A Top-Down Approach Toward Understanding Deep Learning

Weijie Su

University of Pennsylvania

Otaru University of Commerce, November 4, 2021

A new paradigm of "science": deep learning

- Collect data and buy GPU first

A new paradigm of “science”: deep learning

- Collect data and buy GPU first
- Scale model with data and computational resources

A new paradigm of “science”: deep learning

- Collect data and buy GPU first
- Scale model with data and computational resources
- End to end: Representation, computation, prediction

A new paradigm of “science”: deep learning

- Collect data and buy GPU first
- Scale model with data and computational resources
- End to end: Representation, computation, prediction

The Bitter Lesson

Rich Sutton

March 13, 2019

The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin. The ultimate reason for this is Moore's law, or rather its generalization of continued exponentially falling cost per unit of computation. Most AI research has been conducted as if the computation available to the agent were constant (in which case leveraging human knowledge would be one of the only ways to improve performance) but, over a slightly longer time than a typical research project, massively more computation inevitably becomes available. Seeking an improvement that makes a difference in the shorter term, researchers seek to leverage their human knowledge of the domain, but the only thing that matters in the long run is the leveraging of computation. These two need not run counter to each other, but in practice they tend to. Time spent on one is time not spent on the other. There are psychological commitments to investment in one approach or the other. And the human-knowledge approach tends to complicate methods in ways that make them less suited to taking advantage of general methods leveraging computation. There were many examples of AI researchers' belated learning of this bitter lesson, and it is instructive to review some of the most prominent.

However, making deep learning a science requires...

- Why don't heavily parameterized neural networks overfit the data?

However, making deep learning a science requires...

- Why don't heavily parameterized neural networks overfit the data?
- What is the effective number of parameters?

However, making deep learning a science requires...

- Why don't heavily parameterized neural networks overfit the data?
- What is the effective number of parameters?
- Why doesn't backpropagation get stuck in poor local minima with low value of the loss function, yet bad test error?

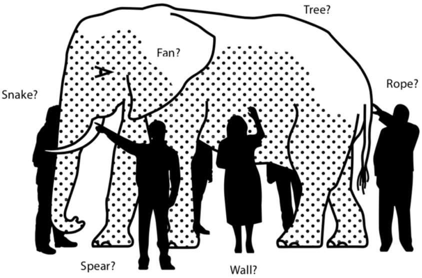
However, making deep learning a science requires...

Leo Breiman

- Why don't heavily parameterized neural networks overfit the data?
- What is the effective number of parameters?
- Why doesn't backpropagation get stuck in poor local minima with low value of the loss function, yet bad test error?

Disclaimer: This talk doesn't attempt to answer these fundamental questions...

Have we really understood deep learning?



Limited scopes...

- Assume extremely large width and shallow depth
- Data assumed to be from Gaussian mixtures
- Linear activation
- Use gradient descent instead of stochastic gradient descent
- ......

Need “small” but useful surrogate models

A bitter lesson learned

Very difficult to build a comprehensive foundation for deep learning...

Need “small” but useful surrogate models

A bitter lesson learned

Very difficult to build a comprehensive foundation for deep learning...

Need “small” but useful surrogate models

A bitter lesson learned

Very difficult to build a comprehensive foundation for deep learning...

What is a good *surrogate* model?

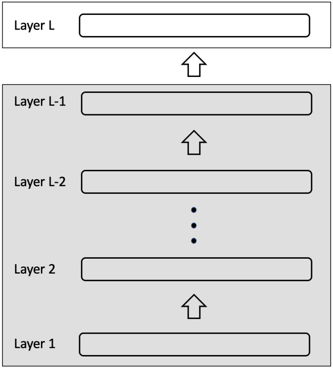
- Mathematically tractable
- Yet maintains some characteristics of deep learning
- Insights into the practice of deep learning

This talk: a top-down viewpoint

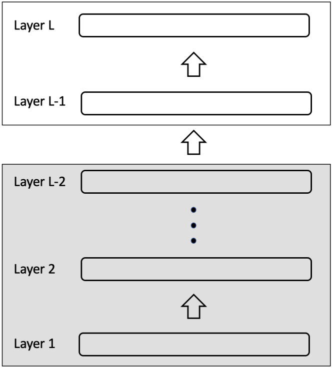
Collaborators

- Cong Fang (Penn CS)
- Hangfeng He (Penn CS)
- Qi Long (Penn Biostats)

Illustration of our top-down approach



(a) 1-Layer-Peeled Model



(b) 2-Layer-Peeled Model

Setup for deep learning

Neural network for K-class classification:

$$\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{W}_{\text{full}}) = \boldsymbol{W}_L \sigma \left(\boldsymbol{W}_{L-1} \sigma(\cdots \sigma(\boldsymbol{W}_1 \boldsymbol{x}) \cdots) \right)$$

- $\sigma(\cdot)$ is a nonlinear activation function
- $\boldsymbol{W}_{\text{full}} := \{\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_L\}$ collects the weights
- Bias omitted

Setup for deep learning

Neural network for K-class classification:

$$\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{W}_{\text{full}}) = \boldsymbol{W}_L \sigma\left(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x})\cdots)\right)$$

- $\sigma(\cdot)$ is a nonlinear activation function
- $\boldsymbol{W}_{\text{full}} := \{\boldsymbol{W}_1, \boldsymbol{W}_2, \ldots, \boldsymbol{W}_L\}$ collects the weights
- Bias omitted

Optimization problem:

$$\min_{\boldsymbol{W}_{\text{full}}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{\text{full}}), \boldsymbol{y}_k) + \frac{\lambda}{2}\|\boldsymbol{W}_{\text{full}}\|^2$$

- $\boldsymbol{y}_k$ is a one-hot vector denoting the k-th class
- λ weight decay parameter, $\mathcal{L}$ cross-entropy loss

A peek at Layer-Peeled Model

$$\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{W}_{\text{full}}) = \boldsymbol{W}_L \sigma\left(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1 \boldsymbol{x})\cdots)\right)$$

$$\min_{\boldsymbol{W}_{\text{full}}} \quad \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{f}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{\text{full}}), \boldsymbol{y}_k) + \frac{\lambda}{2}\|\boldsymbol{W}_{\text{full}}\|^2$$

- Difficult to pinpoint how any layer $\boldsymbol{W}_l$ influences the output

A peek at Layer-Peeled Model

$$f(x; W_{\text{full}}) = W_L \sigma\left(W_{L-1}\sigma(\cdots\sigma(W_1 x)\cdots)\right)$$

$$\min_{W_L, H} \quad \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(W_L h_{k,i}, y_k) + \frac{\lambda}{2}\|W_{\text{full}}\|^2$$

- Difficult to pinpoint how any layer W_l influences the output
- $h_{k,i}$ represents $\sigma\left(W_{L-1}\sigma(\cdots\sigma(W_1 x_{k,i})\cdots)\right)$

