Theoretical issues in deep networks

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

[a]Center for Brains, Minds, and Machines, Massachusetts Institute of Technology, Cambridge, MA 02139

Edited by David L. Donoho, Stanford University, Stanford, CA, and approved May 1, 2020 (received for review June 3, 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, and good out-of-sample performance, despite overparameterization and the absence of explicit regularization. We review our recent results toward this goal. In approximation theory both shallow and deep networks are known to approximate any continuous functions at an exponential cost. However, we proved that for certain types of compositional functions, deep networks of the convolutional type (even without weight sharing) can avoid the curse of dimensionality. In characterizing minimization of the empirical exponential loss we consider the gradient flow of the weight directions rather than the weights themselves, since the relevant function underlying classification corresponds to normalized networks. The dynamics of normalized weights turn out to be equivalent to those of the constrained problem of minimizing the loss subject to a unit norm constraint. In particular, the dynamics of typical gradient descent have the same critical points as the constrained problem. Thus there is implicit regularization in training deep networks under exponential-type loss functions during gradient flow. As a consequence, the critical points correspond to minimum norm infima of the loss. This result is especially relevant because it has been recently shown that, for overparameterized models, selection of a minimum norm solution optimizes cross-validation leave-one-out stability and thereby the expected error. Thus our results imply that gradient descent in deep networks minimize the expected error.

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 refs. 1 and 2). They include issues such as the approximation power of deep networks, the dynamics of the empirical risk minimization by gradient flow, and the generalization properties of gradient descent techniques—Why does the expected error not suffer, despite the absence of explicit regularization, when the networks are overparameterized? In this paper we review briefly our work on approximation and describe recent results in characterizing complexity control in training deep networks. The first part, about approximation power of deep versus shallow architectures, is a brief summary of our main results described in published papers (3) and references there. The second part describes a recent characterization (details in ref. 4 with a short announcement in ref. 5) of the optimization process used to train deep networks and in particular of the properties of the dynamical system corresponding to the gradient flow, which is the continuous equivalent of the gradient descent algorithms in common use (we study gradient flow and leave to others the extension to gradient descent). One of these results, about the equivalence of constrained and unconstrained optimization, implies a classical form of complexity control by gradient descent with respect to the exponential loss.

2. Approximation

We start with the first set of questions, summarizing results in refs. 3 and 6–9. 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 a certain type of compositional functions, that is, functions of functions. An especially interesting subset of compositional functions is the ones that can be written as hierarchically local compositional functions where all of 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 as shown in Fig. 1—although, importantly, weight sharing is not necessary.

Implications of results likely to be relevant in practice are 1) 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, 2) the problems for which certain deep networks are guaranteed to avoid the curse of dimensionality (see for a nice review ref. 10) correspond to input–output mappings that are compositional with local constituent functions, and 3) 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 1980s 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 refs. 11–15). By now, many publications (for instance refs. 16–18) characterize approximation properties of generic multilayer networks. Much interest has focused on separation results, that is, on the fact that specific functions cannot be represented efficiently by shallow networks (19) (see also refs. 14, 20, and 21). An interesting review of approximation of univariate functions by deep rectified linear unit (ReLU) networks has recently appeared (22). Unlike all these papers, we focus on the approximation by deep networks of certain important classes of multivariate functions which avoid the curse of dimensionality. We think that our results are especially interesting because they represent a potential explanation for one of the greatest puzzles that has emerged from the field

This paper results from the Arthur M. Sackler Colloquium of the National Academy of Sciences, "The Science of Deep Learning," held March 13–14, 2019, at the National Academy of Sciences in Washington, DC. NAS colloquia began in 1991 and have been published in PNAS since 1995. From February 2001 through May 2019 colloquia were supported by a generous gift from The Dame Jillian and Dr. Arthur M. Sackler Foundation for the Arts, Sciences, & Humanities, in memory of Dame Sackler's husband, Arthur M. Sackler. The complete program and video recordings of most presentations are available on the NAS website at http://www.nasonline.org/science-of-deep-learning.

Author contributions: 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 competing interest.

Published under the PNAS license.

This article is a PNAS Direct Submission.

Data deposition: All data and code associated with this paper are accessible publicly at GitHub, https://github.com/liaoq/pnas2019.

[1]To whom correspondence may be addressed. Email: tp@ai.mit.edu.

First published June 9, 2020.

