Chapter 12
Generalization Capability of Deep Learning

12.1 Introduction

One of the main reasons for the enormous success of deep neural networks is their amazing ability to generalize, which seems mysterious from the perspective of classic machine learning. In particular, the number of trainable parameters in deep neural networks is often greater than the training data set, this situation being notorious for overfitting from the point of view of classical statistical learning theory. However, empirical results have shown that a deep neural network generalizes well at the test phase, resulting in high performance for the unseen data.

This apparent contradiction has raised questions about the mathematical foundations of machine learning and their relevance to practitioners. A number of theoretical papers have been published to understand the intriguing generalization phenomenon in deep learning models [147–153]. The simplest approach to studying generalization in deep learning is to prove a generalization bound, which is typically an upper limit for test error. A key component in these generalization bounds is the notion of complexity measure: a quantity that monotonically relates to some aspect of generalization. Unfortunately, it is difficult to find tight bounds for a deep neural network that can explain the fascinating ability to generalize.

Recently, the authors in [154, 155] have delivered groundbreaking work that can reconcile classical understanding and modern practice in a unified framework. The so-called "double descent" curve extends the classical U-shaped bias-variance trade-off curve by showing that increasing the model capacity beyond the interpolation point leads to improved performance in the test phase. Particularly, the induced bias by optimization algorithms such as the stochastic gradient descent (SGD) offers simpler solutions that improve generalization in the over-parameterized regime. This relationship between the algorithms and structure of machine learning models describes the limits of classical analysis and has implications for the theory and practice of machine learning.

J. C. Ye, *Geometry of Deep Learning*, Mathematics in Industry 37,
https://doi.org/10.1007/978-981-16-6046-7_12

This chapter also presents new results showing that a generalization bound based on the robustness of the algorithm can be a promising tool to understand the generalization ability of the ReLU network. In particular, we claim that it can potentially offer a tight generalization bound that depends on the piecewise linear nature of the deep neural network and the inductive bias of the optimization algorithms.

12.2 Mathematical Preliminaries

Let Q be an arbitrary distribution over $z := (\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{Y}$ denote the input and output of the learning algorithm, and $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ refer to the sample space. Let $\mathcal{F}$ be a hypothesis class and let $\ell(\boldsymbol{f}, z)$ be a loss function. For the case of regression with MSE loss, the loss can be defined as

$$\ell(\boldsymbol{f}, z) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{f}(\boldsymbol{x})\|^2.$$

Over the choice of an i.i.d. training set $\mathcal{S} := \{z_n\}_{n=1}^N$, which is sampled according to Q, an algorithm $\mathcal{A}$ returns the estimated hypothesis

$$\boldsymbol{f}_{\mathcal{S}} = \mathcal{A}(\mathcal{S}). \tag{12.1}$$

For example, the estimated hypothesis from the popular empirical risk minimization (ERM) principle [10] is given by

$$\boldsymbol{f}_{ERM} = \arg\min_{\boldsymbol{f} \in \mathcal{F}} \hat{R}_N(\boldsymbol{f}), \tag{12.2}$$

where the empirical risk $\hat{R}_N(\boldsymbol{f})$ is defined by

$$\hat{R}_N(\boldsymbol{f}) := \frac{1}{N} \sum_{n=1}^{N} \ell\left(\boldsymbol{f}, z_n\right), \tag{12.3}$$

which is assumed to uniformly converge to the population (or expected) risk defined by:

$$R(\boldsymbol{f}) = \mathbb{E}_{z \sim Q} \ell\left(\boldsymbol{f}, z\right). \tag{12.4}$$

If uniform convergence holds, then the empirical risk minimizer (ERM) is consistent, that is, the population risk of the ERM converges to the optimal population risk, and the problem is said to be learnable using the ERM [10].

In fact, learning algorithms that satisfy such performance guarantees are called the probably approximately correct (PAC) learning [156]. Formally, PAC learnability is defined as follows.

Definition 12.1 (PAC Learnability [156]) A concept class C is PAC learnable if there exist some algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot)$ such that the following holds. Pick any target concept $\boldsymbol{c} \in C$. Pick any input distribution $\mathcal{P}$ over $\mathcal{X}$. Pick any $\epsilon, \delta \in [0, 1]$. Define $\mathcal{S} := \{\boldsymbol{x}_n, \boldsymbol{c}(\boldsymbol{x}_n)\}_{n=1}^{N}$ where $\boldsymbol{x}_n \sim \mathcal{P}$ are i.i.d samples. Given $N \geq poly(1/\epsilon, 1/\delta, \dim(\mathcal{X}), \text{size}(\boldsymbol{c}))$, where $\dim(\mathcal{X})$, $\text{size}(\boldsymbol{c})$ denote the computational costs of representing inputs $\boldsymbol{x} \in \mathcal{X}$ and target $\boldsymbol{c}$, the generalization error is bounded as

$$\mathbb{P}_{\boldsymbol{x} \sim Q} \{\mathcal{A}_{\mathcal{S}}(\boldsymbol{x}) \neq \boldsymbol{c}(\boldsymbol{x})\} \leq \epsilon, \tag{12.5}$$

where $\boldsymbol{A}_{\mathcal{S}}$ denotes the learned hypothesis by the algorithm $\mathcal{A}$ using the training data $\mathcal{S}$.

The PAC learnability is closely related to the generalization bounds. More specifically, the ERM could only be considered a solution to a machine learning problem or PAC-learnable if the difference between the training error and the generalization error, called the generalization gap, is small enough. This implies that the following probability should be sufficiently small:

$$\mathbb{P}\left\{\sup_{\boldsymbol{f} \in \mathcal{F}} |R(\boldsymbol{f}) - \hat{R}_N(\boldsymbol{f})| > \epsilon\right\}. \tag{12.6}$$

Note that this is the worst-case probability, so even in the worst-case scenario, we try to minimize the difference between the empirical risk and the expected risk.

A standard trick to bound the probability in (12.6) is based on concentration inequalities. For example, Hoeffding's inequality is useful.

Theorem 12.1 (Hoeffding's Inequality [157]) *If $x_1, x_2, \cdots, x_N$ are N i.i.d. samples of a random variable X distributed by $\mathcal{P}$, and $a \leq x_n \leq b$ for every n, then for a small positive nonzero value ϵ:*

$$\mathbb{P}\left\{\left|\mathbb{E}[X] - \frac{1}{N}\sum_{n=1}^{N} x_n\right| > \epsilon\right\} \leq 2\exp\left(\frac{-2N\epsilon^2}{(b-a)^2}\right). \tag{12.7}$$

Assuming that our loss is bounded between 0 and 1 using a 0/1 loss function or by squashing any other loss between 0 and 1, (12.6) can be bounded as follows using Hoeffding's inequality:

$$\begin{aligned}\mathbb{P}\left\{\sup_{f\in\mathcal{F}}|R(f)-\hat{R}_N(f)|>\epsilon\right\} &= \mathbb{P}\left\{\bigcup_{f\in\mathcal{F}}|R(f)-\hat{R}_N(f)|>\epsilon\right\} \\ &\overset{(a)}{\leq} \sum_{f\in\mathcal{F}}\mathbb{P}\left\{|R(f)-\hat{R}_N(f)|>\epsilon\right\} \\ &= 2|\mathcal{F}|\exp(-2N\epsilon^2),\end{aligned} \tag{12.8}$$

where $|\mathcal{F}|$ is the size of the hypothesis space and we use the union bound in (a) to obtain the inequality. By denoting the right hand side of the above inequality by δ, we can say that with probability at least $1-\delta$, we have

$$R(f) \leq \hat{R}_N(f) + \sqrt{\frac{\ln|\mathcal{F}| + \ln\frac{2}{\delta}}{2N}}. \tag{12.9}$$

Indeed, (12.9) is one of the simplest forms of the generalization bound, but still reveals the fundamental bias–variance trade-off in classical statistical learning theory. For example, the ERM for a given function class $\mathcal{F}$ results in the minimum empirical loss:

$$\hat{R}_N(f_{ERM}) = \min_{f\in\mathcal{F}}\hat{R}_N(f), \tag{12.10}$$

which goes to zero as the hypothesis class $\mathcal{F}$ becomes bigger. On the other hand, the second term in (12.9) grows with increasing $|\mathcal{F}|$. This trade-off in the generalization bound with respect to the hypothesis class size $|\mathcal{F}|$ is illustrated in Fig. 12.1.

