CENTER FOR
Brains
Minds+
Machines

CBMM Memo No. 109 **June 24, 2020**

Hierarchically Local Tasks and Deep Convolutional Networks

Arturo Deza, Qianli Liao, Andrzej Banburski, Tomaso Poggio
Center for Brains, Minds, and Machines
Massachusetts Institute of Technology

Abstract

The main success stories of deep learning, starting with ImageNet, depend on convolutional networks, which on certain tasks perform significantly better than traditional shallow classifiers, such as support vector machines. Is there something special about deep convolutional networks that other learning machines do not possess? Recent results in approximation theory have shown that there is an exponential advantage of deep convolutional-like networks in approximating functions with hierarchical locality in their compositional structure. These mathematical results, however, do not say which tasks are expected to have input-output functions with hierarchical locality. Among all the possible hierarchically local tasks in vision, text and speech we explore a few of them experimentally by studying how they are affected by disrupting locality in the input images. We also discuss a taxonomy of tasks ranging from local, to hierarchically local, to global and make predictions about the type of networks required to perform efficiently on these different types of tasks.

This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.

Hierarchically Local Tasks and Deep Convolutional Networks

Arturo Deza, Qianli Liao, Andrzej Banburski, Tomaso Poggio

Center for Brains, Minds and Machines
Massachusetts Institute of Technology
{deza,lql,kappa666}@mit.edu;tp@ai.mit.edu

Abstract

The main success stories of deep learning, starting with ImageNet, depend on convolutional networks, which on certain tasks perform significantly better than traditional shallow classifiers, such as support vector machines. Is there something special about deep convolutional networks that other learning machines do not possess? Recent results in approximation theory have shown that there is an exponential advantage of deep convolutional-like networks in approximating functions with hierarchical locality in their compositional structure. These mathematical results, however, do not say which tasks are expected to have input-output functions with hierarchical locality. Among all the possible hierarchically local tasks in vision, text and speech we explore a few of them experimentally by studying how they are affected by disrupting locality in the input images. We also discuss a taxonomy of tasks ranging from local, to hierarchically local, to global and make predictions about the type of networks required to perform efficiently on these different types of tasks.

1 Introduction

In 1967 the book Perceptrons attempted to characterize local and global computations [23]. The attempt failed despite the lovely theorems (*recognition of a geometric pattern invariant to translation is order 3 – thus very local*; *recognition of connectedness is order infinity – thus global*,...) because local and global was defined in terms of computational machines that were too simple – roughly similar to one-hidden layer networks. With the advent of deep networks, it is now interesting to consider again the approach of Minsky & Papert [23] – a characterization of how *local* are various computational tasks and which network architectures are needed to solve them – and aim at a considerable upgrade of such definitions. The plan of the paper is as follows. We will first recall and extend recent results on approximation of a certain class of compositional functions with local constituent functions. For this class of functions, shallow networks suffer from the curse of dimensionality whereas deep networks of the convolutional type (possibly without weight sharing) avoid this curse. Second we will provide examples of computational tasks that correspond to such compositional functions. We will empirically show that the property of hierachical local compositionality depends on the inputs *and* the task. We will conclude by showing experiments that test whether stochastic gradient descent (SGD) can learn the convolutional architecture required by the task when the network is initially a superset of a convolutional network.

Preprint. Under review.

2 There are functions that can be approximated well by deep networks but not by shallow networks

The main question in this section is about the difference between deep networks and shallow ones, such as kernel machines: are deep networks more powerful in terms of approximating functions?

Several papers in the '80s focused on the approximation power and learning properties of one-hidden layer networks (called shallow networks here). It was clear then that one-layer networks can approximate arbitrarily well any continuous function on a compact domain. As a consequence, little appeared on multilayer networks, (but see [16, 21, 3, 4, 27]). By now, however, there are several papers which have studied the approximation properties of multilayer RELU networks. Even for univariate functions it seems that RELU networks have some advantages with respect to classical approximation scheme say by polynomials [5]. Much interest has focused on separation results, similar to classical results on Boolean circuits, that is on the fact that specific functions that can be represented well by deep networks cannot be represented efficiently by shallow networks. The best known results of this type is due to Telgarsky [37] (see also [33, 30, 24, 15, 26]) for univariate functions (the first separation results were older and apparently little known [4]). Telgarsky proves an exponential gap between certain functions produced by deep networks and their approximation by shallow networks. The theorem [37] can be summarized as saying that *a certain family of classification problems with real-valued inputs cannot be approximated well by shallow networks with fewer than exponentially many nodes whereas a deep network achieves zero error*.

The intuition behind these results is that high frequency functions can be represented much more easily by deep networks – that is with fewer units – than by shallow networks. High frequencies correspond to higher degree polynomials. This also is the case for Boolean functions where the low order coefficients of the Fourier representation of a Boolean function correspond to a polynomial of low degree. Hastad [10] proved that highly-variable functions (in the sense of having high frequencies in their Fourier spectrum), in particular the parity function, cannot even be decently approximated by small constant depth Boolean circuits (see also [14]).

2.1 When can deep networks avoid the curse of dimensionality?

The separation results described above are of course important but they do not say more than there exist functions that can be better approximated by deep networks. A result in the same spirit but with more intriguing implications focuses on multivariate approximation. Recall that the greatest success of deep (convolutional) networks have been in dealing with high-dimensional objects such as images (in ImageNet [32] each image is in the order of 10^5 pixels). We begin with the observation [1, 29, 28] that, though both shallow and deep networks are universal approximators, and both suffer from the *curse of dimensionality*: the number of parameters they need to approximate a generic continuous function in d dimensions with an error ε may require a number of parameters in the order of ε^{-d}. Starting from this fact, we asked whether there exist specific classes of functions for which deep networks may be able to avoid the exponential curse and have a non-exponential upper bound. The answer is positive [17]. Deep networks can avoid the curse of dimensionality for functions that are compositions of functions with a small dimensionality: we will call this class of functions *hierarchically compositional*. An example is shown in Figure 1.

