^{(L)}_{p,p'}(x,y)$ depends on the s-valuation of the difference of the two output positions. If $v_s(p'-p)$ is strictly smaller than L, the NTK $\Theta^{(L,pp')}(x,y)$ converges to a constant in the infinite-width limit for any $x, y \in \mathbb{S}^{I_0}_{n_0}$. Again the characteristic value $r_{\sigma,\beta}$ plays a central role in the behavior of the large-depth limit. In this context, we define the rescaled NTK as $\vartheta^{(L,pp')}(x,y) = \Theta^{(L,pp')}(x,y)/\sqrt{\Theta^{(L,pp)}(x,x)\Theta^{(L,p'p')}(y,y)}$ (note that the denominator actually does not depend on p, p', x nor y by Proposition 5.6.1)

Theorem 5.6.2. *Consider a borderless DC-NN with position sets $I_\ell = \mathbb{Z}^D$ for all layers ℓ, upsampling stride $s \in \{2,3,\ldots\}^D$ and window sizes $w \in \{1,2,3,\ldots\}^D$. For a standardized twice differentiable σ, there exist constants $C_1, C_2 > 0$, such that the following holds: for $x, y \in \mathbb{S}^{I_0}_{n_0}$, and any positions $p, p' \in I_L$, we have*

Order: *When $r_{\sigma,\beta} < 1$, taking $v = \min(v_s(p-p'), L-1)$, we have*

$$\frac{1-r_{\sigma,\beta}^{v+1}}{1-r_{\sigma,\beta}^{L}} - C_1(v+1)r_{\sigma,\beta}^{v} \leq \vartheta^{(L,pp')}(x,y) \leq \frac{1-r_{\sigma,\beta}^{v+1}}{1-r_{\sigma,\beta}^{L}}.$$

Chaos: *When $r_{\sigma,\beta} > 1$, if either $v_s(p-p') < L$ or if there exists $c < 1$ such that for all positions $q \in I_0$ which are ancestors of p, $\left| x_q^T y_{q+\frac{p'-p}{s^L}} \right| < c$, then there exists $h < 1$ such that*

$$\left|\vartheta^{(L,pp')}(x,y)\right| \leq C_2 h^L.$$

This theorem suggests that in the order regime, the correlations between differing positions p and p' increase with $v_s(p-p')$, which is a strong feature of checkerboard patterns [165]. These artifacts typically appear in images generated by DC-NNs. The form of the NTK also suggests a strong affinity to these checkerboard patterns: they should dominate the NTK spectral decomposition. This is shown in Figure 5.6.1 where the eigenvectors of the NTK Gram matrix for a DC-NN are computed.

In the chaotic regime, the normalized NTK converges to a "scaled translation invariant" Kronecker delta. For two output positions p and $p' = p + ks^L$ we associate the two regions ω and $\omega' = \omega + k$ of the input space which are connected to p and p'. Then $\vartheta^{(L,p,p+ks^L)}(x,y)$ is one if the patch $y_{\omega'}$ is a k translation of x_ω and approximately zero otherwise.

Layer-dependent learning rate

The NTK is the sum $\Theta^{(L)} = \sum_\ell \Theta^{(L)}_{W^{(\ell)}} + \Theta^{(L)}_{b^{(\ell)}}$ over the contributions of the weights $\Theta^{(L,pq)}_{W^{(\ell)}}(x,y) = \sum_{ij} \partial_{W^{(\ell)}_{ij}} f_{\theta,p}(x) \partial_{W^{(\ell)}_{ij}} f_{\theta,q}(y)$ and biases $\Theta^{(L,pq)}_{b^{(\ell)}}(x,y) = \sum_j \partial_{b^{(\ell)}_j} f_{\theta,p}(x) \partial_{b^{(\ell)}_j} f_{\theta,q}(y)$. At the ℓ-th layer, the weights and biases can only contribute to checkerboard patterns of degree $v = L-\ell$ and $v = L-\ell-1$, i.e. patterns with periods $s^{L-\ell}$ and $s^{L-\ell-1}$ respectively, in the following sense:

Proposition 5.6.3. *In a DC-NN with stride $s \in \{2,3,...\}^d$, we have $\Theta^{(L,pp')}_{\infty,W^{(\ell)}}(x,y) = 0$ if $s^{L-\ell} \nmid p'-p$ and $\Theta^{(L,pp')}_{\infty,b^{(\ell)}}(x,y) = 0$ if $s^{L-\ell-1} \nmid p'-p$.*

This suggests that the supports of $\Theta^{(L)}_{\infty,W^{(\ell)}}$ and $\Theta^{(L)}_{\infty,b^{(\ell)}}$ increase exponentially with ℓ, giving more importance to the last layers during training. This could explain why the checkerboard patterns of lower degree dominate in Figure 5.6.1. In the classical parametrization, the balance is restored by letting the number of channels n_ℓ decrease with depth [174]. In the NTK parametrization, the limiting NTK is not affected by the ratios $\frac{n_\ell}{n_k}$. To achieve the same effect, we divide the learning rate of the weights and bias of the ℓ-th layer by $S^{\frac{\ell}{2}}$ and $S^{\frac{(\ell+1)}{2}}$ respectively, where $S = \prod_i s_i$ is the product of the strides. Together with the 'parent-based' parametrization and the normalization of the nonlinearity (in order to lie in the chaotic regime) this rescaling of the learning rate removes both border and checkerboard artifacts in Figure 5.6.1.

5.7 Conclusion

This article shows how the NTK can be used theoretically to understand the effect of architecture choices (such as decreasing the number of channels or batch normalization) on the training of DNNs. We have shown that DNNs in a "order" regime, have a strong affinity to constant modes and checkerboard artifacts: this slows down training and can contribute to a mode collapse of the DC-NN generator of GANs. We introduce simple modifications to solve these problems: the effectiveness of normalizing the nonlinearity, a parent-based parametrization and a layer-dependent learning rates is shown both theoretically and numerically.

Chapter 6

DNN-Based Topology Optimization: Spatial Invariance and Neural Tangent Kernel

Abstract

We study the Solid Isotropic Material Penalisation (SIMP) method with a density field generated by a fully-connected neural network, taking the coordinates as inputs. In the large width limit, we show that the use of DNNs leads to a filtering effect similar to traditional filtering techniques for SIMP, with a filter described by the Neural Tangent Kernel (NTK). This filter is however not invariant under translation, leading to visual artifacts and non-optimal shapes. We propose two embeddings of the input coordinates, which lead to (approximate) spatial invariance of the NTK and of the filter. We empirically confirm our theoretical observations and study how the filter size is affected by the architecture of the network. Our solution can easily be applied to any other coordinates-based generation method.

6.1 Introduction

Topology optimisation [21], also known as structural optimisation, is a method to find optimal shapes subject to some constraints. It has been widely studied in the field of computational mechanics. Here we are interested in the particular case of the Solid Isotropic Material Penalisation (SIMP) method [143, 3], which is a very common method in this field.

Recently some authors have used Deep Neural Networks (DNNs) to perform topology optimisation. We can differentiate two different approaches in the use of DNNs with SIMP. The first approach consists in generating with the classical algorithms a dataset of optimised shapes and train a DNN on this dataset to produce new optimal shapes [15, 207]. Variations of this approach use Generative Adversarial Networks (GAN) [163, 201] to effectively reproduce classical topology optimisation.

In the second approach, the density is generated pointwise by a DNN, which is trained with gradient descent to optimise the density field with respect to the physical constraints, as proposed in [94] to use the power of deep models without giving up exact physics. We focus on the approach of [31, 30] where the density field is generated by a Fully-Connected Neural Network (FCNN) taking the coordinates of a grid as inputs. Surprisingly, [31] observes that the DNN-generated density fields do not feature checkerboard artifacts, which are common in vanilla SIMP. A traditional method

to avoid checkerboard patterns is to add a filter [202, 22], but it is not needed for DNN-generated density fields.

In this paper, we analyse theoretically how the use of a DNN to generate the density field affects the learning. Our main theoretical tool is the Neural Tangent kernel (NTK) introduced in [105] to describe the dynamics of wide neural networks [105, 6, 128, 95].

While this paper focuses on linear elasticity and SIMP, our analysis can extended to other physical problems such as heat transfer [149], or any model where an image is generated by a DNN taking the pixel coordinates as inputs (like in [155]).

Our contribution

In this paper we study topology optimisation with neural networks. The physical density is represented by a neural network taking an embedding of spatial coordinates as inputs, i.e. the density at a point $x \in \mathbb{R}^d$ is given by $f_\theta(\varphi(x))$ for θ the parameters of the network and φ an embedding. We use theoretical tools, in particular the Neural Tangent Kernel (NTK), to understand how the architecture and hyperparameters of the network affect the optimisation of the density field:

- We show that in the infinite width limit (when the number of neurons in the hidden layers grows to infinity), topology optimisation with a DNN is equivalent to topology optimisation with a density filter equal to the "square root" of the NTK. Filtering is a commonly used technique in topology optimisation, aimed to remove checkerboard patterns.

- In topology optimisation as in other physical optimisation problems, it is crucial to guarantee some spatial invariance properties. If the coordinates are taken as inputs of the network directly, the NTK (and the corresponding filter) is not translation invariant, leading to non-optimal shapes and visual artifacts. We present two methods to ensure the spatial invariance of the NTK: embedding the coordinates on the (hyper-)torus or using a random Fourier features embedding (similar to [214]).

- In traditional topology optimisation, the filter size must be tuned carefully. When optimising with a DNN, the filter size depends on the embedding of the coordinates and the architecture of the network. We define a filter radius for the NTK, which plays a similar role as the classical filter size and discuss how it is affected by the choice of embedding, activation function, depth and other hyperparameters like the importance of bias in the network. This tradeoff can also be analysed in terms of the spectrum of the NTK, explaining why neural networks naturally avoid checkerboard patterns.

We confirm and illustrate these theoretical observations with numerical experiments. Our implementation of the algorithm will be made public at `https://github.com/benjiDupuis/DeepTopo`.

6.2 Presentation of the method

In this paper, we use a DNN to generate the density field used by the Solid Isotropic Material Penalisation (SIMP) method. Our implementation of SIMP is based on [3] and [143]. In this section we introduce the traditional SIMP method and our neural network setting.

SIMP method

We consider a regular grid of N elements where the density of element i is denoted $y_i \in [0,1]$, informally the value y_i represents the presence of material at a point i. Our goal is to optimise over the density $y \in \mathbb{R}^N$ to obtain a shape that can withstand forces applied at certain points, represented by a vector F.

The method uses finite element analysis to define a stiffness matrix $K(y) \in S_N^{++}(\mathbb{R})$ from the density y and computes the displacement vector $U(y)$ (which represent the deformation of the shape at all points i as a result of the applied forces F) by solving a linear system $K(y)U(y) = F$. In our implementation, we performed it either by using sparse Cholesky factorisation [45, 33] or BICGSTAB method [218] (this last one can be used for a high number of pixels).

The loss function is then defined as the compliance $C(y) = U(y)^T K(y)U(y)$, under a volume constraint of the form $\sum_{i=1}^N y_i = V_0$, with $0 \leq V_0 \leq N$ (see [3, 143]).

A modified SIMP approach

Several methods exist to optimise the density field $y \in \mathbb{R}^N$, such as gradient descent or the so-called Optimality Criteria (OC) [233]. We propose here an optimisation method inspired from [94] which we will refer as the Modified Filtering method (MF). The advantage of this method is that it can be used with or without DNNs, hence allowing comparison between these two approaches. We first present here the model without DNNs.

In our method, the densities y_i^{MF} are given by:

$$\forall i \in \{1, ..., N\},\ y_i^{\text{MF}} = \sigma(x_i + \bar{b}(X)), \quad \text{with } \bar{b}(X) \text{ such that } \sum_{i=1}^{N} y_i^{\text{MF}} = V_0, \tag{6.2.1}$$

for $X = (x_1, ..., x_N) \in \mathbb{R}^N$ and the sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$. We will denote this operation as: $Y^{\text{MF}} = \Sigma(X)$. The sigmoid ensures that densities are in $[0,1]$ and the choice of the optimal bias $\bar{b}(X)$ ensures that the volume constraint is satisfied.

Filtering: If the vector X is optimised directly with gradient descent, SIMP often converges toward checkerboard patterns, i.e. some high frequency noise in the image, which is a common issue with SIMP [3]. To overcome this issue a common technique is to use filtering [202]. In this paper, we consider low-pass density filters of the form: $X = T\bar{X}$ where T represents a convolution on the grid, $\bar{X}$ are the design variables and X is the vector in equation 6.2.1. The loss function of this method is then naturally defined as: $\bar{X} \longmapsto C(\Sigma(T\bar{X}))$.

The gradient $\nabla_Y C$ is easily obtained by the self-adjointness of the variational problem [233, 110]. We recover $\nabla_X C$ from $\nabla_Y C$ using an implicit differentiation technique [79]. The following proposition is a consequence of implicit function theorem and chain rules:

Proposition 6.1. *Let $\dot{S}$ be the vector with entries $\dot{\sigma}(x_i + \bar{b}(X))$. We have $\nabla_X C = D_X \nabla_Y C$ with:*

$$D_X := -\frac{1}{|\dot{S}|_1} \dot{S}\dot{S}^T + Diag(\dot{S}). \tag{6.2.2}$$

where $|.|_1$ denotes the l^1 norm of a vector. Furthermore D_X is a symmetric positive semi-definite matrix whose null-space is the space of constant vectors and has eigenvalues smaller than $\frac{1}{4}$.

Proposed algorithm: SIMP with Neural networks

Fully-Connected Neural Networks (FCNN) are characterised by the number of layers $L+1$, the numbers of neurons in each layer $(n_0, n_1, ..., n_L)$ and an activation function $\mu : \mathbb{R} \longrightarrow \mathbb{R}$, here we will use the particular case $n_L = 1$. The activations $a^l \in \mathbb{R}^{n_l}$ and preactivations $\tilde{a}^l \in \mathbb{R}^{n_l}$ are defined recursively for all layers l, using the so-called NTK parameterisation [105]:

$$a^0(x) = x, \quad \tilde{a}^{l+1}(x) = \frac{\alpha}{\sqrt{n_l}} W^l a^l(x) + \beta b^l, \quad a^{l+1}(x) = \mu\big(\tilde{a}^{l+1}(x)\big), \tag{6.2.3}$$

for some hyperparameters $\alpha, \beta \in [0,1]$ representing the contribution of the weights and bias terms respectively. The parameters $\theta = (\theta_p)_p$, consisting in weight matrices W^l and bias vectors b^l are drawn as i.i.d. standard normal random variables $\mathcal{N}(0,1)$. We denote the output of the network as $f_\theta(x) = \tilde{a}^L(x)$.

Remark: To ensure that the variance of the neurons at initialization is the equal to 1 at all layers, we choose α and β such that $\alpha^2 + \beta^2 = 1$ and use a standardised non-linearity, i.e. $\mathbb{E}_{X\sim\mathcal{N}(0,1)}[\mu(X)^2] = 1$ ([104]).

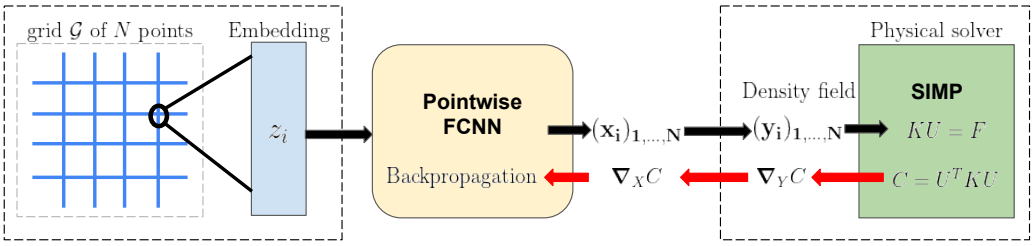


Figure 6.2.1: Illustration of our method

In our approach, the pre-densities $X^{\mathrm{NN}}(\theta) = (x_1^{\mathrm{NN}}, ..., x_N^{\mathrm{NN}})$ are generated by a neural network as $x_i^{\mathrm{NN}} = f_\theta(z_i)$ where $z_i \in \mathbb{R}^{n_0}$ is either the coordinates of the grid elements (in this case $n_0 = d$) or an embedding of those coordinates. We then apply the same transformation Σ to obtain the density field $Y^{\mathrm{NN}}(\theta) = \Sigma(X^{\mathrm{NN}}(\theta))$. Our loss function is then defined as:

$$\theta \longmapsto C(Y^{\mathrm{NN}}(\theta)) = C\big(\Sigma(X(\theta))\big).$$

The design variables are now the parameters θ of the network. The gradient $\nabla_\theta C$ w.r.t. to the parameters is computed by first using Proposition F.1 to get $\nabla_{Y^{\mathrm{NN}}} C$ followed by traditional backpropagation.

Remark: Note the absence of filter T in the above equations, indeed we will show how neural networks naturally avoid checkerboard patterns, making the use of filtering obsolete.

Initial density field: The SIMP method is usually initialised with a constant density field [3]. Since the neural network is initialized randomly, the initial density field is random and non-constant. To avoid this problem, we subtract the initial density field and add a well-chosen constant:

$$\forall i \in \{1, ..., N\},\ x_i(\theta) = \bar{f}_{\theta(t)}(z_i) = f_{\theta(t)}(z_i) - f_{\theta(t=0)}(z_i) + \log\left(\frac{V_0}{N - V_0}\right). \tag{6.2.4}$$

We used equation 6.2.4 to compute $X(\theta)$ in our numerical experiments.

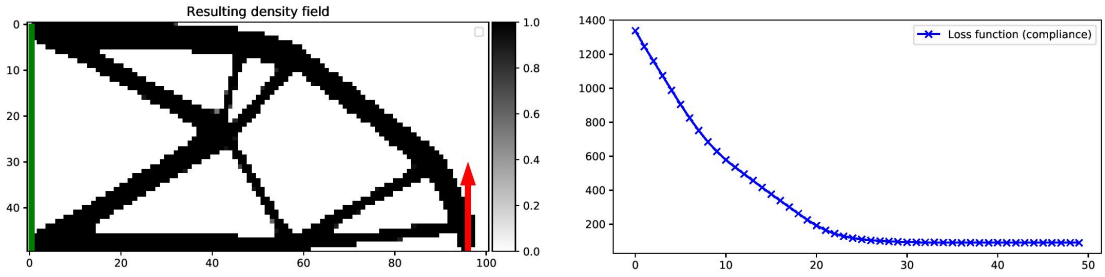


Figure 6.2.2: Example of result of our method with applied forces (red arrow) and a fixed boundary (green). Here we used a Gaussian embedding (see section 4 for details).

6.3 Theoretical Analysis

Analogy between the Neural Tangent Kernel and filtering techniques

In our paper, we use the Neural Tangent Kernel (NTK [105]) as the main tool to analyse the training behaviour of the FCNN. In our setting (where $n_L = 1$) the NTK is defined as:

$$\forall z, z' \in \mathbb{R}^{n_0},\ \Theta_\theta^L(z, z') = \sum_p \frac{\partial f_\theta}{\partial \theta_p}(z)\frac{\partial f_\theta}{\partial \theta_p}(z') = (\nabla_\theta f_\theta(z)|\nabla_\theta f_\theta(z')).$$

This is a positive semi-definite kernel. Given some inputs $z_1, ..., z_N$ we define the NTK Gram matrix as: $\tilde{\Theta}_\theta^L := \left(\Theta^L(z_i, z_j)\right)_{1\leq i,j\leq N} \in \mathbb{R}^{N\times N}$.

Assuming a small enough learning rate, the evolution of the network under gradient descent is well approximated by the gradient flow dynamics $\partial_t\theta(t) = -\nabla_\theta C(t)$. The evolution of the output of the network $X^{\mathrm{NN}}(\theta)$ can then easily be expressed in terms of the NTK Gram matrix [104] for a loss $\mathcal{L}$:

$$\partial_t X^{\mathrm{NN}}(\theta(t)) = -\tilde{\Theta}_{\theta(t)}^L \nabla_{X^{\mathrm{NN}}}\mathcal{L}.$$

From this equation we can derive the evolution of the physical density field Y^{NN} in our algorithm:

Proposition 6.2. *If the network is trained under this gradient flow, then by applying chain rules, we can prove that the density field follows the equation:*

$$\partial_t Y^{NN}(\theta(t)) = -D_X(t)\tilde{\Theta}_{\theta(t)}^L D_X(t)\nabla_Y C(Y^{NN}(\theta(t))). \tag{6.3.1}$$

The analogy between the NTK and filtering techniques comes from the following observation. With Modified Filtering with a filter T, we show similarly that the density field Y^{MF} evolves as

$$\partial_t Y^{\mathrm{MF}}(t) = -D_X(t)TT^T D_X(t)\nabla_Y C(Y^{\mathrm{MF}}(t)). \tag{6.3.2}$$

We see that the NTK Gram matrix and the squared filter TT^T play exactly the same role. An important difference however is that the NTK is random at initialisation and evolves during training.

This difference disappears for large widths (when $n_1, \ldots, n_{L-1}$ are large), since the NTK converges to a deterministic and time independent limit $\tilde{\Theta}_\infty^L$ as $n_1, \ldots, n_{L-1} \to \infty$ [105]. Furthermore,

in contrast to the finite width NTK (also called empirical NTK), we have access to a closed form formula for the limiting NTK $\tilde{\Theta}^L_\infty$ (given in the appendix).

In the infinite width limit, the evolution of the physical densities is then expressed in terms of the limiting NTK Gram matrix $\tilde{\Theta}^L_\infty$:

$$\partial_t Y^{\mathrm{NN}}(\theta(t)) = -D_X(t)\tilde{\Theta}^L_\infty D_X(t)\nabla_Y C(Y^{\mathrm{NN}}(\theta(t))). \tag{6.3.3}$$

From now on we will focus on this infinite-width limit, comparing the NTK Gram matrix $\tilde{\Theta}^L_\infty$ and the squared filter TT^T. Recent results [128, 6, 95] suggest that this limit is a good approximation when the width of the network is sufficiently large. For more details see the appendix, where we compare the empirical NTK with its limiting one and plot its evolution in our setting.

Spatial invariance

In physical problems such as topology optimisation, it is important to ensure that certain physical properties are respected by the model. We focus in this section on the translation and rotation invariance of topology optimisation: if the force constraints are rotated or translated, the resulting shape should remain the same (up to rotation and translation), as in Figure 6.3.2 (b.1 and b.2).

In Modified Filtering method, this property is guaranteed if the filter T is translation and rotation invariant. In contrast the limiting NTK is in general invariant under rotation [105] but not translation. As Figure 6.3.2 shows, this leads to some problematic artifacts. The NTK can be made translation and rotation invariant by first applying an embedding $\varphi : \mathbb{R}^d \longrightarrow \mathbb{R}^{n_0}$ with the properties that for any two coordinates p, p', $\varphi(p)^T\varphi(p')$ only depends on the distance $\|p - p'\|_2$. Since the rotation invariance of the NTK implies that $\Theta^L_\infty(z, z')$ depends only on the scalar products $z^T z'$, zz^T and $z'z'^T$, we have that $\Theta^L_\infty(\varphi(p), \varphi(p'))$ depends only on $\|p - p'\|$ as needed.

The issue is that for finite n_0 there is no non-trivial embedding φ with this property:

Proposition 6.3. *Let $\varphi : \mathbb{R}^d \to \mathbb{R}^{n_0}$ for $d > 2$ and any finite n_0. If φ satisfies $\varphi(x)^T\varphi(x') = K(\|x - x'\|)$ for some continuous function K then both φ and K are constant.*

To overcome this issue, we present two approaches to approximate spatial invariance with finite embeddings: an embedding on a (hyper)-torus and a random feature [176] embedding based on Bochner theorem [186].

Embedding on a hypertorus

In this subsection we consider the following embedding of a $n_x \times n_y$ regular grid on a torus:

$$\mathbb{R}^2 \ni p = (p_1, p_2) \longmapsto \varphi(p) = r(\cos(\delta p_1), \sin(\delta p_1), \cos(\delta p_2), \sin(\delta p_2)), \tag{6.3.4}$$

where $\delta > 0$ is a discretisation angle (our default choice is $\delta = \frac{\pi}{2\max(n_x, n_y)}$). One can use similar formulas for $d > 2$ (leading to an hyper-torus embedding), we used $d = 2$ in equation 6.3.4 for simplicity.

This embedding leads to an exact translation invariance and an approximate rotation invariance:

$$\varphi(p)^T\varphi(p') = r^2(\cos(\delta(p_1 - p'_1)) + \cos(\delta(p_2 - p'_2))) = r^2\left(2 - \frac{\delta^2}{2}\|p - p'\|_2^2\right) + \mathcal{O}\big(\delta^4\|p - p'\|_4^4\big).$$

As a result, the limiting NTK $\Theta_\infty(\varphi(p), \varphi(p'))$ is translation invariant and approximately rotation invariant (for small δ and/or when p, p' are close to each other). Moreover, if we look at the

limiting NTK on the whole torus, we obtain that the gram matrix $\tilde{\Theta}_\infty$ is a discrete convolution on the input grid, with nice properties summed up in the following proposition:

Proposition 6.4. *We can always extend our $n_x \times n_y$ grid and choose δ such that the embedded grid covers the whole torus (typically $\delta = \frac{\pi}{2\max(n_x,n_y)}$ and take a $n \times n$ grid with $n = 4\max(n_x, n_y)$). Then the Gram matrix $\tilde{\Theta}_\infty$ of the limiting NTK is a 2D discrete convolution matrix. Moreover the NTK Gram matrix has a positive definite square root $\sqrt{\tilde{\Theta}_\infty}$ which is also a discrete convolution matrix.*

As we know, the eigenvectors of such a convolution matrix are the 2D Fourier vectors. The corresponding eigenvalues are the discrete Fourier transforms of the convolution kernel.

The square root of the NTK Gram matrix $\sqrt{\tilde{\Theta}_\infty}$ then corresponds to the filtering matrix T in our analogy. Figure 6.3.1 shows that on the full torus, the matrix square root $\sqrt{\tilde{\Theta}_\theta}$ indeed looks like a typical smoothing filter.

As Figure 6.3.2 shows, the torus embedding method gives good numerical results and respect the symmetry of the applied forces F.

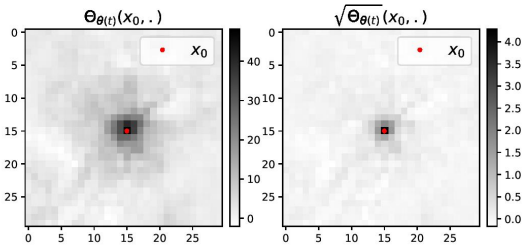


Figure 6.3.1: Representation of one line of $\tilde{\Theta}_\theta$ on the full torus and of its square root. We used $\beta = 0.2$ and $\omega = 3$ (see Section 6.4) here to make the filter visible on the whole torus.

Random embeddings for radial kernels

Another approach to approximate a rotation and translation invariant embedding is to use random Fourier features [176], which is a general method to approximate shift invariant kernels of the form $k(x, y) = k(x - y)$. By Bochner theorem [186], any continuous non-zero radial kernel $k(x - y) = K(\|x - y\|)$ can be written as the the (scaled) Fourier transform of a probability measure $\mathbb{Q}$ on $\mathbb{R}^d$:

$$k(r) = k(0) \int_{\mathbb{R}^d} e^{i\omega.r} d\mathbb{Q}(\omega).$$

For radial kernels, we formulate random Fourier features embeddings $\varphi : \mathbb{R}^d \to \mathbb{R}^{n_0}$ as follows:

$$\varphi(p)_i = \sqrt{2k(0)} \sin(w_i^T p + \frac{\pi}{4} + b_i),$$

for i.i.d. samples $w_1, ..., w_{n_0} \in \mathbb{R}^d$ from $\mathbb{Q}$ (which is also invariant by rotation) and i.i.d. samples $b_1, ..., b_{n_0} \in \mathbb{R}^d$ from any symmetric probability distribution (or uniform laws on $[0, 2\pi]$). By the law of large numbers for large n_0, we have the approximation $\frac{1}{n_0}\varphi(p)^T\varphi(p') \simeq k(p - p')$.

Gaussian embedding: Depending on the kernel k that we want to approximate, it may be difficult to sample from the distribution $\mathbb{Q}$. The simplest case is for a Gaussian kernel $k(d) = e^{-\frac{1}{2\ell^2}d^2}$, where the distribution $\mathbb{Q}$ of the weights w_i is $\mathcal{N}(0, \frac{1}{\ell^2} I_d)$, i.e. the entries w_{ij} are all i.i.d. $\mathcal{N}(0, \frac{1}{\ell^2})$ Gaussians. For this reason this is the embedding that we will use in our numerical experiments. Note the similarity between this type of embedding and an untrained first layer of a FCNN with sine activation function, weights w_i and bias b_i.

Moreover, the following result shows that we can still define a "square root" of the NTK with those types of embedding and thus complete the analogy with equation 6.3.2.

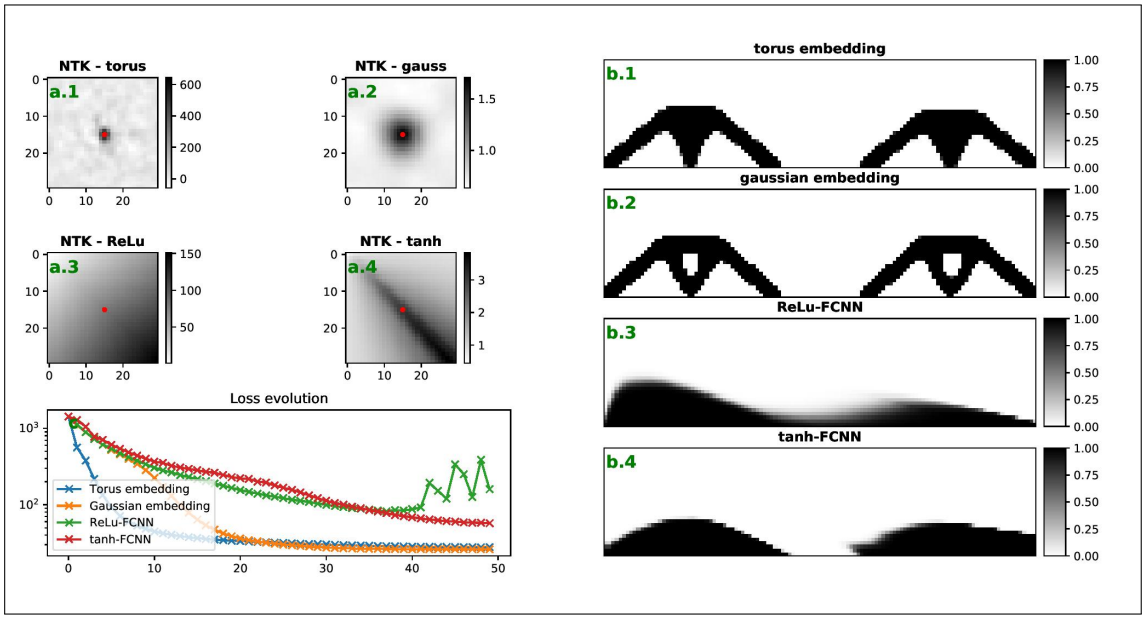


Figure 6.3.2: Left: empirical NTK of FCNNs with both embedding (a.1, a.2, see Section 6.4 for details) or without embedding (a.3 with ReLu, a.4 with tanh). Right: Corresponding shape obtained after training. Note that methods without spatial invariance particularly struggles with this symmetric load case (b.3, b.4) while both "embedded methods" respect the symmetry (b.1, b.2). We also observed that training with non-embedded methods is very unstable

Proposition 6.5. *Let φ be an embedding as described above for a positive radial kernel $k \in L^1(\mathbb{R}^d)$ with $k(0) = 1$, $k \geq 0$. Then there is a filter function $g : \mathbb{R} \to \mathbb{R}$ and a constant C such that for all p, p':*

$$\lim_{n_0 \to \infty} \Theta_\infty(\varphi(p), \varphi(p')) = C + (g \star g)(p - p'), \tag{6.3.5}$$

where Θ_∞ is the limiting NTK of a network with a Lipschitz, non-constant and standardised activation function μ. (Here $\star$ denotes the convolution product).

As the matrix D_X in equation 6.3.3 cancels out the constant frequency (proposition F.1), the constant C doesn't matter, i.e. $D_X \tilde{\Theta}_\infty^{(L)} D_X = D_X \left(\tilde{\Theta}_\infty^{(L)} - C \right) D_X$.

6.4 Experimental analysis

Setup

Most of our experiments were conducted with a torus embedding or a gaussian embedding. For the SIMP algorithm, we adapted the code described in [3, 143]. Here are the hyperparameters used in the experiments.

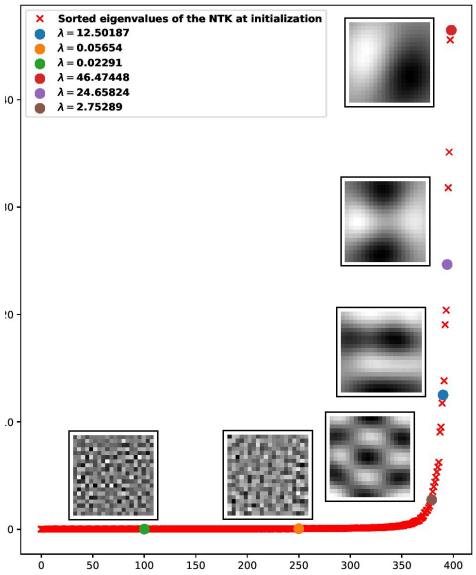


Figure 6.4.1: Sorted eigenvalues of the empirical NTK with some eigenvectors (reshaped as images). Obtained with a Gaussian embedding.

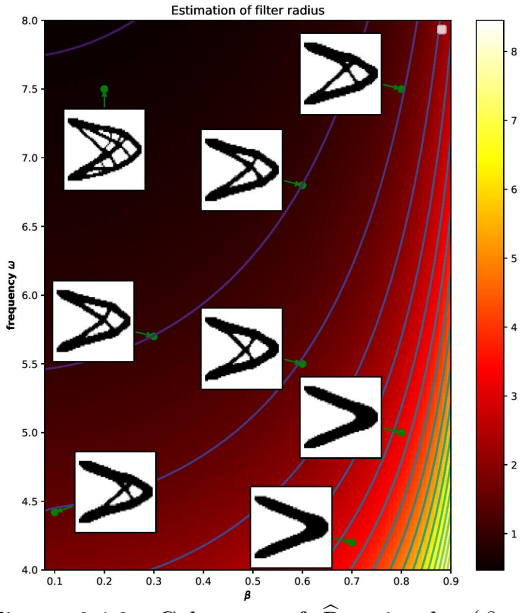


Figure 6.4.2: Colormap of $\widehat{R}_{1/2}$ in the (β, ω) plane, torus embedding. Level lines and shapes obtained for different radius are represented.

For the Gaussian embedding, we used $n_0 = 1000$ and a length scale $\ell = 4$. This embedding was followed by one hidden linear layer of size 1000 with standardized ReLu ($x \mapsto \sqrt{2}\max(0, x)$) and a bias parameter $\beta = 0.5$.

For the torus embedding we set the torus radius to $r = \sqrt{2}$ (to be on a standard sphere) and the discretisation angle to $\delta = \frac{\pi}{2\max(n_x, n_y)}$ (to cover roughly half the torus, which is a good trade-off between rotation invariance and kernel size), where $n_x \times n_y$ is the size of the grid. It was followed by 2 linear layers of size 1000 with $\beta = 0.1$. The ReLu activation is not well-suited in this case because it induces filters that are too wide. The large radius of the NTK kernel can be understood in relation with the order/chaos regimes [197, 173], as observed in [104] the ReLU lies in the ordered regime when $\beta > 0$, leading to a "wide" kernel, a narrower kernel can be achieved with non-linearities which lie in the chaotic regime instead. We used a cosine activation of the form $x \mapsto \cos(\omega x)$, which has the advantage that the width of the filter can be adjusted using the ω hyperparameter, see Section 6.4. When not stated otherwise we used $\omega = 5$.

Even though our theoretical analysis is for gradient flow, we obtain similar results with other optimizers such as RPROP [179] (learning rate 10^{-3}) and ADAM [116] (learning rate 10^{-3}). RPROP gave the fastest results, possibly because it is well-suited for batch learning [180]. Vanilla gradient descent can be very slow due to the vanishing of the gradients when the image becomes almost binary (due to the sigmoid), we therefore gradually increased the learning rate during training to compensate.

Spectral analysis

In SIMP convolution with a low pass filter ensures that low frequencies are optimised faster than high frequencies, to avoid checkerboards.

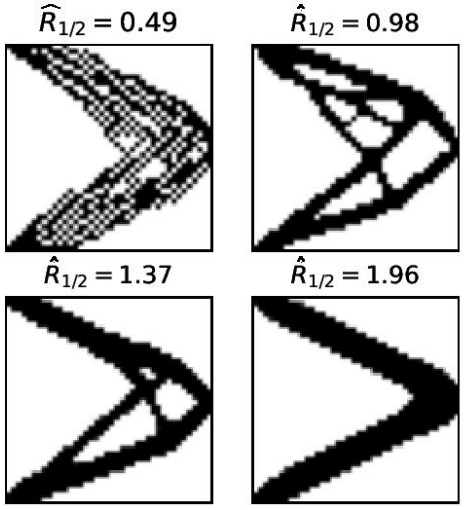


Figure 6.4.3: Shape obtained for different values of $\widehat{R}_{1/2}$ with a Gaussian embedding for different values of $\ell \in \{0.5, 1, 1.4, 2\}$.

With the embeddings proposed in the last two subsections, the limiting NTK takes the form of a convolution over the input space $\mathbb{R}^d$. Figure 6.4.1 represents the eigenvalues and eigenimages of the NTK Gram matrix $\tilde{\Theta}_{\theta(t)}$. Even though this plot is done for a finite width network and a finite random embedding, we see that the eigenimages look like 2D Fourier modes. The fact that the low frequencies have the largest eigenvalues supports the similarity between the NTK and a low pass filter.

This may explain why neural networks naturally avoid checkerboard patterns: the low frequencies of the shape are trained faster than the high frequencies which lead to checkerboard patterns.

Filter radius

In the classical SIMP algorithm, the choice of the radius of the filter T is critical. It controls the appearance of checkerboards or intermediate densities.

When using DNNs, there is no explicit choice of filter radius, since the filter depends on the embedding and the architecture of the network. In Section 6.3 we have shown that the NTK is approximately invariant, it can hence be expressed as:

$$\Theta^L_{\theta(t)}(\varphi(p), \varphi(p')) \simeq \Phi_\infty(\|p - p'\|),$$

where Φ_∞ can be analytically expressed with the embedding and the limiting NTK (see appendix for a detailed example).

The kernels we consider do not have compact support in general, we therefore focus instead on the radius at half-maximum of Φ_∞:

$$\Phi_\infty(\widehat{R}_{1/2}) = \frac{1}{2}\big(\Phi_\infty(0) + \inf_r \Phi_\infty(r)\big).$$

Note that for simplicity we are computing here the radius of the squared filter, since obtaining a closed form formula for the square root of the NTK is more difficult. For Gaussian filters the radius of the squared filter is $\sqrt{2}$ times that of the original, suggesting that the filter radius is well estimated by $\frac{1}{\sqrt{2}}\widehat{R}_{1/2}$.

The quantity $\widehat{R}_{1/2}$ is a function of the hyperparameters of the network (α,β, L, see appendix) and of the embedding (the lengthscale ℓ). Using the formula for $\widehat{R}_{1/2}$, these hyperparameters can be tuned to obtain a specific filter radius.

With the Gaussian embedding, the radius of the filter can easily be adjusted by changing the length-scale ℓ of the embedding. As illustrated in Figure 6.4.

With the torus embedding, we instead have to change the hyperparameters of the network to adjust the radius of the filter. With the ReLU activation function, the radius is very large which makes it impossible to obtain precise shape. The solution we found is to use a cosine activation $x \mapsto \cos(\omega x)$ with hyper-parameter ω. Figure 6.4.2 shows how the radius decreases as ω increases. The β parameter has the opposite effect, as increasing it increases the radius. For different values of ω and β, we obtain a variety of radius and plot the resulting shapes. This plot also illustrates the role of the radius in the determination of the resulting shape. The fact that cosine activation leads to an adjustable NTK radius could explain why periodic activation function help in the representation of high frequency signal as observed in [206].

The effect of depth is more complex. For large depths L the NTK either approaches a constant kernel in the so-called order regime (with infinite radius) or a Kronecker delta kernel in the so-called chaos regime (with zero radius) [173, 197, 104]. Depending on whether we are in the order or chaos regime (which is determined by the activation function μ and the parameters α, β), increasing the depth can either increase or decrease the radius.

We conducted an experimental study of the influence of this parameter on the geometry of the final shape. We observed that its complexity (number of holes, high frequencies) is highly controlled by $\widehat{R}_{1/2}$. We see in Figure 6.4 and 6.4.2 some examples of shape obtained for several values of $\widehat{R}_{1/2}$.

Up-sampling

Since the density field is generated by a DNN, it can be evaluated at any point in $\mathbb{R}^d$, hence allowing upsampling. As Figure 6.4.4 shows, with our method we obtain a smooth and binary shape. Something interesting happens when the network is trained without an embedding: when upsampling we observe some visual artifacts plotted in Figure 6.4.5. We believe that it is due to the lack of spatial invariance.

Note that this second experiment was done with batch norm, as described in [31], since for this problem it was difficult to obtain a good shape with a vanilla ReLU-FCNN. With our embeddings, we can achieve complex shapes without batch-norm.

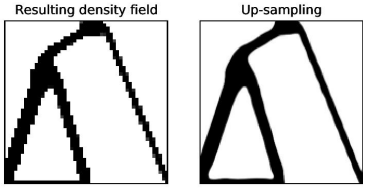


Figure 6.4.4: Density field obtained with a Torus embedding (left) and up sampling of factor 6 of the same network (right).

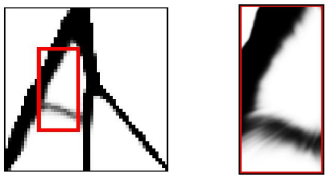


Figure 6.4.5: Exemple of up-sampling of a FCNN (ReLu FCNN with batchnorms) without embedding, exhibing typical visual artifacts.

6.5 Conclusion

Using the NTK, we were able to give a simple theoretical description of topology optimisation with DNNs, showing a similarity to traditional filtering techniques. This theory allowed us to identify a problem: since the NTK is not translation invariant, the spatial invariance of topology optimisation

is not respected, leading to visual artifacts and non-optimal shapes. We propose a simple solution to this problem: adding a spatial invariant embedding to the coordinates before the DNN.

Using this method, our models are able to learn efficient shapes while avoiding checkerboard patterns. We give tools to adjust the implicit filter size induced by the hyperparameters, to give control over the complexity of the final shape. Using the learned network, we can easily perform good quality up-sampling. The techniques described in this paper can easily be translated to any other problem where spatial invariance is needed.

The NTK is a simple yet powerful tool to analyse a practical method such as SIMP when combined with a DNN. Morover it can be used to make informed choices of the DNN's architecture and hyperparameters.

Chapter 7

Scaling Description of Generalization with Numer of Parameters in Deep Learning

Abstract

We provide a description for the evolution of the generalization performance of fixed-depth fully-connected deep neural networks, as a function of their number of parameters N. As N gets large, we observe that increasing N at fixed depth reduces the fluctuations of the output function f_N induced by initial conditions, with $\|f_N - \bar{f}_N\| \sim N^{-1/4}$ where $\bar{f}_N$ denotes an average over initial conditions. We explain this asymptotic behavior in terms of the fluctuations of the so-called Neural Tangent Kernel that controls the dynamics of the output function. For the task of classification, we predict these fluctuations to increase the true test error ϵ as $\epsilon_N - \epsilon_\infty \sim N^{-1/2} + \mathcal{O}(N^{-3/4})$. This prediction is consistent with our empirical results on the MNIST dataset, and it explains in a concrete case the puzzling observation that the predictive power of deep networks improves as the number of fitting parameters grows. For smaller N, this asymptotic description breaks down at a so-called jamming transition which takes place at a critical $N = N^*$, below which the training error is non-zero. In the absence of regularization, we observe an apparent divergence $\|f_N\| \sim (N - N^*)^{-\alpha}$ and provide a simple argument suggesting $\alpha = 1$, consistent with empirical observations. This result leads to a plausible explanation for the cusp in test error known to occur at N^*. Overall, our analysis suggests that once models are averaged, the optimal model complexity is reached just beyond the point where the data can be perfectly fitted, a result of practical importance that needs to be tested in a wide range of architectures and dataset.

7.1 Introduction

Deep neural networks are very successful at various tasks including image classification [119, 122] and speech recognition [89]. Yet, understanding why they work remains a challenge, and central questions need to be clarified.

- First, learning amounts to a descent in a high-dimensional loss landscape, which is a priori non-convex. What guarantees then that the dynamics does not get stuck in a poor minimum of the loss, leading to bad generalization?

- Second, deep networks tend to work in the over-parametrized regime where the number of parameters N can be (much) larger than the number of data points P which are used to optimize them. Thus, they are used in a regime where their capacity is very large (in the sense that they can still classify data even if the labels are randomized [236]), yet they generalize very well, at odds with the traditional VC-dimension learning theory.

Recent works suggest that these two questions are closely connected. Numerical and theoretical studies [63, 216, 92, 208, 39, 189, 190, 14, 141, 12, 68, 211] support that in the over-parametrized regime the loss landscape is not rough with isolated minima (as initially thought in [44, 38]), but instead has connected level sets and presents many flat directions, even near its bottom. When optimizing neural networks using the hinge loss, there is a sharp phase transition (similar to the jamming transition that occurs in granular materials [101]) at some $N^*(P)$ such that for $N \geq N^*$ the dynamic process reaches global minima of the loss [68, 211]. In short, cranking up N guarantees low *training error*. A counter-intuitive aspect of deep learning is that increasing N above N^* does not destroy the predictive power by over-fitting the data, but instead appears to improve the *generalization performance* [161, 160, 16, 1]. Indeed the test error is observed to decrease in a slow power-law fashion [211] toward a limit as $N \to \infty$. Such a monotonic improvement is observed everywhere except near N^*, where the test error displays a cusp [1, 139, 211] (phenomena shown by the blue curve in Fig.7.2.1).

Explaining this observed dependence of the generalization on N in deep networks remains a challenge. In the perceptron, the simplest network without hidden layers, the cusp in the test error at the jamming point is also observed and predicted analytically [188, 57, 26, 120, 60, 59]. For deep linear networks trained with the mean-square loss, this cusp corresponds to an explosion of the norm of the output function precisely at $N = P$ [1, 139]. Yet, what controls its presence in non-linear deep networks that are trained with a descent dynamics is unclear.

Another open question regards the asymptotic improvement of generalization performance with N — a phenomenon that does not happen for perceptrons. Very recently, in the context of least-squares regression, this behavior was linked to the observed diminishing fluctuations of the output function with N [158], a result that is consistent with the notion of stronger implicit regularization with increasing N [209, 137]. Yet, what controls these fluctuations in deep non-linear networks and how they affect the test error in a classification task is not yet clear.

In this work, we address these questions using the recent discovery that in the limit $N \to \infty$, some deep learning models (in particular, fully-connected networks with any depth and a large class of non-linear functions that include the most common ones used in practice) are equivalent to a kernel method, where the kernel (coined the Neural Tangent Kernel or NTK) becomes deterministic and fixed at any finite time during training [105]. This result explains why generalization performance converges to a finite value as $N \to \infty$ (such a result has previously been obtained for single hidden layer neural networks in [34, 184, 153, 205] under a different scaling limit[1]). Here, we use this framework to study the variation of the output function f_N at the end of training. For a fixed algorithm, such variations are induced by the initial conditions. These variations still exist asymptotically for f_∞ [105], yet for a large dataset, this effect appears to be subdominant even for the largest N we can reach. Departing from the $N \to \infty$ limit has two consequences. First, at finite N, the NTK will display a nonzero evolution in time, leading to a systematic difference between f_N and f_∞. This effect on the performance is perceptible but small. Secondly, the NTK at

[1] Weights of a layer of width h are initialized at scale $1/h$ as opposed to the usual $1/\sqrt{h}$.

initialisation has fluctuations around its mean that are of order $N^{-1/4}$, leading to similar variations for f_N which turn out to be dominant.

Next, by considering the decision boundary, we argue that a variation in f of order δf increases the true test error by $\delta\epsilon \sim (\delta f)^2$. We use this asymptotic result to predict (i) the increase in generalization performance obtained by ensemble averaging on n samples of the function f_N as n becomes large and (ii) the increase in generalization performance with N at fixed network depth. This description breaks down at the transition point N^*, where variations in f_N appear to diverge as a powerlaw, justifying the non-analyticity in the training error. We rationalize this divergence with a simple argument on a non-linear network trained with the hinge loss, that leads to $\|f_N\| \sim (N - N^*)^{-1}$. Overall, our work introduces a conceptual framework to describe how generalization error in deep learning evolves with the number of parameters. As an application, we demonstrate how ensemble averaging removes variations in the predictor and enhance generalization. Our result suggests that near-optimal generalization can be obtained by ensemble averaging networks that are slightly larger than N^*.

7.2 Improving generalization by averaging in MNIST

In this section we show how ensemble averaging improves generalization in networks that are slightly larger than the jamming transition N^*. We adopt the experimental setup of [211]. The task is to classify MNIST digits depending on their parity, where the standardized MNIST inputs are reduced to 10 dimensions using their first 10 PCA components (this reduction has been introduced to have a number of weights in the first layer comparable to the ones of the other layers). The architecture is a fully-connected network with L layers, where each of the layers has h nodes. The non-linearity at each hidden layer is the standard rectified linear unit (ReLU). Weights of the network are initialized according to the random orthogonal scheme [195] and all biases are initialized to zero. The network is optimized using ADAM [117] with full batch and the learning rate is set to $\lambda = \min(10^{-1}h^{-1.5}, 10^{-4})$ in order to have a smooth dynamics for all values of h[2].

The network parametrizes an output function $f(x;\theta)$ with some parameters θ — the network's weights and biases. The binary classification task consists in searching the parameters such that $\mathrm{sign} f(x_\mu;\theta) = y_\mu$ for all MNIST images x_μ, where $y_\mu = \pm 1$ according to the parity of the digit depicted in the input image. To do so, we minimize the square-hinge cost function

$$C = \frac{1}{P}\sum_{\mu=1}^{P}\frac{1}{2}\max(0,\Delta_\mu)^2, \tag{7.2.1}$$

where $\Delta_\mu \equiv \epsilon_m - y_\mu f(x_\mu;\theta)$. In all of our simulations we fix the margin ϵ_m to 1. The training process runs for a maximum of $2\cdot 10^6$ steps[3]. Typically, above jamming the training halts earlier as soon as the training reaches zero loss, while below jamming, training stalls at points with non-zero loss.

Our results, shown in Fig.7.2.1, demonstrate that after learning, the test error has a peak near the transition at N^* and then it slowly decreases as N becomes larger. We denote by $\bar{f}_N^n$ the average of the function f_N over n different initial conditions. Remarkably, in our experiments ensemble-averaging over $n = 20$ independent runs led to a nearly flat test error for $N > N^*$; this

[2]The exponent -1.5 has been empirically chosen so that the number of steps to converge is independent of h [105].

[3]Note that the number of steps and the number of epochs are the same in this setup.

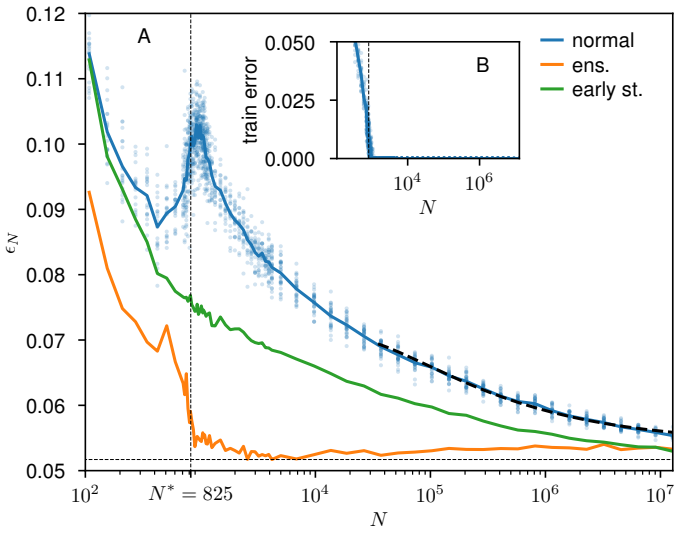


Figure 7.2.1: (A) Empirical test error *v.s.* number of parameters: average curve (blue, averaged over 20 runs); early stopping (green); ensemble average $\bar{f}_N^n$ (orange) over $n = 20$ independent runs. In all the simulations we used fully-connected networks with depth $L = 5$ and input dimension $d = 10$, trained for $t = 2 \cdot 10^6$ epochs to classify $P = 10k$ MNIST images depending on their parity, using their first 10 PCA components, and the test set includes $50K$ images. The vertical dashed line corresponds to the jamming transition: at that point the test error peaks. Ensemble averaging leads to an essentially constant behavior when N becomes larger than N^*. The location of the jamming transition, N^* shown here, is measured in section 7.6 for extrapolated $t = \infty$. Black dashed line: asymptotic prediction of the form $\epsilon_N - \epsilon_\infty = B_0 N^{-1/2} + B_1 N^{-3/4}$, with $\epsilon_\infty = 0.054$, $B_0 = 6.4$ and $B_1 = -49$. (B) Training error *v.s.* number of parameters.

supports that the improvement of generalization performance with N in this classification task originates from reduced variance of f_N when N gets large, as recently observed for mean-square regression [158]. An observation of potential practical interest is that near-optimal generalization is obtained by ensemble averaging slightly above N^*. *Thus the intuition that the most predictive and parsimonious models have just enough parameters to fit the data may indeed be correct, once one averages over differently initialized networks*[4].

[4]This observation carries over to convolutional networks, as well. We train CIFAR10 on a vanilla architecture with 3 convolutional layers with f filters at each layer and a single fully-connected layer. For each f, we train 20 models at different random initial conditions. Just after N^*, the mean accuracy is $\sim$ %66, accuracy of the ensemble averaging is $\sim$ %80, and the average accuracy of widest models we could train (which has 5 orders of parameters more) is a little bit less than $\sim$ %77.

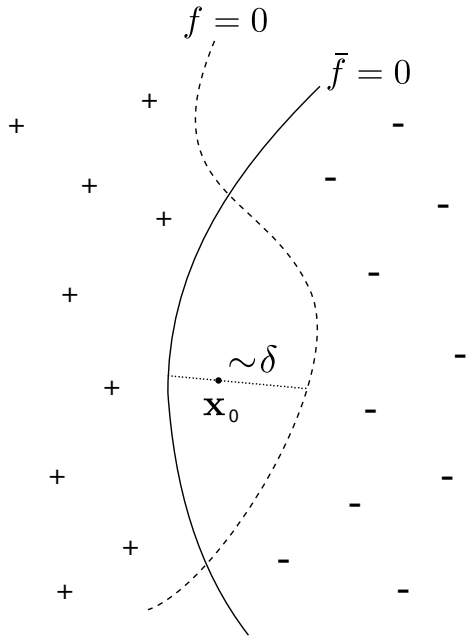


Figure 7.3.1: $f(x)$ and the limiting function $\bar{f}(x)$ (see Section 7.3) classify points according to their sign. They agree on the classification everywhere ($\pm$'s in the figure are examples where the functions are respectively both positive or both negative) except for the points that lie in between the two boundaries $f = 0$ and $\bar{f} = 0$. In the figure, let x be one such point, and δ is the typical distance from the boundary $f = 0$. In the limit where f and $\bar{f}$ are close to each other, δ is of the same order of the distance between the two boundaries.

7.3 Relationship between variance and generalization in classification tasks

We consider an ensemble of functions f that approach pointwise to a limiting function $\bar{f}$, so that $\delta f = f - \bar{f}$ satisfies $\|\delta f\|_\mu \ll \|f\|_\mu$ and $\langle \delta f \rangle = 0$, where the average is made on the ensemble considered. In what follows we define $\|f\|_\mu^2 = \int d\mu(x) f(x)^2$, where μ is some measure (empirically we shall use the uniform distribution on the training or test set or a Gaussian distribution on all x, all leading to similar results). For example, one may consider the ensemble of functions $\bar{f}_N^n = \frac{1}{n}\sum_{i=1}^n f_N^i$ obtained by varying initial conditions and averaging, or $\bar{f}_N = \lim_{n\to\infty} \bar{f}_N^n$. In this setup, the test errors of f and $\bar{f}_N$ will be denoted respectively by ϵ and $\bar{\epsilon}_N$.

Consider a specific function f, and the two decision boundaries of f and $\bar{f}$ defined as the set of inputs for which $f(x) = 0$ or $\bar{f}(x) = 0$. Consider a data point x_0 classified differently by these two functions — i.e. $f(x_0)\bar{f}(x_0) < 0$ — as illustrated in Fig.7.3.1. When the two boundaries are close enough, if the functions f and $\bar{f}$ are smooth, the signed distance $\delta(x_0)$ between the two curves near x_0 must follow $\delta(x_0) = \delta f(x_0)/\|\nabla f(x_0)\| + \mathcal{O}(\delta f(x_0)^2)$, where $\delta f(x_0) = f(x_0) - \bar{f}(x_0)$. If the non-linearity in the network is itself smooth, smoothness of the output function during learning is guaranteed both for N finite or not, as shown in [105] and discussed in Supplementary Materials (S.M.). In the case of the Relu non-linear function, we expect smoothness to hold as $N \to \infty$ except

on the P points of the training set [5]. We show direct measurements of $\delta(x)$ in Section A of S.M., supporting that this estimate still holds and become more and more accurate as $N \to \infty$.

Next we introduce the typical distance δ along the boundary:

$$\delta \equiv \langle |\delta f(x_0)| / \|\nabla f(x_0)\| \rangle_{x_o \sim \text{test} \cap \text{interface}} \tag{7.3.1}$$

where the average is made over all the test data classified differently by f and $\bar{f}$. Such conditioning, however, does not affect the average. Indeed we have checked, as shown in Appendix A, that δ is very well estimated by $\|\delta f\|_\mu / \|\nabla f\|_\mu$ where μ indicates the uniform measure on all the test set. Next, we denote by $\Delta\epsilon$ the difference between the true test error of f and that of $\bar{f}$. Under reasonable assumptions [6] it can be expanded by considering a small motion of the decision boundary B of $\bar{f}$ (that can consist of unconnected parts):

$$\Delta\epsilon = \int_B dx^{d-1} \left[\frac{\partial \epsilon}{\partial \delta(x)} \delta(x) + \frac{1}{2} \frac{\partial^2 \epsilon}{\partial^2 \delta(x)} \delta^2(x) + \mathcal{O}(\delta^3(x)) \right]. \tag{7.3.2}$$

Using that $\langle \delta(x) \rangle = \mathcal{O}(\delta f(x)^2)$ since $\langle \delta f(x) \rangle = 0$, we get that in average the true test error must increase quadratically with the norm of fluctuations δf:

$$\langle \Delta\epsilon \rangle \sim \delta^2 \sim \frac{\|\delta f\|_\mu^2}{\|\nabla f\|_\mu^2}. \tag{7.3.3}$$

Note that if the model $\bar{f}$ displays a minimal true test error, the decision boundary is optimal: $\partial\epsilon / \partial\delta(x) = 0$ and $\partial^2\epsilon / \partial^2\delta(x) \geq 0$ for all $x \in B$, implying that the prefactor in Eq.7.3.3 must be positive [7]. If the true test error is small, the decision boundary will tend to be close to the ideal one, so that the prefactor in Eq.7.3.3 will still be positive. We expect it to be the case for the MNIST model we consider for which the test error is a few percents.

Eq.7.3.3 is a result on the ensemble average of the true test error. Yet, our data in Fig.7.2.1 supports that the test error is a self-averaging quantity: the test error of a given output function (blue points) lies close to its average (blue line). Such a self-averaging behavior is expected if there are many distinct regions where δ changes sign along the decision boundary. In what follows we will always consider averaged quantities, and drop the notation $\langle\rangle$.

7.4 Asymptotic generalization as $n \to \infty$

It is now straightforward to predict how an ensemble average of n networks behaves in the limit $n \to \infty$. The central limit theorem implies $\delta f \sim 1/\sqrt{n}$ while $\|\nabla f\|$ converges to some constant value. Thus $\delta \sim 1/\sqrt{n}$ and $\bar{\epsilon}_N^n - \bar{\epsilon}_N \sim 1/n$. These predictions are confirmed in Fig.7.4.1.

7.5 Asymptotic generalization as $N \to \infty$

We now study the fluctuations of f_N^t throughout training for large networks using the Neural Tangent Kernel [105] or NTK. At initialization $t = 0$, $f_N^{t=0}$ is a random function whose limiting

[5] Indeed in that case the NTK $\Theta(x, x')$ has a cusp for $x = x'$ [105].

[6] The true test error must be a smooth function of the decision boundary if the probability distributions to find data of different labels are themselves smooth functions of the input. It is the case, for instance, if the input data have Gaussian noise.

[7] The pre-factor could be zero if the optimal boundary is degenerate, a situation that will not occur generically if the data have e.g. Gaussian noise.

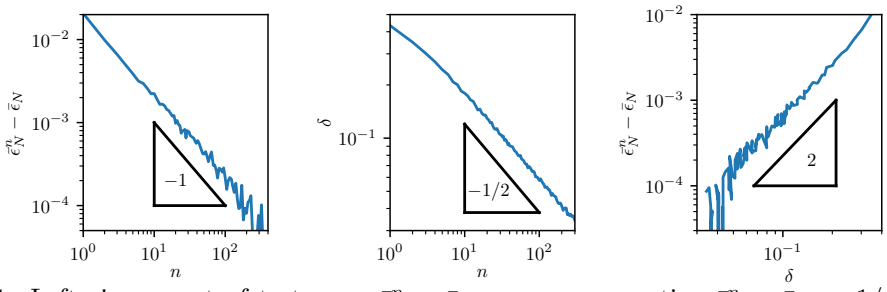


Figure 7.4.1: Left: increment of test error $\bar{\epsilon}_N^n - \bar{\epsilon}_N$ *v.s.* n, supporting $\bar{\epsilon}_N^n - \bar{\epsilon}_N \sim 1/n$. Center: δ as defined in Eq.7.3.1 *v.s.* number of average n, supporting $\delta \sim 1/\sqrt{n}$. Right: increase of test error $\bar{\epsilon}_N^n - \bar{\epsilon}_N$ as a function of the variation of the boundary decision δ, supporting the prediction $\bar{\epsilon}_N^n - \bar{\epsilon}_N \sim \delta^2$. Here $d = 30$, $h = 60$, $L = 5$, $N = 16k$ and $P = 10k$. The value $\bar{\epsilon}_N = 2.148\%$ is extracted from the fit.

distribution as $N \to \infty$ is known to be Gaussian [159, 37, 126]. These types of fluctuations do not vanish as $N \to \infty$: the variance of $f_N^{t=0}$ at initialization is essentially constant in N [8].

However, as the network is trained, the fluctuations of f_N^t will be reduced in time. We shall argue that at the end of training, the dominant source of fluctuations does not stem from the randomness of $f_N^{t=0}$, but from the randomness of the learning dynamics. To understand why the fluctuations of the function at convergence $t \to \infty$ decrease with N, we must thus study the training process. The gradient descent dynamics of f_N^t is described by a kernel, the Neural Tangent Kernel Θ_N^t:

$$\Theta_N^t(x, x') = \sum_{k=1}^{N} \frac{d}{d\theta_k} f_N^t(x) \frac{d}{d\theta_k} f_N^t(x') \tag{7.5.1}$$

where $\frac{d}{d\theta_k} f_N^t$ is the derivative of the output of the network with respect to one parameter θ_k and the sum is over all the network's parameters. For a general cost $C(f) = \frac{1}{P} \sum_i c_i(f(x_i))$, the function follows the kernel gradient $\nabla_{\Theta_N^t} C_{|f_N^t}$ of the cost during training

$$\begin{aligned} \partial_t f_N^t(x) &= -\nabla_{\Theta_N^t} C_{|f_N^t}(x) \\ &= -\frac{1}{P} \sum_i \Theta_N^t(x, x_i) c_i'(f_N^t(x_i)). \end{aligned} \tag{7.5.2}$$

The NTK is random at initialization and varies during training. However as the number h of neurons in each hidden layer goes to infinity, the NTK converges to a deterministic limit $\Theta_N^t \to \Theta_\infty$ which stays constant throughout training [105]. In this limit, deep learning simply corresponds to a kernel method, and the only randomness of f_N^t at convergence $t \to \infty$ is due to the randomness of $f_N^{t=0}$. We shall see below that this effect is subdominant in the range of parameters we probe.

Other sources of variation of f come from the variation of the kernel itself, which occurs at finite N. It varies for two reasons. First, the kernel now evolves in time, in a trajectory that depends

[8] In our setup, the output variance at initialization is smaller than one. It is possible to suppress the randomness of $f_N^{t=0}$ at initialization by training $f'^t = f^t - f^{t=0}$. We have observed that it does not qualitatively affects our results.

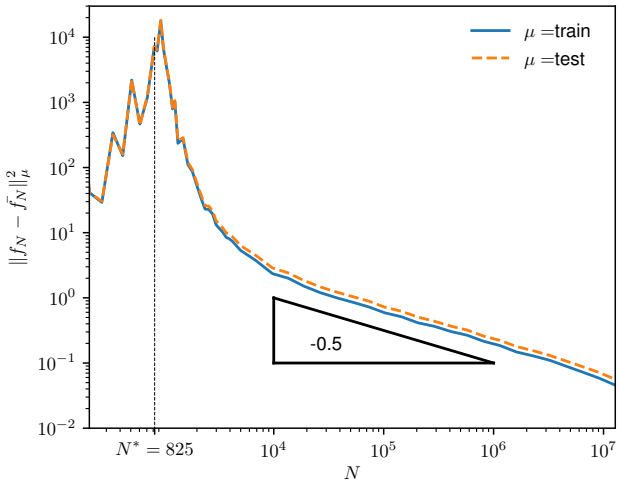


Figure 7.5.1: Variance of the output (averaged over $n = 20$ networks) *v.s.* number of parameters for different measures indicated in legend, showing a peak at jamming followed by a decay as N grows. Here $L = 5$, $d = 10$, $P = 10k$.

on initial conditions. It turns out however that this effect leads to small variations asymptotically. For any finite time T one finds [106]:

$$\left\|\Theta_N^{t=0} - \Theta_N^{t=T}\right\| = \mathcal{O}\left(\frac{1}{h}\right) = \mathcal{O}\left(N^{-1/2}\right). \tag{7.5.3}$$

Secondly, at finite N the kernel varies already at initialization:

$$\left\|\Theta_N^{t=0} - \Theta_\infty\right\| = \mathcal{O}\left(\frac{1}{\sqrt{h}}\right) = \mathcal{O}\left(N^{-1/4}\right). \tag{7.5.4}$$

The variation in Eq.7.5.4 decays much more slowly with N than that in Eq.7.5.3, and is thus expected to be the dominant source of the NTK fluctuations around Θ_∞, as supported empirically below. Eq.7.5.4 can be readily obtained by re-writing Eq.7.5.1 as a sum on neurons and using the central limit theorem, as sketched in Appendix B and derived rigorously in [106].

Because the NTK describes the behaviour of the function f_N^t during training, and because the time to converge to a minimum of the loss converges to a constant as $N \to \infty$, from Eq.7.5.2 we expect the variance of the NTK to induce some variance of the same order to the function at the end of training. We confirm it is the case for the mean square loss in Appendix C. In conclusion for large enough width, the fluctuations of the kernel leads to fluctuations of $f_N^{t=\infty}$ of order $\mathcal{O}\left(N^{-1/4}\right)$. This prediction is checked in Fig.7.5.1, supporting that $\|f_N - \bar{f}_N\| \sim N^{-1/4}$ in the range of N we can explore (eventually this curve must converge to a small constant $\|f_\infty - \bar{f}_\infty\|$, due to the finite fluctuations at initialization and the fact that we consider a large yet finite dataset [105]. In our setting observing this effect would require unreachable values of N, and we neglect this small constant). We expect that the same fluctuations that characterize f_N to also characterize ∇f_N,

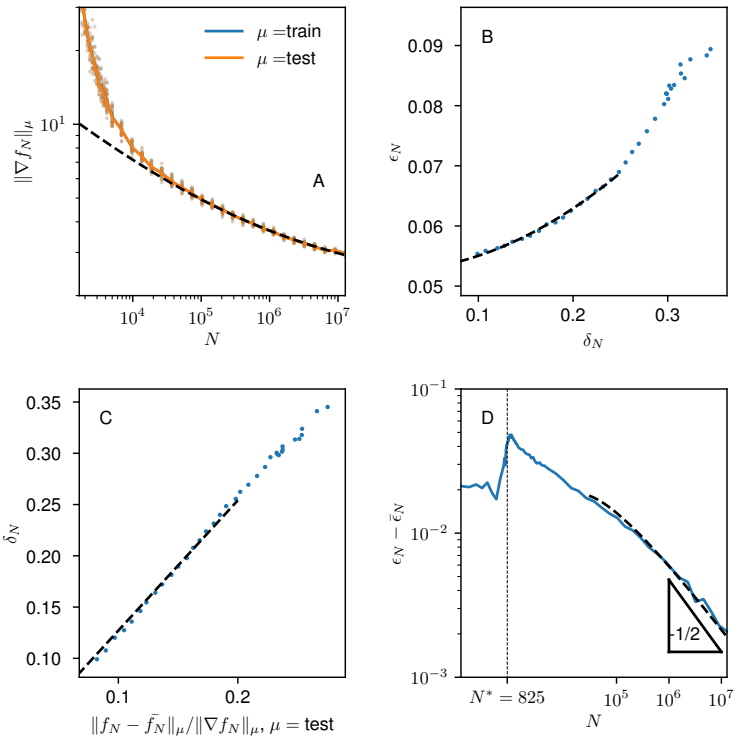


Figure 7.5.2: Here $L = 5$, $d = 10$, $P = 10k$. (A) The median of $\|\nabla f_N\|_\mu = \sqrt{\int d\mu(x) \|\nabla f_N(x)\|^2}$ over 20 runs (each appearing as a dot) is indicated as a full line. The dashed line correspond to our asymptotic prediction $\|\nabla f_N\| = C_0 + C_1 N^{-1/4}$ with $C_0 = 2.1$ and $C_1 = 51$. (B) Test error *v.s.* variation of the boundary, together with fit of the form $\epsilon_N = \epsilon_\infty + D_0 \delta_N^2$. (C) Variation of the boundary δ_N *v.s.* its estimate $\|f_N - \bar{f}_N\|/\|\nabla f_N\|$, well fitted by a linear relationship. (D) $\epsilon_N - \bar{\epsilon}_N$ *v.s.* N, with a fit of the form $\epsilon_N - \bar{\epsilon}_N = E_0 N^{-1/2} + E_1 N^{-3/4}$ with $E_0 = 7.6$ and $E_1 = -59$. If exponents in the fits are not imposed, we find for reasonable fitting ranges -0.28 instead of $-1/4$ in (A), 2.5 instead of 2 in (B), 1.1 instead of 1 in (C) and -0.42 instead of $-1/2$ in (D). Extracting exponents while also fitting for the location of the singularity, as is the case here for (A) and (B), leads to rather sloppy fits.

implying that $\|\nabla f_N\| = C_0 + C_1 N^{-1/4} + o(N^{-1/4})$. This result is consistent with our observations, as shown in Fig.7.5.2.A, in which we find empirically that C_1 is much larger than C_0. We know that $\epsilon_N - \bar{\epsilon}_N \sim \delta_N^2$ where δ_N indicates the typical distance between the decision boundaries $\bar{f}_N = 0$ and $f_N = 0$, as supported by Fig.7.5.2.B. The fluctuations of the decision boundary δ_N can be approximated as $\|f_N - \bar{f}_N\|/\|\nabla f_N\|$, as supported by Fig.7.5.2.C, leading to $\delta_N = A_0 N^{-1/4} + A_1 N^{-1/2} + o(N^{-1/2})$. We then obtain the key prediction $\epsilon_N - \bar{\epsilon}_N = B_0 N^{-1/2} + B_1 N^{-3/4}$. Since we measure both ϵ_N and $\bar{\epsilon}_N$ independently, we can test the prediction for the leading exponent without any fitting parameters, and indeed confirm that asymptotically $\epsilon_N - \bar{\epsilon}_N \sim N^{-1/2}$ as shown in Fig.7.5.2.D.

Finally we estimate the evolution of test error with N. We have:

$$\epsilon_N - \epsilon_\infty = (\epsilon_N - \bar{\epsilon}_N) + (\bar{\epsilon}_N - \bar{\epsilon}_\infty) + (\bar{\epsilon}_\infty - \epsilon_\infty) \tag{7.5.5}$$

The first term was estimated above, and turns out to be the dominant one for MNIST. The last term is independent of N, and should cancel the first term for asymptotically large N unaccessible in our numerics. The middle term is very interesting, as it characterizes the possibility that deep nets do better than kernel methods at finite N. In that case features can be learned, in contrast with the situation at $N \to \infty$ for which the time evolution the activity of any hidden neuron becomes vanishingly small (yet important) [105]. In magnitude, this term corresponds to the distance between the orange curve and its asymptote in Fig.7.2.1. For MNIST we observe that it is negative (which is compatible with the view that learning features improves generalization) for N slightly larger than N^*, but the effect is small. We provide an argument why it may be so. For large N, we expect the difference between $\bar{f}_N$ and $\bar{f}_\infty$ to stem from (i) the evolution of the kernel with time (which corresponds to learning features), described in Eq.7.5.3 and (ii) the fact that the relationship between the kernel and the function at infinite time is not linear, as described for the mean square loss in Eq.G.3.2 of the Supplementary Material. Both effects are $\mathcal{O}(N^{-1/2})$, i.e. much smaller than the $\mathcal{O}(N^{-1/4})$ fluctuations of f_N around its mean. The typical distance $\delta_{N,\infty}$ between the interfaces $\bar{f}_N = 0$ and $\bar{f}_\infty = 0$ is thus small and $\mathcal{O}(N^{-1/2})$. According to Eq.7.3.2 we get:

$$\bar{\epsilon}_N - \bar{\epsilon}_\infty = \int_B dx^{d-1} \left[\frac{\partial \epsilon}{\partial \delta(x)} \delta_{N,\infty}(x) + \mathcal{O}(\delta^2_{N,\infty}(x)) \right] \tag{7.5.6}$$

Thus $\bar{\epsilon}_N - \bar{\epsilon}_\infty = \mathcal{O}(N^{-1/2})$, and thus cannot be neglected a priori. The fact that this term is small in practice presumably reflects that $\frac{\partial \epsilon}{\partial \delta(x)} \delta_{N,\infty}(x)$ often changes sign along the boundary, leading to a small pre-factor. Understanding if the situation can be different for well-chosen architectures, for which learning features would enhance significantly generalization accuracy is an important question for the future [9].

Overall, we get $\epsilon_N - \epsilon_\infty = B_0 N^{-1/2} + B_1 N^{-3/4}$, a form indeed consistent with observation as shown in Fig.7.2.1. Note that a direct fit of the test error *vs* N gives an apparent exponent smaller than 1/2 [211], reflecting that (i) power-law fits are less precise when the value for the asymptote (here the value of ϵ_∞) is a fitting parameter and (ii) that correction to scaling needs to be incorporated for a good comparison with the theory (a fact that ultimately stems from the large correction to scaling of $\|\nabla f_N\|$ shown in Fig.7.5.2.A).

7.6 Vicinity of the jamming transition

The asymptotic description for generalization in the large N limit is not qualitatively useful for $N \leq N^*$, where a cusp in test error is found. We now argue that this cusp is induced by a singularity of $\|f_N\|$ at N^* when no regularization is used, as apparent in Fig.7.6.1.A. Indeed following our argument of Section 7.3, this effect must lead to singular fluctuations of the decision boundary at N^*, suggesting a non-analytical behavior for the true test error. This phenomenon shares some similarity with the norm divergence that occurs in linear networks with mean square loss for which $\|f_N\| \sim |N - P|^{-2}$ [1, 139]. Yet for losses better suited for classification such as the hinge loss, we argue that this explosion occurs at a different location with a different exponent.

[9] Very recently empirical results suggest that the test error can even increase for increasing and large N [34]. Yet, this observation was made in the teacher-student framework, where it is intuitively clear that the student should be penalized when its number of parameters becomes larger than the teacher.

Consider the hinge loss in Eq. 7.2.1. For $N \geq N^*$ the system is able to reach the ground state at $C = 0$, therefore all Δ_μ must be negative, i.e. all patterns must satisfy $y_\mu f(x_\mu) > \epsilon_m$. The parameter ϵ_m plays the role of a margin above which we are confident about the network's prediction. Because we do not use regularization on the norm $\|f\|$, the precise choice of ϵ_m does not affect N^*. Indeed the weights can always increase during learning so as to multiply f by any scalar λ, effectively reducing the margin by a factor $1/\lambda$, making the data easier to fit. By contrast, if a regularization is imposed to fix $\|f\| = \lambda$ (which may be hard to implement in practice), then N^* must be an increasing function of $\tilde{\epsilon}_m \equiv \epsilon_m/\lambda$. We assume that this function is differentiable in its argument around zero, a fact know to be true for the perceptron [62, 61], thus $N^*(\tilde{\epsilon}_m) = N^*(0)+B_0\tilde{\epsilon}_m+o(\tilde{\epsilon}_m)$. Now consider our learning scheme (no regularization) for a network with $0 < N/N^*(0) - 1 \ll 1$, with initial conditions such that before learning $\|f_N^{t=0}\| = 1$. Initially, the effective margin is large with $\tilde{\epsilon}_m = 1$. Yet, all data can be fitted and the loss brought to zero if the norm increases so that $\tilde{\epsilon}_m \approx (N - N^*(0))/B_0$, corresponding to $\|f_N^t\| \sim (N - N^*)^{-1}$ where $N^* = N^*(0)$. At later times, the loss is zero and the dynamics stops.

This predicted inverse relation is tested in Fig.7.6.1.B. It is important to note that, as it is the case for any critical points, working at finite times cuts off a true singularity: as illustrated in Fig.7.6.1.B $\|f_N^t\|$ becomes more and more singular at long times. This effect also causes a shift of the transition N^* where the loss vanishes, that converges asymptotically to a well-defined value in the limit $t \to \infty$ as documented in [68]. N^* is therefore defined when $\|f_N^t\|$ displays a power law as function of $N/N^* - 1$.

Note that for other losses like the cross-entropy, the dynamics never stops completely but becomes extremely slow [12]. In such cases, we expect that asymptotically $\|f_N^t\| = \infty$ as soon as $N > N^*$, although this singularity should build up logarithmically slowly in time. For finite learn-

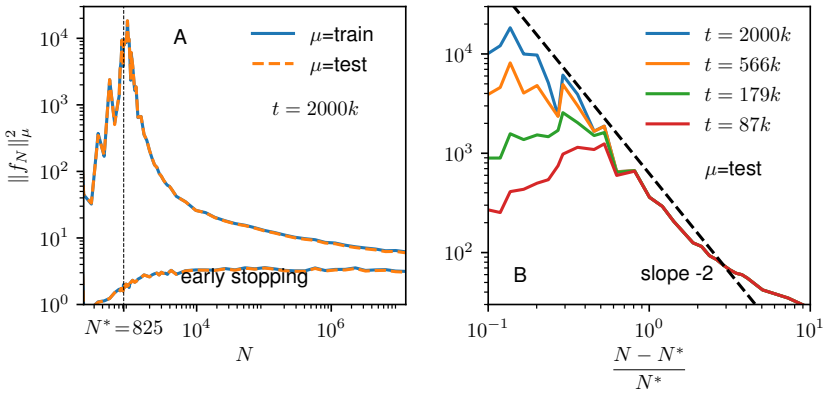


Figure 7.6.1: Here $L = 5$, $d = 10$, $P = 10k$. (A) $\|f\|^2 = \int d\mu(x) f(x)^2$ where for μ we took the uniform measure on the training and test set. We show the mean over the different realizations. Right after the jamming transition, the norm of the network diverges. (B) Same quantity computed after different learning times t as indicated in the legend, as a function of the distance from the transition. One observes that finite times cut off the divergence in the norm. The black line indicates a power-law with slope -2, that appears to fit the data satisfyingly. N^* has been fine tuned to obtain straight curves (power law behavior).

ing times we expect that a singularity will occur near N^*, but will be blurred as for the hinge loss if $t < \infty$.

7.7 Conclusion

We have provided a description for the evolution of the generalization performance of by fixed-depth fully-connected deep neural networks, as a function of their number of parameters N. In the asymptotic regime of very large N, we find empirically that the network output displays reduced fluctuations with $\|f_N - \bar{f}_N\| \sim N^{-1/4}$. We have argued that this scaling behavior is expected from the finite N fluctuations of the Neural Tangent Kernel known to control the dynamics at $N = \infty$. Next we have provided a general argument relating fluctuations of the network output to decreasing generalization performance, from which we predicted for the test error $\epsilon_N - \epsilon_\infty = C_0 N^{-1/2} + C_1 N^{-3/4} + \mathcal{O}(N^{-1})$, consistent with our observation on MNIST. Overall this approach explains the surprising finding that generalization keeps improving with the number of parameters.

Secondly, we have argued that this description breaks down at $N = N^*$ below which the training set is not fitted. For the hinge loss where this jamming transition is akin to a critical point, and in the case where no regularization (such as early stopping) is used, we observe the apparent divergence $\|f_N\| \sim (N - N^*)^{-\alpha}$. We have argued, based on reasonable assumptions, that $\alpha = 1$, consistent with our observations. This predicted enhanced variance of f explains the spike in error observed at N^*.

On the practical side, our analysis suggests that optimal generalization does not require to take N much larger than N^*: since improvement of generalization with N stems from reduced variance in the output function, near-optimal generalization is readily obtained by performing an ensemble average of networks with N fixed, e.g. taken to be a few times N^*. Interestingly, ensemble averaging turns out to be more efficient than increasing N memory-wise, due to the very slow decay $\sim N^{-1/4} \gg N^{-1/2}$ of fluctuations in deep networks. The usefulness of averaging breaks down near N^* where the variance of f is too large. We thus recover the intuition that the optimal model complexity is reached just beyond the point where the data can be perfectly fitted, a result of practical importance that needs to be tested in a wide range of architectures and datasets.

Chapter 8

Implicit Regularization of Random Feature Models

Abstract

Random Feature (RF) models are used as efficient parametric approximations of kernel methods. We investigate, by means of random matrix theory, the connection between Gaussian RF models and Kernel Ridge Regression (KRR). For a Gaussian RF model with P features, N data points, and a ridge λ, we show that the average (i.e. expected) RF predictor is close to a KRR predictor with an *effective ridge* $\tilde{\lambda}$. We show that $\tilde{\lambda} > \lambda$ and $\tilde{\lambda} \searrow \lambda$ monotonically as P grows, thus revealing the *implicit regularization effect* of finite RF sampling. We then compare the risk (i.e. test error) of the $\tilde{\lambda}$-KRR predictor with the average risk of the λ-RF predictor and obtain a precise and explicit bound on their difference. Finally, we empirically find an extremely good agreement between the test errors of the average λ-RF predictor and $\tilde{\lambda}$-KRR predictor.

8.1 Introduction

In this paper, we consider the Random Feature (RF) model which is an approximation of Kernel Methods [176] which has seen many recent theoretical developements.

The conventional wisdom suggests that to ensure good generalization performance, one should choose a model class that is complex enough to learn the signal from the training data, yet simple enough to avoid fitting spurious patterns therein [24]. This view has been questioned by recent developments in machine learning. First, [236] observed that modern neural network models can perfectly fit randomly labeled training data, while still generalizing well. Second, the test error as a function of parameters exhibits a so-called 'double-descent' curve for many models including neural networks, random forests, and random feature models [1, 211, 18, 152, 19, 157].

The above models share the feature that for fixed input, the learned predictor $\hat{f}$ is random: for neural networks, this is due to the random initialization of the parameters and/or to the stochasticity of the training algorithm; for random forests, to the random branching; for random feature models, to the sampling of random features. The somehow surprising generalization behavior of these models has recently been the subject of increasing attention. In general, the risk (i.e. test error) is a random variable with two sources of randomness: the usual one due to the sampling of the training set, and the second one due to the randomness of the model itself.

We consider the Random Feature (RF) model [176] with features sampled from a Gaussian Process (GP) and study the RF predictor $\hat{f}$ minimizing the regularized least squares error, isolating the randomness of the model by considering fixed training data points. RF models have been the subject of intense research activity: they are (randomized) approximations of Kernel Methods aimed at easing the computational challenges of Kernel Methods while being asymptotically equivalent to them [176, 232, 213, 234]. Unlike the asymptotic behavior, which is well studied, RF models with a finite number of features are much less understood.

Contributions

We consider a model of Random Features (RF) approximating a kernel method with kernel K. This model consists of P Gaussian features, sampled i.i.d. from a (centered) Gaussian process with covariance kernel K. For a given training set of size N, we study the distribution of the RF predictor $\hat{f}_\lambda^{(RF)}$ with ridge parameter $\lambda > 0$ (L^2 penalty on the parameters) and denote it by λ-RF. We show the following:

- The distribution of $\hat{f}_\lambda^{(RF)}$ is that of a mixture of Gaussian processes.
- The expected RF predictor is close to the $\tilde{\lambda}$-KRR (Kernel Ridge Regression) predictor for an effective ridge parameter $\tilde{\lambda} > 0$.
- The effective ridge $\tilde{\lambda} > \lambda$ is determined by the number of features P, the ridge λ and the Gram matrix of K on the dataset; $\tilde{\lambda}$ decreases monotonically to λ as P grows, revealing the implicit regularization effect of finite RF sampling. Conversely, when using random features to approximate a kernel method with a specific ridge λ^*, one should choose a smaller ridge $\lambda < \lambda^*$ to ensure $\tilde{\lambda}(\lambda) = \lambda^*$.
- The test errors of the expected λ-RF predictor and of the $\tilde{\lambda}$-KRR predictor $\hat{f}_{\tilde{\lambda}}^{(K)}$ are numerically found to be extremely close, even for small P and N.
- The RF predictor's concentration around its expectation can be explicitly controlled in terms of P and of the data; this yields in particular $\mathbb{E}[L(\hat{f}_\lambda^{(RF)})] = L(\hat{f}_{\tilde{\lambda}}^{(K)}) + \mathcal{O}(P^{-1})$ as $N, P \to \infty$ with a fixed ratio $\gamma = {}^P/_N$ where L is the MSE risk.

Since we compare the behavior of λ-RF and $\tilde{\lambda}$-KRR predictors on the same fixed training set, our result does not rely on any probabilistic assumption on the training data (in particular, we do not assume that our training data is sampled i.i.d.). While our proofs currently require the features to be Gaussian processes, we are confident that they could be generalized to a more general setting [145, 23].

Related works

Generalization of Random Features. The generalization behavior of Random Feature models has seen intense study in the Statistical Learning Theory framework. [177] find that $\mathcal{O}(N)$ features are sufficient to ensure the $\mathcal{O}(\frac{1}{\sqrt{N}})$ decay of the generalization error of Kernel Ridge Regression (KRR). [185] improve on their result and show that $\mathcal{O}(\sqrt{N}\log N)$ features is actually enough to obtain the $\mathcal{O}(\frac{1}{\sqrt{N}})$ decay of the KRR error.

[85] use random matrix theory tools to compute the asymptotic risk when both $P, N \to \infty$ with $\frac{P}{N} \to \gamma > 0$. When the training data is sampled i.i.d. from a Gaussian distribution, the variance is shown to explode at $\gamma = 1$. In the same linear regression setup, [17] establish general upper and lower bounds on the excess risk. [152] prove that the double-descent (DD) curve also arises for random ReLU features, and adding a ridge suppresses the explosion around $\gamma = 1$.

Double-descent and the effect of regularization. For the cross-entropy loss, [162] observed that for two-layer neural networks the test error exhibits the double-descent (DD) curve as the network width increases (without regularizers, without early stopping). For MSE and hinge losses, the DD curve was observed also in multilayer networks on the MNIST dataset [1, 211]. [158] study the variance due to stochastic training in neural networks and find that it increases until a certain width, but then decreases down to 0. [157] establish the DD phenomenon across various models including convolutional and recurrent networks on more complex datasets (e.g. CIFAR-10, CIFAR-100).

[18, 19] find that the DD curve is not peculiar to neural networks and observe the same for random Fourier features and decision trees. In [67], the DD curve for neural networks is related to the variance associated with the random initialization of the Neural Tangent Kernel [105]; as a result, ensembling is shown to suppress the DD phenomenon in this case, and the test error stays constant in the overparameterized regime. Recent theoretical work [43] study the same setting and derive formulas for the asymptotic error, relying on the so-called replica method.

General Wishart Matrices. Our theoretical analysis relies on the study of the spectrum of the so-called general Wishart matrices of the form $W\Sigma W^T$ (for $N \times N$ matrix Σ and $P \times N$ matrix W with i.i.d. standard Gaussian entries) and in particular their Stieltjes transform $m_P(z) = \frac{1}{P}\mathrm{Tr}\left(W\Sigma W^T - zI_P\right)^{-1}$. A number of asymptotic results [203, 11] about the spectrum and Stieltjes transform of such matrices can be understood using the asymptotic freeness of $W^T W$ and Σ [65, 210]. In this paper, we provide non-asymptotic variants of these results for an arbitrary matrix Σ (which in our setting is the kernel Gram matrix); the proofs in our setting are detailed in the Supp. Mat.

Outline

The rest of this paper is organized as follows:

- In Section 8.2, the setup (linear regression, Gaussian RF model, λ-RF predictor, and λ-KRR predictor) is introduced.
- In Section 8.3, preliminary results on the distribution of the λ-RF model are provided: the RF predictors are Gaussian mixtures (Proposition 8.3.1) and the $\lambda \searrow 0$-RF model is unbiased in the overparameterized regime (Corollary 8.3.2). Graphical illustrations of the RF predictors in various regimes are presented (Figure 8.2.1).
- In Section 8.4, the first main theorem is stated (Theorem 8.4.1): the average (expected) λ-RF predictor is close to the $\tilde{\lambda}$-KRR predictor for an explicit $\tilde{\lambda} > \lambda$. As a consequence (Corollary 8.4.3), the test errors of these two predictors are close. Finally, numerical experiments show that the test errors are in fact virtually identical (Figure 8.3.1).
- In Section 8.5, the second main theorem is stated (Theorem H.3): a bound on the variance of the λ-RF predictor is given, which show that it concentrates around the average λ-RF predictor. As a consequence, the test error of the λ-RF predictor is shown to be close to that

of the $\tilde{\lambda}$-KRR predictor (Corollary H.3.16). The ridgeless $\lambda \searrow 0$ case is then investigated (Section 8.5): a lower bound on the variance of the λ-RF predictor is given, suggesting an explanation for the double-descent curve in the ridgeless case.

- In Section 8.6, we summarize our results and discuss potential implications and extensions.

8.2 Setup

Linear regression is a parametric model consisting of linear combinations

$$f_\theta = \frac{1}{\sqrt{P}}\left(\theta_1\phi^{(1)} + \cdots + \theta_P\phi^{(P)}\right)$$

of (deterministic) features $\phi^{(1)}, \ldots, \phi^{(P)} : \mathbb{R}^d \to \mathbb{R}$. We consider an arbitrary training dataset (X, y) with $X = [x_1, ..., x_N] \in \mathbb{R}^{d\times N}$ and $y = [y_1, \ldots, y_N] \in \mathbb{R}^N$, where the labels could be noisy observations. For a ridge parameter $\lambda > 0$, the linear estimator corresponds to the parameters $\hat{\theta} = [\hat{\theta}_1, \ldots, \hat{\theta}_P] \in \mathbb{R}^P$ that minimize the (regularized) Mean Square Error (MSE) functional $\hat{L}_\lambda$ defined by

$$\hat{L}_\lambda(f_\theta) = \frac{1}{N}\sum_{i=1}^{N}\left(f_\theta(x_i) - y_i\right)^2 + \frac{\lambda}{N}\|\theta\|^2. \tag{8.2.1}$$

The *data matrix* F is defined as the $N \times P$ matrix with entries $F_{ij} = \frac{1}{\sqrt{P}}\phi^{(j)}(x_i)$. The minimization of (8.2.1) can be rewritten in terms of F as

$$\hat{\theta} = \mathrm{argmin}_\theta \|F\theta - y\|^2 + \lambda\|\theta\|^2. \tag{8.2.2}$$

The optimal solution $\hat{\theta}$ is then given by

$$\hat{\theta} = F^T\left(FF^T + \lambda I_N\right)^{-1} y \tag{8.2.3}$$

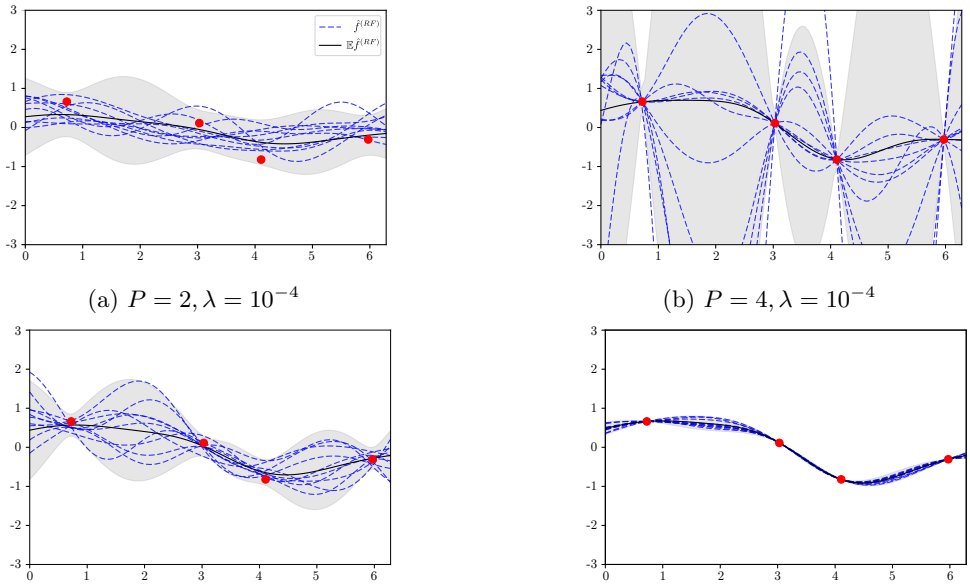
and the optimal predictor $\hat{f} = f_{\hat{\theta}}$ by

$$\hat{f}(x) = \frac{1}{\sqrt{P}}\sum_{j=1}^{P}\phi^{(j)}(x)F^T_{:,j}\left(FF^T + \lambda I_N\right)^{-1} y. \tag{8.2.4}$$

In this paper, we consider linear models of *Gaussian random features* associated with a kernel K: $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. We take $\phi^{(j)} = f^{(j)}$, where $f^{(1)}, \ldots, f^{(P)}$ are sampled i.i.d. from a Gaussian Process of zero mean (i.e. $\mathbb{E}[f^{(j)}(x)] = 0$ for all $x \in \mathbb{R}^d$) and with covariance K (i.e. $\mathbb{E}[f^{(j)}(x)f^{(j)}(x')] = K(x, x')$ for all $x, x' \in \mathbb{R}^d$). In our setup, the optimal parameter $\hat{\theta}$ still satisfies (8.2.3) where F is now a random matrix. The associated predictor, called λ-RF predictor, is then given by

Definition 5 (Random Feature Predictor)**.** *Consider a kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, a ridge $\lambda > 0$, and random features $f^{(1)}, \ldots, f^{(P)}$ sampled i.i.d. from a centered Gaussian Process of covariance K. Let $\hat{\theta}$ be the optimal solution to (8.2.1) taking $\phi^{(j)} = f^{(j)}$. The Random Feature predictor with ridge λ is the random function $\hat{f}^{(RF)}_\lambda : \mathbb{R}^d \to \mathbb{R}$ defined by*

$$\hat{f}^{(RF)}_\lambda(x) = \frac{1}{\sqrt{P}}\sum_{j=1}^{P}\hat{\theta}_j f^{(j)}(x). \tag{8.2.5}$$



(a) $P = 2, \lambda = 10^{-4}$ (b) $P = 4, \lambda = 10^{-4}$

(c) $P = 4, \lambda = 0.1$ (d) $P = 100, \lambda = 10^{-4}$

Figure 8.2.1: *Distribution of the RF Predictor.* Red dots represent a sinusoidal dataset $y_i = \sin(x_i)$ for $N = 4$ points x_i in $[0, 2\pi)$. For selected P and λ, we sample ten RF predictors (blue dashed lines) and compute empirically the average RF predictor (black lines) with ± 2 standard deviations intervals (shaded regions).

The λ-RF can be viewed as an approximation of kernel ridge predictors: observing from (8.2.4) that $\hat{f}_\lambda^{(RF)}$ only depends on the scalar product $K_P(x, x') = \frac{1}{P}\sum_{j=1}^{P} f^{(j)}(x)f^{(j)}(x')$ between data-points, we see that as $P \to \infty$, $K_P \to K$ and hence $\hat{f}_\lambda^{(RF)}$ converges [176] to a kernel predictor with ridge λ [198], which we call λ-KRR predictor.

Definition 6 (Kernel Predictor)**.** *Consider a kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ and a ridge $\lambda > 0$. The Kernel Predictor is the function $\hat{f}_\lambda^{(K)} : \mathbb{R}^d \to \mathbb{R}$*

$$\hat{f}_\lambda^{(K)}(x) = K(x, X)(K(X, X) + \lambda \mathrm{I}_N)^{-1} y$$

where $K(X, X)$ is the $N \times N$ matrix of entries $(K(X, X))_{ij} = K(x_i, x_j)$ and $K(\,\cdot\,, X) : \mathbb{R}^d \to \mathbb{R}^N$ is the map $(K(x, X))_i = K(x, x_i)$.

Bias-Variance Decomposition.

Let us assume that there exists a true regression function $f^* : \mathbb{R}^d \to \mathbb{R}$ and a data generating distribution $\mathcal{D}$ on $\mathbb{R}^d$. The risk of a predictor $f : \mathbb{R}^d \to \mathbb{R}$ is measured by the MSE defined as

$$L(f) = \mathbb{E}_{\mathcal{D}}\left[(f(x) - f^*(x))^2\right].$$

Let π denote the joint distribution of the i.i.d. sample $f^{(1)}, ..., f^{(P)}$ from the centered Gaussian process with covariance kernel K. The risk of $\hat{f}_\lambda^{(RF)}$ can be decomposed into a bias-variance form

as

$$\mathbb{E}_\pi\Big[L(\hat{f}_\lambda^{(RF)})\Big] = L\left(\mathbb{E}_\pi[\hat{f}_\lambda^{(RF)}]\right) + \mathbb{E}_{\mathcal{D}}\Big[\mathrm{Var}_\pi(\hat{f}_\lambda^{(RF)}(x))\Big].$$

This decomposition into the risk of the *average* RF predictor and of the $\mathcal{D}$-expectation of its variance will play a crucial role in the next sections. This is in contrast with the classical bias-variance decomposition in [71]

$$\mathbb{E}_{\mathcal{D}^{\otimes N}}[L(f)] = L(\mathbb{E}_{\mathcal{D}^{\otimes N}}[f]) + \mathbb{E}_{\mathcal{D}}[\mathrm{Var}_{\mathcal{D}^{\otimes N}}[f(x)]]$$

where $\mathcal{D}^{\otimes N}$ denotes the joint distribution on $x_1, ..., x_N$, sampled i.i.d. from $\mathcal{D}$. Note that in our decomposition no probabilistic assumption is made on the data, which is fixed.

Additional Notation

In this paper, we consider a fixed dataset (X, y) with distinct data points and a kernel K (i.e. a positive definite symmetric function $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$). We denote by $\|y\|_{K^{-1}}$ the inverse kernel norm of the labels defined as $y^T(K(X,X))^{-1}y$.

Let UDU^T be the spectral decomposition of the kernel matrix $K(X,X)$, with $D = \mathrm{diag}(d_1, \dots, d_N)$. Let $D^{\frac{1}{2}} = \mathrm{diag}(\sqrt{d_1}, \dots, \sqrt{d_N})$ and set $K^{\frac{1}{2}} = UD^{\frac{1}{2}}U^T$. The law of the (random) data matrix F is now that of $\frac{1}{\sqrt{P}}K^{\frac{1}{2}}W^T$ where W is a $P \times N$ matrix of i.i.d. standard Gaussian entries, so that $\mathbb{E}[FF^T] = K(X,X)$.

We will denote by $\gamma = \frac{P}{N}$ the parameter-to-datapoint ratio: the *underparameterized regime* corresponds to $\gamma < 1$, while the *overparameterized regime* corresponds to $\gamma \geq 1$. In order to stress the dependence on the ratio parameter γ, we write $\hat{f}_{\lambda,\gamma}^{(RF)}$ instead of $\hat{f}_\lambda^{(RF)}$.

8.3 First Observations

The distribution of the RF predictor features a variety of behaviors depending on γ and λ, as displayed in Figure 8.2.1. In the underparameterized regime $P < N$, sample RF predictors induce some *implicit regularization* and do not interpolate the dataset (8.2.1a); at the interpolation threshold $P = N$, RF predictors interpolate the dataset but the variance explodes when there is no ridge (8.2.1b), however adding some ridge suppresses variance explosion (8.2.1c); in the overparameterized regime $P \geq N$ with large P, the variance vanishes thus the RF predictor converges to its average (8.2.1d). We will investigate the average RF predictor (solid lines) in detail in Section 8.4 and study its variance in Section 8.5.

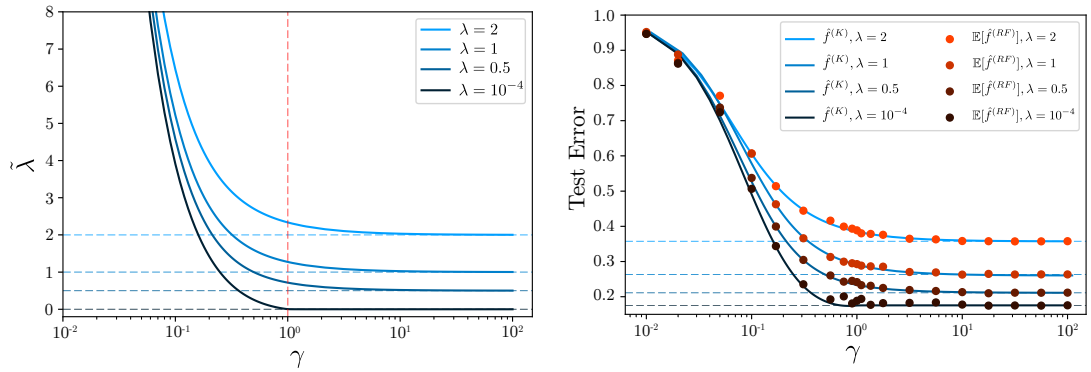
We start by characterizing the distribution of the RF predictor as a Gaussian mixture:

Proposition 8.3.1. *Let $\hat{f}_{\lambda,\gamma}^{(RF)}(x)$ be the random features predictor as in (8.2.5) and let $\hat{y} = F\hat{\theta}$ be the prediction vector on training data, i.e. $\hat{y}_i = \hat{f}_{\lambda,\gamma}^{(RF)}(x_i)$. The process $\hat{f}_{\lambda,\gamma}^{(RF)}$ is a mixture of Gaussians: conditioned on F, we have that $\hat{f}_{\lambda,\gamma}^{(RF)}$ is a Gaussian process. The mean and covariance of $\hat{f}_{\lambda,\gamma}^{(RF)}$ conditioned on F are given by*

$$\mathbb{E}[\hat{f}_{\lambda,\gamma}^{(RF)}(x)|F] = K(x,X)K(X,X)^{-1}\hat{y}, \tag{8.3.1}$$

$$\mathrm{Cov}[\hat{f}_{\lambda,\gamma}^{(RF)}(x), \hat{f}_{\lambda,\gamma}^{(RF)}(x')|F] = \frac{\|\hat{\theta}\|^2}{P}\tilde{K}(x,x'), \tag{8.3.2}$$

with $\tilde{K}(x,x') = K(x,x') - K(x,X)K(X,X)^{-1}K(X,x')$ denoting the posterior covariance kernel.



(a) Evolution of $\tilde{\lambda}$

(b) Average λ-RF predictor vs. $\tilde{\lambda}$-KRR

Figure 8.3.1: *Comparison of the test errors of the average λ-RF predictor and the $\tilde{\lambda}$-KRR predictor.* We train the RF predictors on $N = 100$ MNIST data points where K is the RBF kernel, i.e. $K(x, x') = \exp\left(-\|x - x'\|^2/\ell\right)$. We approximate the average λ-RF on 100 random test points for various ridges λ. In **(a)**, given γ and λ, the effective ridge $\tilde{\lambda}$ is computed numerically using (8.4.2). In **(b)**, the test errors of the $\tilde{\lambda}$-KRR predictor (blue lines) and the empirical average of the λ-RF predictor (red dots) agree perfectly.

The proof of Proposition 8.3.1 relies on the fact that $f^{(j)}$ conditioned on $\left(f^{(j)}(x_i)\right)_{i=1,\dots,N}$ is a Gaussian Process.

Note that (8.3.1) and (8.3.2) depend on λ and P through $\hat{y}$ and $\|\hat{\theta}\|^2$; in fact, as the proof shows, these identities extend to the ridgeless case $\lambda \searrow 0$. For the ridgeless case, when one is in the overparameterized regime ($P \geq N$), one can (with probability one) fit the labels y and hence $\hat{y} = y$:

Corollary 8.3.2. *When $P \geq N$, the average ridgeless RF predictor is equivalent to the ridgeless KRR predictor*

$$\mathbb{E}\left[\hat{f}^{(RF)}_{\lambda\searrow 0,\gamma}(x)\right] = K(x, X)K(X, X)^{-1}y = \hat{f}^{(K)}_{\lambda\searrow 0}(x).$$

This corollary shows that in the overparameterized case, the ridgeless RF predictor is an unbiased estimator of the ridgeless kernel predictor. The difference between the expected loss of ridgeless RF predictor and that of the ridgeless KRR predictor is hence equal to the variance of the RF predictor. As will be demonstrated in this article, outside of this specific regime, a systematic bias appears, which reveals an implicit regularizing effect of random features.

8.4 Average Predictor

In this section, we study the average RF predictor $\mathbb{E}[\hat{f}^{(RF)}_{\lambda,\gamma}]$. As shown by Corollary 8.3.2 above, in the ridgeless overparmeterized regime, the RF predictor is an unbiased estimator of the ridgeless kernel predictor. However, in the presence of a non-zero ridge, we see the following *implicit regularization effect*: the average λ-RF predictor is close to the $\tilde{\lambda}$-KRR predictor for an effective ridge

$\tilde{\lambda} > \lambda$ (in other words, sampling a finite number P of features amounts to taking a greater kernel ridge $\tilde{\lambda}$).