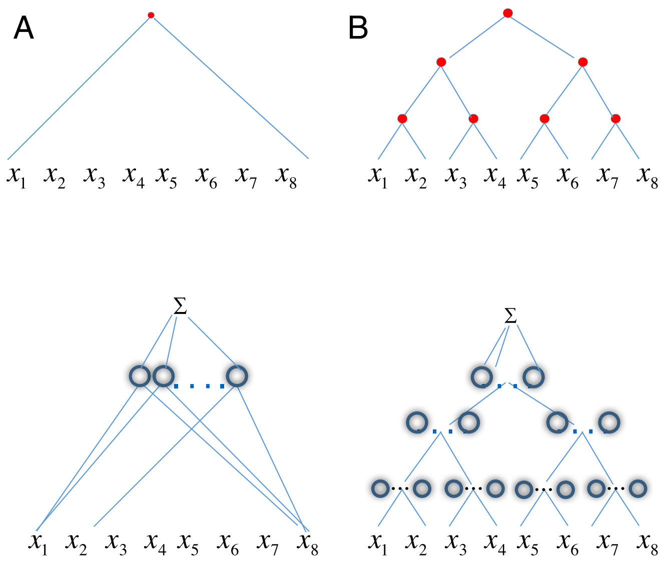

Fig. 1. *Top* graphs are associated to functions. Each *Bottom* diagram (*Insets*) depicts the ideal network approximating the function above. (*Inset A*) A shallow universal network in 8 variables and N units approximates a generic function of 8 variables $f(x_1, \ldots, x_8)$. (*Inset B*) A hierarchical network at the bottom in $n = 8$ variables, which approximates well functions of the form $f(x_1, \ldots, x_8) = h_3(h_{21}(h_{11}(x_1, x_2), h_{12}(x_3, x_4)), h_{22}(h_{13}(x_5, x_6), h_{14}(x_7, x_8)))$ 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 ReLU units. Similar to the shallow network, a hierarchical network is universal; that is, it can approximate any continuous function; the text discusses how it can approximate a subclass of compositional functions exponentially better than a shallow network. Redrawn from ref. 23.

of deep learning, that is, the unreasonable effectiveness of convolutional deep networks in a number of sensory problems.

B. Shallow and Deep Networks. The general paradigm is as follows. We are interested in determining how complex a network, denoted as a function $f(x)$, ought to be to theoretically guarantee approximation of an unknown target function g up to a given accuracy $\epsilon > 0$. In particular, this section characterizes conditions under which deep networks are "better" than shallow networks in approximating functions. Both types of networks use the same small set of operations—dot products, linear combinations, a fixed nonlinear function of one variable, possibly convolution, and pooling. Each node in the networks corresponds to a node in the graph of the function to be approximated, as shown in Fig. 1.

We define a deep network with K layers with the usual coordinate-wise scalar activation functions $\sigma(z): \quad \mathbf{R} \to \mathbf{R}$ as the set of functions $f(W; x) = \sigma(W^K \sigma(W^{K-1} \cdots \sigma(W^1 x)))$, where the input is $x \in \mathbf{R}^d$, and 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 $k = 1, \ldots, K$. There are no bias terms: The bias is instantiated in the input layer by one of the input dimensions being a constant. A shallow network is a deep network with $K = 1$.

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

Note that in our main example (Fig. 1) of a network corresponding to a function with a binary tree graph, the resulting architecture is an idealized version of deep convolutional neural networks described in the literature. In particular, it has only one output at the top unlike most of the deep architectures with many channels and many top-level outputs. Correspondingly, each node computes a single value instead of multiple channels, using the combination of several units. However, our results hold also for these more complex networks (see corollary 4 in ref. 21).

The sequence of results is as follows:

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

The two main theorems (3) are one about shallow networks (originally due to ref. 24) and the second one about deep networks with smooth activation function (7–9). They can be condensed in the following informal statement, where complexity is measured in terms of number N of units:

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

The exponential dependence of the number of parameters required for an accuracy $\mathcal{O}(\epsilon)$ on the dimension n is known as the curse of dimensionality. The precise statement of the result in ref. 3 shows that the curse of dimensionality can be reduced by smoothness of the target functions. Of course, the binary tree case of the theorem is the simplest case: The extension to more complex trees is obvious.

The result of *Theorem 1* can be extended to nonsmooth ReLU for the case of deep networks in a number of ways, one of which, suggested by Mhaskar (25), is as follows. The first step is to recall theorem 3.2 in ref. 25 that applies to smooth activation functions such as the "square" of the ReLU, that is, the function x_+^2 defined as $= x^2$ if $x \geq 0$ and otherwise $= 0$. Loosely speaking, theorem 3.2 implies that for such activation functions deep networks with a sufficient number of neurons and layers can approximate continuous functions as well as free knots splines. The second step is to use theorem 1 in ref. 26 on approximating $(x_+)^2$ with ReLU networks.

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

Consider the space of polynomials, which is of course a smaller space than spaces of Sobolev functions, but which can approximate arbitrarily well continuous functions. Let us call P_k^n the linear space of polynomials of degree at most k in n variables and T_k^n the subset of the space P_k^n which consists of compositional polynomials with a binary tree graph and constituent polynomial functions of degree k (in two variables). Then the following theorems hold (3):

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

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

Theorems 2 and *3* make clear that a hierarchical compositional polynomial corresponding to a binary graph is a very sparse polynomial—among the polynomials of the same degree in the same number of variables. It is possible to show (3) that this intuition extends from polynomials to Sobolev functions. Of course the target function graph in Fig. 1, *Top Right* is just one of the directed acyclic graphs (DAGs) (23) that can describe different forms of compositionality. Prior knowledge of the DAG underlying a learning problem can be exploited to a great advantage by an appropriate deep-learning architecture.

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 as 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 ref. 27 illuminate the apparent absence of "overfitting" 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 deep neural networks (DNNs) under exponential-type losses (Fig. 2). We first outline related work. We then describe our main results, summarized in *Theorem 4* (4, 28).

