EPFL

Thèse n° 7437

On Vacuous and Non-Vacuous Generalization Bounds for Deep Neural Networks.

Présentée le 12 novembre 2020

à la Faculté des sciences et techniques de l'ingénieur
Laboratoire de traitement des signaux 2
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Konstantinos PITAS

Acceptée sur proposition du jury

Prof. F. Rachidi-Haeri, président du jury
Prof. P. Vandergheynst, directeur de thèse
Prof. A. Wilson, rapporteur
Prof. M. A. T. Figueiredo, rapporteur
Prof. M. Jaggi, rapporteur

École polytechnique fédérale de Lausanne

2020

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.
— T.S. Elliot

To Evi, Stergios, Sotiris, Tatiana, Orionas and Michalis for their help, when I needed it most.

Acknowledgements

First of all I would like to thank Prof. Pierre Vandergheynst for giving me the opportunity to complete my PhD studies at the LTS2.

I would like to thank my committee, Prof. Andrew Gordon Wilson, Prof. Mario Figueiredo, and Prof. Martin Jaggi, for reading this thesis and giving me feedback on this work. I would also like to thank Prof. Farhad Rachidi for acting as the president of my PhD jury.

My stay at the LTS floor wouldn't have been as interesting without meeting Volodymyr, Youngjoo, Rodrigo, Andreas, Beril, Hermina, Pinar, Vaggelis and Irene. Hikes, barbecues, and babyfoot were always a good excuse to stop working and take a much needed break.

I would like to thank our secretary, Anne, for always being helpful and positive, solving administrative problems before they appeared.

Marie, Miguelito, Gabriel and Miroslav were always there when I needed a break on Friday night.

Finally in Switzerland I made a number of new friends. Evi, Orionas, Stergios, Sotiris, Tatiana, Michalis, Lynn, Aster, Lydia, Nico, Effie, Yannis, Aliki, Jacomo, Eva, Ilya: the past 6 years wouldn't have been the same without you!

Lausanne, October 29, 2020 Konstantinos Pitas

Abstract

Deep neural networks have been empirically successful in a variety of tasks, however their theoretical understanding is still poor. In particular, modern deep neural networks have many more parameters than training data. Thus, in principle they should overfit the training samples and exhibit poor generalization to the complete data distribution. Counter intuitively however, they manage to achieve both high training accuracy and high testing accuracy. One can prove generalization using a validation set, however this can be difficult when training samples are limited and at the same time we do not obtain any information about *why* deep neural networks generalize well. Another approach is to estimate the complexity of the deep neural network. The hypothesis is that if a network with high training accuracy has high complexity it will have memorized the data, while if it has low complexity it will have learned generalizable patterns.

In the first part of this thesis we explore Spectral Complexity, a measure of complexity that depends on combinations of norms of the weight matrices of the deep neural network. For a dataset that is difficult to classify, with no underlying model and/or no recurring pattern, for example one where the labels have been chosen randomly, spectral complexity has a large value, reflecting that the network needs to *memorize* the labels, and will not generalize well. Putting back the real labels, the spectral complexity becomes lower reflecting that some structure is present and the network has learned patterns that might generalize to unseen data. Spectral complexity results in vacuous estimates of the generalization error (the difference between the training and testing accuracy), and we show that it can lead to counterintuitive results when comparing the generalization error of different architectures.

In the second part of the thesis we explore non-vacuous estimates of the generalization error. In Chapter 2 we analyze the case of PAC-Bayes where a posterior distribution over the weights of a deep neural network is learned using stochastic variational inference, and the generalization error is the KL divergence between this posterior and a prior distribution. We find that a common approximation where the posterior is constrained to be Gaussian with diagonal covariance, known as the mean-field approximation, limits significantly any gains in bound tightness. We find that, if we choose the prior mean to be the random network initialization, the generalization error estimate tightens significantly.

In Chapter 3 we explore an existing approach to *learning* the prior mean, in PAC-Bayes, from the training set. Specifically, we explore differential privacy, which ensures that the training samples contribute only a limited amount of information to the prior,

making it distribution and not training set dependent. In this way the prior should generalize well to unseen data (as it hasn't memorized individual samples) and at the same time any posterior distribution that is close to it in terms of the KL divergence will also exhibit good generalization.
Keywords: Complexity, Generalization Error, Spectral Complexity, PAC-Bayes, Differential Privacy, Invariance, Flat Minima, Algorithmic Stability.

Résumé

Les réseaux neuronaux profonds sont empiriquement efficaces dans une variété de tâches, mais leur compréhension théorique est encore faible. En particulier, les réseaux neuronaux profonds modernes ont beaucoup plus de paramètres qui peuvent être appris que de données dans l'ensemble des échantillons d'entraînement. Ainsi, en principe, ils devraient surapprendre les échantillons d'entraînement et présenter une faible généralisation sur la distribution complète des données. Mais, contre toute attente, ils parviennent à atteindre à la fois une haute précision sur la classification de l'ensemble d'entrainement et de test. On peut prouver la généralisation en utilisant un ensemble de validation, mais cela peut être difficile lorsque les échantillons d'entraînement sont limités et , dans le même temps, cela n'apporte aucune information sur *pourquoi* les réseaux neuronaux profonds se généralisent bien. Une autre approche consiste à estimer la complexité du réseau neuronal profond. L'hypothèse est que si un réseau à faible risque empirique a une complexité élevée, il aura mémorisé les données, tandis que s'il a une faible complexité, il aura appris des modèles généralisables.

Dans la première partie de cette thèse, nous explorons la complexité spectrale, une mesure de la complexité qui dépend des combinaisons de normes des matrices de poids du réseau neuronal profond. Pour un ensemble de données qui est difficile à classifier, sans modèle sous-jacent ou sans motif de données recurrent, par exemple un ensemble dont les étiquettes ont été choisies au hasard, la complexité spectrale a une grande valeur, reflétant le fait que le réseau doit *mémoriser* les étiquettes, et ne généralisera pas bien. Inversement, Quand l'ensemble de données retrouve ses véritables étiquettes, la complexité spectrale devient faible, ce qui reflète le fait qu'une certaine structure est présente et que le réseau a appris des modèles qui pourraient se généraliser à des données invisibles. La complexité spectrale donne lieu à des estimations vides de généralisation, et nous montrons qu'elle peut conduire à des résultats contre-intuitifs lorsque l'on compare l'erreur de généralisation de différentes architectures.

Dans la deuxième partie de la thèse, nous explorons les estimations non vides de la généralisation. Dans le chapitre 2, nous analysons le cas de PAC-Bayes où une distribution postérieure sur les poids d'un réseau neuronal profond est apprise en utilisant l'inférence variationnelle stochastique, et l'erreur de généralisation est la divergence KL entre cette distribution postérieure et une distribution antérieure. Nous constatons qu'une approximation commune où le postérieur est contraint d'être gaussien avec covariance diagonale, connue sous le nom d'approximation du champ moyen, limite de manière

significative tout gain sur la précision des bornes. En revanche, nous constatons que si l'on choisit l'hypothèse comme l'initialisation du réseau neuronal profond aléatoire, l'estimation de la généralisation se resserre considérablement.

Au chapitre 3, nous explorons une approche existante pour *apprendre* l'hypothèse dans PAC-Bayes à partir de l'ensemble de formation. Plus précisément, nous examinons la confidentialité différentielle, qui garantit que les échantillons d'apprentissage n'apportent qu'une quantité limitée d'informations à l'hypothèse, ce qui la rend dépendante de la distribution et non de l'ensemble d'apprentissage. De cette façon, l'hypothèse devrait bien se généraliser aux données non vues (car le réseau n'a pas mémorisé les échantillons individuels) et en même temps toute distribution postérieure qui lui est proche en termes de divergence de KL présentera également une bonne généralisation.
Mots-clés : Complexité, erreur de généralisation, compétitivité spectrale, PAC-Bayes, confidentialité différentielle, invariance, minimum plat, stabilité algorithmique.

Contents

Introduction

The success of deep learning contradicts a number of common machine learning intuitions. Deep learning models have a highly non-convex optimization landscape which, in principle, should present significant difficulties in finding even a good local minimum, i.e. one should have difficulties in obtaining good *training* accuracy (Allen-Zhu et al., 2019; Du et al., 2019). Deep learning models also include a significantly larger number of parameters than the available training data. Thus, even if they managed to achieve good training accuracy, based on the bias-variance trade-off (Bishop, 2006), they should in principle overfit, memorize the training data and have low *testing* accuracy. Nevertheless, deep neural networks manage to achieve both high training *and* testing accuracy, and have evolved from a scientific curiosity to the dominant paradigm in a number of modern machine learning tasks (Bengio et al., 2013). At the same time, as the field has matured, it has been applied to real world applications, such as healthcare (Esteva et al., 2017), finance (Nelson et al., 2017), and autonomous driving (Kendall and Gal, 2017), with interesting new challenges in terms of estimating the risks involved and striving for ever more efficient use of available data.

In this thesis we will explore the challenges and possible solutions to the generalization puzzle of deep learning. In particular, assuming that a supervised deep learning model can perfectly (or near perfectly) fit the training data we are interested in i) finding measures that predict whether the model will perform well on testing data, possibly giving some intuition into ii) *why* common deep neural networks manage to generalize well. We will see why traditional measures of complexity are not well suited for the deep learning setting. We will then explore a number of recent advances, that culminate with non-vacuous generalization bounds for a number of small and intermediate networks.

We denote the learning sample $(X, Y) = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, that contains n input-output pairs. Samples (X, Y) are assumed to be sampled randomly from a distribution $\mathcal{D}$. Thus, we denote $(X, Y) \sim \mathcal{D}^n$ the i.i.d observation of n elements. In this introductory chapter we will refer also to a validation set $(X_{\text{val}}, Y_{\text{val}})$ of m samples which is defined in a similar way as the learning sample. We consider loss functions $\ell : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{F}$ is a set of predictors $f : \mathcal{X} \to \mathcal{Y}$. We also denote the empirical risk $\hat{\mathcal{L}}^{\ell}_{X,Y}(f) = (1/n) \sum_i \ell(f, \boldsymbol{x}_i, y_i)$ and the risk $\mathcal{L}^{\ell}_{\mathcal{D}}(f) = \mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \ell(f, \boldsymbol{x}, y)$.

In the following we will present a number of useful relations in an informal manner, and will then make the statements formal in the main body of the thesis.

In general, the problem we are interested in, is bounding the risk

$$\mathcal{L}^{\ell}_{\mathcal{D}}(f) \leq \text{constant}.$$

By far, the most common way of bounding the above in the context of deep neural networks, is by use of a validation set (Langford, 2005; Kääriäinen and Langford, 2005). One first trains a predictor f using a training set (X, Y), and then computes a validation risk $\hat{\mathcal{L}}^{\ell}_{X_{\text{val}},Y_{\text{val}}}(f)$. For m validation samples this can be readily turned into a bound on the risk $\mathcal{L}^{\ell}_{\mathcal{D}}(f)$, using a tail bound on the corresponding binomial distribution (Langford, 2005). However, this approach has some shortcomings. For one it requires a significant number of samples. For a deep neural network with 0% validation risk to get an estimate $\mathcal{L}^{\ell}_{\mathcal{D}}(f) \leq 0.0029$ with high probability, one requires $m \geq 1000$ validation samples. This can be a problem in that these samples cannot be used for training, possibly hurting the performance of the deep network. At the same time, for a number of fields such as healthcare, the cost of obtaining validation samples can be prohibitively high (Davenport and Kalakota, 2019). Finally, even though we can prove that the true risk will be low, we do not get any information about the reason *why* the classifier performs well in the first place.

As such, researchers often use the empirical risk (on the training set) together with the *complexity* (Mohri et al., 2018) of the classifier to derive bounds roughly of the form

$$\mathcal{L}^{\ell}_{\mathcal{D}}(f) \leq \hat{\mathcal{L}}^{\ell}_{X,Y}(f) + \text{complexity},$$

and the difference $\mathcal{L}^{\ell}_{\mathcal{D}}(f) - \hat{\mathcal{L}}^{\ell}_{X,Y}(f)$ is what we will call in the following the *generalization error* (GE) of the classifier. Intuitively the more complex the classifier the more it is possible for it to have simply memorized the training data, and to not have learned any discriminative patterns, leading to high true risk. Complexity can be measured in a number of different ways, and the above intuition about memorization immediately presents difficulties for traditional complexity measures such as Rademacher complexity (Mohri et al., 2018) and VC dimension (Blumer et al., 1989). The empirical Rademacher complexity is defined as

$$\hat{\mathfrak{R}}_n(\mathcal{F}) = \mathbf{E}_{\sigma}\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=0}^{n} \sigma_i f(\boldsymbol{x}_i)\right]$$

where $\sigma_1, \cdots, \sigma_n \in \{\pm 1\}$ are i.i.d Rademacher random variables with $\mathbb{P}\{\sigma_i = +1\} = \mathbb{P}\{\sigma_i = -1\} = 1/2$. As such, given the supremum, Rademacher complexity can be seen as a measure of how well a classifier can fit a random binary labelling over the training set (Zhang et al., 2016). Unfortunately, for the case of deep neural networks it has been

shown empirically that most common architectures can fit a random binary labelling perfectly over the training set (Zhang et al., 2016), therefore this would imply that $\hat{\mathfrak{R}}_n(\mathcal{F}) = 1$. At the same time a bound on the true risk should be better than random chance to be non-vacuous (for a 10-class problem we already know that $\mathcal{L}^{\ell}_{\mathcal{D}}(f) \leq 0.9$). This leads to bounds based on Rademacher complexity to be vacuous. Similar arguments exist for the VC dimension (Zhang et al., 2016). If modern deep neural networks have enough capacity to memorize random labels they could in principle memorize also *real* labels, therefore overfitting in all problems of interest.

Based on the above results, researchers focused on complexity measures which are data-dependent (Golowich et al., 2017; Arora et al., 2018; Neyshabur et al., 2017b; Sokolić et al., 2016; Bartlett et al., 2017; Dziugaite and Roy, 2017), meaning that they measure the complexity of a deep neural network based on the specific weights that were computed for it using stochastic gradient descent. This allows some room to maneuver. As weights change when optimizing for different datasets, one could in principle identify a quantity that is large for memorization problems such as fitting a random labelling, but that is small for real labels, where the deep network will generalize well. Thus, in principle the network has enough capacity to memorize the labels, but *in practice* architectural biases and the optimization procedure will force it to choose the simplest possible fit, if the learning problem has any structure. Indeed, researchers derived such measures of complexity based on combinations of various norms of the weight matrices (Golowich et al., 2017; Arora et al., 2018; Neyshabur et al., 2017b; Sokolić et al., 2016; Bartlett et al., 2017). We illustrate the intuition behind this, using a simplified example. Assume a one layer fully connected network $f(\boldsymbol{x}) = \max(\mathbf{W}\boldsymbol{x}, 0)$, for example it is obvious that a network with $||\mathbf{W}||_2 > 0$ will include more information about the training set than one with $||\mathbf{W}||_2 = 0$, and one could hypothesize that $||\mathbf{W}_1||_2 > ||\mathbf{W}_2||_2$ might imply that $f_{\mathbf{W}_2}$ generalizes better than $f_{\mathbf{W}_1}$, if they have the same empirical risk.

This is the starting point of this thesis.

In Chapter 1 we elaborate on some weak points of norm based generalization bounds. We focus on a prominent class called "spectral complexity" (Bartlett et al., 2017), however other norm based bounds are quite similar. Spectral complexity based bounds remain vacuous by several orders of magnitude (Arora et al., 2018) and are validated by showing that they correlate empirically with generalization error (Arora et al., 2018; Bartlett et al., 2017). However, we can easily construct datasets with an increasing number of translations and elastic deformations, such that spectral complexity remains constant while the generalization error changes. As such, we advocate that the invariance properties of deep neural networks should be looked at in more detail when constructing generalization bounds, and that the criteria for validating spectral complexity have to be stated with more precision. In the second part of the chapter, we discuss how the vacuity of the bounds can lead to counterintuitive results, if used to compare different architectures, specifically locally connected and convolutional networks. These results

prescribe caution when using vacuous measures of complexity to select classification models. Finally, we review other criticisms of vacuous bounds.

In Chapter 2, motivated by the shortcomings of vacuous bounds, we turn to analyzing bounding techniques that have resulted in non-vacuous complexity estimates. Specifically, we focus on the PAC-Bayes framework (McAllester, 1999; Seeger, 2002; Catoni, 2007; Alquier et al., 2016) which deals with randomized classifiers, and has been applied to deep neural networks in Dziugaite and Roy (2017); Zhou et al. (2018). Here complexity is modeled roughly as

$$\mathrm{KL}(\hat{\rho}||\pi) = (1/\lambda)||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi}||_2^2$$

with posterior and prior distributions over the *weights* of a deep neural network as $\hat{\rho} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda\mathbf{I})$ and $\pi = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\mathbf{I})$. There are two factors in the complexity model. The first, is the variance of the noise λ, the more noise that we can add ($\lambda \to +\infty$) without hurting accuracy, the more *flat* is the minimum of the classifier, and the better it will generalize (at the same time the measurable complexity will tend to 0, $\mathrm{KL}(\hat{\rho}||\pi) \to 0$). The second is the ℓ_2 distance between the deep neural network weights $\boldsymbol{\mu}_{\hat{\rho}}$ and the prior mean $\boldsymbol{\mu}_{\pi}$. As this approach yields non-vacuous bounds, we analyze the contribution of each factor and find that most gains are the result of Dziugaite and Roy (2017) choosing the prior mean to be the random initialization of the neural network such that $||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi}||_2$ is low. Regarding choosing the variance λ, we showcase how better approximations of the curvature at the mimimum can result in adding more noise to weights without hurting accuracy, resulting in tighter bounds.

Given that the distance between the prior and posterior means is empirically important, in Chapter 3 we discuss techniques for computing informative prior means $\boldsymbol{\mu}_{\pi}$, such that $||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi}||_2 \leq ||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\text{init}}||_2$, so that $\mathrm{KL}(\hat{\rho}||\pi)$ is tightened further. A major bottleneck in computing informative priors is that, in PAC-Bayes bounds, the prior cannot depend on the training data, as having computed $\hat{\rho}$, one could then set $\pi = \hat{\rho}$ so that trivially $\mathrm{KL}(\hat{\rho}||\pi) = 0$. However the prior *can* depend on the data distribution. We review existing results (Dziugaite and Roy, 2018a) where one can learn valid prior means $\boldsymbol{\mu}_{\pi}$ from training data by enforcing differential privacy (Dwork, 2011; Dwork et al., 2014; Dwork, 2008) constraints. Intuitively, privacy constraints imply that every data point has only contributed limited information to the mean $\boldsymbol{\mu}_{\pi}$ so that the mean is distribution and not training set dependent. By using existing state of the art differential privacy techniques of Abadi et al. (2016), we obtain even tighter non-vacuous bounds than the ones in Chapter 2. We then discuss inherent tradeoffs in learning classifiers with both low complexity (low mutual information between parameters and training data), and low empirical risk.

Motivation for the analyzed techniques. We elaborate more on the motivation behind analyzing the techniques from Chapters 2 and 3 and for considering them as promising candidates for proving generalization. We will see that both approaches are

based on conventional wisdom about deep neural network *optimization*. In fact they consist of quantitative formulations of heuristic choices which have be shown empirically to result in better generalization.

The PAC-Bayes approach of Chapter 2 is closely related to "flat minima". There has been a wealth of literature (Hochreiter and Schmidhuber, 1997b; Kleinberg et al., 2018; Keskar et al., 2016; Li et al., 2018a; Jastrzębski et al., 2017) linking generalization and flat minima going back to pioneering work by Hochreiter and Schmidhuber (1997a). Of particular importance is the work of Keskar et al. (2016) where the authors, using detailed experiments, show that flatness of minima correlates strongly with better generalization. This corresponds to a popular heuristic which is widely used in practice: setting the batch size of stochastic gradient descent to be small (You et al., 2017a; Masters and Luschi, 2018; You et al., 2017b). To see why this heuristic is linked to flat minima note that using a stochastic version of gradient descent introduces noise to the gradient estimate. Adding noise to the gradient estimate results in turn to small "jumps" around the optimization landscape, which should force the optimization procedure to end only in regions of the loss that are flat. Empirically the above is often confirmed in practice (Keskar et al., 2016), small batch sizes result in flat minima and better generalization. There is however a significantly computation difficulty in estimating the curvature around a given minimum explicitly (Neyshabur et al., 2017a). The PAC-Bayesian approach of Chapter 2 can then be seen as i) providing an implicit measure of flatness (robustness of the empirical risk to noise added to the parameters) that can be computed efficiently using sampling and ii) a formal relation between this flatness measure and generalization error.

Rather than looking at the geometry of favorable minima, on a more fundamental level using Stochastic Gradient Descent as opposed to Gradient Descent limits the ammount of information that each gradient update adds to the learned weights. In fact, explicitly adding noise to the gradient updates oftain leads to better empirical performance (Srivastava et al., 2014; Kingma et al., 2015; Neelakantan et al., 2015). Dropout (Srivastava et al., 2014) is one such approach that has proven extremely popular, and that has been introduced explicitly with the motivation to limit the information flow to each neuron. At the same time measuring the information flow at Stochastic Gradient Descent updates is difficult (Belghazi et al., 2018; Gabrié et al., 2018) and works such as Srivastava et al. (2014) provide only qualitative arguments. The techniques discussed in Chapter 3 deal with tightly measuring the information added by each gradient update and relating this to a generalization bound.

We return now to the principal question: "*why deep neural networks generalize well?*". In short, the present thesis provides further evidence regarding the hypothesis that noise added by stochastic gradient descent plays a crucial role in generalization, by limiting the amount of information in the learned weights.

What we can and can't do. We note here that the techniques presented above are not prescriptive for model construction, that is they are not informative to choosing *architectural* biases that are important for good generalization. In fact our hypothesis set throughout this thesis is not all deep neural networks. Rather, most comparisons are with regards to a *single* deep neural network architecture, then different hypothesis sets correspond to how many weight instantiations a complexity measure considers. Thus, given a fixed architecture, traditional measures of complexity such as Rademacher complexity and VC dimension consider a hypothesis set that consists of *all* different instantiations of the weights using real numbers. The techniques of Chapters 1-3 consider hypothesis sets that are greatly restricted. In Chapter 1 the hypothesis set is restricted so that the norms of the weights of each layer are bounded. In Chapters 2 and 3 the hypothesis set is restricted by modeling a posterior Gaussian distribution over the weights. Thus not all weight instantiations are equally probable, but by centering the posterior mean to be a minimum of the loss landscape the hypothesis set is biased towards instantiations that have low empirical risk.

In the context of deep neural networks finding appropriate architectures for a given task automatically is called Neural Architecture Search (NAS) (Zoph and Le, 2016; Liu et al., 2018; Pham et al., 2018; Elsken et al., 2018). To the best of the authors knowledge there has been only one work that proposes a generalization bound explicitly for the task of neural architecture search (Cortes et al., 2017) with applications only in small scale experiments. More fundamentally, predicting architectural biases that correspond to the symmetries of the training data and predicting the possible generalization performance of such an architecture using a generalization bound would require incorporating in a bound a model of the generating data distribution. Note how this is beyond the scope of most current generalization bounds (Lyle et al., 2020; Sokolic et al., 2016; Achille and Soatto, 2018), which only make the i.i.d assumption on the training and testing data.

Finally the results in this thesis, while more limited in scope than full NAS + parameter optimization, could play a crucial role in practice and can be mainly used as a substitute to validation sets. In fact typically one designs an architecture with some given biases and then wants to estimate the optimal weights such as the empirical risk is low and the network hasn't overfitted. This is where the presented techniques can be applied.

What we are and are not trying to do. One can be tempted to optimize the proposed bounds directly so as to reach a flatter minimum and obtain *better* generalization than with a deterministic neural network obtained by vanilla SGD (Germain et al., 2016; Biggs and Guedj, 2020; Dziugaite and Roy, 2017). Or can be motivated by Chapter 2 to simply set a posterior distribution over the weights, centered at a network minimum, so as to improve generalization by creating an implicit model ensemble (Maddox et al., 2019; Ritter et al., 2018). Some of these approaches work better than others and assessing their practicality in real problems is tricky. For example adding noise to already noisy SGD updates, while in theory beneficial, can in practice force the SGD to diverge (Blundell

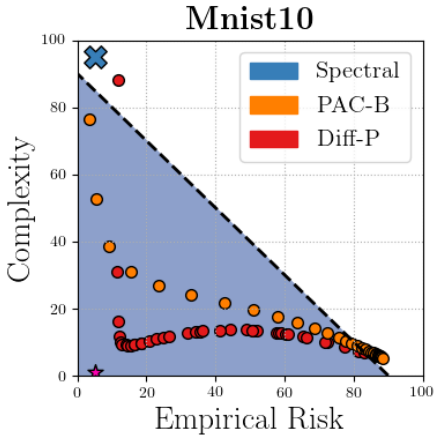
et al., 2015; Wu et al., 2018). However improving generalization, training or testing accuracy, is not the main goal of the thesis and indeed the goal of most of the relevant literature cited. In fact the main goal is *estimating* the generalization error for different weight instantiations of a single architecture. As this is difficult for single hypotheses (Kawaguchi et al., 2017), the aim is to create a sufficiently small hypothesis class which includes the original single hypothesis and for which proving generalization is tractable.

Relationship with other frameworks. There is a close link between PAC-Bayes and Minimum Length Description (MLD) framework (Grunwald, 2004; Barron et al., 1998; Rissanen, 1978). In brief the MLD principle seeks to decribe the training data in a classification task using the least bits possible. Thus it separates two sources of cost: i) the length, in bits, of the description of the classification model and ii) the length in bits of the description of the data when encoded with the help of the model. The principle states that the model that minimizes these two costs should be the one chosen for the classification task. The complexity estimated by PAC-Bayes together with the empirical risk at the given complexity level can be seen as a *variational encoding* of the deep neural network weights under the MLD principle. The variance of the noise in the posterior measures the level of precision used in the encoding (Blier and Ollivier, 2018; Daniely and Granot, 2019). The variational code results in an explicit coding scheme thanks to a *bits-back* argument (Honkela and Valpola, 2004; Hinton and van Camp, 1993).

The MLD can also be related to flatness of minima (Hochreiter and Schmidhuber, 1997a). Thus a neural network with weights that corresponds to a flat minimum, will tolerate significant noise to the parameters. Therefore the estimated codelength will be small for the MLD framework. To see why this is the case we have already seen that intuitively if parameters correspond to a flat minimum they can tolerate significant noise without hurting accuracy. Therefore a consequence is that they can intuitively be removed or compressed (stored with lower precision) which directly limits the codelength in the MLD framework. When one has to choose among minima that have the same empirical risk, choosing the flattest ensures that one chooses the model with the smallest codelength (in bits) in accordance with the MLD principle.

The thesis in a single figure. The techniques in Chapter 1 correspond to a single complexity estimate that depends on the norm of the deep neural network weights, however the techniques in Chapters 2 and 3 actually provide a range of classifiers with different complexity and empirical risk. For example a posterior $\hat{\rho} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda \mathbf{I})$ will correspond to different randomized classifiers for different λ, each with different empirical risk and complexity.

Thus in Chapters 2 and 3 we introduce "Risk-Complexity" plots (Figure 1). On the x-axis we plot the Empirical Risk $\hat{\mathcal{L}}^{\ell}_{X,Y}(f)$, while on the y-axis we plot the estimated Complexity $\mathrm{KL}(\hat{\rho}||\pi)$ or the equivalent complexity metric. The plots have a number of advantages. We can easily plot the region of non-vacuity and the location of the best possible bound



(a)

Figure 1 – **Risk-Complexity plot for MNIST 10**: The area below the dashed line corresponds to non-vacuous pairs of (complexity, empirical risk). The purple star corresponds to the true risk implied by the validation set. "Spectral" corresponds to the complexity measurement using spectral complexity. "PAC-B" corresponds to multiple (empirical risk,complexity) pairs that we derive in Chapter 2 using the PAC-Bayesian approach. "Diff-P" corresponds to multiple (empirical risk,complexity) pairs that we obtain when learning an informative prior mean in Chapter 3. Throughout the thesis we thus go from a single complexity estimate that is vacuous to multiple non-vacuous (empirical risk,complexity) pairs that get progressively tighter. We also derive a number of intuitions along the way.

implied by the *validation* set. For example, looking at Figure 1 and remembering that the general form of the bounds is $\mathcal{L}_{\mathcal{D}}^{\ell}(f) \leq \hat{\mathcal{L}}_{X,Y}^{\ell}(f) + \text{complexity}$, if the empirical risk of a classifier is 20% we should estimate it's complexity as at most 70%. Otherwise, we would be bounding the true risk with an upper bound looser than $\mathcal{L}_{\mathcal{D}}^{\ell}(f) \leq 90\%$, this would a bound worse than random for a 10 class classification problem, and would be therefore vacuous. For any bounding method that provides multiple classifiers, we can then derive multiple (complexity, empirical risk) estimates and plot a front of all combinations. This results in an intuitive way for comparing bounding methods, where one can simply inspect the fronts in relation to the best possible bound implied by the validation set.

In Figure 1 we plot the empirical risk and complexity estimates from different techniques from Chapters 1 to 3, for the case of MNIST-10 with 50000 training and 10000 validation samples. We use a simple fully connected network with two hidden layers

$$\text{input} \rightarrow 300\text{FC} \rightarrow 300\text{FC} \rightarrow \#\textit{classes}\text{FC} \rightarrow \text{output}$$

where xFC denotes a fully connected layer with x neurons, the first two layers are non-linear with a rectifier non-linearity, while the final one is followed by a softmax

function. MNIST-10 is a simple classification problem, as such, both the empirical risk and the true risk implied by the validation set are close to 0. We plot the implied best possible (empirical risk,complexity) pair, using a purple star.

We then plot bounds for this problem, using techniques from Chapters 1-3. Spectral complexity from Chapter 1, provides a single complexity estimate that is vacuous. A simple PAC-Bayesian baseline $\hat{\rho} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda\mathbf{I})$ and $\pi = \mathcal{N}(\boldsymbol{\mu}_{\text{init}}, \lambda\mathbf{I})$ from Chapter 2, where the prior mean is the random neural network initialization, results in non-vacuous bounds. Further learning an prior prior mean using differential privacy in Chapter 3, results in even tighter bounds. The main takeaway is that we go from vacuous bounds to non-vacuous ones, while at the same time we discuss a number related issues, such as which of these bounds can be used for model selection, as well as what properties of the loss landscape and the optimization procedure are relevant to good generalization.

Notation Bold uppercase letters (e.g. $\mathbf{A}$) denote matrices, bold lowercase letters (e.g. $\mathbf{a}$) denote vectors, calligraphic letters (e.g. $\mathcal{A}$) denote sets, $\mathbf{E}_a$ denotes the expectation over the random variable a, $\mathcal{D}$ denotes the data distribution, for $\hat{\rho}$ and π probability measures over $\mathcal{X}$, $\text{KL}(\hat{\rho}||\pi)$ is the KL divergence $\text{KL}(\hat{\rho}||\pi) = \int_{\mathcal{X}} \log(\frac{d\hat{\rho}}{d\pi}) d\hat{\rho}$.

Vacuous Bounds Part I

1 Spectral Complexity and Invariance.

In the introduction we saw how, for deep neural networks, any measure of model complexity which is uniform across all functions representable by a given architecture, such as Rademacher complexity and VC dimension, is doomed to provide contradictory measurements (Bartlett et al., 2017; Arora et al., 2018; Neyshabur et al., 2015). We also described how this motivated researchers to consider measures of complexity that allow, given a specific deep neural network architecture, for high complexity models for difficult datasets (random labels) and low complexity models for easier datasets (real labels).

1.1 Introduction

In this chapter we focus on analyzing one such popular class of complexity measures, *spectral complexity* (Bartlett et al., 2017) normalized by the margin. Spectral complexity consists usually of the combinations of the spectral or other norms of the different layer weight matrices. The average margin quantifies the confidence of the classifier: it is the average difference between the first and second most probable class estimates per sample. Given l neural network layers with weight matrices $\mathbf{W}_i,\ i \in \{0, \cdots, l\}$, so that $\boldsymbol{\theta} = [\mathrm{vec}(\mathbf{W}_0), \mathrm{vec}(\mathbf{W}_1), \ldots, \mathrm{vec}(\mathbf{W}_l)]$, the general form of spectral complexity *normalized by the margin* γ is

$$(1/\gamma)R_{\boldsymbol{\theta}} := \frac{1}{\gamma}\prod_{i=0}^{l}||\mathbf{W}_i||_A \left(\sum_{i=0}^{l}\frac{||\mathbf{W}_i||_B^a}{||\mathbf{W}_i||_A^a}\right)^{1/a},$$

where A, B stand for a generic norm, and a is some constant.

Bounds on the true risk should have values ≤ 1 (in the introduction we used the even stricter definition that they should be better than random), however bounds based on spectral complexity, are larger than 1 by several orders of magnitude, making them

vacuous. However, normalized spectral complexity itself has been shown to correlate empirically with the generalization error, in a number of works (Neyshabur et al., 2017a; Bartlett et al., 2017). Specifically it is shown to correlate empirically across *training epochs*. As the generalization error increases or decreases during training, spectral complexity appears to increase and decrease as well. At the same time, for the same network, this measure of model complexity has high values when the network is trained on data where the labels have been randomized and considerably lower values when the real labels are reinstated. A number of different bounding techniques result in spectral complexity measures, including robustness, PAC-Bayes and Rademacher complexity (Sokolić et al., 2016; Bartlett et al., 2017; Neyshabur et al., 2017b, 2015; Golowich et al., 2017). At the same time, a number of other vacuous complexity measures have been proposed (Wang et al., 2018; Keskar et al., 2016; Thomas et al., 2019; Wei and Ma, 2019; Jiang et al., 2018; Liang et al., 2019; Arora et al., 2018), with some correlating better than others with generalization error.

On a fundamental level, simple correlation with generalization error is unsatisfying given that deep neural networks are increasingly being deployed in critical environments such as healthcare, finance and policing where they can potentially make life altering decisions.

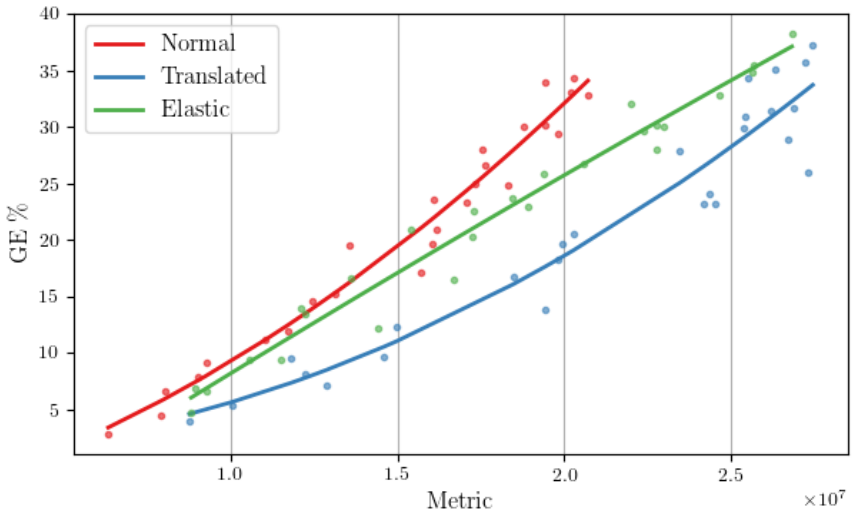


Figure 1.1 – Spectral complexity analyses of generalization error are insensitive to the known invariances of CNNs. We use 10^4 training and 10^4 testing images of Cifar-10 as our *Control* dataset and create two additional datasets called *Elastic Cifar* and *Translated Cifar* by removing half of the training and testing sets and replacing them, respectively, with random elastic deformations and translations of the remaining images. Each data point in the figure corresponds to a measurement at the end of each training epoch. We observe that, for constant spectral complexity (gray vertical lines), trained networks exhibit different generalization error for different datasets. At the same time spectral complexity correlates empirical with the generalization error across epochs.

The role of invariance. We also observe that spectral complexity-based measures feature a sum of the B-stable rank of the weight matrices involved, which for $a = 2$ given by

$$\sigma_B(\mathbf{W}_i) = \frac{\|\mathbf{W}_i\|_B^2}{\|\mathbf{W}_i\|_2^2},$$

where B stands for a generic norm, such as the Frobenius norm in $R_{\boldsymbol{\theta}}$ and the $(2,1)$ norm in $R'_{\boldsymbol{\theta}}$ (see respectively (1.3) and (1.4)) (Arora et al., 2018). The stable rank gives a robust estimate of the degrees of freedom of a matrix: roughly, an $n_1 \times n_2$ matrix $\mathbf{W}$ with constant stable rank has $\mathcal{O}(n_1 + n_2)$ degrees of freedom, instead of $\mathcal{O}(n_1 n_2)$ as usual. Similarly the second term in spectral complexity is $\prod_{i=0}^{l} ||\mathbf{W}_i||_A$ which for $A = 2$ can be seen as an upper bound on the Lipschitz constant of the deep neural network.

This interpretation should give us a pause for thought: bounds based on spectral complexity appear to be sophisticated parameter counting techniques, able to adapt to different neural network realizations. As such, they should in principle not be able to capture the complex interactions between data symmetries and CNN invariance to these symmetries.

Invariances are widely considered to be crucial in deep neural network design (Bengio et al., 2013). On the theoretical side, some CNNs have been proven to be invariant to translations and stable to deformations (Mallat, 2016; Wiatowski and Bölcskei, 2018). Also, CNNs, after training, empirically appear to be invariant to much more complex transformations on the data, such as adding sunglasses to faces (Radford et al., 2015).

While it is generally agreed that invariance to symmetries in the image data is a key property of modern deep convolutional neural networks, the role of invariances is generally absent from the generalization literature. Achille and Soatto (2018) showed that low information content in the network weights corresponds to learning invariant signal representations to various nuisance latent parameters. Their work however results in a vacuous generalization bound. Further, Sokolic et al. (2016) demonstrated that classifiers that are invariant (to a set of discrete transformations of input signals) can potentially have a much lower GE than non-invariant ones.

Similarly, due to the non-trivial correlations between filters, the generalization capacity of deep CNNs has been rarely studied. Works such as Zhou and Feng (2018), Du et al. (2017),Arora et al. (2018),Long and Sedghi (2019),Li et al. (2018b) are typically very involved, analyze greatly restricted settings and do not seem to lead to non-vacuous generalization bounds or to any new intuition apart from better parameter counting.

Research objectives. In this chapter we focus on the following question:

To what extent do existing bounds and complexity measures incorporate the invariance properties induced by deep convolutional architectures?

In trying to answer the above, we make the following contributions.

Contributions.

- We confirm empirically that spectral complexity bounds fail to capture the invariance properties of CNNs to data symmetries, such as elastic deformations and translations. As seen in Figure 1.1, CNNs with the same spectral complexity exhibit different GE when we augment the dataset with perturbations to which the convolutional architecture is inherently invariant. Our experiments suggest that these conclusions are not unique to our approach, but apply to spectral complexity-based generalization bounds in general (Bartlett and Mendelson, 2002; Sokolić et al., 2016; Bartlett et al., 2017; Neyshabur et al., 2017b, 2015; Golowich et al., 2017). We conclude that more research should be conducted in incorporating invariance properties in deriving complexity measures. At the same time claims that spectral complexity correlates empirically with generalization error should be stated with more precision.
- We analyze the case of *locally-connected* layers, i.e., layers constructed to have the same support as convolutional layers but which don't employ weight sharing. As such deep locally connected networks should not have the desired invariance properties of stacked convolutions. Counter-intuitively, we arrive to the same generalization error guarantees as convolutional architectures (up to negligible factors that are artifacts of the derivation). Our experiments indicate that crucial quantities in the bound are tight. Our theoretical result highlights potential contradictions that one might face when comparing different architectures using vacuous complexity measures such as spectral complexity.

While we empirically test only certain spectral complexity based bounds, our results should be meaningful for a number of modern bounds. These typically hold for *any* data generating distribution, and therefore should ignore the input data structure and the corresponding invariance properties of modern CNNs.

1.2 Spectral complexity metrics

We denote the learning sample $(X, Y) = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, that contains n input-output pairs. Samples (X, Y) are assumed to be sampled randomly from a distribution $\mathcal{D}$. Thus, we denote $(X, Y) \sim \mathcal{D}^n$ the i.i.d observation of n elements. We consider loss functions $\ell : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{F}$ is a set of predictors $f : \mathcal{X} \to \mathcal{Y}$. We also denote the empirical risk $\hat{\mathcal{L}}^\ell_{X,Y}(f) = (1/n) \sum_i \ell(f, \boldsymbol{x}_i, y_i)$ and the risk $\mathcal{L}^\ell_{\mathcal{D}}(f) = \mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \ell(f, \boldsymbol{x}, y)$.

A neural network transforms its inputs $\mathbf{a}_0 = \boldsymbol{x}$ to an output $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{a}_l$ through a series of l layers, each of which consists of a bank of units/neurons. The computation performed

by each layer $i \in \{1, ..., l\}$ is given as follows

$$\begin{aligned} \boldsymbol{s}_i &= \mathbf{W}_i \mathbf{a}_{i-1}, \\ \mathbf{a}_i &= \phi_i(\boldsymbol{s}_i), \end{aligned}$$

where ϕ_i is an element-wise non-linear function and $\mathbf{W}_i$ is a weight matrix.

We will define $\boldsymbol{\theta} = [\text{vec}(\mathbf{W}_0)\text{vec}(\mathbf{W}_0) \cdots \text{vec}(\mathbf{W}_l)]$, which is the vector consisting of all the network's parameters concatenated together, where vec is the operator which vectorizes matrices by concatenating their rows horizontally.

We consider the specific case where $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^k$ parameterized by $\boldsymbol{\theta}$ is used to map input vectors $\boldsymbol{x} \in \mathcal{X}$ to a k-dimensional encoding of the integer $y \in \mathcal{Y}$, encoding class membership. Typically the neural network outputs are normalized to form a probability distribution, using a softmax function, such that $\sigma(f_{\boldsymbol{\theta}}(\boldsymbol{x}))[i] = e^{f_{\boldsymbol{\theta}}(\boldsymbol{x})[i]} / \sum_{j=0}^{k} f_{\boldsymbol{\theta}}(\boldsymbol{x})[j]$. For the purposes of this section we will assume that we are dealing with the output representations *before* applying a softmax function)

We may encode the confidence of the classifier by incorporating a dependence on a desired margin $\gamma > 0$. Then, the γ-*margin classification loss* is defined as

$$\ell_\gamma(f_{\boldsymbol{\theta}}, \boldsymbol{x}, y) := \mathbb{I}(f_{\boldsymbol{\theta}}(\boldsymbol{x})[y] \leq \gamma + \max_{j \neq y} f_{\boldsymbol{\theta}}(\boldsymbol{x})[j]).$$

Note that we easily recover the standard 01-loss definition by setting $\gamma = 0$, $\ell_0(f_{\boldsymbol{\theta}})$. Our objective is to obtain bounds of the form

$$\mathcal{L}_{\mathcal{D}}^{\ell_0}(f_{\boldsymbol{\theta}}) \leq \hat{\mathcal{L}}_{X,Y}^{\ell_\gamma}(f_{\boldsymbol{\theta}}) + \text{Complexity}, \tag{1.1}$$

where

$$\begin{aligned} \mathcal{L}_{\mathcal{D}}^{\ell_0}(f_{\boldsymbol{\theta}}) &= \mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \ell_0(f, \boldsymbol{x}, y) \\ \hat{\mathcal{L}}_{X,Y}^{\ell_\gamma}(f_{\boldsymbol{\theta}}) &= (1/n) \sum_i \ell_\gamma(f, \boldsymbol{x}_i, y_i) \end{aligned}$$

are the 01-risk and the γ-*empirical* risk computed over a random training set of size n.

In spectral complexity based bounds the generalization error of a l layer neural network with layer weights $\boldsymbol{\theta}$ is expressed as

$$\mathcal{L}_{\mathcal{D}}^{\ell_0}(f_{\boldsymbol{\theta}}) \leq \hat{\mathcal{L}}_{X,Y}^{\ell_\gamma}(f_{\boldsymbol{\theta}}) + \mathcal{O}\left(\frac{\Psi_f R_{\boldsymbol{\theta}}}{\gamma\sqrt{n}}\right), \tag{1.2}$$

with the term Ψ_f being architecture-dependent and only $R_{\boldsymbol{\theta}}$ depending solely on the network weights. The latter term is the one we will refer to as *spectral complexity* of a

neural network. We will use two definitions. The first is

$$R_{\boldsymbol{\theta}} := \prod_{i=0}^{l} ||\mathbf{W}_i||_2 \left(\sum_{i=0}^{l} \frac{||\mathbf{W}_i||_F^2}{||\mathbf{W}_i||_2^2} \right)^{1/2}, \tag{1.3}$$

this definition corresponds to the one derived in a PAC-Bayes framework (Neyshabur et al., 2017b) together with $\Psi_f = l\sqrt{r}$ where l is the number of layers and r is the number of output dimensions per layer (we assume that each weight matrix has the same number of neurons/output dimensions). The second is due to Bartlett et al. (2017)

$$R'_{\boldsymbol{\theta}} := \prod_{i=0}^{l} ||\mathbf{W}_i||_2 \left(\sum_{i=0}^{l} \frac{||\mathbf{W}_i^\top||_{2,1}^{2/3}}{||\mathbf{W}_i||_2^{2/3}} \right)^{3/2}, \tag{1.4}$$

with $\Psi_f = \sqrt{lr}$ and is obtained using an involved covering argument. Here the Frobenius norm is substituted by the $(2,1)$-matrix norm defined as $\|\mathbf{X}\|_{2,1} = \| \|\mathbf{X}_{:,1}\|_2, \ldots, \|\mathbf{X}_{:,n_2}\|_2 \|_1$ for $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$. In the similar works of Bartlett and Mendelson (2002) and Neyshabur et al. (2015), the authors use the $||\cdot||_{1,\infty}$ norm and the $||\cdot||_F$ norm, respectively.

1.2.1 Empirical investigation of insensitivity

We aim to test whether spectral complexity captures accurately the known invariance properties of modern convolutional neural networks. To do this, we increase the relevance of translations and elastic deformations to the image classification task, aiming to give an advantage to invariant architectures.

