GENERALIZATION BOUNDS FOR NEURAL NETWORKS

Siqi Miao
siqim2@illinois.edu

1 INTRODUCTION

Hornik (1991) showed that neural networks with a single hidden layer and arbitrary bounded and non-constant activation function are capable to approximate any continuous functions on compact subsets of $\mathbb{R}^n$. Besides their great approximation capabilities, neural networks also show great generalization ability on many tasks. However, a classical opinion is that a too capable model would usually generalize poorly. A natural question to ask is that what is the generalization bound of neural networks?

Thus, we study three generalization bounds for neural networks hoping to obtain more insights about their generalization abilities. The three bounds that will be shown later are derived from VC-dimension, PAC-Bayesian theorem and covering number respectively. In this paper we would introduce the first two bounds in more detail and briefly present the third bound. To compare the three bounds more conveniently, throughout this paper we only consider a L-layer neural network with ReLU activations.

2 GENERALIZATION BOUND DERIVED FROM VC-DIMENSION

From the fundamental theorem of concept learning, which gives a generalization bound based on VC-dimension, one intuitive way to obtain a bound for neural networks is to find the VC-dimension of the function class computed by them. Thus, in this section we first introduce some preliminaries that we will use later. Then we present a generalization bound for neural networks based on VC-dimension due to Bartlett et al. (2017b), which proved a nearly-tight VC-dimension bound for networks with ReLU activations.

2.1 PRELIMINARIES

Definition 2.1. The growth function of $\mathcal{F} : \mathcal{X} \mapsto \{-1, 1\}$ is defined as

$$\mathbb{S}_m(\mathcal{F}) := \max_{x_1,\ldots,x_m \in \mathcal{X}} |\{(f(x_1), ..., f(x_m)) : f \in \mathcal{F}\}|.$$

The largest m such that $\mathbb{S}_m(\mathcal{F}) = 2^m$ is defined as the VC-dimension of $\mathcal{F}$.

Theorem 2.1. Let $f \in \mathcal{F} : \mathcal{X} \mapsto \{-1, 1\}$ be the output of a concept learning algorithm, L be the expected risk, $\hat{L}$ be the empirical risk, n be the number of training samples and $V(\mathcal{F})$ be the VC-dimension of a function class $\mathcal{F}$. From the single version of the fundamental theorem of concept learning we know that

$$L(f) \leq \hat{L}(f) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{V(\mathcal{F})}{n}}\right),$$

where we omit the term from McDiarmid inequality for simplicity.

Proof. The proof is based on the mismatched minimization lemma (Lemma 5.1) , the Rademacher averages (Theorem 6.1.) and the Sauer-Shelah lemma (Lemma 7.2) introduced in class. See lecture notes for the detailed proof. □

2.2 GENERALIZATION BOUND

Lemma 2.2. Consider a neural network with ReLU activations, L layers and W parameters. Let $\mathcal{F}$ denote all real-valued functions computed by the network. Let h_i denote the number of units at the ith layer, U denote the total number of computation units and W_i denote the total number of parameters in all the layers up to layer i (i.e., in layers 1,2,...,i). Then, the growth function of $\mathcal{F}$ is bounded by

$$\mathbb{S}_{V(\mathcal{F})}(\mathsf{sgn}(\mathcal{F})) \leq 2^L \left(\frac{2eV(\mathcal{F}) \sum_{i=1}^{L} ih_i}{\sum W_i} \right)^{\sum W_i}.$$

Proof. The proof is based on a bound on the growth function of a polynomially parameterized function class due to Goldberg & Jerrum (1995). See [Bartlett et al. (2017b)] for the detailed proof of this lemma. □

Lemma 2.3. Suppose that $2^v \leq (vr/w)^w$ for some $r \geq 16$ and $v \geq w \geq 0$. Then, $v \leq w \log_2(2r \log_2 r)$.

