Generalization errors for Deep Learning

김지수 (Jisu KIM)

딥러닝의 통계적 이해 (Deep Learning: Statistical Perspective), 2024년 2학기

This lecture note is a combination of Prof. Joong-Ho Won's "Deep Learning: Statistical Perspective" with other lecture notes. Main references are:

Tong Zhang, Mathematical Analysis of Machine Learning Algorithms, https://tongzhang-ml.org/lt-book.html

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

Weinan E, Chao Ma, Stephan Wojtowytsch, Lei Wu, Towards a Mathematical Understanding of Neural Network-Based Machine Learning: what we know and what we don't, https://arxiv.org/abs/2009.10713/

1 Review

1.1 Basic Model for Supervised Learning

- Input(입력) / Covariate(설명 변수) : $x \in \mathbb{R}^d$, so $x = (x_1, \ldots, x_d)$.
- Output(출력) / Response(반응 변수): $y \in \mathcal{Y}$. If y is categorical, then supervised learning is "classification", and if y is continuous, then supervised learning is "regression".
- Model(모형) :

  $$y \approx f(x).$$

  If we include the error ϵ to the model, then it can be also written as

  $$y = \phi(f(x), \epsilon).$$

  For many cases, we assume additive noise, so

  $$y = f(x) + \epsilon.$$

- Assumption(가정): f belongs to a family of functions $\mathcal{M}$. This is the assumption of a model: a model can be still used when the corresponding assumption is not satisfied in your data.
- Loss function(손실 함수): $\ell(y, a)$. A loss function measures the difference between estimated and true values for an instance of data.
- Training data(학습 자료): $\mathcal{T} = \{(y_i, x_i), i = 1, \ldots, n\}$, where (y_i, x_i) is a sample from a probability distribution P_i. For many cases we assume i.i.d., or x_i's are fixed and y_i's are i.i.d..
- Goal(목적): we want to find f that minimizes the expected prediction error,

  $$f^0 = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{(Y,X)\sim P}\left[\ell(Y, f(X))\right].$$

  Here, $\mathcal{F}$ can be different from $\mathcal{M}$; $\mathcal{F}$ can be smaller then $\mathcal{M}$.

- Prediction model(예측 모형): f^0 is unknown, so we estimate f^0 by $\hat{f}$ using data. For many cases we minimizes on the empirical prediction error, that is taking the expectation on the empirical distribution $P_n = \frac{1}{n}\sum_{i=1}^n \delta_{(Y_i, X_i)}$.

  $$\hat{f} = \arg\min_{f \in \mathcal{F}} \mathbb{E}_{P_n}\left[\ell(Y, f(X))\right] = \arg\min_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^n \ell(Y_i, f(X_i)).$$

- Prediction(예측): if $\hat{f}$ is a predicted function, and x is a new input, then we predict unknown y by $\hat{f}(x)$.

1.2 Rademacher complexity

Random variables $\xi_1, \ldots, \xi_n$ are called *Rademacher random variables* if they are independent, identically distributed and $\mathbb{P}(\xi_i = 1) = \mathbb{P}(\xi_i = -1) = 1/2$. Define the *Rademacher complexity* of $\mathcal{F}$ by

$$\mathsf{Rad}_n(\mathcal{F}) = \mathbb{E}\left(\sup_{f \in \mathcal{F}} \left(\frac{1}{n}\sum_{i=1}^{n} \xi_i f(Z_i)\right)\right).$$

Some authors use a slightly different definition, namely,

$$\mathsf{Rad}_n(\mathcal{F}) = \mathbb{E}\left(\sup_{f \in \mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^{n} \xi_i f(Z_i)\right|\right).$$

You can use either one. They lead to essentially the same results. In fact, under mild condition, two Rademacher complexity are closely related, as below:

Lemma 1. *Let $\xi = (\xi_1, \ldots, \xi_n)$ be i.i.d. Rademacher. Suppose that for any $\xi \in \{\pm 1\}^n$, $\sup_{f \in \mathcal{F}} \sum_{i=1}^{n} \xi_i f(Z_i) \geq 0$. Then*

$$\mathbb{E}_\xi\left[\sup_{f \in \mathcal{F}} \left|\frac{1}{n}\sum_{i=1}^{n} \xi_i f(Z_i)\right| \middle| Z\right] \leq \mathbb{E}_\xi\left[\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \xi_i f(Z_i) \middle| Z\right].$$

Proof. Left as HW. □

Intuitively, $\mathsf{Rad}_n(\mathcal{F})$ is large if we can find functions $f \in \mathcal{F}$ that "look like" random noise, that is, they are highly correlated with $\sigma_1, \ldots, \sigma_n$. Here are some properties of the Rademacher complexity.

