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

October 11, 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 ρ, that we call the "norm" of the function implemented by the network and which is the product of the Frobenius norms of each layer weight matrices in the network, is expected when normalization by a Lagrange multiplier (LN) is used for each layer but the last one, together with Weight Decay (WD). For $\lambda \to 0$ this would be the minimum ρ interpolating solution. 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 (see Appendix G). The main property of the minimizers that bounds their expected error is ρ: we prove that among all the interpolating or quasi-interpolating solutions (for $\lambda > 0$), 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 related 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. We perform numerical simulations to support parts of our theoretical analysis.

In summary, this paper describes how gradient descent on the square loss can converge to minimum ρ solutions, corresponding to max margin solutions. Associated with those solutions are upper bounds on the generalization error and properties such as Neural Collapse. Our analysis of the square loss does not uncover any specific bias of gradient descent apart from the preference for minimum ρ solution because of regularization and initialization. This supports the idea that the advantage of deep networks relative to other standard classifiers is restricted to specific deep architectures such as CNNs and is due to their good approximation properties for target functions that are locally compositional.

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 from previous results (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 Lagrange 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 qualitative dynamics of the the deep network parameters and show that ρ which is the product of the Frobeniu norms of each weight matrix, grows non-monotonically until minimum ρ, that is large margin, solutions are reached. Since local minima and saddle points can be avoided this analysis shows that with small initialization and weight decay there will be convergence to a global minimum which, in addition, may have a minimum ρ. In the following, we will occasionally refer to ρ as the "norm" of the function represented by the deep RELU network.

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 $|f(x_n) - y_n| \leq \epsilon, \qquad \forall n$, where ϵ is small. Our experiments and analysis of the dynamics show that, depending on the weight decay parameter, there is a stronger 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 gradient descent. 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 discuss its extension to 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 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 formulate a toy model that assumes three simplifying assumptions. For this model we analyze the dynamics of gradient flow under the square loss for a binary classification problem. We use an analysis of this continuous dynamics to illuminate the role of Weight Decay and Batch/Weight Normalization.

We then apply a simple generalization bound to link margin to expected error, in particular in the interpolation or quasi-interpolation case when the empirical error is zero or close to zero. We then extend in section 5 the toy model from Gradient Flow to Gradient Descent, eliminating our second simplifying assumption. In section 6 we use our analysis of the dynamics in the binary classification case to justify an assumption of margins being very close to each other at convergence to predict the phenomenon of Neural Collapse when training on the square loss. The supplementary material extends the proof to the case of exponential loss functions. In section 7 we present and describe our experiments on CIFAR10 that demonstrate the results of section 3. We conclude in section 8 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

We make three simplifying 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 – usually performed using Batch Normalization – is modeled here by adding a Lagrange multiplier term to a modified square loss function.

In a later section we will analyze Neural Collapse without the first assumption, leaving to later work the task to extend in the same way our analysis of the dynamics. We will also describe how to extend our analysis in the absence of the second assumption.

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 $f_W : \mathbb{R}^d \to \mathbb{R}$, $f_W(x) = W_L \sigma(W_{L-1} \dots \sigma(W_1 x) \dots)$, 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).

3.3 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 $f_W(x) = \rho_L V_L \sigma(\rho_{L-1} \dots \sigma(\rho_1 V_1 x) \dots)$. 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 $f_W(x) = \rho f_V(x) = \rho V_L \sigma(V_{L-1} \dots \sigma(V_1 x) \dots)$. 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.

3.4 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 [6]). 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 $V_k, \quad k = 1, \cdots, L$ 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 and then multiplying the output by ρ, is equivalent to let $W_L = \rho V_L$ be unnormalized. Thus f_V is the network that at convergence has $L-1$ normalized layers.

Constrained minimization of $\mathcal{L} = \frac{1}{N} \sum_n (\rho f_n - y_n)^2 + \lambda \rho^2$ under the constraint $||V_k||^2 = 1$ leads to minimizing

$$\mathcal{L} = \frac{1}{N} \sum_n (\rho f_n - y_n)^2 + \sum_{k=1}^{L} \nu_k (||V_k||^2 - 1) + \lambda \rho^2 = \frac{1}{N} \sum_n (1 - \rho \overline{f}_n)^2 + \sum_{k=1}^{L} \nu_k (||V_k||^2 - 1) + \lambda \rho^2. \quad (1)$$

3.4.1 Separability, margin, average margin and its standard deviation

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 the minimum of $\mathcal{L}$ for $\lambda = 0$ is zero and corresponds to separability. Decreasing $\mathcal{L}$ corresponds to increasing average separability.

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

$$\rho f_n = y_n \quad (2)$$

where we call $y_n f_n = \overline{f}_n$ *the margin for* x_n. By multiplying both sides of Equation 2 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 μ in the limit of $\lambda = 0$, with $\mu = \frac{1}{N} \sum_n \overline{f}_n$. It is also useful to define *the margin variance* $\sigma^2 = M - \mu^2$ *with* $M = \frac{1}{N} \sum_n \overline{f}_n^2$. Notice that $M = \frac{1}{N} \sum_n \overline{f}_n^2 = \sigma^2 + \mu^2$ and that both $M \geq 0$ and $\sigma^2 = \sum_n (\overline{f}_n - \mu)^2 \geq 0$ are not negative.

[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]).

3.4.2 Toy model: dynamics

The gradient flow obtained from Equation 1 gives

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

and

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

In the latter equation we can use the unit norm constraint on the $||V_k||$ to determine the Lagrange multipliers ν_k. Using the structural lemma (Appendix B), the constraint $||V_k||^2 = 1$ implies $\frac{\partial ||V_k||}{\partial t} = V_k^T \dot{V_k} = 0$, which gives

$$\nu_k = \frac{1}{N}\sum_n (\rho\overline{f}_n - \rho^2 f_n^2) = \frac{1}{N}\sum_n \rho\overline{f}_n(1-\rho f_n). \tag{5}$$

Thus the gradient flow is the following dynamical system

$$\dot{\rho} = \frac{2}{N}[\sum_n \overline{f}_n - \sum_n \rho(\overline{f}_n)^2] - 2\lambda\rho \tag{6}$$

$$\dot{V_k} = \frac{2}{N}\sum_n (1-\rho\overline{f}_n)\rho\frac{\partial \overline{f}_n}{\partial V_k} - \frac{2}{N}V_k\rho\sum_n \overline{f}_n(1-\rho\overline{f}_n) = \frac{2}{N}\rho\sum_n [(1-\rho\overline{f}_n)(-V_k\overline{f}_n + \frac{\partial \overline{f}_n}{\partial V_k})] \tag{7}$$

We can also write the same dynamics in terms of μ, M and σ as

$$\dot{\rho} = 2(\mu - \rho(M+\lambda)) \tag{8}$$

$$\dot{V_k} = 2\rho\mu - \rho^{\frac{\partial \mu^2}{\partial V_k}} - V_k\rho\mu + V_k\rho^2 M \tag{9}$$

$$\dot{\mu} = \sum_k \frac{\partial \mu}{\partial V_k}\dot{V_k} = \sum_k \rho\mu\frac{\partial \mu}{\partial V_k} - \rho\frac{\partial \mu}{\partial V_k}\frac{\partial \mu^2}{\partial V_k} - \rho\mu^2 + \mu M\rho^2 \tag{10}$$

3.4.3 Toy model: critical points

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

$$\rho_{\text{eq}} = \frac{\frac{1}{N}\sum_n \overline{f}_n}{\lambda + \frac{1}{N}\sum_n \overline{f}_n^2} = \frac{\sum_n \overline{f}_n}{\lambda + \sum_n \overline{f}_n^2} = \frac{\mu}{M+\lambda} \tag{11}$$

We define ρ_0 the value of ρ_{eq} when $\lambda = 0$.
The critical points of the V_k dynamics – that is when $\dot{V_k} = 0$ – satisfy

$$\frac{1}{N}\sum_n [(1-\rho\overline{f}_n)(V_k\overline{f}_n - \frac{1}{N}\frac{\partial \overline{f}_n}{\partial V_k})] = 0. \tag{12}$$

In the following we *assume that local minima with $\mathcal{L} > 0$ can be avoided* by the optimization procedure (either by restarting it when $\mathcal{L} > 0$ or by exploiting the randomness of SGD).

In general, the critical points of the dynamical system in $\dot{\rho}$ and $\dot{V_k}$ are the critical points of the flow of ρ because of the following

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

For $\lambda = 0$, if the f_n satisfy the condition for $\dot{\rho} = 0$, they also satisfy the conditions for $\dot{V}_k = 0$. Thus for each critical point of ρ there is a critical point of the V_k. In principle, however, the critical points of V_k for a given ρ are not necessarily consistent with the required values of the $\overline{f}_n{}^2$ with the values of f_n required by the critical points of $\dot{\rho}$ (since the V_k determine the f_n). However, if we assume that normalization is "faster" than the dynamics of ρ, then the critical points are determined by the critical points of the ρ flow. A complementary viewpoint is that there many degenerate values of V_k that satisfy a set of $f_n, \forall n$. One can see how this may be considering the degenerate zero-loss minima, which have a largish dimension (in the order of $d - N$, as discussed in Appendix A)[3].

3.4.4 Landscape of the empirical risk

As shown in Appendix A the *landscape of the empirical loss contains a set of degenerate zero-loss global minima (for $\lambda = 0$) that under certain overparametrization assumptions are connected in a single zero-loss degenerate valley for $\rho \geq \rho_0$*. Figures 7 and 8 show a landscape which has a saddle for $\rho = 0$ and then goes to zero loss (water level) for different values of rho (look at the boundary of the mountain). As we will see, the descent from $\rho = 0$ can encounter local minima and saddles with non-zero loss. Furthermore, even though the valley of zero loss is connected (under certain assumptions), the absolute minimum ρ point may be unreachable by gradient flow from another point of zero loss even in the presence of $\lambda > 0$, because of the possible non-convex profile of the coastline (see Figure 9).

3.5 Toy Model: qualitative dynamics

Recall that $0 \leq \overline{f}_n \leq 1, \quad \forall n$ (assuming $||x|| \leq 1$, it holds $||f(x)|| \leq 1$ taking into account the definition of RELUs, the fact that matrix norms are sub-multiplicative, and $||V_k||$=1). Depending on the number of layers, the maximum margin that the network can achieve for a given dataset is usually much smaller than 1 because the weight matrices have unit norm and the bound ≤ 1 is conservative. 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. For instance, in the experiments plotted in this paper, the maximum $\overline{f}_n$ are around 0.01 and thus the ρ needed for interpolation (for $\lambda = 0$) is in the order of 100. Let us assume that for a given data set there is a maximum values of the $y_n f_n$ for which there is interpolation. Correspondingly, there is a minimum value of ρ that we call, as mentioned earlier, ρ_0.

We now provide some intuition for the dynamics of the toy model.

Let us start assuming that the network of Figure 1 is initialized with very small ρ, that is $\rho << \rho_0$. Assume then that at some time t, $\mu > 0$, that is *average separability* holds. Notice that if the f_n were zero-mean, random variables, there would be a 50% chance for average separability to hold. Then Equation 6 shows that $\dot{\rho} > 0$. Observation 1 suggests that $\dot{\rho}$, averaged over fluctuation due to normalization which we neglect here, is positive immediately after initialization and also before reaching ρ_0. In between, there may oscillations in ρ for short intervals, depending on the data, as discussed later. Thus ρ grows (non-monotonically) until it reaches an equilibrium value, close to ρ_0. Recall that for $\lambda = 0$ this corresponds to a degenaret global minimum $\mathcal{L} = 0$, resulting in a large attractive basin in the loss landscape (see Appendix A). For $\lambda = 0$, a zero value of the loss $\mathcal{L} = 0$ implies interpolation: thus all the f_n have the same value, that is all the margins are the same.

If we initialize a network with large norm $\rho > \rho_0$, Equation 3 shows that $\dot{\rho} < 0$. This implies that the norm of the network will decreases until an equilibrium is reached. However since $\rho \gg 1$, our key hypothesis implies that there exists an interpolating (or near interpolating) solution with ρ that is very

[2]We *assume* sufficient overparametrization Equation 7 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 normalization constraints for each k does not eliminate the ability to fit. In other words, though the second term in the equations for $\dot{V}_k$ restricts the V_k weight matrices to have certain properties (in a simple example to be orthogonal matrices), we assume their ability to fit is still intact, modulo the norm constraints. Also notice that the equations for $\dot{V}_k$ determine, from past values of ρ, V_k, f_n, the future values of V_k and thus of f_n, whereas the equation in $\dot{\rho}$ determines, from past values of f_n, the future ρ.

[3]It is interesting to look at the critical points of SGD with minibatch 1 (see Appendix D) then

$$V_k \overline{f}_n (1 - \rho \overline{f}_n) = \frac{\partial \overline{f}_n}{\partial V_k} (1 - \rho \overline{f}_n) \quad \forall n. \tag{13}$$

which can be rewritten as $(V_k \overline{f}_n - \frac{\partial \overline{f}_n}{\partial V_k})(1 - \rho \overline{f}_n) = 0$.

Notice that $V_k \overline{f}_n - \frac{\partial \overline{f}_n}{\partial V_k} = 0$ cannot be satisfied by matrices V_k such that $||V_k|| \neq 1$.

close to the initialization. In fact, for large ρ it is usually possible to find a set of weights V_L such that $\rho|f_n| \approx 1$. To understand why this may be, recall that if there are at least N units in the top layer of the network (layer L) with given activities and $\rho >> \rho_0$ there exist values of V_L that yield interpolation. In other words, it is easy for the network to interpolate with small values $\overline{f}_n$. These large ρ, small $\overline{f}_n$ solutions remind of 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 (corresponding to large margin solutions) may be found before large ρ quasi-interpolating solutions which have worse margin (See Fig. 2). This dynamics takes place *even in the absence of regularization* ($\lambda = 0$); however, as we will see, $\lambda > 0$ makes the process more robust.

Norm dynamics

To explain more in detail the initial dynamics of the gradient flow let us neglect here the dynamics of normalization; thus we assume $||V_k|| = 1$ at all times. This is tantamount to neglecting fluctuations due to normalization in the dynamics of V_k or to assuming that the normalization dynamics is fast (wrt to ρ dynamics)). Then

Observation 1

- *immediately after initialization with $\rho(0) << \rho_0$, ρ grows;*
- *ρ grows non-monotonocally until $\rho(t)$ is within a neighborhood of ρ_{eq} (within $\lambda\rho_0 + \sum f_n^2$);*
- *there may be oscillations in $\rho(t)$, that is time intervals in which $\dot{\rho} < 0$: these intervals are short;*
- *$\dot{\rho} = 0$ between positive and negative values may be a critical point of the system if the $\dot{V}_k = 0$ for the associated ρ give compatible f values; these critical points may be local minima or saddle points.*
- *If assume that local minima and saddles can be avoided (by SGD or by restarting optimization since $\mathcal{L} > 0$) the dynamics will eventually converge to an interpolation or quasi-interpolation solution with $\mathcal{L}$ close to zero.*

Notice that $\rho(t) = 0$ and $f(x) = 0$ (if all weights are zero) is in a critical unstable point. A small perturbation will either result in $\dot{\rho} < 0$ with ρ going back to zero or in ρ growing if the average margin is just positive, that is $\mu > \lambda\rho > 0$.

