CENTER FOR
Brains
Minds+
Machines

CBMM Memo No. 100 **August 24, 2019**

Theoretical Issues in Deep Networks: Approximation, Optimization and Generalization[1]

Tomaso Poggio[1], **Andrzej Banburski** [1], **Qianli Liao**[1]

[1]Center for Brains, Minds, and Machines, MIT

Abstract

While deep learning is successful in a number of applications, it is not yet well understood theoretically. A satisfactory theoretical characterization of deep learning however, is beginning to emerge. It covers the following questions: 1) *representation power* of deep networks 2) *optimization* of the empirical risk 3) *generalization properties* of gradient descent techniques — why the expected error does not suffer, despite the absence of explicit regularization, when the networks are overparametrized? In this review we discuss recent advances in the three areas. In *approximation theory* both shallow and deep networks have been shown to approximate any continuous functions on a bounded domain at the expense of an exponential number of parameters (exponential in the dimensionality of the function). However, for a subset of compositional functions, deep networks of the convolutional type (even without weight sharing) can have a linear dependence on dimensionality, unlike shallow networks. In *optimization* we discuss the loss landscape for the exponential loss function. It turns out that global minima at infinity are completely degenerate. The other critical points of the gradient are less degenerate, with at least one – and typically more – nonzero eigenvalues. This suggests that stochastic gradient descent will find with high probability the global minima. To address the question of *generalization* for classification tasks, we use classical uniform convergence results to justify minimizing a surrogate exponential-type loss function under a unit norm constraint on the weight matrix at each layer. It is an interesting side remark, that such minimization for (homogeneous) ReLU deep networks implies maximization of the margin. The resulting constrained gradient system turns out to be identical to the well-known *weight normalization* technique, originally motivated from a rather different way. We also show that standard gradient descent contains an implicit L_2 unit norm constraint in the sense that it solves the same constrained minimization problem with the same critical points (but a different dynamics). Our approach, which is supported by several independent new results, offers a solution to the puzzle about generalization performance of deep overparametrized ReLU networks, uncovering the origin of the underlying hidden complexity control in the case of deep networks.

[1]This preprint provides an updated summary of, and guide for, the main surviving results of the previous key memos of the theory series (Memo 091, 090, 073, 066, 058) from 2016 to today.

Theoretical Issues in Deep Networks: Approximation, Optimization and Generalization

Tomaso Poggio[a,1], **Andrzej Banburski**[a], **and Qianli Liao**[a]

[a]Center for Brains, Minds and Machines, MIT

This manuscript was compiled on August 25, 2019

While deep learning is successful in a number of applications, it is not yet well understood theoretically. A satisfactory theoretical characterization of deep learning however, is beginning to emerge. It covers the following questions: 1) *representation power* **of deep networks 2)** *optimization* **of the empirical risk 3)** *generalization properties* **of gradient descent techniques — why the expected error does not suffer, despite the absence of explicit regularization, when the networks are overparametrized? In this review we discuss recent advances in the three areas. In** *approximation theory* **both shallow and deep networks have been shown to approximate any continuous functions on a bounded domain at the expense of an exponential number of parameters (exponential in the dimensionality of the function). However, for a subset of compositional functions, deep networks of the convolutional type (even without weight sharing) can have a linear dependence on dimensionality, unlike shallow networks. In** *optimization* **we discuss the loss landscape for the exponential loss function. It turns out that global minima at infinity are completely degenerate. The other critical points of the gradient are less degenerate, with at least one – and typically more – nonzero eigenvalues. This suggests that stochastic gradient descent will find with high probability the global minima. To address the question of** *generalization* **for classification tasks, we use classical uniform convergence results to justify minimizing a surrogate exponential-type loss function under a unit norm constraint on the weight matrix at each layer – since the interesting variables for classification are the weight** *directions* **rather than the weights. As a side remark, such minimization for (homogeneous) ReLU deep networks implies maximization of the margin. The resulting constrained gradient system turns out to be identical to the well-known** *weight normalization* **technique, originally motivated from a rather different way. We also show that standard gradient descent contains an implicit L_2 unit norm constraint in the sense that it solves the same constrained minimization problem with the same critical points (but a different dynamics). Our approach, which is supported by several independent new results (1–4), offers a solution to the puzzle about generalization performance of deep overparametrized ReLU networks, uncovering the origin of the underlying hidden complexity control in the case of deep networks.**

Machine Learning | Deep learning | Approximation | Optimization | Generalization

1. Introduction

In the last few years, deep learning has been tremendously successful in many important applications of machine learning. However, our theoretical understanding of deep learning, and thus the ability of developing principled improvements, has lagged behind. A satisfactory theoretical characterization of deep learning is emerging. It covers the following areas: 1) *approximation* properties of deep networks 2) *optimization* of the empirical risk 3) *generalization* properties of gradient descent techniques – why the expected error does not suffer, despite the absence of explicit regularization, when the networks are overparametrized?

A. When Can Deep Networks Avoid the Curse of Dimensionality?. We start with the first set of questions, summarizing results in (5–7), and (8, 9). The main result is that deep networks have the theoretical guarantee, which shallow networks do not have, that they can avoid the *curse of dimensionality* for an important class of problems, corresponding to *compositional functions*, that is functions of functions. An especially interesting subset of such compositional functions are *hierarchically local compositional functions* where all the constituent functions are local in the sense of bounded small dimensionality. The deep networks that can approximate them without the curse of dimensionality are of the deep convolutional type – though, importantly, weight sharing is not necessary.

Implications of the theorems likely to be relevant in practice are:

a) *Deep convolutional architectures* have the theoretical guarantee that they can be *much better* than one layer architectures such as kernel machines for certain classes of problems; b) the problems for which certain deep networks are guaranteed to avoid the *curse of dimensionality* (see for a nice review (10)) correspond to input-output mappings that are *compo-*

Significance Statement

In the last few years, deep learning has been tremendously successful in many important applications of machine learning. However, our theoretical understanding of deep learning, and thus the ability of developing principled improvements, has lagged behind. A theoretical characterization of deep learning is now beginning to emerge. It covers the following questions: 1) *representation power* of deep networks 2) *optimization* of the empirical risk 3) *generalization properties* of gradient descent techniques – how can deep networks generalize despite being overparametrized – more weights than training data – in the absence of any explicit regularization? We review progress on all three areas showing that 1) for a the class of compositional functions deep networks of the convolutional type are exponentially better approximators than shallow networks; 2) only global minima are effectively found by stochastic gradient descent for over-parametrized networks; 3) there is a hidden norm control in the minimization of cross-entropy by gradient descent that allows generalization despite overparametrization.

T.P. designed research; T.P., A.B., and Q.L. performed research; and T.P. and A.B. wrote the paper.

The authors declare no conflict of interest.

