Towards Quantifying the Hessian Structure of Neural Networks

Yushun Zhang, June 27th, 2025

Presented at FAI Seminar

School of Data Science,

The Chinese University of Hong Kong, Shenzhen
SRIBD

Overview of This Talk

- **Part I: Empirical observations:**
  - Hessian of NNs exhibit near-block-diagonal structure
    (e.g., Collobert 2004; Zhang et al. 2024 a,b; Kunstner et al. 2024)
  - But why? No theory so far

- **Part II: Intuitions:**
  - Intuitions for linear NNs: a linear algebra perspective
  - Intuition for non-linear NNs: linear algebra & probability perspective

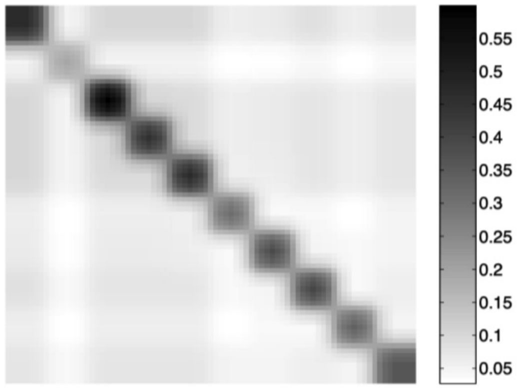
- **Part III: Our theoretical results & technical difficulties**
  - By using random matrix theory (RMT), we rigorously prove the existence of special Hessian structure
  - Explain some challenges and why traditional RMT can NOT be directly applied in our case

- **Part IV: Implications to LLMs**

Contents

Empirical Observations

- **Hessian of NNs are numerically observed to be near-block-diagonal**



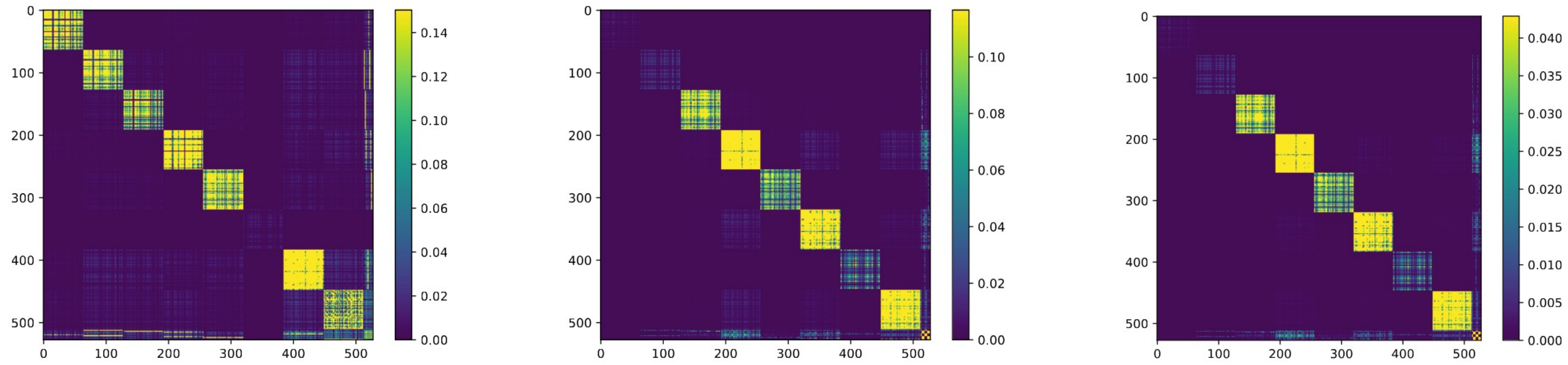
(a) Hessian of an MLP [18] after 1 step

Hessian of an 1-hidden-layer NN

Figure from: Large Scale Machine Learning, Collobert, thesis, 2004

Empirical Observations

- **Hessian of NNs are numerically observed to be near-block-diagonal**



(b) Hessian of an MLP at 1% step

(c) Hessian of an MLP at 50% step

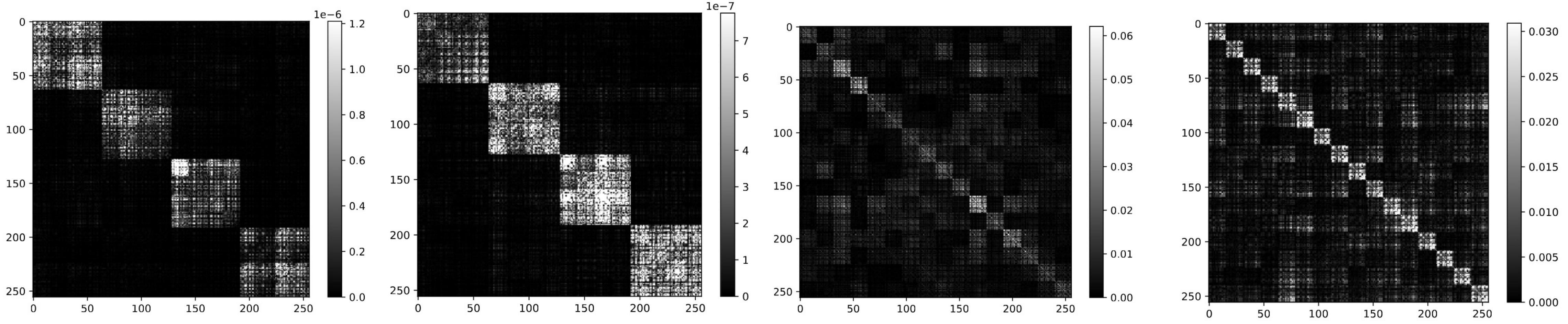
(d) Hessian of an MLP at 100% step

Hessian of 1-hidden-layer NNs

Figure (b,c,d): Why Transformers Need Adam: A Hessian Perspective, **Zhang,** Chen, Ding, Li, Sun, Luo, NeurIPS 2024

Empirical Observations

- **Hessian of NNs are numerically observed to be near-block-diagonal**



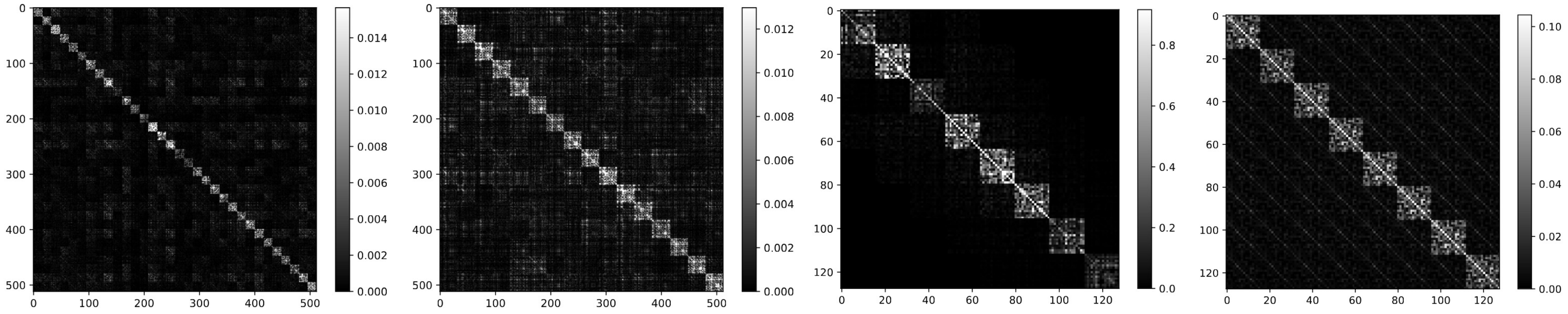
(a) `query` (4 heads) (b) `key` (4 heads) (c) `value` (4 heads) (d) `attn.proj` (16 neurons)

Hessian of Transformers Part I: Attention

Figure from: Adam-mini: Use Fewer Learning Rates To Gain More, **Zhang,** Chen, et al., ICLR 2025

Empirical Observations

- **Hessian of NNs are numerically observed to be near-block-diagonal**



(e) `mlp.fc_1` (32 neurons) (f) `mlp.proj` (16 neurons) (g) `embed` (8 tokens) (h) `output` (8 tokens)

Hessian of Transformers Part II: MLPs and embeddings

Figure from: Adam-mini: Use Fewer Learning Rates To Gain More, **Zhang,** Chen, et al., ICLR 2025

Empirical Observations

- **Hessian of NNs are numerically observed to be near-block-diagonal**

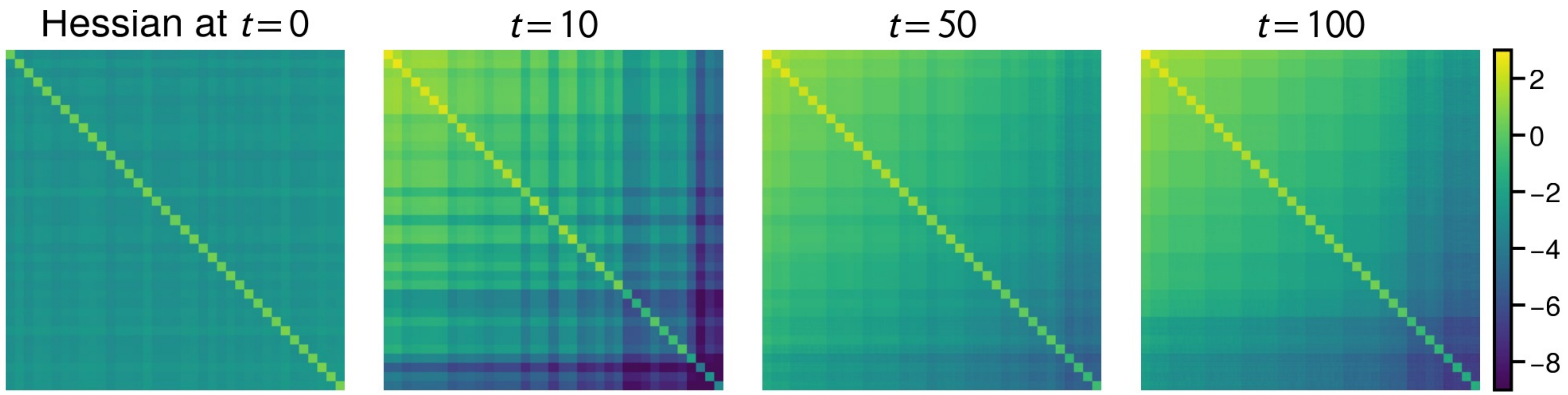


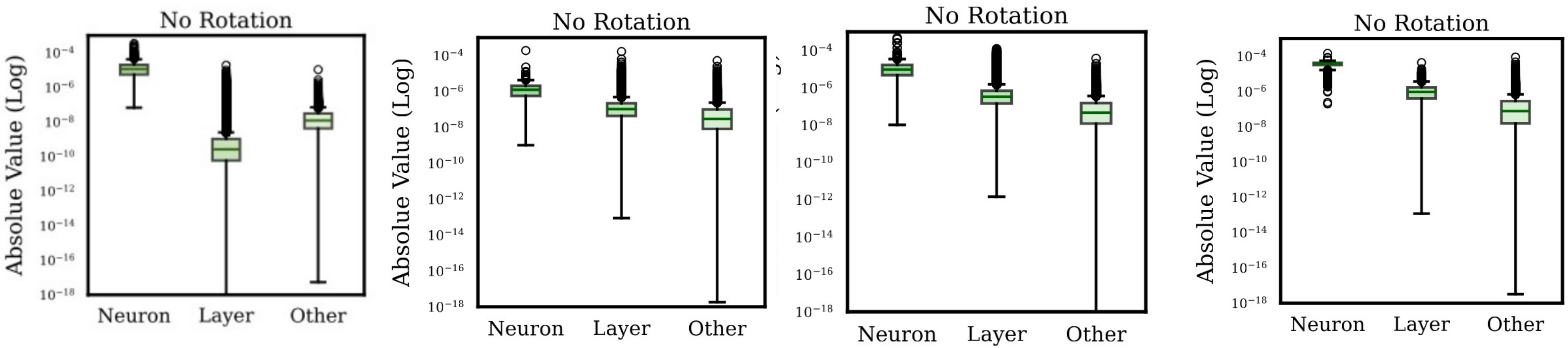
Figure 8: **The diagonal Hessian blocks are orders of magnitude larger than off-diagonal blocks.**

Hessian of a linear model + CE loss

Figure from: Heavy-Tailed Class Imbalance and Why Adam Outperforms GD on LLMs, Kunstner et al. NeurIPS 2024

Empirical Observations

- **Hessian of NNs are numerically observed to be near-block-diagonal**



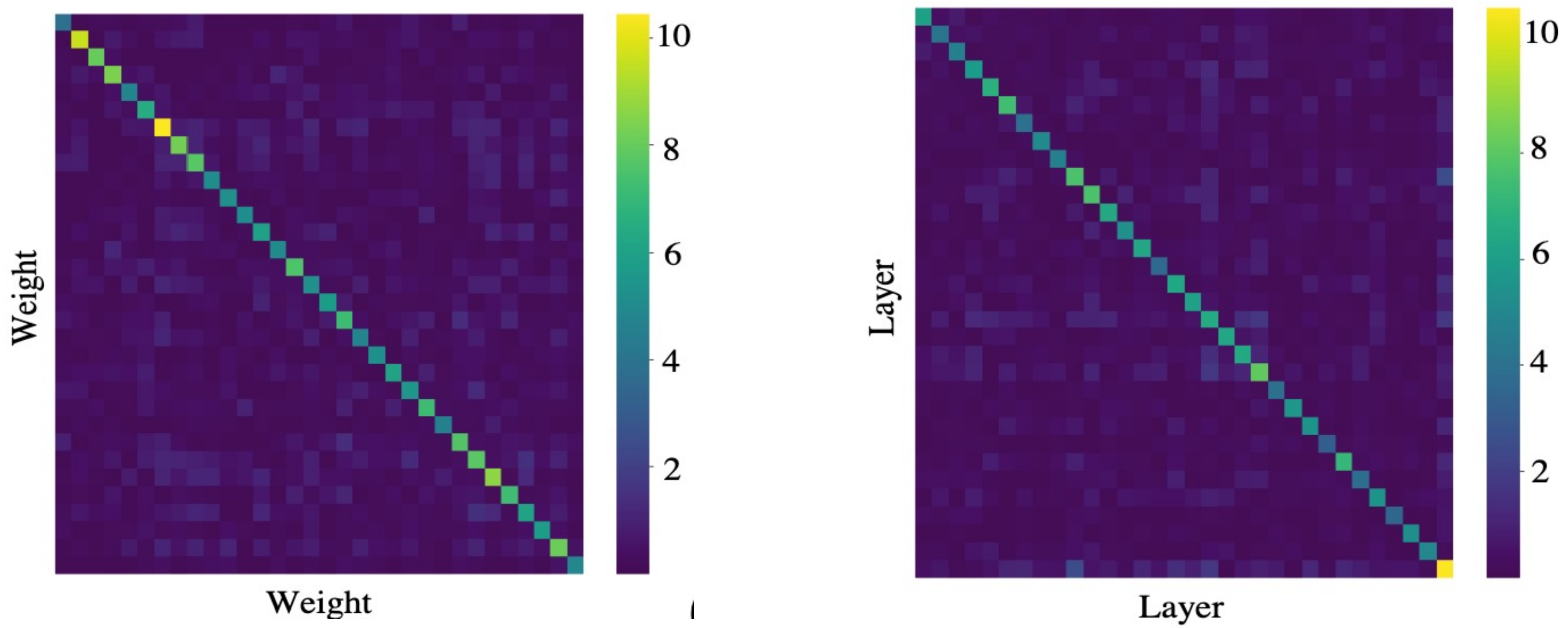
Embedding layer | Query | Value | MLP

Hessian sub-blocks sampled from GPT2-125M
(diag-blocks > 10^4 off-diag-blocks)

Figure from: Understanding Adam Requires Better Rotation Dependent Assumptions, Maes, et al., 2024

Empirical Observations

- **Hessian of NNs are numerically observed to be near-block-diagonal**



Approximated Hessian of 1 layer in Llama-7B & 32 layers in Llama-7B

Figure from: CBQ: Cross-Block Quantization for Large Language Models, Ding, et al., ICLR 2025

Motivation: Why Studying Hessian Structure?

- **1. Hessian structure is crucial for understanding NN training**
  - The effectiveness of Adam
    (Zhang et al 24a, Kunstner et al. 24)
  - The effectiveness of general diagonal-preconditioned methods
    (Sun and Ye, 21, Qu et al. 22, Das et al. 24)
  - The effectiveness of recent block-diagonal-preconditioned methods

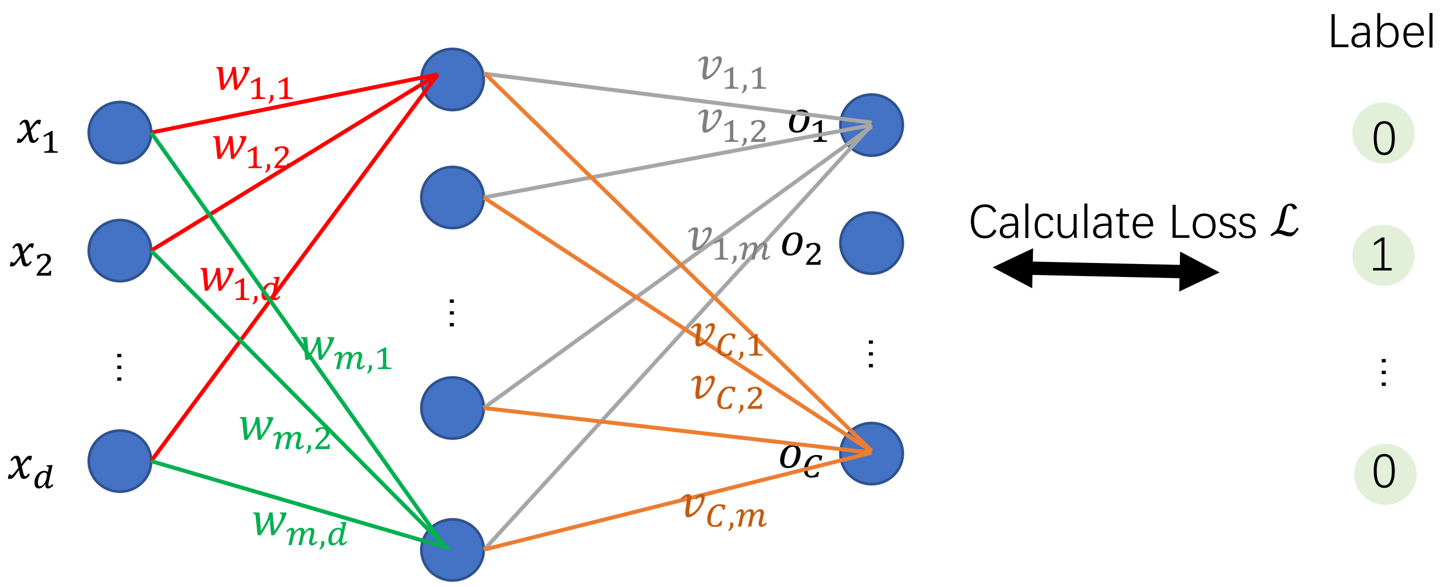
    (Shampoo, Muon)
- **2. Hessian structure can help design new training methods for NNs**
  - Recently, Adam-mini utilizes the block-diag structure to cut down 50% memory in Adam
  - Low precision training (Ding et al. 2025)
  - More is coming..
- **3. Offering a new function class for optimization community**
  - Typical problems do NOT have such structure:
    In classical non-linear programming dataset (Lavezzi et al 22), all problems have non-block-diag Hessian
  - Motivate new study into this specific class of problems

Today, we focus on…

- **Why do Hessian matrices look like this? Is it trivial?**
- **What does one block correspond to?**
- **What is the fundamental reason for this structure?**
  - Does it always hold for arbitrary NNs?
  - If not, is there common factor holds in all above, but we overlooked?
  - Is it a local property or global?
- **Any more structure missed in the previous experiments?**

Review: What is Hessian Matrix for NNs

Data: $x \in R^d, y \in R^C$ one-hot



$$\min_{W \in R^{m \times d}, V \in R^{m \times C}} \frac{1}{N} \sum_{n=1} \ell(f(x_n), y_n)$$

$$W = \begin{bmatrix} w_1^T \\ \vdots \\ w_m^T \end{bmatrix} \in R^{m \times d}, V = \begin{bmatrix} v_1^T \\ \vdots \\ v_C^T \end{bmatrix} \in R^{C \times m}$$

$$f(x_n) = \begin{bmatrix} v_1^T \sigma(W x_n) \\ \vdots \\ v_C^T \sigma(W x_n) \end{bmatrix} \in R^C$$

$$\min_{W,V} \ell_{\mathrm{MSE}}(W,V) := \frac{1}{N} \sum_{n=1}^{N} \|V\sigma(Wx) - \mathcal{Y}_n\|_2^2, \qquad \min_{W,V} \ell_{\mathrm{CE}}(W,V) := -\frac{1}{N} \sum_{n=1}^{N} \log\left(\frac{\exp(v_{y_n}^\top \sigma(Wx))}{\sum_{k=1}^{c} \exp(v_k^\top \sigma(Wx))}\right)$$

Review: What is Hessian Matrix for NNs

	d	d	d	m	m	m
d	$H_{w_1w_1}$	$\cdots$	$H_{w_1w_m}$	$H_{w_1v_1}$	$\cdots$	$H_{w_1v_C}$
d		$\ddots$				
d	$H_{w_mw_1}$	$\cdots$	$H_{w_mw_m}$	$H_{w_mv_1}$	$\cdots$	$H_{w_mv_C}$
m	$H_{v_1w_1}$	$\cdots$	$H_{v_1w_m}$	$H_{v_1v_1}$	$\cdots$	$H_{v_1v_C}$
m		$\ddots$			$\ddots$	
m	$H_{v_Cw_1}$	$\cdots$	$H_{v_Cw_1}$	$H_{v_Cv_1}$	$\cdots$	$H_{v_Cv_C}$

Size of Hessian $= (md + Cm) * (md + Cm)$

$$W = \begin{bmatrix} w_1^T \\ \vdots \\ w_m^T \end{bmatrix} \in R^{m\times d}, V = \begin{bmatrix} v_1^T \\ \vdots \\ v_C^T \end{bmatrix} \in R^{C\times m}$$

$$H_{w_iw_i} = \frac{\partial^2 \mathcal{L}}{\partial w_i \partial w_i^T} \in R^{d\times d}$$

$$H_{w_iw_j} = \frac{\partial^2 \mathcal{L}}{\partial w_i \partial w_j^T} \in R^{d\times d}$$

$$H_{v_iv_i} = \frac{\partial^2 \mathcal{L}}{\partial v_i \partial v_i^T} \in R^{m\times m}$$

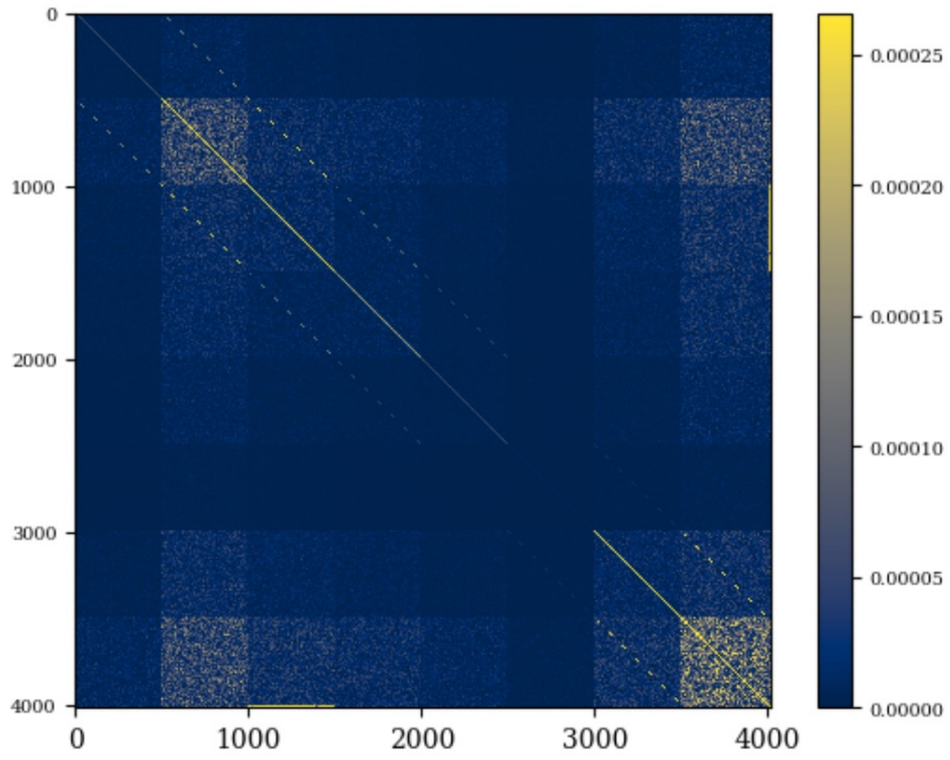
$$H_{v_iv_j} = \frac{\partial^2 \mathcal{L}}{\partial v_i \partial v_j^T} \in R^{m\times m}$$

$$H_{w_iv_i} = \frac{\partial^2 \mathcal{L}}{\partial w_i \partial v_i^T} \in R^{d\times m}$$

$$H_{w_iv_j} = \frac{\partial^2 \mathcal{L}}{\partial w_i \partial v_j^T} \in R^{d\times m}$$

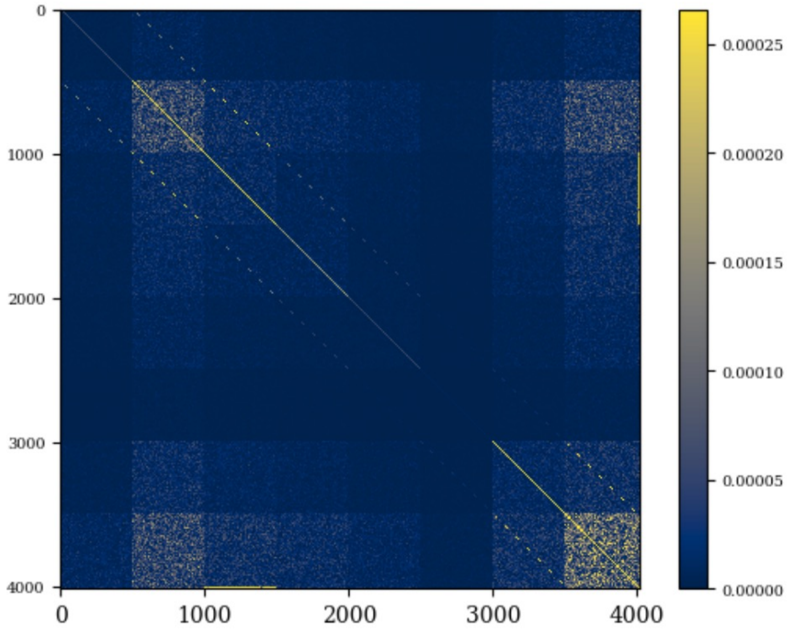
Initial trial: binary classification

- Simple setting: Linear model + CE loss, binary classification
- Cannot see special Hessian structures. Why?

