What Neural Collapse Teaches us about Generalization in Deep Neural Networks

Chaitanya Agarwal* ca2719
Patrizio Caddeo* pc2736
Haoxiang Huang* hh1987
Yi Yang* yy3731

1 Introduction

Over the past decade, neural networks have advanced the state-of-the-art in computing applications, achieving super-human performance in tasks ranging from image classification to speech recognition to game play. However, this performance is for the most part unexplained by the current theory. Papyan et al. (2020) have recently discovered a new pattern, which they denoted *Neural Collapse*, that emerges across a variety of image-classification datasets and architectures. This newly discovered phenomenon has sparked a lot of interest, giving rise to a whole new line of research which has the intent to unpack the mechanics and causes of this phenomenon and hopefully get some insight into the performance of neural networks.

The work of Papyan et al. (2020) focuses on *overparameterized* networks. This is the regime where interpolating the data is possible and one is able to achieve 0 training error. It is a paradigm of classical statistics and machine learning that perfectly fitting the data can hurt the performance of the model (a phenomenon known as *overfitting*), but surprisingly, neural networks capable of interpolating the data have been shown to perform greatly. In particular, it has been observed that training way past the point of 0 training error can actually be beneficial. We refer to the region of training time after the point where 0 error is achieved as the *terminal phase of training (TPT)*.

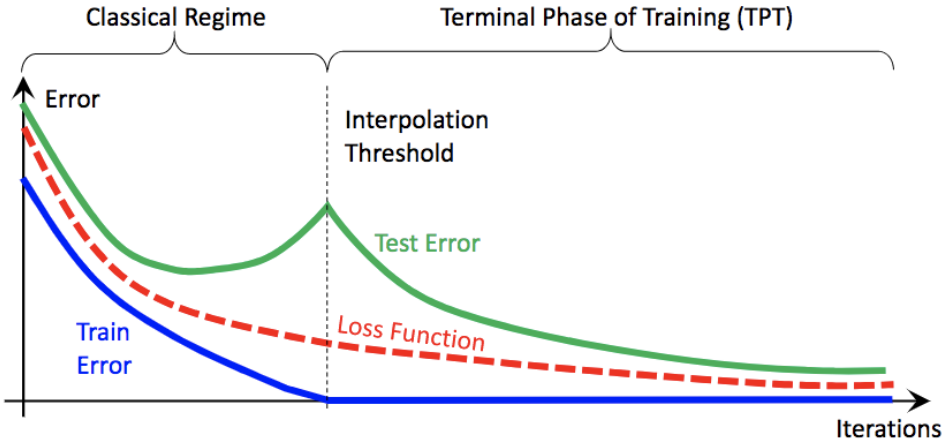


Figure 1: **Phases of training:** the plot shows a common behaviour of overparameterized networks indicating that it can beneficial to train past the 0 training error. The test error experiences *double descent*: it first starts growing when approaching 0 training loss, but then starts decreasing again in the TPT. Elad et al. (2020)

The pattern that Papyan et al. (2020) discovered involves overparameterized classification models with balanced datasets (meaning that the training set has an equal amount of samples for each class) trained in the TPT. Specifically, the authors have observed the following:

(NC1) *Variability collapse*: as training progresses, the within-class variation of the activations becomes negligible as these activations collapse to their class means.

*These authors contributed equally to this work. This is merely an alphabetical ordering.

(NC2) *Convergence to simplex tight frame (ETF)*: the vectors of the re-centered class means converge to having equal length, forming equal-sized angles between any given pair, and being the maximally pairwise-distanced configuration constrained to the previous two properties.

(NC3) *Convergence to self-duality*: despite being mathematically distinct objects, class means and linear classifiers living in dual-vector spaces converge to each other up to rescaling

(NC4) *Simplification to nearest class center (NCC)*: for a given deep net activation, the network classifier converges to choosing the class with the closest train class mean (in standard Euclidean distance).

We will give a formal description of these phenomena in Section 2, but a visual representation is given in Figure 2.

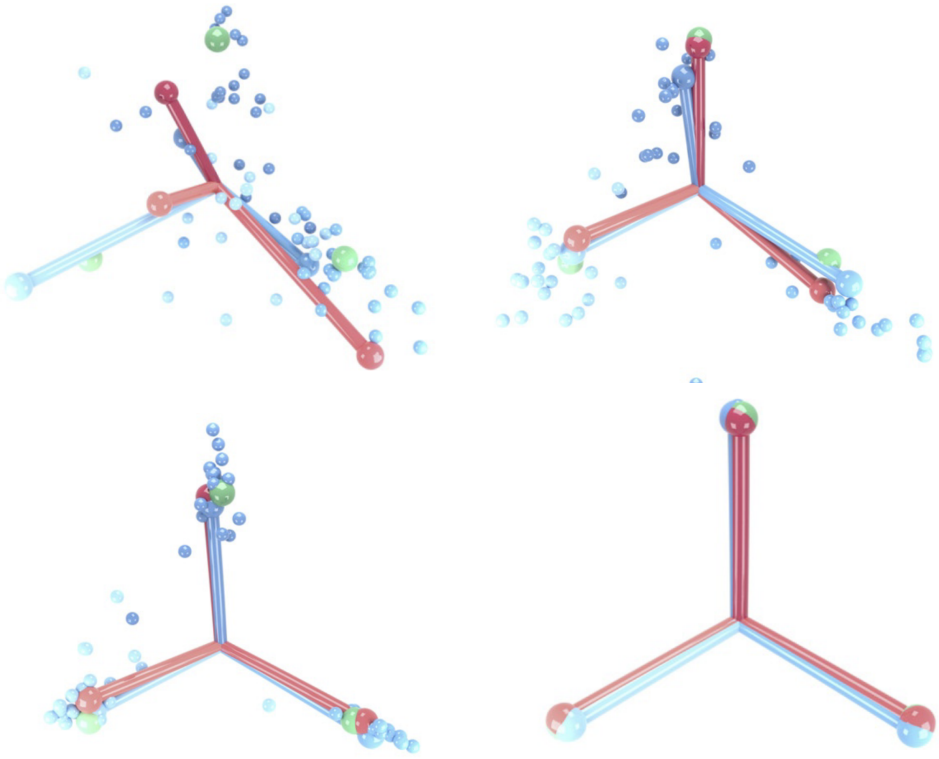


Figure 2: **Visualization of neural collapse:** as training progresses from top to bottom (and left to right), the image illustrates NC in three dimensions. The vertices of the ETF simplex are represented by green spheres; linear classifiers are represented by red balls and sticks; class means are represented by blue balls and sticks; last-layer features are represented by little blue spheres. Different classes are identified by their color shade. Last-layer features collapse into their class means (NC1), class means converge to the vertices of the simplex ETF (NC2), and linear classifiers approach their associated class means as training progresses (NC3). Papyan et al. (2020)

Properties (NC1)-(NC4) indicate that a highly symmetric and rigid mathematical structure concerning both features and classifiers arises during deep learning training across a variety of different architectures and datasets. More specifically: (NC2) indicates that the feature means of training samples corresponding to different classes tend to be equally spaced around the sphere in their constructed feature space, while (NC1) implies that all training samples actually collapse onto these points (i.e. their corresponding means); (NC3) shows that the classifiers align with the aforementioned class means and thus are essentially the same, up to rescaling. This structure has great interpretability and leads to very simple decision rules given by (NC4).

