CENTER FOR
Brains
Minds+
Machines

CBMM Memo No. 90 **September 29, 2019**

Theory III: Dynamics and Generalization in Deep Networks[1]

Andrzej Banburski [1], Qianli Liao[1], Brando Miranda[1], Tomaso Poggio[1], Lorenzo Rosasco[1], Fernanda De La Torre[1], Jack Hidary[2]

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

Abstract

The key to generalization is controlling the complexity of the network. However, there is no obvious control of complexity – such as an explicit regularization term – in the training of deep networks. We will show that a classical form of norm control – but kind of hidden – is responsible for generalization in deep networks trained with gradient descent techniques. In particular, gradient descent induces a dynamics of the normalized weights which converges to a degenerate stationary point which corresponds to a maximum margin solution. For large, but finite, times the dynamics converges to a hyperbolic, stable minimum. Our approach extends some of the results of Srebro from linear networks to deep networks and provides a new perspective on the implicit bias of gradient descent. The elusive complexity control we describe is responsible, at least in part, for the puzzling empirical finding of good generalization despite overparametrization by deep networks.

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

[1]This replaces previous versions of Theory IIIa and TheoryIIIb.

Theory III: Dynamics and Generalization in Deep Networks*

Andrzej Banburski , Qianli Liao, Brando Miranda, Tomaso Poggio,

Lorenzo Rosasco, Fernanda De La Torre, and Jack Hidary

Alphabet (Google) X

Abstract

The key to generalization is controlling the complexity of the network. However, there is no obvious control of complexity – such as an explicit regularization term – in the training of deep networks. We will show that a classical form of norm control – but kind of hidden – is responsible for generalization in deep networks trained with gradient descent techniques. In particular, gradient descent induces a dynamics of the normalized weights which converge for $t \to \infty$ to a degenerate equilibrium which corresponds to the maximum margin solution. For sufficiently large but finite ρ – and thus fnite t – the dynamics converges to an hyperbolic minimum. Such dynamics is equivalent to regularized constrained minimization which is stable and generalizes at finite time. Our approach extends some of the results of Srebro from linear networks to deep networks and provides a new perspective on the implicit bias of gradient descent. The elusive complexity control we describe is responsible, at least in part, for the puzzling empirical finding of good generalization despite overparametrization by deep networks.

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 non-overfitting puzzle, around which several recent papers revolve (see among others [1, 2, 3, 4, 5]). This paper addresses the third question.

*This replaces previous versions of Theory III, that appeared on the CBMM site and (much more sparsely) on arXiv.

We start by reviewing recent observations on the dynamical systems induced by gradient descent methods used for training deep networks and summarize properties of the solutions they converge to. Remarkable results by [6] 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 DNNs in the setting of separable data, under exponential-type losses and square loss, for several variations of the basic gradient descent algorithm.

Our *main result* is that *unconstrained gradient descent* over an exponential-type loss – this is the usual training procedure for deep networks – *converges to solutions that for long but finite t generalize* because they are equivalent to constrained minimization or equivalently to regularization schemes (which are stable and thus generalize). Other results are:

- Because of homogeneity of the RELU, deep networks can be represented as $f(x) = \rho\tilde{f}(x)$ where ρ is the product of the norms of the weight matrix at each layer and $\tilde{f}$ is the network with normalized weights.
- Consider gradient descent algorithms – such as Lagrange multipliers methods – that minimize the exponential loss while enforcing a unit L_p norm constraint on the normalized weights. The assumption of separability and finite, fixed ρ yields convergence of the dynamics to a hyperbolic (stable) minimum.
- For $\rho \to \infty$ convergence is to stationary points which coincide with the equilibria of the dynamical system obtained from the Lagrange multiplier formulation by minimizing also on ρ .
- These asymptotic stationary points consist of solutions to maximum margin.
- Standard gradient descent used in training deep networks in the unnormalized weights followed by L_2 normalization (performed after stopping gradient descent) has the same qualitative dynamics as the Lagrange method on the weights and ρ with the same stationary points.
- Weight normalization and batch normalization have a similar qualitative dynamics and converge to the same stationary points.;

In the perspective of these theoretical results, we discuss experimental evidence around the apparent absence of "overfitting", that is the observation that the expected classification error does not get worse when increasing the number of parameters.

2 Deep networks: definitions and properties

Definitions We define a deep network with K layers with the usual coordinate-wise scalar activation functions $\sigma(z) : \quad \mathbf{R} \to \mathbf{R}$ as the set of functions $f(W; x) = W^K\sigma(W^{K-1} \cdots \sigma(W^1 x))$,

where the input is $x \in \mathbf{R}^d$, the weights are given by the matrices W^k, one per layer, with matching dimensions. We 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 paper is the ReLU activation. The norm we use is the L_2 unless we say otherwise.

Network homogeneity For ReLU activations the following positive one-homogeneity property holds $\sigma(z) = \frac{\partial \sigma(z)}{\partial z} z$. For the network this implies $f(W; x) = \prod_{k=1}^{K} \rho_k f(V_1, \cdots, V_K; x_n)$, where $W_k = \rho_k V_k$ with the matrix norm $||V_k||_p = 1$. This implies the following property of ReLU networks w.r.t. their Rademacher complexity:

$$\mathbb{R}_N(\mathbb{F}) = \rho \mathbb{R}_N(\tilde{\mathbb{F}}), \tag{1}$$

where $\rho = \rho_1 \cdots \rho_K$, $\mathbb{F}$ is the class of neural networks described above and accordingly $\tilde{\mathbb{F}}$ is the corresponding class of normalized neural networks (we call $f(V; x) = \tilde{f}(x)$ with the understanding that $f(x) = f(W; x))$[1]. In the paper we will refer to the product $\rho = \prod_{k=1}^{K} \rho_k$ of the norms of the K weight matrices of f. Thus $f = \rho \tilde{f}$. Note that

$$\frac{\partial f}{\partial \rho_k} = \frac{\rho}{\rho_k} \tilde{f} \tag{2}$$

and that the definitions of ρ_k, V_k and $\tilde{f}$ all depend on the choice of the norm used in normalization.

Structural property The following structural property of the gradient of deep ReLU networks is sometime useful (Lemma 2.1 of [7]):

$$\sum_{i,j} \frac{\partial f(x)}{\partial W_k^{i,j}} W_k^{i,j} = f(x); \tag{3}$$

for $k = 1, \cdot, K$. Equation 3 can be rewritten as an inner product

$$(W_k, \frac{\partial f(x)}{\partial W_k}) = f(x) \tag{4}$$

where W_k is here the vectorized representation of the weight matrices W_k for each of the different layers (each matrix is a vector)[2]. The same property holds for V_k

[1] This invariance property of the function f under transformations of W_k that leaves the product norm the same is typical of ReLU (and linear) networks.

[2] By taking derivatives of both sides of Equation 4, we obtain to the following property

$$(W_k, \frac{\partial^2 f(x)}{\partial {W_k}^2}) = 0. \tag{5}$$

From Equation 3, it follows that the condition $\left(\frac{\partial f(x)}{\partial W_k}\right) = 0$ implies $f(x) = 0$. In the case of square loss, this condition restricts the non-fitting stationary points of the gradient to be linear combinations $\sum_{n=1}^{N} (f(x_n) - y_n) \frac{\partial f}{\partial W_k} = 0$, with $\frac{\partial f}{\partial W_k} \neq 0, \forall k$ or $f(x) = 0$. A similar restriction also holds for the exponential loss (see Equation 48).

$$(V_k, \frac{\partial \tilde{f}(x)}{\partial V_k}) = \tilde{f}(x) \tag{6}$$

Gradient flow and continuous approximation The gradient flow of the empirical risk L is often written as

$$\dot{W} \equiv \frac{dW}{dt} = -\gamma(t)\nabla_W(L(f)), \tag{7}$$

where $\gamma(t)$ is the learning rate (in this paper we will neglect it). We are well aware that the the continuous formulation and the discrete one are not equivalent but we are happy to leave a careful analysis – especially of the discrete case – to better mathematicians.

We conjecture that the hypothesis of smooth activations is just a technicality due to the necessary conditions for existence and uniqueness of solutions to ODEs. Generalizing to differential inclusions and non-smooth dynamical systems should allows for these conditions to be satisfied in the Filippov sense [8]. The Clarke subdifferential is supposed to deal appropriately with functions such as RELU.

Separability When $y_n f(x_n) > 0 \; \forall n = 1, \cdots, N$ we say that the data are separable wrt $f \in \mathbf{F}$, that is they can all be correctly classified. We assume in this paper that there exist T_0 such that for $t > T_0$ gradient descent attains a f that seprates the data. Notice that this is a strong condition on the data if f is linear but it is a weak assumption in the case of overparametrized deep networks. In fact here we mostly assume that the condition of *separability is reached during gradient descent* by the networks we consider.

Minima at infinity, $\rho \to \infty$ One may informally speak of minima at infinity – and other behavior at infinity (of course this does not make sense because minima of a continuous function are guranteed to exist only in a compact domain).

3 Related work

There are many recent papers studying optimization and generalization in deep learning. For optimization we mention work based on the idea that noisy gradient descent [9, 10, 11, 12] can find a global minimum. More recently, several authors studied the dynamics of gradient descent for deep networks with assumptions about the input distribution or on how the labels are generated. They obtain global convergence for some shallow neural networks [13, 14, 15, 16, 17, 18]. Some local convergence results have also been proved [19, 20, 21]. The most interesting such approach is [18], which focuses on minimizing the training loss and proving that randomly initialized gradient descent can achieve zero training loss (see also [22, 23, 24]). In summary, there is by now an extensive literature on optimization that formalizes and refines to different special cases and to the discrete domain our results of Theory II and IIb (see section 6).

For generalization, which is the topic of this paper, existing work demonstrate that gradient descent works under the same situations as kernel methods and random feature methods [25, 26, 27]. Closest to our approach – which is focused on the role of batch and weight

normalization – is the paper [28]. 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 batch normalization, in fact maximize margin while controlling the complexity of the classifier without the need to add a regularizer or to use weight decay. In fact, we will show that even standard gradient descent on the weights implicitely controls the complexity of the normalized weights.

Very recently, well after previous versions of this work, two papers ([29] and [30]) appeared. They develop an elegant but complicated margin maximization based approach, describing the relations between the *margin, the constrained and the optimization paths* deriving some of the same results of this section (and more). Our approach does not need the notion of maximum margin but our theorem 5 establishes a connection with it and thus with the results of [29] and [30]. Our main focus here (and in [31]) is on the puzzle of how complexity of deep nets is controlled during training despite overparametrization and despite the absence of regularization. Our main original contribution is a study of the *gradient flow of the normalized weights* to characterize the implicit *control responsible for generalization* in deep networks trained under the exponential loss, which describe as implicit L_2 normalization by gradient descent. In the process of doing this, we analyze the dynamics of the flow of *the direction of the weights* induced by gradient descent on the unnormalized weights.

