Data Increasingly Massive & High-Dimensional...

Data representation is **critical** for modern machine learning methods.
Unsupervised Learning

- Learning sparsely-used dictionaries:
  Given $Y \in \mathbb{R}^{n \times p}$, jointly find overcomplete dictionary $A \in \mathbb{R}^{n \times m}$ and sparse $X \in \mathbb{R}^{m \times p}$. 
Unsupervised Learning

- Learning sparsely-used dictionaries:

$$\min_{A \in \mathcal{M}, X} f(Y, A \cdot X) + \lambda \cdot g(X)$$

- data fidelity

- regularizer
Unsupervised Learning

\[ y_i \approx \sum_k a_k \ast x_{ki} \]

• *Learning convolutional dictionaries:* Given \( \{y_i\}_i \), jointly learn convolutional dictionaries \( \{a_i\}_i \) and sparse coefficients \( \{x_{ki}\}_{i,k} \).

➤ Translation invariant, can be viewed as one layer of ConvNets
One-hot vectors in Data in the input space

Neural network

Denoising

Image Restoration

Super Resolution

Image Half-toning

- Image courtesy of Julien Mairal et al.
Supervised (Deep) Learning

Deep learning has attained superior performances for many tasks in practice:

- Computer vision
- Natural language processing
- Gameplay
- Protein modeling

“Cat”

(Alex et al., 2012)
Training Deep Neural Networks

\[
\psi_\Theta(x) = W_L \sigma (W_{L-1} \cdots \sigma (W_1 x + b_1) + b_{L-1}) + b_L
\]

\[
\Theta := \{ W_\ell, b_\ell \}_{\ell=1}^L \quad \sigma(\cdot): \text{nonlinear activations}
\]
Training Deep Neural Networks

\[
\min_{\Theta} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}_{CE} (\psi_{\Theta}(x_{k,i}), y_k) + \lambda \| \Theta \|_F^2
\]

- \(i\)-th input in the \(k\)-th class
- One-hot vector for the \(k\)-th class
Nonconvex Problems in Representation Learning

$$\min_x f(x), \text{ s.t. } x \in \mathbb{R}^n$$

Nonconvex landscape  Convex landscape
General Nonconvex Problems

Noncritical Point ($\nabla \varphi \neq 0$)

Minimizer
$\nabla^2 \varphi > 0$

Saddle
$\lambda_{\text{min}} \nabla^2 \varphi < 0$
$\lambda_{\text{max}} \nabla^2 \varphi > 0$

Maximizer
$\nabla^2 \varphi < 0$

Critical Points ($\nabla \varphi = 0$)
General Nonconvex Problems

\[
\min_x f(x), \quad \text{s.t. } x \in \mathbb{R}^n
\]

“bad” local minimizers  “flat” saddle points

local minima  “flat” saddle

global minima
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]
• 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]
• ...
Nonconvex Problems with Benign Landscape


Outline of this Talk

• Learning Shallow Representations: (Convolutional) Dictionary Learning

• Learning Deep Representations: Deep Neural Collapse

• Conclusion & Discussion
Landscape Analysis of Dictionary Learning

1. 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 first global nonconvex landscape analysis for convolutional/overcomplete dictionary learning problems.

• Efficient nonconvex optimization methods to global solutions with applications in imaging.
Convolutional Dictionary Learning (DL)

Given multiple measurements $\{y_i\}_i$ of circulant convolution

$$y_i = \sum_{k=1}^{K} a_k \ast x_{ki}, \quad (1 \leq i \leq p)$$

can we learn all $\{a_k\}_k$ and $\{x_{ki}\}_{k,i}$ simultaneously?
Convolutional Dictionary Learning (DL)

two-photon calcium image $Y$

activation map $X_k (k = 1, 2)$

kernel $A_k (k = 1, 2)$

reconstruction

$Y_k = A_k \otimes X_k (k = 1, 2)$
Convolutional DL vs. Overcomplete DL

For each \( y_i = a \ast x_i \), we can equivalently rewrite in the matrix form as:

\[
C_{y_i} = C_a \cdot C_{x_i}, \quad 1 \leq i \leq p
\]

where a circulant matrix

\[
C_a = \begin{bmatrix}
    s_1[a] & s_2[a] & \cdots & s_n[a]
\end{bmatrix} \in \mathbb{R}^{n \times n}
\]
Convolutional DL vs. Overcomplete DL

