Chapter 14

Generalization properties of deep neural networks

As discussed in the introduction in Section 1.2, we generally learn based on a finite data set. For example, given data $(x_i, y_i)_{i=1}^m$, we try to find a network Φ that satisfies $\Phi(x_i) = y_i$ for $i = 1, \dots, m$. The field of generalization is concerened with how well such Φ performs on *unseen* data, which refers to any x outside of training data $\{x_1, \dots, x_m\}$. In this chapter we discuss generalization through the use of covering numbers.

In Sections 14.1 and 14.2 we revisit and formalize the general setup of learning and empirical risk minimization in a general context. Although some notions introduced in these sections have already appeared in the previous chapters, we reintroduce them here for a more coherent presentation. In Sections 14.3-14.5, we first discuss the concepts of generalization bounds and covering numbers, and then apply these arguments specifically to neural networks. In Section 14.6 we explore the so-called "approximation-complexity trade-off", and finally in Sections 14.7-14.8 we introduce the "VC dimension" and give some implications for classes of neural networks.

14.1 Learning setup

A general learning problem [146, 211, 42] requires a **feature space** X and a **label space** Y, which we assume throughout to be measurable spaces. We observe joint data pairs $(x_i, y_i)_{i=1}^m \subseteq X \times Y$, and aim to identify a connection between the x and y variables. Specifically, we assume a relationship between features x and labels y modeled by a probability distribution $\mathcal{D}$ over $X \times Y$, that generated the observed data $(x_i, y_i)_{i=1}^m$. While this distribution is unknown, our goal is to extract information from it, so that we can make possibly good predictions of y for a given x. Importantly, the relationship between x and y need not be deterministic.

To make these concepts more concrete, we next present an example that will serve as the running example throughout this chapter. This example is of high relevance for many mathematicians, as ensuring a steady supply of high-quality coffee is essential for maximizing the output of our mathematical work.

Example 14.1 (Coffee Quality)**.** Our goal is to determine the quality of different coffees. To this end we model the quality as a number in

$$Y = \Big\{ \frac{0}{10}, \dots, \frac{10}{10} \Big\},$$

Acidity	Caffeine (mg/100ml)	Price	Aftertaste	Roast level	Origin	Quality
7/10	41	5	dry	8/10	Ethiopia	**7/10**
5/10	40	7	lingering	7/10	Brazil	**5/10**
5/10	39	6,5	dry	7/10	Columbia	**6/10**
6/10	39,5	3	sweet/floral	5/10	Vietnam	**6/10**
9/10	40	9	bitter	9/10	Brazil	**8/10**
2/10	40,3	6,2	bitter	8/10	Ethiopia	**9/10**
6/10	39,2	8	fruity	7/10	Brazil	**???**

Figure 14.1: Collection of coffee data. The last row lacks a "Quality" label. Our aim is to predict the label without the need for an (expensive) taste test.

with higher numbers indicating better quality. Let us assume that our subjective assessment of quality of coffee is related to six features: "Acidity", "Caffeine content", "Price", "Aftertaste", "Roast level", and "Origin". The feature space X thus corresponds to the set of six-tuples describing these attributes, which can be either numeric or categorical (see Figure 14.1).

We aim to understand the relationship between elements of X and elements of Y, but we can neither afford, nor do we have the time to taste all the coffees in the world. Instead, we can sample some coffees, taste them, and grow our database accordingly as depicted in Figure 14.1. This way we obtain samples of pairs in $X \times Y$. The distribution $\mathcal{D}$ from which they are drawn depends on various external factors. For instance, we might have avoided particularly cheap coffees, believing them to be inferior. As a result they do not occur in our database. Moreover, if a colleague contributes to our database, he might have tried the same brand and arrived at a different rating. In this case, the quality label is not deterministic anymore.

Based on our database, we wish to predict the quality of an untasted coffee. Before proceeding, we first formalize what it means to be a "good" prediction.

Characterizing how good a predictor is requires a notion of discrepancy in the label space. This is the purpose of the so-called **loss function**, which is a measurable mapping $\mathcal{L}\colon Y \times Y \to \mathbb{R}_+$.

Definition 14.2. Let $\mathcal{L}\colon Y \times Y \to \mathbb{R}_+$ be a loss function and let $\mathcal{D}$ be a distribution on $X \times Y$. For a measurable function $h\colon X \to Y$ we call

$$\mathcal{R}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}(h(x), y)\right]$$

the **risk of** h.

Based on the risk, we can now formalize what we consider a good predictor. The best predictor is one such that its risk is as close as possible to the smallest that any function can achieve. More precisely, we would like a risk that is close to the so-called **Bayes risk**

$$R^* \coloneqq \inf_{h\colon X\to Y} \mathcal{R}(h), \tag{14.1.1}$$

where the infimum is taken over all h such that $h : X \to Y$ is measurable.

Example 14.3 (Loss functions)**.** The choice of a loss function $\mathcal{L}$ usually depends on the application. For a regression problem, i.e., a learning problem where Y is a non-discrete subset of a Euclidean space, a common choice is the square loss $\mathcal{L}_2(\boldsymbol{y}, \boldsymbol{y}') = \|\boldsymbol{y} - \boldsymbol{y}'\|^2$.

For binary classification problems, i.e. when Y is a discrete set of cardinality two, the "$0-1$ loss"

$$\mathcal{L}_{0-1}(y, y') = \begin{cases} 1 & y \neq y' \\ 0 & y = y' \end{cases}$$

is a common choice.

Another frequently used loss for binary classification, especially when we want to predict probabilities (i.e., if $Y = [0, 1]$ but all labels are binary), is the binary cross-entropy loss

$$\mathcal{L}_{\text{ce}}(y, y') = -(y \log(y') + (1 - y) \log(1 - y')).$$

In contrast to the $0-1$ loss, the cross-entropy loss is differentiable, which is desirable in deep learning as we saw in Chapter 10.

In the coffee quality prediction problem, the quality is given as a fraction of the form $k/10$ for $k = 0, \dots, 10$. While this is a discrete set, it makes sense to more heavily penalize predictions that are wrong by a larger amount. For example, predicting $4/10$ instead of $8/10$ should produce a higher loss than predicting $7/10$. Hence, we would not use the $0-1$ loss but, for example, the square loss.

How do we find a function $h\colon X \to Y$ with a risk that is as close as possible to the Bayes risk? We will introduce a procedure to tackle this task in the next section.

14.2 Empirical risk minimization

Finding a minimizer of the risk constitutes a considerable challenge. First, we cannot search through all measurable functions. Therefore, we need to restrict ourselves to a specific set $\mathcal{H} \subseteq \{h : X \to Y\}$ called the **hypothesis set**. In the following, this set will be some set of neural networks. Second, we are faced with the problem that we cannot evaluate $\mathcal{R}(h)$ for non-trivial loss functions, because the distribution $\mathcal{D}$ is unknown. To approximate the risk, we will assume access to an i.i.d. sample of m observations drawn from $\mathcal{D}$. This is precisely the situation described in the coffee quality example of Figure 14.1, where $m = 6$ coffees were sampled.[1] So for a given hypothesis h we can check how well it performs on our sampled data. We call the error on the sample the **empirical risk**.

Definition 14.4. Let $m \in \mathbb{N}$, let $\mathcal{L}\colon Y \times Y \to \mathbb{R}$ be a loss function and let $S = (x_i, y_i)_{i=1}^m \in (X \times Y)^m$ be a sample. For $h\colon X \to Y$, we call

$$\widehat{\mathcal{R}}_S(h) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(h(x_i), y_i)$$

[1] In practice, the assumption of independence of the samples is often unclear and typically not satisfied. For instance, the selection of the six previously tested coffees might be influenced by external factors such as personal preferences or availability at the local store, which introduce bias into the dataset.

the **empirical risk** of h.