A peek at Layer-Peeled Model

$$\boldsymbol{f}(\boldsymbol{x}; \boldsymbol{W}_{\text{full}}) = \boldsymbol{W}_L \sigma\left(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x})\cdots)\right)$$

$$\begin{aligned}
\min_{\boldsymbol{W}_L, \boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}_L\boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \\
\text{s.t.} \quad & \frac{1}{K}\sum_{k=1}^{K}\left\|\boldsymbol{w}_k\right\|^2 \le E_W \\
& \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\left\|\boldsymbol{h}_{k,i}\right\|^2 \le E_H
\end{aligned}$$

- Difficult to pinpoint how any layer $\boldsymbol{W}_l$ influences the output
- $\boldsymbol{h}_{k,i}$ represents $\sigma\left(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}_{k,i})\cdots)\right)$
- Here $\boldsymbol{W}_L = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]^\top$

Derivation

Rewrite the optimization problem as

$$\min_{\boldsymbol{W}_L, \boldsymbol{H}} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}), \boldsymbol{y}_k) + \frac{\lambda}{2} \|\boldsymbol{W}_L\|^2 + \frac{\lambda}{2} \|\boldsymbol{W}_{-L}\|^2$$

- Last-layer feature $\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}) := \sigma(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1 \boldsymbol{x}_{k,i})\cdots))$

Derivation

Rewrite the optimization problem as

$$\min_{\boldsymbol{W}_L, \boldsymbol{H}} \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}_L\boldsymbol{h}(\boldsymbol{x}_{k,i};\boldsymbol{W}_{-L}),\boldsymbol{y}_k) + \frac{\lambda}{2}\|\boldsymbol{W}_L\|^2 + \frac{\lambda}{2}\|\boldsymbol{W}_{-L}\|^2$$

- Last-layer feature $\boldsymbol{h}(\boldsymbol{x}_{k,i};\boldsymbol{W}_{-L}) := \sigma(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1\boldsymbol{x}_{k,i})\cdots))$

From the *dual* viewpoint, a minimum is an optimal solution to

$$\begin{aligned}\min_{\boldsymbol{W}_L, \boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}_L\boldsymbol{h}_{k,i},\boldsymbol{y}_k) \\ \text{s.t.} \quad & \|\boldsymbol{W}_L\|^2 \leqslant C_1 \\ & \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2\end{aligned}$$

Derivation

Rewrite the optimization problem as

$$\min_{\boldsymbol{W}_L, \boldsymbol{H}} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}), \boldsymbol{y}_k) + \frac{\lambda}{2} \|\boldsymbol{W}_L\|^2 + \frac{\lambda}{2} \|\boldsymbol{W}_{-L}\|^2$$

- Last-layer feature $\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}) := \sigma(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1 \boldsymbol{x}_{k,i})\cdots))$

From the *dual* viewpoint, a minimum is an optimal solution to

$$\begin{aligned} \min_{\boldsymbol{W}_L, \boldsymbol{H}} \quad & \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \\ \text{s.t.} \quad & \|\boldsymbol{W}_L\|^2 \leqslant C_1 \\ & \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2 \end{aligned}$$

- Not a one-to-one mapping

Derivation

Rewrite the optimization problem as

$$\min_{\boldsymbol{W}_L, \boldsymbol{H}} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}), \boldsymbol{y}_k) + \frac{\lambda}{2} \|\boldsymbol{W}_L\|^2 + \frac{\lambda}{2} \|\boldsymbol{W}_{-L}\|^2$$

- Last-layer feature $\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}) := \sigma(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1 \boldsymbol{x}_{k,i})\cdots))$

From the *dual* viewpoint, a minimum is an optimal solution to

$$\begin{aligned} \min_{\boldsymbol{W}_L, \boldsymbol{H}} \quad & \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \\ \text{s.t.} \quad & \|\boldsymbol{W}_L\|^2 \leqslant C_1 \\ & \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2 \Leftrightarrow \boldsymbol{H} \in \left\{\boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2\right\} \end{aligned}$$

- Not a one-to-one mapping
- $\boldsymbol{H}(\boldsymbol{W}_{-L}) := [\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}) : 1 \leqslant k \leqslant K, 1 \leqslant i \leqslant n_k]$

Derivation

Rewrite the optimization problem as

$$\min_{\boldsymbol{W}_L, \boldsymbol{H}} \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}), \boldsymbol{y}_k) + \frac{\lambda}{2} \|\boldsymbol{W}_L\|^2 + \frac{\lambda}{2} \|\boldsymbol{W}_{-L}\|^2$$

- Last-layer feature $\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}) := \sigma(\boldsymbol{W}_{L-1}\sigma(\cdots\sigma(\boldsymbol{W}_1 \boldsymbol{x}_{k,i})\cdots))$

From the *dual* viewpoint, a minimum is an optimal solution to

$$\begin{aligned} \min_{\boldsymbol{W}_L, \boldsymbol{H}} \quad & \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \\ \text{s.t.} \quad & \|\boldsymbol{W}_L\|^2 \leqslant C_1 \\ & \boldsymbol{H} \in \left\{ \boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2 \right\} \end{aligned}$$

- Not a one-to-one mapping
- $\boldsymbol{H}(\boldsymbol{W}_{-L}) := [\boldsymbol{h}(\boldsymbol{x}_{k,i}; \boldsymbol{W}_{-L}) : 1 \leqslant k \leqslant K, 1 \leqslant i \leqslant n_k]$

Derivation: an *ansatz*

Assumption

$$\left\{ \boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2 \right\} \approx \left\{ \boldsymbol{H} : \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \|\boldsymbol{h}_{k,i}\|^2 \leqslant C_2' \right\}$$

Derivation: an *ansatz*

Assumption

$$\left\{\boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2\right\} \approx \left\{\boldsymbol{H} : \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \|\boldsymbol{h}_{k,i}\|^2 \leqslant C_2'\right\}$$

$$\begin{aligned}
\min_{\boldsymbol{W}_L, \boldsymbol{H}} \quad & \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \\
\text{s.t.} \quad & \|\boldsymbol{W}_L\|^2 \leqslant C_1 \\
& \boldsymbol{H} \in \left\{\boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2\right\}
\end{aligned}$$

Derivation: an *ansatz*

Assumption

$$\left\{\boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2\right\} \approx \left\{\boldsymbol{H} : \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \|\boldsymbol{h}_{k,i}\|^2 \leqslant C_2'\right\}$$

$$\begin{aligned}
\min_{\boldsymbol{W}_L, \boldsymbol{H}} \quad & \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}_L \boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \\
\text{s.t.} \quad & \|\boldsymbol{W}_L\|^2 \leqslant C_1 \\
& \boldsymbol{H} \in \left\{\boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2\right\}
\end{aligned}$$

$$\begin{aligned}
\min_{\boldsymbol{W}, \boldsymbol{H}} \quad & \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W} \boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \\
\text{s.t.} \quad & \frac{1}{K} \sum_{k=1}^{K} \|\boldsymbol{w}_k\|^2 \leq E_W \\
& \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \|\boldsymbol{h}_{k,i}\|^2 \leq E_H
\end{aligned}$$

