monomial $(x,y) \to xy$ using a ReLU network. From there one then shows that ReLU networks suitably approximate products $x \to x_1 \cdots x_d$ as well as more general monomials. From monomials one can easily approximate polynomials by combining networks, and now one can connect to the first part of the argument, by constructing a network that approximates the piecewise polynomial function mentioned above, that itself approximates $f_0$.

*Approximation, Step 1.* In order to approximate a function $f \in \mathscr{C}_d^\beta([0,1]^d, K)$, we define a grid of $[0,1]^d$ as, for $M \geq 1$ an integer to be chosen below,

$$D(M) = \left\{ x_l = \left(\frac{l_j}{M}\right)_{j=1,\dots,d}, \quad l = (l_1,\dots,l_d) \in \{0,1,\dots,M\}^d \right\}.$$

Around a given point $\boldsymbol{a} \in [0,1]^d$, the function $f$ can be approximated by its Taylor polynomial: in dimension $d$ its expression is, for $\boldsymbol{a} = (a_1,\dots,a_d)$, denoting $\alpha! = \alpha_1 \cdots \alpha_d$, and $u^\alpha = u_1^{\alpha_1} \cdots u_d^{\alpha_d}$ for $u \in \mathbb{R}^d$,

$$P_{\boldsymbol{a}}^\beta f(x) := \sum_{0 \leq |\alpha| < \beta} (\partial^\alpha f)(\boldsymbol{a}) \frac{(x-\boldsymbol{a})^\alpha}{\alpha!}. \tag{4.6}$$

Taylor's expansion with Lagrange remainder gives, for any $f \in \mathscr{C}_d^\beta([0,1]^d, K)$

$$|f(x) - P_{\boldsymbol{a}}^\beta f(x)| \leq K \|x - \boldsymbol{a}\|_\infty^\beta. \tag{4.7}$$

Let us check (4.7). By Taylor's formula there exists $\xi \in [0,1]$ such that

$$f(x) = \sum_{0 \leq |\alpha| < \beta - 1} (\partial^\alpha f)(\boldsymbol{a}) \frac{(x-\boldsymbol{a})^\alpha}{\alpha!} + \sum_{\beta - 1 \leq |\alpha| < \beta} (\partial^\alpha f)\,(\boldsymbol{a} + \xi(x - \boldsymbol{a}))\, \frac{(x-\boldsymbol{a})^\alpha}{\alpha!},$$

so subtracting (4.6) and using the triangle inequality gives

$$\begin{aligned} |f(x) - P_{\boldsymbol{a}}^\beta f(x)| &\leq \sum_{\beta - 1 \leq |\alpha| < \beta} \left| (\partial^\alpha f)\,(\boldsymbol{a} + \xi(x - \boldsymbol{a})) - (\partial^\alpha f)(\boldsymbol{a}\right| \frac{\|x - \boldsymbol{a}\|_\infty^{|\alpha|}}{\alpha!} \\ &\leq K|\xi| \|x - \boldsymbol{a}\|_\infty^{\beta - \lfloor \beta \rfloor} \sum_{\beta - 1 \leq |\alpha| < \beta} \frac{1}{\alpha!} \|x - \boldsymbol{a}\|_\infty^{\lfloor \beta \rfloor} \leq K \|x - \boldsymbol{a}\|_\infty^\beta, \end{aligned}$$

using the fact that $f$ is $\beta$–Hölder.

Define, again for any $f \in \mathscr{C}_d^\beta([0,1]^d, K)$ and $x = (x_1,\dots,x_d)$,

$$P^\beta f(x) := \sum_{x_l \in D(M)} (P_{x_l}^\beta f)(x) \prod_{j=1}^d (1 - M|x_j - x_{l,j}|)_+. \tag{4.8}$$

Inside the hypercubes defined by consecutive gridpoints, $P^\beta f(x)$ is a polynomial, so the overall function $P^\beta f$ is piecewise–polynomial.

**Lemma 4.7** (Approximation of $f$ by a piecewise–polynomial function)**.** *For any $f \in \mathscr{C}_d^\beta([0,1]^d, K)$, define $P^\beta f$ as in (4.8). Then*

$$\|f - P^\beta f\|_\infty \leq K M^{-\beta}.$$

*Proof.* Observe the following sum-product formula (expand the middle term)

$$\sum_{x_l=(l_1/M,\dots,l_d/M)} \prod_{j=1}^{d} (1-M|x_j - x_{l,j}|)_+ = \prod_{j=1}^{d} \sum_{l=0}^{M} (1-M|x_j - l/M|)_+ = 1.$$

Indeed, if $l^* = \lfloor Mx_j \rfloor$, then $(1-M|x_j - l/M|)_+$ is possibly non-zero only for $j = l^*, l^*+1$ and

$$(1-M|x_j - l^*/M|)_+ + (1-M|x_j-(l^*+1)/M|)_+ = 1 - M(x_j - l^*/M) + 1 - M((l^*+1)/M - x_j) = 1.$$

One notes that the terms of the sum in the definition (4.8) are nonzero only at a given $x$ for $x_l$ such that $\|x - x_l\|_\infty \le 1/M$ – the corners of the hypercube of radius $1/M$ the point $x$ belongs to –, otherwise the product in (4.8) is zero. Denoting by $\mathscr{H}_{x_l}(x) = \prod_{j=1}^{d}(1-M|x_j - x_{l,j}|)_+$,

$$\left| f(x) - P^\beta f(x) \right| \le \sum_{x_l} |f(x) - (P^\beta_{x_l} f)(x)| \mathscr{H}_{x_l}(x) \le \max_{\|x-x_l\|_\infty \le 1/M} |f(x)-(P^\beta_{x_l} f)(x)| \sum_{\|x-x_l\|_\infty \le 1/M} \mathscr{H}_{x_l}(x) \le KM^{-\beta},$$

using Taylor's approximation (4.7), which concludes the proof. □

*Approximation, Step 2 (specific to ReLU activation).*

**Lemma 4.8** (Approximating $x(1-x)$ with piecewise affine functions)**.** *Let $T^1 : [0,1] \to [0,1/4]$ and more generally $T^k : [0, 2^{-2(k-1)}] \to [0, 2^{-2k}]$, $k \ge 1$, be the maps*

$$T^1(x) = \frac{x}{2} \wedge \left(\frac{1}{2} - \frac{x}{2}\right), \quad T^k(x) = \frac{x}{2} \wedge \left(\frac{1}{2^{2k-1}} - \frac{x}{2}\right).$$

