Meanfield theory of activation functions in Deep Neural Networks

Abstract

We present a Statistical Mechanics (SM) model of deep neural networks, connecting the energy-based and the feed forward networks (FFN) approach. We infer that FFN can be understood as performing three basic steps: *encoding*, *representation validation* and *propagation*. From the meanfield solution of the model, we obtain a set of natural activations – such as *sigmoid*, $\tanh$ and *ReLU* – together with the state-of-the-art, *Swish*; this represents the expected information propagating through the network and tends to *ReLU* in the limit of zero noise. We study the spectrum of the Hessian on an associated classification task, showing that *Swish* allows for more consistent performances over a wider range of network architectures.

1. Introduction

Advances in modern computing hardware and availability of massive datasets have empowered multilayer artificial neural networks, or deep learning (DL), with unprecedented capabilities for image and speech recognition tasks. Despite these empirical success, theoretical understanding of why and when multilayer neural networks perform well lags far behind (Mézard, 2018). Only recently, theoretical efforts in this direction have been intensively reported. For example, recent works shed light on how FFN attains its expressive power (Pogio et al., 2017; Lin et al., 2017; Raghu et al., 2016; Poole et al., 2016), what contributes to its generalizability (Zhang et al., 2017; Dinh et al., 2017), and how myriad parameters in the network affect the geometry of the loss function (Dauphin et al., 2014; Choromanska et al., 2015; Pennington & Bahri, 2017; Pennington & Worah, 2017). Taken together, these theoretical results have paved the way for a systematic design of robust and explainable FFNs.

Using modern optimization and regularization techniques such as dropout (Srivastava et al., 2014), non-linear activation functions, and complex network architectures (Krizhevsky et al., 2012), to name a few, the FFN can be efficiently trained on large-scale datasets such as ImageNet or CIFAR to achieve low training and generalization errors. While these engineering feats improve the performance of FFN, a clear design principle is still lacking, leading to an unsystematic growth of model complexity.

To assist future systematic studies and construction of FFNs, we propose a theoretical framework based on the tool-set of SM. It allows for the definition of an energy based model in which a hidden unit is regarded as a communication channel, first encoding and then *transmitting* the result of its computation through a gate with a specific transmission probability; the latter is obtained via the maximum entropy principle (Nguyen et al., 2017; Jaynes, 2003) under the biologically inspired constraint that a neuron responds by firing (transmit its signals) or not with a certain probability. By interpreting a hidden unit as a communication channel, its activation takes the form of the expected (mean) signal transmission; remarkably, this activation corresponds to *Swish*, obtained in (Elfwing et al., 2014; Ramachandran et al., 2017) through an extensive search algorithm trained with reinforcement learning and shown to best perform on the CIFAR and ImageNet datasets among all candidate activations. Although some activations may perform better for a specific datasets, they generally fail to generalize. Finally, the *ReLU* activation arises as a limiting, noiseless case of *Swish*. To the best of our knowledge, this provides the first derivation of *ReLU*, typically introduced heuristically to facilitate optimization. Despite restricting our analysis to pure FFNs, most of our conclusions carry on to Convolutional and Recurrent networks.

2. Motivation and Model

2.1. General Setup

A standard task in supervised learning is to determine an input/output relation between a set of m features and the observed labeled outcomes. Let us denote with x_i^μ the input vector, where $i \in [1, n]$ denotes features, and $\mu \in [1, m]$ denotes an example; we also denote the output as y_k^μ, where k is the number of classes. Quite generally, the probability of measuring the output $\mathbf{y}$ given the input can be written as $P(\mathbf{y}) = \int d\mathbf{x}\, P(\mathbf{y}|\mathbf{x})\, P(\mathbf{x}) = \int d\hat{\mathbf{y}}\, P(\mathbf{y}|\hat{\mathbf{y}})\, P(\hat{\mathbf{y}})$ where we have used the chain rule (A) and introduced the output layer ($\hat{\mathbf{y}}$). Once $P(\hat{\mathbf{y}})$ has been learned, one can obtain the loss function by taking the log-likelihood $\mathscr{L} = -\log P(\mathbf{y})$; for example, in a binary classification problem, $P(\hat{\mathbf{y}}[\boldsymbol{\theta}])$ is Bernoulli and parametrized by $\boldsymbol{\theta_l} = \{\mathbf{W}_l, \mathbf{b}_l\}$, being the weights and biases of the neural network, with $l \in [1, L]$ the

number of hidden layers. In this case, the loss function is the binary cross-entropy, but other statistical assumptions lead to different Losses; see (A) for a meanfield derivation of the cross-entropy. To motivate the model, in the next section we begin by discussing a (physical) dimensional inconsistency in the standard formulation of *forward propagation* in FFNs, and how this can be reconciled within our framework.