Specifically, we created three different versions of the CIFAR-10 dataset: (a) The *control version* consists of 10000 training images and 10000 test images sampled randomly from the CIFAR-10 dataset. (b) The *translated version* is constructed by taking 5000 training images and 5000 test images sampled randomly from the CIFAR-10 dataset. These "base" sets are then augmented separately with another 5000 images each, that are random translations of the originals. (c) Finally, the *elastic version* is constructed similarly to the translated one, however the base sets are now augmented with images that are random elastic deformations of the originals.

We train using SGD a deep convolutional neural network on each of the above datasets and calculate the GE and the (normalized) spectral complexity metric $R_{\boldsymbol{\theta}_\star}/\gamma$ defined in (1.3) at the end of each epoch. Here $\boldsymbol{\theta}_\star$ denotes the weights at a given iteration. In all following experiments, we used the following architecture

$$\begin{aligned} &\text{input} \to 32\text{C3} \to \text{MP2} \\ &\to 64\text{C3} \to \text{MP2} \to 10\text{FC} \to \text{output}, \end{aligned} \tag{1.5}$$

where iCj denotes a convolutional layer with i output channels and $j \times j$ filter support,

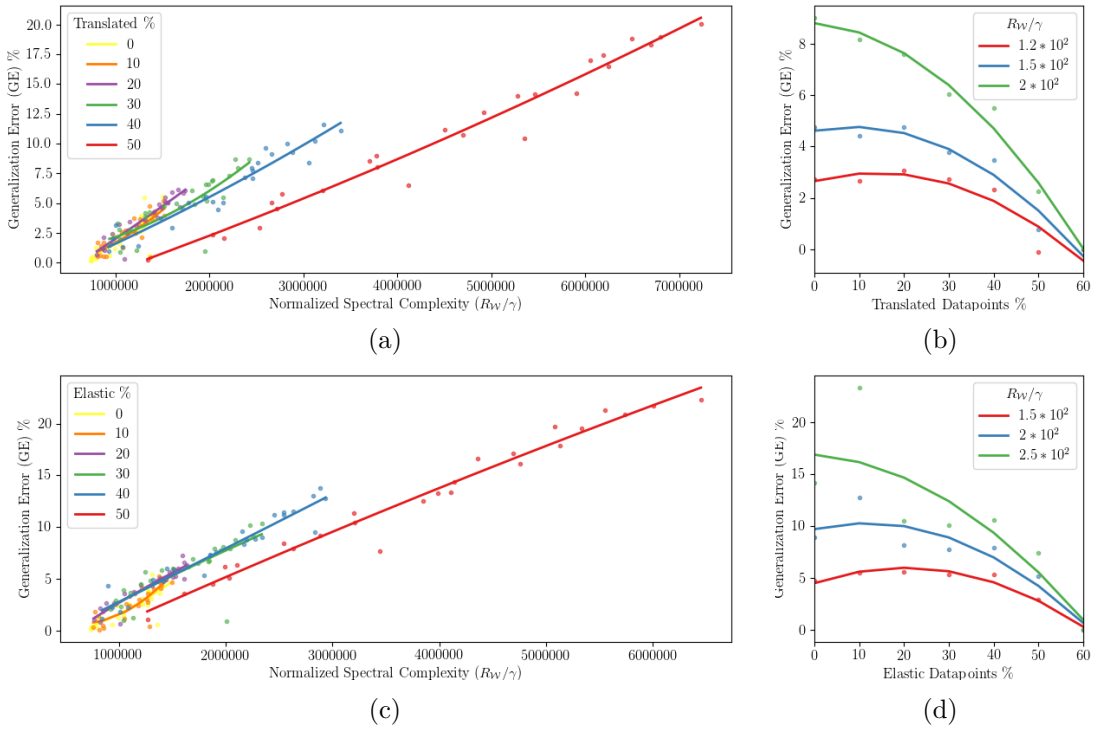


Figure 1.2 – **Varying the percentage of translations (a-b) and elastic deformations (c-d)**: We split Training and Testing datasets of constant size into two parts—the first contains images that form a base space, whereas the rest of the dataset contains images that are augmentations of the base space. The percentage values indicate the percentage of the *augmentations* over the total dataset. (a/c) We plot the GE vs spectral complexity. As we increase the number of translations/elastic deformations (equivalently decrease the percentage of the base space) the slopes of the GE curves decrease and we tend to have lower GE for the same spectral complexity metric values. (b/d) We plot the GE vs % of augmentations for constant complexity values. The percentage of augmentations correlates empirically with the GE indicating that spectral complexity does not account for the architecture invariances.

iFC denotes a fully connected layer with i outputs, and MPi denotes the max-pooling operator with pooling size of i. Our network has 42442 parameters in total. For γ we compute the *average* margin over the training data.

Figure 1.1 depicts the GE as a function of the metric for all three datasets, with markers corresponding to results for different epochs. It is important to compare GE values for the *same* spectral complexity as based on previous literature we should expect no changes to the GE. We see that for the same metric value the CNN exhibits different GE for the different datasets. One explanation is that the network is able to exploit it's architectural translation invariance and deformation stability to obtain a *lower* GE compared to the normal dataset. Intuitively, by replacing part of the variation in the data manifold with variations to which the network is invariant, we are simplifying the manifold for the

CNN improving the GE (even though the complexity of the classifier according to the spectral complexity is the same). We furthermore observe that the CNN is more robust to translations compared to elastic deformations, as it obtains improved GE for the same metric values. Alternatively, the network might have lower GE due to invariances learned through the augmentation. More experiments are needed to separate the two effects, however for our purposes the fact that invariaces (learned or architectural) are not captured in the bounds, is sufficient.

To confirm that our results are not specific to the Frobenius norm, but also representative of other spectral complexity definitions, we repeated the experiment also with the (2,1)-norm metric $R'_{\boldsymbol{\theta}_\star}/\gamma$ defined in (1.4). The results were consistent with those presented here and are deferred to Appendix A.4.

1.2.2 Delving deeper into invariances

To explore further the insensitivity of spectral complexity to data symmetries, we create datasets with constant size and varying percentage of augmentations. In particular, we start from datasets composed entirely of "base" samples and gradually increase the percentage of the dataset's augmented images from 0% from to 50%. Once more, we create two sets: one with translations and one featuring elastic deformations. We use SGD to train a CNN on these datasets and calculate after each epoch the GE and the spectral complexity metric. We use the same setup as the previous section, with 10000 training and 10000 testing samples from the CIFAR-10 dataset.

We plot the results in Figure 1.2. Specifically, Figures 1.2a and 1.2c show for the translated and elastic datasets, respectively, that more augmentation results in GE curves that have gradually smaller slopes. Thus, for the same metric, the GE decreases as the number of augmentations increases. Alternatively, we can fix a metric value and plot the GE vs the percentage of normal data-points. We plot the results in Figures 1.2b and 1.2d. We see that, for fixed metric values, the percentage of augmented data-points, i.e., ones that are translations or deformations of others, correlates empirically with the GE. These findings illustrate how spectral complexity is insensitive to the well-known invariances of CNNs and is therefore likely to lead to sub-optimal generalization bounds.

At the same time another interpretation of the results is that the neural networks *learn* useful invariances due to data augmentation. While further experiments are required to parse the effects of learned invariances and architectural invariances, our current experiments are sufficient to posit that claims such that "spectral complexity correlates empirically with generalization error" should be stated with more precision.

1.3 Comparing convolutional and locally connected networks

This section aims to provide evidence of contradictions that arise when using spectral complexity to estimate and compare the generalization error between different architectures. To do so, in Sections 1.3.1 and 1.3.2, we derive respectively generalization bounds for deep neural networks with convolutional and locally-connected layers—the latter maintain the sparsity structure, but do not employ weight sharing. The tightness of our derivation is investigated in Section 1.3.3. Interestingly, we find that both convolutional and locally-connected bounds take, up to log factors, the same form. Our result suggests that vacuous bounds with tightened constants should be viewed with caution, especially when used to compare different architectures.

1.3.1 Convolutional networks

Being derived for fully-connected neural networks, norm-based generalization bounds are not specifically adapted to convolutional architectures. Our first order of business is thus to understand how much one may gain by explicitly considering the structure of convolutions in the generalization error derivation.

To this end, we first aim to tighten the bound of Neyshabur et al. (2017b) and adapt it to the convolutional case. Specifically we will improve upon the architecture dependent constant Ψ_f. We show that for the case of convolutional layers the original value of $\Psi_f = l\sqrt{r}$ is unacceptably high.

We make the simplifying assumptions that each convolutional or locally connected layer has equal number of input and ouput channels $a_i = b_i$, and that all filters have the same support q^2. We prove the following generalization bound

Theorem 1.3.1. *(Generalization Bound). Let $f_{\boldsymbol{\theta}} : \mathbb{R}^d \to \mathbb{R}^k$ be a l-layer network, consisting of $|\mathcal{C}|$ convolutional layers, $|\mathcal{F}|$ fully-connected layers, and layer-wise ReLU activations. For any $\gamma, \delta > 0$, given weights $\boldsymbol{\theta}_\star$ with probability at least $1 - \delta$ over the training set of size n we have*

$$\mathcal{L}_{\mathcal{D}}^{\ell_0}(f_{\boldsymbol{\theta}_\star}) \leq \hat{\mathcal{L}}_{X,Y}^{\ell_\gamma}(f_{\boldsymbol{\theta}_\star}) + \mathcal{O}\left(\frac{B\,\Psi_f\,R_{\boldsymbol{\theta}_\star}}{\gamma\sqrt{n}}\right),$$

with $\|\boldsymbol{x}\|_2 \leq B$ being a uniform bound on the input vectors, $R_{\boldsymbol{\theta}_\star}$ is as in (1.3)*, and*

$$\Psi_f = q\sum_{i\in\mathcal{C}}\sqrt{b_i} + \sum_{i\in\mathcal{F}}\sqrt{s_i}.$$

Above, q^2 and b_i denote respectively the filter support and number of output channels of the i-th convolutional layer, and s_i counts the number of non-zero entries of the i-th

	LeNet-5	AlexNet	VGG-16
(Neyshabur et al., 2017b)	$\frac{10^{2.5}}{\sqrt{n}}$	$\frac{10^{3.5}}{\sqrt{n}}$	$\frac{10^{4}}{\sqrt{n}}$
Ours	$\frac{10^{2}}{\sqrt{n}}$	$\frac{10^{2.5}}{\sqrt{n}}$	$\frac{10^{2.5}}{\sqrt{n}}$

Table 1.1 – The value of the generalization error bound $((\Psi_f R_{\boldsymbol{\theta}^\star}/(\gamma\sqrt{n}))$ for common feed-forward architectures. For simplicity, we consider a best-case scenario and assume $R_{\boldsymbol{\theta}^\star} \approx \gamma \approx 1$.

fully-connected layer.

The theorem associates the generalization capacity of a deep convolutional neural network to its weights, as well as to key aspects of its architecture. Interestingly, there is a sharp contrast between convolutional and fully-connected layers, simply on account of computing more tightly the architecture dependent constant Ψ_f.

Fully-connected layers When all layers are sparse with sparsity s, ignoring log factors and factors $B, R_{\boldsymbol{\theta}^\star}, \gamma$ our bound implies that $n = \mathcal{O}(sl^2)$ samples suffice to attain good generalization. For the same setting, the sample complexity was determined as $n = \mathcal{O}(rl^2)$ by Neyshabur et al. (2017b) (assuming that input and output layer dimensions are the same across layers and equal to r).

Convolutional layers contribute much more mildly to the sample complexity with the latter increasing linearly on the filter support q^2 and channels b_i, but being independent on the layer input size. A case in point, in a fully convolutional network of l layers, each with and $b_i = b$ output channels, ignoring log factors and factors $B, R_{\boldsymbol{\theta}^\star}, \gamma$, we would need $n = \mathcal{O}(bq^2l^2)$, while previously we would have needed $n = \mathcal{O}(rl^2)$. Clearly $bq^2 << r$ as the first term depends on the convolutional filter support and the second (the ambient dimensionality) would be $r = bN^2$ with N^2 being the dimensionality of the *featuremap*.

To illustrate these differences resulting from Ψ_f, we conduct an experiment on LeNet-5 for the MNIST dataset, and on AlexNet and VGG-16 for the Imagenet dataset. We omit term $R_{\boldsymbol{\theta}^\star} \approx \gamma \approx 1$ assuming that $||\mathbf{W}_i||_F \approx ||\mathbf{W}_i||_2 \approx 1,\ \forall i \in \{1, ..., l\}$. We plot the results in Table 1.1. It can be seen that the proposed bounds are orders of magnitude tighter than the previous PAC-Bayesian approach.

Clearly, the assumption $R_{\boldsymbol{\theta}^\star} \approx \gamma \approx 1$ is unrealistic in practice. For values obtained by trained networks the bounds presented above are still vacuous by several orders of magnitude. We will see that this looseness has consequences when comparing the bound to the one for locally-connected architectures.

In the following we present an outline of the proof for Theorem 1.3.1. We defer the proof to Appendix A.3. Both the outline and the complete proof are quite involved. For the

reader that is not interested in the proof, the main chapter text continues in Section 1.3.2.

Proof outline of Theorem 1.3.1

We begin by presenting two prior results which will be useful later. The first relates the noise robustness to perturbations of a classifier to the GE. The second quantifies the perturbation robustness of general deep neural networks. We then outline how these apply to the convolutional setting. Before proceeding, we recall that, given two probability measures p and q over a set X, the Kullback-Leibler divergence is defined as $\mathrm{KL}(p||q) := \int_X \log \frac{dp}{dq} dp$.

Useful previous results. Let $f_{\boldsymbol{\theta}_\star}$ be any deterministic predictor (not necessarily a neural network). The following lemma from Neyshabur et al. (2017b) introduces the condition $\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}]$ as a probabilistic bound on the Lipschitz constant of the predictor $f_{\boldsymbol{\theta}_\star}$, and relates it to the generalization error:

Lemma 1.3.2 (Neyshabur et al. (2017b))**.** *We assume a distribution $\mathcal{D}$ over $\mathcal{X}\times\mathcal{Y}$, a hypothesis set $\mathcal{F}$ parametrized by classifiers $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$, loss functions $\ell_0, \ell_\gamma : \mathcal{F}\times\mathcal{X}\times\mathcal{Y} \to [0,1]$, a prior distribution π over $\boldsymbol{\theta}$, a real number $\delta \in (0,1]$, a real number $\gamma > 0$ and deterministic weights $\boldsymbol{\theta}_\star$. Then, for any random variable $\boldsymbol{u}$ s.t. $\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}$, we have with probability at least $1-\delta$ over the choice of $(X,Y)\sim\mathcal{D}^n$,*

$$\mathcal{L}_{\mathcal{D}}^{\ell_0}(f_{\boldsymbol{\theta}_\star}) \leq \hat{\mathcal{L}}_{X,Y}^{\ell_\gamma}(f_{\boldsymbol{\theta}_\star}) + \mathcal{O}\left(\sqrt{\frac{\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta}_\star+\boldsymbol{u})||\pi) + \ln\frac{6n}{\delta}}{n-1}}\right) \tag{1.6}$$

where n is the number of training samples.

A trade-off can be observed between the condition $\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}$ and the KL term in the right hand side of the above inequality. The KL term is *inversely* proportional to the variance of the noise $\boldsymbol{u}$ (intuitively if we increase jointly the variance of the posterior and a suitably chosen prior, the two distributions become flatter and more similar). Therefore one would want to maximize the variance of the noise, however the distance $\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2$ can potentially grow unbounded with high probability for high enough values of the variance.

Characterizing the condition $\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}$ entails understanding the sensitivity of our deep convolutional neural network classifier on random perturbations $\boldsymbol{u}$ to the weights. To that end, we review here a useful perturbation bound from Neyshabur et al. (2017b) on the output of a general deep neural network:

Lemma 1.3.3. *[Neyshabur et al. (2017b)] For any $B, l > 0$, let $f_{\boldsymbol{\theta}} : \mathcal{X}_{B,d} \to \mathbb{R}^k$ be a*

l-layer network with ReLU activations.. Then for any $\boldsymbol{\theta}_\star$, *and* $\boldsymbol{x} \in \mathcal{X}_{B,d}$, *and random variable* $\boldsymbol{u} = \text{vec}(\{\mathbf{U}_i\}_{i=0}^l)$ *such that* $||\mathbf{U}_i||_2 \leq \frac{1}{l}||\mathbf{W}_i||_2$, *the change in the output of the network can be bounded as follows*

$$|f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq e^2 B\tilde{\beta}^{l-1}\sum_i ||\mathbf{U}_i||_2, \tag{1.7}$$

where e, B *and* $\tilde{\beta}^{l-1}$ *are considered as constants after an appropriate normalization of the layer weights.*

We note that correctly estimating the spectral norm of the perturbation at each layer is critical to obtaining a tight bound. Specifically, if we exploit the sparsity structure of the perturbation we can increase significantly the variance of the added perturbation for which $\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}$ holds. With a slight abuse of notation we also take this structure into account when forming $\boldsymbol{u} = \text{vec}(\{\mathbf{U}_i\}_{i=0}^l)$, i.e. the vectorization is performed only over non-zero elements of $\mathbf{U}_i$.

The analysis for the convolutional case is difficult due to the fact that the noise per pixel is not independent. We obtain the following lemma, where log parameters have been omitted for clarity:

Lemma 1.3.4. *Let* $\mathbf{U} \in \mathbb{R}^{d_2\times d_1}$ *be the perturbation matrix of a 2d convolutional layer with a input channels, b output channels, convolutional filters* $\phi \in \mathbb{R}^{q\times q}$ *and feature maps* $F \in \mathbb{R}^{N\times N}$. *We assume that elements* $\mathbf{U}_{ij}$ *are non-zero only if they correspond to non-zero locations in the sparsity pattern of the convolution operator. Let these elements follow a Gaussian distribution* $\mathbf{U}_{ij} \sim \mathcal{N}(0, \sigma^2)$. *We have*

$$||\mathbf{U}||_2 \leq \sigma(q[\sqrt{a} + \sqrt{b}] + \sqrt{2\log(\frac{2N^2}{\delta})}), \tag{1.8}$$

with probability at least $1 - \delta$.

We see that the spectral norm of the noise is independent of the dimensions of the latent feature maps, but it is a function of the root of the filter support q, the number of input channels a and the number of output channels b.

With this in place, the following lemma identifies the maximum value of the variance parameter σ^2 that balances the noise sensitivity with the KL term dependence.

Lemma 1.3.5. *(Perturbation Bound). For any* $B, l > 0$, *let* $f_{\boldsymbol{\theta}} : \mathcal{X}_{B,d} \to \mathbb{R}^k$ *be a l-layer network with ReLU activations and we denote by* $\mathcal{C}$ *the set of convolutional layers and* $\mathcal{F}$ *the set of fully connected layers. Then for any deterministic weights* $\boldsymbol{\theta}_\star$, *and* $\boldsymbol{x} \in \mathcal{X}_{B,d}$, *and a perturbation for* $\boldsymbol{u} \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$, *for any* $\gamma > 0$ *with*

$$\sigma = \frac{\gamma}{42B\tilde{\beta}^{l-1}[\sum_{i\in\mathcal{C}} K_i + \sum_{i\in\mathcal{F}} J_i]} \tag{1.9}$$

we have

$$\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2},$$

where e, B, $\tilde{\beta}^{l-1}$ are considered as constants after an appropriate normalization of the layer weights, $K_i = q_i\{\sqrt{a_i} + \sqrt{b_i} + \sqrt{2\log(4N_i^2 l)}\}$ and $J_i = q_i\{2\sqrt{s_i} + \sqrt{2\log(2l)}\}$.

Theorem 1.3.1 follows directly from calculating the KL term in Lemma 1.3.2, by noting that $\boldsymbol{\theta}_\star + \boldsymbol{u} \sim \mathcal{N}(\boldsymbol{\theta}_\star, \sigma^2\boldsymbol{I})$, $\pi \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$, and that then $\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta}_\star + \boldsymbol{u})||\pi) \leq \frac{|\boldsymbol{\theta}_\star|^2}{2\sigma^2}$. We also set $a_i = b_i$ and $q_i = q$ for all layers.

1.3.2 Locally-connected networks

The improvement we attained by taking into account the structure of convolutional layers, though significant, still falls short from explaining why deep CNNs are able to generalize beyond the training set—the bounds are too pessimistic.

Locally-connected layers have a sparse banded structure similar to convolutions, with the simplifying assumption that the weights of the translated filters are not shared. The weight matrix is exemplified in Figure 1.4a for the case of one-dimensional convolutions. While this type of layer is not used in practice, it enables us to isolate the effect of sparsity on the generalization error. We prove the following:

Theorem 1.3.6. *Let $f_{\boldsymbol{\theta}_\star} : \mathbb{R}^d \to \mathbb{R}^k$ be a l-layer network, consisting of $|\mathcal{L}|$ locally-connected layers, $|\mathcal{F}|$ fully-connected layers, and layer-wise ReLU activations. For any $\gamma, \delta > 0$, with probability at least $1 - \delta$ over the training set of size n we have*

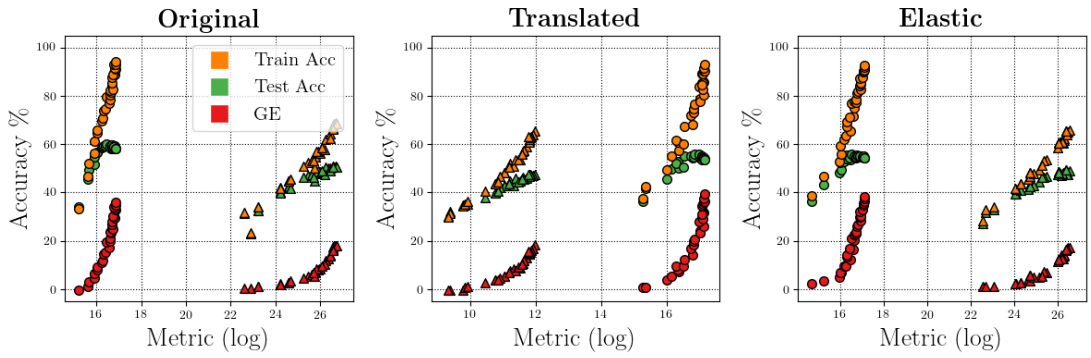
$$\mathcal{L}_{\mathcal{D}}^{\ell_0}(f_{\boldsymbol{\theta}_\star}) \leq \hat{\mathcal{L}}_{X,Y}^{\ell_\gamma}(f_{\boldsymbol{\theta}_\star}) + \mathcal{O}\left(\frac{B\,\Psi_f\,R_{\boldsymbol{\theta}_\star}}{\gamma\sqrt{n}}\right),$$

with $\|\boldsymbol{x}\|_2 \leq B$ being a uniform bound on the input vectors, $R_{\boldsymbol{\theta}_\star}$ is as in (1.3)*, and*

$$\Psi_f = q\sum_{i\in\mathcal{L}}\sqrt{b_i} + \sum_{i\in\mathcal{F}}\sqrt{s_i}.$$

Above, q^2 and b_i denote respectively the filter support and number of output channels of the i-th locally-connected layer, and s_i counts the number of non-zero entries of the i-th fully-connected layer.

Surprisingly for the given choice of spectral complexity the obtained bounds for convolutional and locally connected layers are identical up to log factors that are artifacts of the derivation. Implicitly, the hypothesis class $\mathcal{H}$ induced by spectral complexity is large enough to include both convolutional and non-convolutional architectures. At the same time, the bounds in both cases hold for *any* data distribution $\mathcal{D}$. These two points stand in stark contrast with common design practice where the hypothesis class $\mathcal{H}$ is assumed



(a) (b) (c)

Figure 1.3 – **Convolutional and Locally Connected Networks**: In Figure 1.3a we plot results for the original dataset. In Figures 1.3b and 1.3c we plot results for translations and elastic deformations respectively. There are a number of visible inconsistencies. For example in Figures 1.3a and 1.3c, the convolutional network has the highest training accuracy and lowest estimated complexity, but has the *highest* generalization error.

to be convolutional architectures, with good generalization properties specifically for data distributions $\mathcal{D}$ that represent natural images.

In hindsight, it might be clear that the upper bound on convolutional layers is not tight. However, all of the above are not evident in previous analyses and consequently misleading conclusions can be drawn when validating bounds through simple empirical correlation. In fact a number of works have claimed to provide improvements to generalization bounds by tightening constants such as the ones we analyzed here, while keeping the bound values vacuous (Long and Sedghi, 2019; Li et al., 2018b). It is unknown to us to what extent these results are applied beyond theoretical settings, however our results here should give a pause for thought in that regard.

Experiments

We conduct some experiments which highlight some inconsistencies that occur when comparing different architectures using spectral complexity. We use two architectures, one convolutional neural network

$$\begin{aligned} &\text{input} \to 32\text{C3} \to \text{MP2} \\ &\to 64\text{C3} \to \text{MP2} \to 10\text{FC} \to \text{output}, \end{aligned} \tag{1.10}$$

and one locally connected network

$$\begin{aligned} &\text{input} \to 32\text{LoC3} \to \text{MP2} \\ &\to 64\text{LoC3} \to \text{MP2} \to 10\text{FC} \to \text{output}, \end{aligned} \tag{1.11}$$

where $i\text{LoC}j$ is a locally connected layer with i output channels and $j \times j$ filter support.

We train them on the original CIFAR-10, as well as two additional datasets. Specifically as in Section 1.2.2 we split the original training and testing sets in two, we remove one part and replace it with translations and elastic deformations from the other part. Specifically we replace exactly 50% of the training and testing set as in the final experimental setup of Section 1.2.2. We train the neural networks using stochastic gradient descent for 30 epochs in the convolutional case and 100 epochs in the locally connected case. Looking at Theorems 1.3.1 1.3.6, as $\Psi_f, \sqrt{n}, B$ are the same for both networks, at the end of each epoch we calculate the spectral complexity normalized by the margin $R_{\boldsymbol{\theta}_\star}/\gamma$ with $R_{\boldsymbol{\theta}_\star}$ as in (1.3)

$$R_{\boldsymbol{\theta}_\star} := \prod_{i=0}^{l} ||\mathbf{W}_i||_2 \left(\sum_{i=0}^{l} \frac{||\mathbf{W}_i||_F^2}{||\mathbf{W}_i||_2^2} \right)^{1/2}$$

and use it as a measure of complexity. For γ we compute the *average* margin over the training data. We also calculate the training accuracy, testing accuracy and generalization error. We plot the results in 1.3. Circles denote the convolutional network while triangles denote the locally connected network. A number of inconsistencies are immediately noticeable. In the original dataset (Figure 1.3a) and the dataset of elastic deformations (Figure 1.3c), the convolutional network has both the highest training accuracy and the lowest estimated complexity, however it has the *largest* generalization error. Furthermore for a number of data points, across all figures, the generalization error is ≈ 0 and the training and testing accuracies are non-trivial, however normalized spectral complexity is significantly different between the two architectures. For example in Figure 1.3b, for 40% training accuracy, spectral complexity for the convolutional network is several orders of magnitude *higher* than for the locally connected network, but at the same time the testing accuracy is approximately the same $\approx 40\%$, and the generalization error is for both architectures ≈ 0.

Proof outline of Theorem 1.3.6

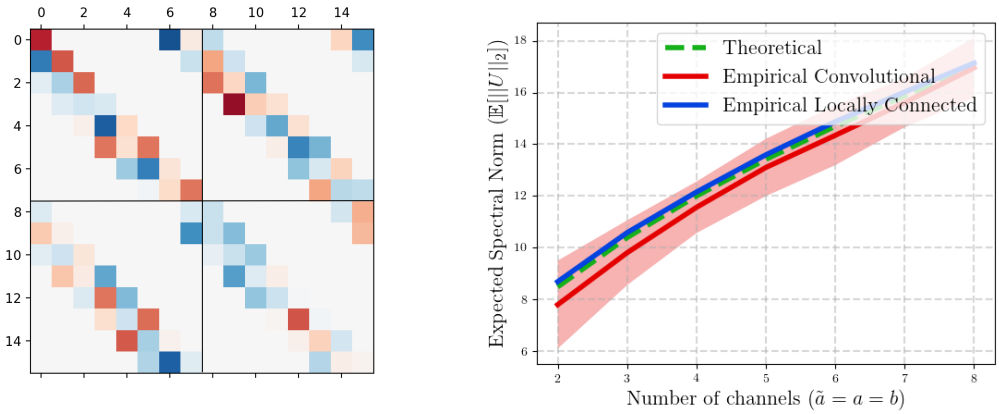
The analysis is similar to the case of convolutional layers, with the exception of how term $\|\mathbf{U}\|_2$ is bounded. For locally connected layers we can derive the following

Lemma 1.3.7. *Let $\mathbf{U} \in \mathbb{R}^{d_2 \times d_1}$ be the perturbation matrix of a 2d locally-connected layer with a input channels, b output channels, filters $\phi \in \mathbb{R}^{q \times q}$ and feature maps $F \in \mathbb{R}^{N \times N}$. Then if non-zero elements follow $\mathbf{U}_{i,j} \sim \mathcal{N}(0, \sigma^2)$, we have*

$$||\mathbf{U}||_2 \leq \mathcal{O}(\sigma(q[\sqrt{a} + \sqrt{b}] + \sqrt{2\log(\frac{1}{\delta})})), \tag{1.12}$$

with probability at least $1 - \delta$.

The rest of the proof technique is identical to that used for convolutional layers and is deferred to Appendix A.2.



(a) (b)

Figure 1.4 – (a) The weight matrix of a 1D locally-connected layer with two input and two output channels. (b) We plot empirical and theoretical estimates of the mean $\mathbf{E}_{\mathbf{U}\sim d}||\mathbf{U}||_2$. Our theoretical upper estimates (in green) closely follow the empirical estimates for both the convolutional (in red) and the locally-connected (in blue) case. Note that the theoretical estimate is identical for both cases, so we only plot it once. For the empirical estimates, we also show one standard deviation confidence intervals. The locally connected case is much more concentrated than the convolutional and the corresponding confidence interval is not visible in the figure.

1.3.3 Empirical investigation of tightness

Theorems 1.3.1 and 1.3.6 depend on $\mathbf{E}_{\mathbf{U}\sim d}||\mathbf{U}||_2$, the expected spectral norm of the layer noise (overloading the notation, d here denotes the distribution of $\mathbf{U}$), as effectively we have been proving concentration inequalities of the form $\mathbb{P}(||\mathbf{U}||_2 \leq \mathbf{E}_{\mathbf{U}\sim d}||\mathbf{U}||_2) \geq c$. We test our concentration bounds by computing analytically and empirically $\mathbf{E}_{\mathbf{U}\sim d}||\mathbf{U}||_2$ for synthetic data.

Our experiment considers 1D signals, filters $\phi \in \mathbb{R}^9$, feature maps $F \in \mathbb{R}^{100}$, a input channels, b output channels. We calculate the spectral norm $||\mathbf{U}||_2$ while increasing the number of input and output channels with $\tilde{a} := a = b$. To obtain empirical estimates, we average the results over $N = 100$ iterations for each choice of $\tilde{a}$. As seen in Figure 1.4b, the theoretical values closely match the empirical estimates.

As such we consider our analysis tight with respect to the constant Ψ_f.

1.4 Relationship with PAC-Bayes

We now discuss how Lemma 1.3.2 is related to a PAC-Bayes bound. PAC-Bayes bounds deal with randomized classifiers with a posterior $\hat{\rho}$ and a prior π distribution, and model

the complexity of the classifier as roughly $\mathrm{KL}(\hat{\rho}||\pi)$. The particular bound that was used in the previous analysis is the one due to McAllester (1999)

Theorem 1.4.1. *McAllester (1999) Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, a hypothesis set $\mathcal{F}$, a loss function $\ell' : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to [0, 1]$, a prior distribution π over $\mathcal{F}$, a real number $\delta \in (0, 1]$, and a real number $\beta > 0$, with probability at least $1 - \delta$ over the choice of $(X, Y) \sim \mathcal{D}^n$, we have*

$$\forall \hat{\rho} \text{ on } \mathcal{F} : \mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell'}_{\mathcal{D}}(f) \leq \mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f) + \sqrt{\frac{\mathrm{KL}(\hat{\rho}||\pi) + \ln\frac{2n}{\delta}}{2n-1}}. \tag{1.13}$$

However typically in the context of deep neural networks we are interested in estimating the complexity of a *deterministic* neural network with parameters $\boldsymbol{\theta}_\star$, which we have computed using vanilla stochastic gradient descent. Thus the benefit of Lemma 1.3.2 is that it is a particular derandomization of the PAC-Bayesian bound 1.13. First one models the posterior over deep networks parametrized by $f = f_{\boldsymbol{\theta}}$, as $\hat{\rho}(\boldsymbol{\theta}) = \hat{\rho}(\boldsymbol{\theta}_\star + \boldsymbol{u})$, for some random variable $\boldsymbol{u}$. One then moves to a *margin* loss which quantifies the robustness of the network to perturbations. Then the assumption $\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}$ allows one to derandomize the bound, removing the expectations $\mathbf{E}_{f\sim\hat{\rho}}$, assuming that the random perturbations $\boldsymbol{u}$ do not result in violating the margin.

1.5 Further criticism of vacuous bounds

There has also been significant criticism of spectral complexity implicitly or explicitly from a number of other angles.

Criticism of uniform convergence. In Nagarajan and Kolter (2019), the authors posit that two sided uniform convergence bounds cannot produce non-vacuous estimates for deep neural networks, even with aggressive pruning of the hypothesis space. The result rests on the fact that uniform convergence establishes a bound as a supremum over all hypotheses $f \in H$ roughly such that

$$\mathbb{P}_{(X,Y)\sim\mathcal{D}^n}\left[\sup_{f\in H} |\mathcal{L}^{01}_{\mathcal{D}}(f) - \hat{\mathcal{L}}^{01}_{X,Y}(f)| \leq \epsilon_{\text{unif}}\right] \geq 1 - \delta.$$

The authors in Nagarajan and Kolter (2019) show empirically in the case of neural networks that for any classifier f that has low risk $\mathcal{L}^{01}_{\mathcal{D}}(f)$ (estimated using a test set), one can construct an adversarial *training* set $(X, Y)' \sim \mathcal{D}^n$ such that $\hat{\mathcal{L}}^{01}_{(X,Y)'}(f) \approx 1$. Going a bit deeper the analysis depends on the empirical observation that deep neural networks have macroscopically simple decision boundaries but "overfit" microscopically around specific data samples, this in turn allows the adversarial training sets $(X, Y)'$ to be constructed. In particular the criticism holds for derandomized PAC-Bayes bounds such as the ones of Chapter 1 (Nagarajan and Kolter, 2019), where we strive for a bound

on a single derandomized network f trained with vanilla SGD. In the following we will be dealing only with bounding the generalization error of stochastic classifiers, where we add small noise to the parameters, which should remove the above microscopic overfitting.

Further empirical evaluation of vacuous bounds. In Jiang et al. (2019) the authors conduct a detailed evaluation of spectral complexity based generalization bounds, among others. For a given dataset they measure the correlation of spectral complexity with the actual generalization error based on a validation set, as one varies the values of hyperparameters such as learning rate, batch size, as well as width and depth of trained architectures. They find that spectral complexity and other norm based complexity measures are strongly *negatively* correlated with the generalization error.

1.6 Discussion and Summary

In this chapter we analyzed spectral complexity based generalization bounds, in the context of deep convolutional neural networks. We showed empirically that one can construct datasets with increasing numbers of translations and elastic deformations, such that spectral complexity remains constant, while the generalization error of the classifier changes. This, implies that claims such as that spectral complexity correlates empirically with generalization need to be more carefully stated. Furthermore, it might be worthwhile to consider measures of complexity that take into account the invariance properties of deep neural networks, in order to tighten existing bounds. At the same time, we compared bounds for convolutional neural networks to bounds for locally connected networks (which have the same sparsity pattern but do not implement weight sharing). We showed how, a particular well known bound, gives complexity estimates of the same order of magnitude for both architectures highlighting the pitfalls of comparing vacuous generalization bounds between different architectures. Together the above imply a number of limitations in current generalization bounds. Locating the source of the above problems (and possible misconceptions) as the *vacuity* of most current bounds, we are motivated to look deeper at existing approaches that derive *non-vacuous* complexity estimates.

Non-Vacuous Bounds Part II

2 Stochastic Variational Inference and PAC-Bayes.

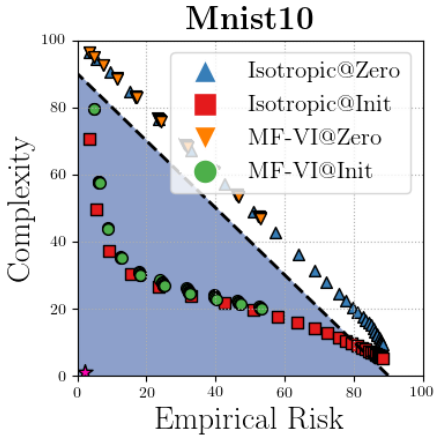
In the previous chapter we highlighted some limitations of spectral complexity, and with it, a number of modern vacuous generalization bounds. In particular, these bounds fail to account for invariance properties of learned classifiers, and one can easily construct datasets with different symmetries, such that spectral complexity doesn't correlate empirically with generalization error. Perhaps more importantly, when comparing the estimates of the generalization error between different *architectures* (possibly trained on the same dataset) we find that we can reach counterintuitive conclusions. Intuitively, for different architectures, one derives upper bounds that are significantly loose and are thus incomparable. We also reviewed a number of other criticisms of modern vacuous bounds.

Together all of the above paint a picture of vacuous complexity measures that fail to predict generalization when scrutinized in a detailed manner, can lead to a false sense of security when trying to compare predicted generalization across different architectures and might have inherent difficulties in providing tight bounds.

2.1 Introduction

In this chapter we focus on the opposite end of this spectrum, where two works (Dziugaite and Roy, 2017; Zhou et al., 2018) based on the PAC-Bayes framework (McAllester, 1999) made steps towards non-vacuous and interpretable bounds. Recall from Chapter 1 that PAC-Bayes bounds deal with randomized classifiers with posterior and prior distributions $\hat{\rho}$ and π respectively. Given that typically one wants to bound the risk of a deterministic classifier f the posterior $\hat{\rho}$ is chosen to be in some sense close to f(i.e. it is usually centered at f). Then, PAC-Bayes theorems make statements that are roughly of the form

$$\mathbf{E}\mathcal{L}(\hat{\rho}) \leq \mathbf{E}\hat{\mathcal{L}}(\hat{\rho}) + \beta \mathrm{KL}(\hat{\rho}||\pi), \tag{2.1}$$



(a)

Figure 2.1 – **Risk-Complexity plot for MNIST 10**: The area below the dashed line corresponds to non-vacuous pairs of (complexity, empirical risk). The purple star corresponds to the optimal bound implied by the testing set. We parametrize the PAC-Bayes bound with different combinations of diagonal Gaussian priors and posteriors. "Isotropic@" corresponds to isotropic priors and posteriors with the prior centered at 0 and at the deep neural network random initialization. Similarly for "Mean-Field VI@" the posterior is diagonal but non-isotropic and we optimize it with variational inference. Choosing the prior mean to be the random initialization improves the bounds greatly in both cases. On the contrary when optimizing with mean-field variational inference there is negligible improvement over the isotropic case.

where $\mathcal{L}(\hat{\rho})$ is the risk, $\hat{\mathcal{L}}(\hat{\rho})$ is the empirical risk and the expectation is over the posterior. The $\beta\mathrm{KL}(\hat{\rho}||\pi)$ term between the prior and posterior acts as a measure of complexity for the classifier.

The RHS of (3.1) corresponds to a variational encoding scheme of the deep neural network weights, where the variance of the noise in the posterior measures the level of precision used in the encoding (Blier and Ollivier, 2018). The weights need to be encoded with a precision $\beta\mathrm{KL}(\hat{\rho}||\pi)$ that is as low as possible, without increasing the empirical risk $\mathbf{E}\hat{\mathcal{L}}(\hat{\rho})$. The procedure in Chapter 1 can be seen as trying to derive this level of precision analytically. Given the highly non-linear nature of deep neural networks it is not easy to analyze how noise applied to the weights affects subsequent layers. At the end, the bound in Chapter 1 assumes that noise applied to a given layer is multiplied by the spectral norms of the subsequent layers. This is a pessimistic assumption that results in vacuity.

By contrast, instead of trying to derive analytical results, in Dziugaite and Roy (2017), the authors minimize this variational code directly using a differentiable surrogate, by parameterizing the prior and posterior as diagonal Gaussians, and optimizing using stochastic variational inference (Hoffman et al., 2013; Kingma and Welling, 2013), aiming to balance the terms $\mathbf{E}\mathcal{L}(\hat{\rho})$ and $\beta\mathrm{KL}(\hat{\rho}||\pi)$. They obtain non-vacuous generalization

bounds on a simplified MNIST(LeCun and Cortes, 2010) dataset, but are unable to scale their result to larger problems.

The reason for failing to scale might lie in the way $\hat{\rho}$ and π are modeled, or in the optimization procedure. Stochastic variational inference in general is known to result in poor weight encodings, but the reasons behind this are unclear (Blier and Ollivier, 2018). Variational inference, in the context of Bayesian neural networks, is thought to suffer from high gradient variance (Kingma et al., 2015; Wu et al., 2018; Wen et al., 2018). In addition, correlations between parameters are often omitted as in Dziugaite and Roy (2017), as storing and manipulating the full covariance matrix is computationally infeasible. This is known as the *mean-field* approximation and might be too restrictive in deriving useful posteriors (Ritter et al., 2018; Mishkin et al., 2018), and therefore tight codes.

Consequently, in Zhou et al. (2018) the authors first compress deep neural networks by sparsifying them and deriving a variational code on the remaining parameters. Off the shelf compression algorithms compress remarkably well and thus Zhou et al. (2018) obtain non-vacuous but loose bounds for the much more complex Imagenet (Deng et al., 2009). A significant drawback of this approach is that the bound is derived for a network whose parameters are not similar *even in expectation* to the original ones (Suzuki, 2019).

Research objectives. We thus focus on analyzing the non-vacuous bounds of Dziugaite and Roy (2017), where variational inference is applied directly on the original weight space. Importantly, we lack meaningful comparison tools. The techniques in Dziugaite and Roy (2017); Zhou et al. (2018) actually provide *multiple* bounds corresponding to different levels of encoding precision of the weights, which is usually controlled by the the parameter β in (3.1). However, results are presented in single (empirical risk, complexity) pairs, making drawing conclusions difficult.

Our first contribution is thus to introduce "Risk-Complexity" plots 2.1 (which we first presented in the introduction). On the x-axis we plot the Empirical Risk $\hat{\mathcal{L}}(\hat{\rho})$, while on the y-axis we plot the estimated Complexity $\beta\mathrm{KL}(\hat{\rho}||\pi)$ or the equivalent complexity metric. The plots have a number of advantages. We can easily plot the region of non-vacuity and the location of the best possible bound implied by the *testing* set. For an optimization based bound method we can then derive multiple (complexity, empirical risk) estimates and plot a Pareto front of all combinations. This results in an intuitive way for comparing bounding methods where one can simply inspect the Pareto fronts in relation to the best possible pair implied by the testing set.

Armed with our new visualization tools we are ready to scrutinize the results of Dziugaite and Roy (2017). The authors combine four elements in deriving non-vacuous bounds: i) changing the prior to be centered at the random initialization instead of at zero ii) optimizing the posterior covariance iii) optimizing the posterior mean iii) simplifying the

classification problem by merging the 10 MNIST classes into 2 aggregate ones. In this way it is unclear what is the contribution of each to obtaining non-vacuous bounds.

In particular, separating the effects of i,ii and iii is important. Flatness at the minimum has been frequently cited as a desirable property for good generalization (Keskar et al., 2016). However, current results show mainly empirical correlations with generalization error (Keskar et al., 2016) and the exact effect of flatness is still debated (Dinh et al., 2017). Point ii is related to flatness at the minimum, as increased posterior variance while $\mathbf{E}\hat{\mathcal{L}}(\hat{\rho})$ remains small implies a flat minimum. Importantly, when relating PAC-Bayes to flatness one needs to keep the mean of the posterior fixed. Optimizing the mean and then the covariance corresponds to measuring the flatness of a *different* minimum.

Our contributions. Through detailed experiments we find that for diagonal Gaussian priors and posteriors the dominant element which turns a vacuous bound to non-vacuous is centering the prior at the random initialization instead of at 0. Optimizing the covariance using stochastic variational inference results in negligible or no gains. In fact, a simple isotropic Gaussian baseline in the prior and posterior results in nearly identical bound values.

We are then motivated to investigate two common explanations for this ineffectiveness. First it could be that stochastic variational inference has not properly converged. Secondly, PAC-Bayes theory allows improved bounds by choosing priors that reflect prior knowledge about the problem, as long as these priors don't depend on the training set. Choosing the random initialization to be the prior mean is already a good prior mean choice. It might be that through a better choice of prior *covariance* the mean-field approximation could yield meaningful improvements to the posterior covariance and hence the bound.

Through a simple theoretical analysis, we explore both of these explanations. Specifically, we leverage the fact that the loss landscape around the minimum is empirically quadratic, to derive closed form bound solutions with respect to both posterior and prior covariance. The second result is invalid under the PAC-Bayes framework but is useful as a sanity check. Our results imply both problems with optimization of VI as well as that significantly better priors can in theory be found. At the same time, the closed form results are far from optimal and point to intrinsic limitations of the mean-field approximation.

We then motivate modeling the curvature at the minimum through a simplified version of K-FAC (Martens and Grosse, 2015). This allows us to efficiently sample (complexity, empirical risk) pairs with improved curvature estimates. Using our Risk-Complexity plots, we find that for randomized classifiers with medium to low empirical risk this results in significant improvements in the generalization bound quality, compared to the implied limits of the mean-field approximation.

2.2 Preliminaries

We denote the learning sample $(X, Y) = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, that contains n input-output pairs. Samples (X, Y) are assumed to be sampled randomly from a distribution $\mathcal{D}$. Thus, we denote $(X, Y) \sim \mathcal{D}^n$ the i.i.d observation of n elements. We consider loss functions $\ell : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{F}$ is a set of predictors $f : \mathcal{X} \to \mathcal{Y}$. We also denote the empirical risk $\hat{\mathcal{L}}^{\ell}_{X,Y}(f) = (1/n) \sum_i \ell(f, \boldsymbol{x}_i, y_i)$ and the risk $\mathcal{L}^{\ell}_{\mathcal{D}}(f) = \mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \ell(f, \boldsymbol{x}, y)$.

