A Comprehensive Review of Generalization Bounds in Neural Networks

Jincheng Song* **Marco Wong†**

Abstract

Stronger and tighter generalization bounds can provide meaningful insights into a model's generalization ability since they directly provide more precise predictions about it. This review project discusses the importance of generalization bounds from different perspectives. Particularly, these generalization bounds provide the upper limit for the generalization error and focus on explaining generalization behaviors, in other words, they measure how well the network can perform from training data to unseen data. This is important especially for neural networks which generalize better with over-parametrization. Additionally, over-parametrization has been shown to improve generalization performance in neural networks, which in turn can lead to tighter generalization bounds. Moreover, generalization bounds across different models typically depend on several factors such as the size of the training data which can be measured by sample complexity. Therefore, although we mainly reviewed the tighter generalization bounds derived for deep nets by a compression approach in the paper (A) "Stronger Generalization Bounds for Deep Nets via a Compression Approach", to get deeper understandings, we also chose two other papers contributing unique insights into these two aspects of this problem. Specifically, the papers (B, C) 'Size-Independent Sample Complexity of Neural Networks' and 'Towards Understanding the Role of Over-Parametrization in Generalization of Neural Networks' investigate the deeper relationship between sample complexity and generalization and the role of over-parametrization in the generalization performance of neural networks, respectively.

Keywords: Generalization bounds, compression approach, size-Independent sample complexity, over-parametrization.

Quick link to each paper:

Paper A(Section 2.1): https://arxiv.org/abs/1802.05296

Paper B(Section 2.2): https://arxiv.org/abs/1712.06541

Paper C(Section 2.3): https://arxiv.org/abs/1805.12076

Quick reference to each section: Sections 2, 3, 4, 5, 6, 7, 8, 9 and 10 below.

1 Introduction

Since we only provide the main contributions of each paper in these main sections, we also provide additional preliminaries in the appendix section for better understanding and convenience, as complementary materials to our explicit explanations.

*University of Toronto, 1265 Military Trail, Scarborough, ON M1C 1A4; E-mail: jincheng.song@mail.utoronto.ca

†University of Toronto, 1265 Military Trail, Scarborough, ON M1C 1A4; E-mail: marcoo.wong@mail.utoronto.ca

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

1.1 Overview of the Purpose and Scope of the Review Project

We are undertaking a review project to assess the results and significance of three research papers: Arora et al. [2018], Golowich et al. [2019], Neyshabur et al. [2018]. By reviewing these papers, we hope to gain a deeper understanding of generalization through drawing connections between each papers' approach to improve generalization. Additionally, we will present and evaluate the main theorems to bound generalization error and its limitations based on given assumptions.

2 Problem Statement

In this section, we state the main problem each paper attempts to solve and offer some explanation as to why the particular problems are significant to generalization bounds. Evaluating generalization performance is always a central question in the Deep Learning field. It encapsulates a networks ability to perform with unseen data and provides theoretical guarantees to understand the limits of neural networks.

2.1 Generalization Bounds Study via Compression-based Framework

This paper A by Arora et al. [2018] proposes a novel compression-based framework for deriving generalization bounds for deep feedforward neural networks. The authors Arora et al. [2018] develop new generalization bounds depending on the complexity of the compressed representation of the network, the number of training examples. The authors defined and utilized noise stability properties of deep nets for this novel compression approach. The paper also presents an experimental evaluation on synthetic and real-world datasets, demonstrating the tighter generalization bounds provided by the compression-based approach in various settings.

Understanding the generalization performance of deep neural networks and developing efficient compression techniques are essential because they allow researchers and practitioners to design more efficient and robust models for real-world applications. By identifying the noise stability properties that contribute to generalization, the paper aims to provide a more accurate and useful framework for analyzing and improving the performance of deep nets while also enabling their compression.

It is worth noting that the problem is challenging for several reasons:

(1) Deep nets have a large number of parameters and complex structures, making it difficult to derive and analyze the properties that affect their generalization performance and to compress them effectively.

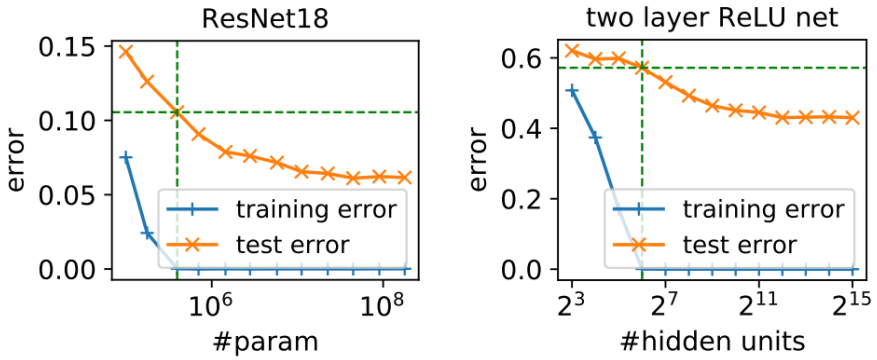
(2) The existing generalization bounds often involve pessimistic assumptions that do not accurately capture the behavior of deep nets in practice, leading to bounds that are not good enough.

2.2 Sample Complexity Study via New Rademacher Bounds

A previous study (Neyshabur et al. [2015]) had used a Rademacher complexity analysis to bound generalization error with exponential dependence on the network depth, 2^d. Exponential dependence was unavoidable when no other properties of sample complexity were considered. Exponential dependence can result with overfitting since the network fits to the noise of data.

2.3 The Impact of Over-Parametrization

Neyshabur et al. [2018] found complexity measures which depended on the total number of parameters of the network but fails to explain the improvement of generalization with over-parameterization. It was observed the test error of large networks continues decreasing when the number of parameters could encapsulate the training data.



