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?

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

Qing Qu (EECS, University of Michigan)

Low-dimensional Representations

June 10th, 2023
What Representations are DNNs Designed to Learn?

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

![Diagram showing layers of features](image)

- Low-level features (VGG-16 Conv1_1)
- Mid-level features (VGG-16 Conv3_2)
- High-level features (VGG-16 Conv5_3)
- Linearly separable classifier
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

1. Low-Dimensional Representation: Neural Collapse (NC)
2. Understanding NC from Optimization
3. Prevalence of NC under Different Training Scenarios
4. Conclusion
Multi-Class Image Classification Problem

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

\[1\text{If not, we can use data augmentation to make them balanced.}\]
Multi-Class Image Classification Problem

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

1\(^{1}\)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 \( \{(x^{(i)}, y^{(i)}) ; i = 1, \ldots , n\} \).

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

- 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:

\[ f_{\Theta}(x) = W_L \underbrace{\sigma(W_{L-1} \cdots \sigma(W_1 x + b_1) + b_{L-1}) + b_L}_\text{linear classifier } W + b_L \]

feature \( \phi_{\theta}(x) = h \)
Deep Neural Network Classifiers

• **A vanilla deep network:**

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

linear classifier \( W \)

feature \( \phi_\theta(x) =: h \)

• **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:

\[
\min_{\theta, W, b} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \left( \mathcal{L}_{\text{CE}}(W \phi_{\theta}(x_{k,i}) + b, y_k) + \lambda \| (\theta, W, b) \|^2_F \right)
\]

\[
\text{cross-entropy (CE) loss}
\]

\[
\text{weight decay}
\]
Deep Neural Network Classifiers

\[ \text{Input: } \mathbf{x} \]

Feature/representation
(Our focus)

Output
\( \{W, b\} \)

\[ \text{Linear classifier: } W \mathbf{h} + b \]

\[ \text{Output: } f(\mathbf{x}; \theta) = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \]

\[ \text{Softmax function: } \begin{bmatrix} 0.6 \\ 0.3 \\ 0.1 \end{bmatrix} \]

Prediction
(probability)

\[ \text{Target: } \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \]

\[ \text{CE(Cat): } = -q(\text{Cat}) \cdot \log p(\text{Cat}) \]
\[ = -1 \cdot \log 0.6 \]
\[ = 0.51... \]
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 Boelscke 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

Neural Collapse: Symmetry and Structures

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

  \[ h_{k,i} \rightarrow \bar{h}_k, \]

  \( k \)-th class, \( i \)-th sample:
Neural Collapse: Symmetry and Structures

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

\[ h_{k,i} \rightarrow \overline{h}_k, \]

\( k \)-th class, \( i \)-th sample
Neural Collapse: Symmetry and Structures

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

\[
\frac{\langle \overline{h}_k, \overline{h}_{k'} \rangle}{\| \overline{h}_k \| \| \overline{h}_{k'} \|} \rightarrow \begin{cases} 1, & k = k' \\ -\frac{1}{K-1}, & k \neq k' \end{cases}
\]
Neural Collapse: Symmetry and Structures

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

\[
\bar{H}^\top \bar{H} \sim I_K - \frac{1}{K} \mathbf{1}_K \mathbf{1}_K^\top,
\]
\[
\bar{H} = \begin{bmatrix} \bar{h}_1 & \cdots & \bar{h}_K \end{bmatrix}
\]
Neural Collapse: Symmetry and Structures

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

\[ \frac{\mathbf{w}^k}{\| \mathbf{w}^k \|} \rightarrow \frac{\overline{\mathbf{h}}_k}{\| \overline{\mathbf{h}}_k \|}, \]

where \( \mathbf{w}^k \) represents the \( k \)-th row of \( \mathbf{W} \).
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

1. Low-Dimensional Representation: Neural Collapse (NC)

2. Understanding NC from Optimization

3. Prevalence of NC under Different Training Scenarios

4. Conclusion
Dealing with a Highly Nonconvex Problem

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

$$\min_{\theta', W, b} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{CE}(W \phi_{\theta'}(x_{k,i}) + b, y_k) + \lambda \| (\theta', W, b) \|_F^2,$$

due to the fact that the network

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

- Nonlinear interaction across layers.
- Nonlinear activation functions.
**Simplification: Unconstrained Feature Model**

**Assumption.** We treat \( H = [h_{1,1} \, \cdots \, h_{K,n}] \) as a **free** optimization variable, ignoring the constraint \( h_{\phi}(x) \).
The Trend of Large Models...