Lemma E shows that we can write the loss as

$$\mathcal{L} = 1 - \rho(\mu + \frac{\dot{\rho}}{2}). \tag{14}$$

Thus if $\dot{\rho}$ decreases the average margin μ must increase to avoid that $\mathcal{L}$ increases. In particular, this implies that $\dot{\rho}$ cannot be negative for long. Notice that short periods of decreasing ρ are "good" since they increase the average margin! If $\dot{\rho}$ turns negative it means it goes through $\dot{\rho} = 0$. This may be a critical point for the system if the values of V_k corresponding to $\dot{V}_k = 0$ are compatible. We assume that this critical point – either a local minimum or a saddle – can be avoided by the randomness of SGD or by an algorithm that "sees" the $\mathcal{L} > 0$ and then restarts optimization.

Other observations are that $\mathcal{L}$ decreases with μ increasing and σ decreasing. The figures show that in our experiments the large margins of some of the data points decrease during GD, contributing to a decrease in σ. Furthermore Equation **??** suggests that for small ρ, the term dominating the decrease in $\mathcal{L}$ is $-2\rho\mu$. For larger ρ, the term $\rho^2 M = \rho^2(\sigma^2 + \mu^2)$ becomes important: eventually $\mathcal{L}$ decreases because σ^2 decreases. The regularization term, for standard small values of λ is relevant only in the final phase, when ρ is in the order of ρ_0. For $\lambda = 0$ the loss at the global equilibrium (which happens at $\rho = \rho_0$) is $\mathcal{L} = 0$ (since $\mu = \frac{1}{\rho_0}$, $M = \mu^2$, $\sigma^2 = 0$).

In all these observations we have assumed gradient flow. The dynamics for gradient descent implies by itself the presence of damped oscillations in ρ[4]. In addition, the randomness of SGD also contributes to transient decreases in the norm ρ.

[4] As we will show later, discretization is similar to adding a second derivative term $\eta\ddot{\rho}$ to the dynamical system which introduces by itself oscillations in the dynamics of ρ.

3.5.1 Margins with and without weight decay

As we stated earlier, we *assume* sufficient overparametrization of the V_k to enable the network to interpolate the y_n labels for different sets of training data, provided ρ is large enough. More specifically *we assume that for $\rho \geq \rho_0$ the network can interpolate the data*, where ρ_0 is the minimum norm solution for $\lambda = 0$. This assumption (interpolation for a continuous of ρ values with $\rho < \rho_0$ makes sense because a linear transformation can be added to the top layer (after V_{L-1} and before V_L) to decrease the output $f(x) = \frac{1}{\rho_0}$ in a continuous way. The assumption is also consistent with the fact that there many degenerate values of the parameters for the global minimum (see Appendix A), which have a large dimension (in the order of $d - N$, as discussed in Appendix A). This assumption implies that if interpolation is possible when all margins $\overline{f}_n = \frac{1}{\rho_0}$, the margins can also take equal values $\overline{f}_n \leq \frac{1}{\rho_0} \quad \forall n$. In particular for $\lambda = 0$, together with Lemma , it follows that for each $\rho \geq \rho_0$, as it happens for large initializations, there should be a critical point of GD corresponding to an interpolatory solution with worse margin than the minimum norm (ρ_0) solution.

Square loss optimization yielding interpolation of labels implies that all margins are the same ($\sigma^2 = \sum_n (\overline{f}_n - \sum \overline{f}_n)^2) = 0$). What may happen however when there is no interpolation and therefore no requirement for σ to be zero? In particular, what may happen when $\lambda > 0$? Remember that optimization should prefer the solution which has the smaller $\mathcal{L}$.

We start by considering and computing $\mathcal{L}$ in the "trivial"case: for $\lambda > 0$ the network reaches the same maximum joint margins $\overline{f}_n = \frac{1}{\rho_0} \quad \forall n$ as before (it is the same network). Another interesting possibility is that *the network reaches margin $f_n = \frac{1}{\rho_0}$ for M points and higher margin for the remaining $N - M$ points allowing them to interpolate.* There are of course many other situations with $\sigma > 0$ but we focus on this one because configurations with interpolating points plus other points with smaller ρ have smaller $\mathcal{L}$ than other configurations (for instance consider a configuration in which $N - M$ points can reach margins larger than $\frac{1}{\rho_0}$, but the remaining M points have lower margins than $\frac{1}{\rho_0}$ resulting in a higher square loss).

It turns out that the latter configuration, depending on parameters values, may have a smaller loss than the configuration with all equal margins. The difference in loss goes to zero for large N and small $\lambda\rho_0^2$. The following

Lemma 2 *For $\lambda > 0$ the network that achieves $\frac{1}{\rho_0}$ margin for all points, yields a solution with norm $\rho_\lambda = \frac{\rho_0}{1+\rho_0^2 \frac{\lambda}{N}}$ that corresponds to quasi-interpolation with a gap to interpolation which is $\epsilon = 1 - \rho_0 \frac{1}{\rho_\lambda} = \lambda\rho_0$. If there exists a set of weights providing interpolation for all but M of the data, the margins will not be all equal but will still be within $\lambda \frac{\rho_0}{M}$ of each other. This second situation corresponds to smaller $\mathcal{L}$ and should be selected by GD if feasible.*

Notice that the gap to interpolation ϵ of the Lemma is the same ϵ relevant for Neural Collapse (see later assumption 1). If, in this binary case, we consider margins separately for the $+$ and $-$ one-hot encodings, as in the multiclass case (see later), then $1 > f^+(x_+) > 1 - \epsilon$ and $\epsilon > f^+(x_-) > 0$, with $\epsilon = \lambda \frac{\rho_0}{}$ and fluctuations around it (for individual data points) of the order of $\delta\epsilon = \lambda \frac{\rho_0}{M}$. The proof of the Lemma is in Appendix F.

3.5.2 Remarks

- *Role of Weight Decay*

  Equation 11 shows that weight decay performs the traditional role of promoting solutions with small norm. In the case of large initialization, we can see from (11) 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 13 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[5]. 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

[5]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.

later, by preventing exact interpolation, gradient flow with weight decay under the square loss shows at convergence the phenomenon of Neural Collapse.

- *Un-normalized vs normalized dynamics*

  As shown in the Appendix G, 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

$$\dot{\rho} = 2L\rho^{\frac{2L-2}{L}}[\sum_n \overline{f}_n - \sum_n \rho(\overline{f}_n)^2] \tag{15}$$

  and

$$\dot{V_k} = -2\rho^{\frac{L-2}{L}} \sum_n (1 - \rho\overline{f}_n)(\frac{\partial \overline{f}_n}{\partial V_k} - V_k \overline{f}_n) \tag{16}$$

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

$$\dot{\rho} = 2[\sum_n f_n y_n - \sum_n \rho(f_n)^2] \tag{17}$$

  and

$$\dot{V_k} = 2\rho \sum_n [(1 - \rho\overline{f}_n)(V_k \overline{f}_n - \frac{\partial \overline{f}_n}{\partial V_k})]. \tag{18}$$

  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 both cases but with a smaller factor in the un-normalized case wrt the normalized case. The proportionality 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]:

$$|L(g) - \hat{L}(g)| \leq c_1 \mathbb{R}_N(\mathbb{G}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}} \tag{19}$$

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,

$$\mathbb{R}_N(\mathbb{G}) = \rho \mathbb{R}_N(\mathbb{F}), \tag{20}$$

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 19 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

$$L_0(f^a) \leq A\rho_a + \epsilon \tag{21}$$

and

$$L_0(f^b) \leq A\rho_b + \epsilon \tag{22}$$

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.

Notice that the this generalization claim is just a *relative* claim. Direct bounds on generalization based on minimum margin (see [32]) are typically loose, though it may be possible to obtain stronger bounds for the deep network we analyze by using properties of the margin distribution such as our σ and μ (following an idea in [34]).

5 Extending the toy model to GD

Work in [35] 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 [36]) 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 6 and 7 become (taking into account that $\mathcal{L} = \sum_n (1 - \rho\overline{f}_n)^2 + \nu \sum_{k=1}^{L-1} ||V_k||^2 + \lambda\rho^2$)

$$\frac{\eta}{2}\ddot{\rho} + \dot{\rho} = 2[\sum_n (1 - \rho\overline{f}_n)\overline{f}_n] - 2\lambda\rho \tag{23}$$

$$\frac{\eta}{2}\ddot{V_k} + \dot{V_k} = 2\rho \sum_n [(1 - \rho\overline{f}_n)(V_k\overline{f}_n - \frac{\partial \overline{f}_n}{\partial V_k})] \quad \forall k \leq L. \tag{24}$$

By using $W_L = \rho V_L$ (see figure 1), we replace this system of equations with the following system

$$\frac{\eta}{2}\ddot{V_k} + \dot{V_k} = 2\rho \sum_n [(1 - \rho\overline{f}_n)(V_k\overline{f}_n - \frac{\partial \overline{f}_n}{\partial V_k})] \quad \forall k < L. \tag{25}$$

$$\frac{\eta}{2}\ddot{W_L} + \dot{W_L} = 2\sum_n [(1 - \overline{g}_n)\frac{\partial \overline{g}_n}{\partial W_L})] - 2\lambda W_L. \tag{26}$$

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

$$\frac{\eta}{2}W^T\ddot{W}_L + W^T\dot{W}_L = 2\sum_n(1-\overline{g}_n)\overline{g}_n - 2\lambda W_L^2 \tag{27}$$

We change variables in the last equation:

$$\frac{\eta}{2}W^T\ddot{W}_L + W^T\dot{W}_L = 2\rho\sum_n[(1-\rho\overline{f}_n)\overline{f}_n) - 2\lambda\rho^2 \tag{28}$$

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 $\dot{\rho} = 0$

$$\rho V^T\frac{\eta}{2}\rho\ddot{V}_L + \rho V^T\rho\dot{V}_L = 2\rho\sum_n[(1-\rho\overline{f}_n)\overline{f}_n) - 2\lambda\rho^2 \tag{29}$$

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

$$\rho^2V^T\frac{\eta}{2}\ddot{V}_L = 2\rho\sum_n((1-\rho\overline{f}_n)\overline{f}_n - 2\lambda\rho^2 \tag{30}$$

$$-\frac{\eta}{2}\rho||\dot{V}_L||^2 = 2\sum_n(1-\rho\overline{f}_n)\overline{f}_n - 2\lambda\rho \tag{31}$$

Equation 23 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 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 $f_W : \mathbb{R}^d \to \mathbb{R}^C$, $f_W(x) = W_L\sigma(W_{L-1}\dots W_2\sigma(W_1x)\dots)$ 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 now a key assumption here for the multiclass case, that we conjecture should follow from our analysis of the dynamics. In fact, for binary classification the assumption is Lemma 2 that we proved earlier (the ϵ appearing below is $\epsilon = \lambda\rho_0$).

The assumption is 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 $f^{(c')}(x_{n(c)}) = \frac{\epsilon}{C-1}$ where ϵ is the interpolation gap.*

We observe that lemma 2 shows that a deep network trained with GD under the square loss satisfies the assumption 1 for the binary case. This brings us to the main result of this section:

Theorem 3 *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:

$$\begin{aligned}\frac{\partial W_L^c}{\partial t} &= \sum_n (y_n^c - \langle W_L^c, h(x_n)\rangle)h(x_n) - \lambda W_L^c \\ &= \sum_{n\in N(c)} (1 - \langle W_L^c, h(x_{n(c)})\rangle)h(x_{n(c)}) + \sum_{n\in N(c'), c'\neq c} (-\langle W_L^c, h(x_{n(c')})\rangle)h(x_{n(c')}) - \lambda W_L^c\end{aligned} \tag{32}$$

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:

$$\frac{\partial W_L^c}{\partial t} = 0 \quad \implies W_L^c = \frac{CN\epsilon}{\lambda(C-1)} \times (\mu_c - \mu_G) \tag{33}$$

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:

$$\begin{aligned}&\left\langle W_L^c, \frac{\partial W_L^c}{\partial t}\right\rangle = \sum_n (y_n^c - f_W^{(c)}(x_n))f_W^{(c)}(x_n) - \lambda||W_L^c||_2^2 = 0 \\ &\implies ||W_L^c||_2^2 = \frac{N}{\lambda}\left(\epsilon - \frac{C}{C-1}\epsilon^2\right)\end{aligned} \tag{34}$$

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 (76)

$$\begin{aligned}\langle W_L^c, \mu_G\rangle &= 1 - \epsilon - \frac{\lambda(C-1)}{CN\epsilon}||W_L^c||^2 \\ &= 1 - \epsilon - \frac{\lambda(C-1)}{CN\epsilon} \times \frac{N}{\lambda}\left(\epsilon - \frac{C}{C-1}\epsilon^2\right) = \frac{1}{C}.\end{aligned} \tag{35}$$

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 (77) yields

$$\begin{aligned}&\frac{\lambda(C-1)}{CN\epsilon} \times \langle W_L^c, W_L^{c'}\rangle = \frac{\epsilon}{C-1} - \frac{1}{C} \\ &\implies \langle V_L^c, V_L^{c'}\rangle = \frac{1}{||W_L^c||_2^2} \times \frac{CN\epsilon}{\lambda(C-1)} \times \left(\frac{\epsilon}{C-1} - \frac{1}{C}\right) = -\frac{1}{C-1}\end{aligned} \tag{36}$$

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. (77) shows that the norms of the classifier weights are given by $||W_L^c||_2^2 = \frac{N\epsilon}{\lambda}\eta$. As we mentioned in lemma 2, for a non-zero value of λ we expect some small interpolation gap ϵ. In the binary case this is given by $\epsilon = \lambda\rho_0$.

Plugging this relationship into Eq. (77) 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 in the paper and an this section about Neural Collapse.

- For binary classification section 2 predicts that the square loss dynamics yields without further assumptions to the binary version of Assumption 2, thus predicting Neural Collapse not only for the case of cross-entropy, for which it was empirically found, but also for the square loss.
- The analysis of the loss landscape and of the qualitative dynamics under the square loss in section 3.5 and in section A implies that all quasi-interpolating solutions with $\rho \geq \rho_0$ and $\lambda > 0$ yield neural collapse and have its four properties, including the ETF property.
- 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 – that turns out to be necessary – 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.

7 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 [37]. 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, **??** and 6 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. **??** 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. 6 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].

