CENTER FOR
Brains
Minds+
Machines

CBMM Memo No. 054 **August 12, 2016**

Deep vs. Shallow Networks: an Approximation Theory Perspective

by

Hrushikesh N. Mhaskar[1] and Tomaso Poggio[2]

1. Department of Mathematics, California Institute of Technology, Pasadena, CA 91125
Institute of Mathematical Sciences, Claremont Graduate University, Claremont, CA 91711.
hrushikesh.mhaskar@cgu.edu
2. Center for Brains, Minds, and Machines, McGovern Institute for Brain Research,
Massachusetts Institute of Technology, Cambridge, MA, 02139.
tp@mit.edu

Abstract: The paper briefly reviews several recent results on hierarchical architectures for learning from examples, that may formally explain the conditions under which Deep Convolutional Neural Networks perform much better in function approximation problems than shallow, one-hidden layer architectures. The paper announces new results for a non-smooth activation function – the ReLU function – used in present-day neural networks, as well as for the Gaussian networks. We propose a new definition of *relative dimension* to encapsulate different notions of sparsity of a function class that can possibly be exploited by deep networks but not by shallow ones to drastically reduce the complexity required for approximation and learning.

This work was supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF - 1231216. H.M. is supported in part by ARO Grant W911NF-15-1-0385.

1 Introduction

Deep Neural Networks especially of the convolutional type (DCNNs) have started a revolution in the field of artificial intelligence and machine learning, triggering a large number of commercial ventures and practical applications. Most deep learning references these days start with Hinton's backpropagation and with Lecun's convolutional networks (see for a nice review [12]). Of course, multilayer convolutional networks have been around at least as far back as the optical processing era of the 70s. Fukushima's Neocognitron [9] was a convolutional neural network that was trained to recognize characters. The HMAX model of visual cortex [23] was described as a series of AND and OR layers to represent hierarchies of disjunctions of conjunctions. A version of the questions about the importance of hierarchies was asked in [22] as follows: "*A comparison with real brains offers another, and probably related, challenge to learning theory. The "learning algorithms" we have described in this paper correspond to one-layer architectures. Are hierarchical architectures with more layers justifiable in terms of learning theory? It seems that the learning theory of the type we have outlined does not offer any general argument in favor of hierarchical learning machines for regression or classification. This is somewhat of a puzzle since the organization of cortex – for instance visual cortex – is strongly hierarchical. At the same time, hierarchical learning systems show superior performance in several engineering applications.*"

Ironically a mathematical theory characterizing the properties of DCNN's and even simply why they work so well is still missing. Two of the basic theoretical questions about Deep Convolutional Neural Networks (DCNNs) are:

- which classes of functions can they approximate well?
- why is stochastic gradient descent (SGD) so unreasonably efficient?

In this paper we review and extend a theoretical framework that we have introduced very recently to address the first question [20]. The theoretical results include answers to why and when deep networks are better than shallow by using the idealized model of a deep network as a directed acyclic graph (DAG), which we have shown to capture the properties a range of convolutional architectures recently used, such as the very deep convolutional networks of the ResNet type [11]. For compositional functions conforming to a DAG structure with a small maximal indegree of the nodes, such as a binary tree structure, one can bypass the curse of dimensionality with the help of the blessings of compositionality (cf. [6] for a motivation for this terminology). We demonstrate this fact using three examples : traditional sigmoidal networks, the ReLU networks commonly used in DCNN's, and Gaussian networks. The results announced for the ReLU and Gaussian networks are new. We then give examples of different notions of sparsity for which we expect better performance of DCNN's over shallow networks, and propose a quantitative measurement, called relative dimension, encapsulating each of these notions, independently of the different roles the various parameters play in each case.

In Section 2, we explain the motivation for considering compositional functions, and demonstrate how some older results on sigmoidal networks apply for approximation of these functions. In Section 3, we announce our new results in the case of shallow networks implementing the ReLU and Gaussian activation functions. The notion of a compositional function conforming to a DAG structure is explained in Section 4, in which we also demonstrate how the results in Section 3 lead to better approximation bounds for such functions. The ideas behind the proofs of these new theorems are sketched in Section 5. Finally, we make some concluding remarks in Section 6, pointing out a quantitative measurement for three notions of sparsity which we feel may be underlying the superior performance of deep networks.

2 Compositional functions

The purpose of this section is to introduce the concept of compositional functions, and illustrate by an example how this leads to a better approximation power for deep networks. In Sub-section 2.1, we explain how such functions arise in image processing and vision. In Sub-section 2.2, we review some older results for approximation by shallow networks implementing a sigmoidal activation function, and explain how a "good error propagation" helps to generalize these results for deep networks.

2.1 Motivation

Many of the computations performed on images should reflect the symmetries in the physical world that manifest themselves through the image statistics. Assume for instance that a computational hierarchy such as

$$h_l(\cdots h_3(h_{21}(h_{11}(x_1, x_2), h_{12}(x_3, x_4)), h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8))\cdots))) \tag{2.1}$$

is given. Then shift invariance of the image statistics is reflected in the following property: the local node "processors" satisfy $h_{21} = h_{22}$ and $h_{11} = h_{12} = h_{13} = h_{14}$ since there is no reason for them to be different across an image. Similar invariances of image statistics – for instance to scale rotation – can be similarly used to constrain visual algorithms and their parts such as the local processes h.

It is natural to ask whether the hierarchy itself – for simplicity the idealized binary tree of the Figure 3 – follows from a specific symmetry in the world and which one. A possible answer to this question follows from the fact that in natural images the target object is usually among several other objects at a range of scales and position. From the physical point of view, this is equivalent to the observation that there are several localized clusters of surfaces with similar properties (object parts, objects, scenes, etc). These basic aspects of the physical world are reflected in properties of the statistics of images: *locality, shift invariance and scale invariance*. In particular, locality reflects clustering of similar surfaces in the world – the closer to each other pixels are in the image, the more likely they are to be correlated. Thus nearby patches are likely to be correlated (because of locality), at all scales. Ruderman's pioneering work [24] concludes that this set of properties is *equivalent to the statement that natural images consist of many object patches that may partly occlude each other* (object patches are image patches which have similar properties because they are induced by local groups of surfaces with similar properties). We argue that Ruderman's conclusion reflects the compositionality of objects and parts: parts are themselves objects, that is self-similar clusters of similar surfaces in the physical world. The property of *compositionality* was in fact a main motivation for hierarchical architectures such as Fukushima's and later imitations of it such as HMAX which was described as a pyramid of AND and OR layers [23], that is a sequence of conjunctions and disjunctions. According to these arguments, compositional functions should be important for vision tasks because they reflect constraints on visual algorithms.

The following argument shows that compositionality of visual computations is a basic property that follows from the simple requirement of *scalability* of visual algorithms: an algorithm should not change if the size of the image (in pixels) changes. In other words, it should be possible to add or subtract simple reusable parts to the algorithm to adapt it to increased or decreased size of the image without changing its basic core.