The phenomenon can be visualized during the training of ResNet18. The reference line (Green) represents the optimal number of parameters.

3 Main Results

In the context of Machine Learning, simpler models with fewer parameters are more likely to generalize well because they have lower capacity and less likely to overfit the training data. Parameter counting has always been an intuitive way to control the model complexity. (Limitations: (1) However, parameters in a model do not contribute equally to the model's complexity, which means that it is not always a good idea to compare the number of parameters across different types of models.(2) while simpler models might have a better chance of generalizing well, this does not guarantee that they will always outperform more complex models since complex model are necessary to capture more complex patterns in the training data. (3) the ability to generalize is not solely determined by the number of parameters. Other factors, such as the quality and diversity of the training data, the choice of the learning algorithm, and the use of regularization techniques, can also play a significant role in a model's generalization performance.)

Above is why we need to explore more in this field.

4 Main Theorems and Algorithm for Paper A

4.1 Theorem 4.1

For a deep network with layers $A^1, A^2, ..., A^d$, where A^i is the weight matrix of the i-th layer, d is the depth of the network. Let h be the number of hidden units in the network, and output margin γ on training dataset S, then the generalization error can be bounded by:

$$\mathcal{O}\left(\sqrt{\frac{hd^2 \max_{x\in S}\|x\| \prod_{i=1}^{d}\|A^i\|_2^2 \sum_{i=1}^{d}\frac{\|A^i\|_F^2}{\|A^i\|_2^2}}{\gamma^2 m}}\right),$$

where

- $\max_{x\in S}\|x\|$ is the maximum norm of the input data vectors in the training dataset S.
- $\|A^i\|_2$ is the spectral norm (maximum singular value of this matrix) of the weight matrix A^i at the i-th layer.
- $\|A^i\|_F$ is the Frobenius norm (square root of the sum of the squared elements) of the weight matrix A^i at the i-th layer.
- $\prod_{i=1}^{d}\|A^i\|_2^2$ is related to the Lipschitz constant[3] of the net, I.e., the maximum norm of the vector it can produce if the input is a unit vector. Note that Lipschitz constant of matrix

[3]Lipschitz constant: A measure of how sensitive the network's output is to the input changes. Smaller Lipschitz constant means that this network is less sensitive to small input changes.

operator B is its spectral norm $\|B\|_2$, net applies a sequence of operations interspersed with ReLU which is 1-Lipschitz, so Lipschitz constant of the full net is at most $\prod_{i=1}^{d} \|A^i\|_2^2$.

- $\sum_{i=1}^{d} \frac{\|A^i\|_F^2}{\|A^i\|_2^2}$ is the sum of stable ranks[4] of the layers, it is natural measure of the parameter count.

Note: (1) It's important to take into account the complexity of the net by using the stable rank, because it combines both abilities of two norms, which is helpful to bound the generalization error.

(2) It's a pessimistic error analysis for this existing generalization error bound since they used Lipschitz constant in a worst-cae manner, Arora et al. [2018] improved this by giving concepts (A.4) of noise sensitivity, in further it improved the achievable compression and better generalization performance of deep nets.

4.2 Theorem 4.2

For any fully connected network f_A with $\rho_\sigma \geq 3d$, any probability $0 < \delta \leq 1$ and any margin γ. Algorithm 4.3 generates weights $\widetilde{A}$ for the network $f_{\bar{A}}$ such that with probability $1 - \delta$ over the training set and $f_{\bar{A}}$, the expected error $L_0(f_{\bar{A}})$ is bounded by:

$$\hat{L_\gamma}(f_{\bar{A}}) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{c^2 d^2 \max_{x \in S} \|f_A(x)\|_2^2 \sum_{i=1}^{d} \frac{1}{\mu_i^2 \mu_{i\rightarrow}^2}}{\gamma^2 m}}\right),$$

where μ_i, $\mu_{i\rightarrow}$, c and ρ_σ are larger cushion, interlayer cushion, activation contraction and interlayer smoothness defined in Definitions A.4, A.6, A.7 and A.8 respectively.

Note: (1) This is the most significant theorem of the paper A (Arora et al. [2018]) since it shows that it is possible to compress a neural network while maintaining or even improving the generalization performance of it.

(2) The above theorem provides a method to compress fully connected neural networks while maintaining a bound on the expected error, which depends on the performance on training data and noise stability properties. And this compression is achieved using the following Algorithm 1(4.3) since it generates new weights A for the compressed network $f_{\bar{A}}$. Additionally, the bounds generated by this theorem utilize the noise stability properties which are defined by Arora et al. [2018] and can be found at additional materials starting from A.4.

4.3 Algorithm 1

Algorithm 1: Matrix-Project(A, ϵ, η)

Require: Layer matrix $A \in \mathbb{R}^{h_1 \times h_2}$, error parameter ϵ, η.
Ensure: Returns $\hat{A}$ s.t. $\forall$ fixed vectors u, v,

$$Pr[\|u^T \hat{A} v - u^T A v\| \geq \epsilon \|A\|_F \|u\| \|v\|] \leq \eta.$$

Sample $k = log(1/\eta)/\epsilon^2$ random matrices $M_1, \ldots, M_k$ with entries i.i.d. ± 1 ("helper string")
for k'= 1 to k **do**
Let $Z_{k'} = \langle A, M_{k'} \rangle M_{k'}$.
end for
Let $\hat{A} = \frac{1}{k} \sum_{k'=1}^{k} Z_{k'}$

Note: (1)This is the most important algorithm for the compression frame. By applying the algorithm 1 to each layer of the network, it becomes compressed.

[4]Stable rank: It is the ratio of its squared Frobenius norm to its squared spectral norm, denoted by $\frac{\|A\|_F^2}{\|A\|_2^2}$, it is a measure of the effective parameter count or the complexity of a layer, and it is used to help bound the generalization error.