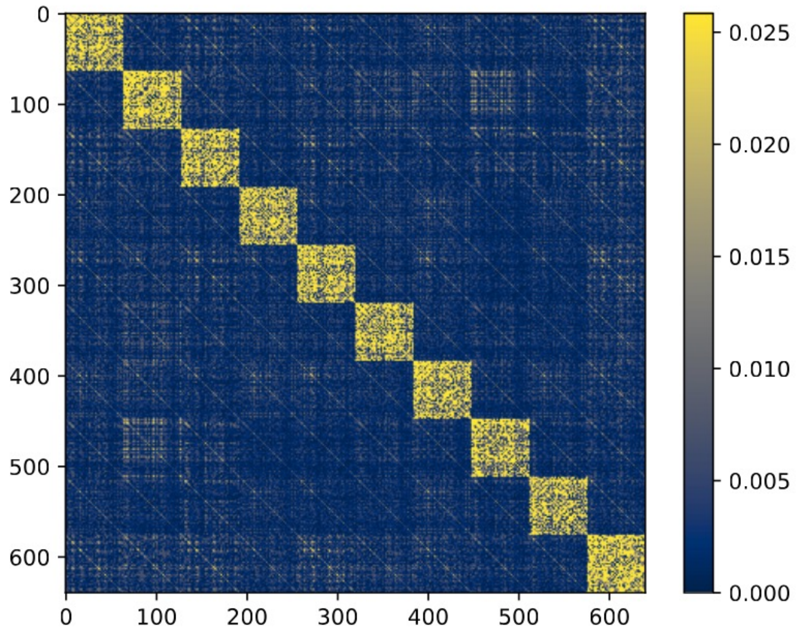


We find a phase transition as # class C → ∞

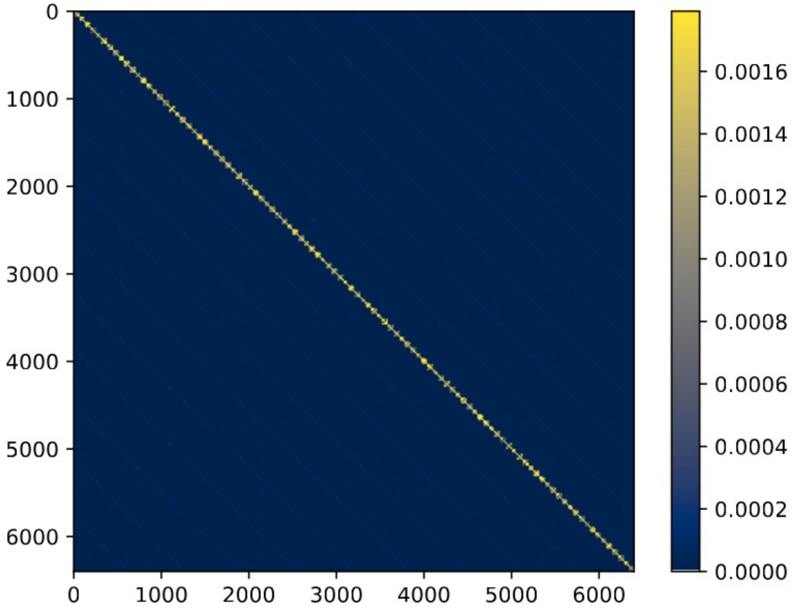
- **Simple setting: Linear model + CE loss, #C class classification**



C = 2



C = 10

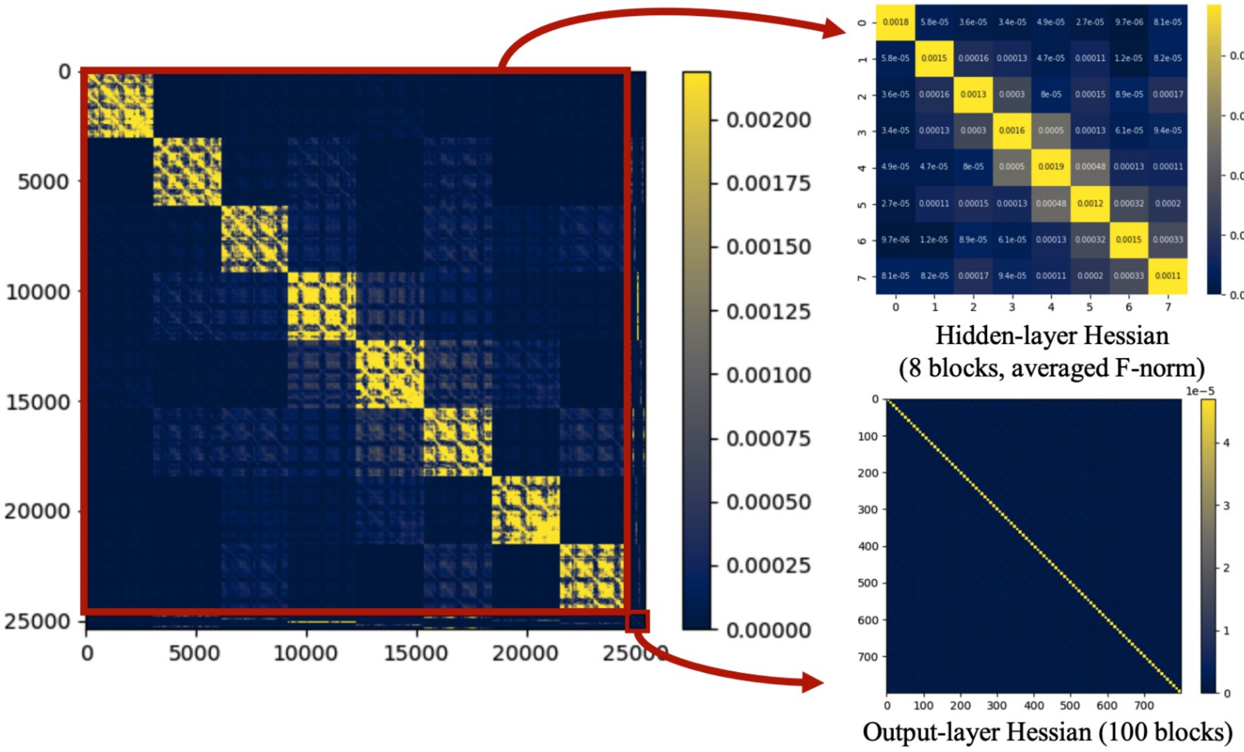


C = 100

🤔 It seems that **large #class C** is important

Empirical Observations: CIFAR-100

- Setup: CIFAR-100, sample size $N = 128$, input dim $d = 32000$, # classes $C = 100$
- 1-hidden-layer NN with 8 neurons, ReLU, random init
- We observe that Hessian is **near-block-diagonal.** Total # blocks = # neuron + # class = 8 + 100 = 108



(b) 1-hidden-layer network with CE loss

🤔

It seems that **large #class C** is important

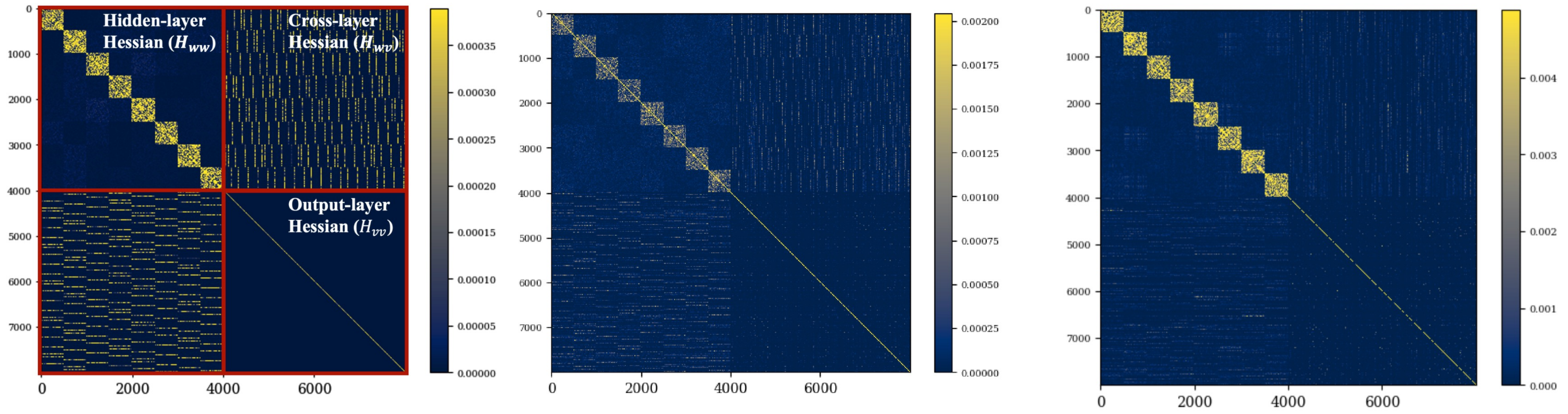
🤔

But the hidden & output layer proportion is **too imbalanced**

Did we miss anything in the cross-layer part?

Empirical Observations: Gaussian Data

- **Setup:** Standard Gaussian $X_N \in R^{d\times N}$, random label in $[C]$, $N = 5000, d = 500, C = 500$
  (we changed d and C to **balance the proportion of Hww and Hvv**)
- 1-hidden-layer NN with 8 neurons, random init.   Total # blocks = # neuron + # class = 8 + 500 = 508

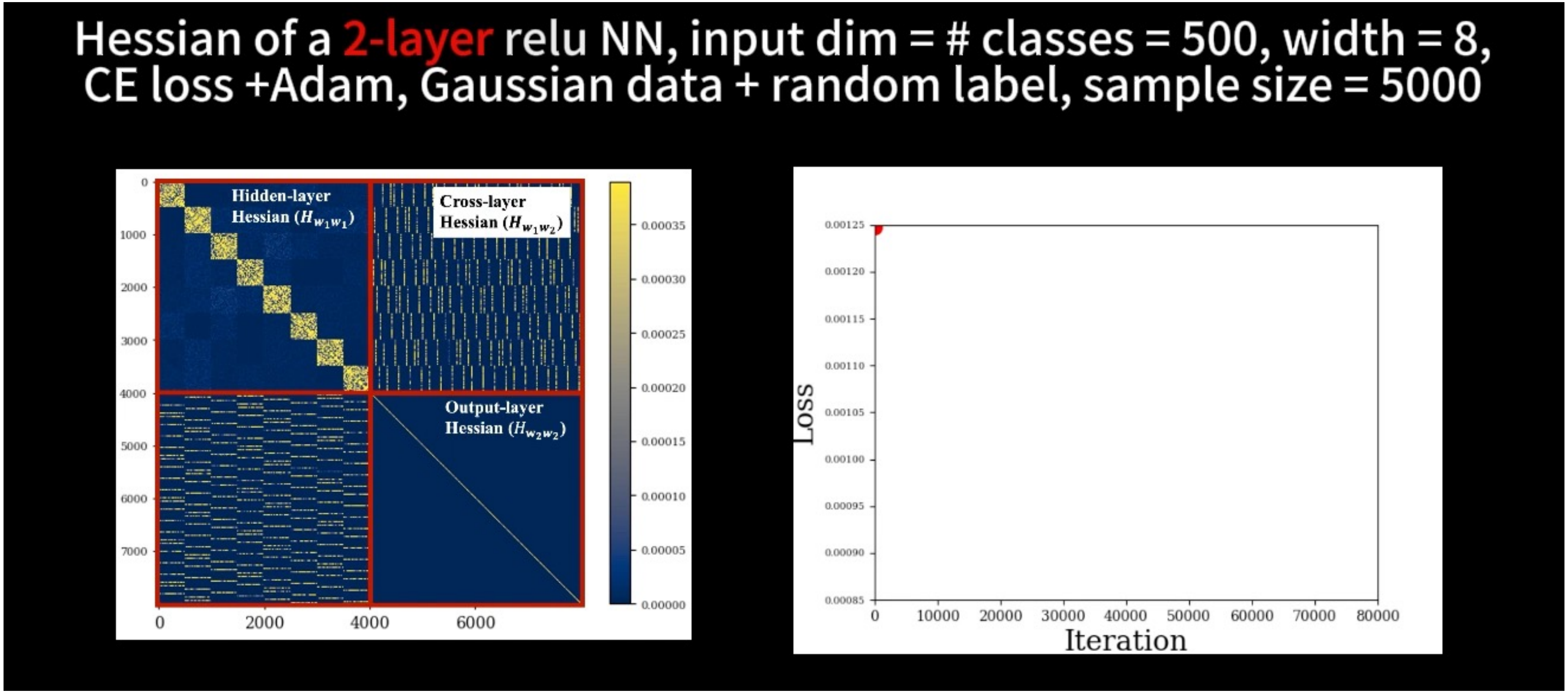


(a) Hessian at initialization   (b) Hessian at 10% steps   (f) Hessian at 100% steps

(i) block-circulant-block-diagonal structure at initialization
(ii) The block-circulant part vanishes along training
(iii) The near-block-diagonal pattern maintains along training

🤔Why?

Empirical Observations: Gaussian Data

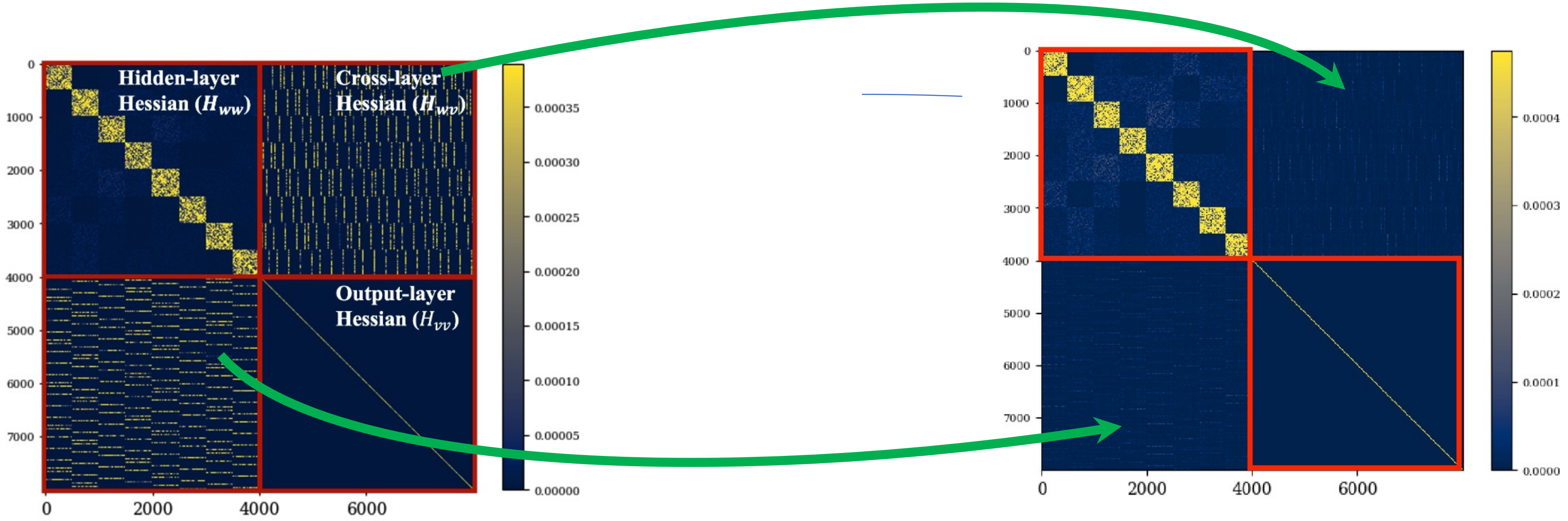


[Click to play the video]

(i) block-circulant-block-diagonal structure at initialization
(ii) The block-circulant part vanishes along training
(iii) The near-block-diagonal pattern maintains along training

🤔Why?

We reveal two forces that shape the Hessian structure:



(a) Hessian at initialization

(f) Hessian at 100% steps

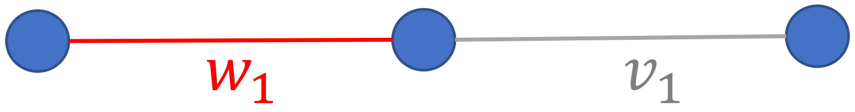
Force I: a ``**static force**'' rooted in the architecture design (e.g., large # Class C);
Force II: and a ``**dynamic force**'' arisen from training.

- In the following:
  - 1. We first provide intuitions on the structure
  - 2. a simple explanation on the ``**dynamic force**''
  - 3. rigorous theory on the ``**static force**'' at random initialization

Contents

Intuition: Example 1

- **Let us start from the most simple NN:**
- **Example 1: Single-input-single-output (SISO):**



Input data x = 1. No activation, label = 0, MSE loss: $\ell(w_1 v_1) = \frac{1}{2}(w_1 v_1)^2$

Gradient:

$$\frac{\partial \ell}{\partial w_1} = w_1 v_1^2 \qquad \frac{\partial \ell}{\partial v_1} = w_1^2 v_1$$

Hessian:

$$\frac{\partial^2 \ell}{\partial w_1 \partial w_1} = v_1^2 \qquad \frac{\partial^2 \ell}{\partial v_1 \partial w_1} = 2 w_1 v_1$$

$$\frac{\partial^2 \ell}{\partial w_1 \partial v_1} = 2 w_1 v_1 \qquad \frac{\partial^2 \ell}{\partial v_1 \partial v_1} = w_1^2$$

Observation: off-diagonal entries are non-zero

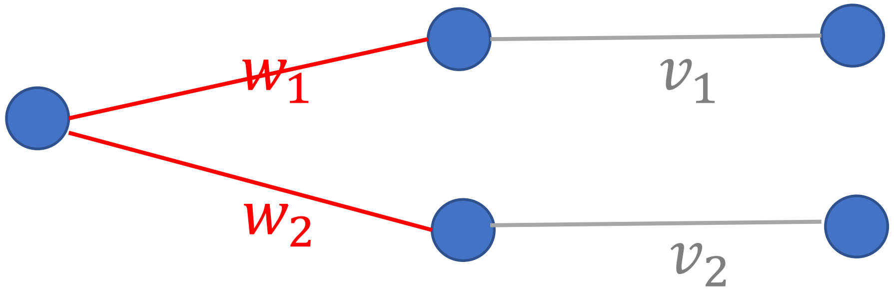
i.e., w1 and v1 has ``correlations"

Why? See from computation graph w1 and v1 are **linked together**

Lesson: learn to check the link!

Intuition: Example 2-1

- **Example 2-1: Single-input-multi-output (SIMO):**
  (this is not a standard NN, but is good for understanding)



Input data x = 1. No activation, label = 0, MSE loss: $\ell(w_1, w_2, v_1, v_2) = \frac{1}{2}(w_1 v_1)^2 + \frac{1}{2}(w_2 v_2)^2$

Gradient: $\frac{\partial \ell}{\partial w_1} = w_1 v_1^2$

Hessian (1st row): $\frac{\partial^2 \ell}{\partial w_1 \partial w_1} = v_1^2 \quad \frac{\partial^2 \ell}{\partial w_1 \partial w_2} = 0 \quad \frac{\partial^2 \ell}{\partial w_1 \partial v_1} = 2 w_1 v_1 \quad \frac{\partial^2 \ell}{\partial w_1 \partial v_2} = 0$

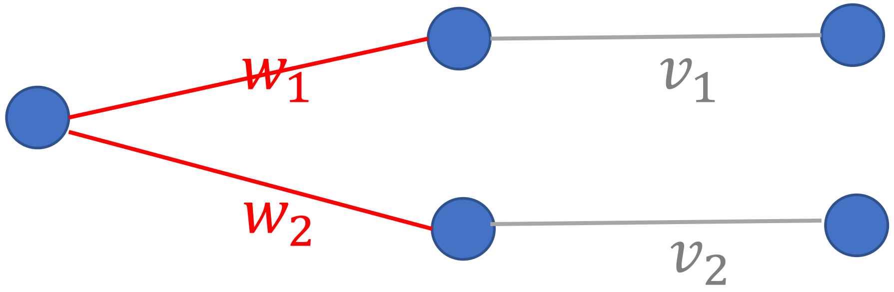
We observer two zeros in the first-row of Hessian

🤔**Why 0? Just check the links!**

E.g., no link between w_1, v_2

Intuition: Example 2-1

- **Example 2-1: Single-input-multi-output (SIMO):**
  (this is not a standard NN, but is good for understanding)



check the links!

Input data x = 1. No activation, label = 0, MSE loss: $\ell(w_1, w_2, v_1, v_2) = \frac{1}{2}(w_1 v_1)^2 + \frac{1}{2}(w_2 v_2)^2$

Hessian:

	w_1	w_2	v_1	v_2
w_1		0		0
w_2	0		0	
v_1		0		0
v_2	0		0	

This is a most simple block-circulant-block-diagonal matrix

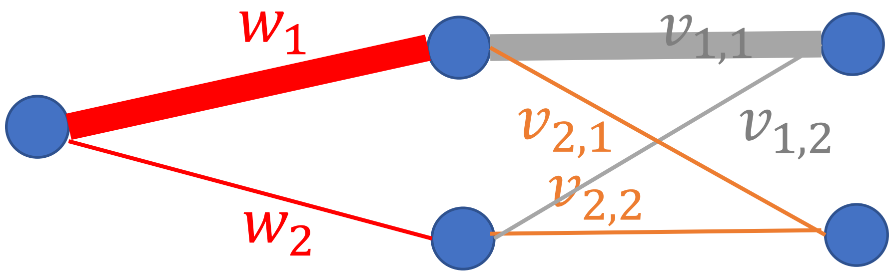
Observation:

$H_{i,j} \neq 0$ ⟸ i *and* j are connected in the graph
(which means: i *and* j has **multiplicative relation**)

$H_{i,j} = 0$ ⟸ i *and* j are not connected in the graph
(which means: i *and* j has **no multiplicative relation**)

Intuition: Example 2-2

- **Example 2-2: Single-input-multi-output (SIMO):**



Denote $w = (w_1,\ w_2)$,
$v_1 = (v_{1,1}, v_{1,2})$,
$v_2 = (v_{2,1}, v_{2,2})$

$$\ell(w, v_1, v_2) = \frac{1}{2}(v_1^T w)^2 + \frac{1}{2}(v_2^T w)^2 = \frac{1}{2}\left(v_{1,1}w_1 + v_{1,2}w_2\right)^2 + \frac{1}{2}\left(v_{2,1}w_1 + v_{2,2}w_2\right)^2$$

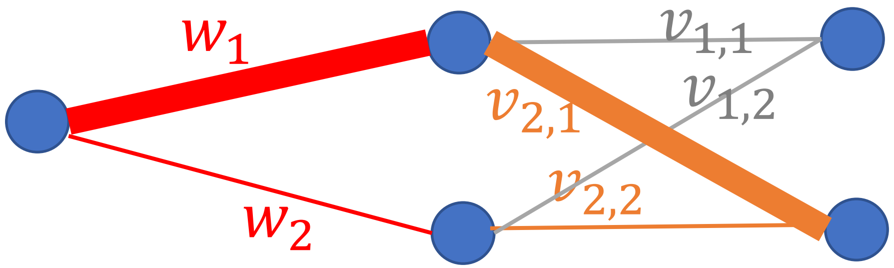
Hessian (1st row):

	w_1	w_2	$v_{1,1}$	$v_{1,2}$	$v_{2,1}$	$v_{2,2}$
w_1						

check the links!