Theorem 8.4.1. *For $N, P > 0$ and $\lambda > 0$, we have*

$$\left|\mathbb{E}[\hat{f}^{(RF)}_{\lambda,\gamma}(x)] - \hat{f}^{(K)}_{\tilde{\lambda}}(x)\right| \leq \frac{c\sqrt{K(x,x)}\,\|y\|_{K^{-1}}}{P} \tag{8.4.1}$$

where the effective ridge $\tilde{\lambda}(\lambda,\gamma) > \lambda$ is the unique positive number satisfying

$$\tilde{\lambda} = \lambda + \frac{\tilde{\lambda}}{\gamma}\frac{1}{N}\sum_{i=1}^{N}\frac{d_i}{\tilde{\lambda}+d_i}, \tag{8.4.2}$$

and where $c > 0$ depends on λ, γ, and $\frac{1}{N}\mathrm{Tr}K(X,X)$ only.

Proof. (Sketch; see Supp. Mat. for details) Set $A_\lambda = F(F^TF + \lambda I_P)^{-1}F^T$. The vector of the predictions on the training set is given by $\hat{y} = A_\lambda y$ and the expected predictor is given by

$$\mathbb{E}\left[\hat{f}^{(RF)}_{\lambda,\gamma}(x)\right] = K(x,X)K(X,X)^{-1}\mathbb{E}\left[A_\lambda\right]y.$$

By a change of basis, we may assume the kernel Gram matrix to be diagonal, i.e. $K(X,X) = \mathrm{diag}(d_1,\dots,d_N)$. In this basis $\mathbb{E}[A_\lambda]$ turns out to be diagonal too. For each $i = 1,\dots,N$ we can isolate the contribution of the i-th row of F: by the Sherman-Morrison formula, we have $(A_\lambda)_{ii} = \frac{d_i g_i}{1+d_i g_i}$, where

$$g_i = \frac{1}{P}W_i^T(F_{(i)}^TF_{(i)} + \lambda \mathrm{I}_P)^{-1}W_i,$$

with W_i denoting the i-th column of $W = \sqrt{P}F^TK^{-\frac{1}{2}}$ and $F_{(i)}$ being obtained by removing the i-th row of F. The g_i's are all within $\mathcal{O}(1/\sqrt{P})$ distance to the Stieltjes transform

$$m_P(-\lambda) = \frac{1}{P}\mathrm{Tr}\left(F^TF + \lambda \mathrm{I}_P\right)^{-1}.$$

By a fixed point argument, the Stieltjes transform $m_P(-\lambda)$ is itself within $\mathcal{O}(1/\sqrt{P})$ distance to the deterministic value $\tilde{m}(-\lambda)$, where $\tilde{m}$ is the unique positive solution to

$$\gamma = \frac{1}{N}\sum_{i=1}^{N}\frac{d_i\tilde{m}(z)}{1+d_i\tilde{m}(z)} - \gamma z\tilde{m}(z).$$

(The detailed proof in the Supp. Mat. uses non-asymptotic variants of arguments found in [11]; the constants in the $\mathcal{O}$ bounds are in particular made explicit).

As a consequence, from the above results, we obtain

$$\mathbb{E}\left[(A_\lambda)_{ii}\right] = \mathbb{E}\left[\frac{d_i g_i}{1+d_i g_i}\right] \approx \frac{d_i\tilde{m}}{1+d_i\tilde{m}} = \frac{d_i}{\tilde{\lambda}+d_i},$$

revealing the effective ridge $\tilde{\lambda} = 1/\tilde{m}(-\lambda)$.

This implies that $\mathbb{E}[A_\lambda] \approx K(X,X)(K(X,X)+\tilde{\lambda}\mathrm{I}_N)^{-1}$ and

$$\mathbb{E}\left[\hat{f}^{(RF)}_{\lambda,\gamma}(x)\right] \approx K(x,X)(K(X,X)+\tilde{\lambda}\mathrm{I}_N)^{-1}y = \hat{f}^{(K)}_{\tilde{\lambda}}(x),$$

yielding the desired result. □

Note that asymptotic forms of equations similar to the ones in the above proof appear in different settings [48, 152, 144], related to the study of the Stieltjes transform of the product of asymptotically free random matrices.

While the above theorem does not make assumptions on P, N, and K, the case of interest is when the right hand side $\frac{cK(x,x)\|y\|_{K^{-1}}}{P}$ is small. The constant $c > 0$ is uniformly bounded whenever γ and λ are bounded away from 0 and $\frac{1}{N}\mathrm{Tr}K(X,X)$ is bounded from above. As a result, to bound the right hand side of (8.4.1), the two quantities we need to bound are $T = \frac{1}{N}\mathrm{Tr}K(X,X)$ and $\|y\|_{K^{-1}}$.

- The boundedness of T is guaranteed for kernels that are translation-invariant, i.e. of the form $K(x,y) = k(\|x-y\|)$: in this case, one has $T = k(0)$.
- If we assume $\mathbb{E}_{\mathcal{D}}[K(x,x)] < \infty$ (as is commonly done in the literature [185]), T converges to $\mathbb{E}_{\mathcal{D}}[K(x,x)]$ as $N \to \infty$ (assuming i.i.d. data points).
- For $\|y\|_{K^{-1}}$, under the assumption that the labels are of the form $y_i = f^*(x_i)$ for a true regression function f^* lying in Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}$ of the kernel K [198], we have $\|y\|_{K^{-1}} \leq \|f^*\|_{\mathcal{H}}$.

Our numerical experiments in Figure (8.3.1b) show excellent agreement between the test error of the expected λ-RF predictor and the one of the $\tilde{\lambda}$-KRR predictor suggesting that the two functions are indeed very close, even for small N, P.

Thanks to the implicit definition of the effective ridge $\tilde{\lambda}$ (which depends on λ, γ, N and on the eigenvalues d_i of $K(X,X)$) we obtain the following:

Proposition 8.4.2. *The effective ridge $\tilde{\lambda}$ satisfies the following properties:*

1. *for any $\gamma > 0$, we have $\lambda < \tilde{\lambda}(\lambda,\gamma) \leq \lambda + \frac{1}{\gamma}T$;*
2. *the function $\gamma \mapsto \tilde{\lambda}(\lambda,\gamma)$ is decreasing;*
3. *for $\gamma > 1$, we have $\tilde{\lambda} \leq \frac{\gamma}{\gamma-1}\lambda$;*
4. *for $\gamma < 1$, we have $\tilde{\lambda} \geq \frac{1-\sqrt{\gamma}}{\sqrt{\gamma}}\min_i d_i$.*

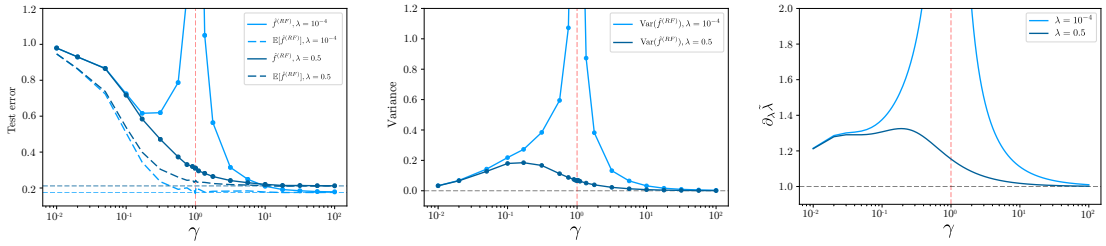
The above proposition shows the implicit regularization effect of the RF model: sampling fewer features (i.e. decreasing γ) increases the effective ridge $\tilde{\lambda}$.

Furthermore, as $\lambda \to 0$ (ridgeless case), the effective ridge $\tilde{\lambda}$ behave as follows:

- in the overparameterized regime ($\gamma > 1$), $\tilde{\lambda}$ goes to 0;
- in the underparameterized regime ($\gamma < 1$), $\tilde{\lambda}$ goes to a limit $\tilde{\lambda}_0 > 0$.

These observations match the profile of $\tilde{\lambda}$ in Figure (8.3.1a).

Remark. When $\lambda \searrow 0$, the constant c in our bound (8.4.1) explodes (see Supp. Mat.). As a result, this bound is not directly useful when $\lambda = 0$. However, we know from Corollary 8.3.2 that in the ridgeless overparametrized case ($\gamma > 1$), the average RF predictor is equal to the ridgeless KRR predictor. In the underparametrized case ($\gamma < 1$), our numerical experiments suggest that the ridgeless RF predictor is an excellent approximation of the $\tilde{\lambda}_0$-KRR predictor.



(a) Ridgeless vs. Ridge (b) Variance of λ-RF (c) Evolution of $\partial_\lambda \tilde{\lambda}$

Figure 8.4.1: *Average test error of the ridgeless vs. ridge λ-RF predictors.* In **(a)**, the average test errors of the ridgeless and the ridge RF predictors (solid lines) and the effect of ensembling (dashed lines) for $N = 100$ MNIST data points. In **(b)**, the variance of the RF predictors and in **(c)**, the evolution of $\partial_\lambda \tilde{\lambda}$ in the ridgeless and ridge cases. The experimental setup is the same as in Figure 1.8.2.

Effective Dimension

The effective ridge $\tilde{\lambda}$ is closely related to the so-called effective dimension appearing in statistical learning theory. For a linear (or kernel) model with ridge λ, the *effective dimension* $\mathcal{N}(\lambda) \leq N$ is defined as $\sum_{i=1}^{N} \frac{d_i}{\lambda + d_i}$ [238, 29]. It allows one to measure the effective complexity of the Hilbert space in the presence of a ridge.

For a given $\lambda > 0$, the effective ridge $\tilde{\lambda}$ introduced in Theorem 8.4.1 is related to the effective dimension $\mathcal{N}(\tilde{\lambda})$ by

$$\mathcal{N}(\tilde{\lambda}) = P\left(1 - \frac{\lambda}{\tilde{\lambda}}\right).$$

In particular, we have that $\mathcal{N}(\tilde{\lambda}) \leq \min(N, P)$: this shows that the choice of a finite number of features corresponds to an automatic lowering of the effective dimension of the related kernel method.

Note that in the ridgeless underparameterized case ($\lambda \searrow 0$ and $\gamma < 1$), the effective dimension $\mathcal{N}(\tilde{\lambda})$ equals precisely the number of features P.

Risk of the Average Predictor

A corollary of Theorem 8.4.1 is that the loss of the expected RF predictor is close to the loss of the KRR predictor with ridge $\tilde{\lambda}$:

Corollary 8.4.3. *If $\mathbb{E}_{\mathcal{D}}[K(x,x)] < \infty$, we have that the difference of errors $\delta_E = \left| L(\mathbb{E}[\hat{f}^{(RF)}_{\lambda,\gamma}]) - L(\hat{f}^{(K)}_{\tilde{\lambda}}) \right|$ is bounded from above by*

$$\delta_E \leq \frac{C \left\| y \right\|_{K^{-1}}}{P} \left(2\sqrt{L\left(\hat{f}^{(K)}_{\tilde{\lambda}}\right)} + \frac{C \left\| y \right\|_{K^{-1}}}{P} \right),$$

where C is given by $c\sqrt{\mathbb{E}_{\mathcal{D}}[K(x,x)]}$, with c the constant appearing in (8.4.1) above.

As a result, δ_E can be bounded in terms of $\lambda, \gamma, T, \|y\|_{K^{-1}}$, which are discussed above, and of the kernel generalization error $L(f_{\tilde{\lambda}}^{(K)})$. Such a generalization error can be controlled in a number of settings as N grows: in [29, 150], for instance, the loss is shown to vanish as $N \to \infty$. Figure (8.3.1b) shows that the two test losses are indeed very close.

8.5 Variance

In the previous sections, we analyzed the loss of the expected predictor $\mathbb{E}[\hat{f}_{\lambda,\gamma}^{(RF)}]$. In order to analyze the expected loss of the RF predictor $\hat{f}_{\lambda,\gamma}^{(RF)}$, it remains to control the variance of the RF predictor: this follows from the bias-variance decomposition

$$\mathbb{E}\Big[L(\hat{f}_{\lambda,\gamma}^{(RF)})\Big] = L\left(\mathbb{E}[\hat{f}_{\lambda,\gamma}^{(RF)}]\right) + \mathbb{E}_{\mathcal{D}}\Big[\mathrm{Var}(\hat{f}_{\lambda,\gamma}^{(RF)}(x))\Big],$$

introduced in Section 8.2.

The variance $\mathrm{Var}\left(\hat{f}_{\lambda,\gamma}^{(RF)}(x)\right)$ of the RF predictor can itself be written as the sum

$$\mathrm{Var}\left(\mathbb{E}\left[\hat{f}_{\lambda,\gamma}^{(RF)}(x) \mid F\right]\right) + \mathbb{E}\left[\mathrm{Var}\left(\hat{f}_{\lambda,\gamma}^{(RF)}(x) \mid F\right)\right].$$

By Proposition 8.3.1, we have

$$\mathbb{E}\left[\hat{f}_{\lambda,\gamma}^{(RF)}(x) \mid F\right] = K(x,X)K(X,X)^{-1}\hat{y}$$

$$\mathrm{Var}\left(\hat{f}_{\lambda,\gamma}^{(RF)}(x) \mid F\right) = \frac{\|\hat{\theta}\|^2}{P}\tilde{K}(x,x).$$

RF Predictor Concentration

The following theorem allows us to bound both terms:

Theorem 8.5.1. *There are constants $c_1, c_2 > 0$ depending on λ, γ, T only such that*

$$\mathrm{Var}\left(K(x,X)K(X,X)^{-1}\hat{y}\right) \leq \frac{c_1 K(x,x)\|y\|_{K^{-1}}^2}{P}$$

$$\left|\mathbb{E}\|[\hat{\theta}\|^2] - \partial_\lambda\tilde{\lambda}\, y^T M_{\tilde{\lambda}} y\right| \leq \frac{c_2\|y\|_{K^{-1}}^2}{P},$$

where $\partial_\lambda\tilde{\lambda}$ is the derivative of $\tilde{\lambda}$ with respect to λ and for $M_{\tilde{\lambda}} = K(X,X)(K(X,X)+\tilde{\lambda}I_N)^{-2}$. As a result

$$\mathrm{Var}\left(\hat{f}_{\lambda,\gamma}^{(RF)}(x)\right) \leq \frac{c_3 K(x,x)\|y\|_{K^{-1}}^2}{P},$$

where $c_3 > 0$ depends on λ, γ, T.

Putting the pieces together, we obtain the following bound on the difference $\Delta_E = |\mathbb{E}[L(\hat{f}_{\lambda,\gamma}^{(RF)})] - L(\hat{f}_{\tilde{\lambda}}^{(K)})|$ between the expected RF loss and the KRR loss:

Corollary 8.5.2. *If $\mathbb{E}_{\mathcal{D}}[K(x,x)] < \infty$, we have*

$$\Delta_E \leq \frac{C_1\|y\|_{K^{-1}}}{P}\left(\sqrt{L(\hat{f}_{\tilde{\lambda}}^{(K)})} + C_2\|y\|_{K^{-1}}\right).$$

where C_1 and C_2 depend on λ, γ, T and $\mathbb{E}_{\mathcal{D}}[K(x,x)]$ only.

Double Descent Curve

We now investigate the neighborhood of the frontier $\gamma = 1$ between the under- and overparameterized regimes, known empirically to exhibit a double descent curve, where the test error explodes at $\gamma = 1$ (i.e. when $P \approx N$) as exhibited in Figure 8.4.1.

Thanks to Theorem H.3, we get a lower bound on the variance of $\hat{f}^{(RF)}_{\lambda,\gamma}$:

Corollary 8.5.3. *There exists $c_4 > 0$ depending on λ, γ, T only such that $\mathrm{Var}(\hat{f}^{(RF)}_{\lambda,\gamma}(x))$ is bounded from below by*

$$\partial_\lambda \tilde{\lambda} \frac{y^T M_{\tilde{\lambda}} y}{P} \tilde{K}(x,x) - \frac{c_4 K(x,x) \|y\|^2_{K^{-1}}}{P^2}.$$

If we assume the second term of Corollary H.3.17 to be negligible, then the only term which depends on P is $\partial_\lambda \tilde{\lambda} \frac{y^T M_{\tilde{\lambda}} y}{P}$. The derivative $\partial_\lambda \tilde{\lambda}$ has an interesting behavior as a function of λ and γ:

Proposition 8.5.4. *For $\gamma > 1$, as $\lambda \to 0$, the derivative $\partial_\lambda \tilde{\lambda}$ converges to $\frac{\gamma}{\gamma - 1}$. As $\lambda\gamma \to \infty$, we have $\partial_\lambda \tilde{\lambda}(\lambda, \gamma) \to 1$.*

The explosion of $\partial_\lambda \tilde{\lambda}$ in $(\gamma = 1, \lambda = 0)$ is displayed in Figure (8.4.1c).

Corollary H.3.17 can be used to explain the double-descent curve numerically observed for small $\lambda > 0$. It is natural to assume that in this case $\partial_\lambda \tilde{\lambda} \gg 1$ around $\gamma = 1$, dominating the lower bound in Corollary H.3.17. In turn, by Proposition H.3.11 this implies that the variance of $\hat{f}^{(RF)}$ gets large. Finally, by the bias-variance decomposition, we obtain a sharp increase of the test error around $\gamma = 1$, which is in line with the results of [85, 152].

8.6 Conclusion

In this paper, we have identified the implicit regularization arising from the finite sampling of Random Features (RF): using a Gaussian RF model with ridge parameter $\lambda > 0$ (λ-RF) and feature-to-datapoints ratio $\gamma = \frac{P}{N}$ is essentially equivalent to using a Kernel Ridge Regression with effective ridge $\tilde{\lambda} > \lambda$ ($\tilde{\lambda}$-KRR) which we characterize explicitly. More precisely, we have shown the following:

- The expectation of the λ-RF predictor is very close to the $\tilde{\lambda}$-KRR predictor (Theorem 8.4.1).
- The λ-RF predictor concentrates around its expectation when λ is bounded away from zero (Theorem H.3); this implies in particular that the test errors of the λ-RF and $\tilde{\lambda}$-KRR predictors are close to each other (Corollary H.3.16).

Both theorems are proven using tools from random matrix theory, in particular finite-size results on the concentration of the Stieltjes transform of general Wishart matrix models. While our current proofs require the assumption that the RF model is Gaussian, it seems natural to postulate that the results and the proofs extend to more general setups, along the lines of [145, 23].

Our numerical verifications on the expected λ-RF predictor and the $\tilde{\lambda}$-KRR predictor have shown that both are in excellent agreement. This shows in particular that in order to use RF predictors to approximate KRR predictors with a given ridge, one should choose both the number of features and the explicit ridge appropriately.

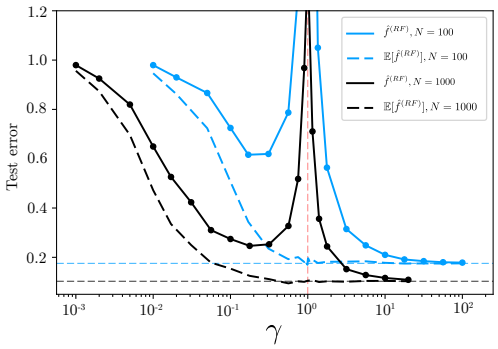


Figure 8.6.1: *Average test error of the λ-RF predictor for two values of N and $\lambda = 10^{-4}$.* For $N = 1000$, the test error is naturally lower and the cusp at $\gamma = 1$ is narrower than for $N = 100$. The experimental setup is the same as in Figure 8.3.1.

Finally, we investigate the ridgeless limit case $\lambda \searrow 0$. In this case, we see a sharp transition at $\gamma = 1$: in the overparameterized regime $\gamma > 1$, the effective ridge goes to zero, while in the underparameterized regime $\gamma < 1$, it converges to a positive value. At the interpolation threshold $\gamma = 1$, the variance of the λ-RF explodes, leading to the double descent curve emphasized in [1, 211, 18, 157]. We investigate this numerically and prove a lower bound yielding a plausible explanation for this phenomenon.

Chapter 9

Saddle-to-Saddle Dynamics in Deep Linear Networks: Small Initialization Training, Symmetry and Sparsity

Abstract

The dynamics of Deep Linear Networks (DLNs) is dramatically affected by the variance σ^2 of the parameters at initialization θ_0. For DLNs of width w, we show a phase transition w.r.t. the scaling γ of the variance $\sigma^2 = w^{-\gamma}$ as $w \to \infty$: for large variance ($\gamma < 1$), θ_0 is very close to a global minimum but far from any saddle point, and for small variance ($\gamma > 1$), θ_0 is close to a saddle point and far from any global minimum. While the first case corresponds to the well-studied NTK regime, the second case is less understood. This motivates the study of the case $\gamma \to +\infty$, where we conjecture a Saddle-to-Saddle dynamics: throughout training, gradient descent visits the neighborhoods of a sequence of saddles, each corresponding to linear maps of increasing rank, until reaching a sparse global minimum. We support this conjecture with a theorem for the dynamics between the first two saddles, as well as some numerical experiments.

9.1 Introduction

In spite of their widespread usage, the theoretical understanding of Deep Neural Networks (DNNs) remains limited. In contrast to more common statistical methods which are built (and proven) to recover the specific structure of the data, the development of DNNs techniques has been mostly driven by empirical results. This has led to a great variety of models which perform consistently well, but without a theory explaining why. In this paper, we provide a theoretical analysis of Deep Linear (Neural) Networks (DLNs), whose simplicity makes them particularly attractive as a first step towards the development of such a theory.

DLNs have a non-convex loss landscape and the behavior of training dynamics can be subtle. For shallow networks, the convergence of gradient descent is guaranteed by the fact that the saddles are strict and that all minima are global [13, 115, 131, 130]. In contrast, the deep case features non-strict saddles [115] and no general proof of convergence exists at the moment, though convergence to a global minimum can be guaranteed in some cases [4, 54].

A recent line of work focuses on the implicit bias of DLNs, and consistently reveals some form of incremental learning and implicit sparsity as in [75]. Diagonal networks are known to learn minimal L_1 solutions [156, 221]. With a specific initialization and the MSE loss, DLNs learn the singular components of the signal one by one [195, 1, 196, 74, 5]. Recently, it has been shown that with losses such as the cross-entropy and the exponential loss, the parameters diverge towards infinity, but end up following the direction of the max-margin classifier w.r.t. the L_p-Schatten (quasi)norm [80, 81, 209, 108, 109, 36, 147, 156, 235].

In parallel, recent works have shown the existence of two regimes in large-width DNNs: a kernel regime (also called NTK or lazy regime) where learning is described by the so-called Neural Tangent Kernel (NTK) guaranteeing linear convergence [105, 51, 34, 6, 128, 95], and an active regime where the dynamics is nonlinear [35, 183, 153, 151, 36]. For DLNs, both regimes can be observed as well, with evidence that while the linear regime exhibits no sparsity, the active regime favors solutions with some kind of sparsity [221, 156].

Contributions

We study deep linear networks $x \mapsto A_\theta x$ of depth $L \geq 1$ and widths $n_0, \cdots, n_L$, that is $A_\theta = W_L \cdots W_1$ where $W_1, \ldots, W_L$ are matrices such that $W_i \in \mathbb{R}^{n_i \times n_{i-1}}$ and θ is a vector that consists of all the (learnable) parameters of the DLN, i.e. the components of the matrices $W_1, \cdots, W_L$. For any general convex cost $C : \mathbb{R}^{n_L \times n_0} \to \mathbb{R}$ on matrices such that the zero matrix is not a global minimum, we investigate the gradient flow minimizing the loss $\mathcal{L}(\theta) = C(A_\theta)$. To ease the notation, suppose that the hidden layers have the same size, that is $w = n_1 = \cdots = n_{L-1}$ for some $w \in \mathbb{N}$.

The variance of the parameters at initialization has a profound effect on the training dynamics. If the parameters are initialized with variance $\sigma^2 = w^{-\gamma}$, where w is the size of the hidden layers, we observe a phase transition in the infinite width limit as $w \to \infty$ and show in Theorem 9.1 that:

- when $\gamma < 1$, the random initialization θ_0 is (with high probability) very close to a global minimum and very far from any saddle,
- when $\gamma > 1$, the initialization is very close to a saddle and far from any global minimum.

The case $\gamma < 1$ corresponds to the NTK regime (or kernel/lazy regime, described in Section 9.4) and the case $\gamma = 1$ corresponds to the Mean-Field limit (or the Maximal Update parametrization of [228]). It appears that the case $\gamma > 1$ has been much less studied in previous works.

To understand this regime, we investigate in Section 9.5 the case $\gamma \to +\infty$. More precisely, we fix the width of the network and let the variance at initialization go to zero. We show in Theorem 9.2 that the gradient flow trajectory asymptotically goes from the saddle at the origin $\vartheta^0 = 0$ to a rank-one saddle ϑ^1, i.e. a saddle where the matrices $W_1, \ldots, W_L$ are of rank 1. The proof is based on a new description (Theorem 9.4), in the spirit of the Hartman-Grobman theorem, of the so-called fast escape paths at the origin. This theorem may be of independent interest.

We propose the Conjecture 9.3, backed by numerical experiments, describing the full gradient flow when the variance at initialization is very small, suggesting that it goes from saddle to saddle, visiting the neighborhoods of a sequence of critical points $\vartheta^0, \ldots, \vartheta^K$ (the first K ones being saddle points, the last one being either a global minimum or a point at infinity) corresponding to matrices of increasing ranks. This is consistent with [75] which shows that incremental learning occurs in a toy model of DLNs and that gradient-based optimization hence has an implicit bias towards simple (sparse) solutions.

In Section 9.5, we show how this Saddle-to-Saddle dynamics can be described using a greedy low-rank algorithm which bears similarities with that of [135] and leads to a low-rank bias of the final learned function. This is in stark contrast to the NTK regime which features no low-rank bias.

Related Works

The existence of distinct regimes in the training dynamics of DNNs has been explored in previous works, both theoretically [34, 228] and empirically [70]. The theoretical works [34, 228] have mostly focused on the transition from the NTK regime ($\gamma < 1$) to the Mean-Field regime ($\gamma = 1$). This paper is focused on the regime beyond the critical one ($\gamma > 1$).

Our study of the Saddle-to-Saddle dynamics can also be understood as a generalization of the works [195, 1, 196, 74, 5] which describe a similar plateau effect in a very specific setting and with a very carefully chosen initialization.

Shortly after the initial publication of this article, we came aware of the paper [135] which provides a similar description to our Saddle-to-Saddle dynamics. For shallow networks, the results are almost equivalent, although the techniques are very different, especially when dealing with the fact that the escape directions (and escape paths) are unique only up to rotations. The paper [135] uses a clever trick that allows them to both study the dynamics of the output matrix $A_{\theta(t)}$, without the need to keep track of the parameters, and obtain a unicity property for the asymptotic dynamics. Instead, we focus on the dynamics of the parameters, give an identification of all optimal escape paths, and show that the path followed by the parameters' dynamics is unique up to symmetries of the network. Note also that, as in our paper, [135] only proves the first step of the Saddle-to-Saddle regime: for the subsequent steps, it is assumed that the next saddle is not approached along a ̀bad' direction (as we discuss in Section 9.5). For deep networks, our results are more general as they hold for more general initializations than in [135]. Indeed, in order to avoid the non-uniqueness problem of the escape paths in the space of parameters, their analysis relies heavily on the assumption that the weights of the network are balanced at initialization, and thus during training. Because we do not rely on this trick, our analysis does not require a balanced initialization.

9.2 Deep Linear Networks

Setup

A DLN of depth L and widths $n_0, \dots, n_L$ is the composition of L matrices

$$A_\theta = W_L \cdots W_1$$

where $W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$. The number of parameters is $P = \sum_{\ell=1}^{L} n_{\ell-1} n_\ell$ and we denote by $\theta = (W_L, \dots, W_1) \in \mathbb{R}^P$ the vector of parameters. The input dimension, resp. the output dimension is n_0, resp. n_L. All parameters are initialized as i.i.d. $\mathcal{N}(0, \sigma^2)$ Gaussian random variables.

We will focus on the so-called rectangular networks, in which the number of neurons in all hidden layers is the same, i.e. $n_1 = \cdots = n_{L-1} = w$. Such rectangular network is called a (L, w)-DLN, and its number of parameters is denoted by $P = P_{(L,w)} = n_0 w + (L-2)w^2 + w n_L$. The proofs given in this article can be extended to the non-rectangular case, but this leads to more complex notations.

We study the dynamics of gradient descent on the loss $\mathcal{L}(\theta) = C(A_\theta)$ for a general differentiable and convex cost C on $n_L \times n_0$ matrices. To ensure a non-trivial minimisation problem, we assume that the null matrix is not a global minimum of C : in this case, the origin in the parameter space

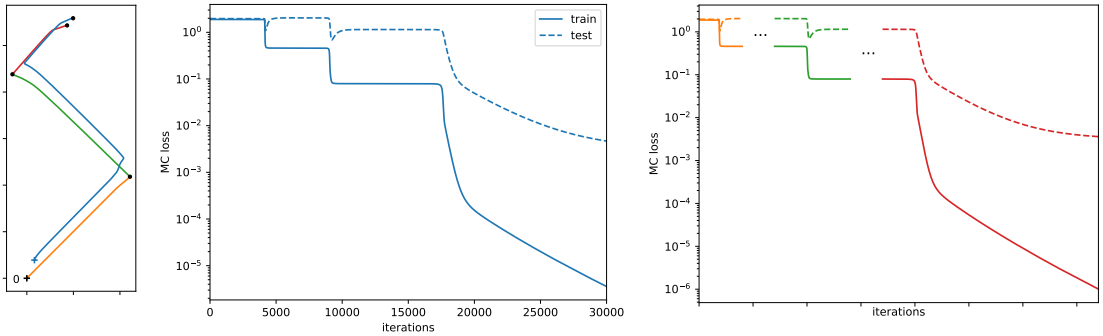


Figure 9.2.1: *Saddle-to-Saddle dynamics:* A DLN ($L = 4, w = 100$) with a small initialization ($\gamma = 2$) trained on a MC loss fitting a 10×10 matrix of rank 3. **Left:** Projection onto a plane of the gradient flow path θ_α in parameter space (in blue) and of the sequence of 3 paths $\theta^1, \theta^2, \theta^3$ (in orange, green and red), described by Algorithm $\mathcal{A}_{\epsilon,T,\eta}$, starting from the origin ($+$) and passing through 2 saddles ($\cdot$) before converging. **Middle:** Train (solid) and test (dashed) MC costs through training. We observe three plateaus, corresponding to the three saddles visited. **Right:** The train (solid) and test (dashed) losses of the three paths plotted sequentially, in the saddle-to-saddle limit; the dots represent an infinite amount of steps separating these paths.

is a saddle of $\mathcal{L}$. Given a starting point $\theta_0 \in \mathbb{R}^P$, we denote by $t \mapsto \Gamma(t, \theta_0)$ the gradient flow path on the cost $\mathcal{L}(\theta)$ starting from θ_0, i.e. $\Gamma(0, \theta_0) = \theta_0$ and $\partial_t \Gamma(t, \theta_0) = -\nabla \mathcal{L}(\Gamma(t, \theta_0))$.

While our analysis applies to general twice differentiable costs C, the typical costs used in practice are:

The *Mean-Squared Error* (MSE) loss $C(A) = \frac{1}{N} \|AX - Y\|_F^2$ for some inputs $X \in \mathbb{R}^{n_0 \times N}$ and labels $Y \in \mathbb{R}^{n_L \times N}$, where $|| \cdot ||_F$ is the Frobenius norm.

The *Matrix Completion* (MC) loss $C(A) = \frac{1}{N} \sum_{i=1}^{N} (A_{k_i,m_i} - A^*_{k_i,m_i})^2$ for some true matrix A^* of which we observe only the N entries $A^*_{k_1,m_1}, \dots, A^*_{k_N,m_N}$.

Symmetries and Invariance

A key tool in this paper is the use of two important symmetries of the parametrization map $\theta \mapsto A_\theta$ in DLNs: rotations of hidden layers and inclusions in wider DLNs.

Rotations: A $L-1$ tuple $R = (O_1, \dots, O_{L-1})$ of orthogonal $w \times w$ matrices is called a w-width network rotation, or in short a rotation. A rotation R acts on a parameter vector $\theta = (W_L, \dots, W_1)$ as $R\theta = (W_L O_{L-1}^T, O_{L-1} W_{L-1} O_{L-2}^T, \dots, O_1 W_1)$. The space of rotations is an important symmetry of DLN: indeed, for any parameter θ, and any cost C, the two following important properties hold:

$$A_{R\theta} = A_\theta, \quad \nabla_\theta C(A_{R\theta}) = R \nabla_\theta C(A_\theta),$$

where we considered $\nabla_\theta C(A_\theta) \in \mathbb{R}^{P_{L,w}}$ as another vector of parameters. These properties imply that if $\theta(t) = \Gamma(t, \theta_0)$ is a gradient flow path, then so is $R\theta(t) = \Gamma(t, R\theta_0)$.

Inclusion: The inclusion $I^{(w\to w')}$ of a network of width w into a network of width $w' > w$ (by adding zero weights on the new neurons) is defined as $I^{(w\to w')}(\theta) = (V_L, \dots, V_1)$ with

$$V_1 = \begin{pmatrix} W_1 \\ 0 \end{pmatrix}, V_\ell = \begin{pmatrix} W_\ell & 0 \\ 0 & 0 \end{pmatrix}, V_L = \begin{pmatrix} W_L & 0 \end{pmatrix}.$$

For any parameters θ and any cost C, we have $A_{I^{(w\to w')}\theta} = A_\theta$ and $\nabla C(A_{I^{(w\to w')}\theta}) = I^{(w\to w')}\nabla C(A_\theta)$: the image of the inclusion map $I^{(w\to w')}$ (as well as any rotation $R\mathrm{Im}I^{(w\to w')}$ thereof) is invariant under gradient flow.

9.3 Proximity of Critical Points at Initialization

It has already been observed that in the infinite width limit, when the width w of the network grows to infinity, the scale at which the variance σ^2 of the parameters at initialization scales with the width can lead to very different behaviors [34, 70, 228]. Let us consider scaling of the variance $\sigma^2 = w^{-\gamma}$ for $\gamma \geq 1 - \frac{1}{L}$. The reason we lower bound γ is that any smaller γ would lead to an explosion of the variance of the matrix A_θ at initialization as the width w grows.

Let d_{m} and d_{s} be the Euclidean distances between the initialization θ and, respectively, the set of global minima and the set of all saddles. For random variables $f(w), g(w)$ which depend on w, we write $f \asymp g$ if both $f(w)/g(w)$ and $g(w)/f(w)$ are stochastically bounded as $w \to \infty$. The following theorem studies how d_{m} and d_{s} scale as $w \to \infty$:

Theorem 9.1. *Suppose that the set of matrices that minimize C is non-empty, has Lebesgue measure zero, and does not contain the zero matrix. Let θ be i.i.d. centered Gaussian r.v. of variance $\sigma^2 = w^{-\gamma}$ where $1 - \frac{1}{L} \leq \gamma < \infty$. Then:*

1. *if $1 - \frac{1}{L} \leq \gamma < 1$, we have $d_{\mathrm{m}} \asymp w^{-\frac{(1-\gamma)(L-1)}{2}}$ and $d_{\mathrm{s}} \asymp w^{\frac{1-\gamma}{2}}$,*
2. *if $\gamma = 1$, we have $d_{\mathrm{m}}, d_{\mathrm{s}} \asymp 1$,*
3. *if $\gamma > 1$ we have $d_{\mathrm{m}} \asymp 1$ and $d_{\mathrm{s}} \asymp w^{-\frac{\gamma-1}{2}}$.*

This theorem shows an important change of behavior between the case $\gamma < 1$ and $\gamma > 1$. When $\gamma < 1$, the network is initialized very close to a global minimum and far from any saddle. When $\gamma > 1$, the parameters are initialized very close to a saddle but far away from any global minimum. The critical case $\gamma = 1$ is the unique limit where both types of critical points are at the same distance from the initialization.

Hence, the landscape of the loss near the initialization displays distinct features in the three regimes highlighted in the previous theorem. In fact, the dynamics of the gradient descent also exhibits very distinctive characteristics in the different regimes. In Appendix I.2, we show that the largest initialization, corresponding to the choice $\gamma = 1 - \frac{1}{L}$, is equivalent to the so-called NTK parametrization of [105], up to a rescaling of the learning rate. In the range $1 - \frac{1}{L} < \gamma < 1$, [229] obtain a similar, yet slightly different, kernel regime. The initialization $\gamma = 1$ corresponds to the Mean-Field limit for shallow networks [35, 183] or, more generally, to the Maximal Update parametrization [229] (see Appendix I.2). The case $\gamma > 1$ is however much less studied and is difficult to study since the initialization approaches a saddle as $w \to \infty$. Thus, in this regime, the wider the network, the longer it takes to escape this nearby saddle and, in the limit as $w \to \infty$, nothing happens over a finite number of gradient steps. With the right time parametrization, we

will observe interesting Saddle-to-Saddle dynamics in this regime, leading to some low-rank bias. This is regime is related to the condensed regime identified in [146].

9.4 NTK regime: $\gamma < 1$

The NTK for linear networks can be expressed easily using the tensor

$$\Theta^{(L)} = \sum_{\theta} \partial_\theta A \otimes \partial_\theta A,$$

which entries are given by $[\Theta^{(L)}]^{j,l}_{i,k} = (\nabla_\theta (A_\theta)_{i,j})^T (\nabla_\theta (A_\theta)_{k,l})$, for $i, k = 1, \dots, n_L$ and $j, l = 1, \dots, n_0$. For any x, y in $\mathbb{R}^{n_0}$, the value of the NTK at x and y is $\Theta^{(L)}(x \otimes y) = \sum_{j,l} [\Theta^{(L)}]^{j,l}_{:,:} x_j y_l$.

When the parameters evolve according to the gradient flow on $\mathcal{L}(\theta) = C(A_\theta)$, the dynamics of $A_{\theta(t)}$ is:

$$\begin{aligned} \partial_t A_{\theta(t)} &= -\Theta^{(L)} \cdot \nabla_A C(A_{\theta(t)}) \\ &= -\sum_{k,l} [\Theta^{(L)}]^{:,l}_{:,k} \frac{d}{dA_{k,l}} C(A_{\theta(t)}), \end{aligned}$$

where $\cdot$ denotes a contraction of the k, l indices of $\Theta^{(L)}$ with the two indices of $\nabla_A C(A_{\theta(t)})$.

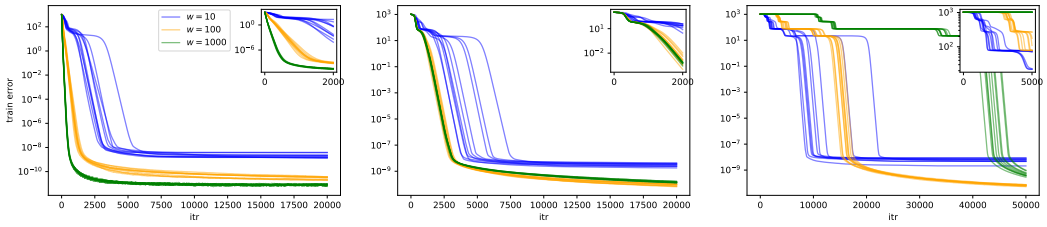
At initialization, $\Theta^{(L)}$ concentrates around its expectation $\mathbb{E}\left[\Theta^{(L)}\right] = L w^{(1-\gamma)(L-1)} \delta_{i,k} \delta_{j,l}$ as the width grows. It was first proven in [105] that for an initialization equivalent to the case $\gamma = 1 - \frac{1}{L}$ (see Appendix I.2 for more details), as $w \to \infty$ the NTK remains constant during training. Recent results [229] have shown that the NTK is asymptotically fixed for all $\gamma \in (1 - \frac{1}{L}, 1)$. In this case, given the asymptotic behavior of the NTK, the evolution of $A_{\theta(t)}$ is the same (up to a change of learning rate) as the one obtained by performing directly a gradient flow on the cost C.

As a result, in the regime $\gamma < 1$, if the cost C is strictly convex (or satisfies the Polyak-Lojasievicz inequality [142]), the loss decays exponentially fast. Besides, the depth of the network has no effect in the infinite width limit (except for a change of learning rate) and the DLN structure adds no specific bias to the global minimum learned with gradient descent. In particular, this regime leads to no low-rank bias.

9.5 Saddle-to-Saddle Dynamics: $\gamma \gg 1$

We now study the dynamics of DLN during training as the variance at initialization goes to zero. Specifically, we sample some random parameters θ_0 with i.i.d. $\mathcal{N}(0, 1)$ entries, consider the gradient flow $\theta_\alpha(t) = \Gamma(t, \alpha\theta_0)$, and let $\alpha \searrow 0$. Since the origin is a saddle, for all fixed times t, $\lim_{\alpha \searrow 0} \theta_\alpha(t) = 0$. We will show however that there is an escape time t_α, which grows to infinity as $\alpha \searrow 0$, such that the limit $\lim_{\alpha \searrow 0} \theta_\alpha(t_\alpha + t)$ is non-trivial for all $t \in \mathbb{R}$.

The study of shallow networks ($L = 2$) is facilitated by the fact that the saddle at the origin is strict: its Hessian has negative eigenvalues. For deeper networks ($L > 2$), the saddle is highly degenerate: the $L - 1$ first order derivatives vanish. In Section 9.5, we develop new theoretical tools to analyze the two types of saddles and their escape paths.



(a) NTK: $\gamma = 0.75$ (b) MF: $\gamma = 1.0$ (c) StS: $\gamma = 1.5$

(a) $\gamma = 0.75$ (NTK) **(b)** $\gamma = 1$ (MF) **(c)** $\gamma = 1.5$ (S-S)

Figure 9.4.1: *Training in* ***(a)*** *the NTK regime,* ***(b)*** *mean-field,* ***(c)*** *saddle-to-saddle regimes in deep linear networks for three widths* $w = 10, 100, 1000$*,* $L = 4$*, and* 10 *seeds.* Parameters are initialized with variance $\sigma^2 = w^{-\gamma}$. We observe that **(a)** in the NTK regime, the training loss shows typical linear convergence behavior for $w = 1000$ and $w = 100$; **(b)** in the mean-field regime, we observe that even the large width networks approach to a saddle at the beginning of the training and that the length of the plateaus remains constant between widths $w = 1000$ and $w = 100$; **(c)** in the saddle-to-saddle regime, the plateaus become longer as the width grows. In all cases, we see a reduction in the variation between the different seeds as $w \to \infty$.

First Path

It turns out that gradient flow paths naturally escape the saddle at the origin along so-called *optimal escape paths*. We say that a gradient flow path $\theta(\cdot) : \mathbb{R} \to \mathbb{R}^P$ is an *escape path* of a critical point θ^* if $\lim_{t\to-\infty} \theta(t) = \theta^*$. Informally, the optimal escape paths, whose precise definition is given in Section 9.5, are the escape paths that allow the fastest exit from a saddle. In DLNs, these optimal escape paths are of the form $RI^{(1\to w)}\underline{\theta}^1(t)$ where $\underline{\theta}^1(t)$ is a path of a width 1 DLN which escapes from the origin:

Theorem 9.2. *Assume that the largest singular value* s_1 *of the gradient of* C *at the origin* $\nabla C(0) \in \mathbb{R}^{n_L \times n_0}$ *has multiplicity 1. There is a deterministic gradient flow path* $\underline{\theta}^1$ *in the space of width-*1 *DLNs such that, with probability* 1 *if* $L \leq 3$*, and probability at least* $1/2$ *if* $L > 3$*, there exists an escape time* t^1_α *and a rotation* R *such that*

$$\lim_{\alpha\to 0} \theta_\alpha(t^1_\alpha + t) = RI^{(1\to w)}\underline{\theta}^1(t).$$

The unicity of the largest singular value of the gradient at the origin guarantees the unicity (up to rotation) of the optimal escape paths. For example, with the MSE loss, the gradient at the origin is $2YX^T$: for generic Y and X, the largest singular value of the gradient has a multiplicity of 1.

The reason why, for DLN with $L > 3$, we can only guarantee a probability of $\frac{1}{2}$ in the previous theorem, is that we need to ensure that gradient descent does not get stuck at the saddle at the origin or at other saddles connected to it. For $L = 2$, this follows from the fact that the saddle is strict. When $L > 2$, the saddle is not strict and we were only able to prove it in the case where $L = 3$. We conjecture that the behavior described in Theorem 9.2 happens with probability 1 for all $L \geq 2$.

As shown in the Appendix I.3, the escape time t_α is of order $-\log \alpha$ for shallow networks and of order $\alpha^{-(L-2)}$ for networks of depth $L > 2$. Hence, the deeper the network, the slower the gradient flow escapes the saddle.

Besides, as also discussed in the Appendix I.3, the norm $\left\|\theta^1(t)\right\|$ of the limiting escape path $\theta^1(t) = RI^{(1\to w)}\underline{\theta}^1(t)$ grows at an optimal speed: as $e^{s^*(t+T)}$ for some T when $L = 2$ and as $(s^*(L-2)(T-t))^{-\frac{1}{L-2}}$ for some T when $L > 2$, where s^* is the optimal escape speed $s^* = L^{-\frac{L-2}{2}}s_1$. These are optimal in the sense that given an other gradient flow path $\theta(t)$ which exits from the origin, there exists a ball B centered at the origin such that, for any small ϵ, if t_1 and t_2 are the times such that $\left\|\theta^1(t_1)\right\| = \epsilon = \|\theta(t_2)\|$, then $\left\|\theta^1(t+t_1)\right\| \geq \|\theta(t+t_2)\|$ for any positive t, until one of the paths exits the ball B.

Subsequent Paths

What happens after this first path? The width-1 gradient flow path $\underline{\theta}^1(t)$ converges to a width-1 critical point $\underline{\vartheta}^1$ as $t \to \infty$. While $\underline{\vartheta}^1$ may be a local minimum amongst width-1 DLNs, its inclusion $\vartheta^1 = RI^{(1\to w)}(\underline{\vartheta}^1)$ will be a saddle assuming it is not a global minimum already and that the network is wide enough, since if $w \geq \min\{n_0, n_L\}$ all critical points are either global minima or saddles [164].

Theorem 9.2 guarantees that, as $\alpha \searrow 0$, the gradient flow path $\theta_\alpha(t)$ will approach the saddle ϑ^1. It is then natural to assume that $\theta_\alpha(t)$ will escape this saddle along an optimal escape path (which is the inclusion of a width-2 path). Repeating this process, we expect gradient flow to converge as $\alpha \searrow 0$ to the concatenation of paths going from saddle to saddle of increasing width:

Conjecture 9.3. *With probability* 1*, there exist* $K+1$ *critical points* $\vartheta^0, \dots, \vartheta^K \in \mathbb{R}^{P_{L,w}}$ *(with* $\vartheta^0 = 0$*) and* K *gradient flow paths* $\theta^1, \dots, \theta^K : \mathbb{R} \to \mathbb{R}^{P_{L,w}}$ *connecting the critical points (i.e.* $\lim_{t\to-\infty}\theta^k(t) = \vartheta^{k-1}$ *and* $\lim_{t\to+\infty}\theta^k(t) = \vartheta^k$*) such that the path* $\theta_\alpha(t)$ *converges as* $\alpha \to 0$ *to the concatenation of* $\theta^1(t), \dots, \theta^K(t)$ *in the following sense: for all* $k < K$*, there exist times* t_α^k *(which depend on* θ_0*) such that*

$$\lim_{\alpha\to 0}\theta_\alpha(t_\alpha^k + t) = \theta^k(t).$$

Furthermore, for all $k < K$*, there is a deterministic path* $\underline{\theta}^k(t)$ *and a local minimum* $\underline{\vartheta}^k$ *of a width-*k *network such that for some rotation* R *(which depends on* θ_0*),* $\theta^k(t) = RI^{(k\to w)}(\underline{\theta}^k(t))$ *and* $\vartheta^k = RI^{(k\to w)}(\underline{\vartheta}^k)$ *for all* k *and* t*.*

This Saddle-to-Saddle behavior explains why for small initialization scale, the train error gets stuck at plateaus during training (Figures 9.2.1 and 9.4.1). Conjecture 9.3 suggests that these plateaus correspond to the saddle visited.

Note that for losses such as the cross-entropy, the gradient descent may diverge towards infinity, as studied in [209, 80]. From now on, we focus on the case where ϑ^K is a finite global minimum. By the invariance under gradient flow of $\mathrm{Im}[I^{(k\to w)}]$ (the image of the inclusion map), the inclusion of a width-k local minimum $\underline{\vartheta}^k$ into a larger network is a saddle ϑ^k (if A_{ϑ^k} is not a global minimum of C). These types of saddles are closely related to the symmetry-induced saddles studied in [204] in non-linear networks.

Remark 9.1. Note that each of the limiting paths θ^k and critical points ϑ^k will be *balanced* (i.e. their weight matrices satisfy $W_\ell W_\ell^T = W_{\ell+1}^T W_{\ell+1}$ for all $\ell = 1, \dots, L-1$). The origin is obviously balanced and since balancedness is an invariant of gradient flow and all other paths and saddles are connected to the origin by a sequence of gradient flow paths, they must be balanced too. Note however that for all $\alpha > 0$, the path $\theta_\alpha(t)$ is almost surely not balanced.

Algorithm $\mathcal{A}_{\epsilon,T,\eta}$

Compute the first singular vectors of $\nabla C(0)$:
$u, s, v \leftarrow \mathrm{SVD}_1(\nabla C(0))$
$\theta \leftarrow (-\epsilon v^T, \epsilon, \dots, \epsilon u)$
$w \leftarrow 1$
while $C(A_\theta) < C_{min} + \epsilon$ **do**
  # T steps of GD on the loss of width-w DLN with lr η
  $\theta \leftarrow \mathrm{SGD}_{w,T,\eta}(\theta)$
  $u, s, v \leftarrow \mathrm{SVD}_1(\nabla C(A_\theta))$
  $\theta \leftarrow \left(\begin{pmatrix} W_1 \\ -\epsilon v^T \end{pmatrix}, \begin{pmatrix} W_2 & 0 \\ 0 & \epsilon \end{pmatrix}, \dots, \begin{pmatrix} W_L & \epsilon u \end{pmatrix} \right)$
  $w \leftarrow w + 1$
end while

Greedy Low-Rank Algorithm

Conjecture 9.3 suggests that the gradient flow with vanishing initialization implements a greedy low-rank algorithm which performs a greedy search for a lowest-rank solution: it first tries to fit a width 1 network, then a width 2 network and so on until reaching a solution. Thus, we expect that as $\alpha \searrow 0$, the dynamics of gradient flow corresponds, up to inclusion and rotation, to the limit of the algorithm $\mathcal{A}_{\epsilon,T,\eta}$ as sequentially $T \to \infty$, $\eta \to 0$ and $\epsilon \to 0$. In particular, we used the Algorithm $\mathcal{A}_{\epsilon,T,\eta}$, with large T and small η and ϵ to approximate the paths $\underline{\theta}^k$ and points $\underline{\vartheta}^k$ in Figure 9.2.1. Note how this limiting algorithm is deterministic. This implies that even for finite widths the dynamics of gradient flow converge to a deterministic limit (up to random rotations R) as the variance at initialization goes to zero.

A similar algorithm has already been described in [135], however thanks to our different proof techniques, we are able to give a more precise description of the evolution of the parameters.

Description of the paths that escape a saddle

Our proof relies on a theorem which relates the escape paths of the saddle at the origin of the cost $\mathcal{L}$ and the escape paths of the L-th order Taylor approximation H of $\mathcal{L}$. This correspondence only applies to paths which escape the saddle sufficiently fast.

We define the set of fast escaping paths $\mathcal{F}_{\mathcal{L}}(s)$ of the cost $\mathcal{L}$ with speed at least s as follows:

- for shallow networks ($L = 2$), it is the set of gradient flow paths that satisfy $\|\theta(t)\| = O(e^{st})$ as $t \to -\infty$,
- for deep networks ($L > 2$), it is the set of gradient flow paths that satisfy $\|\theta(t)\| \leq (s(L-2)(T-t))^{-\frac{1}{L-2}}$ for some T and any small enough t.

The optimal escape speed is $s^* = L^{-\frac{L-2}{2}} s_1$ where s_1 is the largest singular value of $\nabla C(0)$. It is the optimal escape speed in the sense that there are no faster escape paths: $\mathcal{F}_{\mathcal{L}}(s) = \emptyset$ if $s > s^*$. Escape paths which exit the saddle at the optimal escape speed are called optimal escape paths.

There is a bijection between fast escaping paths of the loss $\mathcal{L}$ and those of its Lth order Taylor approximation H:

Theorem 9.4. ***Shallow networks:*** *for all s s.t. $s > \frac{1}{3}s^*$ there is a unique bijection $\Psi : \mathcal{F}_{\mathcal{L}}(s) \to \mathcal{F}_H(s)$ such that for all paths $\theta \in \mathcal{F}_{\mathcal{L}}(s)$, $\|\theta(t) - \Psi(\theta)(t)\| = O(e^{3st})$ as $t \to -\infty$.*

Deep networks: *for all $s > \frac{L-1}{L+1}s^*$, there is a unique bijection $\Psi : \mathcal{F}_{\mathcal{L}}(s) \to \mathcal{F}_H(s)$ such that for all paths $\theta \in \mathcal{F}_{\mathcal{L}}(s)$, $\|\theta(t) - \Psi(\theta)(t)\| = O((-t)^{-\frac{L+1}{L-2}})$ as $t \to -\infty$.*

We believe that this theorem is of independent interest, and it is stated in a more general setting in the Appendix. Theorem 9.4 is similar to the Hartman-Grobman Theorem, which shows a bijection, in the vicinity of a critical point, between the gradient flow paths of $\mathcal{F}$ and of its linearization. The bijection in Theorem 9.4 holds only between fast escaping paths, but it gives stronger guarantees regarding how close the paths $\theta(\cdot)$ and $\Psi(\theta)(\cdot)$ are. In particular, Theorem 9.4 guarantees that a fast escaping path $\theta(\cdot)$ and its image $\Psi(\theta)(\cdot)$ have the same 'escape speed', whereas the correspondence between paths of in the Hartman-Grobman theorem does not in general conserve speed. This is due to the fact that the homeomorphism which allows to construct the bijection in the Hartman-Grobman theorem is only Hölder continuous. This suggests that fast escaping paths can be guaranteed to conserve their speed after the Taylor approximation while slower paths can change speed. Finally, our result has the significant advantage that it may be applied to higher order Taylor approximations, whereas the Hartman-Grobman Theorem only applies to the linearization of the flow (i.e. it could only be useful in the shallow case $L = 2$).

Sketch of Proof

In this section, we provide a sketch of proof for Theorem 9.2.

We fix some small $r > 0$ independent of α. The *escape time* t_α is the earliest time such that $\|\theta_\alpha(t_\alpha)\| = r$. We show that the limiting escape path $(\theta^1(t))_{t\in\mathbb{R}}$ as $\theta^1(t) = \lim_{\alpha\searrow 0} \theta_\alpha(t_\alpha + t)$ is well defined and non-trivial since $\theta^1(0) \neq 0$. The next step of the proof is to show that θ^1 escapes the saddle at an almost optimal speed: for any $\epsilon > 0$, for some T and any small enough t, for shallow network $\left\|\theta^1(t)\right\| = O(e^{(s^*-\epsilon)t})$, and for deeper networks $\left\|\theta^1(t)\right\| \leq [(L-2)(s^*-\epsilon)(T-t)]^{-\frac{1}{L-2}}$. We may therefore apply Theorem 9.4: there exists a unique optimal escape path for the L-th order Taylor approximation H around the origin which is 'close', in the sense given in Theorem 9.4, to θ^1.

For the Taylor approximation H, we have a precise description of the optimal escaping paths for the saddle at the origin. Assuming that the largest singular value s_1 of the gradient matrix $\nabla C(0)$ has multiplicity one, all optimal escape paths of H (i.e. the set of paths that escape with the largest speed) are of the form $\theta_H(t) = d(t)RI^{(1\to w)}(\rho)$ where R is some rotation, the scalar function $d(t)$ is equal to $e^{s^*(t+T)}$ for shallow networks and $(s^*(L-2)(T-t))^{-\frac{1}{L-2}}$ for deep networks, and the vector of parameters ρ is given by:

$$\rho = \left(v_1^T, 1, \ldots, 1, u_1\right)$$

with u_1, v_1 the left and right singular vectors of the largest singular value s_1 of the gradient matrix $\nabla C(0)$.

Let us consider the unique optimal escape path $\theta_H(t) = d(t)RI^{(1\to w)}(\rho)$ for H which is 'close' to θ^1. The path $\underline{\theta}_H(t) = d(t)\rho$ is also an optimal escape path for H: from Theorem 9.4, there exists a unique optimal escape path $\underline{\theta}(t)$ which is 'close' to $\underline{\theta}_H$. The former escape path corresponds to a 1-width DLN and it is easy to show that $RI^{(1\to w)}(\underline{\theta})$ is an optimal escape path for $\mathcal{L}$ which is 'close' to $RI^{(1\to w)}(\underline{\theta}_H) = \theta_H$.

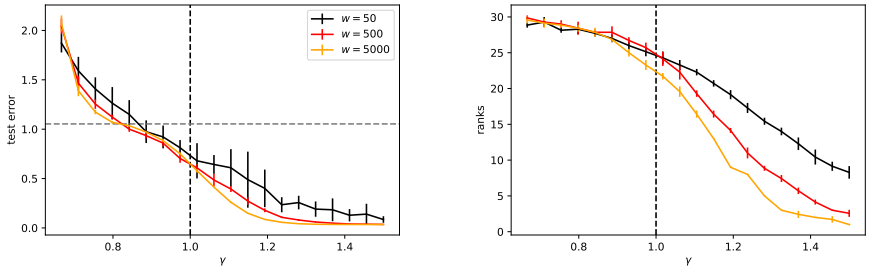


Figure 9.6.1: *Test errors and ranks at convergence as a function of initialization scale γ, matrix completion task.* The task is finding a matrix of size 30×30 and rank 1 from 20% of its entries. The test error and ranks are averaged over 7 seeds (± 1 standard deviations are reported in the error bar). In the NTK regime, the solutions at convergence are almost full-rank and the test error is roughly the same or worse than that of the zero predictor. On the other hand in the Saddle-to-Saddle regime the test error approaches zero. As the width grows the transition between regimes becomes sharper and the test error becomes more consistent within each regimes.

In particular, we obtain that both θ^1 and $RI^{(1\to w)}(\underline{\theta})$ are optimal escape path for $\mathcal{L}$ which are 'close' to θ_H. By the unicity property in Theorem 9.4, we obtain that $\theta^1 = RI^{(1\to w)}(\underline{\theta})$ which allows us to conclude.

Remark 9.2. To prove Conjecture 9.3, one needs to apply a similar argument to understand how gradient flow escapes the subsequent saddles $\vartheta^1, \dots, \vartheta^K$. There are two issues:

First, even though Theorem 9.2 guarantees that gradient descent will come arbitrarily close to the next saddle ϑ^1, it may not approach it along a generic direction: it could approach along a "bad" direction. For the first path, we relied on the fact that θ_0 is Gaussian to guarantee that these bad directions are avoided with probability 1 (or $1/2$). Note that this problem c

ould be addressed using the so-called perturbed stochastic gradient descent described in [111, 50] since, in this learning algorithm, once in the vicinity of the saddle, a small Gaussian noise is added to the parameters: as a consequence, they end up being in a generic position in the neighborhood of the saddle.

Second, for deep networks ($L > 2$), the saddle ϑ^1 has a different local structure to ϑ^0. Indeed, at the origin, the $L-1$ first derivatives vanish, leading to an (approximately) L-homogeneous saddle at the origin. On the contrary, at the rank 1 saddle $\vartheta^1 = RI^{(1\to w)}(\underline{\vartheta}^1)$, if $\underline{\vartheta}^1$ is a local minimum of the width 1 network, the Hessian is positive along the inclusion $\mathrm{Im}\left[RI^{(1\to w)}\right]$. This implies that the dynamics can only escape the saddle through the Hessian null-space, along which the first $L-1$ derivatives vanish. Although the loss restricted to this null-space around ϑ_1 has a similar structure to the loss around the origin, the fact that the Hessian at ϑ_1 is not null complexifies the analysis.

9.6 Characterization of the Regimes of Training

In light of the results presented in this paper, we discuss the three regimes that can be obtained by varying the initialization scale γ: the kernel regime ($\gamma < 1$), the Mean-Field regime ($\gamma = 1$) and the Saddle-to-Saddle regime ($\gamma > 1$).

The NTK limit ($\gamma = 1 - \frac{1}{L}$) [105, 128] is representative of the other scalings $1 - \frac{1}{L} \leq \gamma < 1$ [229]. The critical regime $\gamma = 1$ corresponds to the Mean-Field limit for shallow networks [35, 183] or the Maximal Update parametrization for deep networks [229]. Finally, we conjecture that the last regime where $\gamma > 1$, displays features very akin to the $\gamma = +\infty$ case studied in this article. Under this assumption, we obtain the following list of properties that characterize each of these regimes:

In the **NTK regime** ($1 - \frac{1}{L} \leq \gamma < 1$):

1. During training, the parameters converge to a nearby global minimum, and do not approach any saddle (Figure 9.4.1a shows how the plateaus disappear as w grows).

2. If the cost on matrices C is strictly convex, one can guarantee exponential decrease of the loss (i.e. linear convergence).

3. The NTK is asymptotically fixed during training.

4. No low-rank bias in the learned matrix - as a result the test error for matrix completion is the same (or even larger) than the zero predictor in the NTK regime, as shown in Figure 9.6.1.

The **Saddle-to-Saddle regime** ($\gamma > 1$):

1. The parameters start in the vicinity of a saddle and visit a sequence of saddles during training. They come closer to each of these saddles as the width grows.

2. As the width grows, it takes longer to escape each saddle, leading to long plateaus for the training error. The training time is therefore asymptotically infinite (see Figure 9.4.1c).

3. The rate of change $\|\Theta(\theta_T) - \Theta(\theta_0)\|$ (where $T \in \mathbb{R}$ is the stopping time) of the NTK is infinitely larger than the NTK at initialization $\|\Theta(\theta_0)\|$. This follows from the fact that the NTK at initialization goes to zero, while it has finite size at the end of training.

4. The learned matrix is the result of a greedy algorithm that finds the lowest rank solution.

The **Mean-Field regime** $\gamma = 1$ lies at the transition between the two previous regimes and is more difficult to characterize:

1. In this critical regime, the constant factor c in the variance at initialization $\sigma^2 = cw^{-\gamma}$ can have a strong effect on the dynamics.

2. Plateaus can still be observed (see Figure 9.4.1b), however in contrast to the Saddle-to-Saddle regime, the length of the plateaus does not increase as the width grows, but remains roughly constant.

3. The NTK and its rate of change are of same order.

In general, we observe some tradeoff: the NTK regime leads to fast convergence without low-rank bias, while the Saddle-to-Saddle regime leads to some low-rank bias, but at the cost of an asymptotically infinite training time.

9.7 Conclusion

We propose a simple criterion to identify three regimes in the training of large DLNs: the distances from the initialization to the nearest global minimum and to the nearest saddle. The NTK regime ($1 - \frac{1}{L} \leq \gamma < 1$) is characterized by an initialization which is close to a global minimum and far from any saddle, the Saddle-to-Saddle regime ($\gamma > 1$) is characterized by an initialization which is close to a saddle and (comparatively) far from any global minimum and, finally, in the critical Mean-Field regime ($\gamma = 1$), these two distances are of the same order as the width grows.

While the NTK and Mean-Field limits are well-studied, the Saddle-to-Saddle regime is less understood. We therefore investigate the case $\gamma = +\infty$ (i.e. we fix the width and let the variance at initialization go to zero). In this limit, the initialization converges towards the saddle at the origin $\vartheta^0 = 0$. We show that gradient flow naturally escapes this saddle along an 'optimal escape path' along which the network behaves as a width-1 network. This leads the gradient flow to subsequently visit a second saddle ϑ^1 which has the property that the matrix A_{ϑ^1} has rank 1. We conjecture that the gradient flow next visits a sequence of critical points $\vartheta^2, \dots, \vartheta^K$ of increasing rank, implementing some form of greedy low-rank algorithm. These saddles explain the plateaus in the loss curve which are characteristic of the Saddle-to-Saddle regime.

Similar plateaus can be observed in non-linear networks: this suggests that the regimes and dynamics described in this paper could be generalized to non-linear networks.

Appendix A

General Appendix

A.1 Simple Bound on the Variance of the Random Feature Predictor

In this Appendix we prove the bound on the variance $\mathrm{Var}\left(\hat{f}^{RF}_{\lambda,P}(x)\right)$ of the Random Feature predictor $\hat{f}^{RF}_{\lambda,P}(x)$.

Lemma A.1. *We have*

$$\mathrm{Var}\left(\hat{f}^{RF}_{\lambda,P}(x)\right) \leq \frac{\|Y\|^2}{N^2\lambda^2 P}\left[\left\|K(x,X)K(X,X)^{-1}\right\|^2\left(\|K(X,X)\|_F^2 + \left(\mathrm{Tr}K(X,X)\right)^2\right) + \mathrm{Tr}K(X,X)K(x,x)\right].$$

Proof. We know that

$$\mathrm{Var}\left(\hat{f}^{RF}_{\lambda,P}(x)\right) = \mathrm{Var}\left(K(x,X)K(X,X)^{-1}\tilde{K}(X,X)\left(\tilde{K}(X,X)+N\lambda I_N\right)^{-1}Y\right) + \frac{\mathbb{E}\left[\|\theta\|^2\right]}{P}\left(K(x,x)-K(x,X)K(X,X)^{-1}K(X,x)\right).$$

The first term $\mathrm{Var}\left(K(x,X)K(X,X)^{-1}\tilde{K}(X,X)\left(\tilde{K}(X,X)+N\lambda I_N\right)^{-1}Y\right)$ can bounded by

$$\left\|K(x,X)K(X,X)^{-1}\right\|^2\mathbb{E}\left[\left\|\tilde{K}(X,X)\left(\tilde{K}(X,X)+N\lambda I_N\right)^{-1}Y - K(X,X)\left(K(X,X)+N\lambda I_N\right)^{-1}Y\right\|^2\right].$$

Since $\tilde{K}(X,X)\left(\tilde{K}(X,X)+N\lambda I\right)^{-1} = I_N - N\lambda\left(\tilde{K}(X,X)+N\lambda I\right)^{-1}$ we can bound the expectation in the above by

$$N^2\lambda^2\mathbb{E}\left[\left\|\left[\left(\tilde{K}(X,X)+N\lambda I_N\right)^{-1} - \left(K(X,X)+N\lambda I_N\right)^{-1}\right]Y\right\|^2\right].$$

Since $(A+N\lambda I_N)^{-1} - (B+N\lambda I_N)^{-1} = (A+N\lambda I_N)^{-1}(B-A)(B+N\lambda I_N)^{-1}$ for any matrices A, B, we get the bound

$$\mathbb{E}\left[\left\|K(X,X)-\tilde{K}(X,X)\right\|_F^2\right]\frac{\|Y\|^2}{\lambda^2} = \frac{\|K(X,X)\|_F^2 + \left(\mathrm{Tr}K(X,X)\right)^2}{P}\frac{\|Y\|^2}{N^2\lambda^2}.$$

In the second term $\frac{\mathbb{E}\left[\|\theta\|^2\right]}{P}\left(K(x,x) - K(x,X)K(X,X)^{-1}K(X,x)\right)$, we only need to compute the expected parameter norm $\mathbb{E}\left[\|\theta\|^2\right]$. We have

$$\begin{aligned}\mathbb{E}\left[\|\theta\|^2\right] &= Y^T\mathbb{E}\left[\tilde{K}(X,X)\left(\tilde{K}(X,X) + N\lambda I_N\right)^{-2}\right]Y \\ &\leq \frac{\mathbb{E}\left[\left\|\tilde{K}\right\|_{op}\right]}{N^2\lambda^2}\|Y\|^2\end{aligned}$$

and the operator norm $\left\|\tilde{K}\right\|_{op}$ is bounded by the trace $\mathrm{Tr}\tilde{K}(X,X)$ with mean $\mathrm{Tr}K(X,X)$. Putting it all together, we obtain

$$\mathrm{Var}\left(\hat{f}^{RF}_{\lambda,P}(x)\right) \leq \frac{\|Y\|^2}{N^2\lambda^2 P}\left[\left\|K(x,X)K(X,X)^{-1}\right\|^2\left(\|K(X,X)\|_F^2 + \left(\mathrm{Tr}K(X,X)\right)^2\right) + \mathrm{Tr}K(X,X)K(x,x)\right],$$

where we used the fact that $K(x,X)K(X,X)^{-1}K(X,x) \geq 0$. □

Appendix B

Neural Tangent Kernel: Convergence and Generalization in Neural Networks

B.1 Appendix

This appendix is dedicated to proving the key results of this paper, namely Proposition B.1 and Theorems B.1 and B.2, which describe the asymptotics of neural networks at initialization and during training.

We study the limit of the NTK as $n_1, ..., n_{L-1} \to \infty$ sequentially, i.e. we first take $n_1 \to \infty$, then $n_2 \to \infty$, etc. This leads to much simpler proofs, but our results could in principle be strengthened to the more general setting when $\min(n_1, ..., n_{L-1}) \to \infty$.

A natural choice of convergence to study the NTK is with respect to the operator norm on kernels:

$$\|K\|_{op} = \max_{\|f\|_{p^{in}} \leq 1} \|f\|_K = \max_{\|f\|_{p^{in}} \leq 1} \sqrt{\mathbb{E}_{x,x'}[f(x)^T K(x,x') f(x')]},$$

where the expectation is taken over two independent $x, x' \sim p^{in}$. This norm depends on the input distribution p^{in}. In our setting, p^{in} is taken to be the empirical measure of a finite dataset of distinct samples $x_1, ..., x_N$. As a result, the operator norm of K is equal to the leading eigenvalue of the $Nn_L \times Nn_L$ Gram matrix $(K_{kk'}(x_i, x_j))_{k,k' < n_L, i,j \leq N}$. In our setting, convergence in operator norm is hence equivalent to pointwise convergence of K on the dataset.

Asymptotics at Initialization

It has already been observed [159, 126] that the output functions $f_{\theta,i}$ for $i = 1, ..., n_L$ tend to iid Gaussian processes in the infinite-width limit.

Proposition B.1. *For a network of depth L at initialization, with a Lipschitz nonlinearity σ, and in the limit as $n_1, ..., n_{L-1} \to \infty$ sequentially, the output functions $f_{\theta,k}$, for $k = 1, ..., n_L$, tend (in law) to iid centered Gaussian processes of covariance $\Sigma^{(L)}$, where $\Sigma^{(L)}$ is defined recursively by:*

$$\Sigma^{(1)}(x, x') = \frac{1}{n_0} x^T x' + \beta^2$$
$$\Sigma^{(L+1)}(x, x') = \mathbb{E}_f[\sigma(f(x))\sigma(f(x'))] + \beta^2,$$

taking the expectation with respect to a centered Gaussian process f of covariance $\Sigma^{(L)}$.

Proof. We prove the result by induction. When $L = 1$, there are no hidden layers and f_θ is a random affine function of the form:

$$f_\theta(x) = \frac{1}{\sqrt{n_0}} W^{(0)} x + \beta b^{(0)}.$$

All output functions $f_{\theta,k}$ are hence independent and have covariance $\Sigma^{(1)}$ as needed.

The key to the induction step is to consider an $(L+1)$-network as the following composition: an L-network $\mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ mapping the input to the pre-activations $\tilde{\alpha}_i^{(L)}$, followed by an elementwise application of the nonlinearity σ and then a random affine map $\mathbb{R}^{n_L} \to \mathbb{R}^{n_{L+1}}$. The induction hypothesis gives that in the limit as sequentially $n_1, ..., n_{L-1} \to \infty$ the preactivations $\tilde{\alpha}_i^{(L)}$ tend to iid Gaussian processes with covariance $\Sigma^{(L)}$. The outputs

$$f_{\theta,i} = \frac{1}{\sqrt{n_L}} W_i^{(L)} \alpha^{(L)} + \beta b_i^{(L)}$$

conditioned on the values of $\alpha^{(L)}$ are iid centered Gaussians with covariance

$$\tilde{\Sigma}^{(L+1)}(x, x') = \frac{1}{n_L} \alpha^{(L)}(x; \theta)^T \alpha^{(L)}(x'; \theta) + \beta^2.$$

By the law of large numbers, as $n_L \to \infty$, this covariance tends in probability to the expectation

$$\tilde{\Sigma}^{(L+1)}(x, x') \to \Sigma^{(L+1)}(x, x') = \mathbb{E}_{f \sim \mathcal{N}(0, \Sigma^{(L)})}[\sigma(f(x))\sigma(f(x'))] + \beta^2.$$

In particular the covariance is deterministic and hence independent of $\alpha^{(L)}$. As a consequence, the conditioned and unconditioned distributions of $f_{\theta,i}$ are equal in the limit: they are iid centered Gaussian of covariance $\Sigma^{(L+1)}$. □

In the infinite-width limit, the neural tangent kernel, which is random at initialization, converges in probability to a deterministic limit.

Theorem B.1. *For a network of depth L at initialization, with a Lipschitz nonlinearity σ, and in the limit as the layers width $n_1, ..., n_{L-1} \to \infty$ sequentially, the NTK $\Theta^{(L)}$ converges in probability to a deterministic limiting kernel:*

$$\Theta^{(L)} \to \Theta_\infty^{(L)} \otimes Id_{n_L}.$$

The scalar kernel $\Theta_\infty^{(L)} : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}$ is defined recursively by

$$\Theta_\infty^{(1)}(x, x') = \Sigma^{(1)}(x, x')$$
$$\Theta_\infty^{(L+1)}(x, x') = \Theta_\infty^{(L)}(x, x')\dot{\Sigma}^{(L+1)}(x, x') + \Sigma^{(L+1)}(x, x'),$$

where

$$\dot{\Sigma}^{(L+1)}(x, x') = \mathbb{E}_{f \sim \mathcal{N}\left(0, \Sigma^{(L)}\right)}\left[\dot{\sigma}\left(f\left(x\right)\right)\dot{\sigma}\left(f\left(x'\right)\right)\right],$$

taking the expectation with respect to a centered Gaussian process f of covariance $\Sigma^{(L)}$, and where $\dot{\sigma}$ denotes the derivative of σ.

Proof. The proof is again by induction. When $L = 1$, there is no hidden layer and therefore no limit to be taken. The neural tangent kernel is a sum over the entries of $W^{(0)}$ and those of $b^{(0)}$:

$$\begin{aligned}\Theta_{kk'}(x,x') &= \frac{1}{n_0}\sum_{i=1}^{n_0}\sum_{j=1}^{n_1} x_i x'_i \delta_{jk}\delta_{jk'} + \beta^2 \sum_{j=1}^{n_1}\delta_{jk}\delta_{jk'} \\ &= \frac{1}{n_0} x^T x' \delta_{kk'} + \beta^2 \delta_{kk'} = \Sigma^{(1)}(x,x')\delta_{kk'}.\end{aligned}$$

Here again, the key to prove the induction step is the observation that a network of depth $L+1$ is an L-network mapping the inputs x to the preactivations of the L-th layer $\tilde{\alpha}^{(L)}(x)$ followed by a nonlinearity and a random affine function. For a network of depth $L+1$, let us therefore split the parameters into the parameters $\tilde{\theta}$ of the first L layers and those of the last layer $(W^{(L)}, b^{(L)})$.

By Proposition B.1 and the induction hypothesis, as $n_1, ..., n_{L-1} \to \infty$ the pre-activations $\tilde{\alpha}_i^{(L)}$ are iid centered Gaussian with covariance $\Sigma^{(L)}$ and the neural tangent kernel $\Theta_{ii'}^{(L)}(x,x')$ of the smaller network converges to a deterministic limit:

$$\left(\partial_{\tilde{\theta}} \tilde{\alpha}_i^{(L)}(x;\theta)\right)^T \partial_{\tilde{\theta}} \tilde{\alpha}_{i'}^{(L)}(x';\theta) \to \Theta_\infty^{(L)}(x,x')\delta_{ii'}.$$

We can split the neural tangent network into a sum over the parameters $\tilde{\theta}$ of the first L layers and the remaining parameters $W^{(L)}$ and $b^{(L)}$.

For the first sum let us observe that by the chain rule:

$$\partial_{\tilde{\theta}_p} f_{\theta,k}(x) = \frac{1}{\sqrt{n_L}}\sum_{i=1}^{n_L} \partial_{\tilde{\theta}_p}\tilde{\alpha}_i^{(L)}(x;\theta)\dot{\sigma}(\tilde{\alpha}_i^{(L)}(x;\theta))W_{ik}^{(L)}.$$

By the induction hypothesis, the contribution of the parameters $\tilde{\theta}$ to the neural tangent kernel $\Theta_{kk'}^{(L+1)}(x,x')$ therefore converges as $n_1, ..., n_{L-1} \to \infty$:

$$\begin{aligned}&\frac{1}{n_L}\sum_{i,i'=1}^{n_L}\Theta_{ii'}^{(L)}(x,x')\dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(x;\theta)\right)\dot{\sigma}\left(\tilde{\alpha}_{i'}^{(L)}(x';\theta)\right)W_{ik}^{(L)}W_{i'k'}^{(L)} \\ &\to \frac{1}{n_L}\sum_{i=1}^{n_L}\Theta_\infty^{(L)}(x,x')\dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(x;\theta)\right)\dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(x';\theta)\right)W_{ik}^{(L)}W_{ik'}^{(L)}\end{aligned}$$

By the law of large numbers, as $n_L \to \infty$, this tends to its expectation which is equal to

$$\Theta_\infty^{(L)}(x,x')\dot{\Sigma}^{(L+1)}(x,x')\delta_{kk'}.$$

It is then easy to see that the second part of the neural tangent kernel, the sum over $W^{(L)}$ and $b^{(L)}$ converges to $\Sigma^{(L+1)}\delta_{kk'}$ as $n_1, ..., n_L \to \infty$. □

Asymptotics during Training

Given a training direction $t \mapsto d_t \in \mathcal{F}$, a neural network is trained in the following manner: the parameters θ_p are initialized as iid $\mathcal{N}(0,1)$ and follow the differential equation:

$$\partial_t \theta_p(t) = \left\langle \partial_{\theta_p} F^{(L)}, d_t \right\rangle_{p^{in}}.$$

In this context, in the infinite-width limit, the NTK stays constant during training:

Theorem B.2. *Assume that σ is a Lipschitz, twice differentiable nonlinearity function, with bounded second derivative. For any T such that the integral $\int_0^T \|d_t\|_{p^{in}} dt$ stays stochastically bounded, as $n_1, ..., n_{L-1} \to \infty$ sequentially, we have, uniformly for $t \in [0, T]$,*

$$\Theta^{(L)}(t) \to \Theta_\infty^{(L)} \otimes Id_{n_L}.$$

As a consequence, in this limit, the dynamics of f_θ is described by the differential equation

$$\partial_t f_{\theta(t)} = \Phi_{\Theta_\infty^{(L)} \otimes Id_{n_L}} \left(\langle d_t, \cdot \rangle_{p^{in}} \right).$$

Proof. As in the previous theorem, the proof is by induction on the depth of the network. When $L = 1$, the neural tangent kernel does not depend on the parameters, it is therefore constant during training.

For the induction step, we again split an $L+1$ network into a network of depth L with parameters $\tilde{\theta}$ and top layer connection weights $W^{(L)}$ and bias $b^{(L)}$. The smaller network follows the training direction

$$d'_t = \dot{\sigma}\left(\tilde{\alpha}^{(L)}(t)\right) \left(\frac{1}{\sqrt{n_L}} W^{(L)}(t) \right)^T d_t$$

for $i = 1, \dots, n_L$, where the function $\tilde{\alpha}_i^{(L)}(t)$ is defined as $\tilde{\alpha}_i^{(L)}(\cdot; \theta(t))$. We now want to apply the induction hypothesis to the smaller network. For this, we need to show that $\int_0^T \|d'_t\|_{p^{in}} \mathrm{d}t$ is stochastically bounded as $n_1, \dots, n_L \to \infty$. Since σ is a c-Lipschitz function, we have that

$$\|d'_t\|_{p^{in}} \leq c \| \frac{1}{\sqrt{n_L}} W^{(L)}(t) \|_{op} \|d_t\|_{p^{in}}.$$

To apply the induction hypothesis, we now need to bound $\| \frac{1}{\sqrt{n_L}} W^{(L)}(t) \|_{op}$. For this, we use the following lemma, which is proven in Appendix B.1 below:

Lemma B.1. *With the setting of Theorem B.2, for a network of depth $L+1$, for any $\ell = 1, \dots, L$, we have the convergence in probability:*

$$\lim_{n_L \to \infty} \cdots \lim_{n_1 \to \infty} \sup_{t \in [0,T]} \| \frac{1}{\sqrt{n_\ell}} \left(W^{(\ell)}(t) - W^{(\ell)}(0) \right) \|_{op} = 0$$

From this lemma, to bound $\| \frac{1}{\sqrt{n_L}} W^{(L)}(t) \|_{op}$, it is hence enough to bound $\| \frac{1}{\sqrt{n_L}} W^{(L)}(0) \|_{op}$. From the law of large numbers, we obtain that the norm of each of the n_{L+1} rows of $W^{(L)}(0)$ is bounded, and hence that $\| \frac{1}{\sqrt{n_L}} W^{(L)}(0) \|_{op}$ is bounded (keep in mind that n_{L+1} is fixed, while $n_1, \dots, n_L$ grow).

From the above considerations, we can apply the induction hypothesis to the smaller network, yielding, in the limit as $n_1, \dots, n_L \to \infty$ (sequentially), that the dynamics is governed by the constant kernel $\Theta_\infty^{(L)}$:

$$\partial_t \tilde{\alpha}_i^{(L)}(t) = \frac{1}{\sqrt{n_L}} \Phi_{\Theta_\infty^{(L)}} \left(\left\langle \dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(t)\right) \left(W_i^{(L)}(t) \right)^T d_t, \cdot \right\rangle_{p^{in}} \right).$$

At the same time, the parameters of the last layer evolve according to

$$\partial_t W_{ij}^{(L)}(t) = \frac{1}{\sqrt{n_L}} \left\langle \alpha_i^{(L)}(t), d_{t,j} \right\rangle_{p^{in}}.$$

We want to give an upper bound on the variation of the weights columns $W_i^{(L)}(t)$ and of the activations $\tilde{\alpha}_i^{(L)}(t)$ during training in terms of L^2-norm and p^{in}-norm respectively. Applying the Cauchy-Schwarz inequality for each j, summing and using $\partial_t||\cdot|| \leq ||\partial_t \cdot||$), we have

$$\partial_t \left\| W_i^{(L)}(t) - W_i^{(L)}(0) \right\|_2 \leq \frac{1}{\sqrt{n_L}} ||\alpha_i^{(L)}(t)||_{p^{in}} ||d_t||_{p^{in}}.$$

Now, observing that the operator norm of $\Phi_{\Theta_\infty^{(L)}}$ is equal to $||\Theta_\infty^{(L)}||_{op}$, defined in the introduction of Appendix H.3, and using the Cauchy-Schwarz inequality, we get

$$\partial_t \left\| \tilde{\alpha}_i^{(L)}(t) - \tilde{\alpha}_i^{(L)}(0) \right\|_{p^{in}} \leq \frac{1}{\sqrt{n_L}} \left\| \Theta_\infty^{(L)} \right\|_{op} \left\| \dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(t) \right) \right\|_\infty \left\| W_i^{(L)}(t) \right\|_2 \|d_t\|_{p^{in}},$$

where the sup norm $\|\cdot\|_\infty$ is defined by $\|f\|_\infty = \sup_x |f(x)|$.

To bound both quantities simultaneously, study the derivative of the quantity

$$A(t) = ||\alpha_i^{(L)}(0)||_{p^{in}} + c \left\| \tilde{\alpha}_i^{(L)}(t) - \tilde{\alpha}_i^{(L)}(0) \right\|_{p^{in}} + ||W_i^{(L)}(0)||_2 + \left\| W_i^{(L)}(t) - W_i^{(L)}(0) \right\|_2.$$

We have

$$\begin{aligned} \partial_t A(t) &\leq \frac{1}{\sqrt{n_L}} \left(c^2 \left\| \Theta_\infty^{(L)} \right\|_{op} \left\| W_i^{(L)}(t) \right\|_2 + ||\alpha_i^{(L)}(t)||_{p^{in}} \right) ||d_t||_{p^{in}} \\ &\leq \frac{\max\{c^2 \|\Theta_\infty^{(L)}\|_{op}, 1\}}{\sqrt{n_L}} \|d_t\|_{p^{in}} A(t), \end{aligned}$$

where, in the first inequality, we have used that $|\dot{\sigma}| \leq c$ and, in the second inequality, that the sum $\|W_i^{(L)}(t)\|_2 + ||\alpha_i^{(L)}(t)||_{p^{in}}$ is bounded by $A(t)$. Applying Grönwall's Lemma, we now get

$$A(t) \leq A(0) \exp\left(\frac{\max\{c^2 \|\Theta_\infty^{(L)}\|_{op}, 1\}}{\sqrt{n_L}} \int_0^t \|d_s\|_{p^{in}} ds \right).$$

Note that $\|\Theta_\infty^{(L)}\|_{op}$ is constant during training. Clearly the value inside of the exponential converges to zero in probability as $n_L \to \infty$ given that the integral $\int_0^t \|d_t\|_{p^{in}} ds$ stays stochastically bounded. The variations of the activations $\left\| \tilde{\alpha}_i^{(L)}(t) - \tilde{\alpha}_i^{(L)}(0) \right\|_{p^{in}}$ and weights $\left\| W_i^{(L)}(t) - W_i^{(L)}(0) \right\|_2$ are bounded by $c^{-1}(A(t) - A(0))$ and $A(t) - A(0)$ respectively, which converge to zero at rate $O\left(\frac{1}{\sqrt{n_L}}\right)$.

We can now use these bounds to control the variation of the NTK and to prove the theorem. To understand how the NTK evolves, we study the evolution of the derivatives with respect to the parameters. The derivatives with respect to the bias parameters of the top layer $\partial_{b_j^{(L)}} f_{\theta,j'}$ are always equal to $\delta_{jj'}$. The derivatives with respect to the connection weights of the top layer are given by

$$\partial_{W_{ij}^{(L)}} f_{\theta,j'}(x) = \frac{1}{\sqrt{n_L}} \alpha_i^{(L)}(x;\theta) \delta_{jj'}.$$

The pre-activations $\tilde{\alpha}_i^{(L)}$ evolve at a rate of $\frac{1}{\sqrt{n_L}}$ and so do the activations $\alpha_i^{(L)}$. The summands $\partial_{W_{ij}^{(L)}} f_{\theta,j'}(x) \otimes \partial_{W_{ij}^{(L)}} f_{\theta,j''}(x')$ of the NTK hence vary at rate of $n_L^{-3/2}$ which induces a variation of the NTK of rate $\frac{1}{\sqrt{n_L}}$.

Finally let us study the derivatives with respect to the parameters of the lower layers

$$\partial_{\tilde{\theta}_k} f_{\theta,j}(x) = \frac{1}{\sqrt{n_L}} \sum_{i=1}^{n_L} \partial_{\tilde{\theta}_k} \tilde{\alpha}_i^{(L)}(x;\theta) \dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(x;\theta)\right) W_{ij}^{(L)}.$$

Their contribution to the NTK $\Theta_{jj'}^{(L+1)}(x,x')$ is

$$\frac{1}{n_L} \sum_{i,i'=1}^{n_L} \Theta_{ii'}^{(L)}(x,x') \dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(x;\theta)\right) \dot{\sigma}\left(\tilde{\alpha}_{i'}^{(L)}(x';\theta)\right) W_{ij}^{(L)} W_{i'j'}^{(L)}.$$

By the induction hypothesis, the NTK of the smaller network $\Theta^{(L)}$ tends to $\Theta_\infty^{(L)} \delta_{ii'}$ as $n_1, ..., n_{L-1} \to \infty$. The contribution therefore becomes

$$\frac{1}{n_L} \sum_{i=1}^{n_L} \Theta_\infty^{(L)}(x,x') \dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(x;\theta)\right) \dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(x';\theta)\right) W_{ij}^{(L)} W_{ij'}^{(L)}.$$

The connection weights $W_{ij}^{(L)}$ vary at rate $\frac{1}{\sqrt{n_L}}$, inducing a change of the same rate to the whole sum. We simply have to prove that the values $\dot{\sigma}(\tilde{\alpha}_i^{(L)}(x;\theta))$ also change at rate $\frac{1}{\sqrt{n_L}}$. Since the second derivative of σ is bounded, we have that

$$\partial_t \left(\dot{\sigma}\left(\tilde{\alpha}_i^{(L)}(x;\theta(t))\right)\right) = O\left(\partial_t \tilde{\alpha}_i^{(L)}(x;\theta(t))\right).$$

Since $\partial_t \tilde{\alpha}_i^{(L)}(x;\theta(t))$ goes to zero at a rate $\frac{1}{\sqrt{n_L}}$ by the bound on $A(t)$ above, this concludes the proof.

□

It is somewhat counterintuitive that the variation of the activations of the hidden layers $\alpha_i^{(\ell)}$ during training goes to zero as the width becomes large[1]. It is generally assumed that the purpose of the activations of the hidden layers is to learn "good" representations of the data during training. However note that even though the variation of each individual activation shrinks, the number of neurons grows, resulting in a significant collective effect. This explains why the training of the parameters of each layer ℓ has an influence on the network function f_θ even though it has asymptotically no influence on the individual activations of the layers ℓ' for $\ell < \ell' < L$.

A Priori Control during Training

The goal of this section is to prove Lemma B.2, which is a key ingredient in the proof of Theorem B.2. Let us first recall it:

Lemma B.2. *With the setting of Theorem B.2, for a network of depth $L+1$, for any $\ell = 1, \ldots, L$, we have the convergence in probability:*

$$\lim_{n_L \to \infty} \cdots \lim_{n_1 \to \infty} \sup_{t \in [0,T]} \| \frac{1}{\sqrt{n_\ell}} \left(W^{(\ell)}(t) - W^{(\ell)}(0)\right) \|_{op} = 0$$

[1] As a consequence, the pre-activations stay Gaussian during training as well, with the same covariance $\Sigma^{(\ell)}$.

Proof. We prove the lemma for all $\ell = 1, \dots, L$ simultaneously, by expressing the variation of the weights $\frac{1}{\sqrt{n_\ell}} W^{(\ell)}$ and activations $\frac{1}{\sqrt{n_\ell}} \tilde{\alpha}^{(\ell)}$ in terms of 'back-propagated' training directions $d^{(1)}, \dots, d^{(L)}$ associated with the lower layers and the NTKs of the corresponding subnetworks:

1. At all times, the evolution of the preactivations and weights is given by:

$$\partial_t \tilde{\alpha}^{(\ell)} = \Phi_{\Theta^{(\ell)}} \left(< d_t^{(\ell)}, \cdot >_{p^{in}} \right)$$
$$\partial_t W^{(\ell)} = \frac{1}{\sqrt{n_\ell}} < \alpha^{(\ell)}, d_t^{(\ell+1)} >_{p^{in}},$$

where the layer-wise training directions $d^{(1)}, \dots, d^{(L)}$ are defined recursively by

$$d_t^{(\ell)} = \begin{cases} d_t & \text{if } \ell = L+1 \\ \dot{\sigma}\left(\tilde{\alpha}^{(\ell)}\right) \left(\frac{1}{\sqrt{n_\ell}} W^{(\ell)}\right)^T d_t^{(\ell+1)} & \text{if } \ell \leq L, \end{cases}$$

and where the sub-network NTKs $\Theta^{(\ell)}$ satisfy

$$\Theta^{(1)} = \left[\left[\frac{1}{\sqrt{n_0}} \alpha^{(0)} \right]^T \left[\frac{1}{\sqrt{n_0}} \alpha^{(0)} \right] \right] \otimes Id_{n_\ell} + \beta^2 \otimes Id_{n_\ell}$$
$$\Theta^{(\ell+1)} = \frac{1}{\sqrt{n_\ell}} W^{(\ell)} \dot{\sigma}(\tilde{\alpha}^{(\ell)}) \Theta^{(\ell)} \dot{\sigma}(\tilde{\alpha}^{(\ell)}) \frac{1}{\sqrt{n_\ell}} W^{(\ell)}$$
$$+ \left[\left[\frac{1}{\sqrt{n_\ell}} \alpha^{(\ell)} \right]^T \left[\frac{1}{\sqrt{n_\ell}} \alpha^{(\ell)} \right] \right] \otimes Id_{n_\ell} + \beta^2 \otimes Id_{n_\ell}.$$

2. Set $w^{(k)}(t) := \left\| \frac{1}{\sqrt{n_k}} W^{(k)}(t) \right\|_{op}$ and $a^{(k)}(t) := \left\| \frac{1}{\sqrt{n_k}} \alpha^{(k)}(t) \right\|_{p^{in}}$. The identities of the previous step yield the following recursive bounds:

$$\left\| d_t^{(\ell)} \right\|_{p^{in}} \leq c w^{(\ell)}(t) \left\| d_t^{(\ell+1)} \right\|_{p^{in}},$$

where c is the Lipschitz constant of σ. These bounds lead to

$$\left\| d_t^{(\ell)} \right\|_{p^{in}} \leq c^{L+1-\ell} \prod_{k=\ell}^{L} w^{(k)}(t) \left\| d_t \right\|_{p^{in}}.$$

For the subnetworks NTKs we have the recursive bounds

$$\|\Theta^{(1)}\|_{op} \leq (a^{(0)}(t))^2 + \beta^2.$$
$$\|\Theta^{(\ell+1)}\|_{op} \leq c^2 (w^{(\ell)}(t))^2 \|\Theta^{(\ell)}\|_{op} + (a^{(\ell)}(t))^2 + \beta^2,$$

which lead to

$$\|\Theta^{(\ell+1)}\|_{op} \leq \mathcal{P}\left(a^{(1)}, \dots, a^{(\ell)}, w^{(1)}, \dots, w^{(\ell)}\right),$$

where $\mathcal{P}$ is a polynomial which only depends on ℓ, c, β and p^{in}.

3. Set

$$\tilde{a}^{(k)}(t) := \left\| \frac{1}{\sqrt{n_k}} \left(\tilde{\alpha}^{(k)}(t) - \tilde{\alpha}^{(k)}(0) \right) \right\|_{p^{in}}$$
$$\tilde{w}^{(k)}(t) := \left\| \frac{1}{\sqrt{n_k}} \left(W^{(k)}(t) - W^{(k)}(0) \right) \right\|_{op}$$

and define

$$A(t) = \sum_{k=1}^{L} a^{(k)}(0) + c\tilde{a}^{(k)}(t) + w^{(k)}(0) + \tilde{w}^{(k)}(t).$$

Since $a^{(k)}(t) \leq a^{(k)}(0) + c\tilde{a}^{(k)}(t)$ and $w^{(k)}(t) \leq w^{(k)}(0) + \tilde{w}^{(k)}(t)$, controlling $A(t)$ will enable us to control the $a^{(k)}(t)$ and $w^{(k)}(t)$. Using the formula at the beginning of the first step, we obtain

$$\partial_t \tilde{a}^{(\ell)}(t) \leq \frac{1}{\sqrt{n_\ell}} \|\Theta^{(\ell)}(t)\|_{op} \|d_t^{(\ell)}\|_{p^{in}}$$
$$\partial_t \tilde{w}^{(\ell)}(t) \leq \frac{1}{\sqrt{n_\ell}} a^{(\ell)}(t) \|d_t^{(\ell+1)}\|_{p^{in}}.$$

This allows one to bound the derivative of $A(t)$ as follows:

$$\partial_t A(t) \leq \sum_{\ell=1}^{L} \frac{c}{\sqrt{n_\ell}} \|\Theta^{(\ell)}(t)\|_{op} \|d_t^{(\ell)}\|_{p^{in}} + \frac{1}{\sqrt{n_\ell}} a^{(\ell)}(t) \|d_t^{(\ell+1)}\|_{p^{in}}.$$

Using the polynomial bounds on $\|\Theta^{(\ell)}(t)\|_{op}$ and $\|d_t^{(\ell+1)}\|_{p^{in}}$ in terms of the $a^{(k)}$ and $w^{(k)}$ for $k = 1, \dots \ell$ obtained in the previous step, we get that

$$\partial_t A(t) \leq \frac{1}{\sqrt{\min\{n_1, \dots, n_L\}}} \mathcal{Q}\left(w^{(1)}(t), \dots, w^{(L)}(t), a^{(1)}(t), \dots, a^{(L)}(t) \right) \|d_t\|_{p^{in}},$$

where the polynomial Q only depends on L, c, β and p^{in} and has positive coefficients. As a result, we can use $a^{(k)}(t) \leq a^{(k)}(0) + c\tilde{a}^{(k)}(t)$ and $w^{(k)}(t) \leq w^{(k)}(0) + \tilde{w}^{(k)}(t)$ to get the polynomial bound

$$\partial_t A(t) \leq \frac{1}{\sqrt{\min\{n_1, \dots, n_L\}}} \tilde{\mathcal{Q}}(A(t)) \|d_t\|_{p^{in}}.$$

4. Let us now observe that $A(0)$ is stochastically bounded as we take the sequential limit $\lim_{n_L \to \infty} \cdots \lim_{n_1 \to \infty}$ as in the statement of the lemma. In this limit, we indeed have that $w^{(\ell)}$and $a^{(\ell)}$ are convergent: we have $w^{(\ell)} \to 0$, while $a^{(\ell)}$ converges by Proposition B.1.

   The polynomial control we obtained on the derivative of $A(t)$ now allows one to use (a nonlinear form of, see e.g. [49]) GrÃ¶nwall's Lemma: we obtain that $A(t)$ stays uniformly bounded on $[0, \tau]$ for some $\tau = \tau(n_1, \dots, n_L) > 0$, and that $\tau \to T$ as $\min(n_1, \dots, n_L) \to \infty$, owing to the $\frac{1}{\sqrt{\min\{1, \dots, n_L\}}}$in front of the polynomial. Since $A(t)$ is bounded, the differential bound on $A(t)$ gives that the derivative $\partial_t A(t)$ converges uniformly to 0 on $[0, \tau]$ for any $\tau < T$, and hence $A(t) \to A(0)$. This concludes the proof of the lemma.

□

Positive-Definiteness of $\Theta_\infty^{(L)}$

This subsection is devoted to the proof of Proposition B.2, which we now recall:

Proposition B.2. *For a non-polynomial Lipschitz nonlinearity σ, for any input dimension n_0, the restriction of the limiting NTK $\Theta_\infty^{(L)}$ to the unit sphere $\mathbb{S}^{n_0-1} = \{x \in \mathbb{R}^{n_0} : x^T x = 1\}$ is positive-definite if $L \geq 2$.*

A key ingredient for the proof of Proposition B.2 is the following Lemma, which comes from [42].

Lemma B.3 (Lemma 12(a) in suppl. mat. of [42])**.** *Let $\hat{\mu} : [-1,1] \to \mathbb{R}$ denote the dual of a Lipschitz function $\mu : \mathbb{R} \to \mathbb{R}$, defined by $\hat{\mu}(\rho) = \mathbb{E}_{(X,Y)}[\mu(X)\mu(Y)]$ where (X,Y) is a centered Gaussian vector of covariance Σ, with*

$$\Sigma = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

If the expansion of μ in Hermite polynomials $(h_i)_{i\geq 0}$ is given by $\mu = \sum_{i=0}^{\infty} a_i h_i$, we have

$$\hat{\mu}(\rho) = \sum_{i=0}^{\infty} a_i^2 \rho^i.$$

The other key ingredient for proving Proposition B.2 is the following theorem, which is a slight reformulation of Theorem 1(b) in [77], which itself is a generalization of a classical result of Schönberg:

Theorem B.3. *For a function $f : [-1,1] \to \mathbb{R}$ with $f(\rho) = \sum_{n=0}^{\infty} b_n \rho^n$, the kernel $K_f^{(n_0)} : \mathbb{S}^{n_0-1} \times \mathbb{S}^{n_0-1} \to \mathbb{R}$ defined by*

$$K_f^{(n_0)}(x, x') = f\left(x^T x'\right)$$

is positive-definite for any $n_0 \geq 1$ if and only if the coefficients b_n are strictly positive for infinitely many even and infinitely many odd integers n.

With Lemma B.3 and Theorem B.3 above, we are now ready to prove Proposition B.2.

Proof of Proposition B.2. We first decompose the limiting NTK $\Theta^{(L)}$ recursively, relate its positive-definiteness to that of the activation kernels, then show that the positive-definiteness of the activation kernels at level 2 implies that of the higher levels, and finally show the positive-definiteness at level 2 using Lemma B.3 and Theorem B.3:

1. Observe that for any $L \geq 1$, using the notation of Theorem 2.1, we have

$$\Theta^{(L+1)} = \dot{\Sigma}^{(L)} \Theta^{(L)} + \Sigma^{(L+1)}.$$

   Note that the kernel $\dot{\Sigma}^{(L)}\Theta^{(L)}$ is positive semi-definite, being the product of two positive semi-definite kernels. Hence, if we show that $\Sigma^{(L+1)}$ is positive-definite, this implies that $\Theta^{(L+1)}$ is positive-definite.

2. By definition, with the notation of Proposition B.1 we have

$$\Sigma^{(L+1)}(x,x') = \mathbb{E}_{f\sim\mathcal{N}\left(0,\Sigma^{(L)}\right)}\left[\sigma\left(f\left(x\right)\right)\sigma\left(f\left(x'\right)\right)\right] + \beta^2.$$

This gives, for any collection of coefficients $c_1,\ldots,c_d \in \mathbb{R}$ and any pairwise distinct $x_1,\ldots,x_d \in \mathbb{R}^{n_0}$, that

$$\sum_{i,j=1}^{d} c_i c_j \Sigma^{(L+1)}\left(x_i,x_j\right) = \mathbb{E}\left[\left(\sum_i c_i \sigma\left(f\left(x_i\right)\right)\right)^2\right] + \left(\beta\sum_i c_i\right)^2.$$

Hence the left-hand side only vanishes if $\sum c_i\sigma\left(f\left(x_i\right)\right)$ is almost surely zero. If $\Sigma^{(L)}$ is positive-definite, the Gaussian $\left(f\left(x_i\right)\right)_{i=1,\ldots d}$ is non-degenerate, so this only occurs when $c_1 = \cdots = c_d = 0$ since σ is assumed to be non-constant. This shows that the positive-definiteness of $\Sigma^{(L+1)}$ is implied by that of $\Sigma^{(L)}$. By induction, if $\Sigma^{(2)}$ is positive-definite, we obtain that all $\Sigma^{(L)}$ with $L \geq 2$ are positive-definite as well. By the first step this hence implies that $\Theta^{(L)}$ is positive-definite as well.

3. By the previous steps, to prove the proposition, it suffices to show the positive-definitess of $\Sigma^{(2)}$ on the unit sphere $\mathbb{S}^{n_0-1}$. We have

$$\Sigma^{(2)}(x,x') = \mathbb{E}_{(X,Y)\sim\mathcal{N}\left(0,\tilde{\Sigma}\right)}\left[\sigma\left(X\right)\sigma\left(Y\right)\right] + \beta^2$$

where

$$\tilde{\Sigma} = \begin{pmatrix} \frac{1}{n_0}+\beta^2 & \frac{1}{n_0}x^Tx'+\beta^2 \\ \frac{1}{n_0}x^Tx+\beta^2 & \frac{1}{n_0}+\beta^2 \end{pmatrix}.$$

A change of variables then yields

$$\mathbb{E}_{(X,Y)\sim\mathcal{N}\left(0,\tilde{\Sigma}\right)}\left[\sigma\left(X\right)\sigma\left(Y\right)\right] + \beta^2 = \hat{\mu}\left(\frac{n_0\beta^2 + x^Tx'}{n_0\beta^2+1}\right) + \beta^2, \tag{B.1.1}$$

where $\hat{\mu}:[-1,1]\to\mathbb{R}$ is the dual in the sense of Lemma B.3 of the function $\mu:\mathbb{R}\to\mathbb{R}$ defined by $\mu\left(x\right) = \sigma\left(x\sqrt{\frac{1}{n_0}+\beta^2}\right)$.

4. Writing the expansion of μ in Hermite polynomials $\left(h_i\right)_{i\geq 0}$

$$\mu = \sum_{i=0}^{\infty} a_i h_i,$$

we obtain that $\hat{\mu}$ is given by the power series

$$\hat{\mu}\left(\rho\right) = \sum_{i=0}^{\infty} a_i^2 \rho^i,$$

Since σ is non-polynomial, so is μ, and as a result, there is an infinite number of nonzero a_i's in the above sum.

5. Using (B.1.1) above, we obtain that

$$\Sigma^{(2)}(x, x') = \nu\left(x^T x'\right),$$

where $\nu : \mathbb{R} \to \mathbb{R}$ is defined by

$$\nu(\rho) = \beta^2 + \sum_{i=0}^{\infty} a_i \left(\frac{n_0\beta^2 + \rho}{n_0\beta^2 + 1}\right)^i,$$

where the a_i's are the coefficients of the Hermite expansion of μ. Now, observe that by the previous step, the power series expansion of ν contains both an infinite number of nonzero even terms and an infinite number of nonzero odd terms. This enables one to apply Theorem B.3 to obtain that $\Sigma^{(2)}$ is indeed positive-definite, thereby concluding the proof.

□

Remark B.1. Using similar techniques to the one applied in the proof above, one can show a converse to Proposition B.2: if the nonlinearity σ is a polynomial, the corresponding NTK $\Theta^{(2)}$ is not positive-definite $\mathbb{S}^{n_0-1}$ for certain input dimensions n_0.

Appendix C

The Asymptotic Spectrum of the Hessian of DNN Throughout Training

C.1 Proofs

For the proofs of the theorems and propositions presented in the main text, we reformulate the setup of [105]. For a fixed training set $x_1, ..., x_N$, we consider a (possibly random) time-varying training direction $D(t) \in \mathbb{R}^{Nn_L}$ which describes how each of the outputs must be modified. In the case of gradient descent on a cost $C(Y)$, the training direction is $D(t) = \nabla C(Y(t))$. The parameters are updated according to the differential equation

$$\partial_t \theta(t) = \left(\partial_\theta Y(t)\right)^T D(t).$$

Under the condition that $\int_0^T \|D(t)\|_2 \, dt$ is stochastically bounded as the width of the network goes to infinity, the NTK $\Theta^{(L)}$ converges to its fixed limit uniformly over $[0, T]$.

