UNIVERSITY OF MICHIGAN

Landscape Analysis of Overcomplete Tensor and Neural Collapse

Qing Qu

Dept. of EECS, University of Michigan

May 17, 2021

Outline of this Talk

- Introduction
- Overcomplete Tensor Decomposition (Representation Learning)
- Neural Collapse in Deep Network Training

Outline of this Talk

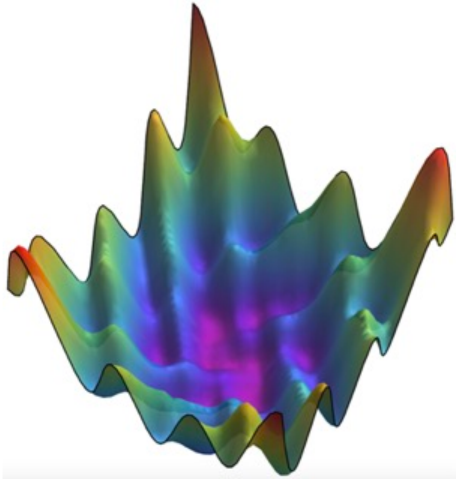
- **Introduction**

- Overcomplete Tensor Decomposition (Representation Learning)

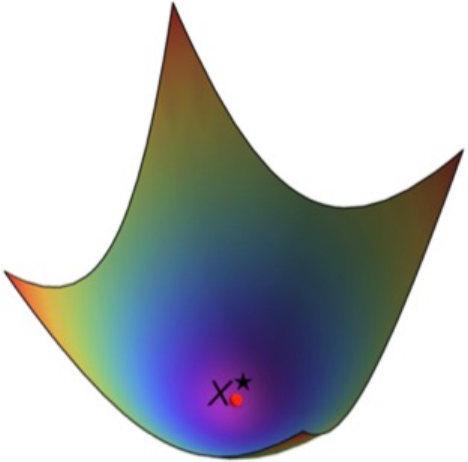
- Neural Collapse in Deep Network Training

Nonconvex Problems in Representation Learning

$$\boxed{\min_{\boldsymbol{x}} \; f(\boldsymbol{x}), \quad \text{s.t.} \;\; \boldsymbol{x} \in \mathbb{R}^n}$$

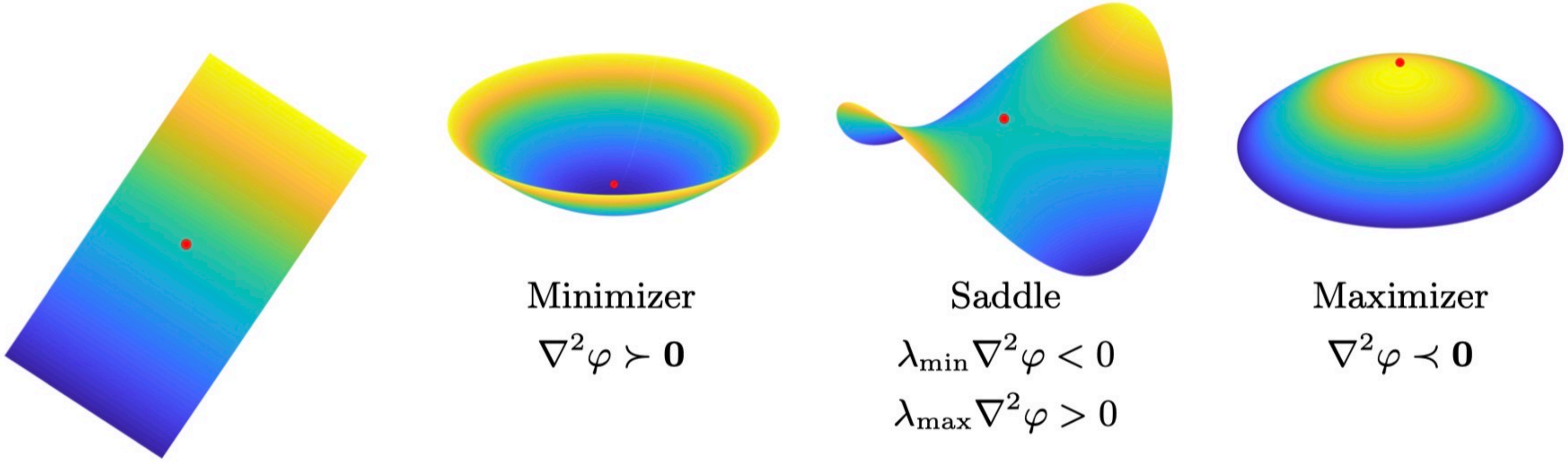


Nonconvex landscape



Convex landscape

General Nonconvex Problems



Minimizer

$\nabla^2 \varphi \succ \mathbf{0}$

Saddle

$\lambda_{\min} \nabla^2 \varphi < 0$

$\lambda_{\max} \nabla^2 \varphi > 0$

Maximizer

$\nabla^2 \varphi \prec \mathbf{0}$

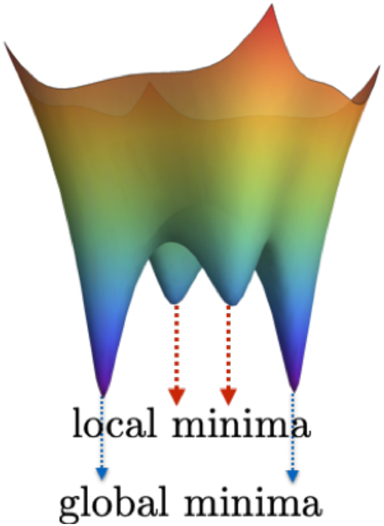
Noncritical Point ($\nabla \varphi \neq \mathbf{0}$)

Critical Points ($\nabla \varphi = \mathbf{0}$)

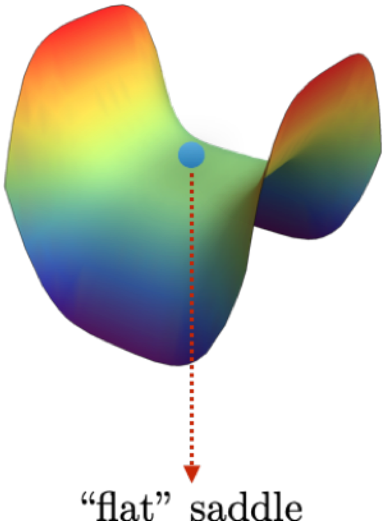
General Nonconvex Problems

$$\boxed{\min_{\boldsymbol{x}} \; f(\boldsymbol{x}), \quad \text{s.t.} \quad \boldsymbol{x} \in \mathbb{R}^n}$$

