6

Generalization

Simply put, generalization relates the performance of a model on seen examples to its performance on *unseen* examples. In this chapter, we discuss the interplay between representation, optimization, and generalization, again focusing on models with more parameters than seen data points. We examine the intriguing empirical phenomena related to overparameterization and generalization in today's machine learning practice. We then review available theory—some old and some emerging—to better understand and anticipate what drives generalization performance.

Generalization gap

Recall, the risk of a predictor $f\colon \mathcal{X} \to \mathcal{Y}$ with respect to a loss function $\mathit{loss}\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is defined as

$$R[f] = \mathbb{E}\left[\mathit{loss}(f(X), Y)\right] .$$

Throughout this chapter, it will often be convenient to stretch the notation slightly by using $\mathit{loss}(f, (x, y))$ to denote the loss of a predictor f on an example (x, y). For predictors specified by model parameters w, we'll also write $\mathit{loss}(w, (x, y))$.

For the purposes of this chapter, it makes sense to think of the n samples as an ordered tuple

$$S = ((x_1, y_1), \ldots\ldots, (x_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n .$$

The empirical risk $R_S[f]$ is, as before,

$$R_S[f] = \frac{1}{n} \sum_{i=1}^{n} \mathit{loss}(f(x_i), y_i) .$$

Empirical risk minimization seeks to find a predictor f^* in a specified class $\mathcal{F}$ that minimizes the empirical risk:

$$R_S[f^*] = \min_{f \in \mathcal{F}} R_S[f]$$

In machine learning practice, the empirical risk is often called *training error* or *training loss*, as it corresponds to the loss achieved by some optimization method on the sample. Depending on the optimization problem, we may not be able to find an exact empirical risk minimizer and it may not be unique.

Empirical risk minimization is commonly used as a proxy for minimizing the unknown population risk. But how good is this proxy? Ideally, we would like that the predictor f we find via empirical risk minimization satisfies $R_S[f] \approx R[f]$. However, this may not be the case, since the risk $R[f]$ captures loss on unseen example, while the empirical risk $R_S[f]$ captures loss on seen examples.

Generally, we expect to do much better on seen examples than unseen examples. This performance gap between seen and unseen examples is what we call *generalization gap*.

Definition 1. *Define the* generalization gap *of a predictor f with respect to a dataset S as*

$$\Delta_{\text{gen}}(f) = R[f] - R_S[f]\,.$$

This quantity is sometimes also called *generalization error* or *excess risk*. Recall the following tautological, yet important identity:

$$R[f] = R_S[f] + \Delta_{\text{gen}}(f)$$

What it says is that if we manage to make the empirical risk $R_S[f]$ small through optimization, then all that remains to worry about is generalization gap.

The last chapter provided powerful tools to make optimization succeed. How we can bound the generalization gap is the topic of this chapter. We first take a tour of evidence from machine learning practice for inspiration.

Overparameterization: empirical phenomena

We previously experienced the advantages of overparameterized models in terms of their ability to represent complex functions and our ability to feasily optimize them. The question remains whether they generalize well to unseen data. Perhaps we simply kicked the can down the road. Does the model size that was previously a blessing now come back to haunt us? We will see that not only do large models often generalize well in practice, but often more parameters lead to better generalization performance. Model size does, however, challenge some theoretical analysis. The empirical evidence will orient our theoretical study towards dimension-free bounds that avoid worst-case analysis.

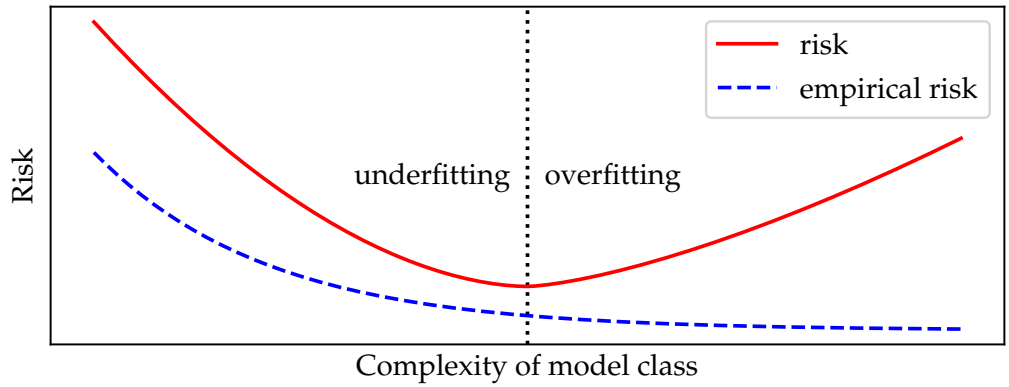

Figure 1: Traditional view of generalization

Effects of model complexity

Think of a model family with an associated measure of complexity, such as number of trainable parameters. Suppose that for each setting of the complexity measure, we can solve the empirical risk minimization problem. We can then plot what happens to risk and empirical risk as we vary model complexity.

A traditional view of generalization posits that as we increase model complexity initially both empirical risk and risk decrease. However, past a certain point the risk begins to increase again, while empirical risk decreases.

The graphic shown in many textbooks is a u-shaped risk curve. The complexity range below the minimum of the curve is called *underfitting*. The range above is called *overfitting*.

This picture is often justified using the bias-variance trade-off, motivated by a least squares regression analysis. However, it does not seem to bear much resemblance to what is observed in practice.

We have already discussed the example of the Perceptron which achieves zero training loss and still generalizes well in theory. Numerous practitioners have observed that other complex models also can simultaneously achieve close to zero training loss and still generalize well. Moreover, in many cases risk continues to decreases as model complexity grows and training data are interpolated exactly down to (nearly) zero training loss. This empirical relationship between overparameterization and risk appears to be robust and manifests in numerous model classes, including overparameterized linear models, ensemble methods, and neural networks.

In the absence of regularization and for certain model families, the empirical relationship between model complexity and risk is more accurately captured by the *double descent* curve in the figure above. There is

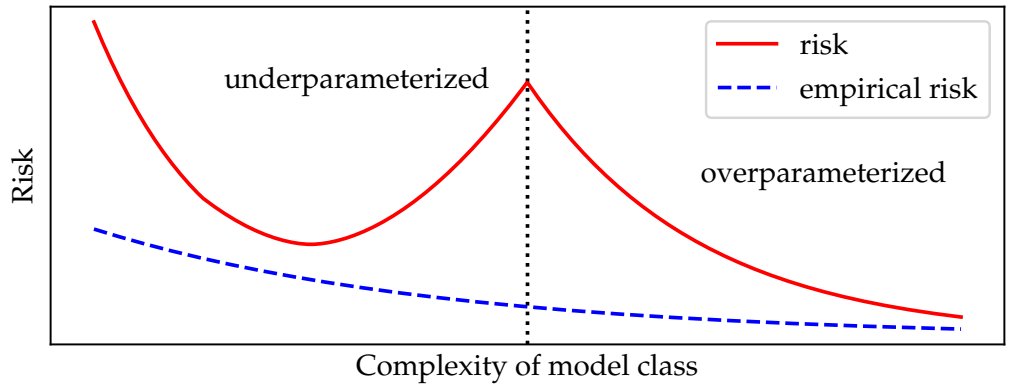

Figure 2: Double descent.