*Let us set $R^k := T^k \circ T^{k-1} \circ \cdots \circ T^1$, for $k \ge 1$. Then for any $m \ge 1$,*

$$\left| x(1-x) - \sum_{k=1}^{m} R^k(x) \right| \le 4^{-m-1}.$$

*Proof.* Let $C(x) = x(1-x)$. The key is to observe the 'fractal'-like property

$$C(x) = T^1(x) + \frac{1}{4} C(4T^1(x)).$$

This can be seen on a picture or just checking algebraically. Next note that by definition $T^2(y) = \frac{1}{4}T^1(4y)$ and more generally $T^{k+1}(y) = \frac{1}{4^k}T^1(4^k y)$. By recursion one immediately obtains

$$C(x) = T^1(x) + T^2 \circ T^1(x) + \cdots + T^k \circ \cdots \circ T^1(x) + \frac{1}{4^k} C(4^k T^k \circ \cdots \circ T^1(x)).$$

The result follows by applying this with $k = m$ and noting that $C(\cdot)$ is bounded by $1/4$ on $[0,1]$. □

**Lemma 4.9** (Approximating $(x,y) \to xy$ by a DNN)**.** *Let $m \ge 1$. There exists a DNN, $Mult_m(x,y)$, with*

$$Mult_m \in \mathscr{F}(m+4, (2,6,\cdots,6,2,2,2,1)),$$

*such that for any $x, y \in [0,1]$ it holds $Mult_m(x,y) \in [0,1]$, $Mult_m(0,y) = Mult_m(x,0) = 0$ and*

$$\left| Mult_m(x,y) - xy \right| \le 4^{-m}.$$

*Proof.* In order to approximate $(x, y) \to xy$, we use a 'polarisation' formula. The most classical polarisation writes $xy$ in terms of squares as $xy = (x+y)^2/4 - (x-y)^2/4$. Here we rather use, since from Lemma 4.8 we have access to $x(1-x) =: C(x)$, the formula

$$xy = C\left(\frac{x-y+1}{2}\right) - C\left(\frac{x+y}{2}\right) + \frac{x+y}{2} - \frac{1}{4},$$

(verify it!) where we know how to approximate every element on the right hand–side by affine functions. Denoting $C_m := \sum_{k=1}^{m} R^k$, Lemma 4.9 gives $\|C_m - C\|_\infty \leq 4^{-m-1}$. Let us further denote

$$Z_m(x, y) := \left(C_m\left(\frac{x-y+1}{2}\right) + \frac{x+y}{2} - C_m\left(\frac{x+y}{2}\right) - \frac{1}{4}\right)_+ \wedge 1. \tag{4.9}$$

Then, since the map $(u, v) \to \Delta(u, v) := (u-v)_+ \wedge 1$ is 1–Lipschitz, we have

$$|Z_m(x, y) - xy| \leq 2\|C - C_m\|_\infty \leq 4^{-m}/2.$$

Let us set $\mathrm{Mult}_m(x, y) := Z_m(x, y)$ and see how to practically implement this into a ReLU network.

*Basic networks for $T_+, T_-^k, T^k, \Delta$.* Recall $T^k(x) = (x/2) \wedge (2^{1-2k} - x/2)$. Since $a \wedge b = a_+ - (a-b)_+$ if $a \geq 0$,

$$T^k(x) = T_+(x) - T_-^k(x), \qquad T_+(x) := (x/2)_+, \;\; T_-^k(x) := (x - 2^{1-2k})_+.$$

So, the functions $T_+, T_-$ and $T^k$ for $k \geq 1$ can all be written with a shallow network using only one neuron, as shown in Figure 4.1. Finally, using that $\Delta(u, v) = 1 \wedge (u-v)_+ = 1 - (1 - (u-v)_+)_+$, one can easily encode $\Delta$ using a ReLU network with two hidden layers.

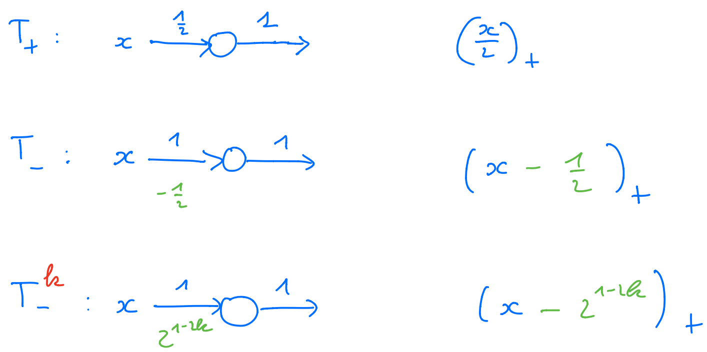


Figure 4.1: DNN representation of basic $T$ functions encoding (on top of arrows: matrix coefficients; below arrows in green: non-zero translations; circle: neuron pass)

*Combining basic networks to compute $Z_m(x, y)$.* The basic networks can then be combined according to the network depicted in Figure 4.2, which taking as input $(T_+(x), h(x), T_-(x))$, computes $C_m(x) + h(x)$, where $h : [0,1] \to [0,\infty)$ is a given function taking nonnegative values. Note that the neuron pass on the middle row in Figure 4.2 is just the identity, as the input is always nonnegative.

