CENTER FOR
Brains
Minds+
Machines

CBMM Memo No. 117 **July 7, 2021**

Dynamics and Neural Collapse in Deep Classifiers trained with the Square Loss

Akshay Rangamani[1], **Mengjia Xu**[1,2], **Andrzej Banburski**[1], **Qianli Liao**[1], **Tomaso Poggio**[1]

[1]Center for Brains, Minds and Machines, MIT,
[2]Division of Applied Mathematics, Brown University

Abstract

Recent results suggest that square loss performs on par with cross-entropy loss in classification tasks for deep networks. While the theoretical understanding of training deep networks with the cross-entropy loss has been growing, the study of square loss for classification has been lacking. Here we study the dynamics of training under Gradient Descent techniques and show that we can expect convergence to minimum norm solutions when both Weight Decay (WD) and normalization techniques, like Batch Normalization (BN), are used. We perform numerical simulations that show approximate independence on initial conditions as suggested by our analysis, while in the absence of BN+WD we find that good solutions can be achieved for small initializations. We prove that quasi-interpolating solutions obtained by gradient descent in the presence of WD are expected to show the recently discovered behavior of Neural Collapse and describe other predictions of the theory.

This is an update to CBMM Memo 112.

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

Normalization and dynamics in Deep Classifiers trained with the Square Loss

Akshay Rangamani[1], Mengjia Xu[1,2], Andrzej Banburski[1], Qianli Liao[1], Tomaso Poggio[1]

[1]Center for Brains, Minds and Machines, MIT
[2]Division of Applied Mathematics, Brown University

September 14, 2021

Abstract

Recent results of [1] suggest that square loss performs on par with cross-entropy loss in classification tasks for deep networks. While the theoretical understanding of training deep networks with the cross-entropy loss has been growing ([2] and [3]), the study of square loss for classification has been lacking. Here we consider a toy model of the dynamics of gradient flow under the square loss in ReLU networks. We show that convergence to a solution with the absolute minimum norm is expected when normalization by a Lagrange multiplier (LN) is used together with Weight Decay (WD). In the absence of LN+WD, good solutions for classification may still be achieved because of the implicit bias towards small norm solutions in the GD dynamics introduced by close-to-zero initial conditions on the norms of the weights, similar to the case of overparametrized linear networks. The main property of the minimizers that bounds their expected error is the product ρ of the Frobenius norms of each layer weight matrices: we prove that among all the close-to-interpolating solutions, the ones associated with smaller ρ have better margin and better bounds on the expected classification error. We also prove that quasi-interpolating solutions obtained by gradient descent in the presence of WD are expected to show the recently discovered behavior of Neural Collapse [4] and describe other predictions. We discuss how to extend our framework to gradient descent and to multiclass classification. Normalization by Lagrange multiplier is similar but not identical to commonly use batch normalization and weight normalization. A comparison between them illuminates why normalization is important for convergence and explain differences between normalization techniques.

We perform numerical simulations to support our theoretical analysis.

1 Introduction

A widely held belief in the last few years has been that the cross-entropy loss is superior to the square loss when training deep networks for classification problems. As such, the attempts at understanding the theory of deep learning has been largely focused on exponential-type losses [2, 3], like the cross-entropy. For these losses, the predictive ability of deep networks depends on the implicit complexity control of Gradient Descent algorithms that leads to asymptotic maximization of the classification margin on the training set [5, 2, 6]. Recently however, [1] has demonstrated empirically that it is possible to achieve the same level of performance, if not better, using the square loss, paralleling older results for Support Vector Machines (SVMs) [7]. Can a theoretical analysis explain when and why regression should work well for classification? This question was the original motivation for previous versions of this paper[8]. In the meantime, several relevant papers have appeared and other related questions, in particular around a better understanding of normalization, have been asked. The present paper tries to cover these topics.

In deep learning, unlike the case of linear networks, we expect (in the absence of regularization) several global minima with zero square loss, thus corresponding to interpolating solutions (in general degenerate, see [9, 10] and reference therein). Although all the interpolating solutions are optimal solutions of the regression problem, they will in general correspond to different margins and to different expected classification performance. In other words, zero square loss does not imply by itself neither large margin nor good classification on a test set. When can we expect the solutions of the regression problem obtained by GD to have large margin?

We introduce a toy model of the training procedure that uses square loss, binary classification, gradient flow and Lagarnage multipliers for normalizing the weights. With this simple model we show that obtaining large margin interpolating solutions depends on the scale of initialization of the weights close to zero, in the absence of weight decay. We describe the dynamics of the norm of the deep network parameters, and show that large margin solutions can be obtained using small initializations. In the presence of weight decay, perfect interpolation cannot occur and is replaced by quasi-interpolation of the labels. In the special case of binary classification case in which $y_n = \pm 1$, quasi-interpolation is defined as $|f(x_n) - y_n| \leq \epsilon, \quad \forall n$, where ϵ is small. Our experiments and analysis of the dynamics show that, depending on the weight decay parameter, there may be independence from initial conditions, as has been observed in [1]. With both weight decay and normalization, we show that weight decay helps stabilize the solutions of the normalized weights, in addition to its role in the dynamics of the norm.

We then describe how to extend our toy model to include multiclass classification, gradient descent and batch normalization. A comparison of LN with BN and WN is particularly interesting for explaining the role of normalization in training deep networks and the differences between different normalization techniques.

Finally, we show that these quasi-interpolating solutions satisfy the recently discovered Neural Collapse (NC) phenomenon [4]. According to Neural Collapse, a dramatic simplification of deep network dynamics takes place – not only do all the margins become very similar to each other, but the last layer classifiers and the penultimate layer features form the geometrical structure of a simplex equiangular tight frame (ETF). Here we prove the emergence of Neural Collapse for the square loss and for exponential-type loss functions.

Our Contributions The main contributions of our paper are:

- We analyze the dynamics of deep network parameters, their norm, and the margins under gradient flow on the square loss, using a simple *Lagrange normalization* (*LN*) technique. We describe the evolution of the norm, and the role of Weight Decay and normalization in the training dynamics.
- We extend the analysis to GD.
- We extend the analysis to multiclass classification. **THIS NEEDS TO BE DONE**
- A comparison between LN and more standard techniques such as batch normalization (BN) and weight normalization (WN) illuminates the role of normalization in training deep networks and the differences between different normalization techniques. **THIS NEEDS TO BE DONE**
- We show that under certain assumptions, critical points of Gradient Descent with Weight Decay satisfy the conditions of Neural Collapse for both square and exponential loss functions. Our proof technique also allows us to find the relationship between the Simplex ETF and the margin of the solution.
- We support our conclusions with experiments.

Outline We structure the rest of the paper as follows. We start describing related work. In section 3 we describe the dynamics of gradient flow training under the square loss for a binary classification problem. We use an analysis of the dynamics to obtain insights about the role of Weight Decay and Batch/Weight Normalization. In section 8 we present and describe our experiments on CIFAR10 that highlight the insights we presented in section 3. In section 7 we present our derivation of Neural Collapse when training on the square loss, while the supplementary material extends the proof to the case of exponential loss functions. We conclude in section 9 with a discussion of our results and their implications for generalization.

2 Related Work

There has been much recent work on the analysis of deep networks and linear models trained using exponential-type losses for classification. The implicit bias of Gradient Descent towards margin maximizing solutions under exponential type losses was shown for linear models with separable data in [11] and for deep networks in [2, 3, 12, 13]. Recent interest in using the square loss for classification has been spurred by the experiments in [1], though the practice of using the square loss is much older [7]. Muthukumar et. al. [14] recently showed for linear models that interpolating solutions for the square loss are equivalent to the solutions to the hard margin SVM problem (see also [8]). Recent work also studied interpolating kernel machines [15, 16] which use the square loss for classification.

We are interested in how this translates to the case of deep networks.

In the recent past, there have been a number of papers analyzing deep networks trained with the square loss. These include [17, 18] that show how to recover the parameters of a neural network by training on data sampled from it. The square loss has also been used in analyzing convergence of training in the Neural Tangent Kernel (NTK) regime [19, 20, 21]. Detailed analyses of two-layer neural networks such as [22, 23, 24] typically use the square loss as an objective function. However these papers do not specifically consider the task of classification.

