基于信息瓶颈理论的神经元竞争初始化策略

主讲人：毛海涛

Microsoft

Neuron Campaign for Initialization Guided by Information Bottleneck Theory

Haitao Mao[1,2], Xu Chen[1,3], Qiang Fu[1], Lun Du[1], Shi Han[1], Dongmei Zhang[1]

1. Microsoft Research Asia
2. University of Electronic Science and Technology of China
3. Peking University

Contents

- Background
  - Information Bottleneck Theory
  - Initialization Strategy
- Our paper
  - Approach
  - Evaluation
  - Conclusion & Future work
- Further Exploration

Background

Microsoft

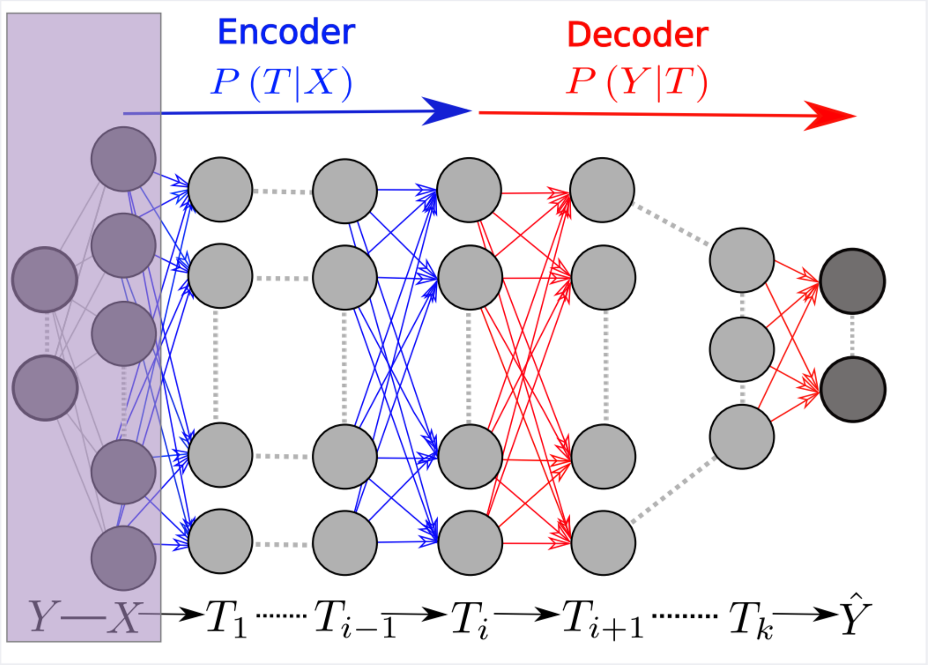
Information Bottleneck Theory

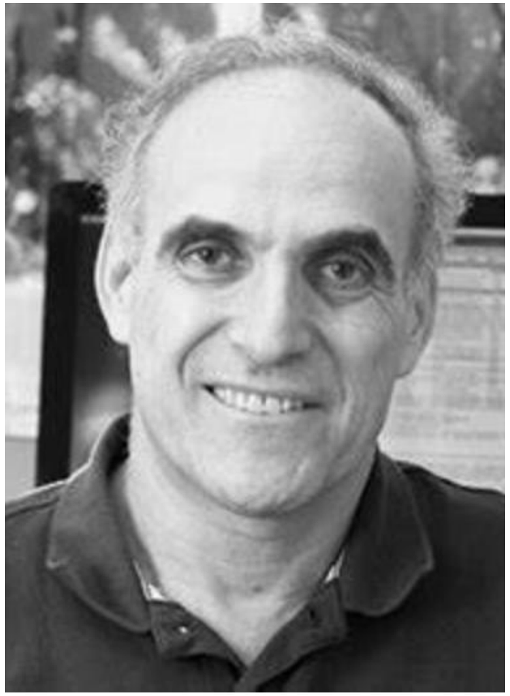
Information Bottleneck

- Tishby et al. believes that DNN training is actually optimizing the following objective:

$$\min_{\Theta} I(X;T) - \beta I(T;Y)$$

where T is the feature of each layer, X is the input and Y is the label

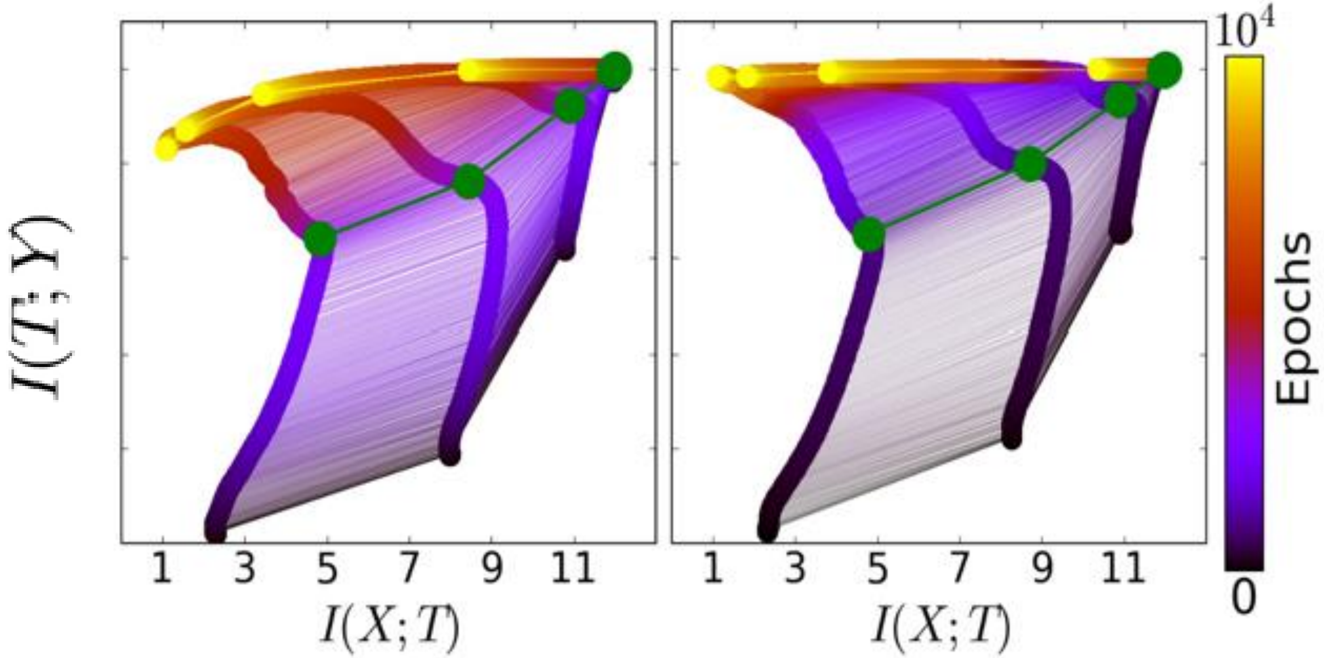




[1] Opening the Black Box of Deep Neural Networks via Information. Arxiv: 1703.00810