Although the expression in (12.9) looks very nice, it turns out that the bound is very loose. This is due to the term $|\mathcal{F}|$ which originates from the union bound of all elements in the hypothesis class $\mathcal{F}$. In the following, we discuss some representative classical approaches to obtain tighter generation bounds.

12.2.1 Vapnik–Chervonenkis (VC) Bounds

One of the key ideas of the work of Vapnik and Chervonenkis [10] is to replace the union bound for all hypothesis class in (12.8) with the union bound of simpler empirical distributions. This idea is historically important, so we will review it here.

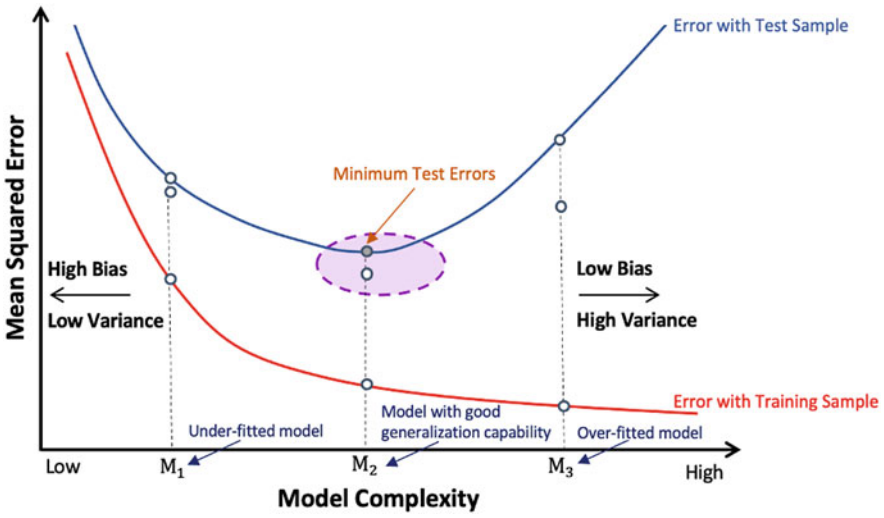

Fig. 12.1 Generation bound behavior according to the hypothesis class size $|\mathcal{F}|$

More specifically, consider independent samples $z'_n := (\boldsymbol{x}'_n, \boldsymbol{y}'_n)$ for $n = 1, \cdots, N$, which are often called "ghost" samples. The associated empirical risk is given by

$$\hat{R}'_N(\boldsymbol{f}) = \frac{1}{N} \sum_{n=1}^{N} \ell\left(\boldsymbol{f}, z'_n\right). \tag{12.11}$$

Then, we have the following symmetrization lemma.

Lemma 12.1 (Symmetrization[10]) *For a given sample set $\mathcal{S} := \{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$ and its ghost samples set $\mathcal{S}' := \{\boldsymbol{x}'_n, \boldsymbol{y}'_n\}_{n=1}^N$ from a distribution $\mathcal{Q}$ and for any $\epsilon > 0$ such that $\epsilon \geq \sqrt{2/N}$, we have*

$$\mathbb{P}\left\{\sup_{\boldsymbol{f}\in\mathcal{F}} |R(\boldsymbol{f}) - \hat{R}_N(\boldsymbol{f})| > \epsilon\right\} \leq 2\mathbb{P}\left\{\sup_{\boldsymbol{f}\in\mathcal{F}} |\hat{R}'_N(\boldsymbol{f}) - \hat{R}_N(\boldsymbol{f})| > \frac{\epsilon}{2}\right\}. \tag{12.12}$$

Vapnik and Chervonenkis [10] used the symmetrization lemma to obtain a much tighter generalization bound:

$$\begin{aligned}\mathbb{P}\left\{\sup_{\boldsymbol{f}\in\mathcal{F}} |R(\boldsymbol{f}) - \hat{R}_N(\boldsymbol{f})| > \epsilon\right\} &\leq 2\mathbb{P}\left\{\sup_{\boldsymbol{f}\in\mathcal{F}_{\mathcal{S},\mathcal{S}'}} |\hat{R}'_N(\boldsymbol{f}) - \hat{R}_N(\boldsymbol{f})| > \frac{\epsilon}{2}\right\} \\ &= 2\mathbb{P}\left\{\bigcup_{\boldsymbol{f}\in\mathcal{F}_{\mathcal{S},\mathcal{S}'}} |\hat{R}'_N(\boldsymbol{f}) - \hat{R}_N(\boldsymbol{f})| > \epsilon\right\}\end{aligned}$$

$$\leq 2G_{\mathcal{F}}(2N) \cdot \mathbb{P}\left\{|\hat{R}'_N(\boldsymbol{f}) - \hat{R}_N(\boldsymbol{f})| > \epsilon\right\}$$
$$\leq 2G_{\mathcal{F}}(2N) \exp(-N\epsilon^2/8),$$

where the last inequality is obtained by Hoeffding's inequality and $\mathcal{F}_{\mathcal{S},\mathcal{S}'}$ denotes the restriction of the hypothesis class to the empirical distribution for $\mathcal{S}, \mathcal{S}'$. Here, $G_{\mathcal{F}}(\cdot)$ is called the *growth function* defined by

$$G_{\mathcal{F}}(2N) := |\mathcal{F}_{\mathcal{S},\mathcal{S}'}|, \tag{12.13}$$

which represents the number of the most possible sets of dichotomies using the hypothesis class $\mathcal{F}$ on any $2N$ points from $\mathcal{S}$ and $\mathcal{S}'$.

The discovery of the growth function is one of the important contributions of Vapnik and Chervonenkis [10]. This is closely related to the concept of *shattering*, which is formally defined as follows.

Definition 12.2 (Shattering) We say $\mathcal{F}$ shatters $\mathcal{S}$ if $|\mathcal{F}| = 2^{|\mathcal{S}|}$.

In fact, the growth function $G_{\mathcal{F}}(N)$ is often called the *shattering number*: the number of the most possible sets of dichotomies using the hypothesis class $\mathcal{F}$ on any N points. Below, we show several facts for the growth function:

- By definition, the shattering number satisfies $G_{\mathcal{F}}(N) \leq 2^N$.
- When $\mathcal{F}$ is finite, we always have $G_{\mathcal{F}}(N) = |\mathcal{F}|$.
- If $G_{\mathcal{F}}(N) = 2^N$, then there is a set of N points such that the class of functions $\mathcal{F}$ can generate any possible classification result on these points. Figure 12.2 shows such a case where $\mathcal{F}$ is the class of linear classifiers.