A neural network transforms its inputs $\mathbf{a}_0 = \boldsymbol{x}$ to an output $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{a}_l$ through a series of l layers, each of which consists of a bank of units/neurons. The computation performed by each layer $i \in \{1, ..., l\}$ is given as follows

$$\begin{aligned} \boldsymbol{s}_i &= \mathbf{W}_i \mathbf{a}_{i-1}, \\ \mathbf{a}_i &= \phi_i(\boldsymbol{s}_i), \end{aligned}$$

where ϕ_i is an element-wise non-linear function and $\mathbf{W}_i$ is a weight matrix.

We will define $\boldsymbol{\theta} = [\mathrm{vec}(\mathbf{W}_0)\mathrm{vec}(\mathbf{W}_0) \cdots \mathrm{vec}(\mathbf{W}_l)]$, which is the vector consisting of all the network's parameters concatenated together, where vec is the operator which vectorizes matrices by concatenating their rows horizontally.

We will use the non-differentiable zero-one loss $\ell_{01}(f, x, y) = \mathbb{I}(\arg\max(f(x)) = y)$, and categorical cross-entropy, which is a commonly used differentiable surrogate $\ell_{\mathrm{cat}}(f, x, y) = -\sum_i \mathbb{I}[i = y] \log(f(x)_i)$, where we assume that the outputs of f are normalized using a softmax function to form a probability distribution.

We will also use the following PAC-Bayes formulation, by Catoni (2007). Note that this is strictly tighter than the formulation of Chapter 1 1.4.1 (Zhou et al., 2018). We get

Theorem 2.2.1. *Catoni (2007) Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, a hypothesis set $\mathcal{F}$, a loss function $\ell' : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to [0, 1]$, a prior distribution π over $\mathcal{F}$, a real number $\delta \in (0, 1]$, and a real number $\beta > 0$, with probability at least $1 - \delta$ over the choice of $(X, Y) \sim \mathcal{D}^n$, we have*

$$\begin{aligned} \forall \hat{\rho} \text{ on } \mathcal{F} : \mathbf{E}_{f\sim\hat{\rho}} \mathcal{L}^{\ell'}_{\mathcal{D}}(f) \leq & \Phi^{-1}_{\beta}(\mathbf{E}_{f\sim\hat{\rho}} \hat{\mathcal{L}}^{\ell'}_{X,Y}(f) \\ & + \frac{1}{\beta n}(\mathrm{KL}(\hat{\rho}||\pi) + \ln \frac{1}{\delta})), \end{aligned} \tag{2.2}$$

where $\Phi^{-1}_{\beta}(x) = \frac{1-e^{-\beta x}}{1-e^{-\beta}}$.

The above PAC-Bayes theorem works with bounded loss functions and as such is typically evaluated with the zero-one loss ℓ_{01}. However, one might want to optimize the above

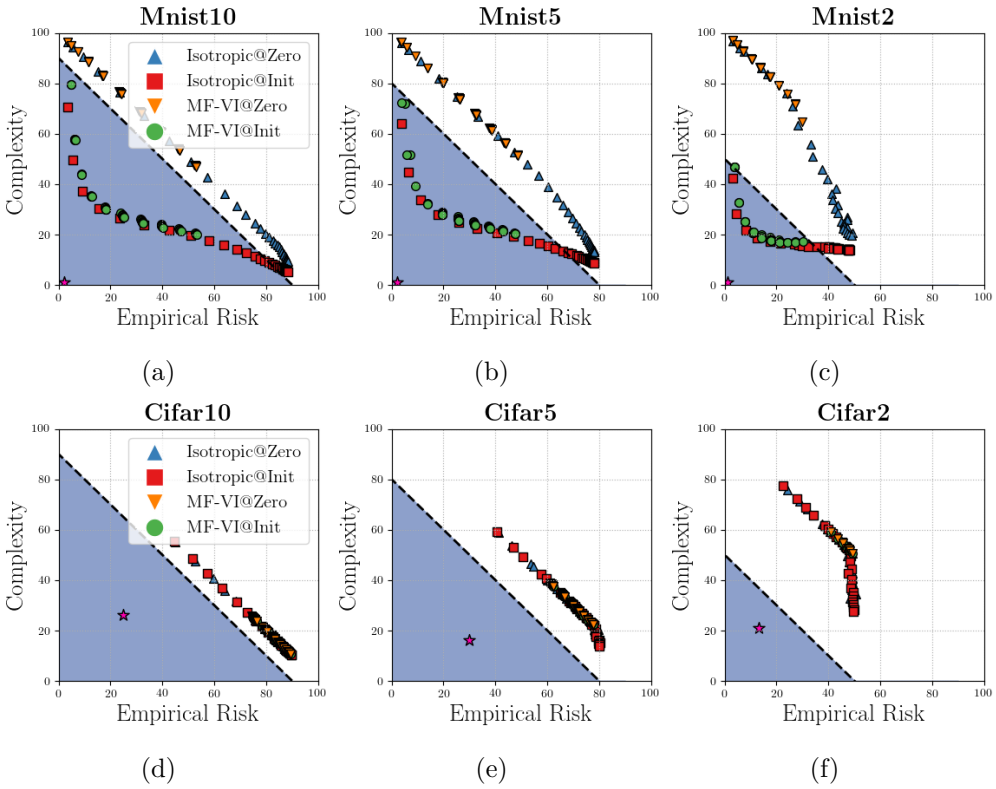


Figure 2.2 – **Detailed comparison of posterior and prior choices**: The area below the dashed line corresponds to non-vacuous pairs of (complexity, empirical risk). The purple star corresponds to the optimal bound implied by the testing set. For the MNIST case there is a significant improvement when changing from a prior centered at 0 to a prior centered at the random initialization. The baseline isotropic bounds are non-vacuous for a prior centered at 0. Optimizing the mean-field approximation using variational inference provides no improvements over the baseline, regardless of prior choice. In the CIFAR case all modeling choices result in vacuous bounds.

bound as proposed in Dziugaite and Roy (2017). One approach, is to then parametrize $f_{\boldsymbol{\theta}}$ using diagonal Gaussians as $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\sigma}_{\hat{\rho}})$ and the prior as $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\mathbf{I})$. Then, one can use the reparametrization trick $\boldsymbol{\theta} = \boldsymbol{\mu}_{\hat{\rho}} + \sqrt{\boldsymbol{\sigma}_{\hat{\rho}}} \odot \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the categorical cross-entropy to optimize the surrogate

$$\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}\hat{\mathcal{L}}^{\ell_{\text{cat}}}_{X,Y}(f_{\boldsymbol{\theta}}) + \frac{1}{\beta n}(\text{KL}(\hat{\rho}(\boldsymbol{\theta})||\mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\mathbf{I})) + \ln\frac{1}{\delta}), \tag{2.3}$$

for $\boldsymbol{\mu}_{\hat{\rho}}$, $\boldsymbol{\sigma}_{\hat{\rho}}$. In practice, one optimizes (2.3), but wants to evaluate (3.20). It's also often beneficial to fine tune λ and we want to approximate $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell_{01}}_{X,Y}(f)$ with an empirical estimate. We take a union bound over values of λ, and apply a Chernoff bound for the tail of the empirical estimate of $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell_{01}}_{X,Y}(f)$. Putting everything together, as proposed in Dziugaite and Roy (2017), one can obtain valid PAC-Bayes bounds subject to a posterior

distribution $\hat{\rho}^*(\boldsymbol{\theta})$ that hold with probability at least $1-\delta-\delta'$ and are of the form

$$\begin{aligned}\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}^*(\boldsymbol{\theta})}\mathcal{L}_{\mathcal{D}}^{\ell_{01}}(f_{\boldsymbol{\theta}}) \leq & \Phi_{\beta}^{-1}(\tilde{\mathcal{L}}_{X,Y}^{\ell_{01}}(f_{\boldsymbol{\theta}}) + \frac{1}{\beta n}\mathrm{KL}(\hat{\rho}^*(\boldsymbol{\theta})||\pi) \\ & + \frac{1}{\beta n}\ln(\frac{\pi^2 b^2 \ln(c/\lambda)^2}{6\delta}) + \sqrt{\frac{\ln\frac{2}{\delta'}}{m}}),\end{aligned} \tag{2.4}$$

where $\Phi_{\beta}^{-1}(x) = \frac{1-e^{-\beta x}}{1-e^{-\beta}}$. Also c, b are constants, m is the number of samples from $\hat{\rho}$ for approximating $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}_{X,Y}^{\ell_{01}}(f)$ and $\tilde{\mathcal{L}}_{X,Y}^{\ell_{01}}(f_{\boldsymbol{\theta}})$ the empirical estimate.

It is not difficult to see, that for a high enough number of samples n and m, the terms in line 2 of (2.4) have a negligible effect on the bound. All proofs are deferred to the Appendix B.1. A discussion on some fine points of PAC-Bayes bounds can be found in Appendix B.6.

2.3 Empirical results

We tested 6 different datasets. These consist of the original MNIST-10 and CIFAR-10 (Krizhevsky and Hinton, 2010) datasets, as well as simplified versions, where we collapsed the 10 classes into 5 and 2 aggregate classes, potentially simplifying the classification problem. All had 50000 training samples. We test the architectures

$$\text{input} \to 300\text{FC} \to 300\text{FC} \to \#classes\text{FC} \to \text{output}$$

on MNIST, and

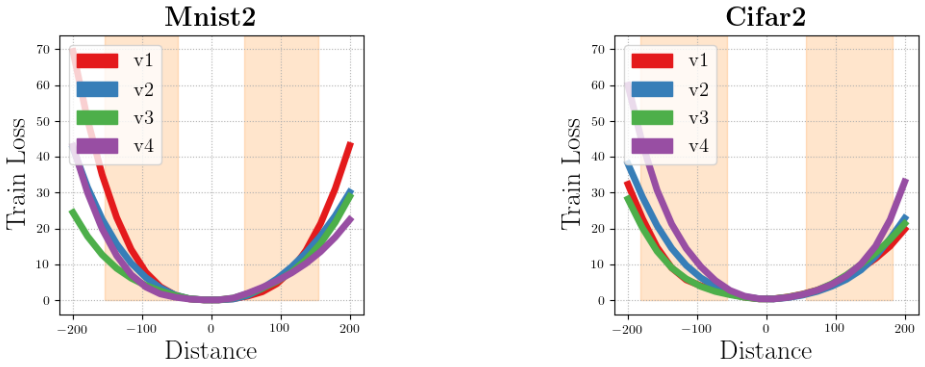
$$\text{input} \to 200\text{FC} \to 200\text{FC} \to \#classes\text{FC} \to \text{output}$$

on CIFAR, where xFC denotes a fully connected layer with x neurons.

We also tested four combinations of prior and posterior

1. $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda\mathbf{I})$, $\pi(\boldsymbol{\theta}) = \mathcal{N}(0, \lambda\mathbf{I})$
2. $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda\mathbf{I})$, $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\text{init}}, \lambda\mathbf{I})$
3. $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\sigma}_{\hat{\rho}})$, $\pi(\boldsymbol{\theta}) = \mathcal{N}(0, \lambda\mathbf{I})$.
4. $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\sigma}_{\hat{\rho}})$, $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\text{init}}, \lambda\mathbf{I})$.

Isotropic posterior. Isotropic combinations 1 and 2 differ only in the prior mean. The first prior is centered at 0, while the second prior is centered at the random deep neural



(a) (b)

Figure 2.3 – **Empirical evaluation of the categorical cross-entropy loss**: We take normalized random directions $\mathbf{v}_i,\ i \in \{1,2,3,4\}$ and plot the deterministic categorical cross-entropy loss $\hat{\mathcal{L}}^{\ell_{\text{cat}}}_{X,Y}(f_{\boldsymbol{\theta}})$ for MNIST2 and CIFAR2 and values on the line $\boldsymbol{\theta} = \boldsymbol{\theta}_* + t\mathbf{v}_i,\ t \in [-200, 200]$. We see that the loss closely reassembles a quadratic around the minimum $\boldsymbol{\theta}_*$. High dimensional Gaussian vectors concentrate close to a hypersphere centered on the mean. We find the radius of the hyperspheres and shade the corresponding 1 dimensional cross sections in the plots. Posteriors relevant to our experiments concentrate within an area well approximated by the quadratic.

network initialization. In practice, to derive multiple (complexity, empirical risk) pairs we sample λ,β in the range $\lambda \in [0.031, 0.3]$ and $\beta \in [1, 5]$. For these we compute $\hat{\mathcal{L}}(\hat{\rho})$ and $\mathrm{KL}(\hat{\rho}||\pi)$. The second can be computed analytically, while we approximate the first using Monte Carlo sampling with $m = 1000$ samples from $\hat{\rho}$. We then plug the results into (2.4). We set the estimated complexity as Complexity $\equiv [\Phi^{-1}_{\beta^*}(\hat{\mathcal{L}}(\hat{\rho}^*) + \frac{1}{\beta^* n}\mathrm{KL}(\hat{\rho}^*||\pi)) - \hat{\mathcal{L}}(\hat{\rho}^*)]$, where β^* is the optimal β.

Diagonal posterior (VI). Combinations 3 and 4 correspond to a posterior with diagonal covariance and a non-informative prior centered at 0 and at the random initialization. For MNIST we do a grid search over $\beta \in [1, 5]$ and $\lambda \in [0.03, 0.1]$ while for CIFAR we search in $\beta \in [1, 5]$ and $\lambda \in [0.1, 0.3]$. For each (β, λ) pair we optimize $\boldsymbol{\sigma}_{\hat{\rho}}$ using the surrogate (2.3). Specifically, we use the state of the art Flipout estimator (Wen et al., 2018). We used 5 epochs of training using the Adam optimizer (Kingma and Ba, 2014) with a learning rate of $1e-1$. Increasing the number of epochs didn't affect the results. We calculate the complexity and empirical risk as in the isotropic case.

We plot the Pareto fronts of all modeling choices in Figure 2.2. For the case of MNIST, changing from the prior centered at 0 to the prior centered at the random initialization resulted in a significant improvement of the bound. The resulting bounds with a prior at the random initialization are non-vacuous, even for the simple isotropic posterior. Optimizing the covariance with VI yields negligible or no improvements, regardless of the prior choice.

For CIFAR, we do not see significant variation in the bounds. The Catoni bound has a saturating effect above the line $y = 1 - x$, s.t. $x \in [0, 1]$. All (complexity,empirical risk) pairs fall into this saturating region. Specifically, mean-field VI fails to meaningfully improve the bound. Looking at the optimal bound points (star shapes), one explanation for the difference with MNIST, is that CIFAR DNNs have overfit the data significantly.

The full parameter choices for the experiments can be found in Appendix B.5.

2.4 Quadratic Approximation

The stochastic and non-convex objective (2.3) is difficult to analyze theoretically. As such we first propose to approximate the cross-entropy loss at the mean of the posterior using a second order Taylor expansion which will make the subsequent analysis tractable (this corresponds to a *Laplace* approximation (Bishop, 2006) to the posterior). We introduce the centered random variable $\boldsymbol{\eta} = \boldsymbol{\theta} - \mathbf{E}[\boldsymbol{\theta}]$ so that $\boldsymbol{\eta} \sim \hat{\rho}'(\boldsymbol{\theta})$, we get

$$\begin{aligned} C_\beta(X, Y; \hat{\rho}, \pi) &= \mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}\hat{\mathcal{L}}^{\ell_{\text{cat}}}_{X,Y}(f_{\boldsymbol{\theta}}) + \beta \text{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) \\ &\approx \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\boldsymbol{\eta}^T\nabla\hat{\mathcal{L}}^{\ell_{\text{cat}}}_{X,Y}(f_{\boldsymbol{\theta}}) + \frac{1}{2}\boldsymbol{\eta}^T\nabla^2\hat{\mathcal{L}}^{\ell_{\text{cat}}}_{X,Y}(f_{\boldsymbol{\theta}})\boldsymbol{\eta}] + \beta \text{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) \\ &\approx \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}\boldsymbol{\eta}] + \beta \text{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})). \end{aligned} \tag{2.5}$$

where $\mathbf{H} \equiv \nabla^2\hat{\mathcal{L}}^{\ell_{\text{cat}}}_{X,Y}(f_{\boldsymbol{\theta}})$ is the Hessian and captures the curvature at the minimum.

In the above we made a number of assumptions. First, we assumed that the gradient at the point of expansion is zero. For a well trained overparametrized DNN this is a reasonable assumption. Secondly, we omit terms of the Taylor expansion of order ≥ 3. This results in a quadratic approximation. We conduct experiments to see whether this is reasonable. Specifically, we take random directions along the loss landscape and plot along them the value of the loss. We see in Figure 2.3 that the loss is indeed approximately quadratic around the minimum. At the same time, we note that approximating the loss as quadratic has been used to obtain state of the art results in the DNN compression literature (Dong et al., 2017; Wang et al., 2019; Peng et al., 2019; LeCun et al., 1990; Hassibi and Stork, 1993).

For the expectation of the quadratic loss to be a good approximation of the expectation of the categorical loss, the mass of the posterior has to be concentrated at locations where the true loss is well approximated by a quadratic. We have thus far dealt with Gaussian posteriors $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\sigma}_{\hat{\rho}})$, where $\forall i,\ \sigma_{\hat{\rho}i} \approx \lambda$, $0.01 \leq \lambda \leq 1$. It is well know that Gaussians in high dimensions concentrate on a thin "bubble" away from the origin. We can use this intuition and make a rough calculation of the radius of this bubble (Vershynin, 2018). Specifically, assuming that $\forall i,\ \boldsymbol{\sigma}_{\hat{\rho}i} = \lambda$, we can calculate

$\mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}||\boldsymbol{\eta}||_2^2 = \mathbf{E}_{\boldsymbol{\eta}\sim\mathcal{N}(0,\boldsymbol{\sigma}_{\hat{\rho}})}[\sum_{i=0}^{d}\eta_i^2] = \sum_{i=0}^{d}\sigma_{\hat{\rho}i} = \lambda d$. Finally we expect that the radius of the "bubble" is $\mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}||\boldsymbol{\eta}||_2 \approx \sqrt{\lambda d}$. We plot these regions in Figure 2.3. We see that posteriors concentrate within areas where the quadratic approximation is reasonable.

2.4.1 Optimal Posterior

Compared to the diagonal modeling of the previous section, we now make the slightly more general modeling choices $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\Sigma}_{\hat{\rho}})$ and $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\boldsymbol{\Sigma}_{\pi})$. We can then show that the optimal posterior covariance of the objective (2.5) for fixed prior and posterior means has a closed form solution.

Lemma 2.4.1. *The optimization problem* $\min_{\boldsymbol{\Sigma}_{\hat{\rho}}} \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}\boldsymbol{\eta}] + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta}))$ *where* $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\Sigma}_{\hat{\rho}})$ *and* $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\boldsymbol{\Sigma}_{\pi})$ *is convex and is minimized at*

$$\boldsymbol{\Sigma}_{\hat{\rho}}^{*} = \beta(\mathbf{H} + \frac{\beta}{\lambda}\boldsymbol{\Sigma}_{\pi}^{-1})^{-1}, \tag{2.6}$$

where $\mathbf{H} \equiv \nabla^2\hat{\mathcal{L}}_{X,Y}^{\ell_{cat}}(f_{\boldsymbol{\theta}})$ *captures the curvature at the minimum, while* $\boldsymbol{\Sigma}_{\pi}$ *is the prior covariance.*

The full derivation is deferred to Appendix B.2.

2.4.2 Optimal Prior

We can relax the modeling choices further by noting that PAC-Bayesian theory allows one to choose an informative prior, with the restriction that the prior can only depend on the data generating distribution and *not* the training set. A number of previous works (Parrado-Hernández et al., 2012; Catoni, 2003; Ambroladze et al., 2007) have used this insight mainly on simpler linear settings and usually by training a classifier on a separate training set and using the result as a prior. Recently, Dziugaite and Roy (2018a) have proposed to use the original training set to derive valid priors by imposing differential privacy constraints.

We ignore these concerns for the moment, and optimize the prior covariance directly. The objective is non-convex, however for the case of *diagonal* prior and posterior covariances we can find the global minimum.

Lemma 2.4.2. *For* $\min_{\boldsymbol{\sigma}_{\hat{\rho}},\boldsymbol{\sigma}_{\pi}} \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}\boldsymbol{\eta}] + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta}))$ *with* $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\sigma}_{\hat{\rho}})$ *and* $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\boldsymbol{\sigma}_{\pi})$*, the optimal prior and posterior covariances have elements*

$$(\sigma_{\hat{\rho}i}^{*})^{-1} = \frac{1}{2\beta}[h_i + \sqrt{h_i^2 + \frac{4\beta h_i}{(\mu_{i\hat{\rho}} - \mu_{i\pi})^2}}], \tag{2.7}$$

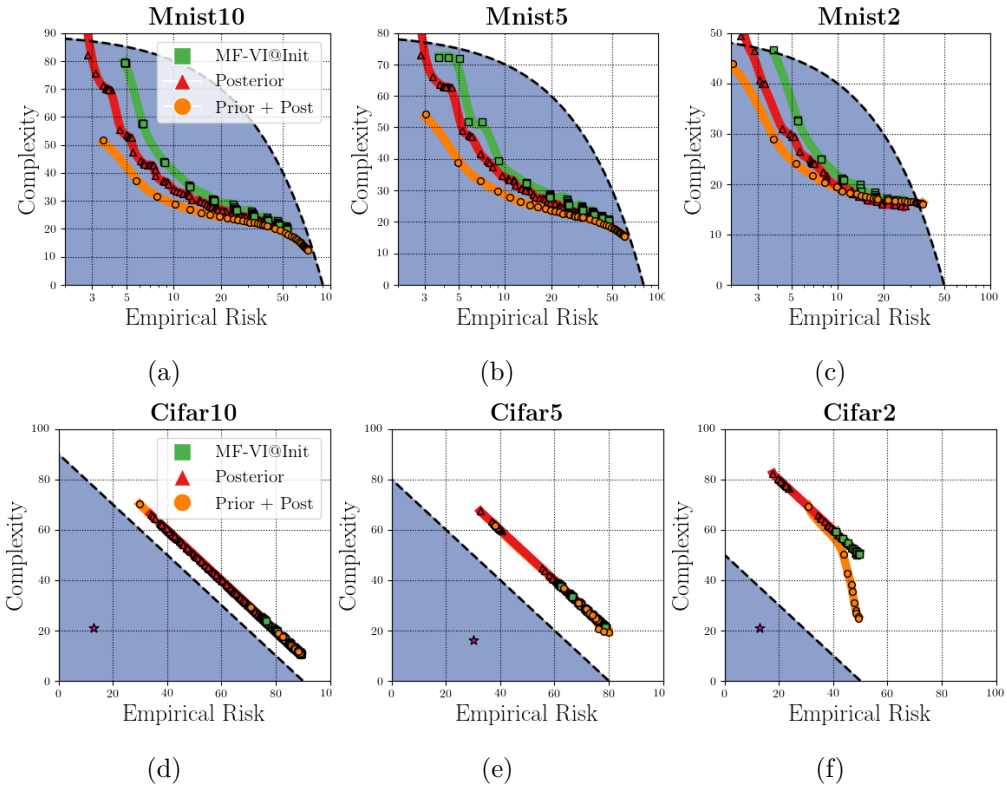


Figure 2.4 – **Closed form posterior and prior**: We plot the results obtained by mean-field variational inference, as well as the closed form bounds with optimized posterior and jointly optimized posterior and prior covariances. For MNIST, we plot the empirical risk in logarithmic scale for ease of exposition. Valid bounds where we only optimize the posterior in closed form get significant benefits over VI of between 5-10%. Optimizing the prior results in further improvements of 5-10%, implying that in theory better priors can be found. The results are far from tight even when optimizing the prior and for CIFAR all bounds are vacuous. This implies inherent limitations of the mean-field approximation, as we typically don't even have access to the optimal prior covariance.

$$(\sigma^*_{\pi i})^{-1} = \frac{\lambda}{2\beta}\left[\sqrt{h_i^2 + \frac{4\beta h_i}{(\mu_{i\hat{\rho}} - \mu_{i\pi})^2}} - h_i\right], \tag{2.8}$$

where $\mathbf{H} \equiv \nabla^2 \hat{\mathcal{L}}^{\ell_{cat}}_{X,Y}(f_{\boldsymbol{\theta}})$ *captures the curvature at the minimum.*

The full derivation is deferred to Appendix B.3. We *cannot* prove generalization using this result. Rather we use it as a sanity check for what is achievable through the mean-field approximation and an optimal informative prior covariance.

To approximate the Hessian we note that for the cross entropy loss and the softmax activation function $p(y = c|f_{\boldsymbol{\theta}}) = \exp(f_{\boldsymbol{\theta}}(\boldsymbol{x})_c)/\sum_i \exp(f_{\boldsymbol{\theta}}(\boldsymbol{x})_i)$ the Fisher Information matrix coincides with the generalized Gauss-Newton approximation of the Hessian

(Kunstner et al., 2019). We sample one output $\tilde{y}_i$ from the model distribution $p(y_i|f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))$ for each input $\boldsymbol{x}_i$, and approximate $\mathbf{H} \approx \sum_{i=0}^{n} \nabla_{\boldsymbol{\theta}} \log p(\tilde{y}_i|f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)) \nabla_{\boldsymbol{\theta}} \log p(\tilde{y}_i|f_{\boldsymbol{\theta}}(\boldsymbol{x}_i))^{\mathrm{T}}$, retaining only the diagonal elements.

Keeping the posterior and prior means fixed, we optimize the posterior covariance, as well as the posterior and *prior* covariance jointly in closed form. We plot the results in Figure 2.4, using the same approach as section 2.3 with $m = 1000$ for (2.7),(2.8), $m = 100$ for (2.6) and sampling over β and λ. For MNIST, valid bounds where we only optimize the posterior in closed form get significant benefits over VI of between 5-10%. Thus, even though Adam is very robust to hyperparameter selection, and the Flipout estimator is state of the art, one might look to hyperparameter tuning for better results. We present arguments in the next section, that hold also for the mean-field case, as to why it should be beneficial to avoid hyperparameter tuning. Invalid bounds where we optimize the prior and posterior jointly result in further improvements of 5-10%, implying that in theory better priors can be found. The bounds are far from tight, even when optimizing the prior, and for CIFAR all bounds are vacuous. This implies that the mean-field approximation is limited in the bound improvements it can provide.

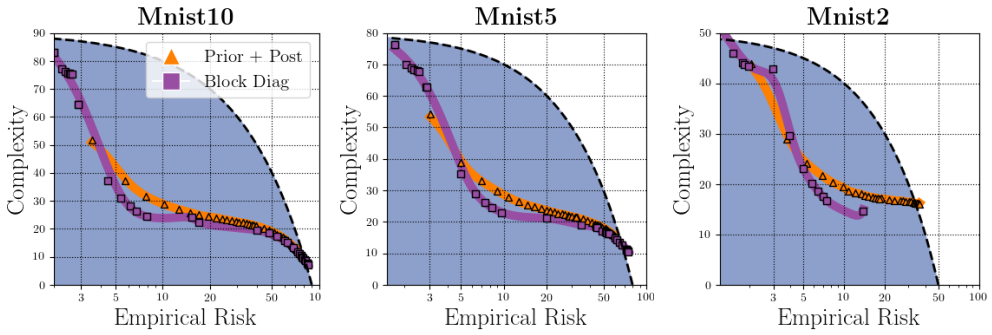
2.5 Beyond the mean-field approximation

2.5.1 Computational Issues

Given the implied shortcomings of the mean-field approximation it is interesting to look at richer posterior distributions. A number of approximations exist to model such posteriors. In Mishkin et al. (2018), the authors model the covariance as having a low-rank + diagonal structure. In normalizing flows (Rezende and Mohamed, 2015) a simple initial density is transformed into a more complex one, by applying a sequence of invertible transformations, until a desired level of complexity is attained. In K-FAC (Martens and Grosse, 2015), the Hessian can be approximated as a Khatri-Rao product to construct a Laplace approximation of the posterior (Ritter et al., 2018). Considerable effort has been placed into

Optimizing multiple variational objectives. To obtain Pareto fronts we will perform a grid search over λ and β, corresponding to $\mathcal{O}(10^2)$ classifiers with different empirical risk and complexity. Optimizing variational objectives is known to be unstable, to scale badly and to require extensive hyperparameter tuning Wu et al. (2018). Optimizing each posterior using SGD as in Mishkin et al. (2018); Rezende and Mohamed (2015), for even a few minutes, can add several hours to obtaining the full grid. Hyperparameter tuning objectives that do not converge can quickly make the task infeasible.

Sampling efficiently from the posterior. At the same time we will need to sample efficiently between $\mathcal{O}(10^4)$ and $\mathcal{O}(10^5)$ posterior samples. This is because we will be



(a) (b) (c)

Figure 2.5 – **Beyond the mean field approximation**: We compared the simplified K-FAC curvature approximation to the closed form *invalid* mean-field inference. Invalid results correspond to an optimal prior and posterior covariance to which we don't typically have access. For medium to low empirical risk the block diagonal curvature improves the bound for MNIST10-5-2 by $8.2\%, 7.5\%, 4.4\%$ respectively.

applying a Chernoff bound on the tail of the empirical risk. For flow based methods, the KL term also has to be approximated with MC sampling.

In the non-flow based methods, one typically seeks to factor $\Sigma = LL^T$. Then $y = Lz$, where z is standard normal, has the appropriate distribution, and can be sampled efficiently. While Mishkin et al. (2018) provide an efficient Cholesky factorization of their low-rank + diagonal approximation, the Khatri-Rao product (Martens and Grosse, 2015) has no obvious Cholesky factorization. Finally inference time in flow based methods will be influenced by the number of mappings used in the flow.

Simplified K-FAC Laplace. We assume a multiclass classification problem with c classes, and that the labels y are one-hot encoded. We then define the mean square error loss $\ell_{\text{mse}}(f, x, y) = (1/c)\sum_{i=0}^{c}(f(x)_i - y_i)^2$. Assuming r neurons per layer, $\boldsymbol{\theta}$ has a form $\boldsymbol{\theta} = [\text{vec}(\mathbf{W}_0^{0,:})\text{vec}(\mathbf{W}_0^{1,:}) \cdots \text{vec}(\mathbf{W}_l^{r,:})]$. We also denote for layer i and neuron j, $\boldsymbol{\theta}_{ij}$, $\boldsymbol{\mu}_{\hat{\rho}ij}$, $\boldsymbol{\Sigma}_{\hat{\rho}ij}$, $\boldsymbol{\mu}_{\pi ij}$ the corresponding split variables. We can then motivate optimizing the following surrogate upper bound

Lemma 2.5.1. *Assuming negligible layerwise derivatives of order other than 2, the differentiable surrogate objective*

$$\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}\hat{\mathcal{L}}_{X,Y}^{\ell_{mse}}(f_{\boldsymbol{\theta}}) + \frac{1}{\beta n}(\text{KL}(\hat{\rho}(\boldsymbol{\theta})||\mathcal{N}(\boldsymbol{\mu}_\pi, \lambda\mathbf{I})) + \ln\frac{1}{\delta}), \tag{2.9}$$

has the following upper bound

$$\begin{aligned}&\sum_{i,j}[\mathbf{E}_{\boldsymbol{\eta}_{ij}\sim\hat{\rho}'_{ij}(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}_{ij}^T\mathbf{H}_i\boldsymbol{\eta}_{ij}] + \frac{1}{\beta n}\text{KL}(\hat{\rho}_{ij}(\boldsymbol{\theta})||\pi_{ij}(\boldsymbol{\theta})]\\ &+ \mathcal{O}(c^l),\end{aligned} \tag{2.10}$$

where $\hat{\rho}_{ij}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}ij}, \boldsymbol{\Sigma}_{\hat{\rho}ij})$, $\pi_{ij}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi ij}, \lambda\mathbf{I})$, $\mathbf{H}_i = (1/n)\sum_{k=0}^{n} \mathbf{a}_i^k \mathbf{a}_i^{kT}$, *are neuronwise posteriors, priors and Hessians.*

The above corresponds to a greatly simplified version of K-FAC, where each layer has a posterior with covariance $\boldsymbol{\Sigma}_i = \mathbf{H}_i \otimes \mathbf{I}$, i.e. we assume correlations only for parameters in each neuron. While this approximation covers our needs, one could in principle use the slightly more expressive $\boldsymbol{\Sigma}_i = \mathbf{H}_i \otimes \mathbf{G}_i$, where $\mathbf{G}_i = \mathbf{E}[\mathbf{g}_i \mathbf{g}_i^T]$ and $\mathbf{g}_i$ are the backpropagated layerwise errors for layer i (Ritter et al., 2018), and we note that optimizing more expressive versions of K-FAC efficiently is an active area of research (Zhang et al., 2018; Bae et al., 2018). We've broken the original into many much smaller subproblems. We can now compute the Hessian efficiently and in a stable way once, and then sample the posterior efficiently in closed form at different variance levels λ. The full derivation is deferred to Appendix B.4.

2.5.2 Empirical Results

We now present results on the MNIST datasets. We run a grid search over β and λ, with 20 samples each, for $\beta \in [0.001, 0.02]$ and $\lambda \in [0.001, 0.1]$. We use $m = 1000$ samples for estimating the empirical risk. For computing the Pareto fronts we optimize (2.10) with (2.6) and evaluate (2.4) following the procedure of Section 2.3. The running time for each experiment was 33h, 30h and 25h respectively.

We plot the results in Figure 2.5 where we compare with the case of jointly optimized diagonal prior and posterior. At very low and very high empirical risk levels the complexity estimates saturate. However, for medium empirical risk levels the block diagonal covariance yields significant improvements to the bounds. The effect is more pronounced on the more difficult MNIST10 and MNIST5 experiments, where using the block diagonal posterior results in a decrease in the estimated complexity of $\sim 10\%$.

2.6 Discussion and Summary

In this chapter we moved from the analytically derived bounds of Chapter 1 to bounds computed using an optimization procedure based on the work of Dziugaite and Roy (2017). These present a significant improvement over bounds in Chapter 1, with estimates becoming non-vacuous in a number of cases. Intuitively, this is because we make significantly tighter estimates of the amount of noise that we can add to the deep neural network weights before hurting accuracy. However, somewhat counterintuitively, we have seen that the main empirical gains in bound tightness are the result of choosing the prior mean to be centered at the random initialization, compared to at the origin. Further optimizing the posterior covariance using mean-field variational inference results in negligible gains. Thus taking the above into account we slightly relax the model so

that the prior and posterior distributions are not diagonal but block diagonal resulting in significant bound improvements. This illustrates the potential for improving bounds by using more expressive priors and posteriors. At the same time, given the significant gains from choosing the prior mean correctly, in the next chapter we will look at computational approaches to deriving informative prior means.

3 Differential Privacy based Generalization Bounds.

In the previous chapter we saw how computing PAC-Bayes bounds numerically led to significant improvements over computing them analytically. At the same time, we saw that the actual source of tightness and non-vacuity included some subtleties. In particular, modeling priors and posteriors with diagonal Gaussians appeared to be very restrictive, while moving to more expressive posteriors resulted in significant gains. Finally, the most important element in tightening the bounds was found to be choosing a good prior mean. In this chapter we elaborate on this point and analyze methods through which one can compute informative prior means, which significantly tighten bounds.

3.1 Introduction

Recall that PAC-Bayes bounds deal with randomized classifiers, with posterior and prior distributions $\hat{\rho}$ and π respectively and make statements that are roughly of the form

$$\mathbf{E}\mathcal{L}(\hat{\rho}) \leq \mathbf{E}\hat{\mathcal{L}}(\hat{\rho}) + \beta \mathrm{KL}(\hat{\rho}||\pi), \tag{3.1}$$

where $\mathcal{L}(\hat{\rho})$ is the risk, $\hat{\mathcal{L}}(\hat{\rho})$ is the empirical risk and the expectation is over the posterior. The $\beta\mathrm{KL}(\hat{\rho}||\pi)$ term between the prior and posterior acts as a measure of complexity for the classifier.

The terms $\mathbf{E}\hat{\mathcal{L}}(\hat{\rho})$ and $\beta\mathrm{KL}(\hat{\rho}||\pi)$ are typically inversely related. For example, with minor modifications, one can model the posterior and prior as $\hat{\rho} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda\mathbf{I})$ and $\pi = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\mathbf{I})$ so that $\beta\mathrm{KL}(\hat{\rho}||\pi) = (\beta/2\lambda)||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi}||_2^2$. Then, $\beta\mathrm{KL}(\hat{\rho}||\pi) \to 0$ as $\lambda \to +\infty$, however for such a choice value of the term $\mathbf{E}\hat{\mathcal{L}}(\hat{\rho}) = \int_{\Omega} \hat{\mathcal{L}}(\omega) d\hat{\rho}(\omega)$ will increase significantly, as we will average the classification loss over increasingly larger regions of the loss landscape.

In Chapter 1 we saw that finding analytically the value of the variance in the prior and posterior distributions that balances the above terms results in vacuous bounds (Neyshabur et al., 2017b). By contrast in Chapter 2, based on the works Dziugaite and

Roy (2017) and Zhou et al. (2018) we saw how optimizing the posterior distribution using an iterative algorithm so as to improve the bound as much as possible, led to significant gains and non-vacuity.

In fact, as noted in Chapter 2 most of the gains in the literature up to now have been due to decreasing the l_2 distance between the prior and posterior means $||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi}||_2^2$. Specifically, simply by choosing the prior mean to be centered at the random neural network initialization instead of at the origin improves the bounds significantly, as $||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\text{init}}||_2^2 \ll ||\boldsymbol{\mu}_{\hat{\rho}} - 0||_2^2$.

One could look at richer covariance structures and the literature on the subject is steadily growing (Mishkin et al., 2018; Lin et al., 2019b,a; Ritter et al., 2018; Zhang et al., 2018; Bae et al., 2018; Sun et al., 2019; Louizos et al., 2019), together with new optimization libraries (Dangel et al., 2019, 2020). However, significant problems still exist (Pitas, 2020) in i) optimizing deep neural networks in this manner efficiently ii) avoiding tedious hyperparameter tuning iii) sampling efficiently from the result posteriors iv) estimating the KL term efficiently in the case where closed form solutions do not exist.

Research objectives. We thus focus on learning better prior *means* $\boldsymbol{\mu}_{\pi}$, potentially decreasing further the term $||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi}||_2^2$. Under the PAC-Bayesian assumptions, the prior cannot depend on the training set however it can depend on the training data *distribution*. Usually, this is exploited in the literature by training a prior using a set of separate datapoints (Parrado-Hernández et al., 2012; Lever et al., 2013). In Dziugaite and Roy (2018a) the authors propose a different approach based on differential privacy (Dwork et al., 2014). One trains a prior on the original training set by adding noise to the learning procedure. Thus, intuitively this ensures that the amount of information from any *specific* data sample retained by a learned prior remains small and the prior is provably only *distribution* dependent.

Unfortunately, this and a subsequent work Dziugaite and Roy (2018b) are plagued by the use of Stochastic Gradient Langevine Dynamics (SGLD) (Welling and Teh, 2011) to compute the required prior. One of the main assumptions for the bounds to be valid is that SGLD has mixed properly, a condition that is difficult to assess in practice. At the same time, SGLD does not provide the prior in closed form and the KL complexity term has to be evaluated numerically, which introduces further looseness.

Contributions. We take a step back and reassess valid PAC-Bayes priors using differential privacy by opting for the state of the art differential privacy approach of Abadi et al. (2016), instead of SGLD. Crucially, there are no assumptions on the mixing time of the procedure, that are difficult to evaluate, and the privacy of the learned prior classifier is evaluated numerically using an efficient and stable procedure (Mironov et al., 2019).

Our results can be summarized as follows

1. With a slight modification, by setting the posterior $\hat{\rho}$ equal to the distribution dependent private prior π, we obtain a PAC-Bayes bound where the dominant complexity term is the privacy parameter ϵ. This, coupled with the approach of Abadi et al. (2016) results in state of the art results in a number of datasets.
2. Further optimizing the *posterior* mean gives mixed results. While for MNIST it results in even tighter bounds, for CIFAR it loosens the bounds. In all cases the required optimization procedure needs significant finetuning.
3. We illustrate using a simple theoretical model how the classification accuracy of a differentially private classifier is affected by the intrinsic dimensionality of the data distribution and the privacy parameter ϵ. We relate this analysis to the deep neural network setting, and to limitations to the tightness of generalization bounds based on differential privacy.

3.2 PAC-Bayes and differential privacy

We first provide a definition for differential privacy

Definition 3.2.1. A randomized algorithm $\mathscr{P} : \mathcal{Z}^n \to T$ is (ϵ, δ)-differentially private if, for all datasets $Z, Z' \in \mathcal{Z}^n$ that differ at only one sample, and all measurable subsets $B \subseteq T$, we have $\mathbb{P}\{\mathscr{P}(Z) \in B\} \leq e^{\epsilon}\mathbb{P}\{\mathscr{P}(Z') \in B\} + \delta$.

Differential privacy is therefore a notion of algorithmic stability. Roughly, it means that a learning algorithm will have an output that doesn't depend much on any specific data sample. An important property of differential privacy is that the composition of two differentially private algorithms is itself differentially private. In particular, a very simple formulation of this property is

Theorem 3.2.1. *For any $\epsilon > 0$ and $\delta \in [0, 1]$ if g is a (ϵ_g, δ_g)-differentially private mechanism and h is a (ϵ_h, δ_h)-differentially private mechanism then $g \circ h$ is $(\epsilon_g + \epsilon_h, \delta_g + \delta_h)$-differentially private.*

While this particular composition theorem is not optimal, it provides some easy intuition. In particular processing the output of a differentially algorithm using a second one, can only provide a limited further amount of information on any data sample. Importantly, as we will see later this property can be used for studying the privacy of iterative algorithms. In particular, differentially private Stochastic Gradient Descent (SGD) steps can themselves be composed so that their final output is differentially private.

With regards to generalization bounds we will again be using PAC-Bayes bounds which have proven to be tight empirically (Dziugaite and Roy, 2017; Zhou et al., 2018; Pitas, 2020). In particular, we will be again using a version due to Catoni Catoni (2007) which is tighter than common alternatives (Zhou et al., 2018).

We denote the learning sample $(X, Y) = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, that contains n input-output pairs. Samples (X, Y) are assumed to be sampled randomly from a distribution $\mathcal{D}$. Thus, we denote $(X, Y) \sim \mathcal{D}^n$ the i.i.d observation of n elements. We consider loss functions $\ell : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$, where $\mathcal{F}$ is a set of predictors $f : \mathcal{X} \to \mathcal{Y}$. We also denote the empirical risk $\hat{\mathcal{L}}^{\ell}_{X,Y}(f) = (1/n) \sum_i \ell(f, \boldsymbol{x}_i, y_i)$ and the risk $\mathcal{L}^{\ell}_{\mathcal{D}}(f) = \mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \ell(f, \boldsymbol{x}, y)$.

A neural network transforms its inputs $\mathbf{a}_0 = \boldsymbol{x}$ to an output $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{a}_l$ through a series of l layers, each of which consists of a bank of units/neurons. The computation performed by each layer $i \in \{1, ..., l\}$ is given as follows

$$\begin{aligned} \boldsymbol{s}_i &= \mathbf{W}_i \mathbf{a}_{i-1}, \\ \mathbf{a}_i &= \phi_i(\boldsymbol{s}_i), \end{aligned}$$

where ϕ_i is an element-wise non-linear function and $\mathbf{W}_i$ is a weight matrix.

We will define $\boldsymbol{\theta} = [\mathrm{vec}(\mathbf{W}_0)\mathrm{vec}(\mathbf{W}_0) \cdots \mathrm{vec}(\mathbf{W}_l)]$, which is the vector consisting of all the network's parameters concatenated together, where vec is the operator which vectorizes matrices by concatenating their rows horizontally.

We will use the non-differentiable zero-one loss $\ell_{01}(f, x, y) = \mathbb{I}(\arg\max(f(x)) = y)$, and the categorical cross-entropy, which is a commonly used differentiable surrogate $\ell_{\mathrm{cat}}(f, x, y) = -\sum_i \mathbb{I}[i = y] \log(f(x)_i)$, where we assume that the outputs of f are normalized using a softmax function to form a probability distribution. Catoni (2007) then get

Theorem 3.2.2. *Catoni (2007) Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, a hypothesis set $\mathcal{F}$, a loss function $\ell' : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to [0, 1]$, a prior distribution π over $\mathcal{F}$, a real number $\delta \in (0, 1]$, and a real number $\beta > 0$, with probability at least $1 - \delta$ over the choice of $(X, Y) \sim \mathcal{D}^n$, we have*

$$\begin{aligned} \forall \hat{\rho} \text{ on } \mathcal{F} : \mathbf{E}_{f\sim\hat{\rho}} \mathcal{L}^{\ell'}_{\mathcal{D}}(f) \leq & \Phi_{\beta}^{-1}(\mathbf{E}_{f\sim\hat{\rho}} \hat{\mathcal{L}}^{\ell'}_{X,Y}(f) \\ & + \frac{1}{\beta n}(\mathrm{KL}(\hat{\rho}||\pi) + \ln \frac{1}{\delta})), \end{aligned} \tag{3.2}$$

where $\Phi_{\beta}^{-1}(x) = \frac{1-e^{-\beta x}}{1-e^{-\beta}}$.

PAC-Bayes bounds such as the above require that the prior distribution not depend on the training set used to compute the posterior. As such, a common choice for the prior is the non-informative centered Gaussian with identity covariance $\pi(\boldsymbol{\theta}) = \mathcal{N}(0, \lambda \mathbf{I})$. As one usually wants to estimate the generalization of a deterministic neural network, the posterior is then commonly chosen to be $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \mathbf{I})$ where $\boldsymbol{\mu}_{\hat{\rho}}$ are weights of a deterministic deep neural network.

3.2.1 Differentially Private Priors

We now turn our attention to differentially private PAC-Bayes bounds. We will essentially restate a result by Dziugaite and Roy (2018a) for the case of the Catoni bound. Assume $p = r \times l$ number of parameters where r is the number of neurons per layer and l is the number of layers.