A way to formalize the argument is the following. Consider the class of nonlinear functions, mapping vectors from $\mathbb{R}^n$ into $\mathbb{R}^d$ (for simplicity we put in the following $d = 1$). Informally we call an algorithm $K_n : \mathbb{R}^n \mapsto \mathbb{R}$ *scalable* if it maintains the same "form" when the input vectors increase in dimensionality; that is, the same kind of computation takes place when the size of the input vector changes. Specific definitions of scalability and shift invariance for any (one-dimensional) image size lead to the following characterization of scalable, shift-invariant functions or algorithms: *Scalable, shift-invariant functions $K : \mathbb{R}^{2m} \mapsto \mathbb{R}$ have the structure $K = H_2 \circ H_4 \circ H_6 \cdots \circ H_{2m}$, with $H_4 = \tilde{H}_2 \oplus \tilde{H}_2$, $H_6 = H_2^* \oplus H_2^* \oplus H_2^*$, etc., where $\tilde{H}_2$ and H_2^* are suitable functions.*

Thus the structure of *shift-invariant, scalable functions* consists of several layers; each layer consists of identical blocks; each block is a function $H : \mathbb{R}^2 \mapsto \mathbb{R}$: see Figure 1. Obviously, shift-invariant scalable functions are equivalent to shift-invariant compositional functions. The definition can be changed easily in several of its specifics.

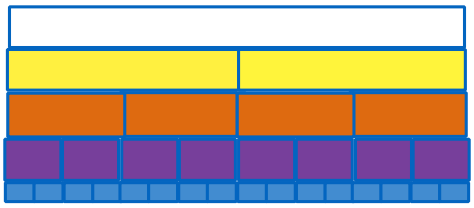

Figure 1: *A scalable function. Each layer consists of identical blocks; each block is a function* $H_2 : \mathbb{R}^2 \mapsto \mathbb{R}$. *The overall function shown in the figure is* $\mathbb{R}^{32} \mapsto \mathbb{R}$

For instance for two-dimensional images the blocks could be operators $H : \mathbb{R}^5 \to \mathbb{R}$ mapping a neighborhood around each pixel into a real number.

The final step in the argument uses the universal approximation property to claim that a nonlinear node with two inputs and enough units (that is, channels) can approximate arbitrarily well each of the H_2 blocks. This leads to conclude that deep convolutional neural networks are natural approximators of *scalable, shift-invariant functions.*

2.2 An example

In this section, we illustrate the advantage of approximating a compositional function using deep networks corresponding to the compositional structure rather than a shallow network that does not take into account this structure.

In the sequel, for any integer $q \geq 1$, $\mathbf{x} = (x_1, \cdots, x_q) \in \mathbb{R}^q$, $|\mathbf{x}|$ denotes the Euclidean ℓ^2 norm of $\mathbf{x}$, and $\mathbf{x} \cdot \mathbf{y}$ denotes the usual inner product between $\mathbf{x}, \mathbf{y} \in \mathbb{R}^q$. In general, we will not complicate the notation by mentioning the dependence on the dimension in these notations unless this might lead to confusion.

Let $I^q = [-1, 1]^q$, $\mathbb{X} = C(I^q)$ be the space of all continuous functions on I^q, with $\|f\| = \max_{\mathbf{x} \in I^q} |f(\mathbf{x})|$. If $\mathbb{V} \subset \mathbb{X}$, we define $\mathsf{dist}(f, \mathbb{V}) = \inf_{P \in \mathbb{V}} \|f - P\|$. Let $\mathcal{S}_n$ denote the class of all shallow networks with n units of the form

$$\mathbf{x} \mapsto \sum_{k=1}^{n} a_k \sigma(\mathbf{w}_k \cdot \mathbf{x} + b_k),$$

where $\mathbf{w}_k \in \mathbb{R}^q$, $b_k, a_k \in \mathbb{R}$. The number of trainable parameters here is $(q + 2)n \sim n$. Let $r \geq 1$ be an integer, and $W_{r,q}^{\mathrm{NN}}$ be the set of all functions with continuous partial derivatives of orders up to r such that $\|f\| + \sum_{1 \leq |\mathbf{k}|_1 \leq r} \|D^{\mathbf{k}} f\| \leq 1$, where $D^{\mathbf{k}}$ denotes the partial derivative indicated by the multi–integer $\mathbf{k} \geq 1$, and $|\mathbf{k}|_1$ is the sum of the components of $\mathbf{k}$.

For explaining our ideas for the deep network, we consider compositional functions conforming to a binary tree. For example, we consider functions of the form (cf. Figure 3)

$$f(x_1, \cdots, x_8) = h_3(h_{21}(h_{11}(x_1, x_2), h_{12}(x_3, x_4)), h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8))). \tag{2.2}$$

For the hierarchical binary tree network, the spaces analogous to $W_{r,q}^{\mathrm{NN}}$ are $W_{H,r,2}^{\mathrm{NN}}$, defined to be the class of all functions f which have the same structure (e.g., (2.2)), where each of the constituent functions h is in $W_{r,2}^{\mathrm{NN}}$ (applied with only 2 variables). We define the corresponding class of deep networks $\mathcal{D}_n$ to be set of all functions with the same structure, where each of the constituent functions is in $\mathcal{S}_n$. We note that in the case when q is an

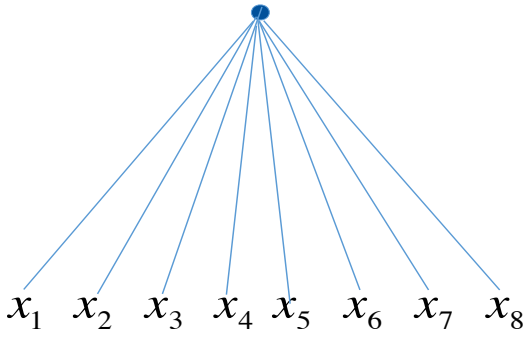

Figure 2: *A shallow universal network in 8 variables and N units which can approximate a generic function $f(x_1, \cdots, x_8)$. The top node consists of n units and computes the ridge function $\sum_{i=1}^{n} a_i \sigma(\langle \mathbf{v}_i, \mathbf{x} \rangle + t_i)$, with $\mathbf{v}_i, \mathbf{x} \in \mathbb{R}^2$, $a_i, t_i \in \mathbb{R}$.*

integer power of 2, the number of parameters involved in an element of $\mathcal{D}_n$ – that is, weights and biases, in a node of the binary tree is $(q-1)(q+2)n$.

The following theorem (cf. [13]) estimates the degree of approximation for shallow and deep networks. We remark that the assumptions on σ in the theorem below are not satisfied by the ReLU function $x \mapsto |x|$, but they are satisfied by smoothing the function in an arbitrarily small interval around the origin.

