Chapter 9

Interpolation

The learning problem associated to minimizing the empirical risk of (1.2.3) is based on minimizing an error that results from evaluating a neural network on a *finite* set of (training) points. In contrast, all previous approximation results focused on achieving uniformly small errors across the entire domain. Finding neural networks that achieve a small training error appears to be much simpler, since, instead of $\|f - \Phi_n\|_\infty \to 0$ for a sequence of neural networks Φ_n, it suffices to have $\Phi_n(\boldsymbol{x}_i) \to f(\boldsymbol{x}_i)$ for all $\boldsymbol{x}_i$ in the training set.

In this chapter, we study the extreme case of the aforementioned approximation problem. We analyze under which conditions it is possible to find a neural network that coincides with the target function f at all training points. This is referred to as *interpolation*. To make this notion more precise, we state the following definition.

Definition 9.1 (Interpolation)**.** Let d, $m \in \mathbb{N}$, and let $\Omega \subseteq \mathbb{R}^d$. We say that a set of functions $\mathcal{H} \subseteq \{h \colon \Omega \to \mathbb{R}\}$ **interpolates m points in Ω**, if for every $S = (\boldsymbol{x}_i, y_i)_{i=1}^m \subseteq \Omega \times \mathbb{R}$, such that $\boldsymbol{x}_i \neq \boldsymbol{x}_j$ for $i \neq j$, there exists a function $h \in \mathcal{H}$ such that $h(\boldsymbol{x}_i) = y_i$ for all $i = 1, \dots, m$.

Knowing the interpolation properties of an architecture represents extremely valuable information for two reasons:

- Consider an architecture that interpolates m points and let the number of training samples be bounded by m. Then (1.2.3) always has a solution.
- Consider again an architecture that interpolates m points and assume that the number of training samples is *less* than m. Then for every point $\tilde{\boldsymbol{x}}$ not in the training set and every $y \in \mathbb{R}$ there exists a minimizer h of (1.2.3) that satisfies $h(\tilde{\boldsymbol{x}}) = y$. As a consequence, without further restrictions (many of which we will discuss below), such an architecture cannot generalize to unseen data.

The existence of solutions to the interpolation problem does not follow trivially from the approximation results provided in the previous chapters (even though we will later see that there is a close connection). We also remark that the question of how many points neural networks with a given architecture can interpolate is closely related to the so-called VC dimension, which we will study in Chapter 14.

We start our analysis of the interpolation properties of neural networks by presenting a result similar to the universal approximation theorem but for interpolation in the following section. In the subsequent section, we then look at interpolation with desirable properties.

9.1 Universal interpolation

Under what conditions on the activation function and architecture can a set of neural networks interpolate $m \in \mathbb{N}$ points? According to Chapter 3, particularly Theorem 3.8, we know that shallow neural networks can approximate every continuous function with arbitrary accuracy, provided the neural network width is large enough. As the neural network's width and/or depth increases, the architectures become increasingly powerful, leading us to expect that at some point, they should be able to interpolate m points. However, this intuition may not be correct:

Example 9.2. Let $\mathcal{H} := \{f \in C^0([0,1]) \,|\, f(0) \in \mathbb{Q}\}$. Then $\mathcal{H}$ is dense in $C^0([0,1])$, but $\mathcal{H}$ does not even interpolate one point in $[0,1]$.

Moreover, Theorem 3.8 is an asymptotic result that only states that a given function can be approximated for sufficiently large neural network architectures, but it does not state how large the architecture needs to be.

Surprisingly, Theorem 3.8 can nonetheless be used to give a guarantee that a fixed-size architecture yields sets of neural networks that allow the interpolation of m points. This result is due to [174]; for a more detailed discussion of previous results see the bibliography section. Due to its similarity to the universal approximation theorem and the fact that it uses the same assumptions, we call the following theorem the "Universal Interpolation Theorem". For its statement recall the definition of the set of allowed activation functions $\mathcal{M}$ in (3.1.1) and the class $\mathcal{N}_d^1(\sigma, 1, n)$ of shallow neural networks of width n introduced in Definition 3.6.

