Implicit dynamic regularization in deep networks

Tomaso Poggio and Qianli Liao

Abstract

Square loss has been observed to perform well in classification tasks. However, a theoretical justification is lacking, unlike the cross-entropy case, for which an asymptotic analysis has been proposed. Here we discuss several observations on the dynamics of gradient flow under the square loss in ReLU networks. We show how convergence to a solution with the absolute minimum norm is expected when normalization techniques such as Batch Normalization (BN) or Weight Normalization (WN) are used together with zero-initial conditions on the layers weights. This is similar to the behavior of linear degenerate networks under gradient descent (GD), though the reason for zero-initial conditions is different. The main property of the minimizer that bounds its expected error is its norm: we prove that among all the interpolating solutions, the ones associated with smaller Frobenius norms of the weight matrices have better margin and better bounds on the expected classification error. The theory yields several predictions, including aspects of Papyan, Han and Donoho’s Neural Collapse and the bias induced by BN on the network weights towards orthogonal Stiefel matrices.

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

Tomaso Poggio and Qianli Liao

Abstract

Square loss has been observed to perform well in classification tasks. However, a theoretical justification is lacking, unlike the cross-entropy case for which an asymptotic analysis has been proposed (see [2] and [3] and references therein). Here we discuss several observations on the dynamics of gradient flow under the square loss in ReLU networks. We show how convergence to a solution with the absolute minimum norm is expected when normalization techniques such as Batch Normalization (BN) or Weight Normalization (WN) are used together with zero-initial conditions on the layers weights. This is similar to the behavior of linear degenerate networks under gradient descent (GD), though the reason for zero-initial conditions is different. The main property of the minimizer that bounds its expected error is its norm: we prove that among all the interpolating solutions, the ones associated with smaller Frobenius norms of the weight matrices have better margin and better bounds on the expected classification error. The theory yields several predictions, including aspects of Papyan, Han and Donoho’s Neural Collapse and the bias induced by BN on the network weights towards orthogonal Stiefel matrices.

1 Introduction

1.1 Why square loss

We start from the assumption that an explanation of the ability of deep ReLU networks to be predictive requires the identification of a hidden mechanism of complexity control at work during the training of deep networks.

In the case of exponential-type loss functions such a mechanism has been identified in the asymptotic margin maximization effect of minimizing exponential-type loss functions [7, 2, 8]. However, this mechanism

- cannot explain the good empirical results that can be obtained using the square loss [9];
- cannot explain the empirical evidence that convergence for cross-entropy loss minimization depends on initialization.

\[^1\]Complexity control should ensure CV stability of the solution (good stability implies good test error [6]).
This is the motivation for focusing here on the square loss.

In the paper we assume commonly used GD-based optimization algorithms such as BN or WN, since such mechanisms seem essential for reliably training deep networks\[^{10}\].

Our explanation for an implicit regularization in overparametrized deep nonlinear networks is similar to why GD converges to a minimal norm solution in the case of overparametrized linear systems\[^{2}\]. Crucially, our explanation requires normalization techniques such as BN or WN. We know that for overparametrized linear systems GD converges to the minimum norm solution if the weights are initialized close to zero values. We also know that in the presence of normalization of the weights – by BN or similar algorithms – the convergence is independent of initial conditions (see Appendix E.1). Thus it is natural to conjecture that a somewhat similar mechanism may ensure generalization in a nonlinear system which can be linearized around a (degenerate) minimum.

1.2 Regression and classification

In our analysis of the square loss, we need to explain when and why regression works well for classification, since the training optimizes square loss regression but we are interested in good performance in binary classification. A few preliminary remarks are useful. Unlike the case of linear networks we expect several global zero square loss minima corresponding to interpolating solutions (in general degenerate, see \[^{11}\] and reference therein). Although all interpolating solutions are optimal solutions of the regression problem, they will in general have different margins and thus different expected classification performance. In other words, zero square loss does not imply by itself large margin. Why and when we expect a bias for the solutions of the regression problem to have large margin? We will show that the bias for large margin interpolating solutions depends on initialization of gradient descent to be close to zero. As we will define later more formally, the deep network function can be written as \( g(x_n) = \rho f_n \) where \( g(x_n) \) is the output of the network for the training example \( x_n \), \( \rho \) is the product of the Frobenius norms of the weight matrices of the network and \( f_n = f(x_n) \) is the output of the normalized network for the input \( x_n \). Notice that if \( g \) is a zero loss solution of the regression problem, then \( g(x_n) = y_n, \forall n \). This is equivalent to \( \rho f_n = y_n \) where \( f_n \) is the margin for \( x_n \). Thus the norm \( \rho \) of a minimizer is inversely related to its average margin (see B). In fact, for an exact zero loss solution of the regression problem, the margin is the same for all training data \( x_n \) and it is equal to \( \frac{1}{\rho_{eq}} \). As we will see in the next section, under the assumption of separability, if \( \rho \) is small at initialization, it will grow monotonically under GD until a critical point of the gradient flow dynamics is reached. In other words, starting from small initialization, GD will explore critical points with \( \rho \) growing from zero. Thus interpolating solutions with small \( \rho_{eq} \) (corresponding to the best margin) may be found before large \( \rho_{eq} \) interpolating solutions which have worse margin (and are likely to be associated with the NTK regime). Thus for best classification, a small initialization is required, as in the linear case, though the reason is quite different.

This note is a preliminary communication with more details, proofs and experiments to follow

\[^{2}\]Minimizing the norm optimizes stability.
in a future update. We first provide the equations describing the dynamics of GD. Then we will describe results and conjectures. Details and experiments are in the Appendix.

2 The dynamics of GD in $\rho$ and $V_k$

2.1 Notation

We define\footnote{For more details about basic properties we use see [8].} a deep network with $L$ layers with the usual coordinate-wise scalar activation functions $\sigma(z) : \mathbf{R} \rightarrow \mathbf{R}$ as the set of functions $g(W; x) = (W_L \sigma(W_{L-1} \cdots \sigma(W_1x)))$, where the input is $x \in \mathbf{R}^d$, the weights are given by the matrices $W_k$, one per layer, with matching dimensions. We sometime use the symbol $W$ as a shorthand for the set of $W_k$ matrices $k = 1, \cdots, L$. There are no bias terms: the bias is instantiated in the input layer by one of the input dimensions being a constant. The activation nonlinearity is a ReLU, given by $\sigma(x) = x_+ = max(0, x)$. Furthermore,

- we define $g(x) = \rho f(x)$ with $\rho$ defined as the product of the Frobenius norms of the weight matrices of the $L$ layers of the network and $f$ as the corresponding network with normalized weight matrices $V_k$ (because the ReLU is homogeneous [8]);
- in the following we use the notation $f_n$ meaning $f(x_n)$, that is the output of the normalized network for the input $x_n$;
- we assume $||x|| = 1$ implying $||f(x)|| \leq 1$ at convergence;
- the following structural property of the gradient of deep ReLU networks is useful (Lemma 2.1 of [12]):

$$\sum_{i,j} \frac{\partial g(W; x)}{\partial W_k^{i,j}} W_k^{i,j} = g(W; x);$$

(1)

for $k = 1, \cdots, L$. Equation 1 can be rewritten as an inner product between $W_k$ as vectors:

$$(W_k, \frac{\partial g(W; x)}{\partial W_k}) = g(W; x)$$

(2)

where $W_k$ is here the vectorized representation of the weight matrices $W_k$ for each of the different layers. We use this vectorized notation in a few places, hoping it will not confuse the reader;

- we assume that $L \geq 2$. The main reason is to avoid the case of linear networks with a unique minimizer of the square loss;

- separability is defined as correct classification for all training data, that is $y_n f_n > 0$, $\forall n$. We call average separability when $\sum y_n f_n > 0$. 
2.2 Gradient descent

The natural approach to training deep networks for binary classification using the square loss is to use stochastic gradient descent to find the weights $W_k$ that minimize $L = \frac{1}{N} \sum_n \ell_n^2 = \frac{1}{N} \sum_n^N (g(x_n) - y_n)^2$, with $y = \pm 1$. In this note, we consider the gradient flow associated with gradient descent. Notice from the outset that, in general, classification with maximum margin is not minimum norm interpolation of the labels (see Appendix B.1).