This surprising phenomenon indicates an implicit bias of neural networks towards solutions with a very simple and

rigid structure, and analyzing the mechanism and causes behind the emergence of this pattern can hopefully give theoretical insight into their performance.

While aiming to get a greater understanding of how neural collapse can help to shed light on the mysterious behaviour of neural networks, this survey focuses on the following two questions:

(1) *What consequences does neural collapse have on the generalization performance of the neural network?*

(2) *What does neural collapse imply about the nature, the strengths and the limitations of the network?*

The survey is outlined as follows:

- In Section 2, we set some notation and give a formal description of neural collapse.
- In Section 3, we deal with question (1) by analyzing some of the works in the literature that focus on the generalization aspects of neural collapse.
- In Section 4, we deal with question (2) by giving an information-theoretic perspective on neural collapse and its relation to representation learning.
- In Section 5, we discuss questions related to the previous sections and state open problems relevant to our discussions.

2 Notations and definitions

2.1 Preliminaries

In this paper, $||\cdot||_F$ denotes the Frobenius Norm, $||\cdot||_2$ denotes the standard Euclidean norm for vectors, and $\langle A, B\rangle$ denotes the dot product between two matrices A, B. We denote the set of the first m natural numbers by $[m]$.

Consider a balanced[1] K class classification problem with input dataset $X = \{x_{k,i} : 1 \leq i \leq N, 1 \leq k \leq K\}$, where $x_{k,i} \in \mathbb{R}^d$ ($x_{k,i}$ is the ith sample from the kth class). Then, a standard fully connected L-layers neural network takes the following form:

$$f_n(x_{k,i}, W_{full}) = b_L + W_L\sigma(b_{L-1} + W_{L-1}\sigma(...\sigma(b_1 + W_1x_{k,i})...)), \tag{1}$$

where $W_{full} = (\{W_1, W_2, ..., W_{L-1}, W_L\}, \{b_1, b_2, ..., b_{L-1}, b_L\})$ denotes the weight matrices and the bias terms in each layer and $\sigma(\cdot)$ denotes the non-linear activation function (for example, ReLU or sigmoid). Here, $W_l \in \mathbb{R}^{d_{l-1}\times d_l}$ for $1 \leq l \leq L$, and $b_l \in \mathbb{R}^{d_l}$ for $1 \leq l \leq L$, where $d_0 = d, d_L = K$ and the rest are fixed arbitrary positive integers.

Classification-oriented neural networks are generally viewed as a the composition of a non-linear feature map f and a linear final-layer classifier W. For a given data point $x_{k,i}$, f returns a feature representation $h_{k,i}$ that encodes the inherent features of the given data. For example, if we are given an image of a dog, a feature representation might encode the presence of limbs and dog-like ears, whereas for an image of a car, a feature representation might encode the presence of wheels. This gives an alternate formulation for f_n:[2],

$$\begin{aligned} f_n(x_{k,i}, W_{full}) = \langle h_{k,i}, W_L\rangle, \quad \text{where } h_{k,i} &= f(x_{k,i}, W_{full}) \\ &= \sigma(b_{L-1} + W_{L-1}\sigma(...\sigma(b_1 + W_1x_{k,i})...)). \end{aligned}$$

We assume that for a dataset X, the set of features is given by $H = \{f(x_{k,i}) : 1 \leq i \leq N, 1 \leq k \leq K\}$.

Note that $W_L \in \mathbb{R}^{d_{l-1}\times K}$ (we will omit the subscript and only write W from now on) is essentially a matrix consisting of K classifiers in $\mathbb{R}^d_{l-1}$, and it can thus be represented as $[w_1, w_2, ..., w_K]^\top$.

[1] Neural collapse has mostly been studied for balanced classification problems until now. A notable exception is Fang et al. (2021), who study the problem in an unbalanced setting.

[2] Note that for simplicity, we have omitted the bias term b_L here. One can modify W_L to get $W'_L = [W_L, b_L] \in \mathbb{R}^{d_{L-1}\times d_L+1}$ and modify h to get $h' = [h, 1] \in \mathbb{R}^{d_{L-1}+1}$. This gives the same formulation as above.

2.2 Formalization of Neural Collapse

In this section, we formalize neural collapse with mathematical descriptions of the properties (NC1)-(NC4). We define the last-layer (training) class means as

$$\mu_k = \frac{1}{N}\sum_{i=1}^{N} h_{k,i}, \quad k = 1, ..., K, \tag{2}$$

and the global mean as

$$\mu_G = \frac{1}{K}\sum_{k=1}^{K} \mu_k. \tag{3}$$

Let $\bar{\mu}_k = \mu_k - \mu_G$ be the centered class means, and let $\bar{H} = [\bar{\mu}_1, ..., \bar{\mu}_K]$. Let $W = [w_1, w_2, ..., w_K]^\top$ and $b = [b_1, b_2, ..., b_K]^\top$ be the classifier matrix and bias terms in the last layer.

Note that all the terms defined above depend on the training time t, but we omit this dependence for brevity. The mathematical description of neural collapse (NC1-NC4) is given below, where $\to$ means convergence as training time $t \to \infty$:

(NC1) *Variability collapse*:

$$h_{k,i} - \mu_k \to 0, \quad \forall i \in [N], k \in [K] \tag{4}$$

(NC2) *Convergence to simplex equiangular tight frame (ETF)*:

$$||\bar{\mu}_k||_2 - ||\bar{\mu}_{k'}||_2 \to 0, \quad \forall k, k' \in [K] \tag{5}$$

$$\left\langle \frac{\bar{\mu}_k}{||\bar{\mu}_k||_2}, \frac{\bar{\mu}'_k}{||\bar{\mu}'_k||_2} \right\rangle \to -\frac{1}{K-1}, \quad \forall k, k' \in [K] \text{ and } k \neq k' \tag{6}$$

(NC3) *Convergence to self-duality*:

$$\frac{W^T}{||W||_F} - \frac{\bar{H}}{||\bar{H}||_F} \to 0 \tag{7}$$

(NC4) *Simplification to nearest class center (NCC)*:

$$\arg\max_k \langle w_k, h\rangle + b_k \to \arg\min_k ||h - \mu_k||_2 \tag{8}$$

3 Neural Collapse and Generalization

As we pointed out in the introduction, the simple structure of the learned features and classifiers leads to great interpretability with simple decision rules. However, this would not be of much use if this structure did not also lead to good generalization performance. Hence, the question of whether Neural Collapse is purely an optimization phenomenon, or a genaralization one too, is an important one. In this section, we provide arguments and evidence as to why neural collapse brings benefits to the performance of neural networks, while also pointing out the limits of the current theoretical evidence.