The reason we consider a general training direction (and not only a gradient of a loss) is that we can split a network in two at a layer ℓ and the training of the smaller network will be according to the training direction $D_i^{(\ell)}(t)$ given by

$$D_i^{(\ell)}(t) = diag\left(\dot{\sigma}\left(\alpha^{(\ell)}(x_i)\right)\right)\left(\frac{1}{\sqrt{n_\ell}} W^{(\ell)}\right)^T ... diag\left(\dot{\sigma}\left(\alpha^{(L-1)}(x_i)\right)\right)\left(\frac{1}{\sqrt{n_{L-1}}} W^{(L-1)}\right)^T D_i(t)$$

because the derivatives $\dot{\sigma}$ are bounded and by Lemma 1 of the Appendix of [105], this training direction satisfies the constraints even though it is not the gradient of a loss. As a consequence, as $n_1 \to \infty, ..., n_{\ell-1} \to \infty$ the NTK of the smaller network $\Theta^{(\ell)}$ also converges to its limit uniformly over $[0, T]$. As we let $n_\ell \to \infty$ the pre-activations $\tilde{\alpha}_i^{(\ell)}$ and weights $W_{ij}^{(\ell)}$ move at a rate of $1/\sqrt{n_\ell}$. We will use this rate of change to prove that other types of kernels are constant during training.

When a network is trained with gradient descent on a loss C with BGOSS, the integral $\int_0^T \|D(t)\|_2 \, dt$ is stochastically bounded. Because the loss is decreasing during training, the outputs $Y(t)$ lie in the sublevel set $U_{C(Y(0))}$ for all times t. The norm of the gradient is hence bounded for all times t. Because the distribution of $Y(0)$ converges to a multivariate Gaussian, $b(C(Y(0)))$ is stochastically bounded as the width grows, where $b(a)$ is a bound on the norm of the gradient on U_a. We then have the bound $\int_0^T \|D(t)\|_2 \, dt \leq T b(C(Y(0)))$ which is itself stochastically bounded.

For the binary and softmax cross-entropy losses the gradient is uniformly bounded:

Proposition C.1. *For the binary cross-entropy loss C and any $Y \in \mathbb{R}^N$, $\|\nabla C(Y)\|_2 \leq \frac{1}{\sqrt{N}}$.*

For the softmax cross-entropy loss C on $c \in \mathbb{N}$ classes and any $Y \in \mathbb{R}^{Nc}$, $\|\nabla C(Y)\|_2 \leq \frac{\sqrt{2c}}{\sqrt{N}}$.

Proof. The binary cross-entropy loss with labels $Y^* \in \{0,1\}^N$ is

$$C(Y) = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{Y_i Y_i^*}}{1+e^{Y_i}} = \frac{1}{N}\sum_{i=1}^{N} \log\left(1+e^{Y_i}\right) - Y_i Y_i^*$$

and the gradient at an input i is

$$\partial_i C(Y) = \frac{1}{N}\frac{e^{Y_i} - Y_i^*(1+e^{Y_i})}{1+e^{Y_i}}$$

which is bounded in absolute value by $\frac{1}{N}$ for both $Y_i^* = 0, 1$ such that $\|\nabla C(Y)\|_2 \leq \frac{1}{\sqrt{N}}$.

The softmax cross-entropy loss over c classes with labels $Y^* \in \{1, \ldots, c\}^N$ is defined by

$$C(Y) = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{Y_{iY_i^*}}}{\sum_{k=1}^{c} e^{Y_{ik}}} = \frac{1}{N}\sum_{i=1}^{N} \log\left(\sum_{k=1}^{c} e^{Y_{ik}}\right) - Y_{iY_i^*}.$$

The gradient is at an input i and output class m is

$$\partial_{im} C(Y) = \frac{1}{N}\left(\frac{e^{Y_{im}}}{\sum_{k=1}^{c} e^{Y_{ik}}} - \delta_{Y_i^* m}\right)$$

which is bounded in absolute value by $\frac{2}{N}$ such that $\|\nabla C(Y)\|_2 \leq \frac{\sqrt{2c}}{\sqrt{N}}$. □

C.2 Preliminaries

To study the moments of the matrix S, we first have to show that two tensors vanish as $n_1, ..., n_{L-1} \to \infty$:

$$\Omega^{(L)}_{k_0,k_1,k_2}(x_0,x_1,x_2) = \left(\nabla f_{\theta,k_0}(x_0)\right)^T \mathcal{H} f_{\theta,k_1}(x_1) \nabla f_{\theta,k_2}(x_2)$$
$$\Gamma^{(L)}_{k_0,k_1,k_2,k_3}(x_0,x_1,x_2,x_4) = \left(\nabla f_{\theta,k_0}(x_0)\right)^T \mathcal{H} f_{\theta,k_1}(x_1) \mathcal{H} f_{\theta,k_2}(x_2) \nabla f_{\theta,k_3}(x_3).$$

We study these tensors recursively, for this, we need a recursive definition for the first derivatives $\partial_{\theta_p} f_{\theta,k}(x)$ and second derivatives $\partial^2_{\theta_p \theta_{p'}} f_{\theta,k}(x)$. The value of these derivatives depend on the layer ℓ the parameters θ_p and $\theta_{p'}$ belong to, and on whether they are connection weights $W^{(\ell)}_{mk}$ or biases $b^{(\ell)}_k$. The derivatives with respect to the parameters of the last layer are

$$\partial_{W^{(L-1)}_{mk}} f_{\theta,k'}(x) = \frac{1}{\sqrt{n_{L-1}}} \alpha^{(L-1)}_m(x) \delta_{kk'}$$
$$\partial_{b^{(L-1)}_k} f_{\theta,k'}(x) = \beta^2 \delta_{kk'}$$

for parameters θ_p which belong to the lower layers the derivatives can be defined recursively by

$$\partial_{\theta_p} f_{\theta,k}(x) = \frac{1}{\sqrt{n_{L-1}}} \sum_{m=1}^{n_{L-1}} \partial_{\theta_p} \tilde{\alpha}^{(L-1)}_m(x) \dot{\sigma}\left(\tilde{\alpha}^{(L-1)}_m(x)\right) W^{(L-1)}_{mk}.$$

For the second derivatives, we first note that if either of the parameters θ_p or $\theta_{p'}$ are bias of the last layer, or if they are both connection weights of the last layer, then $\partial^2_{\theta_p\theta_{p'}} f_{\theta,k}(x) = 0$. Two cases are left: when one parameter is a connection weight of the last layer and the others belong to the lower layers, and when both belong to the lower layers. Both cases can be defined recursively in terms of the first and second derivatives of $\tilde{\alpha}_m^{(L-1)}$:

$$\begin{aligned}
\partial^2_{\theta_p W^{(L)}_{mk}} f_{\theta,k'}(x) &= \frac{1}{\sqrt{n_{L-1}}} \partial_{\theta_p} \tilde{\alpha}_m^{(L-1)}(x) \dot{\sigma}\left(\tilde{\alpha}_m^{(L-1)}(x)\right) \delta_{kk'} \\
\partial^2_{\theta_p\theta_{p'}} f_{\theta,k'}(x) &= \frac{1}{\sqrt{n_{L-1}}} \sum_{m=1}^{n_{L-1}} \partial^2_{\theta_p\theta_{p'}} \tilde{\alpha}_m^{(L-1)}(x) \dot{\sigma}\left(\tilde{\alpha}_m^{(L-1)}(x)\right) W^{(L-1)}_{mk} \\
&\quad + \frac{1}{\sqrt{n_{L-1}}} \sum_{m=1}^{n_{L-1}} \partial_{\theta_p} \tilde{\alpha}_m^{(L-1)}(x) \partial_{\theta_{p'}} \tilde{\alpha}_m^{(L-1)}(x) \ddot{\sigma}\left(\tilde{\alpha}_m^{(L-1)}(x)\right) W^{(L-1)}_{mk}.
\end{aligned}$$

Using these recursive definitions, the tensors $\Omega^{(L+1)}$ and $\Gamma^{(L+1)}$ are given in terms of $\Theta^{(L)}$,$\Omega^{(L)}$ and $\Gamma^{(L)}$, in the same manner that the NTK $\Theta^{(L+1)}$ is defined recursively in terms of $\Theta^{(L)}$ in [105].

Lemma C.1. *For any loss C with BGOSS and $\sigma \in C_b^4(\mathbb{R})$, we have uniformly over $[0,T]$*

$$\lim_{n_{L-1}\to\infty} \cdots \lim_{n_1\to\infty} \Omega^{(L)}_{k_0,k_1,k_2}(x_0,x_1,x_2) = 0$$

Proof. The proof is done by induction. When $L = 1$ the second derivatives $\partial^2_{\theta_p\theta_{p'}} f_{\theta,k}(x) = 0$ and $\Omega^{(L)}_{k_0,k_1,k_2}(x_0,x_1,x_2) = 0$.

For the induction step, we write $\Omega^{(\ell+1)}_{k_0,k_1,k_2}(x_0,x_1,x_2)$ recursively as

$$\begin{aligned}
&n_\ell^{-3/2} \sum_{m_0,m_1,m_2} \Theta^{(\ell)}_{m_0,m_1}(x_0,x_1)\Theta^{(\ell)}_{m_1,m_2}(x_1,x_2)\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_0}(x_0))\ddot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_1}(x_1))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_2}(x_2))W^{(\ell)}_{m_0k_0}W^{(\ell)}_{m_1k_1}W^{(\ell)}_{m_2k_2} \\
&+ n_\ell^{-3/2} \sum_{m_0,m_1,m_2} \Omega^{(\ell)}_{m_0,m_1,m_2}(x_0,x_1,x_2)\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_0}(x_0))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_1}(x_1))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_2}(x_2))W^{(\ell)}_{m_0k_0}W^{(\ell)}_{m_1k_1}W^{(\ell)}_{m_2k_2} \\
&+ n_\ell^{-3/2} \sum_{m_0,m_1} \Theta^{(\ell)}_{m_0,m_1}(x_0,x_1)\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_0}(x_0))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_1}(x_1))\sigma(\tilde{\alpha}^{(\ell)}_{m_1}(x_2))W^{(\ell)}_{m_0k_0}\delta_{k_1k_2} \\
&+ n_\ell^{-3/2} \sum_{m_1,m_2} \Theta^{(\ell)}_{m_1,m_2}(x_1,x_2)\sigma(\tilde{\alpha}^{(\ell)}_{m_1}(x_0))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_1}(x_1))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m_2}(x_2))\delta_{k_0k_1}W^{(\ell)}_{m_2k_2}.
\end{aligned}$$

As $n_1, ..., n_{\ell-1} \to \infty$ and for any times $t < T$, the NTK $\Theta^{(\ell)}$ converges to its limit while $\Omega^{(\ell)}$ vanishes. The second summand hence vanishes and the others converge to

$$\begin{aligned}
&n_\ell^{-3/2} \sum_{m} \Theta^{(\ell)}_{\infty}(x_0,x_1)\Theta^{(\ell)}_{\infty}(x_1,x_2)\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m}(x_0))\ddot{\sigma}(\tilde{\alpha}^{(\ell)}_{m}(x_1))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m}(x_2))W^{(\ell)}_{mk_0}W^{(\ell)}_{mk_1}W^{(\ell)}_{mk_2} \\
&+ n_\ell^{-3/2} \sum_{m} \Theta^{(\ell)}_{\infty}(x_0,x_1)\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m}(x_0))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m}(x_1))\sigma(\tilde{\alpha}^{(\ell)}_{m}(x_2))W^{(\ell)}_{mk_0}\delta_{k_1k_2} \\
&+ n_\ell^{-3/2} \sum_{m} \Theta^{(\ell)}_{\infty}(x_1,x_2)\sigma(\tilde{\alpha}^{(\ell)}_{m}(x_0))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m}(x_1))\dot{\sigma}(\tilde{\alpha}^{(\ell)}_{m}(x_2))\delta_{k_0k_1}W^{(\ell)}_{mk_2}.
\end{aligned}$$

At initialization, all terms vanish as $n_\ell \to \infty$ because all summands are independent with zero mean and finite variance: in the $n_1 \to \infty, \dots, n_{\ell-1} \to \infty$ limit, the $\tilde{\alpha}_m^{(\ell)}(x)$ are independent for different m, see [105]. During training, the weights $W^{(\ell)}$ and preactivations $\tilde{\alpha}^{(\ell)}$ move at a rate of $1/\sqrt{n_\ell}$ (see the proof of convergence of the NTK in [105]). Since $\dot{\sigma}$ is Lipschitz, we obtain that the motion during training of each of the sums is of order $n_\ell^{-3/2+1/2} = n_\ell^{-1}$. As a result, uniformly over times $t \in [0, T]$, all the sums vanish. □

Similarily, we have

Lemma C.2. *For any loss C with BGOSS and $\sigma \in C_b^4(\mathbb{R})$, we have uniformly over $[0, T]$*

$$\lim_{n_{L-1}\to\infty} \cdots \lim_{n_1\to\infty} \Gamma^{(L)}_{k_0,k_1,k_2,k_3}(x_0, x_1, x_2, x_3) = 0$$

Proof. The proof is done by induction. When $L = 1$ the hessian $\mathcal{H}F^{(1)} = 0$, such that $\Gamma^{(L)}_{k_0,k_1,k_2,k_3}(x_0, x_1, x_2, x_3) = 0$.

For the induction step, $\Gamma^{(\ell+1)}$ can be defined recursively:

$$\begin{aligned}
&\Gamma^{(L+1)}_{k_0,k_1,k_2,k_3}(x_0, x_1, x_2, x_3) \\
&= n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \Gamma^{(L)}_{m_0,m_1,m_2,m_3}(x_0, x_1, x_2, x_3)\dot{\sigma}(\alpha^{(L)}_{m_0}(x_0))\dot{\sigma}(\alpha^{(L)}_{m_1}(x_1))\dot{\sigma}(\alpha^{(L)}_{m_2}(x_2))\dot{\sigma}(\alpha^{(L)}_{m_3}(x_3)) \\
&\qquad W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3} \\
&+ n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \Theta^{(L)}_{m_0,m_1}(x_0, x_1)\Omega^{(L)}_{m_1,m_2,m_3}(x_1, x_2, x_3)\dot{\sigma}(\alpha^{(L)}_{m_0}(x_0))\ddot{\sigma}(\alpha^{(L)}_{m_1}(x_1)) \\
&\qquad \dot{\sigma}(\alpha^{(L)}_{m_2}(x_2))\dot{\sigma}(\alpha^{(L)}_{m_3}(x_3)) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3} \\
&+ n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \Omega^{(L)}_{m_0,m_1,m_2}(x_0, x_1, x_2)\Theta^{(L)}_{m_2,m_3}(x_2, x_3)\dot{\sigma}(\alpha^{(L)}_{m_0}(x_0))\dot{\sigma}(\alpha^{(L)}_{m_1}(x_1)) \\
&\qquad \ddot{\sigma}(\alpha^{(L)}_{m_2}(x_2))\dot{\sigma}(\alpha^{(L)}_{m_3}(x_3)) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3} \\
&+ n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \Theta^{(L)}_{m_0,m_1}(x_0, x_1)\Theta^{(L)}_{m_1,m_2}(x_1, x_2)\Theta^{(L)}_{m_2,m_3}(x_2, x_3)\dot{\sigma}(\alpha^{(L)}_{m_0}(x_0))\ddot{\sigma}(\alpha^{(L)}_{m_1}(x_1)) \\
&\qquad \ddot{\sigma}(\alpha^{(L)}_{m_2}(x_2))\dot{\sigma}(\alpha^{(L)}_{m_3}(x_3)) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3} \\
&+ n_L^{-2} \sum_{m_1,m_2,m_3} \Omega^{(L)}_{m_1,m_2,m_3}(x_1, x_2, x_3)\sigma(\alpha^{(L)}_{m_1}(x_0))\dot{\sigma}(\alpha^{(L)}_{m_1}(x_1))\dot{\sigma}(\alpha^{(L)}_{m_2}(x_2))\dot{\sigma}(\alpha^{(L)}_{m_3}(x_3)) \\
&\qquad \delta_{k_0k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3} \\
&+ n_L^{-2} \sum_{m_1,m_2,m_3} \Theta^{(L)}_{m_1,m_2}(x_1, x_2)\Theta^{(L)}_{m_2,m_3}(x_2, x_3)\sigma(\alpha^{(L)}_{m_1}(x_0))\dot{\sigma}(\alpha^{(L)}_{m_1}(x_1))\ddot{\sigma}(\alpha^{(L)}_{m_2}(x_2))\dot{\sigma}(\alpha^{(L)}_{m_3}(x_3)) \\
&\qquad \delta_{k_0k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3} \\
&+ n_L^{-2} \sum_{m_0,m_1,m_2} \Omega^{(L)}_{m_0,m_1,m_2}(x_0, x_1, x_2)\dot{\sigma}(\alpha^{(L)}_{m_0}(x_0))\dot{\sigma}(\alpha^{(L)}_{m_1}(x_1))\dot{\sigma}(\alpha^{(L)}_{m_2}(x_2))\sigma(\alpha^{(L)}_{m_2}(x_3)) \\
&\qquad W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} \delta_{k_2k_3}
\end{aligned}$$

$$+n_L^{-2}\sum_{m_0,m_1,m_2}\Theta^{(L)}_{m_0,m_1}(x_0,x_1)\Theta^{(L)}_{m_1,m_2}(x_1,x_2)\dot\sigma(\alpha^{(L)}_{m_0}(x_0))\ddot\sigma(\alpha^{(L)}_{m_1}(x_1))\dot\sigma(\alpha^{(L)}_{m_2}(x_2))\sigma(\alpha^{(L)}_{m_2}(x_3))$$
$$W^{(L)}_{m_0k_0}W^{(L)}_{m_1k_1}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_{m_1,m_2}\Theta^{(L)}_{m_1,m_2}(x_1,x_2)\sigma(\alpha^{(L)}_{m_1}(x_0))\dot\sigma(\alpha^{(L)}_{m_1}(x_1))\dot\sigma(\alpha^{(L)}_{m_2}(x_2))\sigma(\alpha^{(L)}_{m_2}(x_3))\delta_{k_0k_1}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_{m_0,m_1,m_3}\Theta^{(L)}_{m_0,m_1}(x_0,x_1)\Theta^{(L)}_{m_1,m_3}(x_2,x_3)\dot\sigma(\alpha^{(L)}_{m_0}(x_0))\dot\sigma(\alpha^{(L)}_{m_1}(x_1))\dot\sigma(\alpha^{(L)}_{m_1}(x_2))\dot\sigma(\alpha^{(L)}_{m_3}(x_3))$$
$$W^{(L)}_{m_0k_0}\delta_{k_1k_2}W^{(L)}_{m_3k_3}$$

As $n_1,...,n_{\ell-1}\to\infty$ and for any times $t<T$, the NTK $\Theta^{(\ell)}$ converges to its limit while $\Omega^{(\ell)}$ and $\Gamma^{(\ell)}$ vanishes. $\Gamma^{(L+1)}_{k_0,k_1,k_2,k_3}(x_0,x_1,x_2,x_3)$ therefore converges to:

$$+n_L^{-2}\sum_m\Theta^{(L)}_\infty(x_0,x_1)\Theta^{(L)}_\infty(x_1,x_2)\Theta^{(L)}_\infty(x_2,x_3)\dot\sigma(\alpha^{(L)}_m(x_0))\ddot\sigma(\alpha^{(L)}_m(x_1))\ddot\sigma(\alpha^{(L)}_m(x_2))\dot\sigma(\alpha^{(L)}_m(x_3))$$
$$W^{(L)}_{mk_0}W^{(L)}_{mk_1}W^{(L)}_{mk_2}W^{(L)}_{mk_3}$$
$$+n_L^{-2}\sum_m\Theta^{(L)}_\infty(x_1,x_2)\Theta^{(L)}_\infty(x_2,x_3)\sigma(\alpha^{(L)}_m(x_0))\dot\sigma(\alpha^{(L)}_m(x_1))\ddot\sigma(\alpha^{(L)}_m(x_2))\dot\sigma(\alpha^{(L)}_m(x_3))$$
$$\delta_{k_0k_1}W^{(L)}_{mk_2}W^{(L)}_{mk_3}$$
$$+n_L^{-2}\sum_m\Theta^{(L)}_\infty(x_0,x_1)\Theta^{(L)}_\infty(x_1,x_2)\dot\sigma(\alpha^{(L)}_m(x_0))\ddot\sigma(\alpha^{(L)}_m(x_1))\dot\sigma(\alpha^{(L)}_m(x_2))\sigma(\alpha^{(L)}_m(x_3))$$
$$W^{(L)}_{mk_0}W^{(L)}_{mk_1}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_m\Theta^{(L)}_\infty(x_1,x_2)\sigma(\alpha^{(L)}_m(x_0))\dot\sigma(\alpha^{(L)}_m(x_1))\dot\sigma(\alpha^{(L)}_m(x_2))\sigma(\alpha^{(L)}_m(x_3))\delta_{k_0k_1}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_m\Theta^{(L)}_\infty(x_0,x_1)\Theta^{(L)}_\infty(x_2,x_3)\dot\sigma(\alpha^{(L)}_m(x_0))\dot\sigma(\alpha^{(L)}_m(x_1))\dot\sigma(\alpha^{(L)}_m(x_2))\dot\sigma(\alpha^{(L)}_m(x_3))$$
$$W^{(L)}_{mk_0}\delta_{k_1k_2}W^{(L)}_{mk_3}$$

For the convergence during training, we proceed similarily to the proof of Lemma C.1. At initialization, all terms vanish as $n_\ell\to\infty$ because all summands are independent (after taking the $n_1,\dots,n_{L-1}\to\infty$ limit) with zero mean and finite variance. During training, the weights $W^{(\ell)}$ and preactivations $\tilde\alpha^{(\ell)}$ move at a rate of $1/\sqrt{n_\ell}$ which leads to a change of order $n_\ell^{-2+1/2}=n_\ell^{-1.5}$, which vanishes for all times t too. □

C.3 The Matrix S

We now have the theoretical tools to describe the moments of the matrix S. We first give a bound for the rank of S:

Proposition C.2. *Rank*$(S)\le 2(n_1+...+n_{L-1})Nn_L$

Proof. We first observe that S is given by a sum of Nn_L matrices:

$$S_{pp'} = \sum_{i=1}^{N}\sum_{k=1}^{n_L} \partial_{ik} C \partial^2_{\theta_p \theta_p} f_{\theta,k}(x_i).$$

It is therefore sufficiant to show that the rank of each matrices $\mathcal{H}f_{\theta,k}(x) = \left(\partial^2_{\theta_p\theta_{p'}} f_{\theta,k}(x_i)\right)_{p,p'}$ is bounded by $2(n_1 + ... + n_L)$.

The derivatives $\partial_{\theta_p} f_{\theta,k}(x)$ have different definition depending on whether the parameter θ_p is a connection weight $W^{(\ell)}_{ij}$ or a bias $b^{(\ell)}_j$:

$$\begin{aligned}\partial_{W^{(\ell)}_{ij}} f_{\theta,k}(x) &= \frac{1}{\sqrt{n_\ell}} \alpha^{(\ell)}_i(x;\theta) \partial_{\tilde{\alpha}^{(\ell+1)}_j(x;\theta)} f_{\theta,k}(x) \\ \partial_{b^{(\ell)}_j} f_{\theta,k}(x) &= \beta \partial_{\tilde{\alpha}^{(\ell+1)}_j(x;\theta)} f_{\theta,k}(x)\end{aligned}$$

These formulas only depend on θ through the values $\left(\alpha^{(\ell)}_i(x;\theta)\right)_{\ell,i}$ and $\left(\partial_{\tilde{\alpha}^{(\ell)}_i(x;\theta)} f_{\theta,k}(x)\right)_{\ell,i}$ for $\ell = 1, ..., L-1$ (note that both $\alpha^{(0)}_i(x) = x_i$ and $\partial_{\tilde{\alpha}^{(L)}_i(x;\theta)} f_{\theta,k}(x) = \delta_{ik}$ do not depend on θ). Together there are $2(n_1 + ... + n_{L-1})$ of them. As a consequence, the map $\theta \mapsto \left(\partial_{\theta_p} f_{\theta,k}(x_i)\right)_p$ can be written as a composition

$$\theta \in \mathbb{R}^P \mapsto \left(\alpha^{(\ell)}_i(x;\theta), \partial_{\tilde{\alpha}^{(\ell)}_i(x;\theta)} f_{\theta,k}(x)\right)_{\ell,i} \in \mathbb{R}^{2(n_1+...+n_{L-1})} \mapsto \left(\partial_{\theta_p} f_{\theta,k}(x_i)\right)_p \in \mathbb{R}^P$$

and the matrix $\mathcal{H}f_{\theta,k}(x)$ is equal to the Jacobian of this map. By the chain rule, $\mathcal{H}f_{\theta,k}(x)$ is the matrix multiplication of the Jacobians of the two submaps, whose rank are bounded by $2(n_1 + ... + n_{L-1})$, hence bounding the rank of $\mathcal{H}f_{\theta,k}(x)$. And because S is a sum of Nn_L matrices of rank smaller than $2(n_1 + ... + n_{L-1})$, the rank of S is bounded by $2(n_1 + ... + n_{L-1})Nn_L$. □

Moments

Let us now prove Proposition C.3:

Proposition C.3. *For any loss C with BGOSS and $\sigma \in C^4_b(\mathbb{R})$, the first two moments of S take the form*

$$\begin{aligned}\mathrm{Tr}\left(S(t)\right) &= G(t)^T \nabla C(t) \\ \mathrm{Tr}\left(S(t)^2\right) &= \nabla C(t)^T \tilde{\Upsilon}(t) \nabla C(t)\end{aligned}$$

- At initialization, g_θ and f_θ converge to a (centered) Gaussian pair with covariances

$$\begin{aligned}\mathbb{E}[g_{\theta,k}(x) g_{\theta,k'}(x')] &= \delta_{kk'} \Xi^{(L)}_\infty(x,x') \\ \mathbb{E}[g_{\theta,k}(x) f_{\theta,k'}(x')] &= \delta_{kk'} \Phi^{(L)}_\infty(x,x') \\ \mathbb{E}[f_{\theta,k}(x) f_{\theta,k'}(x')] &= \delta_{kk'} \Sigma^{(L)}_\infty(x,x')\end{aligned}$$

and during training g_θ evolves according to

$$\partial_t g_{\theta,k}(x) = \sum_{i=1}^{N} \Lambda^{(L)}_\infty(x,x_i) \partial_{ik} C(Y(t)).$$

- *Uniformly over any interval* $[0,T]$ *where* $\int_0^T \|\nabla C(t)\|_2 \, dt$ *is stochastically bounded, the kernel* $\Upsilon^{(L)}$ *has a deterministic and fixed limit* $\lim_{n_{L-1}\to\infty} \cdots \lim_{n_1\to\infty} \Upsilon^{(L)}_{kk'}(x,x') = \delta_{kk'}\Upsilon^{(L)}_\infty(x,x')$ *with limiting kernel:*

$$\Upsilon^{(L)}_\infty(x,x') = \sum_{\ell=1}^{L-1} \left(\Theta^{(\ell)}_\infty(x,x')^2 \ddot{\Sigma}^{(\ell)}(x,x') + 2\Theta^{(\ell)}_\infty(x,x')\dot{\Sigma}^{(\ell)}(x,x')\right) \dot{\Sigma}^{(\ell+1)}(x,x') \cdots \dot{\Sigma}^{(L-1)}(x,x').$$

- *The higher moment* $k > 2$ *vanish:* $\lim_{n_{L-1}\to\infty} \cdots \lim_{n_1\to\infty} \mathrm{Tr}\left(S^k\right) = 0$.

Proof. The first moment of S takes the form

$$\mathrm{Tr}\,(S) = \sum_p (\nabla C)^T \mathcal{H}_{p,p} Y = (\nabla C)^T G$$

where G is the restriction to the training set of the function $g_\theta(x) = \sum_p \partial^2_{\theta_p\theta_p} f_\theta(x)$. This process is random at initialization and varies during training. Lemma C.3 below shows that, in the infinite width limit, it is a Gaussian process at initialization which then evolves according to a simple differential equation, hence describing the evolution of the first moment during training.

The second moment of S takes the form:

$$\begin{aligned}\mathrm{Tr}(S^2) &= \sum_{p_1,p_2=1}^{P} \sum_{i_1,i_2=1}^{N} \partial^2_{\theta_{p_1},\theta_{p_2}} f_{\theta,k_1}(x_1) \partial^2_{\theta_{p_2},\theta_{p_1}} f_{\theta,k_2}(x_2) c'_{i_1}(x_{i_1}) c'_{i_2}(x_{i_2}) \\ &= (\nabla C)^T \tilde{\Upsilon} \nabla C\end{aligned}$$

where $\Upsilon^{(L)}_{k_1,k_2}(x_1,x_2) = \sum_{p_1,p_2=1}^P \partial^2_{\theta_{p_1},\theta_{p_2}} f_{\theta,k_1}(x_1) \partial^2_{\theta_{p_2},\theta_{p_1}} f_{\theta,k_2}(x_2)$ is a multidimensional kernel and $\tilde{\Upsilon}$ is its Gram matrix. Lemma C.4 below shows that in the infinite-width limit, $\Upsilon^{(L)}_{k_1,k_2}(x_1,x_2)$ converges to a deterministic and time-independent limit $\Upsilon^{(L)}_\infty(x_1,x_2)\delta_{k_1k_2}$.

To show that $\mathrm{Tr}(S^k) \to 0$ for all $k > 2$, it suffices to show that $\left\|S^2\right\|_F \to 0$ as $\left|\mathrm{Tr}(S^k)\right| < \left\|S^2\right\|_F \|S\|_F^{k-2}$ and we know that $\|S\|_F \to (\partial_Y C)^T \tilde{\Upsilon} \partial_Y C$ is finite. We have that

$$\begin{aligned}\left\|S^2\right\|_F &= \sum_{i_0,i_1,i_2,i_3=1}^{N} \sum_{k_0,k_1,k_2,k_3=1}^{n_L} \Psi^{(L)}_{k_0,k_1,k_2,k_3}(x_{i_0},x_{i_1},x_{i_2},x_{i_3}) \partial_{f_{\theta,k_0}(x_{i_0})} C \partial_{f_{\theta,k_1}(x_{i_1})} C \\ &\qquad\qquad \partial_{f_{\theta,k_2}(x_{i_2})} C \partial_{f_{\theta,k_3}(x_{i_3})} C \\ &= \tilde{\Psi} \cdot (\partial_Y C)^{\otimes 4}\end{aligned}$$

for $\tilde{\Psi}$ the $Nn_L \times Nn_L \times Nn_L \times Nn_L$ finite version of

$$\begin{aligned}\Psi^{(L)}_{k_0,k_1,k_2,k_3}(x_{i_0},x_{i_1},x_{i_2},x_{i_3}) = \sum_{p_0,p_1,p_2,p_3=1}^{P} &\partial^2_{\theta_{p_0},\theta_{p_1}} f_{\theta,k_0}(x_0) \partial^2_{\theta_{p_1},\theta_{p_2}} f_{\theta,k_1}(x_1) \\ &\partial^2_{\theta_{p_2},\theta_{p_3}} f_{\theta,k_2}(x_2) \partial^2_{\theta_{p_3},\theta_{p_0}} f_{\theta,k_3}(x_3).\end{aligned}$$

which vanishes in the infinite width limit by Lemma C.5 below. □

Lemma C.3. *For any loss C with BGOSS and $\sigma \in C_b^4(\mathbb{R})$, at initialization g_θ and f_θ converge to a (centered) Gaussian pair with covariances*

$$\mathbb{E}[g_{\theta,k}(x)g_{\theta,k'}(x')] = \delta_{kk'}\Xi_\infty^{(L)}(x,x')$$
$$\mathbb{E}[g_{\theta,k}(x)f_{\theta,k'}(x')] = \delta_{kk'}\Phi_\infty^{(L)}(x,x')$$
$$\mathbb{E}[f_{\theta,k}(x)f_{\theta,k'}(x')] = \delta_{kk'}\Sigma_\infty^{(L)}(x,x')$$

and during training g_θ evolves according to

$$\partial_t g_\theta(x) = \sum_{i=1}^{N} \Lambda_\infty^{(L)}(x,x_i)D_i(t)$$

Proof. When $L = 1$, $g_\theta(x)$ is 0 for any x and θ.

For the inductive step, the trace $g_{\theta,k}^{(L+1)}(x)$ is defined recursively as

$$\frac{1}{\sqrt{n_L}}\sum_{m=1}^{n_L} g_{\theta,m}^{(L)}(x)\dot\sigma\left(\tilde\alpha_m^{(L)}(x)\right)W_{mk}^{(L)} + \mathrm{Tr}\left(\nabla f_{\theta,m}(x)\left(\nabla f_{\theta,m}(x)\right)^T\right)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x)\right)W_{mk}^{(L)}$$

First note that $\mathrm{Tr}\left(\nabla f_{\theta,m}(x)\left(\nabla f_{\theta,m}(x)\right)^T\right) = \Theta_{mm}^{(L)}(x,x)$. Now let $n_1,...n_{L-1} \to \infty$, by the induction hypothesis, the pairs $(g_{\theta,m}^{(L)}, \tilde\alpha_m^{(L)})$ converge to iid Gaussian pairs of processes with covariance $\Phi_\infty^{(L)}$ at initialization.

At initialization, conditioned on the values of $g_m^{(L)}, \tilde\alpha_m^{(L)}$ the pairs $(g_k^{(L+1)}, f_\theta)$ follow a centered Gaussian distribution with (conditioned) covariance

$$\mathbb{E}[g_{\theta,k}^{(L+1)}(x)g_{\theta,k'}^{(L+1)}(x')|g_{\theta,m}^{(L)},\tilde\alpha_m^{(L)}] = \frac{\delta_{kk'}}{n_L}\sum_{m=1}^{n_L}\left(g_{\theta,m}^{(L)}(x)\dot\sigma\left(\tilde\alpha_m^{(L)}(x)\right) + \Theta_\infty^{(L)}(x,x)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x)\right)\right)$$
$$\left(g_{\theta,m}^{(L)}(x')\dot\sigma\left(\tilde\alpha_m^{(L)}(x')\right) + \Theta_\infty^{(L)}(x',x')\ddot\sigma\left(\tilde\alpha_m^{(L)}(x')\right)\right)$$

$$\mathbb{E}[g_{\theta,k}^{(L+1)}(x)f_{\theta,k'}(x')|g_{\theta,m}^{(L)},\tilde\alpha_m^{(L)}] = \frac{\delta_{kk'}}{n_L}\sum_{m=1}^{n_L}\left(g_{\theta,m}^{(L)}(x)\dot\sigma\left(\tilde\alpha_m^{(L)}(x)\right) + \Theta_\infty^{(L)}(x,x)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x)\right)\right)$$
$$\sigma\left(\tilde\alpha_m^{(L)}(x')\right)$$

$$\mathbb{E}[f_{\theta,k}(x)f_{\theta,k'}(x')|g_{\theta,m}^{(L)},\tilde\alpha_m^{(L)}] = \frac{\delta_{kk'}}{n_L}\sum_{m=1}^{n_L}\sigma\left(\tilde\alpha_m^{(L)}(x)\right)\sigma\left(\tilde\alpha_m^{(L)}(x')\right) + \beta^2.$$

As $n_L \to \infty$, by the law of large number, these (random) covariances converge to their expectations which are deterministic, hence the pairs $(g_k^{(L+1)}, f_{\theta k})$ have asymptotically the same Gaussian distribution independent of $g_m^{(L)}, \tilde\alpha_m^{(L)}$:

$$\mathbb{E}\left[g_{\theta,k}^{(L)}(x)g_{\theta,k'}^{(L)}(x')\right] \to \delta_{kk'}\Xi_\infty^{(L)}(x,x')$$
$$\mathbb{E}\left[g_{\theta,k}^{(L)}(x)f_{\theta,k'}^{(L)}(x')\right] \to \delta_{kk'}\Phi_\infty^{(L)}(x,x)$$

$$\mathbb{E}\left[f_{\theta,k}^{(L)}(x)f_{\theta,k'}^{(L)}(x')\right] \to \delta_{kk'}\Sigma_{\infty}^{(L)}(x,x)$$

with $\Xi_{\infty}^{(1)}(x,x') = \Phi_{\infty}^{(1)}(x,x') = 0$ and

$$\begin{aligned}
\Xi_{\infty}^{(L+1)}(x,x') &= \mathbb{E}\left[gg'\dot{\sigma}(\alpha)\dot{\sigma}(\alpha')\right] \\
&+ \Theta_{\infty}^{(L)}(x',x')\mathbb{E}\left[g\dot{\sigma}(\alpha)\ddot{\sigma}(\alpha')\right] \\
&+ \Theta_{\infty}^{(L)}(x,x)\mathbb{E}\left[g'\dot{\sigma}(\alpha')\ddot{\sigma}(\alpha)\right] \\
&+ \Theta_{\infty}^{(L)}(x,x)\Theta_{\infty}^{(L)}(x',x')\mathbb{E}\left[\ddot{\sigma}(\alpha')\ddot{\sigma}(\alpha)\right] \\
&= \Xi_{\infty}^{(L)}(x,x')\dot{\Sigma}_{\infty}^{(L)}(x,x') + \left(\Phi_{\infty}^{(L)}(x,x')\Phi_{\infty}^{(L)}(x',x) + \Phi_{\infty}^{(L)}(x,x)\Phi_{\infty}^{(L)}(x',x')\right)\ddot{\Sigma}_{\infty}^{(L)}(x,x') \\
&+ \Phi_{\infty}^{(L)}(x,x')\Phi_{\infty}^{(L)}(x',x')\mathbb{E}\left[\dot{\sigma}(\alpha)\,\dddot{\sigma}(\alpha')\right] + \Phi_{\infty}^{(L)}(x,x)\Phi_{\infty}^{(L)}(x',x)\mathbb{E}\left[\,\dddot{\sigma}(\alpha)\dot{\sigma}(\alpha')\right] \\
&+ \Theta_{\infty}^{(L)}(x',x')\left(\Phi_{\infty}^{(L)}(x,x)\ddot{\Sigma}_{\infty}^{(L)}(x,x') + \Phi_{\infty}^{(L)}(x,x')\mathbb{E}\left[\dot{\sigma}(\alpha)\,\ddddot{\sigma}(\alpha')\right]\right) \\
&+ \Theta_{\infty}^{(L)}(x,x)\left(\Phi_{\infty}^{(L)}(x',x')\ddot{\Sigma}_{\infty}^{(L)}(x,x') + \Phi_{\infty}^{(L)}(x',x)\mathbb{E}\left[\,\ddddot{\sigma}(\alpha)\dot{\sigma}(\alpha')\right]\right) \\
&+ \Theta_{\infty}^{(L)}(x,x)\Theta_{\infty}^{(L)}(x',x')\ddot{\Sigma}_{\infty}^{(L)}(x,x')
\end{aligned}$$

and

$$\begin{aligned}
\Phi_{\infty}^{(L+1)}(x,x') &= \mathbb{E}\left[g\dot{\sigma}(\alpha)\sigma(\alpha')\right] + \Theta_{\infty}^{(L)}(x,x)\mathbb{E}\left[\ddot{\sigma}(\alpha)\sigma(\alpha')\right] \\
&= \Phi_{\infty}^{(L)}(x,x')\dot{\Sigma}^{(L+1)}(x,x') + \left(\Phi_{\infty}^{(L)}(x,x) + \Theta_{\infty}^{(L)}(x,x)\right)\mathbb{E}\left[\ddot{\sigma}(\alpha)\sigma(\alpha')\right]
\end{aligned}$$

where (g, g', α, α') is a Gaussian quadruple of covariance

$$\begin{pmatrix}
\Xi_{\infty}^{(L)}(x,x) & \Xi_{\infty}^{(L)}(x,x') & \Phi_{\infty}^{(L)}(x,x) & \Phi_{\infty}^{(L)}(x,x') \\
\Xi_{\infty}^{(L)}(x,x') & \Xi_{\infty}^{(L)}(x',x') & \Phi_{\infty}^{(L)}(x',x) & \Phi_{\infty}^{(L)}(x',x') \\
\Phi_{\infty}^{(L)}(x,x) & \Phi_{\infty}^{(L)}(x',x) & \Sigma_{\infty}^{(L)}(x,x) & \Sigma_{\infty}^{(L)}(x,x') \\
\Phi_{\infty}^{(L)}(x,x') & \Phi_{\infty}^{(L)}(x',x') & \Sigma_{\infty}^{(L)}(x,x') & \Sigma_{\infty}^{(L)}(x',x')
\end{pmatrix}.$$

During training, the parameters follow the gradient $\partial_t\theta(t) = (\partial_\theta Y(t))^T D(t)$. By the induction hypothesis, the traces $g_{\theta,m}^{(L)}$ then evolve according to the differential equation

$$\partial_t g_{\theta,m}^{(L)}(x) = \frac{1}{\sqrt{n_L}}\sum_{i=1}^{N}\sum_{m=1}^{n_L}\Lambda_{mm'}^{(L)}(x,x_i)\dot{\sigma}(\tilde{\alpha}_{m'}^{(L)}(x))\left(W_{m'}^{(L)}\right)^T D_i(t)$$

and in the limit as $n_1, ..., n_{L-1} \to \infty$, the kernel $\Lambda_{mm'}^{(L)}(x,x_i)$ converges to a deterministic and fixed limit $\delta_{mm'}\Lambda_{\infty}^{(L)}(x,x_i)$. Note that as n_L grows, the $g_{\theta,m}^{(L)}(x)$ move at a rate of $1/\sqrt{n_L}$ just like the pre-activations $\tilde{\alpha}_m^{(L)}$. Even though they move less and less, together they affect the trace $g_{\theta,k}^{(L+1)}$ which follows the differential equation

$$\partial_t g_{\theta,k}^{(L+1)}(x) = \sum_{i=1}^{N}\sum_{k'=1}^{n_L}\Lambda_{kk'}^{(L+1)}(x,x_i)D_{ik'}(t)$$

where

$$\begin{aligned}\Lambda^{(L+1)}_{kk'}(x,x') = &\frac{1}{n_L}\sum_{m,m'}\Lambda^{(L)}_{mm'}(x,x')\dot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\dot\sigma\left(\tilde\alpha^{(L)}_{m'}(x')\right)W^{(L)}_{mk}W^{(L)}_{m'k'}\\ &+\frac{1}{n_L}\sum_{m,m'}g^{(L)}_{\theta,m}(x)\Theta^{(L)}_{mm'}(x,x')\ddot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\dot\sigma\left(\tilde\alpha^{(L)}_{m'}(x')\right)W^{(L)}_{mk}W^{(L)}_{m'k'}\\ &+\frac{1}{n_L}\sum_{m}g^{(L)}_{\theta,m}(x)\dot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\sigma\left(\tilde\alpha^{(L)}_m(x')\right)\delta_{kk'}\\ &+\frac{2}{n_L}\sum_{m,m'}\Omega^{(L)}_{m'mm}(x',x,x)\ddot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\dot\sigma\left(\tilde\alpha^{(L)}_{m'}(x')\right)W^{(L)}_{mk}W^{(L)}_{m'k'}\\ &+\frac{1}{n_L}\sum_{m,m'}\Theta^{(L)}_{mm}(x,x)\Theta^{(L)}_{mm'}(x,x')\dddot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\dot\sigma\left(\tilde\alpha^{(L)}_{m'}(x')\right)W^{(L)}_{mk}W^{(L)}_{m'k'}\\ &+\frac{1}{n_L}\sum_{m}\Theta^{(L)}_{mm}(x,x)\ddot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\sigma\left(\tilde\alpha^{(L)}_m(x')\right)\delta_{kk'}.\end{aligned}$$

As $n_1,...,n_{L-1}\to\infty$, the kernels $\Theta^{(L)}_{mm'}(x,x')$ and $\Lambda^{(L)}_{mm'}(x,x')$ converge to their limit and $\Omega^{(L)}_{m'mm}(x',x,x)$ vanishes:

$$\begin{aligned}\Lambda^{(L)}_{kk'}(x,x') \to &\frac{1}{n_L}\sum_{m}\Lambda^{(L)}_{\infty}(x,x')\dot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\dot\sigma\left(\tilde\alpha^{(L)}_m(x')\right)W^{(L)}_{mk}W^{(L)}_{mk'}\\ &+\frac{1}{n_L}\sum_{m}g^{(L)}_{\theta,m}(x)\Theta^{(L)}_{\infty}(x,x')\ddot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\dot\sigma\left(\tilde\alpha^{(L)}_m(x')\right)W^{(L)}_{mk}W^{(L)}_{mk'}\\ &+\frac{1}{n_L}\sum_{m}g^{(L)}_{\theta,m}(x)\dot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\sigma\left(\tilde\alpha^{(L)}_m(x')\right)\delta_{kk'}\\ &+\frac{1}{n_L}\sum_{m}\Theta^{(L)}_{\infty}(x,x)\Theta^{(L)}_{\infty}(x,x')\dddot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\dot\sigma\left(\tilde\alpha^{(L)}_m(x')\right)W^{(L)}_{mk}W^{(L)}_{mk'}\\ &+\frac{1}{n_L}\sum_{m}\Theta^{(L)}_{\infty}(x,x)\ddot\sigma\left(\tilde\alpha^{(L)}_m(x)\right)\sigma\left(\tilde\alpha^{(L)}_m(x')\right)\delta_{kk'}\end{aligned}$$

By the law of large numbers, as $n_L\to\infty$, at initialization $\Lambda^{(L+1)}_{kk'}(x,x')\to\delta_{kk'}\Lambda^{(L+1)}_{\infty}(x,x')$ where

$$\begin{aligned}\Lambda^{(L+1)}_{\infty}(x,x') &= \Lambda^{(L)}_{\infty}(x,x')\dot\Sigma^{(L+1)}_{\infty}(x,x')\\ &+\Theta^{(L)}_{\infty}(x,x')\mathbb{E}\left[g\ddot\sigma\left(\alpha\right)\dot\sigma\left(\alpha'\right)\right]\\ &+\mathbb{E}\left[g\dot\sigma\left(\alpha\right)\sigma\left(\alpha'\right)\right]\\ &+\Theta^{(L)}_{\infty}(x,x)\Theta^{(L)}_{\infty}(x,x')\mathbb{E}\left[\dddot\sigma\left(\alpha\right)\dot\sigma\left(\alpha'\right)\right]\\ &+\Theta^{(L)}_{\infty}(x,x)\mathbb{E}\left[\ddot\sigma\left(\alpha\right)\sigma\left(\alpha'\right)\right]\\ &=\Lambda^{(L)}_{\infty}(x,x')\dot\Sigma^{(L+1)}_{\infty}(x,x')\\ &+\Theta^{(L)}_{\infty}(x,x')\left(\Phi^{(L)}_{\infty}(x,x')\ddot\Sigma^{(L+1)}_{\infty}(x,x')+\Phi^{(L)}_{\infty}(x,x)\mathbb{E}\left[\dddot\sigma\left(\alpha\right)\dot\sigma\left(\alpha'\right)\right]\right)\\ &+\Phi^{(L)}_{\infty}(x,x')\dot\Sigma^{(L+1)}_{\infty}(x,x')+\Phi^{(L)}_{\infty}(x,x)\mathbb{E}\left[\ddot\sigma\left(\alpha\right)\sigma\left(\alpha'\right)\right]\end{aligned}$$

$$+ \Theta_\infty^{(L)}(x,x)\Theta_\infty^{(L)}(x,x')\mathbb{E}\left[\,\ddot{\sigma}\,(\alpha)\,\dot{\sigma}\,(\alpha')\right]$$
$$+ \Theta_\infty^{(L)}(x,x)\mathbb{E}\left[\dot{\sigma}\,(\alpha)\,\dot{\sigma}\,(\alpha')\right]$$

During training $\Theta_\infty^{(L)}$ and $\Lambda_\infty^{(L)}$ are fixed in the limit $n_1,..,n_{L-1}\to\infty$, and the values $g_{\theta,m}^{(L)}(x)$, $\tilde{\alpha}_m^{(L)}(x)$ and $W_{mk}^{(L)}$ vary at a rate of $1/\sqrt{n_L}$ which induce a change of the same rate to $\Lambda_{kk'}^{(L)}(x,x')$, which is therefore asymptotically fixed during training as $n_L\to\infty$. □

The next lemma describes the asymptotic limit of the kernel $\Upsilon^{(L)}$:

Lemma C.4. *For any loss C with BGOSS and $\sigma\in C_b^4(\mathbb{R})$, the second moment of the Hessian of the realization function $\mathcal{H}F^{(L)}$ converges uniformly over $[0,T]$ to a fixed limit as $n_1,...n_{L-1}\to\infty$*

$$\Upsilon_{kk'}^{(L)}(x,x')\to\delta_{kk'}\sum_{\ell=1}^{L-1}\left(\Theta_\infty^{(\ell)}(x,x')^2\ddot{\Sigma}_\infty^{(\ell)}(x,x')+2\Theta_\infty^{(\ell)}(x,x')\dot{\Sigma}_\infty^{(\ell)}(x,x')\right)\dot{\Sigma}_\infty^{(\ell+1)}(x,x')\cdots\dot{\Sigma}_\infty^{(L-1)}(x,x').$$

Proof. The proof is by induction on the depth L. The case $L=1$ is trivially true because $\partial^2_{\theta_p\theta_{p'}}f_{\theta,k}(x)=0$ for all p,p',k,x. For the induction step we observe that

$$\begin{aligned}
&\Upsilon_{k,k'}^{(L)}(x,x')\\
&=\sum_{p_1,p_2=1}^{P}\partial^2_{\theta_{p_1},\theta_{p_2}}f_{\theta,k}(x)\partial^2_{\theta_{p_2},\theta_{p_1}}f_{\theta,k'}(x')\\
&=\frac{1}{n_L}\sum_{m,m'=1}^{n_L}\Upsilon_{m,m'}^{(L)}(x,x')\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x)\right)\dot{\sigma}\left(\tilde{\alpha}_{m'}^{(L)}(x')\right)W_{mk}^{(L)}W_{m'k'}^{(L)}\\
&+\frac{1}{n_L}\sum_{m,m'=1}^{n_L}\Omega_{m',m,m'}^{(L)}(x',x,x')\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x)\right)\ddot{\sigma}\left(\tilde{\alpha}_{m'}^{(L)}(x')\right)W_{mk}^{(L)}W_{m'k'}^{(L)}\\
&+\frac{1}{n_L}\sum_{m,m'=1}^{n_L}\Omega_{m,m',m}^{(L)}(x,x',x)\ddot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x)\right)\dot{\sigma}\left(\tilde{\alpha}_{m'}^{(L)}(x')\right)W_{mk}^{(L)}W_{m'k'}^{(L)}\\
&+\frac{1}{n_L}\sum_{m,m'=1}^{n_L}\Theta_{m,m'}^{(L)}(x,x')\Theta_{m',m}^{(L)}(x',x)\ddot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x)\right)\ddot{\sigma}\left(\tilde{\alpha}_{m'}^{(L)}(x')\right)W_{mk}^{(L)}W_{m'k'}^{(L)}\\
&+\frac{2}{n_L}\sum_{m=1}^{n_L}\Theta_{m,m'}^{(L)}(x,x')\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x)\right)\dot{\sigma}\left(\tilde{\alpha}_{m'}^{(L)}(x')\right)\delta_{kk'}
\end{aligned}$$

if we now let the width of the lower layers grow to infinity $n_1,...n_{L-1}\to\infty$, the tensor $\Omega^{(L)}$ vanishes and $\Upsilon_{m,m'}^{(L)}$ and the NTK $\Theta_{m,m'}^{(L)}$ converge to limits which are non-zero only when $m=m'$. As a result, the term above converges to

$$\begin{aligned}
&\frac{1}{n_L}\sum_{m=1}^{n_L}\Upsilon_\infty^{(L)}(x,x')\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x)\right)\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x')\right)W_{mk}^{(L)}W_{mk'}^{(L)}\\
&+\frac{1}{n_L}\sum_{m=1}^{n_L}\Theta_\infty^{(L)}(x,x')^2\ddot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x)\right)\ddot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x')\right)W_{mk}^{(L)}W_{mk'}^{(L)}
\end{aligned}$$

$$+\frac{2}{n_L}\sum_{m=1}^{n_L}\Theta_\infty^{(L)}(x,x')\dot\sigma\left(\tilde\alpha_m^{(L)}(x)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x')\right)\delta_{kk'}$$

At initialization, we can apply the law of large numbers as $n_L\to\infty$ such that it converges to $\Upsilon_\infty^{(L+1)}(x,x')\delta_{kk'}$, for the kernel $\Upsilon_\infty^{(L+1)}(x,x')$ defined recursively by

$$\Upsilon_\infty^{(L+1)}(x,x')=\Upsilon_\infty^{(L)}(x,x')\dot\Sigma_\infty^{(L)}(x,x')+\Theta_\infty^{(L)}(x,x')^2\ddot\Sigma_\infty^{(L)}(x,x')+2\Theta_\infty^{(L)}(x,x')\dot\Sigma_\infty^{(L)}(x,x')$$

and $\Upsilon_\infty^{(1)}(x,x')=0$.

For the convergence during training, we proceed similarily to the proof of Lemma C.1: the activations $\tilde\alpha_m^{(L)}(x)$ and weights $W_{mk}^{(L)}$ move at a rate of $1/\sqrt{n_L}$ and the change to $\Upsilon_{kk'}^{(L+1)}$ is therefore of order $1/\sqrt{n_L}$ and vanishes as $n_L\to 0$. □

Finally, the next lemma shows the vanishing of the tensor $\Psi^{(L)}_{k_0,k_1,k_2,k_3}$ to prove that the higher moments of S vanish.

Lemma C.5. *For any loss C with BGOSS and $\sigma\in C_b^4(\mathbb{R})$, uniformly over $[0,T]$*

$$\lim_{n_{L-1}\to\infty}\cdots\lim_{n_1\to\infty}\Psi^{(L)}_{k_0,k_1,k_2,k_3}(x_{i_0},x_{i_1},x_{i_2},x_{i_3})=0$$

Proof. When $L=1$ the Hessian is zero and $\Psi^{(1)}_{k_0,k_1,k_2,k_3}(x_{i_0},x_{i_1},x_{i_2},x_{i_3})=0$.

For the induction step, we write $\Psi^{(L+1)}_{k_0,k_1,k_2,k_3}(x_{i_0},x_{i_1},x_{i_2},x_{i_3})$ recursively, because it contains many terms, we change the notation, writing $\begin{bmatrix} x_0 & x_1\\ m_0 & m_1\end{bmatrix}$ for $\Theta^{(L)}_{m_0,m_1}(x_0,x_1)$, $\begin{bmatrix} x_0 & x_1 & x_2\\ m_0 & m_1 & m_2\end{bmatrix}$ for $\Omega^{(L)}_{m_0,m_1,m_2}(x_0,x_1,x_2)$ and $\begin{bmatrix} x_0 & x_1 & x_2 & x_3\\ m_0 & m_1 & m_2 & m_3\end{bmatrix}$ for $\Gamma^{(L)}_{m_0,m_1,m_2,m_3}(x_0,x_1,x_2,x_3)$. The value $\Psi^{(L+1)}_{k_0,k_1,k_2,k_3}(x_{i_0},x_{i_1},x_{i_2},x_{i_3})$ is then equal to

$$\begin{aligned}
&n_L^{-2}\sum_{m_0,m_1,m_2,m_3}\Psi^{(L)}_{m_0,m_1,m_2,m_3}(x_0,x_1,x_2,x_3)\dot\sigma\left(\tilde\alpha^{(L)}_{m_0}(x_0)\right)\dot\sigma\left(\tilde\alpha^{(L)}_{m_1}(x_1)\right)\dot\sigma\left(\tilde\alpha^{(L)}_{m_2}(x_2)\right)\\
&\qquad\dot\sigma\left(\tilde\alpha^{(L)}_{m_3}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_1k_1}W^{(L)}_{m_2k_2}W^{(L)}_{m_3k_3}\\
&+n_L^{-2}\sum_{m_0,m_1,m_2,m_3}\begin{bmatrix} x_0 & x_1\\ m_0 & m_1\end{bmatrix}\begin{bmatrix} x_1 & x_2\\ m_1 & m_2\end{bmatrix}\begin{bmatrix} x_2 & x_3\\ m_2 & m_3\end{bmatrix}\begin{bmatrix} x_3 & x_0\\ m_3 & m_0\end{bmatrix}\ddot\sigma\left(\tilde\alpha^{(L)}_{m_0}(x_0)\right)\\
&\qquad\ddot\sigma\left(\tilde\alpha^{(L)}_{m_1}(x_1)\right)\ddot\sigma\left(\tilde\alpha^{(L)}_{m_2}(x_2)\right)\ddot\sigma\left(\tilde\alpha^{(L)}_{m_3}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_1k_1}W^{(L)}_{m_2k_2}W^{(L)}_{m_3k_3}\\
&+n_L^{-2}\sum_{m_0,m_1,m_2,m_3}\begin{bmatrix} x_0 & x_1 & x_2\\ m_0 & m_1 & m_2\end{bmatrix}\begin{bmatrix} x_2 & x_3\\ m_2 & m_3\end{bmatrix}\begin{bmatrix} x_3 & x_0\\ m_3 & m_0\end{bmatrix}\ddot\sigma\left(\tilde\alpha^{(L)}_{m_0}(x_0)\right)\dot\sigma\left(\tilde\alpha^{(L)}_{m_1}(x_1)\right)\\
&\qquad\ddot\sigma\left(\tilde\alpha^{(L)}_{m_2}(x_2)\right)\ddot\sigma\left(\tilde\alpha^{(L)}_{m_3}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_1k_1}W^{(L)}_{m_2k_2}W^{(L)}_{m_3k_3}\\
&+n_L^{-2}\sum_{m_0,m_1,m_2,m_3}\begin{bmatrix} x_0 & x_1\\ m_0 & m_1\end{bmatrix}\begin{bmatrix} x_1 & x_2 & x_3\\ m_1 & m_2 & m_3\end{bmatrix}\begin{bmatrix} x_3 & x_0\\ m_3 & m_0\end{bmatrix}\ddot\sigma\left(\tilde\alpha^{(L)}_{m_0}(x_0)\right)\ddot\sigma\left(\tilde\alpha^{(L)}_{m_1}(x_1)\right)\\
&\qquad\dot\sigma\left(\tilde\alpha^{(L)}_{m_2}(x_2)\right)\ddot\sigma\left(\tilde\alpha^{(L)}_{m_3}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_1k_1}W^{(L)}_{m_2k_2}W^{(L)}_{m_3k_3}
\end{aligned}$$

$$+n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \begin{bmatrix} x_0 & x_1 \\ m_0 & m_1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 \\ m_1 & m_2 \end{bmatrix} \begin{bmatrix} x_2 & x_3 & x_0 \\ m_2 & m_3 & m_0 \end{bmatrix} \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3}$$

$$+n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \begin{bmatrix} x_1 & x_2 \\ m_1 & m_2 \end{bmatrix} \begin{bmatrix} x_2 & x_3 \\ m_2 & m_3 \end{bmatrix} \begin{bmatrix} x_3 & x_0 & x_1 \\ m_3 & m_0 & m_1 \end{bmatrix} \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3}$$

$$+n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \begin{bmatrix} x_0 & x_1 & x_2 \\ m_0 & m_1 & m_2 \end{bmatrix} \begin{bmatrix} x_2 & x_3 & x_0 \\ m_2 & m_3 & m_0 \end{bmatrix} \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right) \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3}$$

$$+n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \begin{bmatrix} x_1 & x_2 & x_3 \\ m_1 & m_2 & m_3 \end{bmatrix} \begin{bmatrix} x_3 & x_0 & x_1 \\ m_3 & m_0 & m_1 \end{bmatrix} \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3}$$

$$+n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ m_0 & m_1 & m_2 & m_3 \end{bmatrix} \begin{bmatrix} x_3 & x_0 \\ m_3 & m_0 \end{bmatrix} \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right) \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3}$$

$$+n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \begin{bmatrix} x_0 & x_1 \\ m_0 & m_1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 & x_3 & x_0 \\ m_1 & m_2 & m_3 & m_0 \end{bmatrix} \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3}$$

$$+n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \begin{bmatrix} x_1 & x_2 \\ m_1 & m_2 \end{bmatrix} \begin{bmatrix} x_2 & x_3 & x_0 & x_1 \\ m_2 & m_3 & m_0 & m_1 \end{bmatrix} \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3}$$

$$+n_L^{-2} \sum_{m_0,m_1,m_2,m_3} \begin{bmatrix} x_2 & x_3 \\ m_2 & m_3 \end{bmatrix} \begin{bmatrix} x_3 & x_0 & x_1 & x_2 \\ m_3 & m_0 & m_1 & m_2 \end{bmatrix} \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right) \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right) W^{(L)}_{m_0k_0} W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3}$$

$$+n_L^{-2} \sum_{m,m_1,m_2} \begin{bmatrix} x_0 & x_1 \\ m & m_1 \end{bmatrix} \begin{bmatrix} x_1 & x_2 \\ m_1 & m_2 \end{bmatrix} \begin{bmatrix} x_2 & x_3 \\ m_2 & m \end{bmatrix} \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_0)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_3)\right) W^{(L)}_{m_1k_1} W^{(L)}_{m_2k_2} \delta_{k_0k_3}$$

$$+n_L^{-2} \sum_{m,m_2,m_3} \begin{bmatrix} x_1 & x_2 \\ m & m_2 \end{bmatrix} \begin{bmatrix} x_2 & x_3 \\ m_2 & m_3 \end{bmatrix} \begin{bmatrix} x_3 & x_0 \\ m_3 & m \end{bmatrix} \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_0)\right) \dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_1)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right) \ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right) W^{(L)}_{m_2k_2} W^{(L)}_{m_3k_3} \delta_{k_0k_1}$$

$$+n_L^{-2}\sum_{m,m_3,m_0}\left[\begin{array}{cc} x_0 & x_1 \\ m_0 & m \end{array}\right]\left[\begin{array}{cc} x_2 & x_3 \\ m & m_3 \end{array}\right]\left[\begin{array}{cc} x_3 & x_0 \\ m_3 & m_0 \end{array}\right]\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_2)\right)\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_3k_3}\delta_{k_1k_2}$$
$$+n_L^{-2}\sum_{m,m_0,m_1}\left[\begin{array}{cc} x_0 & x_1 \\ m_0 & m_1 \end{array}\right]\left[\begin{array}{cc} x_1 & x_2 \\ m_1 & m \end{array}\right]\left[\begin{array}{cc} x_3 & x_0 \\ m & m_0 \end{array}\right]\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right)\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_1k_1}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_{m,m_1,m_2}\left[\begin{array}{ccc} x_0 & x_1 & x_2 \\ m & m_1 & m_2 \end{array}\right]\left[\begin{array}{cc} x_2 & x_3 \\ m_2 & m \end{array}\right]\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_0)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_3)\right)W^{(L)}_{m_1k_1}W^{(L)}_{m_2k_2}\delta_{k_0k_3}$$
$$+n_L^{-2}\sum_{m,m_2,m_3}\left[\begin{array}{ccc} x_1 & x_2 & x_3 \\ m & m_2 & m_3 \end{array}\right]\left[\begin{array}{cc} x_3 & x_0 \\ m_3 & m \end{array}\right]\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_0)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_1)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right)W^{(L)}_{m_2k_2}W^{(L)}_{m_3k_3}\delta_{k_0k_1}$$
$$+n_L^{-2}\sum_{m,m_3,m_0}\left[\begin{array}{cc} x_0 & x_1 \\ m_0 & m \end{array}\right]\left[\begin{array}{ccc} x_2 & x_3 & x_0 \\ m & m_3 & m_0 \end{array}\right]\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_3k_3}\delta_{k_1k_2}$$
$$+n_L^{-2}\sum_{m,m_0,m_1}\left[\begin{array}{cc} x_1 & x_2 \\ m_1 & m \end{array}\right]\left[\begin{array}{ccc} x_3 & x_0 & x_1 \\ m & m_0 & m_1 \end{array}\right]\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right)\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_1k_1}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_{m,m_1,m_2}\left[\begin{array}{cc} x_0 & x_1 \\ m & m_1 \end{array}\right]\left[\begin{array}{ccc} x_1 & x_2 & x_3 \\ m_1 & m_2 & m \end{array}\right]\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_0)\right)\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_3)\right)W^{(L)}_{m_1k_1}W^{(L)}_{m_2k_2}\delta_{k_0k_3}$$
$$+n_L^{-2}\sum_{m,m_2,m_3}\left[\begin{array}{cc} x_1 & x_2 \\ m & m_2 \end{array}\right]\left[\begin{array}{ccc} x_2 & x_3 & x_0 \\ m_2 & m_3 & m \end{array}\right]\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_0)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_1)\right)$$
$$\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_2}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right)W^{(L)}_{m_2k_2}W^{(L)}_{m_3k_3}\delta_{k_0k_1}$$
$$+n_L^{-2}\sum_{m,m_3,m_0}\left[\begin{array}{cc} x_2 & x_3 \\ m & m_3 \end{array}\right]\left[\begin{array}{ccc} x_3 & x_0 & x_1 \\ m_3 & m_0 & m \end{array}\right]\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_2)\right)\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_3}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_3k_3}\delta_{k_1k_2}$$
$$+n_L^{-2}\sum_{m,m_0,m_1}\left[\begin{array}{ccc} x_0 & x_1 & x_2 \\ m_0 & m_1 & m \end{array}\right]\left[\begin{array}{cc} x_3 & x_0 \\ m & m_0 \end{array}\right]\ddot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_0}(x_0)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m_1}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}^{(L)}_{m}(x_3)\right)W^{(L)}_{m_0k_0}W^{(L)}_{m_1k_1}\delta_{k_2k_3}$$

$$+n_L^{-2}\sum_{m,m_1,m_2}\begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ m & m_1 & m_2 & m \end{bmatrix}\dot\sigma\left(\tilde\alpha_m^{(L)}(x_0)\right)\dot\sigma\left(\tilde\alpha_{m_1}^{(L)}(x_1)\right)\dot\sigma\left(\tilde\alpha_{m_2}^{(L)}(x_2)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_3)\right)$$
$$W_{m_1k_1}^{(L)}W_{m_2k_2}^{(L)}\delta_{k_0k_3}$$
$$+n_L^{-2}\sum_{m,m_2,m_3}\begin{bmatrix} x_1 & x_2 & x_3 & x_0 \\ m & m_2 & m_3 & m \end{bmatrix}\dot\sigma\left(\tilde\alpha_m^{(L)}(x_0)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_1)\right)\dot\sigma\left(\tilde\alpha_{m_2}^{(L)}(x_2)\right)\dot\sigma\left(\tilde\alpha_{m_3}^{(L)}(x_3)\right)$$
$$W_{m_2k_2}^{(L)}W_{m_3k_3}^{(L)}\delta_{k_0k_1}$$
$$+n_L^{-2}\sum_{m,m_3,m_0}\begin{bmatrix} x_2 & x_3 & x_0 & x_1 \\ m & m_3 & m_0 & m \end{bmatrix}\dot\sigma\left(\tilde\alpha_{m_0}^{(L)}(x_0)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_1)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_2)\right)\dot\sigma\left(\tilde\alpha_{m_3}^{(L)}(x_3)\right)$$
$$W_{m_0k_0}^{(L)}W_{m_3k_3}^{(L)}\delta_{k_1k_2}$$
$$+n_L^{-2}\sum_{m,m_0,m_1}\begin{bmatrix} x_3 & x_0 & x_1 & x_2 \\ m & m_0 & m_1 & m \end{bmatrix}\dot\sigma\left(\tilde\alpha_{m_0}^{(L)}(x_0)\right)\dot\sigma\left(\tilde\alpha_{m_1}^{(L)}(x_1)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_2)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_3)\right)$$
$$W_{m_0k_0}^{(L)}W_{m_1k_1}^{(L)}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_{m,m'}\begin{bmatrix} x_0 & x_1 \\ m & m' \end{bmatrix}\begin{bmatrix} x_2 & x_3 \\ m' & m \end{bmatrix}\dot\sigma\left(\tilde\alpha_m^{(L)}(x_0)\right)\dot\sigma\left(\tilde\alpha_{m'}^{(L)}(x_1)\right)\dot\sigma\left(\tilde\alpha_{m'}^{(L)}(x_2)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_3)\right)$$
$$\delta_{k_0k_1}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_{m,m'}\begin{bmatrix} x_1 & x_2 \\ m & m' \end{bmatrix}\begin{bmatrix} x_3 & x_0 \\ m' & m \end{bmatrix}\dot\sigma\left(\tilde\alpha_m^{(L)}(x_0)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_1)\right)\dot\sigma\left(\tilde\alpha_{m'}^{(L)}(x_2)\right)\dot\sigma\left(\tilde\alpha_{m'}^{(L)}(x_3)\right)$$
$$\delta_{k_0k_3}\delta_{k_1k_2}$$

Even though this is a very large formula one can notice that most terms are "rotation of each other". Moreover, as $n_1,...,n_{L-1}\to\infty$, all terms containing either an $\Psi^{(L)}$, an $\Omega^{(L)}$ or a $\Gamma^{(L)}$ vanish. For the remaining terms, we may replace the NTKs $\Theta^{(L)}$ by their limit and as a result $\Psi^{(L+1)}_{k_0,k_1,k_2,k_3}(x_{i_0},x_{i_1},x_{i_2},x_{i_3})$ converges to

$$n_L^{-2}\sum_m\Theta_\infty^{(L)}(x_0,x_1)\Theta_\infty^{(L)}(x_1,x_2)\Theta_\infty^{(L)}(x_2,x_3)\Theta_\infty^{(L)}(x_3,x_0)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_0)\right)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_1)\right)$$
$$\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_2)\right)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_3)\right)W_{mk_0}^{(L)}W_{mk_1}^{(L)}W_{mk_2}^{(L)}W_{mk_3}^{(L)}$$
$$+n_L^{-2}\sum_m\Theta_\infty^{(L)}(x_0,x_1)\Theta_\infty^{(L)}(x_1,x_2)\Theta_\infty^{(L)}(x_2,x_3)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_0)\right)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_1)\right)$$
$$\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_2)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_3)\right)W_{mk_1}^{(L)}W_{mk_2}^{(L)}\delta_{k_0k_3}$$
$$+n_L^{-2}\sum_m\Theta_\infty^{(L)}(x_1,x_2)\Theta_\infty^{(L)}(x_2,x_3)\Theta_\infty^{(L)}(x_3,x_0)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_0)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_1)\right)$$
$$\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_2)\right)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_3)\right)W_{mk_2}^{(L)}W_{mk_3}^{(L)}\delta_{k_0k_1}$$
$$+n_L^{-2}\sum_m\Theta_\infty^{(L)}(x_0,x_1)\Theta_\infty^{(L)}(x_2,x_3)\Theta_\infty^{(L)}(x_3,x_0)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_0)\right)\dot\sigma\left(\tilde\alpha_m^{(L)}(x_1)\right)$$
$$\dot\sigma\left(\tilde\alpha_m^{(L)}(x_2)\right)\ddot\sigma\left(\tilde\alpha_m^{(L)}(x_3)\right)W_{mk_0}^{(L)}W_{mk_3}^{(L)}\delta_{k_1k_2}$$

$$+n_L^{-2}\sum_m \Theta_\infty^{(L)}(x_0,x_1)\Theta_\infty^{(L)}(x_1,x_2)\Theta_\infty^{(L)}(x_3,x_0)\ddot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_0)\right)\ddot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_3)\right)W_{mk_0}^{(L)}W_{mk_1}^{(L)}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_m \Theta_\infty^{(L)}(x_0,x_1)\Theta_\infty^{(L)}(x_2,x_3)\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_0)\right)\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_3)\right)\delta_{k_0k_1}\delta_{k_2k_3}$$
$$+n_L^{-2}\sum_m \Theta_\infty^{(L)}(x_1,x_2)\Theta_\infty^{(L)}(x_3,x_0)\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_0)\right)\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_1)\right)$$
$$\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_2)\right)\dot{\sigma}\left(\tilde{\alpha}_m^{(L)}(x_3)\right)\delta_{k_0k_3}\delta_{k_1k_2}$$

And all these sums vanish as $n_L \to \infty$ thanks to the prefactor n_L^{-2}, proving the vanishing of $\Psi^{(L+1)}_{k_0,k_1,k_2,k_3}(x_{i_0},x_{i_1},x_{i_2},x_{i_3})$ in the infinite width limit.

During training, the activations $\tilde{\alpha}_m^{(L)}(x)$ and weights $W_{mk}^{(L)}$ move at a rate of $1/\sqrt{n_L}$ which induces a change to $\Psi^{(L+1)}$ of order $n_L^{-3/2}$ which vanishes in the infinite width limit. □

C.4 Orthogonality of I and S

From Lemma C.2 and the vanishing of the tensor $\Gamma^{(L)}$ as proven in Lemma C.2, we can easily prove the orthogonality of I and S of Proposition C.4:

Proposition C.4. *For any loss C with BGOSS and $\sigma \in C_b^4(\mathbb{R})$, we have uniformly over $[0,T]$*

$$\lim_{n_{L-1}\to\infty}\cdots\lim_{n_1\to\infty}\|IS\|_F = 0.$$

As a consequence $\lim_{n_{L-1}\to\infty}\cdots\lim_{n_1\to\infty}\mathrm{Tr}\left([I+S]^k\right)-\left[\mathrm{Tr}\left(I^k\right)+\mathrm{Tr}\left(S^k\right)\right]=0.$

Proof. The Frobenius norm of IS is equal to

$$\begin{aligned}\|IS\|_F^2 &= \left\|\mathcal{D}Y\mathcal{H}C\left(\mathcal{D}Y\right)^T\left(\nabla C\cdot\mathcal{H}Y\right)\right\|_F^2\\ &= \sum_{p_1,p_2=1}^{P}\left(\sum_{p=1}^{P}\sum_{i_1,i_2=1}^{N}\sum_{k_1,k_2=1}^{n_L}\partial_{\theta_{p_1}}f_{\theta,k_1}(x_{i_1})c''_{k_1}(x_{i_1})\partial_{\theta_p}f_{\theta,k_1}(x_{i_1})\partial^2_{\theta_p,\theta_{p_3}}f_{\theta,k_2}(x_2)(x_{i_2})c'_{k_2}(x_{i_2})\right)^2\\ &= \sum_{i_1,i_2,i'_1,i'_2=1}^{N}\sum_{k_1,k_2,k'_1,k'_2=1}^{n_L}c''_{k_1}(x_{i_1})c''_{k'_1}(x_{i'_1})c'_{k_2}(x_{i_2})c'_{k'_2}(x_{i'_2})\Theta_{k_1,k'_1}(x_{i_1},x_{i'_1})\Gamma_{k_1,k_2,k'_2,k'_1}(x_{i_1},x_{i_2},x_{i'_2},x_{i'_1})\end{aligned}$$

and Γ vanishes as $n_1,...,n_{L-1}\to\infty$ by Lemma C.2.

The k-th moment of the sum $\mathrm{Tr}\left(I+S\right)^k$ is equal to the sum over all $\mathrm{Tr}\left(A_1\cdots A_k\right)$ for any word $A_1\ldots A_k$ of $A_i\in\{I,S\}$. The difference $\mathrm{Tr}\left([I+S]^k\right)-\left[\mathrm{Tr}\left(I^k\right)+\mathrm{Tr}\left(S^k\right)\right]$ is hence equal to the sum over all mixed words, i.e. words $A_1\ldots A_k$ which contain at least one I and one S. Such words

must contain two consecutive terms $A_m A_{m+1}$ one equal to I and the other equal to S. We can then bound the trace by

$$|\mathrm{Tr}\,(A_1 \cdots A_k)| \leq N n_L \,\|A_1\|_F \cdots \|A_{m-1}\|_F \,\|A_m A_{m+1}\|_F \,\|A_{m+2}\|_F \cdots \|A_k\|_F$$

which vanishes in the infinite width limit because $\|I\|_F$ and $\|S\|_F$ are bounded and $\|A_m A_{m+1}\|_F = \|IS\|_F$ vanishes. □

Appendix D

Kernel Alignment Ridge Estimator: Risk Prediction From Training Data

We organize the Supplementary Material (Supp. Mat.) as follows:

1. In Section D.1, we present the details for the numerical results presented in the main text (and in the Supp. Mat.) and we present additional experiments and some discussions.
2. In Section H.3, we present the proofs of the mathematical results presented in the main text.

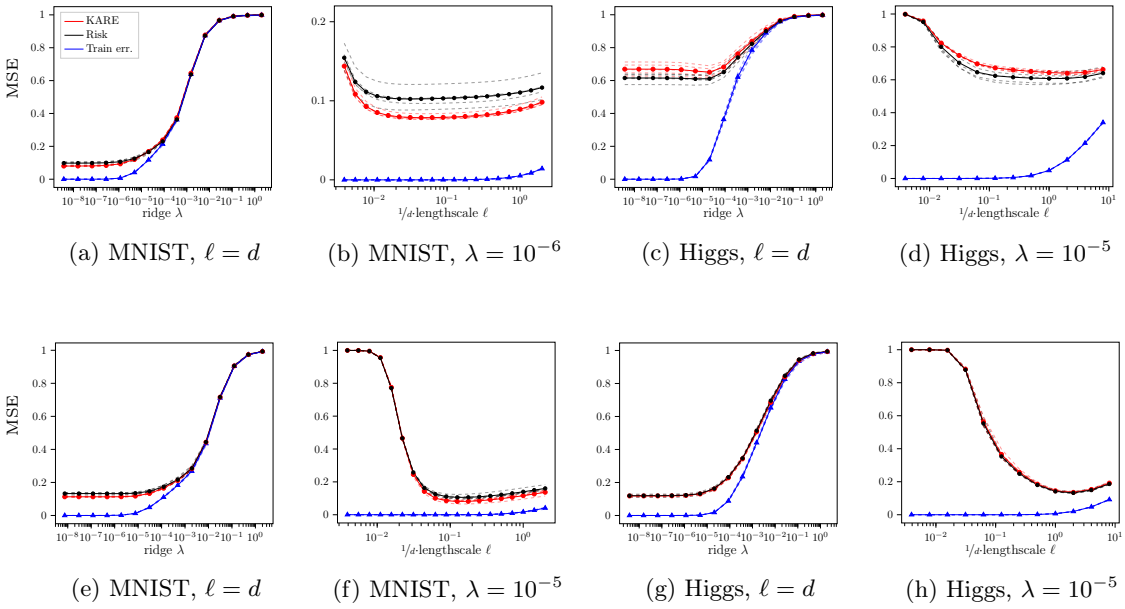
D.1 Numerical Results

Empirical Methods

For the MNIST dataset. We sample N images of digits 7 and 9 from the MNIST training dataset (image size $d = 24 \times 24$, edge pixels cropped, all pixels rescaled down to $[0, 1]$ and recentered around the mean value) and label each of them with $+1$ and -1 labels. We perform KRR with various ridge λ on this dataset with the selected kernel k times and calculate the MSE training error, risk, and the KARE for every trial ($k = 10$ for small N and $k = 5$ for $N = 2000$). The risk is approximated using other $N_2 = 1000$ random samples of the MNIST training data.

For the Higgs Dataset. We randomly choose N samples among those that do not have any missing features marked with -999 from the Higgs training dataset. The samples have $d = 31$ features, and we normalize each feature column down to $[0, 1]$ by dividing by the maximum absolute value observed among the selected samples. We replace the categorical labels 's' and 'b' with regression values $+1$ and -1 respectively and perform KRR with various ridge λ. We repeat this procedure k times, which corresponds to sampling k different training datasets of $N = 1000$ samples to perform kernel regression, and calculate the MSE training error, the risk, and the KARE for every trial ($k = 10$ for small N and $k = 5$ for $N = 1000$). The risk is approximated using other $N_2 = 1000$ random samples of the Higgs training data.

KARE predicts risk for various Kernels



(a) MNIST, $\ell = d$ (b) MNIST, $\lambda = 10^{-6}$ (c) Higgs, $\ell = d$ (d) Higgs, $\lambda = 10^{-5}$

(e) MNIST, $\ell = d$ (f) MNIST, $\lambda = 10^{-5}$ (g) Higgs, $\ell = d$ (h) Higgs, $\lambda = 10^{-5}$

Figure D.1.1: Comparison between the KRR risk and the KARE for various choices of normalized lengthscale ℓ/d and ridge λ on the MNIST dataset (restricted to the digits 7 and 9, labeled by 1 and -1 respectively, $N = 2000$) and on the Higgs dataset (classes 'b' and 's', labeled by -1 and 1, $N = 1000$). We present the results for the Laplacian Kernel $K(x, x') = \exp(-\|x-x'\|_2/\ell)$ (top row) and the ℓ_1-norm Kernel $K(x, x') = \exp(-\|x-x'\|_1/\ell)$ (bottom row). KRR predictor risks, and KARE curves (shown as dashed lines, 5 samples) concentrate around their respective averages (solid lines).

KRR predictor in function space

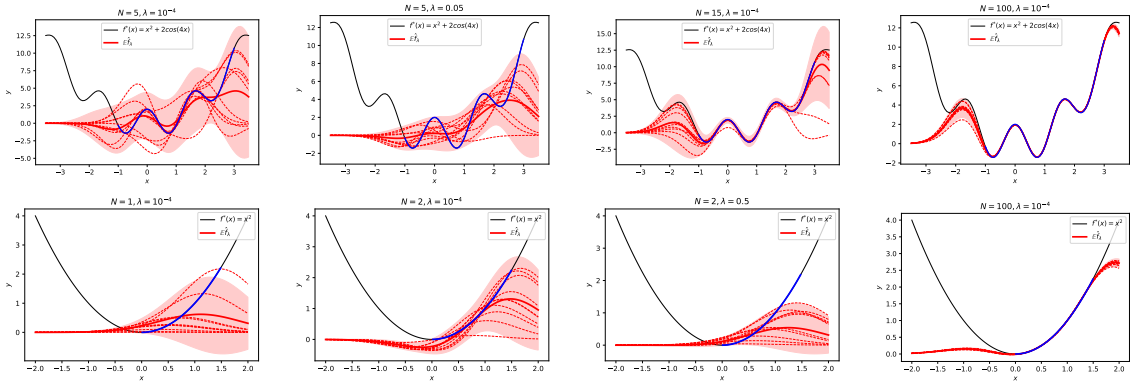
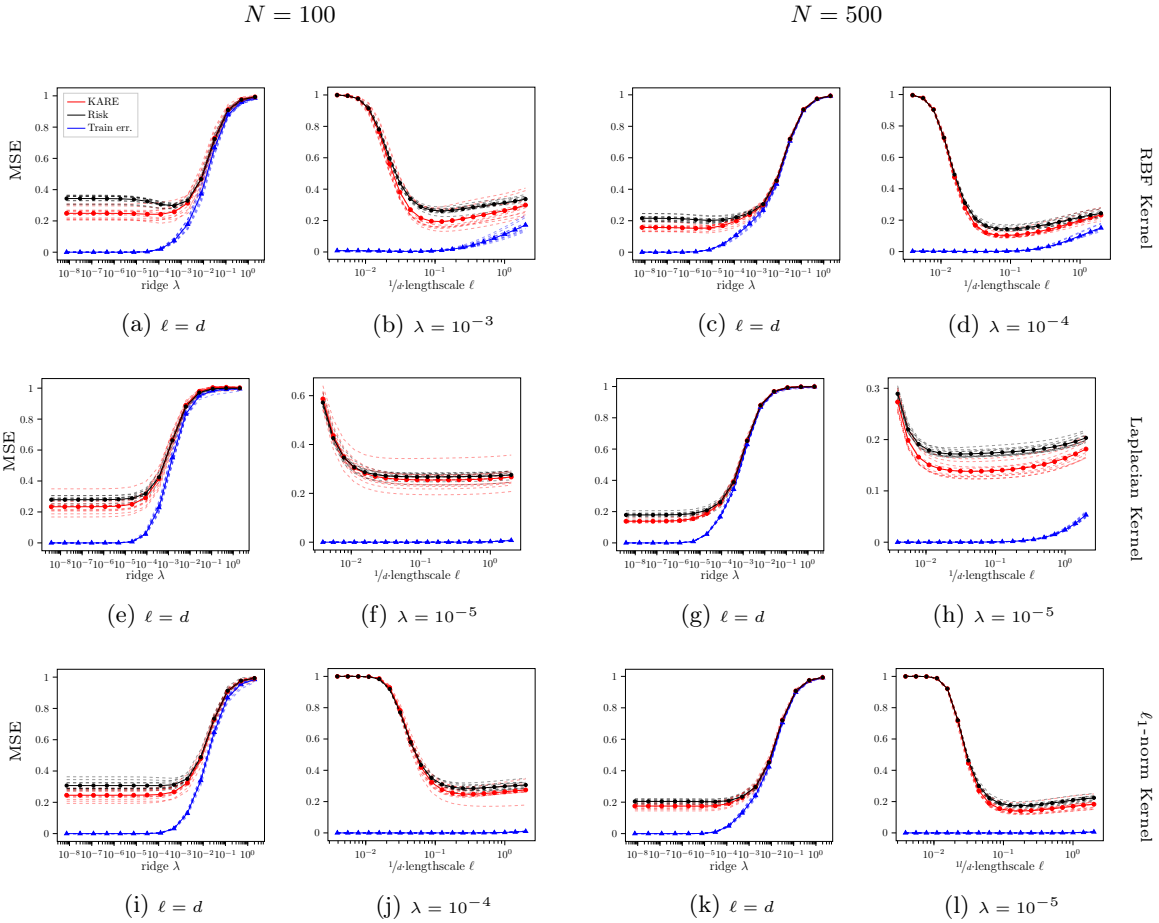


Figure D.1.2: *KRR predictor in function space for various N and λ for the RBF Kernel K with $\ell = d = 1$.* Observations $o = \delta_x$ are sampled with uniform distribution on $x \sim U[-1, 3]$ (shown in blue) $\hat{f}^{\epsilon}_{\lambda}$ is calculated 500 times for different realizations of the training data (10 example predictors are shown in dashed lines), its mean and ± 2 standard deviation are shown in red. The true function $f^*(x) = x^2 + 2\cos(4x)$ is shown in black. *Second row.* Observations $o = \delta_x$ are sampled with uniform distribution $x \sim U[0, 1.5]$ (shown in blue) and $\hat{f}^{\epsilon}_{\lambda}$ is calculated 100 times. The true function $f^*(x) = x^2$ is shown in black.

KARE predicts risk in average for small N



(a) $\ell = d$ (b) $\lambda = 10^{-3}$ (c) $\ell = d$ (d) $\lambda = 10^{-4}$

(e) $\ell = d$ (f) $\lambda = 10^{-5}$ (g) $\ell = d$ (h) $\lambda = 10^{-5}$

(i) $\ell = d$ (j) $\lambda = 10^{-4}$ (k) $\ell = d$ (l) $\lambda = 10^{-5}$

Figure D.1.3: *The estimation predicts the risk in average for small $N = \{100, 500\}$ on MNIST data.* In the top row, we used the RBF Kernel $K(x, z) = \exp(-\|x-z\|_2^2/\ell)$, in the second row, we used the Laplacian Kernel $K(x, z) = \exp(-\|x-z\|_2/\ell)$, and in the bottom row, we used the ℓ_1-norm Kernel $K(x, z) = \exp(-\|x-z\|_1/\ell)$ for various choices of ℓ and λ. The optimal predictor is calculated using N random samples ($N = 100$ for the plots on the left and $N = 500$ for the ones on the right) from the training data 10 times (dashed curves) and their average is plotted in the solid curves.

SCT and its behavior

In general, it is hard to compute the spectrum $(d_k)_{k\in\mathbb{N}}$ of T_K even when one has the knowledge of the true data distribution. Luckily, following an adaptation from [220, 58], we can obtain an explicit formula for d_k for centered d-dimensional Gaussian distribution with covariance matrix $\sigma^2 I_d$, and RBF Kernel $K(x, x') = \exp(-\|x-x'\|^2/\ell)$. The formula for the distinct eigenvalues λ_k is

$$\lambda_k = \left(\sqrt{\frac{1}{2A\sigma^2}}\right)^d B^k, \tag{D.1.1}$$

where $A = \frac{1}{4\sigma^2} + \frac{1}{\ell} + c$, $B = \frac{1}{A\ell}$ with $c = \frac{1}{2\sigma}\sqrt{\frac{1}{4\sigma^2} + \frac{2}{\ell}}$. Each λ_k has multiplicity

$$n_d(k) = \sum_{j=1}^{k} \binom{d}{j}\binom{k-1}{j-1} \tag{D.1.2}$$

for $k \geq 1$. In particular, we have $n_d(0) = 1, n_d(1) = \binom{d}{1}, n_d(2) = \binom{d}{2} + d, \ldots$. In general, $n_d(k)$ is the number of ways to partition k into d non-negative integers.

The true SCT is therefore approximated solving the following equation numerically

$$\vartheta = \lambda + \frac{\vartheta}{N}\sum_{i=1}^{k} \frac{n_d(k)\lambda_k}{\lambda_k + \vartheta}. \tag{D.1.3}$$

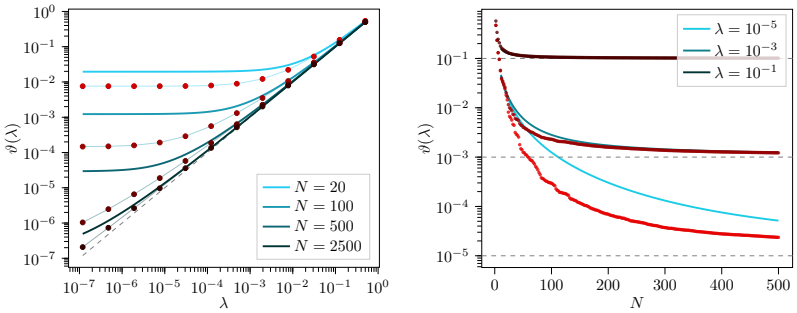


Figure D.1.4: *Behavior of SCT as a function of λ and N.* True SCT is calculated on the $k = 50$ biggest distinct eigenvalues using the formula D.1.3 for $\ell = d = 5$ and $\sigma = 1$. Red dots are the approximations obtained using Proposition 5 in the main text, i.e. $\vartheta \approx 1/\mathrm{Tr}[(\frac{1}{N}K(X,X) - \lambda I)^{-1}]$.

Note that in the Figure 2 in the main text, we limit the approximation to $k = 10$ for $d = 20$ because the multiplicity $n_d(k)$ grows polynomially with d^k.

D.2 Proofs

Preliminary: Big-P notation

Throughout our proofs, we will frequently rely on a polynomial analogue of the big-O notation, which we call big-P:

Definition 7. *For two functions f and g (of one or several variables, defined on an arbitrary common domain $\mathcal{D}$), we write $f = \mathcal{P}(g)$ if g is nonnegative over $\mathcal{D}$ and there exists a polynomial $\mathbf{P}$ with nonnegative coefficients and $\mathbf{P}(0) = 0$ such that $|f| \leq \mathbf{P}(g)$ over $\mathcal{D}$.*

Note that the big-O notation corresponds to the case when the polynomial $\mathbf{P}$ is of degree at most one.

Gaussianity Assumption

For the sake of simplicity, our proof are made under the assumptions that the observations are Gaussian. However we conjecture that as long as the higher moments are bounded/small enough, the general non-Gaussian case can be reduced to the Gaussian case, up to a small error (as it is common in random matrix theory).

There are two special cases where a weaker Gaussianity property applies, i.e. that the $o_i K \in \mathcal{C}$ are Gaussian processes, and is enough for our proofs as everytime $\mathcal{O}$ appears in the formulas, it is composed with K. These two special cases are:

1. For the linear kernel $K(x, y) = x^T y$, $o_i K$ is the linear function $x \mapsto x_i^T x$, which is Gaussian whenever the inputs x_i are sampled from a Gaussian distribution.
2. As noted in [56], this linear case can be generalized to a broader (non-linear) family kernel K in the large input space limit: as shown in [55], in the large width limit the kernel Gram matrix G for such kernel K can be approximated by the Gram matrix of a linear kernel (up to scaling) hence leading back to the previous point.

Let us observe that all the quantities we study (the predictor, the risk and empirical risk) stay the same if any observation o_i is replaced by $-o_i$. Hence a posteriori, by a symmetrization trick we may remove the assumption that the observations are centered (as in general they are not).

Objects of Interest and general strategy

The central object of our analysis is the $N \times N$ Gram matrix $\mathcal{O}K\mathcal{O}^T$, in particular the related Stieltjes transform:

$$m(z) = \frac{1}{N} \mathrm{Tr}\left[B(z)^{-1}\right]$$

where $B(z) = \frac{1}{N}\mathcal{O}K\mathcal{O}^T - zI_N$ and $z \in \mathbb{C} \setminus \mathbb{R}_+$.

From now on, we consider only $z \in \mathbb{H}_{<0} = \{z : \Re(z) < 0\}$. Note that $m(z) = \frac{1}{N}\sum_\ell \frac{1}{\lambda_\ell - z}$ where $\lambda_\ell \geq 0$ are the real eigenvalues of $\frac{1}{N}\mathcal{O}K\mathcal{O}^T$, hence $m(z)$ lies in the cone Γ spanned by 1 and $-1/z$, i.e. $\Gamma = \{a - b\frac{1}{z} | a, b \geq 0\}$. We will first show that for $z \in \mathbb{H}_{<0}$, the Stieltjes transform concentrates around the unique solution $\tilde{m}(z)$ to the equation

$$\tilde{m}(z) = -\frac{1}{z}\left(1 - \frac{1}{N}\mathrm{Tr}\left[\tilde{m}(z) T_K \left(I_{\mathcal{C}} + \tilde{m}(z) T_K\right)^{-1}\right]\right), \tag{D.2.1}$$

and then show that the linear map

$$A(z) = \frac{1}{N} K\mathcal{O}^T \left(\frac{1}{N} \mathcal{O} K \mathcal{O}^T - z I_N \right)^{-1} \mathcal{O} = \frac{1}{N} K \mathcal{O}^T B(z)^{-1} \mathcal{O}$$

concentrates around the map $\tilde{A}_{\vartheta(-z)} = T_K \left(T_K + \vartheta(-z) I_{\mathcal{C}} \right)^{-1}$, where $(T_K f)(x) = \mathbb{E}_{o\sim\pi} \left[o(K(x,\cdot)) o(f) \right] = \langle K(x,\cdot), f \rangle_S$ and $\vartheta(-z) = \frac{1}{\bar{m}(z)}$ is the Signal Capture Threshold. From Equation (D.2.1), the SCT can be also defined as the solution to the equation

$$\vartheta(-z) = -z + \frac{\vartheta(-z)}{N} \mathrm{Tr} \left[T_K \left(T_K + \vartheta(-z) I_{\mathcal{C}} \right)^{-1} \right]. \tag{D.2.2}$$

From now on, we denote $\vartheta(-z)$ by ϑ. Note that here, in the Appendix, we use the resolvent notation: in particular the KRR reconstruction operator A_λ is equal to $A(-\lambda)$.

Spectral decomposition and generalized matrix representation

Throughout this paper it is assumed that there exists an orthonormal basis of continuous functions $\left(f^{(k)}\right)_k$ for the scalar product $\langle \cdot, \cdot \rangle_S$ such that $K = \sum_{k\in\mathbb{N}} d_k f^{(k)} \otimes f^{(k)}$ and $\sum_{k\in\mathbb{N}} d_k < \infty$. For a linear map $M : \mathcal{C} \to \mathcal{C}$, we define the $(k,\ell)-$entry of M as:

$$M_{k\ell} = \left\langle f^{(k)}, M f^{(\ell)} \right\rangle_S.$$

With this notation, the trace of a linear map M becomes $\mathrm{Tr}\,(M) = \sum_{k\in\mathbb{N}} M_{kk}$.

Similarly, using the canonical basis $(b_i)_{i=1,\dots,N}$ of $\mathbb{R}^N$, we define the entries of $\mathcal{O} : \mathcal{C} \to \mathbb{R}^N$ and $\mathcal{O}^T : \mathbb{R}^N \to \mathcal{C}^*$ by

$$\mathcal{O}_{ik} = b_i \cdot \mathcal{O} f^{(k)} = o_i(f^{(k)}), \quad \mathcal{O}^T_{ik} = \mathcal{O}^T b_k(f^{(i)}) = b_k \cdot \mathcal{O} f^{(i)} = o_k(f^{(i)}).$$

Since the observations o_i are i.i.d. Gaussians with zero mean and covariance $\mathbb{E}\left[o_i(f) o_i(g)\right] = \langle f, g \rangle_S$ and since $\left(f^{(k)}\right)_k$ is an orthonormal basis for the scalar product $\langle \cdot, \cdot \rangle_S$, the entries $\mathcal{O}_{ik}$ are i.i.d standard Gaussians.

Using the spectral decomposition of K, the entries of $\mathcal{O} K \mathcal{O}^T$ are given by:

$$\left(\mathcal{O} K \mathcal{O}^T\right)_{i,j} = \sum_\ell d_\ell o_i(f^{(\ell)}) o_j(f^{(\ell)}),$$

where the sum converges absolutely (thanks to the trace assumption on K) and the entries of A are then given by:

$$A_{k\ell}(z) = \frac{d_k}{N} \left(\mathcal{O}_{\cdot k}\right)^T \left(\frac{1}{N} \mathcal{O} K \mathcal{O}^T - z I_N \right)^{-1} \mathcal{O}_{\cdot \ell} \tag{D.2.3}$$

where $\mathcal{O}_{\cdot k} = \left(o_i(f^{(k)})\right)_{i=1,\dots,N}$.

Shermann-Morrison Formula

The Shermann-Morrison formula allows one to study how the inverse of a matrix is modified by a rank one perturbation of the matrix. The matrix $\mathcal{O} K \mathcal{O}^T$ can be seen as a perturbation of $\mathcal{O} K_{(k)} \mathcal{O}^T$

by the rank one matrix $d_k \mathcal{O}_{\cdot k} \mathcal{O}_{\cdot k}^T$, where $K_{(k)} := \sum_{\ell \neq k} d_\ell f^{(\ell)} \otimes f^{(\ell)}$. By doing so, one isolates the contribution of the k-th eigenvalue of K. Thus, one can compute $B(z)^{-1} = \left(\frac{1}{N}\mathcal{O}K\mathcal{O}^T - zI_N\right)^{-1}$ using the Shermann-Morrison formula:

$$B(z)^{-1} = B_{(k)}(z)^{-1} - \frac{1}{N}\frac{d_k}{1+d_k g_k(z)} B_{(k)}(z)^{-1}\mathcal{O}_{\cdot k}\mathcal{O}_{\cdot k}^T B_{(k)}(z)^{-1} \tag{D.2.4}$$

where $B_{(k)}(z) = \frac{1}{N}\mathcal{O}K_{(k)}\mathcal{O}^T - zI_N$ and $g_k(z) = \frac{1}{N}\mathcal{O}_{\cdot k}^T B_{(k)}(z)^{-1}\mathcal{O}_{\cdot k}$. A crucial property is that, since $o_i\left(f^{(k)}\right)$ does not appear anymore in $\mathcal{O}K_{(k)}\mathcal{O}^T$ and, since for any $\ell \neq k$ and any i, j, we have that $o_i\left(f^{(k)}\right)$ is independent from $o_j(f^{(\ell)})$, we obtain that the matrix $B_{(k)}(z)^{-1}$ is independent of $\mathcal{O}_{\cdot k}$.

Remark D.1. Using the diagonalization of $B_{(k)}(z)^{-1} = U^T \mathrm{diag}\left(\frac{1}{\nu_\ell - z}\right) U$ with U orthogonal and $\nu_\ell \geq 0$, we have that $g_k(z) = \frac{1}{N}\sum_\ell \frac{\left[\sum_i U_{\ell,i} o_i(f^{(k)})\right]^2}{\nu_\ell - z}$ lies in the cone spanned by 1 and $-1/z$, in particular, $\Re(g_k) \geq 0$ on $\mathbb{H}_{<0}$.

As a result of Equations (D.2.3) and (D.2.4), the diagonal entries of the operator $A(z) = \frac{1}{N}K\mathcal{O}^T B(z)^{-1}\mathcal{O}$ are equal to

$$A_{kk}(z) = \frac{d_k g_k(z)}{1+d_k g_k(z)}. \tag{D.2.5}$$

Remark D.2. For any $z \in \mathbb{H}_{<0}$, the sum $\sum_k |A_{kk}(z)|$ is almost surely finite. Indeed, notice that

$$\left|\frac{d_k g_k(z)}{1+d_k g_k(z)}\right| \leq |d_k g_k(z)| \leq \frac{1}{N} d_k \left\|\mathcal{O}_{\cdot k}\right\|^2 \left\|B_{(k)}(z)^{-1}\right\|_{\mathrm{op}}.$$

For any $z \in \mathbb{H}_{<0}$, $\left\|B_{(k)}(z)^{-1}\right\|_{\mathrm{op}} \leq \frac{1}{|z|}$ and thus

$$\left|\frac{d_k g_k(z)}{1+d_k g_k(z)}\right| \leq \frac{1}{N|z|} d_k \left\|\mathcal{O}_{\cdot k}\right\|^2.$$

Since $\mathbb{E}\left[\sum_k d_k \left\|\mathcal{O}_{\cdot k}\right\|^2\right] = N\mathrm{Tr}[T_K] < \infty$, we have that $\sum_k |A_{kk}(z)|$ is almost surely finite.

The operator A is therefore a.s. trace-class and $\mathrm{Tr}(A) = \sum_k \frac{d_k g_k(z)}{1+d_k g_k(z)}$, where the sum is absolutely convergent.

Another important observation is that the Stieltjes transform $m(z)$ and the $g_k(z)$ are closely related.

Lemma D.1. *For any $z \in \mathbb{H}_{<0}$, a.s. we have*

$$m(z) = -\frac{1}{z}\left(1 - \frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k g_k(z)}{1+d_k g_k(z)}\right). \tag{D.2.6}$$

Proof. Indeed, using the trivial relation $\mathrm{Tr}\left[B(z)B(z)^{-1}\right] = N$, expanding $B(z)$, we obtain $\mathrm{Tr}\left[\frac{1}{N}\mathcal{O}K\mathcal{O}^T B(z)^{-1}\right] - z\mathrm{Tr}\left[B(z)^{-1}\right] = N$. Since $\mathcal{O}$ is an operator from $\mathcal{C}$ to $\mathbb{R}^N$, which is a finite dimensional space, we can apply the cyclic property of the trace and obtain $\mathrm{Tr}\left[\frac{1}{N}\mathcal{O}K\mathcal{O}^T B(z)^{-1}\right] = \mathrm{Tr}\left[A(z)\right]$. Thus,

$$\mathrm{Tr}\left[A(z)\right] - z\mathrm{Tr}\left[B(z)^{-1}\right] = N.$$

Dividing both sides by N and using Equation (D.2.5), we obtain

$$1 = \frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k g_k(z)}{1+d_k g_k(z)} - zm(z),$$

hence the result. □

Concentration of the Stieltjes Transform

We will now show that $g_k(z) = \frac{1}{N}\mathcal{O}_{\cdot k}^T B_{(k)}(z)^{-1}\mathcal{O}_{\cdot k}$ is close to $\frac{1}{N}\mathrm{Tr}\left(B_{(k)}(z)^{-1}\right)$, as suggested by the fact that by Wick's formula $\mathbb{E}[g_k] = \frac{1}{N}\mathrm{Tr}\left(\mathbb{E}\left[B_{(k)}(z)^{-1}\right]\right)$. Since $B(z)$ is obtained using a rank one permutation of $B_{(k)}(z)$, $\frac{1}{N}\mathrm{Tr}\left(B_{(k)}(z)^{-1}\right)$ is close to the Stieltjes transform m. As a result, all the g_k's are close to the Stieltjes transform m: it is natural to think that for $z \in \mathbb{H}_{<0}$, both $g_k(z)$'s and $m(z)$ should concentrate around the unique solution $\tilde{m}(z)$ in the cone spanned by 1 and $-1/z$ of the equation

$$\tilde{m}(z) = -\frac{1}{z}\left(1 - \frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k\tilde{m}(z)}{1+d_k\tilde{m}(z)}\right). \tag{D.2.7}$$

Remark D.3. The existence and the uniqueness of the solution in the cone spanned by 1 and $-1/z$ of the equation can be argued as follows. If in Equation (D.2.7) we truncate the series and consider the sum of the first M terms, one can show that there exists a unique fixed point $\tilde{m}_M(z)$ in the region R given by intersection between the cone spanned by 1 and $-1/z$ and the cone spanned by z and $1/z$ translated by $+1$ and multiplied by $-1/z$ (see Lemma C.6 in the Supplementary Material of [102]). Since R is a compact region, we can extract a converging subsequence that solves Equation (D.2.7), the limit of which can be showed to be unique, again using the same arguments of Lemma C.6 in the Supplementary Material of [102].

From now on we omit the z dependence and we set $m = m(z)$, $\tilde{m} = \tilde{m}(z)$ and $g_k(z) = g_k$.

Concentration bounds

Using Equation D.2.6 and the definition of the fixed point $\tilde{m}$ (Equation D.2.7), we obtain the following formula for the difference between the Stieltjes transform m and $\tilde{m}$:

$$\begin{aligned}\tilde{m} - m &= \frac{1}{z}\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k\left(\tilde{m}-g_k\right)}{(1+d_k\tilde{m})(1+d_k g_k)}\\ &= \frac{\tilde{m}-m}{z}\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k}{(1+d_k\tilde{m})(1+d_k g_k)} + \frac{1}{z}\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k\left(m-g_k\right)}{(1+d_k\tilde{m})(1+d_k g_k)},\end{aligned}$$

where the well-posedness of the two infinite sums of the r.h.s is granted by the fact that:

1. $\left|\frac{d_k}{(1+d_k\tilde{m})(1+d_k g_k)}\right| \leq d_k$ since $\Re\left(\tilde{m}\right), \Re(g_k)$ are positive, thus the first sum is absolutely convergent,
2. being the difference of two absolutely convergent series, the second sum is also absolutely convergent.

As a consequence, the difference $\tilde{m} - m$ can be expressed as

$$\tilde{m} - m = \frac{\frac{1}{N}\sum_{k=1}^{\infty} \frac{d_k(m-g_k)}{(1+d_k\tilde{m})(1+d_k g_k)}}{z - \frac{1}{N}\sum_{k=1}^{\infty} \frac{d_k}{(1+d_k\tilde{m})(1+d_k g_k)}}, \tag{D.2.8}$$

which allows us to show the concentration of m around $\tilde{m}$ from the concentration of g_k around m.

Regarding the concentration of the g_k's around m, we have the following result:

Lemma D.2. *For any $N, s \in \mathbb{N}$ and any $z \in \mathbb{H}_{<0}$, we have*

$$\begin{aligned} \mathbb{E}\left[\left|m - g_k\right|^{2s}\right] &\leq \frac{\boldsymbol{c}_s}{|z|^{2s} N^s}, \\ \mathbb{E}\left[\left|m - m_{(k)}\right|^{2s}\right] &\leq \frac{1}{|z|^{2s} N^{2s}}. \end{aligned}$$

where $\boldsymbol{c}_s$ only depends on s.

