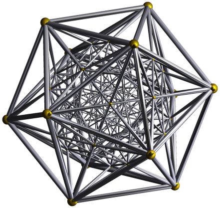
ACDL Summer Course 2023

Lecture 2: Low-Dimensional Structures in Deep Representation Learning I

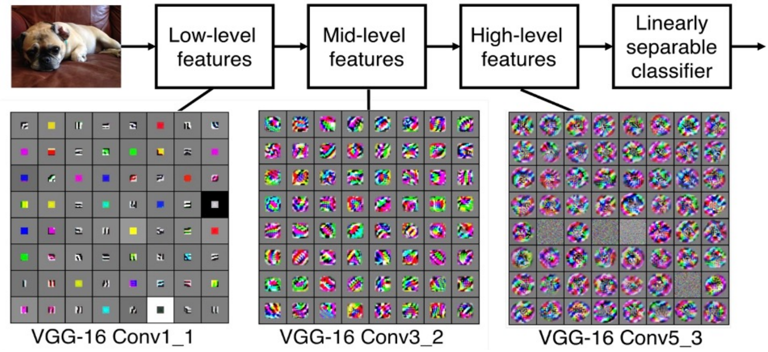
Qing Qu

EECS, University of Michigan

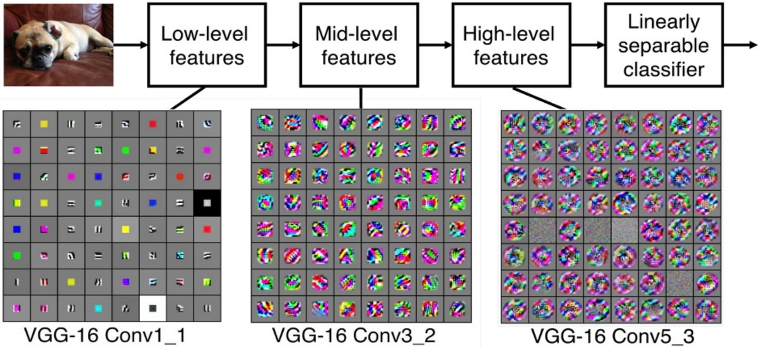
June 10th, 2023



What Representations are DNNs Designed to Learn?

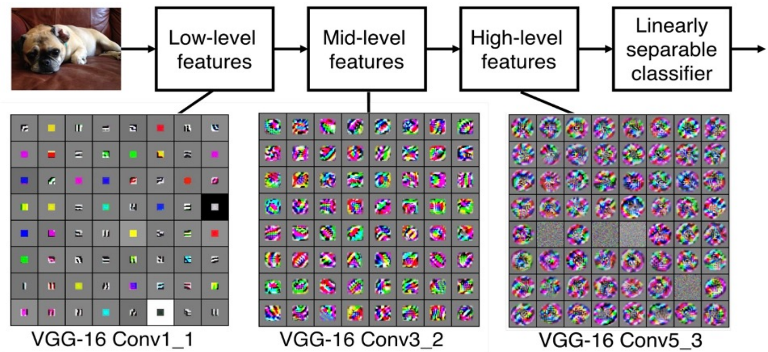


What Representations are DNNs Designed to Learn?



- **Wishful Design:** DNNs learn rich representations across different layers.

What Representations are DNNs Designed to Learn?



- **Wishful Design:** DNNs learn rich representations across different layers.
- **Reality:** Is it really the case in the practice of modern DNNs?

Outline

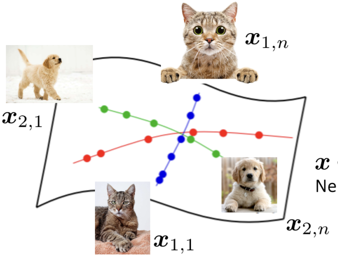
Multi-Class Image Classification Problem

- **Goal:** Learn a deep network predictor from a labelled training dataset $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)});\ i = 1, \cdots, n\}$.

[1] If not, we can use data augmentation to make them balanced

Multi-Class Image Classification Problem

- **Goal:** Learn a deep network predictor from a labelled training dataset $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)});\ i = 1, \cdots, n\}$.
- **Training Labels:** $k = 1, \dots, K$
  - $K = 10$ classes (MNIST, CIFAR10, etc)
  - $K = 1000$ classes (ImageNet)



Data in the input space

$\boldsymbol{x} \longrightarrow \boldsymbol{y}$

Neural network

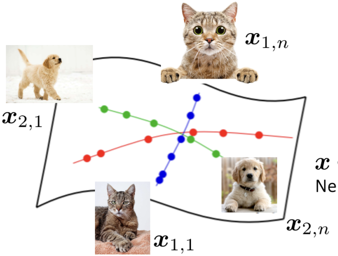
$$\begin{array}{cccc} \text{Cat} & \text{Dog} & & \text{Truck} \\ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} & \cdots & \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \end{array}$$

One-hot labeling vectors in $\mathbb{R}^K$

[1]If not, we can use data augmentation to make them balanced

Multi-Class Image Classification Problem

- **Goal:** Learn a deep network predictor from a labelled training dataset $\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)});\ i = 1, \cdots, n\}$.
- **Training Labels:** $k = 1, \ldots, K$
  - $K = 10$ classes (MNIST, CIFAR10, etc)
  - $K = 1000$ classes (ImageNet)



Data in the input space

$$\boldsymbol{x} \xrightarrow[\text{Neural network}]{} \boldsymbol{y}$$

$$\begin{array}{cccc} \text{Cat} & \text{Dog} & & \text{Truck} \\ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} & \cdots & \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \end{array}$$

One-hot labeling vectors in $\mathbb{R}^K$

- For simplicity, we assume **balanced** dataset where each class has n training samples.[1]

[1] If not, we can use data augmentation to make them balanced.

Deep Neural Network Classifiers

- **A vanilla deep network:**

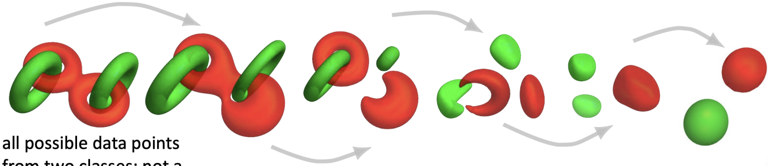
$$f_{\boldsymbol{\Theta}}(\boldsymbol{x}) \;=\; \underbrace{\boldsymbol{W}_L}_{\text{linear classifer } \boldsymbol{W}} \; \underbrace{\sigma\left(\boldsymbol{W}_{L-1}\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1)+\boldsymbol{b}_{L-1}\right)}_{\text{feature } \phi_{\boldsymbol{\theta}}(\boldsymbol{x})=:\boldsymbol{h}} + \boldsymbol{b}_L$$

Deep Neural Network Classifiers

- **A vanilla deep network:**

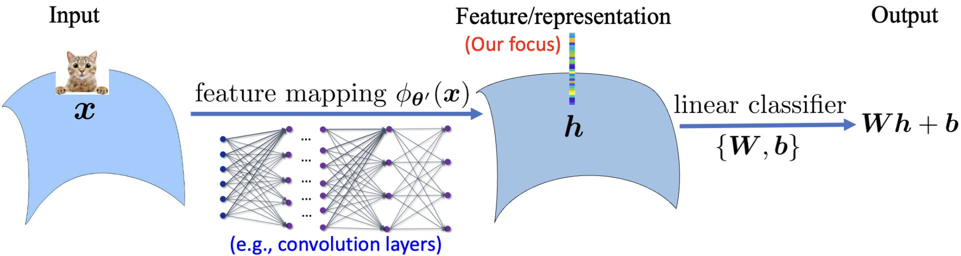
$$f_{\Theta}(\boldsymbol{x}) \;=\; \underbrace{\boldsymbol{W}_L}_{\textbf{linear classifer } \boldsymbol{W}} \; \underbrace{\sigma\left(\boldsymbol{W}_{L-1}\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1)+\boldsymbol{b}_{L-1}\right)}_{\textbf{feature } \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) =: \boldsymbol{h}} + \boldsymbol{b}_L$$

- **Progressive linear separation through nonlinear layers:**



all possible data points from two classes; not a single input!

