On the Neural Tangent Kernel of Deep Networks with Orthogonal Initialization

Wei Huang[1 *†], **Weitao Du**[2 *], **Richard Yi Da Xu**[1]
[1]University of Technology Sydney, Australia
[2]Northwestern University, USA
wei.huang-6@student.uts.edu.au, weitao.du@northwestern.edu, yida.xu@uts.edu.au

Abstract

The prevailing thinking is that orthogonal weights are crucial to enforcing dynamical isometry and speeding up training. The increase in learning speed that results from orthogonal initialization in linear networks has been well-proven. However, while the same is believed to also hold for nonlinear networks when the dynamical isometry condition is satisfied, the training dynamics behind this contention have not been thoroughly explored. In this work, we study the dynamics of ultra-wide networks across a range of architectures, including Fully Connected Networks (FCNs) and Convolutional Neural Networks (CNNs) with orthogonal initialization via neural tangent kernel (NTK). Through a series of propositions and lemmas, we prove that two NTKs, one corresponding to Gaussian weights and one to orthogonal weights, are equal when the network width is infinite. Further, during training, the NTK of an orthogonally-initialized infinite-width network should theoretically remain constant. This suggests that the orthogonal initialization cannot speed up training in the NTK (lazy training) regime, contrary to the prevailing thoughts. In order to explore under what circumstances can orthogonality accelerate training, we conduct a thorough empirical investigation outside the NTK regime. We find that when the hyperparameters are set to achieve a linear regime in nonlinear activation, orthogonal initialization can improve the learning speed with a large learning rate or large depth.

1 Introduction

Deep learning has been responsible for a step-change in performance across machine learning, setting new benchmarks for state-of-the-art performance in many applications, from computer vision [Nixon and Aguado, 2019], natural language processing [Devlin *et al.*, 2018], to reinforcement learning [Mnih *et al.*, 2015], and more. The mean field theory [Poole *et al.*, 2016; Schoenholz *et al.*, 2016] recently opened a gate to analyze the principles behind neural networks with random, infinite width, and fully-connected networks as the first subjects. Broadly, what [Schoenholz *et al.*, 2016] discovered, and then empirically proved, is that there exists a critical initialization called the edge of chaos, allowing the correlation signal from the data to go infinitely far forward and preventing vanishing or exploding gradients.

Critical initialization requires the mean squared singular value of a network's input-output Jacobian to be $O(1)$. It was already known that the learning process in deep linear networks could be dramatically accelerated by ensuring all singular values of the Jacobian being concentrated near 1, a property known as *dynamical isometry* [Saxe *et al.*, 2013]. However, what was not known was how to impose dynamical isometry in deep nonlinear networks. [Pennington *et al.*, 2017] conjectured that they could do so with techniques based on free probability and random matrix theory, giving rise to a new and improved form of initialization in deep nonlinear networks. Since then, dynamical isometry has been introduced to various architectures, such as residual networks [Tarnowski *et al.*, 2018; Ling and Qiu, 2019], convolutional networks [Xiao *et al.*, 2018], or recurrent networks [Chen *et al.*, 2018] with excellent performance on real-world datasets.

In fully connected networks, two key factors help to ensure dynamical isometry. One is orthogonality, and the other is appropriately tuning weights' and biases' parameters to establish a linear regime in nonlinear activation [Pennington *et al.*, 2017]. In straightforward scenarios, orthogonal initialization is usually enough to impose dynamical isometry in a linear network. The benefit of orthogonality in linear networks has been proven recently [Hu *et al.*, 2020]. However, the dynamics of nonlinear networks with orthogonal initialization has not been investigated. The roadblock is that it has been unclear how to derive a simple analytic expression for the training dynamics.

Hence, to fill this gap, we look to a recent technique called neural tangent kernel (NTK) [Jacot *et al.*, 2018; Huang and Yau, 2019; Arora *et al.*, 2019; Allen-Zhu *et al.*, 2019; Du *et al.*, 2018; Zou *et al.*, 2020], developed for studying the evolution of a deep network using gradient descent in the infinite width limit. NTK is a kernel characterized by a derivative of the output of a network to its parameters. It has been shown that the NTK of a network with Gaussian

*Equal contribution.
†Contact Author

initialization converges to a deterministic kernel and remains unchanged during gradient descent in the infinite-width limit. We extend these results to the orthogonal initialization case and find that orthogonal weights contribute to the same properties for NTK. Given a sufficiently small learning rate and wide width, the network optimized by gradient descent behaves as a model linearized about its initial parameters [Lee *et al.*, 2019]. These dynamics are called NTK regime, or *lazy training* [Chizat *et al.*, 2019]. As the learning rate gets larger or the network becomes deeper, outside of the NTK regime, we expect that there will be new phenomena that can differentiate two initialization. To summarize, our contribution is as follows,

- We prove that the NTK of an orthogonally-initialized network converges to the NTK of a network initialized by Gaussian weights in the infinite-width limit. Besides, theoretically, during training, the NTK of an orthogonally-initialized infinite-width network stays constant in the infinite-width limit.
- We prove that the NTK of an orthogonally-initialized network across architectures, including FCNs and CNNs, varies at a rate of the same order for finite-width as the NTK of a Gaussian-initialized network. Therefore, there are no significant improvements brought by orthogonal initialization for wide and nonlinear networks compared with Gaussian initialization in the NTK regime.
- We conduct a thorough empirical investigation of training speed outside the NTK regime to complement theoretical results. We show that orthogonal initialization can speed up training in the large learning rate and depth regime when the hyper-parameters are set to achieve a linear regime in nonlinear activation.