Proof. The second inequality will be proven while proving the first one. Let $m_{(k)} = \frac{1}{N}\mathrm{Tr}\left[B_{(k)}^{-1}\right]$ where $B_{(k)}$ was defined in Section D.2. By convexity:

$$\mathbb{E}\left[\left|m - g_k\right|^{2s}\right] \leq 2^{2s-1}\mathbb{E}\left[\left|m - m_{(k)}\right|^{2s}\right] + 2^{2s-1}\mathbb{E}\left[\left|m_{(k)} - g_k\right|^{2s}\right]. \tag{D.2.9}$$

Bound on $\mathbb{E}[|m-m_{(k)}|^{2s}]$: We obtain the bound on the expectation by showing that a deterministic bound holds for the random variable $|m - m_{(k)}|^{2s}$. Using the Sherman-Morrison formula (Equation (D.2.4)), and using the cyclic property of the trace,

$$m = m_{(k)} - \frac{1}{N}\frac{d_k g_k'}{1 + d_k g_k}$$

since the derivative $g_k'(z)$ of $g_k(z)$ is equal to $\frac{1}{N}\mathcal{O}_{\cdot k}^T B(z)^{-2}\mathcal{O}_{\cdot k}$. As a result, we obtain $\left|m - m_{(k)}\right|^{2s} = \frac{1}{N^{2s}}\frac{d_k^{2s}\left|g_k'\right|^{2s}}{|1+d_k g_k|^{2s}}$. Using the fact that $|1 + d_k g_k| \geq |d_k g_k|$ since $\Re\left(g_k\right) \geq 0$,

$$\left|m - m_{(k)}\right|^{2s} \leq \frac{1}{N^{2s}}\frac{\left|g_k'\right|^{2s}}{\left|g_k\right|^{2s}}.$$

Notice now that

$$\left|\frac{g_k'}{g_k}\right| = \left|\frac{\mathcal{O}_{\cdot k}^T B_{(k)}(z)^{-2}\mathcal{O}_{\cdot k}}{\mathcal{O}_{\cdot k}^T B_{(k)}(z)^{-1}\mathcal{O}_{\cdot k}}\right| \leq \max_{w\in\mathbb{R}^N}\left|\frac{w^T B_{(k)}(z)^{-2} w}{w^T B_{(k)}(z)^{-1} w}\right| \leq \left\|B_{(k)}(z)^{-1}\right\|_{\mathrm{op}}.$$

The eigenvalues of $B_{(k)}(z)^{-1}$ are given by $\frac{1}{\lambda_i - z}$ where the $\lambda_i > 0$ are the eigenvalues of the symmetric matrix $\frac{1}{N}\mathcal{O}K_{(k)}\mathcal{O}^T$: $\left\|B_{(k)}(z)^{-1}\right\|_{\mathrm{op}} \leq \max_i \frac{1}{|\lambda_i - z|}$ is also bounded by $\frac{1}{|z|}$ if $z \in \mathbb{H}_{<0}$. Thus we get

$$\left|m - m_{(k)}\right|^{2s} \leq \frac{1}{|z|^{2s} N^{2s}}.$$

Bound on $\mathbb{E}[|m_{(k)} - g_k|^{2s}]$: The term $\mathbb{E}\left[\left(\left(m_{(k)} - g_k\right)\overline{\left(m_{(k)} - g_k\right)}\right)^s\right]$ is equal to

$$\mathbb{E}\left[\left(\left(\frac{1}{N}\mathrm{Tr}\left[B_{(k)}^{-1}\right] - \frac{1}{N}\mathcal{O}_{\cdot k}^T B_{(k)}^{-1}\mathcal{O}_{\cdot k}\right)\left(\frac{1}{N}\mathrm{Tr}\left[\overline{B_{(k)}^{-1}}\right] - \frac{1}{N}\mathcal{O}_{\cdot k}^T \overline{B_{(k)}^{-1}}\mathcal{O}_{\cdot k}\right)\right)^s\right].$$

Let $\boldsymbol{B} = \left(B_{(k)}, \overline{B_{(k)}}, \ldots, B_{(k)}, \overline{B_{(k)}}\right)$ and let us denote by $\boldsymbol{B}(i)$ the i^{th} element of $\boldsymbol{B}$. Using Wick's formula (Lemma D.9), we have

$$\mathbb{E}\left[\left|m_{(k)} - g_k\right|^{2s}\right] = \frac{1}{N^s}\sum_{\sigma\in\mathfrak{S}_{2s}^\dagger}\frac{1}{N^{s-\boldsymbol{c}(\sigma)}}2^{2s-\boldsymbol{c}(\sigma)}\mathbb{E}\left[\prod_{c \text{ cycle of } \sigma}\frac{1}{N}\mathrm{Tr}\left[\prod_{i\in c}\boldsymbol{B}(i)\right]\right],$$

where we recall that $\mathfrak{S}_{2s}^\dagger$ is the set of permutations with no fixed points and the product over i is taken according to the order given by the cycle c and does not depend on the starting point. Using the fact that the eigenvalues of $B_{(k)}$ are of the form $1/(\lambda_i - z)$ with $\lambda_i \geq 0$,

$$\left|\frac{1}{N}\mathrm{Tr}\left[\prod_{i\in c}\boldsymbol{B}(i)\right]\right| \leq \frac{1}{|z|^{\#c}}.$$

Hence,

$$\mathbb{E}\left[\left|m_{(k)} - g_k\right|^{2s}\right] \leq \frac{1}{N^s}\frac{1}{|z|^{2s}}\sum_{\sigma\in\mathfrak{S}_{2s}^\dagger}\frac{2^{2s-\boldsymbol{c}(\sigma)}}{N^{s-\boldsymbol{c}(\sigma)}}.$$

Note that, since $\sigma \in \mathfrak{S}_{2s}^\dagger$, it has no fixed point, hence $\boldsymbol{c}(\sigma) \leq s$ and thus $K_s := \sup_N \sum_{\sigma\in\mathfrak{S}_{2s}^\dagger}\frac{2^{2s-\boldsymbol{c}(\sigma)}}{N^{s-\boldsymbol{c}(\sigma)}}$ is finite. This yields the inequality

$$\mathbb{E}\left[\left|m_{(k)} - g_k\right|^{2s}\right] \leq \frac{K_s}{|z|^{2s}N^s}.$$

Using the two bounds on $\mathbb{E}\left[\left|m - m_{(k)}\right|^{2s}\right]$ and $\mathbb{E}\left[\left|m_{(k)} - g_k\right|^{2s}\right]$ in Equation (D.2.9), we get

$$\mathbb{E}\left[\left|m - g_k\right|^{2s}\right] \leq \frac{\boldsymbol{c}_s}{|z|^{2s}N^s},$$

where $\boldsymbol{c}_s = 2^{2s-1}\left[1 + K_s\right]$. □

As a result, we can show the concentration of the Stieltjes transform m and of the g_k's around the fixed point $\tilde{m}$:

Proposition D.1. *For any $N, s \in \mathbb{N}$, and any $z \in \mathbb{H}_{<0}$, we have*

$$\mathbb{E}\left[\left|\tilde{m} - m\right|^{2s}\right] \leq \frac{\boldsymbol{c}_s\left(\mathrm{Tr}[T_K]\right)^{2s}}{|z|^{4s}N^{3s}},$$
$$\mathbb{E}\left[\left|\tilde{m} - g_k\right|^{2s}\right] \leq \frac{2^{2s-1}\boldsymbol{c}_s\left(\mathrm{Tr}[T_K]\right)^{2s}}{|z|^{4s}N^{3s}} + \frac{2^{2s-1}\boldsymbol{c}_s}{|z|^{2s}N^s}.$$

where $\boldsymbol{c}_s$ is the same constant as in Lemma D.2.

Proof. The second bound is a direct consequence of the first one, Lemma D.2 and convexity. It remains to prove the first bound. Recall Equation (D.2.8)

$$\tilde{m} - m = \frac{\frac{1}{N}\sum_{k=1}^{\infty} \frac{d_k(m-g_k)}{(1+d_k\tilde{m})(1+d_k g_k)}}{z - \frac{1}{N}\sum_{k=1}^{\infty} \frac{d_k}{(1+d_k\tilde{m})(1+d_k g_k)}}.$$

We first bound from below the norm of the denominator using Lemma D.12: since $\tilde{m}$ and g_k all lie in the cone spanned by 1 and $-1/z$ we have

$$\left| z - \frac{1}{N}\sum_{k=1}^{\infty} \frac{d_k}{(1+d_k\tilde{m})(1+d_k g_k)} \right| \geq |z|.$$

Using this bound, we can bound from below $\mathbb{E}\left[|\tilde{m} - m|^{2s}\right]$ by:

$$\frac{1}{|z|^{2s} N^{2s}} \sum_{k_1,\ldots,k_{2s}=1}^{\infty} \frac{d_{k_1}\cdots d_{k_{2s}}}{|1+d_{k_1}\tilde{m}|\cdots|1+d_{k_{2s}}\tilde{m}|} \mathbb{E}\left[|m-g_{k_1}|\cdots|m-g_{k_{2s}}|\right],$$

and hence, using a generalization of Cauchy-Schwarz inequality (Lemma D.11), by:

$$\frac{1}{|z|^{2s} N^{2s}} \sum_{k_1,\ldots,k_{2s}=1}^{\infty} \frac{d_{k_1}\cdots d_{k_{2s}}}{|1+d_{k_1}\tilde{m}|\cdots|1+d_{k_{2s}}\tilde{m}|} \left(\mathbb{E}\left[|m-g_{k_1}|^{2s}\right]\cdots\mathbb{E}\left[|m-g_{k_{2s}}|^{2s}\right]\right)^{\frac{1}{2s}}.$$

Using the fact that $\Re(\tilde{m}) \geq 0$ and hence $|1+d_{k_1}\tilde{m}| \geq 1$, and using Lemma D.2, this gives the following upper bound:

$$\mathbb{E}\left[|m-g_k|^{2s}\right] \leq \frac{\boldsymbol{c}_s}{|z|^{4s} N^{3s}} \left(\mathrm{Tr}[T_K]\right)^{2s}.$$

□

We now give tighter bounds for $|\tilde{m} - \mathbb{E}[m]|$ and $|\tilde{m} - \mathbb{E}[g_k]|$:

Proposition D.2. *For any $N \in \mathbb{N}$ and any $z \in \mathbb{H}_{<0}$, we have*

$$\begin{aligned}
|\tilde{m} - \mathbb{E}[m]| &\leq \frac{\mathrm{Tr}[T_K]}{|z|^2 N^2} + \frac{2\boldsymbol{c}_1 \left(\mathrm{Tr}[T_K]\right)^2}{|z|^3 N^2} + \frac{2\boldsymbol{c}_1 \left(\mathrm{Tr}[T_K]\right)^4}{|z|^5 N^4}, \\
|\tilde{m} - \mathbb{E}[g_k]| &\leq \frac{1}{|z| N} + \frac{\mathrm{Tr}[T_K]}{|z|^2 N^2} + \frac{2\boldsymbol{c}_1 \left(\mathrm{Tr}[T_K]\right)^2}{|z|^3 N^2} + \frac{2\boldsymbol{c}_1 \left(\mathrm{Tr}[T_K]\right)^4}{|z|^5 N^4},
\end{aligned}$$

where $\boldsymbol{c}_1$ is the constant in Lemma D.2.

Proof. **First bound:** Following similar ideas to the one which provided Equation (D.2.8), notice that

$$\tilde{m} - m = \frac{1}{z}\frac{1}{N}\sum_{k=1}^{\infty} \frac{d_k(\tilde{m} - g_k)}{(1+d_k\tilde{m})(1+d_k g_k)}$$

$$= \frac{1}{z}\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k\left(\tilde{m}-g_k\right)}{(1+d_k\tilde{m})^2} + \frac{1}{z}\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k^2\left(\tilde{m}-g_k\right)^2}{(1+d_k\tilde{m})^2(1+d_kg_k)}$$
$$= \frac{\tilde{m}-m}{z}\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k}{(1+d_k\tilde{m})^2} + \frac{1}{z}\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k\left(m-g_k\right)}{(1+d_k\tilde{m})^2} + \frac{1}{z}\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k^2\left(\tilde{m}-g_k\right)^2}{(1+d_k\tilde{m})^2(1+d_kg_k)},$$

hence the new identity:

$$\tilde{m}-m = \frac{\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k(m-g_k)}{(1+d_k\tilde{m})^2} + \frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k^2(\tilde{m}-g_k)^2}{(1+d_k\tilde{m})^2(1+d_kg_k)}}{z-\frac{1}{N}\sum_{k=1}^{\infty}\frac{d_k}{(1+d_k\tilde{m})^2}}.$$

Again, using Lemma D.12, the norm of the denominator is bounded from below by $|z|$. From Lemma D.9, $\mathbb{E}\left[g_k\right] = \mathbb{E}\left[m_{(k)}\right]$, and thus from Lemma D.2, $\left|\mathbb{E}\left[m-g_k\right]\right| \leq \mathbb{E}\left[\left|m-m_{(k)}\right|\right] \leq \frac{1}{|z|N}$. Furthermore, from Proposition D.1, $\mathbb{E}\left[\left|g_k-\tilde{m}\right|^2\right] \leq \frac{2\boldsymbol{c}_1(\mathrm{Tr}[T_K])^2}{|z|^4N^3} + \frac{2\boldsymbol{c}_1}{|z|^2N}$. Thus, the expectation of the numerator is bounded by

$$\frac{1}{|z|\,N^2}\sum_{k=1}^{\infty}\frac{d_k}{\left|1+d_k\tilde{m}\right|^2} + \left(\frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^4\,N^4} + \frac{2\boldsymbol{c}_1}{|z|^2\,N^2}\right)\sum_{k=1}^{\infty}\frac{d_k^2}{\left|1+d_k\tilde{m}\right|^2}.$$

Hence, using again the inequality $|1+d_k\tilde{m}| \geq 1$, it is bounded by

$$\frac{\mathrm{Tr}[T_K]}{|z|\,N^2} + \frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^2\,N^2} + \frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^4}{|z|^4\,N^4}.$$

This allows us to conclude that

$$\left|\tilde{m}-\mathbb{E}\left[m\right]\right| \leq \frac{\mathrm{Tr}[T_K]}{|z|^2\,N^2} + \frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^3\,N^2} + \frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^4}{|z|^5\,N^4}.$$

Second bound: Since $\mathbb{E}\left[g_k\right] = \mathbb{E}\left[m_{(k)}\right]$, one has

$$\left|\tilde{m}-\mathbb{E}\left[g_k\right]\right| \leq \left|\tilde{m}-\tilde{m}_{(k)}\right| + \left|\tilde{m}_{(k)}-\mathbb{E}\left[m_{(k)}\right]\right|,$$

where $\tilde{m}_{(k)}$ is the unique solution in the cone spanned by 1 and $-1/z$ to the equation

$$\tilde{m}_{(k)} = -\frac{1}{z}\left(1-\frac{1}{N}\sum_{m\neq k}^{\infty}\frac{d_m\tilde{m}_{(k)}}{1+d_m\tilde{m}_{(k)}}\right).$$

From Lemma D.13, $\left|\tilde{m}-\tilde{m}_{(k)}\right| \leq \frac{1}{|z|N}$. The second term $\left|\tilde{m}_{(k)}-\mathbb{E}\left[m_{(k)}\right]\right|$ is bounded by applying the first bound of this proposition to the Stieltjes transform $m_{(k)}$. As a result, we obtain

$$\left|\tilde{m}-\mathbb{E}\left[g_k\right]\right| \leq \frac{1}{|z|\,N} + \frac{\mathrm{Tr}[T_K]}{|z|^2\,N^2} + \frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^3\,N^2} + \frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^4}{|z|^5\,N^4}.$$

□

Properties of the effective dimension and SCT

General properties

We begin with general properties on the Signal Capture Threshold ϑ (which depends on λ, N and on the eigenvalues d_k of T_K), valid for any kernel K.

Proposition D.3. *For any $\lambda > 0$, we have*

$$\lambda < \vartheta(\lambda, N) \leq \lambda + \frac{1}{N}\mathrm{Tr}[T_K], \qquad 1 \leq \partial_\lambda \vartheta(\lambda, N) \leq \frac{1}{\lambda}\vartheta(\lambda, N),$$

moreover $\vartheta(\lambda, N)$ is decreasing as a function of N and $\partial_\lambda \vartheta(\lambda, N)$ is decreasing as a function of λ.

Proof. Let $\lambda > 0$.

1. Recall that $\vartheta(\lambda)$ is the unique positive real number such that

$$\vartheta(\lambda) = \lambda + \frac{\vartheta(\lambda)}{N}\mathrm{Tr}\left[T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1}\right].$$

   Since T_K is a positive operator, $\mathrm{Tr}\left[T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1}\right] \geq 0$ and thus $\vartheta(\lambda) \geq \lambda$. Moreover, $T_K + \vartheta(\lambda) I_{\mathcal{C}} \geq \vartheta(\lambda) I_{\mathcal{C}}$, thus

$$T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1} \leq \frac{T_K}{\vartheta(\lambda)}$$

   and thus $\vartheta(\lambda) \leq \lambda + \frac{1}{N}\mathrm{Tr}\left[T_K\right]$, which gives the desired inequality.

2. Differentiating Equation (D.2.2), the derivative $\partial_\lambda \vartheta(\lambda)$ is given by:

$$\partial_\lambda \vartheta(\lambda) = \frac{1}{\left(1 - \frac{1}{N}\mathrm{Tr}\left[\left(T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1}\right)^2\right]\right)}. \tag{D.2.10}$$

   Using the fact that $T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1} \leq I_{\mathcal{C}}$, one has

$$\left(T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1}\right)^2 \leq T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1},$$

   thus $0 \leq \frac{1}{N}\mathrm{Tr}\left[\left(T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1}\right)^2\right] \leq \frac{1}{N}\mathrm{Tr}\left[T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1}\right]$. Using Equation (D.2.2), $\frac{1}{N}\mathrm{Tr}\left[T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1}\right] = 1 - \frac{\lambda}{\vartheta(\lambda)}$. This yields

$$0 \leq \frac{\lambda}{\vartheta(\lambda)} \leq 1 - \frac{1}{N}\mathrm{Tr}\left[\left(T_K \left(T_K + \vartheta(\lambda) I_{\mathcal{C}}\right)^{-1}\right)^2\right] \leq 1.$$

   Inverting this inequality yields the desired inequalities.

3. In order to study the variation of $\vartheta(\lambda, N)$ as a function of N, we take the derivatives of Equation (D.2.2) w.r.t λ and N, and notice that

$$\partial_N \vartheta(\lambda, N) = \frac{1}{N}(\lambda - \vartheta)\partial_\lambda \vartheta(\lambda, N).$$

   In particular, since $\vartheta > \lambda$ and $\partial_\lambda \vartheta \geq 1$, we get that $\partial_N \vartheta(\lambda, N) < 0$ hence $\vartheta(\lambda, N)$ is decreasing as a function of N.

4. Finally, we conclude by noting that since $\partial_\lambda \vartheta(\lambda, N) > 0$, $\vartheta(\lambda, N)$ is an increasing function of λ and thus, from the Equation (D.2.10) we have that $\partial_\lambda \vartheta(\lambda, N)$ is decreasing as a function of ϑ and thus as a function of λ.

□

Bounds under polynomial decay hypothesis

In this subsection only, we assume that $d_k = \Theta(k^{-\beta})$ with $\beta > 1$, i.e, there exist c_ℓ and c_h positive such that for any $k \geq 1$, $c_\ell k^{-\beta} \leq d_k \leq c_h k^{-\beta}$. We first study the asymptotic behavior of $\vartheta(0, N)$ and $\partial_\lambda \vartheta(0, N)$ as N goes to infinity, then using these results, we investigate the asymptotic behavior of $\vartheta(\lambda, N)$ and $\partial_\lambda \vartheta(\lambda, N)$ as N goes to infinity.

For any $t \in \mathbb{R}^+$, let $\mathcal{N}(t)$ denote the t-effective dimension [238, 29] defined by

$$\mathcal{N}(t) := \sum_{k=1}^{\infty} \frac{d_k}{t + d_k}.$$

For any $\lambda > 0$, the SCT is the unique solution of $\vartheta(\lambda, N) = \lambda + \frac{\vartheta(\lambda, N)}{N} \mathcal{N}(\vartheta(\lambda, N))$. In particular, $\vartheta(0, N)$ is the unique solution of $\mathcal{N}(\vartheta(0, N)) = N$.

Since $\mathcal{N}(t)$ is decreasing from ∞ to 0, in order to study the asymptotic behavior of $\vartheta(0, N)$ as N goes to infinity, one has to understand the rate of explosion of $\mathcal{N}(t)$ as t goes to zero, as given by the following Lemma (also found in [9, 239]):

Lemma D.3. *If $d_k = \Theta(k^{-\beta})$ with $\beta > 1$, then $\mathcal{N}(t) = \Theta(t^{-\frac{1}{\beta}})$ when $t \to 0$.*

Proof. For any $m \in \mathbb{R}_+$, $\mathcal{N}(t) = \sum_{k \leq m} \frac{d_k}{t+d_k} + \sum_{k>m}^{\infty} \frac{d_k}{t+d_k} \leq m + t^{-1} \sum_{k>m} d_k$. Then there exists $c, d > 0$ such that $\sum_{k>m} d_k \leq c \sum_{k>m} k^{-\beta} \leq d m^{1-\beta}$. Thus $\mathcal{N}(t)$ is bounded by $m + dt^{-1} m^{1-\beta}$ for any m. Taking $m = t^{-1} m^{1-\beta}$, i.e. $m = t^{-\frac{1}{\beta}}$, one gets that $\mathcal{N}(t) \leq C t^{-\frac{1}{\beta}}$.

For the lower bound, notice that $\mathcal{N}(t) \geq \sum_{k | d_k \geq t} \frac{d_k}{t+d_k} \geq \frac{1}{2} \# \{k \mid d_k \geq t\}$. Using the fact that there exists $c_\ell > 0$ such that $d_k \geq c_\ell k^{-\beta}$, $\# \{k \mid d_k \geq t\} \geq \# \left\{k \mid c_\ell k^{-\beta} \geq t\right\} = \left\lfloor (t/c_\ell)^{-\frac{1}{\beta}} \right\rfloor$. This yields the lower bound on $\mathcal{N}(t)$. □

Lemma D.4. *If $d_k = \Theta(k^{-\beta})$ with $\beta > 1$, then $\vartheta(0, N) = \Theta\left(N^{-\beta}\right)$.*

Proof. From the previous lemma, there exist $b_\ell, b_h > 0$ such that $b_\ell \vartheta(0, N)^{-\frac{1}{\beta}} \leq \mathcal{N}(\vartheta(0, N)) \leq b_h \vartheta(0, N)^{-\frac{1}{\beta}}$. From the definition of $\vartheta(0, N)$, $\mathcal{N}(\vartheta(0, N)) = N$, thus we get $\left(N/b_\ell\right)^{-\beta} \leq \vartheta(0, N) \leq \left(N/b_h\right)^{-\beta}$. □

With no assumption on the spectrum of T_K, the upper bound for the derivative of the SCT $\partial_\lambda \vartheta$ obtained in Proposition D.3, becomes useless in the ridgeless limit $\lambda \to 0$. Yet, with the assumption of power-law decay of the eigenvalues of T_K we can refine the bound with a meaningful one. In order to obtain this we first prove a technical lemma.

Lemma D.5. *If $d_k = \Theta(k^{-\beta})$ with $\beta > 1$, then $\sup_N \partial_\lambda \vartheta(0, N) < \infty$.*

Proof. The derivative of the SCT with respect to λ at $\lambda = 0$ is given by:

$$\partial_\lambda \vartheta(0, N) = \frac{N}{\vartheta(0, N) \sum_{k=1}^{\infty} \frac{d_k}{(\vartheta(0,N)+d_k)^2}}.$$

Set $\alpha > 1$, then for all $d_k \in [\alpha^{-1}t, \alpha t]$, we have that $\frac{d_k}{(t+d_k)^2} \geq \frac{\alpha t}{(t+\alpha t)^2} = \frac{\alpha}{(1+\alpha)^2}\frac{1}{t}$. Thus,

$$t \sum_{k=1}^{\infty} \frac{d_k}{(t+d_k)^2} \geq t \sum_{\alpha^{-1}t < d_k < \alpha t} \frac{d_k}{(t+d_k)^2} \geq \frac{\alpha}{(1+\alpha)^2} \#\{k \mid \alpha^{-1}t < d_k < \alpha t\}.$$

It follows that

$$\partial_\lambda \vartheta(0, N) \leq N \frac{(1+\alpha)^2}{\alpha} \frac{1}{\#\{k \mid \alpha^{-1}\vartheta(0,N) < d_k < \alpha\vartheta(0,N)\}}$$

Now, using Lemma D.4, we are going to find a value of α such that $\#\{k \mid \alpha^{-1}\vartheta(0,N) < d_k < \alpha\vartheta(0,N)\} \geq cN$ for some universal constant c: this will conclude the proof.

By using the assumption that there exist $c_\ell, c_h > 0$ such that $c_\ell k^{-\beta} \leq d_k \leq c_h k^{-\beta}$, in Lemma D.4 we saw that there exist $c'_\ell, c'_h > 0$ such that $c'_\ell N^{-\beta} \leq \vartheta(0,N) \leq c'_h N^{-\beta}$. For sake of simplicity, let us assume that the ratios $\frac{c_\ell}{c'_\ell}$ and $\frac{c_h}{c'_h}$ are not integer. Hence we have

$$\begin{aligned}
\#\{k \mid \alpha^{-1}\vartheta(0,N) \leq d_k \leq \alpha\vartheta(0,N)\} &\geq \#\left\{k \mid \frac{1}{\alpha c_\ell}\vartheta(0,N) \leq k^{-\beta} \leq \frac{\alpha}{c_h}\vartheta(0,N)\right\} \\
&\geq \#\left\{k \mid \frac{1}{\alpha c_\ell} c'_h N^{-\beta} \leq k^{-\beta} \leq \frac{\alpha}{c_h} c'_\ell N^{-\beta}\right\} \\
&= \left(\left\lfloor \left(\frac{\alpha c_\ell}{c'_h}\right)^{\frac{1}{\beta}} \right\rfloor - \left\lfloor \left(\frac{c_h}{\alpha c'_\ell}\right)^{\frac{1}{\beta}} \right\rfloor\right) N
\end{aligned}$$

For one of the two values $\alpha \in \{\frac{c_h}{c_\ell}, \alpha = \frac{c'_h}{c'_\ell}\}$, we have a meaningful (positive) bound:

$$\#\left\{k \mid \alpha^{-1}\vartheta(0,N) \leq d_k \leq \alpha\vartheta(0,N)\right\} \geq \left|\left\lfloor \left(\frac{c_h}{c'_h}\right)^{\frac{1}{\beta}} \right\rfloor - \left\lfloor \left(\frac{c_\ell}{c'_\ell}\right)^{\frac{1}{\beta}} \right\rfloor\right| N.$$

This allows us to conclude. □

Proposition D.4. *If there exist $\beta > 1$ and $c_\ell, c_h > 0$ s.t. for any $k \in \mathbb{N}$, $c_\ell k^{-\beta} \leq d_k \leq c_h k^{-\beta}$, then for any integer N,*

1. $\lambda + a_\ell N^{-\beta} \leq \vartheta(\lambda, N) \leq c\lambda + a_h N^{-\beta}$,
2. $1 \leq \partial_\lambda \vartheta(\lambda, N) \leq c$,

where $a_\ell, a_h \geq 0$ and $c \geq 1$ depend only on c_ℓ, c_h, β.

Proof. We start by proving the inequalities for the derivative of the SCT $\partial_\lambda \vartheta(\lambda, N)$. The left side of the inequality has already been proven in Proposition D.3. For the right side, from Proposition D.3, the derivative $\partial_\lambda \vartheta(\lambda, N)$ is decreasing in λ. In particular, by Lemma D.5, $\partial_\lambda \vartheta(\lambda, N) \leq \sup_N \partial_\lambda \vartheta(0, N) < \infty$. Thus, the right side holds with $c := \sup_N \partial_\lambda \vartheta(0, N)$.

The inequality for the SCT $\vartheta(\lambda, N)$ is then obtained by integrating the second inequality and by using the initial value condition $a_\ell N^{-\beta} \leq \vartheta(0, N) \leq a_h N^{-\beta}$ provided by Lemma D.4. □

The Operator $A(z)$

We have now the tools to describe the moments of the operator $A(z)$ which allow us to describe the moments of the predictor $\hat{f}_\lambda$.

Expectation

Writing $\tilde{A}_{\vartheta(-z)} = T_K \left(T_K + \vartheta(-z) I_{\mathcal{C}}\right)^{-1}$ and for any diagonalizable operator A writing $|A|$ for the operator with the same eigenfunctions but with eigenvalues replaced by their absolute values, we have:

Theorem D.1. *For any $z \in \mathbb{H}_{<0}$, for any $f, g \in \mathcal{C}$, we have*

$$\left|\left\langle f, \left(\mathbb{E}\left[A(z)\right] - \tilde{A}_{\vartheta(-z)}\right) g\right\rangle_S\right| \leq \left|\left\langle f, |\tilde{A}_{\vartheta(-z)}| |I_{\mathcal{C}} - \tilde{A}_{\vartheta(-z)}| g\right\rangle_S\right| \left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[K]}{|z|N}\right)\right) \tag{D.2.11}$$

using the big-P notation of Definition 7.

Remark D.4. Note that in particular since the polynomial implicitly embedded in $\mathcal{P}$ vanishes at 0, the right hand side tends to 0 as $N \to \infty$.

Proof. As before, let $(f^{(k)})_{k\in\mathbb{N}}$ be the orthonormal basis of $\mathcal{C}$ defined above and $A_{k\ell}(z) = \langle f^{(k)}, A(z) f^{(\ell)}\rangle_S$. Using a symmetry argument, we first show that for any $\ell \neq k$, $\mathbb{E}\left[A_{\ell k}(z)\right] = 0$: this implies that $\mathbb{E}\left[A(z)\right]$ and $\tilde{A}_{\vartheta(-z)}$ have the same eigenfunctions $f^{(k)}$. Thus, to conclude the proof, we only need to prove Equation D.2.11 for $f = g = f^{(k)}$.

- **Off-Diagonal terms:** By a symmetry argument, we show that the off-diagonal terms are null. Consider the map $s_k : \mathcal{C} \to \mathcal{C}$ defined by $s_k : f \mapsto f - 2\left\langle f, f^{(k)}\right\rangle_S f^{(k)}$, and note that $s_k(f^{(m)}) = f^{(m)}$ if $m \neq k$ and $s_k(f^{(k)}) = -f^{(k)}$. The map s_k is a symmetry for the observations, i.e. for any observations $o_1, \ldots, o_N$, and any functions $f_1, \ldots, f_N$, the vector $(o_i(s_k(f_i))_{i=1,\ldots,N}$ and $(o_i(f_i))_{i=1,\ldots,N}$ have the same law. Thus, the sampling operator $\mathcal{O}$ and the operator $\mathcal{O}s_k$ have the same law, hence so do $A(z)$ and $A^{s_k}(z)$, where

$$A^{s_k}(z) := \frac{1}{N} K s_k^T \mathcal{O}^T \left(\frac{1}{N} \mathcal{O} s_k K s_k^T \mathcal{O}^T - z I_N\right)^{-1} \mathcal{O} s_k.$$

  Note that $K s_k^T = s_k K$ and since $s_k^2 = \mathrm{Id}$, $s_k K s_k^T = K$. This implies that $A^{s_k}(z) = s_k A(z) s_k$. For any $\ell \neq k$, $A^{s_k}_{\ell k}(z) = -A_{\ell k}(z)$, hence $\mathbb{E}[A_{\ell k}(z)] = 0$.

- **Diagonal terms:** Using Equation D.2.5, we have

$$\begin{aligned} A_{kk}(z) = \frac{d_k g_k}{1 + d_k g_k} &= \frac{d_k \tilde{m}}{1 + d_k \tilde{m}} + \frac{d_k (g_k - \tilde{m})}{(1 + d_k \tilde{m})(1 + d_k g_k)} \\ &= \frac{d_k \tilde{m}}{1 + d_k \tilde{m}} + \frac{d_k (g_k - \tilde{m})}{(1 + d_k \tilde{m})^2} - \frac{d_k^2 (g_k - \tilde{m})^2}{(1 + d_k \tilde{m})^2 (1 + d_k g_k)}. \end{aligned}$$

  From this, using the fact that $\Re(g_k) > 0$, we obtain

$$\left|\mathbb{E}\left[A_{kk}(z)\right] - \frac{d_k \tilde{m}}{1 + d_k \tilde{m}}\right| \leq \frac{d_k \left|\mathbb{E}\left[g_k\right] - \tilde{m}\right|}{\left|1 + d_k \tilde{m}\right|^2} + \frac{d_k^2 \mathbb{E}\left[\left|g_k - \tilde{m}\right|^2\right]}{\left|1 + d_k \tilde{m}\right|^2}.$$

Using Proposition D.2, we can bound the first fraction by

$$\begin{aligned}\frac{d_k\left|\mathbb{E}\left[g_k\right]-\tilde{m}\right|}{\left|1+d_k\tilde{m}\right|^2} &\leq \frac{d_k}{\left|1+d_k\tilde{m}\right|^2}\left(\frac{1}{|z|\,N}+\frac{\mathrm{Tr}[T_K]}{|z|^2\,N^2}+\frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^3\,N^2}+\frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^4}{|z|^5\,N^4}\right)\\ &\leq \frac{d_k|\vartheta(-z)|^2}{\left|\vartheta(-z)+d_k\right|^2}\left(\frac{1}{|z|\,N}+\frac{\mathrm{Tr}[T_K]}{|z|^2\,N}+\frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^3\,N^2}+\frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^4}{|z|^5\,N^4}\right)\\ &\leq \frac{d_k}{\left|\vartheta(-z)+d_k\right|}\left|1-\frac{d_k}{\vartheta(-z)+d_k}\right|\left(\frac{1}{N}+\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|\,N}\right)\right),\end{aligned}$$

by substituting $\vartheta(-z)=\frac{1}{\tilde{m}(z)}$, using the bound $|\vartheta(-z)|\leq|z|+\frac{\mathrm{Tr}[T_K]}{N}$ (see Proposition D.3).

Using Proposition D.1, the inequality $d_k^2\leq d_k\mathrm{Tr}[T_K]$ and similar arguments as above, we can bound the second fraction by

$$\begin{aligned}\frac{d_k^2\mathbb{E}\left[\left|g_k-\tilde{m}\right|^2\right]}{\left|1+d_k\tilde{m}\right|^2} &\leq \frac{d_k^2}{\left|1+d_k\tilde{m}\right|^2}\left(\frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^4\,N^3}+\frac{2\boldsymbol{c}_1}{|z|^2\,N}\right)\\ &\leq \frac{d_k|\vartheta(-z)|^2}{\left|\vartheta(-z)+d_k\right|^2}\left(\frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^3}{|z|^4\,N^3}+\frac{2\boldsymbol{c}_1\mathrm{Tr}[T_K]}{|z|^2\,N}\right)\\ &\leq \frac{d_k}{\left|\vartheta(-z)+d_k\right|}\left|1-\frac{d_k}{\vartheta(-z)+d_k}\right|\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|\,N}\right)\end{aligned}$$

Finally, putting everything together, we get:

$$\left|\mathbb{E}\left[A_{kk}(z)\right]-\frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right|\leq\frac{d_k}{\left|\vartheta(-z)+d_k\right|}\left|1-\frac{d_k}{\vartheta(-z)+d_k}\right|\left(\frac{1}{N}+\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|\,N}\right)\right)\tag{D.2.12}$$

□

Variance

To study the variance of $A(z)$ we will need to apply the Shermann-Morrison formula twice, to isolate the contribution of the two eigenfunctions $f^{(k)}$ and $f^{(\ell)}$. Similarly to above, we set $K_{(k\ell)}=\sum_{n\notin\{k,\ell\}}d_nf^{(n)}\otimes f^{(n)}$ and we define

$$B_{(k\ell)}(z)=\frac{1}{N}\mathcal{O}K_{(k\ell)}\mathcal{O}^T-zI_N,\qquad m_{(k\ell)}(z)=\frac{1}{N}\mathrm{Tr}\left[B_{(k\ell)}(z)^{-1}\right].$$

Note that the concentration results of Section D.2 apply to $m_{(k\ell)}$: it concentrates around $\tilde{m}_{(k\ell)}$, the unique solution, in the cone spanned by 1 and $-1/z$, to the equation

$$\tilde{m}_{(k\ell)}=-\frac{1}{z}\left(1-\frac{\tilde{m}_{(k\ell)}}{N}\mathrm{Tr}\left[T_{K_{(k\ell)}}\left(T_{K_{(k\ell)}}+\tilde{m}_{(k\ell)}I_{\mathcal{C}}\right)^{-1}\right]\right).$$

In order to compute the off-diagonal entry $A_{k\ell}(z)=\frac{1}{N}d_k\mathcal{O}_{\cdot k}^TB(z)^{-1}\mathcal{O}_{\cdot\ell}$, we use the Shermann-Morrison formula twice: when applied to $B(z)=B_{(k)}(z)+\frac{d_k}{N}\mathcal{O}_{\cdot k}\mathcal{O}_{\cdot k}^T$ we get

$$B(z)^{-1}=B_{(k)}(z)^{-1}-\frac{d_k}{N}\frac{B_{(k)}(z)^{-1}\mathcal{O}_{\cdot k}\mathcal{O}_{\cdot k}^TB_{(k)}(z)^{-1}}{1+\frac{d_k}{N}\mathcal{O}_{\cdot k}^TB_{(k)}(z)^{-1}\mathcal{O}_{\cdot k}};$$

thus, recalling that $g_{(k)} = \frac{1}{N}\mathcal{O}_{\cdot,k}^T B_{(k)}(z)^{-1}\mathcal{O}_{\cdot,k}$, we have

$$A_{k\ell}(z) = \frac{d_k}{1+d_k g_k}\frac{1}{N}\mathcal{O}_{\cdot k}^T B_{(k)}(z)^{-1}\mathcal{O}_{\cdot \ell}.$$

We then apply the Shermann-Morrison formula to $B_{(k)}(z) = B_{(k\ell)}(z) + \frac{d_\ell}{N}\mathcal{O}_{\cdot\ell}\mathcal{O}_{\cdot\ell}^T$ and obtain

$$B_{(k)}(z)^{-1} = B_{(k\ell)}(z)^{-1} - \frac{d_\ell}{N}\frac{B_{(k\ell)}(z)^{-1}\mathcal{O}_{\cdot\ell}\mathcal{O}_{\cdot\ell}^T B_{(k\ell)}(z)^{-1}}{1+\frac{d_\ell}{N}\mathcal{O}_{\cdot\ell}^T B_{(k\ell)}(z)^{-1}\mathcal{O}_{\cdot\ell}}.$$

Thus, we obtain the following formula for the off-diagonal entry:

$$A_{k\ell}(z) = \frac{d_k}{1+d_k g_k}\frac{h_{k\ell}}{1+d_\ell h_\ell} \tag{D.2.13}$$

where $h_\ell = \frac{1}{N}\left(\mathcal{O}_{\cdot\ell}\right)^T B_{(k\ell)}^{-1}(z)\mathcal{O}_{\cdot\ell}$ and $h_{k\ell} = \frac{1}{N}\left(\mathcal{O}_{\cdot k}\right)^T B_{(k\ell)}^{-1}(z)\mathcal{O}_{\cdot\ell}$.

We can apply the results of Section D.2 showing the concentration of g_k around $\tilde{m}_{(k)}$: h_ℓ concentrates around $\tilde{m}_{(k)}$ which itself is close to $\tilde{m}$:

Lemma D.6. *For $z \in \mathbb{H}_{<0}$, and $s \in \mathbb{N}$, we have*

$$\mathbb{E}\left[\left|h_\ell - \tilde{m}\right|^{2s}\right] \le \frac{\boldsymbol{a}_s\left(\mathrm{Tr}[T_K]\right)^{2s}}{|z|^{4s}N^{3s}} + \frac{\boldsymbol{b}_s}{|z|^{2s}N^s},$$

where $\boldsymbol{a}_s, \boldsymbol{b}_s$ only depend on s.

Proof. By convexity, for $k \neq \ell$,

$$\begin{aligned}\mathbb{E}\left[\left|h_\ell - \tilde{m}\right|^{2s}\right] &\le 2^{2s-1}\mathbb{E}\left[\left|h_\ell - \tilde{m}_{(k)}\right|^{2s}\right] + 2^{2s-1}\left|\tilde{m}_{(k)} - \tilde{m}\right|^{2s} \\ &\le 2^{2s-1}\left(\frac{2^{2s-1}\boldsymbol{c}_s\left(\mathrm{Tr}[T_K]\right)^{2s}}{|z|^{4s}N^{3s}} + \frac{2^{2s-1}\boldsymbol{c}_s}{|z|^{2s}N^s}\right) + \frac{2^{2s-1}}{|z|^{2s}N^{2s}}\end{aligned}$$

where for the first term, we applied Proposition D.1 to the matrix $B_{(k)}$ instead of B and the second term is bounded by $\left|\tilde{m}_{(k)} - \tilde{m}\right| \le \frac{1}{|z|N}$ by Lemma D.13. Finally, letting $\boldsymbol{a}_s = 4^{2s-1}\boldsymbol{c}_s$ and $\boldsymbol{b}_s = 4^{2s-1}\boldsymbol{c}_s + 2^{2s-1}$, we obtain the result. □

The scalar $h_{k\ell}$ on the other hand has 0 expectation and, using Wick's formula (Lemma D.9), its variance $\mathbb{E}\left[h_{k\ell}^2\right]$ is equal to $\frac{1}{N^2}\mathbb{E}[\mathrm{Tr}[B_{(k\ell)}^{-2}]] = \frac{1}{N}\mathbb{E}\left[\partial_z m_{(k\ell)}(z)\right]$. Since $\mathbb{E}\left[m_{(k\ell)}(z)\right]$ is close to $\tilde{m}(z)$, from Lemma D.10, its derivative, and hence the variance of $h_{k\ell}$, is close to $\frac{1}{N}\partial_z\tilde{m}$:

Lemma D.7. *For $z \in \mathbb{H}_{<0}$, we have:*

$$\left|\mathbb{E}\left[m_{(k\ell)}(z)\right] - \tilde{m}(z)\right| \le \frac{\mathrm{Tr}[T_K]}{|z|^2N^2} + \frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^3N^2} + \frac{2\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^4}{|z|^5N^4} + \frac{2}{|z|N},$$

where $\boldsymbol{c}_1$ is as in Proposition D.2.

Proof. We use Proposition D.2 and Lemma D.13 twice to obtain

$$\begin{aligned}\left|\mathbb{E}\left[m_{(k\ell)}(z)\right]-\tilde{m}(z)\right| &\leq \left|\mathbb{E}\left[m_{(k\ell)}(z)\right]-\tilde{m}_{(k\ell)}(z)\right|+\left|\tilde{m}_{(k\ell)}(z)-\tilde{m}_{(k)}(z)\right|+\left|\tilde{m}_{(k)}(z)-\tilde{m}(z)\right| \\ &\leq \frac{\operatorname{Tr}[T_K]}{|z|^2 N^2}+\frac{2\boldsymbol{c}_1\left(\operatorname{Tr}[T_K]\right)^2}{|z|^3 N^2}+\frac{2\boldsymbol{c}_1\left(\operatorname{Tr}[T_K]\right)^4}{|z|^5 N^4}+\frac{2}{|z|N},\end{aligned}$$

which yields the desired result. □

To approximate the variance $\operatorname{Var}\left(\left\langle f^{(k)}, A_\lambda f^*\right\rangle_S\right)$ of the coordinate of the noiseless predictor, we need the following results regarding the covariance of the entries of $A(z)$.

Proposition D.5. *For $z \in \mathbb{H}_{<0}$, any $k, \ell \in \mathbb{N}$, we have*

$$\begin{aligned}\left|\operatorname{Var}\left(A_{kk}(z)\right)-\frac{2}{N}\frac{d_k^2\partial_z\tilde{m}}{\left(1+d_k\tilde{m}\right)^4}\right| &\leq \frac{1}{N}\frac{d_k^2|\partial_z\tilde{m}|}{|1+d_k\tilde{m}|^4}\left(\frac{1}{N}+\frac{|z|}{-\Re(z)}\mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right)\right) \\ \left|\operatorname{Var}\left(A_{k\ell}(z)\right)-\frac{1}{N}\frac{d_k^2\partial_z\tilde{m}}{\left(1+d_k\tilde{m}\right)^2\left(1+d_\ell\tilde{m}\right)^2}\right| &\leq \frac{1}{N}\frac{d_k^2|\partial_z\tilde{m}|}{|1+d_k\tilde{m}|^2|1+d_\ell\tilde{m}|^2}\frac{|z|}{-\Re(z)}\mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right) \\ \left|\operatorname{Cov}\left(A_{k\ell}(z),A_{\ell k}(z)\right)-\frac{1}{N}\frac{d_kd_\ell\partial_z\tilde{m}}{\left(1+d_k\tilde{m}\right)^2\left(1+d_\ell\tilde{m}\right)^2}\right| &\leq \frac{1}{N}\frac{d_kd_\ell|\partial_z\tilde{m}|}{|1+d_k\tilde{m}|^2|1+d_\ell\tilde{m}|^2}\frac{|z|}{-\Re(z)}\mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right)\end{aligned}$$

where we use the big-P notation of Definition 7. Whenever a value in the quadruple (k,h,n,ℓ) appears an odd number of times, we have

$$\operatorname{Cov}\left(A_{kh}(z),A_{n\ell}(z)\right)=0.$$

Proof. Let s_k be the symmetry map in the proof of Theorem D.1: the matrices $A(z)$ and $A^{s_k}(z)$ have the same law. Since $A^{s_k}_{\ell n}(z)=-A_{\ell n}(z)$ whenever exactly one of ℓ, n is equal to k, we have for h,n,ℓ distinct from k:

$$\operatorname{Cov}\left(A_{kh}(z),A_{n\ell}(z)\right)=\operatorname{Cov}\left(A^{s_k}_{kh}(z),A^{s_k}_{n\ell}(z)\right)=\operatorname{Cov}\left(-A_{kh}(z),A_{n\ell}(z)\right)$$

which implies that $\operatorname{Cov}\left(A_{kh}(z),A_{n\ell}(z)\right)=0$ when h,n,ℓ are distinct from k. More generally, it is easy to see that $\operatorname{Cov}\left(A_{kh}(z),A_{n\ell}(z)\right)=0$ whenever a value in the quadruple (k,h,n,ℓ) appears an odd number of times.

Approximation of $\operatorname{Var}\left(A_{kk}(z)\right)$**:** Since $\mathbb{E}[A_{kk}(z)]\approx\frac{d_k\tilde{m}}{1+d_k\tilde{m}}$ (Theorem D.1), we decompose the variance of $A_{kk}(z)$ as follows:

$$\operatorname{Var}\left(A_{kk}\right)=\mathbb{E}\left[\left(A_{kk}-\frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right)^2\right]-\left[\mathbb{E}\left[A_{kk}\right]-\frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right]^2.$$

This gives us an approximation $\operatorname{Var}\left(A_{kk}\right)\approx\mathbb{E}\left[\left(A_{kk}-\frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right)^2\right]$ since the term $\left|\mathbb{E}\left[A_{kk}\right]-\frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right|^2$, by using Theorem D.1, we get the following bound :

$$\left|\mathbb{E}\left[A_{kk}\right]-\frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right|^2\leq\left|\frac{d_k\tilde{m}}{(1+d_k\tilde{m})^2}\left(\frac{1}{N}+\mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{|z|N}\right)\right)\right|^2$$

$$= \frac{1}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^4}\left(\frac{1}{N} + 2\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N}\right) + N\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N}\right)^2\right)$$

Since $\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N}\right) = \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{1/2}}\right)$ and $N\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N}\right)^2 = \mathcal{P}\left(\frac{(\mathrm{Tr}[T_K])^2}{|z|^2N}\right)$, we can bound $\left|\mathbb{E}\left[A_{kk}\right] - \frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right|^2$ by

$$\frac{1}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^4}\left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right)\right).$$

Using Formula (D.2.5) for the diagonal entries of A, we have:

$$\left(A_{kk} - \frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right)^2 = \frac{d_k^2\left[g_k - \tilde{m}\right]^2}{(1+d_kg_k)^2(1+d_k\tilde{m})^2}.$$

which can be also expressed as:

$$\left(\frac{d_k\left[g_k-\tilde{m}\right]}{(1+d_kg_k)(1+d_k\tilde{m})}\right)^2 = \left(\frac{d_k\left[g_k-\tilde{m}\right]}{(1+d_k\tilde{m})(1+d_k\tilde{m})} - \frac{d_k^2\left[g_k-\tilde{m}\right]^2}{(1+d_kg_k)(1+d_k\tilde{m})^2}\right)^2.$$

This yields

$$\begin{aligned}\mathbb{E}\left[\left(A_{kk} - \frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right)^2\right] - \mathbb{E}\left[\left(\frac{d_k\left[g_k-\tilde{m}\right]}{(1+d_k\tilde{m})^2}\right)^2\right]& \\ = -\mathbb{E}\left[\frac{d_k^2\left[g_k-\tilde{m}\right]^2}{(1+d_kg_k)(1+d_k\tilde{m})^2}\left(\frac{2d_k\left[g_k-\tilde{m}\right]}{(1+d_k\tilde{m})(1+d_k\tilde{m})} - \frac{d_k^2\left[g_k-\tilde{m}\right]^2}{(1+d_kg_k)(1+d_k\tilde{m})^2}\right)\right]&.\end{aligned}$$

Using Proposition D.1, the absolute value of the r.h.s. can now be bounded by

$$\begin{aligned}\frac{d_k^3\left(2\mathbb{E}\left[|g_k-\tilde{m}|^3\right] + d_k\mathbb{E}\left[|g_k-\tilde{m}|^4\right]\right)}{|1+d_k\tilde{m}|^4} \leq& \frac{d_k^3}{|1+d_k\tilde{m}|^4}2\left(\frac{2^3\boldsymbol{c}_2\left(\mathrm{Tr}[T_K]\right)^4}{|z|^8N^6} + \frac{2^3\boldsymbol{c}_2}{|z|^4N^2}\right)^{\frac{3}{4}} \\ &+ \frac{d_k^4}{|1+d_k\tilde{m}|^4}\left(\frac{2^3\boldsymbol{c}_2\left(\mathrm{Tr}[T_K]\right)^4}{|z|^8N^6} + \frac{2^3\boldsymbol{c}_2}{|z|^4N^2}\right) \\ \leq& \frac{2}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^4}\frac{\mathrm{Tr}[T_K]}{|\tilde{m}|^2}\left(\frac{2^{\frac{9}{4}}\boldsymbol{c}_2^{\frac{3}{4}}\left(\mathrm{Tr}[T_K]\right)^3}{|z|^6N^{\frac{7}{2}}} + \frac{2^{\frac{9}{4}}\boldsymbol{c}_2^{\frac{3}{4}}}{|z|^3N^{\frac{1}{2}}}\right) \\ &+ \frac{1}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^4}\frac{(\mathrm{Tr}[T_K])^2}{|\tilde{m}|^2}\left(\frac{2^3\boldsymbol{c}_2\left(\mathrm{Tr}[T_K]\right)^4}{|z|^8N^5} + \frac{2^3\boldsymbol{c}_2}{|z|^4N}\right),\end{aligned}$$

using the inequality $(a+b)^{\frac{3}{4}} \leq a^{\frac{3}{4}} + b^{\frac{3}{4}}$ and the fact that $d_k \leq \mathrm{Tr}[T_K]$. From Proposition D.3, we have $\frac{1}{\tilde{m}^2} \leq \left(|z| + \frac{\mathrm{Tr}[T_K]}{N}\right)^2$, so that

$$\left|\mathbb{E}\left[\left(A_{kk} - \frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right)^2\right] - \mathbb{E}\left[\left(\frac{d_k\left[g_k-\tilde{m}\right]}{(1+d_k\tilde{m})^2}\right)^2\right]\right| \leq \frac{1}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^4}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right).$$

This yields the approximation $\mathrm{Var}\left(A_{kk}\right) \approx \frac{d_k^2 \mathbb{E}\left[(g_k-\tilde{m})^2\right]}{(1+d_k\tilde{m})^4}$.

Using Wick's formula (Lemma D.9),

$$\mathbb{E}\left[(g_k-\tilde{m})^2\right] = \mathbb{E}\left[\left(m_{(k)}-\tilde{m}\right)^2\right] + \frac{2}{N}\mathbb{E}[\partial_z m_{(k)}(z)],$$

hence we get:

$$\frac{d_k^2\mathbb{E}\left[(g_k-\tilde{m})^2\right]}{(1+d_k\tilde{m})^4} = \frac{\frac{2}{N}d_k^2\partial_z\mathbb{E}[m_{(k)}(z)]}{(1+d_k\tilde{m})^4} + \frac{d_k^2\mathbb{E}\left[\left(m_{(k)}-\tilde{m}\right)^2\right]}{(1+d_k\tilde{m})^4}.$$

Using Proposition D.1,

$$\begin{aligned}\frac{d_k^2\mathbb{E}\left[\left|m_{(k)}-\tilde{m}\right|^2\right]}{|1+d_k\tilde{m}|^4} &\leq \frac{d_k^2}{|1+d_k\tilde{m}|^4}\left|\frac{\boldsymbol{c}_1(\mathrm{Tr}[T_K])^2}{|z|^4N^3}\right| \\ &\leq \frac{1}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^4}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N}\right),\end{aligned}$$

hence the approximation $\mathrm{Var}\left(A_{kk}\right) \approx \frac{\frac{2}{N}d_k^2\partial_z\mathbb{E}[m_{(k)}(z)]}{(1+d_k\tilde{m})^4}$.

At last, by using the approximation $\mathbb{E}[\partial_z m_{(k)}(z)] = \mathbb{E}[\partial_z g_k(z)] \approx \partial_z\tilde{m}(z)$ (Proposition D.2 and Lemma D.10), we obtain

$$\begin{aligned}&\left|\frac{\frac{2}{N}d_k^2\partial_z\mathbb{E}[m_{(k)}(z)]}{(1+d_k\tilde{m})^4} - \frac{\frac{2}{N}d_k^2\partial_z\tilde{m}(z)}{(1+d_k\tilde{m}(z))^4}\right| \\ &\leq \frac{2}{N}\frac{d_k^2}{|1+d_k\tilde{m}|^4}\frac{2}{-\Re(z)}\left(\frac{2^2\mathrm{Tr}[T_K]}{|z|^2N^2} + \frac{2^4\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^3N^2} + \frac{2^6\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^4}{|z|^5N^4} + \frac{2^2}{|z|N}\right) \\ &\leq \frac{2}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^4}\frac{2|z|}{-\Re(z)}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N}\right).\end{aligned}$$

Hence we get the approximation $\mathrm{Var}\left(A_{kk}\right) \approx \frac{\frac{2}{N}d_k^2\partial_z\tilde{m}(z)}{(1+d_k\tilde{m}(z))^4}$, more precisely $\left|\mathrm{Var}\left(A_{kk}\right) - \frac{\frac{2}{N}d_k^2\partial_z\tilde{m}(z)}{(1+d_k\tilde{m}(z))^4}\right|$ is bounded by

$$\frac{2}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^4}\left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right) + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N}\right) + \frac{|z|}{-\Re(z)}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N}\right)\right).$$

Putting everything together, we get

$$\left|\mathrm{Var}\left(A_{kk}\right) - \frac{\frac{2}{N}d_k^2\partial_z\tilde{m}(z)}{(1+d_k\tilde{m}(z))^4}\right| \leq \frac{2}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^4}\left(\frac{1}{N} + \frac{|z|}{-\Re(z)}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right)\right).$$

Since $\partial_z\vartheta = \frac{\partial_z\tilde{m}}{\tilde{m}^2}$, from Proposition D.3 we have $|\partial_\lambda\vartheta(\lambda)| \geq 1$, i.e. $|\tilde{m}|^2 \leq |\partial_\lambda\tilde{m}|$ and thus we conclude.

Approximation of $\mathrm{Cov}\left(A_{k\ell}(z), A_{\ell k}(z)\right)$**:** Note that $A_{k\ell}(z) = \frac{d_k}{N}\mathcal{O}_{\cdot k}^T B(z)^{-1}\mathcal{O}_{\cdot \ell}$, hence, since $B(z)$ is symmetric,

$$A_{k\ell}(z) = \frac{d_k}{d_\ell}A_{\ell k}(z).$$

In particular, we have $\mathrm{Cov}\left(A_{k\ell}(z), A_{\ell k}(z)\right) = \frac{d_\ell}{d_k}\mathrm{Var}\left(A_{k\ell}(z)\right)$. Hence the approximation of $\mathrm{Cov}\left(A_{k\ell}(z), A_{\ell k}(z)\right)$ follows from the one of $\mathrm{Var}\left(A_{k\ell}(z)\right)$.

Approximation of $\mathrm{Var}\left(A_{k\ell}(z)\right)$**:** We have seen in Theorem D.1 that $\mathbb{E}\left(A_{k\ell}(z)\right) = 0$: we need to bound $\mathbb{E}\left(A_{k\ell}(z)^2\right)$. Using Equation (D.2.13):

$$\mathbb{E}\left[A_{k\ell}(z)^2\right] = \mathbb{E}\left[\left(\frac{d_k}{1+d_k g_k}\frac{h_{k\ell}}{1+d_\ell h_\ell}\right)^2\right],$$

where we recall that $h_\ell = \frac{1}{N}\mathcal{O}_{\cdot,\ell}^T B_{(k\ell)}(z)^{-1}\mathcal{O}_{\cdot,\ell}$ and $h_{k\ell} = \frac{1}{N}\mathcal{O}_{\cdot,k}^T B_{(k\ell)}(z)^{-1}\mathcal{O}_{\cdot,\ell}$. Since

$$\frac{d_k}{1+d_k g_k}\frac{h_{k\ell}}{1+d_\ell h_\ell} = \frac{d_k}{1+d_k\tilde{m}}\frac{h_{k\ell}}{1+d_\ell\tilde{m}} - d_k h_{k\ell}\left(\frac{d_k\left(g_k-\tilde{m}\right)\left(1+d_\ell h_\ell\right) + d_\ell\left(1+d_k\tilde{m}\right)\left(h_\ell-\tilde{m}\right)}{\left(1+d_k\tilde{m}\right)\left(1+d_\ell\tilde{m}\right)\left(1+d_k g_k\right)\left(1+d_\ell h_\ell\right)}\right), \tag{D.2.14}$$

using Lemma D.8 below, we get the approximation $\mathbb{E}\left[A_{k\ell}(z)^2\right] \approx \mathbb{E}\left[\frac{d_k^2 h_{k\ell}^2}{(1+d_k\tilde{m})^2(1+d_\ell\tilde{m})^2}\right]$. Using Wick's formula (Lemma D.9 below):

$$\mathbb{E}\left[h_{k\ell}^2\right] = \frac{1}{N}\partial_z\mathbb{E}\left[m_{(k\ell)}(z)\right].$$

Hence the approximation $\mathbb{E}\left[A_{k\ell}(z)^2\right] \approx \frac{\frac{1}{N}d_k^2\partial_z\mathbb{E}\left[m_{(k\ell)}(z)\right]}{(1+d_k\tilde{m})^2(1+d_\ell\tilde{m})^2}$. At last, by using the approximation $\mathbb{E}[\partial_z m_{(kl)}(z)] \approx \partial_z\tilde{m}(z)$ (Lemma D.7 above and the technical complex analysis Lemma D.10 below), we can bound the difference $\left|\frac{\frac{1}{N}d_k^2\partial_z\mathbb{E}\left[m_{(k\ell)}(z)\right]}{(1+d_k\tilde{m})^2(1+d_\ell\tilde{m})^2} - \frac{\frac{1}{N}d_k^2\partial_z\tilde{m}(z)}{(1+d_k\tilde{m})^2(1+d_\ell\tilde{m})^2}\right|$ by

$$\begin{aligned}
&\frac{1}{N}\frac{d_k^2}{|1+d_k\tilde{m}|^2|1+d_\ell\tilde{m}|^2}\frac{2}{-\Re(z)}\left(\frac{2^2\mathrm{Tr}[T_K]}{|z|^2N^2} + \frac{2^4\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^2}{|z|^3N^2} + \frac{2^6\boldsymbol{c}_1\left(\mathrm{Tr}[T_K]\right)^4}{|z|^5N^4} + \frac{2^2}{|z|N}\right)\\
&\leq \frac{1}{N}\frac{d_k^2|\tilde{m}|^2}{|1+d_k\tilde{m}|^2|1+d_\ell\tilde{m}|^2}\frac{2|z|}{-\Re(z)}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N}\right)
\end{aligned}$$

Finally, we can bound the error $\left|\mathbb{E}\left[\left(A_{k\ell}(z)\right)^2\right] - \frac{1}{N}\frac{d_k^2\partial_z\tilde{m}}{(1+d_k\tilde{m})^2(1+d_\ell\tilde{m})^2}\right|$ by

$$\begin{aligned}
&\frac{1}{N}\frac{d_k^2|\partial_z\tilde{m}|}{\left|1+d_k\tilde{m}\right|^2\left|1+d_\ell\tilde{m}\right|^2}\left(\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right) + \frac{2|z|}{-\Re(z)}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right)\right)\\
&\leq \frac{1}{N}\frac{d_k^2|\partial_z\tilde{m}|}{\left|1+d_k\tilde{m}\right|^2\left|1+d_\ell\tilde{m}\right|^2}\frac{|z|}{-\Re(z)}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right).
\end{aligned}$$

□

Lemma D.8. *Using the same notation as in the proof of Proposition D.5,*

$$\epsilon_{k\ell} = \mathbb{E}\left[A_{k\ell}(z)^2\right] - \frac{d_k^2 h_{k\ell}^2}{(1+d_k\tilde{m})^2(1+d_\ell\tilde{m})^2}$$

is bounded by:

$$|\epsilon_{k\ell}| \leq \frac{1}{N}\frac{d_k^2\partial_\lambda\tilde{m}}{\left|1+d_\ell\tilde{m}\right|^2\left|1+d_k\tilde{m}\right|^2}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right)$$

Proof. Using Equation D.2.14, by setting $c = 2\frac{1}{1+d_k\tilde{m}}\frac{1}{1+d_\ell\tilde{m}}$, $X_1 = d_k h_{k\ell}$, and

$$X_2 = \frac{d_k\left(g_k-\tilde{m}\right)}{\left(1+d_k\tilde{m}\right)\left(1+d_\ell\tilde{m}\right)\left(1+d_k g_k\right)} + \frac{d_\ell\left(h_\ell-\tilde{m}\right)}{\left(1+d_\ell\tilde{m}\right)\left(1+d_k g_k\right)\left(1+d_\ell h_\ell\right)},$$

we have that $\epsilon_{k\ell}$ is equal to:

$$\epsilon_{k\ell} = \mathbb{E}\left[-X_1^2 X_2(c-X_2)\right]$$

we can thus control $\epsilon_{k\ell}$ with the following bound

$$\begin{aligned}|\epsilon_{k\ell}| &\leq c\mathbb{E}\left[|X_1|^2|X_2|\right] + \mathbb{E}\left[|X_1|^2|X_2|^2\right] \\ &\leq \mathbb{E}\left[|X_1|^4\right]^{\frac{1}{2}}\left(c\mathbb{E}\left[|X_2|^2\right]^{\frac{1}{2}} + \mathbb{E}\left[|X_2|^4\right]^{\frac{1}{2}}\right).\end{aligned}$$

- Bound on $\mathbb{E}[|X_1|^4]$: using the same argument as for $\mathbb{E}\left[|m_{(k)}-g_k|^{2s}\right]$ and Wick's formula (Lemma D.9), there exists a constant $\boldsymbol{a}$ such that

$$\mathbb{E}\left[|X_1|^4\right]^{\frac{1}{2}} = \mathbb{E}\left[|d_k h_{k\ell}|^4\right]^{\frac{1}{2}} = d_k^2\mathbb{E}\left[|h_{k\ell}|^4\right]^{\frac{1}{2}} \leq \frac{\boldsymbol{a}d_k^2}{|z|^2 N}$$

- Bound on $\mathbb{E}[|X_2|^{2s}]$: in order to bound $\mathbb{E}[|X_2|^{2s}]$ we decompose X_2 as $X_2 = Y_1 + Y_2 + Y_3$ where

$$\begin{aligned}Y_1 &= \frac{d_k\left(g_k-\tilde{m}\right)}{\left(1+d_k\tilde{m}\right)\left(1+d_\ell\tilde{m}\right)\left(1+d_k g_k\right)}, \\ Y_2 &= \frac{d_\ell\left(h_\ell-\tilde{m}\right)}{\left(1+d_\ell\tilde{m}\right)\left(1+d_k\tilde{m}\right)\left(1+d_\ell h_\ell\right)}, \\ Y_3 &= \frac{d_\ell d_k\left(h_\ell-\tilde{m}\right)\left(\tilde{m}-g_k\right)}{\left(1+d_\ell\tilde{m}\right)\left(1+d_k\tilde{m}\right)\left(1+d_k g_k\right)\left(1+d_\ell h_\ell\right)},\end{aligned}$$

  so that by Minkowski inequality,

$$\mathbb{E}\left[|X_2|^{2s}\right]^{\frac{1}{2s}} \leq \mathbb{E}\left[|Y_1|^{2s}\right]^{\frac{1}{2s}} + \mathbb{E}\left[|Y_2|^{2s}\right]^{\frac{1}{2s}} + \mathbb{E}\left[|Y_3|^{2s}\right]^{\frac{1}{2s}},$$

  We can bound the terms in the r.h.s. of the above by applying Proposition D.1 and Lemma D.6:

  – Bound on $\mathbb{E}[|Y_1|^{2s}]$:

$$\mathbb{E}\left[|Y_1|^{2s}\right]^{\frac{1}{2s}} \leq \frac{d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\mathbb{E}\left[|\left(g_k-\tilde{m}\right)|^{2s}\right]^{\frac{1}{2s}} \leq \frac{d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\left[\frac{2^{2s-1}\boldsymbol{c}_s(\mathrm{Tr}[T_K])^{2s}}{|z|^{4s}N^{3s}} + \frac{2^{2s-1}\boldsymbol{c}_s}{|z|^{2s}N^s}\right]^{\frac{1}{2s}}$$

  – Bound on $\mathbb{E}[|Y_2|^{2s}]$:

$$\mathbb{E}\left[|Y_2|^{2s}\right]^{\frac{1}{2s}} \leq \frac{d_\ell}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\mathbb{E}\left[|\left(h_\ell-\tilde{m}\right)|^{2s}\right]^{\frac{1}{2s}} \leq \frac{d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\left[\frac{\boldsymbol{a}_s(\mathrm{Tr}[T_K])^{2s}}{|z|^{4s}N^{3s}} + \frac{\boldsymbol{b}_s}{|z|^{2s}N^s}\right]^{\frac{1}{2s}}$$

– Bound on $\mathbb{E}\left[|Y_3|^{2s}\right]^{\frac{1}{2s}}$:

$$\begin{aligned}\mathbb{E}\left[|Y_3|^{2s}\right]^{\frac{1}{2s}} &\leq \frac{d_\ell d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\mathbb{E}\left[|(h_\ell-\tilde{m})|^{2s}\,|(\tilde{m}-g_k)|^{2s}\right]^{\frac{1}{2s}}\\ &\leq \frac{d_\ell d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\mathbb{E}\left[|(h_\ell-\tilde{m})|^{4s}\right]^{\frac{1}{4s}}\mathbb{E}\left[|(\tilde{m}-g_k)|^{4s}\right]^{\frac{1}{4s}}\\ &\leq \frac{d_\ell d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\left[\frac{\boldsymbol{a}_{2s}(\mathrm{Tr}[T_K])^{4s}}{|z|^{8s}N^{6s}}+\frac{\boldsymbol{b}_{2s}}{|z|^{4s}N^{2s}}\right]^{\frac{1}{4s}}\left[\frac{2^{4s-1}\boldsymbol{c}_{2s}(\mathrm{Tr}[T_K])^{4s}}{|z|^{8s}N^{6s}}+\frac{2^{4s-1}\boldsymbol{c}_{2s}}{|z|^{4s}N^{2s}}\right]^{\frac{1}{4s}}\end{aligned}$$

Let $\boldsymbol{r}_s=\max\{2^{2s-1}\boldsymbol{c}_s,\boldsymbol{a}_s\}$ and $\boldsymbol{t}_s=\max\{2^{2s-1}\boldsymbol{c}_s,\boldsymbol{b}_s\}$; then putting the pieces together we have

$$\mathbb{E}\left[|X_2|^{2s}\right]^{\frac{1}{2s}}\leq\frac{d_\ell+d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\left[\frac{\boldsymbol{r}_s(\mathrm{Tr}[T_K])^{2s}}{|z|^{4s}N^{3s}}+\frac{\boldsymbol{t}_s}{|z|^{2s}N^s}\right]^{\frac{1}{2s}}+\frac{d_\ell d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\left[\frac{\boldsymbol{r}_{2s}(\mathrm{Tr}[T_K])^{4s}}{|z|^{8s}N^{6s}}+\frac{\boldsymbol{t}_{2s}}{|z|^{4s}N^{2s}}\right]^{\frac{1}{2s}}$$

and thus

$$\mathbb{E}\left[|X_2|^{2}\right]^{\frac{1}{2}}\leq\frac{d_\ell+d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\left[\frac{\boldsymbol{r}_1^{1/2}(\mathrm{Tr}[T_K])}{|z|^{2}N^{3/2}}+\frac{\boldsymbol{t}_1^{1/2}}{|z|\sqrt{N}}\right]+\frac{d_\ell d_k}{|1+d_\ell\tilde{m}|\,|1+d_k\tilde{m}|}\left[\frac{\boldsymbol{r}_2^{1/2}(\mathrm{Tr}[T_K])^{2}}{|z|^{4}N^{3}}+\frac{\boldsymbol{t}_2^{1/2}}{|z|^{2}N^{1}}\right]$$

$$\mathbb{E}\left[|X_4|^{4}\right]^{\frac{1}{2}}\leq\frac{2(d_\ell+d_k)^2}{|1+d_\ell\tilde{m}|^2\,|1+d_k\tilde{m}|^2}\left[\frac{\boldsymbol{r}_2^{1/2}(\mathrm{Tr}[T_K])^{2}}{|z|^{4}N^{3}}+\frac{\boldsymbol{t}_2^{1/2}}{|z|^{2}N}\right]+\frac{2d_\ell^2d_k^2}{|1+d_\ell\tilde{m}|^2\,|1+d_k\tilde{m}|^2}\left[\frac{\boldsymbol{r}_4^{1/2}(\mathrm{Tr}[T_K])^{4}}{|z|^{8}N^{6}}+\frac{\boldsymbol{t}_4^{1/2}}{|z|^{4}N^{2}}\right]$$

And finally, putting all the pieces together, we have

$$\begin{aligned}|\epsilon_{k\ell}|&\leq\mathbb{E}\left[|X_1|^4\right]^{\frac{1}{2}}\left(c\mathbb{E}\left[|X_2|^2\right]^{\frac{1}{2}}+\mathbb{E}\left[|X_2|^4\right]^{\frac{1}{2}}\right)\\ &\leq\frac{\boldsymbol{a}d_k^2}{|z|^2\,N}\frac{2(d_\ell+d_k)}{|1+d_\ell\tilde{m}|^2\,|1+d_k\tilde{m}|^2}\left[\frac{\boldsymbol{r}_1^{1/2}(\mathrm{Tr}[T_K])}{|z|^2N^{3/2}}+\frac{\boldsymbol{t}_1^{1/2}}{|z|\sqrt{N}}\right]\\ &\quad+\frac{\boldsymbol{a}d_k^2}{|z|^2\,N}\frac{2(d_\ell+d_k)^2+2d_\ell d_k}{|1+d_\ell\tilde{m}|^2\,|1+d_k\tilde{m}|^2}\left[\frac{\boldsymbol{r}_2^{1/2}(\mathrm{Tr}[T_K])^{2}}{|z|^4N^{3}}+\frac{\boldsymbol{t}_2^{1/2}}{|z|^2N}\right]\\ &\quad+\frac{\boldsymbol{a}d_k^2}{|z|^2\,N}\frac{2d_\ell^2d_k^2}{|1+d_\ell\tilde{m}|^2\,|1+d_k\tilde{m}|^2}\left[\frac{\boldsymbol{r}_4^{1/2}(\mathrm{Tr}[T_K])^{4}}{|z|^8N^{6}}+\frac{\boldsymbol{t}_4^{1/2}}{|z|^4N^{2}}\right].\end{aligned}$$

Using the fact that $|\partial_z\tilde{m}|\leq|\tilde{m}|^2$ and Proposition D.3, we get:

$$\frac{1}{|z|^2}\leq\frac{|\partial_z\tilde{m}|}{|z|^2|\tilde{m}|^2}\leq|\partial_z\tilde{m}|\left(1+2\frac{\mathrm{Tr}[T_K]}{|z|N}+\frac{(\mathrm{Tr}[T_K])^2}{|z|^2N^2}\right),$$

we conclude saying that

$$|\epsilon_{k\ell}|\leq\frac{1}{N}\frac{d_k^2\partial_z\tilde{m}}{|1+d_\ell\tilde{m}|^2\,|1+d_k\tilde{m}|^2}\mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{|z|N^{\frac{1}{2}}}\right).$$

□

Remark D.5. Since $\tilde{m}(z) = \frac{1}{\vartheta(-z)}$, the derivative $\partial_z \tilde{m}(z)$ can also be expressed in terms of the SCT: $\partial_z \tilde{m}(z) = \partial_z \vartheta(-z) \frac{1}{\vartheta(-z)^2}$, hence the previous approximations can also be written as:

$$\mathrm{Var}\left(A_{kk}(z)\right) \approx \frac{2}{N} \frac{d_k^2 \vartheta(-z)^2 \partial_z \vartheta(-z)}{(\vartheta(-z)+d_k)^4} \qquad \mathrm{Var}\left(A_{k\ell}(z)\right) \approx \frac{1}{N} \frac{d_k^2 \vartheta(-z)^2 \partial_z \vartheta(-z)}{(\vartheta(-z)+d_k)^2(\vartheta(-z)+d_\ell)^2}.$$

We can now describe the variance of the predictor. The variance of the predictor along the eigenfunction $f^{(k)}$ is estimated by V_k, where

$$V_k(f^*, \lambda, N, \epsilon) = \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\left\| (I_{\mathcal{C}} - \tilde{A}_\vartheta) f^* \right\|_S^2 + \epsilon^2 + \left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta^2(\lambda)}{(\vartheta(\lambda)+d_k)^2} \right) \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}.$$

Theorem D.2. *There is a constant $\boldsymbol{C}_1 > 0$ such that, with the notation of Definition 7, we have*

$$\left| \mathrm{Var}\left(\left\langle f^{(k)}, \hat{f}_\lambda^\epsilon \right\rangle_S \right) - V_k(f^*, \lambda, N, \epsilon) \right| \leq \left(\frac{\boldsymbol{C}_1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N^{\frac{1}{2}}} \right) \right) V_k(f^*, \lambda, N, \epsilon).$$

Proof. Using the law of total variance, we decompose the variance with respect to the observations $\mathcal{O}$ and the vector of noise $E = (e_1, \ldots, e_N)^T$

$$\begin{aligned}
\mathrm{Var}\left(\left\langle f^{(k)}, \hat{f}_\lambda^\epsilon \right\rangle_S \right) &= \mathrm{Var}_{\mathcal{O}} \left(\left\langle f^{(k)}, \mathbb{E}_E \left[\hat{f}_\lambda^\epsilon \right] \right\rangle_S \right) + \epsilon^2 \mathbb{E}_{\mathcal{O}} \left[\mathrm{Var}_E \left(\frac{d_k}{N} (\mathcal{O}_{\cdot k})^T \left(\frac{1}{N} \mathcal{O} K \mathcal{O}^T + \lambda I_N \right)^{-1} E \right) \right] \\
&= \mathrm{Var}_{\mathcal{O}} \left(\left\langle f^{(k)}, A(-\lambda) f^* \right\rangle_S \right) + \epsilon^2 \mathbb{E}_{\mathcal{O}} \left[\frac{d_k}{N} \partial_\lambda A_{kk}(-\lambda) \right].
\end{aligned}$$

Since the randomness is now only on A through $\mathcal{O}$, from now on, we will lighten the notation by sometimes omitting the $\mathcal{O}$ dependence in the expectations.

We first show how the approximation $V_k(f^*, \lambda, N, \epsilon)$ appears, and then establish the bounds which allow one to study the quality of this approximation.

Approximations: Decomposing the true function along the principal components $f^* = \sum_{k=1}^\infty b_k f^{(k)}$ with $b_k = \left\langle f^{(k)}, f^* \right\rangle_S$, we have

$$\mathrm{Var}(\langle f^{(k)}, A(-\lambda) f^* \rangle_S) = \sum_\ell b_\ell^2 \mathrm{Var}\left(A_{k\ell}(-\lambda)\right).$$

From Proposition D.5 and the remark after, we have two different approximations for $\mathrm{Var}\left(A_{k\ell}(-\lambda)\right)$. For any $\ell \neq k$, we have

$$\mathrm{Var}\left(A_{kk}(-\lambda)\right) \approx \frac{2}{N} \frac{d_k^2 \vartheta(\lambda)^2 \partial_\lambda \vartheta(\lambda)}{(\vartheta(\lambda)+d_k)^4}, \qquad \mathrm{Var}\left(A_{k\ell}(-\lambda)\right) \approx \frac{1}{N} \frac{d_k^2 \vartheta(\lambda)^2 \partial_\lambda \vartheta(\lambda)}{(\vartheta(\lambda)+d_k)^2(\vartheta(\lambda)+d_\ell)^2}.$$

Hence

$$\begin{aligned}
\mathrm{Var}(\langle f^{(k)}, A_\lambda f^* \rangle_S) &\approx \frac{b_k^2}{N} \frac{d_k^2 \vartheta(\lambda)^2 \partial_\lambda \vartheta(\lambda)}{(\vartheta(\lambda)+d_k)^4} + \sum_\ell \frac{b_\ell^2}{N} \frac{d_k^2 \vartheta(\lambda)^2 \partial_\lambda \vartheta(\lambda)}{(\vartheta(\lambda)+d_k)^2(\vartheta(\lambda)+d_\ell)^2} \\
&= \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta^2(\lambda)}{(\vartheta(\lambda)+d_k)^2} + \sum_\ell b_\ell^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_\ell)^2} \right) \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}.
\end{aligned}$$

Since $\sum_\ell b_\ell^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_\ell)^2} = \|(I_{\mathcal{C}} - \tilde{A}_\vartheta) f^*\|_S^2$, this provides the approximation:

$$\mathrm{Var}(\langle f^{(k)}, A_\lambda f^* \rangle_S) \approx \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\|(I_{\mathcal{C}} - \tilde{A}_\vartheta) f^*\|_S^2 + \langle f^{(k)}, f^* \rangle_S^2 \frac{\vartheta^2(\lambda)}{(\vartheta(\lambda)+d_k)^2} \right) \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}. \tag{D.2.15}$$

Now, using Lemma D.10 and Theorem D.1:

$$\epsilon^2 \mathbb{E}_{\mathcal{O}} \left[\frac{d_k}{N} \partial_\lambda A_{kk}(-\lambda) \right] \approx \epsilon^2 \frac{\partial_\lambda \vartheta(\lambda)}{N} \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}. \tag{D.2.16}$$

Combining Equations D.2.15 and D.2.16, we obtain the approximation

$$\mathrm{Var}(\langle f^{(k)}, \hat{f}_\lambda^\epsilon \rangle_S) \approx V_k(f^*, \lambda, N, \epsilon).$$

Now, we explain how to quantify the quality of the approximations, and thus how to get the bound stated in the theorem. Recall that we decomposed $\mathrm{Var}(\langle f^{(k)}, \hat{f}_\lambda^\epsilon \rangle_S)$ into two terms using the law of total variance.

First term: We have seen that:

$$\mathrm{Var}\left(\left\langle f^{(k)}, A_\lambda f^* \right\rangle_S \right) = b_k^2 \mathrm{Var}\left(A_{kk}(-\lambda) \right) + \sum_{\ell \neq k} b_\ell^2 \mathrm{Var}\left(A_{k\ell}(-\lambda) \right).$$

By Proposition D.5, we have

$$\begin{aligned}
\left| b_k^2 \mathrm{Var}\left(A_{kk}(-\lambda) \right) - 2 b_k^2 \frac{\partial_\lambda \vartheta(\lambda)}{N} \frac{\vartheta(\lambda)^2 d_k^2}{(\vartheta(\lambda)+d_k)^4} \right| &= b_k^2 \left| \mathrm{Var}\left(A_{kk}(-\lambda) \right) - \frac{2}{N} \frac{d_k^2 \partial_\lambda \tilde{m}}{(1+d_k \tilde{m})^4} \right| \\
&\leq b_k^2 \frac{|\partial_\lambda \vartheta(\lambda)|}{N} \frac{|\vartheta(\lambda)|^2 d_k^2}{|\vartheta(\lambda)+d_k|^4} \left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N^{\frac{1}{2}}} \right) \right)
\end{aligned}$$

and

$$\left| b_\ell^2 \mathrm{Var}\left(A_{k\ell}(-\lambda) \right) - \frac{1}{N} b_\ell^2 \frac{d_k^2 \partial_\lambda \tilde{m}}{(1+d_k\tilde{m})^2 (1+d_\ell \tilde{m})^2} \right| \leq b_\ell^2 \frac{1}{N} \frac{d_k^2 |\vartheta(\lambda)|^2 |\partial_\lambda \vartheta(\lambda)|}{|\vartheta(\lambda)+d_k|^2 |\vartheta(\lambda)+d_\ell|^2} \mathcal{P}\left(\frac{\mathrm{Tr}\left[T_K\right]}{\lambda N^{\frac{1}{2}}} \right).$$

Thus we have

$$\begin{aligned}
&\left| \sum_\ell b_\ell^2 \mathrm{Var}\left(A_{k\ell}(-\lambda) \right) - \frac{\partial_\lambda \vartheta(\lambda)}{N} \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \left(2 b_k^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2} + \sum_{\ell \neq k} b_\ell^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_\ell)^2} \right) \right| \\
&\leq b_k^2 \left| \mathrm{Var}\left(A_{kk}(-\lambda) \right) - \frac{2}{N} \frac{d_k^2 \partial_\lambda \tilde{m}}{(1+d_k\tilde{m})^4} \right| + \sum_{\ell \neq k} b_\ell^2 \left| \mathrm{Var}\left(A_{k\ell}(-\lambda) \right) - \frac{1}{N} \frac{d_k^2 \partial_\lambda \tilde{m}}{(1+d_k\tilde{m})^2 (1+d_\ell \tilde{m})^2} \right| \\
&\leq b_k^2 \frac{1}{N} \frac{d_k^2 |\vartheta(\lambda)|^2 |\partial_\lambda \vartheta(\lambda)|}{|\vartheta(\lambda)+d_k|^4} \left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N^{\frac{1}{2}}} \right) \right) + \sum_{\ell \neq k} b_\ell^2 \frac{1}{N} \frac{d_k^2 |\vartheta(\lambda)|^2 |\partial_\lambda \vartheta(\lambda)|}{|\vartheta(\lambda)+d_k|^2 |\vartheta(\lambda)+d_\ell|^2} \mathcal{P}\left(\frac{\mathrm{Tr}\left[T_K\right]}{\lambda N^{\frac{1}{2}}} \right) \\
&\leq \frac{|\partial_\lambda \vartheta(\lambda)|}{N} \frac{d_k^2}{|\vartheta(\lambda)+d_k|^2} \sum_\ell b_\ell^2 \frac{|\vartheta(\lambda)|^2}{|\vartheta(\lambda)+d_\ell|^2} \left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}\left[T_K\right]}{\lambda N^{\frac{1}{2}}} \right) \right)
\end{aligned}$$

$$\leq \frac{|\partial_\lambda \vartheta(\lambda)|}{N} \frac{d_k^2}{|\vartheta(\lambda)+d_k|^2} \left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}\right) f^* \right\|_S^2 \left(\frac{1}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{\lambda N^{\frac{1}{2}}}\right)\right).$$

We deduce:

$$\begin{aligned}
&\left| \operatorname{Var}_{\mathcal{O}}\left(\left\langle f^{(k)}, A_\lambda f^* \right\rangle_S\right) - \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\left\| \left(I_{\mathcal{C}} - \tilde{A}_\lambda\right) f^* \right\|_S^2 + \left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2} \right) \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \right| \\
&\leq \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}\right) f^* \right\|_S^2 \right) \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \left(\frac{1}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{\lambda N^{\frac{1}{2}}}\right)\right) \\
&\leq \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}\right) f^* \right\|_S^2 + \left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2} \right) \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \left(\frac{1}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{\lambda N^{\frac{1}{2}}}\right)\right).
\end{aligned}$$

Second term: To approximate, we apply Cauchy's inequality to Equation (D.2.12) of Theorem D.1:

$$\begin{aligned}
\left| \mathbb{E}\left[\partial_z A_{kk}(z)\right] - \partial_z \vartheta(-z) \frac{d_k}{(\vartheta(-z)+d_k)^2} \right| &\leq \frac{2}{-\Re(z)} \sup_{|w-z|=-\frac{1}{2}\Re(z)} \left| \mathbb{E}[A_{kk}(w)] - \frac{d_k}{\vartheta(-w)+d_k} \right| \\
&\leq \frac{2}{-\Re(z)} \sup_{|w-z|=-\frac{1}{2}\Re(z)} \frac{d_k\,|\vartheta(-w)|}{|\vartheta(-w)+d_k|^2} \left(\frac{1}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{|w|\,N}\right)\right).
\end{aligned}$$

By choosing $z=-\lambda$, in the region $\{w \in \mathbb{C} \mid |w+\lambda| = \frac{\lambda}{2}\}$ the polynomial $\mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{|w|N}\right)$ is uniformly bounded by $\mathcal{P}\left(\frac{2\operatorname{Tr}[T_K]}{\lambda N}\right)$ and $\frac{d_k|\vartheta(-w)|}{|\vartheta(-w)+d_k|^2} \leq \frac{d_k|\vartheta(\lambda)|}{|\vartheta(\lambda)+d_k|^2}$. Thus we get

$$\begin{aligned}
\left| \mathbb{E}\left[\partial_\lambda A_{kk}(-\lambda)\right] - \partial_\lambda \vartheta(\lambda) \frac{d_k}{(\vartheta(\lambda)+d_k)^2} \right| &\leq 2 \frac{d_k}{|\vartheta(\lambda)+d_k|^2} \frac{\vartheta(\lambda)}{\lambda} \left(\frac{1}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{\lambda N}\right)\right) \\
&\leq 2 \frac{d_k}{|\vartheta(\lambda)+d_k|^2} \left(1 + \frac{\operatorname{Tr}[T_K]}{\lambda N}\right) \left(\frac{1}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{\lambda N}\right)\right) \\
&\leq \frac{d_k}{|\vartheta(\lambda)+d_k|^2} \left(\frac{2}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{\lambda N}\right)\right).
\end{aligned}$$

By using the fact that $1 \leq |\partial_\lambda \vartheta(\lambda)|$ (see Proposition D.3), we have that

$$\left| \frac{d_k}{N} \mathbb{E}\left[\partial_\lambda A_{kk}(-\lambda)\right] - \frac{\partial_\lambda \vartheta(\lambda)}{N} \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \right| \leq \frac{|\partial_\lambda \vartheta(\lambda)|}{N} \frac{d_k^2}{|\vartheta(\lambda)+d_k|^2} \left(\frac{2}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{|z|\,N}\right)\right).$$

Finally, by putting the bounds for the two terms together we have

$$\begin{aligned}
&\left| \operatorname{Var}\left(\left\langle f^{(k)}, \hat{f}_\lambda^\epsilon \right\rangle_S\right) - \frac{\partial_\lambda \vartheta(\lambda)}{N} \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \left(2b_k^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2} + \sum_{\ell \neq k} b_\ell^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_\ell)^2} + \epsilon^2 \right) \right| \\
&\leq \left| \operatorname{Var}\left(\left\langle f^{(k)}, A_\lambda f^* \right\rangle_S\right) - \frac{\partial_\lambda \vartheta(\lambda)}{N} \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \left(2b_k^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2} + \sum_{\ell \neq k} b_\ell^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_\ell)^2} \right) \right| \\
&\qquad + \epsilon^2 \frac{d_k}{N} \left| \partial_\lambda \mathbb{E}[A_{kk}(-\lambda)] - \partial_\lambda \vartheta(\lambda) \frac{d_k}{(\vartheta(\lambda)+d_k)^2} \right|
\end{aligned}$$

$$\leq \frac{\partial_\lambda \vartheta(\lambda)}{N}\left(\left\|\left(I_{\mathcal{C}}-\tilde{A}_{\vartheta(\lambda)}\right)f^*\right\|_S^2+\epsilon^2\right)\frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}\left(\frac{2}{N}+\mathcal{P}\left(\frac{\mathrm{Tr}\left[T_K\right]}{\lambda N^{\frac{1}{2}}}\right)\right).$$

This concludes the proof. □

Expected Risk

We now have all the tools required to describe the expected risk and empirical risk. In particular, we now show that the distance between the expected risk $\mathbb{E}[R^\epsilon(\hat{f}_\lambda^\epsilon)]$ and

$$\tilde{R}^\epsilon(f^*,\lambda)=\partial_\lambda\vartheta(\lambda)(\|(I_{\mathcal{C}}-\tilde{A}_{\vartheta(\lambda)})f^*\|_S^2+\epsilon^2)$$

is relatively small:

Theorem D.3. *We have*

$$\left|\mathbb{E}\left[R^\epsilon\left(\hat{f}_\lambda^\epsilon\right)\right]-\tilde{R}^\epsilon\left(f^*,\lambda\right)\right|\leq\tilde{R}^\epsilon\left(f^*,\lambda\right)\left(\frac{1}{N}+\mathcal{P}\left(\frac{\mathrm{Tr}\left[T_K\right]}{\lambda N^{\frac{1}{2}}}\right)\right).$$

Proof. The expected risk can be written as $\mathbb{E}[R^\epsilon(\hat{f}_\lambda^\epsilon)]=\mathbb{E}[\|\hat{f}_\lambda^\epsilon-f^*\|_S^2]+\epsilon^2=\sum_k\mathbb{E}[(a_k-b_k)^2]+\epsilon^2$, where $a_k=\langle f^{(k)},\hat{f}_\lambda^\epsilon\rangle_S$ and $b_k=\langle f^{(k)},f^*\rangle_S$. Hence, using the classical bias-variance decomposition for each summand, we get that the expected risk is equal to:

$$\mathbb{E}[R^\epsilon(\hat{f}_\lambda^\epsilon)]=R^\epsilon(\mathbb{E}[\hat{f}_\lambda^\epsilon])+\sum_{k=1}^\infty\mathrm{Var}(\langle f^{(k)},\hat{f}_\lambda^\epsilon\rangle_S).$$

Similarly to the proof of Theorem D.2, we explain how the approximation of the expected arises, then we establish the bounds which allow one to study the quality of this approximation.

Approximations: The bias term $R^\epsilon(\mathbb{E}[\hat{f}_\lambda^\epsilon])$ is equal to $\|\mathbb{E}[\hat{f}_\lambda^\epsilon]-f^*\|_S^2+\epsilon^2=\|(I_{\mathcal{C}}-\mathbb{E}[A_\lambda])f^*\|_S^2+\epsilon^2$. Using Theorem D.1, one gets the approximation of the bias term:

$$R^\epsilon(\mathbb{E}[\hat{f}_\lambda^\epsilon])\approx\|(I_{\mathcal{C}}-\tilde{A}_{\vartheta(\lambda)})f^*\|_S^2+\epsilon^2.$$

As for the variance term $\sum_{k=1}^\infty\mathrm{Var}(\langle f^{(k)},\hat{f}_\lambda^\epsilon\rangle_S)$, we use Theorem D.2.

$$\sum_{k=1}^\infty\mathrm{Var}(\langle f^{(k)},\hat{f}_\lambda^\epsilon\rangle_S)\approx\sum_{k=1}^\infty V_k(f^*,\lambda,N,\epsilon),$$

where

$$V_k(f^*,\lambda,N,\epsilon)=\frac{\partial_\lambda\vartheta(\lambda)}{N}\left(\left\|(I_{\mathcal{C}}-\tilde{A}_\vartheta)f^*\right\|_S^2+\epsilon^2+\left\langle f^{(k)},f^*\right\rangle_S^2\frac{\vartheta^2(\lambda)}{(\vartheta(\lambda)+d_k)^2}\right)\frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}.$$

Thus the variance term is approximately equal to:

$$(\|(I_{\mathcal{C}}-\tilde{A}_\vartheta)f^*\|_S^2+\epsilon^2)\frac{\partial_\lambda\vartheta(\lambda)}{N}\sum_{k=1}^\infty\frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}+\frac{\partial_\lambda\vartheta(\lambda)}{N}\sum_{k=1}^\infty\langle f^{(k)},f^*\rangle_S^2\frac{\vartheta^2(\lambda)d_k^2}{(\vartheta(\lambda)+d_k)^4}.$$

Noting that from Equation D.2.10, we have $(\partial_\lambda \vartheta(\lambda) - 1) = \frac{\partial_\lambda \vartheta(\lambda)}{N} \sum_{k=1}^{\infty} \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}$, we get:

$$\sum_{k=1}^{\infty} \mathrm{Var}(\langle f^{(k)}, \hat{f}_\lambda^\epsilon \rangle_S) \approx (\partial_\lambda \vartheta(\lambda) - 1)(\|(I_\mathcal{C} - \tilde{A}_\vartheta) f^*\|_S^2 + \epsilon^2) + \frac{\partial_\lambda \vartheta(\lambda)}{N} \sum_{k=1}^{\infty} \langle f^{(k)}, f^* \rangle_S^2 \frac{\vartheta^2(\lambda) d_k^2}{(\vartheta(\lambda)+d_k)^4}.$$

The second term in the r.h.s. is a residual term: using the fact that $\frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \leq 1$, this term is bounded by $\frac{\partial_\lambda \vartheta(\lambda)}{N} \|(I_\mathcal{C} - \tilde{A}_{\vartheta(\lambda)}) f^*\|_S^2$.

Hence, we get the following approximation of the variance term:

$$\sum_{k=1}^{\infty} \mathrm{Var}(\langle f^{(k)}, \hat{f}_\lambda^\epsilon \rangle_S) \approx (\partial_\lambda \vartheta(\lambda) - 1)(\|(I_\mathcal{C} - \tilde{A}_\vartheta) f^*\|_S^2 + \epsilon^2).$$

Putting the approximations of the bias and variance terms together, we obtain:

$$\mathbb{E}\left[R^\epsilon\left(\hat{f}_\lambda^\epsilon\right)\right] \approx \tilde{R}^\epsilon\left(f^*, \lambda\right).$$

Now, we explain how to quantify the quality of the approximations, and thus how to get the bound stated in the theorem. Recall that, using the bias-variance decomposition, we split the expected risk into two terms, the bias term and the variance term. We show now that:

$$\left| R^\epsilon(\mathbb{E}_{\mathcal{O},E}[\hat{f}_\lambda^\epsilon]) - \left(\|(I_\mathcal{C} - \tilde{A}_{\vartheta(\lambda)}) f^*\|_S^2 + \epsilon^2\right) \right| \leq \|(I_\mathcal{C} - \tilde{A}_{\vartheta(\lambda)}) f^*\|_S^2 \left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}\left[T_K\right]}{\lambda N}\right)\right)$$

and

$$\left| \sum_{k=1}^{\infty} \mathrm{Var}(\langle f^{(k)}, \hat{f}_\lambda^\epsilon \rangle_S) - (\partial_\lambda \vartheta(\lambda) - 1)\left(\|(I_\mathcal{C} - \tilde{A}_{\vartheta(\lambda)}) f^*\|_S^2 + \epsilon^2\right) \right| \leq$$
$$\partial_\lambda \vartheta(\lambda)\left(\|(I_\mathcal{C} - \tilde{A}_{\vartheta(\lambda)}) f^*\|_S^2 + \epsilon^2\right)\left(\frac{2}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}\left[T_K\right]}{\lambda N^{\frac{1}{2}}}\right)\right).$$

Combining the two inequations, and using the fact that $1 \leq \partial_\lambda \vartheta(\lambda)$, we then get the desired inequality.

Bias term: Since $|\tilde{A}_{\vartheta(\lambda),kk}| \leq 1$, Equation (D.2.12) of Theorem D.1 implies that

$$\left|\tilde{A}_{\vartheta(\lambda),kk} - \mathbb{E}\left[A_{kk}(-\lambda)\right]\right| \leq |1 - \tilde{A}_{\vartheta(\lambda),kk}| \left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N}\right)\right).$$

We then get

$$1 - \mathbb{E}\left[A_{\lambda,kk}\right] \leq 1 - \tilde{A}_{\lambda,kk} + \frac{c}{\lambda^2 N}\left(1 - \tilde{A}_{\lambda,kk}\right)\tilde{A}_{\lambda,kk} \leq \left(1 - \tilde{A}_{\lambda,kk}\right)\left(1 + \frac{c}{\lambda^2 N}\right).$$

We decompose the true function f^* into $f^* = \sum_{k=1}^{\infty} b_k f^{(k)}$ for $b_k = \left\langle f^*, f^{(k)} \right\rangle_S$, and obtain

$$\left| R^\epsilon\left(\mathbb{E}_{\mathcal{O},E}\left[\hat{f}_\lambda^\epsilon\right]\right) - \left(\left\|\left(I_\mathcal{C} - \tilde{A}_{\vartheta(\lambda)}\right) f^*\right\|_S^2 + \epsilon^2\right) \right| = \left| \|(I_\mathcal{C} - \mathbb{E}\left[A_\lambda\right]) f^*\|_S^2 - \left\|\left(I_\mathcal{C} - \tilde{A}_{\vartheta(\lambda)}\right) f^*\right\|_S^2 \right|$$

$$= \left| \sum_{k=1}^{\infty} b_k^2 \left((1 - \mathbb{E}\left[A_{\lambda,kk}\right])^2 - \left(1 - \tilde{A}_{\vartheta(\lambda),kk}\right)^2 \right) \right|$$
$$\leq \sum_{k=1}^{\infty} b_k^2 \left| \tilde{A}_{\vartheta(\lambda),kk} - \mathbb{E}\left[A_{\lambda,kk}\right] \right| \left| 2 - \tilde{A}_{\vartheta(\lambda),kk} - \mathbb{E}\left[A_{\vartheta(\lambda),kk}\right] \right|.$$

By the triangular inequality, we get that

$$\left| 2 - \tilde{A}_{\vartheta(\lambda),kk} - \mathbb{E}\left[A_{\vartheta(\lambda),kk}\right] \right| \leq \left| 1 - \tilde{A}_{\vartheta(\lambda),kk} \right| \left(2 + \left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N} \right) \right) \right)$$

and thus

$$\left| R^{\epsilon}\left(\mathbb{E}_{\mathcal{O},E}\left[\hat{f}_{\lambda}^{\epsilon} \right] \right) - \left(\left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)} \right) f^* \right\|_S^2 + \epsilon^2 \right) \right|$$
$$\leq \sum_{k=1}^{\infty} b_k^2 \left| 1 - \tilde{A}_{\vartheta(\lambda),kk} \right|^2 \left(2 + \frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N} \right) \right) \left(\frac{1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N} \right) \right)$$
$$\leq \sum_{k=1}^{\infty} b_k^2 \left| 1 - \tilde{A}_{\vartheta(\lambda),kk} \right|^2 \left(\frac{\boldsymbol{C}_2}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N} \right) \right).$$

Variance term: For the second term, recall that $(\partial_\lambda \vartheta(\lambda) - 1) = \frac{\partial_\lambda \vartheta(\lambda)}{N} \sum_{k=1}^{\infty} \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}$, and that

$$\left| \sum_{k=1}^{\infty} \mathrm{Var}\left(\left\langle f^{(k)}, \hat{f}_{\lambda}^{\epsilon} \right\rangle_S \right) - (\partial_\lambda \vartheta(\lambda) - 1) \left(\left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)} \right) f^* \right\|_S^2 + \epsilon^2 \right) \right|$$
$$\leq \sum_{k=1}^{\infty} \left| \mathrm{Var}_{\mathcal{O}}\left(\left\langle f^{(k)}, A(-\lambda) f^* \right\rangle_S \right) - \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)} \right) f^* \right\|_S^2 + \epsilon^2 + \left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2} \right) \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \right|$$
$$+ \sum_{k=1}^{\infty} \frac{\partial_\lambda \vartheta(\lambda)}{N} \left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta(\lambda)^2 d_k^2}{(\vartheta(\lambda)+d_k)^4}.$$

Using Theorem D.2, we can control the terms in the first series: there is a constant $\boldsymbol{C}_1 > 0$ such that

$$\sum_{k=1}^{\infty} \left| \mathrm{Var}_{\mathcal{O}}\left(\left\langle f^{(k)}, A(-\lambda) f^* \right\rangle_S \right) - \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)} \right) f^* \right\|_S^2 + \epsilon^2 + \left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2} \right) \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \right|$$
$$\leq \left(\frac{\boldsymbol{C}_1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N^{\frac{1}{2}}} \right) \right) \frac{\partial_\lambda \vartheta(\lambda)}{N} \sum_{k=1}^{\infty} \left(\left\| (I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}) f^* \right\|_S^2 + \epsilon^2 + \left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta^2(\lambda)}{(\vartheta(\lambda)+d_k)^2} \right) \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2}$$
$$\leq \left(\frac{\boldsymbol{C}_1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N^{\frac{1}{2}}} \right) \right) \frac{\partial_\lambda \vartheta(\lambda)}{N} \left(\left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)} \right) \tilde{A}_{\vartheta(\lambda)} f^* \right\|_S^2 + \left(\left\| (I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}) f^* \right\|_S^2 + \epsilon^2 \right) \sum_{k=1}^{\infty} \frac{d_k^2}{(\vartheta(\lambda)+d_k)^2} \right)$$
$$\leq \left(\frac{\boldsymbol{C}_1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N^{\frac{1}{2}}} \right) \right) \left(\frac{\partial_\lambda \vartheta(\lambda)}{N} \left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)} \right) \tilde{A}_{\vartheta(\lambda)} f^* \right\|_S^2 + (\partial_\lambda \vartheta(\lambda) - 1) \left(\left\| (I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}) f^* \right\|_S^2 + \epsilon^2 \right) \right)$$
$$\leq \left(\frac{\boldsymbol{C}_1}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N^{\frac{1}{2}}} \right) \right) \left(\frac{\partial_\lambda \vartheta(\lambda)}{N} \left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)} \right) f^* \right\|_S^2 + (\partial_\lambda \vartheta(\lambda) - 1) \left(\left\| (I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}) f^* \right\|_S^2 + \epsilon^2 \right) \right),$$

whereas for the second series, as explained already above, we have

$$\sum_{k=1}^{\infty} \frac{\partial_\lambda \vartheta(\lambda)}{N} \left\langle f^{(k)}, f^* \right\rangle_S^2 \frac{\vartheta(\lambda)^2 d_k^2}{(\vartheta(\lambda)+d_k)^4} = \frac{\partial_\lambda \vartheta(\lambda)}{N} \left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)} \right) \tilde{A}_{\vartheta(\lambda)} f^* \right\|_S^2 \leq \frac{\partial_\lambda \vartheta(\lambda)}{N} \left\| \left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)} \right) f^* \right\|_S^2 .$$

Finally, putting the pieces together, we conclude. □

Expected Empirical Risk

The expected empirical risk can be approximated as follows:

Theorem D.4. *We have*

$$\left| \mathbb{E}\left[\hat{R}^\epsilon \left(\hat{f}^\epsilon_{\lambda,E} \right) \right] - \frac{\lambda^2}{\vartheta(\lambda)^2} \tilde{R}^\epsilon \left(f^*, \lambda \right) \right| \leq \tilde{R}^\epsilon \left(f^*, \lambda \right) \mathcal{P} \left(\frac{\mathrm{Tr}\left[T_K \right]}{\lambda N} \right).$$

Proof. A small computation allows one to show that:

$$\hat{R}^\epsilon \left(\hat{f}^\epsilon_{\lambda,E} \right) = \frac{\lambda^2}{N} \left(y^\epsilon \right)^T \left(\frac{1}{N} G + \lambda I_N \right)^{-2} y^\epsilon.$$

Using the definition of y^ϵ and the fact that the noise on the labels is centered and independent from the observations, this yields:

$$\begin{aligned} \mathbb{E}\left[\hat{R}^\epsilon \left(\hat{f}^\epsilon_{\lambda,E} \right) \right] &= \frac{\lambda^2}{N} f^* \mathbb{E}\left[\mathcal{O}^T \left(\frac{1}{N} G + \lambda I_N \right)^{-2} \mathcal{O} \right] f^* + \lambda^2 \epsilon^2 \mathbb{E}\left[\frac{1}{N} \mathrm{Tr}\left(\frac{1}{N} G + \lambda I_N \right)^{-2} \right] \\ &= \lambda^2 \sum_{k=1}^{N} \langle f^{(k)}, f^* \rangle_S^2 \frac{\mathbb{E}\left[\partial_\lambda A_{kk}(-\lambda) \right]}{d_k} + \lambda^2 \epsilon^2 \mathbb{E}\left[\partial_z m(-\lambda) \right]. \end{aligned}$$

Similarly to the proof of Theorem D.2, we explain how the approximation of the expected empirical risk appears, then we establish the bounds which allow one to study the quality of this approximation.

Approximations: Using Equation D.2.16, $\mathbb{E}\left[\partial_\lambda A_{kk}(-\lambda) \right] \approx \partial_\lambda \vartheta(\lambda) \frac{d_k}{(\vartheta(\lambda)+d_k)^2}$ hence

$$\begin{aligned} \lambda^2 \sum_{k=1}^{N} \langle f^{(k)}, f^* \rangle_S^2 \frac{\mathbb{E}\left[\partial_\lambda A_{kk}(-\lambda) \right]}{d_k} &\approx \frac{\partial_\lambda \vartheta(\lambda) \lambda^2}{\vartheta(\lambda)^2} \sum_{k=1}^{N} \langle f^{(k)}, f^* \rangle_S^2 \frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2} \\ &= \frac{\partial_\lambda \vartheta(\lambda) \lambda^2}{\vartheta(\lambda)^2} \| (I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}) f^* \|_S^2. \end{aligned}$$

The second term can be approximated using Proposition D.2 and Lemma D.10: this yields

$$\mathbb{E}\left[\partial_\lambda m(-\lambda) \right] \approx \partial_\lambda \tilde{m}(-\lambda) = \frac{\partial_\lambda \vartheta(\lambda)}{\vartheta(\lambda)^2}.$$

Hence, putting the two approximations together, the expected empirical risk is approximated by:

$$\mathbb{E}\left[\hat{R}^\epsilon \left(\hat{f}^\epsilon_{\lambda,E} \right) \right] \approx \frac{\partial_\lambda \vartheta(\lambda) \lambda^2}{\vartheta(\lambda)^2} \left(\| (I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}) f^* \|_S^2 + \epsilon^2 \right) = \frac{\lambda^2}{\vartheta(\lambda)^2} R^\epsilon(f^*, \lambda).$$

Now, we explain how to quantify the quality of the approximations, and thus how to get the bound stated in the theorem. Recall that, we split the expected empirical risk into two terms.

