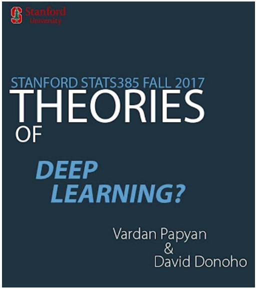
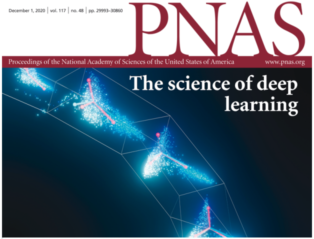
A Universal Law in Deep Learning
from MLP to Transformer, and Beyond

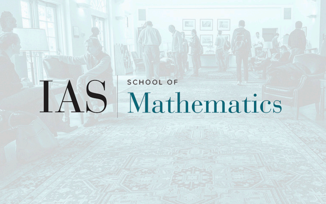
Weijie Su

University of Pennsylvania

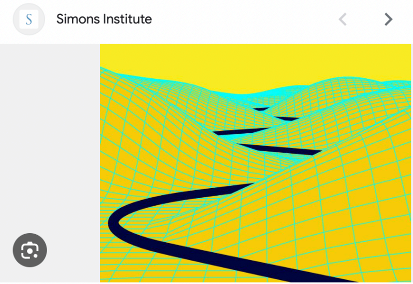
A new physics of the 21st century







Workshop on Theory of Deep Learning: Where next?



Summer Cluster: Deep Learning Theory

Wait, making deep learning a science requires...

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

Wait, making deep learning a science requires...

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

Wait, 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?

Wait, 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?

Wait, 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 (in addition to Sutton's)

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

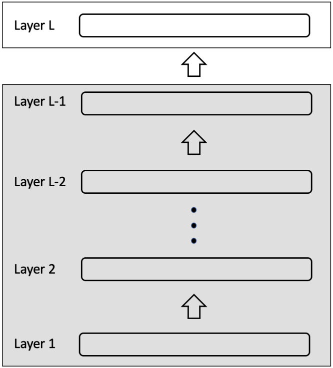
1. A small surrogate model
   - Analyze the last-layer weights and features of well-trained neural networks
2. A simple geometric law in MLP
   - Describe how data are separated through layers in well-trained neural networks
3. Extension of the law to Transformer and beyond
   - Describe how the next token is predicted across layers in Transformer

Part I: A Layer-Peeled Model

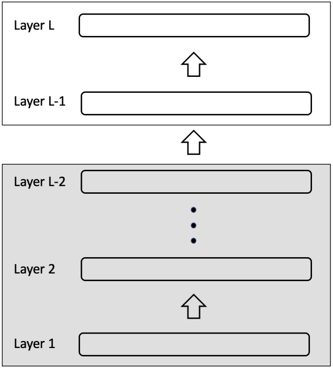
Collaborators

- Cong Fang (Penn$\rightarrow$Peking University)
- Hangfeng He (Penn$\rightarrow$University of Rochester)
- Qi Long (Penn)

Illustration of our approach (for MLP)



1-Layer-Peeled Model



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

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

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

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

- 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 (Papyan et al. 2020)

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

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)*

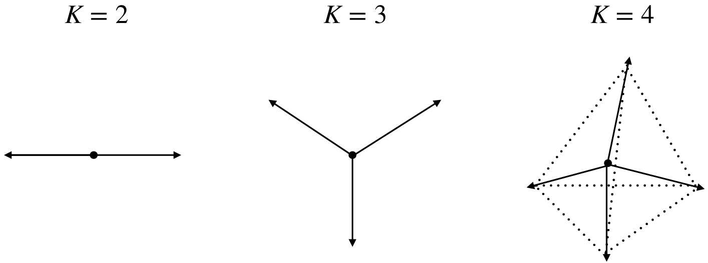
- $\boldsymbol{h}_{k,i}^\star$ depends only on the class membership!
- $C = \sqrt{E_H/E_W}, C' = \sqrt{E_H}$

K-simplex ETF

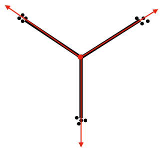
K equal-length vectors form the *largest* possible equal-sized angles between any pair

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?

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



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

Neural Collapse can justify the Layer-Peeled Model

About the ansatz

Recall

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

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:

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

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:

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

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*

- 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

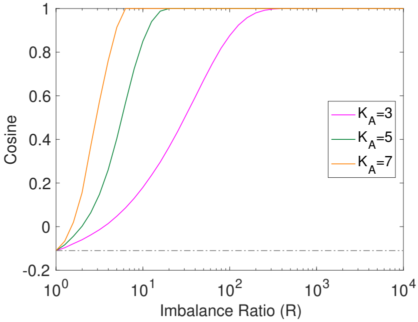
Convex relaxation

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

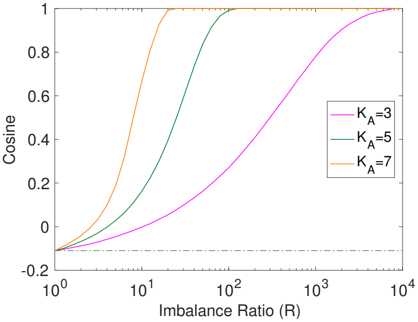
- Not a semidefinite program in the strict sense because a semidefinite program uses a linear objective function

A numerical surprise

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



$E_W = 1, E_H = 5$



$E_W = 1, E_H = 10$

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*

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

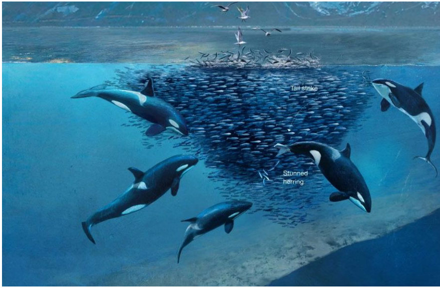
- The prediction on the minority classes becomes *completely at random*
- Fairness issue

Illustration of Minority Collapse

Illustration of Minority Collapse

Intuition for Minority Collapse

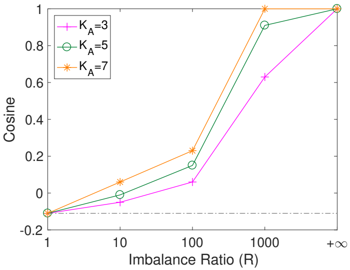
$$\begin{aligned} \min_{\boldsymbol{W},\boldsymbol{H}} \quad & \frac{1}{N}\sum_{k=1}^{K}\sum_{i=1}^{n_k}\mathcal{L}(\boldsymbol{W}\boldsymbol{h}_{k,i},\boldsymbol{y}_k) \\ \text{s.t.} \quad & \frac{1}{K}\sum_{k=1}^{K}\|\boldsymbol{w}_k\|^2 \le E_W \\ & \frac{1}{K}\sum_{k=1}^{K}\frac{1}{n_k}\sum_{i=1}^{n_k}\|\boldsymbol{h}_{k,i}\|^2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



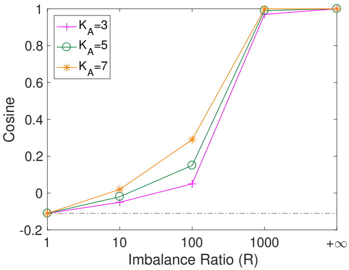
Competition for space!

