Generalization Bounds for Deep Learning: From VC-Dimension to Sharpness

Tyler LaBonte

July 8, 2022

Abstract. We summarize work on generalization bounds for deep learning. We aim to give a brief, concise overview of current research directions and how they relate to each other and to classical learning theory. We also provide a "plain English" Frequently Asked Questions (F.A.Q.).

Contents

1 Introduction **2**

2 VC-Dimension **3**
2.1 Classical Learning 3
2.2 Deep Learning 4

3 Rademacher Complexity **6**
3.1 Classical Learning 6
3.2 Deep Learning 7

4 Margin Theory **9**
4.1 Classical Learning 9
4.2 Deep Learning 11
4.3 Modern Deep Learning 12

5 PAC-Bayes and Sharpness **14**
5.1 Classical Learning 14
5.2 Modern Deep Learning 16

6 Other Approaches **19**
6.1 Algorithmic Stability 19
6.2 Compression and Distillation 22
6.3 Distance from Initialization 25
6.4 Neural Tangent Kernel 26
6.5 Information Theory 27
6.6 *A Posteriori* Bounds 28

7 Bibliography **30**

A Frequently Asked Questions (F.A.Q.) **39**

B Notation **41**

1 Introduction

Deep learning has revolutionized prediction capabilities in a vast number of data-rich domains, leading to much-publicized deployments in high-consequence applications such as autonomous driving [Sun et al., 2020] and nuclear fusion reactors [Degrave et al., 2022]. Yet, our fundamental understanding of deep learning lags behind its meteoric rise in practical applications. The performance of deep models is not captured by classical learning theory, and we lack a framework – even a heuristic – for predicting the behavior of neural networks with respect to architecture, optimizer, and data distribution. As a result, beneath their advertised successes, deep neural networks are brittle, black-box systems prone to unexpected failures and susceptible to simple attacks [Szegedy et al., 2014, Evtimov et al., 2018].

The most apparent contradiction of deep learning is that overparameterized models – those with many more parameters than data – achieve low generalization error. According to existing theory, overparameterized models should overfit the data and generalize poorly, but in fact they become highly accurate after crossing the interpolation threshold [Belkin et al., 2019]. Indeed, parameter-counting is a naïve way to predict the generalization of deep models; the modern regime demands a more sophisticated method. This phenomenon leads to unexpected results in practice because model size and training performance are not indicative of generalization – thus, it is impossible for practitioners to confidently predict model behavior in high-consequence scenarios. New theory is required to understand the generalization of deep learning, and while there have been nontrivial advances in the last few years, the field remains largely open.

The most fundamental type of result in learning theory is the *generalization bound*, an upper bound on the deviation of the generalization error of a particular model from its empirical error. The key ingredient in a generalization bound is one or more *complexity measures*, properties of the trained model which determine its generalization performance. The most successful classical complexity measures result in vacuous generalization bounds for deep learning, and while promising directions exist, we are far from a complete theory. In particular, generalization in deep learning seems intricately related to subtle interactions between the data, model, and optimizer, which can make complexity measures difficult to identify and prove.

In this work, we survey major generalization bounds developed in the last few decades and summarize current research directions in the area. We emphasize the paradigm shift from classical learning theory to deep learning by understanding in a technical sense why classical complexity measures are insufficient for deep learning and which novel directions are promising. Progress in this area would have important practical ramifications by guaranteeing generalization performance and could be used to improve model robustness, fairness, and domain adapation, ultimately enabling the safe and trustworthy deployment of deep learning in high-consequence applications.

Notation. Suppose $\mathcal{X}$ is the input space and $\mathcal{D}$ is the data distribution. We receive an i.i.d. training sample $S \coloneqq \{x_i\}_{i=1}^m$ drawn from $\mathcal{D}$ with $x_i \in \mathcal{X}$ for all $1 \leq i \leq m$. Suppose there exists a labeling function $f^\star : \mathcal{X} \to \{-1, 1\}$; we denote $y \coloneqq f^\star(x)$ when it is clear. We consider a fixed hypothesis class $\mathcal{H}$, and the task is to use S to select $h \in \mathcal{H}$ with small generalization error $L(h) \coloneqq \mathbb{E}_{x\sim\mathcal{D}}[\mathbb{1}_{h(x)\neq y}]$. The empirical error is $\widehat{L}_S(h) \coloneqq \frac{1}{m}\sum_{i=1}^m \mathbb{1}_{h(x_i)\neq y}$. Note that $L(h) = \mathbb{E}_{S\sim\mathcal{D}^m}[\widehat{L}_S(h)]$.

A feedforward neural network is a directed acyclic graph composed of an input layer, one or more hidden layers, and an output layer. It is called fully-connected if it contains all possible edges between each consecutive layer. Each edge i has weight w_i and each node r has activation function ϕ_r and threshold θ_r. Suppose node r receives inputs $z_1, z_2, \ldots, z_k$ from the previous layer, then the output of node r is

$$\phi_r\Big(\sum_{i=1}^{k} w_i z_i - \theta_r\Big). \tag{1}$$

Often we will take a functional perspective. Let $f_{\boldsymbol{w}}$ be the function computed by a neural network with parameters $\boldsymbol{w} = \text{vec}\{\mathbf{W}_i\}_{i=1}^{D}$. Then if all nodes have the same activation ϕ,

$$f_{\boldsymbol{w}}(\boldsymbol{x}) \coloneqq \mathbf{W}_D\phi(\mathbf{W}_{D-1}\phi(\cdots\phi(\mathbf{W}_1\boldsymbol{x}))). \tag{2}$$

The number of parameters W is the number of edges plus the number of nodes, excluding input nodes. The depth D is the number of hidden layers plus one (representing the output layer). The width H is the number of nodes in the largest hidden layer. We use $\mathcal{F}$ to denote the hypothesis class of functions computed by a neural network to distinguish it from general classes $\mathcal{H}$.

Related work. The structure and arguments of this work are inspired by [Tewari and Bartlett, 2014], [Neyshabur et al., 2017], and the appendix of [Jiang et al., 2020]. Section 2.1 and Section 3.1 are based on Chapter 3 of [Mohri et al., 2018]; Section 4.1 is based on Chapter 5. Other excellent resources include [Hardt and Recht, 2021, Telgarsky, 2021]. For a survey of PAC-Bayes theory, see [Guedj, 2019, Alquier, 2021]. A related survey preprint on generalization bounds in deep learning is [Valle-Pérez and Louis, 2020]. They propose seven desiderata for generalization bounds and evaluate existing works, as well as propose their own bound. In contrast, we provide a more concise, objective survey with a greater focus on classical results.

2 VC-Dimension

2.1 Classical Learning

The Vapnik-Chervonenkis dimension (VC-dimension) [Vapnik and Chervonenkis, 1971] is a combinatorial complexity measure which is particularly important in the probably approximately correct (PAC) learning framework [Valiant, 1984]; a hypothesis class is PAC-learnable if and only if it has finite VC-dimension. Essentially, the VC-dimension of a hypothesis class $\mathcal{H}$ is the maximum number of points which hypotheses in $\mathcal{H}$ can classify in all possible ways.

Formally, define the restriction of $\mathcal{H}$ to a set $X \subseteq \mathcal{X}$ of size m as

$$\mathcal{H}_{|X} \coloneqq \{(h(x_1), h(x_2), \ldots, h(x_m)) : h \in \mathcal{H}\}. \tag{3}$$

Then, define the growth function $\Pi_{\mathcal{H}} : \mathbb{N} \to \mathbb{N}$ as the maximum number of distinct ways m points can be classified using hypotheses in $\mathcal{H}$:

$$\Pi_{\mathcal{H}}(m) \coloneqq \max_{X \subseteq \mathcal{X}} |\mathcal{H}_{|X}|. \tag{4}$$

A set $X \subseteq \mathcal{X}$ of size m is shattered by $\mathcal{H}$ if $|\mathcal{H}_{|X}| = 2^m$; clearly, this implies $\Pi_{\mathcal{H}}(m) = 2^m$. The VC-dimension of $\mathcal{H}$ is the size of its largest shatterable set.

$$\text{VC-dim}(\mathcal{H}) \coloneqq \max\{m : \Pi_{\mathcal{H}}(m) = 2^m\}. \tag{5}$$

Note that for finite $\mathcal{H}$ we have $\text{VC-dim}(\mathcal{H}) \leq \log_2 |\mathcal{H}|$. An important connection between the growth function and the VC-dimension is illustrated by the Sauer-Shelah Lemma.

Theorem 1 (Sauer-Shelah Lemma, [Sauer, 1972, Shelah, 1972], [Mohri et al., 2018] Thm. 3.17)**.** *Suppose $\mathcal{H}$ is a hypothesis class with* VC-dim$(\mathcal{H}) = d$*. Then, for all $m \in \mathbb{N}$,*

$$\Pi_{\mathcal{H}}(m) \leq \sum_{i=0}^{d} \binom{m}{i}. \tag{6}$$

As a corollary, for all $m \geq d$,

$$\Pi_{\mathcal{H}}(m) \leq \left(\frac{em}{d}\right)^d = \mathcal{O}(m^d). \tag{7}$$

Thus, the growth function has surprising two-tailed behavior – it is exponential in m until $m = d$ and polynomial in m thereafter. From this result, we obtain the following generalization bound; the proof idea is detailed in Section 3.1.

Theorem 2 (VC-dimension Generalization Bound, [Mohri et al., 2018] Cor. 3.19)**.** *Suppose $\mathcal{H}$ is a hypothesis class with* VC-dim$(\mathcal{H}) = d$*. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size m, the following holds for all $h \in \mathcal{H}$:*

$$L(h) \leq \widehat{L}_S(h) + \sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log 1/\delta}{m}}. \tag{8}$$

The VC-dimension is simple to compute for many hypothesis classes. In particular, using Radon's Theorem, it is straightforward to show that the VC-dimension of linear classifiers (halfspaces) in $\mathbb{R}^n$ is $n + 1$.

2.2 Deep Learning

In most VC-dimension bounds for deep feedforward neural networks, the key assumption is on the activation function. More complex activation functions enable a more expressive model, as they introduce additional capacity for variation and nonlinearity. The simplest neural network is the linear threshold network, where each activation is the sign function – essentially a multilayer perceptron. Other classes of neural networks include those with piecewise linear activation functions, such as the rectified linear unit $\mathrm{ReLU}(x) = \max(0, x)$, and those with exponential terms, such as the logistic or sigmoid activation $\sigma(x) = 1/(1 + e^{-x})$.

Contrary to many classical examples, the hypothesis class corresponding to a neural network is only implicitly defined – it is described as the class of functions computed by a neural network with a particular architecture. Thus, it is more convenient to study the computation of these functions rather than their explicit properties. The following lemma bounds the VC-dimension of general functions based on how many arithmetic operations an algorithm uses to compute them.

Lemma 1 (VC-dimension of Arithmetic Operations, [Anthony and Bartlett, 1999] Thm. 8.4)**.** *Suppose $h : \mathbb{R}^W \times \mathbb{R}^n \to \{0, 1\}$ and let*

$$\mathcal{H} = \{\boldsymbol{x} \mapsto h(\boldsymbol{w}, \boldsymbol{x}) : \boldsymbol{w} \in \mathbb{R}^W\} \tag{9}$$

be the hypothesis class determined by h. Suppose that h can be computed by an algorithm that takes as input the pair $(\boldsymbol{w}, \boldsymbol{x})$ and returns $h(\boldsymbol{w}, \boldsymbol{x})$ after no more than t operations of the following types:

1. *the arithmetic operations $+, -, \times,$ and $/$ on real numbers,*

2. *jumps conditioned on $>, \geq, <, \leq, =,$ and $\neq$ comparisons of real numbers, and*

3. *output 0 or 1.*

Then, VC-dim$(\mathcal{H}) \leq 4W(t+2)$.

This immediately gives a bound of VC-dim$(\mathcal{F}) = \mathcal{O}(W^2)$ for $\mathcal{F}$ the class of functions computed by a linear threshold network. By studying linear threshold networks explicitly, we can obtain a better $\mathcal{O}(W \log W)$ bound [Anthony and Bartlett, 1999]. However, such a direct analysis becomes difficult as the activation functions become more complicated. From Lemma 1 we obtain the following theorem on neural networks with piecewise-polynomial activation functions.

Theorem 3 (VC-dimension of Piecewise-Polynomial Neural Networks, [Anthony and Bartlett, 1999] Thm. 8.8)**.** *Suppose $\mathcal{F}$ is the class of functions computed by a feedforward neural network with W parameters, depth D, piecewise-polynomial activation functions with $O(1)$ degree and pieces, and linear threshold output units. Then,*

$$\text{VC-dim}(\mathcal{F}) = \mathcal{O}(WD \log W + WD^2). \tag{10}$$

For piecewise-linear activation functions such as ReLU, a better bound was recently obtained; the techniques are similar to [Anthony and Bartlett, 1999], and the key idea for the refinement is using *average* depth, which captures how the parameters are distributed in the network.

Theorem 4 (VC-dimension of Piecewise-Linear Neural Networks, [Bartlett et al., 2019] Thm. 6)**.** *Suppose $\mathcal{F}$ is the class of functions computed by a feedforward neural network with W parameters, depth D, piecewise-linear activation functions with $O(1)$ pieces, and identity output units. Then,*

$$\text{VC-dim}(\mathcal{F}) = \mathcal{O}(WD \log W). \tag{11}$$

This result is quite close to the lower bound; however, the lower bound cannot be improved with current techniques.

Theorem 5 (Lower Bound on VC-dimension of Neural Networks, [Bartlett et al., 2019] Thm. 3)**.** *Suppose $\mathcal{F}$ is the class of functions computed by a feedforward neural network with W parameters, depth D, and either ReLU or linear threshold activations with identity outputs. Then,*

$$\text{VC-dim}(\mathcal{F}) = \Omega(WD \log(W/D)). \tag{12}$$

We can obtain a bound only a constant factor worse than Theorem 3 for the VC-dimension of neural networks with the sigmoid activation using the following modification of Lemma 1.

Lemma 2 (VC-Dimension of Arithmetic and Exponential Operations, [Anthony and Bartlett, 1999] Thm. 8.14)**.** *Suppose $h : \mathbb{R}^W \times \mathbb{R}^n \to \{0, 1\}$ and let*

$$\mathcal{H} = \{\boldsymbol{x} \mapsto h(\boldsymbol{w}, \boldsymbol{x}) : \boldsymbol{w} \in \mathbb{R}^W\} \tag{13}$$

be the hypothesis class determined by h. Suppose that h can be computed by an algorithm that takes as input the pair $(\boldsymbol{w}, \boldsymbol{x})$ and returns $h(\boldsymbol{w}, \boldsymbol{x})$ after no more than t operations of the following types:

1. *the exponential function $x \mapsto e^x$ on real numbers,*

2. *the arithmetic operations $+, -, \times,$ and $/$ on real numbers,*

3. *jumps conditioned on $>, \geq, <, \leq, =,$ and $\neq$ comparisons of real numbers, and*

4. *output 0 or 1.*

Then, VC-dim$(\mathcal{H}) \leq t^2 W(W + 19\log(9W))$*.*

We can substitute Theorem 4 into Theorem 2 to obtain a generalization bound for piecewise-linear neural networks.

Theorem 6 (VC-dimension Generalization Bound for Piecewise-Linear Neural Networks)**.** *Suppose $\mathcal{F}$ is the class of functions computed by a feedforward neural networks with W parameters, depth D, piecewise-linear activation functions with $\mathcal{O}(1)$ pieces, and identity output units. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size m, the following holds for all $f_{\boldsymbol{w}} \in \mathcal{F}$:*

$$L(f_{\boldsymbol{w}}) \leq \widehat{L}_S(f_{\boldsymbol{w}}) + \mathcal{O}\left(\sqrt{\frac{WD\log W \log \frac{m}{WD\log W}}{m}}\right) + \sqrt{\frac{\log 1/\delta}{m}}. \tag{14}$$

Unfortunately, this bound is vacuous in all but the most trivial cases. Note that the middle term of Theorem 2 is real only when $m > d$ (disregarding constants). Consider an example similar to that of [Dziugate and Roy, 2017] on the simple MNIST dataset, which contains 60,000 28 × 28 black-and-white images of handwritten digits [LeCun et al., 2010]. A feedforward neural network with a single hidden layer of 8 units and ReLU activation has $D = 2$ and $W = (28^2 \times 8 + 8) + (8 \times 10 + 10) = 6{,}370$. By Theorem 5, the VC-dimension (disregarding constants) is lower bounded by $WD\log(W/D) \approx$ 100,000. This value is greater than 60,000, so we obtain no guarantee on the generalization of the model, but larger neural networks can easily achieve a generalization error of less than one percent! Something more sophisticated is required to characterize the generalization of deep learning.