4 Main results

The standard approach to training deep networks is to find the weights W_k that minimize the empirical exponential loss $L = \sum_n e^{-f(x_n)}$. Here we study three related versions of this problem:

1. the minimization of $L = \sum_n e^{-\rho \tilde{f}(x_n)}$ under the constraint $||V_k|| = 1$ wrt V_k for fixed ρ;
2. the minimization of $L = \sum_n e^{-\rho \tilde{f}(x_n)}$ under the constraint $||V_k|| = 1$ wrt V_k, ρ;
3. the minimization of $L = \sum_n e^{-\rho \tilde{f}(x_n)} = L = \sum_n e^{-f(x_n)}$ wrt V_k, ρ, which is the standard situation for deep nets.

4.1 Constrained minimization of the exponential loss

Generalization bounds suggest constrained optimization of the exponential loss that is to minimize $L = \sum_n e^{-\rho \tilde{f}(x_n)}$ under the constraint $||V_k|| = 1$ which leads to minimize

$$L = \sum_n e^{-\rho \tilde{f}(x_n)} + \sum_k \lambda_k ||V_k||^2 \tag{8}$$

with λ_k such that the constraint $||V_k| = 1$ is satisfied.

4.2 Fixed ρ: hyperbolic minima

Gradient descent on L for fixed ρ wrt V_k yields the dynamical system

$$\dot{V_k} = \rho \sum_n e^{-\rho \tilde{f}(x_n)} (\frac{\partial \tilde{f}(x_n)}{\partial V_k} - V_k \tilde{f}(x_n)) \tag{9}$$

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. They – assuming data separation is achieved – satisfy

$$\sum_n e^{-\rho \tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k} = \sum_n e^{-\rho \tilde{f}(x_n)} V_k \tilde{f}(x_n). \tag{10}$$

The stationary points provided by Equations 10 are in fact *hyperbolic minima* because the Hessian of L is negative definite at the stationary points (see Appendix 13).

4.3 $\rho \to \infty$ has same stationary points as full dynamical system from Lagrange multipliers

Consider the limit of $\rho \to \infty$ in Equation 10 . The asymptotic stationary points of the flow of V_k then satisfy

$$\sum_n e^{-\rho \tilde{f}(x_n)} (\frac{\partial \tilde{f}(x_n)}{\partial V_k} - V_k \tilde{f}(x_n)) = 0 \tag{11}$$

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

$$\sum_n e^{-R \tilde{f}(x_n)} (\frac{\partial \tilde{f}(x_n)}{\partial V_k} - V_k \tilde{f}(x_n)) = 0. \tag{12}$$

Notice that we can write

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

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

$$\begin{aligned} e^{-R\tilde{f}(x_N)}[H_N + e^{-R\Delta} \sum_n^{N-1} H_n] &\geq e^{-R\tilde{f}(x_N)} H_N + \sum_n^{N-1} e^{-R\tilde{f}(x_n)} H_n \\ &= \sum_n^N e^{-R\tilde{f}(x_n)} H_n \end{aligned}$$

where we define $\Delta = \tilde{f}(x^{**}) - \tilde{f}(x^*)$ [3] with $\tilde{f}(x_N) = min_{n\in 1,N}\tilde{f}(x_n)$ as the margin of $\tilde{f}$ and $\tilde{f}(x_{N-1}) = min_{n\in 1,N-1}\tilde{f}(x_n)$ as the second smallest margin in the data set.

The stationary point equation has the form $\epsilon H_N + \epsilon^2 H = 0$; 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 not infinite 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^* provides the maximum margin. Before that limit is reached, the solution V_k changes with increasing ρ.

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

$$\dot{\rho_k} = \frac{\rho}{\rho_k}\sum_n e^{-\rho\tilde{f}(x_n)}\tilde{f}(x_n) \quad \dot{V_k} = \rho[\sum_n e^{-\rho\tilde{f}(x_n)}\frac{\partial\tilde{f}(x_n)}{\partial V_k} + V_k\tilde{f}(x_n)] \tag{14}$$

Observe that after onset of separability $\dot{\rho_k} > 0$ with $\lim_{t\to\infty}\dot{\rho_k} = 0$, $\lim_{\to\infty}\rho(t) = \infty$ (for one layer $\rho \propto \log t$; for more layer it is faster, see Appendix 12). Appendix 12 shows that $\rho(t)$ is a monotonically increasing function from $t = 0$ to $t = \infty$. Thus for any large R there exist T such that $\rho(T) = R$. At time T then, condition for the stationary point of the V_k in Equation 14 satisfies to

$$\dot{V_k} = R\sum_n e^{-R\tilde{f}(x_n)}[\frac{\partial\tilde{f}(x_n)}{\partial V_k} + V_k\tilde{f}(x_n)] = 0 \tag{15}$$

which coincides with Equation 11. Then the same arguments can be used to conclude that the limit for $t \to \infty$ satisfy the same Equations $[(\frac{\partial \tilde{f}(x_*)}{\partial V_k} - V_k\tilde{f}(x_*)) = 0$.

4.4 Asymptotic stationary points coincide with maximum margin (alternative proof)

The previous section concludes that the asymptotic stationary points coincide with maximum margin. An alternative proof follows from the fact that Equations 14 solve the optimization problem 8. Appendix 8 shows that the minimizer of Equation 14 is the maximum margin solution. Thus the *asymptotic stationary points* characterized by Equation 11 *must coincide with maximum margin solutions.*

Another proof can be obtained as follows. Let us define $\eta(f) = \min_n f(x_n)$ as the margin and $\max_{||V_k||=1}\eta(\tilde{f})$ as the maximum margin. Then notice that the limit $\lim_{\rho\to\infty}\sum_n e^{-\rho\tilde{f}(x_n)}G(\tilde{f}(x_n)) \propto \min_n\sum_n G(\max_{||V_k||=1}\eta(\tilde{f}(x_n) \propto \max_{||V_k||=1}\eta(\tilde{f})$ gives the maximum margin. Thus the flow of V_k converges for $\rho \to \infty$ to a stationary point $\dot{V_k} = 0$ which corresponds to a maximum margin solution[4].

[3] If $\Delta = 0$ then all data points are support vectors with the same margin.

[4] Still another proof of the same statement can be obtained using the same argument used in the previous subsection, by rewriting the limit 11 as

$$\lim_{\rho\to\infty} e^{-\rho\tilde{f}(x_*)}[(\frac{\partial\tilde{f}(x_*)}{\partial V_k} - V_k\tilde{f}(x_*)) + Ne^{-\rho\Delta}] = \lim_{\rho\to\infty}\sum_n e^{-\rho\tilde{f}(x_n)}(\frac{\partial\tilde{f}(x_n)}{\partial V_k} - V_k\tilde{f}(x_n)) = 0 \tag{16}$$

4.5 Standard gradient descent for deep networks: generalization without explicit unit norm constraints

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, extending some of them – though not his detailed analysis – to the deep network case. It also throws some light on the nature of the implicit bias or hidden complexity control.

We show this result next.

4.5.1 Reparametrization of standard gradient descent

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):

- The norm $||\cdot||$ is assumed in this section to be the L_2 norm.
- 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. $\frac{\partial ||w||_2}{\partial w} = v$
  2. Define $S = I - vv^T = I - \frac{ww^T}{||w||_2^2}$. S has at most one zero eigenvalue since vv^T is rank 1 with a single eigenvalue $\lambda = 1$. This means also $S \geq 0$, as can be seen directly.
  3. $\frac{\partial v}{\partial w} = \frac{S}{\rho}$.
  4. $Sw = Sv = 0$
  5. $S^2 = S$
  6. 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 of Equation is given by

where $\tilde{f}(x_*) = min_n \tilde{f}(x_n)$ and $\Delta = \tilde{f}(x^*) - \tilde{f}(x^{**})$ where $\tilde{f}(x^{**})$ is the second smallest margin in the data set[5]. Thus the stationarity condition in the limit at infinity 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^* provides the maximum margin.

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

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:

$$\dot{\rho_k} = \frac{\partial ||W_k||}{\partial W_k} \frac{\partial W_k}{\partial t} = V_k^T \dot{W}_k \tag{18}$$

and

$$\dot{V_k} = \frac{\partial V_k}{\partial W_k} \frac{\partial W_k}{\partial t} = \frac{S_k}{\rho_k} \dot{W}_k \tag{19}$$

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

$$\dot{\rho_k} = V_k^T \dot{W}_k = \sum_{n=1}^{N} V_k^T \frac{\partial f(x_n; W)}{\partial W_k} e^{-f(x_n; W)} = \frac{\rho}{\rho_k} \sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n)} \tilde{f}(x_n) \tag{20}$$

and

$$\dot{V_k} = \frac{\rho}{\rho_k^2} \sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n)} (\frac{\partial \tilde{f}(x_n)}{\partial V_k} - V_k \tilde{f}(x_n)). \tag{21}$$

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

Clearly the dynamics of *unconstrained gradient descent* and the dynamics of *constrained gradient descent* are very similar since they differ 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 $t > 0$ $\rho(t) > 0$.

4.5.2 Constrained optimization and weight normalization

We recall that *constrained gradient descent* using Lagrange multipliers yields the dynamical system

$$\dot{\rho_k} = \frac{\rho}{\rho_k} \sum_n e^{-\rho \tilde{f}(x_n)} \tilde{f}(x_n) \quad \dot{V_k} = \rho \sum_n e^{-\rho \tilde{f}(x_n)} (\frac{\partial \tilde{f}(x_n)}{\partial V_k} + V_k \tilde{f}(x_n)). \tag{22}$$

Constrained normalization by tangent gradient gives the same dynamical system as expected. The Appendix shows that it coincides with the so-called weight normalization algorithm.

4.5.3 Main result: unconstrained gradient descent generalizes

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 for which stability and generalization is guaranteed. Since the unconstrained case converges to the same solutions it will also generalize.

4.6 Summary

The following theorem summarizes our main results

Theorem 1 *Assume that separability is reached at time T_0 during gradient descent, that is $y_n f(x_n) > 0, \quad \forall n$. Then unconstrained gradient descent converges to a solution that generalizes. In addition the following facts 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 dynamics has stationary points for $t \to \infty$ that coincide with the limit for $\rho \to \infty$ in the dynamics (A).*

3. *For $\rho \to \infty$ in (A) and for $t \to \infty$ in (B) the stationary points of V_k are maxima of the margin.*

4. *The standard deep networks dynamics converges to the same stationary points of the flow of V_k as (A) and (B).*

5. *Weight normalization has similar qualitative dynamics and exactly the same stationary points as standard deep networks dynamics.*

In the Appendix 14 we describe analysis of convergence for linear networks.

