Theoretical Investigation of Generalization Bound for Residual Networks

Hao Chen[1,*], **Zhanfeng Mo**[1,*], **Zhouwang Yang**[1,†] and **Xiao Wang**[2]
[1]University of Science and Technology of China
[2]Purdue University
{ch330822, oscarmzf}@mail.ustc.edu.com, yangzw@ustc.edu.cn, wangxiao@purdue.edu

Abstract

This paper presents a framework for norm-based capacity control with respect to a Weight-Normalized Residual Neural Networks (ResNets). We first formulate the representation of each residual block. For the regression problem, we analyze the Rademacher Complexity of the ResNets family and establish a tighter generalization upper bound for Weight-Normalized ResNets. Using the $\ell_{p,q}$-norm weight normalization in which $1/p+1/q \geqslant 1$, we discuss the properties of a width-independent capacity control, which only relies on the depth according to a square root term. Several comparisons suggest that our result is tighter than previous work. Parallel results for Deep Neural Networks (DNN) and Convolutional Neural Networks (CNN) are included by introducing the $\ell_{p,q}$-norm weight normalization for DNN and the $\ell_{p,q}$-norm kernel normalization for CNN. Numerical experiments also verify that ResNet structures contribute to better generalization properties.

1 Introduction

Deep neural networks have been applied to many fields, including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, and more. Since ResNets have been introduced to improve imagin recognition [He *et al.*, 2016], [Goodfellow *et al.*, 2016], ResNets have taken the deep learning world by storm. Though the robustness and efficiency of ResNets has proven to be enormously successful in various artificial intelligence tasks [Huang *et al.*, 2017], [Srivastava *et al.*, 2015], [Fan *et al.*, 2018], we aim to examine the capacity of ResNets theoretically. Previous studies have investigated the capacity of several simple network structures, such as unregularized DNN, DNN without bias or DNN with ReLU active functions [Bartlett *et al.*, 2017], [Golowich *et al.*, 2018a], [Neyshabur *et al.*, 2017], [Neyshabur *et al.*, 2015], [Sun *et al.*, 2016], [Shalev-Shwartz and Ben-David, 2014] . ResNets and other powerful structures, however, require further explorations in literature.

Intuitively, great robustness should be compatible with relatively small generalization bounds. Inspired by the amazing performance of ResNets in regression problems, we hope to answer a central question : can we find a tighter generalization bound for ResNets and other developed structures ?

In this paper, we focus on providing a tighter generalization bound with respect to $\ell_{p,q}$-norm for ResNets. Since most prior studies are restricted to simple DNN structures [Golowich *et al.*, 2018b], [Bartlett *et al.*, 2017], [Neyshabur *et al.*, 2017], [Sun *et al.*, 2016], we take the layer-wise weight normalization and hidden layers with bias into consideration to pursue a more general sight. After showing comparisions with some related works , further discussions extend the parallel conclusions to DNN cases.

We first formulate the function class, which is derived from the combination of Residual-Blocks, and define the $\ell_{p,q}$-weight normalization. After providing an upper bound for the Rademacher Complexity for the ResNets function class, we induce the generalization bound with respect to regression problems. In particular, we obtain a width-free generalization bound with $1/p + 1/q \geqslant 1$; therefore, several prior conclusions under $\ell_{2,2}$-norm, $\ell_{1,\infty}$-norm are covered. By utilizing the same method, we extend the width-free generalization bound to DNN and Convolutional-ResNets nontrivialy.

The contributions of this paper are summarized as follows : 1. We obtain a tighter generalization bound for ResNets with layer-wise $\ell_{p,q}$-normalization and bias; 2. We justify the tightness by comparing the results with related works; 3. We extend the parallel conclusions to structures, including DNN and Convolutional-ResNets.

In Section 2, we introduce some notations as well as the formulation of the ResNets function class. To prepare for the main result, we provide the upper bound of the Rademacher Complexity for the ResNets function class. Section 4 discusses the generalization bound for ResNets and several comparisions. We explore the parallel conclusions of DNN and Convolutonal-ResNets in Section 5 and Section 6, respectively. In Section 7, numerical experiments are explained to verify the previous theoretical results. The proof details are concluded in the supplementary materials.[Mo and Chen, 2019]

[*]These authors have contributed equally to this work. The order of name is alphabetical.

[†]Corresponding Author.

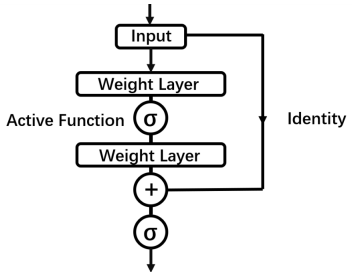

Figure 1: A Residual-Block

2 Preliminary

Traditionally, ResNets refer to networks with forms similar to Figure 1: By adding bias to each hidden layers, we obtain a representation of the mapping of a single residual-block as :

$$\begin{aligned}&\mathcal{F}_i(\mathbf{x}_{i-1})\\ =&\sigma\left[\mathbf{W}_{i,2}\cdot\sigma\left[(\mathbf{W}_{i,1}\cdot\mathbf{x}_{i-1}+\mathbf{b}_{i,1})\right]+\mathbf{b}_{i,2}+\mathbf{I}_i\cdot\mathbf{x}_{i-1}\right],\end{aligned}$$

$\mathcal{F}_i:\mathbb{R}^{d_{i-1,2}}\to\mathbb{R}^{d_{i,2}}$. The bias is forced to be commensurate with the weight matrix in terms of dimension. In order to simplify $\mathcal{F}_i$, we rewrite the function as:

$$\begin{aligned}&\widetilde{\mathcal{F}}_i(\begin{pmatrix}\mathbf{x}_{i-1}\\1\end{pmatrix})\\ =&\sigma\left[\begin{pmatrix}\mathbf{W}_{i,2}&\mathbf{b}_{i,2}\\\mathbf{0}&1\end{pmatrix}\cdot\sigma\left[\begin{pmatrix}\mathbf{W}_{i,1}&\mathbf{b}_{i,1}\\\mathbf{0}&1\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}_{i-1}\\1\end{pmatrix}\right]\right.\\ &\left.+\mathbf{I}_i\cdot\mathbf{x}_{i-1}\right]\\ =&\sigma\left[\begin{pmatrix}\mathbf{W}_{i,2}&\mathbf{b}_{i,2}\\\mathbf{0}&1\end{pmatrix}\cdot\sigma\left[\begin{pmatrix}\mathbf{W}_{i,1}&\mathbf{b}_{i,1}\\\mathbf{0}&1\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}_{i-1}\\1\end{pmatrix}\right]\right.\\ &\left.+\begin{pmatrix}\mathbf{I}_{i\times i}&\mathbf{0}\\\mathbf{0}&0\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}_{i-1}\\1\end{pmatrix}\right].\end{aligned}$$