If the sample S is drawn i.i.d. according to $\mathcal{D}$, then we immediately see from the linearity of the expected value that $\widehat{\mathcal{R}}_S(h)$ is an unbiased estimator of $\mathcal{R}(h)$, i.e., $\mathbb{E}_{S\sim\mathcal{D}^m}[\widehat{\mathcal{R}}_S(h)] = \mathcal{R}(h)$. Moreover, the weak law of large numbers states that the sample mean of an i.i.d. sequence of integrable random variables converges to the expected value in probability. Hence, there is some hope that, at least for large $m \in \mathbb{N}$, minimizing the empirical risk instead of the actual risk might lead to a good hypothesis. We formalize this approach in the next definition.

Definition 14.5. Let $\mathcal{H} \subseteq \{h\colon X \to Y\}$ be a hypothesis set. Let $m \in \mathbb{N}$, let $\mathcal{L}\colon Y \times Y \to \mathbb{R}$ be a loss function and let $S = (x_i, y_i)_{i=1}^m \in (X \times Y)^m$ be a sample. We call a function h_S such that

$$\widehat{\mathcal{R}}_S(h_S) = \inf_{h\in\mathcal{H}} \widehat{\mathcal{R}}_S(h) \tag{14.2.1}$$

an **empirical risk minimizer**.

From a generalization perspective, deep learning is empirical risk minimization over sets of neural networks. The question we want to address next is how effective this approach is at producing hypotheses that achieve a risk close to the Bayes risk.

Let $\mathcal{H}$ be some hypothesis set, such that an empirical risk minimizer h_S exists for all $S \in (X \times Y)^m$; see Exercise 14.25 for an explanation of why this is a reasonable assumption. Moreover, let $h^* \in \mathcal{H}$ be arbitrary. Then

$$\begin{aligned}\mathcal{R}(h_S) - R^* &= \mathcal{R}(h_S) - \widehat{\mathcal{R}}_S(h_S) + \widehat{\mathcal{R}}_S(h_S) - R^* \\ &\leq |\mathcal{R}(h_S) - \widehat{\mathcal{R}}_S(h_S)| + \widehat{\mathcal{R}}_S(h^*) - R^* \\ &\leq 2\sup_{h\in\mathcal{H}} |\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| + \mathcal{R}(h^*) - R^*,\end{aligned} \tag{14.2.2}$$

where in the first inequality we used that h_S is the empirical risk minimizer. By taking the infimum over all h^*, we conclude that

$$\begin{aligned}\mathcal{R}(h_S) - R^* &\leq 2\sup_{h\in\mathcal{H}} |\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| + \inf_{h\in\mathcal{H}} \mathcal{R}(h) - R^* \\ &=: 2\varepsilon_{\text{gen}} + \varepsilon_{\text{approx}}.\end{aligned} \tag{14.2.3}$$

Similarly, considering only (14.2.2), yields that

$$\begin{aligned}\mathcal{R}(h_S) &\leq \sup_{h\in\mathcal{H}} |\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| + \inf_{h\in\mathcal{H}} \widehat{\mathcal{R}}_S(h) \\ &=: \varepsilon_{\text{gen}} + \varepsilon_{\text{int}}.\end{aligned} \tag{14.2.4}$$

How to choose $\mathcal{H}$ to reduce the **approximation error** $\varepsilon_{\text{approx}}$ or the **interpolation error** ε_{int} was discussed at length in the previous chapters. The final piece is to figure out how to bound the **generalization error** $\sup_{h\in\mathcal{H}} |\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)|$. This will be discussed in the sections below.

14.3 Generalization bounds

We have seen that one aspect of successful learning is to bound the generalization error ε_{gen} in (14.2.3). Let us first formally describe this problem.

Definition 14.6 (Generalization bound)**.** Let $\mathcal{H} \subseteq \{h\colon X \to Y\}$ be a hypothesis set, and let $\mathcal{L}\colon Y \times Y \to \mathbb{R}$ be a loss function. Let $\kappa\colon (0,1) \times \mathbb{N} \to \mathbb{R}_+$ be such that for every $\delta \in (0,1)$ holds $\kappa(\delta, m) \to 0$ for $m \to \infty$. We call κ a **generalization bound for** $\mathcal{H}$ if for every distribution $\mathcal{D}$ on $X \times Y$, every $m \in \mathbb{N}$ and every $\delta \in (0,1)$, it holds with probability at least $1-\delta$ over the random sample $S \sim \mathcal{D}^m$ that

$$\sup_{h \in \mathcal{H}} |\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| \leq \kappa(\delta, m).$$

Remark 14.7. For a generalization bound κ it holds that

$$\mathbb{P}\left[\left|\mathcal{R}(h_S) - \widehat{\mathcal{R}}_S(h_S)\right| \leq \varepsilon\right] \geq 1 - \delta$$

as soon as m is so large that $\kappa(\delta, m) \leq \varepsilon$. If there exists an empirical risk minimizer h_S such that $\widehat{\mathcal{R}}_S(h_S) = 0$, then with high probability the empirical risk minimizer will also have a small risk $\mathcal{R}(h_S)$. Empirical risk minimization is often referred to as a "PAC" algorithm, which stands for *probably (δ) approximately correct (ε).*

Definition 14.6 requires the upper bound κ on the discrepancy between the empirical risk and the risk to be independent from the distribution $\mathcal{D}$. Why should this be possible? After all, we could have an underlying distribution that is not uniform and hence, certain data points could appear very rarely in the sample. As a result, it should be very hard to produce a correct prediction for such points. At first sight, this suggests that non-uniform distributions should be much more challenging than uniform distributions. This intuition is incorrect, as the following argument based on Example 14.1 demonstrates.

Example 14.8 (Generalization in the coffee quality problem)**.** In Example 14.1, the underlying distribution describes both our process of choosing coffees and the relation between the attributes and the quality. Suppose we do not enjoy drinking coffee that costs less than 1€. Consequently, we do not have a single sample of such coffee in the dataset, and therefore we have no chance about learning the quality of cheap coffees.

However, the absence of coffee samples costing less than 1€ in our dataset is due to our *general* avoidance of such coffee. As a result, we run a low risk of incorrectly classifying the quality of a coffee that is cheaper than 1€, since it is unlikely that we will choose such a coffee in the future.

To establish generalization bounds, we use stochastic tools that guarantee that the empirical risk converges to the true risk as the sample size increases. This is typically achieved through concentration inequalities. One of the simplest and most well-known is Hoeffding's inequality, see Theorem A.23. We will now apply Hoeffding's inequality to obtain a first generalization bound. This generalization bound is well-known and can be found in many textbooks on machine learning, e.g., [146, 211]. Although the result does not yet encompass neural networks, it forms the basis for a similar result applicable to neural networks, as we discuss subsequently.