2 Related Work

[Hu *et al.*, 2020]'s investigation of orthogonal initialization in linear networks provided a rigorous proof that drawing the initial weights from the orthogonal group speeds up convergence relative to standard Gaussian initialization. However, deep nonlinear networks are much more complicated, making generating proof the same in these nonlinear settings much more difficult. For example, [Sokol and Park, 2018] attempted to explain why dynamical isometry imposed through orthogonal initialization can significantly increase training speed. They showed a connection between the maximum curvature of the optimization landscape, as measured by a Fisher information matrix (FIM) and the spectral radius of the input-output Jacobian, which partially explains why networks with greater isometric are able to train much faster.

[Jacot *et al.*, 2018], who conceived of the neural tangent kernel, shows that NTK both converges to an explicit limiting kernel in the infinite-width networks and remains constant during training with Gaussian initialization. [Lee *et al.*, 2019] reached the same conclusion from a different angle with a demonstration that the gradient descent dynamics of the original neural network fall into its linearized dynamics regime. While the original work of NTK is groundbreaking in producing an equation to predict the behavior of gradient descent in the NTK regime, it assumes the with goes to infinity in a sequential order. [Yang, 2019; Yang, 2020] strengthened the proof by taking the limit simultaneously. Besides, [Arora *et al.*, 2019; Allen-Zhu *et al.*, 2019; Du *et al.*, 2018] have proven the same proprieties of NTK and global convergence of deep networks in non-asymptotic ways. However, all of these studies did not treat the orthogonal initialization as with our work.

3 Preliminaries

3.1 Networks and Parameterization

Suppose there are D training points denoted by $\{(x_d, y_d)\}_{d=1}^{D}$, where input $X = (x_1, \ldots, x_D) \in \mathbb{R}^{n_0 \times D}$, and label $Y = (y_1, \ldots, y_D) \in \mathbb{R}^{n_L \times D}$. We consider the following architectures:

Fully-Connected Network (FCN). Consider a fully-connected network of L layers of widths n_l, for $l = 0, \cdots, L$, where $l = 0$ is the input layer and $l = L$ is output layer. Following the typical nomenclature of literature, we denote synaptic weight and bias for the l-th layer by $W^l \in \mathbb{R}^{n_l \times n_{l-1}}$ and $b^l \in \mathbb{R}^{n_l}$, with a point-wise activations function $\phi : \mathbb{R} \to \mathbb{R}$. For each input $x \in \mathbb{R}^{n_0}$, pre-activations and post-activations are denoted by $h^l(x) \in \mathbb{R}^{n_l}$ and $x^l(x) \in \mathbb{R}^{n_l}$ respectively. The information propagation for $l \in \{1, \ldots, L\}$ in this network is govern by,

$$x_i^l = \phi(h_i^l), \quad h_i^l = \sum_{j=1}^{n_l} W_{ij}^l x_j^{l-1} + b_i^l, \tag{1}$$

Convolutional Neural Network (CNN). For notational simplicity, we consider a 1D convolutional networks with periodic boundary conditions. We denote the filter relative spatial location $\beta \in \{-k, \ldots, 0, \ldots, k\}$ and spatial location $\alpha \in \{1, \ldots, m\}$, where m is the spatial size. The forward propagation for $l \in \{1, \ldots, L-1\}$ is given by,

$$x_{i,\alpha}^l = \phi(h_{i,\alpha}^l), \quad h_{i,\alpha}^l = \sum_{j=1}^{n_l} \sum_{\beta=-k}^{k} W_{ij,\beta}^l x_{j,\alpha+\beta}^{l-1} + b_i^l, \tag{2}$$

where weight $W^l \in \mathbb{R}^{n_l \times n_{l-1} \times (2k+1)}$, and n_l is the number of channels in the l^{th} layer. The output layer of a CNN is processed with a fully-connected layer, $f_i(x) = h_i^L = \sum_{j=1}^{n_L} \sum_{\alpha} W_{ij,\alpha}^L x_{j,\alpha}^{L-1}$.

Standard parameterization requires the parameter set $\theta = \{W_{ij}^l, b_i^l\}$ is an ensemble generated by, $W_{ij}^l \sim \mathcal{N}(0, \frac{\sigma_w^2}{n_{l-1}})$, $b_i^l \sim \mathcal{N}(0, \sigma_b^2)$, where σ_w^2 and σ_b^2 are weight and bias variances. The variance of weights is scaled by the width of previous layer n_{l-1} to preserve the order of post-activations layer to be $O(1)$. We denote this parameterizationas *standard parameterizaiton*. However, this paramterization can lead to a divergence in derivation of neural tangent kernel. To overcome this problem, *ntk-parameterization* was introduced, $W_{ij}^l = \frac{\sigma_w}{\sqrt{n_{l-1}}} \omega_{ij}^l$, $b_i^l = \sigma_b \beta_i^l$, where $\omega_{ij}^l, \beta_i^l \sim \mathcal{N}(0, 1)$.