5 Discussion

Our main results are *for classification* in the setting of *separable data* for *continous gradient flow* under *the exponential loss*:

- Perhaps not surprisingly, for classification, under an exponential loss, there is an *implicit L_2 norm constraint on the V_k dynamics in standard gradient descent for deep networks*

with RELUs. Therefore standard gradient descent on the weights, provides a solution $\tilde{f}$ that generalizes without the need of additional explicit regularization or explicit norm constraints.

- In fact, the dynamics of the normalized weights in *unconstrained gradient descent* on the exponential loss has *the same asymptotic stationary points as gradient descent on the regularized loss (with vanishing λ).*

- gradient descent methods with a L_p norm constraint minimizing an exponential-type loss implement a classical recipe for generalization. They converge to stationary points of the gradient of V_k which attain zero loss – assuming separability occurs during gradient descent – for $\rho \to \infty$.

- The equilibrium point of the flow of V_k for $\rho \to \infty$ corresponds to a maximum margin solution of the constrained optimization problem.

- Examples of techniques commonly used to train over-parametrized, multilayer, RELU, deep networks are *weight normalization* and *batch normalization* enforcing an explicit unit constraint in the L_2 norm of V_k.

- For linear networks the convergence rate of standard GD ($\frac{1}{\log t}$) is slower than the convergence rate ($\frac{1}{t^{\log(\sqrt{(t)}}}$) of weight normalization.

- The dynamics of W_k for separable data attain the global zero minimum of the loss only for $||W_k|| \to \infty$ but diverges in W. The dynamics of the direction of the weights V_k in a multilayer network has hyperbolic quasi-minima for any finite ρ which is sufficiently large.

- Some of these results *apply to kernel machines* for the exponential loss under the separability/interpolation assumption.

Of course the gradient flows corresponding to normalization with different L_p norms are expected to be different and to converge to different solutions, as in the classical case of support vector machines with L_2 vs L_1 regularizers. It is useful to emphasize that despite the similarities between some of the methods enforcing unit constraints in the 2-norm, they usually correspond to different dynamical flows but with the same qualitative dynamics and the same stationary solutions. In particular, batch normalization and weight normalization are not the same and in turn they are slightly different from standard gradient descent. Furthermore, our analysis has been restricted to the continous case; the discrete case may yield greater differences.

The fact that the solution corresponds to a maximum margin or minimum norm solution may explain the puzzling behavior of Figures in the Supplementary Material, in which batch normalization was used. 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 is trying to maximize the *margin* under unit norm of $\tilde{f}$.

An additional implication of our results is that *weight normalization and batch normalization are enforcing a natural constraint (on the dynamics of V_k) for generalization*, deeper than reducing covariate shifts (the properties described in [32] are fully consistent with our characterization in terms of a norm constraint). Controlling the norm of the weights is what the generalization bounds suggest. Normalization is closely related to Halpern iterations used to achieve a minimum norm solution, in the same way that the Lagrange multiplier technique is related to the tangent gradient method. Notice that as in the case of the of the vanishing regularizer assumed in the original theorem of [33], our *Lagrange multipliers λ_k should go to zero fast enough for $\lambda_k \rho_k$ to go to zero while $\rho_k \to \infty$.*

In practice, this basic complexity control may be enhanced by additional mechanisms that improve generalization. For instance, high dimensionality under certain conditions has been shown to lead to better generalization for certain interpolating kernels [34, 35] and [36]. Though this is still an open question, it seems possible that similar results may be valid for deep networks under the exponential loss. In addition, commonly used weight decay with appropriate parameters can induce generalization since it is equivalent to regularization. Furthermore, typical implementations of data augmentation may also effectively avoid overparametrization: at each iteration of SGD only "new" data are used and depending on the number of iterations it is quite possible that the size of the training data exceeds the number of parameters. Within this online framework, one expects convergence to the minimum of the expected risk (see Supplementary Material section on Data Augmentation) without the need to invoke generalization bounds.

Our results assume densely connected deep networks. There are of course several open problems. Does the empirical landscape have multiple global minima with different minimum norms, as we suspect? Is the landscape "nicer" for very large overparametrization – as hinted in several recent papers (see for instance [37] and [38])? Can one ensure convergence to the global empirical minimizer with global minimum norm? What is the theoretical explanation of the empirical observation that batch normalization is better than weight normalization? This requires some explanation because both enforce implicitely or explicitely a unit constraint in the L_2 norm.

Acknowledgments

We thank Yuan Yao, Misha Belkin, Jason Lee and especially Sasha Rakhlin for illuminating discussions. Part of the funding is from 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.

References

[1] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. *CoRR*, abs/1611.04231, 2016.

[2] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. *arXiv:1706.08947*, 2017.

[3] Jure Sokolic, Raja Giryes, Guillermo Sapiro, and Miguel Rodrigues. Robust large margin deep neural networks. *arXiv:1605.08254*, 2017.

[4] P. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. *ArXiv e-prints*, June 2017.

[5] C. Zhang, Q. Liao, A. Rakhlin, K. Sridharan, B. Miranda, N.Golowich, and T. Poggio. Musings on deep learning: Optimization properties of SGD. *CBMM Memo No. 067*, 2017.

[6] D. Soudry, E. Hoffer, and N. Srebro. The Implicit Bias of Gradient Descent on Separable Data. *ArXiv e-prints*, October 2017.

[7] Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. *CoRR*, abs/1711.01530, 2017.

[8] F.M. Arscott and A.F. Filippov. *Differential Equations with Discontinuous Righthand Sides: Control Systems*. Mathematics and its Applications. Springer Netherlands, 1988.

[9] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. *CoRR*, abs/1703.00887, 2017.

[10] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. *CoRR*, abs/1503.02101, 2015.

[11] Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1246–1257, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

[12] Simon S. Du, Jason D. Lee, and Yuandong Tian. When is a convolutional filter easy to learn? In *International Conference on Learning Representations*, 2018.

[13] Yuandong Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 3404–3413. JMLR.org, 2017.

[14] M. Soltanolkotabi, A. Javanmard, and J. D. Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, Feb 2019.

[15] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 597–607, USA, 2017. Curran Associates Inc.

[16] Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 605–614, 2017.

[17] Simon Du, Jason Lee, Yuandong Tian, Aarti Singh, and Barnabas Poczos. Gradient descent learns one-hidden-layer CNN: Don't be afraid of spurious local minima. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1339–1348, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[18] Simon S. Du, Jason D. Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. *CoRR*, abs/1811.03804, 2018.

[19] Kai Zhong, Zhao Song, Prateek Jain, Peter L. Bartlett, and Inderjit S. Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 4140–4149. JMLR.org, 2017.

[20] Kai Zhong, Zhao Song, and Inderjit S. Dhillon. Learning non-overlapping convolutional neural networks with multiple kernels. *CoRR*, abs/1711.03440, 2017.

[21] X. Zhang, Y. Yu, L. Wang, and Q. Gu. Learning One-hidden-layer ReLU Networks via Gradient Descent. *arXiv e-prints*, June 2018.

[22] Yuanzhi Li and Yingyu Liang. Learning overparameterized neural networks via stochastic gradient descent on structured data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8157–8166. Curran Associates, Inc., 2018.

[23] Simon S. Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

[24] Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Stochastic gradient descent optimizes over-parameterized deep relu networks. *CoRR*, abs/1811.08888, 2018.

[25] Amit Daniely. Sgd learns the conjugate kernel class of the network. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2422–2430. Curran Associates, Inc., 2017.

[26] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *CoRR*, abs/1811.04918, 2018.

[27] Sanjeev Arora, Simon S. Du, Wei Hu, Zhi yuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *CoRR*, abs/1901.08584, 2019.

[28] Colin Wei, Jason D. Lee, Qiang Liu, and Tengyu Ma. On the margin theory of feedforward neural networks. *CoRR*, abs/1810.05369, 2018.

[29] Mor Shpigel Nacson, Suriya Gunasekar, Jason D. Lee, Nathan Srebro, and Daniel Soudry. Lexicographic and Depth-Sensitive Margins in Homogeneous and Non-Homogeneous Deep Models. *arXiv e-prints*, page arXiv:1905.07325, May 2019.

[30] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. *CoRR*, abs/1906.05890, 2019.

[31] A. Banburski, Q. Liao, B. Miranda, T. Poggio, L. Rosasco, B. Liang, and J. Hidary. Theory of deep learning III: Dynamics and generalization in deep networks. *CBMM Memo No. 090*, 2019.

[32] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How Does Batch Normalization Help Optimization? *arXiv e-prints*, page arXiv:1805.11604, May 2018.

[33] Saharon Rosset, Ji Zhu, and Trevor Hastie. Margin maximizing loss functions. In *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 1237–1244, 2003.

[34] Tengyuan Liang and Alexander Rakhlin. Just Interpolate: Kernel "Ridgeless" Regression Can Generalize. *arXiv e-prints*, page arXiv:1808.00387, Aug 2018.

[35] Alexander Rakhlin and Xiyu Zhai. Consistency of Interpolation with Laplace Kernels is a High-Dimensional Phenomenon. *arXiv e-prints*, page arXiv:1812.11167, Dec 2018.

[36] Mikhail Belkin, Daniel Hsu, and Ji Xu. Two models of double descent for weak features. *CoRR*, abs/1903.07571, 2019.

[37] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A Mean Field View of the Landscape of Two-Layers Neural Networks. *arXiv e-prints*, page arXiv:1804.06561, Apr 2018.

[38] Phan-Minh Nguyen. Mean Field Limit of the Learning Dynamics of Multilayer Neural Networks. *arXiv e-prints*, page arXiv:1902.02880, Feb 2019.

[39] C. Zhang, Q. Liao, A. Rakhlin, K. Sridharan, B. Miranda, N.Golowich, and T. Poggio. Theory of deep learning IIb: Optimization properties of SGD. *CBMM Memo 072*, 2017.

[40] M. Raginsky, A. Rakhlin, and M. Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: A nonasymptotic analysis. *arXiv:180.3251 [cs, math]*, 2017.

[41] T. Poggio and Q. Liao. Theory II: Landscape of the empirical risk in deep learning. *arXiv:1703.09833, CBMM Memo No. 066*, 2017.

[42] O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. pages 169–207, 2003.

[43] Peter L. Bartlett, Michael I. Jordan, and Jon D. McAuliffe. Convexity, classification, and risk bounds. Technical report, 2003.

[44] S. C. Douglas, S. Amari, and S. Y. Kung. On gradient adaptation with unit-norm constraints. *IEEE Transactions on Signal Processing*, 48(6):1843–1847, June 2000.