By denoting every $\widetilde{\mathbf{x}}_i\triangleq\begin{pmatrix}\mathbf{x}_i\\1\end{pmatrix}$, $\mathbf{M}_{i,j}\triangleq(\mathbf{W}_{i,j},\mathbf{b}_{i,j})$, $\widetilde{\mathbf{M}}_{i,j}^T\triangleq(\mathbf{M}_{i,j}^T,\mathbf{e}_{d_i}^T)^T,\forall i=1,...,k$ and $j=1,2$, $\mathbf{e}_{d_i}^T$ the d_ith natural basis, we rewrite $\widetilde{\mathcal{F}}_i:\mathbb{R}^{d_{i-1,2}+1}\to\mathbb{R}^{d_{i,2}+1}$ as:

$$\begin{aligned}&\widetilde{\mathcal{F}}_i(\widetilde{\mathbf{x}}_{i-1})\\ =&\sigma\left[\begin{pmatrix}\mathbf{W}_{i,2}&\mathbf{b}_{i,2}\\\mathbf{0}&1\end{pmatrix}\cdot\sigma\left[\begin{pmatrix}\mathbf{W}_{i,1}&\mathbf{b}_{i,1}\\\mathbf{0}&1\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}_{i-1}\\1\end{pmatrix}\right]\right.\\ &\left.+\begin{pmatrix}\mathbf{I}_{i\times i}&\mathbf{0}\\\mathbf{0}&0\end{pmatrix}\cdot\begin{pmatrix}\mathbf{x}_{i-1}\\1\end{pmatrix}\right]\\ \triangleq&\sigma\left[\widetilde{\mathbf{M}}_{i,2}\cdot\sigma\left[\widetilde{\mathbf{M}}_{i,1}\cdot\widetilde{\mathbf{x}}_{i-1}\right]+\mathbf{I}_i\cdot\widetilde{\mathbf{x}}_{i-1}\right].\end{aligned}$$

The introduced notations are unambiguous:
(1) $\mathbf{M}_{i,2},\mathbf{M}_{i,1}$: $\mathbf{M}_{i,2}\in\mathbb{R}^{d_{i,2}\times d_{i,1}},\mathbf{M}_{i,1}\in\mathbb{R}^{d_{i,1}\times d_{i-1,2}}$.
(2) $\mathbf{I}_i$: $\mathbf{I}_i\in\mathbb{R}^{d_{i,2}\times d_{i-1,2}}$, instead of the Identity matrix, $\mathbf{I}_i$ is introduced to ensure that $\mathbf{I}_i\cdot\widetilde{\mathbf{x}}_{i-1}$ is commensurate with $\mathbf{M}_{i,2}\cdot\sigma\left[\mathbf{M}_{i,1}\cdot\widetilde{\mathbf{x}}_{i-1}\right]$ in terms of dimension.
(3) $\sigma[\cdot]$: A ρ-lipschitz continuous active function.
(4) $\ell_{p,q}$-norm: The $\ell_{p,q}$-norm of a $n\times m$ matrix $\mathbf{A}$ is defined as $\|\mathbf{A}\|_{p,q}=(\sum_{i=1}^{n}(\sum_{j=1}^{m}|a_{ij}|^p)^{\frac{q}{p}})^{\frac{1}{q}}$

We denote the last weight layer as $\tilde{\mathcal{F}}_{k+1}:\mathbb{R}^{d_{k,2}+1}\to\mathbb{R}^{d_{k+1}+1}$ and the corresponding matrix is defined as $\mathbf{M}_{k+1}\in\mathbb{R}^{d_{k+1}\times(d_{k,2}+1)}$, where $\mathbf{M}_{k+1}\triangleq(\mathbf{W}_{k+1},\mathbf{b}_{k+1})$. With $\ell_{p,q}$-norm, the weight matrixes $\{\mathbf{M}_{i,1},\mathbf{M}_{i,2}\}_{i=1}^k$, $\mathbf{M}_{k+1}$ are constrained by $\|\mathbf{M}_{i,2}\|\leqslant c_{i,2},\|\mathbf{M}_{i,1}\|_{p,q}\leqslant c_{i,1},\quad\forall i=1,\ldots,k;\quad\|\mathbf{M}_{k+1}\|_{p,q}\leqslant c_{k+1}$. The discussion above induces a function class called the Weight-Normalized-Residual-Network class, which contains elements that can be represented by combining residual-blocks.

Definition 2.1. *$\mathcal{RN}_{p,q,\mathbf{c}}^{k,\mathbf{d}}$ This definition depicts a class of functions that can be represented by the previously mentioned WN-RN structure.*
p,q: The norm is set as $\ell_{p,q}$-norm
$\mathbf{d}$: $\mathbf{d}\triangleq(d_{k+1},d_{k,2},d_{k,1}\ldots,d_{1,2},d_{1,1},d_0)$ is a vector with width information. We denote $d_{i,1},d_{i,2}$ as the width of the 1st and 2nd layer of the ith residual block, respectively.
$\mathbf{c}$: $\mathbf{c}\triangleq(c_{k+1},c_{k,2},c_{k,1},...,c_{1,2},c_{1,1})$, where $\|\mathbf{M}_{k+1}\|_{p,q}\leqslant c_{k+1},\|\mathbf{M}_{i,2}\|_{p,q}\leqslant c_{i,2},\|\mathbf{M}_{i,1}\|_{p,q}\leqslant c_{i,1},\quad\forall i=1,\ldots,k;$ $f\in\mathcal{RN}_{p,q,\mathbf{c}}^{k,\mathbf{d}}\Leftrightarrow\exists\quad\{\{\mathbf{M}_{i,1},\mathbf{M}_{i,2}\}_{i=1}^k,\mathbf{M}_{k+1}\},\quad s.t.\quad f=\widetilde{\mathcal{F}}_{k+1}\circ\widetilde{\mathcal{F}}_k\circ\cdots\circ\widetilde{\mathcal{F}}_1$.