8 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 prediction of Neural Collapse for randomly labeled CIFAR10, has been confirmed in preliminary experiments by our collaborators. Furthermore, one of the predictions of our analysis is that all quasi-interpolating solutions with $\rho \geq \rho_0$ will show Neural Collapse and show its properties such as the ETF property.

Independently of Neural Collapse, can our analysis of the square loss dynamics and its connection with margin, 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, 38, 39]. Margin in fact provides an upper bound on generalization error, as shown in section 4. Larger margin gives a better upper bound on the generalization error for the same network trained on the same data. We have verified empirically this property by varying the margin using different degrees of random labels in a binary classification task (see Appendix). 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 supplementary material. However, the simple upper bound given in section 4 does not explain details of the generalization behavior that we observe for different initializations (see figures 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 $\lambda > 0$: there can be generalization without weight decay, depending on the trajectory of the dynamics and thus also on initialization. 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, since a small norm can follow

from small initial conditions without regularization.

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 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.

In this sense, the bias towards small ρ solutions induced by regularization for $\lambda > 0$ can be replaced by an implicit bias induced by small initialization and parameters values that allow convergence to the first quasi-interpolating solution for increasing ρ.

In summary, this paper shows under which conditions gradient descent on the square loss can converge to minimum ρ solutions, corresponding to max margin solutions. Associated with those solutions are upper bounds on the generalization error and properties such as Neural Collapse. Interestingly, these results do not say why deep networks should be better than other classifiers such as kernel machines. We believe that the answer to this question is in approximation properties[40] of a certain class of deep networks and not in any implicit bias of the optimization process. In particular, CNN-like networks, in general without weight sharing, enjoy good approximation properties for a class of functions that can be represented in a compositional form. Furthermore, unpublished results suggest that deep compositional architectures with skip connections also have remarkable optimization properties for a broad subset of the same class of compositional functions.

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 proved that the dynamics of gradient flow implies a symmetric level of near interpolation for small λ in the binary case but not in the multiclass case. We also showed that that gradient flow on a network initialized with small norm will converge to a global minimum which may have close to minimum norm. Our analysis says that, with L2 regularization, the critical points of gradient flow 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) should be 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).

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] Shen-Huan Lv, Lu Wang, and Zhi-Hua Zhou. Optimal margin distribution network. *CoRR*, abs/1812.10761, 2018.

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

[36] 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.

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

[38] 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.

[39] 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.

[40] H.N. Mhaskar and T. Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, pages 829– 848, 2016.

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

[42] 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.

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

[44] 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.

A Landscape of the loss and degeneracy of global minima

We are interested in the global minimizers achieving zero loss of

$$L(f(W)) = \sum_{i=1}^{n} \ell_i^2 \tag{37}$$

with $\ell_i = y_i - f(W; x_i)$. The network f is assumed to be overparametrized with a number of weights $d >> n$ and to be able to interpolate the training data achieving $L(f(W^*)) = 0$ which implies $\ell_i = 0 \quad \forall i = 1, \cdots, N$.

If we assume overparametrized networks with $d >> n$, where d is the number of parameters and n is the number of data points Cooper (see references in [41] proved that the global minima of $L(w)$ are highly degenerate[6] with dimension $d - n$.

Theorem 4 *We assume an overparametrized deep network f with smooth RELU activation functions and square loss. Then the minimizers $W*$ achieve zero loss and are highly degenerate with dimension $d - n$.*

Under certain conditions all the global minima – associated to interpolating solutions – will be connected within a unique, large valley. The argument is based on Theorem 5.1 of [42] :

Theorem 5 (Connected valleys of the loss [42]) *If the first layer of the network has at least $2N$ neurons, where N is the number of training data and if the number of neurons in each subsequent layer decreases, then every sublevel set of the loss is connected.*

In particular, this implies that *zero-square-loss minima with different ρ are connected*. A connected single valley of zero loss *does not* however guarantee that *SGD with WD will converge to the global minimum which is now > 0, independently of initial conditions*. The reason is that the connected valley will in general twist in the space of parameters in such a way that following it does not monotonically increase or decrease ρ.

We expect that for large ρ (with $\lambda = 0$), there are many solutions. The existence of several solutions for large ρ is based on the following intuition: the last linear layer is enough – if the layer before the linear classifier has more units than the number of training points – to provide solutions for a given set of random weights in the previous layers (for large ρ and small f_i). This also means that the intermediate layers 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 ρ smaller than a certain minimum ρ, which is network and data-dependent. With increasing ρ from $\rho = 0$ there will be a continuous set of zero square-loss degenerate minima with the minimizer representing an interpolating (for $\lambda = 0$) or almost interpolating solution (for $\lambda > 0$). We expect that for $\lambda > 0$ there is a "pull" towards the minimum ρ_0 within the local degenerate minimum of the loss.

B Structural Lemma

For Deep ReLU networks we also have the following structural property of the gradient

Lemma 6 (*Lemma 2.1 of [43]*) $\sum_{i,j} \frac{\partial f_W(x)}{\partial W_k^{i,j}} W_k^{i,j}$ *which can be rewritten as* $\left\langle W_k, \frac{\partial f_W(x)}{\partial W_k} \right\rangle = f_W(x)$.

Here W_k refers to the weight matrix of the k^{th} layer of the network; vec(W_k) is its vectorized form. We use W_k for both, trusting that the context disambiguates them.

C Critical points

Proof of Lemma 3.5.1

Since the V_k are bounded functions they must take their maximum and minimum values on their compact domain – the sphere, because of the extremum value theorem.

[6] This result is also what one expects from Bezout theorem for a deep polynomial network. As Terry Tao says in his blog "from the general "soft" theory of algebraic geometry, we know that the algebraic set V is a union of finitely many algebraic varieties, each of dimension at least d-n, with none of these components contained in any other. In particular, in the underdetermined case n<d , there are no zero-dimensional components of V , and thus V is either empty or infinite".

D Critical points of SGD

Consider the case of $\lambda = 0$.

$$\min_W L(f(W)) = \min_W \sum_{i=1}^{N} \ell_i^2 \tag{38}$$

with $\ell_i = y_i - f(W; x_i)$.

We minimize $L(f(W))$ by running the following dynamical system (e.g. gradient flow)

$$\dot{W} = \nabla_W L(f(W)) = \sum_i^N \nabla_W f(W; x_i)(y_i - f(W; x_i)). \tag{39}$$

SGD can be formulated as follows. First define

Definition 7 *A random vector $v \in R^d$ drawn from a distribution $\mathcal{D}$ is a sampling vector if $\mathcal{E}_{\mathcal{D}}[v_i] = 1 \quad \forall i$*

Then the stochastic version of Equation 38 is

$$\min_W \mathcal{E}_{\mathcal{D}}[L(f(W))] = \min_W \mathcal{E}_{\mathcal{D}} \sum_i^n v_i \ell_i \tag{40}$$

Usually the distribution over $\mathcal{D}$ is assumed to be random v with independent components v_i, satisfying condition 7. The literature gives the impression that in expectation SGD is equal to GD, which is true *if* ℓ_i^2 are quadratic functions.

Finding the interpolating global minimizers of $L = \sum \ell_i^2$ is equivalent to finding the set of network weights W^* that solve the system of equations $\ell_i(W^*) = 0 \quad \forall i = 1, \cdots, N$. Thus instead of finding all the critical points of the gradient of L, we would like to find the joint minimizers – that is the W – that minimize $\ell_i^2 \quad \forall i = 1, \cdots, n$.

We define *critical points of SGD* the solutions of $\ell_i \nabla \ell_i = 0, \forall i$. For a discussion see [41].

A critical point of SGD with minibatch of size 1 is defined as $\dot{z} = g(x_n) = 0 \forall n = 1, \cdots, N$. This compares with a critical point of GD defined as $\dot{z} = \sum_{n=1}^{N} g(x_n) = 0$. GD of course is SGD with minibatch size N. What about SGD with intermediate minibatch sizes? The answer is[7]

Lemma 8 *If $\dot{z} = \sum_{n=1}^{M} g(x_n) = 0$ for enough random SGD draws of size $M < N$ then $\dot{z} = g(x_n) = 0 \forall n = 1, \cdots, N$.*

E Lemma on ρ dynamics

Lemma 9 *Assume $\rho \sum \overline{f_n} < 1$ and a normalized network, that is $||V_k|| = 1, \quad k = 1, \cdots, L$. Then the loss can be written as $\mathcal{L} = 1 - \rho(\frac{1}{2}\dot{\rho} + \mu)$.*

Proof

Consider the loss $\mathcal{L} = 1 - 2\rho\mu + \rho^2 M + \lambda\rho^2$. On the other hand $\dot{\rho} = 2\mu - 2\rho M - 2\lambda\rho$ which gives $2\rho M = 2\mu - 2\lambda\rho - \dot{\rho}$

Thus we obtain the result

$$\mathcal{L} = 1 - \frac{1}{2}\rho\dot{\rho} - \rho\mu. \tag{41}$$

[7] A caution here is necessary: the number of random draws of size M from a data set of size N is enormous since it is equal to $\binom{N}{M}$ and thus, though all of them provide an overconstrained set of equations, only a very small subset of meaningful constraints may be available in practice.

F Margins at convergence

Proof of Lemma 2.

Consider first a global minimum with $\lambda = 0$ and $\mathcal{L} = 0$. Such a global minimum corresponds to interpolation of all data points and to $\dot{\rho} = 0$. Thus $\rho_0 f_n = y_n = \pm 1$ and the margins for all n are the same. We assume that this is the minimum ρ with smallest possible norm that we call ρ_0. This also implies that there is no solution such that the f_n are *all equally larger* than $\frac{1}{\rho}$.

Let us now assume that $\lambda > 0$. The margins can still attain the same values as in the case of $\lambda = 0$, that is $\overline{f}_n = \frac{1}{\rho_0}, \quad \forall n$. Recall that by assumption $\frac{1}{\rho_0}$ is the maximum margin that all f_n can attain together (since they correspond to the minimum with minimum norm). Consider two extreme situations:

1. We assume maximum margins $\overline{f}_n = \frac{1}{\rho_0} \quad \forall n$ for a ρ at equilibrium smaller than ρ_0. The margins are $\overline{f}_n = \mu = \frac{1}{\rho_0}, \quad \forall n, \sigma = 0$ and thus $M = \mu^2 = \frac{1}{\rho_0^2}$. At equilibrium $\dot{\rho} = 0 = 2N\mu - 2N\rho M - 2\lambda\rho$ gives $\rho_\lambda = \frac{\rho_0}{1+\rho_0^2\frac{\lambda}{N}}$. The loss is

$$\mathcal{L} = N(\frac{\rho_0^2\frac{\lambda}{N}}{1+\rho_0^2\frac{\lambda}{N}})^2 + \lambda(\frac{\rho_0}{1+\rho_0^2\frac{\lambda}{N}})^2 = \frac{\lambda\rho_0^2}{1+\frac{\lambda}{N}\rho_0^2} \tag{42}$$

   All the margins are the same and equal to their average μ. Standard deviation is zero.

2. All the $\overline{f}_n$ increase to interpolate with the exception of M of them. Notice that it is better, in terms of achieving the smallest loss, to decrease a margin rather than increase it, since it is preferable to have $\rho\overline{f}_j = 1 - \epsilon$ rather than $\rho\overline{f}_j = 1 + \epsilon$ (because of penalty induced by the regularization term). Thus in the case $M = 1$ assume $\overline{f}_1$ remains $\overline{f}_1 = \frac{1}{\rho_0}$ that is its maximum value for $\lambda = 0$), while the other margins increase to allow interpolation. In this case the minimum of $\mathcal{L} = \sum_n (1 - \rho\overline{f}_n)^2 + \lambda\rho^2$ is for $\rho_\lambda = \frac{\rho_0}{1+\lambda\rho_0^2}$. Here the loss becomes $\mathcal{L} = (1 - \frac{1}{1+\lambda\rho_0^2})^2 + \lambda\rho^2$, that is $\mathcal{L} = \frac{(1+\lambda\rho_0^2)(\lambda\rho_0^2)}{(1+\lambda\rho_0^2)^2} = \frac{\lambda\rho_0^2}{1+\lambda\rho_0^2}$.

   The margins $\overline{f}_n, \quad \forall n \neq 1$ are now larger than $\overline{f}_1$, since we assume they interpolate: in fact $\overline{f}_n = \frac{1+\lambda\rho_0^2}{\rho_0}, \quad \forall n > 1$ which are larger than $\overline{f}_1 = \frac{1}{\rho_0}$. The difference between $\overline{f}_1$ and the other margins $\overline{f}_n$ is $\Delta = \lambda\rho_0$; it is controlled by λ. If instead of $N - 1$ interpolating points there are $N - M$, the minimum of $\mathcal{L}$ will be for $\rho = \frac{M\rho_0}{M+\lambda\rho_0^2}$, which is larger than in the case of $M = 1$. Thus the latter is the worse possible case with respect to the largest variance in margin between the data points. With M margins being unable to interpolate (instead on one) $\rho_{\lambda,M} = \frac{\rho_0}{1+\lambda\frac{\rho_0^2}{M}}$ and

$$\mathcal{L} = \frac{\lambda\rho_0^2}{1+\lambda\frac{\rho_0^2}{M}}. \tag{43}$$

In this case, the difference between the margins is $\Delta = \lambda\frac{\rho_0}{M}$.

For small λ the loss in the case of all equal margins (the first case above) is close to the smaller loss of the second case. This is not surprising, because the network is effectively providing a μ larger than $\frac{1}{\rho_0}$.

Additional Remarks

- *SGD and convergence*

  For SGD with minibatch of size $= 1$[8] the flow in ρ and V_k is

$$\dot{\rho} = -2(\rho f_n - y_n)f_n - 2\lambda\rho \quad \forall n \tag{44}$$

  and

$$\dot{V_k} = -2(\rho f_n - y_n)\rho\frac{\partial f_n}{\partial V_k} + 2V_k\sum_n(\rho^2 f_n^2 - \rho y_n f_n) \quad \forall k = 1, \cdots, L-1. \tag{45}$$

[8] With high probability this should hold also for minibatch sizes larger than 1 but smaller than size of the training set N.

- A theorem on continuous dependence of solutions of ODE on their parameters (this theorem is usually applied to gradient flows)can be applied to how solutions depend continuously on λ.
- Notice that for normalized V_k, $V_k^T \dot{V}_k = 0$ always, that is for normalized V_k the change in V_k is always orthogonal to V_k, that is V_k can only rotate. If in addition $\sum_n \ell_n \frac{\partial f_n}{\partial V_k} - V_k f_n = 0$, then $\dot{V}_k = 0$.