[45] Tim Salimans and Diederik P. Kingm. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *Advances in Neural Information Processing Systems*, 2016.

[46] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[47] Simon S Du, Wei Hu, and Jason D Lee. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 384–395. Curran Associates, Inc., 2018.

[48] Paulo Jorge S. G. Ferreira. The existence and uniqueness of the minimum norm solution to certain linear and nonlinear problems. *Signal Processing*, 55:137–139, 1996.

Appendix

6 The optimization landscape of Deep RELU Networks under exponential-type loss

The *first part* of the argument of this section relies on the simple observation that RELU networks, under the hypothesis of an exponential-type loss function, do not have zeros of the gradient (wrt the W_k) that separate the data. In fact, under the hypothesis of an exponential-type loss, separable data and homogeneity of the network – such as kernel machines and deep RELU networks – the only stationary points of the gradient that separate the data are for $\rho = \infty$.

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 being 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: for instance in the linear case the Hessian is proportional to $\sum_n^N x_n x_n^T$.

Consider now that the local minima which are not global minima must missclassify. How degenerate are they? In the case of a linear network in the exponential loss case, assume there is a finite w for which the gradient is zero in some of its components. One question is whether this is similar to the regularization case or not, that is whether *misclassification regularizes*.

Let us look at a linear example:

$$\dot{w} = F(w) = -\nabla_w L(w) = \sum_{n=1}^{n} x_n^T e^{-x_n^T w} \tag{23}$$

in which we assume that there is one classification ' error (say for $n = 1$), meaning that the term $e^{-x_1^T w}$ grows exponentially with w. Let us also assume that gradient descent converges to w^*. This implies that $\sum_{n=2}^{n} x_n^T e^{-x_n^T w^*} = -x_1^T e^{-x_1^T w^*}$: for w^* the gradient is zero and $\dot{w} = 0$. Is this a hyperbolic equilibrium? Let us look at a very simple $1D$, $n = 2$ case:

$$\dot{w} = -x_1 e^{x_1 w^*} + x_2 e^{-x_2 w^*} \tag{24}$$

If $x_2 > x_1$ then $\dot{w} = 0$ for $e^{(x_1+x_2)w^*} = \frac{x_2}{x_1}$ which implies $w^* = \frac{\log(\frac{x_2}{x_1})}{x_1+x_2}$. This is clearly a hyperbolic equilibrium point, since we have

$$\nabla_w F(w) = -x_1^2 e^{x_1 w^*} - x_2^2 e^{-x_2 w^*} < 0, \tag{25}$$

so the single eigenvalue in this case has no zero real part.

In general, if there are only a small number of classification errors, one expects a similar situation for some of the components. *Differently from the regularization case, misclassification errors do not "regularize" all components of w but only the ones in the span of the misclassified examples.*

The more interesting case is with $D > N$. An example of this case is $D = 3$ and $N = 2$ in the above equation. The Hessian wrt W_k at the minimum will be degenerate with at least one zero eigenvalue ad one negative eigenvalue.

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. We thus have the following rather obvious result:

Theorem 2 *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.*

The stationary points of the gradient of f in the nonlinear multilayer separable case under exponential loss are given by

$$\sum_{n=1}^{N} y_n \frac{\partial f(x_n; w)}{\partial W_k^{i,j}} e^{-y_n f(x_n;W)} = 0. \tag{26}$$

This means that the global zeros of the loss are at infinity, that is for $\rho \to \infty$ in the exponential. 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*.

The *second part* of our argument (in [39]) is that SGD concentrates on the most degenerate minima. The argument is based 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 [40]. In addition, more informally, there is a certain similarity between SGD and SGDL suggesting that in practice the solution of SGD may be similar to the solution of SGDL. The Boltzman distribution is

$$p(W_k^{i,j}) = \frac{1}{Z} e^{-\frac{L(f)}{T}}, \tag{27}$$

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 [39]. Taken together, these two facts suggest that SGD selects degenerate minimizers corresponding to larger isotropic flat regions of the loss. Then SDGL shows concentration – *because of the high dimensionality* – of its asymptotic distribution Equation 27.

Together [41] and [39] imply the following

Conjecture *For overparametrized deep networks under an exponential-type loss, SGD selects with high probability global minimizers of the empirical loss, which are fully degenerate (for $\rho \to \infty$, in the separable case.*

7 Uniform convergence bounds: minimizing a surrogate loss under norm constraint

Classical *generalization bounds for regression* [42] suggest that minimizing the empirical loss of a loss function such as the cross-entropy subject to constrained *complexity of the minimizer* is a way to to attain generalization, that is an expected loss close to the empirical loss:

Theorem 3 *The following generalization bounds apply to* $\forall f \in \mathbb{F}$ *with probability at least* $(1-\delta)$*:*

$$L(f) \leq \hat{L}(f) + c_1 \mathbb{R}_N(\mathbb{F}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}} \tag{28}$$

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

Thus minimizing under a constraint on the Rademacher complexity a surrogate function such as the cross-entropy (which becomes the logistic loss in the binary classification case) will minimize an upper bound on the expected classification error because such surrogate functions are upper bounds on the $0-1$ function[6]. Calling $\rho\tilde{f} = f$, using the homogeneity of the network, one can use the following version of the bound above:

Theorem 4 $\forall f \in \mathbb{F}$ *with probability at least* $(1-\delta)$*:*

$$L(\rho\tilde{f}) \leq \hat{L}(\rho\tilde{f}) + \rho \mathbb{R}_N(\tilde{\mathbb{F}}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}} \tag{29}$$

and to use ρ effectively as a parameter (in the Ivanov sense) trading off generalization error with the quality of L in being a proxy for the $0-1$ error.

In this setup, $\tilde{f}$ is obtained by minimizing the exponential loss for $\rho \to \infty$ under a unit norm constraint on the weight matrices of $\tilde{f}$:

$$\lim_{\rho \to \infty} \arg \min_{||V_k||=1,\ \forall k} L(\rho\tilde{f}) \tag{30}$$

As it will become clear later, gradient descent techniques on the exponential loss automatically increase ρ to infinity.

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

[6] Furtermore the excess classification risk $R(f) - R^*$, where $R(f)$ is the classification error associated with f and R^* is the Bayes error [43], can be bounded by a monotonically increasing function of appropriate surrogate functions such as the exponential and the cross-entropy.

8 Constrained minimization of the exponential loss implies margin maximization

Though not critical for our approach to the question of generalization in deep networks it is interesting to observe that constrained minimization of the exponential loss implies margin maximization. This property relates our approach to the results of several recent papers [6, 29, 30]. Notice that our theorem 5 as in [33] is a *sufficient condition for margin maximization.* Sufficiency is not true for general loss functions. In fact [29] seems to require additional conditions for ensuring that the margin path converges to the optimization path.

To state the margin property more formally, we adapt to our setting a different result due to [33] (they consider a vanishing λ regularization term whereas we have a unit norm constraint). First we recall the definition of the empirical loss $L(f) = \sum_{n=1}^{N} \ell(y_n f(x_n))$ with an exponential loss function $\ell(yf) = e^{-yf}$. We define $\eta(f)$ a the *margin* of f, that is $\eta(f) = \min_n f(x_n)$.

Then our margin maximization theorem takes the form

Theorem 5 *Consider the set of $V_k, k = 1, \cdots, K$ corresponding to*

$$\min_{||V_k||=1} L(f(\rho_k, V_k)) \tag{31}$$

where the norm $||V_k||$ is a chosen L_p norm and $L(f)(\rho_k, V_K) = \sum_n \ell(y_n \rho f(V; x_n))$ is the empirical exponential loss. Consider a sequence of increasing ρ. Then the associated sequence of V_k defined by Equation 31 i, converges for $\rho \to \infty$ to the maximum margin of $\tilde{f}$, that is to $\max_{||V_k||\leq 1} \eta(\tilde{f})$.

This means that for $\rho \to \infty$ the weights V that minimize the exponential loss under a unit norm constraint converge to the weights V that maximize the margin of $\tilde{f}$. Thus

$$\arg\min_{||V_k||=1,\rho} L(\rho\tilde{f})(x_n) \to \arg \max_{||V_k||=1} \min_n \tilde{f}(x_n) \tag{32}$$

Proof The proof loosely follows [33, 6], see also [28]. We observe that $\min_{||V_k||=1} L(f(\rho))$ exists because $L(f(\rho))$ is continous since f is continuous and the domain is compact for any finite ρ. We now assume that V^* minimize $L(f(\rho))$. We claim that the associated network $\tilde{f}^*$ maximizes the margin for $\rho \to \infty$. Arguing by contradiction assume that for a given ρ there exist a different $\tilde{f}_1$ with a larger margin, that is $\eta(\tilde{f}_1 > \eta(\tilde{f}^*)$. Then I can choose any $\tilde{f}_2$ with weigths V_2 such that $||V_2 - V^*|| \leq \delta$ such that $\eta(\tilde{f}_2) < \eta(\tilde{f}_1) - \epsilon$. Now I can choose a ρ large enough so that $\rho\tilde{f}_1$ has a smaller loss then $\rho\tilde{f}_2$, implying that $\tilde{f}^*$ cannot be a convergence point. The last step is based on the fact that *if $\eta(\tilde{f}_1) > \eta(\tilde{f}_2)$, then $L(\rho\tilde{f}_1) < L(\rho\tilde{f}_2)$ for large enough ρ.*

This follows from the existence of ρ such that $L(\rho\tilde{f}_1) \leq Ne^{-\rho\eta(\tilde{f}_1)} \leq e^{-\rho\eta(\tilde{f}_2)} \leq L(\rho\tilde{f}_2)$.