Lemma. *1. If $\mathcal{F} \subset \mathcal{G}$ then $\mathsf{Rad}_n(\mathcal{F}, Z^n) \leq \mathsf{Rad}_n(\mathcal{G}, Z^n)$.*

2. *Let $\mathsf{conv}(\mathcal{F})$ denote the convex hull of $\mathcal{F}$. Then $\mathsf{Rad}_n(\mathcal{F}, Z^n) = \mathsf{Rad}_n(\mathsf{conv}(\mathcal{F}), Z^n)$.*

3. *For any $c \in \mathbb{R}$, $\mathsf{Rad}_n(c\mathcal{F}, Z^n) = |c|\mathsf{Rad}_n(\mathcal{F}, Z^n)$.*

4. *Let $g : \mathbb{R} \to \mathbb{R}$ be such that $|g(y) - g(x)| \leq L|x - y|$ for all x, y. Then $\mathsf{Rad}_n(g \circ \mathcal{F}, Z^n) \leq L\mathsf{Rad}_n(\mathcal{F}, Z^n)$.*

5. *Suppose $\{\mathcal{F}_i\}_{i \in I}$ satisfies $0 \in \mathcal{F}_i$ for each $i \in I$. Then $\mathsf{Rad}_n(\bigcup_{i \in I} \mathcal{F}, Z^n) \leq \sum_{i \in I} \mathsf{Rad}_n(\mathcal{F}_i, Z^n)$.*

1.3 Two Layer Neural Networks

A two-layer neural network takes an input vector of d variables $x = (x_1, x_2, \ldots, x_d)$ and builds a nonlinear function $f(x)$ to predict the response $y \in \mathbb{R}^D$. What distinguishes neural networks from other nonlinear methods is the particular structure of the model:

$$f(x) = f_\theta(x) = g\left(\beta_0 + \sum_{j=1}^{m} \beta_j \sigma(b_j + w_j^\top x)\right),$$

where $x \in \mathbb{R}^d, b_j \in \mathbb{R}, w_j \in \mathbb{R}^d, \beta_0 \in \mathbb{R}^D, \beta_j \in \mathbb{R}^D$. See Figure 1.

- $\theta = \{[\beta, a_j, b_j, w_j] : j = 1, \ldots, m\}$ denotes the set of model parameters.
- $x_1, \ldots, x_d$ together is called an input layer.
- $A_j \coloneqq \sigma_j(x) = \sigma(b_j + w_j^\top x)$ is called an activation.
- $A_1, \ldots, A_m$ together is called a hidden layer or hidden unit; m is the number of hidden nodes.
- $f(x)$ is called an output layer.

- g is an output function. Examples are:
  - softmax $g_i(x) = \exp(x_i)/\sum_{l=1}^{D} \exp(x_l)$ for classification. The softmax function estimates the conditional probability $g_i(x) = P(y = i|x)$.

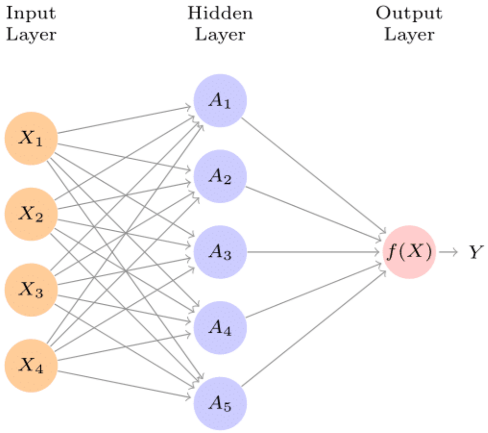

Figure 1: Neural network with a single hidden layer. The hidden layer computes activations $A_j = \sigma_j(x)$ that are nonlinear transformations of linear combinations of the inputs $x_1, \ldots, x_d$. Hence these A_j are not directly observed. The functions σ_j are not fixed in advance, but are learned during the training of the network. The output layer is a linear model that uses these activations A_j as inputs, resulting in a function $f(x)$. Figure 10.1 from [3].