Network	Parameterization	W initialization	b initialization	layer equation
FCN	ntk Gaussian	$\omega_{ij} \sim \mathcal{N}(0,1)$	$\beta_i \sim \mathcal{N}(0,1)$	$h^l = \frac{\sigma_w}{\sqrt{n_{l-1}}}\omega^l x^{l-1} + \sigma_b \beta^l$
	ntk Orthogonal	$(\omega^l)^T \omega^l = n_{l-1}\mathbf{I}$		
	std Gaussian	$W_{ij} \sim \mathcal{N}(0, \frac{\sigma_w^2}{N_{l-1}})$	$b_i \sim \mathcal{N}(0, \sigma_b^2)$	$h^l = \frac{1}{\sqrt{s}} W^l x^{l-1} + b^l$
	std Orthogonal	$W^T W = \sigma_w^2 \mathbf{I}$		
CNN	ntk Gaussian	$\omega_{ij,\alpha} \sim \mathcal{N}(0,1)$	$\beta_i \sim \mathcal{N}(0,1)$	$h^l_\alpha = \sum_{\beta=-k}^{k} \frac{\sigma_w}{\sqrt{(2k+1)n_{l-1}}}\omega^l_\beta x^{l-1}_{\alpha+\beta} + \sigma_b \beta^l$
	ntk Orthogonal	$(\omega^l_\alpha)^T \omega^l_\alpha = n_{l-1}\mathbf{I}$		
	std Gaussian	$W_{ij,\alpha} \sim \mathcal{N}(0, \frac{\sigma_w^2}{(2k+1)N_{l-1}})$	$b_i \sim \mathcal{N}(0, \sigma_b^2)$	$h^l_\alpha = \frac{1}{\sqrt{s}} W^l_\beta x^{l-1}_{\alpha+\beta} + b^l$
	std Orthogonal	$W^T_\alpha W_\alpha = \frac{\sigma_w^2}{2k+1}\mathbf{I}$		

Table 1: Summary of improved standard parameterization and ntk-parameterization for Gaussian and orthogonal initialization. The abbreviation "std" stands for standard, and the "parameterization" is omitted after ntk or std.

3.2 Dynamical Isometry and Orthogonal Initialization

Consider the input-output Jacobian $J = \frac{\partial h^L}{\partial x^0} = \prod_{l=1}^{L} D^l W^l$, where h^L is output function, x^0 is input, and D^l is a diagonal matrix with elements $D^l_{ij} = \phi'(h^l_i)\delta_{ij}$. Ensuring all singular values of the Jacobian concentrate near 1 is a property known as *dynamical isometry*. In particular, It is shown that two conditions regarding singular values of W^l and D^l contribute crucially to the dynamical isometry in non-linear networks [Pennington *et al.*, 2017]. More precisely, the singular values of D^l can be made arbitrarily close to 1 by choosing a linear regime in a nonlinear activation. On the other hand, adopting a random orthogonal initialization can force the singular values of weights into 1. In particular, weights are drawn from a uniform distribution over scaled orthogonal matrices obeying,

$$(W^l)^T W^l = \sigma_w^2 \mathbf{I}, \tag{3}$$

This is the *standard parameterization* for orthogonal weights, and *ntk-parameterization* of orthogonality follows,

$$W^l_{ij} = \frac{\sigma_w}{\sqrt{n_{l-1}}}\omega^l_{ij}, \quad (\omega^l)^T \omega^l = n_{l-1}\mathbf{I}. \tag{4}$$

We show a summary of improved standard parameterization and ntk-parameterization across FCN and CNN for Gaussian and orthogonal initialization in Table 1. The factor s in the layer equation of standard parameterization is introduced to prevent divergence of NTK [Sohl-Dickstein *et al.*, 2020]. The core idea is to write the width of the neural network in each layer in terms of an auxiliary parameter, $n_l = sN_l$. Instead of letting $n_l \to \infty$, we adopt s as the limiting factor.

3.3 Neural Tangent Kernel

The neural tangent kernel (NTK) is originated from [Jacot *et al.*, 2018] and defined as,

$$\Theta_t(X, X) = \nabla_\theta f_t(\theta, X) \nabla_\theta f_t(\theta, X)^T. \tag{5}$$

where function f_t are the outputs of the network at training time t, i.e. $f_t(\theta, X) = h^L_t(\theta, X) \in \mathbb{R}^{D \times n_L}$, and $\nabla_\theta f_t(\theta, X) = \mathrm{vec}([\nabla_\theta f_t(\theta, x)]_{x \in X}) \in \mathbb{R}^{Dn_L}$. As such, the neural tangent kernel is formulated as a $Dn_L \times Dn_L$ matrix. Let η be the learning rate, and $\mathcal{L}$ be the loss function. The ynamics of gradient flow for parameters and output function are given by,

$$\begin{aligned} \frac{\partial \theta}{\partial t} &= -\eta \nabla_\theta \mathcal{L} = -\eta \nabla_\theta f_t(\theta, X)^T \nabla_{f_t(\theta,X)} \mathcal{L} \\ \frac{\partial f_t(\theta, X)}{\partial t} &= \nabla_\theta f_t(\theta, X) \frac{\partial \theta}{\partial t} = -\eta \Theta_t(X, X) \nabla_{f_t(\theta,X)} \mathcal{L}. \end{aligned} \tag{6}$$