Theorem 9.3 (Universal Interpolation Theorem)**.** *Let $d, n \in \mathbb{N}$ and let $\sigma \in \mathcal{M}$ not be a polynomial. Then $\mathcal{N}_d^1(\sigma, 1, n)$ interpolates $n+1$ points in $\mathbb{R}^d$.*

Proof. Fix $(\boldsymbol{x}_i)_{i=1}^{n+1} \subseteq \mathbb{R}^d$ arbitrary. We will show that for any $(y_i)_{i=1}^{n+1} \subseteq \mathbb{R}$ there exist weights and biases $(\boldsymbol{w}_j)_{j=1}^n \subseteq \mathbb{R}^d$, $(b_j)_{j=1}^n$, $(v_j)_{j=1}^n \subseteq \mathbb{R}$, $c \in \mathbb{R}$ such that

$$\Phi(\boldsymbol{x}_i) := \sum_{j=1}^{n} v_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x}_i + b_j) + c = y_i \quad \text{for all} \quad i = 1, \dots, n+1. \tag{9.1.1}$$

Since $\Phi \in \mathcal{N}_d^1(\sigma, 1, n)$ this then concludes the proof.

Denote

$$\boldsymbol{A} := \begin{pmatrix} 1 & \sigma(\boldsymbol{w}_1^\top \boldsymbol{x}_1 + b_1) & \cdots & \sigma(\boldsymbol{w}_n^\top \boldsymbol{x}_1 + b_n) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \sigma(\boldsymbol{w}_1^\top \boldsymbol{x}_{n+1} + b_1) & \cdots & \sigma(\boldsymbol{w}_n^\top \boldsymbol{x}_{n+1} + b_n) \end{pmatrix} \in \mathbb{R}^{(n+1)\times(n+1)}. \tag{9.1.2}$$

Then $\boldsymbol{A}$ being regular implies that for each $(y_i)_{i=1}^{n+1}$ exist c and $(v_j)_{j=1}^n$ such that (9.1.1) holds. Hence, it suffices to find $(\boldsymbol{w}_j)_{j=1}^n$ and $(b_j)_{j=1}^n$ such that $\boldsymbol{A}$ is regular.

To do so, we proceed by induction over $k = 0, \dots, n$, to show that there exist $(\boldsymbol{w}_j)_{j=1}^k$ and $(b_j)_{j=1}^k$ such that the first $k+1$ columns of $\boldsymbol{A}$ are linearly independent. The case $k = 0$ is trivial. Next let $0 < k < n$ and assume that the first k columns of $\boldsymbol{A}$ are linearly independent. We wish to find $\boldsymbol{w}_k$, b_k such that the first $k+1$ columns are linearly independent. Suppose such $\boldsymbol{w}_k$, b_k do not exist and denote by $Y_k \subseteq \mathbb{R}^{n+1}$ the space spanned by the first k columns of $\boldsymbol{A}$. Then for all $\boldsymbol{w} \in \mathbb{R}^n$, $b \in \mathbb{R}$ the vector $(\sigma(\boldsymbol{w}^\top \boldsymbol{x}_i + b))_{i=1}^{n+1} \in \mathbb{R}^{n+1}$ must belong to Y_k. Fix $\boldsymbol{y} = (y_i)_{i=1}^{n+1} \in \mathbb{R}^{n+1} \backslash Y_k$. Then

$$\begin{aligned}\inf_{\tilde{\Phi} \in \mathcal{N}_d^1(\sigma,1)} \|(\tilde{\Phi}(\boldsymbol{x}_i))_{i=1}^{n+1} - \boldsymbol{y}\|_2^2 &= \inf_{N, \boldsymbol{w}_j, b_j, v_j, c} \sum_{i=1}^{n+1} \Big(\sum_{j=1}^{N} v_j \sigma(\boldsymbol{w}_j^\top \boldsymbol{x}_i + b_j) + c - y_i \Big)^2 \\ &\geq \inf_{\tilde{\boldsymbol{y}} \in Y_k} \|\tilde{\boldsymbol{y}} - \boldsymbol{y}\|_2^2 > 0.\end{aligned}$$

Since we can find a continuous function $f : \mathbb{R}^d \to \mathbb{R}$ such that $f(\boldsymbol{x}_i) = y_i$ for all $i = 1, \dots, n+1$, this contradicts Theorem 3.8. □

9.2 Optimal interpolation and reconstruction

Consider a bounded domain $\Omega \subseteq \mathbb{R}^d$, a function $f : \Omega \to \mathbb{R}$, distinct points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m \subseteq \Omega$, and corresponding function values $y_i := f(\boldsymbol{x}_i)$. Our objective is to approximate f based solely on the data pairs $(\boldsymbol{x}_i, y_i)$, $i = 1, \dots, m$. In this section, we will show that, under certain assumptions on f, ReLU neural networks can express an "optimal" reconstruction which also turns out to be an interpolant of the data.