A. Related Work. A number of papers have studied gradient descent for deep networks (29–31). Close to the approach of refs. 4 and 28 reviewed here is ref. 32. 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. Two very recent papers (33, 34) develop an elegant margin maximization-based approach which leads to some of the same results of this section (and others). Our goal here is to review where the complexity control is hiding in the training of deep networks under exponential-type loss functions and how it leads to the selection by gradient descent of minimum norm solutions.

B. The Dynamics of Optimization. We begin by some notation and basic properties that we need in this optimization section.

We consider, for simplicity, the case in which f takes scalar values, implying that the last layer matrix W^K has size $1\times h_{K-1}$, where h_k denotes the size of layer k. The weights of hidden layer k have size $h_k\times h_{k-1}$. In the case of binary classification which we consider here the labels are $y\in\{-1,1\}$. The activation function in this section is the ReLU activation.

For ReLU activations the following important positive one-homogeneity property holds: $\forall z,\forall a\geq 0,\sigma(\alpha z)=\alpha\sigma(z)$. Thus, $\sigma(z)=\frac{\partial\sigma(z)}{\partial z}z$. A consequence of one-homogeneity is a structural lemma (lemma 2.1 of ref. 36, closely related to Euler's theorem for homogeneous functions) $\sum_{i,j}W_k^{i,j}\left(\frac{\partial f(W;x)}{\partial W_k^{i,j}}\right)=f(W;x)$, 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 $f(W;x)=\prod_{k=1}^K\rho_k f(V_1,\ldots,V_K;x)$, where $W_k=\rho_k V_k$ with the matrix norm $\|V_k\|_p=1$. Note that $\frac{\partial f(W;x)}{\partial\rho_k}=\frac{\rho}{\rho_k}f(V;x)$ and that the definitions of ρ_k and V_k all depend on the choice of the norm used in normalization.

We will assume that for some $t>T_0$ the gradient flow converges to an f that separates the data; that is, $f(x_n)y_n>0$

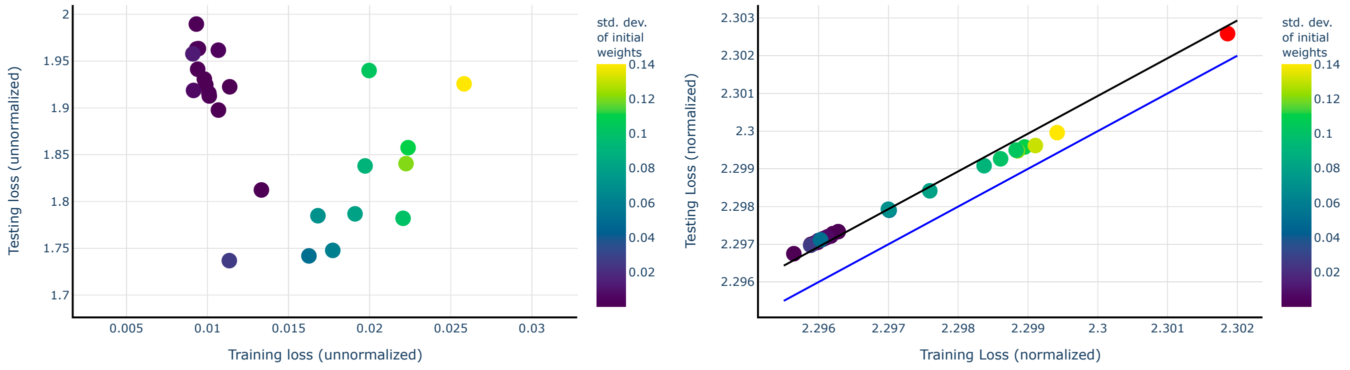

Fig. 2. Empirical evidence of a small generalization gap by normalized networks with respect to the cross-entropy loss. *Left* graph shows testing vs. training cross-entropy loss for networks each trained on the same datasets (CIFAR-10) (46) but with different initializations, yielding zero classification error on the training set but different testing errors. *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. Note that all points have zero classification error at training. The red circle on the top right refers to a network trained on the same CIFAR-10 dataset 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 three-layer convolutional networks. *Left* graph can be considered as a visualization of generalization bounds when the Rademacher complexity is not controlled. *Right* graph is a visualization of the same relation for normalized networks that is $L(\rho f(V;x))\leq\hat{L}(\rho f(V;x))+c_1\rho\mathbb{R}_N(\tilde{\mathbb{F}})+c_2\sqrt{\ln(\frac{1}{\delta})/2N}$ 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. Note that the exponential loss is not a better proxy for the classification loss for large ρ. Empirically for these datasets, the test exponential loss with $\rho=1$ provides a ranking that approximately agrees with the test classification ranking. From ref. 35.

$\forall n = 1, \dots, N$. The assumption of separability, which is very strong in the case of linear networks, is much weaker in the case of overparameterized deep networks that typically can "fit the training set"; that is, they can separate the training data after a relatively small number of gradient descent iterations.

We conjecture that the hypothesis of smooth activations we use here is a technicality due to the necessary conditions for existence and uniqueness of solutions to ordinary differential equations (ODEs). Generalizing to differential inclusions and nonsmooth dynamical systems should allow for these conditions to be satisfied in the Filippov sense (37). The Clarke subdifferential is supposed to deal appropriately with functions such as ReLU (34).

