CS229M/STATS214: Machine Learning Theory

Lecturer: Tengyu Ma
Scribe: Rafael Rafailov and Aidan Perreault

Lecture # 9
Feb 10th, 2021

9.1 Review and overview

In the last lecture, we described the covering number approach for Rademacher complexity, and ended off with a statement of Dudley's chaining theorem. In this lecture, we finish up that section of material with some remarks on how to use these covering number bounds. We then move on to a conceptual introduction of deep learning theory. This will be a higher-level discussion of the ideas in deep learning theory so that we have a big picture view of what is happening in the research on deep learning, and will set us up well for the next few lectures where we dive into the details. (We note that there is no universal agreement on the direction that deep learning research should take as we are still very much in the exploratory phase of research, but there are a few general directions that several researchers are pursuing.)

9.2 Covering number and Dudley's theorem

At the end of last lecture we stated Dudley's Theorem, which gives a link between the covering number of a function class and its Rademacher complexity:

Theorem 9.1 (Dudley's Theorem)**.** *If $\mathcal{F}$ is a function class from Z to $\mathbb{R}$, then*

$$R_S(\mathcal{F}) \leq 12 \int_0^\infty \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon. \tag{9.1}$$

In particular, if $\mathcal{F}$ consists of functions bounded in $[-1, 1]$, then, we have that for all $\epsilon > 1, N(\epsilon, \mathcal{F}, L_2(P_n)) = 1$. (To see this choose $\{f \equiv 0\}$, which is a complete cover for $\epsilon > 1$.) Hence, the limits of integration in (9.1) can be truncated to $[0, 1]$:

$$R_S(\mathcal{F}) \leq 12 \int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon, \tag{9.2}$$

since $\log N(\epsilon, \mathcal{F}, L_2(P_n)) = 0$ for $\epsilon > 1$.

9.2.1 Covering number regimes for which Dudley's theorem is finite

Of course, the bound in (9.1) is only useful if the integral on the RHS is finite. Here are some setups where this is the case (we continue to assume that the functions in $\mathcal{F}$ are bounded in $[-1, 1]$):

1. If $N(\epsilon, \mathcal{F}, L_2(P_n)) \approx (1/\epsilon)^R$ (ignoring multiplicative and additive constants), then we have $\log N(\epsilon, \mathcal{F}, L_2(P_n)) \approx R \log(1/\epsilon)$. We can plug this into the RHS of (9.1) to get

$$\int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon = \int_0^1 \sqrt{\frac{R \log(1/\epsilon)}{n}} d\epsilon \approx \sqrt{\frac{R}{n}}. \tag{9.3}$$

2. If the covering number has the form $N(\epsilon, \mathcal{F}, L_2(P_n)) \approx a^{R/\epsilon}$ for some a, then we have $\log N(\epsilon, \mathcal{F}, L_2(P_n)) \approx \frac{R}{\epsilon} \log a$. The bound in (9.1) becomes

$$\int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon \approx \int_0^1 \sqrt{\frac{R}{n\epsilon} \log a}\, d\epsilon \tag{9.4}$$

$$= \sqrt{\frac{R}{n} \log a} \int_0^1 \sqrt{\frac{1}{\epsilon}} d\epsilon \tag{9.5}$$

$$= \widetilde{O}\left(\sqrt{\frac{R}{n}}\right). \tag{9.6}$$

3. If the covering number has the form $N(\epsilon, \mathcal{F}, L_2(P_n)) \approx a^{R/\epsilon^2}$, then $\log N(\epsilon, \mathcal{F}, L_2(P_n)) \approx \frac{R}{\epsilon^2} \log a$. In this case we have:

$$\int_0^1 \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon \approx \sqrt{\frac{R}{n} \log a} \underbrace{\int_0^1 \frac{1}{\epsilon} d\epsilon}_{=\infty} = \infty, \tag{9.7}$$

i.e. the bound in (9.1) is vacuous. This is because of the behavior of $\epsilon \mapsto 1/\epsilon^2$ near 0: the function goes to infinity too quickly for us to upper bound its integral. Fortunately, there is an "improved" version of Dudley's theorem that is applicable here:

Theorem 9.2 (Improved Dudley's Theorem)**.** *If $\mathcal{F}$ is a function class from Z to $\mathbb{R}$, then for any fixed cutoff $\alpha \geq 0$ we have the bound*

$$R_S(\mathcal{F}) \leq 4\alpha + 12 \int_\alpha^\infty \sqrt{\frac{\log N(\epsilon, \mathcal{F}, L_2(P_n))}{n}} d\epsilon. \tag{9.8}$$

The proof of this theorem is similar to the proof of the original Dudley's theorem, except that the iterative covering procedure is stopped at the threshold $\epsilon = \alpha$ at the cost of the extra 4α term above.