Theorem 2.1 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be infinitely differentiable, and not a polynomial on any subinterval of $\mathbb{R}$.*
(a) *For $f \in W_{r,q}^{NN}$*

$$\mathsf{dist}(f, \mathcal{S}_n) = \mathcal{O}(n^{-r/q}). \tag{2.3}$$

(b) *For $f \in W_{H,r,2}^{NN}$*

$$\mathsf{dist}(f, \mathcal{D}_n) = \mathcal{O}(n^{-r/2}). \tag{2.4}$$

Proof. Theorem 2.1(a) was proved by [13]. To prove Theorem 2.1(b), we observe that each of the constituent functions being in $W_{r,2}^{\mathrm{NN}}$, (2.3) applied with $q = 2$ implies that each of these functions can be approximated from $\mathcal{S}_n$ up to accuracy $n^{-r/2}$. Our assumption that $f \in W_{H,r,2}^{\mathrm{NN}}$ implies that each of these constituent functions is Lipschitz continuous. Hence, it is easy to deduce that, for example, if P, P_1, P_2 are approximations to the constituent functions h, h_1, h_2, respectively within an accuracy of ϵ, then

$$\begin{aligned} \|h(h_1, h_2) - P(P_1, P_2)\| &\leq \|h(h_1, h_2) - h(P_1, P_2)\| + \|h(P_1, P_2) - P(P_1, P_2)\| \\ &\leq c\{\|h_1 - P_1\| + \|h_2 - P_2\| + \|h - P\|\} \leq 3c\epsilon, \end{aligned}$$

for some constant $c > 0$ independent of ϵ. This leads to (2.4). □

The constants involved in $\mathcal{O}$ in (2.3) will depend upon the norms of the derivatives of f as well as σ. Thus, when the only a priori assumption on the target function is about the number of derivatives, then to **guarantee** an accuracy of ϵ, we need a shallow network with $\mathcal{O}(\epsilon^{-q/r})$ trainable parameters. If we assume a hierarchical structure on the target function as in Theorem 2.1, then the corresponding deep network yields a guaranteed accuracy of ϵ only with $\mathcal{O}(\epsilon^{-2/r})$ trainable parameters.

Is this the best? To investigate this question, we digress and recall the notion of non–linear widths [5]. If $\mathbb{X}$ is a normed linear space, $W \subset \mathbb{X}$ be compact, $M_n : W \to \mathbb{R}^n$ be a continuous mapping (parameter selection), and $A_n : \mathbb{R}^n \to \mathbb{X}$ be any mapping (recovery algorithm). Then an approximation to f is given by $A_n(M_n(f))$, where the continuity of M_n means that the selection of parameters is robust with respect to perturbations in f. The nonlinear n–width of the compact set W is defined by

$$d_n(W) = \inf_{M_n, A_n} \sup_{f \in W} \|f, A_n(M_n(f))\|_{\mathbb{X}}. \tag{2.5}$$

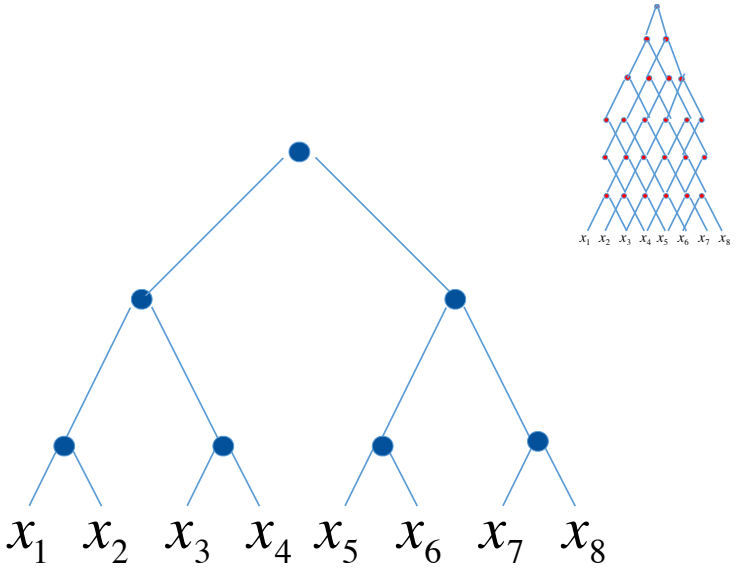

Figure 3: *A binary tree hierarchical network in 8 variables, which approximates well functions of the form (2.2). Each of the nodes consists of n units and computes the ridge function $\sum_{i=1}^{n} a_i\sigma(\langle \mathbf{v}_i, \mathbf{x}\rangle + t_i)$, with $\mathbf{v}_i, \mathbf{x} \in \mathbb{R}^2$, $a_i, t_i \in \mathbb{R}$. Similar to the shallow network such a hierarchical network can approximate any continuous function; the text proves how it approximates compositional functions better than a shallow network. Shift invariance may additionally hold implying that the weights in each layer are the same. The inset at the top right shows a network similar to ResNets: our results on binary trees apply to this case as well with obvious changes in the constants*

We note that the n–width depends only on the compact set W and the space $\mathbb{X}$, and represents the best that can be achieved by **any** continuous parameter selection and recovery processes. It is shown in [5] that $d_n(W_{r,q}^{\mathrm{NN}}) \geq cn^{-r/q}$ for some constant $c > 0$ depending only on q and r. So, the estimate implied by (2.3) is *the best possible* among **all** reasonable methods of approximating arbitrary functions in $W_{r,q}^{\mathrm{NN}}$, although by itself, the estimate (2.3) is blind to the process by which the approximation is accomplished; in particular, this process is not required to be robust. Similar considerations apply to the estimate (2.4).

3 Shallow networks

In this section, we announce our results in the context of shallow networks in two settings. One is the setting of neural networks using the ReLU function $x \mapsto |x| = x_+ + (-x)_+$ (Sub-section 3.1), and the other is the setting of Gaussian networks using an activation function of the form $\mathbf{x} \mapsto \exp(-|\mathbf{x} - \mathbf{w}|^2)$ (Sub-section 3.2). It is our objective to generalize these results to the case of deep networks in Section 4.

Before starting with the mathematical details, we would like to make some remarks regarding the results in this section and in Section 4.

1. It seems unnatural to restrict the range of the constituent functions. Therefore, we are interested in approximating functions on the entire Euclidean space.

2. If one is interested only in error estimates analogous to those in Theorem 2.1, then our results need to be applied to functions supported on the unit cube. One way to ensure that that the smoothness is preserved is to consider a smooth extension of the function on the unit cube to the Euclidean space [25, Chapter VI], and then multiply this extension by a C^∞ function supported on $[-2,2]^q$, equal to 1 on the unit cube. However, this destroys the constructive nature of our theorems.

3. A problem of central importance in approximation theory is to determine what constitutes the right smoothness and the right measurement of complexity. The number of parameters or the number of non–linear units is not necessarily the right measurement for complexity. Likewise, the number of derivatives is not necessarily the right measure for smoothness for every approximation process. In this paper, we illustrate this by showing that different smoothness classes and notions of complexity lead to satisfactory approximation theorems.

3.1 ReLU networks

In this section, we are interested in approximating functions on $\mathbb{R}^q$ by networks of the form $\mathbf{x} \mapsto \sum_{k=1}^n a_k|\mathbf{x}\cdot\mathbf{v}_k + b_k|$, $a_k, b_k \in \mathbb{R}$, $\mathbf{x}, \mathbf{v}_k \in \mathbb{R}^q$. The set of all such functions will be denoted by $\mathcal{R}_{n,q}$. Obviously, these networks are not bounded on the whole Euclidean space. Therefore, we will study the approximation in weighted spaces, where the norm is defined by

$$\|f\|_{w,q} = \operatorname*{ess\,sup}_{\mathbf{x}\in\mathbb{R}^q} \frac{|f(\mathbf{x})|}{\sqrt{|\mathbf{x}|^2+1}}.$$

The symbol $X_{w,q}$ will denote the set of all continuous functions $f : \mathbb{R}^q \to \mathbb{R}$ for which $(|\mathbf{x}|^2+1)^{-1/2}f(\mathbf{x}) \to 0$ as $\mathbf{x} \to \infty$. We will define a "differential operator" $\mathcal{D}$ and smoothness classes $W_{w,\gamma,q}$ in terms of this operator in Section 5.1.

