CENTER FOR
Brains
Minds+
Machines

CBMM Memo No. 115 **February 9, 2021**

Cross-validation Stability of Deep Networks

Andrzej Banburski [1,*], Fernanda De La Torre[1,*], Nishka Pant[1,2,†], Ishana Shastri[1,†], Tomaso Poggio[1]

[1]Center for Brains, Minds, and Machines, MIT
[2]Brown University
[*], [†] equal contributions

Abstract

Recent theoretical results show that gradient descent on deep neural networks under exponential loss functions locally maximizes classification margin, which is equivalent to minimizing the norm of the weight matrices under margin constraints. This property of the solution however does not fully characterize the generalization performance. We motivate theoretically and show empirically that the area under the curve of the margin distribution on the training set is in fact a good measure of generalization. We then show that, after data separation is achieved, it is possible to dynamically reduce the training set by more than 99% without significant loss of performance. Interestingly, the resulting subset of "high capacity" features is not consistent across different training runs, which is consistent with the theoretical claim that all training points should converge to the same asymptotic margin under SGD and in the presence of both batch normalization and weight decay.

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

Distribution of Classification Margins: Are All Data Equal?

Andrzej Banburski [*1] **Fernanda De La Torre** [*1] **Nishka Pant** [†1 2] **Ishana Shastri** [†1] **Tomaso Poggio** [1]

Abstract

Recent theoretical results show that gradient descent on deep neural networks under exponential loss functions locally maximizes classification margin, which is equivalent to minimizing the norm of the weight matrices under margin constraints. This property of the solution however does not fully characterize the generalization performance. We motivate theoretically and show empirically that the area under the curve of the margin distribution on the training set is in fact a good measure of generalization. We then show that, after data separation is achieved, it is possible to dynamically reduce the training set by more than 99% without significant loss of performance. Interestingly, the resulting subset of "high capacity" features is not consistent across different training runs, which is consistent with the theoretical claim that all training points should converge to the same asymptotic margin under SGD and in the presence of both batch normalization and weight decay.

1. Introduction

The key to good predictive performance in machine learning is controlling the complexity of the learning algorithm. Until recently, there was a puzzle surrounding deep neural networks (DNNs): there is no obvious control of complexity – such as an explicit regularization term – in the training of DNNs. Recent theoretical results (Lyu & Li, 2019; Poggio et al., 2020a;b; Shpigel Nacson et al., 2019), however, suggest that a classical form of norm control is present in DNNs trained with gradient descent (GD) techniques on exponential-type losses. In particular, GD induces dynamics of the normalized weights which converge for $t \to \infty$ towards an infimum of the loss that corresponds to a maximum margin solution.

What remains unclear, however, is the link between the minimum norm solutions and expected error. In this paper, we numerically study the behavior of the distribution of margins on the training dataset as a function of time. Inspired by work on generalization bounds (Bartlett et al., 2017), we provide evidence that the the area under the distribution of properly normalized classification margins is a a good approximate measure to rank different minima of a given network architecture.

The intuition is that deep minima should have a margin distribution with a relatively small number of points with small margins. This in turn suggests a training algorithm that focuses only on the training points that contribute to the stability of the algorithm – that is, on data points close to the separation boundary (once it has been established in the terminal phase of training, i.e. after hitting 0% classification error during training). We show, in fact, that, once separation is achieved, good test performance depends on improving the margin of a small number of datapoints, while the majority can be dropped (keeping only 200 from the initial 50k in CIFAR10, for example). These results suggest that certain points in the training set may be more important to classification performance than others. It is then natural to pose the following question: can we, in principle, predict which data are more important? However, we show that the points that mostly support the dynamics are not consistent between different training runs due to the initial randomness. Moreover, it turns out that before data separation, there is no clear pattern to discern which datapoints will contribute the most. Quite interestingly, this result is consistent with a recent theoretical prediction(Poggio & Liao, 2019): under stochastic gradient descent and in the presence of Batch Normalization and weight decay, all training points should asymptotically converge to the same margin and be effectively equivalent to each other. The randomness of which datapoints contribute the most to classification performance is consistent with the prediction. It also suggests the conjecture that in overparametrized models, we should expect the most important features learned by the network to be dependent on random factors such as initialization.

[*,†]Equal contributions [1]Center for Brains, Minds + Machines, MIT, MA, USA, [2]Brown University, RI, USA. Correspondence to: Andrzej Banburski <kappa666@mit.edu>.

1.1. Related Work

Following the work of (Zhang et al., 2016), which showed that overparametrized networks trained on randomized labels can achieve zero training error and expected error at chance level, recent papers analyze the dynamics of gradient descent methods. (Allen-Zhu et al., 2018; Du et al., 2018; 2019; Li & Liang, 2018; Zou et al., 2018) showed convergence of gradient descent on overparametrized non-linear neural networks. Empirical work shows that sharp minima generalize better than flat minima, that the optimization process converges to those with better generalization, (Keskar et al., 2016; Liao et al., 2018) and that better noise stability (stability of the output with respect to the noise injected at the nodes of the network) correlate with lower generalization error (Chaudhari et al., 2019; Langford & Caruana, 2002; Morcos et al., 2018). Several lines of research propose low complexity measures of the learned network to derive generalization bounds. Spectrally-normalized margin-based generalization bounds are derived in (Bartlett et al., 2017), which we test here. Bounds obtained through a compression framework that reduces the effective number of parameters in the networks based on noise stability properties are described in (Arora et al., 2018) who more recently provided a sample complexity bound that is completely independent of the network size (Arora et al., 2019). Our algorithms for dataset compression are indirectly related to the data-distillation approach introduced in (Wang et al., 2018) and to the noise stability described in (Bartlett et al., 2017).