First term: We have already seen in Theorem D.2 that by applying Lemma D.10 to Equation (D.2.12) of Theorem D.1 we get

$$\left|\mathbb{E}\left[\partial_z A_{kk}(-\lambda)\right] - \partial_\lambda \vartheta(\lambda)\frac{d_k}{(\vartheta(\lambda)+d_k)^2}\right| \leq \frac{d_k}{|\vartheta(\lambda)+d_k|^2}\left(\frac{2}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N}\right)\right)$$

and thus

$$\begin{aligned}
&\left|\lambda^2 \sum_{k=1}^N \left\langle f^{(k)}, f^*\right\rangle_S^2 \frac{\mathbb{E}\left[\partial_\lambda A_{kk}(-\lambda)\right]}{d_k} - \frac{\partial_\lambda \vartheta(\lambda)\lambda^2}{\vartheta(\lambda)^2}\left\|\left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}\right) f^*\right\|_S^2\right| \\
&\leq \lambda^2 \sum_{k=1}^N \left\langle f^{(k)}, f^*\right\rangle_S^2 \left|\frac{\mathbb{E}\left[\partial_\lambda A_{kk}(-\lambda)\right]}{d_k} - \frac{\partial_\lambda \vartheta(\lambda)}{\vartheta(\lambda)^2}\frac{\vartheta(\lambda)^2}{(\vartheta(\lambda)+d_k)^2}\right| \\
&\leq \lambda^2 \sum_{k=1}^N \left\langle f^{(k)}, f^*\right\rangle_S^2 \frac{1}{|\vartheta(\lambda)+d_k|^2}\left(\frac{2}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N}\right)\right) \\
&= \frac{\lambda^2}{\vartheta(\lambda)^2}\left\|\left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}\right) f^*\right\|_S^2 \left(\frac{2}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N}\right)\right) \\
&\leq \frac{\partial_\lambda \vartheta(\lambda)\lambda^2}{\vartheta(\lambda)^2}\left\|\left(I_{\mathcal{C}} - \tilde{A}_{\vartheta(\lambda)}\right) f^*\right\|_S^2 \left(\frac{2}{N} + \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N}\right)\right).
\end{aligned}$$

Second Term: Using Proposition D.2 and Lemma D.10:

$$\left|\mathbb{E}\left[\partial_z m(z)\right] - \partial_z \tilde{m}(z)\right| \leq \frac{|z|}{-\Re(z)}\left(\frac{2^3 \mathrm{Tr}\left[T_K\right]}{|z|^3 N^2} + \frac{2^4 \boldsymbol{c}_1 \left(\mathrm{Tr}\left[T_K\right]\right)^2}{|z|^4 N^2} + \frac{2^6 \boldsymbol{c}_1 \left(\mathrm{Tr}\left[T_K\right]\right)^4}{|z|^6 N^4}\right).$$

Thus, since $\partial_\lambda \tilde{m}(-\lambda) = \frac{\partial_\lambda \vartheta(\lambda)}{\vartheta(\lambda)^2}$,

$$\left|\lambda^2 \epsilon^2 \mathbb{E}\left[\partial_\lambda m(-\lambda)\right] - \frac{\lambda^2 \epsilon^2}{\vartheta(\lambda)^2}\partial_\lambda \vartheta(\lambda)\right| \leq \epsilon^2 \mathcal{P}\left(\frac{\mathrm{Tr}[T_K]}{\lambda N}\right).$$

□

Bayesian Setting

In this section, we consider the following Bayesian setting: let the true function f^* be random with zero mean and covariance kernel $\Sigma(x,y) = \mathbb{E}_{f^*}[f^*(x)f^*(y)]$. We will first show that in this setting the KRR predictor with kernel $K = \Sigma$ and ridge $\lambda = \frac{\epsilon^2}{N}$ is optimal amongst all predictors which depend linearly on the noisy labels y^ϵ. Second, given a kernel K and a ridge λ, we provide a simple formula for the expected risk.

Let us consider predictors $\hat{f}$ that depend linearly on the labels y^ϵ, i.e. for all x, there is a $M_x \in \mathbb{R}^N$ such that $\hat{f}(x) = M_x^T y^\epsilon$. Clearly, the KRR predictor belongs to this family of predictors. The pointwise expected squared error can be expressed for any such predictors in terms of the Gram matrix $\mathcal{O}\Sigma\mathcal{O}^T + \epsilon^2 I_N$ and the vector $\mathcal{O}\Sigma(\cdot, x)$

$$\mathbb{E}[(M_x^T y^\epsilon - f^*(x))^2] = M_x^T(\mathcal{O}\Sigma\mathcal{O}^T + \epsilon^2 I_N)M_x - 2M_x^T \mathcal{O}\Sigma(\cdot, x) + \Sigma(x,x).$$

Differentiating w.r.t. M_x, we obtain that the above error is minimized when

$$M_x = \Sigma(x, \cdot)\mathcal{O}^T(\mathcal{O}\Sigma\mathcal{O}^T + \epsilon^2 I_N)^{-1}.$$

In other terms, in this Bayesian setting, the KRR predictor with kernel $K = \Sigma$ and ridge $\lambda = \frac{\epsilon^2}{N}$ minimizes the expected squared error at all points x.

Using Theorem D.3, we obtain the following approximation of the expected risk for a general kernel K and ridge λ:

Corollary 2. *For a random true function of zero mean and covariance kernel Σ the expected risk is approximated by*

$$B(\lambda, K; \epsilon^2, \Sigma) = N\vartheta(\lambda, K) + N\partial_\lambda\vartheta(\lambda, K)(\frac{\epsilon^2}{N} - \lambda) + \partial_\tau\vartheta(\lambda, K + \tau(\Sigma - K))\big|_{\tau=0},$$

in the sense that

$$|\mathbb{E}[R^\epsilon(\hat{f}^\epsilon_\lambda)] - B(\lambda, K; \epsilon^2, \Sigma)| \leq B(\lambda, K; \epsilon^2, \Sigma)\left(\frac{1}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{|z|\, N^{\frac{1}{2}}}\right)\right).$$

Proof. Denoting by $\mathbb{E}$ the expectation taken with respect to the data points and the noise, and by $\mathbb{E}_{f^*}$ the expectation taken with respect to the random true function f^*, from Theorem D.3 we obtain

$$\begin{aligned}\left|\mathbb{E}_{f^*}\left[\mathbb{E}\left[R^\epsilon\left(\hat{f}^\epsilon_\lambda\right)\right]\right] - \mathbb{E}_{f^*}\left[\tilde{R}^\epsilon\left(f^*, \lambda\right)\right]\right| &\leq \mathbb{E}_{f^*}\left[\left|\mathbb{E}\left[R^\epsilon\left(\hat{f}^\epsilon_\lambda\right)\right] - \tilde{R}^\epsilon\left(f^*, \lambda\right)\right|\right] \\ &\leq \mathbb{E}_{f^*}\left[\tilde{R}^\epsilon\left(f^*, \lambda\right)\right]\left(\frac{1}{N} + \mathcal{P}\left(\frac{\operatorname{Tr}[T_K]}{|z|\, N^{\frac{1}{2}}}\right)\right)\end{aligned}$$

it therefore suffices to show that $\mathbb{E}_{f^*}\left[\tilde{R}^\epsilon\left(f^*, \lambda\right)\right] = B(\lambda, K; \epsilon^2, \Sigma)$.

$$\begin{aligned}\mathbb{E}_{f^*}[\tilde{R}^\epsilon(f^*, \lambda, N)] &= \partial_\lambda\vartheta(\lambda)\left(\mathbb{E}_{f^*}\left[\|(I_\mathcal{C} - \tilde{A}_\vartheta)f^*\|_S^2\right] + \epsilon^2\right) \\ &= \partial_\lambda\vartheta(\lambda, K)\left(\operatorname{Tr}\left[T_\Sigma(I_\mathcal{C} - \tilde{A}_\vartheta)^2\right] + \epsilon^2\right) \\ &= \partial_\lambda\vartheta(\lambda, K)\left(\vartheta^2\operatorname{Tr}\left[T_K(T_K + \vartheta(\lambda, K)I_\mathcal{C})^{-2}\right] + \epsilon^2\right) \\ &+ \partial_\lambda\vartheta(\lambda, K)\operatorname{Tr}\left[(T_\Sigma - T_K)(I_\mathcal{C} - \tilde{A}_\vartheta)^2\right].\end{aligned}$$

This formula can be further simplified. First note that differentiating both sides of Equation D.2.2 w.r.t. to λ, we obtain that

$$\frac{\vartheta^2}{N}\operatorname{Tr}\left[T_K(T_K + \vartheta(\lambda, K)I_\mathcal{C})^{-2}\right] = \frac{\vartheta}{\partial_\lambda\vartheta} - \lambda.$$

Secondly, differentiating both sides of Equation D.2.2, we obtain, writing $K(\tau) = K + \tau(\Sigma - K)$

$$\begin{aligned}\partial_\tau\vartheta(\lambda, K(\tau)) &= \frac{\partial_\tau\vartheta}{N}\operatorname{Tr}\left[\tilde{A}_\vartheta\right] + \frac{\vartheta}{N}\operatorname{Tr}\left[\partial_\tau\tilde{A}_\vartheta\right] \\ &= \frac{\partial_\tau\vartheta}{N}\operatorname{Tr}\left[T_K(T_K + \vartheta I_\mathcal{C})^{-1}\right] + \frac{\vartheta^2}{N}\operatorname{Tr}\left[(T_K + \vartheta I_\mathcal{C})^{-1}T_{(\Sigma-K)}(T_K + \vartheta I_\mathcal{C})^{-1}\right]\end{aligned}$$

$$
\begin{aligned}
&- \frac{\partial_\tau \vartheta \vartheta}{N} \mathrm{Tr}\left[T_K (T_K + \vartheta I_{\mathcal{C}})^{-2}\right] \\
&= \frac{\partial_\tau \vartheta}{N} \mathrm{Tr}\left[T_K (T_K + \vartheta I_{\mathcal{C}})^{-1} - \vartheta T_K (T_K + \vartheta I_{\mathcal{C}})^{-2}\right] + \frac{\vartheta^2}{N} \mathrm{Tr}\left[T_{(\Sigma - K)} (T_K + \vartheta I_{\mathcal{C}})^{-2}\right] \\
&= \frac{\partial_\tau \vartheta}{N} \mathrm{Tr}\left[T_K^2 (T_K + \vartheta I_{\mathcal{C}})^{-2}\right] + \frac{\vartheta^2}{N} \mathrm{Tr}\left[T_{(\Sigma - K)} (T_K + \vartheta I_{\mathcal{C}})^{-2}\right] \\
&= \partial_\tau \vartheta - \frac{\partial_\tau \vartheta}{\partial_\lambda \vartheta} + \frac{\vartheta^2}{N} \mathrm{Tr}\left[T_{(\Sigma - K)} (T_K + \vartheta I_{\mathcal{C}})^{-2}\right],
\end{aligned}
$$

where we used the fact that $\frac{1}{N}\mathrm{Tr}\left[T_K^2 (T_K + \vartheta I_{\mathcal{C}})^{-2}\right] = 1 - 1/\partial_\lambda \vartheta$. This implies that

$$
\begin{aligned}
\partial_\tau \vartheta &= \partial_\lambda \vartheta \frac{\vartheta^2}{N} \mathrm{Tr}\left[T_{(\Sigma - K)} (T_K + \vartheta I_{\mathcal{C}})^{-2}\right] \\
&= \partial_\lambda \vartheta(\lambda, K) \mathrm{Tr}\left[(T_\Sigma - T_K)(I_{\mathcal{C}} - \tilde{A}_\vartheta)^2\right].
\end{aligned}
$$

Putting everything together, we obtain that

$$
\mathbb{E}_{f^*}[\tilde{R}^\epsilon(f^*, \lambda, N)] = N\vartheta(\lambda, K) + N\partial_\lambda \vartheta(\lambda, K)(\frac{\epsilon^2}{N} - \lambda) + \partial_\tau \vartheta(\lambda, K + \tau(\Sigma - K))\big|_{\tau = 0}.
$$

□

Technical Lemmas

Matricial observations and Wick formula

For any family $\boldsymbol{A} = \left(A^{(1)}, \ldots, A^{(k)}\right)$ of k square matrices of same size, any permutation $\sigma \in \mathfrak{S}_k$, we define:

$$
\sigma(\boldsymbol{A}) = \prod_{c \text{ cycle of } \sigma} \mathrm{Tr}\left[\prod_{i \in c} A^{(i)}\right],
$$

where the product inside the trace is taken following the order given by the cycle and, by the cyclic property, does not depend on the starting point (see [65]). For example if $k = 4$ and σ is the product of transpositions $(1,3)(2,4)$,

$$
\sigma(\boldsymbol{A}) = \mathrm{Tr}(A^{(1)} A^{(3)}) \mathrm{Tr}(A^{(2)} A^{(4)}).
$$

The number of cycles of σ is denoted by $\mathrm{c}(\sigma)$. The set of permutations without fixed points, i.e. such that $\sigma(i) \neq i$ for any $i \in [1, \ldots, k]$ is denoted by $\mathfrak{S}_k^\dagger$ and the set of permutations with cycles of even size is denoted by $\mathfrak{S}_k^{\mathrm{even}}$.

The following lemma, which is reminiscent of Lemma 4.5 in [8] and which is a rephrasing of Lemma C.3 of [102], is a consequence of Wick's formula for Gaussian random variables and is key to study the g_k and $h_{k,\ell}$.

Lemma D.9. *If $\boldsymbol{A} = \left(A^{(1)}, \ldots, A^{(k)}\right)$ is a family of k square symmetric random matrices of size P independent from a standard Gaussian vector w of size P, we have*

$$
\mathbb{E}\left[\prod_{i=1}^k w^T A^{(i)} w\right] = \sum_{\sigma \in \mathfrak{S}_k} 2^{k - \mathrm{c}(\sigma)} \mathbb{E}\left[\sigma(\boldsymbol{A})\right], \tag{D.2.17}
$$

and,

$$\mathbb{E}\left[\prod_{i=1}^{k}\left(w^T A^{(i)} w - \operatorname{Tr}\left(A^{(i)}\right)\right)\right] = \sum_{\sigma\in\mathfrak{S}_k^\dagger} 2^{k-\mathrm{c}(\sigma)}\mathbb{E}\left[\sigma\left(\boldsymbol{A}\right)\right]. \tag{D.2.18}$$

Furthermore, if w and v are independent Gaussian vectors of size P and independent from $\boldsymbol{A}$, then

$$\mathbb{E}\left[\prod_{i=1}^{k} w^T A^{(i)} v\right] = \sum_{\sigma\in\mathfrak{S}_k^{\mathrm{even}}} \mathbb{E}\left[\sigma\left(\boldsymbol{A}\right)\right]. \tag{D.2.19}$$

Proof. The only differences with Lemma C.3 of [102] are in the r.h.s. and the combinatorial sets used to express the left side. We only prove Equation (H.3.3); Equations (H.3.4) and (D.2.19) can be proven similarly. Let $\boldsymbol{P}_2(2k)$ be the set of pair partitions of $\{1,\dots,2k\}$ and let $p\in\boldsymbol{P}_2(2k)$. Let $p\left[\boldsymbol{A}\right] = \sum_{\substack{p\le \mathrm{Ker}(i_1,\dots,i_{2k})\\ i_1,\dots,i_{2k}\in\{1,\dots,P\}}} \mathbb{E}\left[A^{(1)}_{i_1 i_2}\dots A^{(k)}_{i_{2k-1}i_{2k}}\right]$ where $\le$ is the coarsed order (i.e. $p\le q$ if q is coarser than p) and where for any $i_1,\dots,i_{2k}$ in $1,...,P$, $\mathrm{Ker}(i_1,\dots,i_{2k})$ is the partition of $\{1,\dots,2k\}$ such that two elements u and v in $\{1,...,2k\}$ are in the same block (i.e. pair) of $\mathrm{Ker}(i_1,\dots,i_{2k})$ if and only if $i_u = i_v$. By Wick's formula, we have

$$\mathbb{E}\left[\prod_{i=1}^{k} w^T A^{(i)} w\right] = \sum_{p\in\boldsymbol{P}_2(2k)} p\left[\boldsymbol{A}\right];$$

therefore, it is sufficient to prove that

$$\sum_{p\in\boldsymbol{P}_2(2k)} p\left[\boldsymbol{A}\right] = \sum_{\sigma\in\mathfrak{S}_k} 2^{k-\mathrm{c}(\sigma)}\sigma\left(\boldsymbol{A}\right),$$

Let Po be the set of polygons on $\{1,\dots,k\}$, i.e. the set of collections of non-crossing loops (disjoint unoriented cycles) which cover $\{1,\dots,k\}$. Consider the two maps $F:\boldsymbol{P}_2(2k)\to\mathrm{Po}$ and $G:\mathfrak{S}_k\mapsto\mathrm{Po}$ obtained by forgetting the underlying structure: for any partition $p\in\boldsymbol{P}_2(2k)$, $F(p)$ is the collection of edges (ℓ,m) (viewed as collection of non-crossing loops) such that there exists $u\in\{2\ell-1,2\ell\}$ and $v\in\{2m-1,2m\}$ with $\{u,v\}\in p$; for any permutation $\sigma\in\mathfrak{S}_k$, $G(\sigma)$ is the set of its loops (unoriented cycles).

One can check that for any $\pi\in\mathrm{Po}$,

$$\#\left\{p\in\boldsymbol{P}_2(2k)\mid F(p)=\pi\right\} = 2^{k-\mathrm{c}_{\le 2}(\pi)},\quad \#\left\{\sigma\in\mathfrak{S}_k\mid G(\sigma)=\pi\right\} = 2^{\mathrm{c}(\pi)-\mathrm{c}_{\le 2}(\pi)},$$

where $\mathrm{c}(\pi)$, resp. $\mathrm{c}_{\le 2}(\pi)$, is the number of unoriented cycles, resp. unoriented cycles of size smaller than or equal to 2, of π. Note that $\mathrm{c}(\pi)$, resp. $\mathrm{c}_{\le 2}(\pi)$ are also the number of cycles, resp. cycles of size smaller than or equal to 2 of any σ such that $G(\sigma)=\pi$. Notice also that, since the matrices are symmetric, for any $p,p'\in\boldsymbol{P}_2(2k)$ and any $\sigma\in\mathfrak{S}_k$, if $F(p)=F(p')=G(\sigma)$, then $p\left[\boldsymbol{A}\right]=p'\left[\boldsymbol{A}\right]=\sigma\left[\boldsymbol{A}\right]$. Hence:

$$\sum_{p\in\boldsymbol{P}_2(2k)} p\left[\boldsymbol{A}\right] = \sum_{p\in\boldsymbol{P}_2(2k)}\sum_{\pi=F(p)} p\left[\boldsymbol{A}\right] = \sum_{\pi\in\mathrm{Po}}\sum_{p:F(p)=\pi}\pi\left[\boldsymbol{A}\right] = \sum_{\pi\in\mathrm{Po}} 2^{k-\mathrm{c}_{\le 2}(\pi)}\pi\left[\boldsymbol{A}\right]$$

hence

$$\sum_{p\in\boldsymbol{P}_2(2k)} p\left[\boldsymbol{A}\right] = \sum_{\pi\in\mathrm{Po}} 2^{k-\mathrm{c}_{\le 2}(\pi)}\frac{1}{2^{\mathrm{c}(\pi)-\mathrm{c}_{\le 2}(\pi)}}\sum_{\sigma:G(\sigma)=\pi}\pi\left[\boldsymbol{A}\right] = \sum_{\sigma\in\mathfrak{S}_k} 2^{k-\mathrm{c}(\sigma)}\sigma\left[\boldsymbol{A}\right],$$

as required. □

Bound on derivatives

Given a bound on a holomorphic function, one can obtain a bound on its derivative.

Lemma D.10. *Let $f, g : \mathbb{H}_{<0} \to \mathbb{C}$ be two holomorphic functions such that for any $z \in \mathbb{H}_{<0}$,*

$$|f(z) - g(z)| \leq F(|\, z\,|),$$

where $F : \mathbb{R}^+ \to \mathbb{R}$ is a decreasing function, then for any $z \in \mathbb{H}_{<0}$:

$$|\partial_z f(z) - \partial_z g(z)| \leq \frac{2}{-\Re(z)} F\left(\frac{|\, z\,|}{2}\right),$$

Proof. This is a consequence of Cauchy's inequality: for any $r < -\Re(z)$ (so that the circle of center z and radius r lies inside $\mathbb{H}_{<0}$),

$$|\partial_z f(z) - \partial_z g(z)| \leq \frac{1}{r} \sup_{|w-z|=r} |f(w) - g(w)| \leq \frac{1}{r} \sup_{|w-z|=r} F(|\, w\,|).$$

The inequality follows by considering $r = -\frac{1}{2}\Re(z)$ and using the fact that F is decreasing. □

Generalized Cauchy-Schwarz inequality

Another result that we will use is the following generalization of the Cauchy-Schwarz inequality, which is a consequence of Hölder's inequality.

Lemma D.11. *For complex random variables $a_1, ..., a_s$, we have*

$$\mathbb{E}\left[|a_1 \cdots a_s|\right] \leq \sqrt[s]{\mathbb{E}\left[|a_1|^s\right] \cdots \mathbb{E}\left[|a_s|^s\right]}.$$

Proof. The proof is done using an induction argument. The initialization, i.e. when $s = 1$, is trivial.

For the induction step, assume that the result is true for s terms and let us prove it for $s + 1$ terms. By Hölder's inequality applied for $p = s + 1$ and $q = \frac{s+1}{s}$, we obtain:

$$\begin{aligned}
\mathbb{E}\left[|a_0 a_1 \cdots a_s|\right] &\leq \left(\mathbb{E}\left[|a_0|^{s+1}\right]\right)^{\frac{1}{s+1}} \left(\mathbb{E}\left[|a_1 \cdots a_s|^{\frac{s+1}{s}}\right]\right)^{\frac{s}{s+1}} \\
&\leq \left(\mathbb{E}\left[|a_0|^{s+1}\right]\right)^{\frac{1}{s+1}} \left(\mathbb{E}\left[|a_1|^{s+1}\right] \cdots \mathbb{E}\left[|a_s|^{s+1}\right]\right)^{\frac{1}{s+1}},
\end{aligned}$$

where the second inequality is obtained by the induction hypothesis. □

Control on fixed points

Lemma D.12. *Let $z \in \mathbb{H}_{<0}$, let $(a_k)_k$ and $(b_k)_k$ be sequences of complex numbers in the cone spanned by 1 and $-1/z$ and let $(d_k)_k$ be positive numbers. Then*

$$\left| z - \sum_{k=1}^{\infty} \frac{d_k}{(1 + a_k)(1 + b_k)} \right| \geq |z|\,.$$

Proof. For any complex numbers z_1 and z_2, let Γ_{z_1,z_2} be the cone spanned by z_1 and z_2, i.e. $\Gamma_{z_1,z_2} = \{w \in \mathbb{C} \ : w = az_1 + bz_2 \text{for } a, b \geq 0\}$. Since $a_k, b_k \in \Gamma_{1,-1/z}$, $1/1+a_k$ and $1/1+b_k$ are in $\Gamma_{1,-z}$. All the summands $\frac{d_k}{(1+a_k)(1+b_k)}$ lie in Γ_{1,z^2}, hence so does $\sum_{k=1}^{\infty} \frac{d_k}{(1+a_k)(1+b_k)}$. Since $\Re(z) < 0$, the closest point to z in this cone is 0 and this yields the lower bound:

$$\left| z - \sum_{k=1}^{\infty} \frac{d_k}{(1+a_k)(1+b_k)} \right| \geq |z| \,,$$

hence the result. □

Recall that $\tilde{m}(z)$, resp. $\tilde{m}_{(k)}(z)$, is the unique fixed point of the function $\psi(x) := -\frac{1}{z}\left(1 - \frac{1}{N}\sum_{\ell=1}^{\infty} \frac{d_\ell x}{1+d_\ell x}\right)$, resp. $\psi_{(k)}(x) := -\frac{1}{z}\left(1 - \frac{1}{N}\sum_{\ell\neq k} \frac{d_\ell x}{1+d_\ell x}\right)$, inside the cone spanned by 1 and $-1/z$. We have the following control on the distance between $\tilde{m}(z)$ and $\tilde{m}_{(k)}(z)$.

Lemma D.13. *For any $z \in \mathbb{H}_{<0}$,*

$$\left|\tilde{m}_{(k)}(z) - \tilde{m}(z)\right| \leq \frac{1}{|z|\, N}.$$

Proof. Let $z \in \mathbb{H}_{<0}$, $\tilde{m} = \tilde{m}(z)$ and $\tilde{m}_{(k)} = \tilde{m}_{(k)}(z)$. We have:

$$\begin{aligned}
\tilde{m}_{(k)} - \tilde{m} &= -\frac{1}{z}\left(-\frac{1}{N}\sum_{\ell\neq k} \frac{d_\ell \tilde{m}_{(k)}}{1+d_\ell \tilde{m}_{(k)}} + \frac{1}{N}\sum_{m} \frac{d_\ell \tilde{m}}{1+d_\ell \tilde{m}}\right) \\
&= -\frac{1}{z}\left(\frac{1}{N}\sum_{\ell\neq k}^{\infty} \frac{d_\ell}{\left(1+d_\ell \tilde{m}_{(k)}\right)\left(1+d_\ell \tilde{m}\right)}(\tilde{m} - \tilde{m}_{(k)}) + \frac{1}{N}\frac{d_k \tilde{m}}{1+d_k \tilde{m}}\right)
\end{aligned}$$

which allows us to express the difference $\tilde{m}_{(k)} - \tilde{m}$ as

$$\tilde{m}_{(k)} - \tilde{m} = \frac{\frac{1}{N}\frac{d_k \tilde{m}}{1+d_k \tilde{m}}}{\left(\frac{1}{N}\sum_{\ell\neq k}^{\infty} \frac{d_\ell}{\left(1+d_\ell \tilde{m}_{(k)}\right)\left(1+d_\ell \tilde{m}\right)} - z\right)}.$$

Since $\tilde{m}_{(k)}$ and $\tilde{m}$ lie in the cone spanned by 1 and $-\frac{1}{z}$, from Lemma D.12, we have the lower bound on the norm of the denominator:

$$\left|\frac{1}{N}\sum_{\ell\neq k}^{\infty} \frac{d_\ell}{\left(1+d_\ell \tilde{m}_{(k)}\right)\left(1+d_\ell \tilde{m}\right)} - z\right| \geq |\ z\ | \,.$$

Since $\Re(\tilde{m}) \geq 0$, $|1 + d_k\tilde{m}| \geq |d_k\tilde{m}|$ and hence $\left|\frac{1}{N}\frac{d_k\tilde{m}}{1+d_k\tilde{m}}\right| \leq \frac{1}{N}$. This yields the inequality $\left|\tilde{m}_{(k)}(z) - \tilde{m}(z)\right| \leq \frac{1}{N|z|}$. □

Appendix E

Freeze and Chaos: NTK views on DNN Normalization, Checkerboard and Boundary Artifacts

E.1 Choice of Parametrization

The NTK parametrization for FC-NNs introduced in Section 2 differs slightly from the one commonly used, yet it ensures that the training is consistent as the size of the layers grows. In the standard parametrization, for $\ell = 0..L-1$, the activations are defined by

$$\begin{aligned}\alpha^{(0)}(x) &= x \\ \tilde{\alpha}^{(\ell+1)}(x) &= W^{(\ell)}\alpha^{(\ell)}(x) + b^{(\ell)} \\ \alpha^{(\ell+1)}(x) &= \sigma\left(\tilde{\alpha}^{(\ell+1)}(x)\right).\end{aligned}$$

Let denote by g_θ the output function of the FC-NN thus parametrized, where θ is the concise notation for the vector of free parameters of the FC-NN, and f_θ that of the FC-NN with NTK parametrization. Note the absence of $\frac{1}{\sqrt{n_\ell}}$ in comparison to the NTK parametrization. With LeCun/He initialization [125], the parameters $W^{(\ell)}$ have standard deviation $\frac{1}{\sqrt{n_\ell}}$ (or $\frac{\sqrt{2}}{\sqrt{n_\ell}}$ for the ReLU but this does not change the general analysis). Using this initialization, the activations stay stochastically bounded as the widths of the FC-NN get large. In the forward pass, there is almost no difference between the two parametrizations and for each choice of parameters θ, we can scale down the connection weights by $\frac{\sqrt{1-\beta^2}}{\sqrt{n_\ell}}$ and the bias weights by β to obtain a new set of parameters $\hat{\theta}$ such that

$$f_\theta = g_{\hat{\theta}}.$$

The two parametrizations will exhibit a difference during backpropagation since:

$$\partial_{W_{ij}^{(\ell)}} g_{\hat{\theta}}(x) = \frac{\sqrt{n_\ell}}{\sqrt{1-\beta^2}} \partial_{W_{ij}^{(\ell)}} f_\theta(x), \qquad \partial_{b_j^{(\ell)}} g_{\hat{\theta}}(x) = \frac{1}{\beta} \partial_{b_j^{(\ell)}} f_\theta(x).$$

The NTK is a sum of products of these derivatives over all parameters:

$$\Theta^{(L)} = \Theta^{(L:W^{(0)})} + \Theta^{(L:b^{(0)})} + \Theta^{(L:W^{(1)})} + \Theta^{(L:b^{(1)})} + ... + \Theta^{(L:W^{(L-1)})} + \Theta^{(L:b^{(L-1)})}.$$

Figure E.1.1: Result of two GANs on CelebA. (Left) with Nonlinearity Normalization and (Right) with Batch Normalization. In both cases the discriminator uses a Normalized ReLU.

With our parametrization, all summands converge to a finite limit, while with the Le Cun or He parameterization we obtain

$$\hat{\Theta}^{(L)} = \frac{n_0}{1-\beta^2}\Theta^{(L:W^{(0)})} + \frac{1}{\beta^2}\Theta^{(L:b^{(0)})} + ... + \frac{n_{L-1}}{1-\beta^2}\Theta^{(L:W^{(L-1)})} + \frac{1}{\beta^2}\Theta^{(L:b^{(L-1)})},$$

where some summands, namely the $\left(\frac{n_i}{1-\beta^2}\Theta^{(L:W^{(i)})}\right)_i$, explode in the infinite width limit. One must therefore take a learning rate of order $\frac{1}{\max(n_1,...n_{L-1})}$ [112, 168] to obtain a meaningful training dynamics, but in this case the contributions to the NTK of the first layers connections $W^{(0)}$ and the bias of all layers $b^{(\ell)}$ vanish, which implies that training these parameters has less and less effect on the function as the width of the network grows. As a result, the dynamics of the output function during training can still be described by a modified kernel gradient descent: the modified learning rate compensates for the absence of normalization in the usual parametrization.

The NTK parametrization is hence more natural for large networks, as it solves both the problem of having meaningful forward and backward passes, and to avoid tuning the learning rate, which is the problem that sparked multiple alternative initialization strategies in deep learning [76]. Note that in the standard parametrization, the importance of the bias parameters shrinks as the width gets large; this can be implemented in the NTK parametrization by taking a small value for the parameter β.

E.2 FC-NN Order and Chaos

In this section, we prove the existence of two regimes,'order' and 'chaos', in FC-NNs. First, we improve some results of [42], and study the rate of convergence of the activation kernels as the

depth grows to infinity. In a second step, this allows us to characterise the behavior of the NTK for large depth.

Let us consider a standardized differentiable nonlinearity σ, i.e. satisfying $\mathbb{E}_{x\sim\mathcal{N}(0,1)}\left[\sigma^2(x)\right] = 1$. Recall that the the activation kernels are defined recursively by $\Sigma^{(1)}(x,y) = \frac{1-\beta^2}{n_0}x^T y + \beta^2$ and $\Sigma^{(\ell+1)}(x,y) = (1-\beta^2)\mathbb{L}^{\sigma}_{\Sigma^{(\ell)}}(x,y) + \beta^2$, where $\mathbb{L}^{\sigma}_{\Sigma^{(L)}}$ was introduced in Section 2.2. By induction, for any $x, y \in \mathbb{S}_{n_0}$, $\Sigma^{(\ell+1)}(x,y)$ is uniquely determined by $\rho_{x,y} = \frac{1}{n_0}x^T y$. Defining the two functions $R_\sigma, B_\beta : [-1,1] \to [-1,1]$ by:

$$R_\sigma(\rho) = \mathbb{E}_{v\sim\mathcal{N}\left(0,\begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right)}\left[\sigma(v_0)\sigma(v_1)\right],$$
$$B_\beta(\rho) = \beta^2 + (1-\beta^2)\rho,$$

one can formulate the activation kernels as an alternate composition of B_β and R_σ:

$$\Sigma^{(\ell)}(x,y) = \left(B_\beta \circ R_\sigma\right)^{\circ \ell - 1} \circ B_\beta\left(\rho_{x,y}\right).$$

In particular, this shows that for any $x, y \in \mathbb{S}_{n_0}$, $\Sigma^{(\ell)}(x,y) \leq 1$. Since the activation kernels are obtained by iterating the same function, we first study the fixed points of the composition $B_\beta \circ R_\sigma : [-1,1] \to [-1,1]$. When σ is a standardized nonlinearity, the function R_σ, named the dual of σ, satisfies the following key properties proven in [42]:

1. $R_\sigma(1) = 1$,
2. For any $\rho \in (-1,0)$, $R_\sigma(\rho) > \rho$,
3. R_σ is convex in $[0,1)$,
4. $R'_\sigma(1) = \mathbb{E}\left[\dot\sigma(x)^2\right]$, where R'_σ denotes the derivative of R_σ,
5. $R'_\sigma = R_{\dot\sigma}$.

By definition $B_\beta(1) = 1$, thus 1 is a trivial fixed point: $B_\beta \circ R_\sigma(1) = 1$. This shows that for any $x \in \mathbb{S}_{n_0}$ and any $\ell \geq 1$:

$$\Sigma^{(\ell)}(x,x) = 1.$$

It appears that -1 is also a fixed point of $B_\beta \circ R_\sigma$ if and only if the nonlinearity σ is antisymmetric and $\beta = 0$. From now on, we will focus on the region $(-1,1)$. From the property 2. of R_σ and since B_β is non decreasing, any non trivial fixed point must lie in $[0,1)$. Since $B_\beta \circ R_\sigma(0) > 0$, $B_\beta \circ R_\sigma(1) = 1$ and R_σ is convex in $[0,1)$, there exists a non trivial fixed point of $B_\beta \circ R_\sigma$ if $\left(B_\beta \circ R_\sigma\right)'(1) > 1$ whereas if $\left(B_\beta \circ R_\sigma\right)'(1) < 1$ there is no fixed point in $(-1,1)$. This leads to two regimes shown in [42], depending on the value of $r_{\sigma,\beta} = \left(1-\beta^2\right)\mathbb{E}_{x\sim\mathcal{N}(0,1)}\left[\dot\sigma^2(x)\right]$:

1. "Order" when $r_{\sigma,\beta} < 1$: $B_\beta \circ R_\sigma$ has a unique fixed point equal to 1 and the activation kernels become constant at an exponential rate,
2. "Chaos" when $r_{\sigma,\beta} > 1$: $B_\beta \circ R_\sigma$ has another fixed point $0 \leq a < 1$ and the activation kernels converge to a kernel equal to 1 if $x = y$ and to a if $x \neq y$ and, if the nonlinearity is antisymmetric and $\beta = 0$, it converges to -1 if and only if $x = -y$.

To establish the existence of the two regimes for the NTK, we need the following bounds on the rate of convergence of $\Sigma^{(\ell)}(x,y)$ in the "order" region and on its values in the "chaos" region:

Lemma E.2.1. *If σ is a standardized differentiable nonlinearity,*

If $r_{\sigma,\beta} < 1$, then for any $x, y \in \mathbb{S}_{n_0}$,

$$1 \geq \Sigma^{(\ell)}(x,y) \geq 1 - 2r_{\sigma,\beta}^{\ell-1}(1-\beta^2).$$

If $r_{\sigma,\beta} > 1$, then there exists a fixed point $a \in [0,1)$ of $B_\beta \circ R_\sigma$ such that for any $x, y \in \mathbb{S}_{n_0}$,

$$\left|\Sigma^{(\ell)}(x,y)\right| \leq \max\left\{\left|\beta^2 + \frac{1-\beta^2}{n_0}x^T y\right|, a\right\}.$$

Proof. Let us denote $r = r_{\sigma,\beta}$ and suppose first that $r < 1$. By [42], we know that $R'_\sigma = R_{\dot\sigma}$ and $R_{\dot\sigma}(\rho) \in \left[-\mathbb{E}\left[\dot\sigma(z)^2\right], \mathbb{E}\left[\dot\sigma(z)^2\right]\right]$ where $z \sim \mathcal{N}(0,1)$. From now on, we will omit to specify the distribution asumption on z. The previous equalities and inequalities imply that $R_\sigma(\rho) \geq 1 - \mathbb{E}\left[\dot\sigma(v)^2\right](1-\rho)$, thus we obtain:

$$B_\beta \circ R_\sigma(\rho) \geq \beta^2 + (1-\beta^2)(1 - \mathbb{E}\left[\dot\sigma(z)^2\right](1-\rho)) = 1 - r(1-\rho).$$

By definition, we then have $\Sigma^{(\ell)}(x,y) = (B_\beta \circ R_\sigma)^{\circ \ell-1} \circ B_\beta\left(\frac{1}{n_0}x^T y\right) \geq 1 - 2(1-\beta^2)r^{\ell-1}$. Using the bound $\Sigma^{(\ell)}(x,y) \leq 1$, this proves the first assertion.

When $r > 1$, there exists a fixed point a of $B_\beta \circ R_\sigma$ in $[0,1)$. By a convexity argument, for any ρ in $[a,1)$, $a \leq B_\beta \circ R_\sigma(\rho) \leq \rho$ and because $R_\sigma(\rho)$ is increasing in $[0,1)$, for all $\rho \in [0,a]$, $0 \leq B_\beta \circ R_\sigma(\rho) \leq a$.

For negative ρ, we claim that $|B_\beta \circ R_\sigma(\rho)| \leq B_\beta \circ R_\sigma(|\rho|)$,which entails the second assertion. Since $R_\sigma(\rho) = \sum_{i=0}^\infty b_i\rho^i$ for positive b_is [42], and the composition $B_\beta \circ R_\sigma(\rho) = \sum_{i=0}^\infty c_i\rho^i$ for $c_0 = b_0(1-\beta^2) + \beta^2 \geq 0$ and $c_i = b_i(1-\beta^2) \geq 0$ when $i > 0$, we have

$$|B_\beta \circ R_\sigma(\rho)| = \left|\sum_{i=0}^\infty c_i\rho^i\right| \leq \sum_{i=0}^\infty c_i\,|\rho|^i = B_\beta \circ R_\sigma(|\rho|).$$

This leads to the inequality in the chaos regime. □

Before studying the normalized NTK, let us remark that the NTK on the diagonal (with $x = y$ in $\mathbb{S}_{n_0}$) is equal to:

$$\begin{aligned}\Theta_\infty^{(L)}(x,x) &= \sum_{\ell=1}^L \Sigma^{(\ell)}(x,x)\prod_{k=\ell+1}^L \dot\Sigma^{(k)}(x,x) = \sum_{\ell=1}^L \left((1-\beta^2)\mathbb{E}\left[\dot\sigma(x)^2\right]\right)^{L-\ell} \\ &= \frac{1-r^L}{1-r}.\end{aligned}$$

This shows that in the ordered regime, $\Theta_\infty^{(L)}(x,x) \underset{L\to\infty}{\longrightarrow} \frac{1}{1-r}$ and in the chaotic regime $\Theta_\infty^{(L)}(x,x)$ grows exponentially. At the transition, $r = 1$ and thus $\Theta_\infty^{(L)}(x,x) = L$. Besides, if $x, y \in \mathbb{S}_{n_0}$, using the Cauchy-Schwarz inequality, for any ℓ , $\left|\Sigma^{(\ell)}(x,y)\right| \leq \left|\Sigma^{(\ell)}(x,x)\right|$ and $\left|\dot\Sigma^{(\ell+1)}(x,y)\right| \leq \left|\dot\Sigma^{(\ell+1)}(x,x)\right|$. This implies the following inequality: $\Theta_\infty^{(L)}(x,y) \leq \Theta_\infty^{(L)}(x,x)$.

We now study the normalized NTK $\vartheta_L(x,y) = \frac{\Theta_\infty^{(L)}(x,y)}{\Theta_\infty^{(L)}(x,x)} \leq 1$.

Theorem E.2.2. *Suppose that σ is twice differentiable and standardized.*

If $r < 1$, we are in the ordered regime: there exists C_1 such that for $x, y \in \mathbb{S}_{n_0}$,

$$1 - C_1 L r^L \leq \vartheta^{(L)}(x, y) \leq 1.$$

If $r > 1$, we are in the chaotic regime: for $x \neq y$ in $\mathbb{S}_{n_0}$, there exist $s < 1$ and C_2, such that

$$\left|\vartheta^{(L)}(x, y)\right| \leq C_2 s^L.$$

Proof. First, let us suppose that $r < 1$. Recall that the NTK is defined as

$$\Theta_\infty^{(L)}(x, y) = \sum_{\ell=1}^{L} \Sigma^{(\ell)}(x, y)\dot{\Sigma}^{(\ell+1)}(x, y) \dots \dot{\Sigma}^{(L)}(x, y).$$

Several times in the appendix, we will use the following fact: for any $a_1, \cdots, a_k \in (0, 1)$, we have

$$\prod_{i=1}^{k}(1 - a_i) \geq 1 - \sum_{i=1}^{k} a_i. \tag{E.2.1}$$

For all $\ell = 1..L$, $\Sigma^{(\ell)}(x, y) \leq \Sigma^{(\ell)}(x, x) = 1$ and $\dot{\Sigma}^{(\ell)}(x, y) \leq \dot{\Sigma}^{(\ell)}(x, x) = r$. Writing $\Sigma^{(\ell)}(x, y) = 1 - \epsilon^{(\ell)}$ and $\dot{\Sigma}^{(\ell)}(x, y) = r - \dot{\epsilon}^{(\ell)}$ for $\epsilon^{(\ell)}, \dot{\epsilon}^{(\ell)} \geq 0$, we have that

$$\begin{aligned}
\Theta_\infty^{(L)}(x, y) &= \sum_{\ell=1}^{L} \left(1 - \epsilon^{(\ell)}\right) \prod_{k=\ell+1}^{L} \left(r - \dot{\epsilon}^{(\ell)}\right) \\
&\geq \sum_{\ell=1}^{L} r^{L-\ell} - r^{L-\ell}\epsilon^{(\ell)} - \sum_{k=\ell+1}^{L} r^{L-\ell-1}\dot{\epsilon}^{(\ell)},
\end{aligned}$$

by (E.2.1). Using the bound of Lemma E.2.1 and the fact that for any $x, y \in \mathbb{S}_{n_0}$, $\dot{\Sigma}^{(\ell)}(x, y) = (1-\beta^2)R_{\dot{\sigma}}(\Sigma^{(\ell-1)}(x, y)) \geq r - \psi\epsilon^{(\ell-1)}$ for $\psi = (1-\beta^2)\mathbb{E}_{z\sim\mathcal{N}(0,1)}\left[\ddot{\sigma}(z)\right]$, we obtain $\epsilon^{(\ell)} < 2(1-\beta^2)r^{\ell-1}$ and $\dot{\epsilon}^{(\ell)} \leq 2(1 - \beta^2)\psi r^{\ell-2}$. As a result:

$$\begin{aligned}
\Theta_\infty^{(L)}(x, y) &\geq \sum_{\ell=1}^{L} r^{L-\ell} - 2(1 - \beta^2)r^{L-\ell}r^{\ell-1} - \sum_{k=\ell+1}^{L} 2(1 - \beta^2)\psi r^{L-\ell-1}r^{k-2} \\
&= \Theta_\infty^{(L)}(x, x) - 2(1 - \beta^2)\sum_{\ell=1}^{L} r^{L-1} + \psi \sum_{k=\ell+1}^{L} r^{L-\ell+k-3} \\
&= \Theta_\infty^{(L)}(x, x) - 2(1 - \beta^2)\left[Lr^{L-1} + \psi r^{L-2}\sum_{\ell=1}^{L}\frac{1 - r^{L-\ell}}{1 - r}\right] \\
&\geq \Theta_\infty^{(L)}(x, x) - 2(1 - \beta^2)\left[r + \psi\frac{1}{1 - r}\right] Lr^{L-2} \\
&\geq \Theta_\infty^{(L)}(x, x) - CLr^L.
\end{aligned}$$

Now, let us suppose that $r > 1$. Recall that $B_\beta \circ R_\sigma$ has a unique fixed point a on $[0, 1)$. For any x and y in $\mathbb{S}_{n_0}$, the kernels $\Sigma^{(\ell)}(x, y)$ are bounded in norm by $v = max\left\{\left|\beta^2 + \frac{1-\beta^2}{n_0}x^T y\right|, a\right\}$

from Lemma E.2.1. For the kernels $\dot{\Sigma}^{(\ell)}$ we have $\left|\dot{\Sigma}^{(\ell)}(x,y)\right| = (1-\beta^2)\left|R_{\dot\sigma}(\Sigma^{(\ell-1)}(x,y))\right| \leq (1-\beta^2)R_{\dot\sigma}(\left|\Sigma^{(\ell-1)}(x,y)\right|) \leq (1-\beta^2)R_{\dot\sigma}(v) =: w$ where the first inequality follows from the fact that $R_{\dot\sigma}(\rho) = \sum_i b_i\rho^i$ for $b_i \geq 0$ and the second follows from the monotonicity of $R_{\dot\sigma}$ in $[0,1]$. Applying these two bounds, we obtain:

$$\left|\Theta_\infty^{(L)}(x,y)\right| \leq \sum_{\ell=1}^{L} v \prod_{k=\ell+1}^{L} w = v\frac{1-w^L}{1-w}.$$

Since $\Theta_\infty^{(L)}(x,y) = \frac{1-r^L}{1-r}$, we have that $|\vartheta_L(x,y)| \leq v\frac{1-w^L}{1-r^L}$. If $x \neq y$ then $v < 1$ and since σ is nonlinear, $w = (1-\beta^2)R_{\dot\sigma}(v) < (1-\beta^2)R_{\dot\sigma}(1) = r$. This implies that $|\vartheta_L(x,y)|$ converges to zero at an exponential rate, as $L \to \infty$. □

ReLU FC-NN

For the standardized ReLU nonlinearity, $\sigma(x) = \sqrt{2}\max(x,0)$, the dual activation is computed in [42]:

$$R_\sigma(\rho) = \frac{\sqrt{1-\rho^2} + \left(\pi - \cos^{-1}(\rho)\right)\rho}{\pi},$$

and the dual activation of its derivative is given by:

$$R_{\dot\sigma}(\rho) = \frac{\pi - \cos^{-1}(\rho)}{\pi}.$$

The characteristic value $r = r_{\sigma,\beta}$ of the standardized ReLU is equal to $1-\beta^2$: the ReLU nonlinearity therefore lies in the "order" regime as soon as $\beta > 0$. More explicitly, Lemma E.2.1 still holds of the standardized ReLU and the following inequalities hold for any $x, y \in \mathbb{S}_{n_0}$:

$$1 \geq \Sigma^{(\ell)}(x,y) \geq 1 - 2r^\ell.$$

Using these bounds, we can now prove Theorem E.2.3.

Theorem E.2.3. *With the same notation as in Theorem E.2.2, taking σ to be the standardized ReLU and $\beta > 0$, we are in the weakly ordered regime: there exists a constant C such that $1 - CLr^{L/2} \leq \vartheta^{(L)}(x,y) \leq 1$.*

Proof. The first inequality $\vartheta_L(x,y) \leq 1$ follows the same proof as in the differentiable case.

For the lower bound, using the fact that $(1-\beta)r = 1$, we have $\epsilon^{(\ell)} = 1 - \Sigma^{(\ell)}(x,y) \leq 2r^\ell$ and using the explicit value of $R_{\dot\sigma}(\rho)$, we get that $R_{\dot\sigma}(\rho) \geq 1 - \sqrt{1-\rho}$ which implies that $\dot\epsilon^{(\ell)} = r - \dot{\Sigma}^{(\ell)}(x,y) \leq r\sqrt{2}r^{\frac{\ell-1}{2}}$: using (E.2.1), we write

$$\begin{aligned}\Theta_\infty^{(L)}(x,y) &= \sum_{\ell=1}^{L}\left(1-\epsilon^{(\ell)}\right)\prod_{k=\ell+1}^{L}\left(r - \dot\epsilon^{(k)}\right) \geq \sum_{\ell=1}^{L} r^{L-\ell} - 2r^{L-\ell}r^\ell - \sqrt{2}\sum_{k=\ell+1}^{L} r^{L-\ell-1+\frac{k-1}{2}} \\ &\geq \Theta_\infty^{(L)}(x,x) - 2Lr^L - \sqrt{2}\sum_{\ell=1}^{L} r^{L-\frac{\ell}{2}-1}\sum_{k=0}^{L-\ell-1} r^{\frac{k}{2}}.\end{aligned}$$

Focusing on bounding the double sum from above, we have

$$\sqrt{2}\sum_{\ell=1}^{L} r^{L-\frac{\ell}{2}-1}\sum_{k=0}^{L-\ell-1} r^{\frac{k}{2}} \leq \frac{\sqrt{2}}{1-\sqrt{r}} r^{\frac{L}{2}-1}\sum_{\ell=0}^{L-1} r^{\frac{\ell}{2}} \frac{\sqrt{2}}{1-\sqrt{r}} r^{\frac{L}{2}-1}\frac{1}{1-\sqrt{r}}$$
$$\leq \frac{\sqrt{2}}{r\left(1-\sqrt{r}\right)^2} r^{\frac{L}{2}}$$

Hence, we see that

$$\Theta_{\infty}^{(L)}(x,y) \geq \Theta_{\infty}^{(L)}(x,x) - \left[2Lr^{\frac{L}{2}} - \frac{\sqrt{2}}{r\left(1-\sqrt{r}\right)^2}\right] r^{\frac{L}{2}}.$$

Recall that for any $x \in \mathbb{S}_{n_0}$, $\Theta_{\infty}^{(L)}(x,x) = \frac{1-r^L}{1-r}$ is bounded in L. Dividing the previous inequality by $\Theta_{\infty}^{(L)}(x,x)$ we get: $1 - Cr^{L/2} \leq \vartheta_L(x,y) \leq 1$, as claimed, where the constant C is explicit. □

E.3 Layer Normalization and Nonlinearity Normalization

Layer normalization is asymptotically equivalent to nonlinearity normalization.

With Layer Normalization (LN), the coordinates of the normalized vectors of activations are $\check{\alpha}_j^{(\ell)}(x) = \sqrt{n_\ell}\frac{\alpha_j^{(\ell)}(x)-\mu^{(\ell)}(x)}{||\alpha^{(\ell)}(x)-\underline{\mu}^{(\ell)}(x)||}$, where $\mu^{(\ell)} := \frac{1}{n_\ell}\sum_{i=1}^{n_\ell}\alpha_i^{(\ell)}(x)$ and $\underline{\mu}^{(\ell)} := \begin{pmatrix}\mu^{(\ell)}\\ \vdots \\ \mu^{(\ell)}\end{pmatrix}$. We simplify the notation by making the dependence on x implicite and denote the standardized nonlinearity $\underline{\sigma}(\cdot) := \frac{\sigma(\cdot)-\mathbb{E}(\sigma(Z))}{\sqrt{\mathrm{Var}(\sigma(Z))}}$, where $Z \overset{d}{\sim} \mathcal{N}(0,1)$.

Suppose that $L = 2$, that is we have a single hidden layer after which the LN is applied. More precisely, the output of the network function with LN is $\widetilde{\alpha}^{(2)}(\check{\alpha}^{(1)}(x))$. We rewrite

$$\check{\alpha}^{(1)} = \sqrt{n_1}\frac{\sigma(\widetilde{\alpha}^{(1)}) - \underline{\mu}^{(1)}}{||\sigma(\widetilde{\alpha}^{(1)}) - \underline{\mu}^{(1)}||} = \underline{\sigma}(\widetilde{\alpha}^{(1)})C_1 + C_2,$$
$$\text{where} \qquad C_1 = \sqrt{n_1}\frac{\sqrt{\mathrm{Var}(\sigma(\mathrm{Z}))}}{||\sigma(\widetilde{\alpha}^{(1)}) - \underline{\mu}^{(1)}||}, \quad \text{and} \quad C_2 = \sqrt{n_1}\frac{\mathbb{E}(\sigma(Z)) - \mu^{(1)}}{||\sigma(\widetilde{\alpha}^{(1)}) - \underline{\mu}^{(1)}||}.$$

Note that $C_1 \to 1$ and $C_2 \to 0$ almost surely, as $n_1 \to \infty$. Indeed, since the $\widetilde{\alpha}_i^{(1)}$'s are independent standard Gaussian variables at initialization (recall that we assume that the inputs belong to $\mathbb{S}_{n_0}$), the law of large numbers entails that $\mu^{(1)} \to \mathbb{E}(\sigma(Z))$ almost surely, as $n_1 \to \infty$, and similarly for $\frac{||\sigma(\widetilde{\alpha}^{(1)})-\underline{\mu}^{(1)}||^2}{n_1} \to \mathrm{Var}(\sigma(Z))$.

To show that LN is asymptotically equivalent to centering and standardizing the nonlinearity, we now establish that C_1 and C_2 are constant during training. We have

$$\frac{\partial}{\partial\widetilde{\alpha}_j^{(1)}}||\sigma(\widetilde{\alpha}^{(1)}) - \underline{\mu}^{(1)}|| = \frac{\dot{\sigma}(\widetilde{\alpha}_j^{(1)})\sum_{i=1}^{n_1}(\delta_{ij} - 1/n_1)(\sigma(\widetilde{\alpha}_i^{(1)}) - \mu^{(1)})}{||\sigma(\widetilde{\alpha}^{(1)}) - \underline{\mu}^{(1)}||} = \frac{\dot{\sigma}(\widetilde{\alpha}_j^{(1)})(\sigma(\widetilde{\alpha}_j^{(1)}) - \mu^{(1)})}{||\sigma(\widetilde{\alpha}^{(1)}) - \underline{\mu}^{(1)}||}. \tag{E.3.1}$$

Note that the absolute value of the latter is bounded by $2||\dot{\sigma}||_\infty$. We write $g(t)$ for any function g that depends on the parameters $\theta(t)$ at time $t \geq 0$. Using twice the triangle inequality yields that

$$\begin{aligned}\left|||\sigma(\widetilde{\alpha}^{(1)}(t)) - \underline{\mu}^{(1)}(t)|| - ||\sigma(\widetilde{\alpha}^{(1)}(0)) - \underline{\mu}^{(1)}(0)||\right| &\leq ||\sigma(\widetilde{\alpha}^{(1)}(t)) - \sigma(\widetilde{\alpha}^{(1)}(0))|| + ||\underline{\mu}^{(1)}(t) - \underline{\mu}^{(1)}(0)|| \\ &\leq ||\dot{\sigma}||_\infty \left(\left(\sum_{i=1}^{n_1} (\widetilde{\alpha}_i^{(1)}(t) - \widetilde{\alpha}_i^{(1)}(0))^2 \right)^{1/2} + \frac{1}{\sqrt{n_1}} \sum_{i=1}^{n_1} \left| \widetilde{\alpha}_i^{(1)}(t) - \widetilde{\alpha}_i^{(1)}(0) \right| \right) \leq ct, \end{aligned} \tag{E.3.2}$$

for some constant $c > 0$, where we used that $|\widetilde{\alpha}_i^{(1)}(t) - \widetilde{\alpha}_i^{(1)}(0)| = \mathcal{O}(t/\sqrt{n_1})$, see Appendix A.2 of [105]. Since $||\sigma(\widetilde{\alpha}^{(1)}(0)) - \underline{\mu}^{(1)}(0)|| \sim \sqrt{n_1}$ by the law of large numbers, we can always write $||\sigma(\widetilde{\alpha}^{(1)}(t)) - \underline{\mu}^{(1)}(t)|| > ||\sigma(\widetilde{\alpha}^{(1)}(0)) - \underline{\mu}^{(1)}(0)|| - ct > 0$. Hence, using (E.3.1) then (E.3.2), we get

$$\begin{aligned}\left| \frac{\partial C_1(t)}{\partial \widetilde{\alpha}_j^{(1)}(t)} \right| &= \frac{\sqrt{n_1}\mathrm{Var}(\sigma(Z))}{||\sigma(\widetilde{\alpha}^{(1)}(t)) - \underline{\mu}^{(1)}(t)||^2} \cdot \left| \frac{\dot{\sigma}(\widetilde{\alpha}_j^{(1)}(t))(\sigma(\widetilde{\alpha}_j^{(1)}(t)) - \mu^{(1)}(t))}{||\sigma(\widetilde{\alpha}^{(1)}(t)) - \underline{\mu}^{(1)}(t)||} \right| \\ &\leq \frac{\sqrt{n_1}\mathrm{Var}(\sigma(Z))}{(||\sigma(\widetilde{\alpha}^{(1)}(0) - \underline{\mu}^{(1)}(0))|| - ct)^2} ||\dot{\sigma}||_\infty = \mathcal{O}(1/\sqrt{n_1}), \end{aligned} \tag{E.3.3}$$

by the law of large numbers. The case of C_2 is similar:

$$\begin{aligned}\frac{\partial C_2(t)}{\partial \widetilde{\alpha}_j^{(1)}(t)} &= \frac{-\dot{\sigma}(\widetilde{\alpha}_j^{(1)}(t))}{\sqrt{n_1}||\sigma(\widetilde{\alpha}^{(1)}(t)) - \underline{\mu}^{(1)}(t)||} - \sqrt{n_1} \frac{(\mathbb{E}(\sigma(Z)) - \mu^{(1)}(t))\dot{\sigma}(\widetilde{\alpha}_j^{(1)}(t))(\sigma(\widetilde{\alpha}_j^{(1)}(t)) - \mu^{(1)}(t))}{||\sigma(\widetilde{\alpha}^{(1)}(t)) - \underline{\mu}^{(1)}(t)||^3} \\ &\leq ||\dot{\sigma}||_\infty \left(\frac{1}{n_1} \frac{\sqrt{n_1}}{||\sigma(\widetilde{\alpha}^{(1)}(0)) - \underline{\mu}^{(1)}(0)|| - ct} - \frac{1}{\sqrt{n_1}} \frac{n_1(\mathbb{E}(\sigma(Z)) - \mu^{(1)}(0) + ct)}{(||\sigma(\widetilde{\alpha}^{(1)}(0)) - \underline{\mu}^{(1)}(0)|| - ct)^2} \right) = \mathcal{O}(1/\sqrt{n_1}), \end{aligned} \tag{E.3.4}$$

again by the law of large numbers. For $i = 1, 2$, we now write $\frac{\partial C_i(t)}{\partial t} = \frac{\partial \widetilde{\alpha}_j^{(1)}(t)}{\partial t} \frac{\partial C_i(t)}{\partial \widetilde{\alpha}_j^{(1)}(t)}$ and recall that the first term is changing at rate $\mathcal{O}(1/\sqrt{n_1})$. Therefore, $|C_i(t) - C_i(0)| \leq \mathcal{O}(t/n_1)$. The claim for $L \geq 3$ follows by induction.

Pre-layer normalization has asymptotically no effect.

Normalizing the preactivations has asymptotically no effect on the network at initialization as well as during training. The output of the ℓ-th layer becomes $\check{\alpha}_j^{(\ell)} = \sigma\big(\sqrt{n_\ell} \frac{\widetilde{\alpha}_j^{(\ell)} - \mu^{(\ell)}}{||\widetilde{\alpha}^{(\ell)} - \underline{\mu}^{(\ell)}||}\big)$ where $\mu^{(\ell)}$ and $\underline{\mu}^{(\ell)}$ are computed similarily as before with $\widetilde{\alpha}^{(\ell)}$ in place of $\alpha^{(\ell)}$. As before, we assume $L = 2$ and deduce the general case by induction. We write $\check{\alpha}_j^{(1)} = \sigma(\widetilde{\alpha}_j^{(1)} C_1 + C_2)$, with $C_1 = \sqrt{n_1}/||\widetilde{\alpha}^{(\ell)} - \underline{\mu}^{(\ell)}||$ and $C_2 = -\sqrt{n_1}\mu^{(1)}/||\widetilde{\alpha}^{(\ell)} - \underline{\mu}^{(\ell)}||$. Again, the law of large numbers show that $C_1 \to 1$ and $C_2 \to 0$ almost surely, as $n_1 \to \infty$. Moreover, similarily as (E.3.1) and (E.3.2), we have that

$$\begin{aligned}&\frac{\partial}{\partial \widetilde{\alpha}_j^{(1)}} ||\widetilde{\alpha}^{(1)} - \underline{\mu}^{(1)}|| = \frac{\widetilde{\alpha}_j^{(1)} - \mu^{(1)}}{||\widetilde{\alpha}^{(1)} - \underline{\mu}^{(1)}||}, \\ &\left| ||\widetilde{\alpha}^{(1)}(t) - \underline{\mu}^{(1)}(t)|| - ||\widetilde{\alpha}^{(1)}(0) - \underline{\mu}^{(1)}(0)|| \right| \leq ct, \end{aligned}$$

for some constant $c > 0$. Using the same argument as in (E.3.3) and (E.3.4), one can thus show for $i = 1, 2$ that

$$\left| \frac{\partial C_i(t)}{\partial \widetilde{\alpha}_j^{(1)}} \right| = \mathcal{O}(1/\sqrt{n_1}).$$

We conclude as previously, noting that

$$\frac{\partial \check{\alpha}_j^{(1)}(t)}{\partial t} = \dot{\sigma}\left(\widetilde{\alpha}_j^{(1)}(t)C_1(t) + C_2(t)\right)\left(\frac{\partial \widetilde{\alpha}_j^{(1)}(t)}{\partial t}C_1(t) + \widetilde{\alpha}_j^{(1)}(t)\frac{\partial C_1(t)}{\partial t} + \frac{\partial C_2(t)}{\partial t}\right).$$

E.4 Batch Normalization

If one adds a BatchNorm layer after the nonlinearity of the last hidden layer, we have:

Lemma E.4.1. *Consider a FC-NN with L layers, with a PN-BN after the last nonlinearity. For any $k, k' \in \{1, \dots, n_L\}$ and any parameter θ_p, we have $\sum_{i=1}^N \Theta_{\theta_p}^{(L)}(\cdot, x_i) = \beta^2 \mathrm{Id}_{n_L}$.*

Proof. This is an direct consequence of the definition of the NTK and of the following claim:

Claim. *For a fully-connected DNN with a BatchNorm layer after the nonlinearity of the last hidden layer then $\frac{1}{N}\sum_{i=1}^N \partial_{\theta_p} f_{\theta,k}(x_i)$ is equal to β if θ_p is $b_k^{(L-1)}$, the bias parameter of the last layer, and equal to 0 otherwise.*

The average of $f_{\theta,k}$ on the training set, $\frac{1}{N}\sum_{i=1}^N \partial_{\theta_p} f_{\theta,k}(x_i)$, only depends on the bias of the last layer:

$$\frac{1}{N}\sum_{i=1}^N f_{\theta,k}(x_i) = \frac{\sqrt{1-\beta^2}}{\sqrt{n_{L-1}}} W^{(L-1)} \frac{1}{N}\sum_{i=1}^N \hat{\alpha}^{(L-1)}(x_i) + \beta b_k^{(L-1)} = \beta b_k^{(L-1)}.$$

Thus for any parameter θ_p, $\frac{1}{N}\sum_{i=1}^N \partial_{\theta_p} f_{\theta,k}(x_i) = \partial_{\theta_p}\left(\beta b_k^{(L-1)}\right)$ is equal to β if the parameter is the bias $b_k^{(L-1)}$ and zero otherwise. □

E.5 Graph-based Neural Networks

In this section, we prove the convergence of the NTK at initialization for a general family of DNNs which contain in particular CNNs and DC-NNs. We will consider the Graph-based parametrization introduced in the main.

For each layer $\ell = 0, ..., L$, the neurons are indexed by a position $p \in I_\ell$ and a channel $i = 1, ..., n_\ell$. We may assume that the sets of positions I_ℓ can be any set, in particular any subset of $\mathbb{Z}^D$. For any position $p \in I_{\ell+1}$, we consider a set of parents $P(p) \subset I_\ell$ and we define recursively the set $P^{\circ k}(p) \subset I_{\ell+1-k}$ of ancestors of level k by $P^{\circ k}(p) = \left\{q \mid \exists q' \in P^{\circ k-1}(p), q \in P(q')\right\}$. For each parent $q \in P(p)$, the connections from the position (q, ℓ) to the position $(p, \ell+1)$ are encoded in an $n_\ell \times n_{\ell+1}$ weight matrix $W^{(\ell, q\to p)}$. We define $\chi(q \to p, q' \to p')$ which is equal to 1 if and only if $W^{(\ell,q\to p)}$ and $W^{(\ell,q'\to p')}$ are shared (in the sense that the two matrices are forced to be equal at initialization and during training) and 0 otherwise. It satisfies $\chi(q \to p, q \to p) = 1$ for any neuron p and any $q \in P(p)$ and it is transitive. We will also suppose that for any neuron p and any

$q, q' \in P(p)$, $\chi(q \to p, q' \to p) = \delta_{qq'}$ (i.e. no pair of connections connected to the same neuron p are shared).

In this setting, the activations and preactivations $\alpha^{(\ell)}, \tilde{\alpha}^{(\ell)} \in (\mathbb{R}^{n_\ell})^{I_\ell}$ are recursively constructed using the parent-based NTK parametrization: we set $\alpha^{(0)}(x) = x$ and for $\ell = 0, \dots, L-1$ and any position $p \in I_\ell$:

$$\tilde{\alpha}^{(\ell+1,p)}(x) = \beta b^{(\ell)} + \frac{\sqrt{1-\beta^2}}{\sqrt{|P(p)|\, n_\ell}} \sum_{q \in P(p)} W^{(\ell, q\to p)} x_q, \qquad \alpha^{(\ell+1)}(x) = \sigma\left(\tilde{\alpha}^{(\ell+1)}(x)\right)$$

where σ is applied entry-wise, $\beta \geq 0$ and $|P(p)|$ is the cardinal of $P(p)$. This is a slightly more general formalism than the DC-NNs and it will allow us to obtain simpler formulae which generalize well to other architectures.

Remark. *Notice that the parametrization is slightly different than the traditional one: we divide by $\sqrt{|P(p)|\, n_\ell}$ instead of dividing by $\sqrt{n_\ell \frac{|\omega|}{s_1 \dots s_d}}$. This does not lead to any difference when one consider infinite-sized images as in Section E.6 since in this case the number of parents is constant, equal to $\frac{|\omega|}{s_1 \dots s_d}$. The key difference between the two parametrizations will be investigated in Section E.7.*

Recall, that for a kernel $K : \mathbb{R}^{n_0} \times \mathbb{R}^{n_0} \to \mathbb{R}$, and for any $z_0, z_1 \in \mathbb{R}^{n_0}$, we defined:

$$\mathbb{L}^g_K(z_0, z_1) = \mathbb{E}_{(y_0,y_1)\sim\mathcal{N}\left(0,(K(z_i,z_j))_{i,j=0,1}\right)}\left[g(y_0)\, g(y_1)\right].$$

Proposition E.5.1. *In the above setting, as $n_1 \to \infty$, …,$n_{\ell-1} \to \infty$ sequentially, the preactivations $\left(\tilde{\alpha}_i^{(\ell,p)}(x)\right)_{i=1,\dots,n_\ell, p\in I_\ell}$ of the ℓ^{th} layer converge to a centered Gaussian process with covariance $\Sigma^{(\ell,pp')}(x,y)\delta_{ii'}$, where $\Sigma^{(\ell,pp')}(x,y)$ is defined recursively as*

$$\Sigma^{(1,pp')}(x,y) = \beta^2 + \frac{1-\beta^2}{\sqrt{|P(p)|\,|P(p')|n_0}} \sum_{q\in P(p)} \sum_{q'\in P(p')} \chi(q\to p, q'\to p')\,(x_q)^T y_{q'},$$

$$\Sigma^{(\ell+1,pp')}(x,y) = \beta^2 + \frac{1-\beta^2}{\sqrt{|P(p)|\,|P(p')|}} \sum_{q\in P(p)} \sum_{q'\in P(p')} \chi(q\to p, q'\to p')\mathbb{L}^\sigma_{\Sigma^{(\ell,qq')}}(x,y).$$

Proof. The proof is done by induction on ℓ. For $\ell = 1$ and any $i \in \{1, \dots, n_1\}$, the preactivation

$$\tilde{\alpha}_i^{(1,p)}(x) = \beta b_i^{(0)} + \frac{\sqrt{1-\beta^2}}{\sqrt{|P(p)|\, n_0}} \sum_{q\in P(p)} \left(W_p^{(0,q\to p)} x_q\right)_i$$

is a random affine function of x and its coefficients are centered Gaussian: it is hence a centered Gaussian process whose covariance is easily shown to be equal to $\mathbb{E}\left[\tilde{\alpha}_i^{(1,p)}(x)\tilde{\alpha}_{i'}^{(1,p')}(y)\right] = \Sigma^{(1,pp')}(x,y)\delta_{ii'}$.

For the induction step, we assume that the result holds for the pre-activations of the layer ℓ. The pre-activations of the next layer are of the form

$$\tilde{\alpha}_i^{(\ell+1,p)}(x) = \beta b_i^{(0)} + \frac{\sqrt{1-\beta^2}}{\sqrt{|P(p)|\, n_\ell}} \sum_{q\in P(p)} \left(W^{(\ell,q\to p)}\alpha^{(\ell,q)}(x)\right)_i.$$

Conditioned on the activations $\alpha^{(\ell,q)}$ of the last layer, $\tilde{\alpha}^{(\ell+1,p)}$ is a centered Gaussian process: in other terms, it is a mixture of centered Gaussians with a random covariance determined by the activations of the last layer. The random covariance between $\tilde{\alpha}_{i_0}^{(\ell+1,p_0)}(x)$ and $\tilde{\alpha}_{i_1}^{(\ell+1,p_1)}(y)$ is equal to

$$\beta^2\delta_{i_0i_1} + \frac{1-\beta^2}{\sqrt{|P(p)|\,|P(p')|n_\ell}} \sum_{\substack{q_0 \in P(p_0)\\ q_1 \in P(p_1)}} \sum_{j_0,j_1=1}^{n_\ell} \mathbb{E}\left[W_{i_0j_0}^{(\ell,q_0\to p_0)} W_{i_1j_1}^{(\ell,q_1\to p_1)}\right] \alpha_{j_0}^{(\ell,q_0)}(x)\alpha_{j_1}^{(\ell,q_1)}(y)$$

$$= \delta_{i_0i_1}\left[\beta^2 + \frac{1-\beta^2}{\sqrt{|P(p)|\,|P(p')|}} \sum_{\substack{q_0 \in P(p_0)\\ q_1 \in P(p_1)}} \chi(q_0\to p_0, q_1\to p_1)\frac{1}{n_\ell}\sum_{j=1}^{n_\ell} \sigma\left(\tilde{\alpha}_j^{(\ell,q_0)}(x)\right)\sigma\left(\tilde{\alpha}_j^{(\ell,q_1)}(y)\right)\right],$$

where we used the fact that $\mathbb{E}\left[W_{i_0j_0}^{(\ell,q_0\to p_0)} W_{i_1j_1}^{(\ell,q_1\to p_1)}\right] = \chi(q_0\to p_0, q_1\to p_1)\delta_{i_0i_1}\delta_{j_0j_1}$. Using the induction hypothesis, as $n_1\to\infty, \ldots, n_{\ell-1}\to\infty$ sequentially, the preactivations $\left(\tilde{\alpha}_j^{(\ell,q_0)}(x), \tilde{\alpha}_j^{(\ell,q_1)}(y)\right)_j$ converge to independant centered Gaussian pairs. As $n_\ell\to\infty$, by the law of large numbers, the sum over j along with the $\frac{1}{n_\ell}$ converges to $\mathbb{L}_\sigma^{\Sigma^{(\ell,qq')}}(x,y)$. In this limit, the random covariance of the Gaussian mixture becomes deterministic and as a consequence, the mixture of Gaussian processes tends to a centered Gaussian process with the right covariance. □

Similarly to the activation kernels, one can prove that the NTK converges at initialization.

Proposition E.5.2. *As $n_1\to\infty, \ldots, n_{L-1}\to\infty$ sequentially, the NTK $\Theta^{(L,p_0p_1)}$ of a general convolutional network converges to $\Theta_{\infty,p_0p_1}^{(L)}\otimes \mathrm{Id}_{n_L}$ where $\Theta_\infty^{(L,p_0p_1)}(x,y)$ is defined recursively by:*

$$\Theta_\infty^{(1,p_0p_1)}(x,y) = \Sigma^{(1,p_0p_1)}(x,y),$$

$$\Theta_\infty^{(L,p_0p_1)}(x,y) = \frac{1-\beta^2}{\sqrt{|P(p_0)|\,|P(p_1)|}} \sum_{\substack{q_0 \in P(p_0)\\ q_1 \in P(p_1)}} \chi(q_0\to p_0, q_1\to p_1)\Theta_\infty^{(L-1,q_0q_1)}(x,y)\mathbb{L}_{\Sigma^{(L-1,q_0q_1)}}^{\dot{\sigma}}(x,y) + \Sigma^{(L,p_0p_1)}(x,y).$$

Proof. The proof by induction on L follows the one of [105] for fully-connected DNNs. We present the induction step and assume that the result holds for a general convolutional network with $L-1$ hidden layers. Following the same computations as in [105], the NTK $\Theta_{p_0p_1,jj'}^{(L+1)}(x,y)$ is equal to

$$\frac{1-\beta^2}{\sqrt{|P(p_0)|\,|P(p_1)|n_L}} \sum_{q_0\in P(p_0)}\sum_{q_1\in P(p_1)}\sum_{ii'} \Theta_{ii'}^{(L,q_0q_1)}(x,y)\dot{\sigma}\left(\tilde{\alpha}_i^{(L,q_0)}(x)\right)\dot{\sigma}\left(\tilde{\alpha}_{i'}^{(L,q_1)}(y)\right) W_{ij}^{(L,q_0\to p_0)}W_{i'j'}^{(L,q_1\to p_1)}$$

$$+ \delta_{jj'}\beta^2 + \delta_{jj'}\frac{1-\beta^2}{\sqrt{|P(p_0)|\,|P(p_1)|n_L}} \sum_{q_0\in P(p_0)}\sum_{q_1\in P(p_1)} \chi(q_0\to p_0, q_1\to p_1)\sum_i \alpha_i^{(L,q_0)}(x)\alpha_i^{(L,q_1)}(y)$$

which, by assumption, converges as $n_1 \to \infty, \ldots, n_{L-1} \to \infty$ to

$$\frac{1-\beta^2}{\sqrt{|P(p_0)|\,|P(p_1)|n_L}} \sum_{q_0 \in P(p_0)} \sum_{q_1 \in P(p_1)} \sum_i \Theta_\infty^{(L,q_0q_1)}(x,y)\dot\sigma\left(\tilde\alpha_i^{(L,q_0)}(x)\right)\dot\sigma\left(\tilde\alpha_i^{(L,q_1)}(y)\right) W_{ij}^{(L,q_0\to p_0)} W_{ij'}^{(L,q_1\to p_1)}$$
$$+ \delta_{jj'}\beta^2 + \delta_{jj'}\frac{1-\beta^2}{\sqrt{|P(p_0)|\,|P(p_1)|n_L}} \sum_{q_0 \in P(p_0)} \sum_{q_1 \in P(p_1)} \chi(q_0 \to p_0, q_1 \to p_1) \sum_i \alpha_i^{(L,q_0)}(x)\alpha_i^{(L,q_1)}(y).$$

As $n_L \to \infty$, using the previous results on the preactivations and the law of large number, the NTK converges to

$$\frac{1-\beta^2}{\sqrt{|P(p_0)|\,|P(p_1)|}} \sum_{q_0 \in P(p_0)} \sum_{q_1 \in P(p_1)} \Theta_\infty^{(L,q_0q_1)}(x,y)\mathbb{L}^{\dot\sigma}_{\Sigma^{(L,q_0q_1)}}(x,y)\,\mathbb{E}\left[W_{ij}^{(L,q_0\to p_0)} W_{ij'}^{(L,q_1\to p_1)}\right]$$

$$+ \delta_{jj'}\beta^2 + \delta_{jj'}\frac{1-\beta^2}{\sqrt{|P(p_0)|\,|P(p_1)|}} \sum_{q_0 \in P(p_0)} \sum_{q_1 \in P(p_1)} \chi(q_0 \to p_0, q_1 \to p_1)\mathbb{L}^{\sigma}_{\Sigma^{(L,q_0q_1)}}(x,y)\,,$$

which can be simplified–using the fact that $\mathbb{E}\left[W_{ij}^{(L,q_0\to p_0)} W_{ij'}^{(L,q_1\to p_1)}\right] = \chi(q_0 \to p_0, q_1 \to p_1)\delta_{jj'}$– into:

$$\delta_{jj'}\frac{1-\beta^2}{\sqrt{|P(p_0)|\,|P(p_1)|}} \sum_{q_0 \in P(p_0)} \sum_{q_1 \in P(p_1)} \chi(q_0 \to p_0, q_1 \to p_1)\Theta_\infty^{(L,q_0q_1)}(x,y)\mathbb{L}^{\dot\sigma}_{\Sigma^{(L,q_0q_1)}}(x,y)$$
$$+ \delta_{jj'}\Sigma^{(L+1,p_0p_1)}(x,y),$$

which proves the assertions. □

E.6 DC-NN Order and Chaos

In this section, in order to study the behaviour of DC-NNs in the bulk and to avoid dealing with border effects, studied in Section E.7, we assume that for all layers ℓ there is no border, i.e. the positions p are in $\mathbb{Z}^d$. Let us consider a DC-NN with up-sampling $s \in \{2,3,...\}^d$ where the window sizes for all layers are all set equal to $\pi = \omega = \{0,\cdots,w_1s_1-1\} \times \cdots \times \{0,\cdots,w_ds_d-1\}$. A position p has therefore $w_1 \cdots w_d$ parents which are given by

$$P(p) = \left\{\left\lfloor\frac{p_0}{s_0}\right\rfloor, \left\lfloor\frac{p_0}{s_0}\right\rfloor + 1, \cdots, \left\lfloor\frac{p_0}{s_0}\right\rfloor + w_1\right\} \times \cdots \times \left\{\left\lfloor\frac{p_d}{s_d}\right\rfloor, \left\lfloor\frac{p_d}{s_d}\right\rfloor + 1, \cdots, \left\lfloor\frac{p_d}{s_d}\right\rfloor + w_d\right\}.$$

Two connections $q \to p$ and $q' \to p'$ are shared if and only if $s \mid p - p'$ (i.e. for any $i = 1, ..., d$, $s_i \mid p_i - p_i'$) and $q_i - q_i' = \frac{p_i - p_i'}{s_i}$ for any $i = 1, ..., d$.

Propositions E.5.1 and E.5.2 hold true in this setting. By Proposition E.7.2, if the nonlinearity σ is standardized, $\Sigma^{(\ell,pp)}(x,x) = 1$ for any $x \in \mathbb{S}_{n_0}^{I_0}$ and any $p \in I_\ell$. The activation kernels $\Sigma^{(\ell,pp')}(x,y)$

for any two inputs $x, y \in \mathbb{S}_{n_0}^{I_0}$ and two output positions $p, p' \in \mathbb{Z}^d$ are therefore defined recursively by:

$$\Sigma^{(1,pp')}(x,y) = \beta^2 + \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|\, n_0} \sum_{q \in P(p)} (x_q)^T\, y_{q+\frac{p'-p}{s}},$$

$$\Sigma^{(\ell+1,pp')}(x,y) = \beta^2 + \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|} \sum_{q \in P(p)} R_\sigma\left(\Sigma^{(\ell,q,q+\frac{p'-p}{s})}(x,y)\right),$$

where $\frac{p'-p}{s} = \left(\frac{p'_i-p_i}{s_i}\right)_i$ is a valid position since $s|p-p'$. Similarly, the NTK at initialization satisfies the following recursion:

$$\Theta_\infty^{(L+1,pp')}(x,y) = \Sigma^{(L+1,pp')}(x,y) + \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|} \sum_{q \in P(p)} \Theta_\infty^{(L,q,q+\frac{p'-p}{s})}(x,y) R_{\dot\sigma}\left(\Sigma^{(L,q,q+\frac{p'-p}{s})}(x,y)\right).$$

Remark. *Recall that the s-valuation $v_s(n)$ of a number $n \in \mathbb{Z}^d$ is the largest $k \in \{0,1,2,\ldots\}$ such that $s_i^k \mid n_i$ for all dimensions $i = 1, ..., d$. For two pixels $p, p' \in \mathbb{Z}^d$ and any input vectors $x, y \in \mathbb{S}_{n_0}^{I_0}$, if $v_s(p'-p) < \ell$ the activation kernel $\Sigma^{(\ell,pp')}(x,y)$ does not depend neither on x nor on y. More precisely, if $v = v_s(p'-p) = 0$, we have*

$$\Sigma^{(\ell,pp')}(x,y) = \beta^2,$$

and for a general $v < \ell$:

$$c_v := \Sigma^{(\ell,pp')}(x,y) = (B_\beta \circ R_\sigma)^{\circ v}(\beta^2).$$

In particular, if $v < L$, the NTK is therefore also equal to a constant:

$$\Theta_\infty^{(L,pp')}(x,y) = \sum_{k=0}^{v} c_k (1-\beta^2)^k \prod_{m=0}^{k-1} R_{\dot\sigma}(c_m).$$

We establish the bounds on the rate of convergence in the "order" region and on the values of the activations kernel in the chaos region for DC-NNs.

Proposition E.6.1. *In the setting introduced above, for a standardized twice differentiable σ, for $x, y \in \mathbb{S}_{n_0}^{I_0}$, and any positions $p, p' \in I_\ell$, taking $k = \min\{v_s(p'-p), \ell\}$, we have:*

If $r_{\sigma,\beta} < 1$ then:

$$1 \geq \Sigma^{(\ell,pp')}(x,y) \geq 1 - 2(1-\beta^2) r_{\sigma,\beta}^k.$$

If $r_{\sigma,\beta} > 1$ then there exists a fixed point $a \in [0,1)$ of $B_\beta \circ R_\sigma$ such that:

- *If $k < \ell$:*
$$\left|\Sigma^{(\ell,pp')}(x,y)\right| \leq \max\left\{\beta^2, a\right\},$$
- *If $p'-p = ms^\ell$ and there is a $c \leq 1$ such that for all input positions $q \in P^{\circ\ell}(p)$, $\left|\frac{1}{n_0} x_q^T y_{q+m}\right| \leq c$, then*
$$\left|\Sigma^{(\ell,pp')}(x,y)\right| \leq \max\left\{\beta^2 + (1-\beta^2)c, a\right\}.$$

Proof. Let us denote $r = r_{\sigma,\beta}$. Let us suppose that $r < 1$ and let us prove the first assertion by induction on ℓ. If $\ell = 1$, then

$$\begin{aligned}\Sigma^{(1,pp')}(x,y) &= \beta^2 + \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|\, n_0} \sum_{q \in P(p)} (x_q)^T y_{q+\frac{p'-p}{s}} \geq \beta^2 - \delta_{s|p-p'}(1-\beta^2) \\ &\geq 1 - 2(1-\beta^2)\end{aligned}$$

For the induction step, suppose that the inequality holds true for some $\ell \geq 1$, then

$$\begin{aligned}\Sigma^{(\ell+1,pp')}(x,y) &\geq \beta^2 + \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|} \sum_{q=0}^{\frac{w}{s}} R_\sigma\left(1 - 2(1-\beta^2) r^{k-1}\right) \\ &\geq \beta^2 + \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|} \sum_{q=0}^{\frac{w}{s}} 1 - 2(1-\beta^2) R_{\dot\sigma}(1) r^{k-1} \\ &\geq \beta^2 + \delta_{s|p-p'} \left(1 - \beta^2 - 2(1-\beta^2) r^k\right) \\ &= \begin{cases} 1 - (1-\beta^2) & \text{if } k = 0 \\ 1 - 2(1-\beta^2) r^k & \text{if } k > 0 \end{cases} \\ &\geq 1 - 2(1-\beta^2) r^k\end{aligned}$$

Now let us suppose that $r > 1$. If $k < \ell$, then $\left|\Sigma^{(\ell,pp')}(x,y)\right| = \left|(B_\beta \circ R_\sigma)^{\circ k}\left(\beta^2\right)\right| < \max\left\{\beta^2, a\right\}$. Let us suppose at last that $k = \ell$ and let us prove the last assertion by induction on ℓ. If $\ell = 1$, then

$$\begin{aligned}\left|\Sigma^{(1,pp')}(x,y)\right| &\leq \beta^2 + \frac{1-\beta^2}{|P(p)|\, n_0} \sum_{q \in P(p)} \left|x_q^T y_{q+\frac{p'-p}{s}}^T\right| \leq \beta^2 + \frac{1-\beta^2}{|P(p)|} \sum_{q \in P(p)} c \\ &= \beta^2 + (1-\beta^2) c.\end{aligned}$$

For the induction step, if we suppose that the inequality holds true for ℓ, then

$$\begin{aligned}\left|\Sigma^{(\ell+1,pp')}(x,y)\right| &\leq \beta^2 + \frac{(1-\beta^2)}{|P(p)|} \sum_{q \in P(p)} \left|R_\sigma\left(\Sigma^{(\ell,q,q+\frac{p'-p}{s})}(x,y)\right)\right| \\ &\leq \beta^2 + \frac{(1-\beta^2)}{|P(p)|} \sum_{q \in P(p)} R_\sigma\left(\max\{\beta^2 + (1-\beta^2)c, a\}\right) \\ &= B_\beta \circ R_\sigma\left(\max\{\beta^2 + (1-\beta^2)c, a\}\right) \\ &\leq \max\{\beta^2 + (1-\beta^2)c, a\},\end{aligned}$$

which allows us to conclude. □

The NTK features the same two regimes:

Theorem E.6.2. *Take $I_0 = \mathbb{Z}^d$, and consider a DC-NN with upsampling stride $s \in \{2,3,\ldots\}^d$, windows $\pi = \omega = \{0,\ldots,w_1 s_1 - 1\} \times \ldots \times \{0,\ldots,w_d s_d - 1\}$ for $w \in \{1,2,3,\ldots\}^d$. For a standardized twice differentiable σ, there exist constants $C_1, C_2 > 0$, such that the following holds: for $x, y \in \mathbb{S}_{n_0}^{I_0}$, and any positions $p, p' \in I_L$, we have:*

Order: *When $r_{\sigma,\beta} < 1$, taking $v = \min\left(v_s\left(p-p'\right), L-1\right)$, taking $v = L-1$ if $p = p'$ and $r = r_{\sigma,\beta}$, we have*

$$\frac{1-r^{v+1}}{1-r^L} - C_1(v+1)r^v \leq \vartheta_\infty^{\left(L,p,p'\right)}(x,y) \leq \frac{1-r^{v+1}}{1-r^L}.$$

Chaos: *When $r_{\sigma,\beta} > 1$, if either $v_s\left(p-p'\right) < L$ or if there exists a $c < 1$ such that for all positions $q \in I_0$ which are ancestor of p, $\left|x_q^T y_{q+\frac{p'-p}{s^L}}\right| < c$, then there exists $h < 1$ such that*

$$\left|\vartheta_\infty^{\left(L,p,p'\right)}(x,y)\right| \leq C_2 h^L.$$

Proof. Let us denote $r = r_{\sigma,\beta}$ and let us suppose that $r < 1$. The NTK can be bounded recursively

$$\begin{aligned}\Theta_\infty^{(L,pp')}(x,y) &= \Sigma^{(L,pp')}(x,y) + \delta_{s|p-p'}\frac{1-\beta^2}{|P(p)|}\sum_{q\in P(p)}\Theta_\infty^{(L-1;q,q+\frac{p'-p}{s})}(x,y)R_{\dot\sigma}\left(\Sigma^{(L-1;q,q+\frac{p'-p}{s})}(x,y)\right)\\ &\geq 1-2(1-\beta^2)r^v + \delta_{s|p-p'}\frac{1}{|P(p)|}\sum_{q\in P(p)}\Theta_\infty^{(L-1;q,q+\frac{p'-p}{s})}(x,y)\left(r-\psi 2(1-\beta^2)^2 r^{v-1}\right).\end{aligned}$$

Unrolling this inequality then using (E.2.1), we get

$$\begin{aligned}\Theta_\infty^{(L,pp')}(x,y) &= \sum_{k=0}^{v}\left(1-2(1-\beta^2)r^k\right)\prod_{m=k+1}^{v}\left(r-\psi 2(1-\beta^2)^2 r^{m-1}\right)\\ &\geq \sum_{k=0}^{v} r^{v-k} - 2(1-\beta^2)r^{v-k}r^k - \psi 2(1-\beta^2)^2\sum_{m=k+1}^{v} r^{v-k-1}r^{m-1}\\ &= \frac{1-r^{v+1}}{1-r} - 2(1-\beta^2)(v+1)r^v - \psi 2(1-\beta^2)^2\sum_{k=0}^{v} r^{v-1}\sum_{m=0}^{v-k-1} r^m\\ &\geq \frac{1-r^{v+1}}{1-r} - 2(1-\beta^2)\left[r+\frac{\psi(1-\beta^2)}{1-r}\right](v+1)r^{v-1}\\ &\geq \frac{1-r^{v+1}}{1-r} - C(v+1)r^v,\end{aligned}$$

where the constant C is allowed to depend on σ and β. For the upper bound, we have: $\Theta_\infty^{(L,pp')}(x,y) \leq \sum_{\ell=L-k}^{L} 1\prod_{m=\ell+1}^{L} r = \frac{1-r^{v+1}}{1-r}$. Thus, we get the same bounds as in the FC-NNs case, but with respect to v, which is the maximal integer strictly smaller than L such that $s^v|p-p'$:

$$\frac{1-r^{v+1}}{1-r} \geq \Theta_\infty^{(L,pp')}(x,y) \geq \frac{1-r^{v+1}}{1-r} - C(v+1)r^v.$$

Dividing by $\Theta_\infty^{(L,pp)}(x,x)$ which is bounded in the ordered regime (see proof of Proposition E.7.2) as $L \to \infty$, one gets the desired result.

If $r > 1$, there are two cases. When $p'-p = ks^L$ then if there exists $c < 1$ such that $\left|x_q^T y_{q+k}\right| < cn_0$ for all ancestors q of p. Writing $z = \max\{\beta^2 + (1-\beta^2)c, a\}$ and $w = (1-\beta^2)R_{\dot\sigma}(z) < r$ such

that $\left|\Sigma^{(\ell;q,q+ks^\ell)}(x,y)\right| < z$ for all position q at layer ℓ which is an ancester of p. Then

$$\left|\Theta_\infty^{(L,pp')}(x,y)\right| \leq \sum_{\ell=1}^{L} vw^{L-\ell} = v\frac{1-w^L}{1-w}$$

such that

$$\frac{\left|\Theta_\infty^{(L,pp')}(x,y)\right|}{\left|\Theta_\infty^{(L,pp)}(x,x)\right|} \leq c\frac{1-r}{1-w}\frac{1-w^L}{1-r^L} \leq C(\sigma,\beta)\left(\frac{w}{r}\right)^L$$

which goes to zero exponentially.

If $p'-p$ is not divisible by s^L then for $z=\max\{\beta^2,a\}$ and $w=(1-\beta^2)R_{\dot\sigma}(z)<r$

$$\left|\Theta_\infty^{(L,pp')}(x,y)\right| \leq \sum_{\ell=L-v+1}^{L} zw^{L-\ell} = z\frac{1-w^v}{1-w}$$

which also converges exponentially to 0. □

Adapting the learning rate

Let us suppose that we multiply the learning rate of the ℓ-th layer weights and bias by $S^{-\frac{\ell}{2}}$ where $S=\prod_i s_i$. This is slightly different than what we propose in the main, where the learning rate of the bias are multiplied by $S^{-\frac{\ell+1}{2}}$ instead of $S^{-\frac{\ell}{2}}$, but it greatly simplifies the formulas. Furthermore, the balance between the weights and bias can be modified with the meta-parameter β to achieve a similar result. The NTK then takes the value:

$$\Theta^{(L,pp)}(x,x) = \sum_{\ell=1}^{L} S^{-\frac{\ell}{2}} \prod_{n=\ell+1}^{L} r = \sum_{\ell=1}^{L} S^{-\frac{\ell}{2}} r^{L-\ell} = S^{-\frac{L}{2}} \frac{1-\left(\sqrt{S}r\right)^L}{1-\sqrt{S}r}$$

This leads to another transtion inside the "order" regime: if $\sqrt{S}r<1$ the NTK $\Theta_\infty^{(L,pp)}(x,x)$ goes to zero and if $\frac{1}{\sqrt{S}}<r<1$ it converges to a constant. If we translate the bound of Proposition E.6.2 to the NTK with varying learning rates, the convergence to a constant is only guaranteed when $\sqrt{S}r<1$, which suggests that adapting the learning (or changing the number of channels) does reduce the checkerboard artifacts (as confirmed by numerical experiments):

Proposition E.6.3. *If $r<1$ the limiting NTK at any two inputs x,y such that for all $p\in\mathbb{Z}$, $\|x^p\|=\|y^p\|=\sqrt{n_0}$ and for any two output positions p and p', such that k is the maximal integer in $\{0,...,L-1\}$ such that s^k divides the difference $p-p'$ then:*

$$\frac{1-(\sqrt{S}r)^{k+1}}{1-(\sqrt{S}r)^L} \geq \vartheta_\infty^{(L,pp')}(x,y) \geq \frac{1-(\sqrt{S}r)^{k+1}}{1-(\sqrt{S}r)^L} - \frac{C_{\sigma,\beta}(\sqrt{S}r)^k}{\left|1-(\sqrt{S}r)^L\right|}$$

Proof. The NTK can be bounded recursively

$$\Theta_\infty^{(L,pp')}(x,y) = S^{-\frac{L-1}{2}}\Sigma^{(L,pp')}(x,y) + \delta_{s|p-p'}\frac{1-\beta^2}{|P(p)|}\sum_{q\in P(p)}\Theta_\infty^{(L-1;q,q+\frac{p'-p}{s})}(x,y)R_{\dot\sigma}\left(\Sigma^{(L-1;q,q+\frac{p'-p}{s})}(x,y)\right)$$

$$\geq S^{-\frac{L-1}{2}}(1-2(1-\beta^2)r^k) + \delta_{s|p-p'}\frac{1}{|P(p)|}\sum_{q\in P(p)}\Theta_\infty^{(L;q,q+\frac{p'-p}{s})}(x,y)\left(r-\psi 2(1-\beta^2)^2 r^{k-1}\right)$$

unrolling then using (E.2.1), we get

$$\begin{aligned}
\Theta_\infty^{(L,pp')}(x,y) &\geq \sum_{m=0}^{k} S^{-\frac{L-k+m}{2}}\left(1-2(1-\beta^2)r^m\right)\prod_{n=m+1}^{k}\left(r-\psi 2(1-\beta^2)^2 r^{n-1}\right)\\
&\geq \sum_{m=0}^{k} S^{\frac{k-m-L}{2}} r^{k-m} - S^{\frac{k-m-L}{2}} 2(1-\beta^2) r^{k-m} r^m - S^{\frac{k-m-L}{2}}\psi 2(1-\beta^2)^2 \sum_{n=m+1}^{k} r^{k-m-1} r^{n-1}\\
&\geq S^{-\frac{L}{2}}\frac{1-(\sqrt{S}r)^{k+1}}{1-\sqrt{S}r} - 2\frac{1-\beta^2}{1-S^{-\frac{1}{2}}} S^{\frac{k-L}{2}} r^k - \psi 2(1-\beta^2)^2 r^{k-1}\sum_{m=0}^{k} S^{\frac{k-m-L}{2}}\sum_{n=0}^{k-m-1} r^n
\end{aligned}$$

We can bound the last term:

$$\psi 2(1-\beta^2)^2 r^{k-1}\sum_{m=0}^{k} S^{\frac{k-m-L}{2}}\sum_{n=0}^{k-m-1} r^n \leq \psi 2(1-\beta^2)^2 r^{k-1} S^{\frac{k-L}{2}}\frac{1}{1-S^{-\frac{1}{2}}}\frac{1}{1-r}$$

Hence, we write

$$\begin{aligned}
\Theta_\infty^{(L,pp')}(x,y) &\geq S^{-\frac{L}{2}}\left(\frac{1-(\sqrt{S}r)^{k+1}}{1-\sqrt{S}r} - 2\frac{1-\beta^2}{1-S^{-\frac{1}{2}}}\left[1+\frac{\psi r(1-\beta^2)}{1-r}\right]\left(\sqrt{S}r\right)^k\right)\\
&\geq S^{-\frac{L}{2}}\left(\frac{1-(\sqrt{S}r)^{k+1}}{1-\sqrt{S}r} - C_{\sigma,\beta}\left(\sqrt{S}r\right)^k\right).
\end{aligned}$$

For the upper bound, we have that

$$\Theta_\infty^{(L,pp')}(x,y) \leq \sum_{m=0}^{k} S^{-\frac{L-k+m}{2}}\prod_{n=m+1}^{k} r = S^{-\frac{L}{2}}\frac{1-(\sqrt{S}r)^{k+1}}{1-\sqrt{S}r}.$$

Dividing by $\Theta_\infty^{(L,pp)}(x,x)$ we obtain

$$\frac{1-(\sqrt{S}r)^{k+1}}{1-(\sqrt{S}r)^{L}} \geq \vartheta_\infty^{(L,pp')}(x,y) \geq \frac{1-(\sqrt{S}r)^{k+1}}{1-(\sqrt{S}r)^{L}} - \frac{C_{\sigma,\beta}(\sqrt{S}r)^k}{\left|1-(\sqrt{S}r)^{L}\right|},$$

as claimed. □

E.7 Border Effects

With the usual scaling of $\frac{1}{\sqrt{\frac{|\omega|}{s_1\cdots s_d}}}$, in a General ConvNet, the positions on the border have less parents and hence a lower activation variance. In this section, we show, in a special example, how this parametrization leads to border effects in the limiting activation kernels and NTK. This could

be generalized to a more general setting, yet, our main purpose is to show that with the parent-based parametrization–as defined in Section E.5–no border artifact is present in both kernels in this general setting.

The following proposition illustrates the border artifact present in the usual NTK-parametrization. Let us consider a DC-NN with a standardized ReLU nonlinearity, with $I_0 = I_1 \ldots = \mathbb{N}$, with upsampling stride of 2, and windows $\pi_0 = \omega_0 = \pi_1 = \omega_1 = \ldots = \{-3, -2, -1, 0\}$. In particular, there is only one border at position 0. Using the formalism of Section E.5, the set of parents of a position p is $P(p) = \{\lfloor \frac{p}{2} \rfloor - 1, \lfloor \frac{p}{2} \rfloor\} \cap \mathbb{N}$. In particular, any generic position in any hidden or last layer has 2 parents except for the border $p = 0$ for which $P(0) = \{0\}$.

Proposition E.7.1. *In the setting introduced above, for any $x \in \mathbb{S}^{I_0}_{n_0}$, the kernels satisfy:*

$$\Sigma^{(\ell,00)}(x,x) = \frac{\beta^2 + \left(\frac{r}{2}\right)^{\ell+1}}{1 - \frac{r}{2}} \text{ and } \Theta^{(L,00)}_\infty(x,x) = \frac{\beta^2(1 - \left(\frac{r}{2}\right)^L)}{\left(1 - \frac{r}{2}\right)^2} + L\frac{\left(\frac{r}{2}\right)^{L+1}}{1 - \frac{r}{2}}.$$

In particular $\Sigma^{(\ell,00)}(x,x)$ is smaller than the "bulk-value" $\lim_{p\to\infty} \Sigma^{(\ell,pp)}(x,x) = 1$ and $\Theta^{(L,00)}_\infty(x,x)$ is smaller than the "bulk-value" $\lim_{p\to\infty} \Theta^{(L,pp)}_\infty(x,x) = \frac{1-r^L}{1-r}$.

Proof. Recall that for the standardized ReLU, $r_{\sigma,\beta} = 1 - \beta^2$. From now on, we denote $r = r_{\sigma,\beta}$ and x is an element of $\mathbb{S}^{I_0}_{n_0}$. For any $\ell = 0, 1 \ldots$, we have:

$$\Sigma^{(\ell+1,00)}(x,x) = \beta^2 + \frac{1-\beta^2}{2} \sum_{q\in P(0)} \mathbb{E}_{z\sim\mathcal{N}(0,\Sigma^{(\ell)}_{qq}(x,x))}\left[\sigma(x)^2\right] = \beta^2 + \frac{1-\beta^2}{2}\Sigma^{(\ell,00)}(x,x).$$

Since $x \in \mathbb{S}^{I_0}_{n_0}$, we get $\Sigma^{(1)}(x,x) = \beta^2 + \frac{r}{2}$: this implies the following equalities:

$$\begin{aligned}\Sigma^{(\ell,00)}(x,x) &= \left(\frac{r}{2}\right)^\ell + \sum_{k=0}^{\ell-1} \beta^2 \left(\frac{r}{2}\right)^k = \left(\frac{r}{2}\right)^\ell + \beta^2 \frac{1 - \left(\frac{r}{2}\right)^\ell}{1 - \frac{r}{2}} \\ &= \frac{\beta^2}{1 - \frac{r}{2}} + \frac{\left(\frac{r}{2}\right)^\ell - \left(\frac{r}{2}\right)^{\ell+1} - \beta^2\left(\frac{r}{2}\right)^\ell}{1 - \frac{r}{2}} = \frac{\beta^2 + \left(\frac{r}{2}\right)^{\ell+1}}{1 - \frac{r}{2}}.\end{aligned}$$

For the limiting NTK, with the usual NTK parametrization, the following recursion holds:

$$\Theta^{(L+1,00)}_\infty(x,x) = \Sigma^{(L+1,00)}(x,x) + \frac{r}{2}\Theta^{(L,00)}_\infty(x,x)\mathbb{L}^{\dot\sigma}_{\Sigma^{(L,00)}}(x,x).$$

Note that for the standardized ReLU, $\dot\sigma$ is a rescaled Heaviside, thus

$$\mathbb{L}^{\dot\sigma}_{\Sigma^{(L,00)}}(x,x) = \mathbb{E}_{x\sim\mathcal{N}(0,\Sigma^{(L,00)}(x,x))}\left[\dot\sigma(x)^2\right] = 2\mathbb{E}_{x\sim\mathcal{N}(0,1)}[\mathbb{I}_{x\geq 0}] = 1.$$

This implies:

$$\begin{aligned}\Theta^{(L,00)}(x,x) &= \sum_{\ell=1}^{L} \Sigma^{(\ell,00)}(x,x)\left(\frac{r}{2}\right)^{L-\ell} = \sum_{\ell=1}^{L}\left(\frac{\beta^2}{1-\frac{r}{2}} + \frac{\left(\frac{r}{2}\right)^{\ell+1}}{1-\frac{r}{2}}\right)\left(\frac{r}{2}\right)^{L-\ell} \\ &= \frac{\beta^2(1 - \left(\frac{r}{2}\right)^L)}{\left(1 - \frac{r}{2}\right)^2} + L\frac{\left(\frac{r}{2}\right)^{L+1}}{1 - \frac{r}{2}}.\end{aligned}$$

The "bulk-values" for the activation kernels and the limiting NTK kernel can be deduced from the proof of Proposition E.7.2. A tedious study of variation of functions allows to prove the assertion on the boundary/bulk comparison. □

As a consequence of the previous proposition, in the limits as ℓ and L goes to infinity, the ratio boundary/bulk value is bounded by $\max\left(1, c\beta^2\right)$: the smaller β is, the stronger the boundary effect will be.

In the parent-based parametrization, the variance of the neurons throughout the network is always equal to 1 and the NTK $\Theta_{\infty,pp}^{(L)}(x,x)$ becomes independent of the position p: the border artifacts disappear.

Proposition E.7.2. *For the parent-based parametrization of DC-NNs, if the nonlinearity is standardized,* $\left(\Sigma^{(L)}\right)_{pp}(x)$ *and* $\left(\Theta_\infty^{(L)}\right)_{pp}(x)$ *do not depend neither on* $p \in I_L$ *nor on* $x \in \mathbb{S}_{n_0}^{I_0}$.