2.2. Motivations

Motivated by neurobiology and in analogy with the communication channel scheme of information theory (MacKay, 2003; Jaynes, 2003), we regard the input vector x_i^μ as the information source, while the units constitute the encoders. Quite generally, the encoders can either build a lower (compression) or higher dimensional (redundant) representation of the input data by means of a properly defined transition function. In a FFN, the former corresponds to a compression layer while the latter to an expansion layer. If the encoded information constitutes an informative representation of the output signal (given the input), it is passed over to the next layer for further processing until the output layer is reached. In the biological neuron, this last step is accomplished by the synaptic bouton, that releases information whether or not the input exceeds a local bias b_i. Both in the brain and in electronic devices, the input is often conveyed in the form of an electric signal, with the electron charge being the basic unit of information, the signal has dimension (units) of $Coulomb$ in SI. For an image, the information is the brightness level of each pixel and more in generale, information has units of bits. Clearly, a linear combination of the input signals needs to preserve dimensions: $h_i^\mu = \sum_{j=1}^n w_{ij} x_j^\mu + b_i$, where $i \in [1, n_1]$ indices the receiving units in the first layer and the weight matrix w_{ij} is the coefficient of the linear transformation and it is dimensionless. For noiseless systems, the input is transmitted i.f.f. the overall signal exceeds the bias b_i. However, in the presence of noise, the signal can be transmitted with a certain probability even below this threshold; in the biological neuron, the variance in the number of discharged vesicles in the synaptic bouton and the number of neurotransmitters in each vesicle is responsible for such noisy dynamics (Amit, 1989). Let us now consider the sigmoid non-linearity $\sigma(\beta\,\mathbf{h})$, where β has inverse dimension of $\mathbf{h}$; σ expresses the probability of the binary unit to be active (1) and can be seen as an approximation of the biological neuron's firing probability (Amit, 1989). Being a *distribution* defined in $[0, 1]$, σ is intrinsically *dimensionless*. The parameter β defines the spread of the distribution and one typidally sets it to 1 or reabsorb it inside the weights and bias (Nguyen et al., 2017); here we keep it general for reasons that will be clear later. Defining $\mathbf{a} = \sigma(\beta\,\mathbf{h})$ as the input of the next layer, we can immediately see the dimensional mismatch: a linear combination of $\mathbf{a}$ is dimensionless and when passed through the new non-linearity, $\sigma(\beta\,\mathbf{a})$ necessarily becomes dimensionful ($\beta\,\mathbf{a}$ now has dimension of inverse the dimension of $\mathbf{h}$). In the next section we show how this simple fact relates to gradients vanishing during back propagation. From a conceptual point of view, one is transmitting the expectation value of the transmission gate (synapse) rather than the processed signal. This problem persists with the $\tanh$ activation, but is resolved when using *ReLU*, that correctly transmits the information itself.

2.3. Statistical Mechanics of Feed Forward networks

A prototypical SM formulation of Neural Networks is the *inverse* Ising model (Nguyen et al., 2017), or Boltzmann machine, where one infers the values of the couplings and external fields given a spin configuration. While the Boltzmann machine is an energy-based model, a standard FFN is not commonly regarded as such. Here, we propose to bridge on these two formulations using the maximum entropy principle (Nguyen et al., 2017; Di-Castro & Raimondi, 2003; MacKay, 2003; Jaynes, 2003) to obtain the least biased representation of hidden neurons. Starting from the input layer, each unit takes the same input vector $\mathbf{x}$ and outputs a new variable $\mathbf{h}$. We now regard h_i as a coarse grained field coupling to the synaptic gate variable s_i. The feedforward nature of coarse graining (a directed graph), stems from its irreversibility and endorses the forward pass with a semi-group structure. Considering the first layer, we need to evaluate the probability associated to $\mathbf{h}$, $P(\mathbf{h}) = \int d\mathbf{x}\, Q(\mathbf{h}|\mathbf{x})\, P(\mathbf{x})$, where $Q(\mathbf{h}|\mathbf{x})$ is the transition function modeling the encoder. In DL, the latter is fixed by the forward pass, while the input data are drawn from some unknown distribution $P(\mathbf{x})$. We can follow two different paths: fix the value of $\mathbf{x}$ on the observed sequence, or assume the form of the distribution from which $\mathbf{x}$ has been sampled from; here we choose the former option. Consider then the empirical estimator $P(\mathbf{x}) = \prod_{\mu=1}^m \delta(\mathbf{x} - \mathbf{x}^\mu)$, where the Dirac-delta function fixes the input on the observed sequence. As for the transition function, it enforces the information processing performed by the "soma" of the artificial neuron; in DL, this consists of creating a linear combination of the input information. Information conservation enforces an additional constrain on the weights, $\sum_i w_{ij}^{[l]} = 1\,\forall\, l, j$, formally equivalent to the conservation of charge in physics and to an L_1 regularization in DL:

$$P(\mathbf{h}) = \int d\mathbf{x}\,\Gamma^{[1]}\,\delta\left(\mathbf{h} - \mathbf{w}^T\,\mathbf{x} - \mathbf{b}\right)\prod_{\mu=1}^{m}\delta(\mathbf{x} - \mathbf{x}^\mu) \quad (1)$$

$$\Gamma^{[1]} = \prod_{j=1}^{n}\delta\left(\sum_{i=1}^{n_1} w_{ij}^{[1]} - 1\right).$$

Eq. (1) is akin to the real space block-spin renormalization developed by Kadanoff, reformulated in the more general language of probability theory (Di-Castro & Raimondi, 2003; Ma, 1976; Cassandro & Jona-Lasinio, 1978) or it can

be seen as a functional "change of variables". The relation between DL and the renormalization group was previously observed in the context of restricted Boltzmann machine (RBM) (Mehta & Schwab, 2014); we discuss how this generalizes to FFN via Eq. (1). Once $\mathbf{h}$ has been computed, the information passes through the "synaptic" gate, and transmitted with a certain probability if it exceeds the threshold potential b_i. The core task at hand is to determine the state of the gate (open/close). The absence of lateral connections in FFNs means that each synaptic gate is only influenced by its receptive signal as the intralayer correlations are taken – a priori– to be zero. Given a statistical ensemble of hidden binary gates, the least unbiased distribution can be obtained by maximizing its entropy, subject to constraints imposed by the conserved quantities, i.e. the first and second moments of the data (Nguyen et al., 2017; MacKay, 2003). However, in the absence of lateral connections, the entropy functional of the hidden gate s_i does not account for the second moment. Functionally varying with respect to $P(\mathbf{s})$ and solving for the Lagrange multipliers, one obtains (Di-Castro & Raimondi, 2003):

$$P(\mathbf{s}|\mathbf{h}) = \frac{e^{\sum_i \beta_i\, s_i\, h_i}}{\prod_i\, (1 + e^{\beta_i\, h_i})} \tag{2}$$