G Unnormalized dynamics

We consider the dynamical system induced by GD on a deep net with RELUs (see Figure 1 A). We change variables by using [9] $W_k = \rho_k V_k$, $||V_k|| = 1$. Following the calculations in [6], the following identities hold: $\frac{\partial \rho_k}{\partial W_k} = V_k^T$ and $\frac{\partial g_n}{\partial W_k} = \frac{\rho}{\rho_k} \frac{\partial f_n}{\partial V_k}$.

Thus gradient descent on $L = \mathcal{L} = \sum_n (\rho f_n - y_n)^2$ with the definitions of V_k and ρ_k yields the dynamical system (with $\dot{W}_k = -\frac{\partial L}{\partial W_k}$)

$$\dot{\rho_k} = \frac{\partial \rho_k}{\partial W_k} \dot{W}_k = V_k^T \dot{W}_k = -2 \sum_n (\rho_k^L f_n - y_n) f_n \rho_k^{L-1} = -2\rho_k^{L-1} [\sum_n \rho_k^L (f_n)^2 - \sum_n f_n y_n] \tag{46}$$

and, with $S_k = I - V_k V_k^T$,

$$\dot{V}_k = \frac{\partial V_k}{\partial W_k} \dot{W}_k = \frac{S_k}{\rho_k} \dot{W}_k = -2 \frac{\rho}{\rho_k^2} \sum_n (\rho f_n - y_n)(\frac{\partial f_n}{\partial V_k} - V_k f_n). \tag{47}$$

G.1 Equal growth

If we assume that all the ρ_k are the same at initialization, we can use the following lemma to show that all ρ_k are the same at all times :

Lemma 10 [6] $\frac{\partial \rho_k^2}{\partial t}$ *is independent of* k *for* $\lambda = 0$ *and no normalization.*

Proof
Start from $\mathcal{L} = \frac{1}{N} \sum_n (1 - \rho \overline{f}_n)^2 + \lambda \sum_k ||W_k||^2$ with $\lambda = 0$. Then

$$\dot{W}_k = -\frac{2}{N} \sum_n (1 - \rho \overline{f}_n) \rho \frac{\partial \overline{f}_n}{\partial W_k} \tag{48}$$

and, since $||\dot{W_k}||^2 == W_k \dot{W}_k$, we write

$$||\dot{W_k}||^2 = -\frac{2}{N} \sum_n (1 - \rho \overline{f}_n) \rho \overline{f}_n \tag{49}$$

which is independent of k.

Notice that if $\lambda > 0$, then for weight decay applied to all layers (that is with a regularization term of the form $\lambda \sum_k \rho_k^2$)

$$||\dot{W_k}||^2 = -\frac{2}{N} \sum_n (1 - \rho \overline{f}_n) \rho \overline{f}_n - \lambda W_k \tag{50}$$

Thus the equal growth lemma 10 does not hold in the case of weight decay. Furthermore it does not hold in the case that the regularization term is $\lambda \rho^2$ with $\rho = \Pi_k \rho_k$.

[9] Changing coordinates from W_k to V_k, we can convert the previous dynamical system to one in ρ, V_k. Using some basic vector calculus to get $\frac{\partial \rho_k}{\partial t} = \frac{1}{\rho_k} \left\langle W_k, \frac{\partial W_k}{\partial t} \right\rangle$ and $\frac{\partial V_k}{\partial t} = \frac{1}{\rho_k} \left(I - V_k V_k^\top \right) \frac{\partial W_k}{\partial t}$.

G.2 Equal weight norms at all layers

Assume the setup of the previous section. Then $\rho = \rho_k^L$, where L is the number of layers.

We use Equation 46 to derive the dynamics of $\rho = \rho_k^L$ in terms of $\dot{\rho} = \sum_k \frac{\partial \rho}{\partial \rho_k} \dot{\rho_k}$

Thus

$$\dot{\rho} = 2L\rho^{\frac{2L-2}{L}} [\sum_n f_n y_n - \sum_n \rho (f_n)^2] \tag{51}$$

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 simplified simulations of Figure 10. The equilibrium value for $\dot{\rho_k} = 0$ is

$$\rho_0 = \frac{\sum_n \overline{f}_n}{\sum_n f_n^2}. \tag{52}$$

Similarly for V_k:

$$\dot{V_k} = -2\rho^{\frac{L-2}{L}} \sum_n (\rho f_n - y_n)(\frac{\partial f_n}{\partial V_k} - V_k f_n) \tag{53}$$

At equilibrium for V_k – that is when $\dot{V_k} = 0$ – the equation gives (with $\ell_n = \rho f_n - y_n$ and assuming $\sum f_n \ell_n \neq 0$)

$$\sum_n (\rho f_n - y_n) \frac{\partial f_n}{\partial V_k} = \sum_n (\rho f_n - y_n)(V_k^0 f_n). \tag{54}$$

H BN, GD, LM: remarks

H.1 Remarks on BN

Without BN and without WD $\frac{\partial g_n(W)}{\partial W_k} = \frac{\rho}{\rho_k} \frac{\partial f_n(V)}{\partial V_k}$; with BN but without weight decay this becomes $\frac{\partial g_n(W)}{\partial W_k} = \rho \frac{\partial f_n(V)}{\partial V_k}, \quad \forall k < L$ and $\frac{\partial g_n(W)}{\partial W_L} = \frac{\partial f_n(V)}{\partial V_k}$.

This dynamics 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 [31].

H.2 Lagrange multiplier vs Batch Normalization

The constrained Lagrange dynamics will not change the initial norm of the $L-1$ layers: to ensure that $||V_k|| = 1$ the initial value of the $L-1$ weight matrices must be $||V_k|| = 1, \quad k = 1, \cdots, L-1$.

In our toy model the dynamics above with Lagrange multipliers captures the key normalization property of batch normalization, though not all of its details (see Appendix and discussions in [6] and also [31]). Thus *we assume that for network trained with BN*, following the spirit of the analysis of [31], $\rho_k = 1, \quad \forall k < L$ 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 $L-1$ weight matrices but also to normalization of each row of the matrices [6] because it normalizes separately the activity of each unit i and thus – indirectly – the $W_{i,j}$ for each i separately. This implies that each row i in $(V_k)_{i,j}$ is normalized independently and thus the whole matrix V_k is normalized (assuming the normalization of each row is the same 1 for all rows). 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 $\nu_k = -\sum_n (\rho^2 f_n^2 - \rho y_n f_n)$.

Notice also that in the toy model of Figure 1B, we regularize a single ρ at the top of the network, whereas in the standard usage of BN each layer norm $\rho_k, \quad \forall k = 1, \cdots, L$ is subject to weight decay. All layers $k = 1, \cdots, L-1$, but the last one, are also subject to BN.

H.3 Commonly used normalization (Figure 1A)

In our toy model we considered the situation of 1B, where only the top layer is not normalized and has ρ at the top, subject to weight decay, while the previous layers are all normalized but are not subject

to weight decay. This model neatly separates layers that are normalized from layers that have weight decay; in this model no layer has weight decay *and* normalization.

However in normal practice of training a deep network, the weights W_k in all layers up to layer $L-1$ are subject to weight decay and normalization via batch norm; only the last layer weights W_L are not normalized but still subject to weight decay. In this section we consider this case, using Lagrange multipliers in place of batch norm.

The usual training of a deep net as in Figure 1A corresponds to minimizing the functional

$$\mathcal{L} = \sum_n (1 - \rho \overline{f}_n)^2 + \sum_{k=1}^{L-1} \nu_k \rho_k^2 (||V_k||^2 - 1) + \mu(||V_L||^2 - 1) + \lambda \sum_{k=1}^{L} \rho_k^2 \tag{55}$$

with $\rho_k^2 ||V_k||^2 = 1$ for $k = 1, \cdots, L$ with the definition $\rho = \Pi \rho_k$. Notice that Equation 55 is different from Equation 1 for the toy model.

H.3.1 Gradient flow

Gradient flow in this model is

$$\dot{\rho_k} = -\frac{\partial \mathcal{L}}{\partial \rho_k} = 2\frac{\rho}{\rho_k} \sum_n (1 - \rho \overline{f}_n)\overline{f}_n - 2\nu_k \rho_k ||V_k||^2 - 2\lambda \rho_k \quad \forall k = 1, \cdots, L-1 \tag{56}$$

$$\dot{\rho_L} = -\frac{\partial \mathcal{L}}{\partial \rho_L} = 2\frac{\rho}{\rho_L} \sum_n (1 - \rho \overline{f}_n)\overline{f}_n - 2\lambda \rho_L \tag{57}$$

and for $k < L$

$$\dot{V_k} = -\frac{\partial L}{\partial V_k} = 2\rho \sum_n (1 - \rho \overline{f}_n) \frac{\partial \overline{f}_n}{\partial V_k} - 2\nu \rho_k^2 V_k \tag{58}$$

noindent and

$$\dot{V_L} = -\frac{\partial L}{\partial V_L} = 2\rho \sum_n (1 - \rho \overline{f}_n) \frac{\partial \overline{f}_n}{\partial V_L} - 2\mu V_L \tag{59}$$

To find μ we multiply by V_L^T to get $0 = 2\rho \sum_n (1 - \rho \overline{f}_n)\overline{f}_n - 2\mu ||V_L||^2$

which gives

$$\mu = \frac{\rho}{||V_L||^2} \sum_n (1 - \rho \overline{f}_n)\overline{f}_n. \tag{60}$$

To find ν_k we multiply by V_k^T and obtain $0 = \rho \sum_n (1 - \rho \overline{f}_n)\overline{f}_n - \nu_k ||V_k||^2 \rho_k^2$ which gives

$$\nu_k = \frac{\rho}{||V_k||^2 \rho_k^2} \sum_n (1 - \rho \overline{f}_n)\overline{f}_n. \tag{61}$$

Thus the gradient flow is the following dynamical system for $k < L$

$$\dot{\rho_k} = -2\lambda \rho_k \quad \forall k = 1, \cdots, L-1 \tag{62}$$

$$\dot{V_k} = 2\rho \sum_n (1 - \rho \overline{f}_n)(\frac{\partial \overline{f}_n}{\partial V_k} - V_k \overline{f}_n) \quad \forall k = 1, \cdots, L-1 \tag{63}$$

$$\dot{\rho_L} = 2\frac{\rho}{\rho_L} \sum_n (1 - \rho \overline{f}_n)\overline{f}_n - 2\lambda \rho_L \tag{64}$$

$$\dot{V_L} = 2\rho \sum_n (1 - \rho \overline{f}_n)(\frac{\partial \overline{f}_n}{\partial V_L} - V_L f_n) \tag{65}$$

The solution of Equation 62 is $\rho(t) = 1 - (1 - \rho_{t=0})e^{-2\lambda t}$ because $\rho^2 = 1$ for $t \to \infty$.

An equivalent alternative and simpler formulation is to start from

$$\mathcal{L} = \sum_n (1 - \overline{g}_n)^2 + \sum_{k=1}^{L-1} \nu_k ||W_k||^2 + \lambda \sum_{k=1}^{L} ||W_k||^2 \tag{66}$$

under the constraint $||W_k||^2 = 1$.

Gradient flow in this model is

$$\dot{W}_k = 2\sum_n (1-\overline{g}_n)\frac{\partial \overline{g}_n}{\partial W_k} - 2\nu_k W_k - 2\lambda W_k \quad \forall k = 1, \cdots, L-1 \tag{67}$$

and

$$\dot{W}_L = 2\sum_n (1-\overline{g}_n)\frac{\partial \overline{g}_n}{\partial W_L} - 2\lambda W_L \tag{68}$$

To find ν_k we multiply by W_k^T and obtain $0 = \sum_n (1-\rho\overline{g}_n)\overline{g}_n - \nu_k - \lambda$ which gives

$$\nu_k = \sum_n (1-\rho\overline{g}_n)\overline{g}_n - \lambda. \tag{69}$$

Thus the gradient flow in W_k is the following dynamical system for $k < L$

$$\dot{W}_k = 2\sum_n (1-\overline{g}_n)(\frac{\partial \overline{g}_n}{\partial W_k} - W_k\overline{g}_n) \tag{70}$$

and

$$\dot{W}_L = 2\sum_n (1-\overline{g}_n)\frac{\partial \overline{g}_n}{\partial W_L} - 2\lambda W_L \tag{71}$$

This dynamics corresponds to the dynamics in ρ and V_k of section 3.4.2.

H.3.2 Gradient Descent with typical normalization

Here we take into account the effect of discretization for Figure 1A as we did in a previous section with η terms added to the left-hand side of the previous set of equations obtaining

$$\frac{\eta}{2}\ddot{W}_k + \dot{W}_k = 2\sum_n (1-\overline{g}_n)(\frac{\partial \overline{g}_n}{\partial W_k} - W_k\overline{g}_n) \tag{72}$$

and

$$\frac{\eta}{2}\ddot{W}_L + \dot{W}_L = 2\sum_n (1-\overline{g}_n)\frac{\partial \overline{g}_n}{\partial W_L} - 2\lambda W_L \tag{73}$$

I Experiments on generalization

The property discussed above can be checked qualitatively by varying the margin using different degrees of random labels in CIFAR in a binary classification task. While training gives perfect classification and almost zero square loss, the margin on the training set decreases and the test error also increases with the percentage of random labels as shown in Fig. 11. See also Fig. 13. These results are fully consistent with the generalization bounds Equations 21 and 22.

However, the margin does not explain the behavior shown in Fig. 12 where small differences in margin are actually anticorrelated with small differences in test error. Tighter bounds (see [32]) may explain these small effects. Alternatively, if there exist several almost-interpolating solutions with the same norm ρ_0, they may have similar norm and similar margin but different ranks of the weight matrices (or of the rank of the local Jacobian). 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 [44]. It is tempting to conjecture that the rank may have a role in generalization and in explaining Fig. 12.

J Additional derivations of Neural Collapse

In section 4 of the main paper we showed that the phenomenon of Neural Collapse can be derived from the critical points of gradient flow under the square loss with Weight Decay. In this section we present derivations of Neural Collapse in two more situations. In section J.1 we show that the critical points of gradient flow under weight normalization also exhibits Neural Collapse. In section J.2 we show that NC occurs in the case of deep networks trained with the exponential loss as well.