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

\[
\begin{align*}
\text{~23 million} & \quad \gg \quad \text{~1 million} \\
(\# \text{ Parameters in ResNet-50}) & \quad (\# \text{ Samples in ImageNet})
\end{align*}
\]

*In principle, deep network can fit any training labels! (i.e., not only clean, but also corrupted labels)*
Assumption. We treat $H = [h_{1,1}, \cdots, h_{K,n}]$ as a free optimization variable, ignoring the constraint $h_{\phi\theta}(x)$. 
Simplification: Unconstrained Feature Model

Assumption. We treat $H = [h_{1,1}, \cdots, h_{K,n}]$ 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];
Understanding NC from Optimization

Simplification: Unconstrained Feature Model

**Assumption.** We treat $H = [h_{1,1} \cdots h_{K,n}]$ 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];
- **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

- 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)
Experiments: NC Occurs on Random Labels/Inputs

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

- **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

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

**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. \((W,H)\). Then

- **Global optimality**: Any global solution \((\{H^*, W^*, b^*\})\) obeys Neural Collapse, with \(b^* = 0\) and
  
  \[
  h^*_{k,i} = \overline{h}^*_k,
  \]
  \[
  \langle \overline{h}^*_k, \overline{h}^*_k' \rangle = \begin{cases} 1, & k = k' \\ -\frac{1}{K-1}, & k \neq k' \end{cases}
  \]
  \[
  \frac{w^{k*}}{\|w^{k*}\|} = \frac{\overline{h}^*_k}{\|\overline{h}^*_k\|}
  \]

Qing Qu (EECS, University of Michigan)  Low-dimensional Representations  June 10th, 2023  19 / 50
Geometric Analysis of Global Landscape

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

$$\min_{\{h_k\}} \frac{1}{K} \sum_{k=1}^{K} \mathcal{L}_{CE}(h_k, y_k), \text{ s.t. } \|h_k\|_2 = 1$$

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

$$\min_{\{h_{k,i}\}, W} \frac{1}{Kn} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{CE}(Wh_{k,i}, y_k), \text{ s.t. } \|W\|_F \leq 1, \|h_{k,i}\|_2 \leq 1$$

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 Optimality Does Not Imply Efficient Optimization

Our loss is still highly nonconvex:

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

- **Global optimality:** Any global solution $(\{H^*, W^*, b^*\})$ 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.

![Diagram showing comparison between general nonconvex problems and our training problem.](image)
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. \((W, H)\). Then

- **Global optimality:** Any global solution \((H^*, W^*, b^*)\) 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

• **A feature learning 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

- **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.**
Implications of Our Results

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

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

- 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

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

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

- 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

\[
Y = \begin{bmatrix}
1 & \cdots & 1 \\
1 & \cdots & 1 \\
1 & \cdots & 1 \\
\end{bmatrix} = I_K \otimes 1_n^\top
\]
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

- 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. $d = \#\text{class } K$ (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. $d = 10$

Qing Qu (EECS, University of Michigan)
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. $d = 10$

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

1. Low-Dimensional Representation: Neural Collapse (NC)
2. Understanding NC from Optimization
3. Prevalence of NC under Different Training Scenarios
4. Conclusion
**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\(^3\)

\[ CE(p_t) = - \log(p_t) \]
\[ FL(p_t) = -(1 - p_t)^\gamma \log(p_t) \]

---

\(^3\)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\(^4\).

\[
\begin{align*}
\text{Soft label} & = \begin{bmatrix} 1 - \alpha \\ \alpha/2 \\ \alpha/2 \end{bmatrix} \\
\end{align*}
\]

\[
\text{Output: } Wh + b = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}
\]

\[
\text{Prediction} \quad \text{Target}
\]

\[
\begin{align*}
\text{Cat} & : 0.6 \\
\text{Dog} & : 0.3 \\
\text{Panda} & : 0.1
\end{align*}
\]

\[
\begin{align*}
\text{LS} & = - q(\text{Cat}) \cdot \log p(\text{Cat}) \\
& \quad - q(\text{Dog}) \cdot \log p(\text{Dog}) \\
& \quad - q(\text{Panda}) \cdot \log p(\text{Panda})
\end{align*}
\]

\[
\begin{align*}
& = - (1 - \alpha) \log(0.6) \\
& \quad - \frac{\alpha}{2} \log(0.3) \\
& \quad - \frac{\alpha}{2} \log(0.1)
\end{align*}
\]

\[^4\text{Szegedy et al., Rethinking the inception architecture for computer vision, CVPR’16.}
Muller, Kornblith, Hinton, When does label smoothing help?, NeurIPS’19.\]
Example III: Mean-squared Error (MSE) Loss