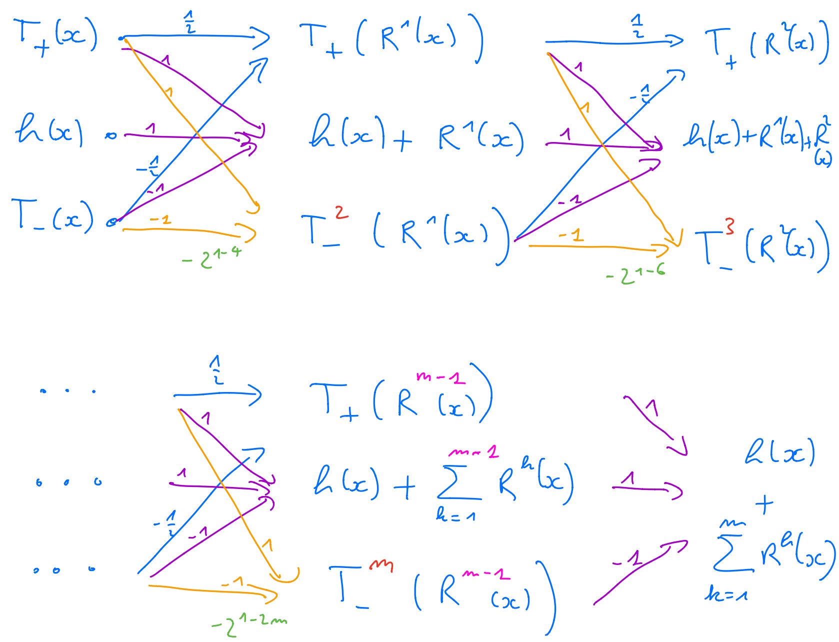

Figure 4.2: Encoding approximation of $h(x)+C(x)$, with $C(x)=x(1-x)$ with ReLU (on top of arrows: matrix coefficients; below arrows in green: non-zero translations; formulas display computed quantity just after neuron pass, except last one for final output, for which we do not apply a neuron)

It is then enough to run two sub-networks in parallel: a first computes

$$(x,y)\to\left(T_+\left(\frac{x-y+1}{2}\right),\frac{x+y}{2},T_-\left(\frac{x-y+1}{2}\right)\right)$$

and applies the network $N_m$ from Figure 4.2, thus computing $C_m\left(\frac{x-y+1}{2}\right)+\frac{x+y}{2}$. A second sub-network does the same after first computing

$$(x,y)\to\left(T_+\left(\frac{x-y}{2}\right),\frac{1}{4},T_-\left(\frac{x+y}{2}\right)\right),$$

thus computing $C_m\left(\frac{x+y}{2}\right)+\frac{1}{4}$. Finally applying $\Delta$ to the two sub-results provides a computation of $Z_m(x,y)$ as required.

To conclude the proof, one checks that the number of parameters is as required, and that the function $\mathrm{Mult}_m(x,y)$ is indeed 0 for inputs of the form $(x,0)$ and $(0,y)$ (this is checked on successive functions $R^1,R^2,\ldots,R^m$ and left as an exercise). □

More generally, the next lemma shows, first, how to construct a DNN approximating the product $x_1\cdots x_r$. Second, one can similarly approximate any monomial, that is a polynomial of the form $x_1^{\alpha_1}\cdots x_r^{\alpha_r}$, and by synchronising the resulting networks, all monomials up to a certain degree $\gamma$ *simultaneously*. Let us write, for $f$ a $\mathbb{R}^{N_L}$–valued function, $\|f\|_\infty$ as a shorthand for $\||f|_\infty\|_\infty$.

**Lemma 4.10** (Approximating products and monomials by a DNN)**.** *Let $m,r\geq 1$ two integers. There exists a DNN, $\mathit{Mult}_m^r:[0,1]^r\to[0,1]$, with depth $L\lesssim m\log r$ and maximum width $6r$ such that for*

$x = (x_1, \dots, x_r) \in [0,1]^r$,

$$\left| Mult_m^r(x) - \prod_{i=1}^r x_i \right| \le r^2 4^{-m},$$

*and $Mult_m^r(x) = 0$ if one of the $x_i$'s is zero. More generally, let $m, \gamma, d \ge 1$ three integers. Let $C_{d,\gamma}$ denote the number of monomials over $d$ variables with degree $|\boldsymbol{\alpha}| < \gamma$. There exists a DNN, $Mon_{m,\gamma}^d : [0,1]^d \to [0,1]^{C_{d,\gamma}}$ with depth $L \lesssim m \log \gamma$ and maximum width $12\gamma C_{d,\gamma}$, that approximates all monomials of degree less than $\gamma$* simultaneously

$$\left\| Mon_{m,\gamma}^d(x) - \left(x^{\boldsymbol{\alpha}}\right)_{|\boldsymbol{\alpha}|<\gamma} \right\|_\infty \le \gamma^2 4^{-m}.$$

*Remark.* The number $C_{d,\gamma}$ of monomials of degree less than $\gamma$ is less than $(\gamma+1)^d$.

*Proof.* We start by the network $\mathrm{Mult}_m^r$ and notice that one can always assume that $r$ is a power of 2. If this is not the case, one just artificially extends the product by multiplying by a number of 1's. So, let us set $r = 2^q$. An approximation to the product of $x_i$'s is computed recursively as follows: first compute

$$(\mathrm{Mult}_m(x_1, x_2), \mathrm{Mult}_m(x_3, x_4), \dots, \mathrm{Mult}_m(x_{r-1}, x_r)),$$

which gives $2^{q-1}$ terms left, and then repeat the same operation until there is only one term left, with an output that we define as $\mathrm{Mult}_m^r(x)$. By Lemma 4.9 and the triangle inequality, for $a, b, c, d \in [0,1]$,

$$|\mathrm{Mult}_m(a,b) - cd| \le |\mathrm{Mult}_m(a,b) - ab| + |(a-c)b + (b-d)c| \le 4^{-m} + |a-c| + |b-d|.$$

An immediate recursion then gives, as announced,

$$\left|\mathrm{Mult}_m^r(x) - (x_1 \cdots x_r)\right| \le 3^{q-1} 4^{-m} \le 4^q 4^{-m} = r^2 4^{-m}.$$

Now turning to the network computing all monomials, one notes that for a degree of at most 1, one can just use a shallow network, with the later computing exactly a constant or a linear function (recall the identity can be obtained as such a network). More generally, one uses the same argument as for $\mathrm{Mult}_m^r$ to compute a given monomial $x_1^{\alpha_1} \cdots x_r^{\alpha_r}$, up to an error, recalling $|\boldsymbol{\alpha}| = \sum_{i=1}^r \alpha_i < \gamma$, bounded from above by $\gamma^2 4^{-m}$. In a last step, we stack all obtained networks in parallel (using depth synchronisation to have the same given depth for all networks, meaning we take the largest depth).

