Mean-field Behaviour of Neural Tangent Kernel for Deep Neural Networks

Soufiane Hayou, Arnaud Doucet, Judith Rousseau
Department of Statistics, University of Oxford

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

February 19, 2020

Abstract

Recent influential work by Jacot et al. [2018] has shown that training a neural network of any kind with gradient descent in parameter space is strongly related to kernel gradient descent in function space with respect to the Neural Tangent Kernel (NTK). Lee et al. [2019] built on this result by establishing that the output of a neural network trained using gradient descent can be approximated by a linear model for wide networks. In parallel, a recent line of studies Schoenholz et al. [2017], Hayou et al. [2019] has suggested that a special initialization known as the Edge of Chaos improves training. In this paper, we bridge the gap between these two concepts by quantifying the impact of the initialization and the activation function on the NTK when the network depth becomes large. 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. [2018a], Zhang et al. [2017]. Although the loss function is not convex, Gradient Descent (GD) methods are often used successfully to learn these models. It has been actually recently shown that for certain overparameterized deep ReLU networks, GD converges to global minima [Du et al., 2018b]. Similar results have been obtained for Stochastic Gradient Descent (SGD) [Zou et al., 2018].

The training dynamics of wide neural networks with GD is directly linked to kernel methods. Indeed, Jacot et al. [2018] showed that training a neural network with GD

in parameter space is equivalent to a GD in a function 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 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 a method to approximate the NTK for different architectures. Finally, Arora et al. [2019] proposed an efficient algorithm to compute the NTK for convolutional architectures (Convolutional NTK). In all of these papers, the authors only studied the effect of infinite width on the NTK. The aim of this paper is to tackle the infinite depth limit.

In parallel, the impact of the initialization and the 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 provide a comprehensive analysis of the forward and backward propagation of some quantities through the network at the initialization step as a function of the initialization hyperparameteres and the activation function. They propose a set of parameters and activation functions 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. In this paper, we link the initialization hyper-parameters and the activation function to the behaviour of the NTK which controls the training of DNNs. 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 (see also Clevert et al. [2016]). However, we show that with ResNet, the NTK has good properties which good explain why this type of networks outperforms other architectures on many tasks. 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 Motivation and Related work

Neural Tangent Kernel

Jacot et al. [2018] showed that in the infinite width limit, the NTK converges to a kernel which remains constant during training. Arora et al. [2019] proposed an algorithm to compute the NTK for convolutional neural networks. However, for finite width neural networks, Arora et al. [2019] observed a gap between the performances of the linear model derived from the NTK and the deep neural network, which is mostly due to the fact that the NTK changes with time. To fill this gap, Huang and Yau [2019] studied the dynamics of the NTK as a function of the training time for finite width neural networks and showed that the NTK dynamics follow an infinite hierarchy of ordinary differential equations baptised Neural Tangent Hierarchy (NTH), which is extremely complex. Hence Jacot et al. [2018]'s approach can be seen as a simplified version of the training dynamics of DNN. We believe that it still allows us to understand some interesting aspects of the training of DNN and in this paper, we follow this idea to study the behaviour of the NTK as the network depth goes to infinity, i.e. we assume we have an infinite width for Fully-Connected Feedforward Neural Networks and an infinite number of channels for Convolutional Neural Networks.

Edge of Chaos and Activation Function

In Schoenholz et al. [2017] and Hayou et al. [2019], the authors demonstrate that only a special initialization known as the Edge of Chaos (EOC) can make very deep (with infinite widths) models trainable. The EOC separates two phases : an ordered phase where the output function of the DNN is constant, because the correlation of the outputs of two different inputs converges to 1 as the number of layers becomes large, and a chaotic phase where the output function is discontinuous almost everywhere (In this case, the correlation between the outputs of two different inputs converges to a value c such that $|c| < 1$, therefore, very close inputs may lead to very different outputs). In Hayou et al. [2019], the authors give a comprehensive analysis of the EOC, and further show that a certain class of smooth activation functions outperform ReLU-like activation functions in term of test accuracy.

Our contributions

In this paper, we bridge the gap between NTK and Edge of Chaos Initialization. Our main results are

1. With an Initialization on the ordered/chaotic phase, the NTK converges exponentially to a constant kernel with respect to the depth L, making the training

impossible for DNNs (Lemma 1 and Proposition 2).

2. The EOC initialization leads to an invertible NTK even in the infinite depth limit, making the model trainable even for very large depths (Theorem 1).

3. The EOC initialization leads to a sub-exponential convergence rate of the NTK to the limiting NTK (w.r.t to L), which means that the NTK is still expressive for very large depths (Theorem 1).

4. A certain class $\mathcal{S}$ of smooth activation functions can further slow this convergence, making this class more suitable for DNNs.

5. With ResNet, we no longer need the EOC initialization, and the NTK always converges to the limiting NTK at a polynomial rate (Theorem 2).

3 Neural Networks and Neural Tangent Kernel

3.1 Setup and notations

Consider a neural network model consisting of L layers $(y^l)_{1\leq l\leq L}$, with $y^l : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$, $n_0 = d$ 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 θ. Recall that θ^l has dimension $n_l + 1$. The output f of the neural network is given by some transformation $s : \mathbb{R}^{n_L} \to \mathbb{R}^o$ 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, θ changes with time t and we denote by θ_t the value of θ 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 $\mathcal{X} = (x_i)_{1\leq i\leq N}$, $\mathcal{Y} = (y_j)_{1\leq j\leq N}$ be the matrices of input and output respectively, with dimension $d \times N$ and $o \times N$. For any function $g : \mathbb{R}^{d\times o} \to \mathbb{R}^k$, $k \geq 1$, 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 θ, 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

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

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

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

This differs from the result by Lee et al. [2019] since we use a discretization step of $\Delta t = \eta$. It is well known that this discretization scheme leads to an error of order $\mathcal{O}(\eta)$ (see Appendix). 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 $f_t(x) = f(x, \theta_t) \in \mathbb{R}^o$ satisfies the following ODE

$$df_t(x) = -\frac{1}{N}\nabla_\theta f(x, \theta_t)\nabla_\theta f(\mathcal{X}, \theta_t)^T \nabla_z \ell(f_t(\mathcal{X}), \mathcal{Y})dt. \tag{3}$$

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

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

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

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

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

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

meaning that for all $j \leq N$

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

Infinite width dynamics : In the case of a FFNN, Jacot et al. [2018] proved that, with GD, the kernel $K_{\theta_t}^L$ converges to a kernel K^L which depends only on L

(number of layers) for all $t < T$ when $n_1, n_2, ..., n_L \to \infty$, where T is an upper bound on the training time, under the technical assumption $\int_0^T ||\nabla_z \ell(f_t(\mathcal{X}, \mathcal{Y}))||_2 dt < \infty$ a.s. with respect to the initialization weights. The infinite width limit of the training dynamics is given by

$$df_t(\mathcal{X}) = -\frac{1}{N} K^L(\mathcal{X}, \mathcal{X}) \nabla_z \ell(f_t(\mathcal{X}), \mathcal{Y}) dt, \tag{6}$$

We note hereafter $\hat{K}^L = K^L(\mathcal{X}, \mathcal{X})$. As an example, with the quadratic loss $\ell(z, y) = \frac{1}{2}||z - y||^2$, (6) is equivalent to

$$df_t(\mathcal{X}) = -\frac{1}{N} \hat{K}^L (f_t(\mathcal{X}) - \mathcal{Y}) dt, \tag{7}$$

which is a simple linear model that has a closed-form solution given by

$$f_t(\mathcal{X}) = e^{-\frac{1}{N}\hat{K}^L t} f_0(\mathcal{X}) + (I - e^{-\frac{1}{N}\hat{K}^L t})\mathcal{Y}. \tag{8}$$

For general input $x \in \mathbb{R}^d$, we have

$$f_t(x) = f_0(x) + \gamma(x, \mathcal{X})(I - e^{-\frac{1}{N}\hat{K}^L t})(\mathcal{Y} - f_0(\mathcal{X})). \tag{9}$$

where $\gamma(x) = K^L(x, \mathcal{X}) K^L(\mathcal{X}, \mathcal{X})^{-1}$. In order for $f_t(x)$ to be defined, $\hat{K}^L$ must be invertible. Thus, training with dynamics 6 is only possible if the NTK is invertible.

Lemma 1 (Trainability of the Neural Network and Invertibility of the NTK)**.** *Assume $f_0(\mathcal{X}) \neq \mathcal{Y}$. Then with dynamics defined by* (16)*, $||f_t(\mathcal{X}) - \mathcal{Y}||$ converges to 0 as $t \to \infty$ if and only if $\hat{K}^L$ is non-singular.*
Moreover, if $\hat{K}^L$ is singular, there exists a constant $C > 0$ such that for all $t > 0$,

$$||f_t(\mathcal{X}) - \mathcal{Y}|| \geq C.$$

Lemma 1 shows that an invertible NTK is crucial for trainability. Since $K^L_{\theta_t}$ is constant w.r.t to training time, it is completely determined at initialization. Thus, it is intuitive to study the impact of the initialization on the NTK as the depth L grows (DNN), which is our focus in this paper. Another interesting aspect is the impact of the NTK on the generalization error of the neural network model. If the NTK is constant for example (there exists a constant δ such that $K^L(x, x') = \delta$ for all $x \neq x'$, this example is useful in the next section), then the second part of $f_t(x)$ in equation 17 is constant w.r.t x. Therefore, the generalization function $f_t(x)$ of the model 17 is entirely given by its value at time zero $f_0(x)$, which means that the generalization error $\mathbb{E}_{x,y}[\|f_t(x) - y\|]$ remains of order $\mathcal{O}(1)$. In the next section, we show that the initialization and the activation function have major impact on the invertibility and 'expressivity' of NTK. More precisely, we show that :

1. Under some constraints, the NTK K^L (or a scaled version of the NTK) converges to a limiting NTK K^∞ as L goes to infinity (otherwise it diverges)

2. A special initialization known as the Edge of Chaos (EOC) leads to an invertible K^∞ which makes it useful for training DNNs

3. The EOC initialization gives a sub-exponential rate for this convergence (w.r.t L), which means for the same depth L, the EOC gives 'richer' limiting NTK, and therefore leading to better generalization properties

4. The smoothness of the activation can further slow this convergence, leading to 'richer' limiting NTK (the convergence to the limiting trivial kernel is slower)

5. With ResNets, the convergence rate for the NTK (w.r.t to L) is always sub-exponential and we no longer need the Edge of Chaos

4 Impact of the Initialization and the Activation function on the Neural Tangent Kernel

In this section, We give a comprehensive analysis of the behaviour of the NTK as L goes to infinity for different architectures : Fully-Connected FeedForward Neural Networks (FFNN), Convolutional Neural Networks (CNN) and Residual Neural Networks (ResNet). For the first two architectures, we prove that only an initialization on the Edge of Chaos (EOC) leads to an invertible NTK for deep 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.

4.1 NTK parameterization and the Edge of Chaos

Let ϕ be the activation function. We consider the following architectures (FFNN and CNN) where we generalize the NTK parameterization in Jacot et al. [2018]

- **FeedForward Fully-Connected Neural Network (FFNN)**
  Conside 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 using the NTK parameterization is given by

$$\begin{aligned} y_i^1(x) &= \frac{\sigma_w}{\sqrt{d}} \sum_{j=1}^{d} w_{ij}^1 x_j + \sigma_b b_i^1 \\ y_i^l(x) &= \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_{j=1}^{n_{l-1}} w_{ij}^l \phi(y_j^{l-1}(x)) + \sigma_b b_i^l, \quad l \geq 2 \end{aligned} \tag{10}$$

- **Convolutional Neural Network (CNN/ConvNet)**
  Consider a 1D convolutional neural network of depth L, the forward propagation is given by

$$\begin{aligned} y^1_{i,\alpha}(x) &= \frac{\sigma_w}{\sqrt{v_1}} \sum_{j=1}^{n_0} \sum_{\beta \in ker_1} w^1_{i,j,\beta} x_{j,\alpha+\beta} + \sigma_b b^1_i \\ y^l_{i,\alpha}(x) &= \frac{\sigma_w}{\sqrt{v_l}} \sum_{j=1}^{n_{l-1}} \sum_{\beta \in ker_l} w^l_{i,j,\beta} \phi(y^{l-1}_{j,\alpha+\beta}(x)) + \sigma_b b^l_i \end{aligned} \tag{11}$$

  where $i \in [|1, n_l|]$ is the channel number, $\alpha \in [|0, M_l - 1|]$ is the neuron location in the channel, n_l is the number of channels in the l^{th} layer and M_l is the number of neurons in each channel, $ker_l = [|-k, k|]$ is a filter with size $2k+1$ and $v_l = n_{l-1}(2k+1)$. Here, $w^l \in \mathbb{R}^{n_l \times n_{l-1} \times (2k+1)}$. We assume periodic boundary conditions, which results in having $y^l_{i,\alpha} = y^l_{i,\alpha+M_l} = y^l_{i,\alpha-M_l}$ and similarly for $l = 0$, $x_{i,\alpha+M_0} = x_{i,\alpha} = x_{i,\alpha-M_0}$.
  For the sake of simplification, we consider only the case of 1D CNN, the generalization for a mD CNN where $m \in \mathbb{N}$ is straightforward.

We initialize the model with $w^l_{ij}, b^l_i \overset{iid}{\sim} \mathcal{N}(0,1)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution of mean μ and variance σ^2. To simplify the analysis, we assume hereafter $M_l = M$ and $k = k$ for all l (constant number of neurons per channel and constant filter size). We denote by $[|n, m|]$ the set of integers $\{n, n+1, ..., m\}$ for $n \leq m$. Jacot et al. [2018] established the following infinite width limit of the NTK of a FFNN.

Lemma 2 (Th. 1 in Jacot et al. [2018])**.** *Let x, x' be two inputs. We consider a FFNN of the form (18). Then, as $n_1, n_2, ..., n_{L-1} \to \infty$, we have for all $i, i' \in [|1, n_l|]$, $K^L_{ii'}(x, x') = \delta_{ii'} K^L(x, x')$, where $K^L(x, x')$ is given by the recursive formula*

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

We generalize this result to Convolutional Neural Networks in the next proposition.