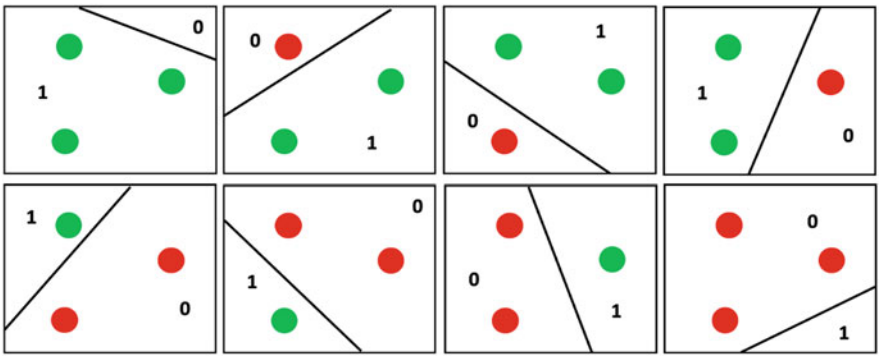

Fig. 12.2 Most possible sets of dichotomies using linear classifier on any three points. The resulting shattering number is $G_{\mathcal{F}}(3) = 8$

Accordingly, we arrive at the following classical VC bound [10]:

Theorem 12.2 (VC Bound) *For any $\delta > 0$, with probability at least $1-\delta$, we have*

$$R(\boldsymbol{f}) \leq \hat{R}_N(\boldsymbol{f}) + \sqrt{\frac{8 \ln G_{\mathcal{F}}(2N) + 8 \ln \frac{2}{\delta}}{N}}. \tag{12.14}$$

Another important contribution of the work by Vapnik and Chervonenkis [10] is that the growth function can be bounded by the so-called *VC dimension*, and the number of data points for which we cannot get all possible dichotomies (=VC dimension +1) is called the *break point.*

Definition 12.3 (VC Dimension) The VC dimension of a hypothesis class $\mathcal{F}$ is the largest $N = d_{VC}(\mathcal{F})$ such that

$$G_{\mathcal{F}}(N) = 2^N.$$

In other words, the VC dimension of a function class $\mathcal{F}$ is the cardinality of the largest set that it can shatter.

This means that the VC dimension is a measure of the capacity (complexity, expressiveness, richness, or flexibility) of a set of functions that can be learned from a statistical binary classification algorithm. It is defined as the cardinality of the largest number of points that the algorithm can classify with zero training error. In the following, we show several examples where we can explicitly calculate the VC dimensions.

Example: Half-Sided Interval
Consider any function of the form $\mathcal{F} = \{f(x) = \chi(x \leq \theta), \theta \in \mathbb{R}\}$. It can shatter two points, but any three points cannot be shattered. Therefore, $d_{VC}(\mathcal{F}) = 2$.

Example: Half Plane
Consider a hypothesis class $\mathcal{F}$ composed of half planes in $\mathbb{R}^d$. It can shatter $d+1$ points, but any $d+2$ points cannot be shattered. Therefore, $d_{VC}(\mathcal{F}) = d+1$.

Example: Sinusoids
f is a single-parametric sine classifier, i.e, for a certain parameter θ, the classifier f_θ returns 1 if the input number x is larger than $\sin(\theta x)$ and 0 otherwise. The VC dimension of f is infinite, since it can shatter any finite subset of the set $\{2^{-m} \mid m \in \mathbb{N}\}$.

Finally, we can derive the generalization bound using the VC dimension. For this, the following lemma by Sauer is the key element.

Lemma 12.2 (Sauer's Lemma[158]) *Suppose that $\mathcal{F}$ has a finite VC dimension d_{VC}. Then*

$$G_{\mathcal{F}}(n) \leq \sum_{i=1}^{d_{VC}} \binom{n}{i} \tag{12.15}$$

and for all $n \geq d_{VC}$,

$$G_{\mathcal{F}}(n) \leq \left(\frac{en}{d_{VC}}\right)^{d_{VC}}. \tag{12.16}$$

Corollary 12.1 (VC Bound Using VC Dimension) *Let $d_{VC} \geq N$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, we have*

$$R(\boldsymbol{f}) \leq \hat{R}_N(\boldsymbol{f}) + \sqrt{\frac{8 d_{VC} \ln \frac{2eN}{d_{VC}} + 8 \ln \frac{2}{\delta}}{N}}. \tag{12.17}$$

Proof This is a direct consequence of Theorem 12.2 and Lemma 12.2. □

The VC dimension has been studied for deep neural networks to understand their generalization behaviors [159]. Bartlett et al. [160] proves bounds on the VC dimension of piece-wise linear networks with potential weight sharing. Although this measure could be predictive when the architecture changes, which happens only in depth and width hyperparameter types, the authors in [159] also found that it is negatively correlated with the generalization gap, which contradicts the widely known empirical observation that over-parametrization improves generalization in deep learning [159].

12.2.2 Rademacher Complexity Bounds

Another important classical approach for the generalization error bound is Rademacher complexity [161]. To understand this concept, consider the following toy example. Let $\mathcal{S} := \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ denote the training sample set, where $y_n \in \{-1, 1\}$. Then, the training error can be computed by

$$err_N(f) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\left[f(\boldsymbol{x}_n) \neq y_n\right], \tag{12.18}$$

where $\mathbf{1}[\cdot]$ is an indicator function computed by

$$\mathbf{1}\left[f(\boldsymbol{x}_n) \neq y_n\right] = \begin{cases} 1, & \{f(\boldsymbol{x}_n), y_n\} = \{1, -1\}, \{-1, 1\} \\ 0, & \{f(\boldsymbol{x}_n), y_n\} = \{1, 1\}, \{-1, -1\} \end{cases}. \tag{12.19}$$

Then, (12.18) can be equivalently represented by

$$\begin{aligned} err_N(f) &= \frac{1}{N}\sum_{n=1}^{N} \frac{1 - y_n f(\boldsymbol{x}_n)}{2} \\ &= \frac{1}{2} - \underbrace{\frac{1}{N}\sum_{i=1}^{N} y_n f(\boldsymbol{x}_n)}_{\text{correlation}}. \end{aligned} \tag{12.20}$$

Therefore, minimizing the training error is equivalent to maximizing the correlation. Now, the core idea of the Rademacher complexity is to consider a game where a player generates random targets $\{y_n\}_{n=1}^N$ and another player provides the hypothesis that maximize the correlation:

$$\sup_{f \in \mathcal{F}} \frac{1}{N}\sum_{n=1}^{N} y_n f(\boldsymbol{x}_n). \tag{12.21}$$

Note that the idea is closely related to the shattering in VC analysis. Specifically, if the hypothesis class $\mathcal{F}$ shatters $\boldsymbol{S} = \{\boldsymbol{x}_n, y_n\}_{n=1}^N$, then the correlation becomes a maximum. However, in contrast to the VC analysis that considers the worst-case scenario, Rademacher complexity analysis deals with average-case analysis. Formally, we define the so-called Rademacher complexity [161].

Definition 12.4 (Rademacher Complexity[161]) Let $\sigma_1 \cdots, \sigma_N$ be independent random variables $\mathbb{P}\{\sigma_n = 1\} = \mathbb{P}\{\sigma_= - 1\} = \frac{1}{2}$. Then, the empirical Rademacher complexity of $\mathcal{F}$ is defined by

$$Rad_N(\mathcal{F}, \mathcal{S}) = \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{n=1}^{N} \sigma_n f(\boldsymbol{x}_n) \right], \tag{12.22}$$

where $\boldsymbol{\sigma} = [\sigma_1, \cdots, \sigma_N]^\top$. In addition, the general notion of Rademacher complexity is computed by

$$Rad_N(\mathcal{F}) := \mathbb{E}_{\mathcal{S}} \left[Rad_N(\mathcal{F}, \mathcal{S}) \right]. \tag{12.23}$$

Another important advantage of Rademacher complexity is that it can be easily generalized to the regression problem for the vector target. For example, (12.23) can be generalized as follows:

$$Rad_N(\mathcal{F}) = \mathbb{E} \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{n=1}^{N} \langle \boldsymbol{\sigma}_n, \boldsymbol{f}(\boldsymbol{x}_n) \rangle \right], \tag{12.24}$$

where $\{\boldsymbol{\sigma}_n\}_{n=1}^N$ refers to the independent random vectors. In the following, we provide some examples where the Rademacher complexity can be explicitly calculated.

Example: Minimum Rademacher Complexity
When the hypothesis class has one element, i.e. $|\mathcal{F}| = 1$, we have

$$Rad(\mathcal{F}) = \mathbb{E} \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{n=1}^{N} \sigma_n f(\boldsymbol{x}_n) \right] = f(\boldsymbol{x}_1) \cdot \mathbb{E} \left[\frac{1}{N} \sum_{n=1}^{N} \sigma_n \right] = 0,$$

where the second equality comes from the fact that $f(\boldsymbol{x}_n) = f(\boldsymbol{x}_1)$ for all n when $|\mathcal{F}| = 1$. The final equation comes from the definition of the random variable σ_n.