Is Minority Collapse a real thing?

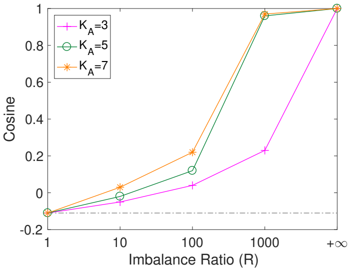
Minority Collapse in experiments



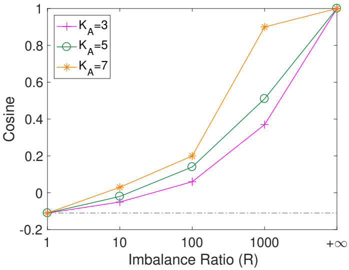
VGG11 on FashionMNIST



VGG13 on CIFAR10



ResNet18 on FashionMNIST

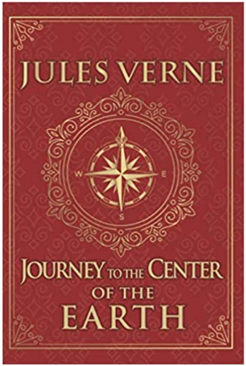


ResNet18 on CIFAR10

Part II: A Law of Data Separation

Let's dig into it

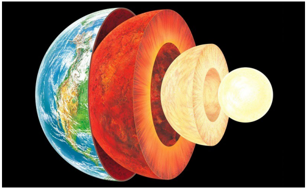
Does neural collapse extend to intermediate layers?

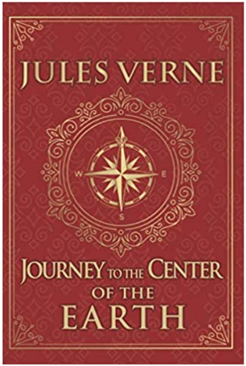


Let's dig into it

Does neural collapse extend to intermediate layers?

- Seems chaotic
- Too many nonlinearities, plus high degrees of non-uniqueness

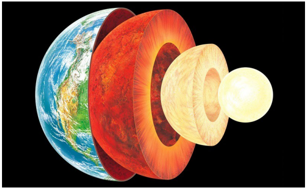


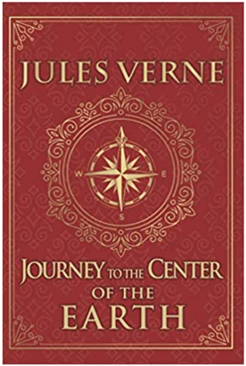


Let's dig into it

Does neural collapse extend to intermediate layers?

- Seems chaotic
- Too many nonlinearities, plus high degrees of non-uniqueness
- Any other patterns?





Collaborator

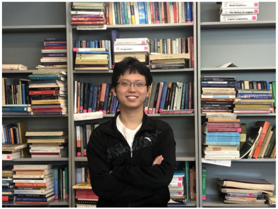
- Hangfeng He (Penn→University of Rochester)

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.



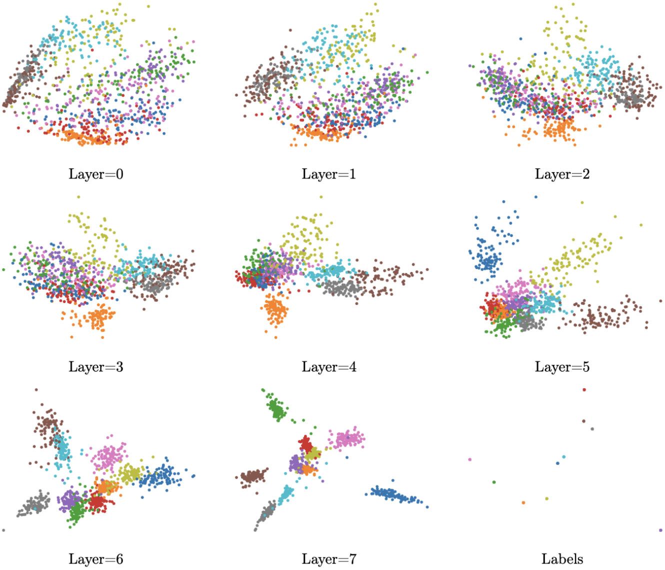
My research interests include machine learning and natural language processing, with a focus on incidental supervision for natural language understanding, interpretability of deep neural networks, and reasoning in natural language.

[Google Scholar] [CV]

Contact

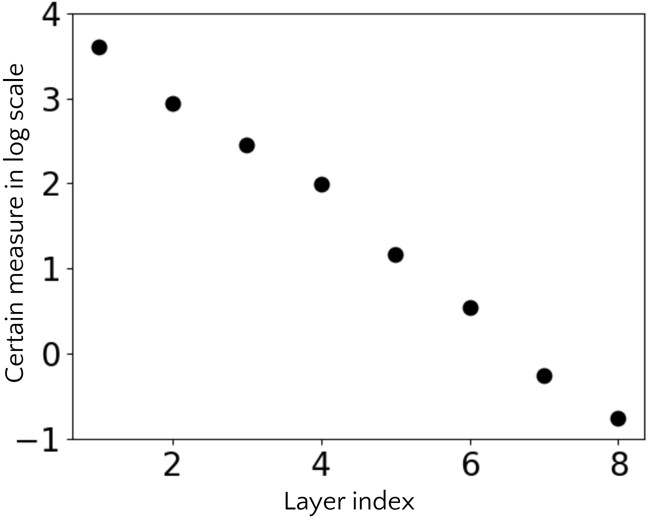
Office: 3009 Wegmans Hall, 250 Hutchison Rd, Rochester, NY 14620
Email: hangfeng.he@rochester.edu