Proposition 14.9 (Finite hypothesis set)**.** *Let $\mathcal{H} \subseteq \{h\colon X \mapsto Y\}$ be a finite hypothesis set. Let $\mathcal{L}\colon Y \times Y \to \mathbb{R}$ be such that $\mathcal{L}(Y \times Y) \subseteq [c_1, c_2]$ with $c_2 - c_1 = C > 0$.*

Then, for every $m \in \mathbb{N}$ and every distribution $\mathcal{D}$ on $X \times Y$ it holds with probability at least $1-\delta$ over the sample $S \sim \mathcal{D}^m$ that

$$\sup_{h\in\mathcal{H}} |\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| \leq C\sqrt{\frac{\log(|\mathcal{H}|) + \log(2/\delta)}{2m}}.$$

Proof. Let $\mathcal{H} = \{h_1, \dots, h_n\}$. Then it holds by a union bound that

$$\mathbb{P}\left[\exists h_i \in \mathcal{H}\colon |\mathcal{R}(h_i) - \widehat{\mathcal{R}}_S(h_i)| > \varepsilon\right] \leq \sum_{i=1}^{n} \mathbb{P}\left[|\mathcal{R}(h_i) - \widehat{\mathcal{R}}_S(h_i)| > \varepsilon\right].$$

Note that $\widehat{\mathcal{R}}_S(h_i)$ is the mean of independent random variables which take their values almost surely in $[0, C]$. Additionally, $\mathcal{R}(h_i)$ is the expectation of $\widehat{\mathcal{R}}_S(h_i)$. The proof can therefore be finished by applying Theorem A.23. This will be addressed in Exercise 14.26. □

Consider now a *non-finite* set of neural networks $\mathcal{H}$, and assume that it can be covered by a *finite* set of (small) balls. Applying Proposition 14.9 to the centers of these balls, then allows to derive a similar bound as in the proposition for $\mathcal{H}$. This intuitive argument will be made rigorous in the following section.

14.4 Generalization bounds from covering numbers

To derive a generalization bound for classes of neural networks, we start by introducing the notion of covering numbers.

Definition 14.10. Let A be a relatively compact subset of a metric space (X, d). For $\varepsilon > 0$, we call

$$\mathcal{G}(A, \varepsilon, (X, d)) := \min\left\{ m \in \mathbb{N} \,\middle|\, \exists\, (x_i)_{i=1}^m \subseteq X \text{ s.t. } \bigcup_{i=1}^m B_\varepsilon(x_i) \supset A \right\},$$

where $B_\varepsilon(x) = \{z \in X \,|\, d(z, x) \leq \varepsilon\}$, the ε-covering number of A in X. In case X or d are clear from context, we also write $\mathcal{G}(A, \varepsilon, d)$ or $\mathcal{G}(A, \varepsilon, X)$ instead of $\mathcal{G}(A, \varepsilon, (X, d))$.

A visualization of Definition 14.10 is given in Figure 14.2. As we will see, it is possible to upper bound the ε-covering numbers of neural networks as a subset of $L^\infty([0,1]^d)$, assuming the weights are confined to a fixed bounded set. The precise estimates are postponed to Section 14.5. Before that, let us show how a finite covering number facilitates a generalization bound. We only consider Euclidean feature spaces X in the following result. A more general version could be easily derived.

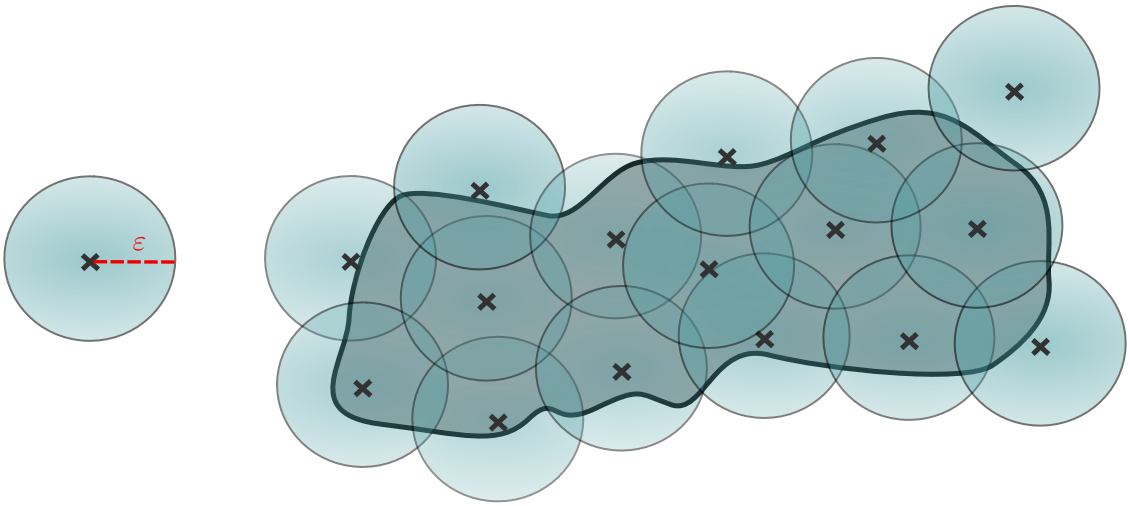

Figure 14.2: Illustration of the concept of covering numbers of Definition 14.10. The shaded set $A \subseteq \mathbb{R}^2$ is covered by sixteen Euclidean balls of radius ε. Therefore, $\mathcal{G}(A, \varepsilon, \mathbb{R}^2) \leq 16$.

Theorem 14.11. *Let C_Y, $C_{\mathcal{L}} > 0$ and $\alpha > 0$. Let $Y \subseteq [-C_Y, C_Y]$, $X \subseteq \mathbb{R}^d$ for some $d \in \mathbb{N}$, and $\mathcal{H} \subseteq \{h \colon X \to Y\}$. Further, let $\mathcal{L} \colon Y \times Y \to \mathbb{R}$ be $C_{\mathcal{L}}$-Lipschitz.*

Then, for every distribution $\mathcal{D}$ on $X \times Y$ and every $m \in \mathbb{N}$ it holds with probability at least $1 - \delta$ over the sample $S \sim \mathcal{D}^m$ that for all $h \in \mathcal{H}$

$$|\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| \leq 4 C_Y C_{\mathcal{L}} \sqrt{\frac{\log(\mathcal{G}(\mathcal{H}, m^{-\alpha}, L^\infty(X))) + \log(2/\delta)}{m}} + \frac{2 C_{\mathcal{L}}}{m^\alpha}.$$

Proof. Let

$$M = \mathcal{G}(\mathcal{H}, m^{-\alpha}, L^\infty(X)) \tag{14.4.1}$$

and let $\mathcal{H}_M = (h_i)_{i=1}^M \subseteq \mathcal{H}$ be such that for every $h \in \mathcal{H}$ there exists $h_i \in \mathcal{H}_M$ with $\|h - h_i\|_{L^\infty(X)} \leq 1/m^\alpha$. The existence of $\mathcal{H}_M$ follows by Definition 14.10.

Fix for the moment such $h \in \mathcal{H}$ and $h_i \in \mathcal{H}_M$. By the reverse and normal triangle inequalities, we have

$$|\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| - |\mathcal{R}(h_i) - \widehat{\mathcal{R}}_S(h_i)| \leq |\mathcal{R}(h) - \mathcal{R}(h_i)| + |\widehat{\mathcal{R}}_S(h) - \widehat{\mathcal{R}}_S(h_i)|.$$

Moreover, from the monotonicity of the expected value and the Lipschitz property of $\mathcal{L}$ it follows that

$$\begin{aligned} |\mathcal{R}(h) - \mathcal{R}(h_i)| &\leq \mathbb{E}|\mathcal{L}(h(x), y) - \mathcal{L}(h_i(x), y)| \\ &\leq C_{\mathcal{L}} \mathbb{E}|h(x) - h_i(x)| \leq \frac{C_{\mathcal{L}}}{m^\alpha}. \end{aligned}$$

A similar estimate yields $|\widehat{\mathcal{R}}_S(h) - \widehat{\mathcal{R}}_S(h_i)| \leq C_{\mathcal{L}}/m^\alpha$.

