Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

When and Why Are Deep Networks Better than Shallow Ones?

Hrushikesh Mhaskar,[1,2] **Qianli Liao,**[3] **Tomaso Poggio**[3]
[1] Department of Mathematics, California Institute of Technology, Pasadena, CA, 91125
[2] Institute of Mathematical Sciences, Claremont Graduate University, Claremont, CA, 91711
[3] Center for Brains, Minds, and Machines, McGovern Institute for Brain Research
Massachusetts Institute of Technology, Cambridge, MA, 02139

Abstract

While the universal approximation property holds both for hierarchical and shallow networks, deep networks can approximate the class of compositional functions as well as shallow networks but with exponentially lower number of training parameters and sample complexity. Compositional functions are obtained as a hierarchy of local constituent functions, where "local functions" are functions with low dimensionality. This theorem proves an old conjecture by Bengio on the role of depth in networks, characterizing precisely the conditions under which it holds. It also suggests possible answers to the the puzzle of why high-dimensional deep networks trained on large training sets often do not seem to show overfit.

1 Introduction

Here are three main open questions about Deep Neural Networks. The first question is about the power of the architecture – which classes of functions can it approximate well? The second question is about generalization capability: why large deep networks trained with SGD often appear to be immune to overfitting? The third question is about learning the unknown coefficients from the data: why is SGD so unreasonably efficient, at least in appearance? Are good minima easier to find in deep rather than in shallow networks? In this paper we describe a set of approximation theory results that include an answer to why and when deep networks are better than shallow – the first question – and suggest a possible answer to the second one. We formulate our results by using the idealized model of a deep network as a binary tree. As we discuss here our main results described for the binary tree model also apply (apart from different constants) to the very deep convolutional networks of the ResNet type which have only a few stages of pooling and subsampling.

This paper compares shallow (one-hidden layer) networks with deep networks (see for example Figure 1). Both types of networks use the same small set of operations – dot products, linear combinations, a fixed nonlinear function of one variable, possibly convolution and pooling. The logic of the paper is as follows.

- Both shallow (a) and deep (b) networks are *universal*, that is they can approximate arbitrarily well any continuous function of d variables on a compact domain.
- We show that the approximation of functions with a *compositional structure* – such as $f(x_1, \cdots, x_d) = h_1(h_2 \cdots (h_j(h_{i1}(x_1, x_2), h_{i2}(x_3, x_4)), \cdots))$ – can be achieved with the same degree of accuracy by deep and shallow networks but that the number of parameters, the VC-dimension and the fat-shattering dimension are much smaller for the deep networks than for the shallow network with equivalent approximation accuracy. It is intuitive that a hierarchical network matching the structure of a compositional function should be "better" at approximating it than a generic shallow network but universality of shallow networks asks for the non-obvious characterization of "better". Our result makes clear that the intuition is indeed correct and provides a formal framework and quantitative bounds. From the point of view of machine learning shallow networks cannot exploit the prior of compositionality and the much smaller associated complexity. From the point of view of approximation theory see (Poggio et al. 2016) for references on lower bounds.
- The most interesting deep networks to which our results apply are the deep convolutional networks. Interestingly, the weight sharing property though helpful is not the key property: locality of the functions approximated at each stage is key.
- Why do compositional functions appear in so many problems in vision, text and speech? We argue that the basic properties of *locality at different scales* and *shift invariance* in many natural signals such as images and text implies that many (but not all) tasks on such inputs can be solved by compositional algorithms. Of course, there are many signals or problems that cannot be solved by shift invariant, scalable algorithms. Thus for the many problems that are not compositional we do not expect any advantage of deep convolutional networks.

2 Previous work

The success of Deep Learning poses again an old theory question: why are multi-layer networks better than one-hidden-layer networks? Under which conditions? The question is relevant in several related fields from machine learning to function approximation and has appeared many times before.