Neural Collapse (NC) [4] is a recently discovered empirical phenomenon that occurs when training deep classifiers using the cross-entropy loss. Since its discovery, there have been a few papers analytically proving its emergence. In [25] Mixon et. al. show NC in the regime of "unconstrained features". Other papers have shown the emergence of NC when using the cross entropy loss [26, 27, 28]. While preparing this paper, we became aware of recent results by Ergen and Pilanci [29] (see also [30]) who derived neural collapse for the square loss, through a convex dual formulation of deep networks. Our independent derivation is different and uses simple properties of the dynamics.

3 Dynamics of Gradient Flow on the Square Loss: a toy model

In this section we study training a deep RELU network by minimizing the square loss for a classification problem. We assume several simplifying conditions, therefore the term *toy model*. We will discuss how to relax them in section5. In our analysis we assume a normalization technique used during training as well as regularization (also called weight decay), since such mechanisms seem essential for reliably training deep networks using gradient descent[31], are commonly used and were used in most of the experiments by [1].

3.1 Assumptions

They consist of the following three assumptions:

- we consider the case of binary classification;
- we model the discrete Gradient Descent algorithm in terms of the continuous Gradient Flow. This is tantamount to assuming that the learning rate in GD is infinitesimally small;
- normalization of the weights is modeled adding a Lagrange multiplier term to a modified square loss function.

3.2 Definitions

We start by considering a binary classification problem given a training dataset $\mathcal{S} = \{(x_n, y_n)\}$ where $x_n \in \mathbb{R}^d$ are the inputs (normalized such that $||x_n|| \leq 1$) and $y_n = \pm 1$ are the labels. We use deep rectified homogenousn network with L layers to solve this problem. The basic form of the networks is $f_W : \mathbb{R}^d \to \mathbb{R}$, $f_W(x) = W_L \sigma(W_{L-1} \ldots \sigma(W_1 x) \ldots)$, where $x \in \mathbb{R}^d$ is the input to the network and $\sigma : \mathbb{R} \to \mathbb{R}$, $\sigma(x) = \max(0, x)$ is the rectified linear unit (ReLU) activation function that is applied coordinate-wise at each layer. The last layer of the network is linear. We define $f_n = f_V(x_n)$ (the network of Figure 1B, evaluated on the training sample x_n).

3.2.1 Network parametrization

Due to the positive homogeneity of ReLU, one can reparametrize $f_W(x)$ by considering normalized [1] weight matrices $V_k = \frac{W_k}{||W_k||}$ and define $\rho_k = ||W_k||$ obtaining $f_W(x) = \rho_L V_L \sigma\left(\rho_{L-1} \ldots \sigma\left(\rho_1 V_1 x\right) \ldots\right)$. Because of homogeneity of the RELU it is possible to pull out the product of the layer norms as $\rho = \prod_k \rho_k$ and write $f_W(x) = \rho f_V(x) = \rho V_L \sigma\left(V_{L-1} \ldots \sigma\left(V_1 x\right) \ldots\right)$. Notice that the two networks – $f_W(x)$ and $\rho f_V(x)$ – are equivalent reparametrizations (if $\rho = \prod_k \rho_k$) but optimization is in general affected by reparametrization.

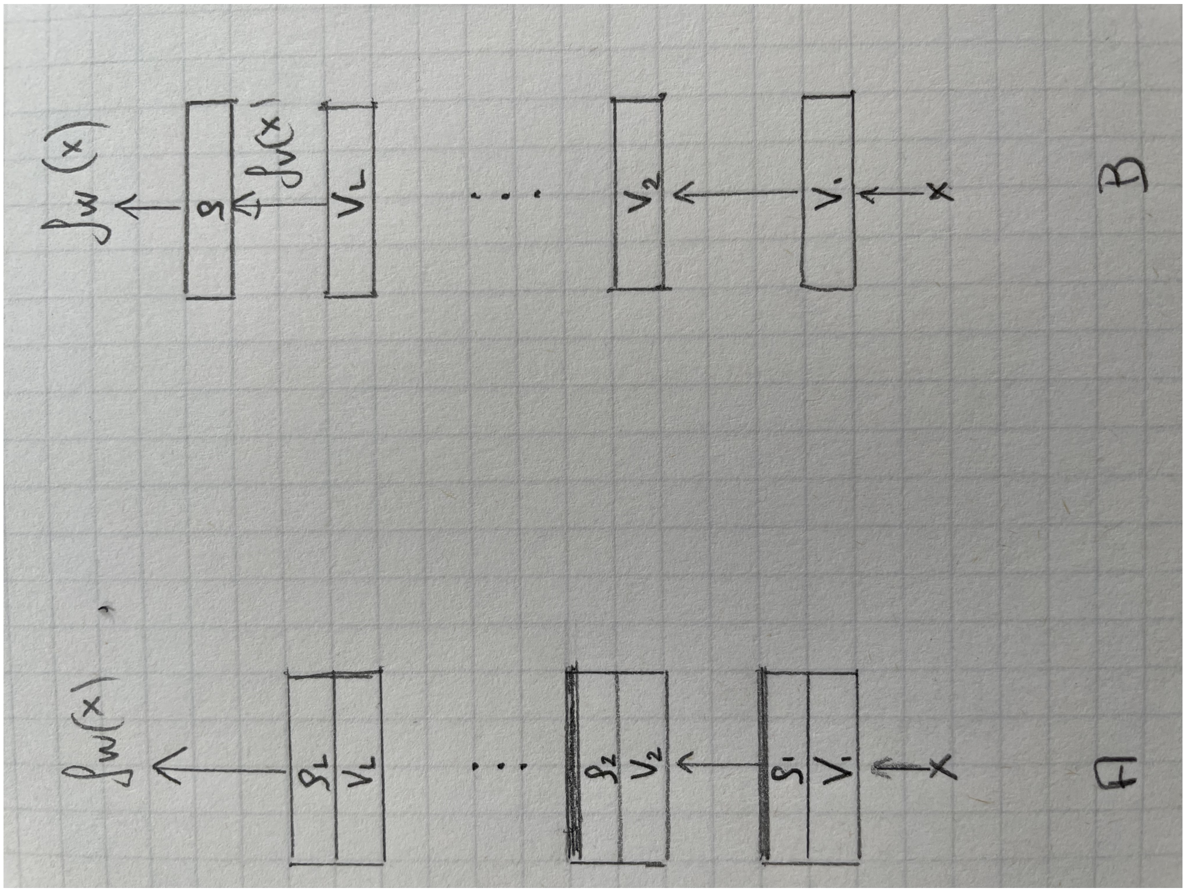

Figure 1: *Two parametrizations of a deep network. The thick line in each box represents the RELU nonlinearity. Each box corresponds to a layer. We use network A) in the case in which the weight matrices $W_k = \rho_k V_k$ with $||V_k|| = 1$ are not normalized by an algorithm like LM or BN. We use network B) when the weight matrices V_k at each layer are actively normalized and only the last layer ($\rho_L V_L$) is not under normalization.*

3.2.2 Toy model

We assume that there is a normalization stage. In practice this is usually performed using either batch normalization (BN) or weight normalization (WN). BN consists of standardizing the output of the units in each layer to have zero mean and unit variance. WN normalizes the weight matrices in a way which is more similar to the tangent gradient method (section 10 in [**?**]). In our toy model we make here a significant simplifying assumption: we model normalization using a Lagrange multiplier term added to the loss. We will later discuss how this is different from the usual normalization algorithms.

In the presence of normalization, we assume as shown in Figure 1 B that all layers but the last one are normalized (at convergence) via a Lagrange multiplier added to the loss. Thus the weight matrices

[1] We choose the Frobenius norm here to simplify our calculations. While a different choice of norm (spectral, $\ell_1, \ell_{2,1}$) may help prove tighter generalization bounds, they are functionally equivalent for the purpose of analyzing dynamics and the margin (as noted in section 3.4 of [32]).