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 $L = \frac{1}{N}\sum_n e^{-y_n f(x_n)}$ by computing

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

on a given dataset $\{x_n, y_n\} \quad \forall n = 1, \dots, N$ with y binary. Since the goal is binary classification, we are interested in $f(V;x)$ (remember $\operatorname{sign} f(V;x) = \operatorname{sign} f(W;x)$). We want to study the dynamics of $f(V;x)$ implied by Eq. **1**. With this goal we study three related versions of this problem:

1) Minimization of $L = \frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)}$ under the constraint $\|V_k\| = 1$ with respect to (wrt) V_k for fixed ρ;
2) Minimization of $L = \frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)}$ under the constraint $\|V_k\| = 1$ wrt V_k and ρ;
3) Minimization of $L = \frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)} = \frac{1}{N}\sum_n e^{-y_n f(W;x_n)}$ wrt V_k, ρ, which is equivalent to typical training, Eq. **1**.

B.1. Constrained minimization. Constrained optimization of the exponential loss, using Lagrange multipliers, minimizes $L = \frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)}$ under the constraint $\|V_k\| = 1$ which leads us to minimize

$$\mathcal{L} = \frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)} + \sum_k \lambda_k \|V_k\|^2 \quad [2]$$

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 (4).

B.2. Fixed ρ: minima. Gradient descent on $\mathcal{L}$ for fixed ρ wrt V_k yields then the dynamical system

$$\dot{V}_k = \rho \frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)} y_n \left(\frac{\partial f(V;x_n)}{\partial V_k} - V_k f(V;x_n)\right) \quad [3]$$

because $\lambda_k = \frac{1}{2}\rho\frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)} y_n f(V;x_n)$, since $V_k^T \dot{V}_k = 0$, which in turn follows from $\|V_k\|^2 = 1$.

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

$$\sum_n e^{-\rho y_n f(V;x_n)} y_n \frac{\partial f(V;x_n)}{\partial V_k} = \sum_n e^{-\rho y_n f(V;x_n)} V_k y_n f(V;x_n). \quad [4]$$

Of course the infimum of the exponential loss L is zero only 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 the critical point of the gradient with a positive semidefinite Hessian. In general it is not unique; it is unique in the case of one-hidden-layer linear networks.

B.3. $\rho \to \infty$ has the same stationary points as the full dynamical system. Consider the limit of $\rho \to \infty$ in Eq. **4**. The asymptotic stationary points of the flow of V_k then satisfy

$$\sum_n e^{-\rho y_n f(V;x_n)} y_n \left(\frac{\partial f(V;x_n)}{\partial V_k} - V_k f(V;x_n)\right) = 0 \quad [5]$$

also in the limit $\lim_{\rho\to\infty}$, that is, for any large ρ. So the stationary V_k points for any large $\rho = R$ satisfy Eq. **5** with $\rho = R$.

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

$$\dot{\rho_k} = \frac{\rho}{\rho_k}\frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)} y_n f(V;x_n) \quad [6]$$

$$\dot{V}_k = \rho\frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)} y_n \left(\frac{\partial f(V;x_n)}{\partial V_k} - V_k f(V;x_n)\right). \quad [7]$$

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 $\rho \propto \log t$) (4). Thus $\rho(t)$ is a monotonically increasing function from t_0 to $t = \infty$. Furthermore, ρ_k grows at a rate which is independent of the layer k (4). In fact, Eq. **1** 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 $\dot{\|W_k\|^2} = 2\sum_{n=1}^{N} y_n f(W;x_n) e^{-y_n f(W;x_n)}$, which shows that the rate of growth of $\|W_k\|^2$ is independent of k. This observation (38) is summarized as follows:

Lemma 1. *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,* $\frac{\partial\rho_k^2}{\partial t} = 2\frac{1}{N}\sum_{n=1}^{N}\rho y_n f(V;x_n) e^{-\rho y_n f(V;x_n)}$.

Because $\rho(t)$ grows monotonically in t, there exists T such that $\rho(T) = R$. At time T then, the condition for a stationary point of V_k in Eq. **7** coincides with Eq. **5**. This leads to the following:

Lemma 2. *The full dynamical system* **7** *in the limit of* $t \to \infty$ *converges to the same limit to which the dynamical system* **3** *converges for* $\rho \to \infty$, *in the sense of having the same sets of stationary points.*

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. We start from Eq. **5**. Without loss of generality let us assume that the training data $f(V;x_n)$ are ranked at $t = T_0$ according to increasing normalized margin; that is, $f(V;x_1) \le f(V;x_2) \le \dots \le f(V;x_N)$. Let us call $B_k(V;x_n) = y_n(\frac{\partial f(V;x_n)}{\partial V_k} - V_k f(V;x_n))$. Then the equilibrium condition of Eq. **5** becomes

$$\sum_n^N e^{-\rho(T_0) y_n f(V;x_n)} B_k(V;x_n) = e^{-R y_1 f(V;x_1)}(B_1 + B_2 e^{-R\Delta_2} + \dots + B_N e^{-R\Delta_N}) = 0, \quad [8]$$

where $\Delta_1 = y_1 f(V;x_1) - y_1 f(V;x_1) = 0$, $\Delta_2 = y_2 f(V;x_n) - y_1 f(V;x_1) \ge 0$, $\Delta_3 = y_3 f(V;x_n) - y_1 f(V;x_1) \ge 0$, and all $\Delta_n \ge 0$ increase with n. Eq. **8** can be rewritten then as

$$e^{-\rho(T_0) y_1 f(V;x_1)}(\alpha_1 B_1 + \alpha_2 B_2 + \dots + \alpha_N B_N) = 0, \quad [9]$$

where the α_n are all positive ($\alpha_1 = 1$) and decreasing with n. The left-hand side of the stationary point equation has the

