# On the Impact of the Activation Function on Deep Neural Networks Training

**Soufiane Hayou** [1] **Arnaud Doucet** [1] **Judith Rousseau** [1]

## Abstract

The weight initialization and the activation function of deep neural networks have a crucial impact on the performance of the training procedure. An inappropriate selection can lead to the loss of information of the input during forward propagation and the exponential vanishing/exploding of gradients during back-propagation. Understanding the theoretical properties of untrained random networks is key to identifying which deep networks may be trained successfully as recently demonstrated by (Schoenholz et al., 2017) who showed that for deep feedforward neural networks only a specific choice of hyperparameters known as the 'Edge of Chaos' can lead to good performance. While the work by (Schoenholz et al., 2017) discuss trainability issues, we focus here on training acceleration and overall performance. We give a comprehensive theoretical analysis of the Edge of Chaos and show that we can indeed tune the initialization parameters and the activation function in order to accelerate the training and improve performance.

## 1. Introduction

Deep neural networks have become extremely popular as they achieve state-of-the-art performance on a variety of important applications including language processing and computer vision; see, e.g., (Goodfellow et al., 2016). The success of these models has motivated the use of increasingly deep networks and stimulated a large body of work to understand their theoretical properties. It is impossible to provide here a comprehensive summary of the large number of contributions within this field. To cite a few results relevant to our contributions, (Montufar et al., 2014) have shown that neural networks have exponential expressive power with respect to the depth while (Poole et al., 2016) obtained similar results using a topological measure of expressiveness.

Since the training of deep neural networks is a non-convex optimization problem, the weight initialization and the activation function will essentially determine the functional subspace that the optimization algorithm will explore. We follow here the approach of (Poole et al., 2016) and (Schoenholz et al., 2017) by investigating the behaviour of random networks in the infinite-width and finite-variance i.i.d. weights context where they can be approximated by a Gaussian process as established by (Neal, 1995), (Matthews et al., 2018) and (Lee et al., 2018).

In this paper, our contribution is three-fold. Firstly, we provide a comprehensive analysis of the so-called Edge of Chaos (EOC) curve and show that initializing a network on this curve leads to a deeper propagation of the information through the network and accelerates the training. In particular, we show that a feedforward ReLU network initialized on the EOC acts as a simple residual ReLU network in terms of information propagation. Secondly, we introduce a class of smooth activation functions which allow for deeper signal propagation (Proposition 3) than ReLU. In particular, this analysis sheds light on why smooth versions of ReLU (such as SiLU or ELU) perform better experimentally for deep neural networks; see, e.g., (Clevert et al., 2016), (Pedamonti, 2018), (Ramachandran et al., 2017) and (Milletarí et al., 2018). Lastly, we show the existence of optimal points on the EOC curve and we provide guidelines for the choice of such point and we demonstrate numerically the consistence of this approach. We also complement previous empirical results by illustrating the benefits of an initialization on the EOC in this context. All proofs are given in the Supplementary Material.

## 2. On Gaussian process approximations of neural networks and their stability

### 2.1. Setup and notations

We use similar notations to those of (Poole et al., 2016) and (Lee et al., 2018). Consider a fully connected feedforward random neural network of depth $L$, widths $(N_l)_{1\leq l\leq L}$, weights $W^l_{ij} \overset{iid}{\sim} \mathcal{N}(0, \frac{\sigma^2_w}{N_{l-1}})$ and bias $B^l_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_b)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution of mean $\mu$

[1] Department of Statistics, University of Oxford, Oxford, United Kingdom. Correspondence to: Soufiane Hayou <soufiane.hayou@stats.ox.ac.uk>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

and variance $\sigma^2$. For some input $a \in \mathbb{R}^d$, the propagation of this input through the network is given for an activation function $\phi : \mathbb{R} \to \mathbb{R}$ by

$$y_i^1(a) = \sum_{j=1}^{d} W_{ij}^1 a_j + B_i^1, \tag{1}$$

$$y_i^l(a) = \sum_{j=1}^{N_{l-1}} W_{ij}^l \phi(y_j^{l-1}(a)) + B_i^l, \quad \text{for } l \geq 2. \tag{2}$$

Throughout this paper we assume that for all $l$ the processes $y_i^l(.)$ are independent (across $i$) centred Gaussian processes with covariance kernels $\kappa^l$ and write accordingly $y_i^l \overset{ind}{\sim} \mathcal{GP}(0, \kappa^l)$. This is an idealized version of the true processes corresponding to choosing $N_{l-1} = +\infty$ (which implies, using Central Limit Theorem, that $y_i^l(a)$ is a Gaussian variable for any input $a$). The approximation of $y_i^l(.)$ by a Gaussian process was first proposed by (Neal, 1995) in the single layer case and has been recently extended to the multiple layer case by (Lee et al., 2018) and (Matthews et al., 2018). We recall here the expressions of the limiting Gaussian process kernels. For any input $a \in \mathbb{R}^d$, $\mathbb{E}[y_i^l(a)] = 0$ so that for any inputs $a, b \in \mathbb{R}^d$

$$\begin{aligned}\kappa^l(a,b) &= \mathbb{E}[y_i^l(a) y_i^l(b)] \\ &= \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y_i^{l-1}(a))\phi(y_i^{l-1}(b))] \\ &= \sigma_b^2 + \sigma_w^2 F_\phi(\kappa^{l-1}(a,a), \kappa^{l-1}(a,b), \kappa^{l-1}(b,b))\end{aligned}$$

where $F_\phi$ is a function that only depends on $\phi$. This gives a recursion to calculate the kernel $\kappa^l$; see, e.g., (Lee et al., 2018) for more details. We can also express the kernel $\kappa^l(a,b)$ (which we denote hereafter by $q_{ab}^l$) in terms of the correlation $c_{ab}^l$ in the $l^{th}$ layer

$$q_{ab}^l = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(\sqrt{q_a^{l-1}} Z_1)\phi(\sqrt{q_b^{l-1}} U_2(c_{ab}^{l-1}))]$$

where $q_a^{l-1} := q_{aa}^{l-1}$, resp. $c_{ab}^{l-1} := q_{ab}^{l-1} / \sqrt{q_a^{l-1} q_b^{l-1}}$, is the variance, resp. correlation, in the $(l-1)^{th}$ layer and $U_2(x) = xZ_1 + \sqrt{1-x^2} Z_2$ where $Z_1$, $Z_2$ are independent standard Gaussian random variables. When it propagates through the network. $q_a^l$ is updated through the layers by the recursive formula $q_a^l = F(q_a^{l-1})$, where $F$ is the 'variance function' given by

$$F(x) = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(\sqrt{x} Z)^2], \qquad Z \sim \mathcal{N}(0,1) \tag{3}$$

Throughout this paper, $Z, Z_1, Z_2$ will always denote independent standard Gaussian variables, and $a, b$ two inputs for the network.

