CBMM Memo No. 100 **November 19, 2019**

Theoretical Issues In Deep Networks

Tomaso Poggio, Andrzej Banburski, Qianli Liao
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 theoretical characterization of deep learning should answer questions about their approximation power, the dynamics of optimization by gradient descent and good out-of-sample performance --- why the expected error does not suffer, despite the absence of explicit regularization, when the networks are overparametrized. We review our recent results towards this goal. In {\it approximation theory} both shallow and deep networks are known to approximate any continuous functions on a bounded domain at a cost which is exponential (the number of parameters is exponential in the dimensionality of the function). However, we proved that for certain types of compositional functions, deep networks of the convolutional type (even without weight sharing) can have a linear dependence on dimensionality, unlike shallow networks. In characterizing {\it minimization} of the empirical exponential loss we consider the gradient descent dynamics of the weight directions rather than the weights themselves, since the relevant function underlying classification corresponds to the normalized network. The dynamics of the normalized weights implied by standard gradient descent turns out to be equivalent to the dynamics of the constrained problem of minimizing an exponential-type loss subject to a unit L_2 norm constraint. In particular, the dynamics of the typical, unconstrained gradient descent converges to the same critical points of the constrained problem. Thus, there is {\it implicit regularization} in training deep networks under exponential-type loss functions with gradient descent. The critical points of the flow are hyperbolic minima (for any long but finite time) and minimum norm minimizers (e.g. maxima of the margin). Though appropriately normalized networks can show a small generalization gap (difference between empirical and expected loss) even for finite N (number of training examples) wrt the exponential loss, they do not generalize in terms of the classification error. Bounds on it for finite N remain an open problem. Nevertheless, our results, together with other recent papers, characterize an implicit vanishing regularization by gradient descent which is likely to be a key prerequisite -- in terms of complexity control -- for the good performance of deep overparametrized ReLU classifiers.

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

Theoretical Issues in Deep Networks

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

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

This manuscript was compiled on November 24, 2019

While deep learning is successful in a number of applications, it is not yet well understood theoretically. A theoretical characterization of deep learning should answer questions about their approximation power, the dynamics of optimization by gradient descent and good out-of-sample performance — why the expected error does not suffer, despite the absence of explicit regularization, when the networks are overparametrized. We review our recent results towards this goal. In *approximation theory* **both shallow and deep networks are known to approximate any continuous functions on a bounded domain at a cost which is exponential (the number of parameters is exponential in the dimensionality of the function). However, we proved that for a certain types of compositional functions, deep networks of the convolutional type (even without weight sharing) can have a linear dependence on dimensionality, unlike shallow networks. In characterizing** *minimization* **of the empirical exponential loss we consider the gradient descent dynamics of the weight directions rather than the weights themselves, since the relevant function underlying classification corresponds to the normalized network. The dynamics of the normalized weights implied by standard gradient descent turns out to be equivalent to the dynamics of the constrained problem of minimizing an exponential-type loss subject to a unit L_2 norm constraint. In particular, the dynamics of the typical, unconstrained gradient descent converges to the same critical points of the constrained problem. Thus there is** *implicit regularization* **in training deep networks under exponential-type loss functions with gradient descent. The critical points of the flow are hyperbolic minima (for any long but finite time) and minimum norm minimizers (e.g. maxima of the margin). Though appropriately normalized networks can show a small generalization gap (difference between empirical and expected loss) even for finite N (number of training examples) wrt the exponential loss, they do not generalize in terms of the classification error. Bounds on it for finite N remain an open problem. Nevertheless, our results, together with other recent papers (1–4), characterize an implicit vanishing regularization by gradient descent which is likely to be a key prerequisite – in terms of complexity control – for the good performance of deep overparametrized ReLU classifiers.**

Machine Learning | Deep learning | Approximation | Optimization | Generalization

1. Introduction

A satisfactory theoretical characterization of deep learning should begin by addressing several questions that are natural in the area of machine learning techniques based on empirical risk minimization (see for instance (5), (6). They include issues such as the approximation power of deep networks, the dynamics of the empirical risk minimization by gradient descent and the 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 paper we review briefly our work on approximation and describe recent results in characterizing complexity control in training deep networks. The paper is organize in two separate parts. The first, about approximation power of deep versus shallow architecture, is a review of recent papers. The second, about optimization and generalization, describes some of our recent results(4) mostly characterizing empirical risk optimization and in particular properties of the dynamical system induced by gradient descent. One of these results, about the equivalence of constrained and unconstrained optimization, implies a classical form of complexity control at any finite times by gradient descent with respect to the exponential loss. We begin by summarizing a number of useful definitions and properties.

A. 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 $f(W;x) = \sigma(W^K\sigma(W^{K-1}\cdots\sigma(W^1x)))$, 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

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. We describe a review of our work on deep learning addressing the following questions: 1) *approximation power* of deep networks 2) *dynamics of optimization* of the empirical risk by gradient descent 3) *complexity control* properties of gradient descent techniques – how can deep networks predict well despite the absence of any explicit regularization? We describe results showing that for a class of compositional functions deep networks of the convolutional type are exponentially better approximators than shallow networks; gradient descent induces a dynamics of the normalized weights that corresponds to vanishing regularization; for any finite time minima are hyperbolic; in this regime there is a (hidden) norm control in the minimization of exponential-type losses by gradient descent that guarantees complexity control by the normalized network despite overparametrization; asymptotic convergence is to a minimum norm minimizer of the loss.