2.2 Shallow and deep networks: theorems

The general paradigm is as follows. We are interested in determining how complex a network, denoted as a function $f(x)$ ought to be to *theoretically guarantee* approximation of an unknown target function g up to a given accuracy $\varepsilon > 0$. In particular, this section characterizes conditions under which deep networks are "better" than shallow network in approximating functions. 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. Each node in the networks corresponds to a node in the graph of the function to be approximated, as shown in Figure 1.

We define a deep network with K layers with the usual coordinate-wise scalar activation functions $\sigma(z): \quad \mathbf{R} \to \mathbf{R}$ as the set of functions $f(W;x) = \sigma(W^K \sigma(W^{K-1} \cdots \sigma(W^1 x)))$, where the input is $x \in \mathbf{R}^d$, the weights are given by the matrices W^k, one per layer, with matching dimensions. We sometime use the symbol W as a shorthand for the set of W^k matrices $k = 1, \cdots, K$. There are no

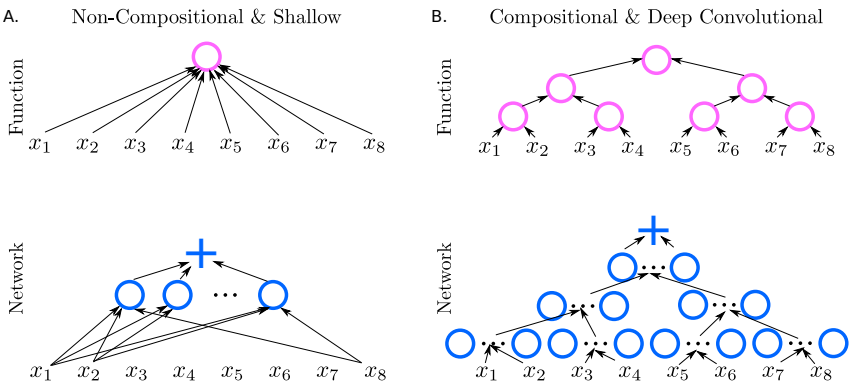

Figure 1: The top graphs are associated to *functions*; each of the bottom diagrams depicts the ideal *network* approximating the function above. In A. a shallow universal network (bottom) in 8 variables and N units approximates a generic function (top) of 8 variables $g(x_1,\cdots,x_8)$. Inset B. shows a hierarchical network at the bottom in $n=8$ variables, which approximates well functions of the form $g(x_1,\cdots,x_8)=h_3(h_{21}(h_{11}(x_1,x_2),h_{12}(x_3,x_4)),h_{22}(h_{13}(x_5,x_6),h_{14}(x_7,x_8)))$ as represented by the binary graph above. In the approximating network each of the $n-1$ nodes in the graph of the function corresponds to a set of $Q=\frac{N}{n-1}$ ReLU units computing the ridge function $\sum_{i=1}^{Q} a_i(\langle \mathbf{v}_i,\mathbf{x}\rangle + t_i)_+$, with $\mathbf{v}_i,\mathbf{x}\in\mathbb{R}^2$, $a_i,t_i\in\mathbb{R}$. Each term in the ridge function corresponds to a unit in the node (this is somewhat different from today's deep networks, but equivalent to them [30]). Similar to the shallow network, a hierarchical network is universal, that is, it can approximate any continuous function; the text proves that it can approximate a compositional functions exponentially better than a shallow network. Redrawn from Mhaskar & Poggio [18].

bias terms: the bias is instantiated in the input layer by one of the input dimensions being a constant. A shallow network is a deep network with $K=1$.

We approximate functions with networks in which the activation nonlinearity is a ReLU, given by $\sigma(x)=x_+=max(0,x)$. The architecture of the deep networks reflects the function graph, with each node h_i being a ridge function, comprising one or more neurons.

Notice that in the main example (see Figure 1) of a network corresponding to a function with a binary tree graph, the resulting architecture is an idealized version of deep convolutional neural networks described in the literature. In particular, it has only one output at the top unlike most of the deep architectures with many channels and many top-level outputs. Correspondingly, each node computes a single value instead of multiple channels, using the combination of several units. However the result holds also for these more complex networks (see Corollary 4 [30]).

The sequence of results is as follows.

- *Both shallow (A.) and deep (B.) networks are universal*, that is they can approximate arbitrarily well any continuous function of n variables on a compact domain. The result for shallow networks is classical.
- We consider a special class of functions of n variables on a compact domain that are *hierarchical compositions of local functions*, such as $g(x_1,\cdots,x_8)=h_3(h_{21}(h_{11}(x_1,x_2),h_{12}(x_3,x_4)),h_{22}(h_{13}(x_5,x_6),h_{14}(x_7,x_8)))$.
  The structure of the function in Figure 1 (B.) is represented by a graph of the binary tree type, reflecting dimensionality $d=2$ for the constituent functions h. In general, d is arbitrary but fixed and independent of the dimensionality n of the function g. Poggio et al. [30] formalizes the more general compositional case using directed acyclic graphs.
- The approximation of functions with such a specific compositional structure can be achieved with the same degree of accuracy by deep and shallow networks but the number of parameters is in general much lower for the deep networks than for the shallow network for the same approximation accuracy.

The two main theorems (see [31]) are one about shallow networks (originally due to [22]) and the second one about deep networks with smooth activation function (see also [28, 29, 19]). They can be

condensed in the following informal statement, where complexity is measured in terms of number N of units:

Theorem 1 *For a continuous target function $g(x)$ with $x \in \mathbb{R}^n$ the complexity of a shallow network that provide accuracy at least ε is $O(\varepsilon^{-n})$. If $g(x)$ has a graph representation in terms of a binary tree, consider a deep network with the same compositional architecture and with an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ which is infinitely differentiable, and not a polynomial. Then the complexity of the network to provide approximation with accuracy at least ε is $O((n-1)\varepsilon^{-2})$.*

The exponential dependence of the number of parameters required for an accuracy $O(\varepsilon)$ on the dimension n is known as the *curse of dimensionality*. The precise statement of the result in [31] shows that the curse of dimensionality can be reduced by smoothness of the target functions. Of course, the binary tree case of the theorem is the simplest case: the extension to more complex trees is obvious.