$V_k, \quad k = 1, \cdots, L$ are constrained by the Lagrange multiplier term during gradient descent to be close to and eventually converge to unit norm matrices; notice that normalizing V_L with a ρ multiplier after it is equivalent to let $W_L = \rho V_L$ be unnormalized. Thus f is a network that under normalization is supposed to converge to a network with $L-1$ normalized layers.

Constrained minimization of $\mathcal{L} = \sum_n (\rho f_n - y_n)^2 + \lambda \rho^2$ under the constraint $||V_k||^2 = 1$ leads to minimizing $\mathcal{L} = \sum_n (\rho f_n - y_n)^2 + \nu \sum_{k=1}^{L} (||V_k||^2 - 1) + \lambda \rho^2$. Here and in the following we rewrite $(\rho f_n - y_n)^2 = (1 - \rho \overline{f}_n)^2$ with $\overline{f}_n = f_n y_n$

Thus the loss functional is

$$\mathcal{L} = \sum_n (1 - \rho \overline{f}_n)^2 + \nu \sum_{k=1}^{L} (||V_k||^2 - 1) + \lambda \rho^2 \tag{1}$$

3.2.3 Dynamics

Gradient flow from Equation 1 gives

$$\dot{\rho} = -\frac{\partial \mathcal{L}}{\partial \rho} = 2 \sum_n (1 - \rho \overline{f}_n) \overline{f}_n - 2\lambda \rho \tag{2}$$

and

$$\dot{V_L} = -\frac{\partial L}{\partial V_L} = -2 \sum_n (1 - \rho \overline{f}_n) \rho \frac{\partial \overline{f}_n}{\partial V_L} \tag{3}$$

and for $k < L$

$$\dot{V_k} = -\frac{\partial L}{\partial V_k} = -2 \sum_n (1 - \rho \overline{f}_n) \rho \frac{\partial \overline{f}_n}{\partial V_k} - 2\nu V_k \tag{4}$$

In the latter equation we can use the unit norm constraint on the $||V_k||$ to determine the Lagrange multiplier ν. Using the structural lemma (Appendix B), the constraint $||V_k||^2 = 1$ implies that $\frac{\partial ||V_k||}{\partial t} = V_k^T \dot{V_k} = 0$, which gives $\nu = -\sum_n (\rho^2 f_n^2 - \rho y_n f_n)$.

Thus the gradient flow is the following dynamical system

$$\dot{\rho} = 2[\sum_n \overline{f}_n - \sum_n \rho (\overline{f}_n)^2] - 2\lambda \rho \tag{5}$$

$$\dot{V_k} = 2\rho \sum_n [(1 - \rho \overline{f}_n)(V_k \overline{f}_n - \frac{\partial \overline{f}_n}{\partial V_k})] \tag{6}$$

$$\dot{V_L} = -\frac{\partial L}{\partial V_L} = -2 \sum_n (1 - \rho \overline{f}_n) \rho \frac{\partial \overline{f}_n}{\partial V_L} \tag{7}$$

3.2.4 Critical points

The critical points of the ρ dynamics with Weight Decay occur when $\frac{\partial \rho}{\partial t} = 0$, which happens when $\rho = \rho_{\text{eq}}$

$$\rho_{\text{eq}} = \frac{\sum_n \overline{f}_n}{\lambda + \sum_n \overline{f}_n^2} \tag{8}$$

The critical points of the V_k dynamics – that is when $\dot{V_k} = 0$ – satisfy

$$\sum_n [(1 - \rho \overline{f}_n)(V_k \overline{f}_n - \frac{\partial \overline{f}_n}{\partial V_k})] = 0. \tag{9}$$

Notice that for $\lambda \to 0$, the critical points $\dot{\rho_{\text{eq}}} = 0$ imply exact interpolation. Furthermore the critical points of the dynamical system in $\dot{\rho}$ and $\dot{V_k}$ are the critical points of $\dot{\rho}$ because of the following

Lemma 1 *For every ρ there are critical points of $\dot{V_k}$.*

Notice however that the critical points of V_k for a given ρ are not necessarily consistent[2] with the values of f_n required by the critical points of $\dot{\rho}$ (since the V_k determine the f_n).

Let us here introduce a somewhat unusual definition. *We define critical points of SGD as* $\dot{z} = g(x_n) = 0 \forall n = 1, \cdots, N$ instead of the standard definition of the critical points of GD defined as $\dot{z} = \sum_{n=1}^{N} g(x_n) = 0$. This definition makes sense for any minibatch size $< N$ because of Lemma 9 in the Appendix.

If we look for the critical points of SGD with minibatch 1 (see Appendix 40) then

$$V_k^0 \overline{f}_n (1 - \rho \overline{f}_n) = \frac{\partial \overline{f}_n}{\partial V_k} (1 - \rho \overline{f}_n) \quad \forall n. \tag{10}$$

3.2.5 Separability, average separability, margin and average margin

Separability is defined as the condition $y_n f_n = \overline{f}_n > 0, \forall n$ (all training samples are classified correctly). If $\sum_n \overline{f}_n > 0$, we say that *average separability* is satisfied.

Notice that if f_W is a zero loss solution of the regression problem, then $f_W(x_n) = y_n, \quad \forall n$. This is equivalent to

$$\rho f_n = y_n \tag{11}$$

where we call f_n *the margin for* x_n. Thus by multiplying both sides of Equation 11 by y_n and summing both sides over n gives $\rho \sum_n \overline{f}_n = N$. Thus the norm ρ of a minimizer is inversely related to its average margin in the limit of $\lambda = 0$, that is $\frac{1}{\rho} = \frac{1}{N} \sum_n \overline{f}_n$, where $\frac{1}{N} \sum_n \overline{f}_n$ is the *average margin*. Note that $\overline{f}_n \leq 1$ since $||x|| \leq 1$, and the weight matrices are normalized.

3.2.6 Norm dynamics

Lemma 2 *Suppose that at a small t after initialization $\rho(t) < \rho_0$ and $\dot{\rho}(t) > 0$. Then $\dot{\rho}(t) > 0, \quad \forall t$ until $\rho(t) = \rho_{eq}$, where ρ_{eq} is the critical point with min ρ.*

If $\rho(t) = 0$ and $f(x) = 0$ the dynamics is in a critical unstable point. A small perturbation will either result in $\dot{\rho} < 0$ with ρ going back to zero or in $\dot{\rho} > 0$ with ρ growing. The observation is that until $\rho(t) = \rho_{eq}$, $r\dot{h}o > 0$, that is ρ grows; otherwise if $r\dot{h}o < 0$ it would have to attain because of continuity of the solutions of a dynamical system, the value $\dot{\rho} = 0$ that is reach a critical point at which point $\rho = \rho_{eq}$. Notice that whereas this holds for gradient flow, it does not for gradient descent. As we will show later, discretization is similar to adding a second derivative in $\eta\ddot{\rho}$ to the dynamical system introducing oscillations in the dynamics.

3.2.7 Margins

Lemma 3 *At a global minimum of the loss that has minimum norm, all margins $f_n, \quad \forall n$ are within $\lambda\rho_0$ of each other, where ρ_0 is the minimum norm attained for $\lambda = 0$.*

For $\lambda = 0$ the margins are all identical. The proof of the Lemma is in Appendix E. The lemma does not say whether SGD will converge to a global minimum. The appendix discusses the issue.

3.2.8 Qualitative description of the dynamics

Recall that $0 \leq \overline{f}_n \leq 1, \quad \forall n$. Depending on the number of layers, the maximum margin that the network can achieve is usually much smaller than 1 because the weight matrices have unit norm and the bound ≤ 1 is usually very loose. Thus in order for $\rho f_n y_n$ to be equal to 1 – which means interpolation – ρ needs to be at least 1 and usually significantly larger. Let us call this minimum value for the given data set $\rho_{\min}$.

