Theory III: Dynamics and Generalization in Deep Networks

Tomaso Poggio\textsuperscript{1}, Qianli Liao\textsuperscript{1}, Brando Miranda\textsuperscript{1}, Andrzej Banburski\textsuperscript{1}, Lorenzo Rosasco\textsuperscript{1}, Bob Liang\textsuperscript{1}, Jack Hidary\textsuperscript{2}

\textsuperscript{1}Center for Brains, Minds, and Machines, MIT
\textsuperscript{2}Alphabet (Google) X

Abstract

We review recent observations on the dynamical systems induced by gradient descent methods used for training deep networks and summarize what is known about the solutions they converge to. Recent results illuminate the puzzle 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. Here we discuss the case of nonlinear multilayer DNNs near zero minima of the empirical loss, under exponential-type losses and square loss, for several variations of the basic gradient descent algorithm, including a new NMGD (norm minimizing gradient descent) version that converges to the minimum norm fixed points of the gradient descent iteration. Our main results are:

- generalization bounds for classification lead to maximize the margin under a unit norm constraint for the product of the Frobenius norms of the weights at different layers;
- gradient descent algorithms on exponential-type loss functions can achieve this goal with appropriate weight normalization;
- existing weight normalization as well as batch normalization techniques can be regarded as approximate implementations of the correct minimization algorithm and this is the fundamental reason for their effectiveness;
- the control of the norm of the weights is related to regularization with vanishing $\lambda(t)$ and to Halpern iterations for minimum norm solutions.

Finally, we show and discuss experimental evidence around the apparent absence of “overfitting”, that is the observation that the expected error does not get worse when increasing the number of parameters. Our explanation focuses on the implicit normalization enforced by algorithms such as batch normalization.

\textsuperscript{1}This replaces previous versions of Theory IIIa and TheoryIIIB.

\textsuperscript{1}This material is based upon work supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF-1231216.
Theory III: Dynamics and Generalization in Deep Networks

Tomaso Poggio, Qianli Liao, Brando Miranda, Andrzej Banburski, Lorenzo Rosasco, Bob Liang, and Jack Hidary

1Center for Brains, Minds and Machines, MIT
1CSAIL, MIT
2Alphabet (Google) X

Abstract

We review recent observations on the dynamical systems induced by gradient descent methods used for training deep networks and summarize what is known about the solutions they converge to. Recent results by [1] illuminate the puzzle 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. Here we discuss the case of nonlinear multilayer DNNs near zero minima of the empirical loss, under exponential-type losses and square loss, for several variations of the basic gradient descent algorithm, including a new NMGD (norm minimizing gradient descent) version that converges to the minimum norm fixed points of the gradient descent iteration. Our main results are:

• generalization bounds for classification lead to maximize the margin under a unit norm constraint for the product of the Frobenius norms of the weights at different layers;
• gradient descent algorithms on exponential-type loss functions can achieve this goal with appropriate weight normalization;
• existing weight normalization as well as batch normalization techniques can be regarded as approximate implementations of the correct minimization algorithm and this is the fundamental reason for their effectiveness;
• the control of the norm of the weights is related to regularization with vanishing $\lambda(t)$ and to Halpern iterations for minimum norm solutions.

Finally, we show and discuss experimental evidence around the apparent absence of “overfitting”, that is the observation that the expected error does not get worse when increasing the number of parameters. Our explanation focuses on the implicit normalization enforced by algorithms such as batch normalization.

---

*This replaces previous versions of Theory IIIa and TheoryIIIb.
†To whom correspondence should be addressed
1 Introduction

In the last few years, deep learning has been tremendously successful in many important applications of machine learning. However, our theoretical understanding of deep learning, and thus the ability of developing principled improvements, has lagged behind. A satisfactory theoretical characterization of deep learning is emerging. It covers the following questions: 1) representation power of deep networks 2) optimization of the empirical risk 3) generalization properties of gradient descent techniques — why the expected error does not suffer, despite the absence of explicit regularization, when the networks are overparametrized? We refer to the latter as the no-overfitting puzzle, around which several recent papers revolve (see among others [2, 3, 4, 5, 6]). This paper addresses the third question. We review properties of the gradient descent solutions for one-layer linear networks [1] and extensions to deep nonlinear networks [7].