Example: Maximum Rademacher Complexity
When $|\mathcal{F}| = 2^N$, we have

$$Rad(\mathcal{F}) = \mathbb{E} \left[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{n=1}^{N} \sigma_n f(\boldsymbol{x}_n) \right] = \mathbb{E} \left[\frac{1}{N} \sum_{n=1}^{N} \sigma_n^2 \right] = 1,$$

(continued)

where the second equality comes from the fact that we can find a hypothesis such that $f(\boldsymbol{x}_n) = \sigma_n$ for all n. The final equation comes from the definition of the random variable σ_n.

Although the Rademacher complexity was originally derived above for the binary classifiers, it can also be used to evaluate the complexity of the regression. The following example shows that a closed form Rademacher complexity can be obtained for ridge regression.

Example: Ridge Regression
Let $\mathcal{F}$ be the class of linear predictors given by $y = \boldsymbol{w}^\top \boldsymbol{x}$ with the restriction of $\|\boldsymbol{w}\| \leq W$ and $\|\boldsymbol{x}\| \leq X$. Then, we have

$$\begin{aligned} Rad(\mathcal{F}, \mathcal{S}) &= \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{w}:\|\boldsymbol{w}\| \leq W} \frac{1}{N} \sum_{n=1}^{N} \sigma_n \boldsymbol{w}^\top \boldsymbol{x}_n \right] \\ &= \frac{1}{N} \mathbb{E}_{\boldsymbol{\sigma}} \left[\sup_{\boldsymbol{w}:\|\boldsymbol{w}\| \leq W} \boldsymbol{w}^\top \left(\sum_{n=1}^{N} \sigma_n \boldsymbol{x}_n \right) \right] \\ &\overset{(a)}{=} \frac{W}{N} \mathbb{E}_{\boldsymbol{\sigma}} \left\| \sum_{n=1}^{N} \sigma_n \boldsymbol{x}_n \right\| \overset{(b)}{\leq} \frac{W}{N} \sqrt{\sum_{n=1}^{N} \mathbb{E}_{\boldsymbol{\sigma}} \|\sigma_n \boldsymbol{x}_n\|^2} \\ &= \frac{W}{N} \sqrt{\sum_{n=1}^{N} \|\boldsymbol{x}_n\|^2} \leq \frac{WX}{\sqrt{N}}, \end{aligned}$$

where (a) comes from the definition of the l_1 norm, and (b) comes from Jensen's inequality.

Using the Rademacher complexity, we can now derive a new type of generalization bound. First, we need the following concentration inequality.

Lemma 12.3 (McDiarmid's Inequality[161]) *Let $x_1, \cdots, x_N$ be independent random variables taking on values in a set $\mathcal{X}$ and let $c_1, \cdots, c_n$ be positive real constants. If $\varphi : \mathcal{X}^N \mapsto \mathbb{R}$ satisfies*

$$\sup_{x_1, \cdots, x_N, x_n' \in \mathcal{A}} |\varphi(x_1, \cdots, x_n, \cdots, x_N) - \varphi(x_1, \cdots, x_n', \cdots, x_N)| \leq c_n,$$

for $1 \leq n \leq N$, *then*

$$\mathbb{P}\{|\varphi(x_1, \cdots, x_N) - \mathbb{E}\varphi(x_1, \cdots, x_N)| \geq \epsilon\} \leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{n=1}^{N} c_n^2}\right). \tag{12.25}$$

In particular, if $\varphi(x_1, \cdots, x_N) = \sum_{n=1}^{N} x_n/N$, *the inequality* (12.25) *reduces to Hoeffding's inequality.*

Using McDiarmid's inequality and symmetrization using "ghost samples", we can obtain the following generalization bound.

Theorem 12.3 (Rademacher Bound) *Let* $\mathcal{S} := \{\boldsymbol{x}_n, y_n\}_{n=1}^{N}$ *denote the training set and* $f(\boldsymbol{x}) \in [a, b]$. *For any* $\delta > 0$, *with probability at least* $1 - \delta$, *we have*

$$R(f) \leq \hat{R}_N(f) + 2Rad_N(\mathcal{F}) + (b-a)\sqrt{\frac{\ln 1/\delta}{2N}}, \tag{12.26}$$

and

$$R(f) \leq \hat{R}_N(f) + 2Rad_N(\mathcal{F}, \mathcal{S}) + 3(b-a)\sqrt{\frac{\ln 2/\delta}{2N}}. \tag{12.27}$$

Unfortunately, many theoretical efforts using the Rademacher complexity to understand the deep neural network were not successful [159], which often resulted in a vacuous bound similar to the attempts using VC bounds. Therefore, the need to obtain a tighter bound has been increasing.

12.2.3 PAC–Bayes Bounds

So far, we have discussed performance guarantees which hold whenever the training and test data are drawn independently from an identical distribution. In fact, learning algorithms that satisfy such performance guarantees are called the probably approximately correct (PAC) learning [156]. It was shown that the concept class C is PAC learnable if and only if the VC dimension of C is finite [162].

In addition to PAC learning, there is another important area of modern learning theory—Bayesian inference. Bayesian inferences apply whenever the training and test data are generated according to the specified prior. However, there is no guarantee of an experimental environment in which training and test data are generated according to a different probability distribution than the previous one. In fact, much of modern learning theory can be broken down into Bayesian inference and PAC learning. Both areas investigate learning algorithms that use training data

as the input and generate a concept or model as the output, which can then be tested on test data.

The difference between the two approaches can be seen as a trade-off between generality and performance. We define an "experimental setting" as a probability distribution over training and test data. A PAC performance guarantee applies to a wide class of experimental settings. A Bayesian correctness theorem applies only to experimental settings that match those previously used in the algorithm. In this restricted class of settings, however, the Bayesian learning algorithm can be optimal and generally outperforms the PAC learning algorithms.

The PAC–Bayesian theory combines Bayesian and frequentist approaches [163]. The PAC–Bayesian theory is based on a prior probability distribution concerning the "situation" occurring in nature, and a "rule" expresses a learner's preference for some rules over others. There is no supposed relationship between the learner's bias for rules and the nature distribution. This differs from the Bayesian inference, where the starting point is a common distribution of rules and situations, which induces a conditional distribution of rules in certain situations.

Under this set-up, the following PAC–Bayes generalization bound can be obtained.

Theorem 12.4 (PAC–Bayes Generalization Bound) *[163] Let Q be an arbitrary distribution over $z := (x, y) \in \mathcal{Z} := \mathcal{X} \times \mathcal{Y}$. Let $\mathcal{F}$ be a hypothesis class and let ℓ be a loss function such that for all f and z we have $\ell(f, z) \in [0, 1]$. Let $\mathcal{P}$ be a prior distribution over $\mathcal{F}$ and let $\delta \in (0, 1)$. Then, with probability of at least $1 - \delta$ over the choice of an i.i.d. training set $\mathcal{S} := \{z_n\}_{n=1}^N$ sampled according to Q, for all distributions Q over $\mathcal{F}$ (even such that depend on $\mathcal{S}$), we have*

$$\mathbb{E}_{f\sim Q}[R(f)] \leq \mathbb{E}_{f\sim Q}\left[\hat{R}_N(f)\right] + \sqrt{\frac{KL(Q||\mathcal{P}) + \ln N/\delta}{2(N-1)}}, \tag{12.28}$$

where

$$KL(Q||\mathcal{P}) := \mathbb{E}_{f\sim Q}[\ln Q(f)/\mathcal{P}(f)] \tag{12.29}$$

is the Kullback–Leibler divergence.