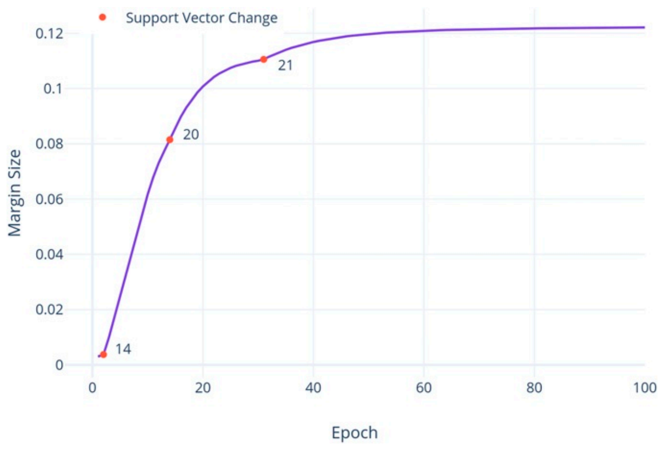

Fig. 3. Monotonic increase of the margin: the growth of the margin $\min_n y_n f(V; x_n)$ in the binary case. The network converges to 100% training accuracy in epoch 3 and changes the support vectors three times (red circles—the numbers above them correspond to the index of the datapoint that becomes the support vector) before stabilizing. As predicted by theory, the margin is nondecreasing. As the support vectors change, note that the rate of margin growth accelerates.

form $\epsilon(B_1 + \epsilon' B) = 0$, with $B = \sum_{n=2}^{N} \alpha_n B_n$, with both ϵ and ϵ' going to zero for $T_0 \to \infty$. Thus $\lim_{T_0 \to \infty} \epsilon(B_1 + \epsilon' B) = \lim_{T_0 \to \infty} \epsilon B_1 = 0$ can be satisfied for sufficiently large T_0 by $B_1 = 0$, that is, by $\frac{\partial f(V; x_1)}{\partial V_k} - V_k f(V; x_1) = 0$. This is the condition in which the stationary point corresponding to x_1 provides the maximum, or more precisely, supremum of the margin. Before that limit is reached, the solution V_k changes with increasing $\rho(t)$. Thus the asymptotic stationary points coincide with maximum margin. The following lemma (4) shows that the margin is increasing with t, after separability is reached and after a single support vector dominates (Fig. 3):

Lemma 3. *There exists a $\rho(T_*)$ such that for $\rho > \rho(T_*)$ the sum $\sum_{n=1}^{N} e^{-\rho y_n f(V; x_n)} \propto e^{-\rho y_1 f(V; x_1)}$. For $\rho > \rho(T_*)$ then the margin increases $y_1 \frac{\partial f(V; x_1)}{\partial t} \geq 0$ (even if ρ is kept fixed).*

Proof: $y_1 f(V; x_1)$ increases monotonically or is constant because

$$y_* \frac{\partial f(V; x_1)}{\partial t} = \sum_k \left(\frac{\partial y_1 f(V; x_1)}{\partial V_k} \right)^T \dot{V}_k$$
$$= \sum_k \frac{\rho}{\rho_k^2} e^{-\rho y_1 f(V; x_1)} \left(\left\| \frac{\partial f(V; x_1)}{\partial V_k} \right\|_F^2 - f(V; x_1)^2 \right). \quad [10]$$

Eq. **10** implies $\frac{\partial y_1 f(V, x_1)}{\partial t} \geq 0$ because $\|f(V; x_1)\|_F = \| V_k^T \frac{\partial f(V; x_1)}{\partial V_k} \|_F \leq \| \frac{\partial f(V; x_1)}{\partial V_k} \|_F$, since the Frobenius norm is submultiplicative and V_k has unit norm.

We finally note that the maximum margin solution in terms of $f(V; x)$ 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 (4):

Lemma 4. *The maximum margin problem*

$$\max_{W_K, \dots, W_1} \min_n y_n f(W; x_n), \quad \textit{subj. to} \quad \| W_k \| = 1, \quad \forall k \quad [11]$$

is equivalent to

$$\min_{W_k} \frac{1}{2} \| W_k \|^2, \ \textit{subj. to} \ y_n f(W; x_n) \geq 1, \quad \forall k, \ n = 1, \dots, N. \quad [12]$$

B.5. Unconstrained gradient descent for deep networks: implicit norm control. Empirically it appears that gradient descent (GD) and stochastic gradient descent (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, irrespective 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 reparameterization $W_k^{i,j} = \rho_k V_k^{i,j}$ 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 $\frac{\partial f(W; x_n)}{\partial W_k} = \frac{\rho}{\rho_k} \frac{\partial f(V; x_n)}{\partial V_k}$.