For each sample

\[ y_i = \sum_{k=1}^{K} a_k \ast x_{ki}, \]

equivalently,

\[ C_{y_i} = \left[ \begin{array}{ccc}
C_{a_1} & \cdots & C_{a_K}
\end{array} \right] \cdot \left[ \begin{array}{c}
C_{x_{1i}} \\
\vdots \\
C_{x_{Ki}}
\end{array} \right], \]

overcomplete \( A_0 \)

sparse \( X_i \)
Given $Y = A_0 \cdot X_0$, learn overcomplete $A_0$ and sparse $X_0$?
Relationship to Dictionary Learning

We can find one column of $A_0$ via

$$\min_q f_{DL}(q) = -\|Y^\top q\|_4^4, \quad \text{s.t.} \quad \|q\|_2 = 1.$$ 

The underlying reasoning is that, in expectation

$$\mathbb{E}_X \left[ \|Y^\top q\|_4^4 \right] = \mathbb{E}_X \left[ \|X^\top A_0^\top q\|_4^4 \right] = c_1 \|A_0^\top q\|_4^4 + c_2$$

for $X$ following some sparse zero-mean distributions (e.g., Bernoulli-Gaussian)
Given $A_0 = [a_1 \cdots a_m] \in \mathbb{R}^{n \times m}$

$$\min_{q} - \|A_0^\top q\|_4^4, \quad \text{s.t.} \quad \|q\|_2 = 1.$$  

- When $m \leq n$, and $\{a_i\}_{i=1}^m$ 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 $\{\pm a_i\}_{i=1}^m$.
- The analysis of orthogonal case cannot be generalized to overcomplete settings.
Global Landscape of Overcomplete DL

\[ \min_{q} f_{DL}(q) = -\|Y^\top q\|_4^4, \quad \text{s.t.} \quad \|q\|_2 = 1. \]

**Theorem (Informal)** Suppose that (i) \( K = m/n \) is a constant, (ii) \( A_0 \) is near orthogonal, and (iii) \( p \geq \Omega(\text{poly}(n)) \). Then with high probability every critical point of \( f(q) \) is either

- a **strict saddle point** exhibits negative curvature;
- or close to a **target solution**: one column \( a_i \) of \( A \).
Assumptions on $A$ (Near Orthogonal)

- Row orthogonal: unit norm tight frame (UNTF)
  \[
  \sqrt{\frac{n}{m}} A_0 A_0^\top = I, \quad \|a_i\|_2 = 1.
  \]

- Incoherence of the columns (near orthogonal)
  \[
  \max_{i \neq j} |\langle a_i, a_j \rangle| \leq \mu.
  \]
Overcomplete Dictionary Learning

Given \( Y = A_0 \cdot X_0 \), learn overcomplete \( A_0 \) and sparse \( X_0 \)?

\[
A_0 \in \mathbb{R}^{n \times m}
\]

Global

Local

\( n = m \)

\( m > n \)

Our Result

Initialization Required

[Sun, Qu, Wright’16]

[Li et al.’18]

[Qu, Sun, Wright’16]

[Zhai et al.’19]

[Arora et al.’14&15]

[Agarwal et al.’16]

[Chatterji et al.’17]

[Awasthi et al.’18]
Find one shift of the kernel $a_i$ via

$$
\min_q f_{CDL}(q) = - \|q^\top Y\|_4^4, \quad \text{s.t.} \quad q \in S^{n-1}
$$
Convolutional Dictionary Learning