By definition, the following theorem is obvious:

Theorem 2.2. *The property of $\mathcal{RN}_{p,q,\mathbf{c}}^{k,\mathbf{d}}$.*
Given $p,q,\mathbf{c}^{(1)},\mathbf{c}^{(2)},k_1,k_2$, and vectors $\mathbf{d}^{(1)},\mathbf{d}^{(2)}$ with $k_1\leqslant k_2,\mathbf{d}^{(1)}\leqslant\mathbf{d}^{(2)},\mathbf{c}^{(1)}\leqslant\mathbf{c}^{(2)}$ w.r.t, the components precede the k_{1th} component:
(1): $\mathcal{RN}_{p,q,\mathbf{c}^{(1)}}^{k_1,\mathbf{d}^{(1)}}\subset\mathcal{RN}_{p,q,\mathbf{c}^{(2)}}^{k_1,\mathbf{d}^{(1)}}$
(2): $\mathcal{RN}_{p,q,\mathbf{c}^{(1)}}^{k_1,\mathbf{d}^{(1)}}\subset\mathcal{RN}_{p,q,\mathbf{c}^{(1)}}^{k_2,\mathbf{d}^{(2)}}$

Intuitively, a shallow and thin ResNet can be represented by a relative deeper and wider one.

3 The Estimation of the Rademacher Complexity for $\mathcal{RN}_{p,q,\mathbf{c}}^{n,\mathbf{d}}$

In this section, we provide an upper bound for the Rademacher Complexity of $\mathcal{RN}_{p,q,\mathbf{c}}^{k,\mathbf{d}}$.

Definition 3.1. *Rademacher Compexity of a function class. Assume that f is a real value function, ϵ_i is the Rademacher Random Variable:*
(1) Empirical Rademacher Compexity:

$$\hat{\mathfrak{R}}_S(\mathcal{N})\triangleq\mathbb{E}_\epsilon\left[\sup_{f\in\mathcal{N}}\frac{1}{n}\sum_{i=1}^{n}\epsilon_i f(\mathbf{z}_i)\right]$$

(2) Rademacher Compexity:

$$\mathfrak{R}_n(\mathcal{N})\triangleq\mathbb{E}_{S\sim\mathcal{D}^n}\left[\hat{\mathfrak{R}}_S(\mathcal{N})\right]$$

Without a loss of generality, we consider the input space as $\mathcal{X}\triangleq\{\mathbf{x}\in\mathbb{R}^{d_0}|\ \|\mathbf{x}\|_{p^*}\leqslant 1\}$. Moreover, we assume $d_{k+1}=1$, so that the Rademacher Complexity is well defined.

Theorem 3.2. *The Rademacher Complexity of $\mathcal{RN}_{p,q,\mathbf{c}}^{k,\mathbf{d}}$.*
If $p\geqslant 1$, $q\geqslant 1$, $k\geqslant 0$ and $\mathbf{c}>\mathbf{0}$, ρ is the lipschitz constant of the active function; $d_{i,j}\in\mathbb{N}^+$, $\forall i=1,\ldots,k$, $j=1,2$.

Then, for every set of $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathcal{X}$, we obtain:

$$\begin{aligned}&\hat{\mathfrak{R}}_{\mathcal{S}}(\mathcal{RN}^{k,\mathbf{d}}_{p,q,\mathbf{c}})\\ \leqslant& c_{k+1}\rho d_{k+1}^{[\frac{1}{p^*}-\frac{1}{q}]_+}\left(\sqrt{\frac{(6k+4)\log 2}{n}}s_k\right.\\ &\left.+\prod_{i=1}^{k}\left(c_{i,2}c_{i,1}\rho^2(d_{i,1}d_{i,2})^{[\frac{1}{p^*}-\frac{1}{q}]_+}+1\right)d_0^{\frac{1}{p^*}}\sqrt{\frac{C(p)}{n}}\right),\end{aligned}$$

where $s_0 = d_0^{\frac{1}{p^*}}$,

$$\begin{aligned}s_i =&(c_{i,2}c_{i,1}\rho^2(d_{i,1}d_{i,2})^{[\frac{1}{p^*}-\frac{1}{q}]_+}+1)s_{i-1}\\ &+c_{i,2}c_{i,1}\rho^2(d_{i,1}d_{i,2})^{[\frac{1}{p^*}-\frac{1}{q}]_+}+c_{i,2}\rho d_{i,2}^{[\frac{1}{p^*}-\frac{1}{q}]_+}\\ &\text{for } i=1,2,\cdots,k,\end{aligned}$$

$$C(p)\triangleq\begin{cases}2\log(2d_0) & p=1\,,\\ \min(p^*-1,2\log(2d_0)) & p>1\,.\end{cases}$$

We prepare this theorem for the main result. According to [Mohri *et al.*, 2012], the generalization bound can be deduced from the upper bound of the Rademacher Complexity.

4 The Generalization Bound for Weight-Normalized-ResNets

In this section, we provide an estimation for the generalization bound in the data regression problem. With $\ell_{p,q}$-norm satisfying $\frac{1}{p}+\frac{1}{q}\geqslant 1$, the estimation has a more laconic form.

The Regression Problem

$\mathcal{S}\triangleq\{(\mathbf{x}_1,y_1),\ldots(\mathbf{x}_n,y_n)\}\sim\mathcal{D}$ are i.i.d samples in $\mathcal{X}\times\mathcal{Y}\subset\mathbb{R}^{d_0+1}$. We are interested in finding $h:\mathcal{X}\to\mathcal{Y}$ in order to satisfy $y_i=h(\mathbf{x}_i)+\varepsilon_i,\ \forall i=1,\ldots,n$, where ε_i represents independent noise. To evaluate the choice of h, we introduce the loss function class :

Definition 4.1. *Loss Function Class $\mathcal{RG}^{k,\mathbf{d}}_{p,q,\mathbf{c}}$:*

$$\mathcal{RG}^{n,\mathbf{d}}_{p,q,\mathbf{c}}\triangleq\{g(f(\mathbf{x}),y)\in\mathbb{R}|\mathbf{x}\in\mathcal{X},y\in\mathcal{Y},f\in\mathcal{RN}^{k,\mathbf{d}}_{p,q,\mathbf{c}}\}$$

If we assume that $\forall g\in\mathcal{RG}^{k,\mathbf{d}}_{p,q,\mathbf{c}}:\mathcal{Y}\times\mathcal{Y}\to[0,1]$ is a γ-Lipschitz function, the Loss Function Class is denoted as $\mathcal{RG}_\gamma\triangleq\{g$ is γ-Lipschitz$|\ g\in\mathcal{RG}^{k,\mathbf{d}}_{p,q,\mathbf{c}}\}$.