2. Theoretical motivations

We start with a short review of the recent theoretical findings that inspire our numerical investigations.

2.1. Notation and Background

In this paper we assume the standard framework of supervised learning via Empirical Risk Minimization (ERM) algorithms for classification problems. For details see Supplementary Material or papers such as (Mukherjee et al., 2006).

Deep Networks We define a deep network with K layers with the usual coordinate-wise scalar activation functions $\sigma(z) : \mathbf{R} \to \mathbf{R}$ as the set of functions $f(W;x) = \sigma(W^K\sigma(W^{K-1}\cdots\sigma(W^1x)))$, where the input is $x \in \mathbf{R}^d$, the weights are given by the matrices W^k, one per layer, with matching dimensions. For simplicity we consider homogenous functions, i.e. without bias terms. In the case of binary classification, the labels are $y \in \{-1, 1\}$. The activation function is the ReLU activation. For the network, homogeneity of the ReLU implies $f(W;x) = \prod_{k=1}^K \rho_k f(V_1, \cdots, V_K; x)$, where $W_k = \rho_k V_k$ with the matrix norm $||V_k||_p = 1$ and $||W_k|| = \rho_k$. In the binary case, when $y_n f(x_n) > 0 \; \forall n = 1, \cdots, N$ we say that the data are *separable* wrt $f \in \mathbb{F}$, that is they can all be correctly classified. We define the margin of x_n as $\eta_n = y_n f(x_n)$ and the margin of the whole dataset as the smallest of all margins $\eta = min_n \eta_n$, corresponding to a *support vector* x^*. For the multi-class case, if the prediction score is a vector $\{f_1(x), \ldots, f_C(x)\}$ for C classes, with $f_{y_n}(x_n)$ the prediction for the true class, then the margin for x_n is $\eta_n = f_{y_n}(x_n) - \max_{j \neq y_n} f_j(x_n)$.

Dynamics & margin maximization It has been known for some time now that the norm $\rho = \prod_k \rho_k$ diverges to infinity as we run GD (Banburski et al., 2019; Lyu & Li, 2019; Poggio et al., 2020b). This means that the weights W_k do not converge in any meaningful sense, and it is only sensible to study the convergence of the normalized weights V_k.

When we minimize exponential-type losses (like the exponential loss, logistic, or cross-entropy), we expect that, asymptotically, the convergence to a data-separating solution only depends on data with the least negative exponents, i.e. the points with the smallest classification margin (the equivalent of support vectors in SVMs). Minimization of exponential-type losses then corresponds to the problem of maximizing the classification margin. Recent body of work has been showing that SGD biases highly over-parametrized deep networks towards solutions that locally maximize margin (Banburski et al., 2019; Lyu & Li, 2019; Poggio et al., 2020b) in presence of normalization techniques (such as batch normalization).

Neural Collapse Given the margin maximization results, the natural question one might ask is which data are the ones that contribute the most to the solution. As in the case of linear systems, the answer clearly depends on the amount of overparametrization. Recent empirical observations suggest that with overparametrization, SGD leads to the phenomenon of Neural Collapse (Papyan et al., 2020) after data separation is achieved. One of the Neural Collapse properties (NC1) says that within-class activations all collapse to their class means, implying that the margins of all the training points converge to the same value. While this might seem unintuitive, embedding N points in D-dimensional space with $D \gg N$ allows for many hyperplanes equidistant from all the points. A recent theoretical analysis predicts that NC1(Poggio & Liao, 2020) depends on both the use of normalization algorithms (such as batch normalization) and L2 regularization during SGD. The prediction is consistent with our results in Figure 3.

The theory thus suggests that effectively only one example per class is needed to describe the decision boundaries of the learned model – and that any of the training points could be used. We explore in this work whether this prediction is correct.

