communications physics

ARTICLE

Check for updates

https://doi.org/10.1038/s42005-023-01416-5 OPEN

The autoregressive neural network architecture of the Boltzmann distribution of pairwise interacting spins systems

Indaco Biazzo[1]✉

Autoregressive Neural Networks (ARNNs) have shown exceptional results in generation tasks across image, language, and scientific domains. Despite their success, ARNN architectures often operate as black boxes without a clear connection to underlying physics or statistical models. This research derives an exact mapping of the Boltzmann distribution of binary pairwise interacting systems in autoregressive form. The parameters of the ARNN are directly related to the Hamiltonian's couplings and external fields, and commonly used structures like residual connections and recurrent architecture emerge from the derivation. This explicit formulation leverages statistical physics techniques to derive ARNNs for specific systems. Using the Curie–Weiss and Sherrington–Kirkpatrick models as examples, the proposed architectures show superior performance in replicating the associated Boltzmann distributions compared to commonly used designs. The findings foster a deeper connection between physical systems and neural network design, paving the way for tailored architectures and providing a physical lens to interpret existing ones.

[1] Institute of Physics, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland. ✉email: indaco.biazzo@epfl.ch

The cross-fertilization between machine learning and statistical physics, in particular of disordered systems, has a long history[1,2]. Recently, the development of deep neural network frameworks[3] have been applied to statistical physics problems[4] spanning a wide range of domains, including quantum mechanics[5,6], classical statistical physics[7,8], chemical and biological physics[9,10]. On the other hand, techniques borrowed from statistical physics have been used to shed light on the behavior of Machine Learning algorithms[11,12], and even to suggest training or architecture frameworks[13,14]. In recent years, the introduction of deep generative autoregressive models[15,16], like transformers[17], has been a breakthrough in the field, generating images and text with a quality comparable to human-generated ones[18]. The introduction of deep Autoregressive Neural Networks (ARNNs) was motivated as a flexible and general approach to sampling from a probability distribution learned from data[19–21].

In classical statistical physics, the ARNN was introduced, in a variational setting, to sample from a Boltzmann distribution (or equivalently an energy-based model[22]) as an improvement over the standard variational approach relying on the high expressiveness of the ARNNs[8].

Then similar approaches have been used in different contexts, and domains of classical[23–27] and quantum statistical physics[28–34]. The ability of ARNNs to efficiently generate samples, thanks to the ancestral sampling procedure, opened the way to overcome the slowdown of Monte Carlo methods for frustrated or complex systems, although two recent works questioned the real gain in very frustrated systems[35,36].

The use of ARNNs in statistical physics problems has largely relied on pre-existing neural network architectures which may not be well-suited for the particular problem at hand. This approach has been largely favored due to the high expressive capacity of ARNNs, which can encapsulate the complexity of the Boltzmann probability distribution, remapped in an autoregressive form, within their parameters that, typically, grow polynomially with system size. To encode this complexity exactly, however, one might expect the need for an exponentially large number of parameters.

This work aims to demonstrate how knowledge of the physics model can inform the design of more effective ARNN architectures. I will present the derivation of an ARNN architecture that encodes exactly the classical Boltzmann distribution associated with a general pairwise interaction Hamiltonian of binary variables. The resulting architecture has the first layer's parameters, which scale polynomially with the system size, fixed by the Hamiltonian parameters. The analytic derivation leads to the emergence of both residual connections and recurrent structures. As expected for the exact architecture of the general case, the resulting deep ARNN architecture has the number of hidden layer parameters scaling exponentially with the system's size. In the general case, it is possible to approximate these hidden layers with feed-forward neural network structures containing a polynomial number of free parameters. The advantage of this approach over existing architectures is that the first layer's parameters can be fixed by the Hamiltonian, reducing the number of parameters to be learned and trained. For instance, the proposed architecture could be used in accelerating Markov chain simulations[23,24].

The quality of the approximation of the Boltzmann distribution relies on both the architecture of the feed-forward neural network used and the complexity of the problem being tackled. However, the physical interpretation of the architecture allows us to leverage problem-specific knowledge to develop specific feed-forward neural network architectures. As an example, standard statistical physics techniques will be used in the following to find feasible ARNN architecture for specific Hamiltonian. To showcase the potential of the derived representation, the ARNN architectures for two well-known mean-field models are derived: the Curie–Weiss (CW) and the Sherrington–Kirkpatrick (SK) models. These fully connected models are chosen due to their paradigmatic role in the history of statistical physics systems.

The CW model, despite its straightforward Hamiltonian, was one of the first models explaining the behavior of ferromagnet systems, displaying a second-order phase transition[37].

The SK model[38] is a fully connected spin glass model of disordered magnetic materials. The system admits an analytical solution in the thermodynamic limit, Parisi's celebrated[39] k-step replica symmetric breaking (k-RSB) solution[40,41]. The complex many-valley landscape of the Boltzmann probability distribution captured by the k-RSB solution of the SK model is the key concept that unifies the description of many different problems, and similar replica computations are applied to very different domains like neural networks[42,43], optimizations[44], inference problems[11], or in characterizing the jamming of hard spheres[45,46].

Thanks to the explicit autoregressive representation of the Boltzmann distribution, an exact ARNN architecture at finite N and an approximated architecture in the thermodynamic limit for the Curie–Weiss model are presented. Both have a number of parameters scaling polynomially with the system's size. Moreover, an ARNN architecture of the Boltzmann distribution of the SK model for a single instance of disorder with a finite number of variables will be shown. The derivation will be based on the k-RSB solution, resulting in a deep ARNN architecture with parameters scaling polynomially with the system size. The proposed architectures exhibit enhanced performance in sampling the Boltzmann distribution of the associated models compared to standard architectures in the literature. This work strengthens the connection between physical systems and neural network design, offering a way to devise tailored architectures and a physical perspective interpretation of existing neural network architecture.

Results and discussion

Autoregressive architecture of the Boltzmann distribution of pairwise interacting systems. The Boltzmann probability distribution of a given Hamiltonian $H[\mathbf{x}]$ of a set of N binary spin variables $\mathbf{x} = (x_1, x_2, \ldots x_N)$ at inverse temperature β is $P_B(\mathbf{x}) = \frac{e^{-\beta H(\mathbf{x})}}{Z}$, where $Z = \sum_{\mathbf{x}} e^{-\beta H(\mathbf{x})}$ is the normalization factor. It is generally challenging to compute marginals and average quantities when N is large and in particular, generate samples on frustrated systems. By defining the sets of variables $\mathbf{x}_{<i} = \left(x_1, x_2 \ldots x_{i-1}\right)$ and $\mathbf{x}_{>i} = \left(x_{i+1}, x_{i+2} \ldots x_N\right)$ respectively with an index smaller and larger than i, then if we can rewrite the Boltzmann distribution in the autoregressive form: $P_B(\mathbf{x}) = \prod_i P\left(x_i|\mathbf{x}_{<i}\right)$, it becomes straightforward to produce independent samples from it, thanks to the ancestral sampling procedure[8]. It has been proposed[8] to use a variational approach to approximate the Boltzmann distribution with trial autoregressive probability distributions where each conditional probability is represented by a feed-forward neural network with a set of parameters θ, $Q^{\theta}(\mathbf{x}) = \prod_i Q^{\theta_i}\left(x_i|\mathbf{x}_{<i}\right)$.

The parameters θ can be learned minimizing the variational free energy of the system:

$$F[P] = \sum_{\{\mathbf{x}\}} P[\mathbf{x}]\left[\frac{1}{\beta}\log P[\mathbf{x}] + H[\mathbf{x}]\right]. \tag{1}$$

Minimizing the variational free energy $F[Q^{\theta}]$ with respect to the parameters of the ARNN is equivalent to minimizing Kullback–Leibler divergence with the Boltzmann distribution as the target[8]. The computation of $F[Q^{\theta}]$ and their derivatives with respect to the ARNN's parameters involve a summation overall the configurations of the systems, that grows exponentially with the system's size, making it unfeasible after a few spins. In practice, they are estimated summing over a subset of configurations sampled directly from the ARNN thanks to the ancestral sampling procedure[8]. Beyond the minimization procedure, the selection of the neural network architecture is crucial for accurately approximating the Boltzmann distribution.