[2] Talk by Prof. Tishby: https://www.youtube.com/watch?v=bLqJHjXihK8&t=262s

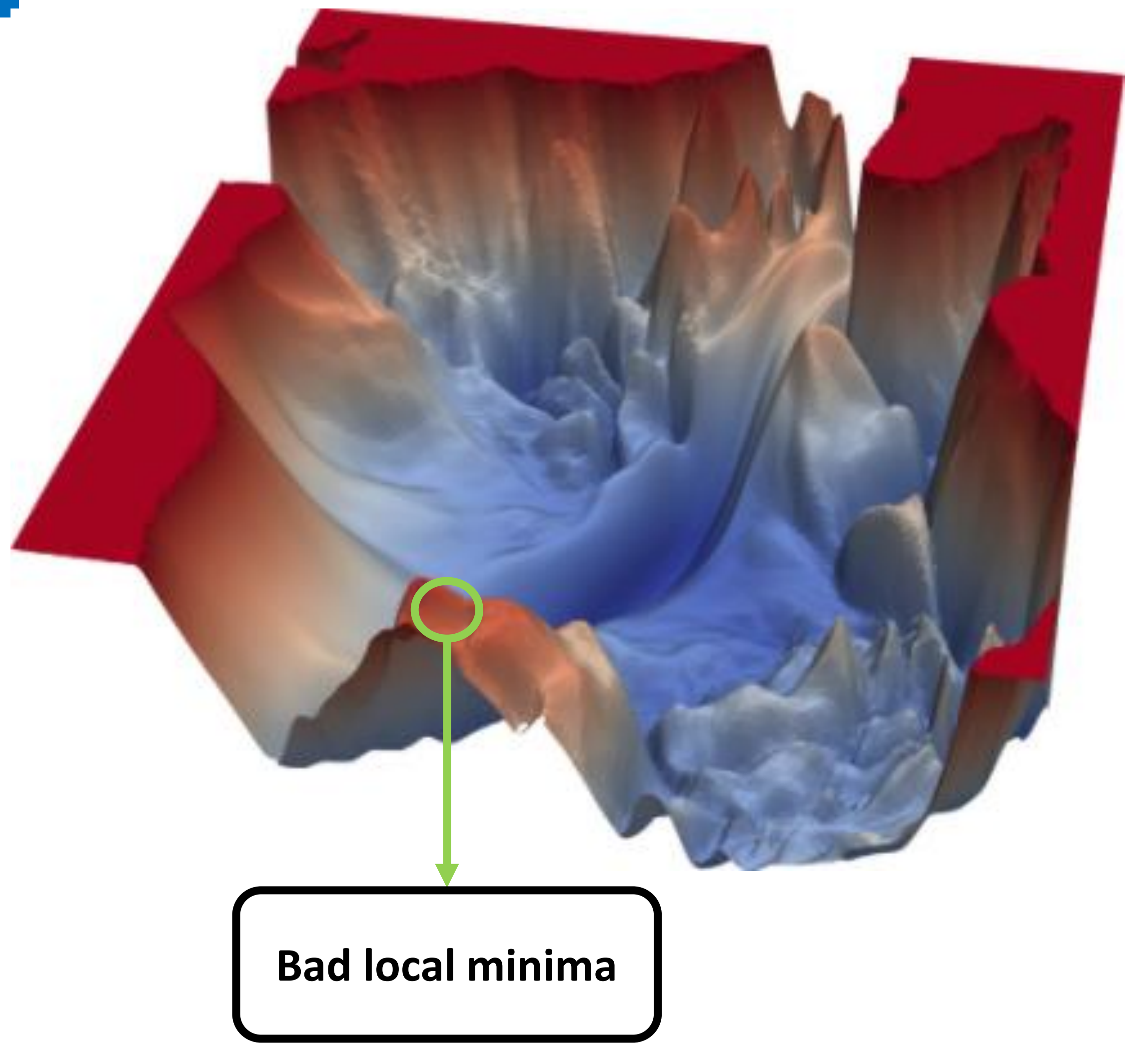
Information Bottleneck



- Phase 1: $I(X;\ T)$ and $I(T;\ Y)$ both increases, which indicates the network memorizes the information about the input
- Phase 2: $I(X;\ T)$ decrease while $I(T;\ Y)$ increases, which indicates that the network drops unimportant information to generalize.

Initialization Strategy

Background



- Training a DNN is to find a good local minima.
- A bad initialization may lead to stuck in a bad local minima.

Gradient vanish/explode

Random Initialization

$W \sim N(0, 0.01^2)$

```
x = torch.randn(512)

for i in range(100):
    a = torch.randn(512,512) * 0.01
    x = a @ x
x.mean(), x.std()
```

```
(tensor(0.), tensor(0.))
```

$W \sim N(0, 1)$

```
x = torch.randn(512)

for i in range(100):
    a = torch.randn(512,512)
    x = a @ x
    if torch.isnan(x.std()): break
i
```

```
28
```

Reason for it

- Gradient Exposure and vanish
  - Forward

$$y = W_3 * W_2 * W_1 * x$$

  - Backward

$$\nabla W_1 = \boxed{\frac{\partial Loss}{\partial f_3}} * \boxed{\frac{\partial f_3}{\partial z_3} * \frac{\partial f_2}{\partial z_2} * \frac{\partial f_1}{\partial z_1}} * \boxed{W_3 * W_2} * X$$

Glorot condition

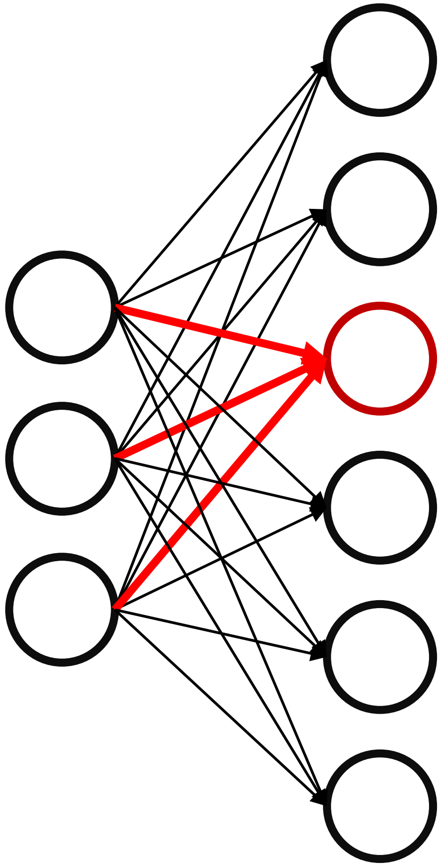
- The variance of the outputs of different hidden layer should be similar
- The variance of gradient from different layer should be similar