  - identity/linear $g(x) = x$ for regression.
  - threshold $g_i(x) = I(x_i > 0)$

- σ is called an activation function. Examples are:
  - sigmoid $\sigma(x) = 1/(1 + e^{-x})$ (see Figure 2)
  - rectified linear (ReLU) $\sigma(x) = \max\{0, x\}$ (see Figure 2)
  - identity/linear $\sigma(x) = x$
  - threshold $\sigma(x) = I(x > 0)$, threshold gives a direct multi-layer extension of the perceptron (as considered by Rosenblatt).

Activation functions in hidden layers are typically nonlinear, otherwise the model collapses to a linear model. So the activations are like derived features - nonlinear transformations of linear combinations of the features.

1.4 Multi Layer Neural Networks

Modern neural networks typically have more than one hidden layer, and often many units per layer. In theory a single hidden layer with a large number of units has the ability to approximate most functions. However, the learning task of discovering a good solution is made much easier with multiple layers each of modest size.

A deep neural network refers to the model allowing to have more than 1 hidden layers: given input $x \in \mathbb{R}^d$ and response $y \in \mathbb{R}^D$, to predict the response y. K-layer fully connected deep neural network is to build a nonlinear function $f(x)$ as

- Let $m^{(0)} = d$ and $m^{(K)} = D$
- Define recursively

$$\begin{aligned} x^{(0)} &= x, \quad (x \in \mathbb{R}^{m^{(0)}}), \\ x_j^{(k)} &= \sigma(b_j^{(k)} + (w_j^{(k)})^\top x^{(k-1)}), \quad w_j^{(k)}, x^{(k-1)} \in \mathbb{R}^{m^{(k-1)}}, b_j \in \mathbb{R}^{m^{(k)}}, \qquad k = 1, \ldots, K. \\ f(x) &= g(x^{(K)}). \end{aligned}$$

- $\theta = \{[b_j^{(k)}, w_j^{(k)}] : k = 1, \ldots, K, \, j = 1, \ldots, m^{(k)}\}$ denotes the set of model parameters.
- $m^{(k)}$ is the number of hidden units at layer k.

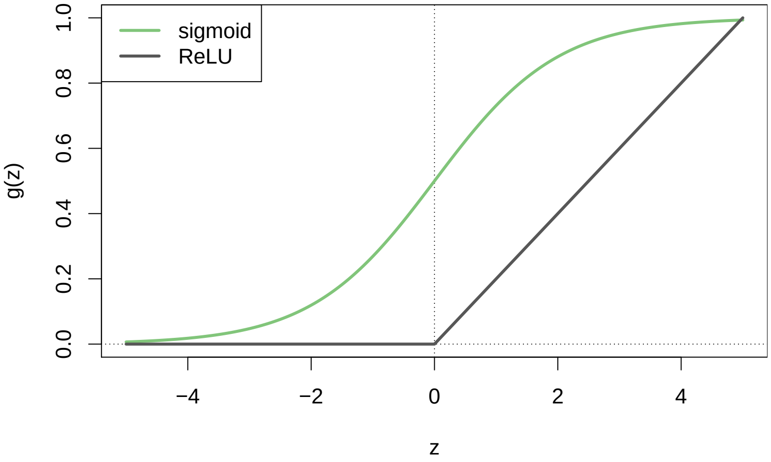

Figure 2: Activation functions. The piecewise-linear ReLU function is popular for its efficiency and computability. We have scaled it down by a factor of five for ease of comparison. Figure 10.2 from [3].

2 Notation and Goal

From here, we only consider regression problem, so $g(x) = x$. We assume $\beta_0 = 0$ and $b_j = 0$.

For the two-layer neural network with the width of the hidden layer m and activation function σ, the function space we consider is

$$\mathcal{F}_{m,\sigma} = \left\{ f_\theta : f_\theta(x) = \sum_{j=1}^{m} \beta_j \sigma(w_j^\top x) \right\},$$

and if we consider all two-layer neural network with arbitrary width, then

$$\mathcal{F}_\sigma = \bigcup_{m=1}^{\infty} \mathcal{F}_{m,\sigma} = \left\{ f_\theta : f_\theta(x) = \sum_{j=1}^{m} \beta_j \sigma(w_j^\top x),\ m \in \mathbb{N} \right\}.$$