[1]To whom correspondence should be addressed. E-mail: tp@csail.mit.edu

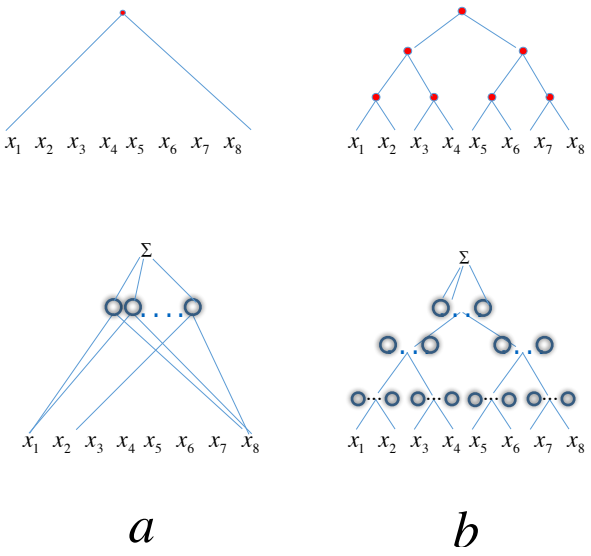

Fig. 1. The top graphs are associated to *functions*; each of the bottom diagrams depicts the ideal *network* approximating the function above. In a) a shallow universal network in 8 variables and N units approximates a generic function of 8 variables $f(x_1, \cdots, x_8)$. Inset b) shows a hierarchical network at the bottom in $n = 8$ variables, which approximates well functions of the form $f(x_1, \cdots, x_8) = h_3(h_{21}(h_{11}(x_1, x_2), h_{12}(x_3, x_4)), h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8)))$ as 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 todays deep networks, but equivalent to them (25)). 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 (9).

sitional with local constituent functions; c) the key aspect of convolutional networks that can give them an exponential advantage is *not weight sharing* but *locality* at each level of the hierarchy.

B. Related Work. Several papers in the '80s focused on the approximation power and learning properties of one-hidden layer networks (called shallow networks here). Very little appeared on multilayer networks, (but see (11–15)). By now, several papers (16–18) have appeared. (8, 19–22) derive new upper bounds for the approximation by deep networks of certain important classes of functions which avoid the curse of dimensionality. The upper bound for the approximation by shallow networks of general functions was well known to be exponential. It seems natural to assume that, since there is no general way for shallow networks to exploit a compositional prior, lower bounds for the approximation by shallow networks of compositional functions should also be exponential. In fact, examples of specific functions that cannot be represented efficiently by shallow networks have been given, for instance in (23–25). An interesting review of approximation of univariate functions by deep networks has recently appeared (26).

C. Degree of approximation. 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 in keeping with the standard choice in approximation theory. As it turns out, the results of this section require the sup norm in order to be independent from the unknown distribution of the input data.

Let V_N be the be set of all networks of a given kind with N units (which we take to be or measure of the complexity of the approximant network). The *degree of approximation* is defined by $\mathsf{dist}(f, V_N) = \inf_{P \in V_N} \|f - P\|$. 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. The only a priori information on the class of target functions f, is codified by the statement that $f \in W$ for some subspace $W \subseteq \mathbb{X}$. This subspace is a smoothness and compositional class, characterized by the parameters m and d ($d = 2$ in the example of Figure 1 ; it is the size of the kernel in a convolutional network).

D. Shallow and deep networks. This section characterizes conditions under which deep networks are "better" than shallow network in approximating functions. Thus we compare shallow (one-hidden layer) networks with deep networks as shown in 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. Each node in the networks corresponds to a node in the graph of the function to be approximated, as shown in the Figure. A unit is a neuron which computes $(\langle x, w \rangle + b)_+$, where w is the vector of weights on the vector input x. Both w and the real number b are parameters tuned by learning. We assume here that each node in the networks computes the linear combination of r such units $\sum_{i=1}^{r} c_i(\langle x, w_i \rangle + b_i)_+$. Notice that in our main example 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 our results hold also for these more complex networks (see (25)).

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

  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 compositional function f. (25) formalizes the more general compositional case using directed acyclic graphs.

- The approximation of functions with a *compositional structure* – can be achieved with the same degree of accuracy by deep and shallow networks but the number of

parameters are much smaller for the deep networks than for the shallow network with equivalent approximation accuracy.

We approximate functions 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 the function graph with each node h_i being a ridge function, comprising one or more neurons.

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

$$x \mapsto \sum_{k=1}^{N} a_k \sigma(\langle w_k, x\rangle + b_k),$$

where $w_k \in \mathbb{R}^n$, $b_k, a_k \in \mathbb{R}$. The number of trainable parameters here is $(n + 2)N \sim n$. Let $m \geq 1$ be an integer, and W_m^n be the set of all functions of n variables with continuous partial derivatives of orders up to $m < \infty$ such that $\|f\| + \sum_{1\leq|\mathbf{k}|_1\leq m} \|D^{\mathbf{k}} f\| \leq 1$, 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 $W_m^{n,2}$ to be the class of all compositional functions f of n variables with a binary tree architecture and constituent functions h in W_m^2. We define the corresponding class of deep networks $\mathcal{D}_{N,2}$ to be the set of all deep networks with a binary tree architecture, where each of the constituent nodes is in $\mathcal{S}_{M,2}$, where $N = |V|M$, V being the set of non–leaf vertices of the tree. 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 $\mathcal{D}_{N,2}$ is $4N$.

The first theorem is about shallow networks.

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

$$N = \mathcal{O}(\epsilon^{-n/m}) \text{ and is the best possible.} \qquad [1]$$

The estimate of Theorem 1 is the best possible if the only a priori information we are allowed to assume is that the target function belongs to $f \in W_m^n$. The exponential dependence on the dimension n of the number $e^{-n/m}$ of parameters needed to obtain an accuracy $\mathcal{O}(\epsilon)$ is known as the *curse of dimensionality*. Note that the constants involved in $\mathcal{O}$ in the theorems will depend upon the norms of the derivatives of f as well as σ.

Our second and main theorem is about deep networks with smooth activations (preliminary versions appeared in (6–8)). We formulate it in the binary tree case for simplicity but it extends immediately to functions that are compositions of constituent functions of a fixed number of variables d (in convolutional networks d corresponds to the size of the kernel).

Theorem 2 *For $f \in W_m^{n,2}$ 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. The complexity of the network to provide approximation with accuracy at least ϵ is*

$$N = \mathcal{O}((n - 1)\epsilon^{-2/m}). \qquad [2]$$

The proof is in (25). 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. The result of the theorem can be extended to non-smooth ReLU(25).

In summary, 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 $\mathcal{O}(\epsilon^{-n/m})$ 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 $\mathcal{O}(\epsilon^{-2/m})$ trainable parameters. Note that 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 the deep network – in this case the graph corresponding to $W_m^{n,d}$.

2. The Optimization Landscape of Deep Nets with Smooth Activation Function

The main question in optimization of deep networks is to the landscape of the empirical loss in terms of its global minima and local critical points of the gradient.

A. Related work. There are many recent papers studying optimization in deep learning. For optimization we mention work based on the idea that noisy gradient descent (27–30) can find a global minimum. More recently, several authors studied the dynamics of gradient descent for deep networks with assumptions about the input distribution or on how the labels are generated. They obtain global convergence for some shallow neural networks (31–36). Some local convergence results have also been proved (37–39). The most interesting such approach is (36), which focuses on minimizing the training loss and proving that randomly initialized gradient descent can achieve zero training loss (see also (40–42)). In summary, there is by now an extensive literature on optimization that formalizes and refines to different special cases and to the discrete domain our results of (43, 44).