Chaotic patterns



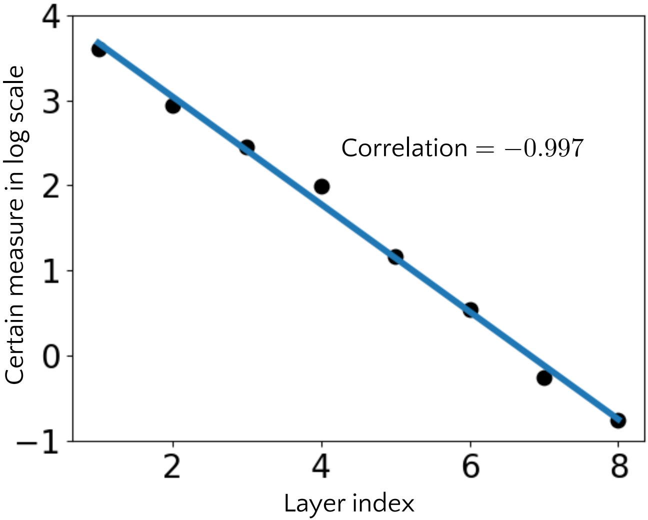
"Big" symmetries are gone. How about "small" symmetries?

A numerical surprise: equi-separation



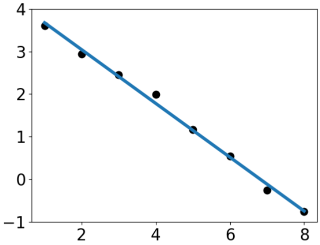
8-layer feedforward network trained on FashinMNIST using Adam

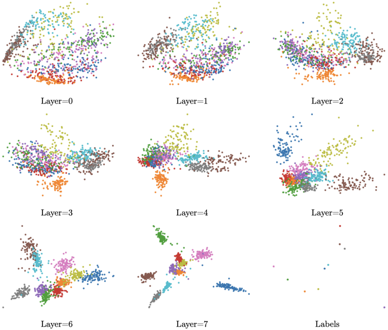
A numerical surprise



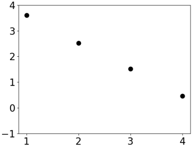
8-layer feedforward network trained on FashinMNIST using Adam

A sharp comparison

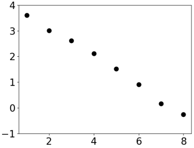




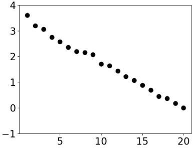
More experimental results



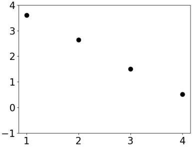
SGD–4



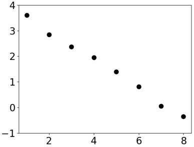
SGD–8



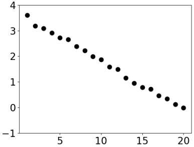
SGD–20



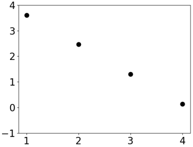
SGD+Momentum–4



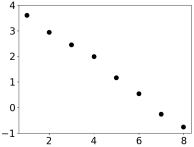
SGD+Momentum–8



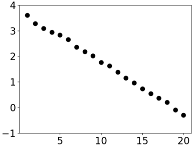
SGD+Momentum–20



Adam–4

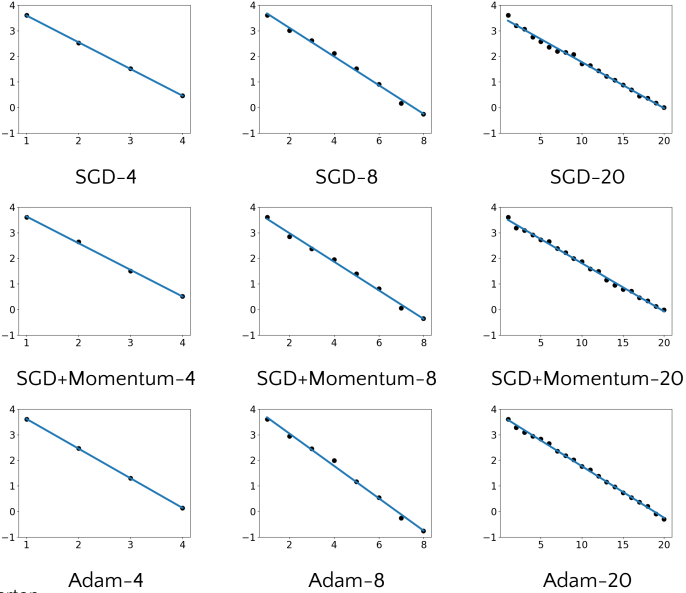


Adam–8



Adam–20

More experimental results



A canonical quantity in multivariate statistics

$\bar{\boldsymbol{x}}_k := (\boldsymbol{x}_{k,1} + \cdots + \boldsymbol{x}_{k,n_k})/n_k$: sample mean of Class k
$\bar{\boldsymbol{x}} := (n_1 \bar{\boldsymbol{x}}_1 + \cdots + n_K \bar{\boldsymbol{x}}_K)/n$: global mean ($n := n_1 + \cdots + n_K$)

A canonical quantity in multivariate statistics

$\bar{\boldsymbol{x}}_k := (\boldsymbol{x}_{k,1} + \cdots + \boldsymbol{x}_{k,n_k})/n_k$: sample mean of Class k
$\bar{\boldsymbol{x}} := (n_1\bar{\boldsymbol{x}}_1 + \cdots + n_K\bar{\boldsymbol{x}}_K)/n$: global mean ($n := n_1 + \cdots + n_K$)

Sum of squares between (*signal*)

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

Sum of squares within (*noise*)

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

A canonical quantity in multivariate statistics

$\bar{\boldsymbol{x}}_k := (\boldsymbol{x}_{k,1} + \cdots + \boldsymbol{x}_{k,n_k})/n_k$: sample mean of Class k
$\bar{\boldsymbol{x}} := (n_1\bar{\boldsymbol{x}}_1 + \cdots + n_K\bar{\boldsymbol{x}}_K)/n$: global mean ($n := n_1 + \cdots + n_K$)

Sum of squares between (*signal*)

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

Sum of squares within (*noise*)

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

Measure of how well data are separated

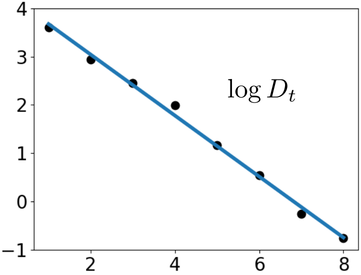
$$D := \mathrm{Tr}(\mathrm{SSW}\,\mathrm{SSB}^+)$$