By definition 4.1, since the first variable of a loss function $g(f(\mathbf{x}),y)$, that is, $f(\mathbf{x})$, belongs to ResNets, connection matrixes of g are the same as the ones of $f\in\mathcal{RN}^{k,\mathbf{d}}_{p,q,\mathbf{c}}$. Hence, we omit the repetitious discussions. In fact, for generalization bound, the only contribution from loss function class is its Lipschitz-constant: γ. We discuss the choice of γ after the main results.

Definition 4.2. *Generalization Error:*
If $\forall g\in\mathcal{RG}_\gamma$, the true risk and the empirical risk is defined as:

$$\text{The true risk:}\quad \mathbb{E}_{\mathcal{D}}[g]\triangleq\mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[g(\mathbf{x},y)\right],$$

$$\text{The empirical risk:}\quad \hat{\mathbb{E}}_{\mathcal{S}}[g]\triangleq\frac{1}{n}\sum_{i=1}^{n}g(f(\mathbf{x}_i),y_i).$$

The generalization error is the deviation between the expected and the empirical error:

$$\left|\mathbb{E}_{\mathcal{D}}[g]-\hat{\mathbb{E}}_{\mathcal{S}}[g]\right|\quad\forall g\in\mathcal{RG}_\gamma.$$

Generalization bound is an upper bound for generalization error. Intuitively, high generalization error indicates that the model is overfitting. Hence, generalization bound guarantees generalization error to be lower than a controllable level. In other words, generalization bound provides us with an explicit guidance to avoid overfitting. A tight generalization bound promises that a network can fit the unseen data well.

4.1 Main Results

Theorem 4.3. *Estimation of Generalization Bound:*
We assume that $\mathbf{z}\triangleq(\mathbf{x},y)\sim\mathcal{D}$, $\mathcal{S}\triangleq\{\mathbf{z}_1,\ldots,\mathbf{z}_n\}$ is a dataset with n i.i.d samples selected from the distribution $\mathcal{D}$. Then, we fix $\delta\in(0,1),\forall k\in\mathbb{N}^+,\forall d_{i,j}\in\mathbb{N}^+\ \ i=1,\ldots,k,\ j=1,2$. With probability of at least $1-\delta$ over the generation of $\mathcal{S}$, it holds that:

$$\begin{aligned}&\sup_{g\in\mathcal{RG}_\gamma}\left|\mathbb{E}_{\mathcal{D}}[g]-\hat{\mathbb{E}}_{\mathcal{S}}[g]\right|\\ \leqslant&\sqrt{\frac{\log(\frac{1}{\delta})}{2n}}+2\gamma c_{k+1}\rho d_{k+1}^{[\frac{1}{p^*}-\frac{1}{q}]_+}\left(\sqrt{\frac{(6k+4)\log 2}{n}}s_k\right.\\ &\left.+\prod_{i=1}^{k}\left(c_{i,2}c_{i,1}\rho^2(d_{i,1}d_{i,2})^{[\frac{1}{p^*}-\frac{1}{q}]_+}+1\right)d_0^{\frac{1}{p^*}}\sqrt{\frac{C(p)}{n}}\right).\end{aligned}$$

According to the given conditions in theorem 4.3, we assume that the range of our loss function is an interval $[a,a+1]$ (we set $a>0$). For MSE, namely $L(v_1,v_2)=|v_1-v_2|^2$, γ can be chosen as $\sqrt{a+1}$. Similarly, we set γ as 1 when we adopt MAE, that is $|v_1-v_2|$. In fact, we can bound the derivative of loss function and hence obtain γ. Generally, since the training data $(\mathbf{x}_i,y_i)$ and $f(\mathbf{x}_i)$ are finite (by definition 4.1), we can rescale the range of $L(f(\mathbf{x}),y)$ to be $[a,a+1]$. In this way, the case is reduced to the previous one.

The rescaling process implies that, once the loss value is small, the γ can be small, so do generalization bound. γ is a relative value to generalization bound and it contributes little effect to generalization bound. Hence, our work focuses on the dominate terms in generalization bound.

When weight matrixes are normalized with respect to the Frobenius norm, we set $p=q=2$. Moreover, when we set $1/p+1/q\geqslant 1$, the bound is reduced to a laconic form.

Corollary 4.4. *The Generalization Bound with $\frac{1}{p}+\frac{1}{q}\geqslant 1$.*
We assume that $\mathbf{z}\triangleq(\mathbf{x},y)\sim\mathcal{D}$, $\mathcal{S}\triangleq\{\mathbf{z}_1,\ldots,\mathbf{z}_n\}$ is a dataset with n i.i.d samples selected from the distribution $\mathcal{D}$. Then, we fix $\delta\in(0,1),\forall k\in\mathbb{N}^+,\forall d_{i,j}\in\mathbb{N}^+\ \ i=1,\ldots,k,\ j=1,2$. With probability of at least $1-\delta$ over the generation of $\mathcal{S}$, it holds that:

$$\begin{aligned}&\sup_{g\in\mathcal{RG}_\gamma}\left|\mathbb{E}_{\mathcal{D}}[g]-\hat{\mathbb{E}}_{\mathcal{S}}[g]\right|\\ \leqslant&\sqrt{\frac{\log(\frac{1}{\delta})}{2n}}+2\gamma c_{k+1}\rho\left(\sqrt{\frac{(6k+4)\log 2}{n}}s_k\right.\\ &\left.+\prod_{i=1}^{k}\left(c_{i,2}c_{i,1}\rho^2+1\right)d_0^{\frac{1}{p^*}}\sqrt{\frac{C(p)}{n}}\right).\end{aligned}$$

More generally, if the ith residual-block contains h_i layers, with the $\ell_{p,q}$-norm bound of each block set as $\|\mathbf{M}_{i,j}\|_{p,q}\leqslant c_{i,j},\ \forall j=1,2,\cdots,h_i$, then there is a similar result.

