Optimal approximation of continuous functions by very deep ReLU networks

Dmitry Yarotsky*†
d.yarotsky@skoltech.ru

Abstract

We prove that deep ReLU neural networks with conventional fully-connected architectures with W weights can approximate continuous ν-variate functions f with uniform error not exceeding $a_\nu \omega_f(c_\nu W^{-2/\nu})$, where ω_f is the modulus of continuity of f and a_ν, c_ν are some ν-dependent constants. This bound is tight. Our construction is inherently deep and nonlinear: the obtained approximation rate cannot be achieved by networks with fewer than $\Omega(W/\ln W)$ layers or by networks with weights continuously depending on f.

1 Introduction

Expressiveness of deep neural networks with piecewise-linear (in particular, ReLU) activation functions has been a topic of much theoretical research in recent years. The topic has many aspects, with connections to combinatorics Montufar et al. [2014], Telgarsky [2016], topology Bianchini and Scarselli [2014], Vapnik-Chervonenkis dimension Bartlett et al. [1998], Sakurai [1999] and fat-shattering dimension Kearns and Schapire [1990], Anthony and Bartlett [2009], hierarchical decompositions of functions Mhaskar et al. [2016], information theory Petersen and Voigtlaender [2017], etc.

Here we adopt the perspective of classical approximation theory, in which the problem of expressiveness can be basically described as follows. Suppose that f is a multivariate function, say on the cube $[0,1]^\nu$, and has some prescribed regularity properties; how efficiently can one approximate f by deep neural networks? The question has been studied in several recent publications. Depth-separation results for some explicit families of functions have been obtained in Safran and Shamir [2016], Telgarsky [2016]. General upper and lower bounds on approximation rates for functions characterized by their degree of smoothness have been obtained in Liang and Srikant [2016], Yarotsky [2017]. Hanin [2017], Lu et al. [2017]

*Skolkovo Institute of Science and Technology, Skolkovo Innovation Center, Building 3, Moscow 143026 Russia

†Institute for Information Transmission Problems, Bolshoy Karetny per. 19, build.1, Moscow 127051, Russia

establish the universal approximation property and convergence rates for deep and "narrow" (fixed-width) networks. Petersen and Voigtlaender [2017] establish convergence rates for approximations of discontinuous functions. Generalization capabilities of deep ReLU networks trained on finite noisy samples are studied in Schmidt-Hieber [2017].

In the present paper we consider and largely resolve the following question: what is the optimal rate of approximation of general continuous functions by deep ReLU networks, in terms of the number W of network weights and the modulus of continuity of the function? Specifically, for any W we seek a network architecture with W weights so that for any continuous $f : [0,1]^\nu \to \mathbb{R}$, as W increases, we would achieve the best convergence in the uniform norm $\|\cdot\|_\infty$ when using these architectures to approximate f.

In the slightly different but closely related context of approximation on balls in Sobolev spaces $\mathcal{W}^{d,\infty}([0,1]^\nu)$, this question of optimal convergence rate has been studied in Yarotsky [2017]. That paper described ReLU network architectures with W weights ensuring approximation with error $O(W^{-d/\nu} \ln^{d/\nu} W)$ (Theorem 1). The construction was linear in the sense that the network weights depended on the approximated function linearly. Up to the logarithmic factor, this approximation rate matches the optimal rate over all parametric models under assumption of continuous parameter selection (DeVore et al. [1989]). It was also shown in Theorem 2 of Yarotsky [2017] that one can slightly (by a logarithmic factor) improve over this conditionally optimal rate by adjusting network architectures to the approximated function.

On the other hand, it was proved in Theorem 4 of Yarotsky [2017] that ReLU networks generally cannot provide approximation with accuracy better than $O(W^{-2d/\nu})$ – a bound with the power $\frac{2d}{\nu}$ twice as big as in the previously mentioned existence result. As was shown in the same theorem, this bound can be strengthened for shallow networks. However, without imposing depth constraints, there was a serious gap between the powers $\frac{2d}{\nu}$ and $\frac{d}{\nu}$ in the lower and upper accuracy bounds that was left open in that paper.

In the present paper we bridge this gap in the setting of continuous functions (which is slightly more general than the setting of the Sobolev space of Lipschitz functions, $\mathcal{W}^{1,\infty}([0,1]^\nu)$). Our key insight is the close connection between approximation theory and VC dimension bounds. The lower bound on the approximation accuracy in Theorem 4 of Yarotsky [2017] was derived using the upper VCdim bound $O(W^2)$ from Goldberg and Jerrum [1995]. More accurate upper and lower bounds involving the network depth L have been given in Bartlett et al. [1998], Sakurai [1999]. The recent paper Bartlett et al. [2017] establishes nearly tight lower and upper VCdim bounds: $cWL\ln(W/L) \le \mathrm{VCdim}(W,L) \le CWL\ln W$, where $\mathrm{VCdim}(W,L)$ is the largest VC dimension of a piecewise linear network with W weights and L layers. The key element in the proof of the lower bound is the "bit extraction technique" (Bartlett et al. [1998]) providing a way to compress significant expressiveness in a single network weight. In the present paper we adapt this technique to the approximation theory setting.

Our main result states that deep ReLU networks having conventional fully-connected architectures with W weights can approximate continuous ν-variate functions f with uniform error not exceeding $a_\nu \omega_f(c_\nu W^{-2/\nu})$, where ω_f is the modulus of continuity of f and a_ν, c_ν are

some ν-dependent constants. A particular class of architectures ensuring this approximation rate is the already mentioned fixed-width architectures with sufficiently large width H; for example, $H = 2\nu + 10$ will suffice. Our approximation is nonlinear and deep and is obtained by combining the two-scales expansion from Theorem 2 in Yarotsky [2017] with the bit-extraction technique. The nonlinearity and depth are not artifacts of our construction, but necessary features of networks providing the rate $a_\nu \omega_f(c_\nu W^{-2/\nu})$. Specifically, this rate has the following remarkable properties:

Optimality: This rate is optimal in the sense that the power $\frac{2}{\nu}$ cannot be improved (by adjusting the weights or by changing the network architecture).

Inherent discontinuity: This rate can only be achieved if the network weights depend on the approximated function discontinuously. If the weights depend on f continuously, the power in the rate cannot be higher than $\frac{1}{\nu}$.

Inherent depth: This rate can only be achieved with very deep networks – namely, only if the network depth L scales with W as $L = \Omega(W/\ln W)$.

We formulate precisely the main result and the above properties in Section 2, discuss the result in Section 3, and give the proof of the main theorem in Section 4.