2 Deep networks: definitions and properties

Definitions

We define a deep network with $K$ layers with the usual coordinate-wise scalar activation functions $\sigma(z) : \mathbb{R} \rightarrow \mathbb{R}$ as the set of functions $f(W; x) = \sigma(W^K \sigma(W^{K-1} \cdots \sigma(W^1 x)))$, where the input is $x \in \mathbb{R}^d$, the weights are given by the matrices $W^k$, one per layer, with matching dimensions. We 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 $W^K \in \mathbb{R}^{1,Kl}$. 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 paper is the ReLU activation.

Homogeneity

For RELU activations the following positive homogeneity property holds $\sigma(z) = \frac{\partial \sigma(z)}{\partial z} z$. For the network this implies $f(W; x) = \prod_{k=1}^{K} \rho_k \tilde{f}(V_1, \cdots, V_K; x_n)$, where $W_k = \rho_k V_k$ with the Frobenius norm $\|V_k\| = 1$ (for convenience). This implies the following property of RELUs networks wrt their Radamecher complexity:

$$\mathbb{R}_N(F) = \rho_1 \cdots \rho_K \mathbb{R}_N(\tilde{F}). \quad (1)$$

Structural property

The following structural property of the gradient of deep RELU networks is sometime useful (Lemma 2.1 of [8]):

$$\sum_{i,j} \frac{\partial f(x)}{\partial W_{i,j}^{k}} W_{i,j}^{k} = f(x); \quad (2)$$

for $k = 1, \cdots, K$. Equation (2) can be rewritten as a dot product that is $\frac{\partial f(x)}{\partial W_k} W_{i,j}^{k}$ using Einstein summation convention, or in short

$$\left(\frac{\partial f(x)}{\partial W}\right)^T W = f(x) \quad (3)$$
where $W$ is the vectorized representation of the weight matrices $W_k$ for each of the different layers (each matrix is a vector).

Gradient flow and continuous approximation

We will speak of the gradient flow of $L$ (or sometime of the flow of $f$ if the context makes clear that one speaks of the gradient of $L(f)$) referring to

$$\dot{w} = \frac{dw}{dt} = -\gamma(t) \nabla_w (L(f)).$$

where $\gamma(t)$ is the learning rate. In the following we will mix the continuous formulation with the discrete version whenever we feel this is appropriate for the specific statement. We are well aware that the two are not equivalent but we are happy to leave a careful analysis to better mathematicians.

Maximization by exponential

With $\tilde{f}$ being the normalized network (weights at each layer are normalized by the Frobenius norm of the layer matrix) and $\rho$ being the product of the Frobenius norms, the exponential loss $L(f) = \sum_n e^{-y_n f(x_n)} = \sum_n e^{-\rho y_n \tilde{f}(x_n)}$ approximates for “large” $\rho$ a max operation, selecting among all the data points $x_n$ the ones with the smallest margin $\rho \tilde{f}$. Thus minimization of $L(f)$ for large $\rho$ corresponds to margin maximization

$$\arg \min \ L(f) \approx \arg \max_{\tilde{V}_k=1} \min_n y_n \tilde{f}(x_n).$$

A more formal argument could be developed extending theorems of Hastie to the nonlinear case.

3 A semi-rigorous theory of the optimization landscape of Deep Nets: Bezout theorem and Boltzman distribution

In [9, 10] we consider Deep Networks in which each RELU nonlinearity is replaced by a univariate polynomial approximating it. Empirically the network behaves in a quantitatively almost identical way. We then consider such a network in the context of regression under a square loss function. As usual we assume that the network is over-parametrized, that is the number of weights $D$ is larger than the number of data points $N$. The critical points of the gradient of consist of

- global minima corresponding to interpolating networks for which $f(x_i) - y_i = 0$ for $i = 1, \cdots, N$;

- critical points which correspond to saddles and to local minima for which the loss is not zero but $\nabla_w \sum_{i=1}^N V(f(x_i), y_i) = 0$,

In the case of the global, interpolating minimizers, the function $f$ is a polynomial in the $D$ weights (and also a polynomial in the inputs $x$). The degree of each equation is determined by the degree of the univariate polynomial $P$ and by the number of layers $K$. Since the system of polynomial equations, unless the equations are inconsistent, is generically underdetermined – as
many equations as data points in more unknowns – Bezout theorem suggests an infinite number of degenerate global minima, under the form of \( Z \) regions of zero empirical error (the set of all solutions is an algebraic set of dimension at least \( Z = D - N \)); if the underdetermined system is chosen at random the dimension is equal to \( D - N \) with probability one.)

