What Should Good Deep Learning Models Look Like? An Optimization Perspective

Weijie Su

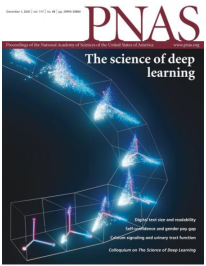
University of Pennsylvania

Jason's Optimization Seminar, Penn, February 8, 2024

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

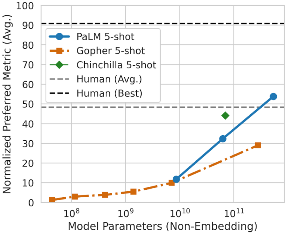
- Collect data and buy GPU first
- Scale model with data and computational resources
- End to end: Representation, computation, prediction

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 whi 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 lever their human knowledge of the domain, but the only thing that matters in the long run is the leveraging computation. These two need not run counter to each other, but in practice they tend to. Time spent on is time not spent on the other. There are psychological commitments to investment in one approach or t 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...

- 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...

- 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?

Yet another bitter lesson

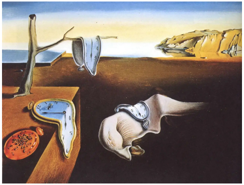
Very difficult to build a mathematical foundation for deep learning...

- Highly incomplete: Kawaguchi'16, Arora et al.'19, Jacot et al.'18, Allen-Zhu et al.'18, Du et al.'19, Mei et al.'19,...
- This talk doesn't attempt to address these fundamental questions
- Instead, we attempt to make deep learning (a bit more) geometrical

When is it easier to *geometrize* deep learning?

Terminal phase of training

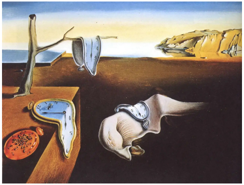
Training toward interpolating in-sample data, beyond zero classification error (Papyan et al.'20)



When is it easier to *geometrize* deep learning?

Terminal phase of training

Training toward interpolating in-sample data, beyond zero classification error (Papyan et al.'20)

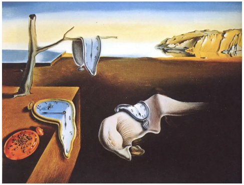


- Better generalization
- Improvement in adversarial robustness

When is it easier to *geometrize* deep learning?

Terminal phase of training

Training toward interpolating in-sample data, beyond zero classification error (Papyan et al.'20)



- Better generalization
- Improvement in adversarial robustness

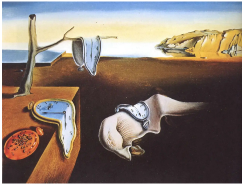
Easier to geometrize neural networks at terminal phase of training

- The training dynamics is chaotic

When is it easier to *geometrize* deep learning?

Terminal phase of training

Training toward interpolating in-sample data, beyond zero classification error (Papyan et al.'20)



- Better generalization
- Improvement in adversarial robustness

Easier to geometrize neural networks at terminal phase of training

- The training dynamics is chaotic
- But, a well-trained neural network is a solution to some optimization problem

This talk

1. A small surrogate model
   - Analyze the last-layer weights and features of well-trained neural networks

This talk

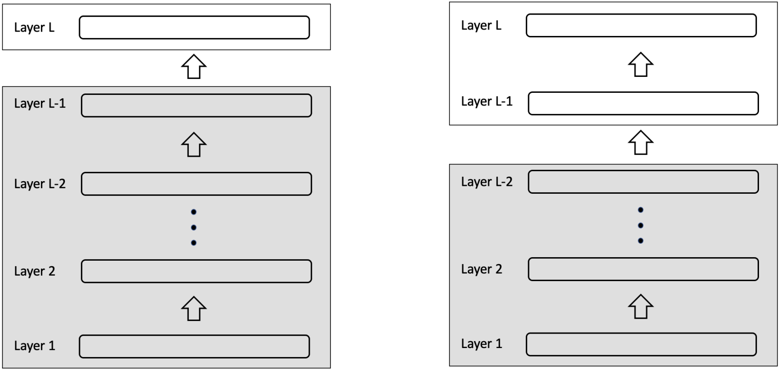
1. A small surrogate model
   - Analyze the last-layer weights and features of well-trained neural networks
2. A simple geometric law
   - Describe how data are separated through layers in well-trained neural networks

Part I: A Layer-Peeled Model

Collaborators

- Cong Fang (Penn→Peking University)
- Hangfeng He (Penn→University of Rochester)
- Qi Long (Penn)

Illustration of our approach



(a) 1-Layer-Peeled Model

(b) 2-Layer-Peeled Model

Setup for deep learning

Neural network for K-class classification:

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

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

Setup for deep learning

Neural network for K-class classification:

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

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

Optimization problem:

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

- $\boldsymbol{y}_k$ 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

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

$$\min_{W_{\mathsf{full}}} \quad \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(f(x_{k,i}; W_{\mathsf{full}}), y_k) + \frac{\lambda}{2}\|W_{\mathsf{full}}\|^2$$

- Difficult to pinpoint how any layer W_l influences the output

A peek at Layer-Peeled Model

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

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

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

A peek at Layer-Peeled Model

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

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

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

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
- Nonconvex but analytically tractable

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 $\boldsymbol{W}^\star \equiv [\boldsymbol{w}_1^\star, \ldots, \boldsymbol{w}_K^\star]^\top, \boldsymbol{H}^\star \equiv [\boldsymbol{h}_{k,i}^\star : 1 \leqslant k \leqslant K, 1 \leqslant i \leqslant n]$ *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!
- $C = \sqrt{E_H/E_W}, C' = \sqrt{E_H}$
- 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?

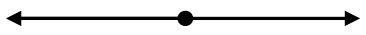
K-simplex ETF

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

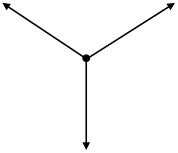
Equivalently, random variables $\xi_1, \ldots, \xi_K$ of mean 0 and variance 1. If $\mathbb{E}\xi_i\xi_j = \rho$ for all $i \neq j$, what's the min of ρ?

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

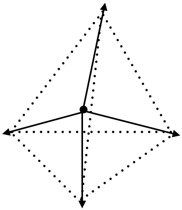
$K = 2$



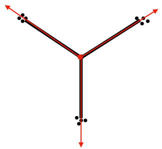
$K = 3$



$K = 4$



This is simply neural collapse



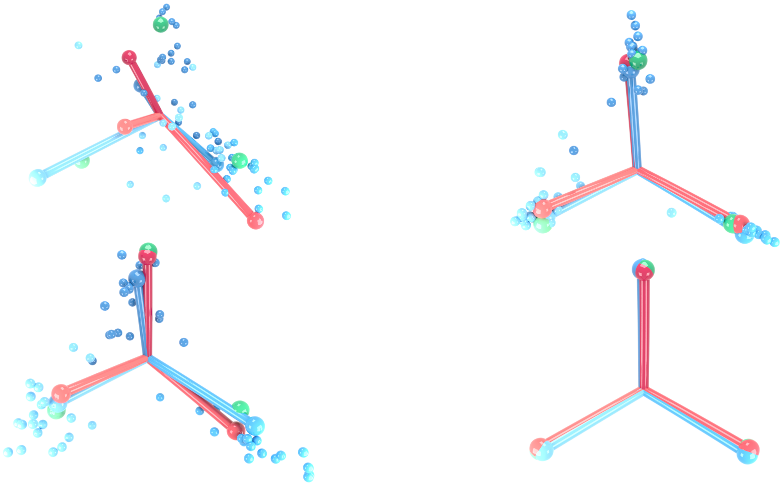
Papyan, Han, and Donoho discovered *neural collapse* in 2020:

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

[Mixon et al.'20, E and Wojtowytsch'20, Lu and Steinerberger'20, Zhu et al.'21] justified neural collapse using different models

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:

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

Proposition

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*

What happens without the ansatz?

