UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Understanding deep learning through ultra-wide neural networks

by

Wei Huang

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2021

Certificate of Authorship/Originality

I, Wei Huang, declare that this thesis, is submitted in fulfilment of the requirements for the award of PhD, in the Faculty of Engineering and IT at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Austraian Government Research Training Program.

Signature: Production Note: Signature removed prior to publication. Date: 09/09/2021

Acknowledgements

In the first place, I want express my sincere gratitude to my supervisor, professor Richard Xu, for his unwavering support and invaluable instructions. Prof. Xu has brought me into the topics of non-parametric Bayesian and deep learning. He taught me a lot of machine learning knowledge by insisting on teaching every week. By revising the paper along with the discussion, he taught me how to write scientific papers. His profound knowledge and rigorous scientific research spirit deeply influenced the research path I would not have been able to carry out research projects without his valuable suggestions.

I would like to extend my sincere thanks to my co-supervisor Prof. Massimo Piccardi for helping with my candidature assessment, and I am greatly inspired by his warm support.

I am grateful to my collaborator Mr. Weitao Du who has been providing valuable support on mathematical techniques and proofs. We have been best friends since undergraduate students. We have spent plenty of time on discussion and collaborating. I have improved a lot in mathematical proofs. I sincerely appreciate my collaborator Mr. Yayong Li for his considerable discussion on the graph neural network. I am thankful for what I have learned from him during our collaborations.

I would like to offer my special thanks to all other members of our machine learning lab, especially, Mr. Chunrui Liu, Mr. Yunce Zhao, Mr. Ziyue Zhang, Mr. Xuan Liang, Mr. Chen Deng, Dr. Caoyuan Li, Miss Ying Li, Miss Sophie Yuan, Mr. Chris Markos, Dr. Jason Traish, Miss Ningkai Xiao, Miss Erica Huang, Mr. Haodong Chang, and Dr. Shawn Jiang.

My sincere thanks go to my friends for a happy time with them during my Ph.D. study. Those thanks especially go to Dr. Haihan Sun, Dr. Ye Shi, Miss Xuan Wang, Miss Ye Su, Mr. Feng Shan, Mr. Haiming Qian, Mr. Leijie Zhang, Dr. Hongwen Yu, Dr. Tianyu Yang, Dr. Wentian Zhang, Miss Maral Ansari, Mr. Li Wang and Miss Chen Wang.

Finally, I would like to express my forever thanks to my parents for their selfless love and eternal support. My parents will always be my strong backing.

ABSTRACT

Understanding deep learning through ultra-wide neural networks

by

Wei Huang

Deep learning has been responsible for a step-change in performance across machine learning, setting new benchmarks in a large number of applications. However, the existing accounts fail to resolve why deep learning can achieve such great success. There is an urgent need to address the deep learning theory caused by the demand of understanding the principles of deep learning. One promising theoretical tool is the infinitely-wide neural network. This thesis focuses on the expressive power and optimization property of deep neural networks through investigating ultra-wide networks with four main contributions.

We first use the mean-field theory to study the expressivity of deep dropout networks. The traditional mean-field analysis adopts the gradient independence assumption that weights used during feed-forward are drawn independently from the ones used in backpropagation, which is not how neural networks are trained in a real setting. By breaking the independence assumption in the mean-field theory, we perform theoretical computation on linear dropout networks and a series of experiments on dropout networks. Furthermore, we investigate the maximum trainable length for deep dropout networks through a series of experiments and provide a more precise empirical formula that describes the trainable length than the original work.

Secondly, we study the dynamics of fully-connected, wide, and nonlinear networks 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. This suggests that the orthogonal initialization cannot speed up training in the NTK regime. Last, with a thorough empirical investigation, we find that orthogonal initialization increases learning speeds in scenarios with a large learning rate or large depth.

The third contribution is characterizing the implicit bias effect of deep linear networks for binary classification using the logistic loss with a large learning rate. We claim that depending on the separation conditions of data, the loss will find a flatter minimum with a large learning rate. We rigorously prove this claim under the assumption of degenerate data by overcoming the difficulty of the non-constant Hessian of logistic loss and further characterize the behavior of loss and Hessian for non-separable data.

Finally, we demonstrate the trainability of deep Graph Convolutional Networks (GCNs) by studying the Gaussian Process Kernel (GPK) and Graph Neural Tangent Kernel (GNTK) of an infinitely-wide GCN, corresponding to the analysis on expressivity and trainability, respectively. We formulate the asymptotic behaviors of GNTK in the large depth, which enables us to reveal the dropping trainability of wide and deep GCNs at an exponential rate.

List of Publications

Journal Papers

J-1. **Wei Huang** and Ricahrd Yi Da Xu, “Gaussian Process Latent Variable Model Factorization for Context-aware Recommender Systems," 2019. Submitted to *Pattern Recognition Letters*.

Conference Papers

C-1. **Wei Huang**, Richard Yi Da Xu, Weitao Du, Yutian Zeng, and Yunce Zhao, “Mean field theory for deep dropout networks: digging up gradient backpropagation deeply,” *ECAI 2020, 24th European Conference on Artificial Intelligence.*

C-2. **Wei Huang**, Weitao Du, and Richard Yi Da Xu, “On the neural tangent kernel of deep networks with orthogonal initialization,” 2020. Submitted to IJCAI 2021.

C-3. **Wei Huang**, Weitao Du, Richard Yi Da Xu, and Chunrui Liu, “Implicit bias of deep linear networks in the large learning rate phase,” 2020. Submitted to IJCAI 2021.

C-4. **Wei Huang**, Yayong Li, Weitao Du, Richard Yi Da Xu, Jie Yin, and Ling Chen, “Wide Graph Neural Networks: Aggregation ProvablyLeads to Exponentially Trainability Loss,” 2021. Submitted to ICML 2021.

Contents

List of Figures

Abbreviation

ML - Machine Learning

DL - Deep Learning

FCN - Fully-Connected Network

MLP - Multi-Layer Perceptro

GP - Gaussian Process

GPK - Gaussian Process Kernel

GNTK - Graph Neural Tangent Kernel

GD - Gradient Descent

Resnet - Residual Network

ReLU - Rectified Linear Unit

Erf - Error Function

FIM - Fisher Information Matrix

CLT - Central Limit Theorem

MSE - Mean Squared Error

SVM - Support Vector Machine

GCN - Graph Convolutional Network

Chapter 1

Introduction

1.1 Background

1.1.1 Machine learning

Machine learning is the study of algorithms that realize pattern precognition from real-world data automatically. By extract relevant information from data through modeling, ML is known to have a wide range of applications.

Machine learning is one of the fastest-growing branches of artificial intelligence. In the 1980s, symbolic learning was the mainstream of machine learning, and from the 1990s to around 2014, it has been the dominant statistical machine learning. Since 2014, deep learning has become the mainstream method, and still maintains its dominant position in academia and industry.

From the perspective of tasks, machine learning can be divided into supervised learning and unsupervised learning respectively. Supervised learning requires knowing the label of the data, and the main tasks are regression and classification. For example, for computer vision, how to recognize handwritten digits is a typical classification task. Instead of providing data labels, unsupervised learning generally processes tasks such as clustering and dimensionality reduction. In this thesis, we mainly account for the theory of supervised learning, especially imagine classification and graph classification.

1.1.2 Deep learning

Deep learning is a sub-field of machine learning. It is a new method of learning representations from data, emphasizing learning from successive layers, which correspond to increasingly meaningful representations. The "depth" in "deep learning" does not refer to any other concept regarding deep, but refers to stacking layers. Figure 1.1 shows a typical fully-connected neural network. Until now, deep learning has been developed with dozens or even hundreds of presentation layers.

At the early stage of deep learning, a fully-connected network (FNN) is the fundamental structure [2, 3, 4]. The full connection layer is generally composed of two parts, one is the linear layer, and another is the non-linear layer. In order to express the following formula more clearly, the superscript in the variable name represents the layer in which it is located. Consider a FCN of L layers of width N with dropout. We denote synaptic weight and bias for the l-th layer by W^l and b^l; pre-activations and post-activations by z^l and y^l respectively, where activation stands for the non-linear layer. Finally, we take the input to be $y_i^0 = x_i$. The linear part is a mainly linear transformation, which can be written as,

$$z_i^l = \sum_j W_{ij}^l y_j^{l-1} + b_i^l, \tag{1.1}$$

and the nonlinear part can be expressed as,

$$y_i^l = \phi(z_i^l). \tag{1.2}$$

Later, the rise of convolutional neural networks (CNN) brings deep learning into a new era. In 2012, Alex Krizhevsky took the convolutional neural network into an ImageNet competition (the equivalent of the Olympics in importance) and made a great hit [5]. Since then, many companies have started to use deep learning as the core of their services. For example, Facebook uses it in their auto-tagging algorithm, Google uses it in photo search, Amazon uses it in product recommendations, and

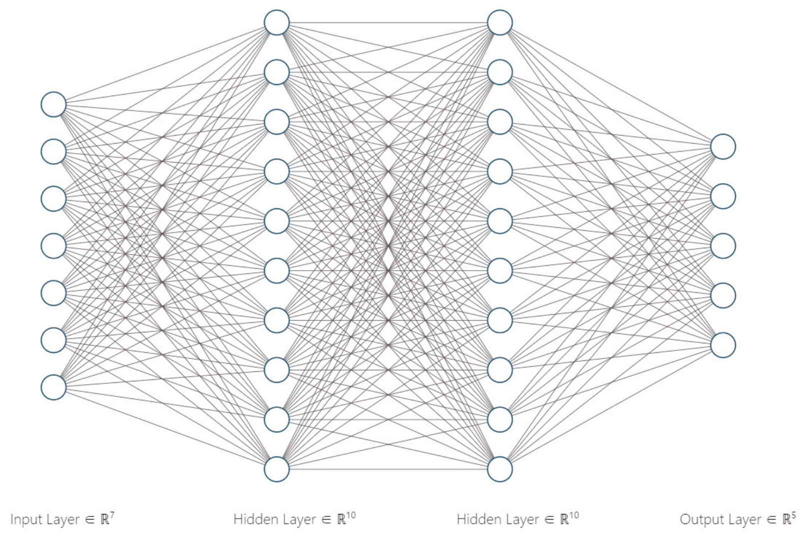

Figure 1.1 : A typical fully-connected neural network.

Instagram is used in their search engine.

For notational simplicity, we consider a 1D convolutional network with periodic boundary conditions. We denote the spatial location $\beta \in [-k, k]$ and spatial location $\alpha \in \{1, \ldots, m\}$, with m being the spatial size. Then the forward propagation is given by,

$$y^l_{i,\alpha} = \phi(z^l_{i,\alpha}), \quad z^l_{i,\alpha} = \sum_{j=1}^{n_l} \sum_{\beta=-k}^{k} W^l_{ij,\beta} y^{l-1}_{j,\alpha+\beta} + b^l_i, \tag{1.3}$$

where n_l is the channel size and weight in each layer is denoted as W^l. The output layer is processed with a fully-connected layer, $f_i(x) = h^L_i = \sum_{j=1}^{n_L} \sum_\alpha W^L_{ij,\alpha} y^{L-1}_{j,\alpha}$. In chapter 4, we consider the training dynamics of networks across FNN and CNN.

To overcome the over-fitting problem, Alex and Hinton used the Dropout algorithm which is a regularization technique to prevent over-fitting [5]. Network units are randomly dropped during training in dropout, which can prevent complex co-adaptations [6]. Dropout is also the focus of chapter 3 in this thesis. We can express an FNN with dropout as,

$$z^l_i = \frac{1}{\rho} \sum_j W^l_{ij} p_j y^{l-1}_j + b^l_i, \quad y^l_i = \phi(z^l_i) + y^{l-1}_i. \tag{1.4}$$

where p_j is sampled from the Bernoulli distribution, $p_j \sim p^\rho(1-p)^{1-\rho}$. In addition, recent seminal innovations have been proposed to improve the performance of neural networks further. For example, residual networks [7] and batch normalization [8], which were introduced to solve the gradient vanishing and exploding problem, enabled the trainable length to be very deep.

Recently, Graph Convolutional Networks (GCN) have shown incredible abilities to learn node or graph representations and achieved superior performances for various downstream tasks, such as node classifications [9, 10, 11, 12], graph classifications [13, 14, 15, 16, 17, 18], link predictions [19], etc. A GCN layer is actually designed as the message-passing operation, where messages are aggregated together from nodes to nodes within the graph connections. It is the mechanism on how each node aggregate neighboring representations and integrate them together to update its own representation that makes different GNN architectures varying in their performances [9, 10, 11].

Graph neural networks (GNNs) are powerful in practical performance when dealing with graph-structured data. However, it is known that GNN suffers from so call over-smoothing problems, where node features tend to be identical after aggregation operation. Despite this understanding is prevailing, rare theoretical analysis has been implemented to exploit the dynamics of deep GNN. In chapter 6, we derive the dynamics of wide GNN for node and graph classification problems based on techniques for infinitely wide GNN. Furthermore, theoretical derivative on the residual connection technique of graph network shows it can mildly alleviate exponential decay. Our work reveals aggregation is not only a successful factor of graph network but also the root cause of graph network can not be too deep. Finally, all theoretical results in this work can be verified by numerical experiments.

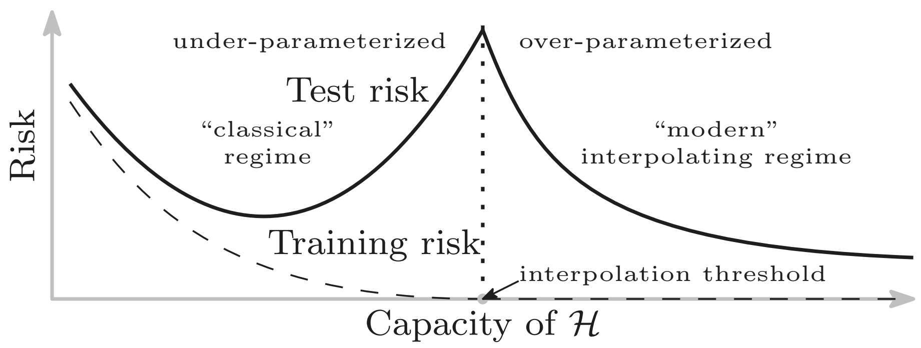

Figure 1.2 : Double descent phenomenon in deep learning versus U-shaped curve in traditional machine learning [1].

1.2 Motivation and Contribution

In deep learning, over-parameterization is a common phenomenon. This is contrary to the traditional convention that there is a trade-off between training error and generalization power in machine learning. On the one hand, we need to tune the number of parameters in the model to be big enough so that the model can be fully fitted to data, thus achieving a very small training error. On the other hand, the complexity of the model should be small enough so as to obtain reasonable test error by preventing the so-called over-fitting problem.

In Figure 1.2, we show that traditional understanding of the relationship between model performance and model complexity can be characterized by a U-shape curve. However, in deep learning, it turns out that as we increase the number of parameters in networks, the test error keeps decreasing instead of increasing. The new understanding is characterized by a double descent curve, as shown in Figure 1.2. There is plenty of experiments confirming that the over-parameterized model performs better. However, our theoretical comprehension is limited regarding the double descent phenomenon. Researchers have been trying to explain why deep learning can achieve such great accomplishment theoretically, and answer this big question through three perspectives: expressivity, optimization, and generalization.

The expressive power (expressivity) of neural networks is usually studied at initialization before training. It is a measure of how complex a function can be represented by a neural network. According to the universal approximation theorem [20, 21], neural networks can represent a wide variety of functions when given appropriate weights. However, whether a neural network can reach a powerful representation depends on its optimization process. In particular, the optimization property of a neural network stands for its ability to find a global minimum on the training dataset by optimization methods such as gradient descent. Nevertheless, in general, getting an error close to zero on the training set does not guarantee that we can use this result to achieve high accuracy on the training set. Thus, researchers introduce the concept of generalization to address the problem of performance on the test (unseen) data.

The aims of the project are to tackle the three aspects of deep learning theory through ultra-wide and deep neural networks. The main contributions are summarized as follows,

i. Conduct studies of expressivity of deep dropout networks through the mean-field theory. The mean-field theory shows that the existence of depth scales that limit the maximum depth of signal propagation and gradient backpropagation. However, the gradient backpropagation is derived under the gradient independence assumption where weights are independent from forward to backward. This is not how neural networks are trained in a real setting. Instead, the same weights used in a feed-forward step need to be carried over to its corresponding backpropagation. Using this realistic condition, we perform theoretical computation on linear dropout networks and a series of experiments on dropout networks with different activation functions. Our empirical results show an interesting phenomenon that the length gradients can backpropagate for a single input and a pair of inputs are governed by the same depth scale. Besides, we investigate the maximum trainable length for deep dropout networks through a series of experiments using MNIST and CIFAR10 and provide a more precise empirical formula that describes the trainable length than the original work.

ii. Conduct studies of optimization of deep networks with orthogonal initialization. In recent years, a critical initialization scheme of orthogonal initialization on deep nonlinear networks has been proposed. The orthogonal weights are crucial to achieve *dynamical isometry* for random networks, where the entire spectrum of singular values of an input-output Jacobian are around one. To understand the optimization of networks with orthogonal initialization, we study the Neural Tangent Kernel (NTK), which can describe dynamics of gradient descent training of wide network, and focus on fully-connected and nonlinear networks with orthogonal initialization. We prove that NTK of Gaussian and orthogonal weights are equal when the network width is infinite, resulting in a conclusion that orthogonal initialization can speed up training is a finite-width effect in the small learning rate regime. Then we find that during training, the NTK of infinite-width network with orthogonal initialization stays constant theoretically and varies at a rate of the same order as Gaussian ones empirically, as the width tends to infinity.

iii. Conduct studies of optimization and generalization of deep linear neural networks with a large learning rate. We characterize the implicit bias effect of deep networks for binary classification using the logistic loss in the large learning rate regime. It was found there is a learning rate regime with a large stepsize named the catapult phase, where the loss grows at the early stage of training and eventually converges to a minimum that is flatter than those found in the small learning rate regime. We claim that depending on the separation conditions of data, the gradient descent iterates will converge to a flatter minimum in the catapult phase. We rigorously prove this claim under the assumption of degenerate data by overcoming the difficulty of the non-constant Hessian of logistic loss and further characterize the behavior of loss and Hessian for non-separable data. Finally, we demonstrate that flatter minima in the space spanned by non-separable data along with the learning rate in the catapult phase can lead to better generalization empirically.

iv. Conduct studies of expressivity and trainability of ultra-wide and deep graph convolutional networks on the node classification task. Graph convolutional networks (GCNs) and their variants have achieved great success in dealing with graph-structured data. However, it is well known that deep GCNs will suffer from over-smoothing problem, where node representations tend to be indistinguishable as we stack up more layers. We demonstrate these characterizations by studying the Gaussian Process Kernel(GPK) and Graph Neural Tangent Kernel (GNTK) of an infinitely-wide GCN, corresponding to the analysis on expressivity and trainability, respectively. We formulate the asymptotic behaviors of GNTK in the large depth, which enables us to reveal the dropping trainability of wide and deep GCNs at an exponential rate. Additionally, we extend our theoretical framework to analyze residual connection-resemble techniques. We found that these techniques can mildly mitigate exponential decay, but they failed to overcome it fundamentally. Finally, all theoretical results in this work are corroborated experimentally on a variety of graph-structured datasets.

1.3 Thesis Organization

This thesis studies the expressivity, optimization and generalization properties of over-parameterized neural networks, ranging from deep dropout networks, orthogonally-initialized networks to graph neural networks. This dissertation contains seven chapters which are organized as follows:

- *Chapter 1*: This chapter introduces the foundation of deep learning with a brief introduction to several neural networks and advanced techniques which are widely used. Later on, motivations and contributions are demonstrated.
- *Chapter 2*: This chapter presents a survey of ultra-wide networks, including the mean-field theory on Gaussian process kernel before training, the neural tangent kernel on Gaussian process with gradient descent training, and the effect of implicit bias on generalization.
- *Chapter 3*: The mean-field theory for deep dropout networks is derived in this chapter. The results for gradient back-propagation are formulated by breaking the gradient independence assumption. Moreover, enormous simulations on the trainable depth with different hyper-parameters with a more precise empirical formula that describes the trainable length are given.
- *Chapter 4*: This chapter presents a study of optimization of orthogonally initialized networks. The theoretical derivation of the neural tangent kernel (NTK) for orthogonal initialization is illustrated. Furthermore, the theoretical results in the NTK regime are demonstrated in the formulation of lemmas and theorems. Finally, an analysis of empirical results outside the NTK regime is included.
- *Chapter 5*: This chapter discusses the implicit bias of ultra-wide linear networks with logistic loss with a large learning rate. The derivation and experi-

ments on the linear networks with different separation conditions are described in the form of theorems and figures. In conclusion, the risk will converge to a flatter minimum with a better performance in the large learning rate regime.

- *Chapter 6*: The dynamics of infinitely-wide graph convolutional networks (GCNs) are characterized in terms of depth are presented in this chapter. Firstly, the over-smoothing problem for deep GCNs is introduced with several techniques that can deepen GCN. Moreover, theoretical characterization of expressivity and trainability of ultra-wide GCNs are presented. With an extension analysis on the techniques for deepening GCN, the conclusion is given with numerical support.

- *Chapter 7*: A brief summary of the complements and room for improvement of the works presented in the previous chapters are given in the final chapter. Plans for the development of future works are discussed as well.

Chapter 2

Literature Review

2.1 Mean-Field Theory

In deep learning theory, there is a branch of study working on the expressivity of deep networks through infinitely-wide networks [22, 23, 24, 25, 26, 27, 28, 29]. The mean field theory utilize the infinite-width limit to make each neuron in the same layer i.i.d. distribution according to the central limit theory (CMT). Based on this observation, we can study how the correlation information between different inputs evolve through deep networks.

As stated before, the information propagation in a fully-connected network is governed by,

$$z_i^l = \sum_j W_{ij}^l y_j^{l-1} + b_i^l, \quad y_i^l = \phi(z_i^l), \tag{2.1}$$

where ϕ is the activation function. Upper index used in this chapter denotes the depth of the neural networks and lower index indicates the index of weights matrix, hidden layer and bias index.

We want to understand the signal information propagation through such networks by adopting the mean-field theory assumption [22, 23]. As the signal propagates through the network, we track a quantity,

$$q_{aa}^l = \frac{1}{N} \sum_{n=1}^{N} (z_{i;a}^l)^2. \tag{2.2}$$

where we denote a *single* input $x_{i;a}$ and a *pair* of inputs $x_{i;a}$ and $x_{i;b}$. As we increase N_l, this empirical distribution of hidden neurons converges to a Gaussian distribution with the help of the central limit theorem. By implementing the Gaussian property into the quantity q_{aa}^l, we can obtain a recursive expression:

$$q_{aa}^l = \sigma_w^2 \int \mathcal{D}z \phi^2(\sqrt{q_{aa}^{l-1}}z) + \sigma_b^2, \tag{2.3}$$

where $\int \mathcal{D}z = \frac{1}{\sqrt{2\pi}} \int dz e^{-\frac{1}{2}z^2}$. The properties of length propagation of a single input can be determined by finding the results of fixed-points.

We define the correlation between the pair of inputs,

$$q_{ab}^l = \frac{1}{N} \sum_{n=1}^{N} (z_{i;a}^l z_{i;b}^l). \tag{2.4}$$

Again, according to the central limit theory, we derive a recursive relation for q_{ab}^l,

$$q_{ab}^l = \sigma_w^2 \int \mathcal{D}z_1 \mathcal{D}z_2 \phi(u_1)\phi(u_2) + \sigma_b^2, \tag{2.5}$$

where

$$u_1 = \sqrt{q_{aa}^{l-1}} z_1, \quad u_2 = \sqrt{q_{bb}^{l-1}} (c_{ab}^{l-1} z_1 + \sqrt{1 - (c_{ab}^{l-1})^2} z_2) \tag{2.6}$$

with

$$c_{ab}^l = q_{ab}^l / \sqrt{q_{aa}^l q_{bb}^l}, \tag{2.7}$$

We can analyze the behavior of q_{aa}^l and c_{ab}^l, as they measure the depth the information can propagate through neural networks. As a result of studying the fixed-point property in fully-connected networks, it is shown that there is a phase transition in the σ_w and σ_b plane. In the chaotic phase, a pair of inputs end up asymptotically decorrelated. Conversely, in the ordered phase, the fixed point is stable, in which a pair of inputs end up asymptotically correlated.

The mean-field theory has been applied to different network architectures, including CNNs [30], RNNs [31], Residual networks [7], Batch normalization [8], LSTM [32], Graph Neural Network [33], and GRUs [34]. These networks have been investigated by [26, 27, 25, 28, 35], respectively, which form a large family of the mean-field theory for deep neural networks.

2.2 Neural Tangent Kernel

While mean-field theory studies the Gaussian process kernel of infinitely-wide networks at initialization, neural tangent kernel (NTK) are can describe the dynamics of corresponding networks during training. In the infinite-width limit, NTK converges to a limiting kernel before training and remains the same during training in the infinite-width limt, thus can provide a convergence guarantee for over-parameterized networks [36, 36, 37, 38, 39, 40, 41, 42, 43]. Besides, [40, 38, 41, 42] have proven the same proprieties of NTK and global convergence of deep networks in a few different ways.

Suppose there is a FCN with L Layer. The width are $n_0, \cdots, n_L$, and θ is the set of weights W and biases b of the network. One thing to note is that we refer to the factor $\frac{1}{\sqrt{n_l}}$ before the weights, which we call NTK parameterization. The aim is to prevent the divergence of NTK.

Considering the training of the network, $\partial_t f_\theta(t) = -\nabla_{\Theta^{(L)}} C|_{f_\theta(t)}$, where C is cost function. The NTK is defined as,

$$\Theta^{(L)}(\theta) = \sum_p \partial_{\theta_p} f^{(L)}(\theta) \otimes \partial_{\theta_p} f^{(L)}(\theta), \tag{2.8}$$

where p is the index of parameters; f is the output function.

It is shown that NTK tends to a limiting kernel at initialization. The usual way to prove it is by induction. Firstly, when the number of layers in the network is one, then NTK has no limit, and the result can be obtained by taking the derivative directly. Then we sssume that the L-layer network satisfies the tendency to a fixed kernel. In this step, we divide the network parameters into two parts. The first part is the parameters of the former L layer. In this part we need to use the law of large numbers. The second part is parameter at layer L. In this part we use the results of the infinitely wide network at initialization. Add the first part and the second part, and we get the recursion formula of NTK.

Researches utilize NTK to obtain a generalization bound for deep networks [44, 45]. NTK has been applied to a various of architecture and obtained a wealth of results, such as orthogonal initialization [46], convolutions [38], graph neural networks (GNTK) [47], attention [48], and see [49] for a summary. Especially, GNTK helps us to understand how GNN learns a class of smooth functions on graphs [47] and how GNN extrapolates differently from NN [50].

Although NTK can describe the infinite wide network well, it will fail in the finite wide network, which is the difference between the neural network and the kernel method. In a paper [51], the finite width and finite depth correction of NTK in fully connected ReLU networks are studied. Another work [40] studies the dynamic NTK of finite width and depth fully connected neural network. In their work, an infinite order ordinary differential equation, namely neural thumbs (NTH), is derived to describe the optimization properties of neural networks with finite width and depth.

2.3 Implicit Bias in Deep Learning

Since the seminal work from [52], implicit bias has led to a fruitful line of research. There are a line of works which treated linear predictors [53, 54, 55]; deep linear networks with a single output [56, 57, 58] and multiple outputs [59]; homogeneous networks (including ReLU, max pooling activation) [60, 61, 62]; ultra-wide networks [63, 64]; matrix factorization [65].

Specifically, consider a linear predictor using gradient descent trained on linearly separable data can yield the convergence in the direction of hard margin and the norm of weight tends to infinity [52]. If the dataset changes to a general set consist of both linear separable and strong convex, then the outputs are biased to follow a unique ray defined by the data [54]. When the linear predictor goes deeper, which is deep linear networks, the maximum margin will remain [56, 58] with the emergence of alignment between the weight in each layer [58].

Consider the dataset $\{x_n, y_n\}_{n=1}^N$, where $x \in \mathbb{R}^d$, and $y \in \{-1, 1\}$. This is a common way of representing binary classification datasets. The expression for loss is,

$$\mathcal{L}(w) = \sum_{n=1}^{N} \ell(y_n w^T x_n) \tag{2.9}$$

As you can see from this, our network has only one linear layer, $f_n = w^T x_n$, no hidden layer, and no nonlinear activation function. For the sake of analysis, we have two hypotheses: (1) Data sets are linearly separable, (2) The loss function is positive, differentiable, and monotonically decreasing to zero. Under gradient descent, the results are as follows,

$$\begin{aligned} &\lim_{t\to\infty} \mathcal{L}(w(t)) = 0 \\ &\lim_{t\to\infty} \|w(t)\| = \infty \\ &\lim_{t\to\infty} w(t)^T x_n = \infty \end{aligned} \tag{2.10}$$

It can be seen from this that eventually the norm of weight will tend to infinity. These three conclusions above are explained by the following formula,

$$w_*^T \nabla \mathcal{L}(w) = \sum_{n=1}^{N} \ell'(w^T x_n) w_*^T x_n \tag{2.11}$$

where w_*^T is the solution of the weight w. We know that for any $w_*^T x_n$ is greater than zero, and since loss is monotonically decreasing, the above formula cannot be satisfied with the weight of finite size. So we know that the norm of weights tends to infinity in the end, and then we infer that loss converges to the global minimum.

While we do know the preliminary result, many of the details are still being worked out. For example,

$$w(t) = \hat{w} \log t + \rho(t), \quad \lim_{t\to\infty} w(t)/\|w(t)\| = \frac{\hat{w}}{\|\hat{w}\|} \tag{2.12}$$

where $\hat{w} = \operatorname{argmin} \|w\|^2 \quad \text{s.t.} \quad w^T x_n \geq 1$. And $\rho(t)$ has the maximum magnitude of $\|\rho(t)\| = O(\log\log(t))$. Those of you familiar with SVM may know that $\hat{w}$ is the solution to maximum margin. And that's the amazing thing about our problem. The final solution is not a straight line that separates the data casually, but a solution that meets the SVM maximum margin.

Chapter 3

Mean Field Theory for Deep Dropout Networks

3.1 Introduction

Deep learning have achieved exceptional results in a range of fields since its inception [3]. More recently, we have witnessed several signs of progress made using mean field theory [22, 23, 24] in deep learning. The mean field considers networks after random initialization, whose weights and biases were i.i.d. Gaussian distributed, and the width of each layer tends to infinity. As a result of studying signal propagation under mean field theory, an order-to-chaos expressivity phase transition split by a critical line has been found [22]. Later, how parameter initialization may impact the gradient of backpropagation was studied, and the conclusion that the ordered and chaotic phases correspond respectively to regions of vanishing and exploding gradient respectively was shown [23]. The results were also equivalently applied to networks with or without dropout.

The main contribution of the mean field theory for random networks is that it shows the existence of depth scales that limit the maximum depth of signal propagation and gradient backpropagation. Practically, the result is to show a hypothesis that random networks may be trained precisely when information can travel through them. Thus, the depth scales provide bounds on how deep a network may be trained for a specific choice of hyper-parameters [23]. This ansatz was tested and verified by practical experiments on MNIST and CIFAR10 dataset with wide width fully-connected networks [23], deep dropout networks [23], and residual networks [25].

However, the mean field calculation for the gradient is based on the so-called *gradient independence assumption*, which states that the weights used during feed forward are drawn independently from the ones used in backpropagation. This is in an effort to make the calculation of gradient feasible regardless of the choice of activation functions. This assumption was later formulated explicitly [25] for residual networks and was illustrated in a review [39]. While it enjoys the correct prediction of gradient dynamics in some cases, our experiments show that under the condition in which the weights in feed-forward are carried over to its backpropagation, the length that gradients can backpropagate for a *single* and a *pair* of inputs are governed by the same depth scale on deep dropout networks instead.

By further studying the mean and variance of gradient statistics metrics on deep dropout networks, we show an emergence of universality for the relationship between the mean and variance. This universality exists regardless of the choice of hyper-parameters, including dropout rate and activation function. After summarizing the theoretical results about the trainable length of deep dropout networks governed by maximum depth of signal propagation and gradient backpropagation, we perform a series of experiments to investigate it. Empirically, we find a more precise way to describe the maximum trainable length for deep dropout networks, compared with the original results [23].

3.2 Background

In this section, we review the mean field theory for deep dropout networks. We give the main definitions, setup, and notations, and introduce the results of theory for random networks at initialization, including signal feed-forward and gradient backpropagation, respectively.

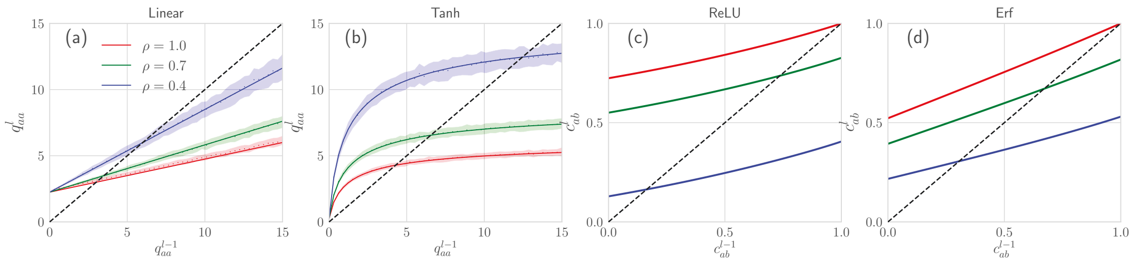

Figure 3.1 : The iterative squared length mapping of Equation (3.2) and Equation (3.4) with different activations and dropout rates. (a) q^l_{aa} in linear network at $\sigma_w = 0.5$ and $\sigma_b = 1.5$. Theoretical results match well with the simulations within a standard error (shadow). Different color correspond to different dropout rates: $\rho = 1$ is red, $\rho = 0.7$ is green, and $\rho = 0.4$ is blue. (b) The iterative length map of q^l_{aa} in Tanh network at $\sigma_w = 2.5$ and $\sigma_b = 0.5$. (c) The iterative length map of c^l_{ab} in ReLU network at $\sigma_w = 0.9$ and $\sigma_b = 0.5$. Only intersection of network at $\rho = 1$ (red) is $c^*_{ab} = 1$, the others are $c^*_{ab} < 1$. (d) The iterative length map of c^l_{ab} in Erf network at $\sigma_w = 0.9$ and $\sigma_b = 0.5$. Again, $c^*_{ab} = 1$ only holds at $\rho = 1$.

3.2.1 Feed Forward

In the context of a fully-connected, feedforward, untrained, and dropout neural network of depth L with layer width N. We denote synaptic weight and bias for the l-th layer by W^l_{ij} and b^l_i; pre-activations and post-activations by z^l_i and y^l_i respectively. Finally, we take the input to be $y^0_i = x_i$ and the dropout keep rate to be ρ. The information propagation in this network is governed by,

$$z^l_i = \frac{1}{\rho} \sum_j W^l_{ij} p^l_j y^{l-1}_j + b^l_i, \quad y^l_i = \phi(z^l_i), \tag{3.1}$$

where ϕ is the activation function and $p \sim \text{Bernoulli}(\rho)$. We adopt the mean field theory assumption [22, 23], where $W^l_{ij} \sim \mathcal{N}(0, \frac{\sigma^2_w}{N})$, $b^l_i \sim \mathcal{N}(0, \sigma^2_b)$, and the width N tends to infinite. Since the weights and biases are randomly distributed, these equations define a probability distribution on the pre-activations over an ensemble of untrained neural networks. Under the mean field approximation, z_i^l can be replaced by a Gaussian distribution with zero mean.

Consider a *single* input $x_{i;a}$, where the subscript a refers to the index of input. We define the length quantities $q_{aa}^l = \frac{1}{N}\sum_{n=1}^{N}(z_{i;a}^l)^2$, which is the mean squared pre-activations. According to the mean field approximation, the length quantity is described by the recursion relation,

$$q_{aa}^l = \frac{\sigma_w^2}{\rho}\int \mathcal{D}z\phi^2(\sqrt{q_{aa}^{l-1}}z) + \sigma_b^2, \tag{3.2}$$

where $\int \mathcal{D}z = \frac{1}{\sqrt{2\pi}}\int dze^{-\frac{1}{2}z^2}$ is the measure for a normal distribution. This equation describes how a single input evolves through a random neural network. To study the property of evolution, we investigate the fixed point at $q_{aa}^* \equiv \lim_{l\to\infty} q_{aa}^l$. One way to estimate the fixed point is to plot Equation (3.2) with the unity line, and the intersection is the fixed point. We show the result for Equation (3.2) with Linear dropout network and Tanh dropout network in Figure 3.1(a)(b). Note that the smaller the dropout rate ρ, the larger the fixed point value q_{aa}^*.

The propagation of a *pair* of inputs $x_{i;a}$ and $x_{i;b}$, where the subscript a and b refer to different inputs, can be studied by looking at the correlation between the two inputs after l layers. We definite this correlation quantity as $q_{ab}^l = \frac{1}{N}\sum_{n=1}^{N}(z_{i;a}^l z_{i;b}^l)$. Similarly, the correlation q_{ab}^l will be given by the recurrence relation,

$$q_{ab}^l = \sigma_w^2\int \mathcal{D}z_1\mathcal{D}z_2\phi(u_1)\phi(u_2) + \sigma_b^2, \tag{3.3}$$

where $u_1 = \sqrt{q_{aa}^{l-1}}z_1$ and $u_2 = \sqrt{q_{bb}^{l-1}}(c_{ab}^{l-1}z_1 + \sqrt{1-(c_{ab}^{l-1})^2}z_2)$, with

$$c_{ab}^l = q_{ab}^l/\sqrt{q_{aa}^l q_{bb}^l}. \tag{3.4}$$

This equation also have a fixed point at $c_{ab}^* \equiv \lim_{l\to\infty} c_{ab}^l$. It is known that $c_{ab}^* = 1$ when $\rho = 1$, while $c_{ab}^* < 1$ when $\rho < 1$ [23]. We show the result of Equation (3.4) on the ReLU and Erf dropout networks in Figure 3.1(c)(d), which demonstrate the main conclusion about fixed-point without ($\rho = 1$) and with ($\rho < 1$) dropout.

The main contribution of mean field theory for the fully-connected networks without dropout ($\rho = 1$) is that it presents a phase diagram, which is determined by a crucial quantity,

$$\chi_1 = \frac{\partial c_{ab}^l}{\partial c_{ab}^{l-1}} = \sigma_w^2 \int \mathcal{D}z[\phi'(\sqrt{q^*}z)]^2. \tag{3.5}$$

This quantity was firstly introduce by [22] to determine whether or not the $c_{ab}^* = 1$ is an attractive fixed point. When $\chi_1 < 1$, the fixed point is unstable. Conversely, when $\chi_1 > 1$, the fixed point is stable. Thus, the critical line $\chi_1 = 1$ separates two phases. One is the chaotic phase ($\chi_1 < 1$), where a pair of inputs end up asymptotically decorrelated, and the other is the ordered phase, in which a pair of inputs end up asymptotically correlated.

We give a comment on the difference between q_{aa}^l and c_{ab}^l here. The random networks under the infinite width limit setting can be viewed as the Gaussian processes, where q_{aa}^l and c_{ab}^l are the diagonal and non-diagonal elements of the compositional kernel[66], respectively. Intuitively, the non-diagonal element of the kernel measures the correlation between different data points while the diagonal component measures the information of one input itself.

