GENERALIZATION IN DEEP LEARNING: FRAMEWORKS AND BOUND ESTIMATION METHODS

Ziwei Su
Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL 60208, USA
{ziwei.su}@northwestern.edu

ABSTRACT

The generalization performance of heavily over-parameterized deep neural networks have been puzzling machine learning researchers since day one of deep learning age. PAC-Bayes framework is the most popular way to understand deep learning generalization, while novel complexity measures and methods to derive generalization bounds have also been proposed. Alternative approaches to understand deep learning generalization challenges the power of PAC-Bayes, and empirical studies are useful in validate new complexity measures. Beginning with Jiang et al. (2021), we introduce the recent progress in generalization in deep learning, and reproduce parts of the results in Jiang et al. (2021), confirming its correctness.

1 INTRODUCTION

The generalization performance of deep neural networks has been attracting widespread theoretical and empirical research attention, as substantially over-parameterized deep neural networks achieve low test error, which defies conventional wisdom that over-parameterized machine learning models have poor generalization properties. This phenomenon still remains a mystery and requires out-of-the-box thinking, as complexity measures based on number of parameters, such as VC-dimension, struggles to explain the generalization of over-parameterized deep neural networks. Therefore, there is a growing need for novel theoretical frameworks, effective complexity measures, and better generalization bounds.

Many of the existing works lie in the area of PAC-learning, with PAC-Bayes theory being the most popular approach to compute nonvacuous generalization error bounds, while several complexity measures were introduced to address generalization. Several large empirical studies also provided a better understanding of generalization, in terms of generalization behaviors and effectiveness of various complexity measures. Beyond PAC-Bayes, there have been alternative perspectives, including kernel learning and distributional robustness. Several novel methods were also introduced to derive generalization bounds.

In this review, we introduce recent progress in deep neural network generalization, with focuses on PAC-Bayes frameworks with its alternatives, and generalization bound estimation methods, in section 2 and 3, respectively. Section 4 presents the contributions of and the implementation of its experiments. The codes for generating the experimental results are included in the separate file submitted.

2 GENERALIZATION FRAMEWORKS: PAC-LEARNING, ITS VARIETIES AND ADVERSARIES

The most popular framework in the studies of neural network generalization is PAC-learning. To begin with, we introduce its basic ideas. The following mathematical formulation is based on the lecture notes of Gormley (2016), from 10-601 Introduction to Machine Learning, Carnegie Mellon University.

2.1 BASIC IDEAS OF PAC-LEARNING

Suppose we have a classification task, with labeled (training) data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m), \mathbf{x}_i \in X, y_i \in Y, i = 1, 2, \ldots, m$, the goal of learning is to choose the 'best' hypothesis $h \in \mathcal{H} : X \to Y$ that minimizes the true error (expected risk) $R(h)$:

$$R(h) = P_{\mathbf{x},y\sim\mu}\left(h(\mathbf{x}) \neq y\right) \tag{1}$$

where $\mu(\cdot)$ is some unknown distribution, and $y \in Y$ is the true label for any given $x \in X$.

In practice, we only have access to the empirical risk $\hat{R}(h)$ given training data:

$$\hat{R}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}\left(h\left(\mathbf{x}_i\right) \neq y_i\right) \tag{2}$$

Therefore, our goal now is to generate meaningful bounds for $R(h)$ given $\hat{R}(h)$.

PAC stands for *Probably Approximately Correct*. PAC criterion is that the learner produces a high accuracy learner with a high probability:

$$P(|R(h) - \hat{R}(h)| \leq \epsilon) \geq 1 - \delta \tag{3}$$

A learner is called consistent if for every ϵ and δ, there exists a positive number of training samples n such that for any distribution $p(\cdot)$,

$$P(|R(h) - \hat{R}(h)| > \epsilon) < \delta \tag{4}$$

The sample complexity is the minimum value of n for which this statement holds. If n is finite, then $\mathcal{H}$ is said to be learnable. If n is a polynomial function of $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$, then $\mathcal{H}$ is said to be PAC learnable.

2.2 UNDERSTANDING DEEP NEURAL NETWORK GENERALIZATION WITH PAC-LEARNING: TRENDS AND ALTERNATIVES

Under the PAC-learning framework, Jiang et al. (2021) points out that generalization can be regarded as bounding the size of the search space of a learning algorithm. There has been a wide variety of studies in deep neural network generalization from this perspective.