Intuition: Example 2-2

- **Example 2-2: Single-input-multi-output (SIMO):**



Denote $w = (w_1,\ w_2)$,
$v_1 = (v_{1,1}, v_{1,2})$,
$v_2 = (v_{2,1}, v_{2,2})$

$$\ell(w, v_1, v_2) = \frac{1}{2}(v_1^T w)^2 + \frac{1}{2}(v_2^T w)^2 \ = \frac{1}{2}\left(v_{1,1}w_1 + v_{1,2}w_2\right)^2 + \frac{1}{2}\left(v_{2,1}w_1 + v_{2,2}w_2\right)^2$$

Hessian (1^{st} row):

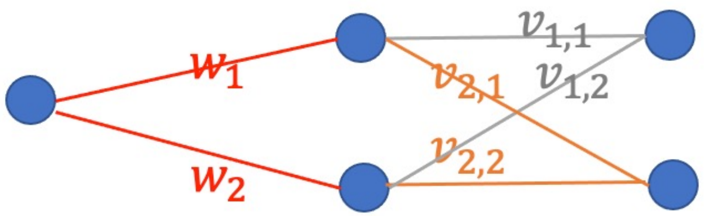
	w_1	w_2	$v_{1,1}$	$v_{1,2}$	$v_{2,1}$	$v_{2,2}$
w_1						

check the links!

Remark: The white box might not be 0 due to the cross-term,
Need more detailed calculation, but usually the signal can be rather weak
(due to indirect muliplicative relation)

Intuition: Example 2-2

$$\ell(w, v_1, v_2) = \frac{1}{2}\left(v_{1,1}w_1 + v_{1,2}w_2\right)^2 + \frac{1}{2}\left(v_{2,1}w_1 + v_{2,2}w_2\right)^2$$



Hessian:

	w_1	w_2	$v_{1,1}$	$v_{1,2}$	$v_{2,1}$	$v_{2,2}$
w_1						
w_2						
$v_{1,1}$						
$v_{1,2}$						
$v_{2,1}$						
$v_{2,2}$						

What about here?
Just follow the same logic

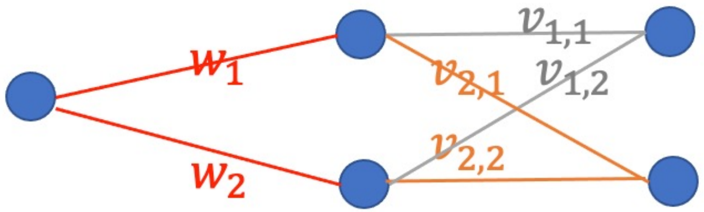
check the links!

Intuition: Example 2-2

$$\ell(w, v_1, v_2) = \frac{1}{2}(v_1^T w)^2 + \frac{1}{2}(v_2^T w)^2 \;= \frac{1}{2}\left(v_{1,1}w_1 + v_{1,2}w_2\right)^2 + \frac{1}{2}\left(v_{2,1}w_1 + v_{2,2}w_2\right)^2$$

Hessian:

	w_1	w_2	$v_{1,1}$	$v_{1,2}$	$v_{2,1}$	$v_{2,2}$
w_1						
w_2						
$v_{1,1}$					0	0
$v_{1,2}$					0	0
$v_{2,1}$			0	0		
$v_{2,2}$			0	0		



No correlation between v_1 and v_2
(Check the graph:
NO link between them!)

Intuition: Example 3

- **Example 3: Multi-input-multi-output (SIMO):**

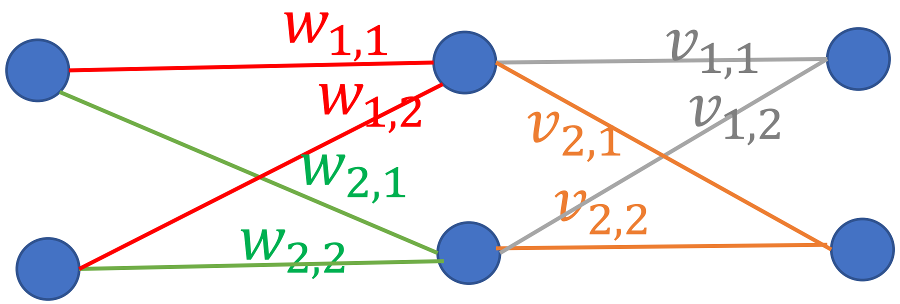
Denote $W = \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix} \in R^{2\times 2}$, $V = \begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \in R^{2\times 2}$

$w_1 = (w_{1,1}\,,\ w_{2,2}\,),$

$w_2 = (w_{2,1}\,,\ w_{2,2}\,),$

$v_1 = (v_{1,1}, v_{1,2}),$

$v_2 = (v_{2,1}, v_{2,2})$



$\ell(W, V) = \ ||VW||_F^2$ (2-layer NN, X = Identity, Y = 0, no activation)

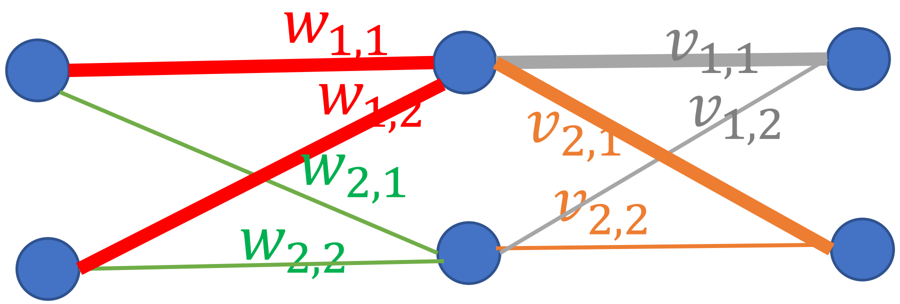
$$\ell(W, V) = \frac{1}{2}||v_1^T W||^2 + \frac{1}{2}||v_2^T W||^2 \ = \frac{1}{2}||v_{11}w_1 \ + v_{12}w_2||^2 + \frac{1}{2}||v_{21}w_1 \ + v_{22}w_2||^2$$

Hessian (1st block-row):

	$w_{1,1}$	$w_{1,2}$	$w_{2,1}$	$w_{2,2}$	$v_{1,1}$	$v_{1,2}$	$v_{2,1}$	$v_{2,2}$
$w_{1,1}$								
$w_{1,2}$								

Intuition: Example 3

- **Example 3: Multi-input-multi-output (SIMO):**



Denote $W = \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix} \in R^{2\times 2}$, $V = \begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix} \in R^{2\times 2}$

$w_1 = (w_{1,1}\,,\; w_{2,2}\,),$

$w_2 = (w_{2,1}\,,\; w_{2,2}\,),$

$v_1 = (v_{1,1}, v_{1,2}),$

$v_2 = (v_{2,1}, v_{2,2})$

$$\ell(W,V) = \;||VW||_F^2$$

(2-layer NN, X = Identity, Y = 0, no activation)

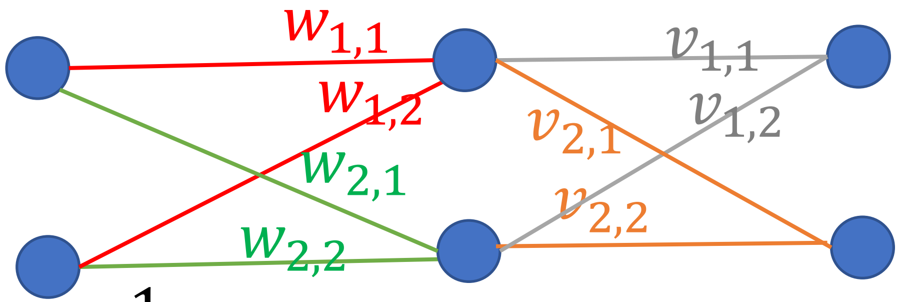
$$\ell(W,V) = \frac{1}{2}||v_1^T W||^2 + \frac{1}{2}||v_2^T W||^2 \;= \frac{1}{2}||v_{11}w_1 \; + v_{12}w_2||^2 + \frac{1}{2}||v_{21}w_1 \; + v_{22}w_2||^2$$

Hessian (1st block-row):

	$w_{1,1}$	$w_{1,2}$	$w_{2,1}$	$w_{2,2}$	$v_{1,1}$	$v_{1,2}$	$v_{2,1}$	$v_{2,2}$
$w_{1,1}$								
$w_{1,2}$								

Remark: here, the white box might not be strictly zero due to the cross-term, but the signal would be rather weak (indirect multiplicative relation)

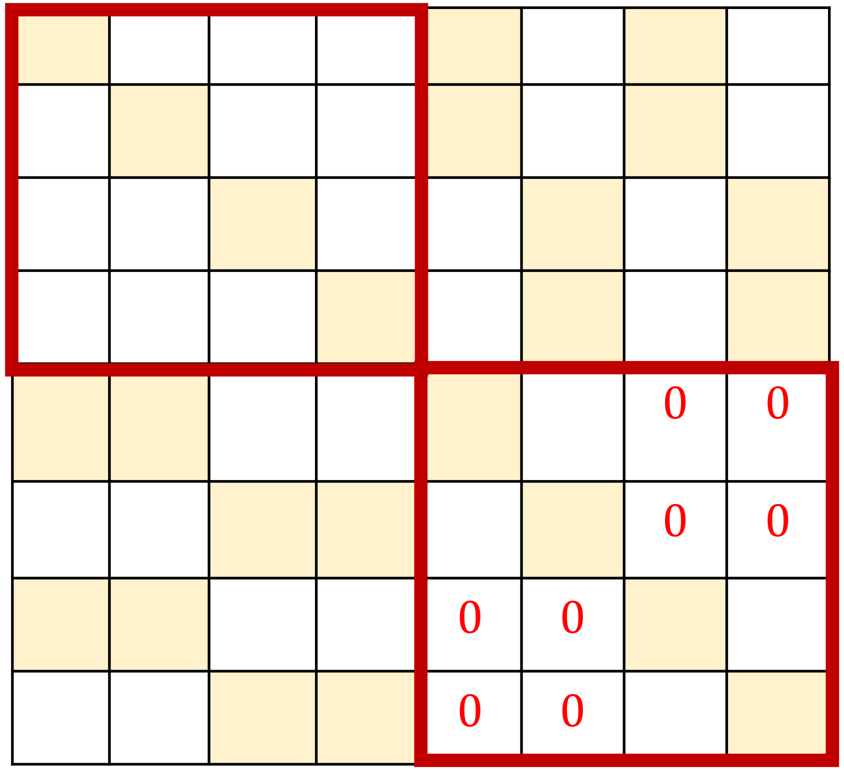
Intuition: Example 3



$$\ell(W,V) = ||VW||_F^2$$

$$\ell(W,V) = \frac{1}{2}||v_1^T W||^2 + \frac{1}{2}||v_2^T W||^2 = \frac{1}{2}||v_{11}w_1 + v_{12}w_2||^2 + \frac{1}{2}||v_{21}w_1 + v_{22}w_2||^2$$

Hessian:
(roughly estimated)



Just check the links!

Remark: here, the white box might not be strictly zero due to the cross-term, but the signal would be rather weak (indirect multiplicative relation)

Summarize so far

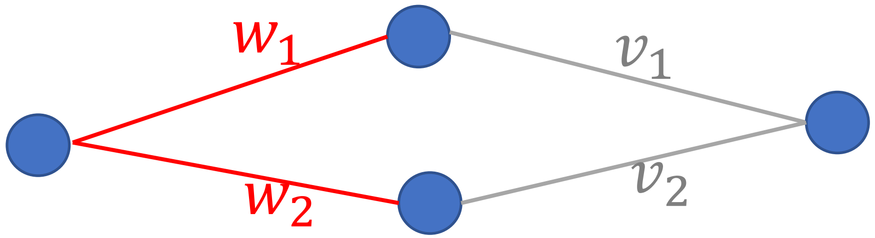
- **The special Hessian structure (partly) stems from the definition of matrix product**

$$f(v^T w) = f(v_1 \cdot w_1 + v_2 \cdot w_2), \qquad w, v \in R^2$$

Multiplicative relation

$\boldsymbol{w_1}$ **and** v_1 **are connected in the graph**

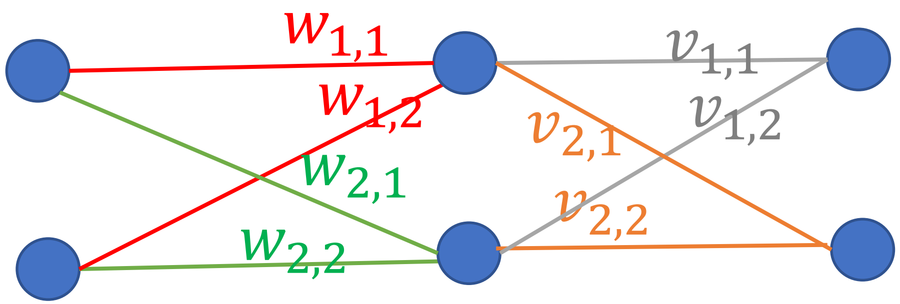
non-zero Hessian entry



Summarize so far

- **The special Hessian structure (partly) stems from the definition of matrix product**

$$f(VW) = f(v_1^T W + v_2^T W) = f(v_{11}w_1 + v_{12}w_2 + v_{21}w_1 + v_{22}w_2)$$



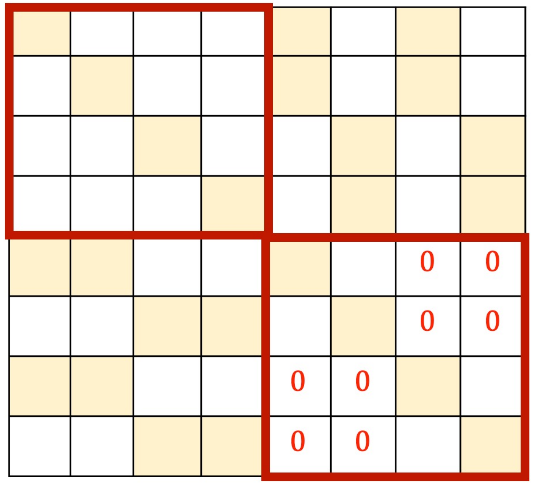
Multiplicative relation

↓

w_1 **and** $v_{1,1}$ **are connected in the graph**

↓

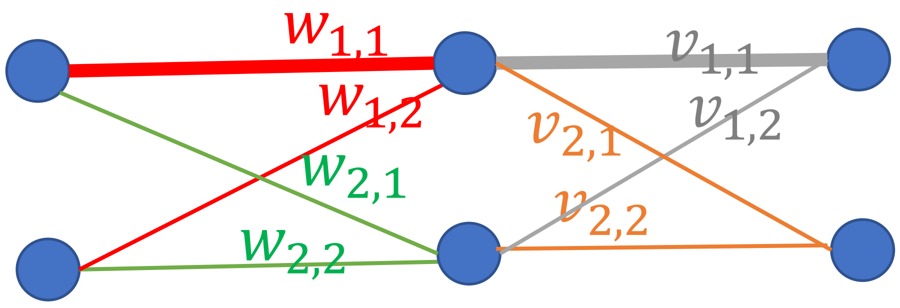
non-zero Hessian entry



Summarize so far

- **The special Hessian structure (partly) stems from the definition of matrix product**

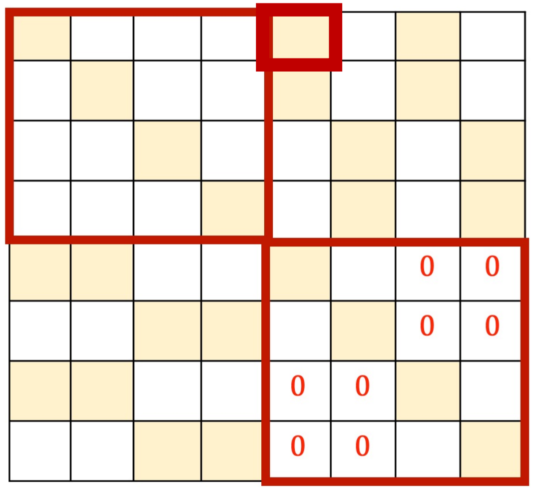
$$f(VW) = f(v_1^T W + v_2^T W) = f(v_{11} w_1 + v_{12} w_2 + v_{21} w_1 + v_{22} w_2)$$



Multiplicative relation

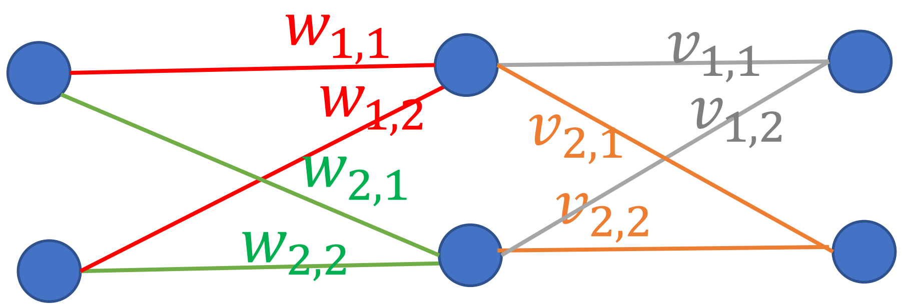
w_1 and $v_{1,1}$ are connected in the graph

non-zero Hessian entry

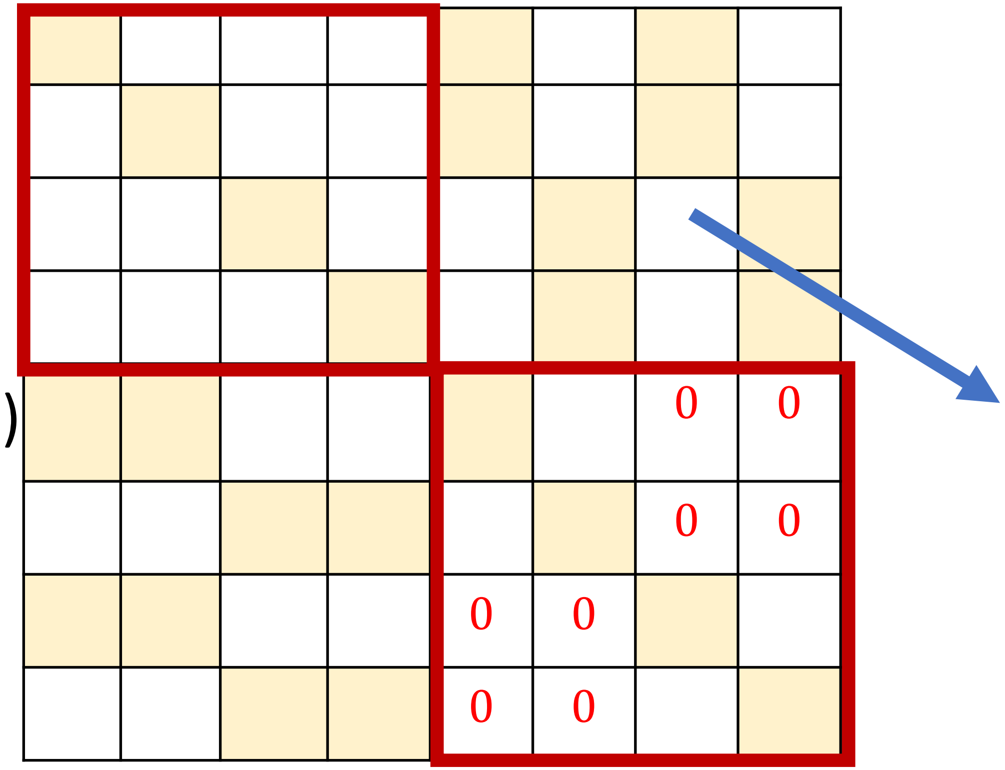


Lesson: learn to check the link in computational graph!

Summarize so far



Hessian:
(roughly
estimated)



We now roughly understand the pattern, but not enough

Q: what about non-linearity? (relu + CE)

Q: Why does large C help?

Q: Why the circulant pattern disappear along training?

Q: Are the white box provably small?

A: Linear algebra might not be enough…😢
Need helps from probability (next part)

Contents

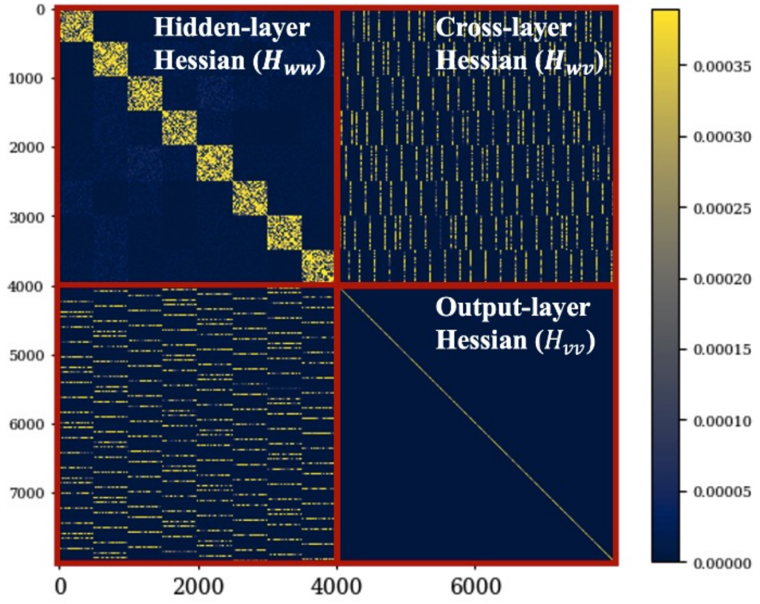
Intuition from probability: the non-linear NNs

Previously, for linear NNs: we discussed why “block-circulant-block-diag structure exists”
Now let‘s move to non-linear NNs (relu + CE loss)

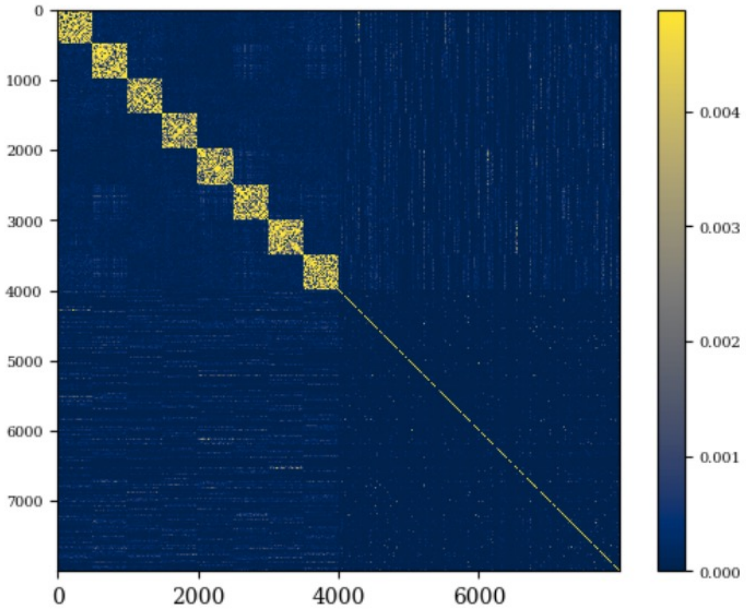
We reveal two forces:

- Force I: a ``static force'' rooted in the architecture design;
- Force II: and a ``dynamic force'' arisen from training.

Let us start with the "dynamic force"



(a) Hessian at initialization



(f) Hessian at 100% steps

Training eliminates the block-circulant structure in H_{wv}. Why?