3 Rademacher Complexity

3.1 Classical Learning

A strength and weakness of VC-dimension generalization bounds is that they are distribution- and optimizer-independent. This ensures that the bounds hold in a very broad sense of generality, but as worst-case bounds they are almost inevitably loose when restricting to situations with benign data distributions or optimization properties (as in deep learning).

Rademacher complexity is a more general formulation, closely related to VC-dimension, which is distribution-dependent. Suppose $\boldsymbol{\sigma} = (\sigma_1, \sigma_2, \dots, \sigma_m)$ is a vector of i.i.d. ±1 uniform random variables (also called Rademacher random variables). Then, the empirical Rademacher complexity of a hypothesis class $\mathcal{H}$ with respect to a sample S of size m is

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) \coloneqq \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h(x_i)\right]. \tag{15}$$

The Rademacher complexity of $\mathcal{H}$ on m points is

$$\mathfrak{R}_m(\mathcal{H}) \coloneqq \mathbb{E}_{S\sim\mathcal{D}^m}[\widehat{\mathfrak{R}}_S(\mathcal{H})]. \tag{16}$$

Essentially, Rademacher complexity measures to what extent hypotheses in $\mathcal{H}$ can fit random noise. An application of McDiarmid's Inequality yields the following generalization bound.

Theorem 7 (Rademacher Complexity Generalization Bound, [Mohri et al., 2018] Thm. 3.5)**.** *Suppose $\mathcal{H}$ is a hypothesis class. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size m, each of the following holds for all $h \in \mathcal{H}$:*

$$L(h) \leq \widehat{L}_S(h) + \mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log 1/\delta}{2m}} \tag{17}$$

and

$$L(h) \leq \widehat{L}_S(h) + \widehat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log 2/\delta}{2m}}. \tag{18}$$

Theorem 2 follows from Theorem 7 by Massart's Lemma, which bounds the Rademacher complexity by the growth function. Note that the second bound in Theorem 7 is data-dependent, so one could hypothetically obtain a generalization guarantee by computing the empirical Rademacher complexity on the training set. Unfortunately, doing so is $\mathcal{NP}$-hard in general.

Similarly to VC-dimension, Rademacher complexity can be nicely bounded for simple hypothesis classes. Using Jensen's Inequality, we can bound the empirical Rademacher complexity of linear hypotheses as follows.

Lemma 3 (Rademacher Complexity of Linear Hypotheses with Bounded Weight Vector, [Mohri et al., 2018] Thm. 5.10)**.** *Let $S \subseteq \{\boldsymbol{x} : \|\boldsymbol{x}\| \leq r\}$ be a sample of size m and let $\mathcal{H} = \{\boldsymbol{x} \mapsto \boldsymbol{w}^\top \boldsymbol{x} : \|\boldsymbol{w}\| \leq \Lambda\}$. Then,*

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) \leq \frac{r\Lambda}{\sqrt{m}}. \tag{19}$$

In comparison to the VC-dimension bound for halfspaces, this bound is dimension-free and scales to the overparameterized case (when $d > m$).

3.2 Deep Learning

Though Rademacher complexity is distribution-dependent, it alone is still not strong enough to characterize deep learning. The seminal works of [Neyshabur et al., 2015a, Zhang et al., 2017] showed that industry-standard neural networks easily fit random noise. In particular, architectures such as Inception [Szegedy et al., 2016] and AlexNet [Krizhevsky et al., 2012] (roughly 1.5 million parameters) were able to fit the CIFAR-10 dataset [Krizhevsky, 2009] to 100% training accuracy when the true labels were replaced by random labels, or even when the images themselves were replaced by Gaussian noise. This suggests $\widehat{\mathfrak{R}}_S(\mathcal{H}) \approx 1$ for practical purposes, implying the vacuousity of Theorem 7. In such cases one would often apply explicit regularization (*e.g.,* ℓ_2 regularization) to reduce the effective Rademacher complexity of the hypothesis class, but [Neyshabur et al., 2015a, Zhang et al., 2017] show that explicit regularization is not necessary for generalization on true labels. This suggests that data-dependence by itself is insufficient, and the implicit regularization of the optimizer must also be considered.

[Zhang et al., 2017] also give a sample expressivity-type[1] result which theoretically justifies this empirical finding; it states that a sufficiently large neural network can perfectly fit any sample.

Theorem 8 (Finite Sample Expressivity of Neural Networks, [Zhang et al., 2017] Thm. 1)**.** *Suppose $\mathcal{F}$ is the class of functions computed by a feedforward neural network with ReLU activations with either (1) $D = 2$ and $W = 2m + n$, or (2) $H = \mathcal{O}(m/D)$ and $W = \mathcal{O}(m + n)$. Then, for every sample $S \subseteq \mathbb{R}^n$ of size m and every labeling function $f^\star : S \to \mathbb{R}$, there exists $f_{\boldsymbol{w}} \in \mathcal{F}$ with $f_{\boldsymbol{w}}(\boldsymbol{x}) = f^\star(\boldsymbol{x})$ for all $\boldsymbol{x} \in S$.*

Despite this fact, Rademacher complexity bounds are still useful from a theoretical point of view. In particular, norm-based Rademacher complexity bounds are more sophisticated than parameter-counting VC-dimension bounds and closer in spirit to a perceptron-like analysis.

[1] Sample expressivity is an active field which characterizes the labeling power of neural networks in terms of their depth, width, and parameters. For example, for $k \in \mathbb{N}$, there is a depth $\Theta(k^3)$ network with $\Theta(1)$ nodes for which a depth $O(k)$ network would need $\Omega(2^k)$ nodes to approximate [Telgarsky, 2016]. A full consideration of results of this type is outside the scope of this survey.

An early such result considers neural networks with Lipschitz activation and weight vectors bounded in ℓ_1 norm. It first appeared in [Bartlett and Mendelson, 2002], but we cite the version from [Rakhlin, 2019] for its concision. A closely related paper is [Koltchinskii and Panchenko, 2002].

Theorem 9 (Rademacher Complexity of ℓ_1-bounded Neural Networks, [Bartlett and Mendelson, 2002] Thm. 18, [Rakhlin, 2019])**.** *Suppose $\mathcal{F}$ is the class of functions computed by a feedforward neural network of depth D with 1-Lipschitz activations and inputs $\boldsymbol{x}$ with $\|\boldsymbol{x}\|_\infty \leq 1$. Then for a sample $S \subseteq \mathbb{R}^n$ of size m*[2]*,*

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \mathcal{O}\left(2^D \sqrt{\frac{\log n}{m}} \prod_{i=1}^{D} \|\mathbf{W}_i\|_{1,\infty}\right). \tag{20}$$

Suppose $\mathcal{F}_D$ is the class of functions computed by a feedforward neural network of depth D. The proof uses a "recursive peeling" technique to obtain the lemma $\widehat{\mathfrak{R}}_S(\mathcal{F}_D) \leq 2B_D\widehat{\mathfrak{R}}_S(\mathcal{F}_{D-1})$. It essentially bounds the complexity of a function which is an ℓ_1-bounded linear combination of lower-level functions; the theorem follows from the complexity of $\mathcal{F}_1$.

While this bound depends on the size of the weights rather than the number of nodes, it is still exponential in the depth, which is far from optimal, and $\|\mathbf{W}_i\|_{1,\infty}$ can be large. The next bound is still exponential in the depth, but is dimension-free and uses the Frobenius norm of the weights.

Theorem 10 (Rademacher Complexity of ReLU Neural Networks, [Neyshabur et al., 2015b] Thm. 1, $p = q = 2$)**.** *Suppose $\mathcal{F}$ is the class of functions computed by a feedforward neural network of depth D with ReLU activations and inputs $\boldsymbol{x}$ with $\|\boldsymbol{x}\| \leq 1$. Then for a sample S of size m,*

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \mathcal{O}\left(2^D \frac{1}{\sqrt{m}} \prod_{i=1}^{D} \|\mathbf{W}_i\|_F\right). \tag{21}$$

The same authors propose a generalization bound based on the path norm, an ℓ_1-style measure that is a more sophisticated version of the per-unit norm approach of [Bartlett and Mendelson, 2002]. The path norm is defined as the sum, over all paths $p = \{r_{\text{in}}, r_1, \dots, r_{\text{out}}\}$ from the input nodes to the output node, of the product of the weights along the path: $\sum_p \prod_i w_i$.

Theorem 11 (Path Norm Rademacher Complexity of ReLU Neural Networks, [Neyshabur et al., 2015b] Cor. 7)**.** *Suppose $\mathcal{F}$ is the class of functions computed by a feedforward neural network of depth D with ReLU activations and inputs $\boldsymbol{x}$ with $\|\boldsymbol{x}\| \leq 1$. Then for a sample S of size m,*

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \mathcal{O}\left(2^D \sqrt{\frac{\log n}{m}} \sum_p \prod_i w_i\right). \tag{22}$$

Possibly include [Liang et al., 2019] but I don't understand it.

Subsequent work removed the explicit exponential dependence on the depth by applying the peeling argument in a more sophisticated way. The resultant bounds are independent of (explicit) network depth and width roughly when $m \geq D^2$ and have mild depth dependence otherwise.

Theorem 12 (Size-Independent Rademacher Complexity of Neural Networks, [Golowich et al., 2018] Thm. 1, [Rakhlin, 2019])**.** *Suppose $\mathcal{F}$ is the class of functions computed by a feedforward neural network of depth D and inputs $\boldsymbol{x}$ with $\|\boldsymbol{x}\| \leq 1$. Then for a sample S of size m,*

$$\widehat{\mathfrak{R}}_S(\mathcal{F}) = \widetilde{\mathcal{O}}\left(\min\left(\prod_{i=1}^{D} \|\mathbf{W}_i\|_F \cdot \frac{1}{m^{1/4}}, \prod_{i=1}^{D} \|\mathbf{W}_i\|_F \cdot \sqrt{\frac{D}{m}}\right)\right). \tag{23}$$

[2] To be completely formal, we should denote that $\mathcal{F}$ consists of models where $\|\mathbf{W}_i\|_{1,\infty} \leq B_i$, then replace $\prod_i \|\mathbf{W}_i\|_{1,\infty}$ by $\prod_i B_i$ in the theorem. We eschew this here and onwards to make the dependencies more explicit.

Note that the product of Frobenius norms across the layers can introduce an implicit exponential term in the depth. Lower bounds show that this is unavoidable for Frobenius-based approaches [Bartlett et al., 2017, Golowich et al., 2018], but better bounds are possible by considering other data-dependent quantities. In particular, the margin by which the training data is classified is critically important for generalization in both classical and modern regimes.

4 Margin Theory

4.1 Classical Learning

The concept of margin is fundamental in learning theory, and it succinctly bounds the performance of many classical algorithms. The simplest illustrative example is the perceptron algorithm for binary classification with a halfspace. Suppose there exists a ground-truth unit-length linear classifier $\boldsymbol{w}^\star$ and we are receive datapoints $\boldsymbol{x} \in \mathbb{R}^n$ with $\|\boldsymbol{x}\| \leq 1$ with labels y online. Then, the perceptron algorithm initializes $\boldsymbol{w} = \boldsymbol{0}$, predicts $\operatorname{sgn}(\langle \boldsymbol{w}, \boldsymbol{x} \rangle)$, and updates $\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{x}y$ on a mistake. Define the margin as $\gamma = \min_{\boldsymbol{x}} |\langle \boldsymbol{w}^\star, \boldsymbol{x} \rangle|$, that is, the closest any point can be to the separating halfspace. A straightforward analysis via potential function shows that γ is closely related to the number of mistakes made by the algorithm.

Theorem 13 (Perceptron Mistake Bound)**.** *Suppose there exists $\boldsymbol{w}^\star$ with $\gamma > 0$. Then the number of mistakes made by the perceptron algorithm is at most $1/\gamma^2$, and this is tight.*

This mistake bound is easily converted to a PAC generalization guarantee. A more sophisticated application of the margin is in the support vector machine (SVM), which shows that selecting the maximum-margin hyperplane gives a very strong generalization guarantee. A more general formulation of the margin in $\mathbb{R}^n$ is

$$\gamma = \max_{\boldsymbol{w}, b : y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 0} \min_{i \in [m]} \frac{|\boldsymbol{w}^\top \boldsymbol{x}_i + b|}{\|\boldsymbol{w}\|}. \tag{24}$$

By separability, scaling invariance of $\boldsymbol{w}$ and b, and optimality of the maximizing pair $(\boldsymbol{w}, b)$, we may rewrite

$$\gamma = \max_{\boldsymbol{w}, b : \forall i \in [m] : y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1} \frac{1}{\|\boldsymbol{w}\|}. \tag{25}$$

That is, minimizing $\|\boldsymbol{w}\|$ is exactly equivalent to maximizing the margin. Then, the SVM formulation for the non-separable case under the hinge loss is

$$\begin{array}{lll} \min & \frac{1}{2}\|\boldsymbol{w}\|^2 + C \sum_{i=1}^{m} \varepsilon_i & \\ \text{subject to} & y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \varepsilon_i & \forall i \in [m] \\ & \varepsilon_i \geq 0 & \forall i \in [m] \end{array}$$

for a parameter $C \geq 0$.

A margin-based analysis of SVM gives strong dimension-free bounds which justify selecting the maximum-margin hyperplane. The analysis is based on the confidence margin for hypothesis h with real-valued outputs: $\gamma = yh(x)$. Note this is distinct from the geometric margin introduced earlier. We define the γ-margin loss for $z \in \mathbb{R}$ as

$$\Phi_\gamma(z) = \min\left(1, \max\left(0, 1 - \frac{z}{\gamma}\right)\right). \tag{26}$$

Note that Φ_γ is $1/\gamma$-Lipschitz and penalizes classifying a point correctly but with confidence less than γ. Let $\widehat{L}_S^\gamma(h) = \frac{1}{m}\sum_{i=1}^m \Phi_\gamma(y_i h(x_i))$. Using Talagrand's Lemma, which bounds the Rademacher complexity of Lipschitz functions, we have the following theorem which is an adaptation of Theorem 7 for γ-margin loss.

Theorem 14 (Margin-based Rademacher Complexity Generalization Bound, [Mohri et al., 2018] Thm. 5.8)**.** *Let $\mathcal{H}$ be a set of real-valued functions. Fix $\gamma > 0$, then for any $\delta > 0$, with probability at least $1-\delta$, each of the following holds for all $h \in \mathcal{H}$:*

$$L(h) \leq \widehat{L}_S^\gamma(h) + \frac{2}{\gamma}\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\log 1/\delta}{2m}} \tag{27}$$

and

$$L(h) \leq \widehat{L}_S^\gamma(h) + \frac{2}{\gamma}\widehat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log 2/\delta}{2m}}. \tag{28}$$

Substituting Lemma 3 into Theorem 14, we obtain a margin-based generalization bound for linear hypotheses.

Theorem 15 (Margin-based Rademacher Complexity Generalization Bound for Linear Hypotheses with Bounded Weight Vector, [Mohri et al., 2018] Cor. 5.11)**.** *Let $\mathcal{H} = \{\boldsymbol{x} \mapsto \boldsymbol{w}^\top \boldsymbol{x} : \|\boldsymbol{w}\| \leq \Lambda\}$ and assume $\mathcal{X} \subseteq \{\boldsymbol{x} : \|\boldsymbol{x}\| \leq r\}$. Fix $\gamma > 0$, then for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, the following holds for any $h \in \mathcal{H}$:*

$$L(h) \leq \widehat{L}_S^\gamma(h) + \frac{2r\Lambda}{\gamma\sqrt{m}} + 3\sqrt{\frac{\log 2/\delta}{2m}}. \tag{29}$$

In comparison to the bound obtained from substituting Lemma 3 into Theorem 7, Theorem 15 uses margin to distinguish which hypotheses are more generalizable even when they have similar empirical error. The bound provides justification for margin-maximization algorithms such as SVM – in a sense, SVM is explicitly seeking a highly generalizable solution by including the complexity measure in the loss function. To see this, choose $\Lambda = 1$ and note that for all $z \in \mathbb{R}$ the γ-margin loss is upper bounded by the γ-hinge loss $\max(0, 1 - z/\gamma)$. Thus, for any $\delta > 0$, the following holds with probability at least $1-\delta$ for all $\gamma > 0$[3]:

$$L(h) \leq \frac{1}{m}\sum_{i=1}^m \max\left(0, 1 - \frac{y_i(\boldsymbol{w}^\top \boldsymbol{x}_i)}{\gamma}\right) + \frac{4r}{\gamma\sqrt{m}} + \sqrt{\frac{\log\log_2 \frac{2r}{\gamma}}{m}} + 3\sqrt{\frac{\log 2/\delta}{m}}. \tag{30}$$

By restricting $\|\boldsymbol{w}\| \leq 1/\gamma$, this is equivalent to

$$L(h) \leq \frac{1}{m}\sum_{i=1}^m \max(0, 1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i)) + \frac{4r}{\gamma\sqrt{m}} + \sqrt{\frac{\log\log_2 \frac{2r}{\gamma}}{m}} + 3\sqrt{\frac{\log 2/\delta}{m}}. \tag{31}$$

Since only the first term depends on $\boldsymbol{w}$, this bound suggests selecting $\boldsymbol{w}$ as the solution of

$$\min_{\|\boldsymbol{w}\|^2 \leq 1/\gamma^2} \frac{1}{m}\sum_{i=1}^m \max(0, 1 - y_i(\boldsymbol{w}_i^\top \boldsymbol{x}_i)). \tag{32}$$

With Lagrange variable $\lambda \geq 0$, this can be rewritten

$$\min_{\boldsymbol{w}} \lambda \|\boldsymbol{w}\|^2 + \frac{1}{m}\sum_{i=1}^m \max(0, 1 - y_i(\boldsymbol{w}_i^\top \boldsymbol{x}_i)). \tag{33}$$

This equation exactly coincides with the SVM.