Before starting our analysis, we define the transform $V$ for a function $\phi$ defined on $\mathbb{R}$ by $V[\phi](x) = \sigma_w^2 \mathbb{E}[\phi(\sqrt{x}Z)^2]$ for $x \geq 0$. We have $F = \sigma_b^2 + V[\phi]$.

Let $E$ and $G$ be two subsets of $\mathbb{R}$. We define the following sets of functions for $k \in \mathbb{N}$ by

$$\begin{aligned}&\mathcal{D}^k(E,G) = \{f : E \to G \text{ such that } f^{(k)} \text{ exists}\} \\ &\mathcal{C}^k(E,G) = \{f \in \mathcal{D}^k(E,G) \text{ such that } f^{(k)} \text{ is continuous}\} \\ &\mathcal{D}_g^k(E,G) = \{f \in \mathcal{D}^k(E,G) : \forall j \leq k, \mathbb{E}[f^{(j)}(Z)^2] < \infty\} \\ &\mathcal{C}_g^k(E,G) = \{f \in \mathcal{C}^k(E,G) : \forall j \leq k, \mathbb{E}[f^{(j)}(Z)^2] < \infty\}\end{aligned}$$

where $f^{(k)}$ is the $k^{\text{th}}$ derivative of $f$. When $E$ and $G$ are not explicitly mentioned, we assume $E = G = \mathbb{R}$.

### 2.2. Limiting behaviour of the variance and covariance operators

We analyze here the limiting behaviour of $q_a^l$ and $c_{a,b}^l$ as $l$ goes to infinity. From now onwards, we will also assume without loss of generality that $c_{ab}^1 \geq 0$ (similar results can be obtained straightforwardly when $c_{ab}^1 \leq 0$). We first need to define the *Domains of Convergence* associated with an activation function $\phi$.

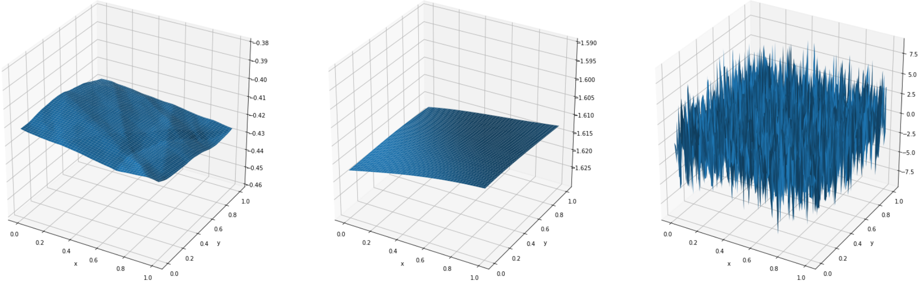
**Definition 1.** *Let $\phi \in \mathcal{D}_g^0$, $(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2$.*
*(i) Domain of convergence for the variance $D_{\phi,var}$ : $(\sigma_b, \sigma_w) \in D_{\phi,var}$ if there exists $K > 0$, $q \geq 0$ such that for any input $a$ with $q_a^1 \leq K$, $\lim_{l\to\infty} q_a^l = q$. We denote by $K_{\phi,var}(\sigma_b, \sigma_w)$ the maximal $K$ satisfying this condition.*
*(ii) Domain of convergence for the correlation $D_{\phi,corr}$: $(\sigma_b, \sigma_w) \in D_{\phi,corr}$ if there exists $K > 0$ such that for any two inputs $a, b$ with $q_a^1, q_b^1 \leq K$, $\lim_{l\to\infty} c_{ab}^l = 1$. We denote by $K_{\phi,corr}(\sigma_b, \sigma_w)$ the maximal $K$ satisfying this condition.*

*Remark*: Typically, $q$ in Definition 1 is a fixed point of the variance function defined in (3). Therefore, it is easy to see that for any $(\sigma_b, \sigma_w)$ such that $F$ is non-decreasing and admits at least one fixed point, we have $K_{\phi,var}(\sigma_b, \sigma_w) \geq q$ where $q$ is the minimal fixed point; i.e. $q := \min\{x : F(x) = x\}$. Thus, if we re-scale the input data to have $q_a^1 \leq q$, the variance $q_a^l$ converges to $q$. We can also re-scale the variance $\sigma_w$ of the first layer (only) to assume that $q_a^1 \leq q$ for all inputs $a$.

The next Lemma gives sufficient conditions under which $K_{\phi,var}$ and $K_{\phi,corr}$ are infinite.

**Lemma 1.** *Assume $\phi''$ exists at least in the distribution sense.*[1]
*Let $M_\phi := \sup_{x \geq 0} \mathbb{E}[|\phi'^2(xZ) + \phi''(xZ)\phi(xZ)|]$. Assume $M_\phi < \infty$, then for $\sigma_w^2 < \frac{1}{M_\phi}$ and $\sigma_b \geq 0$, we have $(\sigma_b, \sigma_w) \in D_{\phi,var}$ and $K_{\phi,var}(\sigma_b, \sigma_w) = \infty$.*
*Let $C_{\phi,\delta} := \sup_{x,y \geq 0, |x-y| \leq \delta, c \in [0,1]} \mathbb{E}[|\phi'(xZ_1)\phi'(y(cZ_1 + \sqrt{1-c^2}Z_2))|]$. Assume $C_{\phi,\delta} < \infty$ for some $\delta > 0$,*

[1] ReLU admits a Dirac mass in 0 as second derivative and so is covered by our developments.