Let us assume that the network of Figure 1 is initialized with small ρ. Assume then that *average separability* holds at some time t while $\rho < \rho_{\min}$. Then Lemma 2 implies $\dot{\rho} > 0$ until $\dot{\rho} = 0$, possibly

[2] We *assume* sufficient overparametrization Equation 6 to enforce normalization of the V_k while still allowing interpolation by the ρf_n. Recall that we assume overparametrization with the W_k: here we assume that adding a single normalization constraint for each k does not eliminate overparametrization.

at $\rho_o = \rho_{min}$. Thus ρ grows monotonically until it reaches an equilibrium value. If we start from small ρ we expect ρ to be close to ρ_{min}. Recall that for $\lambda = 0$ this corresponds to a global minimum $\mathcal{L} = 0$ and that, not only global minima are degenerate, but for sufficient overparametrization they are connected in a single degenarate valley thus resulting in a large attractive basin in the loss landscape (see Appendix A). For $\lambda = 0$ at $\mathcal{L} = 0$ all the f_n have the same value, that is all the margins are the same. Similar considerations hold for $\lambda > 0$, even if in this case interpolation is impossible and is replaced for small λ by what we call quasi-interpolation (with an error $\epsilon < \lambda\rho_0$). Remarkably, under the assumption that SGD with mini batch size 1 reaches an equilibrium – that is ρ does not change between iterations – which is a global minimum, Lemma 3 shows that all the margins can be very close even when $\lambda > 0$, provided $\lambda > 0$ is small enough (which is usually the case).

If we initialize a network with large norm, Equation 2 shows that average separability yields and $\dot{\rho} < 0$. This implies that the norm of the network will decreases until an equilibrium is reached. However since $\rho \gg 1$, we expect to find an interpolating (or near interpolating) solution that is reasonably close to the initialization, since for large ρ it is usually possible to find a set of weights V_L such that $\rho|f_n| \approx 1$. This is because of the following intuition: if there are at least N units in layer L and their values are fixed, we can expect under rather weak conditions that V_L exists to yield interpolation. These large ρ, small $\overline{f}_n$ solutions are related to the Neural Tangent Kernel (NTK) solutions [21], where the parameters do not move too far from their initialization.

To sum up, starting from small initialization, gradient techniques will explore critical points with ρ growing from zero. Thus quasi-interpolating solutions with small ρ_{eq} (corresponding to large margin solutions) may be found before large ρ_{eq} quasi-interpolating solutions which have worse margin (See Fig. 2), even in the absence of regularization but especially in its presence.

3.2.9 Role of Weight Decay

Equation 8 shows that weight decay performs the traditional role of promoting solutions with small norm. In the case of large initialization, we can see from (8) that, since $|f_n|^2 \ll 1$, the scale of ρ_{eq} is determined by λ. Hence weight decay stabilizes the solution of gradient descent with respect to initialization (See Fig. 3).

Norm regularization is however not the only contribution of weight decay. Equation 10 shows that the critical points $\dot{V_k} = 0$ may not be normalized properly if the solution interpolates. In particular an un-normalized interpolating solution can satisfy the equilibrium equations for $\dot{V_k}$. This is expected from the constrained dynamics which by itself constrains the norm of the V_k to not change during the iterations[3]. By preventing exact interpolation, weight decay ensures that the critical points of the V_k dynamics lie on the unit Frobenius norm ball. As we will discuss later, by preventing exact interpolation, gradient flow with weight decay under the square loss shows at convergence the phenomenon of Neural Collapse.

3.2.10 Un-normalized vs normalized dynamics

As shown in the Appendix F, the equilibria with and without normalization are the same for ρ and V_k but the dynamics is somewhat different . Consider Figure 1. Assume that A is un-normalized, that is optimized via GD without Lagrange multipliers and B is normalized, that is optimized via GD with the Lagrange multiplier term. For A, consider, for simplicity, the case in which all the norms ρ_k of the weight matrices $1, \cdots, L-1$ are initialized with the same value. Then because of Lemma 10 all the $\rho_k, \quad \forall k = 1, \cdots, L-1$ will change together and remain equal to each other. It is the possible to consider ρ for the network of Figure 1 A as $\rho = \rho_k^L$ and look at its dynamics. Consider the case of $\lambda = 0$.

The equations for the un-normalized case are

$$\dot{\rho} = 2L\rho^{\frac{2L-2}{L}}[\sum_n \overline{f}_n - \sum_n \rho(\overline{f}_n)^2] \tag{12}$$

and

$$\dot{V_k} = -2\rho^{\frac{L-2}{L}}\sum_n (1-\rho\overline{f}_n)(\frac{\partial \overline{f}_n}{\partial V_k} - V_k\overline{f}_n) \tag{13}$$

[3] Numerical simulations show that even for linear degenerate networks convergence is independent of initial conditions only if $\lambda > 0$. In particular, normalization is then effective at ρ_0, unlike the $\lambda = 0$ case.

The equations for the normalized case (Figure 1B) are

$$\dot{\rho} = 2[\sum_n f_n y_n - \sum_n \rho (f_n)^2] \tag{14}$$

and

$$\dot{V_k} = 2\rho \sum_n [(1 - \rho \overline{f}_n)(V_k \overline{f}_n - \frac{\partial \overline{f}_n}{\partial V_k})]. \tag{15}$$

Recall that for $\lambda = 0$ ρ_0 corresponds to the inverse of the margin: thus $\frac{1}{\rho_0} = f_n$, since f_n is the same for all n. Thus $\dot{V_k}$, is proportional to the inverse of the margin in the normalized case but to something smaller in the un-normalized case. The factor combines with the learning rate when Gradient Descent replaces gradient flow. Intuitively, the strategy to decrease the learning rate when the margin is large seems a good strategy, since large margin corresponds to "good" minima in terms of generalization (for classification).

4 Margin and generalization

Assume that the square loss is exactly zero and the margins f_n are all the same. Then recall simple generalization bounds that hold with probability at least $(1 - \delta)$, $\forall g \in \mathbb{G}$ of the form [33]:

$$|L(g) - \hat{L}(g)| \leq c_1 \mathbb{R}_N(\mathbb{G}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}} \tag{16}$$

where $L(g) = \mathbf{E}[\ell_\gamma(g(x), y)]$ 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* $\ell_\gamma(g(x), y)$ defined as

$$\ell_\gamma(y, y') = \begin{cases} 1, & \text{if} \quad yy' \leq 0, \\ 1 - \frac{yy'}{\gamma}, & \text{if} \quad 0 \leq yy' \leq \gamma, \\ 0, & \text{if} \quad yy' \geq \gamma. \end{cases}$$

We define $\ell_{\gamma=0}(y, y')$ as the standard $0 - 1$ classification error and observe that $\ell_{\gamma=0}(y, y') < \ell_{\gamma>0}(y, y')$.

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 ρs. 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 RELU networks, the empirical Rademacher complexity satisfies the property,

$$\mathbb{R}_N(\mathbb{G}) = \rho \mathbb{R}_N(\mathbb{F}), \tag{17}$$

where $\mathbb{G}$ is the space of functions of our unnormalized networks and $\mathbb{F}$ denotes the corresponding normalized networks. This observation allows us to use the bound Equation 16 and the fact that the empirical $\hat{L_\gamma}$ *for both functions is the same* to write $L_0(f^a) = L_0(F^a) \leq c_1 \rho_a \mathbb{R}_N(\tilde{\mathbb{F}}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}$ and $L_0(f^b) = L_0(F^b) \leq c_1 \rho_b \mathbb{R}_N(\tilde{\mathbb{F}}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}$. The bounds have the form

$$L_0(f^a) \leq A\rho_a + \epsilon \tag{18}$$

and

$$L_0(f^b) \leq A\rho_b + \epsilon \tag{19}$$

Thus the upper bound for the expected error $L_0(f^a)$ is better than the bound for $L_0(f^b)$. Of course this is just an upper bound. Lower bounds are not available. As a consequence this result does not guarantee that a solution with smaller ρ will always have a smaller expected error than a solution with larger ρ.