Theorem 5 can be supplemented with the following lemma, proved in Appendix 9:

Lemma 6 *Margin maximization of the network with normalized weights is equivalent to norm minimization under strict separability ($y_n f(x_n) \geq 1$).*

9 Minimal norm and maximum margin

We discuss the connection between maximum margin and minimal norms problems in binary classification. To do so, we reprise some classic reasonings used to derive support vector machines. The norm assumed in this section is the L_2 norm though the proof can be extended. We show they directly extend beyond linearly parametrized functions as long as there is a one-homogeneity property, namely, for all $\alpha > 0$,

$$f(\alpha W; x) = \alpha f(W; x)$$

Given a training set of N data points $(x_i, y_i)_{i=1}^N$, where labels are ± 1, the functional margin is

$$\min_{i=1,\dots,N} y_i f(W; x_i). \tag{33}$$

If there exists W such that the functional *margin is strictly positive*, then the training set is separable. We assume in the following that this is indeed the case. The maximum (max) margin problem is

$$\max_W \min_{i=1,\dots,N} y_i f(W; x_i), \qquad \text{subj. to} \quad \|W\| = 1. \tag{34}$$

The latter constraint is needed to avoid trivial solutions in light of the one-homogeneity property. We next show that Problem (34) is equivalent to

$$\min_W \frac{1}{2}\|W\|^2, \qquad \text{subj. to} \quad y_i f(W; x_i) \geq 1, \qquad i = 1, \dots, N. \tag{35}$$

To see this, we introduce a number of equivalent formulations. First, notice that functional margin (33) can be equivalently written as

$$\max_{\gamma > 0} \gamma, \qquad \text{subj. to} \quad y_i f(W; x_i) \geq \gamma, \qquad i = 1, \dots, N.$$

Then, the max margin problem (34) can be written as

$$\max_{W, \gamma > 0} \gamma, \qquad \text{subj. to} \quad \|W\| = 1, \quad y_i f(W; x_i) \geq \gamma, \qquad i = 1, \dots, N. \tag{36}$$

Next, we can incorporate the norm constraint noting that using one-homogeneity,

$$y_i f(W; x_i) \geq \gamma \Leftrightarrow y_i f(\frac{W}{\|W\|}; x_i) \geq \gamma' \Leftrightarrow y_i f(W; x_i) \geq \|W\|\gamma = \gamma'$$

so that Problem (36) becomes

$$\max_{W, \gamma' > 0} \frac{\gamma'}{\|W\|}, \qquad \text{subj. to} \qquad y_i f(W; x_i) \geq \gamma', \qquad i = 1, \dots, N. \tag{37}$$

Finally, using again one-homogeneity, without loss of generality, we can set $\gamma' = 1$ and obtain the equivalent problem

$$\max_W \frac{1}{\|W\|}, \qquad \text{subj. to} \qquad y_i f(W; x_i) \geq 1, \qquad i = 1, \dots, N. \tag{38}$$

The result is then clear noting that

$$\max_W \frac{1}{\|W\|} \Leftrightarrow \min_W \|W\| \Leftrightarrow \min_W \frac{\|W\|^2}{2}.$$

10 Gradient techniques for norm control

There are several ways to implement the minimization in the tangent space of $||V||^2 = 1$. In fact, a review of gradient-based algorithms with unit-norm constraints [44] lists

1. the *Lagrange multiplier method*
2. the *coefficient normalization method*
3. the *tangent gradient method*
4. the *true gradient method* using natural gradient.

For small values of the step size, the first three techniques are equivalent to each other and are also good approximations of the true gradient method [44]. The four techniques are closely related and have the same goal: performing gradient descent optimization with a unit norm constraint.

Remarks

- Stability issues for numerical implementations are discussed in [44].
- Interestingly, there is a close relationship between the Fisher-Rao norm and the natural gradient [7]. In particular, the natural gradient descent is the steepest descent direction induced by the Fisher-Rao geometry.
- Constraints in optimization such as $||v||_p = 1$ imposes a geometric structure to the parameter space. If $p = 2$ the weight vectors satisfying the unit norm constraint form a n-dimensional hyperspere of radius = 1. If $p = \infty$ they form an hypercube. If $p = 1$ they form a hyperpolyhedron.

10.1 Lagrange multiplier method

We start with one of the techniques, the Lagrange multiplier method, because it enforces the unit constraint in an especially transparent way. We assume separability and incorporate the constraint in the exponential loss by defining a new loss as

$$L = \sum_{n=1}^{N} e^{-\rho \tilde{f}((x_n)y_n} + \sum_{k=1}^{K} \lambda_k (||V_k||_p^p - 1) \tag{39}$$

where the Lagrange multipliers λ_k are chosen to satisfy $||V_k||_p = 1$ at convergence or when the algorithm is stopped.

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

$$\dot{\rho_k} = \sum_n \frac{\rho}{\rho_k} e^{-\rho(t)\tilde{f}(x_n)y_n} \tilde{f}(x_n), \tag{40}$$

and for each layer k

$$\dot{V_k} = \rho(t) \sum_n e^{-\rho(t)\tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k}(t) + \lambda_k(t) p V_k^{p-1}(t). \tag{41}$$

The sequences $\lambda_k(t)$ must satisfy $\lim_{t\to\infty} ||V_k||_p = 1 \quad \forall k$.
Remarks

1. In the case of $p = 2$, with the conditions $||V_k|| = 1$ at each t, $\lambda_k(t)$ must satisfy

$$||V_k(t) + \rho(t) \sum_n e^{-\rho(t)\tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k} - 2\lambda_k(t) V_k(t)|| = 1. \tag{42}$$

Thus defining $g(t) = \rho(t) \sum_n e^{-\rho(t)\tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k}(t)$ we obtain

$$||V_k(t) + g(t) + 2\lambda_k(t) V_k(t)|| = 1, \tag{43}$$

that is

$$||\alpha(t) V_k(t) + g(t)|| = 1, \tag{44}$$

with $\alpha(t) = 1 + 2\lambda_k(t)$. The solution for α is

$$\alpha(t) = \sqrt{1 - ||g(t)||^2 + (V_k^T g(t))^2} - V^T(t) g(t). \tag{45}$$

Thus λ goes to zero at infinity because $g(t)$ does and $\alpha \to 1$.

2. Since the first term in the right hand side of Equation (41) goes to zero with $t \to \infty$ and the Lagrange multipliers λ_k also go to zero, the normalized weight vectors converge at infinity to $\dot{V}_k = 0$. On the other hand, $\rho(t)$ grows to infinity. As shown in section 4.5, the norm square ρ_k^2 (when $p = 2$) of each layer grows at the same rate.

3. As in the case of the of the vanishing regularizer assumed in the original theorem of [33], the Lagrange multipliers λ_k here go to zero.

4. The Lagrange multiplier approach with $\lambda_k \to 0$ establishes a connection with the Halpern iteration and minimum norm solutions of degenerate minima.

10.2 Coefficient normalization method

If $u(k)$ is unconstrained the gradient maximization of $L(u)$ with respect to u can be performed using the algorithm

$$u(k+1) = u(k) + g(k) \tag{46}$$

where $g(k) = \mu(k) \nabla_u L$. Such an update, however, does not generally guarantee that $||u^T(k+1)|| = 1$. The coefficient normalization method employs a two step update $\hat{u}(k+1) = u(k) + g(k)$ and $u(k+1) = \frac{\hat{u}(k+1)}{||\hat{u}(k+1)||_p}$.

10.3 Tangent gradient method

Theorem 7 *([44]) Let $||u||_p$ denote a vector norm that is differentiable with respect to the elements of u and let $g(t)$ be any vector function with finite L_2 norm. Then, calling $v(t) = \frac{\partial ||u||_p}{\partial u}_{u=u(t)}$, the equation*

$$\dot{u} = h_g(t) = Sg(t) = (I - \frac{vv^T}{||v||_2^2})g(t) \tag{47}$$

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

In particular, a form for g is $g(t) = \mu(t)\nabla_u L$, the gradient update in a gradient descent algorithm. We call $Sg(t)$ the tangent gradient transformation of g. For more details see [44].

In the case of $p = 2$ we replace v in Equation 47 with u because $v(t) = \frac{\partial ||u||_2}{\partial u} = u$. This gives $S = (I - \frac{uu^T}{||u||_2^2})$ and $\dot{u} = Sg(t)$.

Remarks

- For $p = 2$ $v = \frac{\partial ||u||_p}{\partial u}_u$ is $v = \frac{u}{||u||_2}$
- For $p = 1$, $\frac{\partial ||u||_1}{\partial u_j} = \frac{u_j}{|u_j|}$.
- For $p = \infty$, $\frac{\partial ||u||_\infty}{\partial u_j} = \text{sign}(u_k)\delta_{k,j}$, if maximum is attained in coordinate k.

11 Standard dynamics, Weight Normalization and Batch Normalization

We now discuss the relation of some existing techniques for training deep networks with the gradient descent techniques under unit norm constraint of the previous section.

11.1 Standard unconstrained dynamics

The standard gradient dynamics is given by

$$\dot{W}_k^{i,j} = -\frac{\partial L}{\partial W_k^{i,j}} = \sum_{n=1}^{N} y_n \frac{\partial f(x_n; w)}{\partial W_k^{i,j}} e^{-y_n f(x_n; W)} \tag{48}$$

where W_k is the weight matrix of layer k. As we observed, this dynamics has global minima at infinity with zero loss, if the data are separable. The other stationary points have loss greater than zero.

Empirical observations suggest that unconstrained gradient dynamics on deep networks converges to solutions that generalize, especially when SGD is used instead of GD. In our experiments, normalization at each iteration, corresponding to the coefficient normalization method, improves generalization but not as much as Weight Normalization does.

11.2 Weight Normalization

For each layer (for simplicity of notation and consistency with the original weight normalization paper), weight normalization [45] defines v and g in terms of $w = g\frac{v}{|v|}$. The dynamics on g and v is induced by the gradient dynamics of w as follows:

$$\dot{g} = \frac{v^T}{||v||}\frac{\partial L}{\partial w} \tag{49}$$

and

$$\dot{v} = \frac{g}{||v||}S\frac{\partial L}{\partial w} \tag{50}$$

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

We claim that this is the same dynamics obtained from tangent gradient for $p = 2$. In fact, compute the flows in ρ, v from $w = \rho v$ as

$$\dot{\rho} = \frac{\partial w}{\partial \rho}\frac{\partial L}{\partial w} = v^T\frac{\partial L}{\partial w} \tag{51}$$

and $\dot{v} = \frac{\partial w}{\partial v||}\frac{\partial L}{\partial w} = \rho\frac{\partial L}{\partial w}$.

We then have to impose the unit norm constraint on v in the latter equation using the tangent gradient transform that gives

$$\dot{v} = S\rho\frac{\partial L}{\partial w}. \tag{52}$$

Clearly the dynamics of this algorithm is the same as standard weight normalization if $||v||_2 = 1$, because then Equations 49 and 50 become identical to Equations 51 and 52 with g corresponding to ρ. We now observe, multiplying Equation 50 by v^T, that $v^T\dot{v} = 0$ because $v^T S = 0$, implying that $||v||^2$ is constant in time. Thus if $||v|| = 1$ at initialization, it will not change (at least in the noiseless case). Thus *the dynamics of Equations 49 and 50 is the same dynamics as Equations 51 and 52.* It is also easy to see that the dynamics above is not equivalent to the standard dynamics on the w (see also Figure 1).