The result of theorem 1 can be extended to non-smooth ReLU for the case of deep networks in a number of ways, one of which, suggested by Mhaskar [20], is as follows. The first step is to recall Theorem 3.2 in [20] that applies to smooth activation functions such as the "square" of the ReLU, that is the function x_+^2 defined as $= x^2$ if $x \geq 0$ and otherwise $= 0$. Loosely speaking, theorem 3.2 implies that for such activation functions deep networks with a sufficient number of neurons and layers can approximate continuous functions as well as free knots splines. The second step is to use Theorem 1 from Yarotsky [41] on approximating $(x_+)^2$ with ReLU networks.

The intuition of why hierarchical compositional functions can be approximated by deep networks without incurring in the curse of dimensionality can be best explained in the following way.

Consider the space of polynomials, which are of course a smaller space than spaces of Sobolev functions, but which can approximate arbitrarily well continuous functions. Let us call P_k^n the linear space of polynomials of degree at most k in n variables and T_k^n the subset of the space P_k^n which consists of compositional polynomials with a binary tree graph and constituent polynomial functions of degree k (in 2 variables). Then the following theorems hold (see Poggio et al. [31]):

Theorem 2 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be infinitely differentiable, and not a polynomial. Every $g \in P_k^n$ can be realized with an arbitrary accuracy by shallow network with r units, $r = \binom{n+k}{k} \approx k^n$.*

and

Theorem 3 *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be infinitely differentiable, and not a polynomial. Let $n = 2^\ell$. Then $f \in T_k^n$ can be realized by a deep network with a binary tree graph and a total of r units with $r = (n-1)\binom{2+k}{2} \approx (n-1)k^2$.*

The theorems make clear that a hierarchical compositional polynomial corresponding to a binary graph is a very sparse polynomial – among the polynomials of the same degree in the same number of variables. It is possible to show from [31]) that this intuition extends from polynomials to Sobolev functions. Of course the target function graph at the top right of Figure 1 is just one of the many Directed Acyclic Graphs (DAGs, see [18]) that can describe different forms of compositionality. Prior knowledge of the DAG underlying a learning problem can be exploited to a great advantage by an appropriate deep learning architecture.

The theorems above are upper bounds but it is obvious that *shallow networks cannot exploit the prior information about the compositionality of a function in their architecture*. In any case, we claim here that *a formal lower bound* can be given by using Telgarsky result on a function such as the polynomial

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

The claim is that this function cannot be approximated by shallow networks with the same number of parameters as deep networks. The proof relies on the observation that Q_1^{1024} is a function of one variable to which Telgarski theorem can be applied.

2.3 The puzzle of the unreasonable effectiveness of CNNs

We think that the results discussed above are especially interesting because they represent a potential explanation for one of the greatest puzzles that has emerged from the empirical field of deep learn-

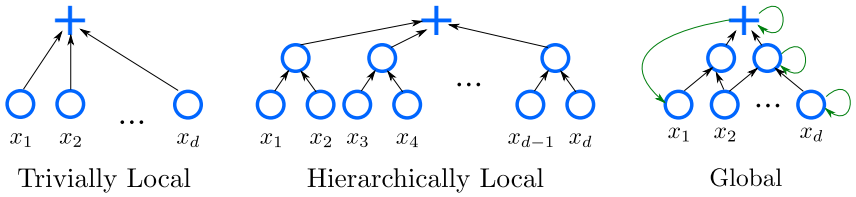

Figure 2: Network schemes for trivially local tasks, for hierachically local and for global tasks. Tasks such as color estimation are trivially local as they do not require structure, while other tasks like object classification and scene gist are hierarchically local due to the composite nature of their images. Global tasks have no hierarchical structure and can *not* be solved by classical feed-forward networks and require more complex operations such as feedback and recursion (van Bergen & Kriegeskorte [39]) – an example of such task would be image insideness as suggested in Villalobos et al. [40].

ing, that is the *unreasonable effectiveness* of convolutional deep networks in a number of sensory problems. Convolutional networks are a special case of the hierarchical networks of the theorem once weight sharing is added to the locality of the constituent functions.

In summary, *deep convolutional architectures* have the theoretical guarantee that they can be *much better* than one hidden layer architectures such as kernel machines for certain classes of problems. These problems correspond to input-output mappings that are *compositional with local constituent functions*. Quite interestingly, and not well known, is that the key aspect of convolutional networks that can give them an exponential advantage is *not weight sharing* but *locality* of the constituent functions.

Weight sharing decreases the complexity of the network but not exponentially so. It is the locality of the constituent functions that avoids the curse of dimensionality. This means that problems which do not admit a translation invariant prior, that corresponds to weight sharing, can benefit greatly by the use of hierarchically local networks (HLNs) which are convolutional networks without weight sharing.

2.4 Hierarchically compositional functions and computational tasks

Thus the important question is: which kind of tasks correspond to hierarchically local functions? An answer to this question suggested by Tegmark [13], is that they are tasks that involve inputs that can be decomposed in parts and wholes, hierarchically. Images, speech and text are obvious examples. We believe that the correct answer is a bit more complex: the property of hierarchical compositionality depends on both x (the data) and f (the task). We will show evidence for this claim by showing a dissociation between two different tasks on the same type of inputs (images).

2.5 Tasks beyond hierarchically local functions

There are tasks such as recognition of inside-out or connectedness that are beyond feedforward networks and require a Turing machine or equivalently a recurrent network which is not local. Evidence for this is in Villalobos et al. [40] where they discuss tasks [36] that require Turing machines. The inside-out problem is mathematically closely related to the problem of checking the parity of a bit string [23]. Rather surprisingly, neural networks perform very poorly on tasks such as bit addition, multiplication or even learning the identity function, if not biased specifically to the task [12, 42, 38]. They also consistently fail to generalize to larger inputs [34]. Solving these types of tasks seems to crucially depend on using differentiable memory to store non-local correlations [9].