- 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_l: separation measure for data before passing through the l^{th} layer



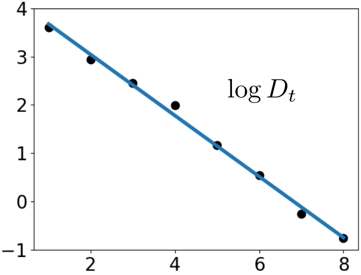
The law of equi-separation

For $1 \leqslant l \leqslant L$ and some $0 < \rho < 1$:

$$D_l \approx c\rho^l$$

An (empirical) law of deep learning

D_l: separation measure for data before passing through the l^{th} layer



The law of equi-separation

For $1 \leqslant l \leqslant L$ and some $0 < \rho < 1$:

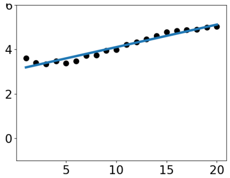
$$D_l \approx c\rho^l$$

- Nonlinearity is crucial
- Equivalently,

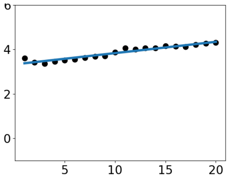
$$\log D_{l+1} - \log D_l \approx -\log \frac{1}{\rho}$$

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

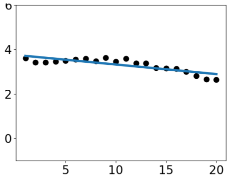
When does it emerge?



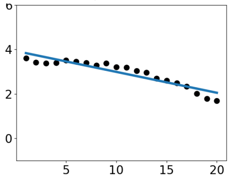
Epoch–0



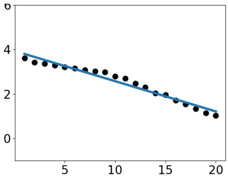
Epoch=10



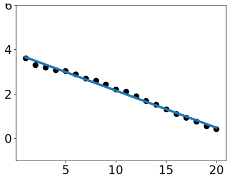
Epoch=20



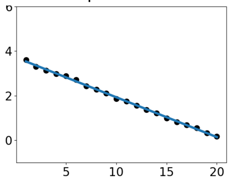
Epoch–30



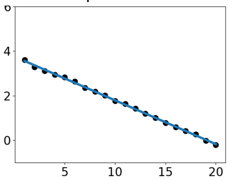
Epoch=50



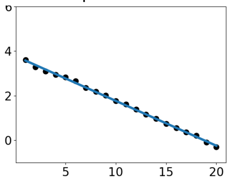
Epoch=100



Epoch–200

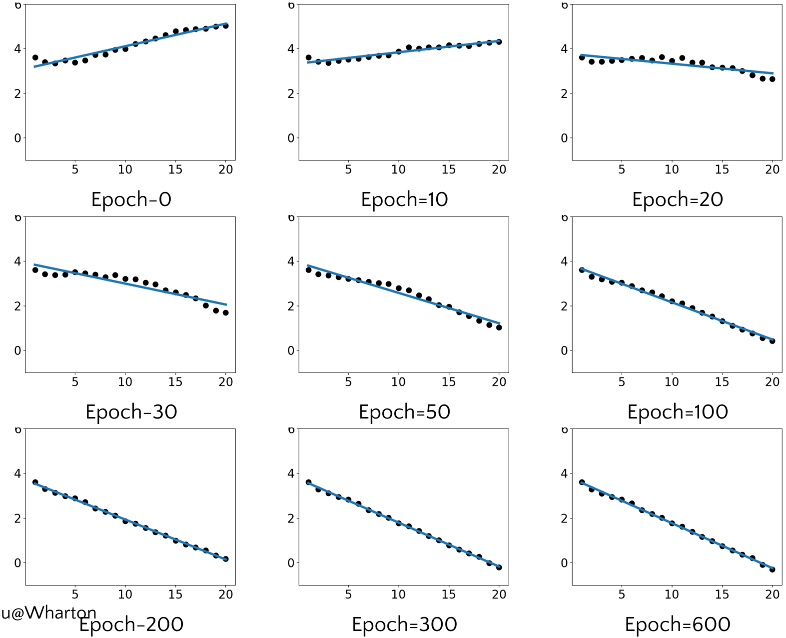


Epoch=300

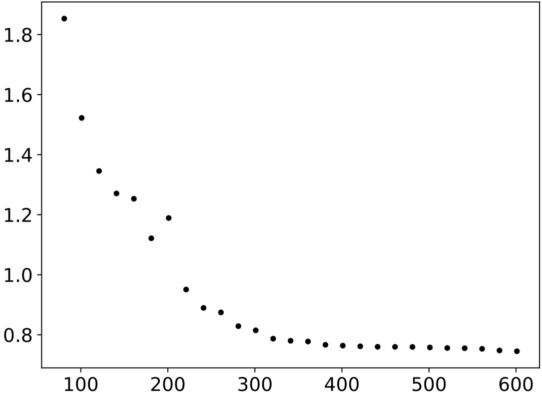


Epoch=600

When does it emerge? Earlier than neural collapse

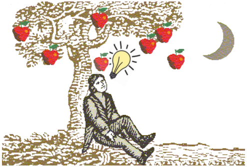


Earlier than neural collapse

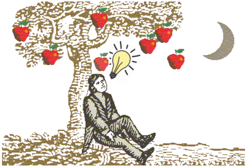


Epoch

Ask me anything about this law

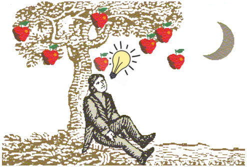


Ask me anything about this law



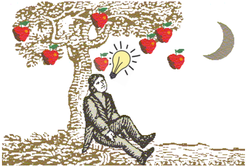
Is this law pervasive?

Ask me anything about this law



Is this law pervasive? Yes

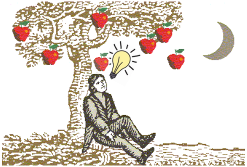
Ask me anything about this law



Is this law pervasive? Yes

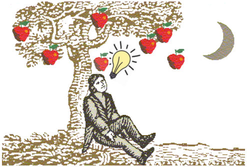
Does this law provide insights into the practice of deep learning?