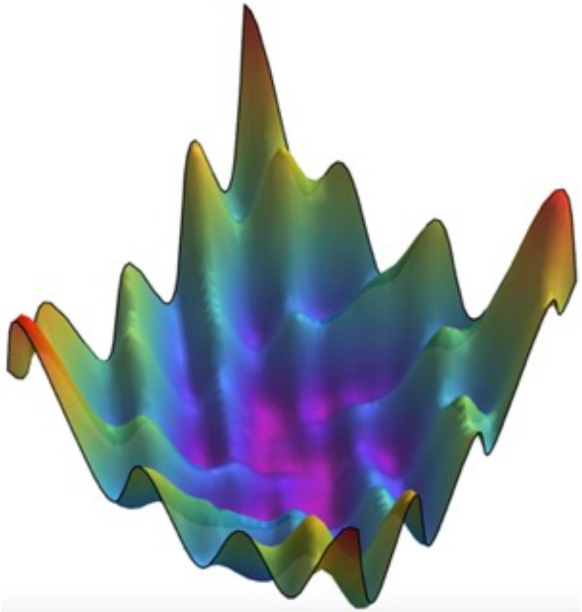
“bad” local minimizers



“flat” saddle points



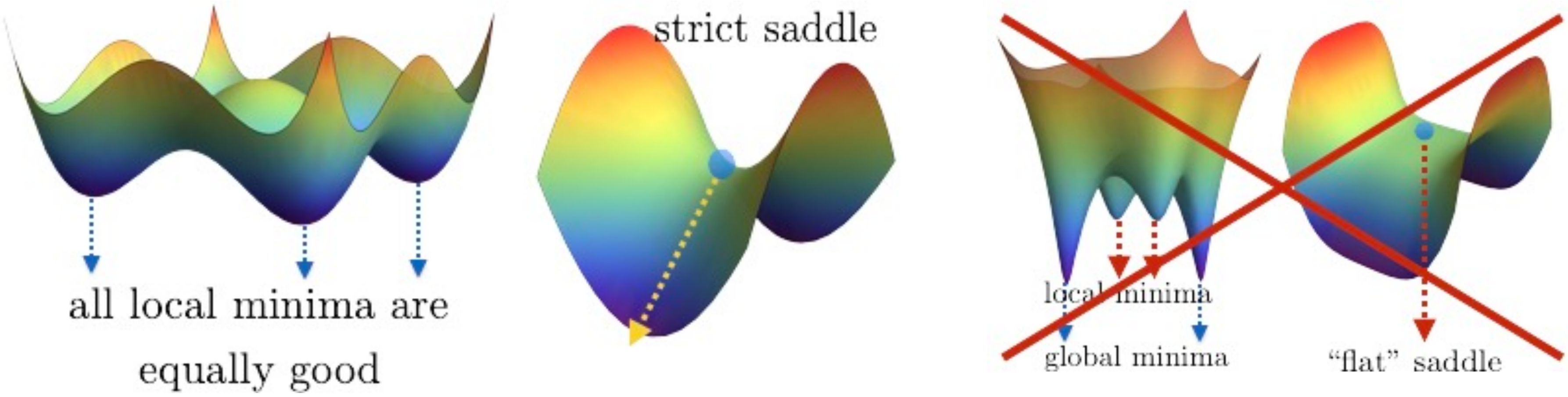
General Nonconvex Problems



$$\boxed{\min_{\boldsymbol{x}} \; f(\boldsymbol{x}), \quad \text{s.t.} \;\; \boldsymbol{x} \in \mathbb{R}^n}$$

In the worst case, even finding a local minimizer is NP-hard (Murty et al. 1987)

Optimizing Nonconvex Problems Globally



Benign nonconvex landscapes enable efficient global optimization!

Nonconvex Problems with Benign Landscape

- Generalized Phase Retrieval [Sun’18]
- Low-rank Matrix Recovery [Ma’16, Jin’17, Chi’19]
- (Convolutional) Sparse Dictionary Learning [Sun’16, Qu’20]
- (Orthogonal) Tensor Decomposition [Ge’15]
- Sparse Blind Deconvolution [Zhang’17, Li’18, Kuo’19]
- Deep Linear Network [Kawaguchi’16]
- …

Outline of this Talk

- Introduction

- **Overcomplete Tensor Decomposition (Representation Learning)**

- Neural Collapse in Deep Network Training

Landscape Analysis of Overcomplete Learning

Q. Qu, Y. Zhai, X. Li, Y. Zhang, Z. Zhu, Analysis of optimization landscapes for overcomplete learning, *ICLR'20*, (oral, top 1.9%)

- Provide the **global landscape** for overcomplete representation learning problems.
- Explains why they can be **efficiently** optimized to global optimality

Overcomplete Tensor Decomposition

We consider decomposing a 4-th order tensor of rank m in the following form

$$\mathcal{T} = \sum_{i=1}^{m} \boldsymbol{a}_i \otimes \boldsymbol{a}_i \otimes \boldsymbol{a}_i \otimes \boldsymbol{a}_i, \quad \boldsymbol{a}_i \in \mathbb{R}^n.$$

- Given $\mathcal{T}$, our goal is to recover each component $\boldsymbol{a}_i \in \mathbb{R}^n$.
- We are interested in the overcomplete regime that $m > n$.

Core problem for several **unsupervised representation** learning problems (ICA and mixture of Gaussian [Anandkumar'12], dictionary learning [Barak'14,Qu'20]), and even **training neural networks** [Ge'17].

Overcomplete Tensor Decomposition

A natural (nonconvex) objective to find one component

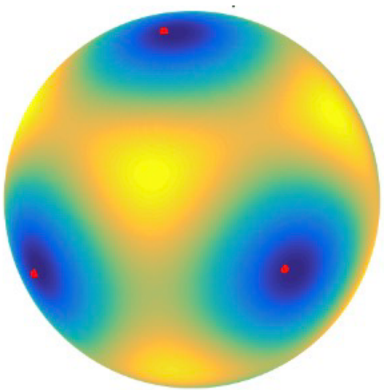
$$\min_{\boldsymbol{q}} \; f(\boldsymbol{q}) \;=\; -\sum_{i,j,k,\ell \in [m]^4} \mathcal{T}_{i,j,k,\ell} q_i q_j q_k q_\ell \;=\; -\sum_{i=1}^{n} \langle \boldsymbol{a}_i, \boldsymbol{q} \rangle^4$$

$$\text{s.t.} \quad \|\boldsymbol{q}\|_2 \;=\; 1.$$

Overcomplete Tensor Decomposition

Let $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_m \end{bmatrix}$, the problem can be written as

$$\boxed{\min_{\boldsymbol{q}} \; -\left\|\boldsymbol{A}^\top \boldsymbol{q}\right\|_4^4, \quad \text{s.t.} \quad \left\|\boldsymbol{q}\right\|_2 = 1.}$$

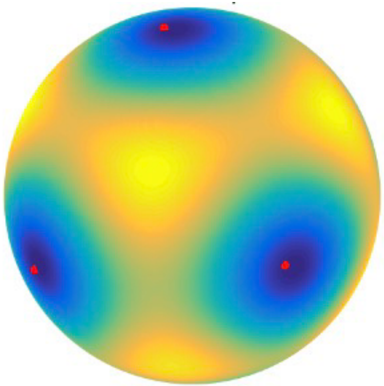


- When $m \leq n$, and $\{\boldsymbol{a}_i\}_{i=1}^m$ are **orthogonal**, existing result [Ge'15] has shown that the function is a **strict saddle function** with benign optimization landscape, all global solutions are approximately $\{\pm\boldsymbol{a}_i\}_{i=1}^m$.
- The analysis of orthogonal case **cannot** be generalized to overcomplete settings.