2.3 Dynamics under normalization

Gradient descent on

$$L = \mathcal{L} + \lambda \sum_k V_k^2 = \sum_n (\rho f_n - y_n)^2 + \lambda \sum_k ||V_k||^2$$

with $||V_k||^2 = 1$ is equivalent to “Weight Normalization”, as proved in [8], for deep networks. This dynamics can be written as $\dot{\rho}_k = V_k^T \dot{W}_k$ and $\dot{V}_k = \rho S \dot{W}_k$ with $S = I - V_k V_k^T$. This shows that if $W_k = \rho_k V_k$ then $\dot{V}_k = \frac{1}{\rho_k} \dot{W}_k$ as mentioned in [10].

Our key assumption in this paper is that the dynamics above with Lagrange multipliers, captures the key normalization property of batch normalization, though not all of its details (see discussions in [8] and also [10]). Thus we assume that for network trained with BN, following the spirit of the analysis of [10], $\rho_k = 1$, \forall $k < L$ and $\rho_L = \rho$ where $L$ is the number of layers. It is important to observe here that batch normalization – unlike Weight Normalization – leads not only to normalization of the weight matrices but also to normalization of each row of the weight matrices [8] because it normalizes separately the activity of each unit $i$ and thus of the $W_{i,j}$ for each $i$. This implies that each row $i$ in $(V_k)_{i,j}$ is normalized and thus the whole matrix $V_k$ is normalized. The equations in the main text involving $V_k$ can be read in this way, that is restricted to each row. The normalization of each weight matrix yields, as shown in Appendix A.1 $\lambda = - \sum_n (\rho^2 f_n^2 - \rho y_n f_n)$ and thus

$$\dot{\rho} = -2\sum_n \rho (f_n)^2 - \sum_n f_n y_n = -2 \sum_n \ell_n f_n,$$

where $\ell_n = \rho f_n - y_n$ and

$$\dot{V}_k = 2 \sum n [(\rho f_n - y_n)\rho (-\frac{\partial f_n}{\partial V_k}) + 2V_k \rho f_n (\rho f_n - y_n)] = 2\rho \sum_n [(\rho f_n - y_n)(V_k f_n - \frac{\partial f_n}{\partial V_k})]$$

More details about the derivation of the Equations are in Appendix A (see also [8]).

2.3.1 Equilibrium values

The equilibrium value at $\dot{\rho}_k = 0$ is

Notice that $\dot{\rho} = 0$ is equivalent to $\sum \ell_n f_n = 0$. Thus the two conditions together $\dot{\rho} = 0$ and $\dot{V}_k = 0$ imply

$\sum \ell_n \frac{\partial f_n}{\partial V_k} = 0$. 

4
\[
\rho_{eq} = \frac{\sum_n y_n f_n}{\sum_n f_n^2}.
\]

Observe that \(\dot{\rho} > 0\) if \(\rho\) is smaller than its equilibrium value and if separability holds (in fact less than separability is sufficient: it is enough that average separability holds, that is \(\sum_n f_n y_n > 0\)).

If we assume that the loss (with the constraint \(||V_k|| = 1\)) is a continuous function of the \(V_k\), then there will be at least one minimum at any fixed \(\rho\), because the domain \(V_k\) is compact. This means that for each \(\rho\) there is at least a critical point of the gradient flow of \(V_k\), implying that for each critical \(\rho\) for which \(\dot{\rho} = 0\), there is a critical point of the dynamical system in \(\rho\) and \(V_k\).

Around \(\dot{V}_k = 0\) we have
\[
\sum_n (\rho f_n - y_n) \frac{\partial f_n}{\partial V_k} = \sum_n (\rho f_n - y_n)(V_k^{eq} f_n).
\]

In appendix \(C\) we describe the interesting dynamics associated with the unnormalized case (when \(\rho_k\) are all the same \(\forall k, k = 1, \ldots, L\)). If under BN, during GD, other layers in addition to the last have \(\rho_k\) different from 1 (which happens in practice during the iterations), the dynamics shows some of the interesting properties described in appendix \(C\), which favor small initializations to reach solutions with greater margin. The dynamics of Equation \(21\) is that the smaller \(\rho_{t=0}\) is, the longer it takes to \(\rho\) to grow (this phenomenon becomes stronger with a larger number of layers \(L\)). Thus \(\rho\) is constrained by the nonlinear dynamics to be very small for a transient phase \(T\) of GD iterations (\(T\) is longer with more layers and longer with smaller initialization).

Appendix \(D\) describes a few additional properties of the dynamics of the normalized weights.

The conclusions of this analysis can be summarized in

**Observation 1** Assuming average separability, and GD starting from \(\rho = \epsilon, \) where \(\epsilon > 0\) is small, \(\rho\) grows monotonically until a minimum is reached at which \(\rho_{eq} = \frac{\sum_n y_n f_n}{\sum_n f_n^2}\).

and

**Observation 2** Minimizers with small \(\rho_{eq}\) correspond to large average margin \(\sum y_n f_n\).

These two observations are supported by our numerical experiments. Figures 1,2,3,4 show the case of gradient descent with BN; figures 5,6,7 show the same conditions without BN.

## 3 Additional properties of minima and margin

In order to see the full implications of the two observations above let us mention two additional observations. We need to specify to what GD converges\(^7\) In addition, we have to prove formally

\(^5\)If \(V_k f_n = \frac{\partial f_n}{\partial V_k}\) then \(V_k = 0\) but this is not a critical point for the system \(\rho, V_k\) unless \(\dot{\rho} = 0\).

\(^6\)In the case of \(V_k\) the learning rate is different for WN vs. no-WN but the equilibrium values are the same.

\(^7\) The unnormalized global minima to which GD converges are typically degenerate (see references in \(\text{[13]}\)) with dimensionality \(W - N\), where \(W\) is the number of weights in the network.
that \(\rho\), which is inversely related to margin as we discussed, indeed controls the expected error.

The first point is answered by

**Observation 3** Suppose that GD or SGD converges to a \(\rho_{eq}\) and \(V_{eq}^{t}\) which correspond to a minimum with zero square loss. The minimum with the smallest \(\rho_{eq}\) (typically found during the GD dynamics when \(\rho\) increases from \(\rho = 0\)), corresponds to a solution with the minimum possible norm.

In general, there may be several critical points of the \(V_k\) for the same \(\rho_{eq}\) and they may be degenerate. All of them will correspond to the same norm and all will have the same margin for each of the training points\(^8\). The type of gradient dynamics leading to \(\rho_{eq}\) and \(V_{eq}^{t}\) remains an interesting open problem\(^9\) which requires an analysis not just of gradient flow but of SGD and of the associated Fokker-Planck equation. A related open problem is whether different solutions for the same minimum \(\rho_{eq}\) may have different expected error. Though they have the same norm and the same margin on each of the data point, they may have different ranks of the weight matrices\(^10\). It in unclear whether the rank has a role in generalization bounds.

The second question can be answered by using classical bounds that lead (see Appendix F) to the following theorem

**Observation 4** With probability \(1 - \delta\)

\[
L(f) \leq c_1\rho \mathbb{R}_N(\tilde{F}) + c_2\epsilon(N, \delta)
\]  

(8)

where \(c_1, c_2\) are constants that reflect the Lipschitz constant of the loss function (for the square loss this requires a bound on \(f(x)\)) and the architecture of the network. The Rademacher average \(\mathbb{R}_N(\tilde{F})\) depends on the normalized network architecture and \(N\). Thus for the same network and the same data, the upper bound for the expected error of the minimizer is smaller for smaller \(\rho\).

### 4 Generalization in Deep Networks

Consider a minimum of GD for which the square loss is zero and \(\dot{V}_k = 0\). Clearly critical points of \(\rho\) cannot exist if \(\rho\) is too small\(^11\). Since usually the maximum output of a network, depending on the number of layers, is \(< < 1\), the first critical point for increasing \(\rho\) will be when \(\rho\) becomes large enough to allow the following equation to have solutions

\[
\sum_n y_n f_n = \rho \sum_n f_n^2.
\]  

(9)

---

\(^8\)This argument is supported is supported by Theorem 4 and the other arguments in Appendix F.

\(^9\)The deterministic dynamics of \(V_k\) has in general a degenerate Hessian at the critical point.

\(^10\)In deep linear networks the GD dynamics seems to bias the solution towards small rank solutions, since large eigenvalues converge much faster the small ones [14].