Intuition from probability: the "dynamic force"

$$\min_{W \in R^{m\times d}, V \in R^{m \times C}} \frac{1}{N}\sum_n \ell(f(x_n), y_n) = \min_{W \in R^{m\times d}, V \in R^{m \times C}} \frac{1}{N}\sum_n -\log \frac{e^{\sigma(Wx_n)^T v_{y_n}}}{\sum_c e^{\sigma(Wx_n)^T v_c}}$$

$$\frac{\partial \ell}{\partial w_i} = -\frac{1}{N}\sum_n \sum_c (\delta_{y_n,c} - p_{n,c}) v_{c,i} \mathbb{1}(w_i^T x_n \geq 0) x_n \in R^d$$

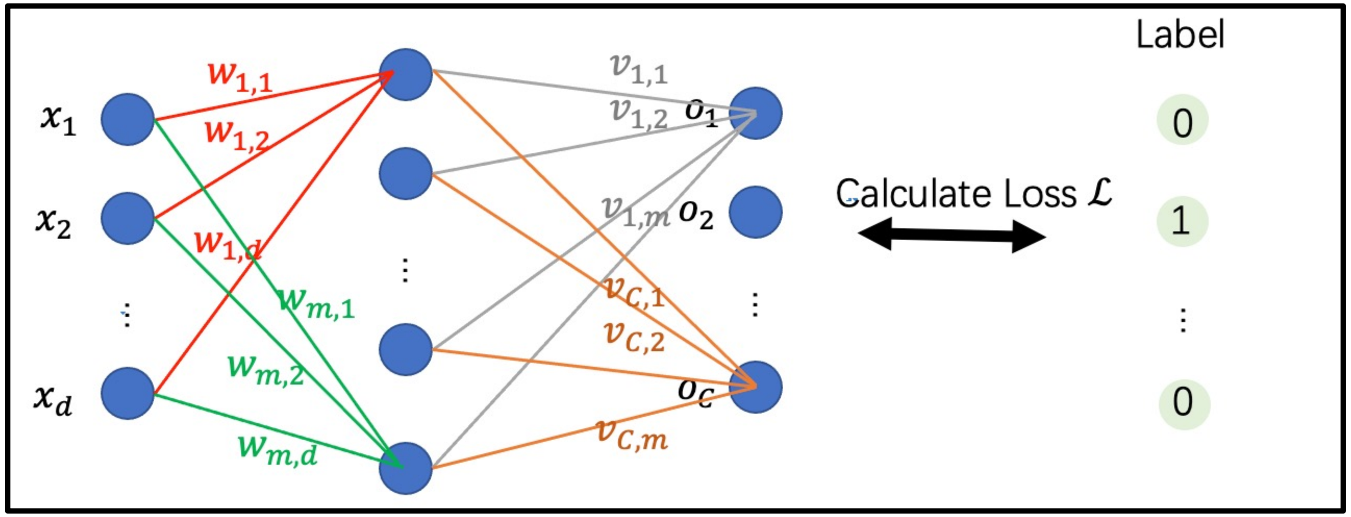
$$H_{w_i v_j} = \frac{\partial^2 \ell}{\partial w_i \, \partial v_j^T} = \begin{bmatrix} 0 & \cdots & a_{i,1} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{i,d} & 0 & \cdots & 0 \end{bmatrix} + O\left(\frac{1}{c}\right) \in R^{d \times m},$$

only the i-th column is non-zero (if ignoring the $+O\left(\frac{1}{c}\right)$ noise)

where $a_{i,d'} = -\frac{1}{N}\sum_n \sum_c (\delta_{y_n,c} - p_{n,c}) v_{c,i} \mathbb{1}(w_i^T x_n \geq 0) x_{n,d'}$

- **Remark:** as training goes on, we have : $p_{n,c} \to 1$ for $c = y_n$, $p_{n,c} \to 0$ for $c \neq y_n$ ⇒ Therefore, $(\delta_{y_n,c} - p_{n,c}) \to 0$ along training
- This can explain the ``dynamic force'': how the "block-circulant" pattern vanishes along training

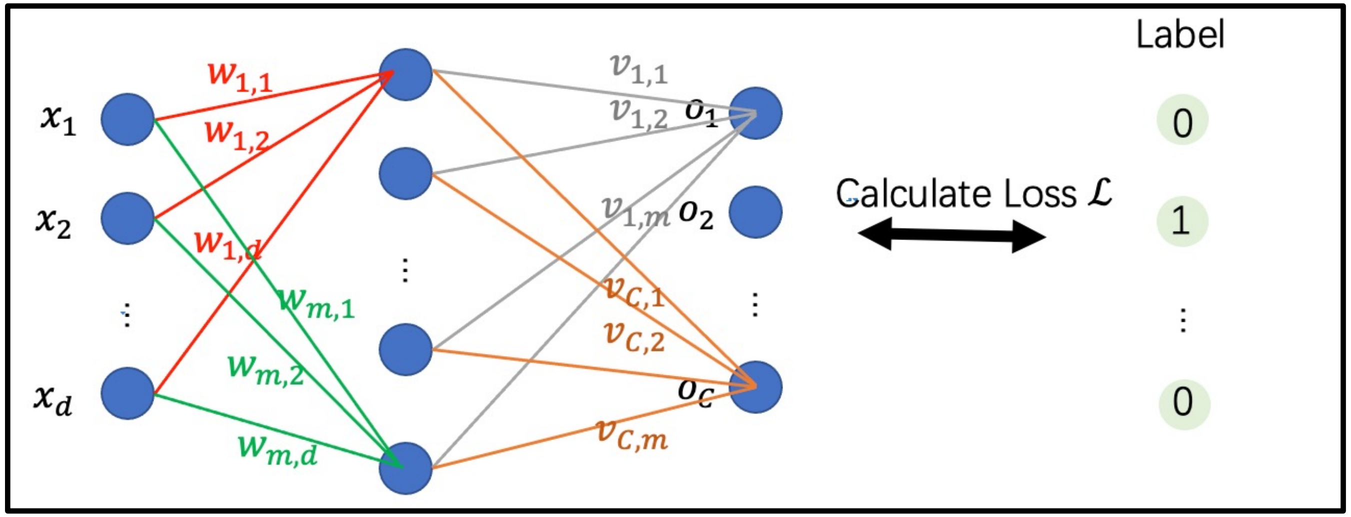
Linear algebra & probability : the "dynamic force"



$$H_{w_i v_j} = \frac{\partial^2 \ell}{\partial w_i \, \partial v_j^T} = \begin{bmatrix} 0 & \cdots & a_{i,1} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{i,d} & 0 & \cdots & 0 \end{bmatrix} + O\left(\frac{1}{c}\right) \in R^{d\times m}, \text{ where } a_{i,d'} = -\frac{1}{N}\sum_n \sum_c (\delta_{y_n,c} - p_{n,c}) v_{c,i} \mathbb{1}(w_i^T x_n \geq 0) x_{n,d'}$$

- **Linear algebra perspective (like previous part):**
  from computation graph, only $v_{1,i}, \cdots, v_{C,i}$ are linked to w_i
  So only i-th column in $H_{w_i v_j}$ is non-zero

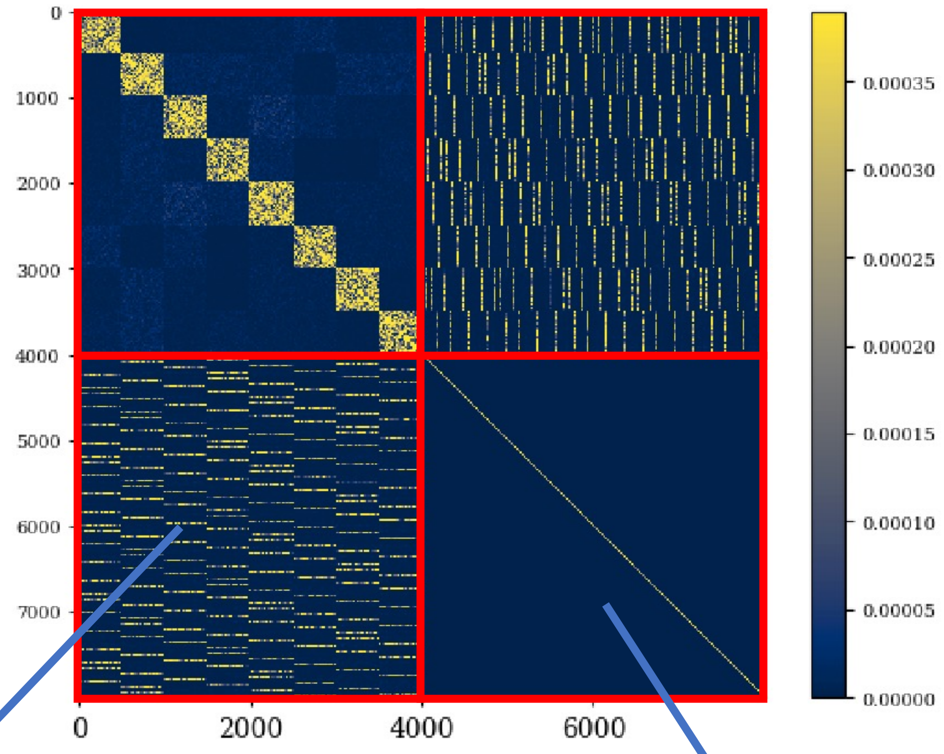
Linear algebra & probability : the "dynamic force"



$$H_{w_i v_j} = \frac{\partial^2 \ell}{\partial w_i \, \partial v_j^T} = \begin{bmatrix} 0 & \cdots & a_{i,1} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & a_{i,d} & 0 & \cdots & 0 \end{bmatrix} + O\left(\frac{1}{c}\right) \in R^{d \times m}, \text{ where } a_{i,d'} = -\frac{1}{N} \sum_n \boxed{\sum_c (\delta_{y_n,c} - p_{n,c})} v_{c,i} \, \mathbb{I}(w_i^T x_n \geq 0) x_{n,d'}$$

Key take-away: $H_{wv} \approx O(optimality\ gap)$, which are expected to vanish (experiments: vanishes quickly as training begins)

What about the “static force”?



Hessian at initialization with CE loss

Training eliminates
“block-circulant” in H_{wv}: **(previous parts)**

H_{ww} and H_{vv} seems always “block-diag”:
(More involved, see next slides)

Case 1: linear model + MSE loss

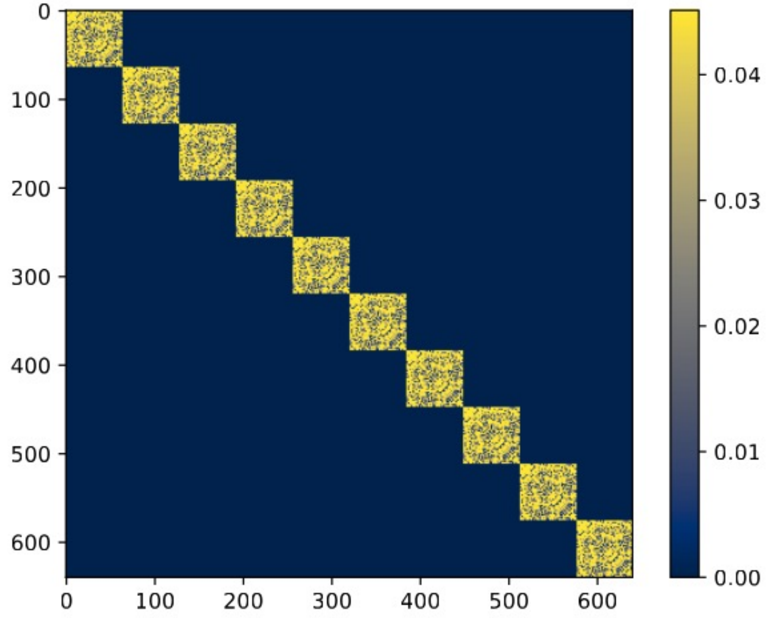
$$\min_{V} \ell_{\text{MSE}}(V) := \frac{1}{N}\sum_{n=1}^{N} \|Vx_n - y_n\|_2^2,$$

$$\begin{cases} \dfrac{\partial^2 \ell_{\text{MSE}}(V)}{\partial v_i \partial v_i^\top} = \frac{1}{N}\sum_{n=1}^{N} x_n x_n^\top \\ \dfrac{\partial^2 \ell_{\text{MSE}}(V)}{\partial v_i \partial v_j^\top} = 0_{d\times d}. \end{cases} \quad \text{for } i, j \in [C]$$

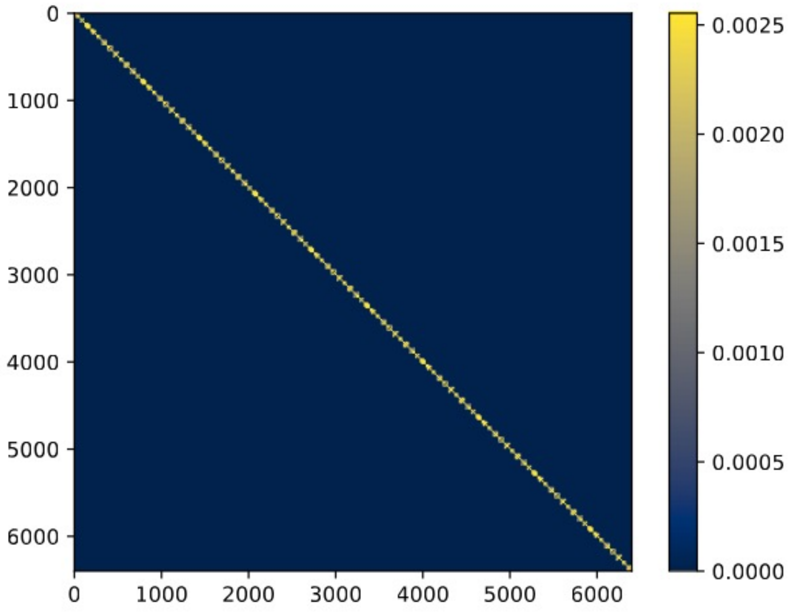
Hessian in Case 1 is trivially block diagonal
We will not discuss this case in the sequel

Case 1: linear model + MSE loss

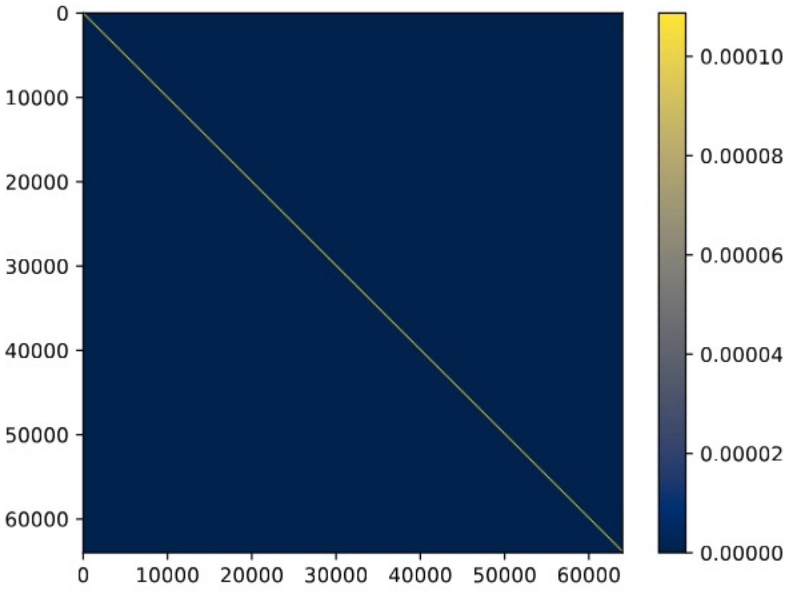
$$\min_{V} \ell_{\mathrm{MSE}}(V) := \frac{1}{N}\sum_{n=1}^{N} \|Vx_n - \mathcal{Y}_n\|_2^2,$$



(a) $C = 10$



(b) $C = 100$



(c) $C = 1000$

Case 2: linear model + CE loss

$$\min_V \ell_{\mathrm{CE}}(V) := -\frac{1}{N}\sum_{n=1}^{N}\log\left(\frac{\exp(v_{y_n}^\top x_n)}{\sum_{c=1}^{C}\exp(v_c^\top x_n)}\right).$$

Define $p_{n,i} := \exp(v_i^\top x_n) / \left(\sum_{c=1}^{C}\exp(v_c^\top x_n)\right)$. The Hessian matrix is, for $i, j \in [C]$.

$$\begin{cases} \frac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_i^\top} = \frac{1}{N}\sum_{n=1}^{N} p_{n,i}(1-p_{n,i}) x_n x_n^\top \\ \frac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_j^\top} = -\frac{1}{N}\sum_{n=1}^{N} p_{n,i} p_{n,j} x_n x_n^\top. \end{cases}$$

Intuitive understanding: at random initialization, suppose each entry in V follows i.i.d. zero-mean Gaussian distribution, we have $p_{n,i} \approx \frac{1}{C}$ for all $n \in [N], i \in [C]$. As such:

$$\frac{\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_j^\top}\right\|_{\mathrm{F}}}{\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_i^\top}\right\|_{\mathrm{F}}} \approx \frac{\sum_{n=1}^{N} p_{n,i} p_{n,j}}{\sum_{n=1}^{N} p_{n,i}(1-p_{n,i})} \approx \frac{\frac{1}{C^2}}{\frac{1}{C}\left(1-\frac{1}{C}\right)} = \frac{1}{C-1}, \tag{6}$$

which pushes the Hessian to become block-diagonal as $C \to \infty$.

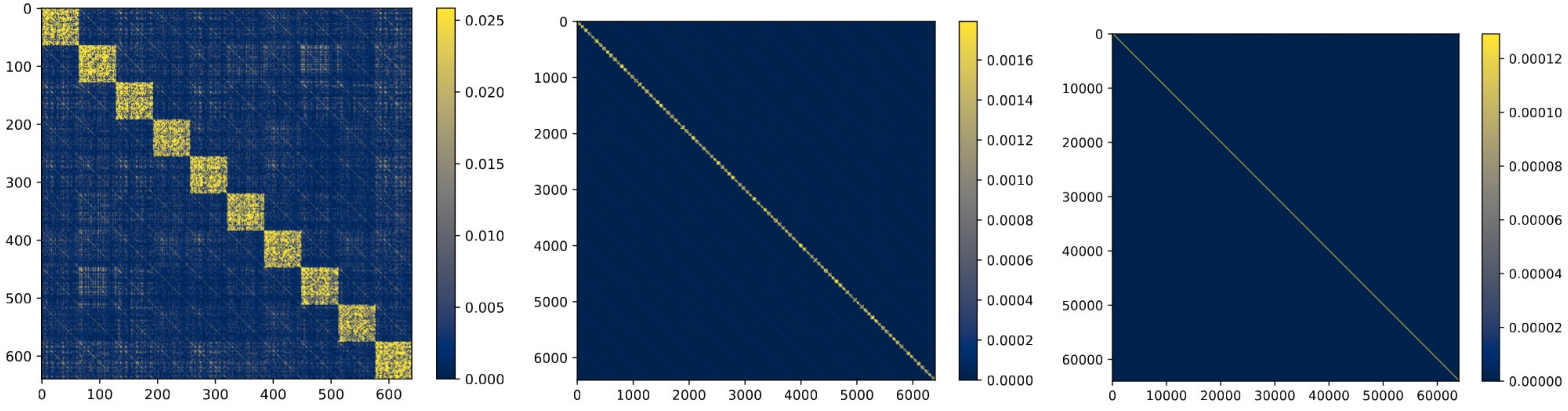
This is why large # class C helps!

Case 2: linear model + CE loss

$$\min_V \ell_{\mathrm{CE}}(V) := -\frac{1}{N}\sum_{n=1}^{N} \log\left(\frac{\exp(v_{y_n}^\top x_n)}{\sum_{c=1}^{C}\exp(v_c^\top x_n)}\right).$$

Define $p_{n,i} := \exp(v_i^\top x_n) / \left(\sum_{c=1}^{C}\exp(v_c^\top x_n)\right)$. The Hessian matrix is, for $i, j \in [C]$.

$$\begin{cases} \dfrac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_i^\top} = \frac{1}{N}\sum_{n=1}^{N} p_{n,i}(1-p_{n,i}) x_n x_n^\top \\ \dfrac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_j^\top} = -\frac{1}{N}\sum_{n=1}^{N} p_{n,i} p_{n,j} x_n x_n^\top . \end{cases}$$



(d) $C = 10$ (e) $C = 100$ (f) $C = 1000$

Case 3: 1-hidden-layer-NN with m neurons + MSE loss

$$\min_{W,V} \ell_{\text{MSE}}(W,V) := \frac{1}{N}\sum_{n=1}^{N} \|V\sigma(Wx) - \mathcal{Y}_n\|_2^2,$$

The hidden-layer Hessian H_{ww} is: for $i, j \in [m]$,

$$\begin{cases} \frac{\partial^2 \ell_{\text{MSE}}(W,V)}{\partial w_i \partial w_i^\top} = \frac{1}{N}\left(\sum_{c=1}^{C} v_{c,i}^2\right)\left(\sum_{n=1}^{N} \mathbf{1}(w_i^\top x_n > 0) x_n x_n^\top\right) \\ \frac{\partial^2 \ell_{\text{MSE}}(W,V)}{\partial w_i \partial w_j^\top} = \frac{1}{N}\left(\sum_{c=1}^{C} v_{c,i} v_{c,j}\right)\left(\sum_{n=1}^{N} \mathbf{1}(w_i^\top x_n > 0)\mathbf{1}(w_j^\top x_n > 0) x_n x_n^\top\right). \end{cases}$$

The output-layer Hessian H_{vv} is: for $i, j \in [C]$,

$$\begin{cases} \frac{\partial^2 \ell_{\text{MSE}}(W,V)}{\partial v_i \partial v_i^\top} = \frac{1}{N}\sum_{n=1}^{N} \sigma(Wx_n)\sigma(Wx_n)^\top \\ \frac{\partial^2 \ell_{\text{MSE}}(W,V)}{\partial v_i \partial v_j^\top} = 0_{d\times d}, \end{cases}$$

Case 3: 1-hidden-layer-NN with m neurons + MSE loss

Intuitive understanding: at random initialization, suppose entries in $v_i \in \mathbb{R}^d$ follow an i.i.d. zero-mean Gaussian distribution, then

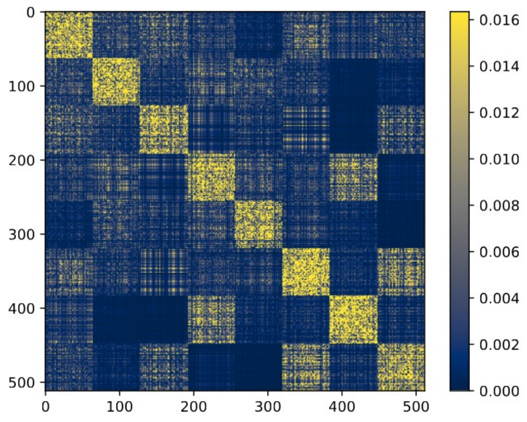
$$\frac{\left\| \frac{\partial^2 \ell_{\text{MSE}}(W,V)}{\partial w_i \partial w_j^\top} \right\|_{\text{F}}}{\left\| \frac{\partial^2 \ell_{\text{MSE}}(W,V)}{\partial w_i \partial w_i^\top} \right\|_{\text{F}}} \approx \frac{\left(\sum_{c=1}^{C} v_{c,i} v_{c,j}\right)}{\left(\sum_{c=1}^{C} v_{c,i}^2\right)} \overset{C \to \infty}{=} \frac{\text{Cov}(v_{i,i}, v_{i,j})}{\text{Var}(v_{i,i})}. \tag{10}$$

Since $v_{i,i}, v_{i,j}$ are independent, $\text{Cov}(v_{i,i}, v_{i,j}) = 0$ and thus the block-diagonal structure occurs as $C \to \infty$.

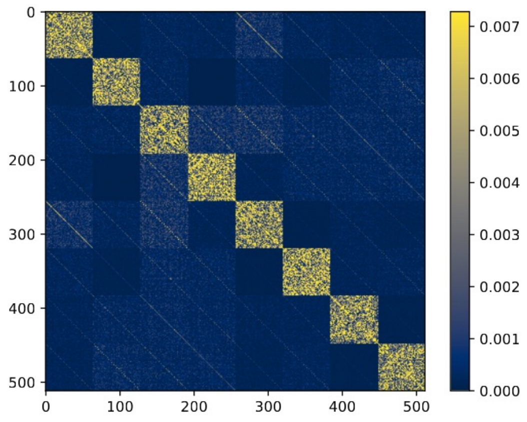
This is why large # class C helps!