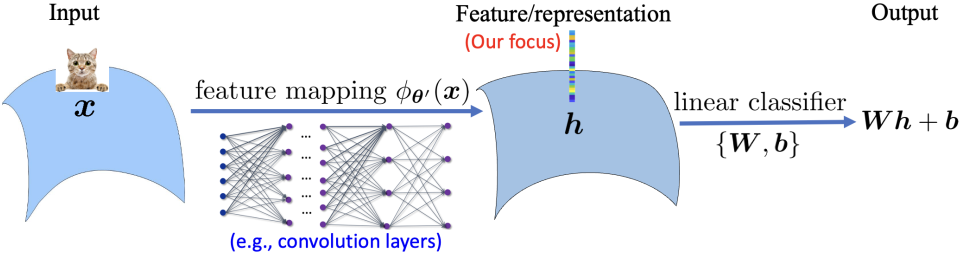
Deep Neural Network Classifiers



- **Training a deep neural network:**

$$\min_{\boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \underbrace{\mathcal{L}_{\mathrm{CE}}\left(\boldsymbol{W}\phi_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i}) + \boldsymbol{b}, \boldsymbol{y}_k\right)}_{\text{cross-entropy (CE) loss}} + \lambda \underbrace{\|(\boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b})\|_F^2}_{\text{weight decay}}$$

Deep Neural Network Classifiers



$$\text{Output: } f(\boldsymbol{x};\boldsymbol{\theta}) = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \xrightarrow{\text{Softmax function}} \begin{bmatrix} 0.6 \\ 0.3 \\ 0.1 \end{bmatrix} \begin{matrix} \text{Cat} \\ \text{Dog} \\ \text{Panda} \end{matrix} \quad \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Prediction (probability)   Target

$$\begin{aligned} \mathrm{CE}(\mathrm{Cat}) :&= -q(\mathrm{Cat}) \cdot \log p(\mathrm{Cat}) \\ &= -1 \cdot \log 0.6 \\ &= 0.51\ldots \end{aligned}$$

Neural Collapse in Multi-Class Classification

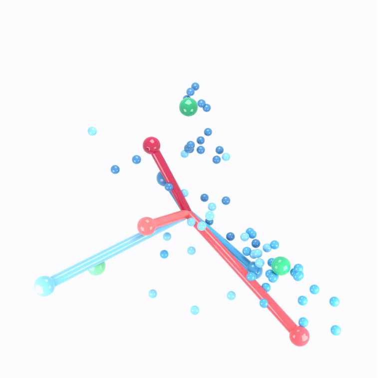
Prevalence of neural collapse during the terminal phase of deep learning training

Vardan Papyan, X. Y. Han, and David L. Donoho

+ See all authors and affiliations

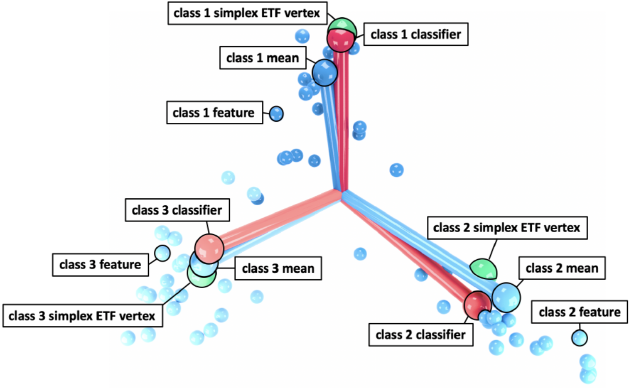
PNAS October 6, 2020 117 (40) 24652-24663; first published September 21, 2020;
https://doi.org/10.1073/pnas.2015509117

Contributed by David L. Donoho, August 18, 2020 (sent for review July 22, 2020; reviewed by Helmut Boelsckei and Stéphane Mallat)



- Reveals common outcome of learned features and classifiers across a variety of architectures and dataset
- Precise mathematical structure within the features and classifier

Neural Collapse in Multi-Class Classification



Credit: Han et al. Neural Collapse Under MSE Loss: Proximity to and Dynamics on the Central Path. ICLR, 2022.

Neural Collapse: Symmetry and Structures

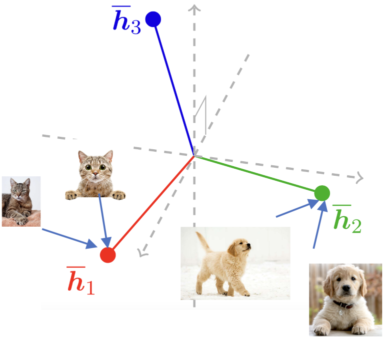
- **NC1: Within-Class Variability Collapse**: features of each class collapse to class-mean with **zero** variability:

$$k\text{-th class}, i\text{-th sample} : \boldsymbol{h}_{k,i} \to \overline{\boldsymbol{h}}_k,$$

Neural Collapse: Symmetry and Structures

- **NC1: Within-Class Variability Collapse**: features of each class collapse to class-mean with **zero** variability:

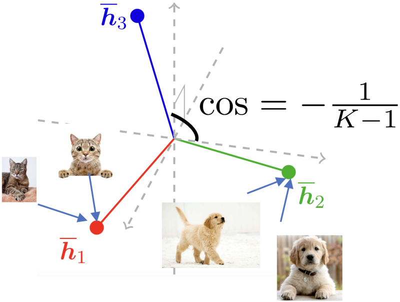
$$k\text{-th class}, i\text{-th sample} : \boldsymbol{h}_{k,i} \to \overline{\boldsymbol{h}}_k,$$



Neural Collapse: Symmetry and Structures

- **NC2: Convergence to Simplex Equiangular Tight Frame (ETF)**: the class means are linearly separable, and maximally distant

$$\frac{\langle \overline{\boldsymbol{h}}_k, \overline{\boldsymbol{h}}_{k'} \rangle}{\|\overline{\boldsymbol{h}}_k\| \|\overline{\boldsymbol{h}}_{k'}\|} \to \begin{cases} 1, & k = k' \\ -\frac{1}{K-1}, & k \neq k' \end{cases}$$

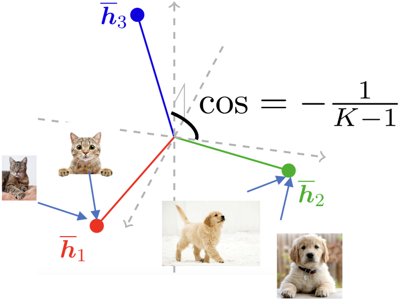


Neural Collapse: Symmetry and Structures

- **NC2: Convergence to Simplex Equiangular Tight Frame (ETF)**: the class means are linearly separable, and maximally distant

$$\overline{\boldsymbol{H}}^\top \overline{\boldsymbol{H}} \sim \boldsymbol{I}_K - \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top,$$

$$\overline{\boldsymbol{H}} = \begin{bmatrix} \overline{\boldsymbol{h}}_1 & \cdots & \overline{\boldsymbol{h}}_K \end{bmatrix}$$

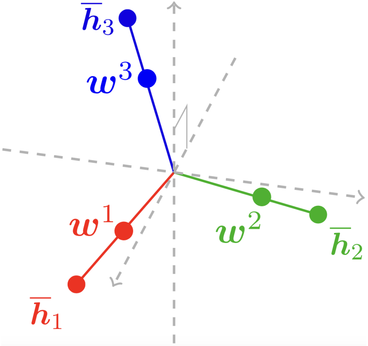


Neural Collapse: Symmetry and Structures

- **NC3: Convergence to Self-Duality**: the last-layer classifiers are perfectly matched with the class-means of features

$$\frac{\boldsymbol{w}^k}{\|\boldsymbol{w}^k\|} \to \frac{\overline{\boldsymbol{h}}_k}{\|\overline{\boldsymbol{h}}_k\|},$$

where $\boldsymbol{w}^k$ represents the k-th row of $\boldsymbol{W}$.