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

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 (7)) $\sum_{i,j} W_k^{i,j} \left(\frac{\partial f(x)}{\partial W_k^{i,j}} \right) = f(x)$ where W_k is here the vectorized representation of the weight matrices W_k for layer k.

For the network, homogeneity of the ReLU 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 $\rho = \prod_{k=1}^{K} \rho_k$, $\mathbb{F}$ 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.

We will assume that for some $t > T_0$ gradient descent realizes a f that separates the data that is $f(x_n) y_n > 0 \quad \forall n = 1, \cdots, N$. Under this assumption, we will sometime write $f(x_n) > 0$ as a shorthand for $y_n f(x_n) > 0$.

2. Approximation

We start with the first set of questions, summarizing results in (8–10), and (11, 12). 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 a certain type of *compositional functions*, that is functions of functions. An especially interesting subset of compositional functions are the ones that can be written as *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 results 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 (13)) correspond to input-output mappings that are *compositional 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.

A. 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 (14–18)). By now, several papers (19–21) are available. We review (11, 22–25) which derive 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 by Telgarsky (26) (see also (27, 28) and, earlier (17)). An interesting review of approximation of univariate functions by deep ReLU networks has recently appeared (29).

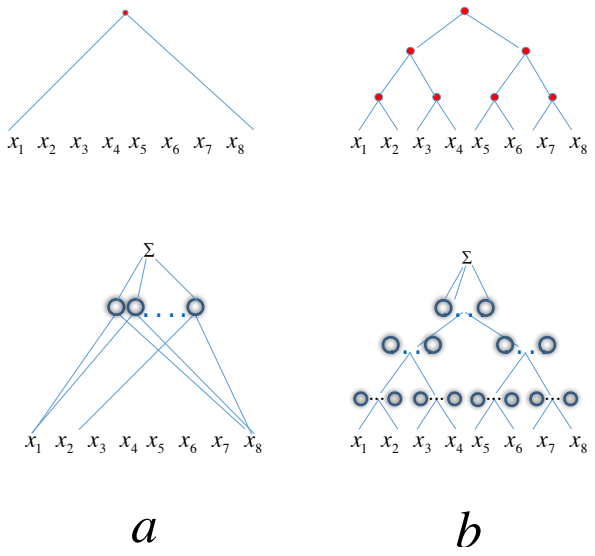

Fig. 1. The top graphs are associated to *functions*; each of the bottom diagrams depicts the ideal *network* approximating the function above. In a) a shallow universal network in 8 variables and N units approximates a generic function of 8 variables $f(x_1, \cdots, x_8)$. Inset b) shows a hierarchical 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 today's deep networks, but equivalent to them (28)). 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 (12).

B. 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. Notice, however, that from the point of view of machine learning, the relevant norm is the L_2 norm. In this sense, several of our results are stronger than needed. Yet our main results on compositionality require the sup norm in order to be independent from the unknown distribution of the input data. This is important for machine learning.

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 approximation 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 ; d corresponds to the size of the kernel in a convolutional network).

C. Shallow and deep networks. This section characterizes conditions under which deep networks are "better" than shallow network in approximating functions, 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 (28)).

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. (28) 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 (9–11)). 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 (28). 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(28).

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 $W_m^{n,d}$.

3. Optimization and complexity control

It has been known for a long time that the key to predictivity in machine learning is controlling the complexity of the network and not simply the raw number of its parameters. This is usually done during optimization by imposing a constraint, often under the form of a regularization penalty, on the norm of the weights, since relevant complexity measures, such the Rademacher complexity, depend on it. The problem is that there is no obvious control of complexity in the training of deep networks!

Recent results by (1) illuminate the apparent absence of "overfitting" (see Figure 3) 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. Our main results are about the dynamics of the normalized network $\tilde{f}$ under gradient flow, its convergence to a maximum margin solution and its generalization properties. We first outline related work. We then describe our main result that consists of several steps, summarized in Theorem 3.

A. Related work. A number of papers have studied gradient descent for deep networks (30–32). Close to the approach summarized here (details are in (4)) is the paper (33). 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 ((3) and (2)) develop an elegant margin maximization based approach which lead to some of the same results of this section (and many more). Our approach does not need the notion of maximum margin but our theorem on margin establishes a connection with it and thus with the results of (3) and (2). Our main goal here (and in (4)) is to achieve a simple understanding of where the complexity control underlying predictivity and generalization is hiding in the training of deep networks. We define generalization as the convergence of the empirical loss of the empirical minimizer to its expected loss for the number of data points growing to infinity. Recent results on regression mostly for linear or quasi-linear networks (34–36) suggests an implicit regularization mechanism somewhat similar to the regularizing effect of the gradient descent iterations (37).Its limit, however, does not enforce a norm constraint unlike the classification case.

B. Main results on the dynamics of optimization. The standard approach to training deep networks is to use stochastic gradient descent to find the weights W_k that minimize the empirical exponential loss $L = \sum_n e^{-y_n f(x_n)}$ by computing

$$\dot{W}_k = -\frac{\partial L}{\partial W k} = \sum_{n=1}^{N} y_n \frac{\partial f(W; x_n)}{\partial W_k} e^{-y_n f(W; x_n)} \qquad [3]$$

on a given dataset $\{x_i, y_i\} \quad \forall i = 1, \ldots, N$ with y binary. Since the goal is binary classification, we are interested in $\tilde{f}$ (remember sign $\tilde{f}$ = sign f. We want to study the dynamics of $\tilde{f}$ implied by Equation 3. With this goal we study three related versions of this problem:

1. minimization of $L = \sum_n e^{-\rho \tilde{f}(x_n)}$ under the constraint $||V_k|| = 1$ wrt V_k for fixed ρ;
2. minimization of $L = \sum_n e^{-\rho \tilde{f}(x_n)}$ under the constraint $||V_k|| = 1$ wrt V_k and ρ;
3. minimization of $L = \sum_n e^{-\rho \tilde{f}(x_n)} = \sum_n e^{-f(x_n)}$ wrt V_k, ρ, which is equivalent to typical training, Equation 3.