Proposition 1 (Infinite width dynamics of the NTK of a CNN)**.** *Let $x, x' \in \mathbb{R}^d$. Consider a CNN of the form (19), we then have that for all n_0,*

$$K^1_{(i,\alpha),(i',\alpha')}(x, x') = \delta_{ii'} \Big(\frac{\sigma_w^2}{n_0(2k+1)} [x, x']_{\alpha,\alpha'} + \sigma_b^2 \Big),$$

where $[x, x']_{\alpha,\alpha'} = \sum_{j=1}^{n_0} \sum_{\beta \in ker_0} x_{j,\alpha+\beta} x_{j,\alpha'+\beta}$

For $l \geq 2$, as $n_1, n_2, ..., n_{l-1} \to \infty$ recursively, we have for all $i, i' \leq n_l$, $\alpha, \alpha' \in [|0, M_l - 1|]$, $K^l_{(i,\alpha),(i',\alpha')}(x, x') = \delta_{ii'} K^l_{\alpha,\alpha'}(x, x')$, where $K^l_{\alpha,\alpha'}$ is given by the recursive formula

$$K^l_{\alpha,\alpha'} = \frac{1}{2k+1} \sum_{\beta \in ker_l} \left[\dot{\Sigma}^l_{\alpha+\beta,\alpha'+\beta} K^{l-1}_{\alpha+\beta,\alpha'+\beta} + \Sigma^l_{\alpha+\beta,\alpha'+\beta} \right]$$

where $\Sigma^l_{\alpha,\alpha'} = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y^{l-1}_{1,\alpha}(x))\phi(y^{l-1}_{1,\alpha'}(x'))]$ and $\dot{\Sigma}^l_{\alpha,\alpha'} = \sigma_w^2 \mathbb{E}[\phi'(y^{l-1}_{1,\alpha}(x))\phi'(y^{l-1}_{1,\alpha'}(x'))]$.

The NTK of a CNN differs from that of a FFNN in the sense that it is an average over the NTK values of the previous layer. This is due to the fact that neurons at the same channel are not independent at initialization (the opposite is true for FFNN). Now that we have recursive formulas for the NTK, we present the theory of the Edge of Chaos which will help us better understand the dynamics of the NTK as L goes to infinity.

Edge of Chaos : For some input 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]. Under some regularity conditions, the authors prove that $q^l(x)$ converges to a point $q(\sigma_b, \sigma_w) > 0$ independent of x as $l \to \infty$. 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 (σ_b, σ_w); the authors define the Edge of Chaos (EOC) as the set of parameters (σ_b, σ_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. Let $c^l(x, x')$ be the correlation between $y^l_1(x)$ and $y^l_1(x')$. In Schoenholz et al. [2017], the authors showed that an Initialization on the Ordered/Chaotic phase leads to the convergence of c^l to a limiting value c independent of x, x' with an exponential convergence rate, whereas in Hayou et al. [2019], the authors showed that an initialization on the Edge of Chaos leads to sub-exponential rates ($\mathcal{O}(l^{-2})$ for ReLU and $\mathcal{O}(l^{-1})$ for a class of smooth activation functions). 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 ϕ be an activation function. Then*

1. *ϕ 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. *ϕ is said to be in $\mathcal{S}$ if $\phi(0) = 0$, ϕ 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 $g_1, g_2, ..., g_n$ such that $\phi^{(2)} = \sum_{i=1}^n 1_{A_i} g_i$, where $\phi^{(2)}$ is the second derivative of ϕ.*

The class of ReLU-like activations includes ReLU and Leaky-ReLU, whereas the $\mathcal{S}$ class includes, among others, Tanh, ELU and SiLU (Swish).

To alleviate notations, we use hereafter the notation K^L for the NTK of both FFNN and CNN. For FFNN, it represents the recursive kernel K^L given by lemma 2, whereas for CNN, it represents the recursive kernel $K^L_{\alpha,\alpha'}$ for any α, α', which means all results that follow are true for any α, α'.

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 (σ_b, σ_w) be either in the ordered or in the chaotic phase. Then, there exist $\lambda, \gamma > 0$ such that*

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

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 $K^L(\mathcal{X}, \mathcal{X}) \approx \lambda \mathbf{1}_{oN}$ where $\mathbf{1}_{oN}$ is the $oN \times oN$ matrix whose elements are equal to one, i.e. $\hat{K}^L$ is at best degenerate and asymptotically (in L) non invertible, making the training impossible by Lemma 1. We illustrate empirically this result in Section 6.

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

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

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, across l. Hence, it is more convenient to study the average NTK (ANTK hereafter) given by $AK^L = K^L/L$. Note that the invertibility of the NTK is equivalent to that of the ANTK. The next proposition shows that on the EOC, the ANTK converges to an invertible kernel AK^∞ as L goes to infinity, at a sub-exponential rate. Moreover, by choosing an activation function in the class $\mathcal{S}$, we can slow the convergence of ANTK with respect to L, which means that, for the same depth L, a smooth activation function from the class $\mathcal{S}$ leads to 'richer' NTK which is crucial for the generalization error of deep models as discussed in Section 3. This confirms the findings in [Hayou et al., 2019]. Hereafter, the notation $g(x) = \Theta(m(x))$ means there exist two constants $A, B > 0$ such that $Am(x) \leq g(x) \leq Bm(x)$.

Theorem 1 (Neural Tangent Kernel on the Edge of Chaos)**.** *Let ϕ be a non-linear activation function, $(\sigma_b, \sigma_w) \in EOC$ and $AK^L = K^L/L$.*

1. *If ϕ is ReLU-like, then for all $x \in \mathbb{R}^d, AK^L(x,x) = AK^\infty(x,x) + \Theta(L^{-1})$. Moreover, there exist $\lambda \in (0,1)$ such that*

$$\sup_{x \neq x' \in \mathbb{R}^d} \left| AK^L(x,x') - AK^\infty(x,x') \right| = \Theta(L^{-1})$$

where $AK^\infty(x,x') = \frac{\sigma_w^2 \|x\| \|x'\|}{d}(1 - (1-\lambda)\mathbb{1}_{x \neq x'})$.

2. *If ϕ is in $\mathcal{S}$, then, there exists $q > 0$ such that $AK^L(x,x) = AK^\infty(x,x) + \Theta(L^{-1}) \to q$. Moreover, there exist $\lambda \in (0,1)$ such that*

$$\sup_{x \neq x' \in \mathbb{R}^d} \left| AK^L(x,x') - AK^\infty(x,x') \right| = \Theta(\log(L) L^{-1})$$

where $AK^\infty(x,x') = q(1 - (1-\lambda)\mathbb{1}_{x \neq x'})$.

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 $L \to \infty$. 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, contrariwise to an initialization on the Ordered/Chaotic phase (see Proposition 2). However the limiting ANTK AK^∞ carry (almost) no information on x, x' and is therefore little expressive. Interestingly the convergence rate of the ANTK to AK^∞ 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 remains expressive compared to the Ordered/Chaotic phase case (exponential convergence rate). This is particularly important for the generalization part (see equation 17). 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 6).

4.2 Residual Neural Networks (ResNet)

Another important feature of deep neural networks which is known to be highly influential is their architecture. For residual networks, the NTK has also a simple recursive formula in the infinite width limit. The residual term appears clearly in the formula.

Lemma 3 (NTK of a ResNet with Fully Connected layers in the infinite width limit)**.** *Let x, x' be two inputs and $K^{res,1}$ be the exact NTK for the Residual Network with 1 layer. Then, we have*

- *For the first layer (without residual connections), we have for all* $x, x' \in \mathbb{R}^d$

$$K_{ii'}^{res,1}(x, x') = \delta_{ii'}(\sigma_b^2 + \frac{\sigma_w^2}{d} x \cdot x'),$$

where $x \cdot x'$ *is the inner product in* $\mathbb{R}^d$.

- *For* $l \geq 2$, *as* $n_1, n_2, ..., n_{l-1} \to \infty$ *recursively, we have for all* $i, i' \in [|1, n_l|]$, $K_{ii'}^{res,l}(x, x') = \delta_{ii'} K_{res}^l(x, x')$, *where* $K_{res}^l(x, x')$ *is given by the recursive formula have for all* $x, x' \in \mathbb{R}^d$ *and* $l \geq 2$, *as* $n_1, n_2, ..., n_l \to \infty$ *recursively, we have*

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

For residual networks with convolutional layers, the formula is very similar to the non residual case. Only an additional residual term appears in the recursive formula.

Lemma 4 (NTK of a ResNet with Convolutional layers in the infinite width limit)**.** *Let* x, x' *be two inputs and* $K^{res,1}$ *be the exact NTK for the Residual Network with 1 layer. Then, we have*

- *For the first layer (without residual connections), we have for all* $x, x' \in \mathbb{R}^d$

$$K_{(i,\alpha),(i',\alpha')}^{1,res}(x, x') = \delta_{ii'}\Big(\frac{\sigma_w^2}{n_0(2k+1)}[x, x']_{\alpha,\alpha'} + \sigma_b^2\Big)$$

where $[x, x']_{\alpha,\alpha'} = \sum_j \sum_\beta x_{j,\alpha+\beta} x_{j,\alpha'+\beta}$.

- *For* $l \geq 2$, *as* $n_1, n_2, ..., n_{l-1} \to \infty$ *recursively, we have for all* $i, i' \in [|1, n_l|]$, $\alpha, \alpha' \in [|0, M-1|]$, $K_{(i,\alpha),(i',\alpha')}^{res,l}(x, x') = \delta_{ii'} K_{\alpha,\alpha'}^{res,l}(x, x')$, *where* $K_{\alpha,\alpha'}^{res,l}$ *is given by the recursive formula*

$$K_{\alpha,\alpha'}^{res,l} = K_{\alpha,\alpha'}^{res,l-1} + \frac{1}{2k+1} \sum_\beta \Big[\dot{\Sigma}_{\alpha+\beta,\alpha'+\beta}^l K_{\alpha+\beta,\alpha'+\beta}^{res,l-1} + \Sigma_{\alpha+\beta,\alpha'+\beta}^l\Big]$$

where $\Sigma_{\alpha,\alpha'}^l = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y_{1,\alpha}^{l-1}(x))\phi(y_{1,\alpha'}^{l-1}(x'))]$ *and* $\dot{\Sigma}_{\alpha,\alpha'}^l = \sigma_w^2 \mathbb{E}[\phi'(y_{1,\alpha}^{l-1}(x))\phi'(y_{1,\alpha'}^{l-1}(x'))]$.

The additional term in the recursive formulas of the NTK for ResNet is due to the ResNet architecture. It turns out that this term helps avoid having an exponential convergence rate of the NTK. The next proposition shows that for any $\sigma_w > 0$, the scaled NTK of a ResNet will always have a subexponential convergence rate to a limiting AK_{res}^∞. We say that ResNet 'live' on the Edge of Chaos.

Theorem 2 (NTK for ResNet)**.** *Consider a Residual Neural Network with the following forward propagation equations*

$$y^l(x) = y^{l-1}(x) + \mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2, \tag{12}$$

where $\mathcal{F}$ is either a convolutional or dense layer (equations 18 and 19) with ReLU activation. Let K^L_{res} be the corresponding NTK. Then for all $x \in \mathbb{R}^d$, $\frac{K^L_{res}(x,x)}{\alpha_L} = AK^{\infty}_{res}(x,x) + \Theta(L^{-1})$ and there exists $\lambda \in (0,1)$ such that

$$\sup_{x \neq x' \in \mathbb{R}^d} \Big| \frac{K^L_{res}(x,x')}{\alpha_L} - \frac{||x|| \times ||x'||}{d} \lambda \Big| = \Theta(L^{-1}),$$

where $AK^{\infty}_{res}(x,x') = \frac{\sigma_w^2 \|x\| \|x'\|}{d}(1 - (1-\lambda)\mathbb{1}_{x \neq x'})$, and $\alpha_L = L(1 + \frac{\sigma_w^2}{2})^{L-1}$.

Theorem 2 shows that the NTK of a ReLU ResNet explodes exponentially with respect to L. However, the normalised kernel $K^L_{res}(x,x')/\alpha_L$ where $x \neq x'$ converges to a limiting kernel AK^{∞}_{res} with a rate $\mathcal{O}(L^{-1})$ for all $\sigma_w > 0$. We say that residual networks 'live' on the Edge of Chaos, i.e. no matter what the choice of σ_w is, the convergence rate of the NTK w.r.t L is polynomial and there is no Ordered/Chaotic phase in this case. However, the exponential exploding in the residual NTK might cause a stability issue (NTK is directly linked to the gradients); it turns out that a simple re-parameterization of the ResNet architecture can solve the problem. We introduce a slightly different ResNet architecture that we name Scaled ResNet.

Proposition 3 (Scaled Resnet)**.** *Consider a Residual Neural Network with the following forward propagation equations*

$$y^l(x) = y^{l-1}(x) + \frac{1}{\sqrt{l}} \mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2. \tag{13}$$

where $\mathcal{F}$ is either a convolutional or dense layer (equations 18 and 19) with ReLU activation. Then the scaling factor α_L in Theorem 2 becomes $\alpha_L = L^{1+\sigma_w^2/2}$ and the convergence rate is $\Theta(\log(L)^{-1})$.

Proposition 3 shows that by scaling the residual blocks by $1/\sqrt{l}$ we stabilizes the NTK. In practice, since the NTK controls the training and the generalization, then having a stable NTK is desirable. Moreover, the convergence rate $\Theta(\log(L)^{-1})$ is very slow compared to the standard resnet convergence $\Theta(L^{-1})$ which means the NTK of scaled resnet remains more expressive compared to the NTK of the standard resnet as depth L grows. We illustrate the effectiveness of Scaled Resnet in section 6.