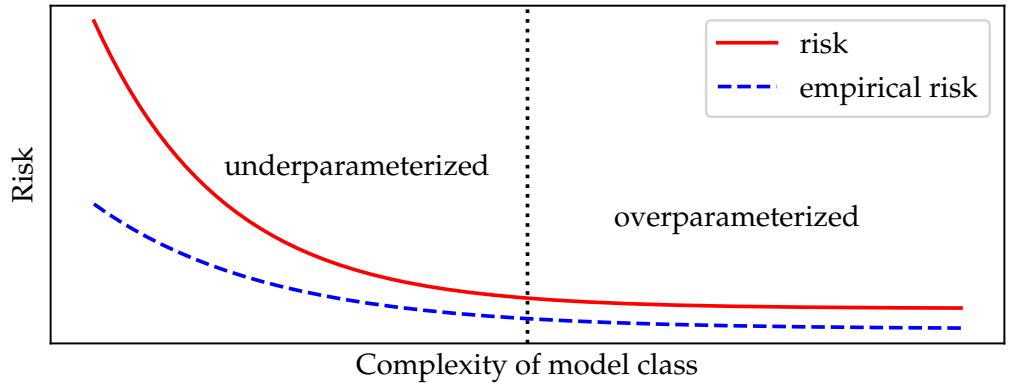

Figure 3: Single descent.

an interpolation threshold at which a model of the given complexity can fit the training data exactly. The complexity range below the threshold is the *underparameterized regime*, while the one above is the overparameterized regime. Increasing model complexity in the overparameterized regime continues to decrease risk indefinitely, albeit at decreasing marginal returns, toward some convergence point.

The double descent curve is not universal. In many cases, in practice we observe a single descent curve throughout the entire complexity range. In other cases, we can see multiple bumps as we increase model complexity.[1] However, the general point remains. There is no evidence that highly overparameterized models do not generalize. Indeed, empirical evidence suggests larger models not only generalize, but that larger models make better out-of-sample predictors than smaller ones.[2,3]

Optimization versus generalization

Training neural networks with stochastic gradient descent, as is commonly done in practice, attempts to solve a non-convex optimization problem. Reasoning about non-convex optimization is known to be difficult. Theoreticians see a worthy goal in trying to prove mathematically that stochastic gradient methods successfully minimize the training objective of large artificial neural networks. The previous chapter discussed some of the progress that has been made toward this goal.

It is widely believed that what makes optimization easy crucially depends on the fact that models in practice have many more parameters than there are training points. While making optimization tractable, overparameterization puts burden on generalization.

We can force a disconnect between optimization and generalization in a simple experiment that we will see next. One consequence is that even if a mathematical proof established the convergence guarantees of stochastic gradient descent for training some class of large neural networks, it would not necessarily on its own tell us much about why the resulting model generalizes well to the test objective.

Indeed, consider the following experiment. Fix training data $(x_1, y_1), \ldots, (x_n, y_n)$ and fix a training algorithm A that achieves zero training loss on these data and achieves good test loss as well.

Now replace all the labels $y_1, \ldots, y_n$ by randomly and independently drawn labels $\tilde{y}_1, \ldots, \tilde{y}_n$. What happens if we run the same algorithm on the training data with noisy labels $(x_1, \tilde{y}_1), \ldots, (x_n, \tilde{y}_n))$?

One thing is clear. If we choose from k discrete classes, we expect the model trained on the random labels to have no more than $1/k$ test accuracy, that is, the accuracy achieved by random guessing. After all, there is no statistical relationship between the training labels and the test labels that the model could learn.

What is more interesting is what happens to optimization. The left panel of the figure shows the outcome of this kind of *randomization test* on the popular CIFAR-10 image classification benchmark for a standard neural network architecture. What we can see is that the training algorithm continues to drive the training loss to zero even if the labels are randomized. The right panel shows that we can vary the amount of randomization to obtain a smooth degradation of the test error. At full randomization, the test error degrades to 90%, as good as guessing one of the 10 classes. The figure shows what happens to a specific model architecture, called Inception, but similar observations hold for most, if not all, overparameterized architectures that have been proposed.

The randomization experiment shows that optimization continues to

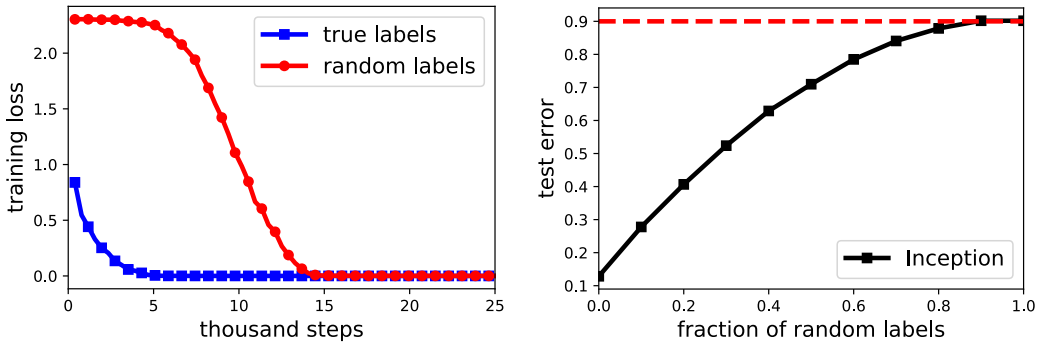

Figure 4: Randomization test on CIFAR-10. Left: How randomization affects training loss. Right: How increasing the fraction of corrupted training labels affects test error.

work well even when generalization performance is no better than random guessing, i.e., 10% accuracy in the case of the CIFAR-10 benchmark that has 10 classes. The optimization method is moreover insensitive to properties of the data, since it works even on random labels. A consequence of this simple experiment is that a proof of convergence for the optimization method may not reveal any insights into the nature of generalization.

The diminished role of explicit regularization

Regularization plays an important role in the theory of convex empirical risk minimization. The most common form of regularization used to be ℓ_2-regularization corresponding to adding a scalar of the squared Euclidean norm of the parameter vector to the objective function.

A more radical form of regularization, called *data augmentation*, is common in the practice of deep learning. Data augmentation transforms each training point repeatedly throughout the training process by some operation, such as a *random crop* of the image. Training on such randomly modified data points is meant to reduce overfitting, since the model never encounters the exact same data point twice.

Regularization continues to be a component of training large neural networks in practice. However, the nature of regularization is not clear. We can see a representative empirical observation in the table below.

Table 1: The training and test accuracy (in percentage) with and without data augmentation and ℓ_2-regularization.

params	random crop	ℓ_2-regularization	train accuracy	test accuracy
1,649,402	yes	yes	100.0	89.05
	yes	no	100.0	89.31
	no	yes	100.0	86.03
	no	no	100.0	85.75

The table shows the performance of a common neural model architecture, called Inception, on the standard CIFAR-10 image classification benchmark. The model has more than 1.5 million trainable parameters, even though there are only $50,000$ training examples spread across 10 classes. The training procedure uses two explicit forms of regularization. One is a form of data augmentation with random crops. The other is ℓ_2-regularization. With both forms of regularization the fully trained model achieves close to 90% test accuracy. But even if we turn both of them off, the model still achieves close to 86% test accuracy (without even readjusting any hyperparameters such as learning rate of the optimizer). At the same time, the model fully interpolates the training data in the sense of making no errors whatsoever on the training data.

These findings suggest that while explicit regularization may help generalization performance, it is by no means necessary for strong generalization of heavily overparameterized models.

Theories of generalization

With these empirical facts in hand, we now turn to mathematical theories that might help explain what we observe in practice and also may guide future empirical and theoretical work. In the remainder of the chapter, we tour several different, seemingly disconnected views of generalization.

We begin with a deep dive into *algorithmic stability*, which posits that generalization arises when models are insensitive to perturbations in the data on which they are trained. We then discuss *VC dimension and Rademacher complexity*, which show how small generalization gaps can arise when we restrict the complexity of models we wish to fit to data. We then turn to *margin bounds* which assert that whenever the data is easily separable, good generalization will occur. Finally we discuss generalization bounds that arise from *optimization*, showing how choice of an algorithmic scheme itself can yield models with desired generalization properties.

In all of these cases, we show that we can recover generalization bounds of the form we saw in the Perceptron: the bounds will decrease with number of data points and increase with "complexity" of the optimal prediction function. Indeed, looking back at the proof of the Perceptron generalization bound, all of the above elements appeared. Our generalization bound arose because we could remove single data points from a set and not change the number of mistakes made by the Perceptron. A large margin assumption was essential to get a small mistake bound. The mistake bound itself was dependent on the iterations of the algorithm. And finally, we related the size of the margin to the scale of the data and optimal separator.