[3] The extra $\log\log$ term come from allowing any $\gamma > 0$ instead of fixing γ.

4.2 Deep Learning

Recall that VC-dimension is defined for models with binary outputs. However, neural networks and associated models often have real-valued outputs which are thresholded to obtain the prediction – for instance, taking the argmax over a vector of sigmoid outputs to obtain a classification result. Such models do not directly minimize the empirical error, but instead a loss function over the real-valued outputs. In this case, we can describe generalization in terms of how close the output is to our desired threshold, a version of the confidence margin previously described. It turns out there is an analogue of VC-dimension for this problem called fat-shattering dimension, which we briefly describe here (for a full treatment, see [Anthony and Bartlett, 1999]).

We re-define the notation of the previous section to be slightly stricter. In particular, let $\widehat{L}_S^\gamma(h) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}_{y_i h(x_i)\leq\gamma}$. This measures the fraction of points classified incorrectly or correctly but with margin less than γ. **Check defn with book.**

Note that the growth function is insufficient for real-valued functions since $|\mathcal{H}_{|X}|$ is infinite. We introduce a generalization of the growth function, called the covering number, which measures the "extent" of $\mathcal{H}_{|X}$. Given $X \subseteq \mathbb{R}^n$ and $C \subseteq \mathbb{R}^n$, we say C is a d_∞ ε-cover for X if $C \subseteq X$ and for every $x \in X$ there is $c \in C$ with $\max\{|x_i - c_i| : i = 1, 2, \ldots, n\} < \varepsilon$. The d_∞ ε-covering number of X, denoted $\mathcal{N}(\varepsilon, X, d_\infty)$, is the minimum cardinality of a d_∞ ε-cover for X. For a real-valued hypothesis class $\mathcal{H}$ and domain $\mathcal{X} \subseteq \mathbb{R}^n$, we define the uniform covering number of $\mathcal{H}$ to be

$$\mathcal{N}_\infty(\varepsilon, \mathcal{H}, n) = \max\{\mathcal{N}(\varepsilon, \mathcal{H}_{|X}, d_\infty) : X \subseteq \mathcal{X}\}. \tag{34}$$

This is exactly a generalization of the growth function and can be interpreted as a measure of the richness of $\mathcal{H}$ at the scale ε. As with the growth function, we can obtain a uniform convergence result for real-valued functions in terms of the covering number.

Theorem 16 (Covering Number Bound for Real-Valued Functions, [Anthony and Bartlett, 1999] Thm. 10.1)**.** *Suppose $\mathcal{H}$ is a real-valued function class on the domain $\mathcal{X} \subseteq \mathbb{R}^n$ and let $\mathcal{D}$ be the data distribution. Let $\varepsilon \in [0, 1]$ and $\gamma > 0$. Then,*

$$\Pr_{S\sim\mathcal{D}^m}[L(h) \geq \widehat{L}_S^\gamma(h) + \varepsilon \text{ for some } h \in \mathcal{H}] \leq 2\mathcal{N}_\infty(\gamma/2, \mathcal{H}, 2m)\exp\Big(-\frac{\varepsilon^2 m}{8}\Big). \tag{35}$$

Covering numbers are in fact intimately related to Rademacher complexity via the following theorem from probability theory.

Theorem 17 (Dudley's Entropy Integral for Rademacher Complexity, [Tewari and Bartlett, 2014])**.** *For any $\mathcal{H}$ consisting of real-valued functions bounded by 1, we have for any sample S of size m,*

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) \leq \alpha + 12\int_\alpha^1 \sqrt{\frac{\log \mathcal{N}(\beta, \mathcal{H}_{|S}, d_2)}{m}}\, d\beta. \tag{36}$$

Recall that a set S is shattered by $\mathcal{H}$ if all 2^m classifications of S are achievable by $\mathcal{H}$. We can define something similar for real-valued functions using margin. Suppose $\mathcal{H}$ is a real-valued function class and $\gamma > 0$. Then, S is γ-shattered by $\mathcal{H}$ if there exists $r_1, r_2, \ldots, r_m \in \mathbb{R}$ such that for each $b \in \{0, 1\}^m$ there is a function $h_b \in \mathcal{H}$ such that

$$\begin{cases} h_b(x_i) \geq r_i + \gamma & b_i = 1, \\ h_b(x_i) \leq r_i - \gamma & b_i = 0. \end{cases} \tag{37}$$

That is, S is γ-shattered by $\mathcal{H}$ if all "positive" and "negative" labelings of S are achievable with "width" at least γ. The fat-shattering dimension $\mathrm{fat}_{\mathcal{H}}(\gamma)$ is the maximum cardinality of a subset S of $\mathcal{X}$ which is γ-shattered by $\mathcal{H}$.

Now, we can bound covering numbers in terms of fat-shattering dimension just as the Sauer-Shelah Lemma (Theorem 1) bounds the growth function in terms of VC-dimension.

Theorem 18 (Covering Number Upper Bound by Fat-Shattering Dimension, [Anthony and Bartlett, 1999] Thm. 12.8)**.** *Let $\mathcal{H}$ be a set of real functions from $\mathcal{X}$ to $[0,1]$. Let $\varepsilon > 0$ and $d = \text{fat}_{\mathcal{H}}(\varepsilon/4)$. Then for all $m \geq d$,*

$$\mathcal{N}_\infty(\varepsilon, \mathcal{H}, m) < 2\Big(\frac{4m}{\varepsilon^2}\Big)^{d \log_2(4em/(d\varepsilon))}. \tag{38}$$

Thus, computing the fat-shattering dimension of the function class allows us to bound the generalization error in terms of the empirical margin error. For neural networks, this typically involves a covering numbers analysis on the class of functions computed by the neural network – many modern results in the next section also utilize this idea. We give two bounds on the fat-shattering dimension of neural networks: one in terms of the number of parameters, and another in terms of the size of parameters. Suppose the neural network has depth D and W parameters, each computation unit maps onto the interval $[-b, b]$, and there are constants $J > 0$ and $K > 1/J$ so that all units have $\|\boldsymbol{w}\|_1 \leq J$ and the activation function is K-Lipschitz.

Theorem 19 (Fat-Shattering Dimension by Number of Parameters, [Anthony and Bartlett, 1999] Thm. 14.9)**.** *For the class $\mathcal{F}$ of functions computed by the neural network described above,*

$$\text{fat}_{\mathcal{F}}(\varepsilon) \leq 16W\Big(D\log(JK) + 2\log(32W) + \log\Big(\frac{b}{\varepsilon(JK-1)}\Big)\Big). \tag{39}$$

Theorem 20 (Fat-Shattering Dimension by Size of Parameters, [Anthony and Bartlett, 1999] Thm. 14.19)**.** *For the class $\mathcal{F}$ of functions computed by the neural network described above,*

$$\text{fat}_{\mathcal{F}}(\varepsilon) \leq 4\Big(\frac{32b}{\varepsilon}\Big)^{2D}(2JK)^{D(D+1)}\log(2n+2). \tag{40}$$

4.3 Modern Deep Learning

As in the SVM, margins in deep learning are intricately related to weight norms. In particular, when training a neural network, the weight norms tend towards infinity, so the results of Section 3.2 will also increase as the model is trained longer. We can avoid this by properly normalizing the weight norm by the margin as in Theorem 15. With a large margin, these generalization bounds become much more reasonable, so the main difficulty lies in finding the tightest definitions of margin and norm-based complexity. A related question is whether deep learning actually achieves large-margin solutions; this topic is a highly active area of research, and recent work has shown that the implicit bias of gradient descent finds the maximum-margin linear classifier (*i.e.,* SVM solution) in simple cases [Soudry et al., 2018, Lyu et al., 2021].

[Bartlett et al., 2017] empirically find that the Lipschitz constant of a neural network (its product of spectral norms) is tightly correlated with excess risk, and normalizing by the margin mitigates its unbounded growth as the weight norms increase. They provide a generalization bound which has no superlogarithmic dependence on number of parameters or depth, instead depending on the Lipschitz constant of the network times a "correcting factor" and normalized by the margin. Suppose a neural network $f_{\boldsymbol{w}}$ has K_i-Lipschitz activations at the i^{th} layer and $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_D$ are "reference matrices" of the same dimension as the weight matrices. The choice of reference matrix depends on the architecture, but common choices include the identity or the zero matrix. Then, the spectral complexity of $f_{\boldsymbol{w}}$ is defined as

$$R := \Big(\prod_{i=1}^{D} K_i \|\mathbf{W}_i\|_2\Big)\Big(\sum_{i=1}^{D} \frac{\|\mathbf{W}_i^\top - \mathbf{M}_i^\top\|_{2,1}^{2/3}}{\|\mathbf{W}_i\|_2^{2/3}}\Big)^{3/2}. \tag{41}$$

They define the margin in the multiclass setting as the gap between the correct label and other labels: $\gamma = f_{\boldsymbol{w}}(x)_y - \max_{j \neq y} f_{\boldsymbol{w}}(x)_j$. Thus $L(f_{\boldsymbol{w}}) = \Pr_{(\boldsymbol{x},y)\sim\mathcal{D}}[\operatorname{argmax}_j f_{\boldsymbol{w}}(\boldsymbol{x})_j \neq y]$ and $\widehat{L}_S^{\gamma}(f_{\boldsymbol{w}}) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}_{f_{\boldsymbol{w}}(x_i)_{y_i} \leq \gamma + \max_{j\neq y_i} f_{\boldsymbol{w}}(x_i)_j}$. They obtain the following generalization bound via a covering numbers argument.

Theorem 21 (Spectrally-Normalized Margin Generalization Bound, [Bartlett et al., 2017] Thm. 1.1)**.** *Let $f_{\boldsymbol{w}}$ and $\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_D$ be given as above. Then for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a (multiclass) sample S of size m, every margin $\gamma > 0$ satisfies*

$$L(f_{\boldsymbol{w}}) \leq \widehat{L}_S^{\gamma}(f_{\boldsymbol{w}}) + \widetilde{\mathcal{O}}\left(\frac{\|X\|\, R}{\gamma\sqrt{m}} \log W + \sqrt{\frac{\log 1/\delta}{m}}\right), \tag{42}$$

where $\|X\| = \sqrt{\frac{1}{m}\sum_{i=1}^m \|\boldsymbol{x}_i\|_2^2}$.

While this bound is conceptually appealing due to its characterization of generalization in terms of the margin and Lipschitz constant, it can be quite poor in practice [Neyshabur et al., 2017, Jiang et al., 2020]. A sharper bound for two-layer ReLU neural networks was proposed by [Neyshabur et al., 2019]. Suppose $\mathbf{U}$ is the weight matrix of the first layer and $\mathbf{V}$ is the weight matrix of the second layer, so $f_{\boldsymbol{w}} = \mathbf{V}[\mathbf{U}\boldsymbol{x}]_+$. Furthermore suppose $\|\boldsymbol{v}_i\|$ and $\left\|\boldsymbol{u}_i - \boldsymbol{u}_i^0\right\|$ are bounded (superscript 0 denotes initialization).

Theorem 22 (Two-Layer ReLU Neural Network Generalization Bound, [Neyshabur et al., 2019] Thm. 2)**.** *Suppose $f_{\boldsymbol{w}}$ is defined as above and has margin $\gamma > 0$ for a k-class classification task. Then for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m,*

$$L(f_{\boldsymbol{w}}) \leq \widehat{L}_S^{\gamma}(f_{\boldsymbol{w}}) + \widetilde{\mathcal{O}}\left(\frac{\sqrt{k}\, \|\mathbf{V}\|_F \left(\left\|\mathbf{U} - \mathbf{U}^0\right\|_F + \left\|\mathbf{U}^0\right\|_2\right) \|X\|}{\gamma\sqrt{m}} + \sqrt{\frac{W}{m}}\right), \tag{43}$$

where $\|X\| = \sqrt{\frac{1}{m}\sum_{i=1}^m \|\boldsymbol{x}_i\|_2^2}$.

In a follow-up work to [Bartlett et al., 2017], [Wei and Ma, 2019] extend their techniques to consider the hidden layer norms and interlayer Jacobian norms in addition to the training margin. This enables them to obtain a bound which does not rely on the implicitly-exponential product of weight Frobenius norms as in Section 3.2. The main technical contribution is augmenting the loss function with indicators for data-dependent quantities, enabling a tighter covering numbers analysis. The following is their simplified result for a model with zero training error; see [Wei and Ma, 2019] for details.

Theorem 23 (Margin Generalization Bound with Lipschitz Augmentation, [Wei and Ma, 2019] Thm. 1.1)**.** *Let t be the maximum ℓ_2 norm of any hidden layer or training datapoint and σ be the maximum operator norm of any interlayer Jacobian, evaluated on the training data. Then for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, every margin $\gamma > 0$ satisfies*

$$L(f_{\boldsymbol{w}}) \leq \widetilde{\mathcal{O}}\left(\frac{(\frac{\sigma}{\gamma}D^3\sigma^2)t\left(1 + \sum_{i=1}^D \|\mathbf{W}_i\|_{2,1}^{2/3}\right)^{3/2} + D^2\sigma\left(1 + \sum_{i=1}^D \|\mathbf{W}_i\|_{1,1}^{2/3}\right)^{3/2}}{\sqrt{m}} + D\sqrt{\frac{\log 1/\delta}{m}}\right). \tag{44}$$

Part of the difficulty of this proof lies in the complicated nature of the margin γ and its unclear relationship to weight norms. To remedy this, [Wei and Ma, 2020] propose an all-layer margin building on [Elsayed et al., 2018], which considers all layers of the network simultaneously and enables simpler layer-wise normalization. The definition is somewhat related to adversarial training [Madry et al., 2018] and indeed is empirically shown to improve robust classification. If $f_{\boldsymbol{w}}$ is a classifier formed by a composition of D functions $f_1, \ldots, f_D$, then the layer-wise output with perturbations $\delta_1, \ldots, \delta_D$ is defined as

$$
\begin{aligned}
h_1(\boldsymbol{x}, \boldsymbol{\delta}) &= f_1(\boldsymbol{x}) + \delta_1 \left\|\boldsymbol{x}\right\|_2 && (45)\\
h_i(\boldsymbol{x}, \boldsymbol{\delta}) &= f_i(h_{i-1}(\boldsymbol{x}, \boldsymbol{\delta}) + \delta_i \left\|h_{i-1}(\boldsymbol{x}, \boldsymbol{\delta})\right\|_2 && (46)\\
f_{\boldsymbol{w}}(\boldsymbol{x}, \boldsymbol{\delta}) &= h_D(\boldsymbol{x}, \boldsymbol{\delta}). && (47)
\end{aligned}
$$

Note that multiplying the perturbation by the previous layer norm "balances" the relative scale of the perturbations at each layer. Then, the all-layer margin is defined as the minimum perturbation needed to force misclassification:

$$
m_{f_{\boldsymbol{w}}}(\boldsymbol{x}, y) = \min_{\delta_1, \ldots, \delta_D} \sqrt{\sum_{i=1}^{D} \left\|\delta_i\right\|_2^2} \text{ s.t. } \operatorname*{argmax}_{y'} f_{\boldsymbol{w}}(\boldsymbol{x}, \delta_1, \ldots, \delta_D)_{y'} \neq y. \tag{48}
$$

With this definition in hand, a similar covering numbers analysis and a lower bound on $m_{f_{\boldsymbol{w}}}$ for smooth activations yields the following bound.

Theorem 24 (All-Layer Margin Generalization Bound, [Wei and Ma, 2020] Thm. 3.1, $q = 2$)**.** *Suppose the activation has a Lipschitz derivative. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size m,*

$$
L(f_{\boldsymbol{w}}) \leq \mathcal{O}\left(\frac{\left(\sum_{i=1}^{D} \left(\sum_{\boldsymbol{x}, y \in S} \kappa_i^2(\boldsymbol{x}, y) \right)^{1/3} a_i^{2/3} \right)^{3/2} \log^2 m}{\sqrt{m}} + \text{ low order terms} \right). \tag{49}
$$

Here, κ_i captures a local Lipschitz constant of perturbations at layer i and is related to the m_F lower bound, while a_i involves $\left\|\mathbf{W}_i\right\|_F$ and $\left\|\mathbf{W}_i\right\|_{1,1}$. See [Wei and Ma, 2020] for more details.

