Chapter 2

Approximation Properties of Neural Networks

Contents

Following up Section 1.4 on the approximation of *specific* functions with neural networks (e.g. $\mathbb{R} \ni x \mapsto x^2$), we now move to a series of constructive and uniform approximation results over *classes* of multivariate functions. That is, given a class $\mathscr{F}$ of functions $f : \mathbb{R}^d \to \mathbb{R}$, we study to what extent they are approximated altogether by neural networks, and what complexity (width, depth, activation function) is necessary to achieve some target precision $\varepsilon > 0$. For sake of simplicity, we focus on the space of Lipschitz functions.

Definition 2.1. *For $d \geq 1$ and $\Omega \subset \mathbb{R}^d$, the space of* 1-Lipschitz *functions over Ω is*

$$\mathrm{Lip}_1(\Omega) := \left\{ f : \Omega \to \mathbb{R}, \forall x, y \in \Omega, |f(x) - f(y)| \leq \|x - y\| \right\},$$

where $\|\cdot\|$ stands for the Euclidean distance.

2.1 Piecewise Constant Activation: A Shallow World

We first examine the case of the piecewise constant Heaviside activation $\rho(u) = \mathbb{1}_{u \geq 0}$, yielding piecewise constant realizations of networks with the following approximation property.

Proposition 2.2 (Upper Bound with Heaviside Activation)**.** *There exists $C_d > 0$ such that the following holds. For all $s \geq C_d$, there exists an architecture of neural networks with at most $\|\Phi\|_0 \leq s$ weights and 2 hidden layers ($L = 3$), denoted by $(\Phi_W)_{W \in \mathbb{R}^s}$, such that with the Heaviside activation $\rho(u) = \mathbb{1}_{u \geq 0}$, we have*

$$\sup_{f \in \mathrm{Lip}_1([0,1)^d)} \inf_{W \in \mathbb{R}^s} \|R(\Phi_W) - f\|_\infty \leq C_d s^{-1/d}.$$

Proof. The idea is to approximate functions $f \in \mathrm{Lip}_1([0,1)^d)$ with neural nets that are piecewise constant over a fixed grid of $[0,1)^d$. More precisely, for all integer $m \geq 1$, consider the regular subdivision of $[0,1]^d$ into m^d closed cubes $(\mathscr{C}^{(j)})_j$ of sidelength $1/m$, and $x^{(j)} \in \mathscr{C}^{(j)}$ for its center. Denote the histogram-like function

$$\tilde{f}(x) := \sum_{j=1}^{m^d} f(x^{(j)}) \mathbb{1}_{\mathscr{C}^{(j)}}(x),$$

for all $x \in [0,1]^d$. As $f \in \mathrm{Lip}_1([0,1)^d)$ and $(\mathscr{C}^{(j)})_j$ is a partition of $[0,1)^d$, we have $\|f - \tilde{f}\|_\infty \leq \sqrt{d}/m$. To end the proof, we will now prove that $\tilde{f}$ can actually be represented as a neural network with Heavyside activation function.

For this, assume for simplicity that $\mathscr{C}^{(j)} = x^{(j)} + \prod_{j=1}^d [0,1/m)$, so that it becomes clear that

$$\begin{aligned}
x \in \mathscr{C}^{(j)} &\Leftrightarrow \forall i \in \{1,\dots,d\}, 0 \leq \langle e_i, x - x^{(j)}\rangle < 1/m \\
&\Leftrightarrow \forall i \in \{1,\dots,d\}, \begin{cases} \langle e_i, x^{(j)}\rangle & \leq \langle e_i, x\rangle \\ 1/m & > \langle e_i, x\rangle \end{cases} \\
&\Leftrightarrow \forall i \in \{1,\dots,d\}, \begin{cases} \rho(\langle e_i, x\rangle - \langle e_i, x^{(j)}\rangle) = 1 \\ 1 - \rho(\langle e_i, x\rangle - 1/m) = 1 \end{cases} \\
&\Leftrightarrow \sum_{i=1}^d \rho(\langle e_i, x\rangle - \langle e_i, x^{(j)}\rangle) + 1 - \rho(\langle e_i, x\rangle - 1/m) \geq 2d \\
&\Leftrightarrow \rho\left(\big(\sum_{i=1}^d \rho(\langle e_i, x\rangle - \langle e_i, x^{(j)}\rangle) - \rho(\langle e_i, x\rangle - 1/m)\big) - d \right) = 1.
\end{aligned}$$