9.2.1 Motivation

In the previous section, we observed that neural networks with $m - 1 \in \mathbb{N}$ hidden neurons can interpolate m points for every reasonable activation function. However, not all interpolants are equally suitable for a given application. For instance, consider Figure 9.1 for a comparison between polynomial and piecewise affine interpolation on the unit interval.

The two interpolants exhibit rather different behaviors. In general, there is no way of determining which constitutes a better approximation to f. In particular, given our limited information about f, we cannot accurately reconstruct any additional features that may exist between interpolation points $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m$. In accordance with Occam's razor, it thus seems reasonable to assume that f does not exhibit extreme oscillations or behave erratically between interpolation points. As such, the piecewise interpolant appears preferable in this scenario. One way to formalize the assumption that f does not "exhibit extreme oscillations" is to *assume* that the Lipschitz constant

$$\mathrm{Lip}(f) := \sup_{\boldsymbol{x} \neq \boldsymbol{y}} \frac{|f(\boldsymbol{x}) - f(\boldsymbol{y})|}{\|\boldsymbol{x} - \boldsymbol{y}\|}$$

of f is bounded by a fixed value $M \in \mathbb{R}$. Here $\|\cdot\|$ denotes an arbitrary fixed norm on $\mathbb{R}^d$.

How should we choose M? For every function $f : \Omega \to \mathbb{R}$ satisfying

$$f(\boldsymbol{x}_i) = y_i \quad \text{for all} \quad i = 1, \dots, m, \tag{9.2.1}$$

we have

$$\mathrm{Lip}(f) = \sup_{\boldsymbol{x} \neq \boldsymbol{y} \in \Omega} \frac{|f(\boldsymbol{x}) - f(\boldsymbol{y})|}{\|\boldsymbol{x} - \boldsymbol{y}\|} \geq \sup_{i \neq j} \frac{|y_i - y_j|}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|} =: \tilde{M}. \tag{9.2.2}$$