3 Experiments

To test the conjecture that both the function and the input are important for hierarchical compositionality we test: a) changing the function by changing the task from very local to hierarchically local (Figure 2); b) destroying the locality properties of the inputs by scrambling the images.

We will show that estimating color in an image can be done by one-hidden-layer classifiers as well as by deep convolutional networks; performance in both cases is not affected by scrambling. We will

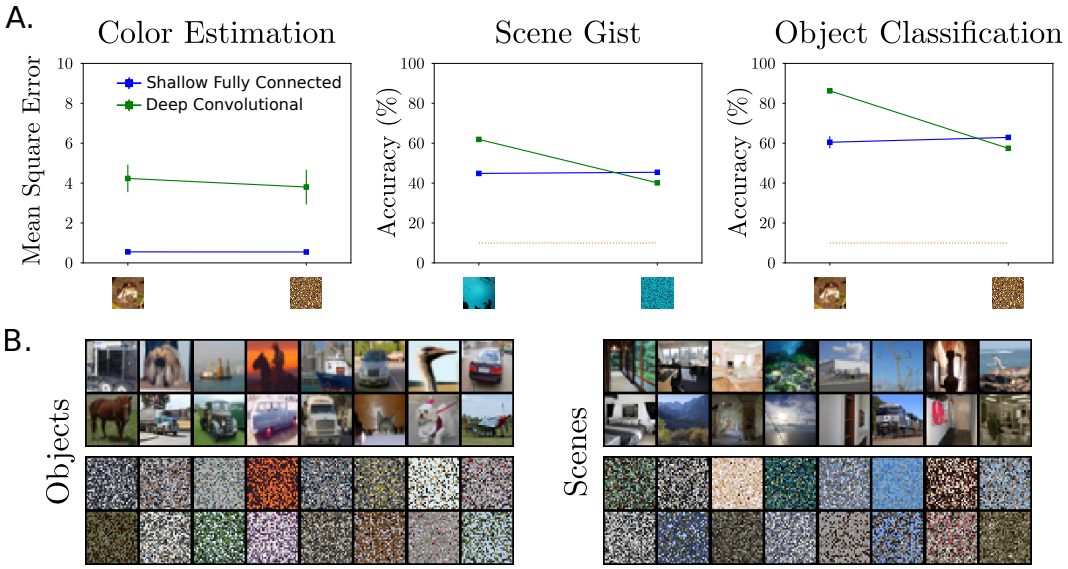

Figure 3: A. We show that performance is independent of scrambling for color estimation for both shallow fully connected networks and deep convolutional ones (left), while deep convolutional networks suffer greatly from the loss of hierarchical locality for scrambled scenes and objects (middle & right). B. Randomly sampled images (unscrambled and scrambled) from both the CIFAR-10 and MiniPlaces dataset [6] used in our experiments. The dotted line represents chance (10%).

also show that other tasks such as CIFAR-10 classification can be done with much better accuracy by a convolutional network than by a shallow network of the same size; here scrambling does not affect performance of the shallow networks but greatly reduces performance of the convolutional network. Thus the first type of tasks is trivially local (see Figure 2) since they can be computed by a linear combination of pixel values. This type of networks are not affected by scrambling. Classification of CIFAR is better when done by a convolution network; the underlying function is hierarchically local; scrambling destroys this hierarchical locality, because it destroys the neighborhoods of the image supporting the constituent functions. Interestingly, gist [25] (scene classification) though not completely local is somewhat hierarchically local: convolutional networks are better performing than linear ones and are affected by scrambling. Figure 3 summarizes these results.

3.1 Methods

Image Manipulation: We will define the *order* of scrambling with the following operator: $(\mathscr{S})$, where $\mathscr{S}_i$ is the *i-th* scrambled image. Here, $\mathscr{S}_0$ is the unscrambled image, and for $i \geq 1$ the scrambling procedure computes a random permutation of sub-dividing the $[M \times M]$ px image into 4 sub-blocks of size $[M/2 \times M/2]$ px– this procedure is done recurrently i times. We will also define $\mathscr{S}_\infty$ as the non-hierarchical scrambling operator *i.e.* a complete random permutation of pixels that have been remapped with no structure or hierarchical prior (Figure 3 B.). Here, we will show results of $\mathscr{S}_0$ vs $\mathscr{S}_\infty$ that strengthen our claim, and the Supp. Mat. will show intermediate results for $\mathscr{S}_i$.

Training: Five shallow fully connected networks with a single 10‘000 neuron hidden layer were trained for 5 epochs with a MSE loss (color estimation) or 80 epochs with a cross-entropy loss (object classification and scene gist) with a batch size of 64 for object classification and scene gist, and an SGD optimizer set at $[\eta = 0.001, \beta = 0.2]$, with the learning rate: η halved $(\times 0.5)$ from its original value after the 20-th epoch, and decimated $(\times 0.1)$ from its original value after the 40th epoch. Five deep convolutional networks (DenseNet121 [11]) were trained with a batch size of 64 for 20 epochs with SGD at $[\eta = 0.001, \beta = 0.2]$ & MSE loss (Color Estimation) or batch size 512, 300 epochs & cross-entropy loss (object classification and scene gist) with an SGD optimizer set at $\eta = 0.1$, nesterov momentum $\beta = 0.9$, dampening set to 1e-4, with the learning rate: η decimated $(\times 0.1)$ at the 150-*th* epoch and decimated again at the 225-*th* epoch.

Testing: To compute each systems accuracy for the scrambling condition, the scrambling noise seed was randomly picked and also saved and matched both at training and testing. In essence, the random scrambling procedure at training was matched at testing, such that although *locality* was altered, it was remapped homogeneously across all images at training and testing.