Overcomplete Tensor Decomposition

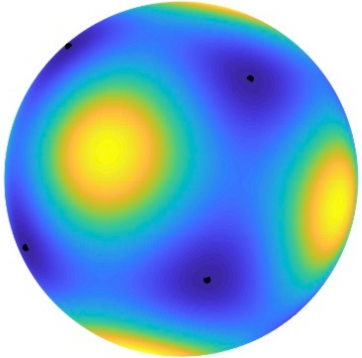
Let $\boldsymbol{A} = \begin{bmatrix} \boldsymbol{a}_1 & \cdots & \boldsymbol{a}_m \end{bmatrix}$, the problem can be written as

$$\boxed{\min_{\boldsymbol{q}} \; -\left\|\boldsymbol{A}^\top \boldsymbol{q}\right\|_4^4, \quad \text{s.t.} \quad \left\|\boldsymbol{q}\right\|_2 = 1.}$$



- For **overcomplete case**, most of existing landscape analysis results [Ge'17] are **local**, or are based on Sum-of-Squares relaxations [Barak'15, Ma'16] which is computationally expensive.
- Empirically, gradient descent or power method find the global solution **efficiently** even when $m \gg n$.

A Global Result in Overcomplete Settings



$$\boxed{\min_{\boldsymbol{q}} \; f(\boldsymbol{q}) \; = \; -\left\|\boldsymbol{A}^\top \boldsymbol{q}\right\|_4^4, \quad \text{s.t.} \quad \|\boldsymbol{q}\|_2 = 1.}$$

Theorem (Informal) *Suppose that (i) $K = m/n$ is a constant, and (ii) $\boldsymbol{A}$ is near orthogonal with small μ. Then every critical point of $f(\boldsymbol{q})$ is either*

- *a **strict saddle point** exhibits negative curvature;*
- *or close to a **target solution**: one column $\boldsymbol{a}_i$ of $\boldsymbol{A}$.*

Assumptions on $\boldsymbol{A}$ (Near Orthogonal)

- Row orthogonal: unit norm **tight frame** (UNTF)

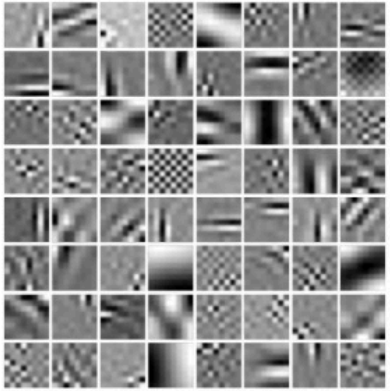
$$\sqrt{\frac{n}{m}}\boldsymbol{A}\boldsymbol{A}^\top = \boldsymbol{I}, \quad \|\boldsymbol{a}_i\|_2 = 1.$$

- **Incoherence** of the columns (near orthogonal)

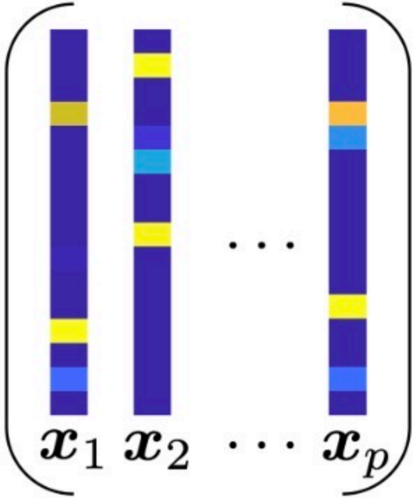
$$\max_{i \neq j} |\langle \boldsymbol{a}_i, \boldsymbol{a}_j \rangle| \leq \mu.$$

Relationship to Dictionary Learning

$\approx$



$\times$



$\boldsymbol{Y}$ $\boldsymbol{A}$ $\boldsymbol{X}$

Given $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} \in \mathbb{R}^{n \times p}$, jointly find overcomplete dictionary $\boldsymbol{A} \in \mathbb{R}^{n \times m}$ and sparse $\boldsymbol{X} \in \mathbb{R}^{m \times p}$.

Relationship to Dictionary Learning

We can find one column of $\boldsymbol{A}$ via

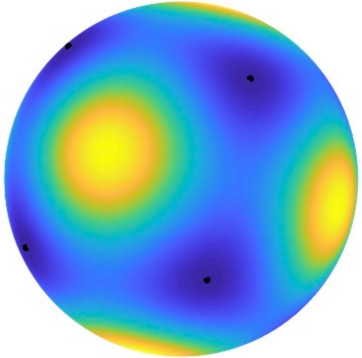
$$\boxed{\min_{\boldsymbol{q}} \; f_{DL}(\boldsymbol{q}) \;=\; -\left\|\boldsymbol{Y}^\top \boldsymbol{q}\right\|_4^4, \quad \text{s.t.} \quad \|\boldsymbol{q}\|_2 = 1.}$$

The underlying reasoning is that, in expectation

$$\mathbb{E}_{\boldsymbol{X}}\left[\left\|\boldsymbol{Y}^\top \boldsymbol{q}\right\|_4^4\right] \;=\; \mathbb{E}_{\boldsymbol{X}}\left[\left\|\boldsymbol{X}^\top \boldsymbol{A}^\top \boldsymbol{q}\right\|_4^4\right] \;=\; c_1 \left\|\boldsymbol{A}^\top \boldsymbol{q}\right\|_4^4 + c_2$$

for $\boldsymbol{X}$ following some sparse zero-mean distributions (e.g., Bernoulli-Gaussian)