5 Beyond GD : Training dynamics with SGD

The NTK dynamics 3 are satisfied with gradient flow. However, with SGD, dynamics are random and the gradient flow is replaced with a Stochastic Differential Equation. More precisely, under boundedness conditions (see the supplementary material), when using SGD, the gradient update can be seen as a discretization of the following SDE [Hu et al., 2018, Li et al., 2017]

$$d\theta_t = -\nabla_\theta \mathcal{L}(\theta_t)dt + \sqrt{\frac{\eta}{S}}\Sigma(\theta_t)^{\frac{1}{2}}dW_t, \tag{14}$$

where $\Sigma(\theta_t)^{\frac{1}{2}}$ is the square-root matrix of $\Sigma(\theta_t) = \mathrm{Cov}\left(\nabla_\theta \ell(f(X_1, \theta_t), Y_1)\right)$ and $(W_t)_{t\geq 0}$ a standard Brownian motion. With dynamics (31) and the quadratic loss, the output function f_t has the following dynamics

$$\begin{aligned} df_t(\mathcal{X}) = &[-\frac{1}{N}K^L_{\theta_t}(\mathcal{X}, \mathcal{X})(f_t(\mathcal{X}) - \mathcal{Y}) + \frac{1}{2}\frac{\eta}{S}\Gamma_t(\mathcal{X})]dt \\ &+ \sqrt{\frac{\eta}{S}}\nabla_\theta f(\mathcal{X}, \theta_t)\Sigma(\theta_t)^{\frac{1}{2}}dW_t, \end{aligned} \tag{15}$$

where $\Gamma_t(\mathcal{X})$ is the concatenated vector of $(\Gamma_t(x) = (\mathrm{Tr}(\Sigma(\theta_t)^{\frac{1}{2}}\nabla_2 f_i(x, \theta_t)\Sigma(\theta_t)^{\frac{1}{2}}))_{1\leq i\leq o})_{x\in\mathcal{X}}$ and $\nabla_2 f_i(x, \theta)$ is the Hessian of f_i (i^{th} component of f) with respect to θ. This is an Ornstein-Uhlenbeck process (mean-reverting process) with time dependent parameters. The additional term Γ_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]).

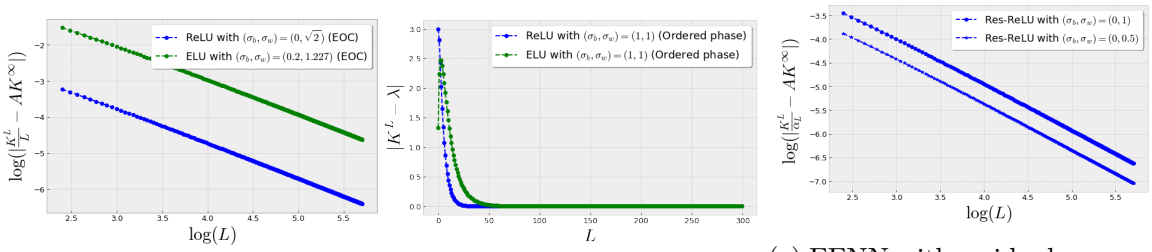
However, with SGD it is still an open question whether the NTK remains constant in time in the limit of infinite width. If we assume this to be true, then the infinite width solution for (33) is given by

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

where $(A_t)_{t\leq T}$ is a random process. The only difference with the GD algorithm is the additional regularization effect of SGD represented by $(A_t)_{t\leq T}$.

6 Experiments

We illustrate empirically the theoretical results obtained in the previous sections. We confirm these results on MNIST, CIFAR10 and CIFAR100 datasets.



(a) EOC (b) Ordered phase (c) FFNN with residual connections

Figure 1: Convergence rates for different initializations and architectures. (a) Edge of Chaos. (b) Ordered phase. (c) Adding residual connections.

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

	MNIST		CIFAR10	
Activation	Epoch 10	Epoch 100	Epoch 10	Epoch 100
ReLU (EOC)	44.53 ± 12.01	83.75 ± 1.32	18.15 ± 3.21	36.62 ± 1.66
$\text{LReLU}_{0.01}$ (EOC)	48.01 ± 4.33	84.35 ± 3.26	19.62 ± 1.13	37.43 ± 3.35
PReLU	49.33 ± 3.32	87.17 ± 1.97	20.63 ± 2.54	37.88 ± 3.77
ELU (EOC)	$\mathbf{75.11 \pm 2.21}$	$\mathbf{97.13 \pm 0.13}$	$\mathbf{31.78 \pm 1.55}$	$\mathbf{48.10 \pm 1.21}$
Tanh (EOC)	71.13 ± 1.53	94.11 ± 0.33	30.11 ± 2.01	47.73 ± 1.02
Softplus	9.89 ± 0.06	10.11 ± 0.17	10.07 ± 0.68	9.92 ± 0.24
Sigmoid	10.05 ± 0.13	10.01 ± 0.09	10.22 ± 0.33	10.15 ± 0.13

6.1 Convergence rate of K^L as L goes to infinity

Proposition 2 and theorems 1 and 2 give theoretical convergence rates for quantities of the form $\left|\frac{K^L}{\alpha_L} - AK^\infty\right|$. We illustrate these results in Figure 1. Figure 1a shows a convergence rate approximately of order L^{-1} for ReLU and ELU. Recall that for ELU the exact rate is $\mathcal{O}(\log(L)L^{-1})$ but one cannot observe experimentally the logarithmic factor. However, ELU performs indeed better than ReLU (see Table 1) which might be partially explained by this $\log(L)$ factor. Figure 1b demonstrates that this convergence occurs at an exponential convergence rate in the Ordered phase for both ReLU and ELU, and Figure 1c shows the convergence rate in the case of FFNN with residual connections. As predicted by theorem 2, the convergence rate $\mathcal{O}(L^{-1})$ is independent of the parameter σ_w in that case.

6.2 Impact of the initialization and smoothness of the activation on the overall performance

We train FFNNs of width 300 and depths 300 with SGD with a batchsize of 128 and a learning rate 10^{-2} (this learning rate was found by a grid search of exponential step size 10; note that the optimal learning rate with NTK parameterization is usually bigger than the optimal learning rate with standard parameterization). 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, $(\sigma_b, \sigma_w) = (0.2, 1.225)$ for ELU and $(\sigma_b, \sigma_w) = (0.2, 1.298)$ 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. Functions in class $\mathcal{S}$ (ELU and Tanh) perform much better than ReLU-like activation functions (ReLU, Leaky ReLU with parameter $\alpha = 0.01$). 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.

6.3 Resnet and Scaled Resnet

To illustrate the theoretical result obtained in proposition 3, we train ResNet with depths 32, 50 and 104 on CIFAR100 with SGD. We use a decaying learning rate schedule; we start with 0.1 and divide by 10 after $n_e/2$ epochs where n_e is the total number of epochs, we scale again by 10 after $n_e/4$ epochs. We use a batch size of 128 and we train the model with 160 epochs. Theoretically, since the NTK of Scaled ResNet is stable compared to that of standard ResNet and since the NTK plays a crucial role in the genezalization function 17, we expect the test accuracy of Scaled Resnet to be better than standard ResNet when the depth becomes large. Table 2 displays test accuracy for standard ResNet and scaled ResNet after 10 and 160 epochs. Scaled ResNet converges faster than standard ResNet for depths 50 and 104 which confirms our result.

Table 2: Test accuracy on CIFAR100 for ResNet with varying depths

	scaling	Epoch 10	Epoch 160
ResNet32	standard	**54.18±1.21**	72.49±0.18
	scaled	53.89±2.32	**74.07±0.22**
ResNet50	standard	51.09±1.73	73.63±1.51
	scaled	**55.39±1.52**	**75.02±0.44**
ResNet104	standard	47.02±3.23	74.77±0.29
	scaled	**56.38±2.54**	**76.14±0.98**

7 Conclusion

Jacot et al. [2018] showed that the training dynamics of DNNs with GD is linked to a GD in function space with respect to the NTK. In the infinite width limit, the NTK has a closed-form expression. This approximation sheds light on how the NTK impacts the training dynamics: it controls the training rate and the generalization function. Using this approximation for wide neural networks (Mean-field approximation), we show that with an initialization in depth 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. Moreover, with ResNet, the convergence rate is always polynomial which make them better suited for deep architectures.

However, recent findings showed that the infinite width approximation of the NTK does not fully capture the dynamics of the training of DNNs. A recent line of work showed that the NTK for finite width neural networks changes with time and might even be random (Chizat and Bach [2018], Ghorbani et al. [2019], Huang and Yau [2019], Arora et al. [2019]). Therefore, we believe that the NTK is a useful tool to partially understand wide deep neural networks (have insights on hyper-parameters choices for example) and not a tool to train neural networks.

References

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *32nd Conference on Neural Information Processing Systems*, 2018.

J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

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

S. Hayou, A. Doucet, and J. Rousseau. On the impact of the activation function on deep neural networks training. *ICML*, 2019.

Q. Nguyen and M. Hein. Optimization landscape and expressivity of deep CNNs. *ICML*, 2018.

S.S. Du, J.D. Lee, Y. Tian, B. Poczos, and A Singh. Gradient descent learns one-hidden-layer CNN: Don't be afraid of spurious local minima. *ICML*, 2018a.

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

S.S. Du, J.D. Lee, H. Li, L. Wang, and X. Zhai. Gradient descent finds global minima of deep neural networks. *arXiv preprint arXiv:1811.03804*, 2018b.

D. Zou, Y. Cao, D. Zhou, and Q. Gu. Stochastic gradient descent optimizes over-parameterized deep ReLU networks. *arXiv preprint arXiv:1811.08888*, 2018.

S.S. Du, X. Zhai, B. Poczos, and A. Singh. Gradient descent provably optimizes over-parameterized neural networks. *ICLR*, 2019.

R. Karakida, S. Akaho, and S. Amari. Universal statistics of Fisher information in deep neural networks: Mean field approach. *arXiv preprint arXiv:1806.01316*, 2018.

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

S. Arora, S.S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019.

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.

G. Yang and S. Schoenholz. Mean field residual networks: On the edge of chaos. *Advances in Neural Information Processing Systems*, 30:2869–2869, 2017.

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

J. Huang and H.T Yau. Dynamics of deep neural networks and neural tangent hierarchy. *arXiv preprint arXiv:1909.08156*, 2019.

W. Hu, C. Junchi Li, L. Li, and J Liu. On the diffusion approximation of nonconvex stochastic gradient descent. *arXiv preprint arXiv:1705.07562*, 2018.

Q. Li, C. Tai, and W E. Stochastic modified equations and adaptive stochastic gradient algorithms. *arXiv preprint arXiv:1511.06251*, 2017.

M. Kubo, R. Banno, H. Manabe, and M. Minoji. Implicit regularization in over-parameterized neural networks. *arXiv preprint arXiv:1903.01997*, 2019.

D. Lei, Z. Sun, Y. Xiao, and W.Y. Wang. Implicit regularization of stochastic gradient descent in natural language processing: Observations and implications. *arXiv preprint arXiv:1811.00659*, 2018.

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

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *International Conference on Artificial Intelligence and Statistics*, 2010.

L. Chizat and F. Bach. A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari. Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*, 2019.

We provide the proofs for theoretical results. Results named "Appendix Lemma" or "Appendix Proposition" are results that are in the appendix but not in the main paper.

A Proofs of Section 3: Neural Networks and Neural Tangent Kernel

Recall the linear model approximation

$$f_t(\mathcal{X}) = e^{-\frac{1}{N}\hat{K}^L t} f_0(\mathcal{X}) + (I - e^{-\frac{1}{N}\hat{K}^L t})\mathcal{Y}. \tag{16}$$

For general input $x \in \mathbb{R}^d$, we have

$$f_t(x) = f_0(x) + \gamma(x, \mathcal{X})(I - e^{-\frac{1}{N}\hat{K}^L t})(\mathcal{Y} - f_0(\mathcal{X})) \tag{17}$$

where $\gamma(x) = K^L(x, \mathcal{X})K^L(\mathcal{X}, \mathcal{X})^{-1}$.

Lemma 1 (Trainability of the Neural Network and Invertibility of the NTK)**.** *Assume $f_0(\mathcal{X}) \neq \mathcal{Y}$. Then with dynamics (16), $||f_t(\mathcal{X}) - \mathcal{Y}||$ converges to 0 as $t \to \infty$ if and only if $\hat{K}^L$ is non-singular.*
Moreover, if K^L is singular, there exists a constant $C > 0$ such that for all $t > 0$,

$$||f_t(\mathcal{X}) - \mathcal{Y}|| \geq C$$

Proof. Assume $f_0(\mathcal{X}) \neq \mathcal{Y}$. Let $\hat{K}^L = Q^T D Q$ be the spectral decomposition of the empirical NTK; i.e. Q is an orthogonal matrix and D is a diagonal matrix.

We have that $e^{-\frac{1}{N}\hat{K}^L t} = Q^T e^{-\frac{1}{N}Dt} Q = Q^T Diag(e^{-\frac{d_i}{N}t})_{1 \leq i \leq oN} Q$ where $(d_i)_{1 \leq i \leq oN}$ are the eigenvalues. We also have $||f_t(\mathcal{X}) - \mathcal{Y}|| = ||e^{-\frac{1}{N}\hat{K}^L t}(f_0(\mathcal{X}) - \mathcal{Y})||$. Therefore, the equivalence holds true.

Moreover, assume $\hat{K}^L$ in singular. Let $Z_t = Q(f_t(\mathcal{X}) - \mathcal{Y})Q^T$. We have that $Z_t = e^{-\frac{1}{N}Dt} Z_0$. Since D has at least one zero diagonal value, then there exists $j \in \{1, 2, ..., oN\}$ such that for all t, $(Z_t)_j = (Z_0)_j$, and we have

$$\begin{aligned} ||f_t(\mathcal{X}) - \mathcal{Y}|| &= ||Z_t|| \\ &\geq |(Z_t)j| = |(Z_0)j| \end{aligned}$$

□

The next Lemma gives an upper bound on the approximation of the GD algorihtm by the Gradient flow.

Appendix 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. □

B Proofs of Section 4: Impact of the Initialization and the Activation function on the Neural Tangent Kernel

Let ϕ be the activation function. We consider the following architectures (FFNN and CNN)

- **FeedForward Fully-Connected Neural Network (FFNN)**
  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 using the NTK parameterization (introduced in Jacot et al. [2018]) is given by

$$\begin{aligned} y_i^1(x) &= \frac{\sigma_w}{\sqrt{d}} \sum_{j=1}^{d} w_{ij}^1 x_j + \sigma_b b_i^1 \\ y_i^l(x) &= \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_{j=1}^{n_{l-1}} w_{ij}^l \phi(y_j^{l-1}(x)) + \sigma_b b_i^l, \quad l \geq 2 \end{aligned} \tag{18}$$

- **Convolutional Neural Network (CNN/ConvNet)**
  1D convolutional neural network of depth L, For some input x, the forward propagation is given by

$$y_{i,\alpha}^l(x) = \frac{\sigma_w}{\sqrt{v_l}} \sum_{j=1}^{n_{l-1}} \sum_{\beta \in ker_l} w_{i,j,\beta}^l \phi(y_{j,\alpha+\beta}^{l-1}(x)) + \sigma_b b_i^l \tag{19}$$

  where $i \in [|1, n_l|]$ is the channel number, $\alpha \in [|0, N_l - 1|]$ is the neuron index in the channel, n_l is the number of channels in the l^{th} layer, $ker_l = [|-k_l, k_l|]$ is a filter with size $2k_l + 1$ and $v_l = n_{l-1}(2k_l + 1)$. Here, $w^l \in \mathbb{R}^{n_l \times n_{l-1} \times (2k_l+1)}$. We assume periodic boundary conditions, which results in having $y_{i,\alpha}^l = y_{i,\alpha+N_l}^l = y_{i,\alpha-N_l}^l$. For the sake of simplification, we consider only the case of 1D CNN, the generalization for a $m\times$D CNN is straightforward.