This equation for f_t has no substantial insight in studying the behavior of networks because $\Theta_t(X, X)$ varies with the time during training. Interestingly, as shown by [Jacot *et al.*, 2018], the NTK $\Theta_t(X, X)$ converges to a deterministic kernel $\Theta_\infty(X, X)$ and does not change during training in the infinite-width limit, i.e. $\Theta_t(X, X) = \Theta_\infty(X, X)$. As a result, the infinite width limit of the training dynamics are given by,

$$\frac{\partial f_t(\theta, X)}{\partial t} = -\eta \Theta_\infty(X, X) \nabla_{f_t(\theta,X)} \mathcal{L}. \tag{7}$$

In the case of an MSE loss, $\mathcal{L}(y, f) = \frac{1}{2} \|y - f_t(\theta, x)\|_2^2$, the Equation (7) becomes a linear model with a solution,

$$f_t(\theta, X) = (\mathbf{I} - e^{-\eta \Theta_\infty(X,X)t})Y + e^{-\eta \Theta_\infty(X,X)t} f_0(\theta, X). \tag{8}$$

4 Theoretical Results

4.1 An Orthogonally Initialized Network is a Gaussian Process in the Infinite Width Limit

As stated in [Lee *et al.*, 2017; Matthews *et al.*, 2018], the pre-activation h^l_i of Gaussian initialized network tends to Gaussian processes (GPs) in the infinite-width limit. This is the proposition to construct the NTK in networks with Gaussian weights [Jacot *et al.*, 2018]. We extend this result to the orthogonal initialization across Fully Connected Networks (FCNs) and Convolutional Neural Networks (CNNs):

Theorem 1. *Consider a FCN of the form (1) at orthogonal initialization, with a Lipschitz nonlinearity ϕ, and in the limit as $n_1, ..., n_{L-1} \to \infty$, the pre-activations h^l_i, for $i = 1, ..., n_l$ and $l \in \{1, \ldots, L\}$, tend to i.i.d centered Gaussian processes of covariance Σ^l which is defined recursively by:*

$$\begin{aligned} \Sigma^1(x, x') &= \frac{\sigma_w^2}{n_0} x^T x' + \sigma_b^2 \\ \Sigma^l(x, x') &= \sigma_w^2 \mathbb{E}_{f \sim \mathcal{N}(0, \Sigma^{l-1})}[\phi(f(x))\phi(f(x'))] + \sigma_b^2, \end{aligned}$$

For a CNN of the form (2) at orthogonal initialization, and in the limit as $n_1, ..., n_{L-1} \to \infty$, the pre-activations $h^l_{i,\alpha}$ tend to Gaussian processes of covariance $\Sigma^l_{\alpha,\alpha'}$, which is defined recursively by:

$$\Sigma^1_{\alpha,\alpha'}(x,x') = \frac{\sigma_w^2}{n_0(2k+1)} \sum_{\beta=-k}^{k} x^T_{\alpha+\beta} x'_{\alpha'+\beta} + \sigma_b^2$$

$$\Sigma^l_{\alpha,\alpha'}(x,x') = \frac{\sigma_w^2}{(2k+1)} \sum_{\beta=-k}^{k} \big[\mathbb{E}_{f\sim\mathcal{N}\left(0,\Sigma^{l-1}_{\alpha+\beta,\alpha'+\beta}\right)} [\phi(f(x_{\alpha+\beta}))\phi(f(x'_{\alpha'+\beta}))]\big] + \sigma_b^2.$$

$$\Sigma^L(x,x') = \sum_{\alpha} \delta_{\alpha,\alpha'} \big[\mathbb{E}_{f\sim\mathcal{N}\left(0,\Sigma^{L-1}_{\alpha,\alpha'}\right)} [\phi(f(x_\alpha))\phi(f(x'_{\alpha'}))]\big]$$

Different from the independence property of Gaussian initialization, the entries of the orthogonal matrix are correlated. We use the Stein method and exchangeable sequence to overcome this difficulty and leave the detailed proof in the appendix. As shown by Theorem 1, neural networks with Gaussian and orthogonal initialization are in correspondence with an identical class of GPs.

4.2 Neural Tangent Kernel at Initialization

According to [Jacot *et al.*, 2018], the NTK of a network with Gaussian weights converges in probability to a deterministic kernel in the infinite-width limit. We show that the NTK of an orthogonally initialized network is identical to the one with Gaussian weights in the infinite-width limit.