(a) ReLU with $(\sigma_b, \sigma_w) = (1, 1)$ (b) Tanh with $(\sigma_b, \sigma_w) = (1, 1)$ (c) Tanh with $(\sigma_b, \sigma_w) = (0.3, 2)$

*Figure 1.* Draws of outputs for ReLU and Tanh networks for different parameters $(\sigma_b, \sigma_w)$. Figures (a) and (b) show the effect of an initialization in the ordered phase, the outputs are nearly constant. Figure (c) shows the effect of an initialization in the chaotic phase.

*then for $\sigma_w^2 < \min(\frac{1}{M_\phi}, \frac{1}{C_\phi})$ and $\sigma_b \geq 0$, we have $(\sigma_b, \sigma_w) \in D_{\phi,var} \cap D_{\phi,corr}$ and $K_{\phi,var}(\sigma_b, \sigma_w) = K_{\phi,corr}(\sigma_b, \sigma_w) = \infty$.*

The proof of Lemma 1 is straightforward. We prove that $\sup F'(x) = \sigma_w^2 M_\phi$ and then apply the Banach fixed point theorem. Similar ideas are used for $C_{\phi,\delta}$.

*Example*: For ReLU activation function, we have $M_{ReLU} = 1/2$ and $C_{ReLU,\delta} \leq 1$ for any $\delta > 0$.

In the domain of convergence $D_{\phi,var} \cap D_{\phi,corr}$, for all $a, b \in \mathbb{R}^d$, we have $y_i^\infty(a) = y_i^\infty(b)$ almost surely and the outputs of the network are constant functions. Figures 1(a) and 1(b) illustrate this behaviour for ReLU and Tanh with inputs in $[0, 1]^2$ using a network of depth $L = 20$ with $N_l = 300$ neurons per layer. The draws of outputs of these networks are indeed almost constant.

Under the conditions of Lemma 1, both the variance and the correlations converge exponentially fast (contraction mapping). To refine this convergence analysis, (Schoenholz et al., 2017) established the existence of $\epsilon_q$ and $\epsilon_c$ such that $|q_a^l - q| \sim e^{-l/\epsilon_q}$ and $|c_{ab}^l - 1| \sim e^{-l/\epsilon_c}$ when fixed points exist. The quantities $\epsilon_q$ and $\epsilon_c$ are called 'depth scales' since they represent the range of depth to which the variance and correlation can propagate without being exponentially close to their limits. More precisely, if we write $\chi_1 = \sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2]$ and $\alpha = \chi_1 + \sigma_w^2 \mathbb{E}[\phi''(\sqrt{q}Z)\phi(\sqrt{q}Z)]$ then the depth scales are given by $\epsilon_q = -\log(\alpha)^{-1}$ and $\epsilon_c = -\log(\chi_1)^{-1}$. The equation $\chi_1 = 1$ corresponds to an infinite depth scale of the correlation. It is called the EOC as it separates two phases: an ordered phase where the correlation converges to 1 if $\chi_1 < 1$ and a chaotic phase where $\chi_1 > 1$ and the correlations do not converge to 1. In this chaotic regime, it has been observed in (Schoenholz et al., 2017) that the correlations converge to some value $c < 1$ when $\phi(x) = \text{Tanh}(x)$ and that $c$ is independent of the correlation between the inputs. This means that very close inputs (in terms of correlation) lead to very different outputs. Therefore, in the chaotic phase, at the limit of infinite width and depth, the output function of the neural network is non-continuous everywhere. Figure 1(c) shows an example of such behaviour for Tanh.

**Definition 2** (Edge of Chaos)**.** *For $(\sigma_b, \sigma_w) \in D_{\phi,var}$, let $q$ be the limiting variance*[2]*. The Edge of Chaos (EOC) is the set of values of $(\sigma_b, \sigma_w)$ satisfying $\chi_1 = \sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2] = 1$.*

To further study the EOC regime, the next lemma introduces a function $f$ called the 'correlation function' showing that that the correlations have the same asymptotic behaviour as the time-homogeneous dynamical system $c_{ab}^{l+1} = f(c_{ab}^l)$.

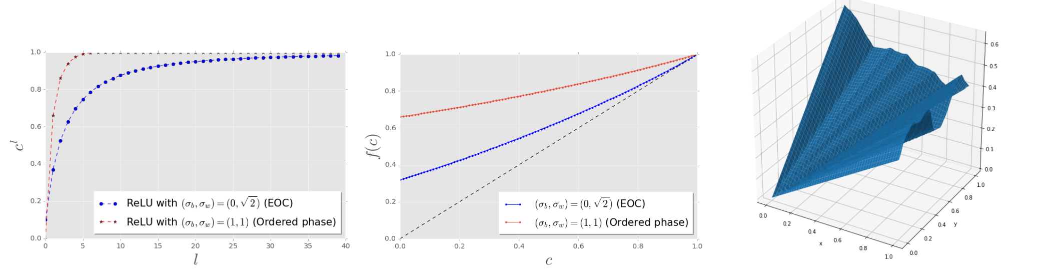
**Lemma 2.** *Let $(\sigma_b, \sigma_w) \in D_{\phi,var} \cap D_{\phi,corr}$ such that $q > 0$, $a, b \in \mathbb{R}^d$ and $\phi$ a measurable function such that $\sup_{x \in S} \mathbb{E}[\phi(xZ)^2] < \infty$ for all compact sets $S$. Define $f_l$ by $c_{ab}^{l+1} = f_l(c_{ab}^l)$ and $f$ by $f(x) = \frac{\sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(\sqrt{q}Z_1)\phi(\sqrt{q}(xZ_1 + \sqrt{1-x^2}Z_2))}{q}$. Then $\lim_{l \to \infty} \sup_{x \in [0,1]} |f_l(x) - f(x)| = 0$.*

The condition on $\phi$ in Lemma 2 is violated only by activation functions with square exponential growth (which are not used in practice), so from now onwards, we use this approximation in our analysis. Note that being on the EOC is equivalent to $(\sigma_b, \sigma_w)$ satisfying $f'(1) = 1$. In the next section, we analyze this phase transition carefully for a large class of activation functions.