(2) The algorithm called Matrix-Project that takes a layer matrix A as input, an error parameter ϵ, and a probability parameter η, and returns a compressed matrix $\hat{A}$ that satisfy a certain property. That is, for any fixed vectors u and v, the probability that the difference between the dot product of u and $\hat{A}v$ the dot product of u and Av is greater than or equal to ϵ times the Frobenius norm of A times the norms of u and v, is at most η. It guarantees that the compressed matrix $\hat{A}$ is a close approximation of the original matrix A.

(3) This algorithm works by sampling k random matrices with entries that are either $+1$ or -1 which are called "help strings", and then computing a weighted average of the dot products of A and each of the sampled matrices, which is the inner mechanism of the network.

(4) This algorithm is applied to the trained weight matrices A at each layer, to obtain corresponding compressed matrices $\hat{A}$ at each layer with reduced size (or complexity) that maintains the similar level of performance of the original matrix.

4.4 Theorem 4.3

For any convolutional neural network f_A with $\rho_\delta \geq 3d$, any probability $0 < \delta \leq 1$ and any margin γ, Algorithm 4 generates weights $\tilde{A}$ for the network $f_{\tilde{A}}$ such that with probability $1 - \delta$ over the training set and $f_{\tilde{A}}$:

$$L_0\left(f_{\tilde{A}}\right) \leq \hat{L}_\gamma\left(f_A\right) + \tilde{O}\left(\sqrt{\frac{c^2 d^2 \max_{x \in S} \|f_A(x)\|_2^2 \sum_{i=1}^{d} \frac{\beta^2([\kappa_i / s_i])^2}{\mu_i^2 \mu_{i\rightarrow}^2}}{\gamma^2 m}}\right)$$

where $\mu_i, \mu_{i\rightarrow}, c, \rho_\delta$ and β are layer cushion, interlayer cushion, activation contraction, interlayer smoothness and well-distributed Jacobian defined in Definitions A.6, A.12, A.8, A.9, A.13, respectively. Furthermore, s_i and κ_i are stride and filter width in layer i.

Note: (1) This theorem provides a bound for the expected error $L_0(f_{\tilde{A}})$ of the compressed convolutional neural network $f_{\tilde{A}}$.

(2) This bound is the sum of the empirical risk with margin $\hat{L}_\gamma(f_A)$ of the original network and a term depending on the product of some properties.

(3) This theorem shows that it is possible to compress convolutional neural networks while maintaining a bound on the expected error.

5 Main Theorems and Algorithm for Paper B

5.1 Theorem 5.1

The generalization error bound derived by Neyshabur et al. [2015] in B.2 and Golowich et al. [2019] in B.3 is further improved by representing the prediction function of a network as a construction of a shallow network and univariate Lipschitz functions. New bound,

$$\mathcal{O}\left(B\left(\prod_{j=1}^{d} M_F(j)\right) \cdot \min\left\{\sqrt{\frac{\log\left(\frac{1}{\Gamma}\prod_{j=1}^{d} M_F(j)\right)}{\sqrt{m}}}, \sqrt{\frac{d}{m}}\right\}\right)$$

where

- $M_F(j)$ is an upper bound on the Schatten p-norm of W_j
- Γ is a lower bound on the product of the *spectral* norms of the parameter matrices, $\prod_{j=1}^{d} \|W_j\|$
- B represents the upper bound of input, $\|\mathbf{x}\| \leq B$

The new generalization bound is the result of converting depth-dependent bounds into depth-independent bounds. If a network finds a non-trivial solution and assuming the product of its Schatten p-norms is bounded, then atleast one parameter matrix (close to rank-1) should exist.

5.2 Theorem 5.2

For any $p \in [1, \infty)$, any network $N_{W_1^d}$ such that $\prod_{j=1}^{d} \|W_j\| \geq \Gamma$ and $\prod_{j=1}^{d} \|W_j\|_p \leq M$, and for any $r \in \{1, \ldots, d\}$, there exists another network $N_{\tilde{W}_1^d}$ (of the same depth and layer dimensions) with the following properties:

- $\tilde{W}_1^d = \{\tilde{W}_1, \ldots, \tilde{W}_d\}$ is identical to W_1^d, except for the parameter matrix $\tilde{W}_{r'}$ in the r'-th layer, for some $r' \in \{1, 2, \ldots, r\}$. The matrix $\tilde{W}_{r'}$ is of rank at most 1, and equals $s\mathbf{u}\mathbf{v}^\top$ where $s, \mathbf{u}, \mathbf{v}$ are some leading singular value and singular vectors pairs of $W_{r'}$.
- $\sup_{\mathbf{x} \in \mathcal{X}} \|N_{W_1^d}(\mathbf{x}) - N_{\tilde{W}_1^d}(\mathbf{x})\| \leq B \left(\prod_{j=1}^{d} \|W_j\| \right) \left(\frac{2p\log(M/\Gamma)}{r} \right)^{1/p}$.