The study of information propagation shows the existence of a depth-scales ξ_2, which represent the length of propagation of the following qualities:

$$|c_{ab}^l - c_{ab}^*| \sim e^{-l/\xi_2}. \tag{3.6}$$

where $\xi_2 = |1/\log \chi_2|$, according to the exponential rate of propagation, with $\chi_2 = \sigma_w^2 \int \mathcal{D}z_1\mathcal{D}z_2\phi'(u_1^*)\phi'(u_2^*)$, where $u_1^* = \sqrt{q_{aa}^*}z_1$ and $u_2^* = \sqrt{q_{bb}^*}(c_{ab}^*z_1 + \sqrt{1-(c_{ab}^*)^2}z_2)$. Intuitively, the depth-scales ξ_2 measures how far can correlation between two different inputs survives through the network.

3.2.2 Back Propagation

There is a duality between the forward propagation of signals and the backpropagation of gradients. Given a loss E, we have

$$\frac{\partial E}{\partial W_{ij}^l} = \frac{p_j^l}{\rho}\phi(z_j^{l-1})\delta_i^l, \quad \delta_i^l = \phi'(z_i^l)\frac{p_i^{l+1}}{\rho}\sum_j \delta_j^{l+1}W_{ji}^{l+1}, \tag{3.7}$$

where $\delta_i^l = \frac{\partial E}{\partial z_i^l}$. We define the metric of gradient for both a *single* input and a *pair* of inputs cases:

$$g_{aa}^l \equiv \frac{1}{N^2}\sum_{ij}(\frac{\partial E_a}{\partial W_{ij}^l})^2, \quad g_{ab}^l \equiv \left|\frac{1}{N^2}\sum_{ij}\frac{\partial E_a}{W_{ij}^l}\frac{\partial E_b}{W_{ij}^l}\right|. \tag{3.8}$$

Within mean field theory, the scale of fluctuations of the gradient of weights in a layer will be proportional to $\tilde{q}_{aa}^l \equiv \mathbb{E}\left[\delta_{i;a}^l\delta_{i;a}^l\right]$, which can be written as, $g_{aa}^l \propto \tilde{q}_{aa}^l$ [23]. On the other hand, the correlation between gradients of a pair of inputs will be proportional to $\tilde{q}_{ab}^l \equiv \mathbb{E}\left[\delta_{i;a}^l\delta_{i;b}^l\right]$, namely, $g_{ab}^l \propto \tilde{q}_{ab}^l$.

In order to work out the recurrence relation for $\tilde{q}_{aa}^l$ and $\tilde{q}_{ab}^l$, an approximation was made [23], named *gradient independence assumption*, that the weights used during forward propagation are drawn independently from the weights used in backpropagation. In this way, the term $\phi'(z_i^l)$, δ_j^{l+1} and W_{ji}^{l+1} in Equation (3.7) can be addressed independently. Then, the recurrence behavior of $\tilde{q}_{aa}^l$ and $\tilde{q}_{ab}^l$ are achieved,

$$\tilde{q}_{aa}^l = \tilde{q}_{aa}^{l+1}\chi_1, \quad \tilde{q}_{ab}^l = \tilde{q}_{ab}^{l+1}\chi_2. \tag{3.9}$$

where we redefine the quantity χ_1 for the dropout networks,

$$\chi_1 = \frac{\sigma_w^2}{\rho}\int \mathcal{D}z[\phi'(\sqrt{q^*}z)]^2, \tag{3.10}$$

Equation (3.9) has an exponential solution with,

$$\tilde{q}_{aa}^l = \tilde{q}_{aa}^L e^{-(L-l)/\xi_1}, \quad \tilde{q}_{ab}^l = \tilde{q}_{ab}^L e^{-(L-l)/\xi_2}. \tag{3.11}$$

Similar to the signal propagation, gradient backpropagation can limit the trainable length in the way of gradient vanishing or gradient exploding, which is measured by the depth-scales ξ_1 and ξ_2.

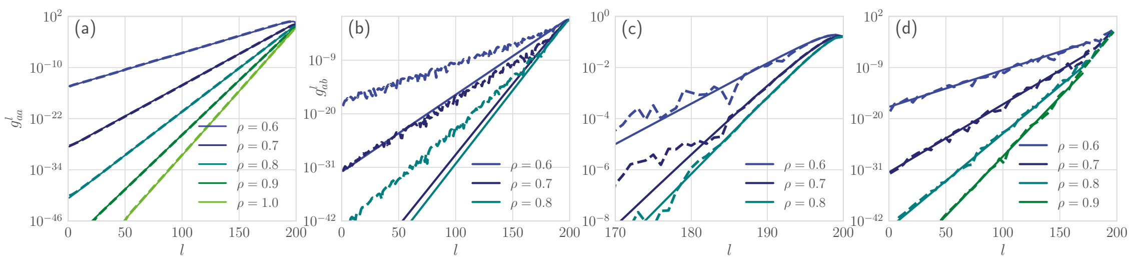

Figure 3.2 : Theoretical calculations versus network simulations for metric of gradient. (a) g^l_{aa} as a function of layer l, for a 200 layers random linear network with $\sigma_w^2 = 0.5$ and $\sigma_b^2 = 0.1$. (b) g^l_{ab} as a function of layer l. Theoretical calculations (solid lines) fail to predict empirical simulations (dashed lines). (c) g^l_{ab} as a function of layer l in the range of length $l = 170 - 200$. Theoretical calculations (solid lines) can predict empirical simulations (dashed lines) in the few last layers. (d) g^l_{ab} as a function of layer l. The solid lines are $g^l_{ab} \propto \chi_1^{L-l}$ for different ρ. Theoretical calculations failed to predict empirical simulations (dashed lines).

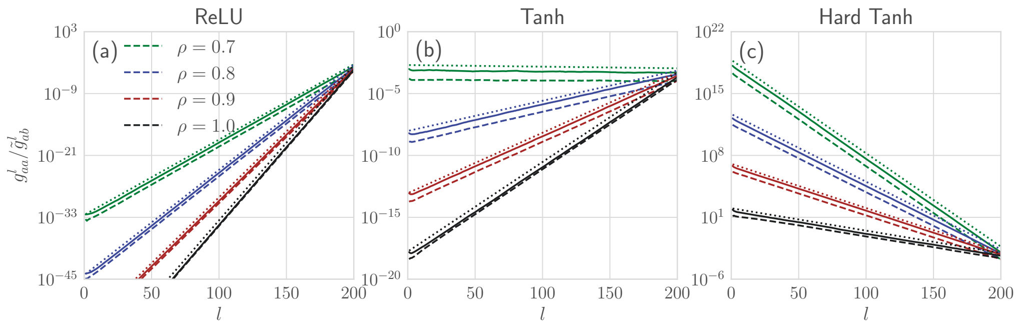

Figure 3.3 : The metric of gradient with one and two different inputs, g^l_{aa} (solid lines), $\tilde{g}^l_{ab}$ (dashed lines), and $g^l \propto \chi_1^{L-l}$ (dotted lines) as a function of layer l with different activation. (a) ReLU network with $\sigma_w^2 = 1.0$ and $\sigma_b^2 = 0.1$. (b) Tanh network with $\sigma_w^2 = 1.4$ and $\sigma_b^2 = 0.1$. (c) Hard Tanh network with $\sigma_w^2 = 1.4$ and $\sigma_b^2 = 0.1$.

3.3 Gradient Backpropagation

In this section, we first calculate the metrics of gradient g_{aa} and g_{ab} theoretically without the gradient independence assumption on linear dropout networks. We then conduct a series experiment for metrics of gradient on deep dropout networks, including non-linear cases. Finally, we show an emergence of a universal relationship between mean and variance of metrics of the gradient.

3.3.1 Breaking the gradient independence assumption

We follow the fact that weights used in a feed-forward are carried over to its back-propagation. We first provide a theoretical treatment to the linear networks in which we assume the output is the last layer of network $y_i^L = z_i^L$ without soft-max. The labels of data are set to be zeros, and the loss is the mean squared loss.

For space reason, we omit details of the calculation and present the primary analysis and final results here. The main problem is that we should expand δ_j^{l+1} when calculating δ_i^l in Equation (3.7), since δ_j^{l+1} can correlate with W_{ji}^{l+1} without the gradient independence assumption. Using g_{aa}^l as an example, we perform:

i. Starting from the last layer L, we compute $\delta_{i,a}^L = \frac{\partial E_a}{\partial z_{i,a}^L} = 2z_{i,a}^L$ and use this result to compute $g_{aa}^L = \mathbb{E}\left[(\frac{p_{j,a}^L}{\rho} z_{j,a}^{L-1}\delta_{i,a}^L)^2\right]$.

ii. Then we compute $g_{aa}^{L-1} = \mathbb{E}\left[(\frac{p_{j,a}^{L-1}}{\rho} z_{j,a}^{L-2}\delta_{i,a}^{L-1})^2\right]$ with the result of $\delta_{i,a}^{L-1} = \frac{\partial E_a}{\partial z_{i,a}^L}\frac{\partial z_{i,a}^L}{\partial z_{i,a}^{L-1}} = \sum_j 2z_{j,a}^L \frac{p_{i,a}^L}{\rho} W_{ji}^L$ and $z_i^L = \frac{1}{\rho}\sum_j W_{ij}^L p_j^l z_j^{L-1} + b_i^L$.

iii. By parity of reasoning, we obtained the results for the penultimate layer g_{aa}^{L-2}. The correlation between terms that contain W_{ij}^L and W_{ij}^{L-1} are considered.

iv. As the index of the layer decreases, the amount of calculation becomes larger and larger. Thus we use the induction method to achieve the results for the left layers.

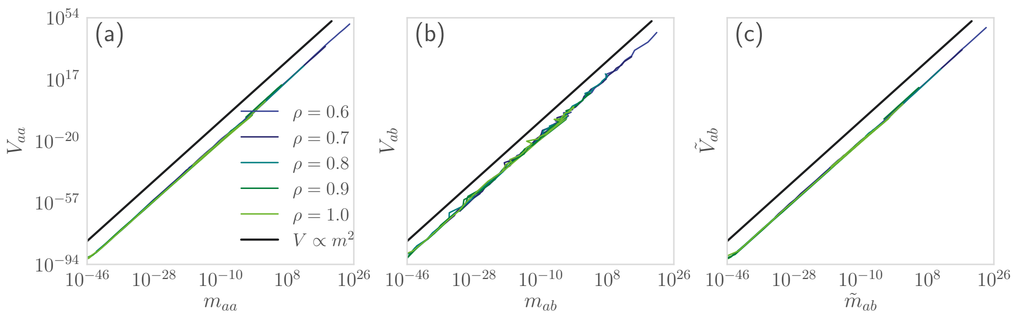

Figure 3.4 : Universal relationship between variance and mean of g^l_{aa}, g^l_{ab}, and $\tilde{g}^l_{ab}$, on the 200 layers and width $N = 500$ random dropout networks. Different color represents a different dropout rate. The black line is the function of $V \propto m^2$. (a) V^l_{aa} as a function m^l_{aa}. (b) V^l_{ab} as a function of m^l_{ab}. (c) $\tilde{V}^l_{ab}$ as a function of $\tilde{m}^l_{ab}$. All the curves regarding different activations collapse to a line, and the power coefficient of all curves is consistent with 2.

We use the same approach to derive the result for g^l_{ab}. As a result, we have,

$$\begin{aligned} g^l_{aa} &= 4(\frac{q^*_{aa}}{\rho})^2(\frac{\sigma^2_w}{\rho})^{L-l}[\rho + \sum_{j=1}^{L-l}(\frac{\sigma^2_w}{\rho})^j], \\ g^l_{ab} &= 4(q^*_{ab})^2(\sigma^2_w)^{L-l}[1 + \sum_{j=1}^{L-l}(\frac{\sigma^2_w}{\rho^2})^j]. \end{aligned} \tag{3.12}$$

By analyzing the first formula of Equation 3.12, we find that $g^l_{aa} = g^{l+1}_{aa}\chi_1$. This can be better observed by dividing the expression related to layer l into two factors: one is $(\frac{\sigma^2_w}{\rho})^{L-l}$, and the other is $\sum_{j=1}^{L-l}(\frac{\sigma^2_w}{\rho})^j$. The first factor accounts for $g^l_{aa} = g^{l+1}_{aa}\chi_1$, where $\chi_1 = \frac{\sigma^2_w}{\rho}$ for linear dropout networks. And second factor will be stable after several layers starting from the last layer L due to $\sigma^2_w < \rho$. We show an excellent match between the theoretical calculation above with simulation using networks with width $N = 500$ and layer $L = 200$ over 100 different instantiations of the network in Figure 3.2(a).

Despite the successful prediction of theoretical calculation for g^l_{aa}, our theoretical results for g^l_{ab} only hold on the case of $\rho = 1$ while fail to predict the experimental

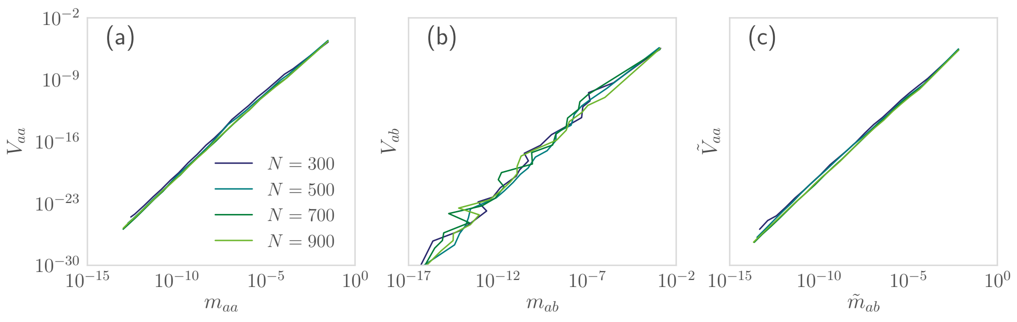

Figure 3.5 : Universal relationship between variance and mean of g^l_{aa}, g^l_{ab}, and $\tilde{g}^l_{ab}$, on the 200 layers, Tanh random dropout networks with $\rho = 0.9$. All the curves regarding different width collapse to a line. Different color represents a different network width. (a) V^l_{aa} as a function m^l_{aa}. (b) V^l_{ab} as a function of m^l_{ab}. (c) $\tilde{V}^l_{ab}$ as a function of $\tilde{m}^l_{ab}$.

behavior except for last few layers when $\rho < 1$, as shown in Figure 3.2(b)(c). After a few layers from L, the variances began to increase dramatically as shown in Figure 3.2(c). We noticed that unlike the case of computing q^l_{ab}, using χ_2 is prohibitive for computing g^l_{ab}. On the other hand, we try a function regarding χ_1 to fit g^l_{ab}, and find an interesting observations that χ_1 is a much more compatible term for g^l_{ab}, i.e, $g^l_{ab} = g^{l+1}_{ab}\chi_1$. This is demonstrated in Figure 3.2(d).

The incompatible phenomenon between theoretical calculation and experimental results for g^l_{ab} begins with the emergence of variance, as shown in Figure 3.2(c). One possible explanation is that the emergence of variance is caused by limited network length. Thus, we can reduce this variance by increasing network length only. To check if this explanation works, we further investigate the relationship between variance and mean of g^l_{ab} with different network widths N. The answer is that $g^l_{ab} = g^{l+1}_{ab}\chi_1$ holds regardless of the finite width. We will demonstrate it in the next section.

After studying the gradient behavior at the linear networks, A series of experiments is conducted on the nonlinear case since theoretical formulations for nonlinear activation or with the soft-max layer is intractable. We firstly use g^l_{ab} as the metric of gradient and find it has a huge variance when $\rho < 1$. This is because the element of the gradient matrix with a pair of inputs can be either negative or positive. To find a metric with low variance, we consider the metric $\tilde{g}^l_{ab} \equiv \frac{1}{N^2} \sum_{ij} |\frac{\partial E_a}{W^l_{ij}} \frac{\partial E_b}{W^l_{ij}}|$ whose elements are all positive. Besides, it is the ℓ_1 norm of the gradient matrix.

We plot g^l_{aa} and $\tilde{g}^l_{ab}$ as a function of l in Figure 3.3. Interestingly, our simulations show that both g^l_{ab} and $\tilde{g}_{ab}$ are governed by χ_1 in a range of activations. Thus we make a conjecture that the relation,

$$g^l_{aa} = g^{l+1}_{aa}\chi_1, \quad g^l_{ab} = g^{l+1}_{ab}\chi_1, \tag{3.13}$$

holds on deep dropout networks.

Table 3.1 : Summary of depth-scale for theoretical results i.e. signal propagation and gradient backpropagation, and empirical results under different condition or assumption.

Summary	feed-forward propagation		gradient backpropagation		empirical results
metric	q_{aa}	q_{ab}	g_{aa}	g_{ab}	
realistic condition (our work)	-	ξ_2	ξ_1	ξ_1	$\min\{12\xi_1, 12\xi_2\}$
independent assumption [23]	-	ξ_2	ξ_1	ξ_2	$6\xi_2$

3.3.2 Emergence of universality

We have studied three statistical metrics of the gradient, i.e. g_{aa}, g_{ab}, and $\tilde{g}_{ab}$ using their mean value. Inevitably, the variance of these metrics can give us essential information about the gradient. To do this, we performed a series of experiments

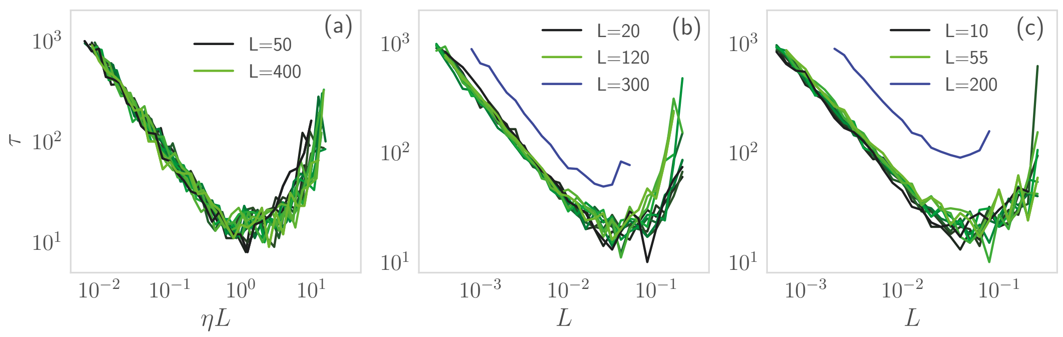

Figure 3.6 : The relation between steps and the learning rate η. (a) Network without dropout, colors reflect different network depth L from 50 (black) to 400 (green). (b) Network with dropout $\rho = 0.99$, colors reflect different network depth L from 20 (black) to 120 (green), additional $L = 300$ is colored blue for comparison. Curves with $L \leq 120$ collapse to a universal curve without any re-scale. (c) Network with dropout $\rho = 0.98$, colors reflect different network depth L from 10 (black) to 55 (green), additional $L = 200$ is colored blue for comparison. Curves with $L \leq 55$ collapse to a universal curve without any re-scale.

to obtain the mean and variance of g_{aa}, g_{ab} and $\tilde{g}_{ab}$ with different activation and different network width N.

First, we show the relationship between variance and mean of the metric of gradient with different activations, including Linear, ReLU, Tanh, and Hard Tanh. We denote the mean of g_{aa}, g_{ab} and $\tilde{g}_{ab}$ as m^l_{aa}, m^l_{ab}, and $\tilde{m}^l_{ab}$, while naming the variance as V^l_{aa}, V^l_{ab}, and $\tilde{V}^l_{ab}$ respectively. We show the variance as a function of mean in Figure 3.4, and find the emergence of universality between the variance and mean regardless of dropout rate and choice of activation for g_{aa}, g_{ab}, and $\tilde{g}^l_{ab}$.

The plot of variance as a function of mean shows a power-law between them since it is like a straight line in the log-log plot. To estimate the power, we use a simple equation $V \propto m^2$ to compare with the experiment results. Surprisingly, all three curves are consistent with $V \propto m^2$. Thus we make a conjecture that the universal power coefficient between the variance and mean is 2.

Then, we investigate the relationship between variance and mean with different network width N and show the results in Figure 3.5. This time, we perform experiments on the $\rho = 0.9$ Tanh networks with different network width N. Again, the relationship between variance and mean satisfies universality, which means the Equation (3.13) does not depend on the network width of N.

We want to point out that we have performed the same investigation on q^l_{aa} and c^l_{ab}. However, we did not observe a similar universal relationship between variance and mean of q^l_{aa} and c^l_{ab}. This may occur due to the different behavior of q^l_{aa} (q^l_{ab}) and g^l_{aa} (g^l_{ab}). As Equation 3.6 shows, the mean of c^l_{ab} will converge to a fixed point after several layers, which means that the mean of c^l_{ab} will be stable in deeper layers. So, we won't expect a universal relation between the mean and the variance in this case.

In summary, we have tried all the parameter freedom that we can tune, the universal power coefficient between the variance and mean remains the same. We conclude that once the topological structure of the neural network is set, the power coefficient is universal.

3.4 Experiments

According to the theoretical results, during feed-forward, we expect that length-scale ξ_2 control the propagation of c^l_{ab}, while ξ_1 measure the number of layers that gradient metrics g^l_{aa} and g^l_{ab} can survive during backpropagation. However, [23] claimed that both networks with or without dropout networks have a limited trainable length, which is governed by the depth-scale ξ_2. As our experimental results show, which be demonstrated later, this statement is not exactly right. To summarize, we present the comparison for the length-scale between [23] and our work in

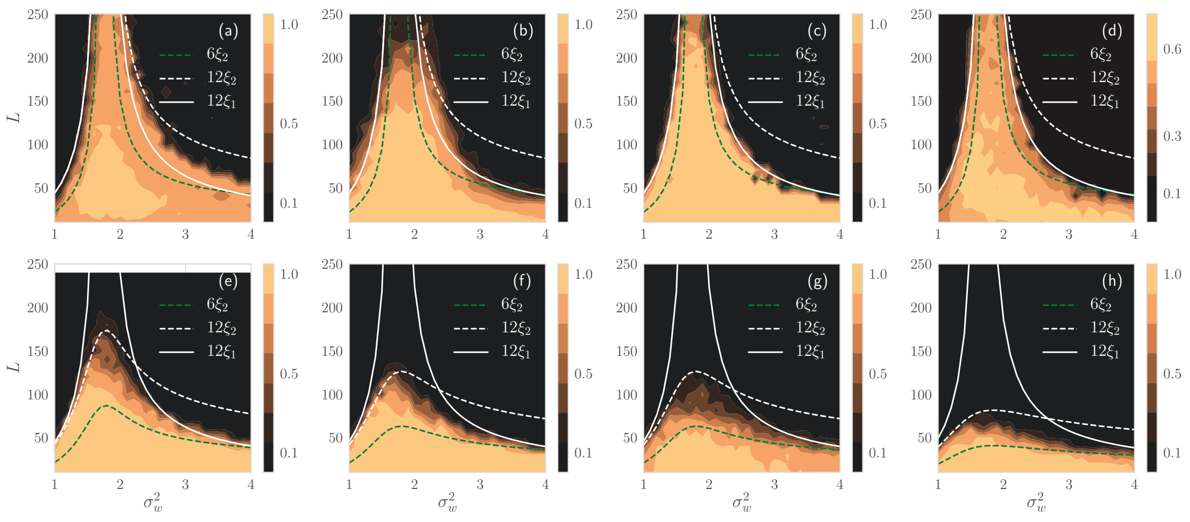

Figure 3.7 : The training accuracy for neural networks as a function of the depth L and initial weight variance σ_w^2 from a high accuracy (bright yellow) to low accuracy (black). Comparison is made by plotting $12\xi_1$ (white solid line), $6\xi_2$ (green dashed line), and $12\xi_2$ (white dashed line). (a) 2000 training steps of $\rho = 1$ network with Gaussian weights on the MNIST using SGD. (b) 1000 training steps of $\rho = 1$ network with Gaussian weights on the MNIST using RMSProp. (c) 2000 training steps of $\rho = 1$ network with Orthogonal weights on the MNIST. (d) 3000 training steps of $\rho = 1$ network with Orthogonal weights on CIFAR10. (e) 3000 training steps of $\rho = 0.99$ network with Orthogonal weights on the MNIST. (f) 3000 training steps of $\rho = 0.98$ network with Orthogonal weights on the MNIST using SGD. (g) 10000 training steps of $\rho = 0.98$ network with Gaussian weights on the MNIST. (h) 3000 training steps of $\rho = 0.95$ network with Orthogonal weights on the MNIST using SGD.

Table 3.1.

3.4.1 Training speed

Before investigating this problem, we study the relationship between training speed and choice of hyper-parameters. We confine the hyper-parameters at the critical line $\chi_1 = 1$ for the network with and without dropout and train networks of a range of length with width $N = 400$ for 10^3 steps with a batch size of 10^3 on the standard CIFAR10 dataset. Strictly speaking, $\chi_1 = 1$ is not the critical line when $\rho < 1$, since $\chi_2 < 1$. For learning rates of each network, we consider logarithmically spaced in steps 10^1. The τ is the steps required to obtain a accuracy threshold $p = 0.25$. We show the relation between steps τ and the learning rate η on the networks of dropout rate $\rho = 1.0, 0.99$, and 0.98 in Figure 3.6.

We find that for networks without dropout, there is a universal scaling $\tau = f_1(\eta L)$ between the steps and learning rate, where f_1 is a scaling function, as shown in Figure 3.6(a). Note that it is different to the result that $\tau/\sqrt{L} = f_1'(\eta L)$ in [24] where they use the standard CIFAR10 dataset augmented with random flips, crops, and so on. The difference may be caused by the pretreatment of the dataset in [24]. Besides, we study the networks with $\rho = 0.99$ and $\rho = 0.98$, and find that the scaling $\tau = f_2(\eta)$ can be kept under a limited length $L = 120$ for $\rho = 0.99$ and $L = 55$ for $\rho = 0.98$, as shown in Figure 3.6(b) and (c) respectively.

3.4.2 Trainable length

Now we study the problem of trainable length. We consider random networks of depth $10 \leq L \leq 250$, and $1 \leq \sigma_w^2 \leq 4$ with $\sigma_b^2 = 0.05$. We train these networks by Stochastic Gradient Descent (SGD) and RMSProp on MNIST and CIFAR10 with Gaussian and Orthogonal weights, which can be seen as another variant of weight initialization in the mean field theory [24]. We perform four experiments on the network without dropout ($\rho = 1$) with different datasets, optimizer, and learning rate to conduct a comprehensive study, and plot the results in Figure 3.7(a)-(d). Besides, four experiments are conducted on the dropout networks ($\rho < 1$), and results are shown in Figure 3.7(e)-(h). The trainable length can be obtained according to the relationship between ξ and χ, which are $\xi_1 = |1/\log \chi_1|$ and $\xi_2 =$

$|1/\log \chi_2|$. We color in bright yellow the training accuracy that networks achieved as a function of σ_w^2 and L for different dropout rates. From the heatmap, we can observe a boundary in which accuracy began to drop. We noticed that there are two boundaries, left and right. In order to show its relationship with ξ_1 and ξ_2, we superimpose them onto the heatmap.

In figure 3.7(a), we use the same learning rate and optimizer as those in Figure 5(a)-(c) of [23]. We use a learning rate of 10^{-3} for SGD when $L \leq 200$, and 10^{-4} for larger L. From the plot, we find the $6\xi_2$ underestimates the scope of train-ability in the σ_w^2-L plane, while $12\xi_1$ is more compatible with the experimental result. We note the phenomenon that $6\xi_2$ underestimates the scope of train-ability also happened in Figure 5(b)(c) of [23]. In figure 3.7(b), we adopt the same learning rate and optimizer as those in Figure 5(d) of [23], where we use a learning of 10^{-5} and RMSProp optimizer. Here, the only difference is that we use 1000 training steps instead of 300 training steps in [23]. According to the simulation result, $12\xi_1$ (solid line) and ξ_2 (dashed line) are identical on the left boundary, while they differ on the right side. We make a comparison between $12\xi_1$ and $12\xi_2$, and find that $12\xi_1$ has a much better argument with the trainable length while $12\xi_2$ overrates the trainable length on the right side.

Based on the analysis of Figure 3.7(a)(b), we may conclude that $12\xi_1$ can be used to measure the maximum trainable length of the network without dropout. We further reinforce this conclusion by performing experiments on different learning rates, weight initialization, and datasets. In figure(c), we use orthogonal weight initialization. In figure(d), we perform experiment on CIFAR-10 dataset and adopt a learning rate of $\eta = c/L$, where c is constant. These learning rates were selected for the reason that each learning rate can lead to the fast step to a certain test accuracy at $\chi_1 = 1$, as shown in Figure 3.6. In a word, we attribute the maximum trainable length to $L \leq \min\{12\xi_1, 12\xi_2\} = 12\xi_1$, where the relation $\xi_1 \leq \xi_2$ holds on the network without dropout.

Furthermore, we consider the dropout case in Figure 3.7(e)-(h). We have studied three different dropout rate: $\rho = 0.99$ (Figure 3.7(e)), $\rho = 0.98$ (Figure 3.7(f)(g)), and $\rho = 0.95$ (Figure 3.7(h)). We find that both ξ_1 and ξ_2 have connections to the trainable length: the networks appear to be trainable when $L \leq \min\{12\xi_1, 12\xi_2\}$. Networks on the left side are influenced by $12\xi_2$ while they are constrained by the $12\xi_1$ on the right size. Note that the formula $L \leq \min\{12\xi_1, 12\xi_2\}$ is valid in the no dropout case as discussed above. To conclude, we show an improved relationship between maximum trainable length and length scale ξ_1 and ξ_2 than [23]. This conclusion that both ξ_1 and ξ_2 have connections to the trainable length instead of only ξ_2 [23] is more compatible with the theoretical results.

3.5 Discussion

In this chapter, we have investigated the dropout networks by calculating its statistical metrics of gradient during the backpropagation at initialization and conjecture that both gradients metric with a single input and a pair of inputs are governed by the same quantity χ_1. We further investigate the relationship between variance and mean of statistical metrics empirically and find an emergence of universality. Our finding of a universal relationship between variance and mean of statistical metrics of gradient backpropagation suggests a deeper mechanism behind it. This mechanism may be comprehended better by studying more different network structures such as Resnet. Finally, for networks with or without dropout, we attribute the maximum trainable length to the formula $L \leq \min\{12\xi_1, 12\xi_2\}$, which is novel and important.

3.6 Proof

3.6.1 Derivation of $\widetilde{q}^{l}_{aa}$ on linear dropout networks with a single input

(1) The L^{th} layer:

$$\delta^L_{i,a} = \frac{\partial E_a}{\partial z^L_{i,a}} = \frac{\partial E_a}{\partial y^L_{i,a}} = 2y^L_{i,a} = 2z^L_{i,a}$$

$$\widetilde{q}^L_{aa} = E[(\delta^L_{i,a})^2] = 4E[(z^L_{i,a})^2] = 4q^*_{aa}$$

(2) The $(L-1)^{th}$ layer:

$$z^L_{j,a} = \frac{1}{\rho}\sum_k W^L_{jk}p^L_{k,a}y^{L-1}_{k,a} + b^L_j, \qquad y^{L-1}_{k,a} = \phi_(z^{L-1}_{k,a}) = z^{L-1}_{k,a}$$

$$\delta^{L-1}_{i,a} = \frac{\partial E_a}{\partial z^L_a}\frac{\partial z^L_a}{\partial z^{L-1}_{i,a}} = \sum_j \delta^L_{j,a}\frac{\partial z^L_{j,a}}{\partial z^{L-1}_{i,a}} = \sum_j \delta^L_{j,a}\frac{p^L_{i,a}}{\rho}W^L_{ji} = \sum_j 2z^L_{j,a}\frac{p^L_{i,a}}{\rho}W^L_{ji}$$

$$\begin{aligned}
\widetilde{q}^{L-1}_{aa} &= 4E\Big[\sum_j\sum_{j'} z^L_{j,a}z^L_{j',a}\frac{(p^L_{i,a})^2}{\rho^2}W^L_{ji}W^L_{j'i}\Big] \\
&= 4E\Big[\sum_j\sum_{j'}\sum_k\sum_{k'}\big(\frac{p^L_{k,a}}{\rho}\frac{p^L_{k',a}}{\rho}W^L_{jk}W^L_{j'k'}z^{L-1}_{k,a}z^{L-1}_{k',a} + b^L_j b^L_{j'}\big)\frac{(p^L_{i,a})^2}{\rho^2}W^L_{ji}W^L_{j'i}\Big] \\
&= 4E\Big[\sum_{j=j'}\sum_{k=k'}\frac{(p^L_{k,a})^2}{\rho^2}\frac{(p^L_{i,a})^2}{\rho^2}(W^L_{jk}W^L_{ji})^2(z^{L-1}_{k,a})^2 \\
&+ \sum_{j\neq j'}\sum_{k=k'=i}\frac{(p^L_{i,a})^4}{\rho^4}(W^L_{ji}W^L_{j'i})^2(z^{L-1}_{i,a})^2 + \sum_{j=j'}(b^L_j)^2\frac{(p^L_{i,a})^2}{\rho^2}(W^L_{ji})^2\Big] \\
&\approx 4\Big[\frac{1}{\rho^2}(\sigma^2_\omega)^2q^*_{aa} + \frac{1}{\rho^3}(\sigma^2_\omega)^2q^*_{aa} + \sigma^2_b\sigma^2_\omega\frac{1}{\rho}\Big]
\end{aligned} \tag{3.14}$$

Since,

$$q^*_{aa} = \frac{\sigma^2_\omega}{\rho}q^*_{aa} + \sigma^2_b$$

We rewrite Eq (3.14) as:

$$\begin{aligned}
\widetilde{q}^{L-1}_{aa} &= 4\Big[\frac{1}{\rho^2}(\sigma^2_\omega)^2q^*_{aa} + \frac{1}{\rho^3}(\sigma^2_\omega)^2q^*_{aa} + \sigma^2_b\sigma^2_\omega\frac{1}{\rho}\Big] \\
&= 4\Big[\frac{\sigma^2_\omega}{\rho}q^*_{aa} + \frac{1}{\rho^3}(\sigma^2_\omega)^2q^*_{aa}\Big] \\
&= 4\frac{q^*_{aa}}{\rho}\frac{\sigma^2_\omega}{\rho}\Big[\rho + \frac{\sigma^2_\omega}{\rho}\Big]
\end{aligned} \tag{3.15}$$

(3) The $(L-2)^{th}$ layer:

$$\begin{aligned}\delta_{i,a}^{L-2} = \sum_j \delta_{j,a}^{L-1}\frac{p_{i,a}^{L-1}}{\rho}W_{ji}^{L-1} &= \sum_j\sum_k 2z_{k,a}^L\frac{p_{j,a}^L}{\rho}W_{kj}^L\frac{p_{i,a}^{L-1}}{\rho}W_{ji}^{L-1} \\ &= 2\sum_j\sum_k z_{k,a}^L\frac{p_{j,a}^L}{\rho}\frac{p_{i,a}^{L-1}}{\rho}W_{kj}^LW_{ji}^{L-1}\end{aligned} \tag{3.16}$$

$$\begin{aligned}\widetilde{q}_{aa}^{L-2} &= 4E\Big[\sum_j\sum_{j'}\sum_k\sum_{k'} z_{k,a}^Lz_{k',a}^L\frac{p_{j,a}^Lp_{j',a}^L}{\rho^2}\frac{(p_{i,a}^{L-1})^2}{\rho^2}W_{kj}^LW_{k'j'}^LW_{ji}^{L-1}W_{j'i}^{L-1}\Big] \\
&= 4E\Big[\sum_j\sum_{j'}\sum_k\sum_{k'}\sum_m\sum_{m'}(z_{m,a}^{L-1}z_{m',a}^{L-1}\frac{p_{m,a}^Lp_{m',a}^L}{\rho^2}W_{km}^LW_{k'm'}^L + b_k^Lb_{k'}^L) \\
&\frac{p_{j,a}^Lp_{j',a}^L}{\rho^2}\frac{(p_{i,a}^{L-1})^2}{\rho^2}W_{kj}^LW_{k'j'}^LW_{ji}^{L-1}W_{j'i}^{L-1}\Big] \\
&= 4E\Big[\sum_j\sum_{j'}\sum_k\sum_{k'}\sum_m\sum_{m'} z_{m,a}^{L-1}z_{m',a}^{L-1}\frac{p_{m,a}^Lp_{m',a}^Lp_{j,a}^Lp_{j',a}^L}{\rho^4}\frac{(p_{i,a}^{L-1})^2}{\rho^2}W_{km}^L \\
&W_{k'm'}^LW_{kj}^LW_{k'j'}^LW_{ji}^{L-1}W_{j'i}^{L-1} \\
&+\sum_j\sum_{j'}\sum_k\sum_{k'} b_k^Lb_{k'}^L\frac{p_{j,a}^Lp_{j',a}^L}{\rho^2}\frac{(p_{i,a}^{L-1})^2}{\rho^2}W_{kj}^LW_{k'j'}^LW_{ji}^{L-1}W_{j'i}^{L-1}\Big] \\
&= 4E\Big[\sum_{\substack{j,j',k,k'\\m,m',n,n'}}(\frac{p_{n,a}^{L-1}p_{n',a}^{L-1}}{\rho^2}W_{mn}^{L-1}W_{m'n'}^{L-1}z_{n,a}^{L-2}z_{n',a}^{L-2} + b_m^{L-1}b_{m'}^{L-1}) \\
&\frac{p_{m,a}^Lp_{m',a}^Lp_{j,a}^Lp_{j',a}^L}{\rho^4}\frac{(p_{i,a}^{L-1})^2}{\rho^2}W_{km}^LW_{k'm'}^LW_{kj}^LW_{k'j'}^LW_{ji}^{L-1}W_{j'i}^{L-1} \\
&+\sum_{j,j',k,k'} b_k^Lb_{k'}^L\frac{p_{j,a}^Lp_{j',a}^L}{\rho^2}\frac{(p_{i,a}^{L-1})^2}{\rho^2}W_{kj}^LW_{k'j'}^LW_{ji}^{L-1}W_{j'i}^{L-1}\Big] \\
&= 4E\Big[\sum_{\substack{j,j',k,k'\\m,m',n,n'}} z_{n,a}^{L-2}z_{n',a}^{L-2}\frac{p_{m,a}^Lp_{m',a}^Lp_{j,a}^Lp_{j',a}^L}{\rho^4}\frac{p_{n,a}^{L-1}p_{n',a}^{L-1}(p_{i,a}^{L-1})^2}{\rho^4} \\
&W_{km}^LW_{k'm'}^LW_{kj}^LW_{k'j'}^LW_{ji}^{L-1}W_{j'i}^{L-1}W_{mn}^{L-1}W_{m'n'}^{L-1} \\
&+\sum_{j,j',k,k',m,m'} b_m^{L-1}b_{m'}^{L-1}\frac{p_{m,a}^Lp_{m',a}^Lp_{j,a}^Lp_{j',a}^L}{\rho^4}\frac{(p_{i,a}^{L-1})^2}{\rho^2}W_{km}^LW_{k'm'}^LW_{kj}^LW_{k'j'}^LW_{ji}^{L-1}W_{j'i}^{L-1} \\
&+\sum_{j,j',k,k'} b_k^Lb_{k'}^L\frac{p_{j,a}^Lp_{j',a}^L}{\rho^2}\frac{(p_{i,a}^{L-1})^2}{\rho^2}W_{kj}^LW_{k'j'}^LW_{ji}^{L-1}W_{j'i}^{L-1}\Big] \\
&= 4E\Big[\mathrm{I}+\mathrm{II}+\mathrm{III}\Big]\end{aligned} \tag{3.17}$$