Understanding the Prevalence of Neural Collapse

Question. Given the prevalence of Neural Collapse across datasets and network architectures, why would such a phenomenon happen in training overparameterized networks?

Outline

Dealing with a Highly Nonconvex Problem

The training problem is highly **nonconvex** [Li et al.'18]:

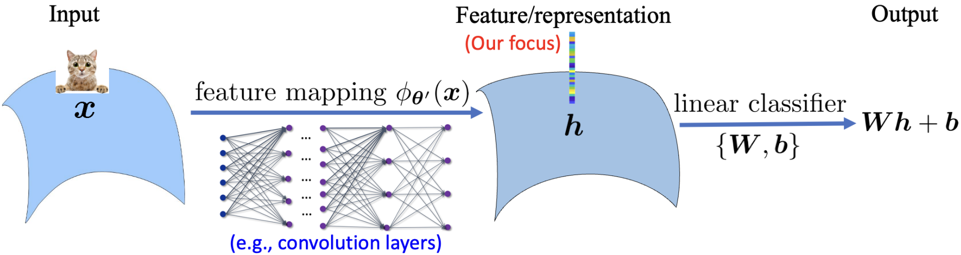
$$\min_{\boldsymbol{\theta}', \boldsymbol{W}, \boldsymbol{b}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}}\left(\boldsymbol{W} \phi_{\boldsymbol{\theta}'}(\boldsymbol{x}_{k,i}) + \boldsymbol{b}, \boldsymbol{y}_k\right) + \lambda \|(\boldsymbol{\theta}', \boldsymbol{W}, \boldsymbol{b})\|_F^2,$$

due to the fact that the network

$$f_{\boldsymbol{\Theta}}(\boldsymbol{x}) \;=\; \underbrace{\boldsymbol{W}_L}_{\textbf{linear classifer } \boldsymbol{W}} \underbrace{\sigma\left(\boldsymbol{W}_{L-1} \cdots \sigma(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_{L-1}\right)}_{\textbf{feature } \phi_{\boldsymbol{\theta}}(\boldsymbol{x}) =: \boldsymbol{h}} + \boldsymbol{b}_L$$

- **Nonlinear interaction across layers.**
- **Nonlinear activation functions.**

Simplification: Unconstrained Feature Model



Assumption. We treat $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}$ as a **free** optimization variable, ignoring the constraint $\boldsymbol{h}\phi_{\boldsymbol{\theta}}(\boldsymbol{x})$.

The Trend of Large Models...

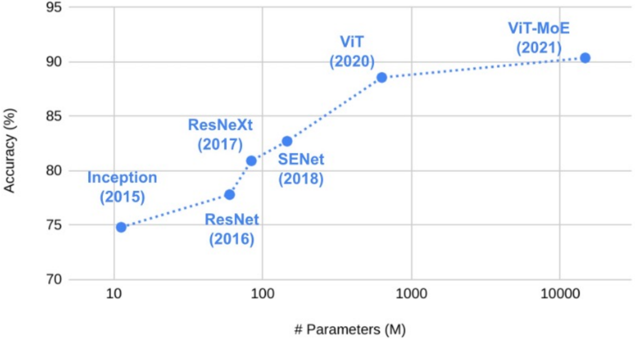
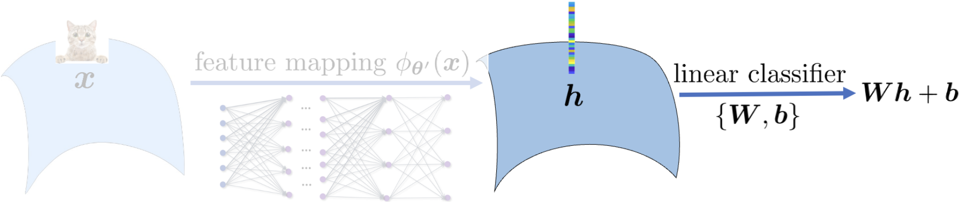


Figure: Accuracy vs. model size for image classification on ImageNet dataset

~23 million (# Parameters in ResNet-50) $\gg$ ~1 million (# Samples in ImageNet)

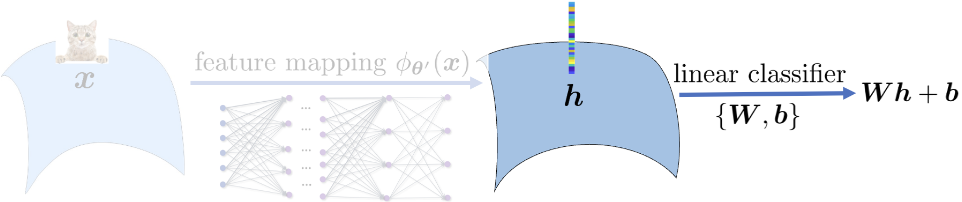
In principle, deep network can fit *any* training labels!
(*i.e.*, not only clean, but also corrupted labels)

Simplification: Unconstrained Feature Model



Assumption. We treat $H = \begin{bmatrix} h_{1,1} & \cdots & h_{K,n} \end{bmatrix}$ as a **free** optimization variable, ignoring the constraint $h\phi_{\theta}(x)$.

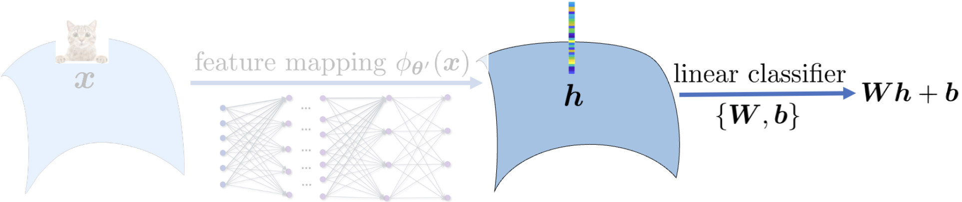
Simplification: Unconstrained Feature Model



Assumption. We treat $H = \begin{bmatrix} h_{1,1} & \cdots & h_{K,n} \end{bmatrix}$ as a **free** optimization variable, ignoring the constraint $h\phi_{\theta}(x)$.

- **Validity:** modern network are highly overparameterized, that they are **universal approximators** [Shaham'18];

Simplification: Unconstrained Feature Model

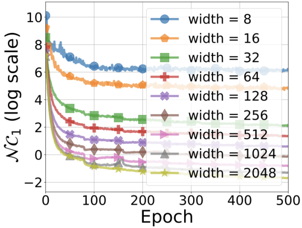


Assumption. We treat $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}$ as a **free** optimization variable, ignoring the constraint $\boldsymbol{h}\phi_{\boldsymbol{\theta}}(\boldsymbol{x})$.