B.1. Constrained minimization of the exponential loss. Constrained optimization of the exponential loss minimizes $L = \sum_n e^{-\rho \tilde{f}(x_n)}$ under the constraint $||V_k|| = 1$ which leads to minimize

$$L = \sum_n e^{-\rho \tilde{f}(x_n)} + \sum_k \lambda_k ||V_k||^2 \qquad [4]$$

with λ_k such that the constraint $||V_k| = 1$ is satisfied. We note that this would be the natural approach for training deep networks while controlling complexity based on classical generalization bounds (see (4)).

B.2. Fixed ρ: hyperbolic minima. Gradient descent on L for fixed ρ wrt V_k yields then the dynamical system

$$\dot{V}_k = \rho \sum_n e^{-\rho \tilde{f}(x_n)} \left(\frac{\partial \tilde{f}(x_n)}{\partial V_k} - V_k \tilde{f}(x_n) \right) \qquad [5]$$

because $\lambda_k = \frac{1}{2} \rho \sum_n e^{-\rho \tilde{f}(x_n)} \tilde{f}(x_n)$, since $V_k^T \dot{V}_k = 0$ because $||V_k||^2 = 1$.

Since for fixed ρ the domain is compact, stationary points $\dot{V}_k = 0$ of the constrained optimization problem must exist. Assuming data separation is achieved (that is $y_n \tilde{f}(x_n) > 0 \quad \forall n$), they satisfy

$$\sum_n e^{-\rho \tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k} = \sum_n e^{-\rho \tilde{f}(x_n)} V_k \tilde{f}(x_n). \qquad [6]$$

The stationary points provided by Equations 6 are in fact *hyperbolic minima* because the Hessian of L (and Jacobian of $\dot{V}_k = -F(V_k, \rho_k) = -\nabla_{V_k} L$) is negative definite at the stationary points. Thus the sufficient conditions for local minima are satisfied. Of course the minimum of the exponential loss L is only zero for the limit $\rho = \infty$; for any finite ρ the minimum of L is at the boundary of the compact domain. For any finite, sufficiently large ρ the minimum is hyperbolic but in general is not unique (it is unique only for linear networks).

The Hessian is

$$\sum_n \left[-\rho^2 \frac{\partial \tilde{f}(x_n)}{\partial V_k} \frac{\partial \tilde{f}(x_n)}{\partial V_{k'}}^T + \rho \frac{\partial^2 \tilde{f}(x_n)}{\partial V_k \partial V_{k'}} \right] e^{-\rho \tilde{f}(x_n)} - 2\lambda(\rho)\mathbf{I}. \qquad [7]$$

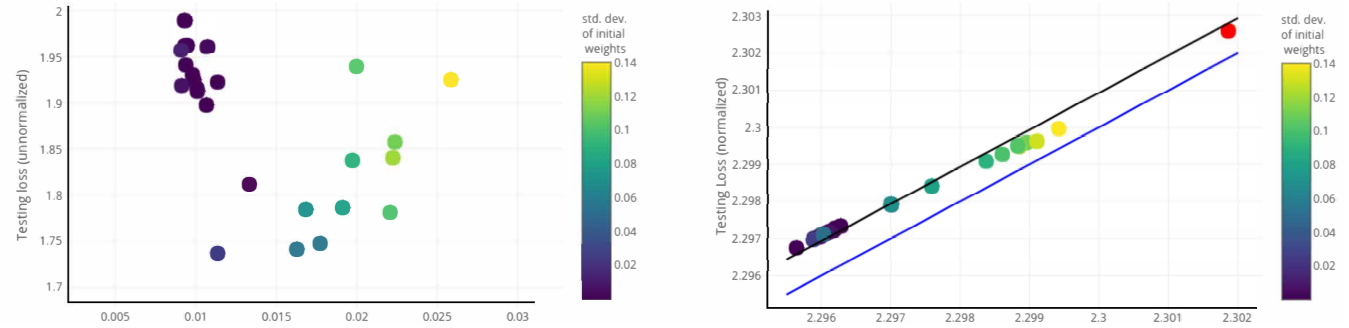

Fig. 2. *Empirical evidence of generalization by normalized networks with respect to the cross entropy loss. The left graph shows testing vs training cross-entropy loss for networks each trained on the same data sets (CIFAR10) but with different initializations, yielding zero classification error on training set but different testing errors. The 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 CIFAR10 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 generalization bounds when the Rademacher complexity is not controlled. The right hand side is a visualization of the same relation for normalized networks that is* $L(\rho\tilde{f}) \leq \hat{L}(\rho\tilde{f}) + c_1\rho\mathbb{R}_N(\tilde{\mathbb{F}}) + c_2\sqrt{\ln(\frac{1}{\delta})/2N}$ *for* $\rho = 1$*. Under our conditions for* N *and for the architecture of the network the terms* $c_1\mathbb{R}_N(\tilde{\mathbb{F}}) + c_2\sqrt{\ln(\frac{1}{\delta})/2N}$ *represent a small offset. Notice that the exponential loss is not a better proxy for the classification loss for* $\rho > 1$*. Empirically for these data sets, the exponential loss with* $\rho = 1$ *provides a ranking that agrees with the classification ranking. From (38).*