Xavier Initialization

$$y_l = W_l x_l + b_l$$

- *Assumptions*
  - *Linear activation function*
  - *The expectation of input and weight are all 0*
  - W_l *are mutually independent and share the same distribution*
  - x_l *are mutually independent and share the same distribution*
  - x_l *and* W_l *are independent of each other*
- Final form

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right]$$



2022/1/5

Variance scaling Initialization strategy

- He initialization (for Relu activation function)

$$W \sim N\left(0, \sqrt{\frac{2}{n_j}}\right)$$

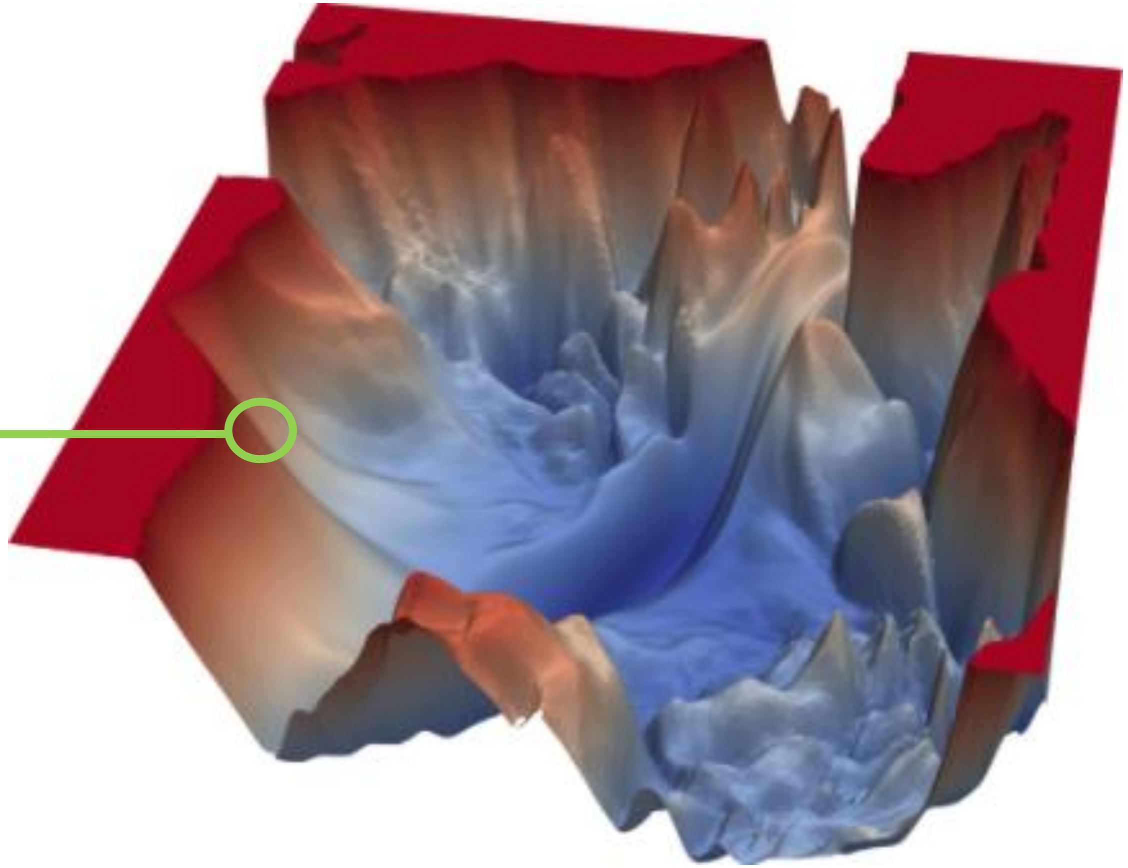
- LSUV (Layer Sequential Unit-Variance initialization)

$$\mathrm{W}_L = \mathrm{W}_L / \sqrt{var(Z_i)}$$

Introduction of Our paper

Motivation

- **What initialization leads to better generalization**
- **How to avoid fluctuation in the training**



