Chapter 4

Splines

In Chapter 3, we saw that sufficiently large neural networks can approximate every continuous function to arbitrary accuracy. However, these results provided little insight into the meaning of "sufficiently large" and the choice of a suitable architecture. Ideally, given a function f, and a desired accuracy $\varepsilon > 0$, we would like to have a (possibly sharp) bound on the required size, depth, and width guaranteeing the existence of a neural network approximating f up to error ε.

The field of approximation theory establishes such trade-offs between properties of the function f (e.g., its smoothness), the approximation accuracy, and the number of parameters needed to achieve this accuracy. For example, given k, $d \in \mathbb{N}$, how many parameters are required to approximate a function $f : [0,1]^d \to \mathbb{R}$ with $\|f\|_{C^k([0,1]^d)} \leq 1$ up to uniform error ε? Splines are known to achieve this approximation accuracy with a superposition of $O(\varepsilon^{-d/k})$ simple (piecewise polynomial) basis functions. In this chapter, following [144], we show that certain sigmoidal neural networks can match this performance in terms of the neural network size. In fact, from an approximation theoretical viewpoint we show that the considered neural networks are at least as expressive as superpositions of splines.

4.1 B-splines and smooth functions

We introduce a simple type of spline and its approximation properties below.

Definition 4.1. For $n \in \mathbb{N}$, the **univariate cardinal B-spline** order $n \in \mathbb{N}$ is given by

$$\mathcal{S}_n(x) \coloneqq \frac{1}{(n-1)!} \sum_{\ell=0}^{n} (-1)^\ell \binom{n}{\ell} \sigma_{\mathrm{ReLU}}(x-\ell)^{n-1} \qquad \text{for } x \in \mathbb{R}, \tag{4.1.1}$$

where $0^0 := 0$ and σ_{ReLU} denotes the ReLU activation function.

By shifting and dilating the cardinal B-spline, we obtain a system of univariate splines. Taking tensor products of these univariate splines yields a set of higher-dimensional functions known as the multivariate B-splines.

Definition 4.2. For $t \in \mathbb{R}$ and n, $\ell \in \mathbb{N}$ we define $\mathcal{S}_{\ell,t,n} \coloneqq \mathcal{S}_n(2^\ell(\cdot - t))$. Additionally, for $d \in \mathbb{N}$, $\boldsymbol{t} \in \mathbb{R}^d$, and n, $\ell \in \mathbb{N}$, we define the **the multivariate B-spline** $\mathcal{S}^d_{\ell,\boldsymbol{t},n}$ as

$$\mathcal{S}^d_{\ell,\boldsymbol{t},n}(\boldsymbol{x}) \coloneqq \prod_{i=1}^{d} \mathcal{S}_{\ell,t_i,n}(x_i) \qquad \text{for } \boldsymbol{x} = (x_1, \dots x_d) \in \mathbb{R}^d,$$

and

$$\mathcal{B}^n \coloneqq \left\{ \mathcal{S}^d_{\ell,\boldsymbol{t},n} \,\middle|\, \ell \in \mathbb{N}, \boldsymbol{t} \in \mathbb{R}^d \right\}$$

is the **dictionary of B-splines of order** n.

Having introduced the system $\mathcal{B}^n$, we would like to understand how well we can represent each smooth function by superpositions of elements of $\mathcal{B}^n$. The following theorem is adapted from the more general result [166, Theorem 7]; also see [139, Theorem D.3] for a presentation closer to the present formulation.