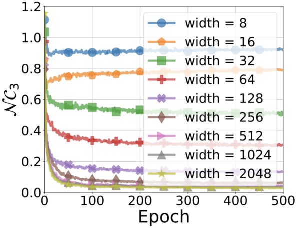
- **Validity:** modern network are highly overparameterized, that they are **universal approximators** [Shaham'18];
- **State-of-the-Art:** also called **Layer-Peeled Model** [Fang'21], existing work [E'20, Lu'20, Mixon'20, Fang'21] only studied global optimality conditions;

Experiments: NC Occurs on Random Labels/Inputs

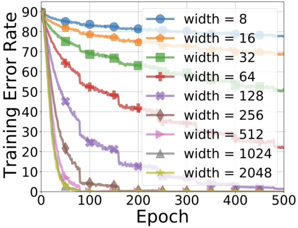
CIFAR-10 with **random** labels, MLP with **varying network widths**



Within-Class Variability (NC1)



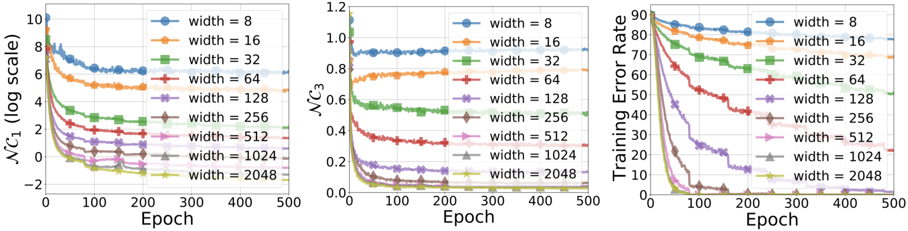
Self-Duality Collapse (NC2)



Training Error

Experiments: NC Occurs on Random Labels/Inputs

CIFAR-10 with **random** labels, MLP with **varying network widths**



Within-Class Variability (NC1) | Self-Duality Collapse (NC2) | Training Error

- **Validity of unconstrained features model**: Learn NC last-layer features and classifiers for any inputs
- The network memorizes training data in a very special way: NC
- We observe similar results on **random inputs (random pixels)**

Geometric Analysis of Global Landscape

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}+\boldsymbol{b},\boldsymbol{y}_k)+\frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2+\frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2+\frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2$$

Theorem (Global Optimality & Benign Global Landscape, Zhu et al.'21)

Let feature dimension d is larger than the class number K, i.e., $d > K$. Consider the above nonconvex optimization problem w.r.t. $(\boldsymbol{W},\boldsymbol{H})$. Then

- **Global optimality:** *Any global solution $(\{\boldsymbol{H}^\star,\boldsymbol{W}^\star,\boldsymbol{b}^\star\})$ obeys Neural Collapse, with $\boldsymbol{b}^\star = 0$ and*

$$\underbrace{\boldsymbol{h}_{k,i}^\star = \overline{\boldsymbol{h}}_k^\star,}_{\textit{NC1}} \quad \underbrace{\frac{\langle \overline{\boldsymbol{h}}_k^\star, \overline{\boldsymbol{h}}_{k'}^\star\rangle}{\|\overline{\boldsymbol{h}}_k^\star\|\|\overline{\boldsymbol{h}}_{k'}^\star\|} = \begin{cases} 1, & k = k' \\ -\frac{1}{K-1}, & k \neq k' \end{cases},}_{\textit{NC2}} \quad \underbrace{\frac{\boldsymbol{w}^{k\star}}{\|\boldsymbol{w}^{k\star}\|} = \frac{\overline{\boldsymbol{h}}_k^\star}{\|\overline{\boldsymbol{h}}_k^\star\|}}_{\textit{NC3}}$$

Geometric Analysis of Global Landscape

[Lu et al.'20] study the following one-example-per class model

$$\min_{\{\boldsymbol{h}_k\}} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{h}_k, \boldsymbol{y}_k), \text{ s.t.} \|\boldsymbol{h}_k\|_2 = 1$$

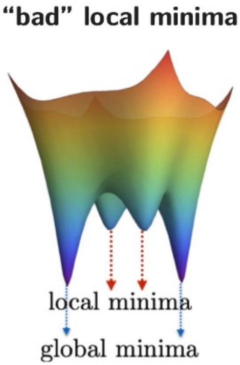
[E et al.'20, Fang et al.'21, Gral et al.'21, etc.] study constrained formulation

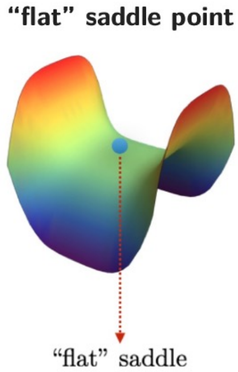
$$\min_{\{\boldsymbol{h}_{k,i}\}, \boldsymbol{W}} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}, \boldsymbol{y}_k), \text{ s.t. } \|\boldsymbol{W}\|_F \leq 1, \|\boldsymbol{h}_{k,i}\|_2 \leq 1$$

These work show that any global solution has NC, but

- What about local minima/saddle points?
- The constrained formulations are not aligned with practice

Global Optimitality Does Not Imply Efficient Optimization





Our loss is still highly nonconvex:

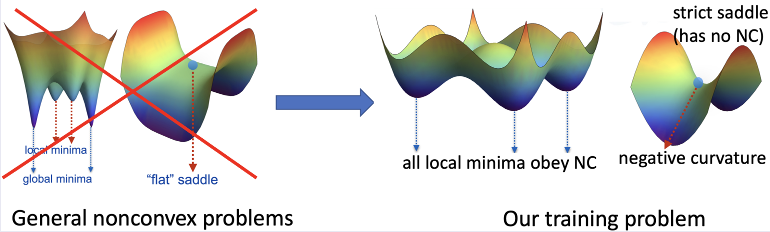
$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}+\boldsymbol{b},\boldsymbol{y}_k)+\frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2+\frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2+\frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2$$

Geometric Analysis of Global Landscape

Theorem (Global Optimality & Benign Global Landscape, Zhu et al.'21)

Let feature dimension d is larger than the class number K, i.e., $d > K$. Consider the above nonconvex optimization problem w.r.t. $(\boldsymbol{W}, \boldsymbol{H})$. Then

- **Global optimality:** *Any global solution $(\{\boldsymbol{H}^\star, \boldsymbol{W}^\star, \boldsymbol{b}^\star\})$ obeys Neural Collapse.*
- **Benign global landscape:** *The objective function (i) has no spurious local minima, and (ii) any non-global critical point is a strict saddle with negative curvature.*



Geometric Analysis of Global Landscape

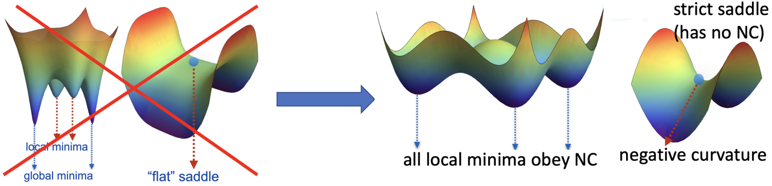
Theorem (Global Optimality & Benign Global Landscape, Zhu et al.'21)

Let feature dimension d is larger than the class number K, i.e., $d > K$. Consider the above nonconvex optimization problem w.r.t. $(\boldsymbol{W}, \boldsymbol{H})$. Then

- **Global optimality:** *Any global solution $(\{\boldsymbol{H}^\star, \boldsymbol{W}^\star, \boldsymbol{b}^\star\})$ obeys Neural Collapse.*
- **Benign global landscape:** *The objective function (i) has no spurious local minima, and (ii) any non-global critical point is a strict saddle with negative curvature.*

Message. Iterative algorithms such as (stochastic) gradient descent will always learn Neural Collapse features and classifiers.

Implications of Our Results

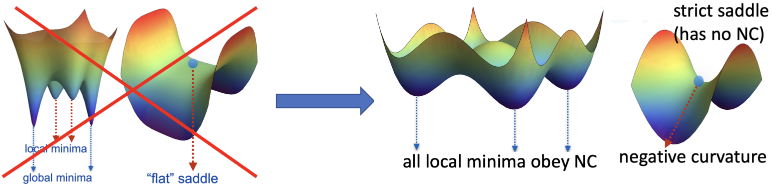


General nonconvex problems

Our training problem

- **A feature learing perspective.**
  - **Top down:** unconstrained feature model, representation learning, but no input information.
  - **Bottom up:** shallow network, strong assumptions, far from practice.

