MATHEMATICAL GUARANTEES OF NEAR-OPTIMAL CROSS-ENTROPY LOSS THROUGH NEURAL COLLAPSE

A Dissertation
Presented to
The Academic Faculty

By

Leyan Pan

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Computer Science
College of Computing

Georgia Institute of Technology

May 2023

MATHEMATICAL GUARANTEES OF NEAR-OPTIMAL CROSS-ENTROPY LOSS THROUGH NEURAL COLLAPSE

Thesis committee:

Dr. Santosh Vempala
College of Computing
Georgia Institute of Technology

Dr. Yongxin Chen
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Sahil Singla
School of Computer Science
Georgia Institute of Technology

Date approved: April 30, 2020

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to the individuals who have made invaluable contributions to my Master's Thesis:

First and foremost, I would like to express my deepest appreciation to Prof. Santosh Vempala, my Committee Chair and Advisor, for his continued and patient guidance on my research methodology, mathematical inspirations, and problem formulation. Prof. Vempala's expertise and insights have been critical to the success of my thesis research, and I am incredibly grateful for his unwavering support throughout this journey.

I would also like to thank Prof. Sahil Singla and Prof. Yongxin Chen, my Committee Members, for their invaluable advise and insightful questions during the committee meetings.

Additionally, I am deeply grateful to Xinyuan Cao and Yifan Wang, my fellow researchers and friends, for their helpful theoretical discussions and for introducing me to other researchers and professors in the field. Their support and encouragement have been greatly appreciated, and I am thankful for their contributions to my academic and personal growth.

Finally, I would like to thank my parents and other friends for their continued mental support throughout my academic journey. Their love and encouragement have been a constant source of motivation, and I am grateful for their unwavering belief in me.

Thank you all for your contributions and support.

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Neural Collapse [1] is a recently observed geometric structure that emerges in the final layer of neural network classifiers. Specifically, Neural Collapse states that at the terminal phase of neural networks training, 1) the intra-class variability of last layer features tends to zero, 2) the class feature means form a simplex Equiangular Tight Frame (ETF) up to scaling, 3) last-layer class features and weights becomes equal up the scaling, and 4) classification behavior collapses to the nearest class center (NCC) decision rule. This thesis investigates the emergence of Neural Collapse for unbiased neural network classifiers when the training cross-entropy loss is near-optimal. Theoretically, we show that, within a small neighborhood of the optimal cross-entropy loss, with C target classes, neural network classifiers with batch normalization and weight decay achieve intra-class feature cosine similarity near one and inter-class cosine similarity near $-\frac{1}{C-1}$, which justifies multiple observations of Neural Collapse under realistic conditions guaranteed by neural network training. Our theorems also imply that higher weight decay values imply more significant Neural Collapse and that batch normalization with affine transformation along with weight decay is critical to the emergence of Neural Collapse, which is supported by experiments. We also empirically investigate the effect of various parameters on the inter- and intra-class cosine similarity of representations obtained.

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Introduction

Over the past decade, deep learning has revolutionized the field of machine learning and artificial intelligence, enabling machines to perform complex tasks previously thought to be beyond their capabilities. At the core of this revolution are deep neural networks, computational models designed to imitate the structure and function of neurons in the human brain. These networks have demonstrated impressive performance in tasks such as image and speech recognition, natural language processing, as well as content generation. Recent claims suggest that neural networks may have achieved sparks of human-level general intelligence. [2] However, despite tremendous empirical advances, a comprehensive theoretical and mathematical understanding of the success behind neural networks, even for the simplest types, is still unsatisfactory. Analyzing Neural Networks using traditional statistical learning theory has had little success due to the high level of non-convexity, over-parameterization, and optimization-dependent properties.

1.2 Neural Collapse

Papyan et al. [1] discovered an elegant mathematical structure in multiple successful neural network-based visual classifiers and named the phenomenon "Neural Collapse." Specifically, Neural Collapse [1] is a geometric structure observed at the terminal phase of training deep neural network classifiers on the penultimate layer features and weights. In its most simplified form, a vanilla neural network classifier is composed of several layers of linear transformation followed by a non-linear activation function. The final output of the network is a vector of dimension equal to the number of classes in the classification task that represents the scores for each class. The basic architecture of a vanilla neural network is illustrated in Figure 2.1

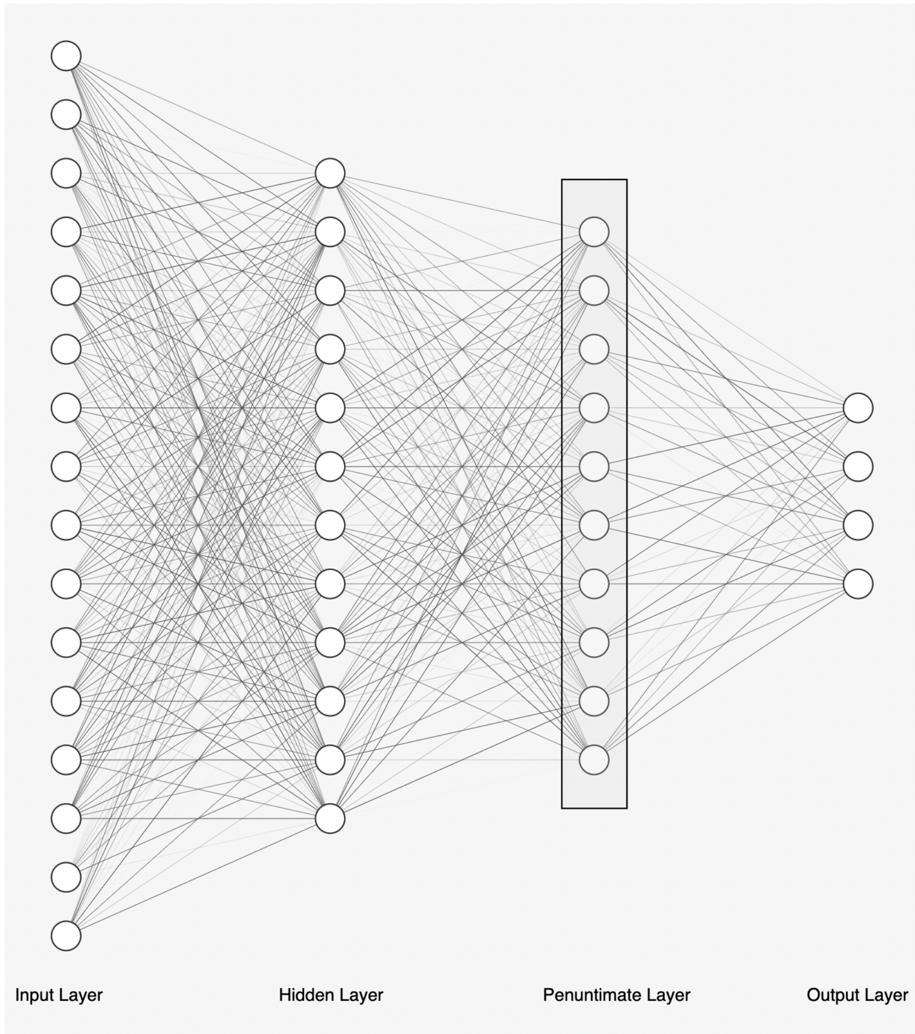

Figure 1.1: Illustration of a vanilla Neural Network Classifier. Each node represents an entry in the corresponding vector of that layer, and each edge denotes a linear weight of the linear transformation of the layer. The penultimate layer is illustrated using a grey box

The penultimate/last layer features of a neural network classifier refer to the feature vectors immediately before linear transformation that computes the network output, as illustrated in the figure as a grey box. Neural Collapse states that after sufficient training of successful neural networks:

(NC1) The intra-class variability of the last-layer feature vectors tends to zero, and individual feature vectors collapse to their corresponding class mean. *(Variability Collapse)*

(NC2) The mean class feature vectors become equal-norm, and every pair of class mean vectors becomes maximally separable, forming a Simplex Equiangular Tight Frame (ETF) around the center up to re-scaling. *(Convergence to Simplex ETF)*

(NC3) The last layer weight vectors converge to match the feature class means up to re-scaling *(Self-Duality)*

(NC4) The last layer of the network behaves the same as a "Nearest Class Center" decision rule, assigning the label of the nearest class mean vector during training to every sample feature. *(Convergence to NCC)*

The ETF structure is a set of maximally separable vectors in space defined as follows:

Definition 1.2.1. *(Equiangular Tight Frame [1]) An ETF structure is a set of vectors in* $\mathbb{R}^d$ *defined by the columns of* $\mathbf{M} = \alpha \mathbf{U} \mathbf{M}^*$*, where* $\alpha \in \mathbb{R}^+$ *is the scale vector,* $\mathbf{U} \in \mathbb{R}^{d \times C}$ *with* $d \geq C$ *is a partial orthogonal matrix (*$\mathbf{U}^\intercal \mathbf{U} = \mathbf{I}$*), and* $\mathbf{M}^*$ *is the standard simplex ETF defined as*

$$\mathbf{M}^* = \sqrt{\frac{C}{C-1}} \left(\mathbf{I}_C - \frac{1}{C} \mathbf{1}_C \mathbf{1}_C^\intercal \right)$$

where $\mathbf{I}_C$ *is the identity matrix and* $\mathbf{1}_C$ *is the vector of all* 1*'s.*

Note that for any ETF of C vectors with $d \geq C$, the cosine similarity between any 2 vectors is $-\frac{1}{C-1}$.

The following Figure 2.1 from Papyan et al. [1] demonstrates the behavior of last-layer feature vectors throughout the training process: Initially, the last-layer feature vectors are scattered across space. As the training progresses, both the mean class vectors and the weight vectors gradually converge to the simplex vertices, and individual feature vectors from the same class become indistinguishable. These observations of Neural Collapse reveal compelling insights into the symmetry and mathematical preferences of over-parameterized neural network classifiers. Subsequently, further work has demonstrated

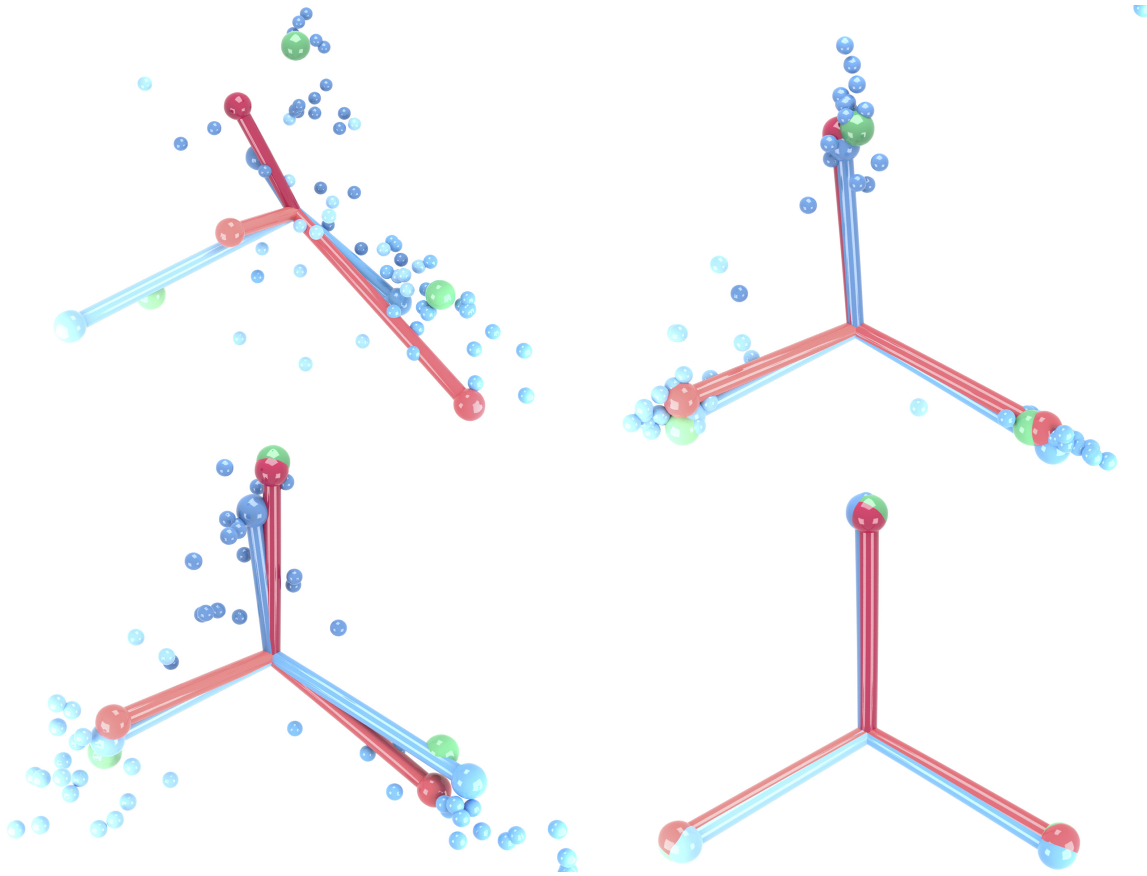

Figure 1.2: Visualization of NC from [1]. The figure depicts, in three dimensions, NC as training proceeds from top to bottom and left to right. Green spheres represent the vertices of the standard simplex, see Definition (Simplex ETF), red balls and sticks represent linear classifiers, blue balls, and sticks represent class means, and small blue spheres represent last-layer features. Different classes are distinguished via the shade of the color. As training proceeds, last-layer features collapse onto their class means (NC1), class means converge to the vertices of the simplex ETF (NC2), and the linear classifiers approach their corresponding class means (NC3).

that Neural Collapse may play a significant role in the generalization, transfer learning [3], model complexity minimization[4], and implicit bias of neural networks [5].