In the parameterization $Q^{\theta_i}(x_i = 1|\mathbf{x}_{<i})$ of the single variable conditional probability distribution $P(x_i = 1|\mathbf{x}_{<i})$ as a feed-forward neural network, the set of variables $\mathbf{x}_{<i}$ is the input, and a nested set of linear transformations, and non-linear activation functions is applied on them. Usually, the last layer is a sigma function $\sigma(x) = \frac{1}{1+e^{-x}}$, assuring the output is between 0 and 1. The set of parameters θ_i are the weights and biases of the linear transformations. Then, the probability $Q^{\theta_i}(x_i = -1|\mathbf{x}_{<i}) = 1 - Q^{\theta_i}(x_i = 1|\mathbf{x}_{<i})$ is straightforward to obtain. In the following, I will rewrite the single variable conditional probability of the Boltzmann distribution as a feed-forward neural network.

The generic i-th conditional probability factor of the Boltzmann distribution can be rewritten in this form:

$$P(x_i|\mathbf{x}_{<i}) = \frac{P(\mathbf{x}_{<i+1})}{P(\mathbf{x}_{<i})} = \frac{\sum_{\mathbf{x}_{>i}} P(\mathbf{x})}{\sum_{\mathbf{x}_{>i-1}} P(\mathbf{x})} = \frac{\sum_{\mathbf{x}_{>i}} e^{-\beta H}}{\sum_{\mathbf{x}_{>i-1}} e^{-\beta H}} = \frac{f(x_i,\mathbf{x}_{<i})}{\sum_{x_i} f(x_i,\mathbf{x}_{<i})}. \tag{2}$$

where I defined:

$$f(x_i = \pm 1, \mathbf{x}_{<i}) = \sum_{\mathbf{x}_{>i}} e^{-\beta H} \delta_{x_i,\pm 1}. \tag{3}$$

The $\delta_{a,b}$ is the Kronecker delta function that is one when the two values (a, b) coincide and zero otherwise. Now, imposing to have as the last activation function a sigma function, with simple algebraic manipulations, we obtain:

$$\begin{aligned} P(x_i = 1|\mathbf{x}_{<i}) &= \frac{f(1,\mathbf{x}_{<i})}{f(1,\mathbf{x}_{<i}) + f(-1,\mathbf{x}_{<i})} = \frac{1}{1 + \frac{f(-1,\mathbf{x}_{<i})}{f(1,\mathbf{x}_{<i})}} \\ &= \sigma\left(\log\left[f(1,\mathbf{x}_{<i})\right] - \log\left[f(-1,\mathbf{x}_{<i})\right]\right) \end{aligned} \tag{4}$$

Consider a generic two-body interaction Hamiltonian of binary spin variables $x_i \in \{-1, 1\}$, $H = -\sum_{i<j} J_{ij} x_i x_j - \sum_i h_i x_i$, where the sets of J_{ij} are the interaction couplings and h_i are the external fields. Taking into account a generic variable x_i the elements of the Hamiltonian can be grouped into the following five sets:

$$\begin{aligned} H_{ss} &= -\sum_{s,s'<i} J_{ss'} x_s x_{s'} - \sum_{s<i} h_s x_s \\ H_{si}[x_i = \pm 1] &= \mp H_{si} = \mp(\sum_{s<i} J_{si} x_s + h_i) \\ H_{il}[x_i = \pm 1] &= \mp H_{il} = \mp \sum_{l>i} J_{il} x_l \\ H_{sl} &= -\sum_{s<i,l>i} J_{sl} x_s x_l \\ H_{ll} &= -\sum_{l,l'>i} J_{ll'} x_l x_{l'} - \sum_{l>i} h_l x_l \end{aligned}$$

where the dependence on the variable x_i has been made explicit. Substituting these expressions in Eq. (4), we obtain:

$$P(x_i = 1|\mathbf{x}_{<i}) = \sigma\left(2\beta H_{si}[\mathbf{x}_{<i}] + \log(\bar{\rho}_i^+[\mathbf{x}_{<i}]) - \log(\bar{\rho}_i^-[\mathbf{x}_{<i}])\right), \tag{5}$$

where:

$$\bar{\rho}_i^{\pm}[\mathbf{x}_{<i}] = \sum_{\mathbf{x}_{>i}} e^{-\beta(\pm H_{il} + H_{sl}[\mathbf{x}_{<i}] + H_{ll})} \tag{6}$$

The set of elements H_{ss} cancels out.

The conditional probability, Eq. (5), can be interpreted as a feed-forward neural network, following, starting from the input, the operation done on the variables $\mathbf{x}_{<i}$. The first operation on the input is a linear transformation. Defining:

$$x_i^1 = 2\beta H_{si} = 2\beta(\sum_{s=1}^{i-1} J_{si} x_s + h_i), \tag{7}$$

$$x_{il}^1 = \sum_{s=1}^{i-1} J_{sl} x_s, \tag{8}$$

as outputs of the first layer (see Fig. 1), we can write the conditional probability as a feed-forward neural network:

$$P_i(x_i = 1|\mathbf{x}_{<i}) = \sigma\left(x_i^1 + \log \rho_i^+ - \log \rho_i^-\right) \tag{9}$$

$$\rho_i^{\pm} = \sum_c e^{b_c^{\pm} + \sum_{l=i+1}^{N} w_{cl} x_{il}^1} \tag{10}$$

As shown in Fig. 1, a second linear transformation acts on the set of x_{il}^1 variables. The parameters of the second layer are

$$b_c^{\pm} = \beta \sum_{l=i+1}^{N} (\pm J_{il} + h_l + \sum_{l'=l+1}^{N} J_{ll'} x_{l'}^c) x_l^c \tag{11}$$

$$w_{cl} = \beta x_l^c, \tag{12}$$

where c is the index of the configuration of the set of $\mathbf{x}_{>i}$ variables. This second linear transformation compute the 2^{N-i} possible values of the exponent in the $\rho_i^{\pm}$ functions, Eq. (10). Next, the two functions $\rho_i^{\pm}$ are obtained by first applying the exponential function to the output of the second layer. Then, for each of $\rho_i^{\pm}$, we sum their elements and finally apply the logarithmic function. As the last layer, the values $\log \rho_i^{\pm}$ and x_i^1 are combined with the right signs, and the sigma function is applied. The entire ARNN architecture of the Boltzmann distribution of the general pairwise interaction Hamiltonian (H$_2$ARNN) is depicted in Fig. 1. The total number of parameters scales exponentially with the system size, making its direct application infeasible for the sampling process. Nevertheless, the H$_2$ARNN architecture shows the following features:

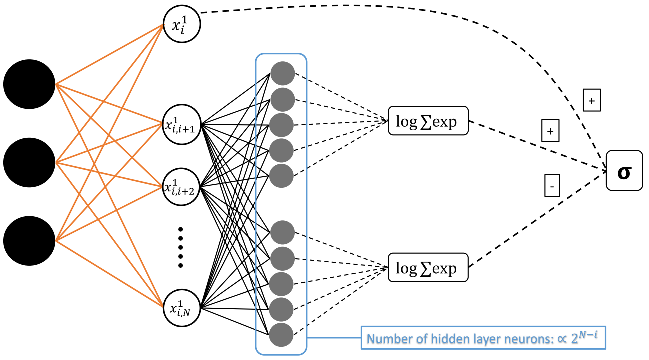


Fig. 1 The autoregressive neural network architecture of pairwise interaction Hamiltonian. The diagram shows the autoregressive neural network architecture of a single Boltzmann conditional probability of a pairwise interaction Hamiltonian, H$_2$ARNN, Eq. (9). The $x_{<i}$ variables are the input, the output provides an estimation of the conditional probability $P(x_i = 1|\mathbf{x}_{<\mathbf{i}})$. The first layer computes the x_i^1 and x_{il}^1 variables, see Eq. (7), where the weight and bias, directly related to the Hamiltonian parameters, are shown in orange. The non-linear operators are represented by square symbols. The width of the second layer increases exponentially with the system size. The $\log\sum\exp(\mathbf{x}) = \log\sum_i e^{x_i}$ represents the set of linear transformations and non-linear activation functions acting on the second layer. The last layer is the sigma function.