6 Main Theorems and Algorithm for Paper C

6.1 Theorem 6.1

Given a training set $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^m$ and $\gamma > 0$, Rademacher complexity of the composition of loss function ℓ_γ over the class $\mathcal{F}_\mathcal{W}$ defined in equations (4) and (5) is bounded as follows:

$$\begin{aligned}
\mathcal{R}_\mathcal{S}\left(\ell_\gamma \circ \mathcal{F}_\mathcal{W}\right) &\leq \frac{2\sqrt{2c}+2}{\gamma m} \sum_{j=1}^{h} \alpha_j \left(\beta_j \|\mathbf{X}\|_F + \left\|\mathbf{u}_j^0 \mathbf{X}\right\|_2 \right) \\
&\leq \frac{2\sqrt{2c}+2}{\gamma\sqrt{m}} \|\boldsymbol{\alpha}\|_2 \left(\|\boldsymbol{\beta}\|_2 \sqrt{\frac{1}{m}\sum_{i=1}^{m} \|\mathbf{x}_i\|_2^2} + \sqrt{\frac{1}{m}\sum_{i=1}^{m} \|\mathbf{U}^0 \mathbf{x}_i\|_2^2} \right).
\end{aligned}$$

6.2 Theorem 6.2

For any $h \geq 2, \gamma > 0, \delta \in (0,1)$ and $\mathbf{U}^0 \in \mathbb{R}^{h \times d}$, with probability $1 - \delta$ over the choice of the training set $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^d$, for any function $f(\mathbf{x}) = \mathbf{V}[\mathbf{U}\mathbf{x}]_+$ such that $\mathbf{V} \in \mathbb{R}^{c \times h}$ and $\mathbf{U} \in \mathbb{R}^{h \times d}$, the generalization error is bounded as follows:

$$\begin{aligned}
L_0(f) &\leq \hat{L}_\gamma(f) + \tilde{O}\left(\frac{\sqrt{c}\|\mathbf{V}\|_F \left(\left\|\mathbf{U} - \mathbf{U}^0\right\|_F \|\mathbf{X}\|_F + \left\|\mathbf{U}^0 \mathbf{X}\right\|_F \right)}{\gamma m} + \sqrt{\frac{h}{m}} \right) \\
&\leq \hat{L}_\gamma(f) + \tilde{O}\left(\frac{\sqrt{c}\|\mathbf{V}\|_F \left(\left\|\mathbf{U} - \mathbf{U}^0\right\|_F + \left\|\mathbf{U}^0\right\|_2 \right) \sqrt{\frac{1}{m}\sum_{i=1}^{m} \|\mathbf{x}_i\|_2^2}}{\gamma\sqrt{m}} + \sqrt{\frac{h}{m}} \right).
\end{aligned}$$

6.3 Theorem 6.3

Theorem 3. Define the parameter set

$$\mathcal{W}' = \left\{ (\mathbf{V}, \mathbf{U}) \mid \mathbf{V} \in \mathbb{R}^{1 \times h}, \mathbf{U} \in \mathbb{R}^{h \times d}, \|\mathbf{v}_j\| \leq \alpha_j, \left\|\mathbf{u}_j - \mathbf{u}_j^0\right\|_2 \leq \beta_j, \left\|\mathbf{U} - \mathbf{U}^0\right\|_2 \leq \max_{j \in h} \beta_j \right\},$$

and let $\mathcal{F}_{\mathcal{W}'}$ be the function class defined on $\mathcal{W}'$ by equation (5). Then, for any $d = h \leq m, \{\alpha_j, \beta_j\}_{j=1}^h \subset \mathbb{R}^+$ and $\mathbf{U}_0 = \mathbf{0}$, there exists $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^d$, such that

$$\mathcal{R}_\mathcal{S}\left(\mathcal{F}_\mathcal{W}\right) \geq \mathcal{R}_\mathcal{S}\left(\mathcal{F}_{\mathcal{W}'}\right) = \Omega\left(\frac{\sum_{j=1}^{h} \alpha_j \beta_j \|\mathbf{X}\|_F}{m} \right)$$

6.4 Corollary 6.4

$\forall h = d \leq m, s_1, s_2 \geq 0, \exists \mathcal{S} \in \mathbb{R}^{d \times m}$ such that $\mathcal{R}_{\mathcal{S}}\left(\mathcal{F}_{\mathcal{W}_{\text{spec}}}\right) = \Omega\left(\frac{s_1 s_2 \sqrt{h} \|\mathbf{X}\|_F}{m}\right)$.

6.5 Theorem 6.5

For any $h, p \geq 2, \gamma > 0, \delta \in (0,1)$ and $\mathbf{U}^0 \in \mathbb{R}^{h \times d}$, with probability $1-\delta$ over the choice of the training set $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^m \subset \mathbb{R}^d$, for any function $f(\mathbf{x}) = \mathbf{V}[\mathbf{U}\mathbf{x}]_+$such that $\mathbf{V} \in \mathbb{R}^{c \times h}$ and $\mathbf{U} \in \mathbb{R}^{h \times d}$, the generalization error is bounded as follows:

$$L_0(f) \leq \hat{L}_\gamma(f) + \tilde{O}\left(\frac{\sqrt{c} h^{\frac{1}{2}-\frac{1}{p}} \left\|\mathbf{V}^T\right\|_{p,2} \left(h^{\frac{1}{2}-\frac{1}{p}} \left\|\mathbf{U}-\mathbf{U}^0\right\|_{p,2} \|\mathbf{X}\|_F + \left\|\mathbf{U}^0\mathbf{X}\right\|_F\right)}{\gamma m} + \sqrt{\frac{e^{-p} h}{m}}\right),$$

6.6 Corollary 6.6

Under the settings of Theorem 5, with probability $1-\delta$ over the choice of the training set $\mathcal{S} = \{\mathbf{x}_i\}_{i=1}^m$, for any function $f(\mathbf{x}) = \mathbf{V}[\mathbf{U}\mathbf{x}]_+$, the generalization error is bounded as follows:

$$\begin{aligned} L_0(f) &\leq \hat{L}_\gamma(f) + \tilde{O}\left(\frac{\sqrt{c} h^{\frac{1}{2}-\frac{1}{\ln h}} \left\|\mathbf{V}^T\right\|_{\ln h,2} \left(h^{\frac{1}{2}-\frac{1}{\ln h}} \left\|\mathbf{U}-\mathbf{U}^0\right\|_{\ln h,2} \|\mathbf{X}\|_F + \|\mathbf{U}_0\mathbf{X}\|_F\right)}{\gamma m}\right) \\ &\leq \hat{L}_\gamma(f) + \tilde{O}\left(\frac{\sqrt{c} h^{\frac{1}{2}-\frac{1}{\ln h}} \left\|\mathbf{V}^T\right\|_{\ln h,2} \left(h^{\frac{1}{2}-\frac{1}{\ln h}} \left\|\mathbf{U}-\mathbf{U}^0\right\|_{\ln h,2} + \|\mathbf{U}_0\|_2\right) \sqrt{\frac{1}{m}\sum_{i=1}^m \|\mathbf{x}_i\|_2^2}}{\gamma\sqrt{m}}\right). \end{aligned}$$