5 PAC-Bayes and Sharpness

5.1 Classical Learning

A longstanding hypothesis in deep learning is that "flat" minima – those solutions in some sense surrounded by solutions of similar training loss – generalize better than "sharp" minima [Hinton and van Camp, 1993, Hochreiter and Schmidhuber, 1997]. These early works utilized minimum description length (MDL) [Rissanen, 1983] arguments to justify why flat minima should generalize. The MDL principle is an information-theoretic complexity tradeoff which suggests that the best model is that which minimizes the bits needed to describe both the model and its misfit to the data. In particular, a sharp minimum corresponds to weights that must be specified with high precision, while a flat minimum requires fewer bits of information. A similar, Bayesian view is that flat minima correspond to maxima of the posterior weight distribution with greater probability mass [Hochreiter and Schmidhuber, 1997]. Accordingly, [Hinton and van Camp, 1993] minimize a conventional error term plus the KL divergence between the prior and posterior distributions on the weights; they do not derive numerical generalization bounds. In constrast, [Hochreiter and

Schmidhuber, 1997] avoid specifying a weight prior, instead searching over axis-aligned boxes within each minimum. Each box encompasses ε-optimal solutions within a minimum, and they search for boxes of maximum volume which (1) minimize the sensitivity of the loss function within the box, and (2) minimize the variance of the loss in any direction. Note that condition (1) is first-order while condition (2) is second-order, so this algorithm requires computation of the Hessian.

With empirical performance of flat minima established and some preliminary MDL/Bayesian justification, a remaining question is how to develop rigorous bounds on the generalization error of such models. In particular, the PAC or uniform convergence bounds previously mentioned are independent of any truth of the prior and cannot be directly applied. The PAC-Bayes framework [McAllester, 1999] combines the advantages of both types of reasoning by provides guarantees on the expected error of a randomized hypothesis ("posterior") drawn from a distribution $\mathcal{Q}$ over the class $\mathcal{H}$ with respect to a prior distribution $\mathcal{P}$. Suppose hypothesis $h_i \in \mathcal{H}$ has probability q_i and p_i in $\mathcal{Q}$ and $\mathcal{P}$ respectively. We redefine our definitions of error to account for these distributions as follows. Let $\ell : \mathcal{H} \times \mathcal{X} \to [0,1]$ be a loss function, and let $L(h) := \mathbb{E}_{x\sim\mathcal{D}}[\ell(h,x)]$, $\widehat{L}_S(h) := \frac{1}{m}\sum_{i=1}^m \ell(h,x)$, $L(\mathcal{Q}) = \mathbb{E}_{h\sim\mathcal{Q}}[L(h)]$, and $\widehat{L}_S(\mathcal{Q}) = \mathbb{E}_{h\sim\mathcal{Q}}[\widehat{L}_S(h)]$. A PAC-Bayes lemma in the form of Theorem 7 is:

Lemma 4 (PAC-Bayes Model Selection, [McAllester, 1998, 1999])**.** *For any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, the following holds for all $h_i \in \mathcal{H}$:*

$$L(h_i) \leq \widehat{L}_S(h_i) + \sqrt{\frac{\log\frac{1}{p_i} + \log\frac{1}{\delta}}{2m}}. \tag{50}$$

The proof is straightforward and utilizes a Chernoff bound. A major PAC-Bayes theorem generalizes this statement to distributions over hypotheses. Recall the definition of the KL divergence as $\mathrm{KL}[\mathcal{Q}\|\mathcal{P}] = \int_{h_i\in\mathcal{H}} q_i \log\frac{q_i}{p_i}$ when $\mathcal{Q}$ and $\mathcal{P}$ are absolutely continuous.

Theorem 25 (PAC-Bayes Generalization Bound, [McAllester, 1999] Thm. 1)**.** *For any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, the following holds for all distributions $\mathcal{P}, \mathcal{Q}$ over $\mathcal{H}$ with $q_i \geq p_i$ for all $h_i \in \mathcal{H}$:*

$$L(\mathcal{Q}) \leq \widehat{L}_S(\mathcal{Q}) + \sqrt{\frac{\mathrm{KL}[\mathcal{Q}\|\mathcal{P}] + \log\frac{1}{\delta} + \frac{5}{2}\log m + 8}{2m-1}}. \tag{51}$$

A useful variant of this theorem is as follows:

Theorem 26 (PAC-Bayes Relative Entropy Bound, [Langford and Seeger, 2001] Thm. 3)**.** *For any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, the following holds for all distributions $\mathcal{P}, \mathcal{Q}$ over $\mathcal{H}$ with $q_i \geq p_i$ for all $h_i \in \mathcal{H}$:*

$$\mathrm{KL}[\widehat{L}_S(\mathcal{Q})\|L(\mathcal{Q})] \leq \frac{\mathrm{KL}[\mathcal{Q}\|\mathcal{P}] + \log\frac{m}{\delta}}{m-1}. \tag{52}$$

The PAC-Bayes framework is able to re-derive results for margin-based linear classifiers similar to 15. An early example is the following.

Theorem 27 (PAC-Bayes Generalization Bound for Linear Hypotheses)**.** *Let $\mathcal{H} = \{\boldsymbol{x} \mapsto \boldsymbol{w}^\top\boldsymbol{x} : \|\boldsymbol{w}\| = 1$. Suppose $\widehat{L}_S^\gamma(h) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}_{y_i h(x_i)\leq\gamma}$ and $\log^+(z) = \max(0, \log z)$. Then for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, the following holds for all $h \in \mathcal{H}$:*

$$L(h) \leq \widehat{L}_S^\gamma(h) + 2\sqrt{\frac{2(\widehat{L}_S^\gamma(h) + \frac{4}{m\gamma^2})\log^+(m\gamma^2)}{m\gamma^2}} + \frac{8(1+\log^+(\frac{m\gamma^2}{4})}{m\gamma^2} + \mathcal{O}\left(\sqrt{\frac{\log m + \log 1/\delta}{m}}\right). \tag{53}$$

Note that this formulation of the PAC-Bayes framework necessitates a stochastic model, as the output is a distribution over hypotheses, and results such as the above must de-randomize the stochastic guarantee. The PAC-Bayes framework was first applied to "flat minima", in the sense of noise sensitivity, by [Langford and Caruana, 2001]. They consider a stochastic neural network with a multivariate isotropic Gaussian "prior" distribution $\mathcal{P}$ over the weights with zero mean and variance b^2, where for the j^{th} weight, $b_j = c\alpha^j$ for constants c, α. Then, the "posterior" $\mathcal{Q}$ is a Gaussian distribution over the weights, centered at the trained weight values of the network w_j and with variance s_j^2 corresponding to the sensitivity of that weight – in practice, they search for values s_j^2 which, when perturbing w_j and leaving other weights constant, the accuracy of the neural network is reduced by (say) 5%. Using Theorem 26 along with an analytical calculation of the KL divergence and a union bound over the weights, they obtain the following generalization bound.

Theorem 28 (PAC-Bayes Relative Entropy for Neural Networks, [Langford and Caruana, 2001] Cor. 2.4)**.** *Suppose $\mathcal{H}$ is the class of functions computed by a stochastic neural network with W parameters as described above. Then for any $\delta > 0$, with probability at most δ over the choice of a sample S, there exists a posterior $\mathcal{Q}$ with the following property.*

$$\mathrm{KL}[\widehat{L}_S(\mathcal{Q}) \| L(\mathcal{Q})] \geq \inf_j \frac{\sum_{i=1}^{P} \left(\log \frac{c\alpha^j}{s_i} + \frac{s_i^2 + w_i^2}{2c^2\alpha^{2j}} - \frac{1}{2} \right) + \log \frac{\pi^2 j^2 m}{3\delta}}{m-1}. \tag{54}$$

5.2 Modern Deep Learning

Sharpness was re-introduced to the modern deep learning community independently and simultaneously by [Keskar et al., 2017, Chaudhari et al., 2017], who showed empirically that SGD tends to converge to flat minima which improve generalization. While [Chaudhari et al., 2017] use the Bayesian language of prior work, minimizing the spectrum of the Hessian at the solution by maximizing local entropy, [Keskar et al., 2017] introduced a new definition measuring robustness to adversarial perturbations in the weight space.

Definition 1 (Adversarial Sharpness, [Keskar et al., 2017, Neyshabur et al., 2017])**.** *Let $f_{\boldsymbol{w}}$ be the function described by a given neural network at weight vector $\boldsymbol{w}$. Then, the adversarial sharpness of $f_{\boldsymbol{w}}$ with scale $\alpha > 0$ with respect to sample S is*

$$\zeta_\alpha(\boldsymbol{w}) = \frac{\max_{|\boldsymbol{\nu}_i| \leq \alpha(|\boldsymbol{w}_i| + \mathbf{1})} \widehat{L}_S(f_{\boldsymbol{w}+\boldsymbol{\nu}}) - \widehat{L}_S(f_{\boldsymbol{w}})}{1 + \widehat{L}_S(f_{\boldsymbol{w}})}. \tag{55}$$

Note that $\widehat{L}_S(f_{\boldsymbol{w}}) \approx 0$ in practical cases, so we typically drop the denominator.

Some shortcomings of the adversarial definition of sharpness was addressed in [Dinh et al., 2017], who showed that it is not invariant to reparameterization techniques such as batch normalization [Ioffe and Szegedy, 2015]. Nevertheless, these empirical observations ignited a new wave of theoretical advancements in using PAC-Bayes theory to explain generalization. Despite the inability of PAC-Bayes to characterize certain simple learning problems [Livni and Moran, 2020], sharpness bounds were surprisingly found to be nonvacuous in real-world overparameterized neural networks where VC-dimension and Rademacher complexity approaches fail [Dziugate and Roy, 2017].

The approach of [Dziugate and Roy, 2017] to derive nonvacuous PAC-Bayes generalization bounds is similar to that of [Langford and Caruana, 2001], with a crucial difference – in modern neural networks, the sensitivity of the loss with respect to a single parameter is negligible, so they instead use SGD to directly optimize the PAC-Bayes bound on the error of a stochastic neural network initialized. Similarly to [Langford and Caruana, 2001], they initialize $\mathcal{Q}$ as a Gaussian distribution centered at the trained weight values of the network w_j with covariance matrix $\mathrm{diag}(\boldsymbol{s})$, and they let $\mathcal{P}$ be a centered Gaussian distribution with covariance matrix $\lambda \mathbf{I}$ for a free variable $\lambda = c\exp(-j/b)$. Since $\mathcal{Q}$ is intractable to compute exactly, they show that a Monte Carlo approximation $\widehat{\mathcal{Q}}_k$ suffices. To utilize the relative entropy bound, they consider the inverse KL divergence $\mathrm{KL}^{-1}[q|c] = \sup\{p \in [0,1] : \mathrm{KL}[q\|c] \le p\}$, which has no closed form but can be approximated by Newton's method or the inequality $\mathrm{KL}^{-1}[q|c] \le q + \sqrt{c/2}$. Then, they obtain the following generalization bound. During experimentation, they numerically optimize over $\boldsymbol{w}, \boldsymbol{s}$, and λ to obtain nonvacuous generalization bounds on the MNIST dataset [LeCun et al., 2010].

Theorem 29 (Nonvacuous PAC-Bayes Generalization Bound, [Dziugate and Roy, 2017])**.** *For any $\delta, \delta' > 0$, with probability at most $1 - \delta - \delta'$ over the choice of a sample S of size m, the following holds for the aforementioned distributions $\mathcal{P}, \mathcal{Q}$ over $\mathcal{H}$:*

$$L(\mathcal{Q}) \le \mathrm{KL}^{-1}\Big[\mathrm{KL}^{-1}[\widehat{L}_S(\widehat{\mathcal{Q}}_k)|k^{-1}\log 2/\delta']\Big| B_{\mathrm{RE}}(\boldsymbol{w}, \boldsymbol{s}, \lambda; \delta)\Big], \tag{56}$$

where

$$B_{\mathrm{RE}}(\boldsymbol{w}, \boldsymbol{s}, \lambda; \delta) = \frac{\mathrm{KL}[\mathcal{Q}\|\mathcal{P}] + 2\log(b\log\frac{c}{\lambda}) + \log\frac{\pi^2 m}{6\delta}}{m-1} \tag{57}$$

with the KL divergence term analytically evaluating to

$$\frac{1}{2}\Big(\frac{1}{\lambda}\|\boldsymbol{s}\|_1 + \frac{1}{\lambda}\|\boldsymbol{w} - \boldsymbol{w}_0\|_2^2 + n\log\lambda - \mathbf{1}_n \log\boldsymbol{s}\Big). \tag{58}$$

In a closely related work, [Neyshabur et al., 2017] argue that the adversarial sharpness of [Keskar et al., 2017] does not capture generalization by itself, as it does not explain the generalization behavior of models trained on true vs. random labels, and it is not scale-invariant. Instead, they advocate expected sharpness in the context of the PAC-Bayes framework. They show that sharpness, like margin, must be explicitly normalized by the weight norm, and this balance determines the variance which should be selected for the prior distribution.

Theorem 30 (Expected Sharpness PAC-Bayes Generalization Bound, [Neyshabur et al., 2017])**.** *Suppose the prior distribution over hypotheses $\mathcal{P}$ and perturbation $\boldsymbol{\nu}$ are centered, σ^2 variance Gaussian distributions. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of a sample S of size m,*

$$\mathbb{E}_{\boldsymbol{\nu}\sim\mathcal{N}(0,\sigma)^n}[L_S(f_{\boldsymbol{w}+\boldsymbol{\nu}})] \le \widehat{L}_S(f_{\boldsymbol{w}}) + \underbrace{\mathbb{E}_{\boldsymbol{\nu}\sim\mathcal{N}(0,\sigma)^n}[\widehat{L}_S(f_{\boldsymbol{w}+\boldsymbol{v}})] - \widehat{L}_S(f_{\boldsymbol{w}})}_{\text{Expected sharpness}} + 4\sqrt{\frac{1}{m}\Big(\underbrace{\frac{\|\boldsymbol{w}\|_2^2}{2\sigma^2}}_{\text{KL}} + \log\frac{2m}{\delta}\Big)}. \tag{59}$$

Then, [Neyshabur et al., 2017] bound sharpness in terms of the weight norms and σ and note that this formulation avoids the explicit exponential dependence on depth as typical in bounds from Section 3.2. This notion of expected sharpness, as well as the adversarial definition of [Keskar et al., 2017], was empirically found by [Jiang et al., 2020] to correlate better with generalization than any norm- or margin-based complexity measure. Notably, some recent theoretical works have shown that the implicit regularization of SGD provably prefers flat minima [He et al., 2019, Mulayoff and Michaeli, 2020, Damian et al., 2021, Xie et al., 2021]. However, explicitly seeking flat minima via stochastic weight averaging (SWA) or sharpness-aware minimization (SAM) empirically performs better than SGD alone [Izmailov et al., 2018, Foret et al., 2021].

The PAC-Bayes framework is quite general and has been used to prove generalization bounds with different complexity measures than sharpness. In particular, [Neyshabur et al., 2018] derive margin-based spectral complexity bounds similar to Theorem 21 using a PAC-Bayes approach. Compared to the covering numbers argument of [Bartlett et al., 2017], the PAC-Bayes framework yields an only slightly weaker bound with a much simpler proof. The key ingredient is a lemma which bounds the change in the output of a neural network with respect to a perturbation of its weights – that is, the model sharpness – in terms of the spectral norm. This lemma gives the following generalization bound.

Theorem 31 (PAC-Bayes Spectrally-Normalized Margin Generalization Bound, [Neyshabur et al., 2018])**.** *Suppose $\|\boldsymbol{x}\|_2 \leq B$ for $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$. Then, for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, every margin $\gamma > 0$ satisfies*

$$L(f_{\boldsymbol{w}}) \leq \widehat{L}_S^{\gamma}(f_{\boldsymbol{w}}) + \mathcal{O}\left(\sqrt{\frac{B^2 D^2 W \log(DW) \prod_{i=1}^{D} \|\mathbf{W}_i\|_2^2 \sum_{i=1}^{D} \frac{\|\mathbf{W}_i\|_F^2}{\|\mathbf{W}_i\|_2^2} + \log \frac{Dm}{\delta}}{\gamma^2 m}}\right). \tag{60}$$

Another PAC-Bayes bound characterizes generalization based on module criticality, an alternative measure of parameter robustness [Chatterji et al., 2020, Zhang et al., 2022]. A module, or layer, is called critical if rewinding its parameters back to initialization while keeping other network parameters at their trained values results in a large drop in network performance. The module's robustness to this rewinding process – that is, the width and flatness of the valley on the loss surface connecting the initial and final weights – is associated with better generalization. Suppose module i has k_i parameters, initial weights $\boldsymbol{\theta}_i^0$, and final weights $\boldsymbol{\theta}_i^F$, then let $\boldsymbol{\theta}_i^{\alpha} = (1-\alpha)\boldsymbol{\theta}_i^0 + \alpha\boldsymbol{\theta}_i^F$ for $\alpha \in [0,1]$. Let $\Theta^{\boldsymbol{\alpha}}$ refer to the model where all parameters are $\boldsymbol{\theta}_i^{\alpha_i}$.