- The weights and biases of the first layer are the parameters of the Hamiltonian of the Boltzmann distribution.
- Residual connections among layers, due to the x_i^1 variables, naturally emerge from the derivation. The importance of residual connections has recently been highlighted[47] and has become a crucial element in the success of the ResNet and transformer architectures[48], in classification and generation tasks. They were presented as a way to improve the training of deep neural networks avoiding the exploding and vanishing gradient problem. In this context, they represent the direct interactions among the variable x_i and all the previous variables $\mathbf{x}_{<i}$.
- The H_2ARNN exhibits a recurrent structure[3,49]. The first layer, as seen in Fig. 1, is composed of a set of linear transformations (see eq. (7) and (8)). The set of $x_{il}^1 = \sum_{s=1}^{i-1} J_{si} x_s$ variables, can be rewritten in recursive form observing that:
$$x_{il}^1 = x_{i-1,l}^1 + J_{i-1,l} x_{i-1} \tag{13}$$
The output of the first layer of the conditional probability of the variable i depends on the output of the first layer, $x_{i-1,l}^1$, of the previous $i-1$ conditional probability. In practice, we can explicitly write the following dependence: $P(x_i = 1|\mathbf{x}_{<i}) = P(x_i = 1|\mathbf{x}_{<i}, x_{i-1,i+1}^1, \ldots x_{i-1,N}^1)$. The recurrent structure can reduce the number of parameters of the neural network and its total computational cost if efficiently implemented.

The most computationally demanding part of the H_2ARNN architecture is the computation of the $\rho_i^{\pm}$ functions, Eq. (10); their parameters scale exponentially with the system size, proportionally to 2^{N-i}. However, generally, the $\rho_i^{\pm}$ functions can be approximated using standard feed-forward neural network structures, possessing a polynomial number of parameters. Here, the input variables are those of the first layer $(x_{i,i+1}^1, \ldots x_{i,N}^1)$, while the parameters of the first layer remain unchanged, maintaining the skip connection. Instead of exploring this possibility, I will show how to derive ARNN architectures for specific systems. In fact, the $\rho_i^{\pm}$ function can be interpreted as the partition function of a system, where the variables are the $\mathbf{x}_{>i}$ and the external fields are determined by the values of the variables $\mathbf{x}_{<i}$. Based on this observation, in Methods, I will show how to use standard tools of statistical physics to derive deep ARNN architectures that eliminate the exponential growth of the number of parameters.

Computational results. In this section, various ARNN architectures are compared for their ability to generate samples from the Boltzmann distribution of the CW and SK models. Additionally, the correlation between the Hamiltonian couplings and the first layer parameters of the derived neural networks, trained on Monte Carlo-generated instances, will be shown. The CW_N, CW_∞ and $\mathrm{SK}_{\mathrm{RS/kRSB}}$ architectures, derived in the Methods section, are compared with:

- The one parameter (1P) architecture, where a single weight parameter is multiplied by the sums of the input variables, and then the sigma function is applied. This architecture was already used for the CW system in[36]. The total number of parameters scales as N.
- The single layer (1L) architecture, where a fully connected single linear layer parametrizes the whole probability distribution, where a mask is applied to a subset of the weights in order to preserve the autoregressive properties. The width of the layer is N, and the total number of parameters scale as N^2[15].
- The MADE architecture[15], where the whole probability distribution is represented with a deep sequence of fully connected layers, with non-linear activation functions and masks in between them, to assure the autoregressive properties. Compared to 1L, MADE offers greater expressive power at the expense of higher computational and parameter costs. The MADE_{dc} used has d hidden layers, each of them with c channels of width N. For instance, the 1L architecture is equivalent to the MADE_{11} and MADE_{23} has two hidden fully connected layers, each of them composed of three channels of width N.

The parameters of the ARNN are trained to minimize the Kullback–Leibler divergence or, equivalently, the variational free energy (see Eq. (1)). Given an ARNN, Q^θ, that depends on a set of parameters θ and the Hamiltonian of the system H, the variational free energy can be estimated as:

$$\begin{aligned} F[Q^\theta] &= \sum_{\{\mathbf{x}\}} Q^\theta \left[\frac{1}{\beta} \log Q^\theta + H[\mathbf{x}]\right] \\ &\approx \sum_{\mathbf{x}\sim Q^\theta} \left[\frac{1}{\beta} \log Q^\theta + H[\mathbf{x}]\right]. \end{aligned}$$

The samples are drawn from the trial ARNN, Q^θ, using ancestral sampling. At each step of the training, the derivative of the variational free energy with respect to the parameters θ is estimated and used to update the parameters of the ARNN. Then a new batch of samples is extracted from the ARNN and used again to compute the derivative of the variational free energy and update the parameters[8]. This process was repeated until a stop criterion is met or a maximum number of steps is reached. For each model and temperature, a maximum 1000 epochs are allowed, with a batch size of 2000 samples, and a learning rate of 0.001. The ADAM algorithm[50] was applied for the optimization of the ARNN parameters. An annealing procedure was used to improve performance and avoid mode-collapse problems[8], where the inverse temperature β was increased from 0.1 to 2.0 in steps of 0.05. The code was developed with the PyTorch framework[51] and has been made publicly available on GitHub[52]. The CW_N has all its parameters fixed and precomputed analytically, see Eq. (18). The CW_∞ has one free parameter for each of its conditional probability distributions to be trained, and one shared parameter, see Eq. (21). The parameters of the first layer of the $\mathrm{SK}_{\mathrm{RS/kRSB}}$ architecture are shared and fixed by the values of the couplings and fields of the Hamiltonian. The parameters of the hidden layers are free and trained. The parameters of the MADE_{dc}, 1L and 1P architectures are free and trained. The variational free energy $F[Q^\theta]$ is always an upper bound of the free energy of the system. Its value will be used, in the following, as a benchmark for the performance of the ARNN architecture in approximating the Boltzmann distribution. After the training procedure, the variational free energy was estimated using 20,000 configurations sampled from each of the considered ARNN architectures. The training procedure was the same for all the experiments unless conversely specified.

The results on the CW model, with Hamiltonian parameters $J=1$ and $h=0$ (see Eq. (14)), are shown in Fig. 2. The panels a, b, and c, in the first row, show the relative error of the free energy density ($fe[P] = F[P]/N$), with respect to the exact one, computed analytically[37], see the Supplementary Note 2 for details, for different system sizes N. The variational free energy density estimated from samples generated with the CW_N architecture does not have an appreciable difference with the analytic solution, and for the CW_∞, it improves as the system size increases. The panel d in Fig. 2 plots the error, in the estimation of the free energy density for the architectures with fewer parameters, 1P