3.1 Test Collapse

A natural question to ask is whether neural collapse on training data induces some form of collapse on unseen data, with the the idea in mind that if unseen data also collapses (to some extent) on the training class means, then we should be able to classify it correctly. We term this new form of collapse as *test collapse*. Naturally, one cannot expect an algorithm trained on a finite subset of the original feature space to capture the entirety of the information

contained in the full space, and hence one should not expect a full collapse in the sense of (NC1) for all possible unseen samples. However, it is intuitively clear that unseen data may concentrate around the collapsed training means in a probabilistic sense, as long as the output of the algorithm is a feature map that has sufficiently nice concentration properties and the dataset is big enough to capture a reasonable amount of information on the original distribution. We will borrow a bound proved by Galanti et al. (2022) to turn this intuitive idea into a quantitative statement. We will also provide some of their empirical results in Section 4.3, in the context of transfer learning.

The authors define a collapse metric for two distributions Q_1 and Q_2 on the feature space $\mathcal{X}$ and a feature map f by

$$V_f(Q_1, Q_2) := \frac{\mathrm{Var}_{Q_1}(f) + \mathrm{Var}_{Q_2}(f)}{2||\mathbb{E}_{Q_1}[f] - \mathbb{E}_{Q_2}[f]||^2} .$$

Given this definition, "test collapse" broadly refers to the quantities $V_f(P_i, P_j)$ being "small" for all classes $i \neq j \in [K]$, where P_l is the conditional distribution of x given that $y = l$. Indeed, the quantity V_f captures the idea that if the variances of two classes are small compared to their relative distance, then the features have "collapsed" onto their corresponding classes and one should be able to classify them with higher accuracy. Also, note that the quantities $V_f(P_i, P_j)$ involve expectations under the general distributions $\{P_l\}$ and thus quantify the variability of potentially unseen data. To relate this to neural collapse on the training set, the authors also define an empirical version of the quantity V_f: given two sets $S_1, S_2 \subset \mathcal{X}$ (to be thought of as samples corresponding to two different classes), the corresponding V_f is defined by taking Q_1 and Q_2 as the uniform distributions over S_1 and S_2:

$$V_f(S_1, S_2) := \frac{\mathrm{Avg}_{x \in S_1}\left[(f(x) - \bar{f}_{S_1})^2\right] + \mathrm{Avg}_{x \in S_2}\left[(f(x) - \bar{f}_{S_2})^2\right]}{2||\bar{f}_{S_1} - \bar{f}_{S_2}||^2} ,$$

where $\bar{f}_{S_i} = \mathrm{Avg}_{y \in S_i}[f(y)]$. We can now state the bound on the test collapse metric: fixing two classes $i, j \in [K]$, one has

$$V_f(P_i, P_j) \leq (1 + A) V_f(S_i, S_j) + B , \tag{9}$$

for some (explicit) terms A and B that depend on i, j and f. We will not write the details of A and B, but we will point out the implicit relevant terms that determine the size of the right hand side (in order to determine how effective the bound is):

(i) First, the right-hand-side has the explicit term $V_f(S_i, S_j)$, while B is proportional to a term of the form $||\mathrm{Avg}_{S_i}[f] - \mathrm{Avg}_{S_j}[f]||^{-1}$ (which also appears in $V_f(S_i, S_j)$ itself). These terms are small if neural collapse occurs on the training set. Indeed, the variances in the numerator of V_f are small by (NC1), whereas $||\mathrm{Avg}_{S_i}[f] - \mathrm{Avg}_{S_j}[f]||$ is lower-bounded by a positive constant if the (training) class means are well-separated, which happens for instance if f converges to a solution satisfying the ETF structure.

(ii) Secondly, both A and B are proportional to terms of the form

$$||\mathbb{E}_{x \sim P_l}[f(x)] - \mathrm{Avg}_{x \in S_l}[f(x)]|| ;$$

$$|\mathbb{E}_{x \sim P_l}[||f(x)||^2] - \mathrm{Avg}_{x \in S_l}[||f(x)||^2]| .$$

These are concentration terms that are generally bound using Rademacher complexities related to the function class $\mathcal{F}$, and usually scale as $O\left(g(\mathcal{F})/\sqrt{n_l}\right)$ for "nice" (e.g. norm-constrained) function classes $\mathcal{F}$. For instance, the authors prove such a bound for ReLU networks with bounded weights. Hence, under the right conditions, one may expect such terms to be small for large training sets.

(iii) Finally, A depends on a term of the form $||\mathbb{E}_{P_i}[f] - \mathbb{E}_{P_j}[f]||^{-1}$. However, note that by triangular inequality $||\mathbb{E}_{P_i}[f] - \mathbb{E}_{P_j}[f]||$ can be lower bounded by $||\mathrm{Avg}_{S_i}[f] - \mathrm{Avg}_{S_j}[f]||$ up to concentration terms as in (ii). Therefore, if the terms in (i) and (ii) are small, then this term is small as well.

Hence, the bound (9) proves that collapse does indeed occur on the test set (in the sense of V_f being small) as a consequence of neural collapse on the training set, as long as the latter is big enough and the algorithm returns "nice" feature maps. While this seems to imply that neural collapse brings good generalization (and, in a sense, it does), we need to acknowledge that something remains unexplained here: none of what we wrote explains *why* the algorithm should return a nice feature map (possibly in relation to the occurrence of neural collapse) without making strong external assumptions. Therefore, whether neural collapse by itself is evidence of some regularity of the model

is yet to be proven. However, we should also stress that the above analysis still brings merits to neural collapse: "regularity" in a model is generally not sufficient to infer quantitative results on robustness or generalization, but the addition of the neural collapse structure does indeed allow such kind of inferences.

3.2 A Max-Margin Perspective

We now present an alternative view of generalization motivated by NC2 and NC3 that focuses on the nature of the classifiers, rather than the nature of the activation means. First, we recall the concept of margin, a classical indicator of generalization, and provide an example illustrating why max-margin classifiers can achieve good generalization. Then, we provide an intuition as to why neural collapse solutions are inherently max-margin classifiers, a characteristic that indicates a strong link to generalization.

3.2.1 Margin-Maximization and Generalization

The concept of margins was first proposed in the classical perceptron analysis in Novikoff (1963). Perhaps, the most popular occurrence of margin-maximization as a heuristic for learning models with good generalization was proposed as a part of the support-vector machines model Cortes and Vapnik (1995). We present the concept below for completeness.

For a given linear classifier $W \in \mathbb{R}^x$, and a linearly separable dataset $Z = \{z_{k,i} \in \mathbb{R}^x : i \in [N], k \in [K]\}$, the margin can be defined as

$$q_{k,i}(W) = w_k^\top z_{k,i} - \max_{j \neq k} w_j^\top z_{k,i}.$$

By extension, the margin for a dataset is defined as the minimum margin over all data points in the dataset, $q(W) = \min_{k,i} q_{k,i}$. The idea behind margin maximization is that classifiers that are able to predict the correct class with greater confidence will increase the distance from the data points to their own classes' classification boundary while increasing the distance from the others. Even if there is sufficient noise, from the training distribution, a big margin could account for this and possibly make the correct prediction. Here, the distance between the point $z_{k,i}$ and a classification boundary for class j is given by $w_j^\top z_{k,i}$. A classifier that can achieve the greatest margin for the given problem is known as the *max-margin classifier*.