Without the ansatz:

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

Proposition

*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 it possible to directly justify the ansatz?

Can the Layer-Peeled Model predict something?

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

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

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

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

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
-

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

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
-

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

-

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

Convex relaxation

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

Convex relaxation

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

- 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. Then the minimizers of the Layer-Peeled Model can be derived from the minimizer of the convex relaxation, up to a rotation*

Nonconvex optimization via convex optimization

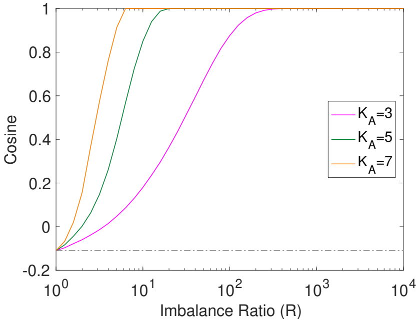
Lemma

Assume $p \geq 2K$ and $\mathcal{L}$ is convex in its first argument. Then the minimizers of the Layer-Peeled Model can be derived from the minimizer of the convex relaxation, up to a rotation

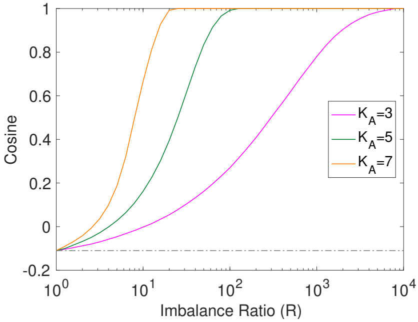
- No loss of information when we study the Layer-Peeled Model through a convex program
- But class means *no longer* collapse to classifiers

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*

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

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

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 (Chen 2023)

Let $(\boldsymbol{H}^\star, \boldsymbol{W}^\star)$ be any global minimizer of the Layer-Peeled Model. When $R \geqslant R^*$, we have

$$\boldsymbol{w}_k^\star = \boldsymbol{w}_{k'}^\star, \text{ for all } K_A < k < k' \leqslant K$$

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

Illustration of Minority Collapse

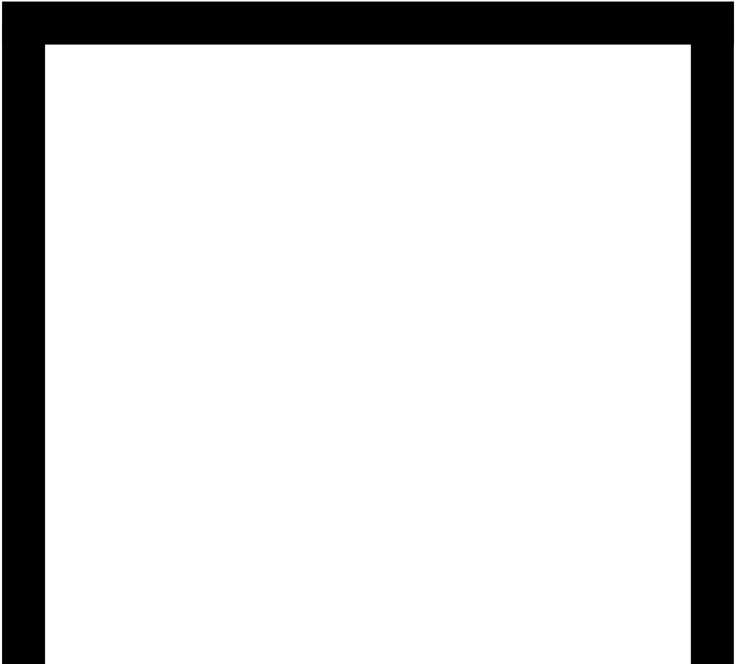
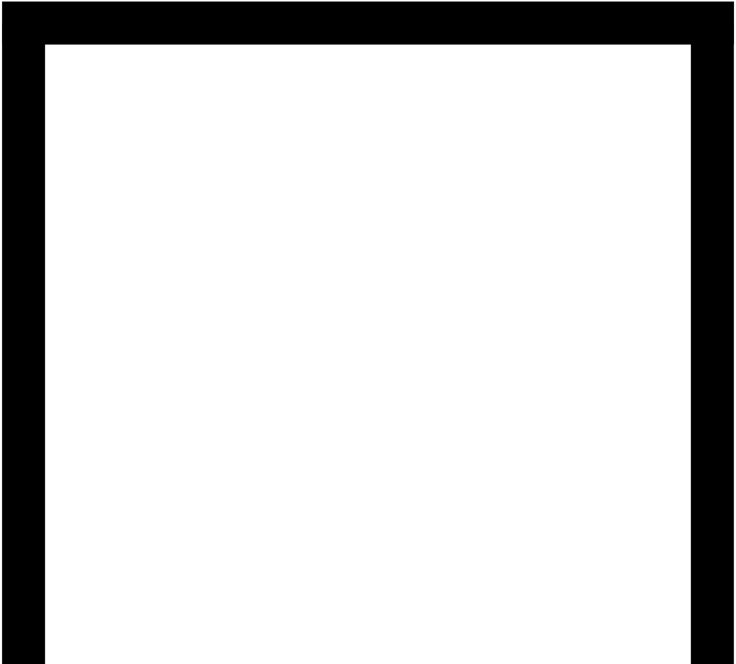
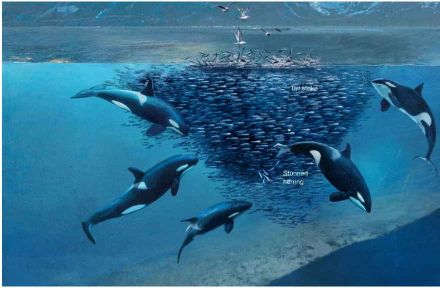


Illustration of Minority Collapse



Intuition for Minority Collapse

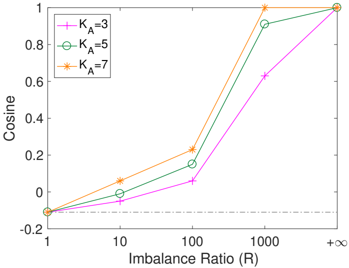
$$
\begin{aligned}
\min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k) \\
\text{s.t.} \quad & \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W \\
& \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2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



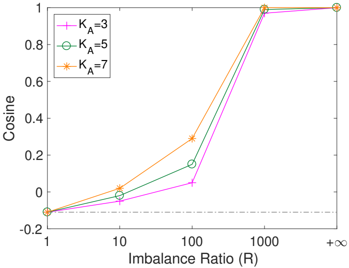
Competition for space!