Recently, PAC–Bayes approaches have been studied extensively to explain the generalization capability of neural networks [149, 153, 164]. According to a recent large scale experiment to test the correlation of different measures with the generalization of deep models [159], the authors confirmed the effectiveness of the PAC–Bayesian bounds and corroborate them as a promising direction for cracking the generalization puzzle. Another nice application of PAC–Bayes bounds is that it provides a mean to find the optimal distribution Q^* by minimizing the upper bounds. This technique has been successfully used for the linear classifier design [164], etc.

12.3 Reconciling the Generalization Gap via Double Descent Model

Recall that the following error bound can be obtained for the ERM estimate in (12.2):

$$R(\boldsymbol{f}^*_{ERM}) \leq \underbrace{\hat{R}_N(\boldsymbol{f}^*_{ERM})}_{\text{empirical risk (training error)}} + \underbrace{O\left(\sqrt{\frac{c}{N}}\right)}_{\text{complexity penalty}}, \tag{12.30}$$

where $O(\cdot)$ denotes the "big O" notation and c refers to the model complexity such as VC dimension, Rademacher complexity, etc.

In (12.30), with increasing hypothesis class size $|\mathcal{F}|$, the empirical risk or training error decreases, whereas the complexity penalty increases. The control of the functional class capacity can be therefore done explicitly by choosing $\mathcal{F}$ (e.g. selection of the neural network architecture). This is summarized in the classic U-shaped risk curve, which is shown in Fig. 12.3a and was often used as a guide for model selection. A widely accepted view from this curve is that a model with zero training error is overfitted to the training data and will typically generalize poorly [10]. Classical thinking therefore deals with the search for the "sweet spot" between underfitting and overfitting.

Lately, this view has been challenged by empirical results that seem mysterious. For example, in [165] the authors trained several standard architectures on a copy of the data, with the true labels being replaced by random labels. Their central finding can be summarized as follows: deep neural networks easily fit random labels. More precisely, neural networks achieve zero training errors if they are trained on a completely random labeling of the true data. While this observation is easy to formulate, it has profound implications from a statistical learning perspective: the effective capacity of neural networks is sufficient to store the entire data set. Despite the high capacity of the functional classes and the almost perfect fit to training data, these predictors often give very accurate predictions for new data in the test phase.

These observations rule out VC dimension, Rademacher complexity, etc. from describing the generalization behavior. In particular, the Rademacher complexity for the interpolation regime, which leads to a training error of 0, assumes the maximum value of 1, as previously explained in an example. Therefore, the classic generalization bounds are vacuous and cannot explain the amazing generalization ability of the neural network.

The recent breakthrough in Belkin et al.'s "double descent" risk curve [154, 155] reconciles the classic bias–variance trade-off with behaviors that have been observed in over-parameterized regimes for a large number of machine learning models. In particular, when the functional class capacity is below the "interpolation threshold", learned predictors show the classic U-shaped curve from Fig. 12.3a, where the function class capacity is identified with the number of parameters needed to specify a function within the class. The bottom of the U-shaped risk can be achieved at

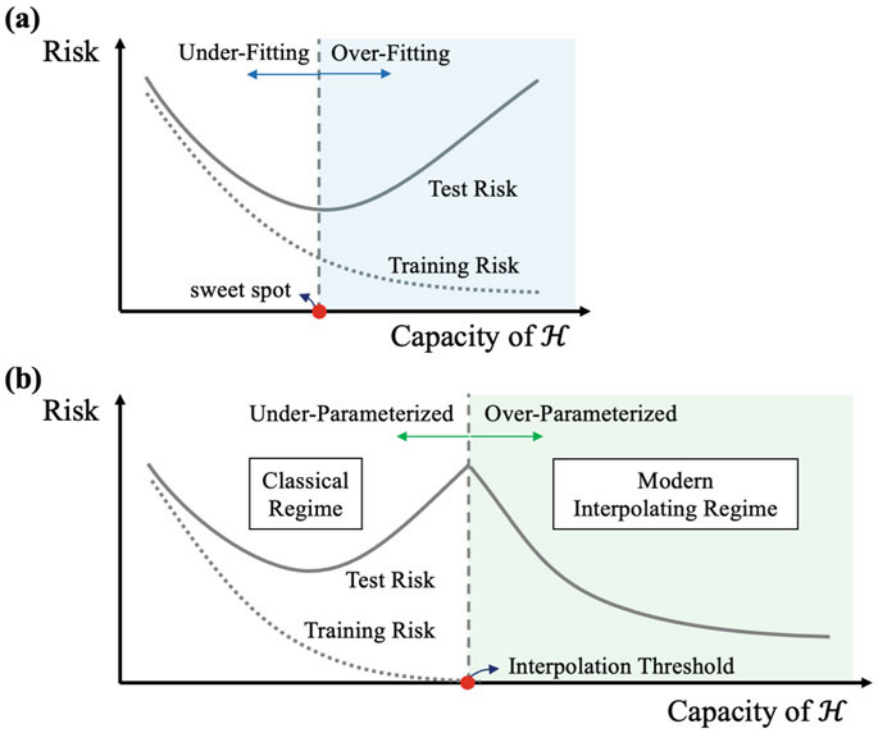

Fig. 12.3 Curves for training risk (dashed line) and test risk (solid line). (**a**) The classical U-shaped risk curve arising from the bias–variance trade-off. (**b**) The double descent risk curve, which incorporates the U-shaped risk curve (i.e., the "classical" regime) together with the observed behavior from using high-capacity function classes (i.e., the "modern" interpolating regime), separated by the interpolation threshold. The predictors to the right of the interpolation threshold have zero training risk

the sweet spot which balances the fit to the training data and the susceptibility to over-fitting. When we increase the function class capacity high enough by increasing the size of the neural network architecture, the learned predictors achieve (almost) perfect fits to the training data. Although the learned predictors obtained at the interpolation threshold typically have high risk, increasing the function class capacity beyond this point leads to decreasing risk, which typically falls below the risk achieved at the sweet spot in the "classic" regime (see Fig. 12.3b).

In the following example we provide concrete and explicit evidence for the double descent behavior in the context of simple linear regression models. The analysis shows the transition from under- to over-parameterized regimes. It also allows us to compare the risks at any point on the curve and explain how the risk in the over-parameterized regime can be lower than any risk in the under-parameterized regime.

Example: Double Descent in Regression [155]
We consider the following linear regression problem:

$$y = \boldsymbol{x}^\top \boldsymbol{\beta} + \epsilon, \tag{12.31}$$

where $\boldsymbol{\beta} \in \mathbb{R}^D$ and $\boldsymbol{x}$ and ϵ are a normal random vector and a variable, where $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}_D)$ and $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Given training data $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, we fit a linear model to the data using only a subset $T \subset [D]$ of cardinality of p, where $[D] := \{0, \cdots, D\}$. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{D \times N}$ be the design matrix, $\boldsymbol{y} = [y_1, \cdots, y_N]^\top$ be the vector of response. For a subset T, we use $\boldsymbol{\beta}_T$ to denote its $|T|$-dimensional subvector of entries from T; we also use $\boldsymbol{X}_T$ to denote an $N \times p$ sub-matrix of $\boldsymbol{X}$ composed of columns in T. Then, the risk of $\hat{\boldsymbol{\beta}}$, where $\hat{\boldsymbol{\beta}}_T = \boldsymbol{X}_T^\dagger \boldsymbol{y}$ and $\hat{\boldsymbol{\beta}}_{T^c} = 0$, is given by

$$\mathbb{E}\left[(y - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}})^2\right] = \begin{cases} (\|\boldsymbol{\beta}_{T^c}\|^2 + \sigma^2)\left(1 + \frac{p}{N-p-1}\right); & if\ p \le N-2 \\ \infty; & if\ N-1 \le p \le N+1 \\ \|\boldsymbol{\beta}_T\|^2\left(1 - \frac{N}{p}\right) & \\ \quad +(\|\boldsymbol{\beta}_{T^c}\|^2 + \sigma^2)\left(1 + \frac{N}{p-N-1}\right); & if\ p \ge N+2. \end{cases} \tag{12.32}$$