Though starting from different places, we will shows that the four different views of generalization can all arrive at similar results. Each of the aforementioned ingredients can alone lead to generalization, but considerations of all of these aspects help to improve machine learning methods. Generalization is multifaceted and multiple perspectives are useful when designing data-driven predictive systems.

Before diving into these four different views, we first take a quick pause to consider how we hope generalization bounds might look.

How should we expect the gap to scale?

Before we turn to analyzing generalization gaps, it's worth first considering how we should expect them to scale. That is, what is the relationship between the expected size of Δ_{gen} and the number of observations, n?

First, note that we showed that for a *fixed* prediction function f, the expectation of the empirical risk is equal to the population risk. That is, the empirical risk of a single function is a sample average of the population risk of that function. As we discussed in Chapter 3, i.i.d. sample averages should *generalize* and approximate the average at the population level. Here, we now turn to describing *how* they might be expected to scale under different assumptions.

Quantitative central limit theorems

The *central limit theorem* formalizes how sample averages estimate their expectations: If Z is a random variable with bounded variance then $\widehat{\mu}_Z^{(n)}$ converges in distribution to a Gaussian random variable with mean zero and variance on the order of $1/n$.

The following inequalities are useful *quantitative* forms of the central limit theorem. They precisely measure how close the sample average will be to the population average using limited information about the random quantity.

- **Markov's inequality:** Let Z be a nonnegative random variable. Then,

$$\mathbb{P}[Z \geq t] \leq \frac{\mathbb{E}[Z]}{t}.$$

This can be proven using the inequality $I_{[Z \geq t]}(z) \leq \frac{z}{t}$.

- **Chebyshev's inequality:** Suppose Z is a random variable with mean μ_Z and variance σ_Z^2. Then,

$$\mathbb{P}[Z \geq t + \mu_Z] \leq \frac{\sigma_Z^2}{t^2}$$

Chebyshev's inequality helps us understand why sample averages are good estimates of the mean. Suppose that $X_1, \ldots, X_n$ are independent samples we were considering above. Let $\widehat{\mu}$ denote the sample mean $\frac{1}{n}\sum_{i=1}^n Z_i$. Chebyshev's inequality implies

$$\mathbb{P}[\widehat{\mu} \geq t + \mu_X] \leq \frac{\sigma_X^2}{nt^2},$$

which tends to zero as n grows. A popular form of this inequality sets $t = \mu_X$ which gives

$$\mathbb{P}[\widehat{\mu} \geq 2\mu_X] \leq \frac{\sigma_X^2}{n\mu_X^2}.$$

- **Hoeffding's inequality:** Let $Z_1, Z_2, \ldots, Z_n$ be independent random variables, each taking values in the interval $[a_i, b_i]$.Let $\widehat{\mu}$ denote the sample mean $\frac{1}{n}\sum_{i=1}^n Z_i$. Then

$$\mathbb{P}[\widehat{\mu} \geq \mu_Z + t] \leq \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

An important special case is when the Z_i are identically distributed copies of Z and take values in $[0, 1]$. Then we have

$$\mathbb{P}[\widehat{\mu} \geq \mu_Z + t] \leq \exp\left(-2nt^2\right).$$

This shows that when random variables are bounded, sample averages concentrate around their mean value exponentially quickly. If we invoke this bound with $t = C/\sqrt{n}$, the point at which it gives non-trivial results, we have an error of $O(1/\sqrt{n})$ with exponentially high probability. We will see shortly that this relationship between error and number of samples is ubiquitous in generalization theory.

These powerful *concentration inequalities* let us precisely quantify how close the sample average will be to the population average. For instance, we know a person's height is a positive number and that there are no people who are taller than nine feet. With these two facts, Hoeffding's inequality tells us that if we sample the heights of thirty thousand individuals, our sample average will be within an inch of the true average height with probability at least 83%. This assertion is true no matter how large the population of individuals. The required sample size is dictated only by the variability of height, not by the number of total individuals.

You could replace "height" in this example with almost any attribute that you are able to measure well. The quantitative central limits tell us that for attributes with reasonable variability, a uniform sample from a general population will give a high quality estimate of the average value.

"Reasonable variability" of a random variable is necessary for quantitative central limit theorems to hold. When random variables have low variance or are tightly bounded, small experiments quickly reveal insights about the population. When variances are large or effectively unbounded, the number of samples required for high precision estimates might be impractical and our estimators and algorithms and predictions may need to be rethought.

Bounding generalization gaps for individual predictors

Let us now return to generalization of prediction, considering the example where the quantity of interest is the prediction error on individuals in a population. There are effectively two scaling regimes of interest in generalization theory. In one case when the empirical risk is large, we expect the generalization gap to decrease inversely proportional to $\sqrt{n}$. When the empirical risk is expected to be very small, on the other hand, we tend to see the generalization gap decrease inversely proportional to n.

Why we see these two regimes is illustrated by studying the case of a *single* prediction function f, chosen *independently* of the sample S. Our ultimate goal is to reason about the generalization gap of predictors chosen by an algorithm running on our data. The analysis we walk through next doesn't apply to data-dependent predictors directly, but it nonetheless provides helpful intuition about what bounds we can hope to get.

For a fixed function f, the zero-one loss on a single randomly chosen data point is a Bernoulli random variable, equal to 1 with probability p and $1 - p$ otherwise. The empirical risk $R_S[f]$ is the sample average of this random variable and the risk $R[f]$ is its expectation. To estimate the generalization gap, we can apply Hoeffding's inequality to find

$$\mathbb{P}[R[f] - R_S[f] \geq \epsilon] \leq \exp\left(-2n\epsilon^2\right) .$$

Hence, we will have with probability $1 - \delta$ on our sample that

$$|\Delta_{\text{gen}}(f)| \leq \sqrt{\frac{\log(1/\delta)}{2n}} .$$

That is, the generalization gap goes to zero at a rate of $1/\sqrt{n}$.

In the regime where we observe no empirical mistakes, a more refined analysis can be applied. Suppose that $R[f] > \epsilon$. Then the probability that we observe $R_S[f] = 0$ cannot exceed

$$\begin{aligned}\mathbb{P}[\forall i\colon \operatorname{sign}(f(x_i)) = y_i] &= \prod_{i=1}^{n} \mathbb{P}[\operatorname{sign}(f(x_i)) = y_i] \\ &\leq (1-\epsilon)^n \leq e^{-\epsilon n} .\end{aligned}$$

Hence, with probability $1 - \delta$,

$$|\Delta_{\text{gen}}(f)| \leq \frac{\log(1/\delta)}{n} ,$$

which is the $1/n$ regime. These two rates are precisely what we observe in the more complex regime of generalization bounds in machine learning. The main trouble and difficulty in computing bounds on the generalization gap is that our prediction function f depends on the data, making the above analysis inapplicable.

In this chapter, we will focus mostly on $1/\sqrt{n}$ rates. These rates are more general as they make no assumptions about the expected empirical risk. With a few notable exceptions, the derivation of $1/\sqrt{n}$ rates tends to be easier than the $1/n$ counterparts. However, we note that every one of our approaches to generalization bounds have analyses for the "low empirical risk" or "large margin" regimes. We provide references at the end of this chapter to these more refined analyses.

Algorithmic stability

We will first see a tight characterization in terms of an algorithmic robustness property we call *algorithmic stability*. Intuitively, algorithmic stability measures how sensitive an algorithm is to changes in a single training example. Whenever a model is insensitive to such perturbations, the generalization gap will be small. Stability gives us a powerful and intuitive way of reasoning about generalization.