- Self-duality of ℓ_2 spaces
- More justification for the ansatz later

More on Layer-Peeled Model

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

$$\text{s.t.} \quad \boxed{\frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W}$$

$$\boxed{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le E_H}$$

Prediction constraint

Representation constraint

More on Layer-Peeled Model

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

$$\text{s.t.} \quad \boxed{\frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W}$$

$$\boxed{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le E_H}$$

Prediction constraint

Representation constraint

- Terminal phase of deep learning training

More on Layer-Peeled Model

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

$$\text{s.t.} \quad \boxed{\frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W}$$

$$\boxed{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le E_H}$$

Prediction constraint

Representation constraint

- Terminal phase of deep learning training
- E_W, E_H depend on weight decay λ

More on Layer-Peeled Model

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

$$\text{s.t.} \quad \boxed{\frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W}$$

$$\boxed{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le E_H}$$

Prediction constraint

Representation constraint

- Terminal phase of deep learning training
- E_W, E_H depend on weight decay λ
- Nonconvex but analytically tractable

More on Layer-Peeled Model

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

$$\text{s.t.} \quad \boxed{\frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W}$$

$$\boxed{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le E_H}$$

Prediction constraint

Representation constraint

- Terminal phase of deep learning training
- E_W, E_H depend on weight decay λ
- Nonconvex but analytically tractable
- Does not explicitly depend on data

More on Layer-Peeled Model

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

$$\text{s.t.} \quad \boxed{\frac{1}{K}\sum_{k=1}^{K}\left\|\boldsymbol{w}_k\right\|^2 \le E_W}$$

$$\boxed{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\left\|\boldsymbol{h}_{k,i}\right\|^2 \le E_H}$$

Prediction constraint

Representation constraint

- Terminal phase of deep learning training
- E_W, E_H depend on weight decay λ
- Nonconvex but analytically tractable
- Does not explicitly depend on data
    - Cons: information lost

More on Layer-Peeled Model

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

$$\text{s.t.} \quad \boxed{\frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W}$$

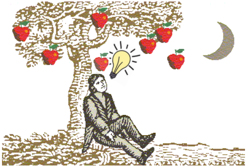
$$\boxed{\frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le E_H}$$

Prediction constraint

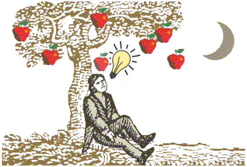
Representation constraint

- Terminal phase of deep learning training
- E_W, E_H depend on weight decay λ
- Nonconvex but analytically tractable
- Does not explicitly depend on data
  - Cons: information lost
  - Pros: robust conclusion

Ask me anything about this "apple"

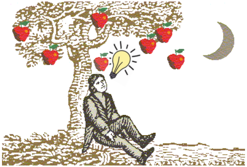


Ask me anything about this "apple"



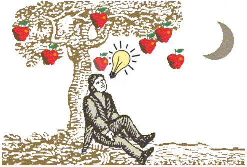
Is it mathematically tractable?

Ask me anything about this “apple”



Is it mathematically tractable? Yes

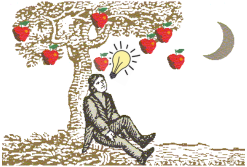
Ask me anything about this “apple”



Is it mathematically tractable? Yes

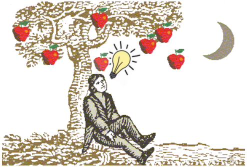
Does it maintain some characteristics of deep learning?

Ask me anything about this "apple"



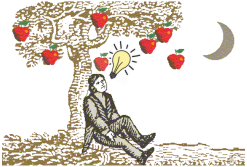
Is it mathematically tractable?	Yes
Does it maintain some characteristics of deep learning?	Yes

Ask me anything about this “apple”



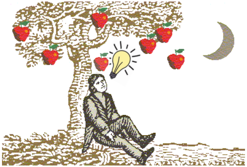
Is it mathematically tractable?	Yes
Does it maintain some characteristics of deep learning?	Yes
Can it provide insights into the practice of deep learning?	

Ask me anything about this “apple”



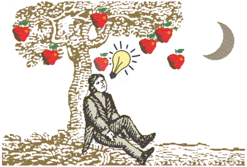
Is it mathematically tractable?	Yes
Does it maintain some characteristics of deep learning?	Yes
Can it provide insights into the practice of deep learning?	I think so

Ask me anything about this "apple"



Is it mathematically tractable?	Yes
Does it maintain some characteristics of deep learning?	Yes
Can it provide insights into the practice of deep learning?	I think so
Does it answer Leo Breiman's questions?	

Ask me anything about this “apple”



Is it mathematically tractable?	Yes
Does it maintain some characteristics of deep learning?	Yes
Can it provide insights into the practice of deep learning?	I think so
Does it answer Leo Breiman’s questions?	Unfortunately, not

Outline

Balanced training

All class sizes are equal: $n_1 = n_2 = \cdots = n_K$

Balanced training

All class sizes are equal: $n_1 = n_2 = \cdots = n_K$

What can the Layer-Peeled Model say?

Balanced training

All class sizes are equal: $n_1 = n_2 = \cdots = n_K$

What can the Layer-Peeled Model say?

Theorem

Any global minimizer $\boldsymbol{W}^\star \equiv [\boldsymbol{w}_1^\star, \ldots, \boldsymbol{w}_K^\star]^\top, \boldsymbol{H}^\star \equiv [\boldsymbol{h}_{k,i}^\star : 1 \leqslant k \leqslant K, 1 \leqslant i \leqslant n]$ *with cross-entropy loss obeys*

$$\boldsymbol{h}_{k,i}^\star = C\boldsymbol{w}_k^\star = C'\boldsymbol{m}_k^\star,$$

where $[\boldsymbol{m}_1^\star, \ldots, \boldsymbol{m}_K^\star]$ *forms a* K*-simplex equiangular tight frame (ETF)*

- $\boldsymbol{h}_{k,i}^\star$ depends only on the class membership!
- $C = \sqrt{E_H/E_W}, C' = \sqrt{E_H}$

Balanced training

All class sizes are equal: $n_1 = n_2 = \cdots = n_K$

What can the Layer-Peeled Model say?

Theorem

Any global minimizer $\boldsymbol{W}^\star \equiv \left[\boldsymbol{w}_1^\star, \ldots, \boldsymbol{w}_K^\star\right]^\top, \boldsymbol{H}^\star \equiv [\boldsymbol{h}_{k,i}^\star : 1 \leqslant k \leqslant K, 1 \leqslant i \leqslant n]$ *with cross-entropy loss obeys*

$$\boldsymbol{h}_{k,i}^\star = C\boldsymbol{w}_k^\star = C'\boldsymbol{m}_k^\star,$$

where $[\boldsymbol{m}_1^\star, \ldots, \boldsymbol{m}_K^\star]$ *forms a* K*-simplex equiangular tight frame (ETF)*

- $\boldsymbol{h}_{k,i}^\star$ depends only on the class membership!
- $C = \sqrt{E_H/E_W}, C' = \sqrt{E_H}$
- What is a K-simplex ETF?