Theorem 2. *Consider a FCN of the form (1) at orthogonal initialization, with a Lipschitz nonlinearity ϕ, and in the limit as the layers width $n_1, ..., n_{L-1} \to \infty$, the NTK $\Theta^L_0(x,x')$, converges in probability to a deterministic limiting kernel:*

$$\Theta^L_0(x,x') \to \Theta^L_\infty(x,x') \otimes \mathbf{I}_{n_L \times n_L}.$$

The scalar kernel $\Theta^L_\infty(x,x')$ is defined recursively by

$$\Theta^1_\infty(x,x') = \Sigma^1(x,x')$$
$$\Theta^l_\infty(x,x') = \sigma_w^2 \dot{\Sigma}^l(x,x') \Theta^{l-1}_\infty(x,x') + \Sigma^l(x,x'),$$

where

$$\dot{\Sigma}^l(x,x') = \mathbb{E}_{f\sim\mathcal{N}\left(0,\Sigma^{(l-1)}\right)} \left[\dot{\phi}(f(x)) \dot{\phi}(f(x'))\right],$$

For a CNN of the form (2) at orthogonal initialization, and in the limit as $n_1, ..., n_{L-1} \to \infty$, the NTK $\Theta^L_0(x,x')$, converges in probability to a deterministic limiting kernel:

$$\Theta^L_0(x,x') \to \Theta^L_\infty(x,x') \otimes \mathbf{I}_{n_L \times n_L}.$$

The scalar kernel $\Theta^L_\infty(x,x')$ is given recursively by

$$\Theta^1_{\alpha,\alpha'\infty}(x,x') = \Sigma^1_{\alpha,\alpha'}(x,x')$$

$$\Theta^l_{\alpha,\alpha'\infty}(x,x') = \frac{\sigma_w^2}{(2k+1)} \sum_{\beta} \big[\dot{\Sigma}^l_{\alpha+\beta,\alpha'+\beta}(x,x') \Theta^{l-1}_{\alpha+\beta,\alpha'+\beta\infty}(x,x') + \Sigma^l_{\alpha+\beta,\alpha'+\beta}(x,x')\big]$$

$$\Theta^L_\infty(x,x') = \sum_{\alpha} \delta_{\alpha,\alpha'} \big[\dot{\Sigma}^L_{\alpha,\alpha'}(x,x') \Theta^{L-1}_{\alpha,\alpha'\infty}(x,x') + \Sigma^L_{\alpha,\alpha'}(x,x')\big]$$

Remark 1. *Since the Lipschitz function is differentiable besides a measure zero set, then taking the expectation would not destroy the whole statement, which allows for the ReLU activation.*

In general, the NTK of CNNs can be computed recursively in a similar manner to the NTK for FCNs. However, the NTK of CNNs propagate differently by averaging over the NTKs regarding the neuron location of the previous layer. According to Theorem 2, the NTK of an orthogonally initialized network converges to an identical kernel as Gaussian initialization. This suggests these two NTKs are equivalent when the network structure (depth of L, filter size of $2k+1$, and activation of ϕ) and choice of hyper-parameters (σ_w^2 and σ_b^2) are the same in the infinite-width limit.

4.3 Neural Tangent Kernel During Training

It is shown that the NTK of a network with Gaussian initialization stays asymptotically constant during gradient descent training in the infinite-width limit, providing a guarantee for loss convergence [Jacot *et al.*, 2018]. We find that the NTK of orthogonally initialized networks have the same property, which is demonstrated below in an asymptotic way,

Theorem 3. *Assume that $\lambda_{\min}(\Theta_\infty) > 0$ and $\eta_{\text{critical}} = \frac{\lambda_{\min}(\Theta_\infty)+\lambda_{\max}(\Theta_\infty)}{2}$. Let $n = n_1, ..., n_{L-1}$ be the width of hidden layers. Consider a FCN of the form (1) at orthogonal initialization, trained by gradient descent with learning rate $\eta < \eta_{\text{critical}}$ (or gradient flow). For every input $x \in \mathbb{R}^{n_0}$ with $\|x\|_2 \le 1$, with probability arbitrarily close to 1,*

$$\sup_{t\ge 0} \frac{\|\theta_t - \theta_0\|_2}{\sqrt{n}},\ \sup_{t\ge 0} \left\|\hat{\Theta}_t - \hat{\Theta}_0\right\|_F = O(n^{-\frac{1}{2}}),\ \text{as} \quad n \to \infty\,. \tag{9}$$

where $\hat{\Theta}_t$ are empirical kernels of networks with finite width.

For a CNN of the form (2) at orthogonal initialization, trained by gradient descent with learning rate $\eta < \eta_{\text{critical}}$ (or gradient flow), for every input $x \in \mathbb{R}^{n_0}$ with $\|x\|_2 \le 1$, and filter relative spatial location $\beta \in \{-k, \ldots, 0, \ldots, k\}$, with probability arbitrarily close to 1,

$$\sup_{t\ge 0} \frac{\|\theta_{\beta,t} - \theta_{\beta,0}\|_2}{\sqrt{n}},\ \sup_{t\ge 0} \left\|\hat{\Theta}_t - \hat{\Theta}_0\right\|_F = O(n^{-\frac{1}{2}}). \tag{10}$$

[Jacot *et al.*, 2018] proved the stability of NTK under the assumption of global convergence of neural networks, while [Lee *et al.*, 2019] provided a self-contained proof of both global convergence and stability of NTK simultaneously. In this work, we refer to the proof strategy from [Lee *et al.*, 2019; Liu *et al.*, 2020] and extend it to the orthogonal case, as shown in the appendix.