2 The result

We define the modulus of continuity ω_f of a function $f : [0,1]^\nu \to \mathbb{R}$ by

$$\omega_f(r) = \max\{|f(\mathbf{x}) - f(\mathbf{y})| : \mathbf{x}, \mathbf{y} \in [0,1]^\nu, |\mathbf{x} - \mathbf{y}| \leq r\}, \tag{1}$$

where $|\mathbf{x}|$ is the euclidean norm of $\mathbf{x}$.

We approximate functions $f : [0,1]^\nu \to \mathbb{R}$ by usual feed-forward neural networks with the ReLU activation function $x \mapsto x_+ \equiv \max(0, x)$. The network has ν input units, some hidden units, and one output unit. The hidden units are assumed to be grouped in a sequence of layers so that the inputs of each unit is formed by outputs of some units from previous layers. The depth L of the network is the number of these hidden layers. A hidden unit computes a linear combination of its inputs followed by the activation function: $x_1, \ldots, x_s \mapsto (\sum_{k=1}^s w_k x_k + h)_+$, where w_k and h are the weights associated with this unit. The output unit acts similarly, but without the activation function: $x_1, \ldots, x_s \mapsto \sum_{k=1}^s w_k x_k + h$.

It will be convenient to consider one specific class of network architectures that we will, for brevity, refer to as *standard*, see Fig. 1. These networks have a constant number of neurons in each hidden layer; we refer to this number as the "width" H of the network. Neighboring layers in a standard network are fully-connected. The standard network of width H and depth L has $W = L(H^2 + H) + H^2 + (\nu + 1)H + 1$ weights. We will be interested in the scenario where H is fixed and the network grows by increasing L; then W grows linearly with L. Below we will refer to the "standard architecture of width H having

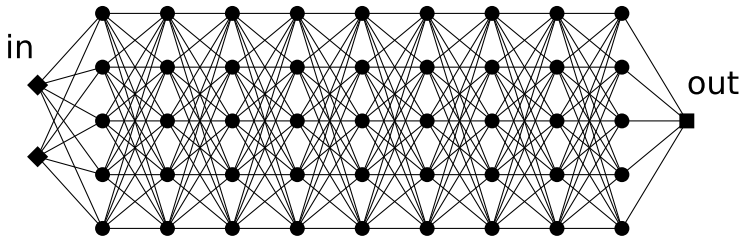

Figure 1: The standard network architecture that has $\nu = 2$ inputs, depth $L = 9$ and width $H = 5$ (i.e., contains 9 fully-connected hidden layers, 5 neurons each).

W weights": the depth L is supposed in this case to be determined from the above equality; we will assume without loss of generality that the equality is satisfied with an integer L.

Our main result is the following theorem.

Theorem 1.

a) For any ν and W, there exist a network architecture with ν inputs and W weights such that any continuous function $f : [0,1]^\nu \to \mathbb{R}$ can be approximated by a ReLU network with this architecture with uniform error not larger than $a_\nu \omega_f(c_\nu W^{-2/\nu})$, where a_ν, c_ν are some positive constants depending only on ν.

b) An example of such architecture is the standard architecture of width $2\nu + 10$.

The full proof of this theorem is given in Section 4; we explain now its main idea. We start with approximating the given function f using a standard piecewise linear interpolation on the grid $(\mathbb{Z}/N)^\nu$ with a large N. This approximation $\widetilde{f}_1$ can be implemented by a ReLU network of size $O(N^\nu)$ and has error $O(\omega_f(O(\frac{1}{N})))$, i.e. $O(\omega_f(O(W^{-1/\nu})))$. We improve this approximation by constructing an additional network that approximately encodes the discrepancy $f - \widetilde{f}_1$. In contrast to $\widetilde{f}_1$, the second approximation is inherently discrete: we consider a finite set of possible shapes of $f - \widetilde{f}_1$ in patches of linear size $\sim \frac{1}{N}$, and in each patch we use a single network weight to encode the closest shape. For a set of M shapes, reconstruction of the shape from the respective weight requires a deep subnetwork with $O(\ln M)$ connections and layers. Choosing $M \sim c^{N^\nu}$, we see that the second approximation allows us to effectively improve the overall approximation scale from $\sim \frac{1}{N}$ down to $\sim \frac{1}{N^2}$ while keeping the total number of weights in the network at $O(N^\nu)$. This gives us the desired error bound $O(\omega_f(O(W^{-2/\nu})))$.

In the second theorem we collect the remarkable properties of the established approximation rate.

Theorem 2.

a) (Optimality) Theorem 1a) does not hold if the power $\frac{2}{\nu}$ is replaced by any number $p > \frac{2}{\nu}$.

b) (Inherent discontinuity) Suppose that, for each network architecture, the network weights depend continuously on the approximated function (with respect to the usual topology of $C([0,1]^\nu)$). Suppose that Theorem 1a) holds with this additional assumption and with the bound $a_\nu \omega_f(c_\nu W^{-2/\nu})$ replaced by $a_\nu \omega_f(c_\nu W^{-p})$ with some p. Then $p \leq \frac{1}{\nu}$.

c) (Inherent depth) The architectures satisfying Theorem 1a) must have depths $L \geq d_\nu W/\ln W$ with some ν-dependent constant $d_\nu > 0$.

Proof. These statements are either direct consequences or slight modifications of existing results. The proofs of these statements have the common element of considering the approximation for functions from the unit ball $F_{\nu,1}$ in the Sobolev space $\mathcal{W}^{1,\infty}([0,1]^\nu)$ of Lipschitz functions. Namely, fix the dimension ν. Suppose that for any W one can choose network architectures η_W with W weights so that any continuous function $f : [0,1]^\nu \to \mathbb{R}$ can be approximated using these architectures with errors not larger than $a\omega_f(cW^{-p})$, where a, c, p are some constants (depending on ν). Then all $f \in F_{\nu,1}$ can be approximated by η_W with accuracy $\epsilon_W = c_1 W^{-p}$ with some constant c_1. The three statements of the theorem are then obtained as follows.

a) This statement is a consequence of Theorem 4a) of Yarotsky [2017], which is in turn a consequence of the upper bound $O(W^2)$ for the VC dimension of a ReLU network (Goldberg and Jerrum [1995]). Precisely, Theorem 4a) states that $W \geq c_2 \epsilon_W^{-\nu/2}$ with some c_2, which implies that $p \leq \frac{2}{\nu}$.

b) This statement is a consequence of the general bound of DeVore et al. [1989] on the efficiency of approximation of Sobolev balls with parametric models having parameters continuously depending on the approximated function. Namely, if the weights of the networks η_W depend on $f \in F_{\nu,1}$ continuously, then Theorem 4.2 of DeVore et al. [1989] implies that $\epsilon_W \geq c_2 W^{-1/\nu}$ with some constant c_2, which implies that $p \leq \frac{1}{\nu}$.

c) This statement can be obtained by combining arguments of Theorem 4 of Yarotsky [2017] with the recently established tight upper bound for the VC dimension of ReLU networks (Bartlett et al. [2017], Theorem 6) with given depth L and the number of weights W:

$$\mathrm{VCdim}(W, L) \leq CWL \ln W, \tag{2}$$

where C is a global constant.