1.3 Releated Theoretical Works on Neural Collapse

On the theoretical side, many recent works have studied the unconstrained features model or layer-peeled model of Neural Collapse [6] [7] [8] [9] initially proposed by Mixon et al. [10], where the last layer features are treated as free optimization variables. Such simplification is based on the observation that most modern neural networks are highly over-parameterized and are capable of learning any feature representations.

In the work of Mixon et al. [10], the authors theoretically analyze the gradient flow with respect to the weight and feature vectors of the unconstrained feature model by solving an approximate linear differential equation with respect to the empirical MSE loss. Their results under this model show that following the gradient flow for both the weight and feature vectors would converge to a subspace that satisfies the properties of Neural Collapse.

Similarly, the theoretical analysis of Han et al. [7] focuses on the gradient dynamics along the "central path" of MSE loss where the last layer weight matrix is assumed to match that of the Ordinary Least Squares regressions. By analyzing the signal-to-noise ratio matrix, they show that the continuously re-normalized gradient flow along the central path converges to the Neural Collapse Structure.

The work of Lu and Steinerberger [8] theoretically demonstrates the simplex ETF structure as the optimal structure for the cross-entropy loss under the constraints of unit-norm feature and weight vectors. While this fundamental work provides significant insights into why cross-entropy loss induces the Neural Collapse structure, the optimal loss is rarely achieved in real neural network training processes, and their work does not provide theoretical guarantees when the loss is non-optimal.

The concurrent recent works of [9] [11] analyze the optimization landscape of the unconstrained features model under Cross-Entropy loss. Their works demonstrate that the final loss as a function of the last layer weight and feature vectors have no spurious local minimum and satisfies the strict saddle property, and any first-order optimization method converges to the global minimum of the loss function, thus implying a Neural Collapse structure.

However, all the above works consider the idealized setting where gradient-based optimization is performed directly on the last layer features. However, this assumption is unrealistic since optimizing the weights in earlier layers can have a significantly different effect from directly optimizing the last-layer features, even in over-parameterized networks. Additionally, Xu et al. [12] also theoretically justifies the Neural Collapse structure for binary classification using their analysis of the training dynamics of MSE loss. However, their work relies on the unrealistic assumption that the gradient of the loss of every sample with respect to the weights is exactly zero. For a comprehensive survey of the recent works on Neural Collapse, see [13].

1.4 Our Contributions

In this thesis, we mathematically investigate the effect of near-optimal cross-entropy loss on Neural Collapse. We show that any neural network classifier with batch normalization and weight decay with near-optimal cross-entropy loss must closely resemble the Neural Collapse structure. Specifically, we mainly focus on the average intra-class feature cosine similarity of each class and the average inter-class feature cosine similarity of each pair of classes. Our work answers the question: **is Neural Collapse guaranteed in every network with low cross-entropy training loss?** Note that showing average intra-class cosine similarity near 1 for all classes is equivalent to NC1 without norm equality, and showing average inter-class similarity for most pairs of classes is equivalent to NC2 without norm equality. We also analyze the bound to show that larger weight decay along with final layer batch normalization provides stronger guarantees on the Neural Collapse structure as long as the model is able to achieve a close-to-optimal loss, which is also supported by empirical experiments.

Lastly, we perform further experiments to investigate the effect of various parameters on the emergence of Neural Collapse and other related phenomenons by measuring intra-class and inter-class cosine similarity and propose conjectures that can guide future theoretical works. Specifically, we show that Neural Collapse is most significant in models with both (affine) batch normalization and weight decay, which matches theoretical predictions. We also show that a training dataset with relatively lower distribution complexity can achieve more significant Neural Collapse after training. Finally, we investigate the presence of neural collapse in deeper layers of trained networks. We show that Neural Collapse becomes sharper with greater model depths, and higher weight decay values strengthen the "Depth Minimization" phenomenon observed by Galanti et al. [4]

CHAPTER 2
THEORETICAL RESULTS

2.1 Mathematical Notations

In the following sections, we use bold lower case symbols $\mathbf{v}$ to denote vectors and bold upper case symbols $\mathbf{M}$ to denote matrices. $\|\mathbf{M}\|_F$ denote the Frobenius form of the matrix $\mathbf{M}$, and $\|\mathbf{v}\|$ denote the 2-norm (Euclidean norm) of the vector $\mathbf{v}$. Without further specification, $\times$ represents the element-wise product between vectors and the usual multiplication between scalers. When the operation is omitted between two consecutive vectors, the operation is the dot product by default (e.g., $\mathbf{wh}$ indicate the dot product of $\mathbf{w}$ and $\mathbf{h}$.) Subscript is used to indicate the index of an element is a set of similar elements. We also use a tilde beyond variable to denote the mean value of a set of scalers or vectors (i.e. $\tilde{\mathbf{v}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{v}_i$), and a bar above a vector to denote the normalized version of the vector (i.e. $\bar{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$). A dot above a variable denotes a centered version of the variable with respect to the mean (i.e., $\dot{\mathbf{v}}_i = \mathbf{v}_i - \tilde{\mathbf{v}}$). We also use the common big-O notation to denote the asymptotic behavior of a variable.

2.2 Problem Setup

Neural Network with Cross-Entropy Loss. In this work, we consider neural network classifiers trained using cross-entropy loss functions. A vanilla neural network classifier trained using cross-entropy loss can be mathematically represented as a function $f_\theta(\boldsymbol{x})$ that maps an input vector $\boldsymbol{x} \in \mathbb{R}^n$ to an output probability distribution over C classes, where θ represents the parameters of the network. The network is composed of L layers, where the l-th layer takes as input the output of the $(l-1)$-th layer, denoted as $\boldsymbol{a}^{(l-1)} \in \mathbb{R}^{d_{l-1}}$, and produces an output vector $\boldsymbol{a}^{(l)} \in \mathbb{R}^{d^{(l)}}$, where $d^{(l)}$ represents the number of neurons in the

l-th layer.

Each layer is parameterized by a weight matrix $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d^{(l)} \times d^{(l-1)}}$ and a bias vector $\boldsymbol{b}^{(l)} \in \mathbb{R}^{d^{(l)}}$. The output of the l-th layer can be computed as $\boldsymbol{h}^{(l)} = \sigma(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)})$, where σ is a non-linear activation function. This activation function can include element-wise application of non-linear functions such as $\text{ReLU}(x) = \max\{x, 0\}$ as well as normalization layers such as Batch Normalization, which will be further elaborated in later paragraphs.

For the task of classification, the output layer is typically followed by a softmax function that produces a probability distribution over the C classes. Let $\boldsymbol{h}^{(L)} \in \mathbb{R}^C$ denote the output activation vector of the last layer, then the predicted probability distribution is given by $\hat{\boldsymbol{y}} = \text{softmax}(\boldsymbol{h}^{(L)})$, where $\hat{y}_c$ represents the predicted probability for the c-th class and for a vector $\mathbf{h} \in \mathbb{R}^C$, the c'th element of $\text{softmax}(\mathbf{h})$ is defined as

$$\text{softmax}(\mathbf{h})_c = \frac{\exp(h_c)}{\sum_{c'=1}^{C} \exp(h_{c'})}$$

We also assume that the training dataset contains C classes with N samples from each class, which correspond to the case of a balanced training set. Let $\mathbf{x}_{c,i}$ denote the input sample with label class c and index i within the class, and $\mathbf{h}_{c,i}^{(l)}$ denote its corresponding features in the network.

The network is trained by minimizing the cross-entropy loss between the predicted probability distribution and the true label distribution. Let $\boldsymbol{y} \in \{0, 1\}^C$ denote the one-hot encoded true label vector, then the cross-entropy loss for a single training example can be expressed as $L_{c,i} = -\sum_{c=1}^{C} y_c \log \hat{y}_c$. Note that for one-hot encoded vectors, the cross-entropy loss with ground-truth class label c can be simplified as

$$L_{c,i} = -\log \hat{y}_c = -\log \left(\frac{\exp(h_c^{(L)})}{\sum_{c'=1}^{C} \exp(h_{c'}^{(L)})} \right)$$

We also denote that total loss over the entire training set as $\mathcal{L} = \frac{1}{CN} \sum_{c=1}^{C} \sum_{i=1}^{N} L_{c,i}$.

During the training process, the training samples are typically organized into mini-batches of size B. At each step, the weights of the model are updated using gradient descent with respect to the average batch loss $\mathcal{L}(\theta) = -\frac{1}{B}\sum_{i=1}^{B} L_i$.

Batch Normalization and Weight Decay. For a given batch of vectors $\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_b \subset \mathbb{R}^d$, let $v^{(k)}$ denote the k'th element of $\mathbf{v}$. Batch Normalization (BN) [14] performs the following operation along each dimension $k \in [d]$:

$$BN(\mathbf{v}_i)^{(k)} = \frac{v_i^{(k)} - \mu^{(k)}}{\sigma^{(k)}} \times \gamma^{(k)} + \beta^{(k)}$$

Where $\mu^{(k)}$ and $(\sigma^{(k)})^2$ are the mean and variance along the k'th dimension of all the vectors in the batch. The vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are trainable parameters that represent the desired variance and mean after Batch Normalization.

When training a neural network, regularization techniques can be used to prevent overfitting the training data. One common regularization technique is Weight Decay (WD), which adds a penalty term to the loss function to discourage the model from overfitting by penalizing large weight parameters. Specifically, the weight decay regularization term is typically formulated as the L2 norm of the weight matrix $\boldsymbol{W}$, with a hyperparameter λ controlling the strength of the regularization. When applied to a linear layer, the regularization term can be written as

$$\mathcal{L}_{\text{reg}}(\theta) = \frac{\lambda}{2}\|\boldsymbol{W}\|_F^2,$$

where $\|\cdot\|_F$ denotes the Frobenius norm.

The weight vector of the batch normalization layer is also regularized using weight decay. When applied to a linear layer followed by a BN layer, the regularization term for that layer can be written as

$$\mathcal{L}_{\text{reg}}(\theta) = \frac{\lambda}{2}\left(\|\boldsymbol{\gamma}\|^2 + \|\boldsymbol{W}\|_F^2\right)$$

where γ is the trainable scale parameter of the BN layer. The regularization term penalizes the magnitude of the weight matrix $\boldsymbol{W}$ and the scale parameter γ of the BN layer. To apply WD during training, the regularization term is added to the original cross-entropy loss, resulting in a total loss function that includes the regularization term for all linear and Batch Normalization layers:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}(\theta) + \sum_{l} \mathcal{L}_{\text{reg}}(\boldsymbol{W}^{(l)}, \boldsymbol{\gamma}^{(l)}).$$

Here, $\mathcal{L}(\theta)$ is the original cross-entropy loss, and the summation is taken over all linear and Batch Normalization layers in the network. The total loss function is minimized during training using gradient descent with respect to the network parameters θ, and the gradients of the regularization term are added to the gradients of the cross-entropy loss during backpropagation.

Layer-Peeled Formulation. Our analysis mainly focuses on the features and weights of the final output layer, similar to the *layer-peeled* model or *unconstrained features* [10] model of Neural Collapse studied in prior works.

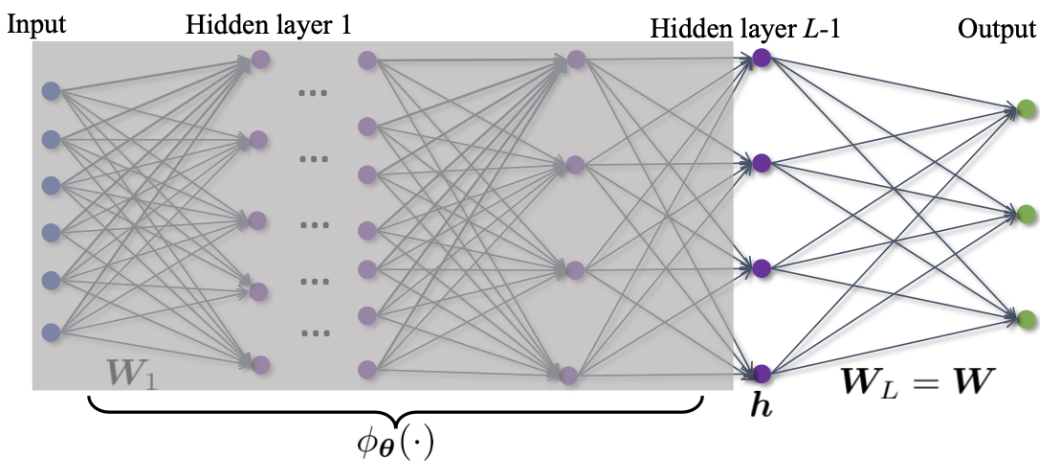

Figure 2.1: An illustration of the layer peeled model from [11]. The parts of the network in the grey box are "peeled off" so that we only consider the last-layer feature representation $\mathbf{h}$

Notably, we denote $\mathbf{W} = \mathbf{W}^{(L)}$ and $\mathbf{h} = \mathbf{h}^{(L-1)}$ when there's no ambiguity. We also use $\mathbf{w}_c$ to represent the c'th row of the weight matrix $\mathbf{W}$. Additionally, we consider network architectures for which batch normalization is applied before the final layer of the network so that the final layer features are normalized according to scale parameter $\boldsymbol{\gamma}$. We also ignore the bias term for both the final batch normalization layer and the final layer, which is equivalent to appending an extra 1 to each feature vector and adding an extra dimension for each weight vector. In this formulation, the final cross-entropy loss for a single sample $\mathbf{x}_{c,i}$ with ground-truth label c and corresponding last-layer feature $\mathbf{h}_{c,i}$ is equal to

$$L_{c,i} = -\log\left(\frac{\exp(\mathbf{w}_c \mathbf{h}_{c,i})}{\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'} \mathbf{h}_{c,i})}\right)$$