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

We initialize the model with $w_{ij}^l, b_i^l \overset{iid}{\sim} \mathcal{N}(0, 1)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution of mean μ and variance σ^2. For some x, we denote by $q^l(x)$ the variance of $y^l(x)$ where y^l is some neuron in the l^{th} layer. In general, $q^l(x)$ converges exponentially to a point $q(\sigma_b, \sigma_w) > 0$ independent of x and the neuron index as $l \to \infty$ (see

Schoenholz et al. [2017] for FFNN and **?** for CNN) . The EOC is defined by the set of parameters (σ_b, σ_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 $c^{l+1} = f(c^l)$.

B.1 Preliminary results

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

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

  More precisely,

  $$f(x) \underset{x \to 1-}{=} x + \frac{2\sqrt{2}}{3\pi}(1-x)^{3/2} + \sum_{k=2} \alpha_k (1-x)^{k+1/2}$$

  where $\alpha_k > 0$.
- 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 $Z(x) = xZ_1 + \sqrt{1-x^2}Z_2$ and Z_1, Z_2 are iid standard Gaussian variables.
- On the EOC, we have $f'(1) = 1$

- For non-linear activatiion functions, f is strictly convex and $f(1) = 1$.
- If f is infinitely differentiable, then for all $j \geq 1$, we have $f^{(j)}(x) = \sigma_w^2 q^{j-1} \mathbb{E}[\phi^{(j)}(Z_1)\phi^{(j)}(Z(x))]$. Moreover, since $f^{(j)}(1) = \sigma_w^2 q^{j-1} \mathbb{E}[\phi^{(j)}(Z_1)^2] > 0$, then Taylor expansion of f near 1 has coefficients that are all positive.

Gradient Independence : In **?**, authors show that we can assume that the weights used for forward propagation are independent of those used for backpropagation for usual architectures. We use this assumption in our proofs.

From Jacot et al. [2018], we have that

Lemma 2 (Th. 1 in Jacot et al. [2018])**.** *Let x, x' be two inputs. We consider a FFNN of the form 18. Then, as $n_1, n_2, ..., n_{L-1} \to \infty$, we have for all $i, i' \in [|1, n_l|]$, $K^L_{ii'}(x, x') = \delta_{ii'} K^L(x, x')$, where $K^L(x, x')$ is given by the recursive formula*

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

For CNNs, the mean field approximation is considered in the limit of infinite number of channels. We define $\Sigma^l_{\alpha,\alpha'}(x, x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y^{l-1}_{1,\alpha}(x))\phi(y^{l-1}_{1,\alpha'}(x'))]$ and $q^l_{\alpha,\alpha'}(x, x') = \mathbb{E}[y^l_{i,\alpha}(x) y^l_{i,\alpha'}(x')]$. From **?**, there exist $q, \lambda > 0$ such that, for all α, α' $\sup_{x \in \mathbb{R}^d} |q^l_{\alpha,\alpha'}(x, x) - q| \leq e^{-\lambda l}$.

Unlike FFNN, neurons in the same channel are correlated since we use the same filters for all of them. Let x, x' be two inputs and α, α' two nodes in the same channel i. Using Central Limit Theorem in the limit of large c (number of channels), we have

$$q^l_{\alpha,\alpha'}(x, x') = \mathbb{E}[y^l_{i,\alpha}(x) y^l_{i,\alpha'}(x')] = \frac{\sigma_w^2}{2k+1} \sum_{\beta \in ker} \mathbb{E}[\phi(y^{l-1}_{1,\alpha+\beta}(x))\phi(y^{l-1}_{1,\alpha'+\beta}(x'))] + \sigma_b^2$$

Let $c^l_{\alpha,\alpha'}(x, x')$ be the corresponding correlation. As in **?** and Hayou et al. [2019], authors show that the the variance of each node for the same input $q^l_{\alpha,\alpha}(x, x)$ converges exponentially to a limiting value q that does not depend on x and α. Therefore, we can approximate the correlation by

$$c^l_{\alpha,\alpha'}(x, x') = \frac{1}{2k+1} \sum_{\beta \in ker} f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x'))$$

where $f(c) = \frac{\sigma_w^2 \mathbb{E}[\phi(\sqrt{q}Z_1)\phi(\sqrt{q}(cZ_1 + \sqrt{1-c^2}Z_2))] + \sigma_b^2}{q}$ and Z_1, Z_2 are independent standard normal variables.

In **?**, authors studied only the limiting behaviour of correlations $c^l_{\alpha,\alpha'}(x,x)$ (same input x). These correlations describe how features are correlated for the same input, however, they do not capture the behaviour of these features for different inputs (ie $c^l_{\alpha,\alpha'}(x,x')$ where $x \neq x'$).

Before moving to the proofs, recall the definition of two classes of activation functions.

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

1. *ϕ 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. *ϕ is said to be in $\mathcal{S}$ if $\phi(0) = 0$, ϕ 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 $g_1, g_2, ..., g_n$ such that $\phi^{(2)} = \sum_{i=1}^n 1_{A_i} g_i$, where $\phi^{(2)}$ is the second derivative of ϕ.*

In the next Lemma, we provide a theoretical analysis of the limiting behaviour of the cross-correlations.

Appendix Lemma 2 (Asymptotic behaviour of the correlation in CNN with smooth activation functions)**.** *We consider a CNN with an activation function from class $\mathcal{S}$. Let $(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2$ and x, x' be two inputs. The following statements hold*

1. *If (σ_b, σ_w) are either on the Ordered or Chaotic phase, then there exists $\beta > 0$ such that*

$$\sup_{\alpha,\alpha'} |c^l_{\alpha,\alpha'}(x,x') - c| = \mathcal{O}(e^{-\beta l})$$

   where $c = 1$ if (σ_b, σ_w) is in the Ordered phase, and $c \in (0,1)$ if (σ_b, σ_w) is in the Chaotic phase.

2. *If $(\sigma_b, \sigma_w) \in EOC$, then there exists constant $\kappa, \zeta > 0$ such that for all α, α'*

$$c^l_{\alpha,\alpha'}(x,x') = 1 - \frac{\kappa}{l} + \beta \frac{\log(l)}{l^2} + O(l^{-2})$$

   where $\kappa = \frac{2}{f''(1)}$ and $\beta > 0$.

Proof. Recall that

$$c^l_{\alpha,\alpha'}(x,x') = \frac{1}{2k+1} \sum_{\beta \in ker} f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x'))$$

we write this in a matrix form

$$C_l = \frac{1}{2k+1} U f(C_{l-1})$$

where $C_l = ((c^l_{\alpha,\alpha+\beta})_{\alpha\in[|0,N-1|]})_{\beta\in[|0,N-1]}$ is a vector in $\mathbb{R}^{N^2}$, U is a convolution matrix and f is applied entry-wise. As an example, for $k = 1$, U given by

$$U = \begin{bmatrix} 1 & 1 & 0 & ... & 0 & 1 \\ 1 & 1 & 1 & 0 & \ddots & 0 \\ 0 & 1 & 1 & 1 & \ddots & 0 \\ 0 & 0 & 1 & 1 & \ddots & 0 \\ & \ddots & \ddots & \ddots & \ddots & \\ 1 & 0 & \dots & 0 & 1 & 1 \end{bmatrix}$$

U is a Circulant symmetric matrix with eigenvalues $\lambda_1 > \lambda_2 \geq \lambda_3 ... \geq \lambda^2_N$. The largest eigenvalue of U is given by $\lambda_1 = 2k + 1$ and its equivalent eigenspace is generated by the vector $e_1 = \frac{1}{N}(1, 1, ..., 1) \in \mathbb{R}^{N^2}$. This yields

$$(1 + 2k)^{-l} U^l = e_1 e_1^T + O(e^{-\beta l})$$

where $\beta = \log(\frac{\lambda_1}{\lambda_2})$

1. We deal with the Ordered phase, the proof in Chaotic phase is similar. Let (σ_b, σ_w) be in the Ordered phase and $c^l_m = \min_{\alpha,\alpha'} c^l_{\alpha,\alpha'}(x, x')$. Using the fact that f is non-decreasing, we have that $c^l_{\alpha,\alpha'}(x, x') \geq \frac{1}{2k+1} \sum_{\beta\in ker} c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x')) \geq f(c^{l-1}_m)$. Taking the min again over α, α', we have $c^l_m \geq f(c^{l-1}_m)$, therefore c^l_m is non-decreasing and converges to a stable fixed point of f. By the convexity of f, the limit is 1 (in the Chaotic phase, f has two fixed point, a stable point $c_1 < 1$ and $c_2 = 1$ unstable). Moreover, the convergence is exponential using the fact that $0 < f'(1) < 1$.

2. On the EOC, recall that we have

   $$f(x) = x + \alpha(1 - x)^2 + \zeta(1 - x)^3 + O((x - 1)^4).$$

   where $\alpha = \frac{f''(1)}{2}$. therefore, applying this entry-wise, we obtain

   $$C_l = \frac{1}{2k+1} U(C_{l-1} + \alpha(1 - C_{l-1})^2 + \zeta(1 - C_{l-1})^3 + O((1 - C_{l-1})^4))$$

Let $A_l = ((2k+1)^{-1}U)^{-l}$ and $\Lambda_l = A_l C_l$, then we have

$$\Lambda_l = \Lambda_{l-1} - \alpha A_l^{-1}\Lambda_{l-1}^2 - \zeta A_l^{-2}\Lambda_{l-1}^3 + \mathcal{O}(A_l^{-3}\Lambda_{l-1}^4)$$

Using Taylor expansion similar to the way we did with FFNN entry-wise, and using $A_l^{-m} = e_1 e_1^T + O(e^{-\beta l})$ for all $m \geq 1$ we obtain

$$\Lambda_l \sim (\alpha^{-1}l^{-1} - \alpha^{-1}\zeta\frac{\log(l)}{l^2} + O(l^{-2}))e_1$$

This yields

$$C_l = (1 - \alpha^{-1}l^{-1} + \alpha^{-1}\zeta\frac{\log(l)}{l^2})e_1 + O(l^{-2})$$

which concludes the proof.

□

We prove a similar result for ReLU-like activation functions

Appendix Lemma 3 (Asymptotic behaviour of the correlation in CNN with ReLU–like activation functions)**.** *We consider a CNN with ReLU activation. Let* $(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2$ *and* x, x' *be two inputs. The following statements hold*

1. *If* (σ_b, σ_w) *are either on the Ordered or Chaotic phase, then there exists* $\beta > 0$ *such that*

$$\sup_{\alpha,\alpha'} |c^l_{\alpha,\alpha'}(x, x') - c| = \mathcal{O}(e^{-\beta l})$$

where $c = 1$ *if* (σ_b, σ_w) *is in the Ordered phase, and* $c \in (0, 1)$ *if* (σ_b, σ_w) *is in the Chaotic phase.*

2. *If* $(\sigma_b, \sigma_w) \in EOC$*, then for all* α, α'

$$c^l_{\alpha,\alpha'}(x, x') = 1 - \frac{s}{l^2} + \Theta(l^{-2})$$

where $s = \frac{9\pi^2}{2}$

Proof. The proof is similar to the case of smooth activation functions (appendix lemma 2). The only difference is in the correlation function f. For ReLU, we have that (see Hayou et al. [2019] for more details)

$$f(x) = x + a(1-x)^{3/2} + b(1-x)^{5/2} + O((1-x)^{7/2}$$

where $s = \frac{2\sqrt{2}}{3\pi}$ and $b > 0$. using the same analysis with Taylor expansion as in the proof of appendix lemma 2, we conclude. □

B.2 Proofs

We start with the by generalizing the result of Jacot et al. [2018] on the NTK for FFNN. We give a recursive formula satisfied by the NTK of a CNN.

Proposition 1 (Infinite width dynamics of the NTK of a CNN)**.** *Let $x, x' \in \mathbb{R}^d$. Consider a CNN of the form 19. In the limit of infinite number of channels $n_1, n_2, ..., n_l \to \infty$ recursively, we have that*

$$K^1_{(i,\alpha),(i',\alpha')}(x,x') = \delta_{ii'}\Big(\frac{\sigma_w^2}{n_0(2k+1)}[x,x']_{\alpha,\alpha'} + \sigma_b^2\Big)$$

where $[x,x']_{\alpha,\alpha'} = \sum_{j,\beta} x_{j,\alpha+\beta}x_{j,\alpha'+\beta}$
and for $l > 1$, there exists a sequence $(K^l_{\alpha,\alpha'}(x,x'))_l$ such that for all i, i', α, α' $K^l_{(i,\alpha),(i',\alpha')}(x,x') = \delta_{ii'}K^l_{\alpha,\alpha'}(x,x')$. By noting $K^l_{\alpha,\alpha'} := K^l_{\alpha,\alpha'}(x,x')$ we have that

$$K^l_{\alpha,\alpha'} = \frac{1}{2k_l+1}\sum_{\beta}\Big[\dot{\Sigma}^l_{\alpha+\beta,\alpha'+\beta}K^{l-1}_{\alpha+\beta,\alpha'+\beta} + \Sigma^l_{\alpha+\beta,\alpha'+\beta}\Big]$$

where $\Sigma^l_{\alpha,\alpha'} = \sigma_b^2 + \sigma_w^2\mathbb{E}[\phi(y^{l-1}_{1,\alpha}(x))\phi(y^{l-1}_{1,\alpha'}(x'))]$ and $\dot{\Sigma}^l_{\alpha,\alpha'} = \sigma_w^2\mathbb{E}[\phi'(y^{l-1}_{1,\alpha}(x))\phi'(y^{l-1}_{1,\alpha'}(x'))]$.