There are three parts in Eq (5.15), we denote them as I, II, III and compute them one by one,

$$
\begin{aligned}
E[\mathrm{I}] = E\Big[& \sum_{\substack{j,j',k,k'\\ m,m',n,n'}} z_{n,a}^{L-2} z_{n',a}^{L-2} \frac{p_{m,a}^{L} p_{m',a}^{L} p_{j,a}^{L} p_{j',a}^{L}}{\rho^4} \frac{p_{n,a}^{L-1} p_{n',a}^{L-1} (p_{i,a}^{L-1})^2}{\rho^4} \\
& W_{km}^{L} W_{k'm'}^{L} W_{kj}^{L} W_{k'j'}^{L} W_{ji}^{L-1} W_{j'i}^{L-1} W_{mn}^{L-1} W_{m'n'}^{L-1} \Big] \\
= E\Big[& \Big(\sum_{\substack{n=n'=i\\ k\neq k'\\ m=j\\ m'=j'}} + \sum_{\substack{n=n'\neq i\\ k=k'\\ m=m'\\ j=j'}} + \sum_{\substack{n=n'\neq i\\ k\neq k'\\ m=m'=j=j'}} \Big) z_{n,a}^{L-2} z_{n',a}^{L-2} \frac{p_{m,a}^{L} p_{m',a}^{L} p_{j,a}^{L} p_{j',a}^{L}}{\rho^4} \\
& \frac{p_{n,a}^{L-1} p_{n',a}^{L-1} (p_{i,a}^{L-1})^2}{\rho^4} W_{km}^{L} W_{k'm'}^{L} W_{kj}^{L} W_{k'j'}^{L} W_{ji}^{L-1} W_{j'i}^{L-1} W_{mn}^{L-1} W_{m'n'}^{L-1} \Big] \\
= E\Big[& \sum_{\substack{n=n'=i,k\neq k'\\ m=j,m'=j'}} (z_{i,a}^{L-2})^2 \frac{(p_{j,a}^{L})^2 (p_{j',a}^{L})^2}{\rho^4} \frac{(p_{i,a}^{L-1})^4}{\rho^4} (W_{kj}^{L} W_{k'j'})^2 (W_{ji}^{L-1} W_{j'i}^{L-1})^2 \\
& + \sum_{\substack{n=n'\neq i,k=k'\\ m=m',j=j'}} (z_{n,a}^{L-2})^2 \frac{(p_{m,a}^{L})^2 (p_{j,a}^{L})^2}{\rho^4} \frac{(p_{n,a}^{L-1})^2 (p_{i,a}^{L-1})^2}{\rho^4} (W_{km}^{L} W_{kj}^{L})^2 (W_{ji}^{L-1} W_{mn}^{L-1})^2 \\
& + \sum_{\substack{n=n'\neq i,k\neq k'\\ m=m'=j=j'}} (z_{n,a}^{L-2})^2 \frac{(p_{j,a}^{L})^4}{\rho^4} \frac{(p_{n,a}^{L-1})^2 (p_{i,a}^{L-1})^2}{\rho^4} (W_{kj}^{L} W_{k'j}^{L})^2 (W_{ji}^{L-1} W_{jn}^{L-1})^2 \Big] \\
\approx\ & q_{aa}^{L-2} \frac{1}{\rho^5} (\sigma_\omega^2)^4 + q_{aa}^{L-2} \frac{1}{\rho^4} (\sigma_\omega^2)^4 + q_{aa}^{L-2} \frac{1}{\rho^5} (\sigma_\omega^2)^4 \\
=\ & q_{aa}^{L-2} \Big[\frac{(\sigma_\omega^2)^4}{\rho^5} + \frac{(\sigma_\omega^2)^4}{\rho^4} + \frac{(\sigma_\omega^2)^4}{\rho^5} \Big]
\end{aligned}
\tag{3.18}
$$

$$
\begin{aligned}
E[\mathrm{II}] = E\Big[&\sum_{j,j',k,k',m,m'} b_m^{L-1} b_{m'}^{L-1} \frac{p_{m,a}^L p_{m',a}^L p_{j,a}^L p_{j',a}^L}{\rho^4} \frac{(p_{i,a}^{L-1})^2}{\rho^2} \\
& W_{km}^L W_{k'm'}^L W_{kj}^L W_{k'j'}^L W_{ji}^{L-1} W_{j'i}^{L-1} \Big] \\
&= E\Big[\Big(\sum_{\substack{k \neq k' \\ m = j = m' = j'}} + \sum_{\substack{k = k' \\ m = m' \\ j = j'}} \Big) b_m^{L-1} b_{m'}^{L-1} \frac{p_{m,a}^L p_{m',a}^L p_{j,a}^L p_{j',a}^L}{\rho^4} \frac{(p_{i,a}^{L-1})^2}{\rho^2} \\
& W_{km}^L W_{k'm'}^L W_{kj}^L W_{k'j'}^L W_{ji}^{L-1} W_{j'i}^{L-1} \Big] \\
&= E\Big[\sum_{\substack{k \neq k' \\ m=j=m'=j'}} (b_m^{L-1})^2 \frac{(p_{j,a}^L)^4}{\rho^4} \frac{(p_{i,a}^{L-1})^2}{\rho^2} (W_{kj}^L W_{k'j'}^L)^2 (W_{ji}^{L-1})^2 \Big] \\
& + \sum_{k=k',m=m',j=j'} (b_m^{L-1})^2 \frac{(p_{m,a}^L)^2 (p_{j,a}^L)^2}{\rho^4} \frac{(p_{i,a}^{L-1})^2}{\rho^2} (W_{km}^L W_{kj}^L)^2 (W_{ji}^{L-1})^2 \Big] \\
&\approx \sigma_b^2 \frac{(\sigma_\omega^2)^3}{\rho^4} + \sigma_b^2 \frac{(\sigma_\omega^2)^3}{\rho^3}
\end{aligned}
\tag{3.19}
$$

$$
\begin{aligned}
E[\mathrm{III}] &= E\Big[\sum_{j,j',k,k'} b_k^L b_{k'}^L \frac{p_{j,a}^L p_{j',a}^L}{\rho^2} \frac{(p_{i,a}^{L-1})^2}{\rho^2} W_{kj}^L W_{k'j'}^L W_{ji}^{L-1} W_{j'i}^{L-1} \Big] \\
&= E\Big[\sum_{j=j',k=k'} (b_k^L)^2 \frac{(p_{j,a}^L)^2}{\rho^2} \frac{(p_{i,a}^{L-1})^2}{\rho^2} (W_{kj}^L)^2 (W_{ji}^{L-1})^2 \Big] \\
&= \sigma_b^2 \frac{(\sigma_\omega^2)^2}{\rho^2}
\end{aligned}
\tag{3.20}
$$

Finally, we have,

$$
\begin{aligned}
\widetilde{q}_{aa}^{L-2} &= 4E\Big[\mathrm{I} + \mathrm{II} + \mathrm{III} \Big] \\
&= 4\Big[q_{aa}^* \Big(\big(\frac{(\sigma_\omega^2)^4}{\rho^5} + \frac{(\sigma_\omega^2)^4}{\rho^4} + \frac{(\sigma_\omega^2)^4}{\rho^5} \Big) + \sigma_b^2 \frac{(\sigma_\omega^2)^3}{\rho^4} + \sigma_b^2 \frac{(\sigma_\omega^2)^3}{\rho^3} + \sigma_b^2 \frac{(\sigma_\omega^2)^2}{\rho^2} \Big] \\
&= 4\Big[q_{aa}^* \frac{(\sigma_\omega^2)^4}{\rho^5} + q_{aa}^* \frac{(\sigma_\omega^2)^3}{\rho^4} + q_{aa}^* \frac{(\sigma_\omega^2)^3}{\rho^3} + \sigma_b^2 \frac{(\sigma_\omega^2)^2}{\rho^2} \Big] \\
&= 4\Big[q_{aa}^* \frac{(\sigma_\omega^2)^4}{\rho^5} + q_{aa}^* \frac{(\sigma_\omega^2)^3}{\rho^4} + q_{aa}^* \frac{(\sigma_\omega^2)^2}{\rho^2} \Big] \\
&= 4 \frac{q_{aa}^*}{\rho} \big(\frac{\sigma_\omega^2}{\rho} \big)^2 \Big[\rho + \frac{\sigma_\omega^2}{\rho} + \big(\frac{\sigma_\omega^2}{\rho} \big)^2 \Big]
\end{aligned}
\tag{3.21}
$$

To summarize, we list the results for $\widetilde{q}_{aa}^{L}$, $\widetilde{q}_{aa}^{L-1}$, and $\widetilde{q}_{aa}^{L-2}$:

$$\begin{aligned} \widetilde{q}_{aa}^{L} &= 4q_{aa}^{*} && (3.22) \\ \widetilde{q}_{aa}^{L-1} &= 4\frac{q_{aa}^{*}}{\rho}\frac{\sigma_{\omega}^{2}}{\rho}\Big[\rho+\frac{\sigma_{\omega}^{2}}{\rho}\Big] && (3.23) \\ \widetilde{q}_{aa}^{L-2} &= 4\frac{q_{aa}^{*}}{\rho}\Big(\frac{\sigma_{\omega}^{2}}{\rho}\Big)^{2}\Big[\rho+\frac{\sigma_{\omega}^{2}}{\rho}+(\frac{\sigma_{\omega}^{2}}{\rho})^{2}\Big] && (3.24) \end{aligned}$$

Using mathematical induction method we draw the conclusion that,

$$\widetilde{q}_{aa}^{l} = 4\frac{q_{aa}^{*}}{\rho}\Big(\frac{\sigma_{\omega}^{2}}{\rho}\Big)^{L-l}\Big[\rho+\sum_{j=1}^{L-l}\Big(\frac{\sigma_{\omega}^{2}}{\rho}\Big)^{j}\Big] \tag{3.25}$$

Using the relation,

$$\frac{\partial E}{\partial W_{ij}^{l}} = \frac{p_{j}^{l}}{\rho}\phi(z_{j}^{l-1})\delta_{i}^{l}, \tag{3.26}$$

we obtain the final result,

$$g_{aa}^{l} = 4(\frac{q_{aa}^{*}}{\rho})^{2}\Big(\frac{\sigma_{\omega}^{2}}{\rho}\Big)^{L-l}\Big[\rho+\sum_{j=1}^{L-l}\Big(\frac{\sigma_{\omega}^{2}}{\rho}\Big)^{j}\Big]. \tag{3.27}$$

3.6.2 Derivation of $\widetilde{q}^{l}_{ab}$ on linear dropout networks with a pair of inputs

(1)The L^{th} layer:

$$\delta^L_{i,a} = 2z^L_{i,a}$$

$$\widetilde{q}^L_{ab} = E[(\delta^L_{i,a}\delta^L_{i,b})] = 4E[(z^L_{i,a}z^L_{i,b})] = 4q^*_{ab}$$

(2)The $(L-1)^{th}$ layer:

$$z^L_{j,a} = \frac{1}{\rho}\sum_k W^L_{jk}p^L_{k,a}y^{L-1}_{k,a} + b^L_j, \qquad y^{L-1}_{k,a} = \phi_(z^{L-1}_{k,a}) = z^{L-1}_{k,a}$$

$$z^L_{j',b} = \frac{1}{\rho}\sum_{k'} W^L_{j'k'}p^L_{k',b}y^{L-1}_{k',b} + b^L_j, \qquad y^{L-1}_{k',b} = \phi_(z^{L-1}_{k',b}) = z^{L-1}_{k',b}$$

$$\delta^{L-1}_{i,a} = \sum_j 2z^L_{j,a}\frac{p^L_{i,a}}{\rho}W^L_{ji}$$

$$\delta^{L-1}_{i,b} = \sum_{j'} 2z^L_{j',b}\frac{p^L_{i,b}}{\rho}W^L_{j'i}$$

$$\begin{aligned}
\widetilde{q}^{L-1}_{ab} &= 4E\Big[\sum_j\sum_{j'} z^L_{j,a}z^L_{j',b}\frac{p^L_{i,a}p^L_{i,b}}{\rho^2}W^L_{ji}W^L_{j'i}\Big] \\
&= 4E\Big[\sum_j\sum_{j'}\sum_k\sum_{k'}(\frac{p^L_{k,a}}{\rho}\frac{p^L_{k',b}}{\rho}W^L_{jk}W^L_{j'k'}z^{L-1}_{k,a}z^{L-1}_{k',b} + b^L_jb^L_{j'})\frac{p^L_{i,a}p^L_{i,b}}{\rho^2}W^L_{ji}W^L_{j'i}\Big] \\
&= 4E\Big[\sum_{j=j'}\sum_{k=k'} z^{L-1}_{k,a}z^{L-1}_{k,b}\frac{p^L_{k,a}p^L_{k,b}p^L_{i,a}p^L_{i,b}}{\rho^4}(W^L_{jk}W^L_{ji})^2 \\
&+\sum_{j\neq j'}\sum_{k=k'=i} z^{L-1}_{i,a}z^{L-1}_{i,b}\frac{(p^L_{i,a})^2(p^L_{i,b})^2}{\rho^4}(W^L_{ji}W^L_{j'i})^2 + \sum_{j=j'}(b^L_j)^2\frac{p^L_{i,a}p^L_{i,b}}{\rho^2}(W^L_{ji})^2\Big] \\
&\approx 4\Big[(\sigma^2_\omega)^2q^*_{ab} + \frac{1}{\rho^2}(\sigma^2_\omega)^2q^*_{ab} + \sigma^2_b\sigma^2_\omega\Big]
\end{aligned} \tag{3.28}$$

Here, we have

$$q^*_{ab} = \sigma^2_\omega q^*_{ab} + \sigma^2_b$$

We rewrite Eq (3.28) as:

$$\begin{aligned}\widetilde{q}_{ab}^{L-1} &= 4\Big[\frac{1}{\rho^2}(\sigma_\omega^2)^2 q_{ab}^* + (\sigma_\omega^2)^2 q_{ab}^* + \sigma_b^2\sigma_\omega^2\Big] \\ &= 4\Big[\frac{1}{\rho^2}(\sigma_\omega^2)^2 q_{ab}^* + \sigma_\omega^2 q_{ab}^*\Big] \\ &= 4q_{ab}^*\sigma_\omega^2\Big[1+\frac{\sigma_\omega^2}{\rho^2}\Big]\end{aligned} \tag{3.29}$$

(3)The $(L-2)^{th}$ layer:

$$\begin{aligned}\delta_{i,a}^{L-2} &= \sum_j \delta_{j,a}^{L-1}\frac{p_{i,a}^{L-1}}{\rho}W_{ji}^{L-1} = \sum_j\sum_k 2z_{k,a}^L\frac{p_{j,a}^L}{\rho}W_{kj}^L\frac{p_{i,a}^{L-1}}{\rho}W_{ji}^{L-1} \\ &= 2\sum_j\sum_k z_{k,a}^L\frac{p_{j,a}^L}{\rho}\frac{p_{i,a}^{L-1}}{\rho}W_{kj}^L W_{ji}^{L-1}\end{aligned} \tag{3.30}$$

$$\begin{aligned}\delta_{i,b}^{L-2} &= \sum_{j'} \delta_{j',b}^{L-1}\frac{p_{i,b}^{L-1}}{\rho}W_{j'i}^{L-1} = \sum_{j'}\sum_{k'} 2z_{k',b}^L\frac{p_{j',b}^L}{\rho}W_{k'j'}^L\frac{p_{i,b}^{L-1}}{\rho}W_{j'i}^{L-1} \\ &= 2\sum_{j'}\sum_{k'} z_{k',b}^L\frac{p_{j',b}^L}{\rho}\frac{p_{i,b}^{L-1}}{\rho}W_{k'j'}^L W_{j'i}^{L-1}\end{aligned} \tag{3.31}$$

$$
\begin{aligned}
\widetilde{q}_{ab}^{L-2} &= 4E\Big[\sum_{j}\sum_{j'}\sum_{k}\sum_{k'} z^{L}_{k,a} z^{L}_{k',b} \frac{p^{L}_{j,a}p^{L}_{j',b}}{\rho^2}\frac{p^{L-1}_{i,a}p^{L-1}_{i,b}}{\rho^2} W^{L}_{kj}W^{L}_{k'j'}W^{L-1}_{ji}W^{L-1}_{j'i}\Big] \\
&= 4E\Big[\sum_{j}\sum_{j'}\sum_{k}\sum_{k'}\sum_{m}\sum_{m'}(z^{L-1}_{m,a} z^{L-1}_{m',b}\frac{p^{L}_{m,a}p^{L}_{m',b}}{\rho^2} W^{L}_{km}W^{L}_{k'm'} + b^{L}_{k}b^{L}_{k'}) \\
&\frac{p^{L}_{j,a}p^{L}_{j',b}}{\rho^2}\frac{p^{L-1}_{i,a}p^{L-1}_{i,b}}{\rho^2} W^{L}_{kj}W^{L}_{k'j'}W^{L-1}_{ji}W^{L-1}_{j'i}\Big] \\
&= 4E\Big[\sum_{j}\sum_{j'}\sum_{k}\sum_{k'}\sum_{m}\sum_{m'} z^{L-1}_{m,a} z^{L-1}_{m',b}\frac{p^{L}_{m,a}p^{L}_{m',b}p^{L}_{j,a}p^{L}_{j',b}}{\rho^4}\frac{p^{L-1}_{i,a}p^{L-1}_{i,b}}{\rho^2} \\
&W^{L}_{km}W^{L}_{k'm'}W^{L}_{kj}W^{L}_{k'j'}W^{L-1}_{ji}W^{L-1}_{j'i} \\
&+\sum_{j}\sum_{j'}\sum_{k}\sum_{k'} b^{L}_{k}b^{L}_{k'}\frac{p^{L}_{j,a}p^{L}_{j',b}}{\rho^2}\frac{p^{L-1}_{i,a}p^{L-1}_{i,b}}{\rho^2} W^{L}_{kj}W^{L}_{k'j'}W^{L-1}_{ji}W^{L-1}_{j'i}\Big] \\
&= 4E\Big[\sum_{\substack{j,j',k,k'\\ m,m',n,n'}} (\frac{p^{L-1}_{n,a}p^{L-1}_{n',b}}{\rho^2} W^{L-1}_{mn}W^{L-1}_{m'n'} z^{L-2}_{n,a} z^{L-2}_{n',b} + b^{L-1}_{m}b^{L-1}_{m'}) \\
&\frac{p^{L}_{m,a}p^{L}_{m',b}p^{L}_{j,a}p^{L}_{j',b}}{\rho^4}\frac{p^{L-1}_{i,a}p^{L-1}_{i,b}}{\rho^2} W^{L}_{km}W^{L}_{k'm'}W^{L}_{kj}W^{L}_{k'j'}W^{L-1}_{ji}W^{L-1}_{j'i} \\
&+\sum_{j,j',k,k'} b^{L}_{k}b^{L}_{k'}\frac{p^{L}_{j,a}p^{L}_{j',b}}{\rho^2}\frac{p^{L-1}_{i,a}p^{L-1}_{i,b}}{\rho^2} W^{L}_{kj}W^{L}_{k'j'}W^{L-1}_{ji}W^{L-1}_{j'i}\Big] \\
&= 4E\Big[\sum_{\substack{j,j',k,k'\\ m,m',n,n'}} z^{L-2}_{n,a} z^{L-2}_{n',b}\frac{p^{L}_{m,a}p^{L}_{m',b}p^{L}_{j,a}p^{L}_{j',b}}{\rho^4}\frac{p^{L-1}_{n,a}p^{L-1}_{n',b}p^{L-1}_{i,a}p^{L-1}_{i,b}}{\rho^4} \\
&W^{L}_{km}W^{L}_{k'm'}W^{L}_{kj}W^{L}_{k'j'}W^{L-1}_{ji}W^{L-1}_{j'i}W^{L-1}_{mn}W^{L-1}_{m'n'} \\
&+\sum_{j,j',k,k',m,m'} b^{L-1}_{m}b^{L-1}_{m'}\frac{p^{L}_{m,a}p^{L}_{m',b}p^{L}_{j,a}p^{L}_{j',b}}{\rho^4}\frac{p^{L-1}_{i,a}p^{L-1}_{i,b}}{\rho^2} W^{L}_{km}W^{L}_{k'm'}W^{L}_{kj}W^{L}_{k'j'}W^{L-1}_{ji}W^{L-1}_{j'i} \\
&+\sum_{j,j',k,k'} b^{L}_{k}b^{L}_{k'}\frac{p^{L}_{j,a}p^{L}_{j',b}}{\rho^2}\frac{p^{L-1}_{i,a}p^{L-1}_{i,b}}{\rho^2} W^{L}_{kj}W^{L}_{k'j'}W^{L-1}_{ji}W^{L-1}_{j'i}\Big] \\
&= 4E\Big[\mathrm{I}+\mathrm{II}+\mathrm{III}\Big]
\end{aligned}
\tag{3.32}
$$

$$
\begin{aligned}
E[\mathrm{I}] = E\Big[\sum_{\substack{j,j',k,k'\\ m,m',n,n'}} & z^{L-2}_{n,a} z^{L-2}_{n',b} \frac{p^L_{m,a} p^L_{m',b} p^L_{j,a} p^L_{j',b}}{\rho^4} \frac{p^{L-1}_{n,a} p^{L-1}_{n',b} p^{L-1}_{i,a} p^{L-1}_{i,b}}{\rho^4} \\
& W^L_{km} W^L_{k'm'} W^L_{kj} W^L_{k'j'} W^{L-1}_{ji} W^{L-1}_{j'i} W^{L-1}_{mn} W^{L-1}_{m'n'} \Big] \\
= E\Big[\Big(& \sum_{\substack{n=n'=i\\ k\neq k'\\ m=j\\ m'=j'}} + \sum_{\substack{n=n'\neq i\\ k=k'\\ m=m'\\ j=j'}} + \sum_{\substack{n=n'\neq i\\ k\neq k'\\ m=m'=j=j'}} \Big) z^{L-2}_{n,a} z^{L-2}_{n',b} \frac{p^L_{m,a} p^L_{m',b} p^L_{j,a} p^L_{j',b}}{\rho^4} \\
& \frac{p^{L-1}_{n,a} p^{L-1}_{n',b} p^{L-1}_{i,a} p^{L-1}_{i,b}}{\rho^4} W^L_{km} W^L_{k'm'} W^L_{kj} W^L_{k'j'} W^{L-1}_{ji} W^{L-1}_{j'i} W^{L-1}_{mn} W^{L-1}_{m'n'} \Big] \\
= E\Big[& \sum_{\substack{n=n'=i,k\neq k'\\ m=j,m'=j'}} z^{L-2}_{n,a} z^{L-2}_{n,b} \frac{(p^L_{j,a} p^L_{j',b})^2}{\rho^4} \frac{(p^{L-1}_{i,a} p^{L-1}_{i,b})^2}{\rho^4} (W^L_{kj} W_{k'j'})^2 (W^{L-1}_{ji} W^{L-1}_{j'i})^2 \\
+ & \sum_{\substack{n=n'\neq i,k=k'\\ m=m',j=j'}} z^{L-2}_{n,a} z^{L-2}_{n,b} \frac{p^L_{m,a} p^L_{m,b} p^L_{j,a} p^L_{j,b}}{\rho^4} \frac{p^{L-1}_{n,a} p^{L-1}_{n,b} p^{L-1}_{i,a} p^{L-1}_{i,b}}{\rho^4} \\
& (W^L_{km} W^L_{kj})^2 (W^{L-1}_{ji} W^{L-1}_{mn})^2 \\
+ & \sum_{\substack{n=n'\neq i,k\neq k'\\ m=m'=j=j'}} (z^{L-2}_{n,a})^2 \frac{(p^L_{j,a} p^L_{j,b})^2}{\rho^4} \frac{p^{L-1}_{n,a} p^{L-1}_{n,b} p^{L-1}_{i,a} p^{L-1}_{i,b}}{\rho^4} \\
& (W^L_{kj} W^L_{k'j})^2 (W^{L-1}_{ji} W^{L-1}_{jn})^2 \Big] \\
\approx\ & q^*_{ab} \frac{1}{\rho^4} (\sigma^2_\omega)^4 + q^*_{ab} (\sigma^2_\omega)^4 + q^*_{ab} \frac{1}{\rho^2} (\sigma^2_\omega)^4 \\
=\ & q^*_{ab} \Big[\frac{(\sigma^2_\omega)^4}{\rho^4} + (\sigma^2_\omega)^4 + \frac{(\sigma^2_\omega)^4}{\rho^2} \Big]
\end{aligned}
\tag{3.33}
$$

$$\begin{aligned}
E[\mathrm{II}] = E\Big[& \sum_{j,j',k,k',m,m'} b_m^{L-1} b_{m'}^{L-1} \frac{p_{m,a}^L p_{m',b}^L p_{j,a}^L p_{j',b}^L}{\rho^4} \frac{p_{i,a}^{L-1} p_{i,b}^{L-1}}{\rho^2} \\
& W_{km}^L W_{k'm'}^L W_{kj}^L W_{k'j'}^L W_{ji}^{L-1} W_{j'i}^{L-1} \Big] \\
& = E\Big[\Big(\sum_{\substack{k \neq k' \\ m=j=m'=j'}} + \sum_{\substack{k=k' \\ m=m' \\ j=j'}} \Big) b_m^{L-1} b_{m'}^{L-1} \frac{p_{m,a}^L p_{m',b}^L p_{j,a}^L p_{j',b}^L}{\rho^4} \frac{p_{i,a}^{L-1} p_{i,b}^{L-1}}{\rho^2} \\
& W_{km}^L W_{k'm'}^L W_{kj}^L W_{k'j'}^L W_{ji}^{L-1} W_{j'i}^{L-1} \Big] \\
& = E\Big[\sum_{\substack{k\neq k' \\ m=j=m'=j'}} (b_m^{L-1})^2 \frac{(p_{j,a}^L p_{j,b}^L)^2}{\rho^4} \frac{p_{i,a}^{L-1} p_{i,b}^{L-1}}{\rho^2} (W_{kj}^L W_{k'j'}^L)^2 (W_{ji}^{L-1})^2 \Big] \\
& + \sum_{k=k', m=m', j=j'} (b_m^{L-1})^2 \frac{p_{m,a}^L p_{m,b}^L p_{j,a}^L p_{j,b}^L}{\rho^4} \frac{p_{i,a}^L p_{i,b}^L}{\rho^2} (W_{km}^L W_{kj}^L)^2 (W_{ji}^{L-1})^2 \Big] \\
& \approx \sigma_b^2 \frac{(\sigma_\omega^2)^3}{\rho^2} + \sigma_b^2 (\sigma_\omega^2)^3
\end{aligned} \tag{3.34}$$

$$\begin{aligned}
E[\mathrm{III}] &= E\Big[\sum_{j,j',k,k'} b_k^L b_{k'}^L \frac{p_{j,a}^L p_{j',b}^L}{\rho^2} \frac{p_{i,a}^{L-1} p_{i,b}^{L-1}}{\rho^2} W_{kj}^L W_{k'j'}^L W_{ji}^{L-1} W_{j'i}^{L-1} \Big] \\
&= E\Big[\sum_{j=j', k=k'} (b_k^L)^2 \frac{p_{j,a}^L p_{j,b}^L}{\rho^2} \frac{p_{i,a}^{L-1} p_{i,b}^{L-1}}{\rho^2} (W_{kj}^L)^2 (W_{ji}^{L-1})^2 \Big] \\
&= \sigma_b^2 (\sigma_\omega^2)^2
\end{aligned} \tag{3.35}$$

Finally, we have,

$$\begin{aligned}
\widetilde{q}_{ab}^{L-2} &= 4E\Big[\mathrm{I} + \mathrm{II} + \mathrm{III} \Big] \\
&= 4\Big[q_{ab}^* \Big(\frac{(\sigma_\omega^2)^4}{\rho^4} + (\sigma_\omega^2)^4 + \frac{(\sigma_\omega^2)^4}{\rho^2} \Big) + \sigma_b^2 \frac{(\sigma_\omega^2)^3}{\rho^2} + \sigma_b^2 (\sigma_\omega^2)^3 + \sigma_b^2 (\sigma_\omega^2)^2 \Big] \\
&= 4\Big[q_{ab}^* \frac{(\sigma_\omega^2)^4}{\rho^4} + q_{ab}^* \frac{(\sigma_\omega^2)^3}{\rho^2} + q_{ab}^* (\sigma_\omega^2)^3 + \sigma_b^2 (\sigma_\omega^2)^2 \Big] \\
&= 4\Big[q_{ab}^* \frac{(\sigma_\omega^2)^4}{\rho^4} + q_{ab}^* \frac{(\sigma_\omega^2)^3}{\rho^2} + q_{ab}^* (\sigma_\omega^2)^2 \Big] \\
&= 4 q_{ab}^* (\sigma_\omega^2)^2 \Big[1 + \frac{\sigma_\omega^2}{\rho^2} + (\frac{\sigma_\omega^2}{\rho^2})^2 \Big]
\end{aligned} \tag{3.36}$$

To summarize, we list the results for $\widetilde{q}_{ab}^{L}$, $\widetilde{q}_{ab}^{L-1}$, and $\widetilde{q}_{ab}^{L-2}$:

$$\begin{aligned}
\widetilde{q}_{ab}^{L} &= 4q_{ab}^{*} && (3.37)\\
\widetilde{q}_{ab}^{L-1} &= 4q_{ab}^{*}\sigma_{\omega}^{2}\Big[1+\frac{\sigma_{\omega}^{2}}{\rho^{2}}\Big] && (3.38)\\
\widetilde{q}_{ab}^{L-2} &= 4q_{ab}^{*}(\sigma_{\omega}^{2})^{2}\Big[1+\frac{\sigma_{\omega}^{2}}{\rho^{2}}+(\frac{\sigma_{\omega}^{2}}{\rho^{2}})^{2}\Big] && (3.39)
\end{aligned}$$

Using mathematical induction method we draw the conclusion that

$$\widetilde{q}_{ab}^{l} = 4q_{ab}^{*}\Big(\sigma_{\omega}^{2}\Big)^{L-l}\Big[1+\sum_{j=1}^{L-l}\Big(\frac{\sigma_{\omega}^{2}}{\rho^{2}}\Big)^{j}\Big] \tag{3.40}$$

Using the relation,

$$\frac{\partial E}{\partial W_{ij}^{l}} = \frac{p_{j}^{l}}{\rho}\phi(z_{j}^{l-1})\delta_{i}^{l}, \tag{3.41}$$

we obtain the final result,

$$g_{ab}^{l} = 4(q_{ab}^{*})^{2}\Big(\sigma_{\omega}^{2}\Big)^{L-l}\Big[1+\sum_{j=1}^{L-l}\Big(\frac{\sigma_{\omega}^{2}}{\rho^{2}}\Big)\Big] \tag{3.42}$$

Chapter 4

Orthogonally-Initialized Networks and the Neural Tangent Kernel

4.1 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 [67], natural language processing [68], to reinforcement learning [69], and more. Beyond its fundamental shift in approach, an array of innovative techniques underpin the success of deep learning, such as residual connections [7], dropout [6], and batch normalization [8]. The mean field theory [22, 23] 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 [23] discovered, and then empirically proved, is that there exists a critical initialization called the edge of chaos, allowing the correlation signal to go infinitely far forward and preventing vanishing or exploding gradients. Later, this theory had been extended to a much wider range of architectures, e.g., convolutional networks [26], recurrent networks [27], dropout networks [70], residual networks [25], and batch normalization [28].

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* [71]. However, what was not known was how to impose dynamical isometry in deep nonlinear networks. Pennington et al. [24, 72] 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 [73, 74], convolutional networks [26], or recurrent networks [27] 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 [24]. In straightforward scenarios, orthogonal initialization is usually enough to impose dynamical isometry in a linear network. The benefit of the orthogonality in linear networks has been proven recently [75]. 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) [36], developed for studying the optimization of deep networks using gradient descent training in the infinite-width limit. In terms of definition, the 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 initialization converges to a deterministic kernel and remains unchanged when trained by the gradient descent under infinite-width limit setting. 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 linearized model [37]. It is known that these dynamics are called the NTK regime, or *lazy training* [76]. As the learning rate gets larger or the network becomes deeper, that is, out 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.

4.2 Preliminaries

4.2.1 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 FCN 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 $l \in \{1, \dots, L\}$ in this network is govern by,

$$x_i^l = \phi(h_i^l), \quad h_i^l = \sum_{j=1}^{n_l} W_{ij}^l x_j^{l-1} + b_i^l, \tag{4.1}$$

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

$$x_{i,\alpha}^l = \phi(h_{i,\alpha}^l), \quad h_{i,\alpha}^l = \sum_{j=1}^{n_l} \sum_{\beta=-k}^{k} W_{ij,\beta}^l x_{j,\alpha+\beta}^{l-1} + b_i^l, \tag{4.2}$$

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, $W_{ij}^l \sim \mathcal{N}(0, \frac{\sigma_w^2}{n_{l-1}})$, $\quad b_i^l \sim \mathcal{N}(0, \sigma_b^2)$, 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, $W_{ij}^l = \frac{\sigma_w}{\sqrt{n_{l-1}}} \omega_{ij}^l$, $\quad b_i^l = \sigma_b \beta_i^l$, where $\omega_{ij}^l, \beta_i^l \sim \mathcal{N}(0, 1)$.

4.2.2 Dynamical Isometry and Orthogonal Initialization

Consider the input-output Jacobian which is defined as $J = \frac{\partial h^L}{\partial x^0}$, where h^L is output function, x^0 is input. 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 [24]. More precisely, the singular values of D^l can be made arbitrarily close to 1 by choosing a linear regime in a nonlinear activation, combined with the fact that orthogonal matrix satisfies dynamical isometry. On the other hand, adopting a random orthogonal initialization can force the singular values of weights into 1. In particular, the weights are randomly generated from the orthogonal distribution,

$$(W^l)^T W^l = \sigma_w^2 \mathbf{I}, \tag{4.3}$$

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

$$W_{ij}^l = \frac{\sigma_w}{\sqrt{n_{l-1}}} \omega_{ij}^l, \quad (\omega^l)^T \omega^l = n_{l-1}\mathbf{I}. \tag{4.4}$$

We show a summary of improved standard parameterization and ntk-parameterization across FCN and CNN for Gaussian and orthogonal initialization in Table 4.1. The factor s in the layer equation of standard parameterization is introduced to prevent divergence of NTK [77].

4.2.3 Neural Tangent Kernel

NTK is originated from [36] and defined as,

$$\Theta_t(X, X) = \nabla_\theta f_t(\theta, X) \nabla_\theta f_t(\theta, X)^T. \tag{4.5}$$

where function f_t was the outputs of the network at training time t, i.e. $f_t(X) = h_t^L(X) \in \mathbb{R}^{D \times n_L}$, and $\nabla_\theta f_t(X) = \mathrm{vec}([\nabla_\theta f_t(x)]_{x \in X}) \in \mathbb{R}^{D n_L}$. Thus, the neural tangent kernel is formulated as a $Dn_L \times Dn_L$ matrix.

The original NTK work [36] studied the behavior of the output function f_t regarding the network under the infinite-width limit setting and trained using a gradient

Table 4.1 : 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.

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	$W_\alpha^T W_\alpha = \frac{\sigma_w^2}{2k+1}\mathbf{I}$		

descent method. Later, Lee et al. [37] presented this result in another statement that infinite width networks are linearized networks in the parameter space. We recall some of these results here.

Let η be the learning rate, and $\mathcal{L}$ be the loss function. Then dynamics of gradient flow for parameters and output function are given by,

$$\begin{aligned} \frac{\partial \theta}{\partial t} &= -\eta \nabla_\theta \mathcal{L} = -\eta \nabla_\theta f_t(\theta, X)^T \nabla_{f_t(\theta,X)} \mathcal{L} \\ \frac{\partial f_t(\theta,x)}{\partial t} &= \nabla_\theta f_t(\theta,x) \frac{\partial \theta}{\partial t} = -\eta \Theta_t(x,X) \nabla_{f_t(\theta,X)} \mathcal{L}. \end{aligned} \tag{4.6}$$

This equation for f_t has no substantial insight in studying behavior of networks since $\Theta_t(x,X)$ varies with the time evolution of training. However, as stated in above, the NTK $\Theta_t(X,X)$ converges to a deterministic and limiting kernel $\Theta_\infty(X,X)$ and does not change during training, under the infinite-width limit setting, i.e. $\Theta_t(X,X) = \Theta_\infty(X,X)$. As a result, the infinite width limit of the training dynamics can be expressed as,

$$\frac{\partial f_t(X)}{\partial t} = -\eta \Theta_\infty(X,X) \nabla_{f_t(\theta,X)} \mathcal{L}. \tag{4.7}$$

If we use an MSE loss, $\mathcal{L}(y, f) = \frac{1}{2}\|y - f_t(\theta, x)\|_2^2$, the Equation (4.7) becomes a linear model with a solution,

$$f_t(\theta, X) = (\mathbf{I} - e^{-\eta\Theta_\infty(X,X)t})Y + e^{-\eta\Theta_\infty(X,X)t} f_0(\theta, X). \tag{4.8}$$

4.3 Theoretical results

4.3.1 An Orthogonally Initialized Network at Initialization

As stated in [66, 78], the pre-activation h_i^l 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 [36]. We extend this result to the orthogonal initialization:

Theorem 4.1. *Consider a FCN of the form (4.1) at orthogonal initialization, with a Lipschitz activation ϕ, and in the limit as $n_1, ..., n_{L-1} \to \infty$, the pre-activations h_i^l at each layer tend to an i.i.d. Gaussian distribution of covariance Σ^l which can be computed recursively by:*

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

For a CNN of the form (4.2) at orthogonal initialization, and in the limit as channels tend to be infinity, the pre-activations $h_{i,\alpha}^l$ tend to Gaussian processes of covariance $\Sigma_{\alpha,\alpha'}^l$, which is defined recursively by:

$$\Sigma_{\alpha,\alpha'}^1(x, x') = \frac{\sigma_w^2}{n_0(2k+1)} \sum_{\beta=-k}^{k} x_{\alpha+\beta}^T x'_{\alpha'+\beta} + \sigma_b^2$$
$$\Sigma_{\alpha,\alpha'}^l(x, x') = \frac{\sigma_w^2}{(2k+1)} \sum_{\beta=-k}^{k} \Big[\mathbb{E}_{f\sim\mathcal{N}\left(0,\Sigma_{\alpha+\beta,\alpha'+\beta}^{l-1}\right)}$$
$$[\phi(f(x_{\alpha+\beta}))\phi(f(x'_{\alpha'+\beta}))]\Big] + \sigma_b^2.$$
$$\Sigma^L(x, x') = \sum_{\alpha} \delta_{\alpha,\alpha'} \Big[\mathbb{E}_{f\sim\mathcal{N}\left(0,\Sigma_{\alpha,\alpha'}^{L-1}\right)}[\phi(f(x_\alpha))\phi(f(x'_{\alpha'}))]\Big]$$

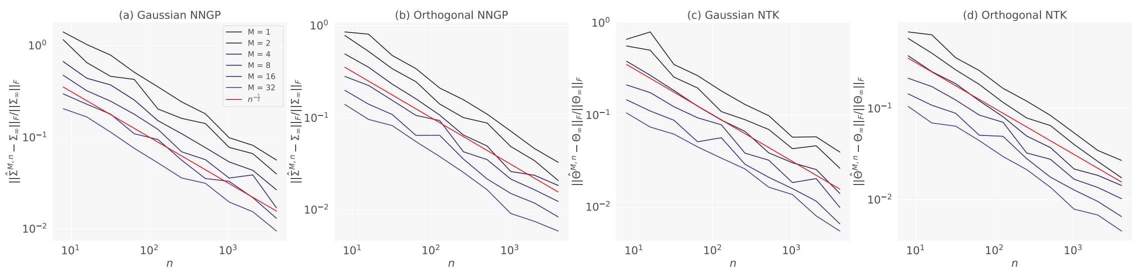

Figure 4.1 : (a) Gaussian initialized network of NNGP. (b) Orthogonally initialized network of NNGP. (c) Gaussian initialized network of NTK. (d) Orthogonally initialized network of NTK. All the kernels are consistent with convergence rate of $O(n^{-\frac{1}{2}})$.