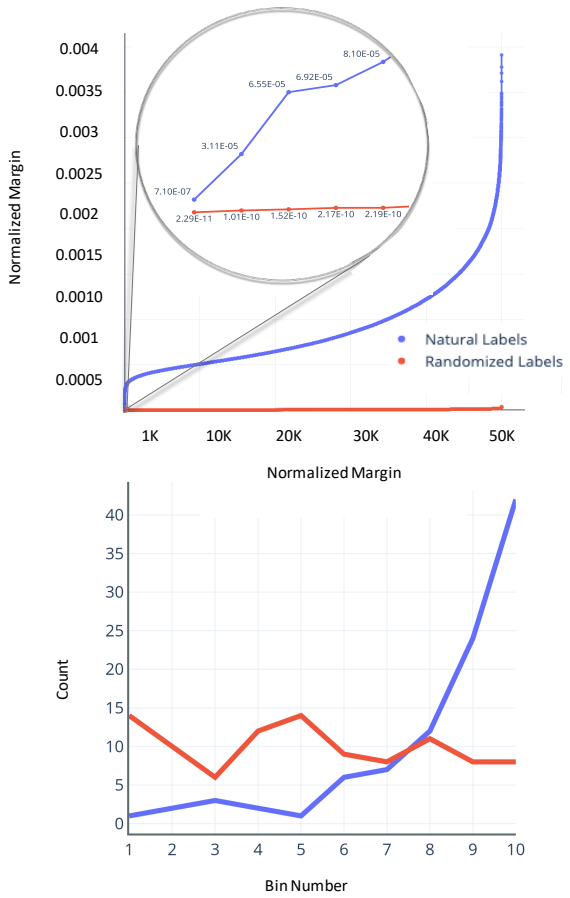

Figure 1. ***Natural and Random Labels - Margins*** *The top figure shows the margin of the 50k datapoints in CIFAR10 ranked by their individual margin size for 2 convolutional networks trained on either natural or randomized labels pass data separation and margin convergence. The circle enlarges the numerical values of the five datapoints with the smallest margins. In the bottom figure, the range of the margin of the first* 100 *datapoints (those with the smallest margin) was equally divided into 10 bins with the count of data points in each bin shown. The first two bins of the network trained with random labels have significantly more datapoints, while the network trained with natural labels ends with less support vectors closer to each other.*

2.2. Margins, ρ and expected error

Assuming that weight decay, small initialization, and batch normalization provide a bias towards a solution with "large" margin, the obvious question is whether we can obtain any guarantees of good test performance. While predicting test performance purely from training behavior is challenging, we use simple bounds (Bousquet et al., 2003) to predict relative performance between different minima for the same network architecture.

A typical generalization bound that holds with probability of at least $(1-\delta)$, $\forall g \in \mathbb{G}$ has the form (Bousquet et al., 2003):

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

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, and 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 in (Bartlett et al., 2017) as discounting predictions with margin below some arbitrary cutoff γ (with ℓ_0 being the 0-1 error, see Sup. Mat.).

We now consider two solutions with the same small training loss obtained with the same ReLU deep network and corresponding to two different minima with two different ρs and different margins. Let us call them $g^a(x) = \rho_a f^a(x)$ and $g^b(x) = \rho_b f^b(x)$ and let us assume that $\rho_a < \rho_b$. Using the notation of this paper, the functions f_a and f_b correspond to networks with normalized weight matrices at each layer.

We now use the observation that, because of homogeneity of the networks, the empirical Rademacher complexity satisfies the property $\mathbb{R}_N(\mathbb{G}) = \rho \mathbb{R}_N(\mathbb{F})$, 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 in Equation 1 and the fact that the empirical $\hat{L_\gamma}$ *for both functions is the same* to write $L_0(f^a) = L_0(F^a) \leq \hat{L_\gamma} + c_1 \rho_a \mathbb{R}_N(\tilde{\mathbb{F}}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}$ and $L_0(f^b) = L_0(F^b) \leq \hat{L_\gamma} + c_1 \rho_b \mathbb{R}_N(\tilde{\mathbb{F}}) + c_2 \sqrt{\frac{\ln(\frac{1}{\delta})}{2N}}$. The bounds have the form

$$L_0(f^a) \leq A\rho_a + \epsilon \qquad \text{and} \qquad L_0(f^b) \leq A\rho_b + \epsilon \quad (2)$$

Thus the bound for the expected error $L_0(f^a)$ is better than the bound for $L_0(f^b)$. Similar results can be obtained taking into account different $\hat{L}(f)$ for the normalized f^a and f^b under different γ in Equation 1.

Can these bounds be meaningful in practice? The solutions a and b achieve the same training loss, which means that they must both have different norms ρ and different distributions of classification margins. In what follows, we show empirically that indeed we can effectively predict the relative generalization performance using the information of the distribution of classification margins on the training set.

3. Experimental methods

In most of the numerical experiments, we used a 5-layer neural network implemented in PyTorch and trained on the CIFAR10 or CIFAR2 (cars and birds from CIFAR10)

datasets using either SGD or full GD with cross-entropy loss. The network has 4 convolutional layers (filter size 3×3, stride 2) and one fully-connected layer. All convolutional layers are followed by a ReLU activation, and for some experiments, batch normalization. The number of channels in hidden layers are 16, 32, 64, 128 respectively. In total, the network has $273,546$ parameters. The dataset was not shuffled. The learning rate was constant and set to 0.01, with momentum set to 0.9 unless otherwise stated. Our test performance is not state of the art, since we wanted to perform neither data augmentation nor any explicit regularization to match the theoretical setting. These results however extend to networks with state of the art performance, see Table 1 for results on DenseNet-BC and more in Sup. Mat.

In the absence of batch normalization, the margin distribution of each network is normalized by ρ, the product of convolutional layer norms. For networks with batch-normalization, the margin distribution was normalized by the product of the batch-normalization layer norms and the norm of the last fully-connected layer.

4. Margin distribution

The recent Neural Collapse (Papyan et al., 2020) results would suggest that at convergence the margin distribution should be flat. Convergence in the margin however is known to be very slow (Soudry et al., 2017). In this section we experimentally study the shape of the distribution of margins on the whole training dataset and then go onto using it to predict generalization performance.

4.1. Natural vs random labels

We ran numerical experiments to find the relation between the margin, stability and generalization gap for two convolutional neural networks. One was trained with natural labels and the second one trained with randomized labels, an idea explored in (Liao et al., 2018; Zhang et al., 2016).