Proof Recall that $\boldsymbol{x}$ is assumed to be a Gaussian distribution with zero mean and identity covariance, so that the mean squared prediction error can be written as

$$\begin{aligned} \mathbb{E}\left[(y - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}})^2\right] &= \mathbb{E}\left[(\boldsymbol{x}^\top \boldsymbol{\beta} + \sigma\epsilon - \boldsymbol{x}^\top \hat{\boldsymbol{\beta}})^2\right] = \sigma^2 + \mathbb{E}\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\|^2 \\ &= \sigma^2 + \|\boldsymbol{\beta}_{T^c}\|^2 + \mathbb{E}\|\boldsymbol{\beta}_T - \hat{\boldsymbol{\beta}}_T\|^2, \end{aligned}$$

where $\boldsymbol{\beta}$ denotes the ground-truth regression parameter and we use the independency of the test phase regressor $\boldsymbol{x}$ and the training phase design matrix $\boldsymbol{X}$. Our goal is now to derive the closed form expression for the second term.
(Classical regime) For the given training data set, we have

$$\begin{aligned} \hat{\boldsymbol{\beta}}_T &= (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{X}_T \boldsymbol{y} = (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{X}_T \boldsymbol{X}_T^\top \boldsymbol{\beta}_T + (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{X}_T \boldsymbol{\eta} \\ &= \boldsymbol{\beta}_T + (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{X}_T \boldsymbol{\eta}, \end{aligned}$$

(continued)

where

$$\boldsymbol{\eta} := \boldsymbol{y} - \boldsymbol{X}_T^\top \boldsymbol{\beta}_T = \boldsymbol{\epsilon} + \boldsymbol{X}_{T^c}^\top \boldsymbol{\beta}_{T^c}.$$

By plugging this into the second term, we have

$$\mathbb{E}\|\boldsymbol{\beta}_T - \hat{\boldsymbol{\beta}}_T\|^2 = \mathbb{E}\left[\boldsymbol{\eta}^\top \boldsymbol{P}_{\mathcal{R}(\boldsymbol{X}_T)} \boldsymbol{\eta}\right] = \mathrm{Tr}\left(\mathbb{E}\left[\boldsymbol{P}_{\mathcal{R}(\boldsymbol{X}_T)}\right] \mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^\top\right]\right).$$

In addition, we have

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^\top\right] &= \mathbb{E}\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right] + \mathbb{E}\left[\boldsymbol{X}_{T^c}^\top \boldsymbol{\beta}_{T^c} \left(\boldsymbol{X}_{T^c}^\top \boldsymbol{\beta}_{T^c}\right)^\top\right] \\ &= (\sigma^2 + \|\boldsymbol{\beta}_{T^c}\|^2) \boldsymbol{I}_N,\end{aligned}$$

where $\mathcal{R}(\boldsymbol{X}_T)$ denotes the range space of $\boldsymbol{X}_T$ and $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{X}_T)}$ denotes the projection to the range space of $\boldsymbol{X}_T$. Furthermore, $\boldsymbol{P}_{\mathcal{R}(\boldsymbol{X}_T)}$ is Hotelling's T-squared distribution with parameter p and $N - p + 1$ so that

$$\mathrm{Tr}\mathbb{E}\left[\boldsymbol{P}_{\mathcal{R}(\boldsymbol{X}_T)}\right] = \begin{cases} \frac{p}{N-p-1}, & \text{if } p \leq N-2 \\ +\infty, & \text{if } p = N-1 \end{cases}. \tag{12.33}$$

Therefore, by putting them together we conclude the proof for the classical regime.

(Modern interpolating regime) We consider $p \geq N$. Then, we have

$$\begin{aligned}\hat{\boldsymbol{\beta}}_T &= \boldsymbol{X}_T^\top (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{y} = \boldsymbol{X}_T^\top (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{X}_T^\top \boldsymbol{\beta}_T + \boldsymbol{X}_T^\top (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{\eta} \\ &= \boldsymbol{X}_T^\top (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{X}_T^\top \boldsymbol{\beta}_T + \boldsymbol{X}_T^\top (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{\eta} \\ &= \mathcal{P}_{\mathcal{R}(\boldsymbol{X}_T^\top)} \boldsymbol{\beta}_T + \boldsymbol{X}_T^\top (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{\eta},\end{aligned}$$

where

$$\boldsymbol{\eta} := \boldsymbol{y} - \boldsymbol{X}_T^\top \boldsymbol{\beta}_T = \boldsymbol{\epsilon} + \boldsymbol{X}_{T^c}^\top \boldsymbol{\beta}_{T^c}.$$

Therefore,

$$\mathbb{E}\left[\|\boldsymbol{\beta}_T - \hat{\boldsymbol{\beta}}_T\|^2\right] = \mathbb{E}\left[\|\mathcal{P}^\perp_{\mathcal{R}(\boldsymbol{X}_T^\top)} \boldsymbol{\beta}_T\|^2\right] + \mathbb{E}\left[\boldsymbol{\eta}^\top (\boldsymbol{X}_T \boldsymbol{X}_T^\top)^{-1} \boldsymbol{\eta}\right].$$

(continued)

Furthermore, we have

$$\mathbb{E}\left[\|\mathcal{P}^{\perp}_{\mathcal{R}(\boldsymbol{X}_T^\top)}\boldsymbol{\beta}_T\|^2\right] = \left(1-\frac{n}{p}\right)\|\boldsymbol{\beta}_T\|^2$$
$$\mathbb{E}\left[\boldsymbol{\eta}^\top(\boldsymbol{X}_T\boldsymbol{X}_T^\top)^{-1}\boldsymbol{\eta}\right] = \mathrm{Tr}\left(\mathbb{E}(\boldsymbol{X}_T\boldsymbol{X}_T^\top)^{-1}\mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^\top\right]\right),$$

where we use the independency between $\boldsymbol{X}_T$ and $\boldsymbol{X}_{T^c}$ and $\boldsymbol{\epsilon}$ for the second equality. In addition, we have

$$\begin{aligned}\mathbb{E}\left[\boldsymbol{\eta}\boldsymbol{\eta}^\top\right] &= \mathbb{E}\left[\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top\right] + \mathbb{E}\left[\boldsymbol{X}_{T^c}^\top\boldsymbol{\beta}_{T^c}\left(\boldsymbol{X}_{T^c}^\top\boldsymbol{\beta}_{T^c}\right)^\top\right] \\ &= (\sigma^2 + \|\boldsymbol{\beta}_{T^c}\|^2)\boldsymbol{I}_N.\end{aligned}$$

Finally, the distribution of $\boldsymbol{P} := (\boldsymbol{X}_T\boldsymbol{X}_T^\top)^{-1}$ is inverse-Wishart with identity scale matrix $\boldsymbol{I}_N$ with p degrees of freedom. Accordingly, we have

$$\mathrm{Tr}\left(\mathbb{E}(\boldsymbol{X}_T\boldsymbol{X}_T^\top)^{-1}\right) = \begin{cases}\frac{N}{p-N-1}, & \text{if } p \geq N+2 \\ +\infty, & \text{if } p = N, N+1\end{cases}.$$

By putting them together, we have

$$\mathbb{E}\left[(y-\boldsymbol{x}^\top\hat{\boldsymbol{\beta}})^2\right] = \left(1-\frac{N}{p}\right)\|\boldsymbol{\beta}_T\|^2 + (\sigma^2+\|\boldsymbol{\beta}_{T^c}\|^2)\left(1+\frac{N}{p-N-1}\right),$$

for $p \geq N$ and $\mathbb{E}\left[(y-\boldsymbol{x}^\top\hat{\boldsymbol{\beta}})^2\right] = \infty$ for $p = N, N+1$. This concludes the proof. □

Figure 12.4 illustrates an example plot for the linear regression problem analyzed above for a particular parameter set.