□

*End of the proof of Theorem 4.5.* Now that we have constructed a network computing all monomials, one can go back to the local polynomial (4.8) approximating $f$, namely

$$P^\beta f(x) := \sum_{x_l \in D(M)} (P_{x_l}^\beta f)(x) \prod_{j=1}^d (1 - M|x_j - x_{l,j}|)_+.$$

Let us define $M$ as the largest integer such that

$$(M+1)^d \le \mathcal{N}. \tag{4.10}$$

To conclude the proof, one constructs the final network in three steps.

*Step (i), hat function network.* One constructs a network $\mathrm{Hat}^d$ approximating the hat functions $\prod_{j=1}^d (M^{-1} - |x_j - x_{l,j}|)_+$ (note the specific normalisation, in order to have an easy construction with weights bounded by 1) *simultaneously* for all $x_\ell$ on the grid $D(M)$.

Since $|x| = x_+ + (-x)_+$, we have the formula, for $a, b, c$ in $[0,1]$,

$$(a - |b-c|)_+ = (a - (b-c)_+ - (c-b)_+)_+.$$

One can use a first hidden layer with width $2d(M+1)^d$ to compute all functions $(x_j - \ell/M)_+$ and $(\ell/M - x_j)_+$ (for $j$ in $1,\dots,d$ and $x_\ell$ in $D(M)$ of cardinality $(M+1)^d$), and a second hidden layer to compute all functions $(1/M - |x_j - \ell/M|)_+$ using the formula in the last display, using this time a width $d(M+1)^d$. All these functions take values in $[0,1]$. Also, the overall sparsity is proportional to the width, as all computations are done in parallel.

If $d = 1$ we are done (the network computes the function exactly). For $d > 1$, one uses the networks $\mathrm{Mult}_m^d$ from Lemma 4.10 to compute the desired products. Each one of these products requires (recalling $\mathrm{Mult}_m^d$ has width less than $Cd$, depth at most $Cm\log d$) at most $Cmd^2\log d$ nonzero parameters. We have of $(M+1)^d$ of these products in parallel which gives sparsity $C'm(M+1)^d d^2\log d \lesssim m\mathcal{N}$ in total (adding also the non-zero parameters of the first two layers from the previous paragraph, which require only $Cd(M+1)$ non-zero weights). The resulting network verifies

$$\Big|\mathrm{Hat}^d(x) - \Big(\prod_{j=1}^d (M^{-1} - |x_j - x_{l,j}|)_+\Big)_{x_l\in D(M)}\Big|_\infty \le d^2 4^{-m}. \tag{4.11}$$

*Step (ii), networks $Q_1$ and $Q_2$.* We now build two networks verifying the following. For $B = 3Ke^d$, we have $Q_1(x) \in [0,1]^{(M+1)^d}$ and

$$\Big|Q_1(x) - \Big(\frac{P_{x_l}^\beta f(x)}{B} + \frac{1}{2}\Big)_{x_l\in D(M)}\Big| \le \beta^2 4^{-m}. \tag{4.12}$$

The role of the constant $B$ is to keep the approximated quantity in the last display between 0 and 1, thanks to Lemma 4.11. Next, the network $Q_2$ verifies

$$\Big|Q_2(x) - \sum_{x_l\in D(M)} \Big(\frac{P_{x_l}^\beta f(x)}{B} + \frac{1}{2}\Big)\prod_{j=1}^d (M^{-1} - |x_j - x_{l,j}|)_+\Big| \le (1 + d^2 + \beta^2)4^{d-m}. \tag{4.13}$$

The construction of $Q_1$ is immediate by forming the weighted sum of the joint network of monomials $\mathrm{Mon}_{m,\gamma}^d$ for every point $x_l \in D(M)$ in parallel (thus getting an output dimension $(M+1)^d$), noting with Lemma 4.11 that the weights are smaller than 1 thanks to the division by $B$. By Lemma 4.10, the depth of $Q_1$ is bounded by $Cm$. The sparsity of $Q_1$ is bounded by that of $\mathrm{Mon}_{m,\gamma}^d$ plus $C_{d,\beta}(M+1)^d$, that is by $Cm + C(M+1)^d \le C(m+\mathcal{N})$, where $C$ depends only on $d, \beta$.

To build $Q_2$, one proceeds in two steps: first, one stacks in parallel the networks $Q_1$ and the $\mathrm{Hat}^d$ network (which simultaneously outputs all hat functions). Next, one notes that both $Q_1$ and $\mathrm{Hat}^d$ have outputs indexed by $x_l$. One pairs these outputs two-by-two and applies to them the $\mathrm{Mult}_m$ network. There are $(M+1)^d \le \mathcal{N}$ pairs, and recall that $\mathrm{Mult}_m$ has of order $m$ active (nonzero) parameters (depth $m$ and constant width). So this part of the network has sparsity at most $m\mathcal{N}$. Finally, one adds all results using a final layer, leading to the term approximating $Q_2$ in (4.13), except that hat functions and local polynomial are replaced by their approximations. Combining (4.11) with (4.12) now gives (4.13). In passing we note that building $C_2$ uses at most $Cm\mathcal{N}$ nonzero parameters, with depth of order $m$.