K-simplex ETF

K equal-length vectors form the *largest* possible equal-sized angles between any pair

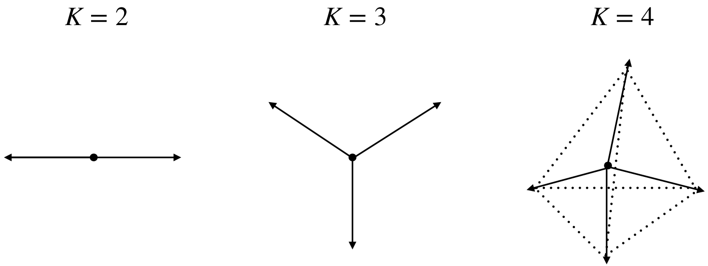
Equivalently, random variables $\xi_1, \ldots, \xi_K$ of mean 0 and variance 1. If $\mathbb{E}\xi_i\xi_j = \rho$ for all $i \neq j$, what's the min of ρ?

K-simplex ETF

K equal-length vectors form the *largest* possible equal-sized angles between any pair

Equivalently, random variables $\xi_1, \ldots, \xi_K$ of mean 0 and variance 1. If $\mathbb{E}\xi_i\xi_j = \rho$ for all $i \neq j$, what's the min of ρ?

$$\text{largest angle} = \arccos\left(-\frac{1}{K-1}\right)$$



Return to the theorem for balanced training

All class sizes are equal: $n_1 = n_2 = \cdots = n_K$

Theorem

The solution to the Layer-Peeled Model in balanced training satisfies

$$\boldsymbol{h}_{k,i}^\star = C\boldsymbol{w}_k^\star = C'\boldsymbol{m}_k^\star$$

- German shepherd, husky, chihuahua, rottweiler are all dogs!

Return to the theorem for balanced training

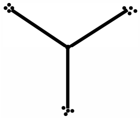
All class sizes are equal: $n_1 = n_2 = \cdots = n_K$

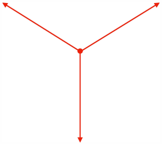
Theorem

The solution to the Layer-Peeled Model in balanced training satisfies

$$\boldsymbol{h}^\star_{k,i} = C\boldsymbol{w}^\star_k = C'\boldsymbol{m}^\star_k$$

- German shepherd, husky, chihuahua, rottweiler are all dogs!





Return to the theorem for balanced training

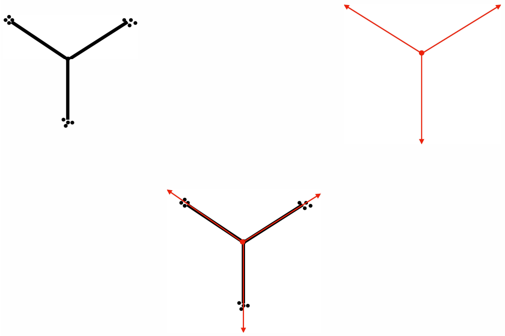
All class sizes are equal: $n_1 = n_2 = \cdots = n_K$

Theorem

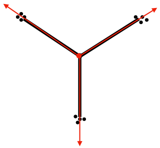
The solution to the Layer-Peeled Model in balanced training satisfies

$$\boldsymbol{h}^\star_{k,i} = C\boldsymbol{w}^\star_k = C'\boldsymbol{m}^\star_k$$

- German shepherd, husky, chihuahua, rottweiler are all dogs!

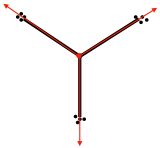


This is simply neural collapse



Papyan, Han, and Donoho discovered *neural collapse* in 2020:

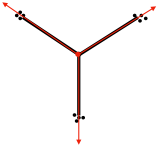
This is simply neural collapse



Papyan, Han, and Donoho discovered *neural collapse* in 2020:

1. Variability collapse: features collapse to their class means

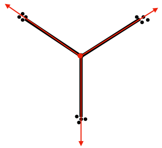
This is simply neural collapse



Papyan, Han, and Donoho discovered *neural collapse* in 2020:

1. Variability collapse: features collapse to their class means
2. Class means centered at their global mean collapse to ETF

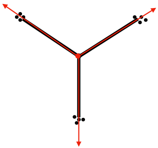
This is simply neural collapse



Papyan, Han, and Donoho discovered *neural collapse* in 2020:

1. Variability collapse: features collapse to their class means
2. Class means centered at their global mean collapse to ETF
3. Up to scaling, last-layer classifiers each collapse to class means

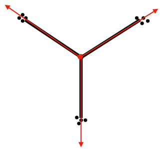
This is simply neural collapse



Papyan, Han, and Donoho discovered *neural collapse* in 2020:

1. Variability collapse: features collapse to their class means
2. Class means centered at their global mean collapse to ETF
3. Up to scaling, last-layer classifiers each collapse to class means
4. Classifier's decision collapses to choosing the closet class mean

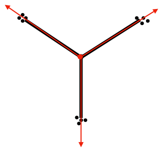
This is simply neural collapse



Papyan, Han, and Donoho discovered *neural collapse* in 2020:

1. Variability collapse: features collapse to their class means
2. Class means centered at their global mean collapse to ETF
3. Up to scaling, last-layer classifiers each collapse to class means
4. Classifier's decision collapses to choosing the closet class mean

This is simply neural collapse



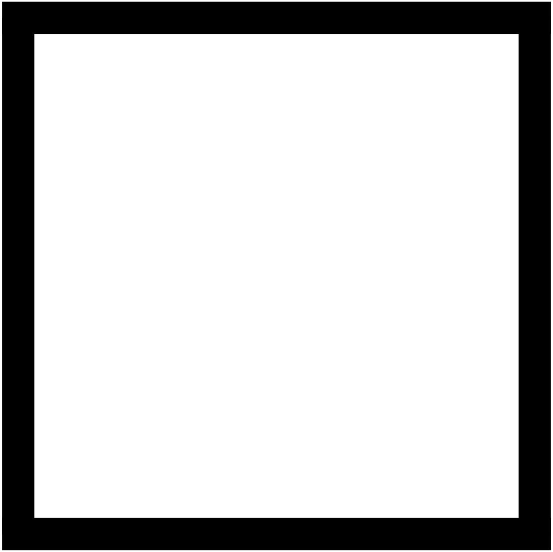
Papyan, Han, and Donoho discovered *neural collapse* in 2020:

1. Variability collapse: features collapse to their class means
2. Class means centered at their global mean collapse to ETF
3. Up to scaling, last-layer classifiers each collapse to class means
4. Classifier’s decision collapses to choosing the closet class mean

Implications on better generalization, large margin, and robustness

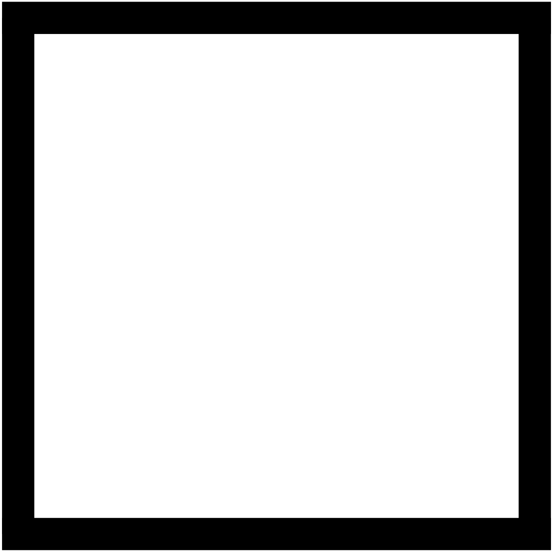
Concurrent works [MPP20, EW20, LS20] also justified neural collapse using different models