12.4 Inductive Bias of Optimization

All learned predictors to the right of the interpolation threshold fit perfectly with the training data and have no empirical risk. Then, why should some—especially those from larger functional classes—have a lower test risk than others so that they generalize better? The answer is that the functional class capacity, such as VC dimension, or Rademacher complexity, does not necessarily reflect the inductive bias of the predictor appropriate for the problem at hand. Indeed, one

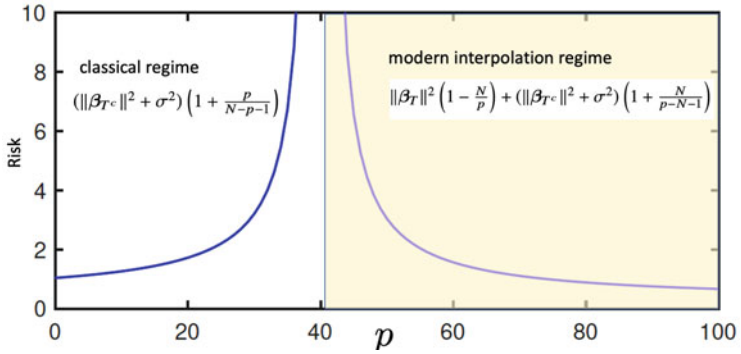

Fig. 12.4 Plot of the risk in (12.32) as a function of p under the random selection of T. Here $\|\boldsymbol{\beta}\|^2 = 1$, $\sigma^2 = 1/25$ and $N = 40$

of the underlying reasons for the appearance of the double descent model in the previous linear regression problem is that we impose an *inductive bias* to choose the minimum norm solution $\hat{\boldsymbol{\beta}}_T = \boldsymbol{X}_T(\boldsymbol{X}_T^\top \boldsymbol{X}_T)^{-1}\boldsymbol{y}$ for the over-parameterized regime, which leads to the smooth solution.

Among the various interpolation solutions, choosing the smooth or simple function that perfectly fits the observed data is a form of Occam's razor: the simplest explanation compatible with the observations should be preferred. By considering larger functional classes that contain more candidate predictors that are compatible with the data, we can find interpolation functions that are "simpler". Increasing the functional class capacity thus improves the performance of classifiers. One of the important advantages of choosing a simpler solution is that it is easy to generalize by avoiding unnecessary glitches in the data. Increasing the functional class capacity to the over-parameterized area thus improves the performance of the resulting classifiers.

Then, one of the remaining questions is: what is the underlying mechanism by which a trained network becomes smooth or simple? This is closely related to the inductive bias (or implicit bias) of an optimization algorithm such as gradient descent, stochastic gradient descent (SGD), etc. [166–171]. Indeed, this is an active area of research. For example, the authors in [168] show that the gradient descent for the linear classifier for specific loss function leads to the maximum margin SVM classifier. Other researchers have shown that the gradient descent in deep neural network training leads to a simple solution [169–171].

12.5 Generalization Bounds via Algorithm Robustness

Another important question is how we can quantify the inductive bias of the algorithm in terms of a generalization error bound. In this section, we introduce a notion of *algorithmic robustness* for quantifying the generalization error, which

was originally proposed in [172], but has been largely neglected in deep learning research. It turns out that the generalization bound based on algorithmic robustness has all the ingredients to quantify the fascinating generalization behavior of the deep neural network, so it can be a useful tool for studying generalization.

Recall that the underlying assumption for the classical generalization bounds is the uniform convergence of empirical quantities to their mean [10], which provides ways to bound the gap between the expected risk and the empirical risk by the complexity of the hypothesis set. On the other hand, robustness requires that a prediction rule has comparable performance if tested on a sample close to a training sample. This is formally defined as follows.

Definition 12.5 (Algorithm Robustness [172]) Algorithm $\mathcal{A}$ is said to be $(K, \epsilon(\cdot))$-robust for $K \in \mathbb{N}$ and $\epsilon(\cdot) : \mathcal{Z} \mapsto \mathbb{R}$, if $\mathcal{Z} := \mathcal{X} \times \mathcal{Y}$ can be partitioned into K disjoint sets, denoted by $\{C_i\}_{i=1}^{K}$ such that the following holds for all training sets $\mathcal{S} \subset \mathcal{Z}$:

$$\forall \boldsymbol{s} \in \mathcal{S}, \forall \boldsymbol{z} \in \mathcal{Z}; \text{if } \boldsymbol{s}, \boldsymbol{z} \in C_i, \text{ then } \quad |\ell(\mathcal{A}_{\mathcal{S}}, \boldsymbol{s}) - \ell(\mathcal{A}_{\mathcal{S}}, \boldsymbol{z})| \leq \epsilon(\mathcal{S}) \tag{12.34}$$

for all $i = 1, \cdots, K$, where $\mathcal{A}_{\mathcal{S}}$ denotes the algorithm $\mathcal{A}$ trained with the data set $\mathcal{S}$.

Then, we can obtain the generalization bound based on algorithmic robustness. First, we need the following concentration inequality.

Lemma 12.4 (Breteganolle–Huber–Carol Inequality [173]) *If the random vector $(N_1, \cdots, N_k)$ is multinomially distributed with parameters N and $(p_1, \cdots, p_k)$, then*

$$\mathbb{P}\left\{\sum_{i=1}^{k} |N_i - Np_i| \geq 2\sqrt{N}\lambda\right\} \leq 2^k \exp(-2\lambda^2), \quad \lambda > 0. \tag{12.35}$$

Theorem 12.5 *If a learning algorithm $\mathcal{A}$ is $(K, \epsilon(\cdot))$-robust, and the training sample set $\mathcal{S}$ is generated by N i.i.d samples from the probability measure μ, then for any $\delta > 0$, with probability at least $1 - \delta$ we have*

$$|R(\mathcal{A}_{\mathcal{S}}) - \hat{R}_N(\mathcal{A}_{\mathcal{S}})| \leq \epsilon(\mathcal{S}) + M\sqrt{\frac{2K \ln 2 + 2\ln(1/\delta)}{N}}, \tag{12.36}$$

where

$$M := \max_{z \in \mathcal{Z}} |\ell(\mathcal{A}_{\mathcal{S}}, z)|.$$

Proof Let N_i be the set of indices of points of $\mathcal{S}$ that fall into the C_i. Note that $(|N_1|, \cdots, |N_K|)$ is an i.i.d. multinomial random variable with parameters N and $(\mu(C_i), \cdots, \mu(C_K))$. Then, the following holds by Lemma 12.4.

$$\mathbb{P}\left\{\sum_{i=1}^{K}\left|\frac{|N_i|}{N}-\mu(C_i)\right| \geq \lambda\right\} \leq 2^K \exp\left(-\frac{N\lambda^2}{2}\right). \tag{12.37}$$

Hence, the following holds with probability at least $1-\delta$,

$$\sum_{i=1}^{K}\left|\frac{|N_i|}{N}-\mu(C_i)\right| \leq \sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{N}}. \tag{12.38}$$