Theorem 4.3. *Let d, n, $k \in \mathbb{N}$ such that $0 < k \leq n$. Then there exists C such that for every $f \in C^k([0,1]^d)$ and every $N \in \mathbb{N}$, there exist $c_i \in \mathbb{R}$ with $|c_i| \leq C\|f\|_{L^\infty([0,1]^d)}$ and $B_i \in \mathcal{B}^n$ for $i = 1, \dots, N$, such that*

$$\left\| f - \sum_{i=1}^{N} c_i B_i \right\|_{L^\infty([0,1]^d)} \leq C N^{-\frac{k}{d}} \|f\|_{C^k[0,1]^d}.$$

Remark 4.4. There are a couple of critical concepts in Theorem 4.9 that will reappear throughout this book. The number of parameters N determines the approximation accuracy $N^{-k/d}$. This implies that achieving accuracy $\varepsilon > 0$ requires $O(\varepsilon^{-d/k})$ parameters (according to this upper bound), which grows exponentially in d. This exponential dependence on d is referred to as the "curse of dimension" and will be discussed again in the subsequent chapters. The smoothness parameter k has the opposite effect of d, and improves the convergence rate. Thus, smoother functions can be approximated with fewer B-splines than rougher functions. This more efficient approximation requires the use of B-splines of order n with $n \geq k$. We will see in the following, that the order of the B-spline is closely linked to the concept of depth in neural networks.

4.2 Reapproximation of B-splines with sigmoidal activations

We now show that the approximation rates of B-splines can be transfered to certain neural networks. The following argument is based on [142].

Definition 4.5. A function $\sigma : \mathbb{R} \to \mathbb{R}$ is called **sigmoidal of order** $q \in \mathbb{N}$, if $\sigma \in C^{q-1}(\mathbb{R})$ and there exists $C > 0$ such that

$$\begin{aligned} &\frac{\sigma(x)}{x^q} \to 0 && \text{as } x \to -\infty, \\ &\frac{\sigma(x)}{x^q} \to 1 && \text{as } x \to \infty, \\ &|\sigma(x)| \leq C \cdot (1 + |x|)^q && \text{for all } x \in \mathbb{R}. \end{aligned}$$

Example 4.6. The rectified power unit $x \mapsto \sigma_{\mathrm{ReLU}}(x)^q$ is sigmoidal of order q.

Our goal in the following is to show that neural networks can approximate a linear combination of N B-splines with a number of parameters that is proportional to N. As an immediate consequence of Theorem 4.9, we then obtain a convergence rate for neural networks. Let us start by approximating a single univariate B-spline with a neural network of fixed size.

Proposition 4.7. *Let n, $d \in \mathbb{N}$, $n \geq 2$, $K > 0$, and let $\sigma : \mathbb{R} \to \mathbb{R}$ be sigmoidal of order $q \geq 2$. There exists a constant $C > 0$ such that for every $\varepsilon > 0$ there is a neural network $\Phi^{\mathcal{S}_n}$ with activation function σ, $\lceil \log_q(n-1) \rceil$ layers, and size C, such that*

$$\left\| \mathcal{S}_n - \Phi^{\mathcal{S}_n} \right\|_{L^\infty([-K,K]^d)} \leq \varepsilon.$$

Proof. By definition (4.1.1), $\mathcal{S}_n$ is a linear combination of $n+1$ shifts of $\sigma_{\mathrm{ReLU}}^{n-1}$. We start by approximating $\sigma_{\mathrm{ReLU}}^{n-1}$. It is not hard to see (Exercise 4.10) that, for every $K' > 0$ and every $t \in \mathbb{N}$

$$\left| a^{-q^t} \underbrace{\sigma \circ \sigma \circ \cdots \circ \sigma(ax)}_{t- \text{ times}} - \sigma_{\mathrm{ReLU}}(x)^{q^t} \right| \to 0 \qquad \text{as } a \to \infty \tag{4.2.1}$$

uniformly for all $x \in [-K', K']$.