11.3 Batch Normalization

Batch normalization [46] for unit i in the network normalizes the input vector of activities to unit i – that is it normalizes $X^j = \sum_j W^{i,j}x_j$, where x_j are the activities of the previous layer. Then it sets the activity to be

$$Y^j = \gamma \cdot \hat{X}^j + \beta = \gamma\frac{X^j - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta,$$

where γ, β are learned subsequently in the optimization and

$$\mu_B = \frac{1}{N}\sum_{n=1}^{N} X_n \qquad \sigma_B^2 = \frac{1}{N}\sum_{n=1}^{N}(X_n - \mu_B)^2.$$

Note that both μ_B and σ_B^2 are vectors, so the division by $\sqrt{\sigma_B^2 + \epsilon}$ has to be understood as a point-wise Hadamard product $\odot(\sigma_B^2 + \epsilon)^{-1/2}$. The gradient is taken wrt the new activations defined by the transformation above.

Unlike Weight Normalization, the Batch Normalization equations do not include an explicit computation of the partial derivatives of L with respect to the new variables in terms of the standard gradient $\frac{\partial L}{\partial w}$. The reason is that Batch Normalization works on an augmented network: a BN module is added to the network and partial derivatives of L with respect to the new variables are directly computed on its output. Thus the BN algorithm uses only the derivative of L wrt the old variables as a function of the derivatives of L wrt new variables in order to update the parameters below the BN module by applying the chain rule. Thus we have to estimate what BN *implies* about the partial derivatives of L with the respect to the new variables as a function of the standard gradient $\frac{\partial L}{\partial w}$.

To see the nature of the dynamics implied by batch normalization we simplify the original Equations (in the Algorithm 1 box in [46]). Neglecting μ_B and β and γ, we consider the core transformation as $\hat{X} = \frac{X}{\sigma_B}$ which, assuming fixed inputs, becomes $\hat{X} = \frac{X}{|X|}$ which is mathematically identical with the transfomation of section 11.1 $v = \frac{w}{|w|}$. In a similar way the dynamics of $w = \frac{\partial L}{\partial w}$ induces the following dynamics on $\hat{X}$:

$$\dot{\hat{X}} = \frac{\partial \hat{X}}{\partial X}\dot{X} \tag{53}$$

where $\dot{x} = \nabla_x L$. We consider $X \in \mathbb{R}^{N\times D}$. In the $D = 1$ case, we get

$$\frac{\partial \hat{X}}{\partial X} = (\sigma_B^2 + \epsilon)^{-1/2}\left[-\frac{1}{N}\hat{X}\hat{X}^T + I\right].$$

In the general D-dimensional vector case, this generalizes to

$$\frac{\partial \hat{X}}{\partial X} = (\sigma_B^2 + \epsilon)^{-1/2}\left[-\frac{1}{N}\hat{X}^T \odot \hat{X} + I\right].$$

Notice that $I - \hat{X}\hat{X}^T = S$. Since $x = Wx_{input}$ this shows that batch normalization is closely related to *gradient descent algorithms with unit L_2 norm constraint of the tangent gradient type.* Because of the simplifications we made, there are other differences between BN and weight normalization, some of which are described in the remarks below.

Remarks

1. Batch normalization (see Supplementary Material), does not control directly the norms of $W_1, W_2, \cdots, W_K$ as WN does. Instead it controls the norms

$$||x||, ||\sigma(W_1 x||, ||\sigma(W_2\sigma(W_1 x))||, \cdots \tag{54}$$

   In this sense it implements a somewhat weaker version of the generalization bound.

2. In the multilayer case, BN controls separately the norms $||V_i||$ of the weights into unit i, instead of controlling the overall Frobenius norm of the matrix of weights as WN does. Of course control of the $||V_i||$ implies control of $||V||$ since $||V||^2 = \sum_i ||V_i||_i^2$.

11.4 Weight Normalization and Batch Normalization enforce an explicit unit 2-norm constraint

Consider the *tangent gradient transformation* to a gradient increment $g(t) = \mu(k)\nabla_u L$ defined as $h_g = Sg(t)$ with $S = I - \frac{uu^T}{||u||_2^2}$. Theorem 7 says that the dynamical system $\dot{u} = h_g$ with $||u(0)||_2 = 1$ describes the flow of a vector u that satisfies $||u(t)|_2 = 1$ for all $t \geq 0$. It is obvious then that

Observation 1 *The dynamical system describing weight normalization (Equations 49 and 50) are not changed by the tangent gradient transformation.*

The proof follows easily for WN by using the fact that $S^2 = S$. The same argument can be applied to BN. The property is consistent with the statement that they enforce an L_2 unit norm constraint.

Thus all these techniques implement margin maximization of $\tilde{f}$ under unit norm constraint of the weight matrices of $\tilde{f}$. Consider for instance the Lagrange multiplier method. Let us assume that starting at some time t, $\rho(t)$ is large enough that the following asymptotic expansion (as $\rho \to \infty$) is a good approximation: $\sum_n e^{-\rho(t)\tilde{f}(x_n)} \sim C \max_n e^{-\rho(t)\tilde{f}(x_n)}$, where C is the multiplicity of the x_n with minimal value of the margin of $\tilde{f}$. The data points with the corresponding minimum value of the margin $y_n\tilde{f}(x_n)$ are called support vectors (the x_i, y_i s.t $\arg\min_n y_n\tilde{f}(x_n)$). They are a subset of cardinality C of the N datapoints, all with the same margin η. In particular, the term $g(t) = \rho(t)\sum_n e^{-\rho(t)\tilde{f}(x_n)}\frac{\partial\tilde{f}(x_n)}{\partial V_k}$ becomes $g(t) \approx \rho(t)e^{-\rho(t)\eta}\sum_i^C \frac{\partial\tilde{f}(x_n)}{\partial V_k}$.

As we mentioned, in GD with unit norm constraint there will be convergence to $\dot{V}_k = 0$ for $t \to \infty$. There may be trajectory-dependent, multiple alternative selections of the support vectors (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 [41] *we expect multiple 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 algorithms ensure control of the norm and thus of the generalization bound even if they cannot ensure that the algorithm converges to the globally best minimum norm solution (this depends on initial conditions for instance).

12 Dynamics of ρ

We have seen that the dynamics of several algorithms to train deep networks – in particular of the standard gradient and of the weight normalization algorithmd – is very similar. In the first subsection we derive some useful properties of all these dynamics. In the second subsection we show that the standard gradient and the Lagrange multipler algorithm have the same stationary points. In the third subsection, we discuss in more detail the dynamics leading to the stationary points, focusing on the Lagrange multiplier algorithm.

12.1 ρ dynamics

First we show that for each layer $\frac{\partial \rho_k^2}{\partial t}$ is the same irrespectively of k. Then we consider the dynamics of ρ. In the 1-layer network case the ρ dynamics yields $\rho \approx \log t$ asymptotically. For deeper networks, we will show that the product of weights at each layer diverges faster than logarithmically, but each individual layer diverges slower than in the 1-layer case.

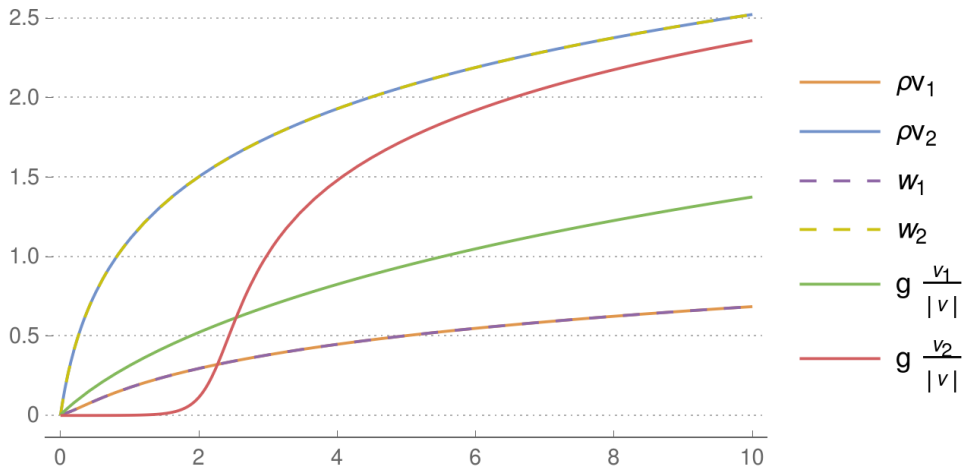

Figure 1: *Example comparison of dynamics* $\dot{W}_k = -\frac{\partial L}{\partial W_k}$ *(dashed lines) and their equivalence to the reparametrization of the unconstrained dynamics (orange and blue) as a function of time. Standard weight normalization leads to different trajectories of gradient descent (in green and red). The example is that of a linear network with only two parameters and two training examples, using an exponential loss.*

12.2 The rate of growth of ρ_k is the same for all layers

A property of the dynamics of W_k, shared with the dynamics of V_k under explicit unit norm constraint, is suggested by recent work [47]: *the rate of change of the squares of the Frobenius norms of the weights of different layers is the same during gradient descent.* This implies that if

the weight matrices are small at initialization, the gradient flow corresponding to gradient descent maintains approximatevely equal Frobenius norms across different layers, which is *a consequence of the norm constraint*. This property is expected in a minimum norm situation, which is itself equivalent to maximum margin under unit norm (see Appendix 9). The observation of [47] is easy to prove in our framework. Consider the gradient descent equations

$$\dot{W}_k^{i,j} = \sum_{n=1}^{N} y_n [\frac{\partial f(W; x_n)}{\partial W_k^{i,j}}] e^{-y_n f(x_n; W)}. \tag{55}$$

The above dynamics induces the following dynamics on $||W_k||$ using the relation $||\dot{W}_k|| = \frac{\partial ||W_k||}{\partial W_k} \frac{\partial W_k}{\partial t} = \frac{W_k}{||W_k||} \dot{W}_k$:

$$||\dot{W}_k|| = \frac{1}{||W_k||} \sum_{n=1}^{N} y_n f(W; x_n) e^{-y_n f(x_n; W)}. \tag{56}$$

because of lemma 3. It follows that

$$||\dot{W}_k||^2 = 2 \sum_{n=1}^{N} y_n f(W; x_n) e^{-y_n f(x_n; W)}, \tag{57}$$

which shows that the rate of growth of $||W_k||^2$ is independent of k. If we assume that $||W_1|| = ||W_2|| = \cdots = ||W_K|| = \rho_1(t)$ initially, they will remain equal while growing throughout training. The norms of the layers are balanced, thus avoiding the situation in which one layer may contribute to decreasing loss by improving $\tilde{f}$ but another may achieve the same result by simply increasing its norm.

