

---

# Training Dynamics of Deep Networks using Stochastic Gradient Descent via Neural Tangent Kernel

---

Soufiane Hayou, Arnaud Doucet, Judith Rousseau

Department of Statistics  
University of Oxford

{soufiane.hayou, arnaud.doucet, judith.rousseau}@stats.ox.ac.uk

## Abstract

Stochastic Gradient Descent (SGD) is widely used to train deep neural networks. However, few theoretical results on the training dynamics of SGD are available. Recent work by Jacot et al. (2018) has showed that training a neural network of any kind with a full batch gradient descent in parameter space is equivalent to kernel gradient descent in function space with respect to the Neural Tangent Kernel (NTK). Lee et al. (2019) built on this result to show that the output of a neural network trained using full batch gradient descent can be approximated by a linear model for wide neural networks. We show here how these results can be extended to SGD. In this case, the resulting training dynamics is given by a stochastic differential equation dependent on the NTK which becomes a simple mean-reverting process for the squared loss. When the network depth is also large, we provide a comprehensive analysis on the impact of the initialization and the activation function on the NTK, and thus on the corresponding training dynamics under SGD. We provide experiments illustrating our theoretical results.

## 1 Introduction

Deep neural networks have achieved state-of-the-art results on numerous tasks; see, e.g., Nguyen and Hein (2018), Du et al. (2018b), Zhang et al. (2017). Although the loss function is not convex, Stochastic Gradient Descent (SGD) is often used successfully to learn these models. It has been actually recently shown that for certain overparameterized deep ReLU networks, SGD converges to an optimum (Zou et al., 2018). Similar results have also been obtained for standard batch Gradient Descent (GD) (Du et al., 2018a).

The aim of this article is to provide an analysis of the training dynamics of SGD for wide and deep neural networks which will help us to better understand the impact of the initialization and activation function. The training dynamics of full batch GD is better understood and we will build upon recent work by Jacot et al. (2018) who showed that training a neural network with full batch GD in parameter space is equivalent to a functional GD i.e. a GD in a functional space with respect to a kernel called Neural Tangent Kernel (NTK). Du et al. (2019) used a similar approach to prove that full batch GD converges to global minima for shallow neural networks and Karakida et al. (2018) linked the Fisher Information Matrix to the NTK and studied its spectral distribution for infinite width networks. The infinite width limit for different architectures was studied by Yang (2019) who introduced a tensor formalism that can express most of the computations in neural networks. Lee et al. (2019) studied a linear approximation of the full batch GD dynamics based on the NTK and gave an method to approximate the NTK for different architectures. Finally, Arora et al. (2019) gives an efficient algorithm to compute exactly the NTK for convolutional architectures (Convolutional NTK or CNTK). In all of these papers, the authors used full batch GD to derive their results for different

neural networks architectures. However, this algorithm is far too expensive for most applications and one often uses SGD instead.

In parallel, the impact of the initialization and activation function on the performance of wide deep neural networks has been studied in Hayou et al. (2019), Lee et al. (2018), Schoenholz et al. (2017), Yang and Schoenholz (2017). These works analyze the forward/backward propagation of some quantities through the network at the initial step to select the initial parameters and the activation function so as to ensure a deep propagation of the information at initialization. While experimental results in these papers suggest that such selection also leads to overall better training procedures (i.e. beyond the initialization step), it remains unexplained why this is the case.

We extend here the results of Jacot et al. (2018) and show that the NTK also plays a major role in the training dynamics when SGD is used instead of full batch GD. Moreover, we provide a comprehensive study of the impact of the initialization and the activation function on the NTK and therefore on the resulting training dynamics for wide and deep networks. In particular, we show that an initialization known as the Edge of Chaos (Yang and Schoenholz, 2017) leads to better training dynamics and that a class of smooth activation functions discussed in (Hayou et al., 2019) also improves the training dynamics compared to ReLU-like activation functions. We illustrate these theoretical results through simulations. All the proofs are detailed in the Supplementary Material which also includes additional theoretical and experimental results.

## 2 Neural Networks and Neural Tangent Kernel

### 2.1 Setup and notations

Consider a neural network model consisting of  $L$  layers  $(y^l)_{1 \leq l \leq L}$ , with  $y^l : \mathbb{R}^{n_{l-1}} \rightarrow \mathbb{R}^{n_l}$ ,  $n_0 = d$  and let  $\theta = (\theta^l)_{1 \leq l \leq L}$  be the flattened vector of weights and bias indexed by the layer's index and  $p$  be the dimension of  $\theta$ . Recall that  $\theta^l$  has dimension  $n_l + 1$ . The output  $f$  of the neural network is given by some transformation  $s : \mathbb{R}^{n_L} \rightarrow \mathbb{R}^o$  of the last layer  $y^L(x)$ ;  $o$  being the dimension of the output (e.g. number of classes for a classification problem). For any input  $x \in \mathbb{R}^d$ , we thus have  $f(x, \theta) = s(y^L(x)) \in \mathbb{R}^o$ . As we train the model,  $\theta$  changes with time  $t$  and we denote by  $\theta_t$  the value of  $\theta$  at time  $t$  and  $f_t(x) = f(x, \theta_t) = (f_j(x, \theta_t), j \leq o)$ . Let  $D = (x_i, y_i)_{1 \leq i \leq N}$  be the data set and let  $\mathcal{X} = (x_i)_{1 \leq i \leq N}$ ,  $\mathcal{Y} = (y_i)_{1 \leq i \leq N}$  be the matrices of input and output respectively, with dimension  $d \times N$  and  $o \times N$ . For any function  $g : \mathbb{R}^{d \times o} \rightarrow \mathbb{R}^k$ ,  $k \geq 1$ , we denote by  $g(\mathcal{X}, \mathcal{Y})$  the matrix  $(g(x_i, y_i))_{1 \leq i \leq N}$  of dimension  $k \times N$ .

Jacot et al. (2018) studied the behaviour of the output of the neural network as a function of the training time  $t$  when the network is trained using a gradient descent algorithm. Lee et al. (2019) built on this result to linearize the training dynamics. We recall hereafter some of these results.

For a given  $\theta$ , the empirical loss is given by  $\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \ell(f(x_i, \theta), y_i)$ . The full batch GD algorithm is given by

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta \nabla_{\theta} \mathcal{L}(\hat{\theta}_t) \quad (1)$$

where  $\eta > 0$  is the learning rate.

Let  $T > 0$  be the training time and  $N_s = T/\eta$  be the number of steps of the discrete GD (1). The continuous time system equivalent to (1) with step  $\Delta t = \eta$  is given by

$$d\theta_t = -\nabla_{\theta} \mathcal{L}(\theta_t) dt \quad (2)$$

This differs from the result by Lee et al. (2019) since we use a discretization step of  $\Delta t = \eta$ . Consider times  $t_k = k\eta$  for  $k \in [0, T/\eta]$ . The following lemma, proved in the supplementary material, controls the difference between the time-continuous system and its discretization.

**Lemma 1** (Discretization Error for Full Batch Gradient Descent). *Assume  $\nabla_{\theta} \mathcal{L}$  is  $C$ -lipschitz and let  $\eta \in (0, 1)$ , then there exists  $C' > 0$  that depends only on  $C$  and  $T$  such that*

$$\sup_{k \in [0, T/\eta]} \|\theta_{t_k} - \hat{\theta}_k\| \leq \eta C'.$$

As in Lee et al. (2019), Equation (2) can be re-written as

$$d\theta_t = -\frac{1}{N} \nabla_{\theta} f(\mathcal{X}, \theta_t)^T \nabla_z \ell(f(\mathcal{X}, \theta_t), \mathcal{Y}) dt$$

where  $\nabla_{\theta} f(\mathcal{X}, \theta_t)$  is a matrix of dimension  $oN \times p$  and  $\nabla_z \ell(f(\mathcal{X}, \theta_t), \mathcal{Y})$  is the flattened vector of dimension  $oN$  constructed from the concatenation of the vectors  $\nabla_z \ell(z, y_i)|_{z=f(x_i, \theta_t)}, i \leq N$ . As a result, the output function  $f_t(x)$  satisfies the following ordinary differential equation

$$df_t(x) = \nabla_{\theta} f(x, \theta_t) d\theta_t = -\frac{1}{N} \nabla_{\theta} f(x, \theta_t) \nabla_{\theta} f(\mathcal{X}, \theta_t)^T \nabla_z \ell(f_t(\mathcal{X}), \mathcal{Y}) dt \in \mathbb{R}^o \quad (3)$$

The Neural Tangent Kernel (NTK)  $K_{\theta}^L$  is defined as the  $o \times o$  dimensional kernel satisfying: for all  $x, x' \in \mathbb{R}^d$ ,

$$\begin{aligned} K_{\theta}^L(x, x') &= \nabla_{\theta} f(x, \theta_t) \nabla_{\theta} f(x', \theta_t)^T \in \mathbb{R}^{o \times o} \\ &= \sum_{l=1}^L \nabla_{\theta^l} f(x, \theta_t) \nabla_{\theta^l} f(x', \theta_t)^T. \end{aligned} \quad (4)$$

We also define  $K_{\theta_t}^L(\mathcal{X}, \mathcal{X})$  as the  $oN \times oN$  matrix defined blockwise by

$$K_{\theta_t}^L(\mathcal{X}, \mathcal{X}) = \begin{pmatrix} K_{\theta_t}^L(x_1, x_1) & K_{\theta_t}^L(x_1, x_2) & \cdots & K_{\theta_t}^L(x_1, x_N) \\ K_{\theta_t}^L(x_2, x_1) & \cdots & \cdots & K_{\theta_t}^L(x_2, x_N) \\ \cdots & \cdots & \cdots & \cdots \\ K_{\theta_t}^L(x_N, x_1) & K_{\theta_t}^L(x_N, x_2) & \cdots & K_{\theta_t}^L(x_N, x_N) \end{pmatrix}$$

By applying equation (3) to the vector  $\mathcal{X}$ , one obtains

$$df_t(\mathcal{X}) = -\frac{1}{N} K_{\theta_t}^L(\mathcal{X}, \mathcal{X}) \nabla_z \ell(f_t(\mathcal{X}), \mathcal{Y}) dt, \quad (5)$$

meaning that for all  $j \leq N$   $df_t(x_j) = -\frac{1}{N} K_{\theta_t}^L(x, \mathcal{X}) \nabla_z \ell(f_t(\mathcal{X}), \mathcal{Y}) dt$ . Zou et al. (2018) proved that while training a wide neural network with ReLU, we do not move far from the initialization point  $\theta_0$ . Following these results, Lee et al. (2018) suggested using a first order linear approximation of the dynamics of  $f_t$  as an approximation to the real training dynamics. This linearized version is given by

$$f_t^{\text{lin}}(x) = f_0(x) + \nabla_{\theta} f_0(x)(\theta_t - \theta_0).$$

Using this linearized version, the dynamics of  $\theta_t$  and  $f_t$  are given by

$$d\theta_t = -\frac{1}{N} \nabla_{\theta} f_0(\mathcal{X})^T \nabla_z \ell(f_t^{\text{lin}}(\mathcal{X}), \mathcal{Y}) dt, \quad df_t^{\text{lin}}(x) = -\frac{1}{N} K_{\theta_0}^L(x, \mathcal{X}) \nabla_z \ell(f_t^{\text{lin}}(\mathcal{X}), \mathcal{Y}) dt.$$

When tested empirically on different models and datasets, Lee et al. (2018) showed that this approximation captures remarkably well the training dynamics. However, in practice, one usually never computes the exact gradient of the empirical loss due to the high computational cost but just an unbiased version of this later. We address this issue hereafter by proving that, even with SGD, the training dynamics follow a simple Stochastic Differential Equation (SDE) that can be explicitly solved in some scenarios.