Case 3: 1-hidden-layer-NN with m neurons + MSE loss

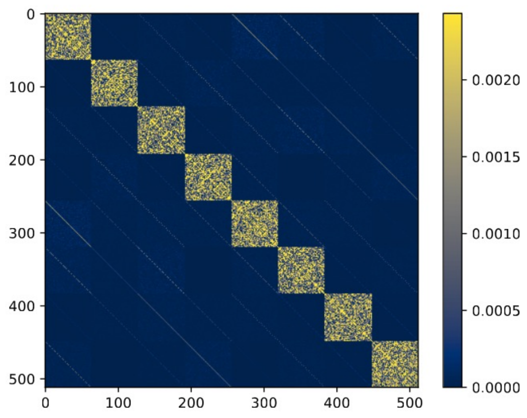
Hidden-weight Hessian:



(a) $C = 10$

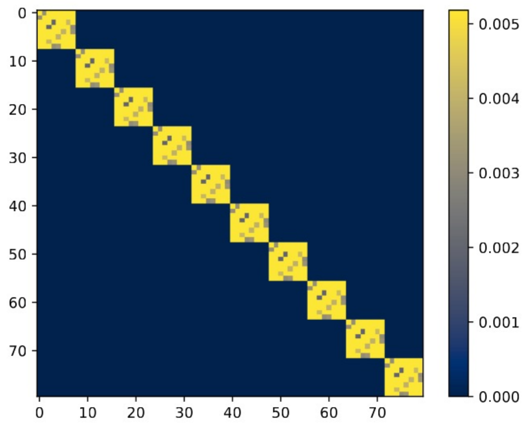


(b) $C = 100$

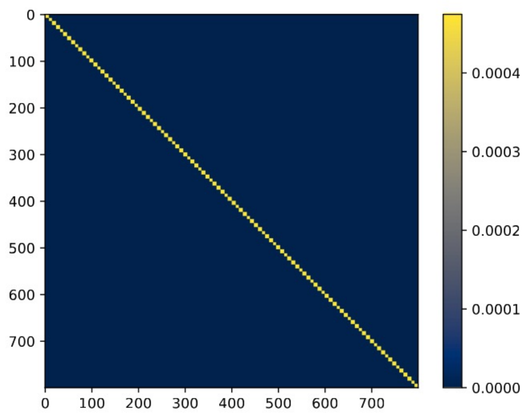


(c) $C = 1000$

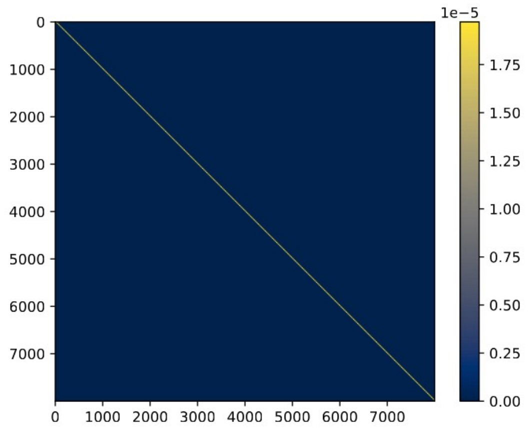
Output-weight Hessian:



(d) $C = 10$



(e) $C = 100$



(f) $C = 1000$

Case 4: 1-hidden-layer-NN with m neurons + CE loss

$$\min_{W,V} \ell_{\mathrm{CE}}(W,V) := -\frac{1}{N}\sum_{n=1}^{N} \log\left(\frac{\exp(v_{y_n}^\top \sigma(Wx_n))}{\sum_{c=1}^{C}\exp(v_c^\top \sigma(Wx_n))}\right).$$

The Hessian matrix for the hidden weights is: for $i, j \in [m]$,

$$\begin{cases} \frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_i \partial w_i^\top} = \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{c=1}^{C} p_{n,c} v_{c,i}^2 - \left(\sum_{c=1}^{C} p_{n,c} v_{c,i}\right)^2\right)\mathbf{1}(w_i^\top x_n > 0) x_n x_n^\top \\ \frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_i \partial w_j^\top} = \frac{1}{N}\sum_{n=1}^{N}\left(\sum_{c=1}^{C} p_{n,c} v_{c,i} v_{c,j} - \left(\sum_{c=1}^{C} p_{n,c} v_{c,i}\right)\left(\sum_{c=1}^{C} p_{n,c} v_{c,j}\right)\right)\mathbf{1}(w_i^\top x_n > 0)\mathbf{1}(w_j^\top x_n > 0) x_n x_n^\top \end{cases} \tag{12}$$

The Hessian matrix for the output weights is: for $i, j \in [C]$,

$$\begin{cases} \frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial v_i \partial v_i^\top} = \frac{1}{N}\sum_{n=1}^{N} p_{n,i}(1 - p_{n,i})\sigma(W x_n)\sigma(W x_n)^\top \\ \frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial v_i \partial v_j^\top} = -\frac{1}{N}\sum_{n=1}^{N} p_{n,i} p_{n,j}\sigma(W x_n)\sigma(W x_n)^\top. \end{cases} \tag{13}$$

Case 4: 1-hidden-layer-NN with m neurons + CE loss

Intuitive understanding: at random initialization, suppose entries in W, V follows i.i.d. zero-mean Gaussian distribution, we have $p_{n,i} \approx \frac{1}{C}$ for all $n \in [N], i \in [C]$. As such:

$$\frac{\left\| \frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_i \partial w_j^\top} \right\|_{\mathrm{F}}}{\left\| \frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_i \partial w_i^\top} \right\|_{\mathrm{F}}} \approx \frac{\left(\sum_{c=1}^C v_{c,i} v_{c,j} - \left(\sum_{c=1}^C v_{c,i}\right)\left(\sum_{c=1}^C v_{c,j}\right)\right)/C}{\left(\sum_{c=1}^C v_{c,i}^2 - \left(\sum_{c=1}^C v_{c,i}\right)^2\right)/C} \overset{C\to\infty}{=} \frac{\mathrm{Cov}(v_{i,i}, v_{i,j})}{\mathrm{Var}(v_{i,i})}. \tag{14}$$

Since $v_{i,i}, v_{i,j}$ are independent, $\mathrm{Cov}(v_{i,i}, v_{i,j}) = 0$ and thus the block-diagonal structure occurs as $C \to \infty$. Similarly, we have

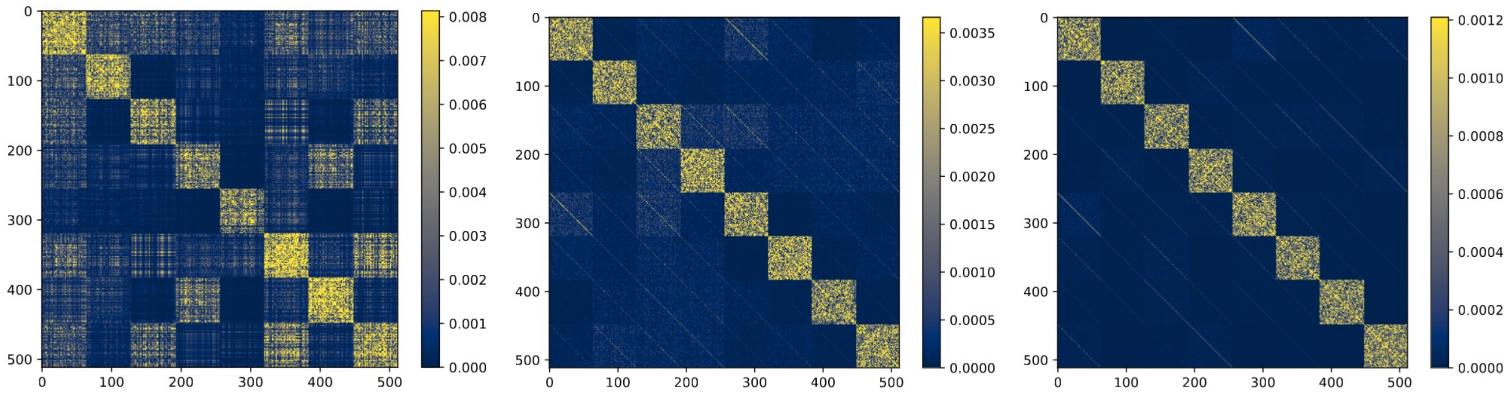
$$\frac{\left\| \frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial v_i \partial v_j^\top} \right\|_{\mathrm{F}}}{\left\| \frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial v_i \partial v_i^\top} \right\|_{\mathrm{F}}} \approx \frac{\sum_{n=1}^N p_{n,i} p_{n,j}}{\sum_{n=1}^N p_{n,i}(1-p_{n,i})} \approx \frac{\frac{1}{C^2}}{\frac{1}{C}\left(1-\frac{1}{C}\right)} = \frac{1}{C-1}, \tag{15}$$

and thus the block-diagonal structure arises as $C \to \infty$.

This is why large # class C helps!

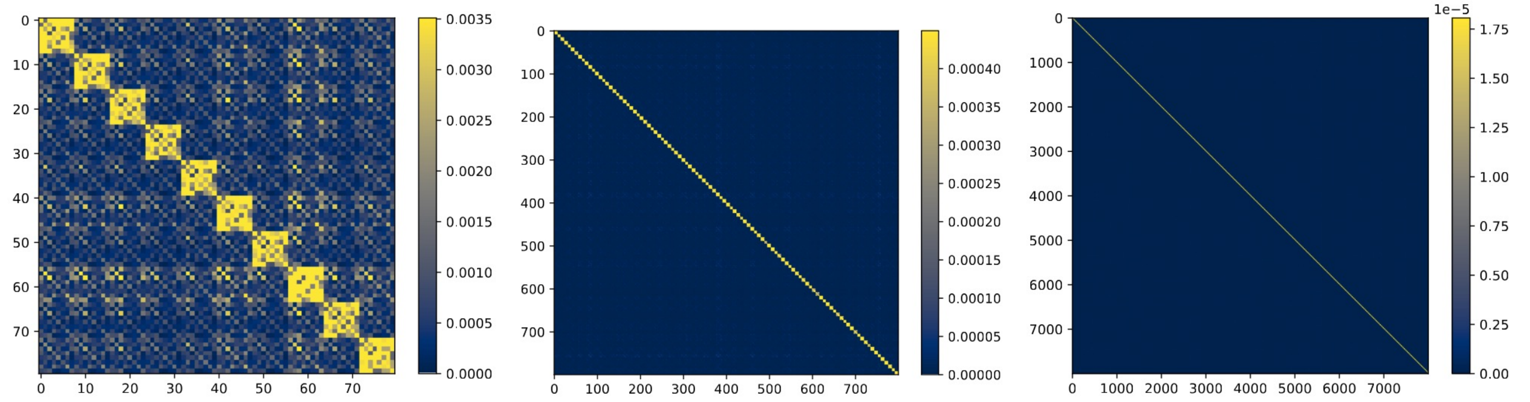
Case 4: 1-hidden-layer-NN with m neurons + CE loss

Hidden-weight Hessian:



(a) $C = 10$ (b) $C = 100$ (c) $C = 1000$

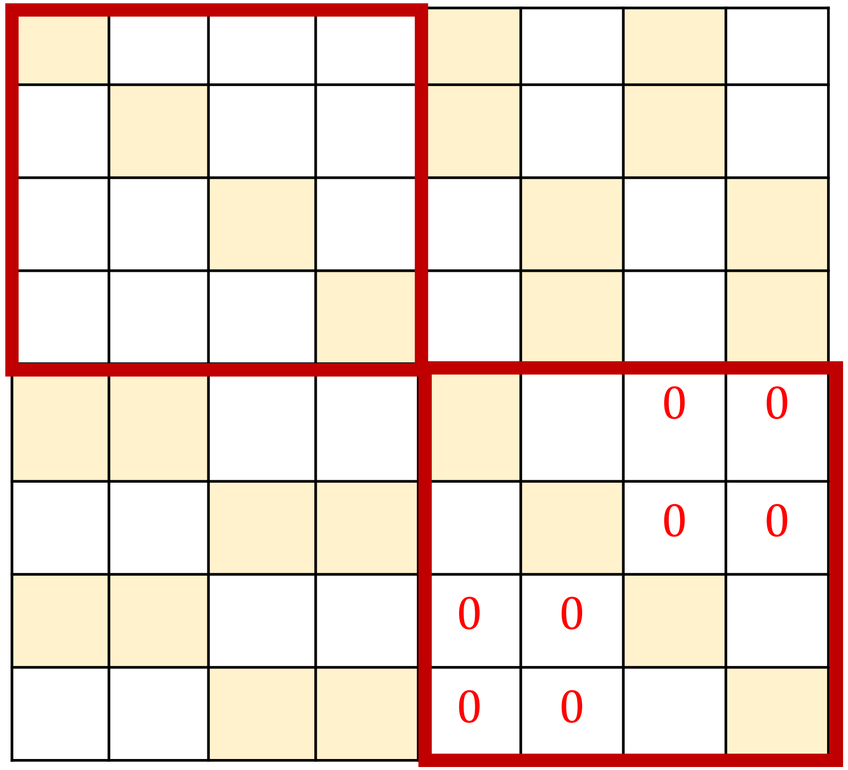
Output-weight Hessian:



(d) $C = 10$ (e) $C = 100$ (f) $C = 1000$

Summary: 3-level sources of block-diag structure

- **Level 1: definition of matrix product: many zeros, no links**



Summary: 3-level sources of block-diag structure

- **Level 1: definition of matrix product: many zeros, no links**
- **Level 2: #Class C goes to infinity: weaken many links in** H_{ww}, H_{vv}

Static force

	≈ 0	≈ 0	≈ 0		≈ 0		≈ 0
≈ 0		≈ 0	≈ 0		≈ 0		≈ 0
≈ 0	≈ 0		≈ 0	≈ 0		≈ 0	
≈ 0	≈ 0	≈ 0		≈ 0		≈ 0	
		≈ 0	≈ 0		≈ 0	0	0
≈ 0	≈ 0			≈ 0		0	0
		≈ 0	≈ 0	0	0		≈ 0
≈ 0	≈ 0			0	0	≈ 0	

Summary: 3-level sources of block-diag structure

- Level 1: definition of matrix product: many zeros, no links
- Level 2: #Class C goes to infinity: weaken many links in H_{ww}, H_{vv}

Static force

- Level 3: Training: eliminates strong links in H_{wv}

Dynamic force

	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
≈ 0		≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
≈ 0	≈ 0		≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
≈ 0	≈ 0	≈ 0		≈ 0	≈ 0	≈ 0	≈ 0
≈ 0	≈ 0	≈ 0	≈ 0		≈ 0	0	0
≈ 0	≈ 0	≈ 0	≈ 0	≈ 0		0	0
≈ 0	≈ 0	≈ 0	≈ 0	0	0		≈ 0
≈ 0	≈ 0	≈ 0	≈ 0	0	0	≈ 0	

- But how to prove rigorously?
- Need tools from Random Matrix Theory (RMT)

Contents

Overview of our results

Settings: Consider general C-class classification problem: $(\mathrm{x_n}, \mathrm{y_n})_{n=1}^{N}, x_n \in R^d, y_n \in \{1,2,\cdots,C\}$

We prove the following results (informal): when $N, d \to \infty$ with $\frac{d}{N} = \gamma$, we have

- **Case 1** (linear model + MSE loss):
  For any C, Hessian is strictly block-diag with C blocks

- **Case 2** (linear model + CE loss):
  Hessian approaches block-diag with C blocks with rate $O(1/C)$

- **Case 3** (1-hidden-layer-NN with m neurons + MSE loss):
  - Hessian of hidden weights approach block-diag with m blocks with rate $O(1/\sqrt{C})$
  - Hessian of output weights approach block-diag with C blocks with rate $O(1/C)$

- **Case 4** (1-hidden-layer-NN with m neurons + CE loss):
  - Hessian of hidden weights approach block-diag with m blocks with rate $O(1/\sqrt{C})$
  - Hessian of output weights approach block-diag with C blocks with rate $O(1/C)$

Main Results

Assumption 1 *The entries of the data matrix* $X_N = (x_1, \cdots, x_N) \in \mathbb{R}^{d\times N}$ *are i.i.d.* $\mathcal{N}(0,1)$.

Assumption 2 *The model weights in* W *and* V *are initialized by LeCun initialization. That is: for the linear model,* $V_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \frac{1}{d})$, $i \in [C], j \in [d]$; *for 1-hidden-layer network,* $W_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \frac{1}{d})$, $i \in [m], j \in [d]$, $V_{i,j} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \frac{1}{m})$, $i \in [C], j \in [m]$.

Remark:

- **Assumption 1** on data distribution is standard in random matrix theory (Pastur, 2020)
- It is possible to extend the Gaussian X_N to, e.g., Gaussian orthogonal ensembles and more general distribution
- However, such generalization is non-trivial and each case may require an independent paper (e.g. Pastur (2022); Pastur and Slavin (2023))

Main Results (Simplified)

Theorem 1 **(Linear models.)** *Consider the Hessian expressions in (5) and assume Assumptions 1 and 2 hold. Suppose* $d, N \to \infty$, $\frac{d}{N} \to \gamma \in (0, +\infty)$, *then for fixed* $C \geq 2$, *it holds almost surely*

$$\lim_{d,N\to\infty} \frac{\left\| \frac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_j^\top} \right\|_{\mathrm{F}}^2}{\left\| \frac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_i^\top} \right\|_{\mathrm{F}}^2} = \frac{g_{ij}(\gamma, C)}{g_{ii}(\gamma, C)}, \quad \lim_{C\to\infty} \frac{C^2 g_{ij}(\gamma, C)}{g_{ii}(\gamma, C)} = \frac{\gamma e^2 + 1}{\gamma e + 1}. \tag{20}$$

When $C \to \infty$, the ratio vanishes at the rate $\mathcal{O}(1/C^2)$, and the block-diagonal structure emerges.

RK: We actually calculate the close form of F-norm for each block, not just their ratio (omitted here for cleanness)

Key messages from Theorem 1:
the block-diagonal structure arises when
classes C → ∞

Main Results (Simplified)

Theorem 2 **(1-hidden-layer networks.)** *Consider the Hessian expressions in (8) to (13), and assume Assumptions 1 and 2 hold. Then for any fixed* $m \geq 3$*, suppose* $d, N \to \infty$, $\frac{d}{N} \to \gamma \in (0, +\infty)$*, it holds that*

$$\lim_{d,N\to\infty} \frac{\mathbf{E}\left[\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_i \partial w_j^\top}\right\|_{\mathrm{F}}^2\right]}{\mathbf{E}\left[\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_i \partial w_i^\top}\right\|_{\mathrm{F}}^2\right]}, \quad \lim_{d,N\to\infty} \frac{\mathbf{E}\left[\left\|\frac{\partial^2 \ell_{\mathrm{MSE}}(W,V)}{\partial w_i \partial w_j^\top}\right\|_{\mathrm{F}}^2\right]}{\mathbf{E}\left[\left\|\frac{\partial^2 \ell_{\mathrm{MSE}}(W,V)}{\partial w_i \partial w_i^\top}\right\|_{\mathrm{F}}^2\right]}, \quad \lim_{d,N\to\infty} \frac{\mathbf{E}\left[\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial v_i \partial v_j^\top}\right\|_{\mathrm{F}}^2\right]}{\mathbf{E}\left[\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial v_i \partial v_i^\top}\right\|_{\mathrm{F}}^2\right]} \qquad (28)$$

vanish at the rate $\mathcal{O}(1/C)$, $\mathcal{O}(1/C)$, $\mathcal{O}(1/C^2)$, respectively, and the block-diagonal structure also emerges as C increases.

RK: We actually calculate the close form of F-norm for each block, not just their ratio (omitted here for cleanness)

Key messages from Theorem 1 & 2:
the block-diagonal structure arises when
classes C → ∞

Roadmap for the Proof

- **Part 3-1:** Some basics of **random matrix theory (RMT): useful for everyone**
  -- What is the goal of RMT?
  -- How is RMT different from classical probability theory?
  -- Introduction to Stieltjes Transform, Semicircular law, MP law

- **Part 3-2:** Hessian expressions and some challenges
  -- why existing RMT tools cannot directly apply

- **Part 3-3:** Our new methods to overcome the challenges
  -- based on some additional insights in Hessian of NNs
  -- Our method implements the Lindeberg Principle, which originally proposed to prove CLT

What is the Goal of RMT?

- **Goal:** RMT studies limit eigenvalue distribution of a random Hermitian A (denoted as μ_A) as its size approaches ∞
- **Def:** we define the eigenvalue distribution of $A \in R^{d \times d}$ as the normalized counting measure of eigenvalues:

$$\mu_A = \frac{1}{d} \sum_j \delta_{\lambda_j(A)}$$

- **A simple example:**

-- What we know before

Let $A = \frac{1}{N} \sum_n x_n x_n^T \in R^{d \times d}$, where $x_n \in R^d$ are i.i.d. standard Gaussian

Then for fixed size d, let $N \to \infty, A \to I_{d \times d}$ (Law of Large Number)

In other words, $\mu_A \to \delta_1$

-- What we might not know before:

What if the size of A increase to ∞?
RMT can answer this question: when $N, d \to \infty, N = \gamma d$,

then $\mu_A \to$ MP-law (γ)

Basic Question I: How to Define Convergence?

- Caveat: A is random, so λ_A is random, so μ_A is a random variable
- Comparison with classical probability:

$x \sim Bernoulli\left(\frac{1}{2}\right) \longrightarrow \mu_x = \frac{1}{2}\delta_0 + \frac{1}{2}\delta_1 \longrightarrow$ A deterministic measure

Random $A \longrightarrow \mu_A = \frac{1}{d}\sum_j \delta_{\lambda_j(A)} \longrightarrow$ A "random measure"! How to define convergence?

Classical prob:
How to define $\{x_n\}$ converges to x_∞?

$$\begin{array}{cccc} \mu_1 & \mu_{100} & \mu_{1000} & \mu_\infty \\ \uparrow & \uparrow & \uparrow & \uparrow \\ x_1 & x_{100} & x_{1000} & x_\infty \end{array}$$

$\longrightarrow$ **Def** (classical convergence of measure):
We say $\{\mu_n\}$ **weakly converge** to μ_∞
(or $\{x_n\}$ **converge in distribution** to x_∞)
if $\forall$ bdd continuous f, $\lim_{n\to\infty} \int f \, d\mu_n = \int f \, d\mu$

RMT:
How to define $\{A_n\}$ converges to A_∞?

$$\begin{array}{cccc} \mu_1 & \mu_{100} & \mu_{1000} & \mu_\infty \\ \uparrow & \uparrow & \uparrow & \uparrow \\ A_1 & x_{100} & x_{1000} & x_\infty \end{array}$$

$\longrightarrow$ **Def:** We say $\{\mu_n\}$ weakly converge a.s. to a deterministic μ_∞ if $\forall$ bdd continuous f:

$$\Pr\left(\lim_{n\to\infty} \int f \, d\mu_n = \int f \, d\mu\right) = 1$$

Basic Question II: How to characterize a distribution?

- In Classical prob, we learned characteristic function (Fourier Transform)

$$\phi(t) = E\left(e^{itx}\right) = \int_R \exp^{itx} \, d\mu(x)$$

- Another one: Steiltjes Transform (S-Transform), which also uniquely determines a prob measure μ

$$S_\mu(z) = \int_R \frac{1}{x - z} \, d\mu(x) \, , \forall z \in C^+ \backslash \mathrm{supp}(\mu)$$

- RMT usually uses $s_\mu(z)$ to recover μ

Theorem (Inversion formula): For any $a < b \in R$ and any probability measure μ, we have

$$\mu([a,b]) = \lim_{\epsilon \to 0^+} \frac{1}{\pi} \int_a^b Im\left(s_\mu(t + i\,\epsilon)\right) dt$$

Basic Question II: How to characterize a distribution?

- $S_\mu(z)$ can also help us to extract the moments of μ

Proposition 1: for any probability measure μ, we have

$S_\mu(z) = -\frac{1}{z} - \frac{m_1}{z^2} - \frac{m_2}{z^3} - \cdots, z \to \infty$, where $m_k = \int_R t^k \mu(t)$ is the k-th order moment of μ

Proof:

$\frac{1}{t-z} = -\frac{1}{z}\left(\frac{1}{1-\frac{t}{z}}\right) = -\frac{1}{z}\left(\sum_{k=0}^{\infty} \frac{t^k}{z^k}\right) = -\sum_{k=0}^{\infty} \frac{t^k}{z^{k+1}}$, when z is sufficiently large

$S_\mu(z) = \int_R \frac{1}{x-z}\, d\mu(x) = -\sum_{k=0}^{\infty} \frac{\int_R t^k d\mu(t)}{z^{k+1}} = -\frac{1}{z} - \frac{m_1}{z^2} - \frac{m_2}{z^3} - \cdots$. Q.E.D.