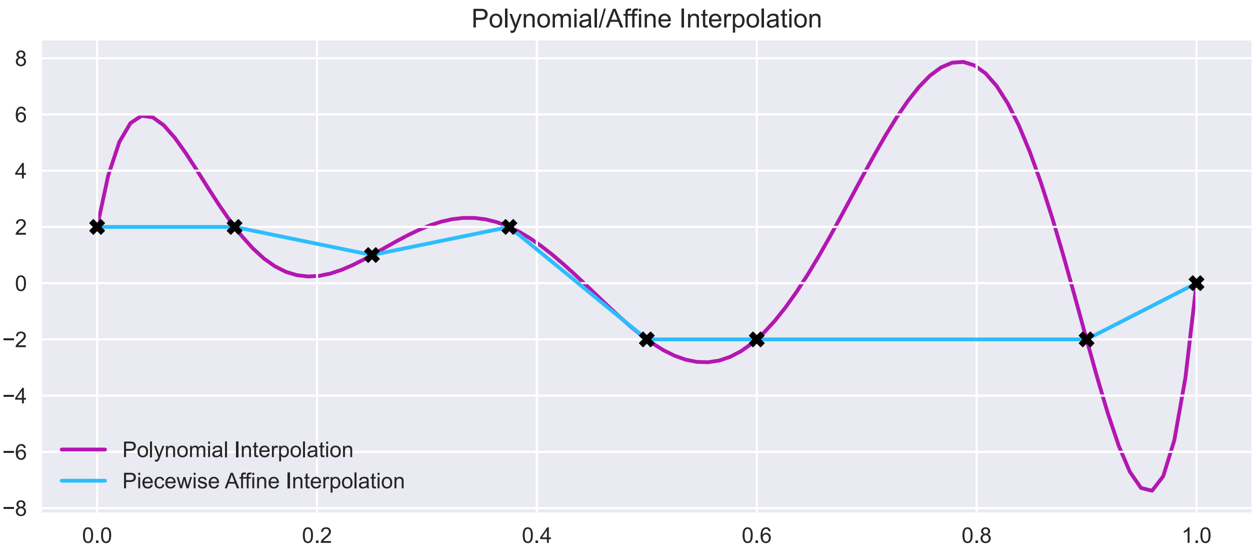

Figure 9.1: Interpolation of eight points by a polynomial of degree seven and by a piecewise affine spline. The polynomial interpolation has a significantly larger derivative or Lipschitz constant than the piecewise affine interpolator.

Because of this, we fix M as a real number greater than or equal to $\tilde{M}$ for the remainder of our analysis.

9.2.2 Optimal reconstruction for Lipschitz continuous functions

The above considerations raise the following question: *Given only the information that the function has Lipschitz constant at most M, what is the best reconstruction of f based on the data?* We consider here the "best reconstruction" to be a function that minimizes the L^∞-error in the worst case. Specifically, with

$$\mathrm{Lip}_M(\Omega) := \{f : \Omega \to \mathbb{R} \,|\, \mathrm{Lip}(f) \leq M\}, \tag{9.2.3}$$

denoting the set of all functions with Lipschitz constant at most M, we want to solve the following problem:

Problem 9.4. We wish to find an element

$$\Phi \in \mathrm{argmin}_{h:\Omega\to\mathbb{R}} \sup_{\substack{f\in\mathrm{Lip}_M(\Omega) \\ f \text{ satisfies (9.2.1)}}} \sup_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x}) - h(\boldsymbol{x})|. \tag{9.2.4}$$

The next theorem shows that a function Φ as in (9.2.4) indeed exists. This Φ not only allows for an explicit formula, it also belongs to $\mathrm{Lip}_M(\Omega)$ and additionally interpolates the data. Hence, it is not just an optimal reconstruction, it is also an optimal interpolant. This theorem goes back to [13], which, in turn, is based on [218].

Theorem 9.5. *Let m, $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, $f : \Omega \to \mathbb{R}$, and let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m \in \Omega$, $y_1, \dots, y_m \in \mathbb{R}$ satisfy* (9.2.1) *and* (9.2.2) *with $\tilde{M} > 0$. Further, let $M \geq \tilde{M}$.*

Then, Problem 9.4 has at least one solution given by

$$\Phi(\boldsymbol{x}) := \frac{1}{2}(f_{\text{upper}}(\boldsymbol{x}) + f_{\text{lower}}(\boldsymbol{x})) \qquad \text{for } \boldsymbol{x} \in \Omega, \tag{9.2.5}$$

where

$$\begin{aligned} f_{\text{upper}}(\boldsymbol{x}) &:= \min_{k=1,\dots,m} (y_k + M\|\boldsymbol{x} - \boldsymbol{x}_k\|) \\ f_{\text{lower}}(\boldsymbol{x}) &:= \max_{k=1,\dots,m} (y_k - M\|\boldsymbol{x} - \boldsymbol{x}_k\|). \end{aligned}$$

Moreover, $\Phi \in \mathrm{Lip}_M(\Omega)$ and Φ interpolates the data (i.e. satisfies (9.2.1)*).*

Proof. First we claim that for all h_1, $h_2 \in \mathrm{Lip}_M(\Omega)$ holds $\max\{h_1, h_2\} \in \mathrm{Lip}_M(\Omega)$ as well as $\min\{h_1, h_2\} \in \mathrm{Lip}_M(\Omega)$. Since $\min\{h_1, h_2\} = -\max\{-h_1, -h_2\}$, it suffices to show the claim for the maximum. We need to check that

$$\frac{|\max\{h_1(\boldsymbol{x}), h_2(\boldsymbol{x})\} - \max\{h_1(\boldsymbol{y}), h_2(\boldsymbol{y})\}|}{\|\boldsymbol{x} - \boldsymbol{y}\|} \leq M \tag{9.2.6}$$

for all $\boldsymbol{x} \neq \boldsymbol{y} \in \Omega$. Fix $\boldsymbol{x} \neq \boldsymbol{y}$. Without loss of generality we assume that

$$\max\{h_1(\boldsymbol{x}), h_2(\boldsymbol{x})\} \geq \max\{h_1(\boldsymbol{y}), h_2(\boldsymbol{y})\} \quad \text{and} \quad \max\{h_1(\boldsymbol{x}), h_2(\boldsymbol{x})\} = h_1(\boldsymbol{x}).$$