For the multi-layer neural network, for $k = 1, \ldots, K$, write $W_k \in \mathbb{R}^{m^{(k)} \times m^{(k-1)}}$ as i-th row of W_k is $(w_j^{(k)})^\top$, i.e.,

$$W_k = \begin{pmatrix} (w_1^{(k)})^\top \\ \vdots \\ (w_{m^{(k)}}^{(k)})^\top \end{pmatrix} \in \mathbb{R}^{m^{(k)} \times m^{(k-1)}},$$

And for $k = 1, \ldots, K-1$, write $\sigma_k : \mathbb{R}^{m^{(k)}} \to \mathbb{R}^{m^{(k)}}$ be coordinatewise application of σ, i.e., $\sigma_k(x_1, \ldots, x_{m^{(k)}}) = (\sigma(x_1), \ldots, \sigma(x_{m^{(k)}}))$, and let $\sigma_K \coloneqq g$. Now, assume $b_j^{(k)} = 0$ for $k = 1, \ldots, K$, and $g = \mathrm{id}$. Then K-layer neural network can be described as

$$f_\theta(x) = \sigma_K(W_K \sigma_{K-1}(W_{K-1} \cdots \sigma_1(W_1 x) \cdots)).$$

Or inductively,

$$f_\theta^{(0)}(x) = x, \qquad f_\theta^{(k)}(x) = \sigma_k(W_k f_\theta^{(k-1)}(x)), \qquad f_\theta(x) = f_\theta^K(x).$$

We impose the condition that $\|W_k\| \le B$ for each k, where $\|\cdot\|$ is an appropriate matrix norm. Hence, for K-layer neural network, the function space we consider is $\mathcal{F}_\sigma^{(K)}$, with $\mathcal{F}_\sigma^{(0)} = \{\mathrm{id}\}$ and

$$\mathcal{F}_\sigma^{(k)} = \left\{ f_\theta^{(k)} : f_\theta^{(k)}(x) = \sigma_k(W_k f_\theta^{(k-1)}(x)),\ f_\theta^{(k-1)} \in \mathcal{F}_\sigma^{(k-1)}, \|W_k\| \le B \right\}.$$

Suppose the true regression function f_* is in a function class $\mathcal{M}$, so

$$y \approx f_*(x), \qquad f_* \in \mathcal{M}.$$

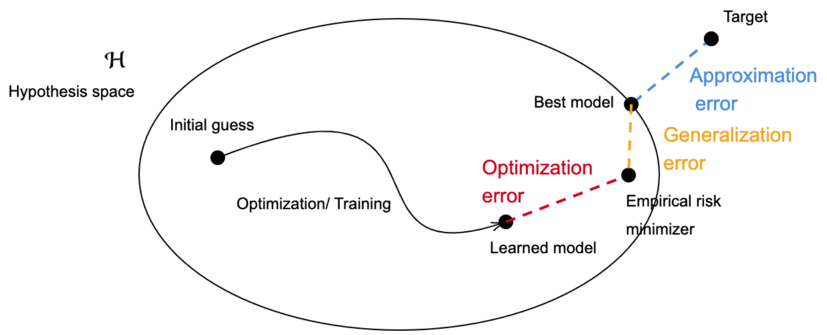

Figure 3: Diagram representing the learning procedure, the three main paradigms and their corresponding errors. Figure 2 from https://dcn.nat.fau.eu/breaking-the-curse-of-dimensionality-with-barron-spaces/.

Suppose are using the ℓ_2-loss, so we find f among deep neural network class $\mathcal{F}$ that minimizes the expected risk (평균위험),

$$f^0 = \arg\min_{f\in\mathcal{F}} \mathbb{E}_{(Y,X)\sim P}\left[(y-f(x))^2\right].$$

f_0 is the expected risk mimizing function (평균위험최소함수). And we estimate f^0 by $\hat{f}$ using data by minimizes on the empirical risk (경험위험) on training dataset, so

$$\hat{f} = \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2.$$

$\hat{f}$ is the empirical risk mimizing function (경험위험최소함수). And we set $\tilde{f}$ be the approximation of $\hat{f}$ by optimization(최적화); $\tilde{f}$ is the learned function (학습된 함수).