Is Minority Collapse a real thing?

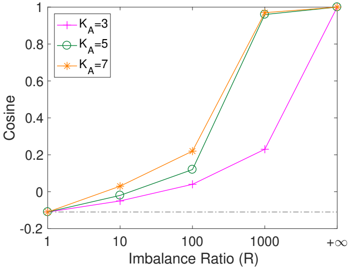
Minority Collapse in experiments



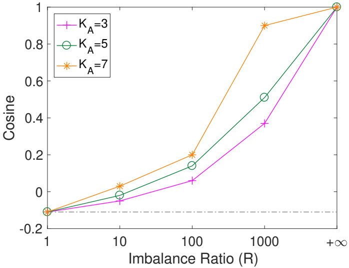
(e) VGG11 on FashionMNIST



(f) VGG13 on CIFAR10



(g) ResNet18 on FashionMNIST

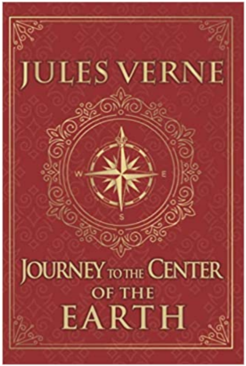


(h) ResNet18 on CIFAR10

Part II: A Law of Data Separation

Let's dig into it

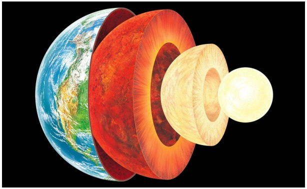
Does neural collapse extend to interior layers?

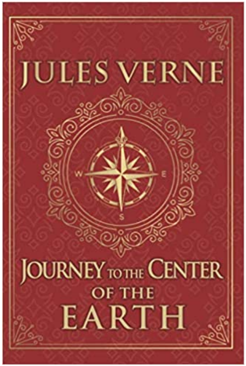


Let's dig into it

Does neural collapse extend to interior layers?

- Unfortunately, no
- Too many nonlinearities, plus high degrees of non-uniqueness

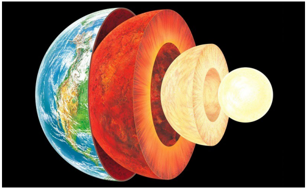


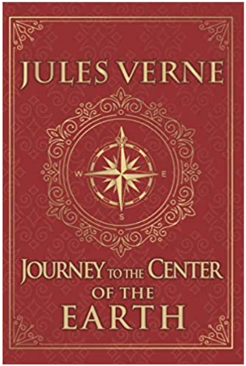


Let's dig into it

Does neural collapse extend to interior layers?

- Unfortunately, no
- Too many nonlinearities, plus high degrees of non-uniqueness
- Any other patterns?





Collaborator

- Hangfeng He (Penn$\to$University of Rochester)

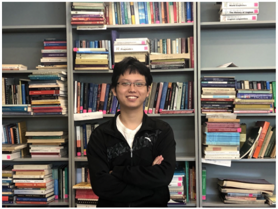
Collaborator

- Hangfeng He (Penn→University of Rochester)

Hangfeng He

Home Research Teaching

I am an Assistant Professor in the Department of Computer Science and the Goergen Institute for Data Science at the University of Rochester. Before this, I was a Ph.D. student at the University of Pennsylvania, where I worked with Dan Roth and Weijie Su. Before that, I received my bachelor's degree from Peking University in 2017.



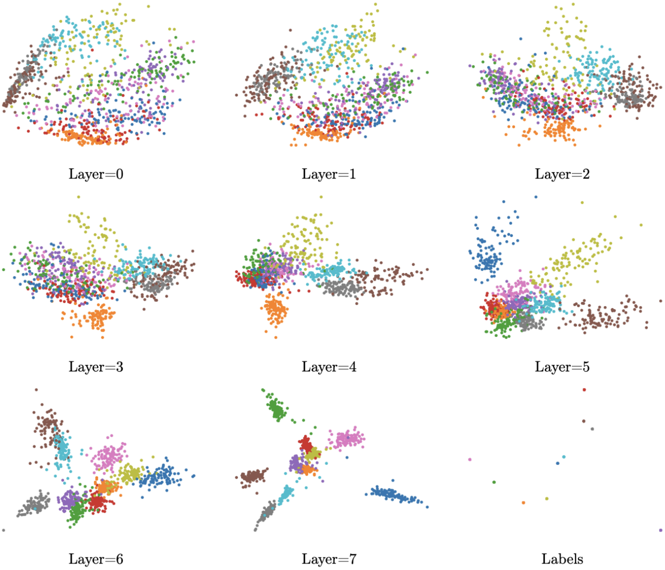
My research interests include machine learning and natural language processing, with a focus on incidental supervision for natural language understanding, interpretability of deep neural networks, and reasoning in natural language.

[Google Scholar] [CV]

Contact

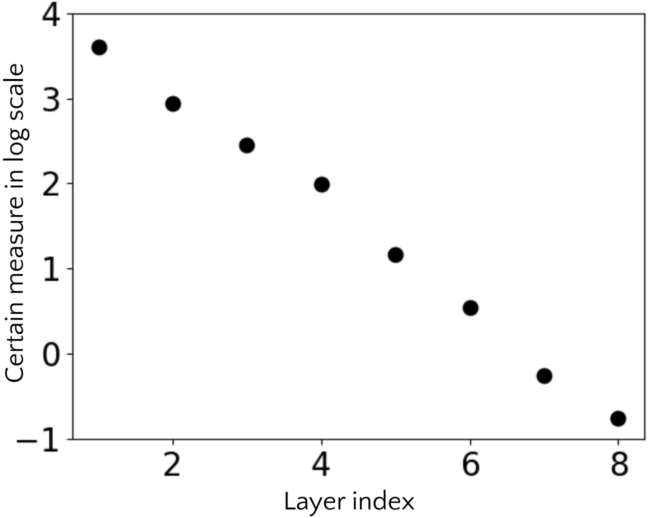
Office: 3009 Wegmans Hall, 250 Hutchison Rd, Rochester, NY 14620
Email: hangfeng.he@rochester.edu

Chaotic patterns



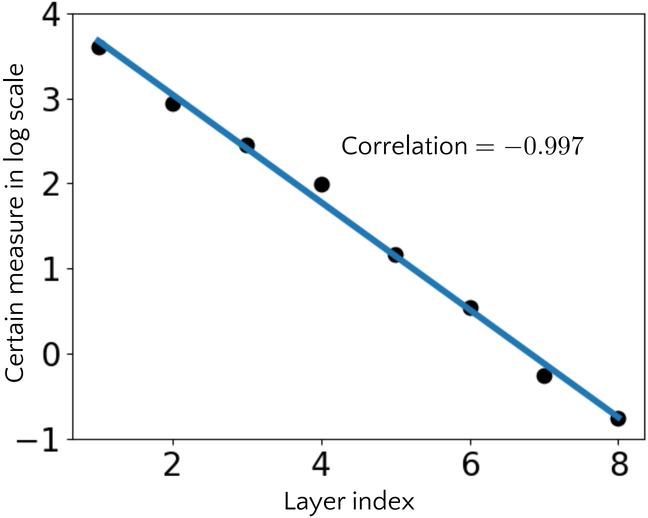
"Big" symmetries are gone. How about "small" symmetries?