12.3 Rate of growth of weights

In linear 1-layer networks the dynamics of gradient descent yield $\rho \sim \log t$ asymptotically. For the validity of the results in the previous section, we need to show that the weights of a deep network also diverge at infinity. In general, the K nonlinearly coupled equations are not easily solved analytically. For simplicity of analysis, let us consider the case of a single training example $N = 1$, as we expect the leading asymptotic behavior to be independent of N. In this regime we have

$$\rho_k \dot{\rho}_k = \tilde{f}(x) \left(\prod_{i=1}^{k} \rho_i \right) e^{-\prod_{i=1}^{K} \rho_i \tilde{f}(x)} \tag{58}$$

Keeping all the layers independent makes it difficult to disentangle for example the behavior of the product of weights $\prod_{i=1}^{K} \rho_i$, as even in the 2-layer case the best we can do is to change variables to $r^2 = \rho_1^2 + \rho_2^2$ and $\gamma = e^{\rho_1 \rho^2 \tilde{f}(x)}$, for which we still get the coupled system

$$\dot{\gamma} = \tilde{f}(x)^2 r^2, \qquad r\dot{r} = 2\frac{\log \gamma}{\gamma}, \tag{59}$$

from which reading off the asymptotic behavior is nontrivial.

We consider the case $\rho := \rho_1 = \rho_2 = \ldots = \rho_k$. This turns out to be true in general (see Equation 57). It gives us the single differential equation

$$\dot{\rho} = \tilde{f}(x) K \rho^{K-1} e^{-\rho_k \tilde{f}(x)}. \tag{60}$$

This implies that for the exponentiated product of weights we have

$$\left(e^{\rho_k \tilde{f}(x)}\right)^{\cdot} = \tilde{f}(x)^2 K^2 \rho^{2K-2}. \tag{61}$$

Changing the variable to $R = e^{\rho_k \tilde{f}(x)}$, we get finally

$$\dot{R} = \tilde{f}(x)^{\frac{2}{K}} K^2 \left(\log R\right)^{2-\frac{2}{K}}. \tag{62}$$

We can now readily 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. We show a comparison of the 1-layer and 2-layer behavior in the left graph in Figure 2. 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 linear network case, as can be seen in the right graph in Figure 2.

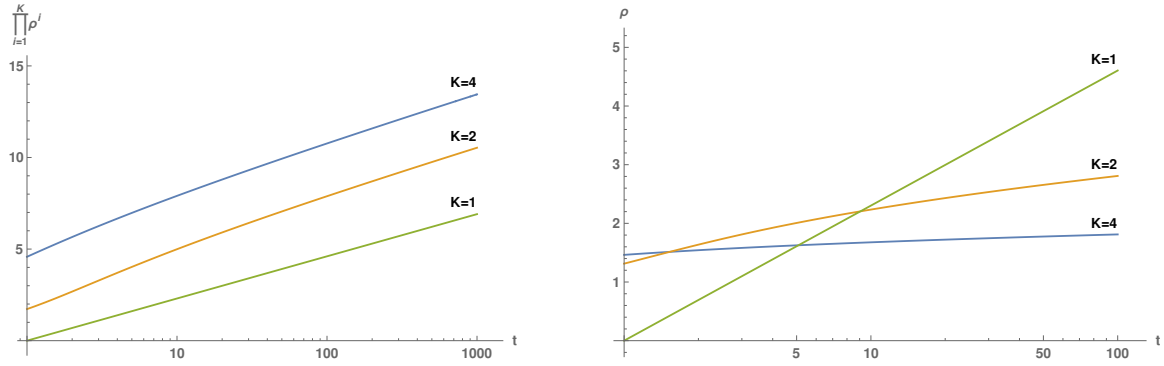

Figure 2: *The left graph shows how the product of weights $\prod_{i=1}^{K}$ scales as the number of layers grows when running gradient descent with an exponential loss. In the 1-layer case we have $\rho = \rho \sim \log t$, whereas for deeper networks the product of norms grows faster than logarithmically. As we increase the number of layers, the individual weights at each layer diverge slower than in the 1-layer case, as seen on the right graph.*

13 Critical points of the flow and their Jacobian

We discuss here in more details the stationary points of the gradient flow.

Lagrange multipliers It is interesting to start this section by characterizing the critical points of the flow induced by the Lagrange multiplier method described in section 10.1. In the case of the L_2 norm, the critical points $\dot{V}_k(t) = 0$ are given by the following set of equations – *as many as the number of weights*:

$$\dot{V}_k = \rho \sum_n e^{-\rho \tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k} - 2\lambda_k(\rho) V_k. \tag{63}$$

Notice that we do not need – though we can – to consider the $\dot{\rho}_k$ equations. A short-hand formulation reads as

$$V_k = \frac{g(t)}{2\lambda}. \tag{64}$$

where $g(t) = \rho(t) \sum_n e^{-\rho(t)\tilde{f}(x_n)} \frac{\partial \tilde{f}(x_n)}{\partial V_k}$ and λ goes to zero at the same rate as $g(t)$ since $1 = \frac{g^2}{4\lambda^2}$ [7].

As discussed elsewhere in the paper, in the linear one-layer case $\tilde{f}(x) = v^T x$, 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 $\sum \alpha_n \rho(t) x_n - \lambda v$, with λ satisfying $\lambda^2 = \sum_n e^{2\rho v^t x_n} \rho^2 x_n$. Thus $\lambda \to 0$ for $\rho \to \infty$ since λ is a function of ρ.

13.1 Jacobian of the Lagrange system

The Jacobian of $\dot{V}_k = -F(V_k, \rho_k) = -\nabla_{V_k} L$ (and Hessian of L) w.r.t. V_k tells us about the linearized dynamics around the asymptotic critical point of the gradient and thus about the sufficient conditions for local minima under the norm constraint. The Hessian H for the Lagrange multiplier case is

$$H^{i,j}_{k,k'} = \sum_n \left[-\left(\prod_{i=1}^K \rho_i^2 \right) \frac{\partial \tilde{f}(V; x_n)}{\partial V_k} \frac{\partial \tilde{f}(V; x_n)}{\partial V_{k'}}^T + \left(\prod_{i=1}^K \rho_i \right) \frac{\partial^2 \tilde{f}(V; x_n)}{\partial V_k \partial V_{k'}} \right] e^{-\prod_{i=1}^K \rho_i \tilde{f}(V; x_n)} - 2\lambda(\rho)\mathbf{I}. \tag{65}$$

The question is whether H is negative semidefinite or negative definite. The second case is needed to have *sufficient conditions* for local minima. We consider the quadratic form $\sum_{k,k'}^{K,K} V_k^T H V_{k'}$, where V_k correspond to critical points of the gradient, that it is satisfy Equations 64. It is easy to check that

[7] This suggests that weight matrices may be in relations of the type $W_K = (W_{K-1}...W_1 x)^T$ for linear K-layers nets. In other words, *gradient descent under unit norm is biased towards balancing the weights of different layers since this is the solution with minimum norm.* This is also consistent with the fact that the norms of the different layer weight matrices grow at the same rate. An instructive case is the linear two-layers autoencoder and the solution to the training of it in terms of transposed weights for the two layers.

- the first term of the quadratic form because of the structural lemma gives $-K^2\rho^2(\tilde{f}(x_n))^2 e^{-\rho\tilde{f}(x_n)} < 0, \forall\rho < \infty$;
- the second term because of Equation 5 is $= 0$ for $k = k'$; for $k \neq k'$ it is $\rho\tilde{f}(x_n)$. Thus the second term is $+K(K-1)\rho\tilde{f}(x_n)e^{-\rho\tilde{f}(x_n)} > 0$ for $K > 1$, if the data is separable.
- the third term gives $-2K\lambda, \forall\rho < \infty$. Notice that to satisfy $|V|^2 = 1$, we have to impose $\lambda = \frac{1}{2}\sum_n e^{-\rho\tilde{f}(x_n)}\tilde{f}(x_n)$, which would give the third term to be $-K\rho\tilde{f}(x_n)$

Thus for sufficiently late times when data have been separated ($\tilde{f}(x_n) > 0 \forall n$), H is negative definite for any finite ρ such that $\rho^2K^2(\tilde{f}(x_n))^2 > K(K-2)\rho\tilde{f}(x_n)$, where K is the number of layers. This only requires $\rho\tilde{f}(x_n) > 1$. It becomes negative semi-definite in the limit $\rho = \infty$. The sufficient conditions for a local minimum in V_k are satisfied for any finite, sufficiently large ρ.

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). Intuitively, for a separable data set there are minimum norm W_k solutions of the Lagrange multiplier constrained problem with negative definite Hessian for finite ρ: the solution is unique when the network is linear. In the nonlinear case overparametrized case it seems obvious that there may exist multiple solutions even under the norm constraint. Notice that definiteness of H requires $\rho > 0$ only for networks with more than 2 layers and effectively more than 3 (assuming our estimate for λ is correct). This property of stability of the critical point as a function of number of layers may be the reason for the important role of architectures such as ResNets.

The equation $\dot{V_k} = \rho\sum_n e^{-\rho\tilde{f}(x_n)}\frac{\partial\tilde{f}(x_n)}{\partial V_k} - 2\lambda_k(\rho)V_k$ becomes $\dot{V_k} = \rho\sum_n e^{-\rho\tilde{f}(x_n)}(\frac{\partial\tilde{f}(x_n)}{\partial V_k} - V_k$, since $\lambda_k = \frac{1}{2}\rho\sum_n e^{-\rho\tilde{f}(x_n)}\tilde{f}(x_n)$. The ρ dynamics for large ρ is much slower than the dynamics of V_k. For very large ρ, only one element in the sum matters and then $V_k = 0$ when $\frac{\partial\tilde{f}(x_n)}{\partial V_k} - 2\lambda_k(\rho)V_k = 0$. Before this, there are time dependent summands that determine a slowly changing V_k. Let us assume that $\rho \approx \log t$. Then

$$\dot{V_k} = \log t\sum_n(\frac{1}{t})^{\tilde{f}(x_n)}(\frac{\partial\tilde{f}(x_n)}{\partial V_k} - V_k \tag{66}$$

For large t the term in the sum with the smallest $\tilde{f}(x_n)$ will dominate. When ρ is smaller, the two smallest term will dominate in

$$\dot{V_k} = \log t[(\frac{1}{t})^{\tilde{f}(x_*)}(\frac{\partial\tilde{f}(x_*)}{\partial V_k} + (\frac{1}{t})^{\tilde{f}(x_{**})}(\frac{\partial\tilde{f}(x_{**})}{\partial V_k}] - V_k \tag{67}$$

and slowly change V_k.

Remarks

1. An examination of the gradient descent dynamics on exponential losses, discussed above, suggests experimenting with *a new family of algorithms* in which $\rho(t)$ is designed and

controlled independently of the gradient descent on the V_k: we need ρ to grow in order to drive the exponential loss towards the ideal classification loss but our real interest is in maximizing the margin of $\tilde{f}$. The ideal time course of $\rho(t)$ could be made adaptive to the data. It seems possible to test dynamics that may be quite different from $\rho \propto \log t$. Of course the role of ρ is complex: in the intial stages of GD it makes sense that the best solutions are solutions with $||V_k|| = 1$ and lowest exponential loss. Intuitively they correspond to solutions that separate and have minimum $||W_k||$.