The critical points of the gradient which are not global minimizer are given by the set of equations \( \nabla_w \sum_{i=1}^{N} V(f(x_i), y_i) = 0 \). This is a set of \( D \) polynomial equations in \( D \) unknowns: \( \sum_{i=1}^{N} \nabla_w f(x_i) - y_i \nabla_w f(x_i) = 0 \). In this case, we generically expect a large set of isolated critical points

Thus, we have

**Theorem 1**: There are a large number of global zero-error minimizers which are highly degenerate; the other critical points – saddles and local minima – are generically non-degenerate.

The second part of our argument (in [10]) is that SGD concentrates on degenerate minima. The argument is based on the similarity between a Langevin equation (GDL) and SGD and on the fact that the Boltzman distribution is formally the asymptotic “solution” of the stochastic differential Langevin equation and also of SGDL, defined as SGD with added white noise (see for instance [11]). The Boltzman distribution is

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

where \( Z \) is a normalization constant, \( L(f) \) is the loss and \( T \) reflects the noise power. The equation implies that SGDL prefers degenerate minima relative to non-degenerate ones of the same depth. In addition, among two minimum basins of equal depth, the one with a larger volume is much more likely in high dimensions as shown by the simulations in [10]. Taken together, these two facts suggest that SGD selects degenerate minimizers corresponding to larger isotropic flat regions of the loss. Suppose the landscape of the empirical minima is well-behaved in the sense that deeper minima have broader basin of attraction. Then SDGL shows concentration – because of the high dimensionality – of its asymptotic distribution Equation 6.

Together [9] and [10] suggest the following

**Theorem 2** (informal): SGD selects with high probability the global minimizer of the empirical loss which are degenerate.

### 4 Preliminaries on Generalization

Generalization bounds for regression suggest that bounding the complexity of the minimizer provides a bound on generalization. Ideally, the optimization algorithm should select the smallest complexity minimizers among the solutions – that is the minimizer with minimum norm. An approach to achieve this goal is to add a vanishing regularization term to the loss function (the parameter goes to zero with iterations) which under certain conditions provides convergence to the minimum norm minimizer, independently of initial conditions. This approach goes back to
Halpern fixed point theorem [12], but we will discuss how the same conclusion follows from other techniques such as Lagrange multipliers, normalization and margin maximization theorems [13].

Known margin bounds for classification suggest a similar approach: maximization of the normalized network (the weights at each layer are normalized by the Frobenius norm of the weight matrix of the layer) over the support vectors (the data with smallest margin $y_n f(x_n)$, assuming $y_n f(x_n) > 0, \forall n$).

In the case of nonlinear deep networks, the critical points include saddles and perhaps local minima but again they correspond to global minima of the original loss function that are in general degenerate [9]. A similar approach to the the linear case leads to minimum norm solutions, independently of initial conditions.

4.1 Regression: (local) minimum norm empirical minimizers

We recall that generalization bounds [14] apply to $\forall f \in F$ with probability at least $(1 - \delta)$ and have the typical form

$$|L(f) - \hat{L}(f)| \leq c_1 \mathbb{R}_N(F) + c_2 \sqrt{\frac{\ln\left(\frac{1}{\delta}\right)}{2N}}$$  \hspace{1cm} (7)

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

The bound together with the property Equation 1 implies that among the minimizers with zero square loss, the optimization algorithm should select the minimum norm solution. In any case, the algorithm should control the norm. Standard GD or SGD algorithms do not provide an explicit control of the norm. Empirically it seems that initialization with small weights helps as in the linear case (see Figures). We propose a slight modification of the standard gradient descent algorithms to provide a norm-minimizing GD update – NMGD in short – with the form

$$w_{n+1} - w_n = -(1 - \lambda_n)\gamma_n \nabla_w L(f) - \lambda_n w_n$$ \hspace{1cm} (8)

where $\gamma_n$ is the learning rate and $\lambda_n = \frac{1}{n}$ (this is one of several choices) is the vanishing regularization-like Halpern (see Appendix 9) term.