Figure 3 gives an example showing why bigger margins can give good generalization[3]. Here, we can see that both H_2 and H_3 can fit the training data very well. However, H_2 doesn't make some of the predictions with great confidence as some points have a very small margin. As shown in the figure, this can cause H_2 to possibly misclassify unseen samples as the underlying distribution of the data intersects the classifier. Comparatively, the training points are classified with greater confidence by H_3 as the points have a much greater margin and it is less likelier to suffer from this problem. In this case, a large margin margin classifier clearly generalizes the data very well.

However, it is important to note that a small margin doesn't necessarily mean bad generalization performance (Bartlett et al. (2017); Jiang et al. (2018, 2020)). Still, classifiers that have big margins have been *usually* known to give good generalization performance (Bousquet et al. (2003), Xu et al. (2021)). We talk more about this later.

3.2.2 Neural Collapse Solutions are Max-Margin solutions

We now explain why neural collapse solutions are inherently max-margin classifiers by showing that the final layer linear classifier is a max-margin classifier over the feature space. We redefine the margin for a linear classifer W, and a feature map $f : X \to H$ as

$$q_{k,i}(W, f) = w_k^\top f(x_{k,i}) - \max_{j \neq k} w_j^\top f(x_{k,i}) \quad \textit{and} \quad q(W, H) = \min_{k,i} q_{k,i}.$$

Recall that the class means form an ETF structure (NC2), i.e., they have the same lengths, and they are equally and maximally pairwise-distanced from each other. In such a structure, classifiers that are aligned with the class-means

[3]Note that this is an example of binary classification and hence, there is only a single classification boundary. K-class classification problem have K boundaries. In this case, the margin is defined as $w^\top x_{k,i} \cdot y_k$, where, w is the classification boundary and y_k denotes the class of the point (± 1).

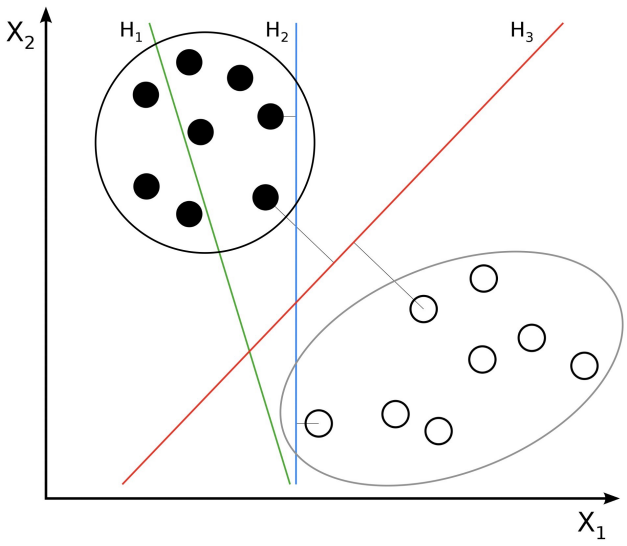


Figure 3: **Max-margin Classifiers.** Various classifers on a binary classification problem with 2 dimensional data (Cyc and Weinberg (2012)). The black and grey regions show the underlying distributions of the classes, while the smaller circles show the training data points. On the training set, H_1 does not separate the classes; H_2 does, but only with a small margin; H_3 separates them with the maximal margin and is the max-margin classifier.

(NC3) give the maximum margin[4]. This has also been supported by empirical evidence as shown in Figure 4. It shows that as training progresses in the TPT, and neural collapse starts emerging, the margin of the dataset also increases.

Notice that, just by scaling W or f, one can still satisfy the neural collapse conditions while obtaining different margins. It must be noted that classification doesn't change as the direction of the individual classifiers is of greater importance in that decision. However, arbitrarily scaling the output before the softmax function could belie the underlying distribution[5], and this could affect other applications that require the confidence behind the predictions as well. This doesn't allow us to look at the margin of the obtained classifier as an absolute metric to evaluate the generalizability of the solution. Thus, neural collapse solutions by themselves don't make a complete statement about generalization. It is also important, that the norm of neural collapse solutions is under control which can be easily achieved by explicit regularization terms.

However, right from the start, we have made an assumption, which is rather benign in most cases, but is very strong in our case. While motivating max-margins in Section 3.2.1, we assumed that max-margin classifiers would be good, *if* the given sample data reasonably represents the underlying distribution (in the case of linearly separable data), which might be true in most cases. In this case, since we are not given a distribution, rather we are learning a distribution in the feature space, neural collapse solutions as max-margin classifiers would be good, *if* the feature representations on unseen data lie around the train class-means. If we aren't able to learn an f that creates feature representations that don't lie around the train class-means, generalization is simply not going to be good. Thus, like in 3.2, we again make a similar statement through a different perspective. A comment on neural collapse and generalization is incomplete without a discussion about why feature maps are able to learn similar representations on unseen data. The presence of neural collapse by itself, without an analysis of the mechanics that determine its emergence, does not yet explain why the learnt feature maps should aide in generalization. We talk more about this in the next section.

[4]See Papyan et al. (2020) Section 6 for a proof. Intuitively, one can imagine that the margin for any point $q_{k,i}(W, f) = w_k^\top f(x_{k,i}) - \max_{j \neq k} w_j^\top f(x_{k,i})$ is maximized when w_k is parallel to $f(x_{k,i})$ such that $w_k^\top f(x_{k,i})$ is maximized, while due to ETF structure, $w_j^\top f(x_{k,i})$ for any $j \neq k$ is minimized.

[5]An output vector [0.2, 0.4, 0.6] after softmax would give probabilities [0.27, 0.33, 0.4]. Whereas an output vector [0.1, 0.2, 0.3] after softmax would give probabilities [0.3, 0.33, 0.37].

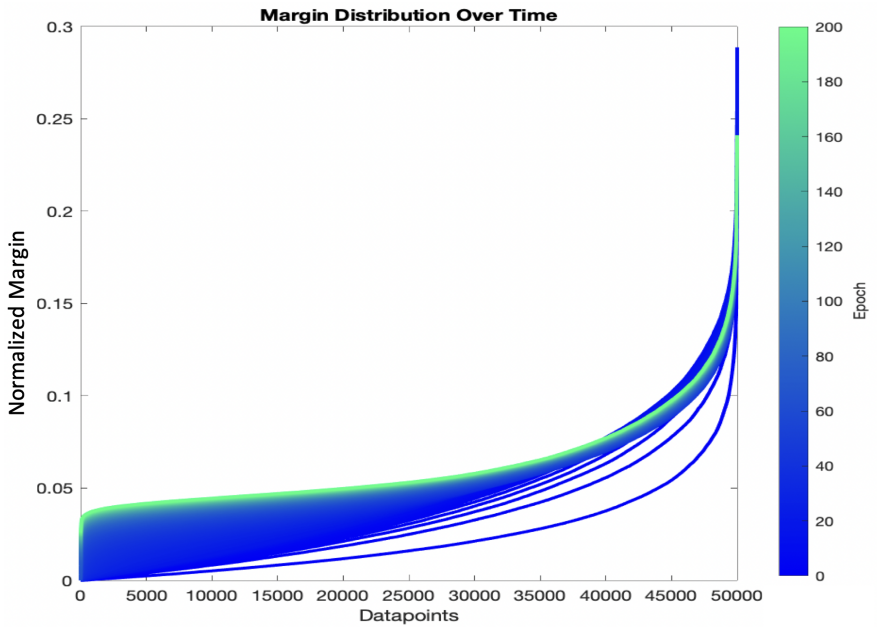


Figure 4: **Margin Distribution over Time.** Banburski et al. (2021) record and sort the margins of all data points as a CNN is trained on the CIFAR10 dataset during the terminal phase of training. One can clearly see that minimum margin increases over time. One can also see the some evidence of the emergence of NC1. As training progresses, margin distribution becomes flatter indicating the collapse of class-means. Figure 9 in Xu et al. (2021) also demonstrates a similar effect.