*Step (iii), shifting and rescaling the entries in* (4.13). Finally, we build a network $Q_3$ that verifies

$$\Big|Q_3(x) - \sum_{x_l\in D(M)} P_{x_l}^\beta f(x) \prod_{j=1}^d (1 - M|x_j - x_{l,j}|)_+\Big| \le (2K+1)(1 + d^2 + \beta^2)(2e)^d \mathcal{N} 4^{-m}. \tag{4.14}$$

The construction of $Q_3$ is based on shifting/rescaling $Q_2$. One needs to pay attention that we wish to keep weights between 0 and 1. Note that to compute $Q_3$ by putting $Q_2$ 'on the right scale', it suffices to build a network computing the scaling $x \to BM^r x =: \mathscr{K} x$. To do so, one may use a network with a weight matrix having all entries equal to 1 and zero shift vectors, which uses $C\mathscr{K} \lesssim \mathscr{N}$ active parameters. The overall network $Q_3$ thus keeps up to a constant the same sparsity (at most $Cm\mathscr{N}$) and depth as $Q_2$. The approximation (4.14) directly follows from (4.13).

Putting (4.14) together with Lemma 4.7, one obtains that the network $Q_3$ verifies

$$\|f - Q_3\|_\infty \lesssim \mathscr{N} 4^{-m} + M^{-\beta} \lesssim \mathscr{N} 4^{-m} + \mathscr{N}^{-\beta/d},$$

with a sparsity $Cm\mathscr{N}$ (we omit the precise dependence of the constants on $K, d$ given in the statement, which can easily be tracked in the above arguments) which concludes the proof of Theorem 4.5.

**Lemma 4.11.** *Let $P_{\boldsymbol{\alpha}}^{\beta} f$ the polynomial defined in (4.6). Writing $P_{\boldsymbol{\alpha}}^{\beta} f(x) = \sum_{0 \le |\gamma| \le \beta} c_\gamma x^\gamma$, for any $f \in \mathscr{C}_d^{\beta}([0,1]^d, K)$,*

$$\sup_{x \in [0,1]^d} \left| (P_{\boldsymbol{\alpha}}^{\beta} f)(x) \right| \le \sum_{0 \le |\gamma| \le \beta} |c_\gamma| \le K e^d.$$

*Proof.* Left as an exercise, see [SH20a], page 6. □

### 4.3.2 Ingredient 2 : entropy and error propagation in DNNs

Let $\rho$ be the ReLU activation, or more generally a 1–Lip function with $\rho(0) = 0$. Considering the class of functions $\mathscr{F}(L, N, s)$, let us denote

$$V = V(N) := \prod_{l=0}^{L} (N_l + 1). \tag{4.15}$$

Since $\rho$ is fixed throughout, we write $R(W)$ for $R(\Phi)$. The next lemma quantifies how much small errors in network parameters propagate into a global error for the network realisation.

**Lemma 4.12.** *Suppose $f = R(W)$ and $f^* = R(W^*)$ belong to $\mathscr{F}(L, N)$ with $W = (A_k, b_k)_{k=1,\dots,L}$ and $W^* = (A_k^*, b_k^*)_{k=1,\dots,L}$. Suppose that individual entries of $A_k$'s and $b_k$'s are at most $\varepsilon > 0$ away from the corresponding entries of $A_k^*$ and $b_k^*$. Then for $V$ as in (4.15),*

$$\|f - f^*\|_\infty \le \varepsilon L V.$$

*Proof.* Recall $f = T_L \circ \rho \circ \cdots \circ \rho \circ T_1$ with $T_k(x) = A_k x + b_k$ and define, for $k = 1, \dots, L$,

$$B_k f = \rho \circ T_k \circ \cdots \circ \rho \circ T_1,$$
$$E_k f = T_L \circ \rho \circ \cdots \circ T_{k+1} \circ \rho,$$

and set $E_L f = B_0 f = \mathrm{Id}$. We first prove two basic facts about $B_k f, E_k f$.

*Fact 1.* If $f \in \mathscr{F}(L, N)$, then $|(B_k f)(x)|_\infty \le \prod_{l=1}^{k} (N_{l-1} + 1)$ for $x \in [0,1]^d$.

Let us check first that $|(\rho \circ T_i)(y)|_\infty \le N_{i-1} |y|_\infty + 1$ for any integer $i$. Indeed, $|\rho(y)|_\infty \le |y|_\infty$ and $|T_k y|_\infty \le |A_k y|_\infty + |b_k|_\infty \le N_{k-1} |y|_\infty + 1$, using $\|A_k\|_\infty \le 1, |b_k|_\infty \le 1$. In particular, if $|y|_\infty \ge 1$ we have $|(\rho \circ T_i)(y)|_\infty \le (N_{i-1} + 1) |y|_\infty$ for any $i$.

The result follows by recursion: for $i = 1$ we get $|(\rho \circ T_1)(x)|_\infty \le N_0|x|_\infty + 1 \le N_0 + 1$. Since $N_0 + 1 \ge 1$ it suffices feeds this bound into the previous inequality in terms of $y$.

*Fact 2.* The map $x \to (E_k f)(x)$ is $\Lambda_k$–Lipschitz, with $\Lambda_k \le \prod_{l=k+1}^{L} N_{l-1}$.

The composition of an $L_1$–Lip by an $L_2$–Lip function is an $L_1L_2$–Lip function. By definition $\rho$ is 1–Lip, while $T_i$ is $N_{i-1}$–Lip for any $i$, from which the fact follows.

Now let us write the difference $f - f^*$ as the telescopic sum

$$f(x) - f^*(x) = \sum_{k=1}^{L} \Big[(E_k f) \circ T_k \circ (B_{k-1} f^*)(x) - (E_k f) \circ T_k^* \circ (B_{k-1} f^*)(x)\Big].$$

Combining the triangle inequality with Fact 2 above,

$$\begin{aligned}
|f(x) - f^*(x)| &\le \sum_{k=1}^{L} \Lambda_k \left|(T_k - T_k^*) \circ (B_{k-1} f^*)(x)\right|_\infty \\
&\le \sum_{k=1}^{L} \Lambda_k \left[\|A_k - A_k^*\|_\infty |(B_{k-1} f^*)(x)|_1 + |b_k - b_k^*|_\infty\right] \\
&\le \sum_{k=1}^{L} \Lambda_k \left[\varepsilon N_{k-1} |(B_{k-1} f^*)(x)|_\infty + \varepsilon\right].
\end{aligned}$$

The term under brackets in the last display is at most, using Fact 1,

$$\varepsilon N_{k-1} \prod_{l=1}^{k-1} (N_{l-1} + 1) + 1 \le \varepsilon \prod_{l=1}^{k} (N_{l-1} + 1).$$

One deduces the announced result

$$|f(x) - f^*(x)| \le \varepsilon \sum_{k=1}^{L} \prod_{l=1}^{L} (N_{l-1} + 1) \le \varepsilon L V.$$

□

The previous lemma allows for a "quantisation" of the set of neural network realisations: in the next result we explicitly construct a finite set of functions (themselves NNs) that cover it.

