CSCI 678: Theoretical Machine Learning
Lecture 4

Fall 2024, Instructor: Haipeng Luo

1 Regression: Fat-Shattering Dimension

In the last lecture, after deriving several upper bounds on the Rademacher complexity of a real-valued function class using different covering numbers, we started looking for a combinatorial parameter that is analogous to the VC dimension for classification problems and that gives a direct upper bound on the covering number. Our first attempt was pseudo-dimension, defined as

$$\text{Pdim}(\mathcal{F}) = \text{VCdim}\left(\{h(x,y) = \text{sign}(f(x) - y) \mid f \in \mathcal{F}\}\right),$$

that is, the largest number n such that there exist n input-output pairs $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times [-1, +1]$, such that for any labeling $s_1, \ldots, s_n \in \{-1, +1\}$, there exists $f \in \mathcal{F}$ with $\text{sign}(f(x_t) - y_t) = s_t$ for all $t = 1, \ldots, n$. While this is a reasonable complexity measure for the linear class, we found that it is infinity for the (learnable) class of non-decreasing functions: for any n, input-output pairs $(0, 0/n), (1, 1/n), (2, 2/n), \ldots$, and any labeling $s_1, \ldots, s_n \in \{-1, +1\}$, we can always find a non-decreasing function that satisfies $\text{sign}(f(x_t) - y_t) = s_t$ for all $t = 1, \ldots, n$ by passing through the points $(0, 0/n + s_1\epsilon), (1, 1/n + s_2\epsilon), (2, 2/n + s_3\epsilon), \ldots$, for some $\epsilon \in (0, \frac{1}{2n}]$. Therefore, finite pseudo-dimension is not necessary for learning.

To fix this issue, we compare the definition of pseudo-dimension and covering number and point out that what is missing for pseudo-dimension is the "scale" α. Intuitively, we need a combinatorial parameter that is also in terms of some scale α, such that it becomes smaller when the scale is larger. One way to do so is to require the induced binary classifier $\text{sign}(f(x) - y)$ to not only predict correctly the labels, but also predict correctly with a certain confidence/margin. This leads to the concept of *fat-shattering*. Specifically, we say that a class $\mathcal{F} \subset [-1, +1]^{\mathcal{X}}$ α-shatters a set $x_1, \ldots, x_n \in \mathcal{X}$, if there exist $y_1, \ldots, y_n \in [-1, +1]$ (called the *witness to shattering*), such that for any labeling $s_1, \ldots, s_n \in \{-1, +1\}$, there exists $f \in \mathcal{F}$ with $s_t(f(x_t) - y_t) \geq \alpha/2$ for all $t = 1, \ldots, n$. The condition $s_t(f(x_t) - y_t) \geq \alpha/2$ exactly corresponds to predicting the label s_t correctly with margin $\alpha/2$. With this concept, the *fat-shattering dimension* of $\mathcal{F}$ at scale α is defined as the size of the largest α-shattered set:

$$\text{fat}(\mathcal{F}, \alpha) = \max\{n \mid \text{there exists a set } x_{1:n} \text{ that is } \alpha\text{-shattered by } \mathcal{F}\}.$$

Clearly, $\text{fat}(\mathcal{F}, \alpha)$ is decreasing in α — if $\mathcal{F}$ α-shatters a set, then it must α'-shatters the same set for any $\alpha' < \alpha$ by definition. It is also clear that when α goes to zero, fat-shattering dimension just becomes pseudo-dimension.

Coming back to the example of the class $\mathcal{F}$ of all non-decreasing functions, we see that in the previous construction of the shattered set, the margin is only $\epsilon \in (0, \frac{1}{2n}]$, which becomes smaller and smaller as we increase n. Therefore, if we require the margin to be at least $\alpha/2$ for some α, then the construction works as long as $n \leq 1/\alpha$, showing that $\text{fat}(\mathcal{F}, \alpha) \geq \lfloor 1/\alpha \rfloor$; however, when $n > 1/\alpha$, the construction no longer works. While this does not prove that $\text{fat}(\mathcal{F}, \alpha)$ is exactly $\lfloor 1/\alpha \rfloor$ (think about why), the following proposition shows that $1/\alpha$ is indeed the right order.

Proposition 1. *If $\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = [-1, +1]$, and $\mathcal{F} \in \mathcal{Y}^{\mathcal{X}}$ is the set of all non-decreasing functions, then* $\text{fat}(\mathcal{F}, \alpha) \leq \frac{4}{\alpha} + 1$ *for any $\alpha > 0$.*

Proof. Suppose $x_1 \le \cdots \le x_n$ is α-shattered by $\mathcal{F}$ with witness $y_1 \le \cdots \le y_n$ (convince yourself that they must be ordered in this way). Let $s_t = +1$ for every odd t and $s_t = -1$ for every even t, and $f \in \mathcal{F}$ be the corresponding function that predicts these labels correctly with margin $\alpha/2$. Then by the fact that f is non-decreasing, for every odd t we must have

$$y_{t+1} - y_t \ge y_{t+1} - f(x_{t+1}) + f(x_t) - y_t \ge s_{t+1}(f(x_{t+1}) - y_{t+1}) + s_t(f(x_t) - y_t) \ge \alpha.$$

Since all y_t's are in the interval $[-1, +1]$ of length 2, we must have $\lfloor \frac{n}{2} \rfloor \times \alpha \le 2$, which means n must not be larger than $4/\alpha + 1$. □

So how is the fat-shattering dimension connected to the covering number? It turns out that there is also an analogue to Sauer's lemma, which we state below without going into the proof.