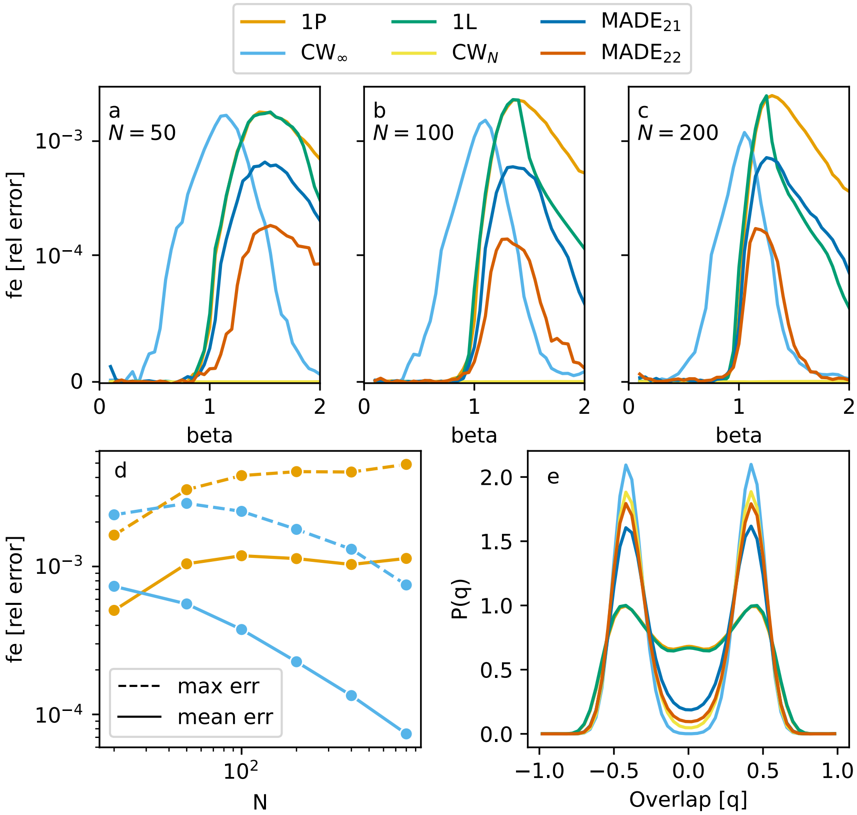


Fig. 2 Results for Curie-Weiss (CW) model. The CW model considered has $J = 1$ and $h = 0$ (see eq. (14)). The system undergoes a second-order phase transition at $\beta = 1$ where a spontaneous magnetization appears[37]. Six different architectures, the 1P, CW_∞, 1L, CW_N, $MADE_{21}$, $MADE_{22}$, are represented in the panels in the figure in, respectively, orange (or orange-circle), light-blue (or light-blue-circle), green, yellow, blue and red. **a–c** Relative error in the estimation of the free energy for different system sizes with respect to the analytic solution. The CW_N architecture has its parameters fixed and precomputed analytically, and the error is too small to be seen at this scale. The y-axis is plotted on a logarithmic scale down to 10^{-4} and then linearly to zero. **d** The dependence on N of the mean and maximum relative error of the two smaller architectures, 1P and CW_∞, both of which scale linearly with the size of the system. **e** Distribution of the overlaps of the samples generated by the ARNNs for the CW system with $N = 200$ variables and $\beta = 1.3$.

and CW_∞ (both scaling linearly with the system's size); It shows clearly that a deep architecture with skip connections, in this case with only one more parameter, in the skip connection, improves the accuracy by orders of magnitude. The need for deep architectures, already on a simple model as the CW, is indicated by the poor performance of the 1L architecture, despite its scaling of parameters as N^2, achieving similar results to the 1P. The MADE architecture obtained good results but was not comparable to CW_N, even though it has a similar number of parameters. The panel e in Fig. 2 shows the distribution of the overlaps, $q_{\mathbf{a},\mathbf{b}} = \frac{1}{N}\sum_i a_i b_i$ where a_i, b_i are two system configurations, between the samples generated by the ARNNs. The distribution is computed at $\beta = 1.3$ for $N = 200$. It can be seen that the poor performance of the 1-layer networks (1P, 1L) is due to the difficulty of correctly representing the configurations with magnetization different from zero in the proximity of the phase transition. This could be due to mode-collapse problems[36], which do not affect the deeper ARNN architectures tested.

In Fig. 3, the results of the SK model, with $J = 1$ and $h = 0$ (see the Hamiltonian definition Eq. (22)) are shown; as before, in panels a, b, and c in the first row, there is the relative error in the estimation of the free energy density at different system sizes. In this case, the exact solution, for a single instance of the disorder and a finite N is not known. The free energy estimation of the SK_{2RSB} was taken as the reference to compute the relative difference. The free energy estimations of SK_{kRSB} with $k = 1, 2$ are very close to each other. The performance of the SK_{RS} net is the same as the 1L architecture even with a much higher number of parameters. The MADE architecture tested, even with a similar number of parameters of the SK_{kRSB} nets, see panel d of Fig. 3, estimate a larger free energy, with differences increasing with N. To better assess the difference in the approximation of the Boltzmann distribution of the architecture tested, I consider to check the distributions of the overlaps q among the generated samples. The SK model, with $J = 1$ and $h = 0$, undergoes a phase transition at $\beta = 1$, where a glassy phase is formed, and an exponential number of metastable states appears[53]. This fact is reflected in the distribution of overlaps that have values different from zero in a wide region of values of q[54]. Observing the distribution of the overlaps in the glassy phase, $\beta = 1.3$, between the samples generated by the ARNNs, panel e in Fig. 3, we can check as the distribution generated by the SK_{kRSB} is higher in the region between the peak and zero overlaps, suggesting that these architectures better capture the complex landscape of the SK Boltzmann probability distribution[54].

The final test for the derived SK_{kRSB} architectures involves assessing the correlation between Hamiltonian couplings and the parameters of the first layers. This is done without fixing these parameters and by using only samples extracted from the Boltzmann distribution of a single instance of the SK model in the glassy phase at $\beta = 2$. The Metropolis Monte Carlo algorithm was used to sample, every 200 Monte Carlo sweeps, 10,000 system configurations. The SK_{1RSB} was trained to minimize the log-likelihood computed on these samples (see Supplementary Note 4 for details). According to the derivation of the SK_{kRSB} architecture, the weights of the first layer of the neural network

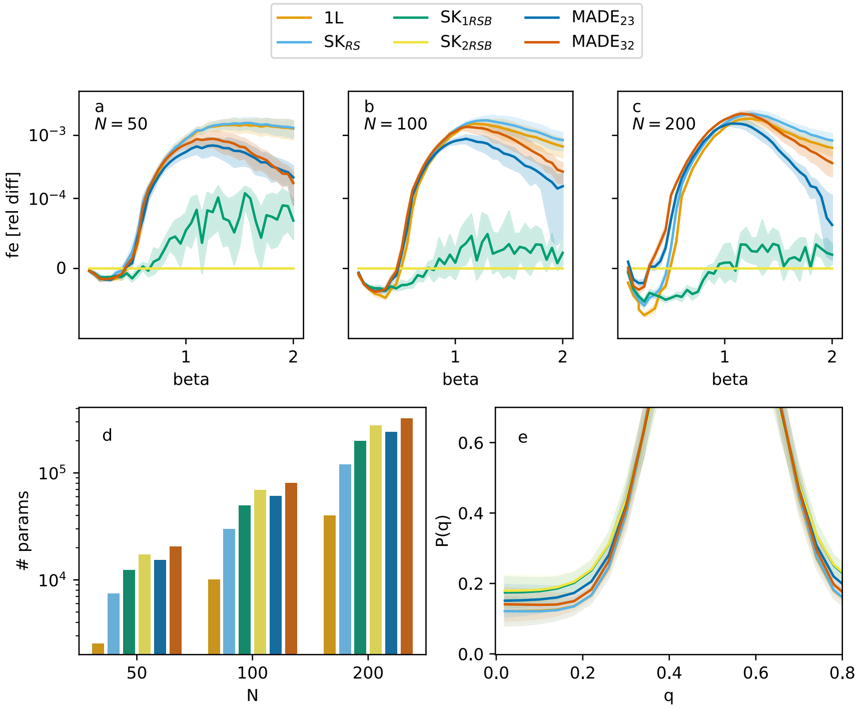


Fig. 3 Results for Sherrington–Kirkpatrick (SK) model. The SK model considered has $J = 1$ and $h = 0$ (see the Hamiltonian definition Eq. (22)). The system undergoes a phase transition at $\beta = 1$[53]. Six different architectures, the 1L, SK$_{RS}$, SK$_{1RSB}$, SK$_{2RSB}$, MADE$_{23}$, MADE$_{32}$, are represented in the panels in the figure in, respectively, orange, light-blue, green, yellow, blue, and red. The translucent error bands surrounding the plotted lines represent the 95% confidence intervals. **a–c** Relative difference in the estimation of the free energy for increasing system sizes with respect to the free energy computed by SK$_{2RSB}$ architecture. The results are averaged over 10 instances of the disorder. The y-axis is plotted on a logarithmic scale down to 10^{-4} and then linearly to -10^4. **d** Scaling with N of the number of parameters of the autoregressive neural network (ARNN) architectures. **e** Distribution of the overlaps of the samples generated by the ARNN architectures for the SK model with $N = 200$ variables and $\beta = 1.5$, averaged over 10 different instances.

should correspond to the coupling parameters of the Hamiltonian. Due to the gauge invariance of the Hamiltonian with respect to the change of sign of all the couplings Js, I will consider their absolute values in the comparison. The weight parameters of the first layers of the SK$_{1RSB}$ were initialized at small random values. As shown in Fig. 4, there is a strong correlation between the weights of the first layer and the couplings of the Hamiltonian, even though the neural network was trained in an over-parametrized setting; it has 60,000 parameters, significantly more than the number of samples.