Implications of Our Results



General nonconvex problems

Our training problem

- **A feature learing perspective.**
  - **Top down:** unconstrained feature model, representation learning, but no input information.
  - **Bottom up:** shallow network, strong assumptions, far from practice.
- **Connections to empirical phenomena.**

Implications of Our Results

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}+\boldsymbol{b},\boldsymbol{y}_k)+\frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2+\frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2+\frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2$$

variational form: $\|\boldsymbol{Z}\|_* = \min_{\boldsymbol{Z}=\boldsymbol{W}\boldsymbol{H}} \frac{1}{2}(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{H}\|_F^2)$

- Closely relates to **low-rank matrix factorization** problems [Burer et al'03, Bhojanapalli et al'16, Ge et al'16, Zhu et al'18,Li et al'19, Chi et al'19]

Implications of Our Results

$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{Kn}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{CE}}(\boldsymbol{W}\boldsymbol{h}_{k,i}+\boldsymbol{b},\boldsymbol{y}_k)+\frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2+\frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2+\frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2$$

$$\text{variational form: } \|\boldsymbol{Z}\|_* = \min_{\boldsymbol{Z}=\boldsymbol{W}\boldsymbol{H}} \frac{1}{2}(\|\boldsymbol{W}\|_F^2+\|\boldsymbol{H}\|_F^2)$$

- Closely relates to **low-rank matrix factorization** problems [Burer et al'03, Bhojanapalli et al'16, Ge et al'16, Zhu et al'18,Li et al'19, Chi et al'19]
- However, we have more **structured** observation

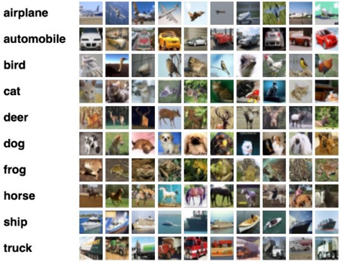
$$\boldsymbol{Y}=\begin{bmatrix}1 & \cdots & 1 & & & & & & \\ & & & 1 & \cdots & 1 & & & \\ & & & & & & 1 & \cdots & 1\end{bmatrix}=\boldsymbol{I}_K\otimes\boldsymbol{1}_n^\top$$

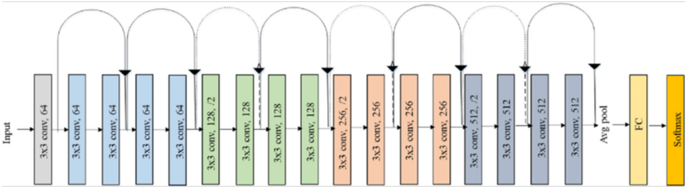
Experiments on Practical Neural Networks

Conduct experiments with **practical networks** to verify our findings:

Use a Residual Neural Network (ResNet) on CIFAR-10 Dataset:

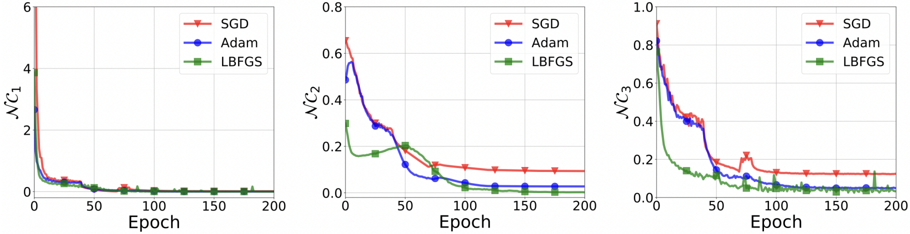
- $K = 10$ classes
- 50K training images
- 10K testing images





Experiments: NC is Algorithm Independent

ResNet18 on CIFAR-10 with **different training algorithms**



Within-Class Variability (NC1) Between-Class Separation (NC2) Self-Duality Collapse (NC3)

- The smaller the quantities, the severer NC
- NC is prevalent across **different training algorithms**

Exploit NC for Improving Training & Memory

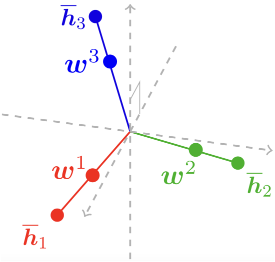
NC is prevalent, and classifier always converges to a Simplex ETF

- **Implication 1: No need to learn the classifier** [Hoffer et al. 2018]
  - Just fix it as a Simplex ETF
  - Save **8%, 12%, and 53%** parameters for ResNet50, DenseNet169, and ShuffleNet!

Exploit NC for Improving Training & Memory

NC is prevalent, and classifier always converges to a Simplex ETF

- **Implication 1: No need to learn the classifier** [Hoffer et al. 2018]
  - Just fix it as a Simplex ETF
  - Save **8%, 12%, and 53%** parameters for ResNet50, DenseNet169, and ShuffleNet!
- **Implication 2: No need of large feature dimension** d
  - Just use feature dim. $d = \#$class K (e.g., $d = 10$ for CIFAR-10)
  - Further saves **21% and 4.5%** parameters for ResNet18 and ResNet50!



Exploit NC for Improving Training & Memory

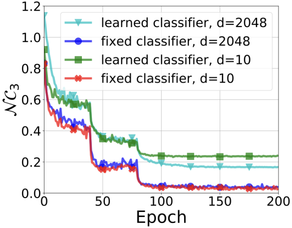
ResNet50 on CIFAR-10 with different settings

- **Learned** classifier (default) vs. **fixed** classifier as a simplex ETF
- Feature dim $d = 2048$ (default) vs. $d = 10$

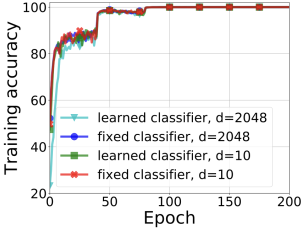
Exploit NC for Improving Training & Memory

ResNet50 on CIFAR-10 with different settings

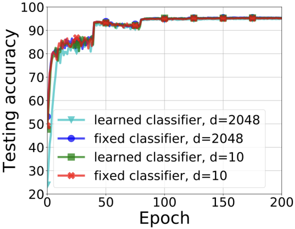
- **Learned** classifier (default) vs. **fixed** classifier as a simplex ETF
- Feature dim $d = 2048$ (default) vs. $d = 10$



Self-Duality Collapse (NC3)



Training Accuracy

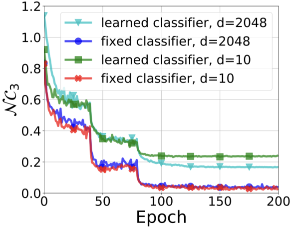


Testing Accuracy

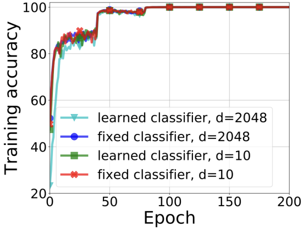
Exploit NC for Improving Training & Memory

ResNet50 on CIFAR-10 with different settings

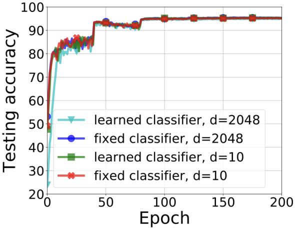
- **Learned** classifier (default) vs. **fixed** classifier as a simplex ETF
- Feature dim $d = 2048$ (default) vs. $d = 10$



Self-Duality Collapse (NC3)



Training Accuracy



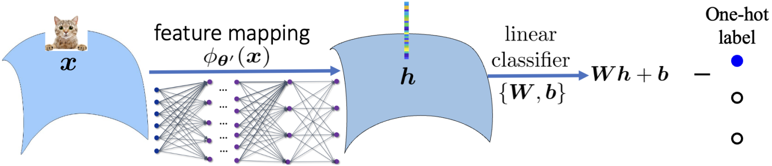
Testing Accuracy

- Training with small dimensional features and fixed classifiers achieves on-par performance with large dimensional features and learned classifiers.

Outline

Is Cross-entropy Loss Essential?

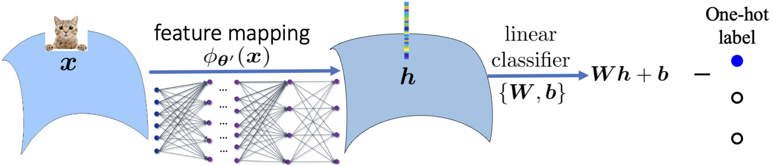
Question. Is cross-entropy loss essential to neural collapse?