3.3 The Implicit Bias of Gradient Descent

Even though, neural networks have enough representation power to fit data, even of a completely random nature (Zhang et al. (2016)), their generalization ability is remarkable. This has been linked to good regularization that prevent the humongous amount of training parameters[6] to overfit on training data. Explicit regularization terms that penalize large norms of network layers and various other techniques like dropout, early stopping, data augmentation, etc., have been commonly known to aide in achieving this effect. However, a recent line of research postulates that gradient descent might also be *implicitly* linked to such a regularization. The goal has been to understand how gradient descent or its variants helps deep learning to find solutions with good generalization performance on the test set. For example, research has yielded that gradient descent is biased toward solutions with smaller weights under l2 loss (Li et al. (2018); Vaswani et al. (2020)) and will converge to large margin solution while using logistic loss (Soudry et al. (2017); Chizat and Bach (2020); Ji et al. (2020)). For linear networks, Arora et al. (2019) and Razin and Cohen (2020) have shown that gradient descent determines a low-rank approximation. All these results say that gradient descent produces solutions with good regularization, in the respective problem cases.

These results, along with empirical evidence, clearly suggest the presence of such an implicit bias in deep neural networks as well. For now, we complement the discussion on neural collapse solutions' generalization ability by assuming that good feature maps can generally be learnt due to the implicit bias of gradient descent. But it is important to note that we don't claim any strong links that relate the presence of neural collapse to this implicit bias of gradient descent. However, we don't preclude the possibility and present a weak relation in Section 5.

4 Neural Collapse and Representation Learning

Modern deep neural network architectures are so large and complex that they even manage to easily fit random labels (Zhang et al. (2016)). Being a phenomenon observed in over-parameterized deep neural networks, neural collapse is another evidence of DNN's capability of fitting the data. Due to the huge amount of parameters, the last-layer

[6]Meta recently announced a new deep learning model with 175 billion parameters (Zhang et al. (2022)).

feature maps learned by the network manage to embed high-dimensional data (e.g. images) to a simplex equiangular tight frame of much lower dimension. While this low-dimensional symmetric feature representation is perfect for the succeeding classification task, the within-class variability and structural information of the original data are suppressed. This might sacrifice the network's robustness to corrupted labels or transferability if one wants to use the learned feature representation for different tasks. We enhance these arguments in the following sections.

4.1 Information Bottleneck Principle and Neural Collapse

The prevalence of neural collapse in today's standard deep networks training paradigm suggests that neural collapse may be related to some intrinsic properties of popular deep networks. One way to build this connection is through the *information bottleneck principle*, which we now describe.

One way to interpret DNNs is to view the outputs of intermediate layers h_ℓ as learning effective representations of the input data:

$$X \xrightarrow{f_1} h_1 \xrightarrow{f_2} h_2 \longrightarrow \ldots \xrightarrow{f_L} h_L \longrightarrow \hat{Y} \tag{10}$$

An ideal deep neural network architecture should be able to extract the most efficient informative features related to labels with the most compact architectures (i.e. minimal number of layers, with minimal number of units within each layer). Based on this belief, Tishby and Zaslavsky (2015) hypothesize that the role of a good neural network is to learn some minimal sufficient statistics that predicts the label Y.

More specifically, the authors hypothesize that the network should progressively seek for some data representation h^* that minimizes the mutual information $I(X;h^*)$ between X and h^* while maximizing the mutual information $I(Y;h^*)$ between h^* and label Y, i.e. h^* should be the optimizer of the problem

$$\min_h \mathcal{L}(X,Y,h) = I(X;h) - \beta I(Y;h), \quad \beta > 0. \tag{11}$$

The mutual information $I(X;h)$ is an information theoretic measure that quantifies the amount of information obtained about X by observing h (see for example Cover and Thomas (2006)). The parameter β quantifies the trade-off between two terms: reducing $I(X;h)$ means that the representation h throws away some information about X, which allows it to be more compressed and simple. However, since the label Y and the input X are correlated, it may result in throwing away relevant information about Y, which reduces $I(Y;h)$. Thus, problem (11) should be able to produce an optimal representation h^* that compresses the original data while maximally retaining the information about Y. Note that the depth L of the network allows representations h_ℓ of deeper layers to have more complicated structures, thus giving a better chance to get a good approximation of the optimizer h^* of the problem (11).

To combine, from the first to the last layer, the network should progressively learn optimal representations h_ℓ that minimize the so-called information bottleneck (11). This is referred to as the information bottleneck principle. Under this principle, representations learned at each layer optimally compress representation from the previous layer while retaining the maximal amount of information about label Y.

How does the information bottleneck principle help to explain the role of neural collapse in deep neural network? One of the most important messages of the IBP is that classification deep networks seek the most compact feature representation that explains the label Y. The simplex ETF produced by neural collapse seems to achieve this optimal trade-off: the low-dimensional ETF representation eliminates within-class variability and structural information of data, and merely serves for the purpose of succeeding classification task. This raises the following concern: since the implicit representation learning in the network highly depends on the label Y, the so learned network might sacrifice some robustness to corrupted labels. We will illustrate this point further in the next section.

4.2 Poor robustness to corrupted labels: a comparison to MCRR networks

In this section, we further enhance the argument that neural collapse is an indicator that the network is not robust to label noises. We do so by introducing the work of Chan et al. (2021), which proposes a *MCRR network*, a new design of the neural network architecture where neural collapse is not favored. The paper compares the MCRR network with the commonly used network trained by minimizing the cross-entropy loss. Empirical results suggest that the MCRR network is much more robust to label noises.

We begin with introducing the so-called maximal coding rate reduction (MCRR) principle. Similar to the information bottleneck principle, the MCRR principle proposes a new information theoretic measure.

Given the sample matrix $X = [x_1, ..., x_n]$, the associated labels $Y = [y_1, ..., y_n]$ and a prescribed precision $\epsilon > 0$, the *maximal coding rate reduction principle* aims to seek a representation $Z(\theta) = [z_1(\theta), ..., z_n(\theta)]$ (where $z_i = f(x_i, \theta)$ for some feature map $f(\cdot, \theta)$) that solves

$$\max_{\theta} \Delta R(Z(\theta), \epsilon) = R(Z(\theta), \epsilon) - R_c(Z(\theta), Y, \epsilon). \tag{12}$$

Rather than giving explicit expressions of R and R_c, we prefer to give a verbal explanation of them along with a visualization in Figure 5 (interested readers may refer to Chan et al. (2021) for detailed expressions).

The term $R(Z(\theta), \epsilon)$ is the minimal number of bits needed to encode $Z(\theta)$ up to precision ϵ. As is seen in Figure 5, $R(Z(\theta), \epsilon)$ corresponds to the total number of balls (of radius ϵ). The second term R_c is the average number of bits needed to encode each class. In Figure 5, $Z(\theta)$ lies in the one-dimensional subspace S^1 or S^2, and so in this case R_c is the number of green balls.