B. Degeneracy of global and local minima under the exponential loss. The *first part* of the argument of this section relies on the obvious fact (see (1)), that for RELU networks under the hypothesis of an exponential-type loss function, there are *no local minima that separate the data* – the only critical points of the gradient that separate the data are the global minima.

Notice that the global minima are at $\rho = \infty$, when the exponential is zero. As a consequence, the Hessian is identically zero with all eigenvalues being zero. On the other hand any point of the loss at a finite ρ has nonzero Hessian: for instance in the linear case the Hessian is proportional to $\sum_n^N x_n x_n^T$. The local minima which are not global minima must misclassify. How degenerate are they?

Simple arguments (1) suggest that the critical points which are not global minima cannot be completely degenerate. We thus have the following

Property 1 *Under the exponential loss, global minima are completely degenerate with all eigenvalues of the Hessian (W of them with W being the number of parameters in the network) being zero. The other critical points of the gradient are less*

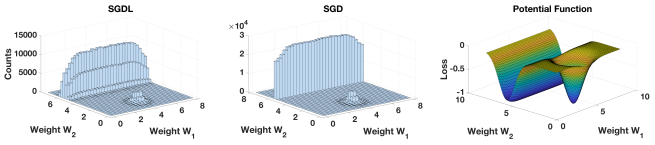

Fig. 2. Stochastic Gradient Descent and Langevin Stochastic Gradient Descent (SGDL) on the 2D potential function shown above leads to an asymptotic distribution with the histograms shown on the left. As expected from the form of the Boltzmann distribution, both dynamics prefer degenerate minima to non-degenerate minima of the same depth. From (1).

degenerate, with at least one – and typically N – nonzero eigenvalues.

For the general case of non-exponential loss and smooth nonlinearities instead of the RELU the following conjecture has been proposed (1):

Conjecture 1 *: For appropriate overparametrization, there are a large number of global zero-error minimizers which are degenerate; the other critical points – saddles and local minima – are generically (that is with probability one) degenerate on a set of much lower dimensionality.*

C. SGD and Boltzmann Equation. The second part of our argument (in (44)) is that SGD concentrates in probability on the most degenerate minima. The argument is based on the similarity between a Langevin equation and SGD and on the fact that the Boltzmann distribution is exactly the asymptotic "solution" of the stochastic differential Langevin equation and also of SGDL, defined as SGD with added white noise (see for instance (45)). The Boltzmann distribution is

$$p(f) = \frac{1}{Z} e^{-\frac{L}{T}}, \qquad [3]$$

where Z is a normalization constant, $L(f)$ is the loss and T reflects the noise power. The equation implies that SGDL prefers degenerate minima relative to non-degenerate ones of the same depth. In addition, among two minimum basins of equal depth, the one with a larger volume is much more likely in high dimensions as shown by the simulations in (44). Taken together, these two facts suggest that SGD selects degenerate minimizers corresponding to larger isotropic flat regions of the loss. Then SDGL shows concentration – *because of the high dimensionality* – of its asymptotic distribution Equation 3.

Together (43) and (1) suggest the following

Conjecture 2 *: For appropriate overparametrization of the deep network, SGD selects with high probability the global minimizers of the empirical loss, which are highly degenerate.*

3. Generalization

Recent results by (2) illuminate the apparent absence of "overfitting" (see Figure 4) in the special case of linear networks for binary classification. They prove that minimization of loss functions such as the logistic, the cross-entropy and the exponential loss yields asymptotic convergence to the maximum margin solution for linearly separable datasets, independently of the initial conditions and without explicit regularization. Here we discuss the case of nonlinear multilayer DNNs under exponential-type losses, for several variations of the basic gradient descent algorithm. The main results are:

- classical uniform convergence bounds for generalization suggest a form of complexity control on the dynamics of the weight *directions* V_k: minimize a surrogate loss subject to a unit L_p norm constraint;
- gradient descent on the exponential loss with an explicit L_2 unit norm constraint is equivalent to a well-known gradient descent algorithms *weight normalization* which is closely related to batch normalization;
- unconstrained gradient descent on the exponential loss yields a dynamics with the same critical points as weight normalization: the dynamics implicitly respects a L_2 unit constraint on the directions of the weights V_k.

We observe that several of these results *directly apply to kernel machines* for the exponential loss under the separability/interpolation assumption, because kernel machines are one-homogeneous.

A. Related work. A number of papers have studied gradient descent for deep networks (46–48). Close to the approach summarized here (details are in (1)) is the paper (49). Its authors study generalization assuming a regularizer because they are – like us – interested in normalized margin. Unlike their assumption of an explicit regularization, we show here that commonly used techniques, such as weight and batch normalization, in fact minimize the surrogate loss margin while controlling the complexity of the classifier without the need to add a regularizer or to use weight decay. Surprisingly, we will show that even standard gradient descent on the weights implicitly controls the complexity through an "implicit" unit L_2 norm constraint. Two very recent papers ((4) and (3)) develop an elegant but complicated margin maximization based approach which lead to some of the same results of this section (and many more). The important question of which conditions are necessary for gradient descent to converge to the maximum of the margin of $\tilde{f}$ are studied by (4) and (3). Our approach does not need the notion of maximum margin but our theorem 3 establishes a connection with it and thus with the results of (4) and (3). Our main goal here (and in (1)) is to achieve a simple understanding of where the complexity control underlying generalization is hiding in the training of deep networks.

B. Deep networks: definitions and properties. We define a deep network with K layers with the usual coordinate-wise scalar activation functions $\sigma(z) : \mathbf{R} \to \mathbf{R}$ 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 $k = 1, \cdots, K$. For simplicity we consider here the case of binary classification in which f takes scalar values, implying that the last layer matrix W^K is $W^K \in \mathbf{R}^{1,K_l}$. The labels are $y_n \in \{-1, 1\}$. The weights of hidden layer l are collected in a matrix of size $h_l \times h_{l-1}$. There are no biases apart form the input layer where the bias is instantiated by one of the input dimensions being a constant. The activation function in this section is the ReLU activation.

For ReLU activations the following important positive one-homogeneity property holds $\sigma(z) = \frac{\partial \sigma(z)}{\partial z} z$. A consequence of one-homogeneity is a structural lemma (Lemma 2.1 of (50))

$\sum_{i,j} W_k^{i,j} \left(\frac{\partial f(x)}{\partial W_k^{i,j}} \right) = f(x)$ where W_k is here the vectorized representation of the weight matrices W_k for each of the different layers (each matrix is a vector).

For the network, homogeneity implies $f(W;x) = \prod_{k=1}^{K} \rho_k f(V_1, \cdots, V_K; x_n)$, where $W_k = \rho_k V_k$ with the matrix norm $||V_k||_p = 1$. Another property of the Rademacher complexity of ReLU networks that follows from homogeneity is $\mathbb{R}_N(\mathbb{F}) = \rho \mathbb{R}_N(\tilde{\mathbb{F}})$ where $\rho = \rho_1 \prod_{k=1}^{K} \rho_k$, $\mathbb{F}$ is the class of neural networks described above.