2.2.1 PAC-BAYES THEORY AND ITS APPLICATIONS

Developed by McAllester (2003), PAC-Bayes theory was originally intended to explain Bayesian learning in a learning theory perspective, but it has now been applied extensively into deep neural network settings. The following mathematical formulation is based on the lecture notes of Dvijotham (2011), from the Winter 2011 CSE 522 Learning Theory, University of Washington.

Suppose $\mathcal{P}$ is the space of probability distributions on $\mathcal{H}$, and $\mathcal{D}_m$ is our dataset of size m. We now introduce the main theorem of PAC-Bayes theory.

Theorem 1 (PAC-Bayes)**.** *For every $\delta > 0, m \in \mathbb{N}$, distribution μ on $\mathbb{R}^k \times \{-1, 1\}$, and distribution $P \in \mathcal{P}$, with probability at least $1 - \delta$ over $D_m \sim \mu^m$, for all distributions $Q \in \mathcal{P}$,*

$$\mathrm{KL}\left(\hat{e}\left(Q, D_m\right) \| e(Q)\right) \leq \frac{\mathrm{KL}(Q\|P) + \log\frac{m}{\delta}}{m-1} \tag{5}$$

where $\hat{e}\left(Q, D_m\right) = \mathop{\mathbb{E}}_{w\sim Q}\left[\hat{R}\left(h_w, D_m\right)\right]$ is the expected empirical error under Q, $e(Q) = \mathop{\mathbb{E}}_{w\sim Q}\left[R\left(h_w\right)\right]$ is the expected error for Q on $\mathcal{H}$.

This bound leads to the following learning algorithm:

- Fix $\delta > 0$, choose a prior distribution $P \in \mathcal{P}$ before seeing any data.
- Observe data D_m and choose posterior $Q \in \mathcal{P}$ that minimizes the error bound

$$\mathrm{KL}^{-1}\left(\hat{e}\left(Q, D_m\right) \mid \frac{\mathrm{KL}(Q\|P) + \log\frac{m}{\delta}}{m-1}\right) \tag{6}$$

- Return the randomized classifier given by Q.

Inspired by Langford & Caruana (2002), who used PAC-Bayes bounds to compute nonvacuous numerical bounds on generalization error for stochastic two-layer two-hidden-unit neural networks, Dziugaite & Roy (2017) pioneered in optimizing PAC-Bayes bound to compute nonvacuous numerical bounds on the generalization error of deep stochastic neural networks. Based on the hypothesis that SGD finds good solutions only when surrounded by relatively large volume of nearly-as-good solutions, the expected error rate of a classifier drawn at random from this volume should match that of the SGD solution. Using theorem 1, the expected error rate of a classifier chosen from a distribution Q is bounded in terms of Kullback-Leibler divergence from some a priori fixed distribution P, hence if the volume of equally good solutions is large, and not too far from the mass of P, a nonvacuous bound can be obtained.

Dziugaite & Roy (2017) used SGD to find a broad distribution Q over neural network parameters that minimizes the PAC-Bayes bound on the error rate of a (stochastic) neural network, in effect mapping out the volume of equally good solutions surrounding the SGD sulution. At each step, they updated the network weights and variances by taking a step along an unbiased estimate if the gradient of an upper bound on the PAC-Bayes bound.

Inspired by Dziugaite & Roy (2017), follow-up studies use PAC-Bayes analysis in the search for appropriate complexity measures. One worth noting complexity measure, proposed by Keskar et al. (2016), is 'sharpness', which could be understood as the robustness of the training error to perturbations in the parameters. The following mathematical formulation of sharpness of minima is based on Keskar et al. (2016).

Given a matrix $A \in \mathbb{R}^{n\times p}$, whose columns are randomly generated, and let $\mathcal{C}_\epsilon$ denote a box around the solution over which the maximization of f is performed, it is defined as follows:

$$\mathcal{C}_\epsilon = \left\{z \in \mathbb{R}^p : -\epsilon\left(\left|\left(A^+x\right)_i\right| + 1\right) \le z_i \le \epsilon\left(\left|\left(A^+x\right)_i\right| + 1\right) \quad \forall i \in \{1, 2, \cdots, p\}\right\}, \tag{7}$$

where A^+ denotes the pseudo-inverse of A. The measure of sharpness (or sensitivity) is defined as:

Definition 1. *Given $x \in \mathbb{R}^n, \epsilon > 0$ and $A \in \mathbb{R}^{n\times p}$, we define the $(\mathcal{C}_\epsilon, A)$-sharpness of f at x as:*

$$\phi_{x,f}(\epsilon, A) := \frac{(\max_{y\in\mathcal{C}_\epsilon} f(x + Ay)) - f(x)}{1 + f(x)} \times 100. \tag{8}$$

Several other complexity measures have been introduced to address deep learning generalization. A margin-based multiclass generalization bound for neural networks that scales with their margin-normalized spectral complexity, namely Lipschitz constant, was introduced by Bartlett et al. (2017). Lipschitz constant is defined as the product of the spectral norms of the weight matrices, times a certain correction factor.

Neyshabur et al. (2017a) investigated several different complexity measures besides Lipschitz constant, including VC-dimension, l_1, l_2 and spectral norm, and sharpness, based on their ability to theoretically guarantee generalization, and their empirical ability in explaining deep neural network generalization, while discussed how sharpness itself is not sufficient for ensuring generalization, but can be combined with PAC-Bayes analysis, with the norm of weights, to obtain appropriate complexity measure. However, their proposed PAC-Bayes measure are not sufficient to explain all the generalization behaviours observed in larger neural networks.

In their following work, Neyshabur et al. (2017b) proposed another generalization bound for feed-forward neural networks with ReLU activations in terms of the product of the spectral norm of the layers and the Frobenius norm of the weights. The sharpness of the network is bounded by a bound on the changes in the output of a network with respect to perturbation of its weights, and

the generalization bound is derived through combining this perturbation bound and the PAC-Bayes analysis.

The studies we introduced so far are providing generalization guarantees for neural networks with stochastic parameters. Nagarajan & Kolter (2019a) presented a general PAC-Bayesian framework to provide a bound on the original network that is deterministic and uncompressed, leveraging the property of SGD that it finds solutions that lie in flat, wide minima in training loss, minima where the output of the network is resilient to small random noise added to its parameters. The novelty in this approach is that the framework enables to show that if on training data, the interactions between the weight matrices satisfy certain conditions that imply a wide training loss minimum, these conditions generalize to the interactions between the matrices on test data, implying a wide test loss minimum. The framewoerk is applied in a setup where it is assumed that the pre-activation values of the network are not too small, and a generalization guarantee that does not scale with the product of spectral norms of the weight matrices could be provided.

Near simultaneously, Nagarajan & Kolter (2019b) introduced a notion of effective model capacity that is dependent on a given random initialization of the network. The distance of the learned network from its initialization is implicitly regularized by SGD to a width-independent value, and such distances alone could not explain generalization.

Recently, distinct from the previous norm-based, PAC-Bayes based, or margin-based analysis, Natekar & Sharma (2020) utilized neuroscientific theories on human visual systems to come up with an interpretation of generalization from the perspective of quality of internal representations of deep neural networks. They provided practical complexity measures that could be computed post-hoc, based on Davies Bouldin index and separability of representations.

In terms of examining different complexity measures, large-scale theoretical empirical studies stand out. Jiang et al. (2019) presented such a study of more than 40 generalization measures. They found that many norm based measures, in particular spectral norm based ones similar to Bartlett et al., performs badly, even has negative correlation with generalization, while sharpness-based measures like PAC-Bayes ones perform best. They also suggested that more rigourous methods are needed to study complexity measures that capture spurious correlations that do not provide casual insights into generalization, and measures related to optimization procedures, like gradient noise, are potentially predictive towards generalization.

2.2.2 ALTERNATIVES OF PAC-BAYES

However, despite its popularity, several works have questioned PAC-Bayes' power in understanding generalization, and proposed alternative perspectives in understanding generalization in deep learning.

Belkin et al. (2018) considered generalization of deep neural networks in a kernel learning perspective, drawing comparisons with classical kernel methods. They argued that good generalization performance of overfitted classifer is not unique of deep learning, as empircial results suggests overfit kernel machines and non-smooth Laplacian kernels fit test data well.

Nagarajan & Kolter (2019c) examined, in practice, weight norms of deep ReLU networks, such as distance from initialization, increase (polynomially) with number of training examples, hence existing generalization bounds, depending on such norms, fail to reflect a dependence on the number of training samples, or even grow with number of training samples. In fact, uniform convergence bounds are problematic in over-parameterized deep neural network settings, except under explicit regularization.