\(^11\) \(\rho \geq 1\) for a critical point to exist because the critical point with smallest possible \(\rho\) is for \(\rho = 1, f_n y_n = 1\).
If gradient descent starts from very small $\rho$ and there is average separability, $\rho$ increases monotonically until such a minimum is found\[^{12}\]. If $\rho$ is large, then $\dot{\rho} < 0$ and $\rho$ will decrease until a minimum is found.

For large $\rho$, we expect many solutions under GD\[^{13}\]. The existence of many solutions is related to arguments showing the existence of NTK-based solutions: intuitively the last layer is enough in an extreme case – if the last layer before the linear classifier is overparametrized wrt training data – to provide solutions for any set of random weights in the previous layers (for large $\rho$ and small $f_i$). Furthermore the intermediate layer do not need to change much under GD in the iterations immediately after initialization. The emerging picture is a landscape in which there are no zero-loss minima for $\rho < \rho_{\text{min}}$ (which, in practice, means $\rho_{\text{min}} >> 1$). With increasing $\rho$ from $\rho = 0$ there will be zero square-loss degenerate (see \[^{11}\] ) minima with the minimizer representing an interpolating solution\[^{13}\].

The summary is as follows: assume that separation is reached and that $\rho$ is small. $\rho$ will eventually grow (and $f_n$ will become smaller), until $\sum \rho f_n^2 = \sum y_n f_n$. Consider instead the case of a large $\rho$ initialization: $\rho$ will decrease until $\sum \rho f_n^2 = \sum y_n f_n$. There are many $V_k$ such that $f_n$ satisfying the condition are now easy to find (small changes of the weight matrices before the linear classifier lead to different, interpolating solutions). Of course, though they are good minimizers of the square loss, they are not optimal for classification because their $f_n$ are smaller and so are the margins $y_n f_n$.

5 Predictions

- In a recent paper Papyan, Han and Donoho\[^{15}\] described four empirical properties of the terminal phase of training (TPT) deep networks, using the cross-entropy loss function. TPT begins at the epoch where training error first vanishes. During TPT, the training error stays effectively zero, while training loss is pushed toward zero. Direct empirical measurements expose an inductive bias they call neural collapse (NC), involving four interconnected phenomena. (NC1) Cross-example within-class variability of last-layer training activations collapses to zero, as the individual activations themselves collapse to their class means. (NC2) The class means collapse to the vertices of a simplex equiangular tight frame (ETF). (NC3) Up to rescaling, the last-layer classifiers collapse to the class means or in other words, to the simplex ETF (i.e., to a self-dual configuration). (NC4) For a given activation, the classifier’s decision collapses to simply choosing whichever class has the closest train class mean (i.e., the nearest class center [NCC] decision rule). We show in Appendix C that these properties of the Neural Collapse\[^{15}\] seem to be predicted by the theory of this paper.

\[^{12}\] This analysis is for gradient flow. A satisfactory theory requires an analysis of gradient descent along the lines of \[^{10}\].

\[^{13}\] It is interesting to recall \[^{11}\] that for SGD – unlike GD – the algorithm will stop only when $\ell_n = 0 \ \forall n$, which is the global minimum and corresponds to perfect interpolation. For the other critical points for which GD will stop, SGD will never stop but just fluctuate around the critical point.

\[^{14}\] Notice that the equilibrium value of $\rho$ is a measure of “sparsity”: small $\rho$ corresponds to $f_i$ being close to either 1 or zero ($\rho$ is in the order of $1/n$).
for each of the global (that is, zero square-loss) minima, irrespectively of the value of $\rho_{eq}$.

- At a global minimum\textsuperscript{15} the constraint Equations\textsuperscript{13} become (see Appendix\textsuperscript{E.2}) $\nabla V_k f(x_j) = V_k f(x_j)$ with $x_j$ in the training set, which is a powerful constraint on the weight matrices to which training converges. It requires a specific dependence of the matrix at each layer on matrices at the other layers. In particular, there are specific relations for each layer matrix $V_k$ of the type, explained in the Appendix,

$$V_k f = [V_L D_{L-1}(x) V_{L-1} \cdots V_{k+1} D_k(x)]^T D_{k-1}(x) V_{k-1} D_{k-2}(x) \cdots D_1(x) V_1 x,$$

(10) where the $D$ matrices are diagonal with components either 0 or 1, depending on whether the corresponding RELU unit is on or off.

As described in the Appendix a class of possible solutions to these constraint equations are projection matrices; another one are orthogonal matrices and more generally orthogonal Stiefel matrices on the sphere. These are sufficient but not necessary conditions to satisfy the constraint equations. The current analysis (in Appendix\textsuperscript{H}) of the constraint equation is quite limited since it holds strictly for deep linear networks: a full analysis is still missing. Interestingly, randomly initialized weight matrices (an extreme case of the NTK regime) are approximately orthogonal.

\textsuperscript{15}Or as a critical point of SGD.
Figure 2: Results with a convolution network and batch normalization: Dynamics of $\rho$ from experiments in Figure 1. Top row: small initialization (0.1). Bottom row: large initialization (5). The plot starts with $\rho(0) = 0$ despite an initialization of $\rho_k = 0$ because the the scaling factor of BN starts from 0. A dashed rectangle denotes the previous subplot’s domain and range in the new subplot.
Figure 3: Results with a convolution network and batch normalization: Margin of all training samples (see previous figures). If the solution were to correspond to exactly zero square loss, the margin distribution would be an horizontal line.

6 Summary

The main results of the paper analysis can be summarized in the following

**Lemma 1** If minimization by gradient descent with batch (or weight) normalization of the empirical square loss, converges to an interpolating solution with near-zero square loss, the following properties hold:

1. The global minima in the square loss with the smallest $\rho$ are the minimum norm solutions and have the best margin and the best bound on expected error;

2. conditions that favour convergence to minimum norm solutions are small initialization (small $\rho$) and normalization techniques such as BN;

3. the condition $\frac{\partial f(x_j)}{\partial V_k} = V_k f(x_j)$ which holds at the critical points of the dynamics that are global minima, is key in predicting several properties of the Neural Collapse [15];

4. the same condition must be satisfied by weight matrices at convergence: a class of possible solutions for the weights of the network can be identified with Stiefel matrices on the sphere.

6.1 Remarks

Some remarks are here, more in Appendix [1].
Figure 4: Results with a convolution network and batch normalization (see previous figures): Histogram of $|f_n|$ over time. Top figure: initial $\rho_k = 0.1$. Bottom figure: initial $\rho_k = 5$. 
• Suppose we control $\rho_k$ independently of $V_k$ and of equation 21: can we find $\rho(t)$ schedules leading more reliably to good solutions?

• The role of the Lagrange multiplier term $\lambda \sum_k ||V_k||^2$ in Equation 3 is different from a standard regularization term because $\lambda$, determined by the constraint $||V_k|| = 1$ can be positive or negative, depending on the sign of the error $\lambda = -\sum_n (\rho^2 f_n^2 - \rho y_n f_n)$. Thus the $\lambda$ term acts as a regularizer when the norm of $V_k$ is larger than 1 but has the opposite effect for $||V_k|| < 1$, thus constraining each $V_k$ to the unit sphere. For the exponential loss the situation is different and $\lambda \sum_k ||V_k||^2$ acts as a positive regularization parameter, albeit a vanishing one (for $t \to \infty$).

• If there exist several interpolating solutions with the same norm, they also have the same margin for each of the training data. It is an open question whether they may differ in terms of rank of the weight matrices or of the rank of the local Jacobian $\partial f_n / \partial V_k$ (at the minimum $W^*$). It is also an open question whether this may affect their expected error.

• Small initialization ensures that $\rho$ grows for small values thus exploring first large margin minima – assuming that average separability is reached early, during the first iterations of GD. Why does GD have difficulties in converging in the absence of BN, especially for very deep networks? At the moment, the best answer is that good tuning of the learning rate is important and BN was shown to provide a remarkable autotuning [10].

• The normalization Equation 3 is a precise model of WN. The same normalization of the weight matrices $V_k$ is also an effect of BN. However, BN is also normalizing each row of each $V_k$ matrix, at least approximately as we mentioned earlier.