In our context: $||A||_F^2 = \text{2nd} - \text{order moment}$ of μ (sum-of-square eigenvalues)

Some Other Properties of Steiltjes Transform

Thm (Continuity theorem, deterministic version [1]): Let $\{\mu_n\}$ be a sequence of deterministic prob measures, then μ_n converges weakly to a prob measure μ_n if and only if

$$\lim_{n\to\infty} S_{\mu_n}(z) = S_\mu(z)$$

Thm (Continuity theorem, random version [1]): Let $\{\mu_n\}$ be a sequence of random prob measures, then μ_n converges weakly almost surely to a prob measure μ_n if and only if

$$\lim_{n\to\infty} S_{\mu_n}(z) = S_\mu(z)$$

Thm ([2]): for any sequence of Hermitian matrices $\{\mathrm{A_n} \in C^{n\times n}\}$, we have

$$\text{For any fixed } \mathrm{z} \in C^+,\ S_{\mu_{A_n}}(z) - \mathrm{E}\, S_{\mu_{A_n}}(z) \to a.s.\ \text{as } n \to \infty$$

Implications: to find μ, we just need to find $S_\mu(z)$ or $\mathrm{E}\, S_\mu(z)$

[1]: Jeff Yao, et al., Large Sample Covariance Matrices and High Dimensional Data Analysis

[2]: Jeff Yao, Lecture notes on the Wigner Semicirclar Law

Summarize so far

- We have discussed:

1. the difference between RMT and classical probability
2. the notion of convergence
3. Steiltjes Transform and properties

- Now, how to find the limit μ_A of a sequence of growing random matrices $\{A_n\}$

Pipeline in RMT:

- **Step 1:** Given the expression of a random matrix A_n, try to find the limit $S_{\mu_A}(z)$ (abbreviation: $S_A(z)$)
  [This step is not easy! Usually worth a top statistic paper if you can find $S_A(z)$ for a new class of A_n (either in explicit form or implicit equations)]

- **Step 2:** Recover μ from $S_A(z)$
  [This step largely based on experience. Has systematic strategies (e.g., Taylor expansion)]

- We now provide two classical examples

1. Semicircular law on A = Wigner matrices
2. M-P law on $\mathrm{A} = XX^T$

Semicircular Law of Wigner Matrices

- **Def:** $A_n = (a_{i,j})_{1\le i,j\le n}$ is called a Wigner Matrix if :
  1. A_n is Hermitian
  2. $a_{i,i}$ are i.i.d. real r.v.s. with unit variance
  3. $a_{i,j}, i > j$ are i.i.d. complex r.v.s with zero mean and unit variance

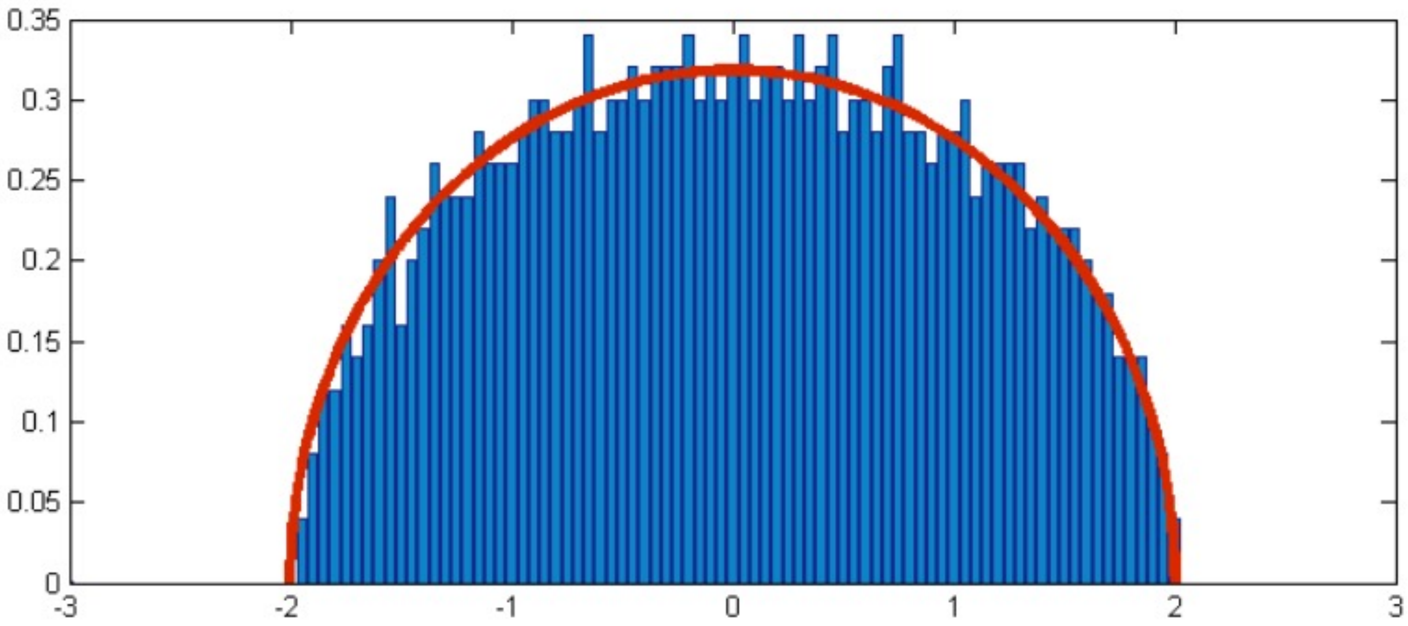
- **Thm (Semicircular law):** Consider normalized Wigner matrices $\widetilde{\boldsymbol{A_n}} = \frac{1}{\sqrt{n}} A_n$, then $\mu_{\widetilde{\boldsymbol{A_n}}}$ converges weakly a.s. to Wigner semicircular distribution:

$$\mu_{SC} := \frac{1}{2\pi} (4 - |x|^2)_+^{\frac{1}{2}} dx$$

- **Proof:** >5 pages, see [3], omitted here

[3]: Tao, Terence. *Topics in random matrix theory*

Semicircular Law of Wigner Matrices



Eigenvalue histogram of Wigner A_n , $n = 1000$, 1000 samples of A_n
Red curve: density of Semicircular distribution

MP Law of Wigner Matrices

Thm (Marchenko–Pastur 1967): Let $X \in R^{d\times n}$ whose entries are i.i.d. zero mean and variance $\sigma^2 < \infty$. Let $A_n = \frac{1}{n} XX^T \in R^{d\times d}$. Assume $n, d \to \infty$ and $\frac{d}{n} = \lambda > 0$, then μ_{A_n} a.s. weakly converges to μ_{MP} , where for any subset Ω in R, we have

$$\mu_{MP}(\Omega) = \begin{cases} \left(1 - \frac{1}{\lambda}\right) 1(0 \in \Omega) + \nu(\Omega), & if\ \lambda > 1 \\ \nu(\Omega), & if\ 0 \le \lambda \le 1 \end{cases}$$

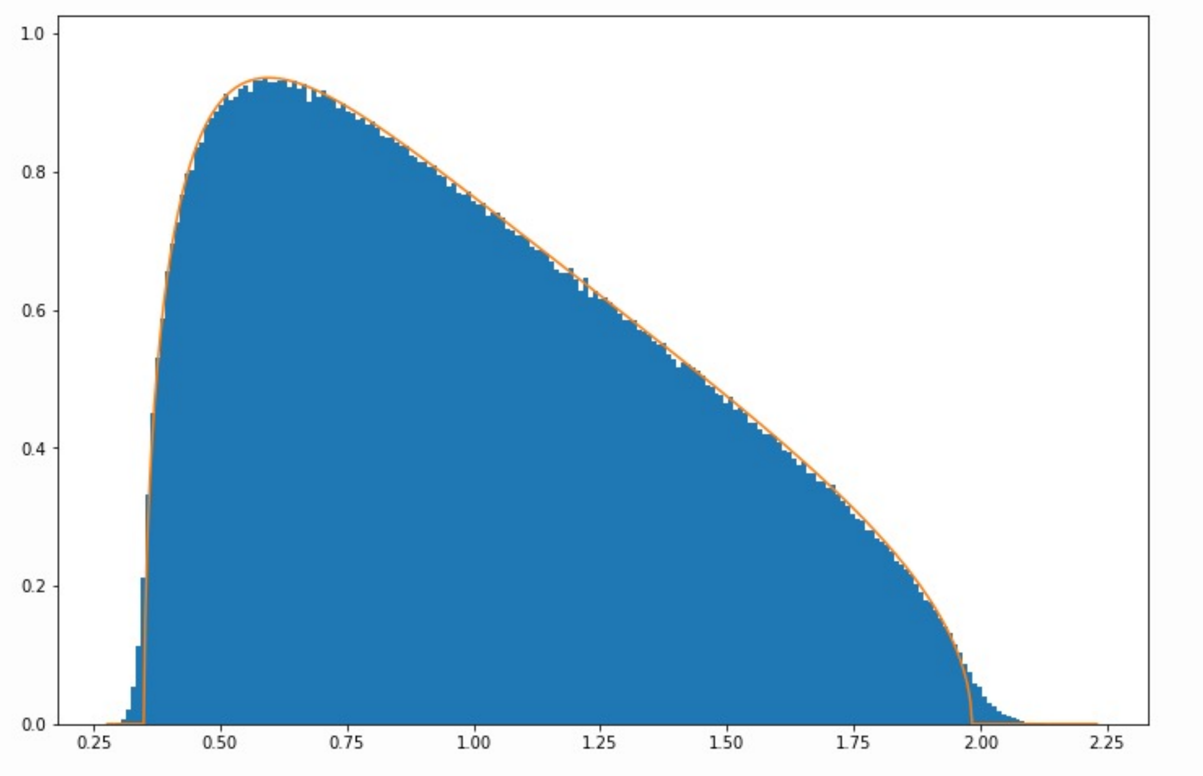
and

$$d\nu(x) = \frac{1}{2\pi\sigma^2} \frac{\sqrt{(\lambda_+ - x)(x - \lambda_-)}}{\lambda x} \mathbf{1}_{x\in[\lambda_-,\lambda_+]}\, dx$$

with

$$\lambda_\pm = \sigma^2(1 \pm \sqrt{\lambda})^2.$$

MP Law of Covariance Matrices



Eigenvalue histogram of $A_n = \frac{1}{n} XX^T \in R^{d \times d}$, $n = 50, d = 300$, 1000 samples of A_n

Yellow curve: density of MP distribution with d/ n = 50 / 300

Now, we are ready for our proof

- Now we discuss the technical challenges for the Hessian in **Case 2 (linear model + CE loss)**
- **Proof Procedure:**

  1. Find diagonal block $||\frac{\partial^2 \ell_{CE}(V)}{\partial v_i \partial v_i^T}||_F$ and off-diagonal block $||\frac{\partial^2 \ell_{CE}(V)}{\partial v_i \partial v_j^T}||_F$ when $N, d \to \infty$
  2. Compare their ratio
- We only discuss the diagonal blocks $||\frac{\partial^2 \ell_{CE}(V)}{\partial v_i \partial v_i^T}||_F$ here, off-diag blocks are proved in the same way

$$\frac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_i^\top} \overset{(5)}{=} \frac{1}{N} \sum_{n=1}^{N} p_{n,i}(1-p_{n,i}) x_n x_n^\top := \frac{1}{N} X_N \Lambda_N X_N^\top \in \mathbb{R}^{d \times d},$$

where $X_N = (x_1, \cdots, x_N) \in R^{d \times N}$,

and $\Lambda_N = diag\big(p_{1i}(1-p_{1i}), \cdots, p_{Ni}(1-p_{Ni})\big) \in R^{N \times N}$, and $p_{n,i} = \frac{\exp(v_i^T x_n)}{\sum_{c=1} \exp(v_c^T x_n)}$

How to characterize $||\frac{1}{N} X_N \Lambda_N X_N^T||_F$?

Key Challenges in the Proof

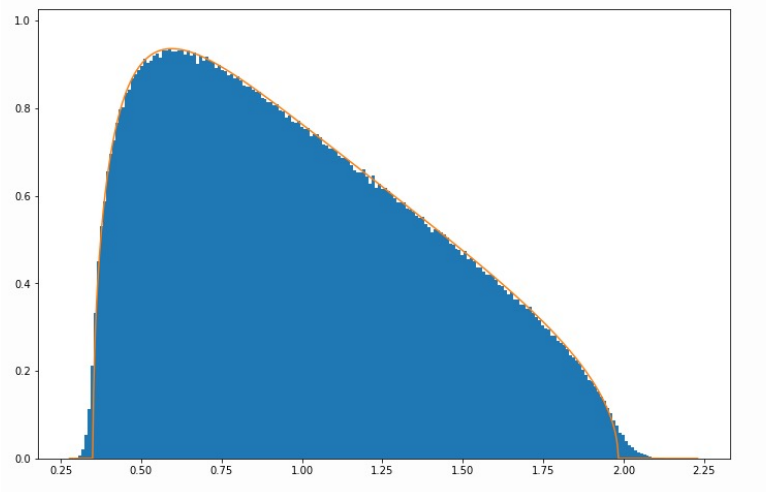
Diagonal Hessian block: $\frac{\partial^2 \ell_{\text{CE}}(V)}{\partial v_i \partial v_i^\top} \overset{(5)}{=} \frac{1}{N} \sum_{n=1}^{N} p_{n,i}(1 - p_{n,i}) x_n x_n^\top := \frac{1}{N} X_N \Lambda_N X_N^\top \in \mathbb{R}^{d \times d},$

Q: How to characterize the Hessian block $||\frac{1}{N} X_N \Lambda_N X_N^T||_F$?

We will use Random Matrix Theory (RMT), but classical methods cannot be directly applied:

- If X_N , Λ_N are **independent**, $||\frac{1}{N} X_N \Lambda_N X_N^T||_F$ can be found by **GMP Theorem (1967)**
- In our $\frac{1}{N} X_N \Lambda_N X_N^T$, X_N, Λ_N are clearly NOT independent, so MP theorem cannot be applied
- Dependent matrix product is a difficult topic in RMT

Exmple of GMP law: (Assume X and Λ are independent)



Eigenvalue histogram of $A_n = \frac{1}{n} X\Lambda X^T \in R^{d\times d}$, $\Lambda = I, n = 50, d = 300$, 1000 samples of A_n

Yellow curve: density of MP distribution with d/ n = 50 / 300

But wait… In our $\frac{1}{N} X_N \Lambda_N X_N^T$, X_N, Λ_N are **clearly NOT independent**, so MP theorem cannot be applied

- Dependent matrix product is a difficult topic in RMT
- Fortunately, we observe additional good properties in our $\frac{1}{N} X_N \Lambda_N X_N^T$

Key properties in our matrix

$$\frac{\partial^2 \ell_{\mathrm{CE}}(V)}{\partial v_i \partial v_i^\top} \overset{(5)}{=} \frac{1}{N}\sum_{n=1}^{N} p_{n,i}(1-p_{n,i}) x_n x_n^\top := \frac{1}{N} X_N \Lambda_N X_N^\top \in \mathbb{R}^{d\times d}, \qquad \text{and } p_{n,i} = \frac{\exp(v_i^T x_n)}{\sum_{c=1} \exp(v_c^T x_n)}$$

Key observation: $\boldsymbol{\Lambda_N}$ and $\boldsymbol{X_N}$ are asymptotic independence

- Recall $v_i \sim \mathcal{N}(0, \frac{1}{d})$ and denote $z = v_i^\top x_n$, then $z|x \sim \mathcal{N}(0, \frac{\|x\|_2^2}{d})$. Further, since $x \sim \mathcal{N}(0,1)$, we have $\frac{\|x\|_2^2}{d} \sim \mathcal{X}^2(1, \frac{2}{d})$, which concentrates to 1 as $d \to \infty$.
- As such, z asymptotically follows $\mathcal{N}(0,1)$ and thus is independent of x. Therefore, Λ_N and X_N are asymptotically independent.

- **Guess:** limiting eigenvalue $X_N \Lambda_N X_N^T \approx$ those as if X_N, Λ_N are independent
- The remaining question is how to prove it rigorously.

Our Proof Strategies

- We propose a systematic proof procedure to address the **"diminishing dependencies as $\boldsymbol{d} \to \infty$"**
- Our approach implements **the Lindeberg interpolation principle** which is originally proposed to prove CLT

Preparation: "decoupling": we introduce the following decoupling matrix

$$\widetilde{H}_{ii}^{\mathrm{CE}} = \frac{1}{N} \sum_{n=1}^{N} \widetilde{p}_{n,i}(1 - \widetilde{p}_{n,i}) x_n x_n^\top, \quad \widetilde{p}_{n,i} := \frac{\exp(v_i^\top \widetilde{x}_n)}{\sum_{c=1}^{C} \exp(v_c^\top \widetilde{x}_n)}, \tag{32}$$

where $\widetilde{X}_N = (\widetilde{x}_1, \cdots, \widetilde{x}_N) \in \mathbb{R}^{d \times n}$ is an independent copy of X_N.

Goal of decouple:

Now we want to prove:

- **Claim 1:** $\widetilde{H_{ii}^{CE}} = \frac{1}{N} X_N \widetilde{\Lambda_N} X_N^T$ and $H_{ii}^{CE} = \frac{1}{N} X_N \Lambda_N X_N^T$ share the same limit eigenvalue distribution

- If so, then we can apply GMP to $\widetilde{H_{ii}^{CE}}$
- Now we prove **Claim 1**

Our Proof Strategies (Overview)

Key challenge: Need $||\frac{1}{N}X_N\Lambda_N X_N^T||_F$, But X_N and Λ_N **are dependent**

Our solution: a new method built upon **the Lindeberg principle** (originally proposed to prove CLT)

Our "decouple" Strategy:

Step 1 (Important): "indept. copy $\tilde{X}_N$ + interpolation": we introduce the following $X_N(t)$

$$X_N(t) = \sqrt{t}\, X_N + \sqrt{1-t}\ \widetilde{X_N},\ \ t \in [0,1].$$ Note that $X_N(0) = X_N, X_N(1) = \tilde{X}_N$

Goal: Wish to show that: for any $z \in C^+$, $\delta_N(z) = \mathrm{E}s_{\widetilde{H_{ii}}}(z) - \mathrm{E}\, s_{H_{ii}}(z)$ vanishes as N increases

Step 2 (Important): Fundamental theorem of calculus

$$\delta_N(z) = \int_0^1 E\left[\frac{d}{dt}\ s_{H_{ii}(t)}\right] dt$$

Step 3 (Important): Using Cauchy Integral Formular, we prove that
$\delta_N(z) \leq Const.\mathrm{E}[Z_1 f(Z_1) - Z_2 f(Z_2)]$, where $\mathrm{Z}_i \sim N(0,1)$

RK: This step will fail if no asymptotic independence

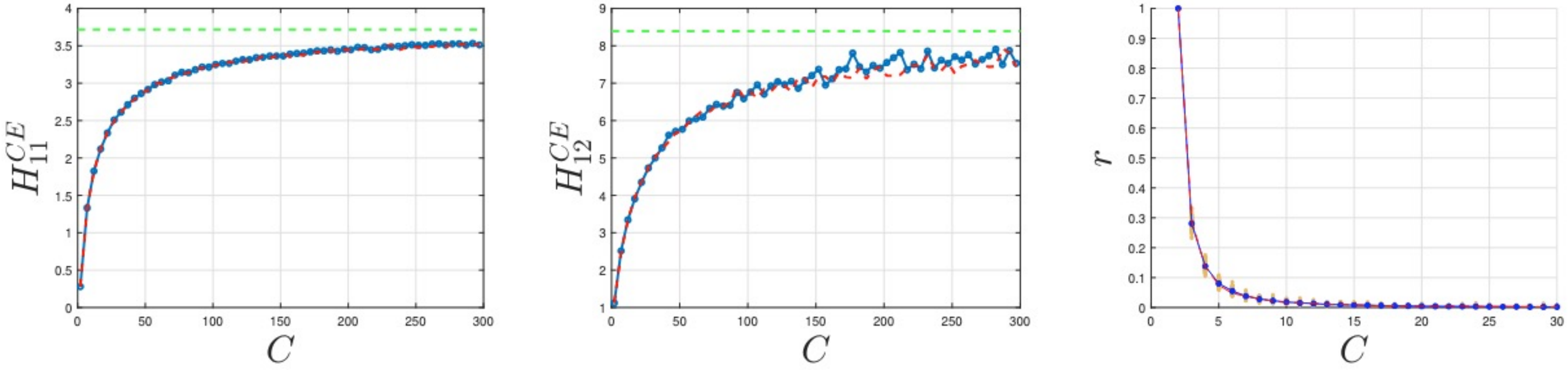
Step 4 (Important): Using Stein's Lemma, we prove that:

$$\mathrm{E}[Z_1 f(Z_1) - Z_2 f(Z_2)] = \mathrm{E}[\, f'(Z_1) - f'(Z_2)] = O\left(\frac{1}{\sqrt{N}}\right)$$

Step 5 (Standard): Apply GMP to recover $\mathbf{S}_{\widetilde{\mathbf{H}_{ii}}}(\mathbf{z})$, $\boldsymbol{\mu}_{\widetilde{H_{ii}}}$, and $\boldsymbol{\mu}_{H_{ii}}$

The Effect of Increasing C

$$\tilde{H}_{11}^{\mathrm{CE}} := \frac{1}{d}\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_1 \partial w_1^\top}\right\|_{\mathrm{F}}^2, \quad \tilde{H}_{12}^{\mathrm{CE}} = \frac{C}{d}\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_1 \partial w_2^\top}\right\|_{\mathrm{F}}^2, \quad \tilde{r} = \frac{\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_1 \partial w_2^\top}\right\|_{\mathrm{F}}^2}{\left\|\frac{\partial^2 \ell_{\mathrm{CE}}(W,V)}{\partial w_2 \partial w_2^\top}\right\|_{\mathrm{F}}^2}$$

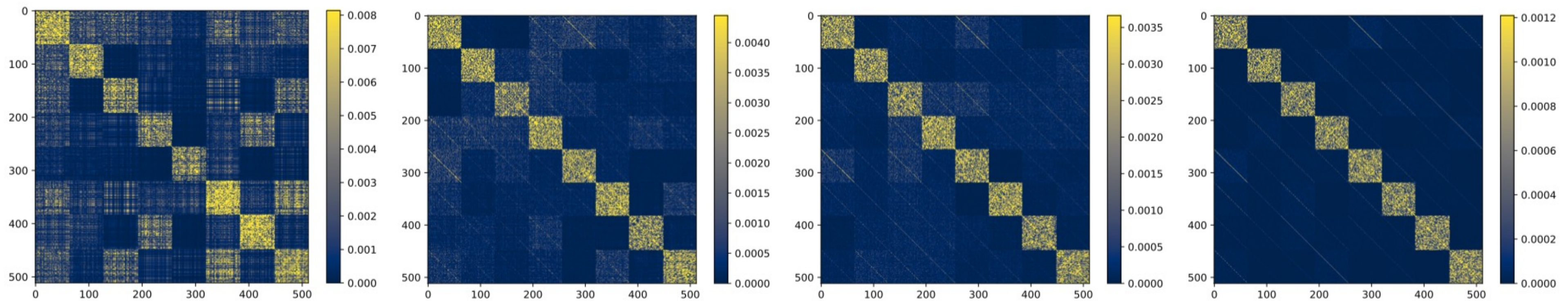


(a) C v.s. H_{11}^{CE} (b) C v.s. H_{12}^{CE} (c) C v.s. r

These quantities match our **theoretical prediction**

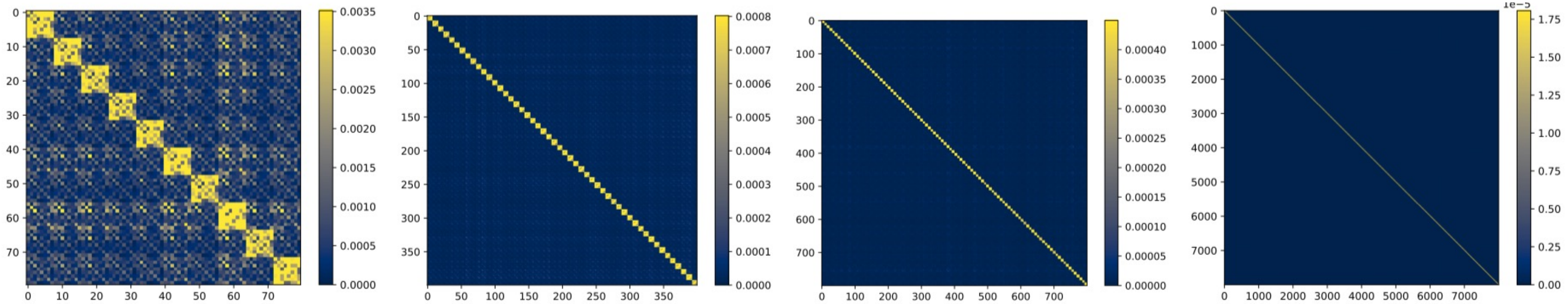
The Effect of Increasing # classes C

Hessian of hidden weights



$(a)\ C = 10$ $(b)\ C = 50$ $(c)\ C = 100$ $(d)\ C = 1000$

Hessian of output weights



$(a)\ C = 10$ $(b)\ C = 50$ $(c)\ C = 100$ $(d)\ C = 1000$

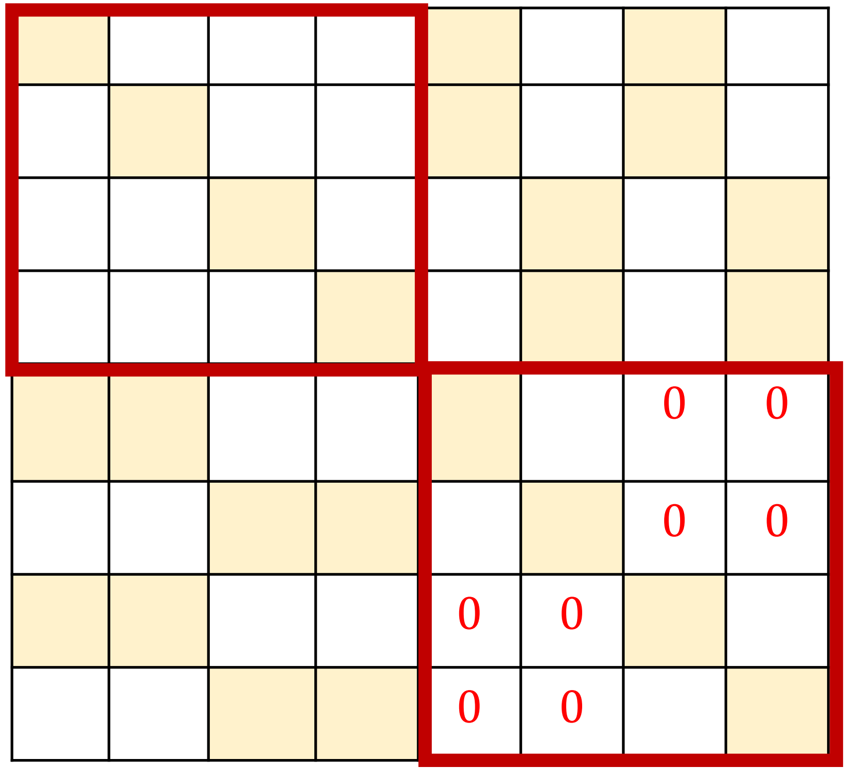
- The Hessian blocks of **1-hidden-laye NN** with 8 hidden neurons + #class C at random init.
- The block-diag structure **becomes clearer as C increases**