J.1 Normalization case

In this subsection we stick to the setting of section 4 of the main paper and 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 $f_W : \mathbb{R}^d \to \mathbb{R}^C$, $f_W(x) = W_L\sigma(W_{L-1}\dots W_2\sigma(W_1x)\dots)$ with Gradient Descent on the square loss with Weight Decay on the parameters of the network. Under the normalized parameterization, we have $f_W(x) = \rho f_V(x)$, where $f_V(x) = V_L\sigma(V_{L-1}\dots V_2\sigma(V_1x)\dots)$ is the normalized network. We use the following array notation to denote the output vectors and the one-hot target vectors respectively: $f_V(x) = [f_V^{(i)}(x)]$, $y_n = [y_n^{(i)}]$. 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 $f_V^{(c)}(x) = \langle V_L^c, h(x)\rangle$. Similar to section 4 of the main paper, we assume that the solution obtained by Gradient Descent satisfies the Symmetric Quasi-interpolation condition which we recall below:

Assumption 2 (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 $f^{(c')}(x_{n(c)}) = \frac{\epsilon}{C-1}$.*

As we mentioned, this condition follows from the dynamics under the square loss for the case of binary classification and thus is likely to hold also for the multiclass case. This brings us to the main result of this section:

Theorem 11 *For a ReLU deep network trained on a balanced dataset using gradient flow on the square loss with Weight Normalization and Weight Decay, critical points of Gradient Flow that satisfy Assumption 2 also satisfy the NC1-4 conditions for Neural Collapse.*

Proof Our training objective is $\mathcal{L}(\rho, V) = \frac{1}{2}\sum_{n=1}^{NC} ||y_n - \rho f_V(x_n)||^2 + \frac{\lambda}{2}\rho^2$. We can use Gradient Flow with Weight Normalization algorithm to train the network. We can relate this algorithm to Gradient Flow on the regular network parameterization f_W through the following equation: $\frac{\partial V_k}{\partial t} = \rho_k S_k \frac{\partial W_k}{\partial t} = -\rho_k S_k \frac{\partial \mathcal{L}(W)}{\partial W}$, where $S_k = I - V_k V_k^\top$ For the parameters of the last layer, this translates to: $\frac{\partial V_L}{\partial t} = -\rho\left(I - [V_L^i (V_L^j)^\top]\right)\frac{\partial \mathcal{L}(W)}{\partial W_L}$. This means:

$$\frac{\partial V_L}{\partial t} = \rho\left[\sum_n [(y_n^{(i)} - \rho f_V^{(i)}(x_n))h(x_n)] - \sum_n \langle y_n - \rho f_V(x_n), f_V(x_n)\rangle V_L\right] \tag{74}$$

Now let us analyze the critical points of the dynamics of the last layer, considering each classifier vector V_L^c of V_L separately:

$$\sum_n \langle y_n - \rho f_V(x_n), f_V(x_n)\rangle V_L^c = \sum_n (y_n^{(c)} - \rho f_V^{(c)}(x_n))h(x_n) \tag{75}$$

Let us consider solutions that achieve *symmetric quasi-interpolation*, with $\rho f_V^{(c)}(x_{n(c)}) = 1 - \epsilon$, and $\rho f_V^{(c)}(x_{n(c')}) = \frac{\epsilon}{C-1}$. It is fairly straightforward to see that since $f_V^{(c)}$ and V_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:

$$\begin{aligned}\frac{CN}{\rho}\left(\epsilon(1-\epsilon) - \frac{\epsilon^2}{C-1}\right)V_L^c &= \epsilon N\mu_c - \frac{\epsilon N}{C-1}\sum_{j\neq c}\mu_j \\ &= \frac{\epsilon NC}{C-1}(\mu_c - \mu_G) \\ \implies V_L^c &= \frac{\rho}{1 - \frac{C}{C-1}\epsilon}\frac{1}{C-1}(\mu_c - \mu_G)\end{aligned} \tag{76}$$

This implies that the last layer parameters V_L are a scaled version of the centered class-wise feature matrix $M = [\dots \mu_c - \mu_G \dots]$. Thus at equilibrium, with quasi interpolation of the training labels, we obtain $\frac{V_L}{||V_L||_F} = \frac{M^\top}{||M|_F}$. 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 (V_L^c, and hence $\mu_c - \mu_G$) have the same norm:

$$\begin{aligned} ||V_L^c||_2^2 &= \frac{\sum_n (y_n^{(c)} - \rho f_V^{(c)}(x_n)) f_V^{(c)}(x_n)}{\sum_n \langle y_n - \rho f_V(x_n), f_V(x_n) \rangle} \\ &= \frac{\frac{N\epsilon}{\rho}(1-\epsilon) - \frac{N\epsilon^2}{\rho(C-1)}}{\frac{CN}{\rho}\left(\epsilon(1-\epsilon) - \frac{\epsilon^2}{C-1}\right)} \\ &= \frac{1}{C} \end{aligned} \tag{77}$$

From the quasi-interpolation of the correct class label we have that $\langle V_L^c, \mu_c \rangle = \frac{1-\epsilon}{\rho}$ which means $\langle V_L^c, \mu_G \rangle + \langle V_L^c, \mu_c - \mu_G \rangle = \frac{1-\epsilon}{\rho}$. Now using (76)

$$\begin{aligned} \langle V_L^c, \mu_G \rangle &= \frac{1-\epsilon}{\rho} - \frac{\left(1 - \frac{C}{C-1}\epsilon\right)(C-1)}{\rho} ||V_L^c||_2^2 \\ &= \frac{1-\epsilon}{\rho} - \frac{\frac{C-1}{C} - \epsilon}{\rho} = \frac{1}{\rho C}. \end{aligned} \tag{78}$$

From the quasi-interpolation of the incorrect class labels, we have that $\langle V_L^c, \mu_{c'} \rangle = \frac{\epsilon}{\rho(C-1)}$, which means $\langle V_L^c, \mu_{c'} - \mu_G \rangle + \langle V_L^c, \mu_G \rangle = \frac{\epsilon}{\rho(C-1)}$. Plugging in the previous result and using (77) yields

$$\begin{aligned} \frac{(C-1)\left(1 - \frac{C}{C-1}\epsilon\right)}{\rho} \times \langle V_L^c, V_L^{c'} \rangle &= \frac{\epsilon}{\rho(C-1)} - \frac{1}{\rho C} \\ \implies \langle \tilde{V}_L^c, \tilde{V}_L^{c'} \rangle &= \frac{1}{||V_L^c||_2^2} \times \frac{-1}{C(C-1)} = -\frac{1}{C-1} \end{aligned} \tag{79}$$

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

J.2 Exponential case

Here we show that in the case of binary classification with exponential loss (a proxy for logistic loss), we can derive the Neural Collapse properties if we assume SGD critical points (that is equilibria achieved with SGD with minibatch size 1). We leave an extension of this calculation to the multi-class case to future work.

For the exponential loss with normalization and weight decay the gradient flow corresponding to GD are $\frac{\partial \rho_k}{\partial t} = \frac{1}{N}\sum_n e^{-\rho y_n f_n} \overline{f}_n - \lambda\rho$ and $\frac{\partial V_k}{\partial t} = \rho \frac{1}{N}\sum_n e^{-\rho y_n f_n} y_n S_k \frac{\partial f_n}{\partial V_k}$

The equations at equilibrium for the ρ flow are

$$\frac{1}{N}\sum_n e^{-\rho y_n f_n} y_n f_n - \lambda\rho = 0 \tag{80}$$

For GD a critical point implies $\lambda\rho = \frac{1}{N}\sum_n e^{-\rho y_n f_n} y_n f_n$

The condition above does not by itself imply that all the margins $y_n f_n$ are the same (which is required for NC1). However, let us add the separability assumption and the assumption of SGD equilibria in the flow of $_k$ and ρ: equilibria for SGD with minibatch[10] size of 1 implies that each of the terms should be vanishing independently, i.e.

$$e^{-\rho y_n f_n} y_n f_n = \lambda\rho, \quad \forall n = 1, \cdots, N \tag{81}$$

[10] We conjecture that the argument is valid also for minibatch sizes larger than 1 but smaller than N, see [9].

and thus $e^{-\rho y_1 f_1} y_1 f_1 = e^{-\rho y_n f_n} y_n f_n, \quad , \forall n = 1, \cdots, N$. These trascendental equations suggest that $f_1 = \cdots = f_n$ because of the constraints on ${}_n f_n$ such that $1 \geq y_n f_n > 0$. Hence, the margins $y_i f_i$ are all equal. Then $V_L h_{i,c} = f^c$ is also independent of i implying that $h_{i,c}$ is independent of i at convergence.

With this result, the other NC properties follow in similar way as in the square loss case.

From the $\frac{\partial V_L}{\partial t} = 0$ equation, we immediately get that at the critical points $h(x_+) \sim V_L f(x_+)$ and $h(x_-) \sim V_L f(x_-)$, giving us $h(x_+) = -h(x_-)$. This also implies that $\mu_+ = -\mu_-$ and $\mu_G = 0$. It follows then that, defining $M = [\mu_+ - \mu_G, \mu_- - \mu_G]$, NC3 follows as $\frac{W_L}{||W_L||_F} \equiv V_L = \frac{M^\top}{||M||_F}$.

The fact that $\mu_+ = -\mu_-$ also immediately gives us the equinorm property, and we get that $\langle \mu_+, \mu_- \rangle = -\frac{1}{2-1} = -1$ and hence $\langle \mu_c, \mu_{c'} \rangle = \frac{C}{C-1}\delta_{c,c'} - \frac{1}{C-1}$. Thus NC2 follows. As in the case of square loss, by the results of [4] we also get NC4, completing the proof.

The proof for the exponential loss requires SGD, unlike the square loss case. We do not know whether this is is just a technicality. *We conjecture that is not. In this case the prediction would be the NC1 should be found under the square loss case with or without SGD, whereas NC1 under the exponential loss requires SGD. We further conjecture that small minibatch sizes should be better than large ones for the exponential loss case.*

K More details on Neural Collapse

In this section we recap the details of Neural Collapse as described in [4]. It is an empirical phenomenon that has been observed in the terminal phase of training deep networks, which we can associate with training beyond the point of separation. We listed the four conditions that are associated with Neural Collapse in section 4 of the main paper. Here we present them in a more formal manner.

We first define a deep network $f_W(x) = W_L h(x)$, where $h(x) \in \mathbb{R}^p$ denotes the last layer features of the deep network, and $W_L \in \mathbb{R}^{C \times p}$ contains the parameters of the classifier. The network is trained on a C-class classification problem on a balanced dataset $\{(x_n, y_n)\}$ with N samples per class. We can compute the per-class mean of the last layer features as:

$$\mu_c = \frac{1}{N} \sum_{n \in N(c)} h(x_n) \tag{82}$$

The global mean of all features can be computed as:

$$\mu_G = \frac{1}{C} \sum_c \mu_c \implies \mu_G = \frac{1}{NC} \sum_{n=1}^{NC} h(x_n) \tag{83}$$

The second order statistics of the last layer features can be computed as:

$$\begin{aligned} \Sigma_W &= \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N} \sum_{n \in N(c)} (h(x_n) - \mu_c)(h(x_n) - \mu_c)^\top \\ \Sigma_B &= \frac{1}{C} \sum_{c=1}^{C} (\mu_c - \mu_G)(\mu_c - \mu_G)^\top \\ \Sigma_T &= \frac{1}{NC} \sum_{n=1}^{NC} (h(x_n) - \mu_G)(h(x_n) - \mu_G)^\top \end{aligned} \tag{84}$$

Where Σ_W is the within class covariance of the features, Σ_B is the between class covariance, and Σ_T is the total covariance of the features ($\Sigma_T = \Sigma_W + \Sigma_B$).

We can now list the formal conditions for Neural Collapse:

NC1 (Variability collapse) $\Sigma_W \to 0$, or within-class variability of last-layer training activations collapses to zero.

NC2 (Convergence to Simplex ETF) $|||\mu_c - \mu_G||_2 - ||\mu_{c'} - \mu_G||_2| \to 0$, or the centered class means of the last layer features become equinorm. Moreover, if we define $\tilde{\mu}_c = \frac{\mu_c - \mu_G}{||\mu_c - \mu_G||_2}$, then we have $\langle \tilde{\mu}_c, \tilde{\mu}_{c'} \rangle = -\frac{1}{C-1}$ for $c \neq c'$, or the centered class means are also equiangular. The equinorm condition also implies that $\sum_c \tilde{\mu}_c = 0$, i.e. the centered features lie on a simplex.

NC3 (Self-Duality) If we collect the centered class means into a matrix $M = [\mu_c - \mu_G]$, we have $\left|\left|\frac{W^\top}{||W||_F} - \frac{M}{||M||_F}\right|\right| \to 0$, or that the classifier W and the last layer feature means M become duals of each other.

NC4 (Nearest Center Classification) The classifier implemented by the deep network eventually boils down to choosing the closest mean last layer feature $\text{argmax}_c \langle W_L^c, h(x)\rangle \to \text{argmin}_c ||h(x) - \mu_c||_2$

In figure 4 of the main paper we track the convergence of a deep network to Neural Collapse by primarily observing NC1 and NC2. We see in the first panel that $\text{Tr}(\Sigma_W \Sigma_B^{-1}) \to 0$, which implies that NC1 is achieved, while in the second and third panel we track the equinorm and equiangular conditions. The second panel plots the ratio of standard deviation to the mean value of $||\mu_c - \mu_G||_2$, and $||W_L^c||_2$ in red and blue respectively, while the third panel plots the ratio of standard deviation to the mean of $\frac{1}{C-1} + \cos(\mu_c - \mu_G, \mu_{c'} - \mu_G)$, and $\frac{1}{C-1} + \cos(W_L^c, W_L^{c'})$ in red and blue respectively. The convergence of all quantities to zero indicates that the NC2 conditions are also achieved. NC3 and NC4 follow accordingly.

L Additional Figures

In our random label experiments, we trained a simple 4-layer ConvNet with BN and Weight Decay 0.01, initialization 0.1 using the square loss and different random label ratios ($r = 20\%, 40\%, 60\%, 80\%$ and 100%) for binary classification on two classes of the CIFAR10 dataset. The total train and test data size are 10000 and 2000, respectively. The training dynamics of the product norm ρ w.r.t. different random label ratios over 10 runs are shown in Fig. 13. The asymptotic product norm ρ values are increased from 80 to 120 with the increasing percentages of random labels, while the margin are decreased with the increasing percentages of random labels.

The dynamics of ρ for the binary classification experiment trained with ground-truth labels using BN and Weight Decay 0.01 are shown in Fig. 14. The results in different rows are based on three different initializations (0.01, 1 and 5). The first two columns show the dynamics of ρ results over 10 runs for the first 3 epochs and 30 epochs, respectively. The last column indicates the ρ dynamics over 1000 epochs, where we can observe that the asymptotic ρ values are mostly similar and are independent of different initializations (0.01, 1 and 5) when we applied Weight Decay during training. However, the dynamics of ρ for the case without Weight Decay (see Fig. 15) are unstable across different initailizations; moreover, large initialization (e.g., init = 5) shown in the third row of Fig. 15 achieved large variations even across different training runs.

Additionally, the final training and test performance for our binary classification experiments with BN and Weight Decay 0.01 are shown in Fig. 16. Two main observations can be made from the results: 1) Small initialization 0.01 achieved the highest mean test accuracy (93.71%) with small standard deviation in the first row; large initializations (1 and 5) in the second and third rows also achieved relative good mean test accuracy but with large standard deviations over 10 runs. 2) The training loss converged faster with small initialization than large initializations (1 and 5) during training over 1000 epochs. The training accuracy for small initialization also converged faster compared to using large initializations (1 and 5).

Fig. 17 shows the binary classification performance trained with BN and no Weight Decay. This generates similar trends as discussed in the previous results with Weight Decay (Fig. 16), i.e., small initialization (0.01) produced smaller training loss and better test accuracy. The training/test accuracy converged much faster compared to the two large initalization cases (1 and 5 in the second and third rows), which also resulted in smaller standard deviations over 10 runs for both training and testing. However, large initializations achieved more *unstable* training and testing performances with large standard deviation over 10 different training runs. Without using Weight Decay, the obtained test performance varies substantially across different initializations. Large initailizations achieved much lower test accuracy.

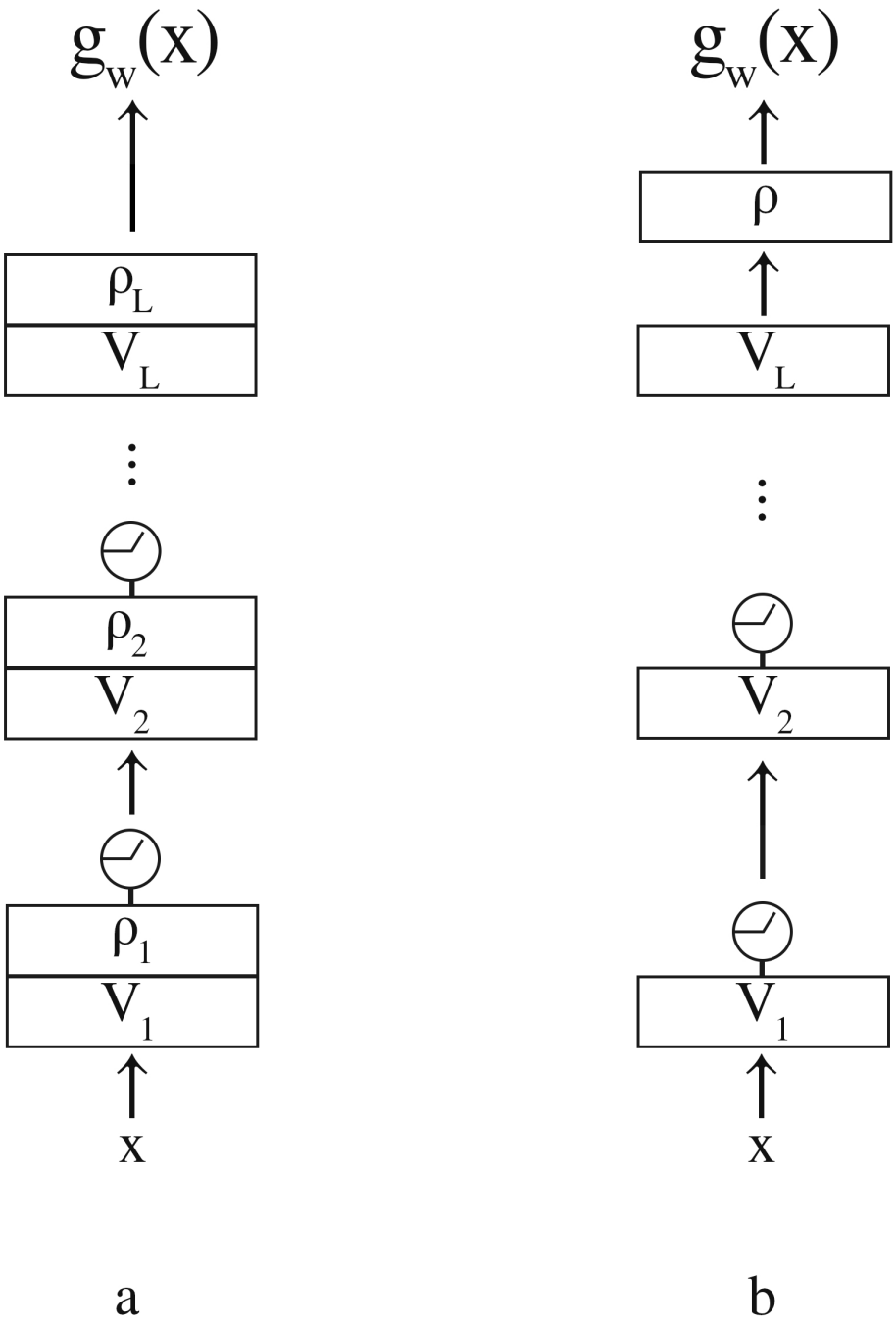


Figure 1: *Two parametrizations of a deep network. The circles represent the RELU nonlinearity. Each box corresponds to a layer. We use network a) in the case in which the weight matrices $W_k = \rho_k V_k$ with $||V_k|| = 1$ are not normalized by an algorithm like LM or BN. We use network b) when the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

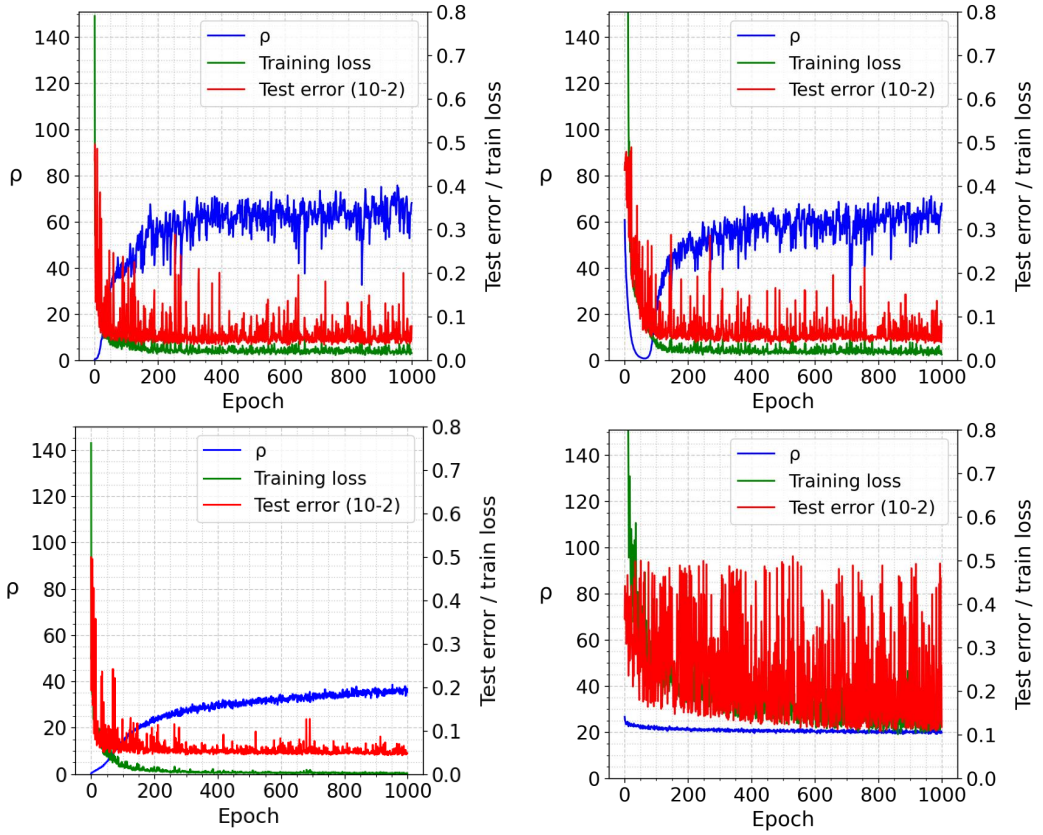


Figure 2: *Training dynamics of product norm ρ, training loss and test error over 1000 epochs with small initialization (0.01) in the first column and large initialization (5) in the second column. The first row is with Weight Decay = 0.01, and the second row is with Weight Decay = 0.*

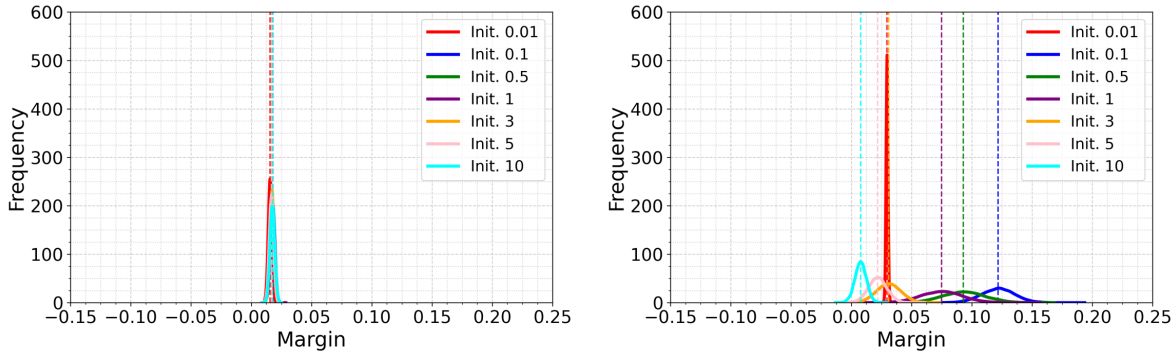
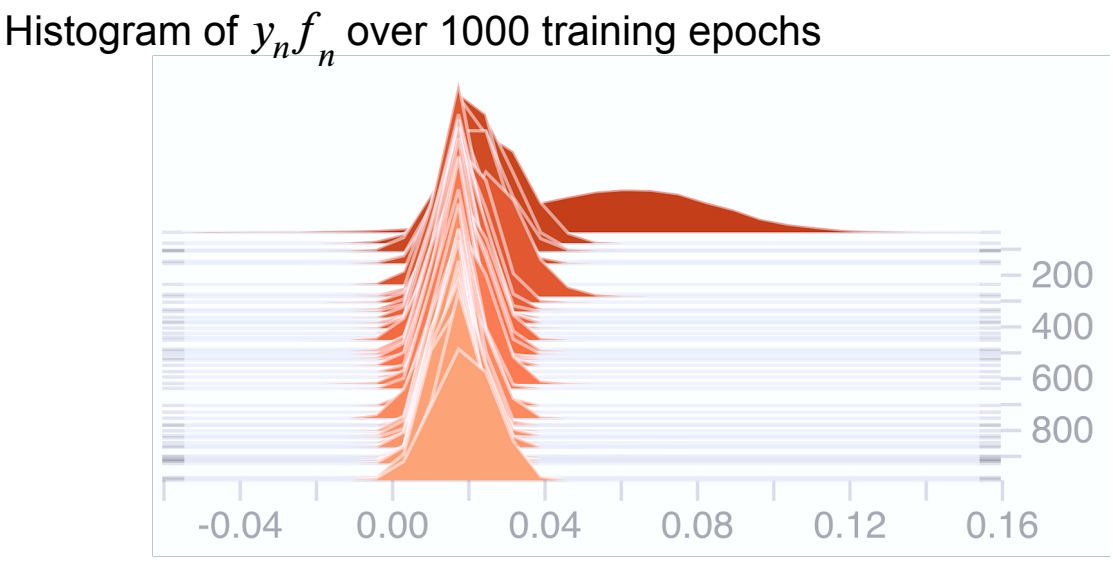


Figure 3: *Mean training margins over 10 runs for binary classification on CIFAR10 trained with Batch Normalization and Weight Decay = 0.01 (left) and without Weight Decay (right) for different initializations ($init.$ = 0.01, 0.1, 0.5, 1, 3, 5 and 10). Weight Decay makes the final training margin robust 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.*



Experimental settings: BN + WD 0.01, learning rate 0.01, binary classification, CIFAR10 dataset, output size = 2.

Figure 4: *Histogram of $y_n f_n$ 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.*

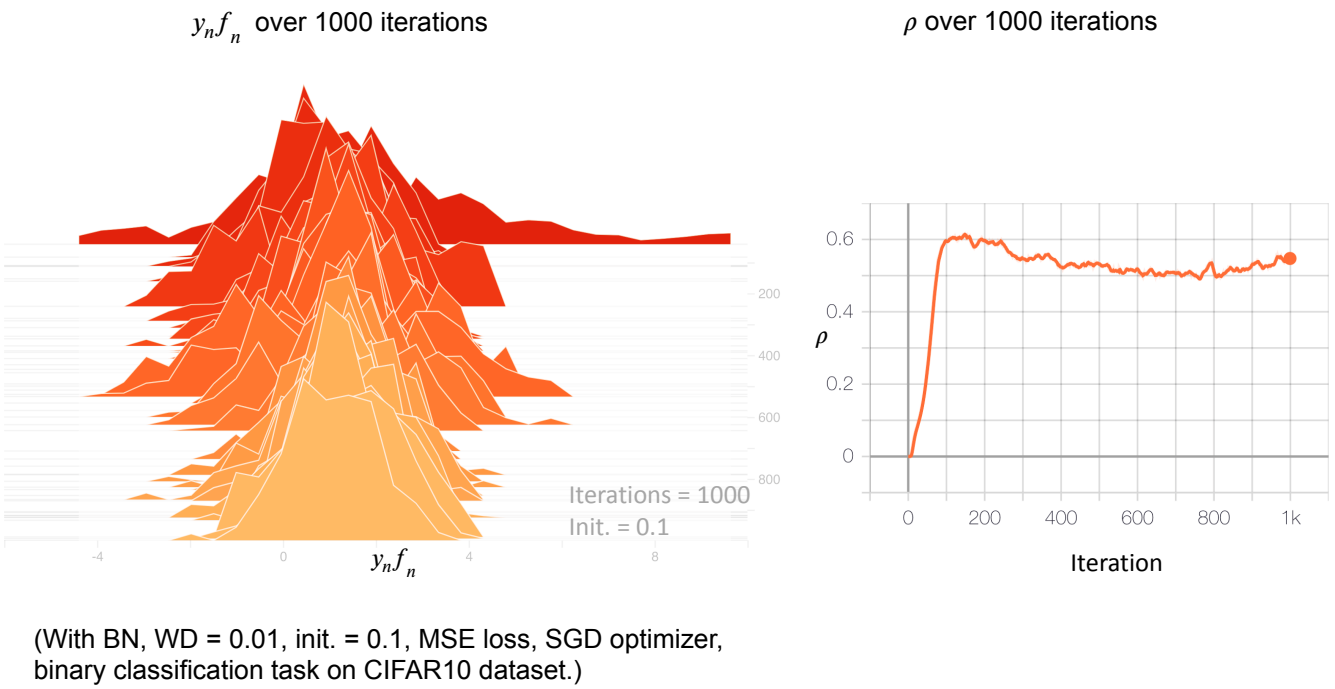


Figure 5: *Histogram of $y_n f_n$ in the initial phase of training – across the first 1000 training iterations of SGD. Same parameters as in Figure 4. The right side of the plot shows the time course of ρ over the same 1000 iterations.*

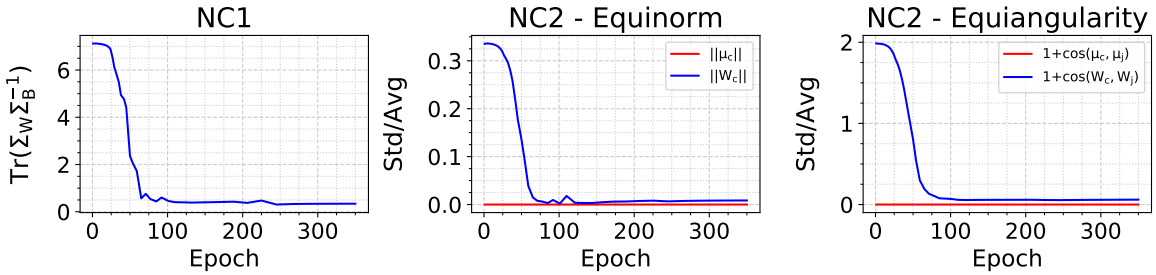


Figure 6: *Neural Collapse occurs during training for binary classification. The key conditions for Neural Collapse are: (i) NC1 - Variability collapse, which is measured by $Tr(\Sigma_W \Sigma_B^{-1})$, where Σ_W, Σ_B are the within and between class covariances, and (ii) NC2 - equinorm and equiangularity 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.*

Figure 7: *The landscape of the loss landscape with a global degenerate valley for $\rho \geq \rho_0$ with V_1 and V_2 weights weights of unit norm.*

Figure 8: *The landscape of the loss landscape with a global degenerate valley for $\rho \geq \rho_0$ with V_1 and V_2 weights of unit norm.*

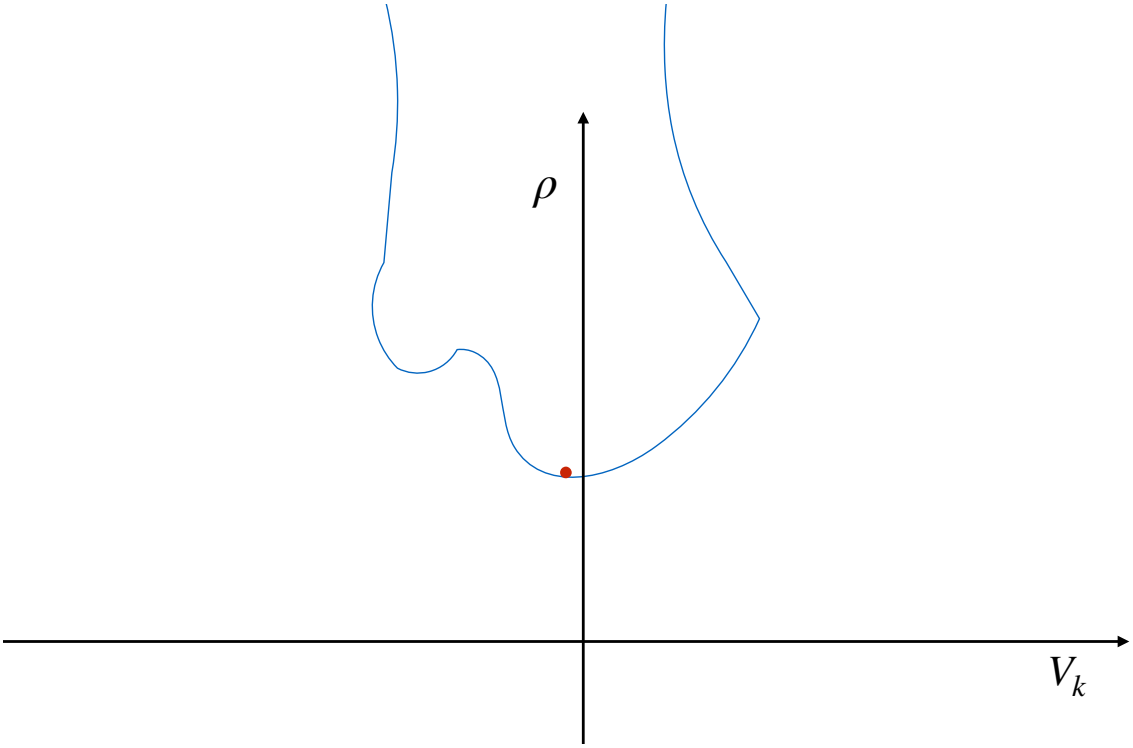


Figure 9: *The boundaries of the global degenerate minimum where $\mathcal{L} = 0$ in the high-dimensional space of ρ and $V_k \forall k = 1, \cdots, L$. The degenerate global minimum is shown here as a connected valley. The red dot marks the minimum norm minimum. Notice that depending on the shape of the multidimensional valley regularization may not guarantee convergence to the minimum norm solution, unlike the linear case.*

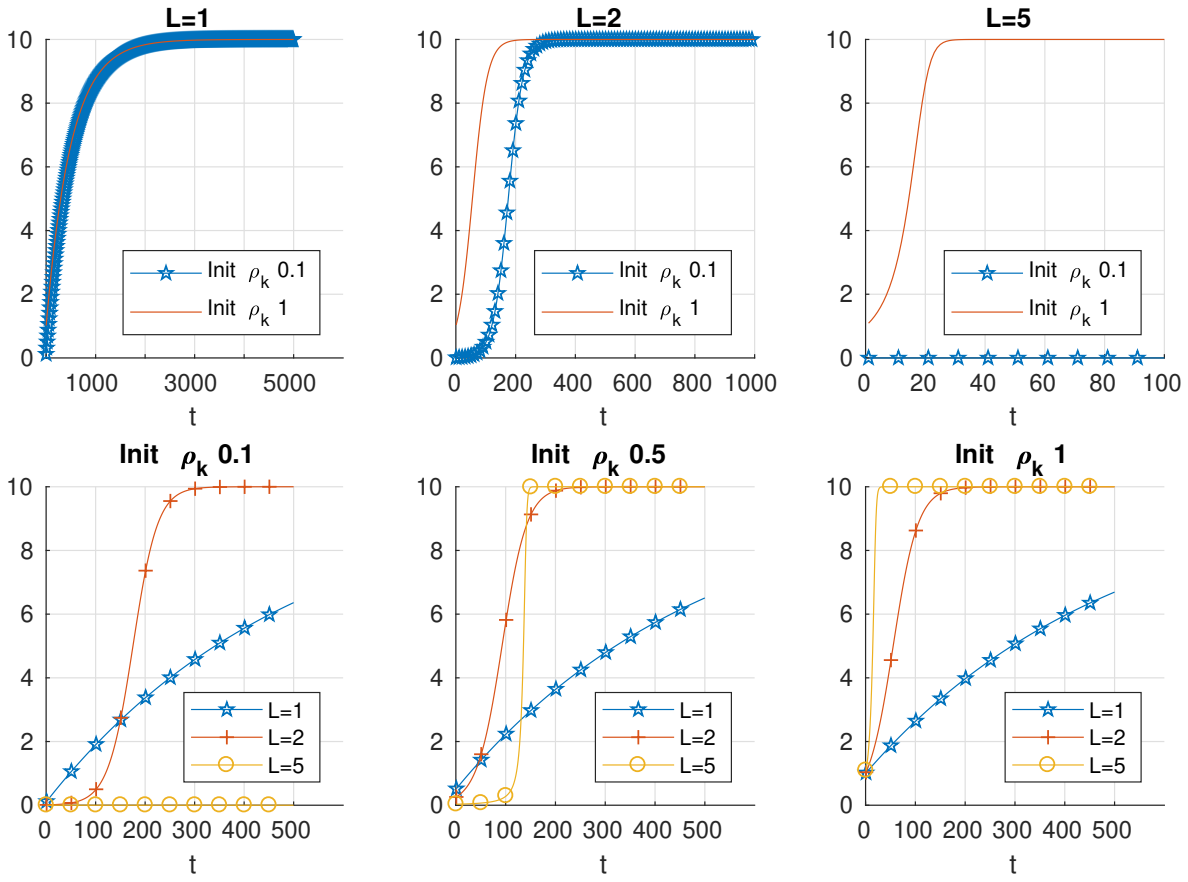


Figure 10: *Simulation of ρ from the logistic equation related to Equation 51, in which the terms $\sum y_n f_n$ and $\sum f_n^2$ are positive constants.*

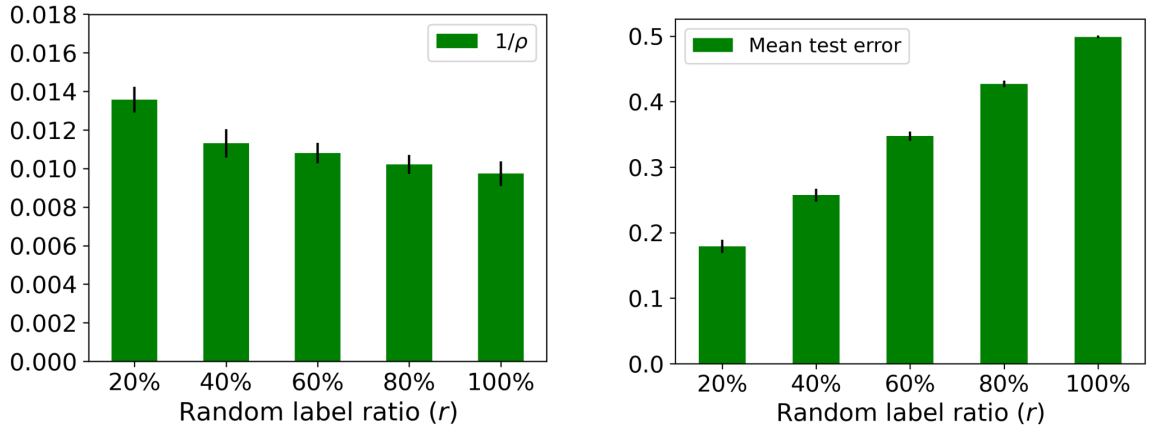


Figure 11: *Mean $1/\rho$ and test error results over 10 runs for binary classification on CIFAR10 trained with batch normalization and different percentages of random labels (r = 20%, 40%, 60%, 80% and 100%), initialization scale 0.1 and weight decay 0.01.*

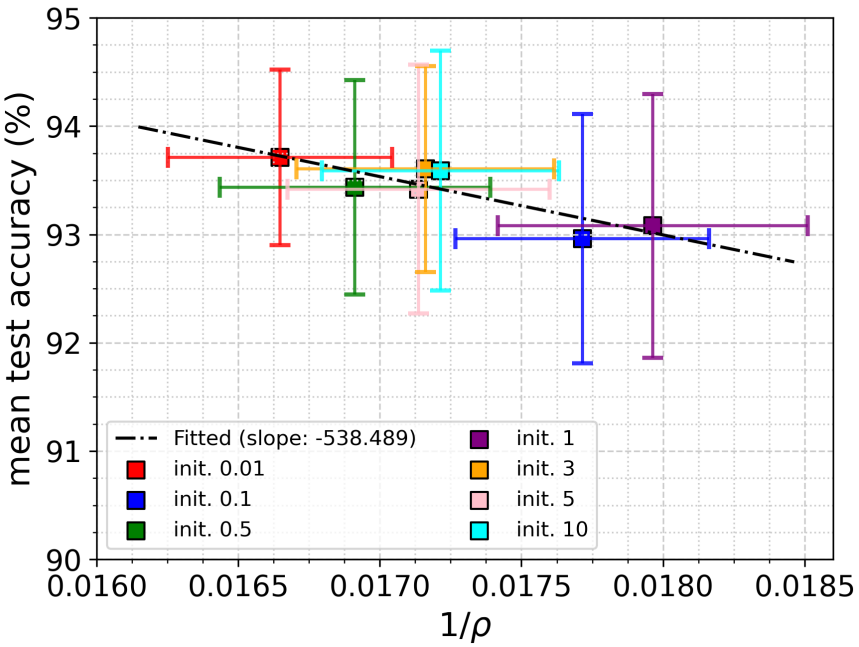


Figure 12: *Scatter plot for $1/\rho$ and mean test accuracy based on 10 runs for binary classification on CIFAR10. The network was trained with different initialization scales ($init.$ = 0.01, 0.1, 0.5, 1, 3, 5 and 10), using batch normalization and weight decay 0.01. The horizontal and vertical error bars correspond to the standard deviations of $1/\rho$ and test accuracy computed over 10 runs for different initializations, while the square dots correspond to the mean values.*

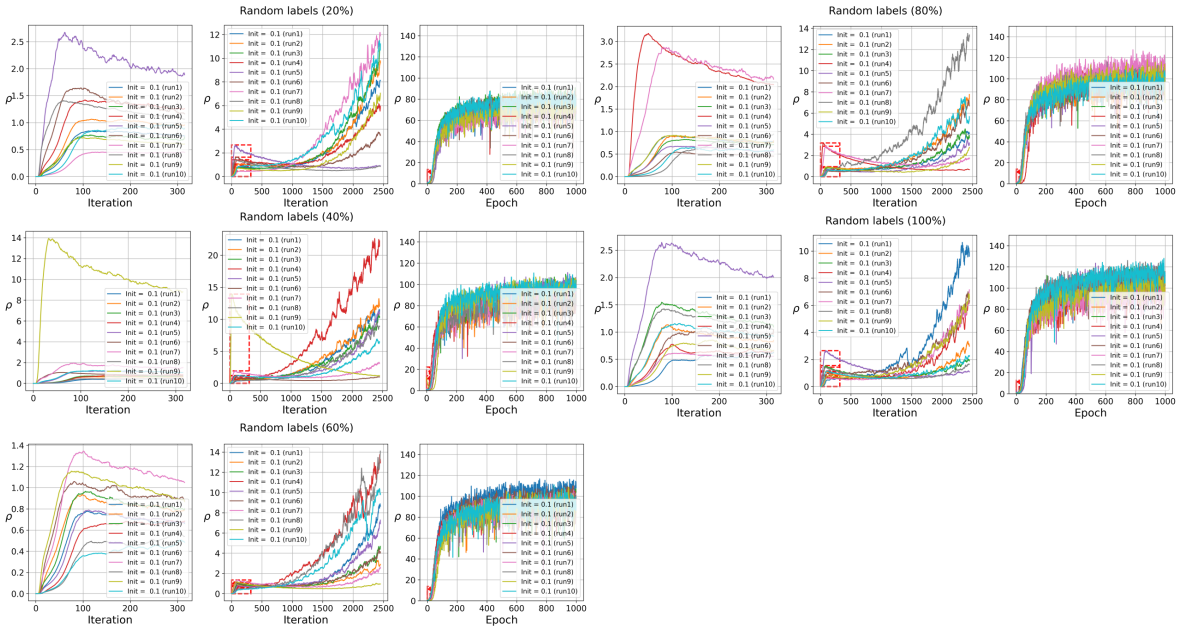


Figure 13: *Dynamics of ρ of 10 runs for the binary classification experiments in Fig. 11. The network was trained with different percentages (r = 20%, 40%, 60%, 80%, 100%) of random labels, batch normalization and weight decay 0.01.*