7 Examples and Counterexamples

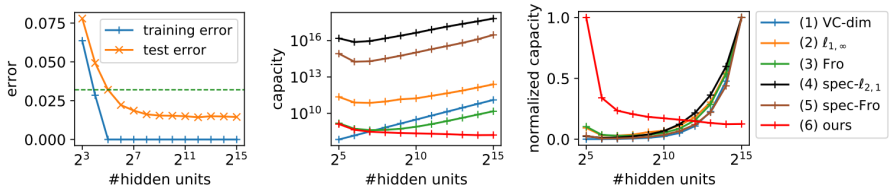
7.1 Compression Algorithm

We can apply the main compression algorithm4.3, Matrix-Project, to a simple feedforward neural network with a single hidden layer. This network will be used for a binary classification task on a small dataset, such as the XOR problem, where input data points are two-dimensional $(x1, x2)$ with labels being 0 or 1 based on whether $x1$ and $x2$ have the same or different signs.

Finally, we can observe the benefits of reduced memory footprint and maintained noise stability and generalization performance, while also getting a better understanding of the algorithm's practical meanings.

7.2 Experiments with over-parametrization

A two layer ReLU network was trained using SGD on CIFAR-10, SVHN and MNIST datasets with architectures varying from size 2^3 to 2^15. Each jump in size corresponded to an increase in hidden units by a factor of 2.



The left panel shows the effects of over-parametrization for a model trained on MNIST dataset. The middle and right panels compare the capacity bounds derived by Neyshabur et al. [2018] compared to previous works.

8 Thoughts about Empirical Studies

Based on the existing studies by these researchers, we can gain much insights into how to conduct effective empirical studies.

For Paper A, conducting further empirical evaluation of the proposed compression algorithm on a broader range of datasets and tasks would help validate the approach's effectiveness and generalizability. This could involve testing the algorithm on larger benchmark datasets, different types of neural networks (e.g., recurrent or transformer-based), and various tasks (e.g., natural language processing, reinforcement learning).

9 Limitations

9.1 Limitations of the Paper A

9.1.1 Limitations of the compression framework and main algorithm

(1) Overly Restrictive Assumptions: Some of the assumptions made in the paper, such as those related to noise stability properties, layer cushion, and interlayer cushion, might limit the applicability of the paper's results in real-world situations.

(2) Computational Practicality: The proposed compression algorithm 4.3, Matrix-Project, may be computationally expensive for large-scale deep neural networks, as it requires sampling and computing a weighted average of multiple random matrices across layers. This might limit the algorithm's applicability in practical settings where resources are constrained or where real-time processing is required.

9.1.2 Evaluations of these Limitations

(1) Although these assumptions might be restrictive, they help in establishing a theoretical framework for understanding the generalization performance of deep neural networks. It is not only a starting point since it has already been extended to accommodate different architectures or training strategies, such as convolutional neural networks.

(2) Although the computational complexity of this novel algorithm exists as a limitation, we should know that further research could lead to more efficient techniques, such as sampling methods or exploiting sparsity in the weight matrices.

9.2 Limitations of Norm-Based Bounds

Constraining multiple norms within bounds had allowed Golowich et al. [2019] to establish generalization guarantees for feed-forward neural networks. But these guarantees do not apply for Graph Neural Networks. GNNs and FFNs vary with their training data structure and the type of data each network can handle. FFNs are designed to accept data formatted as vectors or matrices for tasks such as image classification or natural language processing. Typically, FFN data contains features independent of each other with multiple layers of neurons. Whereas, GNN processes asymmetrical graph-structured data so dependencies and relationships between graph nodes may be captured. Therefore, different applications and training data don't allow generalization guarantees of FFNs to be applied directly to other NNs such as GNNs.

9.3 Limitations of Over-Parametrization

The capacity bounds derived by Neyshabur et al. [2018] successfully increased the lower bound of generalization performance for larger networks, but this understanding only applied to two layer ReLU networks. Moreover, the use of over-parametrization requires increased memory and computational power for model training and inferences.

10 Future Directions

10.1 Future Directions for Paper A

(1) As we mentioned in the limitations 9.1, future research could focus on developing a more robust theoretical framework that can capture the behavior of various types of deep neural networks, such as alternative optimization algorithms, activation functions, and regularization techniques on noise stability properties and generalization performance. Based on that, "how can the theoretical framework be extended to cover a broader range of deep net architectures and training strategies while maintaining the insights on noise stability properties and generalization" would be a good question for this field.

(2) Moreover, we can improving the computational efficiency of the proposed compression algorithm 4.3. For instance, we can develop parallel processing techniques to reduce the computational overhead.

(3) Additionally, researchers can combine the novel approach with other compression techniques to lead a more effective compression strategies, without sacrificing model accuracy.

10.2 Future Directions for Norm-based Lower Bounds

One assumption from Golowich et al. [2019] stated the product of spectral norms was constant and upper bounded for some R, $\prod_j M_F(j) \leq R$. At the time, it was the first norm-based assumptions that led to explicit generalization bounds for neural networks which are size-independent. Bounding by R was a very strong assumption and is very restrictive in practice. Additionally, the lower bounds set on the product of norms were the worst case scenarios of Lipschitz constant.