As a result, the last expression shows that $\mathbb{1}_{\mathscr{C}^{(j)}}$ writes as a neural network with two hidden layers, through the exact representation

$$\tilde{f}(x) := \sum_{j=1}^{m^d} f(x^{(j)}) \rho\left(\big(\sum_{i=1}^d \rho(\langle e_i, x\rangle - \langle e_i, x^{(j)}\rangle) - \rho(\langle e_i, x\rangle - 1/m)\big) - d \right).$$

In all, this neural net has two hidden layers, and contains at most

$$\|\Phi\|_0 \leq \underbrace{m^d}_{\sum_j} + \underbrace{m^d}_{f(x^{(j)})} + m^d\big(\underbrace{1}_{-d} + \underbrace{2d}_{\sum_i} + 2d(\underbrace{1}_{1/m, \langle e_i, x^{(j)}\rangle} + \underbrace{d}_{\langle e_i, x\rangle})\big) \leq c_d m^d$$

weights. Taking $s = c_d m^d$ hence yields $\|f - \tilde{f}\|_\infty \leq c'_d s^{-1/d}$, which concludes the proof. □

The above result does not let any fundamental structure of network appear. That is, the two hidden layers of the networks of Proposition 2.3 are nearly fully connected, and their width scales as their number of weights $N_{\max}(\Phi) \asymp \|\Phi\|_0$. With Proposition 1.14 in mind, one may wonder whether deeper neural networks with the same number of weights s couldn't have a better accuracy over $\mathrm{Lip}_1([0,1]^d)$. As proven in the following result, the answer appears to be negative, even with more complicated piecewise constant ρ.

Proposition 2.3 (Lower Bound with Piecewise Constant Activation)**.** *For s large enough, we have the following. For all architecture of neural networks $(\Phi_W)_{W \in \mathbb{R}^s}$ with at most $\|\Phi\|_0 \leq s$ weights, and all piecewise constant activation $\rho : \mathbb{R} \to \mathbb{R}$ with $p + 1 \geq 2$ pieces, we have*

$$\sup_{f \in \mathrm{Lip}_1([0,1]^d)} \inf_{W \in \mathbb{R}^s} \|R(\Phi_W) - f\|_\infty \geq C'_d (s \log(ps))^{-1/d}.$$

Proof. The proof is omitted, as it rests on complexity ideas that will be further developed in Chapter 3. In short, the main ingredients for the proof are the following:

- On one hand, the space $\mathrm{Lip}_1([0,1]^d)$ is rich enough, so that it requires numerous approximating function. More precisely, one can show that there exists $\varepsilon_d > 0$ such that for all class $\mathscr{A}$ of neural networks from $[0,1]^d$ to $\mathbb{R}$,

$$\sup_{f \in \mathrm{Lip}_1([0,1]^d)} \inf_{\Phi \in \mathscr{A}} \|R(\Phi) - f\|_\infty \geq \varepsilon_d \wedge \left(\frac{1}{d_{\mathrm{VC}}(\mathscr{H}_{\mathscr{A}})}\right)^{1/d}, \tag{2.1}$$

  where $d_{\mathrm{VC}}(\mathscr{H}_{\mathscr{A}})$ stands for the VC-dimension of the class of classifiers $\mathscr{H}_{\mathscr{A}} := \left\{\mathrm{sign} \circ R(\Phi)\right\}_{\Phi \in \mathscr{A}}$.