Animation of neural collapse



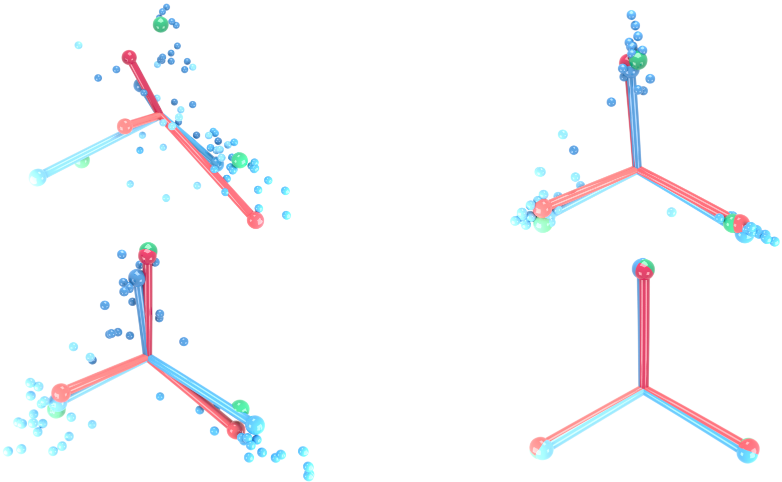
Credit: Papyan, Han, and Donoho

Animation of neural collapse



Credit: Papyan, Han, and Donoho

Snapshot of neural collapse



Credit: Papyan, Han, and Donoho

Neural collapse can justify the Layer-Peeled Model

About the ansatz

Recall

$$\left\{\boldsymbol{H}(\boldsymbol{W}_{-L}) : \|\boldsymbol{W}_{-L}\|^2 \leqslant C_2\right\} \approx \left\{\boldsymbol{H} : \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \|\boldsymbol{h}_{k,i}\|^2 \leqslant C_2'\right\}$$

This gives

$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{N} \sum_{k=1}^{K} \sum_{i=1}^{n_k} \mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i}, \boldsymbol{y}_k) \\ \text{s.t.} \quad & \frac{1}{K} \sum_{k=1}^{K} \|\boldsymbol{w}_k\|^2 \leq E_W \\ & \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \|\boldsymbol{h}_{k,i}\|^2 \leq E_H \end{aligned}$$

What happens without the ansatz?

Without the ansatz:

$$
\begin{aligned}
\min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k) \\
\text{s.t.} \quad & \frac{1}{K}\sum_{k=1}^{K}\left\|\boldsymbol{w}_k\right\|^2 \le E_W \\
& \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{h}_{k,i}\|_q^q \le E_H
\end{aligned}
$$

What happens without the ansatz?

Without the ansatz:

$$
\begin{aligned}
\min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k) \\
\text{s.t.} \quad & \frac{1}{K}\sum_{k=1}^{K}\left\|\boldsymbol{w}_k\right\|^2 \leq E_W \\
& \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{h}_{k,i}\|_q^q \leq E_H
\end{aligned}
$$

Theorem

Assume $K \geqslant 3$ and $p \geq K$. For any $q \in (0,2)\cup(2,\infty)$, neural collapse does ***not*** *emerge in the model above*

Is the Layer-Peeled Model satisfactory?

Is the Layer-Peeled Model satisfactory?
A higher standard: can it predict new stuff?

Outline

1. Explaining Neural Collapse

2. Predicting Minority Collapse

3. How to Mitigate Minority Collapse?

Imbalanced training

Datasets often have a disproportionate ratio of observations in each class

Imbalanced training

Datasets often have a disproportionate ratio of observations in each class

As a simple starting point, assume

- The first K_A majority classes each contain n_A training examples ($n_1 = n_2 = \cdots = n_{K_A} = n_A$)

Imbalanced training

Datasets often have a disproportionate ratio of observations in each class

As a simple starting point, assume

- The first K_A majority classes each contain n_A training examples ($n_1 = n_2 = \cdots = n_{K_A} = n_A$)
- The remaining $K_B := K - K_A$ minority classes each contain n_B examples ($n_{K_A+1} = n_{K_A+2} = \cdots = n_K = n_B$)

Imbalanced training

Datasets often have a disproportionate ratio of observations in each class

As a simple starting point, assume

- The first K_A majority classes each contain n_A training examples $(n_1 = n_2 = \cdots = n_{K_A} = n_A)$
- The remaining $K_B := K - K_A$ minority classes each contain n_B examples $(n_{K_A+1} = n_{K_A+2} = \cdots = n_K = n_B)$
- Call $R := n_A/n_B > 1$ the imbalance ratio

No closed-form expression for the solutions to LPM…

Technique: Convex relaxation

- Define $\boldsymbol{h}_k$ as the feature mean of the k-th class

$$\boldsymbol{h}_k := \frac{1}{n_k} \sum_{i=1}^{n_k} \boldsymbol{h}_{k,i}$$

- Introduce a new decision variable

$$\boldsymbol{X} := \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right]^\top \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right] \in \mathbb{R}^{2K \times 2K}$$

Technique: Convex relaxation

- Define $\boldsymbol{h}_k$ as the feature mean of the k-th class

$$\boldsymbol{h}_k := \frac{1}{n_k} \sum_{i=1}^{n_k} \boldsymbol{h}_{k,i}$$

- Introduce a new decision variable

$$\boldsymbol{X} := \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right]^\top \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right] \in \mathbb{R}^{2K \times 2K}$$

Then

- $\boldsymbol{X}$ is positive semidefinite

Technique: Convex relaxation

- Define $\boldsymbol{h}_k$ as the feature mean of the k-th class

$$\boldsymbol{h}_k := \frac{1}{n_k}\sum_{i=1}^{n_k}\boldsymbol{h}_{k,i}$$

- Introduce a new decision variable

$$\boldsymbol{X} := \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right]^\top \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right] \in \mathbb{R}^{2K\times 2K}$$

Then

- $\boldsymbol{X}$ is positive semidefinite
-
$$\frac{1}{K}\sum_{k=1}^{K}\boldsymbol{X}(k,k) = \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{h}_k\|^2 \leq \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \leq E_H$$

Technique: Convex relaxation

- Define $\boldsymbol{h}_k$ as the feature mean of the k-th class

$$\boldsymbol{h}_k := \frac{1}{n_k} \sum_{i=1}^{n_k} \boldsymbol{h}_{k,i}$$

- Introduce a new decision variable

$$\boldsymbol{X} := \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right]^\top \left[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_K, \boldsymbol{W}^\top\right] \in \mathbb{R}^{2K \times 2K}$$