where β_i encodes the noise. In a physical system, β_i is the inverse temperature in units of Boltzmann's constant and in equilibrium it is the same for each s_i; however, here the network is only in a local equilibrium as the units are not allowed to exchange "energy" (information) among themselves due to the lack of pairwise interactions. We have introduced $P(\mathbf{s}|\mathbf{h})$ to denote the conditional probability of $\mathbf{s}$ given the signal $\mathbf{h}$ – and the partition function Z. Finally, given the distribution of the coarse grained inputs and the conditional probability of the channels $P(\mathbf{s}|\mathbf{h})$, one is left evaluating the channel transmission probability

$$P(\mathbf{s}) = \int d\mathbf{h}\, P(\mathbf{s}|\mathbf{h})\, P(\mathbf{h}) = \frac{e^{-\sum_i\, \beta_i \mathscr{H}_i[s, x^\mu]}}{m^{-1} \prod_{\mu,i} Z_{i,\mu}} \tag{3}$$

where $\hat{h}_i^\mu = \sum_j w_{ij}^{[1]}\, x_j^\mu + b_i$, $Z_{i,\mu} = 1 + e^{\beta_i\, h_i^\mu}$ is the partition function per example/index and we have identified the coarse grained Hamiltonian $\mathscr{H}_i = -\sum_{j=1}^n s_i\, w_{ij}^{[1]} x_j^\mu - s_i\, b_i$. Eq. (3) is the starting point of an *energy based model*, where it is the coarse grained probability $P(\mathbf{s})$ that propagates through the network (see e.g. (Kou et al., 2018)) as opposed to signals in a FFN; indeed, the $\mathscr{H}$ has the form of a RBM with binary hidden units. The expected value of the channel transmission, $\langle s_i \rangle$ is (Hertz et al., 1991) the logistic function, for each empirical realization μ and unit i:

$$\langle s_i \rangle = {\beta_i}^{-1} \partial_{b_i} \log Z_{i,\mu} = \sigma(\beta_i\, \hat{h}_i^\mu). \tag{4}$$

This is the known mean-field solution of the Ising model; here this is exact due to the lack of lateral connections. We stress that this is not the coarse grained input signal *transmitted* by the channel; it solely determines the expectation of channel transmissions *given* the coarsed grained input h_i^μ. To ensure dimensional consistency across hidden layers, the output signal of each unit must have the same dimension as its input. Therefore, the correct quantity to consider is the *expectation value* of the output $j_i = h_i\, s_i$. This can be obtained by summing over all gate states, or by using the partition function of Eq. (3)

$$\langle j_i \rangle = \langle \hat{h}_i^\mu\, s_i \rangle_s = \partial_{\beta_i} \log Z_{i,\mu} = \hat{h}_i^\mu\, \sigma(\beta_i\, \hat{h}_i^\mu), \tag{5}$$

that agrees with the definition of the energy flux in SM (Bellac et al., 2004). Note that contrary to Eq. (4), the noise parameters β_i cannot be rescaled away. This function was recently obtained in (Ramachandran et al., 2017) (*Swish*), through an extensive search algorithm trained with reinforcement learning. In their extensive search, the authors found that activations of the form $a = x f(x)$ better performed on several benchmark datasets. A theoretical study of the performance of *Swish* from the point of view of information propagation has been proposed in (Hayou et al., 2018). Our model naturally complements these studies by identifying *Swish* with the expectation value of the coarse grained input transmitted by each unit. In the noiseless limit:

$$\lim_{\beta_i \to \infty} \langle j_i \rangle = \hat{h}_i^\mu\, \theta(\hat{h}_i^\mu) \equiv \max\left\{\hat{\mathrm{h}}_\mathrm{i}^\mu, 0\right\} = \mathrm{ReLU}(\hat{\mathrm{h}}_\mathrm{i}^\mu), \tag{6}$$

where $\theta(.)$ is the Heaviside step function and in the last equality we have identified *ReLU*. To the best of our knowledge, this is the first consistent derivation of *ReLU*, usually obtained from heuristics. *ReLU* emerges as the noiseless limit of the mean transmitted information across the units; as such, it is not affected by the dimensional mismatch, see Sec. (2.2). We would like to stress that both *Swish* and *ReLU* pass on the expected value of their computation; the latter does it with probability one *only* if the input signal exceeds a threshold, while the former can pass a signal lower than the threshold with a finite probability. A positive signal means that a certain combination of the inputs should be strengthen while a negative one means that it should be weakened, i.e. unlearned; the latter option is absent when using *ReLU*. In the opposite limit $\beta \ll 1$, *Swish* becomes linear; therefore, we can consider linear networks as a noisless limit of non linear ones. The effect of noise in FFNs was recently considered in Ref. (Chaudhari et al., 2017), where an optimization scheme based on the maximum entropy principle was proposed.

2.4. On Back-propagation

At the heart of any FFN model is the back propagation algorithm (Hertz et al., 1991; Bishop, 2006). Consider the

output layer "L", then the gradient of the weights is

$$\frac{\partial \mathscr{L}}{\partial \mathbf{w}_L} = \frac{1}{m}\sum_{\mu=1}^{m}\left[\mathbf{e}^{\mu}\,\mathbf{g}(\hat{\mathbf{h}}_L^{\mu})\right]\mathbf{a}_{L-1} \tag{7}$$

where $\mathbf{e}^{\mu} \propto y^{\mu} - \hat{y}^{\mu}$ is the residual error (R.E.) and $\hat{\mathbf{g}}(\mathbf{h}_l^{\mu}) = \boldsymbol{\sigma}(\boldsymbol{\beta}_l\,\hat{\mathbf{h}}_l^{\mu})[1 + \hat{\mathbf{h}}_l^{\mu}\boldsymbol{\beta}_l\,\boldsymbol{\sigma}(-\boldsymbol{\beta}_l\,\hat{\mathbf{h}}_l^{\mu})]$. In the optimization phase we look for stationary points defined by $\partial\mathscr{L}/\partial\boldsymbol{\theta}_l^{\alpha} = 0$, where $\boldsymbol{\theta}_l^{\alpha} = \{\mathbf{w}_l, \mathbf{b}_l, \boldsymbol{\beta}_l\}$. For a linear network, this condition is strictly satisfied if $\mathbf{e}^{\mu} = 0$. However, in a non linear network we can also have $g(\hat{\mathbf{h}}^{\mu}) = 0$. In principle, there may be situations in which $\mathbf{e}^{\mu}$ is far from zero but $g(\hat{\mathbf{h}}^{\mu}) \simeq 0$, in which case learning will not be effective. For a σ activation, $\mathbf{g}(\hat{\mathbf{h}}_l^{\mu}) = \boldsymbol{\sigma}(\boldsymbol{\beta}_l\,\hat{\mathbf{h}}_l^{\mu})\boldsymbol{\sigma}(-\boldsymbol{\beta}_l\,\hat{\mathbf{h}}_l^{\mu})$ [1]. (Tishby & Zaslavsky, 2015; Schwartz-Ziv & Tishby, 2017) show that there are two distinct phases of learning: empirical error minimization (the residual) and representation learning. We can identify the latter with the task of optimizing $g(\hat{\mathbf{h}}^{\mu})$. When $\hat{h}_i \gg 1/\beta_i$, i.e. when the signal greatly exceeds the noise, then $\sigma(\beta_i\,\hat{h}_i) \equiv P(s_i = 1|\bar{x}_i) \simeq 1$ ($\bar{x}_i$ is the unit's input) and the back propagated signal is small, being proportional to $g(\hat{h}) \simeq 0$. We then have the paradoxical situation in which, although the lower dimensional representation of the information is considered to be relevant, learning is likely to be inefficient.

Consider now the same limiting behavior for *Swish*. If $\hat{h}_i \gg 1/\beta_i$, $g(\hat{h}_i^{\mu}) \simeq 1$, i.e. the representation learning phase is completed and learning moves towards minimising the empirical error. In the opposite limit, the signal is much smaller than the noise and learning is impossible. Finally, in the noiseless case, one obtains the *ReLU* solution $g(\hat{h}_i^{\mu}) = (1, 0)$, for $\hat{h}_i$ respectively greater or smaller than zero. This corresponds to a network in which representation unlearning is impossible. The fact that *Swish* can be negative for $\sigma(\hat{h})/\beta < \hat{h} < 0$ allows to avoid plateaus surrounding local minima and saddle-points. In (Dauphin et al., 2014), it was proposed that loss plateaus of zero (or small) curvatures are responsible for preventing convergence of gradient descent.

3. Numerical Analysis

A thorough performance comparison of *Swish* versus other activations was presented in (Ramachandran et al., 2017), where it was shown that *Swish* outperformed all other choices on image classification tasks. Hereafter, we considered both artificial and experimental datasets. Given that both datasets lead to the same qualitative features, here we discuss the former and come back to the latter in (B.1). We considered two binary classification tasks: a linear and a non-linear one, each trained with one and two hidden layers. For the linear task with a single, 10 units layer, all three activations attain full train/test accuracy but *Swish* is the fastest converging. For the non-linear task, *ReLU* quickly converges to a suboptimal plateau. To obtain a better understanding, we have evaluated two different indices: the fraction of negative eigenvalues of the Hessian –α– and the fraction of zero eigenvalues – γ –. The former measures the ratio of descent to ascent directions on the energy landscape; when α is large, gradient descent can quickly escape a critical point due to the existence of multiple unstable directions. However, when a critical point exhibits multiple near-zero eigenvalues, roughly captured by γ, gradient descent will slowly decrease the training loss. In general, we find that for *ReLU* networks $\gamma \neq 0$, while this is typically not the case for both *Swish* and sigmoid-networks. Taking the two layer case as a representative example, we show that *ReLU* networks are sensitive to fine tuning of the model: choosing a $10-2$ or a $8-5$ configuration over the $8-2$ considered here, greatly improves learning. In stark contrast, *Swish* networks exhibit consistent performance over a wider choice of architecture/learning parameters. Although the performance impact might be fairly small for small networks, it certainly plays an important role for larger ones, as discussed in (Ramachandran et al., 2017). Looking at the fraction of residuals $e^{\mu} \simeq (\hat{y}^{\mu} - y^{\mu})/m$ closer to zero we found, surprisingly, that *Swish* greatly outperforms the other activations in minimizing the empirical error, see (B.1). In addition, we find that the eigenvalue distribution obtained with *ReLU shrinks* with increasing training epochs, giving rise to the singular distribution reported in (Pennington & Bahri, 2017; Sagun et al., 2017). In contrast, *Swish* shows a significantly wider spread in the eigenvalue distribution at the end of training, see sec. (B.1).

4. Conclusion and perspectives

In this work, we have introduced an energy-based framework that allows systematic construction of FFNs by providing a coherent interpretation of the computational process of each and every hidden unit. The framework overcomes the dimensional mismatch stemming from the use of heuristic activations. Enforcing dimensional consistency naturally leads to activations propagating the expectation value of the processed signals, providing a theoretical justification of the *Swish* activation found in (Ramachandran et al., 2017). We also demonstrate the superiority of *Swish* through numerical experiments that reveal the geometry of its loss manifold that facilitates gradient descent optimization. We hope SM methods can complement standard views of FFNs, e.g. help explore the scaling relation between the *Swish* Hessian's eigenvalue distributions and the hyperparameters, similar to the analysis of (Pennington & Bahri, 2017) for standard activations.

[1] in physics,the phase space factor (Di-Castro & Raimondi, 2003)

A. Cross Entropy Loss

In this appendix we present a derivation of the cross entropy loss function using large deviation methods (Mézard & Montanari, 2009). In sect. (2.1) we have introduced the supervised learning task in terms of the probability of the outcome y given the input x; this can be related to the network output by means of the chain rule:

$$P(\mathbf{y}) = \int d\mathbf{x}\, P(\mathbf{y}|\mathbf{x})\, P(\mathbf{x}) = \int d\hat{\mathbf{y}}\, P(\mathbf{y}|\hat{\mathbf{y}}) \int d\mathbf{j} P(\hat{\mathbf{y}}|\mathbf{j}) \times \int d\mathbf{x}\, P(\mathbf{j}|\mathbf{x})\, P(\mathbf{x}) = \int d\hat{\mathbf{y}}\, P(\mathbf{y}|\hat{\mathbf{y}})\, P(\hat{\mathbf{y}}), \quad (8)$$

where $P(\hat{\mathbf{y}})$ is the output probability of the network and $P(\mathbf{j}|\mathbf{x})$ is the transition probability between the input and hidden layer; additional hidden layers can be added by further use of the chain rule. The accuracy of a classification task – the number of times the network prediction $\hat{y}$ equals the provided solution y – corresponds to the conditional probability $P(y|\hat{y}) = \mathbb{I}(y = \hat{y})$. As $\hat{y}$ is a random variable, we need to find the probability of obtaining the true value y in a series of "experiments" in which $\hat{y}$ has a certain probability to be equal to y. According to Eq. (8), and using the standard relation relating the Loss function to the log-probability, we evaluate

$$-\log P(\mathbf{y}) = -\log \int d\hat{\mathbf{y}}\, P(\mathbf{y}|\hat{\mathbf{y}})\, P(\hat{\mathbf{y}}). \quad (9)$$

From a physics perspective, the above expression corresponds to an *annealed* average (Mézard et al., 1987; Dominicis & Giardina, 2016) and we will discuss its meaning at the end of the calculation. For the moment, we assume that we can replace $y \to y^\mu$ and $\hat{y} \to \hat{y}^\mu$ directly in Eq. (9), i.e. we fix the random variables on the observed data. Using the Dirac delta as a representation of the Indicator function we have:

$$P(y) = \int \left[\prod_\mu d\hat{y}^\mu\right] \prod_\mu \delta(y^\mu - \hat{y}^\mu)\, P(\hat{y}) = \int \left[\prod_\mu \frac{d\lambda^\mu}{2\pi}\right] \prod_\mu e^{i \sum_\mu \lambda^\mu y^\mu} \chi(\lambda^\mu), \quad (10)$$

where we have introduced m Lagrange multipliers λ to enforce the δ-function constraint and identified the characteristic function

$$\chi(\lambda^\mu) = \int \left[\prod_\mu d\hat{y}^\mu\right] P(\hat{y}^\mu)\, e^{-i \sum_\mu \lambda^\mu \hat{y}^\mu} \quad (11)$$

Let us consider the case of i.i.d. examples, distributed according to the Bernoulli distribution:

$$P(\hat{y}^\mu) = \int d\boldsymbol{\theta}\, \{q^\mu(\boldsymbol{\theta})\, \delta(\hat{y}^\mu - 1) + (1 - q^\mu(\boldsymbol{\theta}))\, \delta(\hat{y}^\mu)\}, \quad (12)$$

where each outcome is either 1 or 0 with a *sample dependent* probability $q^\mu(\boldsymbol{\theta})$ being the output value of a Neural Network solving a binary classification task; as such, q has the functional form of a sigmoid function with $\boldsymbol{\theta}$ a set of network parameters. Eq. (13) then reads

$$\chi(\lambda^\mu) = \int d\boldsymbol{\theta} \left[q^\mu(\boldsymbol{\theta}) e^{-i\,\lambda^\mu} + (1 - q^\mu(\boldsymbol{\theta}))\right]. \quad (13)$$

Using this expression back in Eq. (12) we arrive at the intermediate result

$$P(y) = \int d\boldsymbol{\theta} \left[\prod_\mu \frac{d\lambda^\mu}{2\pi}\right] e^{i \sum_\mu (\lambda^\mu y^\mu + \log\left[q^\mu e^{-i\,\lambda^\mu} + (1 - q^\mu)\right])} = \int \left[\prod_\mu \frac{d\lambda^\mu}{2\pi}\right] e^{m\, S[\lambda]}. \quad (14)$$

For a large number of training examples m, we can solve the above integral using the steepest descent. This fixes the value of the Lagrange multipliers:

$$\frac{\partial S[\lambda]}{\partial \lambda^\mu} = i\, y^\mu - \frac{i\, q^\mu\, e^{-i\,\lambda^\mu}}{q^\mu\, e^{-i\,\lambda^\mu} + (1 - q^\mu)} = 0 \quad (15)$$
$$\to -i\lambda_c^\mu = \log \frac{y^\mu (1 - q^\mu)}{q^\mu (1 - y^\mu)}$$

Using the optima back in Eq. (14) we arrive after some simple algebra to the expression for the cross-entropy:

$$P(y) \simeq \int d\boldsymbol{\theta}\, e^{m S[\boldsymbol{\theta}]} \quad (16)$$
$$S[\boldsymbol{\theta}] = \frac{1}{m} \sum_\mu \{y^\mu\, \log[q^\mu(\boldsymbol{\theta})] + (1 - y^\mu)\, \log[1 - q^\mu(\boldsymbol{\theta})]\},$$

up to an additive constant depending only on y. The final step consists in evaluating the θ integral again for $m \gg 1$, and take the maximum likelihood value θ^*. This last step is numerically performed by the gradient descent algorithm. Note however that the evaluation of the cost function in the optima negelects "finite size" correcions but it becomes more correct as the number of examples m increases.

Previously, we noticed that from a statistical mechanics point of view we are evaluating an *annealed* average; however, as the input variables are fixed this should rather be an *quenched* average. This is the case of spin glasses (Mézard et al., 1987; Dominicis & Giardina, 2016), where the quenched disorder (a fixed value of the couplings between spins) leads to frustration and therefore a highly non-convex energy landscape. Interestingly, a quenched scenario is similar to the approach proposed in (Chaudhari et al., 2017), leading to an improved optimization protocol. We leave this point as an open question for future research.

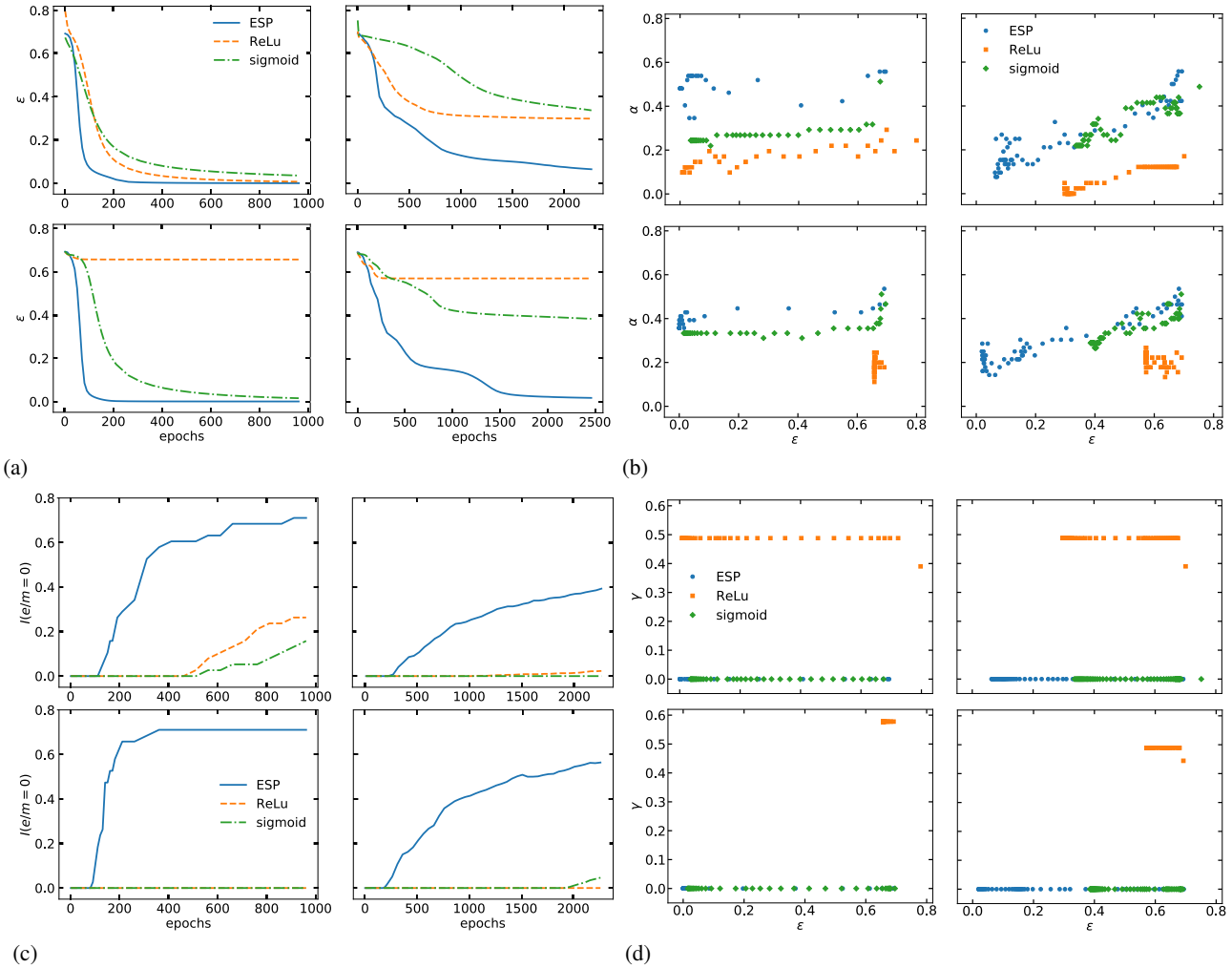


Figure 1. (a) Loss functions for linear (left) and non-linear (right) binary classification. (Top) 10 units hidden layer (Bottom) two hidden layer of 8 and 2 units respectively. All cases have a sigmoid activation in the last layer and a learning rate of 0.01. (b) α index v.s. energy for the single layer network with linear (left) and non-linear (right) decision boundary. (c) Fraction of zero residuals as a function of training epochs for the single 8-2 layer network with linear (top/bottom left) and non-linear (top/bottom right) decision boundary. (d) Fraction of zero eigenvalues. (Top left/right) linear/non-linear dataset with a single, 10 units hidden layer. (Bottom left/right) linear/non-linear dataset with two hidden layers of 8 and 2 units respectively. Curves are evaluated with the same number of epochs within a single batch training.

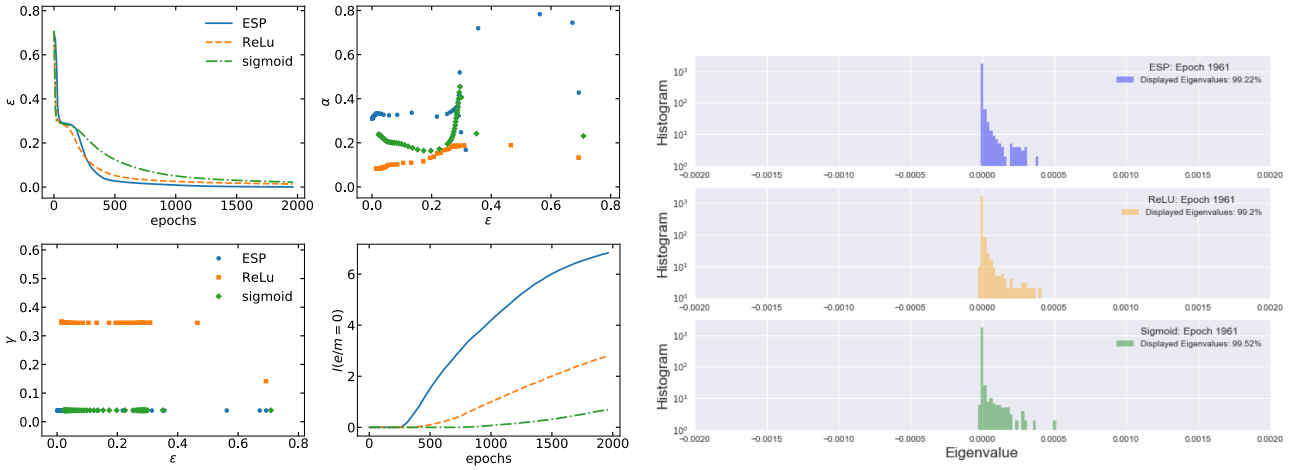
B. Numerical Analysis

B.1. Index of critical points

In section (3) we have qualitatively described the behaviour of the two indices α and γ on two classification tasks, a linear and a non-linear one. Here we report the results of this analysis more in details while dataset and codes are available as a git repository.

The results of figure (1) have been obtained using Adam for batch training in order to distinguish the effect of noise introduced by Swish from the one due to random minibatches. Also notice that the two dataset are fairly small: the linearly separable one contains 51 examples while the non-linear one has 863 examples.

To verify that the observed effects are not specific to small datasets, we have performed the same analysis on the MNIST dataset, comprising 1797 examples and obtained a qualitatively similar result. Further results and analysis, together with a visualization of the time evolution of eigenvalue distribution during training, can be found in the git repository. In the next section, we complete this numerical study with an analysis of the Hessian eigenvectors.



(a) (b)

Figure 2. Single hidden layer network (25 units) trained on the MNIST dataset. (top left) Loss functions, (top right) α index, (bottom left) Fraction of zero eigenvalues γ and (bottom right) fraction of zero residuals. (b) eigenvalue distribution at the end of training

B.2. Hessian eigenvectors

In this section, we describe the nature of the Hessian eigenvectors. [definisci la matrice qui con analogia fisica]. In Fig. (3).b, the Hessian eigenvectors vs the network variables [this should be variables' "indices", not the variables themselves] $\boldsymbol{\theta}$ are plotted for Relu, Sigmoid and Swish trained networks (see the linear binary classification with 8-2 hidden layer units in Fig. (1) bottom left). For Relu, the loss function minimization is almost immediately reached; however, this corresponds to the Hessian eigenvectors still coinciding with the initial values of the $\boldsymbol{\theta}$; in the dynamical matrix [is "dynamical matrix" a common terminology? have we introduced this before?], this situation corresponds to a concentration of eigenvectors intensity along the diagonal. [how about "this situation corresponds to localization of eigenvectors' amplitudes along the diagonal"?] In the Sigmoid case, the Hessian eigenvectors spread more uniformly, exploring the off-diagonal elements of the dynamical matrix and giving rise to a block form [this is not a block diagonal form right? if so, saying "block form" can be confusing] . Nevertheless, there is still a concentration of high-valued coefficients in the two sub-matrices [what are the two sub-matrices?]. Lastly, in the Swish case, the Hessian [eigenvectors'] coefficients spread almost uniformly over the dynamical matrix. This effect is akin to a regularization effect, disfavoring large values of individual eigenvectors while preventing overfitting. In Fig. (B.2), the same analysis is repeated for the non-linear classification task. Whereas the Relu result still exhibits a localization of the eigenvectors' [amplitudes], the Hessian eigenvectors of Sigmoid and Swish are more extended, spreading over a larger number of variables' indices. Nevertheless, in the Swish case, the number of large-valued coefficients is still less than that in the Sigmoid case.

[This parts needs some citations] We argue that networks trained with *Swish* activations can more efficiently break the symmetries of the loss function and reach extremal points that have higher number of zero residuals (in Physical terms, lower energy minima). This is obtained by preventing large values of the individual components in the Hessian eigenvectors-variables transformation matrices at the minima. In Fig.(3) , the Hessian eigenvectors vs the individual variables are plotted in the case of Relu, Sigmoid and Swish networks (see the linear binary classification with two 8-2 hidden layer of 8 and 2 units Fig. (1) bottom left loss function minimization).

References

Amit, D. J. *Modeling Brain Functions, The world of acctractor Neural Networks*. Cambridge University Press, 1989.

Bellac, M. L., Mortessagne, F., and Bastrouni, G. G. *Equilibrium and Non-Equilibrium Statistical Thermodynamics*. Cambridge University Press, 2004.

Bishop, C. *Pattern recognition and machine learning*. Springer, 2006.

Cassandro, M. and Jona-Lasinio, G. Critical point behaviour and probability theory. *Advances in Physics*, 27(6):913 – 941, 1978.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *arXiv:1611:01838*, 2017.