- On the other hand, the class $\mathscr{A}_{\rho_p,s}$ of realisation of neural networks with piecewise constant ρ_p and a limited number $\|\Phi\|_0 \leq s$ of weights yields a VC-dimension (see (3.4))

$$d_{\mathrm{VC}}(\mathscr{H}_{\mathscr{A}_{\rho_p,s}}) \leq c_d s \log(ps).$$

Combining the two bounds yields Proposition 2.3. □

From Proposition 2.3, we see that:

- If a budget s of coefficients is given, the histogram-like construction of Proposition 2.3 yields an optimal approximation of the class $\mathrm{Lip}_1([0,1]^d)$ with Heaviside activation (up to a $\log(s)$ factor).

- Depth does not play any role in the approximation properties of neural networks with piecewise constant activation.

- Adding more values to a piecewise constant activation function can only contributes logarithmically to the expressiveness of the class of networks.

Hence, we now move towards a richer (and more standard) activation function: the Rectified Linear Unit.

2.2 Expressivity of ReLU Networks

This section explains the relations between expressivity, depth, and weight discontinuity of neural networks with ReLU activation $\rho(u) = u_+$.

As seen in Proposition 1.14, shallow (L small) ReLU networks cannot approximate uniformly by shallow non-affine smooth functions, unless their width is large, or vice-versa.

Theorem 2.4 (A First Upper Bound for ReLU)**.** *For s large enough, there exists a neural network architecture $(\Phi_W)_{W \in \mathbb{R}^s}$ with less than s weights, such that with the ReLU activation function function $\rho(u) = u_+$, we have*

$$\sup_{f \in \mathrm{Lip}_1([0,1]^d)} \inf_{W \in \mathbb{R}^s} \|f - R(\Phi_W)\|_\infty \lesssim s^{-1/d}.$$

This architecture can be chosen to consist of $\lesssim N_{\max}(\Phi_W) \lesssim s$ parallel blocks having the same architecture that only depends on d, with depths $L(\Phi_W)$ depending only on d.

Proof. For simplicity of the exposition, we deal with the case $d = 1$. See [Yar18, Proposition 1] for the general case. As for Proposition 2.2, the idea is to use a kernel-like method. Here, as we deal with ReLU networks, we consider the triangle kernel $\phi(x) = \max\{0, \min\{1+x, 1-x\}\} = \rho\big(\min\{1+x, 1-x\}\big)$. As for all $a, b \in \mathbb{R}$,

$$\min\{a,b\} = \frac{1}{2}\big(a+b-|a-b|\big) = \frac{1}{2}\big(a+b-\rho(a-b)-\rho(b-a)\big),$$

we see that ϕ can be represented as a ReLU network with two layers. Now, fix $m \geq 1$. One easily checks that ϕ defines a partition of unity, in the sense that for all $x \in [0,1]$,

$$\sum_{j=0}^{m} \phi(mx - j) = 1,$$

with the support of each $x \mapsto \phi(mx-j)$ equal to $[(j-1)/m, (j+1)/m]$. We then consider the piecewise linear map[1]

$$\tilde{f}_1(x) = \sum_{j=0}^{m} f(j/m)\phi(mx-j).$$

As ϕ has been expressed with ReLU, $\tilde{f}_1(x)$ can be computed with m parallel blocks computing the $\phi(mx-j)$'s, and then combining them together with a final sum weighted by the $f(j/m)$'s[2].

To show that $\|f - \tilde{f}_1\|_\infty$ is small, note that each element $\phi(mx-j)$ of the sum is non-zero if and only if $|x - j/m| < 1/m$, so that we have that $\tilde{f}_1(j/m) = f(j/m)$ for all $j \in \{0, \dots, m\}$. Furthermore $\tilde{f}_1$ is Lipschitz, with gradient equal to $m\big(f(j/m) - f((j+1)/m)\big)$ over the interval $\big(j/m, (j+1)/m\big)$. As a result, $\tilde{f}_1$ has Lipschitz constant at most $m \max_j |\big(f(j/m) - f((j+1)/m)\big)| \leq 1$ since f is 1-Lipschitz. In particular, $\tilde{f}_1$ is 1-Lipschitz, so that $f_2 = f - \tilde{f}_1$ is 2-Lipschitz. To conclude, take $x \in [0,1]$ and $j \in \{1, \dots, m-1\}$ such that $|x - j/m| \leq 1/m$. We write

$$|f(x) - \tilde{f}_1(x)| = |f_2(x)| \leq \underbrace{|f_2(j/m)|}_{=0} + 2|x - j/m| \leq 2/m.$$

The proof is hence complete by taking $s = m$. □

Note that the weights of the network of Theorem 2.4 either do not depend on f, or are of the form $f(x)$ for some $x \in [0,1]^d$. In particular, the weight assignment is continuous in f. In this context of continuous weight assignment, it appears that the expressivity of the neural networks of Theorem 2.4 is optimal. The result holds even beyond neural net classes.

Proposition 2.5 (Importance of Weights discontinuity)**.** *For all* continuous *weight assignment map* $W_s : (\mathrm{Lip}_1([0,1]^d), \|\cdot\|_\infty) \to (\mathbb{R}^s, \|\cdot\|)$ *and all reconstruction map* $\mathscr{R}_s : \mathbb{R}^s \to C([0,1]^d)$,

$$\sup_{f \in \mathrm{Lip}_1([0,1]^d)} \|\mathscr{R}_s(W_s(f)) - f\|_\infty \gtrsim s^{-1/d}.$$

Proof. Follows from a topological argument (Borsuk's antipodality) and a decomposition of Lipschitz functions over a grid. See [DHM89, Theorems 3.1 & 4.2]. □

In addition, the network of Theorem 2.4 is wider as its precision increases ($N_{\max}(\Phi) \asymp \|\Phi\|_0$), but has constant width ($L(\Phi)$ depends only on d). Actually, only considering shallow networks constrains their accuracy, as stated in the following proposition.

[1] For $d > 1$, the kernel ϕ is based on a d-dimensional simplex, and the sum defining $\tilde{f}_1$ ranges over m^d terms.

[2] Slight abuse here: we do not apply ReLU after the linear layer. However, as $t = t_+ - (-t)_+$ for all $t \in \mathbb{R}$, we are good up to adding a single layer with two neurons. See Examples 1.5 and 1.6.

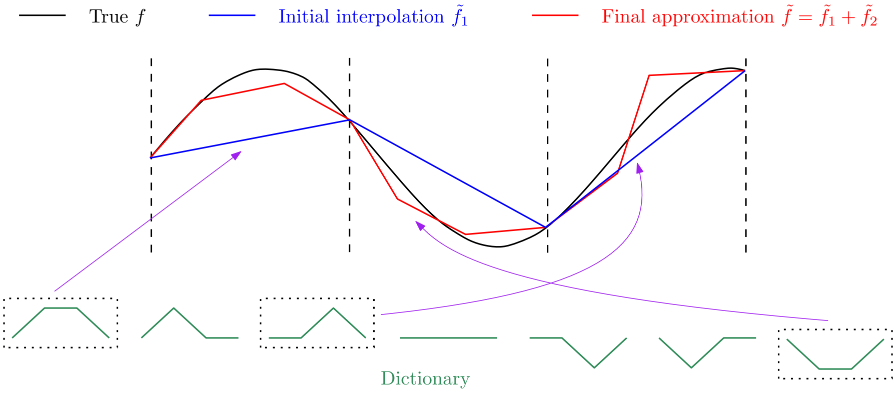
True f
Initial interpolation $\tilde{f}_1$
Final approximation $\tilde{f} = \tilde{f}_1 + \tilde{f}_2$
Dictionary

Figure 2.1: Construction for Theorem 2.7 in dimension $d = 1$.

Proposition 2.6 (Lower Bound on Depth for ReLU)**.** *For all s large enough and all neural network architecture $(\Phi_W)_{W\in\mathbb{R}^s}$ with s weights and ReLu activation $\rho(u) = u_+$, if*

$$\sup_{f\in\mathrm{Lip}_1([0,1]^d)} \inf_{W\in\mathbb{R}^s} \|R(\Phi_W) - f\|_\infty \lesssim s^{-k/d}$$

with $k \in [1,2]$, then this architecture has depth at least

$$\sup_{W\in\mathbb{R}^s} L(\Phi_W) \gtrsim \frac{s^{k-1}}{\log s}.$$

Proof. Follows the same complexity ideas as Proposition 2.3. See (3.4) and [Yar18, Theorem 1]. □

With the two above precision lower bounds, the next natural question is about the actual feasibility of the precision $s^{-k/d}$ with $k \in (1,2]$. We see that any ReLU architecture attaining this rate with $k = 2$ should be very deep (Proposition 2.6), and necessarily be constructed with discontinuous weight assignments in f (Proposition 2.5). It turns out that such a construction is possible.

Theorem 2.7 (Optimal Upper Bound for ReLU)**.** *For s large enough, there exists a neural network architecture $(\Phi_W)_{W\in\mathbb{R}^s}$ with less than s weights, such that with the ReLU activation function function $\rho(u) = u_+$, we have*

$$\sup_{f\in\mathrm{Lip}_1([0,1]^d)} \inf_{W\in\mathbb{R}^s} \|f - R(\Phi_W)\|_\infty \lesssim s^{-2/d}.$$

This architecture has depth $L(\Phi) \asymp s$ and width $N_{\max}(\Phi) \le 2d + 10$.

Proof. Given $f \in \mathrm{Lip}_1([0,1]^d)$, the method consists of first using the network of Theorem 2.4: we get a preliminary piecewise-linear function $\tilde{f}_1$ that interpolates f on a length scale $\asymp s^{-1/d}$. Then, we build a map $\tilde{f}_2$ to approximate the rest $f_2 = f - \tilde{f}_1$ on a smaller length scale $\asymp s^{-2/d}$, by using a finite set of candidate shapes (dictionary), and a fit of one of the shapes in each patch of size $\asymp s^{-1/d}$ through a discrete iterative encoding. The iterative nature of this encoding (bit extraction in base 3 on coordinates of x) is at the origin of the large depth of the final result. The final approximating function is $\tilde{f} = \tilde{f}_1 + \tilde{f}_2$. See Figure 2.1 for insights, and [Yar18, Theorem 2] for a rigorous proof. □

2.3 Curse of Dimensionality & High-dimensional Approximation

At this stage of the chapter, we have discussed two main features of approximation theory with neural networks. First, that the properties of the activation function does impact the capacity of approximation of the associated networks (Proposition 2.3 and Theorem 2.7). Second, that shallow networks can have limited approximation properties compared to deeper ones (Proposition 2.6). However, we do not quite see why neural networks should be preferred as an approximation class, compared to more classical classes such as Fourier sums or Wavelets.

Actually, neural networks adapt more naturally to low-dimensional functional structures. Indeed, the approximation bounds of previous sections deteriorate exponentially with d, as precision over $\mathrm{Lip}_1([0,1]^d)$ are of the form $s^{-k/d}$ with s coefficients. For large d, this yields slow rates as s grows. This phenomenon, called the *curse of dimensionality*, is inherent to $\mathrm{Lip}_1([0,1]^d)$ (or to any reasonable class of functions with d variables), and is not an artefact the neural networks. That is, Fourier and Wavelet series would yield similar bounds.

To overcome this curse of dimensionality, we are hence bound to narrow the study, and consider more specific forms of functions. Indeed, approximating a sum $f(x_1 \dots, x_d) = \sum_{j=1}^d g(x_j)$ of univariate functions depending on each coordinate would require much less coefficients, as it fundamentally amounts to a one-dimensional problem. Such a structural assumption of "sparse dependency" actually is pretty realistic for real data, and is often formalized using submanifolds of $\mathbb{R}^d$.

More generally, it appears that any continuous function can be expressed as a superposition of univariate functions over coordinates. The precise result is the following.

Theorem 2.8 (Kolmogorov Superposition Theorem)**.** *For all $d \geq 1$, there exists $(\lambda_j)_{1\leq j\leq d}$ with $\lambda_j > 0$ and $\sum_{j=1}^d \lambda_j \leq 1$, and $2d+1$ continuous strictly increasing 1-Lipschitz functions $(\chi_i)_{1\leq i\leq 2d+1}$ which map $[0,1]$ to itself, such that every $f \in C([0,1]^d)$ can be represented as*

$$f(x_1,\dots,x_d) = \sum_{i=1}^{2d+1} g_f\Big(\sum_{j=1}^{d} \lambda_j \chi_i(x_j)\Big),$$

for some $g_f \in C([0,1])$ depending on f.

Proof. See [LGM96, p. 553] for a (non-constructive) proof. □

Naturally, Theorem 2.8 is not a free-lunch result: functions g_f and χ_j are intractable to compute and not more regular than continuous. However, it gives us a motivating example to consider simpler classes of d-dimensional functions, for which better approximation rates are hopefully attainable.

Theorem 2.9 (Dimension-Free Upper Bound under Structural Constraints)**.** *Let $\widetilde{\mathrm{F}}_K([0,1]^d)$ be the class of functions $f : [0,1]^d \to \mathbb{R}$ such that there exist $(\lambda_j)_{1\leq j\leq d}$ with $\lambda_j > 0$ and $\sum_{j=1}^d \lambda_j \leq 1$, K $[0,1]$-valued functions $\chi_i \in \mathrm{Lip}_1([0,1])$ ($i \in \{1,\dots,K\}$), and $g_i \in \mathrm{Lip}_1([0,1])$ such that*

$$f(x_1,\dots,x_d) = \sum_{i=1}^{K} g_i\Big(\sum_{j=1}^{d} \lambda_j \chi_i(x_j)\Big),$$

Then for s large enough, there exists a neural network architecture $(\Phi_W)_{W\in\mathbb{R}^s}$ with less than s weights, such that with the ReLU activation function function $\rho(u) = u_+$, we have

$$\sup_{f\in\widetilde{\mathrm{F}}_K([0,1]^d)} \inf_{W\in\mathbb{R}^s} \|f - R(\Phi_W)\|_\infty \lesssim \frac{K^3}{s^2}.$$

This architecture has depth $L(\Phi) \asymp s/K$ and width $N_{\max}(\Phi) \lesssim Kd$.

Proof. Write $(\tilde{g}_j)_{i\leq K}$ and $(\tilde{\chi}_j)_{i\leq K}$ for the realizations of the approximations of $(g_j)_{i\leq K}$ and $(\chi_j)_{i\leq K}$ given by Theorem 2.7 with s' coefficients, for some s' large enough to be chosen later. Note that here, Theorem 2.4 is applied in dimension $d=1$, so that for all $i\leq K$, $\|\tilde{g}_i - g_i\|_\infty \lesssim (1/s')^2$ and $\|\tilde{\chi}_i - \chi_i\|_\infty \lesssim (1/s')^2$, with associated neural network structures independent of d. Furthermore, up to replacing $\tilde{\chi}_i$ by $\min\{(\tilde{\chi}_i)_+, 1\} = (\tilde{\chi}_i)_+ + 1 - \big((\tilde{\chi}_i)_+ - 1\big)_+$ (which only costs two extra ReLU layer), we can assume that $0\leq \tilde{\chi}_i \leq 1$ over $[0,1]$ for all $i\leq K$, with no precision loss since $0\leq \chi_i \leq 1$. Now, consider

$$\tilde{f}(x) := \sum_{i=1}^{K} \tilde{g}_i\Big(\sum_{j=1}^{d} \lambda_j \tilde{\chi}_i(x_j)\Big).$$