Then

- $\boldsymbol{X}$ is positive semidefinite
-

$$\frac{1}{K} \sum_{k=1}^{K} \boldsymbol{X}(k,k) = \frac{1}{K} \sum_{k=1}^{K} \|\boldsymbol{h}_k\|^2 \leq \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \|\boldsymbol{h}_{k,i}\|^2 \leq E_H$$

-

$$\frac{1}{K} \sum_{k=K+1}^{2K} \boldsymbol{X}(k,k) = \frac{1}{K} \sum_{k=1}^{K} \|\boldsymbol{w}_k\|^2 \leq E_W$$

Technique: Convex relaxation

$$
\begin{aligned}
\min_{\boldsymbol{X} \in \mathbb{R}^{2K \times 2K}} \quad & \sum_{k=1}^{K} \frac{n_k}{N} \mathcal{L}(\boldsymbol{z}_k, \boldsymbol{y}_k) \\
\text{s.t.} \quad & \boldsymbol{z}_k = \left[\boldsymbol{X}(k, K+1), \boldsymbol{X}(k, K+2), \ldots, \boldsymbol{X}(k, 2K) \right]^\top \\
& \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{X}(k,k) \le E_H, \quad \frac{1}{K} \sum_{k=K+1}^{2K} \boldsymbol{X}(k,k) \le E_W \\
& \boldsymbol{X} \succeq 0
\end{aligned}
$$

Technique: Convex relaxation

$$
\begin{aligned}
\min_{\boldsymbol{X} \in \mathbb{R}^{2K \times 2K}} \quad & \sum_{k=1}^{K} \frac{n_k}{N} \mathcal{L}(\boldsymbol{z}_k, \boldsymbol{y}_k) \\
\text{s.t.} \quad & \boldsymbol{z}_k = \left[\boldsymbol{X}(k, K+1), \boldsymbol{X}(k, K+2), \ldots, \boldsymbol{X}(k, 2K) \right]^\top \\
& \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{X}(k,k) \le E_H, \quad \frac{1}{K} \sum_{k=K+1}^{2K} \boldsymbol{X}(k,k) \le E_W \\
& \boldsymbol{X} \succeq 0
\end{aligned}
$$

- Not a semidefinite program in the strict sense because a semidefinite program uses a linear objective function

Nonconvex optimization via convex optimization

Lemma

Assume $p \geq 2K$ and $\mathcal{L}$ is convex in its first argument. Let $\boldsymbol{X}^\star$ be a minimizer of the convex relaxation. Define $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ as

$$\left[\boldsymbol{h}_1^\star, \boldsymbol{h}_2^\star, \ldots, \boldsymbol{h}_K^\star,\ (\boldsymbol{W}^\star)^\top\right] = \boldsymbol{P}(\boldsymbol{X}^\star)^{1/2}$$
$$\boldsymbol{h}_{k,i}^\star = \boldsymbol{h}_k^\star,\ \text{for all } 1 \leqslant i \leqslant n, 1 \leqslant k \leqslant K$$

Then $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ is a minimizer of the Layer-Peeled Model

Nonconvex optimization via convex optimization

Lemma

Assume $p \geq 2K$ and $\mathcal{L}$ is convex in its first argument. Let $\boldsymbol{X}^\star$ be a minimizer of the convex relaxation. Define $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ as

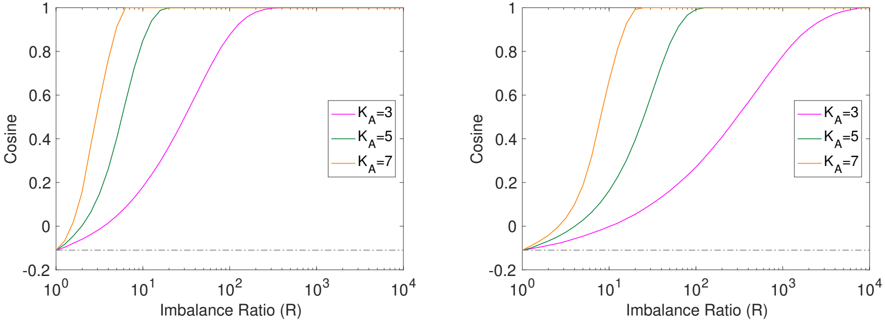
$$\left[\boldsymbol{h}_1^\star, \boldsymbol{h}_2^\star, \ldots, \boldsymbol{h}_K^\star, \ (\boldsymbol{W}^\star)^\top\right] = \boldsymbol{P}(\boldsymbol{X}^\star)^{1/2}$$
$$\boldsymbol{h}_{k,i}^\star = \boldsymbol{h}_k^\star, \ \textit{for all } 1 \leqslant i \leqslant n, 1 \leqslant k \leqslant K$$

Then $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ is a minimizer of the Layer-Peeled Model

- No loss of information when we study the Layer-Peeled Model through a convex program
- But class means *no longer* collapse to classifiers
- Alternatives of convex relaxation exist [BMP08, HV19]

A numerical surprise

Average cosine of between-**minority**-class angles



(c) $E_W = 1, E_H = 5$ (d) $E_W = 1, E_H = 10$

1. When $R < R_0$ for some $R_0 > 0$, average between-minority-class angle becomes smaller as R increases
2. Once $R \geq R_0$, average between-minority-class angle becomes **0**: implying that all minority classifiers collapse!

Minority Collapse

1. When $R < R_0$ for some $R_0 > 0$, average between-minority-class angle becomes smaller as R increases
2. Once $R \geq R_0$, average between-minority-class angle becomes **0**: implying that all minority classifiers collapse!

Proposition

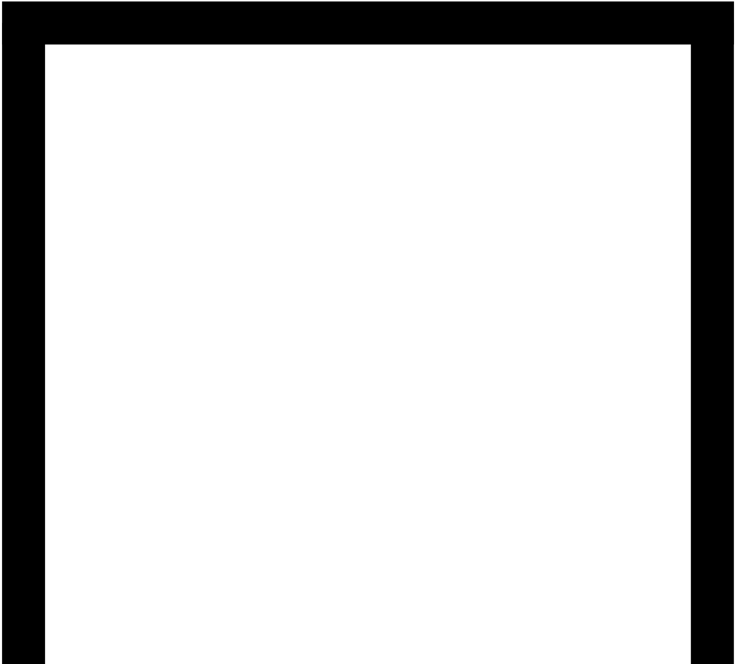
Let $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ be any global minimizer of the Layer-Peeled Model. As $R \equiv n_A/n_B \to \infty$, we have

$$\lim \boldsymbol{w}_k^\star - \boldsymbol{w}_{k'}^\star = \mathbf{0}_p, \quad \textit{for all } K_A < k < k' \leqslant K$$