Find one shift of the kernel \( a_i \) via

\[
\min_q f_{CDL}(q) = - \left\| q^\top P Y \right\|_4^4, \quad \text{s.t.} \quad q \in S^{n-1}
\]

- Preconditioning matrix:

\[
P = \left( (\theta n K)^{-1} Y Y^\top \right)^{-1/2} \approx (A_0 A_0^\top)^{-1/2}
\]

- Effective dictionary is tight frame (but not necessarily unit norm)

\[
P Y \approx \left( A_0 A_0^\top \right)^{-1/2} A_0 X_0 = A X_0
\]
Convolutional Dictionary Learning

Find one shift of the kernel $a_i$ via

$$\min_q - \| q^\top A X_0 \|^4_4, \quad \text{s.t.} \quad q \in S^{n-1}$$

- Preconditioning matrix:

$$P = \left( (\theta n K)^{-1} Y Y^\top \right)^{-1/2} \approx \left( A_0 A_0^\top \right)^{-1/2}$$

- Effective dictionary is tight frame (but not necessarily unit norm)

$$P Y \approx \left( A_0 A_0^\top \right)^{-1/2} A_0 X_0 = A X_0$$
Local Landscape of Convolutional DL

Theorem (Informal) Suppose that (i) $K = m/n$ is a constant, (ii) $A$ is near orthogonal, and (iii) $p \geq \Omega(\text{poly}(n))$. Locally, every critical point of $f_{\text{CDL}}(q)$ is either

- a strict saddle point exhibits negative curvature;
- or close to a target solution: a precond. shift of $a_i$.

- We show the result over a local level-set

$$\mathcal{R}_{\text{CDL}} := \left\{ q \in S^{n-1} \mid \mathbb{E}_x \left[ f_{\text{CDL}}(q) \right] \leq -c \| A^\top q \|_3^2 \right\},$$

- We can cook up smart yet simple initializations, with all future iterations stay in the region.
Learning Random Filters

\[ \min_q f_{CDL}(q) = - \| q^\top P Y \|^4_4, \quad \text{s.t.} \quad q \in \mathbb{S}^{n-1} \]

Learning 3 random filters by the proposed approach.
From Theory to Practical Methods

• Recovering one filter:

$$\min_{q} f_{CDL}(q) = -\left\| q^\top PY \right\|_4^4, \quad \text{s.t.} \quad q \in S^{n-1}$$

• Finding all filters via Bilinear Lasso formulation:

$$\min \frac{1}{2} \left\| y - \sum_{k=1}^{N} \mathbf{a}_k \otimes \mathbf{x}_k \right\|_2^2 + \lambda \sum_{k=1}^{N} \left\| \mathbf{x}_k \right\|_1, \quad \text{s.t.} \quad \left\| \mathbf{a}_k \right\| = 1.$$
From Theory to Practical Methods

• Finding all filters via Bilinear Lasso formulation:

\[
\min_{a_k, x_k} \frac{1}{2} \left\| y - \sum_{k=1}^{N} a_k \otimes x_k \right\|^2 + \lambda \sum_{k=1}^{N} \| x_k \|_1, \quad \text{s.t.} \quad \| a_k \| = 1.
\]

• Optimization. Alternating descent method

\[
x \leftarrow \text{prox} \left( x - \tau \cdot \nabla_x \varphi_{BL}(a, x) \right)
\]

\[
a \leftarrow \mathcal{P}_{S^{n-1}} \left( a - t \cdot \text{grad}_a \varphi_{BL}(a, x) \right),
\]

with few extra caveats.
From Theory to Practical Methods

Comparison of convergence (time).