In the sequel, we will adopt the following convention. The notation $A \lesssim B$ means $A \le cB$ for some generic positive constant c that may depend upon fixed parameters in the discussion, such as γ, q, but independent of the target function and the number of parameters in the approximating network. By $A \sim B$, we mean $A \lesssim B$ and $B \lesssim A$.

Our first main theorem is the following Theorem 3.1. We note two technical novelties here. One is that the activation function $|\cdot|$ does not satisfy the conditions of Theorem 2.1. Second is that the approximation is taking place on the whole Euclidean space rather than on a cube as in Theorem 2.1.

Theorem 3.1 *Let $\gamma > 0$, $n \ge 1$ be an integer, $f \in W_{w,\gamma,q}$. Then there exists $P \in \mathcal{R}_{n,q}$ such that*

$$\|f - P\|_{w,q} \lesssim n^{-\gamma/q}\|f\|_{w,\gamma,q}. \tag{3.1}$$

3.2 Gaussian networks

We wish to consider shallow networks where each channel evaluates a Gaussian non–linearity; i.e., Gaussian networks of the form

$$G(\mathbf{x}) = \sum_{k=1}^n a_k \exp(-|\mathbf{x}-\mathbf{x}_k|^2), \qquad \mathbf{x} \in \mathbb{R}^q. \tag{3.2}$$

It is natural to consider the number of trainable parameters $(q+1)n$ as a measurement of the complexity of G. However, it is known ([15]) that an even more important quantity that determines the approximation power of

Gaussian networks is the minimal separation among the centers. For any subset $\mathcal{C}$ of $\mathbb{R}^q$, the minimal separation of $\mathcal{C}$ is defined by

$$\eta(\mathcal{C}) = \inf_{\mathbf{x},\mathbf{y}\in\mathcal{C},\mathbf{x}\neq\mathbf{y}} |\mathbf{x}-\mathbf{y}|. \tag{3.3}$$

For $n, m > 0$, the symbol $\mathcal{N}_{n,m}(\mathbb{R}^q)$ denotes the set of all Gaussian networks of the form (3.2), with $\eta(\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}) \geq 1/m$.

Let $\mathbb{X}_q$ be the space of continuous functions on $\mathbb{R}^q$ vanishing at infinity, equipped with the norm $\|f\|_q = \max_{\mathbf{x}\in\mathbb{R}^q} |f(\mathbf{x})|$.

In order to measure the smoothness of the target function, we need to put conditions not just on the number of derivatives but also on the rate at which these derivatives tend to 0 at infinity. Generalizing an idea from [8, 14], we define first the space $W_{r,q}$ for integer $r \geq 1$ as the set of all functions f which are r times iterated integrals of functions in $\mathbb{X}$, satisfying

$$\|f\|_{r,q} = \|f\|_q + \sum_{1\leq|\mathbf{k}|_1\leq r} \|\exp(-|\cdot|^2)D^{\mathbf{k}}(\exp(|\cdot|^2)f)\|_q < \infty.$$

Since one of our goals is to show that our results on the upper bounds for the accuracy of approximation are the best possible for individual functions, the class $W_{r,q}$ needs to be refined somewhat. Toward that goal, we define next a regularization expression, known in approximation theory parlance as a K–functional, by

$$K_{r,q}(f,\delta) = \inf_{g\in W_{r,q}} \{\|f-g\|_q + \delta^r(\|g\|_q + \|g\|_{r,q})\}.$$

We note that the infimum above is over **all** g in the class $W_{r,q}$ rather than just the class of all networks. The class $\mathcal{W}_{\gamma,q}$ of functions which we are interested in is then defined for $\gamma > 0$ as the set of all $f \in \mathbb{X}_q$ for which

$$\|f\|_{\gamma,q} = \|f\|_q + \sup_{\delta\in(0,1]} \frac{K_{r,q}(f,\delta)}{\delta^\gamma} < \infty,$$

for some integer $r \geq \gamma$. It turns out that different choices of r yield equivalent norms, without changing the class itself. The following theorem gives a bound on approximation of $f \in \mathbb{X}_q$ from $\mathcal{N}_{N,m}(\mathbb{R}^q)$. The following theorem is proved in [15].

Theorem 3.2 *Let $\{\mathcal{C}_m\}$ be a sequence of finite subsets with $\mathcal{C}_m \subset [-cm, cm]^q$, with*

$$1/m \lesssim \max_{\mathbf{y}\in[-cm,cm]^q} \min_{\mathbf{x}\in\mathcal{C}} |\mathbf{x}-\mathbf{y}| \lesssim \eta(\mathcal{C}_m), \qquad m = 1, 2, \cdots. \tag{3.4}$$

Let $1 \leq p \leq \infty$, $\gamma > 0$, and $f \in \mathcal{W}_{\gamma,q}$. Then for integer $m \geq 1$, there exists $G \in \mathcal{N}_{|\mathcal{C}_m|,m}(\mathbb{R}^q)$ with centers at points in $\mathcal{C}_m$ such that

$$\|f-G\|_q \lesssim \frac{1}{m^\gamma}\|f\|_{\gamma,q}. \tag{3.5}$$

Moreover, the coefficients of G can be chosen as linear combinations of the data $\{f(\mathbf{x}) : \mathbf{x} \in \mathcal{C}_m\}$.

We note that the set of centers $\mathcal{C}_m$ can be chosen arbitrarily subject to the conditions stated in the theorem; **there is no training necessary to determine these parameters**. Therefore, there are only $\mathcal{O}(m^{2q})$ coefficients to be found by training. This means that if we assume a priori that $f \in \mathcal{W}_{\gamma,q}$, then the number of trainable parameters to theoretically guarantee an accuracy of $\epsilon > 0$ is $\mathcal{O}(\epsilon^{-2q/\gamma})$. For the unit ball $\mathcal{B}_{\gamma,q}$ of the class $\mathcal{W}_{\gamma,q}$ as defined in Section 3.2, the Bernstein inequality proved in [19] leads to $d_n(\mathcal{B}_{\gamma,q}) \sim n^{-\gamma/(2q)}$. Thus, the estimate (3.5) is the best possible in terms of widths. This implies in particular that when the networks are computed using samples of f to obtain an accuracy of ϵ in the approximation, one needs $\sim \epsilon^{-2q/\gamma}$ samples. When f is compactly supported, $\|f\|_{\gamma,q}$ is of the same order of magnitude as the norm of f corresponding to the K-functional based on the smoothness class W_r^{NN} in Section 2.2. However, the number of parameters is then not commesurate with the results in that section.

We observe that the width estimate holds for the approximation of the entire class, and hence, an agreement with such width estimate implies only that there exists a possibly pathological function for which the approximation

estimate cannot be improved. How good is the estimate in Theorem 3.2 for individual functions? If we know that some oracle can give us Gaussian networks that achieve a given accuracy with a given complexity, does it necessarily imply that the target function is smooth as indicated by the above theorems? The following is a converse to Theorem 3.2 demonstrating that the accuracy asserted by these theorems is possible if and only if the target function is in the smoothness class required in these theorems. It demonstrates also that rather than the number of nonlinearities in the Gaussian network, it is the minimal separation among the centers that is the "right" measurement for the complexity of the networks. Theorem 3.3 below is a refinement of the corresponding result in [15].