Specifically, let $p = \frac{2}{\nu}$ so that $\epsilon_W = c_1 W^{-2/\nu}$. By considering suitable trial functions, one shows that if we threshold the network output, the resulting networks must have VC dimension $\mathrm{VCdim}_W \geq c_2 \epsilon_W^{-\nu}$ (see Eq.(38) in Yarotsky [2017]), and hence $\mathrm{VCdim}_W \geq c_3 W^2$. Now using the upper bound (2), we conclude that $c_3 W^2 \leq CWL \ln W$, i.e. $L \geq dW/\ln W$ with some constant d. □

We remark that the construction used in Theorem 1 can be easily generalized to produce networks of depth $O(W^{p\nu-1})$ ensuring accuracy $O(\omega_f(O(W^{-p}))$ for any power $p \in (\frac{1}{\nu}, \frac{2}{\nu})$. To this end, the second approximation used in the construction must be performed on the scale $N^{-\nu p}$ (instead of N^{-2} appearing in the proof of Theorem 1). Arguing as in Theorem 2c), we

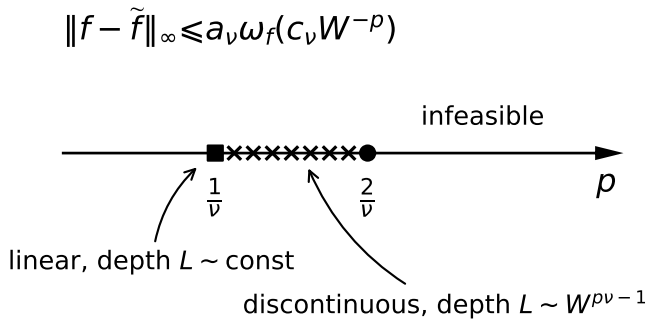

Figure 2: The "phase diagram" of convergence rates. At $p = \frac{1}{\nu}$ the rate is achieved by shallow networks with weights linearly (and continuously) depending on f. At $p \in (\frac{1}{\nu}, \frac{2}{\nu}]$, the rate is achieved by deep networks with weights discontinuously depending on f. Rates with $p > \frac{2}{\nu}$ are infeasible.

also see that, up to a logarithmic factor, the depth $\Omega(W^{p\nu-1})$ is not only sufficient but also necessary for the accuracy $O(\omega_f(O(W^{-p}))$. We summarize these observations in the "phase diagram" shown in Fig. 2.

3 Discussion

We discuss now our result in the context of general approximation theory and practical machine learning. First, a theorem of Kainen et al. [1999] shows that in the optimal approximations by neural networks the weights generally discontinuously depend on the approximated function, so the discontinuity property that we have established is not surprizing. However, this theorem of Kainen et al. [1999] does not in any way quantify the accuracy gain that can be acquired by giving up the continuity of the weights. Our result does this in the practically important case of deep ReLU networks, and explicitly describes a relevant mechanism.

In general, many nonlinear approximation schemes involve some form of discontinuity, often explicit (e.g., using different expansion bases for different approximated functions (DeVore [1998]). At the same time, discontinuous selection of parameters in parametric models is often perceived as an undesirable phenomenon associated with unreliable approximation (DeVore et al. [1989], DeVore [1998]). We point out, however, that the deep narrow architecture considered in the present paper has much in common with popular state-of-the-art practical networks for highly accurate image recognition – residual networks He et al. [2016] and highway networks Srivastava et al. [2015] that may have dozens or even hundreds of layers. While our model does not explicitly include shortcut connections as in ResNets, a very similar element is effectively present in the proof of Theorem 1 (in the form of channels reserved for passing forward the data). We can say, therefore, that our result should be at least to some extent relevant for ResNet-like networks.

Quantized network weights have been previously considered from the information-

theoretic point of view in Bölcskei et al. [2017], Petersen and Voigtlaender [2017]. In the present paper we do not use quantized weights in the statement of the approximation problem, but they appear in the solution (namely, we use them to store small-scale descriptions of the approximated function). One can expect that weight quantization may play an important role in the future development of the theory of deep networks.

4 Proof of Theorem 1

4.1 Preliminaries. The modulus of continuity defined by (1) is monotone nondecreasing in r. By the convexity of the cube $[0,1]^\nu$, for any integer n we have $\omega_f(r) \le n\omega_f(\frac{r}{n})$. More generally, for any $\alpha \in (0,1)$ we can write

$$\alpha\omega_f(r) \le 1/\lfloor 1/\alpha \rfloor \omega_f(r) \le \omega_f(r/\lfloor 1/\alpha \rfloor) \le \omega_f(2\alpha r). \tag{3}$$

The ReLU function $x \mapsto x_+$ allows us to implement the binary max operation as $\max(a,b) = a + (b-a)_+$ and the binary min operation as $\min(a,b) = a - (a-b)_+$. The maximum or minimum of any n numbers can then be implemented by chaining $n-1$ binary max's or min's. Computation of the absolute value can be implemented by $|x| = 2x_+ - x$.

Without loss of generality, we can assume that hidden units in the ReLU network may not include the ReLU nonlinearity (i.e., may compute just a linear combination of the input values). Indeed, we can simulate nonlinearity-free units just by increasing the weight h in the formula $x_1, \dots, x_s \mapsto (\sum_{k=1}^s w_k x_k + h)_+$, so that $\sum_{k=1}^s w_k x_k + h$ is always nonnegative (this is possible since the network inputs are from a compact set and the network implements a continuous function), and then compensating the shift by subtracting appropriate amounts in the units receiving input from the unit in question. In particular, we can simulate in this way trivial "pass forward" (identity) units that simply deliver given values to some later layers of the network.

In the context of standard networks of width H, it will be occasionally convenient to think of the network as consisting of H "channels", i.e. sequences of units with one unit per layer. Channels can be used to pass forward values or to perform computations. For example, suppose that we already have n channels that pass forward n numbers. If we also need to compute the maximum of these numbers, then, by chaining binary max's, this can be done in a subnetwork including one additional channel that spans n layers.