4.2 Classification: maximizing the margin of the normalized minimizer

A typical margin bound for classification is

$$|L_{\text{binary}}(f) - L_{\text{surr}}(f)| \leq b_1 \frac{\mathbb{R}_N(F)}{\eta} + b_2 \sqrt{\frac{\ln\left(\frac{1}{\delta}\right)}{2N}}$$ \hspace{1cm} (9)

where $\eta$ is the margin, $L_{\text{binary}}(f)$ is the expected classification error, $L_{\text{surr}}(f)$ is the empirical loss of a surrogate loss such as the logistic or the exponential. For a point $x$, the margin is
\( \eta \sim y \rho \tilde{f}(x) \). Since \( \mathbb{R}_N(F) \sim \rho \), the margin bound is optimized by effectively maximizing \( \tilde{f} \) on the “support vectors” — that is the \( x_i, y_i \) s.t. \( \arg \min_n y_n \tilde{f}(x_n) \).

Notice that maximizing the margin subject to unit norm constraint is equivalent to minimize the norm of \( f \) subject to a constraint on the margin. A regularized loss is a related way to optimize, with an appropriate regularization parameter. For this reason we will refer sometime to the solutions in all these cases as minimum norm. This view treats interpolation (in the regression case) and classification (in the margin case) in a unified way.

5 Gradient descent with norm constraint

In this section we focus on the classification case with an exponential loss function. The generalization bounds in the previous section lead directly to consider the following optimization problem:

\[
\arg \max \prod_k ||V_k||=1 \min_n \rho \tilde{f}(x_n). \tag{10}
\]

In words find the network weights that maximizes the margin subject to a norm constraint. The latter ensures a bounded Rademacher complexity and together they minimize the term \( \mathbb{R}_N(F) \eta \).

An approach towards this goal is to minimize the exponential loss function \( L(f(w)) = \sum_{n=1}^{N} e^{\rho f(W;x_n) y_n} = \sum_{n=1}^{N} e^{\rho \tilde{f}(W;x_n) y_n} \), subject to \( ||V_k||^2 = \sum_{i,j} (V_k)_{i,j}^2 = 1, \forall k \). As we discuss later and we also examine in the Appendices, there are several ways to implement the minimization in the tangent space of \( ||V||^2 = 1 \). One of the techniques is roughly equivalent to weight normalization (WN) \cite{15} used since a few years for training deep networks (see Appendix 19.1). Batch normalization is also closely related as we will show later. In the following we describe the Lagrange multiplier method because of its direct implementation of the requirements implied by the generalization bounds.

5.1 Lagrange multiplier method

We define the loss

\[
L = \sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n) y_n} + \sum_{k=0}^{K} \lambda_k ||V_k||^2
\tag{11}
\]

where the Lagrange multipliers \( \lambda_k \) are chosen to satisfy \( ||V_k|| = 1 \) at convergence or when the algorithm is stopped (the constraint can also be enforced at each iteration, see later).

We perform gradient descent on \( L \) with respect to \( \rho, V_k \). We obtain for \( k = 1, \cdots, K \)

\[
\rho(t+1) - \rho(t) = \sum_n \rho(t) e^{-\rho(t) \tilde{f}(x_n) y_n} \tilde{f}(x_n), \tag{12}
\]
and for each layer $k$

$$\dot{V}_k \approx V_k(t + 1) - V_k(t) = \rho(t) \sum_n e^{-\rho(t)f(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k}(t) - 2\lambda_k(t)V_k(t). \quad (13)$$

The sequence $\lambda_k(t)$ must satisfy $\lim_{t \to \infty} ||V_k|| = 1$.

Since the first term in the right hand side of Equation $13$ goes to zero with $t \to \infty$ and $\lambda$ also goes to zero, the normalized weight vectors converge at infinity with $\dot{V}_k = 0$. On the other hand, $\rho(t)$ grows to infinity.

Let us assume that starting at some time $t$, $\rho(t)$ is large enough for the following approximation to hold: $\sum_n e^{-\rho(t)f(x_n)} \approx \max_n e^{-\rho(t)f(x_n)}$. The data points with the corresponding minimum value of the margin $y_n\tilde{f}(x_n)$ are the support vectors. They are a subset of cardinality $H$ of the $N$ datapoints, all with the same margin $\eta$. In particular, the term $g(t) \approx \rho(t)e^{-\rho(t)\eta} \sum_i^{H} \frac{\partial \tilde{f}(x_n)}{\partial V_k}$.