Relationship to Dictionary Learning

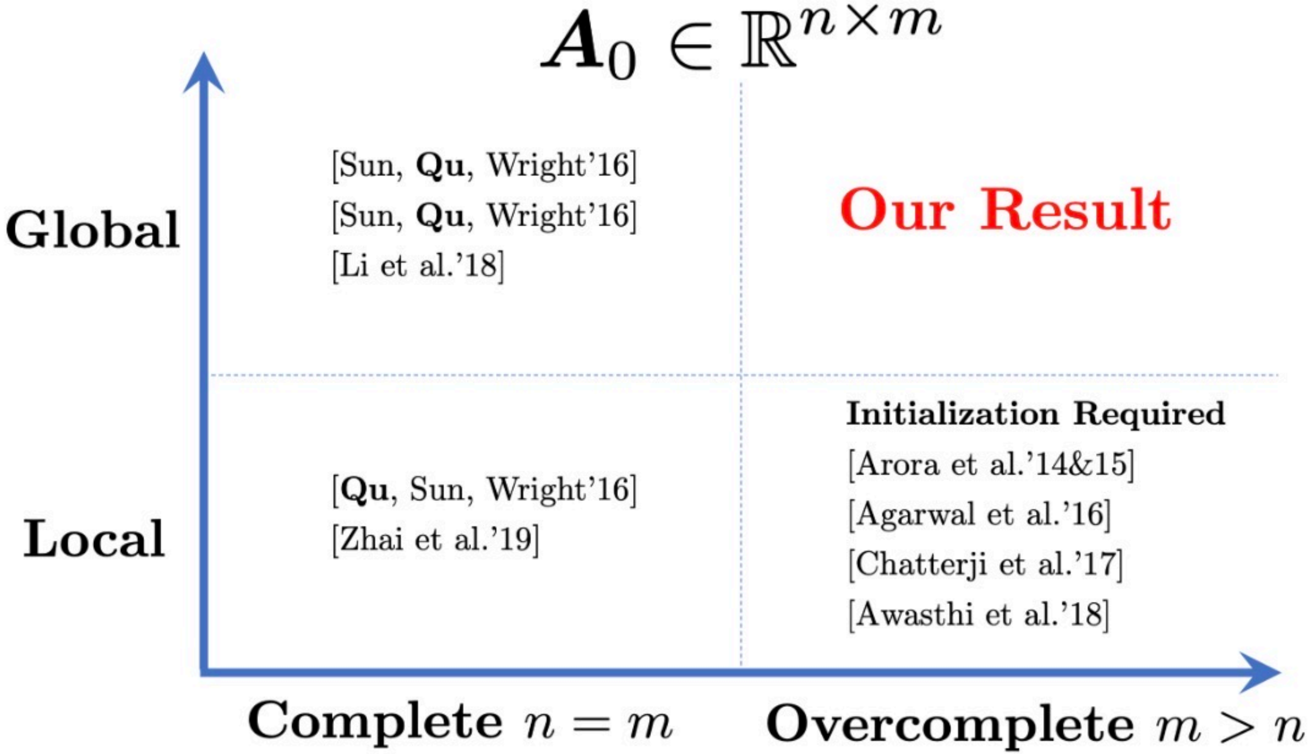


$$\boxed{\min_{\boldsymbol{q}} \; f_{DL}(\boldsymbol{q}) \; = \; -\left\|\boldsymbol{Y}^\top \boldsymbol{q}\right\|_4^4, \quad \text{s.t.} \quad \|\boldsymbol{q}\|_2 = 1.}$$

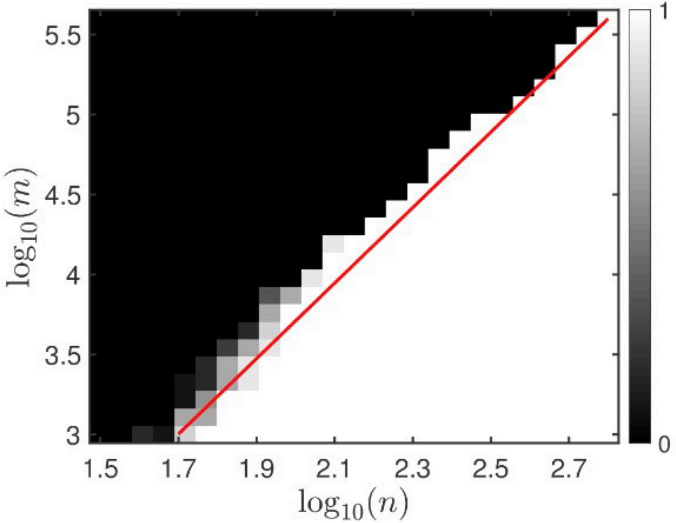
Theorem (Informal) *Suppose that (i) $K = m/n$ is a constant, (ii) $\boldsymbol{A}$ is near orthogonal, and (iii) $p \geq \Omega(\text{poly}(n))$. Then with high probability every critical point of $f(\boldsymbol{q})$ is either*

- *a **strict saddle point** exhibits negative curvature;*
- *or close to a **target solution**: one column $\boldsymbol{a}_i$ of $\boldsymbol{A}$.*

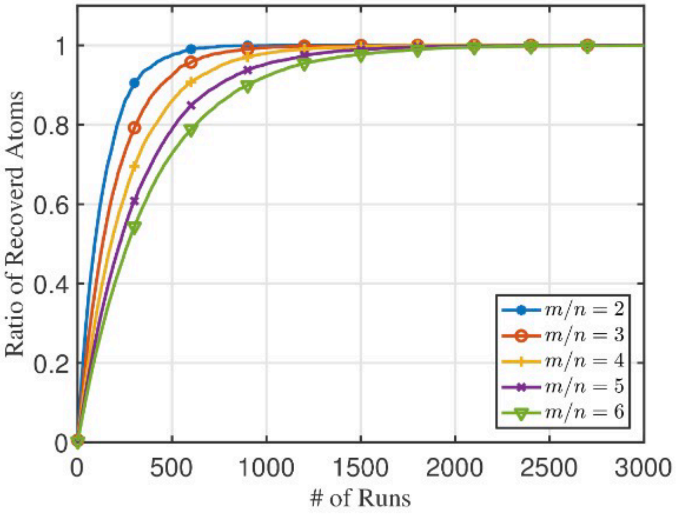
Relationship to Dictionary Learning



Relationship to Dictionary Learning



practice $m < n^2$
vs. theory $m < Cn$



recover full $\boldsymbol{A}_0$ via repeated independent trials

Outline of this Talk

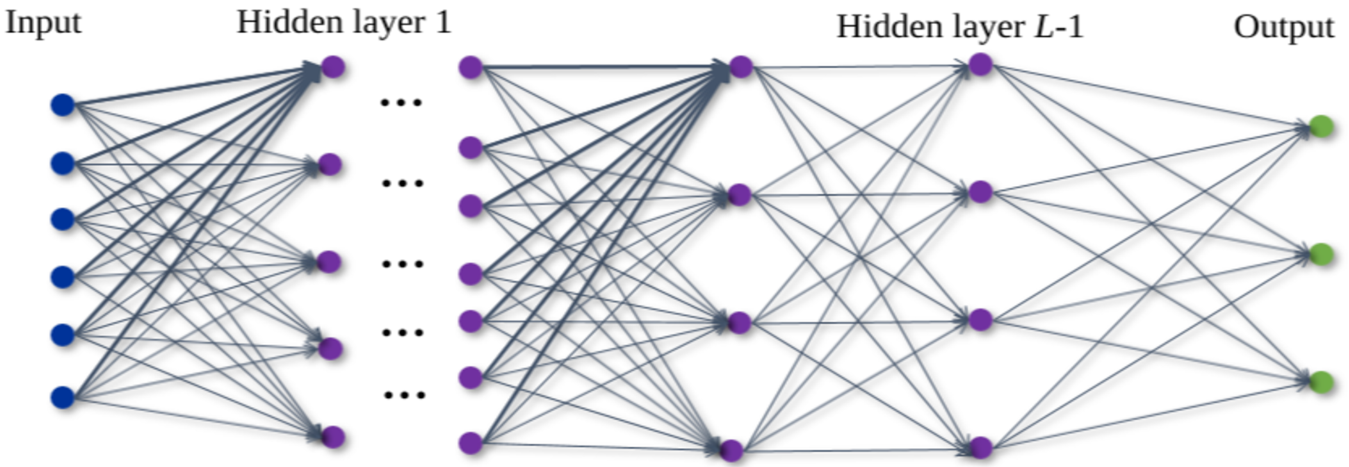
- Introduction
- Overcomplete Tensor Decomposition (Representation Learning)
- **Neural Collapse in Deep Network Training**

Understanding Deep Neural Networks

Z. Zhu, T. Ding, J. Zhou, X. Li, C. You, J. Sulam, and **Q. Qu**, [A Geometric Analysis of Neural Collapse with Unconstrained Features](), *arXiv Preprint arXiv:2105.02375*, May 2021.