Theorem 1. *For any $\mathcal{F} \subset [-1, +1]^{\mathcal{X}}$ and $\alpha \in (0, 1)$, we have for any inputs $x_{1:n}$,*

$$\ln \mathcal{N}_2(\mathcal{F}|_{x_{1:n}}, \alpha) = \mathcal{O}\left(\text{fat}(\mathcal{F}, c\alpha) \ln\left(\frac{1}{\alpha}\right)\right)$$

for some absolute constant $c > 0$.

This bound is tighter than the one for pseudo-dimension since $\text{fat}(\mathcal{F}, c\alpha) \le \text{Pdim}(\mathcal{F})$. Note that it is also independent of n, and roughly indicates that $\mathcal{F}|_{x_{1:n}}$ lies in a $\text{fat}(\mathcal{F}, c\alpha)$-dimensional subspace of $[-1, +1]^n$. Applying Dudley integral entropy further gives us a bound on the Rademacher complexity. For example, applying it to the class of non-decreasing functions gives the following:

Proposition 2. *If $\mathcal{X} = \mathbb{R}$, $\mathcal{Y} = [-1, +1]$, and $\mathcal{F} \in \mathcal{Y}^{\mathcal{X}}$ is the set of all non-decreasing functions, then $\mathcal{R}^{\text{iid}}(\mathcal{F}) = \mathcal{O}(\sqrt{1/n})$.*

Proof. We apply Dudley integral entropy with $\ln \mathcal{N}_2(\mathcal{F}|_{x_{1:n}}, \alpha) = \mathcal{O}\left(\frac{1}{\alpha} \ln\left(\frac{1}{\alpha}\right)\right) = \mathcal{O}\left(\frac{1}{\alpha^{3/2}}\right)$:

$$\mathcal{R}^{\text{iid}}(\mathcal{F}) = \mathcal{O}\left(\inf_{\alpha}\left(\alpha + \frac{1}{\sqrt{n}} \int_{\alpha}^{1} \frac{d\delta}{\delta^{3/4}}\right)\right) = \mathcal{O}\left(\frac{1}{\sqrt{n}}\right).$$

□

Compared to the bound $\mathcal{R}^{\text{iid}}(\mathcal{F}) = \mathcal{O}(\sqrt{(\ln n)/n})$ we obtained in the last lecture via a bound $\mathcal{N}_\infty(\mathcal{F}|_{x_{1:n}}, \alpha) \le (n+1)^{\frac{1}{\alpha}}$ on the ℓ_∞ covering number, here we further improve it (removing the $\ln n$ factor) by using a direct bound on the ℓ_2 covering number via fat-shattering dimension. This shows the advantage of going for the fat-shattering dimension directly.

In fact, unlike the pseudo-dimension, it has been shown that a finite fat-shattering dimension is *necessary* for the learnability of $\mathcal{F}$. Putting everything together, we have thus obtained the following sequence of tight upper bounds on the value of the statistical learning game:

$$\begin{aligned}
\mathcal{V}^{\text{iid}}(\mathcal{F}, n) &\le \sup_{\mathcal{P}}\left(\mathbb{E}\left[\sup_{f \in \mathcal{F}}\left(L(f) - \frac{1}{n}\sum_{t=1}^{n} \ell(f, z_t)\right)\right]\right) && \text{(using ERM)} \\
&\le 2 \sup_{\mathcal{P}} \mathcal{R}^{\text{iid}}(\ell(\mathcal{F})) && \text{(symmetrization)} \\
&\le 2G \sup_{\mathcal{P}} \mathcal{R}^{\text{iid}}(\mathcal{F}) && \text{(erasing the loss)} \\
&\le 2G \sup_{x_{1:n}} \min_{0 \le \alpha \le 1}\left(4\alpha + \frac{12}{\sqrt{n}} \int_{\alpha}^{1} \sqrt{\ln \mathcal{N}_2(\mathcal{F}|_{x_{1:n}}, \delta)} d\delta\right) && \text{(Dudley entropy integral)} \\
&\le 2G \min_{0 \le \alpha \le 1} \mathcal{O}\left(\alpha + \frac{1}{\sqrt{n}} \int_{\alpha}^{1} \sqrt{\text{fat}(\mathcal{F}, c\delta) \ln\left(\frac{1}{\delta}\right)} d\delta\right). && \text{(Theorem 1)}
\end{aligned}$$

2 Towards Understanding the Complexity of Neural Networks

At this point, we have covered all basic concepts and tools in understanding statistical learning. Let's now do a case study on neural networks, the arguably most important models in modern machine learning practice, and see if our theory can explain its practical success (at least partially).