To certificate this theorem empirically, we adopt three hidden layers Erf networks trained by gradient descent with learning rate $\eta = 1.0$ on a subset of the MNIST dataset of $D = 20$. We measure changes of weights, empirical NTK after $T = 2^{15}$ steps of gradient descent for varying width at both Gaussian and orthogonal initialization. Figure 1(a,b) show that the relative change in the first and last layer weights scales as $1/\sqrt{n}$ while second and third layer weights scale as $1/n$ with Gaussian and orthogonal weights respectively. In Figure 2(c), we observe the change in NTK is upper

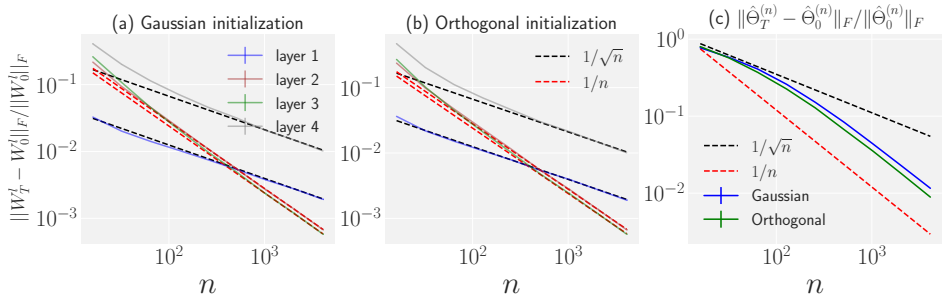

Figure 1: Changes of weights, empirical NTK on a three hidden layer Erf Network. Solid lines correspond to empirical simulation, and dotted lines are theoretical predictions, i.e., black dotted lines are $1/\sqrt{n}$ while red dotted lines are $1/n$. (a) Weight changes on the Gaussian initialized network. (b) Weight changes on the orthogonally initialized network. (c) NTK changes on networks with Gaussian and orthogonal initialization.

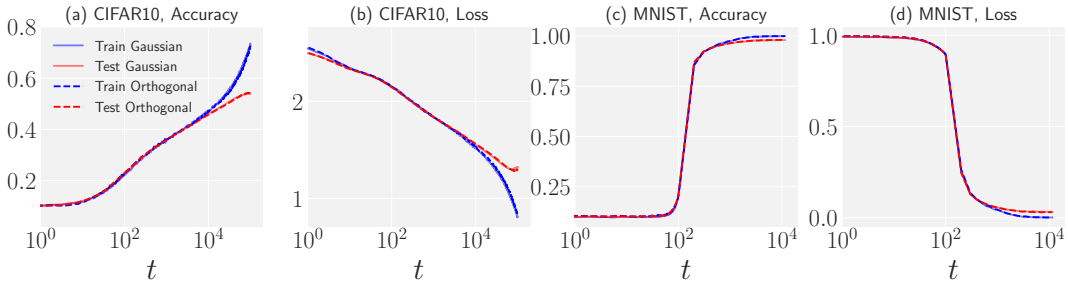

Figure 2: Orthogonally initialized networks behave similarly to the networks with Gaussian initialization in the NTK regime. (a,b) We adopt the network architecture of depth of $L = 5$, width of $n = 800$, activation of tanh function, with $\sigma_w^2 = 2.0$, and $\sigma_b^2 = 0.1$. The networks are trained by SGD with a small learning rate of $\eta = 10^{-3}$ for $T = 10^5$ steps with a batch size of 10^3 on cross-entropy loss on full CIFAR-10. (c,d) We adopt the network architecture of depth of $L = 9$, width of $n = 1600$, activation of ReLU function, with $\sigma_w^2 = 2.0$, and $\sigma_b^2 = 0.1$. The networks are trained by PMSProp with a small learning rate of $\eta = 10^{-5}$ for $T = 1.2 \times 10^4$ steps with a batch size of 10^3 on MSE loss on MNIST. While the solid lines stand for Gaussian weights, dotted lines represent orthogonal initialization.

bounded by $O(1/\sqrt{n})$ but is closer to $O(1/n)$ for both Gaussian and orthogonal initialization. The discrepancy between theoretical bound ($O(n^{-1/2})$) and experimental observation ($O(n^{-1})$) has been solved in [Huang and Yau, 2019], where they prove that relative change of empirical NTK of Gaussian initialized networks is bounded by $O(1/n)$. Without loss of generality, we infer that the proof framework is suitable for orthogonal weights.

5 Numerical Experiments

Our theoretical result indicates that ultra-wide networks with Gaussian and orthogonal initialization should have the same convergence rate during the gradient descent training. This means that two different initializations have similar training dynamics for loss and accuracy function in the NTK regime. Thus, it is now for us to verify our theories in practice. To this end, we perform a series of experiments on MNIST and CIFAR10 dataset. All the experiments are performed with the standard parameterization with TensorFlow.