Data augmentation in the form of random horizontal mirroring with a chance of 0.5, and random cropping followed by re-scaling with an area ratio interval of $[0.7, 1.0]$. Data augmentation was performed *before* the scrambling operation, and was only used at training, and not testing.

3.2 Color Estimation

The luminance of each of the three RGB channels was estimated by modifying the number of output neurons to 3 in the last linear operation of each network. We found that after just 5 epochs of training with a MSE (mean square error) loss, that shallow networks achieved a lower mean square error of 0.552 ± 0.016 compared to deep convolutional networks 4.233 ± 0.689 (Figure 3 A. left). Naturally performance does not significantly change after scrambling (shallow networks: $0.552 \pm 0.016 \rightarrow 0.546 \pm 0.013$ (n.s.), and deep convolutional networks: $4.233 \pm 0.689 \rightarrow 3.803 \pm 0.871$ (n.s.)) given the *trivially local* type of task (Figure 2 (left)).

In a way this result suggests the importance of appropriate matching between *network architecture* (g) and task (f). Both networks are successful and can learn to estimate the average color of an image for both un-scrambled and scrambled images, yet shallow networks achieve smaller error due to a better match to the nature of the task. Interestingly, the deep convolutional network does not achieve the same lower error bound as shallow networks (pre $(\mathscr{S}_0)$ and post $(\mathscr{S}_\infty)$ scrambling), suggesting that hierarchical compositionality may be a non-optimal inductive bias for color estimation – which in turn may induce high perceptual variance across networks (see Emery & Webster [8]).

3.3 Scene Gist

To evaluate effects of scene gist on our two candidate networks, we decided to use a subset of the MiniPlaces dataset as introduced in Deza & Konkle [6]. The MiniPlaces dataset consists of a collection of 5000 high resolution 512×512 sub-selection of 20 scene categories from the original Places dataset [43]. However to make comparisons fair between both experiments the images from the MiniPlaces dataset were rescaled the stimuli to $[32 \times 32 \times 3]$ and only picked a subset of 10 visual categories. These scene categories spanned a variety of visual attributes ranging from indoors to outdoors and open and closed. The final categories that we sub-selected were: [ocean, industrial area, badlands, bedroom, bridge, forest path, kitchen, office, corridor, mountain]. Finally, our newly re-scaled "TinyPlaces" dataset consisted of 4500 images which were used for training per class, 250 for validation per class (not-evaluated), and 250 for testing per class.

We found that deep CNNs achieved $61.904 \pm 1.015\%$ for un-scrambled scene gist recognition and shallow networks achieved $44.848 \pm 0.311\%$. Meanwhile for scrambled scene gist deep CNNs achieved $40.136 \pm 0.497\%$ taking a total accuracy difference of $\Delta = 21.768\%$ *(un-paired)*, while shallow fully connected networks naturally stayed at the same performance of $45.416 \pm 0.261\%$ (Figure 3 A. middle). Interestingly, shallow networks *marginally* overperformed deep convolutional networks for this task although this may be due to our implementation difference where shallow networks had an order of magnitude greater of number of parameters.

3.4 Object Classification

We then carried out an experiment in which we destroyed locality in CIFAR-10 images by scrambling pixels in the same pseudorandom way for all data in the training and testing set (Figure 3 A. right). The performance of a shallow network stayed constant once again from $60.472 \pm 3.049\%$ to $62.902 \pm 0.232\%$; while the better performance of deep convolutional networks *was greatly impacted,*. Un-scrambled performance dropped from $86.248 \pm 0.457\%$ to $57.446 \pm 0.686\%$ yielding a difference of performance of $\Delta = 28.802\%$ (*un-paired*). These results imply that locality is crucial for the performance advantage of deep convolutional networks in such tasks. Further, notice that both the initial unscrambled accuracy and the perceptual drop is greater for object classification than scene-gist for an approximately similar amount of training data (4500 scenes/class; 5000 objects/class), testing data (250 scenes/class; 1000 objects/class), training regime (SGD), loss function (cross-entropy), class labels (10) and chance level (10%).

Our model forecast suggests that the hierarchically local compositional nature of objects (from contrast, to edges, to texture, to shape), will yield high performance for the CNN models that can exploit this structure in the data, while shallow fully connected networks will perform above chance, but not perform as highly as CNNs as they do not exploit locality in the image structure. However, as we increase the degrees of scrambling of the image stimuli ($\mathscr{S}_0 \rightarrow \mathscr{S}_\infty$), the locality prior is lost – in addition to the global shape of the object – thus degrading the performance of the CNN's, while shallow networks remain indifferent to this type of image manipulation.

4 Discussion

So far we have compared deep convolutional networks with shallow networks. The same results, however, suggest that convolutional deep networks are, for certain tasks, exponentially more efficient in terms of required parameters than densely connected deep networks. The obvious question is then: since deep networks contain convolutional ones, may gradient descent converge to the convolutional ones? The question is intriguing, especially in light of the well-recognized benefits of overparametrization. For now, let us articulate this question:

Show experimentally that, for learning tasks that are performed better by CNNs, dense deep networks networks cannot be trained with SGD to reach the same level of performance without an exponential sample complexity problem.

We performed a preliminary experiment (Figure 4) in which the architecture of a fully connected network was initialized to be a "noisy" convolutional one with additional nonconvolutional connections (i.e. a Toeplitz matrix with some additional non-zero entries). In Figure 4 we added non-zero entries to the Toeplitz matrix with probability 0.05% and plotted the Frobenius norm of these additional weights during training. We find that SGD preserves these additional weights and hence dense layers do not converge to convolutions naturally, even if initialized close to them.

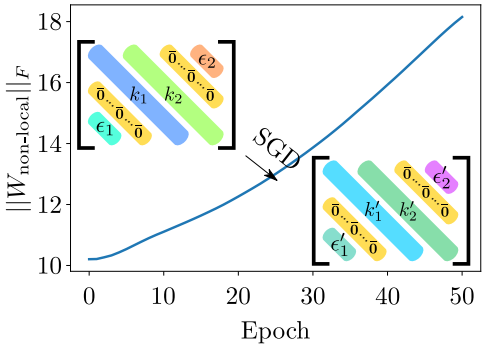

Figure 4: SGD preserves non-locality.