Ask me anything about this law



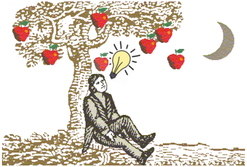
Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes

Ask me anything about this law



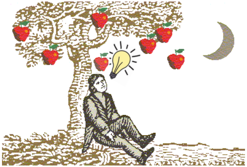
Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes
Any intuition about why this law appears?	

Ask me anything about this law



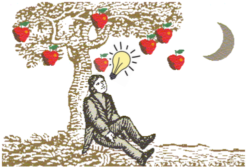
Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes
Any intuition about why this law appears?	I think so

Ask me anything about this law



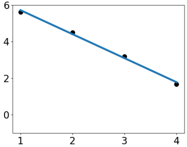
Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes
Any intuition about why this law appears?	I think so
Can we prove this law?	

Ask me anything about this law

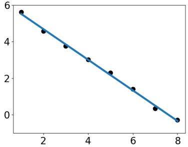


Is this law pervasive?	Yes
Does this law provide insights into the practice of deep learning?	Yes
Any intuition about why this law appears?	I think so
Can we prove this law?	Not yet

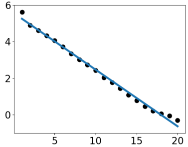
Data, imbalance, and learning rate



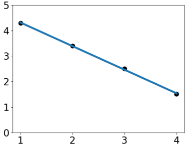
CIFAR10-4



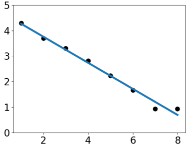
CIFAR10-8



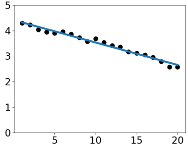
CIFAR10-20



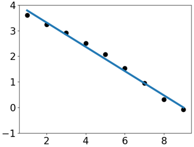
Imbalance-4



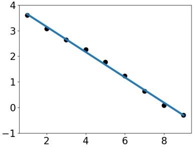
Imbalance-8



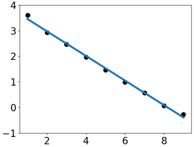
Imbalance-20



Learning rate: 0.01

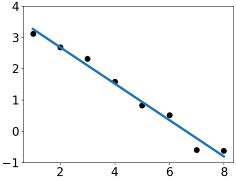


Learning rate: 0.03

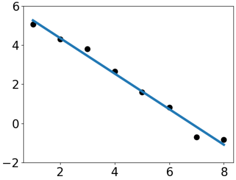


Learning rate: 0.1

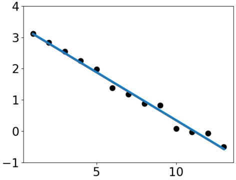
Architecture



AlexNetX-FMNIST



AlexNetX-CIFAR10



VGG13X-FMNIST

Guidelines and insights from the law of equi-separation

The trilogy of the deep learning practice

- Network architecture
- Training
- Interpretation

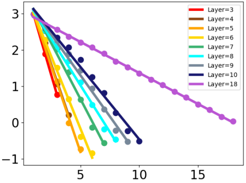
Dependence on the depth

$D_L \approx c\rho^L$: deep learning is necessarily to be deep

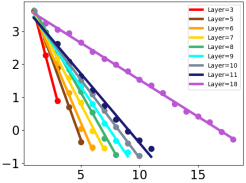
Dependence on the depth

$D_L \approx c\rho^L$: deep learning is necessarily to be deep

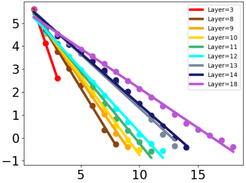
However, a complete story is slightly different



MNIST



FashionMNIST

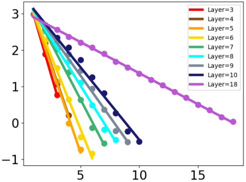


CIFAR10

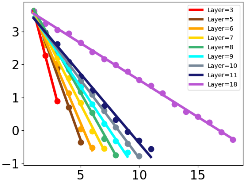
Dependence on the depth

$D_L \approx c\rho^L$: deep learning is necessarily to be deep

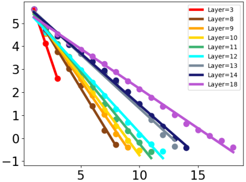
However, a complete story is slightly different



MNIST



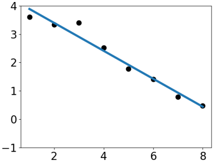
FashionMNIST



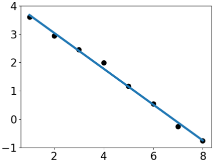
CIFAR10

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

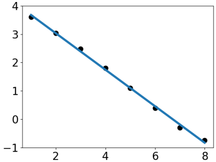
Data-separation perspective on width and shape



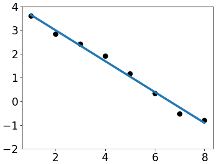
Width: 20



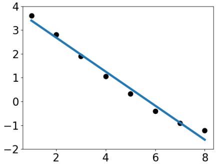
Width: 100



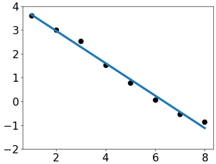
Width: 1000



Shape: narrow-wide

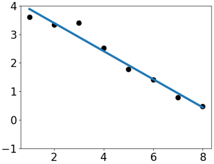


Shape: wide-narrow

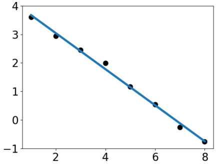


Shape: mix

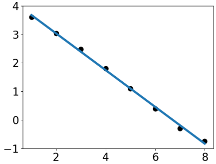
Data-separation perspective on width and shape



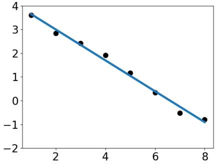
Width: 20



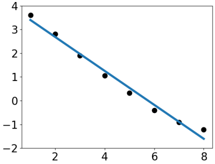
Width: 100



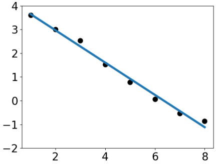
Width: 1000



Shape: narrow-wide



Shape: wide-narrow



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 $R := \frac{D_L}{D_1} = \frac{D_L}{D_{L-1}} \times \frac{D_{L-1}}{D_{L-2}} \times \cdots \times \frac{D_2}{D_1}$

Equi-separation implies robustness

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

Perturb each layer:

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

Equi-separation implies robustness

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

Perturb each layer:

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

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

$$\frac{D_L}{D_{L-1}} = \frac{D_{L-1}}{D_{L-2}} = \cdots = \frac{D_2}{D_1}$$

Equi-separation implies robustness

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

Perturb each layer:

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

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

$$\frac{D_L}{D_{L-1}} = \frac{D_{L-1}}{D_{L-2}} = \cdots = \frac{D_2}{D_1}$$

- Train at least until the law comes into effect

Equi-separation implies robustness

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

Perturb each layer:

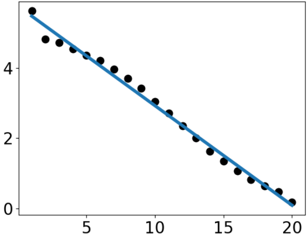
$$\left(\frac{D_L}{D_{L-1}} + \varepsilon\right)\left(\frac{D_{L-1}}{D_{L-2}} + \varepsilon\right)\cdots\left(\frac{D_2}{D_1} + \varepsilon\right)$$
$$= R + R\left(\frac{D_{L-1}}{D_L} + \frac{D_{L-2}}{D_{L-1}} + \cdots + \frac{D_1}{D_2}\right)\varepsilon + O(\varepsilon^2)$$

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

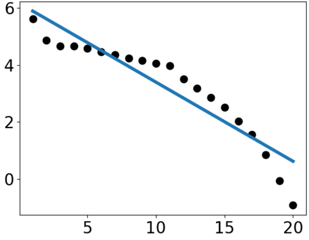
$$\frac{D_L}{D_{L-1}} = \frac{D_{L-1}}{D_{L-2}} = \cdots = \frac{D_2}{D_1}$$

- 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



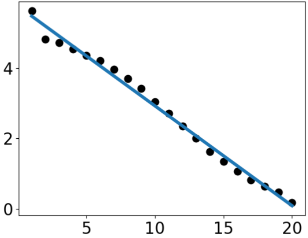
Unfrozen



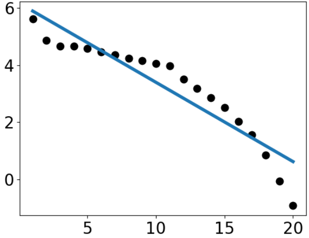
Frozen

- 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



Unfrozen

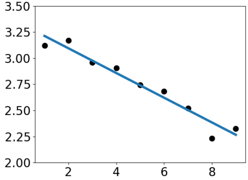


Frozen

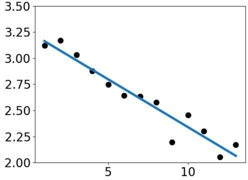
- Frozen training: bottom/top 10 layers are trained while the others are fixed
- Have about the same final separation measure and training loss
- Test accuracy:
  - Unfrozen: 21.46%
  - Frozen: 18.25%

Interpretation from data-separation perspective

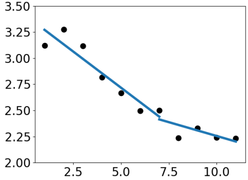
What are the basic operational modules in ResNet?



2 layers in a block



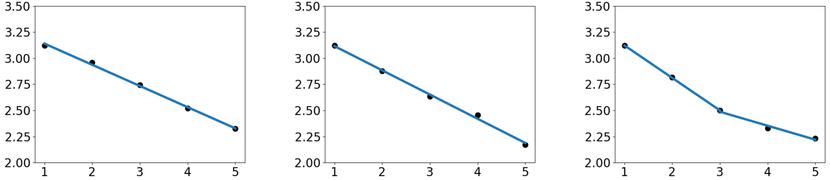
3 layers in a block



Mix

Interpretation from data-separation perspective

What are the basic operational modules in ResNet?

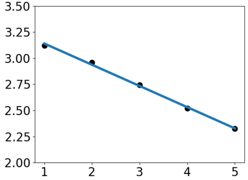


2 layers in a block | 3 layers in a block | Mix

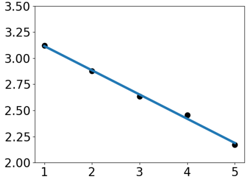
- The right module is block for ResNet

Interpretation from data-separation perspective

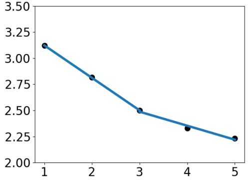
What are the basic operational modules in ResNet?



2 layers in a block



3 layers in a block

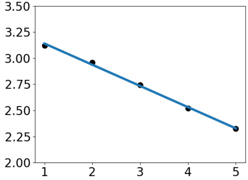


Mix

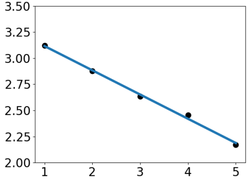
- The right module is block for ResNet
- All layers/modules are created equal

Interpretation from data-separation perspective

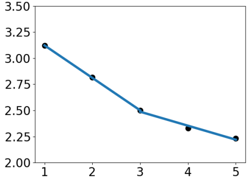
What are the basic operational modules in ResNet?



2 layers in a block



3 layers in a block



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)

Part III: A Law of Next-Token Prediction for LLMs

Collaborator

- Hangfeng He (Penn→University of Rochester)

How about Transformers/large language models?

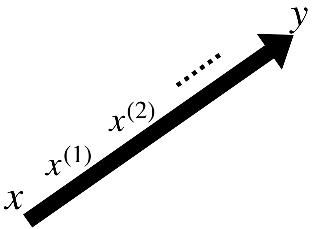
What to predict?

MLP

- Data: raw feature $\boldsymbol{x}$ and label y
- Task: use $\boldsymbol{x}$ to predict y

Transformer (GPT, decoding only)

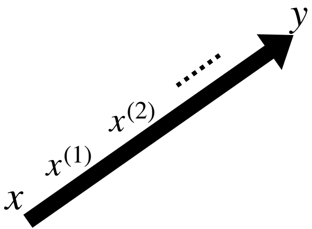
- Data: tokens $x_1, x_2, \ldots, x_T$
- Task: use $x_1 \cdots x_t$ to predict x_{t+1}



What to predict?