Proof. Actually, we will prove the stronger statement: for any General Convolutional Network, as defined in Section E.5, for any standardized nonlinearity, for any $x \in \mathbb{S}_{n_0}^{I_0}$ and any $p \in I_L$,

$$\Sigma^{(L,pp)}(x,x) = 1, \quad \text{and } \Theta_\infty^{(L,pp)}(x,x) = \frac{1-r^L}{1-r}.$$

For the activation kernels,this is proven by induction on ℓ . For any $x \in \mathbb{S}_{n_0}^{I_0}$ and any $p \in I_1$:

$$\begin{aligned}\Sigma^{(1,pp)}(x,x) &= \beta^2 + \frac{1-\beta^2}{|P(p)|\, n_0} \sum_{q\in P(p)} \sum_{q'\in P(p)} \chi(q \to p, q' \to p) x_q^T x_{q'} \\ &= \beta^2 + \frac{1-\beta^2}{|P(p)|\, n_0} \sum_{q\in P(p)} x_q^T x_q = \beta^2 + (1-\beta^2) = 1,\end{aligned}$$

and if the assertion holds true for L, then:

$$\begin{aligned}\Sigma^{(L+1,pp)}(x,x) &= \beta^2 + \frac{1-\beta^2}{|P(p)|\, n_0} \sum_{q\in P(p)} \sum_{q'\in P(p)} \chi(q \to p, q' \to p) \Sigma^{(L,qq')}(x,x) \\ &= \beta^2 + \frac{1-\beta^2}{|P(p)|\, n_0} \sum_{q\in P(p)} \Sigma^{(L,qq)}(x,x) = 1.\end{aligned}$$

For the activation kernels, this is proven by induction on L. It is easy to see that $\Theta_\infty^{(1,pp)}(x,x) = 1$ is valid for any $x \in \mathbb{S}_{n_0}^{I_0}$ and any $p \in I_L$. Let us show the induction step:

$$\begin{aligned}\Theta_\infty^{(L+1,pp)}(x,x) &= \Sigma^{(L+1,pp)}(x,x) + \frac{1-\beta^2}{|P(p)|} \sum_{q\in P(p)} \Theta_\infty^{(L,qq)}(x,x) R_{\dot\sigma}\left(\Sigma^{(L,qq)}(x,x)\right) \\ &= 1 + r\Theta_\infty^{(L,qq)}(x,x).\end{aligned}$$

Thus, $\Theta_\infty^{(L,pp)}(x,x) = \sum_{\ell=1}^{L} r^{L-\ell} = \frac{1-r^L}{1-r}$. □

E.8 Layerwise Contributions to the NTK and Checkerboard Patterns

In a DC-NN with stride $s \in \{2,3,...\}^d$, if two connection weight matrices $W^{(\ell,q\to p)}$ and $W^{(\ell,q'\to p')}$ are shared then $s \mid p'-p$. In other words, $\chi(q\to p, q'\to p') = 0$ as soon as $s \nmid p'-p$. The limiting contribution of the weights $\Theta_\infty^{(L:W^{(\ell)})}$ and bias $\Theta_\infty^{(L:b^{(\ell)})}$ to the limiting NTK can be formulated recursively. For the last layer $L-1$ we have

$$\begin{aligned}
\Theta_\infty^{(L:b^{(L-1)},pp')} &= \beta^2 \\
\Theta_\infty^{(1:W^{(0)},pp')} &= \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|\, n_0} \sum_{q\in P(p)} x_q^T y_{q+\frac{p'-p}{s}} \\
\Theta_\infty^{(L:W^{(L-1)},pp')} &= \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|} \sum_{q\in P(p)} R_\sigma\left(\Sigma^{(L-1,q,q+\frac{p'-p}{s})}(x,y)\right) \text{ for } L>1
\end{aligned}$$

and for the other layers, we have

$$\begin{aligned}
\Theta_\infty^{(L+1:b^{(\ell)},pp')} &= \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|} \sum_{q\in P(p)} \Theta_\infty^{(L;b^{(\ell)},q,q+\frac{p'-p}{s})}(x,y) R_{\dot\sigma}\left(\Sigma^{(L,q,q+\frac{p'-p}{s})}(x,y)\right) \\
\Theta_\infty^{(L+1:W^{(\ell)},pp')} &= \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|} \sum_{q\in P(p)} \Theta_\infty^{(L;W^{(\ell)},q,q+\frac{p'-p}{s})}(x,y) R_{\dot\sigma}\left(\Sigma^{(L,q,q+\frac{p'-p}{s})}(x,y)\right).
\end{aligned}$$

Proposition E.8.1. *In a DC-NN with stride $s \in \{2,3,...\}^d$, we have $\Theta_\infty^{(L:W^{(\ell)},pp')}(x,y) = 0$ if $s^{L-\ell} \nmid p'-p$ and $\Theta_\infty^{(L:b^{(\ell)},pp')}(x,y) = 0$ if $s^{L-\ell-1} \nmid p'-p$.*

Proof. From the formulas of the limiting contributions $\Theta^{(L:W^{(\ell)})}$ and $\Theta^{(L:b^{(\ell)})}$, we see that the bias of the last layer contribute to all pairs p,p' while the bias only contribute to pairs such that $s \mid p'-p$. Now by induction on L, if $\Theta^{(L:b^{(\ell)},qq')}$ and $\Theta^{(L:W^{(\ell)},qq')}$ only contribute to pairs q,q' such that $s^{L-\ell-1} \mid q'-q$ and $s^{L-\ell} \mid q'-q$ then

$$\begin{aligned}
\Theta_\infty^{(L+1:b^{(\ell)},pp')} &= \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|} \sum_{q\in P(p)} \Theta_\infty^{(L;b^{(\ell)},q,q+\frac{p'-p}{s})}(x,y) R_{\dot\sigma}\left(\Sigma^{(L,q,q+\frac{p'-p}{s})}(x,y)\right) \\
\Theta_\infty^{(L+1:W^{(\ell)},pp')} &= \delta_{s|p-p'} \frac{1-\beta^2}{|P(p)|} \sum_{q\in P(p)} \Theta_\infty^{(L;W^{(\ell)},q,q+\frac{p'-p}{s})}(x,y) R_{\dot\sigma}\left(\Sigma^{(L,q,q+\frac{p'-p}{s})}(x,y)\right)
\end{aligned}$$

only contribute to pairs p',p such that $s^{L-\ell} \mid p'-p$ and $s^{L+1-\ell} \mid p'-p$ as needed. □

Appendix F

DNN-Based Topology Optimization: Spatial Invariance and Neural Tangent Kernel

F.1 Derivation of the algorithm

In this section we show how to derive the equations used in our algorithm, especially the ones corresponding to implicit differentiation [79]. Let us recall that we consider a vector $X \in \mathbb{R}^N$ and compute a vector $Y = \Sigma(X) \in [0,1]^N$ (either Y^{MF} or Y^{NN}) by:

$$\forall i \in \{1, ..., N\},\ y_i = \sigma(x_i + \bar{b}(X)), \quad \text{such that: } \sum_{i=1}^{N} y_i = V_0, \quad \sigma(x) = \frac{1}{1+e^{-x}},$$

Where X denotes $(x_1, ..., x_N)$.

We want to show that this operation is well defined and find a formula to recover $\nabla_X C$ from a given $\nabla_Y C$. More precisely we have the following result.

Proposition F.1 (Proposition F.1 in the paper)**.** *Let $X \in \mathbb{R}^N$, the operation $Y = \Sigma(X)$ is well defined. Moreover, let $\dot{S}$ be the vector of the $\dot{\sigma}(x_i + \bar{b}(X))$. Then we have $\nabla_X C = D_X \nabla_Y C$ with:*

$$D_X := -\frac{1}{|\dot{S}|_1} \dot{S}\dot{S}^T + Diag(\dot{S}). \tag{F.1.1}$$

D_X is a symmetric positive semi-definite matrix whose kernel corresponds to constant vectors and has eigenvalues smaller than $\frac{1}{2}$.

Proof: Let us consider the function $F : \mathbb{R}^N \times \mathbb{R} \longrightarrow \mathbb{R}$ defined by: $F(z, b) = \sum_{i=1}^{N} \sigma(z_i + b)$. It is clear that $F(X, .)$ is stricty increasing on $\mathbb{R}$ from 0 to N. Then $\exists! \bar{b} \in \mathbb{R}$ such that $F(X, \bar{b}) = V_0$.

As $\partial_b F(X, \bar{b}) > 0$, by the implicit functions theorem, there exists a neighbourhood V of X in $\mathbb{R}^N$, a neighbourhood U of $\bar{b}$ in $\mathbb{R}$ and a function $\bar{b} : V \longrightarrow \mathbb{R}$ of class $\mathcal{C}^1$ such that:

$$\forall (z, b) \in V \times U,\ F(z, b) = V_0 \iff b = \bar{b}(z).$$

Moreover we also get from the implicit function theorem that:

$$\frac{\partial \bar{b}}{\partial x_i}(X) = -\left(\frac{\partial F}{\partial b}(X, \bar{b})\right)^{-1} \frac{\partial F}{\partial x_i}(X, \bar{b}) = -\left(\sum_{j=1}^{N} \dot{\sigma}(x_j + \bar{b})\right)^{-1} \dot{\sigma}(x_i + \bar{b}),$$

and we can apply chain rules:

$$\begin{aligned}\frac{\partial C}{\partial x_i} &= \sum_{j=1}^{N} \frac{\partial C}{\partial y_j}\frac{\partial y_j}{\partial x_i} \\ &= \sum_{j=1}^{N} \frac{\partial C}{\partial y_j}\dot{\sigma}(x_j + \bar{b}(x))\big(\frac{\partial \bar{b}}{\partial x_i} + \delta_{ij}\big),\end{aligned}$$

So that equation F.1.1 makes sense. Now, if we denote $\dot{S} = (a_1, ..., a_N)$, let us recall that we defined $a_i = \dot{\sigma}x_i + \bar{b}(X)$ where σ is the sigmoid function. By taking any $u \in \mathbb{R}^N$, we remark that:

$$\big(D_X u\big)_i = \frac{a_i}{|\dot{S}|_1}\sum_{j=1}^{N} a_j(u_i - u_j). \tag{F.1.2}$$

We easily deduce from equation F.1.2 that $\ker(D_X) = \text{span}(1_N)$ and that $D_X \in S_N^+(\mathbb{R})$. Indeed:

$$\begin{aligned}\forall u \in \mathbb{R}^N, \quad u^T(D_X)u &= -\frac{1}{|\dot{S}|_1}u^T\dot{S}\dot{S}^T u^T + \sum_{i=1}^{N} a_i u_i^2 \\ &= \frac{1}{|\dot{S}|_1}\Big\{ -\Big(\sum_{i=1}^{N} a_i u_i\Big)^2 + \Big(\sum_{i=1}^{N} a_i\Big)\Big(\sum_{i=1}^{N} a_i u_i^2\Big)\Big\} \\ &= \frac{1}{|\dot{S}|_1}\sum_{1\le i,j\le N} a_i a_j u_i(u_i - u_j) \\ &= \frac{1}{|\dot{S}|_1}\sum_{1\le i<j\le N} a_i a_j (u_i - u_j)^2 \ge 0.\end{aligned}$$

Eigenvalues: We already know that 0 is an eigenvalue with multiplicity 1. So let $u \neq 0$ in $\mathbb{R}^N$ and $\lambda > 0$ such that: $D_X u = \lambda u$. Then we easily show:

$$\forall i \in [\![1, N]\!], \quad \frac{a_i - \lambda}{a_i}u_i = \frac{1}{|\dot{S}|_1}\sum_{j=1}^{N} a_j u_j =: \langle u\rangle_a.$$

If $\langle u\rangle_a = 0$, then necessarily $\lambda \in \{a_1, ..., a_N\}$
If $\langle u\rangle_a \neq 0$, then we can assume (by normalising u) that $\langle u\rangle_a = 1$ and we have $u_i = \frac{a_i}{a_i - \lambda}$. Then we can replace $u_i = \frac{a_i}{a_i - \lambda}$ in the equation $\langle u\rangle_a = 1$:

$$\sum_{j=1}^{N} a_j = \sum_{j=1}^{N}\frac{a_j^2}{a_j - \lambda}, \quad \text{which by substraction leads to} \quad F(\lambda) := \sum_{j=1}^{N}\frac{a_j}{a_j - \lambda} = 0,$$

By studying the function F, we see that $\forall \lambda > \max_i(a_i),\ F(\lambda) < 0$. Therefore an eigenvalue always satisfies the inequality:

$$\lambda \le \max\{a_1, ..., a_N\} \le \|\dot{\sigma}\|_\infty = \frac{1}{4},$$

The last inequality coming from the fact that $a_i = \dot{\sigma}(x_i + \bar{b}(X))$, as mentionned earlier.

Remark: As shown above an important property of the matrix D_X is that it cancels out constants, which allows us to consider the limiting NTK up to some constant. The fact that the eigenvalues of D_X are in $[0, \frac{1}{4}]$ can help to avoid exploding gradients.

F.2 Equations of evolution

We quickly show how equations 5, 6 and 7 of the paper are derived. The proofs are mainly based on chain rules.

Let us first remark that the matrix D_X introduced above actually corresponds to the jacobian matrix $\nabla_X \Sigma$ of the application $\Sigma : \mathbb{R}^N \longrightarrow [0,1]^N$. So we can immediately applied chain rules to $Y^{\text{NN}} = \Sigma(X(\theta))$ and get:

$$\begin{aligned}\frac{\partial Y^{\text{NN}}}{\partial t} &= D_{X(\theta(t))}\frac{\partial X(\theta(t))}{\partial t} \\ &= -D_{X(\theta(t))}\tilde{\Theta}^L_{\theta(t)}\nabla_{X_{\theta(t)}}C \quad \text{(Gradient Descent)} \\ &= -D_{X(\theta(t))}\tilde{\Theta}^L_{\theta(t)}D_{X(\theta(t))}\nabla_{Y^{\text{NN}}}C(\theta(t)) \quad \text{(By proposition F.1).}\end{aligned}$$

Similarly, for the MF method, we set $X = T\bar{X}$ and obtain:

$$\begin{aligned}\frac{\partial Y^{\text{MF}}}{\partial t} &= D_{X(t)}\frac{\partial X(t)}{\partial t} \\ &= D_{X(t)}T\frac{\partial \bar{X}(t)}{\partial t} \quad \text{(Linearity)} \\ &= -D_{X(t)}T\nabla_{\bar{X}}C \quad \text{(Gradient descent)} \\ &= -D_{X(t)}TT^T\nabla_X C \quad \text{(Chain rule)} \\ &= -D_{X(t)}TT^T D_{X(t)}\nabla_{Y^{\text{MF}}}C.\end{aligned}$$

F.3 Details about embeddings

Torus embedding

The aim of this section is to give details about properties of the limiting NTK in case of Torus embedding. As a reminder we consider the following embedding:

$$\mathbb{R}^2 \ni p = (p_1, p_2) \longmapsto \varphi(p) = r(\cos(\delta p_1), \sin(\delta p_1), \cos(\delta p_2), \sin(\delta p_2));$$

In particular we show the following proposition which basically says that $\tilde{\Theta}_\infty$ is in that case a discrete convolution and derive from there its spectral properties and construct its positive semi-definite square root

Proposition F.2 (Proposition 6.4 in the paper)**.** *We can always extend our $n_x \times n_y$ grid and choose δ such that the embedded grid covers the whole torus (typically $\delta = \frac{\pi}{2\max(n_x,n_y)}$ and take a $n \times n$ grid with $n = 4\max(n_x, n_y)$). Then the Gram matrix $\tilde{\Theta}_\infty$ of the limiting NTK is a $2D$ discrete convolution matrix. Moreover the NTK Gram matrix has a positive definite square root $\sqrt{\tilde{\Theta}_\infty}$ which is also a discrete convolution matrix.*

proof: We assume that we extend the grid in a $n \times n$ grid with $n \geq n_x, n_y$. Now we take $\delta = \frac{2\pi}{n}$ and we consider the limiting NTK Gram matrix on $\varphi\big([\![n,n]\!] \times [\![n,n]\!]\big)$.

As $\Theta_\infty(\varphi(p), \varphi(p'))$ depends only on $p - p'$, we can see the limiting NTK Gram Matrix as a discrete convolution kernel $\mathcal{K}$ acting on $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$:

$$\Theta_\infty((k,k'),(j,j')) = \mathcal{K}(k-k', j-j'),$$

For (k,k'), $(j,j') \in \mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$.

We see $\tilde{\Theta}_\infty$ as a n^2 square matrix with each index in $\mathbb{Z}/n\mathbb{Z} \times \mathbb{Z}/n\mathbb{Z}$.

We introduce the Fourier vectors $\Omega_m = (e^{-i2\pi\frac{mk}{n}})_{0\leq k\leq n_x-1}$. As $\tilde{\Theta}_\infty$ is a 2D convolution matrix, we classically have the following results:

The eigenvectors of $\tilde{\Theta}_\infty$ are exactly given by:

$$\Omega_m \otimes \Omega_M,$$

for $0 \leq m \leq n_x - 1$ and $0 \leq M \leq n_y - 1$, $\otimes$ denotes the Kronecker product. The corresponding eigenvalue is given by the discrete Fourier transform $\widehat{\mathcal{K}}(m,M)$ with:

$$\widehat{\mathcal{K}}(m,M) = \sum_{j=0}^{n-1}\sum_{j'=0}^{n-1} e^{-i2\pi\frac{mj}{n}} e^{-i2\pi\frac{Mj'}{n}} \mathcal{K}(j,j').$$

Moreover, as the matrix $\tilde{\Theta}_\infty$ is positive definite (from the positive definiteness of the NTK, [105]) those eigenvalues verify $\widehat{\mathcal{K}}(m,M) \geq 0$ and it makes sense to write the square root of the NTK Gram Matrix as the inverse Fourier transform of the $\sqrt{\widehat{\mathcal{K}}(m,M)}$:

$$\sqrt{\tilde{\Theta}_\infty}((k,k'),(j,j')) = \frac{1}{n^2}\sum_{m=0}^{n-1}\sum_{M=0}^{n-1} e^{i2\pi\frac{m(j-k)}{n}} e^{i2\pi\frac{M(j'-k')}{n}} \sqrt{\widehat{\mathcal{K}}(m,M)}, \tag{F.3.1}$$

It is easy to see that the matrix defined by equation F.3.1 is symmetric and positive semi-definite. Indeed we can write $\sqrt{\tilde{\Theta}_\infty}((k,k'),(j,j')) = g(k-j,k'-j')$ with g the Fourier transform of a positive vector.

Moreover it follows from the (discrete) convolution theorem that $\sqrt{\tilde{\Theta}_\infty}^2 = \tilde{\Theta}_\infty$. Therefore $\sqrt{\tilde{\Theta}_\infty}((k,k'),(j,j'))$ is indeed the positive semi-definite matrix square root of $\tilde{\Theta}_\infty$.

Thus the square root of the NTK Gram matrix can be seen as a convolution filter as well (it is invariant by translation as a function of $(k-j,k'-j')$).

Dimension of radial embeddings

In this section we prove that feature maps associated to continuous radial kernels are either trivial or of infinite dimension. this result is what motivates discussion in section 6.3 of the paper.

Let us first recall Bochner theorem ([186]):

Theorem F.1 (Bochner)**.** *Let $(x,y) \mapsto k(x-y)$ be a continuous shift invariant positive definite kernel on $\mathbb{R}^d$. Then it is the Fourier transform of a finite positive measure Λ on $\mathbb{R}^d$:*

$$k(r) = \int_{\mathbb{R}^d} e^{i\omega\cdot r} d\Lambda(\omega).$$

The function k appearing in the above theorem will be called a positive definite function, according to the following definition:

Definition 2. Let $k : \mathbb{R}^d \longrightarrow \mathbb{R}$, then k is a positive definite function when for all n, all $p_1, \ldots, p_n \in \mathbb{R}^d$ and all $c_1, ..., c_n \in \mathbb{R}$ we have:

$$\sum_{1\leq i,j\leq n} c_i c_j k(x_i - x_j) \geq 0.$$

Moreover we will denote $SO(d)$ the set of rotations matrices of dimension d and the Fourier transform (for an integrable function ψ):

$$\mathcal{F}\psi(\omega) = \int_{\mathbb{R}^p} \psi(p) e^{-i\omega\cdot p} dp.$$

Let us now recall the result that we want to prove:

Proposition F.3 (Proposition 6.3 in the paper)**.** *Let $\varphi : \mathbb{R}^d \to \mathbb{R}^m$ for $d > 2$ and any finite m. If φ satisfies*

$$\varphi(x)^T \varphi(x') = K(\|x - x'\|) \tag{F.3.2}$$

for some continuous function K then both φ and K are constant. We will denote $k(x - x') := K(\|x - x'\|)$.

Proof: We procede in the following way: We consider an embedding φ as described above and we are going to show that, when K is not constant, one can construct arbitrarily big linearly independent families $\varphi(p_1), \dots, \varphi(p_n)$.

For now let us take pairwise distinct $p_1, \dots, p_n \in \mathbb{R}^d$ and $c_1, \dots, c_n \in \mathbb{R}$ such that:

$$\sum_{k=1}^n c_k \varphi(p_k) = 0.$$

A clever choice for $p_1, \dots, p_n$ will be done later.

For any $p \in \mathbb{R}^d$ and any rotation $R \in SO(d)$ we can write:

$$\begin{aligned} 0 = \varphi(p)^T \sum_{k=1}^n c_k \varphi(p_k) = \sum_{k=1}^n c_k K(\|p - p_k\|) = \sum_{k=1}^n c_k K(\|Rp - Rp_k\|) \\ = \varphi(Rp)^T \sum_{k=1}^n c_k \varphi(Rp_k). \end{aligned}$$

Since this is true for all $p' = Rp$ we can deduce that for all $p \in \mathbb{R}^d$ and all $R \in SO(d)$ we have:

$$\sum_{k=1}^n c_k k(p - Rp_k) = 0.$$

We denote by Λ the finite measure on $\mathbb{R}^d$ given by Bochner's theorem applied on k.

Let us take a test function $\psi \in \mathcal{S}(\mathbb{R}^p)$ in the Schwartz space, we can write successively that for all rotation $R \in SO(d)$:

$$\begin{aligned} 0 &= \int_{\mathbb{R}^d} \mathcal{F}\psi(p) \sum_{k=1}^n c_k k(p - Rp_k) dp \\ &= \int_{\mathbb{R}^d} \mathcal{F}\psi(p) \sum_{k=1}^n c_k \int_{\mathbb{R}^d} e^{i\omega\cdot(p - Rp_k)} d\Lambda(\omega) dp, \quad \text{(Bochner's theorem)} \\ &= \int_{\mathbb{R}^d} \left(\sum_{k=1}^n c_k e^{-i\omega\cdot(Rp_k)} \right) \int_{\mathbb{R}^d} \mathcal{F}\psi(p) e^{i\omega\cdot p} dp \; d\Lambda(\omega), \quad \text{(Fubini's theorem)} \\ &= (2\pi)^d \int_{\mathbb{R}^d} \psi(\omega) \sum_{k=1}^n c_k e^{-i\omega\cdot(Rp_k)} d\Lambda(\omega), \quad \text{(Fourier inversion)} \end{aligned}$$

As K is not constant, we can find $\omega_0 \in \mathbb{R}^d \backslash \{0\}$ such that for all $\epsilon > 0$ small enough we have $\Lambda\big(B(\omega_0, \epsilon)\big) > 0$ (otherwise the finite positive measure Λ would be concentrated on 0 and k would be constant).

Let $R \in SO(d)$, if we assume that $S := \sum_{k=1}^{n} c_k e^{-i\omega_0 \cdot (Rp_k)} \neq 0$ then we can find a small enough open ball $B(\omega_0, \epsilon)$ on which $Re(S))$ and $Im(S)$ have constant sign and such that: $|Re(S)| \geq c_1 > 0$ or $|Im(S)| \geq c_1 > 0$.

We choose ψ such that $\psi \geq 0$, ψ has compact support in $B(\omega_0, \epsilon)$ and $\psi \geq c_2 > 0$ on $B(\omega_0, \frac{\epsilon}{2})$. Then we obtain a contradiction by writing $0 \geq (2\pi)^d) c_1 c_2 \Lambda(B(\omega_0, \frac{\epsilon}{2}))$. (We separate real and imaginary parts).

This implies that:

$$\forall R \in \mathrm{SO}(d),\ \sum_{k=1}^{n} c_k e^{-i(R\omega_0)\cdot p_k} = 0, \tag{F.3.3}$$

Now we take a particular choice of (p_i), let $p_k = (k, 0, \ldots, 0) \in \mathbb{R}^d$.

Up to rotations, we can assume without loss of generality that $\omega_0 = (w, 0, \ldots, 0)$ with $w \neq 0$. Moreover, we consider the particular case of rotations in the 2D plane generated by $(1, 0, \ldots, 0)$ and $(0, 1, 0, \ldots, 0)$.

Therefore, equation F.3.3 implies that:

$$\forall \theta \in \mathbb{R},\ \sum_{k=1}^{n} c_k \big(e^{-iw\cos(\theta)}\big)^k = 0,$$

So that the polynomial $\sum_k c_k z^k$ has an infinite number of roots. Thus $c_1 = \cdots = c_n = 0$.

Random features embedding

In this section we give some details about the way we define random embeddings, which is very similar but slightly different than in [176].

If the kernel is properly scaled (i.e. $k(0) = 1$) then Λ defines a probability measure. That's why we introduce a probability measure $\mathbb{Q}$ and write:

$$k(r) = k(0) \int_{\mathbb{R}^d} e^{i\omega \cdot r} d\mathbb{Q}(\omega) = k(0) \mathbb{E}_{\omega \sim \mathbb{Q}}[e^{i\omega \cdot r}].$$

Now, following the reasoning in [176] we consider:

$$\varphi(p)_i = \sqrt{2k(0)} \sin(\omega \cdot p + \frac{\pi}{4} + b)$$

With $\omega \sim \mathbb{Q}$ and b a random variable with a symmetric law (note that $\mathbb{Q}$ is also symmetric). Then we have:

$$\begin{aligned}
\mathbb{E}[\varphi(p)_i \varphi(p')_i] &= 2k(0)\mathbb{E}\left[\left(\frac{e^{i\omega.p + \frac{\pi}{4} + b} - e^{-i\omega.p - \frac{\pi}{4}} - b}{2i}\right)\left(\frac{e^{i\omega.p' + \frac{\pi}{4} + b} - e^{-i\omega.p' - \frac{\pi}{4}} - b}{2i}\right)\right] \\
&= -\frac{k(0)}{2}\Big(e^{i\frac{\pi}{2}}\mathbb{E}[e^{i\omega.(p+p') + 2b}] + e^{-i\frac{\pi}{2}}\mathbb{E}[e^{-i\omega.(p+p') - 2b}] \\
&\qquad - \mathbb{E}[e^{i\omega.(p-p')}] - \mathbb{E}[e^{-i\omega.(p-p')}]\Big) \\
&= k(0)\mathbb{E}[e^{i\omega.(p-p')}] \\
&= k(p - p').
\end{aligned}$$

Therefore we reduce the variance by drawing i.i.d. samples $\omega_1, \dots, \omega_{n_0}$ and $b_1, \dots, b_{n_0}$ as described in section 3 and computing the mean $\frac{1}{n_0}\varphi(p)^T\varphi(p')$. By the strong law of large numbers we have the almost sure convergence:

$$\frac{1}{n_0}\varphi(p)^T\varphi(p') \underset{n_0\to\infty}{\longrightarrow} k(p-p'),$$

Now we can obtain Gaussian embedding by drawing the bias from δ_0 and weights from $\mathcal{N}(0, \frac{1}{\ell^2}I_d)$. from the above formulas we immediately get:

$$k(p-p') = e^{-\frac{\|p-p'\|_2^2}{2\ell^2}}.$$

F.4 Precise computations of the Neural Tangent Kernel

We now give more details about the computation of the limiting NTK and detail how we obtain the limiting kernels used in Figures 6 and 7 of the paper.

Limiting NTK

For this purpose, following several authors ([105], [224], [129]), we need to introduce some gaussian processes and their associated kernels. For a symmetric positive kernel Σ let us define:

$$\begin{cases} \mathcal{T}(\Sigma)(z,z') = \mathbb{E}_{(X,Y)\sim\mathcal{N}(0,\Sigma_{z,z'})}\big[\mu(X)\mu(Y)\big] \\ \dot{\mathcal{T}}(\Sigma)(z,z') = \mathbb{E}_{(X,Y)\sim\mathcal{N}(0,\Sigma_{z,z'})}\big[\dot{\mu}(X)\dot{\mu}(Y)\big] \end{cases} \quad \text{With :} \quad \Sigma_{z,z'} = \begin{pmatrix} \Sigma(z,z) & \Sigma(z,z') \\ \Sigma(z,z') & \Sigma(z',z') \end{pmatrix}.$$

Then we set $\Sigma^1(z,z') = \Theta^1_\infty(z,z') = \beta^2 + \frac{\alpha^2}{n_0}z^Tz'$ and we define recursively:

$$\sigma^{l+1} = \beta^2 + \alpha^2\mathcal{T}(\Sigma^l), \quad \dot{\Sigma}^{l+1} = \alpha^2\dot{\mathcal{T}}(\Sigma^l), \quad \Theta^{l+1}_\infty = \dot{\Sigma}^{l+1}\Theta^l_\infty + \Sigma^{l+1}. \tag{F.4.1}$$

Using those formulas it is clear that the limiting NTK is invariant under rotation.

When neurons have constant variance, the following notion of dual activation function is often very useful:

Definition 3. Let $\mu : \mathbb{R} \longrightarrow \mathbb{R}$ be a function such that $\mathbb{E}_{X\sim\mathcal{N}(0,1)}[\mu(X)^2] < +\infty$, then its dual function $\hat{\mu} : [-1,1] \longrightarrow \mathbb{R}$ is defined by:

$$\hat{\mu}(\rho) = \mathbb{E}_{(X,Y)\sim\mathcal{N}(0,\Sigma_\rho)}[\mu(X)\mu(Y)], \quad \text{With : } \Sigma_\rho = \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}.$$

We will use some properties of the dual function, which are described in [42].

Another way of seeing Gaussian embedding

As explained above (Section 6.3), the Gaussian embedding can be seen as the first hidden layer of a neural network, with the first layer untrained. Thus it actually corresponds to Σ^2 with the above notations.

Let us consider the activation function $\mu : a \longmapsto \lambda \sin(\omega a + \frac{\pi}{4})$ and denote:

$$\forall x, y \in \mathbb{R}^{n_0},\ \Sigma^1_{x,y} = \begin{pmatrix} \beta^2 + \dfrac{1-\beta^2}{n_0}\|x\|_2^2 & \beta^2 + \dfrac{1-\beta^2}{n_0}x^T y \\ \beta^2 + \dfrac{1-\beta^2}{n_0}x^T y & \beta^2 + \dfrac{1-\beta^2}{n_0}\|y\|_2^2 \end{pmatrix},$$

We are looking at:

$$\Sigma^2(x,y) = \beta^2 + (1-\beta^2)\mathbb{E}_{(X,Y)\sim\mathcal{N}(0,\Sigma^{(1)}_{x,y})}[\mu(X)\mu(Y)].$$

Let $(X,Y) \sim \mathcal{N}(0, \Sigma^1_{x,y})$, then $X - Y$ and $X + Y$ are normal random variables and $\mathbb{V}(X-Y) = \frac{1-\beta^2}{n_0}\|x-y\|_2^2$. Thus, using properties of characteristic functions we get:

$$\begin{aligned}
\mathbb{E}[\mu(X)\mu(Y)] &= \lambda^2 \mathbb{E}\left[\left(\frac{e^{i\omega X + \frac{\pi}{4}} - e^{-i\omega X - \frac{\pi}{4}}}{2i}\right)\left(\frac{e^{i\omega Y + \frac{\pi}{4}} - e^{-i\omega Y - \frac{\pi}{4}}}{2i}\right)\right] \\
&= -\frac{\lambda^2}{4}\left(e^{i\frac{\pi}{2}}\mathbb{E}[e^{i\omega(X+Y)}] + e^{-i\frac{\pi}{2}}\mathbb{E}[e^{-i\omega(X+Y)}] - \mathbb{E}[e^{i\omega(X-Y)}] - \mathbb{E}[e^{-i\omega(X-Y)}]\right) \\
&= \frac{\lambda^2}{2}\mathbb{E}[e^{i\omega(X-Y)}] \\
&= \frac{\lambda^2}{2}\exp\left\{-\frac{1}{2}\omega^2\frac{1-\beta^2}{n_0}\|x-y\|_2^2\right\}.
\end{aligned}$$

Computation of the NTK used for Figure 7 in the paper

In this section we show how one can derived analytically the function Φ_∞ described in Section 6.4. This kind of computation can be used to derive numerically the filter radius $\hat{R}_{1/2}$ and tune the hyperparameters.

We use here a Gaussian embedding φ of size n_0 with lenghtscale ℓ followed by one hidden linear layer (activation function $x \to \sqrt{2}\max(0,x)$) of size n_1 and the output layer $n_2 = 1$. We also take $\alpha^2 + \beta^2 = 1$ in those experiments, to ensure constant variance of the neurons.

By the strong law of large numbers we have for the limiting NTK of the first layer:

$$\Theta^1_\infty(\varphi(p),\varphi(p')) = \beta^2 + \frac{1-\beta^2}{n_0}\varphi(p)^T\varphi(p') \underset{n_0\to\infty}{\longrightarrow} \beta^2 + (1-\beta^2)e^{-\frac{\|p-p'\|_2^2}{2\ell^2}} =: G(\|p-p'\|).$$

For the second layer, we use the notion of dual function defined above. In the case of the standardized ReLu it is computed in [42]:

$$\hat{r}(\rho) = \rho - \frac{\rho\arccos(\rho) - \sqrt{1-\rho^2}}{\pi}, \quad \rho \in [-1,1],$$

and:

$$\hat{\dot{r}}(\rho) = \dot{\hat{r}}(\rho) = 1 - \frac{\arccos(\rho)}{\pi}.$$

So that we can write, with $d = \|p-p'\|$:

$$\Phi_\infty(d) = \hat{r}(G(d)) + G(d)\dot{\hat{r}}(G(d)).$$

Therefore Φ_∞ only depends on ℓ and β. From this expression we can use standard Python libraries to approximate $\hat{R}_{1/2}$ for given values of the hyperparameters.

Computation of the NTK used for Figure 6 in the paper

Now we derive an approximate of the quantity $\hat{R}_{1/2}$ used in Figure 6 of the paper. This is a little bit more difficult than with Gaussian embedding because the rotation invariance is now only an approximation, even in the infinite-width limit.

With Torus embedding, we have $n_0 = 4$. The embedding is followed by two hidden linear layers with standardised cosine activation function, and then the last linear layer. We used here $r = \sqrt{2}$ $\delta = \frac{\pi}{80}$ (which is the formula suggested in the paper with $n_x = n_y = 40$). As in the case of Gaussian embedding, we set $\alpha^2 = 1 - \beta^2$. This ensures that neurons have constant variance and allows easy analytical computations.

Thanks to the Torus embedding described above, we get for the first layer:

$$\begin{aligned}\Theta^1_\infty(\varphi(p), \varphi(p')) &= \beta^2 + \frac{1-\beta^2}{n_0}\varphi(p)^T\varphi(p') \\ &= \beta^2 + \frac{1-\beta^2}{2}\big(\cos(\delta(p_1 - p'_1)) + \cos(\delta(p_2 - p'_2))\big)\end{aligned}$$

As rotation invariance is not analytically correct here, we look at the limiting NTK in the direction $p_1 = p_2$. which gives:

$$\Sigma^1(\varphi(p), \varphi(p')) = \Theta^1_\infty(\varphi(p), \varphi(p')) = \beta^2 + (1-\beta^2)\cos(\delta r),$$

with $r = |p_1 - p'_1| = |p_2 - p'_2|$.

For the next layers, we use the dual function of the standardised cosine (see [42]) given by:

$$\hat{\mu}(\rho) = \frac{\cosh(\omega^2\rho)}{\cosh(\omega^2)},$$

and its derivative:

$$\hat{\dot{\mu}}(\rho) = \omega^2\frac{\sinh(\omega^2\rho)}{\cosh(\omega^2)},$$

Then the limiting NTK is simply given by the following formulas:

$$\begin{aligned}\Sigma^{l+1}(\varphi(p), \varphi(p')) &= \beta^2 + (1-\beta^2)\hat{\mu}(\Sigma^l(\varphi(p), \varphi(p'))), \\ \dot{\Sigma}^{l+1}(\varphi(p), \varphi(p')) &= (1-\beta^2)\hat{\mu}(\dot{\Sigma}^l(\varphi(p), \varphi(p'))), \\ \Theta^{l+1}_\infty(\varphi(p), \varphi(p')) &= \Sigma^{l+1}(\varphi(p), \varphi(p')) + \dot{\Sigma}^{l+1}(\varphi(p), \varphi(p'))\Theta^l_\infty(\varphi(p), \varphi(p')).\end{aligned}$$

This way we construct a function $\Phi_\infty(r)$ with r an approximation of the radius and we can use it to compute numerically an approximation of $\hat{R}_{1/2}$ as before.

F.5 Square root of the NTK in the case of random embedding

We now prove that we can define a notion of a square root of the NTK. First we need a technical lemma:

Lemma F.1. *Let μ be a continuous function such that $\mathbb{E}_{X\sim\mathcal{N}(0,1)}[\mu(X)^2] = 1$, $C \in [0,1]$ a constant and $f \geq 0$ a positive definite function (in the sense of definition 2) such that $C + f(p) \leq 1$. Then the function*

$$F : p \longmapsto \hat{\mu}(C + f(p)) - \hat{\mu}(C),$$

is positive definite, where $\hat{\mu}$ denotes the dual function of μ (see definition 3).

Proof:

Let us take $p_1, ..., p_m \in \mathbb{R}^d$ and $c_1, ..., c_m \in \mathbb{R}$. We introduce the Hermite expansion $\sum_k a_k h_k$ of μ and write its dual function as (see [42]):

$$\hat{\mu}(\rho) = \sum_{k=0}^{+\infty} a_k^2 \rho^k, \quad \rho \in [-1, 1],$$

Then by Bernoulli's formula:

$$\hat{\mu}(C + f(p_i - p_j)) - \hat{\mu}(C) = \sum_{k=1}^{+\infty} a_k^2 f(p_i - p_j) \sum_{s=0}^{k-1} C^{k-1-s} (C + f(p_i - p_j))^s.$$

Thus by polynomial combination with positive coefficients of positive semi-definite kernels:

$$\sum_{i,j=1}^{m} c_i c_j F(p_i - p_j) = \sum_{k=1}^{+\infty} \sum_{s=0}^{k-1} a_k^2 C^{k-1-s} \sum_{i,j=1}^{m} c_i c_j f(p_i - p_j)(C + f(p_i - p_j))^s \geq 0,$$

Which achieves the proof.

Let us recall the statement that we want to prove:

Proposition F.4 (Proposition 6.5 in the paper)**.** *Let φ be an embedding as described in section 6.3 of the paper, for a positive radial kernel $k \in L^1(\mathbb{R}^d)$ with $k(0) = 1$. Then there is a filter function $g : \mathbb{R} \to \mathbb{R}$ and a constant C such that for all p, p':*

$$\lim_{n_0 \to \infty} \Theta_\infty(\varphi(p), \varphi(p')) = C + (g \star g)(p - p'), \tag{F.5.1}$$

where Θ_∞ is the limiting NTK of a network with Lipschitz, non constant, and standardized activation function μ.

Before writing the proof, let us make some remarks on the assumptions of this proposition and their immediate implications:

- We recall that the fact that μ is "standardised" means here: $\mathbb{E}_{X \sim \mathcal{N}(0,1)}[\mu(X)^2] = 1$.
- As mentioned before (Section 6.2 of the paper) we assume for simplicity that $\alpha^2 = 1 - \beta^2$ to ensure constant variance of the neurons (we consider $\beta \in [0, 1)$).
- We denote by A the Lipschitz constant of μ. By Rademacher theorem, we know that μ is almost everywhere differentiable and $\|\dot{\mu}\|_\infty \leq A$. The fact that μ is not constant ensures that $\hat{\mu}$ is (strictly) increasing on $[0, 1)$.
- Moreover, the Lipschitz assumption also implies that $|\hat{\mu}(1)| \leq A^2 < +\infty$ and therefore $\hat{\mu}$ is continuous on $[-1, 1]$ by Abel's theorem on entire series.
- The procedure to approximate the kernel k in Section 6.3 of the paper assumes that k is continuous (to be able to apply Bochner's theorem). It is therefore also the case in this proof.

Proof of the proposition:

Step 1: We want to show by recursion that for all $l \geq 1$ there exists some constant $C_l \in [0,1)$ such that for all $p, p' \in \mathbb{R}^d$ we have in probability:

$$\Sigma^l(\varphi(p), \varphi(p')) \underset{n_0 \to \infty}{\longrightarrow} C_l + f_l(p - p'), \tag{F.5.2}$$

With f_l a radial positive definite function such that $f_l \geq 0$ and $f_l \in L^1(\mathbb{R}^d)$.

For $l = 1$, we know that this is true by the law of large numbers:

$$\begin{aligned}\Sigma^1(\varphi(p), \varphi(p')) = \Theta^1_\infty(\varphi(p), \varphi(p')) &= \beta^2 + \frac{1-\beta^2}{n_0}\varphi(p)^T\varphi(p') \\ &\underset{n_0 \to \infty}{\longrightarrow} \beta^2 + (1-\beta^2)k(p-p'),\end{aligned} \tag{F.5.3}$$

We just set $f_1 = (1-\beta^2)k$. Now we assume $l \geq 2$:

We have by our normalisation assumptions $\Sigma^l(\varphi(p), \varphi(p)) = C_l + f_l(0) = 1$. Using the continuity of $\hat{\mu}$ (see [42] for the properties of $\hat{\mu}$), we have:

$$\begin{aligned}\Sigma^{l+1}(\varphi(p), \varphi(p')) &= \beta^2 + (1-\beta^2)\hat{\mu}(\Sigma^l(\varphi(p), \varphi(p'))) \\ &\underset{n_0 \to \infty}{\longrightarrow} \beta^2 + (1-\beta^2)\hat{\mu}(C_l + f_l(p-p')).\end{aligned} \tag{F.5.4}$$

Using properties of the dual function given in [42], we know that $\hat{\mu}$ is positive, increasing and convex in $[0,1]$. Moreover as f_l is radial positive definite we have $f_l \leq f_l(0) = 1 - C_l$. Then by convexity:

$$\begin{aligned}\hat{\mu}(C_l + f_l(p-p')) &= \hat{\mu}\left(\frac{f_l(p-p')}{1-C_l} + \left(1 - \frac{f_l(p-p')}{1-C_l}\right)C_l\right) \\ &\leq \frac{f_l(p-p')}{1-C_l}\hat{\mu}(1) + \left(1 - \frac{f_l(p-p')}{1-C_l}\right)\hat{\mu}(C_l).\end{aligned}$$

Using that $\hat{\mu}$ is increasing:

$$|\hat{\mu}(C_l + f_l(p-p')) - \hat{\mu}(C_l)| \leq \frac{\hat{\mu}(1) - \hat{\mu}(C_l)}{1-C_l} f_l(p-p'),$$

So that we can rewrite equation F.5.4 in the following form:

$$\Sigma^{l+1}(\varphi(p), \varphi(p')) \underset{n_0 \to \infty}{\longrightarrow} \beta^2 + (1-\beta^2)\hat{\mu}(C_l) + f_{l+1}(p-p'),$$

With $f_{l+1}(p-p') = (1-\beta^2)(\hat{\mu}(C_l + f_l(p-p')) - \hat{\mu}(C_l))$ and $C_{l+1} = \beta^2 + (1-\beta^2)\hat{\mu}(C_l)$.

The previous inequality, lemma F.1 and the fact that $\hat{\mu}$ is increasing in $[0,1)$ ensure the properties of f_{l+1} and C_{l+1}.

Step 2: As $\hat{\dot{\mu}}$ is also positive, continuous, increasing and convex in $[0,1]$, we can obtain a convergence in probability similar to equation F.5.2 but for $\dot{\Sigma}^l$:

$$\dot{\Sigma}^l(\varphi(p), \varphi(p')) \underset{n_0 \to \infty}{\longrightarrow} B_l + h_l(p-p'),$$

With $B_l \geq 0$, and h_l a positive definite function such that $h_l \in L^1(\mathbb{R}^d)$ and $h_l \geq 0$.

Now we want to show by recursion that for a fixed l:

$$\forall p, p' \in \mathbb{R}^d,\ \Theta^l_\infty(\varphi(p), \varphi(p')) \underset{n_0 \to \infty}{\longrightarrow} C_{\mu,\beta,l} + \theta_l(p - p'). \tag{F.5.5}$$

With θ_l a positive definite function such that $\theta_l \in L^1(\mathbb{R}^d)$ and $C_{\mu,\beta,l} \geq 0$. Again we know that this is true for $l = 1$ by equation F.5.3.

We have:

$$\Theta^{l+1}_\infty(\varphi(p), \varphi(p')) \underset{n_0 \to \infty}{\longrightarrow} (C_{\mu,\beta,l} + \theta_l(p - p'))\dot{\Sigma}^{(l+1)}(p, p') + C_{l+1} + f_{l+1}(p - p').$$

So that we can set:

$$\theta_{l+1}(\varphi(p), \varphi(p')) = C_{\mu,\beta,l} h_{l+1}(p - p') + \theta_l(p - p')\dot{\Sigma}^{l+1}(\varphi(p), \varphi(p')) + f_{l+1}(p - p'),$$

and:

$$C_{\beta,\mu,l+1} = C_{l+1} + C_{\beta,\mu,l} B_l.$$

Using that $|\theta_l(p - p')\dot{\Sigma}^{l+1}(\varphi(p), \varphi(p'))| \leq A^2 |\theta_l(p - p')|$ and all the previous results, the recursion works automatically and we have equation F.5.5 for all $l \geq 2$.

Moreover $(p, p') \longmapsto \theta_l(p - p')\dot{\Sigma}^{1+l}(p, p')$ is positive semi-definite as a product of two positive semi-definite kernels. By sum we deduce that θ_{l+1} is positive semi-definite and by recursion we have the result for all θ_l.

Step 3: Now, using integrability of θ_l, we know that its Fourier transform defines a function $q \in L^\infty(\mathbb{R}^d)$.

From dominated convergence theorem we deduce that q is continuous.

Therefore in the sense of distributions, the Fourier transform of θ_L is given by a finite positive measure (Bochner's theorem) and also by $q \in L^\infty(\mathbb{R}^d)$. We deduce that q is the density of this finite positive measure (the Radon-Nikodym derivative with respect to the Lebesgue measure).

From those arguments we get $q \geq 0$ and $q \in L^1(\mathbb{R}^d)$. We then have the Fourier inversion formula for θ_L:

$$\theta_L(p - p') = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} q(\omega) e^{i\omega.(p-p')} d\omega, \quad \text{with: } q \geq 0$$

Hence it makes sense to define:

$$g = \mathcal{F}^{-1}(\sqrt{q}),$$

In the sense of the Fourier transform of a L^2 function. Then the convolution theorem ensures:

$$\theta_L = g \star g.$$

Remark: Here we used lemma F.1 and the dual activation function to show that both f_l and θ_l are positive definite. If we only show that $\theta_l \in L^1(\mathbb{R}^d)$ it is still possible to show the same properties of the function q by using positive definiteness of $C + \theta_L$ and take the Fourier transform in the sense of distributions, which leads to $(2\pi)^d C\delta_0 + q = (2\pi)^d M$ with M a finite positive measure. Then arguments based on test functions and the continuity of q give the result. The advantage of lemma F.1 is that it is a bit more general.

F.6 Additional experimental results

Plots of the Neural Tangent Kernel

Here are some additional experimental results regarding the comparison between the theoretical (limiting) NTK $\tilde{\Theta}_\infty$ and the empirical NTK $\tilde{\Theta}_{\theta(t)}$. Here again the "lines" of the Gram matrices are reshaped as images.

Figure F.6.1 represents the comparison between the limiting NTK and the emprirical NTK with a Gaussian embedding. We can observe that the infinite-width limit seems to be well-respected.

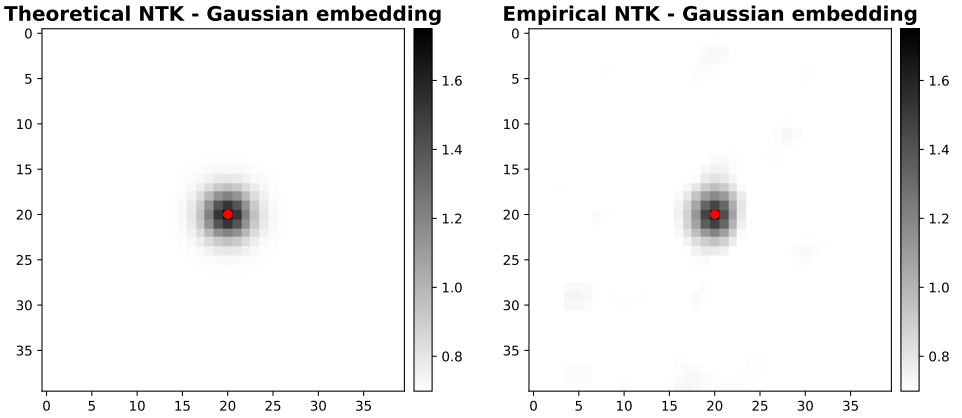


Figure F.6.1: Comparison between one line of the Gram matrix of the empirical NTK $\tilde{\Theta}_{\theta(t)}$ and and of the corresponding limiting NTK $\tilde{\Theta}_\infty$. Here we use a Gaussian embedding as described in the paper

Figure F.6.2 shows the evolution of the NTK during the optimisation process. While the NTK begins to change at the end of training (it is due to the alignment of descent directions, because of the sigmoid we use to control the volume, pre-densities $(x_i)_{1\leq i\leq N}$ tend to infinity) the NTK stays close to Θ_∞ during the part of training where the final shape is created. This justifies even more that it is pertinent to study the effect of the NTK on the final geometry.

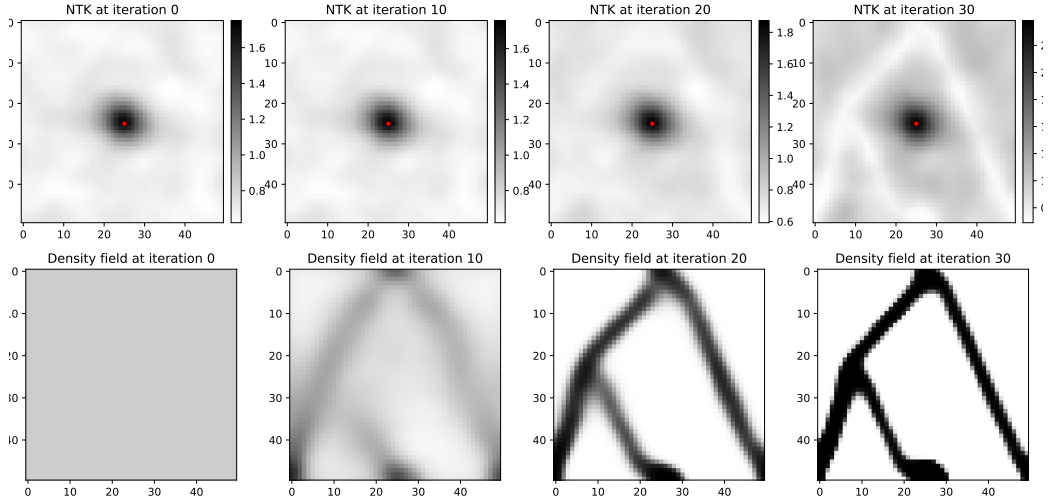


Figure F.6.2: Evolution of the NTK of a network with a Gaussian embedding with hyperparameters as described in Section 6.4. We can see a relative stability of the NTK

Appendix G

Scaling Description of Generalization with Numer of Parameters in Deep Learning

G.1 Robustness of the boundaries distance $\delta(x)$ estimate

Fig.G.1.1 shows that the linear estimate for the distance $\delta(x)$ between two decision boundaries, $\delta(x) = \delta f(x)/\|\nabla f(x)\|$, holds for Relu nonlinear function and improves as $N \to \infty$.

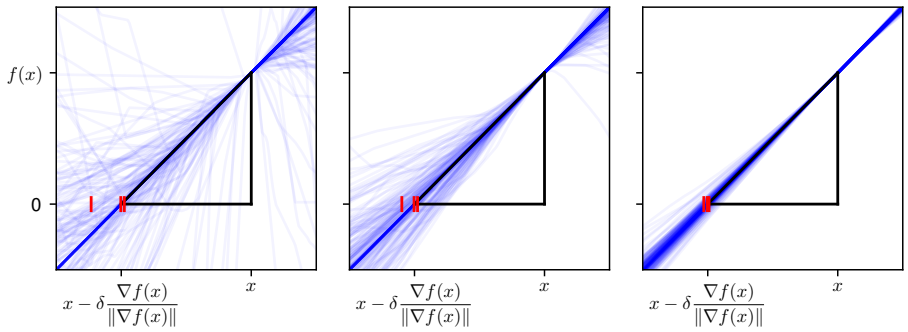


Figure G.1.1: Value of the output function f, in the direction of its gradient starting from x. Here 200 curves are shown, corresponding to 200 data x in the test set within the decision boundaries $f_N = 0$ and $\bar{f}_N = 0$ — i.e. $f_N(x)\bar{f}_N(x) < 0$. If the linear prediction is exact, then we expect $f(x - \delta \frac{\nabla f(x)}{\|\nabla f(x)\|}) = 0$ where $\delta = \delta f(x)/\|\nabla f(x)\|$. This prediction becomes accurate for large N. To make this statement quantitative, The 25%, 50%, 75% percentile of the intersection with zero are indicated with red ticks. Even for small N the interval between the ticks is small, so that the prediction is typically accurate. From left to right $N = 938, 13623, 6414815$. Here $d = 10$, $L = 5$ and $P = 10k$.

Fig.G.1.2 demonstrates the validity of the estimate of the typical distance between two boundary decisions presented in the main text $\delta \sim \|\delta f\|_\mu / \|\nabla f\|_\mu$, where μ corresponds to the uniform measure on all the test points.

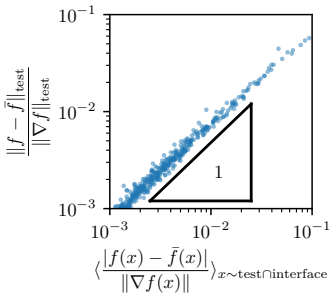


Figure G.1.2: Test for the estimate of the distance δ between the boundary decision of f and $\bar{f}$. Each point is measured from a single ensemble average of various sizes. Here $d = 30$, $h = 60$, $L = 5$, $N = 16k$ and $P = 10k$.

G.2 Central limit theorem of the NTK

In this section, we present a heuristic for the finite-size effects that are displayed by the NTK at initialization: informally, this is the central limit theorem counterpart to the NTK asymptotic result, which can be viewed as a law of large numbers. A rigorous derivation, including the behavior during training, is beyond the scope of this paper and will be presented in [106].

The NTK (see Eq.7.5.1) can be re-written as:

$$\sum_\alpha \left[1+\frac{1}{h}\sum_{\beta\in v^-(\alpha)} a_\beta(x)a_\beta(x') \right] \left[g'(b_\alpha(x))g'(b_\alpha(x'))\frac{\partial f(x)}{\partial a_\alpha}\frac{\partial f(x')}{\partial a_\alpha} \right] \tag{G.2.1}$$

where $a_\alpha(x) = g(b_\alpha(x))$ is the activity of neuron α when data x is shown, while $b_\alpha(x)$ is its pre-activity and $v^-(\alpha)$ is the set of h neurons in the layer preceding α. The first bracket converges to a well-defined limit described by a so-called activation kernel, see [159, 37, 105] which use a law of large numbers to prove this fact. The second bracket has fluctuations of order of its mean. The normalization is chosen such that each layer contributes a finite amount to the kernel, so that the mean is of order $1/h$. For a given hidden layer, the contributions of two neurons can be shown to have a covariance that is positive and decays as $1/h^3$, and thus does not affect the scaling expected from the central limit theorem for uncorrelated variables. For a rectangular network, this suggests that fluctuations associated with the contribution of one layer to the kernel is of order $1/\sqrt{h} \sim N^{-1/4}$.

G.3 Fluctuations of output function for the mean square error loss

In this section, we discuss the fluctuations of the output function after training for the mean square error loss: $C(f) = \frac{1}{2P}\sum_i |y_i - f(x_i)|^2$. We first investigate the variance of f_N^t in the limit $N \to \infty$, then we explain the deviations due to finite size effects, at last we discuss the hing loss case.

Infinite width

Let us first study the variance of f_N^t in the limit $N \to \infty$. In this limit the function $f_\infty^{t=0}$ at initialization follows a centered Gaussian distribution described by a covariance kernel Σ. During training however, the dynamics of f_∞^t is described by a deterministic kernel $\Theta_\infty^{(L)}$, the Neural Tangent Kernel (NTK):

$$\partial_t f_\infty^t(x) = \frac{1}{P}\sum_i \Theta_\infty(x, x_i)\left(y_i - f_\infty^t(x_i)\right).$$

If the NTK is positive definite (which is proven when the inputs all lie on the unit circle and the non-linearity is not a polynomial), the network reaches a global minimum at the end of training $t \to \infty$. In particular the values of the function on training set are deterministic: $f_\infty^{t=\infty}(x_i) = y_i$. The values of the function outside the training set can be studied using the vector of values of f_∞^t on the training set $\tilde{y}^t = (f_\infty^t(x_i))_{i=1,\dots P}$. Denoting by $\tilde{\Theta}_\infty = (\Theta_\infty(x_i, x_j))_{ij}$ the empirical Gram matrix:

$$y = \tilde{y}^{t=\infty} = \tilde{y}^{t=0} + \frac{1}{P}\int_0^\infty \tilde{\Theta}_\infty(y - \tilde{y}^t)dt,$$

so that

$$\frac{1}{P}\int_0^\infty (y - \tilde{y}^t)dt = \tilde{\Theta}_\infty^{-1}\left(y - \tilde{y}^{t=0}\right) = \tilde{\Theta}_\infty^{-1} y - \tilde{\Theta}_\infty^{-1}\tilde{y}^{t=0}.$$

These two terms represent the fact that the network needs to learn the labels y and forget the random initialization. We can therefore give a formula for the values outside the training set, using the vector $\tilde{\Theta}_{\infty,x} = (\Theta_\infty(x, x_i))_{i=1,\dots P}$:

$$\begin{aligned} f_\infty^t(x) &= f_\infty^{t=0}(x) + \tilde{\Theta}_{\infty,x}\frac{1}{P}\int_0^\infty (y - \tilde{y}^t)dt \\ &= f_\infty^{t=0}(x) - \tilde{\Theta}_{\infty,x}\tilde{\Theta}_\infty^{-1}\tilde{y}^{t=0} + \tilde{\Theta}_{\infty,x}\tilde{\Theta}_\infty^{-1} y. \end{aligned} \tag{G.3.1}$$

The first two terms are random, but they partly cancel each other, their sum is a centered Gaussian distribution with zero variance on the training set and a small variance for points close to the training set: the more training data points used, the lower the variance at initialization. The last term is equal to the kernel regression on y with respect to the NTK, it is not random.

This shows that even in the infinite-width limit, $f_\infty^{t=\infty}$ has some variance which is due to the variance of $f_\infty^{t=0}$ at initialization. Yet, in the setup where the number of data points is large enough, the variance due to initialization almost vanishes during training and the scaling of the variance due to finite-size effects in N will appear in the last term.

Finally, note that Eq.G.3.1 of this S.M. implies that $f_\infty^t(x)$ is smooth if both $\Theta_\infty(x, x')$ and $f_\infty^{t=0}(x)$ are smooth functions of x (this implication holds true for other choices of loss function). $\Theta_\infty(x, x')$ is smooth if the activation function is smooth [105], and so does $f_\infty^{t=0}(x)$ which is then a Gaussian function of smooth covariance $\Sigma(x, x')$. For Relu neurons, $\Theta_\infty(x, x')$ displays a cusp at $x = x'$ while $\Sigma(x, x')$ is smooth, so $f_\infty^t(x)$ is smooth except on the training set, as supported by Figure 1 of this S.M.

Finite width

For a finite width N, the training is also described by the NTK Θ_N^t which is random at initialization and varies during training because it depends on the parameters. The integral formula becomes

$$f_N^t(x) = f_N^{t=0}(x) + \int_0^\infty \tilde{\Theta}_{N,x}^t (y - \tilde{y}^t) dt$$

However the noise at initialization is $\Omega(N^{-1/4})$, whereas the rate of change is only $\Omega(N^{-1/2})$ [106], we can therefore make the approximation

$$f_N^t(x) = f_N^{t=0}(x) + \tilde{\Theta}_{N,x}^{t=0} \int_0^\infty (y - \tilde{y}^t) dt + \mathcal{O}(N^{-1/2}).$$

Assuming that there are enough parameters such that the Gram matrix $\tilde{\Theta}_N^{t=0}$ is invertible, we can again decompose the integral into two terms:

$$\int_0^\infty (y - \tilde{y}^t) dt = \tilde{\Theta}_N^{-1} y - \tilde{\Theta}_N^{-1} \tilde{y}^{t=0} + \mathcal{O}(N^{-1/2}),$$

such that

$$f_N^t(x) = f_N^{t=0}(x) - \tilde{\Theta}_{N,x}^{t=0} \tilde{\Theta}_N^{-1} \tilde{y}^{t=0} + \tilde{\Theta}_{N,x}^{t=0} \tilde{\Theta}_N^{-1} y + \mathcal{O}(N^{-1/2}). \tag{G.3.2}$$

Here again the first two terms almost cancel each other, but the third term is random due to the randomness of the NTK which is of order $\mathcal{O}(N^{-1/4})$, as needed.

Hinge Loss

For the hinge loss set-up, we do not have such a strong constraint on the value of the function $f_N^{t=\infty}$ on the training set $\tilde{y}^{t=\infty}$ as for regression, but we still know that they must satisfy the margin constraints

$$\tilde{y}_i^{t=\infty} y_i > 1.$$

The vector $\tilde{y}^{t=\infty}$ is therefore random for the hinge loss as a result of the random initialization of $f_N^{t=0}$ and the fluctuations of the NTK. Again it is natural to assume the first type of fluctuations to be subdominant and the second type to be of order $\mathcal{O}(N^{-1/4})$.

Appendix H

Implicit Regularization of Random Feature Models

We organize the Supplementary Material (Supp. Mat.) as follows:

- In Section H.1, we present the details for the numerical results presented in the main text (and in the Supp. Mat.).
- In Section H.2, we present additional experiments and some discussions.
- In Section H.3, we present the proofs of the mathematical results presented in the main text.

H.1 Experimental Details

The experimental setting consists of N training and N_{tst} test datapoints $\{(x_i, y_i)\}_{i=1}^{N+N_{\text{tst}}} \in \mathbb{R}^d \times \mathbb{R}$. We sample P Gaussian features $f^{(1)}, \dots, f^{(P)}$ of $N + N_{\text{tst}}$ dimension with zero mean and covariance matrix entries thereof $C_{i,j} = K(x_i, x_j)$ where $K(x, x') = \exp(-\|x - x'\|^2/\ell)$ is a Radial Basis Function (RBF) Kernel with lengthscale ℓ. The extended data matrix $\bar{F} = \frac{1}{\sqrt{P}}[f^{(1)}, \dots, f^{(P)}]$ of size $(N + N_{\text{tst}}) \times P$ is decomposed into two matrices: the (training) data matrix $F = \bar{F}_{[:N,:]}$ of size $N \times P$, and a test data matrix $F_{\text{tst}} = \bar{F}_{[N:,:]}$ of size $N_{\text{tst}} \times P$ so that $\bar{F} = [F; F_{\text{tst}}]$. For a given ridge λ, we compute the optimal solution using the data matrix F, i.e. $\hat{\theta} = F^T \left(FF^T + \lambda \mathrm{I}_N\right)^{-1} y$ and obtain the predictions on the test datapoints $\hat{y}_{\text{tst}} = F_{\text{tst}} F^T \left(FF^T + \lambda \mathrm{I}_N\right)^{-1} y$.

Using the procedure above, we performed the following experiments:

Experiments with Sinusoidal data

We consider a dataset of $N = 4$ training datapoints $(x_i, \sin(x_i)) \in [0, 2\pi) \times [-1, 1]$ and $N_{\text{tst}} = 100$ equally spaced test data points in the interval $[0, 2\pi)$. In this experiment, the lengthscale of the RBF Kernel is $\ell = 2$. We compute the average and standard deviation the λ-RF predictor using 500 samplings of $\bar{F}$ (see Figure 1 in the main text and Figure H.2.1 in the Supp. Mat.).

MNIST experiments

We sample $N = 100$ and $N_{\text{tst}} = 100$ images of digits 7 and 9 from the MNIST dataset (image size $d = 24 \times 24$, edge pixels cropped, all pixels rescaled down to $[0, 1]$ and recentered around the mean value) and label each of them with $+1$ and -1 labels, respectively. In this experiment, the lengthscale of the RBF Kernel is $\ell = d\ell_0$ where $\ell_0 = 0.2$. We approximate the expected λ-RF predictor on the test datapoints using the average of $\hat{y}_{\text{tst}}$ over 50 instances of $\bar{F}$ and compute the MSE (see Figures 2, 3 in the main text; in the ridgeless case –$\lambda = 10^{-4}$ in our experiments– when P is close to N, the average is over 500 instances). In Figure 4 of the main text, using $N_{\text{tst}} = 100$ test points, we compare two predictors trained over $N = 100$ and $N = 1000$ training datapoints.

Random Fourier Features

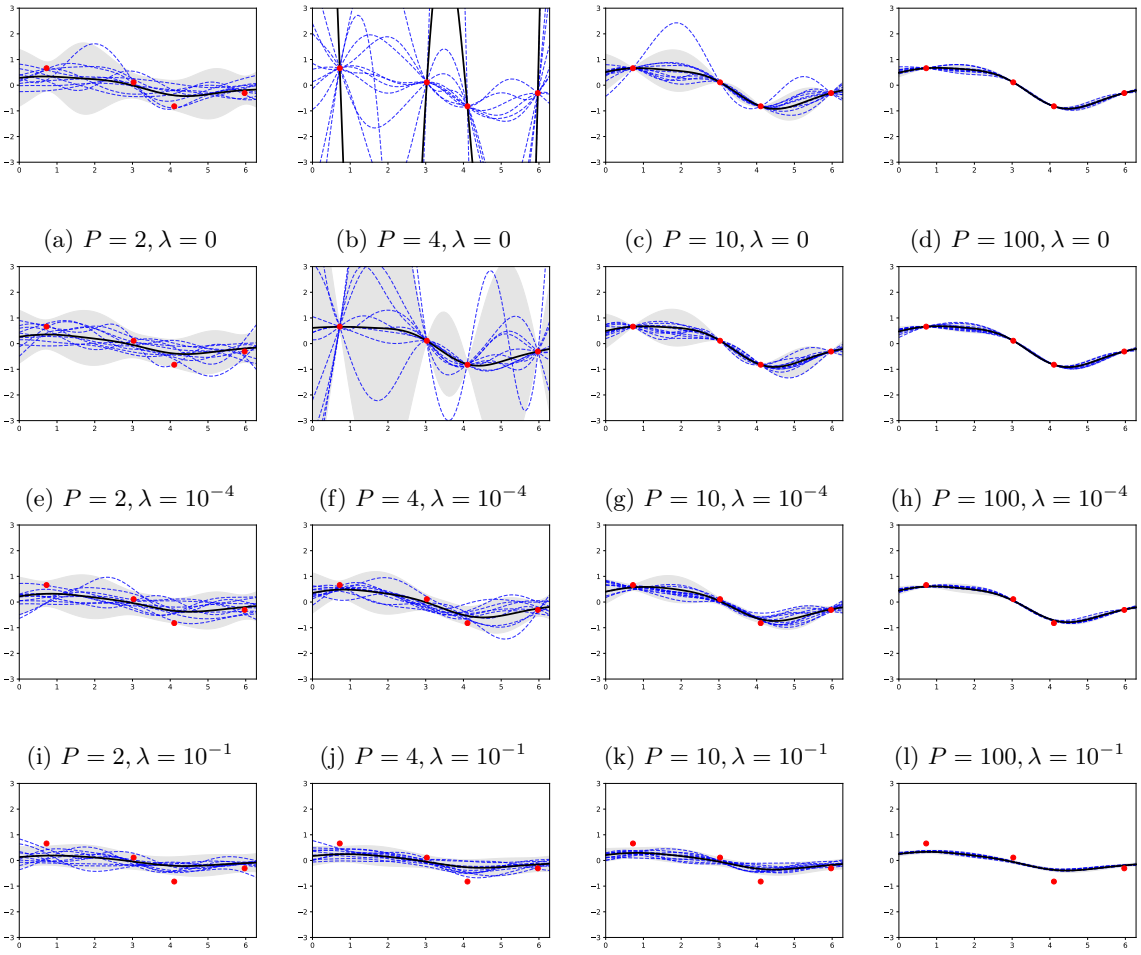
We sample random Fourier Features corresponding to the RBF Kernel with lengthscale $\ell = d\ell_0$ where $\ell_0 = 0.2$ (same as above) and consider the same dataset as in the MNIST experiment. The extended data matrix $\bar{F}$ for Fourier features is obtained as follows: we sample d-dimensional i.i.d. centered Gaussians $w^{(1)}, \dots, w^{(P)}$ with standard deviation $\sqrt{2/\ell}$, sample $b^{(1)}, \dots, b^{(P)}$ uniformly in $[0, 2\pi)$, and define $\bar{F}_{i,j} = \sqrt{\frac{2}{P}} \cos(x_i^T w^{(j)} + b^{(j)})$. We approximate the expected Fourier Features predictor on the test datapoints using the average of $\hat{y}_{\text{tst}}$ over 50 instances of $\bar{F}$ (see Figure H.2.5).

H.2 Additional Experiments

We present the following complementary simulations:

- In Section H.2, we present the distribution of the λ-RF predictor for the selected P and λ.
- In Section H.2, we present the evolution of $\tilde{\lambda}$ and its derivative $\partial_\lambda \tilde{\lambda}$ for different eigenvalue spectra.
- In Section H.2, we show the evolution of the eigenvalue spectrum of $\mathbb{E}[A_\lambda]$.
- In Section H.2, we present numerical experiments on MNIST using random Fourier features.

Distribution of the RF predictor



(a) $P = 2, \lambda = 0$ (b) $P = 4, \lambda = 0$ (c) $P = 10, \lambda = 0$ (d) $P = 100, \lambda = 0$

(e) $P = 2, \lambda = 10^{-4}$ (f) $P = 4, \lambda = 10^{-4}$ (g) $P = 10, \lambda = 10^{-4}$ (h) $P = 100, \lambda = 10^{-4}$

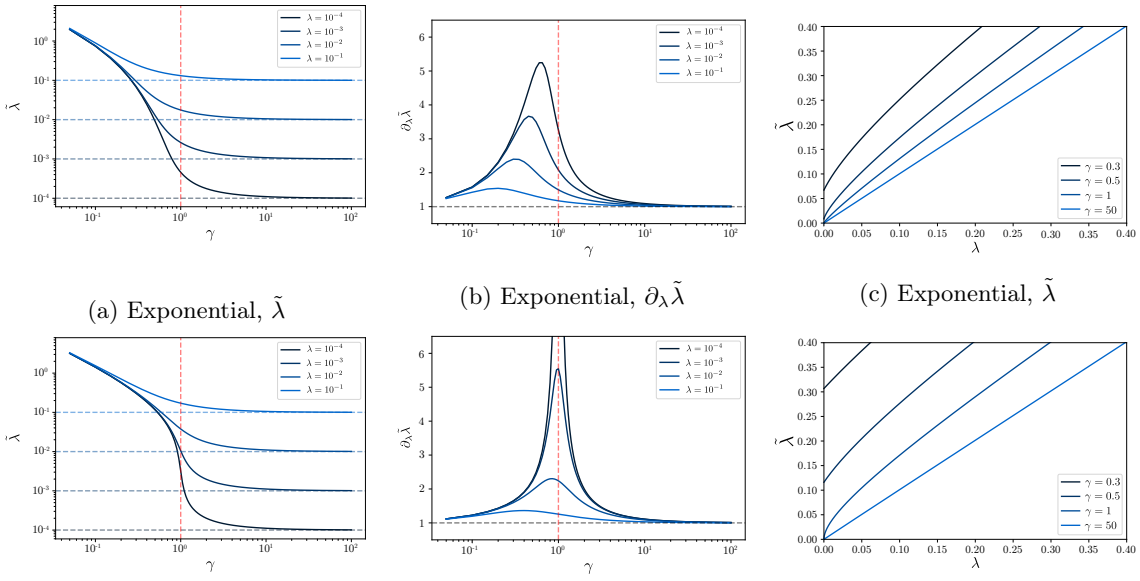
(i) $P = 2, \lambda = 10^{-1}$ (j) $P = 4, \lambda = 10^{-1}$ (k) $P = 10, \lambda = 10^{-1}$ (l) $P = 100, \lambda = 10^{-1}$

(m) $P = 2, \lambda = 1$ (n) $P = 4, \lambda = 1$ (o) $P = 10, \lambda = 1$ (p) $P = 100, \lambda = 1$

Figure H.2.1: *Distribution of the RF predictor.* Red dots represent a sinusoidal dataset $y_i = \sin(x_i)$ for $N = 4$ points x_i in $[0, 2\pi)$. For $P \in \{2, 4, 10, 100\}$ and $\lambda \in \{0, 10^{-4}, 10^{-1}, 1\}$, we sample ten RF predictors (blue dashed lines) and compute empirically the average RF predictor (black lines) with ± 2 standard deviations intervals (shaded regions).

Evolution of the Effective Ridge $\tilde{\lambda}$

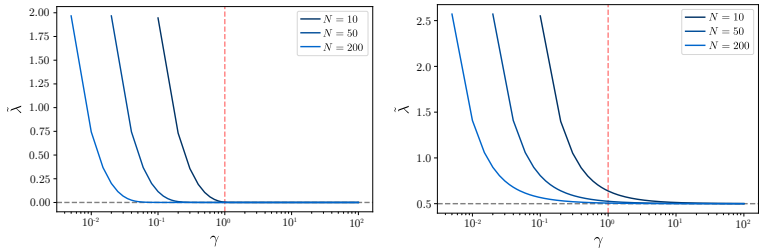
In Figure H.2.2, we show how the effective ridge $\tilde{\lambda}$ and its derivative $\partial_\lambda \tilde{\lambda}$ evolve for the selected eigenvalue spectra with various decays (exponential or polynomial) as a function of γ and λ. In Figure H.2.3, we compare the evolution of $\tilde{\lambda}$ for various N.



(a) Exponential, $\tilde{\lambda}$ (b) Exponential, $\partial_\lambda \tilde{\lambda}$ (c) Exponential, $\tilde{\lambda}$

(d) Polynomial, $\tilde{\lambda}$ (e) Polynomial, $\partial_\lambda \tilde{\lambda}$ (f) Polynomial, $\tilde{\lambda}$

Figure H.2.2: *Evolution of the effective ridge* $\tilde{\lambda}$ *and its derivative* $\partial_\lambda \tilde{\lambda}$ *for various levels of ridge* λ *(or* γ*) and for* $N = 20$. We consider two different decays for $d_1, \ldots, d_N$: (i) exponential decay in i (i.e. $d_i = e^{-\frac{(i-1)}{2}}$, top plots) and (ii) polynomial decay in i (i.e. $d_i = \frac{1}{i}$, bottom plots).



(a) $\lambda = 10^{-4}$ (b) $\lambda = 0.5$

Figure H.2.3: *Evolution of effective ridge* $\tilde{\lambda}$ *as a function of* γ *for two ridges* ***(a)*** $\lambda = 10^{-4}$ *and* ***(b)*** $\lambda = 0.5$ *and for various* N. We consider an exponential decay for $d_1, \dots, d_N$, i.e. $d_i = e^{-\frac{(i-1)}{2}}$.

Eigenvalues of A_λ

The (random) prediction $\hat{y}$ on the training data is given by $\hat{y} = A_\lambda y$ where $A_\lambda = F(F^T F + \lambda I)^{-1} F^T$. The average λ-RF predictor is $\mathbb{E}[\hat{f}_\lambda^{(RF)}(x)] = K(x, X)K(X, X)^{-1}\mathbb{E}[A_\lambda]y$. We denote by $\tilde{d}_1, \dots \tilde{d}_N$ the eigenvalues of $\mathbb{E}[A_\lambda]$. By Proposition H.3.7, the $\tilde{d}_i$'s converge to the eigenvalues $\frac{d_1}{d_1+\tilde{\lambda}}, \dots, \frac{d_N}{d_N+\tilde{\lambda}}$ of $K(K + \tilde{\lambda} I_N)^{-1}$ as P goes to infinity. We illustrate the evolution of $\tilde{d}_i$ and their convergence to $\frac{d_i}{d_i+\tilde{\lambda}}$ for two different eigenvalue spectrums $d_1, \dots d_N$.

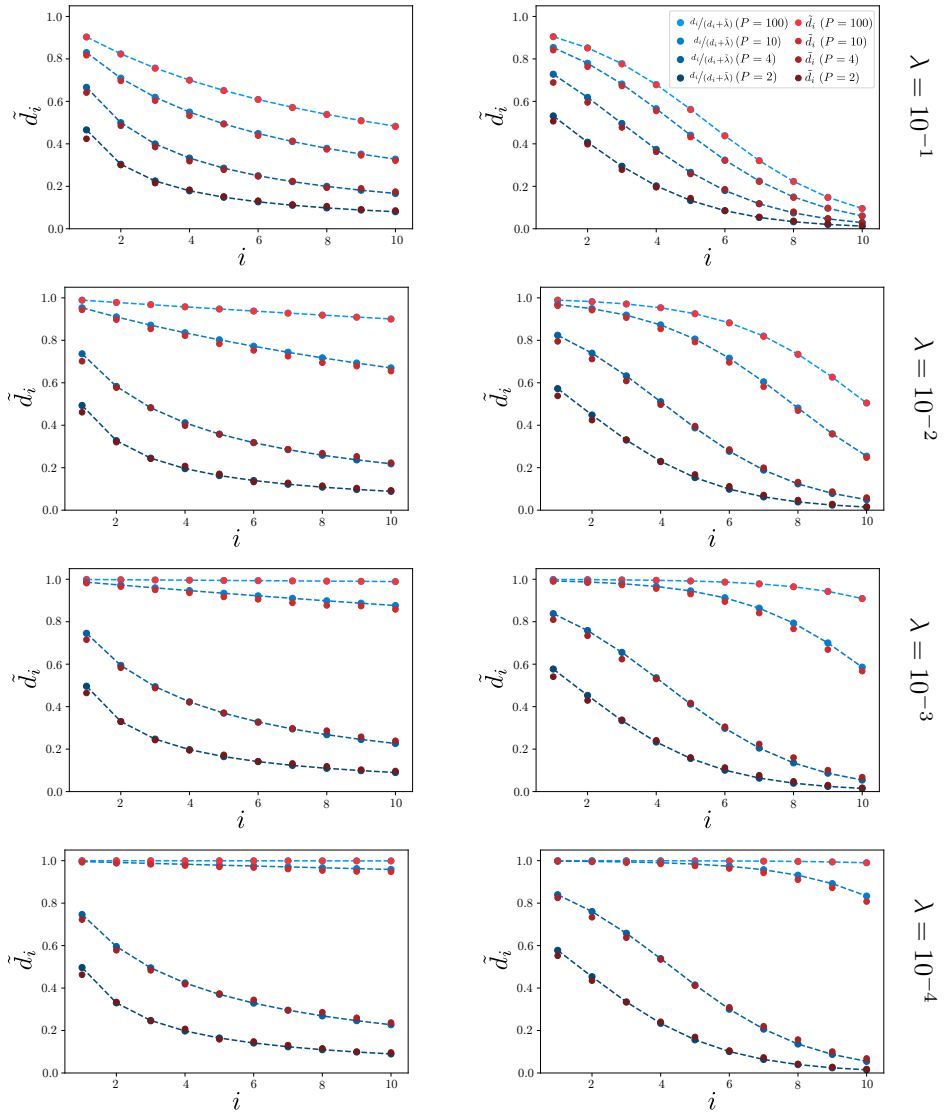


Figure H.2.4: *Eigenvalues* $\tilde{d}_1, \dots \tilde{d}_N$ *(red dots) vs. eigenvalues* $\frac{d_1}{d_1+\tilde{\lambda}}, \dots, \frac{d_N}{d_N+\tilde{\lambda}}$*(blue dots) for* $N = 10$. We consider various values of P and two different decays for $d_1, \dots, d_N$: (i) exponential decay in i, i.e. $d_i = e^{-\frac{(i-1)}{2}}$ (right plots) and (ii) polynomial decay in i, i.e. $d_i = \frac{1}{i}$ (left plots).

Average Fourier Features Predictor

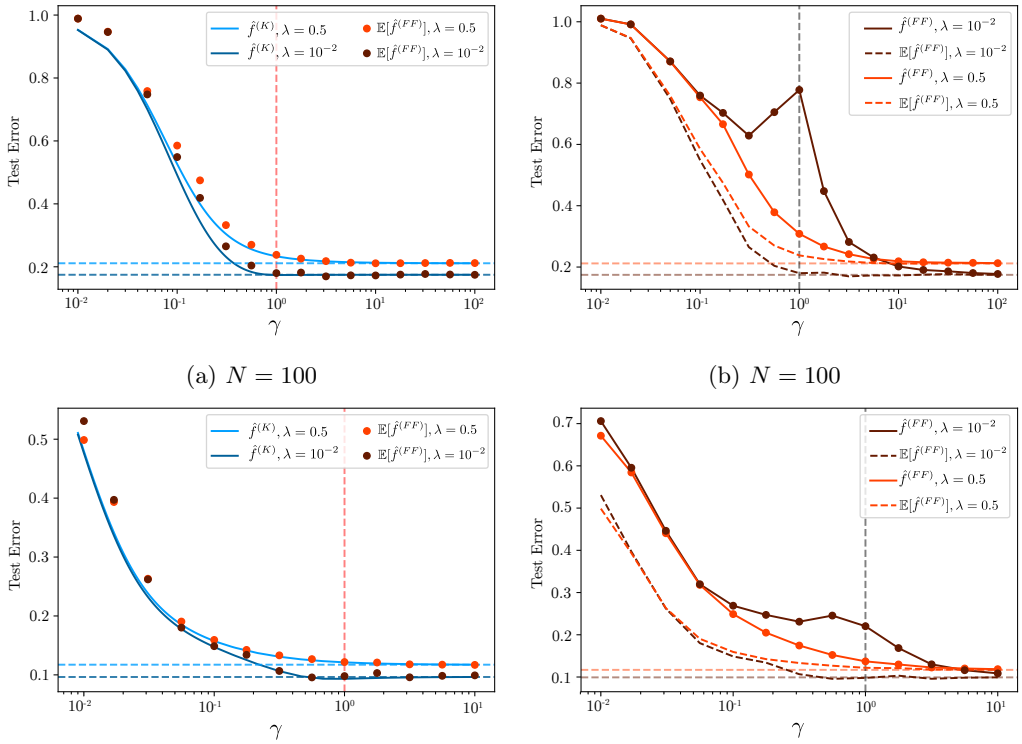
The Fourier Features predictor λ-FF is $\hat{f}^{(FF)}(x) = \frac{1}{\sqrt{P}} \sum_{j=1}^{P} \hat{\theta}_j \phi^{(j)}(x)$ where $\phi^{(j)}(x) = \cos(x^T w^{(j)} + b^{(j)})$ and $\hat{\theta} = F^T \left(FF^T + \lambda \mathrm{I}_N\right)^{-1} y$ with the data matrix F as described in Section H.1.

We investigate how close the average λ-FF predictor is to the $\tilde{\lambda}$-KRR predictor and we observe the following:

1. The difference of the test errors of the two predictors decreases as γ increases.

2. In the overparameterized regime, i.e. $P \geq N$, the test error of the $\tilde{\lambda}$-KRR predictor matches with the test error of the λ-FF predictor.

3. For $N = 1000$, strong agreement between the two test errors is observed already for $\gamma > 0.1$. We also observe that Gaussian features achieve lower (or equal) test error than the Fourier features for all γ in our experiments.



(a) $N = 100$ (b) $N = 100$

(c) $N = 1000$ (d) $N = 1000$

Figure H.2.5: *Comparision of the test errors of the average λ-FF predictor and the $\tilde{\lambda}$-KRR predictor.* In **(a)** and **(c)**, the test errors of the average λ-FF predictor and of the $\tilde{\lambda}$-KRR predictor are reported for various ridge for $N = 100$ and $N = 1000$ MNIST data points (top and bottom rows). In **(b)** and **(d)**, the average test error of the λ-FF predictor and the test error of its average are reported.

H.3 Proofs

Gaussian Random Features

Proposition H.3.1. *Let $\hat{f}_\lambda^{(RF)}$ be the λ-RF predictor and let $\hat{y} = F\hat{\theta}$ be the prediction vector on training data, i.e. $\hat{y}_i = \hat{f}_\lambda^{(RF)}(x_i)$. The process $\hat{f}_\lambda^{(RF)}$ is a mixture of Gaussians: conditioned on F, we have that $\hat{f}_\lambda^{(RF)}$ is a Gaussian process. The mean and covariance of $\hat{f}_\lambda^{(RF)}$ conditioned on F are given by*

$$\mathbb{E}[\hat{f}_\lambda^{(RF)}(x)|F] = K(x,X)K(X,X)^{-1}\hat{y}, \tag{H.3.1}$$

$$\mathrm{Cov}[\hat{f}_\lambda^{(RF)}(x), \hat{f}_\lambda^{(RF)}(x')|F] = \frac{\|\hat{\theta}\|^2}{P}\tilde{K}(x,x') \tag{H.3.2}$$

where $\tilde{K}(x,x') = K(x,x') - K(x,X)K(X,X)^{-1}K(X,x')$ denotes the posterior covariance kernel.

Proof. Let $F = (\frac{1}{\sqrt{P}} f^{(j)}(x_i))_{i,j}$ be the $N \times P$ matrix of values of the random features on the training set. By definition, $\hat{f}_\lambda^{(RF)} = \frac{1}{\sqrt{P}} \sum_{p=1}^P \hat{\theta}_p f^{(p)}$. Conditioned on the matrix F, the optimal parameters $(\hat{\theta}_p)_p$ are not random and $(f^{(p)})_p$ is still Gaussian, hence, conditioned on the matrix F, the process $\hat{f}_\lambda^{(RF)}$ is a mixture of Gaussians. Moreover, conditioned on the matrix F, for any p, p', $f^{(p)}$ and $f^{(p')}$ remain independent, hence

$$\begin{aligned}
\mathbb{E}\left[\hat{f}_\lambda^{(RF)}(x) \mid F\right] &= \frac{1}{\sqrt{P}} \sum_{p=1}^P \hat{\theta}_p \mathbb{E}\left[f^{(p)}(x) \mid f_N^{(p)}\right] \\
\mathrm{Cov}\left[\hat{f}_\lambda^{(RF)}(x), \hat{f}_\lambda^{(RF)}(x') \mid F\right] &= \frac{1}{P} \sum_{p=1}^P \hat{\theta}_p^2 \mathrm{Cov}\left[f^{(p)}(x), f^{(p)}(x') \mid f_N^{(p)}\right].
\end{aligned}$$

where we have set $f_N^{(p)} = (f^{(p)}(x_i))_i \in \mathbb{R}^N$. The value of $\mathbb{E}\left[f^{(p)}(x) \mid f_N^{(p)}\right]$ and $\mathrm{Cov}\left[f^{(p)}(x), f^{(p)}(x') \mid f_N^{(p)}\right]$ are obtained from classical results on Gaussian conditional distributions [53]:

$$\begin{aligned}
\mathbb{E}\left[f^{(p)}(x) \mid f_N^{(p)}\right] &= K(x,X)K(X,X)^{-1} f_N^{(p)}, \\
\mathrm{Cov}\left[f^{(p)}(x), f^{(p)}(x') \mid f_N^{(p)}\right] &= \tilde{K}(x,x'),
\end{aligned}$$

where $\tilde{K}(x,x') = K(x,x') - K(x,X)K(X,X)^{-1}K(X,x')$. Thus, conditioned on F, the predictor $\hat{f}_\lambda^{(RF)}$ has expectation:

$$\mathbb{E}\left[\hat{f}_\lambda^{(RF)}(x) \mid F\right] = K(x,X)K(X,X)^{-1} \frac{1}{\sqrt{P}} \sum_{p=1}^P \hat{\theta}_p f_N^{(p)} = K(x,X)K(X,X)^{-1}\hat{y}$$

and covariance:

$$\mathrm{Cov}\left[\hat{f}_\lambda^{(RF)}(x), \hat{f}_\lambda^{(RF)}(x') \mid F\right] = \frac{1}{P} \sum_{p=1}^P \hat{\theta}_p^2 \tilde{K}(x,x') = \frac{\|\hat{\theta}\|^2}{P} \tilde{K}(x,x').$$

□

Generalized Wishart Matrix

Setup. In this section, we consider a fixed deterministic matrix K of size $N \times N$ which is diagonal positive semi-definite, with eigenvalues $d_1, \dots, d_N$. We also consider a $P \times N$ random matrix W with i.i.d. standard Gaussian entries.

The key object of study is the $P \times P$ generalized Wishart random matrix $F^T F = \frac{1}{P} W K W^T$ and in particular its Stieltjes transform defined on $z \in \mathbb{C} \setminus \mathbb{R}^+$, where $\mathbb{R}^+ = [0, +\infty[$:

$$m_P(z) = \frac{1}{P} \mathrm{Tr} \left[\left(F^T F - z \mathrm{I}_P \right)^{-1} \right] = \frac{1}{P} \mathrm{Tr} \left[\left(\frac{1}{P} W K W^T - z \mathrm{I}_P \right)^{-1} \right],$$

where K is a fixed positive semi-definite matrix.

Since $F^T F$ has positive real eigenvalues $\lambda_1, \dots, \lambda_P \in \mathbb{R}_+$, and

$$m_P(z) = \frac{1}{P} \sum_{p=1}^{P} \frac{1}{\lambda_p - z},$$

we have that for any $z \in \mathbb{C} \setminus \mathbb{R}^+$,

$$|m_P(z)| \leq \frac{1}{d(z, \mathbb{R}_+)},$$

where $d(z, \mathbb{R}_+) = \inf \{ |z - y|, y \in \mathbb{R}^+ \}$ is the distance of z to the positive real line. More precisely, $m_P(z)$ lies in the convex hull $\Omega_z = \mathrm{Conv} \left(\left\{ \frac{1}{d - z} : d \in \mathbb{R}_+ \right\} \right)$. As a consequence, the argument $\arg \left(m_P(z) \right) \in (-\pi, \pi)$ lies between 0 and $\arg \left(-\frac{1}{z} \right)$, i.e. $m_P(z)$ lies in the cone spanned by 1 and $-\frac{1}{z}$.

Our first lemma implies that the Stieljes transform concentrates around its mean as N and P go to infinity with $\gamma = \frac{P}{N}$ fixed.

Lemma H.3.2. *For any integer $m \in \mathbb{N}$ and any $z \in \mathbb{C} \setminus \mathbb{R}^+$, we have*

$$\mathbb{E} \left[|m_P(z) - \mathbb{E} \left[m_P(z) \right]|^m \right] \leq \mathbf{c} P^{-\frac{m}{2}},$$

where $\mathbf{c}$ depends on z, γ, and m only.

Proof. The proof follows Step 1 of [11]. Let $w_1, ..., w_N$ be the columns of W from left to right. Let us introduce the $P \times P$ matrices $B(z) = \frac{1}{P} W K W^T - z \mathrm{I}_P$ and $B_{(i)}(z) = \frac{1}{P} W_{(i)} K_{(i)} W_{(i)}^T - z \mathrm{I}_P$ where $W_{(i)}$ is the $P \times (N-1)$ submatrix of W obtained by removing its i-th column w_i, and $K_{(i)}$ is the $(N-1) \times (N-1)$ submatrix of K obtained by removing both its i-th column and i-th row. Since the eigenvalues of $W K W^T$ and $W_{(i)} K_{(i)} W_{(i)}^T$ are all real and positive, $B(z)$ and $B_{(i)}(z)$ are invertible matrices for $z \notin \mathbb{R}^+$.

Noticing that

$$B(z) = \frac{1}{P} W K W^T - z \mathrm{I}_P = \frac{1}{P} W_{(i)} K_{(i)} W_{(i)}^T - z \mathrm{I}_P + \frac{d_i}{P} w_i w_i^T$$

is a rank one perturbation of the matrix $B_{(i)}(z)$, by the Sherman–Morrison's formula, the inverse of $B(z)$ is given by:

$$B(z)^{-1} = \left(B_{(i)}(z) \right)^{-1} - \frac{d_i}{P} \frac{1}{1 + \frac{d_i}{P} w_i^T \left(B_{(i)}(z) \right)^{-1} w_i} \left(B_{(i)}(z) \right)^{-1} w_i w_i^T \left(B_{(i)}(z) \right)^{-1}.$$

We denote $\mathbb{E}_i$ the conditional expectation given $w_{i+1}, ..., w_N$. We have $\mathbb{E}_0[m_P(z)] = m_P(z)$ and $\mathbb{E}_N[m_P(z)] = \mathbb{E}[m_P(z)]$. As a consequence, we get:

$$\begin{aligned} m_P(z) - \mathbb{E}[m_P(z)] &= \sum_{i=1}^{N} \left(\mathbb{E}_{i-1}[m_P(z)] - \mathbb{E}_i[m_P(z)]\right) \\ &= \frac{1}{P}\sum_{i=1}^{N} \left(\mathbb{E}_{i-1} - \mathbb{E}_i\right)\left[\mathrm{Tr}\left(B(z)^{-1}\right)\right] \\ &= \frac{1}{P}\sum_{i=1}^{N} \left(\mathbb{E}_{i-1} - \mathbb{E}_i\right)\left[\mathrm{Tr}\left(B(z)^{-1}\right) - \mathrm{Tr}\left(B_{(i)}(z)^{-1}\right)\right]. \end{aligned}$$

The last equality comes from the fact that $\mathrm{Tr}\left(B_{(i)}(z)^{-1}\right)$ does not depend on w_i, hence

$$\mathbb{E}_{i-1}\left[\mathrm{Tr}\left(B_{(i)}(z)^{-1}\right)\right] = \mathbb{E}_i\left[\mathrm{Tr}\left(B_{(i)}(z)^{-1}\right)\right].$$

Let $g_i : \mathbb{C} \setminus \mathbb{R}^+ \to \mathbb{C}$ be the holomorphic function given by $g_i(z) := \frac{1}{P} w_i^T \left(B_{(i)}(z)\right)^{-1} w_i$. Its derivative is given by $g_i'(z) = \frac{1}{P} w_i^T \left(B_{(i)}(z)\right)^{-2} w_i$. Hence

$$\begin{aligned} \mathrm{Tr}\left(B(z)^{-1}\right) - \mathrm{Tr}\left(B_{(i)}(z)^{-1}\right) &= -\frac{\frac{d_i}{P}\mathrm{Tr}\left(\left(B_{(i)}(z)\right)^{-1} w_i w_i^T \left(B_{(i)}(z)\right)^{-1}\right)}{1 + d_i g_i(z)} \\ &= -\frac{d_i g_i'(z)}{1 + d_i g_i(z)}, \end{aligned}$$

where we used the cyclic property of the trace. We can now bound this difference:

$$\begin{aligned} \left|\mathrm{Tr}\left(B(z)^{-1}\right) - \mathrm{Tr}\left(B_{(i)}(z)^{-1}\right)\right| &= \left|\frac{d_i g_i'(z)}{1 + d_i g_i(z)}\right| \\ &\leq \left|\frac{w_i^T \left(B_{(i)}(z)\right)^{-2} w_i}{w_i^T \left(B_{(i)}(z)\right)^{-1} w_i}\right| \\ &\leq \max_w \left|\frac{w^T \left(B_{(i)}(z)\right)^{-2} w}{w^T \left(B_{(i)}(z)\right)^{-1} w}\right| \\ &\leq \| \left(B_{(i)}(z)\right)^{-1} \|_{op} = \max_j |\frac{1}{\nu_j - z}| \leq \frac{1}{d(z, \mathbb{R}^+)}, \end{aligned}$$

where ν_j are the eigenvalues of $\frac{1}{P} W_{(i)} K_{(i)} W_{(i)}^T$.

The sequence

$$\left(\left(\mathbb{E}_{N-i} - \mathbb{E}_{N-i+1}\right)\left[\mathrm{Tr}\left(B(z)^{-1}\right) - \mathrm{Tr}\left(B_{(N-i+1)}(z)^{-1}\right)\right]\right)_{i=1,...,N}$$

is a martingale difference sequence. Hence, by Burkholder's inequality, there exists a positive constant K_m such that

$$\begin{aligned}\mathbb{E}\left[|m_P(z) - \mathbb{E}\left[m_P(z)\right]|^m\right] &\leq K_m \frac{1}{P^m}\mathbb{E}\left[\left(\sum_{i=1}^{N}\left|\left[\mathbb{E}_{i-1} - \mathbb{E}_i\right]\left(\mathrm{Tr}\left(B(z)^{-1}\right) - \mathrm{Tr}\left(B_{(i)}(z)^{-1}\right)\right)\right|^2\right)^{\frac{m}{2}}\right] \\ &\leq K_m \frac{1}{P^m}\left(N\left(\frac{2}{d(z,\mathbb{R}_+)}\right)^2\right)^{\frac{m}{2}} \\ &\leq K_m \gamma^{-\frac{m}{2}}\left(\frac{2}{d(z,\mathbb{R}_+)}\right)^m P^{-\frac{m}{2}},\end{aligned}$$

hence the desired result with $\mathbf{c} = K_m\gamma^{-\frac{m}{2}}\left(\frac{2}{d(z,\mathbb{R}_+)}\right)^m$. □

The following lemma, which is reminiscent of Lemma 4.5 in [8], is a consequence of Wick's formula for Gaussian random variables and is key to prove Lemma C.4.

Lemma H.3.3. *If $A^{(1)}, \ldots, A^{(k)}$ are k square random matrices of size P independent from a standard Gaussian vector w of size P,*

$$\mathbb{E}\left[w^T A^{(1)} w w^T A^{(2)} w \ldots w^T A^{(k)} w\right] = \sum_{p\in\boldsymbol{P}_2(2k)}\sum_{\lfloor p\leq \mathrm{Ker}(i_1,\ldots,i_{2k})]i_1,\ldots,i_{2k}\in\{1,\ldots,P\}} \mathbb{E}\left[A^{(1)}_{i_1 i_2}\ldots A^{(k)}_{i_{2k-1}i_{2k}}\right], \tag{H.3.3}$$

where $\boldsymbol{P}_2(2k)$ is the set of pair partitions of $\{1,\ldots,2k\}$, $\leq$ is the coarser (i.e. $p \leq q$ if q is coarser than p), and for any $i_1,\ldots,i_{2k}$ in $\{1,\ldots,P\}$, $\mathrm{Ker}(i_1,\ldots,i_{2k})$ is the partition of $\{1,\ldots,2k\}$ such that two elements u and v in $\{1,...,2k\}$ are in the same block (i.e. pair) of $\mathrm{Ker}\,(i_1,\ldots,i_{2k})$ if and only if $i_u = i_v$.

Furthermore,

$$\begin{aligned}&\mathbb{E}\left[\left(w^T A^{(1)} w - \mathrm{Tr}\left(A^{(1)}\right)\right)\left(w^T A^{(2)} w - \mathrm{Tr}\left(A^{(2)}\right)\right)\ldots\left(w^T A^{(k)} w - \mathrm{Tr}\left(A^{(k)}\right)\right)\right] \\ &\qquad = \sum_{p\in:\boldsymbol{P}_2(2k):}\sum_{\lfloor p\leq \mathrm{Ker}(i_1,\ldots,i_{2k})]i_1,\ldots,i_{2k}\in\{1,\ldots,P\}} \mathbb{E}\left[A^{(1)}_{i_1 i_2}\ldots A^{(k)}_{i_{2k-1}i_{2k}}\right],\end{aligned} \tag{H.3.4}$$

where $:\boldsymbol{P}_2(2k):$ is the subset of partitions p in $\boldsymbol{P}_2(2k)$ for which $\{2j-1, 2j\}$ is not a block of p for any $j \in \{1,\ldots,k\}$.

Proof. Expanding the left-hand side of Equation (H.3.3), we obtain:

$$\mathbb{E}\left[\sum_{i_1,\ldots,i_{2k}\in\{1,\ldots,P\}} w_{i_1} A^{(1)}_{i_1 i_2} w_{i_2} w_{i_3} A^{(2)}_{i_3 i_4} w_{i_4}\ldots w_{i_{2k-1}} A^{(k)}_{i_{2k-1}i_{2k}} w_{i_{2k}}\right].$$

Using Wick's formula, we get:

$$\sum_{i_1,\ldots,i_{2k}\in\{1,\ldots,P\}}\sum_{\lfloor p\leq \mathrm{Ker}(i_1,\ldots,i_{2k})]p\in\boldsymbol{P}_2(2k),} \mathbb{E}\left[A^{(1)}_{i_1 i_2} A^{(2)}_{i_3 i_4}\ldots A^{(k)}_{i_{2k-1}i_{2k}}\right],$$

hence, interchanging the order of summation, we recover the left-hand side of Equation (H.3.3):

$$\sum_{p\in \boldsymbol{P}_2(2k)} \sum_{p\leq \mathrm{Ker}(i_1,\ldots,i_{2k})] i_1,\ldots,i_{2k}\in\{1,\ldots,P\}} \mathbb{E}\left[A^{(1)}_{i_1 i_2}\ldots A^{(k)}_{i_{2k-1}i_{2k}}\right].$$

We now prove Equation (H.3.4). Expanding the product, the left-hand side is equal to:

$$\sum_{I\subset\{1,\ldots,k\}} (-1)^{k-\#I}\mathbb{E}\left[\prod_{i\in I} w^T A^{(i)} w \prod_{i\notin I}\mathrm{Tr}(A^{(i)})\right].$$

Expanding the product and the trace, and using Wick's equation, we obtain: a

$$\sum_{I\subset\{1,\ldots,k\}} (-1)^{k-\#I} \sum_{i_1,\ldots,i_{2k}\in\{1,\ldots,P\}} \sum_{p\leq \mathrm{Ker}(i_1,\ldots,i_{2k})] p\in \boldsymbol{P}_2(2k), p\leq p_I} \mathbb{E}\left[A^{(1)}_{i_1 i_2}\ldots A^{(k)}_{i_{2k-1}i_{2k}}\right].$$

where p_I is the partition composed of blocks of size 2 given by $\{2l, 2l+1\}$ with $l\notin I$ and the rest of the indices contained in a single block. Interchanging the order of summation, we get:

$$\sum_{i_1,\ldots,i_{2k}\in\{1,\ldots,P\}} \sum_{p\leq \mathrm{Ker}(i_1,\ldots,i_{2k})] p\in \boldsymbol{P}_2(2k),} \mathbb{E}\left[A^{(1)}_{i_1 i_2}\ldots A^{(k)}_{i_{2k-1}i_{2k}}\right] \left[\sum_{p\leq p_I] I\subset\{1,\ldots,k\},} (-1)^{k-\#I}\right].$$

Since $\left[\sum_{I\subset\{1,\ldots,k\}, p\leq p_I}(-1)^{\#I}\right] = \delta_{\{I\subset[k], p\leq p_I\}=\{\{1,\ldots,k\}\}}$ and $\{I\subset[k], p\leq p_I\} = \{\{1,\ldots,k\}\}$ if and only if $p\in \boldsymbol{P}_2(2k)$, interchanging a last time the order of summation, we recover the left-hand side of Equation (H.3.4):

$$\sum_{p\in \boldsymbol{P}_2(2k):} \sum_{p\leq \mathrm{Ker}(i_1,\ldots,i_{2k})] i_1,\ldots,i_{2k}\in\{1,\ldots,P\}} \mathbb{E}\left[A^{(1)}_{i_1 i_2}\ldots A^{(k)}_{i_{2k-1}i_{2k}}\right].$$

□

For any $z\in\mathbb{C}\setminus\mathbb{R}^+$, we define the holomorphic function $g_i : \mathbb{C}\setminus\mathbb{R}^+\to\mathbb{C}$ by

$$g_i(z) = \frac{1}{P} w_i^T \left(\frac{1}{P} W_{(i)} K_{(i)} W_{(i)}^T - z\ I_P\right)^{-1} w_i,$$

where $W_{(i)}$ is the $P\times(N-1)$ submatrix of W obtained by removing its i-th column w_i, and $K_{(i)}$ is the $(N-1)\times(N-1)$ submatrix of K obtained by removing both its i-th column and i-th row. In the following lemma, we bound the distance of $g_i(z)$ to its mean. Then we prove that $\mathbb{E}[g_i(z)]$ is close to the expected Stieljes transform of K.

Lemma H.3.4. *The random function $g_i(z)$ satisfies:*

$$|\mathbb{E}[g_i(z)] - \mathbb{E}[m_P(z)]| \quad\leq\quad \frac{\mathbf{c_0}}{P},$$

$$\begin{aligned}\operatorname{Var}\left(g_i(z)\right) &\leq \frac{\mathbf{c_1}}{P},\\ \mathbb{E}\left[\left(g_i(z)-\mathbb{E}\left[g_i(z)\right]\right)^4\right] &\leq \frac{\mathbf{c_2}}{P^2},\\ \mathbb{E}\left[\left(g_i(z)-\mathbb{E}\left[g_i(z)\right]\right)^8\right] &\leq \frac{\mathbf{c_3}}{P^4},\end{aligned}$$

where $\mathbf{c_0}$, $\mathbf{c_1}$, $\mathbf{c_2}$, *and* $\mathbf{c_3}$ *depend on* γ *and* z *only.*

Proof. The random variable w_i is independent from $B_{(i)}(z) = \frac{1}{P}W_{(i)}K_{(i)}W_{(i)}^T - z\mathrm{I}_P$ since the i-th column of W does not appear in the definition of $B_{(i)}(z)$. Using Lemma H.3.3, since there exists a unique pair partition $p \in \boldsymbol{P}_2(2)$, namely $\{\{1,2\}\}$, the expectation of $g_i(z)$ is given by

$$\mathbb{E}\left[g_i(z)\right] = \frac{1}{P}\mathbb{E}\left[\operatorname{Tr}\left[B_{(i)}(z)^{-1}\right]\right].$$

Recall that $\mathbb{E}\left[m_P(z)\right] = \frac{1}{P}\mathbb{E}\left[\operatorname{Tr}\left[B(z)^{-1}\right]\right]$ and $\left|\operatorname{Tr}\left(B(z)^{-1}\right) - \operatorname{Tr}\left(B_{(i)}(z)^{-1}\right)\right| \leq \frac{1}{d(z,\mathbb{R}_+)}$ (from the proof of Lemma H.3.2). Hence

$$\left|\mathbb{E}\left[g_i(z)\right] - \mathbb{E}\left[m_P(z)\right]\right| \leq \frac{1}{P}\mathbb{E}\left[\left|\operatorname{Tr}\left(B(z)^{-1}\right) - \operatorname{Tr}\left(B_{(i)}(z)^{-1}\right)\right|\right] \leq \frac{1}{P}\frac{1}{d(z,\mathbb{R}_+)}.$$

which proves the first assertion with $\mathbf{c_0} = \frac{1}{d(z,\mathbb{R}_+)}$.

Now, let us consider the variance of $g_i(z)$. Using our previous computation of $\mathbb{E}\left[g_i(z)\right]$, we have

$$\operatorname{Var}(g_i(z)) = \mathbb{E}\left[w_i^T\frac{\left(B_{(i)}(z)\right)^{-1}}{P}w_iw_i^T\frac{\left(B_{(i)}(z)\right)^{-1}}{P}w_i\right] - \mathbb{E}\left[\frac{1}{P}\operatorname{Tr}\left[B_{(i)}(z)^{-1}\right]\right]^2.$$

The first term can be computed using the first assertion of Lemma H.3.3: there are 2 matrices involved, thus we have to sum over 3 pair partitions. A simplification arises since $\frac{\left(B_{(i)}(z)\right)^{-1}}{P}$ is symmetric: the partition $\{\{1,2\},\{3,4\}\}$ yields $\mathbb{E}\left[\left(\operatorname{Tr}\left[\frac{\left(B_{(i)}(z)\right)^{-1}}{P}\right]\right)^2\right]$ whereas both $\{\{1,3\},\{2,4\}\}$ and $\{\{1,4\},\{2,4\}\}$ yield $\mathbb{E}\left(\operatorname{Tr}\left[\frac{\left(B_{(i)}(z)\right)^{-2}}{P^2}\right]\right)$.

Thus, the variance of $g_i(z)$ is given by:

$$\operatorname{Var}(g_i(z)) = 2\mathbb{E}\left(\operatorname{Tr}\left[\frac{\left(B_{(i)}(z)\right)^{-2}}{P^2}\right]\right) + \mathbb{E}\left[\left(\frac{1}{P}\operatorname{Tr}\left[\left(B_{(i)}(z)\right)^{-1}\right]\right)^2\right] - \mathbb{E}\left[\frac{1}{P}\operatorname{Tr}\left[\left(B_{(i)}(z)\right)^{-1}\right]\right]^2$$

hence is given by a sum of two terms:

$$\operatorname{Var}(g_i(z)) = \frac{2}{P}\mathbb{E}\left(\frac{1}{P}\operatorname{Tr}\left[\left(B_{(i)}(z)\right)^{-2}\right]\right) + \operatorname{Var}\left(\frac{1}{P}\operatorname{Tr}\left[\left(B_{(i)}(z)\right)^{-1}\right]\right).$$

Using the same arguments as those explained for the bound on the Stieltjes transform, the first term is bounded by $\frac{2}{Pd(z,\mathbb{R}_+)^2}$. In order to bound the second term, we apply Lemma H.3.2 for $W_{(i)}$ and $K_{(i)}$ in place of W and K. The second term is bounded by $\frac{\mathbf{c}}{P}$, hence the bound $\operatorname{Var}\left(g_i(z)\right) \leq \frac{\mathbf{c_1}}{P}$.