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 4.1, neural networks with Gaussian and orthogonal initialization are in correspondence with an identical class of GPs.

4.3.2 The limit of the NTK at initialization

According to [36], the NTK of a network with Gaussian weights is proven to converge with a probability to a deterministic kernel under the infinite-width limit setting. We show that the NTK of an orthogonally initialized network is identical to the one with Gaussian weights in the infinite-width limit.

Theorem 4.2. *Consider a FCN of the form (4.1) at orthogonal initialization, with a Lipschitz activation ϕ, and in the limit as the layers width tend to be infinity, the NTK $\Theta_0^L(x,x')$, converges to a deterministic limiting kernel with high probability:*

$$\Theta_0^L(x,x') \to \Theta_\infty^L(x,x') \otimes \mathbf{I}_{n_L \times n_L}.$$

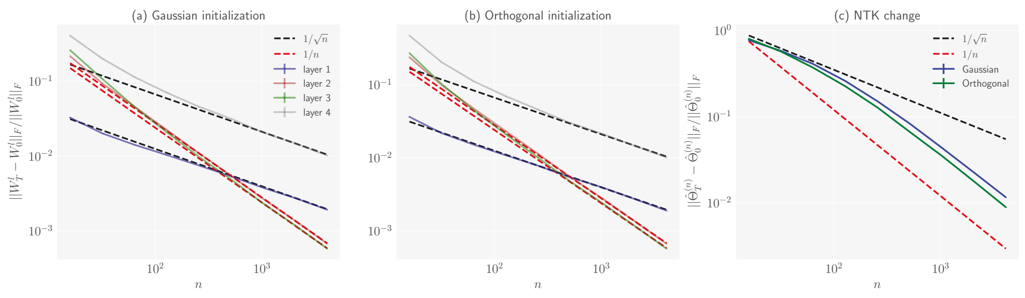

Figure 4.2 : Changes of weights, empirical NTK on a three hidden layer Erf Network. Solid lines correspond to empirical simulation and dotted lines are theoretical predictions, i.e. black dotted lines are $1/\sqrt{n}$ while red dotted lines are $1/n$. (a) weight changes on Gaussian initialized network. (b) weight changes on the orthogonal initialized network. (c) NTK changes on both Gaussian and orthogonal networks.

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

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

where

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

For a CNN of the form (4.2) at orthogonal initialization, and in the limit as channels to be infinity, the NTK $\Theta_0^L(x,x')$, converges to a deterministic limiting kernel:

$$\Theta_0^L(x,x') \to \Theta_\infty^L(x,x') \otimes \mathbf{I}_{n_L\times n_L}.$$

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

$$\begin{aligned}
\Theta^1_{\alpha,\alpha'\infty}(x,x') &= \Sigma^1_{\alpha,\alpha'}(x,x') \\
\Theta^l_{\alpha,\alpha'\infty}(x,x') &= \frac{\sigma_w^2}{(2k+1)}\sum_\beta \big[\dot{\Sigma}^l_{\alpha+\beta,\alpha'+\beta}(x,x') \\
&\qquad \Theta^{l-1}_{\alpha+\beta,\alpha'+\beta\infty}(x,x') + \Sigma^l_{\alpha+\beta,\alpha'+\beta}(x,x')\big] \\
\Theta^L_\infty(x,x') &= \sum_\alpha \delta_{\alpha,\alpha'}\big[\dot{\Sigma}^L_{\alpha,\alpha'}(x,x')\Theta^{L-1}_{\alpha,\alpha'\infty}(x,x') \\
&\qquad + \Sigma^L_{\alpha,\alpha'}(x,x')\big]
\end{aligned}$$

Remark 4.1. *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.*

From the theorem above, the NTK of CNNs propagate differently by averaging over the NTKs regarding the neuron location of the previous layer. According to Theorem 4.2, 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.

We use the Markov chain Monte Carlo (MCMC) estimate of the NNGP and NTK in the finite-width for both Gaussian and orthogonal weights to investigate how these kernels converge. We consider a random inputs generated by a normal distribution. The number training samples is $D = 20$, and dimension of input is $n_0 = 1024$. The depth of networks is $L = 2$ with one hidden layer. We observe the convergence of the NTK as the width of hidden layer n_l increases, as shown in Figure 6.2. For ntk-parameterizaiton of both Gaussian and orthogonal weights, we formulate the convergence rate as $O(1/\sqrt{n})$. The same convergence rate for ntk-parameterization for Gaussian weights has been observed by [37]. Besides, we show that the convergence rate of $O(1/\sqrt{n})$ is also valid for standard parameterization for both Gaussian and orthogonal initialization in the appendix.

4.3.3 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 [36]. We find that the NTK of orthogonally initialized networks have the same property, which is demonstrated below in an asymptotic way,

Theorem 4.3. *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 (4.1) at orthogonal initialization. The learning rate is chosen as $\eta < \eta_{\text{critical}}$ (or gradient flow). Then we have following results for the changes of weights and the NTK,*

$$\sup_{t\geq 0} \frac{\|\theta_t - \theta_0\|_2}{\sqrt{n}}, \ \sup_{t\geq 0} \left\|\hat{\Theta}_t - \hat{\Theta}_0\right\|_F = O(n^{-\frac{1}{2}}), \ \text{as} \quad n \to \infty\,. \tag{4.9}$$

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

For a CNN of the form (4.2) at orthogonal initialization. The learning rate is chosen as $\eta < \eta_{\text{critical}}$ (or gradient flow). Then we have following results for the changes of weights and the NTK,

$$\sup_{t\geq 0} \frac{\|\theta_{\beta,t} - \theta_{\beta,0}\|_2}{\sqrt{n}}, \ \sup_{t\geq 0} \left\|\hat{\Theta}_t - \hat{\Theta}_0\right\|_F = O(n^{-\frac{1}{2}}). \tag{4.10}$$

Jacot et al. [36] proved the stability of NTK under the assumption of global convergence of neural networks, while Lee et al. [37] provided a self-contained proof of both global convergence and stability of NTK simultaneously. In this work, we refer to the proof strategy from [37, 79] and extend it to the orthogonal case, as shown in the proof.

We certificate this theorem empirically. We use three hidden layers ReLU networks with both Gaussian and orthogonal initialization trained by gradient descent.

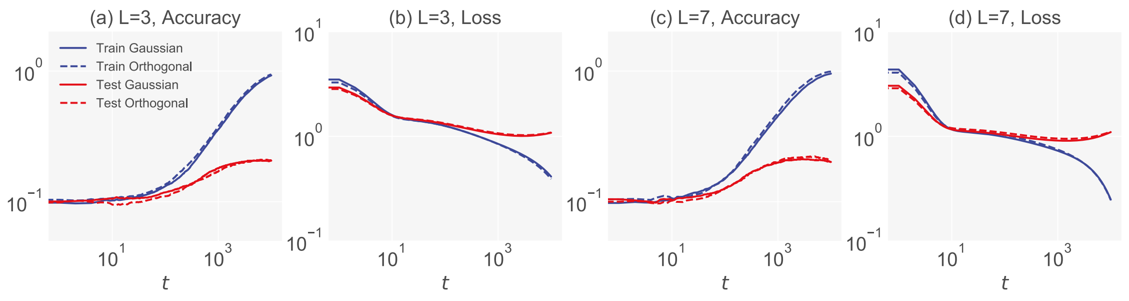

Figure 4.3 : Dynamics of full batch gradient descent on Gaussian and orthogonal initialized networks of $T = 10^4$ steps. Orthogonal networks behaves similarly to dynamics on the corresponding Gaussian networks, for loss and accuracy functions. The dataset is selected from full CIFAR10 with $D = 256$, while MSE loss and tanh fully-connected networks are adopted for the classification task. (a)(b) Network with depth $L = 3$ and width of $n = 400$, with $\sigma_w^2 = 1.5$, and $\sigma_b^2 = 0.01$. (c)(d) Network with depth $L = 7$ and width of $n = 800$, with $\sigma_w^2 = 1.5$, and $\sigma_b^2 = 0.1$. While the solid lines stand for Gaussian weights, dotted lines represent orthogonal initialization.

We choose a small learning rate of $\eta = 1.0$. The dataset in this experiment is from a subset (20 samples) of the MNIST dataset. The variation of weights and empirical NTK is measured after convergence ($T = 2^{15}$). As a result, we found that the both first and last layer exhibit a change of $1/\sqrt{n}$ for weights while second and third layer weights' changes scale as $1/n$. This observation is valid for both Gaussian and orthogonal weights, as shown in Figure 6.3(a)(b). In Figure 6.3(c), we find the change in NTK is consistent with the prediction from our theory. However, we found that the bound is closer to $O(1/n)$ for both Gaussian and orthogonal networks. Note that this is discrepancy has been solved in [40], 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.

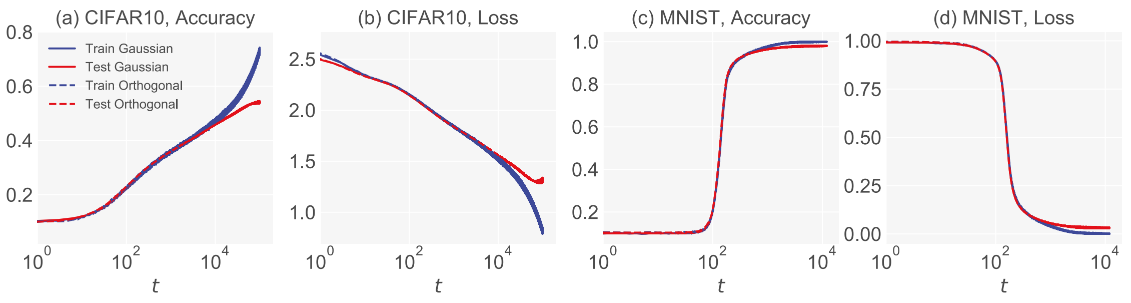

Figure 4.4 : Orthogonally initialized networks behave similarly to the networks with Gaussian initialization in the NTK regime. (a)(b) We adopt the network architecture of depth of $L = 5$, width of $n = 800$, activation of tanh function, with $\sigma_w^2 = 2.0$, and $\sigma_b^2 = 0.1$. The networks are trained by SGD with a learning rate $\eta = 10^{-3}$ with $T = 10^5$. (c)(d) The hyper-parameters are: 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 learning rate $\eta = 10^{-5}$ with $T = 1.2 \times 10^4$ 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.

4.4 Numerical experiments

Our theoretical results indicate that both orthogonal and Gaussian networks should have the same convergence rate at initialization and during training by gradient descent algorithm. This means that two different initializations have similar dynamics for loss and accuracy function during training in the NTK regime. Thus, it is now for us to prove 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.

In the first experiment, summarized in Figure 5.3, we make a comparison loss and accuracy across two different initialization, i.e., Gaussian and orthogonal weights using $D = 256$ samples from the CAFAR10 dataset. To reduce noise, we averaged

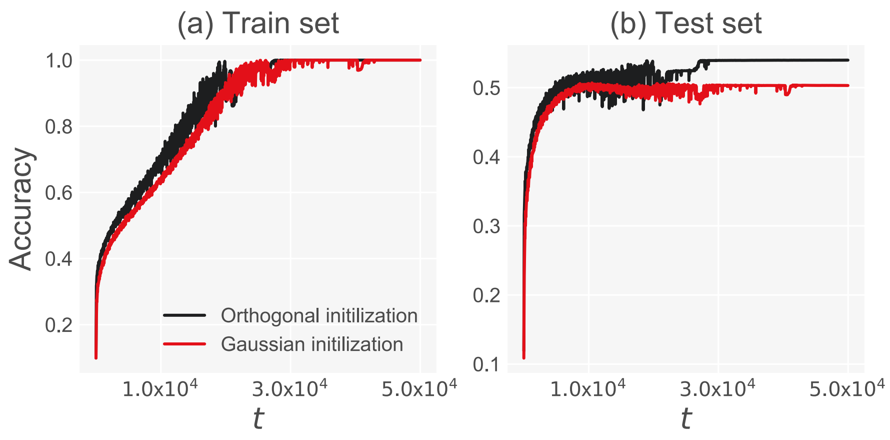

Figure 4.5 : Learning dynamics measured by the optimization and generalization accuracy on train set and test set. The depth is $L = 100$ and width is $n = 400$. Black curves are the results of orthogonal initialization, and red curves are performances of Gaussian initialization. (a) The training speed of an orthogonally initialized network is faster than that of a Gaussian initialized network. (b) On the test set, the orthogonally initialized network not only trains with a higher speed but also ultimately converges to a better generalization performance.

the results over 30 different instantiations of the networks. Figures 5.3(a)(b) show the results of the experiments on the $L = 3$, $n = 400$ network with tanh activation, while figures 5.3(c)(d) display the results for the $L = 7$, $n = 800$ network with tanh activation. All networks are optimized by the vanilla gradient descent. We choose a small learning rate of $\eta = 10^{-4}$ for $T = 10^4$ steps. Consistent with our theoretical findings, the loss and accuracy of both networks are almost the same.

The second experiment is performed to compare the dynamics of two initialization with the result of training and testing on full CIFAR-10 and MNIST dataset, as shown in Figure 5.4. In Figure 5.4(a)(b) we train 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. In Figure 5.4(c)(d) we

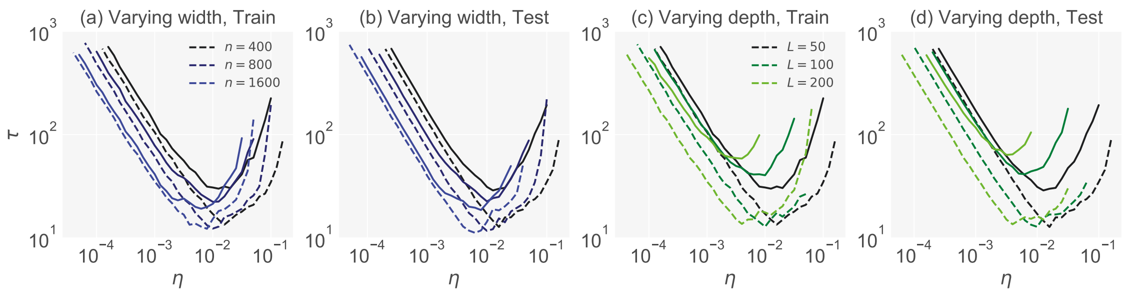

Figure 4.6 : The steps τ as a function of learning rate η of two lines of networks on both train and test dataset. The results of orthogonal networks are marked by dotted lines while those of Gaussian initialization are plotted by solid lines. Networks with varying width, i.e. $n = 400, 800$, and 1600, 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 with a small learning step, the differences become more significant when we increase the learning rate. Besides, the greater the depth of the network, the more significant the difference in performance between orthogonal and Gaussian initialization.

train networks of depth $L = 9$, width $n = 1600$, and activation ReLU function, using PMSProp [80] optimizer with a small learning rate of $\eta = 10^{-5}$ for $T = 1.2 \times 10^4$ steps on MNIST.

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 [24].

Following [24], 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 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 measure the steps τ when it surpass a certain accuracy (in this work we use 0.25). Figure 4.6 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 5.3 suggest a more quantitative analysis of the learning process until convergence. We train networks listed in Figure 5.3 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 4.5. 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 than Gaussian initialization in the large learning rate phase with both train and test set. While it was shown that the network with a large learning rate has many different properties compared to the network trained with a small learning rate [81, 82], our finding can be seen as another effect of networks trained with a large learning rate. 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 [75]. The reason why orthogonally initialized networks can accelerating training may due to that it can achieve tighter distribution of spectrum of NTK matrix and can be studied in the future work.

4.5 Conclusion

This chapter study 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 [38, 76, 83]. Last, we find that orthogonal networks can outperform Gaussian networks in the large learning rate and depth on both train and test sets.

4.6 Proof

This section is dedicated to proving the key results of this chapter, namely Theorem 4.1, Theorem 4.2, and Theorem 4.3 based on a series of lemmas, which describe the asymptotics of neural networks with orthogonal weights at initialization and during training. We prove the Gaussian process behavior of the output function, the limiting deterministic kernel for the orthogonal initialization, and its stability during training in the first three sections.

4.6.1 NNGP at Initialization

In the following proof, our unified hypothesis is that $n_1 = n_2 = \cdots = n_L = n$. That is, $\{W_{ij}\}_{n\times n}$ is an orthogonal mapping. We first cite the following lemma from [84], which describes how the post-activations of one-layer transform by multiplying a random orthogonal matrix.

Lemma 4.1. *Let $(W_{ij})_{n\times n}$ be an orthogonal matrix randomly sampled by the Haar measure of orthogonal matirx. Let B be a $n \times n$ matrix s.t $Tr(BB^T) = n$. Then $Tr(BW)$ converges to a standard Gaussian distribution as the size of the matrix n tends to infinity.*

If we condition on the previous layer's output, the next layer's pre-activations are Gaussian when the width tends to infinity. Therefore if we take the limit of previous layers $n_1, \ldots, n_{l-1} \to \infty$ sequentially, the pre-activation $\{h_i^l\}$ of the l-th layer tends to an i.i.d. Gaussian distribution with respect to the input vector x. However, our goal is to take all the previous layers' width simultaneously. The main technical difficulty is that we lose the independence between different index i in the finite width when we implement orthogonal ensemble. Hence analysis based on the central limit theorem for i.i.d random sequences would not work. Instead, we follow the strategy in [78] to apply a modified version of the exchangeable random sequence central limit theorem. Note that the Haar probability measure is invariant under row and column permutations. This implies that permuting the output index i won't change the joint law of $\{h_i^l(x)\}_{1\le i\le n_l}$.

We will use the following adapted version of the central limit theorem for exchangeable sequences in [78].

Lemma 4.2. *We let $(X_{n,i}; i = 1, 2, ...)$ be an infinitely exchangeable process with zero mean, finite variance σ_n^2. In addition, we set a finite absolute third moment for this* *process. In particular, we assume that the variance has a limit* $\lim_{n\to\infty}\sigma_n^2 = \sigma_*^2$. *Define*

$$S_n = \frac{1}{\sqrt{h(n)}}\sum_{i=1}^{h(n)} X_{n,i}\,, \tag{4.11}$$

where $h : \mathbb{N} \mapsto \mathbb{N}$ *is a strictly increasing function. If the following conditions hold:*

(a) $\mathbb{E}_n[X_{n,1}X_{n,2}] = o(\frac{1}{h(n)})$

(b) $\lim_{n\to\infty}\mathbb{E}_n[X_{n,1}^2 X_{n,2}^2 = \sigma_*^4$

(c) $\mathbb{E}_n[|X_{n,1}|^3] = o(\sqrt{h(n)})$

Then S_n *converges in distribution to* $\mathcal{N}(0,\sigma_*^2)$.

We restate Definition 7 of [78] as follows

Definition 4.1. *The projections are defined in terms of a finite linear projection of the l-th layer's input values without the biases:*

$$S^{(l)}(\mathcal{L},\alpha)[n] = \sum_{(x,i)\in\mathcal{L}} \alpha^{(x,i)}\left[h_i^l(x)[n] - \sigma_b\beta_i^l\right] \tag{4.12}$$

where $\mathcal{L} \subset \mathcal{X}\times\mathbb{N}$ *is a finite set of tuples for data set and indices of pre-activations, with* $\mathcal{X} = (x[i])_{i=1}^{\infty}$. $\alpha \in \mathbb{R}^{|\mathcal{L}|}$ *is a vector parameterising the linear projection. The suffix* $[n]$ *indicates* $\min\{n_1,\ldots,n_l\}$. *Let* $n\to\infty$ *means that the widths of* 1 *to* l *layers tend to infinity simultaneously.*

We reorganize the index and let

$$\gamma_j^l(\mathcal{L},\alpha)[n] := \sum_{(x,i)\in\mathcal{L}} \alpha^{(x,i)} W_{ij}^l \phi(h_i^{l-1}(x))[n]\frac{\sigma_w}{\sqrt{n_{l-1}}}\,, \tag{4.13}$$

so that

$$S^{(l)}(\mathcal{L},\alpha)[n] = \frac{1}{\sqrt{n_{l-1}(n)}}\sum_{j=1}^{n_{l-1}} \gamma_j^l(\mathcal{L},\alpha)[n]\,. \tag{4.14}$$

Since we impose Haar ensemble to $\{W_{ij}\}_{1\leq i,j\leq n_{l-1}}$, it's invariant under the column permutation. This implies that $\{\gamma_j^l(\mathcal{L},\alpha[n])\}_{1\leq j\leq n_{l-1}}$ form an exchangeable sequence with respect to the column index j.

To fit the three conditions of Lemma 4.2, we need the following lemma on the moment calculation of orthogonal random matrices:

Lemma 4.3. *If $(W_{ij})_{n\times n}$ is an orthogonal matrix distributed according to Haar measure, then $\mathbb{E}\left[\prod W_{ij}^{k_{ij}}\right]$ is non-zero if and only if the number of entries from each row and from each column is even. Second and fourth-degree moments are as follows:*

i. For all i,j,

$$\mathbb{E}\left[W_{ij}^2\right]=\frac{1}{n}.$$

ii. For all $i,j,r,s,\alpha,\beta,\lambda,\mu$,

$$\begin{aligned}
&\mathbb{E}\big[W_{ij}W_{rs}W_{\alpha\beta}W_{\lambda\mu}\big]\\
&\qquad=-\frac{1}{(n-1)n(n+2)}\Big[\delta_{ir}\delta_{\alpha\lambda}\delta_{j\beta}\delta_{s\mu}+\delta_{ir}\delta_{\alpha\lambda}\delta_{j\mu}\delta_{s\beta}+\delta_{i\alpha}\delta_{r\lambda}\delta_{js}\delta_{\beta\mu}\\
&\qquad\qquad+\delta_{i\alpha}\delta_{r\lambda}\delta_{j\mu}\delta_{\beta s}+\delta_{i\lambda}\delta_{r\alpha}\delta_{js}\delta_{\beta\mu}+\delta_{i\lambda}\delta_{r\alpha}\delta_{j\beta}\delta_{s\mu}\Big]\\
&\qquad\quad+\frac{n+1}{(n-1)n(n+2)}\Big[\delta_{ir}\delta_{\alpha\lambda}\delta_{js}\delta_{\beta\mu}+\delta_{i\alpha}\delta_{r\lambda}\delta_{j\beta}\delta_{s\mu}+\delta_{i\lambda}\delta_{r\alpha}\delta_{j\mu}\delta_{s\beta}\Big].
\end{aligned}$$

Remark 4.2. *In our scaling setting, the second moment should multiply by n and the fourth moment should multiply by n^2.*

Let $X_{n,i}:=\gamma_j^l(\mathcal{L},\alpha)[n]$, then

$$\begin{aligned}
\mathbb{E}_n[X_{n,i}X_{n,j}]&=\sigma_w^2\cdot\alpha^T\mathbb{E}_n[(W^l\phi(h_i^{l-1}(x)))(W^l\phi(h_i^{j-1}(x)))]\\
&=\sigma_w^2\sum_k\alpha_k^2\mathbb{E}_n[(W_{ki}^l\phi(h_i^{l-1}(x)))(W_{kj}^l\phi(h_i^{j-1}(x)))] \qquad (4.15)
\end{aligned}$$

Note that for $i\neq j$,

$$\mathbb{E}[W_{ki}W_{kj}]=0$$

we have

$$\mathbb{E}_n[X_{n,i}X_{n,j}] = 0.$$

Thus, condition a) of Lemma 4.2 is satisfied.

The rest of the proof goes by induction. We assume that for the $l-1$-th layer, the pre-activations tend to independent Gaussian processes as the previous layers tend to infinite width simultaneously. Then we check that the moment conditions b) and c) of Lemma 4.2 for the l-th layer under the inductive hypothesis and Lemma 4.3. We first calculate the covariance formula non-rigorously where we take the limit sequentially. Since the recursion formula of the covariance doesn't depend on how we take the limit.

We impose an assumption on the nonlinear activation function $\phi(x)$:

Definition 4.2 (Lipschitz nonlinearity). *A nonlinearity ϕ is said to obey the the Lipschitz property if there exist $c, m \geq 0$ such that the following inequality holds*

$$|\phi(x)| \leq c + m|x| \ \ \forall x \in \mathbb{R}\,. \tag{4.16}$$

Proposition 4.1. *For a network of depth L at initialization. We use a Lipschitz activation ϕ which is defined in Definition 4.2, and as $n_1, ..., n_{l-1} \to \infty$ sequentially, the pre-activations h_i^l, for $i = 1, ..., n_l$, tend to i.i.d centered Gaussian processes specified by the covariance $\{\Sigma^l\}_{1\leq l\leq L}$, where Σ^l is defined recursively by:*

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

taking the expectation with respect to a centered Gaussian process of covariance Σ^{l-1}

denoted by f. In the CNN case, $\Sigma^l_{\alpha,\alpha'}$ is defined recursively by:

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

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

where α is the convolution index.

Proof. We show the result in the framework of NTK parameterization, while the argument for standard parameterization can be derived in the same way. In the FCN case, When $L = 1$, there are no hidden layers and h^1_i has the form:

$$h^1_i = \sum_{j=1}^{n_0} \frac{\sigma_w}{\sqrt{n_0}} W^1_{ij} x^0_j + \sigma_b \beta^1_i.$$

Then we check the variance Σ^1 of output layer h^l_i. By Lemma 4.3,

$$\mathbb{E}\big[W^1_{ij}\big] = 0, \quad \mathbb{E}\big[W^1_{ij} W^1_{kl}\big] = \frac{1}{n_0}[n_0 \delta_{ik}\delta_{jl}] = \delta_{ik}\delta_{jl}.$$

We note that the fraction $\frac{1}{n_0}$ is from the scaling of the orthogonal distribution, while the term n_0 comes from the initialization. Thus we can compute the covariance of the first layer explicitly:

$$\begin{aligned}\Sigma^1 =& \mathbb{E}\big[h^1_i h^1_i\big] = \mathbb{E}\big[(\sum_{j=1}^{n_0} \frac{\sigma_w}{\sqrt{n_0}} W^1_{ij} x^0_j + \sigma_b \beta^1_i)(\sum_{j'=1}^{n_0} \frac{\sigma_w}{\sqrt{n_0}} W^1_{ij'} x^0_{j'} + \sigma_b \beta^1_i)\big] \\ =& \frac{\sigma_w^2}{n_0} \mathbb{E}\big[\sum_{j=1}^{n_0}\sum_{j'=1}^{n_0} W^1_{ij} x^0_j W^1_{ij'} x^0_{j'}\big] + \sigma_b^2 = \frac{\sigma_w^2}{n_0} x^T x + \sigma_b^2.\end{aligned}$$

The next step is to use the induction method. Consider an l-network as the function mapping the input to the pre-activations h^l_i. The induction hypothesis gives us that as $n_1, ..., n_{l-2} \to \infty$ sequentially, the pre-activations h^{l-1}_i tend to i.i.d Gaussian processes with covariance Σ^{l-1}. Then the inputs of the l-th layer are governed by:

$$h^l_i = \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_{j=1}^{n_{l-1}} W^l_{ij} x^{l-1}_j + \sigma_b \beta^l_i,$$

where $x_j^{l-1} := \phi(h_j^{l-1})$. When the input $x = x'$, let

$$B = \sqrt{\frac{n_{l-1}}{\sum_{i=1}^{n_{l-1}} (x_i^{l-1})^2}} \begin{bmatrix} x_1^{l-1} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ x_{n_{l-1}}^{l-1} & 0 & \cdots & 0 \end{bmatrix},$$

then $tr(BB^T) = n_{l-1}$. By Lemma 4.1, we have that $lim_{n\to\infty} \sum_{j=1}^{n_{l-1}} \frac{1}{\sqrt{n_{l-1}}} W_{ij}^l x_j^{l-1}$ tends to $\mathcal{N}(0, lim_{n\to\infty} \frac{\sum_{i=1}^{n_{l-1}} (x_i^{l-1})^2}{n_{l-1}})$. As a result, $\{h_i^l\}$ are centered Gaussian variables. With the help of Lemma 4.3, we get the following covariance expression between two input data x and x':

$$\Sigma^l(x,x')[n_{l-1}] = \frac{\sigma_w^2}{n_{l-1}} \sum_j \phi(h_j^{l-1}(x))\phi(h_j^{l-1}(x')) + \sigma_b^2.$$

Since h_j^{l-1}, h_k^{l-1} are independent for $j \neq k$ by the inductive hypothesis, we have

$$Var(\Sigma^l(x,x')) = \frac{\sigma_w^4}{n_{l-1}^2} \sum_j \mathbb{E}\big[\phi(h_j^{l-1}(x))^2 \phi(h_j^{l-1}(x'))^2\big] - \frac{\sigma_w^4}{n_{l-1}^2} \sum_j \mathbb{E}\big[\phi(h_j^{l-1}(x))\phi(h_j^{l-1}(x'))\big]^2.$$

By the symmetry of the underlying index j, we can choose index 1 as a representative:

$$Var(\Sigma^l(x,x')) = \frac{\sigma_w^4}{n_{l-1}} \mathbb{E}\big[\phi(h_1^{l-1}(x))^2 \phi(h_1^{l-1}(x'))^2\big] - \frac{\sigma_w^4}{n_{l-1}} \mathbb{E}\big[\phi(h_1^{l-1}(x))\phi(h_1^{l-1}(x'))\big]^2.$$

Since $\mathbb{E}\big[\phi(h_1^{l-1}(x))^2\phi(h_1^{l-1}(x'))^2\big] - \mathbb{E}\big[\phi(h_1^{l-1}(x))\phi(h_1^{l-1}(x'))\big]$ is bounded, by the Chebyshev's inequality, the covariance kernel tends in probability to its expectation,

$$\Sigma^l(x,x') := \lim_{n_{l-1}\to\infty} \Sigma^l(x,x')[n_{l-1}] = \sigma_w^2 \mathbb{E}_{f\sim\mathcal{N}(0,\Sigma^{l-1})}[\phi(f(x))\phi(f(x'))] + \sigma_b^2.$$

We still need to verify the independence of h_i^l, h_j^l for $i \neq j$. This follows from computing the covariance between h_i^l, h_j^l:

$$\lim_{n_{l-1}\to\infty} Cov(h_i^l(x) h_j^l(x')) = \sum_k \sum_l W_{ik}^l \phi(h_k^{l-1}(x)) W_{jl}^l \phi(h_l^{l-1}(x'))$$

Note that we ignore the bias b_i^l, since they are independent with the weight parameters $\{W_{ij}^l\}$. By Lemma 4.3,

$$\mathbb{E}\big[W_{ik}W_{jl}\big] = 0, for \;\; i \neq j.$$

Therefore we have,

$$\lim_{n_{l-1}\to\infty} Cov(h_i^l(x)h_j^l(x')) = 0.$$

Since we know that as $n_{l-1} \to \infty$, $h_i^l(x)$, $h_j^l(x')$ are Gaussian variables, zero covariance means they are independent.

In the CNN case, we have an extra convolution index α. Since $\mathbb{E}[(W_\alpha^l)(W_{\alpha'}^l)] = 0$ for $\alpha \neq \alpha'$, we have

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

□

With the explicit covariance formula in hand, we can prove by induction that the limiting covariance of the finite width output functions match with the covariance of the infinite Gaussian processes obtained above.

Proposition 4.2. *Let $\sigma^2(l,\mathcal{L},\alpha)[n]$ be the variance of the random variable $\gamma_j^{(l)}(\mathcal{L},\alpha)[n]$, then*

$$\lim_{n\to\infty} \sigma^2(l,\mathcal{L},\alpha)[n] = \sigma^2(l,\mathcal{L},\alpha)[\infty]\,, \tag{4.17}$$

where

$$\sigma^2(l,\mathcal{L},\alpha)[\infty] = \alpha^T[\Sigma^l(x,x') - \sigma_b^2]\alpha. \tag{4.18}$$

Proof.

$$\sigma^2(l,\mathcal{L},\alpha)[n] = \frac{\sigma_w^2}{n_{l-1}}\alpha^T\mathbb{E}[W^l\phi(h_1^{l-1})[W^l\phi(h_1^{l-1})^T]\alpha.$$

By Lemma 4.3 and independence form the layer l with $(l-1)$ layer, we have

$$\mathbb{E}[W^l_{i1}\phi(h_1^{l-1})[W^l_{j1}\phi(h_1^{l-1})^T] = \frac{\sigma_w^2}{n_{l-1}}\delta(i=j)\mathbb{E}[\phi(h_1^{l-1}(x_i)[n])\phi(h_1^{l-1}(x_j)[n])].$$

Once we prove that $\phi(h_1^{l-1}(x_i)[n])\phi(h_1^{l-1}(x_j)[n])$ is uniformly integrable with respect to n, the proof goes exactly the same as the proof of Lemma 17 in [78]. To build the uniform integrability, we need the Lipschitz nonlinearity property of ϕ.

$$\mathbb{E}[\phi(h_1^{l-1}[n]_i)\phi(h_1^{l-1}[n]_j)] \leq \mathbb{E}[(c+m|h_1^{l-1}(x_i)[n])| \cdot (c+m|h_1^{l-1}(x_j)[n])|].$$

To claim we have a uniform bound for the right hand side, it suffices to show that $\mathbb{E}[|h_1^{l-1}(x_i)[n]|]$ and $\mathbb{E}[|h_1^{l-1}(x_i)[n]h_1^{l-1}(x_j)[n]|]$ are uniformly bounded. By the boundedness of Pearson correlation coefficient, we are down if

$$\mathbb{E}[|(h_1^{l-1}(x)[n])|^2]$$

is uniformly bounded as $n \to \infty$. This can be down by induction.

Since $\{W_{ij}\}$ is a scaled orthogonal matrix,

$$\begin{aligned} n_l\mathbb{E}[|h_1^l(x)[n]|^2 &= \mathbb{E}[\|h_{1:n_l}^l(x)[n]\|^2] \\ &= \mathbb{E}[\mathbb{E}[\|h_{1:n_l}^l(x)[n]\|^2|h_{1:n_{l-1}}^{l-1}(x)[n]] \\ &= \sigma_w^2\mathbb{E}[\|\phi(h_{1:n_{l-1}}^{l-1}(x)[n])\|^2] \end{aligned}$$

Applying the lipschitz nonlinearity property, we get

$$\begin{aligned} \mathbb{E}[\|\phi(h_{1:n_{l-1}}^{l-1}(x)[n])\|^2] &= \mathbb{E}[\sum_{j=1}^{j=n}\phi^2(h_j^{l-1}(x)[n])] && (4.19) \\ &\leq \mathbb{E}[\sum_{j=1}^{j=n}(c+m|h_j^{l-1}(x)[n]|)^2]. && (4.20) \end{aligned}$$

Each term in the expansion of the square is bounded by $\mathbb{E}[|h_j^{l-1}(x)[n]|^2]$, which is uniformly bounded by the inductive hypothesis. According to the fact that the number of terms is probably of order n, thus $\mathbb{E}[|h_1^l(x)[n]|^2$ is uniformly bounded. Therefore, we can safely change the order of limit as in the proof of Lemma 17 in [78]. □

Remark 4.3. *Another way to bound the* $\mathbb{E}[|h_1^l(x)[n]|^2$ *is to apply Lemma 4.3 to the* *recursion relation directly:*

$$\mathbb{E}[|h_1^l(x)[n]|^2] = \frac{\sigma_w^2}{n}\mathbb{E}[\sum_j W_{1j}\phi(h_j^{l-1}(x)[n])]^2.$$

To verify condition (2) and (3) of Lemma 4.2 under the inductive hypothesis, we go through the same procedure as the proof of lemmas 15 and 16 in [78]. We neglect the detail and conclude that by Lemma 4.3 and the inductive hypothesis,

$$S^{(l)}(\mathcal{L},\alpha)[n] = \frac{1}{\sqrt{n_{l-1}(n)}}\sum_{j=1}^{n_{l-1}}\gamma_j^l(\mathcal{L},\alpha)[n] \tag{4.21}$$

$$\xrightarrow{P} \mathcal{N}(0,\sigma^2[\infty]). \tag{4.22}$$

Since it holds for arbitary linear functional α, we have shown that $\{h_i^l(x)\}$ converge to independent Gaussian processes, where the covariance is specified by the recursion formula in Proposition 1. In the CNN case, adding the convolution index α won't change the symmetry of index j and Lemma 4.2 can be extended to CNN in the same way without any nontrivial change. Therefore we have simultaneously proved The theorem 4.1 for both FNN and CNN.

4.6.2 NTK at Initialization

Proof. We show the result in the framework of NTK parameterization, while the argument for standard parameterization can be derived in the same way with a similar result, see the details for the Gaussian initialization in [77]. For the input layer $L = 1$, taking the derivative with respect to W_{ij}^1, b_j^1, we have

$$\begin{aligned}\Theta_{kk'}^1(x,x') &= \frac{\sigma_w^2}{n_0}\sum_{i=1}^{n_0}\sum_{j=1}^{n_0} x_i x_i' \delta_{jk}\delta_{jk'} + \sigma_b^2\sum_{j=1}^{n_0}\delta_{jk}\delta_{jk'} \\ &= \frac{\sigma_w^2}{n_0}x^T x'\delta_{kk'} + \sigma_b^2\delta_{kk'}.\end{aligned}$$

From $(l-1)$-th layer to l-th layer, by the inductive hypothesis, as $n_1,\ldots,n_{l-2}\to\infty$, the pre-activations h_i^{l-1} are i.i.d Gaussian distributions. The covariance Σ^{l-1} and

$\Theta^{l-1}_{ii'}(x,x')$ converges to:

$$(\partial_\theta h^{l-1}_i(x))^T(\partial_\theta h^{l-1}_{i'}(x')) \to \Theta^{l-1}_\infty(x,x')\delta_{ii'}.$$

Now we calculate the NTK for l layer network and divide the parameters into two parts. The first part only involves the parameters of the l-th layer, and the other is the collection of parameters from previous $1,\dots,l-1$ layers. The first part of the NTK is given by

$$\alpha_l := \partial_{w^l} h^l_k(x)\cdot\partial_{w^l} h^l_k(x') + \partial_{b^l} h^l_k(x)\cdot\partial_{b^l} h^l_k(x') = \sum_i \frac{\sigma_w^2}{n_{l-1}}\phi(h^{l-1}_i(x))\phi(h^{l-1}_i(x')) + \sigma_b^2.$$

Note that in the Gaussian initialization,

$$\phi(h^{l-1}_i(x))\phi(h^{l-1}_i(x'))$$

is independent of

$$\phi(h^{l-1}_j(x))\phi(h^{l-1}_j(x')),$$

for $i \neq j$ before taking the $n_{l-1}\to\infty$ limit. Thus for the $\frac{1}{n_{l-1}}$ scaling, we already observe that $\sum_i \frac{\sigma_w^2}{n_{l-1}}\phi(h^{l-1}_i(x))\phi(h^{l-1}_i(x'))$ tends to its mean by the classical law of large numbers. If we take the limit sequentially, since $h^{l-1}_i(x)$ and $h^{l-1}_j(x)$ are independent Gaussian by the previous section, we know that it tends to $\Sigma^l(x,x')$ for the same reason.