We define $f = \rho \tilde{f}$; $\tilde{\mathbb{F}}$ is the associated class of normalized neural networks (we call $f(V;x) = \tilde{f}(x)$ with the understanding that $f(x) = f(W;x)$). Note that $\frac{\partial f}{\partial \rho_k} = \frac{\rho}{\rho_k} \tilde{f}$ and that the definitions of ρ_k, V_k and $\tilde{f}$ all depend on the choice of the norm used in normalization.

In the case of training data that can be separated by the networks $f(x_n) y_n > 0 \quad \forall n = 1, \cdots, N$. We will sometime write $f(x_n)$ as a shorthand for $y_n f(x_n)$.

C. Uniform convergence bounds: minimizing a surrogate loss under norm constraint.

Classical *generalization bounds for regression* (51) suggest that minimizing the empirical loss of a loss function such as the cross-entropy subject to constrained *complexity of the minimizer* is a way to to attain generalization, that is an expected loss which is close to the empirical loss:

Proposition 1 *The following generalization bounds apply to* $\forall f \in \mathbb{F}$ *with probability at least* $(1 - \delta)$*:*

$$L(f) \leq \hat{L}(f) + c_1 \mathbb{R}_N(\mathbb{F}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}} \qquad [4]$$

where $L(f) = \mathbf{E}[\ell(f(x), y)]$ is the expected loss, $\hat{L}(f)$ is the empirical loss, $\mathbb{R}_N(\mathbb{F})$ is the empirical Rademacher average of the class of functions $\mathbb{F}$, measuring its complexity; c_1, c_2 are constants that depend on properties of the Lipschitz constant of the loss function, and on the architecture of the network.

Thus minimizing under a constraint on the Rademacher complexity a surrogate function such as the cross-entropy (which becomes the logistic loss in the binary classification case) will minimize an upper bound on the expected classification error because such surrogate functions are upper bounds on the $0-1$ function. We can choose a class of functions $\tilde{\mathbf{F}}$ with normalized weights and write $f(x) = \rho \tilde{f}(x)$ and $\mathbb{R}_N(\mathbb{F}) = \rho \mathbb{R}_N(\tilde{\mathbb{F}})$. One can choose any fixed ρ as a (Ivanov) regularization-type tradeoff.

In summary, the problem of generalization may approached by minimizing the exponential loss – more in general an exponential-type loss, such the logistic and the cross-entropy – under a unit norm constraint on the weight matrices, since we are interested in the directions of the weights:

$$\lim_{\rho \to \infty} \arg \min_{||V_k||=1, \ \forall k} L(\rho \tilde{f}) \qquad [5]$$

where we write $f(W) = \rho \tilde{f}(V)$ using the homogeneity of the network. As it will become clear later, gradient descent techniques on the exponential loss automatically increase ρ to infinity. We will typically consider the sequence of minimizations over V_k for a sequence of increasing ρ. The key quantity for us is $\tilde{f}$ and the associated weights V_k; ρ is in a certain sense an auxiliary variable, a constraint that is progressively relaxed.

In the following we explore the implications for deep networks of this classical approach to generalization.

C.1. Remark: minimization of an exponential-type loss implies margin maximization .

Though not critical for our approach to the question of generalization in deep networks it is interesting that constrained minimization of the exponential loss implies margin maximization. This property relates our approach to the results of several recent papers (2–4). Notice that our theorem 3 as in (52) is a *sufficient condition for margin maximization*. Necessity is not true for general loss functions.

To state the margin property more formally, we adapt to our setting a different result due to (52) (they consider for a linear network a vanishing λ regularization term whereas we have for nonlinear networks a set of unit norm constraints). First we recall the definition of the empirical loss $L(f) = \sum_{n=1}^{N} \ell(y_n f(x_n))$ with an exponential loss function $\ell(yf) = e^{-yf}$. We define $\eta(f)$ a the *margin* of f, that is $\eta(f) = \min_n f(x_n)$.

Then our margin maximization theorem (proved in (1)) takes the form

Theorem 3 *Consider the set of* $V_k, k = 1, \cdots, K$ *corresponding to*

$$\min_{||V_k||=1} L(f(\rho_k, V_k)) \qquad [6]$$

where the norm $||V_k||$ *is a chosen* L_p *norm and* $L(f)(\rho_k, V_K) = L(\tilde{f}(\rho)) = \sum_n \ell(y_n \rho f(V; x_n))$ *is the empirical exponential loss. For each layer consider a sequence of increasing* ρ_k*. Then the associated sequence of* V_k *defined by Equation 6, converges for* $\rho \to \infty$ *to the maximum margin of* $\tilde{f}$*, that is to* $\max_{||V_k|| \leq 1} \eta(\tilde{f})$.

D. Minimization under unit norm constraint: weight normalization.

The approach is then to minimize the loss function $L(f(w)) = \sum_{n=1}^{N} e^{-f(W;x_n) y_n} = \sum_{n=1}^{N} e^{-\rho f(V_k;x_n) y_n}$, with $\rho = \prod \rho_k$, subject to $||V_k||_p^p = 1 \ \forall k$, that is under a unit norm constraint for the weight matrix at each layer (if $p = 2$ then $\sum_{i,j} (V_k)_{i,j}^2 = 1$ is the Frobenius norm). The minimization is understood as a sequence of minimizations for a sequence of increasing ρ_k. Clearly these constraints imply the constraint on the norm of the product of weight matrices for any p norm (because any induced operator norm is a sub-multiplicative matrix norm). The standard choice for a loss function is an exponential-type loss such the cross-entropy, which for binary classification becomes the logistic function. We study here the exponential because it is simpler and retains all the basic properties.

There are several gradient descent techniques that given the unconstrained optimization problem transform it into a *constrained* gradient descent problem. To provide the background let us formulate the standard unconstrained gradient descent problem for the exponential loss as it is used in practical training of deep networks:

$$\dot{W}_k^{i,j} = -\frac{\partial L}{\partial W_k^{i,j}} = \sum_{n=1}^{N} y_n \frac{\partial f(x_n; w)}{\partial W_k^{i,j}} e^{-y_n f(x_n; W)} \qquad [7]$$

where W_k is the weight matrix of layer k. Notice that, since the structural property implies that at a critical point we have $\sum_{n=1}^{N} y_n f(x_n; w) e^{-y_n f(x_n;W)} = 0$, the only critical points of this dynamics that separate the data (i.e. $y_n f(x_n; w) > 0 \ \forall n$) are global minima at infinity. Of course for separable data, while the loss decreases asymptotically to zero, the norm of the weights ρ_k increases to infinity, as we will see later. Equations 7 define a dynamical system in terms of the gradient of the exponential loss L.

The set of gradient-based algorithms enforcing a unit-norm constraints (53) comprises several techniques that are equivalent for small values of the step size. They are all good approximations of the true gradient method. One of them is the *Lagrange multiplier method*; another is the *tangent gradient method* based on the following theorem:

Theorem 4 *(53) Let $||u||_p$ denote a vector norm that is differentiable with respect to the elements of u and let $g(t)$ be any vector function with finite L_2 norm. Then, calling $\nu(t) = \frac{\partial ||u||_p}{\partial u}_{u=u(t)}$, the equation*

$$\dot{u} = h_g(t) = Sg(t) = (I - \frac{\nu\nu^T}{||\nu||_2^2})g(t) \qquad [8]$$

with $||u(0)|| = 1$, describes the flow of a vector u that satisfies $||u(t)||_p = 1$ for all $t \geq 0$.