- FFT operations
  - all-pass filter
  - low-pass filter
Spike Sorting

- [Link to Spike Sorting](https://vis.caltech.edu/~rodri/Waveclus/Waveclus_home.htm)
Defects Detection in Scan Tunneling Microscopy

STM image $Y$

kernel $A_k \ (k = 1, 2)$

activation map $X_k \ (k = 1, 2)$

reconstruction

$Y_k = A_k \otimes X_k \ (k = 1, 2)$
Outline of this Talk

- Learning Shallow Representations: (Convolutional) Dictionary Learning
- Learning Deep Representations: Deep Neural Collapse
- Conclusion & Discussion
Understanding Deep Neural Networks


- Analyzes the \textbf{global landscape} of the training loss based on the \textbf{unconstrained feature model}

- Explains the ubiquity of \textbf{Neural Collapse} of the learned representations of the network
Understanding Deep Neural Networks

\[ \psi_\Theta(x) = W_L \sigma(W_{L-1} \cdots \sigma(W_1 x + b_1) + b_{L-1}) + b_L \]

\[ \Theta := \{ W_\ell, b_\ell \}_{\ell=1}^L \]

\( \sigma(\cdot) \): nonlinear activations

weights
bias
Terminology for Classification

- Labels: $k = 1, \ldots, K$
  - $K = 10$ classes (MNIST, CIFAR10, etc.)
  - $K = 1000$ classes (ImageNet)

![Diagram showing a neural network with input $x$, output $y$, and one-hot vectors in $\mathbb{R}^K$]
Understanding Deep Neural Networks

\[
\min_{\Theta} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}_{CE}(\psi_{\Theta}(x_{k,i}), y_k) + \lambda \| \Theta \|_F^2
\]

- \(i\)-th input in the \(k\)-th class
- One-hot vector for the \(k\)-th class
Mysteries in Deep Learning

- **Architecture design** (before training):
  - Feature dimensionality
  - Network width and depth
  - Activation functions

- **Optimization** (during training):
  - Choices of loss functions
  - Optimization algorithms, normalization

- **Properties of learned network** (after training):
  - Generalization
  - Robustness

---

Goodfellow et al ICLR’15

[Graph showing loss functions over training steps]

(Zhang et al ICLR’17)
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

$$\psi_{\Theta}(x) = W_L \sigma(W_{L-1} \cdots \sigma(W_1 x + b_1) + b_{L-1}) + b_L$$

Last-layer classifier

$$\phi_{\Theta}(x) = : h$$

Last-layer feature

Data in the Input Space

$W_L = [\mu_1 \cdots \mu_K]^T$.

Simplex Equiangular Tight Frames (Simplex ETF)

Neural Collapse in the Feature Space
Prevalence of neural collapse during the terminal phase of deep learning training

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

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 Boelscke and Stéphane Mallat)
Neural Collapse in Classification

• Reveals common outcome of network training **across a variety of architectures** (ResNet, VGG) and **dataset** (CIFAR, ImageNet)

• Precise **mathematical structures** within the features and classifier

Image credited to Han et al. “Neural Collapse Under MSE Loss: Proximity to and Dynamics on the Central Path”
Neural Collapse: Symmetry and Structures

Balanced training dataset with \( n = n_1 = n_2 = \cdots = n_K \), and
\[
W := W_L, \quad H := [h_{1,1} \cdots h_{K,n}].
\]

Neural Collapse (NC) means that

1) **Within-Class Variability Collapse on** \( H \): features of each class collapse to class-mean with zero variability;
   \[
   h_{k,i} \rightarrow \bar{h}_k, \quad \forall k \in [K], i \in [n].
   \]

2) **Convergence to Simplex ETF on** \( H \): the class means are linearly separable, and maximally distant;
   \[
   M^\top M = \frac{K}{K-1} \left( I_K - \frac{1}{K} 1_K 1_K^\top \right), \quad M = \alpha U \bar{H}
   \]
Neural Collapse: Symmetry and Structures

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

\[
W := W_L, \quad H := [h_{1,1} \cdots h_{K,n}].
\]