Theorem 3.3 *Let $\{\mathcal{C}_m\}$ be a sequence of finite subsets of $\mathbb{R}^q$, such that for each integer $m \geq 1$, $\mathcal{C}_m \subseteq \mathcal{C}_{m+1}$, $|\mathcal{C}_m| \leq c\exp(c_1 m^2)$, and $\eta(\mathcal{C}_m) \geq 1/m$. Further, let $f \in \mathbb{X}_q$, and for each $m \geq 1$, let G_m be a Gaussian network with centers among points in $\mathcal{C}_m$, such that*

$$\sup_{m\geq 1} m^\gamma \|f - G_m\|_q < \infty. \tag{3.6}$$

Then $f \in \mathcal{W}_{\gamma,q}$.

We observe that Theorem 3.2 can be interpreted to give estimates on the degree of approximation by Gaussian networks either in terms of the number of non–linear units, or the number of trainable parameters, or the minimal separation among the centers, or the number of samples of the target function. Theorem 3.3 shows that the right model of complexity among these is the minimal separation among the centers. Using this measurement for complexity yields "matching" direct and converse theorems. Based on the results in [18], we expect that a similar theorem should be true also for ReLU networks.

4 Deep networks

The purpose of this section is to generalize the results in Section 3 to the case of deep networks. In Sub-section 4.1, we will formulate the concept of compositional functions in terms of a DAG, and introduce the related mathematical concepts for measuring the degree of approximation and smoothness. The approximation theory results in this context will be described in Sub-section 4.2.

4.1 General DAG functions

Let $\mathcal{G}$ be a directed acyclic graph (DAG), with the set of nodes V. A $\mathcal{G}$–function is defined as follows. The in-edges to each node of $\mathcal{G}$ represents an input real variable. The node itself represents the evaluation of a real valued function of the inputs. The out-edges fan out the result of this evaluation. Each of the source node obtains an input from some Euclidean space. Other nodes can also obtain such an input. We assume that there is only one sink node, whose output is the $\mathcal{G}$-function. For example, the DAG in Figure 4 represents the $\mathcal{G}$–function

$$\begin{aligned} &f^*(x_1,\cdots,x_9) = \\ &\quad h_{19}(h_{17}(h_{13}(h_{10}(x_1,x_2,x_3,h_{16}(h_{12}(x_6,x_7,x_8,x_9))),h_{11}(x_4,x_5)),h_{14}(h_{10},h_{11}),h_{16}),h_{18}(h_{15}(h_{11},h_{12}),h_{16})) \end{aligned} \tag{4.1}$$

We note that if q is the number of source nodes in $\mathcal{G}$, a $\mathcal{G}$–function is a function on $\mathbb{R}^q$. Viewed only as a function on $\mathbb{R}^q$, it is not clear whether two different DAG structures can give rise to the same function. Even if we assume a certain DAG, it is not clear that the choice of the constituent functions is uniquely determined for a given function on $\mathbb{R}^q$. For our mathematical analysis, we therefore find it convenient to think of a $\mathcal{G}$–function as a set of functions

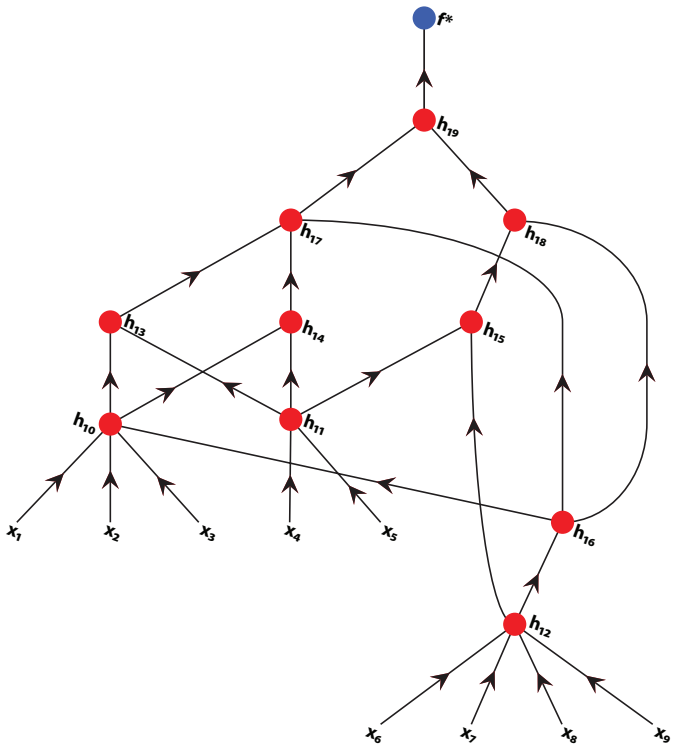

Figure 4: *An example of a $\mathcal{G}$–function (f^* given in (4.1)). The vertices of the DAG $\mathcal{G}$ are denoted by red dots. The black dots represent the input to the various nodes as indicated by the in–edges of the red nodes, and the blue dot indicates the output value of the $\mathcal{G}$–function, f^* in this example.*

$f = \{f_v : \mathbb{R}^{d(v)} \to \mathbb{R}\}_{v\in V}$, rather than a single function on $\mathbb{R}^q$. The individual functions f_v will be called *constituent functions*.

We adopt the convention that for any function class $\mathbb{X}(\mathbb{R}^d)$, the class $\mathcal{G}\mathbb{X}$ denotes the set of $\mathcal{G}$ functions $f = \{f_v\}_{v\in V}$, where each constituent function $f_v \in \mathbb{X}(\mathbb{R}^{d(v)})$. We define

$$\|f\|_{\mathcal{G},\mathbb{X}} = \sum_{v\in V} \|f_v\|_{\mathbb{X}(\mathbb{R}^{d(v)})}. \tag{4.2}$$

4.2 Approximation using deep networks

First, we discuss the analogue of Theorem 3.1 in Section 3.1 for deep networks conforming to the DAG $\mathcal{G}$. We define the classes $\mathcal{G}X_w$ and $\mathcal{G}W_{w,\gamma}$ in accordance with the notation introduced in Section 4.1, and denote the norm on $\mathcal{G}X_w$ (respectively, $\mathcal{G}W_{w,\gamma}$) by $\|\cdot\|_{\mathcal{G},w}$ (respectively, $\|\cdot\|_{\mathcal{G},w,\gamma}$). The symbol $\mathcal{G}\mathcal{R}_n$ denotes the family of networks $\{P_v \in \mathcal{R}_{n,d(v)}\}_{v\in V}$. The analogue of Theorem 3.1 is the following.

Theorem 4.1 *Let $1 \le \gamma \le 2$, $n \ge 1$ be an integer, $f \in \mathcal{G}W_{w,\gamma}$, $d = \max_{v\in V} d(v)$. Then there exists $P \in \mathcal{G}\mathcal{R}_n$ such that*

$$\|f - P\|_{\mathcal{G},w} \lesssim n^{-\gamma/d}\|f\|_{\mathcal{G},w,\gamma}. \tag{4.3}$$

We observe that if $P = \{P_v\} \in \mathcal{G}\mathcal{R}_n$ the number of trainable parameters in each constituent network P_v is $\mathcal{O}(n)$. Therefore, the total number of trainable parameters in P is $\mathcal{O}(|V|n)$. Equivalently, when the target function is in $\mathcal{G}W_{w,\gamma}$, one needs $\mathcal{O}((\epsilon/|V|)^{-d/\gamma})$ units in a deep network to achieve an accuracy of at most ϵ. If one ignores the compositional structure of the target function, (3.1) shows that one needs $O(\epsilon^{-q/\gamma})$ units in a shallow network.