We compare the train and test loss and accuracy with two different initialization, i.e., Gaussian and orthogonal weights using $D = 256$ samples on full CIFAR-10 and MNIST dataset, as summarized in Figure 2. To reduce noise, we averaged the results over 30 different instantiations of the networks. Figures 2(a,b) show the results of the experiments on networks of depth $L = 5$, width $n = 800$, and activation tanh function, using SGD optimizer with a small learning rate of $\eta = 10^{-3}$ for $T = 10^5$ steps on CIFAR-10 dataset. Figure 2(c)(d) display the results on networks of depth $L = 9$, width $n = 1600$, and activation ReLU function, using PMSProp [Hinton *et al.*, 2012] optimizer with a small learning rate of $\eta = 10^{-5}$ for $T = 1.2 \times 10^4$ steps on MNIST. In all cases, we see an excellent agreement between the training dynamics of the two initialization, which is consistent with our theoretical finding (Theorem 3).

Having confirmed the consistency between training speed of networks with Gaussian and orthogonal initialization in the NTK regime, our primary interest is to find when orthogonal initialization accelerates the training speed for nonlinear networks. We need to go beyond the NTK regime and experiment with an additional requirement for hyper-parameters according to the evidence that orthogonal initialization increases learning speeds when the variance of weights and biases is set to achieve a liner regime in nonlinear activation [Pennington *et al.*, 2017].

Following [Pennington *et al.*, 2017], we set $\sigma_w^2 = 1.05$, and $\sigma_b^2 = 2.01 \times 10^{-5}$, and $\phi(x) = \tanh(x)$. We then vary the width of network in one set of experiments as $n = 400, 800$ and 1600 when $L = 50$, and the depth in another

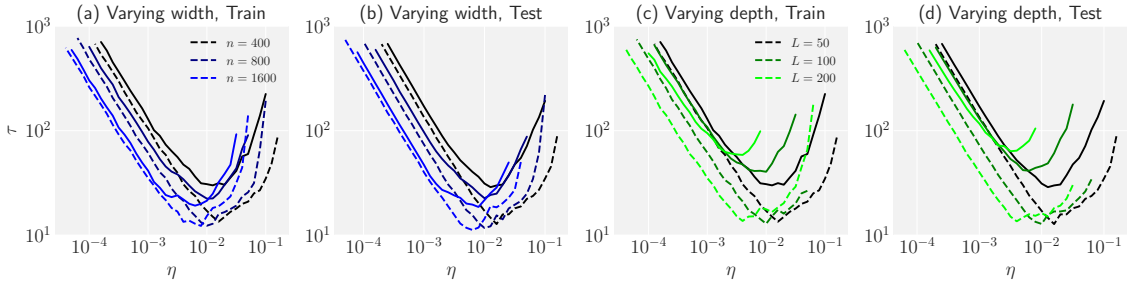

Figure 3: The steps τ as a function of learning rate η of two lines of networks on both train and test dataset. The results of orthogonal networks are marked by dotted lines while those of Gaussian initialization are plotted by solid lines. Networks with varying width, i.e. $n = 400, 800$, and 1600, on (a) train set and (b) test set; Networks with varying depth, i.e. $L = 50, 100$, and 200, on (c) train set and (d) test set. Different colors represent the corresponding width and depth. While curves of orthogonal initialization are lower than those of Gaussian initialization in the small learning rate phase, the differences become more significant in the large learning rate. Besides, the greater the depth of the network, the more significant the difference in performance between orthogonal and Gaussian initialization.

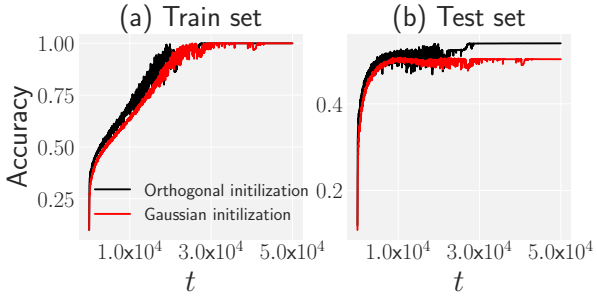

Figure 4: Learning dynamics measured by the optimization and generalization accuracy on train set and test set, for networks of depth $L = 100$ and width $n = 400$. We additionally average our results over 30 different instantiations of the network to reduce noise. Black curves are the results of orthogonal initialization, and red curves are performances of Gaussian initialization. (a) The training speed of an orthogonally initialized network is faster than that of a Gaussian initialized network. (b) On the test set, compared to the network with Gaussian initialization, the orthogonally initialized network not only learns faster but ultimately converges to a higher generalization performance.

as $L = 50, 100$ and 200, when $n = 400$. All networks are trained by SGD optimizer on CIFAR-10 dataset. To evaluate the relationship between the learning rate and training speed, we select a threshold accuracy of $p = 0.25$ and measure the first step τ when accuracy exceeds p. Figure 3 shows the steps of τ as a function of the learning rate of η for both the training and testing sets.