Bound improvements can be made by also considering a network's Jacobian norm. Which introduces empirical Lipschitz constants much smaller than the product of norms, $\prod_j M_F(j)$ (Wei and Ma [2020])

10.3 Future Directions for Over-parametrization

A future goal of over-parametrization would ideally apply the capacity bounds of two layer networks onto deeper networks. Furthermore, the absolute value of the bound described in 6.2 are too large and far exceed the training data size. One can further research the relationship between generalization and parametrization by considering language models. For example, GPT-3 has 175 billion parameters with impressive generalization on NLP tasks.

Acknowledgments and Disclosure of Funding

We would like to express our sincere gratitude to Professor Qiang Sun for his valuable guidance and support throughout the *Big Data* course. His expertise and feedback were instrumental in helping us to navigate the complexities of neural network generalization and to produce a comprehensive review of this important topic. We would also like to thank our teaching assistants, Archer Gong Zhang and Hengchao Chen, for their assistance and support in providing feedback and answering our questions. Their contributions were vital to the success of this project. Finally, we would like to thank the Big Data course for providing us with the opportunity to explore this fascinating and challenging field, and for helping us to develop the skills and knowledge necessary to succeed in the world of big data.

References

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 254–263. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/arora18b.html.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data, 2017.

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. 2019.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. page 1376–1401, 2015.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. 2018.

Edwin PD Pednault. *Statistical learning theory*. Citeseer, 1997.

V.N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10 (5):988–999, 1999. doi: 10.1109/72.788640.

Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation, 2020.

A Preliminaries for Compression Frame

The new defined Definition A.1 and Definition A.2 by Arora et al. [2018] below are defined for the new compression frame. Definition 1 is the concept of compressibility of a classifier in the context of deep neural networks. Definition 2 is an improved version of that concept using a helper string.

Before that, let S be a set of input samples, x be any sample in it, y be the corresponding true label, $f(x) \in \mathbb{R}^k$ be the multi-class classifier. The margin γ is the difference between the score of the predicted label (with the highest score) and the score of the next highest label, it is a measure of how confident the deep nets is in the prediction it made. Then, the classification loss (Arora et al. [2018]) for any classification distribution $\mathcal{D}$ is:

$$\mathbb{P}_{(x,y)\sim\mathcal{D}}[f(x)[y] \leq max_{i\neq y} f(x)[j]]$$

where:

- $f(x)[y]$ is the y-th coordinate of $f(x)$, which is the score of the true label y in the output vector.
- $max_{i\neq y} f(x)[j]$ is the highest score of the output vector.
- $f(x)[y] < max_{i\neq y} f(x)[j]$ means all incorrect labels I.e., the highest score is not associated with the true label y.

Note: The margin $\gamma = 0$ for the classification loss.

Since we want to encourage a large margin to further improve the generalization performance, there would be a desired margin $\gamma > 0$ for the following concept of margin loss (Arora et al. [2018]) on the training data S:

$$L_\gamma(f) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[f(x)[y] \leq \gamma + max_{i\neq y} f(x)[j]]$$

Note: The margin loss can reflect the performance of a model on given training data.

To evaluate the model's performance on unseen data, we need the concept of empirical estimate of the margin loss (I.e., expected loss) $\hat{L_\gamma}(f)$. Then we have the concept of the generalization error which is the difference between $L_\gamma(f)$ and $\hat{L_\gamma}(f)$ I.e., expected difference between the performance of model on training and new data. The generalization error is used to measure how well a model can generalize its learning process trained on training data to make accurate predictions on new data. Finally, the generalization bound we are interested here is the upper bounds of the generalization error, which are derived from Statistics Learning Theory[5].

[5]Statistics Learning Theory(Vapnik [1999]): It is the field of study that combines the statistics and machine learning to provide a rigorous framework for understanding the limits and possibilities of machine learning. It is worth noting that the main problem it focuses is how to quantify the performance of a learning algorithm and how to optimize it based on the available data, such as developing theoretical bounds on expected error, designing learning algorithms(Pednault [1997]).

A.1 Definition 1

((γ, S)-compressible). For any set $\mathcal{A}$ of parameter values, let f be a classifier and $G_{\mathcal{A}} = \{g_A | A \in \mathcal{A}\}$ be a class of classifiers. We say f is (γ, S)-compressible via $G_{\mathcal{A}}$ if there exists a set of trainable parameters $A \in \mathcal{A}$ such that for any $x \in S$, we have for all output label y:

$$|f(x)[y] - g_A(x)[y]| \leq \gamma.$$

In other words, f can be approximated by a classifier g_A in the classifier class $G_{\mathcal{A}}$ within a small positive number γ.

A.2 Definition 2

((γ, S)-compressible using helper string s) Suppose $G_{\mathcal{A},s} = \{g_{A,s} | A \in \mathcal{A}\}$ be a class of classifiers indexed by trainable parameters A and fixed strings s. A classifier f is (γ, S)-compressible with respect to $G_{\mathcal{A},s}$ using helper string s if there exists a set of trainable parameters $A \in \mathcal{A}$ such that for any $x \in S$, we have for all output label y:

$$|f(x)[y] - g_{A,s}(x)[y]| \leq \gamma.$$

Note: Both definitions are connecting the compressibility of a classifier and its generalization performance. Specifically, if a classifier f can be compressed or approximated by a simpler model within a small loss γ in performance on training data, we expect this compressed classifier g to have a better generalization properties than the original classifier f based on the common sense that we mentioned in the Section 3.

The complementary Lemma 1 below is used to prove the Theorem **??**, by compressing each weight matrix A^i across layers to matrices of lower rank.