MLP

- Data: raw feature $\boldsymbol{x}$ and label y
- Task: use $\boldsymbol{x}$ to predict y



Transformer (GPT, decoding only)

- Data: tokens $x_1, x_2, \ldots, x_T$
- Task: use $x_1 \cdots x_t$ to predict x_{t+1}

The right metric for GPT

- Let $\widetilde{\boldsymbol{x}}$ denote the *embedding* of x
- $\widetilde{\boldsymbol{x}}^{(l)}$ denotes the feature passing through l layers in Transformer

Fact

Decoding-only LLM (GPT) predicts the $(t+1)^{\text{st}}$ token based on the last-layer feature of the t^{th} token:

$$\widetilde{\boldsymbol{x}}_t^{(L)}$$

The right metric for GPT

- Let $\widetilde{\boldsymbol{x}}$ denote the *embedding* of x
- $\widetilde{\boldsymbol{x}}^{(l)}$ denotes the feature passing through l layers in Transformer

Fact

Decoding-only LLM (GPT) predicts the $(t+1)^{\text{st}}$ token based on the last-layer feature of the t^{th} token:

$$\widetilde{\boldsymbol{x}}_t^{(L)}$$

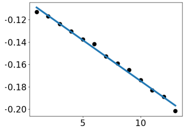
Metric

At each layer, use $\widetilde{\boldsymbol{x}}_t^{(l)}$ to predict the next token x_{t+1}. Use the error as the metric:

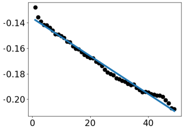
$$\frac{\sum(x_{\text{next}} - \hat{x}_{\text{next}})^2}{\sum(x_{\text{next}} - \bar{x}_{\text{next}})^2}$$