There are a variety of different notions of perturbation. We could consider resampling a single data point and look at how much a model changes. We could also leave one data point out and see how much the model changes. This was the heart of our Perceptron generalization argument. More aggressively, we could study what happens when a single data point is arbitrarily corrupted. All three of these approaches yield similar generalization bounds, though it is often easier to work with one than the others. To simplify the exposition, we choose to focus on only one notion (resampling) here.

To introduce the idea of stability, we first condense our notation to make the presentation a bit less cumbersome. Recall that we operate on tuples of n labeled examples,

$$S = ((x_1, y_1), \ldots\ldots, (x_n, y_n)) \in (\mathcal{X} \times \mathcal{Y})^n .$$

We denote a labeled example as $z = (x, y)$. We will overload our notation and denote the loss accrued by a prediction function f on a data point z as $loss(f, z)$. That is, $loss(f, z) = loss(f(x), y)$. We use the uppercase letters when a labeled example Z is randomly drawn from a population (X, Y).

With this notation in hand, let's now consider two independent random samples $S = (Z_1, \ldots, Z_n)$ and $S' = (Z'_1, \ldots, Z'_n)$, each drawn independently and identically from a population (X, Y). We call the second sample S' a *ghost sample* as it is solely an analytical device. We never actually collect this second sample or run any algorithm on it.

We introduce *n hybrid samples* $S^{(i)}$, for $i \in \{1, \ldots, n\}$ as

$$S^{(i)} = (Z_1, \ldots, Z_{i-1}, Z'_i, Z_{i+1}, \ldots, Z_n) ,$$

where the i-th example comes from S', while all others come from S.

We can now introduce a data-dependent notion of average stability of an algorithm. For this definition, we think of an algorithm as a deterministic map A that takes a training sample in $(\mathcal{X} \times \mathcal{Y})^n$ to some prediction function in a function space Ω. That is $A(S)$ denotes the function from $\mathcal{X}$ to $\mathcal{Y}$ that is returned by our algorithm when run on the sample S.

Definition 2. *The* average stability *of an algorithm* $A \colon (\mathcal{X} \times \mathcal{Y})^n \to \Omega$ *is*

$$\Delta(A) = \mathop{\mathbb{E}}_{S,S'} \left[\frac{1}{n} \sum_{i=1}^{n} \Big(loss(A(S), Z'_i) - loss(A(S^{(i)}), Z'_i) \Big) \right] .$$

There are two useful ways to parse this definition. The first is to interpret average stability as the average sensitivity of the algorithm to a change in a single example. Since we don't know which of its n input samples the

algorithm may be sensitive to, we test all of them and average out the results.

Second, from the perspective of $A(S)$, the example Z_i' is *unseen*, since it is not part of S. But from the perspective of $A(S^{(i)})$ the example Z_i' is seen, since it is part of $S^{(i)}$ via the substitution that defines the i-th hybrid sample. This shows that the instrument $\Delta(A)$ also measures the average loss difference of the algorithm on seen and unseen examples. We therefore have reason to suspect that average stability relates to generalization gap as the next proposition confirms.

Proposition 1. *The expected generalization gap equals average stability:*

$$\mathbb{E}[\Delta_{\text{gen}}(A(S))] = \Delta(A)$$

Proof. By linearity of expectation,

$$\begin{aligned}\mathbb{E}[\Delta_{\text{gen}}(A(S))] &= \mathbb{E}\left[R[A(S)] - R_S[A(S)]\right] \\ &= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathit{loss}(A(S), Z_i')\right] - \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathit{loss}(A(S), Z_i)\right].\end{aligned}$$

Here, we used that Z_i' is an example drawn from the distribution that does not appear in the set S, while Z_i does appear in S. At the same time, Z_i and Z_i' are identically distributed and independent of the other examples. Therefore,

$$\mathbb{E}\, \mathit{loss}(A(S), Z_i) = \mathbb{E}\, \mathit{loss}(A(S^{(i)}), Z_i').$$

Applying this identity to each term in the empirical risk above, and comparing with the definition of $\Delta(A)$, we conclude

$$\mathbb{E}[R[A(S)] - R_S[A(S)]] = \Delta(A). \qquad \square$$

Uniform stability

While average stability gave us an exact characterization of generalization error, it can be hard to work with the expectation over S and S'. Uniform stability replaces the averages by suprema, leading to a stronger but useful notion.

Definition 3. *The* uniform stability *of an algorithm A is defined as*

$$\Delta_{\text{sup}}(A) = \sup_{\substack{S,S' \in (\mathcal{X}\times\mathcal{Y})^n \\ d_H(S,S')=1}} \sup_{z \in \mathcal{X}\times\mathcal{Y}} |\mathit{loss}(A(S), z) - \mathit{loss}(A(S'), z)|,$$

where $d_H(S, S')$ is the Hamming distance between tuples S and S'.

In this definition, it is important to note that the z has nothing to do with S and S'. Uniform stability is effectively computing the worst-case difference in the predictions of the learning algorithm run on two arbitrary datasets that differ in exactly one point.

Uniform stability upper bounds average stability, and hence uniform stability upper bounds generalization gap (in expectation). Thus, we have the corollary

$$\mathbb{E}[\Delta_{\text{gen}}(A(S))] \leq \Delta_{\text{sup}}(A)$$

This corollary turns out to be surprisingly useful since many algorithms are uniformly stable. For example, strong convexity of the loss function is sufficient for the uniform stability of empirical risk minimization, as we will see next.

Stability of empirical risk minimization

We now show that empirical risk minimization is uniformly stable provided under strong assumptions on the loss function. One important assumption we need is that the loss function $loss(w, z)$ is differentiable and *strongly convex* in the model parameters w for every example z. What this means is that for every example z and for all $w, w' \in \Omega$,

$$loss(w', z) \geq loss(w, z) + \langle \nabla loss(w, z), w' - w \rangle + \frac{\mu}{2}\|w - w'\|^2 .$$

There's only one property of strong convexity we'll need. Namely, if $\Phi \colon \mathbb{R}^d \to \mathbb{R}$ is μ-strongly convex and w^* is a stationary point (and hence global minimum) of the function Φ, then we have

$$\Phi(w) - \Phi(w^*) \geq \frac{\mu}{2}\|w - w^*\|^2 .$$

The second assumption we need is that $loss(w, z)$ is *L-Lipschitz* in w for every z, i.e., $\|\nabla loss(w, z)\| \leq L$. Equivalently, this means $|loss(w, z) - loss(w', z)| \leq L\|w - w'\|$.

Theorem 1. *Assume that for every z, $loss(w, z)$ is μ-strongly convex in w over the domain Ω, i.e., Further assume that, that the loss function $loss(w, z)$ is L-Lipschitz in w for every z. Then, empirical risk minimization (ERM) satisfies*

$$\Delta_{\text{sup}}(ERM) \leq \frac{4L^2}{\mu n} .$$

Proof. Let $\widehat{w}_S = \arg\min_{w \in \Omega} \frac{1}{n} \sum_{i=1}^{n} loss(w, z_i)$ denote the empirical risk minimizer on the sample S. Fix arbitrary samples S, S' of size n that differ in

a single index $i \in \{1, \ldots, n\}$ where S contains z_i and S' contains z_i'. Fix an arbitrary example z. We need to show that

$$|loss(\widehat{w}_S, z) - loss(\widehat{w}_{S'}, z)| \leq \frac{4L^2}{\mu n}.$$

Since the loss function is L-Lipschitz, it suffices to show that

$$\|\widehat{w}_S - \widehat{w}_{S'}\| \leq \frac{4L}{\mu n}.$$

On the one hand, since $\widehat{w}_S$ minimizes the empirical risk by definition, it follows from the strong convexity of the empirical risk that

$$\frac{\mu}{2}\|\widehat{w}_S - \widehat{w}_{S'}\|^2 \leq R_S[\widehat{w}_{S'}] - R_S[\widehat{w}_S].$$

On the other hand, we can bound the right hand side as

$$\begin{aligned}
&R_S[\widehat{w}_{S'}] - R_S[\widehat{w}_S] \\
&= \frac{1}{n}(loss(\widehat{w}_{S'}, z_i) - loss(\widehat{w}_S, z_i)) + \frac{1}{n}\sum_{i \neq j}(loss(\widehat{w}_{S'}, z_j) - loss(\widehat{w}_S, z_j)) \\
&= \frac{1}{n}(loss(\widehat{w}_{S'}, z_i) - loss(\widehat{w}_S, z_i)) + \frac{1}{n}(loss(\widehat{w}_S, z_i') - loss(\widehat{w}_{S'}, z_i')) \\
&\quad + (R_{S'}[\widehat{w}_{S'}] - R_{S'}[\widehat{w}_S]) \\
&\leq \frac{1}{n}|loss(\widehat{w}_{S'}, z_i) - loss(\widehat{w}_S, z_i)| + \frac{1}{n}|loss(\widehat{w}_S, z_i') - loss(\widehat{w}_{S'}, z_i')| \\
&\leq \frac{2L}{n}\|\widehat{w}_{S'} - \widehat{w}_S\|.
\end{aligned}$$

Here, we used the assumption that *loss* is L-Lipschitz and the fact that

$$R_{S'}[\widehat{w}_{S'}] - R_{S'}[\widehat{w}_S] \leq 0.$$

Putting together the strong convexity property and our calculation above, we find

$$\|\widehat{w}_{S'} - \widehat{w}_S\| \leq \frac{4L}{\mu n}.$$

Hence, $\Delta_{\text{sup}}(\text{ERM}) \leq \frac{4L^2}{\mu n}$.

□

An interesting point about this result is that there is no explicit reference to the complexity of the model class referenced by Ω.

Stability of regularized empirical risk minimization

Some empirical risk minimization problems, such as the Perceptron (ERM with hinge loss) we saw earlier, are convex but not strictly convex. We can turn convex problems into strongly convex problems by adding an ℓ_2*-regularization* term to the loss function:

$$r(w,z) = loss(w,z) + \frac{\mu}{2}\|w\|^2 .$$

The last term is named ℓ_2-regularization, *weight decay*, or *Tikhonov regularization* depending on field and context.

By construction, if the loss is convex, then the regularized loss $r(w,z)$ is μ-strongly convex. Hence, our previous theorem applies. However, by adding regularization we changed the objective function. The optimizer of the regularized objective is in general not the same as the optimizer of the unregularized objective.

Fortunately, A simple argument shows that solving the regularized objective also solves the unregularized objective. The idea is that assuming $\|w\| \leq B$ we can set the regularization parameter $\mu = \frac{L}{B\sqrt{n}}$. This ensures that the regularization term $\mu\|w\|^2$ is at most $O(\frac{LB}{\sqrt{n}})$ and therefore the minimizer of the regularized risk also minimizes the unregularized risk up to error $O(\frac{LB}{\sqrt{n}})$. Plugging this choice of μ into the ERM stability theorem, the generalization gap will also be $O(\frac{LB}{\sqrt{n}})$.

The case of regularized hinge loss

Let's relate the generalization theory we just saw to the familiar case of the perceptron algorithm from Chapter 3. This corresponds to the special case of minimizing the regularized hinge loss

$$r(w,(x,y)) = \max\{1 - y\langle w,x\rangle, 0\} + \frac{\mu}{2}\|w\|^2 .$$

Moreover, we assume that the data are are linearly separable with margin γ.

Denoting by $\widehat{w}_S$ the empirical risk minimizer on a random sample S of size n, we know that

$$\frac{\mu}{2}\|\widehat{w}_S\|^2 \leq R_S(\widehat{w}_S) \leq R_S(0) = 1 .$$

Hence, $\|\widehat{w}_S\| \leq B$ for $B = \sqrt{2/\mu}$. We can therefore restrict our domain to the Euclidean ball of radius B. If the data are also bounded, say $\|x\| \leq D$, we further get that

$$\|\nabla_w r(w,z)\| \leq \|x\| + \mu\|w\| = D + \mu B .$$

Hence, the regularized hinge loss is L-Lipschitz with

$$L = D + \mu B = D + \sqrt{2\mu}\,.$$

Let w_γ be a maximum margin hyperplane for the sample S. We know that the empirical loss will satisfy

$$R_S[\widehat{w}_S] \leq R_S[w_\gamma] = \frac{\mu}{2}\|w_\gamma\|^2 = \frac{\mu}{2\gamma^2}\,.$$

Hence, by Theorem 1,

$$\mathbb{E}[R[\widehat{w}_S]] \leq \mathbb{E}[R_S[\widehat{w}_S]] + \Delta_{\text{sup}}(\text{ERM}) \leq \frac{\mu}{2\gamma^2} + \frac{4(D+\sqrt{2\mu})^2}{\mu n}$$

Setting $\mu = \frac{2\gamma D}{\sqrt{n}}$ and noting that $\gamma \leq D$ gives that

$$\mathbb{E}[R[\widehat{w}_S]] \leq O\left(\frac{D}{\gamma\sqrt{n}}\right)\,.$$

Finally, since the regularized hinge loss upper bounds the zero-one loss, we can conclude that

$$\mathbb{P}[Y\widehat{w}_S^T X < 0] \leq O\left(\frac{D}{\gamma\sqrt{n}}\right)\,,$$

where the probability is taken over both sample S and test point (X, Y). Applying Markov's inequality to the sample, we can conclude the same bound holds for a typical sample up to constant factors.

This bound is proportional to the square root of the bound we saw for the perceptron in Chapter 3. As we discussed earlier, this rate is slower than the perceptron rate as it does not explicitly take into account the fact that the empirical risk is zero. However, it is worth noting that the relationship between the variables in question—diameter, margin, and number of samples—is precisely the same as for the perceptron. This kind of bound is common and we will derive it a few more times in this chapter.

Stability analysis combined with explicit regularization and convexity thus give an appealing conceptual and mathematical approach to reasoning about generalization. However, empirical risk minimization involving non-linear models is increasingly successful in practice and generally leads to non-convex optimization problems.

Model complexity and uniform convergence

We briefly review other useful tools to reason about generalization. Arguably, the most basic is based on counting the number of different functions that can be described with the given model parameters.

Given a sample S of n independent draws from the same underlying distribution, the empirical risk $R_S[f]$ for a fixed function f is an average of n random variables, each with mean equal to the risk $R[f]$. Assuming for simplicity that the range of our loss function is bounded in the interval $[0, 1]$, Hoeffding's bound gives us the tail bound

$$\mathbb{P}\left[R_S[f] > R[f] + t\right] \leq \exp(-2nt^2).$$

By applying the union bound to a finite set of functions $\mathcal{F}$ we can guarantee that with probability $1 - \delta$, we have for all functions $f \in \mathcal{F}$ that

$$\Delta_{\text{gen}}(f) \leq \sqrt{\frac{\ln|\mathcal{F}| + \ln(1/\delta)}{n}}. \qquad (1)$$

The cardinality bound $|\mathcal{F}|$ is a basic measure of the complexity of the model family $\mathcal{F}$. We can think of the term $\ln(\mathcal{F})$ as a measure of complexity of the function class $\mathcal{F}$. The gestalt of the generalization bound as "$\sqrt{\text{complexity}/n}$" routinely appears with varying measures of complexity.

VC dimension

Bounding the generalization gap from above for all functions in a function class is called *uniform convergence*. A classical tool to reason about uniform convergence is the Vapnik-Chervonenkis dimension (VC dimension) of a function class $\mathcal{F} \subseteq X \to Y$, denoted $\text{VC}(\mathcal{F})$. It's defined as the size of the largest set $Q \subseteq X$ such that for any Boolean function $h\colon Q \to \{-1, 1\}$, there is a predictor $f \in \mathcal{F}$ such that $f(x) = h(x)$ for all $x \in Q$. In other words, if there is a size-d sample Q such that the functions of $\mathcal{F}$ induce all 2^d possible binary labelings of Q, then the VC-dimension of $\mathcal{F}$ is at least d.

The VC-dimension measures the ability of the model class to conform to an arbitrary labeling of a set of points. The so-called VC inequality implies that with probability $1 - \delta$, we have for all functions $f \in \mathcal{F}$

$$\Delta_{\text{gen}}(f) \leq \sqrt{\frac{\text{VC}(\mathcal{F})\ln n + \ln(1/\delta)}{n}}. \qquad (2)$$

We can see that the complexity term $\text{VC}(\mathcal{F})$ refines our earlier cardinality bound since $\text{VC}(\mathcal{F}) \leq \log|\mathcal{F}| + 1$. However VC-dimension also applies to infinite model classes. Linear models over $\mathbb{R}^d$ have VC-dimension d, corresponding to the number of model parameters. Generally speaking, VC dimension tends to grow with the number of model parameters for many model families of interest. In such cases, the bound in Equation 2 becomes useless once the number of model parameters exceeds the size of the sample.

However, the picture changes significantly if we consider notions of model complexity different than raw counts of parameters. Consider two sets of vectors X_0 and X_1 all having Euclidean norm bounded by D. Let $\mathcal{F}$ be the set of all linear functions f such that $f(x) = w^T x$ with $||w|| \leq \gamma^{-}1$, $f(x) \leq -1$ if $x \in X_0$, and $f(x) \geq 1$ if $x \in X_1$. Vapnik showed[4] that the VC dimension of this set of hyperplanes was $\frac{D^2}{\gamma^2}$. As described in a survey of support vector machines by Burges, the worst case arrangement of n data points is a simplex in $n-2$ dimensions.[5] Plugging this VC-dimension into our generalization bound yields

$$\Delta_{\text{gen}}(f) \leq \sqrt{\frac{D^2 \ln n + \gamma^2 \ln(1/\delta)}{\gamma^2 n}}.$$

We again see our Perceptron style generalization bound! This bound again holds when the empirical risk is nonzero. And the dimension of the data, d does not appear at all in this bound. The difference between the parametric model and the margin-like bound is that we considered properties of the data. In the *worst case* bound which counts parameters, it appears that high-dimensional prediction is impossible. It is only by considering data-specific properties that we can find a reasonable generalization bound.

Rademacher complexity

An alternative to VC-dimension is Rademacher complexity, a flexible tool that often is more amenable to calculations that incorporate problem-specific aspects such as restrictions on the distribution family or properties of the loss function. To get a generalization bound in terms of Rademacher complexity, we typically apply the definition not the model class $\mathcal{F}$ itself but to the class of functions $\mathcal{L}$ of the form $h(z) = \mathit{loss}(f, z)$ for some $f \in \mathcal{F}$ and a loss function *loss*. By varying the loss function, we can derive different generalization bounds.

Fix a function class $\mathcal{L} \subseteq Z \to \mathbb{R}$, which will later correspond to the composition of a predictor with a loss function, which is why we chose the symbol $\mathcal{L}$. Think of the domain Z as the space of labeled examples $z = (x, y)$. Fix a distribution P over the space Z.

The *empirical Rademacher complexity* of a function class $\mathcal{L} \subseteq Z \to \mathbb{R}$ with respect to a sample $\{z_1, \ldots, z_n\} \subseteq Z$ drawn i.i.d. from the distribution P is defined as:

$$\widehat{\mathfrak{R}}_n(\mathcal{L}) = \mathop{\mathbb{E}}_{\sigma \in \{-1,1\}^n} \left[\frac{1}{n} \sup_{h \in \mathcal{L}} \left| \sum_{i=1}^{n} \sigma_i h(z_i) \right| \right].$$

We obtain the *Rademacher complexity* $\mathfrak{R}_n(\mathcal{L}) = \mathbb{E}\left[\widehat{\mathfrak{R}}_n(\mathcal{L})\right]$ by taking the

expectation of the empirical Rademacher complexity with respect to the sample. Rademacher complexity measures the ability of a function class to interpolate a random sign pattern assigned to a point set.

One application of Rademacher complexity applies when the loss function is L-Lipschitz in the parameterization of the model class for every example z. This bound shows that with probability $1 - \delta$ for all functions $f \in \mathcal{F}$, we have

$$\Delta_{\text{gen}}(f) \leq 2L\mathfrak{R}_n(\mathcal{F}) + 3\sqrt{\frac{\log(1/\delta)}{n}}.$$

When applied to the hinge loss with the function class being hyperplanes of norm less than γ^{-1}, this bound again recovers the perceptron generalization bound

$$\Delta_{\text{gen}}(f) \leq 2\frac{D}{\gamma\sqrt{n}} + 3\sqrt{\frac{\log(1/\delta)}{n}}.$$

Margin bounds for ensemble methods

Ensemble methods work by combining many weak predictors into one strong predictor. The combination step usually involves taking a weighted average or majority vote of the weak predictors. Boosting and random forests are two ensemble methods that continue to be highly popular and competitive in various settings. Both methods train a sequence of small decision trees, each on its own achieving modest accuracy on the training task. However, so long as different trees make errors that aren't too correlated, we can obtain a higher accuracy model by taking, say, a majority vote over the individual predictions of the trees.

Researchers in the 1990s already observed that boosting often continues to improve test accuracy as more weak predictors are added to the ensemble. The complexity of the entire ensemble was thus often far too large to apply standard uniform convergence bounds.

A proffered explanation was that boosting, while growing the complexity of the ensemble, also improved the *margin* of the ensemble predictor. Assuming that the final predictor $f\colon X \to \{-1, 1\}$ is binary, its *margin* on an example (x, y) is defined as the value $yf(x)$. The larger the margin the more "confident" the predictor is about its prediction. A margin $yf(x)$ just above 0 shows that the weak predictors in the ensemble were nearly split evenly in their weighted votes.

An elegant generalization bound relates the risk of any predictor f to the fraction of correctly labeled training examples at a given margin θ. Below let $R[f]$ be the risk of f w.r.t. zero-one loss. However, let $R_S^\theta(f)$ be the empirical risk with respect to *margin errors* at level θ, i.e., the loss $\mathbf{1}(yf(x) \leq$

θ) that penalizes errors where the predictor is within an additive θ margin of making a mistake.

Theorem 2. *With probability $1-\delta$, every convex combination f of base predictors in $\mathcal{H}$ satisfies the following bound for every $\theta > 0$:*

$$R[f] - R_S^\theta[f] \leq O\left(\frac{1}{\sqrt{n}}\left(\frac{\mathrm{VC}(\mathcal{H})\log n}{\theta^2} + \log(1/\delta)\right)^{1/2}\right)$$

The theorem can be proved using Rademacher complexity. Crucially, the bound only depends on the VC dimension of the base class $\mathcal{H}$ but not the complexity of ensemble. Moreover, the bound holds for all $\theta > 0$ and so we can choose θ after knowing the margin that manifested during training.

Margin bounds for linear models

Margins also play a fundamental role for linear prediction. We saw one margin bound for linear models in our chapter on the Perceptron algorithm. Similar bounds hold for other variants of linear prediction. We'll state the result here for a simple least squares problem:

$$w^* = \arg \min_{w:\ \|w\| \leq B} \frac{1}{n}\sum_{i=1}^{n} \left(\langle x_i, w\rangle - y\right)^2$$

In other words, we minimize the empirical risk w.r.t. the squared loss over norm bounded linear separators, call this class $\mathcal{W}_B$. Further assume that all data points satisfy $\|x_i\| \leq 1$ and $y \in \{-1, 1\}$. Analogous to the margin bound in Theorem 2, it can be shown that with probability $1-\delta$ for every linear predictor f specified by weights in $\mathcal{W}_B$ we have

$$R[f] - R_S^\theta[f] \leq 4\frac{\mathfrak{R}(\mathcal{W}_B)}{\theta} + O\left(\frac{\log(1/\delta)}{\sqrt{n}}\right) .$$

Moreover, given the assumptions on the data and model class we made, the Rademacher complexity satisfies $\mathfrak{R}(\mathcal{W}) \leq B/\sqrt{n}$. What we can learn from this bound is that the relevant quantity for generalization is the ratio of complexity to margin B/θ .

It's important to understand that margin is a scale-sensitive notion; it only makes sense to talk about it after suitable normalization of the parameter vector. If the norm didn't appear in the bound we could scale up the parameter vector to achieve any margin we want. For linear predictors the Euclidean norm provides a natural and often suitable normalization.

Generalization from algorithms

In the overparameterized regime, there are always an infinite number of models that minimize empirical risk. However, when we run a particular algorithm, the algorithm usually returns only one from this continuum. In this section, we show how directly analyzing algorithmic iteration can itself yield generalization bounds.

One pass optimization of stochastic gradient descent

As we briefly discussed in the optimization chapter, we can interpret the convergence analysis of stochastic gradient descent as directly providing a generalization bound for a particular variant of SGD. Here we give the argument in full detail. Suppose that we choose a loss function that upper bounds the number of mistakes. That is $loss(\widehat{y}, y) \geq \mathbb{1}\{y\widehat{y} < 0\}$. The hinge loss would be such an example. Choose the function R to be the risk (not empirical risk!) with respect to this loss function:

$$R[w] = \mathbb{E}[loss(w^T x, y)]$$

At each iteration, suppose we gain access to an example pair (x_i, y_i) sampled i.i.d. from the a data generating distribution. Then when we run the stochastic gradient method, the iterates are

$$w_{t+1} = w_t - \alpha_t e(w_t^T x_t, y_t) x_t \,, \quad \text{where} \quad e(z, y) = \frac{\partial loss(z, y)}{\partial z} \,.$$

Suppose that for all x, $\|x\| \leq D$. Also suppose that $|e(z, y)| \leq C$. Then the SGD convergence theorem tells us that after n steps, starting at $w_0 = 0$ and using an appropriately chosen constant step size, the average of our iterates $\bar{w}_n$ will satisfy

$$\mathbb{P}[\text{sign}(\bar{w}_n^T x) \neq y] \leq \mathbb{E}[R[\bar{w}_n]] \leq R[w_\star] + \frac{CD\|w_\star\|}{\sqrt{n}} \,.$$

This inequality tells us that we will find a distribution boundary that has low *population* risk after seeing n samples. And the population risk itself lets us upper bound the probability of our model making an error on new data. That is, this inequality is a generalization bound.

We note here that this importantly does not measure our empirical risk. By running stochastic gradient descent, we can find a low-risk model without ever computing the empirical risk.

Let us further assume that the population can be separated with large margin. As we showed when we discussed the Perceptron, the margin is

equal to the inverse of the norm of the corresponding hyperplane. Suppose we ran the stochastic gradient method using a hinge loss. In this case, $C = 1$, so, letting γ denote the maximum margin, we get the simplified bound

$$\mathbb{P}[\text{sign}(\bar{w}_n^T x) \neq y] \leq \frac{D}{\gamma\sqrt{n}}\,.$$

Note that the Perceptron analysis did not have a step size parameter that depended on the problem instance. But, on the other hand, this analysis of SGD holds regardless of whether the data is separable or whether zero empirical risk is achieved after one pass over the data. The stochastic gradient analysis is more general but generality comes at the cost of a looser bound on the probability of error on new examples.

Uniform stability of stochastic gradient descent

Above we showed that empirical risk minimization is stable no matter what optimization method we use to solve the objective. One weakness is that the analysis applied to the exact solution of the optimization problem and only applies for strongly convex loss function. In practice, we might only be able to compute an approximate empirical risk minimizer and may be interested in losses that are not strongly convex. Fortunately, we can also show that some optimization methods are stable even if they don't end up computing a minimizer of a strongly convex empirical risk. Specifically, this is true for the stochastic gradient method under suitable assumptions. Below we state one such result which requires the assumption that the loss function is *smooth*. A continuously differentiable function $f\colon \mathbb{R}^d \to \mathbb{R}$ is β-smooth if $\|\nabla f(y) - \nabla f(x)\| \leq \beta\|y - x\|$.

Theorem 3. *Assume a continuously differentiable loss function that is β-smooth and L-Lipschitz on every example and convex. Suppose that we run the stochastic gradient method (SGM) with step sizes $\eta_t \leq 2/\beta$ for T steps. Then, we have*

$$\Delta_{\text{sup}}(SGM) \leq \frac{2L^2}{n}\sum_{t=1}^{T}\eta_t\,.$$

The theorem allows for SGD to sample the same data points multiple times, as is common practice in machine learning. The stability approach also extends to the non-convex case albeit with a much weaker quantitative bound.

What solutions does stochastic gradient descent favor?

We reviewed empirical evidence that explicit regularization is not necessary for generalization. Researchers therefore believe that a combination of

data generating distribution and optimization algorithm perform *implicit regularization*. Implicit regularization describes the tendency of an algorithm to seek out solutions that generalize well on their own on a given a dataset without the need for explicit correction. Since the empirical phenomena we reviewed are all based on gradient methods, it makes sense to study implicit regularization of gradient descent. While a general theory for non-convex problems remains elusive, the situation for linear models is instructive.

Consider again the linear case of gradient descent or stochastic gradient descent:

$$w_{t+1} = w_t - \alpha e_t x_t$$

where e_t is the gradient of the loss at the current prediction. As we showed in the optimization chapter, if we run this algorithm to convergence, we must have the resulting $\widehat{w}$ lies in the span of the data, and that it interpolates the data. These two facts imply that the optimal $\widehat{w}$ is the minimum Euclidean norm solution of $Xw = y$. That is, w solves the optimization problem

$$\begin{array}{ll} \text{minimize} & \|w\|^2 \\ \text{subject to} & y_i w^T x_i = 1 \,. \end{array}$$

Moreover, a closed form solution of this problem is given by

$$\widehat{w} = X^T (X X^T)^{-1} y \,.$$

That is, when we run stochastic gradient descent we converge to a very specific solution. Now what can we say about the generalization properties of this minimum norm interpolating solution?

The key to analyzing the generalization of the minimum norm solution will be a stability-like argument. We aim to control the error of the model trained on the first m data points on the next data point in the sequence, x_{m+1}. To do so, we use a simple identity that follows from linear algebra.

Lemma 1. *Let S be an arbitrary set of $m \geq 2$ data points. Let w_{m-1} and w_m denote the minimum norm solution trained on the first $m-1$ and m points respectively. Then*

$$(1 - y_m \langle w_{m-1}, x_m \rangle)^2 = s_m^2 (\|w_m\|^2 - \|w_{m-1}\|^2) \,,$$

where

$$s_m := \operatorname{dist}\left(\operatorname{span}(x_1, \ldots, x_{m-1}), x_m\right) \,.$$

We hold off on proving this lemma and first prove our generalization result with the help of this lemma.

Theorem 4. *Let S_{n+1} denote a set of $n+1$ i.i.d. samples. Let S_j denote the first j samples and w_j denote the solution of minimum norm that interpolates these j points. Let R_j denote the maximum norm of $\|x_i\|$ for $1 \leq i \leq j$. Let (x, y) denote another independent sample from $\mathcal{D}$. Then if $\epsilon_j := \mathbb{E}[(1 - yf_{S_j}(x))^2]$ is a non-increasing sequence, we have*

$$\mathbb{P}[y\langle w_n, x\rangle < 0] \leq \frac{\mathbb{E}[R_j^2 \|w_{n+1}\|^2]}{n}.$$

Proof. Lemma together with the bound $s_i^2 \leq R_{n+1}^2$ yields the inequality

$$\mathbb{E}[(1 - y\langle w_i, x\rangle)^2] \leq (\mathbb{E}[R_{n+1}^2 \|w_{i+1}\|^2] - \mathbb{E}[R_{n+1}^2 \|w_i\|^2]).$$

Here, we could drop the subscript on x and y on the left-hand side as they are identically distributed to (x_{i+1}, y_{i+1}). Adding these inequalities together gives the bound

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[(1 - yf_{S_i}(x))^2] \leq \frac{\mathbb{E}[R_{n+1}^2 \|w_{n+1}\|^2]}{n}.$$

Assuming the sequence is decreasing means that the minimum summand of the previous inequality is $\mathbb{E}[(1 - yf_i(x))^2]$. This and Markov's inequality prove the theorem.