## 3. Edge of Chaos

To illustrate the effect of the initialization on the EOC, we plot in Figure 2(c) the output of a ReLU neural network with 20 layers and 100 neurons per layer with parameters $(\sigma_b^2, \sigma_w^2) = (0, 2)$ (as we will see later EOC $= \{(0, \sqrt{2})\}$

[2] The limiting variance is a function of $(\sigma_b, \sigma_w)$ but we do not emphasize it notationally.



(a) Convergence of the correlation to 1 with $c^0 = 0.1$

(b) Correlation function $f$

(c) Output of 100x20 ReLU network on the EOC

*Figure 2.* Impact of the EOC initialization on the correlation and the correlation function. In (a), the correlation converges to 1 at a sub-exponential rate when the network is initialized on the EOC. In (b), the correlation function $f$ satisfies $f'(1) = 1$ on the EOC.

for ReLU). Unlike the output in Figure 1(a), this output displays much more variability. However, we prove below that the correlations still converge to 1 even in the EOC regime, albeit at a slower rate.

### 3.1. ReLU-like activation functions

ReLU has replaced classical activations (sigmoid, Tanh,...) which suffer from gradient vanishing (see e.g. (Glorot et al., 2011) and (Nair and Hinton, 2010)). Many variants such as Leaky-ReLU were also shown to enjoy better performance in test accuracy (Xu et al., 2015). This motivates the analysis of such functions from an initialization point of view. Let us first define this class.

**Definition 3** (ReLU-like functions)**.** *A function $\phi$ is ReLU-like if it is of the form*

$$\phi(x) = \begin{cases} \lambda x & \text{if } x > 0 \\ \beta x & \text{if } x \leq 0 \end{cases}$$

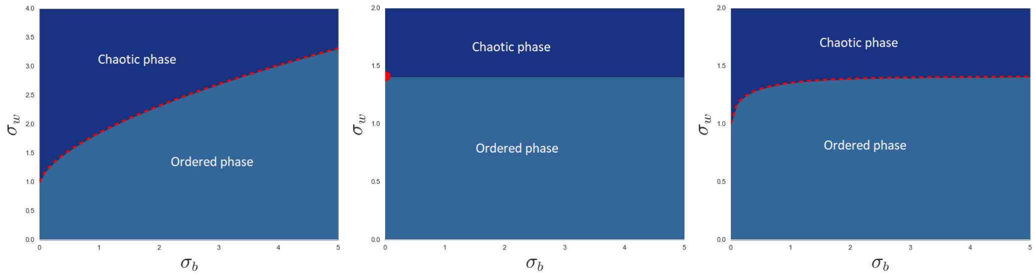
*where $\lambda, \beta \in \mathbb{R}$.*

ReLU corresponds to $\lambda = 1$ and $\beta = 0$. For this class of activation functions, the EOC in terms of definition 2 is reduced to the empty set. However, we can define a weak version of the EOC for this class. From Lemma 1, when $\sigma_w < \sqrt{\frac{2}{\lambda^2+\beta^2}}$, the variances converge to $q = \frac{\sigma_b^2}{1-\sigma_w^2/2}$ and the correlations converge to 1 exponentially fast. If $\sigma_w > \sqrt{\frac{2}{\lambda^2+\beta^2}}$ the variances converge to infinity. We then have the following result.

**Lemma 3** (Weak EOC)**.** *Let $\phi$ be a ReLU-like function with $\lambda, \beta$ defined as above. Then $f_l'$ does not depend on $l$, and $f_l'(1) = 1$ and $q^l$ bounded holds if and only if $(\sigma_b, \sigma_w) = (0, \sqrt{\frac{2}{\lambda^2+\beta^2}})$.*
*We call the singleton $\{(0, \sqrt{\frac{2}{\lambda^2+\beta^2}})\}$ the weak EOC.*

The non existence of EOC for ReLU-like activation in the sense of definition 2 is due to the fact that the variance is unchanged ($q_a^l = q_a^1$) on the weak EOC, so that the limiting variance $q$ depends on $a$. However, this does not impact the analysis of the correlations, therefore, hereafter the weak EOC is also called the EOC.
This class of activation functions has the interesting property of preserving the variance across layers when the network is initialized on the EOC. We show in Proposition 1 below that, in the EOC regime, the correlations converge to 1 at a slower rate (slower than exponential). We only present the result for ReLU but the generalization to the whole class is straightforward.

**Example: ReLU**: The EOC is reduced to the singleton $(\sigma_b^2, \sigma_w^2) = (0, 2)$, hence we should initialize ReLU networks using the parameters $(\sigma_b^2, \sigma_w^2) = (0, 2)$. This result coincides with the recommendation in (He et al., 2015) whose objective was to make the variance constant as the input propagates but who did not analyze the propagation of the correlations. (Klambauer et al., 2017) performed a similar analysis by using the 'Scaled Exponential Linear Unit' activation (SELU) that makes it possible to center the mean and normalize the variance of the post-activation $\phi(y)$. The propagation of the correlations was not discussed therein either.
Figure 2(b) displays the correlation function $f$ for two different sets of parameters $(\sigma_b, \sigma_w)$. The blue graph corresponds to the EOC $(\sigma_b^2, \sigma_w^2) = (0, 2)$, and the red one corresponds to an ordered phase $(\sigma_b, \sigma_w) = (1, 1)$.
In the next result, we show that a fully connected feedforward ReLU network initialized on the EOC (weak sense) acts as if it has residual connections in terms of correlation propagation. This could potentially explain why training ReLU is faster on the EOC (see experimental results). We further show that the correlations converge to 1 at a polynomial rate of $1/l^2$ on the EOC instead of an exponential rate in the ordered phase.