Within the framework of distributional robustness, Dziugaite et al. (2020) proposed a follow-up study, finding that although measures are good at robustly predicting changes due to training set size, no current existing complexity measures, including PAC-Bayes ones, has better robust sign-error than a coin flip. Robustly predicting changes due to width and depth is hard for all measures, and norm-based measures outperform other measures at learning rate interventions.

3 DERIVING GENERALIZATION BOUNDS

There has been exciting developments in new unconventional ways to derive generalization bounds. Negrea et al. (2020) studied the generalization error of a learned predictor $\hat{h}$ in terms of that of a surrogate (potentially randomized) predictor that is coupled to $\hat{h}$ and designed to trade empirical risk for control of generalization error. To address the problem with uniform convergence argued by Nagarajan & Kolter (2019c) that it does not explain generalization in examples that are emblematic of the modern interpolating regime, they defined an alternative notion of uniform convergence for sequences of learning problems, and studied the generalization error of learning algorithms, including interpolating ones. They also introduced a generic technique that introduces surrogate learning algorithms via conditioning, naturally trading empirical risk for generalization error relative to the original learning problem.

Zhou et al. (2020) used an underdetermined noisy linear regression model where the minimum-norm interpolating predictor is known to be consistent, and showed that uniformly bounding the difference between empirical and population errors cannot show any learning in the norm ball and consistency for any set. However, the consistency of the minimal-norm interpolator with a slightly weaker yet standard notion of uniform convergence of zero-error predictors in a norm ball could be proved and used to bound the generalization error of low-norm interpolating predictors.

Garg et al. (2021) leveraged unlabeled data to produce generalization bounds, augmenting labeled training set with randomly labeled data, and train in standard fashion. However, according to Jiang et al. (2021), the augmentation and the performing of a careful early stopping in their training makes their bound inapplicable to interpolating networks, providing vacuous bounds. We will discuss the inspired work of Jiang et al. (2021). in detail in section 4.

4 ASSESSING GENERALIZATION OF SGD VIA DISAGREEMENT

Recently, Jiang et al. (2021) empirically showed that test error of deep networks could be estimated by training the same architecture on the same training set but with a different run of SGD, and measuring the disagreement rate between the two networks on unlabeled test data. A weaker observation was made by Nakkiran & Bansal (2020) that the disagreement rate nearly equals to the test error, by investigating across various models including neural networks, kernal SVMs, and decision trees. By observing on SVHN, CIFAR-10/100 datasets, with variants of Residual Networks and Convolutional Networks, the stronger result was observed. The stronger result also holds on many kinds of out-of-distribution data in the PACS dataset.

This result has provided us a useful tool in estimating test accuracy, as it only requires an unlabeled dataset, and no correlation constants are required. It is also robust to certain kinds of distribution shift.

Theoretically speaking, Jiang et al. (2021) found that if the ensemble of networks learned from different stochastic runs of the training algorithm is well calibrated, then the disagreement rate equals the test error in expectation, hence establishing a new connection between generalization and calibration.

Mathematically speaking, for any stochastic learning algorithm, if the algorithm leads to a well-calibrated ensemble, then the ensemble satisfies the following *Generalization Disagreement Equality* (GDE) *in expectation over the stochasticity*:

Definition 2. *Let $h : \mathcal{X} \to [K]$ denote a hypothesis from a hypothesis space $\mathcal{H}$, where $[K]$ denotes the set of K labels $\{0, 1, \ldots, K-1\}$. Let D be a distribution over $\mathcal{X} \times [K]$. The stochastic algorithm $\mathcal{A}$ satisfies the Generalization Disagreement Equality (GDE) on D if*

$$\mathbb{E}_{h,h' \sim \mathcal{H}_{\mathcal{A}}} \left[\mathrm{Dis}_D \left(h, h' \right) \right] = \mathbb{E}_{h \sim \mathcal{H}_{\mathcal{A}}} [\textit{TestErr}_{\ D}(h)]. \tag{9}$$

4.1 EXPERIMENTS

In Jiang et al. (2021), the main observations were reported on variants of Residual Networks, convolutional neural networks, and fully connected networks trained with momentum SGD on CIFAR-10/100 and SVHN, to come up with the GDE. As the experiments using different architectures and

datasets are in similar manner and produced similar results, we seek to only reproduce the results for training Residual Networks on CIFAR-10 to save time and computing resources without loss of generality. The corresponding result is displayed in Figure 1 of Jiang et al. (2021).