Theorem 3.2.3. *Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, a hypothesis set $\mathcal{F}$, a loss function $\ell' : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to [0,1]$, $\mathscr{P} : (\mathcal{X} \times \mathcal{Y})^n \to \mathscr{M}(\mathbb{R}^p)$ an ϵ-differentially private mechanism for choosing a data-dependent prior over $\mathcal{F}$, a real number $\delta \in (0,1]$, and a real number $\beta > 0$, with probability at least $1-\delta$ over the choice of $(X,Y) \sim \mathcal{D}^n$, we have*

$$\forall \hat{\rho} \in \mathscr{M}(\mathbb{R}^p) \text{ on } \mathcal{F} : \mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell'}_{\mathcal{D}}(f) \leq \Phi^{-1}_{\beta}(\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f) + \frac{1}{\beta}(\frac{\mathrm{KL}(\hat{\rho}||\mathscr{P}(X,Y)) + \ln\frac{2}{\delta}}{n} + \frac{\epsilon^2}{2} + \epsilon\sqrt{\frac{\ln(4/\delta)}{2n}})) \tag{3.3}$$

where $\Phi^{-1}_{\beta}(x) = \frac{1-e^{-\beta x}}{1-e^{-\beta}}$.

We defer the full derivation to Appendix C.1, noting that it closely matches the one presented in Dziugaite and Roy (2018a).

Bound with KL and privacy term

The main way of using differential privacy in the previous literature is to construct valid priors for the PAC-Bayesian bound. Therefore we can run the mechanism to obtain a valid prior mean $\boldsymbol{\mu}_{\mathscr{P}}$ and then construct a prior and posterior pair as $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda\mathbf{I})$ and $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\mathscr{P}}, \lambda\mathbf{I})$. We also need to expand the bound slightly to account for the λ parameter in the prior, however the effect is negligible, in particular δ is substituted by $\frac{6\delta}{\pi^2 b^2 \ln(c/\lambda^2)}$. Finally, we note that the Catoni bound is formulated for bounded loss functions. As such, we will be stating here a specific application using the zero-one loss ℓ_{01}. Taking everything into account, we get

$$\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell_{01}}_{\mathcal{D}}(f) \leq \Phi^{-1}_{\beta}(\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell_{01}}_{X,Y}(f) + \frac{1}{2\beta n\lambda}||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\mathscr{P}}||_2^2 + \frac{\epsilon^2}{2\beta} + \frac{\ln((2\pi^2 b^2 \ln(c/\lambda^2))/(6\delta))}{\beta n} + \frac{\epsilon}{\beta}\sqrt{\frac{\ln((4\pi^2 b^2 \ln(c/\lambda^2))/(6\delta))}{2n}}), \tag{3.4}$$

where we highlight the term that includes the distance between the prior and posterior means. One then typically further optimizes the above so as to obtain a posterior mean $\boldsymbol{\mu}_{\hat{\rho}}$ so that both $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell_{01}}_{X,Y}(f)$ and $\frac{1}{2\beta n\lambda}||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\mathscr{P}}||_2^2$ are low. We note that the zero-one

loss ℓ_{01} is not differentiable. As such, one can use the categorical cross-entropy loss ℓ_{cat} to optimize a surrogate differentiable loss function. Gradients for the stochastic objective can be derived using the reparametrization trick $\boldsymbol{\theta} = \boldsymbol{\mu}_{\hat{\rho}} + \sqrt{\boldsymbol{\sigma}_{\hat{\rho}}} \odot \mathcal{N}(\mathbf{0}, \mathbf{I})$. As our baseline we take a slightly different approach and optimize a *deterministic* objective to obtain the posterior mean. Putting everything together, one can optimize the surrogate objective

$$\hat{\mathcal{L}}^{\ell_{\text{cat}}}_{X,Y}(f(\boldsymbol{\mu}_{\hat{\rho}})) + \frac{1}{2\beta n\lambda}||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\mathscr{P}}||_2^2 + \text{constant} \tag{3.5}$$

to obtain a posterior mean $\boldsymbol{\mu}_{\hat{\rho}}$. The term $\frac{1}{2\beta n\lambda}||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\mathscr{P}}||_2^2$ here acts as a simple ℓ_2 regularizer on the norm of the learned weights. We can then evaluate a valid bound using equation (3.4) and posterior and prior distributions $\hat{\rho} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda\mathbf{I})$ and $\pi = \mathcal{N}(\boldsymbol{\mu}_{\mathscr{P}}, \lambda\mathbf{I})$ and different values of λ.

Bound based only on privacy

It is not obligatory to use the differentially private classifier simply as a prior. In fact, we can construct a prior and posterior pair as $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\mathscr{P}}, \lambda\mathbf{I})$ and $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\mathscr{P}}, \lambda\mathbf{I})$. This results in a valid PAC-Bayes bound, where the KL-divergence is trivially zero. We get

$$\begin{aligned}\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell'}_{\mathcal{D}}(f) \leq \Phi^{-1}_{\beta}(&\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f) \\ &+ \frac{\epsilon^2}{2\beta} + \frac{\ln((2\pi^2 b^2 \ln(c/\lambda^2))/(6\delta))}{n\beta} + \frac{\epsilon}{\beta}\sqrt{\frac{\ln((4\pi^2 b^2 \ln(c/\lambda^2))/(6\delta))}{2n}})\end{aligned} \tag{3.6}$$

or simplifying by keeping only the dominant term $\epsilon^2/2$

$$\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell'}_{\mathcal{D}}(f) \lesssim \Phi^{-1}_{\beta}(\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f) + \mathcal{O}(\frac{\epsilon^2}{2\beta}))$$

where $\Phi^{-1}_{\beta}(x) = \frac{1-e^{-\beta x}}{1-e^{-\beta}}$. Notice that the above holds for any value of $\lambda > 0$, as such we can choose a very small value so that the network becomes almost deterministic. The dominant complexity term then becomes the privacy parameter ϵ.

3.2.2 Computing differentially private classifiers

We see that essentially one needs to train neural networks that have low empirical risk and are at the same time ϵ-differentially private with a small ϵ constant. There are a number of ways in which one can derive a differentially private classifier. One of the

earliest and most common baselines is the *output* perturbation algorithm (Dwork, 2008). Here, one trains a classifier and then adds noise to the learned parameters according to the Gaussian or Laplace distribution. Unfortunately, this approach scales badly with the parameter dimensionality. In *objective* perturbation (Chaudhuri et al., 2011) one adds the noise as a regularization term to the objective and then trains the classifier. The empirical results improve upon output perturbation, however the privacy guarantees hold only for convex losses. Other works are only theoretical (Bassily et al., 2014), having very pessimistic analyses.

Stochastic Gradient Langevin Dynamics

The approach used in Dziugaite and Roy (2018a) circumvents the above restrictions by relying on Gibbs posteriors. We introduce some notation for Gibbs distributions: for a measure P on $\mathbb{R}^p$ and a measurable function $g : \mathbb{R}^p \to \mathbb{R}$, let $P[g]$ denote the expectation $\int g(h)P(dh)$ and, provided $P[g] < \infty$, let P_g denote the probability measure on $\mathbb{R}^p$, absolutely continuous with respect to P, with Radon-Nikodym derivative $\frac{dP_g}{dP}(h) = \frac{g(h)}{P[g]}$. A distribution of the form $P_{\exp(-\tau g)}$ is generally referred to as a Gibbs distribution with energy function g and inverse temperature τ. In the special case where P is a probability measure, we call $P_{\exp(-\tau \mathcal{L}^{\ell'}_{X,Y}(f))}$ a "Gibbs posterior". Then, one can state a result such as

Lemma 3.2.4. *Let $\tau > 0$ and let $\mathcal{L}^{\ell'}_{X,Y}(f)$ denote a surrogate risk function, taking values in an interval of length Δ and $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^n$. One sample from the Gibbs posterior $P_{\exp(-\tau \mathcal{L}^{\ell'}_{X,Y}(f))}$ is $\frac{2\tau\Delta}{n}$-differentially private.*

Unfortunately, sampling exactly from Gibbs posteriors is intractable. Furthermore, existing results that rely on sampling approximately from the posterior have conditions of convergence that are difficult to verify in the non-asymptotic case. In particular, Dziugaite and Roy (2018a) implements Stochastic Gradient Langevin Dynamics (SGLD) to approximate the posterior. A single update of SGLD is given for $k \in \mathbb{N}$ by

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \gamma(\sum_i \nabla \mathcal{L}^{\ell'}_{\boldsymbol{x}_i, y_i}(f(\boldsymbol{\theta}_k))) + \sqrt{2\gamma}\boldsymbol{\eta}_{k+1}$$

where $\boldsymbol{\eta}_{k+1} \sim \mathcal{N}(0, \mathbf{I})$ and when targeting a Gibbs posterior with inverse temperature τ one sets $\gamma = \frac{1}{\tau}$. Note that the distribution of $\boldsymbol{\theta}_{k+1}$ is given by $\mathrm{r}^{k+1}_{\mathrm{SGLD}}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_k - \gamma(\sum_i \nabla \mathcal{L}^{\ell'}_{\boldsymbol{x}_i, y_i}(f(\boldsymbol{\theta}_k))), 2\gamma\mathbf{I})$.

Recall also that the p-Wasserstein distance between distributions μ and ν is given by $\mathcal{W}_p(\mu, \nu) = (\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathbb{R}^p \times \mathbb{R}^p} ||x - y||_2^p \mathrm{d}\gamma(x, y))^{\frac{1}{p}}$ where $\Gamma(\mu, \nu)$ denotes the collection of all measures on $\mathbb{R}^p \times \mathbb{R}^p$. Then, the assumption that needs to be fulfilled in Dziugaite and Roy (2018a) so that the approximation error from SGLD remains small, is that for every $c > 0$, there exists a step size $\gamma > 0$ and number of SGLD iterations $k_* \in \mathcal{O}(\frac{1}{c^{c_1}})$,

such that the k_*-th iterate produced by SGLD satisfies

$$\begin{aligned}&\mathcal{W}_2(\mathrm{r}^{k_*}_{\mathrm{SGLD}}(\boldsymbol{\theta}), P_{\exp(-\tau\mathcal{L}^{\ell'}_{X,Y}(f))}) = \\ &\mathcal{W}_2(\mathcal{N}(\boldsymbol{\theta}_{k_*-1} - \gamma(\sum_i \nabla\mathcal{L}^{\ell'}_{\boldsymbol{x}_i,y_i}(f(\boldsymbol{\theta}_{k_*-1}))), 2\gamma\mathbf{I}), P_{\exp(-\tau\mathcal{L}^{\ell'}_{X,Y}(f))}) \leq c.\end{aligned} \tag{3.7}$$

The above corresponds to assuming that the distribution of $\boldsymbol{\theta}_{k_*}$ at the k_* iteration of SGLD is close to the Gibbs posterior in the 2-Wasserstein distance. This condition is difficult to verify in practice. Consequently, bounds need to be presented with the caveat that they are "optimistic" Dziugaite and Roy (2018b), which is unsatisfying. A number of other problems with this approach exist, such as computing the KL-divergence of two distributions, one of which is approximated by SGLD. One has to compute the KL divergence numerically by sampling using SGLD, this entails further approximations and looseness in the bound Dziugaite and Roy (2018a).

Analytical Moments Accountant

An alternative approach, when dealing with iterative optimization algorithms, is to analyze the privacy loss induced by each iteration. One first assumes that each iteration is an (ϵ, δ)-differentially private mechanism and then *composes* the privacy budgets of all the iterations. We already introduced a composition theorem 3.2.1 in Section 3.2, however this results in loose compositions. The main trick is to tightly compose the loss in privacy from each iteration. In Abadi et al. (2016) the authors start by noting that the definition of (ϵ, δ)-differential privacy is equivalent to a tail bound on a certain random variable. In particular

Lemma 3.2.5. *Let the random variable* $c(o; \mathscr{P}, Z, Z') \triangleq \log \frac{\mathbb{P}\{\mathscr{P}(Z)=o\}}{\mathbb{P}\{\mathscr{P}(Z')=o\}}$ *and the tail bound*

$$\mathbb{P}\{c(o; \mathscr{P}, Z, Z') \geq \epsilon\} \leq \delta \tag{3.8}$$

then the following holds

$$\mathbb{P}\{\mathscr{P}(Z) \in B\} \leq e^{\epsilon}\mathbb{P}\{\mathscr{P}(Z') \in B\} + \delta \tag{3.9}$$

where $\delta = \min_\lambda \exp(\max_{Z,Z'} a_{\mathscr{P}}(\lambda; Z, Z') - \lambda\epsilon)$ *and*

$$a_{\mathscr{P}}(\lambda; Z, Z') \triangleq \log \mathbb{E}_{o\sim\mathscr{P}(Z)}[\exp(\lambda c(o; \mathscr{P}, Z, Z'))]$$

is the $\log\ \lambda^{\text{th}}$*moment of the random variable.*

The authors then proceed by noting that composing tail bounds directly results in loose estimates. On the contrary, they propose to compose the *moments* of the random variables $c(o_i; \mathscr{P}, Z, Z')$ and then translate this into a tail bound. For this the following lemma is required

Lemma 3.2.6. *Suppose that a mechanism $\mathscr{P}$ consists of a sequence of mechanisms $\mathscr{P}_0, \ldots, \mathscr{P}_k$ where $\mathscr{P}_i : \prod_{j=0}^{i-1} \mathcal{W}_j \times (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{W}_i$. Then, for any λ*

$$a_{\mathscr{P}}(\lambda; Z, Z') = \sum_{i=0}^{k} a_{\mathscr{P}_i}(\lambda; Z, Z'). \tag{3.10}$$

Note that from (3.9) one now needs to estimate $\max_{Z,Z'} a_{\mathscr{P}}(\lambda; Z, Z')$. Crucially, the required estimates $\max_{Z,Z'} a_{\mathscr{P}_i}(\lambda; Z, Z')$ can be computed in a numerically stable and efficient way for some cases of differentially private mechanisms $\mathscr{P}_i$. To make the above more formal, we will need the following definition.

Definition 3.2.2. (Sampled Gaussian Mechanism(SGM)). Let f be a function mapping subsets of Z to $\mathbb{R}^p$. We define the Sampled Gaussian Mechanism (SGM) parameterized with the sampling rate $0 < q \leq 1$ and the noise $\sigma > 0$ as

$$\mathrm{SG}_{q,\sigma}(Z) \triangleq f(\{x : x \in Z \text{ is sampled with probability } q\}) + \mathcal{N}(0, \sigma^2 \mathbf{I}) \tag{3.11}$$

where each element of Z is sampled independently at random with probability q, and $\mathcal{N}(0, \sigma^2 \mathbf{I})$ is spherical p-dimensional Gaussian noise with per-coordinate variance σ^2.

Note that, based on our previous discussion, one would like to relate the Sampled Gaussian Mechanism $\mathscr{P}_i$ at i^{th} iteration to the to the privacy variable log moments $a_{\mathscr{P}_i}(\lambda; Z, Z')$, and in particular compute upper bounds on these moments. As such, the authors of Abadi et al. (2016) show the following

Lemma 3.2.7. *Let $\mathrm{SG}_{q,\sigma}$ be the Sampled Gaussian mechanism for some function f. Then*

$$a_{\mathrm{SG}_{q,\sigma}}(\lambda; Z, Z') \leq \log \mathbb{E}_{z \sim \mu_0}[(\mu(z)/\mu_0(z))^{\lambda+1}] \tag{3.12}$$

where $\mu_0 = \mathcal{N}(0, \sigma^2)$, $\mu_1 = \mathcal{N}(1, \sigma^2)$, $\mu \triangleq (1-q)\mu_0 + q\mu_1$ and assuming that $||f(Z) - f(Z')||_2 \leq 1$.

The terms $\mathbb{E}_{z \sim \mu_0}[(\mu(z)/\mu_0(z))^{\lambda}]$ can be computed efficiently, as μ_0, μ_1, μ are simple distributions. We defer details to the Appendix C.6. Putting everything together Abadi et al. (2016), provide the following algorithm 1.

Algorithm 1: Analytical Moments Accountant

input : Samples $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^n$, loss ℓ', classifier $f(\boldsymbol{\theta})$ parametrized by the vectorized weights $\boldsymbol{\theta}$. Hyper-parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound c, number of iterations T.

output : $\boldsymbol{\theta}_T$

Initialize $\boldsymbol{\theta}_0$ randomly;

for $t \in [T]$ **do**

  Take random sample L_t with sampling probability q.

  Compute gradient

  For each $i \in L_t$ compute $\boldsymbol{g}_t(\boldsymbol{x}_i) \leftarrow \nabla \mathcal{L}^{\ell'}_{\boldsymbol{x}_i, y_i}(f(\boldsymbol{\theta}_k)))$

  Clip Gradient

  $\overline{\boldsymbol{g}}_t(\boldsymbol{x}_i) \leftarrow \boldsymbol{g}_t(\boldsymbol{x}_i) / \max(1, \frac{||\boldsymbol{g}_t(\boldsymbol{x}_i)||_2}{c})$

  Add noise

  $\tilde{\boldsymbol{g}}_t(\boldsymbol{x}_i) \leftarrow \frac{1}{L}(\sum_i \overline{\boldsymbol{g}}_t(\boldsymbol{x}_i) + \mathcal{N}(0, \sigma^2 c^2 \mathbf{I}))$

  Descend

  $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_t \tilde{\boldsymbol{g}}_t(\boldsymbol{x}_i)$

end

The following facts are easy to check

Fact 3.2.8. *The step* $\tilde{\boldsymbol{g}}_t(\boldsymbol{x}_i) \leftarrow \frac{1}{L}(\sum_i \overline{\boldsymbol{g}}_t(\boldsymbol{x}_i) + \mathcal{N}(0, \sigma^2 c^2 \mathbf{I}))$ *with* $c = 1$ *and where* L_t *is sampled with probability* q*, is a sampled Gaussian mechanism* $\mathrm{SG}_{q,\sigma}$ *with* $||f(Z) - f(Z')||_2 \leq 1$.

Fact 3.2.9. *The step* $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_t - \eta_t \tilde{\boldsymbol{g}}_t(\boldsymbol{x}_i)$ *is a composition of private mechanisms* $\mathscr{P} = \mathscr{P}_{t-1} \circ \mathscr{P}_t = \mathrm{SG}^{t-1}_{q,\sigma}(X, Y) \circ \mathrm{SG}^{t}_{q,\sigma}(X, Y)$.

We note that the above algorithm is very similar to SGLD, with a few minor modifications in scaling factors. However, the results and assumptions of both approaches are quite different. In particular, in the Analytical Moments Accountant there are no assumptions on the convergence of the stochastic gradient descend. Furthermore, the privacy terms (ϵ, δ) are computed numerically. As obtaining lower privacy loss ϵ is directly related to increasing the noise variance σ in both cases, we can expect that tightly estimating ϵ will result in using a smaller variance σ and thus the iterates of the stochastic gradient descent will have better convergence properties. This is particularly important, as objectives where a high amount of noise is added to the gradients are difficult to optimize empirically.

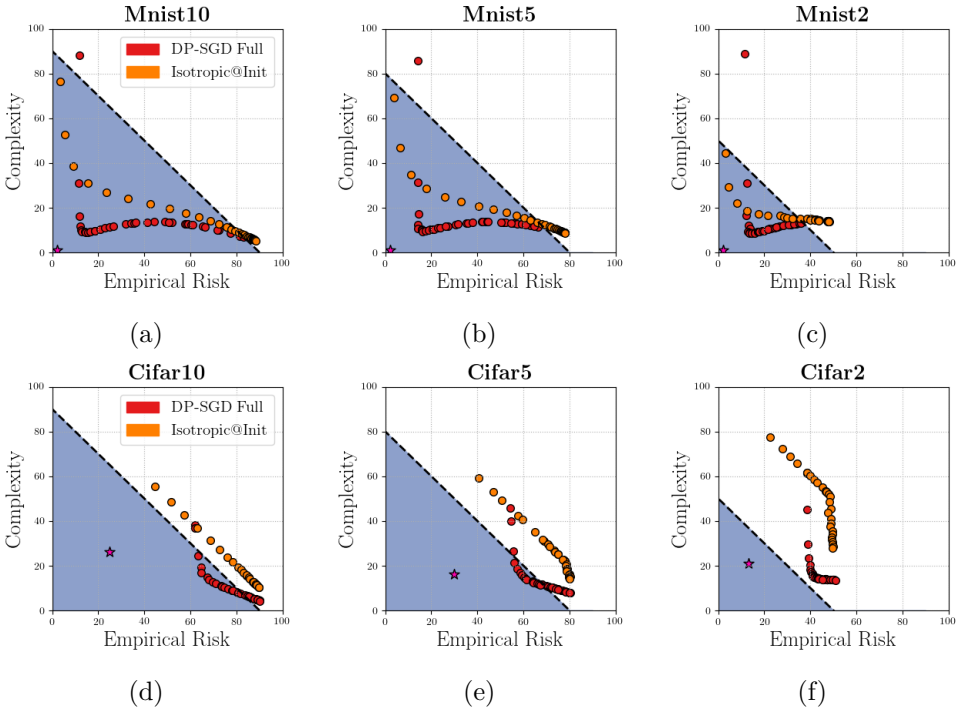


Figure 3.1 – **Detailed comparison of the baseline and the differential privacy bound**: The area below the dashed line corresponds to non-vacuous pairs of (complexity, empirical risk). The purple star corresponds to the optimal bound implied by the testing set. The isotropic baseline "Isotropic@Init" provides non-vacuous, but loose bounds for the MNIST datasets, but is vacuous in all cases of CIFAR. The differentially private approach "DP-SGD Full", results in significant improvements, and non-vacuous bounds even for the CIFAR datasets, apart from CIFAR2.

3.2.3 Experiments

We tested 6 different datasets. These consist of the original MNIST-10 and CIFAR-10 (Krizhevsky and Hinton, 2010) datasets, as well as simplified versions, where we collapsed the 10 classes into 5 and 2 aggregate classes, potentially simplifying the classification problem. All had 50000 training samples and 10000 validation samples. We test the architectures

$$\text{input} \to 300\text{FC} \to 300\text{FC} \to \#\mathit{classes}\text{FC} \to \text{output}$$

on MNIST, and

$$\text{input} \to 200\text{FC} \to 200\text{FC} \to \#\mathit{classes}\text{FC} \to \text{output}$$

on CIFAR, where xFC denotes a fully connected layer with x neurons. To evaluate bounding techniques, whenever the differences are large we use *Risk-Complexity* plots which where introduced in Pitas (2020).

On the $x-$axis we plot $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell_{01}}_{X,Y}(f)$ and on the $y-$axis we plot

$$\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell_{01}}_{X,Y}(f) - \Phi^{-1}_{\beta^*}(\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell_{01}}_{X,Y}(f) + \frac{1}{2\beta^* n\lambda}||\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\mathscr{P}}||_2^2 + \frac{\epsilon^2}{2\beta^*} + \mathrm{c})$$

where $\mathrm{c} = \frac{\ln((2\pi^2 b^2 \ln(c/\lambda^2))/(6\delta))}{\beta^* n} + \frac{\epsilon}{\beta^*}\sqrt{\frac{\ln((4\pi^2 b^2 \ln(c/\lambda^2))/(6\delta))}{2n}}$, for different choices of $\boldsymbol{\mu}_{\hat{\rho}}$, $\boldsymbol{\mu}_{\mathscr{P}}$, and λ. Note that β^* is the optimal value of $\beta > 0$ which is a free parameter and which we can evaluate using a grid search. Furthermore, in some cases we will need to approximate $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell_{01}}_{X,Y}(f)$ using m samples and the constant c becomes $\tilde{c} = c + \sqrt{\frac{\ln\frac{2}{\delta'}}{m}}$.

In the Risk complexity plots, we can also plot the area that corresponds to non-vacuous complexity-empirical risk pairs. We shade the corresponding area with blue. At the same time, for any specific classification problem, we can optimize a deterministic neural network and using a validation set, estimate an "optimal" generalization error bound. We plot the corresponding estimate using a purple star. The more Risk-Complexity estimates are closer to this point the tighter the bound is.

As a baseline, we will use the posterior and prior choices $\hat{\rho} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda\mathbf{I})$ and $\pi = \mathcal{N}(\boldsymbol{\mu}_{\text{init}}, \lambda\mathbf{I})$, note that in this case $\epsilon = 0$ (as the prior is uninformative) and we do not need to compute it using a differentially private algorithm. We saw in Chapter 2 that this simple choice has been shown to provide very tight results for simple datasets and deep neural network architectures.

For all experiments we used the TensorFlow-Privacy package Andrew et al. (2019).

Posterior equal to Prior

We first test the case where the posterior mean is set to be equal to the differentially private mean, $\boldsymbol{\mu}_{\hat{\rho}} = \boldsymbol{\mu}_{\mathscr{P}}$. Given a training set $(X, Y) \in (\mathcal{X}, \mathcal{Y})^n$, we obtain a differentially private prior mean as $\boldsymbol{\mu}_{\mathscr{P}} = \arg\min_{\boldsymbol{\theta}} \hat{\mathcal{L}}^{\text{cat}}_{X,Y}(f(\boldsymbol{\theta}))$, where we optimize using the Analytical Moments Accountant Algorithm 1. For different values of group size L, gradient noise scale σ, gradient norm bound c, and number of iterations T we can then compute the privacy loss ϵ using equations (3.10) and (3.12) while setting δ to a small value. We then set $\boldsymbol{\mu}_{\hat{\rho}} = \boldsymbol{\mu}_{\mathscr{P}}$ and the KL-divergence is trivially equal to zero for any choice of $\lambda > 0$, and then evaluate a bound using (3.6). In particular, the variance λ cannot be equal to 0 as then the Kl-divergence is undefined. However, we consider this point a technicality as we can set λ to be very small and non-zero and both the classification accuracy of a deep neural network remains unaffected and KL=0. We thus simply set $\lambda = 0$, and ignore this technicality.

For all the MNIST cases the hyperparameters of the Analytical Moments Accountant are: group size $L = 250$, gradient norm $c = 1$, and number of iterations $T = 4 * 200$. The noise scale σ controls the amount of differential privacy (ϵ, δ). In the particular

implementation we used, we first set a target $\delta = 1e-5$ and then we choose $\sigma \in [1, 20]$. For all values σ we then compute numerically the corresponding values for ϵ.

For all the CIFAR cases the hyperparameters of the Analytical Moments Accountant are identical to the ones for MNIST apart from the number of iterations T which we increase to $T = 10 * 200$ to compensate for the increased difficulty of the dataset.

For the baseline isotropic posteriors and priors we used $\lambda \in [0.031, 0.3]$ in all cases. We plot the results in Figure 3.1. We see that in all cases the differentially private bound is significantly tighter than the baseline. For the MNIST case the approach manages to tighten substantially the already non-vacuous bounds. In the more difficult CIFAR case the baseline approach fails and results only in vacuous estimates. By contrast, the approach based on differential privacy manages to obtain non-vacuous, albeit loose, bounds. The only case where the bound remains vacuous is in the CIFAR-2 case.

Optimized Posterior

In this section we investigate whether further optimizing the posterior mean results in bound improvements. On a high level, enforcing privacy constraints results in a prior mean that has sub-optimal empirical risk. By optimizing the posterior mean we can lower the empirical risk at a cost of a non-zero KL-divergence between the posterior and prior distribution. Whether this trade-off is beneficial for our bounds depends on the geometry of the optimization landscape and has to be evaluated empirically. In practice, after obtaining a differentially private prior mean as in section 3.2.3 we find a posterior mean as $\boldsymbol{\mu}_{\hat{\rho}}^* = \arg\min_{\boldsymbol{\theta}} \hat{\mathcal{L}}_{X,Y}^{\ell_{\text{cat}}}(f(\boldsymbol{\theta})) + \beta_1 ||\boldsymbol{\theta} - \boldsymbol{\mu}_{\mathscr{P}}||_2^2$. Note that although this objective is conceptually similar to (3.5) it is not necessarily stochastic and as such we can optimize it using vanilla stochastic gradient descent. We can the evaluate bound (3.4) using

$$\hat{\rho} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}^*, \lambda \mathbf{I}), \quad \pi = \mathcal{N}(\boldsymbol{\mu}_{\mathscr{P}}, \lambda \mathbf{I})$$

, for different values of λ and the ϵ privacy loss required to obtain $\boldsymbol{\mu}_{\mathscr{P}}$. Note that in this case the value of λ contributes non-trivially to the complexity of the bound. This, also implies that the term $\mathbf{E}_{f\sim\hat{\rho}} \hat{\mathcal{L}}_{X,Y}^{\ell_{01}}(f)$ needs to be bounded using a Chernoff bound with m samples.

For each case,, we fine tune the regularization free parameter β_1. In particular, for different values of β_1 we compute a grid over $\lambda \in [0.031, 0.3]$ and compute the best Empirical Risk - Complexity pair λ^*. This corresponds to the pair in the Risk-Complexity plot that is closest to the origin. Each β_1 choice corresponds to an optimal pair λ^*, and we furthermore choose the best among this optimal pairs β_1^* for a final hyperparameter choice of (λ^*, β_1^*). We find $\boldsymbol{\mu}_{\hat{\rho}}^*$, in each case, using SGD with $E = 10$ epochs and $\eta = 0.15$ stepsize. We also use $m = 1000$ samples to compute the Chernoff bound. We defer the complete Risk-Complexity plots to the Appendix C.7 as the bound improvements are

	10 Class	5 Class	2 Class
MNIST	20% (23%)	22% (25%)	16% (23%)
CIFAR	86% (81%)	73% (77%)	62% (59%)

Table 3.1 – **Effect of optimizing the posterior**: We present the bound values when optimizing the posterior as resulting from Section 3.2.3. In parentheses we include the best bound values when the posterior distribution is identical to the prior as in Section 3.2.3. The results are mixed. For MNIST there is some improvement, with particularly significant improvement for the 2-class case. In the case of CIFAR the bound actually loosens, apart from the case of CIFAR-5.

minimal. We summarize the results in table 3.1. We see that for the MNIST cases one is able to improve upon the bounds by $\sim 3\% - 7\%$ percentage points. At the same time, experiments on CIFAR result in no improvements. What is particularly notable is that not only are the improvements small, but also for a wide range of β_1 values the bounds actually become more loose.

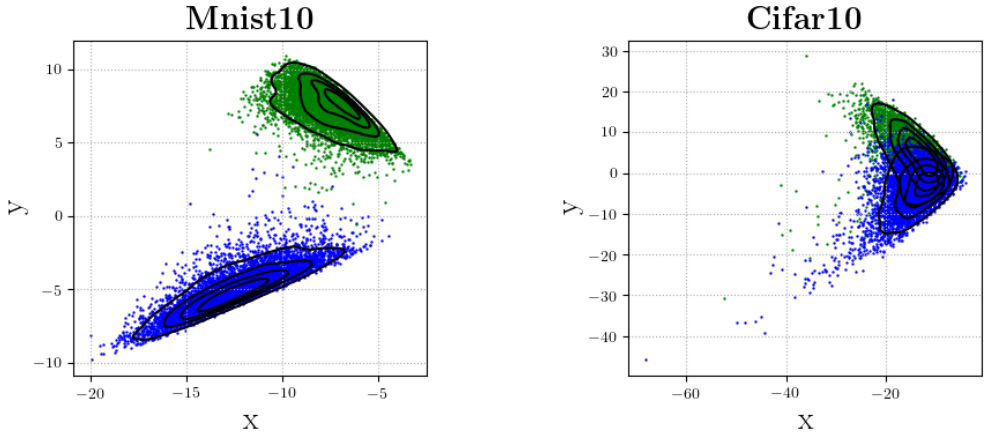
3.3 Lower bound for stable mechanisms

Given the significant improvements obtained by the above approach, it is natural to ask how close one can get to the true generalization error, implied by the validation set. We remind the reader that finding a good bound entails balancing the empirical risk and the estimated complexity, and that the dominant factor of complexity in the previous analysis is the privacy loss ϵ. There is a long line of literature relating privacy constraints to lower bounds on the empirical risk that one can obtain from a given model (De, 2012; Jain and Thakurta, 2014; Chaudhuri and Hsu, 2011; Bassily et al., 2014; Kasiviswanathan et al., 2011). We use ideas from Hardt and Talwar (2010) to derive a simple lower bound for learning the *final* linear classification layer of a deep neural network in a private way. The benefit of this analysis is that it holds for *any* private mechanism, given that we use a slightly relaxed notion of privacy.

We first define a simple generative model and relate it to the input data of the final deep neural network layer. We model the class-conditional densities $p(\boldsymbol{x}|\mathcal{C}_k)$, as well as the class priors $p(\mathcal{C}_k)$, and then use these to compute posterior probabilities $p(\mathcal{C}_k|\boldsymbol{x})$ through Bayes' theorem. For a K>2 class classification problem, Bayes' theorem gives

$$\begin{aligned} p(\mathcal{C}_k|\boldsymbol{x}) &= \frac{p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_j p(\boldsymbol{x}|\mathcal{C}_j)p(\mathcal{C}_j)} \\ &= \frac{\exp(a_k)}{\sum_j \exp(a_j)} \end{aligned} \tag{3.13}$$

which is known as the normalized exponential or *softmax* function. Here we defined a_k



(a) (b)

Figure 3.2 – **Visualizing the latent representations**: We use pretrained deterministic neural networks to visualize the latent representations at the input of the final softmax layer. We compute the singular value decomposition of two classes and project the datapoints along the first two principal directions. We also plot the contours of a Gaussian kernel density estimator that we fit on the data. The samples concentrate closely around a mean, validating partially our modeling choice to approximate the distribution of latent representations as a mixture of Gaussians.

as

$$a_k = \ln(p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)). \tag{3.14}$$

One can then define a classifier as $f(\boldsymbol{x}) = \arg\max_k p(\mathcal{C}_k|\boldsymbol{x})$.

3.3.1 Gaussian Mixture Model

Let us assume that the class-conditional densities are Gaussian and then explore the resulting posterior probabilities. We shall assume that all classes share the same covariance matrix. Thus, the density for class $\mathcal{C}_k$ is given by

$$p(\boldsymbol{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right\} \tag{3.15}$$

For this simple case it is well known that the posterior probabilities for classes $\mathcal{C}_k$ can be found in closed form (Bishop, 2006).

Lemma 3.3.1. *Assume a K-class classification problem where $p(\boldsymbol{x}|\mathcal{C}_k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, then by applying Bayes' theorem, the maximum likelihood solution to the generative classifier is*

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{\exp(a_k(\boldsymbol{x}))}{\sum_j \exp(a_j(\boldsymbol{x}))}, \quad a_k(\boldsymbol{x}) = \mathbf{w}_k^\top \boldsymbol{x} + w_{k0} \tag{3.16}$$

	10 Class	5 Class	2 Class
MNIST	5% (4%)	7% (5%)	24% (6%)
CIFAR	34% (23%)	31% (30%)	15% (14%)

Table 3.2 – **Comparison of generative and discriminative classifier**: We use the latent representations of the penultimate layer of pretrained deterministic neural networks to train two separate final layers. The first, is a generative classifier where we model the class distributions as multivariate Gaussians and it's solution is obtained in closed form. The second (in parentheses), is a simple discriminative classifier where we optimize the final deep neural network softmax layer on the latent representations of the pretrained model, using stochastic gradient descent. We see that the empirical risk of the generative classifier in all cases is close to the one of the discriminative classifier.

where we have defined

$$\mathbf{w}_k = \tilde{\mathbf{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}}_k, \;\; w_{k0} = -\frac{1}{2}\tilde{\boldsymbol{\mu}}_k^\top \tilde{\mathbf{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}}_k + \ln \tilde{p}(\mathcal{C}_k) \tag{3.17}$$

and also we have $\tilde{p}(\mathcal{C}_k) = \frac{n_k}{\sum_i n_i}$, $\tilde{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i\in\mathcal{C}_1} \boldsymbol{x}_i$, $\tilde{\mathbf{\Sigma}}_k = \frac{1}{n_k}\sum_{i\in\mathcal{C}_k}(\boldsymbol{x}_i - \tilde{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \tilde{\boldsymbol{\mu}}_k)^\top$ *and* $\tilde{\mathbf{\Sigma}} = \sum_i \tilde{p}(\mathcal{C}_i)\tilde{\mathbf{\Sigma}}_i$.

As noted in a number of works (Soudry et al., 2018), a discriminative version of the above objective is optimized by the final layer of deep neural networks. Furthermore, the latent representations of a deep neural network are known to become progressively more linearly separable (Jacobsen et al., 2018), and one would hope that their distributions become approximately Gaussian in the penultimate layer.

In Section 3.2.3 we used deterministic deep neural networks to compute an empirical risk, a validation risk, and the implied true generalization error. We use the latent representations at the input of the final layers of these networks to test the assumptions of the generative model. Thus, we first visualize in Figure 3.2 these latent representations. In particular, we remind that the neural networks we consider transform inputs $\mathbf{a}_0 = \boldsymbol{x}$ to an output $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{a}_l$ through a series of l layers, each of which consists of a bank of units/neurons. The computation performed by each layer $i \in \{1, \cdots, l\}$ is given as follows

$$\begin{aligned} \boldsymbol{s}_i &= \mathbf{W}_i \mathbf{a}_{i-1}, \\ \mathbf{a}_i &= \phi_i(\boldsymbol{s}_i). \end{aligned} \tag{3.18}$$

Thus, given a dataset $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^n$ we visualize the latent representations at the input of the final layer for two classes $\mathbf{a}_2(\boldsymbol{x}) = \phi_2(\mathbf{W}_2\phi_1(\mathbf{W}_1\boldsymbol{x}))$, $\boldsymbol{x} \in X, y \in \{0, 1\}$ for all architectures. Given that the representations are high dimensional, we visualize them using PCA and project them along the 2 first principal components. We see that the Gaussian modeling is plausible. In particular the latent representations appear

concentrated around a mean, at least under this projection.

Furthermore, we calculate the accuracy of the generative classifier and we find that it performs reasonably well in the classification task, on par with the discriminative approach of the last neural network layer, despite the restriction on having a shared covariance across classes. In particular, given a dataset $(X, Y) \in (\mathcal{X} \times \mathcal{Y})^n$ and pretrained non-linear layer weights $\mathbf{W}_2^*,\ \mathbf{W}_1^*$ we use $\mathbf{a}_2(\boldsymbol{x}) = \phi_2(\mathbf{W}_2^*\phi_1(\mathbf{W}_1^*\boldsymbol{x})),\ \boldsymbol{x} \in X$ to obtain the parameters of the generative classifier $\mathbf{w}_k,\ w_{k0},\ k \in \{0, \cdots, K\}$, according to equations (3.17). We present these results in Table 3.2. In most cases the empirical risk is almost identical to optimizing the final layer directly using SGD, while in the cases where there is an increase in empirical risk it remains non-vacuous.

3.3.2 Lower bound

Since the maximum likelihood solution presented above depends on the empirical estimates for the mean and covariance of Gaussians, as the number of training samples increases the estimate will converge to the true quantity. Using an simplified analysis we illustrate how introducing stability notions such as differential privacy to the above procedure alters significantly the resulting intuition. We first introduce a different definition of privacy, ϵ-generalized privacy

Definition 3.3.1. A randomized algorithm $\mathscr{P} : \mathcal{Z}^n \to T$ is ϵ-generalized private if, for a dataset $Z \in \mathcal{Z}^n$ and vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ such that $||\boldsymbol{x} - \boldsymbol{y}||_1 \leq k$, and all measurable subsets $B \subseteq T$, we have $\mathbb{P}\{\mathscr{P}(Z\text{Diag}(\boldsymbol{x})) \in B\} \leq e^{\epsilon k}\mathbb{P}\{\mathscr{P}(Z\text{Diag}(\boldsymbol{y})) \in B\}$.

We detail how this notion of privacy can be seen as a relaxation of ϵ-differential privacy in Appendix C.3.

We present now a lower bound on the expected ℓ_2 distance between the maximum likelihood estimate of the hyperplane $\mathbf{w}_k$ for any class k, and the estimate of the same hyperplane for *any* ϵ-generalized private mechanism.

Lemma 3.3.2. *Let* $\mathbf{X}_k \in \mathbb{R}^{d \times n_k}$, $\mathbf{X}_k = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_k}\} \sim p(\boldsymbol{x}|\mathcal{C}_k)^{n_k}$ *following the Gaussian mixture model with known fixed covariance* $\boldsymbol{\Sigma} = \mathbf{I}$. *Then for* ϵ*-generalized private mechanisms* $\mathscr{P}$ *with* $(d/2n_k) \geq \epsilon > 0$, *when* $d - (\ln(0.05) + d(1 - \mathcal{O}(1)))/d \leq n_k$ *we have*

$$\sup_{\mathscr{P}} \mathbb{E}[||\mathscr{P}(\mathbf{X}_k) - \mathbf{w}_k||_2] \geq \mathcal{O}(\epsilon^{-1} d\sqrt{\ln(2n_k/d)}) \tag{3.19}$$

with probability ≥ 0.95 *over the random draws of* $\mathbf{X}_k$, *and the expectation is over the randomness of* $\mathscr{P}$.

In particular, we see that, given a large enough number of samples n_k, with high probability, the expected error for *any* generalized private mechanism $\mathscr{P}$ is proportional

to the dimensionality of the dataset d and *inversely* proportional to the differential privacy ϵ. We note here that the above holds up to a constant which hides dependencies on the mean $\boldsymbol{\mu}_k$ and the covariance $\boldsymbol{\Sigma}$ of the generating distribution. At the same time, we would expect that for real data the above would hold for the *intrinsic* dimensionality and not the ambient dimensionality which is often much higher. Furthermore, we have presented a result in terms of the ℓ_2 distance and not the empirical risk. As such, the result is more qualitative than quantitative, and furthermore it is an empirical question as to whether it provides any useful intuition.

We provide a proof outline in Appendix C.3 and the complete proof in Appendix C.4. We also clarify here, the role of $\sup_{\mathscr{P}}$ in Lemma 3.3.2. In particular the exact formulation of our proof is that for any $\mathscr{P}$ estimating $\mathbf{w}_k$ belongs to a set of similar cases of which one should have error at least $\mathcal{O}(\epsilon^{-1}d\sqrt{\ln(2n_k/d)})$. We then *assume* that the worst case and $\mathbf{w}_k$ coincide for some $\mathscr{P}$. Formally proving this assumption might be interesting in itself.

3.3.3 Experiments

We first test whether the above analysis is validated by synthetic data. For this, we construct a binary classification problem where $p(\boldsymbol{x}|\mathcal{C}_0) = \mathcal{N}(0, \mathbf{I})$ and $p(\boldsymbol{x}|\mathcal{C}_1) = \mathcal{N}(4 * \mathbf{1}, \mathbf{I})$. We then sample $\mathbf{X}_k = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{n_k}\} \sim p(\boldsymbol{x}|\mathcal{C}_k)^{n_k}$ where $n_k = 1000,\ k \in \{0, 1\}$, set $X = [\mathbf{X}_0; \mathbf{X}_1]$ and Y accordingly and optimize

$$\min_{\boldsymbol{\theta}} \mathcal{L}^{\text{cat}}_{X,Y}(f(\boldsymbol{x}; \boldsymbol{\theta})),\ f(\boldsymbol{x}; \boldsymbol{\theta}) = \text{softmax}(\mathbf{W}\boldsymbol{x})$$

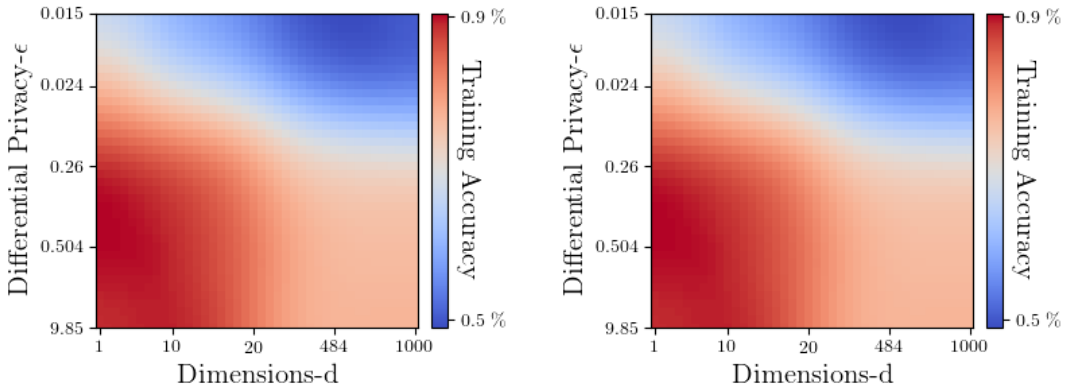
using the Analytical Moments Accountant algorithm 1. In particular, we use the hyperparameters: group size $L = 250$, gradient norm $c = 1$, and number of iterations $T = 10 * 8$. We plot the results in Figure 3.3. We see that the results agree qualitatively with our theoretical analysis. For high values of d and small values of ϵ the accuracy of the learned classifier becomes close to random, both in the training and testing set.

In particular, for real datasets this would imply that for more complex data where the latent representations in the penultimate layer of a deep neural network would have higher intrinsic dimensionality, the differentially private mechanism would have high empirical risk even for high values of ϵ (small privacy).

Qualitative evaluation: We test the qualitative implications of the above for the deep neural networks of section 3.2.3 by using the latent representations of the pretrained non-private classifiers $\mathbf{a}_2(\boldsymbol{x}) = \phi_2(\mathbf{W}_2^*\phi_1(\mathbf{W}_1^*\boldsymbol{x}))$, $\boldsymbol{x} \in X$. We train and compute a bound on only *the final* deep neural network layer by minimizing

$$\min_{\mathbf{W}} \mathcal{L}^{\text{cat}}_{X,Y}(f(\mathbf{a}_2(\boldsymbol{x}); \mathbf{W})),\ f(\mathbf{a}_2(\boldsymbol{x}); \mathbf{W}) = \text{softmax}(\mathbf{W}\mathbf{a}_2(\boldsymbol{x}))$$

using the Analytical Moments Accountant. Using a procedure identical to the one



(a) (b)

Figure 3.3 – **Training and testing accuracy for synthetic data**: We generate a synthetic two class problem according to $p(\boldsymbol{x}|\mathcal{C}_0) = \mathcal{N}(0, \mathbf{I})$ and $p(\boldsymbol{x}|\mathcal{C}_1) = \mathcal{N}(c * \mathbf{1}, \mathbf{I})$ and use it to train a differentially private softmax classifier f defined by $f(\boldsymbol{x}; \boldsymbol{\theta}) = \text{softmax}(\mathbf{W}\boldsymbol{x})$. We achieve differential privacy using the Analytical Moments Accountant, and compute the training and testing accuracy for different values of differential privacy ϵ and ambient dimensionality d. We observe that the accuracy exhibits a phase transition beyond which it drops rapidly both when we increase the dimensionality d and decrease the privacy loss ϵ.

in section 3.2.3, one can then use the computed parameter ϵ and the empirical risk $\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell_{01}}_{X,Y}(f)$ for only this final layer to obtain generalization bounds. Note that these bounds are invalid (we have implicitly learned the first and second layers in a non-private way), however they are illustrative of how close we could possibly get to the true generalization error implied by the validation set. One can argue that learning the final softmax layer privately should be at least as hard as learning the entire network.