Neural Collapse (NC) means that

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

\[
w^k = \beta \overline{h}_k, \quad \forall \ k \in [K].
\]

4) **Simple Decision Rule** via Nearest Class-Center decision.
Simplification: Unconstrained Features

\[ \psi_\theta(x) = W_L \sigma(W_{L-1} \cdots \sigma(W_1 x + b_1) + b_{L-1}) + b_L \]

Last-layer classifier \( \phi_\theta(x) = h \) Last-layer feature

Treat \( H = [h_{1,1} \cdots h_{K,n}] \) as a free optimization variable
Simplification: Unconstrained Features

\[ \psi_\Theta(x) = W_L \sigma(W_{L-1} \cdots \sigma(W_1 x + b_1) + b_{L-1}) + b_L \]

Last-layer classifier \( \phi_\theta(x) = h \)  Last-layer feature

Treat \( H = [h_{1,1} \cdots h_{K,n}] \) as a free optimization variable

\[
\min_{W,H,b} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\text{CE}}(Wh_{k,i} + b, y_k) + \frac{\lambda_W}{2} \|W\|_F^2 + \frac{\lambda_H}{2} \|H\|_F^2 + \frac{\lambda_b}{2} \|b\|_2^2
\]
Simplification: Unconstrained Features

\[ \psi_\Theta(x) = W_L \sigma(W_{L-1} \cdots \sigma(W_1x + b_1) + b_{L-1}) + b_L \]

Last-layer classifier \( \phi_\theta(x) = h \) \hspace{1cm} \text{Last-layer feature}

Treat \( H = [h_{1,1} \cdots h_{K,n}] \) as a free optimization variable

\[
\min_{W,H,b} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{\text{CE}}(Wh_{k,i} + b, y_k) + \frac{\lambda_W}{2} \|W\|_F^2 + \frac{\lambda_H}{2} \|H\|_F^2 + \frac{\lambda_b}{2} \|b\|_2^2
\]

• **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.
Prior Work on Unconstrained Features

• [Lu et al’20] studies the following one-example-per class model
  \[ \min_H \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{CE}(h_k, y_k), \text{ s.t. } \|h_k\|_2 = 1 \]

• [E et al’20] considers
  \[ \min_{W, H} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{CE}(Wh_k, y_k), \text{ s.t. } \|W\|_2 \leq 1, \|h_k, i\|_2 \leq 1 \]

• [Fang et al’21] studies
  \[ \min_{W, H} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{CE}(Wh_k, y_k), \text{ s.t. } \|W\|_F^2 \leq CW, \|H\|_F^2 \leq CH \]

• These work show that any **global** solution has NC, but
  - What about local minima?
  - The constrain formulation are **not aligned with practice**

• [Mixon et al’21, Han et al’21] studies NC under the MSE loss

J. Lu and S. Steinerberger, Neural collapse with cross-entropy loss, 2020
W. E and S. Wojtowytsch, On the emergence of tetrahedral symmetry in the final and penultimate layers of neural network classifiers, 2020
D. Mixon, H. Parshall, J. Pi. Neural collapse with unconstrained features, 2020
X. Han, V. Papyan, D. Donoho, Neural Collapse Under MSE Loss: Proximity to and Dynamics on the Central Path, 2021
Our Main Theoretical Results

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

$$\min_{W,H,b} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{CE}(Wh_{k,i} + b, y_k) + \frac{\lambda_W}{2} \|W\|_F^2 + \frac{\lambda_H}{2} \|H\|_F^2 + \frac{\lambda_b}{2} \|b\|_2^2$$

• **(Global Optimality)** Any global solution $(W_*, H_*)$ 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.
Our Main Theoretical Results

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

- **(Global Optimality)** Any global solution $(W_*, H_*)$ 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
Interpretations of Our Results

