Meanfield theory of activation functions in Deep Neural Networks

Mirco Milletarí [1] **Thiparat Chotibut** [2] **Paolo E. Trevisanutto** [3]

Abstract

We present a statistical mechanics model of deep neural networks, connecting the energy-based and the feed forward (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, 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 behind these developments is still lacking, leading to an unsystematic growth of its complexity.

To assist future systematic studies and construction of FFNs, we propose a theoretical framework based on the tool-set of statistical mechanics. 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 function takes the form of the expected (mean) signal transmission; remarkably, this activation corresponds to (*Swish*), obtained in Ref. (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 the specific datasets they have been designed for, they generally fail to generalize. Finally, the standard *ReLU* activation arises as a limiting case of the *Swish*, corresponding to the noiseless limit of a communication channel. To the best of our knowledge, this provides the first derivation of *ReLU*, typically introduced heuristically to facilitate optimization protocols. Despite restricting our analysis to pure FFNs, most of our conclusions carry on to Convolutional and Recurrent networks.

The paper is organized as follows: In section 2, we provide a plausible argument based on dimensional analysis to explain why a hidden unit should be regarded as a communication channel and discuss the correspondece to an energy-based model. In analogy with Refs. (Tishby & Zaslavsky, 2015; Schwartz-Ziv & Tishby, 2017), we show that each hidden unit acts first as an encoder and then as a filter determining the quality of its input. In section 3, we explore the geometry of loss surfaces associated with training FFNs with *Swish* hidden units on simple classification tasks. The fraction of negative eigenvalues (α) of the Hessian (a measure of descent directions), as well as the fraction of zero eigenvalues, γ (a measure of flat directions), are investigated. We find that FFNs trained with *ReLU* always exhibit

$\gamma \neq 0$ while those trained with *Swish* often show $\gamma \simeq 0$, resulting in faster and more flexible training. These two indices seem to determine not only the speed of learning but also whether learning will be successful Codes and other numerical results, are provided in the appendix.

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 a feature component, 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 introduced the output of the outmost hidden layer ($\hat{\mathbf{y}}$) and the output information of the hidden units ($\mathbf{j}$); additional hidden layers can be introduced by further use of the chain rule. 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 Appendix (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 the proposed framework.

2.2. Motivations

Motivated by neurobiology and in analogy with the communication channel scheme in information theory (MacKay, 2003; Jaynes, 2003), we regard the input vector x_i^μ as the information source entering the processing units (neurons) of the network, 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 (fewer units) while the latter to an expansion layer (more units). 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 signal exceeds a local bias potential b_i. Both in the brain and in electronic devices, the input information 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, physically proportional to the current passing through it; on a computer, this is encoded in bits. Clearly, a linear combination of the input signals, together with the bias, has to preserve dimensions:

$$h_i^\mu = \sum_{j=1}^{n} w_{ij} x_j^\mu + b_i, \tag{1}$$

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 combination 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 the 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). In so far, we just described the general functioning principle underlying FFNs. Let us first 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, tuning to the noise strength; typically, one sets $\beta = 1$ or reabsorb it inside the weights and bias (Nguyen et al., 2017), but 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 noise). 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 when using the $\tanh$ activation, but it is absent when using *ReLU*, that correctly transmits the information itself.

2.3. Statistical mechanics of Feed Forward networks

A prototypical statistical mechanics formulation of Neural Networks is the *inverse* Ising problem (Nguyen et al., 2017), or Boltzmann machine, where one is interested in inferring the values of the couplings and external fields of the Ising

model. While the Boltzmann machine is an energy-based model, standard FFN is not commonly regarded as such. Here, we propose an energy-based formulation of FFNs to bridge on these two formulations, solely based on the maximum entropy principle (Nguyen et al., 2017; Castro & Raimondi, 2003; MacKay, 2003; Jaynes, 2003) to obtain the least biased probabilistic interpretation of hidden neurons. Eq. (**??**) describes the general structure of the network; in the following, we link this structure to a model of artificial neurons qualitatively described in section (2.2). Starting from the input layer, each unit takes the same input vector $\mathbf{x}$ and processed to output a new variable $\mathbf{h}$. One of the central results of this paper is based on the identification of h_i as an effective, coarse grained field coupling to the synaptic gate variable s_i. The feedforward nature of coarse graining (a directed graph), stems from the fact that it is not possible to entirely retrieve the original information after eliminating part of it; this endorses the forward pass with a semi-group structure. Considering the first layer, we need to evaluate the probability associated to the new, coarse grained variable $\mathbf{h}$,

$$P(\mathbf{h}) = \int d\mathbf{x}\, Q(\mathbf{h}|\mathbf{x})\, P(\mathbf{x}), \tag{2}$$

where $Q(\mathbf{h}|\mathbf{x})$ is the transition function modelling the encoder. In deep learning, the form of the transition function is fixed by the forward pass operation, 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 (empirical) sequence, or assume the form of the distribution from which $\mathbf{x}$ has been sampled from. 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 x_i^{μ}. As for the transition function, it enforces the information processing performed by the "soma" of the artificial neuron; in Deep Learning ,this consists of creating a linear combination of the input information. As information needs to be conserved, an additional constrain should be imposed on the coefficients of the transformation, namely $\sum_i w_{ij}^{[l]} = 1\ \forall l = 1, ..., L$, formally equivalent to the conservation of charge in an electronic system and to an L_1-type regularisation in machine learning.

$$\begin{aligned} P(\mathbf{h}) &= \int d\mathbf{x}\, \Gamma^{[1]}\, \delta\left(\mathbf{h} - \mathbf{w}^T\, \mathbf{x} - \mathbf{b}\right) \prod_{\mu=1}^{m} \delta(\mathbf{x} - \mathbf{x}^{\mu}) \quad (3) \\ &= \frac{\Gamma^{[1]}}{m} \sum_{\mu} \delta\left(\mathbf{h} - \mathbf{w}^T\, \mathbf{x}^{\mu} - \mathbf{b}\right) \\ \Gamma^{[1]} &= \prod_{j=1}^{n} \delta\left(\sum_{i=1}^{n_1} w_{ij}^{[1]} - 1\right). \end{aligned} \tag{4}$$

Eq. (3) can be interpreted in different ways: it is akin to the real space block-spin renormalization developed by Kadanoff, reformulated in the more general language of probability theory (Castro & Raimondi, 2003; Ma, 1976; Cassandro & Jona-Lasinio, 1978) or it can be seen as a way to implement a "change of variables" from x to h. The relation between Deep Learning and the renormalization group (RG) was previously observed in the context of restricted Boltzmann machine (Mehta & Schwab, 2014) and we now show that this connection holds here as well.

Once the coarse grained variables have been computed, the information passes through the "synaptic" gate that will transmit it 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, and the intralayer correlations are taken – a priori– to be zero. In other words, each unit within the same layer behaves independently of the others (i.e. no contextual information). Given a statistical ensemble of hidden binary gates, the most likely distribution can be obtained by maximising 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 and it reads

$$\begin{aligned} \mathscr{F} = &- \sum_{\mathbf{s}} P(\mathbf{s})\, \log P(\mathbf{s}) + \eta \left(\sum_{\mathbf{s}} P(\mathbf{s}) - 1\right) \\ &+ \sum_i \lambda_i \left(m_i - \sum_{\mathbf{s}} s_i\, P(\mathbf{s})\right), \end{aligned} \tag{5}$$

where λ_i are Lagrange multipliers chosen to reproduce the first moment of the data, while η enforces normalization. Functionally varying with respect to the probability $P(\mathbf{s})$ and solving for the Lagrange multipliers, one obtains (Castro & Raimondi, 2003):

$$\begin{aligned} P(\mathbf{s}|\mathbf{h}) &= \frac{1}{Z}\, e^{\sum_i \beta_i\, s_i\, h_i} \\ Z[\mathbf{h}] &= \prod_i \sum_{s_i = \{0,1\}} e^{\beta_i\, s_i\, h_i} = \prod_i \left(1 + e^{\beta_i\, h_i}\right). \end{aligned} \tag{6}$$

The parameters β_i encode the noise, or statistical uncertainty of s_i ($\lambda_i = \beta_i\, h_i$). 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 (the lateral connections). We have also introduced the notation $P(\mathbf{s}|\mathbf{h})$ 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

needs to evaluate the channel transmission probability

$$P(\mathbf{s}) = \int d\mathbf{h}\, P(\mathbf{s}|\mathbf{h})\, P(\mathbf{h}) = \frac{1}{Z} e^{-\sum_i \beta_i \mathscr{H}_i[s,x^\mu]} \quad (7)$$
$$Z = \frac{1}{m} \prod_{\mu=1}^{m} \prod_{i=1}^{n_1} \left[1 + e^{\beta_i\, h_i^\mu}\right] = \frac{1}{m} \prod_{\mu=1}^{m} \prod_{i=1}^{n_1} Z_{i,\mu},$$

where $\hat{h}_i^\mu = \sum_j w_{ij}^{[1]}\, x_j^\mu + b_i$, $Z_{i,\mu}$ is the partition function per example/index and we have identified the simple coarse grained Hamiltonian

$$\mathscr{H}_i = -\sum_{j=1}^{n} s_i\, w_{ij}^{[1]} x_j^\mu - s_i\, b_i. \quad (8)$$

Eqs. (7) and (8) can be seen as 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 above Hamiltonian has again the form of an RBM, this time with binary hidden units. The expected value of the channel transmission, $\langle s_i \rangle$ is (Hertz et al., 1991)

$$\langle s_i \rangle = \frac{1}{\beta_i} \frac{\partial}{\partial b_i} \log Z_{i,\mu} = \frac{1}{1 + e^{-\beta_i\, \hat{h}_i^\mu}} \equiv \sigma(\beta_i\, \hat{h}_i^\mu), \quad (9)$$

the logistic function evaluated for each empirical realization μ and unit i. We stress that this quantity is not the coarse grained input signal *transmitted* by the channel; it solely determines the expectation of channel transmissions *given* the coarsed grained input signal h_i^μ. Note that $\langle s_i \rangle$ would be the correct quantity for a single perceptron, or equivalently the output layer of an MLP performing a classification task. However, as discussed in the previous section, the hidden layers should not transmit $\langle s_i \rangle$, as this leads to dimensional mismatch.

To ensure dimensional consistency across hidden layers, the output signal of each hidden unit must have the same dimension as its input signal. Therefore, the correct quantity to consider is the *expectation value* of the output signal $j_i = h_i\, s_i$, see Fig. (**??**). This can be obtained by summing over all gate states, for each unit and example, or by using the partition function of Eq. (7),

$$\langle j_i \rangle = \langle \hat{h}_i^\mu\, s_i \rangle_s = \frac{\partial}{\partial \beta_i} \log Z_{i,\mu} = \hat{h}_i^\mu\, \sigma(\beta_i\, \hat{h}_i^\mu), \quad (10)$$

that agrees with the definition of the energy flux in statistical mechanics (Bellac et al., 2004). Note that contrary to Eq. (9), the noise parameters β_i cannot be rescaled away now. This function was recently obtained in Ref. (Ramachandran et al., 2017) (there named *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.

Now we interpret this activation function as an "Expected Signal Propagation" to stress its meaning and physical analogy, even though here β does not have to be strictly positive. A theoretical study of the performance of *Swish* from the point of view of information propagation has been proposed in Ref. (Hayou et al., 2018). Our model naturally explains these studies by identifying *Swish* with the expectation value of the coarse grained input transmitted by each unit. In the next section we will better analyze *Swish* its meaning within backward propagation, and how it modifies the geometry of the loss surface.

Consider now the limit of a noiseless system, i.e. $\forall\, i$, $\beta_i \to \infty$ in the above equation:

$$\lim_{\beta_i \to \infty} \langle j_i \rangle = \hat{h}_i^\mu\, \theta(\hat{h}_i^\mu) \equiv \max\left\{\hat{\mathrm{h}}_\mathrm{i}^\mu, 0\right\} \equiv \mathrm{ReLU}, \quad (11)$$

where $\theta(.)$ is the Heaviside step function and in the last equality we have identified the *ReLU* activation. To the best of our knowledge, this is the first consistent derivation of *ReLU*, usually obtained following heuristic arguments. *ReLU* emerges as the noiseless limit of the mean transmitted information across the units; as such, it is not affected by the dimensional mismatch as it can be easily checked following the argument presented in Sec. (2.2). In Sec. (2.4) we discuss how the empirical observation that *ReLU* is not affected by the vanishing gradient problem is related to the absence of dimensional mismatch and the concept of expected signal propagation. 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 coarse grained input signal exceeds a threshold, while the former can pass a signal lower than the threshold with a finite probability. In Fig. (**??**) we plot the three activation functions for $\beta = 1$. Clearly, *Swish* encompasses the statistical uncertainty conveyed by each processing unit in estimating the pre-activation value h. A positive value of the activation tells the units in the next layer that a certain combination of the previous inputs should be strengthen while a negative value means that it should be weakened, i.e. unlearned; the latter option is absent when using the *ReLU* function, a feature known to generally slow down learning (Engel & den Broeck, 2004). In the opposite extreme, the noisy limit $\beta \ll 1$, the *Swish* becomes linear. Therefore, we can consider linear networks as a noisy limit of non linear ones.

So far we have discussed the information processing mechanism of a single hidden layer; going back to Eq. (**??**), we see that additional hidden layers can be introduced by exploiting the chain rule further. However, there is a final step we need in order to bridge Eq. (**??**) and Eq. (10) or (11), that is fixing $P(j) \simeq \delta(j - \langle j \rangle)$, i.e. fixing the neuron's output distribution on its first moment. There is no reason *a priori* for this approximation other than computational, and

it would be certainly interesting to study how the network's performance and complexity changes when this approximation is relaxed, for example by introducing fluctuations (second moment) in the output signal.

We would like to conclude this section noticing that in spin glasses (Mézard et al., 1987; Dominicis & Giardina, 2016), the energy landscape presents an increasing number of degenerate local minima as the temperature is lowered, a feature shared with the Hopfield model of associative memory (Amit & Gutfreund, 1985) and RBM. The effect of noise in FFNs was recently considered in Ref. (Chaudhari et al., 2017), where an improved optimization algorithm based on the maximum entropy principle was proposed.

2.4. Back-propagation: a probabilistic point of view

At the heart of any FFN model is the back propagation algorithm (Hertz et al., 1991; Bishop, 2006). Forward propagation returns a first "guess" on the value of the learning parameters, that are subsequently adjusted layer by layer by minimizing a properly defined loss function, i.e. the energy of the system. Consider the output layer "L", then the gradient of the weights is

$$\frac{\partial \mathscr{L}}{\partial \mathbf{w}_L} = \frac{1}{m}\sum_{\mu=1}^{m}\left[\mathbf{e}^\mu\, \mathbf{g}(\hat{\mathbf{h}}_L^\mu)\right]\mathbf{a}_{L-1} \tag{12}$$

$$\hat{\mathbf{g}}(\mathbf{h}_l^\mu) = \boldsymbol{\sigma}(\boldsymbol{\beta}_l\,\hat{\mathbf{h}}_l^\mu)[1+\hat{\mathbf{h}}_l^\mu\boldsymbol{\beta}_l\,\boldsymbol{\sigma}(-\boldsymbol{\beta}_l\,\hat{\mathbf{h}}_l^\mu)],$$

where $\mathbf{e}^\mu$ is the residual error, depending on the difference between the network output $\hat{y}^\mu$ and the ground truth vector y^μ. In the optimization phase we look for the stationary point $\partial\mathscr{L}/\partial\boldsymbol{\theta}_l^\alpha = 0$, where $\boldsymbol{\theta}_l^\alpha = \{\mathbf{w}_l, \mathbf{b}_l, \boldsymbol{\beta}_l\}$ is the set of learning parameters. 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, $\mathbf{g}(\hat{\mathbf{h}}_l^\mu) = \boldsymbol{\sigma}(\boldsymbol{\beta}_l\,\hat{\mathbf{h}}_l^\mu)\boldsymbol{\sigma}(-\boldsymbol{\beta}_l\,\hat{\mathbf{h}}_l^\mu)$, commonly known in physics as a phase space factor; it appears e.g. in the collision integral of the Boltzmann equation (Castro & Raimondi, 2003), which describes the transition probability from an occupied to an empty state. Ref. (Tishby & Zaslavsky, 2015; Schwartz-Ziv & Tishby, 2017) show that there are two distinct phases of learning: empirical error minimization (the residual) and representation learning. Following the discussion of the previous section, 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 $\sigma(\beta_i\,\hat{h}_i) \equiv P(s_i = 1|\bar{x}_i) \simeq 1$ [1] 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 behaviour for *Swish*. If $\hat{h}_i \gg 1/\beta_i$ we now have $g(\hat{h}_i^\mu) \simeq 1$, 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, as expected. 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 purely "excitatory" network, in which representation unlearning is not possible. In other words, the fact that *Swish* can be negative for $\sigma(\hat{h})/\beta < \hat{h} < 0$ allows for greater flexibility and the possibility to avoid plateaus surrounding local minima and saddle-points. In Ref. (Dauphin et al., 2014) it was proposed that plateaus of zero (or small) curvatures in the loss are mostly responsible for slowing down or preventing convergence of gradient descent. These plateaus are a consequence of continuous symmetries in the loss function (Noether's theorem) that are generated, in analogy to the L^2 regularization, by the large values for individual variables. The following section provides a numerical analysis to support this picture. Details of the the back-propagation algorithm with *Swish* for a general L layer network can be found in the SM.

3. Numerical Analysis

A thorough performance analysis of *Swish* versus other popular choices of activations was carried out in Ref. (Ramachandran et al., 2017), where it was shown that the former outperformed all other choices on image classification tasks using a convolutional structure. It was also noted that to take full advantage of *Swish* one should also reconsider the way convolution is performed; for the moment, we leave this interesting point open and rather focus on understanding the optimization dynamics in pure FFNs. We have considered both artificial and experimental datasets trained with ADAM gradient descent (Kingma & Ba, 2015). Given that both datasets lead to the same qualitative features, we focus our analysis on the former and discuss the latter, together with additional details in the SM.

In Fig. (**??**)(a) we show the loss functions for 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 (adding more units does not improve performance), 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–(Fig. **??**(b))and the fraction of zero eigenvalues – γ – (Fig. **??**(b)). The former measures the ratio of

[1] Here $\bar{x}_i$ is the input of the unit, not necessarily the input data of the network.

descent to ascent directions on the energy landscape; when α is large, gradient descent can quickly escape a critical point – a saddle point in this case – due to the existence of multiple unstable directions. However, when a critical point exhibits multiple near-zero eigenvalues, roughly captured by γ, the energy landscape in this neighborhood consists of several near-flat (to second-order) directions; in this situation, gradient descent will slowly decrease the training loss. In Ref. (Dauphin et al., 2014) it was noted that energy plateaus are responsible for slowing down or preventing learning. This can be understood by considering the case in which learning completely fails: the loss does not decrease and all the eigenvalues of the Hessian are zero. Noether's theorem infers that these eigenvalues are originated by symmetries of the loss function. 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 datasets, as discussed in Ref. (Ramachandran et al., 2017). In Fig. (**??**) we show the α index, usually smaller for the *ReLU* network, and a finite value of γ that slows down learning. We find $\gamma \simeq 0.45-0.6$ for *ReLU* and $\gamma = 0$ for *Swish* and σ, both for the linear and non-linear task. We have also evaluated the fraction of residuals $e^\mu \simeq (\hat{y}^\mu - y^\mu)/m$ closer to zero and found, surprisingly, that *Swish* greatly outperforms the other activations in minimizing the empirical error, see Fig. (**??**). In addition, we find that the eigenvalue distribution obtained with *ReLU shrinks* with increasing training epochs, giving rise to the singular distribution reported in Ref. (Pennington & Bahri, 2017; Sagun et al., 2017). In contrast,*Swish* show a significantly wider spread in the eigenvalue distribution at the end of training, see SM for details and discussions, together with results for the MNIST dataset.

Swish training networks can 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.(**??**) , 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. (**??**) bottom left loss function minimization). In the case of Relu, the loss function minimization is almost immediately reached after 300 epocs. This is reflected in the Hessian eigenvectors that still coincide with the variables (in the Figure, the bar indicate the coefficient value of the eigenvectors). In the Sigmoid case, the Hessian eigenvectors are spread more uniformly on the variables showing a sort of graining regularization with respect to the Relu case. Nevertheless, there is still a concentration of high value Hessian coefficients in two sub-matrices. Lastly, in the Swish case, the Hessian coefficients are spread almost uniformly over the transformation matrix. We stress that this effect has some analogies with the regularization that prevents overfitting disfavouring large values for individual variables. In Fig. (**??**), the same analysis is repeated for the non-linear two hidden 8-2 layer network. Whereas in the Relu results still show a localization of variables in the eigenvectors, in the Sigmoid and Swish case the Hessian eigenvectors involve a broader number of variables. Nevertheless, in the Swish case, the number of large value coefficients is still less than in the Sigmoid case.

4. Conclusions and perspectives

In this work we have introduced an energy-based model that allows systematic construction and analysis of FFNs. It provides a coherent interpretation of the computational process of each and every unit (neuron). Furthermore, it presents a method that overcomes the dimensional mismatch that arises from using heuristic activations. Enforcing dimensional consistency naturally leads to a class of activations with the prime focus of propagating the expectation value of processed signals. Our results provide a theoretical justification of the *Swish* activation found in Ref. (Ramachandran et al., 2017). In addition, we demonstrate the superiority of this *Swish* activation through numerical experiments that reveal the geometry of its loss manifold. *ReLU* networks, unlike σ and $tanh$, do not suffer from dimensional mismatch, yet their restricted phase space results in a finite fraction of null directions of gradient descent that can slow down or, in some cases, completely prevent learning. To overcome this problem, one needs a detailed fine tuning of the network's topology that likely increases the complexity of the learning task. In stark contrast, *Swish* trained networks are less prone to fine tuning and outperform other activations on minimizing the empirical error. We hope this statistical mechanics view can facilitate future studies of FFNs, e.g. to explore the scaling relation between the *Swish* Hessian's eigenvalue distributions and the parameters that describe its loss landscape, analogous to those studied in Ref. (Pennington & Bahri, 2017) for standard activations. As previously discussed, the Renormalization Group (RG) approach could hint at designing principles of FFNs, as it delineates how complex behavior emerges from the elementary constituents of a system. Although this framework has been considered in Ref. (Mehta & Schwab, 2014; Koch-Janusz & Ringel, 2018), the complete mathematical formalism to correctly use it in FFNs is still missing. We believe this could answer

some of the most pressing questions in the construction of FFNs, namely: given a dataset showing specific strength and extent of input/output correlations, what is the optimal network topology? Finally, extensions of the methods discussed here to Convolutional and Recurrent architectures could lead to substantial performance improvement and, potentially, faster training algorithms.

5. Acknowledgements

M.M. would like to thank Ned Phillips, Bill Phillips, Sarah Chan and Roberto Raimondi for support and discussions. A special thanks to Fábio Hipólito and Aki Ranin for carefully reading and commenting the manuscript. T.C. would like thank Shaowei Lin for useful discussions and for financial support from the startup research grant SRES15111, and the SUTD-ZJU collaboration research
grant ZJURP1600103. P.E.T would like to thank M. D. Costa for help with HPC. Some of the calculations were carried out at the HPC facilities of the NUS Centre for Advanced 2D materials. Finally, M.M. And T.C. would like to thank the AI Saturday meetup initiative organised by Nurture.ai, that sparked some of the ideas presented in this work.

A. Cross Entropy Loss

In this appendix we present a derivation of the cross entropy loss function using large deviation methods (Mézard & Montanari, 2009). A useful starting point is the definition of the accuracy of a classification task, that is the number of times the network prediction $\hat{y}$ equals the provided solution y. This corresponds to the conditional probability $P(y|\hat{y}) = \mathbb{I}(y = \hat{y})$ in Eq. (**??**). As $\hat{y}$ is a random variable, we want to know which is 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. (**??**), and using the standard definition relating the Loss function to the log-probability $\mathscr{L} = -\log P(\mathbf{y})$, we need to evaluate

$$-\log P(\mathbf{y}) = -\log \int d\hat{\mathbf{y}}\, P(\mathbf{y}|\hat{\mathbf{y}})\, P(\hat{\mathbf{y}}), \qquad (13)$$

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. (14), i.e. we fix the random variables on the observed data. Using the Dirac delta as a representation of the Indicator function we have

$$\begin{aligned} P(y) &= \int \left(\prod_\mu d\hat{y}^\mu\right) \prod_\mu \delta(y^\mu - \hat{y}^\mu)\, P(\hat{y}) \qquad (14)\\ &= \int \left(\prod_\mu \frac{d\lambda^\mu}{2\pi}\right) \prod_\mu e^{i\,\sum_\mu \lambda^\mu\, y^\mu} \chi(\lambda^\mu). \end{aligned}$$

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

$$\chi(\lambda^\mu) = \int \left(\prod_\mu d\hat{y}^\mu\right) P(\hat{y}^\mu)\, e^{-i\,\sum_\mu \lambda^\mu\, \hat{y}^\mu} \qquad (15)$$

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

$$P(\hat{y}^\mu) = \int d\boldsymbol{\theta}\, \{q^\mu(\boldsymbol{\theta})\, \delta(\hat{y}^\mu - 1) + (1 - q^\mu(\boldsymbol{\theta}))\, \delta(\hat{y}^\mu)\}\,, \qquad (16)$$

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. In this case, Eq. (17) reads

$$\chi(\lambda^\mu) = \int d\boldsymbol{\theta} \left[q^\mu(\boldsymbol{\theta}) e^{-i\,\lambda^\mu} + (1 - q^\mu(\boldsymbol{\theta}))\right]. \qquad (17)$$

Using this expression back in Eq. (16) we arrive at the intermediate result

$$\begin{aligned} P(y) &= \int d\boldsymbol{\theta} \int \left(\prod_\mu \frac{d\lambda^\mu}{2\pi}\right) e^{i\,\sum_\mu \lambda^\mu\, y^\mu + \sum_\mu \log\left[q^\mu e^{-i\,\lambda^\mu} + (1-q^\mu)\right]} \qquad (18)\\ &= \int \left(\prod_\mu \frac{d\lambda^\mu}{2\pi}\right) e^{m\, S[\lambda]}. \end{aligned}$$

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

$$\begin{aligned} \frac{\partial S[\lambda]}{\partial \lambda^\mu} &= i\, y^\mu - \frac{i\, q^\mu\, e^{-i\,\lambda^\mu}}{q^\mu\, e^{-i\,\lambda^\mu} + (1 - q^\mu)} = 0 \qquad (19)\\ &\to -i\lambda_c^\mu = \log \frac{y^\mu(1 - q^\mu)}{q^\mu(1 - y^\mu)} \end{aligned}$$

Using the optima back in Eq. (18) we arrive after some simple algebra to the expression for the cross-entropy:

$$\begin{aligned} P(y) &\simeq \int d\boldsymbol{\theta}\, e^{m S[\lambda_c, \boldsymbol{\theta}]} \qquad (20)\\ S[\lambda_c] &= \frac{1}{m} \sum_\mu \{y^\mu\, \log[q^\mu(\boldsymbol{\theta})] + (1 - y^\mu)\, \log[1 - q^\mu(\boldsymbol{\theta})]\}\,, \end{aligned}$$

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^*.

References

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

Amit, D. J. and Gutfreund, H. Spin glass theory model of neural networks. *Physical Review A*, 32:1007, 1985.

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.

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

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.

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.

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.

Engel, A. and den Broeck, C. V. *Statistical Mechanics of Learning*. Cambridge University Press, 2004.

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.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *Proceedings of the 3rd International Conference for Learning Representations*, 2015.

Koch-Janusz, M. and Ringel, Z. Mututal information, neural networks and the renormalization group. *Nature Physics*, 2018.

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.