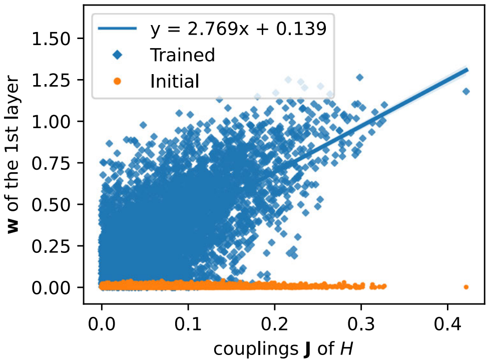


Fig. 4 Scatter plot of the weights vs the couplings. Scatter plot of the absolute values of weights of the first layer of a SK$_{1RSB}$ vs the absolute values of the coupling parameters of the Sherrington–Kirkpatrick (SK) model. The weights are trained over 10,000 samples generated by the Metropolis Monte Carlo algorithm on a single instance of the SK model with $N = 100$ variables at $\beta = 2$. They are initialized at small random values. The blue line is the fit of the blue points, clearly showing a strong correlation between the weights and the coupling parameters of the Hamiltonian. The Pearson coefficient is 0.64 with p-values of 0.0.

Conclusions

In this study, the exact autoregressive neural network architecture (H$_2$ARNN) of the Boltzmann distribution of the pairwise interaction Hamiltonian was derived. The H$_2$ARNN is a deep neural network, with the weights and biases of the first layer corresponding to the couplings and external fields of the Hamiltonian, see eqs. (7) and (8). The H$_2$ARNN architecture has skip connections and a recurrent structure. Although the H$_2$ARNN is not directly usable due to the exponential increase in the number of hidden layer parameters with the size of the system, its explicit formulation allows using statistical physics techniques to derive tractable architectures for specific problems. For example, ARNN architectures, scaling polynomially with the system's size, are derived for the CW and SK models. In the case of the SK model, the derivation is based on the sequence of k-step replica symmetric breaking solutions, which were mapped to a sequence of deeper ARNNs architectures.

The results, checking the ability of the ARNN architecture to learn the Boltzmann distribution of the CW and SK models, indicate that the derived architectures outperform commonly used ARNNs. Furthermore, the close connection between the physics of the problem and the neural network architecture is shown in the results of Fig. 4. In this case, the SK$_{1RSB}$ architecture was trained on samples generated with the Monte Carlo technique from the Boltzmann distribution of an SK model; the weights of the first layer of the $\mathrm{SK}_{1\mathrm{RSB}}$ were found to have a strong correlation with the coupling parameters of the Hamiltonian.

Even though the derivation of a simple and compact ARNN architecture is not always feasible for all types of pairwise interactions and exactly solvable physics systems are rare, the explicit analytic form of the H_2ARNN provides a means to derive approximate architectures for specific Boltzmann distributions.

In this work, while the ARNN architecture of an SK model was derived, its learnability was not thoroughly examined. The problem of finding the configurations of minimum energy for the SK model is known to belong to the NP-hard class, and the effectiveness of the ARNN approach in solving this problem is still uncertain and a matter of ongoing research[27,35,36]. Further systematic studies are needed to fully understand the learnability of the ARNN architecture presented in this work at very low temperatures and also on different systems.

There are several promising directions for future research to expand upon presented ARNN architectures. For instance, deriving the architecture for statistical models with more than binary variables. In statistical physics, the models with variables that have more than two states are called generalize Potts models. The probabilistic model learned by modern generative language systems, where each variable represents a word, and could take values among a huge number of states, usually more than tens of thousands possible words (or states), belong to this set of systems. The generalization of the present work to Potts models could allow us to connect the physics of the problem to recent language generative models like the transformer architecture[55]. Another direction could be to consider systems with interactions beyond pairwise, to describe more complex probability distributions. Additionally, it would be interesting to examine sparse interacting system graphs, such as systems that interact on grids or random sparse graphs. The first case is fundamental for a large class of physics systems and image generation tasks, while the latter type, such as Erdos–Renyi interaction graphs, is common in optimization[44] and inference problems[56].

Methods

Derivation of ARNN architecture for specific models. In the following subsection, the derivation of ARNN architectures for the CW and SK models is shown.

ARNN architectures of the Curie–Weiss model. The Curie–Weiss model (CW) is a uniform, fully connected Ising model. The Hamiltonian, with N spins, is:

$$H(\mathbf{x}) = -h\sum_{i=1}^{N} x_i - \frac{J}{N}\sum_{i<j} x_i x_j. \tag{14}$$

The conditional probability of a spin i, Eq. (5), of the CW model is:

$$\begin{aligned} &P^{CW}(x_i = 1|\mathbf{x}_{<i}) \\ &= \sigma\left(2\beta h + 2\beta\frac{J}{N}\sum_{s=1}^{i-1} x_s + \log(\rho_i^+[\mathbf{x}_{<i}]) - \log(\rho_i^-[\mathbf{x}_{<i}])\right), \end{aligned} \tag{15}$$

where:

$$\rho_i^{\pm}[\mathbf{x}_{<i}] \propto \sum_{\mathbf{x}_{>i}} e^{\beta\left(h \pm \frac{J}{N} + \frac{J}{N}\sum_{s<i} x_s\right)\sum_{l>i} x_l + \frac{\beta J}{2N}(\sum_{l,l'>i} x_l x_{l'})} \tag{16}$$

Defining $h_i^{\pm}[\mathbf{x}_{<i}] = h \pm \frac{J}{N} + \frac{J}{N}\sum_{s=1}^{i-1} x_s$, at given $\mathbf{x}_{<i}$, Eq. (16) is equivalent to the partition function of a CW model, with $N-i$ spins and external fields $h_i^{\pm}$. As shown in Supplementary Note 1, the summations over $\mathbf{x}_{>\mathbf{i}}$ can be easily done, finding the following expression:

$$\rho_i^{\pm}[\mathbf{x}_{<i}] = \sum_{k=0}^{N-i} e^{b_{ik}^{\pm} + w_{ik}\sum_s x_s} \tag{17}$$

where we defined:

$$b_{ik}^{\pm} = \log\binom{N-i}{k} + \frac{\beta J}{2N}(N-i-2k)^2 + (N-i-2k)\left(\beta h \pm \frac{\beta J}{N}\right) \tag{18}$$

$$\omega_{ik} = \frac{\beta J}{N}(N-i-2k). \tag{19}$$

The final feed-forward architecture of the Curie–Weiss Autoregressive Neural Network (CW_N) architecture is:

$$\begin{aligned} &P^{CW_N}(x_i = +1|\mathbf{x}_{<i}) \\ &= \sigma\left[b + \omega\sum_{s=1}^{i-1} x_s + \log\left(\sum_{k=0}^{N-i} e^{b_{ik}^{+} + w_{ik}\sum_{s=1}^{i-1} x_s}\right) - \log\left(\sum_{k=0}^{N-i} e^{b_{ik}^{-} + w_{ik}\sum_{s=1}^{i-1} x_s}\right)\right], \end{aligned} \tag{20}$$

where $b = 2\beta h$, $\omega = \frac{2\beta J}{N}$ are the same, and so shared, among all the conditional probability functions, see Fig. 5. Their parameters have an analytic dependence on the parameters J and h of the Hamiltonian of the systems.

The number of parameters of a single conditional probability of the CW_N is $2 + 4(N-i)$, which decreases as i increases. The total number of parameters of the entire conditional probability distribution scales as $2N^2$.

If we consider the thermodynamical limit, $N \gg 1$, the ARNN architecture of the CW model, named CW_∞, simplifies (see

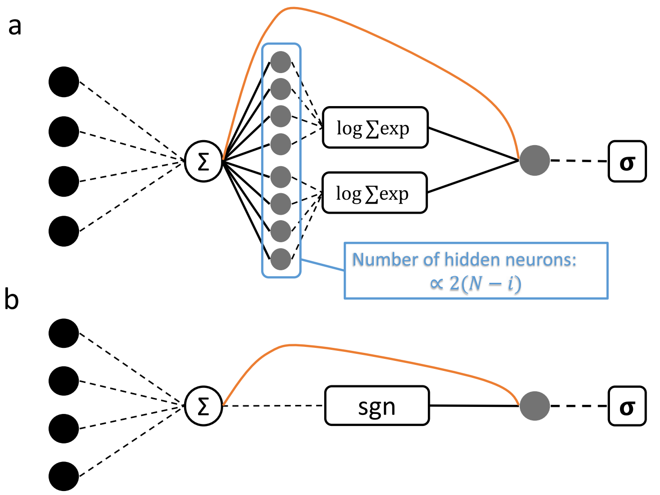


Fig. 5 $\mathbf{CW}_N$ and $\mathbf{CW}_\infty$ architectures of a single conditional probability. Diagrams **a** and **b** represent the CW_N and CW_∞ architectures, respectively. Both diagrams involve the operation of the sum of the input variables $\mathbf{x}_{<i}$. A skip connection, composed of a shared weight (represented by the orange line), is also present in both cases. In the CW_N architecture, $2(N-1)$ linear operations are applied (with fixed weights and biases, as indicated in Eq. (7)), followed by two non-linear operations represented by $\log\sum\exp(x)$. On the other hand, in the CW_∞ architecture, apart from the skip connection, the input variables undergo a *sgn* operation before being multiplied by a free weight parameter and passed through the final layer represented by the sigma function. The number of parameters in the CW_N architecture scales as $2N^2$, while in the CW_∞ architecture, it scales as N plus a shared parameter ω for the skip connection and a bias $b = 2\beta h$ different from zero only when the external field h is present.

Supplementary Note 1 for details) to the following expression:

$$P^{CW_\infty}(x_i = 1|\mathbf{x}_{<i}) = \sigma\left(b + \omega\sum_{s=1}^{i-1} x_s + \omega_i^1 \operatorname{sgn}\left(\sum_{s=1}^{i-1} x_s\right)\right) \quad (21)$$

where $b = 2\beta h$, $\omega = \frac{2\beta J}{N}$ are the same as before, and shared, among all the conditional probability functions, see Fig. 5. The $\omega_i^1 = -2\beta J|m_i|$ is different for each of them and can be computed analytically. The total number of parameters of the CW$_\infty$ scales as $N+2$.

ARNN architectures of the SK model. The SK Hamiltonian, considering zero external fields for simplicity, is given by:

$$H(\mathbf{x}) = -\sum_{i<j} J_{ij} x_i x_j \quad (22)$$

where the set of couplings, $\underline{J}$, are i.i.d. random variable drawn from a Gaussian probability distribution $P(J) = \mathcal{N}(0, J^2/N)$.

To find a feed-forward representation of the conditional probability of its Boltzmann distribution we have to compute the quantities in Eq. (10), that, defining $h_l^{\pm}[\mathbf{x}_{<i}] = \pm J_{il} + x_{il}^1[\mathbf{x}_{<i}]$, can be written as:

$$\rho_i^{\pm}[\mathbf{x}_{<i}] = \sum_{\mathbf{x}_{>i}} \exp\left(\beta \sum_{l=i+1}^{N} h_l^{\pm}[\mathbf{x}_{<i}] x_l + \sum_{l'>l>i}^{N} J_{ll'} x_l x_{l'}\right)$$

The above equation can be interpreted as an SK model over the variables $\mathbf{x}_{>i}$ with site-dependent external fields $h_l^{\pm}[\mathbf{x}_{<i}]$. I will use the replica trick[53], which is usually applied together with the average over the system's disorder. In our case, we deal with a single instance of disorder, with the set of couplings being fixed. In the following I will assume that $N - i \gg 1$, and the average over the disorder $\mathbb{E}$ is taken on the coupling parameters $J_{ll'}$ with $l, l' > i$. In practice, I will use the following approximation to compute the quantity:

$$\log\rho_i^{\pm} \sim \mathbb{E}\left[\log\rho_i^{\pm}\right] = \lim_{n\to 0}\frac{\log(\mathbb{E}\left[(\rho_i^{\pm})^n\right])}{n}$$

In the last equality, I use the replica trick. Implicitly, it is assumed that the quantities $\log\rho_i^{\pm}$ are self-averaged on the $\mathbf{x}_{>i}$ variables. The expression for the average over the disorder of the replicated function is:

$$\mathbb{E}_{\underline{J}_{ll'}}\left[(\rho_i^{\pm}[\mathbf{x}_{<i}])^n\right] = \int\prod_{l<l'} dP_{J_{ll'}}\left\{\sum_{\{\underline{x}^a\}_{i+1}^{N}} \exp\left[\beta\left(\sum_{\substack{i<l\le N\\ 1<a<n}} h_l^{\pm}[\mathbf{x}_{<i}] x_l^a + \sum_{\substack{i<l<l'\le N\\ 1<a\le n}} J_{ll'} x_l^a x_{l'}^a\right)\right]\right\} \quad (23)$$

where $dP_{J_{ll'}} = P(J_{ll'})dJ_{ll'}$, and the set of $\mathbf{x}^a$ are the replicated spin variables. Computing the integrals over the disorder, we find:

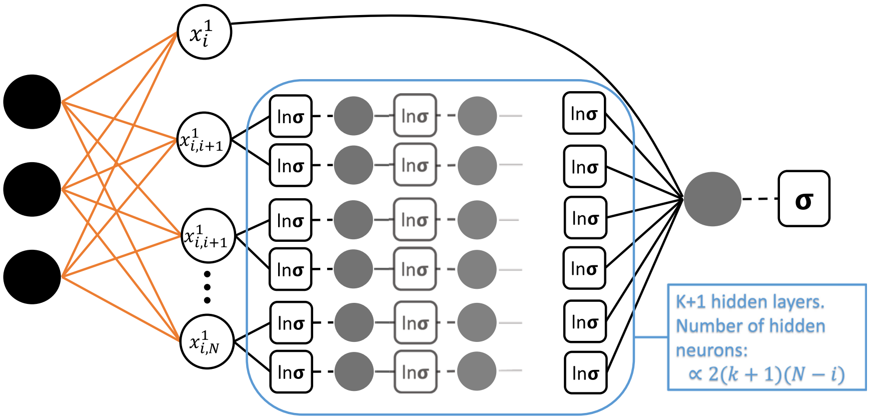
$$\mathbb{E}_{\underline{J}_{ll'}}\left[(\rho_i^{\pm}[\mathbf{x}_{<i}])^n\right] \propto \int\prod_{a<b} dQ_{ab} e^{-\frac{N}{2}\beta^2 Q_{a,b}^2}\prod_{l} \left[\sum_{\{\underline{x}_l^a\}} \exp\left\{\beta\left[h_l^{\pm}[\mathbf{x}_{<i}]\sum_a x_l^a + \beta\sum_{a<b} Q_{ab} x_l^a x_l^b\right]\right\}\right] \quad (24)$$

where in the last line I used the Hubbard–Stratonovich transformation to linearize the quadratic terms. See Supplementary Note 3 or, for instance,[57], for details about the formal mathematical derivations of the previous and following expressions. The Parisi solution of the SK model prescribes how to parametrize the matrix of the overlaps $\{Q_{ab}\}$[53]. The easiest way to parametrize the matrix of the overlaps is the replica symmetric solutions (RS), where the overlaps are equal and independent from the replica index:

$$Q_{ab} = \begin{cases} 0, & \text{if } a = b \\ q, & \text{otherwise} \end{cases},$$

A sequence of better approximations can then be obtained by breaking the replica symmetry step by step, from the 1-step replica symmetric breaking (1-RSB) to the k-step replica symmetric breaking (k-RSB) solution. The infinite k limit of the k-step replica symmetric breaking solution gives the exact solution of the SK model[58]. The sequence of k-RSB approximations can be seen as nested non-linear operations[59], see Supplementary Note 3 for details.