2. In the linear one-layer case $\tilde{f}(x) = v^T x$, 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 $\sum \alpha_n(\rho(t)x_n - \lambda v$, with λ satisfying $\lambda^2 = \sum_n e^{2\rho v^t x_n}\rho^2 x_n$. Thus $\lambda \to 0$ for $\rho \to \infty$.

3. The Lagrange multiplier approach also establishes a connection with the Halpern iteration and minimum norm solutions of degenerate minima.

4. **Lemma 8** *If for $\rho > R$ the sum $\sum_{n=1}^{N} e^{-\rho \tilde{f}(x_n)} \propto e^{-\rho \tilde{f}(x_*)}$ then the margin increases $\frac{\partial \tilde{f}}{\partial t} \geq 0$ (even if ρ is kept fixed).*

   Proof $\tilde{f}(x*)$ increases monotonically or is constant because

$$\frac{\partial \tilde{f}(x*)}{\partial t} = \sum_k (\frac{\partial \tilde{f}(x*)}{\partial V_k})^T \dot{V_k} = \sum_k \frac{\rho}{\rho_k^2} e^{-\rho \tilde{f}(x^*)} \left(\left\| \frac{\partial \tilde{f}(x^*)}{\partial V_k} \right\|_F^2 - \tilde{f}(x^*)^2 \right). \tag{68}$$

   Equation 68 implies $\frac{\partial \tilde{f}}{\partial t} \geq 0$ because $||\tilde{f}(x^*)||_F = ||V_k^T \frac{\partial \tilde{f}(x^*)}{\partial V_k}||_F \leq ||\frac{\partial \tilde{f}(x^*)}{\partial V_k}||_F$, since the Frobenius norm is sub-multiplicative and V_k has unit norm.

13.2 Jacobian of the standard system

The critical points $\dot{V_k}(t) = 0$ can be rewritten as (dropping here the subscript k in V_k for simplicity of notation)

$$0 = \sum_n e^{-y_n f(x_n)} (I - VV^T) \frac{\partial \tilde{f}(x_n)}{\partial V}. \tag{69}$$

The Jacobian for the standard gradient descent case is

$$J_{k,k'}^{i,j} = \sum_n e^{-y_n \rho \tilde{f}(x_n)} (A_n + B_n + C_n) \tag{70}$$

with

$A_n = (-\rho \frac{\partial \tilde{f}(x_n)}{\partial V_k^i} (I - V_{k'}^j (V_{k'}^T)^\ell) \frac{\partial \tilde{f}(x_n)}{\partial V_{k'}^\ell})$

and

$B_n = \delta_{kk'}(-(\delta_{i,j} V_{k'}^\ell + \delta_{\ell,i} V_{k'}^i) \frac{\partial \tilde{f}(x_n)}{\partial V_{k'}^\ell})$

and

$C_n = (I - V_{k'}^j (V_{k'}^T)^\ell) \frac{\partial^2 \tilde{f}(x_n)}{\partial V_k^i \partial V_{k'}^\ell}$.

It is possible to check that for the form $\sum_{k,k'}^{K,K} V_k^i J_{k,k'}^{i,j} V_{k'}^j$ we find

- the first term disappears, because $(I - V_{k'}^j (V_{k'}^T)^\ell) V_{k'}^j = 0$.
- the second term gives $-2K\rho \sum_n e^{-\rho \tilde{f}(x_n)} \tilde{f}(x_n)$
- the third term disappears for the same reason as the first one.

Thus if gradient descent separates the data at T_0, for $t > T_0$ J is negative definite for any finite ρ. It becomes negative semi-definite in the limit $\rho = \infty$. The sufficient conditions for a local minimum in V_k are satisfied for any finite, sufficiently large ρ under the separability assumption.

Thus, in both cases the critical points are local, hyperbolic minima with slightly different stability conditions. While in the case of a linear network the minimum is unique, for nonlinear, multilayer networks there are multiple minima for finite ρ. The "better" ones, in the sense of achieving lower loss (which may correspond to larger margin), for the same ρ correspond, intuitively, to lower $||W_k||$ norm for the same number of iterations.

14 Linear networks: rates of convergence

We consider here the linear case. We rederive some of the results by [6] in our gradient flow framework. In the separable case of a linear network ($f(x) = \rho V^T x$) the dynamics is

$$\dot{\rho} = \frac{1}{\rho} \sum_{n=1}^{N} e^{-\rho V^T x_n} V^T x_n \tag{71}$$

and

$$\dot{V} = \frac{1}{\rho} \sum_{n=1}^{N} e^{-\rho V^T x_n} (x_n - V V^T x_n). \tag{72}$$

As discussed earlier there are K support vectors with the same smallest margin. For t increasing, since $\rho \approx \log t$,

$$\dot{V} \propto \frac{1}{\rho} \sum_{j=1}^{K} \alpha_j (I - V V^T) x_j. \tag{73}$$

with $\alpha_j = e^{-\rho V^T x_j}$. If gradient descent converges, the solution V satisfyies $V V^T x = x$, where $x = \sum_{j=1}^{K} \alpha_j^* x_j$. It is easy to see that $V = ||x||^2 x^\dagger$ – where $x^\dagger$ is the pseudoinverse of x – is a solution since the pseudoinverse of a non-zero vector z is $z^\dagger = \frac{z^T}{||z||^2}$. On the other hand, in the linear case the operator T in $V(t+1) = TV(t)$ associated with equation 73 is not expanding because V has unit norm. Thus [48] there is a fixed point $V = x$ which is *independent of initial conditions.*

To simplify the analysis we assume in the following that there is a single effective support vector (that is a set of points with the same margin). As one of the figures show, this is often not correct. The rates of convergence of the solutions $\rho(t)$ and $v(t)$, derived in different way in [6],

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 $e^{-\rho v^T x_n} \propto \frac{1}{t}$. 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 73 (we also assume for notation convenience a single data point x). We obtain

$$\dot{\epsilon} = \frac{1}{\rho} e^{-\rho v^T x}(x - (x+\epsilon)(x+\epsilon)^T x) = \frac{1}{\rho} e^{-\rho v^T x}(x - x - x\epsilon^T - \epsilon x^T) \tag{74}$$

which has the form $\dot{\epsilon} = -\frac{1}{t \log t}(2x\epsilon^T)$. A solution of the form $\epsilon \propto \frac{1}{\log t}$ satisfies the equation: $-\frac{1}{t \log^2 t} = -B\frac{1}{t \log^2 t}$. Thus the error indeed converges as $\epsilon \propto \frac{1}{\log t}$.

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

$$\dot{v} \propto e^{-\rho} \rho (I - vv^T)x. \tag{75}$$

This equation differs by a factor ρ^2 from equation 74. As a consequence equation 75 is of the form $\dot{\epsilon} = -\frac{\log t}{t}\epsilon$, with a general solution of the form $\epsilon \propto t^{-1/2 \log t}$. Numerical experiments confirm these rates for linear networks with one data point, see Figure 3, but suggest modifications in the case with $N \neq 1$. In summary, *GD with weight normalization converges faster to the same equilibrium than standard gradient descent: the rate for $\epsilon = v - x$ is $\frac{1}{t^{1/2 \log t}}$ vs $\frac{1}{\log t}$.*

Our simplified analysis implies that batch normalization has the same convergence rate as weight normalization (in the linear one layer case BN is identical to WN). As we discussed, it is however different wrt WN in the multilayer case: it has for instance separate normalization for each unit, that is for each row of the weight matrix at each layer.

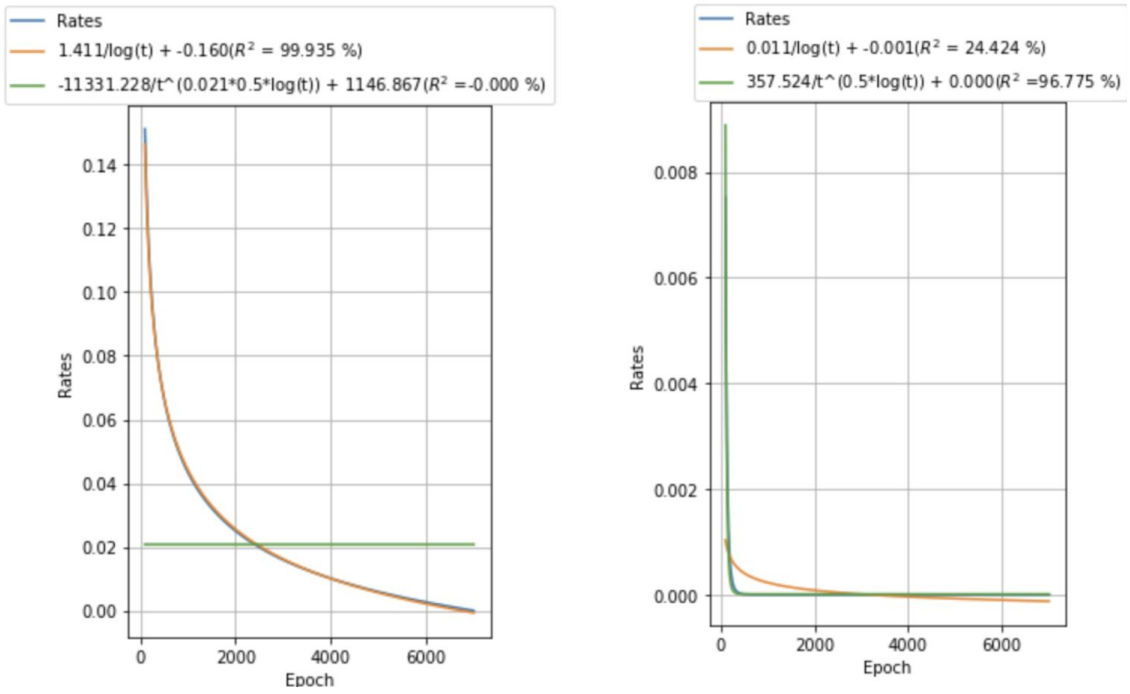

Figure 3: *Rates of convergence of $v(t)$ for a linear network trained on 1 training example on the IRIS dataset. Left: Standard gradient descent converges in direction at the rate $\mathcal{O}(\frac{1}{\log t})$. Right: Weight Normalization changes the convergence rate to that of $\mathcal{O}(t^{-1/2 \log t})$. We obtain more complex behavior in the presence of multiple support vectors with different margins.*