Summary

- **We discussed the intuition behind the special structure of Hessian**
  - linear algebra and & probability perspective
- **We rigorously prove using random matrix theory**
  - Key factor: # classes $c \to \infty$
- **Technical challenges: non-independent random matrix products $X\Lambda X^{\mathrm{T}}$**
  - Our solution: a new method based on the Linderberg Principle

Summary: 3-level sources of block-diag structure

- **Level 1: definition of matrix product: many zeros, no links**



Summary: 3-level sources of block-diag structure

- **Level 1: definition of matrix product: many zeros, no links**
- **Level 2: #Class C goes to infinity: weaken many links in** H_{ww}, H_{vv}

Static force

	≈ 0	≈ 0	≈ 0		≈ 0		≈ 0
≈ 0		≈ 0	≈ 0		≈ 0		≈ 0
≈ 0	≈ 0		≈ 0	≈ 0		≈ 0	
≈ 0	≈ 0	≈ 0		≈ 0		≈ 0	
		≈ 0	≈ 0		≈ 0	0	0
≈ 0	≈ 0			≈ 0		0	0
		≈ 0	≈ 0	0	0		≈ 0
≈ 0	≈ 0			0	0	≈ 0	

Summary: 3-level sources of block-diag structure

- **Level 1: definition of matrix product: many zeros, no links**
- **Level 2: #Class C goes to infinity: weaken many links in** H_{ww}, H_{vv}

Static force

- **Level 3: Training: eliminates strong links in** H_{wv}

Dynamic force

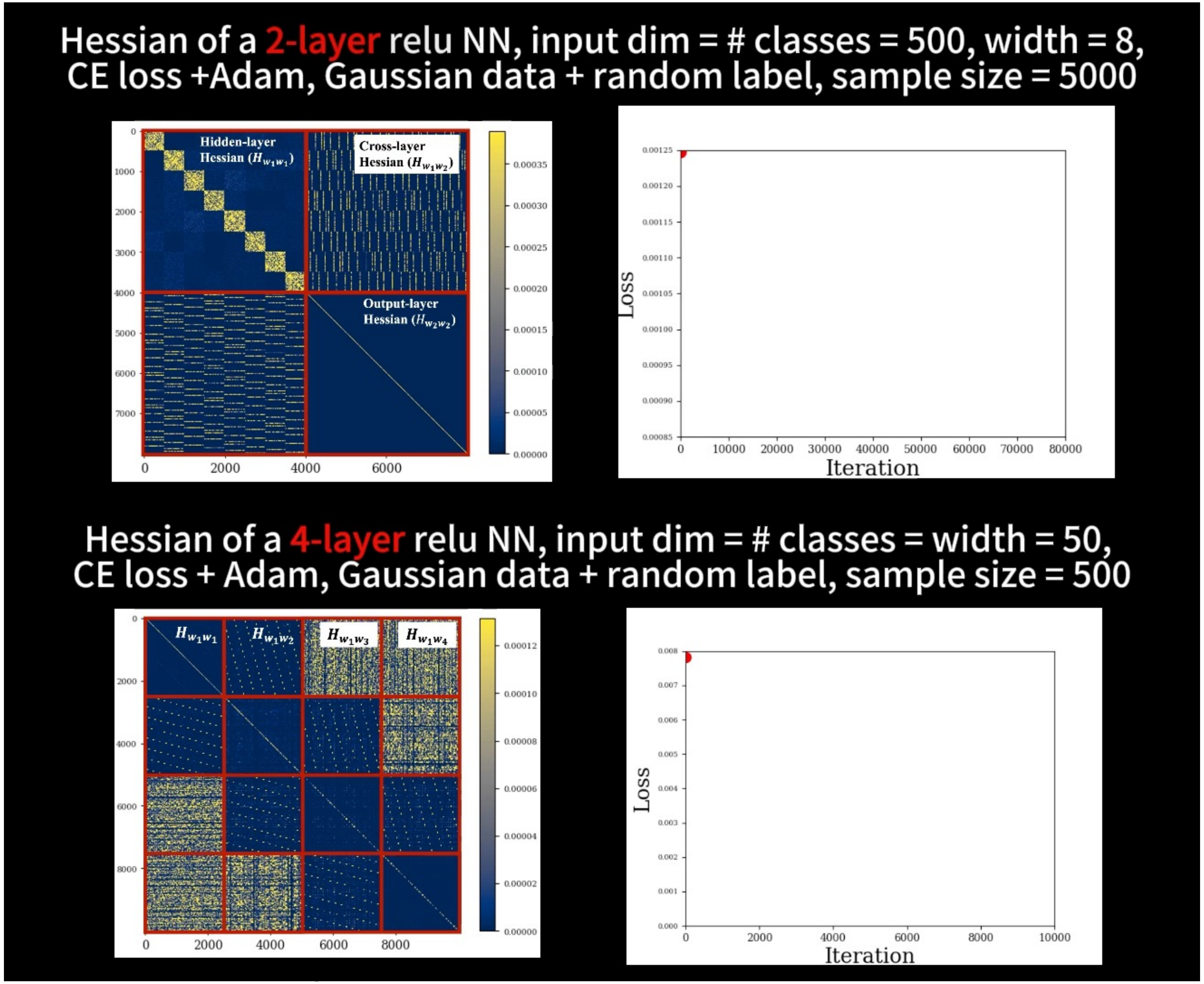
	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
≈ 0		≈ 0	≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
≈ 0	≈ 0		≈ 0	≈ 0	≈ 0	≈ 0	≈ 0
≈ 0	≈ 0	≈ 0		≈ 0	≈ 0	≈ 0	≈ 0
≈ 0	≈ 0	≈ 0	≈ 0		≈ 0	0	0
≈ 0	≈ 0	≈ 0	≈ 0	≈ 0		0	0
≈ 0	≈ 0	≈ 0	≈ 0	0	0		≈ 0
≈ 0	≈ 0	≈ 0	≈ 0	0	0	≈ 0	

Guess: Hessian for Deep NNs?

Guess: Hessian for Deep NNs?

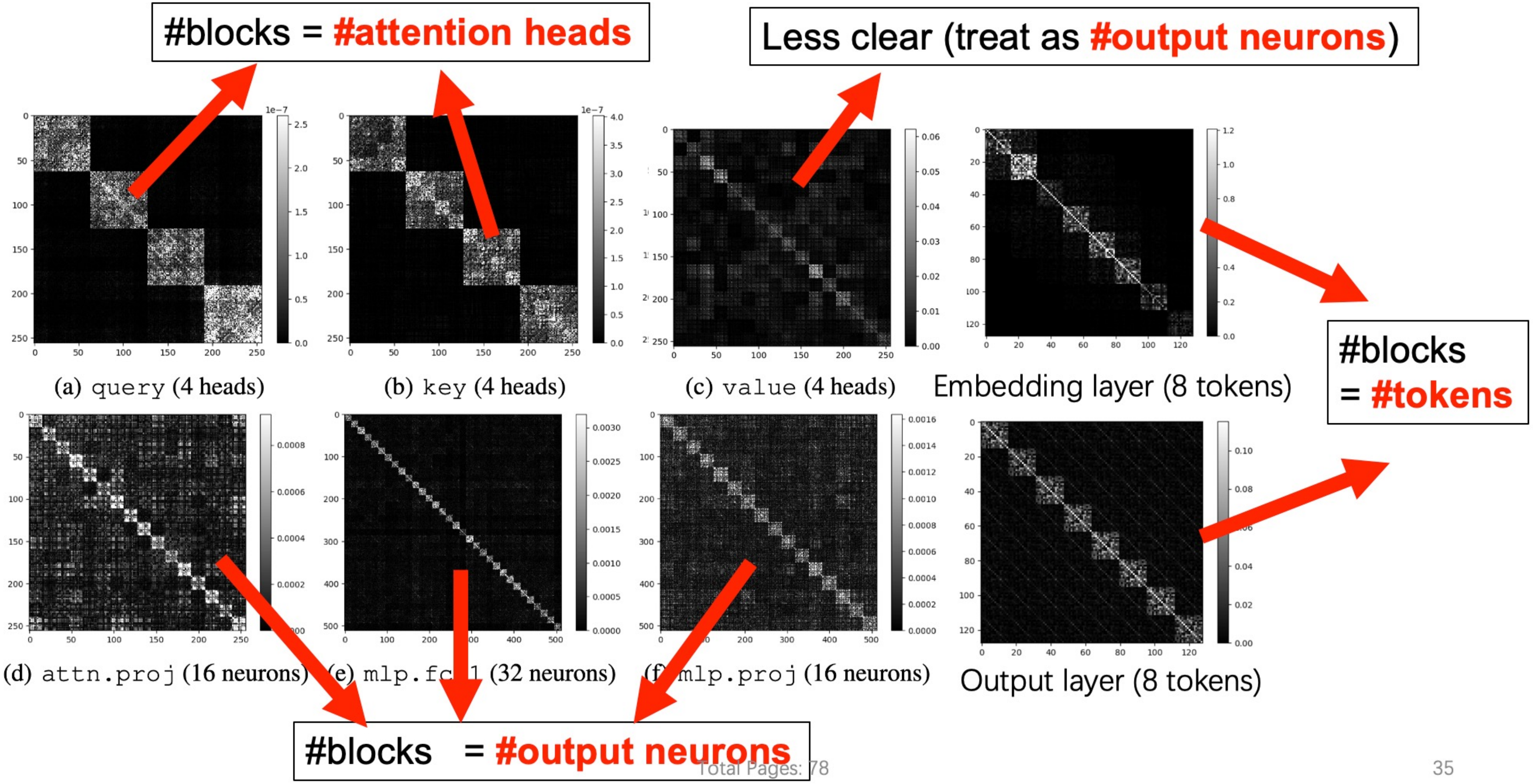
For a rough estimate:
just check the links in the computational graph

Numerical result:
Does it match your estimation?



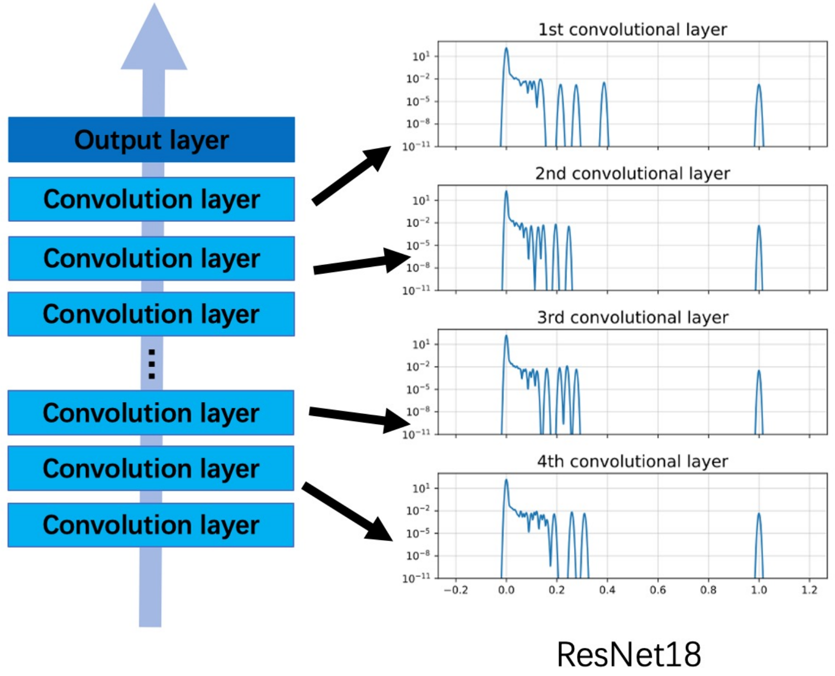
Contents

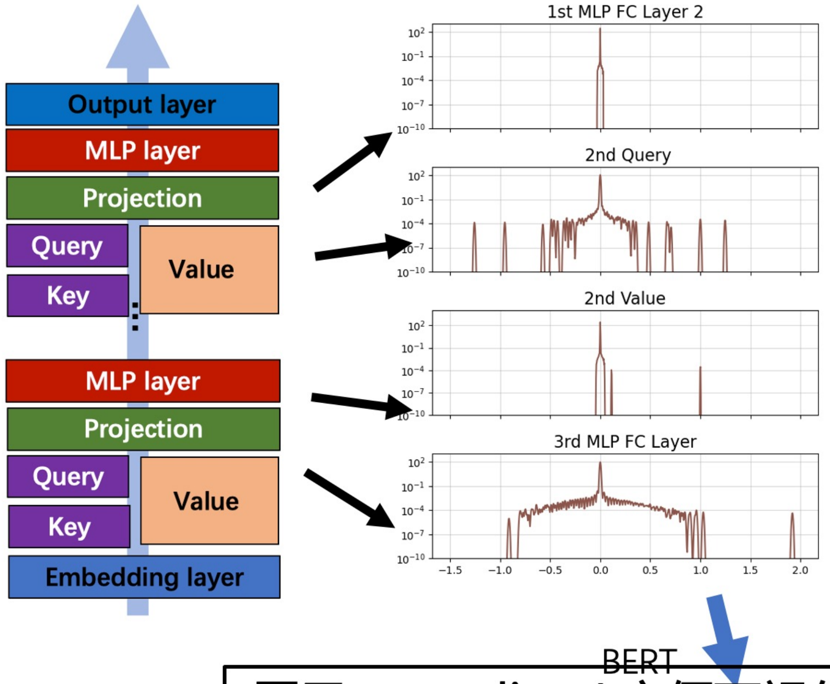
What about the Hessian of Transformers?



Implication I: Why Transformers Need Adam

Blockwise Hessian spectrum





图已normalize（方便可视化）
实际Eigen-range**相差>200倍**

CNNs: blockwise spectrum are quite **similar**
We call it ``**homogeneity**"

Transformers: blockwise spectrum are largely **different**
We call it ``**heterogeneity**"

Total page: 58

36

[1] Why Transformers Need Adam: A Hessian Perspective. **Zhang**, Chen, Ding, Li, Sun, Luo, NeurIPS 2024,

Implication I: Why Transformers Need Adam

JS-distance among blocks

CNNs

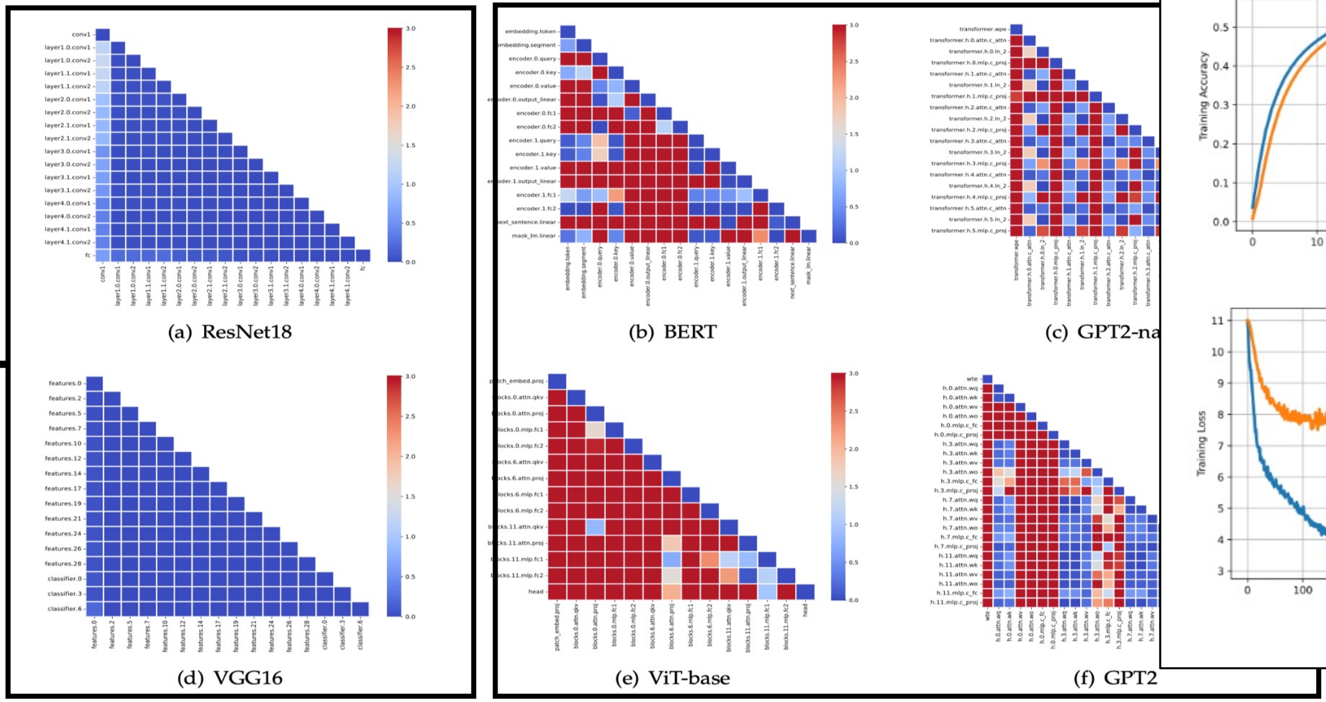
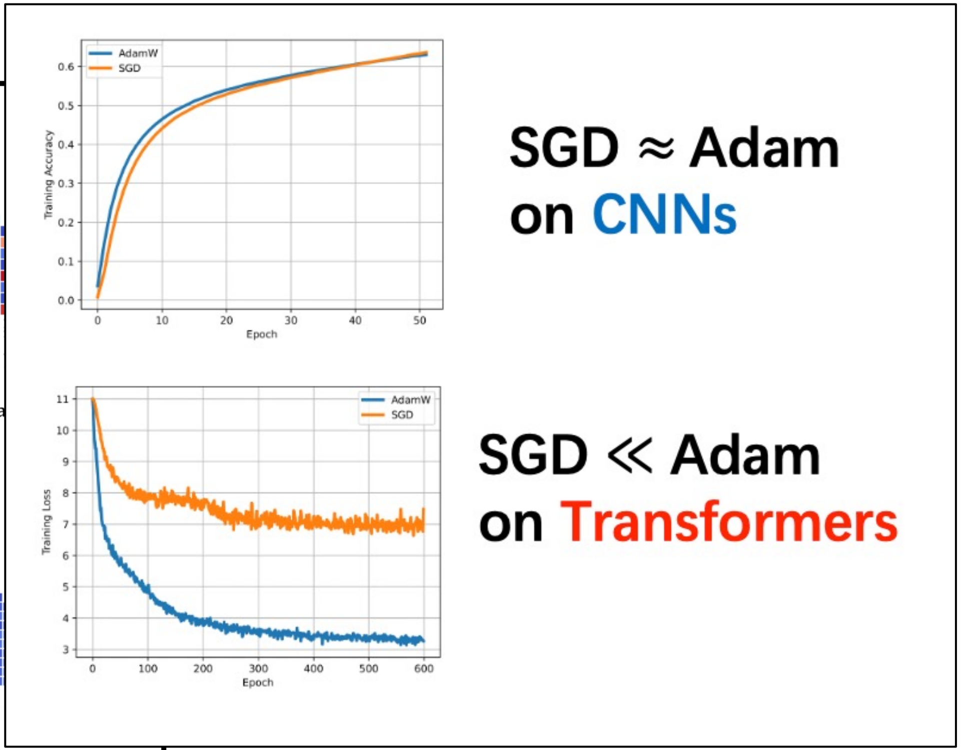


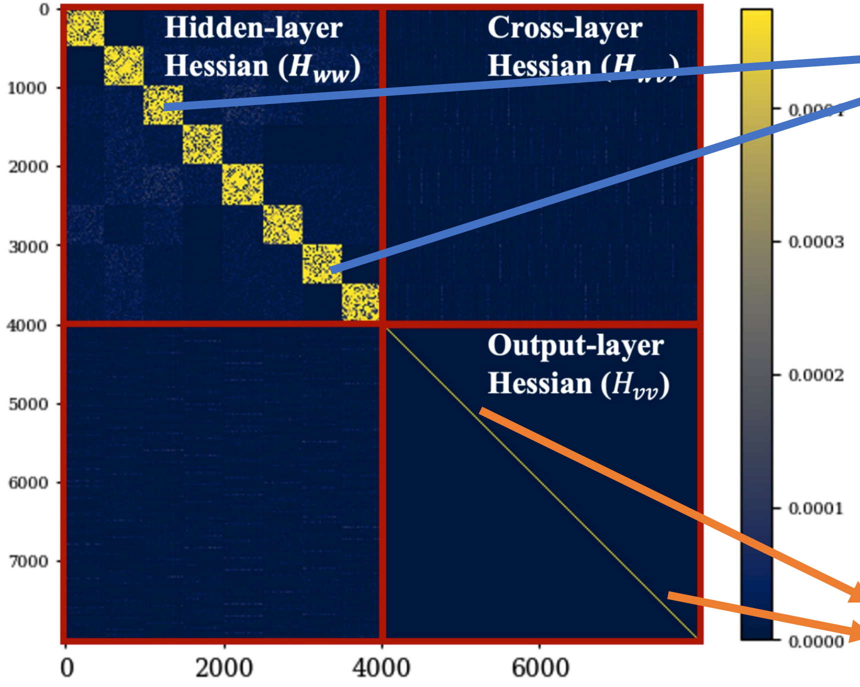
Figure 4: The JS distance among blockwise Hessian spectra for different models at initialization.



Observation 1: Heterogeneity is widely observed in Transformers, but not on CNNs!

