

18.1 Review and overview

In the last lecture, we completed our discussion of online learning and returned to the topic of algorithmic regularization. We described an instance of algorithmic regularization in the classification setting: gradient descent converges to the direction of the max-margin solution.

In this lecture, we discuss the Lipschitzness of models and why they seem to generalize better than arbitrary networks. To do so, we introduce a refined notion of uniform convergence that is data-dependent, and use it to derive a generalization bound for generalized margin. We end by introducing the *all-layer margin*, a specific instance of generalized margin that captures model Lipschitzness, thus allowing us to use the data-dependent generalization bound.

18.2 Stochasticity in algorithmic regularization

We note that in general, when the loss has multiple global minima, any decisions we make about the optimization algorithm will make a difference. Another important source of algorithmic regularization (possibly the most important) comes from the *stochasticity* in the stochastic gradient descent (SGD) algorithm, where the parameters are optimized by updates of the form

$$\theta_{t+1} = \theta_t - \eta(\nabla \hat{L}(\theta_t) + \xi_t), \quad (18.1)$$

where ξ_t is a random noise term, typically with $\mathbb{E}[\xi_t] = 0$ so the noise will not affect the result too much. The variance of ξ_t can sometimes be time-dependent: for example, ξ_t could be dependent on the parameter θ_t .

In practice, it turns out that larger gradient noise can lead to better generalization performance, as long as the algorithm can optimize under such level of noise. The intuition behind this phenomenon is that SGD converges to a “flat” global minimum, i.e. one with small curvature and small noise covariance. On the other hand, if you have a “sharp” local/global minimum with a large amount of noise, SGD will not converge to it stably. There are a number of works on this topic [HWLM20, BGVV20], but a lot of questions in this space remained to be answered.

18.3 Lipschitzness of models and generalization

It has been found that the Lipschitzness of the model plays an important role in algorithmic regularization. As an illustration, note that the curvature (Hessian) of the loss function, the Lipschitzness of the model, and the noise level in SGD are all closely-related. To give a sense of the connections between them, suppose we have a model $f(x; \theta)$, a single example (x, y) , and a loss

function $L(\theta) = \ell(f(x), y) = (f(x) - y)^2$. In this setup, we have the decomposition

$$\nabla^2 L(\theta) = \underbrace{\frac{\partial^2 \ell}{\partial f^2}}_{\text{scalar}} \cdot \underbrace{\frac{\partial f}{\partial \theta}}_{\mathbb{R}^p} \cdot \underbrace{\frac{\partial f^\top}{\partial \theta}}_{\mathbb{R}^p} + \underbrace{\frac{\partial \ell}{\partial f}}_{\mathbb{R}^{p \times p}} \cdot \underbrace{\frac{\partial^2 f}{\partial \theta^2}}_{\mathbb{R}^{p \times p}}. \quad (18.2)$$

This decomposition is useful because it has been found empirically that the second term is relatively small. This implies that the Hessian is somewhat dominated by the first term. The first term, especially $\frac{\partial f}{\partial \theta}$, relates to the Lipschitzness of the model with respect to the parameter. (There are similar connections between other quantities.)

Our algorithmic choices (e.g. SGD) seem to prefer Lipschitz models¹, which implies that such models generalize better. It remains to answer the question: **Why do Lipschitz models generalize better than arbitrary networks?** We want to theoretically analyze the relationship between Lipschitzness and generalization performance, and derive some generalization bounds w.r.t to the Lipschitzness of the models.

First, we note that the idea of using Lipschitzness to obtain generalization bounds is not new: it is the core of non-parametric statistics. However, such bounds suffer from the “curse of dimensionality”, that is, the sample complexity grows exponentially as the data dimension d . Thus, using only Lipschitzness property is not enough to explain the generalization performance of neural networks: we need the help of parameterization.

Consider a deep neural network $f(x) = \sigma(W_r \sigma(W_{r-1} \cdots \sigma(W_1 x)))$ for binary classification. Recall, [BFT17] showed that:

$$L(\theta) \leq \frac{R_S(\mathcal{F})}{\gamma}, \quad (18.3)$$

where γ is the margin of the model, and $R_S(\mathcal{F})$ is some complexity that satisfies

$$R_S(\mathcal{F}) \leq \underbrace{\left(\prod_{i=1}^r \|W_i\|_{\text{op}} \right)}_{\text{relatively large}} \cdot \underbrace{\left(\sum_{i=1}^r \frac{\|W_i^T\|_{2,1}^{2/3}}{\|W_i\|_{\text{op}}^{2/3}} \right)}_{\text{relatively small}}^{3/2}. \quad (18.4)$$

The first term is essentially the upper bound on the Lipschitzness of the model w.r.t. to the input over the entire space.