[2] He et al., Bag of tricks for image classification with convolutional neural networks, CVPR'19.

Is Cross-entropy Loss Essential?

Question. Is cross-entropy loss essential to neural collapse?

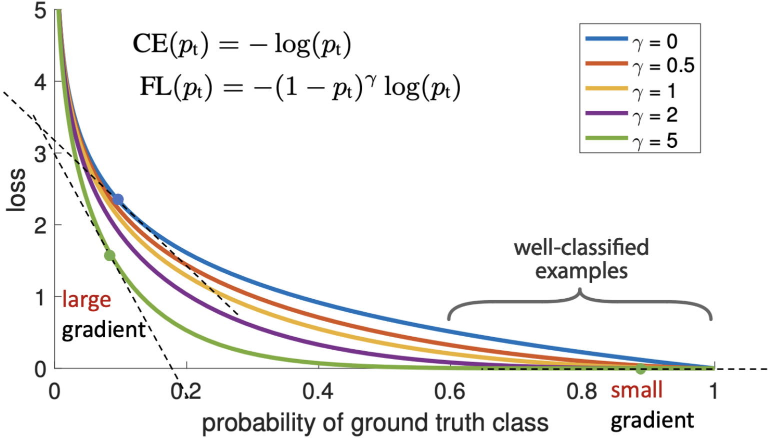


- We can measure the mismatch between the network output and the one-hot label in many ways.
- Various losses and tricks (e.g., label smoothing, focal loss) have been proposed to improve network training and performance[2]

[2] He et al., Bag of tricks for image classification with convolutional neural networks, CVPR'19.

Example I: Focal Loss (FL)

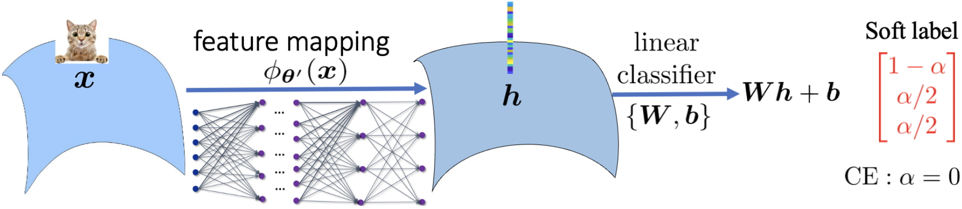
Focal loss puts more focus on hard, misclassified examples[3]



[3] Lin et al., Focal Loss for Dense Object Detection, CVPR'18.

Example II: Label Smoothing (LS)

Label smoothing replaces the hard label by a soft label[4]

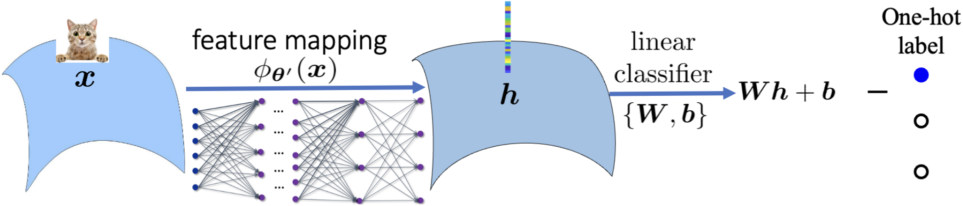


$$\text{Output: } \boldsymbol{W}\boldsymbol{h}+\boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \xrightarrow{\text{Softmax function}} \underbrace{\begin{bmatrix} 0.6 \\ 0.3 \\ 0.1 \end{bmatrix}}_{\text{Prediction}} \begin{matrix} \text{Cat} \\ \text{Dog} \\ \text{Panda} \end{matrix} \underbrace{\begin{bmatrix} 1-\alpha \\ \alpha/2 \\ \alpha/2 \end{bmatrix}}_{\text{Target}}$$

$$\begin{aligned} \mathrm{LS} = & - q(\text{Cat}) \cdot \log p(\text{Cat}) \\ & - q(\text{Dog}) \cdot \log p(\text{Dog}) \\ & - q(\text{Panda}) \cdot \log p(\text{Panda}) \\ = & - (1-\alpha) \log(0.6) \\ & - \frac{\alpha}{2} \log(0.3) \\ & - \frac{\alpha}{2} \log(0.1) \end{aligned}$$

[4] Szegedy et al., Rethinking the inception architecture for computer vision, CVPR'16. Muller, Kornblith, Hinton, When does label smoothing help?, NeurIPS'19.

Example III: Mean-squared Error (MSE) Loss



$$\text{Output: } \boldsymbol{W}\boldsymbol{h} + \boldsymbol{b} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{matrix} \text{Cat} \\ \text{Dog} \\ \text{Panda} \end{matrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

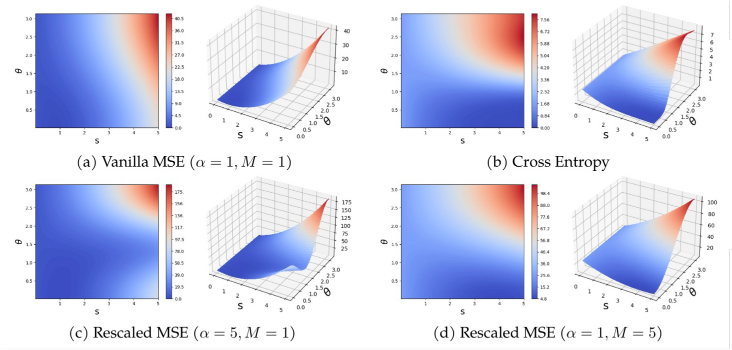
Prediction   Target

$$\text{MSE: } = (1 - 1)^2 + (0 - 0)^2 + (-1 - 0)^2$$

Compared with CE, **rescaled** MSE loss produces on par results for computer vision & NLP tasks.[5]

[5] Hui & Belkin, Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks, ICLR 2021.

Example III: Mean-squared Error (MSE) Loss

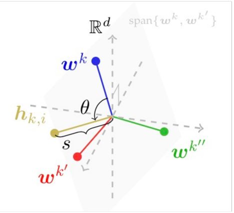


(a) Vanilla MSE ($\alpha = 1, M = 1$)

(b) Cross Entropy

(c) Rescaled MSE ($\alpha = 5, M = 1$)

(d) Rescaled MSE ($\alpha = 1, M = 5$)



$$\min_{\boldsymbol{W},\boldsymbol{H},\boldsymbol{b}} \frac{1}{2N} \|\boldsymbol{\Omega}_\alpha^{\odot 1/2} \odot \left(\boldsymbol{W}\boldsymbol{H} + \boldsymbol{b}\boldsymbol{1}^\top - M\boldsymbol{Y}\right)\|_F^2 + \frac{\lambda_{\boldsymbol{W}}}{2}\|\boldsymbol{W}\|_F^2 + \frac{\lambda_{\boldsymbol{H}}}{2}\|\boldsymbol{H}\|_F^2 + \frac{\lambda_{\boldsymbol{b}}}{2}\|\boldsymbol{b}\|_2^2.$$

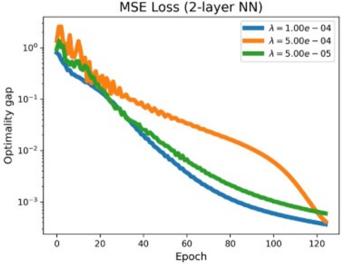
Example III: Mean-squared Error (MSE) Loss

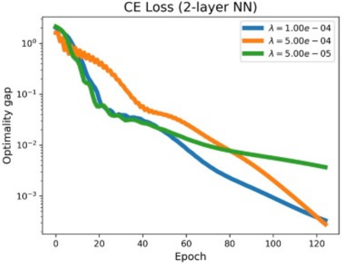
- Error bound condition for vanilla MSE loss:

$$\boxed{\operatorname{dist}((\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}), \mathcal{X}) \leq \kappa \|\nabla F(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})\|_F}$$

for any $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$ with $\operatorname{dist}((\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b}), \mathcal{X}) \leq \delta$.