We plot the results alongside the bounds for the differentially private full networks in Figure 3.4. We see that for the CIFAR dataset even if we only optimize the final softmax layer in a differentially private way we remain quite far from the true generalization error implied by the validation set. Furthermore, the more complex the dataset, from CIFAR-2 to CIFAR-10, the greater the gap is between the differentially private final layer and the generalization error implied by the validation set. For all MNIST datasets the differentially private final layer bound is much closer to the true generalization error, however again the more complex the dataset the greater the distance with the true generalization error.

Quantitative evaluation: We will now try to use our Gaussian Mixture Model to

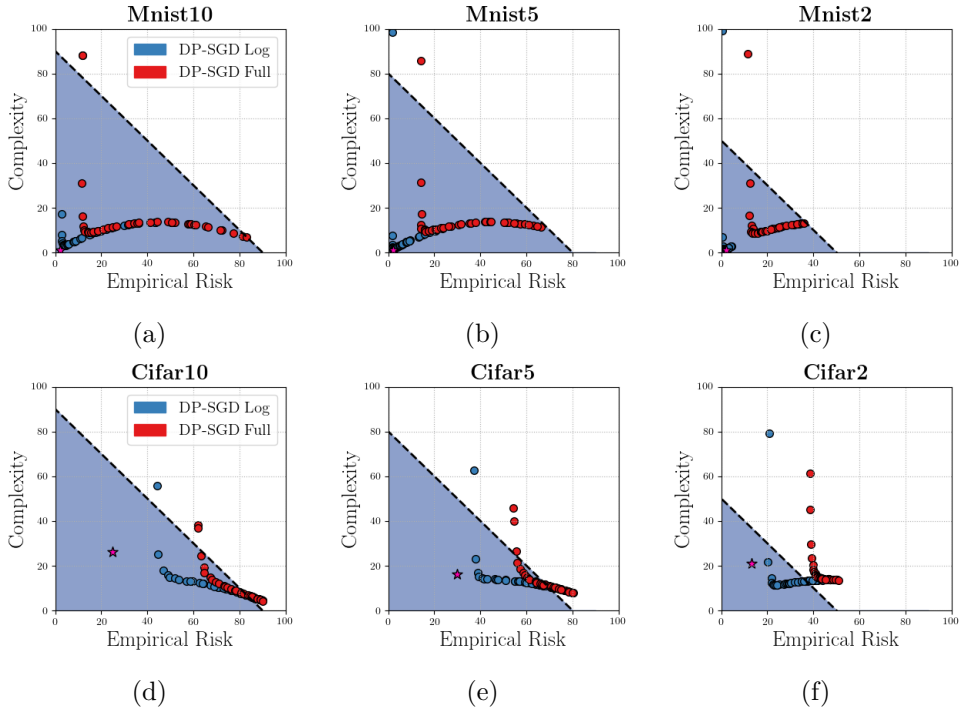


Figure 3.4 – **Optimizing only the final layer**: We compute the accuracy and complexity of only the final softmax deep neural network layer "DP-SGD Log", assuming that the privacy cost of previous layers is negligible. We compare with the complexity and accuracy that we can estimate for the full network "DP-SGD Full". We reason that this serves as an optimistic but valid benchmark on how good of a bound we can obtain in theory. We see that there is indeed a phase transition as below a certain classifier complexity the empirical risk increases rapidly. Note that for CIFAR there is a gap between what we can obtain even with this optimistic approach and the complexity implied by the validation set. For MNIST there is virtually no such gap. Note also that the non-linear distortions are due to the Catoni bound.

make predictions about the *deep neural network*. Specifically we will try to estimate the ϵ for which the empirical risk of the deep neural network will increase significantly. For the deep neural network experiments of Figure 3.4 we assume that the point where significant degradation starts is the bound estimate closest to the origin, as further reducing the complexity/privacy loss, will only result in a big reduction in classification accuracy. For example for CIFAR-10 (Figure 3.4d), this would be the point Complexity $\approx 14\%$ and Empirical Risk $\approx 45\%$.

Our theoretical analysis depends crucially on the dimensionality d of the data at the input of the final layer. We compute the dimensionality in the following way. We first compute the singular value decomposition of the latent representations of the training set in each dataset case, from the pretrained model. We can then train a linear classifier, using the training representations $\mathrm{svd}_c(\mathbf{a}_2(\boldsymbol{x})) = \mathrm{svd}_c(\phi_2(\mathbf{W}_2^*\phi_1(\mathbf{W}_1^*\boldsymbol{x}))),\ \boldsymbol{x} \in X$, where

	Dimensions d	Pred Complexity ($\epsilon/2$)	**Bound Complexity**
MNIST-2	2	0.99%	**0.7%**
MNIST-5	12	1.49%	**1.57%**
MNIST-10	13	1.49%	**3.39%**
CIFAR-2	80	3.25%	**11.36%**
CIFAR-5	100	3.25%	**15.2%**
CIFAR-10	170	4.49%	**14.84%**

Table 3.3 – **Complexity predicted by the synthetic example and the Analytical Moments Accountant**: For the dimensionalities d of each dataset we calculate the parameter ϵ for which the phase transition occurs between good classification accuracy and trivial random predictions using our synthetic example Figure 3.3. Then we estimate using Figure 3.4 the minimal complexity estimated by the Analytical Moments Accountant, beyond which the empirical risk increases sharply. Although the estimates diverge they are roughly within the same order of magnitude and follow the same ordering proving some predictive power.

X is the training set and $\text{svd}_c(\cdot)$ denotes a projection on the c most important singular directions (we re-project back to the original space). We can use the same projection on the validation set and we can then estimate d as the minimal value of c, such that the *validation* accuracy of the linear classifier doesn't degrade.

We can then estimate the privacy ϵ, for which the training accuracy of the private classifier will degrade significantly in the following way. We go back to the synthetic dataset and Figure 3.3 and set this ϵ to the value where the classifier has dropped to 75% accuracy, with 75% being the midpoint between the non-private 100% accuracy and the trivial 50% accuracy.

We present the results in Table 3.3. The synthetic example that we used to make our predictions has a number of shortcomings: i) we have not finetuned the covariance to match the scaling of the real data and the synthetic problem is a *binary* classification problem, while a number of the real datasets are multiclass classification problems. Given the above, the predicted values are close to the empirical values, and our analysis therefore captures approximately the correct dependencies on the intrinsic dimensionality d and the privacy parameter ϵ.

3.4 Conclusion and Summary

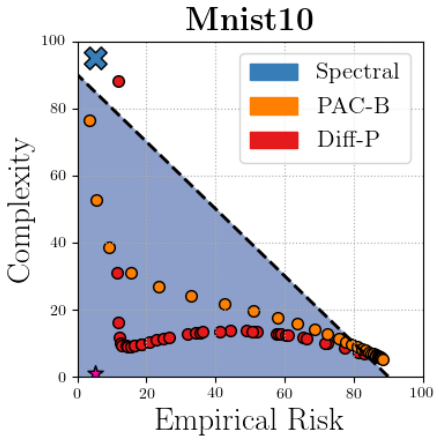
In this chapter we have looked at PAC-Bayesian bounds in the case where the prior and posterior distributions are modeled as Gaussians and where the prior mean is learned in a differentially private way. By applying state of the art differentially private mechanisms, we demonstrated that it is possible to obtain non-vacuous bounds which are significantly tighter than the previous approaches in Chapters 1 and 2. In particular, setting the

posterior to be equal to the prior resulted in significant improvements, while further optimizing the posterior mean resulted in some small gains at the cost spending resources on hyperparameter tuning. At the same time, we presented a simple theoretical analysis which can be interpreted as a lower bound to how much empirical risk one can achieve for a given privacy budget, and related this to the deep learning setting. Our results suggest that properly accounting for the amount of information added to the learned weights by each gradient step can be a promising direction in deriving non-vacuous generalization bounds for more complex architectures and datasets.

Summary and Discussion

Summary

In Chapter 1 we started off with a discussion of spectral complexity. We easily constructed datasets with an increasing number of translations and elastic deformations, such that spectral complexity doesn't correlate empirically with generalization error. As such we advocate that the invariance properties of deep neural networks should be looked at in more detail when constructing generalization bounds. At the same time claims that spectral complexity correlates empirically with generalization error have to be stated with more precision. In the second part of the chapter we compared spectral complexity estimates for convolutional neural networks and locally connected networks and found that they are identical up to log factors that are artifacts of the derivations. We argued



(a)

Figure 3.5 – **Risk-Complexity plot for MNIST 10**: The area below the dashed line corresponds to non-vacuous pairs of (complexity, empirical risk). The purple star corresponds to the optimal bound implied by the validation set. "Spectral" corresponds to the complexity measurement using spectral complexity from Chapter 1. "PAC-B" corresponds to optimal bound combinations that we derive in Chapter 2 using the PAC-Bayesian approach. "Diff-P" corresponds to optimal bound combinations that we obtain when we learn an informative prior mean in Chapter 3.

how this prescribes caution when using vacuous bounds to compare different architectures.

In Chapter 2 we looked at non-vacuous PAC-Bayesian bounds for randomized neural networks. These take the form

$$\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell'}_{\mathcal{D}}(f) \leq \Phi^{-1}_{\beta}(\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f) + \frac{1}{\beta n}(\mathrm{KL}(\hat{\rho}||\pi) + \ln\frac{1}{\delta})), \tag{3.20}$$

where $\Phi^{-1}_{\beta}(x) = \frac{1-e^{-\beta x}}{1-e^{-\beta}}$ and the posterior and priors are Gaussian such that $\hat{\rho} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\Sigma}_{\hat{\rho}})$ and $\pi = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \boldsymbol{\Sigma}_{\pi})$. We found that a baseline approach where the prior mean was chosen to be the random deep neural network initialization, coupled with isotropic covariances with a shared scaling factor λ in both prior and posterior

$$\hat{\rho} = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \lambda\mathbf{I}),$$

$$\pi = \mathcal{N}(\boldsymbol{\mu}_{\text{init}}, \lambda\mathbf{I}),$$

resulted in many cases in non-vacuous estimates. We also saw how relaxing slightly from the baseline, such that the posterior and prior covariances $\boldsymbol{\Sigma}_{\hat{\rho}}, \boldsymbol{\Sigma}_{\pi}$ would be diagonal (known as the mean-field approximation) and then optimizing using variational inference, resulted in negligible improvements. We then presented some experiments were the covariance matrices where relaxed to being block-diagonal, which resulted in a significant tightening of the bounds.

In Chapter 3 noting the importance of choosing the prior mean $\boldsymbol{\mu}_{\pi}$ in PAC-Bayes bounds with Gaussian priors and posteriors, we reviewed existing results from Dziugaite and Roy (2018a) in learning informative prior means from the training set by enforcing differential privacy constraints. We detailed how moving from SGLD (Welling and Teh, 2011) as a mechanism to enforce privacy, to a state of the art approach known as the Analytical Moments Accountant (Abadi et al., 2016), resulted in alleviating a number of theoretical concerns when using the former. We then noted how the posterior could be set to be identical to the prior, circumventing the need to optimize further the posterior. This simplified approach, where we set the posterior equal to the prior, resulted in even tighter bounds than Chapter 2. Through experiments we showed how optimizing further the posterior resulted in limited improvements to the bound. We then presented a theoretical analysis and related it to limits to the tightness of generalization bounds that we can achieve using differential privacy. In particular the intrinsic dimensionality of the dataset in the penultimate layer of a deep neural network was found to contribute linearly to the ℓ_2 error between the maximum likelihood solution of the final layer and *any* solution of a suitably private algorithm. The privacy loss ϵ can be readily linked to an estimate of the complexity of the learned classifier. As such any private algorithm with a low enough ϵ (low model complexity) should also have a high empirical risk.

In Figure 3.5 we plot again the main figure of the thesis.

Compared to proving generalization using a validation set we derived a number of intuitions regarding *why* deep neural networks generalize well. In particular in Chapter 2 we saw that flatness of a minimum is related to good generalization, adding to previous observations by Keskar et al. (2016); Neyshabur et al. (2015, 2017a). Researchers have looked at ways to relate flatness of minima to the noise introduced by stochastic gradient descend (Kleinberg et al., 2018), thus coming closer to painting a complete picture of the solution to the generalization puzzle. At the same time we contribute the refined intuition that the curvature at the minimum is quite complex and that our generalization bounds should reflect this complexity with suitably expressive posteriors. Finally Chapter 3 implies that optimizing a deep neural network with stochastic gradient descent, severely limits the amount of information that the deep neural network weights obtain from the training set. Our results simply add to an already large literature relating algorithmic stability (Kuzborskij and Lampert, 2017; Bousquet and Elisseeff, 2002; Hardt et al., 2016; Chaudhari et al., 2019) and generalization, however we differ in providing non-vacuous bounds.

Discussion

A number of interesting open problems exist.

In Chapter 1 we argued that incorporating invariances into generalization bounds might tighten them and remove a number of inconsistencies. While there have been a few works in this direction (Lyle et al., 2020; Sokolic et al., 2016; Achille and Soatto, 2018), they have dealt mainly with greatly simplified cases, such as when a classifier learns invariances using data augmentation (Lyle et al., 2020). Incorporating architectural invariances, and reasoning about more complex invariances is still largely unexplored. For example, well trained deep neural networks are invariant to complex transformations of images, such as changes to color and texture.

In Chapter 2 we saw that using more expressive Gaussian posteriors with block diagonal covariances, resulted in significantly tighter bounds. One could look at richer approximations of the true covariance, such as the complete K-FAC. The literature on the subject is growing (Mishkin et al., 2018; Lin et al., 2019b,a; Ritter et al., 2018; Zhang et al., 2018; Bae et al., 2018; Sun et al., 2019; Louizos et al., 2019), together with new optimization libraries (Dangel et al., 2019, 2020), however problems persist and existing approaches are limited to simple data distributions and small architectures. Sampling from such Gaussians efficiently and optimizing the corresponding variational optimization problem is a promising area of future research.

In Chapter 3 adding Gaussian noise to gradients resulted limiting the amount of information that training samples conferred to the learned deep neural network weights, resulting in a network that had provably *not* memorized the training data and therefore generalized

well. However, one might argue that not all samples affect all weights equally. Taking into account any latent structure in the gradient updates might allow us to decrease ϵ (increase privacy) while retaining low empirical risk, thus further tightening our bounds.

A Appendix

We denote vectors with bold lowercase letters and matrices with bold capital letters. Given two probability measures p and q over a set X we define the Kullback-Leibler divergence as $\mathrm{KL}(p||q) = \int_X \log \frac{dp}{dq} dp$. We denote with $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$ the Gaussian kernel.

In the derivations below we will rely upon the following useful theorem for the concentration of the spectral norm of sparse random matrices

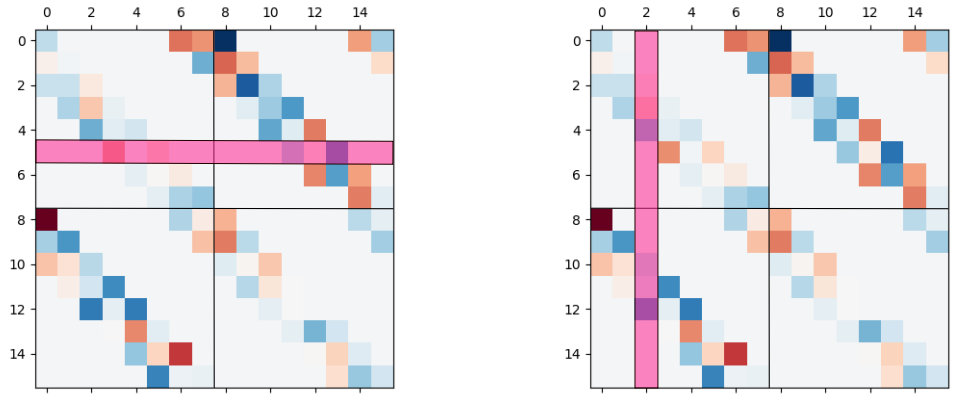
Theorem A.0.1. *Bandeira et al. (2016) Let* $\mathbf{A}$ *be a* $d_2 \times d_1$ *random rectangular matrix with* $\mathbf{A}_{ij} = \xi_{ij}\psi_{ij}$ *where* $\{\xi_{ij} : 1 \leq i \leq d_2, 1 \leq j \leq d_1\}$ *are independent* $\mathcal{N}(0,1)$ *random variables and* $\{\psi_{ij} : 1 \leq i \leq d_2, 1 \leq j \leq d_1\}$ *are scalars. Then*

$$\mathbb{P}(||\mathbf{A}||_2 \geq (1+\epsilon)\{\sigma_1 + \sigma_2 + \frac{5}{\sqrt{\log(1+\epsilon)}}\sigma_* \sqrt{\log(\max(d_2, d_1))} + t\}) \leq e^{-t^2/2\sigma_*^2} \quad \text{(A.1)}$$

for any $0 \leq \epsilon \leq 1/2$ *and* $t \geq 0$ *with*

$$\sigma_1 := \max_i \sqrt{\sum_j \psi_{ij}^2} \qquad \sigma_2 := \max_i \sqrt{\sum_j \psi_{ij}^2} \qquad \sigma_* := \max_{ij} |\psi_{ij}|. \quad \text{(A.2)}$$

In the following we will use the same numbering for Theorems and Lemmas as in the main paper. Theorems and Lemmas unique to the appendix will be numbered with a prefix corresponding to the section where the theorem is introduced and suffix with a corresponding number.



(a) $\sigma_1 := \max_i \sqrt{\sum_j \psi_{ij}^2}$ (b) $\sigma_2 := \max_j \sqrt{\sum_i \psi_{ij}^2}$

Figure A.1 – σ_1 and σ_2: We plot here the structure of a matrix implementing a 1-d locally connected layer with 2 input and 2 output channels. The matrix is a concatenation of 4 matrices that have the structure of 1-d convolutions *without* weight sharing. The purple lines illustrate how for σ_1 and σ_2 are computed. Remembering that $\psi_{ij} = 1$ only if the matrix is non-zero at the location i, j, we see that σ_1 can be seen as the maximal number of non-zeros along the x-axis of the matrix and σ_2 can be found as the maximal number of non-zeros along the y-axis of the matrix. σ_1 is related to the overlap between all input filters for a given output channel and σ_2 is related to the support of the 1-d filters.

A.1 Fully Connected Layers

Lemma A.1.1. *Let* $\mathbf{U} \in \mathbb{R}^{d_2 \times d_1}$ *be the perturbation matrix of a fully connected layer with with row and column sparsity equal to* s*. Then if non-zero elements follow* $\mathbf{U}_{i,j} \sim \mathcal{N}(0, \sigma^2)$*, with probability greater than* $1 - \delta$

$$||\mathbf{U}||_2 \leq \mathcal{O}(\sigma(2\sqrt{s} + \sqrt{2\log(\frac{1}{\delta})})). \tag{A.3}$$

Proof. For $\boldsymbol{u} \sim \mathcal{N}(0, I)$ we need define an index function that allows a Gaussian random noise variable at the locations where the original dense layer is non-zero.

We assume that $\psi_{ij} = 1$ when $\mathbf{U}_{ij} \neq 0$ and is zero otherwise, and get the result trivially from Theorem A.0.1. We can extend the result to $\sigma > 0$ by considering that $||\sigma \mathbf{U}_l||_2 = \sigma ||\mathbf{U}_l||_2$. □

A.2 Locally Connected Layers

Lemma 1.3.7. *Let $\mathbf{U} \in \mathbb{R}^{d_2 \times d_1}$ be the perturbation matrix of a 2d locally connected layer with a input channels, b output channels, filters $\phi \in \mathbb{R}^{q \times q}$ and feature maps $F \in \mathbb{R}^{m \times m}$. Then, if non-zero elements follow $\mathbf{U}_{i,j} \sim \mathcal{N}(0, \sigma^2)$, we have*

$$||\mathbf{U}||_2 \leq \mathcal{O}(\sigma(q[\sqrt{a} + \sqrt{b}] + \sqrt{2\log(\frac{1}{\delta})})), \tag{A.4}$$

with probability greater than $1 - \delta$.

Proof. We will consider first the case $\boldsymbol{u} \sim \mathcal{N}(0, I)$. A convolutional layer is characterized by it's output channels. For each output channel each input channel is convolved with an independent filter resulting in a set of feature maps. For each output channel these feature maps are then summed together. We consider locally connected layers, i.e. the layers are banded in the same way as convolutions but the entries are independent and there is no weight sharing. For the case of one dimensional signals the implied structure is plotted in Figure A.1.

Similar to Lemma A.1.1 we assume that $\psi_{ij} = 1$ when $\mathbf{U}_{ij} \neq 0$ and is zero otherwise. We need to evaluate $\sigma_1 := \max_i \sqrt{\sum_j \psi_{ij}^2}$, $\sigma_2 := \max_j \sqrt{\sum_i \psi_{ij}^2}$ and $\sigma_* := \max_{ij} |\psi_{ij}|$ for a matrix like the one in Figure A.1.

We plot what these sums represent in Figures A.1a, A.1b. We are however working typically with 2 dimensional signals. For σ_1 we can find an upper bound, by considering that the sum for a given filter and a given pixel location represents the maximum number of overlaps for all 2d shifts. For the case of 2d this is q^2, equal to the support of the filters. We plot these shifts in Figure A.2. We also need to consider that there are a input channels. We then get

$$\sigma_1 := \max_i \sqrt{\sum_j \psi_{ij}^2} = \max_i \sqrt{\sum_{j:\psi_{ij}=1} \psi_{ij}^2} = \sqrt{\sum_a \sum_{q^2} 1^2} = \sqrt{aq^2} = q\sqrt{a}. \tag{A.5}$$

For σ_2 each column in the matrix represents a concatenation of convolutional filters $f \in \mathbb{R}^{q \times q}$. The support of the filters is q^2 and there are b filters stacked on top of each other, corresponding to the b output channels. Then it is straight forward to derive that

$$\sigma_2 := \max_j \sqrt{\sum_i \psi_{ij}^2} = \max_j \sqrt{\sum_{i:\psi_{ij}=1} \psi_{ij}^2} = \sqrt{\sum_b \sum_{q^2} 1^2} = \sqrt{bq^2} = q\sqrt{b}. \tag{A.6}$$

Furthermore trivially $\sigma_* = 1$ and when $\sigma > 0$ we can get the result by considering that $||\sigma \mathbf{U}_l||_2 = \sigma ||\mathbf{U}_l||_2$. □

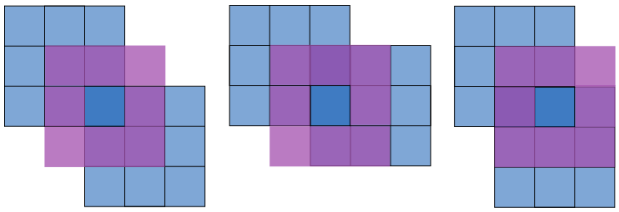


Figure A.2 – Possible shifts with overlap: With blue we plot a 2d filter $f \in \mathbb{R}^{3\times3}$ and 3 filters $f \in \mathbb{R}^{3\times3}$ that overlap with it's bottom right pixel. In purple we plot the box denoting the boundaries of the set of all shifted filters that overlap with the bottom right pixel.

A.3 Detailed proof of Theorem 1.3.1

A.3.1 Convolutional Layers proof of Lemma 1.3.4

Lemma 1.3.4. *Let $\mathbf{U} \in \mathbb{R}^{d_2\times d_1}$ be the perturbation matrix of a 2d convolutional layer with a input channels, b output channels, convolutional filters $\phi \in \mathbb{R}^{q\times q}$ and feature maps $F \in \mathbb{R}^{m\times m}$. We assume that elements $\mathbf{U}_{ij}$ are non-zero only if they correspond to non-zero locations in the sparsity pattern of the convolution operator. Let these elements follow a Gaussian distribution $\mathbf{U}_{ij} \sim \mathcal{N}(0, \sigma^2)$. We have*

$$||\mathbf{U}||_2 \leq \sigma(q[\sqrt{a} + \sqrt{b}] + \sqrt{2\log(\frac{2m^2}{\delta})}), \tag{A.7}$$

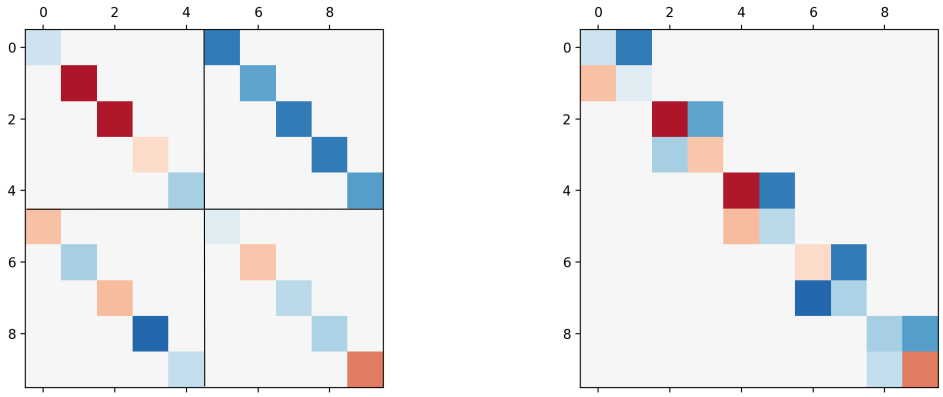
with probability greater than $1 - \delta$.

Proof. We consider "noise" filters $f \in \mathbb{R}^{q\times q}$ and feature maps $F \in \mathbb{R}^{m\times m}$. We define the convolutional noise matrix from input channel j to output channel i in the spatial domain as $\mathbf{A}^{ij} \in \mathbb{R}^{m^2\times m^2}$ and in the frequency domain as $\tilde{\mathbf{A}}^{ij} \in \mathbb{C}^{m^2\times m^2}$ and we denote the Fourier transform matrix as $\mathbf{F} \in \mathbb{C}^{m^2\times m^2}$. Each convolutional matrix corresponds to one convolutional noise filter $f^{ij} \in \mathbb{R}^{q\times q}$. We can now define the structure of the 2d convolutional noise matrix $\mathbf{U}$. Given a input channels and b output channels the noise matrix $\mathbf{U}$ is structured as

$$\mathbf{U} = \begin{bmatrix} \mathbf{A}^{00} & \dots & \mathbf{A}^{0a} \\ \vdots & \ddots & \vdots \\ \mathbf{A}^{b0} & \dots & \mathbf{A}^{ba} \end{bmatrix}. \tag{A.8}$$

For an output channel b the operation $[\mathbf{A}^{00} \cdots \mathbf{A}^{0a}]\boldsymbol{x} = \sum_{i=0}^{a} \mathbf{A}^{0i}\boldsymbol{x}_i$ results in multiplying each channel representation $\boldsymbol{x}_i \in \mathbb{R}^{m^2}$ of the complete signal $\boldsymbol{x} \in \mathbb{R}^{am^2}$ with the convolution filter matrix $\mathbf{A}^{0i}$, where in our case the filter is Gaussian noise.

By exploiting the unitary-invariance property of the spectral norm, we transform this



(a) (b)

Figure A.3 – **Concatenation of diagonal matrices**: Matrix $\mathbf{U}$ can be transformed to the frequency domain and can be written as a concatenation of diagonal frequency matrices A.3a. We can rearrange the columns and rows of this concatenated matrix to obtain a *block diagonal* matrix A.3b, which is easier to analyze theoretically. For example we can create the top left block by moving column 5 to column 1 and then moving row 5 to row 1.

matrix into the Fourier domain to obtain

$$\begin{aligned} ||\mathbf{U}||_2 &= ||(\mathbf{I}_b \otimes \mathbf{F}^T) \begin{bmatrix} \tilde{\mathbf{A}}^{00} & \dots & \tilde{\mathbf{A}}^{0a} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{A}}^{b0} & \dots & \tilde{\mathbf{A}}^{ba} \end{bmatrix} (\mathbf{I}_a \otimes \mathbf{F})||_2 \\ &= || \begin{bmatrix} \tilde{\mathbf{A}}^{00} & \dots & \tilde{\mathbf{A}}^{0a} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{A}}^{b0} & \dots & \tilde{\mathbf{A}}^{ba} \end{bmatrix} ||_2 = || \begin{bmatrix} \tilde{\mathbf{B}}_0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \tilde{\mathbf{B}}_{m^2} \end{bmatrix} ||_2, \end{aligned} \tag{A.9}$$

where we have used the fact that the matrices $\tilde{\mathbf{A}}^{ij}$ are diagonal and a concatenation of diagonal matrices can always be rearranged into block diagonal form. We explain why this the case schematically in Figure A.3. In our case, we have defined blocks

$$\tilde{\mathbf{B}}_{\boldsymbol{n}} = \begin{bmatrix} \lambda_n^{00} & \dots & \lambda_n^{0a} \\ \vdots & \ddots & \dots \\ \lambda_n^{b0} & \dots & \lambda_n^{ba} \end{bmatrix}. \tag{A.10}$$

Where $n \in \{0, \cdots, m^2\}$ denotes a frequency and can be mapped one to one with pairs of 2-dimensional coordinates $n_1 \in \{0, \cdots, m\}, n_2 \in \{0, \cdots, m\}$. The entries λ_n^{ij} of this matrix are

$$\lambda_n^{ij} = \lambda_{n_1,n_2}^{ij} = \sum_{k_1=0}^{q-1}\sum_{k_2=0}^{q-1} e^{-2\pi i(\frac{k_1 n_1}{q}+\frac{k_2 n_2}{q})} f_{k_1,k_2}^{ij}$$
$$= \sum_{k_1=0}^{q-1}\sum_{k_2=0}^{q-1} \cos(2\pi(\frac{k_1 n_1}{q}+\frac{k_2 n_2}{q}))f_{k_1,k_2}^{ij} + i\sum_{k_1=0}^{q-1}\sum_{k_2=0}^{q-1} \sin(2\pi(\frac{k_1 n_1}{q}+\frac{k_2 n_2}{q}))f_{k_1,k_2}^{ij}, \tag{A.11}$$

where n_1,n_2 are the 2-dimensional frequency coordinates, i, j denote the i-th output and j-th input channel.

In this way the n-th block $\tilde{\mathbf{B}}_{\boldsymbol{n}}$ corresponds to the n-th frequency components from the Fourier transforms of all the filters of all the channels, $f^{ij}\ \ \forall i \in \{1,...,b\}, \forall i \in \{1,...,a\}$. We will also need the matrices $\mathrm{Re}(\tilde{\mathbf{B}}_n)$ and $\mathrm{Im}(\tilde{\mathbf{B}}_n)$

$$\mathrm{Re}(\tilde{\mathbf{B}}_n) = \begin{bmatrix} \mathrm{Re}(\lambda_n^{00}) & \dots & \mathrm{Re}(\lambda_n^{0a}) \\ \vdots & \ddots & \dots \\ \mathrm{Re}(\lambda_n^{b0}) & \dots & \mathrm{Re}(\lambda_n^{ba}) \end{bmatrix}, \quad \mathrm{Im}(\tilde{\mathbf{B}}_n) = \begin{bmatrix} \mathrm{Im}(\lambda_n^{00}) & \dots & \mathrm{Im}(\lambda_n^{0a}) \\ \vdots & \ddots & \dots \\ \mathrm{Im}(\lambda_n^{b0}) & \dots & \mathrm{Im}(\lambda_n^{ba}) \end{bmatrix}. \tag{A.12}$$

Given that the elements of all the "noise" filters f_{k_1,k_2}^{ij} have Gaussian distributions, the entries $\mathrm{Re}(\lambda_n^{ij}), \mathrm{Im}(\lambda_n^{ij})$ of these matrices have the following distributions

$$\begin{aligned} \mathrm{Re}(\lambda_n^{ij}) &\sim \mathcal{N}(0, \sigma_{\mathrm{re},n}^2) = \mathcal{N}(0, \sum_{k_1=0}^{q-1}\sum_{k_2=0}^{q-1} \cos^2(2\pi(\frac{k_1 n_1}{q}+\frac{k_2 n_2}{q})) \\ \mathrm{Im}(\lambda_n^{ij}) &\sim \mathcal{N}(0, \sigma_{\mathrm{im},n}^2) = \mathcal{N}(0, \sum_{k_1=0}^{q-1}\sum_{k_2=0}^{q-1} \sin^2(2\pi(\frac{k_1 n_1}{q}+\frac{k_2 n_2}{q})). \end{aligned} \tag{A.13}$$

We have now turned our initial problem into a form that lends itself more easily to a solution. Our original matrix has been turned into a block diagonal form and each block can be split into real and imaginary parts that have independent Gaussian entries, we note however that blocks are not independent of each other. We will now derive a concentration bound on the original matrix by using the fact that the spectral norm of a block diagonal matrix is equal to the maximum of the spectral norms of the individual blocks.

We can write the following inequalities

$$\mathbb{P}(||\mathbf{U}||_2 \leq \epsilon) = \mathbb{P}(\bigcap_n \{||\tilde{\mathbf{B}}_n||_2 \leq \epsilon\}) \geq \mathbb{P}(\bigcap_n \{||\mathrm{Re}(\tilde{\mathbf{B}}_n)||_2 + ||\mathrm{Im}(\tilde{\mathbf{B}}_n)||_2 \leq \epsilon\}). \tag{A.14}$$

By setting $\epsilon = \max_n(\epsilon_n)$ and ϵ_n arbitrary constants, we can furthermore write

$$\begin{aligned}
&\mathbb{P}(||\mathbf{U}||_2 \leq \max_n(\epsilon_n)) \\
&\geq \mathbb{P}(\bigcap_n \{||\mathrm{Re}(\tilde{\mathbf{B}}_n)||_2 + ||\mathrm{Im}(\tilde{\mathbf{B}}_n)||_2 \leq \max_n(\epsilon_n)\}) \\
&\geq \mathbb{P}(\bigcap_n \{||\mathrm{Re}(\tilde{\mathbf{B}}_n)||_2 + ||\mathrm{Im}(\tilde{\mathbf{B}}_n)||_2 \leq \epsilon_n\}) \\
&\geq \mathbb{P}(\bigcap_n [\{||\mathrm{Re}(\tilde{\mathbf{B}}_n)||_2 \leq \epsilon_{n,\mathrm{re}}\} \cap \{||\mathrm{Im}(\tilde{\mathbf{B}}_n)||_2 \leq \epsilon_{n,\mathrm{im}}\}]) \\
&\geq 1 - \sum_{n=1}^{m^2} [\delta_{n,\mathrm{re}} + \delta_{n,\mathrm{re}}],
\end{aligned} \tag{A.15}$$

where in line 4 we set $\epsilon_n = \epsilon_{n,\mathrm{re}} + \epsilon_{n,\mathrm{im}}$ and in line 5 we used a union bound and assumed that

$$\mathbb{P}(||\mathrm{Re}(\tilde{\mathbf{B}}_{\boldsymbol{n}})||_2 \geq \epsilon_{n,\mathrm{re}}) \leq \delta_{n,\mathrm{re}},$$
$$\mathbb{P}(||\mathrm{Im}(\tilde{\mathbf{B}}_{\boldsymbol{n}})||_2 \geq \epsilon_{n,\mathrm{im}}) \leq \delta_{n,\mathrm{im}},$$

for positive constants $\{\epsilon_{n,\mathrm{re}}, \epsilon_{n,\mathrm{im}}, \delta_{n,\mathrm{re}}, \delta_{n,\mathrm{im}}\} \in \mathbb{R}_+$.

We will now calculate concentration inequalities for the matrices $\mathrm{Re}(\tilde{\mathbf{B}}_n)$ and $\mathrm{Im}(\tilde{\mathbf{B}}_n)$, to derive a specific instantiation of the above general formula, for our setting. To do that we first apply the following concentration inequality by Vershynin (2010) on the matrices $\mathrm{Re}(\tilde{\mathbf{B}}_n)$ and $\mathrm{Im}(\tilde{\mathbf{B}}_n)$.

Theorem A.3.1. *Let* $\mathbf{A}$ *be an* $a \times b$ *matrix whose entries are independent Gaussian random variables with variance* σ^2*. Then for every* $t \geq 0$

$$\mathbb{P}(||\mathbf{A}||_2 \geq \sigma(\sqrt{a} + \sqrt{b} + \sqrt{2\ln(\frac{1}{\delta})})) \leq \delta. \tag{A.16}$$

We obtain the following concentration inequalities

$$\begin{aligned}
\mathbb{P}(||\mathrm{Re}(\tilde{\mathbf{B}}_{\boldsymbol{n}})||_2 \geq \sigma_{\mathrm{re},n}(\sqrt{a} + \sqrt{b} + \sqrt{2\ln(\frac{1}{\delta_{n,\mathrm{re}}})})) \leq \delta_{n,\mathrm{re}}, \\
\mathbb{P}(||\mathrm{Im}(\tilde{\mathbf{B}}_{\boldsymbol{n}})||_2 \geq \sigma_{\mathrm{im},n}(\sqrt{a} + \sqrt{b} + \sqrt{2\ln(\frac{1}{\delta_{n,\mathrm{im}}})})) \leq \delta_{n,\mathrm{im}}.
\end{aligned} \tag{A.17}$$

We will use the fact that $\max_n[\sigma_{\mathrm{re},n} + \sigma_{\mathrm{im},n}] \leq 1.5q$ and substitute $\delta_{n,\mathrm{re}} = \delta_{n,\mathrm{im}} =$

$\delta/(2m^2)$, $\forall n \in \{1, ..., m^2\}$ in equation (A.15). We get

$$\begin{aligned}
&\mathbb{P}(||\mathbf{U}||_2 \leq \max_n(\epsilon_n)) \\
&= \mathbb{P}(||\mathbf{U}||_2 \leq \max_n[(\sigma_{\mathrm{re},n} + \sigma_{\mathrm{im},n})(\sqrt{a} + \sqrt{b} + \sqrt{2\ln(\frac{2m^2}{\delta})})]) \\
&= \mathbb{P}(||\mathbf{U}||_2 \leq 1.5q(\sqrt{a} + \sqrt{b} + \sqrt{2\ln(\frac{2m^2}{\delta})})) \\
&\geq 1 - \sum_{n=1}^{m^2}[\delta_{n,\mathrm{re}} + \delta_{n,\mathrm{im}}] \\
&= 1 - \sum_{n=1}^{m^2}[\frac{\delta}{2m^2} + \frac{\delta}{2m^2}] = 1 - \sum_{n=1}^{m^2}\frac{\delta}{m^2} = 1 - \delta,
\end{aligned} \tag{A.18}$$

which implies the desired result.

We now prove that $\max_n[\sigma_{\mathrm{re},n} + \sigma_{\mathrm{im},n}] \leq 1.5q$. First using the derivative test we will the maxima of $\sqrt{\sum_l \sin^2(\theta_l)} + \sqrt{\sum_l \cos^2(\theta_l)}$. We see that

$$\begin{aligned}
\frac{\partial}{\partial\theta_l}(\sqrt{\sum_l \sin^2(\theta_l)} + \sqrt{\sum_l \cos^2(\theta_l)}) &= \frac{1}{2}\frac{2\cos(\theta_l)\sin(\theta_l)}{|\sin(\theta_l)|} - \frac{1}{2}\frac{2\cos(\theta_l)\sin(\theta_l)}{|\cos(\theta_l)|} \\
&= \frac{\sin(\theta_l)\cos(\theta_l)}{|\sin(\theta_l)||\cos(\theta_l)|}(|\cos(\theta_l)| - |\sin(\theta_l)|) = 0,
\end{aligned} \tag{A.19}$$

which implies that at the maximum

$$\cos(\theta_l) = \sin(\theta_l) = \pm\frac{1}{\sqrt{2}}. \tag{A.20}$$

Then we can calculate

$$\begin{aligned}
\max_n[\sigma_{\mathrm{re},n} + \sigma_{\mathrm{im},n}] = \max_n[&\sqrt{\sum_{k_1=0}^{q-1}\sum_{k_2=0}^{q-1}\cos^2(2\pi(\frac{k_1 n_1}{q} + \frac{k_2 n_2}{q})} \\
&+ \sqrt{\sum_{k_1=0}^{q-1}\sum_{k_2=0}^{q-1}\sin^2(2\pi(\frac{k_1 n_1}{q} + \frac{k_2 n_2}{q})}] \\
&\leq \sqrt{\sum_{k_1=0}^{q-1}\sum_{k_2=0}^{q-1}\frac{1}{2}} + \sqrt{\sum_{k_1=0}^{q-1}\sum_{k_2=0}^{q-1}\frac{1}{2}} = \frac{2}{\sqrt{2}}q \leq 1.5q.
\end{aligned} \tag{A.21}$$

□

A.3.2 Putting everything together

We now restate the PAC-Bayes theorem of McAllester (1999).

Theorem 1.4.1. *McAllester (1999) Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, a hypothesis set $\mathcal{F}$, a loss function $\ell' : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to [0,1]$, a prior distribution π over $\mathcal{F}$, a real number $\delta \in (0,1]$, and a real number $\beta > 0$, with probability at least $1-\delta$ over the choice of $(X,Y) \sim \mathcal{D}^n$, we have*

$$\forall \hat{\rho} \text{ on } \mathcal{F} : \mathbf{E}_{f\sim\hat{\rho}} \mathcal{L}^{\ell'}_{\mathcal{D}}(f) \leq \mathbf{E}_{f\sim\hat{\rho}} \hat{\mathcal{L}}^{\ell'}_{X,Y}(f) + \sqrt{\frac{\mathrm{KL}(\hat{\rho}||\pi) + \ln \frac{2n}{\delta}}{2n-1}}. \tag{A.22}$$

Notice that the above gives a generalization result over a distribution of predictors. We now restate a useful lemma which can be used to give a generalization result for a single predictor instance.

Lemma 1.3.2 (Neyshabur et al. (2017b))**.** *We assume a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, a hypothesis set $\mathcal{F}$ parametrized by classifiers $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$, loss functions $\ell_0, \ell_\gamma : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to [0,1]$, a prior distribution π over $\boldsymbol{\theta}$, a real number $\delta \in (0,1]$, a real number $\gamma > 0$ and deterministic weights $\boldsymbol{\theta}_\star$. Then, for any random variable $\boldsymbol{u}$ s.t. $\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}$, we have with probability at least $1-\delta$ over the choice of $(X,Y) \sim \mathcal{D}^n$,*

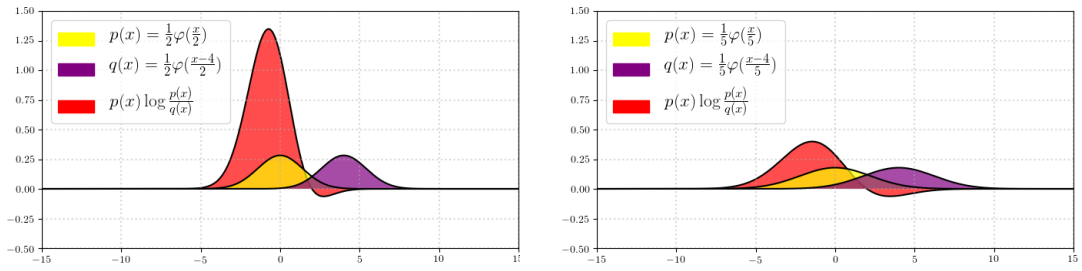
$$\mathcal{L}^{\ell_0}_{\mathcal{D}}(f_{\boldsymbol{\theta}_\star}) \leq \hat{\mathcal{L}}^{\ell_\gamma}_{X,Y}(f_{\boldsymbol{\theta}_\star}) + \mathcal{O}\left(\sqrt{\frac{\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta}_\star + \boldsymbol{u})||\pi) + \ln\frac{6n}{\delta}}{n-1}}\right) \tag{A.23}$$

where n is the number of training samples.

Contrary to Theorem 1.4.1, Lemma 1.3.2 links the empirical risk $\hat{\mathcal{L}}^{\ell_\gamma}_{X,Y}(f_{\boldsymbol{\theta}_\star})$ of the predictor to the true risk $\mathcal{L}^{\ell_0}_{\mathcal{D}}(f_{\boldsymbol{\theta}_\star})$, for a specific predictor and not a posterior distribution of predictors. We have also moved to using a margin γ based loss. The perturbation $\boldsymbol{u}$ quantifies quantifies the simplicity of the predictor. The more noise we can add to the parameters without raising the empirical risk, the lower the precision with which we can encode the parameters, and the simpler the classifier is. The condition $\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}$ can be interpreted as choosing a posterior with *small variance*, sufficiently concentrated around the current empirical estimate $\boldsymbol{\theta}_\star$, so that we can remove the randomness assumption with high confidence.

How small should we choose the the variance of $\boldsymbol{u}$? The choice is complicated because the KL term in the bound is *inversely proportional* to the variance of the perturbation (Figure A.4). Therefore we need to find the largest possible variance for which our stability condition holds.

Let $\beta = (\prod_{i=0}^{l} ||\mathbf{W}_i||_2)^{1/l}$ and consider a network with the normalized weights $\tilde{\mathbf{W}}_i =$



(a) $\sigma = 2$ (b) $\sigma = 5$

Figure A.4 – $D_{\mathrm{KL}}(p||q)$ **with** $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{x^2}{2\sigma^2}}$ **and** $q(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-4)^2}{2\sigma^2}}$: By definition the KL divergence of the two distributions is the integral of the red curve. We see that as the variance increases the KL divergence between the two distributions decreases.