- Analyzes the **global landscape** of the training loss based on the **unconstrained feature model**
- Explains the ubiquity of **Neural Collapse** of the learned representations of the network

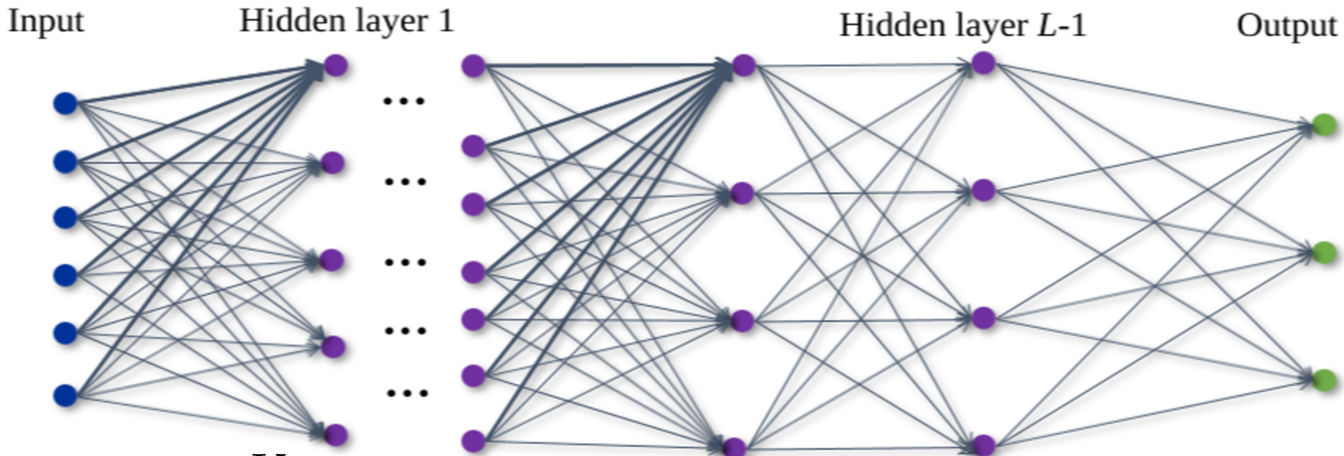
Understanding Deep Neural Networks



$$\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}) \;=\; \boldsymbol{W}_L \sigma\left(\boldsymbol{W}_{L-1} \cdots \sigma(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_{L-1}\right) + \boldsymbol{b}_L$$

$$\boldsymbol{\Theta} \;:=\; \{\underbrace{\boldsymbol{W}_\ell}_{\text{weights}}, \underbrace{\boldsymbol{b}_\ell}_{\text{bias}}\}_{\ell=1}^{L} \qquad \sigma(\cdot)\text{: nonlinear activations}$$

Understanding Deep Neural Networks

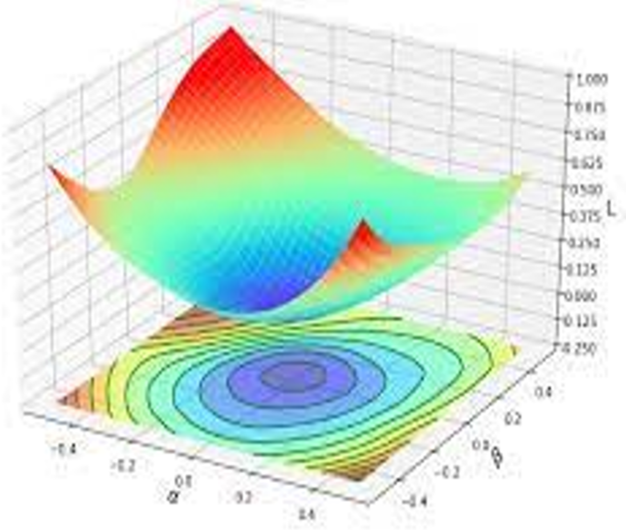


$$\min_{\boldsymbol{\Theta}} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathscr{L}_{\mathrm{CE}}\left(\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}_{k,i}), \boldsymbol{y}_k\right) + \|\boldsymbol{\Theta}\|_F^2$$

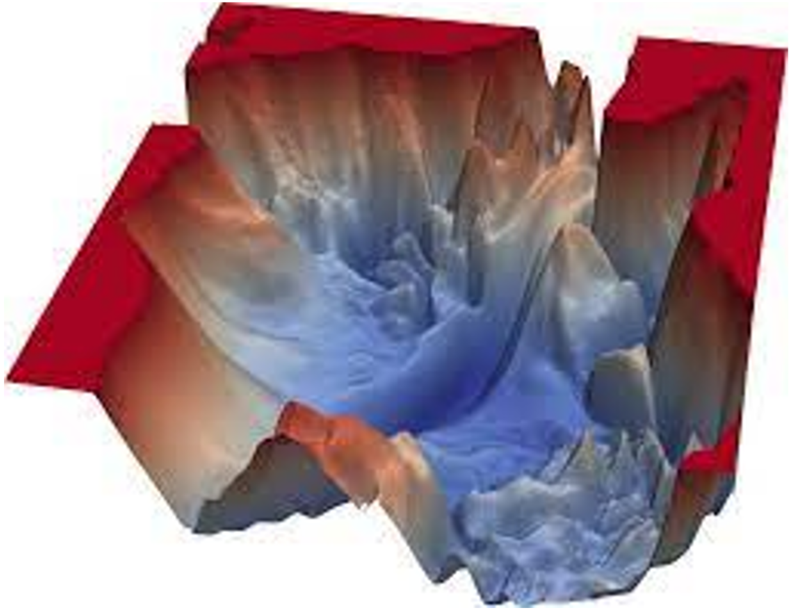
i-th input in the k-th class

One-hot vector for the k-th class

Fundamental Challenges: Optimization



Landscape in Classical Optimization
(abundant algorithms & theory)



Landscape of Modern Deep Neural Networks
Credited to [Li'17]