One of the major puzzles of neural nets is why it allows generalization, even when the number of parameters is several orders more than the size of the training set. Indeed, a modern neural net architecture could easily have millions or even billions parameters, leading to a highly expressive model that often enjoys *zero* training error. To make sure such amazing performance on the training set generalizes to unseen data, our theory says that uniform convergence needs to hold, or equivalently, the class of neural nets needs to be learnable.

For classification problem, we know that learnability can be characterized by VC-dimension. However, the VC-dimension of a neural net appears to be often at least as large as the size of the training set: for example, a fully connected feed-forward neural net with about 1 million parameters can perfectly fit the CIFAR10 dataset with 50K images and *completely random labels*, a strong evidence that this model class shatters the CIFAR10 dataset [Zhang et al., 2017]. Based on the theory we have discussed, in particular, $\mathcal{V}^{\text{iid}}(\mathcal{F}, n) \lessapprox \sqrt{\text{VCdim}(\mathcal{F})/n}$, we should not expect that this class generalizes well on unseen data. This is, however, contrary to the fact that when trained on clean CIFAR10, the model does achieve about 50% accuracy (highly nontrivial for a 10-class problem). In fact, a much higher (close to 90%) accuracy can be achieved by a convolutional neural net with a similar number of parameters.

In fact, other complexity measures such as Rademacher complexity and covering number seemingly also fail to explain why neural nets generalize, because even for the class of linear functions, a highly degenerated special case of neural nets, the excess risk bounds we have derived are all of the form $\mathcal{V}^{\text{iid}}(\mathcal{F}, n) \lessapprox \sqrt{d/n}$ with d bing the dimension (essentially the same as number of parameters), which, as mentioned, is often much larger than n when training neural nets. So, is all the theory we have discussed so far useless for understanding neural nets?

Not quite. The discussion above just indicates that VC-dimension or number of parameters is likely not the real intrinsic quantity that measures the complexity of a neural net. Indeed, intuitively, the magnitude of the weights of a neural net should play an even more important role in determining its complexity. In fact, we have already seen one such example in HW1, where we show that for linear functions with weight vector bounded by b in ℓ_2 norm, its Rademacher complexity is at most $\frac{b}{n}\sqrt{\sum_{t=1}^n \|x_t\|_2^2}$, which is dimension-independent and shows that the real complexity of the class is determined by the norm of the weight vectors instead of their dimension.

In the rest of the lecture, we will derive something similar for neural nets, mainly taken from [Bartlett et al., 2017]. As a warm-up, we will first discuss how to derive an almost dimension-independent covering number bound for linear class, then generalize it to the matrix case which corresponds to one layer of a neural net, and finally further generalize it to a full neural net. After all the theoretical derivations, we will come back to some empirical results and discuss whether these new bounds indeed explain why neural nets work in practice.

2.1 Almost dimension-independent covering number: a warm-up

First, we prove the following log covering number bound for a class of linear functions (you have already seen this result in HW1). Importantly, its explicit dependence on d is only logarithmic.

Theorem 2. *For the class* $\mathcal{F} = \left\{ f_\theta(x) = \langle \theta, x \rangle \mid \theta \in \mathbb{R}^d, \|\theta\|_2 \le b \right\}$, *we have* $\ln \mathcal{N}_2(\mathcal{F}|_{x_{1:n}}, \alpha) \le \frac{b^2 \|X\|_F^2 \ln(2d)}{n\alpha^2}$, *where* $X \in \mathbb{R}^{n\times d}$ *is the data matrix obtained by stacking* $x_1^\top, \dots, x_n^\top$ *as rows and* $\|X\|_F = \sqrt{\sum_{t=1}^n \|x_t\|_2^2}$ *is the Frobenius norm of* X.

In contrast, the bound we developed last time is $\ln \mathcal{N}_2(\mathcal{F}|_{x_{1:n}}, \alpha) \le \ln \mathcal{N}(\mathcal{F}, \alpha) \le d \ln\left(\frac{2b}{\alpha} + 1\right)$ (assuming $\|x_t\|_2 \le 1$), which is linear in d. Note that, even though this new bound has a worse dependence on α, in the end it does not really affect the rate in n for the Rademacher complexity (thanks to the Dudley entropy integral) — indeed, in HW1, you are asked to use this result to prove a Rademacher complexity bound $\frac{b}{n}\sqrt{\sum_{t=1}^n \|x_t\|_2^2}$ (up to log factors), which is of order $1/\sqrt{n}$, the same as what we proved last time, but without polynomial dependence on the dimension d.

To prove this Theorem 2, we recall the following result, also proven in HW1:

Lemma 1. *Let $v_1, \dots, v_d \in B_2^n$ be d points within the n-dimensional ℓ_2-norm unit ball and*

$$\mathcal{S} = \left\{ \sum_{i=1}^d \beta_i v_i \;\middle|\; \beta_i \geq 0,\ \forall i,\ \text{and}\ \sum_{i=1}^d \beta_i \leq B \right\}$$

be the convex hull of these d points scaled by $B > 0$. We have $\ln \mathcal{N}_2(\mathcal{S}, \alpha) \leq \frac{B^2 \ln d}{n\alpha^2}$.