- Local linear convergence of GD:





Which Loss is the Best to Use?

Testing accuracy (%) for WideResNet18 on mini-ImageNet with different widths and training iterations

Loss	CE	FL	LS	MSE
Width = $\times$ 0.25 Epoches = 200	71.95	70.20	70.40	69.15
Width = $\times$ 2 Epoches = 800	79.30	79.32	80.20	79.62

- The performance is also affected by the choice of network architecture, training iterations, dataset, etc.

Are All Loses Created Equal?—A NC Perspective

Theorem (Informal, Zhou et al.'22)

Under the unconstrained feature model, with feature dim. $d \geq$ *#class* $K - 1$, *for all the one-hot labeling based losses (e.g., CE, FL, LS, MSE),*

- *NC are the only global solutions for all losses.*
- *All losses have benign global landscape w.r.t.* $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$

Are All Losses Created Equal?—A NC Perspective

Theorem (Informal, Zhou et al.'22)

Under the unconstrained feature model, with feature dim. $d \geq \#$class $K - 1$, for all the one-hot labeling based losses (e.g., CE, FL, LS, MSE),

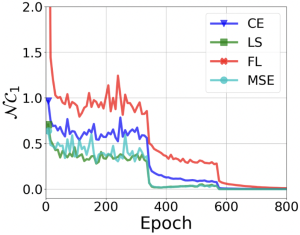
- *NC are the only global solutions for all losses.*
- *All losses have benign global landscape w.r.t. $(\boldsymbol{W}, \boldsymbol{H}, \boldsymbol{b})$*

Implication for practical networks If network is *large enough and trained longer enough*

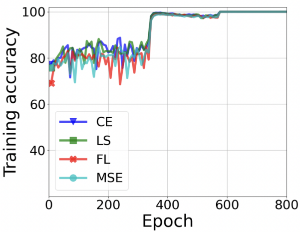
- All losses lead to largely identical features on training data—NC phenomena
- All losses lead to largely identical performance on test data (experiments in the following slides)

Are All Loses Created Equal?—A NC Perspective

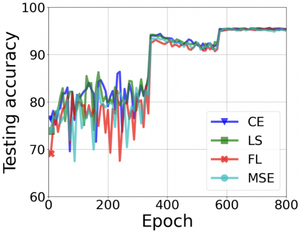
ResNet50 (with different network widths and training epoches) on CIFAR-10 with **different training losses**



Within-Class Variability (NC1)



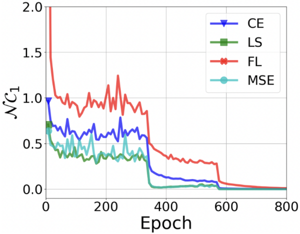
Train accuracy



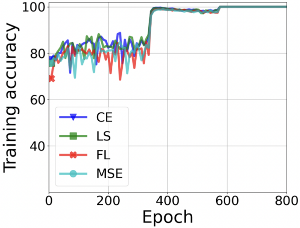
Testing accuracy

Are All Loses Created Equal?—A NC Perspective

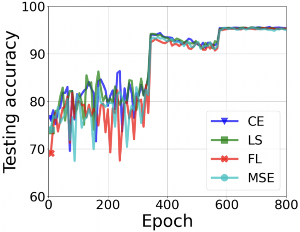
ResNet50 (with different network widths and training epoches) on CIFAR-10 with **different training losses**



Within-Class Variability (NC1)



Train accuracy

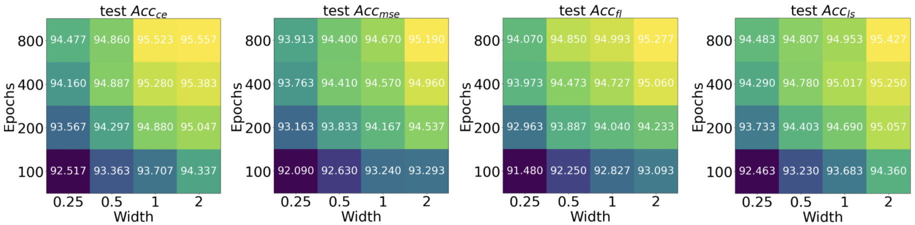


Testing accuracy

Observation: If network is *large enough and trained longer enough*, all losses lead to largely identical NC features on training data.

All Losses Are Almost Created Equal

ResNet50 (with different network widths and training epoches) on CIFAR-10 with **different training losses**

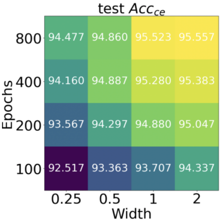


Cross-entropy | Mean-squared Error | Focal loss | Label smoothing

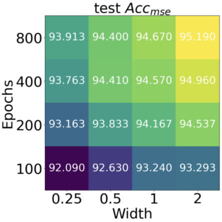
- Right top corners not only have better performance, but also have smaller variance than left bottom corners

All Losses Are Almost Created Equal

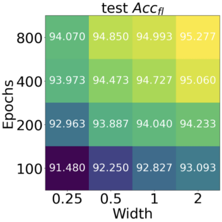
ResNet50 (with different network widths and training epoches) on CIFAR-10 with **different training losses**



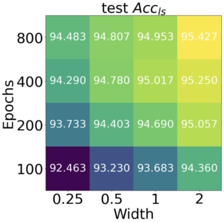
Cross-entropy



Mean-squared Error



Focal loss



Label smoothing

- Right top corners not only have better performance, but also have smaller variance than left bottom corners

Observation: If network is *large enough and trained longer enough*, all losses lead to largely identical performance on test data.

Neural Collapse with Feature Normalization

$$\min_{\boldsymbol{W},\boldsymbol{H}} \ \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}\left(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k\right)$$

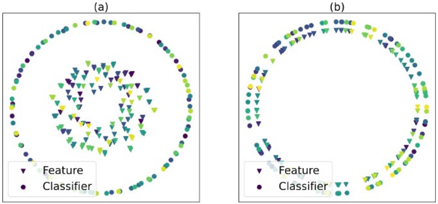
s.t. $\|\boldsymbol{w}_k\|_2 = \tau, \ \|\boldsymbol{h}_{k,i}\|_2 = 1, \ \boldsymbol{h}_{k,i} = \phi_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i}), \ \ \forall\, i \in [n], \forall\, k \in [K].$

Neural Collapse with Feature Normalization

$$\min_{\boldsymbol{W},\boldsymbol{H}} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}\left(\boldsymbol{W}\boldsymbol{h}_{k,i}, \boldsymbol{y}_k\right)$$

s.t. $\|\boldsymbol{w}_k\|_2 = \tau, \ \|\boldsymbol{h}_{k,i}\|_2 = 1, \ \boldsymbol{h}_{k,i} = \phi_{\boldsymbol{\theta}}(\boldsymbol{x}_{k,i}), \ \forall\, i \in [n], \forall\, k \in [K].$

- Improve the quality of learned features with larger class separation [Yu et al., 2020, Wang and Isola, 2020]
- Improve test performance in practice [Graf et al., 2021, Liu et al., 2021]



Neural Collapse with Feature Normalization

- Under the unconstrained feature model, a similar global landscape result can be shown for:

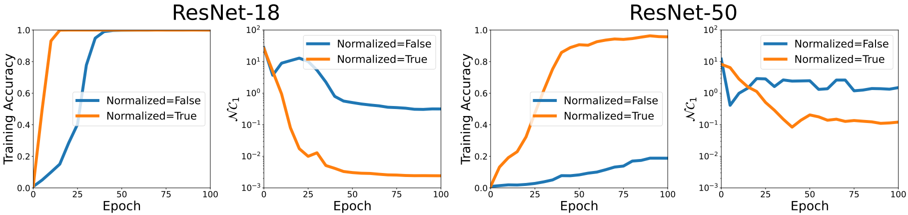
$$\min_{\boldsymbol{W},\boldsymbol{H}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}_{\mathrm{CE}}\left(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k\right)$$

$$\text{s.t. } \|\boldsymbol{w}_k\|_2 = \tau,\ \|\boldsymbol{h}_{k,i}\|_2 = 1,\ \forall\, i \in [n],\ \forall\, k \in [K].$$