5 Extending the toy model

6 Toy model with GD

Work in [34] shows that a natural approximation to gradient descent within a continuous gradient flow formulation is to add to the loss functional $\mathcal{L}$ a term proportional to $\frac{\eta}{4}$, consisting of the norm square of the gradient of $\mathcal{L}$. This is equivalent (see [35]) to replacing in the gradient flow equation terms like $\dot{x}$ with terms that are $\frac{\eta}{2}\ddot{x} + \dot{x}$. The informal explanation is that the gradient descent term $x(t+\eta) - x(t) = -\eta F$ can be approximated by expanding $x(t+\eta)$ in a Taylor series for small η to a quadratic approximation, that is $x(t+\eta) \approx x(t) + \eta\dot{x}(t) + \frac{\eta^2}{2}\ddot{x}(t)$. Thus the gradient descent equation becomes $\dot{x}(t) + \frac{\eta}{2}\ddot{x}(t) = -F$.

With this approximation Equations 5, 6 and 7 become (taking into account that $\mathcal{L} = \sum_n (1-\rho\overline{f}_n)^2 + \nu\sum_{k=1}^{L-1}||V_k||^2 + \lambda\rho^2$)

$$\frac{\eta}{2}\ddot{\rho} + \dot{\rho} = 2[\sum_n (1-\rho\overline{f}_n)\overline{f}_n] - 2\lambda\rho \tag{20}$$

$$\frac{\eta}{2}\ddot{V_k} + \dot{V_k} = 2\rho\sum_n [(1-\rho\overline{f}_n)(V_k\overline{f}_n - \frac{\partial\overline{f}_n}{\partial V_k})] \quad \forall k < L \tag{21}$$

$$\frac{\eta}{2}\ddot{W}_L + \dot{W}_L = 2\sum_n [(1-\overline{g}_n)\frac{\partial\overline{g}_n}{\partial W_L})] - 2\lambda W_L \tag{22}$$

As a sanity check we see that $W^T\dot{W} = \rho\dot{\rho}$ since $W_L = \rho_L V_L$, $||V_L|| = 1$, $g_n = \rho f_n$.
We multiply the last equation on the left by W^T obtaining

$$\frac{\eta}{2}W^T\ddot{W}_L + W^T\dot{W}_L = 2\sum_n (1-\overline{g}_n)\overline{g}_n - 2\lambda W_L^2 \tag{23}$$

We change variables in the last equation:

$$\frac{\eta}{2}W^T\ddot{W}_L + W^T\dot{W}_L = 2\rho\sum_n [(1-\rho\overline{f}_n)\overline{f}_n) - 2\lambda\rho^2 \tag{24}$$

Since $\dot{W}_L = \rho\dot{V}_L + \dot{\rho}V_L$ and $\ddot{W}_L = 2\dot{\rho}\dot{V}_L + \rho\ddot{V}_L + \ddot{\rho}V_L$, the last equation becomes for $\dot{\rho} = 0$

$$\rho V^T\frac{\eta}{2}\rho\ddot{V}_L + \rho V^T\rho\dot{V}_L = 2\rho\sum_n [(1-\rho\overline{f}_n)\overline{f}_n) - 2\lambda\rho^2 \tag{25}$$

Now notice the following simple relation between accelerations and velocities: $\frac{\partial(V^TV)}{\partial t} = 2V^T\dot{V}$ and $\frac{\partial^2(V^TV)}{\partial t^2} = 2(\dot{V}^T\dot{V} + V^T\ddot{V}) = 2(||\dot{V}||^2 + V^T\ddot{V})$. If the norm $||V_L||$ is constant, then $\frac{\partial^2(V^TV)}{\partial t^2} = 0$. It follows $V^T\ddot{V} = -||\dot{V}||^2$.

Thus

$$\rho^2 V^T\frac{\eta}{2}\ddot{V}_L = 2\rho\sum_n ((1-\rho\overline{f}_n)\overline{f}_n - 2\lambda\rho^2 \tag{26}$$

$$-\frac{\eta}{2}\rho||\dot{V}_L||^2 = 2\sum_n (1-\rho\overline{f}_n)\overline{f}_n - 2\lambda\rho \tag{27}$$

Equation 20 implies that when $\dot{\rho} = 0$ then $||\dot{V}_L||^2 = 0$.

The new dynamical system with the η term coming from discretization, may show oscillations(the frequency of the "undamped oscillations" is $\frac{4(\frac{1}{N}\sum\overline{f}_n^2 + 2\lambda)}{\eta}$). In the equations above whenever $1 > \sqrt{\frac{2}{N}\sum_n\overline{f}_n^2 + 2\lambda}$, the linearized dynamics is the dynamics of a damped oscillator.

6.1 Multiclass

For Andy and Akshay to do

7 Neural Collapse

In a recent paper Papyan, Han and Donoho[4] 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).

In this section we show that the phenomenon of neural collapse can be derived from the critical points of gradient flow under the square loss with Weight Decay. We consider a multiclass classification problem with C classes with a balanced training dataset $\mathcal{S} = \{(x_n, y_n)\}$ that has N training examples per class. We train a ReLU deep network $f_W : \mathbb{R}^d \to \mathbb{R}^C$, $f_W(x) = W_L\sigma(W_{L-1} \dots W_2\sigma(W_1 x)\dots)$ with Gradient Descent on the square loss with Weight Decay on the parameters of the network. This architecture differs from the one considered in section 3 in that it has C outputs instead of a scalar output. Let the output of the network be $f_W(x) = [f_W^{(1)}(x) \dots f_W^{(C)}(x)]^\top$, and the one-hot target vectors be $y_n = [y_n^{(1)} \dots y_n^{(C)}]^\top$. We will also follow the notation of [4] and use $h(x)$ to denote the last layer features of the deep network. This means that $f_W^{(c)}(x) = \langle W_L^c, h(x)\rangle$. We make a key assumption here, that the solution obtained by Gradient Descent satisfies the following condition

Assumption 1 (Symmetric Quasi-interpolation) *Consider a C-class classification problem with inputs in a feature space $\mathcal{X}$, a classifier $f : \mathcal{X} \to \mathbb{R}^C$ symmetrically quasi-interpolates a training dataset $S = \{(x_n, y_n)\}$ if for all training examples $x_{n(c)}$ in class c, $f^{(c)}(x_{n(c)}) = 1 - \epsilon$, and $f^{(c')}(x_{n(c)}) = \frac{\epsilon}{C-1}$.*

We observe here that ϵ depends on the Weight Decay parameter λ. In the case of binary classification for instance, with Neural Collapse and quasi-interpolation, and using similar notation as in the previous section:

$$\epsilon \approx \frac{\lambda}{\lambda + \sum_n |f_n|^2}, \qquad \lambda \approx \frac{\epsilon \sum_n |f_n|^2}{1-\epsilon} \tag{28}$$

This brings us to the main result of this section:

Theorem 4 *For a ReLU deep network trained on a balanced dataset using gradient flow on the square loss with weight decay λ, critical points of Gradient Flow that satisfy Assumption 2 also satisfy the NC1-4 conditions for Neural Collapse.*

Proof Our training objective is $\mathcal{L}(W) = \frac{1}{2}\sum_{n=1}^{NC}\sum_{i=1}^{C}\left(y_n^{(i)} - f_W^{(i)}(x_n)\right)^2 + \frac{\lambda}{2}\sum_l ||W_l||_F^2$. We use gradient flow to train the network: $\frac{\partial W}{\partial t} = -\frac{\partial \mathcal{L}}{\partial W}$. Let us analyze the dynamics of the last layer, considering each classifier vector W_L^c of W_L separately:

$$\begin{aligned}\frac{\partial W_L^c}{\partial t} &= \sum_n (y_n^c - \langle W_L^c, h(x_n)\rangle)h(x_n) - \lambda W_L^c \\ &= \sum_{n \in N(c)} (1 - \langle W_L^c, h(x_{n(c)})\rangle)h(x_{n(c)}) + \sum_{n \in N(c'), c' \neq c} (-\langle W_L^c, h(x_{n(c')})\rangle)h(x_{n(c')}) - \lambda W_L^c\end{aligned} \tag{29}$$

Let us consider solutions that achieve *symmetric quasi-interpolation*, with $f_W^{(c)}(x_{n(c)}) = 1 - \epsilon$, and $f_W^{(c)}(x_{n(c')}) = \frac{\epsilon}{C-1}$. It is fairly straightforward to see that since $f_W^{(c)}$ and W_L^c do not depend on n, neither does $h(x_n)$, which shows NC1. Under the conditions of NC1 we know that all feature vectors in a class collapse to the class mean, i.e., $h(x_{n(c)}) = \mu_c$. Let us denote the global feature mean by $\mu_G = \frac{1}{C}\sum_c \mu_c$. This means we have:

$$\frac{\partial W_L^c}{\partial t} = 0 \qquad \implies W_L^c = \frac{CN\epsilon}{\lambda(C-1)} \times (\mu_c - \mu_G) \tag{30}$$

This implies that the last layer parameters W_L are a scaled version of the centered class-wise feature matrix $M = [\ldots \mu_c - \mu_G \ldots]$. Thus at equilibrium, with quasi interpolation of the training labels, we obtain $\frac{W_L}{||W_L||_F} = \frac{M^\top}{||M|_F}$. This is the condition for NC3.