We thus conclude that for every $\varepsilon > 0$

$$\begin{aligned}&\mathbb{P}_{S\sim\mathcal{D}^m}\left[\exists h \in \mathcal{H}\colon |\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| \geq \varepsilon\right] \\ &\quad \leq \mathbb{P}_{S\sim\mathcal{D}^m}\left[\exists h_i \in \mathcal{H}_M\colon |\mathcal{R}(h_i) - \widehat{\mathcal{R}}_S(h_i)| \geq \varepsilon - \frac{2C_{\mathcal{L}}}{m^\alpha}\right]. \end{aligned} \tag{14.4.2}$$

From Proposition 14.9, we know that for $\varepsilon > 0$ and $\delta \in (0,1)$

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left[\exists h_i \in H_M\colon |\mathcal{R}(h_i) - \widehat{\mathcal{R}}_S(h_i)| \geq \varepsilon - \frac{2C_{\mathcal{L}}}{m^\alpha}\right] \leq \delta \tag{14.4.3}$$

as long as

$$\varepsilon - \frac{2C_{\mathcal{L}}}{m^\alpha} > C\sqrt{\frac{\log(M) + \log(2/\delta)}{2m}},$$

where C is such that $\mathcal{L}(Y \times Y) \subseteq [c_1, c_2]$ with $c_2 - c_1 \leq C$. By the Lipschitz property of $\mathcal{L}$ we can choose $C = 2\sqrt{2}C_{\mathcal{L}}C_Y$.

Therefore, the definition of M in (14.4.1) together with (14.4.2) and (14.4.3) give that with probability at least $1-\delta$ it holds for all $h \in \mathcal{H}$

$$|\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| \leq 2\sqrt{2}C_{\mathcal{L}}C_Y\sqrt{\frac{\log(\mathcal{G}(\mathcal{H}, m^{-\alpha}, L^\infty)) + \log(2/\delta)}{2m}} + \frac{2C_{\mathcal{L}}}{m^\alpha}.$$

This concludes the proof. □

14.5 Covering numbers of deep neural networks

We have seen in Theorem 14.11, estimating L^∞-covering numbers is crucial for understanding the generalization error. How can we determine these covering numbers? The set of neural networks of a fixed architecture can be a quite complex set (see Chapter 13), so it is not immediately clear how to cover it with balls, let alone know the number of required balls. The following lemma suggest a simpler approach.

Lemma 14.12. *Let X_1, X_2 be two metric spaces and let $f\colon X_1 \to X_2$ be Lipschitz continuous with Lipschitz constant C_{Lip}. For every relatively compact $A \subseteq X_1$ it holds that for all $\varepsilon > 0$*

$$\mathcal{G}(f(A), C_{\mathrm{Lip}}\varepsilon, X_2) \leq \mathcal{G}(A, \varepsilon, X_1).$$

The proof of Lemma 14.12 is left as an exercise. If we can represent the set of neural networks as the image under the Lipschitz map of another set with known covering numbers, then Lemma 14.12 gives a direct way to bound the covering number of the neural network class.

Conveniently, we have already observed in Proposition 13.1, that the set of neural networks is the image of $\mathcal{PN}(\mathcal{A}, B)$ as in Definition 12.1 under the Lipschitz continuous realization map R_σ. It thus suffices to establish the ε-covering number of $\mathcal{PN}(\mathcal{A}, B)$ or equivalently of $[-B, B]^{n_{\mathcal{A}}}$. Then, using the Lipschitz property of R_σ that holds by Proposition 13.1, we can apply Lemma 14.12 to find the covering numbers of $\mathcal{N}(\sigma; \mathcal{A}, B)$. This idea is depicted in Figure 14.3.

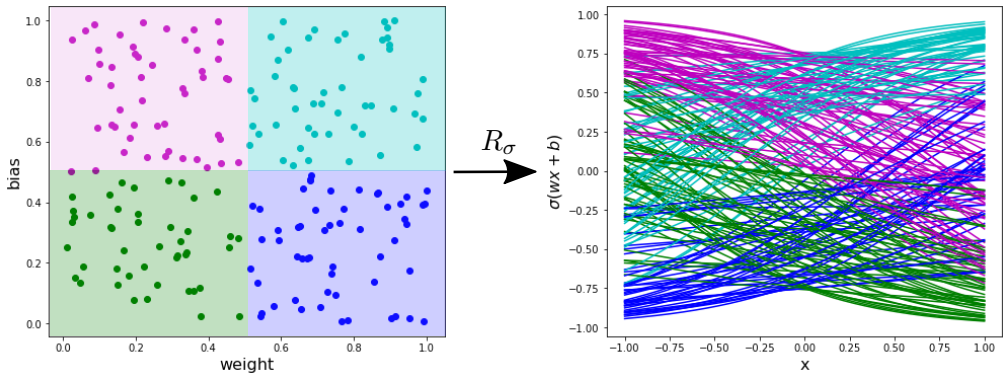

Figure 14.3: Illustration of the main idea to deduce covering numbers of neural network spaces. Points $\theta \in \mathbb{R}^2$ in parameter space in the left figure correspond to functions $R_\sigma(\theta)$ in the right figure (with matching colors). By Lemma 14.12, a covering of the parameter space on the left translates to a covering of the function space on the right.

Proposition 14.13. *Let B, $\varepsilon > 0$ and $q \in \mathbb{N}$. Then*

$$\mathcal{G}([-B,B]^q, \varepsilon, (\mathbb{R}^q, \|\cdot\|_\infty)) \leq \lceil B/\varepsilon \rceil^q.$$

Proof. We start with the one-dimensional case $q = 1$. We choose $k = \lfloor B/\varepsilon \rfloor$

$$x_0 = -B + \varepsilon \text{ and } x_j = x_{j-1} + 2\varepsilon \text{ for } j = 1, \dots, k-1.$$

It is clear that all points between $-B$ and x_{k-1} have distance at most ε to one of the x_j. Also, $x_{k-1} = -B + \varepsilon + 2(k-1)\varepsilon \geq B - \varepsilon$. We conclude that $\mathcal{G}([-B,B], \varepsilon, \mathbb{R}) \leq \lceil B/\varepsilon \rceil$. Set $X_k := \{x_0, \dots, x_{k-1}\}$.

For arbitrary q, we observe that for every $x \in [-B,B]^q$ there is an element in $X_k^q = \bigotimes_{j=1}^q X_k$ with $\|\cdot\|_\infty$ distance less than ε. Clearly, $|X_k^q| = \lceil B/\varepsilon \rceil^q$, which completes the proof. □

Having established a covering number for $[-B,B]^{n_\mathcal{A}}$ and hence $\mathcal{PN}(\mathcal{A}, B)$, we can now estimate the covering numbers of deep neural networks by combining Lemma 14.12 and Propositions 13.1 and 14.13 .

Theorem 14.14. *Let $\mathcal{A} = (d_0, d_1, \dots, d_{L+1}) \in \mathbb{N}^{L+2}$, let $\sigma\colon \mathbb{R} \to \mathbb{R}$ be C_σ-Lipschitz continuous with $C_\sigma \geq 1$, let $|\sigma(x)| \leq C_\sigma|x|$ for all $x \in \mathbb{R}$, and let $B \geq 1$. Then*

$$\begin{aligned}\mathcal{G}(\mathcal{N}(\sigma;\mathcal{A},B), \varepsilon, L^\infty([0,1]^{d_0})) &\leq \mathcal{G}([-B,B]^{n_\mathcal{A}}, \varepsilon/(2C_\sigma B d_{\max})^L, (\mathbb{R}^{n_\mathcal{A}}, \|\cdot\|_\infty)) \\ &\leq \lceil n_\mathcal{A}/\varepsilon \rceil^{n_\mathcal{A}} \lceil 2C_\sigma B d_{\max} \rceil^{n_\mathcal{A} L}.\end{aligned}$$

We end this section, by applying the previous theorem to the generalization bound of Theorem 14.11 with $\alpha = 1/2$. To simplify the analysis, we restrict the discussion to neural networks with range $[-1, 1]$. To this end, denote

$$\mathcal{N}^*(\sigma; \mathcal{A}, B) := \big\{\Phi \in \mathcal{N}(\sigma; \mathcal{A}, B)\big| \\ \Phi(\boldsymbol{x}) \in [-1, 1] \text{ for all } \boldsymbol{x} \in [0, 1]^{d_0}\big\}. \tag{14.5.1}$$

Since $\mathcal{N}^*(\sigma; \mathcal{A}, B) \subseteq \mathcal{N}(\sigma; \mathcal{A}, B)$ we can bound the covering numbers of $\mathcal{N}^*(\sigma; \mathcal{A}, B)$ by those of $\mathcal{N}(\sigma; \mathcal{A}, B)$. This yields the following result.