In (4) we consider the quadratic form $\sum_{k,k'} V_k^T H V_{k'}$, where V_k correspond to critical points of the gradient and prove the following

Lemma 1 *For sufficiently late times after data separation* $(\forall n, \tilde{f}(x_n) > 0)$*, H is negative definite for any finite* ρ *such that* $\rho^2 K^2(\tilde{f}(x_n))^2 > K(K-2)\rho\tilde{f}(x_n)$*, where K is the number of layers. This only requires* $\rho\tilde{f}(x_n) > 1$*. H becomes negative semi-definite in the limit* $\rho = \infty$*.*

B.3. $\rho \to \infty$ ***has same stationary points as the full dynamical system.*** Consider the limit of $\rho \to \infty$ in Equation 6 . The asymptotic stationary points of the flow of V_k then satisfy

$$\sum_n e^{-\rho\tilde{f}(x_n)}(\frac{\partial\tilde{f}(x_n)}{\partial V_k} - V_k\tilde{f}(x_n)) = 0 \qquad [8]$$

also in the limit $\lim_{\rho\to\infty}$, that is for any large ρ. So the stationary V_k points for any large $\rho = R$ satisfies

$$\sum_n e^{-R\tilde{f}(x_n)}(\frac{\partial\tilde{f}(x_n)}{\partial V_k} - V_k\tilde{f}(x_n)) = 0. \qquad [9]$$

Consider now gradient descent for the full system obtained with Lagrange multipliers, that is, on $L = \sum_n e^{-\rho\tilde{f}(x_n)} + \sum_k \lambda_k||V_k||^2$ wrt V_k *and* ρ_k, with λ_k chosen (as before) to implement the unit norm constraint. The full gradient dynamical system is

$$\dot{\rho_k} = \frac{\rho}{\rho_k}\sum_n e^{-\rho\tilde{f}(x_n)}\tilde{f}(x_n) \qquad [10]$$

$$\dot{V_k} = \rho\sum_n e^{-\rho\tilde{f}(x_n)}(\frac{\partial\tilde{f}(x_n)}{\partial V_k} + V_k\tilde{f}(x_n)). \qquad [11]$$

Observe that after onset of separability $\dot{\rho_k} > 0$ with $\lim_{t\to\infty}\dot{\rho_k} = 0$, $\lim_{t\to\infty}\rho(t) = \infty$ (for one layer $\rho \propto \log t$ as shown in a later section). Thus $\rho(t)$ is a monotonically increasing function from t_0 to $t = \infty$. Furthermore, ρ_k grows at a rate which is independently of the layer k. In fact, Equation 3 via the relation $||\dot{W_k}|| = \frac{\partial||W_k||}{\partial W_k}\frac{\partial W_k}{\partial t} = \frac{W_k}{||W_k||}\dot{W_k}$ implies

$$||\dot{W_k}||^2 = 2\sum_{n=1}^{N} y_n f(W;x_n)e^{-y_n f(x_n;W)}, \qquad [12]$$

which shows that the rate of growth of $||W_k||^2$ is independent of k. This observation (39) is summarized by

Lemma 2 *During gradient descent, the rate of change of the squares of the Frobenius norms of the weights is the same for each layer, that is* $||\dot{\rho_k}||^2 = 2\sum_{n=1}^{N} y_n\rho\tilde{f}(x_n)e^{-y_n\rho\tilde{f}(x_n)}$*.*

Because $\rho(t)$ grows monotonically in t for any large R in Equation 9, there exist T such that $\rho(T) = R$. At time T then, the condition for a stationary point of V_k in Equation 11 coincides with Equation 8. This leads to

Lemma 3 *The full dynamical system 11 in the limit of* $t \to \infty$ *converges to the same limit to which the dynamical system Equation 5 converges for* $\rho \to \infty$*.*

B.4. Asymptotic stationary points coincide with maximum margin. Here we show that the limit for the two systems exists, is not trivial and corresponds to maximum margin/minimum norm solutions. First notice that we can write

$$\sum_n^N e^{-R\tilde{f}(x_n)}(\frac{\partial\tilde{f}(x_n)}{\partial V_k} - V_k\tilde{f}(x_n)) = 0. \qquad [13]$$

We assume without loss of generality that $\tilde{f}(x_N) = min_n \tilde{f}(x_n)$, we define $H_n = (\frac{\partial \tilde{f}(x_n)}{\partial V_k} - V_k \tilde{f}(x_n))$, and write

$$e^{-R\tilde{f}(x_N)}[H_N + e^{-R\Delta_{min}} \sum_n^{N-1} H_n] \geq$$

$$e^{-R\tilde{f}(x_N)} H_N + \sum_n^{N-1} e^{-R\tilde{f}(x_n)} H_n = \sum_n^N e^{-R\tilde{f}(x_n)} H_n \geq \quad [14]$$

$$e^{-R\tilde{f}(x_N)} H_N \geq e^{-R\tilde{f}(x_N)}[H_N + e^{-R\Delta_{max}} \sum_n^{N-1} H_n],$$

where $\Delta_n = \tilde{f}(x_n) - \tilde{f}(x_N)$ with $\tilde{f}(x_N) = min_n \tilde{f}(x_n)$ as the margin of $\tilde{f}$ and $\Delta_{min} = \min_{n \neq N} \Delta_n$, $\Delta_{max} = \max_n \Delta_n$ [*].