Theorem 32 (PAC-Bayes Module Criticality Generalization Bound, [Chatterji et al., 2020])**.** *Suppose the posterior $\mathcal{Q}_i$ for module i is a Gaussian centered at $\boldsymbol{\theta}_i^{\alpha_i}$ with covariance matrix $\sigma^2\mathbf{I}$. Let $\mathbf{u}$ be the vector collating the noise from all the modules. Then for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m,*

$$\mathbb{E}_{\mathbf{u}}[L(f_{\Theta^{\boldsymbol{\alpha}}+\mathbf{u}})] \leq \mathbb{E}_{\mathbf{U}}[\widehat{L}_S(f_{\Theta^{\boldsymbol{\alpha}}+\mathbf{u}})] + \sqrt{\frac{\frac{1}{4}\sum_{i=1}^{D} k_i \log\left(1 + \frac{\alpha_i^2 \|\boldsymbol{\theta}_i^F - \boldsymbol{\theta}_i^0\|_F^2}{k_i \sigma_i^2}\right) + \log\frac{m}{\delta} + \widetilde{\mathcal{O}}(1)}{m-1}}. \tag{61}$$

The proof technique of [Chatterji et al., 2020] turns out to be applicable to sharpness as well, enabling a bound in terms of the loss under adversarial perturbation of the weights in a certain neighborhood – harkening back to the adversarial sharpness of [Keskar et al., 2017]. Note that the total number of parameters W of the model appears in this bound.

Theorem 33 (Sharpness-Aware Minimization Generalization Bound, [Foret et al., 2021])**.** *Assume that $L(f_{\boldsymbol{w}}) \leq \mathbb{E}_{\nu_i \sim \mathcal{N}(0,\sigma)} L(f_{\boldsymbol{w}+\boldsymbol{\nu}})$. Then for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m,*

$$L(f_{\boldsymbol{w}}) \leq \max_{\|\boldsymbol{\nu}\|_2 \leq \sigma} \widehat{L}_S(f_{\boldsymbol{w}+\boldsymbol{\nu}}) + \sqrt{\frac{W \log\left(1 + \frac{\|\boldsymbol{w}\|_2^2}{\sigma^2}\left(1 + \sqrt{\frac{\log m}{W}}\right)^2\right) + 4\log\frac{m}{\delta} + \widetilde{\mathcal{O}}(1)}{m-1}}. \tag{62}$$

While some bounds (*e.g.,* Theorem 31 and 33) obtain deterministic PAC-Bayes bounds in their specific setup, a remaining question is how to de-randomize PAC-Bayes bounds in general, and whether we can avoid the implicit exponential dependence on depth as in Theorem 31. Towards this goal, [Nagarajan and Kolter, 2019a] identify two properties which enable a tighter deterministic PAC-Bayes bound. In an abstract sense, [Nagarajan and Kolter, 2019a] call a classifier noise-resilient if it satisfied R different conditions – typically one per layer – on input-dependent properties of the weights. Suppose the r^{th} condition involves properties $\rho_{r,1}(\boldsymbol{w}, \boldsymbol{x}, y), \rho_{r,2}(\boldsymbol{w}, \boldsymbol{x}, y), \ldots$ with corresponding set of positive constants $\Delta^\star_{r,1}, \Delta^\star_{r,2}, \ldots$ Then, the weights $\boldsymbol{w}$ satisfy the r^{th} condition on the input $(\boldsymbol{x}, y)$ if

$$\forall l \quad \rho_{r,l}(\boldsymbol{w}, \boldsymbol{x}, y) > \Delta^\star_{r,l}. \tag{63}$$

Crucially, we must also have the property that if the first $r-1$ conditions are satisfied, then the r^{th} condition is noise resilient – this is natural for deep networks, as if (say) the weights in the first $r-1$ layers are small, then the weights in the r^{th} layer should also be small. Formally, define $\Delta_{r,l}(\sigma)$ which bound the perturbation in the property $\rho_{r,l}$ in terms of the variance σ^2 of the parameter perturbations. For all r, if for all $q < r$ and for all l we have $\rho_{q,l}(\boldsymbol{w}, \boldsymbol{x}, y) > 0$, then

$$\Pr_{u_i \sim \mathcal{N}(0,\sigma^2)} \Bigg[\forall l |\rho_{r,l}(\boldsymbol{w}+\boldsymbol{u}, \boldsymbol{x}, y) - \rho_{r,l}(\boldsymbol{w}, \boldsymbol{x}, y)| > \frac{\Delta_{r,l}(\sigma)}{2} \text{ and } \\ \forall q < r, \forall l |\rho_{q,l}(\boldsymbol{w}+\boldsymbol{u}, \boldsymbol{x}, y) - \rho_{q,l}(\boldsymbol{w}, \boldsymbol{x}, y)| < \frac{\Delta_{q,l}(\sigma)}{2} \Bigg] \leq \frac{1}{R\sqrt{m}}. \tag{64}$$

Now, we can state the general theorem. Note that the bound scales linearly with R.

Theorem 34 (Deterministic Noise-Resilient PAC-Bayes Generalization Bound [Nagarajan and Kolter, 2019a])**.** *Let $\sigma^\star$ be the maximum value of the Gaussian parameter perturbation such that Equation 64 holds with $\Delta_{r,l}(\sigma^\star) \leq \Delta^\star_{r,l} \quad \forall r, l$. Then for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, for any $\boldsymbol{w}$ we have that, if $\boldsymbol{w}$ satisfies the conditions in Equation 63 for all r and training samples $(\boldsymbol{x}, y) \in S$, then for any prior $\mathcal{P}$ and margin γ,*

$$L(f_{\boldsymbol{w}}) \leq \widehat{L}^\gamma_S(f_{\boldsymbol{w}}) + \widetilde{\mathcal{O}}\left(R\sqrt{\frac{2\mathrm{KL}(\mathcal{N}(\boldsymbol{w}, (\sigma^\star)^2\mathbf{I}) \| \mathcal{P}) + \ln\frac{2mR}{\delta}}{m-1}} \right). \tag{65}$$

To apply this bound to ReLU networks, they instantiate the framework with a particular set of $\mathcal{O}(D)$ properties including bounds on the (non-spectral) norms of the weight matrices, Jacobians, and pre-activations at each layer. Here, $1/\sigma^\star = \widetilde{\mathcal{O}}(\sqrt{W})$ times the maximum of these properties (see [Nagarajan and Kolter, 2019a] for the exact characterization).

Theorem 35 (Deterministic Noise-Resilient PAC-Bayes Generalization Bound for ReLU Neural Networks [Nagarajan and Kolter, 2019a])**.** *For any margin $\gamma > 0$ and $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample S of size m, for any $\boldsymbol{w}$ and initialization $\boldsymbol{z}$, we have that*

$$L(f_{\boldsymbol{w}}) \leq \widehat{L}^\gamma_S(f_{\boldsymbol{w}}) + \widetilde{\mathcal{O}}\left(D\sqrt{\|\boldsymbol{w}-\boldsymbol{z}\|_2^2 / ((\sigma^\star)^2 m)} \right). \tag{66}$$

6 Other Approaches

6.1 Algorithmic Stability

All generalization bounds previously discussed (including deterministic PAC-Bayes) rely on uniform convergence – the idea that all hypotheses in the class obey a certain generalization bound, so no matter which hypothesis the algorithm actually selects, the bound will hold. However, uniform convergence has been called into question by [Nagarajan and Kolter, 2019b], who show that uniform convergence can provably fail to explain generalization in common cases such as high-dimensional linear classifier and ReLU neural networks. In essence, their proofs show that gradient descent learns "simple" decision boundaries which generalize well, but contain "microscopic complexities" which are problematic for uniform convergence. For each hypothesis h, one can take advantage of these complexities to construct a dataset where h incurs large empirical error, rendering uniform convergence bounds vacuous even when taking the implicit bias of SGD in account to the fullest extent possible. Several recent works challenge the impact and universality of this phenomenon, from using an NTK-based analysis to show that the failure of uniform convergence is just a result of the specific dataset and architecture bias [Bachmann et al., 2021] to showing that restricting uniform convergence to zero-error predictors can help explain the generalization of low-norm interpolating learners [Zhou et al., 2020] or extending the definition of uniform convergence to handle learning problems of increasing complexity [Negrea et al., 2020]. Nonetheless, the results of [Nagarajan and Kolter, 2019b] have spurred new interest in alternative methods to explain generalization.

A particularly rich field of non-uniform convergence generalization bounds is that of algorithmic stability, which characterizes generalization in terms of the sensitivity of the training error to the removal of any one datapoint. The classical reference for this topic is [Bousquet and Elisseeff, 2002], who define and prove generalization bounds for several notions of stability. Different from uniform convergence, which essentially bounds

$$\Pr_{S\sim\mathcal{D}}\Big[\sup_{h\in\mathcal{H}}\big|L(h)-\widehat{L}_S(h)\big|\Big], \tag{67}$$

the stability approach fixes a (real-valued) algorithm A and bounds

$$\Pr_{S\sim\mathcal{D}}\Big[\big|L(A(S))-\widehat{L}_S(A(S))\big|\Big]. \tag{68}$$

In the following section, we denote $h \coloneqq A(S)$ and $h' \coloneqq A(S')$, and let $\ell(h,x)=\ell(h(x)x)$ be a loss function. The most important notion of stability is uniform stability.

Definition 2 (Uniform Stability, [Bousquet and Elisseeff, 2002] Def. 6)**.** *An algorithm A has uniform stability β with respect to the loss function ℓ if for all $S, S' \in \mathcal{X}^m$ differing in at most one element, we have*

$$\sup_{x\in\mathcal{X}}\big\|\ell(h(x),x)-\ell(h'(x),x)\big\|_\infty \leq \beta. \tag{69}$$

A is called stable if β decreases with $1/m$.

A related definition is classification stability – necessary because, under the zero-one loss, the only possible values of β in the original definition are 0 or 1.

Definition 3 (Classification Stability, [Bousquet and Elisseeff, 2002] Def. 15)**.** *A real-valued classification algorithm A has classification stability β if for all $S, S' \in \mathcal{X}^m$ differing in at most one element, we have*

$$\sup_{x\in\mathcal{X}}\big\|h(x)-h'(x)\big\|_\infty \leq \beta. \tag{70}$$

The following lemma relates uniform stability, classification stability, and margin.

Lemma 5 (Uniform and Classification Stability, [Bousquet and Elisseeff, 2002] Lem. 16)**.** *A real-valued classification algorithm A with classification stability β has uniform stability β/γ with respect to the γ-margin loss Φ_γ (see Equation 26).*

An application of McDiarmid's Inequality yields the following generalization bound.

Theorem 36 (Classification Stability Generalization Bound, [Bousquet and Elisseeff, 2002] Thm. 17)**.** *Let A be a real-valued classification algorithm with classification stability β. Then for all $\delta > 0$ with probability at least $1 - \delta$ over the choice of a sample S of size m,*

$$L(h) \leq \widehat{L}_S^{\gamma}(h) + 2\frac{\beta}{\gamma} + \left(4m\frac{\beta}{\gamma} + 1\right)\sqrt{\frac{\log 1/\delta}{2m}}. \tag{71}$$

A useful application of this theorem is to minimizing convex functions regularized by norms in a reproducing kernel Hilbert space (RKHS).

Theorem 37 (Uniform Stability of RKHS Learning, [Bousquet and Elisseeff, 2002] Thm. 22)**.** *Let $\mathcal{F}$ be a reproducing kernel Hilbert space with kernel k such that for all $x \in \mathcal{X}$, $k(x, x) \leq \kappa^2 < \infty$. Suppose ℓ is convex and K-Lipschitz with respect to its first argument. The learning algorithm A defined by*

$$A(S) = \operatorname*{argmin}_{g \in \mathcal{F}} \frac{1}{m}\sum_{i=1}^{m} \ell(g, z_i) + \lambda \|g\|_k^2 \tag{72}$$

has uniform stability β with respect to ℓ with

$$\beta \leq \frac{K^2\kappa^2}{2\lambda m}. \tag{73}$$

Theorem 36 and Theorem 37 imply a generalization bound for the soft-margin kernel SVM.

Theorem 38 (Uniform Stability Kernel SVM Generalization Bound, [Bousquet and Elisseeff, 2002])**.** *Consider the soft margin SVM with loss function $\ell(g, x) = \Phi_1(g(x)x)$. Clearly we have $K = 1, \gamma = 1$. The kernel SVM has classification stability β with*

$$\beta \leq \frac{\kappa^2}{2\lambda m}. \tag{74}$$

Thus,

$$L(h) \leq \frac{1}{m}\sum_{i=1}^{m} \ell(h, x_i) + \frac{\kappa^2}{\lambda m} + \left(\frac{2\kappa^2}{\lambda} + 1\right)\sqrt{\frac{\log 1/\delta}{2m}}. \tag{75}$$

A rich exploration of the relationship between stability and learning followed. In particular, [Shalev-Shwartz et al., 2010] showed that there are nontrivial learning problems which are learnable (in the sense of [Vapnik, 1995]), yet ERM fails and uniform convergence does not hold. Instead, [Shalev-Shwartz et al., 2010] establish stability as the necessary and sufficient condition for learning, and show that for strongly convex losses the ERM solution is stable.

In the deep learning era, a major contribution is that of [Hardt et al., 2016], who extended uniform stability to iterative algorithms and showed that stochastic gradient methods (SGM) are stable in the convex and non-convex cases, thus deriving generalization bounds for neural networks trained via (for example) SGD. We first state the stochastic definition of uniform stability and its relationship with generalization.

Definition 4 (Stochastic Uniform Stability, [Bousquet and Elisseeff, 2002, Shalev-Shwartz et al., 2010, Hardt et al., 2016])**.** *A randomized algorithm A is ε-uniformly stable with respect to a loss function ℓ if for all datasets $S, S' \in \mathcal{X}^m$ which differ in at most one element,*

$$\sup_{x \in \mathcal{X}} \mathbb{E}_A[\ell(h, x) - \ell(h', x)] \leq \varepsilon. \tag{76}$$

We denote by ε_{stab} the infimum over all ε for which this holds.

Theorem 39 (Stochastic Uniform Stability Generalization Bound, [Bousquet and Elisseeff, 2002, Shalev-Shwartz et al., 2010, Hardt et al., 2016])**.** *Suppose A is an ε-uniformly stable randomized algorithm. Then,*

$$\left|\mathbb{E}_{S,A}\left[\widehat{L}_S(h) - L(h)\right]\right| \leq \varepsilon. \tag{77}$$

In the convex case, the stability of SGM depends on the step sizes.

Theorem 40 (Uniform Stability of Convex SGM [Hardt et al., 2016])**.** *Assume that the loss function ℓ is convex, β-smooth, and K-Lipschitz for all $x \in \mathcal{X}$. Suppose we run SGM with step sizes $\alpha_t \leq 2/\beta$ for T steps. Then, SGM satisfies uniform stability with*

$$\varepsilon_{stab} \leq \frac{2K^2}{m} \sum_{t=1}^{T} \alpha_t. \tag{78}$$

In the strongly convex case for the projected SGM, this dependence is erased.

Theorem 41 (Uniform Stability of Strongly Convex SGM [Hardt et al., 2016])**.** *Assume that the loss function ℓ is γ-strongly convex and β-smooth for all $x \in \mathcal{X}$. Suppose we run projected SGM with constant step size $\alpha \leq 1/\beta$ for T steps. Then, SGM satisfies uniform stability with*

$$\varepsilon_{stab} \leq \frac{2K^2}{\gamma m}. \tag{79}$$

The non-convex case has additional dependence on T; in particular, models which train faster enjoy better uniform stability.

Theorem 42 (Uniform Stability of Non-convex SGM [Hardt et al., 2016])**.** *Assume that the loss function ℓ is bounded in $[0, 1]$, β-smooth, and K-Lipschitz for all $x \in \mathcal{X}$. Suppose we run SGM for T steps with monotonically non-increasing step sizes $\alpha_t \leq c/t$. Then, SGM satisfies uniform stability with*

$$\varepsilon_{stab} \leq \frac{1 + 1/\beta c}{m - 1} (2cK^2)^{\frac{1}{\beta c+1}} T^{\frac{\beta c}{\beta c+1}}. \tag{80}$$

Interestingly, this analysis gives some theoretical justification for techniques used to train neural networks in practice. For example, weight decay counts towards the smoothness parameter, while dropout and gradient clipping improve the effective Lipschitz constant. Additional recent contributions include [Feldman and Vondrak, 2018] who substantially sharpen the bounds of [Bousquet and Elisseeff, 2002], and [Kuzborskij and Lampert, 2018] who obtain data-dependent generalization bounds for SGD using a notion of on-average stability. The observation of [Hardt et al., 2016] that fast training time is sufficient for generalization was further explored in [Nakkiran et al., 2021].