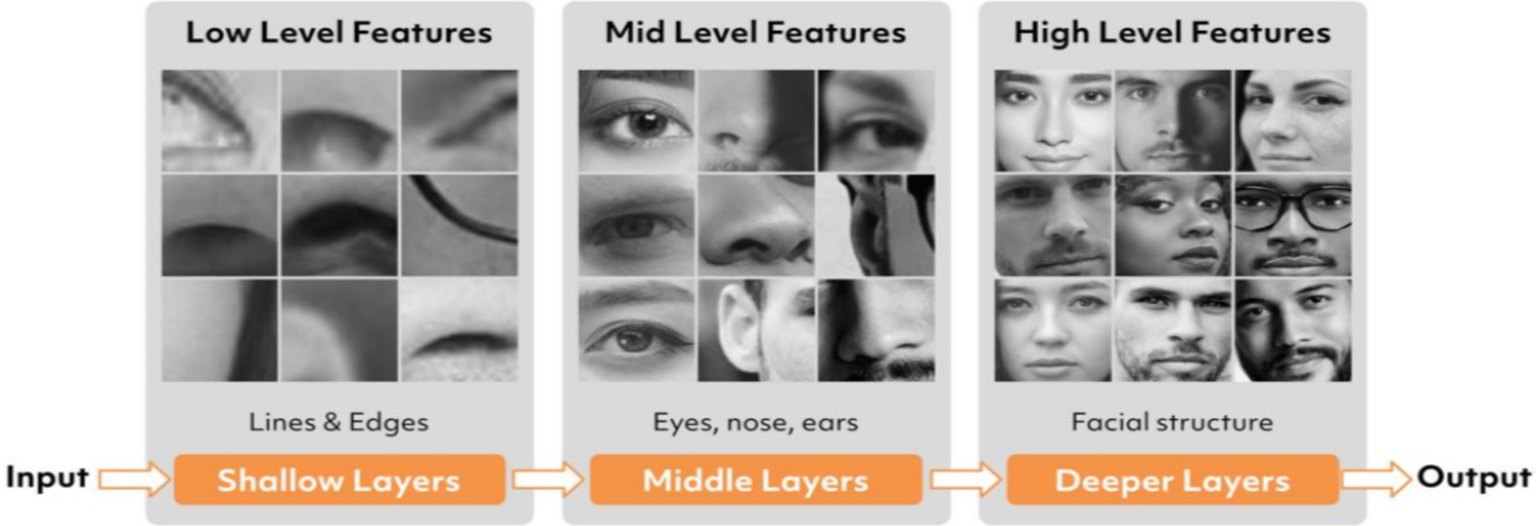
Optimization: Existing Results

Existing analysis are based on various **simplifications**:

- **Go Linear:** deep linear networks [Kawaguchi'16], deep matrix factorizations [Arora'19], etc.
- **Go Shallow:** Two-layer neural networks [Safran'18, Liang'18], etc.
- **Go Wide:** Neural tangent kernels [Jacot'18, Allen-Zhu'18, Du'19], mean-field analysis [Mei'19, Sirignano'19], etc.

Most of results ***hardly*** provide much insights for **practical** neural networks.

Features – What NNs (Conceptually) Designed to Learn



Wishful Design: NNs learn rich feature representations across different levels?

Neural Collapse in 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

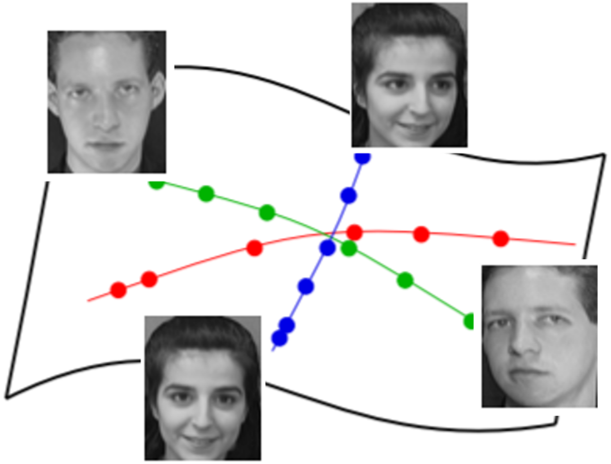
Contributed by David L. Donoho, August 18, 2020 (sent for review July 22, 2020; reviewed by Helmut Boelsckei and Stéphane Mallat)

Neural Collapse in Classification

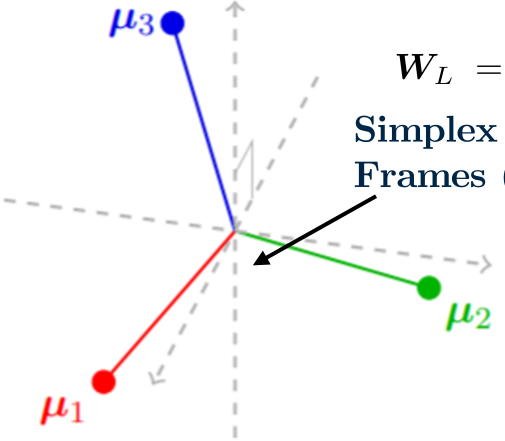
$$\psi_{\boldsymbol{\Theta}}(\boldsymbol{x}) = \boldsymbol{W}_L \underbrace{\sigma\left(\boldsymbol{W}_{L-1}\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_{L-1}\right)}_{\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) =: \boldsymbol{h}} + \boldsymbol{b}_L$$

Last-layer classifier

Last-layer feature



$\xrightarrow{\phi_{\boldsymbol{\theta}}(\cdot)}$



$$\boldsymbol{W}_L = \begin{bmatrix} \boldsymbol{\mu}_1 & \cdots & \boldsymbol{\mu}_K \end{bmatrix}^\top.$$

Simplex Equiangular Tight Frames (Simplex ETF)

Data in the Input Space

Neural Collapse in the Feature Space

Neural Collapse: Symmetry and Structures

Balanced training dataset with $n = n_1 = n_2 = \cdots = n_K$, and

$$\boldsymbol{W} := \boldsymbol{W}_L, \quad \boldsymbol{H} := \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}.$$

Neural Collapse (NC) means that

1) ***Within-Class Variability Collapse on H:*** features of each class collapse to class-mean with **zero** variability;

2) ***Convergence to Simplex ETF on H:*** the class means are **linearly separable**, and **maximally distant*****;***

3) ***Convergence to Self-Duality (W,H):*** the last-layer classifiers are **perfected matched** with the class-means of features.

4) ***Simple Decision Rule*** via Nearest Class-Center decision.

Simplification: Unconstrained Features

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

Last-layer classifier

Last-layer feature

Treat $\boldsymbol{H} = \begin{bmatrix}\boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n}\end{bmatrix}$ as a free optimization variable

Simplification: Unconstrained Features

$$\psi_{\mathbf{\Theta}}(\boldsymbol{x}) \;=\; \boldsymbol{W}_L\, \underbrace{\sigma\left(\boldsymbol{W}_{L-1} \cdots \sigma(\boldsymbol{W}_1 \boldsymbol{x} + \boldsymbol{b}_1) + \boldsymbol{b}_{L-1}\right)}_{\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) =: \boldsymbol{h}} + \boldsymbol{b}_L$$

Last-layer classifier

Last-layer feature

Treat $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n} \end{bmatrix}$ as a free optimization variable

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

Simplification: Unconstrained Features

$$\psi_{\mathbf{\Theta}}(\boldsymbol{x}) \;=\; \underbrace{\boldsymbol{W}_L}_{\text{Last-layer classifier}} \underbrace{\sigma\left(\boldsymbol{W}_{L-1}\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1)+\boldsymbol{b}_{L-1}\right)}_{\phi_{\boldsymbol{\theta}}(\boldsymbol{x}) =: \boldsymbol{h}} + \boldsymbol{b}_L$$

Last-layer classifier

Last-layer feature

Treat $\boldsymbol{H} = \begin{bmatrix}\boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{K,n}\end{bmatrix}$ as a free optimization variable

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

- **Validity:** Modern network are highly **overparameterized**, that can **approximate any point** in the feature space [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.