The left hand side of the stationary point equation has the form $\epsilon(H_N + \epsilon' H) = 0$, with $H = \sum_n^{N-1} H_n$, $\epsilon = e^{-R\tilde{f}(x_N)}$ and $\epsilon' = e^{-R\Delta_{min}}$; for decreasing $\epsilon > 0$, there will be an $\epsilon^* > 0$ for which the equation is satisfied by $H_N = 0$ before it is trivially satisfied at the limit $\epsilon = 0$. Thus the stationarity condition for large but finite ρ is $\frac{\partial \tilde{f}(x_*)}{\partial V_k} - V_k \tilde{f}(x_*) = 0$, that is the condition in which the stationary point x_N provides the maximum margin. Before that limit is reached, the solution V_k changes with increasing ρ. Thus the asymptotic stationary points coincide with maximum margin. The following lemma (4) shows that the margin is increasing for sufficiently late t:

Lemma 4 *There exists a R such that for $\rho > R$ the sum $\sum_{n=1}^N e^{-\rho \tilde{f}(x_n)} \propto e^{-\rho \tilde{f}(x_*)}$. For $\rho > R$ then the margin increases $\frac{\partial \tilde{f}}{\partial t} \geq 0$ (even if ρ is kept fixed).*

We finally note that the maximum margin solution in terms of $\tilde{f}$ and V_k is equivalent to a minimum norm solution in terms of W_k under the condition of the margin being at least 1. This is stated in the following lemma (see (4)):

Lemma 5 *The maximum (max) margin problem*

$$\max_{W_k} \min_{i=1,\ldots,N} y_i f(W; x_i), \qquad \textit{subj. to} \quad \|W\| = 1. \quad [15]$$

is equivalent to

$$\min_W \frac{1}{2}\|W\|^2, \quad \textit{subj. to} \quad y_i f(W; x_i) \geq 1, \; i = 1, \ldots, N. \quad [16]$$

B.5. Typical gradient descent for deep networks: implicit norm control. Empirically it appears that GD and SGD converge to solutions that can generalize even without any explicit capacity control such as a regularization term or a constraint on the norm of the weights. How is this possible? The answer is provided by the fact – trivial or surprising – that the unit vector $\frac{w(T)}{||w(T)||_2}$ computed from the solution $w(T)$ of gradient descent $\dot{w} = -\nabla_w L$ at time T is the same, irrespectively of whether the constraint $||v||_2 = 1$ is enforced during gradient descent. This confirms Srebro results for linear networks and throws some light on the nature of the implicit bias or hidden complexity control. We show this result next.

We study the new dynamical system induced by the dynamical system in $\dot{W_k}^{i,j}$ under the reparametrization $W_k^{i,j} = \rho_k V_k^{i,j}$ with $||V_k||_2 = 1$. This is equivalent to changing coordinates from W_k to V_k and $\rho_k = ||W_k||_2$. For simplicity of notation we consider here for each weight matrix V_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 $\frac{w}{\rho} = v$; thus $w = \rho v$ with $||v||_2 = 1$ and $\rho = ||w||_2$. The following relations are easy to check:

1. Define $S = I - vv^T = I - \frac{ww^T}{||w||_2^2}$; $\frac{\partial v}{\partial w} = \frac{S}{\rho}$.
2. $Sw = Sv = 0$ and $S^2 = S$
3. In the multilayer case $\frac{\partial f(x_n;W)}{\partial W_k} = \frac{\rho}{\rho_k} \frac{\partial \tilde{f}(V;x_n)}{\partial V_k}$

The unconstrained gradient descent dynamic system used in training deep networks for the exponential loss is given in Equation 3, that is

$$\dot{W}_k = -\frac{\partial L}{\partial W k} = \sum_{n=1}^N y_n \frac{\partial f(W; x_n)}{\partial W_k} e^{-y_n f(W;x_n)}. \quad [17]$$

Following the chain rule *for the time derivatives*, the dynamics for W_k induces the following dynamics for $||W_k|| = \rho_k$ and V_k:

$$\begin{aligned} \dot{\rho_k} &= \frac{\partial ||W_k||}{\partial W_k} \frac{\partial W_k}{\partial t} = V_k^T \dot{W_k} \\ \dot{V_k} &= \frac{\partial V_k}{\partial W_k} \frac{\partial W_k}{\partial t} = \frac{S_k}{\rho_k} \dot{W_k} \end{aligned} \quad [18]$$

where $S_k = I - V_k V_k^T$. We now obtain the time derivatives of V_k and ρ_k from the time derivative of W_k; the latter is computed from the gradients of L with respect to W_k that is from the gradient dynamics of W_k. Thus *unconstrained gradient descent* coincides with the following dynamical system

$$\dot{\rho_k} = \sum_{n=1}^N V_k^T \frac{\partial f(x_n; W)}{\partial W_k} e^{-f(x_n;W)} = \frac{\rho}{\rho_k} \sum_{n=1}^N e^{-\rho \tilde{f}(x_n)} \tilde{f}(x_n)$$

$$\dot{V_k} = \frac{\rho}{\rho_k^2} \sum_{n=1}^N e^{-\rho \tilde{f}(x_n)} (\frac{\partial \tilde{f}(x_n)}{\partial V_k} - V_k \tilde{f}(x_n)).$$

where we use the structural lemma to write $V_k^T \frac{\partial \tilde{f}(x_n)}{\partial V_k} = \tilde{f}(x_n)$.