Proof. Let x, x' be two inputs. We have that

$$y^1_{i,\alpha}(x) = \frac{\sigma_w}{\sqrt{v_1}}\sum_{j=1}^{n_0}\sum_{\beta\in ker_1} w^1_{i,j,\beta}x_{j,\alpha+\beta} + \sigma_b b^1_i$$
$$y^l_{i,\alpha}(x) = \frac{\sigma_w}{\sqrt{v_l}}\sum_{j=1}^{n_{l-1}}\sum_{\beta\in ker_l} w^l_{i,j,\beta}\phi(y^{l-1}_{j,\alpha+\beta}(x)) + \sigma_b b^l_i$$

therefore

$$\begin{aligned}K^1_{(i,\alpha),(i',\alpha')}(x,x') &= \sum_k(\sum_j\sum_\beta \frac{\partial y^1_{i,\alpha}(x)}{\partial w^1_{k,j,\beta}}\frac{\partial y^1_{i',\alpha'}(x)}{\partial w^1_{k,j,\beta}}) + \frac{\partial y^1_{i,\alpha}(x)}{\partial b^1_k}\frac{\partial y^1_{i',\alpha'}(x)}{\partial b^1_k}\\ &= \delta_{ii'}\Big(\frac{\sigma_w^2}{n_0(2k+1)}\sum_j\sum_\beta x_{j,\alpha+\beta}x_{j,\alpha'+\beta} + \sigma_b^2\Big)\end{aligned}$$

Assume the reslult is true for $l-1$, let us prove it for l. Let $\theta_{1:l-1}$ be model weights and bias in the layers 1 to $l-1$. Let $\partial_{\theta_{1:l-1}}y^l_{i,\alpha}(x) = \frac{\partial y^l_{i,\alpha}(x)}{\partial\theta_{1:l-1}}$. We have that

$$\partial_{\theta_{1:l-1}}y^l_{i,\alpha}(x) = \frac{\sigma_w}{\sqrt{n_{l-1}(2k+1)}}\sum_j\sum_\beta w^l_{i,j,\beta}\phi'(y^{l-1}_{j,\alpha+\beta})\partial_{\theta_{1:l-1}}y^{l-1}_{i,\alpha+\beta}(x)$$

this yields

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x)\partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T =$$
$$\frac{\sigma_w^2}{n_{l-1}(2k+1)} \sum_{j,j'} \sum_{\beta,\beta'} w^l_{i,j,\beta} w^l_{i',j',\beta'} \phi'(y^{l-1}_{j,\alpha+\beta}) \phi'(y^{l-1}_{j',\alpha'+\beta}) \partial_{\theta_{1:l-1}} y^{l-1}_{j,\alpha+\beta}(x) \partial_{\theta_{1:l-1}} y^{l-1}_{j',\alpha'+\beta}(x)^T$$

as $n_1, n_2, ..., n_{l-2} \to \infty$ and using the induction hypothesis, we have

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x)\partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T \to$$
$$\frac{\sigma_w^2}{n_{l-1}(2k+1)} \sum_{j} \sum_{\beta,\beta'} w^l_{i,j,\beta} w^l_{i',j,\beta'} \phi'(y^{l-1}_{j,\alpha+\beta}) \phi'(y^{l-1}_{j,\alpha'+\beta}) K^{l-1}_{(j,\alpha+\beta),(j,\alpha'+\beta)}(x,x')$$

note that $K^{l-1}_{(j,\alpha+\beta),(j,\alpha'+\beta)}(x,x') = K^{l-1}_{(1,\alpha+\beta),(1,\alpha'+\beta)}(x,x')$ for all j since the variables are iid across the channel index j.

Now letting $n_{l-1} \to \infty$, we have that

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x)\partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T \to$$
$$\delta_{ii'}\Big(\frac{1}{(2k+1)} \sum_{\beta,\beta'} f'(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')) K^{l-1}_{(1,\alpha+\beta),(1,\alpha'+\beta)}(x,x')\Big)$$

where $f'(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')) = \sigma_w^2 \mathbb{E}[\phi'(y^{l-1}_{j,\alpha+\beta})\phi'(y^{l-1}_{j,\alpha'+\beta})]$.
We conclude using the fact that

$$\partial_{\theta_l} y^l_{i,\alpha}(x)\partial_{\theta_l} y^l_{i',\alpha'}(x)^T \to \delta_{ii'}(\frac{\sigma_w^2}{2k+1} \sum_{\beta} \mathbb{E}[\phi(y^{l-1}_{\alpha+\beta}(x))\phi(y^{l-1}_{\alpha'+\beta}(x'))] + \sigma_b^2)$$

□

To alleviate notations, we use hereafter the notation K^L for the NTK of both FFNN and CNN. For FFNN, it represents the recursive kernel K^L given by lemma 2, whereas for CNN, it represents the recursive kernel $K^L_{\alpha,\alpha'}$ for any α, α', which means all results that follow are true for any α, α'.
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 (σ_b, σ_w) be either in the ordered or in the chaotic phase. Then, there exist $\lambda, \gamma > 0$ such that*

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

We will use the next lemma in the proof of proposition 2.

Appendix Lemma 4. *Let (a_l) be a sequence of non-negative real numbers such that $\forall l \geq 0, a_{l+1} \leq \alpha a_l + ke^{-\beta l}$, 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 γ is straightforward. □

Now we prove Proposition 2

Proof. We prove the result for FFNN first. From lemma 2, we have that

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

where $\Sigma^1(x,x') = \sigma_b^2 + \frac{\sigma_w^2}{d}x^Tx'$ 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 4. For CNN, the proof is similar using appendix lemma 2. □

Now, we show that the Initialization on the EOC leads to an invertible NTK 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. Hereafter, the notation $g(x) = \Theta(m(x))$ means there exist two constants $A, B > 0$ such that $Am(x) \leq g(x) \leq Bm(x)$.

Appendix Lemma 5. *Let $(a_l), (b_l), (\lambda_l)$ be three sequences of non-negative real numbers such that*

$$\begin{aligned} a_l &= a_{l-1}\lambda_l + b_l, \\ \lambda_l &= 1 - \frac{\alpha}{l} + \Theta(l^{-2}), \\ b_l &= q + o(l^{-1}), \end{aligned}$$

where $\alpha \in \mathbb{N}^*, q \in \mathbb{R}^+$.
Then, we have

$$\frac{a_l}{l} = \frac{q}{1+\alpha} + \Theta(l^{-1}).$$

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

$$\begin{aligned} r_l &= r_{l-1}(1 - \frac{1}{l})(1 - \frac{\alpha}{l} + \Theta(l^{-2})) + \frac{q}{l} + o(l^{-2}) \\ &= r_{l-1}(1 - \frac{1+\alpha}{l}) + \frac{q}{l} + \Theta(l^{-2}). \end{aligned}$$

Letting $x_l = r_l - \frac{q}{1+\alpha}$, there is exist two constants $M, K > 0$ such that

$$x_{l-1}(1 - \frac{1+\alpha}{l}) + \frac{K}{l^2} \leq x_l \leq x_{l-1}(1 - \frac{1+\alpha}{l}) + \frac{M}{l^2}.$$

Let us deal with the right hand inequality first. Using the recursive form, We have

$$x_l \leq x_0 \prod_{k=1}^{l}(1 - \frac{1+\alpha}{k}) + M \sum_{k=1}^{l} \prod_{j=k+1}^{l} (1 - \frac{1+\alpha}{j}) \frac{1}{k^2}.$$

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

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

For the second part, observe that

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

and

$$\frac{k!}{(k-\alpha-1)!}\frac{1}{k^2} \sim_{k\to\infty} k^{\alpha-1}.$$

Hence, we have

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

Therefore, it follows that

$$\begin{aligned}\sum_{k=1}^{l}\prod_{j=k+1}^{l}(1-\frac{1+\alpha}{j})\frac{1}{k^2} &= \frac{(l-\alpha-1)!}{l!}\sum_{k=1}^{l}\frac{k!}{(k-\alpha-1)!}\frac{1}{k^2} \\ &\sim \frac{1}{\alpha}l^{-1}.\end{aligned}$$

This proves that

$$x_l \leq \frac{M}{\alpha}l^{-1} + o(l^{-1}).$$

Using the same approach for the left-hand inequality, we prove that

$$x_l \geq \frac{K}{\alpha}l^{-1} + o(l^{-1}).$$

This concludes the proof.

□

The next lemma is a different version of the previous lemma which will be useful for other applications.

Appendix Lemma 6. *Let $(a_l), (b_l), (\lambda_l)$ be three sequences of non-negative 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

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

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

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

Let $x_l = r_l - \frac{q}{1+\alpha}$. It is clear that $\lambda_l = 1 - \alpha/l + O(l^{-3/2})$. Therefore, using appendix lemma 5 with $\beta = 1/2$, we have $r_l \to \frac{q}{1+\alpha}$. Thus, there exists $\kappa_1, \kappa_2, M, l_0 > 0$ such that for all $l \geq l_0$

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

Similarly to the proof of appendix lemma 5, it follows that

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

and

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

Recall that we have

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

and

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

so that

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

Therefore, we obtain

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

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

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

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

□

Theorem 1 (Neural Tangent Kernel on the Edge of Chaos)**.** *Let ϕ be a non-linear activation function and $(\sigma_b, \sigma_w) \in EOC$.*

1. *If ϕ is ReLU-like, then for all $x \in \mathbb{R}^d$, $\frac{K^L(x,x)}{L} = AK^{\infty}(x,x) + \Theta(L^{-1})$. Moreover, there exists $\lambda \in (0,1)$ such that*

$$\sup_{x \neq x' \in \mathbb{R}^d} \Big| \frac{K^L(x,x')}{L} - AK^{\infty}(x,x') \Big| = \Theta(L^{-1}),$$

   where $AK^{\infty}(x,x') = \frac{\sigma_w^2 \|x\| \|x'\|}{d}(1 - (1-\lambda)\mathbb{1}_{x \neq x'})$.

2. *If ϕ is in $\mathcal{S}$, then, there exists $q > 0$ such that $\frac{K^L(x,x)}{L} = AK^{\infty}(x,x) + \Theta(L^{-1}) \to q$. Moreover, there exists $\lambda \in (0,1)$ such that*

$$\sup_{x \neq x' \in \mathbb{R}^d} \Big| \frac{K^L(x,x')}{L} - AK^{\infty}(x,x') \Big| = \Theta(\log(L) L^{-1}),$$

   where $AK^{\infty}(x,x') = q(1 - (1-\lambda)\mathbb{1}_{x \neq x'})$.

Proof. We start by proving the result for FFNN, then we generalize the results to CNN.

Case 1 : FFNN

1. We use some results from Hayou et al. [2019] in this proof. Let $x, x' \in \mathbb{R}^d$ and $c^l_{x,x'} = \frac{\Sigma(x,x')}{\sqrt{\Sigma(x,x)\Sigma(x',x')}}$. Let $\gamma_l := 1 - c^l_{x,x'}$ and f be the correlation function defined by the recursive equation $c^{l+1} = f(x^l)$. 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 $\gamma_{l+1} = \gamma_l - s\gamma_l^{3/2} - \kappa\gamma_l^{5/2} + O(\gamma_l^{7/2})$ where $\kappa > 0$, this yields

$$\gamma_{l+1}^{-1/2} = \gamma_l^{-1/2} + \frac{s}{2} + \frac{\kappa}{2}\gamma_l + O(\gamma_l^2).$$

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 $\gamma_{l+1}^{-1/2} = \gamma_l^{-1/2} + \frac{s}{2} + \frac{\kappa}{2}\gamma_l + O(\gamma_l^2)$, summing again and inversing the formula, we obtain $c^l_{x,x'} = 1 - \frac{9\pi^2}{2l^2} + \Theta(l^{-3})$.
We also have

$$\begin{aligned} f'(x) &= \frac{1}{\pi}\arcsin(x) + \frac{1}{2} \\ &= 1 - \frac{\sqrt{2}}{\pi}(1-x)^{1/2} + O((1-x)^{3/2}). \end{aligned}$$

Thus, it follows that

$$f'(c^l_{x,x'}) = 1 - \frac{3}{l} + \Theta(l^{-2}).$$

Moreover, $q^l_{x,x'} = q + O(l^{-2})$ where q is the limiting variance of y^l.
Using appendix lemma 5, 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'. Therefore, we can take the supremum over $x, x' \in \mathbb{R}^d$.

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

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

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 $t \to f(1-t^2)$ 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,

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

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 $f'(c^l_{x,x'}) = 1 - \frac{2}{l} + \kappa\frac{\log(l)}{l^2} + O(l^{-2})$. We can now conclude using appendix lemma 6. Using again the argument of the boundedness of $c^1_{x,x'}$, we can take the supremum.

Case 2 : CNN

Now let us go back to the proof of the limiting behaviour of NTK for CNN on the EOC. To simplify the notation, let $K^l_{\alpha,\alpha'} = K^l_{(1,\alpha),(1,\alpha')}$. The choice of the channel 1 is not important since $K^l_{(i,\alpha),(i,\alpha')} = K^l_{(1,\alpha),(1,\alpha')}$. We do not consider the values of the NTK for $i \neq i'$ since they are always zero.

- For smooth activation function belonging to the class $\mathcal{S}$.
  Recall that on the EOC (appendix lemma 2) there exists constant $\kappa, \zeta > 0$ such that for all α, α'

$$c^l_{\alpha,\alpha'}(x,x') = 1 - \frac{\kappa}{l} + \beta\frac{\log(l)}{l^2} + O(l^{-2})$$

  Using the fact that $f'(x) = 1+(x-1)f''(1)+O((x-1)^2)$, we have $f'(c^l_{\alpha,\alpha'}(x,x')) = 1 - \frac{2}{l} + \kappa\frac{\log(l)}{l^2} + O(l^{-2})$ for all α, α'.

  Moreover, we have that $\Sigma^l_{\alpha,\alpha'}(x,x') = qf(c^l_{\alpha,\alpha'}(x,x')) = q + O(l^{-1})$ for all α, α'. Therefore, using this result with the recursive formula of the NTK from Lemma 1 we have that

$$K^l_{\alpha,\alpha'}(x,x') = \lambda_l \frac{1}{2k+1}\sum_\beta K^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x') + b_l, \tag{20}$$

  where $\lambda_l = 1 - \frac{2}{l} + \kappa\frac{\log(l)}{l^2} + O(l^{-2})$ and $b_l = q + O(l^{-1})$.