So there are three sources of errors: approximation error, generalization error, and optimization error. See Figure 3.

$$f_* - \tilde{f} = \underbrace{f_* - f^0}_{\text{approximation error}} + \underbrace{f^0 - \hat{f}}_{\text{generalization error}} + \underbrace{\hat{f} - \tilde{f}}_{\text{optimization error}}.$$

We focus on approximation error and generalization error. What we would like to achieve is that:

For the approximation error: we would like to control $\left\|f_* - f^0\right\|_{L^2(P)}$ appropriately in terms of the width of the neural network m. Ideally, we would like to restrict the function class $\mathcal{M}$ where f_*comes from, and define an appropriate norm $\|f_*\|_*$, so that

$$\inf_{f\in\mathcal{F}_{m,h}} \left\|f_* - f^0\right\|_{L^2(P)}^2 \lesssim \frac{\|f_*\|_*^2}{m}.$$

For the generalization error: we have seen from the concentration lecture note that with probability at least $1-\delta$,

$$\sup_{f\in\mathcal{F}_{m,h}} \left|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \mathbb{E}[f]\right| \leq 2\mathrm{Rad}(\mathcal{F}_{m,\sigma}) + \sqrt{\frac{1}{2n}\log\left(\frac{2}{\delta}\right)}.$$

Hence we would like to see that, with appropriate norm $\|f\|_{**}$ for $f\in\mathcal{F}_{m,\sigma}$, define $\mathcal{F}_{m,\sigma,Q} := \{f\in\mathcal{F}_{m,\sigma} : \|f\|_{**} \leq Q\}$, and then

$$\mathrm{Rad}(\mathcal{F}_{m,\sigma,Q}) \lesssim \frac{Q}{\sqrt{n}}.$$

If both holds, then

$$\left\|f_* - \hat{f}\right\|_{L^2(P)}^2 = O_P\left(\frac{\|f_*\|_*^2}{m} + \frac{Q}{\sqrt{n}}\right).$$

3 Generalization error: two layer network

We now compute a bound for the Rademacher complexity of two-layer neural networks.

Theorem 2. *For some constants $B_w > 0$ and $B_\beta > 0$, let*

$$\mathcal{F}_{m,\sigma,B} = \left\{ f_\theta \in \mathcal{F}_{m,\sigma} : \|\beta\|_2 \le B_\beta, \|w_j\|_2 \le B_w, j = 1, \dots, m \right\},$$

and suppose $\|Z_i\|_2 \le C$ for all $i = 1, \dots, n$. Let σ be 1-Lipschitz. Then,

$$\mathrm{Rad}(\mathcal{F}_{m,\sigma,B}; Z^n) \le 2 B_\beta B_u C \sqrt{\frac{m}{n}}.$$

This bound is not ideal as it depends on the number of neurons m. Empirically, it has been found that the generalization error does not increase monotonically with m. As more neurons are added to the model, thereby giving it more expressive power, studies have shown that generalization is improved [Belkin et al., 2019]. This contradicts the bound above, which states that more neurons leads to worse generalization.

Next, we look at a finer bound that results from dening a new complexity measure. A recurring theme in subsequent proofs will be the functional invariance of two-layer neural networks under a class of rescaling transformations. The key ingredient will be the positive homogeneity of the ReLU function, i.e.,

$$\alpha\sigma(x) = \sigma(\alpha x), \qquad \forall \alpha > 0.$$

This implies that for any $\lambda_i > 0$ $(i = 1, \dots, m)$, the transformation $\theta = \{(\beta_j, w_j)\}_{1 \le j \le m} \mapsto \theta' = \{(\lambda_j \beta_j, w_j/\lambda_j)\}_{1 \le j \le m}$ has no net effect on the neural network's functionality (i.e., $f_\theta = f_{\theta'}$) since

$$\beta_j \cdot \phi\left(w_j^\top x\right) = (\lambda_j \beta_j) \cdot \phi\left(\left(\frac{w_j}{\lambda_j}\right)^\top x\right).$$

In light of this, we devise a new complexity measure $\|\cdot\|_1$ that is also invariant under such transformations and use it to prove a better bound for the Rademacher complexity. For $f_\theta \in \mathcal{F}_{m,\sigma}$, we can write $f_\theta(x) = \sum_{j=1}^m \beta_j \sigma(w_j^\top x)$. Define a complexity of θ as

$$C(\theta) := \sum_{j=1}^m |\beta_j| \, \|w_j\|_2 .$$

Theorem 3. *For some constant $B > 0$ consider the function class*

$$\mathcal{F}_{m,\sigma,B} = \{ f_\theta \in \mathcal{F}_{m,\sigma} : C(\theta) \le B \}. \tag{1}$$

If $\|Z_i\|_2 \le C$ for all $i = 1, \dots, n$. Let σ be ReLU function, then

$$\mathrm{Rad}(\mathcal{F}_{m,\sigma,B}; Z^n) \le \frac{2BC}{\sqrt{n}}.$$

Remark 4. Compared to Theorem 2, this bound does not explicitly depend on the number of neurons m. Thus, it is possible to use more neurons and still maintain a tight bound if the value of the new complexity measure $\|\theta\|_1$ is reasonable. In contrast, the bound of Theorem 2 explicitly grows with the total number of neurons.