A numerical surprise: equi-separation



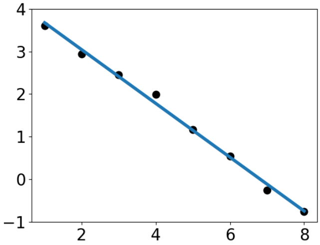
8-layer feedforward network trained on FashinMNIST using Adam

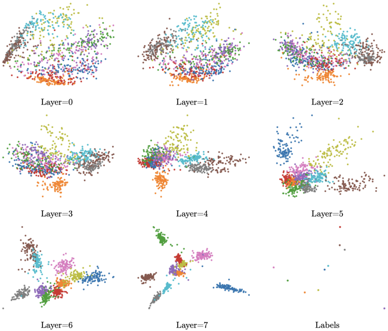
A numerical surprise



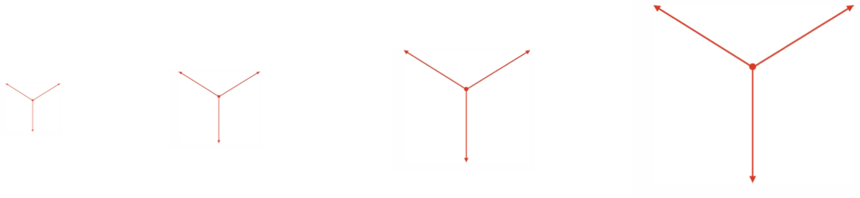
8-layer feedforward network trained on FashinMNIST using Adam

A sharp comparison



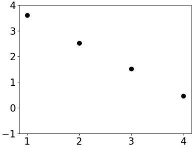


This is NOT the reality

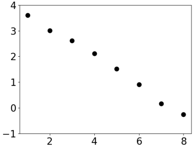


This is the reality

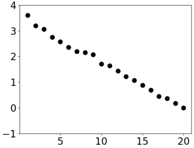
More experimental results



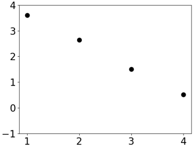
(a) SGD-4



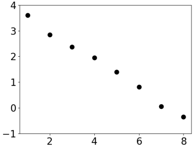
(b) SGD-8



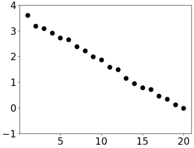
(c) SGD-20



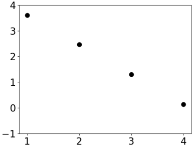
(d) SGD+Momentum-4



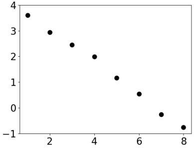
(e) SGD+Momentum-8



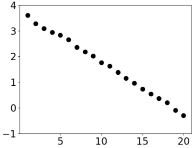
(f) SGD+Momentum-20



(g) Adam-4

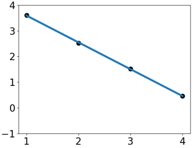


(h) Adam-8

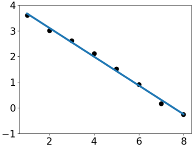


(i) Adam-20

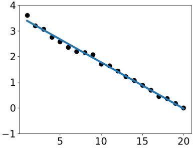
More experimental results



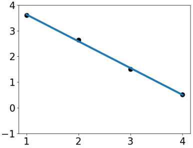
(a) SGD-4



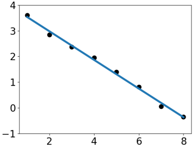
(b) SGD-8



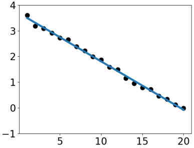
(c) SGD-20



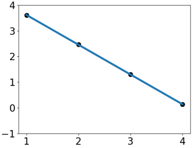
(d) SGD+Momentum-4



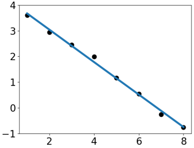
(e) SGD+Momentum-8



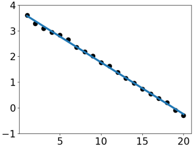
(f) SGD+Momentum-20



(g) Adam-4



(h) Adam-8



(i) Adam-20

Separation fuzziness

$\bar{x}_k := (x_{k1} + \cdots + x_{kn_k})/n_k$: sample mean of Class k
$\bar{x} := (n_1\bar{x}_1 + \cdots + n_K\bar{x}_K)/n$: global mean ($n := n_1 + \cdots + n_K$)

Separation fuzziness

$\bar{x}_k := (x_{k1} + \cdots + x_{kn_k})/n_k$: sample mean of Class k
$\bar{x} := (n_1\bar{x}_1 + \cdots + n_K\bar{x}_K)/n$: global mean ($n := n_1 + \cdots + n_K$)

Sum of squares between (*signal*)

$$\mathrm{SSB} := \frac{1}{n}\sum_{k=1}^{K} n_k(\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^\top$$

Sum of squares within (*noise*)

$$\mathrm{SSW} := \frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n_k}(x_{ki} - \bar{x}_k)(x_{ki} - \bar{x}_k)^\top$$

Separation fuzziness

$\bar{x}_k := (x_{k1} + \cdots + x_{kn_k})/n_k$: sample mean of Class k
$\bar{x} := (n_1\bar{x}_1 + \cdots + n_K\bar{x}_K)/n$: global mean ($n := n_1 + \cdots + n_K$)

Sum of squares between (*signal*)

$$\mathrm{SSB} := \frac{1}{n}\sum_{k=1}^{K} n_k(\bar{x}_k - \bar{x})(\bar{x}_k - \bar{x})^\top$$

Sum of squares within (*noise*)

$$\mathrm{SSW} := \frac{1}{n}\sum_{k=1}^{K}\sum_{i=1}^{n_k}(x_{ki} - \bar{x}_k)(x_{ki} - \bar{x}_k)^\top$$

Measure of how well data are separated

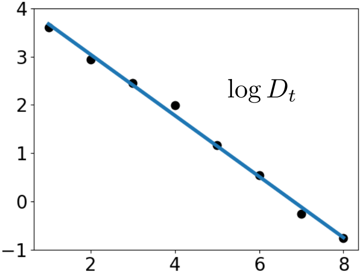
$$D := \mathrm{Tr}(\mathrm{SSW}\,\mathrm{SSB}^+)$$