The unconstrained gradient descent dynamic system used in training deep networks for the exponential loss is given in Eq. **1**; that is,

$$\dot{W}_k = -\frac{\partial L}{\partial W_k} = \frac{1}{N} \sum_{n=1}^{N} y_n \frac{\partial f(W; x_n)}{\partial W_k} e^{-y_n f(W; x_n)}. \quad [13]$$

Following the chain rule for the time derivatives, the dynamics for W_k induce the following dynamics for $\| W_k \| = \rho_k$ and V_k by using relations 1) to 3) above,

$$\dot{\rho}_k = \frac{\partial \| W_k \|}{\partial W_k} \frac{\partial W_k}{\partial t} = V_k^T \dot{W}_k$$
$$\dot{V}_k = \frac{\partial V_k}{\partial W_k} \frac{\partial W_k}{\partial t} = \frac{S_k}{\rho_k} \dot{W}_k, \quad [14]$$

where $S_k = I - V_k V_k^T$. Thus, unconstrained gradient descent coincides with the dynamical system

$$\dot{\rho}_k = \frac{\rho}{\rho_k} \frac{1}{N} \sum_{n=1}^{N} e^{-\rho y_n f(V; x_n)} y_n f(V; x_n)$$
$$\dot{V}_k = \frac{\rho}{\rho_k^2} \frac{1}{N} \sum_{n=1}^{N} e^{-\rho y_n f(V; x_n)} y_n \left(\frac{\partial f(V; x_n)}{\partial V_k} - V_k f(V; x_n) \right)$$

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

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 as follows:

Lemma 5. *Constrained and unconstrained gradient descents have the same set of minima or infima.*

In ref. 4 we show that the dynamics of the constrained gradient descent Eq. **7** are the same as those of the well-known and much used algorithm called "weight normalization" (39).

Actual convergence for the constrained case happens if

$$\frac{\partial f(V;x_*)}{\partial V_k} - V_k f(V;x_*) = 0 \qquad \textbf{[15]}$$

which corresponds to a long but finite time and thus a large but finite ρ and a small but nonzero λ. Eq. **15** is thus the solution that corresponds to a regularization problem with a nonzero but vanishing λ.

Note that Eq. **15** gives immediately the solution in the one-layer linear case of $f(V;x) = V^T x$. The stationary points of the gradient are given by $\sum \alpha_n \rho(t)(x_n - VV^T x_n)$. The Lagrange multiplier case is similar, giving $\sum \alpha_n \rho(t) x_n - \lambda V = 0$, with λ satisfying $\lambda^2 = \frac{1}{N}\sum_n e^{2\rho y_n V^T x_n} \rho^2 y_n x_n$. Thus $\lambda \to 0$ for $\rho \to \infty$. In this case, if $f(x_1) = 1$, the solution is $V = x_1$. In the two-layer linear case, the equations provide a direct relation between V_1 and V_2, which can be extended to the nonlinear, ReLU case.

C. Summary. There are several additional results on the dynamics of ρ and on the different convergence times for linear networks in the case of weight normalization vs. unconstrained gradient descent, weight normalization showing a faster convergence. For these additional results we refer to ref. 4. We describe here a few implications about the loss landscape.

C.1. Landscape and minima. ReLU networks with exponential-type loss functions do not have zeros of the gradient (wrt the unnormalized 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 $\sum_{n=1}^{N} y_n \frac{\partial f(W;x_n)}{\partial W_k^{i,j}} e^{-y_n f(W;x_n)} = 0$. 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 $\frac{\partial f(W;x_n)}{\partial W_k^{i,j}}$. Use of the structural lemma shows that $\frac{\partial f(W;x)}{\partial W_k^{i,j}} = 0, \forall i,j,k$ 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. Note 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, since it has at least some positive eigenvalues (4).

Clearly, it would be interesting to characterize better the degeneracy of the local minima (we conjecture that the loss function of a deep network is a Morse–Bott function). For the goals of this section, however, the fact that they cannot be completely degenerate is sufficient. As shown in ref. 4, under the exponential loss, zero loss (at infinite ρ) corresponds to a degenerate infimum, 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.2. Summary theorem. The following theorem (informal statement) summarizes the main results on minimization of the exponential loss in deep ReLU networks:

Theorem 4. *Assume that separability is reached at time T_0 during gradient descent on the exponential loss; that is, $y_n f(W;x_n) > 0$, $\forall n$. 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 $L = \sum_n e^{-\rho y_n f(V;x_n)}$ under the constraint $\|V_k\| = 1$ wrt V_k." The dynamics converge for any fixed ρ to stationary points of the V_k flow that are minima with a positive semidefinite Hessian.*
2) *Consider the dynamics (B) resulting from using Lagrange multipliers on the constrained optimization problem: "Minimize $L = \frac{1}{N}\sum_n e^{-\rho y_n f(V;x_n)}$ 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 converge to the same stationary points of the flow of V_k as (A) and (B).*
4) *Weight normalization* (39) *corresponds to dynamics (B).*
5) *For each layer $\frac{\partial \rho_k^2}{\partial t}$ is the same irrespective of k.*
6) *In the one-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 one-layer case.*
7) *Gradient flow converges for infinite time to directions given by a set of support vectors with the same value $y_1 f(V;x_1)$ that satisfy the set of K coupled vector equations $(\frac{\partial f(V;x_1)}{\partial V_k} = V_k f(V;x_1))$.*

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 as those that are obtained by regularized optimization. Convergence to a specific solution instead of another of course depends on the trajectory of gradient flow and corresponds to one of multiple infima 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 specific 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 can be represented in this way. Although 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 local hierarchical compositionality is imposed by the wiring of our sensory cortex and, critically, is reflected in language. Thus the specific compositionality of some of the most common visual tasks may simply reflect the way our brain works. From a general point of view, it is clear that there are many different forms of compositionality that translate in different DAGs (3).