To account for the effect of BN and WD in the layer-peeled formulation, we make the following observation on the effect of BN on the last layer features:

Proposition 2.2.1. *Let $\{\mathbf{h}_i\}_{i=1}^{N}$ be a set of feature vectors immediately after Batch Normalization with variance vector $\boldsymbol{\gamma}$ and bias term $\boldsymbol{\beta} = 0$ (i.e. $\mathbf{h}_i = BN(\mathbf{x}_i)$ for some $\{\mathbf{x}_i\}_{i=1}^{N}$). Then*

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{h}_i\|_2^2} = \|\boldsymbol{\gamma}\|_2$$

For the penalty term of the final layer, we only consider the weight matrix and the Using this observation, we can derive that the total regularized loss for the layer-peeled model without the bias terms can be described as the following:

$$\mathcal{L}_{total} = \frac{1}{CN}\sum_{c=1}^{C}\sum_{n=1}^{N} L_{c,i} + \frac{\lambda}{2}(\|\mathbf{W}\|_F^2 + \frac{1}{CN}\sum_{c=1}^{C}\sum_{n=1}^{N}\|\mathbf{h}_i\|_2^2)$$

For simplicity, let $\alpha = \frac{1}{CN}\sum_{c=1}^{C}\sum_{n=1}^{N}\|\mathbf{h}_i\|_2^2$ and $\beta = \frac{\|\mathbf{W}\|_F}{\sqrt{C}}$, then

$$\mathcal{L}_{total} = \frac{1}{CN}\sum_{c=1}^{C}\sum_{n=1}^{N} L_{c,i} + \frac{\lambda}{2}(C\beta^2 + \alpha^2) \tag{2.1}$$

In a simplified setting, we consider the case where α and β are fixed constants and investigate the average cross-entropy loss

$$\mathcal{L} = \frac{1}{CN} \sum_{c=1}^{C} \sum_{n=1}^{N} L_{c,i}$$

Neural Collapse Measures Numerous measures of NC have been used in past literature, including within-class covariance, signal-to-noise (SNR) ratio, as well as class distance normalized variance (CDNV). While these measures all indicate the emergence of NC when the measured value approaches zero, they do not provide a straightforward and geometrically meaningful measure of how close a given structure is to NC when the values are non-zero. In this work, we measure NC using cosine similarity. Specifically, given two vectors $\mathbf{v}_1$ and $\mathbf{v}_2$, the cosine similarity is defined as $\cos_\angle(\mathbf{v}_1, \mathbf{v}_2) = \frac{\mathbf{v}_1 \cdot \mathbf{v}_2}{\|\mathbf{v}_1\| \|\mathbf{v}_2\|}$, where $\cdot$ denotes the dot product and $\| \cdot \|$ denotes the Euclidean norm. Cosine similarity ranges between -1 and 1, where a value of 1 indicates that the two vectors are in the same direction, a value of 0 indicates that the two vectors are orthogonal (i.e., have no correlation), and a value of -1 indicates that the two vectors are diametrically opposed.

For a given class c, the average intra-class cosine similarity for class c is defined as the average cosine similarity of picking two feature vectors in the class:

$$\text{intra}_c = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \cos_\angle(\mathbf{h}_{c,i}, \mathbf{h}_{c,j})$$

and the intra-class cosine similarity between two classes c, c' is defined as the average cosine similarity of picking one feature vector of class c and another from class c':

$$\text{inter}_c = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \cos_\angle(\mathbf{h}_{c,i}, \mathbf{h}_{c',j})$$

While cosine-similarity does not measure the degree of norm equality and does not guarantee equal norm even at optimal values, cosine-similarity measures can serve as an indicator all other observations of Neural Collapse as follows:

(NC1) *(Variability Collapse)* Achieving an intra-class cosine similarity of 1 for each class indicates that all feature vectors within the same class exactly the same up to scaling.

(NC2) *(Convergence to Simplex ETF)* Achieving an inter-class cosine similarity of $-\frac{1}{C-1}$ for every pair of classes indicates that the feature vectors form a Simplex ETF after normalization.

(NC3) *(Self-Duality)* Achieving $\cos_{\angle}(\mathbf{w}_c, \tilde{\mathbf{h}_c}) = 1$ where $\tilde{\mathbf{h}_c} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{h}_{c,i}$ for every class c implies that each class mean feature vector is the same as the corresponding weight vector up to scaling.

(NC4) *(Convergence to NCC)* We show that with high intra-class cosine similarity and low inter-class cosine similarity, an algorithm that finds the mean feature vector with the highest cosine similarity can achieve high accuracy.

With the above mathematical problem formulation, we now present our main theorem for NC in neural network classifiers with near-optimal cross-entropy loss.

2.3 Main Results

The following theorem demonstrates that if a neural network model with batch normalization achieves near-optimal average cross-entropy loss on the training set, then most classes have average intra-class cosine similarity near one, have the cosine similarity between the weight and mean feature vectors near one, and most pairs of classes have average inter-class cosine similarity near $-\frac{1}{C-1}$

Theorem 2.3.1. *For any $C \geq 3$ and $N \geq 1$, under the following assumptions:*

1. *The quadratic average of the feature norms $\sqrt{\frac{1}{CN}\sum_{c=1}^{C}\sum_{i=1}^{N}\|\mathbf{h}_{c,i}\|^2} \leq \alpha$*
2. *The corresponding Frobenius norm of the last-layer weight $\|\mathbf{W}\|_F \leq \sqrt{C}\beta$*

3. *The average cross-entropy loss over all samples* $\frac{1}{CN}\sum_{c=1}^{C}\sum_{i=1}^{N} L_{c,i} \leq m+\epsilon$ *for small* ϵ

where $m = \log(1+(C-1)\exp(-\frac{C}{C-1}\alpha\beta))$ *is the minimum achievable loss for any set of weight and feature vectors satisfying the norm constraints, then for at least* $1-\delta$ *fraction of all classes , with* $\frac{\epsilon}{\delta} \ll 1$, *there is*

$$\textit{intra}_c \geq 1 - O(\frac{e^{O(\alpha\beta)}}{\alpha\beta}\sqrt{\frac{\epsilon}{\delta}})$$

and for at least $1-\delta$ *fraction of all pairs of classes* c, c', *with* $\frac{\epsilon}{\delta} \ll 1$, *there is*

$$\textit{inter}_{c,c'} \leq -\frac{1}{C-1} + O(\frac{e^{O(\alpha\beta)}}{\alpha\beta}(\frac{\epsilon}{\delta})^{1/6})$$

Remarks. The current state of the literature on Neural Collapse has largely focused on the study of global minimizers, landscape analysis, and training dynamics of the layer-peeled model. These works assume the final layer's feature vectors can be arbitrarily assigned or modified using gradient descent. However, this assumption is unrealistic, even in over-parameterized networks. Our work builds upon the layer-peeled model but introduces a novel approach that doesn't require specific training dynamics performed directly on the feature vectors. Our result only requires that the model achieves low loss and does not require any additional assumptions on the training method and the structure of deeper layers (except for a final Batch Normalization before the last-layer features). Therefore, our result has direct implications for a realistic trained neural network that fits the architectural assumptions and achieves low training loss.

Below are a few remarks on the terms of the main theorem:

- Since δ is the fraction of classes or pairs of classes, it can only take values that are multiples of $\frac{1}{C}$ for the first case and $\frac{1}{C(C-1)}$ for the second. By assigning any value of $\delta < \frac{1}{C}$, the intra-class cosine similarity bound would hold for every class, and by assigning any value of $\delta < \frac{1}{C(C-1)}$, the inter-class bound would hold for every pair of classes.

- We only consider the near-optimal regime where $\epsilon \ll 1$, and our bound is vacuous if this assumption does not hold. However, a near-optimal cross-entropy loss is demonstrated in most successful neural network classifiers, including all the original experiments of the NC observation [1], at the terminal phase of training.

- The $e^{O(\alpha\beta)}$ term is about $e^{\frac{C}{C-1}\alpha\beta}$, which is close to $e^{\alpha\beta}$. This means that the term $\frac{e^{O(\alpha\beta)}}{\alpha\beta}$ is increasing when $\alpha\beta > \frac{C-1}{C}$ and decreasing when $\alpha\beta < \frac{C-1}{C}$. However, in practice, it's almost always the case that $\alpha\beta > \frac{C-1}{C}$, therefore the term $\frac{e^{O}(\alpha\beta)}{\alpha\beta}$ can be viewed as an increasing term with regard to both α and β. Note that while α and β are constants in our theoretical bound, they are instead regularized as part of the loss function in (Equation 2.1) during the training process. Under the same cross-entropy loss function, larger weight decay values result in smaller α and β. Therefore, our bound indicates that larger weight decay values can NC at the near-optimal loss regime.

Proof Idea of Theorem 2.3.1. Our proof is inspired by the optimal-case proof of Lu & Steinerberger [8], which shows the global optimality conditions using Jensen's inequality. For any set of vectors $\{\mathbf{x}\}_{i=1}^{N}$, let $\tilde{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{n}\mathbf{x}$. In essence, our proof first shows that the average intra-class and inter-class cosine similarity is heavily related to the vector norm and between-class cosine similarities of $\tilde{\tilde{\mathbf{h}}}_c = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbf{h}_{c,i}}{\|\mathbf{h}_{c,i}\|}$. Then, using the strong convexity of $\exp(x)$ and $\log(1+(C-1)\exp(x))$, we show a upper bound on $(\mathbf{w}_c - \tilde{\mathbf{w}})\tilde{\mathbf{h}}_c$ and a lower bound of $(\mathbf{w}_c' - \mathbf{w}_c)\tilde{\mathbf{h}}_c$ for any subset of classes S. We then prove a lemma that converts these bounds to statements on $\tilde{\tilde{\mathbf{h}}}_c$ and therefore implies the desired upper and lower bound.

2.4 Proof of Theorem 2.3.1

In order to prove the main theorem, we first present several lemmas that facilitate the proof technique used in the main proof. The first two lemmas demonstrate that if a set of variables achieves roughly equal value on the LHS and RHS of Jensen's inequality for a strongly convex function, then the mean of every subset cannot deviate too far from the global mean.

Our first lemma states that, For λ-strongly-convex-function f and a set of numbers $\{x_i\}_{i=1}^N$, if Jensen's inequality has its gap bounded by ϵ, then the mean of any subset that includes δ fraction of all samples can not deviate from global mean of all samples by more than $\sqrt{\frac{2\epsilon(1-\delta)}{\lambda\delta}}$:

Lemma 2.4.1. *Let $\{x_i\}_{i=1}^N \subset \mathcal{I}$ be any set of N real numbers, let $\tilde{x} = \frac{1}{N}\sum_{i=1}^N x_i$ be the mean over all x_i and f be a function that is λ-strongly-convex on $\mathcal{I}$. If*

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i) \leq f(\tilde{x}) + \epsilon$$

Then for any subset of samples $S \subseteq [N]$, let $\delta = \frac{|S|}{N}$, there is

$$\tilde{x} + \sqrt{\frac{2\epsilon(1-\delta)}{\lambda\delta}} \geq \frac{1}{|S|}\sum_{i\in S} x_i \geq \tilde{x} - \sqrt{\frac{2\epsilon(1-\delta)}{\lambda\delta}}$$

Proof. For the proof, we use a result from [15] which bounds the Jensen inequality gap using the variance of the variables for strongly convex functions:

Lemma 2.4.2 (Theorem 4 from [15])**.** *If $f : I \to \mathbb{R}$ is strongly convex with modulus c, then*

$$f\left(\sum_{i=1}^{n} t_i x_i\right) \leq \sum_{i=1}^{n} t_i f(x_i) - c\sum_{i=1}^{n} t_i (x_i - \bar{x})^2$$

for all $x_1, \ldots, x_n \in I$, $t_1, \ldots, t_n > 0$ with $t_1 + \cdots + t_n = 1$ and $\bar{x} = t_1 x_1 + \cdots + t_n x_n$

In the original definition of the authors, a strongly convex function with modulus c is equivalent to a $2c$-strongly-convex function. We can apply $t_i = \frac{1}{N}$ for all i and substitute the definition for strong convexity measure to obtain the following corollary:

Corollary 2.4.1. *If $f : I \to \mathbb{R}$ is λ-strongly-convex on $\mathcal{I}$, and*

$$\frac{1}{N}\sum_{i=1}^{N} f(x_i) = f\left(\frac{1}{N}\sum_{i=1}^{N} x_i\right) + \epsilon$$

for $x_1, \ldots, x_N \in \mathcal{I}$, then $\frac{1}{N}\sum_i (x_i - \bar{x})^2 \leq \frac{2\epsilon}{\lambda}$