If we want to know how α_l behaves when the width tends to infinity simultaneously, the nonlinear activation would break the non-asymptotic independence. So we first assume that we work in the linear network category. Then

$$\begin{aligned}
Var[\alpha_l[n]] &= (\frac{\sigma_w^2}{n_{l-1}[n]})^2 Var[\sum_i h^{l-1}_i(x)\cdot h^{l-1}_i(x')] \\
&= (\frac{\sigma_w^2}{n_{l-1}[n]})^2(\sum_{i=1}^n Var[h^{l-1}_i(x)h^{l-1}_i(x')] \\
&\quad + \sum_{i\neq j} Cov[h^{l-1}_i(x)h^{l-1}_i(x'), h^{l-1}_j(x)h^{l-1}_j(x')]),
\end{aligned}$$

where

$$Cov[h_i^{l-1}(x)h_i^{l-1}(x')h_j^{l-1}(x)h_j^{l-1}(x')] = \mathbb{E}[h_i^{l-1}(x)h_i^{l-1}(x'), h_j^{l-1}(x)h_j^{l-1}(x')] - \mathbb{E}[h_i^{l-1}(x)h_i^{l-1}(x')]\mathbb{E}[h_j^{l-1}(x)h_j^{l-1}(x')].$$

By Lemma 4.3,

$$\mathbb{E}[h_i^{l-1}h_i^{l-1}] = \frac{\sigma_w^2}{n_{l-2}}\sum_{k=1}^{n_{l-2}}\mathbb{E}[\phi^2(h_k^{l-2})].$$

The right-hand side is independent of index i as expected.

$$\mathbb{E}[h_i^{l-1}(x)h_i^{l-1}(x'), h_j^{l-1}(x)h_j^{l-1}(x')] = (\frac{\sigma_w^2}{n_{l-2}[n]})^2\frac{n_{l-2}^2(n_{l-2}+1)}{(n-1)n(n+2)}(\sum_{k=1}^{n_{l-2}}\mathbb{E}[\phi^2(h_k^{l-2})])^2 + (\frac{\sigma_w^2}{n_{l-2}[n]})^2\frac{n_{l-2}^3(n_{l-2}-1)}{(n-1)n(n+2)}\mathbb{E}[\phi^2(h_1^{l-2})\phi^2(h_2^{l-2})]$$

Thus

$$Cov[h_i^{l-1}(x)h_i^{l-1}(x')h_j^{l-1}(x)h_j^{l-1}(x')] \sim O(\frac{1}{n_{l-2}[n]}).$$

This implies that

$$Var[\alpha_l[n]] \sim O(\frac{1}{n}).$$

Let μ_n denote the mean of $\alpha_l[n]$, then by Chebyshev's inequality, for $\forall\epsilon > 0$,

$$P(|\alpha_l[n] - \mu_n| \geq \epsilon) \sim O(\frac{1}{n}).$$

Since $\lim_{n\to\infty}\mu_n = \Sigma_l$, we have

$$P(|\alpha_l[n] - \Sigma_l| \geq \epsilon) \sim O(\frac{1}{n}),$$

for n large enough. In conclusion, we have $\alpha_l[n]$ tends to $\Sigma_l(x, x')$.

In the non-linear activation case, if the activation satisfies definition 4.2, we still have the asymptotic result

$$Var[\alpha_l[n]] \to 0.$$

Therefore, by Chebyshev's inequality, we get

$$P(|\alpha_l[n] - \Sigma_l| \geq \epsilon) \to 0,$$

for $\forall\ \epsilon > 0$.

For the second part, denoting the parameters of the previous $l-1$ layers as $\tilde{\theta}$, we have

$$\partial_{\tilde{\theta}} h_k^l(x) = \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_{i=1}^{n_{l-1}} \partial_{\tilde{\theta}} h_i^{l-1}(x) \dot{\phi}(h_i^{l-1}(x)) W_{ik}^l.$$

By the induction hypothesis, the NTK of $(l-1)$-layer networks $\Theta_{kk'}^{l-1}(x, x')$ converges to a diagonal kernel as $n_1, \ldots, n_{l-2} \to \infty$. we denote the second part of NTK by,

$$\alpha_{l-1}[n] := \frac{\sigma_w^2}{n_{l-1}} \sum_{i,i'=1}^{n_{l-1}} \Theta_{ii'}^{l-1}(x, x') \dot{\phi}(h_i^{l-1}(x)) \dot{\phi}(h_{i'}^{l-1}(x')) W_{ik}^l W_{i'k'}^l.$$

Note that $\{W_{ik}^l W_{i'k'}^l\}$ is independent of $\Theta_{ii'}^{l-1}(x, x')\dot{\phi}(h_i^{l-1}(x))\dot{\phi}(h_{i'}^{l-1}(x'))$, this allows us to prove the result of convergence by induction.

By Lemma 4.3,

$$\mathbb{E}[\alpha_{l-1}[n]] = \frac{\sigma_w^2}{n_{l-1}} \delta_{kk'} \delta_{ii'} \sum_{i,i'=1}^{n_{l-1}} \mathbb{E}[\Theta_{ii'}^{l-1}(x, x') \dot{\phi}(h_i^{l-1}(x)) \dot{\phi}(h_{i'}^{l-1}(x'))],$$

$$Var[\alpha_{l-1}[n]] = \mathbb{E}\big[(\alpha_{l-1} - \mathbb{E}[\alpha_{l-1}])^2\big] = \frac{\sigma_w^4}{n_{l-1}^2} \mathbb{E}\big[\big(\sum_{i,i'=1}^{n_{l-1}} \Theta_{ii'}^{l-1}(x, x')$$

$$\dot{\phi}(h_i^{l-1}(x)) \dot{\phi}(h_{i'}^{l-1}(x')) (W_{ik}^l W_{i'k'}^l - \delta_{kk'}\delta_{ii'})\big)^2\big].$$

Expanding the square and note that

$$\mathbb{E}[W_{ik} W_{i'k'} W_{jk} W_{j'k'}] = 0,$$

if $i \neq j \neq i' \neq j'$. This implies that $n_l(n_{l-1}-1)(n_{l-1}-2)(n_{l-1}-3)$ number of terms in the expansion are zero. Now we reorganize the left terms into three groups:

- $i \neq i'$:

  The only terms in the expansion that survive after taking the expectation are of the form: $W_{ik} W_{i'k'} W_{ik'} W_{i'k}$. The expectation is separated into

$$\mathbb{E}[\Theta_{ii'}^{l-1}(x, x') \dot{\phi}(h_i^{l-1}(x)) \dot{\phi}(h_{i'}^{l-1}(x'))]^2 \cdot \mathbb{E}[W_{ik} W_{i'k'} W_{ik'} W_{i'k}].$$

There are $O(n_{l-1} \cdot (n_{l-1} - 1))$ such terms in the expansion, by Lemma 4.3,

$$\mathbb{E}[W_{ik} W_{i'k'} W_{ik'} W_{i'k}] \sim O(1).$$

We conclude that

$$\frac{\sigma_w^4}{n_{l-1}^2} \mathbb{E}\big[\big(\sum_{i \neq i'} \Theta_{ii'}^{l-1}(x, x') \dot{\phi}(h_i^{l-1}(x)) \dot{\phi}(h_{i'}^{l-1}(x'))(W_{ik}^l W_{ik'}^l)\big)^2\big]$$
$$\sim O(\mathbb{E}[\Theta_{ii'}^{l-1}(x, x') \dot{\phi}(h_i^{l-1}(x)) \dot{\phi}(h_{i'}^{l-1}(x'))]^2).$$

- $(i = i') \neq (j = j')$:

  We need to estimate a fourth order moment

$$\gamma := \mathbb{E}\big[(W_{ik}^l W_{ik'}^l - \delta_{kk'})(W_{i'k}^l W_{i'k'}^l - \delta_{kk'})\big].$$

  If $k = k'$, by Lemma 4.3,

$$\gamma = -\frac{2n_{l-1}^2}{(n_{l-1} - 1) n_{l-1} (n_{l-1} + 2)} + \frac{(n_{l-1} + 1) n_{l-1}^2}{(n_{l-1} - 1) n_{l-1} (n_{l-1} + 2)} - 1 \sim O(\frac{1}{n_{l-1}}).$$

  If $k \neq k'$,

$$\mathbb{E}\big[(W_{ik} W_{ik'} W_{i'k} W_{i'k'})\big] = -\frac{n_{l-1}^2}{(n_{l-1} - 1) n_{l-1} (n_{l-1} + 2)} \sim O(\frac{1}{n_{l-1}}).$$

  In each case, we get that $\gamma \sim O(\frac{1}{n_{l-1}})$. Thus we have

$$\frac{\sigma_w^4}{n_{l-1}^2} \mathbb{E}\big[\sum_{i=1}^{n_{l-1}} \sum_{j=1}^{n_{l-1}} \big(\Theta_{ii}^{l-1} \Theta_{jj}^{l-1}(x, x') \dot{\phi}^2(h_i^{l-1}(x)) \dot{\phi}^2(h_j^{l-1}(x'))\big)$$
$$\big(W_{ik}^l W_{ik'}^l - \delta_{kk'}\big)\big(W_{jk}^l W_{jk'}^l - \delta_{kk'}\big)\big] \sim O(\frac{1}{n_{l-1}}).$$

- $(i = i') = (j = j')$: We have

$$\mathbb{E}\big[(W_{ik}^l W_{ik'}^l - \delta_{kk'})^2\big] = \mathbb{E}[W_{ik}^l W_{ik'}^l]^2 - \delta_{kk'}.$$

  Note that the weights are drawn from the NTK parameterization, and by Lemma 4.3,

$$\mathbb{E}[(W_{ik}^l)^2 (W_{ik'}^l)^2] \sim O(1).$$

Therefore the number of terms with the same index i are of order $O(\frac{1}{n_{l-1}})$ and we get

$$\frac{\sigma_w^4}{n_{l-1}^2}\mathbb{E}\big[\sum_{i=1}^{n_{l-1}}\big(\Theta_{ii}^{l-1}(x,x')\dot{\phi}(h_i^{l-1}(x))\dot{\phi}(h_i^{l-1}(x'))\big)^2\big(W_{ik}^l W_{ik'}^l-\delta_{kk'}\big)^2\big]\sim O(\frac{1}{n_{l-1}}).$$

Combing the three groups, we have

$$Var[\alpha_{l-1}[n]]\sim O(\frac{1}{n_{l-1}})+O(\mathbb{E}[\Theta_{ii'}^{l-1}(x,x')\dot{\phi}(h_i^{l-1}(x))\dot{\phi}(h_{i'}^{l-1}(x'))]^2).$$

By the inductive hypothesis,

$$\mu:=\lim_{n_{l-1}\to\infty}\mathbb{E}[\alpha_{l-1}[n]]=\sigma_w^2\Theta_\infty^{l-1}(x,x')\dot{\Sigma}^l\,(x,x')\,\delta_{kk'}.$$

By Chebyshev's inequality,

$$P(|\alpha_{l-1}[n]-\mu|\geq\epsilon)\leq\frac{2}{\epsilon^2}(O(\frac{1}{n_{l-1}})+O(\mathbb{E}[\Theta_{ii'}^{l-1}(x,x')\dot{\phi}(h_i^{l-1}(x))\dot{\phi}(h_{i'}^{l-1}(x'))]^2)).$$

By the inductive hypothesis and the integrability result from the last section,

$$\mathbb{E}[\Theta_{ii'}^{l-1}(x,x')\dot{\phi}(h_i^{l-1}(x))\dot{\phi}(h_{i'}^{l-1}(x'))]^2\overset{n_{l-1}\to\infty}{\longrightarrow}0.$$

Thus we obtain that second part of the NTK tends to $\sigma_w^2\Theta_\infty^{l-1}(x,x')\dot{\Sigma}^l(x,x')$ if we let the width go to infinity simultaneously.

Combining the two parts, we have

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

In the CNN case, when $L=1$,

$$\Theta_{\alpha,\alpha'}^l(x,x')=\frac{\sigma_w^2}{n_0(2k+1)}\sum_\beta x_{\alpha+\beta}^T x_{\alpha'+\beta}+\sigma_b^2.$$

Here we omit the subscript ∞ for simplicity. Assume the NTK formula is true for the layer of $l-1$, then the first part of the NTK in layer l is given by

$$\alpha_L\to(\frac{\sigma_w^2}{n_0(2k+1)}\sum_\beta\mathbb{E}[\phi(h_{\alpha+\beta}^{l-1}(x))\phi(h_{\alpha'+\beta}^{l-1}(x'))]+\sigma_b^2),$$

by the same moment argument as before. For the second part, we still have

$$\Theta^{l-1}_{(i,\alpha+\beta),(i',\alpha'+\beta)}(x,x') = \delta_{ii'}\Theta^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x').$$

Thus, for the second part,

$$\partial_{\tilde{\theta}} h^l_{k,\alpha}(x) \cdot \partial_{\tilde{\theta}} h^l_{k,\alpha'}(x) \to \sigma_w^2 \Theta^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')\dot{\Sigma}^l_{\alpha+\beta,\alpha'+\beta}(x,x'),$$

since $\mathbb{E}[W_\alpha W_{\alpha'}] = 0$ for $\alpha \neq \alpha'$. We conclude that

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

□

Remark 4.4. *Note that the proof for the NNGP and NTK behavior is mainly based on the moment calculation in Lemma 4.3. Let's take the second moment as an example.* $\mathbb{E}[W_{ij}W_{rs}]$ *can be seen as the correlation between one normalized vector and another normal vector sampled uniformly from the orthogonal complement. From this point view, the moment is independent of the matrix structure, and the results of Lemma 3 can be extended to the case where* $(W_{ij})_{n\times m}$ *satisfies*

$$WW^T = I, \qquad n \leq m.$$

Thus, we believe that the previous theorems also hold without assuming that the weight matrix is a square matrix.

4.6.3 NTK during Training

We can give an upper bound for the change of parameters and NTK at wide width. We fist prove the *local Lipschitzness* of $J(\theta) := \nabla_\theta h^L(\theta) \in \mathbb{R}^{(Dn_L)\times|\theta|}$, where $|\theta|$ is the number of parameters. Once we have the upper bound of $J(\theta)$, the stability comes from proving that the difference is small of the NTK form time t to time $t+1$. As it has been observed in [37], this step is universal of the network structure and also the initialization. Thus, our primary target is construct local Lipschitzness of orthogonal initialization. The Lemma below illustrates the local Lipschitzness of $J(\theta)$ without scale condition for the input layer.

Lemma 4.4. *(local Lipschitzness of $J(\theta)$) $\exists K > 0$ s.t for every $c > 0$, $\exists n_c$ s.t for $n \geq n_c$, we have the following bound:*

$$||J(\theta) - J(\hat{\theta})||_F \leq K||\theta - \hat{\theta}||_2$$
$$||J(\theta)||_F \leq K,$$

for all $|\theta - \hat{\theta}| < \frac{c}{n}$

Proof. We prove the result by induction for the standard parameterization, while the proof for ntk parameterization can be derived in the same way. For $l \geq 1$, let

$$\delta^l(\theta, x) := \nabla_{h^l(\theta,x)} h^L(x) \in \mathbb{R}^{n_L n}$$

$$\delta^l(\theta, X) := \nabla_{h^l(\theta,X)} h^L(X) \in \mathbb{R}^{(n_L D)\times(n_L D)}$$

Let $\theta = \{W^l, b^l\}$, $\hat{\theta} = \{\hat{W}^l, \tilde{b}^l\}$ be two points in $B(\theta_0, \frac{c}{n})$. Since the initialization is to choose an orthogonal matrix, i.e. $W^T W = \sigma_w^2 \mathbf{I}$. If the width n is large enough, we have,

$$||W^l||, \ \ ||\hat{W}^l|| \leq 3\sigma_w \quad \text{for all } l.$$

As in the original proof for Gaussian initialization [37], there is a constant K_1, depending on D, σ_w^2, σ_b^2, and number of layers L s.t with high probability over orthogonal initialization,

$$||x^l(\theta, X)||_2, \qquad ||\delta^l(\theta, X)||_2 \leq K_1$$

$$||x^l(\theta, X) - x^l(\hat{\theta}, X)||_2, \quad ||\delta^l(\theta, X) - \delta^l(\hat{\theta}, X)||_2 \leq K_1||\hat{\theta} - \theta||_2$$

Note: there is a scaling factor $\frac{1}{\sqrt{n}}$ along with $||x^l(\theta, X)||_2$ in the Gaussian case.

Decomposing the $J(\theta)$ into two parts, we have

$$\begin{aligned}
||J(\theta)||_F^2 &= \sum_l ||\nabla_{W^l} h^L(\theta)||_F^2 + ||\nabla_{b^l} L^h(\theta)||_F^2 \\
&= \sum_l \sum_{x\in X} ||x^{l-1}(\theta, x)\delta^l(\theta, x)^T||_F^2 + ||\delta^l(\theta, x)^T||_F^2 \\
&\le \sum_l \sum_{x\in X} (1 + ||x^{l-1}(\theta, x)\delta^l(\theta, x)^T||_F^2) \cdot ||\delta^l(\theta, x)^T||_F^2 \\
&\le \sum_l (1 + K_1^2) \sum_x ||\delta^l(\theta, x)^T||_F^2 \\
&\le 2LK_1^4
\end{aligned}$$

and similarly,

$$\begin{aligned}
&||J(\theta) - J(\tilde{\theta})||_F^2 = \\
&= \sum_l \sum_{x\in X} ||x^{l-1}(\theta, x)\delta^l(\theta, x)^T - x^{l-1}(\tilde{\theta}, x)\delta^l(\theta, x)^T||^2 \\
&+ ||x^{l-1}(\tilde{\theta}, x)\delta^l(\theta, x)^T - x^{l-1}(\tilde{\theta}, x)\delta^l(\tilde{\theta}, x)^T||^2 + ||\delta^l(\theta, x)^T - \delta^l(\tilde{\theta}, x)^T||^2 \\
&\le \sum_l 2K_1^4||\tilde{\theta} - \theta||^2 + K_1^2||\tilde{\theta} - \theta||^2 \\
&\le 3K_1^4 L||\tilde{\theta} - \theta||^2
\end{aligned}$$

□

Note: In our setting, since we want the orthogonal and Gaussian initialization have the same NTK, we need to scale the initialization for the input layer.

If we impose this condition, then,

$$||W^1||_{op}, \quad ||\tilde{W}^1||_{op} \le 3\sigma_w \frac{\sqrt{n}}{\sqrt{n_0}}$$

.

Chapter 5

Implicit bias of deep linear networks in the large learning rate phase

5.1 Introduction

Deep neural networks have been responsible for a variety of breakthroughs and other successes in both supervised and unsupervised learning. And, even after many decades of study, our understandings of the theoretical mechanisms that underlie the power of deep learning are continually being refined and expanded by researchers. Remarkably, recent progress in deep learning theory has shown that optimizing an over-parameterized network with gradient descent can result in very low or even zero training errors [36, 41, 42, 85, 43]. Meanwhile, these over-parameterized networks tend to generalize well to the test set, in a phenomenon known as double descent [86].

Yet, with far more parameters than training samples, or even input dimensions, how can an over-parameterized network possibly withstand the overfitting problem? Among the most promising explanations for this question is implicit bias [52] also referred to as implicit regularization [87]. Specifically, a large body of works studying exponential tailed losses, including logistic and exponential losses, have reported that strong regularization results using maximum margin [52, 53, 54, 60, 61].

Notably, all theoretical results associated with implicit bias are based on the assumption that the learning rate is sufficiently small. At larger learning rates, our theoretical understandings of the properties of optimization and generalization are limited. One thing we do know from real-world observations is that adopting a learning rate annealing scheme with a large initial setting often performs better than the methods prescribing a steady small rate [7, 88]. Lewkowycz et al. [81]'s observation that the local curvature of the loss landscape drops significantly with large learning rates shed some light on this issue, offering a promising direction to explore the training process at a large learning rate typically delivers the best performance in practice.

Hence, following the threads of [81], we sought to characterize training with gradient descent at three different learning rate phases. Treating each rate as a learning scenario or phase, the three are: (i) the lazy phase $\eta < \eta_0$, where the learning rate is small. In these scenarios, the dynamics of the neural network are linearized, and the model converges to a nearby point in parameter space. Also known as lazy training, this case is characterized by the neural tangent kernel [36, 38, 39, 40, 41, 42]. (ii) the catapult phase $\eta_0 < \eta < \eta_1$, where the loss initially grows but then drops until it converges to the solution with a flatter minimum. (iii) the divergent phase $\eta > \eta_1$, where the loss diverges and the model will not train.

The training behaviors in these three phases have been reported in regression settings with mean squared error (MSE) loss. However, it remains unclear whether these same results extend to cross-entropy (logistic) loss. To fill this gap, we have examined the effects of a large learning rate on deep linear networks with both logistic loss and exponential loss. What we find is that, unlike with MSE loss, the characteristics of gradient descent with logistic loss at a large learning rate depend on the separation conditions of the data. The main contributions of this chapter include:

- A theoretical examination characterizing the dynamics of gradient descent with logistic and experiential losses given different learning rates and data separability conditions. We find that the network ultimately finds a flatter minimum in the catapult phase with a large learning rate when the data is non-separable. Such three learning rate phases do not apply to the linearly separable data since the optimum is approaching infinity.

- A theoretical analysis that ranges from a linear predictor to a hidden layer network. By comparing the behavior of convex (Theorem 5.1) and non-convex optimizations (Theorem 5.2), we show that the catapult phase is a phenomenon unique to non-convex optimization.

- Experiments with practical classification tasks that show the best generalization results tend to occur in the catapult phase. Given that infinite-width analysis (i.e., lazy training) typically works to the detriment of generalization and does not explain the practically-observed power of deep learning [89, 76], our results partially fill this gap.

5.2 Background

A large learning rate with SGD training is often set initially to achieve good performance empirically in deep learning [5, 7, 88]. Existing theoretical explanation of the benefit of the large learning rate contributes to two classes. One is that a large learning rate with SGD leads to flat minima [90, 91, 81] and the other is that the large learning rate acts as a regularizer [82]. Especially, Lewkowycz et al. [81] find a large learning rate phase can result in flatter minima without the help of SGD for mean squared loss. In this work, we ask whether large learning rate still has this advantage for logistic loss. We expect this loss function to have a different outcome because the logistic loss is sensitive to the separable property of the data and the loss surface is quite different from that of MSE. For example, it is shown that gradient descent can learn less over-parameterized neural networks with logistic loss than mean squared loss based on separability [92].

5.2.1 Setup

Suppose there is a dataset $\{x_i, y_i\}_{i=1}^n$, with inputs $x_i \in \mathbb{R}^d$ and binary labels $y_i \in \{-1, 1\}$. The risk of classification task follows the form,

$$\mathcal{L} = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i)y_i), \tag{5.1}$$

where $f(x_i)$ is the output of network regarding the input x_i, and ℓ is the loss. In this work, we study two exponential tail losses which are exponential loss $\ell_{\exp}(u) = \exp(-u)$ and logistic loss $\ell_{\log}(u) = \log(1 + \exp(-u))$. The reason we look at these two loss together is that they show the same phenomenon under gradient descent on a linearly separable data set [52]. We adopt gradient descent optimization with learning rate η to minimize empirical risk,

$$w(t+1) = w(t) - \eta \nabla \mathcal{L}(w(t)) = w(t) - \eta \sum_{i=1}^{n} \ell'(f(x_i)y_i). \tag{5.2}$$

5.2.2 Separation Conditions of Dataset

It is known that landscapes of cross-entropy loss on linearly separable data and non-separable data are different. Thus the separation condition plays an important role in understanding the dynamics of gradient descent in terms of learning rate. To build towards this, we define the two classes of separation conditions and review existing results for loss landscapes of a linear predictor in terms of separability.

Assumption 5.1. *The dataset is linearly separable, i.e., there exists a weight w_* so that the dataset can be separated by a linear predictor.*

Assumption 5.2. *The dataset is non-separable, i.e., there is no weight w_* so that the dataset can be separated by a linear predictor.*

Linearly separable. Consider the data under assumption 5.1, one can examine that the loss of a linear predictor, i.e., $f(x) = w^T x$, is β-smooth convex. On the other hand, wecan say that the global minimum is at infinity. The implicit bias of gradient descent training on a network with a small learning rate ($\eta < \frac{2}{\beta}$) in this phase has been studied by [52]. They show the network's output function can converge towards the maximum margin (hard margin SVM) solution, which implies the gradient descent method itself will find a proper solution with an implicit regularization instead of picking up a random solver. If one increases the learning rate until it exceeds $\eta < \frac{2}{\beta}$, then the result of converging to maximum margin will not be guaranteed, though loss can still converge to a global minimum.

Non-separable. Suppose we consider the data under assumption 5.2, which is not linearly separable. In this case, we can find that the empirical risk of a linear predictor on this data is α-strongly convex, and the global minimum is finite. In this case, given an appropriate small learning rate ($\eta < \frac{2}{\beta}$), the gradient descent converges towards the unique finite solution. When the learning rate is large enough, i.e., $\eta > \frac{2}{\alpha}$, we can rigorously show that gradient descent update with this large learning rate leads to risk exploding or saturating.

We formally construct the relationship between loss surfaces and learning dynamics of gradient descent with respect to different learning rates on the two classes of data through the following proposition,

Proposition 5.1. *For a linear predictor $f = w^T x$, along with a loss $\ell \in \{\ell_{\exp}, \ell_{\log}\}$.*

1 *Under Assumption 5.1, the empirical loss is β-smooth. Then,*

$$\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \leq 0, \quad \lim_{t\to\infty} \mathcal{L}(w_t) = 0, \text{ with } \eta < \frac{2}{\beta}$$

2 *Under Assumption 5.2, the empirical loss is β-smooth and α-strongly convex,*

where $\alpha \leq \beta$. Then,

$$\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \leq 0, \quad \lim_{t\to\infty} \mathcal{L}(w_t) = G_0, \text{ with } \eta < \frac{2}{\beta}$$
$$\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \geq 0, \quad \lim_{t\to\infty} \mathcal{L}(w_t) = G_1, \text{ with } \eta > \frac{2}{\alpha}$$

where G_0 is the value of a global minimum while $G_1 = \infty$ for exploding situation or $G_0 < G_1 < \infty$ when saturating.

5.3 Theoretical results

5.3.1 Convex Optimization

The Hessian of the logistic and exponential loss with respect to the linear predictor is non-constant. Moreover, the estimated β-smooth convexity and α-strongly convexity vary across different finite bounded subspace. As a result, the learning rate threshold in Proposition 5.1 is not detailed in terms of optimization trajectory. However, we can obtain more elaborate thresholds of the learning rate for linear predictor by considering the *degeneracy assumption*:

Assumption 5.3. *The dataset contains two data points where they have same feature and opposite label, that is*

$$(x_1 = 1, y_1 = 1) \quad \textit{and} \quad (x_2 = 1, y_2 = -1).$$

We call this assumption the *degeneracy assumption* since the features from opposite label degenerate. Without loss of generality, we simplify the dimension of data and fix the position of the feature. Note that this assumption can be seen as a special case of non-separable data. There is a work theoretically characterizing general non-separable data [54], and we leave the analysis of this setting for the large learning rate to future work. Thanks to the symmetry of the risk function in space at the basis of *degeneracy assumption*, we can construct the exact dynamics of empirical risk with respect to the whole learning rate space.

Theorem 5.1. *For a linear predictor $f = w^T x$ equipped with exponential (logistic) loss under assumption 5.3, there is a critical learning rate that separates the whole learning rate space into two (three) regions. The critical learning rate satisfies*

$$\mathcal{L}'(w_0) = -\mathcal{L}'(w_0 - \eta_{\text{critical}}\mathcal{L}'(w_0)),$$

where w_0 is the initial weight. Moreover,

1 For exponential loss,

$$\eta < \eta_{\text{critical}} : \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) < 0, \ \lim_{t\to\infty} \mathcal{L}(w_t) = 1;$$
$$\eta = \eta_{\text{critical}} : \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) = 0, \ \lim_{t\to\infty} \mathcal{L}(w_t) = \mathcal{L}(w_0);$$
$$\eta > \eta_{\text{critical}} : \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) > 0, \ \lim_{t\to\infty} \mathcal{L}(w_t) = \infty.$$

2 For logistic loss,

$$\eta < 8 : \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) < 0, \ \lim_{t\to\infty} \mathcal{L}(w_t) = \log(2);$$
$$8 \le \eta < \eta_{\text{critical}} : \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \le 0, \ \lim_{t\to\infty} \mathcal{L}(w_t) = \mathcal{L}(w_*) < \mathcal{L}(w_0);$$
$$\eta \ge \eta_{\text{critical}} : \mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \ge 0, \ \lim_{t\to\infty} \mathcal{L}(w_t) = \mathcal{L}(w_*) \ge \mathcal{L}(w_0).$$

where w_* satisfies $-w_* = w_* - \frac{\eta}{2}\frac{\sinh(w_*)}{1+\cosh(w_*)}$.

We demonstrate the gradient descent dynamics with a degenerate and non-separable example.

Example 5.1. *Consider optimizing $\mathcal{L}(w)$ with dataset $\{(x_1 = 1, y_1 = 1)$ and $(x_2 = 1, y_2 = -1).\}$ using gradient descent with constant learning rates. Figure 5.1(a,c) show the dependence of different dynamics on the learning rate η for exponential and logistic loss respectively.*

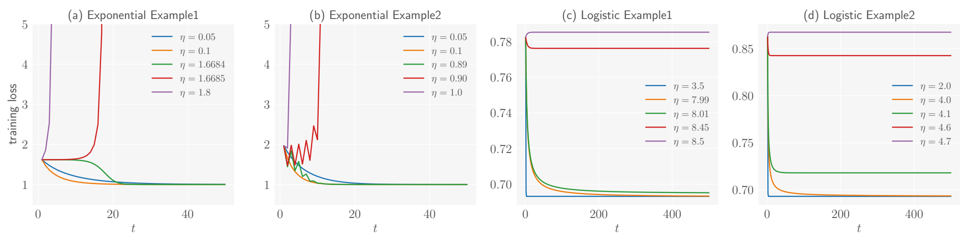

Figure 5.1 : Dependence of dynamics of training loss on the learning rate for linear predictor, with (a,b) exponential loss and (c,d) logistic loss on Example 5.1 and 5.2. (a,c) The experimental learning curves are consistent with the theoretical prediction, and the critical learning rates are $\eta_{\text{critical}} = 1.66843$ and $\eta_{\text{critical}} = 8.485$ respectively. (b,d) For non-separable data, the dynamics of training loss regarding the learning rate for non-separable data are similar to those of a degenerate case. Hence the critical learning rates can be approximated by $\eta_{\text{critical}} = 0.895$ and $\eta_{\text{critical}} = 4.65$ respectively.

Example 5.2. *Consider optimizing $\mathcal{L}(w)$ with dataset $\{(x_1 = 1, y_1 = 1), \quad (x_2 = 2, y_2 = -1) \quad and \quad (x_3 = -1, y_3 = 1).\}$ using gradient descent with constant learning rates. Figure 5.1(b,d) show the dependence of different dynamics on the learning rate η for exponential and logistic loss, respectively.*

Remark 5.1. *The dataset considered here is an example of a non-separable case, and the dynamics of loss behave similarly to those with Example 5.1. We use this example to show that our theoretical results on the degenerate data can be extended to the non-separable data empirically.*

5.3.2 Non-convex Optimization

The typical non-convex optimization model in machine learning is deep networks. For simplicity, we focus on a two-layer linear network, and the information propagation in these networks is governed by

$$f(x) = m^{-1/2} w^{(2)} w^{(1)} x, \tag{5.3}$$

where m is the width, i.e., number of neurons in the hidden layer, $w^{(1)} \in \mathbb{R}^{m\times d}$ and $w^{(2)} \in \mathbb{R}^m$ are the parameters of the model. Taking the exponential loss as an example, the gradient descent dynamics follow,

$$\begin{aligned} w^{(1)}_{t+1} &= w^{(1)}_t - \frac{1}{n}\frac{\eta}{m^{1/2}}(-e^{-y_\alpha f_t(x_\alpha)}) w^{(2)}_t x_\alpha y_\alpha, \\ w^{(2)}_{t+1} &= w^{(2)}_t - \frac{1}{n}\frac{\eta}{m^{1/2}}(-e^{-y_\alpha f_t(x_\alpha)}) w^{(1)}_t x_\alpha y_\alpha, \end{aligned} \tag{5.4}$$

where we use the Einstein summation convention to simplify the expression and will apply this convention in the following derivation.

We introduce the neural tangent kernel, an essential element for the evolution of output function in equation 5.8. The neural tangent kernel (NTK) is originated from [36] and formulated as,

$$\Theta_{\alpha\beta} = \frac{1}{m}\sum_{p=1}^{P} \frac{\partial f(x_\alpha)}{\partial_{\theta_p}} \frac{\partial f(x_\beta)}{\partial_{\theta_p}}. \tag{5.5}$$

where P is the number of parameters. For a two-layer linear neural network, the NTK can be written as,

$$\Theta_{\alpha\beta} = \frac{1}{mn}\big((w^{(1)}x_\alpha)(w^{(1)}x_\beta) + (w^{(2)})^2(x_\alpha x_\beta)\big). \tag{5.6}$$

Here we use normalized NTK which is divided by the number of samples n. Under the degeneracy assumption 5.3, the loss function becomes $\mathcal{L} = \cosh(m^{-1/2}w^{(2)}w^{(1)})$. Then the equation 5.4 reduces to

$$\begin{aligned} w^{(1)}_{t+1} &= w^{(1)}_t - \frac{\eta}{m^{1/2}} w^{(2)}_t \sinh(m^{-1/2} w^{(2)}_t w^{(1)}_t), \\ w^{(2)}_{t+1} &= w^{(2)}_t - \frac{\eta}{m^{1/2}} w^{(1)}_t \sinh(m^{-1/2} w^{(2)}_t w^{(1)}_t). \end{aligned} \tag{5.7}$$

The updates of output function f_t and the eigenvalue of NTK λ_t, which are both scalars in our setting:

$$\begin{aligned} f_{t+1} &= f_t - \eta\lambda_t \tilde{f}_{t\mathrm{exp}} + \frac{\eta^2}{m} f_t \tilde{f}^2_{t\ \mathrm{exp}}, \\ \lambda_{t+1} &= \lambda_t - \frac{4\eta}{m} f_t \tilde{f}_{t\mathrm{exp}} + \frac{\eta^2}{m}\lambda_t \tilde{f}^2_{t\ \mathrm{exp}}. \end{aligned} \tag{5.8}$$

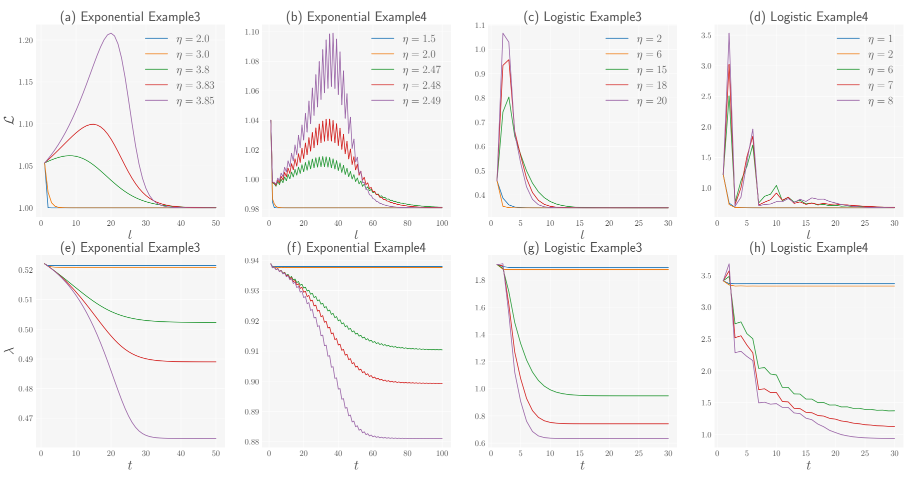

Figure 5.2 : Dependence of dynamics of training loss and maximum eigenvalue of NTK on the learning rate, with (a,b,e,f) exponential loss and (c,d,g,h) logistic loss on Example 5.3 and 5.4. (a,b,c,d) The loss increases at the beginning and converges to a global minimum. (e,f,g,h) The maximum eigenvalue of NTK converges to a value which is lower than its initial position.

where $\tilde{f}_{t\exp} := \sinh(f_t)$ while $\tilde{f}_{t\log} := \frac{\sinh(f_t)}{1+\cosh(f_t)}$ for logistic loss.

We have introduced the catapult phase where the learning rate is larger than a threshold. In this phase, the loss increases at the beginning and then drops until it converges to a global minimum. In the following theorem, we prove the existence of the catapult phase on the degenerate data with exponential and logistic loss.

Theorem 5.2. *Under appropriate initialization and assumption 5.3, there exists a catapult phase for both the exponential and logistic loss. More precisely, when η belongs to this phase, there exists a $T > 0$ such that the output function f_t and the eigenvalue of NTK λ_t update in the following way:*

i. $\mathcal{L}_t$ keeps increasing when $t < T$.

ii. After the T step and its successors, the loss decreases, which is equivalent to:

$$|f_{T+1}| > |f_{T+2}| \geq |f_{T+3}| \geq \dots .$$

iii. The eigenvalue of NTK keeps dropping after the T steps:

$$\lambda_{T+1} > \lambda_{T+2} \geq \lambda_{T+3} \geq \dots.$$

Moreover, we have the inverse relation between the learning rate and the final eigenvalue of NTK: $\lambda_\infty \leq \lim_{t\to\infty} \frac{4f_t}{\eta \tilde{f}_{t_{\exp}}}$ with exponential loss or $\lambda_\infty \leq \lim_{t\to\infty} \frac{4f_t}{\eta \tilde{f}_{t_{\log}}}$ with logistic loss.

We demonstrate that the catapult phase can be found in both degenerate and non-separable data through the following examples. The weights matrix is initialized by i.i.d. Gaussian distribution, i.e. $w^{(1)}, w^{(2)} \sim \mathcal{N}(0, \sigma_w^2)$. For exponential loss, we adopt the setting of $\sigma_w^2 = 0.5$ and $m = 1000$ while we set $\sigma_w^2 = 1.0$ and $m = 100$ for logistic loss.