Moreover, Theorem 3 is stronger as we have more neurons - this is because the function class $\mathcal{F}_{m,\sigma,B}$ as defined in 1 is bigger as m increases. Because of this, it's possible to obtain a generalization guarantee that decreases as m increases, as we will see later.

Proof. Due to the positive homogeneity of the ReLU function , it will be useful to define the ℓ_2-normalized weight vector $\bar{w}_j := w_j/\|w_j\|_2$ so that $\phi(w_j^\top x) = \|w_j\|_2 \phi(\bar{w}_j^\top x)$. Let $\xi_i = \pm 1$ being i.i.d. with probability 1/2 be Rademacher variables, then the empirical Rademacher complexity satisfies

$$\begin{aligned}
\mathrm{Rad}(\mathcal{F}_{m,\sigma,B}; Z^n) &= \mathbb{E}_\xi \left[\sup_{f_\theta \in \mathcal{F}_{m,\sigma,B}} \frac{1}{n} \sum_{i=1}^n \xi_i f_\theta(Z_i) \middle| Z \right] \\
&= \mathbb{E}_\xi \left[\sup_{\theta : C(\theta) \le B} \frac{1}{n} \sum_{i=1}^n \xi_i \sum_{j=1}^m \beta_j \sigma\left(w_j^\top Z_i\right) \middle| Z \right] \\
&= \frac{1}{n} \mathbb{E}_\xi \left[\sup_{\theta : C(\theta) \le B} \sum_{i=1}^n \xi_i \sum_{j=1}^m \beta_j \left\|w_j\right\|_2 \sigma\left(\bar{w}_j^\top Z_i\right) \middle| Z \right] \quad \text{(by positive homogeneity of } \sigma) \\
&= \frac{1}{n} \mathbb{E}_\xi \left[\sup_{\theta : C(\theta) \le B} \sum_{j=1}^m \beta_j \left\|w_j\right\|_2 \sum_{i=1}^n \xi_i \sigma\left(\bar{w}_j^\top Z_i\right) \middle| Z \right] \\
&\le \frac{1}{n} \mathbb{E}_\xi \left[\sup_{\theta : C(\theta) \le B} \sum_{j=1}^m \beta_j \left\|w_j\right\|_2 \max_{1 \le k \le m} \left| \sum_{i=1}^n \xi_i \sigma\left(\bar{w}_k^\top Z_i\right) \right| \middle| Z \right],
\end{aligned}$$

since $\sum_j \alpha_j \beta_j \le \sum_j |\alpha_j| \max_k |\beta_k|$. Then from $C(\theta) \le B$, we can further bound as

$$\begin{aligned}
\mathrm{Rad}(\mathcal{F}_{m,\sigma,B}; Z^n) &\le \frac{B}{n} \mathbb{E}_\xi \left[\sup_{\theta : C(\theta) \le B} \max_{1 \le k \le m} \left| \sum_{i=1}^n \xi_i \sigma\left(\bar{w}_k^\top Z_i\right) \right| \middle| Z \right] \\
&= \frac{B}{n} \mathbb{E}_\xi \left[\sup_{\bar{w} : \|\bar{w}\|_2 = 1} \left| \sum_{i=1}^n \xi_i \sigma\left(\bar{w}^\top Z_i\right) \right| \middle| Z \right] \\
&\le \frac{B}{n} \mathbb{E}_\xi \left[\sup_{\bar{w} : \|\bar{w}\|_2 \le 1} \left| \sum_{i=1}^n \xi_i \sigma\left(\bar{w}^\top Z_i\right) \right| \middle| Z \right] \\
&\le \frac{2B}{n} \mathbb{E}_\xi \left[\sup_{\bar{w} : \|\bar{w}\|_2 \le 1} \sum_{i=1}^n \xi_i \sigma\left(\bar{w}^\top Z_i\right) \middle| Z \right] \\
&= 2B \mathrm{Rad}(\mathcal{H}; Z^n),
\end{aligned}$$