Corollary 4.5. *Generalization Bound in General Cases. We assume that* $\mathbf{z} \triangleq (\mathbf{x}, y) \sim \mathcal{D}$, $\mathcal{S} \triangleq \{\mathbf{z}_1, ..., \mathbf{z}_n\}$ *is a dataset with m i.i.d samples selected from the distribution* $\mathcal{D}$. *Then, we fix* $\delta \in (0,1), \forall k \in \mathbb{N}^+, \forall d_{i,j} \in \mathbb{N}^+,\ j = 1, ..., h_i,\ i = 1, ..., k$. *With probability of at least* $1-\delta$ *over the generation of* $\mathcal{S}$, *it holds that:*

$$\sup_{g \in \mathcal{RG}_\gamma} \left| \mathbb{E}_{\mathcal{D}}[g] - \hat{\mathbb{E}}_{\mathcal{S}}[g] \right| \leqslant c_{k+1} \rho d_{k+1}^{[\frac{1}{p^*} - \frac{1}{q}]_+} \left(\sqrt{\frac{(6k+4)\log 2}{n}} s_k + \prod_{i=1}^{k} \Big(\prod_{j=1}^{h_i} c_{i,j} + 1 \Big) d_0^{\frac{1}{p^*}} \sqrt{\frac{C(p)}{n}} \right),$$

where $s_0 = d_0^{\frac{1}{p^*}}$, *for* $i = 1, 2, \cdots, k$,

$$s_i = \Big(\prod_{j=1}^{h_i} c_{i,j} + 1 \Big) s_{i-1} + \sum_{j=1}^{h_i} \prod_{l=j}^{h_i} c_{i,l}.$$

When we set ReLU or another homogeneity functions as activation functions, the corresponding upper bounds $\{c_{i,j}\}_{j=1}^{h_i}$ and $\{c'_{i,j}\}_{j=1}^{h_i}$ hold $\prod_{j=1}^{h_i} c_{i,j} = \prod_{j=1}^{h_i} c'_{i,j}$. We can obtain a tighter bound with $s_0 = d_0^{\frac{1}{p^*}}, s_i = \big(\prod_{j=1}^{h_i} c_{i,j} + 1 \big) s_{i-1} + \prod_{l=1}^{h_i} c_{i,l},\ \forall i = 1, 2, \cdots, k$, by taking $c_{i,h_i}, \cdots, c_{i,2} \to 0^+$.

4.2 Comparisons

We focus on the term $\mathfrak{R}_n(\mathcal{RG}_\gamma)$ with $p = q = 2, \rho = 1$:

$$\begin{aligned} &\mathfrak{R}_n(\mathcal{RG}_\gamma) \\ =&\mathcal{O}\left\{ \gamma c_k \cdot \prod_{l=1}^{k-1} (c_l + 1) \cdot \left(2\sqrt{\frac{6k \log 2}{n}} + \sqrt{\frac{C(p)}{n}} \right) \right\} \\ \approx&\mathcal{O}\left\{ \gamma c_k \cdot \prod_{l=1}^{k-1} (c_l + 1) \cdot \sqrt{\frac{k}{n}} \right\}. \end{aligned}$$

A recent conclusion [Li *et al.*, 2018] argues that:

$$\begin{aligned} &\mathfrak{R}_n(\mathcal{RG}_\gamma) \\ =&\mathcal{O}\Bigg\{ \left(d_1 d_2 \cdot \log \frac{\gamma \sqrt{kn} \cdot \max_j \{B_{j,1}, B_{j,2}\}}{\min_j \{B_{j,1} + B_{j,2}\} \cdot \min_j \{B_{j,1} B_{j,2} + 1\}} \right)^{\frac{1}{2}} \\ &\cdot \gamma \cdot \prod_{j=1}^{k} (B_{j,1} B_{j,2} + 1) \cdot \sqrt{\frac{k}{n}} \Bigg\}, \end{aligned}$$

where $B_{j,i}$ is the upper bound of $\|\mathbf{M}_{j,i}\|_{2,2}$ with bias $\mathbf{b}_{j,i} = \mathbf{0}, j = 1, ..., k;\ i = 1, 2$. The widths of the 1st and 2nd layer in the jth residual-block are denoted as d_1, d_2, respectively. Since $c_j \leqslant B_{j,1} B_{j,2},\ \forall j = 1, ..., k$, we obtain a tighter generalization bound for the ResNets structures with bias, which can be extended from the Frobenius Norm to a general $\ell_{p,q}$-norm. In particular, $\forall p, q$ s.t. $1/p + 1/q \geqslant 1$ can reduce the generalization bound to a more laconic form .

The comparison shows that our generalization bound is tighter than [Li *et al.*, 2018]'s , which is the tightest previous generalization bound for ResNets and DNN. Comparisons between [Li *et al.*, 2018]'s work and previous work are included in that paper. While [Li *et al.*, 2018]'s work is restricted to Frobenius-norm, we generally obtain generalization bound with respect to $\ell_{p,q}$-norm.

5 Parallel Extensions for Weight-Normalized DNN

Using the same methods, our conclusions can be extended to Deep Neural Networks. We introduce the definition of the DNN function class as, $\widetilde{\mathcal{T}}_i : \mathbb{R}^{d_{i-1}+1} \to \mathbb{R}^{d_i+1}$, $\widetilde{\mathcal{T}}_i(\begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}) \triangleq \sigma[\begin{pmatrix} \mathbf{W}_i & \mathbf{b}_i \\ \mathbf{0} & 1 \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}], \forall \mathbf{x} \in \mathbb{R}^{d_{i-1}}$. We also introduce the notation as, $\mathbf{U}_i \triangleq \begin{pmatrix} \mathbf{W}_i & \mathbf{b}_i \\ \mathbf{0} & 1 \end{pmatrix}$. By sharing several notations $\widetilde{\mathcal{F}}_{k+1}, \mathbf{M}_{k+1}, \mathbf{c}, \mathbf{d}, k, n$, we obtain the parallel definitions:

Definition 5.1. $\mathcal{N}_{p,q,\mathbf{c}}^{k,\mathbf{d}}$ *This definition describes the class of functions* f *that can be represented by DNN.*
p, q: *The norm is set as* $\ell_{p,q}$*-norm.*
$\mathbf{d}$: $\mathbf{d} = (d_{k+1}, \ldots, d_1, d_0)$ *is a vector with width information.*
$\mathbf{c}$: $\mathbf{c} \triangleq (c_{k+1}, c_k, c_{k-1}, ..., c_1)$, *where* $\|\mathbf{U}_i\|_{p,q} \leqslant c_i,\ \forall i = 1, \ldots, k+1$;
$f \in \mathcal{N}_{p,q,\mathbf{c}}^{k,\mathbf{d}} \Leftrightarrow \exists\ \{\mathbf{U}_i\}_{i=1}^{k+1},\ s.t.\ f = \widetilde{\mathcal{T}}_{k+1} \circ \widetilde{\mathcal{T}}_k \circ \cdots \circ \widetilde{\mathcal{T}}_1$.