**Lemma 4.13.** *For $V$ as in (4.15) and any $\delta > 0$,*

$$\log \mathrm{N}(\delta, \mathscr{F}(L, N, s), \|\cdot\|_\infty) \le (s+1) \log\left(\frac{2LV^2}{\delta}\right).$$

*In particular if $L \lesssim \log n$ and $N_l \le n$ for any integer $l$, we have*

$$\log \mathrm{N}(\delta, \mathscr{F}(L, N, s), \|\cdot\|_\infty) \lesssim s\left[(\log n)^2 + \log(1/\delta)\right].$$

The proof has been given in Chapter 3.

### 4.3.3 A generic oracle inequality for the prediction risk

Let us now prove Theorem 4.6. It is helpful to relate the prediction risk $R(\hat f, f_0) = E\left[(\hat f(X) - f_0(X))^2\right]$ to the empirical risk (4.3). The proof will be complete once we show

$$R(\hat f, f_0) \le (1+\varepsilon)\hat R(\hat f, f_0) + \frac{1+\varepsilon}{\varepsilon^2} 3\frac{F^2}{n}\log \mathcal{N}_n + (1+\varepsilon)\delta\frac{F}{n}, \tag{4.16}$$

as well as the following direct bound on the empirical risk, for $F \ge 1$,

$$\hat R(\hat f, f_0) \le (1+\varepsilon)\left\{\inf_{f\in\mathcal{F}} R(f, f_0) + 3\frac{1+\varepsilon}{\varepsilon}\frac{F^2}{n}\log\mathcal{N}_n + F\delta\right\}. \tag{4.17}$$

As follows from the proofs below, the upper-bound (4.16) of the prediction risk by the 'empirical' risk holds for *any* estimator $\hat f$, not necessarily the ERM. The bound (4.17) uses crucially that $\hat f$ is the ERM.

*Proof of* (4.16). (a) Let us cover $\mathcal{F}$ by $N := \mathcal{N}_n$ balls of radius $\delta$ and centers $f_1, \dots, f_N$. One may assume $\|f_i\|_\infty \le F$ (otherwise consider balls centered at $\bar f_i = (f_i \wedge F) \vee (-F)$ instead).

Let $j^*$ be a random integer such that $\|\hat f - f_{j^*}\|_\infty \le \delta$. For $\Delta = \hat f - f_{j^*}$ (so $\|\Delta\|_\infty \le \delta$), let us write

$$\hat f - f_0 = (\hat f - f_{j^*}) + f_{j^*} - f_0 = \Delta + f_{j^*} - f_0,$$

(b) In order to more easily compare the risks, let us note that the prediction risk may be written

$$R(\hat f, f_0) = E\left[\frac{1}{n}\sum_{i=1}^n (\hat f - f_0)^2(T_i)\right],$$

where $T_i$ are iid variables with law $\mathcal{L}(X_i) = \mathcal{L}(X_1)$ (recall the $X_i$ are iid). One may now write

$$R(\hat f, f_0) - \hat R(\hat f, f_0) = E\left[\frac{1}{n}\sum_{i=1}^n (f_{j^*} - f_0)^2(T_i) - (f_{j^*} - f_0)^2(X_i)\right] + \mathcal{R}_1,$$

where the remainder term $\mathcal{R}_1$ verifies $|\mathcal{R}_1| \le 2\delta^2 + 2\times 4\delta F \le 10\delta F$ for small $\delta$, which is obtained by expanding the squares and using Cauchy–Schwarz' inequality. Deduce

$$|R(\hat f, f_0) - \hat R(\hat f, f_0)| \le E\left|\frac{1}{n}\sum_{i=1}^n g_{j^*}(X_i, T_i)\right| + 10\delta F,$$

where we have set, for any integer $j \le \mathcal{N}_n$,

$$g_j(X_i, T_i) := (f_j - f_0)^2(T_i) - (f_j - f_0)^2(X_i).$$

(c) Let us set, for $j = 1, \dots, N$, and $a \vee b = \max(a, b)$,

$$r_j^2 = \frac{\log \mathcal{N}_n}{n} \vee E_T[(f_j - f_0)^2(T)],$$

where $E_T$ means that one takes the expectation with respect to $T$ (only). Let us further set

$$\mathcal{U}^2 := E_T[(\hat f - f)^2(T)], \qquad \mathcal{T} := \max_j \left|\sum_{i=1}^n \frac{g_j(X_i, T_i)}{r_j F}\right|.$$

Since $T$ is independent of $(X_i, Y_i)_i$, one has $E_T[(f_j - f_0)^2(T)] = E_T[(f_j - f_0)^2(T) \mid (X_i, Y_i)_i]$, which allows to define $r_{j^*}$ as $r_j$ with $j$ replaced by the (random) quantity $j^*$. Namely,

$$r_{j^*}^2 = \frac{\log \mathcal{N}_n}{n} \vee E_T[(f_{j^*} - f_0)^2(T)],$$

where the last display is random (via $j^*$). One may bound, using $(a+b)^2 \le 2a^2 + 2b^2$,

$$r_{j^*}^2 \le \frac{\log \mathcal{N}_n}{n} + 2E_T[(f_{j^*} - \hat{f})^2(T)] + 2\mathcal{U}^2 \le \frac{\log \mathcal{N}_n}{n} + 2\delta^2 + 2\mathcal{U}^2,$$

using that $f_{j^*}$ is uniformly at most $\delta$ away from $f_0$. Then

$$\left|\sum_{i=1}^n g_{j^*}(X_i, T_i)\right| = \left|\sum_{i=1}^n \frac{g_{j^*}(X_i, T_i)}{r_{j^*}F}\right| r_{j^*}F \le \max_j \left|\sum_{i=1}^n \frac{g_j(X_i, T_i)}{r_j F}\right| r_{j^*}F = \mathcal{T} r_{j^*}F.$$

Combining with the above bound on $r_{j^*}$ and since by definition of the prediction risk $E[\mathcal{U}^2] = R(\hat{f}, f_0)$,

$$\begin{aligned}\left|\sum_{i=1}^n g_{j^*}(X_i, T_i)\right| / F &\le E\left[\mathcal{T} \cdot \sqrt{\frac{\log \mathcal{N}_n}{n} + 2\delta^2 + 2\mathcal{U}^2}\right] \le \sqrt{2}E[\mathcal{T} \cdot \mathcal{U}] + \left\{\sqrt{\frac{\log \mathcal{N}_n}{n}} + \sqrt{2}\delta\right\} E[\mathcal{T}] \\ &\le \sqrt{2E[\mathcal{T}^2]}\sqrt{R(\hat{f}, f_0)} + \left\{\sqrt{\frac{\log \mathcal{N}_n}{n}} + \sqrt{2}\delta\right\} E[\mathcal{T}].\end{aligned}$$

(d) Let us now provide bounds for $E[\mathcal{T}], E[\mathcal{T}^2]$. We start by deriving a deviation bound $P[\mathcal{T} > t]$ for $t > 0$ to be chosen later. A union bound gives, setting $Z_{ij} := g_j(X_i, T_i)/(r_j F)$,

$$P[\mathcal{T} > t] \le \sum_{j=1}^{\mathcal{N}_n} P\left[\left|\sum_{i=1}^n Z_{ij}\right| \ge t\right].$$

The variables $Z_{ij}$ are centered and bounded in absolute value by $[(2F)^2 + (2F)^2]/(r_j F) \le 8F/r_j =: M_j$. Also, using the definition of $r_j$,

$$\mathrm{Var}[Z_{ij}] \le \frac{2}{r_j^2 F^2} E[(f_j - f_0)(X_1)^4] \le \frac{2(2F)^2}{r_j^2 F^2} E[(f_j - f_0)(X_1)^2] \le 8 =: v_{ij}.$$

An application of Bernstein's inequality to the independent variables $Z_{ij}$ gives

$$P[\mathcal{T} > t] \le \sum_{j=1}^{\mathcal{N}_n} 2\exp\left(-\frac{t^2}{2M_j t/3 + 2\sum_{i=1}^n v_{ij}}\right) \le 2\mathcal{N}_n \exp\left(-\frac{t^2}{\frac{16Ft}{3r_j} + 16n}\right).$$

Using that $r_j \ge \sqrt{\log \mathcal{N}_n / n}$ by definition and choosing $t \ge t_1 := C_F\sqrt{n \log \mathcal{N}_n}$ for some large enough $C_F = C(F)$ leads to

$$P[\mathcal{T} > t] \le 2\mathcal{N}_n \exp\left(-Ct\sqrt{\frac{\log \mathcal{N}_n}{n}}\right).$$

From this one easily sees that $\mathcal{T}$ is of order $\sqrt{n \log \mathcal{N}_n}$. More precisely, using the formulas $E\mathcal{T} = \int_0^\infty P[\mathcal{T} \ge t]dt$ and $E\mathcal{T}^2 = \int_0^\infty P[\mathcal{T}^2 \ge t]dt$, one obtains (check it as an exercise)

$$E\mathcal{T} \lesssim \sqrt{n \log \mathcal{N}_n}, \qquad E[\mathcal{T}^2] \lesssim n \log \mathcal{N}_n.$$

(e) Combining the points (b), (c), (d) above leads to

$$|\underbrace{R(\hat f,f_0)}_{a}-\underbrace{\hat R_n(\hat f,f_0)}_{b}|\le \underbrace{\frac{F}{n}\sqrt{2n\log\mathcal{N}_n}}_{2c}\underbrace{\sqrt{R(\hat f,f_0)}}_{\sqrt a}+\underbrace{\frac{F}{n}\left\{\sqrt{\frac{\log\mathcal{N}_n}{n}}+\sqrt{2}\delta\right\}\sqrt{n\log\mathcal{N}_n}+10\delta F}_{d}.$$

Inequality (4.16) is obtained upon noting the following: for reals $b,c,d$ and $a>0$ such that $|a-b|\le 2\sqrt{a}c+d$, for any $\varepsilon>0$ it holds

$$a\le(1+\varepsilon)(b+d)+\frac{(1+\varepsilon)^2}{\varepsilon}c^2,$$

obtained by using the inequality $\sqrt{a}c\le\frac{\varepsilon}{1+\varepsilon}a+\frac{1+\varepsilon}{\varepsilon}c^2$ (itself a variant of $ac\le(a^2+c^2)/2$).

*Proof of* (4.17). In the sequel we write $Y$ to mean the vector of observed $Y_i$'s, and in slight abuse of notation also interpret it as the function that takes values $Y_i$'s at $X_i$'s (so as to evaluate it under the empirical norm $\|\cdot\|_n$). For any $f\in\mathcal{F}$, using the definition of the ERM,

$$\begin{aligned}\|\hat f-f_0\|_n^2&=\|\hat f-Y\|_n^2+\|Y-f_0\|_n^2+2\langle Y-f_0,\hat f-Y\rangle_n\\&\le\|f-Y\|_n^2+\|Y-f_0\|_n^2+2\langle Y-f_0,\hat f-Y\rangle_n,\end{aligned}$$

where $\langle\cdot,\cdot\rangle_n$ is the inner–product associated to the empirical norm. Now using the definition of the model, $Y=f_0+\varepsilon$, so that

$$\begin{aligned}\|f-Y\|_n^2+\|Y-f_0\|_n^2+2\langle Y-f_0,\hat f-Y\rangle_n&=\|f-f_0\|_n^2-2\langle f-f_0,\varepsilon\rangle_n+\|\varepsilon\|_n^2+\|\varepsilon\|_n^2+2\langle\varepsilon,\hat f-f_0-\varepsilon\rangle_n\\&=\|f-f_0\|_n^2-2\langle f,\varepsilon\rangle_n+2\langle\varepsilon,\hat f\rangle_n.\end{aligned}$$

Combining the above inequalities, taking expectations and using $E\|f-f_0\|_n^2=E[(f-f_0)(X_1)^2]=R(f,f_0)$,

$$\hat R_n(\hat f,f_0)=E\|\hat f-f_0\|_n^2\le R(f,f_0)+2E\langle\varepsilon,\hat f\rangle_n,$$

where we have used that $\langle f,\varepsilon\rangle_n$ is centered, as $E[f(X_1)\varepsilon_1]=E[f(X_1)]E[\varepsilon_1]=0$. To derive (4.17), it suffices to bound $E\langle\varepsilon,\hat f\rangle_n$, that is the expectation of an empirical process. We will bound this classically by replacing the quantity $\hat f$ by a maximum over a finite set of functions given to us by the entropy covering. We then conclude using a bound on the maximum in expectation (something sometimes called a “maximal inequality").

Let $j^*$ be a random index such that $\|\hat f-f_j\|_\infty\le\delta$. Let us denote, for an integer $j\le\mathcal{N}_n$,

$$\xi_j=\frac{1}{\sqrt n}\sum_{i=1}^n\frac{\varepsilon_i(f_j-f_0)(X_i)}{\|f_j-f_0\|_n}.$$

This is a centered variable, whose distribution given the $X_i$'s is standard normal. By writing

$$E\left[\max_{1\le j\le\mathcal{N}_n}\xi_j^2\right]=E\left[E\left[\max_{1\le j\le\mathcal{N}_n}\xi_j^2\,|\,(X_i)_i\right]\right],$$

we see that to bound the first expectation it is enough to bound the conditional expectation on the right hand side, that is the expectation of the maximum of chi-squared(1) variables, so that using Lemma 4.14 below, the last display is at most $3\log\mathcal{N}_n+1$.

Recalling the definition of $j^*$ and using the triangle inequality,

$$
\begin{aligned}
\left|E\langle\varepsilon,\hat{f}\rangle_n\right| &= \left|\frac{1}{n}E\sum_{i=1}^n \varepsilon_i(\hat{f}(X_i)-f_0(X_i))\right| \\
&\le \delta E\frac{1}{n}\sum_{i=1}^n|\varepsilon_i| + \left|\frac{1}{n}E\sum_{i=1}^n \varepsilon_i(f_{j^*}(X_i)-f_0(X_i))\right| \\
&\le \delta + \frac{1}{\sqrt{n}}E\left[|\xi_{j^*}|\|f_{j^*}-f_0\|_n\right].
\end{aligned}
$$

One further bounds $\|f_{j^*}-f_0\|_n \le \|f_{j^*}-\hat{f}\|_n + \|\hat{f}-f_0\|_n \le \delta + \|\hat{f}-f_0\|_n$ and via Cauchy-Schwarz,

$$
\begin{aligned}
E\left[|\xi_{j^*}|\|f_{j^*}-f_0\|_n\right] &\le \sqrt{2E\|\hat{f}-f_0\|_n^2+2\delta^2}\sqrt{E\left[\max_{1\le j\le \mathcal{N}_n}\xi_j^2\right]} \\
&\le \sqrt{2}\left[\sqrt{\hat{R}(\hat{f},f_0)}+\delta\right]\sqrt{3\log\mathcal{N}_n+1},
\end{aligned}
$$

where we have used the property on maxima mentioned above. Deduce

$$
\left|E\langle\varepsilon,\hat{f}\rangle_n\right| \le \delta + \sqrt{2}\frac{\sqrt{4n}}{\sqrt{n}}\delta + \sqrt{2}\sqrt{\hat{R}(\hat{f},f_0)\cdot\frac{4\log\mathcal{N}_n}{n}} \le 5\delta + 4\sqrt{\hat{R}(\hat{f},f_0)\frac{\log\mathcal{N}_n}{n}},
$$

where we use the assumption $1\le\log\mathcal{N}_n\le n$. One obtains

$$
\hat{R}(\hat{f},f_0) \le R(f,f_0) + 4\sqrt{\hat{R}(\hat{f},f_0)\frac{\log\mathcal{N}_n}{n}} + 5\delta.
$$

To conclude, one uses a similar argument as at the end of the proof of (4.16), so that one moves both $\hat{R}(\hat{f},f_0)$ terms to the left hand side of the inequality, which concludes the proof of (4.17).

**Lemma 4.14.** *Let $\xi_1,\dots,\xi_N$ be standard normal variables (but not necessarily independent). Then, for all $N\ge1$,*

$$
E\left[\max_{1\le j\le N}\xi_j^2\right] \le 3\log N + 1.
$$

This is standard (see [SH20a], Lemma C.1, or e.g. [BLM13] Corollary 2.6 for a more general result for sub-exponential variables); the way to understand it: for Gaussian variables the maximum is of order at most $\sqrt{\log N}$ so the squares of $N$ Gaussians have their maximum at most of size $(\log N)$.

## 4.4 Compositional structures: towards solving the curse of dimensionality

*Discovering a hidden 'structure'.* The 'raw' regression data collected by the statistician takes the form, in the setting model (4.1), of $n$ vectors of size $d+1$: the $n$ pairs $(X_i^T, Y_i)$ with $X_i\in[0,1]^d$ and $Y_i$ a real, with the dimension $d$ possibly large (think for instance of e.g. $d=10$ or 20). The unknown regression function $f_0(x_1,\dots,x_d)$ depends on $d$ of variables, and we have seen that if $d$ is larger than a few units this may lead to a slow uniform convergence rate of the form $n^{-2\beta/(2\beta+d)}$ for the prediction risk. It is often the case though that the problem is effectively of smaller dimension than $d$. We give a number of frequently encountered examples