The ResNet18 hyperparameter configurations in are as following:

1. width multiplier: $\{1\times, 2\times\}$
2. initial learning rate: $\{0.1, 0.05\}$
3. weight decay: $\{0.0, 0.0001\}$
4. minibatch size: $\{100, 200\}$
5. data augmentation: $\{$ No, Yes $\}$

A GTX-1080Ti GPU is used for training purposes. However, as there are $2^5 = 32$ configurations of hyperparameters in total, and training each ResNet costs approximately 2 hours, we only used the following $2^3 = 8$ hyperparameter configurations:

1. width multiplier: $\{1\}$
2. initial learning rate: $\{0.1, 0.05\}$
3. weight decay: $\{0.0001\}$
4. minibatch size: $\{100, 200\}$
5. data augmentation: $\{$ No, Yes $\}$

For each configuration, we train a pair of networks on the same dataset (CIFAR-10). Our setting is the same as the **DiffOrder** in Jiang et al. (2021), models share the same initialization and see the same data, but in different orders. Besides, in accordance with , all models are trained with SGD with momentum of 0.9, the learning rate decays $10\times$ every 50 epochs, and the training stops when the training accuracy reaches 100%.

The codes for implementing ResNet18 on CIFAR-10 is based on the open-source code from Joo et al. (2020). The results are as follows.

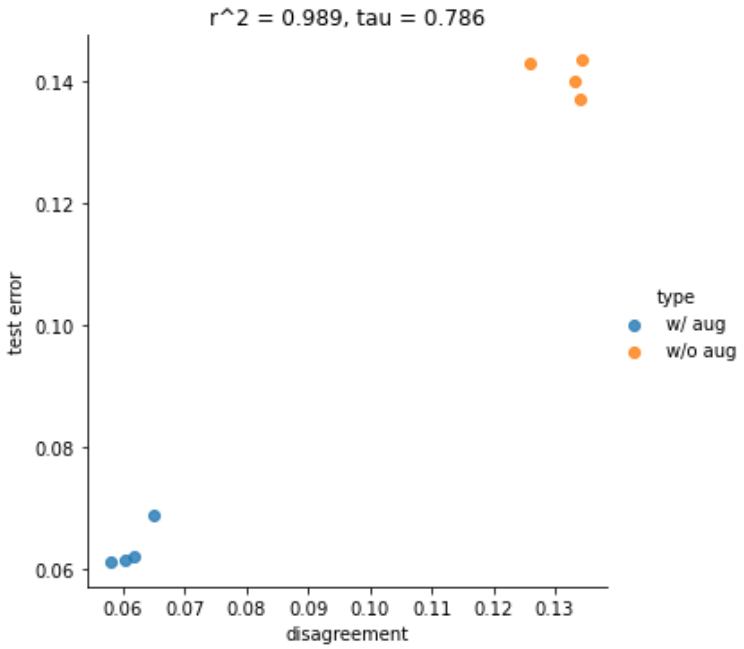

Figure 1: GDE on CIFAR-10: r^2 coefficient and Kendall's Ranking coefficient (*tau*) is reported on top.

	test error	disagreement	type	init learning rate	minibatch size
0	0.0613	0.0579	w/ aug	0.10	100
1	0.1400	0.1333	w/o aug	0.10	100
2	0.0616	0.0604	w/ aug	0.05	100
3	0.1370	0.1342	w/o aug	0.05	100
4	0.0621	0.0617	w/ aug	0.10	200
5	0.1434	0.1345	w/o aug	0.10	200
6	0.0688	0.0649	w/ aug	0.05	200
7	0.1428	0.1258	w/o aug	0.05	200

Figure 2: Observations

It is worth noting that same as the **DiffOrder** result in Figure 1 of , our observations generally lie above the line $y = x$. In conclusion, our result matches that of Jiang et al. (2021).

5 Conclusion

In this review, we have gone through a wide range of ideas in deep learning and SGD generalization, including PAC-Bayes framework, various complexity measures, ways to derive generalization bounds, and alternative viewpoints for understanding deep learning generalization, starting from Jiang et al. (2021). Our numerical experiment successfully (although with less observations and under only ResNet18 on CIFAR-10) replicates the empirical results in Jiang et al. (2021), proving its correctness.