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

\[
\min_{W,H,b} \frac{1}{2N} \| \Omega_\alpha^{1/2} \odot (WH + b1^\top - MY) \|_F^2 + \frac{\lambda W}{2} \| W \|_F^2 + \frac{\lambda H}{2} \| H \|_F^2 + \frac{\lambda b}{2} \| b \|_2^2.
\]
Example III: Mean-squared Error (MSE) Loss

- Error bound condition for vanilla MSE loss:

\[
\text{dist}((W, H, b), \mathcal{X}) \leq \kappa \| \nabla F(W, H, b) \|_F
\]

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

- Local linear convergence of GD:

![Graphs showing the optimization progress for MSE and CE loss with different regularization strengths.](graphs.png)
Which Loss is the Best to Use?

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

<table>
<thead>
<tr>
<th>Loss</th>
<th>CE</th>
<th>FL</th>
<th>LS</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width = × 0.25, Epoches = 200</td>
<td>71.95</td>
<td>70.20</td>
<td>70.40</td>
<td>69.15</td>
</tr>
<tr>
<td>Width = × 2, Epoches = 800</td>
<td>79.30</td>
<td>79.32</td>
<td>80.20</td>
<td>79.62</td>
</tr>
</tbody>
</table>

- 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. 
\[ d \geq \text{\#class } K - 1 \], 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. \((W, H, b)\)
Are All Loses Created Equal?—A NC Perspective

Theorem (Informal, Zhou et al.'22)

Under the unconstrained feature model, with feature dim. \( d \geq \#\text{class } K - 1 \), 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. \((W, H, b)\)

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 epochs) on CIFAR-10 with **different training losses**

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

Qing Qu (EECS, University of Michigan)
Prevalence of NC under Different Training Scenarios

Are All Losses Created Equal?—A NC Perspective

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

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 epochs) on CIFAR-10 with different training losses

- 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**

- 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

\[
\min_{W,H} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}(W h_{k,i}, y_k)
\]

s.t. \( \|w_k\|_2 = \tau, \|h_{k,i}\|_2 = 1, h_{k,i} = \phi_{\theta}(x_{k,i}), \forall i \in [n], \forall k \in [K]. \)
Neural Collapse with Feature Normalization

\[
\min_{W,H} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}(Wh_{k,i}, y_k)
\]

s.t. \( \|w_k\|_2 = \tau, \|h_{k,i}\|_2 = 1, h_{k,i} = \phi_\theta(x_{k,i}), \forall i \in [n], \forall k \in [K]. \)

- 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:

\[
\min_{W, H} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n} \mathcal{L}_{CE}(W h_{k,i}, y_k)
\]

s.t. \(||w_k||_2 = \tau, \; ||h_{k,i}||_2 = 1, \; \forall \; i \in [n], \; \forall \; k \in [K].\)

- More advanced analysis based upon Riemannian optimization tools.
Experimental Results with Feature Normalization

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

Qing Qu (EECS, University of Michigan)
Neural Collapse for Multi-Label Learning

<table>
<thead>
<tr>
<th>C = 3</th>
<th>Multi-Class</th>
<th>Multi-Label</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Samples</strong></td>
<td><img src="sun.png" alt="Samples" />, <img src="cloud.png" alt="Samples" />, <img src="moon.png" alt="Samples" /></td>
<td><img src="cloud.png" alt="Samples" />, <img src="sun.png" alt="Samples" />, <img src="moon.png" alt="Samples" /></td>
</tr>
<tr>
<td><strong>Labels (t)</strong></td>
<td>[0 0 1], [1 0 0], [0 1 0]</td>
<td>[1 0 1], [0 1 0], [1 1 1]</td>
</tr>
</tbody>
</table>
Multi-label Learning Setup

**Single-label**

\[ \mathcal{L}_{CE}(\psi_\Theta(x), y) \]

\[ \text{Loss} \]

**Multi-label**

Set \( S \)

\[ \sum_{i=1}^{\vert S \vert} \mathcal{L}_{CE}(\psi_\Theta(x), y_i) \]

"Pick-all" Loss

\[ \text{Label } y_1 \]

\[ \text{Label } y_{\mid S \mid} \]
• 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) $\mathcal{N}_1$ (MLab-MNIST) (b) $\mathcal{N}_2$ (MLab-MNIST) (c) $\mathcal{N}_3$ (MLab-MNIST) (d) $\mathcal{N}_m$ (MLab-MNIST)

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

1. Low-Dimensional Representation: Neural Collapse (NC)
2. Understanding NC from Optimization
3. Prevalence of NC under Different Training Scenarios
4. Conclusion
References


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.

SCAN ME