We denote vectors by boldface characters; the scalar components of a vector $\mathbf{x}$ are denoted $x_1, x_2, \dots$

4.2 Initial coarse approximation. We start the proof by constructing an initial approximation $\widetilde{f}_1$ to the function f on a scale $\frac{1}{N}$, where N is some fixed large integer. The approximation $\widetilde{f}_1$ is constructed as an interpolation of f on the grid $(\mathbb{Z}/N)^\nu$. We consider the standard triangulation P_N of the space $\mathbb{R}^\nu$ that partitions it into the simplexes

$$\Delta^{(N)}_{\mathbf{n},\rho} = \{\mathbf{x} \in \mathbb{R}^\nu : 0 \le x_{\rho(1)} - \tfrac{n_{\rho(1)}}{N} \le \dots \le x_{\rho(\nu)} - \tfrac{n_{\rho(\nu)}}{N} \le \tfrac{1}{N}\},$$

where $\mathbf{n} = (n_1, \ldots, n_\nu) \in \mathbb{Z}^\nu$ and ρ is a permutation of ν elements. This triangulation can be described as resulting by dissecting the space $\mathbb{R}^\nu$ by the hyperplanes $x_k - x_s = \frac{n}{N}$ and $x_k = \frac{n}{N}$ with various $1 \le k, s \le \nu$ and $n \in \mathbb{Z}$.

The vertices of the simplexes of the triangulation P_N are the points of the grid $(\mathbb{Z}/N)^\nu$. Each such point $\frac{\mathbf{n}}{N}$ is a vertex in $(\nu+1)!$ simplexes. The union of these $(\nu+1)!$ simplexes is the convex set $\{\mathbf{x} \in \mathbb{R}^\nu : (\max(\mathbf{x} - \frac{\mathbf{n}}{N}))_+ - (\min(\mathbf{x} - \frac{\mathbf{n}}{N}))_- \le \frac{1}{N}\}$, where $a_- = \min(a, 0)$.

We define the "spike function" $\phi : \mathbb{R}^\nu \to \mathbb{R}$ as the continuous piecewise linear function such that 1) it is linear on each simplex of the partition $P_{N=1}$, and 2) $\phi(\mathbf{n}) = \mathbb{1}[\mathbf{n} = \mathbf{0}]$ for $\mathbf{n} \in \mathbb{Z}^\nu$. The spike function can be expressed in terms of linear and ReLU operations as follows. Let R be the set of the $(\nu+1)!$ simplexes having $\mathbf{0}$ as a vertex. For each $\Delta \in R$ let $l_\Delta : \mathbb{R}^\nu \to \mathbb{R}$ be the affine map such that $l_\Delta(\mathbf{0}) = 1$ and l_Δ vanishes on the face of Δ opposite to the vertex $\mathbf{0}$. Then

$$\phi(\mathbf{x}) = \big(\min_{\Delta \in R}(l_\Delta(\mathbf{x}))\big)_+. \tag{4}$$

Indeed, if $\mathbf{x} \notin \cup_{\Delta \in R}\Delta$, then it is easy to see that we can find some $\Delta \in R$ such that $l_\Delta(\mathbf{x}) < 0$, hence ϕ vanishes outside $\cup_{\Delta \in R}\Delta$, as required. On the other hand, consider the restriction of ϕ to $\cup_{\Delta \in R}\Delta$. Each l_Δ is nonnegative on this set. For each Δ_1 and $\mathbf{x} \in \cup_{\Delta \in R}\Delta$, the value $l_{\Delta_1}(\mathbf{x})$ is a convex combination of the values of l_{Δ_1} at the vertices of the simplex $\Delta_\mathbf{x}$ that $\mathbf{x}$ belongs to. Since l_Δ is nonnegative and $l_\Delta(\mathbf{0}) = 1$ for all Δ, the minimum in (4) is attained at Δ such that l_Δ vanishes at all vertices of $\Delta_\mathbf{x}$ other than $\mathbf{0}$, i.e. at $\Delta = \Delta_\mathbf{x}$.

Note that each map l_Δ in (4) is either of the form $1 + x_k - x_s$ or $1 \pm x_k$ (different simplexes may share the same map l_Δ), so the minimum actually needs to be taken only over $\nu(\nu+1)$ different $l_\Delta(\mathbf{x})$: $\phi(\mathbf{x}) = \big(\min(\min_{k \ne s}(1 + x_k - x_s), \min_k(1 + x_k), \min_k(1 - x_k))\big)_+$.

We define now the piecewise linear interpolation $\widetilde{f}_1$ by

$$\widetilde{f}_1(\mathbf{x}) = \sum_{\mathbf{n} \in \{0,1,\ldots,N\}^\nu} f(\tfrac{\mathbf{n}}{N})\phi(N\mathbf{x} - \mathbf{n}). \tag{5}$$

The function $\widetilde{f}_1$ is linear on each simplex $\Delta^{(N)}_{\mathbf{n},\rho}$ and agrees with f at the interpolation knots $\mathbf{x} \in (\mathbb{Z}/N)^\nu \cap [0,1]^\nu$. We can bound $\||\nabla \widetilde{f}_1|\|_\infty \le \sqrt{\nu} N \omega_f(\frac{1}{N})$, since any partial derivative $\partial_k \widetilde{f}_1$ in the interior of a simplex equals $N(f(\frac{\mathbf{n}_1}{N}) - f(\frac{\mathbf{n}_2}{N}))$, where $\frac{\mathbf{n}_1}{N}, \frac{\mathbf{n}_2}{N}$ are the two vertices of the simplex having all coordinates the same except x_k. It follows that the modulus of continuity for $\widetilde{f}_1$ can be bounded by $\omega_{\widetilde{f}_1}(r) \le \sqrt{\nu} N \omega_f(\frac{1}{N}) r$. Moreover, by Eq.(3) we have $\frac{Nr}{2}\omega_f(\frac{1}{N}) \le \omega_f(r)$ as long as $r < \frac{2}{N}$. Therefore we can also write $\omega_{\widetilde{f}_1}(r) \le 2\sqrt{\nu}\omega_f(r)$ for $r \in [0, \frac{2}{N}]$.