The limitation of this bound is that if $\|W_i\|_{\text{op}} > 1$, then it grows exponentially in depth. On the other hand, if $\|W_i\|_{\text{op}} < 1$, then the $f_\theta(x)$ is exponentially small. Thus, it is very hard to make the spectral norm small while keeping the margin large. In the typical case, we have

$$\|W_1 x\| \approx \|W_1\|_{\text{op}} \|x\|, \quad (18.5)$$

$$\|\sigma(W_1 x)\| \approx \frac{1}{\sqrt{2}} \|W_1\|_{\text{op}} \|x\|. \quad (18.6)$$

(The second approximation comes from the heuristic that the ReLU function σ will zero out about half of the entries.) Thus, heuristically the output shrinks by a factor of $\sqrt{2}$ when passing through

¹By this we mean the Lipschitz constant of the model is small. Also, we are not distinguishing between the Lipschitz constant w.r.t to the input and that w.r.t. to the parameter because they are actually related (not covered in the lecture).

each layer. To make the output $f(x) \approx \Theta(1)$, we need $\|W_i\|_{\text{op}} \approx \sqrt{2}$, which makes the generalization bound very large.

The deeper cause to this problem is that $\prod_i \|W_i\|_{\text{op}}$ is a worst-case bound on the Lipschitzness of models, since it is data-independent and assumes that the input spans over the entire space. Thus one way to improve the bound is by replacing “worse-case Lipschitzness” with the Lipschitzness at the data points $x^{(1)}, \dots, x^{(n)}$. This also allows us to estimate the Lipschitzness on the empirical data, and gives us a regularizer roughly in accordance with what SGD prefers.

We want to prove a bound of the form

$$L(w) \leq \text{poly}(\text{Lipschitzness of } f_w \text{ on } x^{(1)}, \dots, x^{(n)}, \text{ norms of } W_i\text{'s}). \quad (18.7)$$

The RHS of (18.7) can be used as an explicit regularizer in model training to improve the generalization performance.

18.3.1 Proving data-dependent generalization bounds

Before we prove a bound in the of form (18.7), we first discuss why classical uniform convergence does not work. Note that the RHS of (18.7) is dependent on random variables $x^{(1)}, \dots, x^{(n)}$. But typical bound of uniform convergence using Rademacher complexity is in the form

$$\forall f \in \mathcal{F}, \quad L(f) \leq \text{comp}(\mathcal{F}, n), \quad (18.8)$$

where comp is some complexity measure, or in the form

$$\forall f \in \mathcal{F}, \quad L(f) \leq \text{comp}(f, n). \quad (18.9)$$

The second bound can be achieved by defining $\mathcal{F}_C = \{f : \text{comp}(f, n) \leq C\}$ first, applying the first type of bound for the class \mathcal{F}_C , then performing a union bound over C . However, this approach does not work for obtaining a bound like the RHS of (18.7) because the the corresponding hypothesis class is

$$\mathcal{F}_C = \left\{ f : \text{comp}(f, \{(x^{(i)}, y^{(i)})\}_{i=1}^n, n) \leq C \right\}. \quad (18.10)$$

There are random variables in the definition of the hypothesis class, which is not allowed for Rademacher complexity. Hence, we cannot leverage such techniques directly.

To tackle this issue, we introduce a refined version of uniform convergence. Suppose we can decompose the complexity measure into the sum of a property related to each data point and the function we care about:

$$\text{comp}\left(f, \{(x^{(i)}, y^{(i)})\}_{i=1}^n, n\right) = \sum_{i=1}^n g((x^{(i)}, y^{(i)}), f). \quad (18.11)$$

We can define the *augmented loss* as

$$\tilde{l}(f) = l(f) \cdot \mathbf{1}(g((x, y), f) \leq C). \quad (18.12)$$

This means that we are changing the loss function to include the data-dependent term. An intuitive example can be found in Figure 18.1, where we have an empirical loss with very bizarre behavior, but only outside the low complexity region that we really care about. The augmented