From the gradient flow equations, we can also see that at equilibrium, with quasi interpolation, all classifier vectors in the last layer (W_L^c, and hence $\mu_c - \mu_G$) have the same norm:

$$\begin{aligned} \left\langle W_L^c, \frac{\partial W_L^c}{\partial t} \right\rangle &= \sum_n (y_n^c - f_W^{(c)}(x_n)) f_W^{(c)}(x_n) - \lambda ||W_L^c||_2^2 = 0 \\ \implies ||W_L^c||_2^2 &= \frac{N}{\lambda}\left(\epsilon - \frac{C}{C-1}\epsilon^2\right) \end{aligned} \tag{31}$$

From the quasi-interpolation of the correct class label we have that $\langle W_L^c, \mu_c \rangle = 1 - \epsilon$ which means $\langle W_L^c, \mu_G \rangle + \langle W_L^c, \mu_c - \mu_G \rangle = 1 - \epsilon$. Now using (71)

$$\begin{aligned} \langle W_L^c, \mu_G \rangle &= 1 - \epsilon - \frac{\lambda(C-1)}{CN\epsilon} ||W_L^c||^2 \\ &= 1 - \epsilon - \frac{\lambda(C-1)}{CN\epsilon} \times \frac{N}{\lambda}\left(\epsilon - \frac{C}{C-1}\epsilon^2\right) = \frac{1}{C}. \end{aligned} \tag{32}$$

From the quasi-interpolation of the incorrect class labels, we have that $\langle W_L^c, \mu_{c'} \rangle = \frac{\epsilon}{C-1}$, which means $\langle W_L^c, \mu_{c'} - \mu_G \rangle + \langle W_L^c, \mu_G \rangle = \frac{\epsilon}{C-1}$. Plugging in the previous result and using (72) yields

$$\begin{aligned} \frac{\lambda(C-1)}{CN\epsilon} \times \langle W_L^c, W_L^{c'} \rangle &= \frac{\epsilon}{C-1} - \frac{1}{C} \\ \implies \langle V_L^c, V_L^{c'} \rangle &= \frac{1}{||W_L^c||_2^2} \times \frac{CN\epsilon}{\lambda(C-1)} \times \left(\frac{\epsilon}{C-1} - \frac{1}{C}\right) = -\frac{1}{C-1} \end{aligned} \tag{33}$$

Here $V_L^c = \frac{W_L^c}{||W_L^c||_2}$, and we use the fact that all the norms $||W_L^c||_2$ are equal. This completes the proof that the normalized classifier parameters form an ETF. Moreover since $W_L^c \propto \mu_c - \mu_G$ and all the proportionality constants are independent of c, we obtain $\sum_c W_L^c = 0$. This completes the proof of the NC2 condition. NC4 follows then from NC1-NC2, as shown by theorems in [4]. ■

It is of interest to note here that in this quasi interpolation setting, the functional classification margin is given by $\eta_n = f_{y_n} - \max_{c \neq y_n} f_c = 1 - \epsilon - \frac{\epsilon}{C-1} = 1 - \frac{C}{C-1}\epsilon$. The larger the margin, the smaller is ϵ. Eq. (72) shows that the norms of the classifier weights are given by $||W_L^c||_2^2 = \frac{N\epsilon}{\lambda}\eta$. As we mentioned earlier, for a non-zero value of λ we expect some small interpolation error ϵ. In the binary case this is given by Eq. (28). Plugging this relationship into Eq. (72) we obtain $||W_c||^2 \approx \frac{N(1-\epsilon)}{\sum_n |f_n|^2}\eta$. This means that the lengths of the classifier simplex ETF are proportional to the margin.

Other settings The main assumptions in the above proof are symmetric quasi-interpolation and the use of Weight Decay (see section 5 of Supplementary Material). A similar version of the above proof can be adapted to the case of Stochastic Gradient Descent (SGD), where we can show that the NC conditions are met in expectation. We also show in section 5 of the Supplementary Material that an extension of this proof technique to the exponential loss case (a proxy for cross-entropy loss) requires small batch SGD to achieve the NC1 property.

Predictions We summarize here the main predictions of our analysis about Neural Collapse.

- The theoretical analysis predicts Neural Collapse not only for the case of cross-entropy, for which it was empirically found, but also for the square loss;
- SGD is required in our proof of NC1 (and hence the other NC conditions) for cross entropy while for the square loss neural collapse is predicted for SGD as well as GD (under Assumption 2);

- Our proof uses Weight Decay for neural collapse (NC1 to NC4) under both the square loss and the cross entropy, and can also be adapted to the case of normalization and Weight Decay;
- The length of the vectors in the Simplex ETF that defines the classifier is proportional to the training margin;
- NC1 to NC4 should take place for any quasi - interpolating solutions (in the square loss case), including solutions that do not have large margin (that is, small ρ);
- in particular the analysis above predicts Neural Collapse for randomly labeled CIFAR10.

8 Experiments

We conducted a number of experiments on binary classification to support our claims from the analysis of the dynamics. We conducted our experiments on the standard CIFAR10 dataset [36]. Image samples with class label indices 1 and 2 were extracted for the binary classification task. The total training and test data sizes are 10000 and 2000, respectively. The model architecture contains 3 convolutional Layers (the number of channels are 32, 64 and 128, filter size is 3×3) and one fully connected classifier layer with output number 2. Following each convolution layer, we applied a ReLU nonlinear activation function and Batch Normalization. Batch Normalization is used with learnable "affine" shifting and scaling parameters turned off (since they can always be learned by the next layer). 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 one of the initialization value set {0.01, 0.1, 0.5, 1, 3, 5, 10}. The network was trained using square loss and SGD with batch size 128, momentum 0.9, Weight Decay (0.01 or 0), constant learning rate 0.01 for 1000 epochs and no data augmentation. Every input to the network is scaled such that it has norm ≤ 1. The plots in figure and 3 are averaged over 10 different runs, while figures 2, 4 and 5 were made from a single run.

In Fig. 2 we show the dynamics of ρ alongside train loss and test error. We show results with and without Weight Decay in the top and bottom rows of Fig. 2 respectively. The left and right columns correspond to small (0.01) and large (5) initializations respectively. We see that without Weight Decay, with small initializations, ρ grows monotonically, while with large initializations it decays monotonically. We can also see that small initializations without Weight Decay reach minima with smaller train loss. The top row plots also show that Weight Decay makes the final solutions robust to the scale of initialization, in terms of ρ and of the train loss. This robustness is also seen in Fig. 3, where we plot the training margins ($y_n f_n$) obtained with and without Weight Decay. In the right plot, without Weight Decay, the margin distributions depend on the initialization, while in the left plot they cluster around the same values.

Finally, we would like to setup some motivating empirical evidence for our discussion of Neural Collapse [4]. Neural Collapse is the phenomenon in which within class variability disappears, and for all training samples, the last layer features collapse to their mean. This means that the outputs and margins also collapse to the same value. We can see this in the left plot of Fig. 3 where all of the margin histograms are concentrated around a single value. We visualize the evolution of the training margins over the training epochs in Fig. 4 which shows that the margin distribution concentrates over time. At the final epoch the margin distribution (colored in yellow) is much narrower than at any intermediate epochs. We also used measurements similar to those in [4] to confirm that Neural Collapse indeed occurs by the appropriate metrics. This is shown in Fig. 5 where we trained the same network as described earlier with a modified learning rate schedule for 350 epochs, and plot the conditions for NC1 and NC2. Section 6 of the supplementary material contains a longer discussion of these conditions, though one can also be found in [4].