- SSB^+ is the Moore–Penrose inverse of the matrix SSB
- Inverse signal-to-noise ratio (Papyan et al.'20)
- Weighted projection of noise onto $(K-1)$-D space spanned by SSB. Thus no need to normalize D by the dimension

It’s well separated

An (empirical) law of deep learning

D_t: separation measure for data before passing through the t^{th} layer



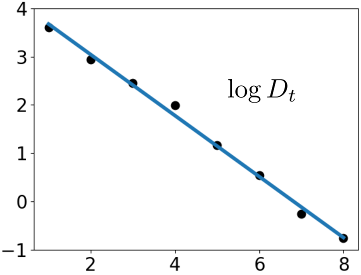
The law of equi-separation

For $1 \leqslant t \leqslant m$ and some $0 < \rho < 1$:

$$D_t \approx c\rho^t$$

An (empirical) law of deep learning

D_t: separation measure for data before passing through the t^{th} layer



The law of equi-separation

For $1 \leqslant t \leqslant m$ and some $0 < \rho < 1$:

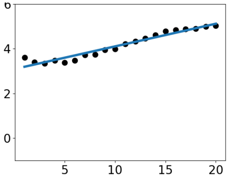
$$D_t \approx c\rho^t$$

- Nonlinearity is crucial
- Equivalently,

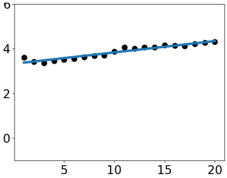
$$\log D_{t+1} - \log D_t \approx -\log \frac{1}{\rho}$$

- $\rho = 0.53$ above. So half-life: $t_{\frac{1}{2}} = \dfrac{\log 2}{\log \rho^{-1}} = 1.1$

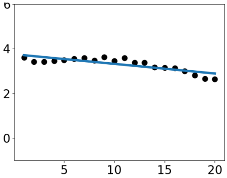
When does it emerge?



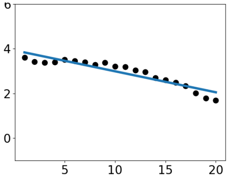
(a) Epoch-0



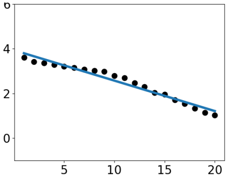
(b) Epoch=10



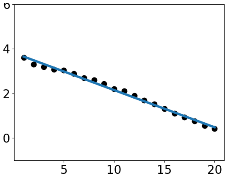
(c) Epoch=20



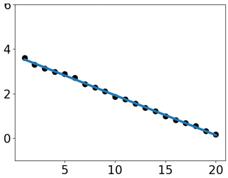
(d) Epoch-30



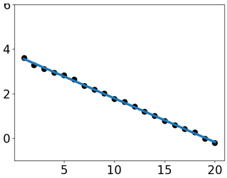
(e) Epoch=50



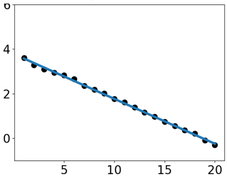
(f) Epoch=100



(g) Epoch-200

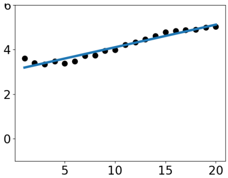


(h) Epoch=300

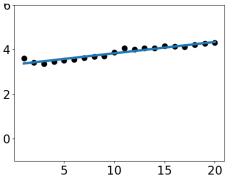


(i) Epoch=600

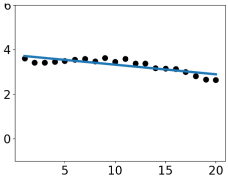
When does it emerge? Earlier than neural collapse



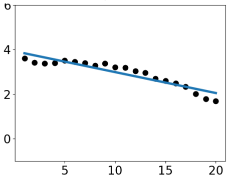
(a) Epoch-0



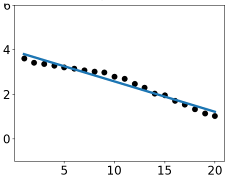
(b) Epoch=10



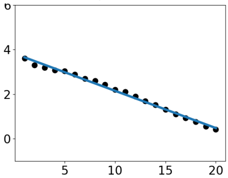
(c) Epoch=20



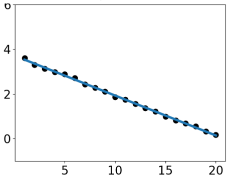
(d) Epoch-30



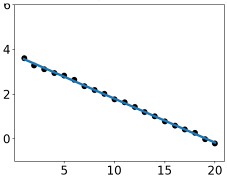
(e) Epoch=50



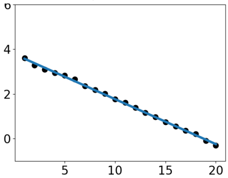
(f) Epoch=100



(g) Epoch-200

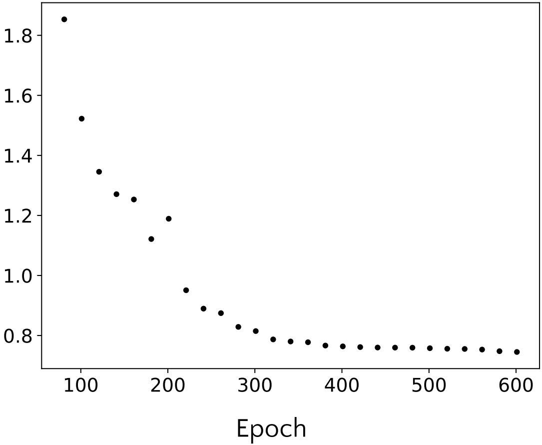


(h) Epoch=300



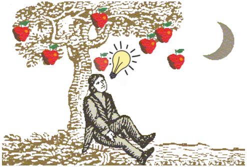
(i) Epoch=600

Earlier than neural collapse

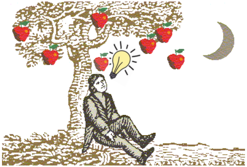


Separation fuzziness of last-layer features

Ask me anything about this law

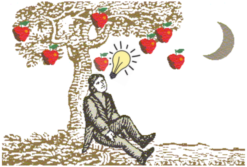


Ask me anything about this law



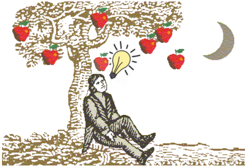
Is this law pervasive?

Ask me anything about this law



Is this law pervasive? Yes

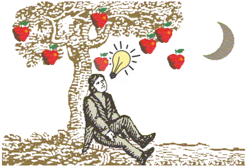
Ask me anything about this law



Is this law pervasive? Yes

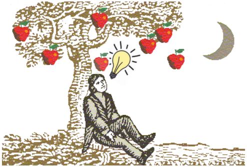
Does this law provide insights into the practice of deep learning?

Ask me anything about this law



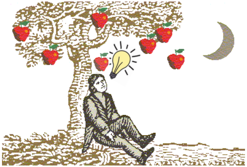
Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes

Ask me anything about this law



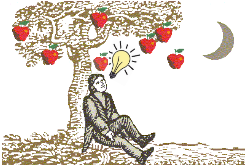
Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes
Any intuition about why this law appears?	