In Figure 1, we took both networks after data separation and close to margin convergence and extracted the margins for each data point, which allowed us to sort them according to the margin. The margins for all datapoints are larger for natural labels than for randomized labels. Theory suggests that the datapoints important for margin maximization should be closer to each other in the feature space for randomized labels than for natural labels, consistent with lower algorithmic stability (Kutin & Niyogi, 2002) and chance performance. To test this, we took the 100 datapoints with the smallest margin for both networks, divided the margin range into ten bins and counted the number of datapoints in each bin. The first five bins for the network trained with natural labels had less datapoints than for randomized labels, with the first bin only having one datapoint for natural labels while for randomized labels there were 14. This experiment supports the idea that having a smaller set of datapoints with small margin (the equivalent of support vectors in SVMs) leads to both better stability and test performance. This should be contrasted with Neural Collapse – we find that the margin distribution after 200 epochs of SGD is far from flat, but rather has a few small margin datapoints and a similar number of high margin points, with a flatter middle range.

4.2. Margin distribution and generalization performance

How can we use the information about the margin distribution to predict generalization performance? In (Liao et al., 2018), it was shown that the training loss evaluated on the normalized deep network allows for a reasonable prediction of test loss, which conforms to arguments from Section 2.2. It is natural to ask then whether the smallest normalized margin or a simple function of all the margins is a potentially finer measure.

To probe several metrics, we ran 100 networks on a CIFAR2 classification task, where architecture and hyperparameters stayed constant across all networks but the standard deviation for random initialization was varied. This was motivated by (Liao et al., 2018), since we wanted to obtain networks that converge to different minima and analyze their resulting margin distributions. We found that the area under the curve (AUC) of the margin distribution is a good predictor of generalization performance as seen in the bottom of Figure 2. Moreover, the shape of the margin distribution is also a predictor: the initial curvature of the distribution indicates how many datapoints have small margins. Our experiments show that the number of small margin data can predict the range of test error.

These results are shown in Figure 2: on the top we can see the sorted margin distributions of 7 representative networks for each minima (there are 10,000 datapoints in CIFAR2). For smaller test errors, the normalized margin distribution contains higher values, higher curvature, and starts off with a higher slope relative to the margin distributions of large test error (larger initialization). In the middle, we counted the number of datapoints with small margins for each network (using a margin cutoff of 0.01 above the smallest margin, which corresponds to setting $\gamma = 0.01$ in the ramp loss) and calculated the average test error for networks with different ranges of small margin points. As shown, we find that larger proportion of small margin data results in higher averages of test error. On the bottom, we divided the 100 networks into sorted bins of 10 and calculated the average AUC and average test error for $\gamma = 0.1$. We see that the larger the margin AUC, the better the test performance.

In Section 2.2 we derived bounds for two different minima and asked if these bounds could be meaningful in practice.

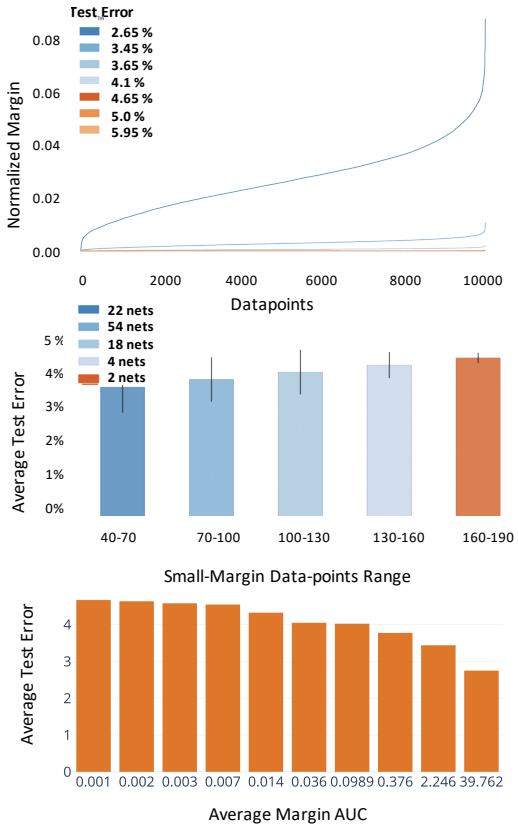

Figure 2. ***Different Minima - Margin Distributions*** *100 conv nets (as in Section 3) were initialized with varying standard deviation (from 0.01 - 0.05) so that they converge to different test errors. The top figure shows the margin distribution of 7 representative networks for each test performance. The middle image shows all 100 networks divided into bins given their number of small margin data and the average test error of these ranges. The bottom provides evidence that the AUC of the margin distribution is a predictor of generalization performance. Here we plotted the result with a cutoff of* $\gamma = 0.1$.

These experiments suggests that the shape of the margin distributions and area under it can indeed effectively predict relative generalization performance.

4.3. Time evolution of the margin distribution

Do these results mean that we are finding no NC1? On the left of Figure 3, we find that in the presence of both regularization and batch normalization, the distribution of margins does indeed get flatter with time. In further experiments we found however that without either L2 regularization or batch normalization, such flattening is not apparent, see Sup. Mat. This is in line with all the experiments in (Papyan et al., 2020) using both of these techniques, and suggests that Neural Collapse relies on both regularization and normalization in agreement with the theoretical predictions of (Poggio & Liao, 2020).