Proof of Theorem 2. It suffices to write $\mathcal{F}|_{x_{1:n}}$ in the form of $\mathcal{S}$ in Lemma 1. To do this, note that each element in $\mathcal{F}|_{x_{1:n}}$ is in the form $X\theta$ for some θ. To write this as $\sum_{i=1}^d \beta_i v_i$ where $v_i \in B_2^n$, we let $v_i = \frac{X_{:,i}}{\|X_{:,i}\|_2}$ where $X_{:,i}$ is the i-th column of X, so that $\|v_i\|_2 = 1$. Thus, with $\beta_i = \theta_i \|X_{:,i}\|_2$, we have $X\theta = \sum_{i=1}^d \beta_i v_i$. However, one caveat is that β_i might be negative, inconsistent with the definition of $\mathcal{S}$. This can be fixed by realizing

$$X\theta = \sum_{i=1}^d \beta_i v_i = \sum_{i=1}^d (\mathbb{I}\{\beta_i \geq 0\}\beta_i v_i + \mathbb{I}\{\beta_i < 0\}(-\beta_i) \cdot (-v_i));$$

hence, to be consistent with the definition of $\mathcal{S}$, all we need is to double the dimension (making it $2d$ instead of d, with $2d$ base vectors $\pm v_1, \dots, \pm v_d$). It remains to calculate the value B:

$$\sum_{i=1}^d (\mathbb{I}\{\beta_i \geq 0\}\beta_i + \mathbb{I}\{\beta_i < 0\}(-\beta_i)) = \|\beta\|_1 \leq \|\theta\|_2 \|X\|_F \leq b \|X\|_F \triangleq B,$$

where the first inequality is by Cauchy-Schwarz inequality. Applying Lemma 1 then finishes the proof. □

We note without going into details that similar results can also be proven for other prime-dual norm pairs using the same argument (you should give it a try).

2.2 Almost dimension-independent covering number: one-layer neural nets

Next, we consider a one-layer neural net that maps an input x in $\mathbb{R}^d$ to an output in $\mathbb{R}^m$ via a weight matrix $W \in \mathbb{R}^{m \times d}$ and a coordinate-wise ReLU activation function σ.[1] For reason that will become clear in the analysis, we care about the $(1,2)$ mixed norm of W: $\|W\|_{1,2} = \left\|\left(\|W_{:,1}\|_1, \dots, \|W_{:,d}\|_1\right)\right\|_2$, that is, the ℓ_2 norm of the ℓ_1 norms of the columns.

Theorem 3. *For the class $\mathcal{F} = \left\{x \to \sigma(Wx) \mid W \in \mathbb{R}^{m \times d}, \|W\|_{1,2} \leq b\right\}$ (where σ is ReLU), we have $\ln \mathcal{N}_2(\mathcal{F}|_{x_{1:n}}, \alpha) \leq \frac{b^2 \|X\|_F^2 \ln(2dm)}{nm\alpha^2}$.*

Proof. Note that each element of $\mathcal{F}|_{x_{1:n}}$ is of the form $\sigma(XW^\top) \in \mathbb{R}^{n \times m}$. Since ReLU is 1-Lipschitz, it suffices to cover $\{XW^\top : \|W\|_{1,2} \leq b\}$. To do this, we rewrite $XW^\top$ as

$$XW^\top = X \sum_{i=1}^d \sum_{j=1}^m W_{ji} e_i e_j^\top = \sum_{i=1}^d \sum_{j=1}^m W_{ji} X e_i e_j^\top = \sum_{i=1}^d \sum_{j=1}^m \beta_{ij} v_{ij}$$

where $\beta_{ij} = W_{ji} \left\|X e_i e_j^\top\right\|_F$ and $v_{ij} = X e_i e_j^\top / \left\|X e_i e_j^\top\right\|_F$. By seeing each v_{ij} as an nm dimensional vector, we can thus apply Lemma 1 (with n there being nm here and d there being $2dm$ here; the factor of 2 is again due to the caveat that β_{ij} might be negative). It remains to calculate the value of B:

$$\begin{aligned}
\sum_{i=1}^d \sum_{j=1}^m |\beta_{ij}| &= \sum_{i=1}^d \sum_{j=1}^m |W_{ji}| \left\|X e_i e_j^\top\right\|_F = \sum_{i=1}^d \left(\sum_{j=1}^m |W_{ji}|\right) \|X e_i\|_2 \\
&\leq \sqrt{\sum_{i=1}^d \left(\sum_{j=1}^m |W_{ji}|\right)^2}\ \|X\|_F = \|W\|_{1,2} \|X\|_F \leq b \|X\|_F \triangleq B,
\end{aligned}$$

[1] A one-dimensional ReLU is simply defined as $\sigma(x) = \max\{x, 0\}$.

where the first inequality is again by Cauchy-Schwarz inequality. Applying Lemma 1 then finishes the proof. □

Note that the bound again has very mild dependence on the number of parameters dm. Also, note that the $(1,2)$ mixed norm naturally appears in the analysis. In fact, Bartlett et al. [2017] compute the value B in a slightly different way and obtain a bound in terms of the $(2,1)$ mixed norm of $W^\top$ instead (also using Cauchy-Schwarz inequality):

$$\sum_{i=1}^{d}\sum_{j=1}^{m}|\beta_{ij}| = \sum_{j=1}^{m}\sum_{i=1}^{d}|W_{ji}|\left\|Xe_ie_j^\top\right\|_F \le \sum_{j=1}^{m}\|W_{j:}\|_2\|X\|_F = \left\|W^\top\right\|_{2,1}\|X\|_F.$$

However, since $\|W\|_{1,2} \le \left\|W^\top\right\|_{2,1}$ always holds (you can prove this by squaring both sides and then applying Cauchy-Schwarz), what we present here appears to be a slight improvement over Bartlett et al. [2017].