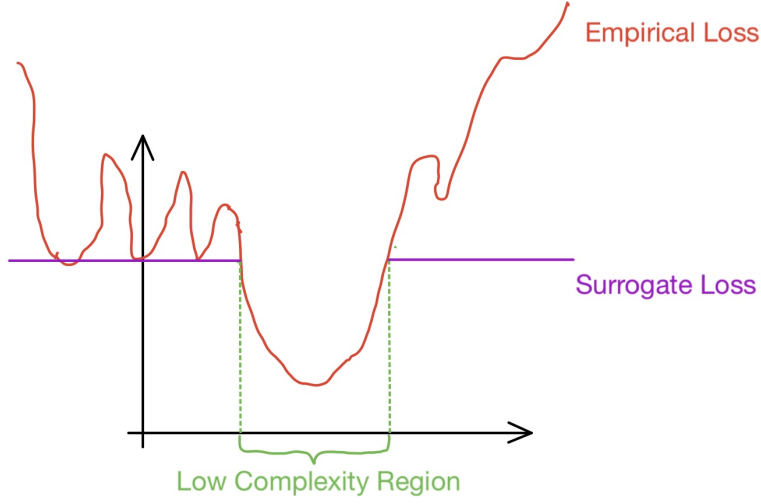


Figure 18.1: The empirical loss has bizarre behavior, but only outside the region of interest. The general idea is to define a surrogate loss and prove uniform convergence over the surrogate loss so as to avoid the bizarre behavior of the empirical loss.

loss notices this and “smooths out” the irregularities outside the low complexity region by ignoring those terms.

The difficulty with taking this approach is that the low complexity region is random: if it was fixed, we could just zoom into that region and prove something directly by uniform convergence. We deal with this difficulty by defining a *surrogate loss* which it could just be constant outside of the low complexity region. We may then apply uniform convergence over the entire space.

We have talked about the notion of surrogate losses before in this class. For example, the margin loss/ramp loss is a type of surrogate loss. There, we thought of using the surrogate loss to make the zero-one loss more continuous. Here, we use the surrogate loss to avoid dealing with the loss function in “bad” regions. Let us define a generalized version of margin loss:

Definition 18.1 (Generalized margin). Let $f : \mathbb{R}^d \mapsto \mathbb{R}$ be a classification model. We call $g_f(x, y)$ a *generalized margin* if $g_f(x, y)$ satisfies

$$g_f(x, y) = \begin{cases} 0 & \text{if } f(x) \cdot y \leq 0 \quad (\text{wrong prediction}), \\ > 0 & \text{if } f(x) \cdot y > 0. \end{cases} \quad (18.13)$$

Given this definition, we have the following lemma:

Lemma 18.2 (Generalization bound for general margin). Suppose g_f is a generalized margin. Let $G = \{g_f : f \in \mathcal{F}\}$, and assume we have an ϵ -covering of G under the $\|\cdot\|_\infty$ metric, $N_\infty(\epsilon, G)$, with $|N_\infty(\epsilon, G)| \leq \lfloor R^2/\epsilon^2 \rfloor$, where R is the Rademacher complexity of the model.

Then with probability larger than $1 - \delta$ over the draw of training data, $\forall f \in \mathcal{F}$ that correctly predicts the labels on the training data, we have

$$L_{0-1}(f) \leq \tilde{O} \left(\frac{R}{\min_i g_f(x^{(i)}, y^{(i)})} \cdot \frac{1}{\sqrt{n}} \right) + \tilde{O} \left(\frac{1}{\sqrt{n}} \right). \quad (18.14)$$

The proof is similar to that for the bound we proved with margin loss earlier in the class, with only a few technical details changed.

18.3.2 All-layer margin

To use Lemma 18.2, we want to design a generalized margin $g_f(x, y)$ such that $G = \{g_f : f \in \mathcal{F}\}$ has low complexity. We want this margin to capture the Lipschitzness of the model so that the bound will not scale badly in the worst case. If we use the standard margin $g_f(x, y) = yf(x)$, then G depends on $\prod_i \|W_i\|_{\text{op}}$; our goal is to do something better than this. To do so, we want to somehow have the margin depend on Lipschitzness.