- The prediction on the minority classes becomes *completely at random*

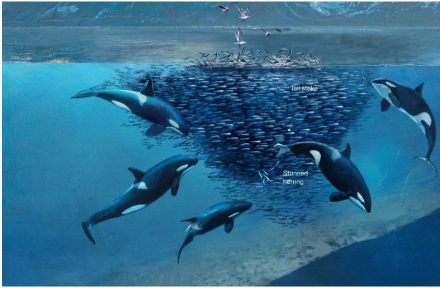
Illustration of Minority Collapse

Illustration of Minority Collapse



Intuition for Minority Collapse

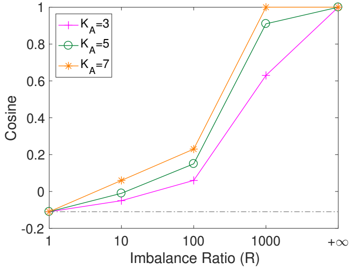
$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k) \\ \text{s.t.} \quad & \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W \\ & \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2 \le E_H \end{aligned}$$



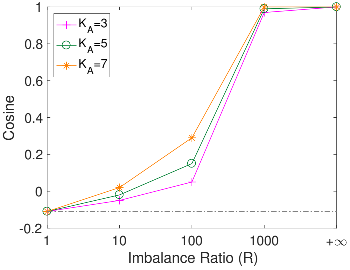
Competition for space!

Is Minority Collapse a real thing?

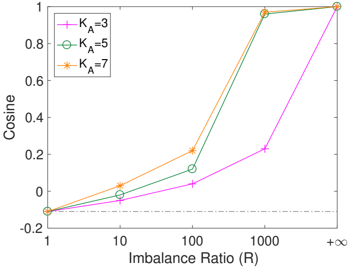
Minority Collapse in experiments



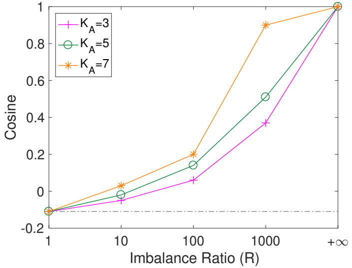
(e) VGG11 on FashionMNIST



(f) VGG13 on CIFAR10

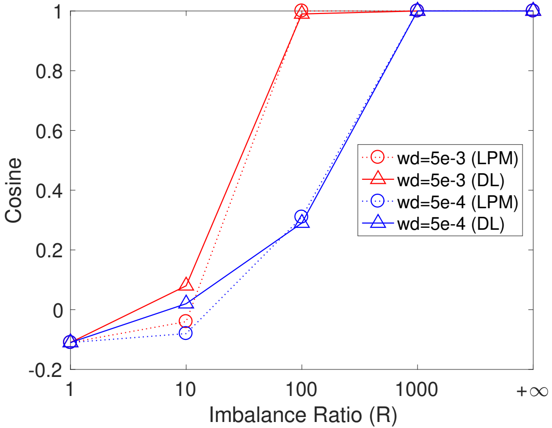


(g) ResNet18 on FashionMNIST



(h) ResNet18 on CIFAR10

LPM predictions match experiments



Layer-Peeled Model (LPM, in dotted lines) and real DNNs (DL, in solid lines) with VGG on CIFAR10

Outline

1. Explaining Neural Collapse

2. Predicting Minority Collapse

3. How to Mitigate Minority Collapse?

Idea: make the minority stronger!

Oversample minority classes

Oversampling duplicates training example from minority classes [JK09]

Oversample minority classes

Oversampling duplicates training example from minority classes [JK09]

The adjusted optimization problem:

$$\frac{1}{n_A K_A + w_r n_B K_B}\left[\sum_{k=1}^{K_A}\sum_{i=1}^{n_A}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}_{k,i};\boldsymbol{W}_{\mathsf{full}}),\boldsymbol{y}_k)\right.$$
$$\left.+\, w_r\sum_{k=K_A+1}^{K}\sum_{i=1}^{n_B}\mathcal{L}(\boldsymbol{f}(\boldsymbol{x}_{k,i};\boldsymbol{W}_{\mathsf{full}}),\boldsymbol{y}_k)\right]$$

while keeping the penalty term $\frac{\lambda}{2}\|\boldsymbol{W}_{\mathsf{full}}\|^2$

Layer-Peeled Model with oversampling

$$\begin{aligned}
\min_{\boldsymbol{H},\boldsymbol{W}} \quad & \frac{1}{n_A K_A + w_r n_B K_B}\left[\sum_{k=1}^{K_A}\sum_{i=1}^{n_A}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k) + w_r \sum_{k=K_A+1}^{K}\sum_{i=1}^{n_B}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k)\right] \\
\text{s.t.} \quad & \frac{1}{K}\sum_{k=1}^{K}\left\|\boldsymbol{w}_k\right\|^2 \le E_W \\
& \frac{1}{K}\sum_{k=1}^{K_A}\frac{1}{n_A}\sum_{i=1}^{n_A}\left\|\boldsymbol{h}_{k,i}\right\|^2 + \frac{1}{K}\sum_{k=K_A+1}^{K}\frac{1}{n_B}\sum_{i=1}^{n_B}\left\|\boldsymbol{h}_{k,i}\right\|^2 \le E_H
\end{aligned}$$

Layer-Peeled Model with oversampling

Theorem

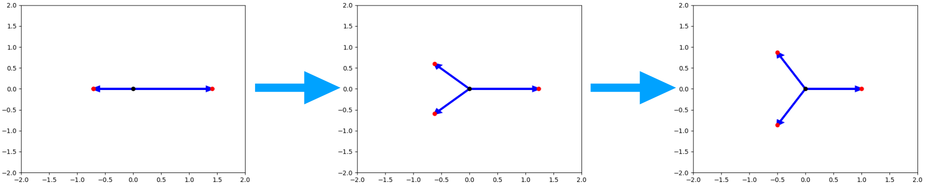
Assume $p \geq 2K$ and $\mathcal{L}$ is convex in the first argument. Let $\boldsymbol{X}^\star$ be any minimizer of the convex relaxation with $n_1 = n_2 = \cdots = n_{K_A} = n_A$ and $n_{K_A+1} = n_{K_A+2} = \cdots = n_K = w_r n_B$. Define $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ as

$$\left[\boldsymbol{h}_1^\star, \boldsymbol{h}_2^\star, \ldots, \boldsymbol{h}_K^\star, (\boldsymbol{W}^\star)^\top\right] = \boldsymbol{P}(\boldsymbol{X}^\star)^{1/2}$$
$$\boldsymbol{h}_{k,i}^\star = \boldsymbol{h}_k^\star, \ \ \text{for all } 1 \leqslant i \leqslant n_A, 1 \leqslant k \leqslant K_A$$
$$\boldsymbol{h}_{k,i}^\star = \boldsymbol{h}_k^\star, \ \ \text{for all } 1 \leqslant i \leqslant n_B, K_A < k \leqslant K$$

Then $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ is a global minimizer of the oversampling-adjusted Layer-Peeled Model.