Ask me anything about this law



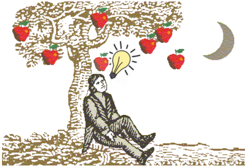
Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes
Any intuition about why this law appears?	I think so

Ask me anything about this law



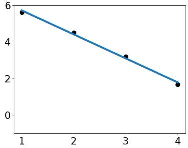
Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes
Any intuition about why this law appears?	I think so
Can we prove this law?	

Ask me anything about this law

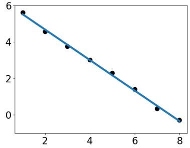


Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes
Any intuition about why this law appears?	I think so
Can we prove this law?	Not yet

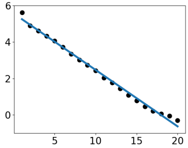
Data, imbalance, and learning rate



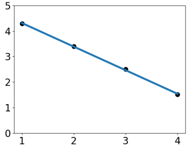
(a) CIFAR10-4



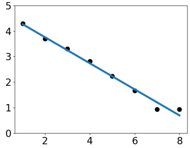
(b) CIFAR10-8



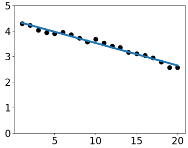
(c) CIFAR10-20



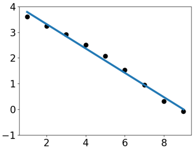
(d) Imbalance-4



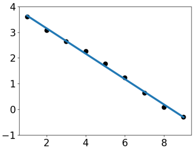
(e) Imbalance-8



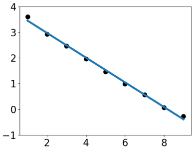
(f) Imbalance-20



(g) Learning rate: 0.01

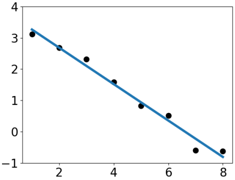


(h) Learning rate: 0.03

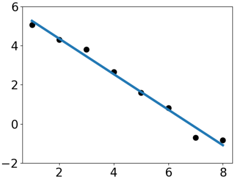


(i) Learning rate: 0.1

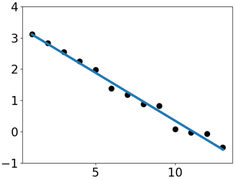
Architecture



(a) AlexNetX-FMNIST



(b) AlexNetX-CIFAR10



(c) VGG13X-FMNIST

Guidelines and insights from the law of equi-separation

The trilogy of the deep learning practice

- Network architecture
- Training
- Interpretation

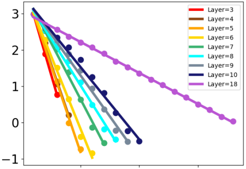
Dependence on the depth

$D_m \approx c\rho^m$: deep learning is necessarily to be deep

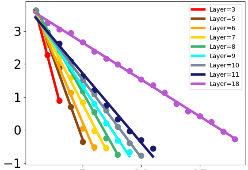
Dependence on the depth

$D_m \approx c\rho^m$: deep learning is necessarily to be deep

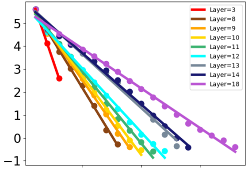
However, a complete story is slightly different



(a) MNIST



(b) FashionMNIST

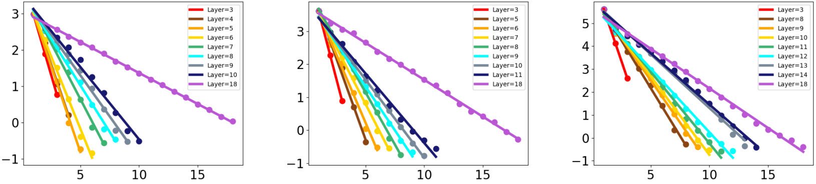


(c) CIFAR10

Dependence on the depth

$D_m \approx c\rho^m$: deep learning is necessarily to be deep

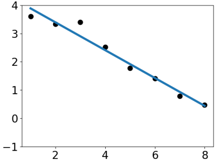
However, a complete story is slightly different



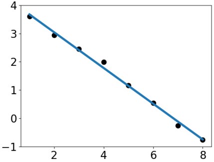
(a) MNIST (b) FashionMNIST (c) CIFAR10

- The choice of depth should consider the complexity of the applications
- Prior literature does not take the data-separation perspective (Srivastava et al.'15)

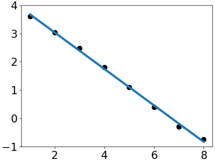
Data-separation perspective on width and shape



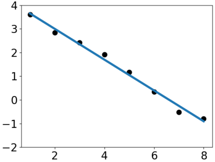
(a) Width: 20



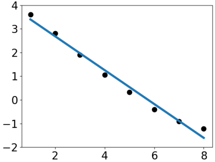
(b) Width: 100



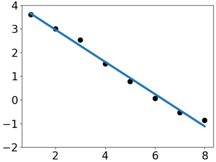
(c) Width: 1000



(d) Shape: narrow-wide

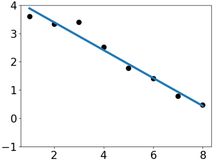


(e) Shape: wide-narrow

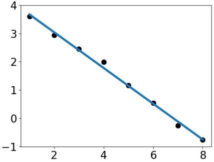


(f) Shape: mix

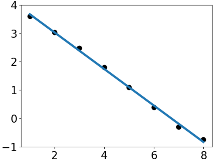
Data-separation perspective on width and shape



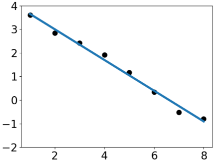
(a) Width: 20



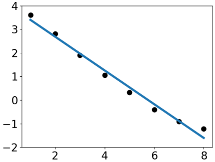
(b) Width: 100



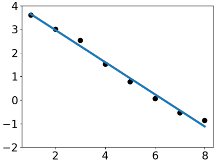
(c) Width: 1000



(d) Shape: narrow-wide



(e) Shape: wide-narrow



(f) Shape: mix

- Very wide neural networks should not be recommended (Tan and Le’19)
- Look vertically rather than horizontally when judging a network

Equi-separation implies robustness

Overall separation ability $R := \frac{D_m}{D_1} = \frac{D_m}{D_{m-1}} \times \frac{D_{m-1}}{D_{m-2}} \times \cdots \times \frac{D_2}{D_1}$

Equi–separation implies robustness

Overall separation ability $R := \frac{D_m}{D_1} = \frac{D_m}{D_{m-1}} \times \frac{D_{m-1}}{D_{m-2}} \times \cdots \times \frac{D_2}{D_1}$