## 2.2 Training with Stochastic Gradient Descent

In this section, we use an approximation of the SGD dynamics by a diffusion process. We assume implicitly the existence of the triplet  $(\Omega, \mathbb{P}, \mathcal{F})$  where  $\Omega$  is the probability space,  $\mathbb{P}$  is a probability measure on  $\Omega$ , and  $\mathcal{F}$  is the natural filtration of the Brownian motion. Under boundedness conditions (see the supplementary material), when using SGD, the gradient update can be seen as a GD with a Gaussian noise (Hu et al., 2018; Li et al., 2017). More precisely, let  $S = o(N)$  be the batchsize. The SGD update is given by

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta \nabla_{\theta} \mathcal{L}^{(S)}(\hat{\theta}_t), \quad (6)$$

where  $\mathcal{L}^{(S)} = \frac{1}{S} \sum_{i=1}^S \ell(f(\tilde{x}_i, \theta), \tilde{y}_i)$  with  $(\tilde{x}_i, \tilde{y}_i)_{1 \leq i \leq S}$  being a randomly selected batch of size  $S$ .

Combining Hu et al. (2018) and Li et al. (2017), in the time-continuous limit, the previous dynamics can be seen as a discretization of the following SDE

$$d\theta_t = -\nabla_{\theta} \mathcal{L}(\theta_t) dt + \sqrt{\frac{\eta}{S}} \Sigma(\theta_t)^{\frac{1}{2}} dW_t, \quad (7)$$

with time step  $\Delta = \eta$ , and where  $\Sigma(\theta_t)^{\frac{1}{2}}$  is the square-root matrix of  $\Sigma(\theta_t) = \text{Cov}(\nabla_{\theta} \ell(f(X_1, \theta_t), Y_1))$  and  $(W_t)_{t \geq 0}$  a standard Brownian motion.

Since the dynamics of  $\theta_t$  are described by an SDE, the dynamics of  $f_t$  can also be described by an SDE which can be obtained from Itô's lemma, see Section 2.1 of the supplementary material.

**Proposition 1.** Under the dynamics of the SDE (16), the vector  $f_t(\mathcal{X})$  is the solution of the following SDE

$$df_t(\mathcal{X}) = [-\frac{1}{N}K_{\theta_t}^L(\mathcal{X}, \mathcal{X})\nabla_z \ell(f_t(\mathcal{X}), Y) + \frac{1}{2}\frac{\eta}{S}\Gamma_t(\mathcal{X})]dt + \sqrt{\frac{\eta}{S}}\nabla_{\theta}f(\mathcal{X}, \theta_t)\Sigma(\theta_t)^{\frac{1}{2}}dW_t \quad (8)$$

where  $\Gamma_t(\mathcal{X})$  is the concatenated vector of  $(\Gamma_t(x) = (\text{Tr}(\Sigma(\theta_t)^{\frac{1}{2}}\nabla_2 f_i(x, \theta_t)\Sigma(\theta_t)^{\frac{1}{2}}))_{1 \leq i \leq o})_{x \in \mathcal{X}}$  and  $\nabla_2 f_i(x, \theta)$  is the Hessian of  $f_i$  ( $i^{\text{th}}$  component of  $f$ ) with respect to  $\theta$ .

With the quadratic loss  $\ell(z, y) = \frac{1}{2}\|z - y\|^2$ , the SDE (17) is equivalent to

$$df_t(\mathcal{X}) = [-\frac{1}{N}K_{\theta_t}^L(\mathcal{X}, \mathcal{X})(f_t(\mathcal{X}) - \mathcal{Y}) + \frac{1}{2}\frac{\eta}{S}\Gamma_t(\mathcal{X})]dt + \sqrt{\frac{\eta}{S}}\nabla_{\theta}f(\mathcal{X}, \theta_t)\Sigma(\theta_t)^{\frac{1}{2}}dW_t. \quad (9)$$

This is an Ornstein-Uhlenbeck process (mean-reverting process) with time dependent parameters. The additional term  $\Gamma_t$  is due to the randomness of the mini-batch, it can be seen as a regularization term and could partly explain why SGD gives better generalization errors compared to GD (Kubo et al. (2019), Lei et al. (2018)).

#### Dynamics of $f_t$ for wide FeedForward neural networks :

In the case of a fully connected feedforward neural network (FFNN hereafter) of depth  $L$  and widths  $n_1, n_2, \dots, n_L$ , Jacot et al. (2018) proved that, with GD, the kernel  $K_{\theta_t}^L$  converges to a kernel  $K^L$  that depends only on  $L$  (number of layers) for all  $t < T$  when  $n_1, n_2, \dots, n_L \rightarrow \infty$ , where  $T$  is an upper bound on the training time, under the technical assumption  $\int_0^T \|\nabla_z \ell(f_t(\mathcal{X}, \mathcal{Y}))\|_2 dt < \infty$  almost surely with respect to the initialization. For SGD, we assume that the convergence result of the NTK holds true as well, this is illustrated empirically in Section 4 but we leave the theoretical proof for future work. With this approximation, the dynamics of  $f_t(\mathcal{X})$  for wide networks is given by

$$df_t(\mathcal{X}) = -\frac{1}{N}\hat{K}^L(f_t(\mathcal{X}) - M_t)dt + \sqrt{\frac{\eta}{S}}\nabla_{\theta}f(\mathcal{X}, \theta_t)\Sigma(\theta_t)^{\frac{1}{2}}dW_t,$$

where  $\hat{K}^L = K^L(\mathcal{X}, \mathcal{X})$  and  $M_t = Y - \frac{\eta N}{2S}(\hat{K}^L)^{-1}\Gamma_t(\mathcal{X})$ . This is an Ornstein-Uhlenbeck process whose closed-form expression is given by

$$f_t(\mathcal{X}) = e^{-\frac{t}{N}\hat{K}^L}f_0(\mathcal{X}) + (I - e^{-\frac{t}{N}\hat{K}^L})\mathcal{Y} + A_t(\mathcal{X}) \quad (10)$$

where  $A_t(\mathcal{X}) = -\frac{\eta}{2S}\int_0^t e^{-\frac{t-s}{N}\hat{K}^L}\Gamma_s(\mathcal{X})ds + \sqrt{\frac{\eta}{S}}\int_0^t e^{-\frac{t-s}{N}\hat{K}^L}\nabla_{\theta}f(\mathcal{X}, \theta_s)\Sigma(\theta_s)^{\frac{1}{2}}dW_s$ ; see supplementary material for the proof. So for any (test) input  $x \in \mathbb{R}^d$ , we have

$$f_t(x) = f_0(x) + K^L(x, \mathcal{X})(\hat{K}^L)^{-1}(I - e^{-\frac{t}{N}\hat{K}^L})(\mathcal{Y} - f_0(\mathcal{X})) + Z_t(x) + R_t(x), \quad (11)$$

where  $R_t(x) = \sqrt{\frac{\eta}{S}}\int_0^t [K^L(x, \mathcal{X})(\hat{K}^L)^{-1}(e^{-\frac{t}{N}\hat{K}^L} - I)\nabla_{\theta}f(\mathcal{X}, \theta_s) + \nabla_{\theta}f(x, \theta_s)]\Sigma(\theta_s)^{\frac{1}{2}}dW_s$  and  $Z_t(x) = \frac{\eta}{2S}\left[\int_0^t \Gamma_s(x)ds + \int_0^t K(x, \mathcal{X})(\hat{K}^L)^{-1}(I - e^{-\frac{t-s}{N}\hat{K}^L})\Gamma_s(\mathcal{X})ds\right]$ .

The infinite width approximation with squared loss shows precisely how the kernel  $K^L$  controls the training speed and the generalization of the model through equations (10) and (11). This also holds true for other loss functions (e.g. cross-entropy) as NTK is also involved in the training dynamics (17). For deep neural networks, understanding the behaviour of  $K^L$  as  $L$  goes to infinity is thus crucial to understand the training dynamics. More precisely, two concepts are crucial for good training : invertibility of  $K^L$  since only an invertible kernel can make the training possible (equations (17) and (10)) and expressiveness of  $K^L$  since it is directly involved in the generalization function (equation (11)).

### 3 Impact of the Initialization and the Activation function on the Neural Tangent Kernel

In this section we study the impact of the initialization and the activation function on the limiting NTK for Fully-connected Feed-forward neural networks (FFNN). More precisely, we prove that only

an initialization on the Edge of Chaos (EOC) leads to an invertible NTK for deep neural networks. All other initializations will lead to a trivial non-invertible NTK. We also show that the smoothness of the activation function plays a major role in the behaviour of NTK. To simplify notations, we restrict ourselves to the case  $s(x) = x$  and  $o = 1$ , since generalization to any function  $s$  and any  $n_L$  is straightforward.

Consider a FFNN of depth  $L$ , widths  $(n_l)_{1 \leq l \leq L}$ , weights  $w^l$  and bias  $b^l$ . For some input  $x \in \mathbb{R}^d$ , the forward propagation is given by

$$y_i^1(x) = \sum_{j=1}^d w_{ij}^1 x_j + b_i^1, \quad y_i^l(x) = \sum_{j=1}^{n_{l-1}} w_{ij}^l \phi(y_j^{l-1}(x)) + b_i^l, \quad \text{for } l \geq 2, \quad (12)$$

where  $\phi$  is the activation function.

We initialize the model with  $w_{ij}^l \stackrel{iid}{\sim} \mathcal{N}(0, \frac{\sigma_w^2}{n_{l-1}})$  and  $b_i^l \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_b^2)$ , where  $\mathcal{N}(\mu, \sigma^2)$  denotes the normal distribution of mean  $\mu$  and variance  $\sigma^2$ . For some  $x$ , we denote by  $q^l(x)$  the variance of  $y^l(x)$ . The convergence of  $q^l(x)$  as  $l$  increases is studied in Lee et al. (2018), Schoenholz et al. (2017) and Hayou et al. (2019). In particular under weak regularity conditions they prove that  $q^l(x)$  converges to a point  $q(\sigma_b, \sigma_w) > 0$  independent of  $x$  as  $l \rightarrow \infty$ . Also the asymptotic behaviour of the correlations between  $y^l(x)$  and  $y^l(x')$  for any two inputs  $x$  and  $x'$  is driven by  $(\sigma_b, \sigma_w)$ ; the authors define the EOC as the set of parameters  $(\sigma_b, \sigma_w)$  such that  $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)} Z)^2] = 1$  where  $Z \sim \mathcal{N}(0, 1)$ . Similarly the Ordered, resp. Chaotic, phase is defined by  $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)} Z)^2] < 1$ , resp.  $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)} Z)^2] > 1$ ; more details are recalled in Section 2 of the supplementary material. It turns out that the EOC plays also a crucial role on the NTK. Let us first define two classes of activation functions.

**Definition 1.** Let  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  be a measurable function. Then

1.  $\phi$  is said to be ReLU-like if there exist  $\lambda, \beta \in \mathbb{R}$  such that  $\phi(x) = \lambda x$  for  $x > 0$  and  $\phi(x) = \beta x$  for  $x \leq 0$ .
2.  $\phi$  is said to be in  $\mathcal{S}$  if  $\phi(0) = 0$ ,  $\phi$  is twice differentiable, and there exist  $n \geq 1$ , a partition  $(A_i)_{1 \leq i \leq n}$  of  $\mathbb{R}$  and infinitely differentiable functions  $g_1, g_2, \dots, g_n$  such that  $\phi^{(2)} = \sum_{i=1}^n 1_{A_i} g_i$ , where  $\phi^{(2)}$  is the second derivative of  $\phi$ .

The class of ReLU-like activations includes ReLU and Leaky-ReLU, whereas the  $\mathcal{S}$  class includes, among others, Tanh, ELU and SiLU (Swish). The following proposition establishes that any initialization on the Ordered or Chaotic phase, leads to a trivial limiting NTK as the number of layers  $L$  becomes large.