Finally, we prove the bound on the fourth moment of $g_i(z) - \mathbb{E}[g_i(z)]$. We denote $m_{(i)}(z) = \frac{1}{P}\mathrm{Tr}\left[\left(B_{(i)}(z)\right)^{-1}\right]$. Recall that $\mathbb{E}[g_i(z)] = \mathbb{E}\left[m_{(i)}(z)\right]$. Using the convexity of $t \mapsto t^4$, we have

$$\begin{aligned}\mathbb{E}\left[(g_i(z) - \mathbb{E}[g_i(z)])^4\right] &= \mathbb{E}\left[\left(g_i(z) - m_{(i)}(z) + m_{(i)}(z) - \mathbb{E}\left[m_{(i)}(z)\right]\right)^4\right] \\ &\leq 8\mathbb{E}\left[\left(g_i(z) - m_{(i)}(z)\right)^4\right] + 8\mathbb{E}\left[\left(m_{(i)}(z) - \mathbb{E}\left[m_{(i)}(z)\right]\right)^4\right].\end{aligned}$$

We bound the second term using the concentration of the Stieljes transform (Lemma H.3.2): it is bounded by $\frac{8\mathbf{c}}{P^2}$. The first term is bounded using the second assertion of Lemma H.3.3. Using the symmetry of $B_{(i)}(z)$, the partitions in : $\boldsymbol{P}_2(4)$: yield two different terms, namely:

1. $\frac{1}{P^2}\mathbb{E}\left[\left(\frac{1}{P}\mathrm{Tr}\left[\left(B_{(i)}(z)\right)^{-2}\right]\right)^2\right]$, for example if $p = \{\{1,3\},\{2,4\},\{5,7\},\{6,8\}\}$
2. $\frac{1}{P^3}\mathbb{E}\left[\frac{1}{P}\mathrm{Tr}\left[\left(B_{(i)}(z)\right)^{-4}\right]\right]$, for example if $p = \{\{2,3\},\{4,5\},\{6,7\},\{8,1\}\}$.

We bound the two terms using the same arguments as those explained for the bound on the Stieljes transform at the beginning of the section. The first term is bounded by $\frac{d(z,\mathbb{R}^+)^{-4}}{P^2}$ and the second term by $\frac{d(z,\mathbb{R}^+)^{-4}}{P^3}$ hence the bound $\mathbb{E}\left[(g_i(z) - \mathbb{E}[g_i(z)])^4\right] \leq \frac{\mathbf{c_2}}{P^2}$.

The bound $\mathbb{E}[(g_i(z) - \mathbb{E}[g_i(z)])^8] \leq \frac{\mathbf{c_3}}{P^4}$ is obtained in a similar way, using the second assertion of Lemma H.3.3 and simple bounds on the Stieljes transform. □

In the next proposition we show that the Stieltjes transform $m_P(z)$ is close in expectation to the solution of a fixed point equation.

Proposition H.3.5. *For any $z \in \mathbb{H}_{<0} = \{z : \mathrm{Re}(z) < 0\}$,*

$$|\mathbb{E}[m_P(z)] - \tilde{m}(z)| \leq \frac{\mathbf{e}}{P},$$

where $\mathbf{e}$ depends on z, γ, and $\frac{1}{N}\mathrm{Tr}(K)$ only and where $\tilde{m}(z)$ is the unique solution in the cone $\mathcal{C}_z := \{u - \frac{1}{z}v : u, v \in \mathbb{R}_+\}$ spanned by 1 and $-\frac{1}{z}$ of the equation

$$\gamma = \frac{1}{N}\sum_{i=1}^{N}\frac{d_i\tilde{m}(z)}{1 + d_i\tilde{m}(z)} - \gamma z\tilde{m}(z).$$

Proof. We use the same notation as in the previous proofs, namely $B(z) = \frac{1}{P}WKW^T - z\mathrm{I}_P$, $B_{(i)}(z) = \frac{1}{P}W_{(i)}K_{(i)}W_{(i)}^T - z\mathrm{I}_P$ and $g_i(z) = \frac{1}{P}w_i^T\left(B_{(i)}(z)\right)^{-1}w_i$. Let $\nu_j \geq 0,\ j = 1,\ldots,P$ be the spectrum of the positive semi-definite matrix $\frac{1}{P}W_{(i)}K_{(i)}W_{(i)}^T$. After diagonalization, we have

$$B_{(i)}(z)^{-1} = O^T\mathrm{diag}(\frac{1}{\nu_1 - z},\ldots,\frac{1}{\nu_P - z})O,$$

with O an orthogonal matrix. Then

$$g_i(z) = \frac{1}{P}\mathrm{Tr}\left(\left(B_{(i)}(z)\right)^{-1}w_iw_i^T\right) = \frac{1}{P}\sum_{j=1}^{P}\frac{((Ow_i)_{jj})^2}{\nu_j - z}. \tag{H.3.5}$$

Since $z \in \mathbb{H}_{<0}$, we conclude that $\Re[g_i(z)] \geq 0$ for all $i = 1, \dots, P$.

In order to prove the proposition, the key remark is that, since $\operatorname{Tr}\left((\frac{1}{P}WKW^T - z\mathrm{I}_P)(B(z))^{-1}\right) = P$, the Stieltjes transform $m_P(z)$ satisfies the following equation:

$$P = \operatorname{Tr}\left(\frac{1}{P}KW^T B(z)^{-1}W\right) - zPm_P(z).$$

From the proof of Lemma H.3.2, recall that $B^{-1}(z) = B_{(i)}^{-1}(z) - \frac{d_i}{P}\frac{1}{1+\frac{d_i}{P}w_i^T B_{(i)}^{-1}(z)w_i}B_{(i)}^{-1}(z)w_i w_i^T B_{(i)}^{-1}(z)$, hence:

$$\begin{aligned}\frac{1}{P}w_i^T B^{-1}(z)w_i &= g_i(z) - \frac{d_i g_i(z)^2}{1+d_i g_i(z)} \\ &= \frac{g_i(z)}{1+d_i g_i(z)}.\end{aligned} \tag{H.3.6}$$

Expanding the trace,

$$\operatorname{Tr}\left(\frac{1}{P}KW^T B(z)^{-1}W\right) = \sum_{i=1}^N d_i \frac{1}{P}w_i^T B^{-1}(z)w_i = \sum_{i=1}^N \frac{d_i g_i(z)}{1+d_i g_i(z)}.$$

Thus, the Stieljes transform $m_P(z)$ satisfies the following equation $P = \sum_{i=1}^N \frac{d_i g_i(z)}{1+d_i g_i(z)} - zPm_P(z)$, or equivalently

$$\gamma = \frac{1}{N}\sum_{i=1}^N \frac{d_i g_i(z)}{1+d_i g_i(z)} - z\gamma m_P(z).$$

Recall that $\gamma > 0$ and $\operatorname{Re}(z) < 0$. The Stieljes transform $m_P(z)$ can be written as a function of $g_i(z)$ for $i = 1, \dots, n$: $m_P(z) = f(g_1(z), ..., g_N(z))$ where

$$f(g_1, \dots, g_N) = \frac{1}{\gamma z N}\sum_{i=1}^N \frac{d_i g_i}{1+d_i g_i} - \frac{1}{z} = -\frac{1}{z}\left(1 - \frac{1}{\gamma} + \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^N \frac{1}{1+d_i g_i}\right).$$

From Lemma H.3.6, the map $f(m) = f(m, ..., m)$ has a unique non-degenerate fixed point $\tilde{m}(z)$ in the cone $\mathcal{C}_z$. We will show that $\mathbb{E}[m_P(z)]$ is close to $\tilde{m}(z)$ using the following two steps: we show a non-tight bound $|\mathbb{E}[m_P(z)] - \tilde{m}(z)| \leq \frac{\mathbf{e}'}{\sqrt{P}}$ and use it to obtain the tighter bound $|\mathbb{E}[m_P(z)] - \tilde{m}(z)| \leq \frac{\mathbf{e}}{P}$.

Let us prove the $\frac{\mathbf{e}'}{\sqrt{P}}$ bound. From Lemma H.3.6, the distance between $m_P(z)$ and the fixed point $\tilde{m}(z)$ of f is bounded by the distance between $f(m_P(z), \dots, m_P(z))$ and $m_P(z)$. Using the fact that $m_P(z) = f(g_1(z), ..., g_N(z))$, we obtain

$$|\mathbb{E}[m_P(z)] - \tilde{m}(z)| \leq \mathbb{E}\left[|m_P(z) - \tilde{m}(z)|\right] \leq \mathbb{E}\left[|f(m_P(z), \dots, m_P(z)) - f(g_1(z), ..., g_N(z))|\right].$$

Recall that for any $z \in \mathbb{H}_{<0}$, $\Re(g_i(z)) \geq 0$: we need to study the function f on $\mathbb{H}_{\geq 0}^N$ where $\mathbb{H}_{\geq 0} = \{z \in \mathbb{C} | \Re(z) \geq 0\}$. On $\mathbb{H}_{\geq 0}^N$, the function f is Lipschitz:

$$|\partial_{g_i} f(g_1, .., g_N)| = \left|\frac{1}{\gamma z N}\frac{d_i}{(1+d_i g_i)^2}\right| \leq \frac{d_i}{\gamma |z| N}.$$

Thus,

$$\mathbb{E}\left[\left|f\left(m_P(z),...,m_P(z)\right)-f\left(g_1(z),...,g_N(z)\right)\right|\right]\leq\sum_{i=1}^{N}\frac{d_i}{\gamma\left|z\right|N}\mathbb{E}\left[\left|m_P(z)-g_i(z)\right|\right].$$

Since

$$\mathbb{E}\left[\left|m_P(z)-g_i(z)\right|\right]\leq\mathbb{E}\left[\left|m_P(z)-\mathbb{E}\left[m_P(z)\right]\right|\right]+\left|\mathbb{E}\left[m_P(z)\right]-\mathbb{E}\left[g_i(z)\right]\right|+\mathbb{E}\left[\left|g_i(z)-\mathbb{E}\left[g_i(z)\right]\right|\right],$$

using Lemmas H.3.2 and H.3.4, we get that $\mathbb{E}\left[\left|m_P(z)-g_i(z)\right|\right]\leq\frac{\mathbf{d}}{\sqrt{P}}$, where $\mathbf{d}$ depends on γ and z only. This implies that

$$\mathbb{E}\left[\left|f\left(m_P(z),...,m_P(z)\right)-f\left(g_1(z),...,g_N(z)\right)\right|\right]\leq\frac{1}{\sqrt{P}}\frac{\mathbf{d}}{N}\mathrm{Tr}\left(K\right),$$

which allows to conclude that $\left|\mathbb{E}[m_P(z)]-\tilde{m}(z)\right|\leq\frac{\mathbf{e}'}{\sqrt{P}}$ where $\mathbf{e}'$ depends on γ, z and $\frac{1}{N}\mathrm{Tr}(K)$ only.

We strengthen this inequality and show the $\frac{\mathbf{e}}{P}$ bound. Using again Lemma H.3.6, we bound the distance between $\mathbb{E}[m_P(z)]$ and the fixed point $\tilde{m}(z)$ by

$$\left|\mathbb{E}[m_P(z)]-\tilde{m}(z)\right|\leq\left|\mathbb{E}[f(g_1(z),\ldots,g_N(z))]-f(\mathbb{E}[m_P(z)],\ldots,\mathbb{E}[m_P(z)])\right|$$

and study the r.h.s. using a Taylor approximation of f near $\mathbb{E}\left[m_P(z)\right]$. For $i=1,\ldots,N$ and $m_0\in\mathbb{H}_{\geq 0}$, let $\mathrm{T}_{m_0}h_i$ be the first order Taylor approximation of the map $h_i:m\mapsto\frac{1}{1+d_im}$ at a point m_0. The error of the first order Taylor approximation is given by

$$h_i(m)-\mathrm{T}_{m_0}h_i(m)=\frac{1}{1+d_im}-\left(\frac{1}{1+d_im_0}-\frac{d_i(m-m_0)}{\left(1+d_im_0\right)^2}\right)=\frac{d_i^2\left(m_0-m\right)^2}{\left(1+d_im\right)\left(1+d_im_0\right)^2},$$

which, for $m\in\mathbb{H}_{\geq 0}$ can be upper bounded by a quadratic term:

$$\left|h_i(m)-\mathrm{T}_{m_0}h_i(m)\right|=\left|\frac{d_i^2}{\left(1+d_im\right)\left(1+d_im_0\right)^2}\right|\left|m_0-m\right|^2\leq\frac{1}{\left|m_0\right|^2}\left|m_0-m\right|^2.\tag{H.3.7}$$

The first order Taylor approximation $\mathrm{T}f$ of f at the N-tuple $\left(\mathbb{E}\left[m_P(z)\right],...,\mathbb{E}\left[m_P(z)\right]\right)$ is

$$\mathrm{T}f(g_1,..,g_N)=-\frac{1}{z}\left(1-\frac{1}{\gamma}+\frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^{N}\mathrm{T}_{\mathbb{E}[m_P(z)]}h_i(g_i)\right).$$

Using this Taylor approximation, $\mathbb{E}[f(g_1(z),\ldots,g_N(z))]-f(\mathbb{E}[m_P(z)],\ldots,\mathbb{E}[m_P(z)])$ is equal to:

$$\mathbb{E}\left[\mathrm{T}f(g_1(z),..,g_N(z))\right]-f(\mathbb{E}[m_P(z)],\ldots,\mathbb{E}[m_P(z)])+\mathbb{E}\left[f(g_1(z),...,g_N(z))-\mathrm{T}f(g_1(z),..,g_N(z))\right].$$

Using Lemma H.3.4, we get

$$\left|\mathbb{E}\left[f(g_1(z),...,g_N(z))-\mathrm{T}f(g_1(z),..,g_N(z))\right]\right|\leq\frac{1}{\left|z\right|\gamma}\frac{1}{N}\sum_{i=1}^{N}\frac{1}{\left|\mathbb{E}[m_P(z)]\right|^2}\mathbb{E}\left[\left|g_i(z)-\mathbb{E}\left[m_P(z)\right]\right|^2\right]$$

$$\leq \frac{1}{P}\frac{\alpha}{|\mathbb{E}[m_P(z)]|^2}$$

and

$$|\mathbb{E}\left[\mathrm{T}f(g_1(z),..,g_N(z))\right] - f(\mathbb{E}\left[m_P(z)\right],...,\mathbb{E}\left[m_P(z)\right])| \leq \frac{1}{|z|}\frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^{N}\frac{d_i\left|\mathbb{E}\left[g_i\right]-\mathbb{E}\left[m_P(z)\right]\right|}{\left|1+d_i\mathbb{E}\left[m_P(z)\right]\right|^2}$$
$$\leq \frac{\beta\left(\frac{1}{N}\mathrm{Tr}K\right)}{P}$$

where α and β depends on z and γ only. From the bounds $|\mathbb{E}[m_P(z)] - \tilde{m}(z)| \leq \frac{\mathbf{e}'}{\sqrt{P}}$ and $|\tilde{m}(z)| \geq (|z| + \frac{1}{N\gamma}\mathrm{Tr}(K))^{-1}$ (Lemma H.3.6), the bound $\frac{1}{P}\frac{\alpha}{|\mathbb{E}[m_P(z)]|^2}$ yields a $\frac{\tilde{\alpha}}{P}$ bound. This implies that $|\mathbb{E}[m_P(z)] - f(\mathbb{E}[m_P(z)],\ldots,\mathbb{E}[m_P(z)])| \leq \frac{\mathbf{e}}{P}$, hence the desired inequality $|\mathbb{E}\left[m_P(z)\right] - \tilde{m}(z)| \leq \frac{\mathbf{e}}{P}$. □

For the proof of Proposition H.3.5, we have used the fact that the map f_z introduced therein has a unique non-degenerate fixed point in the cone $\mathcal{C}_z := \{u - \frac{1}{z}v : u, v \in \mathbb{R}_+\}$. We now proceed with proving this statement.

Lemma H.3.6. *Let $d_1, \ldots, d_n \geq 0$ and let $\gamma \geq 0$. For any fixed $z \in \mathbb{H}_{<0}$, let $f_z : \mathbb{H}_{\geq 0} \to \mathbb{C}$ be the function $t \mapsto f_z(t) = -\frac{1}{z}\left(1 - \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^{N}\frac{d_i t}{1+d_i t}\right)$. Let $\mathcal{C}_z := \{u - \frac{1}{z}v : u, v \in \mathbb{R}_+\}$ be the convex region spanned by the half-lines $\mathbb{R}_+$ and $-\frac{1}{z}\mathbb{R}_+$. Then for every $z \in \mathbb{H}_{<0}$ there exists a unique fixed point $\tilde{t}(z) \in \mathcal{C}_z$ such that $\tilde{t}(z) = f_z(\tilde{t}(z))$. The map $\tilde{t} : z \mapsto \tilde{t}(z)$ is holomorphic in $\mathbb{H}_{<0}$ and*

$$|\tilde{t}(z)| \geq \left(|z| + \frac{\sum_i d_i}{\gamma N}\right)^{-1}.$$

Furthermore for every $z \in \mathbb{H}_{<0}$ and any $t \in \mathbb{H}_{\geq 0}$, one has

$$|t - \tilde{t}(z)| \leq |t - f_z(t)|.$$

Proof. By means of Schwarz reflection principle, we can assume that $\Im(z) \geq 0$. Let $z \in \mathbb{H}_{<0}$ and let $\Pi_z := \{-\frac{w}{z} : \Im(w) \leq 0\}$ and let $\mathcal{C}_z$ be the wedged region $\mathcal{C}_z := \Pi_z \cap \{w \in \mathbb{C} : \Im(w) \geq 0\}$. To show the existence of a fixed point in $\mathcal{C}_z$ we show that 0 is in the image of the function $\psi : t \mapsto f_z(t) - t$. Note that since $d_i \geq 0$, the eventual poles of f_z are all strictly negative real numbers, hence $\psi : \mathcal{C}_z \to \mathbb{C}$ is an holomorphic function.

To prove that $0 \in \psi(\mathcal{C}_z)$ we proceed with a geometrical reasoning: the image $\psi(\mathcal{C}_z)$ is (one of) the region of the plane confined by $\psi\left(\partial\mathcal{C}_z\right)$, so we only need to "draw" $\psi\left(\partial\mathcal{C}_z\right)$ and show that 0 belongs to the "good" connected component confined by it.

The boundary of C_z is made up of two half-lines $\mathbb{R}_+$ and $-\frac{1}{z}\mathbb{R}_+$. Under the map f_z, 0 is mapped to $-\frac{1}{z}$ and ∞ is mapped to $-\frac{1-\frac{1}{\gamma}}{z}$, the two half-lines are hence mapped to paths from $-\frac{1}{z}$ to $-\frac{1-\frac{1}{\gamma}}{z}$. Now under ψ the half-lines will be mapped to paths going $-\frac{1}{z}$ to ∞ because by our assumption $-\frac{1}{z}$ lies in the upper right quadrant, we will show that the image of $\mathbb{R}_+$ under ϕ goes 'above' the origin while the image of $-\frac{1}{z}\mathbb{R}_+$ goes 'under' the origin:

- $\mathbb{R}_+$ is mapped under f_z to the segment $-\frac{1}{z}[1, 1 - \frac{1}{\gamma}]$, as a result, its map under ψ lies in the Minkowski sum $-\frac{1}{z}[1, 1 - \frac{1}{\gamma}] + (-\mathbb{R}_+)$ which is contained in $\overline{\mathbb{C} \setminus \Pi_z}$.

- For any $t \in -\frac{1}{z}\mathbb{R}_+$ we have for all d_i

$$\Im\left(\frac{d_i t}{1+d_i t}\right) = \Im\left(1 - \frac{1}{1+d_i t}\right) = \Im\left(\frac{1}{1+d_i t}\right) \leq 0,$$

since $\Im(t) \geq 0$. As a result the image of $-\frac{1}{z}\mathbb{R}_+$ under f_z lies in Π_z and its image under ψ lies in the Minkovski sum $\Pi_z + (-\frac{1}{z}\mathbb{R}_+) = \Pi_z$.

Thus we can conclude that $0 \in \psi(\mathcal{C}_z)$, which shows that there exists at least a fixed point $\tilde{m}$ in $\mathcal{C}_z$.

We observe that, for every $t \in \mathcal{C}_z$, the derivative of f has negative real part:

$$\begin{aligned}\operatorname{Re}(f_z'(t)) &= \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^{N}\operatorname{Re}\left(\frac{d_i}{z\,(1+d_i t)^2}\right)\\ &= \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^{N}\frac{d_i\left[\Re(z)+2d_i\Re(z)\Re(t)-2d_i\Im(z)\Im(t)+d_i^2\Re(zt^2)\right]}{|z|^2\,|1+d_i t|^4} \leq 0,\end{aligned}$$

where we concluded the last inequality by using that $\Re(z) \leq 0$, $\Re(t) \geq 0$, $\Im(z)\Im(t) \geq 0$ and $\Re(zt^2) \leq 0$. Thus, since for no point $t \in \mathcal{C}_z$ has $f_z'(t) = 1$, any fixed point of f_z is a simple fixed point.

We now proceed to show the uniqueness of the fixed point in the region $\mathcal{C}_z$. Suppose there are two fixed points t_1 and t_2, then

$$\begin{aligned}t_1 - t_2 &= f_z(t_1) - f_z(t_2)\\ &= (t_1 - t_2)\,\frac{1}{z}\frac{1}{\gamma N}\sum_{i=1}^{N}\frac{d_i}{(1+d_i t_1)(1+d_i t_2)}.\end{aligned}$$

Again, since $\Re(z) \leq 0$, $\Re(t_1), \Re(t_2) \geq 0$, $\Im(z)\Im(t_1), \Im(z)\Im(t_2), \geq 0$ and $\Re(z t_1 t_2) \leq 0$, the factor $\frac{1}{z}\frac{1}{N}\sum_{i=1}^{N}\frac{d_i}{(1+d_i t_1)(1+d_i t_2)}$ has negative real part, and thus the identity is possible only if $t_1 = t_2$. Let's then $\tilde{t}(z)$ be the only fixed point in $\mathcal{C}_z$.

We proceed now to show that $|t - f_z(t)| \geq |t - \tilde{t}(z)|$, i.e. if t and its image are close, then t is not too far from being a fixed point, and so it is close to $\tilde{t}(z)$.

For any $t \in \mathcal{C}_z$, we have

$$\begin{aligned}|t - f_z(t)| &= |t - \tilde{t}(z) + f_z(\tilde{t}(z)) - \tilde{f}_z(t)|\\ &= \left|(t-\tilde{t}(z)) - \big(t-\tilde{t}(z)\big)\left(\frac{1}{z}\frac{1}{\gamma N}\sum_{i=1}^{N}\frac{d_i}{(1+d_i t)(1+d_i\tilde{t}(z))}\right)\right|\\ &= \big|t-\tilde{t}(z)\big|\left|1 - \frac{1}{z}\frac{1}{\gamma N}\sum_{i=1}^{N}\frac{d_i}{(1+d_i t)(1+d_i\tilde{t}(z))}\right|\\ &\geq \big|t-\tilde{t}(z)\big|\end{aligned}$$

where we have used again that $\frac{1}{z}\frac{1}{N}\sum_{i=1}^{N}\frac{d_i}{(1+d_i t)(1+d_i\tilde{t}(z))}$ has negative real part.

We provide a lower bound on the norm of the fixed point:

$$\big|\tilde{t}(z)\big| = \frac{1}{|z|}\left|1 - \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^{N}\frac{d_i\tilde{t}(z)}{1+d_i\tilde{t}(z)}\right| \geq \frac{1}{|z|}\left(1 - \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^{N}\left|\frac{d_i\tilde{t}(z)}{1+d_i\tilde{t}(z)}\right|\right) \geq \frac{1}{|z|}\left(1 - \frac{|\tilde{t}(z)|}{\gamma N}\sum_{i=1}^{N}d_i\right).$$

hence

$$|\tilde{t}(z)| \geq \left(|z| + \frac{\sum_i d_i}{\gamma N}\right)^{-1}.$$

Finally, note that z can be expressed from the fixed point $\tilde{m}$, hence defining an inverse for the map $\tilde{t}$:

$$\tilde{t}^{-1}(\tilde{m}) = z = -\frac{1}{\tilde{m}}\left(1 - \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^{N}\frac{d_i\tilde{m}}{1+d_i\tilde{m}}\right)$$

because the inverse is holomorphic, so is $\tilde{t}$. □

Ridge

Using Proposition H.3.1, in order to have a better description of the distribution of the predictor $\hat{f}^{(RF)}_{\lambda,\gamma}$, it remains to study the distributions of both the final labels $\hat{y}$ on the training set and the parameter norm $\|\hat{\theta}\|^2$. In Section H.3, we first study the expectation of the final labels $\hat{y}$: this allows us to study the loss of the average predictor $\mathbb{E}\left[\hat{f}^{(RF)}_{\lambda,\gamma}\right]$. Then in Section H.3, a study of the variance of the predictor allows us to study the average loss of the RF predictor.

Expectation of the predictor

The optimal parameters $\hat{\theta}$ which minimize the regularized MSE loss is given by $\hat{\theta} = F^T(FF^T + \lambda \mathrm{I}_N)^{-1}y$, or equivalently by $\hat{\theta} = (F^TF+\lambda)^{-1}F^Ty$. Thus, the final labels take the form $\hat{y} = A(-\lambda)y$ where $A(z)$ is the random matrix defined as

$$\begin{aligned} A(z) &:= F\left(F^TF - z\mathrm{I}_P\right)^{-1}F^T \\ &= \frac{1}{P}K^{\frac{1}{2}}W^T\left(\frac{1}{P}WKW^T - z\mathrm{I}_P\right)^{-1}WK^{\frac{1}{2}}. \end{aligned}$$

Note that the matrix A_λ defined in the proof sketch of Theorem 4.1 in the main text is given by $A_\lambda = A(-\lambda)$.

Proposition H.3.7. *For any $\gamma > 0$, any $z \in \mathbb{H}_{<0}$, and any symmetric positive definite matrix K,*

$$\|\mathbb{E}\left[A(z)\right] - K(K + \tilde{\lambda}(-z)I_N)^{-1}\|_{op} \leq \frac{c}{P}, \tag{H.3.8}$$

where $\tilde{\lambda}(z) := \frac{1}{\tilde{m}(-z)}$ and $c > 0$ depends on z, γ and $\frac{1}{N}Tr(K)$ only.

Proof. Since the distribution of W is invariant under orthogonal transformations, by applying a change of basis, in order to prove Inequality (H.3.8), we may assume that K is diagonal with diagonal entries $d_1, \ldots, d_N$. Denoting $w_1, \ldots, w_N$ the columns of W, for any $i, j = 1, \ldots, N$,

$$(A(z))_{ij} = \frac{1}{P}\sqrt{d_id_j}w_i^T\left(\frac{1}{P}WKW^T - zI_P\right)^{-1}w_j,$$

where $WKW^T = \sum_{i=1}^N d_i w_i w_i^T$. Replacing w_i by $-w_i$ does not change the law W hence does not change the law of $(A(z))_{ij}$. Since WKW^T is invariant under this change of sign, we get that for $i \neq j$, $\mathbb{E}\left[(A(z))_{ij}\right] = -\mathbb{E}\left[(A(z))_{ij}\right]$, hence the off-diagonal terms of $\mathbb{E}\left[A(z)\right]$ vanish.

Consider a diagonal term $(A(z))_{ii}$. From Equation (H.3.6), we get

$$(A(z))_{ii} = \frac{d_i}{P} w_i^T B^{-1}(z) w_i = \frac{d_i g_i(z)}{1 + d_i g_i(z)}. \tag{H.3.9}$$

By Lemma H.3.4, g_i lies close to $m_P(z)$ which itself is approximatively equal to $\tilde{m}(z)$ by Proposition H.3.5. Therefore, we expect $\mathbb{E}\left[(A(z))_{ii}\right] = \mathbb{E}\left[\frac{d_i g_i}{1+d_i g_i}\right]$ to be at short distance from $\frac{d_i \tilde{m}(z)}{1+d_i \tilde{m}(z)}$.

In order to make rigorous this heuristic and to prove that $\mathbb{E}\left[(A(z))_{ii}\right]$ is within $\mathcal{O}(\frac{1}{P})$ distance to $\frac{d_i \tilde{m}(z)}{1+d_i \tilde{m}(z)}$, we consider the first order Taylor approximation $\mathrm{T}_{\tilde{m}(z)} h_i$ of the map $h_i : g \mapsto \frac{1}{1+d_i g}$ (as in the proof Proposition H.3.5 but this time centered at $\tilde{m}(z)$). Using the fact that $\frac{d_i t}{1+d_i t} = 1 - \frac{1}{1+d_i t} = 1 - h_i(t)$, and inserting the Taylor approximation, $\mathbb{E}\left[(A(z))_{ii}\right] - \frac{d_i \tilde{m}(z)}{1+d_i \tilde{m}(z)}$ is equal to:

$$h_i(\tilde{m}(z)) - h_i(g_i(z)) = \frac{1}{1 + d_i \tilde{m}(z)} - \mathbb{E}\left[\mathrm{T}_{\tilde{m}(z)} h(g_i(z))\right] + \mathbb{E}\left[\mathrm{T}_{\tilde{m}(z)} h(g_i(z)) - h(g_i(z))\right].$$

Thus,

$$\left|\mathbb{E}\left[(A(z))_{ii}\right] - \frac{d_i \tilde{m}(z)}{1 + d_i \tilde{m}(z)}\right| \leq \left|\frac{1}{1 + d_i \tilde{m}(z)} - \mathbb{E}\left[\mathrm{T}_{\tilde{m}(z)} h(g_i(z))\right]\right| + \left|\mathbb{E}\left[\mathrm{T}_{\tilde{m}(z)} h(g_i(z)) - h(g_i(z))\right]\right|.$$

Using Lemma H.3.4 and Proposition H.3.5, the first term $\left|\frac{1}{1+d_i \tilde{m}(z)} - \mathbb{E}\left[\mathrm{T}_{\tilde{m}(z)} h(g_i(z))\right]\right| = \frac{d_i |\mathbb{E}[g_i(z)] - \tilde{m}(z)|}{|1+d_i \tilde{m}(z)|^2}$ can be bounded by $\frac{\delta}{P} \frac{d_i}{|1+d_i \tilde{m}(z)|^2}$ where δ depends on z, γ and $\frac{1}{N}\mathrm{Tr}(K)$ only. Since $\mathrm{Re}\left[\tilde{m}(z)\right] \geq 0$ thus $|1 + d_i \tilde{m}(z)| \geq \max(1, |d_i \tilde{m}(z)|)$, and $|\tilde{m}(z)| \geq \frac{1}{|z| + \frac{1}{\gamma}\frac{1}{N}\mathrm{Tr}K}$ (Lemma H.3.6), the denominator can be lower bounded:

$$|1 + d_i \tilde{m}(z)|^2 \geq |d_i \tilde{m}(z)| \geq \frac{d_i}{|z| + \frac{1}{\gamma}\frac{1}{N}\mathrm{Tr}K},$$

yielding the upper bound:

$$\left|\frac{1}{1 + d_i \tilde{m}(z)} - \mathbb{E}\left[\mathrm{T}_{\tilde{m}(z)} h(g_i(z))\right]\right| \leq \frac{1}{P}\delta\left[|z| + \frac{1}{\gamma}\frac{1}{N}\mathrm{Tr}K\right].$$

For the second term, using the same arguments as for the proof of Proposition H.3.5, we have:

$$\left|\mathbb{E}\left[\mathrm{T}_{\tilde{m}(z)} h(g_i(z)) - h(g_i(z))\right]\right| \leq \frac{\mathbb{E}\left[\left|\tilde{m}(z) - g_i(z)\right|^2\right]}{\left|\tilde{m}(z)\right|^2}.$$

Recall that $|\tilde{m}(z)| \geq \frac{1}{|z| + \frac{1}{\gamma}\frac{1}{N}\mathrm{Tr}K}$ and that, by Lemma H.3.4 and Proposition H.3.2, $\mathbb{E}\left[\left|\tilde{m}(z) - g_i(z)\right|^2\right] \leq \frac{\tilde{\delta}}{P}$ where $\tilde{\delta}$ depends on z, γ and $\frac{1}{N}\mathrm{Tr}(K)$ only. This implies that

$$\left|\mathbb{E}\left[\mathrm{T}_{\tilde{m}(z)} h(g_i(z)) - h(g_i(z))\right]\right| \leq \frac{\tilde{\delta}}{P}\left[|z| + \frac{1}{\gamma}\frac{1}{N}\mathrm{Tr}K\right]^2.$$

As a consequence, there exists a constant c which depends on z, γ and $\frac{1}{N}\mathrm{Tr}(K)$ only such that:

$$\left| \mathbb{E}\left[(A(z))_{ii}\right] - \frac{d_i \tilde{m}(z)}{1 + d_i \tilde{m}(z)} \right| \leq \frac{c}{P}.$$

Using the effective ridge $\tilde{\lambda}(z) := \frac{1}{\tilde{m}(-z)}$, the term $\frac{d_i \tilde{m}(z)}{1+d_i \tilde{m}(z)} = \frac{d_i}{d_i + \tilde{\lambda}(-z)}$ is equal to $(K(K+\tilde{\lambda} I_N)^{-1})_{ii}$ since, in the basis considered, $K(K + \tilde{\lambda} I_N)^{-1}$ is a diagonal matrix. Hence, we obtain:

$$\left\| \mathbb{E}\left[A(z)\right] - K(K + \tilde{\lambda} I_N)^{-1} \right\|_{op} \leq \frac{c}{P}$$

which allows us to conclude. □

Using the above proposition, we can bound the distance between the expected λ-RF predictor and the $\tilde{\lambda}$-RF predictor.

Theorem H.3.8. *For $N, P > 0$ and $\lambda > 0$, we have*

$$\left| \mathbb{E}[\hat{f}^{(RF)}_{\lambda,\gamma}(x)] - \hat{f}^{(K)}_{\tilde{\lambda}}(x) \right| \leq \frac{c\sqrt{K(x,x)} \|y\|_{K^{-1}}}{P} \tag{H.3.10}$$

where the effective ridge $\tilde{\lambda}(\lambda, \gamma) > \lambda$ is the unique positive number satisfying

$$\tilde{\lambda} = \lambda + \frac{\tilde{\lambda}}{\gamma} \frac{1}{N} \sum_{i=1}^{N} \frac{d_i}{\tilde{\lambda} + d_i}, \tag{H.3.11}$$

and where $c > 0$ depends on λ, γ, and $\frac{1}{N}\mathrm{Tr}K(X,X)$ only.

Proof. Recall that $\tilde{m}(-\lambda)$ is the unique non negative real such that $\gamma = \frac{1}{N}\sum_{i=1}^{N} \frac{d_i \tilde{m}(-\lambda)}{1+d_i \tilde{m}(-\lambda)} + \gamma\lambda\tilde{m}(-\lambda)$. Dividing this equality by $\gamma\tilde{m}(-\lambda)$ yields Equation (H.3.11). From now on, let $\tilde{\lambda} = \tilde{\lambda}(\lambda, \gamma)$.

We now bound the l.h.s. of Equation (H.3.10). By Proposition H.3.1, since $\hat{y} = A(-\lambda)y$, the average λ-RF predictor is $\mathbb{E}\left[f^{(RF)}_{\lambda,\gamma}(x)\right] = K(x,X)K^{-1}\mathbb{E}\left[A(-\lambda)\right] y$. The $\tilde{\lambda}$-KRR predictor is $f^{(K)}_{\tilde{\lambda}}(x) = K(x,X)\left(K + \tilde{\lambda} I_N\right)^{-1} y$. Thus:

$$\left| \mathbb{E}[f^{(RF)}_{\lambda,\gamma}(x)] - f^{(K)}_{\tilde{\lambda}}(x) \right| = \left| K(x,X)K^{-1}\left[\mathbb{E}\left[A(-\lambda)\right] - K\left(K + \tilde{\lambda} I_N\right)^{-1}\right] y \right|.$$

The r.h.s. can be expressed as the absolute value of the scalar product $|\langle w, v\rangle_{K^{-1}}| = \left|v^T K^{-1} w\right|$ where $v = K(x,X)$ and $w = [\mathbb{E}\left[A(-\lambda)\right] - K(K + \tilde{\lambda} I_N)^{-1}]y$. By Cauchy-Schwarz inequality, $|\langle v, w\rangle_{K^{-1}}| \leq \|v\|_{K^{-1}} \|w\|_{K^{-1}}$.

For a general vector v, the K^{-1}-norm $\|v\|_{K^{-1}}$ is equal to the norm mininum Hilbert norm (for the RKHS associated to the kernel K) interpolating function:

$$\|v\|_{K^{-1}} = \min_{f \in \mathcal{H}, f(x_i) = v_i} \|f\|_{\mathcal{H}}.$$

Indeed the minimal interpolating function is the kernel regression given by $f^{(K)}(\cdot) = K(\cdot, X)K(X, X)^{-1}v$ which has norm (writing $\beta = K^{-1}v$):

$$\left\|f^{(K)}\right\|_{\mathcal{H}} = \left\|\sum_{i=1}^{N} \beta_i K(\cdot, x_i)\right\|_{\mathcal{H}} = \sqrt{\sum_{i,j=1}^{N} \beta_i \beta_j K(x_i, x_j)} = \sqrt{v^T K^{-1} K K^{-1} v} = \|v\|_{K^{-1}}.$$

We can now bound the two norms $\|v\|_{K^{-1}}$ and $\|w\|_{K^{-1}}$. For $v = K(x, X)$, we have

$$\|v\|_{K^{-1}} = \min_{f \in \mathcal{H}, f(x_i) = v_i} \|f\|_{\mathcal{H}} \leq \|K(x, \cdot)\|_{\mathcal{H}} = K(x, x)^{\frac{1}{2}}. \tag{H.3.12}$$

since $K(x, \cdot)$ is an interpolating function for v.

It remains to bound $\|w\|_{K^{-1}}$. Recall that $K = UDU^T$ with D diagonal, and that, from the previous proposition, $\mathbb{E}[A(-\lambda)] = UD_AU^T$ where $D_A = \operatorname{diag}\left(\frac{d_1 g_1(-\lambda)}{1+d_1 g_1(-\lambda)}, \ldots, \frac{d_N g_N(-\lambda)}{1+d_N g_N(-\lambda)}\right)$. The norm $\|w\|_{K^{-1}}$ is equal to

$$\sqrt{\tilde{y}^T \left[D_A - D\left(D + \tilde{\lambda}(\lambda) I_N\right)^{-1}\right]^T D^{-1} \left[D_A - D\left(D + \tilde{\lambda}(\lambda) I_N\right)^{-1}\right] \tilde{y}},$$

where $\tilde{y} = U^T y$. Expanding the product, $\|w\|_{K^{-1}} = \sqrt{\sum_{i=1}^{N} \frac{\tilde{y}_i^2}{d_i} \left((D_A)_{ii} - \frac{d_i}{\tilde{\lambda}(\lambda)+d_i}\right)^2}$, hence by Proposition H.3.7, $\|w\|_{K^{-1}} \leq \frac{c}{P}\sqrt{\sum_{i=1}^{N} \frac{\tilde{y}^2}{d_i}}$. The result follows from noticing that $\sum_{i=1}^{N} \frac{\tilde{y}^2}{d_i} = \tilde{y}^T D^{-1} \tilde{y} = \|y\|_{K^{-1}}^2$:

$$\left|\mathbb{E}[f_{\lambda,\gamma}^{(RF)}(x)] - f_{\tilde{\lambda}}^{(K)}(x)\right| \leq \|v\|_{K^{-1}} \|w\|_{K^{-1}} \leq \frac{cK(x, x)^{\frac{1}{2}} \|y\|_{K^{-1}}}{P}.$$

which allows us to conclude. □

Corollary H.3.9. *If $\mathbb{E}_{\mathcal{D}}[K(x, x)] < \infty$, we have that the difference of errors $\delta_E = \left|L(\mathbb{E}[\hat{f}_{\lambda,\gamma}^{(RF)}]) - L(\hat{f}_{\tilde{\lambda}}^{(K)})\right|$ is bounded from above by*

$$\delta_E \leq \frac{C\|y\|_{K^{-1}}}{P}\left(2\sqrt{L\left(\hat{f}_{\tilde{\lambda}}^{(K)}\right)} + \frac{C\|y\|_{K^{-1}}}{P}\right),$$

where C is given by $c\sqrt{\mathbb{E}_{\mathcal{D}}[K(x, x)]}$, with c the constant appearing in (H.3.10) above.

Proof. For any function $f : \mathbb{R}^d \to \mathbb{R}$, we denote by $\|f\| = (\mathbb{E}_{\mathcal{D}}\left[f(x)^2\right])^{\frac{1}{2}}$ its $L^2(\mathcal{D})$-norm. Integrating $\left|\mathbb{E}[f_{\lambda,\gamma}^{(RF)}(x)] - f_{\tilde{\lambda}}^{(K)}(x)\right|^2 \leq \frac{c^2 K(x,x)\|y\|_{K^{-1}}^2}{P^2}$ over $x \sim \mathcal{D}$, we get the following bound:

$$\|\mathbb{E}[f_{\lambda,\gamma}^{(RF)}] - f_{\tilde{\lambda}}^{(K)}\| \leq \frac{c\left[\mathbb{E}_{\mathcal{D}}\left[K(x, x)\right]\right]^{\frac{1}{2}} \|y\|_{K^{-1}}}{P}.$$

Hence, if f^* is the true function, by the triangular inequality,

$$\left|\|\mathbb{E}[f_{\lambda,\gamma}^{(RF)}] - f^*\| - \|f_{\tilde{\lambda}}^{(K)} - f^*\|\right| \leq \frac{c\left[\mathbb{E}_{\mathcal{D}}\left[K(x, x)\right]\right]^{\frac{1}{2}} \|y\|_{K^{-1}}}{P}.$$

Notice that $L(\mathbb{E}[\hat{f}^{(RF)}_{\gamma,\lambda}]) = \|\mathbb{E}[f^{(RF)}_{\lambda,\gamma}] - f^*\|^2$ and $L(\hat{f}^{(K)}_{\tilde{\lambda}}) = \|f^{(K)}_{\tilde{\lambda}} - f^*\|^2$. Since $\left|a^2 - b^2\right| \leq |a-b|\left(|a-b| + 2\,|b|\right)$, we obtain

$$\left|L\left(\mathbb{E}[\hat{f}^{(RF)}_{\gamma,\lambda}]\right) - L\left(\hat{f}^{(K)}_{\tilde{\lambda}}\right)\right| \leq \frac{c\left[\mathbb{E}_{\mathcal{D}}\left[K(x,x)\right]\right]^{\frac{1}{2}}\|y\|_{K^{-1}}}{P}\left(2\sqrt{L\left(\hat{f}^{(K)}_{\tilde{\lambda}}\right)} + \frac{c\left[\mathbb{E}_{\mathcal{D}}\left[K(x,x)\right]\right]^{\frac{1}{2}}\|y\|_{K^{-1}}}{P}\right),$$

which allows us to conclude. □

Properties of the effective ridge

Thanks to the implicit definition of the effective ridge $\tilde{\lambda}$, we obtain the following:

Proposition H.3.10. *The effective ridge $\tilde{\lambda}$ satisfies the following properties:*

1. *for any $\gamma > 0$, we have $\lambda < \tilde{\lambda}(\lambda,\gamma) \leq \lambda + \frac{1}{\gamma}T$;*
2. *the function $\gamma \mapsto \tilde{\lambda}(\lambda,\gamma)$ is decreasing;*
3. *for $\gamma > 1$, we have $\tilde{\lambda} \leq \frac{\gamma}{\gamma-1}\lambda$;*
4. *for $\gamma < 1$, we have $\tilde{\lambda} \geq \frac{1-\sqrt{\gamma}}{\sqrt{\gamma}}\min_i d_i$.*

Proof. **(1)** The upper bound in the first statement follows directly from Lemma H.3.6 where it was shown that $\tilde{m}(-\lambda) \geq \frac{1}{\lambda + \frac{1}{\gamma}\frac{1}{N}\mathrm{Tr}K}$ and from the fact that $\tilde{\lambda}(\lambda,\gamma) = \frac{1}{\tilde{m}(-\lambda)}$. For the lower bound, remark that Equation (H.3.11) can be written as:

$$\tilde{\lambda}(\lambda,\gamma) = \lambda + \frac{1}{\gamma}\frac{1}{N}\mathrm{Tr}[\tilde{\lambda}(\lambda,\gamma)K(\tilde{\lambda}(\lambda,\gamma)I_N + K)^{-1}].$$

Since $\tilde{\lambda}(\lambda,\gamma) \geq 0$ and K is a positive symmetric matrix, $\mathrm{Tr}[K[\tilde{\lambda}(\lambda,\gamma)I_N + K]^{-1}] \geq 0$: this yields $\tilde{\lambda}(\lambda,\gamma) \geq \lambda$.

(2) We show that $\gamma \mapsto \tilde{\lambda}(\lambda,\gamma)$ is decreasing by computing the derivative of the effective ridge with respect to γ. Differentiating both sides of Equation (H.3.11), $\partial_\gamma\tilde{\lambda} = \partial_\gamma\left[\lambda + \frac{\tilde{\lambda}}{\gamma}\frac{1}{N}\sum_{i=1}^N \frac{d_i}{\tilde{\lambda}+d_i}\right]$. The r.h.s. is equal to:

$$\frac{\partial_\gamma\tilde{\lambda}}{\gamma}\frac{1}{N}\sum_{i=1}^N\frac{d_i}{\tilde{\lambda}+d_i} - \frac{\tilde{\lambda}}{\gamma^2}\frac{1}{N}\sum_{i=1}^N\frac{d_i}{\tilde{\lambda}+d_i} - \frac{\tilde{\lambda}}{\gamma}\frac{1}{N}\sum_{i=1}^N\frac{d_i\partial_\gamma\tilde{\lambda}}{(\tilde{\lambda}+d_i)^2}.$$

Using Equation (H.3.11), $\frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^N\frac{d_i}{\tilde{\lambda}+d_i} = \frac{\tilde{\lambda}-\lambda}{\tilde{\lambda}}$ and thus:

$$\partial_\gamma\tilde{\lambda}\left[\frac{\lambda}{\tilde{\lambda}} + \frac{\tilde{\lambda}}{\gamma}\frac{1}{N}\sum_{i=1}^N\frac{d_i}{\left(\tilde{\lambda}+d_i\right)^2}\right] = -\frac{\tilde{\lambda}-\lambda}{\gamma}.$$

Since $\tilde{\lambda} \geq \lambda \geq 0$, the derivative of the effective ridge with respect to γ is negative: the function $\gamma \mapsto \tilde{\lambda}(\lambda,\gamma)$ is decreasing.

(3) Using the bound $\frac{d_i}{\tilde{\lambda}+d_i} \leq 1$ in Equation (H.3.11), we obtain $\tilde{\lambda} \leq \lambda + \frac{\tilde{\lambda}}{\gamma}$ which, when $\gamma \geq 1$, implies that $\tilde{\lambda} \leq \lambda \frac{\gamma}{\gamma-1}$.

(4) Recall that $\lambda > 0$ and that the effective ridge $\tilde{\lambda}$ is the unique fixpoint of the map $f(t) = \lambda + \frac{t}{\gamma}\frac{1}{N}\sum_i \frac{d_i}{t+d_i}$ in $\mathbb{R}_+$. The map is concave and, at $t = 0$, we have $f(t) = \lambda > 0 = t$: this implies that $f'(\tilde{\lambda}) < 1$ otherwise by concavity, for any $t \leq \tilde{\lambda}$ one would have $f(t) \leq t$. The derivative of f is $f'(t) = \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^N \frac{d_i^2}{(t+d_i)^2}$, thus $\frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^N \frac{d_i^2}{\left(\tilde{\lambda}+d_i\right)^2} < 1$. Using the fact that d_0 is the smallest eigenvalue of $K(X,X)$, i.e. $d_i \geq d_0$, we get $1 > \frac{1}{\gamma}\frac{d_0^2}{\left(\tilde{\lambda}+d_0\right)^2}$ hence $\tilde{\lambda} \geq d_0 \frac{1-\sqrt{\gamma}}{\sqrt{\gamma}}$. □

Similarily, we gather a number of properties of the derivative $\partial_\lambda \tilde{\lambda}(\lambda, \gamma)$.

Proposition H.3.11. *For $\gamma > 1$, as $\lambda \to 0$, the derivative $\partial_\lambda \tilde{\lambda}$ converges to $\frac{\gamma}{\gamma-1}$. As $\lambda\gamma \to \infty$, we have $\partial_\lambda \tilde{\lambda}(\lambda, \gamma) \to 1$.*

Proof. Differentiating both sides of Equation (H.3.11),

$$\partial_\lambda \tilde{\lambda} = 1 + \partial_\lambda \tilde{\lambda} \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^N \frac{d_i}{\tilde{\lambda}+d_i} - \tilde{\lambda}\partial_\lambda \tilde{\lambda}\frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^N \frac{d_i}{(\tilde{\lambda}+d_i)^2}.$$

Hence the derivative $\partial_\lambda \tilde{\lambda}$ satisfies the following equality

$$\partial_\lambda \tilde{\lambda}\left(1 - \frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^N \frac{d_i}{\tilde{\lambda}+d_i} + \tilde{\lambda}\frac{1}{\gamma}\frac{1}{N}\sum_{i=1}^N \frac{d_i}{(\tilde{\lambda}+d_i)^2}\right) = 1. \tag{H.3.13}$$

(1) Assuming $\gamma > 1$, from the point 3. of Proposition H.3.10, we already know that $\tilde{\lambda}(\lambda, \gamma) \leq \lambda\frac{\gamma}{\gamma-1}$ hence $\tilde{\lambda}(0, \gamma) = 0$. Actually, using similar arguments as in the proof of point 3., this holds also for $\gamma = 1$. Using the fact that $\tilde{\lambda}(0, \gamma) = 0$, we get $\partial_\lambda \tilde{\lambda}(0, \gamma) = 1 + \frac{\partial_\lambda \tilde{\lambda}(0,\gamma)}{\gamma}$, hence $\partial_\lambda \tilde{\lambda}(0, \gamma) = \frac{\gamma}{\gamma-1}$.

(2) From the first point of Proposition H.3.10, $\tilde{\lambda} \sim \lambda$ as $\lambda\gamma \to \infty$. Since Equation (H.3.13) can be expressed as:

$$\partial_\lambda \tilde{\lambda}\left(1 - \frac{1}{\gamma\lambda}\frac{1}{N}\sum_{i=1}^N \frac{d_i}{\frac{\tilde{\lambda}}{\lambda}+d_i} + \frac{1}{\gamma\lambda}\frac{\tilde{\lambda}}{\lambda}\frac{1}{N}\sum_{i=1}^N \frac{d_i}{(\frac{\tilde{\lambda}}{\lambda}+d_i)^2}\right) = 1,$$

we obtain that $\partial_\lambda \tilde{\lambda} \to 1$ as $\lambda \to \infty$. □

Variance of the predictor

By the bias-variance decomposition, in order to bound the difference between $\mathbb{E}[L(\hat{f}^{(RF)}_{\gamma,\lambda})]$ and $L(\hat{f}^{(K)}_{\tilde{\lambda}})$, we have to bound $\mathbb{E}_{\mathcal{D}}[\mathrm{Var}(f(x))]$. The law of total variance yields $\mathrm{Var}(\hat{f}(x)) = \mathrm{Var}(\mathbb{E}[\hat{f}(x)|F]) + \mathbb{E}[\mathrm{Var}[\hat{f}(x)|F]]$. By Proposition H.3.1, we have $\mathbb{E}[\hat{f}(x)|F] = K(x,X)K(X,X)^{-1}\hat{y}$ and $\mathrm{Var}[\hat{f}(x)|F] = \frac{1}{P}\|\hat{\theta}\|^2\tilde{K}(x,x)$. Hence, it remains to study $\mathrm{Var}\left(K(x,X)K(X,X)^{-1}\hat{y}\right)$ and $\mathbb{E}[\|\hat{\theta}\|^2]$. Recall that we denote $T = \frac{1}{N}\mathrm{Tr}K(X,X)$.

This section is dedicated to the proof of the variance bound of Theorem 5.1 of the paper:

Theorem 5.1 *There are constants $c_1, c_2 > 0$ depending on λ, γ, T only such that*

$$\mathrm{Var}\left(K(x,X)K(X,X)^{-1}\hat{y}\right) \leq \frac{c_1 K(x,x)\|y\|^2_{K^{-1}}}{P}$$
$$\left|\mathbb{E}\|[\hat{\theta}\|^2] - \partial_\lambda\tilde{\lambda} y^T M_{\tilde{\lambda}} y\right| \leq \frac{c_2\|y\|^2_{K^{-1}}}{P},$$

where $\partial_\lambda\tilde{\lambda}$ is the derivative of $\tilde{\lambda}$ with respect to λ and for $M_{\tilde{\lambda}} = K(X,X)(K(X,X)+\tilde{\lambda}I_N)^{-2}$. As a result

$$\mathrm{Var}\left(\hat{f}^{(RF)}_\lambda(x)\right) \leq \frac{c_3 K(x,x)\|y\|^2_{K^{-1}}}{P},$$

where $c_3 > 0$ depends on λ, γ, T.

• **Bound on** $\mathrm{Var}\left(K(x,X)K(X,X)^{-1}\hat{y}\right)$**.** We first study the covariance of the entries of the matrix

$$A_\lambda = \frac{1}{P}K^{\frac{1}{2}}W^T\left(\frac{1}{P}WKW^T + \lambda \mathrm{I}_P\right)^{-1}WK^{\frac{1}{2}},$$

where $K = \mathrm{diag}(d_1, \ldots, d_N)$ is a positive definite diagonal matrix and W is a $P\times N$ matrix with i.i.d. Gaussian entries. In the next proposition we show a $\frac{c_1}{P}$ bound for the covariance of the entries of A_λ, then we exploit this result in order to prove the bound on the variance of $K(x,X)K(X,X)^{-1}\hat{y}$.

Proposition H.3.12. *There exists a constant $c_1' > 0$ depending on λ, γ, and $\frac{1}{N}\mathrm{Tr}(K)$ only, such that the following bounds hold:*

$$|\mathrm{Cov}\left((A_\lambda)_{ii}, (A_\lambda)_{jj}\right)| \leq \frac{c_1'}{P}$$
$$\mathrm{Var}\left((A_\lambda)_{ij}\right) \leq \min\left\{\frac{d_i}{d_j}, \frac{d_j}{d_i}\right\}\frac{c_1'}{P}.$$

For all other cases (i.e. if i,j, k and l take more than two different values), $\mathrm{Cov}\left((A_\lambda)_{ij}, (A_\lambda)_{kl}\right) = 0$.

Proof. We want to study the covariances $\mathrm{Cov}\left((A_\lambda)_{ij}, (A_\lambda)_{kl}\right)$ for any i,j,k,l. Using the same symmetry argument as in the proof of Proposition H.3.7, $\mathbb{E}\left[(A_\lambda)_{ij}(A_\lambda)_{kl}\right] = 0$ whenever each value in $\{i,j,k,l\}$ does not appear an even number of times in (i,j,k,l). Using the fact that A_λ is symmetric, it remains to study $\mathrm{Cov}\left((A_\lambda)_{ii}, (A_\lambda)_{jj}\right)$, $\mathrm{Var}\left((A_\lambda)_{ii}\right)$ and $\mathrm{Var}\left[(A_\lambda)_{ij}\right]$ for all $i \neq j$. By the Cauchy-Schwarz inequality, any bound on $\mathrm{Var}\left((A_\lambda)_{ii}\right)$ will imply a similar bound on $\mathrm{Cov}\left((A_\lambda)_{ii}, (A_\lambda)_{jj}\right)$. Besides, as we have seen in the proof of Proposition H.3.7, $\mathbb{E}\left[(A_\lambda)_{ij}\right] = 0$ for any $i \neq j$. Thus, we only have to study $\mathrm{Var}\left((A_\lambda)_{ii}\right)$ and $\mathbb{E}\left[(A_\lambda)^2_{ij}\right]$.

• Bound on $\mathrm{Var}\left((A_\lambda)_{ii}\right)$: From Equation (H.3.9),

$$\mathrm{Var}\left((A_\lambda)_{ii}\right) = \mathrm{Var}\left(\frac{d_i g_i}{1+d_i g_i}\right) = \mathrm{Var}\left(1 - \frac{1}{1+d_i g_i}\right) = \mathrm{Var}\left(\frac{1}{1+d_i g_i}\right) \leq \mathbb{E}\left[\left(\frac{1}{1+d_i g_i} - \frac{1}{1+d_i\tilde{m}}\right)^2\right],$$

where $g_i := g_i(-\lambda)$. Again, we use the first order Taylor approximation $\mathrm{T}h$ of $h : x \to \frac{1}{1+d_i x}$ centered at $\tilde{m} := \tilde{m}(-\lambda)$, as well as the bound (H.3.7), to obtain

$$\mathbb{E}\left[\left(\frac{1}{1+d_i g_i} - \frac{1}{1+d_i\tilde{m}}\right)^2\right] = \mathbb{E}\left[\left(-\frac{d_i}{(1+d_i\tilde{m})^2}(g_i - \tilde{m}) + h(g_i) - \mathrm{T}h(g_i)\right)^2\right]$$

$$\leq \frac{2d_i^2}{(1+d_i\tilde{m})^4}\mathbb{E}\left[(g_i-\tilde{m})^2\right] + 2\mathbb{E}\left[(h(g_i)-\mathrm{T}h(g_i))^2\right]$$
$$\leq \frac{2}{6\tilde{m}^2}\mathbb{E}\left[(g_i-\tilde{m})^2\right] + \frac{2}{\tilde{m}^4}\mathbb{E}\left[(g_i-\tilde{m})^4\right].$$

Using Lemma H.3.4, we get $\mathrm{Var}\left((A_\lambda)_{ii}\right) \leq \frac{c_1'}{P}$, where $c_1' > 0$ depends on λ, γ, and $\frac{1}{N}\mathrm{Tr}(K)$ only.

• Bound on $\mathbb{E}\left((A_\lambda)_{ij}\right)$ for $i \neq j$: Following the same arguments as for Equation (H.3.9), $(A_\lambda)_{ij}$ is equal to

$$(A_\lambda)_{ij} = \frac{\sqrt{d_i d_j}}{P}\left[w_i^T B_{(i)}^{-1} w_j - \frac{d_i g_i}{1+d_i g_i} w_i^T B_{(i)}^{-1} w_j\right] = \frac{\sqrt{d_i d_j}}{1+d_i g_i}\frac{1}{P} w_i^T B_{(i)}^{-1} w_j,$$

where we set $B_{(i)} := B_i(-\lambda)$. Since w_i and $B_{(i)}$ are independent, $\mathbb{E}\left[\left(w_i^T B_{(i)}^{-1} w_j\right)^2\right] = \mathbb{E}\left[w_j^T B_{(i)}^{-2} w_j\right]$, and thus, by the Cauchy-Schwarz inequality, we have

$$\mathbb{E}\left[(A_\lambda)_{ij}^2\right] \leq \frac{1}{P^2}\sqrt{\mathbb{E}\left[\frac{d_i^2 d_j^2}{(1+d_i g_i)^4}\right]}\sqrt{\mathbb{E}\left[\left(w_j^T B_{(i)}^{-2} w_j\right)^2\right]}. \tag{H.3.14}$$

Recall that $\tilde{m} := \tilde{m}(-\lambda)$. Using the fact that $\frac{1}{1+d_i g_i} = \frac{1}{1+d_i\tilde{m}} + \frac{1}{1+d_i g_i} - \frac{1}{1+d_i\tilde{m}}$ and inserting the first Taylor approximation $\mathrm{T}h$ of $h : x \to \frac{1}{1+d_i x}$ centered at $\tilde{m}$, we get:

$$\mathbb{E}\left[\left(\frac{1}{1+d_i g_i}\right)^4\right] = \mathbb{E}\left[\left(\frac{1}{1+d_i\tilde{m}} - \frac{d_i}{(1+d_i\tilde{m})^2}(g_i-\tilde{m}) + h(g_i) - \mathrm{T}h(g_i)\right)^4\right].$$

Using a convexity argument, the bound (H.3.7), and the lower bound on $\tilde{m}$ given by Lemma H.3.6, there exists three constants $\tilde{c}_1$, $\tilde{c}_2$, $\tilde{c}_3$, which depend on λ, γ and $\frac{1}{N}\mathrm{Tr}(K)$ only, such that $\mathbb{E}\left[\left(\frac{1}{1+d_i g_i}\right)^4\right]$ is bounded by

$$\frac{\tilde{c}_1}{(1+d_i\tilde{m})^4} + \frac{\tilde{c}_2 d_i^4}{(1+d_i\tilde{m})^8}\mathbb{E}\left[(g_i-\tilde{m})^4\right] + \tilde{c}_3\mathbb{E}\left[(g_i-\tilde{m})^8\right].$$

Thanks to Lemma H.3.4 and Proposition H.3.5, this last expression can be bounded by an expression of the form $\frac{\tilde{e}_1}{d_i^4} + \frac{\tilde{e}_2}{P^2 d_i^4} + \frac{\tilde{e}_3}{P^4}$. Note that $\frac{\tilde{e}_2}{P^2 d_i^4} \leq \frac{\tilde{e}_2}{d_i^4}$ and $\frac{\tilde{e}_3}{P^4} \leq \frac{\tilde{e}_3}{\gamma^4}\frac{(\frac{1}{N}\mathrm{Tr}(K))^4}{d_i^4}$. Hence, we obtain the bound:

$$\mathbb{E}\left[\left(\frac{1}{1+d_i g_i}\right)^4\right] \leq \frac{\tilde{c}}{d_i^4},$$

where $\tilde{c} = \tilde{e}_1 + \tilde{e}_2 + \frac{\tilde{e}_3(\frac{1}{N}\mathrm{Tr}(K))^4)}{\gamma^4}$ depends on λ, γ and and $\frac{1}{N}\mathrm{Tr}(K)$ only.

Let us now consider the second term in the r.h.s. of (H.3.14) . Using the fact that $\|B_{(i)}\|_{op} \geq \frac{1}{\lambda}$, we get

$$\sqrt{\mathbb{E}\left[\left(w_j^T B_{(i)}^{-2} w_j\right)^2\right]} \leq \sqrt{\frac{1}{\lambda^4}\mathbb{E}\left[\left(w_j^T w_j\right)^2\right]} = \sqrt{\frac{1}{\lambda^4}N(N+2)} \leq \frac{N+1}{\lambda^2},$$

where we have used the fact that the second moment of a $\chi^2(N)$ distribution is $N(N+2)$. Together, we obtain

$$\begin{aligned}\mathbb{E}\left[(A)_{ij}^2\right] &\leq \frac{1}{P^2}\sqrt{\mathbb{E}\left[\frac{d_i^2 d_j^2}{(1+d_i g_i)^4}\right]}\sqrt{\mathbb{E}\left[\left(w_j^T B_{(i)}^{-2} w_j\right)^2\right]} \\ &\leq \frac{\tilde{c} d_i d_j}{d_i^2}\frac{N+1}{P^2\lambda^2} \\ &\leq \frac{\tilde{c} d_j}{P d_i \lambda^2 \gamma}\frac{N+1}{N} \leq \frac{c_1'}{P}\frac{d_i}{d_j},\end{aligned}$$

for $c_1' = 2\frac{\tilde{c}}{\lambda^2\gamma}$. Since the matrix A_λ is symmetric, we finally conclude that

$$\mathbb{E}\left[(A_\lambda)_{ij}^2\right] \leq \frac{c_1'}{P}\min\left\{\frac{d_i}{d_j}, \frac{d_j}{d_i}\right\}.$$

Note that c_1' is a constant related to the bounds constructed in Lemma H.3.2 and Proposition H.3.5 and as such it depends on $\frac{1}{N}\mathrm{Tr}(K)$, γ and λ only. □

Proposition H.3.13. *There exists a constant $c_1 > 0$ (depending on λ, γ, T only) such that the variance of the estimator is bounded by*

$$\mathrm{Var}\left(K(x,X)K(X,X)^{-1}\hat{y}\right) \leq \frac{c_1 \|y\|_{K^{-1}}^2 K(x,x)}{P}.$$

Proof. As in the proof of Theorem H.3.8, with the right change of basis, we may assume the Gram matrix $K(X,X)$ to be diagonal.

We first express the covariances of $\hat{y} = A(-\lambda)y$. Using Proposition Proposition H.3.12, for $i \neq j$ we have

$$\mathrm{Cov}\left(\hat{y}_i, \hat{y}_j\right) = \sum_{k,l=1}^{N}\mathrm{Cov}\left((A_\lambda)_{ik}, (A_\lambda)_{lj}\right) y_k y_l = \mathrm{Cov}\left((A_\lambda)_{ii}, (A_\lambda)_{jj}\right) y_i y_j + \mathbb{E}\left[(A_\lambda)_{ij}^2\right] y_j y_i,$$

whereas for $i = j$ we have

$$\mathrm{Cov}\left(\hat{y}_i, \hat{y}_i\right) = \sum_{k=1}^{N}\mathrm{Cov}\left((A_\lambda)_{ik}, (A_\lambda)_{ki}\right) y_k^2 = \mathrm{Var}\left((A_\lambda)_{ii}\right) y_i^2 + \sum_{k\neq i}\mathbb{E}\left[(A_\lambda)_{ik}^2\right] y_k^2.$$

We decompose $K^{-\frac{1}{2}}\mathrm{Cov}(\hat{y},\hat{y})K^{-\frac{1}{2}}$ into two terms: let C be the matrix of entries

$$C_{ij} = \frac{\mathrm{Cov}((A_\lambda)_{ii}, (A_\lambda)_{jj}) + \delta_{i\neq j}\mathbb{E}\left[(A_\lambda)_{ij}^2\right]}{\sqrt{d_i d_j}} y_i y_j,$$

and let D the diagonal matrix with entries

$$D_{ii} = \frac{\sum_{k\neq i}\mathbb{E}\left[(A_\lambda)_{ik}^2\right] y_k^2}{d_i}.$$

We have the decomposition $K^{-\frac{1}{2}}\mathrm{Cov}(\hat{y},\hat{y})K^{-\frac{1}{2}} = C + D$.

Proposition H.3.12 asserts that $\mathrm{Cov}((A_\lambda)_{ii},(A_\lambda)_{jj} \leq \frac{c_1'}{P}$ and $\mathbb{E}\left[(A_\lambda)_{ij}^2\right] \leq \frac{c_1'}{P}$, and thus the operator norm of C is bounded by

$$\begin{aligned}\|C\|_{op} &\leq \|C\|_F \\ &= \sqrt{\sum_{i,j} \frac{\left(\mathrm{Cov}((A_\lambda)_{ii},(A_\lambda)_{jj}) + \delta_{i\neq j}\mathbb{E}\left[(A_\lambda)_{ij}^2\right]\right)^2}{d_i d_j} y_i^2 y_j^2} \\ &\leq \frac{2c_1'}{P}\sqrt{\sum_{ij}\frac{1}{d_i d_j}y_i^2 y_j^2} \;=\; \frac{2c_1'\|y\|_{K^{-1}}^2}{P}\end{aligned}$$

For the matrix D, we use the bound $\mathbb{E}\left[(A_\lambda)_{ik}^2\right] \leq \frac{c_1'}{P}\frac{d_i}{d_k}$ to obtain

$$D_{ii} = \frac{\sum_{k\neq i}\mathbb{E}\left[(A_\lambda)_{ik}^2\right] y_k^2}{d_i} \leq \frac{c_1'}{P}\sum_{k\neq i}\frac{y_k^2}{d_k} \leq \frac{c_1'\|y\|_{K^{-1}}^2}{P},$$

which implies that $\|D\|_{op} \leq \frac{c_1'\|y\|_{K^{-1}}^2}{P}$. As a result

$$\begin{aligned}\mathrm{Var}\left(K(x,X)K^{-1}\hat{y}\right) &= K(x,X)K^{-1}\mathrm{Cov}(\hat{y},\hat{y})K^{-1}K(X,x) \\ &\leq K(x,X)K^{-\frac{1}{2}}\|C+D\|_{op}K^{-\frac{1}{2}}K(X,x) \\ &\leq \frac{3c_1'\|y\|_{K^{-1}}^2}{P}\|K(x,X)\|_{K^{-1}}^2 \\ &\leq \frac{3c_1'K(x,x)\|y\|_{K^{-1}}^2}{P},\end{aligned}$$

where we used Inequality (H.3.12). This yields the result with $c_1 = 3c_1'$. □

• **Bound on $\mathbb{E}_\pi\left[\|\hat{\theta}\|^2\right]$.** To understand the variance of the λ-RF estimator $\hat{f}_\lambda^{(RF)}$, we need to describe the distribution of the squared norm of the parameters:

Proposition H.3.14. *For $\gamma,\lambda > 0$ there exists a constant $c_2 > 0$ depending on λ,γ,T only such that*

$$\left|\mathbb{E}[\|\hat{\theta}\|^2] - \partial_\lambda\tilde{\lambda}y^T K(X,X)\left(K(X,X)+\tilde{\lambda}I_N\right)^{-2}y\right| \leq \frac{c_2\|y\|_{K^{-1}}^2}{P}. \tag{H.3.15}$$

Proof. As in the proof of Theorem H.3.8, with the right change of basis, we may assume the Gram matrix $K(X,X)$ to be diagonal. Recall that $\hat{\theta} = \frac{1}{\sqrt{P}}\left(\frac{1}{P}WK(X,X)W^T + \lambda I_N\right)^{-1}WK(X,X)^{\frac{1}{2}}y$, thus we have:

$$\|\hat{\theta}\|^2 \;=\; \frac{1}{P}y^T K(X,X)^{\frac{1}{2}}W^T(\frac{1}{P}WK(X,X)W^T + \lambda I_P)^{-2}WK(X,X)^{\frac{1}{2}}y \;=\; y^T A'(-\lambda)y, \tag{H.3.16}$$

where $A'(-\lambda)$ is the derivative of

$$A(z) = \frac{1}{P}K(X,X)^{\frac{1}{2}}W^T\left(\frac{1}{P}WK(X,X)W^T - z\mathrm{I}_P\right)^{-1}WK(X,X)^{\frac{1}{2}}$$

with respect to z evaluated at $-\lambda$. Let

$$\tilde{A}(z) = K(X,X)(K(X,X) + \tilde{\lambda}(-z)\mathrm{I}_N)^{-1}.$$

Remark that the derivative of $\tilde{A}(z)$ is given by $\tilde{A}'(z) = \tilde{\lambda}'(-z)K(X,X)(K(X,X) + \tilde{\lambda}(-z)I_N)^{-2}$. Thus, from Equation (H.3.16), the l.h.s. of (H.3.15) is equal to:

$$\left| y^T \left(\mathbb{E}[A'(-\lambda)] - \tilde{A}'(-\lambda) \right) y \right|. \tag{H.3.17}$$

Using a classical complex analysis argument, we will show that $\mathbb{E}[A'(-\lambda)]$ is close to $\tilde{A}'(-\lambda)$ by proving a bound of the difference between $\mathbb{E}[A(z)]$ and $\tilde{A}(z)$ for any $z \in \mathbb{H}_{<0}$.

Note that the proof of Proposition H.3.7 provides a bound on the diagonal entries of $\mathbb{E}[A(z)]$, namely that for any $z \in \mathbb{H}_{<0}$,

$$\left| \mathbb{E}[(A(z))_{ii}] - (\tilde{A}(z))_{ii} \right| \leq \frac{c}{P},$$

where $\hat{c}$ depends on z, γ and T only. Actually, in order to prove (H.3.15), we will derive the following slightly different bound: for any $z \in \mathbb{H}_{<0}$,

$$\left| \mathbb{E}[(A(z))_{ii}] - (\tilde{A}(z))_{ii} \right| \leq \frac{\hat{c}}{d_i P}, \tag{H.3.18}$$

where $\hat{c}$ depends on z, γ and T only. Let $g_i := g_i(z)$ and $\tilde{m} := \tilde{m}(z)$. Recall that for $h_i : x \mapsto \frac{d_i x}{1+d_i x}$, one has $(A(z))_{ii} = h_i(g_i)$, $(\tilde{A}(z))_{ii} = h_i(\tilde{m})$ and

$$\begin{aligned} \mathrm{T}_{\tilde{m}} h_i(g_i) &= \frac{d_i \tilde{m}}{1 + d_i \tilde{m}} - \frac{d_i \left(g_i - \tilde{m}\right)}{\left(1 + d_i \tilde{m}\right)^2}, \\ h_i(g_i) - \mathrm{T}_{\tilde{m}} h_i(g_i) &= \frac{d_i^2 \left(g_i - \tilde{m}\right)^2}{\left(1 + d_i g_i\right)\left(1 + d_i \tilde{m}\right)^2}, \end{aligned}$$

where $\mathrm{T}_{\tilde{m}} h_i$ is the first order Taylor approximation of h_i centered at $\tilde{m}$. Using this first order Taylor approximation, we can bound the difference $|\mathbb{E}[h_i(g_i)] - h_i(\tilde{m})|$:

$$\begin{aligned} |\mathbb{E}[h_i(g_i)] - h_i(\tilde{m})| &\leq \frac{d_i \left|\mathbb{E}[g_i] - \tilde{m}\right|}{\left(1 + d_i \tilde{m}\right)^2} + \frac{d_i^2}{\left(1 + d_i \tilde{m}\right)^2} \mathbb{E}\left[\frac{\left|g_i - \tilde{m}\right|^2}{1 + d_i g_i} \right] \\ &\leq \frac{\mathbf{a}}{d_i P} + \mathbf{a} \sqrt{\mathbb{E}\left[\frac{1}{\left(1 + d_i g_i\right)^2} \right] \mathbb{E}\left[\left| g_i - \tilde{m} \right|^4 \right]}, \end{aligned}$$

where $\mathbf{a}$ depends on z, γ and T. We need to bound $\mathbb{E}\left[\frac{1}{(1+d_i g_i)^2}\right]$. Recall that in the proof of Proposition H.3.12, we bounded $\mathbb{E}\left[\frac{1}{(1+d_i g_i)^4}\right]$. Using similar arguments, one shows that

$$\mathbb{E}\left[\frac{1}{\left(1 + d_i g_i\right)^2} \right] \leq \frac{\hat{e}^2}{d_i^2},$$

where $\hat{e}$ depends on z, γ and $\frac{1}{N}\mathrm{Tr}(K(X,X))$ only. The term $\mathbb{E}\left[|g_i - \tilde{m}|^4\right]$ is bounded using Lemmas H.3.4, H.3.2 and Proposition H.3.5. This allows us to conclude that:

$$|\mathbb{E}[h_i(g_i)] - h_i(\tilde{m})| \leq \frac{\hat{c}}{d_i P},$$

where $\hat{c}$ depends on z, γ and $\frac{1}{N}\mathrm{Tr}(K(X,X))$ only, hence we obtain the Inequality (H.3.18).

We can now prove Inequality H.3.15. We bound the difference of the derivatives of the diagonal terms of $A(z)$ and $\tilde{A}(z)$ by means of Cauchy formula. Consider a simple closed path $\phi : [0,1] \to \mathbb{H}_{<0}$ which surrounds z. Since

$$\mathbb{E}[(A'(z))_{ii}] - (\tilde{A}'(z))_{ii} = \frac{1}{2\pi i}\oint_\phi \frac{\mathbb{E}[(A(z))_{ii}] - (\tilde{A}(z))_{ii}}{(w-z)^2} dw,$$

using the bound (H.3.18), we have:

$$\left|\mathbb{E}[(A'(z))_{ii}] - (\tilde{A}'(z))_{ii}\right| \leq \frac{\hat{c}}{d_i P}\frac{1}{2\pi}\oint_\phi \frac{1}{|w-z|^2} dw \leq \frac{c_2}{d_i P},$$

where c_2 depends on z, γ, and T only. This allows one to bound the operator norm of $K(X,X)(\mathbb{E}[A'(z)] - \tilde{A}'(z))$:

$$\|K(X,X)(\mathbb{E}[A'(z)] - \tilde{A}'(z))\|_{op} \leq \frac{c_2}{P}.$$

Using this bound and (H.3.17), we have

$$\left|\mathbb{E}[\|\hat{\theta}\|^2] - \partial_\lambda \tilde{\lambda}\; y^T K(X,X)\left(K(X,X) + \tilde{\lambda} I_N\right)^{-2} y\right| = \left|y^T\left(\mathbb{E}[A'(-\lambda)] - \tilde{A}'(-\lambda)\right) y\right| \leq \frac{c_2 \|y\|_{K^{-1}}^2}{P},$$

which allows us to conclude. □

• **Bound on** $\mathrm{Var}\left(\hat{f}_\lambda^{(RF)}(x)\right)$**.** We have shown all the bounds needed in order to prove the following proposition.

Proposition H.3.15. *For any $x \in \mathbb{R}^d$, we have*

$$\mathrm{Var}\left(\hat{f}_\lambda^{(RF)}(x)\right) \leq \frac{c_3 K(x,x)\|y\|_{K^{-1}}^2}{P},$$

where $c_3 > 0$ depends on λ, γ, T.

Proof. Recall that for any $x \in \mathbb{R}^d$,

$$\begin{aligned}\mathrm{Var}(\hat{f}_\lambda^{(RF)}(x)) &= \mathrm{Var}\left(\mathbb{E}\left[\hat{f}_\lambda^{(RF)}(x) \mid F\right]\right) + \mathbb{E}\left[\mathrm{Var}\left[\hat{f}_\lambda^{(RF)}(x) \mid F\right]\right] \\ &= \mathrm{Var}\left(K(x,X)K(X,X)^{-1}\hat{y}\right) + \frac{1}{P}\mathbb{E}\left[\|\hat{\theta}\|^2\right]\left[K(x,x) - K(x,X)K(X,X)^{-1}K(X,x)\right].\end{aligned}$$

From Proposition H.3.13,

$$\mathrm{Var}\left(K(x,X)K(X,X)^{-1}\hat{y}\right) \leq \frac{c_1 K(x,x)\|y\|_{K^{-1}}^2}{P},$$

and from Proposition H.3.14, we have:

$$\mathbb{E}\left[\|\hat{\theta}\|^2\right] \leq \partial_\lambda \tilde{\lambda}\ y^T K \left(K + \tilde{\lambda} I_N\right)^{-2} y + \frac{c_2 \|y\|_{K^{-1}}^2}{P} \leq \partial_\lambda \tilde{\lambda}\ \|y\|_{K^{-1}}^2 + \frac{c_2 \|y\|_{K^{-1}}^2}{P} \leq \alpha \|y\|_{K^{-1}}^2,$$

where $\alpha = \partial_\lambda \tilde{\lambda} + c_2$. Using the fact that $\tilde{K}(x,x) \leq K(x,x)$, we get

$$\begin{aligned}\mathbb{E}\left[\mathrm{Var}\left[\hat{f}(x) \mid F\right]\right] &= \frac{1}{P}\mathbb{E}\left[\|\hat{\theta}\|^2\right]\left[K(x,x) - K(x,X)K(X,X)^{-1}K(X,x)\right] \\ &\leq \frac{\alpha \|y\|_{K^{-1}}^2 K(x,x)}{P}.\end{aligned}$$

This yields

$$\mathrm{Var}\left(\hat{f}_\lambda^{(RF)}(x)\right) \leq \frac{c_3 \|y\|_{K^{-1}}^2 K(x,x)}{P},$$

where $c_3 = \alpha + c_1$. □

Average loss of λ-RF predictor and loss of $\tilde{\lambda}$-KRR:

Putting the pieces together, we obtain the following bound on the difference $\Delta_E = |\mathbb{E}[L(\hat{f}_{\lambda,\gamma}^{(RF)})] - L(\hat{f}_{\tilde{\lambda}}^{(K)})|$ between the expected RF loss and the KRR loss:

Corollary H.3.16. *If $\mathbb{E}_{\mathcal{D}}[K(x,x)] < \infty$, we have*

$$\Delta_E \leq \frac{C_1 \|y\|_{K^{-1}}}{P}\left(2\sqrt{L(\hat{f}_{\tilde{\lambda}}^{(K)})} + C_2 \|y\|_{K^{-1}}\right),$$

where C_1 and C_2 depend on λ, γ, T and $\mathbb{E}_{\mathcal{D}}[K(x,x)]$ only.

Proof. Using the bias/variance decomposition, Corollary H.3.9, and the bound on the variance of the predictor, we obtain

$$\begin{aligned}\left|\mathbb{E}\left[L\left(\hat{f}_{\gamma,\lambda}^{(RF)}\right)\right] - L\left(\hat{f}_{\tilde{\lambda}}^{(K)}\right)\right| &\leq \left|L\left(\mathbb{E}\left[\hat{f}_{\gamma,\lambda}^{(RF)}\right]\right) - L\left(\hat{f}_{\tilde{\lambda}}^{(K)}\right)\right| + \mathbb{E}_{\mathcal{D}}\left[\mathrm{Var}\left(\hat{f}(x)\right)\right] \\ &\leq \frac{C\|y\|_{K^{-1}}}{P}\left(2\sqrt{L\left(\hat{f}_{\tilde{\lambda}}^{(K)}\right)} + \frac{C\|y\|_{K^{-1}}}{P}\right) + \frac{c_3 \|y\|_{K^{-1}}^2 \mathbb{E}_{\mathcal{D}}\left[K(x,x)\right]}{P} \\ &\leq \frac{C_1\|y\|_{K^{-1}}}{P}\left(2\sqrt{L\left(\hat{f}_{\tilde{\lambda}}^{(K)}\right)} + C_2\|y\|_{K^{-1}}\right),\end{aligned}$$

where C_1 and C_2 depends on λ, γ, T and $\mathbb{E}_{\mathcal{D}}\left[K(x,x)\right]$ only. □

Double descent curve

Recall that for any $\tilde{\lambda}$, we denote $M_{\tilde{\lambda}} = K(X,X)(K(X,X) + \tilde{\lambda} I_N)^{-2}$. A direct consequence of Proposition H.3.14 is the following lower bound on the variance of the predictor.

Corollary H.3.17. *There exists $c_4 > 0$ depending on λ, γ, T only such that* $\mathrm{Var}\left(\hat{f}_\lambda^{(RF)}(x)\right)$ *is bounded from below by*

$$\partial_\lambda \tilde{\lambda} \frac{y^T M_{\tilde{\lambda}} y}{P} \tilde{K}(x,x) - \frac{c_4 K(x,x) \|y\|_{K^{-1}}^2}{P^2}.$$

Proof. By the law of total cumulance,

$$\operatorname{Var}\left(\hat{f}_\lambda^{(RF)}(x)\right) \geq \mathbb{E}\left[\operatorname{Var}\left[\hat{f}_\lambda^{(RF)}(x) \mid F\right]\right] \geq \frac{1}{P}\mathbb{E}\left[\|\hat{\theta}\|^2\right]\tilde{K}(x,x).$$

From Proposition H.3.14, $\mathbb{E}[\|\hat{\theta}\|^2] \geq \partial_\lambda\tilde{\lambda}\; y^T M_{\tilde{\lambda}} y - \frac{c_2\|y\|_{K^{-1}}^2}{P}$, hence

$$\operatorname{Var}\left(\hat{f}_\lambda^{(RF)}(x)\right) \geq \partial_\lambda\tilde{\lambda}\frac{y^T M_{\tilde{\lambda}} y}{P}\tilde{K}(x,x) - \frac{c_4\tilde{K}(x,x)\|y\|_{K^{-1}}^2}{P^2}.$$

The result follows from the fact that $\tilde{K}(x,x) \leq K(x,x)$. □

Appendix I

Saddle-to-Saddle Dynamics in Deep Linear Networks: Small Initialization Training, Symmetry and Sparsity

We organize the Appendix as follows:

- In Section I.1, we present the details for the numerical results presented in the main text together with some discussions.
- In Section I.2, we present the proofs for the result on the proximity of critical points, i.e. Theorem 9.1.
- In Section I.3, we present the proofs for the Saddle-to-Saddle regime, in particular Theorems 9.2 and 9.4.
- In Section I.4, we state and prove a few technical results.

I.1 Further Experimental Details

Experimental details of Fig. 9.4.1: A teacher network matrix of size 5×5 is generated as $10\text{diag}([1, 2, 3, 4, 5])$. The input data is i.i.d. 5-dim. standard Gaussian samples, and the number of training samples is 100. The labels are generated by the teacher, no noise is added. Training is performed with gradient descent for 50000 epochs and a learning rate of $1e-4$ is used.

Experimental details of Fig. 9.6.1: A random matrix A^* of size 30×30 is generated by multiplying two i.i.d. matrices of size 30×1 with i.i.d. standard Gaussian entries. 0.2 of the entries of this matrix was accecible in training, and the training objecive is the squared difference between the (observed) entries of the linear network matrix and those of the matrix A^*. The training is performed for 20000 gradient descent iterations with a learning rate of $\eta_0 = 0.05$ if $\gamma > 1$, and $\eta = \eta_0 w^{(L-1)(\gamma-1)}$ for $\gamma \leq 1$. The tolerance for computing the rank is set to 0.1.

Experimental details of Fig. I.1.3 in the Appendix: We created a rank 3 teacher weight matrix $W_T = W_0 W_0^T$ of size 10×10 where W_0 is a 10×3 matrix with all entries independent Gaussian with zero mean and where all entries in i-th column has variance i for all $i \in \{1, 2, 3\}$. We corrupted the teacher weight matrix by an addition of a 10×10 matrix where each entry is

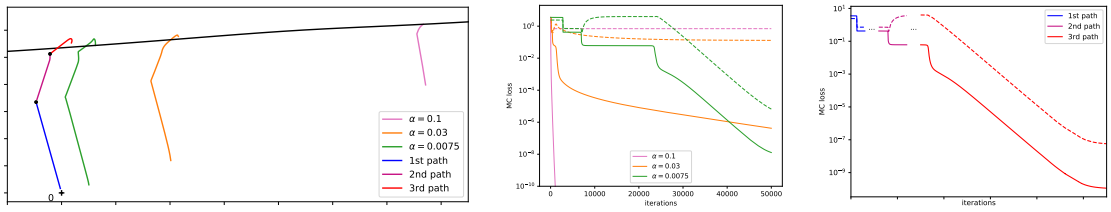


Figure I.1.1: *Matrix Completion in linear/lazy vs. saddle-to-saddle regimes.* 3 DLNs ($L = 4, w = 100$) trained on a MC loss fitting a 10×10 matrix of rank 3 with initialization $\alpha\theta_0$ for a fixed random θ_0 and three values of α. **Left:** Train (solid) and test (dashed) MC cost for the three networks, for large α the network is in the linear/lazy regime and does not learn the low-rank structure. For smaller α plateaus appear and the network generalizes. **Middle:** Visualization of the gradient paths in parameter space. The black line represents the manifold of solutions to which all example paths converge. As $\alpha \to 0$ the training trajectory converges to a sequence of 3 paths (in blue, purple and red) starting from the origin ($+$) and passing through 2 saddles ($\cdot$) before converging. **Right:** The train (solid) and test (dashed) loss of the three paths plotted sequentially, in the saddle-to-saddle limit; $\cdots$ represent an infinite amount of steps separating these paths.

i.i.d. centered Gaussian with standard deviation 0.2. Input points are isotropic Gaussians. The training outputs are generated by the noisy teacher, and the test outputs are generated by the noiseless teacher. We generated 100 training and 1000 test data points. Different runs of the same experiment yielded effectively the same figure. The learning rate is 0.001 both for the shallow and the deep case. Tolerance for the rank is set to 10^{-4} (i.e. eigenvalues smaller than 10^{-4} are set to 0 for the rank calculation).

I.2 Regimes of Training

In this section we describe the regimes of training depending on the scaling γ of the variance at initialization $\sigma^2 = w^{-\gamma}$.

Equivalence of Parametrization/Initializations

NTK Parametrization

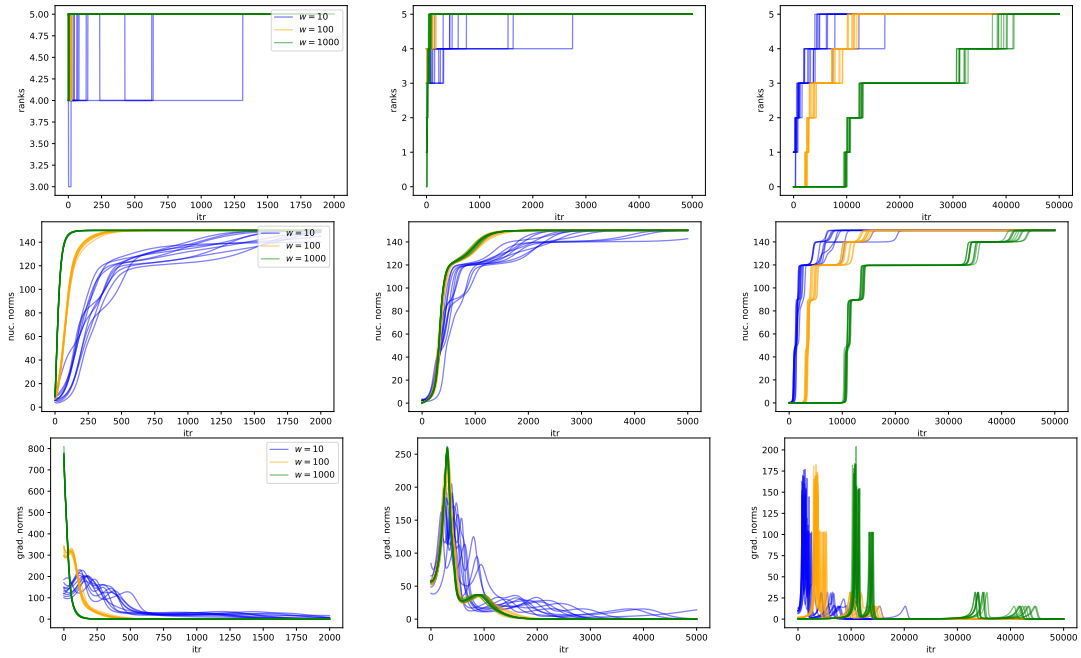
Let us show that the NTK parametrization corresponds to a scaling of $\gamma = 1 - \frac{1}{L}$.

The NTK parametrization [105] for linear networks is

$$A_\theta^{NTK} = \frac{W_L}{\sqrt{n_{L-1}}} \cdots \frac{W_1}{\sqrt{n_0}} = \frac{1}{\sqrt{n_0 \cdots n_{L-1}}} W_L \cdots W_1$$

with all parameters initialized with a variance of 1. One can show that gradient flow $\theta^{NTK}(t)$ with the NTK parametrization, initialized at some parameters θ_0^{NTK} is equivalent (up to a rescaling of the learning rate) to gradient flow $\theta(t)$ with the classical parametrization with an initialization of $\theta_0 = (n_0 \cdots n_{L-1})^{-\frac{1}{2L}} \theta_0^{NTK}$:

Proposition I.1. *Let $\theta^{NTK}(t)$ be gradient flow on the loss $\mathcal{L}^{NTK}(\theta) = C(A_\theta^{NTK})$ initialized at some parameters θ_0^{NTK} and $\theta(t)$ be gradient flow on the cost $\mathcal{L}(\theta) = C(A_\theta)$ initialized at $\theta_0 =$*



(a) $\gamma = 0.75$ (NTK) **(b)** $\gamma = 1$ (MF) **(c)** $\gamma = 1.5$ (S-S)

Figure I.1.2: *Training in* ***(a)*** *the NTK regime,* ***(b)*** *mean-field,* ***(c)*** *saddle-to-saddle regimes in deep linear networks for three widths* $w = 10, 100, 1000$*,* $L = 4$*, and* 10 *seeds; extension of Fig. 9.4.1 in the main.* **Top:** The evolution of the rank of the network matrices during training. Tolerance of the matrix is set at $1e-1$. **Middle:** The evolution of the nuclear norm during training, we can see that the smooth jumps are aligned with the rank transitions. **Bottom:** The evolution of the gradient norm of the parameters. Decrease of the gradient norm down to zero indicates approaching to a saddle, and the following increase indicates escaping it.

$(n_0 \cdots n_{L-1})^{-\frac{1}{2L}} \theta_0^{NTK}$. *We have*

$$A_{\theta(t)} = A^{NTK}_{\theta^{NTK}(\sqrt{n_0 \cdots n_{L-1}} t)}.$$

Proof. We will show that $\theta(t) = (n_0 \cdots n_{L-1})^{-\frac{1}{2L}} \theta^{NTK}(\sqrt{n_0 \cdots n_{L-1}} t)$ which implies that $A_{\theta(t)} = A^{NTK}_{\theta^{NTK}(t)}$. This is obviously true at $t = 0$. Now assuming it is true at a time t, we show that the time derivatives of $\theta(t)$ and $(n_0 \cdots n_{L-1})^{-\frac{1}{2L}} \theta^{NTK}(\sqrt{n_0 \cdots n_{L-1}} t)$ match:

$$\partial_t \theta^{NTK}(\sqrt{n_0 \cdots n_{L-1}} t) = \frac{\sqrt{n_0 \cdots n_{L-1}}}{\sqrt{n_0 \cdots n_{L-1}}} \partial_t \theta(t) = \partial_t \theta(t).$$

□

This implies that the NTK parametrization with $\mathcal{N}(0, 1)$ initialization is equivalent to the classical parametrization with $\mathcal{N}(0, (n_0 \cdots n_{L-1})^{-\frac{1}{L}})$ initialization, which for rectangular networks corresponds to a $\mathcal{N}(0, n_0^{-\frac{1}{L}} w^{-\frac{L-1}{L}})$ initialization with scaling $\gamma = \frac{L-1}{L} = 1 - \frac{1}{L}$.

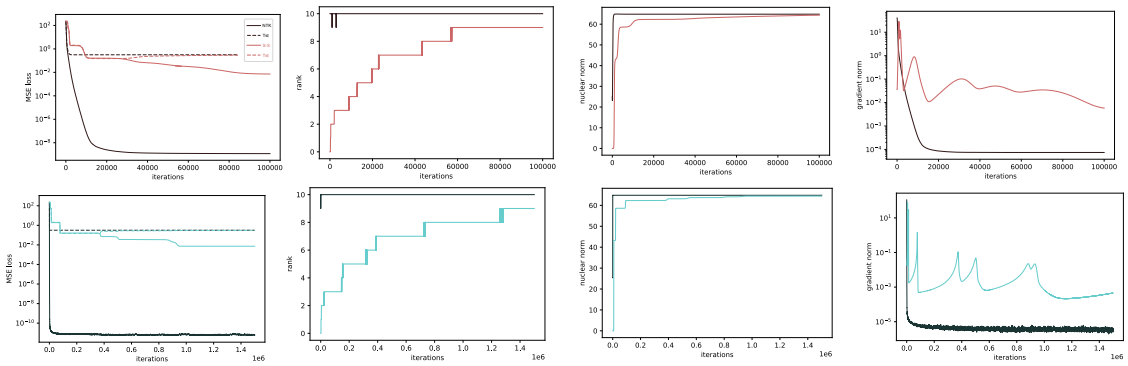


Figure I.1.3: *Training in the NTK vs. saddle-to-saddle regimes in shallow (top) and deep (bottom) networks when learning a low rank matrix corrupted with noise.* Black lines (the NTK regime): the parameters are initialized with the standard deviation $\tilde{\sigma} = w^{-L-1/2L}$. The rank of the network matrix increases incrementally as the gradient trajectory follows the paths between the saddles. **Top/Shallow case:** $L = 2$ and $w = 50$; in the saddle-to-saddle regime (shown in red), the initialization scale is $\tilde{\sigma} = w^{-2}$. Bigger initialization scales result in shorter plateaus in the loss curve if the same learning rate is used. **Bottom/Deep case:** $L = 4$ and $w = 100$; in the saddle-to-saddle regime (shown in blue), the initialization scale is $\tilde{\sigma} = w^{-1}$. We observe that the transitions from saddles to saddles are sharper. We observe that the gradient norm of the parameters is highly non-monotonic; a decrease down to 0 indicates approaching to a saddle, and a following increase indicates escaping it. We note that the peaks of the gradient norm are sharper in the deep case, suggesting a different rate of escape. In the NTK regime, the gradient norm decreases down to 0 monotonically. In the deep case the GD training is implemented for 1500000 iterations whereas in the shallow case it is only 100000 iterations.The input data is standard Gaussian, the outputs are generated by a rank 3 teacher of size 10×10 corrupted with noise, and the loss is MSE.

Maximal Update Parametrization

The Maximal Update parametrization (or μ-parametrization) [229] is equivalent to $\gamma = 1$. The μ-parametrization for linear rectangular networks is the same the classical one, since

$$A_\theta^\mu = \frac{W_L}{\sqrt{w}} W_{L-1} \cdots W_2 \left(\sqrt{w} W_1\right) = W_L \cdots W_1$$

and the parameters are initialized with variance w^{-1}, i.e. $\gamma = 1$.

Distance to Different Critical Points

Let d_{m} and d_{s} be the Euclidean distances between the initialization θ and, respectively, the set of global minima and the set of all saddles. For random variables $f(w), g(w)$ which depend on w, we write $f \asymp g$ if both $f(w)/g(w)$ and $g(w)/f(w)$ are stochastically bounded as $w \to \infty$. The following theorem studies how d_{m} and d_{s} scale as $w \to \infty$:

Theorem I.1 (Theorem 9.1 in the main)**.** *Suppose that the set of matrices that minimize C is non-empty, has Lebesgue measure zero, and does not contain the zero matrix. Let θ be i.i.d. centered Gaussian r.v. of variance $\sigma^2 = w^{-\gamma}$ where $1 - \frac{1}{L} \leq \gamma < \infty$. Then:*

1. *if* $1 - \frac{1}{L} \leq \gamma < 1$, *we have* $d_{\mathrm{m}} \asymp w^{-\frac{(1-\gamma)(L-1)}{2}}$ *and* $d_{\mathrm{s}} \asymp w^{\frac{1-\gamma}{2}}$,
2. *if* $\gamma = 1$, *we have* $d_{\mathrm{m}}, d_{\mathrm{s}} \asymp 1$,
3. *if* $\gamma > 1$ *we have* $d_{\mathrm{m}} \asymp 1$ *and* $d_{\mathrm{s}} \asymp w^{-\frac{\gamma-1}{2}}$.

To prove this result, we require a few Lemmas:

Lemma I.1. *Let* θ *be the vector of parameters of a DLN with i.i.d.* $\mathcal{N}(0, w^{-\gamma})$ *Gaussian entries, and let* $\mathcal{A}_{\min} = \{A \in \mathbb{R}^{n_L \times n_0} : C(A) = 0\}$ *be the set of global minimizers of* C. *Under the same assumptions on the cost* C *as Proposition I.1, we have* $d(A_\theta, \mathcal{A}_{\min}) \asymp 1$ *as* $w \to \infty$.

Proof. If $\gamma > 1 - \frac{1}{L}$ then A_θ converges in distribution to the zero matrix as $w \to \infty$, the distance $d(A_\theta, \mathcal{A}_{\min})$ therefore converges to the finite value $d(0, \mathcal{A}_{\min}) \neq 0$.

If $\gamma = 1 - \frac{1}{L}$, then A_θ converges in distribution to random Gaussian matrix with iid $\mathcal{N}(0,1)$ entries (this can seen as a consequence of the more general results for non-linear networks [127, 46]). As a result the distribution of $d(A_\theta, \mathcal{A}_{\min})$ converges to the distribution of $d(B, \mathcal{A}_{\min})$ for a matrix B with iid Gaussian $\mathcal{N}(0,1)$ entries. Since $\mathbb{P}\left[d(B, \mathcal{A}_{\min}) = 0\right] = 0$ and $\mathbb{P}\left[d(B, \mathcal{A}_{\min}) > b\right] \to 0$ as $b \to \infty$ we have that $d(A_\theta, \mathcal{A}_{\min}) \asymp 1$ as needed. □

Lemma I.2. *Let* θ *be the vector of parameters of a DLN with iid* $\mathcal{N}(0, w^{-\gamma})$ *Gaussian entries. For all* ϵ, *there is a constant* $C_{\epsilon,L}$ *that does not depend on* w *s.t. with prob.* $1 - \epsilon$, *we have for all* $\theta' \in \mathbb{R}^P$ *that*

$$\|A_{\theta'} - A_\theta\|_F^2 \leq C_{\epsilon,L} \sum_{k=1}^{L} \|\theta - \theta'\|^{2k} w^{(1-\gamma)(L-k)}.$$

Proof. By Corollary 5.35 in [217], reformulated as Theorem I.2 below, we know that for all ϵ, there is a constant $c_\epsilon$that does not depend on w s.t. with prob. $1 - \epsilon$, we have for all ℓ

$$\|W_\ell\|_{op}^2 \leq c_\epsilon w^{1-\gamma}.$$

We now write $d\theta = \theta' - \theta$ (and the corresponding matrices $dW_\ell = W_\ell' - W_\ell$) so that we may write the difference $A_{\theta + d\theta} - A_\theta$ as the following sum

$$\sum_{\substack{a_1, \ldots, a_L \in \{0,1\} \\ \exists \ell, a_\ell \neq 0}} \left(\begin{cases} W_L & \text{if } a_L = 0 \\ dW_L & \text{if } a_L = 1 \end{cases} \right) \cdots \left(\begin{cases} W_1 & \text{if } a_1 = 0 \\ dW_1 & \text{if } a_1 = 1 \end{cases} \right)$$

where the indicator a_ℓ determines whether we take W_ℓ or dW_ℓ in the product. We can therefore bound

$$\|A_{\theta+d\theta} - A_\theta\|_F^2 \leq \left(\sum_{\substack{a_1, \ldots, a_L \in \{0,1\} \\ \exists \ell, a_\ell \neq 0}} \left\| \left(\begin{cases} W_L & \text{if } a_L = 0 \\ dW_L & \text{if } a_L = 1 \end{cases} \right) \cdots \left(\begin{cases} W_1 & \text{if } a_1 = 0 \\ dW_1 & \text{if } a_1 = 1 \end{cases} \right) \right\|_F \right)^2$$

$$\leq (2^L-1) \sum_{\substack{a_1,\dots,a_L\in\{0,1\}\\ \exists \ell, a_\ell \neq 0}} \left\| \left(\begin{cases} W_L & \text{if } a_L = 0 \\ dW_L & \text{if } a_L = 1 \end{cases} \right) \cdots \left(\begin{cases} W_1 & \text{if } a_1 = 0 \\ dW_1 & \text{if } a_1 = 1 \end{cases} \right) \right\|_F^2$$

$$\leq (2^L-1) \sum_{\substack{a_1,\dots,a_L\in\{0,1\}\\ \exists \ell, a_\ell \neq 0}} \left(\begin{cases} \|W_L\|_{op}^2 & \text{if } a_L = 0 \\ \|dW_L\|_F^2 & \text{if } a_L = 1 \end{cases} \right) \cdots \left(\begin{cases} \|W_1\|_{op}^2 & \text{if } a_1 = 0 \\ \|dW_1\|_F^2 & \text{if } a_1 = 1 \end{cases} \right)$$

We now bound $\|W_L\|_{op}^2$ by $c_\epsilon w^{1-\gamma}$ and $\|dW_L\|_F^2$ by $\|d\theta\|^2$ so that we obtain the bound

$$\|A_{\theta+d\theta} - A_\theta\|_F^2 \leq (2^L-1) \sum_{k=1}^{L} \begin{pmatrix} L \\ k \end{pmatrix} \|d\theta\|^{2k} c_\epsilon^{L-k} w^{(1-\gamma)(L-k)} \leq C_{\epsilon,L} \sum_{k=1}^{L} \|d\theta\|^{2k} w^{(1-\gamma)(L-k)}$$

for $C_{\epsilon,L} = (2^L-1) \max_{k=1,\dots,L} \begin{pmatrix} L \\ k \end{pmatrix} c_\epsilon^{L-k}$. □

Let us now prove Theorem I.1:

Proof. **(1) Distance to minimum:** Let us first give an lower bound on the distance from initialization to a global minimum. Let θ be the intialization and $\theta + d\theta$ be the closest minimum. By Lemma I.2, we obtain

$$\|A_{\theta+d\theta} - A_\theta\|_F^2 \leq C'_L \sum_{k=1}^{L} \|d\theta\|^{2k} w^{(1-\gamma)(L-k)}.$$

If $\gamma > 1$, the term with $k = L$ dominates, in which case $\|A_{\theta+d\theta} - A_\theta\|_F^2 \leq \|d\theta\|^{2L}$ which implies that $\|d\theta\| \geq \|A_{\theta+d\theta} - A_\theta\|_F^{\frac{1}{L}} \geq d(A_\theta, \mathcal{A}_{\min})^{\frac{1}{L}} \asymp 1$ by Lemma I.1.

If $\gamma < 1$, the term $k = 1$ dominates, which implies $\|A_{\theta+d\theta} - A_\theta\|_F^2 \leq \|d\theta\|^2 w^{(1-\gamma)(L-1)}$ which implies that $\|d\theta\| \geq \|A_{\theta+d\theta} - A_\theta\|_F w^{-\frac{(1-\gamma)(L-1)}{2}} = O(w^{-\frac{(1-\gamma)(L-1)}{2}})$, which decreases with width.

Let us now show upper bounds on $\|d\theta\|$. When $\gamma > 1$, we will construct a closeby minimum. Let us first define the parameters $\bar\theta = (\bar W_1, \dots, \bar W_L)$ where $\bar W_1 = 0$ and $\bar W_L = 0$ and $\bar W_{\ell,ij} = \begin{cases} W_{\ell,ij} & \text{if } i,j > \min\{n_0, n_L\} \\ 0 & \text{otherwise} \end{cases}$. Since we have set only $O(w)$ parameters to zero, we have $\left\|\theta - \bar\theta\right\|^2 = O(\sigma^2 w) = O(w^{1-\gamma})$. Now let the matrix A be a global minimum of the cost C with SVD $A = USV^T$ (with inner dimension equal to the rank k of A), we then set $\theta^* = \bar\theta + I^{(k\to w)}(S^{\frac{1}{L}}V^T, S^{\frac{1}{L}}, \dots, S^{\frac{1}{L}}, US^{\frac{1}{L}})$. The parameters θ^* are a global minimum since $A_{\theta^*} = A$ and $\|\theta^* - \theta\| \leq \left\|\theta^* - \bar\theta\right\| + \left\|\bar\theta - \theta\right\| = O(1) + O(w^{\frac{1-\gamma}{2}}) = O(1)$.