• Are there any implications of the theory sketched here for mechanisms of learning in cortex? Somewhat intriguingly, some form of normalization, often described as a balance of excitation and inhibition, has long been thought to be a key function of intracortical circuits in cortical areas[16]. One of the first deep models of visual cortex models, HMAx, explored the biological plausibility of specific normalization circuits with spiking and non-spiking neurons. It is also interesting to note that the Oja rule describing synaptic plasticity in terms of changes to the synaptic weight is the Hebb rule plus a normalization term that corresponds to a Lagrange multiplier.

• The main problems left open by this paper are:
  - The analysis is so far restricted to gradient flow. It should be extended to gradient descent along the lines of [10].
  - The behavior of gradient descent around the global minima should be better characterized: Equation 5 contains two terms, one reflecting the normalization and the

---


\[\text{D matrices, effectively switching on and off some weights in the network, depending on } x_n \text{ and on whether BN is before or after the RELU nonlinearity (as pointed out by A. Banbuski).}\]
other the regression error. Zero regression error implies that normalization may fail at that point.

– In this context, an extension of the analysis to SGD may also be critical for providing a satisfactory analysis of convergence to these doubly degenerate minima.

Acknowledgments We are grateful to Shai Shalev-Schwartz, Arturo Desza, Akshay Rangamani, Santosh Vempala and especially to Eran Malach for very useful comments. Thanks also to Andy Banburski who explained to me what NC1 means. This material is based upon work supported by the Center for Minds, Brains 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


Figure 5: Binary classification on two classes from CIFAR-10, trained with MSE loss. The model is a very simple network with 4 layers of fully-connected Layers. The ReLU nonlinearity is used. The weight matrices of all layers are initialized with zero-mean normal distribution, scaled by a constant such that the Frobenius norm of each matrix is either 5, 15 or 30. We run SGD with batch size 128, constant learning rate 0.1 and momentum 0.9 for 1000 epochs. No Batch normalization. No weight decay. No data augmentation. Every input to the network is scaled such that it has Frobenius norm 1.

A  Gradient flow for $\rho$ and $V_k$

Gradient descent on $L = \frac{1}{N}(\sum_n g_n^2 - 2 \sum_n y_n g_n + N)$ (using $g_n = g(x_n)$) gives

$$\dot{W}_k = -\frac{2}{N} \sum_n (g_n - y_n) \frac{\partial g_n}{\partial W_k}$$

that is

$$\dot{W}_k = -\frac{\partial L}{\partial W_k} = -\frac{2}{N} \sum_n g_n \frac{\partial g_n}{\partial W_k} + \frac{2}{N} \sum_n y_n \frac{\partial g_n}{\partial W_k}$$

(12)

We now derive the dynamics of the norm and of the normalized weights. We define $g(x) = \rho f(x)$. $\rho$ is the product of the Frobenius norms of the weight matrices of the $L$ layers in the network. $f$ is the corresponding network with normalized weight matrices (because the ReLU is homogeneous [8]). In the following we use the notation $f_n$ meaning $f(x_n)$. We also assume $\|x\| = 1$ implying $\|f(x)\| \leq 1$ at convergence;

A.1 Dynamics under normalization

Gradient descent on $L = L + \lambda \sum_k V_k^2 = \sum_n (\rho f_n - y_n)^2 + \lambda \sum_k \|V_k\|^2$ with $\|V_k\|^2 = 1$ is completely equivalent to “Weight Normalization”[8] for deep networks.

Assuming that $\rho_k = 1$, $\forall k < L$ and $\rho_L = \rho$, gradient descent on $L$ wrt $\rho$ gives,

$$\dot{\rho} = -2 \sum_n \ell_n f_n.$$
Figure 6: Dynamics of $\rho$ from experiments in Figure 5. First row: small initialization (5). Second row: large initialization (15). Third row: extra large initialization (30). A dashed rectangle denotes the previous subplot’s domain and range in the new subplot. More details to be added.
Gradient descent on $L$ wrt $V_k$ gives
\[
\dot{V}_k = -\frac{\partial L}{\partial V_k} = -2 \sum_n (\rho f_n - y_n) \rho \frac{\partial f_n}{\partial V_k} - 2\lambda V_k. \tag{14}
\]

Because of the constraint imposed via Lagrange multipliers $||V_k||^2 = 1$, $V_k^T \dot{V}_k = 0$, which gives $\lambda = -\sum_n (\rho^2 f_n^2 - \rho y_n f_n)$.

In summary, gradient descent on $L$ wrt $\rho$ and $V_k$ gives
\[
\dot{\rho} = -2\left(\sum_n \rho(f_n)^2 - \sum_n f_n y_n\right) = -2\sum_n \ell_n f_n. \tag{15}
\]

where $\ell_n = \rho f_n - y_n$ and
\[
\dot{V}_k = 2 \sum_n [(\rho f_n - y_n) \rho (\frac{\partial f_n}{\partial V_k}) + 2V_k \rho f_n (\rho f_n - y_n)] = 2\rho \sum_n [(\rho f_n - y_n) (V_k f_n - \frac{\partial f_n}{\partial V_k})] \tag{16}
\]

Without BN $\frac{\partial g_k(W)}{\partial W_k} = \frac{\rho}{\rho_k} \frac{\partial f_n(V)}{\partial V_k}$; with BN this becomes $\frac{\partial g_k(W)}{\partial W_k} = \rho \frac{\partial f_n(V)}{\partial V_k}$, $\forall k < L$ and $\frac{\partial g_k(W)}{\partial V_k} = \frac{\partial f_n(V)}{\partial V_k}$.

This dynamics – where there is a “vanishing” Lagrange multiplier $\lambda$ – can also be written as $\dot{\rho}_k = V_k^T \dot{W}_k$ and $\dot{V}_k = \rho S \dot{W}_k$ with $S = I - V_k V_k^T$. This shows that if $W_k = \rho_k V_k$ then $\dot{V}_k = \frac{1}{\rho_k} \dot{W}_k$ as mentioned in [10].

Notice that $\dot{\rho} = 0$ if $y_n f_n = 1$ and $\rho_k = 1$; $\rho_{eq}$ in Equation 6 is a critical point for the dynamics of $\rho$ under GD. In the case of SGD the asymptotic value of $\rho$ for fixed $\sum f_i y_i$ may fluctuate randomly around the $\frac{\sum y_i f_n}{\sum_i f_n}$. Furthermore, the lowest possible value of $\rho_k$ at equilibrium ($\dot{\rho}_k = 0$) is $\rho_k = 1$ which can be achieved if $y_n f_n$ is either $1$ or $0$. Values $y_n f_n = 1$, $\rho = 1$ are stationary points of the dynamics of $V_k$ given by $\dot{V}_k = 0$: they are minimizers with zero square loss.

Figure 7: Margin of all training samples
B Maximum margin and minimum norm

Lemma 2 [8] The maximizer of the margin under the constraint \(||V_k|| = 1\) is the minimum norm solution under the constraint \(y_n f_n \geq 1, \\forall n\).

B.1 In general classification with maximum margin is not minimum norm interpolation of the labels

Minimum norm regression of binary labels is

\[
\min_{W_k} \frac{1}{2} ||W_k||^2, \quad \forall k \quad \text{subj. to} \quad y_i f(W_K, \cdots, W_1; x_i) = 1, \quad i = 1, \ldots, N. \tag{17}
\]

Minimum norm binary classification is

\[
\min_{W_k} \frac{1}{2} ||W_k||^2, \quad \forall k \quad \text{subj. to} \quad y_i f(W_K, \cdots, W_1; x_i) \geq 1, \quad i = 1, \ldots, N. \tag{18}
\]

Clearly classification involves minimizing over a larger class of functions than regression. The result will be in general different.

Observation 5 Minimum norm binary classification under the square loss with margin 1 (implying \(f_V(x_i) \geq 1 \forall i\)) is not (in general) interpolation of all the data.

Notice that hard margin SVM is a case in point: the SVs interpolate their data point, but other non-support vectors have margin greater than one.

C Unnormalized GD

Here we assume gradient flow without BN or WN, assuming (for simplicity of derivations) that at initialization all the layers have the same norm, that is \(\rho_k\) is the same for all \(k\) at initialization. Because of this assumption we can use the following

Lemma 3 \(\frac{\partial \rho_k^2}{\partial t}\) is independent of \(k\).

to claim that all \(\rho_k\) are the same at all times. Thus \(\rho = \rho_k^L\), where \(L\) is the number of layers.