Clearly the dynamics of *unconstrained gradient descent* and the dynamics of *constrained gradient descent* are very similar since they differ only by a ρ^2 factor in the $\dot{v}$ equations. The conditions for the stationary points of the gradient for the v vectors – that is the values for which $\dot{v} = 0$ – are *the same in both cases*, since for any $t > 0$ we have $\rho(t) > 0$. This is summarized in

Lemma 6 *Constrained and unconstrained gradient descent have the same minima which are hyperbolic for finite times and asymptotically degenerate.*

Since there are multiple minima and the trajectories depend on the dynamics, the statement above does not mean that even with the same initial conditions the two dynamics will converge to the same minima among the multiple ones. In (4) we show that the dynamics of the constrained gradient descent Equation 11 is the same as the algorithm called "weight normalization" (40).

[*] We consider only distinct "support vectors", aggregating together data points with the same margin, thus $\Delta_{min} > 0$

B.6. Complexity control in unconstrained gradient descent. Actual convergence for the constrained case happens after $(\frac{\partial \tilde{f}(x_*)}{\partial V_k} - V_k \tilde{f}(x_*)) = 0$ is valid, which corresponds to a long but finite time and thus a large but finite ρ and a small but non-zero λ. In conclusion, the solution corresponds to solving a regularization problem with a non-zero λ. Beta-stability and asymptotic generalization are guaranteed(41), as long as λ implied by the stopping time satisfies $\frac{1}{\lambda N} << 1$. Since the unconstrained case converges to the same solutions *it will also generalize.* There are obvious limitations in this statement: though the normalized network $\tilde{f}$ generalizes under the exponential loss, bounds on the classification error remain an open problem. In fact, the classification error does not show generalization for datasets such as CIFAR10 when the number of training data is significantly less than the number of parameters: the training error is zero while the test error is not (see Figure 3). As we will discuss later, deep overparametrized nets may show a regime, in which there is simultaneously interpolation of the training data and good expected error . For this regime, the implicit regularization described here is a prerequisite for a full explanation.

C. Additional results.

C.1. Dynamics of ρ. In linear 1-layer networks the dynamics of gradient descent yield $\rho \sim \log t$ asymptotically. In the K-layer case the nonlinearly coupled equations are not easily solved analytically. (4) gives an approximation of the form

$$\dot{R} = \tilde{f}(x)^{\frac{2}{K}} K^2 (\log R)^{2-\frac{2}{K}}, \qquad [19]$$

where $R = e^{\rho_k \tilde{f}(x)}$. We can check that for $K = 1$ we get $R \sim t$, so $\rho \sim \log t$. It is also immediately clear that for $K > 1$ the product of weights diverges faster than logarithmically. In the case of $K = 2$ we get $R(t) = \mathrm{li}^{-1}(\tilde{f}(x)K^2 t + C)$, where $\mathrm{li}(z) = \int_0^z dt/\log t$ is the logarithmic integral function. For larger K we get faster divergence, with the limit $K \to \infty$ given by $R(t) = \mathcal{L}^{-1}(\alpha_\infty t + C)$, where $\alpha_\infty = \lim_{K\to\infty} \tilde{f}(x)^{\frac{2}{K}} K^2$ and $\mathcal{L}(z) = \mathrm{li}(z) - \frac{z}{\log z}$. Interestingly, while the product of weights scales faster than logarithmically, the weights at each layer diverge slower than in the 1-layer case.

C.2. Landscape and minima. ReLU networks with exponential-type loss functions do not have zeros of the gradient (wrt the W_k) that separate the data. The stationary points of the gradient of f in the nonlinear multilayer separable case under exponential loss are given by $\sum_{n=1}^{N} y_n \frac{\partial f(x_n;w)}{\partial W_k^{i,j}} e^{-y_n f(x_n;W)} = 0$. Thus, the only stationary points of the gradient that separate the data are for $\rho = \infty$. If other stationary points were to exist for a value W^* of the weights, they would be given by zero-valued linear combinations with positive coefficients of $\frac{\partial f(x_n;w)}{\partial W_k^{i,j}}$. Use of the structural lemma shows that $\frac{\partial f(x;w)}{\partial W_k^{i,j}} = 0, \forall i,j,k$ implies $f(W^*;x) = 0$. So stationary points of the gradient wrt W_k that are data-separating do not exist for any finite ρ. The situation is quite different if we consider *stationary points wrt* V_k. Notice that minima arbitrarily close to zero loss exist for any finite, large ρ. For $\rho \to \infty$, the Hessian becomes arbitrarily close to zero, with all eigenvalues close to zero. On the other hand, any point of the loss at a finite ρ has a Hessian wrt W_k which is not identically zero.

Clearly, it would be interesting to characterize better the degeneracy of the local minima. For the goals of this section however the fact that they cannot be completely degenerate is sufficient. (4) shows that *under the exponential loss, the weight W_k for zero loss at infinite ρ are completely degenerate, with all eigenvalues of the Hessian being zero. The other stationary points of the gradient are less degenerate, with at least one nonzero eigenvalue.*

C.3. Linear networks and rates of convergence. The linear $f(x) = \rho v^T x$ networks case (1) 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

$$\dot{\rho} = \sum_{n=1}^{N} \frac{e^{-\rho v^T x_n}}{\rho} v^T x_n \quad \dot{v} = \sum_{n=1}^{N} \frac{e^{-\rho v^T x_n}}{\rho} (x_n - v v^T x_n). \qquad [20]$$

If gradient descent in v converges to $\dot{v} = 0$ at finite time, v satisfies $v v^T x = x$, where $x = \sum_{j=1}^{C} \alpha_j x_j$ with positive coefficients α_j and x_j are the C support vectors (see (4)). 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 20 is non-expanding, because $||v|| = 1$, $\forall t$. Thus in the linear case there is a unique fixed point $v \propto x$ which is *independent of initial conditions* (42).