Proof. It can be shown that if $v > w \log_2(2r \log_2 r)$ for any $v \geq w \geq 0$ and some $r \geq 16$, then $2^v > (vr/w)^w$. Thus, if $2^v \leq (vr/w)^w$ under the specified conditions, $v \leq w \log_2(2r \log_2 r)$. □

Theorem 2.4. Consider a neural network with ReLU activations, L layers and W parameters. Let $\mathcal{F}$ denote all real-valued functions computed by the network. Then, there exists a constant C such that

$$\mathrm{V}(\mathcal{F}) \leq C \cdot WL \log W.$$

Proof. By the definition of growth function of VC-dimension,

$$2^{V(\mathcal{F})} = \mathbb{S}_{V(\mathcal{F})}(\mathsf{sgn}(\mathcal{F})).$$

According to Lemma 2.2. and the fact that $L \leq \sum W_i$ and $\sum h_i = U$, yield

$$\begin{aligned} 2^{V(\mathcal{F})} &= \mathbb{S}_{V(\mathcal{F})}(\mathsf{sgn}(\mathcal{F})) \\ &\leq 2^L \left(\frac{2eV(\mathcal{F}) \sum_{i=1}^{L} ih_i}{\sum W_i} \right)^{\sum W_i} && \text{(Lemma 2.2.)} \\ &\leq \left(\frac{4eV(\mathcal{F})L \sum h_i}{\sum W_i} \right)^{\sum W_i} && (L \leq \textstyle\sum W_i) \\ &\leq \left(\frac{V(\mathcal{F}) \cdot 4eLU}{\sum W_i} \right)^{\sum W_i} && (\textstyle\sum h_i = U). \end{aligned}$$

Since $LU \geq 2$ is usually the case for neural networks, this implies $4eLU \geq 16$. Hence Lemma 2.3. gives

$$\begin{aligned} V(\mathcal{F}) &\leq \sum W_i \log_2(8eLU \log_2 4eLU) \\ &= \bar{L}W \log_2(8eLU \log_2 4eLU) \\ &= \mathcal{O}(\bar{L}W \log U) \qquad (1) \\ &= \mathcal{O}(LW \log W), \end{aligned}$$

where $\bar{L} := \frac{1}{W} \sum_{i=1}^{L} W_i$ and is called *effective depth* or *average depth*. □

Remark. By the definition of $\bar{L}$, it is always between 1 and L and it can capture how the parameters are distributed in the network. For example, consider two networks both with three hidden layers. The numbers of hidden units of the first network are 1, 1 and 100 respectively, then $\bar{L}_1 = \frac{1+(1+1)+(1+1+100)}{102} \approx 1.03$; The numbers of hidden units of the second network are 100, 1 and 1 respectively, then $\bar{L}_2 = \frac{100+(100+1)+(100+1+1)}{102} \approx 2.97$. This shows that $\bar{L}$ is close to 1 if parameters are concentrated near the output. Based on this observation and Eq. (1), $V(\mathcal{F}) \leq \mathcal{O}(\bar{L}W \log U)$, it implies that in a neural network nodes closer to the input have a larger effect in increasing the VC-dimension.

Theorem 2.5. Consider a neural network with ReLU activations, L layers and W parameters. Combine Theorem 2.1. and 2.4., yielding

$$L(f) \leq \hat{L}(f) + \widetilde{\mathcal{O}}\left(\sqrt{\frac{WL}{n}}\right). \tag{2}$$

Thus, we have obtained a generalization bound for neural networks based on VC-dimension. However, it depends on the number parameters and the number of layers in the network. This makes it not that useful since today many deep neural networks are over-parameterized, that is, $W \gg n$, which means the right hand side of Eq. (2) would be greater than 1. In other word, VC-dimension of neural networks can give us some insights about the approximation ability of them, but we probably need a bound that does not depend on the number of parameters. Therefore, we introduce two bounds that do not depend on the number of parameters in next two sections.

3 Generalization bound derived from PAC-Bayesian framework

In this section, we introduce a generalization bound that depends on the product of the spectral norm of the layers and the Frobenius norm of the weights of neural networks in a PAC-Bayesian framework from Neyshabur et al. (2017).

3.1 Preliminaries

We first introduce the definitions and notations of three types of norms that will be used later: (1) Spectral norm is a matrix norm induced by the L_2 norm. For matrix A and vector x, the spectral norm is $\|A\|_2$ and $\|A\|_2 = \max_{|x|_2=1} |Ax|_2$. This norm is heavily used in numerical analysis; it can measure how large the matrix can "stretch" a vector and is related to the condition number of a matrix. (2) Frobenius norm of matrix A is defined as $\|A\|_F = (\sum_i \sum_j a_{ij}^2)^{1/2}$. (3) Vector p-norm is denoted by $|\cdot|_p$.

The following is the notations that will be used to describe a neural network for both section 3 and section 4. Let $\mathcal{X}_{B,n} = \{x \in \mathbb{R}^n | \sum_{i=1}^n x_i^2 \leq B^2\}$ and $f_w(x) : \mathcal{X}_{B,n} \mapsto \mathbb{R}^k$ be the function computed by a L layer neural network for a k-class classification task with parameters $w = \text{vec}(\{W_i\}_{i=1}^L)$, $f_w(x) = W_L\phi(W_{L-1}\phi(...\phi(W_1 x)))$, where W_i is the weight matrix of layer i and ϕ is the ReLU activation function. Let $f_w^i(x)$ denote the output of layer i before activation and h be an upper bound on the number of output units in each layer. Note that W_i here differs from the W_i defined in section 2.

Definition 3.1. For any distribution $\mathcal{D}$ and margin $\gamma > 0$, define the expected margin loss as follows:

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

where vec$[i]$ returns the ith element in the vector. Let $\hat{L}_\gamma(f_w)$ be the empirical estimate of the expected margin loss. Then, $L_0(f_w)$ is the expected risk and $\hat{L}_0(f_w)$ is the empirical risk.

Definition 3.2. Sharpness is a generalization measure that corresponds to robustness to adversarial perturbations on the parameter space of neural networks. One way to define it is that

$$\zeta(w) = \max_v \hat{L}_0(f_{w+v}) - \hat{L}_0(f_w).$$

3.2 PAC-Bayesian framework

PAC-Bayesian framework was proposed to prove generalization bounds without the use of VC-dimension by providing an informative prior distribution on the parameters. Due to the work done by McAllester (2003), we have the following PAC-Bayesian theorem.

Theorem 3.1. Let P be a prior distribution on (the parameters of) the function class $\mathcal{F}$. Let $f_Q \in \mathcal{F}$ be a predictor that is parameterized by Q, where Q is a random variable and its distribution is also on $\mathcal{F}$. The two-sided PAC-Bayesian theorem states that for any Q and a fixed P, with probability at least $1-\delta$ that

$$D_{KL}\left(\mathbb{E}_Q[\hat{L}_0(f_Q)]\,||\,\mathbb{E}_Q[L_0(f_Q)]\right) \leq \frac{D_{KL}(Q||P)+\ln\frac{2n}{\delta}}{n-1}, \tag{3}$$

where n denotes the sample size. This theorem implies that if $D_{KL}(Q||P)$ is small, then $\mathbb{E}_Q[\hat{L}_0(f_Q)]$ is near $\mathbb{E}_Q[L_0(f_Q)]$. A one-sided version states that for any Q and a fixed P, with probability at least $1-\delta$ that

$$\mathbb{E}_Q[L_0(f_Q)] \leq \sup\left\{\epsilon : D_{KL}(\mathbb{E}_Q[\hat{L}_0(f_Q)]\,||\,\epsilon) \leq \frac{D_{KL}(Q||P)+\ln\frac{n}{\delta}}{n-1}\right\}. \tag{4}$$

Proof. See [McAllester (2003)] for the proof. □

Corollary 3.1.1. With the same setting in Theorem 3.1., for any Q and a fixed P, with probability at least $1-\delta$ that

$$\mathbb{E}_Q[L_0(f_Q)] \leq \mathbb{E}_Q[\hat{L}_0(f_Q)] + \sqrt{\frac{2\mathbb{E}_Q[\hat{L}_0(f_Q)](D_{KL}(Q||P)+\ln\frac{n}{\delta})}{n-1}} + \frac{2(D_{KL}(Q||P)+\ln\frac{n}{\delta})}{n-1}. \tag{5}$$

Proof. For $q > p$ we have $D_{KL}(p||q) \geq (q-p)^2/(2q)$, which implies if $D_{KL}(p||q) \leq x$ then $q \leq p + \sqrt{2px} + 2x$. Based on this, Eq. (5) is directly from Eq. (4). □

Since the Q in Theorem 3.1. and Corollary 3.1.1. is a random variable, we can relate the PAC-Bayesian framework to the sharpness of neural networks by replacing it with the addition of a fixed w and a random variable u. Then we have the following theorem mentioned in [Neyshabur et al. (2017)].

Theorem 3.2. Let f_w be any predictor learned from the training data and parametrized by fixed w. Consider the distribution over predictors of the form f_{w+u}, where u is a random variable whose distribution may also depend on the training data. P is the prior distribution on the parameters that is independent of the training data. From the PAC-Bayesian Theorem 3.1., with probability at least $1-\delta$, we have

$$\mathbb{E}_u[L_0(f_{w+u})] \leq \mathbb{E}_u[\hat{L}_0(f_{w+u})] + 2\sqrt{\frac{2(D_{KL}(w+u||P)+\ln\frac{2n}{\delta})}{n-1}}.$$

Proof. This is directly from Corollary 3.1.1. and is a weaker form of it. □

To turn the bound for the perturbed predictor f_{w+u} in Theorem 3.2., to a bound that works for the unperturbed predictor f_w, we need the following lemma.

Lemma 3.3. Let $f_w(x) : \mathcal{X} \mapsto \mathbb{R}^k$ be any predictor with fixed parameters w, and P be any distribution on the parameters that is independent of the training data. Then, for any $\gamma, \delta > 0$, with probability at least $1-\delta$ over the training set of size n, for any w, and any random perturbation u that satisfies $\mathbb{P}_u[\max_{x\in\mathcal{X}} |f_{w+u}(x) - f_w(x)|_\infty < \frac{\gamma}{4}] \geq \frac{1}{2}$, we have

$$L_0(f_w) \leq \hat{L}_\gamma(f_w) + 4\sqrt{\frac{D_{KL}(w+u||P)+\ln\frac{6n}{\delta}}{n-1}}.$$

Proof. See [Neyshabur et al. (2017)] for the proof. □

3.3 Generalization bound

Neyshabur et al. (2017) proved a generalization bound for neural networks in a PAC-Bayesian framework that does not depend on the number of parameters in the network. The key of their proof is to apply Lemma 3.3. in the PAC-Bayesian framework. They first bound the KL divergence in the lemma by pre-defining the distribution of u and P. Then, they choose proper parameters for the two distributions u and P using the perturbation bound Lemma 3.4. such that it can satisfy the condition that $\mathbb{P}_u[\max_{x \in \mathcal{X}} |f_{w+u}(x) - f_w(x)|_\infty < \frac{\gamma}{4}] \geq \frac{1}{2}$ and therefore they can apply Lemma 3.3. to obtain a generalization bound for neural networks.

Thus, next we present a perturbation bound that bounds the change in the output of the network when the weights are perturbed, thereby bounds the sharpness of the network, in terms of the spectral norm of the layers. Then we use the perturbation bound Lemma 3.4. as well as the Lemma 3.3. to derive a generalization bound for neural networks.

Lemma 3.4. For any $B, L > 0$, let $f_w : \mathcal{X}_{B,n} \mapsto \mathbb{R}^k$ be a L-layer neural network with ReLU activations. Then for any w, and $x \in \mathcal{X}_{B,n}$, and any perturbation $u = \mathrm{vec}(\{U_i\})_{i=1}^{L}$ such that $\|U_i\|_2 \leq \frac{1}{L}\|W_i\|_2$, the change in the output of the network can be bounded as follows:

$$|f_{w+u}(x) - f_w(x)|_2 \leq eB \left(\prod_{i=1}^{L} \|W_i\|_2 \sum_{i=1}^{L} \frac{\|U_i\|_2}{\|W_i\|_2} \right).$$

Proof. Let's define $\Delta_i = |f^i_{w+u}(x) - f^i_w(x)|_2$ for i from 0 to L, where f^i_w denotes the output of layer i before activation given weights w. Thus Δ_0 is the perturbation for the input x, which is zero since perturbation on weights will not influence inputs, and Δ_L is the perturbation of the output of the neural networks; in other words, $\Delta_L = |f^L_{w+u}(x) - f^L_w(x)|_2 = |f_{w+u}(x) - f_w(x)|_2$. Then, by induction on Δ_i, we obtain the upper bound of Δ_L. □

With Lemma 3.3. and Lemma 3.4. we can now present the following generalization bound for neural networks based on PAC-Bayesian theorem.

Theorem 3.5. For any $B, L, h > 0$, let $f_w : \mathcal{X}_{B,n} \mapsto \mathbb{R}^k$ be a L-layer neural network with ReLU activations. Then, for any $\delta, \gamma > 0$, with probability $\geq 1 - \delta$ over a training set of size n, for any w, we have

$$L_0(f_w) \leq \hat{L}_\gamma(f_w) + \mathcal{O}\left(\frac{B^2 L^2 h \ln(Lh) \prod_{i=1}^{L} \|W_i\|_2^2 \sum_{i=1}^{L} \frac{\|W_i\|_F^2}{\|W_i\|_2^2} + \ln \frac{Ln}{\delta}}{\gamma^2 n} \right).$$

Proof. Now, we first find a bound for the KL divergence in Lemma 3.3. by choosing the distribution of the prior P to be $\mathcal{N}(0, \sigma^2 I)$ and $u \sim \mathcal{N}(0, \sigma^2 I)$ as well, where the value of σ will be determined later. Based on the fact that the KL-divergence between two Gaussian distributions with mean μ_1, μ_2 and same variance σ^2 is $\frac{(\mu_1 - \mu_2)^2}{2\sigma^2}$ and w is fixed, we get the following bound: $D_{KL}(w + u||P) \leq \frac{|w|^2}{2\sigma^2}$. Thus, if we can find a proper value for σ, then we can plugin it in the Lemma 3.3. and get a generalization bound for neural networks.

It is obvious that to apply Lemma 3.3., the σ should be chosen in a way that the condition $\mathbb{P}_u[\max_{x \in \mathcal{X}} |f_{w+u}(x) - f_w(x)|_\infty < \frac{\gamma}{4}] \geq \frac{1}{2}$ will be satisfied. Hence, next we choose the value of σ based on Lemma 3.4.

Let's assume the spectral norm is equal across layers, i.e. for any layer i, $\|W_i\|_2 = \beta$. This can be assumed without loss of generality since using ReLU as the activation function, the output of a network will not change with weights to be normalized or not and therefore we can assume w.l.o.g. the norms are all equal to β. However, β here is related to the learned w of the predictor and σ is the parameter of the prior distributions. Since the prior cannot depend on the learned predictor, we will set σ based on an approximation $\widetilde{\beta}$, which satisfies that $|\beta - \widetilde{\beta}| \leq \frac{1}{L}\beta$, and hence $\frac{1}{e}\beta^{L-1} \leq \widetilde{\beta}^{L-1} \leq e\beta^{L-1}$.

Then, according to Lemma 3.4.,

$$\max_{x\in\mathcal{X}_{B,n}} |f_{w+u}(x) - f_w(x)|_2 \le eB\beta^L \sum_i \frac{\|U_i\|_2}{\beta}. \tag{6}$$

To relate this bound to σ, note that $u \sim \mathcal{N}(0, \sigma^2 I)$ and we get the following bound on $\|U_i\|_2$ due to Tropp (2012):

$$\mathbb{P}_{U_i\sim\mathcal{N}(0,\sigma^2 I)}[\|U_i\|_2 > t] \le 2he^{-t^2/2h\sigma^2}.$$

Then, we take the union bound over all layers to get an upper bound of $\sum_i \|U_i\|_2$. To satisfy the condition that $\mathbb{P}_u[\max_{x\in\mathcal{X}} |f_{w+u}(x) - f_w(x)|_\infty < \frac{\gamma}{4}] \ge \frac{1}{2}$, we set the probability of the union bound to be at least $\frac{1}{2}$, then we get the following: $\sum_i \|U_i\|_2 \le \sigma\sqrt{2h\ln(4Lh)}$ w.p. $\ge \frac{1}{2}$.

Thus, based on this bound and Eq. (6), we have that with probability at least $\frac{1}{2}$,

$$\begin{aligned}
\max_{x\in\mathcal{X}_{B,n}} |f_{w+u}(x) - f_w(x)|_2 &\le eB\beta^{L-1} \sum_i \|U_i\|_2 \\
&\le eB\beta^{L-1}\sigma\sqrt{2h\ln(4Lh)} \\
&\le e^2 LB\widetilde{\beta}^{L-1}\sigma\sqrt{2h\ln(4Lh)} \\
&\le \frac{\gamma}{4}.
\end{aligned}$$

Now, we can choose $\sigma = \frac{\gamma}{42LB\widetilde{\beta}^{L-1}\sqrt{h\ln(4hL)}}$ and we get the upper bound of the KL divergence

$$\begin{aligned}
D_{KL}(w+u||P) \le \frac{|w|^2}{2\sigma^2} &= \frac{42^2 L^2 B^2 \widetilde{\beta}^{2L-2} h\ln(4hL)}{2\gamma^2} \sum_{i=1}^{L} \|W_i\|_F^2 \\
&\le \mathcal{O}\left(B^2 L^2 h \ln(Lh) \frac{\prod_{i=1}^L \|W_i\|_2^2}{\gamma^2} \sum_{i=1}^{L} \frac{\|W_i\|_F^2}{\|W_i\|_2^2} \right).
\end{aligned}$$

Finally, we plugin this in Lemma 3.3. and complete the proof. □

4 Generalization bound derived from covering number

In this section, we briefly introduce a generalization bound that is even tighter than the bound in section 3 due to Bartlett et al. (2017a). This bound is proved by a complex covering number argument, but it is strictly better than the bound from the PAC-Bayesian framework and it improves over existing results.

We state the following theorem with the same setting and notations used in section 3 except that we denote $\|A\|_{2,1}$ as the sum of the Euclidean norms of the columns of the matrix, that is, $\|A\|_{2,1} = \sum_j \left(\sum_i a_{i,j}^2\right)^{1/2}$.

Theorem 4.1. For any $B, L, h > 0$, let $f_w : \mathcal{X}_{B,n} \mapsto \mathbb{R}^k$ be a L-layer neural network with ReLU activations. Then, for any $\delta, \gamma > 0$, with probability $\ge 1-\delta$ over a training set of size n, for any w, we have

$$L_0(f_w) \le \hat{L}_\gamma(f_w) + \widetilde{\mathcal{O}}\left(\frac{B^2\ln(h)\prod_{i=1}^L \|W_i\|_2^2 \left(\sum_{i=1}^L (\frac{\|W_i\|_{2,1}}{\|W_i\|_2})^{2/3}\right)^{3/2}}{\gamma^n} + \sqrt{\frac{\ln(1/\delta)}{n}} \right).$$

5 Discussion and Conclusion

Now, we have presented three generalization bounds for neural networks. The first bound is derived from VC-dimension and it depends on the number of parameters in the network, which makes the bound not that useful since many deep neural networks are over-parameterized. However, one intermediate result from the proof of the VC-dimension of neural networks shows that nodes in the network contribute unevenly in increasing the VC-dimension of the network and nodes closer to the input would have a larger effect. Since the bound for the VC-dimension is nearly-tight, this might give us a insight that increasing the number of nodes of the early layers would increase the approximation ability of neural networks more significantly.

To obtain more meaningful bounds that do not depend on the number of parameters in the network, we then introduced two spectrally-normalized margin bounds. We observe that the term $\sum_{i=1}^{L} \frac{\|W_i\|_F^2}{\|W_i\|_2^2}$ in the second bound and the term $(\sum_{i=1}^{L} (\frac{\|W_i\|_{2,1}}{\|W_i\|_2})^{2/3})^{3/2}$ in the third bound may suggest that regularization on L_2 norm of weights might have limited influence on generalization since these two terms are relatively small compared with the product of the spectral norms. In addition, the factor $\mathcal{O}(\frac{1}{\gamma^2} B^2 \prod_{i=1}^{L} \|W_i\|_2^2)$ appears in both bounds. It may suggest that (1) data normalization could help generalization; (2) the product of the spectral norms of weight matrices could play an important role for generalization. This might indicate that although usually one would do L_1 or L_2 regularization when training neural networks, what we really need might be a regularization on the spectral norm of the weight matrices. Thus, it might be worth trying to develop some regularization technique based on the spectral norm to see if there will be improvements on the generalization ability of networks.

In conclusion, these bounds give us more insights about the neural networks and they help us to understand and interpret how neural networks work. Although these bounds cannot tell us everything about neural networks, we now at least know a little aspects about them instead of viewing them completely as black boxes. Maybe in future we could have a even tighter bound and then we might be able to better understand why neural networks are so powerful.

References

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6240–6249, 2017a.

Peter L Bartlett, Nick Harvey, Chris Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *arXiv preprint arXiv:1703.02930*, 2017b.

Paul W Goldberg and Mark R Jerrum. Bounding the vapnik-chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18(2-3):131–148, 1995.

Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4 (2):251–257, 1991.

David McAllester. *Simplified pac-bayesian margin bounds*, pp. 203–215. Lecture notes in computer science, 2003.

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

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.