To further explore the relationship of margin distribution in the context of Neural Collapse, we visualize and analyze the margin distribution for individual classes. On the top-right of Figure 3, we observe that for some classes, the margin distribution shifts and flattens more than for other classes. For class-label 9, the margin distribution shifts and flattens more over time (going from blue at epoch 0 to green at epoch 200) than for class-label 3. This effect is absent if we do not use batch normalization, as shown in the bottom right. Thus, although margin distributions of individual classes are potentially shifting the margin distribution at different scales, this is dependent on batch-normalization and regularization (see Sup. Mat. for regularization experiments), as suggested in (Poggio & Liao, 2020).

5. Compressing the training set dynamically

As suggested by the notion of stability (Kutin & Niyogi, 2002), datapoints close to the separation boundary are crucial for good test performance. Data with large margin, however, do not contribute to stability. It has been long observed that training long past the time of achieving the separation of the data (i.e. 0% training error) leads to improved test performance. This has been understood (Soudry et al., 2017) to result from the fact that while the training classification accuracy converges fast, the margin converges much slower. As we keep training the network past separability, the margin keeps improving (Banburski et al., 2019; Lyu & Li, 2019; Shpigel Nacson et al., 2019) (see also the Sup. Mat.), with the largest contributions to this improvement coming from the datapoints with the current smallest margin.

These theoretical considerations suggest that after we have separated the data, we should be able to safely drop training datapoints with large margin. The question now is: how much can we compress, without spoiling the generalization performance? We can see in Figure 4 that gradually removing all but 200 datapoints with the smallest margin has no impact on the test performance for a well performing network (solid red line), if we start removing the data after separation has been achieved. More interestingly, we get a very minimal drop in performance ($\sim 2\%$) when we keep only 20 datapoints in the training set. We find that minima that perform better can be compressed more. Figure 6 compares two minimizers with different test performance: we can readily see that further removal of datapoints leads to more degradation of performance for the network with worse test error, as compared to the better performing network. We thus have a demonstration that, at finite times, larger number of small margin data leads to worse test performance.

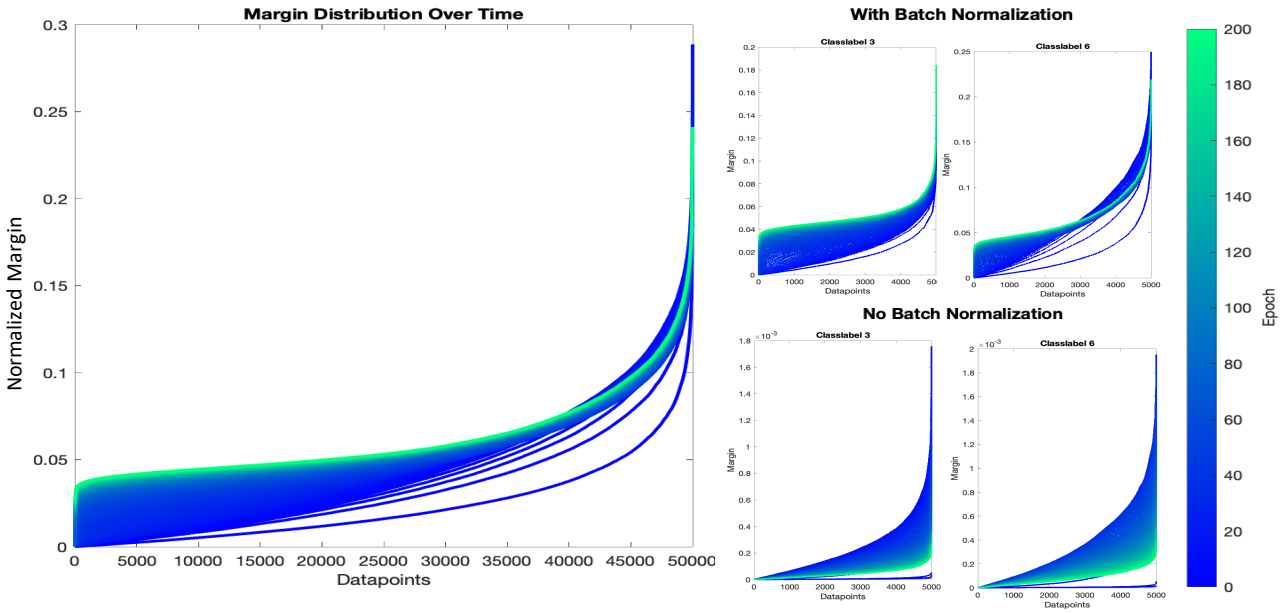

Figure 3. ***Margin Distribution over Time*** *A convolutional network (using the architecture and parameters in Section 3) was trained on CIFAR10, the margin distribution was recorded and sorted at every epoch. On the left, we see the margin distribution of all 50,000 datapoints from epoch 1 (dark-blue) to epoch 200 (light-green) as indicated by the color bar on the far-right. To explore the results in neural collapse, we visualize the margin distribution of individual classes for networks trained with batch normalization and without. Some class-labels seem to have more of an effect on the distribution for that label than others but this is dependent on batch normalization.*

This approach is indirectly related to data distillation that has been studied in (Wang et al., 2018). There, the authors noticed that it is possible to train a CIFAR10 classifier on just 10 synthetic datapoints and achieve 54% accuracy on the test set. Unlike in the case of data distillation however, here we first train the network to the point of separability and by gradually removing datapoints achieve *no significant drop in performance up to keeping* ~ 200 *datapoints*. These results strongly suggest a novel training scheme for speeding up convergence, in which we remove most of the training data (those with large margin) right after reaching separation. Rapid compression of the dataset down to 100 examples hurts performance, but keeping 200 points only leads to $\sim 2.89\%$ reduction of accuracy for our CNN architecture trained with SGD, a batch-size of 50 and a learning rate of 0.01, see Figure 5. A more thorough search of the hyperparameter space reveals that with large learning rates and small batch size, compression of CIFAR10 down to 200 data can lead to drop in performance as low as 0.18% for our CNN and 0.11% for DenseNet-BC, as seen in Table 1. For more details on the experiments, see Sup. Mat.