Proof Consider \(\frac{\partial ||W_k||^2}{\partial t}\). The calculation follows the case for the exponential loss:

\[
\frac{\partial ||W_k||^2}{\partial t} = 2W_k \frac{\partial W_k}{\partial t} = \frac{4}{N} \sum_n g_n^2 + \frac{4}{N} \sum_n g_n \tag{19}
\]

because of the structural lemma. Thus the time evolution of \(\rho^2 = ||W_K||^2\) is independent of \(k\).

Then we obtain the dynamical system

\[
\dot{\rho}_k = - \frac{\partial L}{\partial \rho_k} = -2L \sum_n (\rho_k^L f_n - y_n) f_n \rho_k^{L-1} = -2 \rho_k^{L-1} \left[\sum_n \rho_k^L (f_n)^2 - \sum_n f_n y_n\right] \tag{20}
\]
which can be rewritten in terms of \( \rho = \rho_L^k \) using \( \dot{\rho} = \sum_k \frac{\partial \rho}{\partial y_k} \dot{\rho}_k \) as

\[
\dot{\rho} = 2L \rho^{2L-2} \left[ \sum_n f_n y_n - \sum_n \rho(f_n)^2 \right]
\]

which is an equation of the type known as differential logistic equation used for instance to model sigmoidal population growth. It has an interesting dynamics as shown in the simulations in the appendix (look at \( \rho \) for small initialization during the first 50 or so iterations). This is less the case for the BN dynamics of Equation 4.

The dynamics of Equation 21 is that the smaller \( \rho_{t=0} \) is, the longer it takes to \( \rho \) to grow (this phenomenon increase with larger number of layers \( L \)). Thus \( \rho \) is constrained by the nonlinear dynamics to be very small for a transient phase \( T \) of GD iterations (as we mentioned, \( T \) is longer with more layers and longer with smaller initialization) and then to grow slowly while \( f_n \) grows towards 1 (implying that \( \sum f_n^2 \) approaches \( \sum y_n f_n \)). Part of this dynamics was analyzed by Shalev-Schwartz [14] in a somewhat different context.

D Dynamics and equilibria for \( V_k \) and \( \rho \)

Equation 7 suggests the following form for \( V_{eq} \) close to the minimum (when \( \dot{V}_k = 0 \))

\[
V_{eq}^k = \sum_{n} \alpha_n \frac{\partial f_n}{\partial V_k}
\]

where \( \alpha_n = \frac{\rho f_n - y_n}{\sum_n (\rho f_n - y_n)f_n} \).

Notice that in general not all of the \( N \) terms in Equation 22 are different from zero\(^{18}\) or independent of each other (an upper bound is set by the rank of Jacobian \( \frac{\partial f_n}{\partial V_k} \)). As an example consider the degenerate linear case when \( f \) is a linear function.

D.1 Dynamics of minimization

If the initial conditions are \( \rho_{t=0} \approx 0 \), \( \rho(t) \) will eventually grow (most of the time, when it does not go to zero), but slowly for a longish time. Part of this behavior can be explained by the logistic equation in which the coefficients change with time, with \( y_n f_n \) decreasing slowly. As a consequence, the rate of increase of \( \rho \) decreases, though the asymptotic value of \( \rho = \frac{\sum y_n f_n}{\sum f_n^2} \) increases, until a critical point of the flow is reached.

Compare this with the case in which \( \rho \) is large at initialization: then \( \rho \) may decrease until a critical point is reached. As we already noticed, there are plenty of critical points at large \( \rho \) (stationary points for \( \rho \) and \( V_k \)) because under overparametrization almost every perturbations of

\[^{17}\text{Since assuming that } \sum \ell_n \neq 0 \text{ we obtain } V_{eq}^k = \sum_{n} \frac{\ell_n}{\sum_n \ell_n} \frac{\partial f_n}{\partial V_k}.
\]

\[^{18}\text{Equation 22 has the same general form as provided by the Fritz-John lemma for the solution of a constrained optimization problem.}\]
the weights before the linear classifier at the top yields a different interpolating solution. Notice that a minimum with a small $\rho$ may be seen as an implicit constrain\footnote{It is important to notice a key difference: the constraint induced by normalization algorithms is of the form $\rho = C$ rather than $\rho \leq C$.} corresponding to

$$
\min_{W_k}\ L = \sum (\rho f(x_i) - y_i)^2 \quad \text{s.t. } \rho \leq C
$$

(23)

which is Ivanov regularization, which is itself “equivalent” to Tikhonov regularization with appropriate $\lambda(\rho)$.

Since $\rho$ always increases from zero, after average separability is reached, up to an equilibrium value, small initialization allow the gradient flow to find minima with small $\rho$ (and large margin), whereas large initialization makes this much more difficult because of the many minima encountered for decreasing $\rho$ at large $\rho$. Once a small $\rho$ minimum is found, then the behavior with BN or WN is similar to iterative gradient descent in the case of linear networks (see for instance [17]) with convergence to the local minimum norm solution. In the nonlinear case, the same behavior could be expected once GD is close to a minimum because around a degenerate minimum (local or global), the loss should be locally equivalent to a Morse-Bott function (we are thinking about a positive definite Hessian in some directions and degenerate in the others). Unlike the case of linear networks, BN or WN are critical here to ensure convergence to a solution with minimum norm.

D.2 Weight decay and label noise

Adding a term $\lambda|W_k|^2$ to the Lagrangian corresponding to weight decay will change the dynamics of $\rho$ but not the dynamics of $V_k$. The Equation for $\rho$ becomes

$$
\dot{\rho}_k = -2\rho_k^{L-1}v[\sum_n \rho_k^L(f_n)^2 - \sum_n f_n y_n] - 2\lambda \rho
$$

(24)

and

$$
\dot{\rho} = -2[\sum_n \rho(f_n)^2 - \sum_n f_n y_n] - 2\lambda \rho
$$

(25)

with an equilibrium given by

$$
\rho_{eq} = \frac{\sum_n y_n f_n}{\lambda + \sum_n f_n^2}.
$$

(26)

Notice that label noise – suggested by Jason Lee and coworkers\cite{19} – may play a role somewhat similar to a regularization $\lambda$ by making $\sum_n f_n y_n$ smaller, thus also decreasing the equilibrium $\rho$ and biasing the final solution to have larger margin. In this view, both label noise and the regularization $\lambda$ should vanish with $t \to \infty$, keeping $\rho$ small during GD.

19It is important to notice a key difference: the constraint induced by normalization algorithms is of the form $\rho = C$ rather than $\rho \leq C$. 
Degenerate minima, rank and minimum norm

Consider a K-layers network $g(W; x_i)$ parametrized by its weights $W$.

We first consider a simple setup with simplifying assumptions. We discuss later how to weaken the requirements.

Consider the zero-loss, interpolating equations

$$ F(X, Y, W) = 0 $$

where $Z = X, Y$ is the training set, $W$ is the set of weights across layers and $F(X, Y, W)$ is a set of $N$ equations, one for each of the $N$ data points (columns of $X$ and $Y$), that is $F_j(X, Y, W) = f(W; x_j) - y_j, \ j = 1, \cdots, N$. We assume $X \in \mathbb{R}^{N_x}$, $Y \in \mathbb{R}^{N_y}$ and $W \in \mathbb{R}^{D}$.

Under assumptions of differentiability of $F$, Equation 27 defines a mapping $W(X, Y)$ in the neighborhood of the solution $X^*, Y^*, W^*$ such that $F(X, Y, W(X, Y)) = 0$ in that neighborhood.

We state here an implicit function theorem for selections (reformulated from [19]) that does not require the Jacobian to be invertible.

**Theorem 4** Consider the function $F : X \times W \rightarrow Y$, that is $F : \mathbb{R}^{N_x} \times \mathbb{R}^{D} \rightarrow \mathbb{R}^{N_y}$ with $N_x = N_y = N < D$ along with the associated solution mapping

$$ S : X \rightarrow W \in \mathbb{R}^{D}|F(W, Z) = 0. $$

Let $F(W, Z) = 0$ so that $W^* \in S(Z^*)$. Assume that $F$ is strictly differentiable at $(X^*, W^*)$ and suppose that the partial Jacobian $\nabla_W F(X^*, W^*)$ is of rank $M \leq N$. Then the mapping $S$ has a “local selection” $s$ around $X^*$ for $W^*$ which is a unique function of $X$ (which is in addition strictly differentiable with Jacobian $\nabla s(X) = A^T(AA^T)^{-1}\nabla_X g$, where $A = \nabla_W F(W^*, X^*)$).