In particular, a form for g is $g(t) = \mu(t)\nabla_u L$, the gradient update in a gradient descent algorithm. We call $Sg(t)$ the tangent gradient transformation of g. In the case of $p = 2$ we replace ν in Equation 8 with u because $\nu(t) = \frac{\partial ||u||_2}{\partial u} = u$. This gives $S = I - \frac{uu^T}{||u||_2^2}$ and $\dot{u} = Sg(t)$.

Consider now the empirical loss L written in terms of V_k and ρ_k instead of W_k, using the change of variables defined by $W_k = \rho_k V_k$ but without imposing a unit norm constraint on V_k. The flows in ρ_k, V_k can be computed as $\dot{\rho_k} = \frac{\partial W_k}{\partial \rho_k}\frac{\partial L}{\partial W_k} = V_k^T \frac{\partial L}{\partial W_k}$ and $\dot{V_k} = \frac{\partial W_k}{\partial V_k}\frac{\partial L}{\partial W_k} = \rho_k \frac{\partial L}{\partial W_k}$, with $\frac{\partial L}{\partial W_k}$ given by Equations 7.

We now enforce the unit norm constraint on V_k by using the tangent gradient transform on the V_k flow. This yields

$$\dot{\rho_k} = V_k^T \frac{\partial L}{\partial W_k} \quad \dot{V_k} = S_k \rho_k \frac{\partial L}{\partial W_k}. \qquad [9]$$

Notice that the dynamics above follows from the classical approach of controlling the Rademacher complexity of $\tilde{f}$ during optimization (suggested by bounds such as Equation 4. The approach and the resulting dynamics for the directions of the weights would seem different from the standard unconstrained approach in training deep networks. It turns out, however, that the dynamics described by Equations 9 is the same dynamics of *Weight Normalization*.

The technique of *Weight normalization* (54) was originally proposed as a small improvement on standard gradient descent "to reduce covariate shifts". It was defined for each layer in terms of $w = g\frac{v}{||v||}$, as

$$\dot{g} = \frac{v}{||v||}\frac{\partial L}{\partial w}\dot{v} = \frac{g}{||v||} S \frac{\partial L}{\partial w} \qquad [10]$$

with $S = I - \frac{vv^T}{||v||^2}$.

It is easy to see that Equations 9 are the same as the weight normalization Equations 10, if $||v||_2 = 1$. We now observe, multiplying Equation 9 by v^T, that $v^T\dot{v} = 0$ because $v^T S = 0$, implying that $||v||^2$ is constant in time with a constant that can be taken to be 1. Thus the two dynamics are the same.

E. Generalization with hidden complexity control. Empirically it appears that GD and SGD converge to solutions that can generalize even without batch or weight normalization. Convergence may be difficult for quite deep networks and generalization may not be as good as with batch normalization but it still occurs. How is this possible?

We study the dynamical system $\dot{W_k}^{i,j}$ under the reparametrization $W_k^{i,j} = \rho_k V_k^{i,j}$ with $||V_k||_2 = 1$. We consider for each weight matrix W_k the corresponding "vectorized" representation in terms of vectors $W_k^{i,j} = W_k$. We use the following definitions and properties (for a vector w):

- Define $\frac{w}{||w||_2} = \tilde{w}$; thus $w = ||w||_2\tilde{w}$ with $||\tilde{w}||_2 = 1$. Also define $S = I - \tilde{w}\tilde{w}^T = I - \frac{ww^T}{||w||_2^2}$.
- The following relations are easy to check:
  1. $\frac{\partial ||w||_2}{\partial w} = \tilde{w}$
  2. $\frac{\partial \tilde{w}}{\partial w} = \frac{S}{||w||_2}$.
  3. $Sw = S\tilde{w} = 0$
  4. $S^2 = S$

The gradient descent dynamic system used in training deep networks for the exponential loss is given by Equation 7. Following the chain rule *for the time derivatives*, the dynamics for W_k is exactly (see (1)) equivalent to the following dynamics for $||W_k|| = \rho_k$ and V_k:

$$\dot{\rho_k} = \frac{\partial ||W_k||}{\partial W_k}\frac{\partial W_k}{\partial t} = V_k^T \dot{W_k} \qquad [11]$$

and

$$\dot{V_k} = \frac{\partial V_k}{\partial W_k}\frac{\partial W_k}{\partial t} = \frac{S_k}{\rho_k}\dot{W_k} \qquad [12]$$

where $S_k = I - V_k V_k^T$. We used property 1 in 4 for Equation 11 and property 2 for Equation 12.

The key point here is that the dynamics of $\dot{V_k}$ includes a unit L_2 norm constraint: using the tangent gradient transform will not change the equation because $S^2 = S$.

As separate remarks , notice that if for $t > t_0$, f separates all the data, $\frac{d}{dt}\rho_k > 0$, that is ρ diverges to ∞ with $\lim_{t\to\infty} \dot{\rho} = 0$. In the 1-layer network case the dynamics yields $\rho \approx \log t$ asymptotically. For deeper networks, this is different. (1) shows (for one support vector) that the product of weights at each layer diverges faster than logarithmically, but each individual layer diverges slower than in the 1-layer case. The norm of the each layer grows at the same rate $\dot{\rho_k^2}$, independent of k. The V_k dynamics has stationary or critical points given by

$$W \sum \alpha_n(\rho(t)) \left(\frac{\partial \tilde{f}(x_n)}{\partial V_k^{i,j}} - V_k^{i,j} \tilde{f}(x_n) \right), \qquad [13]$$

where $\alpha_n = e^{-y_n \rho(t) \tilde{f}(x_n)}$. We examine later the linear one-layer case $\tilde{f}(x) = v^T x$ in which case the stationary points of the gradient are given by $\sum \alpha_n(\rho(t)(x_n - vv^T x_n)$. In the linear overparametrized case the critical point corresponds to the maximum margin of a degenerate minimum. In the general

case the critical points correspond for $\rho \to \infty$ to degenerate zero "asymptotic minima" of the loss.

To understand whether there exists a hidden complexity control in standard gradient descent, we check whether there exists an L_p norm for which unconstrained normalization is equivalent to constrained normalization.

From Theorem 4 we expect the constrained case to be given by the action of the following projector onto the tangent space:

$$S_p = I - \frac{\nu\nu^T}{||\nu||_2^2} \quad \text{with} \quad \nu_i = \frac{\partial ||w||_p}{\partial w_i} = \text{sign}(w_i) \circ \left(\frac{|w_i|}{||w||_p} \right)^{p-1}. \tag{14}$$

The constrained Gradient Descent is then

$$\dot{\rho_k} = V_k^T \dot{W}_k \quad \dot{V}_k = \rho_k S_p \dot{W}_k. \tag{15}$$

On the other hand, reparametrization of the unconstrained dynamics in the p-norm gives (following Equations 11 and 12)

$$\dot{\rho_k} = \frac{\partial ||W_k||_p}{\partial W_k} \frac{\partial W_k}{\partial t} = \text{sign}(W_k) \circ \left(\frac{|W_k|}{||W_k||_p} \right)^{p-1} \cdot \dot{W}_k$$

$$\dot{V_k} = \frac{\partial V_k}{\partial W_k} \frac{\partial W_k}{\partial t} = \frac{I - \text{sign}(W_k) \circ \left(\frac{|W_k|}{||W_k||_p} \right)^{p-1} W_k^T}{||W_k||_p^{p-1}} \dot{W}_k. \tag{16}$$

These two dynamical systems are clearly different for generic p reflecting the presence or absence of a regularization-like constraint on the dynamics of V_k.