Thus, a deep network conforming to the structure of the target function yields a substantial improvement over a shallow network if $d \ll q$.

Next, we discuss deep Gaussian networks. As before, the spaces $\mathcal{G}\mathbb{X}$ and $\mathcal{G}W_\gamma$ are as described in Section 4.1, and denote the corresponding norms $\|\cdot\|_{\mathcal{G}\mathbb{X}}$ (respectively, $\|\cdot\|_{\mathcal{G}W_\gamma}$) by $\|\cdot\|_{\mathcal{G}}$ (respectively, $\|\cdot\|_{\mathcal{G},\gamma}$).

The analogue of Theorem 3.2 and Theorem 3.3 are parts (a) and (b) respectively of the following Theorem 4.2.

Theorem 4.2 (a) *For each $v \in V$, let $\{\mathcal{C}_{m,v}\}$ be a sequence of finite subsets as described in Theorem 3.2. Let $\gamma \geq 1$ and $f \in \mathcal{GW}_\gamma$. Then for integer $m \geq 1$, there exists $G \in \mathcal{GN}_{\max|\mathcal{C}_{m,v}|,m}$ with centers of the constituent network G_v at vertex v at points in $\mathcal{C}_{m,v}$ such that*

$$\|f - G\|_{\mathcal{G}} \lesssim \frac{1}{m^\gamma}\|f\|_{\mathcal{G},\gamma}. \tag{4.4}$$

Moreover, the coefficients of each constituent G_v can be chosen as linear combinations of the data $\{f_v(\mathbf{x}) : \mathbf{x} \in \mathcal{C}_{m,v}\}$.
(b) *For each $v \in V$, let $\{\mathcal{C}_{m,v}\}$ be a sequence of finite subsets of $\mathbb{R}^{d(v)}$, satisfying the conditions as described in part (a) above. Let $f \in \mathcal{G}\mathbb{X}$, $\gamma > 0$, and $\{G_m \in \mathcal{GN}_{n,m}\}$ be a sequence where, for each $v \in V$, the centers of the constitutent networks $G_{m,v}$ are among points in $\mathcal{C}_{m,v}$, and such that*

$$\sup_{m\geq 1} m^\gamma \|f - G_m\|_{\mathcal{G}} < \infty. \tag{4.5}$$

Then $f \in \mathcal{GW}_\gamma$.

5 Ideas behind the proofs

5.1 Theorem 3.1.

The proof of this theorem has two major steps. One is a reproduction formula ((5.9) below), and the other is the definition of smoothness. Both are based on "wrapping" the target function from $\mathbb{R}^q$ to a function $\mathcal{S}(f)$ (cf. (5.3) below) on the unit Euclidean sphere $\mathbb{S}^q$, defined by

$$\mathbb{S}^q = \{\mathbf{u} \in \mathbb{R}^{q+1} : |\mathbf{u}| = 1\}.$$

A parametrization of the upper hemisphere $\mathbb{S}^q_+ = \{\mathbf{u} \in \mathbb{S}^q : u_{q+1} > 0\}$ of $\mathbb{S}^q$ is given by

$$u_j = \frac{x_j}{\sqrt{|\mathbf{x}|^2+1}}, \quad j = 1, \cdots, q, \quad u_{q+1} = (|\mathbf{x}|^2+1)^{-1/2}, \quad \mathbf{u} \in \mathbb{S}^q_+, \ \mathbf{x} \in \mathbb{R}^q, \tag{5.1}$$

with the inverse mapping

$$x_j = \frac{u_j}{u_{q+1}}, \qquad j = 1, \cdots, q, \qquad \mathbf{u} \in \mathbb{S}^q_+, \ \mathbf{x} \in \mathbb{R}^q. \tag{5.2}$$

Next, we define an operator $\mathcal{S}$ on $X_{w,q}$ by

$$\mathcal{S}(f)(\mathbf{u}) = |u_{q+1}| f\left(\frac{u_1}{u_{q+1}}, \cdots, \frac{u_q}{u_{q+1}}\right), \qquad f \in X_{w,q}. \tag{5.3}$$

We note that if $f \in X_{w,q}$, then $(|\mathbf{x}|^2+1)^{-1/2} f(\mathbf{x}) \to 0$ as $|\mathbf{x}| \to \infty$. Therefore, $\mathcal{S}(f)$ is well defined, and defines an even, continuous function on $\mathbb{S}^q$, equal to 0 on the "equator" $u_{q+1} = 0$.

Next, let μ^* be the Riemannian volume measure on $\mathbb{S}^q$, with $\mu^*(\mathbb{S}^q) = \omega_q$. In this subsection, we denote the dimension of the space of all homogeneous spherical polynomials of degree ℓ by d_ℓ, $\ell = 0, 1, \cdots$, and the set of orthonormalized spherical harmonics on $\mathbb{S}^q$ by $\{Y_{\ell,k}\}_{k=1}^{d_\ell}$. If $F \in L^1(\mathbb{S}^q)$, then

$$\hat{F}(\ell, k) = \int_{\mathbb{S}^q} F(\mathbf{u}) Y_{\ell,k}(\mathbf{u}) d\mu^*(\mathbf{u}). \tag{5.4}$$

We note that if F is an even function, then $\hat{F}(2\ell+1,k)=0$ for $\ell=0,1,\cdots$.

Next, we recall the addition formula

$$\sum_{k=1}^{d_\ell} Y_{\ell,k}(\mathbf{u})\overline{Y_{\ell,k}(\mathbf{v})} = \omega_{q-1}^{-1} p_\ell(1) p_\ell(\mathbf{u}\cdot\mathbf{v}), \tag{5.5}$$

where p_ℓ is the degree ℓ ultraspherical polynomial with positive leading coefficient, with the set $\{p_\ell\}$ satisfying

$$\int_{-1}^{1} p_\ell(t)p_j(t)(1-t^2)^{q/2-1}dt = \delta_{j,\ell}, \qquad j,\ell=0,1,\cdots. \tag{5.6}$$

The function $t\to|t|$ can be expressed in an expansion

$$|t| \sim p_0 - \sum_{\ell=1}^{\infty} \frac{\ell-1}{\ell(2\ell-1)(\ell+q/2)} p_{2\ell}(0)p_{2\ell}(t), \qquad t\in[-1,1], \tag{5.7}$$

with the series converging on compact subsets of $(-1,1)$.