(a) Tanh (b) ReLU (c) ELU

*Figure 3.* EOC curves for different activation functions (red dashed line). For smooth activation functions (Figures (a) and (c)), the EOC is a curve in the plane $(\sigma_b, \sigma_w)$, while it is reduced to a single point for ReLU.

**Proposition 1** (EOC acts as Residual connections)**.** *Consider a ReLU network with parameters $(\sigma_b^2, \sigma_w^2) = (0, 2) \in EOC$ and correlations $c_{ab}^l$. Consider also a ReLU network with simple residual connections given by*

$$\overline{y}_i^l(a) = \overline{y}_i^{l-1}(a) + \sum_{j=1}^{N_{l-1}} \overline{W}_{ij}^l \phi(\overline{y}_j^{l-1}(a)) + \overline{B}_i^l$$

*where $\overline{W}_{ij}^l \sim \mathcal{N}(0, \frac{\overline{\sigma}_w^2}{N_{l-1}})$ and $\overline{B}_i^l \sim \mathcal{N}(0, \overline{\sigma}_b^2)$. Let $\overline{c}_{ab}^l$ be the corresponding correlation. Then, for any $\overline{\sigma}_w > 0$ and $\overline{\sigma}_b = 0$, there exists a constant $\gamma > 0$ such that*

$$1 - c_{ab}^l \sim \gamma(1 - \overline{c}_{ab}^l) \sim \frac{9\pi^2}{2l^2} \quad as \quad l \to \infty$$

### 3.2. Smooth activation functions

We show that smooth activation functions provide better signal propagation through the network. We start by a result on the existence of the EOC.

**Proposition 2.** *Let $\phi \in \mathcal{D}_g^1$ be non ReLU-like such that $\phi(0) = 0$ and $\phi'(0) \neq 0$. Assume that $V[\phi]$ is non-decreasing and $V[\phi']$ is non-increasing. Let $\sigma_{max} := \sqrt{\sup_{x \geq 0} |x - \frac{V[\phi](x)}{V[\phi'](x)}|}$ and for $\sigma_b < \sigma_{max}$ let $q_{\sigma_b}$ be the smallest fixed point of the function $\sigma_b^2 + \frac{V[\phi]}{V[\phi']}$. Then we have $EOC = \{(\sigma_b, \frac{1}{\sqrt{\mathbb{E}[\phi'(\sqrt{q_{\sigma_b}}Z)^2]}}) : \sigma_b < \sigma_{max}\}$.*

*Example :* Tanh and ELU (defined by $\phi_{ELU}(x) = x$ for $x \geq 0$ and $\phi_{ELU}(x) = e^x - 1$ for $x < 0$) satisfy all conditions of Proposition 2. We prove in the Appendix that SiLU (a.k.a Swish) has an EOC.
Using Proposition 2, we propose Algorithm 1 to determine the EOC curves.
Figure 3 shows the EOC curves for different activation functions. For ReLU, the EOC is reduced to a point while smooth activation functions have an EOC curve (ELU is a smooth approximation of ReLU).
A natural question which arises from the analysis above is whether we can have $\sigma_{max} = \infty$. The answer is yes for the following large class of 'Tanh-like' activation functions.

**Algorithm 1** EOC curve

```
Input: $\phi$ satisfying conditions of Proposition 2, $\sigma_b$
Initialize $q = 0$
while $q$ has not converged do
    $q = \sigma_b^2 + \frac{V[\phi](q)}{V[\phi'](q)}$
end while
return $(\sigma_b, \frac{1}{\sqrt{V[\phi'](q)}})$
```

**Definition 4** (Tanh-like activation functions)**.** *Let $\phi \in \mathcal{D}^2(\mathbb{R}, \mathbb{R})$. $\phi$ is Tanh-like if*

1. *$\phi$ bounded, $\phi(0) = 0$, and for all $x \in \mathbb{R}$, $\phi'(x) \geq 0$, $x\phi''(x) \leq 0$ and $x\phi(x) \geq 0$.*
2. *There exist $\alpha > 0$ such that $|\phi'(x)| \gtrsim e^{-\alpha|x|}$ for large $x$ (in norm).*

**Lemma 4.** *Let $\phi$ be a Tanh-like activation function, then $\phi$ satisfies all conditions of Proposition 2 and $EOC = \{(\sigma_b, \frac{1}{\sqrt{\mathbb{E}[\phi'(\sqrt{q}Z)^2]}}) : \sigma_b \in \mathbb{R}^+\}$.*

Recall that the convergence rate of the correlation to 1 for ReLU-like activations on the EOC is $\mathcal{O}(1/\ell^2)$. We can improve this rate by taking a sufficiently regular activation function. Let us first define a regularity class $\mathcal{A}$.

**Definition 5.** *Let $\phi \in \mathcal{D}_g^2$. We say that $\phi$ is in $\mathcal{A}$ if there exists $n \geq 1$, a partition $(S_i)_{1 \leq i \leq n}$ of $\mathbb{R}$ and $g_1, g_2, ..., g_n \in \mathcal{C}_g^2$ such that $\phi^{(2)} = \sum_{i=1}^n 1_{S_i} g_i$.*

This class includes activations such as Tanh, SiLU, ELU (with $\alpha = 1$). Note that $\mathcal{D}_g^k \subset \mathcal{A}$ for all $k \geq 3$.
For activation functions in $\mathcal{A}$, the next proposition shows that the correlation converges to 1 at the rate $\mathcal{O}(1/\ell)$ which is better than $\mathcal{O}(1/\ell^2)$ of ReLU-like activation functions.

**Proposition 3** (Convergence rate for smooth activations)**.** *Let $\phi \in \mathcal{A}$ such that $\phi$ is non-linear (i.e. $\phi^{(2)}$ is non-*

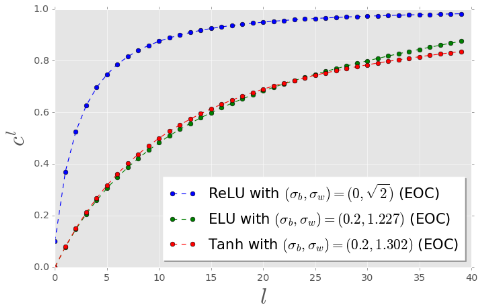


*Figure 4.* Impact of the smoothness of the activation function on the convergence of the correlations on the EOC. The convergence rate for ReLU is $\mathcal{O}(1/\ell^2)$ and $\mathcal{O}(1/\ell)$ for Tanh and ELU.