6.2 Compression and Distillation

An intriguing phenomenon in deep learning is that extremely large neural networks can, in a variety of ways, be compressed or distilled into a much smaller model which achieves roughly the same generalization performance [Hinton et al., 2014, Han et al., 2015, Frankle and Carbin, 2019]. Since the original problem with parameter-counting based generalization bounds (see Section 2.2) is that they are vacuous for overparameterized models, studying the compressed network may enable simple, nonvacuous parameter-based bounds which avoid the problems of uniform convergence [Nagarajan and Kolter, 2019b].

The framework for compression bounds introduced by [Arora et al., 2018] enables this type of analysis. Suppose $\mathcal{W}$ is the space of parameterizations of a neural networks and $f_{\boldsymbol{w}}$ is the function computed by the model with parameterization $\boldsymbol{w} \in \mathcal{W}$.

Definition 5 ((γ, S)-compressible with Helper String, [Arora et al., 2018] Def. 2)**.** *Suppose $G_{\mathcal{W},s} = \{g_{\boldsymbol{w},s} : \boldsymbol{w} \in \mathcal{W}\}$ is a class of classifiers indexed by trainable parameters $\boldsymbol{w}$ and fixed strings s. A classifier f is (γ, S)-compressible with respect to $G_{\mathcal{W},s}$ using helper string s if there exists $\boldsymbol{w} \in \mathcal{W}$ such that, for any $x \in S$, we have that for all y,*

$$|f(x)_y - g_{\boldsymbol{w},s}(x)_y| \leq \gamma. \tag{81}$$

The helper string is often the random initialization in the sense of [Dziugate and Roy, 2017] or a collection of random matrices as in the Johnson-Lindenstrauss Lemma.

A straightforward Chernoff bound analysis yields the following generalization bound.

Theorem 43 ((γ, S)-compression Generalization Bound, [Arora et al., 2018] Thm. 2.1)**.** *Suppose $G_{\mathcal{W},s} = \{g_{\boldsymbol{w},s} : \boldsymbol{w} \in \mathcal{W}\}$ where $\boldsymbol{w}$ is a set of q parameters each of which can have at most r discrete values and s is a helper string. Let S be a training set with m samples. If the trained classifier is (γ, S)-compressible with respect to $G_{\mathcal{W},s}$ using helper string s, then there exists $\boldsymbol{w} \in \mathcal{W}$ such that, with high probability over S,*

$$L(g_{\boldsymbol{w}}) \leq \widehat{L}_S^{\gamma}(f) + \mathcal{O}\left(\sqrt{\frac{q \log r}{m}}\right). \tag{82}$$

This approach allows an elegant proof of Theorem 21 and 31 for the compressed model. Furthermore, utilizing noise stability properties of neural networks which enable better compression (reminiscent of PAC-Bayes), they derive a bound specifically for deep models with ReLU activation ϕ. The following four definitions capture the model's resilience to the noise of Johnson-Lindenstrauss-like random compressions; intuitively, a model which is more resilient to noise compresses better and therefore yields tighter generalization bounds.

Definition 6 (Noise-Resilience Properties, [Arora et al., 2018] Def. 4-7)**.** *The layer cushion of layer i is the largest number μ_i such that for any $x \in S$,* ***Notation***

$$\mu_i \left\|\mathbf{W}_i\right\|_F \left\|\phi(x_{i-1})\right\| \leq \left\|\mathbf{W}_i \phi(x_{i-1})\right\|. \tag{83}$$

We have that $1/\mu_i^2$ is the noise sensitivity of $\mathbf{A}_i$ at $\phi(x_{i-1})$ with respect to standard Gaussian noise. Suppose $\mathbf{J}_{i,j}$ is the Jacobian of the operator $\mathbf{M}_{i,j}$ corresponding to the portion of the network between layers i and j, then the interlayer cushion of layers i, j is the largest number $\mu_{i,j}$ such that for any $x \in S$,

$$\mu_{i,j} \left\|\mathbf{J}_{i,j}[x_i]\right\|_F \left\|x_i\right\| \leq \left\|\mathbf{J}_{i,j}[x_i] x_i\right\|. \tag{84}$$

Furthermore, the minimal interlayer cushion of layer i is $\mu_{i\rightarrow} = \min_{i \leq j \leq D} \mu_{i,j}$. The activation contraction c is the smallest number such that for any layer i and $x \in S$,

$$\left\|\phi(x_i)\right\| \geq \left\|x_i\right\| / c. \tag{85}$$

Finally, the interlayer smoothness ρ_δ is the smallest number such that with probability at least $1-\delta$ over noise η and layers $i < j$ and any $x \in S$,

$$\left\|\mathbf{M}_{i,j}(x_i + \eta) - \mathbf{J}_{i,j}[x_i](x_i + \eta)\right\| \leq \frac{\left\|\eta\right\| \left\|x_j\right\|}{\rho_\delta \left\|x_i\right\|}. \tag{86}$$

See [Arora et al., 2018] for further explanation of these quantities.

Theorem 44 (Compression Generalization Bound for Neural Networks, [Arora et al., 2018] Thm. 4.1)**.** *For any fully connected network $f_{\boldsymbol{w}}$ with $\rho_\delta \geq 3D$, and $\delta, \gamma > 0$, there exists a (Johnson-Lindenstrauss-like) random compression $\tilde{\boldsymbol{w}}$ of $\boldsymbol{w}$ such that with probability $1-\delta$ over the training set S of size m and compressed model $f_{\tilde{\boldsymbol{w}}}$,*

$$L(f_{\tilde{\boldsymbol{w}}}) \leq \widehat{L}_S^\gamma(f_{\boldsymbol{w}}) + \mathcal{O}\left(\sqrt{\frac{c^2 D^2 \max_{x\in S} \|f_{\boldsymbol{w}}(x)\|_2^2 \sum_{i=1}^D \frac{1}{\mu_i^2 \mu_{i\rightarrow}^2}}{\gamma^2 m}}\right). \tag{87}$$

This framework proves the generalization of the compressed model $f_{\tilde{\boldsymbol{w}}}$ instead of the original model $f_{\boldsymbol{w}}$ just as PAC-Bayes bounds prove generalization of the expected/noised version of the original model. Similarly to how PAC-Bayes bounds have been de-randomized [Neyshabur et al., 2018, Nagarajan and Kolter, 2019a], it may also be possible to prove compression bounds with respect to the original model if certain technical conditions can be proved [Arora et al., 2018]. One approach, using the (uniform convergence-based) technique of local Rademacher complexity, is detailed in [Suzuki et al., 2020].

A subsequent work [Zhou et al., 2019] uses a PAC-Bayes approach to study the generalization of models compressed with a pruning, quantization, and Huffman coding scheme [Han et al., 2016]. In particular, suppose the output of the compression scheme is a triplet (S, C, Q) where $S = \{s_1, s_2, \ldots, s_k\}$ denotes the locations of the nonzero weights, $C = \{c_1, c_2, \ldots, c_r\}$ is a codebook, and $Q = \{q_1, q_2, \ldots, q_k\}$ are the quantized values. Then the corresponding network weights are

$$w_i(S, Q, C) = \begin{cases} c_{q_j} & \text{if } i = s_j, \\ 0 & \text{otherwise.} \end{cases} \tag{88}$$

They obtain a PAC-Bayes bound by applying independent random noise to the nonzero weights of the network, that is $\rho \sim \mathcal{N}(w, \sigma^2 \mathbf{J})$ where $\mathbf{J}_{ii} = 1$ if $i \in S$ and 0 otherwise.

Theorem 45 (KL Divergence of Compressed Hypotheses, [Zhou et al., 2019] Thm. 4.3.)**.** *Let (S, C, Q) be the output of a compression scheme, and let $\rho_{S,C,Q}$ be the stochastic estimator given by the weights decoded from the triplet and variance σ^2. Let c denote some arbitrary fixed coding scheme and let m denote an arbitrary distribution on the positive integers. Then for any $\tau > 0$, there is a PAC-Bayes prior π such that*

$$\begin{aligned}\mathrm{KL}(\rho_{S,C,Q}, \pi) \leq (k\lceil \log r\rceil + |S|_c + |C|_c)\log 2 - \log m(k\lceil \log r\rceil + |S|_c + |C|_c) \\ + \sum_{i=1}^k \mathrm{KL}(\mathcal{N}(c_{q_i}, \sigma^2), \sum_{j=1}^r \mathcal{N}(c_j, \tau^2)).\end{aligned} \tag{89}$$

This result can be substituted into a PAC-Bayes theorem such as Theorem 25 to obtain a generalization bound; their experiments show it is non-vacuous on ImageNet.

Another approach is based on the concept of model distillation [Hinton et al., 2014] and obtains bounds on the original, undistilled network [Hsu et al., 2021]. Given a multiclass predictor f, distillation finds another predictor g which is simpler, but close in distillation distance $\Phi_{\gamma,m}$ meaning the softmax outputs ϕ_γ are close on average over a set of points $\{z_i\}_{i=1}^m$:

$$\Phi_{\gamma,m}(f, g) = \frac{1}{m}\sum_{i=1}^m \|\phi_\gamma(f(z_i)) - \phi_\gamma(g(z_i))\|_1 \text{ where } \phi_\gamma(f(z)) \propto \exp(f(z)/\gamma). \tag{90}$$

The term γ here is often referred to as the softmax temperature, but is in fact naturally related to the margin, as decreasing γ increases sensitivity near the decision boundary. [Hsu et al., 2021] propose an analysis based on data augmentation. Suppose $\{(x_i, y_i)\}_{i=1}^n$ is drawn from a measure μ with marginal distribution $\mu_{\mathcal{X}}$. Then suppose we also have $\{z_i\}_{i=1}^m$ drawn from a data augmentation distribution ν_n which depends on $\{x_i\}_{i=1}^n$. Then, if the ratio $\left\|\frac{\mathrm{d}\mu_{\mathcal{X}}}{\mathrm{d}\nu_n}\right\|_\infty$ is finite and reasonable – they give an example where it is $\mathcal{O}(\sqrt{\log n}/n^{\alpha/(2\alpha+d)})$ for $\alpha \in [0,1]$ – we obtain the following generalization bound.

Theorem 46 (Distillation and Data Augmentation Generalization Bound, [Hsu et al., 2021] Lem. 1.1)**.** *Let temperature parameter $\gamma > 0$ be given along with sets of k-class predictors $\mathcal{F}$ and $\mathcal{G}$. Then with probability at least $1 - 2\delta$ over an iid draw of data $\{(x_i, y_i)\}_{i=1}^n$ from μ and $\{z_i\}_{i=1}^m$ from ν_n, every $f \in \mathcal{F}$ and $g \in \mathcal{G}$ satisfy*

$$\begin{aligned}\Pr[\operatorname*{argmax}_{y'} f(x)_{y'} \neq y] \leq 2\left\|\frac{\mathrm{d}\mu_{\mathcal{X}}}{\mathrm{d}\nu_n}\right\|_\infty \Phi_{\gamma,m}(f,g) + \frac{2}{n}\sum_{i=1}^m (1 - \phi_\gamma(g(x_i))_{y_i}) \\ + \widetilde{\mathcal{O}}\left(\frac{k^{3/2}}{\gamma}\left\|\frac{\mathrm{d}\mu_{\mathcal{X}}}{\mathrm{d}\nu_n}\right\|_\infty (\mathrm{Rad}_m(\mathcal{F}) + \mathrm{Rad}_m(\mathcal{G})) + \frac{\sqrt{k}}{\gamma}\mathrm{Rad}_n(\mathcal{G})\right) \\ + 6\sqrt{\frac{\log 1/\delta}{2n}}\left(1 + \left\|\frac{\mathrm{d}\mu_{\mathcal{X}}}{\mathrm{d}\nu_n}\right\|_\infty \sqrt{\frac{n}{m}}\right). \qquad (91)\end{aligned}$$

Here, $\mathrm{Rad}_m(\mathcal{F})$ is a multiclass extension of the empirical Rademacher complexity. This bound can be extended to general graph architectures such as ResNets via the covering numbers technique of [Bartlett et al., 2017]; see Theorem 1.3 in [Hsu et al., 2021] for more details.

6.3 Distance from Initialization

Since stochastic gradient descent seems essential for the performance of neural networks, its implicit regularization can give us clues as to what properties may be important for generalization. Sharpness is one candidate, but it is by no means the only optimization-based measure which is interesting to study. [Nagarajan and Kolter, 2017] empirically observe that SGD tends to implicitly regularize the distance from initialization – as the model becomes more overparameterized, the ℓ_2 distance from initialization of the SGD solution remains constant or even decreases. Utilizing this property enables an initialization-dependent analysis (as in PAC-Bayes) which removes the width dependence of Frobenius norm-based bounds. For a deep *linear* network, [Nagarajan and Kolter, 2017] show the following generalization bound.

Theorem 47 (Rademacher Complexity of Distance from Initialization-Bounded Linear Neural Networks, [Nagarajan and Kolter, 2017] Thm. 5.1)**.** *Suppose $(\boldsymbol{w}, \boldsymbol{b})$ is the collection of weights and biases of a trained neural network with linear activations, initialized at $(\boldsymbol{z}, \boldsymbol{0})$ where each parameter in $\boldsymbol{z}$ is sampled randomly as $\mathcal{N}(0, \mathcal{O}(1/\sqrt{H}))$. The empirical Rademacher complexity corresponding to the hypotheses an ℓ_2 distance of at most r from initialization is*

$$\mathbb{E}_{\boldsymbol{\sigma}}\Big[\sup_{(\boldsymbol{w},\boldsymbol{b}):\|(\boldsymbol{w},\boldsymbol{b})-(\boldsymbol{z},\boldsymbol{0})\|_F \leq r} \sum_{i=1}^m \sigma_i f_{(\boldsymbol{w},\boldsymbol{b})}(\boldsymbol{x}_i)\Big] = \widetilde{\mathcal{O}}\Big(\frac{Dc^D(r+1)^D \max_i \|\boldsymbol{x}_i\|}{\sqrt{m}}\Big), \qquad (92)$$

where $c = \widetilde{\Theta}(1)$.

The proof follows a peeling argument similar to Theorem 9 and 10. See the next section for a different interpretation of the distance-to-initialization idea.

6.4 Neural Tangent Kernel

Since overparameterization appears to benefit generalization, it is interesting to study what happens when the model is *as overparameterized as possible* – in other words, the infinite-width limit. It is a perhaps surprising fact that in this limit, deep neural networks at initialization are equivalent to Gaussian processes [Lee et al., 2018]. It is even more surprising that their behavior during training can also be described by a kernel, called the Neural Tangent Kernel (NTK) [Jacot et al., 2018].

The key observation is that for models of greater width, the weights don't change much from their random initialization during training. This is often called the *lazy regime*. If the weights don't change much, the model's first-order approximation via Taylor expansion – its *linearization* – is likely to be accurate. Under this approximation, the model is linear in the weights, but nonlinear in the input; it is a linear model using a feature map (the gradient at initialization), which, when studied in the context of GD training dynamics, induces the NTK.

A first generalization result [Arora et al., 2019] does not use the NTK specifically, but instead the Gram matrix of a kernel associated with the ReLU function defined as

$$\begin{aligned}\mathbf{H}^{\infty} &= \mathbb{E}_{\boldsymbol{w}\sim\mathcal{N}(0,\mathbf{I})}[\boldsymbol{x}_i^\top \boldsymbol{x}_j \mathbb{1}_{\boldsymbol{w}^\top \boldsymbol{x}_i \geq 0, \boldsymbol{w}^\top \boldsymbol{x}_j \geq 0}] &(93)\\ &= \frac{\boldsymbol{x}_i^\top \boldsymbol{x}_j(\pi - \arccos(\boldsymbol{x}_i^\top \boldsymbol{x}_j))}{2\pi}. &(94)\end{aligned}$$

If the matrix $\mathbf{H}^{\infty}$ is positive definite for a two-layer ReLU network, then gradient descent converges to 0 loss if m is large enough [Du et al., 2019]. Bounding the distance to initialization of the weights and a Rademacher complexity analysis yields the following generalization bound.