Example 5.3. *Consider optimizing $\mathcal{L}(w)$ using one hidden layer linear networks with dataset $\{(x_1 = [1,0], y_1 = 1)$ and $(x_2 = [1,0], y_2 = -1).\}$ and exponential (logistic) loss using gradient descent with constant learning rate. Figure 5.2(a,c,e,g) show how different choices of learning rate η change the dynamics of the networks with exponential and logistic loss.*

Example 5.4. *Consider optimizing $\mathcal{L}(w)$ using one hidden layer linear networks with dataset $\{(x_1 = [1,1], y_1 = -1)$, $(x_2 = [1,-1], y_1 = 1)$, $(x_3 = [-1,-2], y_1 = 1)$ and $(x_4 = [-1,1], y_4 = 1).\}$ and exponential (logistic) loss using gradient descent with constant learning rate. Figure 5.2(b,d,f,h) show how different choices of learning rate η change the dynamics of networks with exponential and logistic loss.*

As Figure 5.2 shows, in the *catapult phase*, the eigenvalue of the NTK decreases to a lower value than its initial point, while it keeps unchanged in the *lazy phase* where the learning rate is small. For MSE loss, the lower value of the NTK indicates the flatter curvature given the training loss is low [81]. Yet, it is unknown whether the

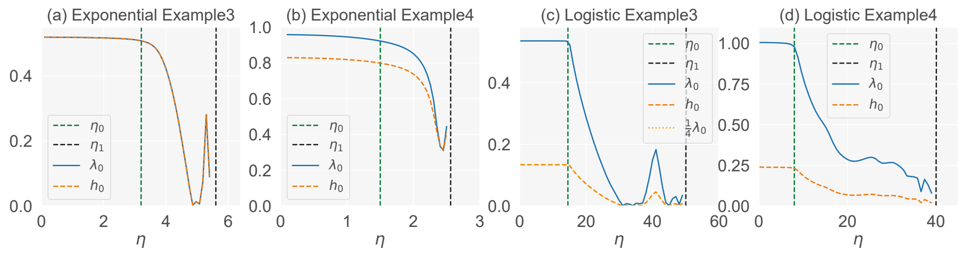

Figure 5.3 : Top eigenvalue of NTK (λ_0) and Hessian (h_0) measured at $t = 100$ as a function of the learning rate, with (a,b) exponential loss and (c,d) logistic loss on Example 5.3 and 5.4. The green dashed line $\eta = \eta_0$ represents the boundary between the lazy phase and catapult phase, while black dashed line $\eta = \eta_1$ separates the other two phases. The setting are: $\sigma_w^2 = 0.5$ and $m = 100$ for exponential loss, and the setting for logistic loss is $\sigma_w^2 = 0.5$ and $m = 200$. (a,c) The curves of maximum eigenvalue of NTK and Hessian coincide as predicted by the corollary 5.1. (b,d) For the non-separable data, the trend of the two eigenvalue curves is consistent with the change of learning rate.

aforementioned conclusion can be applied to exponential and logistic loss. Through the following corollary, we show that the Hessian is equivalent to the NTK when the loss converges to a global minimum for degenerate data.

Corollary 5.1 (Informal)**.** *Consider optimizing $\mathcal{L}(w)$ with one hidden layer linear network under assumption 5.3 and exponential (logistic) loss using gradient descent with a constant learning rate. For any learning rate that loss can converge to the global minimum, the larger the learning rate, the flatter curvature that gradient descent will achieve at the end of training.*

We demonstrate the flatter curvature can be achieved in the catapult phase through Example 5.3 and 5.4, using the code provided by [93] to measure Hessian, as shown in Figure 5.3. In the lazy phase, both curvature and eigenvalue of NTK are independent of the learning rate at the end of training. However, in the *catapult* *phase*, the curvature decreases to the value smaller than that in the lazy phase. In conclusion, the NTK and Hessian have similar behavior at the end of training on non-separable data.

Finally, we compare our results for the catapult phase to the result with MSE loss and show the summary in Table 5.1.

Table 5.1 : A summary of the relationship between the separation conditions of the data and the catapult phase for different losses

separation condition	linear separable	degenerate	non-separable
exponential loss (this work)	✗	✓	✓
logistic loss (this work)	✗	✓	✓
squared loss ([81])	✓	✓	✓

5.4 Experiments

In this section, we present the numerical results conducted with deep linear NN, equipped with logistic loss, and CIFAR10 to examine whether reaching a flatter minimuma in the catapult phase leads to better generalization with real applications. We selected two of the ten categories in the CIFAR-10 dataset, "cars" and "dogs", to form a binary classification problem and compared the generalization performance of models at end of training with different learning rates and with two different stopping scenarios.

In Figure 5.4 we show the final performance of two linear networks, one with one hidden layer and no bias; and the other with two hidden layers and bias. Both networks were trained on CIFAR-10. Figure 5.4(a,c) show the results with a fixed training time. Figure 5.4(b,d) show the results with a fixed physical training time,

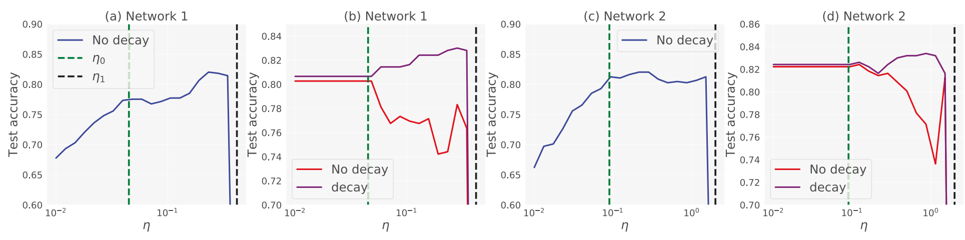

Figure 5.4 : Test performance of deep linear networks with respect to different learning rate phases. The data size is of $n_{\text{train}} = 2048$ and $n_{\text{test}} = 512$. (a,b) A two-layer linear network without bias of $\sigma_w^2 = 0.5$ and $m = 500$. (c,d) A three-layer linear network with a bias of $\sigma_w^2 = 0.5$, $\sigma_b^2 = 0.01$, and $m = 500$. (a,c) The test accuracy is measured at the time step $t = 500$ and $t = 300$ respectively. (b,d) The test accuracy is measured at the physical time step (red curve), after which it continues to evolve for a period of time at a small learning rate (purple): $t_{\text{phy}} = 50/\eta$ and extra time $t = 500$ at $\eta = 0.01$ for the decay case. Although the results in the catapult phase do not perform as well as the lazy phase when there is no decay, the best performance can be found in the catapult phase when adopting learning rate annealing.

defined as $t_{\text{phy}} = t_0\eta$, where t_0 is a constant. As shown in 5.4, fixing the training time leads to higher test accuracy in the catapult phase. However, it also results in a bias in favor of large learning rates, since large learning rates naturally run faster. In the second scenario (Figures 5.4(b,d)), the models trained at a large learning rate did not perform nearly as well as those trained at a steady small learning rate. Nevertheless, we can still achieve highest accuracy in the catapult phase where the learning rate exceeds a critical threshold when adopting the learning rate annealing strategy.

According to theorem 2 from [58], data can be divided into two partitions: one linearly separable; the other non-separable. When tuning a learning rate to large, the algorithm quickly iterates to a flat minimum in the space spanned by non-separable data. But this large learning rate prevents gradient descent from achieving the maximum margin with the linearly separable data. As a result, generalization ability deteriorates for some of the data. This explains why performance degrades to below a small learning rate with a fixed number of physical steps. On the other hand, an annealing strategy with a large learning rate followed by a small learning rate, produces good results. The large learning rate has learned has learned a flat curvature, the subsequent small learning rate will not affect this result for non-separable data. Further, reducing the learning rate can restore the maximum margin of the linearly separable data, ultimately delivering much better performance overall.

5.5 Discussion

Inspired by the seminal work [81], this study characterizes the dynamics of deep linear networks on binary classification problem at large learning rates. We find that producing the catapult effect in the large learning rate phase depends on the separation conditions associated with logistic and exponential loss. According to our theoretical analysis, the loss in the catapult phase can converge to a flatter minimum than that in the lazy phase, from the perspective of the Hessian. We show empirically that, even without SGD optimization, the best generalization performance for linear networks can be achieved in the catapult phase. This work advances our theoretical understanding of the training behaviors resulting from large learning rates with linear networks for binary classification. However, many open questions remain in this nascent field of study. For example, there is no solid theory on the effects of a large learning rate for non-linear networks. The same can be said of stochastic gradient descent with large learning rates. We leave these unsolved problems for future work.

5.6 Proof

This appendix is dedicated to proving the key results of this paper, namely Proposition 5.1 and Theorems 5.1, 5.2, and Corollary 5.1 which describe the dynamics of gradient descent with logistic and exponential loss in different learning rate phases.

Proposition 5.1. *For a linear predictor $f = w^T x$, along with a loss $\ell \in \{\ell_{\exp}, \ell_{\log}\}$.*

1 *Under Assumption 5.1, the empirical loss is β-smooth. Then the gradient descent with constant learning rate $\eta < \frac{2}{\beta}$ never increase the risk, and empirical loss will converge to zero:*

$$\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \leq 0, \quad \lim_{t\to\infty} \mathcal{L}(w_t) = 0, \quad \textit{with} \quad \eta < \frac{2}{\beta}$$

2 *Under Assumption 5.2, the empirical loss is β-smooth and α-strongly convex, where $\alpha \leq \beta$. Then the gradient descent with a constant learning rate $\eta < \frac{2}{\beta}$ never increases the risk, and empirical loss will converge to a global minimum. On the other hand, the gradient descent with a constant learning rate $\eta > \frac{2}{\alpha}$ never decrease the risk, and empirical loss will explode or saturate:*

$$\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \leq 0, \quad \lim_{t\to\infty} \mathcal{L}(w_t) = G_0, \quad \textit{with} \quad \eta < \frac{2}{\beta}$$
$$\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \geq 0, \quad \lim_{t\to\infty} \mathcal{L}(w_t) = G_1, \quad \textit{with} \quad \eta > \frac{2}{\alpha}$$

where G_0 is the value of a global minimum while $G_1 = \infty$ for exploding situation or $G_0 < G_1 < \infty$ when saturating.

Proof. 1 We first prove that empirical loss $\mathcal{L}(u)$ regrading data-scaled weight $u_i \equiv w^T x_i y_i$ for the linearly separable dataset is smooth. The empirical loss can be written as $\mathcal{L} = \sum_{i=1}^n \ell(u_i)$, then the second derivatives of logistic and exponential loss are,

$$\mathcal{L}''_{\exp} = \sum_{i=1}^n \ell''_{\exp}(u_i) = \sum_{i=1}^n \exp''(-u_i) = \sum_{i=1}^n \exp(-u_i)$$

$$\mathcal{L}''_{\log} = \sum_{i=1}^{n} \ell''_{\log}(u_i) = \sum_{i=1}^{n} \log''(1+\exp(-u_i)) = \sum_{i=1}^{n} \frac{\exp(-u_i)}{(1+\exp(-u_i))^2}$$

when w_t is limited, there will be a β such that $\mathcal{L}'' < \beta$. Besides, because there exists a separator w_* such that $\forall i : w_*^T x_i y_i > 0$, the second derivative of empirical loss can be arbitrarily close to zero. This implies that the empirical loss function is not strongly convex.

Recall a property of β-smooth [94],

$$f(y) \leq f(x) + (\nabla_x f)^T (y-x) + \frac{1}{2}\beta \|y-x\|^2$$

Taking the gradient descent into consideration,

$$\begin{aligned}
\mathcal{L}(w_{t+1}) &\leq \mathcal{L}(w_t) + \big(\nabla_{w_t}\mathcal{L}(w_t)\big)^T \big(w_{t+1} - w_t\big) + \frac{1}{2}\beta \|w_{t+1} - w_t\|^2 \\
&= \mathcal{L}(w_t) + \big(\nabla_{w_t}\mathcal{L}(w_t)\big)^T \big(-\eta \nabla_{w_t}\mathcal{L}(w_t)\big) + \frac{1}{2}\beta \|-\eta \nabla_{w_t}\mathcal{L}\|^2 \\
&= \mathcal{L}(w_t) - \eta(1 - \frac{\eta\beta}{2}) \|\nabla_{w_t}\mathcal{L}\|^2
\end{aligned}$$

when $1 - \frac{\eta\beta}{2} > 0$, that is $\eta < \frac{2}{\beta}$,

$$\mathcal{L}(w_{t+1}) \leq \mathcal{L}(w_t) - \eta(1 - \frac{\eta\beta}{2}) \|\nabla_{w_t}\mathcal{L}\|^2 \leq \mathcal{L}(w_t)$$

We now prove that output function of the network will converge to a zero with learning rate $\eta < \frac{2}{\beta}$. We change the form of inequality above,

$$\frac{\mathcal{L}(w_t) - \mathcal{L}(w_{t+1})}{\eta(1 - \frac{\eta\beta}{2})} \geq \|\nabla_{w_t}\mathcal{L}(w_t)\|^2$$

this implies,

$$\begin{aligned}
\sum_{t=0}^{T} \|\nabla_{w_t}\mathcal{L}(w_t)\|^2 &\leq \sum_{t=0}^{T} \frac{\mathcal{L}(w_t) - \mathcal{L}(w_{t+1})}{\eta(1 - \frac{\eta\beta}{2})} \\
&= \frac{\mathcal{L}(w_0) - \mathcal{L}(w_T)}{\eta(1 - \frac{\eta\beta}{2})} < \infty
\end{aligned}$$

therefore we have $\lim_{t\to\infty} \|\nabla_{w_t}\mathcal{L}(w_t)\| = 0$.

2 When the data is not linear separable, there is no w_* such that $\forall i : w_*^T x_i y_i > 0$. Thus, at least one $w_*^T x_i y_i$ is negative when the other terms are positive. This

implies that the solution of the loss function is finite and the empirical loss is both α-strongly convex and β-smooth.

Recall a property of α-strongly convex function f [94],

$$f(y) \geq f(x) + (\nabla_x f)^T (y-x) + \frac{1}{2}\alpha \left\| y-x \right\|^2$$

Taking the gradient descent into consideration,

$$\begin{aligned}\mathcal{L}(w_{t+1}) &\geq \mathcal{L}(w_t) + \big(\nabla_{w_t}\mathcal{L}(w_t)\big)^T \big(w_{t+1}-w_t\big) + \frac{1}{2}\alpha \left\| w_{t+1}-w_t \right\|^2 \\ &= \mathcal{L}(w_t) + \big(\nabla_{w_t}\mathcal{L}(w_t)\big)^T \big(-\eta\nabla_{w_t}\mathcal{L}(w_t)\big) + \frac{1}{2}\alpha \left\| -\eta\nabla_{w_t}\mathcal{L} \right\|^2 \\ &= \mathcal{L}(w_t) - \eta(1-\frac{\eta\alpha}{2}) \left\| \nabla_{w_t}\mathcal{L} \right\|^2\end{aligned}$$

when $1-\frac{\eta\alpha}{2} < 0$, that is $\eta > \frac{2}{\alpha}$, we have,

$$\mathcal{L}(w_{t+1}) \geq \mathcal{L}(w_t) - \eta(1-\frac{\eta\alpha}{2}) \left\| \nabla_{w_t}\mathcal{L} \right\|^2 \geq \mathcal{L}(w_t).$$

$\square$

Theorem 5.1. *For a linear predictor $f = w^T x$ equipped with exponential (logistic) loss under assumption 5.3, there is a critical learning rate that separates the whole learning rate space into two (three) regions. The critical learning rate satisfies*

$$\mathcal{L}'(w_0) = -\mathcal{L}'(w_0 - \eta_{\text{critical}}\mathcal{L}'(w_0)),$$

where w_0 is the initial weight. Moreover,

1 *For exponential loss, the gradient descent with a learning rate $\eta < \eta_{critical}$ never increases loss, and the empirical loss will find the global minimum. On the other hand, the gradient descent with learning rate $\eta = \eta_{\text{critical}}$ will oscillate. Finally, when the learning rate $\eta > \eta_{\text{critical}}$, the training process never decreases the loss and the empirical loss will explode to infinity:*

$$\begin{aligned}&\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) < 0, \quad \lim_{t\to\infty}\mathcal{L}(w_t) = 1, \quad \textit{with} \quad \eta < \eta_{\text{critical}}, \\ &\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) = 0, \quad \lim_{t\to\infty}\mathcal{L}(w_t) = \mathcal{L}(w_0), \quad \textit{with} \quad \eta = \eta_{\text{critical}}, \\ &\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) > 0, \quad \lim_{t\to\infty}\mathcal{L}(w_t) = \infty, \quad \textit{with} \quad \eta > \eta_{\text{critical}}.\end{aligned}$$

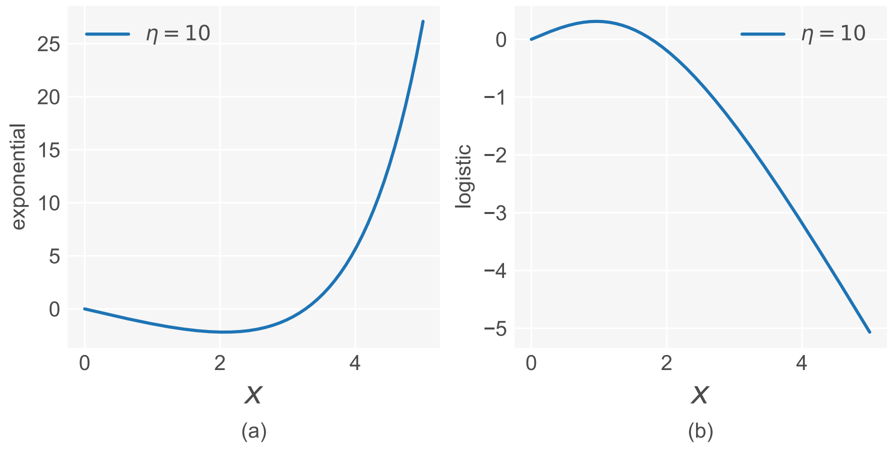

Figure 5.5 : Graph of $\phi(x)$ for the two losses. (a) Exponential loss with learning rate $\eta = 10$. (b) Logistic loss with learning rate $\eta = 10$.

2 For logistic loss, the critical learning rate satisfies a condition: $\eta_{\text{critical}} > 8$. *The gradient descent with a constant learning rate* $\eta < 8$ *never increases the loss, and the loss will converge to the global minimum. On the other hand, the loss along with a learning rate* $8 \leq \eta < \eta_{\text{critical}}$ *will not converge to the global minimum but oscillate. Finally, when the learning rate* $\eta > \eta_{\text{critical}}$*, gradient descent never decreases the loss, and the loss will saturate:*

$$\begin{aligned}
&\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) < 0, \quad \lim_{t\to\infty} \mathcal{L}(w_t) = \log(2), \ \textit{with } \eta < 8,\\
&\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \leq 0, \quad \lim_{t\to\infty} \mathcal{L}(w_t) = \mathcal{L}(w_*) < \mathcal{L}(w_0), \ \textit{with } 8 \leq \eta < \eta_{\text{critical}},\\
&\mathcal{L}(w_{t+1}) - \mathcal{L}(w_t) \geq 0, \quad \lim_{t\to\infty} \mathcal{L}(w_t) = \mathcal{L}(w_*) \geq \mathcal{L}(w_0), \ \textit{with } \eta \geq \eta_{\text{critical}}.
\end{aligned}$$

where w_* *satisfies* $-w_* = w_* - \frac{\eta}{2}\frac{\sinh(w_*)}{1+\cosh(w_*)}$.

Proof. 1 Under the *degeneracy assumption*, the risk is given by the hyperbolic function $\mathcal{L}(w_t) = \cosh(w_t)$. The update function for the single weight is,

$$w_{t+1} = w_t - \eta \sinh(w_t).$$

To compare the norm of the gradient $\|\sinh(w_t)\|$ and the norm of loss, we introduce the following function:

$$\phi(x) = \eta\mathcal{L}'(x) - 2x \tag{5.9}$$

$$= \eta \sinh(x) - 2x, \quad \text{for } x \geq 0. \tag{5.10}$$

Then it's easy to see that

$$\mathcal{L}(w_{t+1}) > |\mathcal{L}(w_t)| \iff \phi(|w_t|) > 0.$$

In this way, we have transformed the problem into studying the iso-surface of $\phi(x)$. Define Phase_1 by

$$\text{Phase}_1 = \{x|\phi(x) < 0\}.$$

Let Phase_2 be the complementary set of Phase_1 in $[0, +\infty)$. Since $\frac{\sin x}{x}$ is monotonically increasing, we know that Phase_2 is connected and contains $+\infty$. Suppose $\eta > \eta_{\text{critical}}$, then $\phi(w_0) > 0$, which implies that

$$\mathcal{L}(w_1) > \mathcal{L}(w_0) \text{ and } |w_1| > |w_0|.$$

Thus, the first step gets trapped in Phase_2:

$$\phi(w_1) > 0.$$

By induction, we can prove that $\phi(w_t) > 0$ for arbitrary $t \in \mathbb{N}$, which is equivalent to

$$\mathcal{L}(w_t) > \mathcal{L}(w_{t-1}).$$

Similarly, we can prove the theorem under another toe initial conditions: $\eta = \eta_{\text{critical}}$ and $\eta < \eta_{\text{critical}}$.

2 Under the *degeneracy assumption*, the risk is governed by the hyperbolic function $\mathcal{L}(w_t) = \frac{1}{2}\log(2 + 2\cosh(w_t))$. The update function for the single weight is,

$$w_{t+1} = w_t - \frac{\eta}{2}\frac{\sinh(w_t)}{1 + \cosh(w_t)}.$$

Thus,

$$\phi(x) = \eta\mathcal{L}'(x) - 2x \tag{5.11}$$
$$= \frac{\eta}{2}\frac{\sinh(x)}{1+\cosh(x)} - 2x, \quad \text{for } x \geq 0. \tag{5.12}$$

Unlike the exponential loss, $\frac{\sinh(x)}{x(1+\cosh(x))}$ is monotonically decreasing, which means that Phase$_2$ of $\phi(x)$ doesn't contain $+\infty$.(see figure 5.5).

Suppose $8 < \eta < \eta_{\text{critical}}$, then w_0 lies in Phase$_2$. In this situation, we denote the critical point that separates Phase$_1$ and Phase$_2$ by w_*. That is

$$-w_* = w_* - \eta\frac{\sinh(w_*)}{1+\cosh(w_*)}.$$

Then it's obvious that before w_t arrives at w_*, it keeps decreasing and will eventually get trapped at w_*:

$$\lim_{t\to\infty} w_t = w_*,$$

and we have

$$\lim_{t\to\infty} \mathcal{L}(w_t) - \mathcal{L}(w_{t-1}) = 0.$$

When $\eta < 8$, Phase$_2$ is empty. In this case, we can prove by induction that $\phi(w_t) > 0$ for arbitrary $t \in \mathbb{N}$, which is equivalent to

$$\mathcal{L}(w_t) > \mathcal{L}(w_{t-1}).$$

□

Theorem 5.2. *Under appropriate initialization and assumption 5.3, there exists a catapult phase for both the exponential and logistic loss. More precisely, when η belongs to this phase, there exists a $T > 0$ such that the output function f_t and the eigenvalue of NTK λ_t update in the following way:*

i. $\mathcal{L}_t$ keeps increasing when $t < T$.

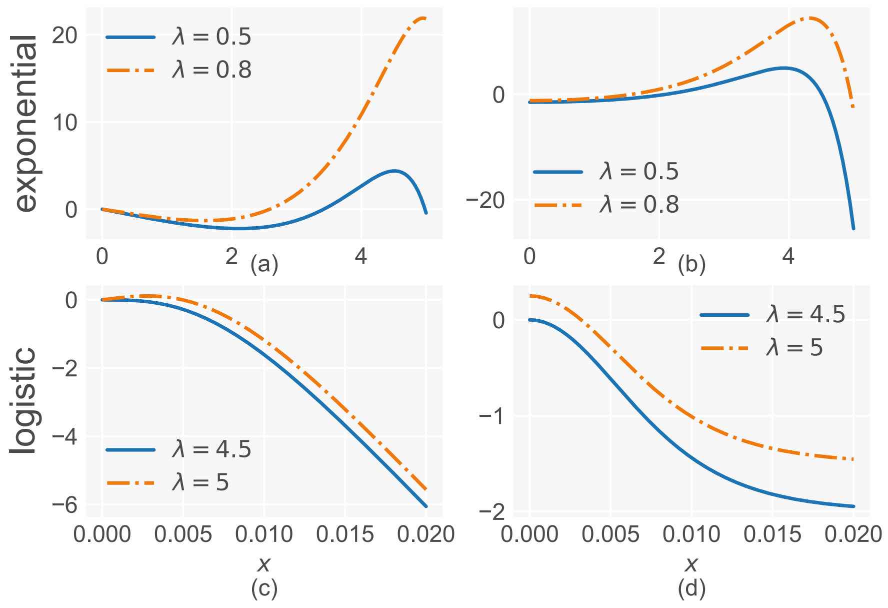

Figure 5.6 : Different colors represent different λ(NTK) values. (a) graph of $\phi_\lambda(x)$ equipped with the exponential loss. (b) graph of the derivative of $\phi_\lambda(x)$ equipped with the exponential loss. (c) graph of $\phi_\lambda(x)$ equipped with the logistic loss. (d) graph of the derivative of $\phi_\lambda(x)$ equipped with the logistic loss. Notice that the critical point of the exponential loss moves to the right as λ decreases.

ii. After the T step and its successors, the loss decreases, which is equivalent to:

$$|f_{T+1}| > |f_{T+2}| \geq |f_{T+3}| \geq \dots .$$

iii. The eigenvalue of NTK keeps dropping after the T steps:

$$\lambda_{T+1} > \lambda_{T+2} \geq \lambda_{T+3} \geq \dots .$$

Moreover, we have the inverse relation between the learning rate and the final eigenvalue of NTK: $\lambda_\infty \leq \lim_{t\to\infty} \frac{4f_t}{\eta \tilde{f}_{t\text{exp}}}$ with exponential loss, or $\lambda_\infty \leq \lim_{t\to\infty} \frac{4f_t}{\eta \tilde{f}_{t\text{log}}}$ with logistic loss.

Proof. **Exponential loss**

$\tilde{f}_{\text{exp}}$ satisfies:

i. $|\tilde{f}_{\exp}(x)| = |\tilde{f}_{\exp}(-x)|$.

ii. $\lim_{x\to 0} \frac{\tilde{f}_{\exp}(x)}{x} = 1$.

iii. $\tilde{f}_{\exp}(x)$ has exponential growth as $x \to \infty$.

By the definition of the normalized NTK, we automatically get

$$\lambda_t \geq 0.$$

From the numerical experiment, we observe that at the ending phase of training, λ_t keeps non-increasing. Thus, λ_t must converge to a non-negative value, which satisfies

$$\frac{\eta^2}{m}\lambda \tilde{f}_t^2 - \frac{4\eta}{m} f_t \tilde{f}_t \leq 0. \tag{5.13}$$

Thus, $\lambda \leq \lim_{t\to\infty} \frac{4f_t}{\eta \tilde{f}_t}$.
Since the output f converges to the global minimum, a larger learning rate will lead to a lower limiting value of the NTK. As it was pointed out in [81], a flatter NTK corresponds to a smaller generalization error in the experiment. However, we still need to verify that large learning rate exists.
Note that during training, the loss function curve may experience more than one wave of uphill and downhill. To give a precise definition of large learning rate, it should satisfy the following two conditions:

i. $|f_{T+1}| > |f_T|$, this implies that

$$\mathcal{L}_{T+1} > \mathcal{L}_T.$$

For the $T+1$ step and its successors,

$$|f_{T+1}| > |f_{T+2}| \geq |f_{T+3}| \geq \dots.$$

ii. The norm of NTK keeps dropping after T steps:

$$\lambda_T > \lambda_{T+1} \geq \lambda_{T+2} \geq \dots.$$

If we already know that the loss keeps decreasing after the $T+1$ step, then

$$\Delta\lambda = \frac{\eta}{m}\tilde{f} \cdot (\eta\lambda\tilde{f} - 4f). \tag{5.14}$$

Since $\frac{|\tilde{f}|}{|f|} \geq 1$ and is monotonically increasing when $\tilde{f} = \sinh f$, we automatically have

$$\lambda_T > \lambda_{T+1} > \lambda_{T+2} \geq \dots,$$

If

$$\lambda_T < \frac{4f_T}{\eta\tilde{f}_T} \quad \textit{and} \quad \lambda_{T+1} < \frac{4f_{T+1}}{\eta\tilde{f}_{T+1}}.$$

This condition holds if the parameters are close to zero initially.

To check condition (1), the following function which plays an essential role as in the non-hidden layer case:

$$\phi_\lambda(x) = \eta\lambda\sinh(x) - \frac{\eta^2}{m}x\sinh^2(x) - 2x, \quad \text{for } x \geq 0.$$

Notice that an extra parameter λ emerges with the appearance of the hidden layer. We call it the control parameter of the function $\phi(x)$.

For a fixed λ, since now $\phi(x)$ becomes linear, the whole $[0, +\infty)$ is divided into three phases (see figure 5.6):

Phase$_1$:= the connected component of $\{x|\ \phi_\lambda(x) < 0\}$ that contains 0,

Phase$_2$:= $\{x|\ \phi_\lambda(x) > 0\}$,

Phase$_3$:= the connected component of $\{x|\ \phi_\lambda(x) < 0\}$ that contains $+\infty$.

It's easy to see that $\mathcal{L}_{\exp}(f_{T+1}) > \mathcal{L}_{\exp}(f_T)$ if and only if

$$\phi_{\lambda_T}(f_T) > 0.$$

That is, f_T lies in Phase$_2$ of ϕ_{λ_T}. Similarly,

$$\mathcal{L}_{\exp}(f_{T+2}) < \mathcal{L}_{\exp}(f_{T+1}) \iff \phi_{\lambda_{T+1}}(f_{T+1}) < 0.$$

That is, f_{T+1} jumps into Phase$_1$ of $\phi_{\lambda_{T+1}}$. Denote the point that separates Phase$_1$ and Phase$_2$ by x_*, then form the graph of $\phi_\lambda(x)$ with different λ, we know that

$$x_*(\lambda') > x_*(\lambda) \text{ if } \lambda' < \lambda.$$

Therefore, condition (1) is satisfied if

$$x_*(\lambda_{T+1}) > f_T + \phi_{\lambda_T}(f_T) \tag{5.15}$$

and at the same time,

$$\lambda_{T+1} - \lambda_T > 0.$$

For simplicity, we reset T as our initial step. Write the output function f_t as

$$f_{t+1} = f_t(1 + \mathcal{A}_t), \tag{5.16}$$

where $\mathcal{A}_t = \frac{\eta^2}{m}\tilde{f}_t^2 - \eta\lambda_t\tilde{f}_t/f_t$. Thus, $\phi_{\lambda_0}(f_0) > 0$ is equivalent to $\mathcal{A}_0 < -2$. Similarly, write the update function for λ_t as

$$\lambda_{t+1} = \lambda_t(1 + \mathcal{B}_t), \tag{5.17}$$

where $\mathcal{B}_t = \frac{\eta^2}{m}\tilde{f}_t^2 - \frac{4\eta}{m}\tilde{f}_t f_t/\lambda_t$. To fulfill the above condition on NTK, we need

$$\mathcal{B}_0 < 0.$$

To check (5.15), let the initial output f_0 be close to $X_*(\lambda_0)$ (this can be done by adjusting w_0):

$$0 < f_0 - X_* < \epsilon.$$

Then by the mean value theorem,

$$x_*(\lambda_1) - x_*(\lambda_0) = \frac{\partial x_*}{\partial \lambda_*}(\lambda_*) \cdot \Delta\lambda.$$

The derivative $\frac{\partial x_*}{\partial \lambda_*}$ can be calculated by the implicit function theorem:

$$\begin{aligned}\frac{\partial x_*}{\partial \lambda} &= -\frac{\partial \phi_\lambda(x_*)}{\partial \lambda} / \frac{\partial \phi_\lambda(x_*)}{\partial x_*} \\ &= -\eta \sinh(x_*) / \frac{\partial \phi_\lambda(x_*)}{\partial x_*}.\end{aligned}$$

It's easy to see that $|\frac{\partial x_*}{\partial \lambda}|$ is bounded away from zero if the initial output is in Phase$_2$ and near x_* of $\phi_{\lambda_0}(x)$ (see Figure 5.6).

On the other hand, we have the freedom to move f_0 towards x_* of $\phi_{\lambda_0}(x)$ without breaking the $\Delta\lambda < 0$ condition. Since

$$|\frac{\eta\lambda\tilde{f}}{4f}| < |\frac{\eta\lambda\tilde{f}'}{4f'}| \quad \text{if} \;\; f < f'.$$

Therefore, we can always find a $\epsilon > 0$ such that $0 < f_0 - x_* < \epsilon$ and (5.15) is satisfied. Combining the above, we have demonstrated the existence of the *catapult phase* for the exponential loss.

logistic loss

When Considering the degeneracy case for the logistic loss, the loss will be

$$\mathcal{L} = \frac{1}{2}\log(2 + 2\cosh(m^{-1/2}w^{(2)}w^{(1)})). \tag{5.18}$$

Much of the argument is similar. For example, Equation (5.13) still holds if we replace $\tilde{f}_{\exp}$ by

$$\tilde{f}_{\log}(x) := \frac{\sinh(x)}{1 + \cosh(x)}.$$

$\tilde{f}_{\log}$ satisfies

i. $|\tilde{f}_{\log}(x)| = |\tilde{f}_{\log}(-x)|$.

ii. $|\tilde{f}_{\log}(x)| \leq 1 \quad \text{for } x \in (-\infty, \infty)$.

This implies that

$$|\frac{\tilde{f}}{f}| \leq \frac{1}{2}.$$

Then by (5.14), we have $\Delta\lambda < 0$ if $\lambda \leq \frac{8}{\eta}$. Thus, condition 2 is satisfied for both loss functions. Now, $\phi_\lambda(x)$ becomes:

$$\phi_\lambda(x) := \eta\lambda\frac{\sinh(x)}{1 + \cosh(x)} - \frac{\eta^2}{m}x\frac{\sinh^2(x)}{(1 + \cosh(x))^2} - 2x,$$

along with its derivative:

$$\begin{aligned}\phi_\lambda'(x) :=& \eta\lambda\frac{\cosh(x)}{1+\cosh(x)} - \frac{\eta\lambda\sinh^2(x)}{(1+\cosh(x))^2} - 2\\ &-\frac{\eta^2}{m}\frac{\sinh^2(x)}{(1+\cosh(x))^2} - 2\frac{\eta^2}{m}\frac{\sinh(x)}{1+\cosh(x)}\left[\frac{\cosh(x)}{1+\cosh(x)} - \frac{\sinh^2(x)}{(1+\cosh(x))^2}\right].\end{aligned}$$

The method of verifying condition 1 is similar with the exponential case, except that $\phi_\lambda(x)$ has only Phase$_1$ and Phase$_2$ (see figure 5.6). As the NTK λ goes down, Phase$_1$ will disappear and at that moment the loss will keep decreasing. Let λ_* be the value such that $\phi'_{\lambda_*}(x) = 0$, then

$$\lambda_* = 4/\eta.$$

During the period when $4/\eta < \lambda_t < 8/\eta$, the NTK keeps dropping and the loss may oscillate around x_*. However, we may encounter the scenario that both the loss and the λ_t are going up before dropping down simultaneously (see the first three steps in figure 5.6). Theoretically, it corresponds to jump from Phase$_2$ to Phase$_3$ and then to Phase$_1$ of $\phi_{\lambda_1}(x)$ in the first two steps. This is possible since $\tilde{f}_{\log}$ is decreasing when $x > 0$. This implies that

$$\left|\frac{\eta\lambda\tilde{f}'}{4f'}\right| < \left|\frac{\eta\lambda\tilde{f}}{4f}\right| \quad \text{if} \;\; f < f'.$$

So an increase of the output will cause the NTK to drop faster. □

Corollary 5.1 (Informal)**.** *Consider optimizing $\mathcal{L}(w)$ with one hidden layer linear network under assumption 5.3 and exponential (logistic) loss using gradient descent with a constant learning rate. For any learning rate that loss can converge to the global minimal, the larger the learning rate, the flatter curvature the gradient descent will achieve at the end of training.*

Proof. The Hessian matrix is defined as the second derivatives of the loss with respect to the parameters,

$$H_{\alpha\beta} = \frac{\partial^2\mathcal{L}}{\partial\theta_\alpha\partial\theta_\beta}$$

where $\theta_\alpha, \theta_\beta \in \{w^{(1)}, w^{(2)}\}$ for our setting of linear networks. For logistic loss,

$$\begin{aligned} H_{\alpha\beta} &= \frac{1}{n}\sum_i \frac{\partial^2 \exp(-y_i f_i)}{\partial\theta_\alpha \partial\theta_\beta} \\ &= \frac{1}{n}\sum_i \left[\frac{\partial^2 f_i}{\partial\theta_\alpha \partial\theta_\beta} \exp(-y_i f_i)(-y_i) + \frac{\partial f_i}{\partial\theta_\alpha}\frac{\partial f_i}{\partial\theta_\beta} \exp(-y_i f_i)\right] \end{aligned}$$

We want to make a connection from Hessian matrix to the neural tangent kernel. Note that the second term contains $\frac{\partial f_i}{\partial\theta_\alpha}\frac{\partial f_i}{\partial\theta_\beta}$, which can be written as JJ^T, where $J = \text{vec}[\frac{\partial f_i}{\partial\theta_j}]$. While NTK can be expressed as $J^T J$. It is known that they have the same eigenvalue. Further more, under assumption 5.3, we have $n = 2$ and $f_1 = f_2$, thus,

$$H_{\alpha\beta} = \frac{1}{n}\sum_i \left[\frac{\partial^2 f_i}{\partial\theta_\alpha \partial\theta_\beta}\frac{\partial \mathcal{L}}{\partial f_\theta} + \frac{\partial f_i}{\partial\theta_\alpha}\frac{\partial f_i}{\partial\theta_\beta}\mathcal{L}\right]$$

Suppose at the end of gradient descent training, we can achieve a global minimum. Then we have, $\frac{\partial \mathcal{L}}{\partial f_\theta} = 0$, and $\mathcal{L} = 1$. Thus, the Hessian matrix reduce to,

$$H_{\alpha\beta} = \frac{1}{n}\sum_i \frac{\partial f_i}{\partial\theta_\alpha}\frac{\partial f_i}{\partial\theta_\beta}$$

In this case, the eigenvalues of Hessian matrix are equal to those of neural tangent kernel. Combine with the 5.2, we can prove the result.

For logistic loss,

$$\begin{aligned} H_{\alpha\beta} &= \frac{1}{n}\sum_i \frac{\partial^2 \log(1+\exp(-y_i f_i))}{\partial\theta_\alpha \partial\theta_\beta} \\ &= \frac{1}{n}\sum_i \left[\frac{\partial^2 f_i}{\partial\theta_\alpha \partial\theta_\beta}\frac{\exp(-y_i f_i)(-y_i)}{1+\exp(-y_i f_i)} + \frac{\partial f_i}{\partial\theta_\alpha}\frac{\partial f_i}{\partial\theta_\beta}\frac{\exp(-y_i f_i)}{(1+\exp(-y_i f_i))^2}\right] \end{aligned}$$

Under assumption 5.3, we have $n = 2$ and $f_1 = f_2$, thus,

$$H_{\alpha\beta} = \frac{1}{n}\sum_i \big[\frac{\partial^2 f_i}{\partial\theta_\alpha \partial\theta_\beta}\frac{\partial\mathcal{L}}{\partial f_\theta} + \frac{\partial f_i}{\partial\theta_\alpha}\frac{\partial f_i}{\partial\theta_\beta}\frac{\exp(-y_i f_i)}{(1+\exp(-y_i f_i))^2}\big]$$

Suppose at the end of gradient descent training, we can achieve a global minimum. Then we have, $\frac{\partial\mathcal{L}}{\partial f_\theta} = 0$, and $f_i = 0$. Thus, the Hessian matrix reduce to,

$$H_{\alpha\beta} = \frac{1}{4n}\sum_i \frac{\partial f_i}{\partial\theta_\alpha}\frac{\partial f_i}{\partial\theta_\beta}$$

In this case, the eigenvalues of Hessian matrix and NTK have the relation $\frac{1}{4}\lambda_{\text{NTK}} = \lambda_{\text{Hessian}}$.