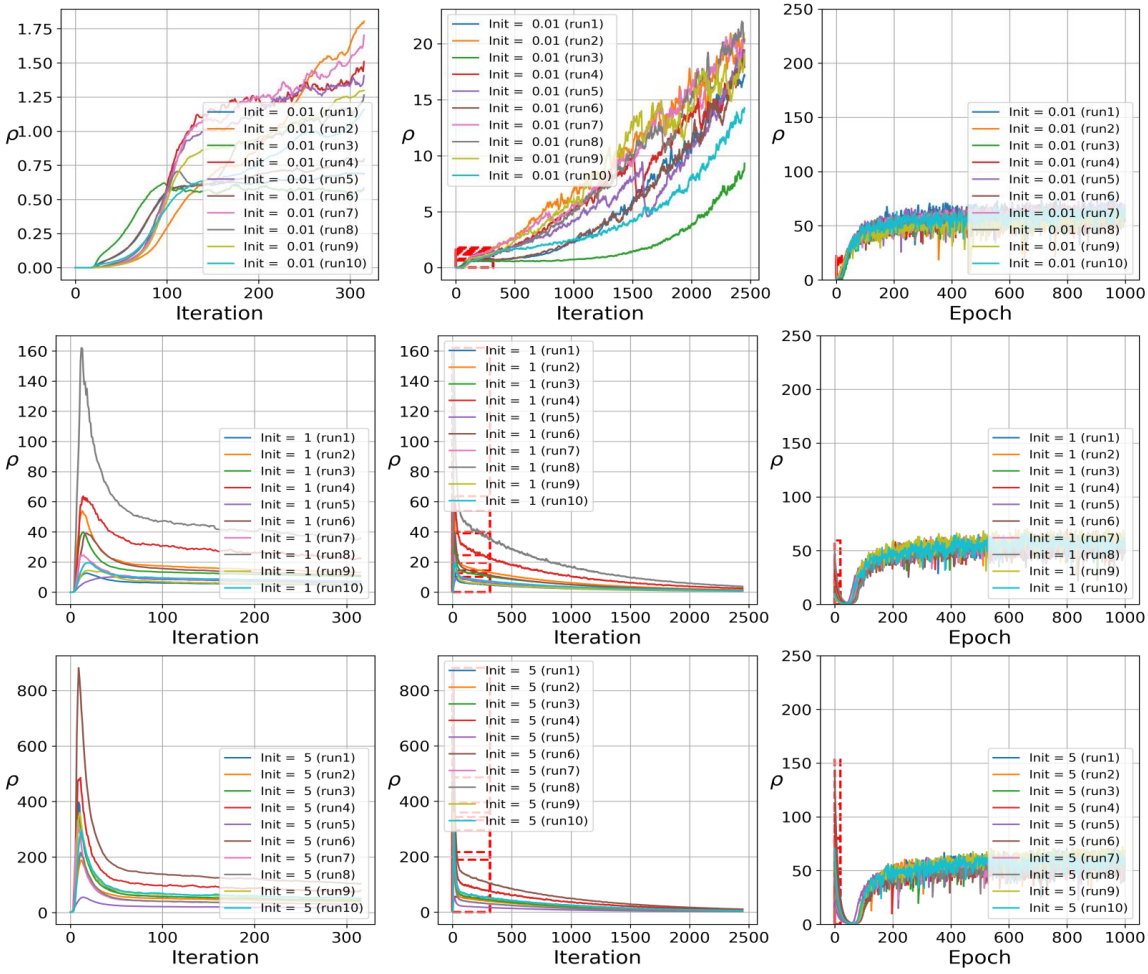


Figure 14: *Dynamics of ρ for the binary classification experiments trained with square loss, batch normalization and weight decay 0.01 for 10 runs. From top to bottom correspond to initialization 0.01, 1 and 5.*

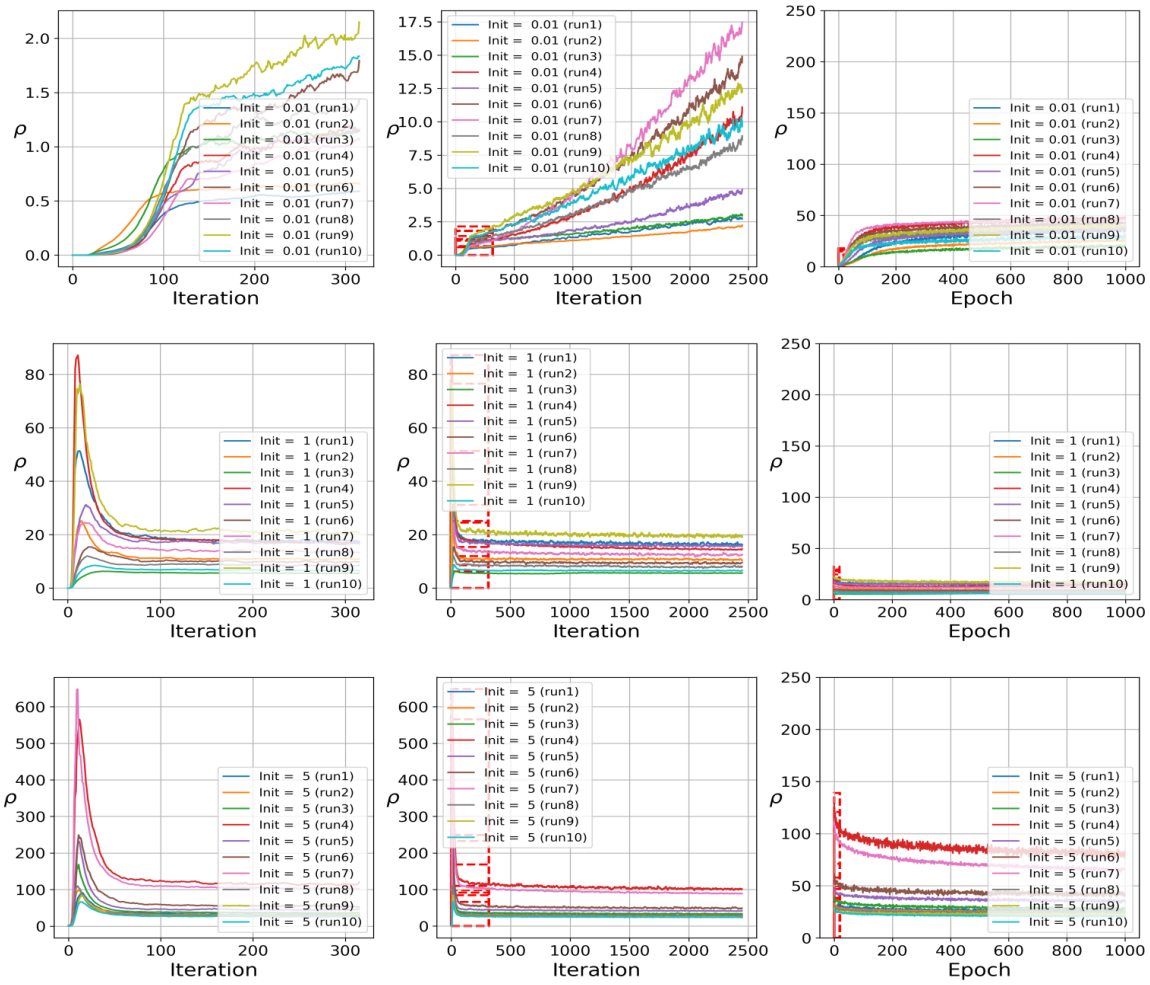


Figure 15: *Dynamics of ρ for the binary classification experiments trained with square loss over 10 runs using batch normalization and no weight decay. From the top to bottom correspond to initialization 0.01, 1 and 5, respectively.*

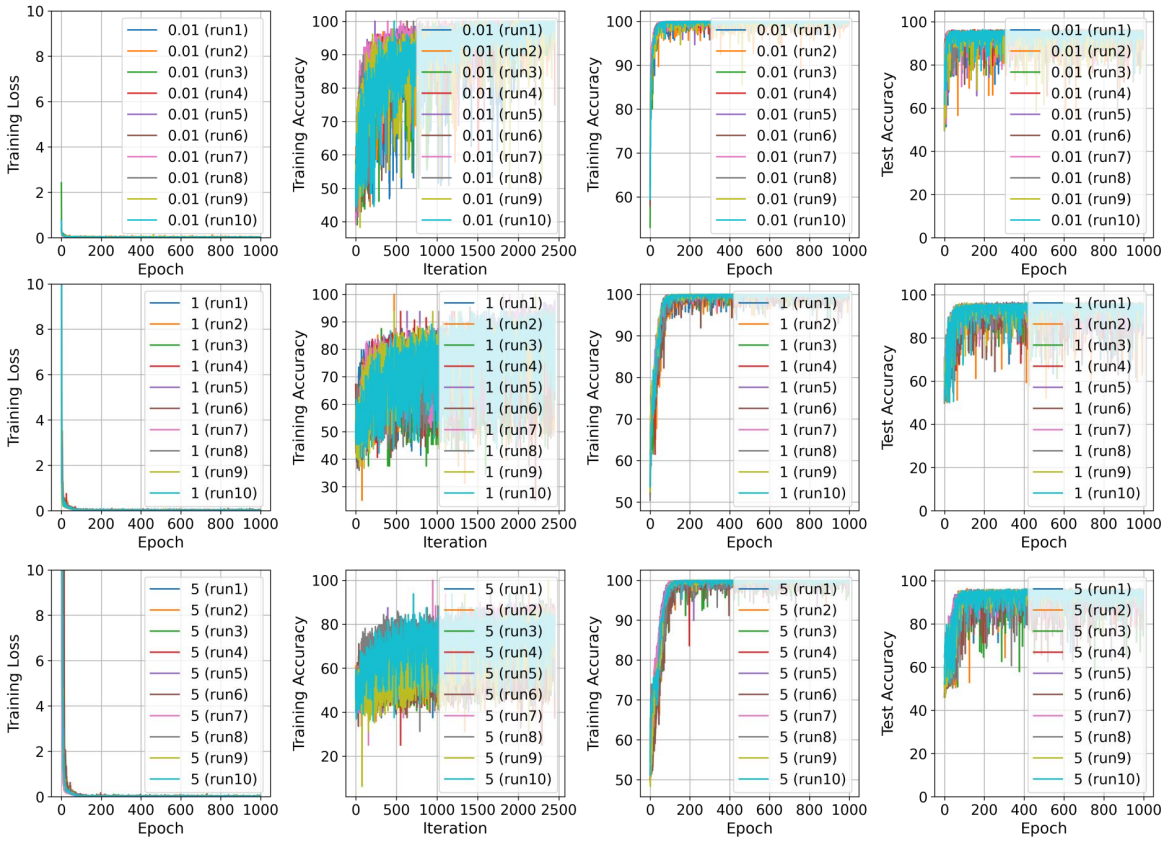


Figure 16: *Performance of binary classification over 10 runs on two classes of CIFAR10 trained with batch normalization and weight decay 0.01. From top to bottom correspond to initialization 0.01, 1 and 5, respectively.*

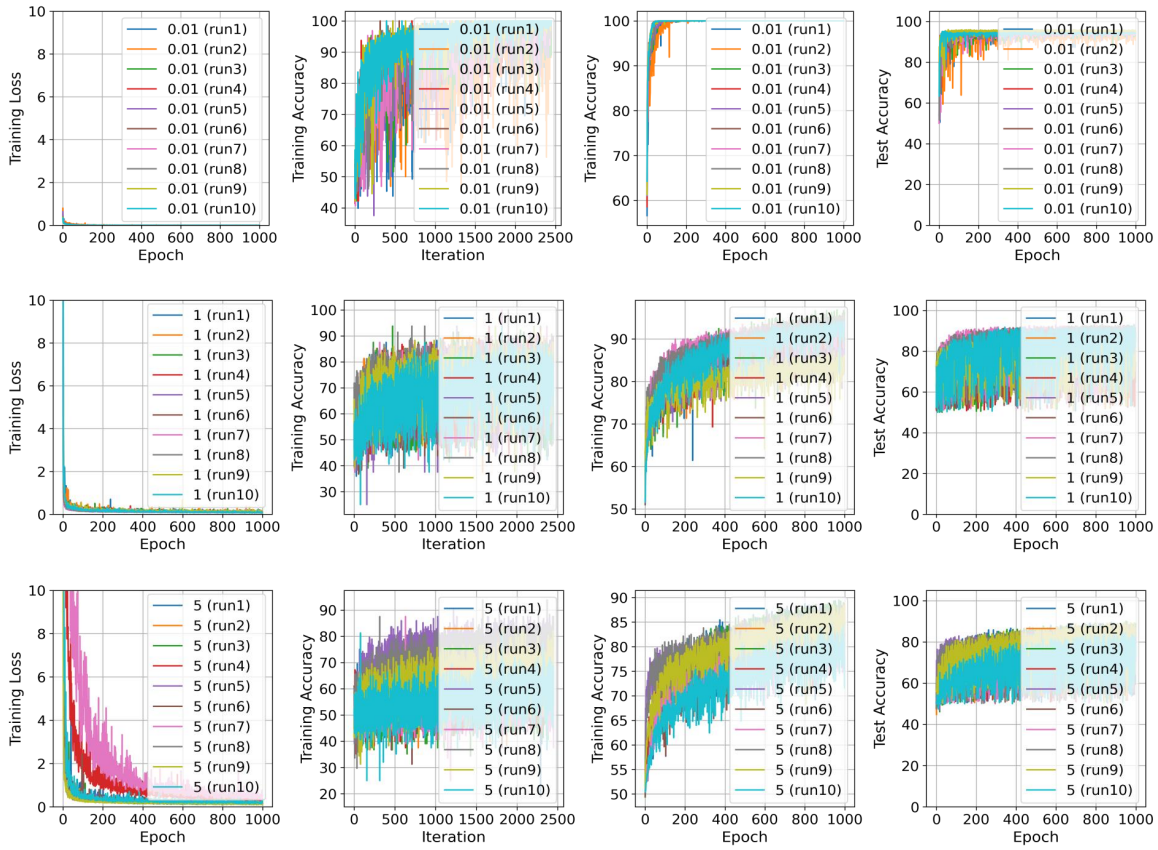


Figure 17: *Performance of binary classification on two classes of CIFAR10 trained with batch normalization and no weight decay for 10 runs. From the top to bottom correspond to initialization 0.01, 1 and 5, respectively.*