Notice that Brower’s invariance of domain theorem says that the inverse $A^{-1}$ since $N_y < D$ fails to have a localization which is single-valued. However, though multivalued, $A^{-1}$ may contain a function with the property of single-valued localization for the case $N_y = D$. Such functions are called “selections” [19]. In our case, $\nabla_X W$ depends on the pseudoinverse of $\nabla_W g$.

The relevance of theorem 4 for this paper consists of the existence of a differentiable function $W(x_1, \cdots, x_n)$ around the minimum $W^*$ (and about its rank $M$, which depends on the rank of the Jacobian $\frac{\partial f}{\partial W}$). The selection results in the function $W(X)$ around the minimum $W^*(X^*)$ which is a linear function of the pseudoinverse of the Jacobian $\frac{\partial g}{\partial W}$.

Thus for each global minimum of the square loss, there is a minimizer with weights $W(X)$ (where $X$ is the training set) which depends on a selection of an inverse of the (degenerate) Jacobian $\frac{\partial f}{\partial W}$ at a global minimum of $L$.

Without WN or BN, gradient descent converges to the degenerate minimum of the loss but in general this does not guarantee convergence to a minimum norm solution because $W$ will contain in general non-zero components outside the range of $A^T$ – which correspond to non-zero initial conditions for convergence (in terms of the the Jacobian). However, normalization algorithms such as WN or BN ensure that GD will converge to the minimum norm solution in agreement with the (different) argument in the main text.
Notice that the rank of the product of two matrices is the minimum of the two ranks: this helps interpreting the implications of the statement in the theorem that \( \nabla s(X) = A^T(AA^T)^{-1}\nabla Xg \). Notice also that the rank of \( \nabla s(X) \) is an upper bound (usually not a good one) on the CVloo stability of the network since it corresponds to the number of independent training data to which \( W^* \) is sensitive. Thus the larger is the number of \( y_n g_n \) that are significantly larger than zero, the better is the stability. Finally notice that if GD converges then \( V_k = \frac{\partial f}{\partial V_k} = \nabla V_k f \) as shown in the next subsections. The arguments in the previous section suggest that convergence will be to the \( \rho V_k \) of minimum norm.

E.1 Convergence of Linear Networks with Normalization

Consider the separable case of a linear network \( (f(x) = \rho v^T x) \) the dynamics is with \( \ell_n = e^{-\rho y_n v^T x_n} \) for the exponential loss and \( \ell_n = (\rho v^T x_n - y_n) \) for the square loss

\[
\dot{\rho} = \frac{1}{\rho} \sum_{n=1}^{N} \ell_n y_n v^T x_n
\]

and

\[
\dot{v} \propto \sum_{n=1}^{N} \ell_n (x_n - vv^T x_n).
\]

Let us assume\(^{20}\) that \( \sum_{n=1}^{N} \ell_n y_n \neq 0 \). Thus

\[
\dot{v} \propto \sum_{j=1}^{N} \alpha_j (I - vv^T) x_j.
\]

If gradient descent converges to \( \dot{v} = 0 \), the solution \( v \) must satisfy \( vv^T x = x \), where \( x = \sum_{j=1}^{N} \alpha_j x_j \). Assume \( |x| = 1 \). Then \( v = x \). Since the operator \( T \) in \( v(t + 1) = T v(t) \) associated with equation \(^{31}\) is not expanding \(^{20}\) (because \( v \) has unit norm), there is a fixed point \( v = x \) which is independent of initial conditions.

E.2 Convergence of Networks with Normalization

Consider

\[
\dot{V}_k = \alpha_n (I - V_k V_k^T) \sum_{n=1}^{N} (y_n - \rho f_n) \frac{\partial f_n}{\partial V_k}.
\]

Let us define \( \alpha_n = (y_n - \rho f_n) \) and an “average” \( \frac{\partial \hat{f}_n}{\partial V_k} = \sum_j \alpha_j \frac{\partial f_j}{\partial V_k} \). Then \( (I - V_k V_k^T) \frac{\partial \hat{f}}{\partial V_k} = \dot{V}_k \).

If there is convergence, that is \( \dot{V}_k = 0 \), then

\(^{20}\)This assumption hints at a role of “noisy” gradient descent such as SGD in exploring all weights for effective normalization.
\[ V_k \dot{f} = \frac{\partial \hat{f}}{\partial V_k} \]  

(33)

with \( \dot{f} = V^T \frac{\partial f}{\partial V_k} = \sum_j \alpha_j \dot{f}_j \). The equation provides constraints on the weights \( V_k \) and the other layer weights at convergence. Notice at the critical points which are global minima we have equilibrium, that is \( \dot{V}_k = 0 \) for each of the data points that is

\[ (\rho f_n - y_n) \frac{\partial f_n}{\partial V_k} = (\rho f_n - y_n)(V_k^{eq} f_n), \quad \forall n = 1, \ldots, N. \]  

(34)

Thus arbitrarily close to equilibrium (but not at equilibrium) the following holds

\[ \frac{\partial f_n}{\partial V_k} = V_k^{eq} f_n, \quad \forall n = 1, \ldots, N. \]  

(35)

Parenthetically, SGD with minibatches of size 1 (the argument can be extended to other sizes < N) has stationary points given by

\[ 0 = (I - V_k V_k^T) \ell(x_n) \frac{\partial f(x_n)}{\partial V_k}, \quad \forall n. \]  

(36)

which is equivalent to

\[ \ell(x_n) \frac{\partial f(x_n)}{\partial V_k} = \ell(x_n) V_k f(x_n), \quad \forall n. \]  

(37)

Clearly the case of \( \ell_n = 0, \quad \forall n \) is a pathological point of the dynamics.

**F Margins, \( \rho \) and expected error**

Assuming that small initialization provides a bias towards solution with “large” margin, the next step is to use simple bounds \cite{21} to claim better expected error (and better stability) for those solutions.

A typical generalization bound that holds with probability at least \( (1 - \delta), \forall g \in \mathbb{G} \) has the form \cite{21}:

\[ |L(g) - \hat{L}(g)| \leq c_1 \mathbb{R}_N(\mathbb{G}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}} \]  

(38)

where \( L(g) = \mathbb{E}[\ell_{\text{gamma}}(g(x), y)] \) is the expected loss, \( \hat{L}(g) \) is the empirical loss, \( \mathbb{R}_N(\mathbb{G}) \) is the empirical Rademacher average of the class of functions \( \mathbb{G} \) measuring its complexity; \( c_1, c_2 \) are constants that reflect the Lipschitz constant of the loss function and the architecture of the network. The loss function here is the *ramp loss* \( \ell_{\text{gamma}}(g(x), y) \) defined as
We define $\ell_{\text{gamma}}(y, y') = 0$ as the standard 0–1 classification error and observe that $\ell_{\text{gamma}}(y, y') < \ell_{\text{gamma}=0}(y, y')$.

We now consider two solutions with zero empirical loss of the square loss regression problem obtained with the same ReLU deep network and corresponding to two different minima with two different $\rho$s. Let us call them $g^a(x) = \rho_a f^a(x)$ and $g^b(x) = \rho_b f^b(x)$. Using the notation of this paper, the functions $f_a$ and $f_b$ correspond to networks with normalized weight matrices at each layer.

Let us assume that $\rho_a < \rho_b$.

We now use the observation that, because of homogeneity of the networks, the empirical Rademacher complexity satisfies the property,

$$R_N(G) = \rho R_N(F),$$

where $G$ is the space of functions of our unnormalized networks and $F$ denotes the corresponding normalized networks. This observation allows us to use the bound Equation 38 and the fact that the empirical $\hat{L}_\gamma$ for both functions is the same to write

$$L_0(f^a) = L_0(F^a) \leq c_1 \rho_a R_N(F) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}$$

and

$$L_0(f^b) = L_0(F^b) \leq c_1 \rho_b R_N(F) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}.$$  

The bounds have the form

$$L_0(f^a) \leq A \rho_a + \epsilon$$

and

$$L_0(f^b) \leq A \rho_b + \epsilon$$

Thus the bound for the expected error $L_0(f^a)$ is better than the bound for $L_0(f^b)$.