Most Deep Learning references these days start with

Hinton's backpropagation and with LeCun's convolutional networks (see for a nice review (LeCun, Bengio, and G. 2015)). Of course, multilayer convolutional networks have been around at least as far back as the optical processing era of the 70s. The Neocognitron (Fukushima 1980) was a convolutional neural network that was trained to recognize characters. The property of *compositionality* was a main motivation for hierarchical models of visual cortex such as HMAX which can be regarded as a pyramid of AND and OR layers (Riesenhuber and Poggio 1999), that is a sequence of conjunctions and disjunctions. There are several recent papers addressing the question of why hierarchies. Sum-Product networks, which are equivalent to polynomial networks (see (B. Moore and Poggio 1998; Livni, Shalev-Shwartz, and Shamir 2013)), are a simple case of a hierarchy that can be analyzed (Delalleau and Bengio 2011). (Montufar, Cho, and Bengio 2014) provided an estimation of the number of linear regions that a network with ReLU nonlinearities can synthesize in principle but leaves open the question of whether they can be used for learning and which conditions. Examples of functions that cannot be represented efficiently by shallow networks have been given very recently by (Telgarsky 2015). Most relevant to this paper is the work on hierarchical quadratic networks (Livni, Shalev-Shwartz, and Shamir 2013), together with function approximation results (Pinkus 1999; Mhaskar 1993). This paper extends and explains recent results that appeared in online publications (Poggio, Anselmi, and Rosasco 2015; Mhaskar, Liao, and Poggio 2016; Mhaskar and Poggio 2016; Poggio et al. 2015a).

3 Compositional functions

We assume that the shallow networks do not have any structural information on the function to be learned (here its hierarchically local structure), because they cannot represent it directly. Deep networks with standard architectures on the other hand *do represent* compositionality and can be adapted to such prior information. Thus, it is natural to conjecture that hierarchical compositions of functions such as

$$f(x_1, \cdots, x_8) = h_3(h_{21}(h_{11}(x_1, x_2), h_{12}(x_3, x_4)), \\ h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8))) \quad (1)$$

are approximated more efficiently by deep than by shallow networks. The structure of the function in equation 1 is a graph of the binary tree type, which is one of the simplest example of compositional functions (see later for even simpler cases of one-dimensional functions), reflecting dimensionality $d = 2$ for the constituent functions h. In general, d is arbitrary but fixed and independent of the dimensionality n of the compositional function f. In particular, d corresponds to the support of the convolutional kernel in deep convolutional networks.

In addition, both shallow and deep representations may or may not reflect invariance to group transformations of the inputs of the function ((Soatto 2011; Anselmi et al. 2015)). Invariance is expected to decrease the complexity of the network, for instance its VC-dimension. Since we are interested in the comparison of shallow vs deep architectures, here we consider the generic case of networks (and functions) for which invariance is not assumed. In the real world of applications the networks corresponding to our deep architectures are deep convolutional networks, in which there is locality at each level in the architecture and in addition shift invariance, that is weight sharing.

We approximate functions of n variables of the form of Equation (1) with networks in which the activation nonlinearity is a smoothed version of the so called ReLU, originally called *ramp* by Breiman and given by $\sigma(x) = x_+ = max(0, x)$. The architecture of the deep networks reflects Equation (1) with each node h_i being a ridge function, comprising one or more neurons.

It is *important to emphasize* that our results described in terms of binary trees apply to state-of-art Deep Learning Neural Networks (DLNNs), for instance of the ResNet type (He et al. 2015), with small kernel size and many layers. Visual cortex has a similar compositional architecture with receptive fields becoming larger and larger in higher and higher visual areas, with each area corresponding to a recurrent layer in a deep neural network (Liao and Poggio 2016).

4 Degree of approximation

In this section, we describe the approximation properties of the shallow and deep networks in the case of ReLU nonlinearities. Similar and even stronger results hold for deep Gaussian networks (Mhaskar, Liao, and Poggio 2016). The general paradigm is as follows. We are interested in determining how complex a network ought to be to **theoretically guarantee** approximation of an unknown target function f up to a given accuracy $\epsilon > 0$. To measure the accuracy, we need a norm $\| \cdot \|$ on some normed linear space $\mathbb{X}$. As we will see the norm used in the results of this paper is the sup norm. Let V_N be the be set of all networks of a given kind with complexity N which we take here to be the total number of units in the network (e.g., all shallow networks with N units in the hidden layer). It is assumed that the class of networks with a higher complexity include those with a lower complexity; i.e., $V_N \subseteq V_{N+1}$. The *degree of approximation* is defined by

$$\mathsf{dist}(f, V_N) = \inf_{P \in V_N} \|f - P\|. \quad (2)$$

For example, if $\mathsf{dist}(f, V_N) = \mathcal{O}(N^{-\gamma})$ for some $\gamma > 0$, then a network with complexity $N = \mathcal{O}(\epsilon^{-\frac{1}{\gamma}})$ will be sufficient to guarantee an approximation with accuracy at least ϵ. Since f is unknown, in order to obtain theoretically proved upper bounds, we need to make some assumptions on the class of functions from which the unknown target function is chosen. This a priori information is codified by the statement that $f \in W$ for some subspace $W \subseteq \mathbb{X}$. This subspace is usually a smoothness class characterized by a smoothness parameter r. Here it will be generalized to a smoothness and compositional class, characterized by the parameters r and d ($d = 2$ in the example of the deep network corresponding to the right hand side of Figure 1). In general, a deep network architecture (in this paper, we restrict ourselves to the binary tree structure as in (1)) has an advantage over the shallow

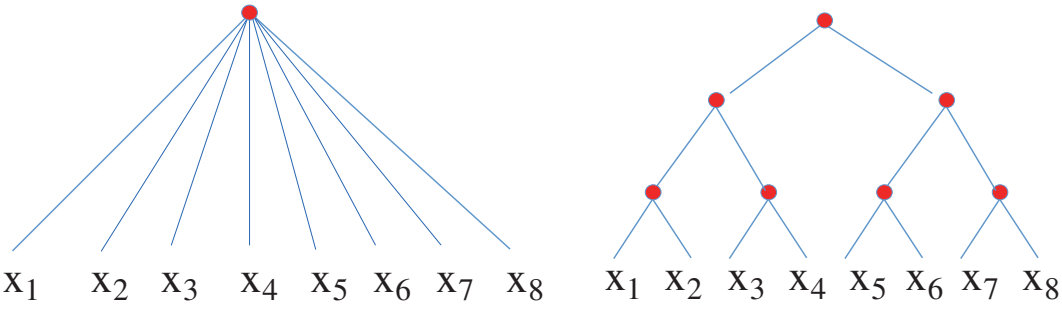

Figure 1: On the left a shallow universal network in 8 variables and N units which can approximate a generic function $f(x_1, \cdots, x_8)$. On the right, a binary tree hierarchical network in $n = 8$ variables, which approximates well functions of the form $f(x_1, \cdots, x_8) = h_3(h_{21}(h_{11}(x_1, x_2), h_{12}(x_3, x_4)), h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8)))$. Each of the $n - 1$ nodes consists of Q smoothed ReLU units with $Q(n - 1) = N$ and computes the ridge function (Pinkus 1999) $\sum_{i=1}^{Q} a_i(\langle \mathbf{v}_i, \mathbf{x} \rangle + t_i)_+$, with $\mathbf{v}_i, \mathbf{x} \in \mathbb{R}^2$, $a_i, t_i \in \mathbb{R}$. Each term, that is each unit in the node, corresponds to a "channel". In a binary tree with n inputs, there are $log_2 n$ levels and a total of $n - 1$ nodes. Similar to the shallow network, a hierarchical network can approximate any continuous function; the text proves how it approximates a compositional functions better than a shallow network. No invariance is assumed here.

networks when the target function itself has the same hierarchical, compositional structure, e.g., (1) or, as we will see later, a structure that is part of the network structure.

4.1 Main results

Let $I^n = [-1, 1]^n$, $\mathbb{X} = C(I^n)$ be the space of all continuous functions on I^n, with $\|f\| = \max_{\mathbf{x} \in I^n} |f(\mathbf{x})|$. Let $\mathcal{S}_{N,n}$ denote the class of all shallow networks with N units and n inputs of the form

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

where $\mathbf{w}_k \in \mathbb{R}^n$, $b_k, a_k \in \mathbb{R}$. The number of trainable parameters here is $(n + 2)N \sim N$. Let $r \geq 1$ be an integer, and W_r^n be the set of all functions of n variables 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 the hierarchical binary tree network, the analogous spaces are defined by considering the compact set $W_r^{n,2}$ to be the class of all compositional functions f of n variables with a binary tree architecture and constituent functions h in W_r^2. We define the corresponding class of deep networks $\mathcal{D}_{N,2}$ to be the set of all deep networks with N units and a binary tree architecture, where each of the $n - 1$ constituent nodes contains Q units (with $(n - 1)Q = N$ is in $\mathcal{S}_{Q,2}$. We note that in the case when n is an integer power of 2, the total number of parameters involved in a deep network in $\mathcal{D}_{N,2}$ – that is, weights and biases – is $4N \sim N$.

The following two theorems estimate the degree of approximation for shallow and deep networks. Two observations are critical to understand the meaning of our results:

- *compositional functions of n variables are a subset of functions of n variables, that is $W_r^n \supset W_r^{n,2}$.* Deep networks can exploit in their architecture the special structure of compositional functions, whereas shallow networks are blind to it. Thus from the point of view of shallow networks, functions in $W_r^{n,2}$ are just functions in W_r^n; this is not the case for deep networks.
- *a deep network does not need to have exactly the same compositional architecture as the compositional function to be approximated*. It is sufficient that the acyclic graph representing the structure of the function (see (Mhaskar and Poggio 2016)) is *a subgraph of the graph representing the structure of the deep network*. The degree of approximation estimates depend on the graph associated with the network and are an upper bound on what could be achieved by a network exactly matched to the function architecture.

The first theorem is about shallow networks and is a classical result.

Theorem 1. *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be infinitely differentiable, and not a polynomial on any subinterval of $\mathbb{R}$.*
For $f \in W_r^n$ the complexity of shallow networks that provide accuracy at least ϵ is

$$N = \mathcal{O}(\epsilon^{-n/r}) \text{ and is the best possible.} \qquad (3)$$

Notes The proof in (Mhaskar 1996) relies on the fact that when σ satisfies the conditions of the theorem, the algebraic polynomials in n variables of (total or coordinatewise) degree $< q$ are in the uniform closure of the span of $\mathcal{O}(q^n)$ functions of the form $\mathbf{x} \mapsto \sigma(\mathbf{w} \cdot \mathbf{x} + b)$. The estimate itself is an upper bound on the degree of approximation by such polynomials. Since it is based on the approximation of the polynomial space contained in the ridge functions implemented by shallow networks, one may ask whether it could be improved by using a different approach. The answer relies on the concept of nonlinear n–width of the compact set W_r^n (cf. (DeVore, Howard, and Micchelli 1989; Mhaskar, Liao, and Poggio 2016)). The n-width results im-

ply that the estimate in Theorem (1) is *the best possible* among **all** reasonable (DeVore, Howard, and Micchelli 1989) methods of approximating arbitrary functions in W_r^n. □

The second theorem is about deep networks and is new (preliminary versions appeared in (Poggio, Anselmi, and Rosasco 2015; Poggio et al. 2015b; Mhaskar, Liao, and Poggio 2016)). We formulate it in the binary tree case for simplicity but it extends immediately to functions that are compositions of functions of d variables instead than of 2 variables as in the theorem.

Theorem 2. *For $f \in W_r^{n,2}$ the complexity of a deep network (with the same compositional architecture) that provide approximation with accuracy at least ϵ is*

$$N = \mathcal{O}((n-1)\epsilon^{-2/r}). \tag{4}$$

Proof To prove Theorem 2, we observe that each of the constituent functions being in W_r^2, (1) applied with $n = 2$ implies that each of these functions can be approximated from $\mathcal{S}_{N,2}$ up to accuracy $\epsilon = cN^{-r/2}$. Our assumption that $f \in W_r^{N,2}$ implies that each of these constituent functions is Lipschitz continuous. Hence, one infers 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 since $\|h - P\| \leq \epsilon$, $\|h_1 - P_1\| \leq \epsilon$ and $\|h_2 - P_2\| \leq \epsilon$, then $\|h(h_1, h_2) - P(P_1, P_2)\| = \|h(h_1, h_2) - h(P_1, P_2) + h(P_1, P_2) - P(P_1, P_2)\| \leq \|h(h_1, h_2) - h(P_1, P_2)\| + \|h(P_1, P_2) - P(P_1, P_2)\| \leq c\epsilon$ by Minkowski inequality. Thus

$$\|h(h_1, h_2) - P(P_1, P_2)\| \leq c\epsilon,$$

for some constant $c > 0$ independent of the functions involved. This, together with the fact that there are $(n-1)$ nodes, leads to (4). □

Remarks

1. In the statement of the theorem we assume that the dimensionality of the functions h, that together compose f, is fixed (in the theorem equal to 2 and more in general to d, whereas the dimensionality n of f can increase and with it the depth and the number of nodes of the graph associated with f.
2. The constants involved in $\mathcal{O}$ in the theorems 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 $\mathcal{O}(\epsilon^{-n/r})$ trainable parameters. If we assume a hierarchical structure on the target function as in Theorem 2, then the corresponding deep network yields a guaranteed accuracy of ϵ with $\mathcal{O}(\epsilon^{-2/r})$ trainable parameters.
3. Theorem 2 applies to all f with a compositional architecture given by a graph which correspond to, or is a subgraph of, the graph associated with $W_r^{n,d}$.
4. The assumptions on σ in the theorems 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, which will not change any of the empirical results testing deep networks reported in the literature. This strongly suggests that the result of the theorem should be valid also for the non-smooth ReLU. In fact, such a proof, technically more involved, is now available (Mhaskar and Poggio 2016) (see also (Poggio et al. 2016)).
5. Similar – actually stronger – results (see (Mhaskar and Poggio 2016)) hold for networks where each channel evaluates a Gaussian non–linearity; i.e., Gaussian networks of the form

$$G(\mathbf{x}) = \sum_{k=1}^{N} a_k \exp(-|\mathbf{x} - \mathbf{x}_k|^2), \qquad \mathbf{x} \in \mathbb{R}^d \tag{5}$$

   where the approximation is on the entire Euclidean space.
6. The estimates on the n–width imply that there is some function in either W_r^n (theorem 1) or $W_r^{n,2}$ (theorem 2) for which the approximation cannot be better than that suggested by the theorems above. This is of course the guarantee we want but it would also be interesting to know whether these functions are somewhat pathological. In the case of Gaussian networks, it is proved in (Mhaskar and Poggio 2016) that even for individual functions, it is not possible to achieve the degree of approximation unless the function is smooth as indicated by the above theorems.
7. Compositional functions with an associated binary tree graph are the simplest example of *hierarchically local compositional functions*(Poggio et al. 2016). They are also a good model for deep convolutional networks, for example of the ResNet type. According to our theory, weight sharing is not their key advantage but hierarchical locality is. In fact simple ConvNets with and without weight sharing perform similarly on CIFAR-10 (see (Poggio et al. 2016)).
8. *Remark: Function Composition* There are compositional functions of n variables such as $f(x) = h_3(h_2(h_1(x)))$, where $f : \mathbb{R}^n \mapsto \mathbb{R}^m$, $h_3 : \mathbb{R}^q \mapsto \mathbb{R}^m$, $h_2 : \mathbb{R}^d \mapsto \mathbb{R}^q$, $h_1 : \mathbb{R}^n \mapsto \mathbb{R}^d$, that may be approximated more efficiently by deep than shallow networks. The intuition is that sometime the constituent functions can be approximated by simpler polynomials (either degree or number of variables or both) than the full function. For instance, in the case of one variable consider that the proof of the theorems in Section 4.1 implies that a hierarchical network can approximate more efficiently than a shallow network a high degree polynomial P in the input variables $x_1, \cdots, x_d$, that can be written as a hierarchical composition of lower degree polynomials. For example, let

$$P(x, y) = (Ax^2y^2 + Bx^2y + I)^{2^{10}}.$$

   Since P is nominally a polynomial of coordinatewise degree 2^{11}, (Mhaskar 1996, Lemma 3.2) shows that a shallow network with $2^{11} + 1$ units is able to approximate P arbitrarily well on I^d. However, because of the hierarchical structure of P, (Mhaskar 1996, Lemma 3.2) shows also that a hierarchical network with 9 units is more than sufficient to approximate the quadratic expression, and 10

further layers, each with 3 units can approximate the successive powers. Thus, a hierarchical network with 11 layers and 39 units can approximate P arbitrarily well. We note that even if P is nominally of degree 2^{11}, each of the monomial coefficients in P is a function of only 9 variables, $A, \cdots, I$. A much simpler example was tested using standard DLNN software and is shown in Figure 2. In (Poggio et al. 2016) we extend Theorem 4 to deal with the general compositional case and, in particular, with the example above.

On the other hand it is easy to construct counterexamples where the composition of relatively simple functions h is more complex than the full function f, in which case the shallow network will be more efficient than the deep one in learning and approximating it. Compositionality by itself – defined just as a hierarchy of functions – cannot avoid the curse of dimensionality but locality in the hierarchy does as our theorems show. Thus : *hierarchical locality is sufficient to avoid the curse of dimensionality*.

4.2 Generalization bounds

Our estimate of the number of units and parameters needed for a deep network to approximate compositional functions with an error ϵ_G allow the use of one of several available bounds for the generalization error of the network. For instance theorem 16.2 in (Anthony and Bartlett 2002) provides the following sample bound for a generalization error ϵ_G with probability at least $1 - \delta$ in a network in which the W parameters (weights and biases) are expressed in terms of k bits:

$$m(\epsilon_G, \delta) \leq \frac{2}{\epsilon_G^2}(kW \log 2 + \log(\frac{2}{\delta})) \tag{6}$$

This suggests the following comparison between shallow and deep compositional (here binary tree-like networks). Assume networks size that ensure the same training error ϵ . Then in order to achieve the same generalization error ϵ_G, the sample size $m_{shallow}$ of the shallow network must be much larger than the sample size m_{deep} of the deep network:

$$\frac{m_{deep}}{m_{shallow}} \approx \epsilon^n. \tag{7}$$

This implies that for large n there is a (large) range of training set sizes between m_{deep} and $m_{shallow}$ for which deep networks will not overfit (corresponding to small ϵ_G) but shallow networks will (for dimensionality $n \approx 10^4$ and $\epsilon \approx 0.1$ Equation 7 yields $m_{shallow} \approx 10^{10^4} m_{deep}$). This observation holds under the assumption that the optimization process during training finds optimum values of the parameters for both deep and shallow networks. For stochastic gradient descent – which, depending on adaptive step size, may increase with iteration number the number of effective parameters $W_{effective}$ towards W while maintaining a well conditioned behavior (L. Rosasco, pers. com.) – the situation is likely to be more complicated but the overall conclusion about relative overfitting for deep vs. shallow networks should still hold true.

5 Discussion

- The simplest compositional function – addition – is trivial in the sense that it offers no approximation advantage to deep networks. A key function is multiplication which is the prototypical compositional functions. It is not an accident that in the case of Boolean variables the parity function $f(x_1, ...x_n) = x_1 \cdots x_n$ is at the core of a classical result on Boolean circuits (Hastad 1987).

- It has been often argued that not only text and speech are compositional but so are images. There are many phenomena in nature that have descriptions along a range of rather different scales. An extreme case consists of fractals which are infinitely self-similar, iterated mathematical constructs. As a reminder, a self-similar object is similar to a part of itself (i.e. the whole is similar to one or more of the parts). Many objects in the real world are statistically self-similar, showing the same statistical properties at many scales: clouds, river networks, snow flakes, crystals and neurons branching. A relevant point is that the shift-invariant scalability of image statistics follows from the fact that objects contain smaller clusters of similar surfaces in a selfsimilar fractal way. Ruderman (Ruderman 1997) analysis shows that image statistics reflects what has been known as the property of compositionality of objects and parts: parts are themselves objects, that is selfsimilar clusters of similar surfaces in the physical world. Notice however that, from the point of view of this paper, it is misleading to say that an image is compositional: in our terminology a *function on an image may be compositional but not its argument*. In fact, functions to be learned may or may not be compositional even if their input is an image since they depend on the input but also on the task (for instance in the supervised case of deep learning networks all weights depend on x and y). Conversely, a network may be given a function which can be written in a compositional form, independently of the nature of the input vector such as the function "multiplication of all scalar inputs' components". Thus a more reasonable statement is that "many natural questions on images correspond to algorithms which are compositional". Why this is the case is an interesting open question. A possible answer is inspired by our theorems and by the empirical success of deep convolutional networks. It seems likely that in the natural sciences– physics, chemistry, biology – many phenomena may be described by processes that *that take place at a sequence of increasing scales and are local at each scale in the sense that they can be described well by neighbor-to-neighbor interactions*.

  Notice that this is a much less stringent requirement than renormalizable physical processes (Lin 2016) where the *same* Hamiltonian (apart from a scale factor) is required to describe the process at each scale. Tegmark and Lin (Lin 2016) have also suggested that a sequence of generative processes can be regarded as a Markov sequence that can be inverted to provide an inference problem with a similar compositional structure. The resulting compositionality they describe does not, however, correspond to our notion of *hierarchical locality* and thus our theorems

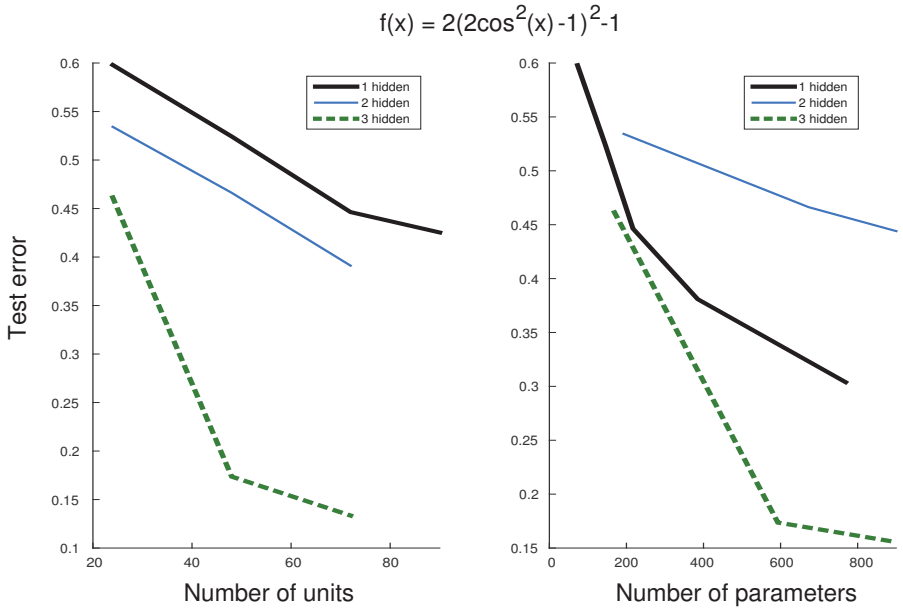

Figure 2: A sparse (because it has only a few of the possible terms) trigonometric polynomial $f(x) = 2(2\cos^2(x) - 1)^2 - 1$ (shown on the top of the figure) with one input variable is learned in a regression set-up using standard deep networks with 1, 2 or 3 hidden layers. In the 1 hidden layer setting, 24, 48, 72, 128 and 256 hidden units were tried. With 2 hidden layers, 12, 24 and 36 units per layer were tried. With 3 hidden layers, 8, 16 and 24 units per layer were tried. Each of the above settings was repeated 5 times, reporting the lowest test error. Mean squared error (MSE) was used as the objective function; the y axes in the above figures are the square root of the testing MSE. For the experiments with 2 and 3 hidden layers, batch normalization (Ioffe and Szegedy 2015) was used between every two hidden layers. 60k training and 60k testing samples were drawn from a uniform distribution over $[-2\pi, 2\pi]$. The training process consisted of 2000 passes through the entire training data with mini batches of size 3000. Stochastic gradient descent with momentum 0.9 and learning rate 0.0001 was used. Implementations were based on MatConvNet (Vedaldi and Lenc 2015). Same data points are plotted in 2 sub-figures with x axes being number of units and parameters, respectively. Note that with the input being 1-D, the number of parameters of a shallow network scales slowly with respect to the number of units, giving a shallow network some advantages in the right sub-figure. Although not shown here, the training errors are very similar to those of testing. The advantage of deep networks is expected to increase with increasing dimensionality of the function. Even in this simple case the solution found by SGD are almost certain to be suboptimal. Thus the figure cannot be taken as fully reflecting the theoretical results of this paper.

cannot be used to support their claims. As discussed previously (Poggio, Anselmi, and Rosasco 2015) hierarchical locality may be related to properties of basic physics that imply local interactions *at each level in a sequence of scales*, possibly different at each level. To complete the argument one would have then to assume that several different questions on sets of natural images may share some of the initial inference steps (first layers in the associated deep network) and thus share some of features computed by intermediate layers of a deep network. In any case, at least two open questions remain that require formal theoretical results in order to explain the connection between hierarchical, local functions and physics:

- can hierarchical locality be derived from the Hamiltonians of physics? In other words, under which conditions does coarse graining lead to local Hamiltonians?
- is it possible to formalize how and when the local hierarchical structure of computations on images is related to the hierarchy of local physical process that describe the physical world represented in the image?

- We recall that several properties of certain Boolean functions can be "read out" from the terms of their Fourier expansion corresponding to "large" coefficients, that is from a polynomial that represents the function (see (Poggio, Anselmi, and Rosasco 2015)).

Classical results (Hastad 1987) about the depth-breadth tradeoff in circuits design show that deep circuits are more efficient in representing certain Boolean functions than shallow circuits. These results have been often quoted in support of the claim that deep neural networks can represent functions that shallow networks cannot. For instance (Bengio and LeCun 2007) write *"We claim that most functions that can be represented compactly by deep architectures cannot be represented by a compact shallow architecture"*. The results reported here should settle the issue, justifying the original conjecture by restricting it to a class of functions and providing an approach connecting results on Boolean functions with real valued neural networks.

6 Acknowledgements

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

References

Anselmi, F.; Leibo, J. Z.; Rosasco, L.; Mutch, J.; Tacchetti, A.; and Poggio, T. 2015. Unsupervised learning of invariant representations. *Theoretical Computer Science*.

Anthony, M., and Bartlett, P. 2002. *Neural Network Learning - Theoretical Foundations*. Cambridge University Press.

B. Moore, B., and Poggio, T. 1998. Representations properties of multilayer feedforward networks. *Abstracts of the First annual INNS meeting* 320:502.

Bengio, Y., and LeCun, Y. 2007. Scaling learning algorithms towards ai. In Bottou, L.; Chapelle, O.; and DeCoste, D.and Weston, J., eds., *Large-Scale Kernel Machines*. MIT Press.

Delalleau, O., and Bengio, Y. 2011. Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, 666–674.

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

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

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

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

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

LeCun, Y.; Bengio, Y.; and G., H. 2015. Deep learning. *Nature* 436–444.

Liao, Q., and Poggio, T. 2016. Bridging the gap between residual learning, recurrent neural networks and visual cortex. *Center for Brains, Minds and Machines (CBMM) Memo No. 47, also in arXiv*.

Lin, H.and Tegmark, M. 2016. Why does deep and cheap learning work so well? *arXiv:1608.08225* 1–14.

Livni, R.; Shalev-Shwartz, S.; and Shamir, O. 2013. A provably efficient algorithm for training deep networks. *CoRR* abs/1304.7045.

Mhaskar, H., and Poggio, T. 2016. Deep versus shallow networks: an approximation theory perspective. *Center for Brains, Minds and Machines (CBMM) Memo No. 54, arXiv, to appear in Analysis and Applications*.

Mhaskar, H.; Liao, Q.; and Poggio, T. 2016. Learning real and boolean functions: When is deep better than shallow? *Center for Brains, Minds and Machines (CBMM) Memo No. 45, also in arXiv*.

Mhaskar, H. N. 1993. Neural networks for localized approximation of real functions. In *Neural Networks for Processing [1993] III. Proceedings of the 1993 IEEE-SP Workshop*, 190–196. IEEE.

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

Montufar, G. F.and Pascanu, R.; Cho, K.; and Bengio, Y. 2014. On the number of linear regions of deep neural networks. *Advances in Neural Information Processing Systems* 27:2924–2932.

Pinkus, A. 1999. Approximation theory of the mlp model in neural networks. *Acta Numerica* 8:143–195.

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

Poggio, T.; Rosaco, L.; Shashua, A.; Cohen, N.; and Anselmi, F. 2015a. Notes on hierarchical splines, dclns and i-theory. *CBMM memo 037*.

Poggio, T.; Rosasco, L.; Shashua, A.; Cohen, N.; and Anselmi, F. 2015b. Notes on hierarchical splines, dclns and i-theory. Technical report, MIT Computer Science and Artificial Intelligence Laboratory.

Poggio, T.; Mhaskar, H.; Rosasco, L.; Miranda, B.; and Liao, Q. 2016. Why and when can deep–but not shallow–networks avoid the curse of dimensionality. *Center for Brains, Minds and Machines (CBMM) Memo No. 58, arXiv preprint arXiv:1611.00740*.

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

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

Soatto, S. 2011. Steps Towards a Theory of Visual Information: Active Perception, Signal-to-Symbol Conversion and the Interplay Between Sensing and Control. *arXiv:1110.2053* 0–151.

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

Vedaldi, A., and Lenc, K. 2015. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd Annual ACM Conference on Multimedia Conference*, 689–692. ACM.