From Corollary 2.4.1, we know that $\frac{1}{N}\sum_{i=1}^{n}(x_i - \tilde{x})^2 \leq \frac{2\epsilon}{\lambda}$. Let $D = \sum_{i \in S}(x_i - \tilde{x})$, by the convexity of x^2, there is

$$\begin{aligned}
\sum_{i=1}^{n}(x_i - \tilde{x})^2 &= \sum_{i \in S}(x_i - \tilde{x})^2 + \sum_{i \notin S}(x_i - \tilde{x})^2 \\
&\geq |S|(\frac{1}{|S|}\sum_{i \in S}(x_i - \tilde{x}))^2 + (N - |S|)(\frac{1}{N - |S|}\sum_{i \notin S}(x_i - \tilde{x}))^2 \\
&= \frac{1}{S}(\sum_{i \in S}(x_i - \tilde{x}))^2 + \frac{1}{N - |S|}(\sum_{i \notin S}(x_i - \tilde{x}))^2 \\
&= \frac{1}{S}D^2 + \frac{1}{N - |S|}(-D)^2 \\
&= \frac{D^2}{N}(\frac{1}{\delta} + \frac{1}{1 - \delta}) \\
&= \frac{D^2}{N}(\frac{1}{\delta(1 - \delta)})
\end{aligned}$$

Therefore $\frac{D^2}{N}(\frac{1}{\delta(1-\delta)}) \leq \frac{2\epsilon N}{\lambda}$, and $|D| \leq \sqrt{\frac{2\epsilon\delta(1-\delta)N^2}{\lambda}}$. Using $\frac{1}{|S|}\sum_{i \in S} x_i = \frac{1}{|S|}(|S|\tilde{x} + D)$ and $|S| = \delta N$ completes the proof. □

Our second lemma states a similar result specific to the function e^x and only provides the upper bound. Note that, within any predefined range $[a, b]$, $\exp(x)$ can only be guaranteed to be e^a strongly convex, which may be bad if the lower bound a is small or does not exist. Our further result in the following lemma shows that we can provide a better upper bound of the subset mean for the exponential function that is dependent on $\exp(\tilde{x})$ and does not require other prior knowledge of the range of x_i:

Lemma 2.4.3. *Let* $\{x_i\}_{i=1}^N \subset \mathbb{R}$ *be any set of* N *real numbers, let* $\tilde{x} = \frac{1}{N}\sum_{i=1}^N x_i$ *be the mean over all* x_i*. If*

$$\frac{1}{N}\sum_{i=1}^{N} \exp(x_i) \le \exp(\tilde{x}) + \epsilon$$

then for any subset $S \subseteq [N]$*, let* $\delta = \frac{|S|}{N}$*, the there is*

$$\frac{1}{|S|}\sum_{i\in S} x_i \le \tilde{x} + \sqrt{\frac{2\epsilon}{\delta\exp(\tilde{x})}}$$

.

Proof. Let $D = \sum_{i\in S}(x_i - \tilde{x})$. Note that if $D < 0$ then the upper bound is obviously satisfied since the subset mean will be smaller than the global mean. Therefore, we only consider the case when $D > 0$

$$\begin{aligned}
\sum_{i=1}^{N}\exp(x_i) &= \sum_{i\in S}\exp(x_i) + \sum_{i\notin S}\exp(x_i) \\
&\ge |S|\exp(\frac{1}{|S|}\sum_{i\in S}x_i) + (N-|S|)\exp(\frac{1}{N-|S|}\sum_{i\notin S}x_i) \\
&\ge |S|\exp(\tilde{x} + \frac{D}{|S|}) + (N-|S|)\exp(\tilde{x} - \frac{D}{N-|S|}) \\
&\ge |S|\exp(\tilde{x})(1 + \frac{D}{|S|} + \frac{D^2}{2|S|^2}) + (N-|S|)\exp(\tilde{x})(1 - \frac{D}{N-|S|}) \\
&= (N + \frac{D^2}{2|S|})\exp(\tilde{x}) \\
N\exp(\tilde{x}) + N\epsilon &\ge (N + \frac{D^2}{2|S|})\exp(\tilde{x}) \\
D^2 &\le \frac{2|S|N\epsilon}{\exp(\tilde{x})} \\
D &\le N\sqrt{\frac{2\delta\epsilon}{\exp(\tilde{x})}}
\end{aligned}$$

Using $\frac{1}{|S|}\sum_{i\in S} x_i = \frac{1}{|S|}(|S|\tilde{x} + D)$ and $|S| = \delta N$ completes the proof. $\square$

Directly approaching the average intra-class and inter-class cosine similarity of vector set(s) is a relatively difficult task. Our following lemma shows that the inter-class and inter-class cosine similarities can be computed as the norm and dot product of the vectors $\tilde{\bar{\mathbf{h}}}_c$, respectively, where $\tilde{\bar{\mathbf{h}}}_c$ is the mean *normalized* vector among all vectors in a class.

Lemma 2.4.4. *Let c, c' be 2 classes, each containing N feature vectors $\mathbf{h}_{c,i} \in \mathbb{R}^d$. Define the average intra-class cosine similarity of picking two vectors from the same class c as*

$$\text{intra}_c = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \cos_\angle(\mathbf{h}_{c,i}, \mathbf{h}_{c,j})$$

and the intra-class cosine similarity between two classes c, c' is defined as the average cosine similarity of picking one feature vector of class c and another from class c' as

$$\text{inter}_c = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \cos_\angle(\mathbf{h}_{c,i}, \mathbf{h}_{c',j})$$

Let $\tilde{\bar{\mathbf{h}}}_c = \frac{1}{N} \sum_{i=1}^{N} \frac{\mathbf{h}_{c,i}}{\|\mathbf{h}_{c,i}\|}$. Then intra$_c = \|\tilde{\bar{\mathbf{h}}}_c\|^2$ and inter$_{c,c'} = \tilde{\bar{\mathbf{h}}}_c \cdot \tilde{\bar{\mathbf{h}}}_{c'}$

Proof. For the intra-class cosine similarity,

$$\begin{aligned}
\text{intra}_c &= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \bar{\mathbf{h}}_{c,i} \cdot \bar{\mathbf{h}}_{c,j} \\
&= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\mathbf{h}_{c,i}}{\|\mathbf{h}_{c,i}\|} \cdot \frac{\mathbf{h}_{c,j}}{\|\mathbf{h}_{c,j}\|} \\
&= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\mathbf{h}_{c,i} \cdot \mathbf{h}_{c,j}}{\|\mathbf{h}_{c,i}\|\|\mathbf{h}_{c,j}\|} \\
&= \left(\frac{1}{N} \sum_{i=1}^{N} \frac{\mathbf{h}_{c,i}}{\|\mathbf{h}_{c,i}\|} \right) \cdot \left(\frac{1}{N} \sum_{j=1}^{N} \frac{\mathbf{h}_{c,j}}{\|\mathbf{h}_{c,j}\|} \right) \\
&= \|\tilde{\bar{\mathbf{h}}}_c\|^2
\end{aligned}$$

and for the inter-class cosine similarity,

$$\begin{aligned}
\text{inter}_{c,c'} &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\bar{\mathbf{h}}_{c,i}\cdot\bar{\mathbf{h}}_{c',j} \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\mathbf{h}_{c,i}}{\|\mathbf{h}_{c,i}\|}\cdot\frac{\mathbf{h}_{c',j}}{\|\mathbf{h}_{c',j}\|} \\
&= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\mathbf{h}_{c,i}\cdot\mathbf{h}_{c',j}}{\|\mathbf{h}_{c,i}\|\|\mathbf{h}_{c',j}\|} \\
&= \left(\frac{1}{N}\sum_{i=1}^{N}\frac{\mathbf{h}_{c,i}}{\|\mathbf{h}_{c,i}\|}\right)\cdot\left(\frac{1}{N}\sum_{j=1}^{N}\frac{\mathbf{h}_{c',j}}{\|\mathbf{h}_{c',j}\|}\right) \\
&= \tilde{\bar{\mathbf{h}}}_c\cdot\tilde{\bar{\mathbf{h}}}_{c'}
\end{aligned}$$

□

We prove the intra-class cosine similarity by first showing that the norm of the mean (un-normalized) class-feature vector for a class is near the quadratic average of feature means (i.e., $\|\tilde{\mathbf{h}}_c\| = \|\frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_{c,i}\| \approx \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{h}_{c,i}\|^2}$). However, to show intra-class cosine similarity, we need instead a bound on $\|\tilde{\bar{\mathbf{h}}}_c\| = \|\frac{1}{N}\sum_{i=1}^{N}\bar{\mathbf{h}}_{c,i}\|$. The following lemma provides a conversion between these requirements:

Lemma 2.4.5. *Suppose* $\mathbf{u}\in\mathbb{R}^d$ *and* $\|\mathbf{u}\|\leq\beta$. *Let* $\{\mathbf{v}_i\}_{i=1}^{N}\subset\mathbb{R}^d$ *such that* $\frac{1}{N}\|\mathbf{v}_i\|^2\leq\alpha^2$. *If*

$$\frac{1}{N}\sum_{i=1}^{N}\langle\mathbf{u},\mathbf{v}_i\rangle\geq c$$

, for $\frac{\alpha\beta}{\sqrt{2}}\leq c\leq\alpha\beta$ *and let* $\bar{\mathbf{v}}=\frac{\mathbf{v}}{\|\mathbf{v}\|}$ *then*

$$\tilde{\bar{\mathbf{v}}} = \|\frac{1}{N}\sum_{i=1}^{N}\bar{\mathbf{v}}\| \geq 2(\frac{c}{\alpha\beta})^2 - 1$$

Proof. Divide into 2 cases: the set of indices

$$pos = \{i\in[N]|\langle\mathbf{u},\mathbf{v}_i\rangle > 0\}$$

and

$$neg = \{i \in [N] | \langle \mathbf{u}, \mathbf{v}_i \rangle < 0\}$$

Let $M = |pos|$, then note that

$$\begin{aligned}
\sum_{i \in pos} \langle \mathbf{u}, \mathbf{v}_i \rangle &\geq Nc \\
\sum_{i \in pos} \langle \mathbf{u}, \mathbf{v}_i \rangle &\leq \|\mathbf{u}\| \sum_{i \in pos} \|\mathbf{v}_i\| \\
&\leq \beta \sqrt{M \sum_{i \in pos} \|\mathbf{v}_i\|^2} && E[X^2] \geq E[X]^2 \\
&\leq \beta \sqrt{MN\alpha^2} \\
&= \alpha\beta\sqrt{MN}
\end{aligned}$$

Therefore

$$N \geq M \geq N(\frac{c}{\alpha\beta})^2$$

First, consider $\sum_{i \in pos} \langle \mathbf{u}, \bar{\mathbf{v}}_i \rangle$. Note that $\sum_{i \in pos} \|\mathbf{v}_i\|^2 \leq N\alpha^2$ and $\sum_{i \in pos} \langle \mathbf{u}, \mathbf{v}_i \rangle \geq Nc$. We will use the following proposition that can be easily shown through Lagrange multipliers: Given $\{a_i\}_{i=1}^N$ and $\{b_i\}_{i=1}^N$ such that $a_i \geq 0$ and $b_i > 0$ for all i, if $\sum_{i=1}^n a_i = A$ and $\sum_{i=1}^n b_i^2 \leq B$, then $\sum_{i=1}^N \frac{a_i}{b_i} \geq \frac{A\sqrt{N}}{\sqrt{B}}$

Therefore

$$\begin{aligned}
\sum_{i \in pos} \langle \mathbf{u}, \bar{\mathbf{v}}_i \rangle &= \sum_{i \in pos} \frac{\langle \mathbf{u}, \mathbf{v}_i \rangle}{\|\mathbf{v}_i\|} \\
&\geq \frac{c}{\alpha} \cdot \sqrt{MN} && \text{Proposition} \\
&\geq \frac{c}{\alpha} \cdot (N\frac{c}{\alpha\beta}) \\
&= N\beta(\frac{c}{\alpha\beta})^2
\end{aligned}$$

On the other hand, for neg, since $\langle \mathbf{u}, \bar{\mathbf{v}}_i \rangle \geq -\|u\| \geq -\beta$, we get

$$\begin{aligned}
\sum_{i \in pos} \langle \mathbf{u}, \bar{\mathbf{v}}_i \rangle \geq \sum_{i \in neg} -\beta \\
&= -\beta(N - M) \\
&\geq -\beta N(1 - (\frac{c}{\alpha\beta})^2) = N\beta((\frac{c}{\alpha\beta})^2 - 1)
\end{aligned}$$

Therefore

$$\begin{aligned}
\|\mathbf{u}\| \| \frac{1}{N} \sum_{i=1}^{N} \bar{\mathbf{v}}_i \| \geq \frac{1}{N} \sum_{i=1}^{N} \langle \mathbf{u}, \bar{\mathbf{v}}_i \rangle &\geq \frac{1}{N} (\sum_{i \in pos} \langle \mathbf{u}, \bar{\mathbf{v}}_i \rangle + \sum_{i \in neg} \langle \mathbf{u}, \bar{\mathbf{v}}_i \rangle) \\
&\geq \frac{1}{N} (N\beta((\frac{c}{\alpha\beta})^2 - 1) + N\beta(\frac{c}{\alpha\beta})^2) \\
&= \beta(2(\frac{c}{\alpha\beta})^2 - 1) \\
\| \frac{1}{N} \sum_{i=1}^{N} \bar{\mathbf{v}} \|^2 &\geq 2(\frac{c}{\alpha\beta})^2 - 1
\end{aligned}$$

□

To make this lemma generalize to other proofs in future work, we provide the generalized corollary of the above lemma by setting $\mathbf{u}$ to be the normalized mean vector of $\mathbf{v}$:

Corollary 2.4.2. *Let* $\{\mathbf{v}_i\}_{i=1}^N \subset \mathbb{R}^d$ *such that* $\frac{1}{N}\|\mathbf{v}_i\|^2 \leq \alpha^2$. *If*

$$\| \frac{1}{N} \sum_{i=1}^{N} \mathbf{v}_i \| \geq c$$

, for $\frac{\alpha}{\sqrt{2}} \leq c \leq \alpha$ *and let* $\bar{\mathbf{v}} = \frac{\mathbf{v}}{\|\mathbf{v}\|}$ *then*

$$\tilde{\bar{\mathbf{v}}} = \| \frac{1}{N} \sum_{i=1}^{N} \bar{\mathbf{v}} \| \geq 2(\frac{c}{\alpha})^2 - 1$$

Similarly, for inter-class cosine similarity, we have the following lemma:

Lemma 2.4.6. *Let* $\mathbf{w} \in \mathbb{R}^d$, $\{\mathbf{h}_i\}_{i=1}^N \subset \mathbb{R}^d$. *Let* $\tilde{\mathbf{h}} = \frac{1}{N}\sum_{i=1}^N \mathbf{h}_i$ *and* $\tilde{\tilde{\mathbf{h}}} = \frac{1}{N}\sum_{i=1}^N \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|}$. *If the following condition is satisfied:*

$$\frac{1}{N}\mathbf{w} \cdot \sum_{i=1}^N \mathbf{h}_i \le c \qquad \textit{for } c < 0$$

$$\|\mathbf{w}\| \le \beta$$

$$\frac{1}{N}\sum_{i=1}^n \|\mathbf{h}_i\|^2 \le \alpha^2$$

$$\exists \mathbf{w}' \in \mathbb{R}^d, \|\mathbf{w}'\| \le \beta, \frac{1}{N}\mathbf{w}'\sum_{i=1}^N \mathbf{h}_i \ge \alpha\beta - \epsilon'$$

$$\epsilon' \ll \alpha\beta$$

Then $\cos_\angle(\mathbf{w}, \tilde{\tilde{\mathbf{h}}}) \le -\frac{c}{\alpha\beta} + 4(\frac{\epsilon'}{\alpha\beta})^{1/3}$

Proof. For $\mathbf{w} \in \mathbb{R}^d$, $\{\mathbf{h}_i\}_{i=1}^N \subset \mathbb{R}^d$

Let $a_i = \frac{1}{N}\mathbf{w}\mathbf{h}_i$, $b_i = \|\mathbf{h}_i\|$, $\epsilon = \frac{\epsilon'}{\beta}$, then the constraints of the above problem relaxed as follows:

$$\max \sum_{i=1}^N \frac{a_i}{b_i}$$

$$s.t. \sum_{i=1}^N a_i \le c$$

$$\frac{1}{N}\sum_{i=1}^N b_i^2 = \alpha^2$$

$$\frac{1}{N}\sum_{i=1}^N b_i \ge \alpha - \frac{\epsilon}{\beta}$$

$$\forall i, |\frac{a_i}{b_i}| \le \beta$$

First, consider the case when $\frac{1}{N}\sum_{i=1}^N b_i \ge \alpha - \epsilon$ Consider a random variable B that uniformly picks a value from $\{b_i\}_{i=1}^N$. Then $\mathbb{E}[B] \ge \alpha - \frac{\epsilon}{\beta}$, $\mathbb{E}[B^2] = \alpha^2$, and therefore

$\sigma_B = \sqrt{\mathbb{E}[B^2] - \mathbb{E}[B]^2} \leq \sqrt{2\alpha\epsilon}$. According to Chebyshev's inequality

$$P(|B - (\alpha - \epsilon)| \geq k\sqrt{2\alpha\epsilon}) \leq \frac{1}{k^2}$$

Note that for positive a_i, smaller b_i means larger $\frac{a_i}{b_i}$ and for negative a_i, higher b_i means larger $\frac{a_i}{b_i}$. Suppose that ϵ is sufficiently small such that $\epsilon \ll \sqrt{\epsilon}$.Therefore, an upper bound for $\frac{a_i}{b_i}$ when $a_i > 0$ is

$$\frac{a_i}{b_i} \leq \begin{cases} \frac{a_i}{\alpha - k\sqrt{2\alpha\epsilon}} & b_i \geq \alpha - k\sqrt{2\alpha\epsilon} \\ \beta & b_i < \alpha - k\sqrt{2\alpha\epsilon} \end{cases}$$

and an upper bound for $a_i < 0$ would is

$$\frac{a_i}{b_i} \leq \begin{cases} \frac{a_i}{\alpha + k\sqrt{2\alpha\epsilon}} & b_i \leq \alpha + k\sqrt{2\alpha\epsilon} \\ 0 & b_i > \alpha + k\sqrt{2\alpha\epsilon} \end{cases}$$

Suppose that $k\sqrt{\frac{2\epsilon}{\alpha}}$ is less than $\frac{1}{2}$, then

$$\frac{a_i}{\alpha - k\sqrt{2\alpha\epsilon}} = \frac{a_i}{\alpha} \cdot \frac{1}{1 - k\sqrt{\frac{2\epsilon}{\alpha}}} < \frac{a_i}{\alpha} \cdot (1 + 2k\sqrt{\frac{2\epsilon}{\alpha}}) = \frac{a_i}{\alpha} + |\frac{a_i}{\alpha}| \cdot 2k\sqrt{\frac{2\epsilon}{\alpha}}$$

when $a_i > 0$, and similarly

$$\frac{a_i}{\alpha + k\sqrt{2\alpha\epsilon}} = \frac{a_i}{\alpha} \cdot \frac{1}{1 + k\sqrt{\frac{2\epsilon}{\alpha}}} < \frac{a_i}{\alpha} \cdot (1 - 2k\sqrt{\frac{2\epsilon}{\alpha}}) = \frac{a_i}{\alpha} + |\frac{a_i}{\alpha}| \cdot 2k\sqrt{\frac{2\epsilon}{\alpha}}$$

when $a_i < 0$. Note that

$$\sum_{i=1}^{N} |\frac{a_i}{\alpha}| \cdot 2k\sqrt{\frac{2\epsilon}{\alpha}} \leq \sum_{i=1}^{N} \frac{\beta}{N} \cdot 2k\sqrt{\frac{2\epsilon}{\alpha}} = 2k\beta\sqrt{\frac{2\epsilon}{\alpha}}$$

Therefore, an upper bound on the total sum would be:

$$\frac{c}{\alpha} + 2k\beta\sqrt{\frac{2\epsilon}{\alpha}} + \frac{\beta}{k^2}$$

Set $k = (\sqrt{\frac{8\epsilon}{\alpha}})^{-\frac{1}{3}}$ to get:

$$\frac{c}{\alpha} + 2\beta(\sqrt{\frac{8\epsilon}{\alpha}})^{\frac{2}{3}} = \frac{c}{\alpha} + 4\beta(\frac{\epsilon}{\alpha})^{\frac{1}{3}}$$

Now, we substitute $\epsilon = \frac{\epsilon'}{\beta}$ we get: $\mathbf{w} \cdot \tilde{\tilde{\mathbf{h}}} \leq \frac{c}{\alpha} + 4\beta(\frac{\epsilon'}{\alpha\beta})^{1/3}$ Since $|\mathbf{w}| \leq \beta$ and $|\tilde{\tilde{\mathbf{h}}}| \leq 1$, we get that

$$\cos_{\angle}(\mathbf{w}, \tilde{\tilde{\mathbf{h}}}) \leq \frac{c}{\alpha\beta} + 4(\frac{\epsilon'}{\alpha\beta})^{1/3}$$

□

Now we proceed to the main proof: First, consider the minimum achievable average loss for a single class c:

$$\frac{1}{N}\sum_{i=1}^{N} L_{c,i} = \frac{1}{N}\sum_{i=1}^{N} \text{softmax}(\mathbf{W}\mathbf{h}_{c,i})_c \tag{2.2}$$

$$\geq \text{softmax}(\frac{1}{N}\sum_{i=1}^{N} \mathbf{W}\mathbf{h}_{c,i})_c \tag{2.3}$$

$$= \log\left(1 + \sum_{c' \neq c} \exp(\frac{1}{N}\sum_{i=1}^{N}(\mathbf{w}_{c'} - \mathbf{w}_c)\mathbf{h}_{c,i})\right) \tag{2.4}$$

$$= \log\left(1 + \sum_{c' \neq c} \exp((\mathbf{w}_{c'} - \mathbf{w}_c)\tilde{\mathbf{h}}_c)\right) \tag{2.5}$$

$$\geq \log\left(1 + (C-1)\exp(\frac{1}{(C-1)}(\sum_{c'=1}^{C} \mathbf{w}_{c'}\tilde{\mathbf{h}}_c - C\mathbf{w}_c\tilde{\mathbf{h}}_c))\right) \tag{2.6}$$

$$= \log\left(1 + (C-1)\exp(\frac{1}{(C-1)}(\sum_{c'=1}^{C} \mathbf{w}_{c'} - C\mathbf{w}_c)\tilde{\mathbf{h}}_c)\right) \tag{2.7}$$

$$= \log\left(1 + (C-1)\exp(\frac{C}{C-1}(\tilde{\mathbf{w}} - \mathbf{w}_c)\tilde{\mathbf{h}}_c)\right) \tag{2.8}$$

$$= \log\left(1 + (C-1)\exp(-\frac{C}{C-1}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c)\right) \tag{2.9}$$

Let $\overrightarrow{\mathbf{w}} = [\mathbf{w}_1 - \tilde{\mathbf{w}}, \mathbf{w}_2 - \tilde{\mathbf{w}}, \ldots, \mathbf{w}_C - \tilde{\mathbf{w}}] = [\dot{\mathbf{w}}_1, \dot{\mathbf{w}}_2, \ldots, \dot{\mathbf{w}}_C]$, and $\overrightarrow{\mathbf{h}} = [\tilde{\mathbf{h}}_1, \tilde{\mathbf{h}}_2, \ldots, \tilde{\mathbf{h}}_c] \in \mathbf{R}^{Cd}$. Note that

$$\begin{aligned}
\|\overrightarrow{\mathbf{w}}\|^2 &= \sum_{c=1}^{C}\|\mathbf{w}_c - \tilde{\mathbf{w}}\|^2 = \sum_{c=1}^{C}\left(\|\mathbf{w}_c\|^2 - 2\mathbf{w}_c\tilde{\mathbf{w}} + \|\tilde{\mathbf{w}}\|^2\right) \\
&= \sum_{c=1}^{C}\|\mathbf{w}_c\|^2 - C\|\tilde{\mathbf{w}}\|^2 \leq \sum_{c=1}^{C}\|\mathbf{w}_c\|^2 = \|\mathbf{W}\|_F^2 \leq C\beta^2
\end{aligned}$$

and also

$$\begin{aligned}
\|\overrightarrow{\mathbf{h}}\|^2 &= \sum_{c=1}^{C}\|\tilde{\mathbf{h}}_c\|^2 = \sum_{c=1}^{C}\|\frac{1}{N}\sum_{i=1}^{N}\mathbf{h}_{c,i}\|^2 \leq \sum_{c=1}^{C}\left(\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{h}_{c,i}\|\right)^2 \\
&\leq \frac{1}{N}\sum_{c=1}^{C}\sum_{i=1}^{N}\|\mathbf{h}_{c,i}\|^2 = C\alpha^2
\end{aligned}$$

The first inequality uses the triangle inequality and the second uses $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$ Now consider the total average loss over all classes:

$$\begin{aligned}
\mathcal{L} &= \frac{1}{CN}\sum_{c=1}^{C}\sum_{i=1}^{N}L_{c,i} \\
&\geq \frac{1}{C}\sum_{c=1}^{C}\log\left(1 + (C-1)\exp(\frac{C}{C-1}(\tilde{\mathbf{w}} - \mathbf{w}_c)\tilde{\mathbf{h}}_c)\right) \\
&\geq \log\left(1 + (C-1)\exp(\frac{C}{C-1}\cdot\frac{1}{C}\sum_{c=1}^{C}(\tilde{\mathbf{w}} - \mathbf{w}_c)\tilde{\mathbf{h}}_c)\right) && \text{Jensen's} \\
&\geq \log\left(1 + (C-1)\exp(-\frac{1}{C-1}\overrightarrow{\mathbf{w}}\cdot\overrightarrow{\mathbf{h}})\right) \\
&\geq \log\left(1 + (C-1)\exp(-\frac{C}{C-1}\alpha\beta\right) \\
&= m
\end{aligned}$$

showing that m is indeed the minimum achievable average loss among all samples.