9 Discussion

An important question is whether Neural Collapse is related to good generalization of the solution of training. Our analysis suggests that this is not the case: Neural Collapse is a property of the dynamics independently of the size of the margin which provides an upper bound on the expected error – even if margin is likely to be just one of the factors determining out-of-sample performance. In fact, our

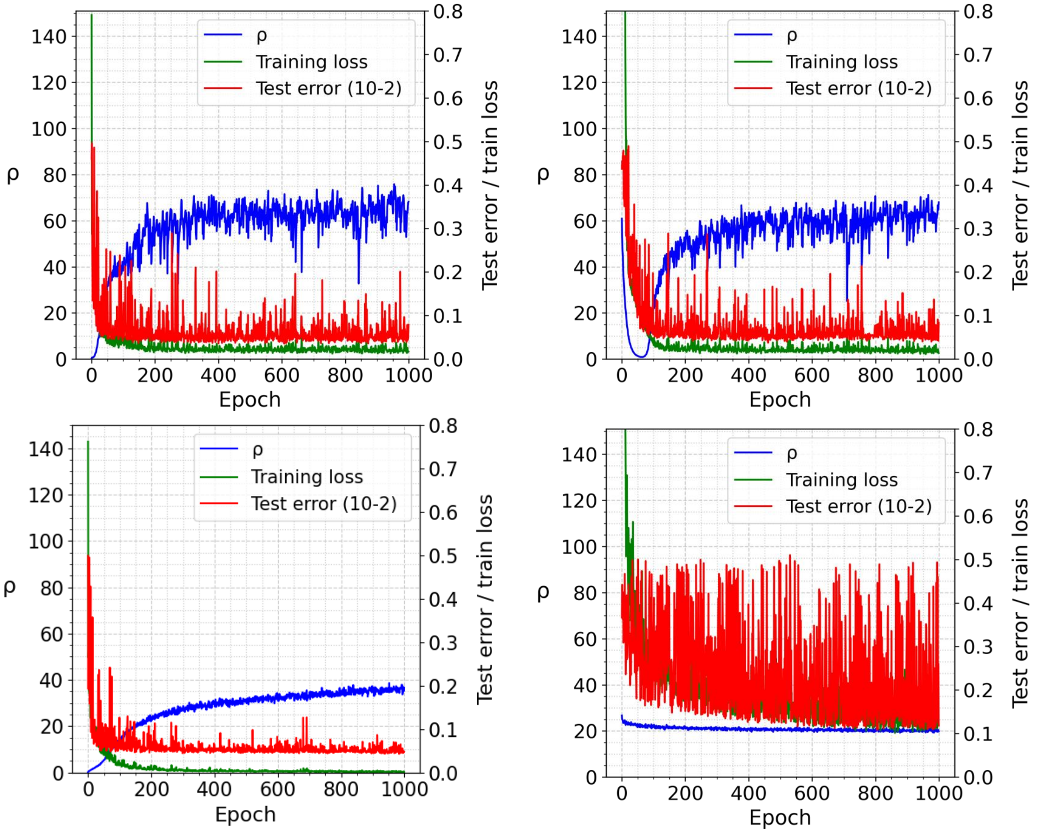

Figure 2: *Training dynamics of product norm ρ, training loss and test error over 1000 epochs with small initialization (0.01) in the first column and large initialization (5) in the second column. The first row is with Weight Decay = 0.01, and the second row is with Weight Decay = 0.*

prediction of Neural Collapse for randomly labeled CIFAR10, has been confirmed in preliminary experiments by our collaborators.

Despite the fact that Neural Collapse is independent of generalization, can our analysis of the square loss provide insights on generalization of the solutions of gradient flow? It is well known that large margin is usually associated with good generalization[33]; in the meantime it is also broadly recognized that margin alone does not fully account for generalization in deep nets[32, 37, 38]. Margin in fact provides an upper bound on generalization error, as shown in section 4 of the supplementary material. Larger margin gives a better upper bound on the generalization error for the same network trained on the same data. This property can be checked qualitatively by varying the margin using different degrees of random labels in a binary classification task (see Figure 1 in supplementary material). While training gives perfect classification and almost zero square loss, the margin on the training set increases and the test error also increases with the percentage of random labels as shown in the figure 1 in the supplementary material. However, the simple upper bound given in the same section does not explain the generalization behavior that we observe for different initializations (see Figure 2 in supplementary material), where small differences in margin are actually anticorrelated with small differences in test error.

Notice that the generalization bound in Section 4 of the supplementary material does not directly rely on the Weight Decay parameter $\lambda > 0$. However, robust convergence to large margins is helped by a non-zero λ, even if λ is quite small, because of the associated greater independence from initial conditions in degenerate minima. This effect is different from the standard explanation that regularization is needed to force the norm to be small.

The main effect of $\lambda > 0$ is to eliminate degeneracy of the dynamics at the zero-loss critical points, where Equation 4 is degenerate if $\lambda = 0$. In fact, $\dot{V}_k = 0$ can be satisfied even when $(V_k f_n - \frac{\partial f_n}{\partial V_k}) \neq 0$, implying that any interpolating solution can satisfy the equilibrium equations independently of its

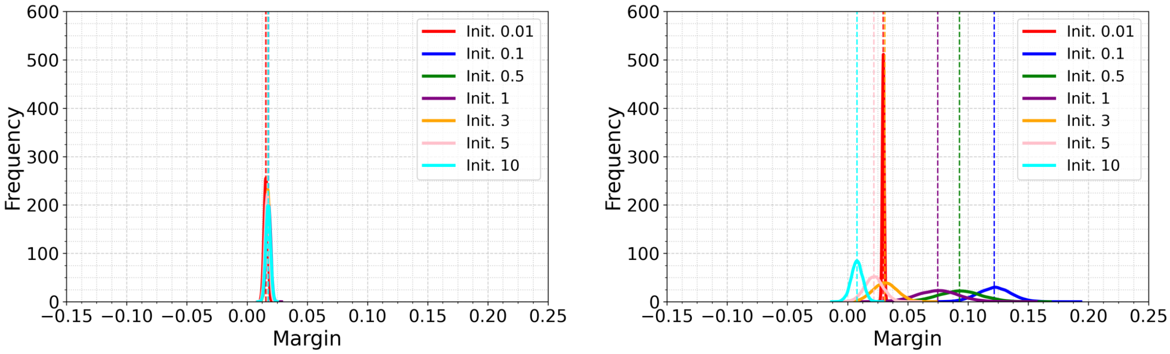

Figure 3: *Mean training margins over 10 runs for binary classification on CIFAR10 trained with Batch Normalization and Weight Decay = 0.01 (left) and without Weight Decay (right) for different initializations ($init.$ = 0.01, 0.1, 0.5, 1, 3, 5 and 10). Weight Decay makes the final training margin robust to initialization, and concentrates the margin in a narrow band over the training set. The results without Weight Decay are dependent on initialization, and may result in a wide range of margin values.*

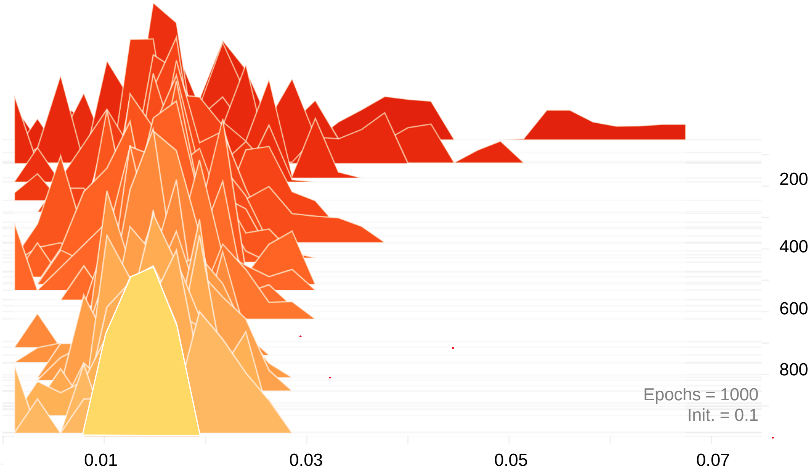

Figure 4: *Histogram of $|f_n|$ across 1000 training epochs for binary classification with batch normalization and weight decay = 0.01, learning rate 0.01, initialization 0.1. We can see that the histogram narrows as training progresses. The final histogram (in yellow) is concentrated in a narrow band, as expected for the emergence of NC1.*

normalization. This degeneracy is expected, since there are infinite sets of ρ and V_k satisfying $\rho V_k = W_k$. Normalization thus is not effective at the critical points. Setting $\lambda > 0$ avoids this degeneracy.