Set $t := \lceil \log_q(n-1) \rceil$. Then $t \geq 1$ since $n \geq 2$, and $q^t \geq n-1$. Thus, for every $K' > 0$ and $\varepsilon > 0$ there exists a neural network $\Phi_\varepsilon^{q^t}$ with $\lceil \log_q(n-1) \rceil$ layers satisfying

$$\left| \Phi_\varepsilon^{q^t}(x) - \sigma_{\mathrm{ReLU}}(x)^{q^t} \right| \leq \varepsilon \qquad \text{for all } x \in [-K', K']. \tag{4.2.2}$$

This shows that we can approximate the ReLU to the power of $q^t \geq n-1$. However, our goal is to obtain an approximation of the ReLU raised to the power $n-1$, which could be smaller than q^t. To reduce the order, we emulate approximate derivatives of $\Phi_\varepsilon^{q^t}$. Concretely, we show the following claim: For all $1 \leq p \leq q^t$ for every $K' > 0$ and $\varepsilon > 0$ there exists a neural network Φ_ε^p having $\lceil \log_q(n-1) \rceil$ layers and satisfying

$$|\Phi_\varepsilon^p(x) - \sigma_{\mathrm{ReLU}}(x)^p| \leq \varepsilon \qquad \text{for all } x \in [-K', K']. \tag{4.2.3}$$

The claim holds for $p = q^t$. We now proceed by induction over $p = q^t, q^t - 1, \dots$ Assume (4.2.3) holds for some $p \in \{2, \dots, q^t\}$. Fix $\delta \geq 0$. Then

$$\begin{aligned}&\left|\frac{\Phi^p_{\delta^2}(x+\delta) - \Phi^p_{\delta^2}(x)}{p\delta} - \sigma_{\mathrm{ReLU}}(x)^{p-1}\right| \\ &\qquad \leq 2\frac{\delta}{p} + \left|\frac{\sigma_{\mathrm{ReLU}}(x+\delta)^p - \sigma_{\mathrm{ReLU}}(x)^p}{p\delta} - \sigma_{\mathrm{ReLU}}(x)^{p-1}\right|.\end{aligned}$$

Hence, by the binomial theorem it follows that there exists $\delta_* > 0$ such that

$$\left|\frac{\Phi^p_{\delta_*^2}(x+\delta_*) - \Phi^p_{\delta_*^2}(x)}{p\delta_*} - \sigma_{\mathrm{ReLU}}(x)^{p-1}\right| \leq \varepsilon,$$

for all $x \in [-K', K']$. By Proposition 2.3, $(\Phi^p_{\delta_*^2}(x+\delta_*) - \Phi^p_{\delta_*^2})/(p\delta_*)$ is a neural network with $\lceil \log_q(n-1) \rceil$ layers and size independent from ε. Calling this neural network Φ^{p-1}_ε shows that (4.2.3) holds for $p-1$, which concludes the induction argument and proves the claim.

For every neural network Φ, every spatial translation $\Phi(\cdot - t)$ is a neural network of the same architecture. Hence, every term in the sum (4.1.1) can be approximated to arbitrary accuracy by a neural network of a fixed size. Since by Proposition 2.3, sums of neural networks of the same depth are again neural networks of the same depth, the result follows. □

Next, we extend Proposition 4.7 to the multivariate splines $\mathcal{S}^d_{\ell,\boldsymbol{t},n}$ for arbitrary ℓ, $d \in \mathbb{N}$, $\boldsymbol{t} \in \mathbb{R}^d$.

Proposition 4.8. *Let n, $d \in \mathbb{N}$, $n \geq 2$, $K > 0$, and let $\sigma : \mathbb{R} \to \mathbb{R}$ be sigmoidal of order $q \geq 2$. Further let $\ell \in \mathbb{N}$ and $\boldsymbol{t} \in \mathbb{R}^d$.*

Then, there exists a constant $C > 0$ such that for every $\varepsilon > 0$ there is a neural network $\Phi^{\mathcal{S}^d_{\ell,\boldsymbol{t},n}}$ with activation function σ, $\lceil \log_2(d) \rceil + \lceil \log_q(k-1) \rceil$ layers, and size C, such that

$$\left\| \mathcal{S}^d_{\ell,\boldsymbol{t},n} - \Phi^{\mathcal{S}^d_{\ell,\boldsymbol{t},n}} \right\|_{L^\infty([-K,K]^d)} \leq \varepsilon.$$