where the last inequality is from Lemma 1, and $\mathcal{H} = \left\{ x \mapsto \sigma(\bar{w}^\top x) : \bar{w} \in \mathbb{R}^d, \|\bar{w}\|_2 \le 1 \right\}$. Since the ReLU function σ is 1-Lipschitz, $\mathrm{Rad}(\mathcal{H}; Z^n) \le \mathrm{Rad}(\mathcal{H}'; Z^n)$, where $\mathcal{H}' = \left\{ x \mapsto \bar{w}^\top x : \bar{w} \in \mathbb{R}^d, \|\bar{w}\|_2 \le 1 \right\}$. Then $\mathrm{Rad}(\mathcal{H}'; Z^n) \le \frac{C}{\sqrt{n}}$ from below concludes the proof. □

Proposition 5. *Let $\mathcal{F} = \left\{ x \mapsto \langle w, x \rangle : w \in \mathbb{R}^d, \|w\|_2 \le B \right\}$ for some constant $B > 0$, and suppose $\|Z_i\|_2 \le C$ for all $i = 1, \ldots, n$. Then*

$$\mathrm{Rad}(\mathcal{F}; Z^n) \le \frac{BC}{\sqrt{n}}.$$

Proof. Left as HW. □

Then as suggested in the concentration lecture not, with probability at least $1 - \delta$,

$$\begin{aligned}
\sup_{f \in \mathcal{F}_{m,\sigma,B}} \left| \frac{1}{n} \sum_{i=1}^n f(x_i) - \mathbb{E}[f] \right| &\le 2\mathrm{Rad}(\mathcal{F}_{m,\sigma,B}) + \sqrt{\frac{1}{2n} \log\left(\frac{2}{\delta}\right)} \\
&\le \frac{4BC}{\sqrt{n}} + \sqrt{\frac{1}{2n} \log\left(\frac{2}{\delta}\right)}.
\end{aligned}$$

There is a direct result with Barron class as well.

Theorem ([2, Theorem 15])**.** *Let $\mathcal{F}_{m,\sigma,Q} \coloneqq \{ f \in \mathcal{F}_{m,\sigma} : \|f\|_{\mathcal{B}} \le Q \}$. Then we have*

$$\mathrm{Rad}(\mathcal{F}_{m,\sigma,Q}; Z^n) \le 2Q \sqrt{\frac{2 \log(2d)}{n}}.$$

Instead of minimizing the training error, we can also consider the regularized term. For $f_\theta \in \mathcal{F}_{m,\sigma}$, we can write $f_\theta(x) = \sum_{j=1}^m \beta_j \sigma(w_j^\top x)$. Define the 1-norm of θ as

$$\|\theta\|_1 \coloneqq \frac{1}{m} \sum_{j=1}^m |\beta_j| \, \|w_j\|_1 .$$

And consider the minimization problem

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^n (y_i - f_\theta(x_i))^2 + \lambda \sqrt{\frac{\log(2d)}{n}} \, \|\theta\|_1 ,$$

and let $\hat{\theta}^{(1)}$ be its minimizer.

Theorem ([2, Theorem 16])**.** *Suppose $\mathcal{X} \subset \mathbb{R}^d$ is compact, and assume $f_* : \mathcal{X} \to [0,1]$. There exists some $\lambda_0 > 0$ such that for $\lambda \geq \lambda_0$, with probability $1 - \delta$,*

$$\frac{1}{n} \sum_{i=1}^n (y_i - f_{\hat{\theta}^{(1)}}(x_i))^2 \lesssim \frac{\|f_*\|_{\mathcal{B}}^2}{m} + \lambda \, \|f_*\|_{\mathcal{B}} \sqrt{\frac{\log(2d)}{n}} + \sqrt{\frac{\log(n/\delta)}{n}}.$$

4 Generalization error: multi layer network

We will give one Rademacher complexity bound, obtained by inductively peeling off layers. This will depend on $\left\|W_i^\top\right\|_{1,\infty}$. This bound has a pretty clean proof, and appeared in [1].

4.1 First "layer peeling" proof

We'll prove this with an induction "peeling" off layers. This peeling will use the following lemma, which collects many standard Rademacher properties.