We define the ϕ–derivative of F formally by

$$\widehat{\mathcal{D}_\phi F}(2\ell,k) = \begin{cases} \hat{F}(0,0), & \text{if } \ell=0, \\ -\dfrac{\ell(2\ell-1)(\ell+q/2)p_{2\ell}(1)}{\omega_{q-1}(\ell-1)p_{2\ell}(0)}\hat{F}(2\ell,k), & \text{if } \ell=1,2,\cdots, \end{cases} \tag{5.8}$$

and $\widehat{\mathcal{D}_\phi F}(2\ell+1,k)=0$ otherwise. Then for an even function $F\in L^1(\mathbb{S}^q)$ for which $\mathcal{D}_\phi F\in L^1(\mathbb{S}^q)$, we deduce the reproducing kernel property:

$$F(\mathbf{u}) = \int_{\mathbb{S}^q} |\mathbf{u}\cdot\mathbf{v}|\mathcal{D}_\phi F(\mathbf{v})d\mu^*(\mathbf{v}). \tag{5.9}$$

A careful discretization of this formula using polynomial approximations of both the terms in the integrand as in [17, 18] leads to a zonal function network of the form $\mathbf{u}\mapsto\sum_{k=0}^n a_k|\mathbf{u}\cdot\mathbf{v}_k|$, $a_k\in\mathbb{R}$, $\mathbf{v}_k\in\mathbb{S}^q$, satisfying

$$\left|F(\mathbf{u}) - \sum_{k=0}^n a_k|\mathbf{u}\cdot\mathbf{v}_k|\right| \lesssim n^{-1/q} \operatorname*{ess\,sup}_{\mathbf{v}\in\mathbb{S}^q} |\mathcal{D}_\phi F(\mathbf{v})|, \qquad \mathbf{u}\in\mathbb{S}^q. \tag{5.10}$$

Next, we define formally

$$\mathcal{D}(f)(\mathbf{x}) = (|\mathbf{x}|^2+1)^{1/2}\mathcal{D}_\phi(\mathcal{S}(f))\left(\frac{x_1}{\sqrt{|\mathbf{x}|^2+1}},\cdots,\frac{x_q}{\sqrt{|\mathbf{x}|^2+1}},\frac{1}{\sqrt{|\mathbf{x}|^2+1}}\right). \tag{5.11}$$

The estimate (5.10) now leads easily for all $f\in X_{w,q}$ for which $\mathcal{D}(f)\in X_{w,q}$ to

$$\left|f(\mathbf{x}) - \sum_{k=1}^n \frac{a_k}{\sqrt{|\mathbf{x}_k|^2+1}}|\mathbf{x}\cdot\mathbf{x}_k+1|\right| \lesssim n^{-1/q}\|\mathcal{D}(f)\|_{w,q}, \tag{5.12}$$

where $\mathbf{x}_k$ is defined by $(\mathbf{x}_k)_j = (\mathbf{v}_k)_j/(\mathbf{v}_k)_{q+1}$, $j=1,\cdots,q$.

In order to define the smoothness class $W_{w,\gamma,q}$, we first define the K–functional

$$K_w(f,\delta) = \inf\left\{\|f-g\|_{w,q} + \delta\|\mathcal{D}(g)\|_{w,q}\right\}, \tag{5.13}$$

where the infimum is taken over all g for which $\mathcal{D}g\in X_{w,q}$. Finally, the smoothness class $W_{w,\gamma,q}$ is defined to be the set of all $f\in X_{w,q}$ such that

$$\|f\|_{w,\gamma,q} = \|f\|_{w,q} + \sup_{0<\delta<1}\frac{K_w(f,\delta)}{\delta^\gamma} < \infty.$$

The estimate (5.12) then leads to (3.1) in a standard manner.

We remark here that the unit cube $[-1,1]^q$ is mapped to some compact subset of $\mathbb{S}^q_+$. However, the operator $\mathcal{D}$ does not have an obvious interpretation in terms of ordinary derivatives on the cube.

5.2 Theorems 3.2 and 3.3.

In this section, let $\{\psi_j\}$ denote the sequence of orthonormalized Hermite functions; i.e., [26, Formulas (5.5.3), (5.5.1)]

$$\psi_j(x) = \frac{(-1)^j}{\pi^{1/4}2^{j/2}\sqrt{j!}}\exp(x^2/2)\left(\frac{d}{dx}\right)^j(\exp(-x^2)), \qquad x \in \mathbb{R},\ j = 0, 1, \cdots . \tag{5.14}$$

The multivariate Hermite functions are defined by

$$\psi_{\mathbf{j}}(\mathbf{x}) = \prod_{\ell=1}\psi_{j_\ell}(x_\ell). \tag{5.15}$$

We note that

$$\int_{\mathbb{R}^q}\psi_{\mathbf{j}}(\mathbf{z})\psi_{\mathbf{k}}(\mathbf{z})d\mathbf{z} = \delta_{\mathbf{j},\mathbf{k}}, \qquad \mathbf{j},\mathbf{k} \in \mathbb{Z}_+^q. \tag{5.16}$$

Using the Mehler formula [1, Formula (6.1.13)], it can be shown that

$$\psi_{\mathbf{j}}(\mathbf{y}) = \frac{3^{|\mathbf{j}|/2}}{(2\pi)^{q/2}}\int_{\mathbb{R}^q}\exp(-|\mathbf{y}-\mathbf{w}|^2)\exp(-|\mathbf{w}|^2/3)\psi_{\mathbf{j}}(2\mathbf{w}/\sqrt{3})d\mathbf{w}. \tag{5.17}$$

We combine the results on function approximation and quadrature formulas developed in [19] to complete the proof of Theorem 3.2.

To prove Theorem 3.3, we modify the ideas in [16] to obtain a Berstein–type inequality for Gaussian networks of the form

$$\|g\|_{r,q} \lesssim m^r\|g\|_q, \qquad g \in N_{N,m}, \quad N \lesssim \exp(cm^2). \tag{5.18}$$

The proof of Theorem 3.3 then follows standard arguments in approximation theory.

5.3 Results in Section 4.2.

Theorems 4.1 and 4.2(a) follow from Theorems 3.1 and 3.2 respectively by the "good error propagation property" as in the proof of Theorem 2.1(b) from Theorem 2.1(a). Our definitions of the norms for function spaces associated with deep networks ensure that a bound of the form (4.5) implies a bound of the form (3.6) for each of the constituent functions. Therefore, Theorem 3.3 leads to Theorem 4.2(b).

6 Blessed representations

As pointed out in Sections 2.2 and 4, *there are deep networks – for instance of the convolutional type – that can bypass the curse of dimensionality when learning functions blessed with compositionality.* In this section, we explore possible definitions of blessed function representations that can be exploited by deep but not by shallow networks to reduce the complexity of learning. We list three examples, each of a different type.

- The main example consists of the compositional functions defined in this paper in terms of DAGs (Figure 4). The simplest DAG is a binary tree (see Figure 3) corresponding to compositional functions of the type

$$f(x_1, \cdots, x_8) = h_3(h_{21}(h_{11}(x_1, x_2), h_{12}(x_3, x_4)), h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8))).$$

As explained in previous sections, such compositional functions can be approximated well by deep networks. In particular, if the function form above has shift symmetry, it takes the form

$$f(x_1, \cdots, x_8) = h_3(h_2(h_1(x_1, x_2), h_1(x_3, x_4)), h_2(h_1(x_5, x_6), h_1(x_7, x_8))).$$

that can be approximated well by a Deep Convolutional Network (that is with "weight sharing") but not by a shallow one. This first example is important because compositionality seems a common feature of algorithms applied to signals originating from our physical world, such as images. Not surprisingly, binary-like tree structures (the term binary-like covers obvious extensions to two-dimensional inputs such as images) represemt well the architecture of the most successful DCNN.