Theorem 9.2 allows us to avoid the problematic region around $\epsilon = 0$ in the integral in (9.1). If we let $\alpha = 1/poly(n)$, the bound in (9.8) becomes

$$R_S(\mathcal{F}) \leq \frac{1}{poly(n)} + \frac{\sqrt{R \log a}}{\sqrt{n}} \int_\alpha^1 \frac{1}{\epsilon} d\epsilon \tag{9.9}$$

$$= \frac{1}{poly(n)} + \frac{\sqrt{R \log a}}{\sqrt{n}} \log(1/\alpha) \tag{9.10}$$

$$= \widetilde{O}\left(\sqrt{\frac{R}{n}}\right). \tag{9.11}$$

In summary, we have that $R_S(\mathcal{F}) \leq \widetilde{O}\left(\sqrt{\frac{R}{n}}\right)$ for these three dependencies on ϵ: when $\log N(\epsilon, \mathcal{F}, L_2(P_n)) \approx R \log(1/\epsilon)$, $\frac{R}{\epsilon} \log a$, or $\frac{R}{\epsilon^2} \log a$ for some a. Note that if the dependence

on ϵ is $1/\epsilon^c$ for $c > 2$, then even the improved Dudley's theorem does not help us. This is because the $\log(1/\alpha)$ term above becomes $\alpha^{1-c/2}$, and when $\alpha = 1/poly(n)$, this term leads to a bad dependence on n.

9.2.2 Regimes where we can get covering number bounds

The previous remarks discuss how strong our bounds on covering number need to be in order to get a useful result. Here we mention some situations in which we know how to obtain these covering number bounds:

1. Covering number and corresponding Rademacher complexity bounds for linear models are well-known, but fairly technical (see [Zha02]).

2. Covering numbers interact nicely with composition by Lipschitz functions. If ϕ is a ρ-Lipschitz function, then the following bound holds:

$$N(\epsilon/\rho, \mathcal{F}, L_2(P_n)) \geq N(\epsilon, \phi \circ \mathcal{F}, L_2(P_n)). \tag{9.12}$$

   This result is the analog of Talagrand's lemma for covering numbers. The proof follows easily if one considers ϕ as a change of measure: informally, the Lipschitz condition on ϕ means that a distance of ϵ/ρ in the original space $\mathcal{F}$ can be increased to at most ϵ in the space $\phi \circ \mathcal{F}$.

3. Using these results we can obtain a bound on the Rademacher complexity of a dense neural network. Consider a deep network

$$f(x) = W_r \sigma(W_{r-1} \sigma(\cdots \sigma(W_1 x) \ldots), \tag{9.13}$$

   where W_i are layer-wise weights and σ is an activation function which is 1-Lipschitz. For this setup we have the following Rademacher complexity bound:

$$R_S(\mathcal{F}) \leq \underbrace{\left(\prod_{i=1}^{r} \|W_i\|_{\text{op}} \right)}_{\text{relatively large}} \cdot \underbrace{\left(\sum_{i=1}^{r} \frac{\|W_i^T\|_{2,1}^{2/3}}{\|W_i\|_{\text{op}}^{2/3}} \right)^{3/2}}_{\text{relatively small}}. \tag{9.14}$$

   Here $\|W\|_{\text{op}}$ is the operator norm (or spectral norm) of W, and $\|W_i^T\|_{2,1}$ denotes the sum of the l_2 norms of the rows of W_i. The second term is relatively small as it is a sum of matrix norms, and so the bound is dominated by the first term, which is a product of matrix norms. This first term comes from composition of Lipschitz functions as in (9.12) above, since the Lipschitz constant of a linear operator is its spectral norm. The full details are presented in [BFT17].

9.3 Deep learning theory

We now turn to a high-level overview of deep learning theory. To begin, we outline a framework for classical machine learning theory, then discuss how the situation is different from deep learning theory.

9.3.1 Framework for classical machine learning theory

At the risk of oversimplification, we can divide classical machine learning theory into three parts:

1. **Approximation theory** attempts to answer whether there is any choice of parameters θ that achieves low population error. In other words, is the choice of hypothesis class good enough to approximate the ground truth function? Using notation from earlier in this course, the goal is to upper bound $L(\theta^*) = \min_{\theta \in \Theta} L(\theta)$.

2. **Statistical generalization** focuses on bounding the excess risk $L(\hat{\theta}) - L(\theta^*)$. In Lecture 2 we obtained the following bound:

$$L(\hat{\theta}) - L(\theta^*) \leq \underbrace{L(\hat{\theta}) - \hat{L}(\hat{\theta})}_{\text{generalization error}} + |L(\theta^*) - \hat{L}(\theta^*)|. \tag{9.15}$$

   The first term here is the generalization error, which usually has an upper bound of the form $R(\theta)/\sqrt{n}$, where $R(\theta)$ is some complexity measure.[1] This is a demonstration of *Occam's Razor*: the principle that simple (low-complexity) explanations generalize better.

   This statistical approach allows us to define a regularized loss $\hat{L}_{\text{reg}}(\theta) = \hat{L}(\theta) + \lambda R(\theta)$. Minimizing this loss gives us a solution $\hat{\theta}_\lambda$ which simultaneously has low training error and low complexity, which lets us bound both the training error and the generalization error. To summarize, in the classical setting, we can prove statements of the form

$$\text{If } \hat{\theta}_\lambda \text{ minimizes } \hat{L}_{\text{reg}}, \text{ then } L(\hat{\theta}_\lambda) - L(\theta^*) \text{ is small.} \tag{9.16}$$

3. **Optimization** considers how to obtain the minimizer $\hat{\theta}$ or $\hat{\theta}_\lambda$ computationally. This usually involves convex optimization: if $\hat{L}$ or $\hat{L}_{\text{reg}}$ is convex, then we have a polynomial-time algorithm to find the global minimum.