[1] Why Transformers Need Adam: A Hessian Perspective. **Zhang**, Chen, Ding, Li, Sun, Luo, NeurIPS 2024,

When and Why Adam ≫ SGD? Hessian Structure Might Help



Hessian of NN has very special Structure

- Proved in [1]
- Why? large # output dim + training

Architecture

CNN: blockwise spectrum is observed to be **similar** [2]

- No proof now
- **SGD ≈ Adam**

Transformer: blockwise spectrum is observed to be **heterogeneous** [2]

- Later proved in [3]. Why? Softmax is the one to blame
- **SGD ≪ Adam**

Data

Balanced label: blockwise spectrum of lm_head is observed to be **similar** [4]

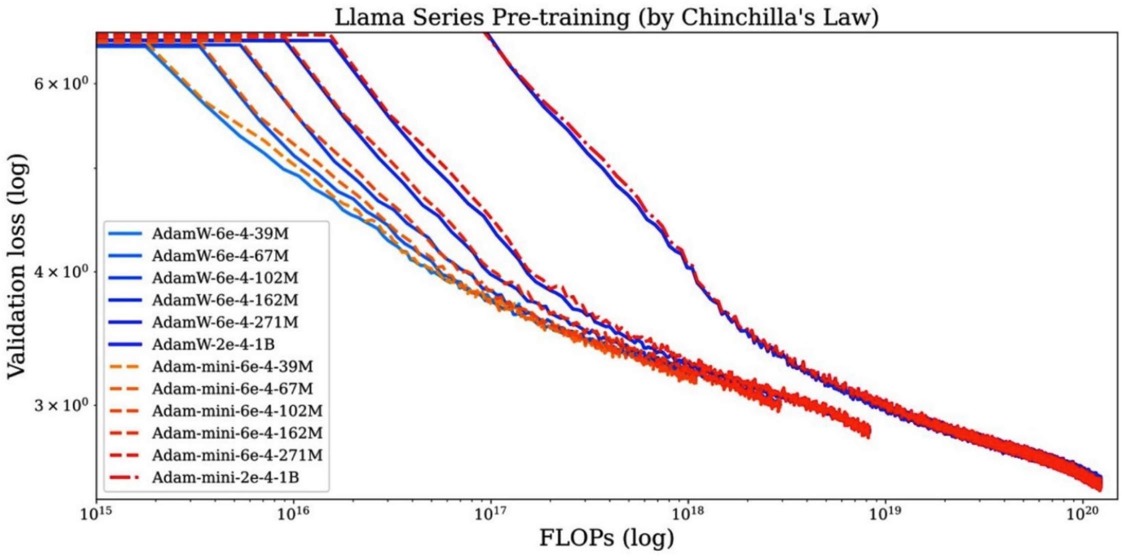
- Preliminary explanation in [4]
- **SGD ≈ Adam**

Imbalanced label: blockwise spectrum of lm_head is observed to be **heterogeneous** [4]

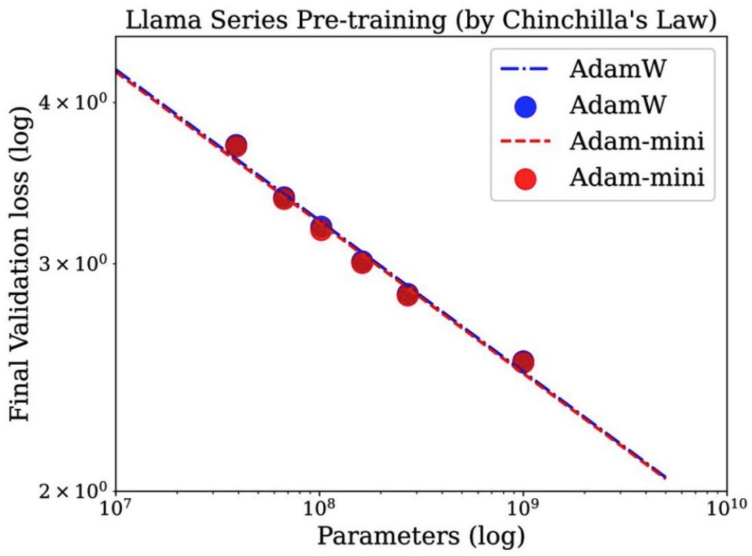
- Preliminary explanation in [4]
- **SGD ≪ Adam**

[1] Towards Quantifying the Hessian Structure of Neural Networks.
[2] Why Transformers Need Adam: A Hessian Perspective
[3] What Does It Mean to Be a Transformer? Insights from a Theoretical Hessian Analysis
[4] Heavy-Tailed Class Imbalance and Why Adam Outperforms Gradient Descent on LLMs

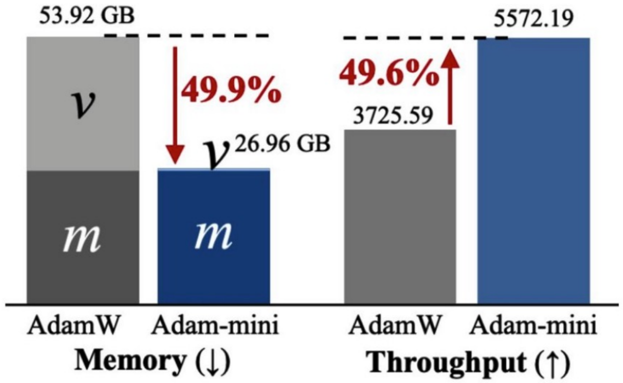
Implication II: New algorithm Adam-mini



(a) Scaling laws in terms of compute



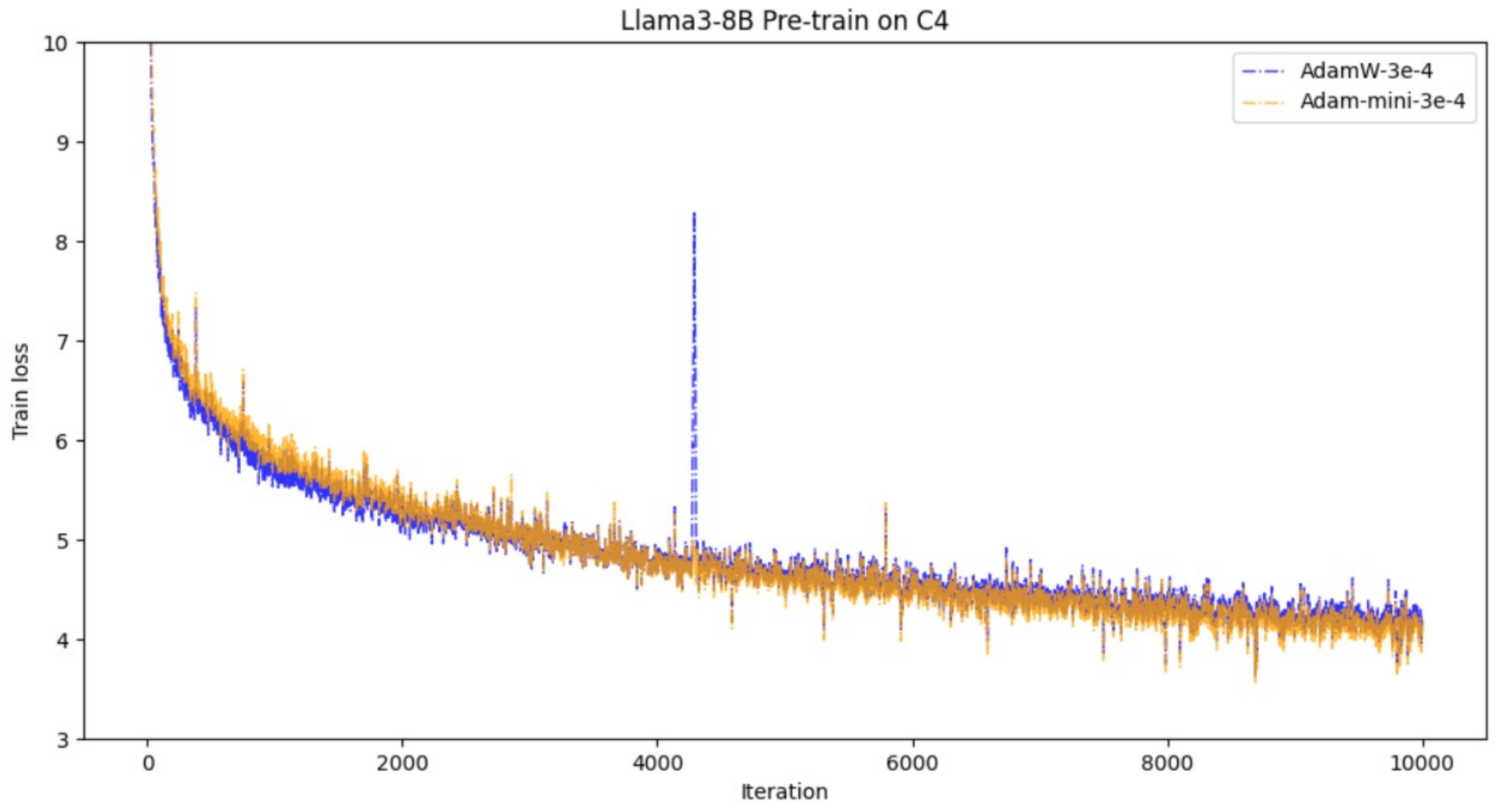
(b) Scaling laws in terms of parameters

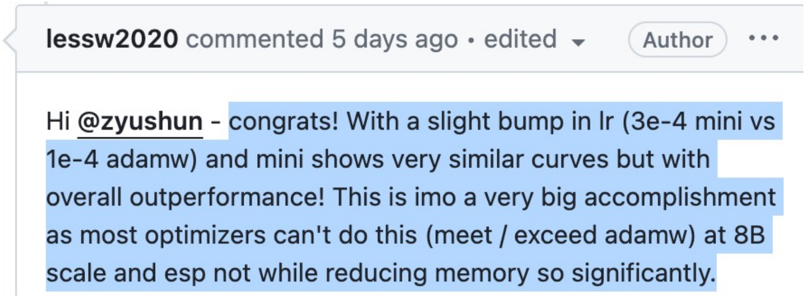


Chinchilla Scaling laws of Adam-mini: same performance as AdamW, but with **50%** less memory

[2] Adam-mini: Use Fewer Learning Rates To Gain More, **Zhang,** Chen, et al., ICLR 2025

LLama3-8B Pretrain: Independent verifier from PyTorch team





Highlight:

"This is imo a very big accomplishment as most optimizers can't do this (meet / exceed adamw) at 8B

… and especially not while reducing memory so significantly"

Acknowledgements from the Authors of Adam

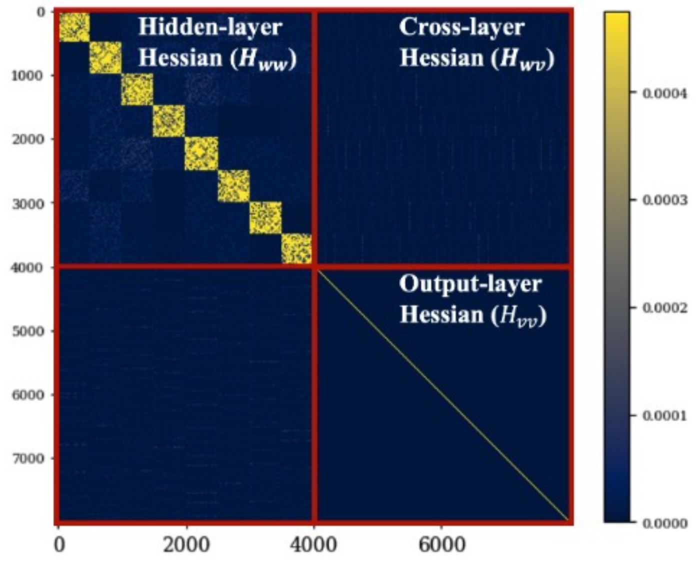
- **Photo shot at ICLR 2025 Test of Time Speech by Dr. Durk Kingma and Prof. Jimmy Ba**
- “This work allows you to reduce the memory of Adam by a large factor …

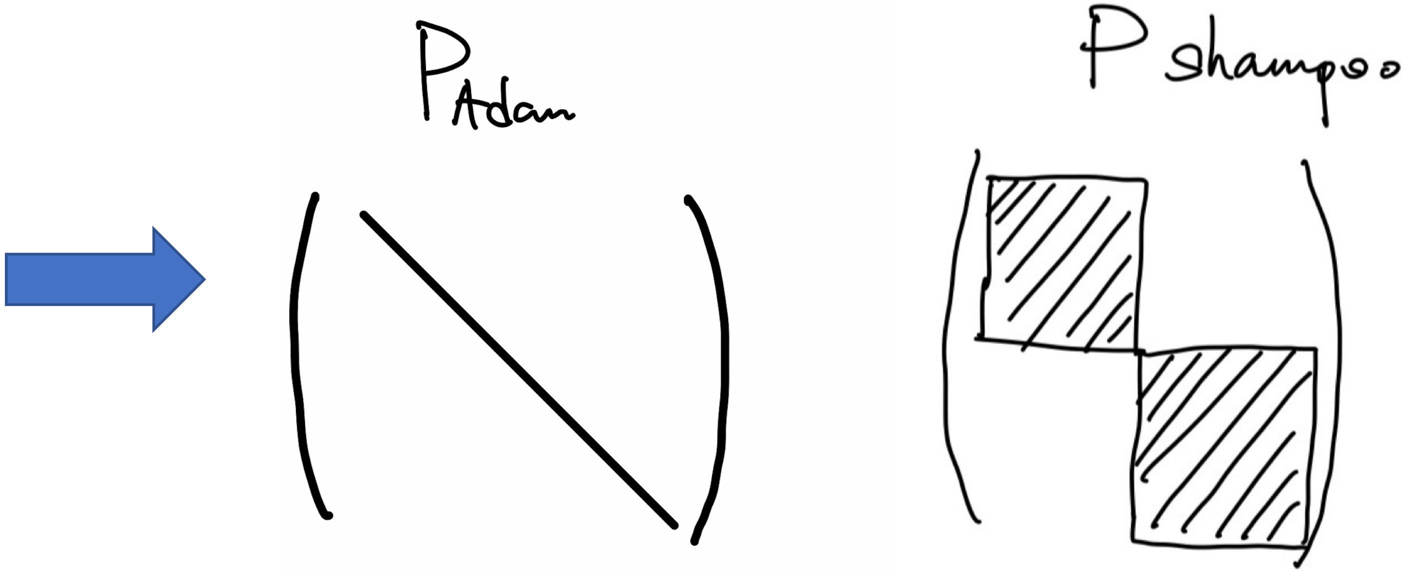
  This is, I think, a great result that argued from theory ”

Implication III: Shampoo & Muon

$$w = w - \eta P^{-\frac{1}{2}} m$$

True Hessian (Supported by our theory)





Our theory can support Shampoo (and Muon)

Implication IV: New algorithm ASGO

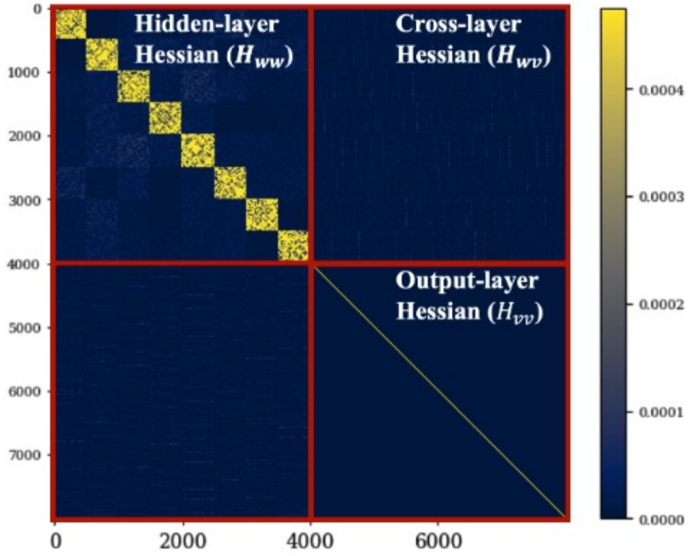
ASGO: Adaptive Structured Gradient Optimization

Kang An[1*], Yuxing Liu[2*], Rui Pan[2], Shiqian Ma[1], Donald Goldfarb[3], Tong Zhang[2]

[1]Rice University [2]University of Illinois Urbana-Champaign [3]Columbia University

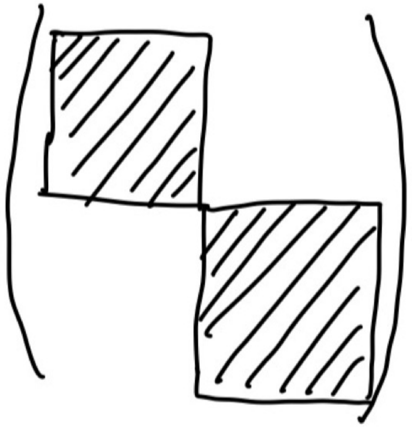
{kang.an,shiqian.ma}@rice.edu, {yuxing6,ruip4,tozhang}@illinois.edu, goldfarb@columbia.edu

True Hessian (Supported by our theory)

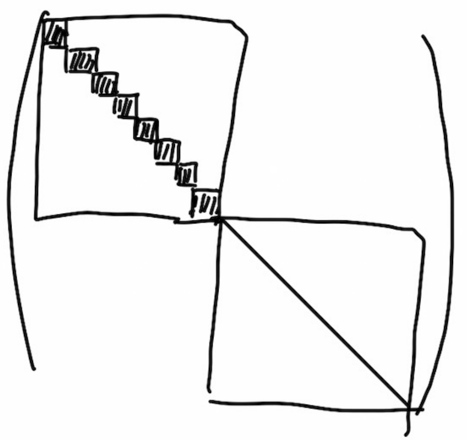


$$w = w - \eta P^{-\frac{1}{2}} m$$

$P_{shampoo}$

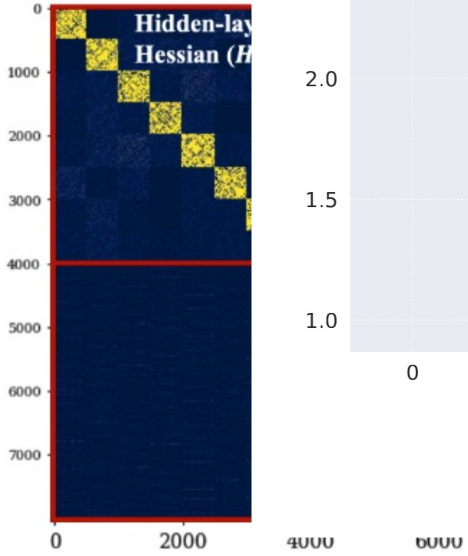


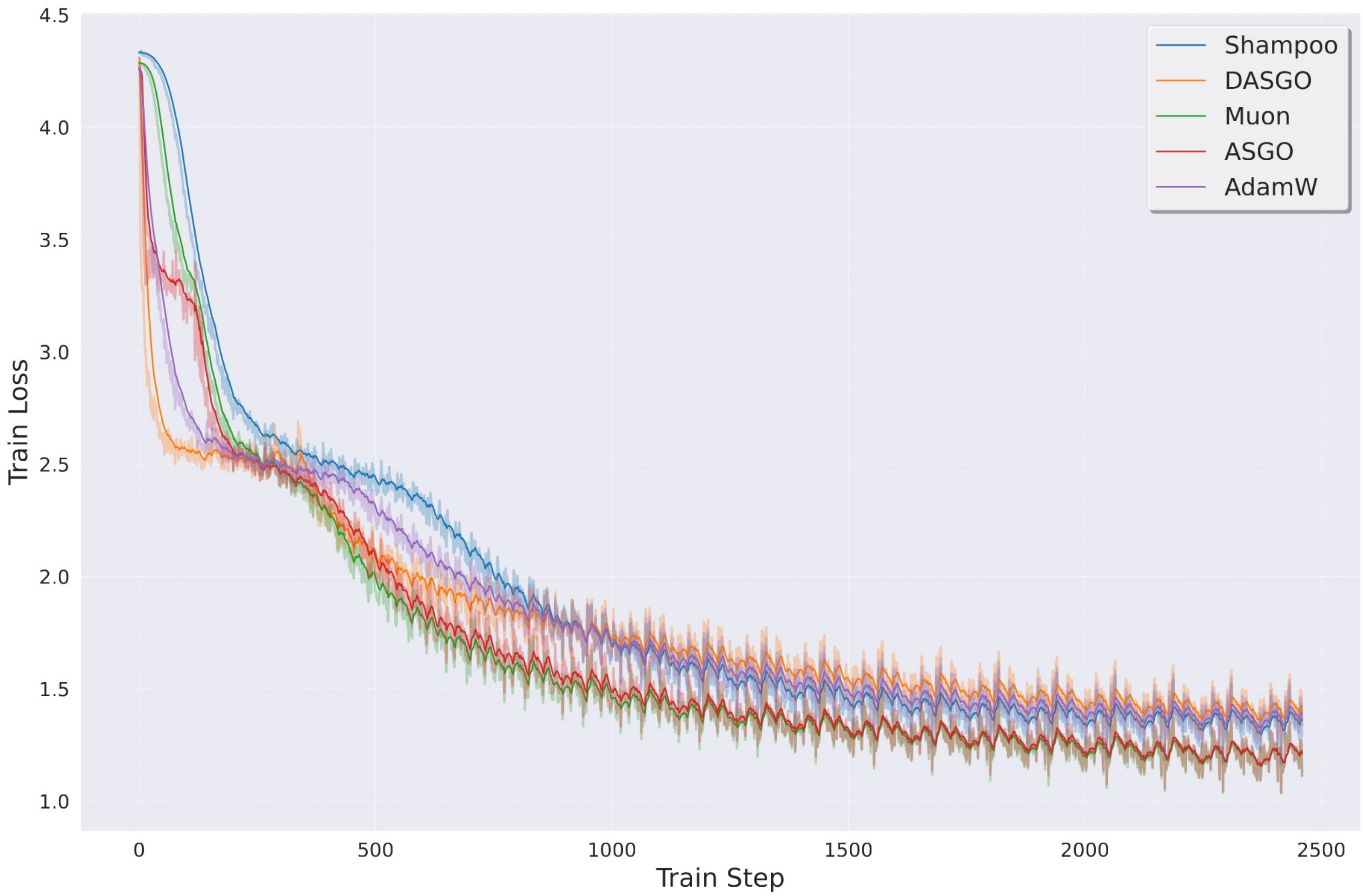
P_{ASGO}



Implication IV: New algorithm ASGO

True Hessian (Su

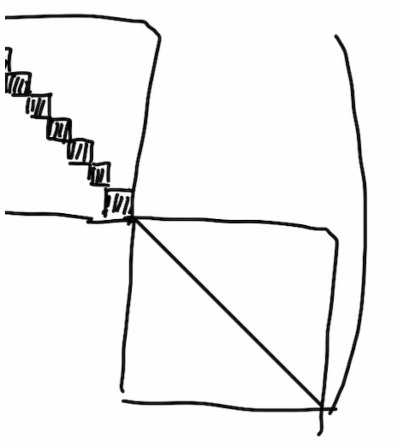




(a) Train Loss

$\frac{1}{2}\, m$

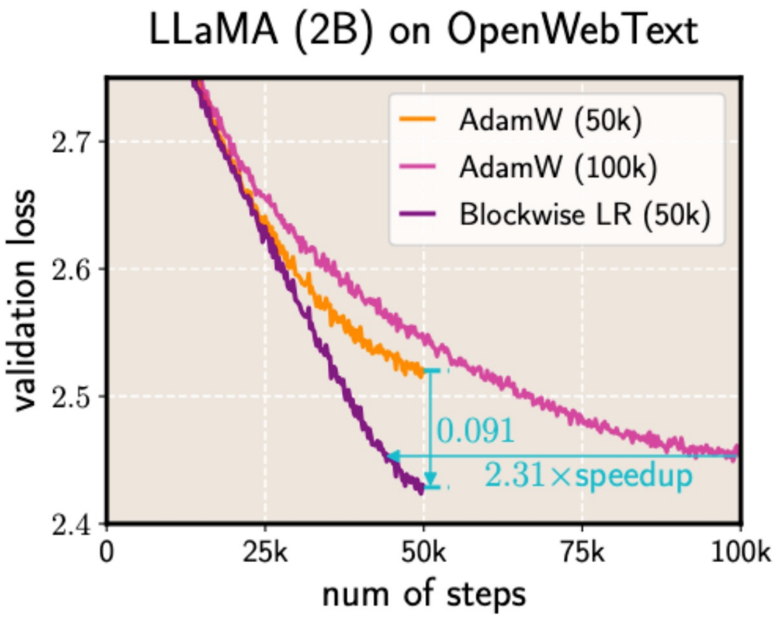
P_{ASGO}

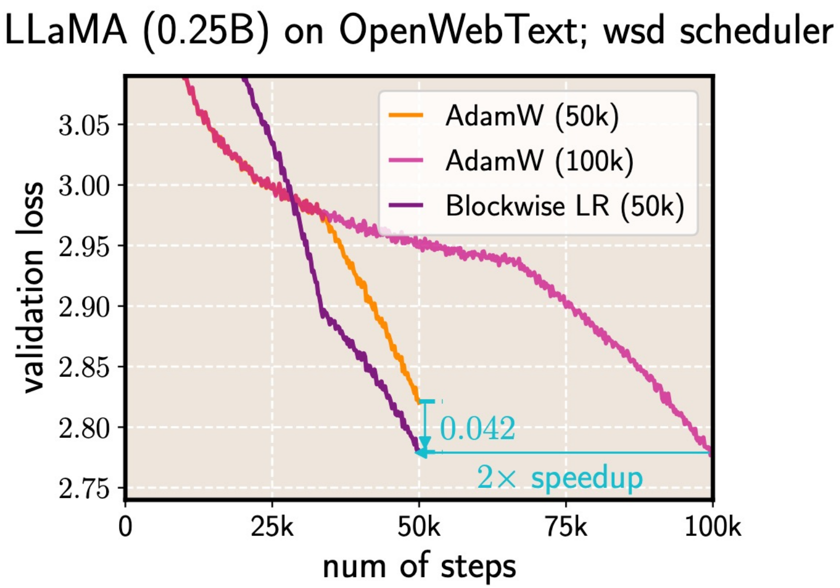


Implication V: block-wise learning rate

The Sharpness Disparity Principle in Transformers for Accelerating Language Model Pre-Training

Jinbo Wang [*1] **Mingze Wang** [*1] **Zhanpeng Zhou** [*2] **Junchi Yan** [2] **Weinan E** [1 3 4] **Lei Wu** [1 3 4]





Mainly based on:

- **Zhang**, Chen, Ding, Li, Sun, & Luo; Why Transformers Need Adam: A Hessian Perspective, **NeurIPS 2024**
- **Zhang**, Chen, Li, Ding, Wu, Kingma, Ye, Luo & Sun; Adam-mini: Use Fewer Learning Rate To Gain More, **ICLR 2025**
- Dong*, **Zhang*** (Alphabetically ordered), Luo, Yao, Sun; Towards Quantifying the Hessian Structure of Neural Networks, **Preprint**
- **Thanks to all the collaborators!**

Jeff J. Yao

D.P. Kingma

Yinyu Ye

Ruoyu Sun

Zhi-Quan Luo

How to Use Adam-mini? Just 1-line code change

```
pip install adam-mini
```

```
from adam_mini import Adam_mini
```

```
optimizer = Adam_mini(
            named_parameters = model.named_parameters(),
            lr = lr,
            betas = (beta1,beta2),
            eps = eps,
            weight_decay = weight_decay,
            model_sharding = True,
            dim = model_config.dim,
            n_heads = model_config.n_heads,
            n_kv_heads = model_config.n_kv_heads,
            )
```

Same values as AdamW!

Your model config

We support: DDP, FSDP, Deepspeed, Torchtitan, HF trainer 🤗

👆 Code for Adam-mini
Currently:
-- 400+ stars
-- 2000+ download via pip install
(in the last two weeks)

Code: https://github.com/zyushun/Adam-mini

Hessian and classical ideas are still powerful!

Thanks for listening!