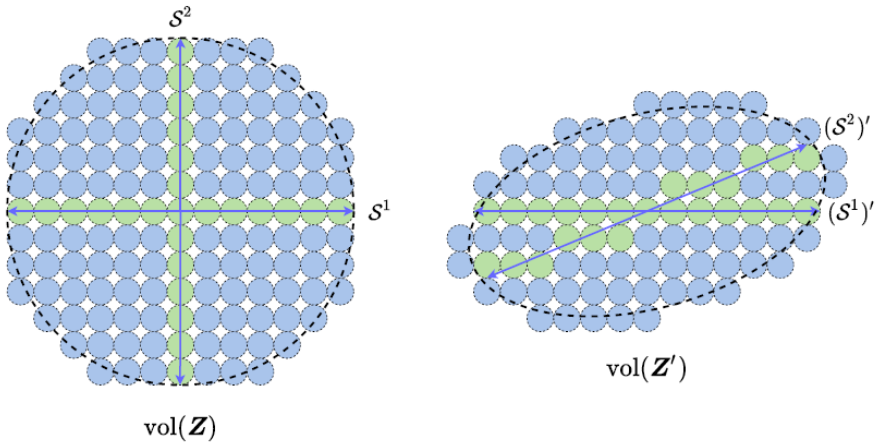
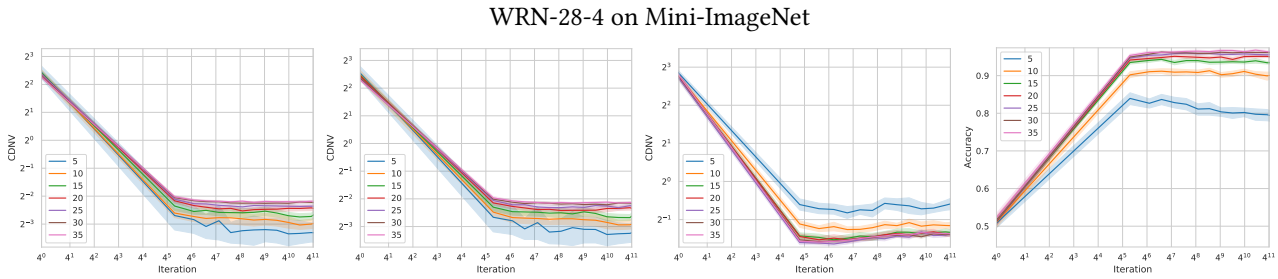


Figure 5: Comparison of two learned representations Z and Z' via reduced rates: R is the number of ϵ-balls packed in the joint distribution and R_c is the sum of the numbers for all the subspaces (the green balls). ΔR is their difference (the number of blue balls). The MCRR principle prefers Z (the left one).

As is pointed out by Chan et al. (2021), the term $R(Z(\theta), \epsilon)$ encourages features of different classes to be maximally separated from each other, hence spanning a space of larger volume. On the other hand, the term $R_c(Z(\theta), Y, \epsilon)$ encourages coherence of the within-class features. Hence each class should only span small volumes. The so learned features follow the basic rule that *similarity contracts and dissimilarity contrasts*. This highly aligns with the philosophy of contrastive representation learning, one of the most powerful approaches in self-supervised learning.

Is neural collapse favored by the MCRR principle? We emphasize that while neural collapse achieves small coding rate of the "parts" $R_c(Z(\theta), Y, \epsilon)$ (since the within-class features collapse to a single vector that belongs to a one dimensional subspace), it also leads to small coding rate for the "whole" $R(Z(\theta), \epsilon)$. This is because the dimension of the simplex ETF is too low to allow diverse representations of the data, and hence it is not the optimal representation under the MCRR principle. As is argued in Chan et al. (2021), the introduction of the term "whole" coding rate $R(Z(\theta), \epsilon)$ can prevent the collapse of features to low-dimensional ETF and produce maximally diverse representations.

Since the MCRR principle focuses more on learning representations rather than fitting labels, it should be expected that DNNs constructed under the MCRR principle are more robust to corrupted labels. We attach here the experimental results given in Chan et al. (2021). The authors apply widely adopted neural network architectures such as ResNet and optimize the network parameters θ to achieve either maximal coding rate reduction in (12) or minimal



(a) train V_f **(b)** test V_f **(c)** target V_f **(d)** target accuracy

Figure 6: **Within-class variation collapse. (a)** Variability collapse on the source training data. **(b)** Variability collapse on the source test data. **(c)** Variability collapse over the target classes. **(d)** Target accuracy rate. In each experiment, the model was trained using SGD on different number of source classes $l \in \{5, 10, 15, 20, 25, 30, 35\}$ (as indicated in the legend). The plots show similar values of train and test collapse for the source data, while the target model gets good collapse and achieves high accuracy when using a large number of classes in the source problem.

cross entropy loss. The table attached below compares the generalization performance of the two networks under different label corruption ratios. It can be seen that while both networks achieve similar test accuracy on clean data, the MCRR network is significantly more robust to label corruptions.

	RATIO $= 0.0$	RATIO $= 0.1$	RATIO $= 0.2$	RATIO $= 0.3$	RATIO $= 0.4$	RATIO=0.5
CE	0.939	0.909	0.861	0.791	0.724	0.603
MCRR	**0.940**	**0.911**	**0.897**	**0.881**	**0.866**	**0.843**

As a final remark, we point out that there are other techniques to improve networks robustness to label corruptions. Early stopping is one such tool (see for example Li et al. (2019)). By stopping early, the algorithm will not enter terminal phase of training, and thus neural collapse will not occur.

4.3 Transferability of Neural Collapse

In machine learning tasks where one has a very limited amount of labeled data, training directly on the available data is unlikely to lead to good generalization. To overcome this problem, an approach that seems to have been very effective is to train features $f(x)$ on a similar (*foundational*) problem where there is much more available data, and then train the learned features on the smaller (*target*) problem using its limited data to hopefully get better performance. This is referred to as *transfer learning*.

In the previous sections, we interpreted neural collapse as an indication of the neural network's tendency to optimize learning *only* for the task at hand while getting rid of inessential information. It is natural to expect that this behaviour may lead to poor transferability. For instance, it is clear that if one tries to solve a target problem by first training a foundational model with *fewer* classes, neural collapse will cause an excessive loss of information and the low-dimensionality of the learned features will make learning in the target problem impossible. For these reasons, we tend to consider neural collapse a non-ideal phenomenon for transferability.

However, it must be noted that transfer learning has proven to be a very successful technique in applications, and this seems to conflict with the fact that neural collapse also appears to be a common pattern in standard networks. Hence, while we can argue that networks working under principles that encourage representation learning may indeed work better in the context of transferability, we should also wonder if neural collapse is *actually* bad in this regard. For instance, how does neural collapse behave if we use a foundational model that involves a much larger amount of classes compared to the target problem?