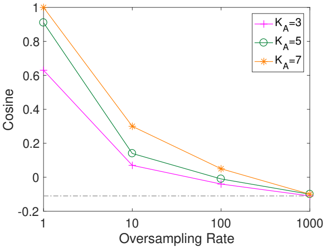
- The size of minority class is now in effect $w_r n_B$ instead of n_B
- If the oversampling rate $w_r = n_A/n_B \equiv R$, neural collapse is back!

Effect of oversampling, in theory

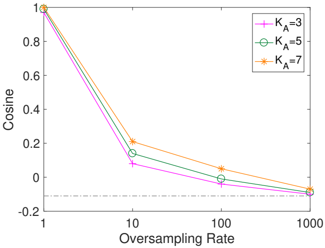


Can oversampling really resolve Minority Collapse?

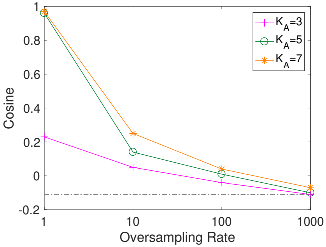
Oversampling mitigates Minority Collapse



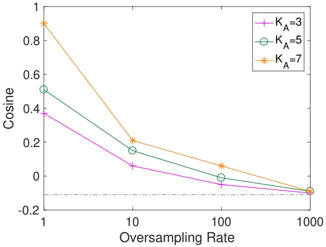
(i) VGG11 on FashionMNIST



(j) VGG13 on CIFAR10



(k) ResNet18 on FashionMNIST



(l) ResNet18 on CIFAR10

Test performance

Network architecture	VGG11			ResNet18		
No. of majority classes	$K_A = 3$	$K_A = 5$	$K_A = 7$	$K_A = 3$	$K_A = 5$	$K_A = 7$
Original (minority)	15.29	20.30	17.00	30.66	34.26	5.53
Oversampling (minority)	**41.13**	**57.22**	**30.50**	**37.86**	**53.46**	**8.13**
Improvement (minority)	25.84	36.92	13.50	7.20	19.20	2.60
Original (overall)	40.10	57.61	69.09	50.88	64.89	66.13
Oversampling (overall)	**58.25**	**76.17**	**73.37**	**55.91**	**74.56**	**67.10**
Improvement (overall)	18.15	18.56	4.28	5.03	9.67	0.97

Table: Test accuracy (%) on FashionMNIST when $R = 1000$. "Original (minority)" means that the test accuracy is evaluated only on the minority classes and oversampling is not used. When oversampling is used, we report the best test accuracy among four oversampling rates: 1, 10, 100, and 1000.

Test performance

Network architecture	VGG11			ResNet18		
No. of majority classes	$K_A = 3$	$K_A = 5$	$K_A = 7$	$K_A = 3$	$K_A = 5$	$K_A = 7$
Original (minority)	15.29	20.30	17.00	30.66	34.26	5.53
Oversampling (minority)	**41.13**	**57.22**	**30.50**	**37.86**	**53.46**	**8.13**
Improvement (minority)	25.84	36.92	13.50	7.20	19.20	2.60
Original (overall)	40.10	57.61	69.09	50.88	64.89	66.13
Oversampling (overall)	**58.25**	**76.17**	**73.37**	**55.91**	**74.56**	**67.10**
Improvement (overall)	18.15	18.56	4.28	5.03	9.67	0.97

Table: Test accuracy (%) on FashionMNIST when $R = 1000$. "Original (minority)" means that the test accuracy is evaluated only on the minority classes and oversampling is not used. When oversampling is used, we report the best test accuracy among four oversampling rates: 1, 10, 100, and 1000.

The best test accuracy is not achieved at $w_r = 1000$, indicating that oversampling with a large w_r would impair the test performance

Remarks on oversampling

- Large value of w_r can mitigate Minority Collapse on the training set
- But might degrade test accuracy

Remarks on oversampling

- Large value of w_r can mitigate Minority Collapse on the training set
- But might degrade test accuracy

- Remains open: how to select an oversampling rate?
- Other approaches such as *fixing* the classifiers?

Concluding remarks

One-line summary

It's a small but useful surrogate model

Future directions

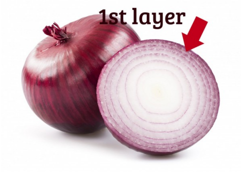
Immediate connections:

- Go diverse: general imbalanced datasets
- Try various loss functions
- Relate Minority Collapse to fairness

Future directions

Immediate connections:

- Go diverse: general imbalanced datasets
- Try various loss functions
- Relate Minority Collapse to fairness



More broadly:

- Multiple Layer-Peeled Model:

$$
\begin{aligned}
\min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{f}(\boldsymbol{h}_{k,i},\boldsymbol{W}_{(L-m+1):L}),\boldsymbol{y}_k) \\
\text{s.t.} \quad & \frac{1}{K}\left\|\boldsymbol{W}_{(L-m+1):L}\right\|^2 \le E_W \\
& \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\left\|\boldsymbol{h}_{k,i}\right\|^2 \le E_H
\end{aligned}
$$

- Model the training dynamics and test performance
- Why does the ansatz yield reasonable prediction?

Take-home messages

Layer-Peeled Model = minimal integration of

prediction $(\boldsymbol{W})$ + *representation* $(\boldsymbol{H})$

- Nonconvex but analytical
- Explain neural collapse
- Predict Minority Collapse
- Practical insights into deep learning

Reference

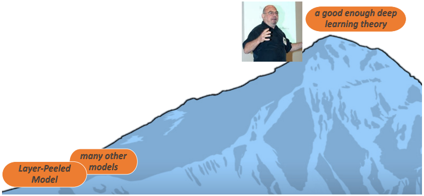
Exploring Deep Neural Networks via Layer-Peeled Model: Minority Collapse in Imbalanced Training
with Cong Fang, Hangfeng He, Qi Long. *Proceedings of the National Academy of Sciences (PNAS)*, 2021

- Code: `https://github.com/HornHehhf/LPM`
- NSF CAREER and TRIPODS, and Sloan

Take-home messages

Layer-Peeled Model = minimal integration of

prediction $(\boldsymbol{W})$ + *representation* $(\boldsymbol{H})$



- Nonconvex but analytical
- Explain neural collapse
- Predict Minority Collapse
- Practical insights into deep learning

Reference

Exploring Deep Neural Networks via Layer-Peeled Model: Minority Collapse in Imbalanced Training
with Cong Fang, Hangfeng He, Qi Long. *Proceedings of the National Academy of Sciences (PNAS)*, 2021

- Code: `https://github.com/HornHehhf/LPM`
- NSF CAREER and TRIPODS, and Sloan