If $\max\{h_1(\boldsymbol{y}), h_2(\boldsymbol{y})\} = h_1(\boldsymbol{y})$ then the numerator in (9.2.6) equals $h_1(\boldsymbol{x}) - h_1(\boldsymbol{y})$ which is bounded by $M\|\boldsymbol{x} - \boldsymbol{y}\|$. If $\max\{h_1(\boldsymbol{y}), h_2(\boldsymbol{y})\} = h_2(\boldsymbol{y})$, then the numerator equals $h_1(\boldsymbol{x}) - h_2(\boldsymbol{y})$ which is bounded by $h_1(\boldsymbol{x}) - h_1(\boldsymbol{y}) \leq M\|\boldsymbol{x} - \boldsymbol{y}\|$. In either case (9.2.6) holds.

Clearly, $\boldsymbol{x} \mapsto y_k - M\|\boldsymbol{x} - \boldsymbol{x}_k\| \in \mathrm{Lip}_M(\Omega)$ for each $k = 1, \dots, m$ and thus f_{upper}, $f_{\text{lower}} \in \mathrm{Lip}_M(\Omega)$ as well as $\Phi \in \mathrm{Lip}_M(\Omega)$.

Next we claim that for all $f \in \mathrm{Lip}_M(\Omega)$ satisfying (9.2.1) holds

$$f_{\text{lower}}(\boldsymbol{x}) \leq f(\boldsymbol{x}) \leq f_{\text{upper}}(\boldsymbol{x}) \quad \text{for all} \quad \boldsymbol{x} \in \Omega. \tag{9.2.7}$$

This is true since for every $k \in \{1, \dots, m\}$ and $\boldsymbol{x} \in \Omega$

$$|y_k - f(\boldsymbol{x})| = |f(\boldsymbol{x}_k) - f(\boldsymbol{x})| \leq M\|\boldsymbol{x} - \boldsymbol{x}_k\|$$

so that for all $\boldsymbol{x} \in \Omega$

$$f(\boldsymbol{x}) \leq \min_{k=1,\dots,m} (y_k + M\|\boldsymbol{x} - \boldsymbol{x}_k\|), \qquad f(\boldsymbol{x}) \geq \max_{k=1,\dots,m} (y_k - M\|\boldsymbol{x} - \boldsymbol{x}_k\|).$$

Since f_{upper}, $f_{\text{lower}} \in \mathrm{Lip}_M(\Omega)$ satisfy (9.2.1), we conclude that for every $h : \Omega \to \mathbb{R}$ holds

$$\begin{aligned} \sup_{\substack{f \in \mathrm{Lip}_M(\Omega) \\ f \text{ satisfies (9.2.1)}}} \sup_{\boldsymbol{x} \in \Omega} |f(\boldsymbol{x}) - h(\boldsymbol{x})| &\geq \sup_{\boldsymbol{x} \in \Omega} \max\{|f_{\text{lower}}(\boldsymbol{x}) - h(\boldsymbol{x})|, |f_{\text{upper}}(\boldsymbol{x}) - h(\boldsymbol{x})|\} \\ &\geq \sup_{\boldsymbol{x} \in \Omega} \frac{|f_{\text{lower}}(\boldsymbol{x}) - f_{\text{upper}}(\boldsymbol{x})|}{2}. \end{aligned} \tag{9.2.8}$$

Moreover, using (9.2.7),

$$\begin{aligned}\sup_{\substack{f\in \mathrm{Lip}_M(\Omega)\\ f \text{ satisfies (9.2.1)}}} \sup_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x})-\Phi(\boldsymbol{x})| &\le \sup_{\boldsymbol{x}\in\Omega} \max\{|f_{\mathrm{lower}}(\boldsymbol{x})-\Phi(\boldsymbol{x})|, |f_{\mathrm{upper}}(\boldsymbol{x})-\Phi(\boldsymbol{x})|\}\\ &= \sup_{\boldsymbol{x}\in\Omega} \frac{|f_{\mathrm{lower}}(\boldsymbol{x})-f_{\mathrm{upper}}(\boldsymbol{x})|}{2}. \end{aligned} \tag{9.2.9}$$

Finally, (9.2.8) and (9.2.9) imply that Φ is a solution of Problem 9.4. □

Figure 9.2 depicts f_{upper}, f_{lower}, and Φ for the interpolation problem shown in Figure 9.1, while Figure 9.3 provides a two-dimensional example.

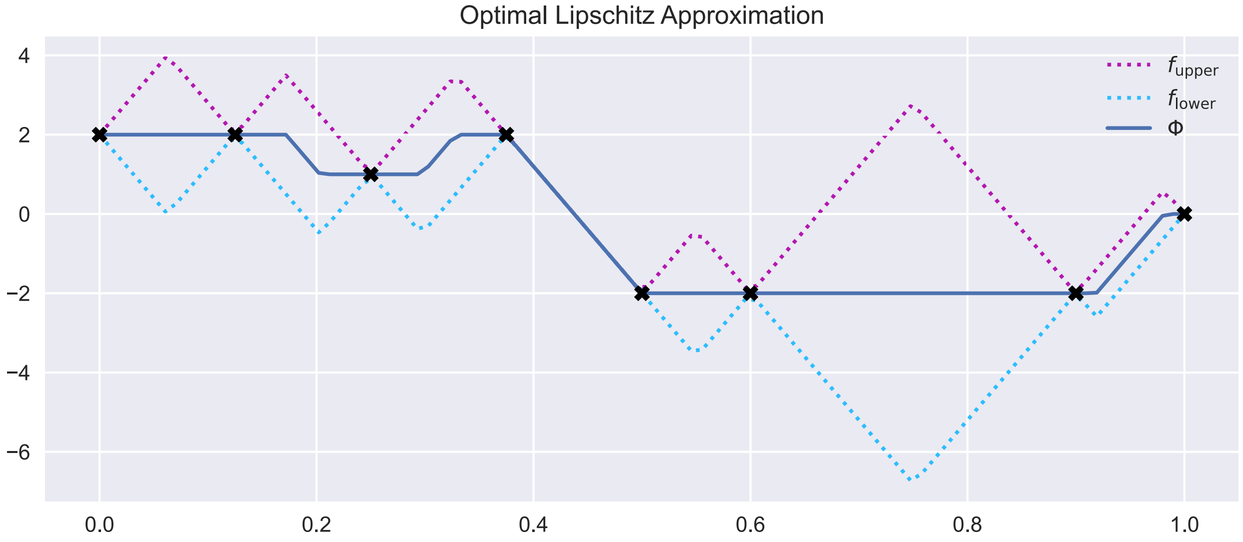

Figure 9.2: Interpolation of the points from Figure 9.1 with the optimal Lipschitz interpolant.

9.2.3 Optimal ReLU reconstructions

So far everything was valid with an arbitrary norm on $\mathbb{R}^d$. For the next theorem, we will restrict ourselves to the 1-norm $\|\boldsymbol{x}\|_1 = \sum_{j=1}^d |x_j|$. Using the explicit formula of Theorem 9.5, we will now show the remarkable result that ReLU neural networks can exactly express an optimal reconstruction (in the sense of Problem 9.4) with a neural network whose size scales linearly in the product of the dimension d and the number of data points m. Additionally, the proof is constructive, thus allowing in principle for an explicit construction on the neural network without the need for training.

Theorem 9.6 (Optimal Lipschitz Reconstruction)**.** *Let m, $d \in \mathbb{N}$, $\Omega \subseteq \mathbb{R}^d$, $f : \Omega \to \mathbb{R}$, and let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m \in \Omega$, $y_1, \dots, y_m \in \mathbb{R}$ satisfy* (9.2.1) *and* (9.2.2) *with $\tilde{M} > 0$. Further, let $M \ge \tilde{M}$ and let $\|\cdot\| = \|\cdot\|_1$ in* (9.2.2) *and* (9.2.3)*.*

Then, there exists a ReLU neural network $\Phi \in \mathrm{Lip}_M(\Omega)$ *that interpolates the data (i.e. satisfies* (9.2.1)*) and satisfies*

$$\Phi \in \mathrm{argmin}_{h:\Omega\to\mathbb{R}} \sup_{\substack{f\in\mathrm{Lip}_M(\Omega)\\ f \text{ satisfies } (9.2.1)}} \sup_{\boldsymbol{x}\in\Omega} |f(\boldsymbol{x}) - h(\boldsymbol{x})|.$$

Moreover, $\mathrm{depth}(\Phi) = O(\log(m))$, $\mathrm{width}(\Phi) = O(dm)$ *and all weights of* Φ *are bounded in absolute value by* $\max\{M, \|\boldsymbol{y}\|_\infty\}$.

Proof. To prove the result, we simply need to show that the function in (9.2.5) can be expressed as a ReLU neural network with the size bounds described in the theorem. First we notice, that there is a simple ReLU neural network that implements the 1-norm. It holds for all $\boldsymbol{x} \in \mathbb{R}^d$ that

$$\|\boldsymbol{x}\|_1 = \sum_{i=1}^{d} (\sigma(x_i) + \sigma(-x_i)).$$

Thus, there exists a ReLU neural network $\Phi^{\|\cdot\|_1}$ such that for all $\boldsymbol{x} \in \mathbb{R}^d$

$$\mathrm{width}(\Phi^{\|\cdot\|_1}) = 2d, \qquad \mathrm{depth}(\Phi^{\|\cdot\|_1}) = 1, \qquad \Phi^{\|\cdot\|_1}(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$$

As a result, there exist ReLU neural networks $\Phi_k : \mathbb{R}^d \to \mathbb{R}$, $k = 1, \dots, m$, such that

$$\mathrm{width}(\Phi_k) = 2d, \qquad \mathrm{depth}(\Phi_k) = 1, \qquad \Phi_k(\boldsymbol{x}) = y_k + M\|\boldsymbol{x} - \boldsymbol{x}_k\|_1$$

for all $\boldsymbol{x} \in \mathbb{R}^d$. Using the parallelization of neural networks introduced in Section 5.1.3, there exists a ReLU neural network $\Phi_{\mathrm{all}} := (\Phi_1, \dots, \Phi_m)\colon \mathbb{R}^d \to \mathbb{R}^m$ such that

$$\begin{aligned} &\mathrm{width}(\Phi_{\mathrm{all}}) = 4md, \qquad \mathrm{depth}(\Phi_{\mathrm{all}}) = 2, \text{ and} \\ &\Phi_{\mathrm{all}}(\boldsymbol{x}) = (y_k + M\|\boldsymbol{x} - \boldsymbol{x}_k\|_1)_{k=1}^m \qquad \text{for all } \boldsymbol{x} \in \mathbb{R}^d. \end{aligned}$$

Using Lemma 5.11, we can now find a ReLU neural network Φ_{upper} such that $\Phi_{\mathrm{upper}} = f_{\mathrm{upper}}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega$, $\mathrm{width}(\Phi_{\mathrm{upper}}) \le \max\{16m, 4md\}$, and $\mathrm{depth}(\Phi_{\mathrm{upper}}) \le 1 + \log(m)$.

Essentially the same construction yields a ReLU neural network Φ_{lower} with the respective properties. Lemma 5.4 then completes the proof. □

Bibliography and further reading

The universal interpolation theorem stated in this chapter is due to [174, Theorem 5.1]. There exist several earlier interpolation results, which were shown under stronger assumptions: In [199], the interpolation property is already linked with a rank condition on the matrix (9.1.2). However, no general conditions on the activation functions were formulated. In [104], the interpolation theorem is established under the assumption that the activation function σ is continuous and nondecreasing, $\lim_{x\to-\infty} \sigma(x) = 0$, and $\lim_{x\to\infty} \sigma(x) = 1$. This result was improved in [96], which dropped the nondecreasing assumption on σ.

The main idea of the optimal Lipschitz interpolation theorem in Section 9.2 is due to [13]. A neural network construction of Lipschitz interpolants, which however is not the optimal interpolant in the sense of Problem 9.4, is given in [106, Theorem 2.27].