Galanti et al. (2022), whose work we have already partially analyzed in Section 3.1, deal with these questions and argue that neural collapse can actually be beneficial for transfer learning, under the appropriate conditions. Indeed, the authors prove that the test collapse phenomenon we talked about in Section 3.1 can actually extend to *unseen classes*. We briefly present their results using the same notations as in Section 3.1.

To model the transfer learning setting, the authors assume that both the source classes (those corresponding to the foundational model) and the target classes are "samples" from a collection of classes $\mathcal{C}$ endowed with some distribution $\mathcal{D}_{\mathcal{C}}$ (and as a technical note, they also identify each class $i \in \mathcal{C}$ with the corresponding class-conditional distribution P_i on $\mathcal{X}$). Then, the authors prove a bound of the following form: with high probability over the choice of the source classes $\{\tilde{P}_1, \dots, \tilde{P}_l\}$, one has

$$\mathbb{E}_{P_i \neq P_j \sim \mathcal{D}_{\mathcal{C}}}[V_f(P_i, P_j)] \leq \text{Avg}_{i \neq j \in [l]} \left[V_f(\tilde{P}_i, \tilde{P}_j) \right] + \mathcal{E} \,,$$

where $\mathcal{E}$ is a generalization error term that depends on quantities with a similar flavour as those in the bound on test collapse (9). This bound aims to say that if neural collapse happens on a test-level for the source classes (in the sense that the quantities $V_f(\tilde{P}_i, \tilde{P}_j)$ are small), then one can expect neural collapse to occur on unseen classes as well. In order for this to happen though, the generalization term $\mathcal{E}$ needs to be small. As we mentioned, this quantity depends on terms similar to those appearing in the points (i) and (iii) below the bound (9), but with an important difference: while one can still deal with terms like those in (i) in terms of properties of the network architecture, the same cannot be said for the terms in (iii), as concentration properties for potentially unseen classes cannot generally be inferred from those of the source classes. This is to be expected: transfer learning cannot work if there is no relation or similarity between the source and the target classes. Hence, in order to obtain bounds on the terms in question, one needs to assume that the output of the algorithm (i.e. the feature map) generally keeps the classes apart. The authors provide some conditions that ensure that this holds, but we do not pursue the details here and we refer to the original work instead.

We report some of their empirical results in Figure 4.3, which shows both successful test collapse and transferability.

In conclusion, while the success of transfer learning still depends on the artificial and empirical selection of a reasonable foundational problem for the task at hand, there is some ground to believe that neural collapse can provide an explanation for the good performance of the procedure since the collapse can successfully transfer between source and target problem.

5 Discussion and Open Problems

We have presented a discussion on the recently discovered neural collapse phenomenon, and what it entails in the context of generalization. We have presented arguments that establish links between neural collapse and generalization. However, at the same time, we address gaps that the neural collapse phenomenon fails to explain. We present this as an open problem later and invite more research on the issue. Further, we see that neural collapse is an indicator that the representation learning conducted by network is highly label oriented, which harms network's robustness to corrupted labels and transferrability. We further present more open problems that can possibly solidify links between neural collapse and generalization.

Effect of Neural Collapse on Middle Layers In Section 3, we provided arguments as for why the inherent structure of neural collapse solutions implies classical generalization properties such as margin-maximization. However, we only analyzed the consequences of neural collapse on the final layer of the network. Some aspects of our analyses relied on the assumptions that neural networks are somehow able to learn feature maps that

(1) maximizing the margin of the solutions; (2) simplifying the nature of the solution. However, we show that the neural collapse phenomenon is only restricted to classification in the feature space. The analysis relies on the assumption that neural networks are "somehow" able to learn feature maps that are able to generalize well. A more complete link to establish the relation of neural collapse phenomenon to generalization would require an analysis of what happens in the middle layers of the neural networks. In particular, is it possible that neural collapse imposes a similar, or any unusual structure on the middle layers of the networks? Hui et al. (2022) present a preliminary

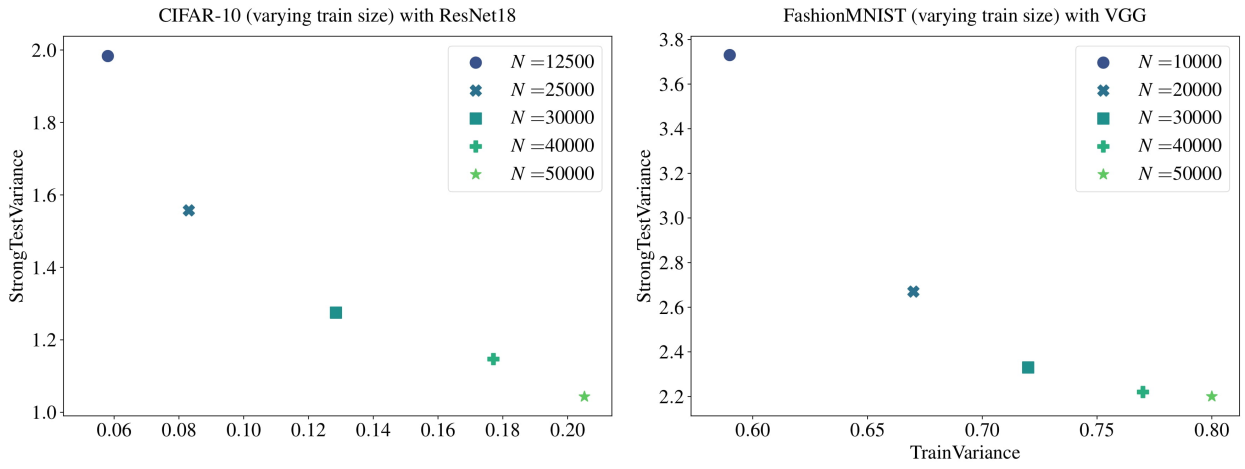


Figure 7: **Train vs test apparent anti-correlation.** Relation between train and test collapse for different sizes of the dataset. All models are trained past 0 training error up to reaching a training loss of 10^{-6}. *Hui et al. (2022)*

experiment in this direction and show the evidence of "Cascading collapse" in the middle layers. Particularly, the feature variance seems to monotonically increase as we go to earlier layers. Further, Ben-Shaul and Dekel (2022) also independently observe the NC4 effect in earlier layers. Further analysis could help us understand the impact of neural collapse on the learnt feature maps and help us to solidify the assumptions we made in our analysis in Section 3.

Neural Collapse and the Implicit Bias of Gradient Descent: In Section 3.3 we hinted at a weak link between the implicit bias of gradient descent towards good generalization and neural collapse. Ji et al. (2022) do a landscape analysis on certain toy models to show that gradient descent has an implicit bias towards neural collapse solutions, even in the absence of any form of regularization. Unregularized solutions are expected to diverge to infinity, and yet, the authors show that their convergent direction is towards a neural collapse solution. Zhou et al. (2022) show a similar result for regularized models. This still does not constitute a direct link between neural collapse and generalization. However, if we hold on to the popular belief that the gradient descent implicit bias is largely responsible for the excellent generalization performance of neural networks, then we may posit that neural collapse solutions should bring good generalization.