The argument above can be made rigorous with an extension to deep nets of the main theorem in $[13]$.

Remarks

1. If we impose the conditions $||V_k|| = 1$ at each $t$, $\lambda_k(t)$ must satisfy $||V_k(t) + \rho(t)\sum_n e^{-\rho(t)f(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k} - \lambda_k(t)V_k(t)|| = 1$ where we redefined as $\lambda$ the quantity $2\lambda$. Calling $g(t) = \rho(t)\sum_n e^{-\rho(t)f(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k}$, gives

$$\lambda(t) = 1 - \sqrt{1 - ||g(t)||^2 + ||V_k^T g(t)||^2 + V^T(t)g(t)}. \quad (14)$$

2. It is possible to add a regularization term to the equation for $\dot{\rho}$. The effect of regularization is to bound $\rho(t)$ to a maximum size $\rho_{\text{max}}$, controlled by a fixed regularization parameter $\lambda_P$: in this case the dynamics of $\rho$ converges to a (very large) $\rho_{\text{max}}$ set by a (very small) value of $\lambda$.

5.2 Related techniques: weight normalization and batch normalization

As we discuss in the appendices $[19.2]$ $[19.1]$, in the continuous framework BN would be equivalent to WN. In the actual discrete setup they are not equivalent but still quite similar. A review of gradient-based algorithms with unit-norm constraints $[16]$ discusses the Lagrange multiplier method of our section $5.1$, then a coefficient normalization method that corresponds to batch normalization $[19.2]$ and finally a tangent gradient method that corresponds exactly to weight normalization $[19.1]$. Implementation and stability issues are also examined in $[16]$. The main point for this section is that the three techniques are closely related and have the same goal: performing gradient descent with a norm constraint. It seems fair to say that in the case of GD (a single minibatch including all data) the two techniques behave in a similar way.

5.3 Maximizing margin and generalization

As we mentioned there will be convergence to $\dot{V}_k = 0$ for $t \to \infty$. There may be trajectory-dependent, multiple alternative selections of the SVs during the course of the iteration while
ρ grows: each set of SVs may correspond to a max margin, minimum norm solution without being the global minimum norm solution. Because of Bezout-type arguments [9] we expect several maxima. They should generically be degenerate even under the normalization constraints – which enforce each of the K sets of $V_k$ weights to be on a unit hypersphere. Importantly, the normalization algorithm ensures control of the norm and thus of the generalization bound even if it cannot ensure that the algorithm converges to the best minimum norm solution. In summary

**Proposition 3**

The GD equations 12 and 13 converge to maximum margin solutions with fixed complexity $\mathbb{R}_N$.

### 5.4 Dynamics

In the appendices we discuss the dynamics of gradient descent in the continuous framework for a variety of losses and in the presence of regularization or normalization. Typically, normalization is similar to a vanishing regularization term. The gradient descent equations are asymptotically hyperbolic and with a degenerate Hessian.

The Lagrange multiplier case is a simple example (see Appendix 5.1). For $\dot{V}_k(t) = 0$ the following equations – as many as the number of weights – have to be satisfied asymptotically

$$V_k = \frac{g(t)}{2\lambda},$$

where $g(t) = \rho(t) \sum_n e^{-\rho(t)\tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k}$ and $\lambda(t)$ goes to zero at infinity at the same rate as $g(t)$ (see the special case of Equation 14).

The Hessian of $L$ wrt $V_k$ tells us about the linearized dynamics around the asymptotic critical point of the gradient. The Hessian (see Appendix 15)

$$\sum_n \left[ -\left( \prod_{i=1}^K \rho_i^2 \right) \frac{\partial \tilde{f}(V; x_n)}{\partial V_k} \frac{\partial \tilde{f}(V; x_n)}{\partial V_k'} + \left( \prod_{i=1}^K \rho_i \right) \frac{\partial^2 \tilde{f}(V; x_n)}{\partial V_k \partial V_k'} \right] e^{-\prod_{i=1}^K \rho_i \tilde{f}(V; x_n)} - 2\lambda(t)I.$$ (16)

is in general degenerate corresponding to an asymptotically degenerate hyperbolic equilibrium (biased towards minimum norm solutions if the rate of decay of $\lambda(t)$ implements correctly an Halpern iteration).