The rates of convergence of the solutions $\rho(t)$ and $v(t)$, derived in different way in (1), 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 21. We obtain (assuming normalized input $||x|| = 1$)

$$\dot{\epsilon} = \frac{e^{-\rho v^T x}}{\rho} (x - (x+\epsilon)(x+\epsilon)^T x) \approx -\frac{e^{-\rho v^T x}}{\rho} (x\epsilon^T + \epsilon x^T), \qquad [21]$$

which has the form $\dot{\epsilon} = -\frac{1}{t \log t}(2x\epsilon^T)$. This indeed has the error converging as $\epsilon \propto \frac{1}{\log t}$.

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

$$\dot{v} \propto e^{-\rho} \rho (I - v v^T) x. \qquad [22]$$

This equation differs by a factor ρ^2 from equation 21. As a consequence equation 22 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 $\frac{1}{\log t}$.*

The linear case shows that different forms of gradient descent enforce different paths in increasing ρ that 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 ρ from 0 to ∞.

D. Summary. The following theorem (informal statement) summarizes our main results on minimizing the exponential loss in deep ReLU networks.

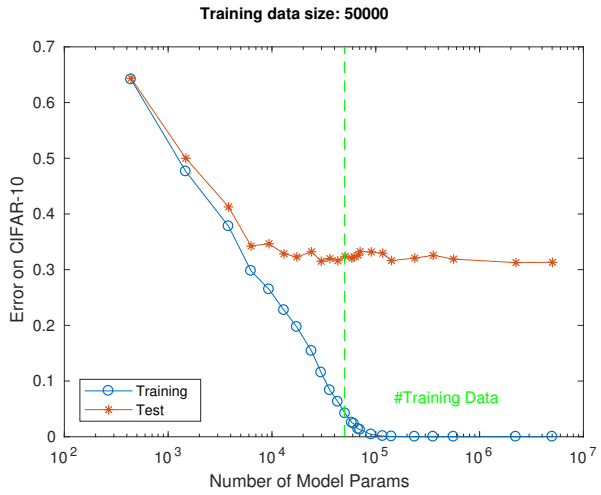

Fig. 3. Empirical and expected error in CIFAR 10 as a function of number of neurons in a 5-layer convolutional network. The expected classification error does not increase when increasing the number of parameters beyond the size of the training set.

Theorem 3 *Assume that separability is reached at time T_0 during gradient descent on the exponential loss, that is $y_n f(x_n) > 0, \quad \forall n$. Then unconstrained gradient descent converges in terms of the normalized weights to a solution that is under complexity control for any finite time. In addition the following properties hold:*

1. *Consider the dynamics (A) resulting from using Lagrange multipliers on the constrained optimization problem: "minimize $L = \sum_n e^{-\rho \tilde{f}(x_n)}$ under the constraint $||V_k|| = 1$ wrt V_k". The dynamics converges for any fixed ρ to stationary points of the V_k flow that are hyperbolic minima.*
2. *Consider the dynamics (B) resulting from using Lagrange multipliers on the constrained optimization problem: "minimize $L = \sum_n e^{-\rho \tilde{f}(x_n)}$ under the constraint $||V_k|| = 1$ wrt V_k and ρ_k". The stationary points of V_k in (B) in the limit of $t \to \infty$ coincide with the limit $\rho \to \infty$ in the dynamics (A) and they are maxima of the margin.*
3. *The unconstrained gradient descent dynamics converges to the same stationary points of the flow of V_k as (A) and (B).*
4. *Weight normalization(40) corresponds to dynamics (B).*
5. *For each layer $\frac{\partial \rho_k^2}{\partial t}$ is the same irrespectively of k.*
6. *In the 1-layer network case $\rho \approx \log t$ asymptotically. For deeper networks, the product of weights at each layer diverges faster than logarithmically, but each individual layer diverges slower than in the 1-layer case.*

In summary, there is an implicit regularization in deep networks trained on exponential-type loss functions, originating in the gradient descent technique used for optimization. The solutions are in fact the same that are obtained by regularized optimization. Convergence to a specific solution instead of another, depends on the trajectory of gradient flow and corresponds to one of multiple minima of the loss (linear networks will have a unique minimum), each one being a margin maximizer. In general each solution will show a different test performance. Characterizing the conditions that lead to the best among the margin maximizers is an open problem.

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 in the weights W_k that are typically more degenerate (for the exponential loss) than other local critical points.

Gradient descent yields generalization by the normalized network in terms of the exponential loss (but not in terms of classification), despite overparametrization and even in the absence of explicit norm control or regularization, because in the case of exponential-type losses, the directions of the weights converges to one of several stable minima for finite times (and to a minimum norm solution for time going to infinity). This basic complexity control mechanism – regularization – however, does not fully explain the behavior of overparametrized deep networks, which fit the training data and perform well on out-of-sample points. Remember that the classical analysis of Empirical Risk Minimization (ERM) algorithms studies their asymptotic behavior for the number of data n going to infinity. In this limiting regime, $n > D$ where D is the fixed number of weights; consistency (informally the expected error of the empirical minimizer converges to the best in the class) and generalization (the empirical error of the minimizer converges to the expected error of the minimizer) are equivalent. The capacity control described in this note implies that there is asymptotic generalization and consistency in deep networks but, in addition, for certain regimes with $n < D$ there can be good expected error in the absence of generalization, in analogy with regression cases of some kernel methods (35, 43–45) (see also (36)). This suggests that under certain conditions, the pseudoinverse may perform well in terms of expected error while the generalization gap (difference between expected and empirical loss) is large. Our analysis of the dynamics of deep networks, once adapted to the square loss, suggests that under gradient descent, the weights W_k of each layer should converge to minimum norm minimizers, in analogy with the linear case, because of the iterative regularization properties(37) of gradient descent.