- Consider that the proof of Theorem 2.1 relies upon the fact that when σ satisfies the conditions of that theorem, the algebraic polynomials in q variables of (total or coordinatewise) degree $< n$ are in the uniform closure of the span of $\mathcal{O}(n^q)$ functions of the form $\mathbf{x} \mapsto \sigma(\mathbf{w} \cdot \mathbf{x} + b)$. The advantage of deep nets is due to the fact that polynomials of smaller number of variables lead to a nominally high degree polynomial through repeated composition. As a simple example, we consider the polynomial

$$Q(x_1, x_2, x_3, x_4) = (Q_1(Q_2(x_1, x_2), Q_3(x_3, x_4)))^{1024},$$

where Q_1, Q_2, Q_3 are bivariate polynomials of total degree ≤ 2. Nominally, Q is a polynomial of total degree 4096 in 4 variables, and hence, requires $\binom{4100}{4} \approx (1.17) * 10^{13}$ parameters without any prior knowledge of the compositional structure. However, the compositional structure implies that each of these coefficients is a function of only 18 parameters. In this case, the representation which makes deep networks approximate the function with a smaller number of parameters than shallow networks is based on polynomial approximation of functions of the type $g(g(g()))$.

- As a different example, we consider a function which is a linear combination of n tensor product Chui–Wang spline wavelets [2], where each wavelet is a tensor product cubic spline. It is shown in [3, 4] that is impossible to implement such a function using a shallow neural network with a sigmoidal activation function using $\mathcal{O}(n)$ neurons, but a deep network with the activation function $(x_+)^2$ can do so. This case is even less general than the previous one but it is interesting because shallow networks are provably unable to implement these splines using a fixed number of units. In general, this does not avoid the curse of dimensionality, but it shows that deep networks provide, unlike shallow networks, local and multi–scale approximation since the spline wavelets are compactly supported with shrinking supports.

- Examples of functions that cannot be represented efficiently by shallow networks have been given very recently by [27]. The results in [7] illustrate the power of deep networks compared to shallow ones, similar in spirit to [3, 4].

The previous examples show three different kinds of "sparsity" that allow a blessed representation by deep networks with a much smaller number of parameters than by shallow networks. This state of affairs motivates the following general definition of *relative dimension*. Let $d_n(W)$ be the non–linear n-width of a function class W. For the unit ball $\mathcal{B}_{\gamma,q}$ of the class $\mathcal{W}_{\gamma,q}$ as defined in Section 3.2, the Bernstein inequality proved in [19] leads to $d_n(\mathcal{B}_{\gamma,q}) \sim n^{-\gamma/(2q)}$. In contrast, for the unit ball $\mathcal{GB}_\gamma$ of the class we have shown that $d_n(\mathcal{GB}_\gamma) \leq cn^{-\gamma/(2d)}$, where $d = \max_{v \in V} d(v)$.

Generalizing, let $\mathbb{V}$, $\mathbb{W}$ be compact subsets of a metric space $\mathbb{X}$, and $d_n(\mathbb{V})$ (respectively, $d_n(\mathbb{W})$) be their n–widths. We define the *relative dimension* of $d_n(\mathbb{V})$ with respect to $d_n(\mathbb{W})$ by

$$D(\mathbb{V}, \mathbb{W}) = \limsup_{n \to \infty} \frac{\log d_n(\mathbb{W})}{\log d_n(\mathbb{V})}. \tag{6.1}$$

Thus, $D(\mathcal{GB}_\gamma, \mathcal{B}_{\gamma,q}) \leq d/q$. This leads us to say that $\mathbb{V}$ is *parsimonious* with respect to $\mathbb{W}$ if $D(\mathbb{V}, \mathbb{W}) \ll 1$.

As we mentioned in previous papers [21, 20] this definition, and in fact most of the previous results, can be specialized to the class of Boolean functions which map the Boolean cube into reals, yielding a number of known [10] and new results. This application will be described in a forthcoming paper.

7 Conclusion

A central problem of approximation theory is to determine the correct notions of smoothness classes of target functions and the correct measurement of complexity for the approximation spaces. This definition is dictated by having "matching" direct and converse theorems. In this paper, we have demonstrated how different smoothness classes lead to satisfactory results for approximation by ReLU networks and Gaussian networks on the entire Euclidean space. Converse theorem is proved for Gaussian networks, and results in [18] suggest that a similar statement ought to be true for ReLU networks as well. These results indicate that the correct measurement of network complexity is not necessarily the number of parameters. We have initiated a discussion of notions of sparsity which we hope would add deeper insights into this area.

References

[1] G. E. Andrews, R. Askey, and R. Roy. *Special functions*, volume 71. Cambridge university press, 1999.

[2] C. K. Chui. *An introduction to wavelets*. Academic press, 1992.

[3] C. K. Chui, X. Li, and H. N. Mhaskar. Neural networks for localized approximation. *Mathematics of Computation*, 63(208):607–623, 1994.

[4] C. K. Chui, X. Li, and H. N. Mhaskar. Limitations of the approximation capabilities of neural networks with one hidden layer. *Advances in Computational Mathematics*, 5(1):233–243, 1996.

[5] R. A. DeVore, R. Howard, and C. A. Micchelli. Optimal nonlinear approximation. *Manuscripta mathematica*, 63(4):469–478, 1989.

[6] D. L. Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000.

[7] R. Eldan and O. Shamir. The power of depth for feedforward neural networks. *arXiv preprint arXiv:1512.03965*, 2015.

[8] G. Freud. On direct and converse theorems in the theory of weighted polynomial approximation. *Mathematische Zeitschrift*, 126(2):123–134, 1972.

[9] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, Apr. 1980.

[10] J. T. Hastad. *Computational Limitations for Small Depth Circuits*. MIT Press, 1987.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385v1 [cs.CV] 10 Dec 2015*, 2015.

[12] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[13] H. N. Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural Computation*, 8(1):164–177, 1996.

[14] H. N. Mhaskar. On the degree of approximation in multivariate weighted approximation. In *Advanced Problems in Constructive Approximation*, pages 129–141. Springer, 2003.

[15] H. N. Mhaskar. When is approximation by Gaussian networks necessarily a linear process? *Neural Networks*, 17(7):989–1001, 2004.

[16] H. N. Mhaskar. A Markov-Bernstein inequality for Gaussian networks. In *Trends and applications in constructive approximation*, pages 165–180. Springer, 2005.

[17] H. N. Mhaskar. Weighted quadrature formulas and approximation by zonal function networks on the sphere. *Journal of Complexity*, 22(3):348–370, 2006.

[18] H. N. Mhaskar. Eignets for function approximation on manifolds. *Applied and Computational Harmonic Analysis*, 29(1):63–87, 2010.

[19] H. N. Mhaskar. Local approximation using Hermite functions. In *Progress in Approximation Theory and Applicable Complex Analysis – In the Memory of Q.I. Rahman*. Springer, To appear, arXiv:1608.01959.

[20] H. N. Mhaskar, Q. Liao, and T. Poggio. Learning real and boolean functions: When is deep better than shallow. *arXiv preprint arXiv:1603.00988, also Center for Brains, Minds and Machines (CBMM) Memo No. 45*, 2016.

[21] T. Poggio, F. Anselmi, and L. Rosasco. I-theory on depth vs width: hierarchical function composition. *CBMM memo 041*, 2015.

[22] T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society (AMS)*, 50(5):537–544, 2003.

[23] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, Nov. 1999.

[24] D. Ruderman. Origins of scaling in natural images. *Vision Res.*, pages 3385 – 3398, 1997.

[25] E. M. Stein. *Singular integrals and differentiability properties of functions (PMS-30)*, volume 30. Princeton university press, 2016.

[26] G. Szegö. Orthogonal polynomials. In *Colloquium publications/American mathematical society*, volume 23. Providence, 1975.

[27] M. Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101v2 [cs.LG] 29 Sep 2015*, 2015.