Proof. By definition $\mathcal{S}^d_{\ell,\boldsymbol{t},n}(\boldsymbol{x}) = \prod_{i=1}^d \mathcal{S}_{\ell,t_i,n}(x_i)$ where

$$\mathcal{S}_{\ell,t_i,n}(x_i) = \mathcal{S}_n(2^\ell(x_i - t_i)).$$

By Proposition 4.7 there exist a constant $C' > 0$ such that for each $i = 1, \dots, d$ and all $\varepsilon > 0$, there is a neural network $\Phi^{\mathcal{S}_{\ell,t_i,n}}$ with size C' and $\lceil \log_q(n-1) \rceil$ layers such that

$$\left\| \mathcal{S}_{\ell,t_i,n} - \Phi^{\mathcal{S}_{\ell,t_i,n}} \right\|_{L^\infty([-K,K]^d)} \leq \varepsilon.$$

If $d = 1$, this shows the statement. For general d, it remains to show that the product of the $\Phi^{\mathcal{S}_{\ell,t_i,n}}$ for $i = 1, \dots, d$ can be approximated.

We first prove the following claim by induction: For every $d \in \mathbb{N}$, $d \geq 2$, there exists a constant $C'' > 0$, such that for all $K' \geq 1$ and all $\varepsilon > 0$ there exists a neural network $\Phi_{\mathrm{mult},\varepsilon,d}$ with size

C'', $\lceil \log_2(d) \rceil$ layers, and activation function σ such that for all $x_1, \dots, x_d$ with $|x_i| \leq K'$ for all $i = 1, \dots, d$,

$$\left| \Phi_{\text{mult},\varepsilon,d}(x_1, \dots, x_d) - \prod_{i=1}^{d} x_i \right| < \varepsilon. \tag{4.2.4}$$

For the base case, let $d = 2$. Similar to the proof of Proposition 4.7, one can show that there exists $C''' > 0$ such that for every $\varepsilon > 0$ and $K' > 0$ there exists a neural network $\Phi_{\text{square},\varepsilon}$ with one hidden layer and size C''' such that

$$|\Phi_{\text{square},\varepsilon} - \sigma_{\text{ReLU}}(x)^2| \leq \varepsilon \qquad \text{for all } |x| \leq K'.$$

For every $x = (x_1, x_2) \in \mathbb{R}^2$

$$\begin{aligned} x_1 x_2 &= \frac{1}{2}\left((x_1 + x_2)^2 - x_1^2 - x_2^2\right) \\ &= \frac{1}{2}\left(\sigma_{\text{ReLU}}(x_1 + x_2)^2 + \sigma_{\text{ReLU}}(-x_1 - x_2)^2 - \sigma_{\text{ReLU}}(x_1)^2 \right. \\ &\quad \left. - \sigma_{\text{ReLU}}(-x_1)^2 - \sigma_{\text{ReLU}}(x_2)^2 - \sigma_{\text{ReLU}}(-x_2)^2\right). \end{aligned} \tag{4.2.5}$$

Each term on the right-hand side can be approximated up to uniform error $\varepsilon/6$ with a network of size C''' and one hidden layer. By Proposition 2.3, we conclude that there exists a neural network $\Phi_{\text{mult},\varepsilon,2}$ satisfying (4.2.4) for $d = 2$.

Assume the induction hypothesis (4.2.4) holds for $d - 1 \geq 1$, and let $\varepsilon > 0$ and $K' \geq 1$. We have

$$\prod_{i=1}^{d} x_i = \prod_{i=1}^{\lfloor d/2 \rfloor} x_i \cdot \prod_{i=\lfloor d/2 \rfloor + 1}^{d} x_i. \tag{4.2.6}$$

We will now approximate each of the terms in the product on the right-hand side of (4.2.6) by a neural network using the induction assumption.