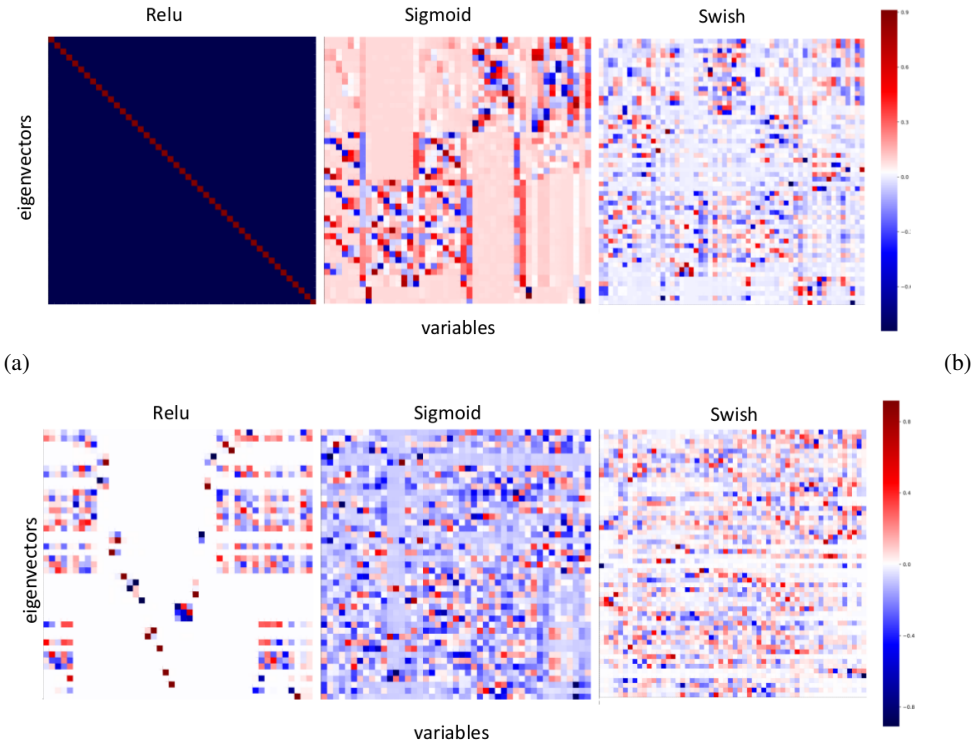


Figure 3. Hessian eigenvectors for the two hidden layer network trained on the linear (top) and non-linear (bottom) classification task. The bar indicates the value of the eigenvector coefficients.

Choromanska, A., Henaff, M., Mathieu, M., Michael, A., Gerard, B., and LeCun, Y. The loss surface of multilayer networks. *JMLR*, 38, 2015.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., and Bengio., Y. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems*, 27:2933 – 2941, 2014.

Di-Castro, C. and Raimondi, R. *Statistical mechanics and applications in condensed matter*. Cambridge University Press, 2003.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, J. Sharp minima can generalize for deep nets. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Dominicis, C. D. and Giardina, I. *Random Fields and Spin Glasees, A Field Theory approach*. Cambridge University Press, 2016.

Elfwing, S., Uchibe, E., and Doya, K. Expected energy-based restricted boltzmann machine for classification. *Neural Networks*, 64:29–38, 2014. doi: http://dx.doi.org/10.1016/j.neunet.2014.09.006.

Hayou, S., Doucet, A., and Rousseau, J. On the selection of initialization and activation function for deep neural networks. *arXiv:1805.08266v1*, 2018.

Hertz, J., Krogh, A., and Palmer, R. G. *Introduction to the Theory of Neural Computation*. Introduction to the theory of Neural computation, 1991.

Jaynes, E. T. *Probability Theory, the logic of science*. Cambridge University Press, 2003.

Kou, C., Lee, H. K., and Ng, T. K. Distribution regression networks. *arXiv:1409.6179v1*, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Lin, H. W., Tegmark, M., and Rolnick, D. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168:1223–1247, 2017.

Ma, S.-K. *Modern Theory of Critical Phenomena*. Advanced book program. Westview press, 1976.

MacKay, D. J. *Information Theory, Inference and Learning algorithms*. Cambridge University Press, 2003.

Mehta, P. and Schwab, D. J. An exact mapping between the variational renormalization group and deep learning. *arXiv:1410.3831*, 2014.

Mézard, M. Artificial intelligence and its limits. *Europhysics News*, 49, 2018.

Mézard, M. and Montanari, A. *Information, Physics and Computation*. Oxford University Press, 2009.

Mézard, M., Parisi, G., and Virasoro, M. A. *Spin Glass Theory and Beyond*, volume 9 of *Lecture notes in Physics*. World Scientific, 1987.

Nguyen, H. C., Zecchina, R., and Berg, J. Inverse statistical problems: from the inverse ising problem to data science. *Advances in Physics*, pp. 197–261, 2017.

Pennington, J. and Bahri, Y. Geometry of neural network loss surfaces via random matrix theory. *Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia*, 2017.

Pennington, J. and Worah, P. Nonlinear random matrix theory for deep learning. *31st Conference on Neural Information Processing Systems*, 2017.

Pogio, T., Mhaskar, H., Rosasco, L., Miranda, B., and Liao., Q. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14:503–519, 2017.

Poole, B., Lahiri, S., Raghu, M., Sohl-Dickstein, J., and Ganguli, S. Exponential expressivity in deep neural networks through transient chaos. *Advances in Neural Information Processing Systems*, 29:3360 – 3368, 2016.

Raghu, M., Poole, B., Kleinberg, J., Ganguli, S., and Sohl-Dickstein., J. On the expressive power of deep neural networks. *Proceedings of the 34th International Conference on Machine Learning*, 1(2847-2854), 2016.

Ramachandran, P., Zoph, B., and Le, Q. V. Searching for activation functions. *arXiv:1710.05941*, 2017.

Sagun, L., Bottou, L., and LeCun., Y. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv:1611.07476v2*, 2017.

Schwartz-Ziv, R. and Tishby, N. Opening the black box of deep neural networks via information. *arXiv:1703.00810*, 2017.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929 – 1958, 2014.

Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. *Invited paper to ITW 2015; 2015 IEEE Information Theory Workshop*, 2015.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv:1611.03530*, 2017.