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

The gradient flow on the appropriately normalized network corresponding to gradient descent is under complexity control, despite overparameterization and even in the absence of explicit regularization, because in the case of exponential-type losses, the directions of the weights converge to one of several minima for finite times (and to a minimum norm solution for time going to infinity). This implicit complexity control mechanism—implicit regularization—however, does not by itself fully explain the behavior of overparameterized 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. Empirically we know that for a nonasymptotic regime with $N < D$ deep networks can yield good expected error in the absence of generalization. This is similar to the regression case for some linear kernel methods (40–44). This phenomenon 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. We note that is not surprising that complexity control is a prerequisite for good performance even in an overparameterized regime in which the classical analysis via generalization does not apply. The minimum-norm pseudoinverse solution is unique and continuous, satisfying the key conditions—including stability with respect to noise in the data—of a well-posed problem. In fact, a recent paper (45) shows that solutions of overparameterized deep networks that optimize cross-validation leave-one-out stability minimize the expected error. Furthermore they suggest that maximum stability corresponds to minimum norm solutions. Thus the main results of this paper imply that gradient flow in deep networks with an exponential-type loss does (locally) minimize the expected error because it selects minimum norm solutions.

5. Materials and Data Availability

All data and code associated with this paper are accessible publicly at GitHub, https://github.com/liaoq/pnas2019.

ACKNOWLEDGMENTS. We are grateful to Sasha Rakhlin, Nati Srebro, Arturo Deza, and Akshay Rangamani for useful comments and suggestions, particularly about the structural lemma and about separating critical points. Part of the funding for this work is from the Center for Brains, Minds and Machines, funded by National Science Foundation Science and Technology Center Award CCF-1231216, and part is from Center for Brain-Inspired Computing, one of six centers in Joint University Microelectronics Program, a Semiconductor Research Corporation program sponsored by Defense Advanced Research Projects Agency.