Before that, we should know that a m-by-n matrix of rank r can be decomposed into the product of two matrices with inner dimension r and totally $mr + nr$ parameters which is reduced. Specifically, a square matrix with dimension $h \times h$ (I.e., h^2 entities or parameters) and rank r, can be decomposed to two matrices with inner dimension r and totally $hr + hr = 2hr$ parameters which is reduced a lot if $r \ll h$.

A.3 Lemma 1.

For any matrix $A \in \mathbb{R}^{m \times n}$, let $\hat{A}$ be the truncated version of A where singular values that are smaller than $\delta \|A\|_2$ are removed. Then $\|\hat{A} - A\|_2 \leq \delta \|A\|_2$ and $\hat{A}$ has rank at most $\frac{\|A\|_F^2}{\delta^2 \|A\|_2^2}$.

Note: This lemma provides a way to truncate any matrix A through removing singular values which are smaller than a certain threshold $\delta \|A\|_2$. Also, the resulting spectral norm of the truncated matrix $\hat{A}$ is still close to the original matrix A's, and we effectively reduce the rank of the matrix in this way.

Base on the Lemma 1 we can use a simple induction to prove that γ is the upper bound of the total error incurred in all layers. And then we can get the generalization bound based on the Theorem 4.1.

A.4 Definition 3

(noise sensitivity for a mapping M.) If M is a mapping from real-valued vectors to real-valued vectors, and $\mathcal{N}$ is some noise distribution then noise sensitivity of M at x with respect to $\mathcal{N}$, is

$$\psi_{\mathcal{N}}(M, x) = \mathbb{E}_{\eta \in \mathcal{N}} \left[\frac{\|M(x + \eta \|x\|) - M(x)\|^2}{\|M(x)\|^2} \right],$$

The noise sensitivity of M with respect to $\mathcal{N}$ on a set of inputs S, denoted $\psi_{\mathcal{N},S}(M)$, is the maximum of $\psi_{\mathcal{N}}(M, x)$ over all inputs x in S.

Note: This definition provides a sense of how sensitive a mapping is to the injected noise which follows some certain distribution $\mathcal{N}$, on some given input dataset S.

A.5 Proposition 3.1.

(noise sensitivity for a matrix M.) The noise sensitivity of a matrix M at any vector $x \neq 0$ with respect to Gaussian distribution $\mathcal{N}(0, I)$ is exactly $\|M\|_F^2 \|x\|^2 / \|Mx\|^2$, and at least its stable rank.

Note: This proposition provides a quantitative measure of the noise sensitivity of a given matrix M, in further we can use this measure to the analysis of the noise stability properties of the deep nets.

A.6 Definition 4

(layer cushion). The layer cushion of layer i is similarly defined to be the largest number μ_i such that for any $x \in S, \mu_i \left\|A^i\right\|_F \left\|\phi\left(x^{i-1}\right)\right\| \leq \left\|A^i \phi\left(x^{i-1}\right)\right\|$.

Note: This is a measure of error-resilience properties for a single layer in the deep nets. It quantifies how much the output of a single layer is affected by the perturbation in input data.

A.7 Definition 5

(Interlayer Cushion). For any two layers $i \leq j$, we define the interlayer cushion $\mu_{i,j}$ as the largest number such that for any $x \in S$:

$$\mu_{i,j} \left\|J_{x^i}^{i,j}\right\|_F \left\|x^i\right\| \leq \left\|J_{x^i}^{i,j} x^i\right\|$$

Furthermore, for any layer i we define the minimal interlayer cushion as $\mu_{i\rightarrow} = \min_{i \leq j \leq d} \mu_{i,j} = \min\left\{1/\sqrt{h^i}, \min_{i<j\leq d} \mu_{i,j}\right\}\Big\}$.

Note: (1) This is a measure of error-resilience properties between two layers in the deep nets. It quantifies how much the output of a single layer j is affected by the perturbation in the layer i, where $i \leq j$. Specifically, the larger the Cushion value is, the less sensitive the network is to the small perturbations between these two layers, which means the error resilience is high.

(2) For any layer i, the minimal Interlayer Cushion value can be considered as the indicator of the weakest error resilience between it and all layers j come after.

A.8 Definition 6

(Activation Contraction). The activation contraction c is defined as the smallest number such that for any layer i and any $x \in S$,

$$\left\|\phi\left(x^i\right)\right\| \geq \left\|x^i\right\| / c.$$

Note: It is a property of the activation function ϕ and it is a measure of how much the activation function shrinks the norm of the input vector x^i. Specifically, the smaller the activation contraction value c is, the less the norm of the input vector shrinks, which is a good phenomenon since it maintains the overall magnitude of the input vector as it goes through the whole network. In this way, we can better understand the behaviours of the network and its resilience to input perturbations so that gain more insights about factors of the generalization performance.

A.9 Definition 7

(Interlayer Smoothness). Let η be the noise generated as a result of substituting weights in some of the layers before layer i using Algorithm 1. We define interlayer smoothness ρ_δ to be the smallest number such that with probability $1-\delta$ over noise η for any two layers $i < j$ any $x \in S$:

$$\left\|M^{i,j}\left(x^i+\eta\right) - J_{x^i}^{i,j}\left(x^i+\eta\right)\right\| \leq \frac{\|\eta\| \left\|x^j\right\|}{\rho_\delta \left\|x^i\right\|}$$

Note: It is a measure of how well the mapping between two layers is preserved, under the noise perturbation in the weights. In particular, the higher interlayer smoothness value ρ_δ is, the more stable and robust the mapping is against the noise, which in further is good for the compression and generalization performance. It provides sights into nets' resilience and generalization ability, by quantifying the robustness of the inter-layer mapping.

A.10 Lemma 2.