While there are many tradeoffs to consider between these three components (for example, we may be able to find a loss function for which optimization is easy, but generalization becomes worse), it is still possible to study each area individually, then combine all three to get a result.

9.3.2 Deep learning theory and its differences

The situation is not so simple for deep learning theory. Let us consider how this is the case for each of the three components described for classical machine learning theory

1. **Approximation theory:** Large neural net models are considered to be very expressive. That is, both the population loss $L(\theta^*)$ and the finite sample loss $\hat{L}(\hat{\theta})$ can be made small. In fact, neural networks are *universal approximators*; see for example [Hor91]. This can be a somewhat misleading statement as the definition of universal approximator allows for the size of the network to be impracticably large, but morally it seems to hold true in practice anyway.

[1] In earlier lectures, we defined the complexity of a hypothesis class, not of a specific parameter value. To reconcile these two approaches, think of R as a measure of complexity (such as a norm) that we can then use to define a hypothesis class Θ, i.e. $\Theta = \{\theta' : R(\theta') \leq R(\theta)\}$.

This expressivity is possible because neural networks are usually highly *over-parametrized*: they have many more parameters than samples. It is possible to prove that in this regime, the network can "memorize" the entire dataset and achieve approximately zero training error.

2. **Statistical generalization:** Relatively weak regularization is used in practice. In many cases only weak ℓ_2 regularization is used, i.e.

$$\widehat{L}_{\text{reg}}(\theta) = \hat{L}(\theta) + \lambda\|\theta\|_2^2. \tag{9.17}$$

   The first interesting fact is that this regularized loss does not have a unique (approximate) global minimizer. This is due to overparametrization: there are so many degrees of freedom that there are many approximate global minimizers with approximately the same ℓ_2 norm.

   However, it turns out that these global minimizers are not equally good: many models which achieve zero training error may have very bad test error. Take, for example, using stochastic gradient descent (SGD) to learn a model to classify the dataset CIFAR-10. Consider two instantiations of this: one starting with a large learning rate and slowly decreasing it, and one with a small learning rate throughout. Even though both instantiations result in approximately zero training error, the former leads to much better test performance.

   Therefore, the goal in deep learning theory is not just to find an arbitrary global minimum: we need to find the right global minimum. This contrasts sharply with (9.16) from the classical setting, where achieving a global minimum leads to good guarantees on generalization error. This means that (9.16) is simply not powerful enough to deal with deep learning, because it cannot distinguish between θ's with good test error and bad test error.

3. **Optimization:** The discussion above means that optimization plays a significant role in generalization for deep learning. Different training algorithms have different "implicit biases", causing them to converge to different global minimizers. Understanding the implicit biases of algorithms is thus a central goal of deep learning theory. It is impossible to design a good optimization algorithm without also considering its impact on generalization. In fact, many algorithms for non-convex optimization have been proposed that work well for minimizing training loss, but because their implicit bias is different, they lead to worse test performance and are therefore not too useful.

   Often these implicit biases can be interpreted as encouraging $\hat{\theta}$ to have low complexity in some sense. The deep learning analog of (9.16) is that "low complexity solutions generalize". This means that we end up doing more work on the optimization front in order to understand the implicit bias of our algorithm, and then proving generalization bounds works similarly to the classical setting once we understand how our optimizer finds a low-complexity solution.

To explain the success of deep learning, we will cover three tasks over the next several classes:

1. Prove that our optimization algorithm converges to an approximate global minimum, even though the objective function is non-convex. Our results here will mostly be for simplified models (e.g. linearized neural nets). (We will also show later that this can be accomplished separately from the other tasks using a special optimization setup (the "NTK approach"). However, the generalization of this approach can be poor.)

2. Show that the solution $\hat{\theta}$ has low complexity $R(\hat{\theta}) \leq C$. We can only answer this question for some special cases of models (e.g. logistic regression, matrix factorization) and optimizers (e.g. gradient descent, label noise in SGD, dropout, learning rate).

3. Show that for all θ such that $R(\theta) \leq C$ with $\hat{L}(\theta) \approx 0$, we have $L(\theta)$ is small. That is, we show that low-complexity solutions to the empirical risk problem generalize well. We will be working with more fine-grained complexity measures, and several of the tools we used in classical machine learning can still apply.

Bibliography

[BFT17] Peter Bartlett, Dylan J. Foster, and Matus Telgarsky, *Spectrally-normalized margin bounds for neural networks*, NeurIPS (2017).

[Hor91] Kurt Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural Networks **4** (1991), 251–257.

[Zha02] Tong Zhang, *Covering number bounds of certain regularized linear function classes*, Journal of Machine Learning Research **2** (2002), 527–550.