As we have seen however, for $p = 2$ the 1-layer dynamical system obtained by minimizing L in ρ_k and V_k with $W_k = \rho_k V_k$ under the constraint $||V_k||_2 = 1$, is the weight normalization dynamics

$$\dot{\rho_k} = V_k^T \dot{W}_k \quad \dot{V}_k = S \rho_k \dot{W}_k, \tag{17}$$

which is quite similar to the standard gradient equations

$$\dot{\rho_k} = V_k^T \dot{W}_k \quad \dot{v} = \frac{S}{\rho_k} \dot{W}_k. \tag{18}$$

The two dynamical systems differ only by a ρ_k^2 factor in the $\dot{V_k}$ equations. However, the critical points of the gradient for the V_k flow, that is the point for which $\dot{V_k} = 0$, are the same in both cases since for any $t > 0$ $\rho_k(t) > 0$ and thus $\dot{V_k} = 0$ is equivalent to $S\dot{W_k} = 0$. Hence, gradient descent with unit L_p-norm constraint is equivalent to the standard, unconstrained gradient descent but only when $p = 2$. Thus

Fact 1 *The standard dynamical system used in deep learning, defined by $\dot{W_k} = -\frac{\partial L}{\partial W_k}$, implicitly respects a unit L_2 norm constraint on V_k with $\rho_k V_k = W_k$. Thus, under an exponential loss, if the dynamics converges, the V_k represent the minimizer under the L_2 unit norm constraint.*

Thus standard GD implicitly enforces the L_2 norm constraint on $V_k = \frac{W_k}{||W_k||_2}$, consistently with Srebro's results on implicit bias of GD. Other minimization techniques such as coordinate descent may be biased towards different norm constraints.

F. Linear networks and rates of convergence. The linear ($f(x) = \rho v^T x$) networks case (2) is an interesting example of our analysis in terms of ρ and v dynamics. We start with unconstrained gradient descent, that is with the dynamical system

$$\dot{\rho} = \frac{1}{\rho} \sum_{n=1}^{N} e^{-\rho v^T x_n} v^T x_n \quad \dot{v} = \frac{1}{\rho} \sum_{n=1}^{N} e^{-\rho v^T x_n} (x_n - v v^T x_n). \tag{19}$$

If gradient descent in v converges to $\dot{v} = 0$ at finite time, v satisfies $vv^T x = x$, where $x = \sum_{j=1}^{C} \alpha_j x_j$ with positive coefficients α_j and x_j are the C support vectors (see (1)). A solution $v^T = ||x|| x^\dagger$ then *exists* ($x^\dagger$, the pseudoinverse of x, since x is a vector, is given by $x^\dagger = \frac{x^T}{||x||^2}$). On the other hand, the operator T in $v(t+1) = Tv(t)$ associated with equation 19 is non-expanding, because $||v|| = 1$, $\forall t$. Thus there is a fixed point $v \propto x$ which is *independent of initial conditions* (56).

The rates of convergence of the solutions $\rho(t)$ and $v(t)$, derived in different way in (2), may be read out from the equations for ρ and v. It is easy to check that a general solution for ρ is of the form $\rho \propto C \log t$. A similar estimate for the exponential term gives $e^{-\rho v^T x_n} \propto \frac{1}{t}$. Assume for simplicity a single support vector x. We claim that a solution for the error $\epsilon = v - x$, since v converges to x, behaves as $\frac{1}{\log t}$. In fact we write $v = x + \epsilon$ and plug it in the equation for v in 20. We obtain (assuming normalized input $||x|| = 1$)

$$\dot{\epsilon} = \frac{1}{\rho} e^{-\rho v^T x} (x - (x+\epsilon)(x+\epsilon)^T x) \approx \frac{1}{\rho} e^{-\rho v^T x} (x - x - x\epsilon^T - \epsilon x^T), \tag{20}$$

which has the form $\dot{\epsilon} = -\frac{1}{t \log t}(2x\epsilon^T)$. Assuming ϵ of the form $\epsilon \propto \frac{1}{\log t}$ we obtain $-\frac{1}{t \log^2 t} = -B \frac{1}{t \log^2 t}$. Thus the error indeed converges as $\epsilon \propto \frac{1}{\log t}$.

A similar analysis for the weight normalization equations 17 considers the same dynamical system with a change in the equation for v, which becomes

$$\dot{v} \propto e^{-\rho} \rho (I - vv^T) x. \tag{21}$$

This equation differs by a factor ρ^2 from equation 20. As a consequence equation 21 is of the form $\dot{\epsilon} = -\frac{\log t}{t} \epsilon$, with a general solution of the form $\epsilon \propto t^{-\frac{1}{2} \log t}$. In summary, *GD with weight normalization converges faster to the same equilibrium than standard gradient descent: the rate for $\epsilon = v - x$ is $t^{-\frac{1}{2} log(t)}$ vs $\frac{1}{\log t}$.*

Our goal was to find $\lim_{\rho \to \infty} \arg\min_{||V_k||=1, \ \forall k} L(\rho \tilde{f})$. We have seen that various forms of gradient descent enforce different paths in increasing ρ that empirically have different effects on convergence rate. It is an interesting theoretical and practical challenge to find the optimal way, in terms of generalization and convergence rate, to grow $\rho \to \infty$.

Our analysis of simplified batch normalization (1) suggests that several of the same considerations that we used for weight normalization should apply (in the linear one layer case BN is identical to WN). However, BN differs from WN in the multilayer case in several ways, in addition to weight normalization: it has for instance separate normalization for each unit, that is for each row of the weight matrix at each layer.