The results in Figure 3 suggest a more quantitative analysis of the learning process until convergence. We train networks listed in Figure 3 for 5×10^4 steps with a certain learning rate. We show the results of a certain network of depth $L = 100$ and width $n = 400$ trained with a learning rate $\eta = 0.01$ as a typical example in Figure 4. The results of other network structures can be found in the appendix. It is shown that the training speed of orthogonally initialized networks is faster than that of Gaussian initialized networks *outside* the NTK regime. At the same time, orthogonally initialized networks can finally obtain a higher generalization result.

We draw two main conclusions from these experiments. First, orthogonal initialization results in faster training speeds and better generalization than Gaussian initialization in the large learning rate phase. It was shown that the large learning rate phase has many different properties from the small learning rate phase [Lewkowycz *et al.*, 2020; Li *et al.*, 2019]. Our finding can be seen as another effect in the large learning rate phase. Second, given the constant width, the greater the depth of the network, the more significant the difference in performance between orthogonal and Gaussian initialization. This phenomenon is consistent with the theoretical result observed in deep linear networks. It was found that the width needed for efficient convergence to a global minimum with orthogonal initialization is independent of the depth. In contrast, the width needed for efficient convergence with Gaussian initialization scales proportionally in depth [Hu *et al.*, 2020].

6 Conclusion

This study on the neural tangent kernel of wide and nonlinear networks with orthogonal initialization has proven, theoretically and empirically, that the NTK of an orthogonally-initialized network across both FCN and CNN converges to the same deterministic kernel of a network initialized from Gaussian weights in the finite-width limit. We find that with an infinite-width network and a gradient descent (gradient flow) training scheme, the NTK of an orthogonally initialized network does not change during training. Further, it has the same order convergence rate from a finite to an infinite width limit as that of a Gaussian initialized network. Our theoretical results suggest that the dynamics of wide networks with orthogonal initialization behave similarly to that of Gaussian networks with a small learning rate verified by experiments. This observation implies that orthogonal initialization is only effective when not in the lazy (NTK) regime. And it is consistent with the fact that the infinite-width analysis does not explain the practically observed power of deep learning [Arora *et al.*, 2019; Chizat *et al.*, 2019]. Last, we find that orthogonal networks can outperform Gaussian networks in the large learning rate and depth on both train and test sets.

References

[Allen-Zhu *et al.*, 2019] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

[Arora *et al.*, 2019] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pages 8139–8148, 2019.

[Chen *et al.*, 2018] Minmin Chen, Jeffrey Pennington, and Samuel S Schoenholz. Dynamical isometry and a mean field theory of rnns: Gating enables signal propagation in recurrent neural networks. *arXiv preprint arXiv:1806.05394*, 2018.

[Chizat *et al.*, 2019] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2937–2947, 2019.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Du *et al.*, 2018] Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

[Hinton *et al.*, 2012] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8), 2012.

[Hu *et al.*, 2020] Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. *arXiv preprint arXiv:2001.05992*, 2020.

[Huang and Yau, 2019] Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. *arXiv preprint arXiv:1909.08156*, 2019.

[Jacot *et al.*, 2018] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

[Lee *et al.*, 2017] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[Lee *et al.*, 2019] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pages 8570–8581, 2019.

[Lewkowycz *et al.*, 2020] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

[Li *et al.*, 2019] Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems*, pages 11674–11685, 2019.

[Ling and Qiu, 2019] Zenan Ling and Robert C Qiu. Spectrum concentration in deep residual learning: a free probability approach. *IEEE Access*, 7:105212–105223, 2019.

[Liu *et al.*, 2020] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. On the linearity of large non-linear models: when and why the tangent kernel is constant. *Advances in Neural Information Processing Systems*, 33, 2020.

[Matthews *et al.*, 2018] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

[Mnih *et al.*, 2015] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

[Nixon and Aguado, 2019] Mark Nixon and Alberto Aguado. *Feature extraction and image processing for computer vision*. Academic Press, 2019.

[Pennington *et al.*, 2017] Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. In *Advances in neural information processing systems*, pages 4785–4795, 2017.

[Poole *et al.*, 2016] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in neural information processing systems*, pages 3360–3368, 2016.

[Saxe *et al.*, 2013] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[Schoenholz *et al.*, 2016] Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. *arXiv preprint arXiv:1611.01232*, 2016.

[Sohl-Dickstein *et al.*, 2020] Jascha Sohl-Dickstein, Roman Novak, Samuel S Schoenholz, and Jaehoon Lee. On the infinite width limit of neural networks with a standard parameterization. *arXiv preprint arXiv:2001.07301*, 2020.

[Sokol and Park, 2018] Piotr A Sokol and Il Memming Park. Information geometry of orthogonal initializations and training. *arXiv preprint arXiv:1810.03785*, 2018.

[Tarnowski *et al.*, 2018] Wojciech Tarnowski, Piotr Warchoł, Stanisław Jastrzębski, Jacek Tabor, and Maciej A Nowak. Dynamical isometry is achieved in residual networks in a universal way for any activation function. *arXiv preprint arXiv:1809.08848*, 2018.

[Xiao *et al.*, 2018] Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. *arXiv preprint arXiv:1806.05393*, 2018.

[Yang, 2019] Greg 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.

[Yang, 2020] Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

[Zou *et al.*, 2020] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109(3):467–492, 2020.