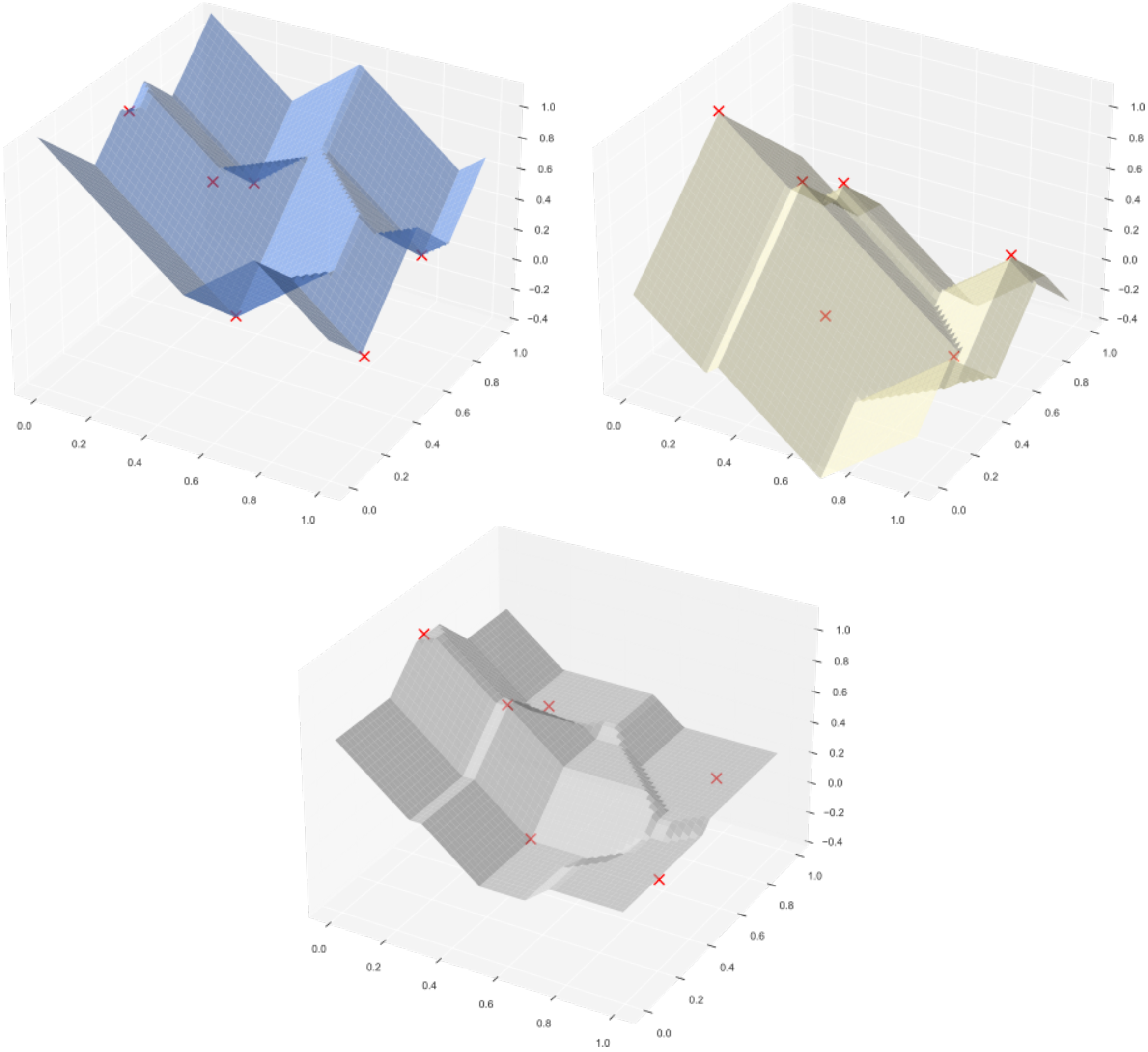

Figure 9.3: Two-dimensional example of the interpolation method of (9.2.5). From top left to bottom we see f_{upper}, f_{lower}, and Φ. The interpolation points $(\boldsymbol{x}_i, y_i)_{i=1}^6$ are marked with red crosses.