The generalization error is then given by

$$\begin{aligned}
|R(\mathcal{A}_\mathcal{S}) - \hat{R}_N(\mathcal{A}_\mathcal{S})| &\leq \left|\sum_{i=1}^{K}\mathbb{E}_{z\sim\mu}\ell(\mathcal{A}_\mathcal{S}, z|z\in C_i)\mu(C_i) - \frac{1}{N}\sum_{n=1}^{N}\ell(\mathcal{A}_\mathcal{S}, \boldsymbol{s}_i)\right| \\
&\overset{(a)}{\leq} \left|\sum_{i=1}^{K}\mathbb{E}_{z\sim\mu}\ell(\mathcal{A}_\mathcal{S}, z|z\in C_i)\frac{|N_i|}{N} - \frac{1}{N}\sum_{n=1}^{N}\ell(\mathcal{A}_\mathcal{S}, \boldsymbol{s}_i)\right| \\
&\quad + \left|\sum_{i=1}^{K}\mathbb{E}_{z\sim\mu}\ell(\mathcal{A}_\mathcal{S}, z|z\in C_i)\mu(C_i) - \sum_{n=1}^{N}\mathbb{E}_{z\sim\mu}\ell(\mathcal{A}_\mathcal{S}, z|z\in C_i)\frac{|N_i|}{N}\right| \\
&\overset{(b)}{\leq} \frac{1}{N}\left|\sum_{i=1}^{K}\sum_{j\in N_i}\max_{z_2\in C_j}|\ell(\mathcal{A}_\mathcal{S}, \boldsymbol{s}_j) - \ell(\mathcal{A}_\mathcal{S}, z_2)|\right| \\
&\quad + \max_{z\in\mathcal{Z}}|\ell(\mathcal{A}_\mathcal{S}, z)|\sum_{i=1}^{K}\left|\frac{|N_i|}{N}-\mu(C_i)\right| \\
&\overset{(c)}{\leq} \epsilon(\mathcal{S}) + M\sum_{i=1}^{K}\left|\frac{|N_i|}{N}-\mu(C_i)\right| \\
&\overset{(d)}{\leq} \epsilon(\mathcal{S}) + M\sqrt{\frac{2K\ln 2 + 2\ln(1/\delta)}{N}},
\end{aligned}$$

where (a), (b), and (c) are due to the triangle inequality, the definition of N_i, and the definition of $\epsilon(\mathcal{S})$ and M, respectively. □

Note that the definition of robustness requires that (12.34) holds for every training sample. The parameters K and $\epsilon(\cdot)$ quantify the robustness of an algorithm. Since $\epsilon(\cdot)$ is a function of training samples, an algorithm can have different robustness properties for different training patterns. For example, a classification algorithm is more robust to a training set with a larger margin. Since (12.34) includes both the trained solution $\mathcal{A}_\mathcal{S}$ and the training set $\mathcal{S}$, robustness is a property of the learning

algorithm, rather than the property of the "effective hypothesis space". This is why the robustness-based generalization bound can account for the inductive bias from the algorithm.

For example, for the case of a single-layer ReLU neural network $f_\Theta : \mathbb{R}^2 \to \mathbb{R}^2$ with the following weight matrix and bias:

$$W^{(0)} = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}, \; b^{(0)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

the corresponding neural network output is given by

$$\boldsymbol{o}^{(1)} = \begin{cases} [0, 0]^\top, & 2x - y + 1 < 0, x + y - 1 < 0, \\ [2x - y + 1, 0]^\top, & 2x - y + 1 \geq 0, x + y - 1 < 0, \\ [0, x + y - 1]^\top, & 2x - y + 1 < 0, x + y - 1 \geq 0, \\ [2x - y + 1, x + y - 1]^\top, & 2x - y + 1 \geq 0, x + y - 1 \geq 0. \end{cases}$$

Here, the number of partitions is $K = 4$.

On the other hand, consider a two-layer ReLU network with the weight matrices and biases given by

$$W^{(0)} = \begin{bmatrix} 2 & -1 \\ 1 & 1 \end{bmatrix}, \; b^{(0)} = \begin{bmatrix} 1 \\ -1 \end{bmatrix},$$
$$W^{(1)} = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}, \; b^{(1)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The corresponding neural network output is given by

$$\boldsymbol{o}^{(2)} = \begin{cases} [0, 1]^\top, & 2x - y + 1 < 0, x + y - 1 < 0, \\ [2x - y + 1, 1]^\top, & 2x - y + 1 \geq 0, x + y - 1 < 0, \\ [2x + 2y - 2, x + y]^\top, & 2x - y + 1 < 0, x + y - 1 \geq 0, \\ [4x + y - 1, x + y]^\top, & 2x - y + 1 \geq 0, x + y - 1 \geq 0. \end{cases}$$

Therefore, in spite of the twice larger parameter sizes, the number of partitions is $K = 4$, which is the same as the single-layer neural network. Therefore, in terms of the generalization bounds, the two algorithms have same upper bound up to the parameter $\epsilon(\mathcal{S})$. This example clearly confirms that generalization is a property of the learning algorithm, rather than the property of the effective hypothesis space or the number of parameters.

12.6 Exercises

1. Compute the VC dimension of the following function classes:
   (a) Interval $[a, b]$.
   (b) Disc in $\mathbb{R}^2$.
   (c) Half space in $\mathbb{R}^d$.
   (d) Axis-aligned rectangles.
2. Show that the classifier f_θ that returns 1 if the input number x is larger than $\sin(\theta x)$ and 0 otherwise can shatter any finite subset of the set $\{2^{-m} \mid m \in \mathbb{N}\}$.
3. Prove the following properties of Rademacher complexity:
   (a) (Monotonicity) If $\mathcal{F} \subset \mathcal{G}$, then $Rad_N(\mathcal{F}) \leq Rad_N(\mathcal{G})$.
   (b) (Convex hull) Let $conv(\mathcal{F})$ be the convex hull of $\mathcal{F}$. Then $Rad_N(\mathcal{F}) = Rad_N(conv(\mathcal{F}))$.
   (c) (Scale and shift) For any function class $\mathcal{F}$ and $c, d \in \mathbb{R}$. $Rad_N(c\mathcal{F} + d) = |c| Rad_N(\mathcal{F})$.
   (d) (Lipschitz composition) If ϕ is an L-Lipschitz function, then $Rad_N(\phi \cdot \mathcal{F}) \leq L \cdot Rad_N(\mathcal{F})$.
4. Let $\mathcal{F}$ be the class of linear predictors given by $y = \boldsymbol{w}^\top \boldsymbol{x}$ with the restriction of $\|\boldsymbol{w}\|_1 \leq W_1$ and $\|\boldsymbol{x}\|_\infty \leq X_\infty$ for $\boldsymbol{x} \in \mathbb{R}^d$. Then, show that
$$Rad_N(\mathcal{F}) \leq \frac{W_1 X_\infty \sqrt{2 \ln(d)}}{\sqrt{N}}.$$
5. Let $\mathcal{A}$ be a set of N vectors in $\mathbb{R}^m$, and let $\bar{\boldsymbol{a}}$ be the mean of the vectors in $\mathcal{A}$. Then:
$$Rad_N(\mathcal{A}) \leq \max_{\boldsymbol{a} \in \mathcal{A}} \|\boldsymbol{a} - \bar{\boldsymbol{a}}\|_2 \cdot \frac{\sqrt{2 \log N}}{m}.$$
   In particular, if $\mathcal{A}$ is a set of binary vectors,
$$Rad_N(\mathcal{A}) \leq \sqrt{\frac{2 \log N}{m}}.$$
6. For a metric space $\mathcal{S}, \rho$ and $\mathcal{T} \subset \mathcal{S}$ we say that $\hat{\mathcal{T}} \subset \mathcal{S}$ is an ϵ-cover of $\mathcal{T}$, if $\forall t \in \mathcal{T}$, there exists $t' \in \mathcal{T}$ such that $\rho(t, t') \leq \epsilon$. The ϵ-covering number of $\mathcal{T}$ is defined by
$$N(\epsilon, \mathcal{T}, \rho) = \min\{|\mathcal{T}'| : \mathcal{T}' \text{ is an } \epsilon\text{-cover of } \mathcal{T}\}.$$

If $\mathcal{Z}$ is compact w.r.t. metric ρ, $\ell(\mathcal{A}_{\mathcal{S}}, \cdot)$ is Lipschitz continuous with Lipschitz constant $c(\mathcal{S})$, i.e.,

$$|\ell(\mathcal{A}_{\mathcal{S}}, z_1) - \ell(\mathcal{A}_{\mathcal{S}}, z_2)| \leq c(\mathcal{S})\rho(z_1, z_2), \quad \forall z_1, z_2 \in \mathcal{Z},$$

then show that $\mathcal{A}$ is $(K, \epsilon(\mathcal{S}))$-robust, where

$$K = N(\gamma/2, \mathcal{Z}, \rho), \ \ \epsilon(\mathcal{S}) = c(\mathcal{S})\gamma$$

for $\gamma > 0$.