Theorem 5.2. *The properties of* $\mathcal{N}_{p,q,\mathbf{c}}^{k,\mathbf{d}}$
Since we have $p, q, \mathbf{c}^{(1)}, \mathbf{c}^{(2)}, k_1, k_2$ *and vectors* $\mathbf{d}^{(1)}, \mathbf{d}^{(2)}$ *with* $k_1 \leqslant k_2, \mathbf{d}^{(1)} \leqslant \mathbf{d}^{(2)}, \mathbf{c}^{(1)} \leqslant \mathbf{c}^{(2)}$ *w.r.t the components precede the* k_1*th component:*
(1): $\mathcal{N}_{p,q,\mathbf{c}^{(1)}}^{k_1,\mathbf{d}^{(1)}} \subset \mathcal{N}_{p,q,\mathbf{c}^{(2)}}^{k_1,\mathbf{d}^{(1)}}$
(2): $\mathcal{N}_{p,q,\mathbf{c}^{(1)}}^{k_1,\mathbf{d}^{(1)}} \subset \mathcal{N}_{p,q,\mathbf{c}^{(1)}}^{k_2,\mathbf{d}^{(2)}}$.

Definition 5.3. *The Loss Function Class* $\mathcal{G}_{p,q,\mathbf{c}}^{k,\mathbf{d}}$.
$\mathcal{G}_{p,q,\mathbf{c}}^{k,\mathbf{d}} \triangleq \{g(f(\mathbf{x}), y) \in \mathbb{R} |\ \mathbf{x} \in \mathcal{X}, y \in \mathcal{Y}, f \in \mathcal{N}_{p,q,\mathbf{c}}^{k,\mathbf{d}}\}$. *If we assume that* $\forall g \in \mathcal{G}_{p,q,\mathbf{c}}^{k,\mathbf{d}} : \mathcal{Y} \times \mathcal{Y} \to [0,1]$ *is a* γ*-Lipschitz function, the Loss Function Class is denoted as* $\mathcal{G}_\gamma \triangleq \{g \text{ is } \gamma\text{-Lipschitz} |\ g \in \mathcal{G}_{p,q,\mathbf{c}}^{k,\mathbf{d}}\}$.

Theorem 5.4. *An Estimation of Generalization Bound.*
We assume that $\mathbf{z} \triangleq (\mathbf{x}, y) \sim \mathcal{D}$, $\mathcal{S} \triangleq \{\mathbf{z}_1, ..., \mathbf{z}_n\}$ *is a dataset with m i.i.d samples selected from the distribution* $\mathcal{D}$. *Then, we fix* $\delta \in (0,1), \forall k \in \mathbb{N}^+, \forall d_i \in \mathbb{N}^+ \quad i = 1, ..., k$. *With probability of at least* $1-\delta$ *over the generation of* $\mathcal{S}$, $s_{k+1} \triangleq \sum_{i=1}^{k+1} \big(\prod_{l=i}^{k+1} c_l \rho d_l^{[\frac{1}{p^*} - \frac{1}{q}]_+} \big) + d_0^{\frac{1}{p^*}} \prod_{l=1}^{k+1} c_l \rho d_l^{[\frac{1}{p^*} - \frac{1}{q}]_+}$, *it holds that:*

$$\sup_{g \in \mathcal{G}_\gamma} \left| \mathbb{E}_{\mathcal{D}}[g] - \hat{\mathbb{E}}_{\mathcal{S}}[g] \right| \leqslant \sqrt{\frac{\log(\frac{1}{\delta})}{2n}} + 2\gamma \cdot \left(s_{k+1} \sqrt{\frac{(2k+4)\log 2}{n}} + \prod_{i=1}^{k+1} c_i \rho d_i^{[\frac{1}{p^*} - \frac{1}{q}]_+} d_0^{\frac{1}{p^*}} \sqrt{\frac{C(p)}{n}} \right).$$

Corollary 5.5. *The Generalization Bound with* $\frac{1}{p} + \frac{1}{q} \geqslant 1$.
We assume that $\mathbf{z} \triangleq (\mathbf{x}, y) \sim \mathcal{D}$, $\mathcal{S} \triangleq \{\mathbf{z}_1, ..., \mathbf{z}_n\}$ *is a*

dataset with n i.i.d samples selected from the distribution $\mathcal{D}$. Then, we fix $\delta \in (0,1), \forall k \in \mathbb{N}^+, \forall d_i \in \mathbb{N}^+ \quad i = 1, ..., k$. With probability of at least $1-\delta$ over the generation of $\mathcal{S}$, it holds that::

$$\begin{aligned}&\sup_{g\in\mathcal{G}_\gamma}\left|\mathbb{E}_{\mathcal{D}}[g]-\hat{\mathbb{E}}_{\mathcal{S}}[g]\right|\\ \leqslant&\sqrt{\frac{\log(\frac{1}{\delta})}{2n}}+2\gamma\bigg(\Big(\sum_{i=1}^{k+1}(\prod_{l=i}^{k+1}c_l\rho)+d_0^{\frac{1}{p^*}}\prod_{l=1}^{k+1}c_l\rho\Big)\\ &\cdot\sqrt{\frac{(2k+4)\log 2}{n}}+\prod_{j=1}^{k+1}c_j\rho d_0^{\frac{1}{p^*}}\sqrt{\frac{C(p)}{n}}\bigg).\end{aligned}$$

With ReLU as activation function, we obtain:

$$\begin{aligned}&\sup_{g\in\mathcal{G}_\gamma}\left|\mathbb{E}_{\mathcal{D}}[g]-\hat{\mathbb{E}}_{\mathcal{S}}[g]\right|\\ \leqslant&\sqrt{\frac{\log(\frac{1}{\delta})}{2n}}+2\gamma\bigg((d_0+1)^{\frac{1}{p^*}}\prod_{l=1}^{k+1}c_l\rho\sqrt{\frac{(2k+4)\log 2}{n}}\\ &+\prod_{j=1}^{k+1}c_j\rho d_0^{\frac{1}{p^*}}\sqrt{\frac{C(p)}{n}}\bigg).\end{aligned}$$

Comparisions:
We focus on the term $\mathfrak{R}_n(\mathcal{G}_\gamma)$ with $p=q=2, \rho=1$:

$$\begin{aligned}&\mathfrak{R}_n(\mathcal{G}_\gamma)\\ =&\mathcal{O}\bigg\{\gamma\bigg(2\Big(\sum_{i=1}^{k}(\prod_{l=i}^{k}c_l)+\prod_{l=1}^{k}c_l\Big)\sqrt{\frac{k\log 2}{n}}\\ &+\prod_{j=1}^{k}c_j\sqrt{\frac{C(p)}{n}}\bigg)\bigg\}\\ \approx&\mathcal{O}\left\{\gamma\left(\sum_{i=1}^{k}(\prod_{l=i}^{k}c_l)+\prod_{l=1}^{k}c_l\right)\sqrt{\frac{k}{n}}\right\}.\end{aligned}$$