*identically zero). Then, on the EOC, we have* $1 - c^l \sim \frac{\beta_q}{l}$ *where* $\beta_q = \frac{2\mathbb{E}[\phi'(\sqrt{q}Z)^2]}{q\mathbb{E}[\phi''(\sqrt{q}Z)^2]}$

Choosing a smooth activation function is therefore better for deep neural networks since it provides deeper information propagation. This could explain for example why smooth versions of ReLU such as ELU perform better (Clevert et al., 2016) (see experimental results). Figure 3.2 shows the evolution of the correlation through the network layers for different activation functions. For function in $\mathcal{A}$ (Tanh and ELU), the graph shows a rate of $\mathcal{O}(1/\ell)$ as expected compared to $\mathcal{O}(1/\ell^2)$ for ReLU.

So far, we have discussed the impact of the EOC and the smoothness of the activation function on the behaviour of $c^l$. We now refine this analysis by studying $\beta_q$ as a function of $(\sigma_b, \sigma_w)$. We also show that $\beta_q$ plays a more important role in the information propagation process. Indeed, we show that $\beta_q$ controls the propagation of the correlation and the back-propagation of the Gradients. For the back-propagation part, we use the approximation that the weights used during forward propagation are independent of the weights used during backpropagation. This simplifies the calculations for the gradient backpropagation; see (Schoenholz et al., 2017) for details and (Yang, 2019) for a theoretical justification.

**Proposition 4.** *Let* $\phi \in \mathcal{A}$ *be a non-linear activation function such that* $\phi(0) = 0$, $\phi'(0) \neq 0$. *Assume that* $V[\phi]$ *is non-decreasing and* $V[\phi'])$ *is non-increasing, and let* $\sigma_{max} > 0$ *be defined as in Proposition 2. Let* $E$ *be a differentiable loss function and define the gradient with respect to the* $l^{th}$ *layer by* $\frac{\partial E}{\partial y^l} = (\frac{\partial E}{\partial y^l_i})_{1 \leq i \leq N_l}$ *and let* $\tilde{Q}^l_{ab} = \mathbb{E}[\frac{\partial E}{\partial y^l(a)}^T \frac{\partial E}{\partial y^l(b)}]$ *(Covariance matrix of the gradients during backpropagation). Recall that* $\beta_q = \frac{2\mathbb{E}[\phi'(\sqrt{q}Z)^2]}{q\mathbb{E}[\phi''(\sqrt{q}Z)^2]}$.

*Then, for any* $\sigma_b < \sigma_{max}$, *by taking* $(\sigma_b, \sigma_w) \in EOC$ *we have*

- $\sup_{x \in [0,1]} |f(x) - x| \leq \frac{1}{\beta_q}$
- *For* $l \geq 1$, $|\frac{\mathrm{Tr}(\tilde{Q}^l_{ab})}{\mathrm{Tr}(\tilde{Q}^{l+1}_{ab})} - 1| \leq \frac{2}{\beta_q}$

*Moreover, we have*

$$\lim_{\substack{\sigma_b \to 0 \\ (\sigma_b, \sigma_w) \in EOC}} \beta_q = \infty.$$

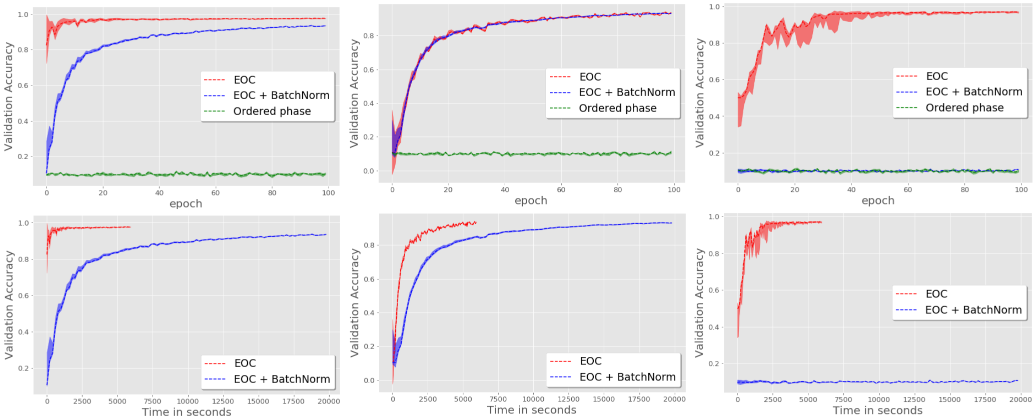
The result of Proposition 4 suggests that by taking small $\sigma_b$, we can achieve two important things. First, it makes the function $f$ close to the identity function, this slows further the convergence of the correlations to 1, i.e., the information propagates deeper inside the network. Note that the only activation functions satisfying $f(x) = x$ for all $x \in [0, 1]$ are linear functions which are not useful. Second, it makes the Trace of the covariance matrix of the gradients approximately constant through layers, which means, we avoid vanishing of the information during backpropagation (More precisely, we preserve the overall spectrum of the covariance matrix since the Trace is the sum of the eigenvalues).
We also have $\lim_{\sigma_b \to 0} q = 0$ so that if $\sigma_b$ too small then $y^l(a) \approx 0$. Hence, a trade-off has to be taken into account when initializing on the EOC. Using Proposition 4, we can deduce the maximal depth to which the correlations can propagate without being within a distance $\epsilon$ to 1. Indeed, we have for all $l$, $|c^{l+1} - c^l| \leq \frac{1}{\beta_q}$, therefore for $L \geq 1$, $|c^L - c^0| \leq \frac{L}{\beta_q}$. Assuming $c^0 < c < 1$ for all inputs where $c$ is a constant, the maximal depth we can reach without loosing $(1 - \epsilon) \times 100\%$ of the information is $L_{max} = \lfloor \beta_q(1 - c - \epsilon) \rfloor$, this satisfies $\lim_{\sigma_b \to 0} L_{max} = \infty$.