$\square$

Chapter 6

Wide graph neural networks: aggregation provably leads to exponentially trainability loss

6.1 Introduction

Recently, Graph Convolutional Networks (GCN) have shown incredible abilities to learn node or graph representations and achieved superior performances for various downstream tasks, such as node classifications [9, 10, 11, 12], graph classifications [13, 14, 15, 16, 17, 18], link predictions [19], etc. However, most GCNs achieve their best at a shallow depth, e.g., 2 or 3 layers, and their performance on those tasks promptly degrade as the number of layers grow. Towards this phenomenon, some attempts have been made expecting to deepen understanding of current GNN architectures and their expressive power. Li et al. [95] showed that GCN mixes node representations with nearby neighbors. This mechanism potentially posed the risk of *over-smoothing* as more layers were stacked together, where node representations in an extensive range tended to be indistinguishable from each other. Alon et al. [96] analyzed the propagation mechanism of GCN and attribute performance deterioration to the over-squashing problem. As each node iteratively aggregates messages from its neighbors, exponentially-growing information is embedded into a fixed-length vector, which causes information loss and thus undermines the expressive power of generated representations. Oono et al. [97] investigated the expressive power of GNNs using the asymptotic behaviors as the layer goes to infinity. They proved that under certain conditions, the expressive power of GCN is determined by the topological information of the underlying graphs inherent in the graph spectra.

More recently, a series of latest works [98, 99, 100, 101, 102] explored the underlying reasons for performance degrading and corresponding solutions, but rare of them theoretically investigated the expressivity and trainability of deep GNNs.

To fill this gap, we resort to the infinitely-wide multi-layer GCN. The research on infinitely-wide networks was originated from a work by [103]. Neal showed the one hidden layer networks with random weights at initialization (without training) are Gaussian Processes (GPs). Later, the connection between GPs and multi-layer infinitely-wide networks with Gaussian initialization [66, 78] and orthogonal weights [46] was reported. Meanwhile, from a “mean-field theory” perspective [22, 23], the GP Kernel (GPK) of a deep network can be used to measure the expressivity. As a result of studying the GPK under mean-field theory, an order-to-chaos expressivity phase transition split by a critical line was found [22, 23]. It is also hypothesized that wide networks can be trained successfully when the expressivity is maintained as the network goes deeper. This ansatz was tested and verified by practical experiments with a wide range of architectures, including convolutions [26], recurrent networks [27], and residual networks [25], and more.

Despite the significant development from the mean-field theory, it is beyond the scope of this study to examine the optimization property of infinitely-wide networks. Recent trends in Neural Tangent Kernel (NTK) have led to a proliferation of studies on the optimization and generalization of infinitely-wide and ultra-wide networks. In particular, Jacot et al. [36] made a groundbreaking discovery that the evolution of neural networks by gradient descent training with a sufficient small learning rate or gradient flow in the infinite width limit can be captured by an NTK. Meanwhile, a considerable literature has grown up around the theme of NTK and gained a great deal of successes in deep learning theory for ultra-wide (over-parameterized) networks [37, 38, 39, 40, 41, 42, 43]. Recently, [47] formulated Graph Neural Tangent Kernel for infinitely-wide GNNs and shed light on theoretical guarantees for GNN.

Prior to the discovery of GNTK, people had been puzzled by the non-convexity of graph neural networks, which is analytically intractable. In the learning regime of GCN governed by GNTK, the optimization becomes an almost convex problem, making GNTK a promising method to study deep GCN's trainability.

In this work, we leverage these techniques on infinitely-wide networks and investigate whether ultra-wide GCNs are trainable in the large depth. In particular, we formulate the large-depth asymptotics behavior of GPK inspired by recent advances on network's expressivity through mean-field theory [22, 23]. Furthermore, we study the trainability of ultra-wide GCN by computing the GNTK in the large-depth limit illuminated by innovative works on the deep networks [104, 105]. Besides, we extend our theoretical framework to the residual connection-like techniques and show these techniques can mildly slow down the exponential decay instead of overcoming it thoroughly. Our contributions are as follows:

- We compute the large-depth limiting behavior of GPK to measure expressivity of deep GCNs. We prove that the infinitely-wide GCNs lose expressivity exponentially due to aggregation.
- As far as we know, we are the first to investigate the trainability of deep and wide GCN through GNTK. We prove that all elements of a GNTK matrix regarding a pair of graphs converge exponentially to the same value, making the GNTK matrix singular in the large depth. Based on this result, we make a corollary that ultra-wide GCNs' trainability exponentially collapses on node classification tasks.
- We apply our theoretical analysis to the residual connection-resemble techniques for GCNs and show that residual connection can, to some extent, slow down the exponential decay rate. However, our theory shows that residual connection can not fundamentally solve the exponential decay problem. This

result can be used to explain to what extent the recent residual connection-resemble method works.

- The experiments are conducted to validate our theoretical analysis quantitatively. Together with the above findings, we provide comprehensive guidance for the development of deep GCNs.

6.2 Background

We review the mean-field theory of Gaussian Process Kernel and Neural Tangent Kernel for infinitely-wide networks. Furthermore, we introduce Graph Convolutional Networks along with our setup and notation.

6.2.1 Mean Field Theory and Expressivity

We begin by considering a fully-connected network of depth L of widths m_l in each layer. The weight and bias in the l-th layer are denoted by $W^{(l)} \in \mathbb{R}^{m_l \times m_{l-1}}$ and $b^{(l)} \in \mathbb{R}^{m_l}$. Letting the pre-activations be given by $h_i^{(l)}$. Then the information propagation in this network is governed by,

$$h_i^{(l)} = \frac{\sigma_w}{\sqrt{m_l}} \sum_{j=1}^{m_l} W_{ij}^{(l)} \phi(h_j^{(l-1)}) + \sigma_b b_i^{(l)} \tag{6.1}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is the activation function, σ_w and σ_b define the variance scale of the weights and biases respectively. Given the parameterization that weights and biases are randomly generated by i.i.d. normal distribution, the pre-activations are Gaussian distributed in the infinite width limit as $m_1 \to \infty, m_2 \to \infty, \cdots, m_{L-1} \to \infty$. This results from the central limit theorem (CLT). Consider a dataset $X \in \mathbb{R}^{n \times m_0}$ of size $n = |X|$, the covariance matrix of Gaussian process kernel regarding infinitely-wide network is defined by $\Sigma^{(l)}(x, x') = \mathbb{E}[h_i^{(l)}(x) h_i^{(l)}(x')]$. According to the signal propagation Equation (6.1), the covariance matrix or GPK with respect to layer can be described by a recursion relation,

$$\Sigma^{(l)}(x,x') = \sigma_w^2 \mathbb{E}_{h\sim\mathcal{N}(0,\Sigma^{(l-1)})}[\phi(h(x))\phi(h(x'))] + \sigma_b^2 \tag{6.2}$$

The mean-field theory is a paradigm to study the limiting behavior of GPK, which is a measure of expressivity for networks [22, 23]. In particular, expressivity describes to what extent can two different inputs be distinguished. The property of evolution for expressivity $\Sigma^{(l)}(x,x')$ is determined by how fast it converges to its fixed point $\Sigma^*(x,x') \equiv \lim_{l\to\infty}\Sigma^{(l)}(x,x')$. It is shown that in almost the entire parameter space spanned by hyper-parameters σ_w and σ_b, the evolution exhibits a dramatical convergence rate formulated by an exponential function except for a critical line known as the *edge of chaos* [22, 23, 25]. Consequently, an infinitely-wide network loses its expressivity exponentially in most cases while retaining the expressivity at the edge of chaos.

6.2.2 Neural Tangent Kernel and Trainability

Most studies on infinitely-wide networks through mean-field theory have only focused on initialization without training. Jacot et al. [36] took a step further by considering the infinitely-wide network trained with the gradient descent method. Let $\mathcal{L}$ be the cost function. In the general case, the NTK varies with the training time, thus providing no substantial insight into the convergence property of neural networks. Interestingly, as shown by [36], the NTK converges to a limit kernel and does not change during training in the infinite width limit. This leads to a simple but profound result when using MSE loss, $\mathcal{L} = \frac{1}{2}\|f_t(X) - Y\|_F^2$, where Y is the label regarding the input X,

$$f_t(X) = (I - e^{-\eta\Theta_\infty(X,X)t})Y + e^{-\eta\Theta_\infty(X,X)t}f_0(X) \tag{6.3}$$

where Θ_∞ is the limiting kernel. As the training time t tends to infinity, the output function fits the label very well, i.e., $f_\infty(X) = Y$. As proved by [104] (Lemma 1), the network is trainable only if $\Theta_\infty(X, X)$ is non-singular. Quantitatively, the condition number $\kappa \equiv \lambda_{\max}/\lambda_{\min}$ can be a measure of trainability as applied and confirmed by [105].

6.2.3 GCNs

We define a graph as $G = (V, E)$. In this graph the V is a collection of nodes, on the other hand, E is a set of edges. We denote the number of nodes in graph G by $n = |V|$. The node features are denoted as $h_v \in \mathbb{R}^d$ for each $v \in V$, with d being the dimension of node features. Node classification task is to predict labels of unseen nodes by learning from a set of graphs $\mathcal{G} = \{G_i\}_{i=1}^{N_G}$ and their labels $\mathcal{Y} = \{y_i\}_{i=1}^{N_G}$, where N_G is the number of graph in the dataset. In this work, we develop our theory towards understanding the expressivity and trainability of GCNs on the node classification task.

GCNs iteratively update node features through aggregating and transforming the representation of their neighbors. Figure 6.1 illustrates an overview of the information propagation in a general GCN. We define a propagation unit to be the combination of a R-layer MLP and one aggregation operation. We use subscript (r) to denote the layer index of MLP in each propagation unit and use superscript (l) to indicate the index of aggregation operation, which is also the index of the propagation unit. L is the total number of propagation units. Specifically, the node representation propagation in GCNs through a multi-layer perceptron (MLP) follows the expression,

$$h_{(0)}^{(l)}(u) = \frac{1}{|\mathcal{N}(u)| + 1} \sum_{v \in \mathcal{N}(u) \cup u} h_{(R)}^{(l-1)}(v) \tag{6.4}$$

$$h_{(r)}^{(l)}(u) = \frac{\sigma_w}{\sqrt{m}} W_{(r)}^{(l)} \phi\big(h_{(r-1)}^{(l)}(u)\big) + \sigma_b b_{(r)}^{(l)} \tag{6.5}$$

where $W_{(r)}^{(l)}$ and $b_{(r)}^{(l)}$ are the learnable weights and biases respectively, ϕ is the activation function, $\mathcal{N}(u)$ is the neighborhood of node u, and $\mathcal{N}(u) \cup u$ is the union

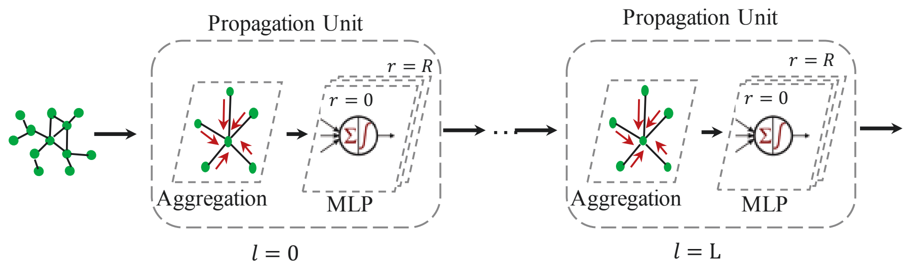

Figure 6.1 : Overview of the information propagation in a general GCN

of node v and its neighbors. Equation (6.4) describes the node feature aggregation operation among its neighborhood according to a GCN variant [11]. Equation (6.5) is a standard non-linear transformation with NTK-parameterization [36], where m is the width. With regard to the activation function, we focus on both ReLU and tanh, which are denoted as $\phi(x) = \max\{0, x\}$ and $\phi(x) = \tanh(x)$ respectively. Without loss of generality, our theoretical framework can handle other common activations. By comparison, the GNTK work [47] only adopted ReLU activation.

6.3 Theoretical Results

6.3.1 Expressivity of Infinitely Wide GCNs

To investigate the expressivity property of infinitely-wide GCNs, we define the covariance matrix between two input graphs G, G' by measuring correlation between node features as,

$$\Sigma^{(l)}_{(r)}(G, G')_{uu'} \equiv \mathbb{E}[h^{(l)}_{(r)}(u) h^{(l)}_{(r)}(u')], \tag{6.6}$$

where $\Sigma^{(l)}_{(r)}(G, G') \in \mathbb{R}^{n \times n'}$, $|V| = n$ and $|V'| = n'$. According to the CLT, the node representation $h^{(l)}_{(r)}(u)$ is Gaussian distributed in the infinite width limit. Applying this result to Equation (6.4), we can obtain the recursive expression for covariance matrix as follows,

$$\Sigma^{(l)}_{(0)}(G, G')_{uu'} = c_u c_{u'} \sum_{v} \sum_{v'} \Sigma^{(l-1)}_{(R)}(G, G')_{vv'}, \tag{6.7}$$

where $c_u = \frac{1}{|\mathcal{N}(u)|+1}$, $c_{u'} = \frac{1}{|\mathcal{N}(u')|+1}$. This equation is resulting from the neighborhood aggregation operation.

Next, we apply the GP result of infinitely-wide GCN to Equation (6.5). The propagation of the covariance matrix corresponds to the R times non-linear transformations, which is expressed as,

$$\Sigma^{(l)}_{(r)}(G, G')_{uu'} = \sigma_w^2 \mathbb{E}_{z_1,z_2}\big[\phi(z_1)\phi(z_2)\big] + \sigma_b^2 \tag{6.8}$$

where z_1, z_2 are drawn from a 2-dimension Gaussian distribution in the previous MLP layer, i.e., $z_1, z_2 \sim \mathcal{N}\big(0, \tilde{\Sigma}^{(l)}_{(r-1)}(G, G')\big) \in \mathbb{R}^{2\times 2}$, where the variance $\tilde{\Sigma}^{(l)}_{(r-1)}(G, G')$ is defined as a 2×2 matrix, where elements are:

$$\tilde{\Sigma}^{(l)}_{(r)}(G, G')_{uu'} = \begin{pmatrix} \Sigma^{(l)}_{(r)}(G, G)_{uu} & \Sigma^{(l)}_{(r)}(G, G')_{uu'} \\ \Sigma^{(l)}_{(r)}(G', G)_{u'u} & \Sigma^{(l)}_{(r)}(G', G')_{u'u'} \end{pmatrix} \tag{6.9}$$

We study the behavior of $\Sigma^{(l)}_{(r)}(G, G')$ as $l \to \infty$ following the diagram of mean-field theory. However, both aggregation (Equation 6.7) and transformation (Equation 6.8) contribute simultaneously to the limiting result. To make the analysis easy to understand, we first disassemble the two operations to study their limiting behaviors separately.

Consider a GCN without non-linear transformation, i.e. $R = 0$ and $\Sigma^{(l)}_{(r)}(G, G') = \Sigma^{(l)}(G, G')$. In order to facilitate calculation, we rewrite Equation (6.7) as the following format,

$$\vec{\Sigma}^{(l)}(G, G') = \mathcal{A}(G, G')\vec{\Sigma}^{(l-1)}(G, G') \tag{6.10}$$

where $\vec{\Sigma}^{(l)}(G, G') \in \mathbb{R}^{nn'\times 1}$ is the result of being vectorized, and $\mathcal{A}(G, G') = \mathcal{A}^{(1)}(G, G') = \cdots = \mathcal{A}^{(l)}(G, G') \in \mathbb{R}^{nn'\times nn'}$ is a probability transition matrix, and the limiting behavior of $\Sigma^{(l)}(G, G')$ is shown by the following lemma,

Lemma 6.1 (Convergence of aggregation)**.** *Assume R=0, then*

$$\lim_{l\to\infty} \Sigma^{(l)}(G, G')_{uu'} = \pi(G, G')\vec{\Sigma}^{(0)}(G, G')$$

where $\pi(G,G') \in \mathbb{R}^{1\times nn'}$*, satisfying,*

$$\pi(G,G')\mathcal{A}^{(l)}(G,G') = \pi(G,G')$$

In Lemma 6.1, we demonstrate that the propagation of the transition matrix $\mathcal{A}(G,G')$ conforms to the transitions of the Markov chain [106, 107]. In the limiting case, i.e. $l \to \infty$, the transition of the $\mathcal{A}(G,G')$ evolves as a stationary state $\pi(G,G')\mathcal{A}(G,G') = \pi(G,G')$. As a result, according to Equation (6.10), the covariance matrix $\Sigma^{(l)}(G,G')$ collapses exponentially to a constant matrix.

Then we consider a network with only non-linear transformation, which is known as a pure MLP. This leads to $\Sigma^{(l)}_{(r)}(G,G) = \Sigma_{(r)}(G,G)$, where we use subscript (r) to denote the layer index. The large depth behavior has been well studied in [29], and we introduce the result when the *edge of chaos* is realized. In particular, we set the value of hyper-parameters to satisfy,

$$\sigma_w^2 \int \mathcal{D}_z[\phi'(\sqrt{q^*}z)]^2 = 1 \tag{6.11}$$

where q^* is the fixed point of diagonal elements in the covariance matrix, and $\int \mathcal{D}z = \frac{1}{\sqrt{2\pi}} \int dz e^{-\frac{1}{2}z^2}$ is the measure for a normal distribution. For the ReLU activation, Equation (6.11) requires $\sigma_w^2 = 2$ and $\sigma_b^2 = 0$.

The key idea is to study the asymptotic behavior of the normalized correlation defined as,

$$C_r(G,G')_{uu'} \equiv \frac{\Sigma_{(r)}(G,G')_{uu'}}{\sqrt{\Sigma_{(r)}(G,G)_{uu}\,\Sigma_{(r)}(G',G')_{u'u'}}}. \tag{6.12}$$

Lemma 6.2 (Proposition 1 and 3 in [29]). *Consider an infinitely-wide fully-connected network with a Lipschitz nonlinearity ϕ, then,*

- $\phi(x) = \max\{0,x\}$, $1 - C_r(G,G')_{uu'} \sim \frac{9\pi^2}{2r^2}$ as $r \to \infty$
- $\phi(x) = \tanh(x)$, $1 - C_{(r)}(G,G')_{uu'} \sim \frac{\beta}{r}$ as $r \to \infty$ *where* $\beta = \frac{2\int \mathcal{D}_z[\phi'(\sqrt{q^*}z)^2]}{q^*\mathcal{D}_z[\phi''(\sqrt{q^*}z)^2]}$.

Lemma 6.2 shows the covariance matrix converges to a constant matrix at a polynomial rate of $1/r^2$ for ReLU and of $1/r$ for tanh activation on the edge of chaos. This implies that a network without aggregation could retain its expressivity at a large depth.

Taking both aggregation and transformation into consideration, we derive our theorem on the asymptotic behavior of infinitely-wide GCN in the large depth limit, which is as follows:

Theorem 6.1 (Convergence rate for GPK)**.** *If $\mathcal{A}(G, G')$ is ireducible and aperiodic, with a stationary distribution vector $\pi(G, G')$, then there exist constants $0 < \alpha < 1$ and $C > 0$, and constant vector $v \in \mathbb{R}^{nn' \times 1}$ depending on the number of MLP iterations R, such that*

$$|\Sigma^{(l)}_{(r)}(G, G')_{uu'} - \pi(G, G')v| \leq C\alpha^l$$

In Theorem 6.1, we rigorously characterize the convergence properties of GCNs' expressivity in the large depth limit. It reveals the covariance matrix $\Sigma^{(l)}_{(r)}(G, G')$ converges to a constant matrix at an exponential rate. Notably, the vector v depends on the number of non-linear transformations R, which implies that non-linear transformation may mildly slow down the convergence rate.

6.3.2 Trainability of Infinitely Wide GCNs

According to the definition of NTK (Equation 5.6), we can formulate the propagation of GNTK with respect to aggregation (Equation 6.4) and non-linear transformation (Equation 6.5) as follows,

$$\Theta^{(l)}_{(0)}(G, G')_{uu'} = c_u c_{u'} \sum_{v} \sum_{v'} \Theta^{(l-1)}_{(R)}(G, G')_{vv'} \tag{6.13}$$

and,

$$\Theta^{(l)}_{(r)}(G, G')_{uu'} = \Theta^{(l)}_{(r-1)}(G, G')_{uu'} \dot{\Sigma}^{(l)}_{(r)}(G, G')_{uu'} + \Sigma^{(l)}_{(r)}(G, G')_{uu'} \tag{6.14}$$

where $\Sigma^{(l)}_{(r)}(G,G')_{uu'}$ is defined by Equation (6.8) and $\dot{\Sigma}^{(l)}_{(r)}(G,G')_{uu'}$ is governed by,

$$\dot{\Sigma}^{(l)}_{(r)}(G,G')_{uu'} \equiv \sigma_w^2 \mathbb{E}_{z_1,z_2}\big[\dot{\phi}(z_1)\dot{\phi}(z_2)\big]. \tag{6.15}$$

Note that the propagation of the GNTK corresponding to aggregation (Equation 6.13) is the same as that of GPK (Equation 6.7). Meanwhile, the GNTK with regarding to non-linear transformation (Equation 6.14) can be computed in closed form in terms of GPK. The following theorem demonstrates that the aggregation leads to an exponential decay of GNTK as the depth becomes large.

Theorem 6.2 (Convergence rate for GNTK)**.** *If $\mathcal{A}(G,G')$ is ireducible and aperiodic, with a stationary distribution $\pi(G,G')$, then there exist constant matrix v and v' depending on R, such that*

$$|\Theta^{(l)}_{(r)}(G,G')_{uu'} - \pi(G,G')\big(Rlv + v'\big)| \leq C\alpha^l$$

As depth l goes to infinity, all elements in the GNTK converge to a unique quantity at an exponential rate. This result is a little different to the convergence result of the GPK in Theorem 6.1, in which the convergence limit of the GPK is a constant rather than a quantity that varies with depth. We thus have $\Theta^{(L)}_{(R)}(G,G') \approx \pi(G,G')(RLv)\mathbf{1}_{nn'}$ as $L \to \infty$, where $\mathbf{1}_{nn'}$ is the $(n \times n')$-dimensional matrix whose elements are one. The exponential convergence rate of $\Theta^{(L)}_{(R)}(G,G')$ implies that the trainability of infinitely wide GCNs degenerate dramatically, as shown in following corollary.

Corollary 6.1 (Trainability of ultra-wide GCNs)**.** *Consider a GCN of the form (6.4) and (6.5), with depth L, non-linear transformations number R, an MSE loss, and a Lipchitz activation $\phi(x)$, trained with gradient descent on a node classification task. Then the output function follows,*

$$f_t(\mathcal{G}) = e^{-\eta\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G})t} f_0(\mathcal{G}) + (I - e^{-\eta\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G})t}\mathcal{Y})$$

where $\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G}) \in \mathbb{R}^{\sum_{i=1}^{N_G} n_i \times \sum_{i=1}^{N_G} n_i}$ *with the expression,*

$$\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G}) = \begin{pmatrix} \Theta^{(L)}_{(R)}(G_1,G_1) & \cdots & \Theta^{(L)}_{(R)}(G_1,G_N) \\ \vdots & \ddots & \vdots \\ \Theta^{(L)}_{(R)}(G_N,G_1) & \cdots & \Theta^{(L)}_{(R)}(G_N,G_N) \end{pmatrix}$$

There exists a positive integer L_0 *such that,* $\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G})$ *is singular when* $L > L_0$*. Moreover, there exist a constant* $C > 0$ *such that,* $\|f_t(\mathcal{G}) - \mathcal{Y}\| > C$*.*

According to the above corollary, as $L \to \infty$, the GNTK matrix will become a singular matrix. This would lead to a discrepancy between outputs $f_t(\mathcal{G})$ and labels $\mathcal{Y}$, which means GNTK loses the ability to fit the label. Therefore, an ultra-wide GCN with a large depth cannot be trained successfully on node classification tasks.

6.3.3 Analysis on techniques to deepen GCNs

We have characterized the expressivity and trainability of deep GCNs through GPKs and GNTKs, respectively. We show that both expressivity and trainability of ultra-wide GCNs drop at an exponential rate. Recently, enormous efforts have been made to deepen GCNs, of which residual connection-resemble techniques are widely applied to resolve the overs-moothing problem, including PPNP [108], DropEdge [109], and residual GCN [110].

Analysis on PPNP and DropEdge

We first consider techniques of PPNP and DropEdge. Although they are formed with different principles, the core idea is the same under the infinite-width limit. From the perspective of probability flow, it reduces the probability flow from neighboring nodes to itself, and correspondingly, increases the probability flow from the node to itself. PPNP is based on personalized PageRank, and has a similar mechanism as DropEdge [108]. Therefore, DropEdge and PPNP actually lead to the same consequence, which is to preserve node locality to make node representations discriminative. We formulate aggregation of PPNP and DropEdge as follows,

$$h_{(0)}^{(l)}(u) = (1-\delta)c_u \sum_{v\in\mathcal{N}(u)\cup u} h_{(R)}^{(l-1)}(v) + \delta h_{(R)}^{(l-1)}(u) \tag{6.16}$$

where $c_u = \frac{1}{\mathcal{N}(u)+1}$, and $0 < \delta \leq 1$. Here δ is teleport (or restart) probability in PPNP [108], and drop rate in DropEdge [109]. Then the propagation of aggregation with Equation (6.16) can be reformulated as,

$$\vec{\Sigma}^{(l)}(G,G') = (1-\tilde{\delta})\mathcal{A}(G,G')\vec{\Sigma}^{(l-1)}(G,G') + \tilde{\delta}\vec{\Sigma}^{(l-1)}(G,G'), \tag{6.17}$$

where $0 < \tilde{\delta} < 1$ is an effective drop rate or teleport probability after re-normalization. This is equivalent to the residual connection between aggregations. Thus we attribute these techniques as residual connection-resemble methods. We take Equation (6.17) as a new aggregation operation matrix, $\vec{\Sigma}^{(l)}(G,G') = \tilde{\mathcal{A}}(G,G')\vec{\Sigma}^{(l-1)}(G,G')$, where $\tilde{\mathcal{A}}(G,G') = (1-\tilde{\delta})\mathcal{A}(G,G') + \tilde{\delta}I$. We prove that $\tilde{\mathcal{A}}(G,G')$ is also a transition matrix with a greater second largest eigenvalue compared to the original matrix $\mathcal{A}(G,G')$.

Theorem 6.3 (Convergence rate for residual connection-resemble aggregation)**.** *Consider a covariance matrix of GPK with the form (6.17) without non-linear transformation, i.e., $R = 0$. Then with a stationary vector $\tilde{\pi}(G,G')$ for $\tilde{\mathcal{A}}(G,G')$,*

$$|\Sigma_{(r)}^{(l)}(G,G')_{uu'} - \tilde{\pi}(G,G')\vec{\Sigma}^{(0)}(G,G')| \leq C\alpha^l$$

Furthermore, we denote the second largest eigenvalue of $\tilde{\mathcal{A}}(G,G')$ and $\tilde{\mathcal{A}}(G,G')$ as λ_2 and $\tilde{\lambda}_2$, respectively. Then,

$$\tilde{\lambda}_2 > \lambda_2$$

Remark 6.1. *In Theorem 6.3, we show that the second largest of the new transition matrix $\tilde{\mathcal{A}}(G,G')$ is greater than that of the original transition matrix $\mathcal{A}(G,G')$. This is consistent with what demonstrated in Theorem 1 of [109] that DropEdge alleviates the ϵ-smoothing issue.*

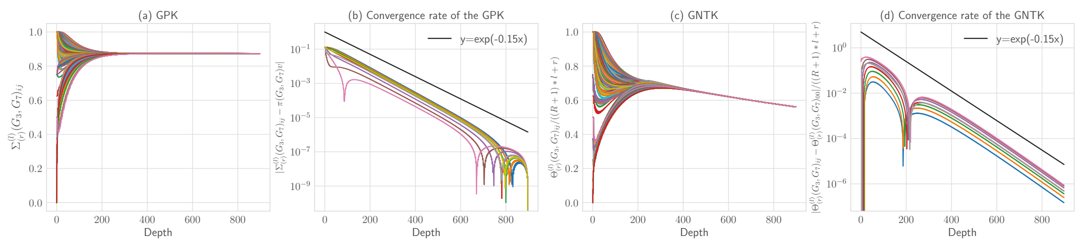

Figure 6.2 : Convergence rate for the GPK and the GNKT. (a) Value changes of the GPK elements as the depth grows; although their initial values are different, they all tend to the same value as depth increases. (b) The distance changes between GPK elements and their limiting value as the depth grows; the converge rate can be bounded by a exponential function $y = \exp(-0.15x)$. (c) Value changes of re-normalized GNTK elements as the depth grows. (d) The distance changes between re-normalized GNTK elements and a random element from GNTK as the depth grows. The converge rate can be bounded by a exponential function $y = \exp(-0.15x)$.

Since the decay speed is determined by the second largest eigenvalue of the transition matrix [106, 107], the DropEdge, PPNP methods can slow down the decay rate.

Analysis on residual-connection GCN

We firstly consider residual connection that is only applied on non-linear transformations (MLP). In this case, the signal propagation change from Equation (6.5) to:

$$h^{(l)}_{(r)}(u) = h^{(l)}_{(r-1)}(u) + \frac{\sigma_w}{\sqrt{m}} W^{(l)}_{(r)} \phi\big(h^{(l)}_{(r-1)}(u)\big) \tag{6.18}$$

As a result, the recursive equation for the corresponding GNTK can be expressed as,

$$\Theta^{(l)}_{(r)}(G, G')_{uu'} = \Theta^{(l)}_{(r-1)}(G, G')_{uu'} \big(\dot{\Sigma}^{(l)}_{(r)}(G, G')_{uu'} + 1\big) + \Sigma^{(l)}_{(r)}(G, G')_{uu'} \tag{6.19}$$

This formula is similar to the vanilla GNTK in the infinitely wide limit. Only an additional residual term appears according to the residual connection. It turns out that this term can help to slow down the convergence rate for non-linear transformation. We then consider the effect made by the residual connection on aggregation.

Theorem 6.4 (Convergence rate for GNTK with residual connection between transformations)**.** *Consider a GCN of the form (6.4) and (6.18). If $\mathcal{A}(G, G')$ is irreducible and aperiodic, with a stationary distribution $\pi(G, G')$, then there exist constant matrix v and v' depending on R, such that,*

$$|\Theta^{(l)}_{(r)}(G, G')_{uu'} - \pi(G, G')\big(Rl(1 + \frac{\sigma_w^2}{2})^{Rl} v + v'\big)| \leq C\alpha^l$$

Theorem 6.4 demonstrates that even though the residual connection can slow down the decay during non-linear transformation, it cannot stop the exponential decay rate resulting from aggregation.

Finally, we consider the residual connection applied to both aggregation and non-linear transformation simultaneously:

Corollary 6.2 (Convergence rate for GNTK with residual connection between aggregation and transformations)**.** *Consider a GCN of the form (6.16) and (6.18). If $\tilde{\mathcal{A}}(G, G')$ is irreducible and aperiodic, with a stationary distribution $\tilde{\pi}(G, G')$, then*

$$|\Sigma^{(l)}_{(r)}(G, G')_{uu'} - \tilde{\pi}(G, G')(1 + \frac{\sigma_w^2}{2})^{Rl} v| \leq C\beta^l$$

$$|\Theta^{(l)}_{(r)}(G, G')_{uu'} - \tilde{\pi}(G, G')\big(Rl(1 + \frac{\sigma_w^2}{2})^{Rl} v\big)| \leq C\beta^l$$

According to Corollary 6.2, the residual connection-resemble techniques can slow down the decay speed to a certain extent, though they cannot solve exponential trainability loss for node classification.

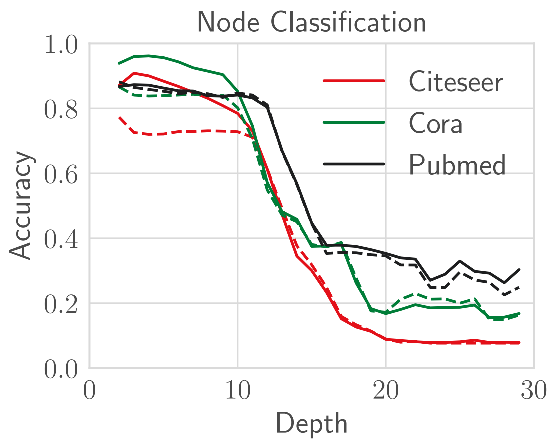

Figure 6.3 : Train and test accuracy depending on the depth on different datasets. Solid lines are train accuracy and dashed lines are test accuracy.

6.4 Experiments

6.4.1 Setup

We illustrate empirically the theoretical results obtained in section 6.3. We use one bioinformatics dataset (i.e., MUTAG) and three citation network datasets. We summarize the properties of datasets in Table 6.1. To verify Theorems 6.1, 6.2, and 6.4 in general, i.e., G, G' can be two different graphs, we use the MUTAG data set. At the same time, the dataset with only one graph, i.e., $G = G'$, is a special case and can be described by our theorems in this work. The remaining three data sets are used to verify the trainability of the network on node classification tasks.

6.4.2 Convergence Rate of GPKs and GNTKs

Theorems 6.1 and 6.2 provide theoretical convergence rates for the GPK $\Sigma(G, G')$ and the GNTK $\Theta(G, G')$, respectively. We demonstrate these results in Figure 6.2. We select two graphs denoted as G_3 and G_7 from the MUTAG dataset. The GPK

Table 6.1 : Details of Datasets

Dataset	Graphs	Nodes	Classes	Features	Avg. nodes
MUTAG	188	-	2	-	17.9
Citeseer	1	3327	6	3703	-
Cora	1	2708	7	1433	-
Pubmed	1	19717	3	500	-

and GNTK are generated from the infinitely-wide GCNs with ReLU activation[*], with $R = 3$, and $L = 300$. Figure 6.2(a) shows all elements of a covariance matrix kernel (GPK) converge to an identical value as the depth goes larger. Figure 6.2(b) further illustrates the convergence rate of the GPK is exponential, as predicted by Theorem 6.1. Then we demonstrate the exponential convergence rate of GNTK in Theorem 6.2 by comparing the distance between elements of the GNKT matrix, summarized in Figure 6.2(c,d). Additionally, we verify the convergence results of GCNs with residual connection demonstrated in Theorem 6.4 and leave the demonstration in the appendix.

6.4.3 Trainability of Ultra-Wide GCNs

We examine whether ultra-wide GCNs can be trained successfully on node classification. We conduct the experiment on a GCN [9], where we apply a width of 1000 at each hidden layer and the depth ranging from 2 to 29. Figure 6.3 displays the train and test accuracy on various datasets after 300 training epochs. These results show the dramatic drop on both train and test accuracy as the depth grows, comfirming that wide GCNs lose trainability significantly in the large depth on node classification as predicted by Corollary 6.1.

[*]We use the implementation of GNTK available at https://github.com/KangchengHou/gntk [47].

6.5 Conclusion

We have characterized the asymptotic behavior of GPK and GNTK to respectively measure the expressivity and trainability of infinitely-wide GCNs in the large depth. We prove that both expressivity and trainability of infinitely-wide GCNs drop at an exponential rate due to the aggregation operation. Furthermore, we apply our theoretical framework to investigate the extent to which residual connection-resemble techniques could alleviate over-smoothing problem on node classification tasks, including PPNP, DropEdge, and residual GCNs. We demonstrate that these techniques can only slow down the decay speed, but unable to solve the exponential decay problem in essence. Our theory is finally verified by the experimental results.

6.6 Proof

This section is dedicated to provide proofs of all lemmas and theorems in this article

Lemma 6.1. *Assume R=0, then for $\forall u, u'$,*

$$\lim_{l\to\infty} \Sigma^{(l)}(G,G')_{uu'} = \pi(G,G')\vec{\Sigma}^{(0)}(G,G')$$

where $\pi(G,G') \in \mathbb{R}^{1\times nn'}$, satisfying,

$$\pi(G,G')\mathcal{A}^{(l)}(G,G') = \pi(G,G')$$

Proof. When $R=0$ and $c_u = \frac{1}{|\mathcal{N}(u)|+1}$, equation (6.7) reduces to,

$$\Sigma^{(l)}_{(0)}(G,G')_{uu'} = \frac{1}{|\mathcal{N}(u)|+1}\frac{1}{|\mathcal{N}(u')|+1}\sum_{v\in\mathcal{N}(u)\cup u}\sum_{v'\in\mathcal{N}(u')\cup u'}\Sigma^{(l-1)}_{(0)}(G,G')_{vv'}$$

In order to facilitate calculation, we define the fraction and sum operation in the format of matrix,

$$\vec{\Sigma}^{(l)}_{(0)}(G,G') = \mathcal{A}^{(l)}\vec{\Sigma}^{(l-1)}_{(0)}(G,G')$$

where $\vec{\Sigma}^{(l)}_{(0)}(G, G') \in \mathbb{R}^{nn'\times 1}$, is the result of being vectorized. Thus the matrix operation $\mathcal{A}^{(l)} \in \mathbb{R}^{nn'\times nn'}$.

This implies that the aggregate operation is the same for each layer. The next step is to prove $\mathcal{A}^{(l)}$ is a stochastic matrix (transition matrix):

$$\sum_j \mathcal{A}^{(l)}(G, G')_{ij} = 1.$$

According to Equation 6.7, we know that $\mathcal{A}^{(l)}$ is a Kronecker product of two matrix,

$$\mathcal{A}^{(l)}(G, G')_{uu'} = \big[\mathcal{B}(G)\mathcal{C}(G)\big] \otimes \big[\mathcal{B}(G')\mathcal{C}(G')\big]$$

where $\mathcal{B}(G), \mathcal{C}(G) \in \mathbb{R}^{n\times n}$ and $\mathcal{B}(G'), \mathcal{C}(G') \in \mathbb{R}^{n'\times n'}$.

(1) $\mathcal{B}(G)$ and $\mathcal{B}(G')$ are diagonal matrix, which correspond to the factor $\frac{1}{\mathcal{N}(u)+1}$ or $\frac{1}{\mathcal{N}(u')+1}$.

$$\mathcal{B}(G) = \begin{pmatrix} \frac{1}{\mathcal{N}(u_1)+1} & & \\ & \ddots & \\ & & \frac{1}{\mathcal{N}(u_n)+1} \end{pmatrix}$$

(2) The element of matrix $\mathcal{C}(G)$ and $\mathcal{C}(G')$ are determined by whether there is a edge between two vertexes,

$$\mathcal{C}(G)_{ij} = \tilde{\delta}_{ij}$$

where $\tilde{\delta}_{ij} = 1$ if $i == j$ or there is edge between vertex i and j, else $\tilde{\delta}_{ij} = 0$.

We first use the property of matrix $\mathcal{B}$ and $\mathcal{C}$ before Kronecker product,

$$\sum_j \big[\mathcal{B}(G)\mathcal{C}(G)\big]_{ij} = \frac{1}{\mathcal{N}(u_i)+1}\sum_j \tilde{\delta}_{ij} = \frac{1}{\mathcal{N}(u_i)+1}(\mathcal{N}(u_i)+1) = 1$$

According to the definition of Kronecker product,

$$\sum_j \mathcal{A}^{(l)}(G, G')_{ij} = \sum_b \sum_d [\mathcal{B}(\mathcal{G})\mathcal{C}(\mathcal{G})]_{ab}[\mathcal{B}(\mathcal{G}')\mathcal{C}(\mathcal{G}')]_{cd} = 1$$

where $i = a + (c-1)n$, and $j = b + (d-1)n$.

So far, we have proved that $\mathcal{A}^{(l)}(G,G')$ is a stochastic matrix. According to the Perron-Frobenius Theory,

$$\pi(G,G')\mathcal{A}^{(l)} = \pi(G,G')$$

The convergence rate is governed by the second largest eigenvalue.