Similar results can be obtained taking into account different $\hat{L}(f)$ for the normalized $f^a$ and $f^b$ under different $\gamma$ in Equation 38 that is

$$|L(f) - \hat{L}(f)| \leq c_1 R_N(F) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}.$$  

It is unclear whether these bounds are meaningful in practice. It is likely there exist better ways to bound the expected error.

---

Furthermore, the Rademacher complexity of the space of functions associated with normalized networks of the same architecture is the same (see [22]).
F.1 The Linear Case

Consider two different linear problem where the interpolating minimizers of the square loss are respectively $F^a(x) = \rho_a f^a(x) = \rho_a v^a_0 \cdot x$ and $F^b(x) = \rho_b f^b(x) = \rho_b v^b_0 \cdot x$ with $\|v_a\| = \|v_b\| = 1$. Let us assume that $\rho_a < \rho_b$.

We then know (from Lemma 15 in [23]) that the upper bound on the $CV_{\text{loo}}$ stability for $f_b$ is $\rho_b$ and for $f_a$ is $\rho_a$. Thus the bound on stability and expected error of $f_a$ vs. $f_b$ is $\rho_a/\rho_b$ times smaller.

F.2 Remarks

Notice that stability (and expected error) is upper bounded by the rank of the Jacobian at $W^*$ which is at most $N$. Equation 22 shows that $V_k$ depends on the $\frac{\partial f}{\partial V_k}$ with $M$ non-zero coefficients, where $M$ is related to the rank of the Jacobian of $f(X;V)$ wrt $V_k$. In particular, if $M = N_y$ that is the number of data points, the upper bound on $CV_{\text{loo}}$ stability of such a solution – obtained by just counting every choice of leave-one-out example – will be very poor.

Notice also is that $\rho$ is by itself a probably more direct measure of complexity of the hypothesis space; interestingly small $\rho$ roughly correlates with large “average” margin that is with $\sum y_n f_n \sum f_n^2$.

G Towards Predicting NC1 to NC4 (with A. Banburski)

We sketch here a proof of how our theoretical framework predicts the four properties NC1 to NC4 in the restricted case of binary classification under the square loss. In a network with $L$ layers the last layer activations for an input $x_{i,c}$ where $c$ is the class – here we consider just two classes $c = +1$ or $c = -1$ – are called $h_{i,c}$ by [15]. To conform to their notation we consider in this section a slightly different network from the one considered in the paper. Until now, we assumed that the network has one scalar output which is ideally $\pm 1$. Here we consider instead a network with two outputs, one representing the positive class and the other the negative one both trained to take the value $+1$ for the respective class. Thus the last layer weights are $V^c_{L} = +1$, $-V^c_{L}$. Convergence to a global minimum, that is $\dot{V}_k = 0$, $\dot{\rho} = 0$, implies $f^c(x_{i,c}) = 1$ independently of $i$, because $\rho f_i = \frac{1}{\rho_{eq}}$. The key observation is that $h^T_{i,c} = \frac{\partial f(x_{i,c})}{\partial V^c_L}$.

Corollary 5 Equations 4 and 5 imply NC1, NC2, NC3, NC4.

Proof

• NC1: since $V_k f_c(x_i) = \frac{\partial f_c(x_i)}{\partial V_k}$ at convergence (see Equation 35) $h^T(x_{i,c}) = V^c_{L} f^c(x_i)$ at convergence. Since $f^c(x_i)$ does not depend on $i$ at convergence, $V^c_{L} f^c(x_i)$ does not depend on $i$ implying that the standard deviation of $h_{i,c}$ wrt $i$ converges to zero with $V_k$ and $\dot{\rho}$ converging to zero.
• NC2: in the binary case \( h_{+1} \to V_L^T \) and \( h_{-1} \to -V_L^T \) (see above). Thus \( \mu_1 = V^T \) and \( \mu_2 = -V^T \). This is the special binary case of NC2.

• NC3: since \( h(x_i,c) = V_L^T f_c(x_i) \) and \( f_c \) is a number (that depends on \( c \) but not on \( i \)) at convergence \( h \) is proportional to \( V_L^T \).

• NC4: the result of [24], as shown by [15], implies together with NC1 and NC2, that NC3 and NC4 hold. Since our theory (in the previous sections), implies the result of [24], it also implies NC4.

We note that for the square loss these results apply to each global minimum, irrespectively of its \( \rho_{eq} \). For the exponential loss a similar argument may be used if we assume that the network at convergence (for infinite time!) only depends on the training data with the same maximum margin (usually all of the training data).

H Remarks on Constraints on the Weight Matrices

With \( f = \sum_j \alpha_j f_j \) and \( \alpha_j = y_j - \rho f(x_j) \), the condition we have derived in the main theory

\[
V_k f = \frac{\partial f}{\partial V_k}, \quad \forall k = 1, \ldots, L \tag{43}
\]

(assuming \( f \) is not zero) imposes a strong set of constraints on the weights of the network, since \( \frac{\partial f}{\partial V_k} \) depends on all weight matrices with the exception of \( V_k \). Suppose

\[
f(x) = (V_L \sigma(V_{L-1} \cdots \sigma(V_1 x))) \tag{44}
\]

where \( \sigma(x) = \sigma'(x)x \). The equation can be rewritten for each training example as

\[
f(x_j) = V_L D_{L-1}(x_j)V_{L-1} \cdots V_{k+1} D_k(x_j)V_k \cdots D_1(x_j)V_1 x_j \tag{45}
\]

where \( D_k(x_j) \) is a diagonal matrix with 0 and 1 entries depending on whether the corresponding RELU is active or not for the specific input \( x_j \), that is \( D_{k-1}(x_j) = \text{diag}[\sigma'(N_k(x_j))] \) with \( N_k(x_j) \) the input to layer \( k \).

Call \( V_L D_{L-1}(x)V_{L-1} \cdots V_{k+1} D_k(x) = a^T \) and \( D_{k-1}(x)V_{k-1} D_{k-2}(x) \cdots D_1(x)V_1 x = b \). Then \( f(x) = a^T V_k b \) and \( \frac{\partial f}{\partial V_k} = ab^T \). As sanity checks, \( f^T = b^T V_k^T a = f \); furthermore, the structural lemma Equation [1] gives

\[
\sum_{i,j} \frac{\partial f(V; x)}{\partial V_{i,j}} V_{i,j} = \sum_{i,j} a^i b^j V_{i,j} = f(x). \tag{46}
\]

Then Equation [43] becomes

\[
V_k f = [V_L D_{L-1}(x)V_{L-1} \cdots V_{k+1} D_k(x)]^T D_{k-1}(x)V_{k-1} D_{k-2}(x) \cdots D_1(x)V_1 x \tag{47}
\]
Let us now make some strong assumptions to get some intuition about the potential impact of the constraints on the weight matrices.

**Assumptions**

- Let us assume that all $V_k \forall k = 1, \cdots (L - 1) \in \mathbb{R}^{p,p}$ have the same dimensions, whereas $V_L \in \mathbb{R}^{1,p}$.

- We assume linear deep networks at training time, that is without RELUs. This is inspired by the observation that if solutions $V_k$ are found that satisfy Equation 47 then they will also satisfy the same Equations when the matrices $D_k$ are replaced by $I$, while the converse is not true.

  The intuition is that this big oversimplification may still be interesting, because the “training” equations above have to hold for all the $x_n$ in the training set. This implies that the $D$ matrices at each level are likely to eventually have 1 in each position of the diagonal across the whole of the training set. Of course, this will not hold completely and for all layers, especially if $p > N$ and especially for the layers at the top of the network, in which case the presence of the $D_k$ makes the constraint Equation 43 effectively weaker.

Under these three assumptions, let us consider two natural types of solutions that are consistent with Equation 43.

**H.1 The $V_k$ are projection matrices**

A class of solution which is consistent with the constraints represented by Equation 43 is

$$V_1 = V_2 = \cdots = V_{L-1}$$

and

$$V_L = D_{L-1}(x)V_{L-1} \cdots V_{k+1}D_k(x)V_k \cdots D_1(x)V_1x = \frac{\partial f}{\partial V_L}$$

In order for this to be true, the $V_k \ k < L$ matrices can be projection matrices ($P$ is a projection if $P^2 = P$; it is an orthogonal projection if $P = P^T$). Then, all the weight matrices are proportional to each other apart from the weight matrix of the last ($L$) layer which must be a vector proportional to the vector of activities of the units in layer $L - 1$.