5.1. Similar Initialization Leads to Different Important Data

It might be tempting to try to understand why certain datapoints seem to drive the dynamics in the terminal phase of training (post data separation). After all, we can see in Figure 4 that minima of different levels of test performance have a totally different set of small margin data. The converse does not seem to be however true, as can be seen in Figure 6, where we initialized the network several times from the same statistical distribution (normal with std 0.01). We find that the initial randomness propagates through the training procedure, leaving us with similarly performing minima (3.54% and 4.65% test errors) with different smallest margin data.

Table 1. Drop (in %) of test performance after compression.

Convolutional Network (CIFAR10) with SGD			
Batch Size \ Learning Rate	10^{-1}	10^{-2}	10^{-3}
200	18.29	25.15	X
100	2.55	22.18	11.810
50	1.68	4.84	11.55
20	0.45	4.21	8.99
10	1.75	0.86	6.48
1	0.18	0.46	1.91

Dense Network (CIFAR10)						
Optimizer	Adam			SGD		
Batch Size \ Learning Rate	10^{-1}	10^{-2}	10^{-3}	10^{-1}	10^{-2}	10^{-3}
200	7.25	2.78	4.11	1.54	5.99	33.31
100	1.88	0.95	2.70	0.72	3.94	3.60
50	0.12	0.85	1.72	0.11	2.38	4.08

On the top of Figure 6, we can see the training and test error of two networks trained on CIFAR2. The vertically vectorized images at each epoch represent the overlapping set of the 20 datapoints with thee smallest margins ("support vectors") between the two networks. Visually, we can see there are not many overlapping datapoints. The bottom plot