Since most of the success stories of DL involve convolutional networks applied to recognition and classification of images, speech, and text, it is tempting to conclude that the underlying reason is hierarchical compositionality of the function associated with those tasks. The reason, of course, is that the theorem from Mhaskar & Poggio [17] discussed in this paper shows that deep networks but not shallow ones can avoid the curse of dimensionality in approximating hierarchical functions (the theorem, is not, strictly speaking, a separation result but we have shown in this paper that there exist compositional tasks which require exponential complexity by shallow networks). The question left open by the mathematics is which tasks correspond to hierarchically local functions. What we have shown here is that the domain *and* the task itself are important. Images have a local structure at several scales that is a prerequisite for compositionality (see also Brendel & Bethge [2]). However, a simple task such as estimating the color of an object does not require this hierarchical compositionality as does scene gist or object classification itself.

Further one can see how playing with different function mappings (g) and tasks (f) can tell us about the structure of images. For example, suppose we did not know that scenes 'have a shape' as suggested in Oliva & Torralba [25]; we could perform an experiment like the one from Figure 3 to test if the compositional structure of scenes is similar to the compositional structure of objects. If hierarchical locality is not a prior of scenes (unlike of objects) we would expect little to no drop in performance for the scrambled condition. However this is not the case, as we have shown in our experiments and recently in Tadros et al. [35] – thus giving insight on the compositionality and structure of scenes.

Finally, the mathematics provides two important lessons for future applications: weight sharing helps reducing the complexity of approximation for appropriate problems, but it is *locality* that matters for eliminating the curse of dimensionality. Thus non-convolutional, hierarchically local networks may be appropriate for certain tasks. Recent work by Elsayed et al. [7] supports this prediction in showing that a controlled departure from convolutions can lead to better performance.

Broader Impact

In terms of ethical aspects and future societal consequences the impact of this paper is quite indirect, as far as we can see. The main contribution is towards a characterization of the convolutional priors used in current deep networks and their relation with the mathematical theory of function approximation. Other broader implications were mentioned in the discussion with regards to explainable models and a way to test compositionality theories potentially in vision science and cognition. Indirectly, of course, a better theory of learning machines may well have huge society impacts in some good ways – helping developing machines that can better solve problems for us – and in bad way – reducing jobs available to humans.

Acknowledgements This work was supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF 1231216. This work was also supported in part by C-BRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA, the National Science Foundation, Intel Corporation, and the DoD Vannevar Bush Fellowship. Authors would also like to thank Brando Miranda for providing the authors with image hierarchical image scrambling code.

References

[1] Anselmi, F., Rosasco, L., Tan, C., and Poggio, T. Deep convolutional network are hierarchical kernel machines. *Center for Brains, Minds and Machines (CBMM) Memo No. 35, also in arXiv*, 2015.

[2] Brendel, W. and Bethge, M. Approximating CNNs with bag-of-local-features models works surprisingly well on imagenet. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SkfMWhAqYQ.

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

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

[5] Daubechies, I., DeVore, R., Foucart, S., Hanin, B., and Petrova, G. Nonlinear approximation and (deep) relu networks. *arXiv e-prints*, art. arXiv:1905.02199, May 2019.

[6] Deza, A. and Konkle, T. Emergent properties of foveated perceptual systems. *arXiv preprint arXiv:2006.07991*, 2020.

[7] Elsayed, G. F., Ramachandran, P., Shlens, J., and Kornblith, S. Revisiting spatial invariance with low-rank local connectivity. *International Conference on Machine Learning (ICML)*, 2020.

[8] Emery, K. J. and Webster, M. A. Individual differences and their implications for color perception. *Current Opinion in Behavioral Sciences*, 30:28–33, 2019.

[9] Graves, A., Wayne, G., Reynolds, M., Harley, T., Danihelka, I., Grabska-Barwiska, A., Colmenarejo, S. G., Grefenstette, E., Ramalho, T., Agapiou, J., Badia, A. P., Hermann, K. M., Zwols, Y., Ostrovski, G., Cain, A., King, H., Summerfield, C., Blunsom, P., Kavukcuoglu, K., and Hassabis, D. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, October 2016. ISSN 00280836. URL http://dx.doi.org/10.1038/nature20101.

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

[11] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[12] Kaiser, L. and Sutskever, I. Neural gpus learn algorithms. *ArXiv*, 1511.08228, 2015.

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

[14] Linial, N., Y., M., and N., N. Constant depth circuits, fourier transform, and learnability. *Journal of the ACM*, 40(3):607620, 1993.

[15] Livni, R., Shalev-Shwartz, S., and Shamir, O. A provably efficient algorithm for training deep networks. *CoRR*, abs/1304.7045, 2013. URL http://arxiv.org/abs/1304.7045.

[16] Mhaskar, H. Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics*, pp. 61–80, 1993.

[17] Mhaskar, H. and Poggio, T. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, pp. 829– 848, 2016.

[18] Mhaskar, H. and Poggio, T. A. Deep vs. shallow networks : An approximation theory perspective. *CoRR*, abs/1608.03287, 2016. URL http://arxiv.org/abs/1608.03287.

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

[20] Mhaskar, H. N. Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics*, 1:61–80, 1993.

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

[22] Mhaskar, H. N. Neural networks for optimal approximation of smotth and analytic functions. *Neural Computation*, 8:164–177, 1996.

[23] Minsky, M. and Papert, S. *Perceptrons: An Introduction to Computational Geometry*. The MIT Press, ISBN 0-262-63022-2, Cambridge MA, 1972.

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

[25] Oliva, A. and Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[26] Petersen, P. and Voigtlaender, F. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *arXiv e-prints*, art. arXiv:1709.05289, September 2017.