$\frac{\beta}{||\mathbf{W}_i||_2}\mathbf{W}_i$. Due to the homogeneity of the ReLu and Max-Pooling, we have that for feedforward neural networks with ReLu activations $f_{\tilde{\boldsymbol{\theta}}_\star} = f_{\boldsymbol{\theta}_\star}$ and so the (empirical and the expected) loss (including margin loss) is the same for $\tilde{\boldsymbol{\theta}}_\star = \boldsymbol{\theta}_\star$. We can also verify that $(\prod_{i=0}^{l}||\mathbf{W}_i||_2) = (\prod_{i=0}^{l}||\tilde{\mathbf{W}}_i||_2)$ and $\frac{||\mathbf{W}_i||_F}{||\mathbf{W}_i||_2} = \frac{||\tilde{\mathbf{W}}_i||_F}{||\tilde{\mathbf{W}}_i||_2}$, and so the excess error in the theorem statement is also invariant to this transformation. It is therefore sufficient to prove the theorem only for normalized weights $\tilde{\boldsymbol{\theta}}_\star$, and hence we assume w.l.o.g. that the spectral norm is equal across layers, i.e. for any layer i, $||\mathbf{W}_i||_2$.

The prior cannot depend on the learned predictor $\boldsymbol{\theta}_\star$ or it's norm, we will set σ based on an approximation $\tilde{\beta}$. For each value of $\tilde{\beta}$ on a pre-determined grid, we will compute the PAC-Bayes bound, establishing the generalization guarantee for all $\boldsymbol{\theta}_\star$ for which $|\beta - \tilde{\beta}| \leq \frac{1}{d}\beta$, and ensuring that each relevant value of β is covered by some $\tilde{\beta}$ on the grid. We will then take a union bound over all $\tilde{\beta}$ on the grid. In the previous we have considered a fixed $\tilde{\beta}$ and the $\boldsymbol{\theta}_\star$ for which $|\beta - \tilde{\beta}| \leq \frac{1}{d}\beta$, and hence $\frac{1}{e}\beta^{d-1} \leq \tilde{\beta}^{d-1} \leq e\beta^{d-1}$.

Characterizing the condition $\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}}|f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}$ entails understanding the sensitivity of our classifier on random perturbations. To that end, we review here a useful perturbation bound from Neyshabur et al. (2017b) on the output of a DNN

Lemma 1.3.3. *(Perturbation Bound). For any $B, l > 0$, let $f_{\boldsymbol{\theta}_\star} : \mathcal{X}_{B,d} \to \mathbb{R}^k$ be a d-layer network with ReLU activations. Then, for any $\boldsymbol{\theta}_\star$, and $\boldsymbol{x} \in \mathcal{X}_{B,d}$, and perturbation $\boldsymbol{u} = vec(\{\mathbf{U}_i\}_{i=0}^{l})$ such that $||\mathbf{U}_i||_2 \leq \frac{1}{l}||\mathbf{W}_i||_2$, the change in the output of the network can be bounded as follows*

$$|f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq e^2 B\tilde{\beta}^{l-1}\sum_i ||\mathbf{U}_i||_2, \tag{A.24}$$

where e, B and $\tilde{\beta}^{l-1}$ are considered as constants after an appropriate normalization of the layer weights.

We note that correctly estimating the spectral norm of the perturbation at each layer is critical to obtaining a tight bound. Specifically if we exploit the structure of the perturbation we can *increase significantly* the variance of the added perturbation for which our stability condition holds.

We need to find the maximum variance for which

$$\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}.$$

For this we will use Lemmas 1.3.4 and A.1.1 which bound the spectral norm of the noise at each convolutional layer and sparse fully connected layer respectively.

Lemma 1.3.5. *(Perturbation Bound). For any $B, l > 0$, let $f_{\boldsymbol{\theta}_\star} : \mathcal{X}_{B,d} \to \mathbb{R}^k$ be a l-layer network with ReLU activations and we denote $\mathcal{C}$ the set of convolutional layers and $\mathcal{F}$ the set of fully connected layers. Then for any $\boldsymbol{\theta}_\star$, and $\boldsymbol{x} \in \mathcal{X}_{B,d}$, and a perturbation for $\boldsymbol{u} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$, for any $\gamma > 0$ with*

$$\sigma = \frac{\gamma}{42B\tilde{\beta}^{l-1}[\sum_{i\in\mathcal{C}} K_i + \sum_{i\in\mathcal{F}} J_i]}, \tag{A.25}$$

we have

$$\mathbb{P}_{\boldsymbol{u}}[\max_{\boldsymbol{x}\in\boldsymbol{\mathcal{X}}} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}] \geq \frac{1}{2}, \tag{A.26}$$

where e, B, $\tilde{\beta}^{l-1}$ are considered as constants after an appropriate normalization of the layer weights

$$K_i = q_i\{\sqrt{a_i} + \sqrt{b_i} + \sqrt{2\log(4m_i^2 l)}\}, \tag{A.27}$$

and

$$J_i = q_i\{2\sqrt{s_i} + \sqrt{2\log(2l)}\}. \tag{A.28}$$

Proof. We denote $\mathcal{C}$ the set of convolutional layers, $\mathcal{F}$ the set of fully connected layers and assume $|\mathcal{C}| + |\mathcal{F}| = l$ where l is the total number of layers. We define events $||\mathbf{U}_i||_2 \leq 2\sqrt{s_i}+\sqrt{2\log(2l)}$ for the fully connected layers and $||\mathbf{U}_i||_2 \leq q_i\{\sqrt{a_i}+\sqrt{b_i}+\sqrt{2\log(4m_i^2 l)}\}$ for the convolutional layers. We then assume that the probability for each of the $|\mathcal{F}|$ and $|\mathcal{C}|$ events is upper bounded by $\frac{1}{2l}$. We set $K_i = q_i\{\sqrt{a_i} + \sqrt{b_i} + \sqrt{2\log(4m_i^2 l)}\}$ and $J_i = \{2\sqrt{s_i} + \sqrt{2\log(2l)}\}$ and take a union bound over the above events. After some calculations we obtain

$$\mathbb{P}(\sum_i ||\mathbf{U}_i||_2 \leq \sigma[\sum_{i\in\mathcal{C}} K_i + \sum_{i\in\mathcal{F}} J_i]) \geq 1 - (\sum_{i\in\mathcal{C}} \frac{1}{2l} + \sum_{i\in\mathcal{F}} \frac{1}{2l}) = 1 - \frac{1}{2} = \frac{1}{2}. \tag{A.29}$$

We are then ready to apply our result directly to Lemma 1.3.3. We calculate that with probability $\geq \frac{1}{2}$

$$\begin{aligned} |f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 &\leq e^2 B\tilde{\beta}^{l-1}\sum_i ||\mathbf{U}_i||_2 \\ &\leq \sigma e^2 B\tilde{\beta}^{l-1}[\sum_{i\in\mathcal{C}} K_i + \sum_{i\in\mathcal{F}} J_i]. \end{aligned} \tag{A.30}$$

We have now found a bound on the perturbation at the final layer of the network as a function of σ with probability $\geq \frac{1}{2}$. What remains is to find the specific value of σ such that $|f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4}$. We calculate

$$\begin{aligned} &|f_{\boldsymbol{\theta}_\star+\boldsymbol{u}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})|_2 \leq \frac{\gamma}{4} \\ &\Rightarrow \sigma e^2 B\tilde{\beta}^{l-1}[\sum_{i\in\mathcal{C}} K_i + \sum_{i\in\mathcal{F}} J_i] \leq \frac{\gamma}{4} \\ &\Rightarrow \sigma \leq \frac{\gamma}{42B\tilde{\beta}^{l-1}[\sum_{i\in\mathcal{C}} K_i + \sum_{i\in\mathcal{F}} J_i]}. \end{aligned} \tag{A.31}$$

□

We can now calculate the KL term in Theorem 4.1. by noting that $\hat{\rho}(\boldsymbol{\theta}_\star+\boldsymbol{u}) = \mathcal{N}(\boldsymbol{\theta}_\star, \sigma^2\mathbf{I})$, $\pi(\boldsymbol{\theta}) = \mathcal{N}(0, \sigma^2\mathbf{I})$, and that then $\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta}_\star + \boldsymbol{u})||\pi) \leq \frac{|\boldsymbol{\theta}_\star|^2}{2\sigma^2}$. We get that for any $\tilde{\beta}$, with probability $\geq 1-\delta$ and for all $\boldsymbol{\theta}_\star$ such that, $|\beta - \tilde{\beta}| \leq \frac{1}{l}\beta$

$$\mathcal{L}^{\ell_0}_{\mathcal{D}}(f_{\boldsymbol{\theta}_\star}) \leq \hat{\mathcal{L}}^{\ell_\gamma}_{X,Y}(f_{\boldsymbol{\theta}_\star}) + \mathcal{O}\left(\frac{B\,\Psi_f\,R_{\boldsymbol{\theta}_\star}}{\gamma\sqrt{n}}\right),$$

with $\|x\|_2 \leq B$ being a uniform bound on the input vectors, $\Psi_f = q\sum_{i\in\mathcal{C}}\sqrt{b_i} + \sum_{i\in\mathcal{F}}\sqrt{s_i}$, and

$$R_{\boldsymbol{\theta}_\star} := \prod_{i=0}^{l} ||\mathbf{W}_i||_2 \left(\sum_{i=0}^{l} \frac{||\mathbf{W}_i||_F^2}{||\mathbf{W}_i||_2^2}\right)^{\frac{1}{2}}. \tag{A.32}$$

Finally, we need to take a union bound over different choices of $\tilde{\beta}$. Let us see how many choices of $\tilde{\beta}$ we need to ensure we always have $\tilde{\beta}$ in the grid s.t. $|\beta - \tilde{\beta}| \leq \frac{1}{l}\beta$. We only need to consider values of β in the range $(\frac{\gamma}{2B})^{1/l} \leq \beta \leq (\frac{\gamma\sqrt{n}}{2B})^{1/l}$. For β outside this range the theorem statement holds trivially: Recall that the LHS of the theorem statement, $\mathcal{L}^{\ell_0}_{\mathcal{D}}(f_{\boldsymbol{\theta}_\star})$ is always bounded by 1. If $\beta^l < \frac{\gamma}{2B}$, then for any $\boldsymbol{x}$, $|f_{\boldsymbol{\theta}_\star}(\boldsymbol{x})| \leq \beta^l B \leq \gamma/2$ and therefore $\hat{\mathcal{L}}^{\ell_\gamma}_{X,Y}(f_{\boldsymbol{\theta}_\star}) = 1$. Alternatively, if $\beta^l > \frac{\gamma\sqrt{n}}{2B}$, then the second term in equation A.22 is greater than one. Hence, we only need to consider values of β in the range discussed above. Since we need $\tilde{\beta}$ to satisfy $|\beta - \tilde{\beta}| \leq \frac{1}{l}\beta \leq \frac{1}{l}(\frac{\gamma}{2B})^{1/l}$, the size of the cover we need to consider is bounded by $n^{\frac{1}{2l}}l$. Taking a union bound over the choices of $\tilde{\beta}$ in this cover and using the bound $\mathcal{L}^{\ell_0}_{\mathcal{D}}(f_{\boldsymbol{\theta}_\star}) \leq \hat{\mathcal{L}}^{\ell_\gamma}_{X,Y}(f_{\boldsymbol{\theta}_\star}) + \mathcal{O}\left(\frac{B\,\Psi_f\,R_{\boldsymbol{\theta}_\star}}{\gamma\sqrt{n}}\right)$, gives us the

theorem statement.

A.4 Additional experiments on the Bartlett Metric

We include a number of additional experiments on the metric by Bartlett et al. (2017). The experimental setup is identical to the own used for the Neyshabur metric. We note that the conclusions we can draw are similar in both cases. They indicate a limitations of spectral complexity based generalization bounds in general.

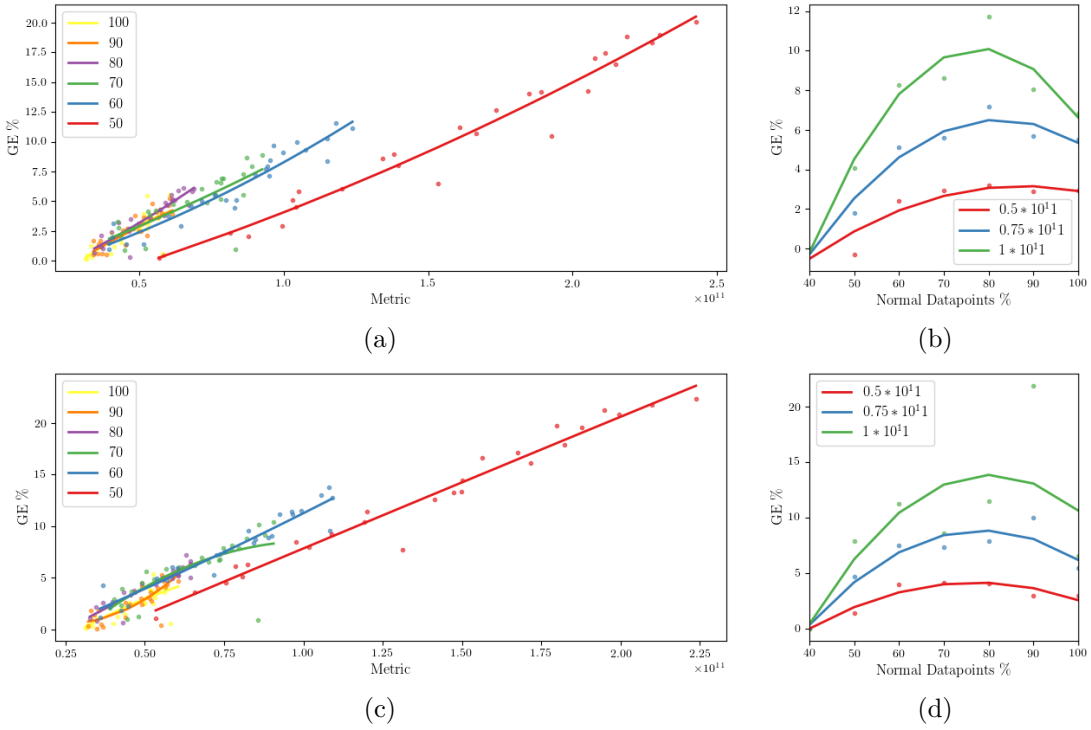


Figure A.5 – **Varying the percentage of translations (a-b) and elastic deformations (c-d)**: We split Training and Testing datasets of constant size into two parts—the first contains images that form a base space, whereas the rest of the dataset contains images that are augmentations of the base space. The percentage values indicate the percentage of the *augmentations* over the total dataset. (a/c) We plot the GE vs spectral complexity. As we increase the number of translations/elastic deformations (equivalently decrease the percentage of the base space) the slopes of the GE curves decrease and we tend to have lower GE for the same spectral complexity metric values. (b/d) We plot the GE vs % of augmentations for constant complexity values. The percentage of augmentations correlates empirically with the GE indicating that spectral complexity does not account for the architecture invariances.

B Appendix

B.1 Derivations for valid bound

We present again for clarity the PAC-Bayes bound by Catoni (2007).

Theorem 2.2.1. *Catoni (2007) Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, a hypothesis set $\mathcal{F}$, a loss function $\ell' : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to [0,1]$, a prior distribution π over $\mathcal{F}$, a real number $\delta \in (0,1]$, and a real number $\beta > 0$, with probability at least $1-\delta$ over the choice of $(X,Y) \sim \mathcal{D}^n$, we have*

$$\begin{aligned} \forall \hat{\rho} \text{ on } \mathcal{F} : \mathbf{E}_{f\sim\hat{\rho}} \mathcal{L}^{\ell'}_{\mathcal{D}}(f) \leq & \Phi^{-1}_{\beta}(\mathbf{E}_{f\sim\hat{\rho}} \hat{\mathcal{L}}^{\ell'}_{X,Y}(f) \\ & + \frac{1}{\beta n}(\mathrm{KL}(\hat{\rho}||\pi) + \ln\frac{1}{\delta})), \end{aligned} \tag{B.1}$$

where $\Phi^{-1}_{\beta}(x) = \frac{1-e^{-\beta x}}{1-e^{-\beta}}$.

Evaluating a valid PAC-Bayes bound, using empirical estimates, requires some care.

Optimizing λ. For a start, when modeling $\pi(\boldsymbol{\theta}) = \mathcal{N}(0, \lambda\mathbf{I})$, it is often beneficial to optimize the hyperparameter λ. As the PAC-Bayes theorem requires the prior to be independent from the posterior, we need to take a union bound over an appropriately chosen grid, representing different possible values of λ. Following Dziugaite and Roy (2017), we can choose $\lambda = c\exp\{-j/b\}$ for $j \in \mathbb{N}$ and fixed $b, c \geq 0$, where c corresponds to the grid scale and b to it's precision. Then, if the PAC-Bayes bound for each $j \in \mathbb{N}$ is designed to hold with probability at least $1 - \frac{6\delta}{\pi^2 j^2}$, by union bound it will hold uniformly for all $j \in \mathbb{N}$ with probability at least $1 - (\frac{6\delta}{\pi^2})\sum_{j\in\mathbb{N}} \frac{1}{j^2} = 1 - \delta$. We solve for $j = b\log\frac{c}{\lambda}$ and substitute this value in the probability for each term in the union bound. We get that *any* bound corresponding to $j \in \mathbb{N}$ holds with probability $1 - \frac{6\delta}{\pi^2 b^2 \ln(c/\lambda^2)}$. Thus looking back to theorem 2.2.1 the term $\ln\frac{1}{\delta}$ becomes $\ln\frac{\pi^2 b^2 \ln(c/\lambda^2)}{6\delta}$. In practice we see that even for very large numbers c, b, δ when divided by the number of samples n the term $\ln\frac{\pi^2 b^2 \ln(c/\lambda^2)}{6\delta}$ is negligible and we treat j as a continuous number.

Empirical estimate of $\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}^*(\boldsymbol{\theta})}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})$. Furthermore, assuming an optimized posterior $\hat{\rho}^*(\boldsymbol{\theta})$ directly evaluating $\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}^*(\boldsymbol{\theta})}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})$ is intractable. Instead, since $\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})$ is a bounded random variable, one can approximate the expectation using Monte Carlo sampling and use a Chernoff bound to bound it's tail. Let $\tilde{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}}) \equiv (1/m)\sum_{i=0}^{m}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}_i})$ be the observed failure rate of m random hypotheses drawn according to $\hat{\rho}^*(\boldsymbol{\theta})$. One can then show the following (Langford and Caruana, 2002) (presented here without proof)

Theorem B.1.1. *(Sample Convergence Bound) For all distributions, $\hat{\rho}^*(\boldsymbol{\theta})$, for all sample sets (X, Y), assuming that $\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}}) \in [0, 1]$*

$$\begin{aligned}&\Pr_{\hat{\rho}^*(\boldsymbol{\theta})}(\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}^*(\boldsymbol{\theta})}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}}) \leq \tilde{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}}) + \sqrt{\frac{\ln\frac{2}{\delta'}}{m}})\\ &\leq \delta',\end{aligned} \tag{B.2}$$

where m is the number of evaluations of the stochastic hypothesis.

We take a union bound over values of λ, and apply the Chernoff bound for the tail of the empirical estimate of $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f)$. Putting everything together, one can obtain valid PAC-Bayes bounds subject to a posterior distribution $\hat{\rho}^*(\boldsymbol{\theta})$ that hold with probability at least $1-\delta-\delta'$ and are of the form

$$\begin{aligned}\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}^*(\boldsymbol{\theta})}\mathcal{L}^{\ell'}_{\mathcal{D}}(f_{\boldsymbol{\theta}}) \leq &\Phi_{\beta}^{-1}(\tilde{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}}) + \frac{1}{\beta n}\mathrm{KL}(\hat{\rho}^*(\boldsymbol{\theta})||\pi)\\ &+ \frac{1}{\beta n}\ln(\frac{\pi^2 b^2\ln(c/\lambda)^2}{6\delta}) + \sqrt{\frac{\ln\frac{2}{\delta'}}{m}}),\end{aligned} \tag{B.3}$$

where $\Phi_{\beta}^{-1}(x) = \frac{1-e^{-\beta x}}{1-e^{-\beta}}$. Also c, b are constants, m is the number of samples from $\hat{\rho}$ for approximating $\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}^{\ell'}_{X,Y}(f)$ and $\tilde{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}})$ the empirical estimate.

Number of samples for Chernoff bound. In our experiments we use $m = 1000$ for all experiments including VI experiments. We make a single exception due to time constraints for the case of optimizing the posterior in closed form (Section 4.1 equation 6) where we use $m = 100$.

- For $m = 1000$ and $\delta' = 0.05$, this gives bounds with confidence $\sqrt{\frac{\log(2/0.05)}{1000}} \approx 0.06$.
- For $m = 100$ and $\delta' = 0.05$, this gives bounds with confidence $\sqrt{\frac{\log(2/0.05)}{100}} \approx 0.19$.

Importantly bounds with even higher confidence $\sqrt{\frac{\log(2/0.05)}{10000}} \approx 0.019$ and sample size $m = \mathcal{O}(10^4)$ are possible for all experiments with a computational time in the order of

weeks. However we consider this point a technicality as the Chernoff bound is quite pessimistic. Empirically the estimates in our experiments converge much faster than implied by the bound analysis, exhibiting no significant difference between $m = 1000$, $m = 100$ or even $m = 10$ in the isotropic cases. This is because this particular Chernoff bound is an application of Hoeffding's inequality for general bounded random variables (Vershynin, 2018)[p. 25]. The only assumption is that the random variable is bounded $\hat{\mathcal{L}}^{\ell'}_{X,Y}(f_{\boldsymbol{\theta}}) \in [0, 1]$, and thus the variance of the random variable is significantly overestimated.

B.2 Proof of Lemma 2.4.1

Lemma 2.4.1. *The optimization problem* $\min_{\mathbf{\Sigma}_{\hat{\rho}}} \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}\boldsymbol{\eta}] + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta}))$ *where* $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \mathbf{\Sigma}_{\hat{\rho}})$ *and* $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi}, \lambda\mathbf{\Sigma}_{\pi})$ *is convex and is minimized at*

$$\mathbf{\Sigma}^*_{\hat{\rho}} = \beta(\mathbf{H} + \frac{\beta}{\lambda}\mathbf{\Sigma}_{\pi}^{-1})^{-1}, \tag{B.4}$$

where $\mathbf{H} \equiv \nabla^2\hat{\mathcal{L}}^{\ell_{cat}}_{X,Y}(f_{\boldsymbol{\theta}})$ *captures the curvature at the minimum, while* $\mathbf{\Sigma}_{\pi}$ *is the prior covariance.*

Proof.

$$\begin{aligned}
C_\beta(X, Y; \hat{\rho}, \pi) &= \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}\boldsymbol{\eta}] + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) \\
&= \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\mathrm{tr}(\mathbf{H}\boldsymbol{\eta}\boldsymbol{\eta}^T)] + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) \\
&= \frac{1}{2}\mathrm{tr}(\mathbf{H}\mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\boldsymbol{\eta}\boldsymbol{\eta}^T]) + \beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) \\
&= \frac{1}{2}\mathrm{tr}(\mathbf{H}\mathbf{\Sigma}_{\hat{\rho}}) + \frac{\beta}{2}(\mathrm{tr}(\frac{1}{\lambda}\mathbf{\Sigma}_{\pi}^{-1}\mathbf{\Sigma}_{\hat{\rho}}) - k + \frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi})^{\mathrm{T}}\mathbf{\Sigma}_{\pi}^{-1}(\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_{\pi}) \\
&\quad + \ln\left(\frac{\det\lambda\mathbf{\Sigma}_{\pi}}{\det\mathbf{\Sigma}_{\hat{\rho}}}\right))
\end{aligned} \tag{B.5}$$

The gradient with respect to $\mathbf{\Sigma}_{\hat{\rho}}$ is

$$\frac{\partial C_\beta(X, Y; \hat{\rho}, \pi)}{\partial\mathbf{\Sigma}_{\hat{\rho}}} = [\frac{1}{2}\mathbf{H} + \frac{\beta}{2\lambda}\mathbf{\Sigma}_{\pi}^{-1} - \frac{\beta}{2}\mathbf{\Sigma}_{\hat{\rho}}^{-1}]. \tag{B.6}$$

Setting it to zero, we obtain the minimizer $\mathbf{\Sigma}^*_{\hat{\rho}} = \beta(\mathbf{H} + \frac{\beta}{\lambda}\mathbf{\Sigma}_{\pi}^{-1})^{-1}$. □

B.3 Proof of Lemma 2.4.2

Lemma 2.4.2. *For* $\min_{\boldsymbol{\sigma}_{\hat{\rho}},\boldsymbol{\sigma}_\pi} \mathbf{E}_{\boldsymbol{\eta}\sim\hat{\rho}'(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}^T\mathbf{H}\boldsymbol{\eta}]+\beta\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta}))$ *with* $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}}, \boldsymbol{\sigma}_{\hat{\rho}})$ *and* $\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_\pi, \lambda\boldsymbol{\sigma}_\pi)$ *the optimal prior and posterior covariances have elements*

$$(\sigma^*_{\hat{\rho}i})^{-1} = \frac{1}{2\beta}[h_i + \sqrt{h_i^2 + \frac{4\beta h_i}{(\mu_{i\hat{\rho}} - \mu_{i\pi})^2}}], \tag{B.7}$$

$$(\sigma^*_{\pi i})^{-1} = \frac{\lambda}{2\beta}[\sqrt{h_i^2 + \frac{4\beta h_i}{(\mu_{i\hat{\rho}} - \mu_{i\pi})^2}} - h_i], \tag{B.8}$$

where $\mathbf{H} \equiv \nabla^2\hat{\mathcal{L}}^{\ell_{cat}}_{X,Y}(f_{\boldsymbol{\theta}})$ *captures the curvature at the the minimum. Then*

$$\begin{aligned}\min_{\boldsymbol{\sigma}_{\hat{\rho}},\boldsymbol{\sigma}_\pi} C_\beta(X, Y; \hat{\rho}, \pi) \geq \frac{1}{2}(&\sum_i a_i(\mu_{i\hat{\rho}} - \mu_{i\pi})^2 \\ &+ \beta\sum_i \ln(\frac{h_i + a_i}{a_i})),\end{aligned} \tag{B.9}$$

where $a_i \triangleq a_i(\beta, \mu_{i\hat{\rho}}, \mu_{i\pi}, h_i) = \frac{1}{2}[\sqrt{h_i^2 + \frac{4\beta h_i}{(\mu_{i\hat{\rho}} - \mu_{i\pi})^2}} - h_i]$.

Proof. The developed objective (B.5) is

$$C_\beta(X, Y; \hat{\rho}, \pi) = \frac{1}{2}\mathrm{tr}(\mathbf{H}\boldsymbol{\Sigma}_{\hat{\rho}}) + \frac{\beta}{2}(\mathrm{tr}(\frac{1}{\lambda}\boldsymbol{\Sigma}_\pi^{-1}\boldsymbol{\Sigma}_{\hat{\rho}}) - k + \frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_\pi)^{\mathrm{T}}\boldsymbol{\Sigma}_\pi^{-1}(\boldsymbol{\mu}_{\hat{\rho}} - \boldsymbol{\mu}_\pi) + \ln\left(\frac{\det\lambda\boldsymbol{\Sigma}_\pi}{\det\boldsymbol{\Sigma}_{\hat{\rho}}}\right)) \tag{B.10}$$

We substitute the precision matrix $\boldsymbol{\Lambda}_\pi = \boldsymbol{\Sigma}_\pi^{-1}$ and $\boldsymbol{\Sigma}_{\hat{\rho}}$ with the minimizer $\boldsymbol{\Sigma}^*_{\hat{\rho}} = \beta(\mathbf{H} +$

$\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1}$ in (B.10), we obtain

$$\begin{aligned}
C_\beta(X,Y;\hat{\rho},\pi)|_{\mathbf{\Sigma}_{\hat{\rho}}=\mathbf{\Sigma}_{\hat{\rho}}^*} =& \frac{1}{2}\mathrm{tr}(\mathbf{H}\beta(\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1}) + \frac{\beta}{2}(\mathrm{tr}(\frac{1}{\lambda}\mathbf{\Lambda}_\pi\beta(\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1}) \\
&+ \frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)^{\mathrm{T}}\mathbf{\Lambda}_\pi(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi) - k + \ln\left(\frac{\det\lambda\mathbf{\Lambda}_\pi^{-1}}{\det\beta(\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1}}\right)) \\
=& \frac{\beta}{2}\mathrm{tr}(\mathbf{H}(\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1}) + \frac{\beta^2}{2\lambda}(\mathrm{tr}(\mathbf{\Lambda}_\pi(\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1})) \\
&+ \frac{\beta}{2}(+\frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)^{\mathrm{T}}\mathbf{\Lambda}_\pi(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi) - k + \ln\left(\frac{\det\lambda\mathbf{\Lambda}_\pi^{-1}}{\det\beta(\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1}}\right)) \\
=& \frac{\beta}{2}(\mathrm{tr}((\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)(\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1}) \\
&\frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)^{\mathrm{T}}\mathbf{\Lambda}_\pi(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi) - k + \ln\left(\frac{\det\lambda\mathbf{\Lambda}_\pi^{-1}}{\det\beta(\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1}}\right)) \\
=& \frac{\beta}{2}[+\frac{1}{\lambda}(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi)^{\mathrm{T}}\mathbf{\Lambda}_\pi(\boldsymbol{\mu}_{\hat{\rho}}-\boldsymbol{\mu}_\pi) + \ln\left(\frac{\det\lambda\mathbf{\Lambda}_\pi^{-1}}{\det\beta(\mathbf{H}+\frac{\beta}{\lambda}\mathbf{\Lambda}_\pi)^{-1}}\right)].
\end{aligned} \tag{B.11}$$

Substituting $\mathbf{\Lambda}_\pi = \mathrm{diag}(\Lambda_{1\pi}, \Lambda_{2\pi}, ..., \Lambda_{k\pi})$ and $\mathbf{H} = \mathrm{diag}(h_1, h_2, ..., h_k)$ in the above expression we get

$$C_\beta(X,Y;\hat{\rho},\pi)|_{\mathbf{\Sigma}_{\hat{\rho}}=\mathbf{\Sigma}_{\hat{\rho}}^*} = \frac{\beta}{2}(\frac{1}{\lambda}\sum_i \Lambda_{i\pi}(\mu_{i\hat{\rho}}-\mu_{i\pi})^2 - \sum_i \ln(\frac{\Lambda_{i\pi}}{\lambda}) + \sum_i \ln(\frac{h_i+\frac{\beta}{\lambda}\Lambda_{i\pi}}{\beta})) \tag{B.12}$$

The above expression is easy to optimize. We see that the sole stationary point exists at

$$\Lambda_{i\pi}^* = \frac{\lambda}{2\beta}[\sqrt{h_i^2 + \frac{4\beta h_i}{(\mu_{i\hat{\rho}}-\mu_{i\pi})^2}} - h_i]. \tag{B.13}$$

We now need to calculate second derivatives so as to prove that the stationary point is a local optimum. We go back to the developed objective (B.10), and substitute $\mathbf{\Sigma}_{\hat{\rho}} = \mathrm{diag}(\boldsymbol{\sigma}_{\hat{\rho}})$ and $\mathbf{\Sigma}_\pi = \mathrm{diag}(\boldsymbol{\sigma}_\pi)$. For the diagonal approximation the objective turns

into a sum of separable functions.

$$\begin{aligned} C_\beta(X,Y;\hat{\rho},\pi) &= \sum_i \frac{h_i}{2}\sigma_{i\hat{\rho}} + \sum_i \frac{\beta}{2\lambda}\frac{\sigma_{i\hat{\rho}}}{\sigma_{i\pi}} - \sum_i \frac{\beta}{2} + \sum_i \frac{\beta(\mu_{i\hat{\rho}}-\mu_{i\pi})^2}{2\lambda}\frac{1}{\sigma_{i\pi}} \\ &+ \frac{\beta}{2}[\sum_i \ln(\lambda\sigma_{i\pi}) - \sum_i \ln(\sigma_{i\hat{\rho}})] \\ &= \sum_i A_i\sigma_{i\hat{\rho}} + \sum_i B_i\frac{\sigma_{i\hat{\rho}}}{\sigma_{i\pi}} - \sum_i \frac{\beta}{2} + \sum_i C_i\frac{1}{\sigma_{i\pi}} + D_i[\sum_i \ln(\lambda\sigma_{i\pi}) - \sum_i \ln(\sigma_{i\hat{\rho}})] \\ &= \sum_i [A_i\sigma_{i\hat{\rho}} + B_i\frac{\sigma_{i\hat{\rho}}}{\sigma_{i\pi}} - \frac{\beta}{2} + C_i\frac{1}{\sigma_{i\pi}} + D_i(\ln(\lambda\sigma_{i\pi}) - \ln(\sigma_{i\hat{\rho}}))] \end{aligned} \tag{B.14}$$

where we have set $A_i = \frac{h_i}{2}$, $B_i = \frac{\beta}{2\lambda}$, $C_i = \frac{\beta(\mu_{i\hat{\rho}}-\mu_{i\pi})^2}{2\lambda}$, $D_i = \frac{\beta}{2}$.

We take the derivatives of one of these functions with respect to $\sigma_{i\hat{\rho}}, \sigma_{i\pi}$ and drop the indices i for clarity

$$\frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial\sigma_{\hat{\rho}}} = A + \frac{B}{\sigma_\pi} - \frac{D}{\sigma_{\hat{\rho}}}, \quad \frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial\sigma_\pi} = -\frac{B\sigma_{\hat{\rho}}}{\sigma_\pi^2} - \frac{C}{\sigma_\pi^2} + \frac{D}{\sigma_\pi} \tag{B.15}$$

and

$$\frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial^2\sigma_{\hat{\rho}}} = \frac{D}{\sigma_{\hat{\rho}}^2}, \quad \frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial^2\sigma_\pi} = 2(B\sigma_{\hat{\rho}} + C)\frac{1}{\sigma_\pi^3} - \frac{D}{\sigma_\pi^2} \tag{B.16}$$

$$\frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial\sigma_{\hat{\rho}}\partial\sigma_\pi} = -\frac{B}{\sigma_\pi^2}, \quad \frac{\partial C_\beta(X,Y;\hat{\rho},\pi)}{\partial\sigma_\pi\partial\sigma_{\hat{\rho}}} = -\frac{B}{\sigma_\pi^2} \tag{B.17}$$

We need to check whether the Hessian matrix is PSD so that the stationary point we found is a local minimum and the function is convex. We do that by calculating whether all principal minors of the Hessian are positive.

$$\nabla^2 C_\beta(\sigma_{\hat{\rho}},\sigma_\pi) = \begin{bmatrix} \frac{D}{\sigma_{\hat{\rho}}^2} & -\frac{B}{\sigma_\pi^2} \\ -\frac{B}{\sigma_\pi^2} & 2(B\sigma_{\hat{\rho}} + C)\frac{1}{\sigma_\pi^3} - \frac{D}{\sigma_\pi^2} \end{bmatrix} \tag{B.18}$$

We see easily that $\det(\frac{D}{\sigma_{\hat{\rho}}^2}) > 0$. While

$$\begin{aligned}\det(\nabla^2 C_\beta(\sigma_{\hat{\rho}}, \sigma_\pi)) &= \frac{D}{\sigma_{\hat{\rho}}^2}\left(2(B\sigma_{\hat{\rho}} + C)\frac{1}{\sigma_\pi^3} - \frac{D}{\sigma_\pi^2}\right) - \frac{B^2}{\sigma_\pi^4} \\ &= \frac{1}{\sigma_{\hat{\rho}}^2\sigma_\pi^4}\left(2CD\sigma_\pi - (D\sigma_\pi - B\sigma_{\hat{\rho}})^2\right) \\ &= \left(\frac{1}{\sigma_{\hat{\rho}}^2\sigma_\pi^4}\frac{\beta^2}{2}\right)\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda}\sigma_\pi - \frac{1}{2}(\sigma_\pi - \frac{\sigma_{\hat{\rho}}}{\lambda})^2\right)\end{aligned} \tag{B.19}$$

The determinant is not always positive and the function is not convex. We now check whether the sole stationary point is always a local minimum. We start by substituting $\sigma_{\hat{\rho}}^\star = \beta(h + \frac{\beta}{\lambda}\frac{1}{\sigma_\pi})^{-1}$ in the multiplicand of (B.19) as the multiplier is positive by definition

$$\begin{aligned}&\det(\nabla^2 C_\beta(\sigma_{\hat{\rho}}^\star, \sigma_\pi)) = \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^4}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda}\sigma_\pi - \frac{1}{2}(\sigma_\pi - \frac{\beta}{\lambda}(h + \frac{\beta}{\lambda}\frac{1}{\sigma_\pi})^{-1})^2\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^4}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda}\sigma_\pi - \frac{1}{2}(\sigma_\pi - \frac{\beta}{\lambda}(\frac{\sigma_\pi\lambda}{h\lambda\sigma_\pi + \beta}))^2\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^4}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda}\sigma_\pi - \frac{\sigma_\pi^2}{2}(1 - (\frac{\beta}{h\lambda\sigma_\pi + \beta}))^2\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^3}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda} - \frac{\sigma_\pi}{2}(\frac{h\lambda\sigma_\pi}{h\lambda\sigma_\pi + \beta})^2\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^3}\frac{\beta^2}{2}\left(\frac{(\mu_{\hat{\rho}} - \mu_\pi)^2}{\lambda} - \frac{\lambda^2 h^2\sigma_\pi^3}{2(h\lambda\sigma_\pi + \beta)^2}\right) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2}\sigma_\pi^3 2\lambda(h\lambda\sigma_\pi + \beta)^2}(2(\mu_{\hat{\rho}} - \mu_\pi)^2(h\lambda\sigma_\pi + \beta)^2 - \lambda^3 h^2\sigma_\pi^3) \\ &= \frac{1}{\sigma_{\hat{\rho}}^{\star 2} 2\lambda(h\lambda\Lambda_\pi^{-1} + \beta)^2}(2\Lambda_\pi(\mu_{\hat{\rho}} - \mu_\pi)^2(h\lambda + \Lambda_\pi\beta)^2 - \lambda^3 h^2)\end{aligned} \tag{B.20}$$

Where we substituted $\sigma_\pi = \Lambda_\pi^{-1}$ as this will make the calculations easier. We now show

a useful identity for $\Lambda_\pi^\star = \frac{\lambda}{2\beta}[\sqrt{h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}} - h]$

$$
\begin{aligned}
(\Lambda_\pi^\star)^2 &= \frac{\lambda^2}{4\beta^2}\left(h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2} - 2h\sqrt{h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}} + h^2\right) \\
&= \frac{\lambda^2}{4\beta^2}\left(2h\left(h - \sqrt{h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}}\right) + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}\right) \\
&= \frac{h\lambda}{\beta}\frac{\lambda}{2\beta}\left(\left(h - \sqrt{h^2 + \frac{4\beta h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}}\right) + \frac{2\beta}{(\mu_{\hat{\rho}} - \mu_\pi)^2}\right) \\
&= \frac{h\lambda}{\beta}\left(\frac{\lambda}{(\mu_{\hat{\rho}} - \mu_\pi)^2} - \Lambda_\pi^\star\right)
\end{aligned}
\tag{B.21}
$$

We substitute $\Lambda_\pi = \Lambda_\pi^\star$ in (B.20) and again develop only the multiplicand

$$
\begin{aligned}
&\det(\nabla^2 C_\beta(\sigma_{\hat{\rho}}^\star, \sigma_\pi^\star)) = \frac{1}{\sigma_{\hat{\rho}}^{\star 2} 2\lambda(h\lambda\Lambda_\pi^{\star -1} + \beta)^2}(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h\lambda + \Lambda_\pi^\star\beta)^2 - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h\lambda + \Lambda_\pi^\star\beta)^2 - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + 2h\lambda\Lambda_\pi^\star\beta + (\Lambda_\pi^\star)^2\beta^2) - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + 2h\lambda\Lambda_\pi^\star\beta + \frac{h\lambda}{\beta}\left(\frac{\lambda}{(\mu_{\hat{\rho}} - \mu_\pi)^2} - \Lambda_\pi^\star\right)\beta^2) - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + h\lambda\Lambda_\pi^\star\beta + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) + 2(\Lambda_\pi^\star)^2(\mu_{\hat{\rho}} - \mu_\pi)^2 h\lambda\beta - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) \\
&+ 2\frac{h\lambda}{\beta}\left(\frac{\lambda}{(\mu_{\hat{\rho}} - \mu_\pi)^2} - \Lambda_\pi^\star\right)(\mu_{\hat{\rho}} - \mu_\pi)^2 h\lambda\beta - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) + 2\lambda^3 h^2 - 2h^2\lambda^2(\mu_{\hat{\rho}} - \mu_\pi)^2\Lambda_\pi^\star - \lambda^3 h^2) \\
&= A(2\Lambda_\pi^\star(\mu_{\hat{\rho}} - \mu_\pi)^2(h^2\lambda^2 + \frac{\beta\lambda^2 h}{(\mu_{\hat{\rho}} - \mu_\pi)^2}) + \lambda^3 h^2 - 2h^2\lambda^2(\mu_{\hat{\rho}} - \mu_\pi)^2\Lambda_\pi^\star) \\
&= A(2\Lambda_\pi^\star\beta\lambda^2 h + \lambda^3 h^2) \\
&> 0
\end{aligned}
\tag{B.22}
$$

where we have set $A = \frac{1}{\sigma_{\hat{\rho}}^{\star 2} 2\lambda(h\lambda(\Lambda_\pi^\star)^{-1}+\beta)^2} > 0$. We have used (B.21) in lines 4 and 7.

Indeed the stationary point is a local minimum. We now show that there are no other local minima at the boundaries of the domain. From (B.14) we see that we only need to evaluate expressions of the form $f(\sigma_{\hat{\rho}}) = \sigma_{\hat{\rho}} - \ln(\sigma_{\hat{\rho}})$ and $g(\sigma_\pi) = \frac{1}{\sigma_{\hat{\rho}}} + \ln(\sigma_{\hat{\rho}})$. By application of L'Hôpital's rule it's easy to show that

$$\begin{aligned}\lim_{\substack{\sigma_{\hat{\rho}}\to 0\\ \sigma_\pi=\text{ct}}} C_\beta(\sigma_{\hat{\rho}},\sigma_\pi) &= \lim_{\substack{\sigma_{\hat{\rho}}\to +\infty\\ \sigma_\pi=\text{ct}}} C_\beta(\sigma_{\hat{\rho}},\sigma_\pi) \\ &= \lim_{\substack{\sigma_{\hat{\rho}}=\text{ct}\\ \sigma_\pi\to 0}} C_\beta(\sigma_{\hat{\rho}},\sigma_\pi) = \lim_{\substack{\sigma_{\hat{\rho}}=\text{ct}\\ \sigma_\pi\to +\infty}} C_\beta(\sigma_{\hat{\rho}},\sigma_\pi) = +\infty\end{aligned} \tag{B.23}$$

□

B.4 Proof of Lemma 2.5.1

Preliminaries We remind that a neural network transforms it's inputs $\mathbf{a}_0 = \boldsymbol{x}$ to an output $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathbf{a}_l$ through a series of l layers, each of which consists of a bank of units/neurons. The computation performed by each layer $i \in \{1, ..., l\}$ is given as

$$\begin{aligned}\boldsymbol{s}_i &= \mathbf{W}_i \mathbf{a}_{i-1}, \\ \mathbf{a}_i &= \phi_i(\boldsymbol{s}_i).\end{aligned}$$

We also denote the vectorization of the weights as $\boldsymbol{\theta} = [\text{vec}(\mathbf{W}_0^{0,:})\text{vec}(\mathbf{W}_0^{1,:}) \cdots \text{vec}(\mathbf{W}_0^{r,:})]$, where $\text{vec}(\mathbf{W}_i^{j,:})$ are the weights corresponding to layer i and neuron j. We assume trained vectorized weights $\boldsymbol{\mu}_{\hat{\rho}i}$ and trained weights in matrix form $\mathbf{W}_{\hat{\rho}i}$ for layer i. We will be adding bounded perturbations to the weights of each layer i so that $||\mathbf{W}_i - \mathbf{W}_{\hat{\rho}i}||_F \leq C$. We will want to quantify the effect of these perturbations on the latent representations of the network.

We then define $\mathbf{A}_i = [\mathbf{a}_i^0, \cdots, \mathbf{a}_i^n]$, where $\mathbf{a}_i^j$ is the unperturbed latent representation of sample j at layer i, where $\mathbf{A}_i$ is produced by the operation $\mathbf{A}_i = \text{rect}(\mathbf{W}_{\hat{\rho}i}\mathbf{A}_{i-1})$. We perturb only layer i and define $\hat{\mathbf{A}}_i$, as the representations resulting from the new perturbed matrix $\mathbf{W}_i$, $\hat{\mathbf{A}}_i = \text{rect}(\mathbf{W}_i\mathbf{A}_{i-1})$. We then define $\tilde{\mathbf{A}}_i$ as the representations at layer i with accumulated error from layers $\leq i$. Similarly we can define the same quantities for the pre-activations $\boldsymbol{s}_i^j$, we denote the corresponding matrices as $\hat{\boldsymbol{S}}_i$ and $\tilde{\boldsymbol{S}}_i$.

We can then define the layerwise mean square error from perturbing only layer i

$$\hat{e}_i^2 = (1/n)||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2,$$
$$\hat{E}_i^2 = (1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2,$$

as well as the accumulated mean square error

$$\tilde{e}_i^2 = (1/n)||\mathbf{A}_i - \tilde{\mathbf{A}}_i||_F^2,$$
$$\tilde{E}_i^2 = (1/n)||\boldsymbol{S}_i - \tilde{\boldsymbol{S}}_i||_F^2,$$

where the true representations are considered as constants. We make a simplifying assumption, assuming that the mean square error of our *trained* classifier is 0. In this case we can set $\hat{\mathcal{L}}_{X,Y}^{\ell_{\text{mse}}}(f_{\boldsymbol{\theta}}) \equiv \tilde{e}_l^2 = (1/n)||\mathbf{A}_l - \tilde{\mathbf{A}}_l||_F^2$, as $\mathbf{A}_l$ now correspond to the ground truth vectors. We can easily extend to the non-zero error case using the triangle inequality.