2.3 Almost dimension-independent covering number: multi-layer neural nets

Finally, we are ready to generalize the result to general fully connected feed-forward multi-layer neural nets. To do so, we first establish a few notations:

- We use H to denote the total number of layers. Each layer $h \in \{1,\dots,H\}$ maps an input in $\mathbb{R}^{d_{h-1}}$ to an output in $\mathbb{R}^{d_h}$, with $d_0 = d$. Denote $d_{\max} = \max\{d_0, d_1, \dots, d_H\}$.
- In particular, layer h is parametrized by a weight matrix from the space $\mathcal{W}_h = \left\{W \in \mathbb{R}^{d_h \times d_{h-1}} \mid \|W\|_{1,2} \le b_h, \|W\|_2 \le s_h\right\}$ for some positive numbers b_h and s_h. Here, $\|W\|_2 = \max_{x\neq 0}\|Wx\|_2 / \|x\|_2$ is the spectral norm of W (the reason for considering spectral norm will become clear soon).
- Let $\mathcal{F}_h = \{x \to \sigma(W_h\cdots\sigma(W_2\sigma(W_1x))\cdots) \mid W_k \in \mathcal{W}_k,\ \forall k \le h\}$ be a class of h-layer neural nets. The class $\mathcal{F} = \mathcal{F}_H$ of H-layer neural nets is what we ultimately care about.
- For some $\gamma_h \ge 0$ and $M \in \mathbb{R}^{n\times d_{h-1}}$ (think of it as n inputs to layer h), define $\mathcal{C}(M, \mathcal{W}_h, \gamma_h)$ as a minimum $\frac{\gamma_h}{\sqrt{nd_h}}$-cover of the set $\left\{\sigma(MW^\top) \mid W \in \mathcal{W}_h\right\}$ with respect to the Frobenius norm. By Theorem 3, we know that $\ln|\mathcal{C}(M, \mathcal{W}_h, \gamma_h)| \le \frac{b_h^2\|M\|_F^2\ln(2d_{h-1}d_h)}{\gamma_h^2} \le \frac{b_h^2\|M\|_F^2\ln(2d_{\max}^2)}{\gamma_h^2}$. Also, without loss of generality, assume $\mathcal{C}(M, \mathcal{W}_h, \gamma_h) \subset \left\{\sigma(MW^\top) \mid W \in \mathcal{W}_h\right\}$.[2]
- Further define recursively $S_h = \cup_{M\in S_{h-1}}\mathcal{C}(M, \mathcal{W}_h, \gamma_h)$, with $S_0 = \{X\}$ where $X \in \mathbb{R}^{n\times d}$ is the data matrix. Note that each element M_h of S_h is in the form $\sigma(M_{h-1}W_h^\top)$ for some $M_{h-1} \in S_{h-1}$ and satisfies $\|M_h\|_F = \left\|\sigma(M_{h-1}W_h^\top) - \sigma(0)\right\|_F \le \left\|M_{h-1}W_h^\top\right\|_F \le s_h\|M_{h-1}\|_F$ where the inequalities are by the Lipschitzness of ReLU and the fact $\|W_h\|_2 \le s_h$. Applying this recursively shows $\|M_h\|_F \le \|X\|_F\prod_{k=1}^h s_k$ and thus

$$\ln|S_h| \le \ln|S_{h-1}| + \frac{b_h^2\|X\|_F^2\left(\prod_{k<h}s_k^2\right)\ln(2d_{\max}^2)}{\gamma_h^2}. \tag{1}$$

We now prove that each S_h is a cover for $\mathcal{F}_h|_{x_{1:n}}$ (both of which are subsets of $\mathbb{R}^{n\times d_h}$).

Lemma 2. *For each $h = 1,\dots,H$, S_h is an $\frac{\alpha_h}{\sqrt{nd_h}}$-cover of $\mathcal{F}_h|_{x_{1:n}}$ with respect to the Frobenius norm, where $\alpha_h = \gamma_h + s_h\alpha_{h-1}$ and $\alpha_0 = 0$.*

Proof. We prove the claim by induction on h. For the base case $h = 1$, we have $S_1 = \mathcal{C}(X, \mathcal{W}_1, \gamma_1)$ and $\alpha_1 = \gamma_1$, so the claim holds trivially by the definition of $\mathcal{C}$. For a general h, assume that the statement holds for $h-1$. Then, for any $\sigma(\cdots\sigma(\sigma(XW_1^\top)W_2^\top)\cdots W_h^\top) \in \mathcal{F}_h|_{x_{1:n}}$, we first find an $M_{h-1} \in S_{h-1}$ such that

$$\left\|M_{h-1} - \sigma(\cdots\sigma(\sigma(XW_1^\top)W_2^\top)\cdots W_{h-1}^\top)\right\|_F \le \alpha_{h-1}, \tag{2}$$