For simplicity assume in the following that $\lceil \log_2(\lfloor d/2 \rfloor) \rceil = \lceil \log_2(d - \lfloor d/2 \rfloor) \rceil$. The general case can be addressed via Proposition 3.16. By the induction assumption there then exist neural networks $\Phi_{\text{mult},1}$ and $\Phi_{\text{mult},2}$ both with $\lceil \log_2(\lfloor d/2 \rfloor) \rceil$ layers, such that for all x_i with $|x_i| \leq K'$ for $i = 1, \dots, d$

$$\left| \Phi_{\text{mult},1}(x_1, \dots, x_{\lfloor d/2 \rfloor}) - \prod_{i=1}^{\lfloor d/2 \rfloor} x_i \right| < \frac{\varepsilon}{4((K')^{\lfloor d/2 \rfloor} + \varepsilon)},$$

$$\left| \Phi_{\text{mult},2}(x_{\lfloor d/2 \rfloor + 1}, \dots, x_d) - \prod_{i=\lfloor d/2 \rfloor + 1}^{d} x_i \right| < \frac{\varepsilon}{4((K')^{\lfloor d/2 \rfloor} + \varepsilon)}.$$

By Proposition 2.3, $\Phi_{\text{mult},\varepsilon,d} \coloneqq \Phi_{\text{mult},\varepsilon/2,2} \circ (\Phi_{\text{mult},1}, \Phi_{\text{mult},2})$ is a neural network with $1 + \lceil \log_2(\lfloor d/2 \rfloor) \rceil = \lceil \log_2(d) \rceil$ layers. By construction, the size of $\Phi_{\text{mult},\varepsilon,d}$ does not depend on K' or ε. Thus, to complete the induction, it only remains to show (4.2.4).

For all a, b, c, $d \in \mathbb{R}$ holds

$$|ab - cd| \leq |a||b - d| + |d||a - c|.$$

Hence, for $x_1, \dots, x_d$ with $|x_i| \leq K'$ for all $i = 1, \dots, d$, we have that

$$\begin{aligned}
&\left| \prod_{i=1}^{d} x_i - \Phi_{\text{mult},\varepsilon,d}(x_1, \dots, x_d) \right| \\
&\leq \frac{\varepsilon}{2} + \left| \prod_{i=1}^{\lfloor d/2 \rfloor} x_i \cdot \prod_{i=\lfloor d/2 \rfloor + 1}^{d} x_i - \Phi_{\text{mult},1}(x_1, \dots, x_{\lfloor d/2 \rfloor}) \Phi_{\text{mult},2}(x_{\lfloor d/2 \rfloor + 1}, \dots, x_d) \right| \\
&\leq \frac{\varepsilon}{2} + |K'|^{\lfloor d/2 \rfloor} \frac{\varepsilon}{4((K')^{\lfloor d/2 \rfloor} + \varepsilon)} + (|K'|^{\lceil d/2 \rceil} + \varepsilon) \frac{\varepsilon}{4((K')^{\lfloor d/2 \rfloor} + \varepsilon)} < \varepsilon.
\end{aligned}$$

This completes the proof of (4.2.4).

The overall result follows by using Proposition 2.3 to show that the multiplication network can be composed with a neural network comprised of the $\Phi^{\mathcal{S}_{\ell,t_i,n}}$ for $i = 1, \dots, d$. Since in no step above the size of the individual networks was dependent on the approximation accuracy, this is also true for the final network. □

Proposition 4.8 shows that we can approximate a single multivariate B-spline with a neural network with a size that is independent of the accuracy. Combining this observation with Theorem 4.9 leads to the following result.

Theorem 4.9. *Let d, n, $k \in \mathbb{N}$ such that $0 < k \leq n$ and $n \geq 2$. Let $q \geq 2$, and let σ be sigmoidal of order q.*

Then there exists C such that for every $f \in C^k([0,1]^d)$ and every $N \in \mathbb{N}$ there exists a neural network Φ^N with activation function σ, $\lceil \log_2(d) \rceil + \lceil \log_q(k-1) \rceil$ layers, and size bounded by CN, such that

$$\left\| f - \Phi^N \right\|_{L^\infty([0,1]^d)} \leq C N^{-\frac{k}{d}} \|f\|_{C^k([0,1]^d)}.$$

Proof. Fix $N \in \mathbb{N}$. By Theorem 4.9, there exist coefficients $|c_i| \leq C\|f\|_{L^\infty([0,1]^d)}$ and $B_i \in \mathcal{B}^n$ for $i = 1, \dots, N$, such that

$$\left\| f - \sum_{i=1}^{N} c_i B_i \right\|_{L^\infty([0,1]^d)} \leq C N^{-\frac{k}{d}} \|f\|_{C^k([0,1]^d)}.$$

Moreover, by Proposition 4.8, for each $i = 1, \dots, N$ exists a neural network Φ^{B_i} with $\lceil \log_2(d) \rceil + \lceil \log_q(k-1) \rceil$ layers, and a fixed size, which approximates B_i on $[-1,1]^d \supseteq [0,1]^d$ up to error of $\varepsilon := N^{-k/d}/N$. The size of Φ^{B_i} is independent of i and N.

By Proposition 2.3, there exists a neural network Φ^N that uniformly approximates $\sum_{i=1}^{N} c_i B_i$ up to error ε on $[0,1]^d$, and has $\lceil \log_2(d) \rceil + \lceil \log_q(k-1) \rceil$ layers. The size of this network is linear in N (see Exercise 4.11). This concludes the proof. □

Theorem 4.9 shows that neural networks with higher-order sigmoidal functions can approximate smooth functions with the same accuracy as spline approximations while having a comparable number of parameters. The network depth is required to behave like $O(\log(k))$ in terms of the smoothness parameter k, cp. Remark 4.4.

Bibliography and further reading

The argument of linking sigmoidal activation functions with spline based approximation was first introduced in [144, 142]. For further details on spline approximation, see [166] or the book [206].

The general strategy of approximating basis functions by neural networks, and then lifting approximation results for those bases has been employed widely in the literature, and will also reappear again in this book. While the following chapters primarily focus on ReLU activation, we highlight a few notable approaches with non-ReLU activations based on the outlined strategy: To approximate analytic functions, [143] emulates a monomial basis. To approximate periodic functions, a basis of trigonometric polynomials is recreated in [145]. Wavelet bases have been emulated in [169]. Moreover, neural networks have been studied through the representation system of ridgelets [29] and ridge functions [102]. A general framework describing the emulation of representation systems to transfer approximation results was presented in [21].

Exercises

Exercise 4.10. Show that (4.2.1) holds.

Exercise 4.11. Let $L \in \mathbb{N}$, $\sigma\colon \mathbb{R} \to \mathbb{R}$, and let Φ_1, Φ_2 be two neural networks with architecture $(\sigma; d_0, d_1^{(1)}, \dots, d_L^{(1)}, d_{L+1})$ and $(\sigma; d_0, d_1^{(2)}, \dots, d_L^{(2)}, d_{L+1})$. Show that $\Phi_1 + \Phi_2$ is a neural network with $\mathrm{size}(\Phi_1 + \Phi_2) \leq \mathrm{size}(\Phi_1) + \mathrm{size}(\Phi_2)$.

Exercise 4.12. Show that, for $\sigma = \sigma_{\mathrm{ReLU}}^2$ and $k \leq 2$, for all $f \in C^k([0,1]^d)$ all weights of the approximating neural network of Theorem 4.9 can be bounded in absolute value by $O(\max\{2, \|f\|_{C^k([0,1]^d)}\})$.