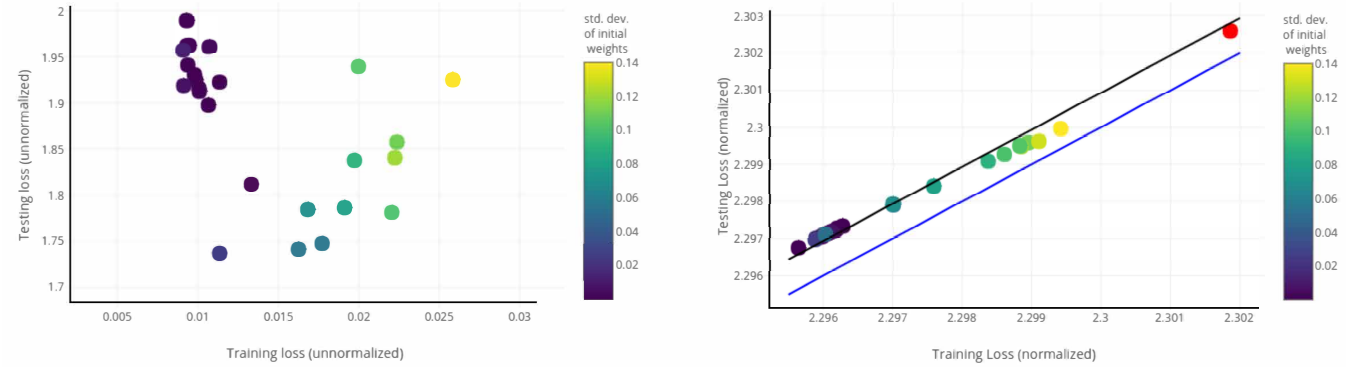

Fig. 3. *The top left graph shows testing vs training cross-entropy loss for networks each trained on the same data sets (CIFAR10) but with a different initializations, yielding zero classification error on training set but different testing errors. The top right graph shows the same data, that is testing vs training loss for the same networks, now normalized by dividing each weight by the Frobenius norm of its layer. Notice that all points have zero classification error at training. The red point on the top right refers to a network trained on the same CIFAR-10 data set but with randomized labels. It shows zero classification error at training and test error at chance level. The top line is a square-loss regression of slope* 1 *with positive intercept. The bottom line is the diagonal at which training and test loss are equal. The networks are 3-layer convolutional networks. The left can be considered as a visualization of Equation 4 when the Rademacher complexity is not controlled. The right hand side is a visualization of the same relation for normalized networks that is* $L(\tilde{f}) \leq \hat{L}(\tilde{f}) + c_1 \mathbb{R}_N(\tilde{\mathbb{F}}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}$. *Under our conditions for* N *and for the architecture of the network the terms* $c_1 \mathbb{R}_N(\tilde{\mathbb{F}}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}$ *represent a small offset. From (55).*

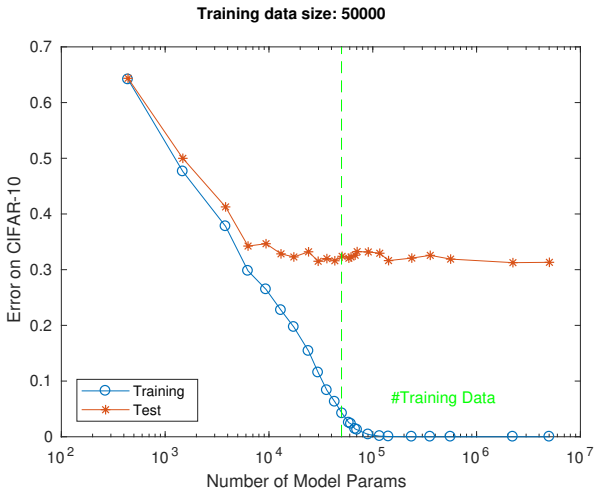

Fig. 4. Empirical and expected error in CIFAR 10 as a function of number of neurons in a 5-layer convolutional network. The expected classification error does not increase when increasing the number of parameters beyond the size of the training set in the range we tested.

4. Discussion

A main difference between shallow and deep networks is in terms of *approximation* power or, in equivalent words, of the ability to learn good representations from data based on the compositional structure of certain tasks. Unlike shallow networks, deep local networks – in particular convolutional networks – can avoid the curse of dimensionality in approximating the class of hierarchically local compositional functions. This means that for such class of functions deep local networks represent an appropriate hypothesis class that allows good approximation with a minimum number of parameters. It is not clear, of course, why many problems encountered in practice should match the class of compositional functions. Though we and others have argued that the explanation may be in either the physics or the neuroscience of the brain, these arguments are not rigorous. Our conjecture at present is that compositionality is imposed by the wiring of our cortex and, critically, is reflected in language. Thus compositionality of some of the most common visual tasks may simply reflect the way our brain works.

Optimization turns out to be surprisingly easy to perform for overparametrized deep networks because SGD will converge with high probability to global minima that are typically much more degenerate for the exponential loss than other local critical points.

More surprisingly, gradient descent yields *generalization* in classification performance, despite overparametrization and even in the absence of explicit norm control or regularization, because standard gradient descent in the weights is subject to an implicit unit (L_2) norm constraint on the *directions of the weights* in the case of exponential-type losses for classification tasks.

In summary, it is tempting to conclude that the practical success of deep learning has its roots in the almost magic syn-

ergy of unexpected and elegant theoretical properties of several aspects of the technique: the deep convolutional network architecture itself, its overparametrization, the use of stochastic gradient descent, the exponential loss, the homogeneity of the RELU units and of the resulting networks.

Of course many problems remain open on the way to develop a full theory and, especially, in translating it to new architectures. More detailed results are needed in approximation theory, especially for densely connected networks. Our framework for optimization is missing at present a full classification of local minima and their dependence on overparametrization. The analysis of generalization should include an analysis of convergence of the weights for multilayer networks (see (4) and (3)). A full theory would also require an analysis of the tradeoff for deep networks between approximation and estimation error, relaxing the separability assumption.

ACKNOWLEDGMENTS. We are grateful to Sasha Rakhlin and Nate Srebro for useful suggestions about the structural lemma and about separating critical points. Part of the funding is from the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216, and part by C-BRIC, one of six centers in JUMP, a Semiconductor Research Corporation (SRC) program sponsored by DARPA.