[2] For the reason why this is without loss of generality, refer to HW1 Question 3(a)i.

which must exist due to the inductive hypothesis. Then, we find $M_h \in \mathcal{C}(M_{h-1}, \mathcal{W}_h, \gamma_h)$ (consequently, $M_h \in S_h$) such that

$$\left\|M_h - \sigma(M_{h-1}W_h^\top)\right\|_F \le \gamma_h, \tag{3}$$

which must exist due to the definition of $\mathcal{C}$. Combining these two, we thus have

$$\begin{aligned}
&\left\|M_h - \sigma(\cdots\sigma(\sigma(XW_1^\top)W_2^\top)\cdots W_h^\top)\right\|_F \\
&\le \left\|M_h - \sigma(M_{h-1}W_h^\top)\right\|_F + \left\|\sigma(M_{h-1}W_h^\top) - \sigma(\cdots\sigma(\sigma(XW_1^\top)W_2^\top)\cdots W_h^\top)\right\|_F \\
&\qquad\qquad \text{(triangle inequality)} \\
&\le \gamma_h + \left\|M_{h-1}W_h^\top - \sigma(\cdots\sigma(\sigma(XW_1^\top)W_2^\top)\cdots W_{h-1}^\top)W_h^\top\right\|_F \\
&\qquad\qquad \text{(Eq. (3) and ReLU is 1-Lipschitz)} \\
&\le \gamma_h + \left\|M_{h-1} - \sigma(\cdots\sigma(\sigma(XW_1^\top)W_2^\top)\cdots W_{h-1}^\top)\right\|_F \|W_h\|_2 \quad \text{(definition of spectral norm)} \\
&\le \gamma_h + s_h\alpha_{h-1} \qquad\qquad \text{(Eq. (2) and } \|W_h\|_2 \le s_h\text{)} \\
&= \alpha_h,
\end{aligned}$$

which finishes the induction. □

Using this lemma and setting $\gamma_1, \ldots, \gamma_H$ appropriately, we finally prove our main theorem.

Theorem 4. *For the class of multi-layer neural nets $\mathcal{F} = \mathcal{F}_H$, we have*

$$\ln \mathcal{N}(\mathcal{F}|_{x_{1:n}}, \alpha) \le \frac{\|X\|_F^2 \ln(2d_{\max}^2)}{nd_H\alpha^2}\left(\prod_{h=1}^H s_h^2\right)\left(\sum_{h=1}^H \left(\frac{b_h}{s_h}\right)^{\frac{2}{3}}\right)^3.$$

Proof. By Lemma 2, we know that $\ln \mathcal{N}\left(\mathcal{F}|_{x_{1:n}}, \frac{\alpha_H}{\sqrt{nd_H}}\right) \le \ln |S_H|$, which, based on Eq. (1), is at most

$$\sum_{h=1}^H \frac{b_h^2 \|X\|_F^2 \left(\prod_{k<h} s_k^2\right)\ln(2d_{\max}^2)}{\gamma_h^2} = \|X\|_F^2 \ln(2d_{\max}^2)\sum_{h=1}^H \frac{b_h^2\left(\prod_{k<h} s_k^2\right)}{\gamma_h^2}.$$

Therefore, our goal is to minimize the bound above by picking $\gamma_1, \ldots, \gamma_H$, subject to the constraint

$$\alpha_H = \gamma_H + s_H\alpha_{H-1} = \cdots = \sum_{h=1}^H \gamma_h \prod_{k=h+1}^H s_k.$$

To do so, it is convenient to think of distributing α_H total error over the H layers according to some distribution $\rho \in \Delta_H$, where the h-th layer is allowed to have error $\alpha_H\rho_h = \gamma_h \prod_{k=h+1}^H s_k$. Plugging $\gamma_h = \alpha_H\rho_h / \prod_{k=h+1}^H s_k$ into the objective function, we obtain

$$\frac{\|X\|_F^2 \ln(2d_{\max}^2)}{\alpha_H^2}\left(\prod_{h=1}^H s_h^2\right)\sum_{h=1}^H \frac{b_h^2}{\rho_h^2 s_h^2},$$

so it remains to minimize $\sum_{h=1}^H \frac{b_h^2}{\rho_h^2 s_h^2}$ subject to $\rho \in \Delta_H$. Using KKT condition, it can be verified that the optimal ρ_h is proportional to $\left(\frac{b_h}{s_h}\right)^{\frac{2}{3}}$, and the optimal objective value is $\left(\sum_{h=1}^H \left(\frac{b_h}{s_h}\right)^{\frac{2}{3}}\right)^3$. Combining everything and rewriting α_H as $\alpha\sqrt{nd_H}$ competes the proof. □

Since the dependence on α is again $1/\alpha^2$, the same as in the simple linear case, via direct calculation similar to that in HW1 Question 1(b), we know that the Rademacher complexity of $\mathcal{F}_H$ is of order

$$\widetilde{\mathcal{O}}\left(\frac{\|X\|_F}{n}\left(\prod_{h=1}^H s_h\right)\left(\sum_{h=1}^H \left(\frac{b_h}{s_h}\right)^{\frac{2}{3}}\right)^{\frac{3}{2}}\right),$$

which again has no explicit dependence on the number of parameters (other than logarithmic factors). In light of this bound, for a neural net with weight matrices $W = (W_1, \ldots, W_H)$, we define its *spectral complexity* as

$$R(W) = \left(\prod_{h=1}^H \|W_h\|_2\right)\left(\sum_{h=1}^H \left(\frac{\|W_h\|_{1,2}}{\|W_h\|_2}\right)^{\frac{2}{3}}\right)^{\frac{3}{2}}.$$

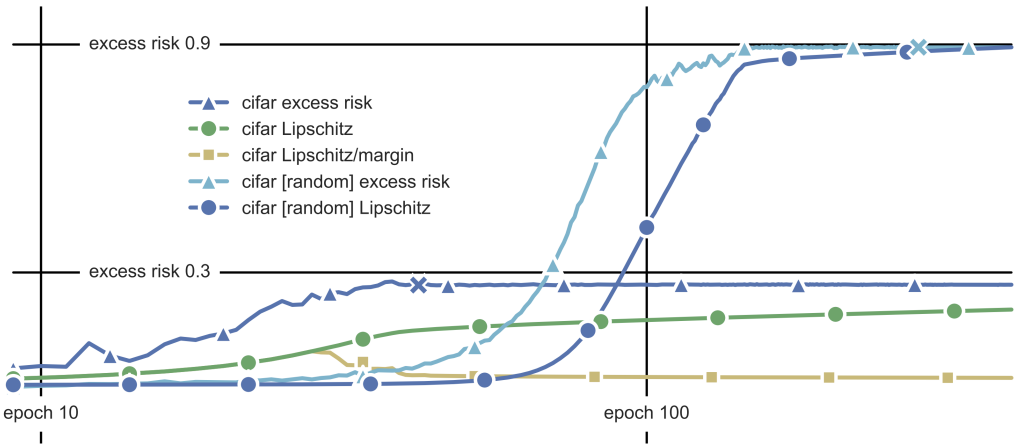


Figure 1: Experiment results for training an AlexNet on CIFAR10 with original or random labels

2.4 Explaining the generalization of neural nets using their spectral complexity and margin

What we have derived so far suggests that the generalization error of a neural net is determined by its spectral complexity, rather than its number of parameters. We will discuss several empirical results from [Bartlett et al., 2017] that support this claim.

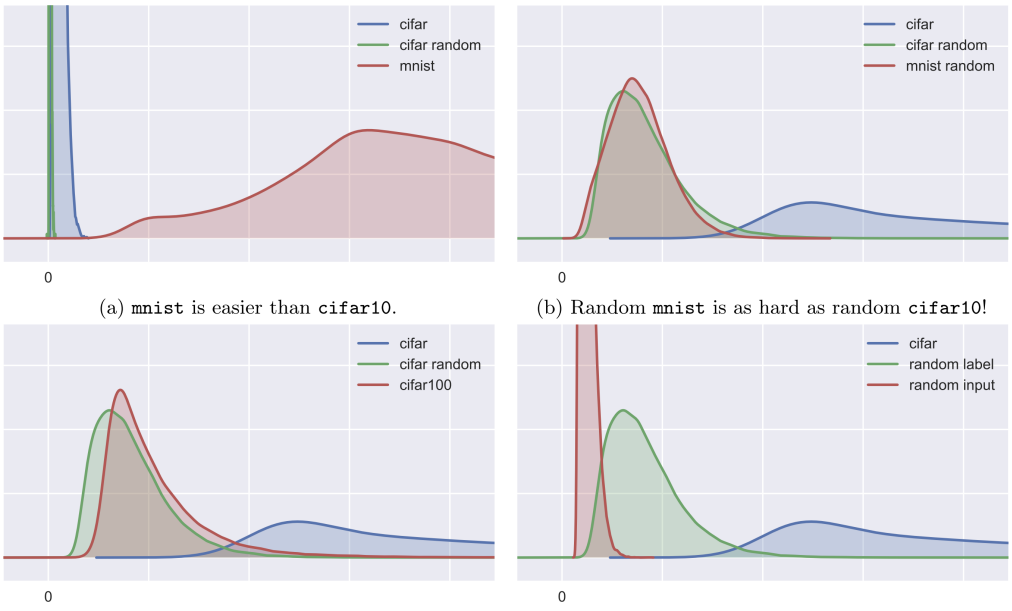
First, Figure 1 is the result for training an AlexNet (a kind of convolutional neural networks) on CIFAR10 with original or random labels. The plots show how the "excess risk" and "Lipschitzness" of the trained AlexNet change over different epochs; here, "excess risk" in fact means test error minus training error (i.e., generalization error), which is why it generally goes up over time, and the "Lipschitzness" is basically the spectral complexity up to constants. For the generalization error plot, the cross mark indicates the first time when the training error gets to zero, which happens much earlier when trained on original labels than on random labels. After this point, the generalization error is simply the test error, which plateaus to about 0.3 for original labels and 0.9 for random labels (as it should). Apparently, the number of parameters, which is the same for both cases, cannot explain the substantial difference in the generalization error. The spectral complexity, on the other hand, is tightly correlated with the generalization error as the plots show: the AlexNet trained on original labels has a much lower spectral complexity compared to that trained on random labels. This indicates that spectral complexity is indeed a better complexity measure for neural nets.