A recent conclusion [Li *et al.*, 2018] argues that:

$$\begin{aligned}&\mathfrak{R}_n(\mathcal{G}_\gamma)\\ =&\mathcal{O}\left\{\gamma c_{k+1}\cdot\prod_{j=1}^{k}B_j\cdot\sqrt{d^2\cdot\log\left(\frac{\gamma\sqrt{kn}\cdot\max_j B_j}{\min_j B_j}\right)}\cdot\sqrt{\frac{k}{n}}\right\},\end{aligned}$$

where B_j is the upper bound of the corresponding $\|\mathbf{U}_j\|_{p,q}$, with bias $\mathbf{b}_j=\mathbf{0}, j=1,...,k$; $i=1,2$. The widths of the jth layer is denoted as d. We obtain a tighter generalization bound for DNN structures with bias, which can be extended from the Frobenius Norm ($p=q=2$) to a general $\ell_{p,q}$-norm.

For DNN, our result is tighter than [Xu and Wang, 2018]'s, since our $2(k+2)log2$ is smaller than their $(k+1)log16$. Moreover, the normalization bounds $\{c_i\}_{i=1}^k$ need not to be the same value and the weight normalization constrains are inequalities rather than equalities. Hence, our generalization bound is tighter and more general than Xu Y's.

6 Parallel Extensions for Kernel-Normalized CNN

Our conclusions can be extended to CNN cases, since a convolutional connection is tantamount to a full connection with sparse matrix. In this section, we represent the CNN structure by reducing the convolution blocks to a combination of linear functions and active functions.

In image recognition tasks, we feed a CNN with image data (pixel matrixes and RGB channels). To obtain a similar form as Section 4 and 5, we reshape the input matrixes for each layers as vectors. In practice, we rarely adopt networks that consist only of convolutional connection blocks. Without a loss of generality, we focus on a single convolution connection block. Hence, we introduce the following notations.

Definition 6.1. *The size of input image: $w_1 \times w_2$, The number of input and output channels: v_{in}, v_{out}, The size of kernels: u.*

For convenience, we reshape the image data as: $x[l] = image^{m_3}[m_2][m_1]$, for $l=(m_3-1)w_1w_2+(m_2-1)w_1+m_1$, $1\leqslant m_1\leqslant v_{in}, 1\leqslant m_2\leqslant w_2, 1\leqslant m_3\leqslant w_1$. This suggests that $x[l]$ corresponds to the element at the m_1th row and m_2th column of the kernel in the m_3th channel. Thus, we define the row of the weight matrix as:

$$\mathbf{V}_{i,j}^{(k)}(l)\triangleq\begin{cases}\mathcal{K}^{(m_1,k)}[m_3][m_2] & l\in A(m_1,m_2,m_3),\\ 0 & \text{else .}\end{cases}$$

where $\mathcal{K}^{(m_1,k)}$ is the m_1th kernel in kth channel,

$$\begin{aligned}&A(m_1,m_2,m_3)\\ \triangleq&\Big\{a\ \Big|\ a=(m_1-1)w_1w_2+(m_2-1)w_1+m_3,\\ &1\leqslant m_1\leqslant v_{in}, j\leqslant m_2\leqslant j+u-1, i\leqslant m_3\leqslant i+u-1\Big\}.\end{aligned}$$

By definition, the ℓ_p-norm of $\mathbf{V}_{i,j}^{(k)}$ is $\|(\mathcal{K}^{(1,k)},\mathcal{K}^{(2,k)},\ldots,\mathcal{K}^{(v_{in},k)})\|_p$. With a fixed value for k, there are w_1w_2 rows; therefore, we can represent the $\ell_{p,q}$-norm of matrix $\mathbf{M}$ as:

$$\|\mathbf{M}\|_{p,q}=\sum_{i=1}^{v_{out}}(w_1w_2\|(\mathcal{K}^{(1,i)},\mathcal{K}^{(2,i)},\cdots,\mathcal{K}^{(v_{in},i)},1)\|_p^q)^{\frac{1}{q}}$$

If we constraint each kernel as $\|\mathcal{K}\|_p\leqslant\hat{c}$, then

$$\|\mathbf{M}\|_{p,q}\leqslant(v_{in}\hat{c}^p+1)^{\frac{1}{p}}(w_1w_2)^{\frac{1}{q}}.$$

In these cases, we prefer kernel normalization to weight normalization. By substituting the normalization-bound c in Chapter 3 with $(v_{in}\hat{c}^p+1)^{\frac{1}{p}}(w_1w_2)^{\frac{1}{q}}$, we extend our conclusion to apply to CNN cases. Moreover, c can be rewritten as $\hat{c}(v_{in})^{\frac{1}{p}}(w_1w_2)^{\frac{1}{q}}$ when we set the bias as $\mathbf{0}$.

7 Numerical Experiments

In this section, we discuss numerical experiments that implies the generalization properties of ResNets is better than DNN's. In particular, we train two simple networks to solve a regression problem and calculate the generalization bound for the networks. While both of the networks share the same initialization and parameters, a residual shortcut is added to the second network.

Intuitively, we hypothesize that ResNets has the capacity to generalize better than DNN.

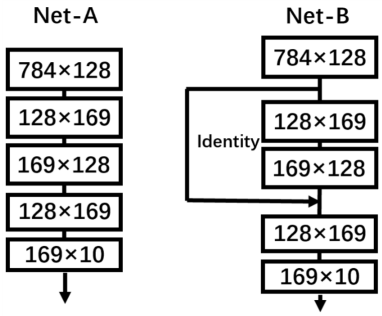

Figure 2: The parameter setting of Net-A and Net-B.

7.1 Experiment Settings

Dataset

We convey our experiment on a non-linear regression problem: $y = \frac{1}{1+\exp(-t)}$, $t = \mathbf{1}^T\mathbf{x} + \cos(\mathbf{1}^T\mathbf{x})$. We sample 500 random samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{500} \subset \mathbb{R}^{300} \times \mathbb{R}$ for training set, and 1500 samples for testing set from the following procedure.