Every k-step replica symmetric breaking solution leads to adding a Gaussian integral and two more free variational parameters to the representation of the $\rho^{\pm}$ functions. In the following, I will use a feed-forward representation that enlarges the space of parameters, using a more computationally friendly



K+1 hidden layers. Number of hidden neurons: ∝ $2(k+1)(N-i)$

Fig. 6 SK$_{\text{RS/kRSB}}$ architectures of the single variable conditional probability. The diagram depicts the SK$_{\text{RS/kRSB}}$ architectures that approximate a single conditional probability of the Boltzmann distribution in the Sherrington–Kirkpatrick (SK) model. The input variables are $\mathbf{x}_{<i}$, and the output is the conditional probability $Q^{\text{RS/k-RSB}}(x_i = 1|\mathbf{x}_{<i})$. The non-linear operations are represented by squares and the linear operations by solid lines. The parameters, in the orange lines, are equal to the Hamiltonian parameters and shared among the conditional probabilities, as indicated in Eq. (7). The depth of the network is determined by the level of approximation used, with the Q^{RS} architecture having only one hidden layer and the $Q^{\text{k-SRB}}$ architecture having a sequence of $k+1$ hidden layers. The total number of parameters scales as $2(k+1)N^2 + \mathcal{O}(N)$, where the Replica Symmetric (RS) case corresponds to $k = 0$.

non-linear operator. Numerical evidence of the quality of the approximation used is shown in Supplementary Note 3. Overall, the parameterization of the overlaps matrix, which introduces free parameters in the derivation, allows the summing of all the configurations of the variables $\mathbf{x}_{i>}$ eliminating the exponential scaling with the system's size of the number of parameters. The final ARNN architecture of the SK model is as follows (see Supplementary Note 3 for details):

$$\begin{aligned} & Q^{\text{RS/k-RSB}}(x_i = 1|\mathbf{x}_{<i}) \\ & = \sigma\left(x_i^1(\mathbf{x}_{<i}) + \log\left(\rho_i^{+,\text{(RS/kRSB)}}\right) - \log\left(\rho_i^{-,\text{(RS/kRSB)}}\right)\right). \end{aligned} \tag{25}$$

For the RS and 1-RSB cases, we have:

$$\log\rho^{\pm,\text{RS}} = \sum_{l=i+1}^{N} w_{il}^{0\pm} \log\sigma\left(b_{il}^{1\pm} + w_{il}^{1\pm} x_{il}^1(\mathbf{x}_{<i})\right)$$

$$\log\rho^{\pm,\text{1RSB}} = \sum_{l=i+1}^{N} w_{il}^{0\pm} \log\sigma\left(b_{il}^{1\pm} + w_{il}^{1\pm} \log\sigma\left(b_{il}^{2\pm} + w_{il}^{2\pm} x_{il}^1(\mathbf{x}_{<i})\right)\right).$$

The set of $x_{il}^1(\mathbf{x}_{<i})$ is the output of the first layer of the ARNN, see eqs. (7)-(8), and $(w_{il}^{0\pm}, b_{il}^{1\pm}, w_{il}^{1\pm}, b_{il}^{2\pm}, w_{il}^{2\pm})$ are free variational parameters of the ARNN (see Fig. 6). The number of parameters of a single conditional probability distribution scales as $2(k+1)(N-i)$ where k is the level of the k-RSB solution used, assuming $k=0$ as the RS solution.

Data availability

The data used to produce the results of this study are generated by the code released on GitHub with https://zenodo.org/records/8383403[52].

Code availability

The code needed to reproduce the results is released on GitHub with https://zenodo.org/records/8383403[52].

Received: 13 April 2023; Accepted: 4 October 2023;
Published online: 14 October 2023