- **A Feature Learning 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.**
Interpretations of Our Results

\[
\min_{W,H,b} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{CE}(Wh_{k,i} + b, y_k) + \frac{\lambda W}{2} \|W\|_F^2 + \frac{\lambda H}{2} \|H\|_F^2 + \frac{\lambda b}{2} \|b\|_2^2
\]

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]

- **Difference in tasks:** classification training vs recovery
- **Difference in global solutions.**
- **Difference in loss functions, statistical properties:** cross-entropy vs least-squares; randomness or statistical properties of the sensing matrices
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
Generalization is Algorithm dependent

MINST

CIFAR-10
Experiment: NC Occurs for Random Labels

CIFAR-10 Dataset, MLP, random labels with varying network width

Validity of Unconstrained Feature Model: Learned last-layer features and classifiers seems to be independent of input!
**Experiment: NC with Different Weight Decays**

CIFAR-10 Dataset, ResNet18, *different weight decay*

- **Test Accuracy**: 99.57% vs. 99.60% (*MINST*); 77.92% vs. 78.42% (*CIFAR-10*)
- Weight decay on the parameters (implicitly) regularizes the features

Measure of Within-Class Variability  
Measure of Between-Class Separation  
Measure of Self-Duality Collapse
Implications for Practical Network Training

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

\[ W_\star = \begin{bmatrix} \mu_1 & \cdots & \mu_K \end{bmatrix}^\top. \]
Implications for Practical Network Training

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

\[
W_* = [\mu_1 \ldots \mu_K]^T.
\]

• 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

$$W_\star = [\mu_1 \cdots \mu_K]^\top.$$  

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

• Learning Shallow Representations: (Convolutional) Dictionary Learning

• Learning Deep Representations: Deep Neural Collapse

• Conclusion & Discussion
Summary and Discussion

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


Future Directions: Beyond Last-layer Features

• **Study Deeper Networks**
  • Fix the last layer classifier $W$ as the Simplex ETF, and conduct NTK analysis for the learning dynamics of features $H$?
  • Recursively study the features of each layer from output?

Vardan Panyan. Traces of class/cross-class structure pervade deep learning spectra, JMLR’19.
Future Directions: Is NC a Blessing or Curse?

- Study generalization through the representation?

- Study tradeoff between accuracy and robustness via NC?

Adaptive to the Intrinsic Data Structures

- Can we learn diverse features that are adaptive to the intrinsic data structures?
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!
Relationship to Dictionary Learning

\[
\log_{10}(m) \quad \log_{10}(n)
\]

practice \( m < n^2 \)
vs. theory \( m < Cn \)

recover full \( A_0 \) via repeated independent trials
Experiment: NC Occurs for Random Labels

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

Measure of Within-Class Variability
Measure of Self-Duality Collapse
Training Error

Validity of Unconstrained Feature Model: Learned last-layer features and classifiers seems to be independent of input!
Comparisons to MCR$^2$

- [Yu et al, NeurIPS’20] learns not only discriminative but also diverse representations via maximizing the difference between the coding rate of all features and the average rate of features in the classes:

\[
\Delta R(H, \epsilon) = \frac{1}{2} \log \det(I + \frac{d}{n\epsilon^2}) - \sum_{k=1}^{K} \frac{n_k}{2n} \log \det(I + \frac{d}{n_k\epsilon^2} H_k H_k^\top)
\]

- $R$: expand all features $H$ as large as possible.

- $R^c$: compress all each class $H_k$ as small as possible.

- For balanced data, learned features $H_k$ span an entire $d/K$ subspace, and the subspaces are orthogonal to each other.

Comparisons to MCR$^2$

- [Yu et al, NeurIPS’20] learns not only **discriminative** but also **diverse** representations via maximizing the difference between the coding rate of all features and the average rate of features in the classes:

$$\text{MCR}^2$$

$$\text{Cross-entropy}$$


Comparisons to MCR²

- [Yu et al, NeurIPS’20] learns not only discriminative but also diverse representations via maximizing the difference between the coding rate of all features and the average rate of features in the classes: