PAC-Bayes ReLu NN Lecture Notes

Aaron Geelon So

April 5, 2018

Last time, we saw the PAC-Bayes bound, which states:

Theorem 1 (PAC-Bayes)**.** *Let P be a prior distribution over $\mathcal{H}$, L be a loss bounded between 0 and 1, and $\hat{L}$ be the empirical loss using m samples. Then, with probability at least $1-\delta$, for all posterior distributions Q,*

$$\mathbb{E}_Q\left[L(h)\right] \leq \mathbb{E}_Q\left[\hat{L}(h)\right] + \sqrt{\frac{2\left(\mathrm{KL}(Q||P) + \ln\frac{2m}{\delta}\right)}{m-1}}. \tag{1}$$

Today, we'll see an application of this theorem to give a risk bound for a ReLu neural network h performing a k-classification task of some bounded set $B \subset \mathbb{R}^n$ into $\mathbb{R}^k$. Here, as usual, the point $x \in B$ is classified into:

$$j^* = \arg\max_{j\in[k]} h(x)_j,$$

and the *margin* by which j^* beats the other classes is just:

$$\text{margin} = \max_i \left|h(x)_i - h(x)_{j^*}\right|.$$

Here's the high-level idea of the rest of the lecture. Suppose we've produced a classifier h^* through training (using m samples), and now we want to obtain an upper bound on $L(h^*)$, saying something like:

$$L(h^*) \leq \hat{L}(h^*) + \varepsilon.$$

We'd like to just apply PAC-Bayes, but we can't do that without getting a vacuous bound. That's because the posterior distribution here would be δ_{h^*}, with all the probability mass at a single point. And in contrast, the prior is probably much more dispersed, so $\mathrm{KL}(\delta_{h^*}||P) \gg 1$. In fact, if P is a continuous probability density, then the KL-divergence is actually infinite. (See Figure 1).

But if we know that classifiers h around h^* don't behave all that differently from h^*, we can afford to diffuse δ_{h^*} to some other posterior Q, whose 'mean' behavior is still h^*. Though we may pay some in the $\mathbb{E}_Q\left[\hat{L}(h)\right]$ term, we might gain a lot in the remainder term.

Indeed, this interpretation might even be preferable—after training on just m samples, we shouldn't be absolutely confident that h^* is the correct classifier. It's just the *mean* classifier.

If we're going to move our analysis to a 'weaker' posterior distribution Q (compared to δ_{h^*}), then we also need to compensate by using a loss L' that's 'stronger' than L. That way, we can provide an upper bound:

$$L(h^*) \lesssim \mathbb{E}_Q\left[L'(h)\right].$$

(Spoiler: L' will be a margin loss). Following that, we can pass to empirical loss estimates over Q using PAC-Bayes. And finally, if it's true that most classifiers distributed according to Q are close to h^*, we can bound the empirical loss estimates over Q by empirical loss estimates over h^*.

In the remainder of the lecture, we'll first prove a lemma that does precisely this. Then, we'll prove some properties about ReLu neural networks, which we'll use to specialize our analysis to this family of neural networks. This lecture follows [Neyshabur 2018].

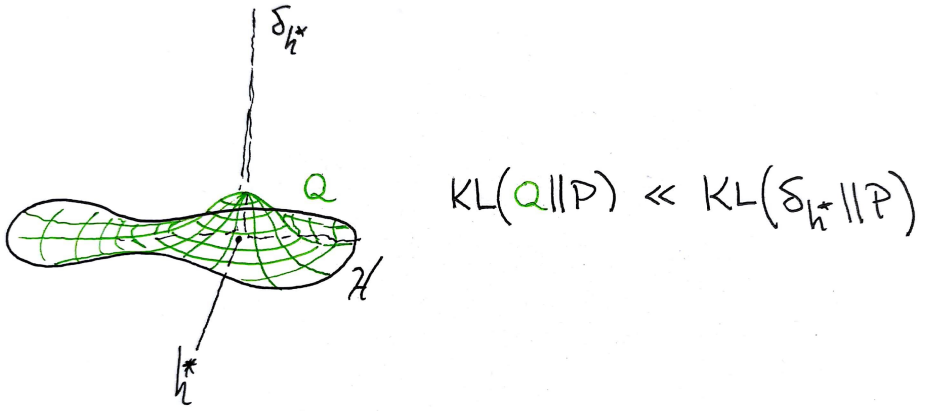

Figure 1: Two posterior distributions on $\mathcal{H}$, δ_{h^*} and Q. If the prior P is the uniform distribution on $\mathcal{H}$, then $\mathrm{KL}(\delta_{h^*}||P)$ might even be infinite.

Passing to Margins

Above, we defined the *margin* as the amount by which the chosen class beats the other classes. We can define the *margin loss* L_γ so that the correct class must not only beat all other classes, but it must have them beat by some margin γ:

$$L_\gamma(h) = \Pr_{(x,j)\sim\mathcal{D}}\left[f(x)_j \leq \gamma + \max_{i\neq j} f(x)_i\right].$$

Some useful properties about the margin loss:

- $L_0(h)$ is equivalent to the 0-1 loss.
- if $\gamma \leq \eta$, then $L_\gamma(h) \leq L_\eta(h)$.
- if for all $x \in \mathcal{X}$, two hypotheses h and h' give similar predictions, where $|h(x) - h'(x)|_\infty \leq \frac{\gamma}{2}$, then

  $$L_\eta(h) \leq L_{\eta+\gamma}(h') \quad \text{and} \quad \hat{L}_\eta(h) \leq \hat{L}_{\eta+\gamma}(h').$$

  Note that the conditions are symmetric in h and h', so the same inequalities holds when h and h' are reversed.

These properties let us easily prove the following:

Proposition 2. *Let P be a prior distribution over a hypothesis class $\mathcal{H}$. Suppose we obtain a classifier $h^* \in \mathcal{H}$ after training with m samples. If Q be a (posterior) distribution over $\mathcal{H}$ such that for any $h \sim Q$, for all $x \in \mathcal{X}$*

$$|h(x) - h^*(x)|_\infty \leq \frac{\gamma}{4}, \tag{2}$$

then, the 0-1 loss of h^* has an upper bound, with probability $1-\delta$:

$$L_0(h^*) \leq \hat{L}_\gamma(h^*) + \sqrt{\frac{2(\mathrm{KL}(Q||P) + \ln\frac{2m}{\delta})}{m-1}}. \tag{3}$$

Proof. This simply follows from the fact that for all $h \sim Q$,

$$L_0(h^*) \leq L_{\gamma/2}(h),$$

so it also holds in expectation, $L_0(h^*) \leq \mathbb{E}_Q\left[L_{\gamma/2}(h)\right]$. Then, we apply PAC-Bayes:

$$L_0(h^*) \leq \mathbb{E}_Q\left[\hat{L}_{\gamma/2}(h)\right] + \sqrt{\frac{2(\mathrm{KL}(Q||P) + \ln\frac{2m}{\delta})}{m-1}}.$$

And because for all h sampled from Q, we also have $\hat{L}_{\gamma/2}(h) \leq \hat{L}_{\gamma}(h^*)$, this inequality also holds in expectation, and thus we obtain the bound in the proposition statement. □

Now, perhaps it is unreasonable to expect that we can easily find a distribution Q where any h sampled from Q will satisfy Equation 2. But maybe we know that we can obtain some distribution $\widetilde{Q}$ where a constant fraction of them does. That is,

$$Z := \Pr_{h\sim\widetilde{Q}}\left[\forall x \in \mathcal{X}, |h(x) - h^*(x)| \leq \frac{\gamma}{4}\right] \geq c > 0, \tag{4}$$

for some constant c. Then, we can decompose $\widetilde{Q}$ as a sum of two distributions: one that has support over the classifiers satisfying Equation 2, say $Q_{\text{small perturbations}}$, and one that has support over those that don't, $Q_{\text{large perturbations}}$. That is,

$$\widetilde{Q} = ZQ_{\text{small perturbations}} + (1-Z)Q_{\text{large perturbations}}.$$

From this, we can immediately obtain a bound on $L_0(h^*)$ using Proposition 2, where $Q = Q_{\text{small perturbations}}$. But as we can't compute $\mathrm{KL}(Q_{\text{small perturbations}}||P)$ directly, we need to bound that through $\mathrm{KL}(\widetilde{Q}||P)$. We achieve this via the identity:

$$\mathrm{KL}(\widetilde{Q}||P) = Z\mathrm{KL}(Q_{\text{small perturbations}}||P) + (1-Z)\mathrm{KL}(Q_{\text{large perturbations}}||P) - H(Z), \tag{5}$$

where it follows that

$$\mathrm{KL}(Q_{\text{small perturbations}}||P) \leq \frac{1}{Z}\left(\mathrm{KL}(\widetilde{Q}||P) + 1\right). \tag{6}$$

Plugging this bound into the KL-term in Proposition 2 gives us:

Proposition 3. *Let P be a prior over $\mathcal{H}$. Let $h^* : \mathcal{X} \to \mathbb{R}^k$ be a classifier obtained after training with m samples. If Q is a posterior distribution over $\mathcal{H}$ such that Equation 4 is satisfied for some $c > 0$, then, the 0-1 loss of h^* is upper bounded with probability $1-\delta$,*

$$L_0(h^*) \leq \hat{L}_{\gamma}(h^*) + \frac{2}{c}\sqrt{\frac{\mathrm{KL}(Q||P) + c^2\ln\frac{2e^{1/c^2}m}{\delta}}{2(m-1)}}. \tag{7}$$

□

We'll now use this framework to prove bounds for ReLu neural networks.

ReLu Neural Networks

We start by defining this family of neural networks, then proving some useful properties about them.

Definition 4. *Let $\mathcal{F}$ be family of d-layered ReLu neural networks $f_{\mathbf{w}} : \mathbb{R}^n \to \mathbb{R}^k$ parametrized by its weights $\mathbf{w} = \text{vec}\left(\{W_i\}_{i=1}^d\right)$. That is, let $\phi : \mathbb{R}^s \to \mathbb{R}^s$ be the (vectorized) activation function:*

$$\phi(x)_i = \begin{cases} x_i & x_i \geq 0 \\ 0 & \text{o.w.} \end{cases}$$

Then, $f_{\mathbf{w}}$ is decomposed into:

$$f_{\mathbf{w}}(\mathbf{x}) = \phi(W_d \phi(W_{d-1} \phi \cdots \mathbf{x})).$$

Furthermore, let h be an upper bound on the number of output unit of any layer, so that we also have $W_i : \mathbb{R}^s \to \mathbb{R}^t$, where $s, t \leq h$.

Here, we'll show that we can restrict our analysis to neural networks $f_{\mathbf{v}} : B_n(0,1) \to \mathbb{R}^k$, where $B_n(0,1)$ is the unit n-ball in $\mathbb{R}^n$, and each V_i in $\mathbf{v}$ is normalized: $\|V_i\|_2 = 1$. Briefly, this is because ReLu neural networks are homogeneous, and margin losses scale inversely with the 'magnitude' the weights and the radius of the domain. More formally,

Lemma 5. *ReLu neural networks are homogeneous. That is, let $f_{\mathbf{w}} \in \mathcal{F}$. Then, $f_{\mathbf{w}}(\lambda \mathbf{x}) = \lambda f_{\mathbf{w}}(\mathbf{x})$ if $\lambda \geq 0$.*

Proof. Multiplication by W_i is linear and ϕ is homogeneous. Thus, any sequence of compositions of W_i's and ϕ's is homogeneous. □

Corollary 6. *Every ReLu neural network $f_{\mathbf{w}}$ may be written as a product of a nonnegative scalar α with a neural network $f_{\mathbf{v}}$, where each V_i in $\mathbf{v}$ has $\|V_i\|_2 = 1$. Thus,*

$$f_{\mathbf{w}} = \alpha f_{\mathbf{v}}.$$

Proof by induction. For each layer, we can rewrite the computation as:

$$\phi W_i \mathbf{x} = \phi \frac{W_i}{\|W_i\|_2} \left(\|W_i\|_2 \mathbf{x}\right) = \|W_i\|_2 \cdot \phi V_i \mathbf{x},$$

where $V_i = W_i / \|W_i\|_2$. We can pull the $\|W_i\|_2$ through the rest of the layers by Lemma 5. Therefore, without loss of generality, we may take any ReLu neural network to be of the form $\alpha f_{\mathbf{v}}$, where $\mathbf{v}$ is a collection of L^2-normalized weights for each of the d layers. □

This also shows that we can (multiplicatively) redistribute the weight α among the different layers without changing the output of the classifier. Now, we show how scaling a classifier affects margin losses. While the following isn't strictly needed, it gives a relationship that can help us check units later.

Corollary 7. *If $f_{\mathbf{w}} = \alpha f_{\mathbf{v}}$, then $L_\gamma(f_{\mathbf{w}}) = L_{\gamma/\alpha}(f_{\mathbf{v}})$.*

Proof. Expanding out the margin loss, we have:

$$\begin{aligned} L_\gamma(f_{\mathbf{w}}) &= \Pr_{(x,j)\sim\mathcal{D}}\left[f_{\mathbf{w}}(x)_j \leq \gamma + \max_{i \neq j} f_{\mathbf{w}}(x)_i\right] \\ &= \Pr_{(x,j)\sim\mathcal{D}}\left[f_{\mathbf{v}}(x)_j \leq \frac{\gamma}{\alpha} + \max_{i \neq j} f_{\mathbf{v}}(x)_i\right] = L_{\gamma/\alpha}(f_{\mathbf{v}}). \end{aligned}$$

Thus, we may restrict our analysis to normalized ReLu neural networks, and pass the scalar α into the margin loss parameter γ. □

By similar argument, it is not surprising we can also assume that the domain $\mathcal{X} \subset B_n(0,1)$ is contained in the unit n-ball. Briefly, if $\sup_{x\in\mathcal{X}} \|x\|_2 = r$, then just let $\mathcal{X}' = \mathcal{X}/r$; for every x, we have a corresponding $x' = x/r$, and its clear from above that $\alpha f_{\mathbf{v}}(x) = r\alpha f_{\mathbf{v}}(x')$. So in fact, the prior analysis for α also provides the analysis for the magnitude term r. Formally, this is summarized as follows:

Corollary 8. *Let $\mathcal{X} \subset \mathbb{R}^n$ be bounded, with $\|\mathbf{x}\|_2 \leq r$ for all $\mathbf{x} \in \mathcal{X}$. Let $\mathcal{X}' = \mathcal{X}/r$ just be its normalization. Let $f_{\mathbf{w}} = \alpha f_{\mathbf{v}}$ be a ReLu neural network (where $f_{\mathbf{v}}$ is normalized). Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, and let $\mathcal{D}'$ be the natural distribution over $\mathcal{X}' \times \mathcal{Y}$ where the likelihood of (x, y) being sampled from $\mathcal{D}$ is equal to the likelihood that $(x/r, y)$ is sampled from $\mathcal{D}'$. Then,*

$$L_{\gamma,\mathcal{X}}(f_{\mathbf{w}}) = L_{\gamma/r\alpha,\mathcal{X}'}(f_{\mathbf{v}}).$$

With this in mind, let's see how similarly classifiers around $f_{\mathbf{v}}$ behave. First, a quick technical lemma that we state without proof:

Lemma 9. *Let $\phi : \mathbb{R}^s \to \mathbb{R}^s$. Then, ϕ is a contraction. That is, if $x, y \in \mathbb{R}^s$, then:*

$$\|\phi(x) - \phi(y)\|_2 \leq \|x - y\|_2 .$$

Proposition 10. *Let $\mathbf{u} = \mathrm{vec}(\{U_i\}_{i=1}^d)$ be a small perturbation, where each $\|U_i\|_2 \leq \frac{1}{d}$. Let $f_{\mathbf{v}}$ be a (normalized) ReLu neural network. Then for all $x \in B_n(0,1)$,*

$$\|f_{\mathbf{v}+\mathbf{u}}(x) - f_{\mathbf{v}}(x)\|_2 \leq \left(1 + \frac{1}{d}\right)^d \sum_{i=1}^d \|U_i\|_2 \leq e \sum_{i=1}^d \|U_i\|_2 .$$

Note that the second inequality follows immediately from the fact that the Taylor expansion for $e^x \geq 1 + x$ implies $\left(1 + \frac{1}{x}\right)^x \leq e$. So, all we need to do is prove the first inequality.

Proof by induction. We may assume that $\|x\|_2 = 1$, for if the bound holds for all such x's, then by homogeneity of $f_{\mathbf{v}}$, it holds for any $x \in B_n(0,1)$.

Denote by $f^i_{\mathbf{w}}(x)$ the output of the ith layer of $f_{\mathbf{w}}$ before applying the activation function. Let us bound how far $f^i_{\mathbf{v}+\mathbf{u}}(x)$ and $f^i_{\mathbf{v}}(x)$ have diverged from each other:

$$\Delta_i = f^i_{\mathbf{v}+\mathbf{u}}(x) - f^i_{\mathbf{v}}(x).$$

And let Δ'_i denote the similar quantity after activation:

$$\Delta'_i = \phi\left(f^i_{\mathbf{v}+\mathbf{u}}(x)\right) - \phi\left(f^i_{\mathbf{v}}(x)\right).$$

This lets us inductively define Δ_{i+1}, most easily seen by the following figure:

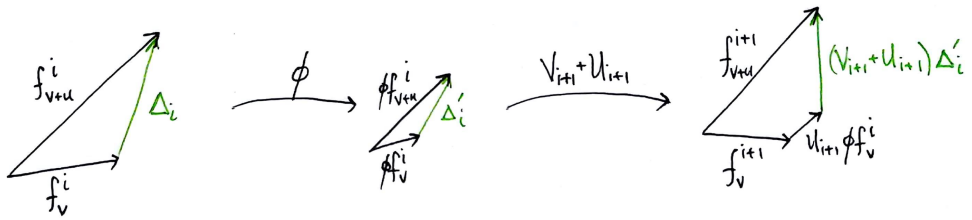

Figure 2: Transformation of $f^i_{\mathbf{v}}(x)$ and $f^i_{\mathbf{v}+\mathbf{u}}(x)$ at the $(i+1)$th layer. Note that ϕ is a contraction.

This quickly shows that:

$$\Delta_{i+1} = U_{i+1}\phi\left(f^i_{\mathbf{v}}(x)\right) + (V_{i+1} + U_{i+1})\Delta'_i,$$

which we can bound using triangle inequality, Cauchy-Schwarz, and the fact that ϕ is a contraction:

$$\begin{aligned}\|\Delta_{i+1}\|_2 &\leq \|U_{i+1}\|_2 \left\|f^i_{\mathbf{v}}(x)\right\|_2 + (\|V_{i+1}\|_2 + \|U_{i+1}\|_2)\|\Delta'_i\|_2 \\ &\leq \|U_{i+1}\|_2 + (1 + \|U_{i+1}\|_2)\|\Delta_i\|_2 \leq \|U_{i+1}\|_2 + \left(1 + \frac{1}{d}\right)\|\Delta_i\|_2 .\end{aligned}$$

From this fact, the rest of the proof follows easily by inducting over the hypothesis:

$$\|\Delta_i\|_2 \leq \left(1 + \frac{1}{d}\right)^i \sum_{k=1}^{i} \|U_k\|_2 .$$

□

Corollary 11. *Let $f_{\mathbf{w}}$ be a ReLu neural network. Let $\mathbf{u} = \mathrm{vec}(\{U_i\}_{i=1}^d)$ be a small perturbation, where each $\|U_i\|_2 / \|W_i\|_2 \leq \frac{1}{d}$. Then for all $x \in B_n(0, r)$,*

$$\|f_{\mathbf{v}+\mathbf{u}} - f_{\mathbf{v}}(x)\|_2 \leq er \sum_{i=1}^{d} \frac{\|U_i\|_2}{\|W_i\|_2}.$$

We are now ready to put all these pieces together to give a risk bound for ReLu neural networks.

Risk Bounds for ReLu Neural Networks

Suppose that we've run our training algorithm using m samples dranw from $B_n(0, r)$, and we've obtained $f_{\mathbf{w}}$. By above, we know we can write $f_{\mathbf{w}} = \alpha f_{\mathbf{v}}$, but we can also redistribute the α even across the layers, so that $f_{\mathbf{w}} = f_{\beta\mathbf{v}}$, where $\alpha = \beta^d$. Thus, we may assume, without loss of generality, that $\mathbf{w} = \beta\mathbf{v}$; that is, the (spectral) norm of each layer is the same.

To bound the risk, our main workhorse will be Proposition 3. That requires producing a posterior distribution Q where a constant fraction of sampled classifiers have similar margins as $f_{\mathbf{w}}$. Let's looks at classifiers whose parameters are $\mathbf{w} + \mathbf{u}$, where $\mathbf{u} \sim \mathcal{N}(0, \sigma^2 I)$ are small perturbations.

To control the variation of the margins, we then need Corollary 11 to control the size of $\|U\|_i$. We know:

$$\Pr_{U_i \sim \mathcal{N}(0,\sigma^2 I)}\left[\|U_i\|_2 > t\right] \leq 2he^{-t^2/2h\sigma^2}.$$

Then, we can union bound over all d layers. Suppose we want to ensure the bound for a constant fraction, say, $1/2$. Then, bound the above by $1/2d$. Now, with probability $1/2$, for all U_i in $\mathbf{u}$,

$$\|U_i\|_2 \leq \sigma\sqrt{2h \ln 4hd}.$$

Applying Corollary 11, we get see that with probability $1/2$, for all $x \in \mathcal{X}$,

$$\|f_{\mathbf{w}+\mathbf{u}}(x) - f_{\mathbf{w}}(x)\|_2 \leq er\beta^d \sum_{i=1}^{d} \frac{1}{\beta}\|U_i\|_2 \leq er\beta^{d-1} d\sigma\sqrt{2h \ln 4hd}.$$

This, we'll want to upper bound by $\gamma/4$, for Proposition 3. However, σ can't depend on β in our analysis, so what we'll do instead is cover all possible β's by a grid of $\widetilde{\beta}$ so that for any β, there is an approximator where:

$$\left|\beta - \widetilde{\beta}\right| \leq \frac{1}{d}\beta.$$

This lets us bound β^{d-1} by $e\widetilde{\beta}^{d-1}$ in the above expression. We'll just have to remember to union bound over all possible $\widetilde{\beta}$'s later. This lets us produce a value for σ:

$$\sigma = \frac{\gamma}{42dr\widetilde{\beta}^{d-1}\sqrt{h\ln(4hd)}}.$$

So, we have $Q = \mathcal{N}(\mathbf{w}, \sigma^2 I)$. A natural prior we could have chosen is $P = \mathcal{N}(0, \sigma^2 I)$, in which case the KL-divergence is:

$$\mathrm{KL}(Q||P) \leq \frac{|\mathbf{w}|^2}{2\sigma^2} = \frac{1}{2\sigma^2}\sum_{i=1}^{d}\|W_i\|_F^2 = O\left(\frac{r^2\alpha^2}{\gamma^2}\ \underbrace{d^2h\ln(dh)}_{\geq\sum_{i=1}^{d}\|\overline{U}_i\|_F^2}\ \sum_{i=1}^{d}\|V_i\|_F^2\right),$$

where $\overline{U}_i$ is the normalized version ($\overline{U}_i = U_i/\|W_i\|_2$).

Plugging the bound for the KL-divergence lets us get a risk bound for this one β. The final step is to figure out how many $\widetilde{\beta}$'s we need for the union bound. Since

$$|f_{\mathbf{w}}(x)| \leq \alpha r \leq \gamma/2,$$

we know that $\beta \geq \left(\frac{\gamma}{2r}\right)^{1/d}$. And if the $\sqrt{\mathrm{KL}/m}$ term is greater than one, then the bound is vacuous. That forces $\beta \leq \left(\frac{\gamma\sqrt{m}}{2r}\right)^{1/d}$. Given that we only need to approximators on this compact interval, it turns out that we only need a cover of size at most $dm^{1/2d}$. It follows that:

Theorem 12. *Let $\mathcal{F}$ be a family of ReLu neural networks $f_{\mathbf{w}} : B_n(0,r) \to \mathbb{R}^k$. Let the dimension of each output layer be bounded by h. Let $f_{\mathbf{w}} = \alpha f_{\mathbf{v}}$ be a classifier obtained after training on m samples. Then, for all margins $\gamma > 0$ and confidence $\delta > 0$,*

$$L_0(f_{\mathbf{w}}) \leq \hat{L}_\gamma(f_{\mathbf{w}}) + \frac{r\alpha}{\gamma}\cdot O\left(\sqrt{\frac{d^2h\ln(dh)\sum_{i=1}^{d}\|V_i\|_F^2 + \frac{\ln dm}{\delta}}{m}}\right).$$

□

References

[Neyshabur 2018] Neyshabur, B., Bhojanapalli, S., and Srebro, N. *A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks.* ICLR. 2018.