Theorem 14.15. *Let $C_{\mathcal{L}} > 0$ and let $\mathcal{L}\colon [-1, 1]\times[-1, 1] \to \mathbb{R}$ be $C_{\mathcal{L}}$-Lipschitz continuous. Further, let $\mathcal{A} = (d_0, d_1, \dots, d_{L+1}) \in \mathbb{N}^{L+2}$, let $\sigma\colon \mathbb{R} \to \mathbb{R}$ be C_σ-Lipschitz continuous with $C_\sigma \geq 1$, and $|\sigma(x)| \leq C_\sigma|x|$ for all $x \in \mathbb{R}$, and let $B \geq 1$.*

Then, for every $m \in \mathbb{N}$, and every distribution $\mathcal{D}$ on $X \times [-1, 1]$ it holds with probability at least $1 - \delta$ over $S \sim \mathcal{D}^m$ that for all $\Phi \in \mathcal{N}^*(\sigma; \mathcal{A}, B)$

$$|\mathcal{R}(\Phi) - \widehat{\mathcal{R}}_S(\Phi)| \leq 4C_{\mathcal{L}}\sqrt{\frac{n_{\mathcal{A}} \log(\lceil n_{\mathcal{A}}\sqrt{m}\rceil) + Ln_{\mathcal{A}} \log(\lceil 2C_\sigma B d_{\max}\rceil) + \log(2/\delta)}{m}} \\ + \frac{2C_{\mathcal{L}}}{\sqrt{m}}.$$

14.6 The approximation-complexity trade-off

We recall the decomposition of the error in (14.2.3)

$$\mathcal{R}(h_S) - R^* \leq 2\varepsilon_{\text{gen}} + \varepsilon_{\text{approx}},$$

where R_* is the Bayes risk defined in (14.1.1). We make the following observations about the approximation error $\varepsilon_{\text{approx}}$ and generalization error ε_{gen} in the context of neural network based learning:

- *Scaling of generalization error:* By Theorem 14.15, for a hypothesis class $\mathcal{H}$ of neural networks with $n_{\mathcal{A}}$ weights and L layers, and for sample of size $m \in \mathbb{N}$, the generalization error ε_{gen} essentially scales like

$$\varepsilon_{\text{gen}} = \mathcal{O}(\sqrt{(n_{\mathcal{A}} \log(n_{\mathcal{A}} m) + Ln_{\mathcal{A}} \log(n_{\mathcal{A}}))/m}) \qquad \text{as } m \to \infty.$$

- *Scaling of approximation error:* Assume there exists h^* such that $\mathcal{R}(h^*) = R^*$, and let the loss function $\mathcal{L}$ be Lipschitz continuous in the first coordinate. Then

$$\varepsilon_{\text{approx}} = \inf_{h\in\mathcal{H}} \mathcal{R}(h) - \mathcal{R}(h^*) = \inf_{h\in\mathcal{H}} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}(h(x), y) - \mathcal{L}(h^*(x), y)] \\ \lesssim \inf_{h\in\mathcal{H}} \|h - h^*\|_{L^\infty}.$$

We have seen in Chapters 5 and 7 that if we choose $\mathcal{H}$ as a set of neural networks with size $n_{\mathcal{A}}$ and L layers, then, for appropriate activation functions, $\inf_{h\in\mathcal{H}} \|h - h^*\|_{L^\infty}$ behaves like ${n_{\mathcal{A}}}^{-r}$ if, e.g., h^* is a d-dimensional s-Hölder regular function and $r = s/d$ (Theorem 5.22), or $h^* \in C^{k,s}([0,1]^d)$ and $r < (k+s)/d$ (Theorem 7.7).

By these considerations, we conclude that for an empirical risk minimizer Φ_S from a set of neural networks with $n_{\mathcal{A}}$ weights and L layers, it holds that

$$\mathcal{R}(\Phi_S) - R^* \leq \mathcal{O}(\sqrt{(n_{\mathcal{A}} \log(m) + L n_{\mathcal{A}} \log(n_{\mathcal{A}}))/m}) + \mathcal{O}({n_{\mathcal{A}}}^{-r}), \tag{14.6.1}$$

for $m \to \infty$ and for some r depending on the regularity of h^*. Note that, enlarging the neural network set, i.e., increasing $n_{\mathcal{A}}$ has two effects: The term associated to approximation decreases, and the term associated to generalization increases. This trade-off is known as **approximation-complexity trade-off**. The situation is depicted in Figure 14.4. The figure and (14.6.1) suggest that, the perfect model, achieves the optimal trade-off between approximation and generalization error. Using this notion, we can also separate all models into three classes:

- *Underfitting*: If the approximation error decays faster than the estimation error increases.
- *Optimal*: If the sum of approximation error and generalization error is at a minimum.
- *Overfitting:* If the approximation error decays slower than the estimation error increases.

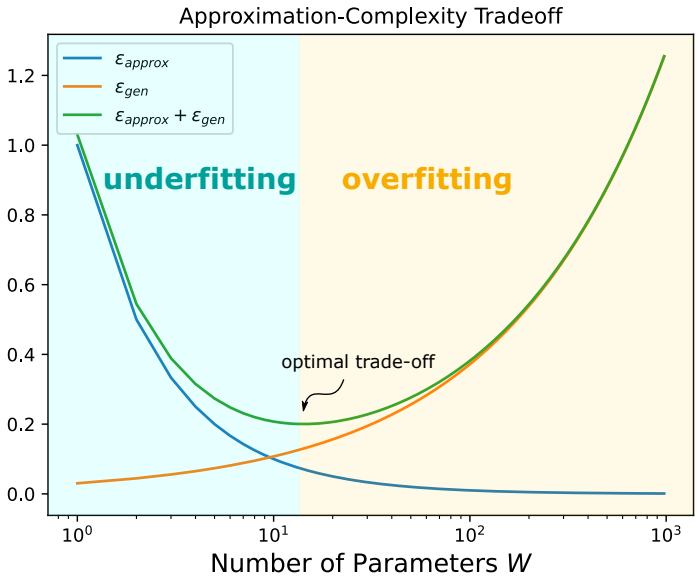

Figure 14.4: Illustration of the approximation-complexity-trade-off of Equation (14.6.1). Here we chose $r = 1$ and $m = 10^4$, also all implicit constants are assumed to be equal to 1.

In Chapter 15, we will see that deep learning often operates in the regime where the number of parameters $n_{\mathcal{A}}$ exceeds the optimal trade-off point. For certain architectures used in practice, $n_{\mathcal{A}}$ can be so large that the theory of the approximation-complexity trade-off suggests that learning should be impossible. However, we emphasize, that the present analysis only provides upper bounds. It does not prove that learning is impossible or even impractical in the overparameterized regime. Moreover, in Chapter 11 we have already seen indications that learning in the overparametrized regime need not necessarily lead to large generalization errors.