Now we instead consider when the final average loss is near-optimal of value $m + \epsilon$ with $\epsilon \ll 1$. We use a new ϵ to represent the gap introduced by each inequality in the above proof. Additionally, since the average loss is near-optimal, there must be $\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c \geq 0$ for any sufficiently small ϵ:

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^{N} L_{c,i} &= \frac{1}{N}\sum_{i=1}^{N} \mathrm{softmax}(\mathbf{W}\mathbf{h}_{c,i})_c && (2.10)\\
&\geq \mathrm{softmax}(\frac{1}{N}\sum_{i=1}^{N}\mathbf{W}\mathbf{h}_{c,i})_c && (2.11)\\
&= \log\left(1 + \sum_{c'\neq c}\exp(\frac{1}{N}\sum_{i=1}^{N}(\mathbf{w}_{c'} - \mathbf{w}_c)\mathbf{h}_{c,i})\right) && (2.12)\\
&= \log\left(1 + \sum_{c'\neq c}\exp((\mathbf{w}_{c'} - \mathbf{w}_c)\tilde{\mathbf{h}}_c)\right) && (2.13)\\
&= \log\left(1 + (C-1)\exp(\frac{1}{(C-1)}(\sum_{c'=1}^{C}\mathbf{w}_{c'}\tilde{\mathbf{h}}_c - C\mathbf{w}_c\tilde{\mathbf{h}}_c)) + \epsilon_1'\right) && (2.14)\\
&= \log\left(1 + (C-1)\exp(\frac{1}{(C-1)}(\sum_{c'=1}^{C}\mathbf{w}_{c'} - C\mathbf{w}_c)\tilde{\mathbf{h}}_c) + \epsilon_1'\right) && (2.15)\\
&= \log\left(1 + (C-1)\exp(\frac{C}{C-1}(\tilde{\mathbf{w}} - \mathbf{w}_c)\tilde{\mathbf{h}}_c) + \epsilon_1'\right) && (2.16)\\
&\geq \log\left(1 + (C-1)\exp(-\frac{C}{C-1}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c)\right) + \frac{\epsilon_1'}{1 + (C-1)\exp(-\frac{C}{C-1}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c)} && (2.17)\\
&\geq \log\left(1 + (C-1)\exp(-\frac{C}{C-1}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c)\right) + \frac{\epsilon_1'}{C} && (2.18)
\end{aligned}$$

and also

$$\begin{aligned}
\mathcal{L} &= \frac{1}{CN}\sum_{c=1}^{C}\sum_{i=1}^{N} L_{c,i}\\
&\geq \frac{1}{C}\sum_{c=1}^{C}\left(\log\left(1 + (C-1)\exp(-\frac{C}{C-1}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c)\right) + \frac{\epsilon_{1,c}'}{C}\right) + \epsilon_2'
\end{aligned}$$

$$\geq \log\left(1 + (C-1)\exp(-\frac{C}{C-1}\cdot\frac{1}{C}\sum_{c=1}^{C}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c)\right) + \frac{1}{C}\sum_{c=1}^{C}\frac{\epsilon'_{1,c}}{C} + \epsilon'_2 \quad \text{Jensen's}$$

$$= \log\left(1 + (C-1)\exp(-\frac{1}{C-1}\overrightarrow{\mathbf{w}}\cdot\overrightarrow{\mathbf{h}})\right) + \frac{1}{C}\sum_{c=1}^{C}\frac{\epsilon'_{1,c}}{C} + \epsilon'_2$$

$$= \log\left(1 + (C-1)\exp(-\frac{C}{C-1}\alpha\beta + \epsilon'_3)\right) + \frac{1}{C}\sum_{c=1}^{C}\frac{\epsilon'_{1,c}}{C} + \epsilon'_2$$

Consider $\log(1 + (C-1)\exp(-\frac{C\alpha\beta}{C-1} + \epsilon'_3))$: Let $\gamma' = (C-1)\exp(-\frac{C\alpha\beta}{C-1})$

$$\begin{aligned}
\log(1 + (C-1)\exp(-\frac{C\alpha\beta}{C-1} + \epsilon'_3)) &= \log(1 + (C-1)\exp(-\frac{C\alpha\beta}{C-1})\exp(\epsilon'_3)) \\
&= \log(1 + (C-1)\exp(-\frac{C\alpha\beta}{C-1})\exp(\epsilon'_3)) \\
&= \log(1 + \gamma'\exp(\epsilon'_3)) \\
&\geq \log(1 + \gamma'(1 + \epsilon'_3)) \\
&= \log(1 + \gamma' + \gamma'\epsilon'_3) \\
&\approx \log(1 + \gamma') + \frac{\gamma'}{1+\gamma'}\epsilon'_3
\end{aligned}$$

Thus using the fact that $1 + (C-1)\exp(-\frac{C\alpha\beta}{C-1}) \leq C$

$$\mathcal{L} \geq \log(1 + (C-1)\exp(-\frac{C\alpha\beta}{C-1})) + \frac{1}{C}\sum_{c=1}^{C}\frac{\epsilon'_{1,c}}{C} + \epsilon'_2 + \frac{\gamma'}{1+\gamma'}\epsilon'_3$$

$$\epsilon \geq \frac{1}{C}\sum_{c=1}^{C}\frac{\epsilon'_{1,c}}{C} + \epsilon'_2 + \frac{\gamma'}{1+\gamma'}\epsilon'_3$$

Note that while we do not know how ϵ is distributed among the different gaps, all the bounds involving $\epsilon'_{1,c}, \epsilon'_2, \epsilon'_3$ always hold in the worst case scenario subject to the constraint $\epsilon \geq \frac{1}{C}\sum_{c=1}^{C}\frac{\epsilon'_{1,c}}{C} + \epsilon'_2 + \frac{\gamma'}{1+\gamma'}\epsilon'_3$. Note that $\|\tilde{\mathbf{h}}_c\| \leq \alpha$, therefore $(\mathbf{w}_{c'} - \mathbf{w}_c)\tilde{\mathbf{h}}_c \geq -C\alpha\beta$, and the second-order derivative of $\log(1 + (C-1)\exp(x))$ is

$$\frac{(C-1)\exp(x)}{(1+(C-1)\exp(x))^2}$$

, which is $e^{-O(\alpha\beta)}$ for any $x \in [-C\alpha\beta, C\alpha\beta]$. Therefore, the function $\log(1 + (C-1)\exp(x))$ is λ-strongly-convex for $\lambda = e^{-O(\alpha\beta)}$ Thus, for any subset $S \subseteq [C]$, let $\delta = \frac{|S|}{C}$, by Lemma 2.4.1:

$$\begin{aligned}
-\frac{C}{C-1}\sum_{c\in S}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c &\leq \delta C(-\frac{1}{C-1}\overrightarrow{\mathbf{w}}\cdot\overrightarrow{\mathbf{h}}) + C\sqrt{\frac{2\epsilon_2'\delta(1-\delta)}{\lambda}} \\
\sum_{c\in S}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c &\geq \delta\overrightarrow{\mathbf{w}}\cdot\overrightarrow{\mathbf{h}} - (C-1)\sqrt{\frac{2\epsilon_2'\delta(1-\delta)}{\lambda}} \\
\sum_{c\in S}\alpha_c\beta_c &= \sum_{c\in[C]}\alpha_c\beta_c - \sum_{c\notin S}\alpha_c\beta_c \\
&\leq \sum_{c\in[C]}\alpha_c\beta_c - \sum_{c\notin S}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c \\
&\leq C\alpha\beta - \sum_{c\notin[C]-S}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c \\
&\leq C\alpha\beta - (1-\delta)\overrightarrow{\mathbf{w}}\cdot\overrightarrow{\mathbf{h}} + (C-1)\sqrt{\frac{2\epsilon_2'\delta(1-\delta)}{\lambda}}
\end{aligned}$$

Let $\alpha_c = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{h}_{c,i}\|^2}$ and $\beta_c = \|\dot{\mathbf{w}}_c\|$. Note that since $-\frac{1}{C-1}\overrightarrow{\mathbf{w}}\cdot\overrightarrow{\mathbf{h}} = -\frac{C}{C-1}\alpha\beta + \epsilon_3'$, there is $\overrightarrow{\mathbf{w}}\cdot\overrightarrow{\mathbf{h}} = C\alpha\beta - (C-1)\epsilon_3'$. Therefore,

$$\begin{aligned}
\sum_{c\in S}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c &\geq \delta C\alpha\beta - \delta(C-1)\epsilon_3' - (C-1)\sqrt{\frac{2\epsilon_2'\delta(1-\delta)}{\lambda}} \\
\sum_{c\in S}\alpha_c\beta_c &\leq \delta C\alpha\beta + (1-\delta)(C-1)\epsilon_3' + (C-1)\sqrt{\frac{2\epsilon_2'\delta(1-\delta)}{\lambda}}
\end{aligned}$$

Therefore, there are at most δC classes for which

$$\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c \leq \alpha\beta - \frac{(C-1)}{C}\epsilon_3' - \frac{C-1}{C}\sqrt{\frac{2\epsilon_2'(1-\delta)}{\delta\lambda}} \tag{2.19}$$

and also there are at most δC classes for which

$$\alpha_c\beta_c \geq \alpha\beta + \frac{(1-\delta)(C-1)}{\delta C}\epsilon_3' + \frac{C-1}{C}\sqrt{\frac{2\epsilon_2'(1-\delta)}{\delta\lambda}} \tag{2.20}$$

Thus, for at least $(1-2\delta)C$ classes, there is

$$\frac{\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c}{\alpha_c\beta_c} \geq 1 - (\frac{C-1}{C\alpha\beta})(\frac{\epsilon_3'}{\delta} - 2\sqrt{\frac{2\epsilon_2'(1-\delta)}{\delta\lambda}}) \tag{2.21}$$

By setting $\epsilon_2' = \epsilon$ and $\epsilon_3' = 0$, we get the following upper bound on

$$\cos_\angle(\dot{\mathbf{w}}_c,\ \tilde{\tilde{\mathbf{h}}}_c) \geq 1 - 2\sqrt{\frac{2\epsilon(1-\delta)}{\delta\lambda}}$$

Using $\lambda = e^{-O(\alpha\beta)}$ gives the NC3 bound in the theorem. Therefore, applying lemma Lemma 2.4.5 these classes, there is

$$\text{intra}_c \geq 1 - 4(\frac{C-1}{C\alpha\beta})(\frac{\epsilon_3'}{\delta} - 2\sqrt{\frac{2\epsilon_2'(1-\delta)}{\delta\lambda}})$$

Assuming that $\epsilon \ll 1$, then $\epsilon \ll \sqrt{\epsilon}$. Therefore, then worst case bound when $\epsilon \geq \epsilon_2' + \frac{\gamma'}{1+\gamma'}\epsilon_3'$ is achieved when $\epsilon_2' = \epsilon$:

$$\text{intra}_c \geq 1 - 8(\frac{C-1}{C\alpha\beta})\sqrt{\frac{2\epsilon(1-\delta)}{\delta\lambda}}$$

Plug in $\lambda = \exp(-O(\alpha\beta))$ and with simplification we get:

$$\text{intra}_c \geq 1 - \frac{\exp(O(\alpha\beta))}{\alpha\beta}\sqrt{\frac{128\epsilon}{\delta}} = 1 - O(\frac{e^{O(\alpha\beta)}}{\alpha\beta}\sqrt{\frac{\epsilon}{\delta}})$$

Now consider the inter-class cosine similarity. Let $m_c = -\frac{C}{C-1}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c$, by Lemma Lemma 2.4.3 we know that for any set S of $\delta(C-1)$ classes in $[C] - \{c\}$, using the definition that $\dot{\mathbf{w}}_c = \mathbf{w}_c - \tilde{\mathbf{w}}$ there is

$$\sum_{c' \in S} (\dot{\mathbf{w}}_{c'} - \dot{\mathbf{w}}_c)\tilde{\mathbf{h}}_c = \sum_{c' \in S} (\mathbf{w}_{c'} - \mathbf{w}_c)\tilde{\mathbf{h}}_c \leq \delta(C-1)m_c + (C-1)\sqrt{\frac{2\delta\epsilon'_{1,c}}{\exp(m_c)}}$$

Therefore, for at least $(1-\delta)(C-1)$ classes, there is

$$(\dot{\mathbf{w}}_{c'} - \dot{\mathbf{w}}_c)\tilde{\mathbf{h}}_c \leq m_c + \sqrt{\frac{2\epsilon'_{1,c}}{\exp(m_c)\delta}} = -\frac{C}{C-1}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c + \sqrt{\frac{2\epsilon'_{1,c}}{\exp(m_c)\delta}} \tag{2.22}$$

$$\dot{\mathbf{w}}_{c'}\tilde{\mathbf{h}}_c \leq -\frac{1}{C-1}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c + \sqrt{\frac{2\epsilon'_{1,c}}{\exp(m_c)\delta}} \tag{2.23}$$

Combining with (Equation 2.19) (Equation 2.20), we get that there are at least $(1-2\delta)C \times (1-3\delta)C \geq (1-5\delta)C^2$ pairs of classes c, c' that satisfies the following: for both c and c', equations (Equation 2.19) (Equation 2.20) are not satisfied (i.e. satisfied in reverse direction), and (Equation 2.22) is satisfied for the pair c', c. Note that this implies

$$m_c = -\frac{C}{C-1}\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c \leq -\frac{C}{C-1}\alpha\beta + \epsilon'_3 + \sqrt{\frac{2\epsilon'_2(1-\delta)}{\delta\lambda}}$$

and

$$\dot{\mathbf{w}}_{c'}\tilde{\mathbf{h}}_c \leq -\frac{\alpha\beta}{C-1} + \frac{1}{C}(\epsilon'_3 + \sqrt{\frac{2\epsilon'_2(1-\delta)}{\delta\lambda}}) + \sqrt{\frac{2\epsilon'_{1,c}}{\exp(m_c)\delta}}$$

We now seek to simplify the above bounds using the constraint that $\epsilon \geq \frac{1}{C}\sum_{c=1}^{C}\frac{\epsilon'_{1,c}}{C} + \epsilon'_2 + \frac{\gamma'}{1+\gamma'}\epsilon'_3$. Note that $\epsilon \ll \sqrt{\epsilon}$, and both λ and $\exp(m_c)$ are $\exp(-O(\alpha\beta))$, therefore, we can achieve the maximum bound by setting $\epsilon'_{1,c} = \epsilon$,

$$\dot{\mathbf{w}}_{c'}\tilde{\mathbf{h}}_c \leq -\frac{\alpha\beta}{C-1} + \exp(O(\alpha\beta))\sqrt{\frac{2\epsilon}{\delta}}$$