### 5.5 NMGD

For classification with exponential-type losses the Lagrange multiplier technique, WN and BN are trying to achieve approximately the same result – maximize the margin while constraining the norm. An even higher level unifying view of several different optimization techniques including the case of regression is to regard them as instances of Halpern iterations. Appendix 9 describes the technique. The gradient flow corresponds to an operator $T$ which is non-expansive. The fixed
Figure 1: Landscape of the empirical loss with unnormalized weights. Suppose the empirical loss above the blue line is $\leq \epsilon$. Then there are various global minima each with a different minimum norm. The corresponding picture for the normalized network shows a series of local minima with different levels of small loss. Because of the universality of deep networks from the point of view of function approximation, it seems likely that such landscape may be realizable. It is however an open question whether overparametrization may induce “nicer” landscapes.

points are degenerate. Minimization with a regularization term in the weights that vanishes at the appropriate rate (Halpern iterations) converges to the minimum norm minimizer associated to the local minimum. Halpern iterations are a form of regularization with a vanishing $\lambda(t)$ (which is the form of regularization used to define the pseudoinverse). From this perspective, the Lagrange multiplier term can be seen as a Halpern term which “attracts” the solution towards zero norm. This corresponds to a local minimum norm solution for the unnormalized network (imagine for instance in 2D that there is a surface of zero loss with a boundary as in Figure 1). The minimum norm solution in the classification case corresponds to a maximum margin solution for the normalized network. Globally optimal generalization is not guaranteed but generalization bounds such as Equation 9 are locally optimized. It should be emphasized however that it is not yet clear whether all the algorithms we mentioned implement the correct dependence of the Halpern term on the number of iterations. We will examine this issue in future work.
6 Discussion

In summary, our results imply that multilayer, nonlinear, deep networks under properly ormalized gradient descent converge to maximum margin solutions. This is similar to the situation for linear networks. Thus the prototypical (linear) example for over-parametrized deep networks is convergence of gradient descent to weights that represent the pseudoinverse of the input matrix.

We have to distinguish between square loss regression and classification via an exponential-type loss. In the case of square loss regression NMGD converges to the minimum norm solution independently of initial conditions – under the assumption that the global minimum is achieved. We were unable to prove that standard GD – without a vanishing regularization term – converges to the minimum norm solution (apart from the linear case).

Consider now the case of classification by minimization of exponential losses using the Lagrange normalization algorithm. The main result is that the dynamical system in the normalized weights converges to a solution that (locally) maximizes margin. We discuss the close relations between this algorithm and weight normalization algorithms, which are themselves related to batch normalization. All these algorithms are commonly used. The fact that the solution corresponds to a maximum margin solution under a fixed norm constraint also explains the puzzling behavior of Figure 3. The test classification error does not get worse when the number of parameters increases well beyond the number of training data because the dynamical system maximizes the margin under unit norm of $\tilde{f}$.

An additional implication of our results is that the effectiveness of batch normalization is based on more fundamental reasons than reducing covariate shifts (the properties described in [17] are fully consistent with our characterization in terms of a regularization-like effect). Controlling the norm of the weights is exactly what generalization bounds imply: normalization is the correct way to do it. Normalization is closely related to Halpern iterations used to achieve a minimum norm solution (under certain conditions).

The theoretical framework described in this paper leaves a number of important open problems. Does the empirical landscape have multiple global minima with different minimum norms? Or is the landscape “nicer” for large overparametrization – as hinted in several very recent papers (see for instance [18] and [19])? Can one ensure convergence to the global empirical minimizer with global minimum norm? How? Which conditions on the Lagrange multiplier term – and on corresponding parameters for weight and batch normalization – ensure convergence to a maximum margin solution independently of initial conditions? Furthermore, we did not explain in this paper why deep networks generalize as well as they do relative to shallow networks. We conjecture that part of the answer to this particular question lies in the approximation theory of locally hierarchical deep networks [20]: unlike shallow networks deep networks – for instance of the convolutional type – can approximate the class of hierarchically local functions without incurring in the curse of dimensionality ([21] [20]). From this point of view, the 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 a realizable setting, that is zero approximation error, with a minimum capacity.

Acknowledgments

We thank Yuan Yao, Misha Belkin and especially Sasha Rakhlin for illuminating discussions. Part of the funding is from NSF via CBMM and part is from JUMP.
References