□

This proof reveals that the minimum norm solution, the one found by running stochastic gradient descent to convergence, achieves a nearly identical generalization bound as the Perceptron, even with the fast $1/n$ rate. Here, nothing is assumed about margin, but instead we assume that the complexity of the interpolating solution does not grow rapidly as we increase the amount of data we collect. This proof combines ideas from stability, optimization, and model complexity to find yet another explanation for why gradient methods find high-quality solutions to machine learning problems.

Proof of Lemma 1

We conclude with the deferred proof of Lemma 1.

Proof. Let $K = XX^T$ denote the kernel matrix for S. Partition K as

$$K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

where K_{11} is $(m-1) \times (m-1)$ and K_{22} is a scalar equal to $\langle x_m, x_m \rangle$. Similarly, partition the vector of labels y so that $y^{(m-1)}$ denotes the first $m-1$ labels. Under this partitioning,

$$\langle w_{m-1}, x_m \rangle = K_{21} K_{11}^{-1} y^{(m-1)} .$$

Now note that

$$s_m^2 = K_{22} - K_{21} K_{11}^{-1} K_{12} .$$

Next, using the formula for inverting partitioned matrices, we find

$$K^{-1} = \begin{bmatrix} (K_{11} - K_{12} K_{21} K_{22}^{-1})^{-1} & s_m^{-2} K_{11}^{-1} K_{12} \\ s_m^{-2} (K_{11}^{-1} K_{12})^T & s_m^{-2} \end{bmatrix} .$$

By the matrix inversion lemma we have

$$(K_{11} - K_{12} K_{21} K_{22}^{-1})^{-1} = K_{11}^{-1} + s_m^{-2} \left(K_{21} K_{11}^{-1} \right)^T \left(K_{21} K_{11}^{-1} \right) .$$

Hence,

$$\begin{aligned} \|w_i\| &= y^T K^{-1} y \\ &= s_m^{-2} (y_m^2 - 2 y_m \langle w_{m-1}, x_m \rangle + \langle w_{m-1}, x_m \rangle^2) + {y^{(m-1)}}^T K_{11}^{-1} y^{(m-1)} . \end{aligned}$$

Rearranging terms proves the lemma.

□

Looking ahead

Despite significant effort and many recent advances, the theory of generalization in overparameterized models still lags behind the empirical phenomenology. What governs generalization remains a matter of debate in the research community.

Existing generalization bounds often do not apply directly to practice by virtue of their assumptions, are quantitatively too weak to apply to heavily overparameterized models, or fail to explain important empirical observations. However, it is not just a lack of quantitative sharpness that limits our understanding of generalization.

Conceptual questions remain open: What is it a successful theory of generalization should do? What are formal success criteria? Even a qualitative theory of generalization, that is not quantitatively precise in concrete settings, may be useful if it leads to the successful algorithmic interventions. But how do we best evaluate the value of a theory in this context?

Our focus in this chapter was decidedly narrow. We discussed how to related risk and empirical risk. This perspective can only capture questions that relate performance on a sample to performance on the very same distribution that the sample was drawn from. What is left out are important questions of *extrapolation* from a training environment to testing conditions that differ from training. Overparameterized models that generalize well in the narrow sense can fail dramatically even with small changes in the environment. We will revisit the question of generalization for overparameterized models in our chapter on deep learning.

Chapter notes

The tight characterization of generalization gap in terms of average stability, as well as stability of regularized empirical risk minimization (Theorem 1), is due to Shalev-Shwartz et al.[6] Uniform stability was introduced by Bousquet and Elisseeff.[7] For additional background on VC dimension and Rademacher complexity, see, for example, the text by Shalev-Shwartz and Ben-David.[8]

The double descent figure is from work of Belkin et al.[9] Earlier work pointed out similar empirical risk-complexity relationships.[10] The empirical findings related to the randomization test and the role of regularization are due to Zhang et al.[11]

Theorem 2 is due to Schapire et al.[12] Later work showed theoretically that boosting maximizes margin.[13,14] The margin bound for linear models follows from more general results of Kakade, Sridharan, and Tewari[15] that build on earlier work by Bartlett and Mendelson,[16] as well as work of Koltchinskii and Panchenko.[17] Rademacher complexity bounds for family of neural networks go back to work of Bartlett[18] and remain and active research topic. We will see more on this in our chapter on deep learning.

The uniform stability bound for stochastic gradient descent is due to Hardt, Recht, and Singer.[19] Subsequent work further explores the generalization performance stochastic gradient descent in terms of its stability properties. Theorem 4 and Lemma 1 are due to Liang and Recht.[20]

There has been an explosion of work on generalization and overparameterization in recent years. See, also, recent work exploring how other norms shed light on generalization performance.[21] Our exposition is by no means a representative survey of the broad literature on this topic. There are several ongoing lines of work we did not cover: PAC-Bayes bounds,[22] compression bounds,[23] and arguments about the properties of the optimization landscape.[24] This chapter builds on a chapter by Hardt,[25] but contains several structural changes as well as different results.

Bibliography

[1] Tengyuan Liang, Alexander Rakhlin, and Xiyu Zhai. On the multiple descent of minimum-norm interpolants and restricted lower isometry of kernels. In *Conference on Learning Theory*, 2020.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, 2016.

[3] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism. *Advances in Neural Information Processing Systems*, 32:103–112, 2019.

[4] Vladimir Vapnik. *Statistical Larning Theory*. Wiley, 1998.

[5] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[6] Shai Shalev-Shwartz, Ohad Shamir, Nathan Srebro, and Karthik Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11(Oct):2635–2670, 2010.

[7] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2(Mar):499–526, 2002.

[8] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[9] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias-variance trade-off. *Proceedings of the National Academy of Sciences*, 2019.

[10] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv:1412.6614*, 2014.

[11] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

[12] Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651–1686, 1998.

[13] Tong Zhang and Bin Yu. Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, 33:1538–1579, 2005.

[14] Matus Telgarsky. Margins, shrinkage, and boosting. In *International Conference on Machine Learning*, 2013.

[15] Sham M. Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in Neural Information Processing Systems*, pages 793–800, 2009.

[16] Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[17] Vladimir Koltchinskii and Dmitry Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 2002.

[18] Peter L. Bartlett. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *Transactions on Information Theory*, 44(2):525–536, 1998.

[19] Moritz Hardt, Benjamin Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, 2016.

[20] Tengyuan Liang and Benjamin Recht. Interpolating classifiers make few mistakes. *arXiv:2101.11815*, 2021.

[21] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5947–5956, 2017.

[22] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv:1703.11008*, 2017.

[23] Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. *arXiv:1802.05296*, 2018.

[24] Chiyuan Zhang, Qianli Liao, Alexander Rakhlin, Karthik Sridharan, Brando Miranda, Noah Golowich, and Tomaso Poggio. Theory of deep learning III: Generalization properties of SGD. Technical report, Discussion paper, Center for Brains, Minds and Machines (CBMM). Preprint, 2017.

[25] Moritz Hardt. Generalization in overparameterized models. In Tim Roughgarden, editor, *Beyond the Worst-Case Analysis of Algorithms*, page 486–505. Cambridge University Press, 2021.