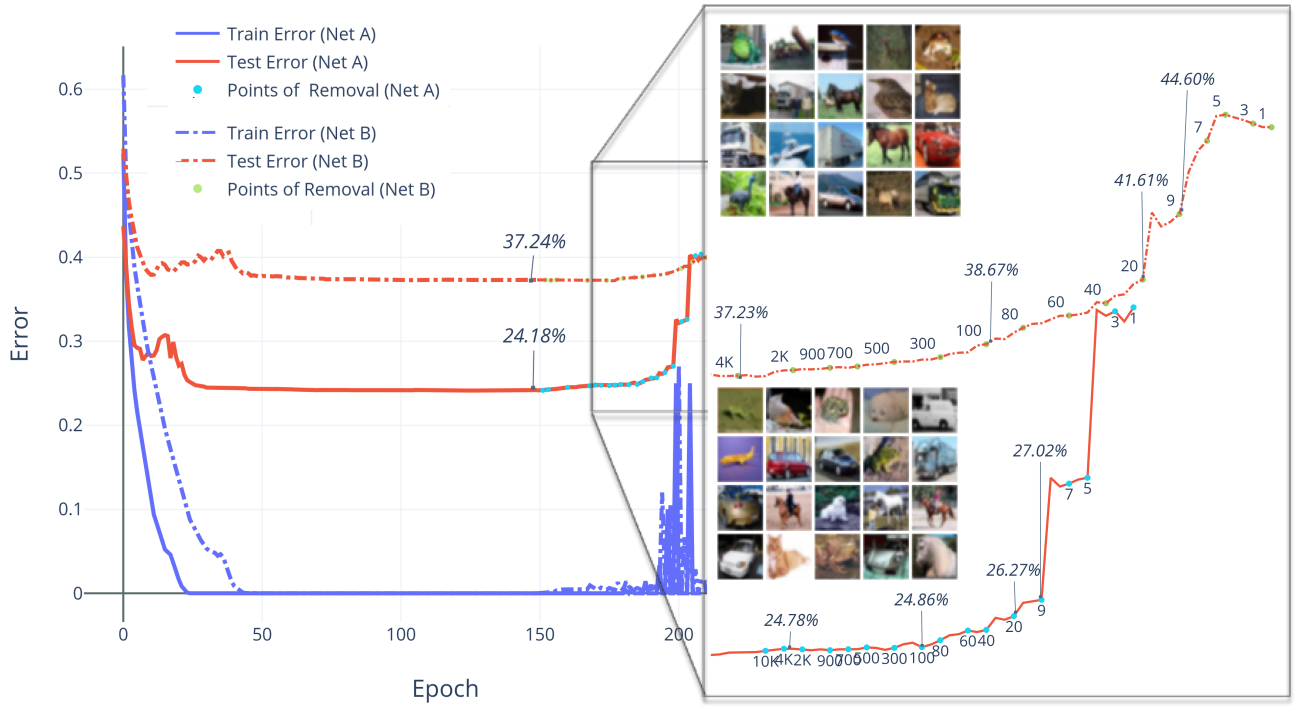

Figure 4. ***Data Points Removal Algorithm*** *Networks (A) and (B) were randomly initialized from a normal distribution with std of 0.01 and 0.09, respectively. This leads the networks to converge to different minima with different test errors ($\sim 13\%$). After data separation and full convergence, the algorithm ranks the datapoints based on their margin size and starts removing datapoints during further training, removing those with highest margins at every point. At every blue and green point, a set of points were removed, with the numbers displaying the amount of datapoints left in the training set. The right side zooms in to show that the test error does not significantly change until 100 datapoints are left (by 0.68% for (A) and 1.43 % for (B)). The test error changes more when there are only 20 datapoints left for both (A) and (B). The figure inserts show these 20 datapoints for both networks. The two sets are different, showing that the networks converged to infima with different support.*

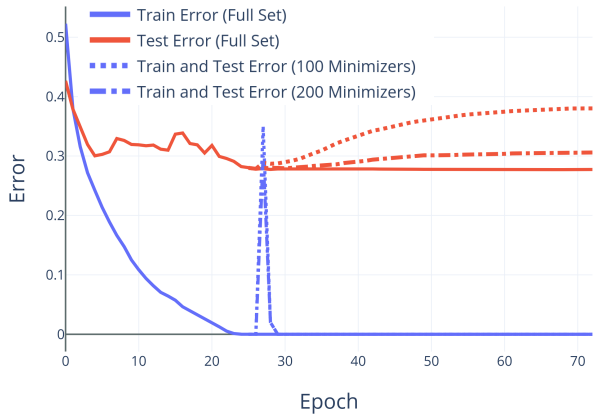

Figure 5. ***Compression After Data Separation*** *During training, right after data separation, datapoints with the large margins were removed, leaving either 100 or 200 datapoints with the smallest margins. When the dataset is compressed to 200 datapoints the test error increases slightly but plateaus to a good test performance for the network architecture (2.89 % change).*

shows the percentage of overlapping datapoints in the 200 smallest margin datapoints of the networks. Although over time the percentage of overlapping data increases, only 63 percent of the same datapoints in this bin of 200 are present in both networks at margin convergence (last epoch).

Not only does the randomness at initialization play a role, but even more importantly, we find that for a given training run, we cannot reasonably predict which data are going to support the dynamics the most until just before data separation takes place, since only 40 percent of the support vectors at margin convergence are present at data separation, see Sup. Mat. Figure 7 shows the margins of 600 datapoints (200 smallest, 200 in the middle and the largest ones) throughout training. We see from this that before data separation happens, it seems impossible to discover using only the margin information which datapoints will end up having the smallest margin.

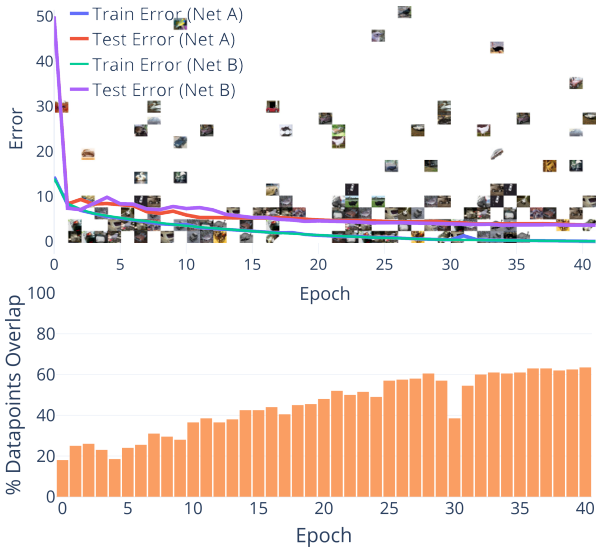

Figure 6. ***Same Initialization - Different small margin datapoints*** *Two networks were initialized from the same margin distribution (standard deviation 0.01) and trained on CIFAR2. Both networks converge to similar minima but the exact datapoints with the smallest margins differ. On the top, the errors are depicted and the vectorized images (cars and birds) represent the overlapped datapoints in the smallest 20 – there are only a few overlapping points. The percentage of overlapping datapoints for the smallest 200 margins are depicted at the bottom (60 % of them overlap).*

6. Discussion and Conclusions

Recent theoretical results (Lyu & Li, 2019; Poggio et al., 2020b; Shpigel Nacson et al., 2019) have shown that gradient descent techniques on exponential-type loss functions converge to solutions of locally maximum classification margin for overparametrized deep networks. In this paper we studied the distribution of margins on the entire training dataset and demonstrated that the area under the distribution is a good approximate measure for ranking different minima of the same network.