Limitations The theoretical analysis in this paper rests on several assumptions. We showed that the assumption of a symmetric level of near interpolation implies Neural Collapse. We did not, however, formally prove that SGD on a randomly initialized network will necessarily converge to near interpolation. Our analysis only says that, with L2 regularization, the critical points of SGD that coincide with the minimum for the associated ρ_0 yield similar margins for all n (our proof is for the binary case). Thus SGD with weight decay and normalization techniques (see supplementary material) is sufficient to yield Neural Collapse. Importantly, the necessity of all the conditions remains an open problem.

Acknowledgments This material is based upon work supported by the Center for Minds, Brains and Machines (CBMM), funded by NSF STC award CCF-1231216. This research was also sponsored by grants from the National Science Foundation (NSF-0640097, NSF-0827427), and AFSOR-THRL (FA8650-05-C-7262).

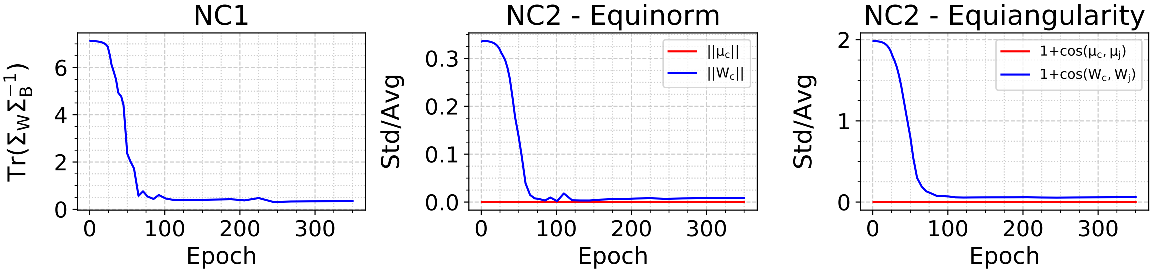

Figure 5: *Neural Collapse occurs during training for binary classification. The key conditions for Neural Collapse are: (i) NC1 - Variability collapse, which is measured by $Tr(\Sigma_W \Sigma_B^{-1})$, where Σ_W, Σ_B are the within and between class covariances, and (ii) NC2 - equinorm and equiangularity of the mean features $\{\mu_c\}$ and classifiers $\{W_c\}$. We measure the equinorm condition by the standard deviation of the norms of the means (in red) and classifiers (in blue) across classes, divided by the average of the norms, and the equiangularity condition by the standard deviation of the inner products of the normalized means (in red) and the normalized classifiers (in blue), divided by the average inner product. This network was trained on two classes of CIFAR10 with Batch Normalization and Weight Decay = 0.01, learning rate 0.01, initialization 3 for 350 epochs with a stepped learning rate decay schedule.*

References

[1] Like Hui and Mikhail Belkin. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks. *arXiv preprint arXiv:2006.07322*, 2020.

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

[3] Tomaso Poggio, Andrzej Banburski, and Qianli Liao. Theoretical issues in deep networks. *PNAS*, 2020.

[4] Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[5] 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.

[6] 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.

[7] Ryan M. Rifkin. *Everything Old Is New Again: A Fresh Look at Historical Approaches to Machine Learning*. PhD thesis, Massachusetts Institute of Technology, 2002.

[8] T. Poggio and Q. Liao. Generalization in deep network classifiers trained with the square loss. *CBMM Memo No. 112*, 2019.

[9] T. Poggio and Y. Cooper. Loss landscape: Sgd has a better view. *CBMM Memo 107*, 2020.

[10] Yaim Cooper. Global minima of overparameterized neural networks. *SIAM Journal on Mathematics of Data Science*, 3(2):676–691, 2021.

[11] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[12] Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

[13] Tengyu Xu, Yi Zhou, Kaiyi Ji, and Yingbin Liang. When will gradient methods converge to max-margin classifier under relu models? *Stat*, 10(1):e354, 2021.

[14] Vidya Muthukumar, Adhyyan Narang, Vignesh Subramanian, Mikhail Belkin, Daniel Hsu, and Anant Sahai. Classification vs regression in overparameterized regimes: Does the loss function matter? *arXiv e-prints*, page arXiv:2005.08054, May 2020.

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

[16] Tengyuan Liang and Benjamin Recht. Interpolating classifiers make few mistakes. *arXiv preprint arXiv:2101.11815*, 2021.

[17] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *International conference on machine learning*, pages 4140–4149. PMLR, 2017.

[18] Mahdi Soltanolkotabi, Adel Javanmard, and Jason D Lee. Theoretical insights into the optimization landscape of over-parameterized shallow neural networks. *IEEE Transactions on Information Theory*, 65(2):742–769, 2018.

[19] 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.

[20] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.

[21] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*, 2018.

[22] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[23] Zhengdao Chen, Grant M Rotskoff, Joan Bruna, and Eric Vanden-Eijnden. A dynamical central limit theorem for shallow neural networks. *arXiv preprint arXiv:2008.09623*, 2020.

[24] Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

[25] Dustin G. Mixon, Hans Parshall, and Jianzong Pi. Neural collapse with unconstrained features. *CoRR*, abs/2011.11619, 2020.

[26] Jianfeng Lu and Stefan Steinerberger. Neural collapse with cross-entropy loss. *CoRR*, abs/2012.08465, 2020.

[27] Cong Fang, Hangfeng He, Qi Long, and Weijie J. Su. Layer-peeled model: Toward understanding well-trained deep neural networks. *CoRR*, abs/2101.12699, 2021.

[28] Stephan Wojtowytsch et al. On the emergence of tetrahedral symmetry in the final and penultimate layers of neural network classifiers. *arXiv preprint arXiv:2012.05420*, 2020.

[29] Tolga Ergen and Mert Pilanci. Revealing the structure of deep neural networks via convex duality. *arXiv preprint arXiv:2002.09773*, 2020.

[30] T. Poggio and Q. Liao. Generalization in deep network classifiers trained with the square loss1. *Center for Brains, Minds and Machines (CBMM) Memo No. 112*, 2021.

[31] Sanjeev Arora, Zhiyuan Li, and Kaifeng Lyu. Theoretical analysis of auto rate-tuning by batch normalization. *CoRR*, abs/1812.03981, 2018.

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

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

[34] David G. T. Barrett and Benoit Dherin. Implicit gradient regularization, 2021.

[35] Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel L. K. Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics, 2021.

[36] A Krizhevsky. Learning multiple layers of features from tiny images. *Technical Report*, 2009.

[37] Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. Predicting the generalization gap in deep networks with margin distributions. *arXiv preprint arXiv:1810.00113*, 2018.

[38] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

[39] T. Poggio and Y. Cooper. Loss landscape: Sgd can have a better view than gd. *CBMM memo 107*, 2020.

[40] Quynh Nguyen. On connected sublevel sets in deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4790–4799. PMLR, 09–15 Jun 2019.

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

[42] Daniel Gissin, Shai Shalev-Shwartz, and Amit Daniely. The Implicit Bias of Depth: How Incremental Learning Drives Generalization. *arXiv e-prints*, page arXiv:1909.12051, September 2019.