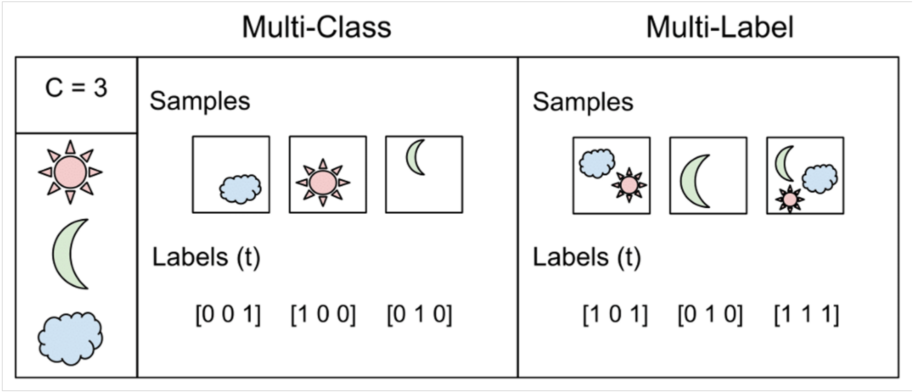
- More advanced analysis based upon Riemannian optimization tools.

Experimental Results with Feature Normalization

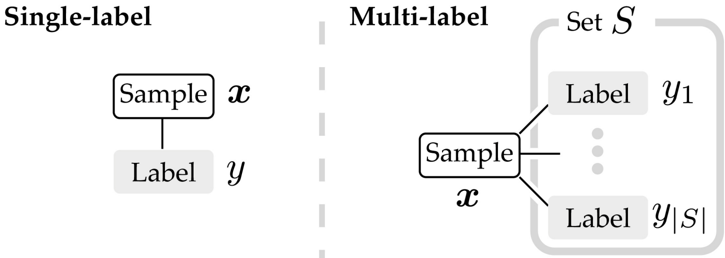
Faster training/feature collapse with RseNet on CIFAR100 with feature normalization



Neural Collapse for Multi-Label Learning



Multi-label Learning Setup



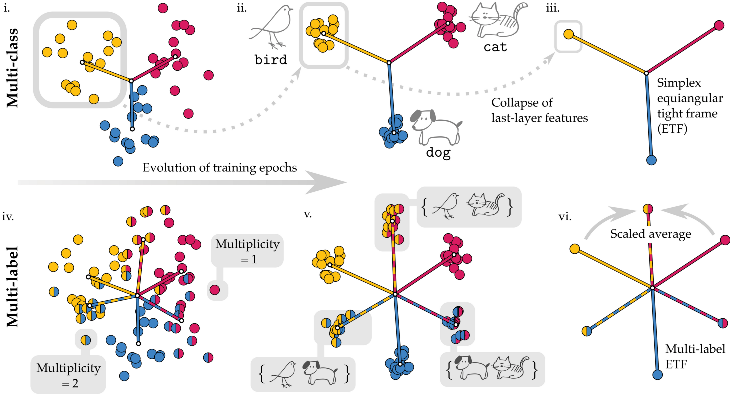
$$\mathcal{L}_{\mathrm{CE}}(\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}), y)$$

Loss

$$\sum_{i=1}^{|S|} \mathcal{L}_{\mathrm{CE}}(\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}), y_i)$$

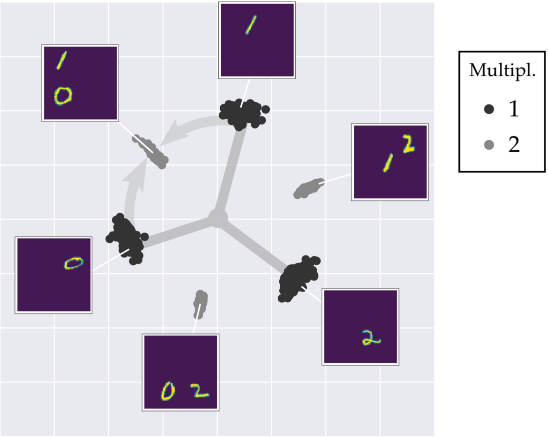
"Pick-all" Loss

Last-Layer Geometry of Multi-label Learning



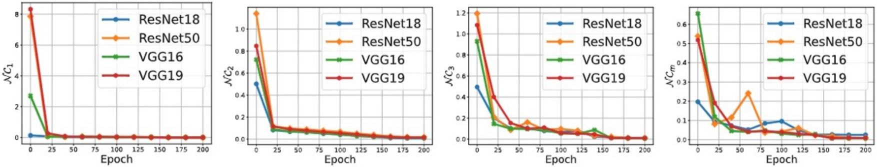
- Neural collapse in multi-label learning with 3 classes where the colors denote the class label;
- Respectively, left/mid/right panel shows representations during early/mid/late phase of training unconstrained feature model.

Multilabel-MNIST Synthetic Example

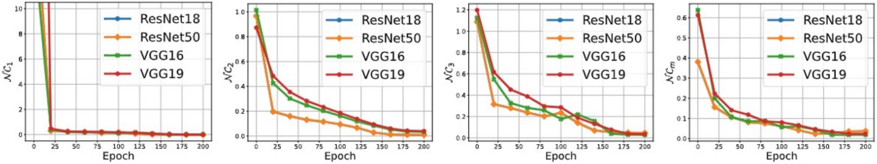


- Experiments with simple MLP architectures.
- The ETF structure still holds for data imbalancedness.

Neural Collapse for Multi-Label Learning



(a) $\mathcal{NC}_1$ (MLab-MNIST) (b) $\mathcal{NC}_2$ (MLab-MNIST) (c) $\mathcal{NC}_3$ (MLab-MNIST) (d) $\mathcal{NC}_m$ (MLab-MNIST)



(e) $\mathcal{NC}_1$ (MLab-Cifar10) (f) $\mathcal{NC}_2$ (MLab-Cifar10) (g) $\mathcal{NC}_3$ (MLab-Cifar10) (h) $\mathcal{NC}_m$ (MLab-Cifar10)

Outline

References

1 Z. Zhu*, T. Ding*, J. Zhou, X. Li, C. You, J. Sulam, and Q. Qu, A Geometric Analysis of Neural Collapse with Unconstrained Features, NeurIPS'2021 (spotlight, top 3%).

2 J. Zhou*, X. Li*, T. Ding, C. You, Q. Qu*, Z. Zhu*. On the Optimization Landscape of Neural Collapse under MSE Loss: Global Optimality with Unconstrained Features. ICML'2022.

3 C. Yaras*, P. Wang*, Z. Zhu, L. Balzano, Q. Qu, Neural Collapse with Normalized Features: A Geometric Analysis over the Riemannian Manifold. NeurIPS'2022.

4 J. Zhou, C. You, X. Li, K. Liu, S. Liu, Q. Qu, Z. Zhu. Are All Losses Created Equal? A Neural Collapse Perspective. NeurIPS'2022.

5 P. Wang*, H. Liu*, C. Yaras*, L. Balzano, Q. Qu. Linear Convergence Analysis of Neural Collapse with Unconstrained Features. NeurIPS OPT Workshop, 2022.

Conclusion and Coming Attractions

Learning common deep networks for low-dim structure

- **Low-dimensional features**: understand low-dim. features (sparse and neural collapse (NC)) learned in deep classifiers trained with one-hot labeling based losses in generic settings

Thank You! Questions?

Call for Papers

- IEEE JSTSP Special Issue on Seeking Low-dimensionality in Deep Neural Networks (SLowDNN) Manuscript Due: **Nov. 30, 2023**.
- Conference on Parsimony and Learning (CPAL) January 2024, Hongkong, Manuscript Due: **Aug. 28, 2023**.

SCAN ME

SCAN ME