Proof. Let $\mathcal{F}_k$ be the collection of functions computed by each node in kth layer, i.e.,

$$\begin{aligned} \mathcal{F}_k &= \left\{ (f_\theta^{(k)})_j : \mathbb{R}^{m^{(k-1)}} \to \mathbb{R} : f_\theta^{(k)} \in \mathcal{F}_\sigma^{(k)}, j = 1, \ldots, m^{(k)} \right\} \\ &= \left\{ x \mapsto \sigma_k(W_k f_\theta^{(k-1)}(x)), \ f_\theta^{(k-1)} \in \mathcal{F}_\sigma^{(k-1)}, \|W_k\| \leq B \right\}. \end{aligned}$$

Then It'll be shown by induction that

$$\mathsf{Rad}(\mathcal{F}_k, Z) \leq \|Z\|_{2,\infty} (2LB)^k \sqrt{2 \log d}.$$

Base case ($i = 0$):

$$\begin{aligned} \mathsf{Rad}(\mathcal{F}_\sigma^{(k)}, Z) &= \mathsf{Rad}\left(\{x \mapsto x_j, 1 \leq j \leq d\}, Z \right) \\ & \mathbb{E}\left[\sup_{1 \leq j \leq d} \frac{1}{n} \sum_{i=1}^n \sigma_i Z_{ij} \middle| Z^n \right] \\ &\leq \left(\max_{1 \leq j \leq d} \left\|Z^j\right\|_2 \right) \sqrt{2 \log d} \\ &= \|Z\|_{2,\infty} \sqrt{2 \log d}. \end{aligned}$$

Inductive step. Note that

$$\mathcal{F}_k = \left\{ x \mapsto \left(\sigma_k(W_k f_\theta^{(k-1)}(x)) \right)_j , \ f_\theta^{(k-1)} \in \mathcal{F}_\sigma^{(k-1)}, \|W_k\| \leq B \right\}.$$

Now for $a \in \mathbb{R}^{m_{(k-1)}}$ with $\|a\|_1 \leq 1$,

$$a^\top f_\theta^{(k-1)}(x) = \sum_{j=1}^{m_{(k-1)}} a_j (f_\theta^{(k-1)})_j(x),$$

so

$$x \mapsto a^\top f_\theta^{(k-1)}(x) \in \mathrm{conv}(-\mathcal{F}_k \cup \mathcal{F}_k).$$

Now note that, for W_k with $\left\|W_k^\top\right\|_{1,\infty} \leq B$, $\frac{W_k}{B}$ has its each row with 1 norm bounded by 1, so $x \mapsto \frac{1}{B} W_k f_\theta^{(k-1)}(x)$ has its component functions (i.e., $\left(\frac{1}{B} W_k f_\theta^{(k-1)}(x)\right)_j$) in $\mathrm{conv}(-\mathcal{F}_k \cup \mathcal{F}_k)$. And therefore,

$$\mathcal{F}_k = \left\{x \mapsto \sigma(Bg(x)),\ g \in \mathrm{conv}(-\mathcal{F}_{k-1} \cup \mathcal{F}_{k-1})\right\}.$$

Hence by applying Lipschitz peeling,

$$\begin{aligned}\mathsf{Rad}(\mathcal{F}_k, Z) &= \mathsf{Rad}\left(\left\{x \mapsto \sigma(Bg(x)),\ g \in \mathrm{conv}(-\mathcal{F}_{k-1} \cup \mathcal{F}_{k-1})\right\}, Z\right) \\ &\leq LB\mathsf{Rad}\left(-\mathcal{F}_{k-1} \cup \mathcal{F}_{k-1}, Z\right) \\ &\leq (2LB)\mathsf{Rad}\left(\mathcal{F}_{k-1}, Z\right) \\ &\leq (2LB)^k \left\|Z\right\|_{2,\infty} \sqrt{2 \log d}.\end{aligned}$$

□

References

[1] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *J. Mach. Learn. Res.*, 3:463–482, 2002.

[2] Weinan E, Chao Ma, Stephan Wojtowytsch, and Lei Wu. Towards a mathematical understanding of neural network-based machine learning: what we know and what we don't. *CoRR*, abs/2009.10713, 2020.

[3] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning—with applications in R*. Springer Texts in Statistics. Springer, New York, [2021] ©2021. Second edition [of 3100153].