Since $g_i \in \mathrm{Lip}_1([0,1])$ for all $i\leq K$, we have

$$\begin{aligned}\|f-\tilde{f}\|_\infty &\leq \sum_{i=1}^{K}\Big\|g_i\Big(\sum_{j=1}^{d}\lambda_j\chi_i(\cdot_j)\Big) - \tilde{g}_i\Big(\sum_{j=1}^{d}\lambda_j\tilde{\chi}_i(\cdot_j)\Big)\Big\|_\infty \\ &\leq \sum_{i=1}^{K}\Big\{\|g_i-\tilde{g}_i\|_\infty + \Big\|\sum_{j=1}^{d}\lambda_j\chi_i(\cdot_j) - \sum_{j=1}^{d}\lambda_j\tilde{\chi}_i(\cdot_j)\Big\|_\infty\Big\}, \\ &\leq \sum_{i=1}^{K}\{\|g_i-\tilde{g}_i\|_\infty + \max_{1\leq j\leq d}\|\chi_i - \tilde{\chi}_i\|_\infty\},\end{aligned}$$

where the last line uses that $\sum_{j=1}^{d}\lambda_j \leq 1$ and $\lambda_j \geq 0$. As a result, we obtain $\|f-\tilde{f}\|_\infty \lesssim K/(s')^2$.

Finally, we note that $\tilde{f}$ can realized by stacking all the networks that compute the $\tilde{\chi}_i(x_j)$ on the first layer (depth $\asymp s'$, width $\lesssim Kd$ in total), then computing $\sum_{j=1}^{d}\lambda_j\tilde{\chi}_i(\cdot_j)$ with one linear layer[3], feeding this layer to $\tilde{g}_i$ for all $1\leq i\leq K$ in parallel (depth $\asymp s'$, width $\lesssim K$ in total) and then summing again the results. Putting everything together, we obtain $\|f-\tilde{f}\|_\infty \lesssim K/(s')^2$ with $\lesssim s'K$ neurons and depth $\lesssim s'$. Taking $s = s'K$ yields the announced result. □

As will be developed in Chapter 4 in more details, functions that can be represented with composition structures of regular maps, each with a few variables $d_0 \ll d$, yield approximation rates that depend on d_0 and not (or mildly) on d.

Exercise 2.10. *Derive a result similar to Theorem 2.9, over the class of functions $f:[0,1]^4\to\mathbb{R}$ that can be expressed as*

$$f(x_1,x_2,x_3,x_4) = f_3\big(f_{2,1}(x_1) + f_{2,2}(x_2,x_3) + f_{2,3}(x_3, f_1(x_1,x_2))\big),$$

where all the functions appearing are real-valued and 1-Lipschitz.

Exercise 2.11. *Given a fixed modulus of continuity $\omega:\mathbb{R}_+\to\mathbb{R}_+$, reproduce the proofs of Proposition 2.2 and Theorem 2.4 for the class of function*

$$\mathrm{F}_\omega([0,1]^d) := \Big\{f:[0,1]^d\to\mathbb{R}, \forall x,y\in[0,1]^d, |f(x)-f(y)|\leq \omega\big(\|x-y\|\big)\Big\}$$

Bibliography

[DHM89] Ronald A. DeVore, Ralph Howard, and Charles Micchelli. Optimal nonlinear approximation. *Manuscripta Math.*, 63(4):469–478, 1989.

[3] Same abuse as page 18.

[LGM96] George G. Lorentz, Manfred v. Golitschek, and Yuly Makovoz. *Constructive approximation*, volume 304 of *Grundlehren der mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1996. Advanced problems.

[Yar18] Dmitry Yarotsky. Optimal approximation of continuous functions by very deep relu networks. In Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet, editors, *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, pages 639–649. PMLR, 06–09 Jul 2018.