From our current viewpoint, there are many interesting questions to be answered: how could we further leverage (unlabeled) data to estimate generalization? What's the exact role SGD plays in generalization, in terms of 'implicit regularization'? Are there better complexity measures? Why different sources of stochasticity have a similar effect on calibration? As the author was recently inducted to calibration, machine learning calibration would be an interesting topic to further explore in independent studies and further in PhD research.

It is worth noting that this work is a very brief summary of relevant research in deep learning generalization, as the author is totally new to this area, but it could serve as a useful cornerstone for further exploration into the interplay between machine learning, optimization and calibration.

Acknowledgments

The author would like to thank Taejong Joo, a fellow first-year IEMS PhD for valuable discussions in the course of implementing ResNet18 on CIFAR-10. The base open-source code is from his github site: `https://tjoo512.github.io/`.

References

Peter Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1706.08498*, 2017.

Mikhail Belkin, Siyuan Ma, and Soumik Mandal. To understand deep learning we need to understand kernel learning. In *International Conference on Machine Learning*, pp. 541–549. PMLR, 2018.

Krishnamurthy Dvijotham. Pac-bayes analysis, 2011. URL `https://courses.cs.washington.edu/courses/cse522/11wi/scribes/lecture13.pdf`.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M Roy. In search of robust measures of generalization. *arXiv preprint arXiv:2010.11924*, 2020.

Saurabh Garg, Sivaraman Balakrishnan, J Zico Kolter, and Zachary C Lipton. Ratt: Leveraging unlabeled data to guarantee generalization. *arXiv preprint arXiv:2105.00303*, 2021.

Matt Gormley. Pac learning, 2016. URL `https://www.cs.cmu.edu/~mgormley/courses/10601-s17/slides/lecture28-pac.pdf`.

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.

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J Zico Kolter. Assessing generalization of sgd via disagreement. *arXiv preprint arXiv:2106.13799*, 2021.

Taejong Joo, Uijung Chung, and Min-Gwan Seo. Being bayesian about categorical probability. In *International Conference on Machine Learning*, pp. 4950–4961. PMLR, 2020.

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.

John Langford and Rich Caruana. (not) bounding the true error. *Advances in Neural Information Processing Systems*, 2:809–816, 2002.

David McAllester. Simplified pac-bayesian margin bounds. In *Learning theory and Kernel machines*, pp. 203–215. Springer, 2003.

Vaishnavh Nagarajan and J Zico Kolter. Deterministic pac-bayesian generalization bounds for deep networks via generalizing noise-resilience. *arXiv preprint arXiv:1905.13344*, 2019a.

Vaishnavh Nagarajan and J Zico Kolter. Generalization in deep networks: The role of distance from initialization. *arXiv preprint arXiv:1901.01672*, 2019b.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *arXiv preprint arXiv:1902.04742*, 2019c.

Preetum Nakkiran and Yamini Bansal. Distributional generalization: A new kind of generalization. *arXiv preprint arXiv:2009.08092*, 2020.

Parth Natekar and Manik Sharma. Representation based complexity measures for predicting generalization in deep learning. *arXiv preprint arXiv:2012.02775*, 2020.

Jeffrey Negrea, Gintare Karolina Dziugaite, and Daniel Roy. In defense of uniform convergence: Generalization via derandomization with an application to interpolating predictors. In *International Conference on Machine Learning*, pp. 7263–7272. PMLR, 2020.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv preprint arXiv:1706.08947*, 2017a.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017b.

Lijia Zhou, Danica J Sutherland, and Nathan Srebro. On uniform convergence and low-norm interpolation learning. *arXiv preprint arXiv:2006.05942*, 2020.

A Appendix

A.1 Instructions on code implementation

Please first make sure you have properly installed cuda on your device. The requirements.txt lists all the requirements for this implementation. You can install all the requirements by running 'pip3 install -r requirements.txt' in the prompt under your code folder.

To train a ResNet18 on CIFAR-10, use the following command (in windows enviroment):

```
python cifar_trainer.py --arch resnet18 --coeff -1.0
--dataset cifar10 --save-dir folder_name --gpu 0
```

To check test error and disagreement rate, please run the following command:

```
.\eval.py --path1 folder_path_of_model_1
--path2 folder_path_of_model_2 --gpu 0 --output_name test_name
```

It is worth noting that the first value displayed after running the command is the test accuracy in %.