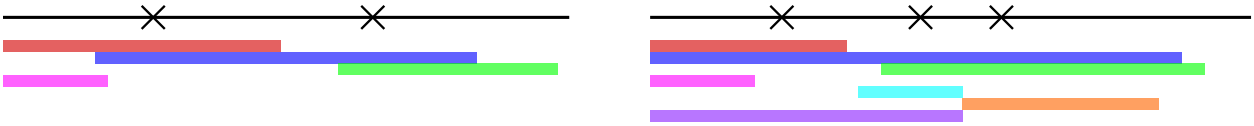

Figure 14.5: Different ways to classify two or three points. The colored-blocks correspond to intervals that produce different classifications of the points.

14.7 PAC learning from VC dimension

In addition to covering numbers, there are several other tools to analyze the generalization capacity of hypothesis sets. In the context of classification problems, one of the most important is the so-called Vapnik–Chervonenkis (VC) dimension.

14.7.1 Definition and examples

Let $\mathcal{H}$ be a hypothesis set of functions mapping from $\mathbb{R}^d$ to $\{0,1\}$. A set $S = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_n\} \subseteq \mathbb{R}^d$ is said to be **shattered** by $\mathcal{H}$ if for every $(y_1, \dots, y_n) \in \{0,1\}^n$ there exists $h \in \mathcal{H}$ such that $h(\boldsymbol{x}_j) = y_j$ for all $j \in \mathbb{N}$.

The VC dimension quantifies the complexity of a function class via the number of points that can in principle be shattered.

Definition 14.16. The **VC dimension** of $\mathcal{H}$ is the cardinality of the largest set $S \subseteq \mathbb{R}^d$ that is shattered by $\mathcal{H}$. We denote the VC dimension by $\mathrm{VCdim}(\mathcal{H})$.

Example 14.17 (Intervals)**.** Let $\mathcal{H} = \{\mathbb{1}_{[a,b]} \,|\, a, b \in \mathbb{R}\}$. It is clear that $\mathrm{VCdim}(\mathcal{H}) \geq 2$ since for $x_1 < x_2$ the functions

$$\mathbb{1}_{[x_1-2,x_1-1]}, \quad \mathbb{1}_{[x_1-1,x_1]}, \quad \mathbb{1}_{[x_1,x_2]}, \quad \mathbb{1}_{[x_2,x_2+1]},$$

are all different, when restricted to $S = (x_1, x_2)$.

On the other hand, if $x_1 < x_2 < x_3$ then, since $h^{-1}(\{1\})$ is an interval for all $h \in \mathcal{H}$ we have that $h(x_1) = 1 = h(x_3)$ implies $h(x_2) = 1$. Hence, no set of three elements can be shattered. Therefore, $\mathrm{VCdim}(\mathcal{H}) = 2$. The situation is depicted in Figure 14.5.

Example 14.18 (Two-dimensional half-spaces)**.** Let $\mathcal{H} = \{\mathbb{1}_{[0,\infty)}(\langle \boldsymbol{w}, \cdot \rangle + b) \,|\, \boldsymbol{w} \in \mathbb{R}^2, b \in \mathbb{R}\}$ be a hypothesis set of rotated and shifted two-dimensional half-spaces. In Figure 14.6 we see that $\mathcal{H}$ shatters a set of three points. More general, with

$$\mathcal{H}_d := \{\boldsymbol{x} \mapsto \mathbb{1}_{[0,\infty)}(\boldsymbol{w}^\top \boldsymbol{x} + b) \,|\, \boldsymbol{w} \in \mathbb{R}^d,\ b \in \mathbb{R}\}$$

the VC dimension of $\mathcal{H}_d$ equals $d + 1$.

In the example above, the VC dimension coincides with the number of parameters. However, this is not true in general as the following example shows.

Example 14.19 (Infinite VC dimension)**.** Let for $x \in \mathbb{R}$

$$\mathcal{H} := \{x \mapsto \mathbb{1}_{[0,\infty)}(\sin(wx)) \,|\, w \in \mathbb{R}\}.$$

Then the VC dimension of $\mathcal{H}$ is infinite (Exercise 14.29).

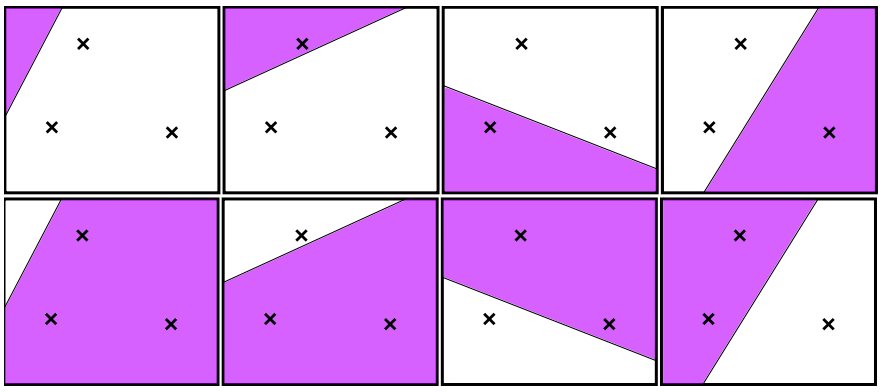

Figure 14.6: Different ways to classify three points by a half-space.

14.7.2 Generalization based on VC dimension

In the following, we consider a classification problem. Denote by $\mathcal{D}$ the data-generating distribution on $\mathbb{R}^d \times \{0,1\}$. Moreover, we let $\mathcal{H}$ be a set of functions from $\mathbb{R}^d \to \{0,1\}$.

In the binary classification set-up, the natural choice of a loss function is the $0-1$ loss $\mathcal{L}_{0-1}(y,y') = \mathbb{1}_{y\neq y'}$. Thus, given a sample S, the empirical risk of a function $h \in \mathcal{H}$ is

$$\widehat{\mathcal{R}}_S(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}_{h(\boldsymbol{x}_i)\neq y_i}.$$

Moreover, the risk can be written as

$$\mathcal{R}(h) = \mathbb{P}_{(\boldsymbol{x},y)\sim\mathcal{D}}[h(\boldsymbol{x}) \neq y],$$

i.e., the probability under $(\boldsymbol{x},y) \sim \mathcal{D}$ of h misclassifying the label y of $\boldsymbol{x}$.

We can now give a generalization bound in terms of the VC dimension of $\mathcal{H}$, see, e.g., [146, Corollary 3.19]:

Theorem 14.20. *Let $d,k \in \mathbb{N}$ and $\mathcal{H} \subseteq \{h\colon \mathbb{R}^d \to \{0,1\}\}$ have VC dimension k. Let $\mathcal{D}$ be a distribution on $\mathbb{R}^d \times \{0,1\}$. Then, for every $\delta > 0$ and $m \in \mathbb{N}$, it holds with probability at least $1-\delta$ over a sample $S \sim \mathcal{D}^m$ that for every $h \in \mathcal{H}$*

$$|\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| \leq \sqrt{\frac{2k\log(em/k)}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}}. \tag{14.7.1}$$

In words, Theorem 14.20 tells us that if a hypothesis class has finite VC dimension, then a hypothesis with a small empirical risk will have a small risk if the number of samples is large. This shows that empirical risk minimization is a viable strategy in this scenario. Will this approach also work if the VC dimension is not bounded? No, in fact, in that case, no learning algorithm will

succeed in reliably producing a hypothesis for which the risk is close to the best possible. We omit the technical proof of the following theorem from [146, Theorem 3.23].