  Let $K_l = ((K^l_{\alpha,\alpha+\beta}(x,x'))_{\alpha\in[|0,N-1|]})_{\beta\in[|0,N-1]}$ is a vector in $\mathbb{R}^{N^2}$. Writing (20) in matrix form, we have

$$K_l = \lambda_l \frac{1}{2k+1} U K_{l-1} + b_l z_1,$$

  where $z_1 = (1,1,...,1) \in \mathbb{R}^{N^2}$.
  Let $A_l = ((2k+1)^{-1}U)^{-l}$ and $\Gamma_l = A_l K_l$, and using the fact that $A_l z_1 = z_1$, we have

$$\Gamma_l = \lambda_l \Gamma_{l-1} + b_l z_1.$$

  Using appendix lemma 6 element-wise, we obtain for all α, α',

$$L^{-1}\Gamma_L - \frac{q}{1+\alpha} z_1 = \Theta(\log(L)L^{-1}).$$

  We conclude using $A_L^{-1} = e_1 e_1^T + O(e^{-\beta L})$.

- For ReLU, recall that on the EOC (appendix lemma 3), for all α, α'

$$c^l_{\alpha,\alpha'}(x,x') = 1 - \frac{s}{l^2} + \Theta(l^{-2})$$

  where $s = \frac{9\pi^2}{3}$. As in the smooth activation case, we obtain the system

$$\Gamma_l = \lambda_l \Gamma_{l-1} + b_l z_1$$

with $\lambda_l = 1 - \frac{1}{l} + \Theta(l^{-2})$and $b_l = q + O(l^{-2})$ with appendix lemma 3 and the form of f' for ReLU. We conclude using appendix lemma 5.

□

B.3 Proofs for ResNets

For Residual Networks, we first prove a result on the NTK in the infinite width limit

Lemma 3 (NTK of a ResNet with Fully Connected layers in the infinite width limit)**.** *Let x, x' be two inputs and $K^{res,1}$ be the exact NTK for the Residual Network with 1 layer. Then, we have*

- *For the first layer (without residual connections), we have for all $x, x' \in \mathbb{R}^d$*

$$K^{res,1}_{ii'}(x, x') = \delta_{ii'}(\sigma_b^2 + \frac{\sigma_w^2}{d} x \cdot x'),$$

  where $x \cdot x'$ is the inner product in $\mathbb{R}^d$.

- *For $l \geq 2$, as $n_1, n_2, ..., n_{L-1} \to \infty$, we have for all $i, i' \in [|1, n_l|]$, $K^{res,l}_{ii'}(x, x') = \delta_{ii'} K^l_{res}(x, x')$, where $K^l_{res}(x, x')$ is given by the recursive formula have for all $x, x' \in \mathbb{R}^d$ and $l \geq 2$, as $n_1, n_2, ..., n_l \to \infty$ recursively, we have*

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

Proof. The first result is the same as in the FFNN case since we assume there is no residual connections between the first layer and the input. We prove the second result by induction.

- Let $x, x' \in \mathbb{R}^d$. We have

$$K^1_{res}(x, x') = \sum_j \frac{\partial y^1_1(x)}{\partial w^1_{1j}} \frac{\partial y^1_1(x)}{\partial w^1_{1j}} + \frac{\partial y^1_1(x)}{\partial b^1_1} \frac{\partial y^1_1(x)}{\partial b^1_1} = \frac{\sigma_w^2}{d} x \cdot x' + \sigma_b^2.$$

- The proof is similar to the FeedForward network NTK. For $l \geq 2$ and $i \in [|1, n_l|]$

$$\partial_{\theta_{1:l}} y^{l+1}_i(x) = \partial_{\theta_{1:l}} y^l_i(x) + \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_l} w^{l+1}_{ij} \phi'(y^l_j(x)) \partial_{\theta_{1:l}} y^l_j(x).$$

  Therefore, we obtain

$$\begin{aligned}(\partial_{\theta_{1:l}} y^{l+1}_i(x))(\partial_{\theta_{1:l}} y^{l+1}_i(x'))^t = &(\partial_{\theta_{1:l}} y^l_i(x))(\partial_{\theta_{1:l}} y^l_i(x'))^t \\ &+ \frac{\sigma_w^2}{n_l} \sum_{j,j'}^{n_l} w^{l+1}_{ij} w^{l+1}_{ij'} \phi'(y^l_j(x)) \phi'(y^l_{j'}(x')) \partial_{\theta_{1:l}} y^l_j(x) (\partial_{\theta_{1:l}} y^l_{j'}(x'))^t + I\end{aligned}$$

where

$$I = \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_l} w_{ij}^{l+1} (\phi'(y_j^l(x)) \partial_{\theta_{1:l}} y_i^l(x) (\partial_{\theta_{1:l}} y_j^l(x'))^t + \phi'(y_j^l(x')) \partial_{\theta_{1:l}} y_j^l(x) (\partial_{\theta_{1:l}} y_i^l(x'))^t).$$

Using the induction hypothesis, as $n_0, n_1, ..., n_{l-1} \to \infty$, we have that

$$(\partial_{\theta_{1:l}} y_i^{l+1}(x))(\partial_{\theta_{1:l}} y_i^{l+1}(x'))^t + \frac{\sigma_w^2}{n_l} \sum_{j,j'}^{n_l} w_{ij}^{l+1} w_{ij'}^{l+1} \phi'(y_j^l(x)) \phi'(y_{j'}^l(x')) \partial_{\theta_{1:l}} y_j^l(x) (\partial_{\theta_{1:l}} y_{j'}^l(x'))^t + I$$

$$\to K_{res}^l(x, x') + \frac{\sigma_w^2}{n_l} \sum_{j}^{n_l} (w_{ij}^{l+1})^2 \phi'(y_j^l(x)) \phi'(y_j^l(x')) K_{res}^l(x, x') + I',$$

where $I' = \frac{\sigma_w^2}{n_l} w_{ii}^{l+1} (\phi'(y_i^l(x)) + \phi'(y_i^l(x'))) K_{res}^l(x, x')$.

As $n_l \to \infty$, we have that $I' \to 0$. Using the law of large numbers, as $n_l \to \infty$

$$\frac{\sigma_w^2}{n_l} \sum_{j}^{n_l} (w_{ij}^{l+1})^2 \phi'(y_j^l(x)) \phi'(y_j^l(x')) K_{res}^l(x, x') \to \dot{\Sigma}^{l+1}(x, x') K_{res}^l(x, x').$$

Moreover, we have that

$$(\partial_{w^{l+1}} y_i^{l+1}(x))(\partial_{w^{l+1}} y_i^{l+1}(x'))^t + (\partial_{b^{l+1}} y_i^{l+1}(x))(\partial_{b^{l+1}} y_i^{l+1}(x'))^t = \frac{\sigma_w^2}{n_l} \sum_j \phi(y_j^l(x)) \phi(y_j^l(x')) + \sigma_b^2$$

$$\underset{n_l \to \infty}{\to} \sigma_w^2 \mathbb{E}[\phi(y_i^{l+1}(x)) \phi(y_i^{l+1}(x'))] + \sigma_b^2 = \Sigma^{l+1}(x, x').$$

□

Now we proof the recursive formula for ResNets with Convolutional layers.

Lemma 4 (NTK of a ResNet with Convolutional layers in the infinite width limit)**.** *Let x, x' be two inputs and $K^{res,1}$ be the exact NTK for the Residual Network with 1 layer. Then, we have*

- *For the first layer (without residual connections), we have for all $x, x' \in \mathbb{R}^d$*

$$K_{(i,\alpha),(i',\alpha')}^{1,res}(x, x') = \delta_{ii'} \Big(\frac{\sigma_w^2}{n_0(2k+1)} [x, x']_{\alpha,\alpha'} + \sigma_b^2 \Big),$$

where $[x, x']_{\alpha,\alpha'} = \sum_j \sum_\beta x_{j,\alpha+\beta} x_{j,\alpha'+\beta}$.

- *For $l \geq 2$, as $n_1, n_2, ..., n_{L-1} \to \infty$, we have for all $i, i' \in [|1, n_l|]$, $K^{res,l}_{(i,\alpha),(i',\alpha')}(x, x') = \delta_{ii'} K^{res,l}_{\alpha,\alpha'}(x, x')$, where $K^{res,l}_{\alpha,\alpha'}$ is given by the recursive formula*

$$K^{res,l}_{\alpha,\alpha'} = K^{res,l-1}_{\alpha,\alpha'} + \frac{1}{2k_l+1} \sum_{\beta} \left[\dot{\Sigma}^l_{\alpha+\beta,\alpha'+\beta} K^{l-1}_{\alpha+\beta,\alpha'+\beta} + \Sigma^l_{\alpha+\beta,\alpha'+\beta} \right]$$

where $\Sigma^l_{\alpha,\alpha'} = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y^{l-1}_{1,\alpha}(x))\phi(y^{l-1}_{1,\alpha'}(x'))]$ and $\dot{\Sigma}^l_{\alpha,\alpha'} = \sigma_w^2 \mathbb{E}[\phi'(y^{l-1}_{1,\alpha}(x))\phi'(y^{l-1}_{1,\alpha'}(x'))]$.

Proof. Let x, x' be two inputs. We have that

$$\begin{aligned} K^1_{(i,\alpha),(i',\alpha')}(x, x') &= \sum_j \Big(\sum_\beta \frac{\partial y^1_{i,\alpha}(x)}{\partial w^1_{i,j,\beta}} \frac{\partial y^1_{i',\alpha'}(x)}{\partial w^1_{i',j,\beta}} + \frac{\partial y^1_{i,\alpha}(x)}{\partial b^1_j} \frac{\partial y^1_{i',\alpha'}(x)}{\partial b^1_j} \Big) \\ &= \delta_{ii'} \Big(\frac{\sigma_w^2}{n_0(2k+1)} \sum_j \sum_\beta x_{j,\alpha+\beta} x_{j,\alpha'+\beta} + \sigma_b^2 \Big). \end{aligned}$$

Assume the result is true for $l-1$, let us prove it for l. Let $\theta_{1:l-1}$ be model weights and bias in the layers 1 to $l-1$. Let $\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x) = \frac{\partial y^l_{i,\alpha}(x)}{\partial \theta_{1:l-1}}$. We have that

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x) = \partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha}(x) + \frac{\sigma_w}{\sqrt{n_{l-1}(2k+1)}} \sum_j \sum_\beta w^l_{i,j,\beta} \phi'(y^{l-1}_{j,\alpha+\beta}) \partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha+\beta}(x)$$

this yields

$$\begin{aligned} &\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x) \partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T = \partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha}(x) \partial_{\theta_{1:l-1}} y^{l-1}_{i',\alpha'}(x)^T + \\ &\frac{\sigma_w^2}{n_{l-1}(2k+1)} \sum_{j,j'} \sum_{\beta,\beta'} w^l_{i,j,\beta} w^l_{i',j',\beta'} \phi'(y^{l-1}_{j,\alpha+\beta}) \phi'(y^{l-1}_{j',\alpha'+\beta}) \partial_{\theta_{1:l-1}} y^{l-1}_{j,\alpha+\beta}(x) \partial_{\theta_{1:l-1}} y^{l-1}_{j',\alpha'+\beta}(x)^T + I, \end{aligned}$$

where

$$I = \frac{\sigma_w}{\sqrt{n_{l-1}(2k+1)}} \sum_{j,\beta} w^l_{i,j,\beta} \phi'(y^{l-1}_{j,\alpha+\beta}) (\partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha}(x) \partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha+\beta}(x)^T + \partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha+\beta}(x) \partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha}(x)^T).$$

As $n_1, n_2, ..., n_{l-2} \to \infty$ and using the induction hypothesis, we have

$$\begin{aligned} &\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x) \partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T \to \delta_{ii'} K^{l-1}_{\alpha,\alpha'}(x, x') + \\ &\frac{\sigma_w^2}{n_{l-1}(2k+1)} \sum_j \sum_{\beta,\beta'} w^l_{i,j,\beta} w^l_{i',j,\beta'} \phi'(y^{l-1}_{j,\alpha+\beta}) \phi'(y^{l-1}_{j,\alpha'+\beta}) K^{l-1}_{(j,\alpha+\beta),(j,\alpha'+\beta)}(x, x'). \end{aligned}$$

Note that $K^{l-1}_{(j,\alpha+\beta),(j,\alpha'+\beta)}(x,x') = K^{l-1}_{(1,\alpha+\beta),(1,\alpha'+\beta)}(x,x')$ for all j since the variables are iid across the channel index j. Now letting $n_{l-1} \to \infty$, we have that

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x) \partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T \to$$
$$\delta_{ii'} K^{l-1}_{\alpha,\alpha'}(x,x') + \delta_{ii'}\Big(\frac{1}{(2k+1)} \sum_{\beta,\beta'} f'(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')) K^{l-1}_{(1,\alpha+\beta),(1,\alpha'+\beta)}(x,x')\Big),$$

where $f'(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')) = \sigma_w^2 \mathbb{E}[\phi'(y^{l-1}_{j,\alpha+\beta})\phi'(y^{l-1}_{j,\alpha'+\beta})]$.
We conclude using the fact that

$$\partial_{\theta_l} y^l_{i,\alpha}(x) \partial_{\theta_l} y^l_{i',\alpha'}(x)^T \to \delta_{ii'}(\frac{\sigma_w^2}{2k+1} \sum_{\beta} \mathbb{E}[\phi(y^{l-1}_{\alpha+\beta}(x))\phi(y^{l-1}_{\alpha'+\beta}(x'))] + \sigma_b^2).$$

□

Before moving to the main proposition on ResNets, We first prove a Lemma on the asymptotic behaviour of c^l for ResNet.

Appendix Lemma 7 (Asymptotic expansion of c^l for ResNet)**.** *Let $x \neq x' \in \mathbb{R}^d$. We have for FFNN*

$$c^l(x,x') = 1 - \delta l^{-2} + \zeta l^{-3} + o(l^{-3}),$$

and for CNN

$$\forall \alpha, \alpha', c^l_{\alpha,\alpha'}(x,x') = 1 - \delta l^{-2} + \zeta_{\alpha,\alpha'} l^{-3} + o(l^{-3}),$$

where $\delta = \frac{9\pi^2(1+\frac{\sigma_w^2}{2})^2}{2(\frac{\sigma_w^2}{2})^2}$ and $\zeta, \zeta_{\alpha,\alpha'} > 0$.

Proof. We first prove the result for FFNN, then generalize it to CNN.