If we assume that feedforward networks with $T$ layers converge to this type of solution, the interesting prediction is that recurrent networks (therefore with weight sharing across layers) under $T$ iterations should be identical to forward networks with $T$ layers (without weight sharing).

**H.2 The $V_k$ are orthogonal matrices (with normalization)**

Another possible set of solutions consists of matrices $V_k$ (assuming the weight matrices are all square matrices) each proportional to an orthogonal matrix. The constraint Equations 43 suggest the structure of a group since $V_k$ is proportional to the product of similar matrices.
A key property of orthogonal matrices is that $V_k^TV_k = V_k^{-1}$. Because of this property the constraint equations are always satisfied. For instance assume $f(x) = V_4V_3V_2V_1x$. Then it is easy to check that the constraint equations yield $V_3 \propto \frac{\partial f}{\partial V_3} = V_4^T(V_2V_1)^T$ and $V_2 \propto (V_4V_3)^T(V_1)^T$. Together they give $V_3 = V_3$.

We observe that the underlying reason for restricting this class of solutions to the orthogonal group is BN or WN, since they are equivalent to constrained optimization with Lagrange multipliers. As observed in [25] regularization of each weight matrix of a linear network reduces the symmetry group of the loss function from $GL_p(\mathbb{R})$ to the orthogonal group $O_p(\mathbb{R})$. Furthermore, it is interesting to notice, as they do, that

- Orthogonal matrices are the determinant $\pm 1$ matrices of minimal Frobenius norm (the squared determinant is the product of the squared singular values);
- Orthogonal matrices are the inverse matrices of minimum total squared Frobenius norm (sum of the squared singular values);
- A square matrix is orthogonal iff $A^{-1} = A^T$;
- Orthogonal matrices diagonalize any symmetric real-valued matrix $A = U\Lambda U^T$.

There is a large number of papers (for a random one see [26] and references therein) discussing the advantages of orthogonality for generalization in deep networks and probably as many papers proposing regularization-like algorithms in order to impose orthogonality in the weight matrices in a deep network. More generally, the discussion above should be extended from orthogonal matrices to non-square matrices in an orthogonal Stiefel manifold on the sphere. As far as we know, this appendix represents the first time that commonly used normalization algorithms, such as BN, are shown to bias weight matrices towards being orthogonal (or projection) vectors. It is natural to conjecture that additional properties of deep networks may be derived from the rich structure induced by this bias.

**H.3 Diagonal networks (with A. Banburski)**

Diagonal networks have been recently analyzed in a number of theoretical papers (see [14] and references therein) and are particularly interesting here for two reasons which we show below: diagonal initializations are preserved by gradient descent and the D-matrices corresponding to the RELUs stages commute.

First, notice that gradient updates are given by

$$w_{t+1} - wt = -\eta \sum_n \ell(f(w; x_n), y_n)\nabla w_t f(w; x_n)$$

and so an update to any off-diagonal weight will be non-zero only if $\nabla w_t f(w; x_n)$ for that weight is non-zero. We trivially find that

$$\frac{\partial f(x)}{\partial W_k} = W_L D_{L-1}(x)W_{L-1} \cdots W_{k+1}^{-1} D_k(x) D_{k-1}(x) W_{k-1}^{-1} D_{k-2}(x) \cdots D_1(x) W_1 x.$$  

(51)
This means that a off-diagonal layer-\( k \) entry \((i, j)\) update depends on a term in \( f \) that has an \( i \)-th column in layer \( k + 1 \) and \( j \)-th row in layer \( k - 1 \). But if at time \( t = 0 \) the off-diagonal terms \( i \neq j \) vanish, then the off-diagonal gradient update also vanishes for \( t = 1 \) and similarly for all subsequent times. Hence GD preserves diagonal initial conditions.

Assuming the same setting of square matrices as in the previous subsections, we can immediately see the usefulness of the diagonal assumption, from the simple fact that diagonal square matrices commute, so we can write
\[
f = V_L \ldots V_1 D_{L-1}(x) \ldots D_1(x)x = V_L \ldots V_1 \tilde{x} \quad \text{with} \quad \tilde{x} = D_{L-1}(x) \ldots D_1(x)x,
\]
i.e. we can push all the nonlinearities to the end and absorb them now into a single nonlinear transformation of the data \( x \). This then allows us to deal with the sum over all training examples: by defining \( \hat{x} = \sum_n \alpha_n \tilde{x}_n \) we have \( \hat{f} = V_L \ldots V_1 \hat{x} \).

We can now write the equation at the critical point here as
\[
\frac{\partial \hat{f}}{\partial V_k} = V_L^i \ldots V_{k+1}^i V_{k-1}^i \ldots x^i
\]
and
\[
V_k^i \hat{f} = V_k^i V_L \ldots V_{k+1} V_{k-1} \ldots \hat{x}
\]
where we now can label the diagonal layers with a single index \( i \). Let us look at a specific example of a 4-layer network now, \( \hat{f} = V_4 V_3 V_2 V_1 \hat{x} \). The equations are of the form
\[
V_k^i \hat{f} = V_k^i V_3 V_2 V_1 \hat{x}
\]
and similarly for other layers. Solving these, we get the very interesting constraints \( \hat{f}^2 = (V_3^i V_2^i \hat{x}_i)^2 = (V_3^i V_2^i \hat{x}_i)^2 = \ldots \), or that some \( V_k^i = 0 \). This gives us that at the critical point, \( V_k^i = \pm \hat{x}_i \) or \( V_k^i = 0 \).

Putting back the definitions for the different expressions, we get that at the critical points, the normalized weights are given by
\[
V_k^i = \pm \frac{\sum_n \alpha_n D_{L-1}(x_i^n) \ldots D_1(x_i^n) x_i^n}{\sum_n \alpha_n f(V; x_n)},
\]
with \( i = 1, \ldots, p \) being one of the input channels (and the diagonal path through the network). Notice that the dependence on the layer \( k \) is only in the \( \pm \) sign.

### I Additional Remarks

- Convergence to an interpolating solution with absolute minimum norm is expected when normalization techniques such as Batch Normalization (BN) or Weight Normalization (WN) are used together with near zero initialization of the weights. This is superficially similar to the behavior of linear degenerate networks under GD. Zero initial conditions induce GD
to explore first the small norm, large margin solutions and only later the large norm, small
margin ones. The bounds of Theorem 4 relate smaller $\rho$ solutions to better bounds on the
expected error.

- Equation 6 is a critical point for the dynamics of $\rho$ under GD but NOT under SGD. In
  the case of SGD the asymptotic value of $\rho$ for fixed $\sum f_i y_i$ can be expected to fluctuate
  randomly around $\sum \frac{y_n f_n}{n}$. Similar comments apply to the dynamics of $V_k$.

- For the square loss, convergence of the gradient flow to a minimum norm solution requires
  BN or WN, unlike the case of linear networks. For the exponential loss, BN is strictly not
  needed since minimization of the exponential loss maximizes the margin and minimizes the
  norm without BN. Thus under the exponential loss, we expect a margin maximization effect
  for $t \to \infty$ as shown in [3]. Deep nets under the square loss are more likely to overfit at
  long times than under exponential-type loss functions (unless momentum or regularization
  is used). As a consequence, early stopping is more likely to be effective for the square loss
  than for exponential-type loss functions. Empirically, it seems that square loss reaches
  solutions with good test error in multiclass CIFAR10 faster than cross-entropy. Continuing
  GD, however, sometime yields overfitting for the square loss (and worse test error) but
  not for cross-entropy. This is interesting because it validates the asymptotic complexity
  control we described in [27]. This suggests that in the experiments of [9], early stopping
  may play a role to obtain results with the square loss case that are as good or better than
  cross-entropy.

- Notice that in the case of BN $\frac{\partial g_n(W)}{\partial W_k} = \rho \frac{\partial f_n(V)}{\partial V_k}$ for $k \neq L$ and $\frac{\partial g_n(W)}{\partial W_L} = \frac{\partial f_n(V)}{\partial V_L}$ otherwise. This suggests that for all layers apart the final one, the Jacobians are $\frac{\partial f_n(V)}{\partial V_k} = \frac{1}{\rho} \frac{\partial g_n(W)}{\partial W_k}$.

- An assumption which is critical in our analysis is that normalization of the weights enforced
  by Lagrange multipliers captures the key effect of BN.