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

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

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

[30] Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., and Liao, Q. Theory I: Why and when can deep - but not shallow - networks avoid the curse of dimensionality. Technical report, CBMM Memo No. 058, MIT Center for Brains, Minds and Machines, 2016.

[31] Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., and Liao, Q. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14(5):503–519, Oct 2017. ISSN 1751-8520. doi: 10.1007/s11633-017-1054-2. URL https://doi.org/10.1007/s11633-017-1054-2.

[32] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[33] Safran, I. and Shamir, O. Depth separation in relu networks for approximating smooth non-linear functions. *arXiv:1610.09887v1*, 2016.

[34] Shalev-Shwartz, S., Shamir, O., and Shammah, S. Failures of gradient-based deep learning. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3067–3075, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/shalev-shwartz17a.html.

[35] Tadros, T., Cullen, N. C., Greene, M. R., and Cooper, E. A. Assessing neural network scene classification from degraded images. *ACM Trans. Appl. Percept.*, 16(4), September 2019. ISSN 1544-3558. doi: 10.1145/3342349. URL https://doi.org/10.1145/3342349.

[36] Tang, H., Schrimpf, M., Lotter, W., Moerman, C., Paredes, A., Caro, J. O., Hardesty, W., Cox, D., and Kreiman, G. Recurrent computations for visual pattern completion. *Proceedings of the National Academy of Sciences*, 115(35):8835–8840, 2018.

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

[38] Trask, A., Hill, F., Reed, S., Rae, J., Dyer, C., and Blunsom, P. Neural arithmetic logic units. *ArXiv*, 1808.00508, 2018.

[39] van Bergen, R. S. and Kriegeskorte, N. Going in circles is the way forward: the role of recurrence in visual inference. *arXiv preprint arXiv:2003.12128*, 2020.

[40] Villalobos, K., Stih, V., Ahmadinejad, A., Sundaram, S., Dozier, J., Francl, A., Azevedo, F., Sasaki, T., and Boix, X. Do neural networks for segmentation understand insideness? Technical report, Center for Brains, Minds and Machines (CBMM), 2020.

[41] Yarotsky, D. Error bounds for approximations with deep relu networks. *CoRR*, abs/1610.01145, 2016. URL http://arxiv.org/abs/1610.01145.

[42] Zaremba, W. and Sutskever, I. Learning to execute. *ArXiv*, abs/1410.4615, 2014.

[43] Zhou, D. The regularity of reproducing kernel hilbert spaces in learning theory. 2001. preprint.

5 Supplementary Material

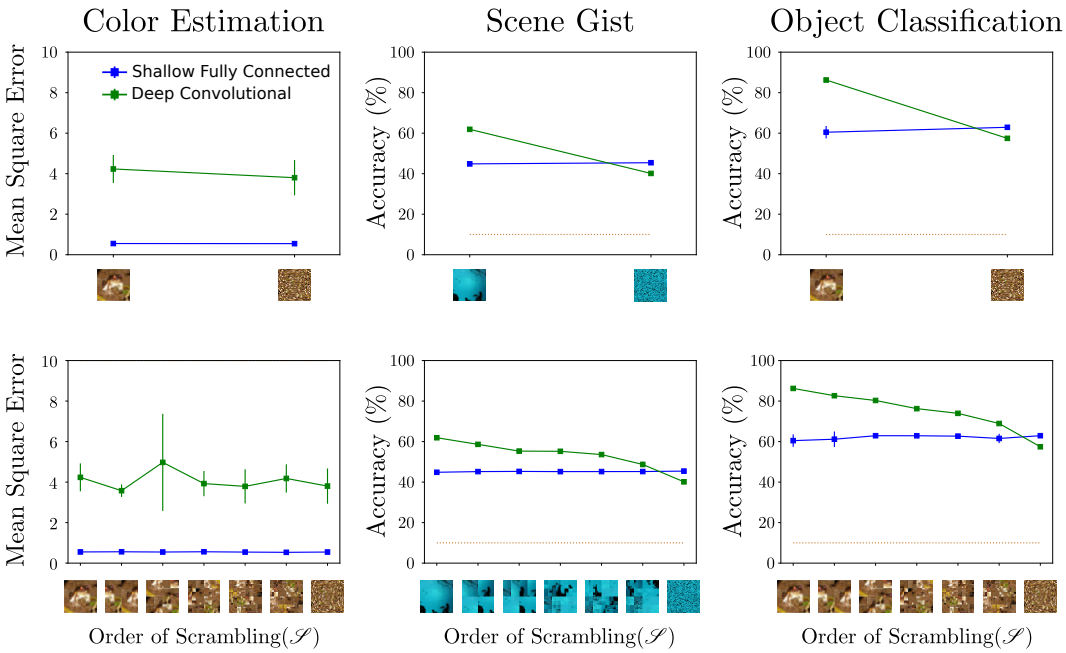

Figure 5: Main (top) and extended (bottom) results from our paper. See section 5.1 for a full description. The dotted line represents chance (10%).

5.1 Hierarchical Scrambling

The full and extended set of results with errorbars (mean $\pm$ standard deviation) from 5 randomly initialized networks (single hidden layer – fully connected and DenseNet121) across the multiple levels of hierarchical scrambling $\mathscr{S}_0 \rightarrow \mathscr{S}_5$ and asynchronous scrambling $\mathscr{S}_\infty$ is shown in Figure 5. These extended pattern of results agree with out notion of hierarchical locality. **Color Estimation**: Shallow fully connected networks converge to similar low Mean Square Error (MSE) estimation shown via the little to no variation of the variance per data point; deep convolutional networks perform the same *independent* of *order of scrambling* to approximate color (albeit greater error and variance) – though still being able to compute the task. Approximate chance for this task (not shown) is 128 pixels of MSE for a randomly initialized network with no training. There is no obvious decrease in performance for both the tasks: stable and constant for shallow fully connected networks, unstable and constant for deep convolutional networks. Interestingly, even across different initializations, deep convolutional networks never achieve lower MSE than shallow fully connected networks. Further experiments should explore how *stability* and training regimes (*i.e.* optimizer) affect the convergence of these results (mainly targeted at *trivially local tasks*) for deep convolutional networks. **Scene Gist**: As hierarchical scrambling is increased, scene gist performance for deep convolutional networks decreases gradually – an expected consequence of the piece-wise disruption of hierarchical locality. Performance for shallow fully connected network stays constant. **Object Classification**: The same effect of reduction in performance (as in scene gist) is shown for object classification as a function of hierarchical scrambling, though this disruption is stronger than scene gist.