Theorem 48 (Kernel Generalization Bound for Two-Layer ReLU Networks, [Arora et al., 2019] Cor. 5.2)**.** *A distribution $\mathcal{D}$ is called $(\lambda_0, \delta, m$-non-degenerate if, for m iid samples $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$ from $\mathcal{D}$, with probability at least $1-\delta$ we have $\lambda_{\min}(\mathbf{H}^{\infty}) \geq \lambda_0 \geq 0$. Fix δ and suppose $\mathcal{D}$ is a $(\lambda_0, \delta/3, m)$-non-degenerate distribution and $\kappa = \mathcal{O}(\frac{\lambda_0 \delta}{m})$, $m \geq \kappa^{-2}\mathrm{poly}(n, \lambda_0^{-1}, \delta^{-1})$. Consider any centered and 1-Lipschitz loss function ℓ. Then with probability at least $1-\delta$ over the random initialization and training samples, the two-layer neural network $f_{\boldsymbol{w}}$ trained on S by gradient descent with learning rate η for $k \geq \Omega(\frac{1}{\eta\lambda_0}\log\frac{m}{\delta})$ iterations has*

$$L(f_{\boldsymbol{w}}) \leq \sqrt{\frac{2\boldsymbol{y}^\top(\mathbf{H}^{\infty})^{-1}\boldsymbol{y}}{m}} + \mathcal{O}\left(\sqrt{\frac{\log\frac{m}{\lambda_0\delta}}{m}}\right). \tag{95}$$

Note that the bound is independent of network width H and the trained values of the weights. A sharper bound which uses the NTK to generalize this result to neural networks of any depth was introduced by [Cao and Gu, 2019]. Instead of $\mathbf{H}^{\infty}$, they utilize the NTK matrix, defined as follows. Let σ denote the ReLU function, then

$$\widetilde{\mathbf{\Theta}}_{i,j}^{(1)} = \mathbf{\Sigma}_{i,j}^{(1)} = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle, \quad \mathbf{A}_{ij}^{(l)} = \begin{bmatrix} \mathbf{\Sigma}_{i,i}^{(l)} & \mathbf{\Sigma}_{i,j}^{(l)} \\ \mathbf{\Sigma}_{i,j}^{(l)} & \mathbf{\Sigma}_{j,j}^{(l)} \end{bmatrix}, \tag{96}$$

$$\mathbf{\Sigma}_{i,j}^{(l+1)} = 2\mathbb{E}_{(u,v)\sim\mathcal{N}(\mathbf{0},\mathbf{A}_{ij}^{(l)})}[\sigma(u)\sigma(v)], \tag{97}$$

$$\widetilde{\mathbf{\Theta}}_{i,j}^{(l+1)} = \widetilde{\mathbf{\Theta}}_{i,j}^{(l)} \cdot 2 \cdot \mathbb{E}_{(u,v)\sim\mathcal{N}(\mathbf{0},\mathbf{A}_{ij}^{(l)})}[\sigma'(u)\sigma'(v)] + \mathbf{\Sigma}_{i,j}^{(l+1)}. \tag{98}$$

Then, $\mathbf{\Theta}^{(D)} = [\widetilde{\mathbf{\Theta}}_{i,j}^{(D)} + \mathbf{\Sigma}_{i,j}^{(D)}/2]_{m\times m}$ is called the NTK matrix of a D-depth ReLU network on inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$. [Jacot et al., 2018] show that $\mathbf{\Theta}^{(D)}$ is the infinite-width limit of the Gram matrix of the model gradients, and if $\mathbf{\Theta}^{(D)}$ is positive-definite, then gradient descent converges to 0 loss if m is large enough. A random features analysis gives the following generalization bound.

Theorem 49 (NTK Generalization Bound for ReLU Networks, [Cao and Gu, 2019] Cor. 3.10)**.** *Let* $\lambda_0 = \lambda_{\min}(\mathbf{\Theta}^{(D)}$*. For any* $\delta \in [0, e^{-1}]$*, there exists* $\tilde{m}^\star(\delta, D, m, \lambda_0)$ *such that if* $m \geq \tilde{m}^\star$*, then with probability at least* $1-\delta$ *over the initialization, a variant of gradient descent with small enough step size satisfies*

$$\mathbb{E}[L(f_{\boldsymbol{w}})] \leq \widetilde{\mathcal{O}}\left(\sqrt{\frac{\boldsymbol{y}^\top(\widetilde{\mathbf{\Theta}}^{(D)})^{-1}\boldsymbol{y}}{m}}\right) + \mathcal{O}\left(\sqrt{\frac{\log 1/\delta}{m}}\right). \tag{99}$$

There has been a significant amount of work in this setting, and we only highlighted a couple early influential papers. Other important works include [Allen-Zhu et al., 2019], who propose a second-order variant of NTK, and [Chen et al., 2020], who extend the NTK generalization bounds to regimes with explicit regularization such as weight decay.

6.5 Information Theory

Recall from Section 6.1 that an algorithm is stable if a small change to the dataset does not affect the loss too much, and stability implies generalization. Stability can also be defined information-theoretically – for example, in differential privacy [Dwork et al., 2006] – and has since seen application to generalization of learning algorithms. A natural information-theoretic measure of the dependence between S and $h := (A(S))$ for a randomized algorithm A is the mutual information between them, denoted $I(S; h) := \mathrm{KL}(P_{(S,h)} \| P_S \otimes P_h)$. This approach was introduced by [Russo and Zou, 2016, 2019] for adaptive data analysis and improved for statistical learning by [Xu and Raginsky, 2017]. Suppose $L(h)$ and $L_S(h)$ are defined with respect to a loss function $\ell : \mathcal{H} \times \mathcal{X} \to \mathbb{R}^+$. An application of the Donsker-Varadhan variational formula yields the following bound.

Theorem 50 (Mutual Information Generalization Bound, [Xu and Raginsky, 2017])**.** *Suppose* ℓ *is* σ*-subgaussian under* $\mathcal{D}$ *for all* $h \in \mathcal{H}$*, then for a sample* S *of size* m*,*

$$\left|\mathbb{E}[L(h) - \widehat{L}_S(h)]\right| \leq \sqrt{\frac{2\sigma^2 I(S; h)}{m}}. \tag{100}$$

Note that bounded loss functions are subgaussian. The main issue with this bound is that mutual information can be unbounded in even simple settings such as linear regression and threshold learning due to continuous data distributions. [Steinke and Zakynthinou, 2020] propose conditional mutual information (CMI) as a solution, which they interpret as "normalizing" the information content of each datapoint to one bit. Formally, let Z be a sample of $2m$ points drawn from $\mathcal{D}$, $S \in \{0,1\}^m$ be drawn uniformly at random, and Z_S be the subset of Z indexed by S. Then,

$$\mathrm{CMI}_{\mathcal{D}}(A) = I(A(Z_S); S|Z), \tag{101}$$

and it is bounded by $m \log 2$. They obtain the following bound based on CMI.

Theorem 51 (Conditional Mutual Information Generalization Bound, [Steinke and Zakynthinou, 2020] Thm. 1.3)**.** *Suppose* ℓ *is an unbounded loss function, then for a sample* S *of size* m*,*

$$\left|\mathbb{E}[L(h) - \widehat{L}_S(h)]\right| \leq \sqrt{\frac{8\mathrm{CMI}_{\mathcal{D}}(A)\mathbb{E}_x[\sup_{h\in\mathcal{H}}(\ell(h,x))^2]}{m}}. \tag{102}$$

Extensions to this work include [Haghifam et al., 2020] who show CMI bounds are empirically nonvacuous for neural networks, and [Haghifam et al., 2021] who show that the CMI framework is sufficient to recover the optimal bound on the error of SVMs.

A significant generalization of the mutual information approach is due to [Lugosi and Neu, 2022], who show that the technique is not strictly information-theoretic at all – in fact, the mutual information term may be replaced by any strongly convex function of the joint distribution. Suppose F is a dependence measure given by $F(P) = \mathbb{E}_S[f(P_{|S})]$.

Theorem 52 (Convex Analysis Generalization Bound, [Lugosi and Neu, 2022])**.** *Let f be an α-strongly convex function with respect to the norm $\|\cdot\|$ and let $\|\cdot\|_*$ denote its dual norm. Then for a sample S of size m,*

$$\left|\mathbb{E}[L(h) - \widehat{L}_S(h)]\right| \leq \sqrt{\frac{4F(P_{(S,h)})\mathbb{E}_x\big[\,\big\|\sup_{h\in\mathcal{H}}\left|\ell(h,x) - \mathbb{E}_x[\ell(h,x)]\right|\big\|_*^2\,\big]}{\alpha m}}. \tag{103}$$

Interesting functions enabled by this framework include KL divergence (recovering the mutual information bound), p-norm divergences, and Wasserstein distances. Other references in this area include [Bu et al., 2020] who show that the single-datapoint mutual information $I(x;h)$ is sufficient and [Hellström and Durisi, 2020] who highlight connections to PAC-Bayes theory.

6.6 *A Posteriori* Bounds

Most generalization bounds explored in this survey are *a priori* – they use the model complexity to estimate the generalization gap, and one substitutes the empirical error to obtain a bound on generalization. With some *a priori* techniques such as uniform convergence being questioned [Nagarajan and Kolter, 2019b], there has been recent interest in an *a posteriori* approach, which estimates the generalization of the post-training model based on properties evaluated on an unlabeled dataset [Chatterjee, 2020, Garg et al., 2021, Jiang et al., 2022]. Often, these analyses are informed by properties of the optimizer such as stability [Bousquet and Elisseeff, 2002, Hardt et al., 2016] and the early-learning phenomenon, where SGD seems to fit signal before noise (or simpler functions before more complicated ones) [Zhang et al., 2017, Li et al., 2020].

[Garg et al., 2021] propose to augment the training set with randomly labeled data and evaluate the errors on the clean and noisy data individually. They show that the error on the mislabeled training data upper bounds the error on the mislabeled population – if the model fits the clean data well but attains high error on the noisy data, then we will have good generalization. Formally, suppose we have a clean dataset $S \sim \mathcal{D}^n$ and a randomly labeled dataset $\tilde{S} \sim \mathcal{D}^m$ where $m < n$.

Theorem 53 (RATT Generalization Bound, [Garg et al., 2021] Thm. 1, Prop. 1)**.** *For any 0-1 ERM classifier f trained on $S \cup \tilde{S}$, for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of S and $\tilde{S}$, we have*

$$L(f) \leq L_S(f) + (1 - 2L_{\tilde{S}}(f)) + \left(\sqrt{2}L_{\tilde{S}}(f) + 2 + \frac{m}{2n}\right)\sqrt{\frac{\log 4/\delta}{m}}. \tag{104}$$

Furthermore, with high probability we have $1 - 2L_{\tilde{S}}(f) = \mathcal{O}\big(\mathfrak{R}(f) + \sqrt{\frac{\log 1/\delta}{m}}\big)$, so this is never worse than a Rademacher complexity bound.

They also present results for models trained with SGD under the squared loss, which has similar properties to cross-entropy loss [Muthukumar et al., 2021]. The proof requires a stability property of [Bousquet and Elisseeff, 2002] called hypothesis stability. Denote f the function obtained by training on S and $f_{(i)}$ the function obtained by training on S with the i^{th} point removed. An algorithm A has hypothesis stability β if for all $i \in [n]$,

$$\mathbb{E}_{S,x\sim\mathcal{D}}\left[\left|\ell(f(x),x) - \ell(f_{(i)}(x),x)\right|\right] \leq \frac{\beta}{n}. \tag{105}$$

Theorem 54 (RATT SGD Generalization Bound, [Garg et al., 2021] Thm. 4)**.** *Suppose SGD on $S \cup \tilde{S}$ under the squared loss has hypothesis stability β. Then for any $\delta > 0$, with probability at least $1 - \delta$ over the choice of S and $\tilde{S}$, we have*

$$L(f) \leq L_S(f) + (1 - 2L_{\tilde{S}}(f)) + \sqrt{\frac{4}{\delta}\Big(\frac{1}{m} + \frac{3\beta}{m+n}\Big)} + \Big(\sqrt{2}L_{\tilde{S}}(f) + 1 + \frac{m}{2n}\Big)\sqrt{\frac{\log 4/\delta}{m}}. \tag{106}$$

Notably, this bound may help to explain SGD generalization using early stopping, since the model is likely to fit S but not $\tilde{S}$ in the first stages of learning. However, without early stopping, it is likely that the model will completely interpolate $S \cup \tilde{S}$ and render this bound vacuous.

7 Bibliography

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. In *Conference on Neural Information Processing Systems*, 2019. https://arxiv.org/abs/1811.04918. 27

Pierre Alquier. User-friendly introduction to PAC-Bayes bounds, 2021. https://arxiv.org/abs/2110.11216. 3

Martin Anthony and Peter L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, first edition, 1999. 4, 5, 11, 12

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning (ICML)*, 2018. https://arxiv.org/abs/1802.05296. 22, 23, 24

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning (ICML)*, 2019. https://arxiv.org/abs/1901.08584. 26

Gregor Bachmann, Seyed-Mohsen Moosavi-Dezfooli, and Thomas Hofmann. Uniform convergence, adversarial spheres, and a simple remedy. In *International Conference on Machine Learning (ICML)*, 2021. https://arxiv.org/abs/2105.03491. 20

Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research (JMLR)*, 3:463–482, 2002. https://www.jmlr.org/papers/volume3/bartlett02a/bartlett02a.pdf. 8

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. https://arxiv.org/abs/1706.08498. 9, 12, 13, 18, 25, 39

Peter L. Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research (JMLR)*, 23(63):1–17, 2019. https://arxiv.org/abs/1703.02930. 5

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning practice and the classical bias-variance tradeoff. *Proceedings of the National Academy of Sciences (PNAS)*, 116(32):15849–15854, 2019. https://www.pnas.org/doi/10.1073/pnas.1903070116. 2

Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research (JMLR)*, 2:499–526, 2002. https://www.jmlr.org/papers/volume2/bousquet02a/bousquet02a.pdf. 20, 21, 22, 28

Yuheng Bu, Shaofeng Zou, and Venugopal V. Veeravalli. Tightening mutual information-based bounds on generalization error. *Journal on Selected Areas in Information Theory (JSAIT)*, 1: 121–130, 2020. https://arxiv.org/abs/1901.04609. 28

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1905.13210. 26, 27

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations (ICLR)*, 2020. https://arxiv.org/abs/2002.10657. 28, 40

Niladri Chatterji, Behnam Neyshabur, and Hanie Sedghi. The intriguing role of module criticality in the generalization of deep networks. In *International Conference on Learning Representations (ICLR)*, 2020. https://arxiv.org/abs/1912.00528. 18

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations (ICLR)*, 2017. https://arxiv.org/abs/1611.01838. 16

Zixiang Chen, Yuan Cao, Quanquan Gu, and Tong Zhang. A generalized neural tangent kernel analysis for two-layer neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2002.04026. 27

Alex Damian, Tengyu Ma, and Jason D. Lee. Label noise SGD provably prefers flat global minimizers. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. https://arxiv.org/abs/2106.06530. 17, 39

Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, and Martin Riedmiller. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602:414–419, 02 2022. https://www.nature.com/articles/s41586-021-04301-9. 2

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning (ICML)*, 2017. https://arxiv.org/abs/1703.04933. 16

Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations (ICLR)*, 2019. https://arxiv.org/abs/1810.02054. 26

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, 2006. https://people.csail.mit.edu/asmith/PS/sensitivity-tcc-final.pdf. 27

Gintare Karolina Dziugate and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017. https://arxiv.org/abs/1703.11008. 6, 16, 17, 23, 39

Gamaleldin Fathy Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018. https://arxiv.org/abs/1803.05598. 14

Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. http://arxiv.org/abs/1707.08945. 2

Vitaly Feldman and Jan Vondrak. Generalization bounds for uniformly stable algorithms. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018. https://arxiv.org/abs/1812.09859. 22

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021. https://arxiv.org/abs/2010.01412. 18, 39, 40

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *International Conference on Learning Representations (ICLR)*, 2019. https://arxiv.org/abs/1803.03635. 22

Saurabh Garg, Sivaraman Balakrishnan, J. Zico Kolter, and Zachary C. Lipton. RATT: Leveraging unlabeled data to guarantee generalization. In *International Conference on Machine Learning (ICML)*, 2021. https://arxiv.org/abs/2105.00303. 28, 29

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference on Learning Theory (COLT)*, 2018. https://arxiv.org/abs/1712.06541. 8, 9

Benjamin Guedj. A primer on PAC-Bayesian learning. *Proceedings of the Congress of the Société Mathématique de France*, 2(1):391–414, 2019. 3

Mahdi Haghifam, Jeffrey Negrea, Ashish Khisti, Daniel M. Roy, and Gintare Karolina Dziugaite. Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2004.12983. 27

Mahdi Haghifam, Gintare Karolina Dziugaite, Shay Moran, and Daniel M. Roy. Towards a unified information-theoretic framework for generalization. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. https://arxiv.org/abs/2111.05275. 27

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2015. https://arxiv.org/abs/1506.02626. 22

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016. https://arxiv.org/abs/1510.00149. 24

Moritz Hardt and Benjamin Recht. *Patterns, predictions, and actions: A story about machine learning*. https://mlstory.org, 2021. 3

Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning (ICML)*, 2016. https://arxiv.org/abs/1509.01240. 21, 22, 28, 40

Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1902.00744. 17

Fredrik Hellström and Giuseppe Durisi. Generalization bounds via information density and conditional information density. *Journal on Selected Areas in Information Theory (JSAIT)*, 1:824–839, 2020. https://arxiv.org/abs/2005.08044. 28

Geoffrey Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Conference on Learning Theory*, 1993. https://dl.acm.org/doi/10.1145/168304.168306. 14

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Conference on Neural Information Processing Systems (NeurIPS) Deep Learning Workshop*, 2014. https://arxiv.org/abs/1503.02531. 22, 24

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997. http://www.bioinf.jku.at/publications/older/3304.pdf. 14

Daniel Hsu, Ziwei Ji, Matus Telgarsky, and Lan Wang. Generalization bounds via distillation. In *International Conference on Learning Representations (ICLR)*, 2021. https://arxiv.org/abs/2104.05641. 24, 25, 40

Sergey Ioffe and Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015. https://arxiv.org/abs/1502.03167. 16

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018. https://arxiv.org/abs/1803.05407. 18

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2018. https://arxiv.org/abs/1806.07572. 26

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations (ICLR)*, 2020. https://arxiv.org/abs/1912.02178. 3, 13, 17, 39, 40

Yiding Jiang, Parth Natekar, Manik Sharma, Sumukh K. Aithal, Dhruva Kashyap, Natarajan Subramanyam, Carlos Lassance, Daniel M. Roy, Gintare Karolina Dziugaite, Suriya Gunasekar, Isabelle Guyon, Pierre Foret, Scott Yak, Hossein Mobahi, Behnam Neyshabur, and Samy Bengio. Methods and analysis of the first competition in predicting generalization of deep learning. *Proceedings of Machine Learning Research (PMLR) NeurIPS 2020 Competition and Demonstration Track*, 133(1):170–190, 2021. 40

Yiding Jiang, Vaishnavh Nagarajan, Christina Baek, and J. Zico Kolter. Assessing generalization of SGD via disagreement. In *International Conference on Learning Representations (ICLR)*, 2022. https://openreview.net/pdf?id=WvOGCEAQhxl. 28

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. In *International Conference on Learning Representations (ICLR)*, 2017. https://arxiv.org/abs/1609.04836. 16, 17, 18, 40

Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of Statistics*, 30(1):1–50, 2002. https://arxiv.org/abs/math/0405343. 8

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. 7

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2012. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf. 7

Ilja Kuzborskij and Christoph H. Lampert. Data-dependent stability of stochastic gradient descent. In *International Conference on Machine Learning (ICML)*, 2018. https://arxiv.org/abs/1703.01678. 22

John Langford and Rich Caruana. (not) bounding the true error. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2001. https://www.cs.cmu.edu/~jcl/papers/nn_bound/not_bound.pdf. 16, 17

John Langford and Matthias Seeger. Bounds for averaging classifiers. Technical report, Carnegie Mellon University, 2001. https://www.cs.cmu.edu/~jcl/papers/averaging/averaging_tech.pdf. 15

Yann LeCun, Corinna Cortes, and C.J. Burges. Mnist handwritten digit database. *AT&T Labs*, 2, 2010. http://yann.lecun.com/exdb/mnist. 6, 17

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S. Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2018. https://arxiv.org/abs/1711.00165. 26

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020. https://arxiv.org/abs/1903.11680. 28

Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-Rao Metric, geometry, and complexity of neural networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019. https://arxiv.org/abs/1711.01530. 8

Roi Livni and Shay Moran. A limitation of the PAC-Bayes framework. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2006.13508. 16

Gábor Lugosi and Gergely Neu. Generalization bounds via convex analysis. In *Conference on Learning Theory (COLT)*, 2022. https://arxiv.org/abs/2202.04985. 28

Kaifeng Lyu, Zhiyuan Li, Runzhe Wang, and Sanjeev Arora. Gradient descent on two-layer nets: margin maximization and simplicity bias. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021. https://arxiv.org/abs/2110.13905. 12, 39

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. https://arxiv.org/abs/1706.06083. 14

David A. McAllester. Some PAC-Bayesian theorems. In *Conference on Learning Theory (COLT)*, 1998. https://dl.acm.org/doi/pdf/10.1145/279943.279989. 15

David A. McAllester. PAC-Bayesian model averaging. In *Conference on Learning Theory (COLT)*, 1999. https://dl.acm.org/doi/pdf/10.1145/307400.307435. 15

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. MIT Press, second edition, 2018. https://cs.nyu.edu/~mohri/mlbook/. 3, 4, 6, 7, 10

Rotem Mulayoff and Tomer Michaeli. Unique properties of flat minima in deep networks. In *International Conference on Machine Learning (ICML)*, 2020. https://arxiv.org/abs/2002.04710. 17

Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *Journal of Machine Learning Research (JMLR)*, 22:1–69, 2021. https://arxiv.org/abs/2005.08054. 28

Vaishnavh Nagarajan and J. Zico Kolter. Generalization in deep networks: The role of distance from initialization. In *Conference on Neural Information Processing Systems (NeurIPS) Workshop on Deep Learning: Bridging Theory and Practice*, 2017. https://arxiv.org/abs/1901.01672. 25, 39, 40

Vaishnavh Nagarajan and J. Zico Kolter. Deterministic PAC-Bayesian generalization bounds for deep networks via generalizing noise-resilience. In *International Conference on Learning Representations (ICLR)*, 2019a. https://arxiv.org/abs/1905.13344. 19, 24

Vaishnavh Nagarajan and J. Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019b. https://arxiv.org/abs/1902.04742. 20, 22, 28, 39

Preetum Nakkiran and Yamini Bansal. Distributional generalization: A new kind of generalization, 2020. https://arxiv.org/abs/2009.08092. 39

Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. In *International Conference on Learning Representations (ICLR)*, 2021. https://arxiv.org/abs/2010.08127. 22

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 (ICML)*, 2020. https://arxiv.org/abs/1912.04265. 20

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In *International Conference on Learning Representations (ICLR)*, 2015a. Workshop track. https://arxiv.org/abs/1412.6614. 7

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory (COLT)*, 2015b. https://arxiv.org/abs/1503.00036. 8

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. https://arxiv.org/abs/1706.08947. 3, 13, 16, 17, 39, 40

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations (ICLR)*, 2018. https://arxiv.org/abs/1707.09564. 18, 24

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations (ICLR)*, 2019. https://arxiv.org/abs/1805.12076. 13

Sasha Rakhlin. Sample complexity of neural networks, 2019. https://www.mit.edu/~9.520/fall19/slides/Class21.pdf. 8

Jorma Rissanen. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, 11(2):416–431, 1983. https://www.ccs.neu.edu/home/jaa/CS188.01W/Papers/Rissanen83.pdf. 14

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016. http://proceedings.mlr.press/v51/russo16.html. 27

Daniel Russo and James Zou. How much does your data exploration overfit? controlling bias via information usage. *IEEE Transactions on Information Theory*, 66:302–323, 2019. https://arxiv.org/abs/1511.05219. 27

Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory*, 13(1):145–147, 1972. https://www.sciencedirect.com/science/article/pii/0097316572900192. 4

Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability, and uniform convergence. *Journal of Machine Learning Research (JMLR)*, 2010. https://dl.acm.org/doi/pdf/10.5555/1756006.1953019. 21, 22

Saharon Shelah. A combinatorial problem: stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972. https://msp.org/pjm/1972/41-1/p21.xhtml. 4

Umut Şimşekli, Levent Sagun, and Mert Gürbüzbalaban. A tail-index analysis of stochastic gradient descent noise in deep neural networks. In *International Conference on Machine Learning (ICML)*, 2019. http://proceedings.mlr.press/v97/simsekli19a/simsekli19a.pdf. 40

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *Journal of Machine Learning Research (JMLR)*, 19(70):1–57, 2018. https://arxiv.org/abs/1710.10345. 12, 39

Thomas Steinke and Lydia Zakynthinou. Reasoning about generalization via conditional mutual information. In *Conference on Learning Theory (COLT)*, 2020. https://arxiv.org/abs/2001.09122. 27

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. http://arxiv.org/abs/1912.04838. 2

Taiji Suzuki, Hiroshi Abe, and Tomoaki Nishimura. Compression based bound for non-compressed network: Unified generalization error analysis of large compressible deep neural network. In *International Conference on Learning Representations (ICLR)*, 2020. https://arxiv.org/abs/1909.11274. 24

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014. http://arxiv.org/abs/1312.6199. 2

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. https://arxiv.org/abs/1512.00567. 7

Matus Telgarsky. Benefits of depth in neural networks. In *Conference on Learning Theory (COLT)*, 2016. https://arxiv.org/abs/1602.04485. 7

Matus Telgarsky. Deep learning theory lecture notes, 2021. https://mjt.cs.illinois.edu/dlt/. 3

Ambuj Tewari and Peter L. Bartlett. Learning theory. *Signal Processing Theory and Machine Learning*, 1(14):775–816, 2014. https://ambujtewari.github.io/research/tewari13learning.pdf. 3, 11

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 11 1984. http://web.mit.edu/6.435/www/Valiant84.pdf. 3

Guillermo Valle-Pérez and Ard A. Louis. Generalization bounds for deep learning, 2020. https://arxiv.org/abs/2012.04115. 3

Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Springer, first edition, 1995. 21

Vladimir N. Vapnik and Alexey Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16(2):264–280, 1971. Translated by B. Seckler. https://epubs.siam.org/doi/10.1137/1116025. 3

Colin Wei and Tengyu Ma. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. https://arxiv.org/abs/1905.03684. 13

Colin Wei and Tengyu Ma. Improved sample complexities for deep neural networks and robust classification via an all-layer margin. In *International Conference on Learning Representations (ICLR)*, 2020. https://arxiv.org/abs/1910.04284v5. 14, 39

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *International Conference on Machine Learning (ICML)*, 2020. https://arxiv.org/abs/2002.08791. 39

Zeke Xie, Issei Sato, and Masashi Sugiyama. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations (ICLR)*, 2021. https://arxiv.org/abs/2002.03495. 17, 40

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. https://arxiv.org/abs/1705.07809. 27

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017. https://arxiv.org/abs/1611.03530. 7, 28

Chiyuan Zhang, Samy Bengio, and Yoram Singer. Are all layers created equal? *Journal of Machine Learning Research (JMLR)*, 23:1–28, 2022. https://arxiv.org/abs/1902.01996. 18

Lijia Zhou, Danica J. Sutherland, and Nathan Srebro. On uniform convergence and low-norm interpolation learning. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2006.05942. 20

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz. Non-vacuous generalization bounds at the ImageNet scale: A PAC-Bayesian compression approach. In *International Conference on Learning Representations (ICLR)*, 2019. https://arxiv.org/abs/1804.05862. 24

A Frequently Asked Questions (F.A.Q.)

Do we understand generalization in deep learning? Not yet!

Why is it important to understand generalization in deep learning? The situation is dire; we are deploying deep neural networks in high-consequence applications without a clear picture of when or why they succeed or fail. As a fundamental question in learning theory, understanding generalization will likely shed new light on practical considerations such as adversarial robustness, model fairness, and out-of-domain generalization. Additionally, with a clear generalization measure, we can regularize explicitly towards a more generalizable solution (as in the SVM), which is already shown to increase performance in difficult learning tasks [Wei and Ma, 2020, Foret et al., 2021].

Why is generalization in deep learning different from classical ML? Overparameterized deep neural networks generalize well without overfitting, contradicting the bias-variance tradeoff and rendering classical bounds vacuous. This phenomenon seems intricately connected to interactions within the (data, model, optimizer) triplet. For example, in the SVM, any optimizer which minimizes the loss (maximizes the margin) enjoys strong generalization bounds. In deep learning, generalization depends on the particular minima found by the optimizer; so, the algorithm, initialization, and learning rate may impact generalization [Neyshabur et al., 2017].

Why are classical generalization bounds vacuous for overparameterized deep neural networks? Classical bounds, such as VC-dimension bounds, crucially depend on the ratio between the total number of parameters in the model and the number of training data. In even moderately overparameterized neural networks, this ratio is very large, so the bounds essentially predict a generalization error of at most 1. Thus, more sophisticated measures are required.

What does implicit regularization have to do with generalization? There are several solution-dependent complexity measures associated with better generalization – for example, a larger margin or a flatter minima are both proven to result in more generalizable solutions [Bartlett et al., 2017, Dziugate and Roy, 2017]. However, this does not answer the question of why neural networks trained with gradient descent actually find solutions with these amicable properties without any explicit regularization. This phenomenon is termed the implicit regularization of SGD and is an active area of research [Soudry et al., 2018, Lyu et al., 2021, Damian et al., 2021]. Conversely, if SGD satisfies a certain simplicity property – *e.g.,* it does not stray too far from its initialization – that property may be a candidate for explaining generalization [Nagarajan and Kolter, 2017].

Why doesn't a margin-based analysis work for deep learning? A margin-based analysis "works" in the sense that the confidence margin of the model, properly normalized, yields plausible generalization bounds [Bartlett et al., 2017, Wei and Ma, 2020]. However, there are a few problems. First, in contrast to linear classifiers, it is difficult to pin down the proper definition of margin and corresponding weight norms. Second, these bounds are often exponential in the depth or rely on local Lipschitzness properties which are complicated to analyze. Finally, [Jiang et al., 2020] show a surprising empirical failure of margin-based complexity measures; in particular, margin-based spectral complexity is highly *negatively* correlated with generalization.

What are some promising future directions? While studying which complexity measures are most important for generalization is intriguing and useful, an important meta-topic is which generalization framework is most appropriate for deep learning. The uniform convergence, PAC-Bayes, and uniform stability frameworks each have their own strengths and weaknesses, and proving fundamental limits on each of these methods in the sense of [Nagarajan and Kolter, 2019b] is critical for determining which of these frameworks, if any, are the "right" way to analyze generalization in deep learning. Or, perhaps a new, more comprehensive framework is needed; notably, recent works have proposed an alternative distributional or Bayesian perspective [Nakkiran and Bansal, 2020, Wilson and Izmailov, 2020].

As sharpness is currently the best candidate for explaining generalization [Jiang et al., 2020], its properties should be investigated in greater detail. In particular, the impact of batch- or accelerator-wise stochasticity on per-data-point sharpness is intriguing and appears to be a better predictor of generalization than global sharpness [Foret et al., 2021]. Another potential avenue is characterizing the difference between worst-case (adversarial) and expected sharpness, as expected sharpness fits better into the PAC-Bayes framework but adversarial sharpness is a better empirical indicator of generalization [Keskar et al., 2017, Neyshabur et al., 2017, Jiang et al., 2020].

Another interesting direction is to analyze the optimization properties of SGD and how they may affect generalization. In particular, [Jiang et al., 2020] find that the variance of the gradients is empirically indicative of generalization, and [Şimşekli et al., 2019, Xie et al., 2021] show theoretically that the gradient noise enables escaping sharp minima. Initialization-based properties, such as distance traveled by SGD [Nagarajan and Kolter, 2017], data-based properties, such as robustness to data augmentation [Hsu et al., 2021, Jiang et al., 2021], and stability-based properties [Hardt et al., 2016, Chatterjee, 2020] may also be promising candidates.

B Notation

Symbol	Definition
$\mathcal{X}$	Input space with members x or $\boldsymbol{x}$ and labels y; often $\mathbb{R}^n$
$\mathcal{D}$	Data distribution
$\mathcal{H}$	Hypothesis class with members h
$\mathbb{E}$	Expectation
$f^\star$	Labeling function
$f_{\boldsymbol{w}}$	Function computed by a neural network with weight vector $\boldsymbol{w}$
S	Training sample of size m
W	Number of parameters (weights and thresholds)
D	Depth of the neural network (including output layer)
H	Width of the neural network
L	Generalization error
$\widehat{L}_S$	Empirical error on sample S
VC-dim	VC-dimension; also written d
$\Pi_{\mathcal{H}}$	Growth function of class $\mathcal{H}$
$\widehat{\mathfrak{R}}_S$	Empirical Rademacher complexity on sample S
$\mathfrak{R}_m$	Rademacher complexity on m points
δ	Failure probability or small perturbation
ReLU	Rectified linear unit
σ	Sigmoid function or Rademacher random variable
γ	Margin
Φ_γ	γ-margin loss
$\widehat{L}_S^\gamma$	Empirical margin error on sample S
λ	Regularization parameter
$\mathcal{N}$	Covering number or Gaussian distribution
$\mathrm{fat}_{\mathcal{H}}$	Fat-shattering dimension of real-valued class $\mathcal{H}$
KL	KL divergence
A	Algorithm producing a classifier h given input sample S
β	Stability constant
I	Mutual information

Table 1: Notation used in this work, roughly in order of introduction. Unless otherwise specified, a and A are scalars, $\boldsymbol{a}$ is a vector, and $\mathbf{A}$ is a matrix or tensor.