Theorem 14.21. *Let $k \in \mathbb{N}$ and let $\mathcal{H} \subseteq \{h\colon X \to \{0,1\}\}$ be a hypothesis set with VC dimension k. Then, for every $m \in \mathbb{N}$ and every learning algorithm $\mathrm{A}\colon (X \times \{0,1\})^m \to \mathcal{H}$ there exists a distribution $\mathcal{D}$ on $X \times \{0,1\}$ such that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left[\mathcal{R}(\mathrm{A}(S)) - \inf_{h\in\mathcal{H}} \mathcal{R}(h) > \sqrt{\frac{k}{320m}}\right] \geq \frac{1}{64}.$$

Theorem 14.21 immediately implies the following statement for the generalization bound.

Corollary 14.22. *Let $k \in \mathbb{N}$ and let $\mathcal{H} \subseteq \{h\colon X \to \{0,1\}\}$ be a hypothesis set with VC dimension k. Then, for every $m \in \mathbb{N}$ there exists a distribution $\mathcal{D}$ on $X \times \{0,1\}$ such that*

$$\mathbb{P}_{S\sim\mathcal{D}^m}\left[\sup_{h\in\mathcal{H}} |\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| > \sqrt{\frac{k}{1280m}}\right] \geq \frac{1}{64}.$$

Proof. For a sample S, let $h_S \in \mathcal{H}$ be an empirical risk minimizer, i.e., $\widehat{\mathcal{R}}_S(h_S) = \min_{h\in\mathcal{H}} \widehat{\mathcal{R}}_S(h)$. Let $\mathcal{D}$ be the distribution of Theorem 14.21. Moreover, for $\delta > 0$, let $h_\delta \in \mathcal{H}$ be such that

$$\mathcal{R}(h_\delta) - \inf_{h\in\mathcal{H}} \mathcal{R}(h) < \delta.$$

Then, applying Theorem 14.21 with $\mathrm{A}(S) = h_S$ it holds that

$$\begin{aligned}
2\sup_{h\in\mathcal{H}} |\mathcal{R}(h) - \widehat{\mathcal{R}}_S(h)| &\geq |\mathcal{R}(h_S) - \widehat{\mathcal{R}}_S(h_S)| + |\mathcal{R}(h_\delta) - \widehat{\mathcal{R}}_S(h_\delta)| \\
&\geq \mathcal{R}(h_S) - \widehat{\mathcal{R}}_S(h_S) + \widehat{\mathcal{R}}_S(h_\delta) - \mathcal{R}(h_\delta) \\
&\geq \mathcal{R}(h_S) - \mathcal{R}(h_\delta) \\
&> \mathcal{R}(h_S) - \inf_{h\in\mathcal{H}} \mathcal{R}(h) - \delta,
\end{aligned}$$

where we used the definition of h_S in the third inequality. The proof is completed by applying Theorem 14.21 and using that δ was arbitrary. □

We have seen now, that we have a generalization bound scaling like $\mathcal{O}(1/\sqrt{m})$ for $m \to \infty$ if and only if the VC dimension of a hypothesis class is finite. In more quantitative terms, we require the VC dimension of a neural network to be smaller than m.

What does this imply for neural network functions? For ReLU neural networks there holds the following [3, Theorem 8.8].

Theorem 14.23. *Let $\mathcal{A} \in \mathbb{N}^{L+2}$, $L \in \mathbb{N}$ and set*

$$\mathcal{H} := \{\mathbb{1}_{[0,\infty)} \circ \Phi \,|\, \Phi \in \mathcal{N}(\sigma_{\mathrm{ReLU}}; \mathcal{A}, \infty)\}.$$

Then, there exists a constant $C > 0$ independent of L and $\mathcal{A}$ such that

$$\mathrm{VCdim}(\mathcal{H}) \leq C \cdot (n_{\mathcal{A}} L \log(n_{\mathcal{A}}) + n_{\mathcal{A}} L^2).$$

The bound (14.7.1) is meaningful if $m \gg k$. For ReLU neural networks as in Theorem 14.23, this means $m \gg n_{\mathcal{A}} L \log(n_{\mathcal{A}}) + n_{\mathcal{A}} L^2$. Fixing $L = 1$ this amounts to $m \gg n_{\mathcal{A}} \log(n_{\mathcal{A}})$ for a shallow neural network with $n_{\mathcal{A}}$ parameters. This condition is contrary to what we assumed in Chapter 11, where it was crucial that $n_{\mathcal{A}} \gg m$. If the VC dimension of the neural network sets scale like $\mathcal{O}(n_{\mathcal{A}} \log(n_{\mathcal{A}}))$, then Theorem 14.21 and Corollary 14.22 indicate that, at least for certain distributions, generalization should not be possible in this regime. We will discuss the resolution of this potential paradox in Chapter 15.

14.8 Lower bounds on achievable approximation rates

We conclude this chapter on the complexities and generalization bounds of neural networks by using the established VC dimension bound of Theorem 14.23 to deduce limitations to the approximation capacity of neural networks. The result described below was first given in [244].

Theorem 14.24. *Let k, $d \in \mathbb{N}$. Assume that for every $\varepsilon > 0$ exists $L_\varepsilon \in \mathbb{N}$ and $\mathcal{A}_\varepsilon$ with L_ε layers and input dimension d such that*

$$\sup_{\|f\|_{C^k([0,1]^d)} \leq 1} \inf_{\Phi \in \mathcal{N}(\sigma_{\mathrm{ReLU}}; \mathcal{A}, \infty)} \|f - \Phi\|_{C^0([0,1]^d)} < \frac{\varepsilon}{2}.$$

Then there exists $C > 0$ solely depending on k and d, such that for all $\varepsilon \in (0, 1)$

$$n_{\mathcal{A}_\varepsilon} L_\varepsilon \log(n_{\mathcal{A}_\varepsilon}) + n_{\mathcal{A}_\varepsilon} L_\varepsilon^2 \geq C \varepsilon^{-\frac{d}{k}}.$$

Proof. For $\boldsymbol{x} \in \mathbb{R}^d$ consider the "bump function"

$$\tilde{f}(\boldsymbol{x}) := \begin{cases} \exp\left(1 - \frac{1}{1 - \|\boldsymbol{x}\|_2^2}\right) & \text{if } \|\boldsymbol{x}\|_2 < 1 \\ 0 & \text{otherwise,} \end{cases}$$

and its scaled version

$$\tilde{f}_\varepsilon(\boldsymbol{x}) := \varepsilon f\left(2\varepsilon^{-1/k} \boldsymbol{x}\right),$$

for $\varepsilon \in (0,1)$. Then

$$\operatorname{supp}(\tilde{f}_\varepsilon) \subseteq \Big[-\frac{\varepsilon^{1/k}}{2}, \frac{\varepsilon^{1/k}}{2}\Big]^d$$

and

$$\|\tilde{f}_\varepsilon\|_{C^k} \leq 2^k \|\tilde{f}\|_{C^k} =: \tau_k > 0.$$

Consider the equispaced point set $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_{N(\varepsilon)}\} = \varepsilon^{1/k}\mathbb{Z}^d \cap [0,1]^d$. The cardinality of this set is $N(\varepsilon) \simeq \varepsilon^{-d/k}$. Given $\boldsymbol{y} \in \{0,1\}^{N(\varepsilon)}$, let for $\boldsymbol{x} \in \mathbb{R}^d$

$$f_{\boldsymbol{y}}(\boldsymbol{x}) \coloneqq \tau_k^{-1} \sum_{j=1}^{N(\varepsilon)} y_j \tilde{f}_\varepsilon(\boldsymbol{x} - \boldsymbol{x}_j). \tag{14.8.1}$$

Then $f_{\boldsymbol{y}}(\boldsymbol{x}_j) = \tau_k^{-1}\varepsilon y_j$ for all $j = 1, \dots, N(\varepsilon)$ and $\|f_{\boldsymbol{y}}\|_{C^k} \leq 1$.