A full list of sample objects and scenes across varying levels of scrambling can be seen in Figure 6. It is worth noticing that in our experiments each randomly initialized network was also paired with a separate form of random scrambling depending on the order. For example, consider two networks A and B: network A may be exposed to one candidate mapping for $\mathscr{S}_1$ while network B may be exposed to another candidate mapping for $\mathscr{S}_4$ under a different recursive noise seed. Even under such levels of randomness in our experiments, the standard deviation is only highly variable for deep convolutional networks performing color estimation.

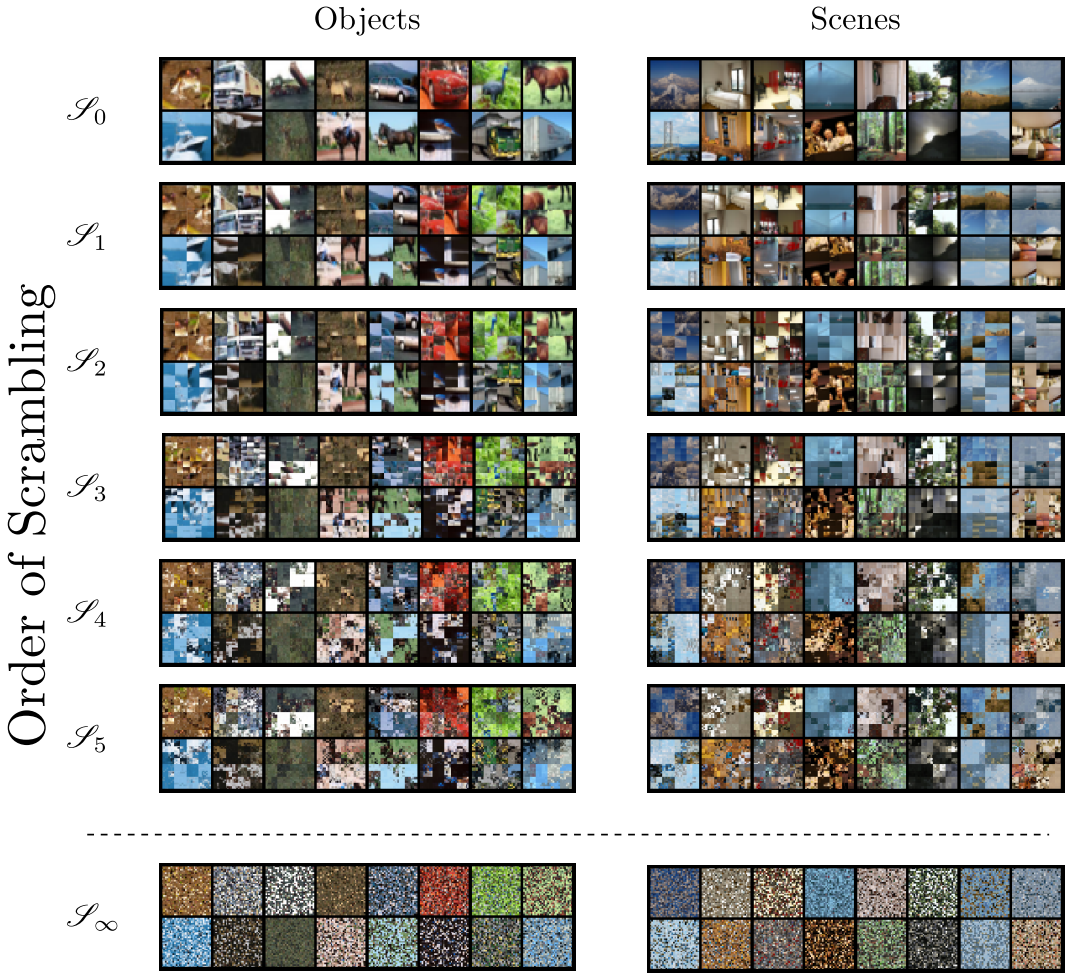

Figure 6: A full collection of randomly picked images from objects and scenes user in our experiments ranging from hierarchical scrambling: $\mathscr{S}_0 \to \mathscr{S}_5$, to asynchronous scrambling: $\mathscr{S}_\infty$.

5.2 Details of experiment on Toeplitz matrices with nonlocal connections

We implemented the addition of nonlocal connection to a convolutional network by constructing two networks – one CNN and one fully connected with compatible architecture. The CNN had 4 convolutional layers with kernel size 3, stride of 1 and number of output channels 3, 6, 12, 12 respectively, followed by two fully connected layers (with 1024 hidden units) and no batch normalization layers. The models were trained with SGD with learning rate $\eta = 0.01$, momentum of 0.9 and batch size of 64.

After initialization of the CNN, we transformed the convolution layers into Toeplitz matrices and used these as initialization for the fully connected network. We then randomly added additional nonzero weights to the sparse Toeplitz matrices to implement the nonlocal connections. During training, we masked the zero entries to keep them from being updated by backpropagation steps. We added connections with probability of 0.05%, which resulted in increase of nonzero weights to 77005 up from 72900 for the first layer (note that in a typically initialized dense layer this would be ~ 8.3M parameters). In Figure 4 in the main text we plotted the Frobenius norm of only the additional nonlocal connections in the first layer. We observed that this norm kept increasing throughout training, signalling that the nonlocal connections were not pruned out by SGD, agreeing with the hypothesis that fully connected layers do not converge to convolutions with standard SGD.