References

1. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl Acad. Sci. USA* **79**, 2554–2558 (1982).
2. Amit, D. J., Gutfreund, H. & Sompolinsky, H. Spin-glass models of neural networks. *Phys. Rev. A* **32**, 1007–1018 (1985).
3. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
4. Carleo, G. et al. Machine learning and the physical sciences. *Rev. Mod. Phys.* **91**, 045002 (2019).
5. Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017).
6. van Nieuwenburg, E. P. L., Liu, Y.-H. & Huber, S. D. Learning phase transitions by confusion. *Nat. Phys.* **13**, 435–439 (2017).
7. Carrasquilla, J. & Melko, R. G. Machine learning phases of matter. *Nat. Phys.* **13**, 431–434 (2017).
8. Wu, D., Wang, L. & Zhang, P. Solving statistical mechanics using variational autoregressive networks. *Phys. Rev. Lett.* **122**, 1–8 (2019).
9. Noé, F., Olsson, S., Köhler, J. & Wu, H. Boltzmann generators: sampling equilibrium states of many-body systems with deep learning. *Science* **365**, eaaw1147 (2019).
10. Jumper, J. et al. Highly accurate protein structure prediction with alphafold. *Nature* **596**, 583–589 (2021).
11. Zdeborová, L. & Krzakala, F. Statistical physics of inference: thresholds and algorithms. *Adv. Phys.* **65**, 453–552 (2016).
12. Nguyen, H. C., Zecchina, R. & Berg, J. Inverse statistical problems: from the inverse ising problem to data science. *Adv. Phys.* **66**, 197–261 (2017).
13. Chaudhari, P. et al. Entropy-SGD: biasing gradient descent into wide valleys*. *J. Stat. Mech. Theory Exp.* **2019**, 124018 (2019).
14. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N. & Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proc. 32nd International Conference on Machine Learning*, Vol. 37 of *Proc. Machine Learning Research* (eds. Bach, F. & Blei, D.) 2256–2265 (PMLR, Lille, France, 2015). https://proceedings.mlr.press/v37/sohl-dickstein15.html.
15. Germain, M., Gregor, K., Murray, I. & Larochelle, H. Made: Masked autoencoder for distribution estimation. In *Proc. 32nd International Conference on Machine Learning*, Vol. 37 of *Proc. Machine Learning Research* (eds. Bach, F. & Blei, D.) 881–889 (PMLR, Lille, France, 2015). https://proceedings.mlr.press/v37/germain15.html.
16. van den Oord, A. et al. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems*, Vol. 29 (eds. Lee, D., Sugiyama, M., Luxburg, U., Guyon, I. & Garnett, R.) (Curran Associates, Inc., 2016). https://proceedings.neurips.cc/paper/2016/file/b1301141feffabac455e1f90a7de2054-Paper.pdf.
17. Vaswani, A. et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, Vol. 30 (eds. Guyon, I. et al.) (Curran Associates, Inc., 2017). https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.
18. Brown, T. et al. Language models are few-shot learners. In *Advances in neural information processing systems*, (eds. Larochelle H. et al.) Vol. 33, (Curran Associates, Inc., 2020). https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.
19. Gregor, K., Danihelka, I., Mnih, A., Blundell, C. & Wierstra, D. Deep autoregressive networks. In *Proc. 31st International Conference on Machine Learning*, Vol. 32 of *Proc. Machine Learning Research* (eds. Xing, E. P. & Jebara, T.) 1242–1250 (PMLR, Bejing, China, 2014). https://proceedings.mlr.press/v32/gregor14.html.
20. Larochelle, H. & Murray, I. The neural autoregressive distribution estimator. In *Proc. 14th International Conference on Artificial Intelligence and Statistics*, Vol. 15 of *Proc. Machine Learning Research* (eds. Gordon, G., Dunson, D. & Dudík, M.) 29–37 (PMLR, Fort Lauderdale, FL, USA, 2011). https://proceedings.mlr.press/v15/larochelle11a.html.
21. van den Oord, A., Kalchbrenner, N. & Kavukcuoglu, K. Pixel recurrent neural networks. In *Proc. 33rd International Conference on Machine Learning*, Vol. 48 of *Proc. Machine Learning Research* (eds. Balcan, M. F. & Weinberger, K. Q.) 1747–1756 (PMLR, New York, New York, USA, 2016). https://proceedings.mlr.press/v48/oord16.html.
22. Nash, C. & Durkan, C. Autoregressive energy machines. In *Proc. 36th International Conference on Machine Learning*, Vol. 97 of *Proc. Machine Learning Research* (eds. Chaudhuri, K. & Salakhutdinov, R.) 1735–1744 (PMLR, 2019). https://proceedings.mlr.press/v97/durkan19a.html.
23. Nicoli, K. A. et al. Asymptotically unbiased estimation of physical observables with neural samplers. *Phys. Rev. E* **101**, 023304 (2020).
24. McNaughton, B., Milošević, M. V., Perali, A. & Pilati, S. Boosting Monte Carlo simulations of spin glasses using autoregressive neural networks. *Phys. Rev. E* **101**, 053312 (2020).
25. Pan, F., Zhou, P., Zhou, H.-J. & Zhang, P. Solving statistical mechanics on sparse graphs with feedback-set variational autoregressive networks. *Phys. Rev. E* **103**, 012103 (2021).
26. Wu, D., Rossi, R. & Carleo, G. Unbiased Monte Carlo cluster updates with autoregressive neural networks. *Phys. Rev. Res.* **3**, L042024 (2021).
27. Hibat-Allah, M., Inack, E. M., Wiersema, R., Melko, R. G. & Carrasquilla, J. Variational neural annealing. *Nat. Mach. Intell.* **3**, 1–10 (2021).
28. Luo, D., Chen, Z., Carrasquilla, J. & Clark, B. K. Autoregressive neural network for simulating open quantum systems via a probabilistic formulation. *Phys. Rev. Lett.* **128**, 090501 (2022).
29. Wang, Z. & Davis, E. J. Calculating Rényi entropies with neural autoregressive quantum states. *Phys. Rev. A* **102**, 062413 (2020).
30. Sharir, O., Levine, Y., Wies, N., Carleo, G. & Shashua, A. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Phys. Rev. Lett.* **124**, 020503 (2020).
31. Hibat-Allah, M., Ganahl, M., Hayward, L. E., Melko, R. G. & Carrasquilla, J. Recurrent neural network wave functions. *Phys. Rev. Res.* **2**, 023358 (2020).
32. Liu, J.-G., Mao, L., Zhang, P. & Wang, L. Solving quantum statistical mechanics with variational autoregressive networks and quantum circuits. *Mach. Learn. Sci. Technol.* **2**, 025011 (2021).
33. Barrett, T. D., Malyshev, A. & Lvovsky, A. I. Autoregressive neural-network wavefunctions for ab initio quantum chemistry. *Nat. Mach. Intell.* **4**, 351–358 (2022).
34. Cha, P. et al. Attention-based quantum tomography. *Mach. Learn. Sci. Technol.* **3**, 01LT01 (2021).
35. Inack, E. M., Morawetz, S. & Melko, R. G. Neural annealing and visualization of autoregressive neural networks in the newman-moore model. *Condens. Matter.* **7** https://www.mdpi.com/2410-3896/7/2/38 (2022).
36. Ciarella, Simone, et al. "Machine-learning-assisted Monte Carlo fails at sampling computationally hard problems." Machine Learning: Science and Technology 4.1 (2023): 010501.

37. Kadanoff, L. P. *Statistical physics: statics, dynamics and renormalization* (World Scientific, 2000).
38. Sherrington, D. & Kirkpatrick, S. Solvable model of a spin-glass. *Phys. Rev. Lett.* **35**, 1792–1796 (1975).
39. The Nobel Committee for Physics. For groundbreaking contributions to our understanding of complex physical systems. [Nobel to G. Parisi] https://www.nobelprize.org/prizes/physics/2021/advanced-information/ (2021).
40. Parisi, G. Toward a mean field theory for spin glasses. *Phys. Lett. A* **73**, 203–205 (1979).
41. Parisi, G. Infinite number of order parameters for spin-glasses. *Phys. Rev. Lett.* **43**, 1754–1756 (1979).
42. Gardner, E. Maximum storage capacity in neural networks. *Europhys. Lett.* **4**, 481 (1987).
43. Amit, D. J., Gutfreund, H. & Sompolinsky, H. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Phys. Rev. Lett.* **55**, 1530–1533 (1985).
44. Mézard, M., Parisi, G. & Zecchina, R. Analytic and algorithmic solution of random satisfiability problems. *Science* **297**, 812–815 (2002).
45. Parisi, G. & Zamponi, F. Mean-field theory of hard sphere glasses and jamming. *Rev. Mod. Phys.* **82**, 789–845 (2010).
46. Biazzo, I., Caltagirone, F., Parisi, G. & Zamponi, F. Theory of amorphous packings of binary mixtures of hard spheres. *Phys. Rev. Lett.* **102**, 195701 (2009).
47. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun; Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.
48. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 5998–6008 (2017).
49. Lipton, Z. C., Berkowitz, J. & Elkan, C. A critical review of recurrent neural networks for sequence learning. Preprint at https://arxiv.org/abs/1506.00019 (2015).
50. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2014).
51. Paszke, A. et al. Pytorch: an imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, Vol. 32 (eds. Wallach, H. et al.) (Curran Associates, Inc., 2019). https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.
52. Biazzo, I. h2arnn. GitHub repository. https://zenodo.org/records/8383403 (2023).
53. Mezard, M., Parisi, G. & Virasoro, M. *Spin Glass Theory and Beyond*. World Scientific Publishing Company (1986).
54. Young, A. P. Direct determination of the probability distribution for the spin-glass order parameter. *Phys. Rev. Lett.* **51**, 1206–1209 (1983).
55. Rende, R., Gerace, F., Laio, A. & Goldt, S. Optimal inference of a generalised potts model by single-layer transformers with factored attention. Preprint at https://arxiv.org/abs/2304.07235 (2023).
56. Biazzo, I., Braunstein, A., Dall'Asta, L. & Mazza, F. A Bayesian generative neural network framework for epidemic inference problems. *Sci. Rep.* **12**, 19673 (2022).
57. Nishimori, H.*Statistical Physics of Spin Glasses and Information Processing: an Introduction* (Clarendon Press, 2001).
58. Talagrand, M. The Parisi formula. *Ann. Math.* **163**, 221–263 (2006).
59. Parisi, G. A sequence of approximated solutions to the s-k model for spin glasses. *J. Phys. A Math. Gen.* **13**, L115 (1980).

Acknowledgements

I.B. thanks Giuseppe Carleo, Giovanni Catania, Vittorio Erba, Guido Uguzzoni, Martin Weigt, Francesco Zamponi, Lenka Zdeborová, and all the members of the SPOC-lab of the EPFL for useful discussions. I.B. thanks also Christian Keup for reading the manuscript and for his valuable comments.

Author contributions

I.B. conceived the idea, wrote the code, ran the simulations, analyzed the data, and wrote the manuscript.

Competing interests

The author declares no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42005-023-01416-5.

Correspondence and requests for materials should be addressed to Indaco Biazzo.

Peer review information *Communications Physics* thanks Aleksei Malyshev and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.