1. Banburski A, et al. (2019) Theory of deep learning III: Dynamics and generalization in deep networks. *CBMM Memo No. 090*.
2. Soudry D, Hoffer E, Srebro N (2017) The Implicit Bias of Gradient Descent on Separable Data. *ArXiv e-prints*.
3. Lyu K, Li J (2019) Gradient descent maximizes the margin of homogeneous neural networks. *CoRR* abs/1906.05890.
4. Shpigel Nacson M, Gunasekar S, Lee JD, Srebro N, Soudry D (2019) Lexicographic and Depth-Sensitive Margins in Homogeneous and Non-Homogeneous Deep Models. *arXiv e-prints* p. arXiv:1905.07325.
5. Anselmi F, Rosasco L, Tan C, Poggio T (2015) Deep convolutional network are hierarchical kernel machines. *Center for Brains, Minds and Machines (CBMM) Memo No. 35, also in arXiv*.
6. Poggio T, Rosasco L, Shashua A, Cohen N, Anselmi F (2015) Notes on hierarchical splines, dclns and i-theory, (MIT Computer Science and Artificial Intelligence Laboratory), Technical report.
7. Poggio T, Anselmi F, Rosasco L (2015) I-theory on depth vs width: hierarchical function composition. *CBMM memo 041*.
8. Mhaskar H, Liao Q, 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*.
9. Mhaskar H, Poggio T (2016) Deep versus shallow networks: an approximation theory perspective. *Center for Brains, Minds and Machines (CBMM) Memo No. 54, also in arXiv*.
10. Donoho DL (2000) High-dimensional data analysis: The curses and blessings of dimensionality in *AMS CONFERENCE ON MATH CHALLENGES OF THE 21ST CENTURY*.
11. Mhaskar H (1993) Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics* pp. 61–80.
12. Mhaskar HN (1993) Neural networks for localized approximation of real functions in *Neural Networks for Processing [1993] III. Proceedings of the 1993 IEEE-SP Workshop*. (IEEE), pp. 190–196.
13. Chui C, Li X, Mhaskar H (1994) Neural networks for localized approximation. *Mathematics of Computation* 63(208):607–623.
14. Chui CK, Li X, Mhaskar HN (1996) Limitations of the approximation capabilities of neural networks with one hidden layer. *Advances in Computational Mathematics* 5(1):233–243.
15. Pinkus A (1999) Approximation theory of the mlp model in neural networks. *Acta Numerica* 8:143–195.
16. Poggio T, Smale S (2003) The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society (AMS)* 50(5):537–544.
17. Montufar, G. F.and Pascanu R, Cho K, Bengio Y (2014) On the number of linear regions of deep neural networks. *Advances in Neural Information Processing Systems* 27:2924–2932.
18. Livni R, Shalev-Shwartz S, Shamir O (2013) A provably efficient algorithm for training deep networks. *CoRR* abs/1304.7045.
19. Anselmi F, et al. (2014) Unsupervised learning of invariant representations with low sample complexity: the magic of sensory cortex or a new framework for machine learning?. *Center for Brains, Minds and Machines (CBMM) Memo No. 1. arXiv:1311.4158v5*.
20. Anselmi F, et al. (2015) Unsupervised learning of invariant representations. *Theoretical Computer Science*.
21. Poggio T, Rosaco L, Shashua A, Cohen N, Anselmi F (2015) Notes on hierarchical splines, dclns and i-theory. *CBMM memo 037*.
22. Liao Q, 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*.
23. Telgarsky M (2015) Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101v2 [cs.LG] 29 Sep 2015*.
24. Safran I, Shamir O (2016) Depth separation in relu networks for approximating smooth nonlinear functions. *arXiv:1610.09887v1*.
25. Poggio T, Mhaskar H, Rosasco L, Miranda B, Liao Q (2016) Theory I: Why and when can deep - but not shallow - networks avoid the curse of dimensionality, (CBMM Memo No. 058, MIT Center for Brains, Minds and Machines), Technical report.
26. Daubechies I, DeVore R, Foucart S, Hanin B, Petrova G (2019) Nonlinear approximation and (deep) relu networks. *arXiv e-prints* p. arXiv:1905.02199.
27. Jin C, Ge R, Netrapalli P, Kakade SM, Jordan MI (2017) How to escape saddle points efficiently. *CoRR* abs/1703.00887.
28. Ge R, Huang F, Jin C, Yuan Y (2015) Escaping from saddle points - online stochastic gradient for tensor decomposition. *CoRR* abs/1503.02101.
29. Lee JD, Simchowitz M, Jordan MI, Recht B (2016) Gradient descent only converges to minimizers in *29th Annual Conference on Learning Theory*, Proceedings of Machine Learning Research, eds. Feldman V, Rakhlin A, Shamir O. (PMLR, Columbia University, New York, New York, USA), Vol. 49, pp. 1246–1257.
30. Du SS, Lee JD, Tian Y (2018) When is a convolutional filter easy to learn? in *International Conference on Learning Representations*.
31. Tian Y (2017) An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17. (JMLR.org), pp. 3404–3413.
32. Soltanolkotabi M, Javanmard A, Lee JD (2019) Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory* 65(2):742–769.
33. Li Y, Yuan Y (2017) Convergence analysis of two-layer neural networks with relu activation in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17. (Curran Associates Inc., USA), pp. 597–607.
34. Brutzkus A, Globerson A (2017) Globally optimal gradient descent for a convnet with gaussian inputs in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. pp. 605–614.
35. Du S, Lee J, Tian Y, Singh A, Poczos B (2018) Gradient descent learns one-hidden-layer CNN: Don't be afraid of spurious local minima in *Proceedings of the 35th International Conference on Machine Learning*, Proceedings of Machine Learning Research, eds. Dy J, Krause A. (PMLR, Stockholmsmässan, Stockholm Sweden), Vol. 80, pp. 1339–1348.
36. Du SS, Lee JD, Li H, Wang L, Zhai X (2018) Gradient descent finds global minima of deep neural networks. *CoRR* abs/1811.03804.
37. Zhong K, Song Z, Jain P, Bartlett PL, Dhillon IS (2017) Recovery guarantees for one-hidden-layer neural networks in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17. (JMLR.org), pp. 4140–4149.
38. Zhong K, Song Z, Dhillon IS (2017) Learning non-overlapping convolutional neural networks with multiple kernels. *CoRR* abs/1711.03440.
39. Zhang X, Yu Y, Wang L, Gu Q (2018) Learning One-hidden-layer ReLU Networks via Gradient Descent. *arXiv e-prints*.
40. Li Y, Liang Y (2018) Learning overparameterized neural networks via stochastic gradient descent on structured data in *Advances in Neural Information Processing Systems 31*, eds. Bengio S, et al. (Curran Associates, Inc.), pp. 8157–8166.
41. Du SS, Zhai X, Poczos B, Singh A (2019) Gradient descent provably optimizes over-parameterized neural networks in *International Conference on Learning Representations*.
42. Zou D, Cao Y, Zhou D, Gu Q (2018) Stochastic gradient descent optimizes over-parameterized deep relu networks. *CoRR* abs/1811.08888.
43. Poggio T, Liao Q (2017) Theory II: Landscape of the empirical risk in deep learning. *arXiv:1703.09833, CBMM Memo No. 066*.
44. Zhang C, et al. (2017) Theory of deep learning IIb: Optimization properties of SGD. *CBMM Memo 072*.
45. Raginsky M, Rakhlin A, Telgarsky M (2017) Non-convex learning via stochastic gradient langevin dynamics: A nonasymptotic analysis. *arXiv:180.3251 [cs, math]*.
46. Daniely A (2017) Sgd learns the conjugate kernel class of the network in *Advances in Neural Information Processing Systems 30*, eds. Guyon I, et al. (Curran Associates, Inc.), pp. 2422–2430.
47. Allen-Zhu Z, Li Y, Liang Y (2018) Learning and generalization in overparameterized neural networks, going beyond two layers. *CoRR* abs/1811.04918.
48. Arora S, Du SS, Hu W, yuan Li Z, Wang R (2019) Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *CoRR* abs/1901.08584.
49. Wei C, Lee JD, Liu Q, Ma T (2018) On the margin theory of feedforward neural networks. *CoRR* abs/1810.05369.
50. Liang T, Poggio T, Rakhlin A, Stokes J (2017) Fisher-rao metric, geometry, and complexity of neural networks. *CoRR* abs/1711.01530.
51. Bousquet O, Boucheron S, Lugosi G (2003) Introduction to statistical learning theory. pp. 169–207.
52. Rosset S, Zhu J, Hastie T (2003) Margin maximizing loss functions in *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*. pp. 1237–1244.
53. Douglas SC, Amari S, Kung SY (2000) On gradient adaptation with unit-norm constraints. *IEEE Transactions on Signal Processing* 48(6):1843–1847.
54. Salimans T, Kingm DP (2016) Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in Neural Information Processing Systems*.
55. Liao Q, Miranda B, Banburski A, Hidary J, Poggio TA (2018) A surprising linear relationship predicts test performance in deep networks. *CoRR* abs/1807.09659.
56. Ferreira PJSG (1996) The existence and uniqueness of the minimum norm solution to certain linear and nonlinear problems. *Signal Processing* 55:137–139.