Empirical evidence of Test Collapse: Theoretical and empirical evidence regarding generalization in the context of neural collapse is limited. The work by Galanti et al. (2022) we analyzed in sections 3.1 and 4.3 provides theoretical and empirical evidence that neural collapse helps generalization, but we have also pointed out the limited nature of their results. One of the few other works that deal with this topic is the one by Hui et al. (2022), which actually goes in the opposite direction: the authors argue that neural collapse may actually be *not good* for generalization. We believe, however, that the arguments in their work have flaws: the authors claim that the right asymptotic to consider when trying to determine if test collapse occurs is to take a fixed dataset size n and send the training time t to ∞; then, they argue that collapse does not occur because the relative metric (similar to the V_f we defined in Section 3.1) is not seen to vanish in practical experiments. However, as we noted at the beginning of Section 3.1, one should not expect a *full* collapse to occur on the test set, because an algorithm trained on a finite set cannot capture all the information on the original distribution. While the authors note this too, they still insist on dismissing test collapse on this basis, without acknowledging that sending the training size n to ∞ is *also* a meaningful asymptotic to determine test collapse (and an essential one if one wants to experience a "full" collapse).

As a final note, we point out that the authors also provide empirical evidence of some kind of "anti-correlation" between train and test collapse, as shown in Figure 7. First, let us point out that the authors reported higher values of train collapse in models with bigger datasets simply because they trained all models til reaching a common training loss value of 10^{-6} (and clearly bigger datasets are expected to collapse more slowly). With that said, it is legit to wonder why bigger values of train collapse correspond to lower values of test collapse, but we argue that these

results are actually consistent with our analysis in Section 3.1. Indeed, the lower values of the test collapse were observed for the biggest datasets, for which we know by the bound (9) that train and test collapse are supposed to be very close to each other. Hence, this apparent anti-correlation can be explained by the test collapse decaying (and approaching the train collapse) fast as the dataset size increases.

Further comment on Transfer Learning: As was discussed in section 4, neural collapse pushes the last-layer features to a low-dimensional simplex ETF. Such last-layer representation eliminates within-class variability and suppresses structural information of the data, possibly leading to poor transferability. On the other hand, the intermediate layer features may not yet collapse and still retain some within-class variability. This leads to the following question: could some intermediate layers' feature maps be more suitable for transfer learning tasks?

Practical Side-Benefits: Besides studying the direct consequences of neural collapse on generalization, it is also important to consider how neural collapse could inspire solutions that lead to better generalization. We cite two recent works that argue that due to the presence of neural collapse, one could be allowed to make certain assumptions on the network and reduce the complexity during training. We expect more such innovative solutions in the future.

Zhou et al. (2022) say that due to the fact that one would expect NC2 and NC3 to happen, one does not need to explicitly train a final-layer classifier. This could result in a decrease in training time and memory usage, consequently leading to better performance. Zhou et al. (2022) show that for popular architectures like ResNet, DenseNet and ShuffleNet, if one fixes the final-layer classifiers to an ETF structure, one could reduce the number of trainable parameters by 8 to 50% without degradation in testing accuracy. Moreover, they also show that due to the fact that one would expect NC1 and NC2 to happen, one does not need to have more than K dimensions for the final layer activation output of the data. Zhou et al. (2022) show that one could reduce the number of trainable parameters by 8 to 20% by making this assumption without degradation in performance.

Further, Banburski et al. (2021) propose that after a time feature maps are learned well enough and data separation is achieved, since one expects all train points to eventually have the same margin (by (NC1)), training on all the samples might not be needed. They propose an algorithm that removes several training samples as training progresses which remarkably does not affect testing accuracy a lot. This can potentially save a lot of training time.

References

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization, 2019. URL https://arxiv.org/abs/1905.13655.

Andrzej Banburski, Fernanda De La Torre, Nishka Plant, Ishana Shastri, and Tomaso Poggio. Cross-validation stability of deep networks. Technical report, Center for Brains, Minds and Machines (CBMM), 2021.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

Ido Ben-Shaul and Shai Dekel. Nearest class-center simplification through intermediate layers, 2022. URL https://arxiv.org/abs/2201.08924.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Summer school on machine learning*, pages 169–207. Springer, 2003.

Kwan Ho Ryan Chan, Yaodong Yu, Chong You, Haozhi Qi, John Wright, and Yi Ma. Redunet: A white-box deep network from the principle of maximizing rate reduction. *CoRR*, abs/2105.10446, 2021. URL https://arxiv.org/abs/2105.10446.

Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss, 2020. URL https://arxiv.org/abs/2002.04486.

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, USA, 2006. ISBN 0471241954.

Cyc and Zack Weinberg. Support-vector machine. *Wikipedia*, 2012. URL https://en.wikipedia.org/wiki/Support-vectormachine.

Michael Elad, Dror Simona, and Aviad Aberdam. Anothersteptowarddemystifyingdeepneural networks. *CoRR*, 2020. URL https://doi.org/10.1073/pnas.2018957117.

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. URL https://arxiv.org/abs/2101.12699.

Tomer Galanti, András György, and Marcus Hutter. On the role of neural collapse in transfer learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=SwIp410B6aQ.

Like Hui, Mikhail Belkin, and Preetum Nakkiran. Limitations of neural collapse for understanding generalization in deep learning. *arXiv preprint arXiv:2202.08384*, 2022.

Wenlong Ji, Yiping Lu, Yiliang Zhang, Zhun Deng, and Weijie J Su. An unconstrained layer-peeled perspective on neural collapse. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=WZ3yjh8coDg.

Ziwei Ji, Miroslav Dudík, Robert E. Schapire, and Matus Telgarsky. Gradient descent follows the regularization path for general losses, 2020. URL https://arxiv.org/abs/2006.11226.

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.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *ICLR*, 2020. URL https://arxiv.org/abs/1912.02178.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. *CoRR*, abs/1903.11680, 2019. URL http://arxiv.org/abs/1903.11680.

Yuanzhi Li, Tengyu Ma, and Hongyang Zhang. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pages 2–47. PMLR, 2018.

Albert B Novikoff. On convergence proofs for perceptrons. Technical report, STANFORD RESEARCH INST MENLO PARK CA, 1963.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *CoRR*, abs/2008.08186, 2020. URL https://arxiv.org/abs/2008.08186.

Noam Razin and Nadav Cohen. Implicit regularization in deep learning may not be explainable by norms, 2020. URL https://arxiv.org/abs/2005.06398.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data, 2017. URL https://arxiv.org/abs/1710.10345.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. *CoRR*, abs/1503.02406, 2015. URL http://arxiv.org/abs/1503.02406.

Sharan Vaswani, Reza Babanezhad, Jose Gallego, Aaron Mishkin, Simon Lacoste-Julien, and Nicolas Le Roux. To each optimizer a norm, to each norm its generalization. *arXiv preprint arXiv:2006.06821*, 2020.

M. Xu, A. Rangamani, A. Banburski, Q. Liao, T. Galanti, and T. Poggio. Dynamics and neural collapse in deep classifiers trained with the square loss. *CBMM*, 2021.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530, 2016. URL http://arxiv.org/abs/1611.03530.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. URL https://arxiv.org/abs/2205.01068.

Jinxin Zhou, Xiao Li, Tianyu Ding, Chong You, Qing Qu, and Zhihui Zhu. On the optimization landscape of neural collapse under mse loss: Global optimality with unconstrained features. *arXiv preprint arXiv:2203.01238*, 2022.