1. Independently sample an auxiliary variable $\mathbf{z}_i \in \mathbb{R}^{300}$ from $N(\mathbf{0}, \mathbf{I})$.
2. Generate $\mathbf{x}_i$ by $\mathbf{x}_i[1] = \mathbf{z}_i[1]$, $\mathbf{x}_i[j] = \mathbf{z}_i[j] + 0.2(\mathbf{z}_i[j-1] + \mathbf{z}_i[j+1])$, for $j = 2, ..., 300$.
3. Compute $y_i = \frac{1}{1+\exp(-t_i)}$, $t_i = \mathbf{1}^T\mathbf{x}_i + \cos(\mathbf{1}^T\mathbf{x}_i)$ for $i = 1, ..., 500$ (for $i = 1, ..., 1500$ for test dataset).

Network Model

We first set Net-A and Net-B as two fully connected DNNs with four hidden layers. Then, we add a residual shortcut to Net-B between the second layer and the third one. The parameters are shown in Figure 2. Through the course of several experiments, we found that the choice of widths did not affect the general conclusion; hence, we selected the width parameters arbitrarily.

We first initialize the weights of Net-A by the Xavier Initialization and set all the bias components as 0.1, as the choice in bias does not greatly affect the conclusion. As a control group, Net-B shares all the initialized parameters with Net-A. We vary the scale of the initialization before training by dividing the weights from the Xavier Initialization by the 'scale'. Then, we adopt Mean Square Loss as loss function and ReLU as active functions.

After training Net-A and Net-B with the same strategy, we calculate the $\ell_{2,2}$-norm of their weights, respectively. By Thereom 4.3, 5.4, we compare the generalization bound of the two models (A_{GB} and B_{GB} for short). $A_{GB} > B_{GB}$ suggests that the ResNets structure has a relatively lower generalization bound while the testing error of Net-A and Net-B are close. We obtain evidence that supports our hypothesis by setting the scale as 10, 15, 20, 25, and other larger numbers. For each scale, we repeat the experiment for fifty times.

7.2 Results

In Figure 3, we display a representative result where the scale is set as 10. The rest of experiment data is concluded in the supplementary material [Mo and Chen, 2019].

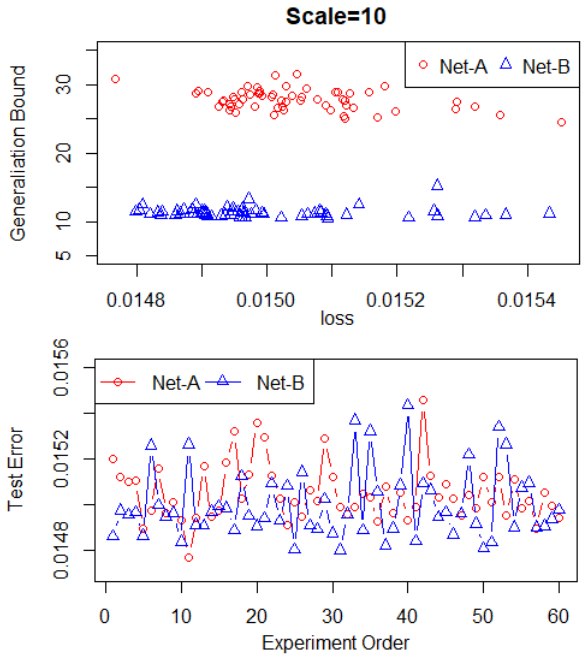

Figure 3: Generalization bound v.s. loss, test error v.s. experiment order for Net-A and Net-B.

The results suggest that ResNet structure has a lower generalization bound than DNN, while the test error of two models are close (more than 95% of the data holds $A_{GB} > B_{GB}$). The experiment implies that the ResNets structure contributes to better generalization properties.

8 Conclusion

8.1 Guidance on parameters to avoid overfitting

According to theorem 4.3, the following settings can improve the network structures' generalization capacity: (1).$1/p + 1/q >= 1$; (2).As the p,q-norm of each matrix $\mathbf{M}_i$ is constrained to be less than c_i, we can set c_i as constants that small enough to force generalization bound to be a small value; (3).The residual connection structures (short cut) render the generalization bound smaller, which explains why ResNets generalizes better than DNN does.

8.2 Future work

With the guidance, we can develop efficient algorithms to train Weight-Normalized networks with small generalization bound. In the future, we will focus on the generalization bound when it is restricted on a smaller function space (i.e. a vicinity of a local maximal solution) for further exploration of the generalization properties.

Acknowledgments

We would like to thank the anonymous reviewers for their comments and suggestions which greatly improve the manuscript. The work is supported by the NSF of China (No. 11871447), and Anhui Initiative in Quantum Information Technologies (AHY150200).

References

[Bartlett *et al.*, 2017] Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *NIPS*, 2017.

[Fan *et al.*, 2018] Yuchen Fan, Jincheng Yu, and Thomas S. Huang. Wide-activated deep residual networks based restoration for bpg-compressed images. In *CVPR Workshops*, 2018.

[Golowich *et al.*, 2018a] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 297–299. PMLR, 06–09 Jul 2018.

[Golowich *et al.*, 2018b] Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *COLT*, 2018.

[Goodfellow *et al.*, 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. 2016.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

[Li *et al.*, 2018] Xingguo Li, Junwei Lu, Zhaoran Wang, Jarvis Haupt, and Tuo Zhao. On tighter generalization bound for deep neural networks: Cnns, resnets, and beyond. *arXiv preprint arXiv:1806.05159*, 2018.

[Mo and Chen, 2019] Mo and Chen. IJCAI supplementary material for theoretical investigation of generalization bound for residual networks. http://s000.tinyupload.com/?file_id=52981413374013645026, 2019.

[Mohri *et al.*, 2012] Mehryar Mohri, Ameet Talwalkar, and Afshin Rostamizadeh. *Foundations of machine learning (adaptive computation and machine learning series)*. Mit Press Cambridge, MA, 2012.

[Neyshabur *et al.*, 2015] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *COLT*, 2015.

[Neyshabur *et al.*, 2017] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *CoRR*, abs/1707.09564, 2017.

[Shalev-Shwartz and Ben-David, 2014] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

[Srivastava *et al.*, 2015] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015.

[Sun *et al.*, 2016] Shizhao Sun, Wei Chen, Liwei Wang, Xiaoguang Liu, and Tie-Yan Liu. On the depth of deep neural networks: A theoretical view. In *AAAI*, 2016.

[Xu and Wang, 2018] Yixi Xu and Xiao Wang. Understanding weight normalized deep neural networks with rectified linear units. In *Advances in Neural Information Processing Systems*, pages 130–139, 2018.