1. P. Niyogi, F. Girosi, On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Comput.* **8**, 819–842 (1996).
2. T. Poggio, S. Smale, The mathematics of learning: Dealing with data. *Not. Am. Math. Soc.* **50**, 537–544 (2003).
3. T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, Q. Liao, Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *Int. J. Autom. Comput.* **14**, 503–519 (2017).
4. A. Banburski *et al.*, *Theory of Deep Learning III: Dynamics and Generalization in Deep Networks* (Center for Brains, Minds and Machines [CBMM], Cambridge, MA, 2019).
5. T. Poggio, Q. Liao, A. Banburski, Complexity control by gradient descent in deep networks. *Nat. Commun.* **11**, 1027 (2020).
6. F. Anselmi, L. Rosasco, C. Tan, T. Poggio *Deep Convolutional Networks are Hierarchical Kernel Machines* (Center for Brains, Minds and Machines (CBMM), Cambridge, MA, 2015).
7. T. Poggio, L. Rosasco, A. Shashua, N. Cohen, F. Anselmi, *Notes on Hierarchical Splines, dclns and i-Theory* (Center for Brains, Minds and Machines [CBMM], Cambridge, MA, 2015).
8. T. Poggio, F. Anselmi, L. Rosasco, *I-Theory on Depth vs Width: Hierarchical Function Composition* (Center for Brains, Minds and Machines [CBMM], Cambridge, MA, 2015).
9. H. Mhaskar, Q. Liao, T. Poggio, *Learning Real and Boolean Functions: When is Deep Better than Shallow?* (Center for Brains, Minds and Machines [CBMM] Cambridge, MA, 2016).
10. D. L. Donoho, "High-dimensional data analysis: The curses and blessings of dimensionality" in *AMS Conference on Math Challenges of the 21st Century* (AMS, 2000).
11. H. Mhaskar, Approximation properties of a multilayered feedforward artificial neural network. *Adv. Comput. Math.* **1**, 61–80 (1993).
12. H. N. Mhaskar, "Neural networks for localized approximation of real functions" in *Neural Networks for Processing [1993] III. Proceedings of the 1993 IEEE-SP Workshop* (IEEE, Piscataway, NJ, 1993), pp. 190–196.
13. C. Chui, X. Li, H. Mhaskar, Neural networks for localized approximation. *Math. Comput.* **63**, 607–623 (1994).
14. C. K. Chui, X. Li, H. N. Mhaskar, Limitations of the approximation capabilities of neural networks with one hidden layer. *Adv. Comput. Math.* **5**, 233–243 (1996).
15. A. Pinkus, Approximation theory of the MLP model in neural networks. *Acta Numer.* **8**, 143–195 (1999).
16. G. Montufar, R. Pascanu, K. Cho, Y. Bengio, On the number of linear regions of deep neural networks. *Adv. Neural Inf. Process. Syst.* **27**, 2924–2932 (2014).
17. R. Livni, S. Shalev-Shwartz, O. Shamir, A provably efficient algorithm for training deep networks. arXiv:1304.7045 (26 April 2013).
18. P. Petersen, F. Voigtlaender, Optimal approximation of piecewise smooth functions using deep ReLU neural networks. arXiv:1709.05289 (15 September 2017).
19. M. Telgarsky, Representation benefits of deep feedforward networks. arXiv:1509.08101v2 [cs.LG] (29 September 2015).
20. I. Safran, O. Shamir, Depth separation in ReLU networks for approximating smooth non-linear functions. arXiv:1610.09887v1 (31 October 2016).
21. T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, Q. Liao, *Theory I: Why and When Can Deep - But not Shallow - Networks Avoid the Curse of Dimensionality* (Center for Brains, Minds and Machines [CBMM], Cambridge, MA, 2016).
22. I. Daubechies, R. DeVore, S. Foucart, B. Hanin, G. Petrova, Nonlinear approximation and (deep) ReLU networks. arXiv:1905.02199 (5 May 2019).
23. H. Mhaskar, T. A. Poggio, Deep vs. shallow networks: An approximation theory perspective. arXiv:1608.03287 (10 August 2016).
24. H. N. Mhaskar, Neural networks for optimal approximation of smooth and analytic functions. *Neural Comput.* **8**, 164–177 (1996).
25. H. N. Mhaskar, Approximation properties of a multilayered feedforward artificial neural network. *Adv. Comput. Math.* **1**, 61–80 (1993).
26. D. Yarotsky, Error bounds for approximations with deep ReLU networks. arXiv:1610.01145 (3 October 2016).
27. D. Soudry *et al.*, The implicit bias of gradient descent on separable data[J]. *J. Mach. Learn. Res.* **19**, 2822–2878 (2018).
28. T. Poggio, Q. Liao, A. Banburski, Complexity control by gradient descent in deep networks. *Nat. Commun.* **11**, 1027 (2020).
29. A. Daniely, "SGD learns the conjugate kernel class of the network" in *Advances in Neural Information Processing Systems 30*, I. Guyon *et al.*, Eds. (Curran Associates, Inc., 2017), pp. 2422–2430.
30. Z. Allen-Zhu, Y. Li, Y. Liang, Learning and generalization in overparameterized neural networks, going beyond two layers. arXiv:1811.04918 (12 November 2018).
31. S. Arora, S. S. Du, W. Hu, Z. yuan Li, R. Wang, Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. arXiv:1901.08584 (24 January 2019).
32. C. Wei, J. D. Lee, Q. Liu, T. Ma, On the margin theory of feedforward neural networks. arXiv:1810.05369 (12 October 2018).
33. M. Shpigel Nacson, S. Gunasekar, J. D. Lee, N. Srebro, D. Soudry, Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. arXiv:1905.07325 (17 May 2019).
34. K. Lyu, J. Li, Gradient descent maximizes the margin of homogeneous neural networks. arXiv:1906.05890 (13 June 2019).
35. Q. Liao, B. Miranda, A. Banburski, J. Hidary, T. A. Poggio, A surprising linear relationship predicts test performance in deep networks. arXiv:1807.09659 (25 July 2018).
36. T. Liang, T. Poggio, A. Rakhlin, J. Stokes, Fisher-Rao metric, geometry, and complexity of neural networks. arXiv:1711.01530 (5 November 2017).
37. F. Arscott, A. Filippov, *Differential Equations with Discontinuous Righthand Sides: Control Systems* (Mathematics and Its Applications, Springer, The Netherlands, 1988).
38. S. S. Du, W. Hu, J. D. Lee, "Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced" in *Advances in Neural Information Processing Systems 31*, S. Bengio *et al.*, Eds. (Curran Associates, Inc., 2018), pp. 384–395.
39. T. Salimans, D. P. Kingm, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks" in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, D. D. Lee, U. von Luxburg, Eds. (Curran Associates, Inc., Red Hook, NY, 2016), pp. 901–909.
40. T. Liang, A. Rakhlin, X. Zhai, On the risk of minimum-norm interpolants and restricted lower isometry of kernels. arXiv:1908.10292 (27 August 2019).
41. T. Liang, A. Rakhlin, Just interpolate: Kernel "ridgeless" regression can generalize. arXiv:1808.00387 (1 August 2018).
42. A. Rakhlin, X. Zhai, Consistency of interpolation with Laplace kernels is a high-dimensional phenomenon. arXiv:1812.11167 (28 December 2018).
43. M. Belkin, D. Hsu, J. Xu, Two models of double descent for weak features. arXiv:1903.07571 (18 March 2019).
44. S. Mei, A. Montanari, The generalization error of random features regression: Precise asymptotics and double descent curve. arXiv:1908.05355 (14 August 2019).
45. T. Poggio, *Stable Foundations for Learning* (Center for Brains, Minds and Machines [CBMM], Cambridge, MA, 2020).
46. A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images. https://www.cs.toronto.edu/∼kriz/cifar.html. Accessed 28 May 2020 (2009).