Since $\lim_{l\to\infty}\mathcal{A}^l_{ij}(G,G') = \pi_j(G,G')$, we have,

$$\begin{aligned}\lim_{l\to\infty}\vec{\Sigma}^{(l)}(G,G') &= \lim_{l\to\infty}\mathcal{A}^l(G,G')\vec{\Sigma}^{(0)}(G,G') \\ &= \Pi(G,G')\vec{\Sigma}^{(0)}(G,G')\end{aligned}$$

where $\Pi(G,G') = \begin{pmatrix}\pi(G,G') \\ \pi(G,G') \\ \vdots \\ \pi(G,G')\end{pmatrix}$ is the $nn' \times nn'$ matrix all of whose rows are the stationary distribution. Then, we can see that every element in $\vec{\Sigma}^{(l)}(G,G')$ converges exponentially to an identical value, depending on the stationary distribution and initial state,

$$\lim_{l\to\infty}\Sigma^{(l)}(G,G')_{uu'} = \pi(G,G')\vec{\Sigma}^{(0)}(G,G')$$

$\square$

Lemma 6.2 (Proposition 1 and 3 in [29]). *Assume $L=0$, with a Lipschitz nonlinearity ϕ, then,*

- $\phi(x) = (x)_+$, $1 - C_r(G,G')_{uu'} \sim \frac{9\pi^2}{2r^2}$ as $r\to\infty$
- $\phi(x) = \tanh(x)$, $1 - C_{(r)}(G,G')_{uu'} \sim \frac{\beta}{r}$ as $r\to\infty$ *where* $\beta = \frac{2\mathbb{E}[\phi'(\sqrt{q^*}z)^2]}{q\mathbb{E}[\phi''(\sqrt{q^*}z)^2]}$.

Proof. We first decompose the integral calculation in the equation 6.8 into two parts, one is diagonal element and the other is non-diagonal element:

$$\Sigma_{(r)}(G,G)_{uu} = \sigma_w^2\int_{\mathcal{D}z}\phi^2(\sqrt{\Sigma_{(r-1)}(G,G)_{uu}}z) + \sigma_b^2$$

$$\Sigma_{(r)}(G, G')_{uu'} = \sigma_w^2 \int_{\mathcal{D}z_1\mathcal{D}z_2} \phi(u_1)\phi(u_2) + \sigma_b^2$$

For simplicity, we define $q_r(G) = \Sigma_{(r)}(G, G)_{uu}$, $q_r(G') = \Sigma_{(r)}(G', G')_{u'u'}$, and $c_r(G, G') = C_{(r)}(G, G')_{uu'}$.

When $\phi(x) = \max\{0, x\}$. The first integration for $q_r(G)$ reduces to,

$$q_r(G) = \frac{\sigma_w^2}{2} q_{r-1}(G) + \sigma_b^2$$

Next step is to find the fixed-point,

$$\lim_{r\to\infty} q_r(G) = q(G)$$

For all G, G' there exists r_0 such that $|q_r(G) - q_r(G')| < \delta$ for all $r > r_0$. Then the second integration for $c_r(G, G')$ becomes,

$$c_r(G, G') = \frac{\sigma_w^2 \int_{\mathcal{D}z_1\mathcal{D}z_2} \phi\big(\sqrt{q_{r-1}(G)}z_1\big)\phi\big(\sqrt{q_{r-1}(G)}(c_{r-1}(G, G')z_1 + \sqrt{1 - c_{r-1}(G, G')^2}z_2)\big) + \sigma_b^2}{q_{r-1}(G)}$$

To investigate the propagation of $c_r(G, G')$, we set $q_r(G) = q_r(G') = q$ for any $r > r_0$, and define,

$$f(x) = \frac{\sigma_w^2 \int_{\mathcal{D}z_1\mathcal{D}z_2} \phi\big(\sqrt{q}z_1\big)\phi\big(\sqrt{q}(xz_1 + \sqrt{1 - x^2}z_2)\big) + \sigma_b^2}{q}$$

(1) when $\sigma_w^2 < 2$, there is a unique fixed point $c_r(G, G') = 1$.

(2) when $\sigma_w^2 = 2$. The derivative of $f(x)$ satisfies,

$$f'(x) = 2 \int_{\mathcal{D}z_1\mathcal{D}z_2} 1_{z_1>0} 1_{xz_1+\sqrt{1-x^2}z_2>0}$$

So using the equation above and the condition $f'(0) = \frac{1}{2}$, we can get another formation of the derivative of $f(x)$,

Since $\int \arcsin = x \arcsin + \sqrt{1 - x^2}$ and $f(1) = 1$, then for $x \in [0, 1]$,

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

Substituting $f(x) = c_r(G, G')$ and $x = c_{r-1}(G, G')$, into expression up here, we have,

$$c_r(G, G') = \frac{2c_{r-1}(G, G')\arcsin(c_{r-1}(G, G')) + 2\sqrt{1 - c_{r-1}^2(G, G')} + c_{r-1}(G, G')\pi}{2\pi}$$

Now we study the behavior of $c_r(G, G')$ as r tends to infinity. Using Taylor expansion, we have,

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

By induction analysis, the sequence $c_r(G, G')$ is increasing to the fixed point 1. Besides, we can replace x with $c_r(G, G')$,

$$c_{r+1}(G, G') = c_r(G, G') + \frac{2\sqrt{2}}{3\pi}(1 - c_r(G, G'))^{3/2} + O((1 - c_r(G, G'))^{5/2})$$

Let $\gamma_r = 1 - c_r(G, G')$, then we have,

$$\gamma_{r+1} = \gamma_r - \frac{2\sqrt{2}}{3\pi}\gamma_r^{3/2} + O(\gamma_r^{5/2})$$

so that,

$$\gamma_{r+1}^{-1/2} = \gamma_r^{-1/2}(1 - \frac{2\sqrt{2}}{3\pi}\gamma_r^{1/2} + O(\gamma_r^{3/2}))^{-1/2} = \gamma_r^{-1/2} + \frac{\sqrt{2}}{3\pi} + O(\gamma_r)$$

As r tends to infinity, we have,

$$\gamma_{r+1}^{-1/2} - \gamma_r^{-1/2} \sim \frac{\sqrt{2}}{3\pi}$$

It means,

$$\gamma_r^{-1/2} \sim \frac{\sqrt{2}}{3\pi}r$$

Therefore, we have,

$$1 - c_r(G, G') \sim \frac{9\pi^2}{2r^2}$$

When $\phi(x) = \tanh(x)$, a Taylor expansion near 1 yields,

$$f(x) = 1 + (x-1)f'(1) + \frac{(x-1)^2}{2}f''(1) + O((x-1)^{5/2})$$

Denote $\gamma_r = 1 - c_r(G, G')$, then we have,

$$\gamma_{r+1} = \gamma_r - \frac{\gamma_r^2}{\beta} + O(\gamma_l^{5/2})$$

where $\beta = \chi_2$. Therefore,

$$\gamma_{r+1}^{-1} = \gamma_r^{-1}(1 - \frac{\gamma_r}{\beta} + O(\gamma_r^{3/2})) = \gamma_r^{-1} + \frac{1}{\beta} + O(\gamma_r^{1/2}).$$

Thus, we have,

$$1 - c_{(r)}(G, G') \sim \frac{\beta}{l} \text{ as } l \to \infty$$

$\square$

Theorem 6.1. *If $\mathcal{A}(G, G')$ is ireducible and aperiodic, with a stationary distribution vector $\pi(G, G')$, then*

$$|\Sigma_{(r)}^{(l)}(G, G')_{uu'} - \pi(G, G')v| \leq C\alpha^l$$

Proof. We prove the result by induction method. For $l = 0$, according to the Cauchy-Buniakowsky-Schwarz Inequality

$$\Sigma_{(0)}^{(0)}(G, G')_{uu'} = h_u^{(0)} h_{u'}^{(0)} \leq ||h_u^{(0)}||_2 ||h_{u'}^{(0)}||_2 = 1$$

Thus over feature initialization, such that,

$$|\Sigma_{(0)}^{(0)}(G, G')_{uu'} - \pi(G, G')v| < C$$

Assume the result is valid for $\Sigma_{(r)}^{(l)}(G, G')_{uu'}$, then we have,

$$|\Sigma_{(r)}^{(l)}(G, G') - \pi(G, G')v| \leq C\alpha^l$$

Now we consider the distance between $\Sigma_{(r)}^{(l+1)}(G, G')$ and $C\alpha^{l+1}$. To compute this, we need to divide the propagation from l layer to $l + 1$ layer into three parts:

(i) $\Sigma_{(r)}^{(l)} \to \Sigma_{(r+1)}^{(l)} \to \cdots \to \Sigma_{(R)}^{(l)}$

(ii) $\Sigma^{(l)}_{(R)} \to \Sigma^{(l+1)}_{(0)}$

(iii) $\Sigma^{(l+1)}_{(0)} \to \Sigma^{(l+1)}_{(1)} \to \cdots \to \Sigma^{(l+1)}_{(r)}$.

For (i), we first prove that the correlation term $C_{(r)}(G, G')_{uu'}$ is close to 1 sufficiently. Recall that,

$$C_{(r)}(G, G')_{uu'} = \Sigma_{(r)}(G, G')_{uu'} / \sqrt{\Sigma_{(r)}(G, G)_{uu} \Sigma_{(r)}(G', G')_{u'u'}}$$

then we have,

$$1 - C_{(r)}(G, G')_{uu'} = \frac{\sqrt{\Sigma_{(r)}(G, G)_{uu} \Sigma_{(r)}(G', G')_{u'u'}} - \Sigma_{(r)}(G, G')_{uu'}}{\sqrt{\Sigma_{(r)}(G, G)_{uu} \Sigma_{(r)}(G', G')_{u'u'}}}(\alpha^l)$$

The last equation is due to the fact that every element is exponentially close to a bounded constant.

Recall the property of MLP propagation function $f(x)$ for $x = C_{(r)}(G, G')_{uu'}$, when $C_{(r)}(G, G')_{uu'}$ is close to 1:

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

This implies,

$$\begin{aligned} 1 - C_{(r+1)}(G, G')_{uu'} = 1 - C_{(r)}(G, G')_{uu'} - \frac{2\sqrt{2}}{3\pi}(1 - C_{(r)}(G, G')_{uu'})^{\frac{3}{2}} \\ + O(1 - C_{(r)}(G, G')_{uu'})^{\frac{5}{2}} \le 1 - C_{(r)}(G, G') = O(\alpha^l) \end{aligned}$$

From this result, we have,

$$\begin{aligned} &|\Sigma^{(l)}_{(r+1)}(G, G')_{uu'} - \pi(G, G')v| \\ &= |\Sigma^{(l)}_{(r+1)}(G, G')_{uu'} - \Sigma^{(l)}_{(r)}(G, G')_{uu'} + \Sigma^{(l)}_{(r)}(G, G')_{uu'} - \pi(G, G')v| \\ &\le |\Sigma^{(l)}_{(r)}(G, G')_{uu'} - \pi(G, G')v| \\ &\le C'\alpha^l. \end{aligned}$$

Repeat the proof process, we have a relation for $\Sigma^{(l)}_{(R)}(G, G')_{uu'}$ at the last step of (i),

$$|\Sigma^{(l)}_{(R)}(G, G')_{uu'} - \pi(G, G')v| \le C\alpha^l.$$

Secondly, (ii) we go through an aggregation operation $\mathcal{A}(G, G')$, then we have,

$$\vec{\Sigma}^{(l+1)}_{(0)}(G, G') = \mathcal{A}(G, G')\vec{\Sigma}^{(l)}_{(R)}(G, G') = \mathcal{A}(G, G')(\Pi(G, G')v + \vec{O}(\alpha^l))$$

where $\vec{O}(\alpha^l)$ denotes a vector in which every element is bounded by α^l.

Because $\Pi(G, G')v$ can be seen as a result of $\mathcal{A}^l(G, G')$, with a standard argument for convergence Theorem of stochastic matrix [111], we can obtain

$$\vec{\Sigma}^{(l+1)}_{(0)}(G, G') = \Pi(G, G')v + \vec{O}(\alpha^{l+1})$$

Therefore,

$$|\Sigma^{(l+1)}_{(0)}(G, G') - \pi(G, G')v| \leq C\alpha^{l+1}.$$

Finally, (iii) Repeat the result in step (i), we have,

$$|\Sigma^{(l+1)}_{(r)}(G, G') - \pi(G, G')v| \leq C\alpha^{l+1}.$$

$\square$

Theorem 6.2. *If $\mathcal{A}(G, G')$ is ireducible and aperiodic, with a stationary $\pi(G, G')$, then*

$$|\Theta^{(l)}_{(r)}(G, G')_{uu'} - \pi(G, G')\big(Rlv + v'\big)| \leq C\alpha^l$$

Proof. This proof has the same strategy to that of Theorem 6.1. The first step is to under the equation 6.14 in the large-depth limit.

$$\Theta^{(l)}_{(r)}(G, G')_{uu'} = \Theta^{(l)}_{(r-1)}(G, G')_{uu'}\dot{\Sigma}^{(l)}_{(r)}(G, G')_{uu'} + \Sigma^{(l)}_{(r)}(G, G')_{uu'}$$

According to the result of Theorem 6.1, we have already known,

$$\Sigma^{(l)}_{(r)}(G, G')_{uu'} = \pi(G, G')v + O(\alpha^l)$$

To proceed the proof, we need to work out the behavior of $\dot{\Sigma}^{(l)}_{(r)}(G, G')_{uu'}$ in the large depth.

(i) when $\phi(x) = \max\{0, x\}$

Recall that we define $c_{(r+1)} = f(c_{(r)})$, and we have,

$$f'(x) = \frac{1}{\pi}\arcsin(x) + \frac{1}{2}$$

Then, at the critical line,

$$\begin{aligned}\dot{\Sigma}_{(r)}(G, G')_{uu'} &= f'(c_r(G, G')) = \frac{1}{\pi}\arcsin(c_r(G, G')) + \frac{1}{2} \\ &= 1 - \frac{2}{\pi}(1 - c_r(G, G'))^{1/2} + O(1 - c_r(G, G'))^{3/2} \\ &= 1 + O(\alpha^{l/2})\end{aligned}$$

(ii) $\phi(x) = \tanh(x)$, we have

$$f'(x) = 1 - (x-1)f''(1) + O((x-1)^2)$$

At the critical line,

$$\dot{\Sigma}_{(r)}(G, G')_{uu'} = 1 + O(\alpha^l)$$

Now we prove the result by induction method. For $l = 0$, according to the definition and the result in Theorem 1,

$$\Theta^{(0)}_{(0)}(G, G')_{uu'} = \Sigma^{(0)}_{(0)}(G, G')_{uu'} \leq ||h^{(0)}_u||_2 ||h^{(0)}_{u'}||_2 = 1$$

Thus there is a constant C, depending on $G(V, E)$, $G'(V', E')$, and the number of MLP operation R, over feature initialization,

$$|\Theta^{(0)}_{(0)}(G, G')_{uu'} - \pi(G, G')v'| < C$$

Assume the result is valid for $\Theta^{(l)}_{(r)}(G, G')_{uu'}$, then we have,

$$|\Theta^{(l)}_{(r)}(G, G')_{uu'} - \pi(G, G')(Rlv + v')| \leq C\alpha^l$$

Now we consider the distance between $\Theta^{(l+1)}_{(r)}(G, G')_{uu'}$ and $\pi(G, G')(R(l+1)v + v')$. To prove this, we need to divide the propagation from l layer to $l+1$ layer into three parts:

(i) $\Theta^{(l)}_{(r)} \to \Theta^{(l)}_{(r+1)} \to \cdots \to \Theta^{(l)}_{(R)}$

(ii) $\Theta^{(l)}_{(R)} \to \Theta^{(l+1)}_{(0)}$

(iii) $\Theta^{(l+1)}_{(0)} \to \Theta^{(l+1)}_{(1)} \to \cdots \to \Theta^{(l+1)}_{(r)}$.

For (i),

$$\begin{aligned}
&|\Theta^{(l)}_{(r+1)}(G,G')_{uu'} - \pi(G,G')((lR+1)v+v')| \\
&= |\Sigma^{(l)}_{(r+1)}(G,G')_{uu'} + \dot{\Sigma}^{(l)}_{(r+1)}(G,G')_{uu'}\Theta^{(l)}_{(r)}(G,G')_{uu'} - \pi(G,G')((lR+1)v+v')| \\
&= |\pi(G,G')v(1+O(\alpha^l)) + (1+O(\alpha^{l/2}))\Theta^{(l)}_{(r)}(G,G')_{uu'} - \pi(G,G')((lR+1)v+v')| \\
&\leq C\alpha^l
\end{aligned}$$

Repeat the process, we have a relation for $\Theta^{(l)}_{(R)}(G,G')_{uu'}$ at the last step in (i),

$$|\Theta^{(l)}_{(R)}(G,G')_{uu'} - \pi(G,G')((lR+R-r)v+v')| \leq C\alpha^l.$$

Secondly, (ii) we go through an aggregation operation. Since it is can seen as a Markov chain step,

$$|\Theta^{(l+1)}_{(0)}(G,G')_{uu'} - \pi(G,G')((Rl+R-r)v+v')| \leq C\alpha^{l+1}.$$

(iii) Repeat the result in step (i), we have,

$$\begin{aligned}
&|\Theta^{(l)}_{(R)}(G,G')_{uu'} - \pi(G,G')((Rl+R)v+v')| \\
=&|\Theta^{(l)}_{(R)}(G,G')_{uu'} - \pi(G,G')(R(l+1)v+v'| \leq C\alpha^{l+1}.
\end{aligned}$$

□

Corollary 6.1 (Informal)**.** *Consider a GCN of the form (6.4) and (6.5), with depth L, non-linear transformations number R, an MSE loss, and a Lipchitz activation $\phi(x)$, trained with gradient descent on a node classification task. Then the output function follows,*

$$f_t(\mathcal{G}) = e^{-\eta\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G})t} f_0(\mathcal{G}) + (I - e^{-\eta\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G})t}\mathcal{Y})$$

where $\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G}) \in \mathbb{R}^{\sum_{i=1}^{N_G} n_i \times \sum_{i=1}^{N_G} n_i}$ *with the expression,*

$$\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G}) = \begin{pmatrix} \Theta^{(L)}_{(R)}(G_1,G_1) & \cdots & \Theta^{(L)}_{(R)}(G_1,G_N) \\ \vdots & \ddots & \vdots \\ \Theta^{(L)}_{(R)}(G_N,G_1) & \cdots & \Theta^{(L)}_{(R)}(G_N,G_N) \end{pmatrix}$$

There exists a positive integer L_0 *such that,* $\Theta^{(L)}_{(R)}(\mathcal{G},\mathcal{G})$ *is singular when* $L > L_0$*. Moreover, there exist a constant* $C > 0$ *such that for all* $t > 0$*,*

$$\|f_t(\mathcal{G}) - \mathcal{Y}\| > C$$

Proof. According to the result from [36], GNTK $\Theta^{(l)}_{(r)}(\mathcal{G},\mathcal{G})$ converges to a deterministic kernel and remains constant during gradient descent in the infinite width limit. We omit proof procedure for this result, since it is now a standard conclusion in the NTK study.

Based on the conclusion above, Lee et al. [37] proved that the infinitly-wide neural network is equivalent to its linearized mode,

$$f^{\text{lin}}_t(G) = f_0(G) + \nabla_\theta f_0(G)|_{\theta=\theta_0}\omega_t$$

where $\omega_t = \theta_t - \theta_0$. We call it linearized model because only zero and first order term of Taylor expansion are kept. Since we know dynamics of gradient flow using this linearized function are governed by,

$$\dot{\omega}_t = -\eta \nabla_\theta f_0(\mathcal{G})^T \nabla_{f^{\text{lin}}_t(\mathcal{G})}\mathcal{L}$$

$$\dot{f}^{\text{lin}}_t(\mathcal{G}) = -\eta \Theta^{(l)}_{(r)}(\mathcal{G},\mathcal{G}) \nabla_{f^{\text{lin}}_t(\mathcal{G})}\mathcal{L}$$

where $\mathcal{L}$ is an MSE loss, then the above equations have closed form solutions

$$f^{\text{lin}}_t(\mathcal{G}) = e^{-\eta\Theta^{(l)}_{(r)}(\mathcal{G},\mathcal{G})t} f_0(\mathcal{G}) + (I - e^{-\eta\Theta^{(l)}_{(r)}(\mathcal{G},\mathcal{G})t}\mathcal{Y})$$

Since Lee et al. [37] showed that $f^{\text{lin}}_t(\mathcal{G}) = f_t$ in the infinite width limit, thus we have,

$$f_t(\mathcal{G}) = e^{-\eta\Theta^{(l)}_{(r)}(\mathcal{G},\mathcal{G})t} f_0(\mathcal{G}) + (I - e^{-\eta\Theta^{(l)}_{(r)}(\mathcal{G},\mathcal{G})t}\mathcal{Y}) \tag{6.20}$$

In theorem 6.2, we have shown $\Theta^{(l)}_{(r)}(G, G')$ converges a constant matrix at an exponential rate when l and r are fixed. the GNTK $\hat{\Theta}^{(l)}_{(r)}(\mathcal{G}, \mathcal{G})$ can be written as,

$$\hat{\Theta}^{(l)}_{(r)}(\mathcal{G}, \mathcal{G}) = \begin{pmatrix} \Theta^{(l)}_{(r)}(G_1, G_1) & \Theta^{(l)}_{(r)}(G_1, G_2 & \cdots & \Theta^{(l)}_{(r)}(G_1, G_N) \\ & \vdots & & \\ \Theta^{(l)}_{(r)}(G_N, G_1) & \Theta^{(l)}_{(r)}(G_N, G_2 & \cdots & \Theta^{(l)}_{(r)}(G_N, G_N) \end{pmatrix}$$

According to equation 6.20, we know that,

$$||f_t(\mathcal{G}) - \mathcal{Y}|| = ||e^{-\eta\Theta^{(l)}_{(r)}(\mathcal{G},\mathcal{G})t}(f_0(\mathcal{G}) - \mathcal{Y})||$$

According to Theorem 6.2, there exists a l_0 such that, for any $l > l_0$ GNTK between any two graphs converges to a constant matrix,

$$\Theta^{(l)}_{(r)}(G_i, G_j) = \Theta(G_i, G_j)$$

Then the whole GNTK $\hat{\Theta}^{(l)}_{(r)}(\mathcal{G}, \mathcal{G})$ is singular. Let $\hat{\Theta}^{(l)}_{(r)}(\mathcal{G}, \mathcal{G}) = Q^T D Q$ be the decomposition of the GNTK, where Q is an orthogonal matrix and D is a diagonal matrix. Because $\hat{\Theta}^{(l)}_{(r)}(\mathcal{G}, \mathcal{G})$ is singular, D has at least one zero value $d_j = 0$, then

$$||f_t(\mathcal{G}) - \mathcal{Y}|| = ||Q^T(f_t(\mathcal{G}) - \mathcal{Y})Q|| \geq ||[Q^T(f_0(\mathcal{G}) - \mathcal{Y})Q]_j||$$

□

Theorem 6.3. *Consider a covariance matrix of GPK with the form (6.17) without non-linear transformation, i.e., $R = 0$. Then with a stationary vector $\tilde{\pi}(G, G')$ for $\tilde{\mathcal{A}}(G, G')$, such that*

$$|\Sigma^{(l)}_{(r)}(G, G')_{uu'} - \tilde{\pi}(G, G')\vec{\Sigma}^{(0)}(G, G')| \leq C\alpha^l$$

Furthermore, we denote the second largest eigenvalue of $\tilde{\mathcal{A}}(G, G')$ and $\tilde{\mathcal{A}}(G, G')$ as λ_2 and $\tilde{\lambda}_2$, respectively. Then,

$$\tilde{\lambda}_2 > \lambda_2$$

Proof. According to the aggregation function for covariance matrix, we have

$$\begin{aligned}\vec{\Sigma}^{(l)}(G,G') &= (1-\delta)\mathcal{A}(G,G')\vec{\Sigma}^{(l-1)}(G,G') + \delta\vec{\Sigma}^{(l-1)}(G,G') \\ &= ((1-\delta)\mathcal{A} + \delta I)\vec{\Sigma}^{(l-1)}(G,G') \\ &:= \tilde{\mathcal{A}}(G,G')\end{aligned}$$

Since new aggregation matrix $\tilde{\mathcal{A}}(G,G')$ is a stochastic matrix, which can seen from,

$$\sum_j \tilde{\mathcal{A}}(G,G')_{ij} = (1-\delta)\sum_j \mathcal{A}(G,G')_{ij} + \delta\sum_j I_{ij} = 1$$

The new aggregation can slow down the convergence rate can seen from considering their eigenvalues. Suppose the eigenvalues of original matrix $\mathcal{A}(G,G')$ are $\{\lambda_1 > \lambda_2 > \cdots > \lambda_{nn'}\}$. We already know that the maximum eigenvalue is $\lambda_1 = 1$, and the converge speed is governed by the second largest eigenvalue λ_2. Now we consider the second largest eigenvalue $\tilde{\lambda}_2$ of matrix $\tilde{\mathcal{A}}$:

$$\tilde{\lambda}_2 = (1-\delta)\lambda_2 + \delta = \lambda_2 + \delta(1-\lambda_2) > \lambda_2$$

Since $\lambda_2 < \tilde{\lambda}_2 < 1$, the convergence speed will be slow down, through it is still exponential convergent.

For the limit behavior of $\Sigma^{(l)}_{(r)}(G,G')_{uu'}$ as l tends to infinity, we can directly use the proof strategy from Theorem 1 and 2. □

Theorem 6.4. *Consider a GCN of the form (6.4) and (6.18). If $\mathcal{A}(G,G')$ is irreducible and aperiodic, with stationary $\pi(G,G')$, then*

$$|\Theta^{(l)}_{(r)}(G,G')_{uu'} - \pi(G,G')\big(Rl(1+\frac{\sigma_w^2}{2})^{Rl}v + v'\big)| \le C\alpha^l$$

Proof. According to the signal propagation equation 6.18. We have,

$$q_r(G) = q_{r-1}(G) + \frac{\sigma_w^2}{2}q_{r-1}(G) = (1+\frac{\sigma_w^2}{2})q_{r-1}(G)$$

Since $\sigma_w^2 > 0$, $q_r(G)$ grows at an exponential rate.

Now, we turn to compute the correlation term $c_r(G, G')$. For convenience, we suppose $q_r(G) = q_r(G')$. Then,

$$\begin{aligned} c_{r+1}(G, G') &= \frac{\Sigma_{r+1}(G, G')}{q_{r+1}(G)} \\ &= \frac{1}{1+\frac{\sigma_w^2}{2}} \frac{\Sigma_r(G, G')}{q_r(G)} + \frac{1}{1+\frac{\sigma_w^2}{2}} \frac{\sigma_w^2}{2} f(c_r(G, G')) \\ &= \frac{1}{1+\frac{\sigma_w^2}{2}} c_r(G, G') + \frac{\frac{\sigma_w^2}{2}}{1+\frac{\sigma_w^2}{2}} f(c_r(G, G')) \end{aligned}$$

Using Taylor expansion of f near 1, as have been done in the proof of Lemma 2,

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

we have,

$$c_{r+1}(G, G') = c_r(G, G') + \frac{2\sqrt{2}}{3\pi} \frac{\frac{\sigma_w^2}{2}}{1+\frac{\sigma_w^2}{2}} \big[(1 - c_r(G, G')^{3/2} + O((1 - c_r(G, G')^{5/2}))\big]$$

Note that it is similar to the case of MLP without residual connection:

$$c_{r+1}(G, G') = c_r(G, G') + \frac{2\sqrt{2}}{3\pi} \big[(1 - c_r(G, G')^{3/2} + O((1 - c_r(G, G')^{5/2}))\big]$$

Following the proof diagram in Theorem 1 and 2, we can obtain the behavior of $\Sigma_{(r)}^{(l)}(G, G')_{uu'}$ and $\Theta_{(r)}^{(l)}(G, G')_{uu'}$ in the large depth limit,

$$|\Sigma_{(r)}^{(l)}(G, G')_{uu'} - \pi(G, G')\big((1 + \frac{\sigma_w^2}{2})^{Rl} v\big)| \leq C\alpha^l$$

$$|\Theta_{(r)}^{(l)}(G, G')_{uu'} - \pi(G, G')\big(Rl(1 + \frac{\sigma_w^2}{2})^{Rl} v\big)| \leq C\alpha^l$$

$\square$

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In Chapter 3, we studied the expressive power of deep dropout networks under the mean-filed theory. Both information propagation with feed-forward and gradient back-propagation were studied in the infinite width limit. In particular, we characterized the back-propagation process without gradient independence assumption and found that both gradients metric with a single input and a pair of inputs are determined by a same quantity. Furthermore, a better empirical formula that can describe the trainable length for deep dropout networks was figured out through performing experiments on the finite-width but wide networks with gradient descent.

In Chapter 4, we investigated the optimization property of orthogonally-initialized networks through the neural tangent kernel. In the infinite wide limit, the NTK of networks with orthogonal initialization converges to the same deterministic kernel of a Gaussian initialized network both before training and under gradient descent training. With this result, we found the dynamics including training loss and training accuracy of orthogonally initialized networks behave similarly to those of networks with Gaussian initialization. We further confirm this conclusion by conducting experiments on finite-width networks with orthogonal and Gaussian initialization.

In Chapter 5, we worked on the optimization and generalization of deep linear networks for binary classification in the large learning rate regime. With help of NTK, we derived a formula that can describe the dynamics of both linear predictor and linear network with wide width. We showed that there is a catapult phase for deep linear networks with logistic loss. In this phase, the gradient descent can find a flatter minimum than that can be found in small learning rate phase. With empirical evidence, we found that best generalization performance can be obtained in the catapult phase. Besides, the best performance is typically achieved with learning rate annealing strategy.

In Chapter 6, we characterized the expressivity and trainability of deep and wide graph convolutional networks (GCNs). Under the mean-field theory and NTK theory, we found that both expressivity and trainability described by Gaussian process kernel and Graph neural tangent kernel respectively, drop at an exponential rate, in the infinite-width limit. In addition, we extend our theoretical framework to the techniques that can deepen GCNs, such as DropEdge and residual connections. Even though, these techniques can mitigate the exponential decay problem, they can solve it in essence. We use several experiments to support our theoretical conclusions.

7.2 Future Work

In addition to the study on the ultra-wide networks (Chapters 3, 4, 5 and 6) in this thesis, there are several directions where we can make breakthroughs in the future, which are summarized as follows,

i. For learning setting discussed in Chapters 3, 4, and 5 that deal with the optimization property of ultra-wide networks focus on the supervised learning setting which is too basic with limited application scenarios. On the contrary, unsupervised learning can produce richer applications. However, it is a difficult thing to study its optimal properties. Therefore, it would be meaningful to apply NTK technology to unsupervised learning.

ii. While the work introduced on Chapter 5 concentrates on the large learning rate for linear networks in the binary classification, there are several remaining open questions. For non-linear networks, the effect of a large learning rate is not clear in theory. In addition, the stochastic gradient descent algorithm also needs to be explored when the learning rate is large. We leave these unsolved problems for future work.

iii. In Chapter 6, we show that deep GCN suffer from the exponential decay problem on both expressivity and trainability at large depth. The fundamental reason for this problem is that the matrix describing the information transmitted by the neighbor nodes of the graph network is a probability transition matrix. How to solve the problem of rapid loss of information is currently the most core problem in the graph network. It is a potential direction to use percolation model to try to solve this problem.

Bibliography

[1] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[2] Simon Haykin. *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[6] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[9] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[10] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034, 2017.

[12] Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. *arXiv preprint arXiv:1902.07153*, 2019.

[13] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[14] Hongyang Gao and Shuiwang Ji. Graph u-nets. *arXiv preprint arXiv:1905.05178*, 2019.

[15] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212*, 2017.

[16] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. Self-attention graph pooling. *arXiv preprint arXiv:1904.08082*, 2019.

[17] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*, pages 4800–4810, 2018.

[18] Hao Yuan and S. Ji. Structpool: Structured graph pooling via conditional random fields. In *ICLR*, 2020.

[19] Thomas N Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

[20] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[21] Balázs Csanád Csáji et al. Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, 24(48):7, 2001.

[22] 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.

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

[24] 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.

[25] Ge Yang and Samuel Schoenholz. Mean field residual networks: On the edge

of chaos. In *Advances in neural information processing systems*, pages 7103–7114, 2017.

[26] 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.

[27] 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.

[28] Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S Schoenholz. A mean field theory of batch normalization. *arXiv preprint arXiv:1902.08129*, 2019.

[29] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. *arXiv preprint arXiv:1902.06853*, 2019.

[30] Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[31] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[32] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. 1999.

[33] Jilin Hu, Jianbing Shen, Bin Yang, and Ling Shao. Infinitely wide graph convolutional networks: semi-supervised learning via gaussian processes. *arXiv preprint arXiv:2002.12168*, 2020.

[34] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[35] Dar Gilboa, Bo Chang, Minmin Chen, Greg Yang, Samuel S Schoenholz, Ed H Chi, and Jeffrey Pennington. Dynamical isometry and a mean field theory of LSTMs and GRUs. *arXiv preprint arXiv:1901.08987*, 2019.

[36] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8580–8589, 2018.

[37] Jaehoon Lee, Lechao Xiao, Samuel S Schoenholz, Yasaman Bahri, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

[38] 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.

[39] 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.

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

[41] 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.

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

[43] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. corr. *arXiv preprint arXiv:1811.08888*, pages 9–77, 2018.

[44] Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Advances in Neural Information Processing Systems*, pages 10836–10846, 2019.

[45] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019.

[46] Wei Huang, Weitao Du, and Richard Yi Da Xu. On the neural tangent kernel of deep networks with orthogonal initialization. *arXiv preprint arXiv:2004.05867*, 2020.

[47] Simon S Du, Kangcheng Hou, Russ R Salakhutdinov, Barnabas Poczos, Ruosong Wang, and Keyulu Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. In *Advances in Neural Information Processing Systems*, pages 5723–5733, 2019.

[48] Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: Nngp and ntk for deep attention networks. *arXiv preprint arXiv:2006.10540*, 2020.

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

[50] Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848*, 2020.

[51] Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. *arXiv preprint arXiv:1909.05989*, 2019.

[52] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[53] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. *arXiv preprint arXiv:1802.08246*, 2018.

[54] Ziwei Ji and Matus Telgarsky. The implicit bias of gradient descent on non-separable data. In *Conference on Learning Theory*, pages 1772–1798, 2019.

[55] Alnur Ali, Edgar Dobriban, and Ryan J Tibshirani. The implicit regularization of stochastic gradient flow for least squares. *arXiv preprint arXiv:2003.07802*, 2020.

[56] Mor Shpigel Nacson, Jason D Lee, Suriya Gunasekar, Pedro HP Savarese, Nathan Srebro, and Daniel Soudry. Convergence of gradient descent on separable data. *arXiv preprint arXiv:1803.01905*, 2018.

[57] Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pages 9461–9471, 2018.

[58] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. *arXiv preprint arXiv:1810.02032*, 2018.

[59] Adityanarayanan Radhakrishnan, Eshaan Nichani, Daniel Bernstein, and Caroline Uhler. Balancedness and alignment are unlikely in linear neural networks. *arXiv preprint arXiv:2003.06340*, 2020.

[60] Mor Shpigel Nacson, Suriya Gunasekar, Jason D Lee, Nathan Srebro, and Daniel Soudry. Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. *arXiv preprint arXiv:1905.07325*, 2019.

[61] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

[62] Ziwei Ji and Matus Telgarsky. Directional convergence and alignment in deep learning. *arXiv preprint arXiv:2006.06657*, 2020.

[63] L. Chizat and F. Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. *arXiv preprint arXiv:2002.04486*, 2020.

[64] Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? *arXiv preprint arXiv:1812.10004*, 2018.

[65] Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms. *arXiv preprint arXiv:2005.06398*, 2020.

[66] 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.

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

[68] 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.

[69] 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.

[70] Wei Huang, Richard Yi Da Xu, Weitao Du, Yutian Zeng, and Yunce Zhao. Mean field theory for deep dropout networks: digging up gradient backpropagation deeply. *arXiv preprint arXiv:1912.09132*, 2019.

[71] 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.

[72] Jeffrey Pennington, Samuel S Schoenholz, and Surya Ganguli. The emergence of spectral universality in deep networks. *arXiv preprint arXiv:1802.09979*, 2018.

[73] 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.

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

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

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

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] 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.

[82] 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.

[83] Stefano Favaro, Sandra Fortini, and Stefano Peluchetti. Stable behaviour of infinitely wide deep neural networks. *arXiv preprint arXiv:2003.00394*, 2020.

[84] Sourav Chatterjee and Elizabeth Meckes. Multivariate normal approximation using exchangeable pairs. *arXiv preprint math/0701464*, 2007.

[85] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in neural information processing systems*, pages 3036–3046, 2018.

[86] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*, 2019.

[87] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

[88] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[89] Zeyuan Allen-Zhu and Yuanzhi Li. What can resnet learn efficiently, going beyond kernels? In *Advances in Neural Information Processing Systems*, pages 9017–9028, 2019.

[90] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

[91] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

[92] Atsushi Nitanda, Geoffrey Chinot, and Taiji Suzuki. Gradient descent can learn less over-parameterized two-layer neural networks on classification problems. *arXiv preprint arXiv:1905.09870*, 2019.

[93] Geir K Nilsen, Antonella Z Munthe-Kaas, Hans J Skaug, and Morten Brun. Efficient computation of hessian matrices in tensorflow. *arXiv preprint arXiv:1905.05559*, 2019.

[94] Sébastien Bubeck. Convex optimization: Algorithms and complexity. *arXiv preprint arXiv:1405.4980*, 2014.

[95] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. *arXiv preprint arXiv:1801.07606*, 2018.

[96] Uri Alon and Eran Yahav. On the bottleneck of graph neural networks and its practical implications. *arXiv preprint arXiv:2006.05205*, 2020.

[97] Kenta Oono and Taiji Suzuki. Graph neural networks exponentially lose expressive power for node classification. *arXiv preprint arXiv:1905.10947*, 2019.

[98] Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 338–348, 2020.

[99] Hanqing Zeng, Muhan Zhang, Yinglong Xia, Ajitesh Srivastava, Rajgopal Kannan, Viktor Prasanna, Long Jin, Andrey Malevich, and Ren Chen. Deep graph neural networks with shallow subgraph samplers. *ICLR*, 2020.

[100] Kaixiong Zhou, Xiao Huang, Yuening Li, Daochen Zha, Rui Chen, and Xia Hu. Towards deeper graph neural networks with differentiable group normalization. *arXiv preprint arXiv:2006.06972*, 2020.

[101] Guohao Li, Chenxin Xiong, Ali Thabet, and Bernard Ghanem. Deepergcn: All you need to train deeper gcns. *arXiv preprint arXiv:2006.07739*, 2020.

[102] Ming Chen, Zhewei Wei, Zengfeng Huang, Bolin Ding, and Yaliang Li. Simple and deep graph convolutional networks. In *International Conference on Machine Learning*, pages 1725–1735. PMLR, 2020.

[103] Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer, 1996.

[104] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. Mean-field behaviour of neural tangent kernel for deep neural networks. *arXiv preprint arXiv:1905.13654*, 2019.

[105] Lechao Xiao, Jeffrey Pennington, and Samuel S Schoenholz. Disentangling trainability and generalization in deep learning. *arXiv preprint arXiv:1912.13053*, 2019.

[106] James R Norris and James Robert Norris. *Markov chains*. Number 2. Cambridge university press, 1998.

[107] Paul A Gagniuc. *Markov chains: from theory to implementation and experimentation*. John Wiley & Sons, 2017.

[108] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. *arXiv preprint arXiv:1810.05997*, 2018.

[109] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2019.

[110] Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9267–9276, 2019.

[111] Ari Freedman. Convergence theorem for finite markov chains. *Proc. REU*, 2017.