When $\gamma < 1$, with prob. $1-\epsilon$, we have $s_{min}\left(W_{L-1}\cdots W_1\right) > \frac{1}{2}\sigma^{(L-1)} w^{\frac{L-1}{2}} = w^{\frac{(1-\gamma)(L-1)}{2}}$, we can reach a global minimum by only changing W_L, we need $dW_L W_{L-1}\cdots W_1 = A^* - A_\theta$ hence we take $dW_L = (A^* - A_\theta)\left(W_L\cdots W_1\right)^+$ with norm $\|d\theta\| = \|dW_L\|_F \leq \frac{\|A^* - A_\theta\|}{s_{min}(W_{L-1}\cdots W_1)} = O(w^{-\frac{(1-\gamma)(L-1)}{2}})$.

(2) Distance to saddles: Given parameters $\theta = (W_1, \dots, W_L)$, we can obtain a saddle θ^* by setting all entries of W_1 and W_L to zero. We have

$$\mathbb{E}\left[\|\theta - \theta^*\|^2\right] = \mathbb{E}\left[\|W_1\|_F^2\right] + \mathbb{E}\left[\|W_L\|_F^2\right] = \sigma^2(n_0 + n_L)w = O(w^{1-\gamma}).$$

This gives an upper bound of order $w^{1-\gamma}$ on the distance between θ and the set of saddles θ^*.

Now let $\theta^* = \theta + d\theta$ be the saddle closest to θ, we know that

$$0 = \partial_{W_L} \mathcal{L}(\theta^*) = \nabla C(A_{\theta^*}) \left(W_1^*\right)^T \cdots \left(W_{L-1}^*\right)^T.$$

Since A_{θ^*} is not a global minimum, $\nabla C(A_{\theta^*}) \neq 0$, for the above to be zero, we therefore need $\left(W_1^*\right)^T \cdots \left(W_{L-1}^*\right)^T$ to not have full column rank, i.e. $\text{Rank}\left(W_1^*\right)^T \cdots \left(W_{L-1}^*\right)^T = n_0$.

We will show that at initialization $\left(W_1\right)^T \cdots \left(W_{L-1}\right)^T$ has rank n_0 and its smallest non-zero singular value s_{min} is of order $w^{\frac{(1-\gamma)(L-1)}{2}}$. We will use the fact that $\left\|\left(W_1\right)^T \cdots \left(W_{L-1}\right)^T - \left(W_1^*\right)^T \cdots \left(W_{L-1}^*\right)^T\right\|_F \geq s_{min}$ to lower bound the distance $\|\theta - \theta^*\|$ using Lemma I.2.

The singular values of $W_1^T \cdots W_{L-1}^T$ are the squared root of the eigenvalues of the $n_0 \times n_0$ matrix $W_1^T \cdots W_{L-1}^T W_{L-1} \cdots W_1$. One can show that as $w \to \infty$ this matrix concentrates in its expectation

$$\mathbb{E}\left[W_1^T \cdots W_{L-1}^T W_{L-1} \cdots W_1\right] = \sigma^{2(L-1)} w^{L-1} = w^{(1-\gamma)(L-1)}.$$

which implies that $s_{\min}$ concentrates in $w^{\frac{(1-\gamma)(L-1)}{2}}$ and therefore $s_{\min} \asymp w^{\frac{(1-\gamma)(L-1)}{2}}$.

Now by Lemma I.2 (applied to the depth $L-1$ this time), we have with prob. $1-\epsilon$

$$\begin{aligned} s_{\min}^2 &\leq \left\|\left(W_1\right)^T \cdots \left(W_{L-1}\right)^T - \left(W_1^*\right)^T \cdots \left(W_{L-1}^*\right)^T\right\|_F^2 \\ &\leq C_{\epsilon, L-1} \sum_{k=1}^{L-1} \|\theta - \theta'\|^{2k} w^{(1-\gamma)(L-1-k)} \end{aligned}$$

and $\|\theta - \theta'\|$ needs to be at least of order $w^{\frac{(1-\gamma)}{2}}$ for any of the terms in the sum to be at least of order $w^{(1-\gamma)(L-1)}$ (actually all these become of the right order at the same time). □

Spectrum bounds

An important tool in our analysis is the following Theorem (which is a reformulation of Corollary 5.35 in [217])

Theorem I.2. *Let A be a $m \times n$ matrix with i.i.d. $\mathcal{N}(0, \sigma^2)$ entries. For all $t \geq 0$, with probability at least $1 - 2e^{-\frac{t^2}{2}}$, it holds that*

$$\sigma(-\sqrt{m} - \sqrt{n} - t) \leq s_{min}(A) \leq s_{max}(A) \leq \sigma\left(\sqrt{m} + \sqrt{n} + t\right).$$

Corollary 3. *If the parameters θ are independent centered Gaussian with variance σ^2, for all $t \geq 0$, with probability at least $1 - 2Le^{-\frac{t^2}{2}}$, it holds that*

$$\|A_\theta\|_{op} \leq (1+t)^L \sigma^L \left(\sqrt{n_0} + \sqrt{w}\right) (4w)^{\frac{L-2}{2}} \left(\sqrt{w} + \sqrt{n_L}\right).$$

Proof. By Theorem I.2, with probability greater than $1 - 2Le^{-\frac{t^2}{2}}$, for all $\ell = 1, \ldots, L$, $\|W_\ell\|_{op} \leq \sigma\left(\sqrt{n_{\ell-1}} + \sqrt{n_\ell} + t\right)$, where $n_\ell = w$ for $\ell \in \{1, \cdots, L-1\}$. Hence

$$\|A_\theta\|_{op} \leq \|W_L\|_{op} \cdots \|W_1\|_{op} \leq \sigma^L \prod_{\ell=1}^{L} \left(\sqrt{n_{\ell-1}} + \sqrt{n_\ell} + t\right) \leq (1+t)^L \sigma^L \prod_{\ell=1}^{L} \left(\sqrt{n_{\ell-1}} + \sqrt{n_\ell}\right).$$

□

I.3 Proofs for the Saddle-to-Saddle regime

In this section, we prove Theorem 4 of the main. Given a saddle $\vartheta^* = RI^{(k\to w)}(\vartheta)$ where ϑ is a local minimum in a width k network, we want to describe the dynamics of gradient descent $\theta_\alpha(t) = \gamma(t, \vartheta^* + \alpha\theta_0)$, initialized close to ϑ^*. We shall consider $\vartheta^* = 0$ for convenience, though the same arguments could be applied for $\vartheta^* \neq 0$. We will start by studying the case of homogeneous costs, which will allow us to describe costs that locally look homogeneous around 0. Later on, after having defined the notion of *escape paths,* we will show that as $\alpha \to 0$, the path $(\theta_\alpha(t))_{t\in\mathbb{R}_+}$ converges to an escape path with specific direction and speed. We will then show that the escape paths which escape at this speed are unique in some aspects.

Homogeneous Costs

As in the main text, we use θ to denote an element in the parameter space $\mathbb{R}^P$. Let $k \geq 2$ be an integer. We say that a cost H is k-homogeneous if $H(\alpha\theta) = \alpha^k H(\theta)$ for all $\theta \in \mathbb{R}^P$ and all scalar $\alpha > 0$. Later in this paper, we will be particularly interested in the case where $H(\theta) = \mathrm{Tr}\left[GA_\theta\right]$ for a linear network A_θ of depth L and some $n_L \times n_0$ matrix G. Thus defined, H is a L-homogeneous polynomial.

Throughout, when studying a k-homogeneous cost H, we will always assume that it is twice differentiable.

A useful property of gradient descent on a homogeneous cost is that:

Lemma I.3. *Gradient flow on a twice-differentiable k-homogeneous cost H satisfies*

$$\gamma_H(t, \lambda\theta_0) = \lambda\gamma_H(\lambda^{k-2}t, \theta_0)$$

for all $\theta_0 \in \mathbb{R}^P$,all $\lambda > 0$ and all $t \geq 0$.

Proof. We simply need show that for all $\theta_0 \in \mathbb{R}^P, \lambda > 0, t \geq 0$, we have $\frac{1}{\lambda}\gamma_H(\lambda^{2-k}t, \lambda\theta_0) = \gamma_H(t, \theta_0)$, i.e. that the path $t \mapsto \frac{1}{\lambda}\gamma_H(\lambda^{2-k}t, \lambda\theta_0)$ is the solution of gradient descent starting at θ_0. Clearly, the path starts at θ_0 and satisfies

$$\partial_t \frac{1}{\lambda}\gamma_H(\lambda^{2-k}t, \lambda\theta_0) = -\lambda^{1-k}\nabla H\left(\gamma_H(\lambda^{2-k}t, \lambda\theta_0)\right) = -\nabla H\left(\frac{1}{\lambda}\gamma_H(\lambda^{2-k}t, \lambda\theta_0)\right)$$

since, using the fact that H is k-homogeneous, for all scalar $\alpha > 0$, and any $\theta \in \mathbb{R}^P$, $\alpha\nabla H(\alpha\theta) = \alpha^k\nabla H(\theta)$. One concludes using Picard-LindelÃ¶f Theorem, using that ∇H is locally Lipschitz around 0 since H is twice differentiable. □

An **Escape Direction** at 0 of H is a vector on the sphere $\rho \in \mathbb{S}^{P-1}$ such that $\nabla H(\rho) = -s\rho$ for some $s \in \mathbb{R}_+$ which we call the *escape speed* associated with ρ. A path $(\theta(t))_{t<0}$ indexed by negative times and following gradient flow on H such that $\theta(t)$ is on one escape direction for some $t < 0$ will remain along this direction (these paths are equal for $t < 0$ to $\theta(t) = \rho e^{st}$ when $k = 2$ and $\theta(t) = \rho\left(-(k-2)st\right)^{-\frac{1}{k-2}}$ when $k > 2$). Note that this entails that $\theta(t) \to 0$ as $t \to -\infty$. When $H(\theta) = \theta^T A\theta$ for some symmetric matrix A, the escape directions are simply the eigenvectors of the Hessian A and the escape speeds are twice the eigenvalues of A.

An **Optimal Escape Direction** $\rho^* \in \mathbb{S}^{P-1}$ is an escape direction with the largest speed $s^* > 0$. It is a minimizer of H restricted to $\mathbb{S}^{P-1}$:

$$\rho^* \in \arg\min_{\rho \in \mathbb{S}^{P-1}} H(\rho). \tag{I.3.1}$$

Indeed, critical points of $H(\rho)$ restricted to the sphere are the escape directions, and by Euler's condition (i.e. $\nabla H(\rho)^T \rho = kH(\rho)$ if H is k-homogeneous), if ρ is an escape direction with speed s, then $H(\rho) = -\frac{s}{k}$: optimal escape directions are thus global minimizers of H restricted on the unit sphere.

Under some conditions on the Hessian along the escape directions, one can guarantee that gradient descent will escape along an optimal escape path:

Proposition I.2. *Assume that the optimal escape speed s^* is positive and that for all escape directions ρ which are not optimal, there is a vector $v \perp \rho$ such that $v^T \mathcal{H}H(\rho) v < -s^* v^T v$. Let Ω be the set of θ_0 such that the direction $\frac{\gamma(t,\theta_0)}{\|\gamma(t,\theta_0)\|}$ of the gradient descent flow converges towards an optimal escape direction as $t \to T$, where T is the explosion time of the path (which can be infinite). The set $\mathbb{S}^{P-1} \setminus \Omega$ has spherical measure zero.*

Proof. Let $\Omega' \subset \mathbb{R}^P$ be the set of points $\theta \neq 0$ such that gradient flow on the 0-homogeneous cost $\bar{H}(\theta) = H\left(\frac{\theta}{\|\theta\|}\right)$ converges to a global minimum. Our proof is divided in two steps: (1) we show that $\Omega' \subset \Omega$, (2) we show that $\Omega' \cap \mathbb{S}^{P-1}$ has spherical measure 1.

(1) Note that both sets Ω and Ω' are cones: for any $\alpha > 0$, $\Omega = \alpha\Omega$ and $\Omega' = \alpha\Omega'$. Therefore, we only need to show that $\Omega' \cap \mathbb{S}^{P-1} \subset \Omega \cap \mathbb{S}^{P-1}$. Besides, note that since $\bar{H}$ is 0-homogenous, $\nabla \bar{H}(\theta)^T \theta = 0$ for all $\theta \in \mathbb{R}^P$ and thus, the norm is an invariant of the descent gradient flow for $\bar{H}$: for any $\theta_0 \in \mathbb{R}^P$, $t \mapsto \|\gamma_{\bar{H}}(t, \theta_0)\|$ is constant.

In particular, if $\theta_0 \in \Omega' \cap \mathbb{S}^{P-1}$, then $\bar{\theta}(t) = \gamma_{\bar{H}}(t, \theta_0)$ converges to an optimal escape direction ρ^* as $t \to \infty$, by Equation (I.3.1). The gradient flow path $\theta(t)$ can be obtained from the gradient flow path $\bar{\theta}(s)$ directly. First we define the function $\alpha(s) = e^{-k \int_0^s H(\bar{\theta}(r))dr}$ such that

$$\begin{aligned}\partial_s \left[\bar{\theta}(s)\alpha(s)\right] &= -(I - \bar{\theta}(s)\bar{\theta}(s)^T)\nabla H(\bar{\theta}(s))\alpha(s) - k\bar{\theta}(s)H(\bar{\theta}(s))\alpha(s) \\ &= -\nabla H(\bar{\theta}(s))\alpha(s)\end{aligned}$$

where we used the fact that $\theta^T \nabla H(\theta) = kH(\theta)$. Let us now define $\tau(t) = \int_0^t \alpha(s)^{k-2} ds$, we have

$$\bar{\theta}(\tau(t))\alpha(\tau(t)) = -\nabla H(\bar{\theta}(\tau(t)))\alpha(\tau(t))^{k-1} = -\nabla H\left(\bar{\theta}(\tau(t))\alpha(\tau(t))\right)$$

which implies that $\theta(t) = \bar{\theta}(\tau(t))\alpha(\tau(t))$. As $r \to \infty$, we have $H(\bar{\theta}(r)) \to -s^* < 0$, which implies that $\alpha(s) \to \infty$ as $s \to \infty$. This in turn implies that $\tau(t) \to \infty$ as $t \to \infty$. As a result, we obtain that

$$\lim_{t\to\infty} \frac{\theta(t)}{\|\theta(t)\|} = \lim_{t\to\infty} \frac{\bar{\theta}(\tau(t))\alpha(\tau(t))}{\left\|\bar{\theta}(\tau(t))\alpha(\tau(t))\right\|} = \lim_{t\to\infty} \bar{\theta}(\tau(t)) = \rho^*$$

and hence $\theta_0 \in \Omega$ as needed.

(2) We now show that $\Omega' \cap \mathbb{S}^{P-1}$ has spherical measure 1: this is a consequence of the fact that the critical points of $\bar{H}$ are global minima or strict saddle points. By taking the gradient of $\bar{H}$, one sees that the critical points of $\bar{H}$ on the sphere $\mathbb{S}^{P-1}$ are the points $\theta \in \mathbb{S}^{P-1}$ such that

$$\nabla H(\theta) = \theta\theta^T \nabla H(\theta).$$

Since $\theta\theta^T$ is the orthogonal projection on the line $\mathbb{R}.\theta$, the critical points of $\bar{H}$ on $\mathbb{S}^{P-1}$ are the escape directions. As explained before, global minima of $\overline{H}$ are optimal escape directions. The other escape directions are strict saddle points: consider such ρ and let $v \perp \rho$ be such that $v^T\mathcal{H}H(\rho)v < -s^*v^Tv$. Differentiating $\bar{H}$ twice and using that $v \perp \rho$, one can show that

$$v^T\mathcal{H}\bar{H}(\rho)v = v^T\mathcal{H}H(\rho)v - \nabla H(\rho)^T\rho v^Tv.$$

Since ρ is an escape direction, $\nabla H(\rho) = -s\rho$ with $s < s^*$ and $\|\rho\| = 1$: this implies $v^T\mathcal{H}\bar{H}(\rho)v < (s - s^*)\, v^Tv < 0$. In particular, the points such that the gradient descent on $\bar{H}$ converge to a saddle have spherical measure 0 on $\mathbb{S}^{P-1}$. This shows that $\Omega' \cap \mathbb{S}^{P-1}$ has spherical measure 1, and allows us to conclude. □

Deep Linear Networks

For a depth L DLN and the homogeneous cost $H(\theta) = \mathrm{Tr}\left[G^TA_\theta\right]$ with SVD decomposition $G = USV^T$, the escape directions ρ are of the form

$$\frac{1}{\sqrt{L}}(\pm u_i w_{L-1}^T, w_{L-1}w_{L-2}^T, \dots, w_1 v_i^T)$$

with speed $s = \mp\frac{s_i}{L^{\frac{L-2}{2}}}$, where u_i, v_i are the i-th columns of U, V respectively. The optimal speed is $\frac{s_1}{L^{\frac{L-2}{2}}}$, where s_1 is the largest singular value of G.

Furthermore this loss satisfies the property required to ensure convergence along the fastest escape path:

Lemma I.4. *For a network of depth L and width $w \geq 1$, for any escape direction of the form $\rho = \frac{1}{\sqrt{L}}(\pm u_i w_{L-1}^T, w_{L-1}w_{L-2}^T, \dots, w_1 v_i^T)$ with speed $\mp\frac{s_i}{L^{\frac{L-2}{2}}} < \frac{s_1}{L^{\frac{L-2}{2}}}$ the vector $v = (-u_1 w_{L-1}^T, 0, \dots, 0, w_1 v_1^T)$ satisfies*

$$v^T\mathcal{H}H(\rho)v < \mp\frac{s_i}{L^{\frac{L-2}{2}}}v^Tv.$$

Proof. We have $v^T\mathcal{H}H(\rho)v = -\frac{2s_1}{L^{\frac{L-2}{2}}}$ and $\pm\frac{s_i}{L^{\frac{L-2}{2}}}v^Tv = \pm\frac{2s_i}{L^{\frac{L-2}{2}}}$ as needed. □

This guarantees that gradient flow will not escape along a non-optimal direction, but it does not rule out the possibility that it converges to a saddle of the loss $H(\theta) = \mathrm{Tr}\left[G^TA_\theta\right]$. Each non-zero saddle $\theta^* = (W_1, \dots, W_L)$ is technically proportional to an escape direction ρ with escape speed 0, since $\nabla H(\theta^*) = 0$. For shallow networks these saddles are strict [115] and so they are almost surely avoided, guaranteeing convergence in direction. For depth $L = 3$ we can apply Proposition I.2 since we have:

Lemma I.5. *Consider the cost $H(\theta) = \mathrm{Tr}\left[GA_\theta\right]$ for a rank $\min\{n_0, n_L\}$ matrix G and a network of depth $L = 3$ and width $w \geq 1$. For any escape direction ρ with speed 0 there is a vector v such that $v^T\mathcal{H}H(\rho)v < 0$.*

Proof. Since $\rho \neq 0$ there must a non-zero W_1,W_2 or W_3. We separate the case $W_2 \neq 0$ from W_1 or W_3 is non-zero.

Case $W_2 \neq 0$: let u_1, v_1 be the largest singular vectors of G and $\tilde{u}_1, \tilde{v}_1$ the largest singular vectors of W_2, then $v = (-\tilde{u}_1 v_1^T, 0, u_1\tilde{v}_1^T)$ satisfies

$$v^T\mathcal{H}H(\rho)v = -\mathrm{Tr}\left[Gu_1\tilde{v}_1^TW_2\tilde{u}_1v_1^T\right] = -s_1\tilde{s}_1 < 0.$$

Case $W_1 \neq 0$ (the case $W_3 \neq 0$ is similar): Let u_1, v_1 be the largest singular vectors of $W_1 G$ and b be any unitary w-dim vector, then the parameters $v = (0, bv_1^T, u_1 b^T)$ satisfy

$$v^T \mathcal{H} H(\rho) v = -\mathrm{Tr}\left[G u_1 b^T b v_1^T W_1\right] = -s_1 < 0.$$

□

For $L > 3$ we were not able to prove that the saddles can be avoided with prob. 1, we therefore introduce the assumption:

Assumption A. *Let I be the set of initializations which converge to a saddle of the cost $H(\theta) = \mathrm{Tr}\left[GA_\theta\right]$. We shall work on the event $E = \theta_0 \notin I$.*

It can easily be proven for a Gaussian initialization that $\mathbb{P}(E) \geq 1/2$, i.e. that saddles can be avoided with probability at least $1/2$, since $P\left(H(\theta_0) < 0\right) = \frac{1}{2}$ at initialization (this follows from the fact that $H((W_1, \ldots, W_L)) = -H((-W_1, \ldots, W_L))$).

Another motivation for this assumption is the fact that if the network is initialized with balanced weights [4, 5], i.e. if $W_\ell^T W_\ell = W_{\ell-1} W_{\ell-1}^T$ for $1 < \ell < L$, then necessarily $\theta_0 \notin I$. This is because the balancedness is conserved during training: if gradient flow converges to a saddle, this saddle must be balanced. However the only balanced saddle of H is the origin, which can only be approached along an escape direction ρ with positive speed $s > 0$, which are avoided with prob. 1 by Proposition I.2.

Approximately Homogeneous Costs

In the previous section, we studied the escape paths for homogeneous costs H. We extend these results to more general cost functions, which are only locally homogeneous around a saddle ϑ^*, i.e. we consider costs of the form

$$C(\theta) = H(\theta - \vartheta^*) + e(\theta - \vartheta^*), \tag{I.3.2}$$

where H is a k-homogeneous cost H, and where e is infinitely differentiable such that its $m-1$ first derivatives vanish at 0 for a given $m > k$. We call such costs (k, m)-*approximately homogeneous*. In the setting of a cost $C(A_\theta)$ for a neural network of depth L, the saddle at the origin $\theta = 0$ is $(L, 2L)$-approximately homogeneous, since the only non-vanishing derivatives are the kL-th derivatives for $k = 0, 1, 2, \ldots$.

Since we are only interested in the local behaviour around the saddle ϑ^*, we localize the cost: let $h : \mathbb{R}_+ \to \mathbb{R}_+$ be a smooth cut-off function such that $h(x) = 1$ if $0 \leq x \leq 1$, $0 \leq h(x) \leq 1$ if $1 < x < 2$ and $h(x) = 0$ when $x \geq 2$. For $r > 0$, we define the localization C_r of the cost C as

$$C_r(\theta) = H(\theta - \vartheta^*) + e(\theta - \vartheta^*) h\left(\frac{\|\theta - \vartheta^*\|}{r}\right). \tag{I.3.3}$$

As usual, we assume for simplicity that $\vartheta^* = 0$. We note for later use that by assumption on e, for all compact set K containing 0, there exists a finite constant $c > 0$ such that

$$||\nabla e(\theta)|| \leq c||\theta||^k, \forall \theta \in K. \tag{I.3.4}$$

Lemma I.6. *Let $h : \mathbb{R}_+ \to \mathbb{R}$ and $e : \mathbb{R}^P \to \mathbb{R}$ be as above. The correction $e_r(\theta) = e(\theta) h\left(\frac{1}{r}\|\theta\|\right)$ satisfies*

$$\left\|\partial_\theta^k [e_r]\right\|_\infty = O\left(r^{m-k}\right), \quad \text{as } r \to 0.$$

Proof. We have

$$\partial_\theta^k \left[e(\theta) h\left(\frac{1}{r}\|\theta\|\right)\right] = \sum_{k_1+k_2=k} \partial_\theta^{k_1} e(\theta) \partial_\theta^{k_2} h\left(\frac{1}{r}\|\theta\|\right)$$

Since $\partial_\theta^{k_2} h\left(\frac{1}{r}\|\theta\|\right) = 0$ whenever $\|\theta\| > 2r$ and $\left\|\partial_\theta^{k_2} h\left(\frac{1}{r}\|\theta\|\right)\right\|_\infty = O\left(r^{-k_2}\right)$, while $\left\|\partial_\theta^{k_1} e(\theta)\right\| = O(\|\theta\|^{m-k_1})$ we see from the above equation that

$$\left\|\partial_\theta^k \left[e(\theta) h\left(\frac{1}{r}\|\theta\|\right)\right]\right\|_\infty = O(r^{m-k_1} r^{-k_2}) = O(r^{m-k}),$$

as claimed. □

Escape Cones

We will approximate approximately homogeneous costs by homogeneous ones using the following approximation:

Lemma I.7. *Suppose that $C(\theta) = H(\theta) + e(\theta)$ is (k,m)-approximately homogeneous around 0 as defined in I.3.2. Let $\theta_0 \in \mathbb{R}^P$. It holds that for all $t \geq 0$ and all $\alpha > 0$, there is a finite constant $c_1(t)$ that does not depend on α such that*

$$\left\|\gamma_C(\alpha^{2-k}t, \alpha\theta_0) - \gamma_H(\alpha^{2-k}t, \alpha\theta_0)\right\| \leq c_1(t)\alpha^2.$$

Proof. Fix $\theta_0 \in \mathbb{R}^P$ and $t \geq 0$ and let $d_t = d(t,\theta_0) := \sup_{s\leq t} \gamma_C(s,\theta_0), \gamma_H(s,\theta_0)$. We can bound how fast the distance between the two paths γ_C and γ_H increases as follows:

$$\begin{aligned}
\partial_t \|\gamma_C(t,\theta_0) - \gamma_H(t,\theta_0)\| &= -\frac{(\gamma_C(t,\theta_0) - \gamma_H(t,\theta_0))^T}{\|\gamma_C(t,\theta_0) - \gamma_H(t,\theta_0)\|} \left(\nabla H\left(\gamma_C(t,\theta_0)\right) + \nabla e\left(\gamma_C(t,\theta_0)\right) - \nabla H\left(\gamma_H(t,\theta_0)\right)\right) \\
&\leq \left(\sup_{\|\theta\|\leq d_t} \|\mathcal{H}H(\theta)\|_{op}\right) \|\gamma_C(t,\theta_0) - \gamma_H(t,\theta_0)\| + \|\nabla e\left(\gamma_C(t,\theta_0)\right)\| \\
&\leq c' d_t^{k-2} \|\gamma_C(t,\theta_0) - \gamma_H(t,\theta_0)\| + c d_t^k
\end{aligned}$$

where c comes from I.3.4 and $c' = \sup_{\|x\|\leq 1} \|\mathcal{H}H(x)\|_{op}$. Applying GrÃ¶nwall's inequality on $[0,t]$ to $A(s) = \|\gamma_C(s,\theta_0) - \gamma_H(s,\theta_0)\| + \frac{c}{c'} d_t^2$ (such that $\partial_s A(s) \leq c' d_t^{k-2} A(s)$), we obtain

$$\|\gamma_C(s,\theta_0) - \gamma_H(s,\theta_0)\| + \frac{c}{c'} d_t^2 = A(s) \leq A(0) e^{c' d_t^{k-2} s} = \frac{c}{c'} d_t^2 e^{c' d_t^{k-2} s}.$$

Hence $\|\gamma_C(t,\theta_0) - \gamma_H(t,\theta_0)\| \leq \frac{c}{c'} d_t^2 e^{c' d_t^{k-2} t}$ for all times $t \geq 0$. To finish the proof, one uses that for a fixed $t \geq 0$, $d(t, \alpha\theta_0) = O(\alpha)$ as $\alpha \to 0$, which is true because 0 is a saddle of C and H so their gradient tends to 0 with α. □

We define the ϵ-**Escape Cone** as the set $\mathcal{C}_\epsilon = \left\{\theta \in \mathbb{R}^P : \frac{H(\theta)}{\|\theta\|^k} < \frac{-s^* + \epsilon}{k}\right\}$ where we recall that s^* denotes the optimal escape speed.

Proposition I.3. *For all $\epsilon > 0$ small enough there is a $r > 0$ such that*

1. *for any $\theta \in \partial\mathcal{C}_\epsilon$ with $\|\theta\| < r$, the negative of the gradient of C at θ points inside the cone, i.e. denoting by n the normal of $\mathcal{C}_\epsilon$ at θ pointing inside of the cone, we have $-\nabla C(\theta)^T n \geq 0$.*
2. *for any point θ inside the cone with $\|\theta\| < r$, we have $\|\theta\|^{k-1}(-s^* - \epsilon) \leq \nabla C(\theta)^T \frac{\theta}{\|\theta\|} \leq \|\theta\|^{k-1}(-s^* + 2\epsilon)$.*
3. *Let $\theta_0 \in \mathcal{C}_\epsilon$ and $\|\theta_0\| < r$, let T be the time when $\|\gamma_C(t, \theta_0)\| = r$. When $k = 2$ we have for all time $0 \leq t < T$*
$$\|\theta_0\| e^{-(-s^*+2\epsilon)t} \leq \|\gamma(t, \theta_0)\| \leq \|\theta_0\| e^{-(-s^*-\epsilon)t}$$
and when $k \neq 2$ we have for all time $0 \leq t < T$
$$\left[\|\theta_0\|^{-(k-2)} + (k-2)(-s^* + 2\epsilon)t\right]^{-\frac{1}{k-2}} \leq \|\gamma(t, \theta_0)\| \leq \left[\|\theta_0\|^{-(k-2)} + (k-2)(-s^* - \epsilon)t\right]^{-\frac{1}{k-2}}.$$

Proof. For all non-zero $\theta \in \mathbb{R}^P$, define $P_\theta = \left[I_d - \frac{\theta\theta^T}{\|\theta\|^2}\right]$, which is the orthogonal projection to the tangent space of $\mathbb{S}^{P-1}$ at $\frac{\theta}{\|\theta\|}$. Denote by $\partial\mathcal{C}_\epsilon$ the boundary of the cone and note that for any $\theta \in \partial\mathcal{C}_\epsilon$, it holds that $H(\theta) = (-s^* + \epsilon)/k$. Choose

$$r = r(\epsilon) = \min\left\{ \frac{\inf_{\rho \in \mathbb{S}^{P-1} \cap \partial\mathcal{C}_\epsilon} \left\{\nabla H(\rho)^T P_\rho \nabla H(\rho)\right\}}{c \sup_{\rho \in \mathbb{S}^{P-1} \cap \partial\mathcal{C}_\epsilon} \left\{\sqrt{\nabla H(\rho)^T P_\rho \nabla H(\rho)}\right\}}, \sqrt{\frac{\epsilon}{c}} \right\},$$

where the constant c comes from I.3,4.

(1) Let $\theta \in \partial\mathcal{C}_\epsilon$, so that $\frac{H(\theta)}{\|\theta\|^k} = H\left(\frac{\theta}{\|\theta\|}\right) = (-s^* + \epsilon)/k$. The normal pointing inside the cone is equal (up to a positive scaling) to

$$-\nabla_\theta\left(H\left(\frac{\theta}{\|\theta\|}\right)\right) = -\nabla H\left(\frac{\theta}{\|\theta\|}\right)\frac{P_\theta}{\|\theta\|}.$$

We then have that

$$\begin{aligned}
\left(-\nabla_\theta\left(\frac{H(\theta)}{\|\theta\|^k}\right)\right)^T (-\nabla C(\theta)) &= -\nabla H\left(\frac{\theta}{\|\theta\|}\right)^T \frac{P_\theta}{\|\theta\|}(-\nabla H(\theta) - \nabla e(\theta)) \\
&= \|\theta\|^{k-1} \nabla H\left(\frac{\theta}{\|\theta\|}\right)^T P_\theta \nabla H\left(\frac{\theta}{\|\theta\|}\right) + \nabla H\left(\frac{\theta}{\|\theta\|}\right)^T \frac{P_\theta}{\|\theta\|}\nabla e(\theta) \\
&\geq \|\theta\|^{k-1} \inf_{\rho \in \mathbb{S}^{P-1} \cap \partial\mathcal{C}_\epsilon} \left\{\nabla H(\rho)^T P_\rho \nabla H(\rho)\right\} \\
&\quad - \frac{1}{\|\theta\|} \sup_{\rho \in \mathbb{S}^{P-1} \cap \partial\mathcal{C}_\epsilon} \sqrt{\nabla H(\rho)^T P_\rho \nabla H(\rho)}\, \|\nabla e(\theta)\| \\
&\geq \|\theta\|^{k-1} \inf_{\rho \in \mathbb{S}^{P-1} \cap \partial\mathcal{C}_\epsilon} \left\{\nabla H(\rho)^T P_\rho \nabla H(\rho)\right\} \\
&\quad - c\,\|\theta\|^k \sup_{\rho \in \mathbb{S}^{P-1} \cap \partial\mathcal{C}_\epsilon} \left\{\sqrt{\nabla H(\rho)^T P_\rho \nabla H(\rho)}\right\},
\end{aligned}$$

where we used I.3.4 for the last inequality. The right-hand side above is positive since $\|\theta\| < r(\epsilon) \leq \frac{\inf_{H(\rho)=s^*+\epsilon}\left\{\nabla H(\rho)^T P_\rho \nabla H(\rho)\right\}}{c \sup_{H(\rho)=s^*+\epsilon}\left\{\sqrt{\nabla H(\rho)^T P_\rho \nabla H(\rho)}\right\}}$.

(2) Let $\theta \in \mathcal{C}_\epsilon$. By I.3.4, we have that

$$\begin{aligned}
\nabla C(\theta)^T \theta &= \partial_\lambda H(\lambda\theta)\Big|_{\lambda=1} + (\nabla e(\theta))^T \theta \\
&\leq kH(\theta) + c\|\theta\|^{k+2} \\
&= k\|\theta\|^k H\left(\frac{\theta}{\|\theta\|}\right) + c\|\theta\|^{k+2} \\
&\leq \|\theta\|^k \left(-s^* + \epsilon + c\|\theta\|^2\right) \\
&\leq \|\theta\|^k (-s^* + 2\epsilon)
\end{aligned}$$

where we used that $\|\theta\|^2 < r^2 \leq \frac{\epsilon}{c}$. In the other direction we obtain

$$\begin{aligned}
\nabla C(\theta)^T \theta &= \partial_\lambda H(\lambda\theta)\Big|_{\lambda=1} + (\nabla e(\theta))^T \theta \\
&\geq kH(\theta) - c\|\theta\|^{k+2} \\
&= k\|\theta\|^k H\left(\frac{\theta}{\|\theta\|}\right) - c\|\theta\|^{k+2} \\
&\geq \|\theta\|^k \left(-s^* - c\|\theta\|^2\right) \\
&\geq \|\theta\|^k (-s^* - \epsilon).
\end{aligned}$$

(3) Applying GrÃ¶nwall's inequality generalized to polynomial bounds (Lemma I.12), we have

$$\partial_t \|\theta\|^2 = -2\nabla C(\theta)^T \theta \leq c_1 \left(\|\theta\|^2\right)^{\frac{k}{2}}.$$

□

Putting it all together, this guarantees that with probability 1 over the initialization, gradient flow escapes the saddle at a specific speed along a path θ^1:

Proposition I.4. *Let $\theta_\alpha(t) = \gamma_C(t, \alpha\theta_0)$ for all $t \geq 0$. With prob. 1 over initialization (and under Assumption A when $L > 3$) there is a time horizon t_α^1 that tends to ∞ as $\alpha \to 0$ and a path $(\theta^1(t))_{t\in\mathbb{R}}$ such that for all $t \in \mathbb{R}$, $\lim_{\alpha\to 0} \theta_\alpha(t_\alpha^1 + t) = \theta^1(t)$. Furthermore, for all $\epsilon > 0$ s.t. $\epsilon < s^*/2$, there exists $T \in \mathbb{R}_+$ such that:*

(1) Shallow networks: $e^{(s^*-2\epsilon)(T+t)} \leq \left\|\theta^1(t)\right\| \leq e^{(s^*+\epsilon)(T+t)}$ for all $t \in \mathbb{R}$.

(2) Deep networks: $[(L-2)(s^*-2\epsilon)(T-t)]^{-\frac{1}{L-2}} \leq \left\|\theta^1(t)\right\| \leq [(L-2)(s^*+\epsilon)(T-t)]^{-\frac{1}{L-2}}$ for all $t < T$ (the path θ^1 is defined up to time T in this case).

Proof. We consider the gradient flow path $\tilde{\theta}_\alpha(t) = \gamma_H(t, \alpha\theta_0)$ on the k-homogeneous cost H. With prob. 1 (and under Assumption A when $L > 3$), we have $\frac{H(\tilde{\theta}_\alpha(t))}{\left\|\tilde{\theta}_\alpha(t)\right\|^L} \to -\frac{s^*}{k}$ as $t \to \infty$. In particular,

for all $\epsilon > 0$, there exists a finite $t \in \mathbb{R}$ such that $\tilde{\theta}_{\alpha=1} \in \mathcal{C}_\epsilon$ and more generally, by Lemma I.3, we have $\tilde{\theta}_\alpha(\alpha^{-(L-2)}t) = \alpha\tilde{\theta}_1(t) \in \mathcal{C}_\epsilon$. Lemma I.7 then shows that there exists $\alpha_0 > 0$ such that for all $\alpha < \alpha_0$, it holds that $\theta_\alpha(\alpha^{-(L-2)}t) \in \mathcal{C}_\epsilon$. Define $t_0 := \inf\left\{t \in \mathbb{R} : \tilde{\theta}_{\alpha=1}(t_0) \in \mathcal{C}_\epsilon\right\} < +\infty$.

By Proposition I.3, once the gradient flow path is inside $\mathcal{C}_\epsilon$, it cannot leave the escape cone until the norm $\|\theta_\alpha(t)\|$ is larger than some radius r. We define the time horizon $t_\alpha^1 = \inf\left\{t \in \mathbb{R} : \|\theta_\alpha(t)\| = \frac{r}{2}\right\}$ and the escape path θ^1 as the limit $\theta^1(t) = \lim_{\alpha\to 0}\theta_\alpha(t_\alpha^1 + t)$ for $t \in \mathbb{R}$ (the limit is well defined by continuity of $\theta \mapsto \gamma_C(t,\theta)$ and is an escape path by continuity of $\theta \mapsto \nabla\gamma_C(t,\theta)$). One can see that for any $t < 0$, there exists $\alpha > 0$ small enough such that $t_\alpha + t > \alpha^{-(L-2)}t_0$, thus it holds that $\theta^1(t) \in \mathcal{C}_\epsilon$ since for a small enough α, we have $\theta(t_\alpha^1 + t) \in \mathcal{C}_\epsilon$. Proposition I.3 then implies the escape rates for deep and shallow networks. □

Optimal Escape Paths

In this section, we define the notions of escape paths, optimal escape paths and we give a description of the optimal escape paths at the origin.

Proposition I.4 shows that as $\alpha \searrow 0$ one has convergence to an escape path which escapes with an almost optimal speed $s^* - 2\epsilon$ for a small $\epsilon > 0$. We will show that the only such escape paths are the optimal escape paths, i.e. those that escape exactly at a speed of s^*, furthermore these escape paths are unique up to rotations of the network.

We understand well the escape paths of the homogeneous loss H, and want to use this knowledge to describe the escape paths of the locally homogogeneous loss C. We will show a bijection between the escape paths of H and those of C such that their speed is preserved, but only between the set of escape paths which escape faster than a certain speed. It seems that in general there is no speed-preserving bijection between escape paths, indeed while for shallow networks (when the saddle is strict) one may apply the Hartman-Grobman Theorem to obtain a bijection, it does not preserve speed (since the bijection is in general not differentiable, only HÃ¶lder continuous).

This bijection is described by the following theorem (which is a more general version of Theorem 5 [] from the main - one simply needs to set $k = L$ and $m = 2L$ and $s^* = L\|H\|_\infty$ to recover theorem 5, i.e. the DLN case):

Theorem I.3 (Theorem 9.4 of the main text)**.** *Let $C = H + e$ be a (k,m)-approximately homogeneous loss, where H is a polynomial.*

***When** $k = 2$**:** for all s_0 s.t. $s_0 > \frac{k\|H\|_\infty}{m-1}$ there is a unique bijection $\Psi : \mathcal{F}_H(s_0) \to \mathcal{F}_C(s_0)$ such that for all paths $x \in \mathcal{F}_C(s_0)$, we have $\|x(t) - \Psi(x)(t)\| = O(e^{(m-1)s_0 t})$ as $t \to -\infty$.*

***When** $k > 2$**:** for all $s_0 > \frac{k-1}{m-k+1}k\|H\|_\infty$ there is a unique bijection $\Psi : \mathcal{F}_H(s_0) \to \mathcal{F}_C(s_0)$ such that for all paths $x \in \mathcal{F}_C(s_0)$, we have $\|x(t) - \Psi(x)(t)\| = O((-t)^{-\frac{m-k+1}{k-2}})$ as $t \to -\infty$.*

Note that in the case $s_0 > k\|H\|_\infty$, the set $\mathcal{F}_H(s_0)$ is empty (and therefore so is $\mathcal{F}_C(s_0)$).

Proof. For $r > 0$, recall that C_r denotes the localization of the cost C as introduced in Section I.3. It is readily seen that for all $r > 0$, there is a bijection Ψ_r between $\mathcal{F}_C[s_0]$ and $\mathcal{F}_{C_r}[s_0]$ such that for all $x_0 \in \mathcal{F}_C[s_0]$, $\|x_0(t) - \Psi_r(x_0)(t)\| = O(0)$ (i.e. the difference is zero for small enough $t < 0$). We therefore only need to show a bijection between $\mathcal{F}_{C_r}[s_0]$ and $\mathcal{F}_H[s_0]$.

Consider a fast escaping path $x_0 \in \mathcal{F}_H(s_0)$ of the homogeneous approximation of the loss. The escape paths of the origin w.r.t. to gradient flow on the cost C_r are fixed points of the following

map:

$$\Phi_{C_r} : x_0 \mapsto \left(t \mapsto \int_{-\infty}^{t} -\nabla C_r(x_0(u))du \right).$$

Our strategy is simply to iterate this map starting from the path x_0 to find such a fixed point (note that any fixed point of Φ_{C_r} is differentiable by the fundamental theorem of calculus). We will show that this iteration converges to a gradient flow path x_0' of the cost C_r which is, as $t \to -\infty$, $O(e^{(m-1)s_0 t})$-close to x_0 when $k = 2$ and $O((-t)^{\frac{k-m-1}{k-2}})$-close to x_0 when $k > 2$.

For $c > 0$, let B_c be the *set of corrections*, that is the set of all paths $b : \mathbb{R}_- \to \mathbb{R}^P$ (which are Lebesgue measurable functions) such that when $k = 2$, $\|b(t)\| \leq ce^{(m-1)s_0 t}$ for all $t \leq 0$ and when $k > 2$, $\|b(t)\| \leq c(-t)^{\frac{k-m-1}{k-2}}$ for all $t \leq 0$.

The convergence of the iteration process follows from the fact that Φ is a contraction w.r.t. to some norm on the set of paths $x_0 + B_c$ (the set of possible corrections around x_0). Indeed, Lemma I.9 (case $k = 2$, stated and proven in Section I.3) and Lemma I.11 (case $k > 2$, stated and proven in Section I.3) show that for all $x_0 \in \mathcal{F}_H[s_0]$, there exist $r > 0$ small enough and $c > 0$ large enough, such that Φ_{C_r} is a contraction on $x_0 + B_c$ for some well-suited norm (defined below), hence guaranteeing the existence and uniqueness of a fixpoint x_0' of Φ_{C_r} (which is obtained by interating Φ_{C_r} infinitely many times). We thus define the map $\Psi : x_0 \mapsto \Psi(x_0) = x_0'$.

We need to show that Ψ has an inverse that maps a fast escaping path $y_0 \in \mathcal{F}_C(s_0)$ back to a path $\Psi^{-1}(y_0) \in \mathcal{F}_H(s_0)$. We iterate the map

$$\Phi_H : y_0 \mapsto \left(t \mapsto \int_{-\infty}^{t} -\nabla H(y_0(u))du \right)$$

whose fixed points are the escape paths w.r.t. to gradient flow on the cost H. By a similar argument we can show that this map is a contraction on $x_0 + B_c$. Choosing $y_0 = x_0'$, this again implies the existence of a unique path $\Psi^{-1}(x_0')$ which is $O(e^{(m-1)s_0 t})$-close to x_0' when $k = 2$ and $O((-t)^{-\frac{m-k+1}{k-2}})$-close to x_0' when $k > 2$. Because $x_0' \in x_0 + B_c$ The uniqueness implies that since $x_0' = \Psi(x_0)$, the path $\Psi^{-1}(x_0')$ must be x_0. This shows that Ψ is a bijection and it is the only bijection between fast escaping paths with the property of mapping a path to a closeby path as in the statement of the theorem. □

Shallow networks

For the case $k = 2$, we consider the following norm for $\beta < (m-1)s_0$

$$\|b\|_\beta^2 = \int_{-\infty}^{0} e^{-2\beta t} \|b(t)\|^2 dt$$

defined on the corrections $b \in B_c$, where the set of corrections B_c is the set of all paths $b : \mathbb{R}_- \to \mathbb{R}^P$ such that $\|b(t)\| \leq ce^{(m-1)s_0 t}$ for all $t \leq 0$. The condition $\beta < (m-1)s_0$ ensures that

$$\|b\|_\beta^2 = \int_{-\infty}^{0} e^{-2\beta t} \|b(t)\|^2 dt \leq c^2 \int_{-\infty}^{0} e^{2((m-1)s_0 - \beta)t} dt < \infty.$$

As a result the set $x_0 + B_c$ equipped with the distance induced by the norm $\|\cdot\|_\beta$ is a complete metric space.

We define the scalar product for two corrections $x, y \in B$

$$\langle x, y \rangle_\beta = \int_{-\infty}^{\infty} e^{-2\beta t} \langle x(t), y(t) \rangle \, dt.$$

We first state a few useful properties of $\langle \cdot, \cdot \rangle_\beta$. In the following, $\dot{x}$ is the path obtained by considering the derivative of x.

Lemma I.8. *For any two corrections $x, y \in B$, we have*

1. $\langle x, \dot{y} \rangle_\beta = 2\beta \langle x, y \rangle_\beta - \langle \dot{x}, y \rangle_\beta$.
2. $\langle x, \dot{x} \rangle_\beta = \beta \|x\|_\beta^2$.
3. $\|x\|_\beta \leq \frac{1}{\beta} \|\dot{x}\|_\beta$.

Proof. The first point is obtained by integration by part:

$$\begin{aligned}\langle x, \dot{y} \rangle_\beta = \int_{-\infty}^{\infty} e^{-2\beta t} (x(t))^T \dot{y}(t) dt &= \int_{-\infty}^{\infty} 2\beta e^{-2\beta t} (x(t))^T y(t) dt - \int_{-\infty}^{\infty} e^{-2\beta t} \dot{x}(t) y(t) dt \\ &= 2\beta \langle x, y \rangle_\beta - \langle \dot{x}, y \rangle_\beta .\end{aligned}$$

The second point is a consequence of the first one, by taking $x = y$. Finally, the last point follows from the second one since $\|x\|_\beta^2 = \frac{1}{\beta} \langle x, \dot{x} \rangle_\beta \leq \frac{1}{\beta} \|x\|_\beta \|\dot{x}\|_\beta$, by Cauchy-Schwarz Inequality. □

We may now describe how for large enough β, one can guarantee that the map Φ_{C_r} is a contraction on the set $x_0 + B_c$:

Lemma I.9. *Let $C_r = H + e_r$ be a localized $(2, m)$-approximately homogeneous loss as in I.3.3, where H is a polynomial. Choose a $s_0 > \frac{2\|H\|_\infty}{m-1}$. There is a r small enough such that for any $x_0 \in \mathcal{F}_H[s_0]$ there is a constant c such that the map Φ_{C_r} is contraction on the set $x_0 + B_c = \left\{x_0 + b : \|b(t)\| \leq c e^{s_0(m-1)t}, \forall t < 0\right\}$ w.r.t. the norm on paths $\|\cdot\|_\beta$ for some β.*

Proof. We first show that for $r > 0$ small enough and $c > 0$ large enough, the image of $x_0 + B_c$ under Φ_{C_r} is contained in itself and then show that $\Phi_{C_r}$is a contraction w.r.t. the norm $\|\cdot\|_\beta$ for an adequate β.

(1) **Self-map:** Let $x \in x_0 + B_c$, i.e. $x = x_0 + b$ for some $b \in B_c$, then using the linearity of ∇H and the fact x_0 is a gradient flow path of H, we obtain

$$\begin{aligned}\Phi_{C_r}(x)(t) &= -\int_{-\infty}^{t} \nabla H(x(u)) + \nabla e_r(x(u)) du \\ &= -\int_{-\infty}^{t} \nabla H(x_0(u)) + \nabla H(b(u)) + \nabla e_r(x(u)) du \\ &= x_0(t) - \int_{-\infty}^{t} \nabla H(b(u)) + \nabla e_r(x(u)) du.\end{aligned}$$

Writing $b'(t) = \int_{-\infty}^{t} [\nabla H(b(u)) + \nabla e_r(x(u))]\, du$ we need to show that $b' \in B_c$. We can bound $\|b'(t)\|$ by

$$\|b'(t)\| \leq \int_{-\infty}^{t} (\|\nabla H(b(u))\| + \|\nabla e_r(x_0(u))\| + \|\nabla e_r(x(u)) - \nabla e_r(x_0(u))\|)\, du.$$

Using the fact that for a map g with uniformly bounded Hessian $\|\mathcal{H}g\|_\infty < \infty$, we have $\|\nabla g(x) - \nabla g(y)\| \leq \|\mathcal{H}g\|_\infty \|x - y\|$, it follows that $\|\nabla H(b(u))\| \leq \sup_{z\in\mathbb{R}^P} \|\mathcal{H}H(z)\|_{\text{op}} \|b(u)\|$ and $\|\nabla e_r(x(u)) - \nabla e_r(x_0(u))\| \leq \sup_{z\in\mathbb{R}^P} \|\mathcal{H}e_r(z)\|_{\text{op}} \|b(u)\|$. The last term $\|\nabla e_r(x_0(u))\|$ can be bounded by $\frac{\sup_z \|\partial_z^m e_r(z)\|_{\text{op}} \|x_0(u)\|^{m-1}}{(m-1)!}$ since the first $m-1$ derivatives of e_r vanish at 0 (see point 1 of Lemma I.13).

We therefore get

$$\|b'(t)\| \leq \int_{-\infty}^{t} \left(\left(\sup_{z\in\mathbb{R}^P} \|\mathcal{H}H(z)\|_{\text{op}} + \sup_{z\in\mathbb{R}^P} \|\mathcal{H}e_r(z)\|_{\text{op}} \right) \|b(u)\| + \sup_{z\in\mathbb{R}^P} \|\partial_z^m e_r(z)\|_{\text{op}} \|x_0(u)\|^{m-1} \right) du.$$

Since $\sup_{z\in\mathbb{R}^P} \|\mathcal{H}H(z)\|_{\text{op}} = 2\|H\|_{\text{op}}$ (where $\|\mathcal{H}H(x)\|_{\text{op}}$ is the operator norm of the Hessian $\mathcal{H}H$, while $\|H\|_{\text{op}} = \max_{x\in\mathbb{S}^{P-1}} |H(x)|$) and by Lemma I.6 $\sup_{z\in\mathbb{R}^P} \|\mathcal{H}e_r(z)\|_\infty \leq \kappa_0 r^{m-2}$ and $\sup_{z\in\mathbb{R}^P} \|\partial_x^m e_r(z)\|_\infty \leq \kappa_1$

$$\begin{aligned}
&\leq \frac{2\|H\|_{\text{op}} + \kappa_0 r^{m-2}}{s_0(m-1)} c e^{s_0(m-1)t} + \kappa_1 c_0 e^{s_0(m-1)t} \\
&\leq \left(\frac{2\|H\|_{\text{op}} + \kappa_0 r^{m-2}}{s_0(m-1)} c + \kappa_1 c_0 \right) e^{s_0(m-1)t}
\end{aligned}$$

Since by assumption $s_0 > \frac{2\|H\|_{\text{op}}}{m-1}$, we can choose r small enough such that $\frac{2\|H\|_{\text{op}} + \kappa_0 r^{m-2}}{s_0(m-1)} < 1$. We can then choose c large enough so that $\frac{2\|H\|_{\text{op}} + \kappa_0 r^{m-2}}{s_0(m-1)} c + \kappa_1 c_0 \leq c$. With these choices of r and c, we obtain that $\|b'(t)\| \leq c e^{s_0(m-1)t}$ and therefore $b' \in B_c$ as needed.

(2) **Contraction:** We need to bound for any $x, y \in x_0 + B_c$

$$\|\Phi_{C_r}(x) - \Phi_{C_r}(y)\|_\beta^2 = \left\| t \mapsto \int_{-\infty}^{t} [\nabla C_r(x(u)) - \nabla C_r(y(u))]\, du \right\|_\beta^2,$$

for any $\beta < (m-1)s_0$.

From point (1), we know that $\Phi_{C_r}(x), \Phi_{C_r}(y) \in x_0 + B_c$ and hence $\|\Phi_{C_r}(x) - \Phi_{C_r}(y)\|_\beta^2 \leq \infty$ (since $\beta < (m-1)s_0$).

From point (3) of Lemma I.8 we have:

$$\begin{aligned}
\left\| t \mapsto \int_{-\infty}^{t} [\nabla C_r(x(u)) - \nabla C_r(y(u))]\, du \right\|_\beta &\leq \frac{1}{\beta} \|t \mapsto \nabla C_r(x(t)) - \nabla C_r(y(t))\|_\beta \\
&\leq \frac{\sup_z \|\mathcal{H}C_r(z)\|_{\text{op}}}{\beta} \|x - y\|_\beta .
\end{aligned}$$

By the localization, we have $\sup_z \|\mathcal{H}C_r(z)\|_{\text{op}} \leq \sup_z \|\mathcal{H}H(z)\|_{\text{op}} + \sup_z \|\mathcal{H}e_r(z)\|_{\text{op}} \leq 2\|H\|_{\text{op}} + \kappa_0 r^{m-2}$.

Therefore to guarantee a contraction, we choose $\beta > 2\left\|H\right\|_{\text{op}} + \kappa_0 r^{m-2}$, so that $\frac{\sup_z \|\mathcal{H}C_r(z)\|_{\text{op}}}{\beta} < 1$. Therefore β lies in an open interval

$$\left(\frac{2\left\|H\right\|_{\text{op}} + \kappa_0 r^{m-2}}{(m-1)s_0}\right)(m-1)s_0 < \beta < (m-1)s_0$$

which is non-empty since we have chosen r small enough in point (1) such that $\frac{2\|H\|_{\text{op}} + \kappa_0 r^{m-2}}{s_0(m-1)} < 1$. □

Deep case

For the case $k > 2$, we consider the following norm

$$\|b\|_\alpha^2 = \int_{-\infty}^0 (-t)^{2\alpha-1} \|b(t)\|^2 \, dt.$$

If $\alpha < \frac{m-k+1}{k-2}$ then this norm is finite on any corrections $b \in B_c$ (i.e. if $\|b(t)\| \leq c(-t)^{-\frac{m-k+1}{k-2}}$), since

$$\|b\|_\alpha^2 \leq c^2 \int_{-\infty}^0 (-t)^{2(\alpha - \frac{m-k+1}{k-2})-1} dt < \infty.$$

The set $x_0 + B_c$ equipped with the distance $\|\cdot\|_\alpha$ therefore defines a complete metric space.

Again, for paths x, y such that $\|x\|_w, \|y\|_w < \infty$, we define the scalar product

$$\langle x, y\rangle_w = \int_{-\infty}^0 (-t)^{2\alpha-1} \langle x(t), y(t)\rangle \, dt.$$

Lemma I.8 is now replaced by the following:

Lemma I.10. *For any differentiable paths x, y with $\|x\|_w, \|y\|_w < \infty$, we have*

1. $\langle x, -t\dot{y}\rangle_w = 2\alpha \langle x, y\rangle_w - \langle -t\dot{x}, y\rangle_w$.
2. $\frac{1}{\alpha}\langle x, -t\dot{x}\rangle_w = \|x\|_w^2$.
3. $\|x\|_w \leq \frac{1}{\alpha}\|-t\dot{x}\|_w$.

Proof. The first point is obtained by integration by part:

$$\begin{aligned}\langle x, -t\dot{y}\rangle_w &= \int_{-\infty}^0 (-t)^{2\alpha} x(t)\dot{y}(t) dt \\ &= \int_{-\infty}^0 2\alpha(-t)^{2\alpha-1} x(t)y(t)dt - \int_{-\infty}^0 (-t)^{2\alpha}\dot{x}(t)y(t)dt \\ &= 2\alpha\langle x, y\rangle_w - \langle -t\dot{x}, y\rangle_w .\end{aligned}$$

Taking $x = y$, we obtain the second point. Finally, the last point follows from the second one since:

$$\|x\|_w^2 = \frac{1}{\alpha}\langle x, -t\dot{x}\rangle_w \leq \frac{1}{\alpha}\|x\|_w \|-t\dot{x}\|_w .$$

Under certain conditions, we can ensure that there is an α such that Φ is a contraction on $x_0 + B_c$ w.r.t. the norm $\|\cdot\|_\alpha$: □

Lemma I.11. *Let $C_r = H + e_r$ be a localized (k, m)-approximately homogeneous loss as in I.3.3, where H is a polynomial, with $k > 2$. Choose a $s_0 > \frac{k-1}{m-k+1} k \|H\|_\infty$. Let $x_0 \in \mathcal{F}_H[s_0]$, there exist $r > 0$ small enough, $c > 0$ large enough and $T < 0$ small enough, such that the map Φ is a contraction on the set $x_0 + B_{c,T} = \left\{x_0 + b : \|b(t)\| \leq ce^{s(m-1)t}, \forall t < T\right\}$ w.r.t. to the norm $\|\cdot\|_\alpha$ for some well-suited α.*

Proof. (1) **Self-map:** Let $x_0 + b \in x_0 + B_{c,T}$, we first show that $\Phi(x_0 + b) \in x_0 + B_{c,T}$. Let us rewrite

$$\begin{aligned}\Phi(x_0 + b) &= -\int_{-\infty} \nabla C_r(x_0(u) + b(u))du \\ &= -\int_{-\infty} \nabla H(x_0(u))du + \int_{-\infty} \nabla H(x_0(u)) - \nabla C_r(x_0(u) + b(u))du \\ &= x_0 + b'\end{aligned}$$

where

$$\begin{aligned}b'(t) &= \int_{-\infty}^{t} \nabla H(x_0(u)) - \nabla C_r(x_0(u) + b(u))du \\ &= \int_{-\infty}^{t} \nabla H(x_0(u)) - \nabla C_r(x_0(u))du \\ &+ \int_{-\infty}^{t} \nabla C_r(x_0(u)) - \nabla C_r(x_0(u) + b(u))du \\ &= -\int_{-\infty}^{t} \nabla e_r(x_0(u))du \\ &+ \int_{-\infty}^{t} \nabla C_r(x_0(u)) - \nabla C_r(x_0(u) + b(u))du.\end{aligned}$$

Our goal is to show that $b'(t) \in B_c$, i.e. that $\|b'(t)\| \leq c(-t)^{\frac{k-m-1}{k-2}}$. We bound the two terms separately:

$$\|b'(t)\| \leq \left\|\int_{-\infty}^{t} \nabla e_r(x_0(u))du\right\| + \left\|\int_{-\infty}^{t} \left[\nabla C_r(x_0(u)) - \nabla C_r(x_0(u) + b(u))\right] du\right\|.$$

The first term $\left\|\int_{-\infty}^{t} \nabla e_r(x_0(u))du\right\|$ is bounded by

$$\begin{aligned}&\int_{-\infty}^{t} \|\nabla e_r(x_0(u))\| \, du \\ &\leq m \|\partial_x^m e_r\| \int_{-\infty}^{t} \|x_0(u)\|^{m-1} \, du \\ &\leq m\kappa s_0^{-\frac{m-1}{k-2}} (k-2)^{-\frac{m-1}{k-2}} \int_{-\infty}^{t} (-u)^{-\frac{m-1}{k-2}} \, du \\ &= m\kappa s_0^{-\frac{m-1}{k-2}} (k-2)^{-\frac{m-1}{k-2}} \frac{k-2}{m-k+1} (-t)^{\frac{k-m-1}{k-2}}\end{aligned}$$

$$= m\kappa s_0^{-\frac{m-1}{k-2}} \frac{(k-2)^{\frac{k-m-1}{k-2}}}{(m-k+1)} (-t)^{\frac{k-m-1}{k-2}}.$$

The second term $\left\| \int_{-\infty}^{t} \nabla C_r(x_0(u)) - \nabla C_r(x_0(u)+b(u)) du \right\|$ is bounded by

$$\begin{aligned}
&\int_{-\infty}^{t} \|\nabla C_r(x_0(u)) - \nabla C_r(x_0(u)+b(u))\| \, du \\
&\leq \frac{\sup_z \left\|\partial_z^k C_r(z)\right\|_{\text{op}}}{(k-2)!} \int_{-\infty}^{t} \|b(u)\| \max\left\{\|x_0(u)\|, \|x_0(u)+b(u)\|\right\}^{k-2} du
\end{aligned}$$

by Lemma I.13. Let us first bound $\max\left\{\|x_0(u)\|, \|x_0(u)+b(u)\|\right\}^{k-2}$ by

$$\begin{aligned}
(\|x_0(u)\| + \|b(u)\|)^{k-2} &\leq \left((s_0(k-2)(-u))^{-\frac{1}{k-2}} + c(-u)^{-\frac{m-k+1}{k-2}} \right)^{k-2} \\
&= (s_0(k-2)(-u))^{-1} + \sum_{i=1}^{k-2} \begin{pmatrix} k-2 \\ i \end{pmatrix} (s_0(k-2)(-u))^{-1+\frac{i}{k-2}} c(-u)^{-\frac{m-k+1}{k-2} i} \\
&= (s_0(k-2)(-u))^{-1} + (s_0(k-2)(-u))^{-1} \sum_{i=1}^{k-2} \begin{pmatrix} k-2 \\ i \end{pmatrix} (s_0(k-2))^{\frac{i}{k-2}} c^i (-u)^{-\frac{m-k}{k-2} i} \\
&\leq (s_0(k-2)(-u))^{-1} \left[1 + \sum_{i=1}^{k-2} \begin{pmatrix} k-2 \\ i \end{pmatrix} (s_0(k-2))^{\frac{i}{k-2}} c^i (-T)^{-\frac{m-k}{k-2} i} \right],
\end{aligned}$$

for any ϵ, we can choose $T < 0$ small enough so that $\max\left\{\|x_0(u)\|, \|x_0(u)+b(u)\|\right\}^{k-2}$ is bounded by $(s_0(k-2)(-u))^{-1}[1+\epsilon]$.

Using also the bounds $\frac{\sup_z \left\|\partial_z^k C_r(z)\right\|_{\text{op}}}{(k-2)!} \leq k(k-1)\|H\|_\infty + \frac{\kappa}{(k-2)!} r^{m-k}$ and $\|b(u)\| \leq c(-u)^{-\frac{m-k+1}{k-2}}$, the second term $\left\| \int_{-\infty}^{t} \nabla C_r(x_0(u)) - \nabla C_r(x_0(u)+b(u)) du \right\|$ can be bounded by

$$\begin{aligned}
&\frac{k(k-1)\|H\|_{\text{op}} + \frac{\kappa}{(k-2)!} r^{m-k}}{s_0(k-2)} (1+\epsilon) \int_{-\infty}^{t} c(-u)^{-\frac{m-k+1}{k-2}-1} du \\
&= \frac{k(k-1)\|H\|_{\text{op}} + \frac{\kappa}{(k-2)!} r^{m-k}}{s_0(m-k+1)} (1+\epsilon) c(-t)^{-\frac{m-k+1}{k-2}}.
\end{aligned}$$

We choose $r > 0$ small enough, $c > 0$ large enough, and $T < 0$ small enough so that $\frac{k(k-1)\|H\|_{\text{op}} + \frac{\kappa}{(k-2)!} r^{m-k}}{s_0(m-k+1)} (1+\epsilon) < 1$ and $\left[m\kappa s_0^{-\frac{m-1}{k-2}} \frac{(k-2)^{-\frac{m-k+1}{k-2}}}{m-k+1} + \frac{k(k-1)\|H\|_{\text{op}} + \frac{\kappa}{(k-2)!} r^{m-k}}{s_0(m-k+1)} (1+\epsilon) c \right] \leq c$ so that

$$\begin{aligned}
\|b'(t)\| &\leq \left[m\kappa s_0^{-\frac{m-1}{k-2}} \frac{(k-2)^{-\frac{m-k+1}{k-2}}}{m-k+1} + \frac{k(k-1)\|H\|_{\text{op}} + \frac{\kappa}{(k-2)!} r^{m-k}}{s_0(m-k+1)} (1+\epsilon) c \right] (-t)^{-\frac{m-k+1}{k-2}} \\
&\leq c(-t)^{-\frac{m-k+1}{k-2}}
\end{aligned}$$

and therefore $b' \in B_c$.

(2) **Contraction:** We have, for any $x, y \in x_0 + B_c$

$$\begin{aligned}\|\Phi(x) - \Phi(y)\|_w &= \left\| \int_{-\infty}^{t} \nabla C_r(x(s)) - \nabla C_r(y(s)) ds \right\|_\alpha \\ &\leq \frac{1}{\alpha} \left\| -t \left(\nabla C_r(x) - \nabla C_r(y) \right) \right\|_\alpha \\ &\leq \frac{\|\partial_k C_r(x_0)\|_\infty}{\alpha(k-2)!} \left\| -t \|x-y\| \left(\max\{\|x\|, \|y\|\} \right)^{k-2} \right\|_\alpha\end{aligned}$$

by Lemma I.13. Using the same argument as in point (1) to bound $(\max\{\|x\|, \|y\|\})^{k-2}$, we obtain

$$\begin{aligned}&\frac{k(k-1)\|H\|_\infty + \frac{\kappa}{(k-2)!} r^{m-k}}{\alpha} \left\| -t \|x-y\| \left(s_0(k-2)(-t) \right)^{-1} \right\|_\alpha \\ &\leq \frac{k(k-1)\|H\|_\infty + \frac{\kappa}{(k-2)!} r^{m-k}}{\alpha s_0(k-2)} (1+\epsilon) \|x-y\|_\alpha\end{aligned}$$

To obtain a contraction, we need to choose $\alpha > \frac{k(k-1)\|H\|_\infty + \frac{\kappa}{(k-2)!} r^{m-k}}{s_0(k-2)}(1+\epsilon)$. To summarize α must lie within the two bounds:

$$\frac{k(k-1)\|H\|_\infty + \frac{\kappa}{(k-2)!} r^{m-k}}{(m-k+1)s}(1-\epsilon)\frac{m-k+1}{k-2} < \alpha < \frac{m-k+1}{k-2}$$

which is possible since we have chosen r, c and T such that $\frac{k(k-1)\|H\|_\infty + \frac{\kappa}{(k-2)!} r^{m-k}}{(m-k+1)s}(1-\epsilon) < 1$. □

Proof of Theorem 9.2

We have now all the tools to prove the Theorem 4 of the main:

Theorem I.4 (Theorem 9.2 of the main text)**.** *Assume that the largest singular value s_1 of the gradient of C at the origin $\nabla C(0) \in \mathbb{R}^{n_L \times n_0}$ has multiplicity 1. There is a deterministic gradient flow path $\underline{\theta}^1$ in the space of width-1 DLNs such that, with probability 1 if $L \leq 3$, and probability at least 1/2 if $L > 3$, there exists an escape time t_α^1 and a rotation R such that*

$$\lim_{\alpha \to 0} \theta_\alpha(t_\alpha^1 + t) = R I^{(1 \to w)} \underline{\theta}^1(t).$$

Proof. From Proposition I.4 we know that with prob. 1 there is a time horizon t_α^1 and an escape path such that $\lim_{\alpha \to 0} \theta_\alpha(t_\alpha^1 + t) = \theta^1(t)$ which for any $\epsilon > 0$ escapes the origin at a rate of at least $e^{(s^* + \epsilon)t}$ for shallow networks and $[(k-2)(s^* + \epsilon)t]^{\frac{1}{2-k}}$ for deep networks, where $s^* = -L^{-\frac{L-2}{2}} s_1$.

Since the loss C^{NN} is $(L, 2L)$-approximately homogeneous, we can apply Theorem I.3 to obtain that θ^1 must be in bijection with an escape path of the homogeneous loss H of the same speed. For small enough ϵ the only escape path of H of at least this speed are of the form $\rho^* e^{s^*(t+T)}$ for shallow networks and $\rho^* \left((k-1)s^*(-t-T) \right)^{-\frac{1}{k-1}}$ for some constant T and an optimal escape

direction ρ^*. We therefore call θ^1 an optimal escape path since it belongs to the unique set of paths which escape at an optimal speed and are in bijection to the optimal escape directions.

Assuming that the largest singular value of s_1 of $\nabla C(0)$ has multiplicity 1, with singular vectors u_1, v_1, the optimal escape directions are of the form

$$\rho^* = RI^{(1\to w)}(\underline{\rho}^*) = \frac{1}{\sqrt{L}} RI^{(1\to w)}\left(-v_1^T, 1, \dots, 1, u_1\right)$$

for any rotation R. In the width 1 network, there is an optimal escape path $\underline{\theta}^1$ corresponding to the escape direction $\underline{\rho}^*$, then by the unicity of the bijection of Theorem I.3, the escape path $RI^{(1\to w)}(\underline{\theta}^1)$ is the unique optimal escape path escaping along $RI^{(1\to w)}(\underline{\rho}^*)$, as a result, we know that $\theta^1 = RI^{(1\to w)}(\underline{\theta}^1)$ for some rotation R. □

I.4 Technical Results

In this section, we state and prove a few technical lemmas used throughout the appendix.

Let us first prove a generalization of GrÃ¶nwall's inequality for polynomial bounds:

Lemma I.12. *Let $x : \mathbb{R}^+ \to \mathbb{R}$ which satisfy*

$$\partial_t x(t) \le c x(t)^\alpha,$$

for some $c > 0$ and $\alpha > 1$. Then, for all $t < \frac{x(0)^{1-\alpha}}{c(\alpha-1)}$,

$$x(t) \le \left[x(0)^{1-\alpha} - c(\alpha-1)t\right]^{-\frac{1}{\alpha-1}}.$$

Proof. Note that the function $y(t) = \left[x(0)^{1-\alpha} + c(1-\alpha)t\right]^{-\frac{1}{\alpha-1}}$ satisfies $y(0) = x(0)$ and for all $t < \frac{x(0)^{1-\alpha}}{c(\alpha-1)}$:

$$\partial_t y(t) = c y(t)^\alpha.$$

We conclude by showing that if $x(t) \le y(t)$ then $x(s) \le y(s)$ for all $t \le s \le \frac{x(0)^{1-\alpha}}{c(\alpha-1)}$: this follows from the fact that on the diagonal, i.e. when $x(t) = y(t)$, we have

$$\partial_t x(t) - \partial_t y(t) \le c x(t)^\alpha - c y(t)^a = 0$$

which implies that the flow points towards the inside of $\{(x, y) : x \le y\}$. □

Let us now state a lemma to bound the gradient of a cost C in terms of its high order derivatives:

Lemma I.13. *Let C be a cost and k the largest integer such that $\partial_x^n C(0) = 0$ for all $n < k$ and $\left\|\partial_x^k C\right\|_\infty < \infty$, then*

1. For all x, $\|\nabla C(x)\| \le \frac{\left\|\partial_x^k C\right\|_\infty \|x\|^{k-1}}{(k-1)!}$.
2. For all x, y, $\|\nabla C(x) - \nabla C(y)\| \le \frac{1}{(k-2)!} \left\|\partial_x^k C\right\|_\infty \|x - y\| \left(\max\{\|x\|, \|y\|\}\right)^{k-2}$.

Proof. (1)

$$\begin{aligned}\|\nabla C(x)\| &= \left\| \int_0^1 \mathcal{H}C(\lambda x)[x] d\lambda \right\| \\ &= \left\| \int_0^1 \int_0^{\lambda_1} \cdots \int_0^{\lambda_{k-2}} \partial_x^k C(\lambda_1 \cdots \lambda_{k-1} z_t)[x, \ldots, x] dt_1 \cdots dt_1 \right\| \\ &\leq \int_0^1 \int_0^{\lambda_1} \cdots \int_0^{\lambda_{m-2}} \left\|\partial_x^k C\right\|_\infty \|x\|^{k-1} dt_1 \cdots dt_1 \\ &\leq \frac{\left\|\partial_x^k C\right\|_\infty \|x\|^{k-1}}{(k-1)!}\end{aligned}$$

(2) First note that $\nabla C(x) - \nabla C(y)$ is equal to

$$\int_0^1 \mathcal{H}C(z_t)[x-y] dt$$

where $z_t = tx + (1-t)y$. This can further be rewritten as

$$\int_0^1 \int_0^1 \partial_x^3 C(t_1 z_{t,t_1})[x-y, z_t] dt_1 \, dt.$$

Iterating this procedure, we obtain that $\nabla C(x) - \nabla C(y)$ equals

$$\int_0^1 \int_0^1 \int_0^{t_1} \cdots \int_0^{t_{k-3}} \partial_x^k C(t_1 \cdots t_{k-2} z_{t,t_1})[x-y, z_t, \ldots, z_t] dt_{k-2} \cdots dt_2 \, dt_1 \, dt.$$

Since the volume of the set $\{(t_1, \ldots, t_{k-2}) : 0 \geq t_1 \geq \cdots \geq t_{k-2} \geq 0\}$ is $\frac{1}{(k-2)!}$ we have

$$\begin{aligned}\|\nabla C(x) - \nabla C(y)\| &\leq \int_0^1 \int_0^1 \int_0^{t_1} \cdots \int_0^{t_{k-3}} \left\|\partial_x^k C\right\| \|x-y\| \|z_t\|^{k-2} dt_{k-2} \cdots dt_2 \, dt_1 \, dt \\ &\leq \frac{1}{(k-2)!} \left\|\partial_x^k C\right\|_\infty \|x-y\| \left(\max\{\|x\|, \|y\|\}\right)^{k-2}.\end{aligned}$$

□

Bibliography

[1] Madhu S. Advani and Andrew M. Saxe. High-dimensional dynamics of generalization error in neural networks, 2017.

[2] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. pages 242–252, 2019.

[3] Erik Andreassen, Anders Clausen, Mattias Schevenels, Boyan Lazarov, and Ole Sigmund. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization*, 43:1–16, 11 2011.

[4] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2019.

[5] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32, 2019.

[6] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*, 32, 2019.

[7] Devansh Arpit, Yingbo Zhou, Bhargava Kota, and Venu Govindaraju. Normalization propagation: A parametric technique for removing internal covariate shift in deep networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1168–1176, New York, New York, USA, 20–22 Jun 2016. PMLR.

[8] Benson Au, Guillaume Cébron, Antoine Dahlqvist, Franck Gabriel, and Camille Male. Large permutation invariant random matrices are asymptotically free over the diagonal, 2018. To appear in Annals of Probability.

[9] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, pages 185–209, 2013.

[10] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017.

[11] Zhidong Bai and Zhou Wang. Large sample covariance matrices without independence structures in columns. *Statistica Sinicia*, 18:425–442, 2008.

[12] Marco Baity-Jesi, Levent Sagun, Mario Geiger, Stefano Spigler, Gerard Ben Arous, Chiara Cammarota, Yann LeCun, Matthieu Wyart, and Giulio Biroli. Comparing Dynamics: Deep Neural Networks versus Glassy Systems. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 314–323. PMLR, 10–15 Jul 2018.

[13] Pierre Baldi and Kurt Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[14] Andrew J Ballard, Ritankar Das, Stefano Martiniani, Dhagash Mehta, Levent Sagun, Jacob D Stevenson, and David J Wales. Energy landscapes for machine learning. *Physical Chemistry Chemical Physics*, 2017.

[15] Saurabh Banga, Harsh Gehani, Sanket Bhilare, Sagar Patel, and Levent Kara. 3d topology optimization using convolutional neural networks. *CoRR*, abs/1808.07440, 2018.

[16] Yamini Bansal, Madhu Advani, David D Cox, and Andrew M Saxe. Minnorm training: an algorithm for training over-parameterized deep neural networks. *CoRR*, 2018.

[17] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *arXiv preprint arXiv:1906.11300*, 2019.

[18] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[19] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *arXiv preprint arXiv:1903.07571*, 2019.

[20] Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. *arXiv preprint*, Feb 2018.

[21] Bendsoe and Sigmund. Topology optimization: Theory, methods and applications. *Springer Science and Business*, April 2013.

[22] Martin Bendsøe. Optimal shape design as a material distribution problem. structural optimization 1, 193-202. *Structural Optimization*, 1:193–202, 01 1989.

[23] Lucas Benigni and Sandrine Péché. Eigenvalue distribution of nonlinear models of random matrices. *arXiv preprint arXiv:1904.03090*, 2019.

[24] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[25] Blake Bordelon, Abdulkadir Canatar, and Cengiz Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. *arXiv preprint arXiv:2002.02561*, 2020.

[26] Siegfried Bös and Manfred Opper. Dynamics of training. In *Advances in Neural Information Processing Systems*, pages 141–147, 1997.

[27] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[28] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[29] Andrea Caponnetto and Ernesto De Vito. Optimal rates for the regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2007.

[30] Aaditya Chandrasekhar and K. Suresh. Length scale control in topology optimization using fourier enhanced neural networks. 2020.

[31] Aaditya Chandrasekhar and Krishnan Suresh. Tounn: Topology optimization using neural networks. *Structural and Multidisciplinary Optimization*, 2020.

[32] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv preprint arXiv:1611.01838*, 2016.

[33] Yanqing Chen, Timothy A. Davis, and William W. Hager. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software*, pages 1–14, 2008.

[34] Lenaic Chizat and Francis Bach. A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

[35] Lénaïc Chizat and Francis Bach. On the Global Convergence of Gradient Descent for Over-parameterized Models using Optimal Transport. In *Advances in Neural Information Processing Systems 31*, pages 3040–3050. Curran Associates, Inc., 2018.

[36] Lénaïc Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1305–1338. PMLR, 09–12 Jul 2020.

[37] Youngmin Cho and Lawrence K. Saul. Kernel Methods for Deep Learning. In *Advances in Neural Information Processing Systems 22*, pages 342–350. Curran Associates, Inc., 2009.

[38] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The Loss Surfaces of Multilayer Networks. *Journal of Machine Learning Research*, 38:192–204, nov 2015.

[39] Yaim Cooper. The loss landscape of overparameterized neural networks. *arXiv preprint arXiv:1804.10200*, 2018.

[40] Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz S Kandola. On kernel-target alignment. In *Advances in neural information processing systems*, pages 367–373, 2002.

[41] Zhen Dai, Mina Karzand, and Nathan Srebro. Representation costs of linear neural networks: Analysis and design. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[42] Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. volume abs/1602.05897. 2016.

[43] Stéphane d'Ascoli, Maria Refinetti, Giulio Biroli, and Florent Krzakala. Double trouble in double descent: Bias and variance (s) in the lazy regime. *arXiv preprint arXiv:2003.01054*, 2020.

[44] Yann N. Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and Attacking the Saddle Point Problem in High-dimensional Non-convex Optimization. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2933–2941, Cambridge, MA, USA, 2014. MIT Press.

[45] Timothy A. Davis. User guide for cholmod: a sparse cholesky factorization and modification package. 2009.

[46] Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

[47] Alexander G. de G. Matthews, Jiri Hron, Richard E. Turner, and Zoubin Ghahramani. Sample-then-optimize posterior sampling for bayesian linear models. In *NIPS workshop on Advances in Approximate Bayesian Inference*, 2017.

[48] Edgar Dobriban and Stefan Wager. High-dimensional asymptotics of prediction: Ridge regression and classification. *Ann. Statist.*, 46(1):247–279, 02 2018.

[49] Sever Silvestru Dragomir. *Some Gronwall Type Inequalities and Applications*. Nova Science Publishers, 2003.

[50] Simon S. Du, Chi Jin, Jason D. Lee, Michael I. Jordan, Barnabás Póczos, and Aarti Singh. Gradient descent can take exponential time to escape saddle points. In *Proceedings of the 31st International Conference on Neural Information Processing SystemsDecember 2017, NIPS'17*, pages 1067–1077. Curran Associates, Inc., 2017.

[51] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

[52] Benjamin Dupuis and Arthur Jacot. Dnn-based topology optimisation: Spatial invariance and neural tangent kernel. *Advances in Neural Information Processing Systems*, 34, 2021.

[53] Morris Eaton. Multivariate statistics: A vector space approach. *Journal of the American Statistical Association*, 80, 01 2007.

[54] Armin Eftekhari. Training linear neural networks: Non-local convergence and complexity results. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2836–2847. PMLR, 13–18 Jul 2020.

[55] Noureddine El Karoui et al. The spectrum of kernel random matrices. *The Annals of Statistics*, 38(1):1–50, 2010.

[56] Khalil Elkhalil, Abla Kammoun, Xiangliang Zhang, Mohamed-Slim Alouini, and Tareq Al-Naffouri. Risk convergence of centered kernel ridge regression with large dimensional data. *IEEE Transactions on Signal Processing*, 68:1574–1588, 2020.

[57] Andreas Engel and Christian Van den Broeck. *Statistical mechanics of learning*. Cambridge University Press, 2001.

[58] Gregory E Fasshauer and Michael J McCourt. Stable evaluation of gaussian radial basis function interpolants. *SIAM Journal on Scientific Computing*, 34(2):A737–A762, 2012.

[59] Silvio Franz, Sungmin Hwang, and Pierfrancesco Urbani. Jamming in multilayer supervised learning models. *arXiv preprint arXiv:1809.09945*, 2018.

[60] Silvio Franz and Giorgio Parisi. The simplest model of jamming. *Journal of Physics A: Mathematical and Theoretical*, 49(14):145001, 2016.

[61] Silvio Franz, Giorgio Parisi, Maxime Sevelev, Pierfrancesco Urbani, and Francesco Zamponi. Universality of the sat-unsat (jamming) threshold in non-convex continuous constraint satisfaction problems. *SciPost Physics*, 2(3):019, 2017.

[62] Silvio Franz, Giorgio Parisi, Pierfrancesco Urbani, and Francesco Zamponi. Universal spectrum of normal modes in low-temperature glasses. *Proceedings of the National Academy of Sciences*, 112(47):14539–14544, 2015.

[63] C Daniel Freeman and Joan Bruna. Topology and geometry of deep rectified network optimization landscapes. *International Conference on Learning Representations*, 2017.

[64] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.

[65] Franck Gabriel. Combinatorial theory of permutation-invariant random matrices ii: Cumulants, freeness and Levy processes. *arXiv preprint arXiv:1507.02465*, 2015.

[66] Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning . abs/1901.01608, 2019.

[67] Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler, and Matthieu Wyart. Scaling description of generalization with number of parameters in deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(2):023401, 2020.

[68] Mario Geiger, Stefano Spigler, Stéphane d'Ascoli, Levent Sagun, Marco Baity-Jesi, Giulio Biroli, and Matthieu Wyart. The jamming transition as a paradigm to understand the loss landscape of deep neural networks. *arXiv preprint arXiv:1809.09349*, 2018.

[69] Mario Geiger, Stefano Spigler, Stéphane d'Ascoli, Levent Sagun, Marco Baity-Jesi, Giulio Biroli, and Matthieu Wyart. Jamming transition as a paradigm to understand the loss landscape of deep neural networks. *Physical Review E*, 100(1):012115, 2019.

[70] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020.

[71] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.

[72] L Lo Gerfo, Lorenzo Rosasco, Francesca Odone, E De Vito, and Alessandro Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008.

[73] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2232–2241, Long Beach, California, USA, 09–15 Jun 2019. PMLR.

[74] Gauthier Gidel, Francis Bach, and Simon Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[75] Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. The implicit bias of depth: How incremental learning drives generalization. In *International Conference on Learning Representations*, 2020.

[76] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[77] Tilmann Gneiting. Strictly and non-strictly positive definite functions on spheres. *Bernoulli*, 19(4):1327–1349, 2013.

[78] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, pages 2672–2680, jun 2014.

[79] Andreas Griewank and Christèle Faure. Reduced functions, gradients and hessians from fixed-point iterations for state equations. *Numerical Algorithms*, 30:113–139, 06 2002.

[80] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841. PMLR, 10–15 Jul 2018.

[81] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[82] Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *CoRR*, abs/1812.04754, 2018.

[83] Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? *arXiv preprint arXiv:1801.03744*, 2018.

[84] Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel, 2019.

[85] Trevor Hastie, Andrea Montanari, Saharon Rosset, and Ryan J Tibshirani. Surprises in high-dimensional ridgeless least squares interpolation. *arXiv preprint arXiv:1903.08560*, 2019.

[86] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. Mean-field behaviour of neural tangent kernel for deep neural networks. *arXiv preprint arXiv:1905.13654*, 2019.

[87] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. In *International Conference on Machine Learning*, pages 2672–2680. PMLR, 2019.

[88] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. Training dynamics of deep networks using stochastic gradient descent via neural tangent kernel. *arXiv preprint arXiv:1905.13654*, 2019.

[89] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

[90] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[91] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[92] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *Advances in Neural Information Processing Systems*, pages 1729–1739, 2017.

[93] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359 – 366, 1989.

[94] Stephan Hoyer, Jascha Sohl-Dickstein, and Sam Greydanus. Neural reparameterization improves structural optimization. *CoRR*, abs/1909.04240, 2019.

[95] Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. 119:4542–4551, 13–18 Jul 2020.

[96] Kaixuan Huang, Yuqing Wang, Molei Tao, and Tuo Zhao. Why do deep residual networks generalize better than deep feedforward networks?—a neural tangent kernel perspective. *Advances in Neural Information Processing Systems*, 33, 2020.

[97] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology*, 148(3):574, 1959.

[98] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 448–456. JMLR.org, 2015.

[99] A. G. Ivakhnenko. Polynomial theory of complex systems. *IEEE Trans. Syst. Man Cybern.*, 1:364–378, 1971.

[100] Aleksei Grigorevich Ivakhnenko and Valentin Grigor'evich Lapa. Cybernetic predicting devices. 1966.

[101] Andrea J Liu, Sidney R Nagel, W Saarloos, and Matthieu Wyart. *The jamming scenario - an introduction and outlook*. OUP Oxford, 06 2010.

[102] Arthur Jacot, Berfin Şimşek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Implicit regularization of random feature models. In H. Daumé and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 12-18 July 2020, Vienna, Austria*, Proceedings of Machine Learning Research. 2020.

[103] Arthur Jacot, Berfin Şimşek, Francesco Spadaro, Clement Hongler, and Franck Gabriel. Kernel alignment risk estimator: Risk prediction from training data. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15568–15578. Curran Associates, Inc., 2020.

[104] Arthur Jacot, Franck Gabriel, François Ged, and Clément Hongler. Order and chaos: Ntk views on dnn normalization, checkerboard and boundary artifacts. *arXiv preprint arXiv:1907.05715*, 2019.

[105] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems 31*, pages 8580–8589. Curran Associates, Inc., 2018.

[106] Arthur Jacot, Franck Gabriel, and Clement Hongler. The asymptotic spectrum of the hessian of dnn throughout training. In *International Conference on Learning Representations*, 2020.

[107] Arthur Jacot, François Ged, Berfin Şimşek, Clément Hongler, and Franck Gabriel. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity, 2022.

[108] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *CoRR*, abs/1810.02032, 2018.

[109] Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17176–17186. Curran Associates, Inc., 2020.

[110] Jiaqi Jiang and Jonathan A. Fan. Global optimization of dielectric metasurfaces using a physics-driven neural network. *Nano Letters*, 19(8):5366–5372, Jul 2019.

[111] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1724–1732. JMLR.org, 2017.

[112] Ryo Karakida, Shotaro Akaho, and Shun-Ichi Amari. Universal Statistics of Fisher Information in Deep Neural Networks: Mean Field Approach. jun 2018.

[113] Ryo Karakida, Shotaro Akaho, and Shun-ichi Amari. The normalization method for alleviating pathological sharpness in wide neural networks. In *Advances in Neural Information Processing Systems*, pages 6403–6413, 2019.

[114] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *arXiv preprint arXiv:1812.04948*, 2018.

[115] Kenji Kawaguchi. Deep learning without poor local minima. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[116] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[117] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.

[118] Günter Klambauer, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. Self-normalizing neural networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 971–980. Curran Associates, Inc., 2017.

[119] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[120] Yann Le Cun, Ido Kanter, and Sara A Solla. Eigenvalues of covariance matrices: Application to neural-network learning. *Physical Review Letters*, 66(18):2396, 1991.

[121] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[122] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[123] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[124] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[125] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[126] Jae Hoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep Neural Networks as Gaussian Processes. *ICLR*, 2018.

[127] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[128] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pages 8572–8583, 2019.

[129] Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124002, Dec 2020.

[130] Jason D Lee, Ioannis Panageas, Georgios Piliouras, Max Simchowitz, Michael I Jordan, and Benjamin Recht. First-order methods almost always avoid strict saddle points. *Mathematical programming*, 176(1):311–337, 2019.

[131] Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1246–1257, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[132] J. Lei Ba, J. R. Kiros, and G. E. Hinton. Layer Normalization. *arXiv e-prints*, July 2016.

[133] Moshe Leshno, Vladimir Lin, Allan Pinkus, and Shimon Schocken. Multilayer Feedforward Networks with a Non-Polynomial Activation Function Can Approximate Any Function. *Neural Networks*, 6(6):861–867, 1993.

[134] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

[135] Zhiyuan Li, Yuping Luo, and Kaifeng Lyu. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *International Conference on Learning Representations*, 2020.

[136] Zhiyuan Li, Ruosong Wang, Dingli Yu, Simon S Du, Wei Hu, Ruslan Salakhutdinov, and Sanjeev Arora. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809*, 2019.

[137] Tengyuan Liang and Alexander Rakhlin. Just interpolate: Kernel" ridgeless" regression can generalize. *arXiv preprint arXiv:1808.00387*, 2018.

[138] Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai. On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels. *arXiv preprint arXiv:1908.10292 [cs, math, stat]*, 2020.

[139] Zhenyu Liao and Romain Couillet. The dynamics of learning: A random matrix approach. *arXiv preprint arXiv:1805.11917*, 2018.

[140] Seppo Linnainmaa. *The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors*. PhD thesis, Master's Thesis (in Finnish), Univ. Helsinki, 1970.

[141] Zachary C Lipton. Stuck in a what? adventures in weight space. *International Conference on Learning Representations*, 2016.

[142] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Toward a theory of optimization for over-parameterized systems of non-linear equations: the lessons of deep learning. *arXiv preprint arXiv:2003.00307*, 2020.

[143] Kai Liu and Andres Tovar. An efficient 3d topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 50, 12 2014.

[144] Sifan Liu and Edgar Dobriban. Ridge regression: Structure, cross-validation, and sketching. In *International Conference on Learning Representations*, 2020.

[145] Cosme Louart, Zhenyu Liao, and Romain Couillet. A random matrix approach to neural networks. *The Annals of Applied Probability*, 28, 02 2017.

[146] Tao Luo, Zhi-Qin John Xu, Zheng Ma, and Yaoyu Zhang. Phase diagram for two-layer relu neural networks at infinite-width limit. *Journal of Machine Learning Research*, 22(71):1–47, 2021.

[147] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020.

[148] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

[149] Gilles Marck, Maroun Nemer, Jean-Luc Harion, Serge Russeil, and Daniel Bougeard. Topology optimization using the simp method for multiobjective conductive problems. *Numerical Heat Transfer Part B-fundamentals - NUMER HEAT TRANSFER PT B-FUND*, 61:439–470, 06 2012.

[150] Ulysse Marteau-Ferey, Dmitrii Ostrovskii, Francis Bach, and Alessandro Rudi. Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. *CoRR*, abs/1902.03046, 2019.

[151] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. *arXiv preprint arXiv:1902.06015*, 2019.

[152] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv preprint arXiv:1908.05355*, 2019.

[153] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[154] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, USA, 1969.

[155] Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 2018. https://distill.pub/2018/differentiable-parameterizations.

[156] Edward Moroshko, Blake E Woodworth, Suriya Gunasekar, Jason D Lee, Nati Srebro, and Daniel Soudry. Implicit bias in deep linear classification: Initialization scale vs training accuracy. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22182–22193. Curran Associates, Inc., 2020.

[157] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

[158] Brady Neal, Sarthak Mittal, Aristide Baratin, Vinayak Tantia, Matthew Scicluna, Simon Lacoste-Julien, and Ioannis Mitliagkas. A Modern Take on the Bias-Variance Tradeoff in Neural Networks. *arXiv preprint arXiv:1810.08591*, 2018.

[159] Radford M. Neal. *Bayesian Learning for Neural Networks*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1996.

[160] Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

[161] Behnam Neyshabur, Ryota Tomioka, Ruslan Salakhutdinov, and Nathan Srebro. Geometry of optimization and implicit regularization in deep learning. *arXiv preprint arXiv:1705.03071*, 2017.

[162] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[163] Zhenguo Nie, Tong Lin, Haoliang Jiang, and Levent Burak Kara. Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain. *CoRR*, abs/2003.04685, 2020.

[164] Maher Nouiehed and Meisam Razaviyayn. Learning deep models: Critical points and local openness. *INFORMS Journal on Optimization*, 2021.

[165] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.

[166] Greg Ongie, Rebecca Willett, Daniel Soudry, and Nathan Srebro. A function space view of bounded norm infinite width relu nets: The multivariate case. In *International Conference on Learning Representations*, 2020.

[167] Vardan Papyan. Measurements of three-level hierarchical structure in the outliers in the spectrum of deepnet hessians. *CoRR*, abs/1901.08244, 2019.

[168] Daniel S Park, Samuel L Smith, Jascha Sohl-dickstein, and Quoc V Le. Optimal SGD Hyperparameters for Fully Connected Networks. 2018.

[169] Razvan Pascanu and Yoshua Bengio. Revisiting Natural Gradient for Deep Networks. jan 2013.

[170] Razvan Pascanu, Yann N Dauphin, Surya Ganguli, and Yoshua Bengio. On the saddle point problem for non-convex optimization. *arXiv preprint*, 2014.

[171] Jeffrey Pennington and Yasaman Bahri. Geometry of Neural Network Loss Surfaces via Random Matrix Theory. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 2798–2806. PMLR, 06–11 Aug 2017.

[172] Jeffrey Pennington and Pratik Worah. The Spectrum of the Fisher Information Matrix of a Single-Hidden-Layer Neural Network. In *Advances in Neural Information Processing Systems 31*, pages 5415–5424. Curran Associates, Inc., 2018.

[173] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3360–3368. Curran Associates, Inc., 2016.

[174] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[175] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[176] Ali Rahimi and Benjamin Recht. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems 20*, pages 1177–1184. Curran Associates, Inc., 2008.

[177] Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Advances in neural information processing systems*, pages 1313–1320, 2009.

[178] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*, volume 2. MIT Press, 2006.

[179] M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: the rprop algorithm. In *IEEE International Conference on Neural Networks*, pages 586–591 vol.1, 1993.

[180] Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. pages 586–591, 1993.

[181] Ryan M Rifkin and Ross A Lippert. Notes on regularized least squares. 2007.

[182] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[183] Grant Rotskoff and Eric Vanden-Eijnden. Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks. In *Advances in Neural Information Processing Systems 31*, pages 7146–7155. Curran Associates, Inc., 2018.

[184] Grant M Rotskoff and Eric Vanden-Eijnden. Neural networks as interacting particle systems: Asymptotic convexity of the loss landscape and universal scaling of the approximation error. *arXiv preprint arXiv:1805.00915*, 2018.

[185] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. In *Advances in Neural Information Processing Systems*, pages 3215–3225, 2017.

[186] W. Rudin. *Fourier Analysis on Groups*. Wiley Classics Library. Wiley, 1990.

[187] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[188] David Saad and Sara A Solla. On-line learning in soft committee machines. *Physical Review E*, 52(4):4225, 1995.

[189] Levent Sagun, Léon Bottou, and Yann LeCun. Singularity of the hessian in deep learning. *CoRR*, abs/1611.07476, 2016.

[190] Levent Sagun, Utku Evci, V. Ugur Güney, Yann Dauphin, and Léon Bottou. Empirical Analysis of the Hessian of Over-Parametrized Neural Networks. *CoRR*, abs/1706.04454, 2017.

[191] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[192] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 901–909. Curran Associates, Inc., 2016.

[193] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 2483–2493. Curran Associates, Inc., 2018.

[194] Pedro Savarese, Itay Evron, Daniel Soudry, and Nathan Srebro. How do infinite width bounded norm networks look in function space? In Alina Beygelzimer and Daniel Hsu, editors, *Proceedings of the Thirty-Second Conference on Learning Theory*, volume 99 of *Proceedings of Machine Learning Research*, pages 2667–2690. PMLR, 25–28 Jun 2019.

[195] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014.

[196] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. A mathematical theory of semantic development in deep neural networks. *Proceedings of the National Academy of Sciences*, 116(23):11537–11546, 2019.

[197] Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. 2017.

[198] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[199] Vaishaal Shankar, Alex Fang, Wenshuo Guo, Sara Fridovich-Keil, Ludwig Schmidt, Jonathan Ragan-Kelley, and Benjamin Recht. Neural kernels without tangents. *arXiv preprint arXiv:2003.02237*, 2020.

[200] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[201] M.-H. Herman Shen and Liang Chen. A new CGAN technique for constrained topology design optimization. *CoRR*, abs/1901.07675, 2019.

[202] Ole Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33:401–424, 04 2007.

[203] J.W. Silverstein. Strong convergence of the empirical distribution of eigenvalues of large dimensional random matrices. *Journal of Multivariate Analysis*, 55(2):331 – 339, 1995.

[204] Berfin Simsek, François Ged, Arthur Jacot, Francesco Spadaro, Clement Hongler, Wulfram Gerstner, and Johanni Brea. Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9722–9732. PMLR, 18–24 Jul 2021.

[205] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks. *arXiv preprint arXiv:1805.01053*, 2018.

[206] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *CoRR*, abs/2006.09661, 2020.

[207] Ivan Sosnovik and Ivan V. Oseledets. Neural networks for topology optimization. *CoRR*, abs/1709.09578, 2017.

[208] Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

[209] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[210] Roland Speicher. Free probability and random matrices. In *Free Probability and Random Matrices*, 2017.

[211] Stefano Spigler, Mario Geiger, Stéphane d'Ascoli, Levent Sagun, Giulio Biroli, and Matthieu Wyart. A jamming transition from under-to over-parametrization affects loss landscape and generalization. *arXiv preprint arXiv:1810.09665*, 2018.

[212] Karthik Sridharan, Shai Shalev-Shwartz, and Nathan Srebro. Fast rates for regularized objectives. In *Advances in neural information processing systems*, pages 1545–1552, 2009.

[213] Bharath Sriperumbudur and Zoltán Szabó. Optimal rates for random fourier features. In *Advances in Neural Information Processing Systems*, pages 1144–1152, 2015.

[214] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 7537–7547. Curran Associates, Inc., 2020.

[215] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[216] Luca Venturi, Afonso Bandeira, and Joan Bruna. Neural networks with finite intrinsic dimension have no spurious valleys. *arXiv preprint arXiv:1802.06384*, 2018.

[217] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[218] H. Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM J. Sci. Comput.*, 13:631–644, 1992.

[219] Daniel Wagenaar. Information geometry of neural networks. 1998.

[220] Christopher Williams and Matthias Seeger. The effect of the input density distribution on kernel-based classifiers. In *Proceedings of the 17th international conference on machine learning*. Citeseer, 2000.

[221] Blake Woodworth, Suriya Gunasekar, Pedro Savarese, Edward Moroshko, Itay Golan, Jason Lee, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models, 2020.

[222] Lei Wu, Zhanxing Zhu, and Weinan E. Towards Understanding Generalization of Deep Learning: Perspective of Loss Landscapes. *CoRR*, abs/1706.10239, 2017.

[223] Sitao Xiang and Hao Li. On the effects of batch and weight normalization in generative adversarial networks. *arXiv preprint arXiv:1704.03971*, 2017.

[224] Lechao Xiao, Jeffrey Pennington, and Samuel Schoenholz. Disentangling trainability and generalization in deep neural networks. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10462–10472. PMLR, 13–18 Jul 2020.

[225] Lechao Xiao, Jeffrey Pennington, and Samuel S. Schoenholz. Disentangling trainability and generalization in deep learning. *CoRR*, abs/1912.13053, 2019.

[226] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

[227] Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *International Conference on Neural Information Processing*, pages 264–274. Springer, 2019.

[228] Greg Yang. Scaling Limits of Wide Neural Networks with Weight Sharing: Gaussian Process Behavior, Gradient Independence, and Neural Tangent Kernel Derivation. *arXiv e-prints*, page arXiv:1902.04760, Feb 2019.

[229] Greg Yang and Edward J. Hu. Feature learning in infinite-width neural networks, 2020.

[230] Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. A mean field theory of batch normalization. *CoRR*, abs/1902.08129, 2019.

[231] Greg Yang and Samuel Schoenholz. Mean field residual networks: On the edge of chaos. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 7103–7114. Curran Associates, Inc., 2017.

[232] Tianbao Yang, Yu-Feng Li, Mehrdad Mahdavi, Rong Jin, and Zhi-Hua Zhou. Nyström method vs random Fourier features: A theoretical and empirical comparison. In *Advances in neural information processing systems*, pages 476–484, 2012.

[233] Luzhong Yin and Wei Yang. Optimality criteria method for topology optimization under multiple constraints. *Computers and Structures*, 79(20):1839–1850, 2001.

[234] Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. In *Advances in Neural Information Processing Systems*, pages 1975–1983, 2016.

[235] Chulhee Yun, Shankar Krishnan, and Hossein Mobahi. A unifying view on implicit bias in training linear neural networks. In *International Conference on Learning Representations*, 2021.

[236] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *ICLR 2017 proceedings*, Feb 2017.

[237] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018.

[238] Tong Zhang. Effective dimension and generalization of kernel learning. In *Advances in Neural Information Processing Systems*, pages 471–478, 2003.

[239] Yuchen Zhang, John Duchi, and Martin Wainwright. Divide and conquer kernel ridge regression: A distributed algorithm with minimax optimal rates. *The Journal of Machine Learning Research*, 16(1):3299–3340, 2015.

Arthur Jacot

Nationality: Swiss
E-mail: arthur.jacot@netopera.net
Website: https://sites.google.com/view/arthurjacot/

Education

2018-2022 **PhD in Mathematics on the Theory of Deep Learning**
École Polytechnique Fédérale de Lausanne
under the supervision of Prof. Clément Hongler.

2015-2017 **M.Sc. in Mathematics with a minor in Computational Neurosciences**
École Polytechnique Fédérale de Lausanne.

2011-2015 **B.Sc. in Mathematics with a minor in Computer Science**
Freie Universität Berlin.

Work Experience

2017-2018 **Civil Service - Science Activity Leader**
Animascience, Geneva
Creation of scientific activities, math puzzles, and more.

2016-2017 **Substitute Teacher in Mathematics**
Département de l'Instruction Publique (DIP), Geneva.

2014 **Internship as Programmer/Musician**
Studio for Electro-Instrumental Music (STEIM), Amsterdam
Developed RoSa, an audio live sampling program in C++.

Publications (Google Scholar)

1. *Feature Learning in L_2-regularized DNNs: Attraction/Repulsion and Sparsity*, Arthur Jacot, Eugene Golikov, Clément Hongler, Franck Gabriel, 2022. [arXiv link]
2. *Deep Linear Network Dynamics: Low-Rank Biases Induced by Initialization Scale and L2 Regularization*, Arthur Jacot, François Ged, Franck Gabriel, Berfin Şimşek, Clément Hongler, 2022. [arXiv link]
3. *DNN-Based Topology Optimization: Spatial Invariance and Neural Tangent Kernel*, Benjamin Dupuis, Arthur Jacot, NeurIPS 2021. [conference paper]
4. *Geometry of the Loss Landscape in Overparameterized Neural Networks: Symmetries and Invariances*, Berfin Şimşek, François Ged, Arthur Jacot, Francesco Spadaro, Clément Hongler, Wulfram Gerstner, Johanni Brea, ICML 2021. [conference paper]
5. *Kernel Alignment Risk Estimator: Risk Prediction from Training Data*, Arthur Jacot, Berfin Şimşek, Francesco Spadaro, Clément Hongler, Franck Gabriel, NeurIPS 2020. [conference paper]

6. *Implicit regularization of Random Feature Models*, Arthur Jacot, Berfin Şimşek, Francesco Spadaro, Clément Hongler, Franck Gabriel, ICML 2020. [conference paper]

7. *The asymptotic spectrum of the Hessian of DNN throughout training*, Arthur Jacot, Franck Gabriel, Clément Hongler. ICLR 2020. [conference paper]

8. *Order and Chaos: NTK views on DNN Normalization, Checkerboard and Boundary Artifacts*, Arthur Jacot, Franck Gabriel, François Ged, Clément Hongler, to appear at MSML 2022. [arXiv link]

9. *Disentangling feature and lazy learning in deep neural networks: an empirical study*, Mario Geiger, Stefano Spigler, Arthur Jacot, Matthieu Wyart, Journal of Statistical Mechanics: Theory and Experiment 2020. [arXiv link]

10. *Scaling description of generalization with number of parameters in deep learning*, Mario Geiger, Arthur Jacot, Stefano Spigler, Franck Gabriel, Levent Sagun, Stéphane d'Ascoli, Giulio Biroli, Clément Hongler, Matthieu Wyart, Journal of Statistical Mechanics: Theory and Experiment 2020. [arXiv link]

11. *Neural Tangent Kernel: Convergence and Generalization in Neural Networks*, Arthur Jacot, Franck Gabriel, Clément Hongler, NeurIPS 2018. [conference paper]

Prizes

1. 2021 SwissMAP Innovator Prize.

2. 2022 Google PhD Fellowship (declined).

Talks

May 2022 Workshop 'New Interactions Between Statistics and Optimization' at BIRS.

Sep. 2021 SwissMAP General Meeting, Les Diablerets, Switzerland.

Feb. 2021 Seminar at RWTH Chair for Mathematics of Information Processing, RWTH Aachen University, Germany (online).

Feb. 2021 4th Mini-workshop on Deep Learning Theory, Huawei Beijing, China (online).

Aug. 2020 Mathematics of Machine Learning Seminar, University of California, Los Angeles, USA (online).

July 2020 Online Summer School of Deep Learning Theory, Shanghai Jiao Tong University, China (online).

May 2020 Data Science Seminar, Shanghai Jiao Tong University, China (online).

Mar. 2020 DeepMind London, UK.

Feb. 2020 Statistics Seminar, University of Oxford, UK.

Feb. 2020 Neural Net Theory Group, École Polytechnique Fédérale de Lausanne, Switzerland.

Oct. 2019 Google Brain, Mountain View, USA.

Oct. 2019 Analyses of Deep Learning, Stanford University, USA.

Aug. 2019 Theoretical Advances in Deep Learning Workshop, Istanbul, Turkey.

Mar. 2019 CRiSM day on Bayesian Intelligence, Warwick University, UK.

Apr. 2019 Seminar in Probability: Theory of Deep Learning, Universität Basel, Switzerland.

Dec. 2018 Spotlight Presentation, NeurIPS 2018, Montréal, Canada.

Skills and Other Activities

- Languages: French (native), English (fluent), German (fluent), Spanish (beginner).
- Programming: Python, Pytorch, Haskell, Scala, C, C++, Java.
- Music: Singer and bassist in two bands, piano.