For any $0 < \delta, \varepsilon \leq 1$, et $G = \left\{\left(U^i, x^i\right)\right\}_{i=1}^m$ be a set of matrix/vector pairs of size m where $U \in \mathbb{R}^{n \times h_1}$ and $x \in \mathbb{R}^{h_2}$, let $\hat{A} \in \mathbb{R}^{h_1 \times h_2}$ be the output of Algorithm 11 with $\eta = \delta/\mathrm{mn}$. With probability at least $1 - \delta$ we have for any $(U, x) \in G$, $\|U(\hat{A} - A)x\| \leq \varepsilon \|A\|_F \|U\|_F \|x\|$.

Remark 1.

Lemma 2 can be used to upper bound the change in the network output after compressing a single layer if the activation patterns remain the same. For any layer, in the lemma statement take x to be the input to the layer, A to be the layer weight matrix, and U to be the Jacobian of the network output with respect to the layer output. Network output before and after compression can then be calculated by the matrix products UAx and $U\hat{A}x$ respectively. Hence, the lemma bounds the distance between network output before and after compression.

Next Lemma bounds the number of parameters of the compressed network resulting from applying Algorithm 1 to all the layer matrices of the net. The proof does induction on the layers and bounds the effect of the error on the output of the network using noise properties defined by Arora et al. [2018].

A.11 Lemma 3.

For any fully connected network f_A with $\rho_\delta \geq 3d$, any probability $0 < \delta \leq 1$ and any error $0 < \varepsilon \leq 1$, Algorithm 1 generates weights $\tilde{A}$ for a network with $\frac{72c^2d^2\log(\mathrm{mdh}/\delta)}{\varepsilon^2} \cdot \sum_{i=1}^d \frac{1}{\mu_i^2 \mu_{i\to}^2}$ total parameters such that with probability $1 - \delta/2$ over the generated weights $\tilde{A}$, for any $x \in S$:

$$\|f_A(x) - f_{\tilde{A}}(x)\| \leq \varepsilon \|f_A(x)\| .$$

where $\mu_i, \mu_{i\to}, c$ and ρ_δ are layer cushion, interlayer cushion, activation contraction and interlayer smoothness defined in Definitions A.6, A.7, A.8, A.9, respectively.

Some obvious improvements

(i) Empirically it has been observed that deep net training introduces fairly small changes to parameters as compared to the (random) initial weights Dziugaite and Roy [2017]. We can exploit this by incorporating the random initial weights into the helper string and do the entire proof above not with the layer matrices A^i but only the difference from the initial starting point. Experiments in Section 6 of the original paper A show this improves the bounds.

(ii) Cushions and other quantities defined earlier are data-dependent, and required to hold for the entire training set. However, the proofs go through if we remove say ζ fraction of outliers that violate the definitions; this allows us to use more favorable values for cushion etc. and lose an additive factor ζ in the generalization error.

A.12 Definition 8

(Interlayer Cushion, Convolution Setting). For any two layers $i \leq j$, we define the interlayer cushion $\mu_{i,j}$ as the largest number such that for any $x \in S$:

$$\mu_{i,j} \cdot \frac{1}{\sqrt{n_1^i n_2^i}} \left\| J_{x^i}^{i,j} \right\|_F \left\| x^i \right\| \leq \left\| J_{x^i}^{i,j} x^i \right\|$$

Furthermore, for any layer i we define the minimal interlayer cushion as $\mu_{i\to} = \min_{i \leq j \leq d} \mu_{i,j} = \min\left\{1/\sqrt{h^i}, \min_{i<j\leq d} \mu_{i,j}\right\}$.

A.13 Definition 9

(Well-distributed Jacobian). Let $J_x^{i,j}$ be the Jacobian of $M^{i,j}$ at x, we know $J_x^{i,j} \in \mathbb{R}^{h^i \times n_1^i \times n_2^i \times h^j \times n_1^j \times n_2^j}$. We say the Jacobian is β well-distributed if for any $x \in S$, any i, j, any $(a, b) \in \left[n_1^i \times n_2^i\right]$

$$\left\| \left[J_x^{i,j}\right]_{:,a,b,:,:,:} \right\|_F \leq \frac{\beta}{\sqrt{n_1^i n_2^i}} \left\| J_x^{i,j} \right\|_F$$

A.14

B Preliminaries for Generalization Error Bounds

B.1 Lemma 2. Contraction

Let function $\phi : \mathbb{R} \to \mathbb{R}$ be Lipschitz with constant $\mathcal{L}_\phi$ such that ϕ satisfies $\phi(0) = 0$. Then for any class $\mathcal{F}$ of functions mapping from $\mathcal{X}$ to $\mathbb{R}$ and any set $S = \{x_1, \ldots, x_m\}$:

$$(\mathbb{E}_{\xi\in\{\pm1\}^m}\left[\frac{1}{m}\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{m}\xi_i\phi(f(x_i))\right|\right] \leq 2\mathcal{L}_\phi\mathbb{E}_{\xi\in\{\pm1\}^m}\left[\frac{1}{m}\sup_{f\in\mathcal{F}}\left|\sum_{i=1}^{m}\xi_i f(x_i))\right|\right]$$

B.2 Lemma 3

A Rademacher complexity analysis showed if parameter matrices $W_1, \ldots, W_d$ for each d layers have Frobenius norm $\|\cdot\|_F$ upper-bounded by $(M_F(1), \ldots, M_F(d)$, the generalization error is bounded by:

$$\mathcal{O}\left(\frac{B2^d\prod_{j=1}^{d}M_F(j)}{\sqrt{m}}\right)$$

B.3 Lemma 4

Applying contraction to a different object in the Rademacher complexity analysis changes the exponential depth-dependence found in Lemma 4 into polynomial depth-dependence. Further improving the generalization error bounds to:

$$\mathcal{O}\left(\frac{B\sqrt{d}\prod_{j=1}^{d}M_F(j)}{\sqrt{m}}\right)$$