Main Theoretical Results

Theorem (Informal) *Consider the nonconvex loss with unconstrained feature model with $K < d$ and balanced data*

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

- ***(Global Optimality)*** *Any global solution* $(\mathbf{W}_\star, \mathbf{H}_\star)$ *satisfies the NC properties (1-4).*
- ***(Benign Global Landscape)*** *The function has* ***no spurious*** *local minimizer and is a* ***strict saddle function****, with negative curvature for non-global critical point.*

Main Theoretical Results

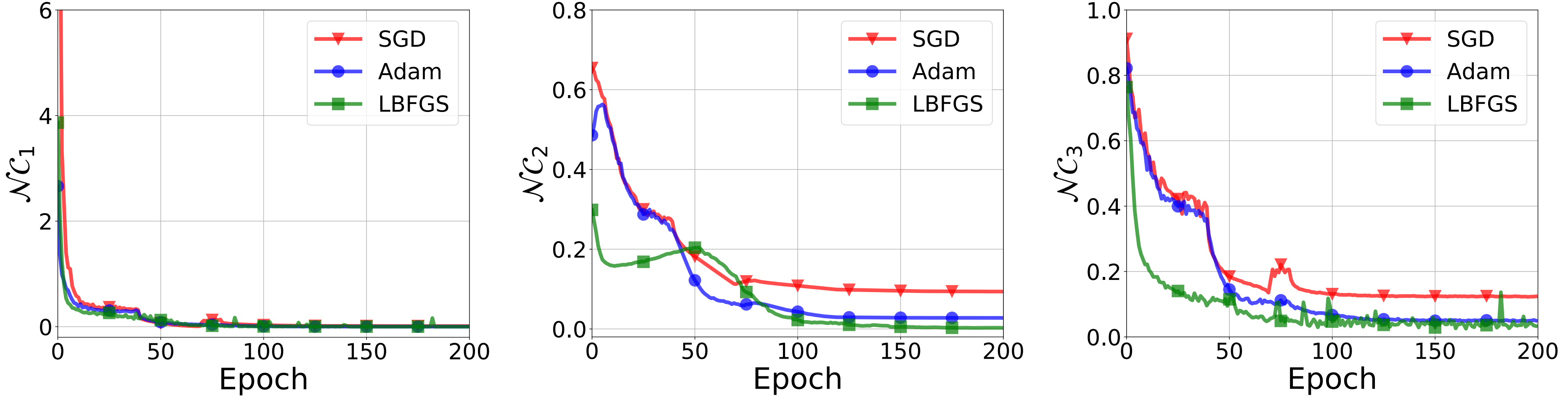
Theorem (Informal) *Consider the nonconvex loss with unconstrained feature model with* $K < d$ and balanced data

- ***(Global Optimality)*** *Any global solution* $(\mathbf{W}_\star, \mathbf{H}_\star)$ *satisfies the NC properties (1-4).*
- ***(Benign Global Landscape)*** *The function has* ***no spurious*** *local minimizer and is a* ***strict saddle function****, with negative curvature for nonglobal critical point.*

Message: deep networks always learn Neural Collapse features and classifiers, provably

Experiment: NC is Algorithm Independent

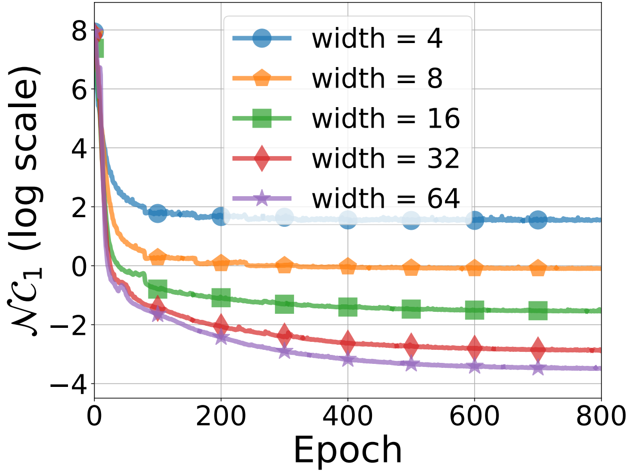
CIFAR-10 Dataset, ResNet18, with **different training algorithms**



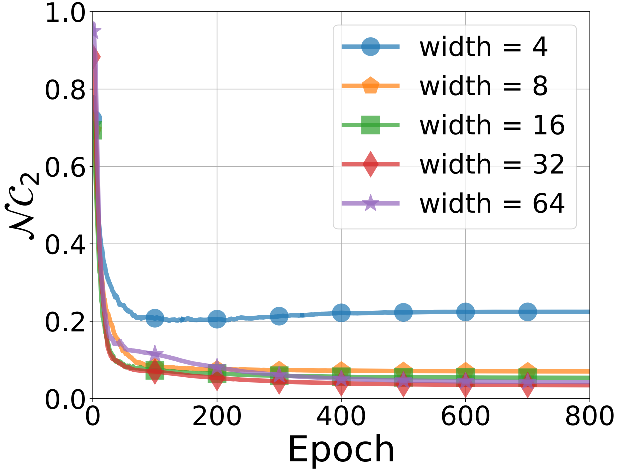
Measure of Within-Class Variability | Measure of Between-Class Separation | Measure of Self-Duality Collapse

Experiment: NC Occurs for Random Labels

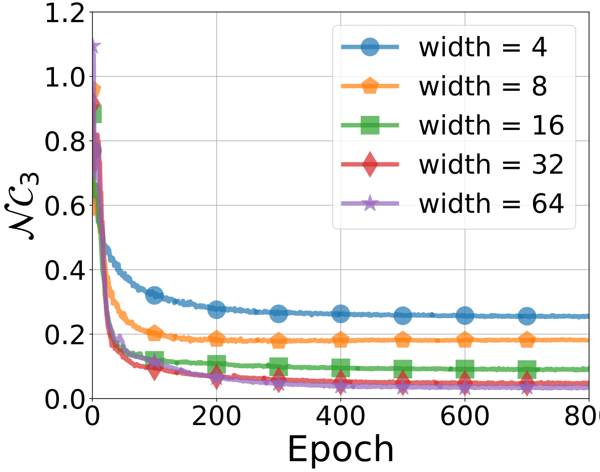
CIFAR-10 Dataset, ResNet18, **random labels with varying network width**



Measure of Within-Class Variability



Measure of Between-Class Separation



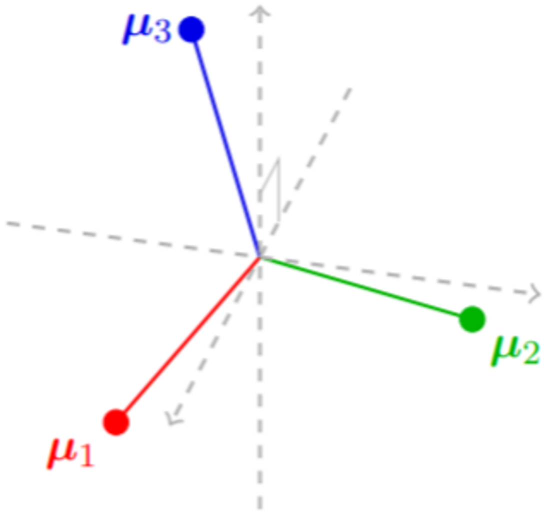
Measure of Self-Duality Collapse

Validity of Unconstrained Feature Model: Learned last-layer features and classifiers seems to be **independent of input!**