Margin However, a closer look at the plots reveals that the spectral complexity does keep growing even after the generalization error plateaus, so the previous bound of order $\frac{1}{n}\|X\|_F R(W)$ is still not perfectly capturing the generalization error. To further explain this, [Bartlett et al., 2017] use the concept of margin, which, as discussed earlier in this lecture, is a way to measure the confidence on predictions. Roughly speaking, as the number of training epochs increases after the training error gets to zero, even though the spectral complexity of the neural net still keeps increasing, it also becomes more and more confident on its predictions.

More formally, note that a neural net f_W parametrized by W makes its prediction on x via $\operatorname{argmax}_j f_W(x)_j$. We can thus define its margin on example (x, y) via $\mathcal{M}(f_W, x, y) \triangleq f_W(x)_y - \max_{j \neq y} f_W(x)_j$, which is nonnegative if and only if f_W makes a correct prediction. When the margin is positive, the larger it is, the more confident the neural net is on this prediction. Using what we have discussed so far and other standard tools, [Bartlett et al., 2017] show that with high probability, for any neural net f_W and any value of $\gamma > 0$, we have

$$\Pr\left\{\max_j f_W(x)_j \neq y\right\} \leq \frac{1}{n}\sum_{t=1}^{n} \mathbb{I}\{\mathcal{M}(f_W, x_t, y_t) < \gamma\} + \widetilde{\mathcal{O}}\left(\frac{\|X\|_F R(W)}{\gamma n}\right). \tag{4}$$

Note that the value of γ controls the trade-off between the two terms in the bound (with the first term increasing in γ and the second decreasing in it), but the bound holds for all γ *simultaneously*, so one



(a) `mnist` is easier than `cifar10`.

(b) Random `mnist` is as hard as random `cifar10`!

(c) `cifar100` is as hard as `cifar10` with random labels!

(d) Random inputs are harder than random labels.

Figure 2: Comparing (normalized) margin distributions for different tasks

can pick the one that achieves the best trade-off. Importantly, this happens only in the analysis but not the algorithm (indeed, γ is not a hyper-parameter of the algorithm).

In particular, when f_W has zero training error, we can pick $\gamma_{\min} = \min_t \mathcal{M}(f_W, x_t, y_t)$ to be the smallest margin on the training set and obtain $\Pr\{\max_j f_W(x)_j \neq y\} \leq \widetilde{\mathcal{O}}\left(\frac{\|X\|_F R(W)}{\gamma_{\min} n}\right)$. This shows that even if the spectral complexity keeps increasing, as long as the margin also increases accordingly, the generalization error does not necessarily go up. Indeed, the square-marked curve in Figure 1 shows how the spectral complexity normalized by the margin behaves over time and confirms that it does stop growing.

Margin distributions Moreover, by normalizing the margin appropriately and looking at its empirical distribution, we can qualitatively compare the difficulties of different tasks, as done in Figure 2. Here, each curve is the empirical distribution of the margin normalized by $\|X\|_F R(W)/n$ for training an AlexNet on a particular task. To understand why the normalization is done in this way, note that we can equivalently write Eq. (4) as

$$\Pr\left\{\max_j f_W(x)_j \neq y\right\} \leq \frac{1}{n}\sum_{t=1}^{n} \mathbb{I}\left\{\frac{\mathcal{M}(f_W, x_t, y_t)}{\|X\|_F R(W)/n} < \gamma\right\} + \widetilde{\mathcal{O}}\left(\frac{1}{\gamma}\right),$$

so the CDF of the distribution plots in Figure 2 is basically the term $\frac{1}{n}\sum_{t=1}^{n} \mathbb{I}\left\{\frac{\mathcal{M}(f_W, x_t, y_t)}{\|X\|_F R(W)/n} < \gamma\right\}$.

If a margin distribution puts more mass towards the right, then its corresponding task is qualitatively easier. Therefore, Figure 2 tells us that (a) the MNIST dataset is easier than CIFAR10; (b) they are almost as hard with random labels; (c) CIFAR100 (with 100 classes) is almost as hard as CIFAR10 with random labels; (d) learning with random inputs (x's) is even harder than with random labels.

Closing remark To recap, we have used spectral complexity and margin to provide a reasonable explanation to the generalization ability of neural nets, despite its huge number of parameters compared to the size of the training set. We remark that what we did not cover at all is why the neural net trained on clean data tends to enjoy a low spectral complexity. Note that the training here is not ERM over a class of neural nets with a particular spectral complexity; rather, it is often just running variants of stochastic gradient descent (SGD) without any explicit constraints on the norm of weight

matrices. This is the (nonconvex) optimization part that we have been ignoring in this course, and understanding why SGD has such implicit bias towards networks with low complexity is an active research area.

References

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

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.