**Choice of $\sigma_b$ on the Edge of Chaos :** Given a network of depth $L$, it follows that selecting a value of $\sigma_b$ on the EOC such that $\beta_q \approx L$ appears appropriate.

We verify numerically the benefits of this rule in the next section.
Note that ReLU-like activation functions do not satisfy conditions of Proposition 4. The next lemma gives easy-to-verify sufficient conditions for Proposition 4.

**Lemma 5.** *Let* $\phi \in \mathcal{A}$ *such that* $x\phi(x)\phi'(x) \geq 0$ *and* $\phi(x)\phi''(x) \leq 0$ *for all* $x \in \mathbb{R}$. *Then,* $\phi$ *satisfies all conditions of Proposition 4.*

*Example*: Tanh and ELU satisfy all conditions of Lemma 5. This may partly explain why ELU performs experimentally better than ReLU (see next section). Another example is an activation function of the form $\lambda x + \beta \mathrm{Tanh}(x)$ where $\lambda, \beta \in \mathbb{R}$. We check the performance of these activations in the next section.



(a) ELU (b) ReLU (c) Tanh

*Figure 5.* 100 epochs of the training curve (test accuracy) for different activation functions for depth 200 and width 300 using SGD. The red curves correspond to the EOC, the green ones corresponds to an ordered phase, and the blue curves corresponds to an Initialization on the EOC plus a Batch Normalization after each layer. The upper figures show the test accuracies with respect to the epochs while the lower figures show the accuracies with respect to time.

## 4. Experiments

In this section, we demonstrate empirically the theoretical results established above. We show that:

- For deep networks, only an initialization on the EOC could make the training possible, and the initialization on the EOC performs better than Batch Normalization.
- Smooth activation functions in the sense of Proposition 3 perform better than ReLU-like activation, especially for very deep networks.
- Choosing the right point on the EOC further accelerates the training.

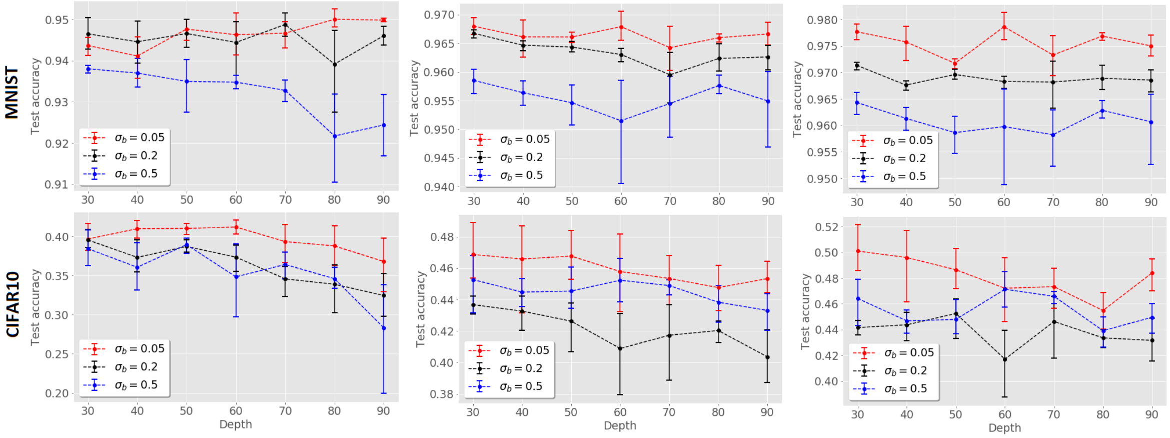
*Table 1.* Test accuracies for width 300 and depth 200 with different activation function on MNIST and CIFAR10 after 100 epochs

| **MNIST** | EOC | EOC + BN | ORD PHASE |
|---|---|---|---|
| RELU | **93.57± 0.18** | 93.11± 0.21 | 10.09± 0.61 |
| ELU | **97.62± 0.21** | 93.41± 0.3 | 10.14± 0.51 |
| TANH | **97.20± 0.3** | 10.74± 0.1 | 10.02± 0.13 |

| **CIFAR10** | EOC | EOC + BN | ORD PHASE |
|---|---|---|---|
| RELU | **36.55± 1.15** | 35.91± 1.52 | 9.91± 0.93 |
| ELU | **45.76± 0.91** | 44.12± 0.93 | 10.11± 0.65 |
| TANH | **44.11± 1.02** | 10.15± 0.85 | 9.82± 0.88 |

We demonstrate empirically our results on the MNIST and CIFAR10 datasets for depths $L$ between 10 and 200 and width 300. We use SGD and RMSProp for training. We performed a grid search between $10^{-6}$ and $10^{-2}$ with exponential step of size 10 to find the optimal learning rate. For SGD, a learning rate of $\sim 10^{-3}$ is nearly optimal for $L \leq 150$, for $L > 150$, the best learning rate is $\sim 10^{-4}$. For RMSProp,$10^{-5}$ is nearly optimal for networks with depth $L \leq 200$ (for deeper networks, $10^{-6}$ gives better results). We use a batchsize of 64.