Implications for Practical Network Training

Observation: For NC features, when $K \leq d$ the best classifier is given by the Simplex ETF

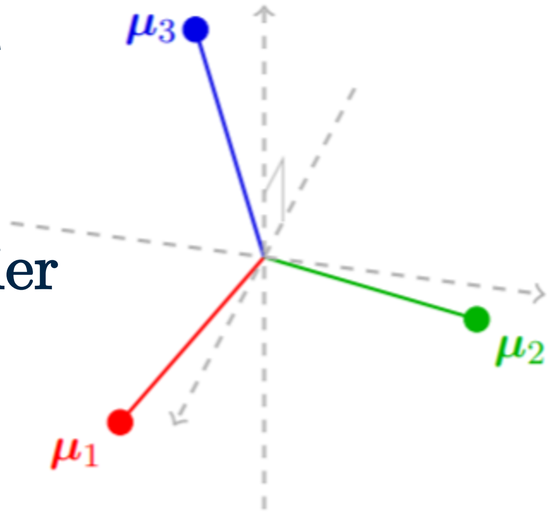
$$\boldsymbol{W}_\star = \begin{bmatrix} \boldsymbol{\mu}_1 & \cdots & \boldsymbol{\mu}_K \end{bmatrix}^\top.$$



Implications for Practical Network Training

Observation: For NC features, when $K \leq d$ the best classifier is given by the Simplex ETF

$$\boldsymbol{W}_{\star} = \begin{bmatrix} \boldsymbol{\mu}_1 & \cdots & \boldsymbol{\mu}_K \end{bmatrix}^{\top}.$$

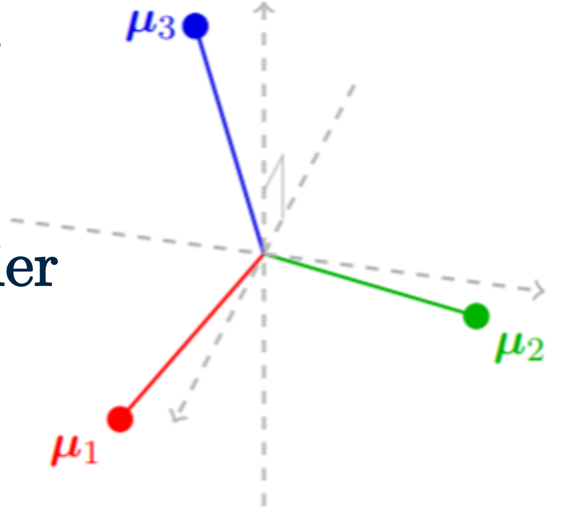


- **Implication 1: No need to learn the classifier**
  - ❑ Just fix them as a Simplex ETF
  - ❑ Save **8%, 12%,** and **53%** parameters for ResNet50, DenseNet169, and ShuffleNet!

Implications for Practical Network Training

Observation: For NC features, when $K \leq d$ the best classifier is given by the Simplex ETF

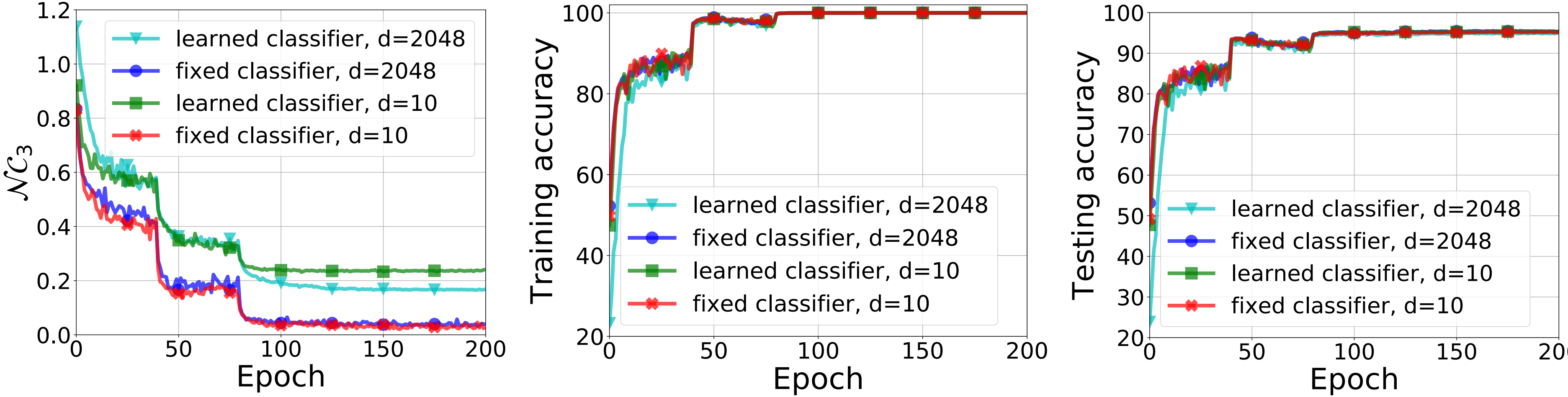
$$\boldsymbol{W}_\star = \begin{bmatrix} \boldsymbol{\mu}_1 & \cdots & \boldsymbol{\mu}_K \end{bmatrix}^\top.$$



- **Implication 1: No need to learn the classifier**
  - ❑ Just fix them 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 $d = \#\text{class } K$ (e.g., d=10 for CIFAR10)
  - ❑ Further saves **21%** and **4.5%** parameters for ResNet18 and ResNet50!

Experiment: Fixed Classifier with $d = K$

ResNet50, CIFAR10, Comparison of **Learned vs. Fixed Classifiers of W**



Measure of Between-Class Separation | Training Accuracy | Testing Accuracy

Training with fixed last-layer classifiers achieves **on-par performance** with learned classifiers.

Summary and Discussion

Z. Zhu, T. Ding, J. Zhou, X. Li, C. You, J. Sulam, and **Q. Qu**, A Geometric Analysis of Neural Collapse with Unconstrained Features, *arXiv Preprint arXiv:2105.02375*, May 2021.

- Through landscape analysis under unconstrained feature model, we provide a **complete characterization of learned representation** of deep networks.

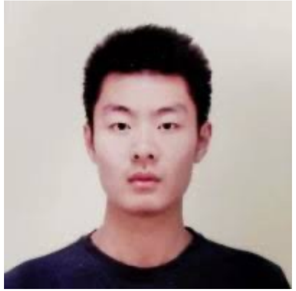
- The understandings of learned representations could shed lights on generalization, robustness, and transferability.

Outline of this Talk

- Introduction
- Overcomplete Tensor Decomposition (Representation Learning)
- Neural Collapse in Deep Network Training

Acknowledgement

Tianyu Ding
(Johns Hopkins)



Xiao Li
(U. Michigan)

Xiao Li
(CUHK-Shen Zhen)

Jeremias Sulam
(Johns Hopkins)

Chong You
(Google Research)

Yuexiang Zhai
(UC Berkeley)

Zhihui Zhu
(University of Denver)

Thank You!