Similarly, we can achieve the smallest bound on $\alpha_c\beta_c$ (the reverse of (Equation 2.20))by setting $\epsilon_2' = \epsilon$ and using $\lambda = \exp(-O(\alpha\beta))$ we get for both c and c'

$$\alpha_c\beta_c \leq \alpha\beta + \exp(O(\alpha\beta))\sqrt{\frac{2\epsilon}{\delta}}$$

and achieve the largest bound on $\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c$ (the reverse of (Equation 2.19)) by setting $\epsilon_2' = \epsilon$ we get for both c and c':

$$\dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c \leq \alpha\beta - \exp(O(\alpha\beta))\sqrt{\frac{2\epsilon}{\delta}}$$

Therefore, we can apply Lemma Lemma 2.4.6 with $\alpha = \alpha_c$, $\beta = \beta_c$, $\epsilon' = \alpha_c\beta_c - \dot{\mathbf{w}}_c\tilde{\mathbf{h}}_c \leq 2\exp(O(\alpha\beta))\sqrt{\frac{2\epsilon}{\delta}}$ bound to get:

$$\begin{aligned}\cos_\angle(\dot{\mathbf{w}}_{c'}, \tilde{\tilde{\mathbf{h}}}_c) &\leq -\frac{1}{C-1} + \frac{C}{C-1}\frac{\exp(O(\alpha\beta))}{\alpha\beta}\sqrt{\frac{2\epsilon}{\delta}} + 4(\frac{2\exp(O(\alpha\beta))}{\alpha\beta}\sqrt{\frac{2\epsilon}{\delta}})^{1/3} \\ &\leq -\frac{1}{C-1} + O(\frac{e^{O(\alpha\beta)}}{\alpha\beta}(\frac{\epsilon}{\delta})^{1/6})\end{aligned}$$

Where the last inequality is because $\frac{e^{O(\alpha\beta)}}{\alpha\beta} > 1, \frac{\epsilon}{\delta} < 1$. Finally, we derive an upper bound on $\cos_\angle(\tilde{\tilde{\mathbf{h}}}_{c'}, \tilde{\tilde{\mathbf{h}}}_c)$ and thus intra-class cosine similarity by combining the above bounds. Note that for $\frac{\pi}{2} < a < \pi$ and $0 < b < \frac{pi}{2}$ we have:

$$\begin{aligned}\cos(a-b) &= \cos(a)\cos(b) + \sin(a)\sin(b) \\ &\leq \cos(a) + \sin(b) \\ &\leq \cos(a) + \sqrt{1-\cos^2(b)} \\ &\leq cos(a) + \sqrt{1-2\cos(b)}\end{aligned}$$

by (Equation 2.21) we get that

$$\cos_\angle(\dot{\mathbf{w}}_{c'}, \tilde{\tilde{\mathbf{h}}}_{c'}) \geq 1 - (\frac{C-1}{C\alpha\beta})(\frac{\epsilon_3'}{\delta} - 2\sqrt{\frac{2\epsilon_2'(1-\delta)}{\delta\lambda}}) \geq 1 - \frac{\exp(O(\alpha\beta))}{\alpha\beta}\sqrt{\frac{2\epsilon}{\delta}}$$

Therefore,

$$
\begin{aligned}
\cos_\angle(\tilde{\bar{\mathbf{h}}}_{c'}, \tilde{\bar{\mathbf{h}}}_c) &\leq -\frac{1}{C-1} + O(\frac{e^{O(\alpha\beta)}}{\alpha\beta}(\frac{\epsilon}{\delta})^{1/6}) + O((\frac{e^{O(\alpha\beta)}}{\alpha\beta})^{1/2}(\frac{\epsilon}{\delta})^{1/4}) \\
&\leq -\frac{1}{C-1} + O(\frac{e^{O(\alpha\beta)}}{\alpha\beta}(\frac{\epsilon}{\delta})^{1/6})
\end{aligned}
$$

Since $\|\tilde{\bar{\mathbf{h}}}_c\| \leq 1$, there is

$$
\tilde{\bar{\mathbf{h}}}_{c'} \cdot \tilde{\bar{\mathbf{h}}}_c = \|\tilde{\bar{\mathbf{h}}}_{c'}\|\|\tilde{\bar{\mathbf{h}}}_c\| \cos_\angle(\tilde{\bar{\mathbf{h}}}_{c'}, \tilde{\bar{\mathbf{h}}}_c) \leq -\frac{1}{C-1} + O(\frac{e^{O(\alpha\beta)}}{\alpha\beta}(\frac{\epsilon}{\delta})^{1/6})
$$

Applying Lemma 2.4.4 shows the bound on inter-class cosine similarity. Note that although this bound holds only for $1-5\delta$ fraction of pairs of classes, changing the fraction to $1-\delta$ only changes δ by a constant factor and does not affect the asymptotic bound.

CHAPTER 3
EMPIRICAL RESULTS

In this chapter, we present empirical evidence to verify the predictions made by our theoretical results. Specifically, we investigate the role of Batch Normalization[14] and weight decay in the emergence of Neural Collapse, as well as the recent depth minimization phenomenon observed by [4].

3.1 Neural Collapse Emergence with Batch Normalization and Weight Decay

Our theoretical results predict that lower feature norms and weight Frobenius norm of the final layer result in stronger guarantees of intra-class and inter-class cosine similarity at a close-to-optimal cross-entropy loss. Specifically, if the model batch normalizes the last-layer features and applies weight decay to the last-layer weight as well as the weight for Batch Normalization, then both the α and the β terms are directly regularized using weight decay.

To empirically investigate the effect of batch normalization and weight decay, we distinguish between Batch Normalization with affine transformation, typically used in practice, and Batch Normalization without affine transformation. Specifically, non-affine BN refers to Batch Normalization with the $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ terms removed, and the formula for non-affine BN is

$$BN(\mathbf{v}_i)^{(k)} = \frac{v_i^{(k)} - \mu^{(k)}}{\sigma^{(k)}}$$

This is equivalent to fixing $\boldsymbol{\gamma} = \mathbf{1}$ and $\boldsymbol{\beta} = 0$ instead of using trainable parameters regularized using weight decay. We perform experiments on non-affine BN to investigate the effect of weight decay on the last-layer feature norms.

3.1.1 Experimental Setup

We experiment with three models: models with affine BN layers, models with non-affine BN layers, and models without BN layers. The affine BN and non-affine BN models have a batch normalization between every linear layer after activation except for the output layer. For each model type, we perform experiments with different weight decay values, including 0.0001, 0.0003, 0.0005, 0.001, 0.003, 0.007, and 0.01. For each experiment, we record the smallest intra-class cosine similarity among all classes and the largest inter-class cosine similarity among all pairs of classes for the last-layer features.

For each experiment, the model used is a 6-layer vanilla neural network (commonly referred to as multi-layer perceptron) with ReLU activation. In order to explicitly control the distribution complexity, we generated an artificial dataset using a 3-layer neural network of the same architecture and generated 10000 samples from a normal distribution. We used the SGD optimizer with a learning rate of 0.0679, which decays times during the training process by a factor of 0.1 and a momentum value of 0.9. The models are trained using batch size 128 and trained for 300 epochs. The models achieved convergence in all settings and achieved perfect training accuracy, except for weight decay 0.007 and 0.01 for the no BN setting, whose results are discarded and not included in the figure.

3.1.2 Experimental Results

As can be observed in Figure 3.1, our experimental results indicate that significant Neural Collapse structure only emerges in models with affine-BN with sufficiently large weight decay. Within the range of convergence, the degree of neural collapse also increases along with the weight decay value. This fits our theoretical predictions as larger weight decay results in small feature and weight norms and, thus, stronger guarantees of Neural Collapse at near-optimal values.

Our results also indicate that both the BN and its affine transformation are critical for the emergence of Neural Collapse under WD. Notably, without affine transformation, there

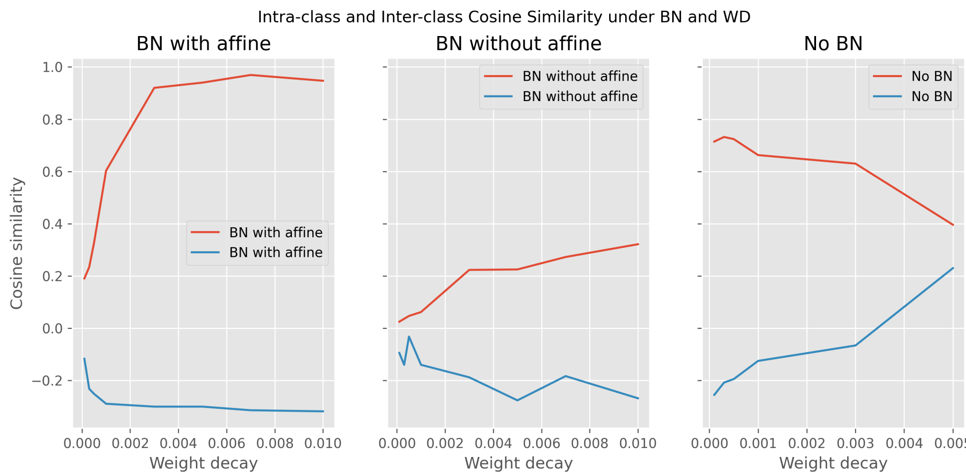

Figure 3.1: Intra-class and Inter-class Cosine Similarity for Neural Networks under Different WD and BN

is significantly less degree of neural collapse under all decay values despite perfect accuracy. Interestingly, in the no BN case, weight decay negatively impacts the emergence of Neural Collapse and fails to converge when $WD > 0.005$. This fits prior empirical results that Batch Normalization facilitates training convergences in deeper networks. Our results provide more insights into this observation by noting that BN allows the model to benefit positively from larger weight decay and thus form a successful classification structure.

To clarify our observations from the experiment above, we propose the following conjecture that may provide further directions for future theoretical work:

Conjecture 3.1.1 (NC improves with weight decay)**.** *Let $f(\mathbf{x};\theta) \in \mathbb{R}^C$ be any neural network classifier architecture with Batch Normalization before the last layer with combined weight θ. Let $S = \{\mathbf{x}_i, y_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^d, y_i \in [C]$ be a conic hull dataset composed of N samples. Let $\theta_{\lambda,S}$ denote the final weights of $f_\theta(x)$* **at convergence** *when training the training dataset S with weight decay parameter λ using gradient descent with an appropriate learning rate using the cross-entropy loss function. There exist small $\epsilon > 0$ such that for any $\lambda_1 < \lambda_2$, if both $f(\mathbf{x};\theta_{\lambda_1,S})$ and $f(\mathbf{x};\theta_{\lambda_2,S})$ classifies S with perfect accuracy, then*

$$\left(\min_{c\in[C]} \text{intra}_c^{\lambda_2} - \max_{c\neq c'\in[C]} \text{inter}_{c,c'}^{\lambda_2}\right) - \left(\min_{c\in[C]} \text{intra}_c^{\lambda_1} - \max_{c\neq c'\in[C]} \text{inter}_{c,c'}^{\lambda_1}\right) \geq -\epsilon \log(\frac{\lambda_2}{\lambda_1})$$

where intra_c^λ *denotes the average intra-class cosine similarity of class* c *using the model trained with weight decay parameter* λ*, and* $\mathrm{inter}_{c,c'}^\lambda$ *denotes the average inter-class cosine similarity between class* c *and* c' *using the model trained with weight decay parameter* λ*.*

The conic hull data set is an easily separable dataset where each class contains the area within an infinite cone with the vertex at the origin. Specifically, each class c in the conic hull dataset is defined using a set of vectors $\{\mathbf{b}_i\}_{i=1}^k$ as follows:

$$\mathcal{H}_c = \{\mathbf{x} \in \mathbb{R}^d | \forall i \in [k], \mathbf{x} \cdot \mathbf{b}_i > 0\}$$

and

$$\bigcup_{c=1}^{C} \mathcal{H}_c = \mathbb{R}^d$$

A simple example of a conic hull dataset with four classes in 2D can be illustrated in the following figure:

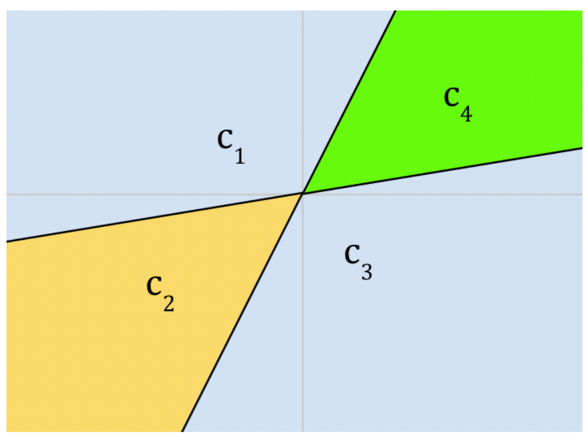

Figure 3.2: An illustration of a simple conic hull dataset with four classes in 2D. Grey lines denote the axis, and solid black lines denote the class decision boundaries

3.2 Depth Minimization: Neural Collapse across Intermediate Layers

Inspired by our experimental and theoretical results in the previous sections, we will also investigate the emergence of Neural Collapse across the intermediate layers of the network. A closely related work is the observation of the Depth Minimization phenomenon of

Galanti et al. [4]. Specifically, they observed that neural networks at the terminal phase of training have the implicit bias to minimize the effective depth of the network, as measured by the minimum number of layers from the input that allows the model to achieve near-perfect classification accuracy using the NCC decision rule. Based on their observation, we further investigate the effect of different weight decay parameters on this phenomenon and focus on not only the NCC classification accuracy but also the inter-class and intra-class cosine similarity across different layers of the network.