In the remainder of the proof we will improve the initial approximation by approximating the discrepancy $f_2 = f - \widetilde{f}_1$. We can bound the modulus of continuity of f_2 by $\omega_{f_2}(r) \le \omega_f(r) + \omega_{\widetilde{f}_1}(r) \le (2\sqrt{\nu} + 1)\omega_f(r)$ for $r \in [0, \frac{2}{N}]$. Since any point $\mathbf{x} \in [0,1]^\nu$ is within the distance $\frac{\sqrt{\nu}}{2N}$ of one of the interpolation knots $\frac{\mathbf{n}}{N}$ where f_2 vanishes, we can also write

$$\|f_2\|_\infty \le \omega_{f_2}(\tfrac{\sqrt{\nu}}{2N}) \le \sqrt{\nu}\omega_{f_2}(\tfrac{1}{N}) \le \sqrt{\nu}(2\sqrt{\nu} + 1)\omega_f(\tfrac{1}{N}) \le 3\nu\omega_f(\tfrac{1}{N}), \tag{6}$$

where in the second inequality we again used Eq.(3).

4.3 Decomposition of the discrepancy. It is convenient to represent f_2 as a finite sum of functions with supports consisting of disjoint "patches" of linear size $\sim \frac{1}{N}$. Precisely, let $S = \{0,1,2\}^\nu$ and consider the partition of unity on the cube $[0,1]^\nu$

$$1 = \sum_{\mathbf{q}\in S} g_{\mathbf{q}},$$

where

$$g_{\mathbf{q}}(\mathbf{x}) = \sum_{\mathbf{n}\in(\mathbf{q}+(3\mathbb{Z})^\nu)\cap[0,N]^\nu} \phi(N\mathbf{x}-\mathbf{n}). \tag{7}$$

We then write

$$f_2 = \sum_{\mathbf{q}\in S} f_{2,\mathbf{q}},$$

where

$$f_{2,\mathbf{q}} = f_2 g_{\mathbf{q}}.$$

Each function $f_{2,\mathbf{q}}$ is supported on the disjoint union of cubes $Q_{\mathbf{n}} = \bigtimes_{s=1}^{\nu}[\frac{n_s-1}{N}, \frac{n_s+1}{N}]$ with $\mathbf{n} \in (\mathbf{q}+(3\mathbb{Z})^\nu)\cap[0,N]^\nu$ corresponding to the spikes in the expansion (7). With some abuse of terminology, we will refer to both these cubes and the respective restrictions $f_{2,\mathbf{q}}|_{Q_{\mathbf{n}}}$ as "patches".

Since $\|g_{\mathbf{q}}\|_\infty \le 1$, we have $\|f_{2,\mathbf{q}}\|_\infty \le \|f_2\|_\infty \le 3\nu\omega_f(\frac{1}{N})$. Also, if $r \in [0, \frac{2}{N}]$ then $\omega_{f_{2,\mathbf{q}}}(r) \le \|f_2\|_\infty\omega_{g_{\mathbf{q}}}(r) + \|g_{\mathbf{q}}\|_\infty\omega_f(r) \le 3\nu\omega_f(\frac{1}{N})\sqrt{\nu}Nr + \omega_f(r) \le (6\nu^{3/2}+1)\omega_f(r)$. In particular,

$$\omega_{f_{2,\mathbf{q}}}(\tfrac{1}{N^2}) \le \lambda, \text{ where } \lambda = (6\nu^{3/2}+1)\omega_f(\tfrac{1}{N^2}). \tag{8}$$

4.4 Second approximation. We will construct the full approximation $\widetilde{f}$ in the form

$$\widetilde{f} = \widetilde{f}_1 + \widetilde{f}_2 = \widetilde{f}_1 + \sum_{\mathbf{q}\in S} \widetilde{f}_{2,\mathbf{q}}, \tag{9}$$

where $\widetilde{f}_{2,\mathbf{q}}$ are approximations to $f_{2,\mathbf{q}}$. We define $\widetilde{f}_{2,\mathbf{q}}$ to be piecewise linear with respect to the refined triangulation P_{N^2} and to be given on the refined grid $(\mathbb{Z}/N^2)^\nu$ by

$$\widetilde{f}_{2,\mathbf{q}}(\tfrac{\mathbf{m}}{N^2}) = \lambda\lfloor f_{2,\mathbf{q}}(\tfrac{\mathbf{m}}{N^2})/\lambda\rfloor, \quad \mathbf{m} \in [0,\dots,N^2]^\nu, \tag{10}$$

where λ is defined in Eq.(8).

Let us estimate $\|f_{2,\mathbf{q}} - \widetilde{f}_{2,\mathbf{q}}\|_\infty$. Consider the piecewise linear function $\widehat{f}_{2,\mathbf{q}}$ defined similarly to $\widetilde{f}_{2,\mathbf{q}}$, but exactly interpolating $f_{2,\mathbf{q}}$ at the knots $(\mathbb{Z}/N^2)^\nu$. Then $\|\widehat{f}_{2,\mathbf{q}} - \widetilde{f}_{2,\mathbf{q}}\|_\infty \le \lambda$ since in each simplex the difference $\widehat{f}_{2,\mathbf{q}}(\mathbf{x}) - \widetilde{f}_{2,\mathbf{q}}(\mathbf{x})$ is a convex combination of the respective values at the vertices. Also, $\|\widehat{f}_{2,\mathbf{q}} - f_{2,\mathbf{q}}\|_\infty \le 3\nu\omega_f(\frac{1}{N^2})$ by applying the interpolation error bound (6) with N^2 instead of N. It follows that

$$\|f_{2,\mathbf{q}} - \widetilde{f}_{2,\mathbf{q}}\|_\infty \le \|f_{2,\mathbf{q}} - \widehat{f}_{2,\mathbf{q}}\|_\infty + \|\widehat{f}_{2,\mathbf{q}} - \widetilde{f}_{2,\mathbf{q}}\|_\infty \le (6\nu^{3/2}+3\nu+1)\omega_f(\tfrac{1}{N^2}). \tag{11}$$

Summing the errors over all $\mathbf{q}$, we can bound the error of the full approximation:

$$\|f - \widetilde{f}\|_\infty \le 3^\nu(6\nu^{3/2}+3\nu+1)\omega_f(\tfrac{1}{N^2}). \tag{12}$$

4.5 Patch encoding. Fix $\mathbf{q} \in S$. Note that, like $f_{2,\mathbf{q}}$, the approximation $\widetilde{f}_{2,\mathbf{q}}$ vanishes outside of the cubes $Q_{\mathbf{n}}$ with $\mathbf{n} \in (\mathbf{q}+(3\mathbb{Z})^\nu) \cap [0,N]^\nu$. Fix one of these $\mathbf{n}$ and consider those values $\lfloor f_{2,\mathbf{q}}(\frac{\mathbf{m}}{N^2})/\lambda \rfloor$ from Eq.(10) that lie in this patch:

$$A_{\mathbf{q},\mathbf{n}}(\mathbf{m}) = \lfloor f_{2,\mathbf{q}}(\tfrac{\mathbf{n}}{N} + \tfrac{\mathbf{m}}{N^2})/\lambda \rfloor, \quad \mathbf{m} \in [-N,\dots,N]^\nu.$$