These errors are difficult to analyze theoretically. As such we will make the useful assumption that they are well approximated by a *quadratic*, which will make the analysis tractable. This assumption is quite strong and we do not claim that the approximation is tight. Furthermore Figure 3 of the main text does not directly apply in this setting; we will be dealing with the mean-square error instead of the categorical cross-entropy and we will be analyzing layerwise errors instead of the error at the output. At the same time our aim is only to derive a useful *surrogate* objective. The empirical results in Section 5 provide evidence that the surrogate we propose is indeed useful in providing tighter bounds.

Useful Lemmata We prove the following Lemma which will be useful later. We first show that the mean square error at the output of a deep neural network can be decomposed as a sum of mean square errors for intermediate representations.

Lemma B.4.1. *Assuming layerwise perturbations that are bounded by a constant* $||\mathbf{W}_i - \mathbf{W}_{\hat{\rho}i}||_F \leq C$*, the accumulated mean square error* $\tilde{e}_l^2$ *at layer* l *can be bounded as*

$$(1/n)||\mathbf{A}_l - \tilde{\mathbf{A}}_l||_F^2 \leq \sum_{i=0}^{l} c_i (1/n)||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2 + \mathcal{O}(c^l) \tag{B.24}$$

where $\forall i < l,\ c_i = \prod_{k=i+1}^{l} ||\mathbf{W}_k||_F^2$*,* $c_l = 1$ *and* c *is some constant.*

Proof. We denote $\hat{a}_{i+1}$ a single element of $\hat{\mathbf{a}}_{i+1}$ and $\mathbf{w}_i^{\mathrm{T}}$ the corresponding row of $\mathbf{W}_i$ where we drop the indices for individual samples and neurons for clarity. One can easily

see through the properties of the rectifier function that

$$\begin{aligned}\hat{a}_{i+1} &= \text{rect}(\mathbf{w}_{i+1}^{\mathrm{T}}\tilde{\mathbf{a}}_i + \mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)) \\ &\leq \tilde{a}_{i+1} + \text{rect}(\mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)) \\ &\leq \tilde{a}_{i+1} + |\mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)|\end{aligned} \tag{B.25}$$

Similarly we can obtain $\tilde{a}_{i+1} \leq \hat{a}_{i+1} + |\mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)|$ and therefore we can write

$$|\tilde{a}_{i+1} - \hat{a}_{i+1}| \leq |\mathbf{w}_{i+1}^{\mathrm{T}}(\mathbf{a}_i - \tilde{\mathbf{a}}_i)|.$$

In matrix notation this becomes

$$||\tilde{\mathbf{A}}_{i+1} - \hat{\mathbf{A}}_{i+1}||_F \leq ||\mathbf{W}_{i+1}(\mathbf{A}_i - \tilde{\mathbf{A}}_i)||_F \leq ||\mathbf{W}_{i+1}||_F||\tilde{\mathbf{A}}_i - \mathbf{A}_i||_F$$

By the triangle inequality we can then write

$$\begin{aligned}\tilde{e}_{i+1} = (1/\sqrt{n})||\tilde{\mathbf{A}}_{i+1} - \mathbf{A}_{i+1}||_F &\leq (1/\sqrt{n})||\tilde{\mathbf{A}}_{i+1} - \hat{\mathbf{A}}_{i+1}||_F + (1/\sqrt{n})||\hat{\mathbf{A}}_{i+1} - \mathbf{A}_{i+1}||_F \\ &\leq (1/\sqrt{n})||\mathbf{W}_{i+1}||_F||\tilde{\mathbf{A}}_i - \mathbf{A}_i||_F + (1/\sqrt{n})||\hat{\mathbf{A}}_{i+1} - \mathbf{A}_{i+1}||_F \\ &\leq \sum_{t=0}^{i}(\prod_{k=t+1}^{i+1}||\mathbf{W}_k||_F||\hat{\mathbf{A}}_t - \mathbf{A}_t||_F) + (1/\sqrt{n})||\hat{\mathbf{A}}_{i+1} - \mathbf{A}_{i+1}||_F \\ &= \sum_{t=0}^{i}(\prod_{k=t+1}^{i+1}||\mathbf{W}_k||_F\hat{e}_t) + \hat{e}_{i+1}\end{aligned} \tag{B.26}$$

If $||\mathbf{W}_i - \mathbf{W}_{\hat{\rho}i}||_F \leq C$, then the errors $\hat{e}_t = ||\mathbf{A}_t - \hat{\mathbf{A}}_t||_F$ and also all terms $\prod_{k=t+1}^{i+1}||\mathbf{W}_k||_F\hat{e}_t$ are bounded. We raise both sides to the power of 2. We get the desired terms as well as terms of the form $\prod_{k=a+1}^{i+1}||\mathbf{W}_k||_F\prod_{k=b+1}^{i+1}||\mathbf{W}_k||_F\hat{e}_a\hat{e}_b$ assuming that $||\mathbf{W}_i||_F \leq \sqrt{c}$ we see that these are of the order $\mathcal{O}((\sqrt{c})^{2l}) = \mathcal{O}(c^l)$ and we get the desired result. □

In the following it will be useful to deal with the preactivations $\boldsymbol{s}_i^j$ instead of the representations $\mathbf{a}_i^j$ so as to avoid taking derivatives of the rectifier non-linearity. We will then find useful the following simple Lemma.

Lemma B.4.2. *Given the true preactivations $\boldsymbol{S}_i$ and representations $\mathbf{A}_i$, as well as the perturbed $\hat{\boldsymbol{S}}_i$ and $\hat{\mathbf{A}}_i$ for layer i the following holds*

$$(1/n)||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2 \leq (1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2. \tag{B.27}$$

Proof. We assume $(\text{rect}(x) - \text{rect}(y))^2 \leq (x-y)^2$ and check that it holds for different signs of x, y. □

We will now approximate the precativation error for each layer using a second order

Taylor expansion. We prove the following.

Lemma B.4.3. *We apply a Taylor expansion of the layerwise preactivation error $\hat{E}_i^2(\boldsymbol{\theta})$ of layer i, around a point $\boldsymbol{\mu}$. Given j neurons and n training samples, $\hat{E}_i^2(\boldsymbol{\theta})$ can be approximated as*

$$\hat{E}_i^2(\boldsymbol{\theta}) = (1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2 = \sum_j (\boldsymbol{\theta}_{ij} - \boldsymbol{\mu}_{ij})^T \mathbf{H}_i (\boldsymbol{\theta}_{ij} - \boldsymbol{\mu}_{ij}) + \mathcal{O}(||\boldsymbol{\theta}_i - \boldsymbol{\mu}_i||^3). \quad \text{(B.28)}$$

where $\mathbf{H}_i = (1/n)\sum_{k=0}^n \mathbf{a}_{i-1}^k {\mathbf{a}_{i-1}^k}^T$.

Proof. It will be easier to work with the vectorized weights per neuron $\boldsymbol{\theta}_{ij}$ directly. We note that the unperturbed representations $\boldsymbol{S}_i$ are considered as constants, and get

$$\begin{aligned}
\frac{\partial \hat{E}_i^2}{\partial \boldsymbol{\theta}_{ij}} &= \frac{\partial}{\partial \boldsymbol{\theta}_{ij}} (1/n)||\hat{\boldsymbol{S}}_i - \boldsymbol{S}_i||_F^2 \\
&= \frac{\partial}{\partial \boldsymbol{\theta}_{ij}} (1/n)||\mathbf{W}_i \mathbf{A}_{i-1} - \boldsymbol{S}_i||_F^2 \\
&= \frac{\partial}{\partial \boldsymbol{\theta}_{ij}} (1/n) \sum_{k=0}^n ||\mathbf{W}_i \mathbf{a}_{i-1}^k - \boldsymbol{s}_i^k||_2^2 \\
&= \frac{\partial}{\partial \boldsymbol{\theta}_{ij}} (1/n) \sum_{k=0}^n \sum_{t=0}^r ||\boldsymbol{\theta}_{it}^T \mathbf{a}_{i-1}^k - s_{it}^k||_2^2 \\
&= \frac{1}{n} \sum_{k=0}^n \sum_{t=0}^r \frac{\partial}{\partial \boldsymbol{\theta}_{ij}} ||\boldsymbol{\theta}_{it}^T \mathbf{a}_{i-1}^k - s_{it}^k||_2^2 = \frac{2}{n} \sum_{k=0}^n (\boldsymbol{\theta}_{ij}^T \mathbf{a}_{i-1}^k - s_{ij}^k) {\mathbf{a}_{i-1}^k}^T
\end{aligned} \quad \text{(B.29)}$$

where in the third line we expand with respect to the samples and in the fourth line we expand with respect to each neuron. Then we can calculate the second order derivatives.

$$\frac{\partial^2 \hat{E}_i^2}{\partial^2 \boldsymbol{\theta}_{ij}} = \frac{\partial}{\partial \boldsymbol{\theta}_{ij}} \frac{2}{n} \sum_{k=0}^n (\boldsymbol{\theta}_{ij}^T \mathbf{a}_{i-1}^k - s_{ij}^k) {\mathbf{a}_{i-1}^k}^T = \frac{2}{n} \sum_{k=0}^n \mathbf{a}_{i-1}^k {\mathbf{a}_{i-1}^k}^T. \quad \text{(B.30)}$$

From the above, it is clear that the Hessian is block diagonal, with identical blocks for each neuron j. We can the approximate the layerwise error $\hat{e}_i^2$ using a second order Taylor expansion around a point $\boldsymbol{\mu}$ as

$$\begin{aligned}
\hat{E}_i^2 &= \frac{\partial \hat{E}_i^2}{\partial \boldsymbol{\theta}_i} (\boldsymbol{\theta}_i - \boldsymbol{\mu}_i)^T + \frac{1}{2} (\boldsymbol{\theta}_i - \boldsymbol{\mu}_i)^T \frac{\partial^2 \hat{E}_i^2}{\partial^2 \boldsymbol{\theta}_i} (\boldsymbol{\theta}_i - \boldsymbol{\mu}_i) + \mathcal{O}(||\boldsymbol{\theta}_i - \boldsymbol{\mu}_i||^3) \\
&= \sum_j [(\boldsymbol{\theta}_{ij} - \boldsymbol{\mu}_{ij})^T \sum_{k=0}^n \frac{1}{n} \mathbf{a}_{i-1}^k {\mathbf{a}_{i-1}^k}^T (\boldsymbol{\theta}_{ij} - \boldsymbol{\mu}_{ij})] + \mathcal{O}(||\boldsymbol{\theta}_i - \boldsymbol{\mu}_i||^3)
\end{aligned} \quad \text{(B.31)}$$

where we assume that the derivatives with respect to the layer weights of order other than two are negligible. This is a strong but useful assumption to make, and one that will make the analysis tractable. □

We are now ready to prove our main lemma.

Lemma 2.5.1. *The differentiable surrogate objective*

$$\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}\hat{\mathcal{L}}^{\ell_{mse}}_{X,Y}(f_{\boldsymbol{\theta}}) + \frac{1}{\beta n}(\mathrm{KL}(\hat{\rho}(\boldsymbol{\theta})||\mathcal{N}(\boldsymbol{\mu}_\pi, \lambda\mathbf{I})) + \ln\frac{1}{\delta}) \tag{B.32}$$

, assuming that the layerwise derivatives of order other than 2 are negligible, has the following upper bound

$$\begin{aligned}&\sum_{i,j}[\mathbf{E}_{\boldsymbol{\eta}_{ij}\sim\hat{\rho}'_{ij}(\boldsymbol{\theta})}[\frac{1}{2}\boldsymbol{\eta}_{ij}^T\mathbf{H}_i\boldsymbol{\eta}_{ij}] + \frac{1}{\beta n}\mathrm{KL}(\hat{\rho}_{ij}(\boldsymbol{\theta})||\pi_{ij}(\boldsymbol{\theta})]\\ &+\mathcal{O}(c^l)\end{aligned} \tag{B.33}$$

where $\hat{\rho}_{ij}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\hat{\rho}ij}, \boldsymbol{\Sigma}_{\hat{\rho}ij})$, $\pi_{ij}(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}_{\pi ij}, \lambda\mathbf{I})$, $\mathbf{H}_i = (2/n)\sum_{k=0}^n \mathbf{a}_{i-1}^k {\mathbf{a}_{i-1}^k}^T$, *are neuronwise posteriors, priors and Hessians.*

Proof. We assume that the prior $\pi(\boldsymbol{\theta})$ and posterior $\hat{\rho}(\boldsymbol{\theta})$ are block diagonal, with blocks corresponding to weights in each neuron.

$$\begin{aligned}\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[\hat{\mathcal{L}}^{\ell_{\mathrm{mse}}}_{X,Y}(f_{\boldsymbol{\theta}})] &\leq \mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[\sum_{i=0}^{l} c_i(1/n)||\mathbf{A}_i - \hat{\mathbf{A}}_i||_F^2 + \mathcal{O}(c^l)]\\ &\leq \mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[\sum_{i=0}^{l} c_i(1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2 + \mathcal{O}(c^l)]\\ &= \sum_{i=0}^{l}\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[c_i]\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[(1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2] + \mathcal{O}(c^l)]\\ &\leq \sum_{i=0}^{l} c^*\mathbf{E}_{\boldsymbol{\theta}\sim\hat{\rho}(\boldsymbol{\theta})}[(1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2] + \mathcal{O}(c^l)\\ &= \sum_{i=0}^{l} c^*\mathbf{E}_{\boldsymbol{\eta}_{ij}\sim\hat{\rho}'_{ij}(\boldsymbol{\theta})}[\sum_j \boldsymbol{\eta}_{ij}^T\mathbf{H}_i\boldsymbol{\eta}_{ij}] + \mathcal{O}(c^l)\\ &= \sum_{i,j} c^*\mathbf{E}_{\boldsymbol{\eta}_{ij}\sim\hat{\rho}'_{ij}(\boldsymbol{\theta})}[\boldsymbol{\eta}_{ij}^T\mathbf{H}_i\boldsymbol{\eta}_{ij}] + \mathcal{O}(c^l).\end{aligned} \tag{B.34}$$

In line 3 we used the fact that the constant c_i for layer i depends only on layers $k \geq i+1$, thus the two random variables are independent and the expectation operator is multiplicative. In line 4 we assume that the terms $c_i = \prod_{k=i+1}^{l}||\mathbf{W}_k||_F^2$ are upper bounded by the constant c^*. This is reasonable as in practice we will be adding Gaussian noise with bounded variance to the layer weights. In line 5 we approximate the error $\hat{E}_i^2 = (1/n)||\boldsymbol{S}_i - \hat{\boldsymbol{S}}_i||_F^2$ using (B.28) at point $\mu_{\hat{\rho}}$ which is the mean of the posterior $\hat{\rho}(\boldsymbol{\theta})$, then we use that $\hat{\rho}'(\boldsymbol{\theta})$ is a centered version of $\hat{\rho}(\boldsymbol{\theta})$. We finally assume that the term $\mathcal{O}(c^l)$ dominates the remainders from the Taylor expansion.

We then absorb the constant c^* in the hyperparameter β. By noting that the KL divergence of block-diagonal Gaussians can be decomposed as $\mathrm{KL}(\mathcal{N}(\hat{\rho}(\boldsymbol{\theta})||\pi(\boldsymbol{\theta})) = \sum_{ij} \mathrm{KL}(\mathcal{N}(\hat{\rho}_{ij}(\boldsymbol{\theta})||\pi_{ij}(\boldsymbol{\theta}))$ we get the desired result. □

Importantly we don't require that the deep neural network was trained using the mean square error. Rather we can optimize (B.33) for any network and assume that it's representations remain close based on the mean square error. Our experiments however show that optimizing (B.33) is also a good surrogate for keeping the 01-error small.

B.5 Experimental Setup

Experiments for Variational Inference were performed on NVIDIA Tesla K40c GPU. All other experiments were performed on an NVIDIA GEFORCE GTX 1080 GPU. The libraries used were Tensoflow 1.15.0 (Abadi et al., 2015), Keras 2.2.4 (Chollet et al., 2015) and Tensorflow-Probability 0.8.0 (Dillon et al., 2017).

When training the original deterministic classifiers, for the MNIST architectures we used the Keras implementation SGD with a learning rate of 0.01, momentum value of 0.9 and exponential decay with decay factor 0.001. For CIFAR architectures we used the Keras implementation of Adam with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, decay value of 0.00005 and the default value for the epsilon parameter. We used the softmax activation as well as the categorical cross-entropy in both cases. MNIST architectures were trained for 10 epochs while CIFAR architectures where trained for 200 epochs, which was sufficient for the training loss to stop decreasing.

When optimizing the posterior distributions centered at the deterministic classifier we used a grid search over β and/or λ where appropriate, with limits specified in the following tables. The computational time reported refers to the total time required to compute the plots in the main text for each setup, including computing the posterior and/or prior distributions as well as sampling m number of samples for estimating the expected empirical risk of the stochastic classifier.

MNIST. We report the following values for the MNIST experiments.

Experiment	β	λ	Time
MNIST Is@0	-	[0.031,0.3]	14h
MNIST Is@Init	-	[0.031,0.3]	14h
MNIST VI	[1,5]	[0.03,0.1]	11h
MNIST Post	[0.001,0.07]	[0.00005,0.01]	33h
MNIST Post+Prior	[0.000007,0.001]	-	10h
MNIST sK-FAC	[0.001,0.02]	[0.001,0.1]	33h

The β and λ ranges are identical for MNIST10, MNIST5, MNIST2 while computation times are of the same order of magnitude.

CIFAR. We report the following values for the CIFAR experiments.

Experiment	β	λ	Time
CIFAR Is@0	-	[0.031,0.3]	15h
CIFAR Is@Init	-	[0.031,0.3]	15h
CIFAR VI	[1,2]	[0.1,0.3]	10h
CIFAR Post	[0.001,0.1]	[0.001,0.1]	32h
CIFAR Post+Prior	[0.0001,0.001]	-	11h
CIFAR sK-FAC	-	-	-

The β and λ ranges are identical for CIFAR10, CIFAR5, CIFAR2 while computation times are of the same order of magnitude.

For the Variational Inference experiments we used the Adam (Kingma and Ba, 2014) optimizer with a learning rate of $1e-1$ for 5 epochs of training. For efficient inference we used the Tensorflow-Probability (Dillon et al., 2017) implementation of the Flipout (Wen et al., 2018) estimator.

B.6 Notes on PAC-Bayes

We note here some important differences between the PAC-Bayesian setting and the standard Bayesian treatment of deep neural networks, as there are some important overlaps in the terms used.

First, while PAC-Bayes refers to a "posterior" $\hat{\rho}$ this distribution is not required to be a posterior in the Bayesian sense. On the contrary it can be chosen to be *any* distribution. As such we are free to model $\hat{\rho}$ using different distributions centered on the deterministic

neural networks, decoupled from how we trained the original deterministic network. In particular in Section 5 we can minimize the mean square error surrogate from Lemma 5.1. even though the deterministic networks are trained using the categorical cross-entropy loss.

Second, as noted in the main text the prior π in PAC-Bayes has to be independent of the training set but can depend on the data distribution.

C Appendix

C.1 Proof of Theorem 3.2.3

Theorem 3.2.3. *Given a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, a hypothesis set $\mathcal{F}$, a loss function $\ell' : \mathcal{F} \times \mathcal{X} \times \mathcal{Y} \to [0,1]$, $\mathscr{P} : (\mathcal{X} \times \mathcal{Y})^n \to \mathscr{M}(\mathbb{R}^p)$ an ϵ-differentially private mechanism for choosing a data-dependent prior over $\mathcal{F}$, a real number $\delta \in (0,1]$, and a real number $\beta > 0$, with probability at least $1-\delta$ over the choice of $(X,Y) \sim \mathcal{D}^n$, we have*

$$\forall \hat{\rho} \in \mathscr{M}(\mathbb{R}^p) \text{ on } \mathcal{F} : \mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell'}_{\mathcal{D}}(f) \leq \Phi^{-1}_{\beta}(\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}^{\ell'}_{X,Y}(f) + \frac{1}{\beta}(\frac{\mathrm{KL}(\hat{\rho}||\mathscr{P}(X,Y)) + \ln\frac{2}{\delta}}{n} + \frac{\epsilon^2}{2} + \epsilon\sqrt{\frac{\ln(4/\delta)}{2n}})) \tag{C.1}$$

where $\Phi^{-1}_{\beta}(x) = \frac{1-e^{-\beta x}}{1-e^{-\beta}}$.

Proof. The proof is identical to the one found in Dziugaite and Roy (2018a), but we include all details for completeness. We first present some useful definitions of concepts related to differential privacy

Definition C.1.1. Let $\beta \geq 0$, let X and Y be random variables in arbitrary measurable spaces and, let X' be independent of Y and equal in distribution to X. The *β-approximate max-information* between X and Y, denoted $I^{\beta}_{\infty}(X;Y)$, is the least value k such that, for all product measurable events E,

$$\mathbb{P}\{(X,Y) \in E\} \leq e^k \mathbb{P}\{(X',Y) \in E\} + \beta. \tag{C.2}$$

Similarly the *max-information* $I_{\infty}(X;Y)$ is defined to be $I^{\beta}_{\infty}(X;Y)$ for $\beta = 0$.

Given an ϵ-differentially private mechanism $\mathscr{P} : \mathcal{Z}^n \to \mathscr{M}(\mathbb{R}^p)$ one can also define the β-approximate max-information $I^{\beta}_{\infty}(Z;\mathscr{P}(Z))$ between the mechanism output $\mathscr{P}(Z)$

and the dataset Z. The following theorem shows how ϵ-differential privacy controls the β-approximate max-information $I_\infty^\beta(Z;\mathscr{P}(Z))$ between these two random variables.

Theorem C.1.1. *Let $\mathscr{P}$ be an ϵ-differentially private algorithm. For a distribution $\mathcal{D}$ over $\mathcal{Z}$, let Z be a random variable $Z \sim \mathcal{D}^n$. then for any $\beta > 0$*

$$I_\infty^\beta(Z;\mathscr{P}(Z)) \leq \frac{n\epsilon^2}{2} + \epsilon\sqrt{\frac{n\ln(2/\beta)}{2}} \tag{C.3}$$

For every distribution $q \in \mathscr{M}(\mathbb{R}^p)$, let

$$R(q) = \{Z \in \mathcal{Z}^n : (\exists\hat{\rho} \text{ on } \mathcal{F})\mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}_\mathcal{D}^{\ell'}(f) \geq \Phi_\beta^{-1}(\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}_{X,Y}^{\ell'}(f) + \frac{1}{\beta n}(\mathrm{KL}(\hat{\rho}||q) + \ln\frac{1}{\delta'}))\}. \tag{C.4}$$

It follows from the Catoni bound 3.20 that $\mathbb{P}_{Z\sim\mathcal{D}^n}\{Z \in R(q)\} \leq \delta'$. Let $\beta > 0$. Then, by the definition of approximate max-information, we have

$$\begin{aligned}\mathbb{P}_{Z\sim\mathcal{D}^n}\{Z \in R(\mathscr{P}(Z))\} &\leq e^{I_\infty^\beta(Z;\mathscr{P}(Z))}\mathbb{P}_{(Z,Z')\sim\mathcal{D}^{2n}}\{Z \in R(\mathscr{P}(Z'))\} + \beta \\ &\leq e^{I_\infty^\beta(Z;\mathscr{P}(Z))}\delta' + \beta \triangleq \delta.\end{aligned} \tag{C.5}$$

We have $\delta' = e^{-I_\infty^\beta(Z;\mathscr{P}(Z))}(\delta - \beta)$. Therefore with probability no more than δ over $Z \sim \mathcal{D}^n$,

$$\exists\hat{\rho} \text{ on } \mathcal{F}, \quad \mathbf{E}_{f\sim\hat{\rho}}\mathcal{L}_\mathcal{D}^{\ell'}(f) \geq \Phi_\beta^{-1}(\mathbf{E}_{f\sim\hat{\rho}}\hat{\mathcal{L}}_{X,Y}^{\ell'}(f) + \frac{1}{\beta n}(\mathrm{KL}(\hat{\rho}||\mathscr{P}(Z)) + \ln\frac{2\sqrt{n}}{\delta-\beta} + I_\infty^\beta(Z;\mathscr{P}(Z)))). \tag{C.6}$$

The results follows by replacing the approximate max-information $I_\infty^\beta(Z;\mathscr{P}(Z))$ with the bound provided by Theorem C.1.1, and setting $\beta = \delta/2$ (β is a free parameter).

□

C.2 Proof of Lemma 3.3.1

Lemma 3.3.1. *Assume a K-class classification problem where $p(\boldsymbol{x}|\mathcal{C}_k) = \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, then by applying Bayes' theorem, the maximum likelihood solution to the generative classifier is*

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{\exp(a_k(\boldsymbol{x}))}{\sum_j \exp(a_j(\boldsymbol{x}))}, \quad a_k(\boldsymbol{x}) = \mathbf{w}_k^\top\boldsymbol{x} + w_{k0} \tag{C.7}$$

where we have defined

$$\mathbf{w}_k = \tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}}_k, \quad w_{k0} = -\frac{1}{2}\tilde{\boldsymbol{\mu}}_k^\top\tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}}_k + \ln\tilde{p}(\mathcal{C}_k) \tag{C.8}$$

and also we have $\tilde{p}(\mathcal{C}_k) = \frac{n_k}{\sum_i n_i}$, $\tilde{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i\in\mathcal{C}_1} \boldsymbol{x}_i$, $\tilde{\boldsymbol{\Sigma}}_k = \frac{1}{n_k}\sum_{i\in\mathcal{C}_k}(\boldsymbol{x}_i - \tilde{\boldsymbol{\mu}}_k)(\boldsymbol{x}_i - \tilde{\boldsymbol{\mu}}_k)^\top$ *and* $\tilde{\boldsymbol{\Sigma}} = \sum_i \tilde{p}(\mathcal{C}_i)\tilde{\boldsymbol{\Sigma}}_i$.

Proof. Consider the case of two classes, each having a Gaussian class-conditional density with a shared covariance matrix, and suppose we have a data set $\{\boldsymbol{x}_n, t_n\}$ where $n = 1, ..., N$. Here $t_n = 1$ denotes class $\mathcal{C}_1$ and $t_n = 0$ denotes class $\mathcal{C}_2$. We denote the prior class probability $p(\mathcal{C}_1) = \beta$, so that $p(\mathcal{C}_2) = 1 - \beta$. For a data point $\boldsymbol{x}_n$ from class $\mathcal{C}_1$, we have $t_n = 1$ and hence

$$p(\boldsymbol{x}_n, \mathcal{C}_1) = p(\boldsymbol{x}_n|\mathcal{C}_1)p(\mathcal{C}_1) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})\beta \tag{C.9}$$

Similarly for class $\mathcal{C}_2$, we have $t_n = 0$ and hence

$$p(\boldsymbol{x}_n, \mathcal{C}_2) = p(\boldsymbol{x}_n|\mathcal{C}_2)p(\mathcal{C}_2) = \mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})(1 - \beta) \tag{C.10}$$

Thus the likelihood function is given by

$$p(\boldsymbol{t}, \mathbf{X}|\beta, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}) = \prod_{n=1}^{N} [\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})\beta]^{t_n} [\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})(1 - \beta)]^{1-t_n}. \tag{C.11}$$

We take the derivatives of the above with respect to $\beta, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}$ and set them to zero. We thus obtain the following maximum likelihood estimates

$$\tilde{\beta} = \frac{N_1}{N_1 + N_2} \tag{C.12}$$

$$\tilde{\boldsymbol{\mu}}_1 = \frac{1}{N_1}\sum_{n\in\mathcal{C}_1} \boldsymbol{x}_n \tag{C.13}$$

$$\tilde{\boldsymbol{\mu}}_2 = \frac{1}{N_2}\sum_{n\in\mathcal{C}_2} \boldsymbol{x}_n. \tag{C.14}$$

For the empirical covariance $\tilde{\boldsymbol{\Sigma}}$ we get

$$\tilde{\boldsymbol{\Sigma}} = \frac{N_1}{N}\tilde{\boldsymbol{S}}_1 + \frac{N_2}{N}\tilde{\boldsymbol{S}}_2 \tag{C.15}$$

where

$$\tilde{\boldsymbol{S}}_1 = \frac{1}{N_1}\sum_{n\in\mathcal{C}_1} (\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}}_1)(\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}}_1)^\top \tag{C.16}$$

$$\tilde{\boldsymbol{S}}_2 = \frac{1}{N_2}\sum_{n\in\mathcal{C}_2} (\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}}_2)(\boldsymbol{x}_n - \tilde{\boldsymbol{\mu}}_2)^\top. \tag{C.17}$$

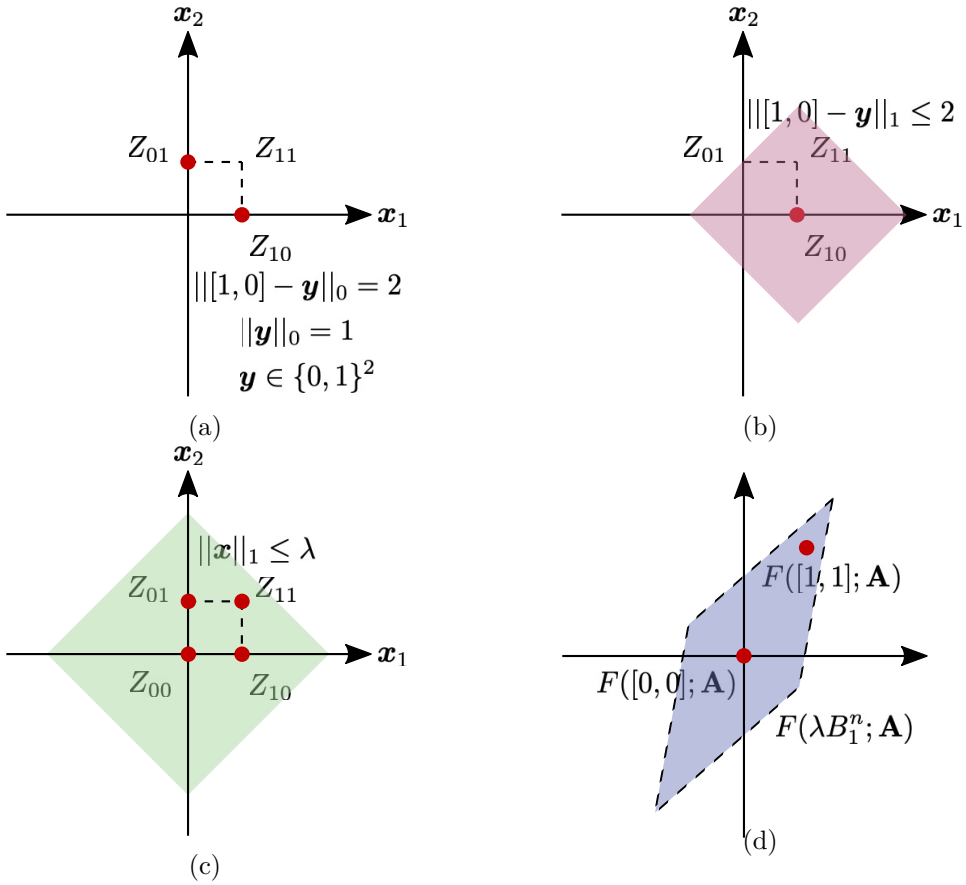


Figure C.1 – **ϵ-generalized privacy and overview of the proof.**

We can then get the result by noting that for the Gaussian model that we consider

$$p(\boldsymbol{x}|\mathcal{C}_k) = \frac{1}{(2\pi)^{D/2}} \frac{1}{|\boldsymbol{\Sigma}|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right\} \tag{C.18}$$

and that

$$p(\mathcal{C}_k|\boldsymbol{x}) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \tag{C.19}$$

where we defined a_k as

$$a_k = \ln(p(\boldsymbol{x}|\mathcal{C}_k)p(\mathcal{C}_k)). \tag{C.20}$$

□

C.3 Relating ϵ-generalized privacy and ϵ-differential privacy and proof outline

We first state the definition of ϵ-generalized privacy for dimensions $\mathbb{R}^{n+1}$

Definition C.3.1. A randomized algorithm $\mathscr{P} : \mathcal{Z}^{n+1} \to T$ is ϵ-generalized private if, for a dataset $Z \in \mathcal{Z}^{n+1}$ and vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{n+1}$ such that $||\boldsymbol{x} - \boldsymbol{y}||_1 \leq k$, and all measurable subsets $B \subseteq T$, we have $\mathbb{P}\{\mathscr{P}(Z\text{Diag}(\boldsymbol{x})) \in B\} \leq e^{\epsilon k}\mathbb{P}\{\mathscr{P}(Z\text{Diag}(\boldsymbol{y})) \in B\}$.

Recall the definition of ϵ-differential privacy

Definition C.3.2. A randomized algorithm $\mathscr{P} : \mathcal{Z}^n \to T$ is (ϵ, δ)-differentially private if, for all datasets $Z, Z' \in \mathcal{Z}^n$ that differ at only one sample, and all measurable subsets $B \subseteq T$, we have $\mathbb{P}\{\mathscr{P}(Z) \in B\} \leq e^{\epsilon}\mathbb{P}\{\mathscr{P}(Z') \in B\} + \delta$.

one can reformulate this definition as

Definition C.3.3. A randomized algorithm $\mathscr{P} : \mathcal{Z}^{n+1} \to T$ is (ϵ, δ)-differentially private if, for a dataset $Z \in \mathcal{Z}^{n+1}$ and all measurable subsets $B \subseteq T$, we have $\mathbb{P}\{\mathscr{P}(\{Z\text{Diag}(\boldsymbol{x})\}/\{\boldsymbol{0}\}) \in B\} \leq e^{\epsilon}\mathbb{P}\{\mathscr{P}(\{Z\text{Diag}(\boldsymbol{y})\}/\{\boldsymbol{0}\}) \in B\} + \delta$, with $||\boldsymbol{x} - \boldsymbol{y}||_0 = 2$, $||\boldsymbol{x}||_0 = ||\boldsymbol{y}||_0 = n$ and $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^{n+1}$.

and we can see that ϵ-generalized privacy is a relaxation of ϵ-differential privacy.

We will see now how measurable events can be related between both cases. We are interested in events that are based on averages of the datasets. As an example we define the measurable event B given by $B(Z) = \{\theta : ||\theta - (1/n)\sum_{i,i\neq k_1} \boldsymbol{z}_i|| \leq c\}$ where k_1 is the indice of one element of Z that we remove. Alternatively we can write this event as $B(Z;\boldsymbol{x}) = \{\theta : ||\theta - Z\text{Diag}(\boldsymbol{x})\boldsymbol{m}|| \leq c,\ ||\boldsymbol{x}||_0 = n, \boldsymbol{x} \in \{0,1\}^{n+1}\}$ where $\boldsymbol{m} = [1/n, \ldots, 1/n]^\top \in \mathbb{R}^{(n+1)\times 1}$. In this way a pair of events $B(Z), B(Z')$ can be uniquely defined as $B(Z;\boldsymbol{x}), B(Z;\boldsymbol{y})$ with $||\boldsymbol{x} - \boldsymbol{y}||_0 = 2$, $||\boldsymbol{x}||_0 = ||\boldsymbol{y}||_0 = n$ and $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^{n+1}$.

One can rewrite a generalization of these events based on ϵ-generalized privacy as $B(Z;\boldsymbol{x}) = \{\theta : ||\theta - Z\text{Diag}(\boldsymbol{x})\boldsymbol{m}|| \leq c,\ \boldsymbol{x} \in \mathbb{R}^{n+1}\}$ where $\boldsymbol{m} = [1/(n+1), \ldots, 1/(n+1)]^\top \in \mathbb{R}^{(n+1)\times 1}$, so that a pair of events $B(Z), B(Z')$ can be defined as $B(Z;\boldsymbol{x}), B(Z;\boldsymbol{y})$ such that $||\boldsymbol{x} - \boldsymbol{y}||_1 \leq k = 1$.

Relating ϵ-differential privacy and generalized privacy though a visualization We visualize the relationship between ϵ-differential privacy and ϵ-generalized privacy in Figures C.1a,C.1b. We assume a dataset with two samples $Z = [\boldsymbol{z}_1, \boldsymbol{z}_2]$ and vector $\boldsymbol{x} = [x_1, x_2]^\top$ then $Z_{x_1x_2} = Z\text{Diag}([x_1, x_2])$. Let an 1-differentially private algorithm and an 1/2-generalized private algorithm a dataset Z_{10} and Z_{per} a perturbed dataset. For

both the 1-differentially private algorithm and the 1/2-generalized private algorithm one has $\mathbb{P}\{\mathscr{P}(Z_{10}) \in B\} \leq e^{\epsilon}\mathbb{P}\{\mathscr{P}(Z_{\text{per}}) \in B\}$ but for different $\boldsymbol{y}$. For differential privacy $\boldsymbol{y}$ is the sparse vector $\boldsymbol{y} = [0, 1]$ C.1a. For generalized privacy it is the convex set $||[1, 0] - \boldsymbol{y}||_1 \leq 2$ C.1b that *includes* $\boldsymbol{y} = [0, 1]$ together with numerous other $\boldsymbol{y}$.

Proof outline We will define a set $||\boldsymbol{x} - 0||_1 \leq \lambda = d/2\epsilon$ visualized in C.1c which contains a number of perturbed datasets, and which for specific λ includes also the complete dataset Z_{11}. For each dataset $Z_{\boldsymbol{x}}$ we aim to compute mean$[Z_{\boldsymbol{x}}]$. The image of this operation for all $Z_{\boldsymbol{x}}$ is visualized in C.1d. For all $||\boldsymbol{x}||_1 \leq \lambda = d/2\epsilon$ and an ϵ-generalized private algorithm $\mathscr{P}$ if follows that

$$\mu_{\mathbf{0}}(\text{“close}'' \text{ to mean}[Z_{\boldsymbol{x}}]) \geq \exp(-d/2)\mu_{\boldsymbol{x}}(\text{“close}'' \text{ to mean}[Z_{\boldsymbol{x}}]).$$

In effect if given a dataset $Z_{\boldsymbol{x}}$ the algorithm is accurate for mean$[Z_{\boldsymbol{x}}]$, it will also return the same result with non-zero probability given the trivial dataset $Z_{\mathbf{0}}$.

We can formalize this “closeness” using the ℓ_2 norm, for a mechanism $\mathscr{P}$ we then set the expected error for all mean$[Z_{\boldsymbol{x}}]$ as $\mathbb{E}[||\mathscr{P}(Z_{\boldsymbol{x}}) - \text{mean}(Z_{\boldsymbol{x}})||_2]$ and using Markov's inequality

$$\mathbb{P}\{||\mathscr{P}(Z_{\boldsymbol{x}}) - \text{mean}(Z_{\boldsymbol{x}})||_2 \leq 2\mathbb{E}[\text{error}]\} \geq \frac{1}{2}.$$

Intuitively this implies that we are hypothesizing a mechanism that returns for all mean$(Z_{\boldsymbol{x}})$ with probability $\geq \frac{1}{2}$ a result within an $2\mathbb{E}$[error]-ball around the correct solution. Note also that due to ϵ-generalized privacy

$$\mu_{\mathbf{0}}(\{||\mathscr{P}(Z_{\boldsymbol{x}}) - \text{mean}(Z_{\boldsymbol{x}})||_2 \leq 2\mathbb{E}[\text{error}]\}) \geq \exp(-d/2)\frac{1}{2},$$

this means that we have a non-zero probability with mechanism $\mathscr{P}$ given the trivial dataset $Z_{\mathbf{0}}$ to return an output within an $2\mathbb{E}$[error]-ball of mean$(Z_{\boldsymbol{x}})$. However as visualized in C.1d there are *a lot* of means that the algorithm $\mathscr{P}$ returns with sufficient accuracy using the trivial dataset $Z_{\mathbf{0}}$. We complete the proof by showing that for a low enough error $\mathbb{E}$[error] the corresponding error balls are disjoint, count them and using a union bound show the contradiction

$$1 \geq \mu_{\mathbf{0}}(\cup_{\boldsymbol{x}\in X}\{||\mathscr{P}(Z_{\boldsymbol{x}}) - \text{mean}(Z_{\boldsymbol{x}})||_2 \leq 2\mathbb{E}[\text{error}]\}) > 1.$$

As such there is at least one dataset $||\boldsymbol{x}||_1 \leq d/2\epsilon$ such that the expected error is greater than what we assumed in the proof. That the complete unperturbed dataset is Z_{11} results in a condition of the form $n \leq d/2\epsilon \Rightarrow \epsilon \leq d/2n$, in effect the privacy needs to be relatively *high*.

C.4 Proof of Lemma 3.3.2

Lemma 3.3.2. *Let* $\mathbf{X}_k \in \mathbb{R}^{d\times n_k}$, $\mathbf{X}_k = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{n_k}\} \sim p(\boldsymbol{x}|\mathcal{C}_k)^{n_k}$ *following the Gaussian mixture model with known fixed covariance* $\boldsymbol{\Sigma} = \mathbf{I}$. *Then for* ϵ*-generalized private mechanisms* $\mathscr{P}$ *with* $(d/2n_k) \geq \epsilon > 0$, *when* $d - (\ln(0.05) + d(1 - \mathcal{O}(1)))/d \leq n_k$ *we have*

$$\sup_{\mathscr{P}} \mathbb{E}[||\mathscr{P}(\mathbf{X}_k) - \mathbf{w}_k||_2] \geq \mathcal{O}(\epsilon^{-1} d\sqrt{\ln(2n_k/d)}) \tag{C.21}$$

with probability ≥ 0.95 *over the random draws of* $\mathbf{X}_k$, *and the expectation is over the randomness of* $\mathscr{P}$.

Proof. We first restate the definition of a ϵ-generalized private algorithm.

Definition C.4.1. A randomized algorithm $\mathscr{P} : \mathcal{Z}^n \to T$ is ϵ-generalized private if, for a dataset $Z \in \mathcal{Z}^n$ and vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ such that $||\boldsymbol{x} - \boldsymbol{y}||_1 \leq k$, and all measurable subsets $B \subseteq T$, we have $\mathbb{P}\{\mathscr{P}(Z\text{Diag}(\boldsymbol{x})) \in B\} \leq e^{\epsilon k}\mathbb{P}\{\mathscr{P}(Z\text{Diag}(\boldsymbol{y})) \in B\}$.

We have assumed that the covariance $\tilde{\boldsymbol{\Sigma}}$ is known and fixed, therefore ϵ-generalized private mechanism needs to target $\tilde{\boldsymbol{\mu}}_k$, this can be seen by calculating $||\mathscr{P}(\mathbf{X}_k) - \mathbf{w}_k||_2 = ||\mathscr{P}(\mathbf{X}_k) - \tilde{\boldsymbol{\Sigma}}^{-1}\tilde{\boldsymbol{\mu}}_k||_2 = ||\mathscr{P}(\mathbf{X}_k) - \mathbf{I}^{-1}\tilde{\boldsymbol{\mu}}_k||_2 = ||\mathscr{P}(\mathbf{X}_k) - \tilde{\boldsymbol{\mu}}_k||_2$ and taking the expectation over the randomness of $\mathscr{P}$ for both sides.

Suppose $F : \mathbb{R}^n \times \mathbb{R}^{d\times n} \to \mathbb{R}^d$ so that $F(\boldsymbol{x}; \mathbf{A}) = \mathbf{A}\text{Diag}(\boldsymbol{x})\boldsymbol{m} = \mathbf{A}\text{Diag}(\boldsymbol{m})\boldsymbol{x}$ is a linear map, with $\boldsymbol{m} = [(1/n), \cdots, (1/n)] \in \mathbb{R}^{n\times 1}$. We now note that estimating the mean $\tilde{\boldsymbol{\mu}}_k$ can be seen as estimating using a private mechanism the result of the linear map

$$F(\mathbf{1}; \mathbf{X}_k) = \mathbf{X}_k\text{Diag}(\mathbf{1})[\frac{1}{n_k}, \frac{1}{n_k}, \dots, \frac{1}{n_k}]^\top = \frac{1}{n_k}\sum_{i\in\mathcal{C}_k} \boldsymbol{x}_i.$$

where $\mathbf{X}_k$ is the dataset and $\mathbf{1} \in \mathbb{R}^n$ is a vector of ones.

We now recall that a set of points $Y \subseteq \mathbb{R}^d$ is called a $r-$packing if $||y - y'|| \geq r$ for any $y, y' \in Y, y \neq y'$. The following fact will also be useful

Fact C.4.1. *Let* $K \subseteq \mathbb{R}^d$ *such that* $R = \text{vol}(K)^{\frac{1}{d}}$. *Then,* K *contains an* $\mathcal{O}(R\sqrt{d})$*-packing of size at least* $\exp(d)$.

We now need the following lemma from Hardt and Talwar (2010)

Lemma C.4.2. *Let* $\epsilon > 0$ *and suppose* $F : \mathbb{R}^n \times \mathbb{R}^{d\times n} \to \mathbb{R}^d$, $F(\boldsymbol{x}; \mathbf{A}) = \mathbf{A}\text{Diag}(\boldsymbol{m})\boldsymbol{x}$ *is a linear map and let* $K = F(B_1^n; \mathbf{A})$. *Then for every* ϵ*-generalized private mechanism* $\mathscr{P}$, $\exists \boldsymbol{x} \in \mathbb{R}^n : ||\boldsymbol{x}||_1 \leq \lambda = d/2\epsilon$ *such that*

$$\mathbb{E}[||\mathscr{P}(\mathbf{A}\text{Diag}(\boldsymbol{x})) - F(\boldsymbol{x}; \mathbf{A})||_2] \geq \mathcal{O}(\epsilon^{-1} d\sqrt{d} \cdot \text{vol}_d(K)^{\frac{1}{d}}) \tag{C.22}$$

where $\mathrm{vol}_d(\cdot)$ *denotes the n-dimensional Lebesque measure and the expectation is over the randomness of M.*

Proof. Let λ be some scalar and put $R = \mathrm{vol}_d(K)^{\frac{1}{d}}$. By Fact C.4.1 and our assumption, $\lambda K = \lambda F(B_1^n; \mathbf{A})$ contains an $\mathcal{O}(\lambda R\sqrt{d})$-packing Y of size at least $\exp(d)$. Let $X \subseteq \mathbb{R}^n$ be a set of arbitrarily chosen preimages of $y \in Y$ so that $|X| = |Y|$ and $F(X; \mathbf{A}) = Y$. By linearity, $\lambda F(B_1^n; \mathbf{A}) = F(\lambda B_1^n; \mathbf{A}) = \mathbf{A}\mathrm{Diag}(\boldsymbol{m})(\lambda B_1^n)$ and hence we may assume that every $\boldsymbol{x} \in X$ satisfies $||\boldsymbol{x}||_1 \leq \lambda$.