Perturb each layer:

$$\left(\frac{D_m}{D_{m-1}} + \varepsilon\right)\left(\frac{D_{m-1}}{D_{m-2}} + \varepsilon\right)\cdots\left(\frac{D_2}{D_1} + \varepsilon\right)$$
$$= R + R\left(\frac{D_{m-1}}{D_m} + \frac{D_{m-2}}{D_{m-1}} + \cdots + \frac{D_1}{D_2}\right)\varepsilon + O(\varepsilon^2)$$

Equi–separation implies robustness

Overall separation ability $R := \frac{D_m}{D_1} = \frac{D_m}{D_{m-1}} \times \frac{D_{m-1}}{D_{m-2}} \times \cdots \times \frac{D_2}{D_1}$

Perturb each layer:

$$\left(\frac{D_m}{D_{m-1}} + \varepsilon\right)\left(\frac{D_{m-1}}{D_{m-2}} + \varepsilon\right)\cdots\left(\frac{D_2}{D_1} + \varepsilon\right)$$
$$= R + R\left(\frac{D_{m-1}}{D_m} + \frac{D_{m-2}}{D_{m-1}} + \cdots + \frac{D_1}{D_2}\right)\varepsilon + O(\varepsilon^2)$$

The perturbation $R\left(\frac{D_{m-1}}{D_m} + \frac{D_{m-2}}{D_{m-1}} + \cdots + \frac{D_1}{D_2}\right)\varepsilon$ is minimized in absolute value when

$$\frac{D_m}{D_{m-1}} = \frac{D_{m-1}}{D_{m-2}} = \cdots = \frac{D_2}{D_1}$$

Equi-separation implies robustness

Overall separation ability $R := \frac{D_m}{D_1} = \frac{D_m}{D_{m-1}} \times \frac{D_{m-1}}{D_{m-2}} \times \cdots \times \frac{D_2}{D_1}$

Perturb each layer:

$$\left(\frac{D_m}{D_{m-1}} + \varepsilon\right)\left(\frac{D_{m-1}}{D_{m-2}} + \varepsilon\right)\cdots\left(\frac{D_2}{D_1} + \varepsilon\right) = R + R\left(\frac{D_{m-1}}{D_m} + \frac{D_{m-2}}{D_{m-1}} + \cdots + \frac{D_1}{D_2}\right)\varepsilon + O(\varepsilon^2)$$

The perturbation $R\left(\frac{D_{m-1}}{D_m} + \frac{D_{m-2}}{D_{m-1}} + \cdots + \frac{D_1}{D_2}\right)\varepsilon$ is minimized in absolute value when

$$\frac{D_m}{D_{m-1}} = \frac{D_{m-1}}{D_{m-2}} = \cdots = \frac{D_2}{D_1}$$

- Train at least until the law comes into effect

Equi-separation implies robustness

Overall separation ability $R := \frac{D_m}{D_1} = \frac{D_m}{D_{m-1}} \times \frac{D_{m-1}}{D_{m-2}} \times \cdots \times \frac{D_2}{D_1}$

Perturb each layer:

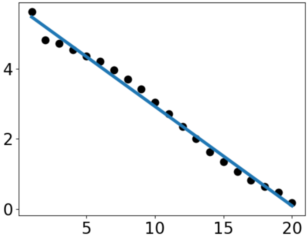
$$\left(\frac{D_m}{D_{m-1}} + \varepsilon\right)\left(\frac{D_{m-1}}{D_{m-2}} + \varepsilon\right)\cdots\left(\frac{D_2}{D_1} + \varepsilon\right)$$
$$= R + R\left(\frac{D_{m-1}}{D_m} + \frac{D_{m-2}}{D_{m-1}} + \cdots + \frac{D_1}{D_2}\right)\varepsilon + O(\varepsilon^2)$$

The perturbation $R\left(\frac{D_{m-1}}{D_m} + \frac{D_{m-2}}{D_{m-1}} + \cdots + \frac{D_1}{D_2}\right)\varepsilon$ is minimized in absolute value when

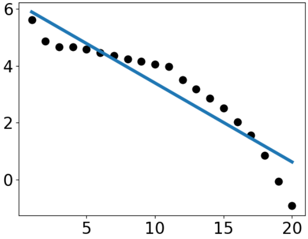
$$\frac{D_m}{D_{m-1}} = \frac{D_{m-1}}{D_{m-2}} = \cdots = \frac{D_2}{D_1}$$

- Train at least until the law comes into effect
- An analog: if Wakanda wants to double GDP in 10 years, the most robust way is to fix annual growth rate at $2^{\frac{1}{10}} - 1 = 7.2\%$

Equi-separation implies better generalization



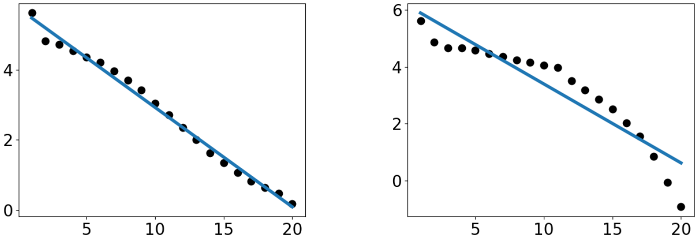
(a) Unfrozen



(b) Frozen

- Frozen training: bottom/top 10 layers are trained while the others are fixed
- Have about the same final separation measure and training loss

Equi-separation implies better generalization

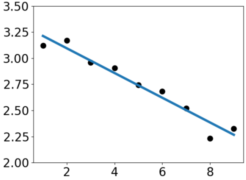


(a) Unfrozen (b) Frozen

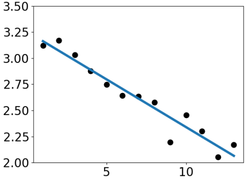
- Frozen training: bottom/top 10 layers are trained while the others are fixed
- Have about the same final separation measure and training loss
- Test accuracy:
    - Unfrozen: 21.46%
    - Frozen: 18.25%

Interpretation from data-separation perspective

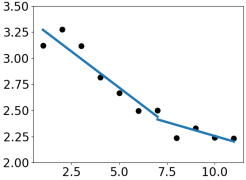
What are the basic operational modules in ResNet?



(a) 2 layers in a block



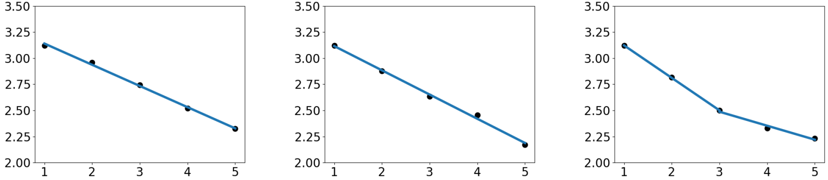
(b) 3 layers in a block



(c) Mix

Interpretation from data-separation perspective

What are the basic operational modules in ResNet?

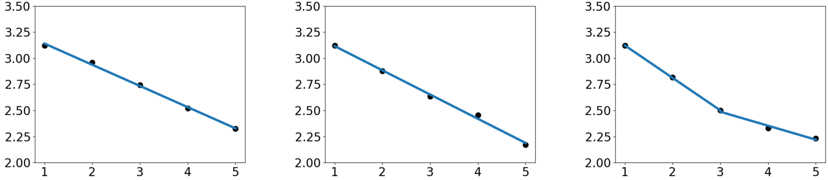


(a) 2 layers in a block (b) 3 layers in a block (c) Mix

- The right module is block for ResNet

Interpretation from data-separation perspective

What are the basic operational modules in ResNet?