By the bound (8), if $\mathbf{m}_1, \mathbf{m}_2$ are neighboring points on the grid $\mathbb{Z}^\nu$, then $A_{\mathbf{q},\mathbf{n}}(\mathbf{m}_1) - A_{\mathbf{q},\mathbf{n}}(\mathbf{m}_2) \in \{-1,0,1\}$. Moreover, since $f_{2,\mathbf{q}}$ vanishes on the boundary of $Q_{\mathbf{n}}$, we have $A_{\mathbf{q},\mathbf{n}}(\mathbf{m}) = 0$ if one of the components of $\mathbf{m}$ equals $-N$ or N. Let us consider separately the first component in the multi-index $\mathbf{m}$ and write $\mathbf{m} = (m_1, \overline{\mathbf{m}})$. Denote

$$B_{\mathbf{q},\mathbf{n}}(\mathbf{m}) = A_{\mathbf{q},\mathbf{n}}(m_1, \overline{\mathbf{m}}) - A_{\mathbf{q},\mathbf{n}}(m_1+1, \overline{\mathbf{m}}), \quad \mathbf{m} \in [-N+1,\dots,N-1]^\nu.$$

Since $B_{\mathbf{q},\mathbf{n}}(\mathbf{m}) \in \{-1,0,1\}$, we can encode all the $(2N-1)^\nu$ values $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ by a single ternary number

$$b_{\mathbf{q},\mathbf{n}} = \sum_{t=1}^{(2N-1)^\nu} 3^{-t}(B_{\mathbf{q},\mathbf{n}}(\mathbf{m}_t)+1),$$

where $t \mapsto \mathbf{m}_t$ is some enumeration of the multi-indices $\mathbf{m}$.

4.6 Reconstruction of the values $\{B_{\mathbf{q},\mathbf{n}}(\mathbf{m})\}$. The values $\{B_{\mathbf{q},\mathbf{n}}(\mathbf{m})\}$ are, up to the added constant 1, just the digits in the ternary representation of $b_{\mathbf{q},\mathbf{n}}$ and can be recovered from $b_{\mathbf{q},\mathbf{n}}$ by a deep ReLU network that iteratively implements "ternary shifts". Specifically, consider the sequence z_t with $z_0 = b_{\mathbf{q},\mathbf{n}}$ and $z_{t+1} = 3z_t - \lfloor 3z_t \rfloor$. Then $B_{\mathbf{q},\mathbf{n}}(\mathbf{m}_t) = \lfloor 3z_{t-1} \rfloor - 1$ for all t. To implement these computations by a ReLU network, we need to show how to compute $\lfloor 3z_t \rfloor$ for all z_t. Consider a piecewise-linear function $\chi_\epsilon : [0,3) \to \mathbb{R}$ such that

$$\chi_\epsilon(x) = \begin{cases} 0, & x \in [0, 1-\epsilon], \\ 1, & x \in [1, 2-\epsilon], \\ 2, & x \in [2, 3-\epsilon]. \end{cases} \tag{13}$$

Such a function can be implemented by $\chi_\epsilon(x) = \frac{1}{\epsilon}(x-(1-\epsilon))_+ - \frac{1}{\epsilon}(x-1)_+ + \frac{1}{\epsilon}(x-(2-\epsilon))_+ - \frac{1}{\epsilon}(x-2)_+$. Observe that if $\epsilon < 3^{-(2N-1)^\nu}$, then for all t the number $3z_t$ belongs to one of the three intervals in the r.h.s. of Eq.(13) and hence $\chi_\epsilon(3z_t) = \lfloor 3z_t \rfloor$. Thus, we can reconstruct the values $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ for all $\mathbf{m}$ by a ReLU network with $O(N^\nu)$ layers and weights.

4.7 Computation of $\widetilde{f}_{2,\mathbf{q}}$ in a patch. On the patch $Q_{\mathbf{n}}$, the function $\widetilde{f}_{2,\mathbf{q}}$ can be expanded over the spike functions as

$$\widetilde{f}_{2,\mathbf{q}}(\mathbf{x}) = \lambda \sum_{\overline{\mathbf{m}} \in [-N+1,\dots,N-1]^{\nu-1}} \sum_{m_1=-N+1}^{N-1} \phi(N^2(\mathbf{x} - (\tfrac{\mathbf{n}}{N} + \tfrac{(m_1,\overline{\mathbf{m}})}{N^2}))A_{\mathbf{q},\mathbf{n}}(m_1,\overline{\mathbf{m}}).$$

It is convenient to rewrite this computation in terms of the numbers $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ and the expressions

$$\Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1,\mathbf{x}) = \sum_{s=-N+1}^{m_1} \phi(N^2(\mathbf{x} - (\tfrac{\mathbf{n}}{N} + \tfrac{(m_1,\overline{\mathbf{m}})}{N^2})) \tag{14}$$

using summation by parts in the direction x_1:

$$\begin{aligned}\widetilde{f}_{2,\mathbf{q}}(\mathbf{x}) &= \lambda \sum_{\overline{\mathbf{m}}\in[-N+1,\dots,N-1]^{\nu-1}} \sum_{m_1=-N+1}^{N} (\Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1,\mathbf{x}) - \Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1-1,\mathbf{x}))A_{\mathbf{q},\mathbf{n}}(m_1,\overline{\mathbf{m}}) \\ &= \lambda \sum_{\overline{\mathbf{m}}\in[-N+1,\dots,N-1]^{\nu-1}} \sum_{m_1=-N+1}^{N-1} \Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1,\mathbf{x})B_{\mathbf{q},\mathbf{n}}(m_1,\overline{\mathbf{m}}),\end{aligned} \tag{15}$$

where we used the identities $A_{\mathbf{q},\mathbf{n}}(-N,\overline{\mathbf{m}}) = A_{\mathbf{q},\mathbf{n}}(N,\overline{\mathbf{m}}) = 0$.