- It's 1 minus the coefficient of determination

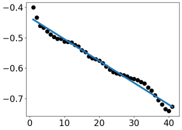
Experiments



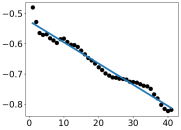
GPT-1



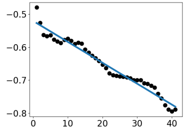
GPT-2 XL



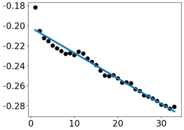
Llama-1-13B



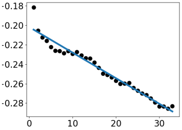
Llama-2-13B



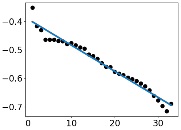
Llama-2-13B-Chat



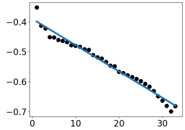
Llama-3-8B



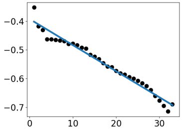
Llama-3-8B-Instruct



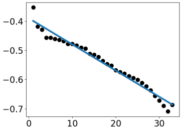
Mistral-7B-v0.1



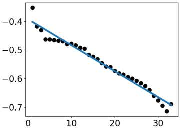
Mistral-7B-Instruct-v0.1



Mistral-7B-v0.2

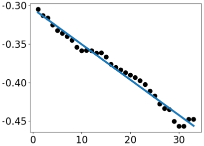


Mistral-7B-Instruct-v0.2

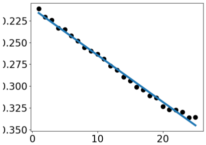


Mistral-7B-v0.3

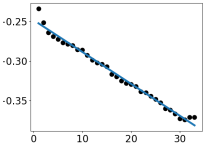
Non-Transformer architectures



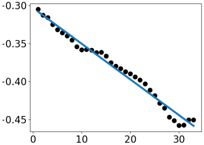
RWKV-7B



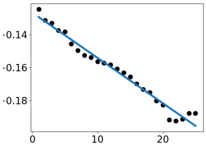
RWKV-Raven-1.5B



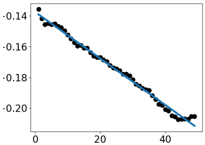
RWKV-Raven-3B



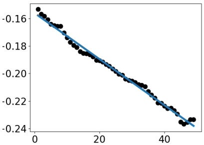
RWKV-Raven-7B



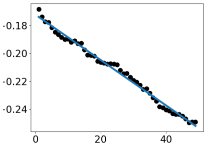
Mamba-130M



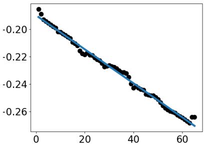
Mamba-370M



Mamba-790M

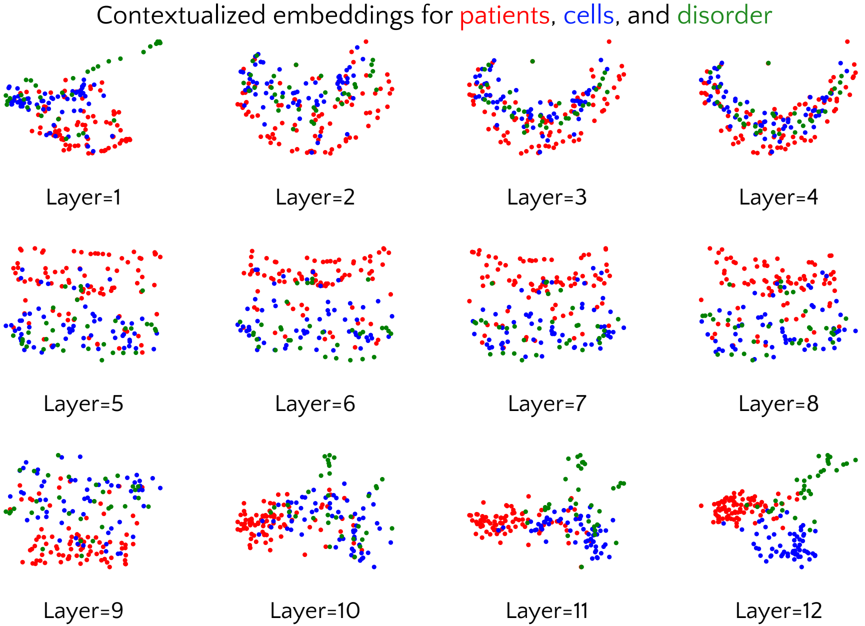


Mamba-1.4B

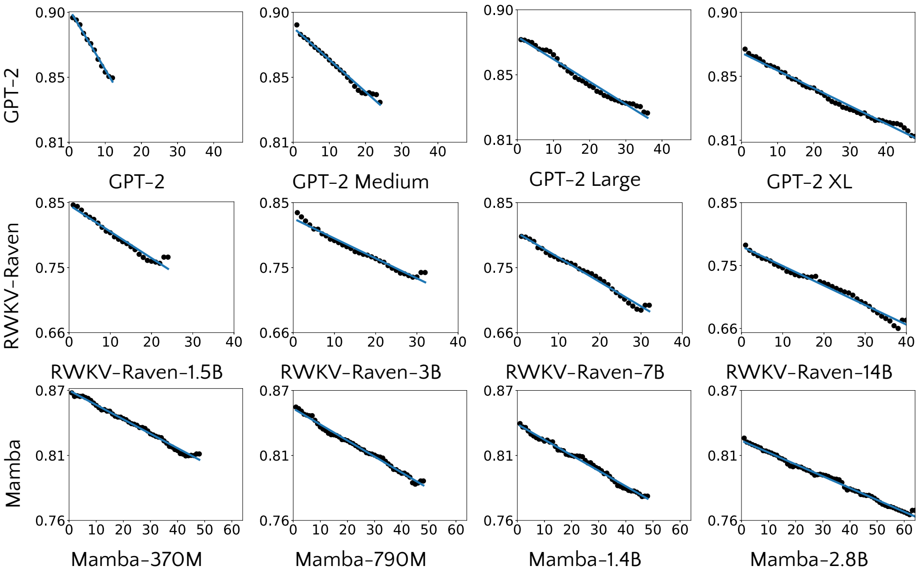


Mamba-2.8B

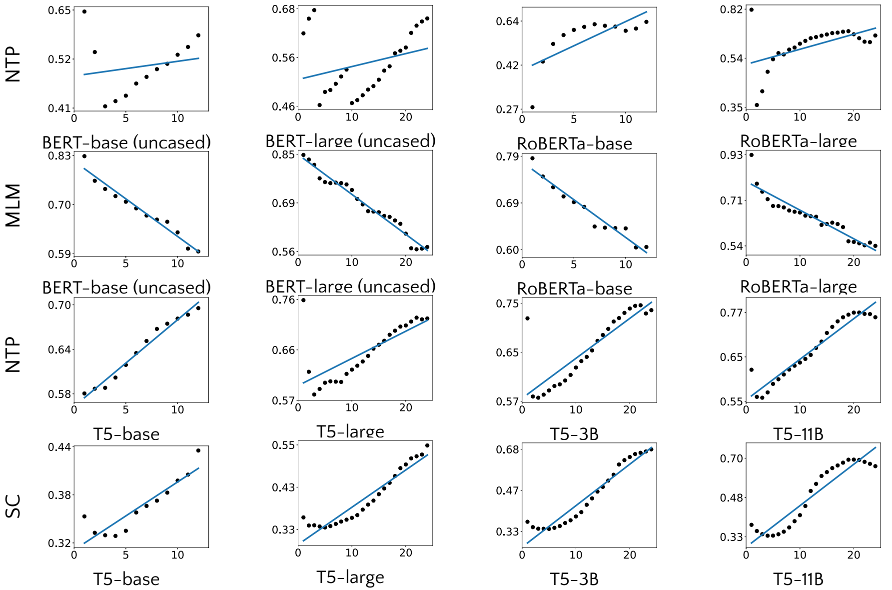
In contrast, (raw) embeddings are chaotic



The law of equi-learning with varying model sizes

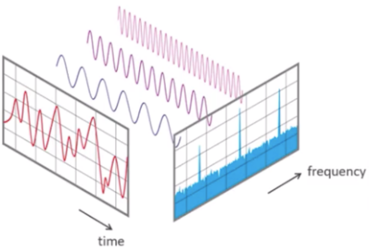


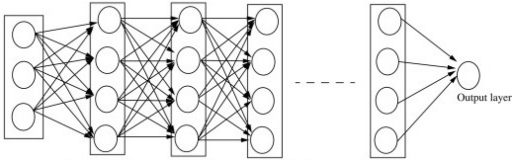
Tasks matter



Concluding remarks

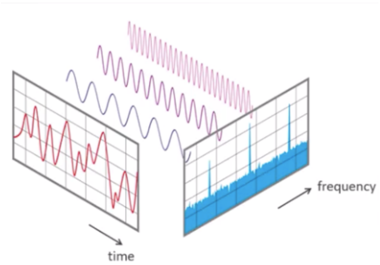
Rambling thoughts

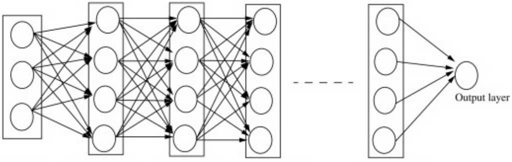




- Model the world as additive
- Model the world as a composition

Rambling thoughts

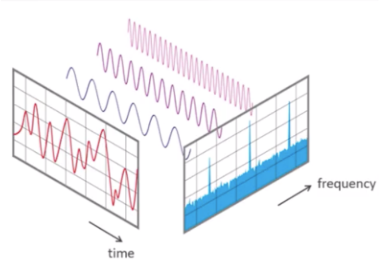


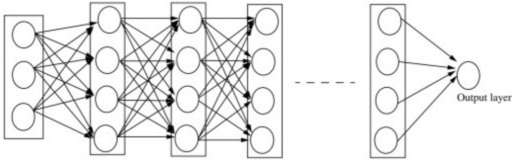


- Model the world as additive
- $f = f_1 + f_2 + \cdots + f_m$

- Model the world as a composition
- $f = f_1 \circ f_2 \circ \cdots \circ f_m$

Rambling thoughts

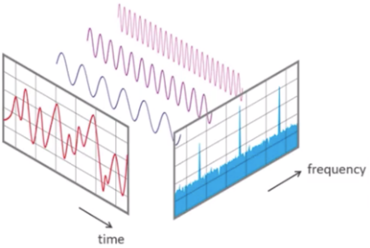


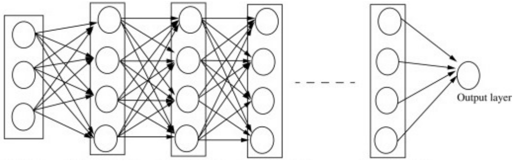


- Model the world as additive
- $f = f_1 + f_2 + \cdots + f_m$
- Tons of beautiful mathematics

- Model the world as a composition
- $f = f_1 \circ f_2 \circ \cdots \circ f_m$
- Mathematically, little is known

Rambling thoughts





- Model the world as additive
- $f = f_1 + f_2 + \cdots + f_m$
- Tons of beautiful mathematics

- Model the world as a composition
- $f = f_1 \circ f_2 \circ \cdots \circ f_m$
- Mathematically, little is known
- But equi-separation/learning laws show $f_1, \ldots, f_m$ are structured

Take-home messages

A law governing how data is processed in intermediate layers

- For both MLP and Transformer (and beyond)
- No mathematical proof yet

Take-home messages

A law governing how data is processed in intermediate layers

- For both MLP and Transformer (and beyond)
- No mathematical proof yet

References

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
3. *A Law of Next-Token Prediction in Large Language Models*
   with Hangfeng He
   arXiv:2408.13442, 2024