The *all-layer margin* [WM19] is one such margin. Consider a perturbed model, where $\delta = (\delta_1, \dots, \delta_r)$ is the perturbation and the original neural network model is perturbed in the following way:

$$h_1(x, \delta) = \sigma(W_1 \cdot x) + \delta_1 \|x\|_2, \quad (18.15)$$

$$h_2(x, \delta) = \sigma(W_2 \cdot h_1(x, \delta)) + \delta_2 \|h_1(x, \delta)\|_2, \quad (18.16)$$

\vdots

$$f(x, \delta) = h_r(x, \delta) = \sigma(W_r \cdot h_{r-1}(x, \delta)) + \delta_r \|h_{r-1}(x, \delta)\|_2. \quad (18.17)$$

We can then define the margin of the model as

$$m_f(x, y) \triangleq \min_{\delta} \sqrt{\sum_{i=1}^r \|\delta_i\|^2} \quad \text{s.t.} \quad f(x, \delta)y \leq 0 \quad (\text{incorrect prediction}). \quad (18.18)$$

(It can be proven that m_f is indeed a generalized margin.) Under this definition, $m_f(x, y)$ is large if $f(x)$ is large (i.e. correct) and f is robust to perturbation of example x . The good property is that under this definition, the margin already captures some Lipschitzness of the model. Applying Lemma 18.2 with m_f gives the the following theorem.

Theorem 18.3 (Generalization bound for all-layer margin). *With probability larger than $1 - \delta$ over the draw of training data,*

$$L_{0-1}(f) \leq \tilde{O} \left(\frac{\sum_{i=1}^r \|W_i\|_{1,1}}{\min_i m_f(x^{(i)}, y^{(i)})} \cdot \frac{1}{\sqrt{n}} \right) + \tilde{O} \left(\frac{1}{\sqrt{n}} \right), \quad (18.19)$$

where $\|W\|_{1,1}$ is the sum of the absolute value of entries of W .

This theorem implies that a larger all-layer-margin implies better generalization. To get a larger all-layer-margin, we should make the network more robust to perturbation, i.e. more Lipschitz.

Proof. We present just the main proof ideas here. To use Lemma 18.2, it suffices to show that

$$N_{\infty}(\epsilon, G) \leq O \left(\frac{\sum_{i=1}^r \|W_i\|_{1,1}}{\epsilon^2} \right), \quad (18.20)$$

where $G = \{m_f : f \in \mathcal{F}\}$. Let $\mathcal{F}_1, \dots, \mathcal{F}_r$ be a sequence of hypothesis classes (corresponding to each layer in the network), and let $\mathcal{F} = \{f_r \circ f_{r-1} \circ \dots \circ f_1 : f_i \in \mathcal{F}_i\}$. [WM19] prove the following lemma:

Lemma 18.4 (Decomposition lemma). *Let $m \circ \mathcal{F} = \{m_f : f \in \mathcal{F}\}$ denote the family of all-layer margins of function compositions in \mathcal{F} . Then*

$$\log N_\infty \left(\sqrt{\sum_{i=1}^r \epsilon_i^2}, m \circ \mathcal{F} \right) \leq \sum_i \log N_\infty(\epsilon_i, \mathcal{F}_i).$$

This reduces the problem to bounding the covering number for each layer which is much easier, since each layer is basically a linear transformation plus a non-linearity. \square

18.4 Summary of the course

In the first part of the course, we talked about Rademacher complexity and concentration inequalities. They are classical but important tools in statistical machine learning: any time we have i.i.d. data, we will have to think about concentration bounds. We then spent some time discussing deep learning theory, covering non-convex optimization and algorithmic regularization. Within algorithmic regularization, we talked about the implicit biases of the optimizers, as well as the generalization bounds for deep learning models. We also spent time on online learning theory, a more classical approach for dealing with domain shift and potentially adversarial data.

Bibliography

- [BFT17] Peter Bartlett, Dylan J. Foster, and Matus Telgarsky, *Spectrally-normalized margin bounds for neural networks*, NeurIPS (2017).
- [BGVV20] Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant, *Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process*, Conference on learning theory, PMLR, 2020, pp. 483–513.
- [HWLM20] Jeff Z HaoChen, Colin Wei, Jason D Lee, and Tengyu Ma, *Shape matters: Understanding the implicit bias of the noise covariance*, arXiv preprint arXiv:2006.08680 (2020).
- [WM19] Colin Wei and Tengyu Ma, *Improved sample complexities for deep neural networks and robust classification via an all-layer margin*, International Conference on Learning Representations, 2019.