For every $\boldsymbol{y} \in \{0,1\}^{N(\varepsilon)}$ let $\Phi_{\boldsymbol{y}} \in \mathcal{N}(\sigma_{\mathrm{ReLU}}; \mathcal{A}_{\tau_k^{-1}\varepsilon}, \infty)$ be such that

$$\sup_{\boldsymbol{x}\in[0,1]^d} |f_{\boldsymbol{y}}(\boldsymbol{x}) - \Phi_{\boldsymbol{y}}(\boldsymbol{x})| < \frac{\varepsilon}{2\tau_k}.$$

Then

$$\mathbb{1}_{[0,\infty)}\Big(\Phi_y(\boldsymbol{x}_j) - \frac{\varepsilon}{2\tau_k}\Big) = y_j \qquad \text{for all } j = 1, \dots, N(\varepsilon).$$

Hence, the VC dimension of $\mathcal{N}(\sigma_{\mathrm{ReLU}}; \mathcal{A}_{\tau_k^{-1}\varepsilon}, \infty)$ is larger or equal to $N(\varepsilon)$. Theorem 14.23 thus implies

$$N(\varepsilon) \simeq \varepsilon^{-\frac{d}{k}} \leq C \cdot \Big(n_{\mathcal{A}_{\tau_k^{-1}\varepsilon}} L_{\tau_k^{-1}\varepsilon} \log(n_{\mathcal{A}_{\tau_k^{-1}\varepsilon}}) + n_{\mathcal{A}_{\tau_k^{-1}\varepsilon}} L^2_{\tau_k^{-1}\varepsilon}\Big)$$

or equivalently

$$\tau_k^{\frac{d}{k}} \varepsilon^{-\frac{d}{k}} \leq C \cdot \Big(n_{\mathcal{A}_{\tau_k^{-1}\varepsilon}} L_\varepsilon \log(n_{\mathcal{A}_{\tau_k^{-1}\varepsilon}}) + n_{\mathcal{A}_{\tau_k^{-1}\varepsilon}} L^2_\varepsilon\Big).$$

This completes the proof. □

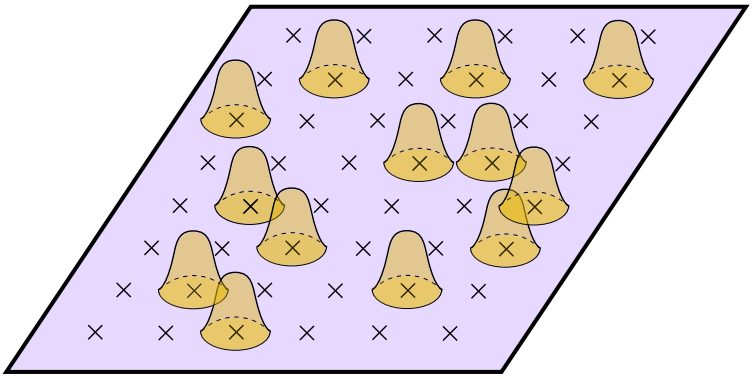

Figure 14.7: Illustration of $f_{\boldsymbol{y}}$ from Equation (14.8.1) on $[0,1]^2$.

To interpret Theorem 14.24, we consider two situations:

- In case the depth is allowed to increase at most logarithmically in ε, then reaching uniform error ε for all $f \in C^k([0,1]^d)$ with $\|f\|_{C^k([0,1]^d)} \leq 1$ requires

$$n_{\mathcal{A}_\varepsilon} \log(n_{\mathcal{A}_\varepsilon}) \log(\varepsilon) + n_{\mathcal{A}_\varepsilon} \log(\varepsilon)^2 \geq C\varepsilon^{-\frac{d}{k}}.$$

  In terms of the neural network size, this (necessary) condition becomes $n_{\mathcal{A}_\varepsilon} \geq C\varepsilon^{-d/k}/\log(\varepsilon)^2$. As we have shown in Chapter 7, in particular Theorem 7.7, up to log terms this condition is also sufficient. Hence, while the constructive proof of Theorem 7.7 might have seemed rather specific, under the assumption of the depth increasing at most logarithmically (which the construction in Chapter 7 satisfies), it was essentially optimal! The neural networks in this proof are shown to have size $O(\varepsilon^{-d/k})$ up to log terms.

- If we allow the depth L_ε to increase faster than logarithmically in ε, then the lower bound on the required neural network size improves. Fixing for example $\mathcal{A}_\varepsilon$ with L_ε layers such that $n_{\mathcal{A}_\varepsilon} \leq WL_\varepsilon$ for some fixed ε independent $W \in \mathbb{N}$, the (necessary) condition on the depth becomes

$$W \log(WL_\varepsilon) L_\varepsilon^2 + WL_\varepsilon^3 \geq C\varepsilon^{-\frac{d}{k}}$$

  and hence $L_\varepsilon \gtrsim \varepsilon^{-d/(3k)}$.

  We add that, for arbitrary depth the upper bound on the VC dimension of Theorem 14.23 can be improved to $n_{\mathcal{A}}^2$, [3, Theorem 8.6], and using this, would improve the just established lower bound to $L_\varepsilon \gtrsim \varepsilon^{-d/(2k)}$.

  For fixed width, this corresponds to neural networks of size $O(\varepsilon^{-d/(2k)})$, which would mean twice the convergence rate proven in Theorem 7.7. Indeed, it turns out that neural networks can achieve this rate in terms of the neural network size [245].

To sum up, in order to get error ε uniformly for all $\|f\|_{C^k([0,1]^d)} \leq 1$, the size of a ReLU neural network is required to increase at least like $O(\varepsilon^{-d/(2k)})$ as $\varepsilon \to 0$, i.e. the best possible attainable convergence rate is $2k/d$. It has been proven, that this rate is also achievable, and thus the bound is sharp. Achieving this rate requires neural network architectures that grow faster in depth than in width.

Bibliography and further reading

Classical statistical learning theory is based on the foundational work of Vapnik and Chervonenkis [232]. This led to the formulation of the probably approximately correct (PAC) learning model in [231], which is primarily utilized in this chapter. A streamlined mathematical introduction to statistical learning theory can be found in [42].

Since statistical learning theory is well-established, there exists a substantial amount of excellent expository work describing this theory. Some highly recommended books on the topic are [146, 211, 3]. The specific approach of characterizing learning via covering numbers has been discussed extensively in [3, Chapter 14]. Specific results for ReLU activation used in this chapter were derived in [203, 18]. The results of Section 14.8 describe some of the findings in [244, 245], and we also refer to [50] for general lower bounds (also applicable to neural networks) when approximating classes of Sobolev functions.

Exercises

Exercise 14.25. Let $\mathcal{H}$ be a set of neural networks with fixed architecture, where the weights are taken from a compact set. Moreover, assume that the activation function is continuous. Show that for every sample S there always exists an empirical risk minimizer h_S.

Exercise 14.26. Complete the proof of Proposition 14.9.

Exercise 14.27. Prove Lemma 14.12.

Exercise 14.28. Show that, the VC dimension of $\mathcal{H}$ of Example 14.18 is indeed 3, by demonstrating that no set of four points can be shattered by $\mathcal{H}$.

Exercise 14.29. Show that the VC dimension of

$$\mathcal{H} := \{x \mapsto \mathbb{1}_{[0,\infty)}(\sin(wx)) \,|\, w \in \mathbb{R}\}$$

is infinite.