We will now assume that $\mathscr{P} = \{\mu_{\mathbf{A}\mathrm{Diag}(\boldsymbol{x})} : \boldsymbol{x} \in \mathbb{R}^n\}$ is an ϵ-generalized private mechanism with expected error $cd\sqrt{d}R/\epsilon$ for all $||\boldsymbol{x}||_1 \leq \lambda$ and lead this to a contradiction for small enough $c > 0$. For this we set $\lambda = d/2\epsilon$. By the assumption on the expected error, Markov's inequality

$$\begin{aligned}\mathbb{P}\{||\mathscr{P}(\mathbf{A}\mathrm{Diag}(\boldsymbol{x})) - F(\boldsymbol{x}; \mathbf{A})||_2 \geq 2cd\sqrt{d}R/\epsilon\} &\leq \frac{\mathbb{E}[||\mathscr{P}(\mathbf{A}\mathrm{Diag}(\boldsymbol{x})) - F(\boldsymbol{x}; \mathbf{A})||_2]}{2cd\sqrt{d}R/\epsilon} \\ &\leq \frac{cd\sqrt{d}R/\epsilon}{2cd\sqrt{d}R/\epsilon} \\ &\leq \frac{1}{2}\end{aligned} \tag{C.23}$$

this implies that for all preimages $\boldsymbol{x} \in X$, $||\boldsymbol{x}||_1 \leq \lambda$, we have $\mu_{\boldsymbol{x}}(B(\mathbf{A}; \boldsymbol{x})) \geq \frac{1}{2}$, where $B(\mathbf{A}; \boldsymbol{x})$ is a ball of radius $2cd\sqrt{d}R/\epsilon = 4c\lambda R\sqrt{d}$ centered at $F(\boldsymbol{x}; \mathbf{A})$. Since $Y = F(X; \mathbf{A})$ is an $\mathcal{O}(\lambda R\sqrt{d})$-packing, the balls $\{B(\mathbf{A}; \boldsymbol{x}) : \boldsymbol{x} \in X\}$ are disjoint for small enough constant $c > 0$.

Since $||\boldsymbol{x}||_1 \leq \lambda$, it follows from ϵ-generalized privacy with Fact C.4.1 that

$$\mu_0(B(\mathbf{A}; \boldsymbol{x})) \geq \exp(-\epsilon\lambda)\mu_{\boldsymbol{x}}(B(\mathbf{A}; \boldsymbol{x})) \geq \frac{1}{2}\exp(-d/2).$$

Since the balls $B(\mathbf{A}; \boldsymbol{x})$ are pairwise disjoint,

$$1 \geq \mu_0(\cup_{\boldsymbol{x} \in X} B(\mathbf{A}; \boldsymbol{x})) = \sum_{\boldsymbol{x} \in X} \mu_0(B(\mathbf{A}; \boldsymbol{x})) \geq \exp(d)\frac{1}{2}\exp(-d/2) > 1$$

Therefore $\exists \boldsymbol{x} \in \mathbb{R}^n : ||\boldsymbol{x}||_1 \leq \lambda,\ \mathbb{E}[||\mathscr{P}(\mathbf{A}\mathrm{Diag}(\boldsymbol{x})) - F(\boldsymbol{x}; \mathbf{A})||_2] \geq \mathcal{O}(\epsilon^{-1}d\sqrt{d} \cdot \mathrm{vol}_d(K)^{\frac{1}{d}})$. □

How we proceed, now becomes more clear. We assume that there exists a private algorithm $\mathscr{P}$ for which the "worst case" point in C.22 coincides with $\boldsymbol{x} = \mathbf{1} = [1, \cdots, 1] \in \mathbb{R}^{n_k}$. We set $\mathbf{A} = \mathbf{X}_k = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_k}\} \sim \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})^{n_k}$ and $\boldsymbol{x} = \mathbf{1} = [1, \cdots, 1] \in \mathbb{R}^{n_k}$ in C.22.

Remember that $K = F(B_1^n; \mathbf{A}) = \mathbf{A}\text{Diag}(\boldsymbol{m})B_1^n$, we then get

$$\sup_{\mathscr{P}} \mathbb{E}[||\mathscr{P}(\mathbf{X_k}) - \tilde{\boldsymbol{\mu}}_k||_2] \geq \mathcal{O}(\epsilon^{-1} d\sqrt{d} \cdot \text{vol}_d((1/n_k)\mathbf{X}_k B_1^{n_k})^{\frac{1}{d}}). \tag{C.24}$$

and we note that for $\boldsymbol{x} = [1, \cdots, 1] \in \mathbb{R}^{n_k}$ we have $||\boldsymbol{x}||_1 \leq d/2\epsilon \Rightarrow n_k \leq d/2\epsilon \Rightarrow \epsilon \leq d/2n_k$.

We see that what's needed is to estimate the volume of the random Gaussian polytope $K = (1/n_k)\mathbf{X}_k B_1^{n_k}$. This can also be seen as the absolute convex hull of $(1/n_k)\mathbf{X}_k$. The small-ball probabilities for the volume of random convex sets has been extensively studied. Note that typically constants such as $(1/n_k)$ do not alter the results in this sort of analyses. In fact, results such as the following, are derived with minimal non-degeneracy assumptions on the distribution of $\mathbf{X}_k$. As such, we absorb $(1/n_k)$ into $\mathbf{X}_k$. We will use the following lemma from Paouris and Pivovarov (2013)

Lemma C.4.3. *Let $n \geq d$ and let $\mathbf{G} = [\mathbf{g}_1, .., \mathbf{g}_n]$ be an $d \times n$ matrix where $\mathbf{g}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})\ \forall i$, $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\boldsymbol{\Sigma} \in \boldsymbol{S}_+^d$. Then let $\delta > 1$, for $d \leq n \leq de^{\delta^2}$, we have*

$$\mathbb{P}(\{\text{vol}_d(\mathbf{G}B_1^n)^{\frac{1}{d}} \leq \frac{c_1}{\delta e^4}\sqrt{\frac{\ln(2n/d)}{d}}\}) \leq e^{-d(n+1-o(1))+d^2}. \tag{C.25}$$

We then upper bound the probability of the above tail event

$$\begin{aligned} & e^{-d(n+1-o(1))+d^2} \leq 0.05 \\ & \Rightarrow d - \left\lceil \frac{\ln(0.05) + d(1-\mathcal{O}(1))}{d} \right\rceil \leq n \end{aligned} \tag{C.26}$$

For $d - \left\lceil \frac{\ln(0.05)+d(1-\mathcal{O}(1))}{d} \right\rceil \leq n_k$ by applying Lemma C.4.3 over the random draws of $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_{n_k}\} \sim p(\boldsymbol{x}|\mathcal{C}_k)^{n_k}$ on equation (C.24) we get

$$\mathbb{P}(\{\mathbb{E}[||\mathscr{P}(\mathbf{X_k}) - \tilde{\boldsymbol{\mu}}_k||_2] \geq \mathcal{O}(\frac{\epsilon^{-1} d c_1}{\delta e^4}\sqrt{\ln(2n_k/d))}\}) \geq \frac{1}{2}. \tag{C.27}$$

We can then absorb all constants in the $\mathcal{O}$ notation.

□

C.5 Additional limitations for SGLD

Furthermore, SGLD does not provide a closed form solution. The SGLD procedure has to be iterated to sample from the posterior, after some burn-in phase is completed. In particular when estimating KL terms $\mathrm{KL}(\hat{\rho}_\tau||\pi)$ for a fixed prior $\pi(\boldsymbol{\theta})$ and $\tau \geq 0$ one can set $\hat{\rho}_\tau = P_{\exp(-\tau\mathcal{L}^{\ell'}_{X,Y}(f))}$ and $Z_\tau = P[\exp(-\tau\mathcal{L}^{\ell'}_{X,Y}(f))]$. Then they estimate

$$\begin{aligned}
\mathrm{KL}(\hat{\rho}_\tau||\pi) &= \hat{\rho}_\tau[\ln \frac{d\hat{\rho}_\tau}{d\pi}] \\
&= \hat{\rho}_\tau[\ln \frac{\exp(-\tau\mathcal{L}^{\ell'}_{X,Y}(f))}{Z_\tau}] \\
&= \cancel{-\tau\hat{\rho}_\tau[\mathcal{L}^{\ell'}_{X,Y}(f))]} - \ln Z_\tau \\
&\leq -\ln P[\exp(-\tau\mathcal{L}^{\ell'}_{X,Y}(f))] \\
&= -\ln \mathbb{E}[\frac{1}{k}\sum_{i=1}^{k}\exp(-\tau\mathcal{L}^{\ell'}_{X,Y}(f(\boldsymbol{\theta}^i_\pi)))] \\
&\leq \mathbb{E}[-\ln \frac{1}{k}\sum_{i=1}^{k}\exp(-\tau\mathcal{L}^{\ell'}_{X,Y}(f(\boldsymbol{\theta}^i_\pi)))]
\end{aligned} \tag{C.28}$$

In the third line Dziugaite and Roy (2018a) set the first term equal to 0, as it is difficult to compute, and in the last line they use Jensen's inequality. The final term in this approximation will only provide meaningful results for data dependent priors, that provide prior samples close to regions with low empirical loss. As noted in Dziugaite and Roy (2018a), the numerical integration performed in the final term rapidly diminishes in accuracy with increasing dimensionality of the parameter space. Furthermore, with high probability samples from a high dimensional Gaussian lie far in terms of ℓ_2 norm from the mean, and consequently in areas where the empirical loss might be high. As such, the upper bound on the KL divergence is very loose. Furthermore, note that for small values of τ the iterations of SGLD are often unstable and do not converge.

C.6 Additional proofs for moments accountant

Simplifying to 1-dimensional integral

Lemma 3.2.7. *Let* $\mathrm{SG}_{q,\sigma}$ *be the Sampled Gaussian mechanism for some function* f*. Then*

$$a_{\mathrm{SG}_{q,\sigma}}(\lambda; Z, Z') \leq \log \mathbb{E}_{z\sim\mu_0}[(\mu(z)/\mu_0(z))^{\lambda+1}] \tag{C.29}$$

where $\mu_0 = \mathcal{N}(0, \sigma^2)$*,* $\mu_1 = \mathcal{N}(1, \sigma^2)$*,* $\mu \triangleq (1-q)\mu_0 + q\mu_1$ *and assuming that* $||f(Z) - f(Z')||_2 \leq 1$*.*

Proof. Let $Z, Z' \in \hat{\mathcal{Z}} \subseteq \mathcal{Z}^n$ be a pair of adjacent datasets such that $Z' = Z \cup \{\boldsymbol{x}\}$. We will frame our derivations in terms of the Rényi divergence

$$\mathrm{D}_\lambda(p||q) = \frac{1}{1-\lambda} \log \mathbb{E}_{z\sim q} \left(\frac{p(z)}{q(z)}\right)^\lambda.$$

We wish to bound the Rényi divergences $\mathrm{D}_\lambda(\mathscr{P}(Z')||\mathscr{P}(Z))$ and $\mathrm{D}_\lambda(\mathscr{P}(Z)||\mathscr{P}(Z'))$, where $\mathscr{P}$ is the sampled Gaussian mechanism for some function f with ℓ_2-sensitivity 1.

Let L denote a set-values random variable defined by taking a random subset of $\mathcal{Z}^n$, where each $Z \in \hat{\mathcal{Z}} \subseteq \mathcal{Z}^n$ is independently placed in L with probability q. Conditioned on L, the mechanism $\mathscr{P}(Z)$ samples from a Gaussian with mean $f(L)$. Thus

$$\mathscr{P}(Z) = \sum_L p_L \mathcal{N}(f(L), \sigma^2 \mathbf{I}^d), \tag{C.30}$$

where the sum here denotes mixing of the distributions with weights p_L. Similarly,

$$\mathscr{P}(Z') = \sum_L p_L \left((1-q)\mathcal{N}(f(L), \sigma^2\mathbf{I}^d) + q\mathcal{N}(f(L \cup \{\boldsymbol{x}\}), \sigma^2\mathbf{I}^d)\right). \tag{C.31}$$

Rényi divergence is quasi-convex, allowing one to bound

$$\begin{aligned}
&\mathrm{D}_\lambda(\mathscr{P}(Z)||\mathscr{P}(Z')) \\
&\leq \sup_L \mathrm{D}_\lambda \left(\mathcal{N}(f(L), \sigma^2\mathbf{I}^d) \;||\; (1-q)\mathcal{N}(f(L), \sigma^2\mathbf{I}^d) + q\mathcal{N}(f(L \cup \{\boldsymbol{x}\}), \sigma^2\mathbf{I}^d)\right) \\
&\leq \sup_L \mathrm{D}_\lambda \left(\mathcal{N}(0, \sigma^2\mathbf{I}^d) \;||\; (1-q)\mathcal{N}(0, \sigma^2\mathbf{I}^d) + q\mathcal{N}(f(L \cup \{\boldsymbol{x}\}) - f(L), \sigma^2\mathbf{I}^d)\right)
\end{aligned} \tag{C.32}$$

where we have used the translation invariance of Rényi divergence. Since the covariances are symmetric, we can, by applying a rotation, assume that $f(L \cup \{\boldsymbol{x}\}) - f(L) = c_L \boldsymbol{e}_1$ for some constant $c_L \leq 1$. The two distributions at hand are then both product distributions that are identical in all coordinates except the first. By additivity of Rényi divergence

for product distributions, we have that

$$\begin{aligned}\mathrm{D}_\lambda(\mathscr{P}(Z)||\mathscr{P}(Z')) &\leq \sup_{c\leq 1} \mathrm{D}_\lambda\left(\mathcal{N}(0,\sigma^2)\;||\;(1-q)\mathcal{N}(0,\sigma^2)+q\mathcal{N}(c,\sigma^2)\right)\\ &= \sup_{c\leq 1} \mathrm{D}_\lambda\left(\mathcal{N}(0,(\sigma/c)^2)\;||\;(1-q)\mathcal{N}(0,(\sigma/c)^2)+q\mathcal{N}(c,(\sigma/c)^2)\right)\end{aligned} \tag{C.33}$$

For any $c \leq 1$, the noise $\mathcal{N}(0,(\sigma/c)^2)$ can be obtained from $\mathcal{N}(0,\sigma^2)$ by adding noise from $\mathcal{N}(0,(\sigma/c)^2-\sigma^2)$, and the same operation allows us to obtain $(1-q)\mathcal{N}(0,(\sigma/c)^2)+q\mathcal{N}(1,(\sigma/c)^2)$ from $(1-q)\mathcal{N}(0,\sigma^2)+q\mathcal{N}(1,\sigma^2)$. Thus by the data processing inequality for Rényi divergence, we conclude

$$\mathrm{D}_\lambda(\mathscr{P}(Z)||\mathscr{P}(Z')) \leq \mathrm{D}_\lambda\left(\mathcal{N}(0,\sigma^2)\;||\;(1-q)\mathcal{N}(0,\sigma^2)+q\mathcal{N}(1,\sigma^2)\right) = A_\lambda \tag{C.34}$$

An identical argument implies that

$$\mathrm{D}_\lambda(\mathscr{P}(Z')||\mathscr{P}(Z)) \leq \mathrm{D}_\lambda\left((1-q)\mathcal{N}(0,\sigma^2)+q\mathcal{N}(1,\sigma^2)\;||\;\mathcal{N}(0,\sigma^2)\right) = B_\lambda. \tag{C.35}$$

In effect we have derived an upper bound on $\mathrm{D}_\lambda(\mathscr{P}(Z^{(\cdot)})||\mathscr{P}(Z^{(\cdot)}))$ so that

$$\mathrm{D}_\lambda(\mathscr{P}(Z^{(\cdot)})||\mathscr{P}(Z^{(\cdot)})) \leq \max(A_\lambda, B_\lambda).$$

We include also the following without proof

Lemma C.6.1. *Given* $A_\lambda = \mathrm{D}_\lambda\left(\mathcal{N}(0,\sigma^2)\;||\;(1-q)\mathcal{N}(0,\sigma^2)+q\mathcal{N}(1,\sigma^2)\right)$ *and* $B_\lambda = \mathrm{D}_\lambda\left((1-q)\mathcal{N}(0,\sigma^2)+q\mathcal{N}(1,\sigma^2)\;||\;\mathcal{N}(0,\sigma^2)\right)$

$$B_\lambda \leq A_\lambda.$$

Lemma 3.2.7 follows by noting that

$$\begin{aligned}a_{\mathrm{SG}_{q,\sigma}}(\lambda;Z,Z') &= \log \mathbb{E}_{o\sim\mathscr{P}(Z)}\left[\exp \lambda c(o;\mathscr{P},Z,Z')\right]\\ &= \log \mathbb{E}_{o\sim\mathscr{P}(Z)}\left[\exp \lambda \log \frac{\mathbb{P}\{\mathscr{P}(Z)=o\}}{\mathbb{P}\{\mathscr{P}(Z')=o\}}\right]\\ &= \log \mathbb{E}_{o\sim\mathscr{P}(Z)}\left[\left(\frac{\mathbb{P}\{\mathscr{P}(Z)=o\}}{\mathbb{P}\{\mathscr{P}(Z')=o\}}\right)^\lambda\right]\\ &= \lambda\frac{1}{\lambda}\log \mathbb{E}_{o\sim\mathscr{P}(Z')}\left[\left(\frac{\mathbb{P}\{\mathscr{P}(Z)=o\}}{\mathbb{P}\{\mathscr{P}(Z')=o\}}\right)^{\lambda+1}\right]\\ &= \lambda \mathrm{D}_{\lambda+1}(\mathscr{P}(Z')||\mathscr{P}(Z))\end{aligned} \tag{C.36}$$

□

Numerically Stable Computation

Naively, $C_\lambda = \mathbb{E}_{z\sim\mu_0}[(\mu(z)/\mu_0(z))^\lambda]$ can be approximated as an integral using standard numerical libraries. It however leads to the problem of computing an integral over the whole real line of a quantity that can vary a lot. We sidestep this difficulty by expressing C_λ as a finite sum (or a convergent series), swap the order of the integration and summation operators, and compute the integrals analytically.

To compute $C_\lambda = \mathbb{E}_{z\sim\mu_0}[(\mu(z)/\mu_0(z))^\lambda]$, we write

$$\left(\frac{\mu(z)}{\mu_0(z)}\right)^\lambda = \left((1-q) + q\frac{\mu_1(z)}{\mu_0(z)}\right)^\lambda \tag{C.37}$$

and consider two cases.

Case I: Integer λ. Applying a binomial expansion to C.37 we have

$$\left(\frac{\mu(z)}{\mu_0(z)}\right)^\lambda = \sum_{k=0}^{\lambda} \binom{\lambda}{k} (1-q)^{\lambda-k} q^k \left(\frac{\mu_1(z)}{\mu_0(z)}\right)^k$$

Thus it suffices to compute for $k \in \{0, \cdots, \lambda\}$ the expectation

$$\mathbb{E}_{z\sim\mu_0}\left[\left(\frac{\mu_1(z)}{\mu_0(z)}\right)^\lambda\right]$$

We observe that the terms of the form $\mathbb{E}_{z\sim\mu_0}\left[\left(\frac{\mu_1(z)}{\mu_0(z)}\right)^\lambda\right]$ have an analytical closed form that can be obtained by integration. Indeed,

$$\begin{aligned}
\mathbb{E}_{z\sim\mu_0}\left[\left(\frac{\mu_1(z)}{\mu_0(z)}\right)^\lambda\right] &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left\{-\frac{x^2}{2\sigma^2} + k\cdot\frac{x^2-(x-1)^2}{2\sigma^2}\right\} \mathrm{d}x \\
&= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left\{-\frac{x^2}{2\sigma^2} + \frac{2kx-k}{2\sigma^2}\right\} \mathrm{d}x \\
&= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left\{-\frac{(x-k)^2}{2\sigma^2} + \frac{k^2-k}{2\sigma^2}\right\} \mathrm{d}x \\
&= \exp\left(\frac{k^2-k}{2\sigma^2}\right) \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(-\frac{y^2}{2\sigma^2}\right) \mathrm{d}y \quad \text{(substituting } y \triangleq x-k\text{)} \\
&= \exp\left(\frac{k^2-k}{2\sigma^2}\right).
\end{aligned} \tag{C.38}$$

Case II: Fractional λ. To rewrite C.37 as a convergent series, consider two cases depending on how $1-q$ compares with $q\mu_1(z)/\mu_0(z)$. The inflection point is z_1 where

the two quantities are equal:

$$\begin{aligned}(1-q)\mu_0(z_1) = q\mu_1(z_1) &\Leftrightarrow 1-q = qe^{\frac{2z_1-1}{2\sigma^2}} \\ &\Leftrightarrow z_1 = \frac{1}{2} + \sigma^2 \ln(q^{-1}-1)\end{aligned}$$

Thus we can express

$$\left(\frac{\mu(z)}{\mu_0(z)}\right)^\lambda = \begin{cases} \sum_{k=0}^{\infty} \binom{\lambda}{k} (1-q)^{\lambda-k} q^k \left(\frac{\mu_1(z)}{\mu_0(z)}\right)^k & \text{when } z \leq z_1 \\ \sum_{k=0}^{\infty} \binom{\lambda}{k} (1-q)^{k} q^{\lambda-k} \left(\frac{\mu_1(z)}{\mu_0(z)}\right)^k & \text{when } z > z_1 \end{cases}. \tag{C.39}$$

Analogously to the case of integer λ, we compute the expectations of both series under $z \sim \mu_0$, where the integrals are taken over the half lines $(-\infty, z_1]$ and $[z_1, +\infty)$:

$$\begin{aligned}\int_{-\infty}^{z_1} \mu_0(x) \left(\frac{\mu_1(x)}{\mu_0(x)}\right)^k \mathrm{d}x &= \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{z_1} \exp\left\{-\frac{x^2}{2\sigma^2} + k \cdot \frac{x^2-(x-1)^2}{2\sigma^2}\right\} \mathrm{d}x \\ &= \exp\left(\frac{k^2-k}{2\sigma^2}\right) \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{z_1-k} \exp\left(-\frac{y^2}{2\sigma^2}\right) \mathrm{d}x \\ &= \frac{1}{2} \exp\left(\frac{k^2-k}{2\sigma^2}\right) \operatorname{erfc}\left(\frac{k-z_1}{\sqrt{2}\sigma}\right), \\ \int_{z_1}^{+\infty} \mu_0(x) \left(\frac{\mu_1(x)}{\mu_0(x)}\right)^k \mathrm{d}x &= \frac{1}{2} \exp\left(\frac{k^2-k}{2\sigma^2}\right) \operatorname{erfc}\left(\frac{z_1-k}{\sqrt{2}\sigma}\right).\end{aligned} \tag{C.40}$$

The computation done in the privacy accountant proceeds by plugging in these quantities into the series (C.39), and carrying out the summation to convergence.

C.7 Additional Figures

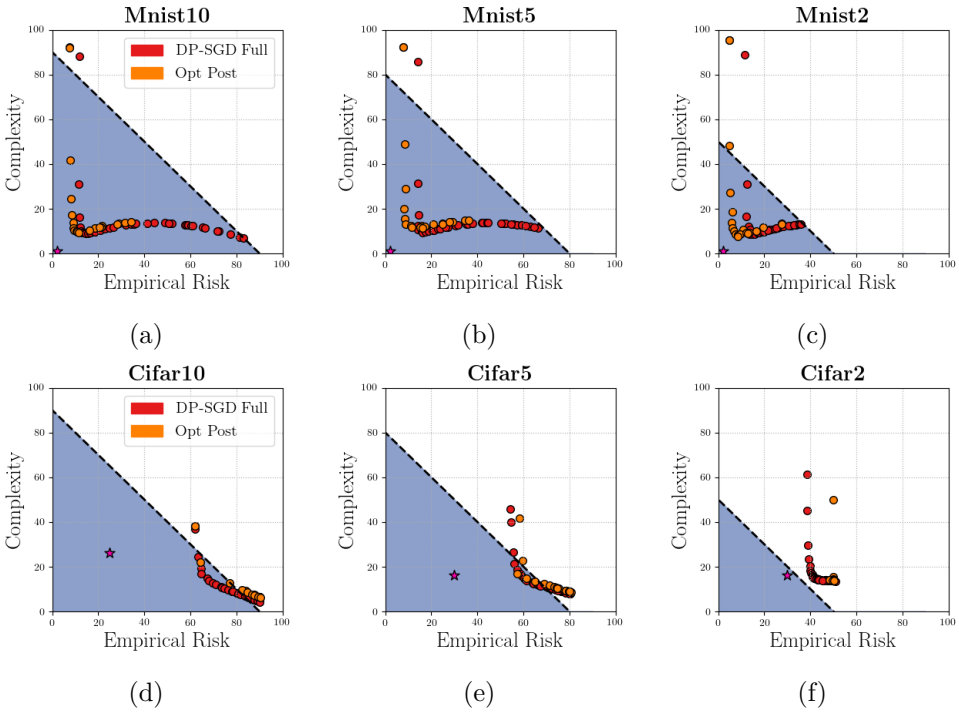
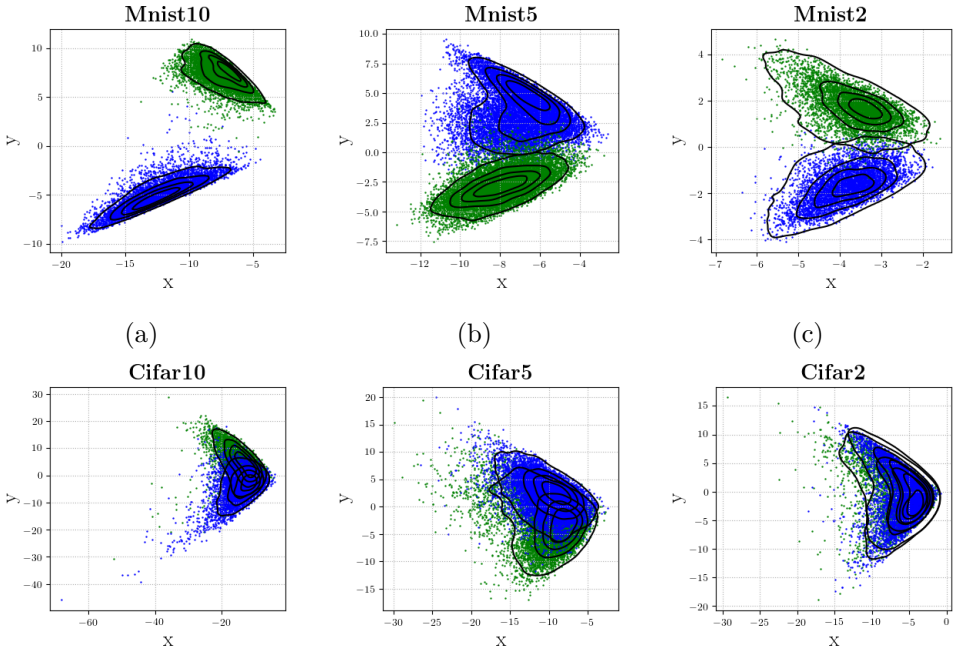


Figure C.2 – **The effect of optimizing the posterior**: We see that the change in bounds is relatively small across the different datasets. Furthermore the results are mixed. In MNIST cases there is a small improvement. At the same time for CIFAR there is a loosening of the bounds.



(a) (b) (c)

(d) (e) (f)

Figure C.3 – **Visualizing the latent representations**: We use pretrained deterministic neural networks to visualize the latent representations at the input of the final softmax layer. We compute the singular value decomposition of two classes and project the datapoints along the first two principal directions. We also plot the contours of a Gaussian kernel density estimator that we fit on the data. The samples concentrate closely around a mean validating partially our modeling choice to approximate the distribution of latent representations as a mixture of Gaussians.

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. (2016). Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318.

Achille, A. and Soatto, S. (2018). Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980.

Allen-Zhu, Z., Li, Y., and Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252.

Alquier, P., Ridgway, J., and Chopin, N. (2016). On the properties of variational approximations of gibbs posteriors. *The Journal of Machine Learning Research*, 17(1):8374–8414.

Ambroladze, A., Parrado-Hernández, E., and Shawe-taylor, J. S. (2007). Tighter pac-bayes bounds. In *Advances in neural information processing systems*, pages 9–16.

Andrew, G., Chien, S., and Papernot, N. (2019). Tensorflow privacy. Software available from https://github.com/tensorflow/privacy.

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. (2018). Stronger generalization bounds for deep nets via a compression approach. *arXiv preprint arXiv:1802.05296*.

Bae, J., Zhang, G., and Grosse, R. (2018). Eigenvalue corrected noisy natural gradient. *arXiv preprint arXiv:1811.12565*.

Bandeira, A. S., Van Handel, R., et al. (2016). Sharp nonasymptotic bounds on the norm of random matrices with independent entries. *The Annals of Probability*, 44(4):2479–2506.

Barron, A., Rissanen, J., and Yu, B. (1998). The minimum description length principle in coding and modeling. pages 531–540.

Bartlett, P. L., Foster, D. J., and Telgarsky, M. J. (2017). Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pages 6241–6250.

Bartlett, P. L. and Mendelson, S. (2002). Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482.

Bassily, R., Smith, A., and Thakurta, A. (2014). Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE.

Belghazi, M. I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., and Hjelm, D. (2018). Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540.

Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

Biggs, F. and Guedj, B. (2020). Differentiable pac-bayes objectives with partially aggregated neural networks. *arXiv preprint arXiv:2006.12228*.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Blier, L. and Ollivier, Y. (2018). The description length of deep learning models. In *Advances in Neural Information Processing Systems*, pages 2216–2226.

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1989). Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*.

Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *Journal of machine learning research*, 2(Mar):499–526.

Catoni, O. (2003). A pac-bayesian approach to adaptive classification. *preprint*, 840.

Catoni, O. (2007). Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. (2019). Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018.

Chaudhuri, K. and Hsu, D. (2011). Sample complexity bounds for differentially private learning. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 155–186.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. (2011). Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3).

Chollet, F. et al. (2015). keras.

Cortes, C., Gonzalvo, X., Kuznetsov, V., Mohri, M., and Yang, S. (2017). Adanet: Adaptive structural learning of artificial neural networks. In *International conference on machine learning*, pages 874–883. PMLR.

Dangel, F., Harmeling, S., and Hennig, P. (2020). Modular block-diagonal curvature approximations for feedforward architectures. In *International Conference on Artificial Intelligence and Statistics*, pages 799–808.

Dangel, F., Kunstner, F., and Hennig, P. (2019). Backpack: Packing more into backprop. *arXiv preprint arXiv:1912.10985*.

Daniely, A. and Granot, E. (2019). Generalization bounds for neural networks via approximate description length. In *International Conference on Machine Learning*, pages 531–540.

Davenport, T. and Kalakota, R. (2019). The potential for artificial intelligence in healthcare. *Future healthcare journal*, 6(2):94.

De, A. (2012). Lower bounds in differential privacy. In *Theory of cryptography conference*, pages 321–338. Springer.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. (2017). Tensorflow distributions. *arXiv preprint arXiv:1711.10604*.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. (2017). Sharp minima can generalize for deep nets. *arXiv preprint arXiv:1703.04933*.

Dong, X., Chen, S., and Pan, S. (2017). Learning to prune deep neural networks via layer-wise optimal brain surgeon. In *Advances in Neural Information Processing Systems*, pages 4860–4874.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685.

Du, S. S., Lee, J. D., Tian, Y., Poczos, B., and Singh, A. (2017). Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. *arXiv preprint arXiv:1712.00779*.

Dwork, C. (2008). Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer.

Dwork, C. (2011). Differential privacy. *Encyclopedia of Cryptography and Security*, pages 338–340.

Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407.

Dziugaite, G. K. and Roy, D. M. (2017). Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*.

Dziugaite, G. K. and Roy, D. M. (2018a). Data-dependent pac-bayes priors via differential privacy. In *Advances in Neural Information Processing Systems*, pages 8430–8441.

Dziugaite, G. K. and Roy, D. M. (2018b). Entropy-sgd optimizes the prior of a pac-bayes bound: Data-dependent pac-bayes priors via differential privacy.

Elsken, T., Metzen, J. H., and Hutter, F. (2018). Neural architecture search: A survey. *arXiv preprint arXiv:1808.05377*.

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., and Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118.

Gabrié, M., Manoel, A., Luneau, C., Macris, N., Krzakala, F., Zdeborová, L., et al. (2018). Entropy and mutual information in models of deep neural networks. In *Advances in Neural Information Processing Systems*, pages 1821–1831.

Germain, P., Bach, F., Lacoste, A., and Lacoste-Julien, S. (2016). Pac-bayesian theory meets bayesian inference. In *Advances in Neural Information Processing Systems*, pages 1884–1892.

Golowich, N., Rakhlin, A., and Shamir, O. (2017). Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*.

Grunwald, P. (2004). A tutorial introduction to the minimum description length principle. pages 531–540.

Hardt, M., Recht, B., and Singer, Y. (2016). Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pages 1225–1234.

Hardt, M. and Talwar, K. (2010). On the geometry of differential privacy. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 705–714.

Hassibi, B. and Stork, D. G. (1993). Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in neural information processing systems*, pages 164–171.

Hinton, G. and van Camp, D. (1993). Keeping neural networks simple by minimising the description length of weights. 1993. In *International Conference on Machine Learning*, pages 531–540.

Hochreiter, S. and Schmidhuber, J. (1997a). Flat minima. *Neural Computation*, 9(1):1–42.

Hochreiter, S. and Schmidhuber, J. (1997b). Low-complexity coding and decoding. *Theoretical aspects of neural computation (TANC 97), Hong Kong*, pages 297–306.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Honkela, A. and Valpola, H. (2004). Variational learning and bits-back coding: an information-theoretic view to bayesian learning. pages 531–540.

Jacobsen, J.-H., Smeulders, A., and Oyallon, E. (2018). i-revnet: Deep invertible networks. *arXiv preprint arXiv:1802.07088*.

Jain, P. and Thakurta, A. G. (2014). (near) dimension independent risk bounds for differentially private learning. In *International Conference on Machine Learning*, pages 476–484.

Jastrzębski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y., and Storkey, A. (2017). Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*.

Jiang, Y., Krishnan, D., Mobahi, H., and Bengio, S. (2018). Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D., and Bengio, S. (2019). Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*.

Kääriäinen, M. and Langford, J. (2005). A comparison of tight generalization error bounds. In *Proceedings of the 22nd international conference on Machine learning*, pages 409–416.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Kawaguchi, K., Kaelbling, L. P., and Bengio, Y. (2017). Generalization in deep learning. *arXiv preprint arXiv:1710.05468*.

Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems*, pages 2575–2583.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kleinberg, R., Li, Y., and Yuan, Y. (2018). An alternative view: When does sgd escape local minima? *arXiv preprint arXiv:1802.06175*.

Krizhevsky, A. and Hinton, G. (2010). Convolutional deep belief networks on cifar-10. *Unpublished manuscript*, 40(7):1–9.

Kunstner, F., Balles, L., and Hennig, P. (2019). Limitations of the empirical fisher approximation. *arXiv preprint arXiv:1905.12558*.

Kuzborskij, I. and Lampert, C. (2017). Data-dependent stability of stochastic gradient descent. *arXiv preprint arXiv:1703.01678*.

Langford, J. (2005). Tutorial on practical prediction theory for classification. *Journal of machine learning research*, 6(Mar):273–306.

Langford, J. and Caruana, R. (2002). (not) bounding the true error. In *Advances in Neural Information Processing Systems*, pages 809–816.

LeCun, Y. and Cortes, C. (2010). MNIST handwritten digit database.

LeCun, Y., Denker, J. S., and Solla, S. A. (1990). Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605.

Lever, G., Laviolette, F., and Shawe-Taylor, J. (2013). Tighter pac-bayes bounds through distribution-dependent priors. *Theoretical Computer Science*, 473:4–28.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018a). Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399.

Li, X., Lu, J., Wang, Z., Haupt, J., and Zhao, T. (2018b). On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond. *arXiv preprint arXiv:1806.05159*.

Liang, T., Poggio, T., Rakhlin, A., and Stokes, J. (2019). Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 888–896.

Lin, W., Khan, M. E., and Schmidt, M. (2019a). Fast and simple natural-gradient variational inference with mixture of exponential-family approximations. *arXiv preprint arXiv:1906.02914*.

Lin, W., Khan, M. E., and Schmidt, M. (2019b). Stein's lemma for the reparameterization trick with exponential family mixtures. *arXiv preprint arXiv:1910.13398*.

Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.-J., Fei-Fei, L., Yuille, A., Huang, J., and Murphy, K. (2018). Progressive neural architecture search. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34.

Long, P. M. and Sedghi, H. (2019). Size-free generalization bounds for convolutional neural networks. *arXiv preprint arXiv:1905.12600*.

Louizos, C., Shi, X., Schutte, K., and Welling, M. (2019). The functional neural process. In *Advances in Neural Information Processing Systems*, pages 8746–8757.

Lyle, C., van der Wilk, M., Kwiatkowska, M., Gal, Y., and Bloem-Reddy, B. (2020). On the benefits of invariance in neural networks. *arXiv preprint arXiv:2005.00178*.

Maddox, W., Garipov, T., Izmailov, P., Vetrov, D., and Wilson, A. G. (2019). A simple baseline for bayesian uncertainty in deep learning. *arXiv preprint arXiv:1902.02476*.

Mallat, S. (2016). Understanding deep convolutional networks. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150203.

Martens, J. and Grosse, R. (2015). Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417.

Masters, D. and Luschi, C. (2018). Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*.

McAllester, D. A. (1999). Some pac-bayesian theorems. *Machine Learning*, 37(3):355–363.

Mironov, I., Talwar, K., and Zhang, L. (2019). R\'enyi differential privacy of the sampled gaussian mechanism. *arXiv preprint arXiv:1908.10530*.

Mishkin, A., Kunstner, F., Nielsen, D., Schmidt, M., and Khan, M. E. (2018). Slang: Fast structured covariance approximations for bayesian deep learning with natural gradient. In *Advances in Neural Information Processing Systems*, pages 6245–6255.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Nagarajan, V. and Kolter, J. Z. (2019). Uniform convergence may be unable to explain generalization in deep learning. *arXiv preprint arXiv:1902.04742*.

Neelakantan, A., Vilnis, L., Le, Q. V., Sutskever, I., Kaiser, L., Kurach, K., and Martens, J. (2015). Adding gradient noise improves learning for very deep networks. *arXiv preprint arXiv:1511.06807*.

Nelson, D. M., Pereira, A. C., and de Oliveira, R. A. (2017). Stock market's price movement prediction with lstm neural networks. In *2017 International joint conference on neural networks (IJCNN)*, pages 1419–1426. IEEE.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. (2017a). Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. (2017b). A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*.

Neyshabur, B., Tomioka, R., and Srebro, N. (2015). Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401.

Paouris, G. and Pivovarov, P. (2013). Small-ball probabilities for the volume of random convex sets. *Discrete & Computational Geometry*, 49(3):601–646.

Parrado-Hernández, E., Ambroladze, A., Shawe-Taylor, J., and Sun, S. (2012). Pac-bayes bounds with data dependent priors. *Journal of Machine Learning Research*, 13(Dec):3507–3531.

Peng, H., Wu, J., Chen, S., and Huang, J. (2019). Collaborative channel pruning for deep networks. In *International Conference on Machine Learning*, pages 5113–5122.

Pham, H., Guan, M. Y., Zoph, B., Le, Q. V., and Dean, J. (2018). Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*.

Pitas, K. (2020). Dissecting non-vacuous generalization boundsbased on the mean-field approximation. *Proceedings of the 37th International Conference on Machine learning*.

Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*.

Rissanen, J. (1978). Modeling by shortest data description. pages 531–540.

Ritter, H., Botev, A., and Barber, D. (2018). A scalable laplace approximation for neural networks. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, volume 6. International Conference on Representation Learning.

Seeger, M. (2002). Pac-bayesian generalisation error bounds for gaussian process classification. *Journal of machine learning research*, 3(Oct):233–269.

Sokolic, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. (2016). Generalization error of invariant classifiers. *arXiv preprint arXiv:1610.04574*.

Sokolić, J., Giryes, R., Sapiro, G., and Rodrigues, M. R. (2016). Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. (2018). The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Sun, S., Zhang, G., Shi, J., and Grosse, R. (2019). Functional variational bayesian neural networks. *arXiv preprint arXiv:1903.05779*.

Suzuki, T. (2019). Compression based bound for non-compressed network: unified generalization error analysis of large compressible deep neural network. *arXiv preprint arXiv:1909.11274*.

Thomas, V., Pedregosa, F., van Merriënboer, B., Mangazol, P.-A., Bengio, Y., and Roux, N. L. (2019). Information matrices and generalization. *arXiv preprint arXiv:1906.07774*.

Vershynin, R. (2010). Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*.

Vershynin, R. (2018). *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press.

Wang, C., Grosse, R., Fidler, S., and Zhang, G. (2019). Eigendamage: Structured pruning in the kronecker-factored eigenbasis. *arXiv preprint arXiv:1905.05934*.

Wang, H., Keskar, N. S., Xiong, C., and Socher, R. (2018). Identifying generalization properties in neural networks. *arXiv preprint arXiv:1809.07402*.

Wei, C. and Ma, T. (2019). Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *arXiv preprint arXiv:1905.03684*.

Welling, M. and Teh, Y. W. (2011). Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688.

Wen, Y., Vicol, P., Ba, J., Tran, D., and Grosse, R. (2018). Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*.

Wiatowski, T. and Bölcskei, H. (2018). A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, 64(3):1845–1866.

Wu, A., Nowozin, S., Meeds, E., Turner, R. E., Hernández-Lobato, J. M., and Gaunt, A. L. (2018). Deterministic variational inference for robust bayesian neural networks. *arXiv preprint arXiv:1810.03958*.

You, Y., Gitman, I., and Ginsburg, B. (2017a). Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*.

You, Y., Gitman, I., and Ginsburg, B. (2017b). Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 6.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zhang, G., Sun, S., Duvenaud, D., and Grosse, R. (2018). Noisy natural gradient as variational inference. In *International Conference on Machine Learning*, pages 5852–5861.

Zhou, P. and Feng, J. (2018). Understanding generalization and optimization performance of deep cnns. *arXiv preprint arXiv:1805.10767*.

Zhou, W., Veitch, V., Austern, M., Adams, R. P., and Orbanz, P. (2018). Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach.

Zoph, B. and Le, Q. V. (2016). Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*.

Konstantinos Pitas

Email: konstantinos.pitas@epfl.ch
LinkedIn: konstantinos-pitas-lts2-epfl

Education

École Polytechnique Fédérale de Lausanne — Lausanne, Switzerland
Ph.D. in Machine Learning, Advisor: Pierre Vandergheynst — 2015–Current

Aristotle University of Thessaloniki — Thessaloniki, Greece
M.S. in Electrical, Electronics and Communications Engineering, GPA: 8.26/10.00 — 2014–2015

- Thesis: "Disjunctive Programming for Parametric Dictionary Learning on Multi-Layer Graphs"

Aristotle University of Thessaloniki — Thessaloniki, Greece
B.S. in Electrical, Electronics and Communications Engineering, GPA: 8.26/10.00 — 2009–2013

Experience

École Polytechnique Fédérale de Lausanne — Lausanne, Switzerland
PhD. Candidate - LTS2 Laboratory — 2015- Current

École Polytechnique Fédérale de Lausanne — Lausanne, Switzerland
Master Thesis Student/LTS4 Laboratory — September 2014-September2015

- Dictionary Learning For Graph Signals

Marie Curie Initial Training Network Workshops — Edinburgh,Lausanne,Surrey
Trainee — 2015-2018 (3 years)

- Took part in a number of PhD related workshops and seminars as part of the SpaRTaN Training Network | Project number - 607290 funded under the FP7-PEOPLE-2013-ITN of the **Marie Curie Actions**— Initial Training Networks

Noiseless Imaging — Tampere, Finland
Visiting Researcher — June 2018-July 2018 (2 months)

- Collaborated with scientist from Noiseless Imaging, on a number of signal processing problems.

IDCOM Laboratory - The University of Edinburgh — Edinburgh, United Kingdom
Visiting Researcher, Advisor: Mike Davies — September 2017-November 2017 (3 months)

Institute of Automation, Chinese Academy of Sciences — Beijing, China
Master Student, Advisor: Tan Tieniu — August 2013 - September 2013 (1 month)

- Semester Project on Dictionary Learning for Face Clustering

Publications

[1] K. Pitas, "Differential privacy and pac-bayes", *preprint*, 2020.

[2] K. Pitas, "Dissecting non-vacuous generalization boundsbased on the mean-field approximation", *Proceedings of the 37th International Conference on Machine learning*, 2020.

[3] K. Pitas, A. Loukas, M. Davies, and P. Vandergheynst, "Some limitations of norm based generalization bounds in deep neural networks", *Workshop on Machine Learning with Guarantees Neurips 2019 arXiv:1905.09677*, 2019.

Teaching

- **Teaching Assistant** at EPFL 2015- 2019
  Signals and Systems I (MICRO-310)
- **Teaching Assistant** at EPFL 2015- 2019
  Signals and Systems II (MICRO-311)

Skills

- **Programming:** Python,Matlab,C
- **Machine Learning:** Tensorflow,Keras,PyCharm

Languages

- **Greek:** Native Speaker
- **English:** Full Working Proficiency
- **French:** Limited Working Proficiency
- **Chinese:** Elementary Level

Scholarships and Awards

- M.S. in Electrical, Electronics and Communications Engineering with "Merrit" - Aristotle University of Thessaloniki 2015
- Master Thesis Scholarship, École Polytechnique Fédérale de Lausanne 2014
- Highschool Class Valedictorian - Anatolia College of Thessaloniki 2009

Extracurricular Activities

- Lindyhop Lausanne 2018–Current
  Lindyhop Lausanne is an association giving courses in a variety of dances such as Lindy Hop and Solo Jazz.
- Sports –
  Climbing, Snowboarding, Hiking
- Arts –
  I occasionally enjoy painting using acrylics.