- Let $x \neq x'$ be two inputs and $c^l := c^l(x,x')$. Using the fact that $\Sigma^l(x,x) = \Sigma^{l-1}(x,x) + \sigma_w^2/2\Sigma^{l-1}(x,x) = (1+\sigma_w^2/2)^{l-1}\Sigma^1(x,x)$, we have

  $$c^{l+1} = \frac{1}{1+\alpha} c^l + \frac{\alpha}{1+\alpha} f(c^l),$$

  where $f(x) = 2\mathbb{E}[(Z_1)_+(xZ_1 + \sqrt{1-x^2}Z_2)_+]$ and $\alpha = \frac{\sigma_w^2}{2}$. It was shown in Hayou et al. [2019] that

  $$f(x) = \frac{1}{\pi} x \arcsin(x) + \frac{1}{\pi}\sqrt{1-x^2} + \frac{1}{2}x.$$

Using a Taylor expansion near 1 yields

$$f(x) \underset{x\to 1-}{=} x + \frac{2\sqrt{2}}{3\pi}(1-x)^{3/2} + \kappa(1-x)^{5/2} + \mathcal{O}((1-x)^{7/2}) \tag{21}$$

where $\kappa > 0$. We obtain

$$c^{l+1} = c^l + s(1-c^l)^{3/2} + \kappa'(1-c^l)^{5/2} + \mathcal{O}((1-c^l)^{7/2}),$$

where $s = \frac{2\lambda\sqrt{2}}{3(1+\alpha)\pi}$ and $\kappa' > 0$. Letting $\gamma_l = 1 - c^l$, we have

$$\gamma_{l+1} = \gamma_l - s\gamma_l^{3/2} - \kappa'\gamma_l^{5/2} + O(\gamma_l^{7/5}).$$

which leads to

$$\gamma_{l+1}^{-1/2} = \gamma_l^{-1/2} + \frac{s}{2} + \frac{\kappa'}{2}\gamma_l + O(\gamma_l^2). \tag{22}$$

Therefore one has

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

Summing again in equation (27), we obtain a more precise expansion

$$\gamma_l^{-1/2} = \frac{s}{2}l + \beta + o(1)$$

where $\beta > 0$ because all coefficients of Taylor of Expansion of f near 1 are positive (which is easy to see recursively). Therefore, one has

$$\gamma_l = \frac{4}{s^2}l^{-2}(1 - \beta l^{-1} + o(l^{-1})).$$

This concludes the proof for FFNN.

- For CNN, let $q_\alpha^l(x) = q_{\alpha,\alpha}^l(x,x)$. We first prove that for all x, there exists $\beta > 0$ such that for all $x \in \mathbb{R}^d$ and α,

$$q_\alpha^l(x) = (1 + \frac{\sigma_w^2}{2})^l q_{0,x} + \mathcal{O}((1 + \frac{\sigma_w^2}{2})^l e^{-\beta l}))$$

where $q_{0,x}$ is a constant that depends on x.

To see this, recall that

$$q_\alpha^l(x) = q_\alpha^{l-1}(x) + \frac{\sigma_w^2}{2k+1}\sum_{\beta\in ker}\frac{q_{\alpha+\beta}^{l-1}(x)}{2}.$$

We write this expression in a matrix form

$$A_l = (I + \frac{\sigma_w^2}{2(2k+1)}U)A_{l-1},$$

where $A_l = (q_\alpha^l(x))_\alpha$ is a vector in $\mathbb{R}^N$ and U is defined in the proof of appendix lemma 2. Let $\delta = \frac{\sigma_w^2}{2(2k+1)}$. The largest eigenvalue of $I + \frac{\sigma_w^2}{2(2k+1)}U$ is given by $\lambda_1 = 1 + (2k+1)\delta = 1 + \frac{\sigma_w^2}{2}$ and its equivalent eigenspace is generated by the vector $e_1 = (1, 1, ..., 1) \in \mathbb{R}^N$. This yields

$$(1 + \frac{\sigma_w^2}{2})^{-l}(I + \frac{\sigma_w^2}{2(2k+1)}U)^l = e_1 e_1^T + O(e^{-\beta l})$$

where $\beta = \log(\frac{\lambda_1}{\lambda_2})$. It follows that

$$\lambda_1^{-l} A_l = (\lambda_1^{-l}(I + \frac{\sigma_w^2}{2(2k+1)}U)^l)A_0 = e_1 e_1^T A_0 + O(e^{-\beta l}).$$

This shows that the quantities $q_\alpha^l(x)$ become similar (at an exponential rate) as l grows. With this result, let us first prove the result of lemma **??**to fix assuming $q_\alpha^0(x) = q_1^0(x)$ for all α and x. Using the recursive formula of $q_\alpha^l(x)$, we obtain $q_\alpha^l(x) = q_1^l(x) = (1 + \frac{\sigma_w^2}{2})^l q_{0,x}$ for all l, α, x. Now recall that

$$q_{\alpha,\alpha'}^l(x, x') = q_{\alpha,\alpha'}^{l-1}(x, x') + \frac{\sigma_w^2}{2k+1} \sum_{\beta \in ker} \mathbb{E}[\phi(y_{1,\alpha+\beta}^{l-1}(x))\phi(y_{1,\alpha'+\beta}^{l-1}(x'))].$$

To simplify notation, write $c_{\alpha,\alpha'}^l = c_{\alpha,\alpha'}^l(x, x')$ and $\alpha = \frac{\sigma_w^2}{2}$. This yields

$$c_{\alpha,\alpha'}^l = \frac{1}{1+\alpha} c_{\alpha,\alpha'}^{l-1} + \frac{\alpha}{(1+\alpha)(2k+1)} \sum_{\beta \in ker} f(c_{\alpha+\beta,\alpha'+\beta}^{l-1}).$$

In a matrix form, we have

$$C_l = \frac{1}{1+\alpha} C_{l-1} + \frac{\alpha}{(1+\alpha)(2k+1)} U f(C_{l-1}),$$

where $C_l = ((c_{\alpha,\alpha+\beta}^l)_{\alpha \in [|0,N-1|]})_{\beta \in [|0,N-1]}$ is a vector in $\mathbb{R}^{N^2}$, U is the convolution matrix introduced in the proof of appendix lemma 2 and f is applied element-wise. Recall the following Taylor expansion of f

$$f(x) \underset{x \to 1-}{=} x + \frac{2\sqrt{2}}{3\pi}(1-x)^{3/2} + \kappa(1-x)^{5/2} + \mathcal{O}((1-x)^{7/2}). \tag{23}$$

This yields

$$C_l = (\frac{1}{1+\alpha}I + \frac{\alpha}{(1+\alpha)(2k+1)}U)C_{l-1}$$
$$+ \frac{\alpha}{(1+\alpha)(2k+1)}U(\frac{2\sqrt{2}}{3\pi}(1-C_{l-1})^{3/2} + \kappa(1-C_{l-1})^{5/2} + \mathcal{O}((1-C_{l-1})^{7/2}).$$

As in the in the proof of appendix lemma 2, letting $A_l = (\frac{1}{1+\alpha}I + \frac{\alpha}{(1+\alpha)(2k+1)}U)^{-l}$ and $\Gamma_l = 1 - A_l C_l$ yields

$$\Gamma_l = \Gamma_{l-1} - E_l\Gamma_{l-1}^{3/2} - F_l\Gamma_{l-1}^{5/2} + O(G_l\Gamma_{l-1}^{7/2})),$$

where $E_l = \frac{2\sqrt{2}\alpha}{3\pi(1+\alpha)(2k+1)}A_l U A_{l-1}^{-3/2}$, $F_l = \kappa\frac{\alpha}{(1+\alpha)(2k+1)}A^l U A_{l-1}^{-5/2}$ and $G_l = \frac{\alpha}{(1+\alpha)(2k+1)}A^l U A_{l-1}^{-7/2}$. Since $A_l^{-1} = e_1 e_1^T + O(e^{-\beta l})$, we have that $F_l = \gamma(e_1 e_1^T + O(e^{-\beta l}))$ and $G_l = \rho(e_1 e_1^T + O(e^{-\beta l}))$ where $\gamma, \rho \in \mathbb{R}$ are positive constants.

Using another Taylor expansion, we obtain

$$\Gamma_l^{-1/2} = \Gamma_{l-1}^{-1/2} + \frac{1}{2}E_l e_1 + \frac{1}{2}F_l\Gamma_{l-1} + O(G_l\Gamma_{l-1}^2).$$

As e_1 is an eigenvector of U with eigenvalue $2k+1$, we have $E_l e_1 = \frac{2\sqrt{2}\alpha}{3\pi(1+\alpha)}e_1$, therefore

$$\Gamma_l^{-1/2} \sim \frac{\sqrt{2}\alpha}{3\pi(1+\alpha)}l e_1.$$

Summing again in the Taylor expansion, we get

$$\Gamma_l^{-1/2} = (\frac{\sqrt{2}\alpha}{3\pi(1+\alpha)}l + Z + o(1))e_1,$$

where Z is a constant vector of the same size as C_l with positive entries. We conclude the proof knowing that $A_l^{-1} = e_1 e_1^T + O(e^{-\beta l})$.

□

The next theorem shows that no matter what the choice of $\sigma_w > 0$, the scaled NTK of a ResNet will always have a subexponential convergence rate to a limiting AK_{res}^{∞}. We say that ResNet 'live' on the Edge of Chaos.

Theorem 2 (NTK for ResNet)**.** *Consider a Residual Neural Network with the following forward propagation equations*

$$y^l(x) = y^{l-1}(x) + \mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2. \tag{24}$$

where $\mathcal{F}$ is either a convolutional or dense layer (equations 18 and 19) and ϕ is the ReLU activation function. Let K^L_{res} be the corresponding NTK. Then for all $x \in \mathbb{R}^d$, $\frac{K^L_{res}(x,x)}{\alpha_L} = AK^{\infty}_{res}(x,x) + \Theta(L^{-1})$ and there exists $\lambda \in (0,1)$ such that

$$\sup_{x \neq x' \in \mathbb{R}^d} \Big| \frac{K^L_{res}(x,x')}{\alpha_L} - \frac{||x|| \times ||x'||}{d}\lambda \Big| = \Theta(L^{-1}),$$

where $AK^{\infty}_{res}(x,x') = \frac{\sigma_w^2 \|x\| \|x'\|}{d}(1 - (1-\lambda)\mathbb{1}_{x \neq x'})$, and $\alpha_L = L(1 + \frac{\sigma_w^2}{2})^{L-1}$.

Proof. **Case 1 : FFNN**
We first prove the result for $K^L_{res}(x,x)$ then $K^L_{res}(x,x')$.

- We have that $\dot{\Sigma}^l(x,x) = \frac{\sigma_w^2}{2} f(1) = \frac{\sigma_w^2}{2}$. Moreover, we have $\Sigma^l(x,x) = \Sigma^{l-1}(x,x) + \sigma_w^2/2\Sigma^{l-1}(x,x) = (1 + \sigma_w^2/2)^{l-1}\frac{\sigma_w^2}{d}||d||$.
- Let $x, x' \in \mathbb{R}^d$. Recall that (Lemma 3)

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

Let $\alpha = \frac{\sigma_w^2}{2}$. From appendix lemma 7 we have that

$$c^l(x,x') = 1 - \delta l^{-2} + \zeta l^{-3} + o(l^{-3})$$

$\delta = \frac{9\pi^2(1+\alpha)^2}{2(\alpha)^2}$ and $\zeta > 0$.
We also have $\dot{\Sigma}^l(x,x') = \alpha f'(c^l(x,x'))$ where $f(x) = \frac{1}{\pi}x\arcsin(x) + \frac{1}{\pi}\sqrt{1-x^2} + \frac{1}{2}x$. As in the proof of Theorem 1, we have that

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

it follows that

$$f'(c^l(x,x')) = 1 - \frac{\sqrt{2}}{\pi}\delta^{1/2}l^{-1} + \frac{\delta^{-1/2}\zeta}{2}l^{-2} + O(l^{-3})$$

and we obtain

$$1 + \dot{\Sigma}^l(x, x') = (1 + \alpha)(1 - 3l^{-1} + \kappa l^{-2} + O(l^{-3}))$$

Now let $a_l = \frac{K^l_{res}(x,x')}{(1+\alpha)^{l-1}}$. Using the recursive formula of the NTK, we obtain

$$a_l = \lambda_l a_{l-1} + b_l$$

where $\lambda_l = 1 - 3l^{-1} + \kappa l^{-2} + O(l^{-3})$, $b_l = \sqrt{\Sigma^1(x,x)}\sqrt{\Sigma^1(x,x)}f(c^l(x,x')) = q + O(l^{-2})$ with $q = \sqrt{\Sigma^1(x,x)}\sqrt{\Sigma^1(x,x)}$ and where we used the fact that $c^l = 1 + O(l^{-2})$. We conclude using appendix lemma 5.

Case 2 : CNN

Let x, x' be two inputs. Using lemma 4, we have that for all α, α'

$$K^{res,l}_{\alpha,\alpha'} = K^{res,l-1}_{\alpha,\alpha'} + \frac{1}{2k+1}\sum_{\beta}\left[\dot{\Sigma}^l_{\alpha+\beta,\alpha'+\beta}K^{l-1}_{\alpha+\beta,\alpha'+\beta} + \Sigma^l_{\alpha+\beta,\alpha'+\beta}\right]$$

where we have $\frac{1}{2k+1}\sum_\beta \Sigma^l_{\alpha+\beta,\alpha'+\beta} = (1+\alpha)^{l-1}(q + O(l^{-2}))$ by appendix lemma 7, where $q > 0$ is a constant.

Writing this with Hadamard product

$$K^{res,l} = (1 + \alpha f'(C_{l-1})) \circ K^{res,l-1} + (1+\alpha)^{l-1}(q + O(l^{-2}))e_1$$

Letting $\Theta_l = K^{res,l}/(1+\alpha)^{l-1}$, we have that

$$\Theta_l = \frac{1 + \alpha f'(C_{l-1})}{1+\alpha} \circ \Theta_{l-1} + (q + O(l^{-2}))e_1$$

we apply appendix lemma 7 to get

$$\frac{1 + \alpha f'(C_{l-1})}{1+\alpha} = (1 - 3l^{-1})e_1 + \zeta l^{-2} + O(l^{-3}))$$

where $\zeta = (\zeta_{\alpha,\alpha'})$. from here the proof is similar to the FFNN case, we apply appendix lemma 5 elementwise which concludes the proof for CNN.