**Proposition 2** (Limiting Neural Tangent Kernel with Ordered/Chaotic Initialization). Let  $(\sigma_b, \sigma_w)$  be either in the ordered or in the chaotic phase. Then, there exist  $\lambda, \gamma > 0$  such that

$$\sup_{x, x' \in \mathbb{R}^d} |K^L(x, x') - \lambda| \leq e^{-\gamma L} \rightarrow_{L \rightarrow \infty} 0$$

As a result, as  $L$  goes to infinity,  $K^L$  converges to a constant kernel  $K^\infty(x, x') = \lambda$  for all  $x, x' \in \mathbb{R}^d$ . The training is then impossible. Indeed, we have  $K^L(\mathcal{X}, \mathcal{X}) \approx \lambda U$  where  $U$  is the matrix with all elements equal to one, i.e.  $U = Q D Q^T$  where  $Q$  is an orthogonal matrix and  $D = \text{Diag}(N-1, 0, \dots, 0)$ . Hence,  $\hat{K}^L$  is at best degenerate and asymptotically (in  $L$ ) non invertible. Also,  $e^{-\frac{t}{N} \hat{K}^L} = Q e^{-\frac{t}{N} D} Q^T$  where  $e^{-\frac{t}{N} D} = \text{Diag}(e^{-t(1-\frac{1}{N})}, 1, \dots, 1)$  so that  $e^{-\frac{t}{N} \hat{K}^L}$  does not converge to 0 as  $t$  grows, rendering the training impossible. We illustrate empirically this result in Section 4.

Recall that the (matrix) NTK for input data  $\mathcal{X}$  is given by

$$K_{\theta_t}^L(\mathcal{X}, \mathcal{X}) = \nabla_{\theta} f(\mathcal{X}, \theta_t) \nabla_{\theta} f(\mathcal{X}, \theta_t)^T = \sum_{l=1}^L \nabla_{\theta_l} f(\mathcal{X}, \theta_t) \nabla_{\theta_l} f(\mathcal{X}, \theta_t)^T$$

As shown in Schoenholz et al. (2017) and Hayou et al. (2019), an initialization on the EOC preserves the norm of the gradient as it back-propagates through the network. This means that the terms  $\nabla_{\theta_l} f(\mathcal{X}, \theta_t) \nabla_{\theta_l} f(\mathcal{X}, \theta_t)^T$  are of the same order. Hence, it is more convenient to study the average

NTK (ANTK hereafter) with respect to the number of layers  $L$ . The next proposition shows that on the EOC, the ANTK converges to an invertible kernel as  $L \rightarrow \infty$  at a polynomial rate. Moreover, by choosing an activation function in the class  $\mathcal{S}$ , we can slow the convergence of ANTK with respect to  $L$ , and therefore train deeper models. This confirms the findings in (Hayou et al., 2019).

**Proposition 3** (Neural Tangent Kernel on the Edge of Chaos). *Let  $\phi$  be a non-linear activation function and  $(\sigma_b, \sigma_w) \in \text{EOC}$ .*

1. *If  $\phi$  is ReLU-like, then for all  $x \in \mathbb{R}^d$ ,  $\frac{K^L(x, x)}{L} = \frac{\sigma_w^2 \|x\|^2}{d} + \frac{K^0(x, x)}{L}$ . Moreover, there exist  $A, \lambda \in (0, 1)$  such that*

$$\sup_{x \neq x' \in \mathbb{R}^d} \left| \frac{K^L(x, x')}{L} - \lambda \frac{\sigma_w^2}{d} \|x\| \|x'\| \right| \leq \frac{A}{L}, \quad K_\infty(x, x') = \frac{\sigma_w^2}{d} \|x\| \|x'\| (1 - (1 - \lambda) 1_{x \neq x'})$$

2. *If  $\phi$  is in  $\mathcal{S}$ , then, there exists  $q > 0$  such that  $\frac{K^L(x, x)}{L} = q + \frac{K^0(x, x)}{L} \rightarrow q$ . Moreover, there exist  $B, C, \lambda \in (0, 1)$  such that*

$$\frac{B \log(L)}{L} \leq \sup_{x \neq x' \in \mathbb{R}^d} \left| \frac{K^L(x, x')}{L} - q \right| \leq \frac{C \log(L)}{L}, \quad K_\infty(x, x') = q(1 - (1 - \lambda) 1_{x \neq x'})$$

Since  $0 < \lambda < 1$ , on the EOC there exists a matrix  $J$  invertible such that  $K^L(\mathcal{X}, \mathcal{X}) = L \times J(1 + o(1))$  as  $L \rightarrow \infty$ . Hence, although the NTK grows linearly with  $L$ , it remains asymptotically invertible. This makes the training possible for deep neural networks when initialized on the EOC, contrarywise to an initialization on the Ordered/Chaotic phase, see Proposition 2). However the limiting kernels  $K_\infty$  carry (almost) no information on  $x, x'$  and have therefore little expressive power. Interestingly the convergence rate of the ANTK to  $K_\infty$  is slow in  $L$  ( $\mathcal{O}(L^{-1})$  for ReLU-like activation functions and  $\mathcal{O}(\log(L)L^{-1})$  for activation functions of type  $\mathcal{S}$ ). This means that as  $L$  grows, the NTK remain expressive compared to the Ordered/Chaotic phase case (exponential convergence rate). This is particularly important for the generalization part (see equation 11). The  $\log(L)$  gain obtained when using smooth activation functions of type  $\mathcal{S}$  means we can train deeper neural networks with this kind of activation functions compared to the ReLU-like activation functions and could explain why ELU and Tanh tend to perform better than ReLU and Leaky-ReLU (see Section 4).

Another important feature of deep neural network which is known to be highly influential is their architecture. The next proposition shows that adding residual connections to a ReLU network causes the NTK to explode exponentially.

**Proposition 4.** *Consider the following network architecture (FFNN with residual connections)*

$$y_i^l(x) = y_i^{l-1}(x) + \sum_{j=1}^{n_{l-1}} w_{ij}^l \phi(y_j^{l-1}(x)) + b_i^l, \quad \text{for } l \geq 2. \quad (13)$$

*with initialization parameters  $\sigma_b = 0$  and  $\sigma_w > 0$ . Let  $K_{res}^L$  be the corresponding NTK. Then for all  $x \in \mathbb{R}^d$ ,  $\frac{K_{res}^L(x, x)}{\alpha_L \times 2^L} = \frac{\|x\|^2}{d} + \mathcal{O}(\gamma_L)$  and there exists  $\lambda \in (0, 1)$  such that*

$$\sup_{x \neq x' \in \mathbb{R}^d} \left| \frac{K_{res}^L(x, x')}{\alpha_L \times 2^L} - \frac{\|x\| \times \|x'\|}{d} \lambda \right| = \mathcal{O}(L^{-1}),$$

*where  $\alpha_l$  and  $\gamma_l$  are given by*

- if  $\sigma_w < \sqrt{2}$ , then  $\alpha_L = 1$  and  $\gamma_L = (\frac{1 + \sigma_w^2/2}{2})^L$
- if  $\sigma_w = \sqrt{2}$ , then  $\alpha_L = L$  and  $\gamma_L = L^{-1}$
- if  $\sigma_w > \sqrt{2}$ , then  $\alpha_L = (\frac{1 + \sigma_w^2/2}{2})^L$  and  $\gamma_L = (\frac{1 + \sigma_w^2/2}{2})^{-L}$

Proposition 4 shows that the NTK of a ReLU FFNN with residual connections explodes exponentially with respect to  $L$ . However, the normalised kernel  $K_{res}^L(x, x')/\alpha_L 2^L$  where  $x \neq x'$  converges to a limiting kernel similar to  $k_\infty$  with a rate  $\mathcal{O}(L^{-1})$  for all  $\sigma_w > 0$ . This could potentially explain why residual networks perform better than FFNN (ReLU) in many tasks when the initialization is not on the EOC. We illustrate this result in section 4.

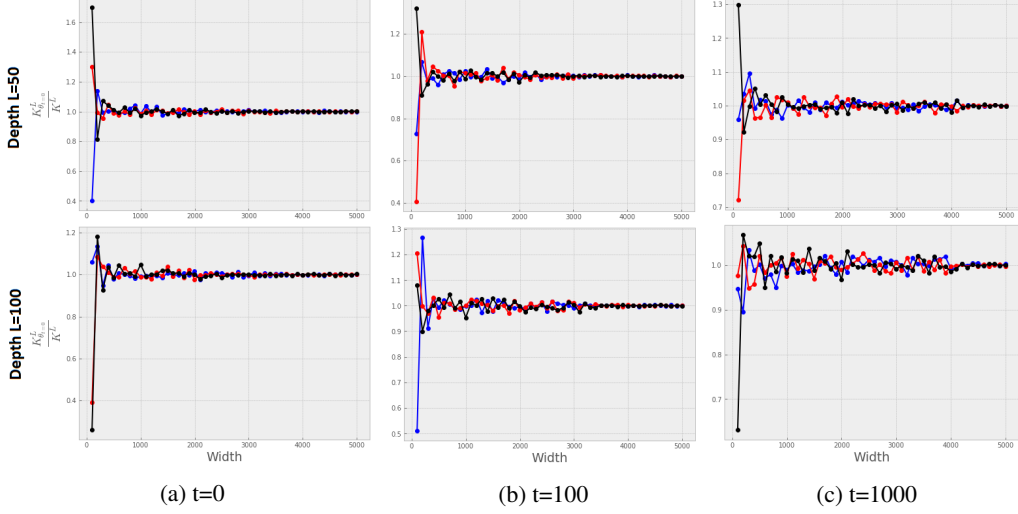


Figure 1: Ratio  $K_{\theta_t}^L / K^L$  for three randomly selected pairs from MNIST dataset as a function of width for three training times  $t = 0$ ,  $t = 100$  and  $t = 1000$  (training time is measured by SGD updates)

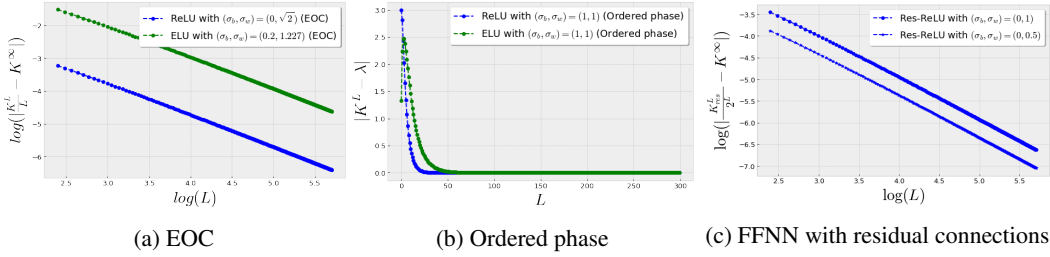


Figure 2: Convergence rates for different initializations and architectures. (a) Edge of Chaos. (b) Ordered phase. (c) Adding residual connections.

## 4 Experiments

In this section, we illustrate empirically the theoretical results obtained in the previous sections, both in terms of the existence of the convergence of  $K_{\theta_t}^L$  to  $K^L$  as  $\min_{l \leq L} n_l \rightarrow \infty$  and of its rate. Lastly, we confirm the impact of NTK on the overall performance of the model (FFNN), on MNIST and CIFAR10 datasets.

### 4.1 Convergence of $K_{\theta}^L$ to $K^L$ with SGD

For NTK calculations, we use the Neural Tangent Kernel library (Schoenholz et al., 2019) which is based on JAX. We consider FFNN with equal widths (i.e.  $n_1 = n_2 = \dots = n_L$ ) and we choose randomly three inputs from the MNIST dataset and track the value of the ratio  $K_{\theta_t}^L / K^L$  for the different widths  $\{100 \times k, 1 \leq k \leq 50\}$ , depths  $\{50, 100\}$  and training times  $\{0, 100, 1000\}$ . The training time  $t$  is measured by the number of SGD updates. In Figure 1 we observe that for both depths and the three training times, the ratio  $K_{\theta_t}^L / K^L$  converge to 1 as the width increase. However as  $t$  and  $L$  grow, the convergence of  $|K_{\theta_t}^L / K^L - 1|$  slows down.

### 4.2 Convergence rate of $K^L$ as $L$ goes to infinity

Propositions 2, 3 and 4 give theoretical convergence rates for quantities of the form  $|\frac{K^L}{\alpha_L} - k_\infty|$ . We illustrate these results in Figure 2. Figure 4a illustrates a convergence rate approximately equal to  $\mathcal{O}(L^{-1})$  for ReLU and ELU. Recall that for ELU the exact rate is  $\mathcal{O}(\log(L)L^{-1})$  but one cannot

Table 1: Test accuracy for a FFNN with width 300 and depth 300 for different activation functions on MNIST and CIFAR10. We show test accuracy after 10 epochs and 100 epochs

Activation	MNIST		CIFAR10	
	Epoch 10	Epoch 100	Epoch 10	Epoch 100
ReLU (EOC)	46.53 $\pm$ 12.01	82.11 $\pm$ 4.51	20.38 $\pm$ 1.85	35.88 $\pm$ 0.6
LReLU <sub>0.01</sub> (EOC)	48.10 $\pm$ 3.31	84.71 $\pm$ 3.39	22.62 $\pm$ 1.15	29.44 $\pm$ 4.14
LReLU <sub>0.02</sub> (EOC)	49.09 $\pm$ 3.58	84.3. $\pm$ 3.98	18.62 $\pm$ 4.56	30.78 $\pm$ 6.33
LReLU <sub>0.03</sub> (EOC)	50.94 $\pm$ 4.48	85.49 $\pm$ 2.71	21.19 $\pm$ 6.53	34.54 $\pm$ 2.32
PReLU	51.94 $\pm$ 5.51	87.49 $\pm$ 1.58	22.95 $\pm$ 3.57	36.13 $\pm$ 3.83
ELU (EOC)	<b>91.63 <math>\pm</math> 2.21</b>	<b>96.07 <math>\pm</math> 0.13</b>	<b>33.81 <math>\pm</math> 1.55</b>	<b>46.14 <math>\pm</math> 1.49</b>
Tanh (EOC)	91.16 $\pm$ 1.21	95.75 $\pm$ 0.27	32.37 $\pm$ 1.88	42.40 $\pm$ 1.13
Softplus	10.11 $\pm$ 0.09	10.13 $\pm$ 0.18	11.13 $\pm$ 0.15	11.09 $\pm$ 0.36
Sigmoid	9.85 $\pm$ 0.11	9.87 $\pm$ 0.10	10.65 $\pm$ 0.25	10.33 $\pm$ 0.17

observe experimentally the logarithmic factor. However, ELU does indeed better than ReLU (see Table 1) which might be explained by this  $\log(L)$  factor. Figure 4b demonstrates that this convergence occurs at an exponential convergence rate in the Ordered phase for both ReLU and ELU, and Figure 2c the convergence rate in the case of FFNN with residual connections. As predicted by Proposition 4, the convergence rate  $\mathcal{O}(L^{-1})$  is independent of the parameter  $\sigma_w$ .

### 4.3 Impact of the initialization and smoothness of the activation on the overall performance

We train FFNN of width 300 and depths  $L \in \{100, 200, 300\}$  with SGD and categorical cross-entropy loss. We use SGD for training with batchsize of 64 and a learning rate  $10^{-3}$  for  $L = 100$  and  $10^{-4}$  for  $L \in 200, 300$  (this learning rate was found by a grid search of exponential step size 10). For each activation function, we use an initialization on the EOC when it exists, we add the symbol (EOC) after the activation when this is satisfied. We use  $(\sigma_b, \sigma_w) = (0, \sqrt{2})$  for ReLU,  $(\sigma_b, \sigma_w) = (0.2, 1.227)$  for ELU and  $(\sigma_b, \sigma_w) = (0.2, 1.302)$  for Tanh. These values are all on the EOC (see Hayou et al. (2019) for more details). Table 1 displays the test accuracy for different activation functions on MNIST and CIFAR10 after 10 and 100 training epochs for depth 300 and width 300 (numerical results for other depths are provided in the supplementary material). Functions in class  $\mathcal{S}$  (ELU and Tanh) perform much better than ReLU-like activation functions (ReLU, Leaky-Relu- $\alpha$  with  $\alpha \in \{0.01, 0.02, 0.03\}$ ). Even with Parametric ReLU (PReLU) where the parameter of the leaky-ReLU is also learned by backpropagation, we obtain only a small improvement over ReLU. For activation functions that do not have an EOC, such as Softplus and Sigmoid, we use He initialization for MNIST and Glorot initialization for CIFAR10 (see He et al. (2015) and Glorot and Bengio (2010)). For Softplus and Sigmoid, the training algorithm is stuck at a low test accuracy  $\sim 10\%$  which is the test accuracy of a uniform random classifier with 10 classes.

## 5 Conclusion

We have shown here that the training dynamics of SGD for deep neural networks can be approximated by a SDE dependent on the NTK. This approximation sheds light on how the NTK impacts the training dynamics: it controls the training rate and the generalization function. Additionally, as the number of layers becomes very large, the NTK (resp. ANTK on the EOC) ‘forgets’ the data by converging to some limiting data-independent kernel  $K^\infty$ . More precisely, for an initialization in the Ordered/Chaotic phase, NTK converges exponentially fast to a non-invertible kernel as the number of layers goes to infinity, making training impossible. An initialization on the EOC leads to an invertible ANTK (and NTK) even for an infinite number of layers: the convergence rate is  $\mathcal{O}(L^{-1})$  for ReLU-like activation functions and  $\mathcal{O}(\log(L)L^{-1})$  for a class of smooth activation functions. We believe that the NTK is a useful tool to (partially) understand wide deep neural networks even if we are aware of the limitations of such an approach; see, e.g., Chizat and Bach (2018) and Ghorbani et al. (2019).



## References

- Arora, S., Du, S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. (2019). On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*.
- Chizat, L. and Bach, F. (2018). A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*.
- Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. (2018a). Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*.
- Du, S., Lee, J., Tian, Y., Póczos, B., and Singh, A. (2018b). Gradient descent learns one-hidden-layer CNN: Don’t be afraid of spurious local minima. *ICML*.
- Du, S., Zhai, X., Póczos, B., and Singh, A. (2019). Gradient descent provably optimizes over-parameterized neural networks. *ICLR*.
- Ghorbani, B., Mei, S., Misiakiewicz, T., and Montanari, A. (2019). Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*.
- Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *International Conference on Artificial Intelligence and Statistics*.
- Hayou, S., Doucet, A., and Rousseau, J. (2019). On the impact of the activation function on deep neural networks training. *ICML*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *ICCV*.
- Hu, W., Junchi Li, C., Li, L., and Liu, J. (2018). On the diffusion approximation of nonconvex stochastic gradient descent. *arXiv preprint arXiv:1705.07562*.
- Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. *32nd Conference on Neural Information Processing Systems*.
- Karakida, R., Akaho, S., and Amari, S. (2018). Universal statistics of Fisher information in deep neural networks: Mean field approach. *arXiv preprint arXiv:1806.01316*.
- Kubo, M., Banno, R., Manabe, H., and Minoji, M. (2019). Implicit regularization in over-parameterized neural networks. *arXiv preprint arXiv:1903.01997*.
- Lee, J., Bahri, Y., Novak, R., Schoenholz, S., Pennington, J., and Sohl-Dickstein, J. (2018). Deep neural networks as Gaussian processes. *6th International Conference on Learning Representations*.
- Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Sohl-Dickstein, J., and Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*.
- Lei, D., Sun, Z., Xiao, Y., and Wang, W. (2018). Implicit regularization of stochastic gradient descent in natural language processing: Observations and implications. *arXiv preprint arXiv:1811.00659*.
- Li, Q., Tai, C., and E, W. (2017). Stochastic modified equations and adaptive stochastic gradient algorithms. *arXiv preprint arXiv:1511.06251*.
- Nguyen, Q. and Hein, M. (2018). Optimization landscape and expressivity of deep CNNs. *ICML*.
- Schoenholz, S., Gilmer, J., Ganguli, S., and Sohl-Dickstein, J. (2017). Deep information propagation. *5th International Conference on Learning Representations*.
- Schoenholz, S., Lee, J., Novak, R., Xiao, L., Bahri, Y., and Sohl-Dickstein, J. (2019). Neural tangents.
- Yang, G. (2019). Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*.
- Yang, G. and Schoenholz, S. (2017). Mean field residual networks: On the edge of chaos. *Advances in Neural Information Processing Systems*, 30:2869–2869.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.
- Zou, D., Cao, Y., Zhou, D., and Gu, Q. (2018). Stochastic gradient descent optimizes over-parameterized deep ReLU networks. *arXiv preprint arXiv:1811.08888*.

## A Proofs of Section 2: Neural Networks and Neural Tangent Kernel

### A.1 Proofs of Subsection 2.1

**Lemma 1** (Discretization Error for Full-Batch Gradient Descent). *Assume  $\nabla_\theta \mathcal{L}$  is  $C$ -lipschitz, then there exists  $C' > 0$  that depends only on  $C$  and  $T$  such that*

$$\sup_{k \in [0, T/\eta]} \|\theta_{t_k} - \hat{\theta}_k\| \leq \eta C'$$

*Proof.* For  $t \in [0, T]$ , we define the stepwise constant system  $\tilde{\theta}_t = \hat{\theta}_{\lfloor t/\eta \rfloor}$ . Let  $t \in [0, T]$ , we have

$$\begin{aligned} \tilde{\theta}_t &= \theta_0 - \eta \sum_{k=0}^{\lfloor t/\eta \rfloor - 1} \nabla_\theta \mathcal{L}(\hat{\theta}_k) \\ &= \theta_0 - \int_0^t \nabla_\theta \mathcal{L}(\tilde{\theta}_s) ds + \eta \int_{\lfloor t/\eta \rfloor - 1}^{t/\eta} \nabla_\theta \mathcal{L}(\hat{\theta}_{\lfloor s \rfloor}) ds \end{aligned}$$

Therefore,

$$\begin{aligned} \|\theta_t - \tilde{\theta}_t\| &\leq \int_0^t \|\nabla_\theta \mathcal{L}(\theta_s) - \nabla_\theta \mathcal{L}(\tilde{\theta}_s)\| ds + \eta \int_{\lfloor t/\eta \rfloor - 1}^{t/\eta} \|\nabla_\theta \mathcal{L}(\hat{\theta}_{\lfloor s \rfloor})\| ds \\ &\leq C \int_0^t \|\theta_s - \tilde{\theta}_s\| ds + \eta(t/\eta - \lfloor t/\eta \rfloor) \|\nabla_\theta \mathcal{L}(\hat{\theta}_{\lfloor t/\eta \rfloor})\| + \eta \|\nabla_\theta \mathcal{L}(\hat{\theta}_{\lfloor t/\eta \rfloor - 1})\| \end{aligned}$$

Moreover, for any  $k \in [0, \lfloor T/\eta \rfloor]$ , we have

$$\begin{aligned} \|\hat{\theta}_k - \theta_0\| &\leq (1 + \eta C) \|\hat{\theta}_{k-1} - \theta_0\| \\ &\leq (1 + \eta C)^{T/\eta} \|\hat{\theta}_1 - \theta_0\| \\ &\leq e^{CT} \|\hat{\theta}_1 - \theta_0\| \end{aligned}$$

where we have used  $\log(1 + \eta C) \leq \eta C$ . Using this result, there exists a constant  $\tilde{C}$  depending on  $T$  and  $C$  such that

$$\begin{aligned} \eta(t/\eta - \lfloor t/\eta \rfloor) \|\nabla_\theta \mathcal{L}(\hat{\theta}_{\lfloor t/\eta \rfloor})\| + \eta \|\nabla_\theta \mathcal{L}(\hat{\theta}_{\lfloor t/\eta \rfloor - 1})\| &\leq \eta(2 \|\nabla_\theta \mathcal{L}(\hat{\theta}_0)\| + C \|\hat{\theta}_{\lfloor t/\eta \rfloor} - \theta_0\| + C \|\hat{\theta}_{\lfloor t/\eta \rfloor - 1} - \theta_0\|) \\ &\leq \eta \tilde{C} \end{aligned}$$

Now we have

$$\|\theta_t - \tilde{\theta}_t\| \leq C \int_0^t \|\theta_s - \tilde{\theta}_s\| ds + \eta \tilde{C},$$

so we can conclude using Gronwall's lemma.  $\square$

### A.2 Proofs of Subsection 2.2

Recall that

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta \nabla_\theta \mathcal{L}^{(S)}(\hat{\theta}_t) \quad (14)$$

where  $\mathcal{L}^{(S)} = \frac{1}{S} \sum_{i=1}^S \ell(f(\tilde{x}_i, \theta), \tilde{y}_i)$  where  $(\tilde{x}_i, \tilde{y}_i)_{1 \leq i \leq S}$  is a randomly selected batch of size  $S$ . Then for all  $\theta$

$$\nabla_\theta \mathcal{L}^{(S)}(\theta) - \nabla_\theta \mathcal{L}(\theta) = \sum_i \frac{Z_i(S)}{S} (\nabla_\theta \ell(f_\theta(x_i), y_i) - E_0(\theta)) - \sum_{i=1}^N \frac{(\nabla_\theta \ell(f_\theta(x_i), y_i) - E_0(\theta))}{N}$$

where  $Z_i(S) = 1$  if observation  $i$  belongs to the batch  $(\tilde{x}_j, \tilde{y}_j), j \leq S$  and equals 0 otherwise and  $E_0(\theta) = \mathbb{E}_0 \nabla_\theta \ell(f(X_1, \theta), Y_1)$ . We have

$$\text{tr} \left[ \text{Cov} \left( \sum_{i=1}^N \frac{(\nabla_\theta \ell(f_\theta(x_i), y_i) - E_0(\theta))}{N} \right) \right] = \sum_{l=1}^p \frac{\text{Var}(\partial \ell(f_\theta(X_1), Y_1) / \partial \theta_l)}{N}.$$

So that if  $S = o(N)$  and if

$$\text{tr}(\text{Cov}(\nabla_\theta \ell(f(X_1, \theta), Y_1))) = o(S)$$

where  $\text{Cov}(\cdot)$  denotes the covariance matrix under  $\mathbb{P}_0$ . Then

$$\nabla_{\theta} \mathcal{L}^{(S)}(\theta) - \nabla_{\theta} \mathcal{L}(\theta) = \frac{Z_S(\theta)}{\sqrt{S}} + o_{P_0}(S^{-1/2})$$

where  $Z_S(\theta)$  converges in distribution (as  $S$  goes to infinity) to a Gaussian random vector with covariance matrix  $\Sigma(\theta) = \text{Cov}(\nabla_{\theta} \ell(f(X_1, \theta), Y_1))$  and we have, neglecting the term  $o_{P_0}(S^{-1/2})$ ,

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta \nabla_{\theta} \mathcal{L}(\hat{\theta}_t) + \frac{\eta}{\sqrt{S}} Z(\theta_t). \quad (15)$$

We can in particular bound the difference between (15) and the continuous time SDE approximation

$$d\theta_t = -\nabla_{\theta} \mathcal{L}(\theta_t) dt + \sqrt{\frac{\eta}{S}} \Sigma(\theta_t)^{\frac{1}{2}} dW_t. \quad (16)$$

**Proposition 1.** *Under the SDE (16), the vector  $f_t(\mathcal{X})$  is the solution of the following SDE*

$$df_t(\mathcal{X}) = \left[ -\frac{1}{N} K_{\theta_t}(\mathcal{X}, \mathcal{X}) \nabla_z \ell(f_t(\mathcal{X}), Y) + \frac{1}{2} \frac{\eta}{S} \Gamma_t(\mathcal{X}) \right] dt + \sqrt{\frac{\eta}{S}} \nabla_{\theta} f(\mathcal{X}, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}} dW_t \quad (17)$$

where  $\Gamma_t(\mathcal{X})$  is the concatenated vector of  $(\Gamma_t(x) = (\text{Tr}(\Sigma(\theta_t)^{\frac{1}{2}} \nabla_2 f_i(x, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}}))_{1 \leq i \leq k})_{x \in \mathcal{X}}$  and  $\nabla_2 f_i(x, \theta)$  is the Hessian of  $f_i$  ( $i^{\text{th}}$  component of  $f$ ) with respect to  $\theta$ .

*Proof.* Since  $\theta_t$  is a diffusion process, we can use Itô's lemma to deduce how the randomness propagates to  $f_t$ . We denote by  $f_{t,i}$  the  $i^{\text{th}}$  coordinate of  $f_t$ , i.e., for an input  $x$ ,  $f_t(x) = (f_{t,i}(x))_{1 \leq i \leq k}$ . Let  $i \in 1, \dots, k$ , we have

$$\begin{aligned} df_{t,i}(x) &= \nabla_{\theta} f_i(x, \theta_t) d\theta_t + \frac{1}{2} \frac{\eta}{S} \text{Tr}(\Sigma(\theta_t)^{\frac{1}{2}} \nabla_2 f_i(x, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}}) dt \\ &= [-\nabla_{\theta} f_{t,i}(x) \nabla_{\theta} f_t(\mathcal{X}) \nabla_z \ell(f_t(\mathcal{X}), Y) + \frac{1}{2} \frac{\eta}{S} \text{Tr}(\Sigma(\theta_t)^{\frac{1}{2}} \nabla_2 f_i(x, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}})] dt \\ &\quad + \sqrt{\frac{\eta}{S}} \nabla_{\theta} f_i(x, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}} dW_t \end{aligned}$$

where  $\nabla_2 f_i(x, \theta_t)$  is the hessian of  $f_i$  with respect to  $\theta$ . Aggregating these equations with respect to  $i$  yields

$$df_t(x) = \left[ -\frac{1}{N} \nabla_{\theta} f_t(x) \nabla_{\theta} f_t(\mathcal{X}) \nabla_z \ell(f_t(\mathcal{X}), Y) + \frac{1}{2} \frac{\eta}{S} \Gamma_t(x) \right] dt + \sqrt{\frac{\eta}{S}} \nabla_{\theta} f(x, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}} dW_t$$

where  $\Gamma_t(x) = (\text{Tr}(\Sigma(\theta_t)^{\frac{1}{2}} \nabla_2 f_i(x, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}}))_{1 \leq i \leq k}$ .

Therefore, the dynamics of the vector  $f_t(\mathcal{X})$  is given by

$$df_t(\mathcal{X}) = \left[ -\frac{1}{N} K_{\theta_t}(\mathcal{X}, \mathcal{X}) \nabla_z \ell(f_t(\mathcal{X}), Y) + \frac{1}{2} \frac{\eta}{S} \Gamma_t(\mathcal{X}) \right] dt + \sqrt{\frac{\eta}{S}} \nabla_{\theta} f(\mathcal{X}, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}} dW_t$$

where  $\Gamma_t(\mathcal{X})$  is the concatenated vector of  $(\Gamma_t(x))_{x \in \mathcal{X}}$ .  $\square$

With the quadratic loss  $\ell(z, y) = \frac{1}{2} \|z - y\|^2$ , the SDE (17) is given by

$$df_t(\mathcal{X}) = \left[ -\frac{1}{N} K_{\theta_t}(\mathcal{X}, \mathcal{X}) (f_t(\mathcal{X}) - Y) + \frac{1}{2} \frac{\eta}{S} \Gamma_t(\mathcal{X}) \right] dt + \sqrt{\frac{\eta}{S}} \nabla_{\theta} f(\mathcal{X}, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}} dW_t. \quad (18)$$

This is an Ornstein-Uhlenbeck process (mean-reverting process) with time dependent parameters.

**Dynamics of  $f_t$  for wide feed-Forward neural networks under the quadratic loss:**

Using the approximation of the NTK by  $K^L$  as  $n_1, n_2, \dots, n_L \rightarrow \infty$ , the dynamics of  $f_t(\mathcal{X})$  for wide networks are given by

$$df_t(\mathcal{X}) = -\frac{1}{N} \hat{K}^L (f_t(\mathcal{X}) - M_t) dt + \sqrt{\frac{\eta}{S}} \nabla_{\theta} f(\mathcal{X}, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}} dW_t,$$

To solve it, we use the change of variable  $A_t = e^{\frac{t}{N} \hat{K}^L} f_t(\mathcal{X})$ . Using Ito's lemma, we have

$$\begin{aligned} dA_t &= \frac{1}{N} \hat{K}^L A_t dt + e^{\frac{t}{N} \hat{K}^L} df_t(\mathcal{X}) \\ &= \frac{1}{N} \hat{K}^L e^{\frac{t}{N} \hat{K}^L} M_t dt + \sqrt{\frac{\eta}{S}} e^{\frac{t}{N} \hat{K}^L} \nabla_{\theta} f(\mathcal{X}, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}} dW_t \end{aligned}$$

By integrating, we conclude that

$$f_t(\mathcal{X}) = e^{-\frac{t}{N}\hat{K}^L} f_0(\mathcal{X}) + \frac{1}{N} \int_0^t \hat{K}^L e^{-\frac{(t-s)}{N}\hat{K}^L} M_s ds + \sqrt{\frac{\eta}{S}} \int_0^t e^{-\frac{t-s}{N}\hat{K}^L} \nabla_\theta f(\mathcal{X}, \theta_s) \Sigma(\theta_s)^{\frac{1}{2}} dW_s$$

we conclude for  $f_t(\mathcal{X})$  using the fact that  $\frac{1}{N} \int_0^t \hat{K}^L e^{-\frac{(t-s)}{N}\hat{K}^L} M_s ds = (I - e^{-\frac{t}{N}\hat{K}^L})\mathcal{Y} - \frac{\eta}{2S} \int_0^t e^{-\frac{(t-s)}{N}\hat{K}^L} \Gamma_s ds$ .

Recall that for any input  $x \in \mathbb{R}^d$ ,

$$df_t(x) = [-\frac{1}{N} K^L(x, \mathcal{X})(f_t(\mathcal{X}) - Y) + \frac{1}{2} \frac{\eta}{S} \Gamma_t(x)] dt + \sqrt{\frac{\eta}{S}} \nabla_\theta f(x, \theta_t) \Sigma(\theta_t)^{\frac{1}{2}} dW_t$$

To prove the expression of  $f_t(x)$  for general  $x \in \mathbb{R}^d$ , we substitute  $f_t(\mathcal{X})$  by its value in the SDE of  $f_t(x)$  and integrate.

## B Proofs of Section 3: Impact of the Initialization and the Activation function on the Neural Tangent Kernel

We first recall the results obtained in Lee et al. (2018), Schoenholz et al. (2017) and Hayou et al. (2019) where the impact of the EOC (Edge of Chaos) on the initialization is studied. We also present some results that we will be used below.

Consider a FFNN of depth  $L$ , widths  $(n_l)_{1 \leq l \leq L}$ , weights  $w^l$  and bias  $b^l$ . For some input  $x \in \mathbb{R}^d$ , the forward propagation is given by

$$y_i^1(x) = \sum_{j=1}^d w_{ij}^1 x_j + b_i^1, \quad y_i^l(x) = \sum_{j=1}^{n_{l-1}} w_{ij}^l \phi(y_j^{l-1}(x)) + b_i^l, \quad \text{for } l \geq 2, \quad (19)$$

where  $\phi$  is the activation function.

We initialize the model with  $w_{ij}^l \stackrel{iid}{\sim} \mathcal{N}(0, \frac{\sigma_w^2}{n_{l-1}})$  and  $b_i^l \stackrel{iid}{\sim} \mathcal{N}(0, \sigma_b^2)$ , where  $\mathcal{N}(\mu, \sigma^2)$  denotes the normal distribution of mean  $\mu$  and variance  $\sigma^2$ . For some  $x$ , we denote by  $q^l(x)$  the variance of  $y^l(x)$ . In general,  $q^l(x)$  converges to a point  $q(\sigma_b, \sigma_w) > 0$  independent of  $x$  as  $l \rightarrow \infty$ . The EOC is defined by the set of parameters  $(\sigma_b, \sigma_w)$  such that  $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)} Z)^2] = 1$  where  $Z \sim \mathcal{N}(0, 1)$ . Similarly the Ordered, resp. Chaotic, phase is defined by  $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)} Z)^2] < 1$ , resp.  $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)} Z)^2] > 1$  (see Hayou et al. (2019) for more details). For two inputs  $x, x' \in \mathbb{R}^d$ , define  $\Sigma^l(x, x') = \mathbb{E}[y^l(x) y^l(x')]$  and let  $c^l(x, x')$  be the corresponding correlation. Let  $f$  be the correlation function defined implicitly by  $c^{l+1} = f(c^l)$ . In the limit of infinitely wide networks, we have the following results (Hayou et al. (2019)):

- $\Sigma^l(x, x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}_{z \sim \mathcal{N}(0, \Sigma^{l-1})} [\phi(z(x)) \phi(z(x'))]$ .
- There exist  $q, \lambda > 0$  such that, for all  $\sup_{x \in \mathbb{R}^d} |\Sigma^l(x, x) - q| \leq e^{-\lambda l}$ .
- On the Ordered phase, there exists  $\gamma > 0$  such that  $\sup_{x, x' \in \mathbb{R}^d} |c^l(x, x') - 1| \leq e^{-\gamma l}$ .
- On the chaotic phase, there exist  $\gamma > 0$  and  $c < 1$  such that  $\sup_{x \neq x' \in \mathbb{R}^d} |c^l(x, x') - c| \leq e^{-\gamma l}$ .
- For ReLU network on the EOC, we have that  $\Sigma^l(x, x) = \frac{\sigma_w^2}{d} \|x\|^2$  for all  $l \geq 1$ . Moreover, we have

$$f(x) \underset{x \rightarrow 1-}{=} x + \frac{2\sqrt{2}}{3\pi} (1-x)^{3/2} + O((1-x)^{5/2})$$

- In general, we have

$$f(x) = \frac{\sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(\sqrt{q} Z_1) \phi(\sqrt{q} Z(x))]}{q}$$

where  $Z(x) = x Z_1 + \sqrt{1-x^2} Z_2$  and  $Z_1, Z_2$  are iid standard Gaussian variables.

- On the EOC, we have  $f'(1) = 1$
- If  $\phi$  is  $k$ -times differentiable, then  $f$  is  $k$ -times differentiable and for all  $1 \leq j \leq k$ , we have  $f^{(j)}(x) = \sigma_w^2 q^{j-1} \mathbb{E}[\phi^{(j)}(Z_1) \phi^{(j)}(Z(x))]$
- From Jacot et al. (2018), we have that

$$K^l(x, x') = K^{l-1}(x, x') \dot{\Sigma}^l(x, x') + \Sigma^l(x, x').$$

where the definition of  $\dot{\Sigma}^l(x, x')$  is given in Proposition 2 below.

**Definition 1.** Let  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  be a measurable function. Then

1.  $\phi$  is said to be ReLU-like if there exist  $\lambda, \beta \in \mathbb{R}$  such that  $\phi(x) = \lambda x$  for  $x > 0$  and  $\phi(x) = \beta x$  for  $x \leq 0$ .
2.  $\phi$  is said to be in  $\mathcal{S}$  if  $\phi(0) = 0$ ,  $\phi$  is twice differentiable, and there exist  $n \geq 1$ , a partition  $(A_i)_{1 \leq i \leq n}$  of  $\mathbb{R}$  and infinitely differentiable functions  $g_1, g_2, \dots, g_n$  such that  $\phi^{(2)} = \sum_{i=1}^n 1_{A_i} g_i$ , where  $\phi^{(2)}$  is the second derivative of  $\phi$ .

The following two lemmas will be useful to prove the results of Section 3 in the main paper.

**Lemma 2.** Let  $(a_l)$  be a sequence of non-negative real numbers such that  $\forall l \geq 0, a_{l+1} \leq \alpha a_l + k e^{-\beta l}$ , where  $\alpha \in (0, 1)$  and  $k, \beta > 0$ . Then there exists  $\gamma > 0$  such that  $\forall l \geq 0, a_l \leq e^{-\gamma l}$ .

*Proof.* Using the inequality on  $a_l$ , we can easily see that

$$\begin{aligned} a_l &\leq a_0 \alpha^l + k \sum_{j=0}^{l-1} \alpha^j e^{-\beta(l-j)} \\ &\leq a_0 \alpha^l + k \frac{l}{2} e^{-\beta l/2} + k \frac{l}{2} \alpha^{l/2} \end{aligned}$$

where we divided the sum into two parts separated by index  $l/2$  and upper-bounded each part. The existence of  $\gamma$  is straightforward.  $\square$

**Proposition 2** (Limiting Neural Tangent Kernel with Ordered/Chaotic Initialization). Let  $(\sigma_b, \sigma_w)$  be in the ordered or chaotic phase. Then, there exist  $\lambda, \gamma > 0$  such that

$$\sup_{x, x' \in \mathbb{R}^d} |K^L(x, x') - \lambda| \leq e^{-\gamma L} \rightarrow_{L \rightarrow \infty} 0$$

*Proof.* From Jacot et al. (2018), we have that

$$K^l(x, x') = K^{l-1}(x, x') \dot{\Sigma}^l(x, x') + \Sigma^l(x, x')$$

where  $\Sigma^1(x, x') = \sigma_b^2 + \frac{\sigma_w^2}{d} x^T x'$  and  $\Sigma^l(x, x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}_{f \sim \mathcal{N}(0, \Sigma^{l-1})} [\phi(f(x)) \phi(f(x'))]$  and  $\dot{\Sigma}^l(x, x') = \mathbb{E}_{f \sim \mathcal{N}(0, \Sigma^{l-1})} [\phi'(f(x)) \phi'(f(x'))]$ . In the ordered/chaotic phase, Hayou et al. (2019) showed that there exist  $k, \gamma, l_0 > 0$  and  $\alpha \in (0, 1)$  such that for all  $l \geq l_0$  we have

$$\sup_{x, x' \in \mathbb{R}^d} |\Sigma^l(x, x') - k| \leq e^{-\gamma l}$$

and

$$\sup_{x, x' \in \mathbb{R}^d} \dot{\Sigma}^l(x, x') \leq \alpha.$$

Therefore we have for any  $l \geq l_0$  and  $x, x' \in \mathbb{R}^d$

$$K^l(x, x') \leq \alpha K^{l-1}(x, x') + k + e^{-\gamma l}.$$

Letting  $r_l = K^l(x, x') - \frac{k}{1-\alpha}$ , we have

$$r_l \leq \alpha r_{l-1}.$$

We can now conclude using Lemma 2.  $\square$

Now, we show that the Initialization on the EOC leads to an invertible NTKl even if the number of layers  $L$  goes to infinity. We first prove two preliminary lemmas that will be useful for the proof of the next proposition.

**Lemma 3.** Let  $(a_l), (b_l), (\lambda_l)$  be three sequences of real numbers such that

$$\begin{aligned} a_l &= a_{l-1} \lambda_l + b_l \\ \lambda_l &= 1 - \frac{\alpha}{l} + O(l^{-1-\beta}) \\ b_l &= q + O(l^{-1}) \end{aligned}$$

where  $\alpha \in \mathbb{N}^*, \beta, q \in \mathbb{R}^+$  and  $\alpha > \beta - 1$ .

Then,

$$\frac{a_l}{l} = \frac{q}{1+\alpha} + O(l^{-\min(1, \beta)})$$

*Proof.* It is easy to see that  $|a_l| \leq l + |a_0|$  for all  $l \geq 0$ , therefore  $(a_l/l)$  is bounded. Now let  $\zeta = \min(1, \beta)$  and  $r_l = \frac{a_l}{l}$ . We have

$$\begin{aligned} r_l &= r_{l-1} \left(1 - \frac{1}{l}\right) \left(1 - \frac{\alpha}{l} + O(l^{-1-\beta})\right) + \frac{q}{l} + O(l^{-2}) \\ &= r_{l-1} \left(1 - \frac{1+\alpha}{l}\right) + \frac{q}{l} + O(l^{-1-\zeta}). \end{aligned}$$

Letting  $x_l = \left|r_l - \frac{q}{1+\alpha}\right|$ , there is exist a constant  $M > 0$  such that

$$x_l \leq x_{l-1} \left(1 - \frac{1+\alpha}{l}\right) + \frac{M}{l^{1+\zeta}}.$$

Hence, we have

$$x_l \leq x_0 \prod_{k=1}^l \left(1 - \frac{1+\alpha}{k}\right) + M \sum_{k=1}^l \prod_{j=k+1}^l \left(1 - \frac{1+\alpha}{j}\right) \frac{1}{k^{1+\zeta}}.$$

By taking the logarithm of the first term in the right hand side and using the fact that  $\sum_{k=1}^l \frac{1}{k} \sim \log(l)$ , we have

$$\prod_{k=1}^l \left(1 - \frac{1+\alpha}{k}\right) \sim l^{-1-\alpha}$$

For the second part, observe that

$$\prod_{j=k+1}^l \left(1 - \frac{1+\alpha}{j}\right) = \frac{(l-\alpha-1)!}{l!} \frac{k!}{(k-\alpha-1)!}$$

and

$$\frac{k!}{(k-\alpha-1)!} \frac{1}{k^{1+\zeta}} \sim_{k \rightarrow \infty} k^{\alpha-\zeta}$$

so that,

$$\begin{aligned} \sum_{k=1}^l \frac{k!}{(k-\alpha-1)!} \frac{1}{k^{1+\zeta}} &\sim \sum_{k=1}^l k^{\alpha-\zeta} \\ &\sim \int_1^l t^{\alpha-\zeta} dt \\ &\sim \frac{1}{\alpha-\zeta+1} l^{\alpha-\zeta+1} \end{aligned}$$

therefore,

$$\begin{aligned} \sum_{k=1}^l \prod_{j=k+1}^l \left(1 - \frac{1+\alpha}{j}\right) \frac{1}{k^{1+\zeta}} &= \frac{(l-\alpha-1)!}{l!} \sum_{k=1}^l \frac{k!}{(k-\alpha-1)!} \frac{1}{k^{1+\zeta}} \\ &\sim \frac{1}{\alpha-\zeta+1} l^{-\zeta} \end{aligned}$$

We can now conclude using the fact that  $\alpha > \beta - 1$ .

□

We now introduce a different form of the previous Lemma that will be useful for other applications.

**Lemma 4.** Let  $(a_l), (b_l), (\lambda_l)$  be three sequences of real numbers such that

$$\begin{aligned} a_l &= a_{l-1} \lambda_l + b_l \\ \lambda_l &= 1 - \frac{\alpha}{l} + \kappa \frac{\log(l)}{l^2} + O(l^{-1-\beta}) \\ b_l &= q + O(l^{-1}) \end{aligned}$$

where  $\alpha \in \mathbb{N}^*, \beta, q \in \mathbb{R}^+$  and  $\alpha > \beta - 1, \beta \geq 1$ .

Then, there exists  $A, B > 0$  such that

$$A \frac{\log(l)}{l} \leq \left| \frac{a_l}{l} - \frac{q}{1+\alpha} \right| \leq B \frac{\log(l)}{l}$$

*Proof.* It is easy to see that  $|a_l| \leq l + |a_0|$  for all  $l \geq 0$ , therefore  $(a_l/l)$  is bounded. Let  $r_l = \frac{a_l}{l}$ . We have

$$\begin{aligned} r_l &= r_{l-1} \left(1 - \frac{1}{l}\right) \left(1 - \frac{\alpha}{l} + \kappa \frac{\log(l)}{l^2} + O(l^{-1-\beta})\right) + \frac{q}{l} + O(l^{-2}) \\ &= r_{l-1} \left(1 - \frac{1+\alpha}{l}\right) + r_{l-1} \kappa \frac{\log(l)}{l^2} + \frac{q}{l} + O(l^{-2}) \end{aligned}$$

Let  $x_l = r_l - \frac{q}{1+\alpha}$ . It is clear that  $\lambda_l = 1 - \alpha/l + O(l^{-3/2})$ . Therefore, using Lemma 3 with  $\beta = 1/2$ , we have  $r_l \rightarrow \frac{q}{1+\alpha}$ . Thus, there exists  $\kappa_1, \kappa_2, M, l_0 > 0$  such that for all  $l \geq l_0$

$$x_{l-1} \left(1 - \frac{1+\alpha}{l}\right) + \kappa_1 \frac{\log(l)}{l^2} - \frac{M}{l^2} \leq x_l \leq x_{l-1} \left(1 - \frac{1+\alpha}{l}\right) + \kappa_2 \frac{\log(l)}{l^2} + \frac{M}{l^2}$$

Similarly to the proof of Lemma 3, it follows that

$$x_l \leq x_{l_0} \prod_{k=l_0}^l \left(1 - \frac{1+\alpha}{k}\right) + \sum_{k=l_0}^l \prod_{j=k+1}^l \left(1 - \frac{1+\alpha}{j}\right) \frac{\kappa_2 \log(k) + M}{k^2}$$

and

$$x_l \geq x_0 \prod_{k=0}^l \left(1 - \frac{1+\alpha}{k}\right) + \sum_{k=l_0}^l \prod_{j=k+1}^l \left(1 - \frac{1+\alpha}{j}\right) \frac{\kappa_1 \log(k) - M}{k^2}$$

Recall that we have

$$\prod_{k=1}^l \left(1 - \frac{1+\alpha}{k}\right) \sim l^{-1-\alpha}$$

and

$$\prod_{j=k+1}^l \left(1 - \frac{1+\alpha}{j}\right) = \frac{(l-\alpha-1)!}{l!} \frac{k!}{(k-\alpha-1)!}$$

so that

$$\frac{k!}{(k-\alpha-1)!} \frac{\kappa_1 \log(k) - M}{k^2} \sim_{k \rightarrow \infty} \log(k) k^{\alpha-1}$$

Therefore, we obtain

$$\begin{aligned} \sum_{k=1}^l \frac{k!}{(k-\alpha-1)!} \frac{\kappa_1 \log(k) - M}{k^2} &\sim \sum_{k=1}^l \log(k) k^{\alpha-1} \\ &\sim \int_1^l \log(t) t^{\alpha-1} dt \\ &\sim C_1 l^\alpha \log(\alpha) \end{aligned}$$

where  $C_1 > 0$  is a constant. Similarly, there exists a constant  $C_2 > 0$  such that

$$\sum_{k=1}^l \frac{k!}{(k-\alpha-1)!} \frac{\kappa_2 \log(k) + M}{k^2} \sim C_2 l^\alpha \log(\alpha)$$

We conclude using the fact that  $\frac{(l-\alpha-1)!}{l!} \sim l^{-1-\alpha}$ .

□

**Proposition 3** (Neural Tangent Kernel on the Edge of Chaos). *Let  $\phi$  be a non-linear activation function and  $(\sigma_b, \sigma_w) \in \text{EOC}$ .*

1. *If  $\phi$  is ReLU-like, then for all  $x \in \mathbb{R}^d$ ,  $\frac{K^L(x, x)}{L} = \frac{\sigma_w^2 \|x\|^2}{d} + \frac{K^0(x, x)}{L}$ . Moreover, there exist  $A, \lambda \in (0, 1)$  such that*

$$\sup_{x \neq x' \in \mathbb{R}^d} \left| \frac{K^L(x, x')}{L} - \lambda \frac{\sigma_w^2}{d} \|x\| \|x'\| \right| \leq \frac{A}{L}$$

2. *If  $\phi$  is in  $\mathcal{S}$ , then, there exist  $q > 0$  such that  $\frac{K^L(x, x)}{L} = q + \frac{K^0(x, x)}{L} \rightarrow q$ . Moreover, there exist  $B, C, \lambda \in (0, 1)$  such that*

$$\frac{B \log(L)}{L} \leq \sup_{x \neq x' \in \mathbb{R}^d} \left| \frac{K^L(x, x')}{L} - q \lambda \right| \leq \frac{C \log(L)}{L}$$

*Proof.* We use some results from Hayou et al. (2019) in this proof.

Let  $x, x' \in \mathbb{R}^d$  and  $c_{x,x'}^l = \frac{\Sigma^l(x,x')}{\sqrt{\Sigma^l(x,x)\Sigma^l(x',x')}}.$  Let  $\gamma_l := 1 - c_{x,x'}^l$  and  $f$  be the correlation function defined by the recursive equation  $c^{l+1} = f(c^l)$ . From the preliminary results, we know that  $\Sigma^l(x, x) = \frac{\sigma_w^2}{d} \|x\|^2$  and that  $K^l(x, x') = K^{l-1}(x, x')\dot{\Sigma}^l(x, x') + \Sigma^l(x, x')$ . This concludes the proof for  $K^L(x, x)$ . We denote  $s = \frac{2\sqrt{2}}{3\pi}$ . From Hayou et al. (2019), we have on the EOC  $\gamma_{l+1} = \gamma_l - s\gamma_l^{3/2} + O(\gamma_l^{5/2})$  so that

$$\begin{aligned}\gamma_{l+1}^{-1/2} &= \gamma_l^{-1/2}(1 - s\gamma_l^{1/2} + O(\gamma_l^{3/2}))^{-1/2} = \gamma_l^{-1/2}(1 + \frac{s}{2}\gamma_l^{1/2} + O(\gamma_l^{3/2})) \\ &= \gamma_l^{-1/2} + \frac{s}{2} + O(\gamma_l).\end{aligned}$$

Thus, as  $l$  goes to infinity

$$\gamma_{l+1}^{-1/2} - \gamma_l^{-1/2} \sim \frac{s}{2}$$

and by summing and equivalence of positive divergent series

$$\gamma_l^{-1/2} \sim \frac{s}{2}l.$$

Moreover, since  $\gamma_{l+1}^{-1/2} - \gamma_l^{-1/2} = \frac{s}{2} + O(\gamma_l) = \frac{s}{2} + O(l^{-2})$ , we have  $\gamma_l^{-1/2} = \frac{s}{2}l + O(1)$ . Therefore,  $c_{x,x'}^l = 1 - \frac{9\pi^2}{2l^2} + O(l^{-3})$ . we also have

$$\begin{aligned}f'(x) &= \frac{1}{\pi} \arcsin(x) + \frac{1}{2} \\ &= 1 - \frac{\sqrt{2}}{\pi}(1-x)^{1/2} + O((1-x)^{5/2}).\end{aligned}$$

Thus, it follows that

$$f'(c_{x,x'}^l) = 1 - \frac{3}{l} + O(l^{-2})$$

Moreover,  $q_{x,x'}^l = q + O(l^{-2})$  where  $q$  is the limiting variance of  $y^l$ .

Using Lemma 3, we conclude that  $\frac{K^l(x,x')}{l} = \frac{1}{4} \frac{\sigma_w^2}{d} \|x\| \|x'\| + O(l^{-1})$ . Since  $c^{x,x'}$  is bounded, this result is uniform in  $x, x' \in \mathbb{R}^d$ .

2. We prove the result when  $\phi^{(2)}(x) = 1_{x < 0}g_1(x) + 1_{x \geq 0}g_2(x)$ . The generalization to the whole class is straightforward. Let  $f$  be the correlation function. We first show that for all  $k \geq 3$   $f^{(k)}(x) = \frac{1}{(1-x^2)^{(k-2)/2}}g_k(x)$  where  $g_k \in \mathcal{C}^\infty$ .

We have

$$\begin{aligned}f''(x) &= \sigma_w^2 q \mathbb{E}[\phi''(\sqrt{q}Z_1)\phi''(\sqrt{q}U_2(x))] \\ &= \sigma_w^2 q \mathbb{E}[\phi''(\sqrt{q}Z_1)1_{U_2(x) < 0}g_1(\sqrt{q}U_2(x))] + \sigma_w^2 q \mathbb{E}[\phi''(\sqrt{q}Z_1)1_{U_2(x) > 0}g_2(\sqrt{q}U_2(x))].\end{aligned}$$

Let  $G(x) = \mathbb{E}[\phi''(\sqrt{q}Z_1)1_{U_2(x) < 0}g_1(\sqrt{q}U_2(x))]$  then

$$\begin{aligned}G'(x) &= \mathbb{E}[\phi''(\sqrt{q}Z_1)(Z_1 - \frac{x}{\sqrt{1-x^2}}Z_2)\delta_{U_2(x)=0}\frac{1}{\sqrt{1-x^2}}g_1(\sqrt{q}U_2(x))] \\ &\quad + \mathbb{E}[\phi''(\sqrt{q}Z_1)1_{U_2(x) < 0}\sqrt{q}(Z_1 - \frac{x}{\sqrt{1-x^2}}Z_2)g_1'(\sqrt{q}U_2(x))].\end{aligned}$$

It is easy to see that  $G'(x) = \frac{1}{\sqrt{1-x^2}}G_1(x)$  where  $G_1 \in \mathcal{C}^1$ . A similar analysis can be applied to the second term of  $f''$ . We conclude that there exists  $g_3 \in \mathcal{C}^\infty$  such that  $f^{(3)}(x) = \frac{1}{\sqrt{1-x^2}}g(x)$ . We obtain the result by induction.

Since  $f^{(k)}$  are potentially not defined at 1, we use the change of variable  $x = 1 - t^2$  to obtain a Taylor expansion near 1. Simple algebra shows that the function  $t \rightarrow f(1 - t^2)$  has a Taylor expansion near 0:

$$f(1 - t^2) = 1 - t^2 f'(1) + \frac{t^4}{2} f''(1) + \frac{t^6}{6} f^{(3)}(1) + O(t^8).$$



Therefore,

$$f(x) = 1 + (x-1)f'(1) + \frac{(x-1)^2}{2}f''(1) + \frac{(1-x)^3}{6}f^{(3)}(1) + O((x-1)^4).$$

Letting  $\lambda_l := 1 - c^l$ , there exist  $\alpha, \beta > 0$  such that

$$\lambda_{l+1} = \lambda_l - \alpha\lambda_l^2 - \beta\lambda_l^3 + O(\lambda_l^4)$$

therefore,

$$\begin{aligned}\lambda_{l+1}^{-1} &= \lambda_l^{-1}(1 - \alpha\lambda_l - \beta\lambda_l^2 + O(\lambda_l^3))^{-1} \\ &= \lambda_l^{-1}(1 + \alpha\lambda_l + \beta\lambda_l^2 + O(\lambda_l^3)) \\ &= \lambda_l^{-1} + \alpha + \beta\lambda_l + O(\lambda_l^2).\end{aligned}$$

By summing (divergent series), we have that  $\lambda_l^{-1} \sim \frac{l}{\beta_q}$ . Therefore,

$$\lambda_{l+1}^{-1} - \lambda_l^{-1} - \alpha = \beta\alpha^{-1}l^{-1} + O(l^{-2})$$

By summing a second time, we obtain

$$\lambda_l^{-1} = \alpha l + \beta\alpha^{-1} \log(l) + O(1)$$

so that  $\lambda_l = \alpha^{-1}l^{-1} - \alpha^{-1}\beta \frac{\log(l)}{l^2} + O(l^{-2})$ .

Using the fact that  $f'(x) = 1 + (x-1)f''(1) + O((x-1)^2)$ , we have  $f'(c_{x,x'}^l) = 1 - \frac{2}{l} + \kappa \frac{\log(l)}{l^2} + O(l^{-2})$ . We can now conclude using Lemma 4. Using again the argument of the boundedness of  $c_{x,x'}^1$ , we can take the supremum.

□

**Proposition 4.** *Consider the following network architecture (FFNN with residual connections)*

$$y_i^l(x) = y_i^{l-1}(x) + \sum_{j=1}^{n_l-1} w_{ij}^l \phi(y_j^{l-1}(x)) + b_i^l, \quad \text{for } l \geq 2. \quad (20)$$

with initialization parameters  $\sigma_b = 0$  and  $\sigma_w > 0$ . Let  $K_{res}^L$  be the corresponding NTK. For all  $x \in \mathbb{R}^d$ ,  $\frac{K_{res}^L(x,x)}{L \times 2^L} = \sigma_w^2 \frac{\|x\|^2}{d} + O(L^{-1})$  and there exists  $\lambda \in (0, 1)$  such that

$$\sup_{x \neq x' \in \mathbb{R}^d} \left| \frac{K_{res}^L(x, x')}{L \times 2^L} - \sigma_w^2 \frac{\|x\| \times \|x'\|}{d} \lambda \right| = O(L^{-1})$$

*Proof.* We only give a sketch of the proof. A more rigorous proof can be easily done but it unnecessary. As in the feedforward without residual connections case, it is easy to see that  $K_{res}^L$  satisfies the following recursive equation

$$K_{res}^l(x, x') = K_{res}^{l-1}(x, x')(\dot{\Sigma}^l(x, x') + 1) + \Sigma^l(x, x')$$

we already now that  $\dot{\Sigma}^l(x, x) = 1$ . Moreover, we have  $\Sigma^l(x, x) = \Sigma^{l-1}(x, x) + \sigma_w^2/2 \Sigma^{l-1}(x, x) = (1 + \sigma_w^2/2)^{l-1} \frac{\sigma_w^2}{d} \|d\|$ . Depending on the value of  $\sigma_w$ , the behaviour of  $K^l(x, x')$  changes. However, from Hayou et al. (2019), we have that  $\dot{\Sigma}^l(x, x') = 1 - \beta l^{-1} + O(l^{-2})$ . So by scaling with  $\alpha_L 2^L$  and using Lemma 3, we conclude on the convergence rate of  $O(l^{-1})$ . We can take the supremum as the result of the boundedness of  $c^1(x, x')$ .

□

## C Further theoretical results: Impact of the Initialization the output function

In this section, we show how an initialization on the EOC impacts on the output function of the neural network. More precisely, we show that it leads to a larger range compared to an initialization in the Ordered/chaotic phase.

Let us start by a simple Lemma that compares the expectations of some smooth mapping with respect to two different Gaussian vectors.

**Lemma 5.** Let  $X = (X_i)_{1 \leq i \leq n}$ ,  $Y = (Y_i)_{1 \leq i \leq N}$  be two centered Gaussian vectors in  $\mathbb{R}^n$ . Let  $g \in \mathcal{D}^2(\mathbb{R}^n, \mathbb{R})$ . Then we have

$$\mathbb{E}[g(X)] - \mathbb{E}[g(Y)] = \frac{1}{2} \int_0^1 \sum_{1 \leq i, j \leq n} (\mathbb{E}[X_i X_j] - \mathbb{E}[Y_i Y_j]) \mathbb{E}\left[\frac{\partial g}{\partial x_i \partial x_j}(\sqrt{1-u}X + \sqrt{u}Y)\right] du \quad (21)$$

The result of Lemma 5 is valid when the second derivatives of  $g$  exist only in the distribution sense (e.g. Dirac mass).

*Proof.* We define the function  $G$  on  $\mathbb{R}$  by

$$G(t) = \mathbb{E}[g(tX + \sqrt{1-t^2}Y)]$$

we have that

$$G'(t) = \sum_{i=1}^n \mathbb{E}\left[\left(X_i - \frac{t}{\sqrt{1-t^2}}Y_i\right) \frac{\partial g}{\partial x_i}(tX + \sqrt{1-t^2}Y)\right]$$

Moreover, it is easy to see that for any random vector  $Z$  in  $\mathbb{R}^n$  we have  $\mathbb{E}[Z_i g(Z)] = \sum_{j=1}^n \text{cov}(X_i, X_j) \mathbb{E}\left[\frac{\partial g}{\partial x_j}(Z)\right]$ , this yields

$$G'(t) = t \sum_{i,j=1}^n (\mathbb{E}[X_i X_j] - \mathbb{E}[Y_i Y_j]) \mathbb{E}\left[\frac{\partial g}{\partial x_i \partial x_j}(tX + \sqrt{1-t^2}Y)\right]$$

We conclude by integrating  $G'(t)$  between 0 and 1.  $\square$

Now let  $\mathcal{D} = \{(x_i, z_i) : 1 \leq i \leq N\}$  be the datapoints. Using the same notations as in the previous chapter, let  $y^L(x_i)$  denotes on the neurons of  $l^{th}$  layer (the neurons are iid). Assume  $c_{x_i, x_j}^1 \geq 0$  for all  $i, j \in [1, N]$  (this is almost always the case, but in general, we can re-scale the input data to satisfy this assumption). We have the following result

**Lemma 6.** Let  $\phi$  be a non ReLU-like activation function and  $(\sigma_b, \sigma_w) \notin EOC$ . Then there exists  $(\sigma_{b,EOC}, \sigma_{w,EOC}) \in EOC$  such that for any function  $g \in \mathcal{D}^2$  such that for all  $i, j \in [1, N]$ ,  $\frac{\partial g}{\partial x_i \partial x_j} \geq 0$ , there exist  $\beta > 0, \zeta_N > 0$  such that

$$\mathbb{E}[g(y_{eoc}^L(X))] \leq \mathbb{E}[g(y_{ord}^L(X))] - \frac{\zeta_N}{L^\beta}$$

*Proof.* Let  $(\sigma_b, \sigma_w) \notin EOC$  and  $q$  be the corresponding limiting variance. From the previous chapter, it is easy to see that there exists  $\sigma_0 > 0$  such that  $\sigma_0^2 + \frac{\mathbb{E}[\phi(\sqrt{q}Z)^2]}{\mathbb{E}[\phi'(\sqrt{q}Z)^2]} = q$ . Let  $(\sigma_{b,EOC}, \sigma_{w,EOC}) = (\sigma_0, 1/\sqrt{\mathbb{E}[\phi'(\sqrt{q}Z)^2]}) \in EOC$ . There exists a constant  $\kappa > 0$  (independent of  $N$ ) such that for all  $i, j \in [1, N]$ ,  $\mathbb{E}[y_{eoc}^L(X_i) y_{eoc}^L(X_j)] \leq \mathbb{E}[y_{ord}^L(X_i) y_{ord}^L(X_j)] - \kappa L^{-\beta}$  where  $\beta = 1$  for a smooth activation functions in  $\mathcal{S}$  and  $\beta = 2$  for ReLU-like activation functions (see Hayou et al. (2019)). Let  $\lambda_N = \inf_{i,j} \inf_{u \in [0,1]} \mathbb{E}\left[\frac{\partial g}{\partial x_i \partial x_j}(\sqrt{1-u}X + \sqrt{u}Y)\right]$ . Using Lemma 5, we have

$$\mathbb{E}[y_{eoc}^L(X_i) y_{eoc}^L(X_j)] \leq \mathbb{E}[y_{ord}^L(X_i) y_{ord}^L(X_j)] - \frac{1}{2} \kappa N^2 \lambda_N L^{-\beta}$$

$\square$

As a simple application, using the function  $g(X) = \prod_{i=1}^N 1_{x_i \leq t_i}$ , we have the following

**Lemma 2.** Let  $t_1, t_2, \dots, t_N \in \mathbb{R}$ . Then there exist  $\beta > 0, \zeta_N > 0$  such that

$$\mathbb{P}(y_{eoc}^L(x_1) \leq t_1, \dots, y_{eoc}^L(x_N) \leq t_N) \leq \mathbb{P}(y_{ord}^L(x_1) \leq t_1, \dots, y_{ord}^L(x_N) \leq t_N) - \frac{\zeta_N}{L^\beta}$$

as a Corollary, we have a generalized form of Slepian's Lemma for  $y_{eoc}^L(X)$  and  $y_{ord}^L(X)$ .

**Corollary 1** (Max Range and Min Range).

$$\mathbb{P}(\max_i y_{eoc}^L(x_i) \geq t) \geq \mathbb{P}(\max_i y_{ord}^L(x_i) \geq t) + \frac{\zeta_N}{L^\beta}$$

and

$$\mathbb{P}(\min_i y_{eoc}^L(x_i) \leq t) \geq \mathbb{P}(\min_i y_{ord}^L(x_i) \leq t) + \frac{\zeta_N}{L^\beta}$$

where  $\beta = 1$  for activation functions of type  $\mathcal{S}$  and  $\beta = 2$  for ReLU-like activation functions.

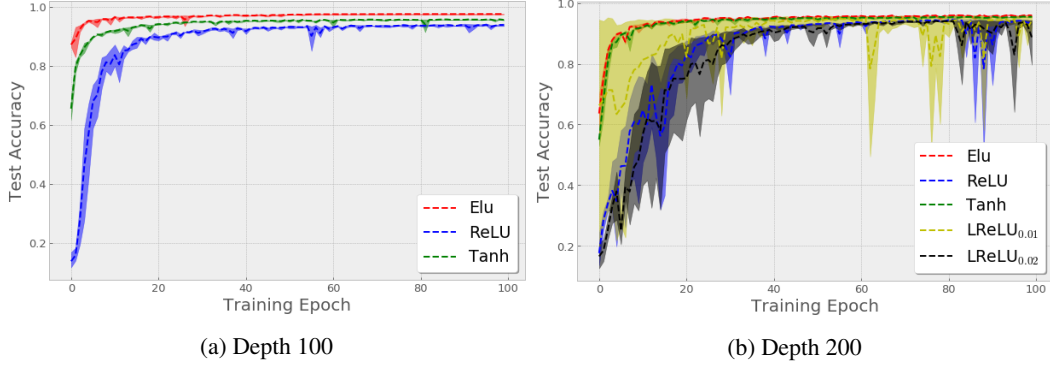


Figure 3: Test accuracy on MNIST as a function of the training epoch for different depths

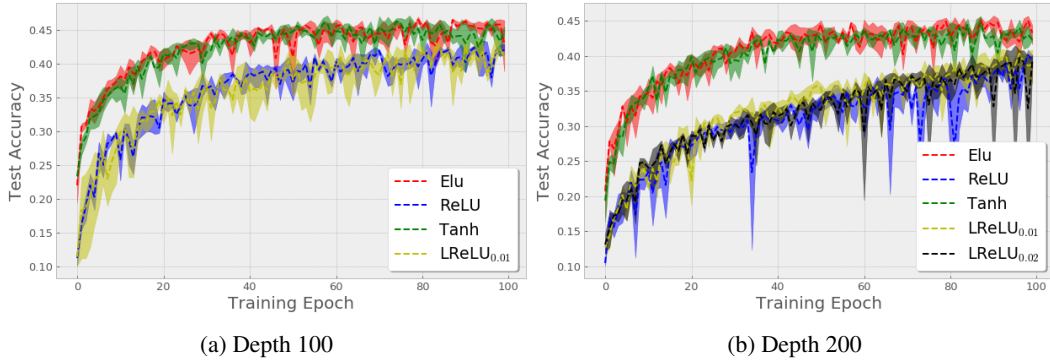


Figure 4: Test accuracy on CIFAR10 as a function of the training epoch for different depths

## D Further empirical results

Figures 3 and 4 show test accuracy for different activation functions for depths 100 and 200. We observe a clear advantage for activation functions of type  $\mathcal{S}$  (Elu and Tanh). This proves the consistency of the results across different depths.