Inspired by the recent theory(Poggio & Liao, 2020) predicting various properties of Neural Collapse (Papyan et al., 2020), we investigated the prediction that none of the training data contribute more towards good generalization performance than others, at least asymptotically. We found that while on long timescales the distribution does get flatter, the dynamics effectively depends only on a few datapoints. Once separation sets in, we can successfully compress most of the training datapoints (removing those $\{x_n, y_n\}$ with largest margins from the training set), going down from 50k examples to less than 200, without compromising on performance. In fact, since the property NC1 implies that we could in principle compress the training dataset down to one datapoint per class, very much in line with the distillation results of (Wang et al., 2018). Thus, in the presence of SGD and both batch normalization and L2 regularization

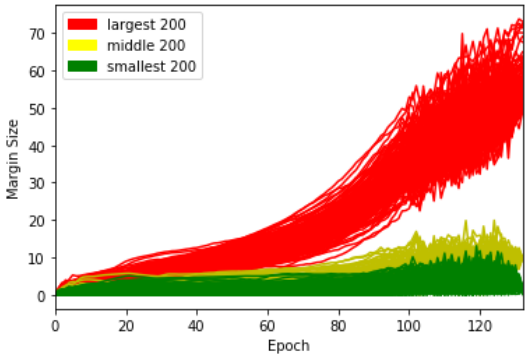

Figure 7. ***Visualization of the 200 smallest, middle and largest margins over time****. Data separation occurs at epoch 40. This network was trained with a batch size of 1 using SGD with a learning rate of 0.01. There is no clear way of predicting which datapoints will have smallest margin before data separation.*

(all three seem important for Neural Collapse to happen), we expect that all data are equally important to classification.

In practice, we find that the compressed dataset we can obtain is highly dependent on the randomness of initialization. An obstacle to effectively predicting a good compressed set is the fact that the relevant datapoints only emerge around the time of data separation. This means that the algorithm for compressing the training dataset does not provide a massive speed boost, as one could hope. More importantly however, these results cast doubt on the endeavour of interpreting the features that a network learns – we can expect that the randomness of which training points contribute the most to the solution of the optimization problem implies that the "high capacity" features most relevant to classification are also random and inconsistent between different training runs.

The results in this article motivate potential more fine-grained investigations into the early pre-separation dynamics – if we could earlier predict the compressed dataset, we would have a way to drastically speed up training. Additionally, it would be interesting to understand why dataset compression is so successful with small batch size and large learning rate – is it connected to the suggestion in (Poggio & Cooper, 2020) that small batch SGD is more likely to find global minima of the loss?

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

References

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. *arXiv preprint arXiv:1811.03962*, 2018.

Arora, S., Ge, R., Neyshabur, B., and Zhang, Y. Stronger generalization bounds for deep nets via a compression approach. *arXiv e-prints*, art. arXiv:1802.05296, February 2018.

Arora, S., Du, S. S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019.

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

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

Bousquet, O., Boucheron, S., and Lugosi, G. Introduction to statistical learning theory. pp. 169–207, 2003.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Du, S. S., Lee, J. D., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. *CoRR*, abs/1811.03804, 2018. URL http://arxiv.org/abs/1811.03804.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1eK3i09YQ.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *arXiv:1609.04836 [cs, math]*, September 2016. URL http://arxiv.org/abs/1609.04836. arXiv:1609.04836.

Kutin, S. and Niyogi, P. Almost-everywhere algorithmic stability and generalization error. Technical report TR-2002-03, University of Chicago, 2002.

Langford, J. and Caruana, R. (not) bounding the true error. In *Advances in Neural Information Processing Systems*, pp. 809–816, 2002.

Li, Y. and Liang, Y. Learning overparameterized neural networks via stochastic gradient descent on structured data. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 8157–8166. Curran Associates, Inc., 2018.

Liao, Q., Miranda, B., Banburski, A., Hidary, J., and Poggio, T. A surprising linear relationship predicts test performance in deep networks, 2018.

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. *CoRR*, abs/1906.05890, 2019. URL http://arxiv.org/abs/1906.05890.

Morcos, A. S., Barrett, D. G., Rabinowitz, N. C., and Botvinick, M. On the importance of single directions for generalization. *arXiv preprint arXiv:1803.06959*, 2018.

Mukherjee, S., Niyogi, P., Poggio, T., and Rifkin, R. Learning theory: stability is sufficient for generalization and necessary and sufficient for consistency of empirical risk minimization. *Advances in Computational Mathematics*, 25(1):161–193, 2006. ISSN 1572-9044. doi: 10.1007/s10444-004-7634-z. URL http://dx.doi.org/10.1007/s10444-004-7634-z.

Papyan, V., Han, X. Y., and Donoho, D. L. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020. ISSN 0027-8424. doi: 10.1073/pnas.2015509117. URL https://www.pnas.org/content/117/40/24652.

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

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

Poggio, T., Banburski, A., and Liao, Q. Theoretical issues in deep networks. *PNAS*, 2020a.

Poggio, T., Liao, Q., and Banburski, A. Complexity control by gradient descent in deep networks. *Nat Commun 11, 1027*, 2020b.

Poggio, T. A. and Liao, Q. Implicit dynamic regularization in deep networks. 08/2020 2020.

Shpigel Nacson, M., Gunasekar, S., Lee, J. D., Srebro, N., and Soudry, D. Lexicographic and Depth-Sensitive Margins in Homogeneous and Non-Homogeneous Deep Models. *arXiv e-prints*, art. arXiv:1905.07325, May 2019.

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

Wang, T., Zhu, J., Torralba, A., and Efros, A. A. Dataset distillation. *CoRR*, abs/1811.10959, 2018. URL http://arxiv.org/abs/1811.10959.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016. URL http://arxiv.org/abs/1611.03530.

Zou, D., Cao, Y., Zhou, D., and Gu, Q. Stochastic gradient descent optimizes over-parameterized deep relu networks. *CoRR*, abs/1811.08888, 2018. URL http://arxiv.org/abs/1811.08888.