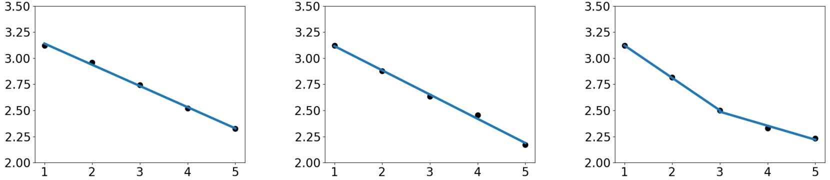


(a) 2 layers in a block (b) 3 layers in a block (c) Mix

- The right module is block for ResNet
- All layers/modules are created equal

Interpretation from data-separation perspective

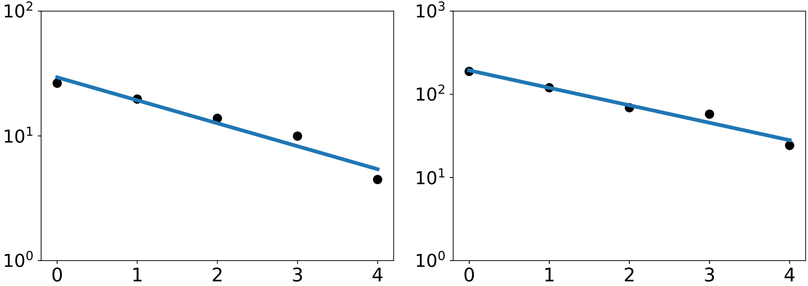
What are the basic operational modules in ResNet?



(a) 2 layers in a block (b) 3 layers in a block (c) Mix

- The right module is block for ResNet
- All layers/modules are created equal
- Need to take all layers collectively for interpretation, challenging layer-wise approaches (Zeiler and Fergus'14)

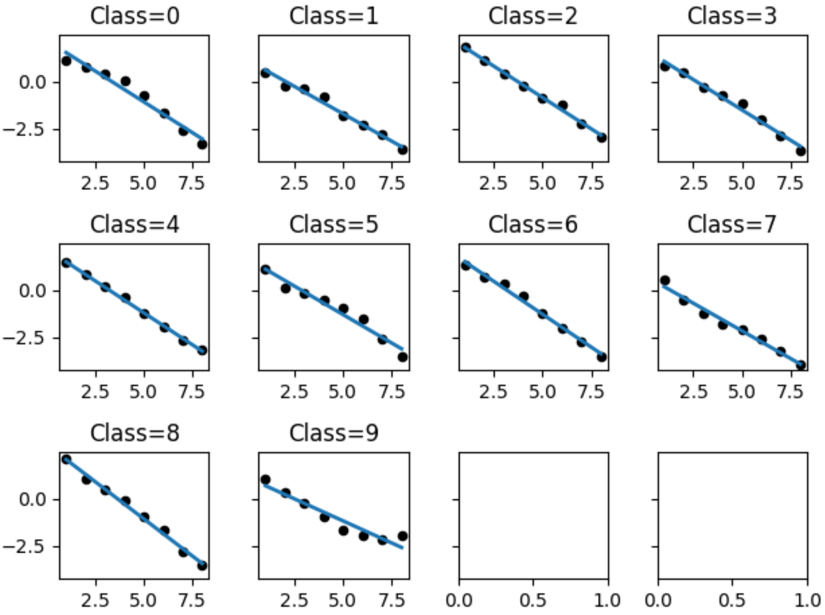
The same story for DenseNet (Gao et al.'19)



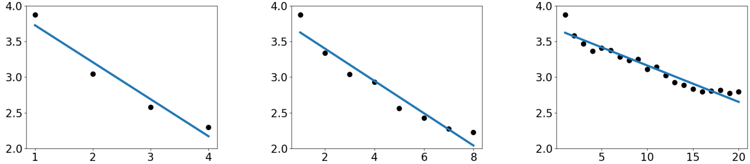
DenseNet161 by identifying a block as a module

The law from other angles

The law for each class

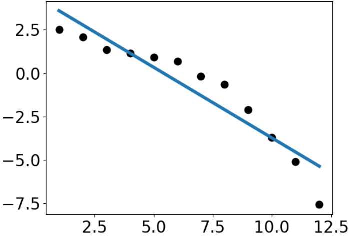


The equi-separation law in test

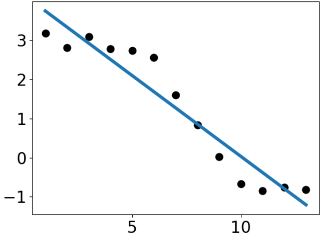


(a) Adam-4-Test (b) Adam-8-Test (c) Adam-20-Test

Language models?



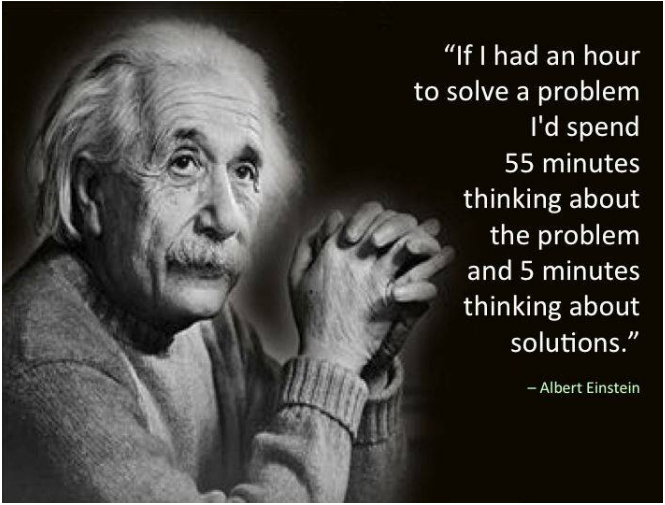
(a) BERT-CLS



(b) BERT-AVG

- Trained on a binary sentiment classification task (SST-2)
- Perhaps because it learns a sequence of token-level representations instead of sentence-level representations for each layer

Asking right questions about deep learning theory



Take-home messages

Layer-Peeled Model: Last-layer weights and features are free except for norm constraints

- Explain neural collapse
- Predict Minority Collapse

Equi-Separation Law: A data-separation perspective

- All layers/modules are created equal
- Guidelines and insights into architecture design, training, and interpretation

Reference

1. *Exploring Deep Neural Networks via Layer-Peeled Model: Minority Collapse in Imbalanced Training*
   with Cong Fang, Hangfeng He, and Qi Long
   Proceedings of the National Academy of Sciences (PNAS), 2021
2. *A Law of Data Separation in Deep Learning*
   with Hangfeng He
   Proceedings of the National Academy of Sciences (PNAS), 2023