The above representation involves products $\Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1,\mathbf{x})B_{\mathbf{q},\mathbf{n}}(m_1,\overline{\mathbf{m}})$, and we show now how to implement them by a ReLU network. Note that $\Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1,\mathbf{x}) \in [0,1]$ and $B_{\mathbf{q},\mathbf{n}}(m_1,\overline{\mathbf{m}}) \in \{-1,0,1\}$. But for any $x \in [0,1], y \in \{-1,0,1\}$ we can write

$$xy = (x+y-1)_+ + (-x-y)_+ - (-y)_+. \tag{16}$$

4.8 Mapping x to the respective patch. We will use the obtained formula (15) as a basis for computing $\widetilde{f}_{2,\mathbf{q}}(\mathbf{x})$. But the formula is dependent on the patch the input $\mathbf{x}$ belongs to: the value $\Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1,\mathbf{x})$ depends on the patch index $\mathbf{n}$ by Eq.(14), and the values $B_{\mathbf{q},\mathbf{n}}(m_1,\overline{\mathbf{m}})$ must be recovered from the patch-specific encoding weight $b_{\mathbf{q},\mathbf{n}}$.

Given $\mathbf{x}$, the relevant value of $b_{\mathbf{q},\mathbf{n}}$ can be selected among the values for all patches by the ReLU network computing

$$b_{\mathbf{q}}(\mathbf{x}) = \sum_{\mathbf{n}\in(\mathbf{q}+(3\mathbb{Z})^\nu)\cap[0,N]^\nu} \tfrac{b_{\mathbf{q},\mathbf{n}}}{2}((2-u)_+ - (1-u)_+), \text{ where } u = \max_{s=1,\dots,\nu} |Nx_s - n_s|. \tag{17}$$

If $\mathbf{x}$ belongs to some patch $Q_{\mathbf{n}}$, then $b_{\mathbf{q}}(\mathbf{x}) = b_{\mathbf{q},\mathbf{n}}$, as required. If $\mathbf{x}$ does not belong to any patch, then $b_{\mathbf{q}}(\mathbf{x})$ computes some garbage value which will be unimportant because we will ensure that all the factors Φ appearing in the sum (15) will vanish for such $\mathbf{x}$.

Now we show how to perform a properly adjusted computation of $\Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1,\mathbf{x})$. Namely, we will compute the function $\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1,\mathbf{x})$ such that

$$\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1,\mathbf{x}) = \begin{cases} \Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1,\mathbf{x}), & \mathbf{x}\in Q_{\mathbf{n}}, \mathbf{n}\in(\mathbf{q}+(3\mathbb{Z})^\nu)\cap[0,N]^\nu, \quad (18) \\ 0, & \text{otherwise.} \quad (19)\end{cases}$$

First construct, using linear and ReLU operations, a map $\Psi_{\mathbf{q}} : [0,1]^\nu \to (\mathbb{Z}/N)^\nu$ mapping the patches $Q_{\mathbf{n}}$ to their centers:

$$\Psi_{\mathbf{q}}(\mathbf{x}) = \frac{\mathbf{n}}{N}, \quad \text{if } \mathbf{x}\in Q_{\mathbf{n}}, \mathbf{n}\in(\mathbf{q}+(3\mathbb{Z})^\nu)\cap[0,N]^\nu. \tag{20}$$

Such a map can be implemented by

$$\Psi_{\mathbf{q}}(\mathbf{x}) = (\psi_{q_1}(x_1), \dots, \psi_{q_\nu}(x_\nu)), \tag{21}$$

where

$$\psi_q(x) = \frac{q}{N} + \sum_{k=0}^{\lceil N/3 \rceil} ((x - \tfrac{q+3k+1}{N})_+ - (x - \tfrac{q+3k+2}{N})_+). \tag{22}$$

Now if we try to define $\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1, \mathbf{x})$ just by replacing $\frac{\mathbf{n}}{N}$ with $\Psi_{\mathbf{q}}(\mathbf{x})$ in the definition (14), then we fulfill the requirement (18), but not (19). To also fulfill (19), consider the auxiliary "suppressing" function

$$\theta_{\mathbf{q}}(\mathbf{x}) = N \sum_{\mathbf{n} \in (\mathbf{q}+(3\mathbb{Z})^\nu) \cap [0,N]^\nu} (1 - \max_{s=1,\dots,\nu} |Nx_s - n_s|)_+. \tag{23}$$

If $\mathbf{x}$ does not lie in any patch $Q_{\mathbf{n}}$, then $\theta_{\mathbf{q}}(\mathbf{x}) = 0$. On the other hand, $\theta_{\mathbf{q}}(\mathbf{x}) \geq \Phi_{\mathbf{n},\overline{\mathbf{m}}}(m_1, \mathbf{x})$ for all $\mathbf{n} \in (\mathbf{q} + (3\mathbb{Z})^\nu) \cap [0, N]^\nu$, $(m_1, \overline{\mathbf{m}}) \in [-N+1, N-1]^\nu$ and $\mathbf{x} \in [0,1]^\nu$. It follows that if we set

$$\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1, \mathbf{x}) = \min\Big(\sum_{s=-N+1}^{m_1} \phi(N^2(\mathbf{x} - (\Psi_{\mathbf{q}}(\mathbf{x}) + \tfrac{(m_1,\overline{\mathbf{m}})}{N^2}))), \theta_{\mathbf{q}}(\mathbf{x}) \Big), \tag{24}$$

then this $\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1, \mathbf{x})$ satisfies both (18) and (19). Then, based on (15), we can write the final formula for $\widetilde{f}_{2,\mathbf{q}}(\mathbf{x})$ as

$$\widetilde{f}_{2,\mathbf{q}}(\mathbf{x}) = \lambda \sum_{\overline{\mathbf{m}} \in [-N+1,\dots,N-1]^{\nu-1}} \sum_{m_1=-N+1}^{N-1} \widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1, \mathbf{x}) B_{\mathbf{q},\mathbf{n}}(m_1, \overline{\mathbf{m}}). \tag{25}$$

4.9 Summary of the computation of $\widetilde{f}_{2,\mathbf{q}}$. We summarize now how $\widetilde{f}_{2,\mathbf{q}}(\mathbf{x})$ is computed by the ReLU network.

1. The patch-encoding number $b_{\mathbf{q}}(\mathbf{x})$ and the auxiliary functions $\Psi_{\mathbf{q}}(\mathbf{x}), \theta_{\mathbf{q}}(\mathbf{x})$ are computed by Eqs.(17), (20)-(22), (23), respectively.

2. The numbers $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ that describe the patch containing $\mathbf{x}$ are decoded from $b_{\mathbf{q}}(\mathbf{x})$ by a deep network with $O(N^\nu)$ layers and weights as detailed in Section 4.6. We choose the enumeration $t \mapsto \mathbf{m}_t$ so that same-$\overline{\mathbf{m}}$ multi-indices $\mathbf{m}$ form contiguous blocks of length $2N-1$ corresponding to summation over m_1 in (25); the ordering of the multi-indices $\overline{\mathbf{m}}$ is arbitrary.

3. The values $\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1, \mathbf{x})$ are computed iteratively in parallel to the respective $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$, using Eq.(24). For each $m_1 > -N+1$, the sum $\sum_{s=-N+1}^{m_1} \phi(\dots)$ is computed by adding the next term to the sum $\sum_{s=-N+1}^{m_1-1} \phi(\dots)$ retained from the previous step. This shows that all the values $\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1, \mathbf{x})$ can be computed with $O(N^\nu)$ linear and ReLU operations.

4. The sum (25) is computed with the help of multiplication formula (16).

4.10 Computation complexity and implementation by the standard network. It is clear from the construction that the total number of weights and layers in the network implementing the full approximation $\widetilde{f}$ is not greater than $c_\nu N^\nu$ with some ν-dependent constant c_ν. Since, by (12), the approximation error of the network does not exceed $3^\nu(6\nu^{3/2}+3\nu+1)\omega_f(\frac{1}{N^2})$, we obtain the desired bound $\|f-\widetilde{f}\|_\infty \le a_\nu \omega_f(c_\nu W^{-2/\nu})$.

We estimate now the width of the standard network sufficient to implement all the computations. We reserve ν channels to pass forward the ν components of the input $\mathbf{x}$ and one channel to store partial results of the sum $\widetilde{f}(\mathbf{x}) = \widetilde{f}_1(\mathbf{x}) + \sum_{\mathbf{q}\in\{0,1,2\}^\nu} \widetilde{f}_{2,\mathbf{q}}(\mathbf{x})$. The terms in this sum can be computed in a serial way, so we only need to estimate the network width sufficient for implementing $\widetilde{f}_1$ and any $\widetilde{f}_{2,\mathbf{q}}$.

Given $\mathbf{x}$, the value $\phi(\mathbf{x})$ or, more generally, $\phi(N\mathbf{x}-\mathbf{n})$, can be computed by (4) using chained binary min's and just one additional channel. Therefore, by (5), $\widetilde{f}_1$ can be implemented with the standard network of width $\nu+2$.

Now consider implementation of $\widetilde{f}_{2,\mathbf{q}}$ as detailed in Section 4.9. In stage 1, we compute the numbers $b_{\mathbf{q}}(\mathbf{x}), \theta_{\mathbf{q}}(\mathbf{x})$ and the ν-dimensional vector $\Psi_{\mathbf{q}}(\mathbf{x})$. This can be done with $\nu+3$ channels (one extra channel is reserved for storing partial sums). We reserve $\nu+1$ channels for the next stages to pass forward the values $\Psi_{\mathbf{q}}(\mathbf{x}), \theta_{\mathbf{q}}(\mathbf{x})$. In stage 2, we decode the numbers $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ from $b_q(\mathbf{x})$. This can be done with 4 channels: one is used to store and refresh the values z_t, another to output the sequence $B_{\mathbf{q},\mathbf{n}}(\mathbf{m}_t)$, and two more channels to compute $\chi_\epsilon(3z_t)$. In stage 3, we compute the numbers $\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1,\mathbf{x})$. This can be done with 3 channels: one is used to keep partial sums $\sum_{s=-N+1}^{m_1}\phi(\ldots)$, another to compute current $\phi(\ldots)$, and the third to compute $\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1,\mathbf{x})$ by Eq.(24). Stages 2 and 3 are performed in parallel and their channels are aligned so that for each $\mathbf{m}$ the values $B_{\mathbf{q},\mathbf{n}}(\mathbf{m})$ and $\widetilde{\Phi}_{\overline{\mathbf{m}},\mathbf{q}}(m_1,\mathbf{x})$ can be found in the same layer. In stage 4, we compute the sum (25). This stage is performed in parallel to stages 2 and 3 and requires one more channel for computing the ReLU operations in the multiplication formula (16). We conclude that $2\nu+10$ channels are sufficient for the whole computation of $\widetilde{f}_{2,\mathbf{q}}$ and hence for the whole computation of $\widetilde{f}$.

References

Martin Anthony and Peter L Bartlett. *Neural network learning: Theoretical foundations*. Cambridge university press, 2009.

Peter L Bartlett, Vitaly Maiorov, and Ron Meir. Almost linear VC-dimension bounds for piecewise polynomial networks. *Neural computation*, 10(8):2159–2173, 1998.

Peter L Bartlett, Nick Harvey, Chris Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *arXiv preprint arXiv:1703.02930*, 2017.

Monica Bianchini and Franco Scarselli. On the complexity of neural network classifiers: A comparison between shallow and deep architectures. *IEEE transactions on neural networks and learning systems*, 25(8):1553–1565, 2014.

Helmut Bölcskei, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. Memory-optimal neural network approximation. In *Wavelets and Sparsity XVII*, volume 10394, page 103940Q. International Society for Optics and Photonics, 2017.

Ronald A DeVore. Nonlinear approximation. *Acta numerica*, 7:51–150, 1998.

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

Paul W Goldberg and Mark R Jerrum. Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18(2-3):131–148, 1995.

Boris Hanin. Universal function approximation by deep neural nets with bounded width and relu activations. *arXiv preprint arXiv:1708.02691*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Paul C Kainen, Věra Kůrková, and Andrew Vogt. Approximation by neural networks is not continuous. *Neurocomputing*, 29(1):47–56, 1999.

Michael J Kearns and Robert E Schapire. Efficient distribution-free learning of probabilistic concepts. In *Foundations of Computer Science, 1990. Proceedings., 31st Annual Symposium on*, pages 382–391. IEEE, 1990.

Shiyu Liang and R. Srikant. Why deep neural networks? *arXiv preprint arXiv:1610.04161*, 2016.

Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: A view from the width. In *Advances in Neural Information Processing Systems*, pages 6232–6240, 2017.

Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. Learning real and boolean functions: When is deep better than shallow. *arXiv preprint arXiv:1603.00988*, 2016.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932, 2014.

Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep relu neural networks. *arXiv preprint arXiv:1709.05289*, 2017.

Itay Safran and Ohad Shamir. Depth separation in relu networks for approximating smooth non-linear functions. *arXiv preprint arXiv:1610.09887*, 2016.

Akito Sakurai. Tight Bounds for the VC-Dimension of Piecewise Polynomial Networks. In *Advances in Neural Information Processing Systems*, pages 323–329, 1999.

J. Schmidt-Hieber. Nonparametric regression using deep neural networks with ReLU activation function. *arXiv preprint arXiv:1708.06633*, 2017.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

Matus Telgarsky. Benefits of depth in neural networks. *arXiv preprint arXiv:1602.04485*, 2016.

Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.