**Initialization on the Edge of Chaos**. We initialize randomly the network by sampling $W^l_{ij} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_w/N_{l-1})$ and $B^l_i \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_b)$. Figure 5 shows that the initialization on the EOC dramatically accelerates the training for ELU, ReLU and Tanh. The initialization in the ordered phase (here we used $(\sigma_b, \sigma_w) = (1, 1)$ for all activations) results in the optimization algorithm being stuck eventually at a very poor test accuracy of $\sim 0.1$ (equivalent to selecting the output uniformly at random). Figure 5 also shows that EOC combined to BatchNorm results in a worse learning curve and dramatically increases the training time. Note that it is crucial here to initialize BatchNorm parameters to $\alpha = 1$ and $\beta = 0$ in order to keep our analysis on the forward propagation on the EOC valid for networks with BatchNorm. Table 4 presents test accuracy after 100 epochs for different activation functions and different training methods (EOC, EOC+BatchNorm, Ordered phase) on MNIST and CIFAR10. For all activation functions but Softplus, EOC initialization leads to the best performance. Adding BatchNorm to the EOC initialization makes the training worse, this can be explained the fact that parameters $\alpha$ and $\beta$ are also modified during the first backpropagation. This invalidates the EOC results for gradient backpropagation (see proof of Proposition 4).



(a) Epoch 10 (b) Epoch 50 (c) Epoch 100

*Figure 6.* Test accuracies for Tanh network with depths between 30 and 90 and width 300 using different points on the EOC.

*Table 2.* Test accuracies for width 300 and depth 200 with different activation function on MNIST and CIFAR10 after 10, 50 and 100 epochs

| **MNIST** | Epoch 10 | Epoch 50 | Epoch 100 |
|---|---|---|---|
| ReLU | 66.76± 1.95 | 88.62± 0.61 | 93.57± 0.18 |
| ELU | **96.09± 1.55** | **97.21± 0.31** | **97.62± 0.21** |
| Tanh | 89.75± 1.01 | 96.51± 0.51 | 97.20± 0.3 |

| **CIFAR10** | Epoch 10 | Epoch 50 | Epoch 100 |
|---|---|---|---|
| ReLU | 26.46± 1.68 | 33.74± 1.21 | 36.55± 1.15 |
| ELU | **35.95± 1.83** | **45.55± 0.91** | **47.76± 0.91** |
| Tanh | 34.12± 1.23 | 43.47± 1.12 | 44.11± 1.02 |

**Impact of the smoothness of the activation function on the training.** Table 4 shows the test accuracy at different epochs for ReLU, ELU, Tanh. Smooth activation functions perform better than ReLU. More experimental results with RMSProp and other activation functions of the form $x + \alpha \text{Tanh}(x)$ are provided in the supplementary material.

**Selection of a point on the EOC.** We have showed that a sensible choice is to select $\sigma_b$ such that $L \sim \beta_q$ on the EOC. Figure 6 shows test accuracy of a Tanh network for different depths using $\sigma_b \in \{0.05, 0.2, 0.5\}$. With $\sigma_b = 0.05$, we have $\beta_q \sim 50$. We see for depth 50, the red curve ($\sigma_b = 0.05$) is the best. For other depths $L$ between 30 and 90, $\sigma_b = 0.05$ is the value that makes $\beta_q$ the closest to $L$ among $\{0.05, 0.2, 0.5\}$, which explains why the red curve is approximately better for all depths between 30 and 90. To further confirm this finding, we search numerically for the best $\sigma_b \in \{2k \times 10^{-2} : k \in [1, 50]\}$ for depths $30, 100, 200$. Table 4 shows the results.

*Table 3.* Best test accuracy achieved after 100 epochs with Tanh on MNIST

| Depth | $L = 30$ | $L = 50$ | $L = 200$ |
|---|---|---|---|
| $2k \times 10^{-2}$ | 0.080 | 0.040 | 0.020 |
| with Rule $\beta_q \approx L$ | 0.071 | 0.030 | 0.022 |

## 5. Discussion

The Gaussian process approximation of Deep Neural Networks was used by (Schoenholz et al., 2017) to show that very deep Tanh networks are trainable only on the EOC. We give here a comprehensive analysis of the EOC for a large class of activation functions. We also prove that smoothness plays a major role in terms of signal propagation. Numerical results in Table 4 confirm this finding. Moreover, we introduce a rule to choose the optimal point on the EOC, this point is a function of the depth. As the depth goes to infinity (e.g. $L = 400$), we need smaller $\sigma_b$ to achieve the best signal propagation. However, the limiting variance $q$ also becomes close to zero as $\sigma_b$ goes to zero. To avoid this problem, one possible solution is to change the activation function to ensure that the coefficient $\beta_q$ becomes large independently of the choice of $\sigma_b$ on the EOC (see supplementary material).

Our results have implications for Bayesian neural networks which have received renewed attention lately; see, e.g., (Hernandez-Lobato and Adams, 2015) and (Lee et al., 2018). They indeed indicate that, if one assigns i.i.d. Gaussian prior distributions to the weights and biases, we need to select not only the prior parameters $(\sigma_b, \sigma_w)$ on the EOC but also an activation function satisfying Proposition 3 to obtain a non-degenerate prior on the induced function space.

## References

S.S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein. Deep information propagation. *5th International Conference on Learning Representations*, 2017.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

G.F. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. *Advances in Neural Information Processing Systems*, 27: 2924–2932, 2014.

B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. *30th Conference on Neural Information Processing Systems*, 2016.

R.M. Neal. *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media, 1995.

A.G. Matthews, J. Hron, M. Rowland, R.E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. *6th International Conference on Learning Representations*, 2018.

J. Lee, Y. Bahri, R. Novak, S.S. Schoenholz, J. Pennington, and J. Sohl-Dickstein. Deep neural networks as Gaussian processes. *6th International Conference on Learning Representations*, 2018.

D.A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *ICLR*, 2016.

D. Pedamonti. Comparison of non-linear activation functions for deep neural networks on mnist classification task. *arXiv 1804.02763*, 2018.

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

M. Milletarí, T. Chotibut, and P. Trevisanutto. Expectation propagation: a probabilistic view of deep feed forward networks. *arXiv:1805.08786*, 2018.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. *AISTATS*, 2011.

V. Nair and G.E. Hinton. Rectified linear units improve restricted boltzmann machines. *ICML*, 2010.

B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolution network. *arXiv:1505.00853*, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *ICCV*, 2015.

G. Klambauer, T. Unterthiner, and A. Mayr. Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

G. Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv:1902.04760*, 2019.

J. M. Hernandez-Lobato and R.P. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. *ICML*, 2015.