□

Now let prove the the Scaled Resnet result. Before that, we prove the following Lemma

Appendix Lemma 8. *Consider a Residual Neural Network with the following forward propagation equations*

$$y^l(x) = y^{l-1}(x) + \frac{1}{\sqrt{l}}\mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2. \tag{25}$$

where $\mathcal{F}$ is either a convolutional or dense layer (equations 18 and 19) with ReLU activation. Let $c^l(x, x')$ be either the correlation between $y_i^l(x)$ and $y_i^l(x')$ for dense layers (choice of i is not important since they're iid) or the correlation between $y_{i,\alpha}^l(x)$ and $y_{i,\alpha'}^l(x')$ for convolutional layers for some α, α'. Then there exists $\kappa, \zeta > 0$ such that

$$1 - c^l = \frac{\kappa}{\log(l)^2} - \frac{\zeta}{\log(l)^3} + o(\frac{1}{\log(l)^3})$$

Proof. We first start with the dense layer case. Let x, x' be two inputs and denote by $c^l = c^l(x, x')$. With the same approach as in the proof of appendix lemma 7,we have that

$$c^l = \frac{1}{1+\alpha_l}c^{l-1} + \frac{\alpha_l}{1+\alpha_l}f(c^{l-1})$$

where $\alpha_l = \frac{\sigma_w^2}{2l}$. Since f is non-decreasing, we have that $c^l \geq c^{l-1}$, therefore c^l converges to a fixed point c. Let us prove that $c = 1$. By contradiction, suppose $c < 1$ so that $f(c) - c > 0$. This yields

$$c^l - c = c^{l-1} - c + \frac{f(c) - c}{l} + O(\frac{c^l - c}{l}) + O(l^{-2})$$

by summing, this leads to $c^l - c \sim (f(c) - c)\log(l)$ which is absurd. We conclude that $c = 1$.

Now let us find the asymptotic expansion of $1 - c^l$. Using a Taylor expansion of f near 1 yields

$$f(x) \underset{x \to 1-}{=} x + \frac{2\sqrt{2}}{3\pi}(1-x)^{3/2} + \kappa(1-x)^{5/2} + \mathcal{O}((1-x)^{7/2}) \tag{26}$$

where $\kappa > 0$. We obtain

$$c^l = c^{l-1} + \delta_l(s(1-c^{l-1})^{3/2} + \kappa'(1-c^{l-1})^{5/2} + \mathcal{O}((1-c^{l-1})^{7/2})),$$

where $s = \frac{2\sqrt{2}}{3\pi}$ and $\kappa' > 0$ and $\delta_l = \frac{\alpha_l}{1+\alpha_l}$. Letting $\gamma_l = 1 - c^l$, we have

$$\gamma_l = \gamma_{l-1} - s\delta_l\gamma_{l-1}^{3/2} - \kappa'\delta_l\gamma_{l-1}^{5/2} + O(\delta_l\gamma_{l-1}^{7/5}).$$

which leads to

$$\gamma_l^{-1/2} = \gamma_{l-1}^{-1/2} + \frac{s}{2}\delta_l + \frac{\kappa'}{2}\delta_l\gamma_{l-1} + O(\delta_l\gamma_{l-1}^2). \tag{27}$$

therefore, we have that

$$\gamma_l^{-1/2} \sim \frac{s\sigma_w^2}{4}\log(l)$$

and $1 - c^l \sim \frac{\kappa}{\log(l)^2}$ where $\kappa = 16/s^2\sigma_w^4$.

we can further expand the asymptotic approximation using the same approach as in the proof of appendix lemma 7, we have that

$$1 - c^l = \frac{\kappa}{\log(l)^2} - \frac{\zeta}{\log(l)^3} + o(\frac{1}{\log(l)^3})$$

the proof with convolutional layers is similar also to the proof of appendix lemma 7.

□

Proposition 3 (Scaled Resnet)**.** *Consider a Residual Neural Network with the following forward propagation equations*

$$y^l(x) = y^{l-1}(x) + \frac{1}{\sqrt{l}}\mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2. \tag{28}$$

where $\mathcal{F}$ is either a convolutional or dense layer (equations 18 and 19) with ReLU activation. Then the scaling factor α_L in Theorem 2 becomes $\alpha_L = L^{1+\sigma_w^2/2}$ and the convergence rate is $\Theta(\log(L)^{-1})$.

Proof. We use the same calculus as in the non scaled case. Let us prove the result for dense layers, the proof for convolutional layers follows the same analysis. We first prove the result for $K_{res}^L(x,x)$ then $K_{res}^L(x,x')$.

- We have that $\dot{\Sigma}^l(x,x) = \frac{\sigma_w^2}{2l}f(1) = \frac{\sigma_w^2}{2l}$. Moreover, we have $\Sigma^l(x,x) = \Sigma^{l-1}(x,x) + \sigma_w^2/2l \times \Sigma^{l-1}(x,x) = [\prod_{k=1}^l(1+\sigma_w^2/2k)]\frac{\sigma_w^2}{d}\|x\|^2$. Recall that

$$K_{res}^l(x,x) = K_{res}^{l-1}(x,x)(1 + \frac{\sigma_w^2}{2l}) + \Sigma^l(x,x)$$

letting $k'_l = \frac{K^l_{res}(x,x')}{\prod_{k=1}^{l}(1+\sigma_w^2/2k)}$,we have that

$$k'_l = k'_{l-1} + \frac{\sigma_w^2}{d}\|x\|$$

using the fact that $\prod_{k=1}^{l}(1+\sigma_w^2/2k) \sim l^{\sigma_w^2/2}$, we conclude for $K^l_{res}(x,x')$.

- Let $x, x' \in \mathbb{R}^d$. Recall that

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

From appendix lemma 8 we have that

$$1 - c^l = \frac{\kappa}{\log(l)^2} - \frac{\zeta}{\log(l)^3} + o(\frac{1}{\log(l)^3})$$

$\kappa = \frac{16}{s^2\sigma_w^4}$ and $\zeta > 0$. Using the expression of f' from the proof of Theorem 2, it follows that

$$f'(c^l(x,x')) = 1 - \frac{6}{\sigma_w^2}\log(l)^{-1} + \zeta'\log(l)^{-2} + O(\log(l)^{-3})$$

and we obtain

$$1 + \dot{\Sigma}^l(x,x') = 1 + \frac{\sigma_w^2}{2l} - 3l^{-1}\log(l)^{-1} + \zeta'' l^{-1}\log(l)^{-2} + O(l^{-1}\log(l)^{-3})$$

Letting $a_l = \frac{K^l_{res}(x,x')}{\prod_{k=1}^{l}(1+\sigma_w^2/2k)}$, we obtain

$$a_l = \lambda_l a_{l-1} + b_l$$

where $\lambda_l = 1 - l^{-1} - 3l^{-1}\log(l)^{-1} + O(l^{-1}\log(l)^{-2})$, $b_l = \sqrt{\Sigma^1(x,x)}\sqrt{\Sigma^1(x,x)}f(c^l(x,x')) = q + O(\log(l)^{-2})$ with $q = \sqrt{\Sigma^1(x,x)}\sqrt{\Sigma^1(x,x)}$ and where we used the fact that $c^l = 1 + O(\log(l)^{-2})$.

Now we proceed in the same way as in the proof of appendix lemma 6. Let $x_l = \frac{a_l}{l} - q$, then there exists $A, B > 0$ such that

$$x_{l-1}(1 - \frac{1}{l}) - Al^{-1}\log(l)^{-1} \leq x_l \leq x_{l-1}(1 - \frac{1}{l}) - Bl^{-1}\log(l)^{-1}$$

therefore, there exists l_0 such that

$$x_l \leq x_{l_0} \prod_{k=l_0}^{l} (1 - \frac{1}{k}) - B \sum_{k=l_0}^{l} \prod_{j=k+1}^{l} (1 - \frac{1}{j}) k^{-1} \log(k)^{-1}$$

and

$$x_l \geq x_{l_0} \prod_{k=l_0}^{l} (1 - \frac{1}{k}) - A \sum_{k=l_0}^{l} \prod_{j=k+1}^{l} (1 - \frac{1}{j}) k^{-1} \log(k)^{-1}$$

after simplification, we have that

$$\sum_{k=l_0}^{l} \prod_{j=k+1}^{l} (1 - \frac{1}{j}) k^{-1} \sim \frac{1}{l} \int^{l} \frac{1}{\log(t)} dt = \log(l)^{-1}$$

where we have used the asymptotic approximation of the Logarithmic Intergal function $\mathrm{Li}(x) = \int^t \frac{1}{\log(t)} \sim_{x \to \infty} \frac{x}{\log(x)}$

we conclude that $\alpha_L = L \times \prod_{k=1}^{l} (1 + \sigma_w^2 / 2k) \sim L^{1 + \frac{\sigma_w^2}{2}}$ and the convergence rate of the NTK is now $\Theta(\log(L)^{-1})$ which is better than $\Theta(L^{-1})$.

In the limit of large L, the matrix NTK of the scaled resnet has the following form

$$\hat{K}_{res}^{l} = qU + \log(L)^{-1} \Theta(M_L)$$

where U is the matrix of ones, and M_L has all elements but the diagonal equal to 1 and the diagonal has $L^{-1} \log(L) \to 0$. Therefore, M_L is inversible for large L which makes $\hat{K}_{res}^{l}$ also inversible. Moreover, observe that the convergence rate for scaled resnet is $\log(L)^{-1}$ which means that for the same depth L, the NTK remains far more expressive for scaled resnet compared to standard resnet, this is particularly important for the generalization.

□

C Training with SGD instead of GD

In this section, we extend the results of the NTK to the case of SGD. 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 Ω is the probability space, $\mathbb{P}$ is a probability measure on Ω, and $\mathcal{F}$ is the natural filtration of the Brownian motion. Under

boundedness conditions, 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

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta \nabla_\theta \mathcal{L}^{(S)}(\hat{\theta}_t), \tag{29}$$

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 θ

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

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

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

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

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

where $\operatorname{Cov}(\cdot)$ 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 $\Sigma(\theta) = \operatorname{Cov}\left(\nabla_\theta \ell(f(X_1, \theta), Y_1)\right)$ and we have, neglecting the term $o_{P_0}(S^{-1/2})$,

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta \nabla_\theta \mathcal{L}(\hat{\theta}_t) + \frac{\eta}{\sqrt{S}} Z(\theta_t). \tag{30}$$

We can in particular bound the difference between (30) and the continuous time SDE approximation (see also Hu et al. [2018] and Li et al. [2017])

$$d\theta_t = -\nabla_\theta \mathcal{L}(\theta_t) dt + \sqrt{\frac{\eta}{S}} \Sigma(\theta_t)^{\frac{1}{2}} dW_t. \tag{31}$$

SGD updates can therefore be seen as a discretization of the previous SDE with time step $\Delta = \eta$, and where $\Sigma(\theta_t)^{\frac{1}{2}}$ is the square-root matrix of $\Sigma(\theta_t) = \operatorname{Cov}\left(\nabla_\theta \ell(f(X_1, \theta_t), Y_1)\right)$ and $(W_t)_{t \geq 0}$ a standard Brownian motion.

Since the dynamics of θ_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.

Proposition 4. *Under the dynamics of the SDE* (31)*, the vector* $f_t(\mathcal{X})$ *is the solution of the following SDE*

$$df_t(\mathcal{X}) = [-\frac{1}{N}K^L_{\theta_t}(\mathcal{X},\mathcal{X})\nabla_z\ell(f_t(\mathcal{X}),Y) + \frac{1}{2}\frac{\eta}{S}\Gamma_t(\mathcal{X})]dt + \sqrt{\frac{\eta}{S}}\nabla_\theta f(\mathcal{X},\theta_t)\Sigma(\theta_t)^{\frac{1}{2}}dW_t \tag{32}$$

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

Proof. Since θ_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^{th} coordinate of f_t, i.e., for an input x, $f_t(x) = (f_{t,i}(x))_{1\leq i\leq k}$. Let $i \in 1, ..., k$, we have

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

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

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

where $\Gamma_t(x) = (\mathrm{Tr}(\Sigma(\theta_t)^{\frac{1}{2}}\nabla_2 f_i(x,\theta_t)\Sigma(\theta_t)^{\frac{1}{2}}))_{1\leq i\leq k}$.
Therefore, the dynamics of the vector $f_t(\mathcal{X})$ is given by

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

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

With the quadratic loss $\ell(z,y) = \frac{1}{2}||z-y||^2$, the SDE (32) is equivalent to

$$df_t(\mathcal{X}) = [-\frac{1}{N}K^L_{\theta_t}(\mathcal{X},\mathcal{X})(f_t(\mathcal{X})-\mathcal{Y}) + \frac{1}{2}\frac{\eta}{S}\Gamma_t(\mathcal{X})]dt + \sqrt{\frac{\eta}{S}}\nabla_\theta f(\mathcal{X},\theta_t)\Sigma(\theta_t)^{\frac{1}{2}}dW_t. \tag{33}$$

This is an Ornstein-Uhlenbeck process (mean-reverting process) with time dependent parameters. The additional term Γ_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 $n_1, n_2, ..., n_L$, Jacot et al. [2018] proved that, with GD, the kernel $K^L_{\theta_t}$ converges to a kernel K^L that depends only on L (number of layers) for all $t < T$ when $n_1, n_2, ..., n_l \to \infty$, 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 figure **??** 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

$$f_t(\mathcal{X}) = e^{-\frac{t}{N}\hat{K}^L} f_0(\mathcal{X}) + (I - e^{-\frac{t}{N}\hat{K}^L})\mathcal{Y} + A_t(\mathcal{X}) \tag{34}$$

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

$$f_t(x) = f_0(x) + K^L(x, \mathcal{X})(\hat{K}^L)^{-1}(I - e^{-\frac{t-s}{N}\hat{K}^L})(\mathcal{Y} - f_0(\mathcal{X})) + Z_t(x) + R_t(x), \tag{35}$$

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 $Z_t(x) = \frac{\eta}{2S}\left[\int_0^t \Gamma_s(x)ds + \int_0^t K(x, \mathcal{X})(\hat{K}^L)^{-1}(I - e^{-\frac{(t-s)}{N}\hat{K}^L})\Gamma_s(\mathcal{X})ds\right]$.

Proof. Using the approximation of the NTK by K^L as $n_1, n_2, ..., n_L \to \infty$, 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.

□