3.2.1 Experiemental Setup

We follow a similar experimental setup as the experiments in the previous section. However, in order to investigate this phenomenon in deep networks, we increased the training network depths to 15 while keeping the same training dataset. Batch normalization (with affine transformation) is applied after each layer except for the output. The optimizer and batch size hyperparameters are the same as described in the previous experiment.

For each experiment, we focus on the post-BN feature output of each layer. For each set of features, we record the smallest intra-class cosine similarity among all classes and the largest inter-class cosine similarity among all pairs of classes, as well as the classification accuracy when applying the NCC decision rule.

3.2.2 Experimental results

As shown in Figure 3.3, our experimental results support the findings of Galanti et al. [4] in Depth Minimization. We were able to recover the original results through the near-perfect NCC classification accuracy achieved in the early layers of the network, except when the weight decay parameter was set to 0.01, causing the model to fail to converge.

We also make two additional observations. First, our results show that with very few exceptions, the NCC classification accuracy, minimum average intra-class cosine similarity, as well as the difference between intra-class and inter-class cosine similarity all mono-

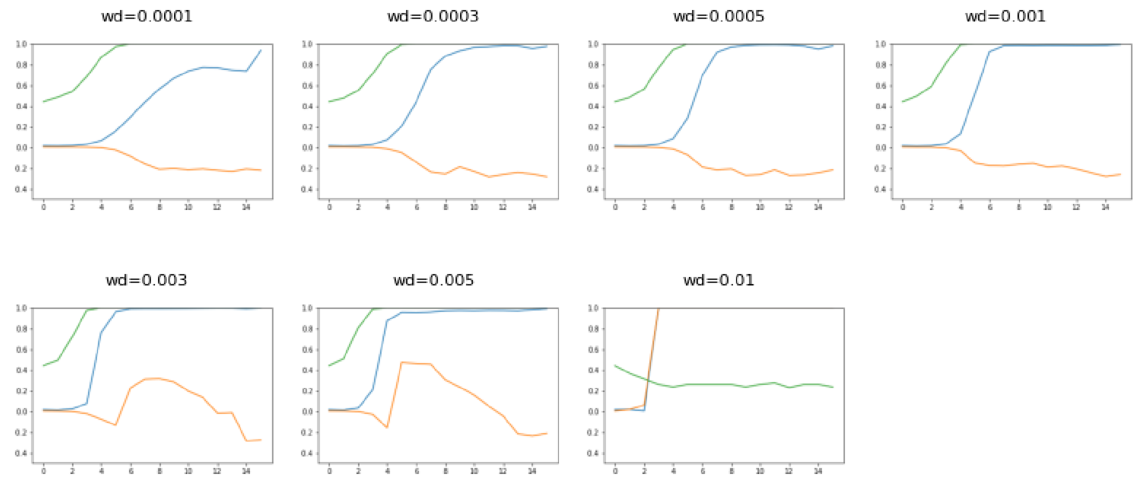

Figure 3.3: NC measurements of intermediate layers under different weight decay parameters. Green: NCC classification accuracy. Blue: Minimum intra-class cosine similarity. Yellow: Maximum inter-class cosine similarity

tonically increase as the number of layers from the input increases. This shows that the operation performed by each individual layer of the network makes the feature vectors of the layer closer to the optimal geometric configuration of Neural Collapse. Additionally, we have noticed that as the weight decay increases, the number of layers needed for the NCC classification to achieve near-perfect accuracy decreases. Note that these observations are only applicable when the model is capable of converging to a solution that attains perfect training accuracy.

In order mathematically formulate our experimental findings, we present the following two conjectures for trained MLP Neural Networks:

Definition 3.2.1. *A multi-layer perceptron (MLP) classifier with batch normalization of depth L is a function $f(\mathbf{x};\theta) \in \mathbb{R}^C$ parameterized by θ is defined as $f(\mathbf{x};\theta) = \phi^{(L)} \circ \phi^{(L-1)} \circ \cdots \circ \phi^{(1)}(\mathbf{x})$, where $\phi^{(l)}(\mathbf{h}^{(l-1)}) = BN(\max(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}, 0))$ for all non-ultimate layer $l < L$ and $\phi^{(L)}(\mathbf{h}^{(L-1)}) = \mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}$ for the final layer. θ denotes all trainable parameters in the network, including $\mathbf{W}^{(l)}, \mathbf{b}^{(l)}$ as well as $\boldsymbol{\gamma}^{(l)}$ and $\boldsymbol{\beta}^{(l)}$ from batch normalization layers.*

Definition 3.2.2 (ϵ-effective depth [4]). *Suppose we have a dataset S and a neural network $h = e \circ g^L \circ \cdots \circ g^1$ with $g^1 : \mathbb{R}^n \to \mathbb{R}^{p_2}, g^i : \mathbb{R}^{p_i} \to \mathbb{R}^{p_{i+1}}$ and linear classifier $e : \mathbb{R}^{p_{L+1}} \to \mathbb{R}^C$. Let $\hat{h}_i(x) := \arg\min_{c \in [C]} \| f_i(x) - \mu_{f_i}(S_c) \|$. The ϵ-effective depth $d_\epsilon^S(h)$ of the network h is the minimal value $i \in [L]$, such that, $\mathrm{err}_S\left(\hat{h}_i\right) \leq \epsilon$ (and $d_\epsilon^S(h) = L$ if such $i \in [L]$ is non-existent).*

Conjecture 3.2.1 (NC sharpens with model depth). *Let $f(\mathbf{x}; \theta) \in \mathbb{R}^C$ be a multi-layer perceptron classifier with batch normalization of depth L. Let $S = \{\mathbf{x}_i, y_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^d, y_i \in [C]$ be a conic hull dataset composed of N samples. Let $\theta_{\lambda,S}$ denote the final weights of $f(x;\theta)$* **at convergence** *when trained using the dataset S with weight decay parameter λ using gradient descent with an appropriate learning rate using the cross-entropy loss function. Let $\mathbf{h}^{(l)} = \phi^{(l)} \circ \phi^{(l-1)} \circ \cdots \circ \phi^{(1)}(\mathbf{x})$, then there exists small $\epsilon > 0$ such that for any $l \in [L-1]$,*

$$\left(\min_{c \in [C]} \mathrm{intra}_c^{(l+1)} - \max_{c \neq c' \in [C]} \mathrm{inter}_{c,c'}^{(l+1)}\right) - \left(\min_{c \in [C]} \mathrm{intra}_c^{(l)} - \max_{c \neq c' \in [C]} \mathrm{inter}_{c,c'}^{(l)}\right) \geq -\epsilon$$

where $\mathrm{intra}_c^{(l)}$ denotes the average intra-class cosine similarity of class c for the features $\mathbf{h}^{(l)}$, and $\mathrm{inter}_{c,c'}^{(l)}$ denotes the average inter-class cosine similarity between class c and c' for the features $\mathbf{h}^{(l)}$.

Conjecture 3.2.2 (Depth minimization improves with weight decay). *Let $f(\mathbf{x};\theta) \in \mathbb{R}^C$ be a multi-layer perceptron classifier with batch normalization of depth L. Let $S = \{\mathbf{x}_i, y_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^d, y_i \in [C]$ be a conic hull dataset composed of N samples. Let $\theta_{\lambda,S}$ denote the final weights of $f_\theta(x)$* **at convergence** *when training the training dataset S with weight decay parameter λ using gradient descent with an appropriate learning rate using the cross-entropy loss function. For some fixed small error ϵ and any $\lambda_1 < \lambda_2$, if both $f(\mathbf{x};\theta_{\lambda_1,S})$ and $f(\mathbf{x};\theta_{\lambda_2,S})$ classifies S with perfect accuracy,*

$$d_\epsilon^S(f_{\lambda_1,S}) \geq d_\epsilon^S(f_{\lambda_2,S})$$

, *where* $f_{\lambda,S}(\mathbf{x}) = f(\mathbf{x};\theta_{\lambda,S})$ *the model trained using weight decay* λ *and dataset* S*.*

Interestingly, our experiments also revealed that the maximum inter-class cosine similarity becomes unstable when the weight decay parameter increases. For example, when the weight decay parameter was set to 0.003 and 0.005, the inter-class cosine similarity exceeded zero in many intermediate layers, despite the NCC classification accuracy being perfect. This observation raises questions that require further empirical and theoretical investigations. Although we were unable to provide a theoretical justification for the above conjectures, our study's insights can contribute to future theoretical research.

Appendices

APPENDIX A
MINOR PROOFS

A.1 Proof of Proposition 2.2.1

Let $\boldsymbol{\gamma}$ be the scale vector for the Batch Normalization layer, and consider a single batch $\{\mathbf{x}_i\}_{i=1}^{B}$ be a batch of B vectors, and

$$h_i^{(k)} = \frac{x_i^{(k)} - \tilde{x}^{(k)}}{\sigma^{(k)}} \times \gamma^{(k)}$$

for all B. By the linearity of mean and standard deviation, $\hat{x}_i^{(k)} = \frac{x_i^{(k)} - \tilde{x}^{(k)}}{\sigma_{\mathbf{x}}^{(k)}}$ must have mean 0 and standard deviation 1. Therefore, $\sum_{i=1}^{B} \hat{x}_i^{(k)} = 0$ and $\frac{1}{B}\sum_{i=1}^{B}(\hat{x}_i^{(k)})^2 = 1$. Therefore, $\sum_{i=1}^{B}(h_i^{(k)})^2 = \sum_{i=1}^{B} \gamma^{(k)}(\hat{x}_i^{(k)})^2 = B(\gamma^{(k)})^2$, and $\sum_{i=1}^{B} \|\mathbf{h}_i\|^2 = \sum_{k=1}^{d}\sum_{i=1}^{B}(h_i^{(k)})^2 = \sum_{k=1}^{d}\sum_{i=1}^{B} \gamma^{(k)}(\hat{x}_i^{(k)})^2 = \sum_{k=1}^{d} B(\gamma^{(k)})^2 = B\|\boldsymbol{\gamma}\|^2$.

Consider a set of N vectors divided into m batches of size $\{B_j\}_{j=1}^{m}$. (This accounts for the fact that during training, the last mini-batch may have a different size than the other mini-batches if the number of training data is not a multiple of B). Then, $\sum_{i=1}^{N} \|\mathbf{h}_i\|^2 = \sum_{j=1}^{m}\sum_{i=1}^{B_j} \|\mathbf{h}_{j,i}\|^2 = \sum_{j=1}^{m} B_j\|\boldsymbol{\gamma}\|^2 = N\|\boldsymbol{\gamma}\|^2$. Therefore, $\sqrt{\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{h}_i\|^2} = \|\boldsymbol{\gamma}\|^2$

REFERENCES

[1] V. Papyan, X. Y. Han, and D. L. Donoho, "Prevalence of neural collapse during the terminal phase of deep learning training," *Proceedings of the National Academy of Sciences*, vol. 117, no. 40, pp. 24 652–24 663, 2020. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.2015509117.

[2] S. Bubeck *et al.*, *Sparks of artificial general intelligence: Early experiments with gpt-4*, 2023. arXiv: 2303.12712 `[cs.CL]`.

[3] T. Galanti, A. György, and M. Hutter, *On the role of neural collapse in transfer learning*, 2022. arXiv: 2112.15121 `[cs.LG]`.

[4] T. Galanti, L. Galanti, and I. Ben-Shaul, *On the implicit bias towards minimal depth of deep neural networks*, 2022.

[5] T. Poggio and Q. Liao, *Explicit regularization and implicit bias in deep network classifiers trained with the square loss*, 2020. arXiv: 2101.00072 `[cs.LG]`.

[6] D. G. Mixon, H. Parshall, and J. Pi, "Neural collapse with unconstrained features," *Sampling Theory, Signal Processing, and Data Analysis*, vol. 20, no. 11, 2022.

[7] X. Han, V. Papyan, and D. L. Donoho, "Neural collapse under mse loss: Proximity to and dynamics on the central path," in *International Conference on Learning Representations*, 2022.

[8] J. Lu and S. Steinerberger, "Neural collapse under cross-entropy loss," *Applied and Computational Harmonic Analysis*, vol. 59, pp. 224–241, 2022, Special Issue on Harmonic Analysis and Machine Learning.

[9] W. Ji, Y. Lu, Y. Zhang, Z. Deng, and W. J. Su, *An unconstrained layer-peeled perspective on neural collapse*, 2022. arXiv: 2110.02796 `[cs.LG]`.

[10] D. G. Mixon, H. Parshall, and J. Pi, *Neural collapse with unconstrained features*, 2020. arXiv: 2011.11619 `[cs.LG]`.

[11] Z. Zhu *et al.*, "A geometric analysis of neural collapse with unconstrained features," 2021. arXiv: 2105.02375 `[cs.LG]`.

[12] M. Xu, A. Rangamani, Q. Liao, T. Galanti, and T. Poggio, "Dynamics in deep classifiers trained with the square loss: Normalization, low rank, neural collapse, and generalization bounds," *RESEARCH*, vol. 6, p. 0024, 2023.

[13] V. Kothapalli, "Neural collapse: A review on modelling principles and generalization," *Transactions on Machine Learning Research*, 2023.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[15] N. Merentes and K. Nikodem, "Remarks on strongly convex functions," *Aequationes mathematicae*, vol. 80, no. 1, pp. 193–199, Sep. 2010.