Of course many other problems also 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 analysis for optimization under the exponential loss is missing a classification of local minima and their dependence on overparametrization. A full theory would also require an analysis of the trade-off 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. Soudry D, Hoffer E, Srebro N (2017) The Implicit Bias of Gradient Descent on Separable Data. *ArXiv e-prints*.
2. Lyu K, Li J (2019) Gradient descent maximizes the margin of homogeneous neural networks. *CoRR* abs/1906.05890.
3. 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.
4. Banburski A, et al. (2019) Theory of deep learning III: Dynamics and generalization in deep networks. *CBMM Memo No. 090*.
5. Niyogi P, Girosi F (1996) On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation* 8:819–842.
6. Poggio T, Smale S (2003) The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society (AMS)* 50(5):537–544.
7. Liang T, Poggio T, Rakhlin A, Stokes J (2017) Fisher-rao metric, geometry, and complexity of neural networks. *CoRR* abs/1711.01530.
8. 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*.
9. 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.
10. Poggio T, Anselmi F, Rosasco L (2015) I-theory on depth vs width: hierarchical function composition. *CBMM memo 041*.
11. 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*.
12. 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*.
13. Donoho DL (2000) High-dimensional data analysis: The curses and blessings of dimensionality in *AMS CONFERENCE ON MATH CHALLENGES OF THE 21ST CENTURY*.
14. Mhaskar H (1993) Approximation properties of a multilayered feedforward artificial neural network. *Advances in Computational Mathematics* pp. 61–80.
15. 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.
16. Chui C, Li X, Mhaskar H (1994) Neural networks for localized approximation. *Mathematics of Computation* 63(208):607–623.
17. 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.
18. Pinkus A (1999) Approximation theory of the mlp model in neural networks. *Acta Numerica* 8:143–195.
19. Poggio T, Smale S (2003) The mathematics of learning: Dealing with data. *Notices of the American Mathematical Society (AMS)* 50(5):537–544.
20. 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.
21. Livni R, Shalev-Shwartz S, Shamir O (2013) A provably efficient algorithm for training deep networks. *CoRR* abs/1304.7045.
22. 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*.
23. Anselmi F, et al. (2015) Unsupervised learning of invariant representations. *Theoretical Computer Science*.
24. Poggio T, Rosaco L, Shashua A, Cohen N, Anselmi F (2015) Notes on hierarchical splines, dclns and i-theory. *CBMM memo 037*.
25. 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*.
26. Telgarsky M (2015) Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101v2 [cs.LG] 29 Sep 2015*.
27. Safran I, Shamir O (2016) Depth separation in relu networks for approximating smooth nonlinear functions. *arXiv:1610.09887v1*.
28. 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.
29. Daubechies I, DeVore R, Foucart S, Hanin B, Petrova G (2019) Nonlinear approximation and (deep) relu networks. *arXiv e-prints* p. arXiv:1905.02199.
30. 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.
31. Allen-Zhu Z, Li Y, Liang Y (2018) Learning and generalization in overparameterized neural networks, going beyond two layers. *CoRR* abs/1811.04918.
32. 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.
33. Wei C, Lee JD, Liu Q, Ma T (2018) On the margin theory of feedforward neural networks. *CoRR* abs/1810.05369.
34. Belkin M, Ma S, Mandal S (2018) To understand deep learning we need to understand kernel learning. *ArXiv e-prints*.
35. Liang T, Rakhlin A (2018) Just Interpolate: Kernel "Ridgeless" Regression Can Generalize. *arXiv e-prints* p. arXiv:1808.00387.
36. Mei S, Montanari A (2019) The generalization error of random features regression: Precise asymptotics and double descent curve. *arXiv e-prints* p. arXiv:1908.05355.
37. Rosasco L, Villa S (2015) Learning with incremental iterative regularization in *Advances in Neural Information Processing Systems*. pp. 1630–1638.
38. 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.
39. Du SS, Hu W, Lee JD (2018) Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced in *Advances in Neural Information Processing Systems 31*, eds. Bengio S, et al. (Curran Associates, Inc.), pp. 384–395.
40. Salimans T, Kingm DP (2016) Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in Neural Information Processing Systems*.
41. Bousquet O, Elisseeff A (2001) Stability and generalization. *Journal Machine Learning Research*.
42. Ferreira PJSG (1996) The existence and uniqueness of the minimum norm solution to certain linear and nonlinear problems. *Signal Processing* 55:137–139.
43. Liang T, Rakhlin A, Zhai X (2019) On the Risk of Minimum-Norm Interpolants and Restricted Lower Isometry of Kernels. *arXiv e-prints* p. arXiv:1908.10292.
44. Rakhlin A, Zhai X (2018) Consistency of Interpolation with Laplace Kernels is a High-Dimensional Phenomenon. *arXiv e-prints* p. arXiv:1812.11167.
45. Belkin M, Hsu D, Xu J (2019) Two models of double descent for weak features. *CoRR* abs/1903.07571.