Good point we want

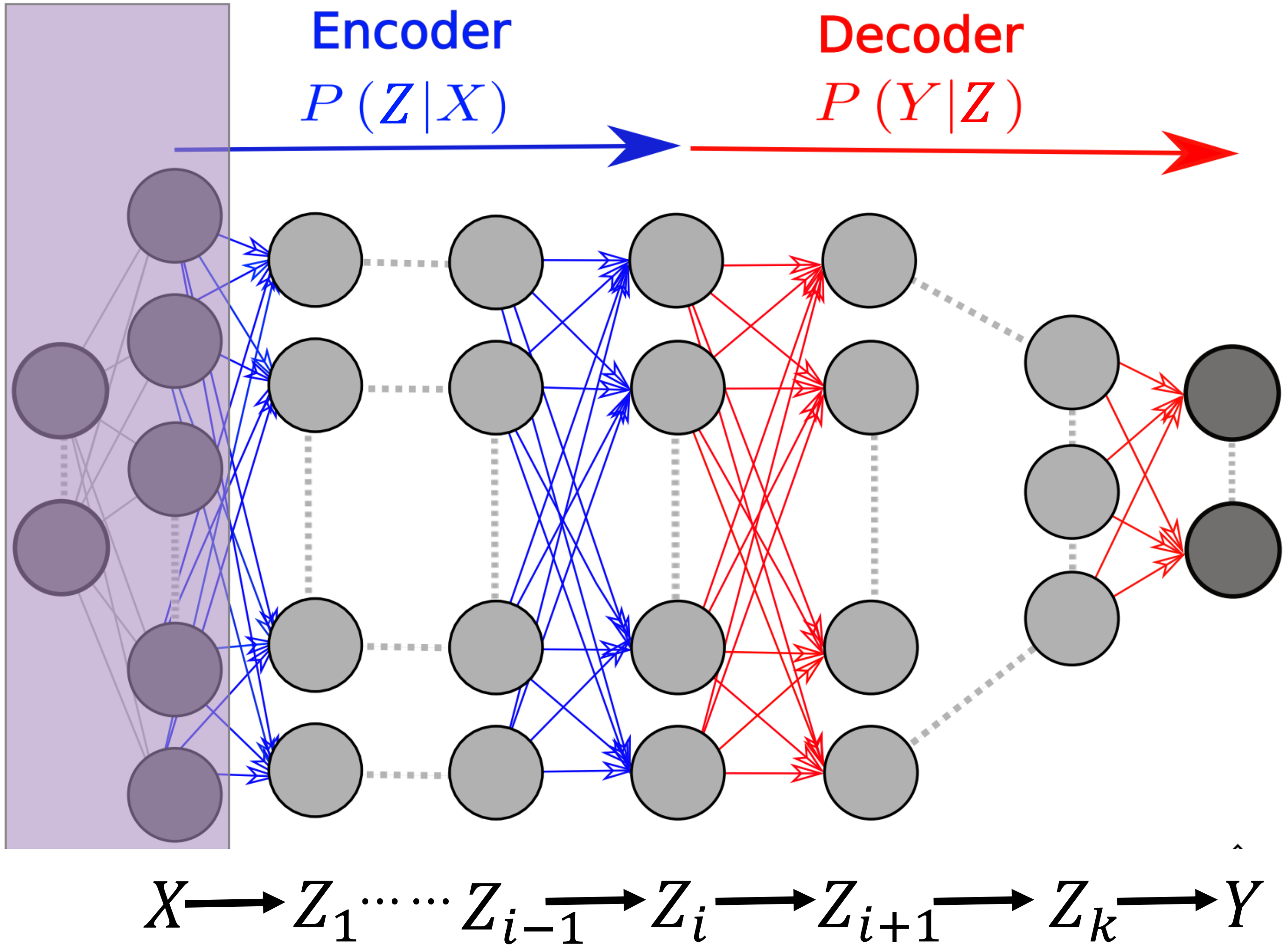
Microsoft

Approach

- **What is the good initialization point leads to better generalization**
  Two criteria guided by the Information Bottleneck Theory

- **How to find the local minima?**
  Neuron Campaign Initialization algorithm

Information Bottleneck Theory and measurement



- Input information maintenance:
  $$I(X;Z_i)$$
- Target-related information enhancement
  $$I(Z_i;Y)$$
- Criterion:
  $$\alpha I(X;Z_i) + (1-\alpha) I(Z_i;Y)$$
- The front layer should focus more on input information maintenance

Criteria Simplification with High Efficiency

- $I(X;Z_i)$ input information maintenance criterion

  $I(X;Z_i)$= H(Z) – H(Z|X) = H(Z)

$$tr(\Sigma_i)$$

  where Σ_i is the covariance matrix of Z_i

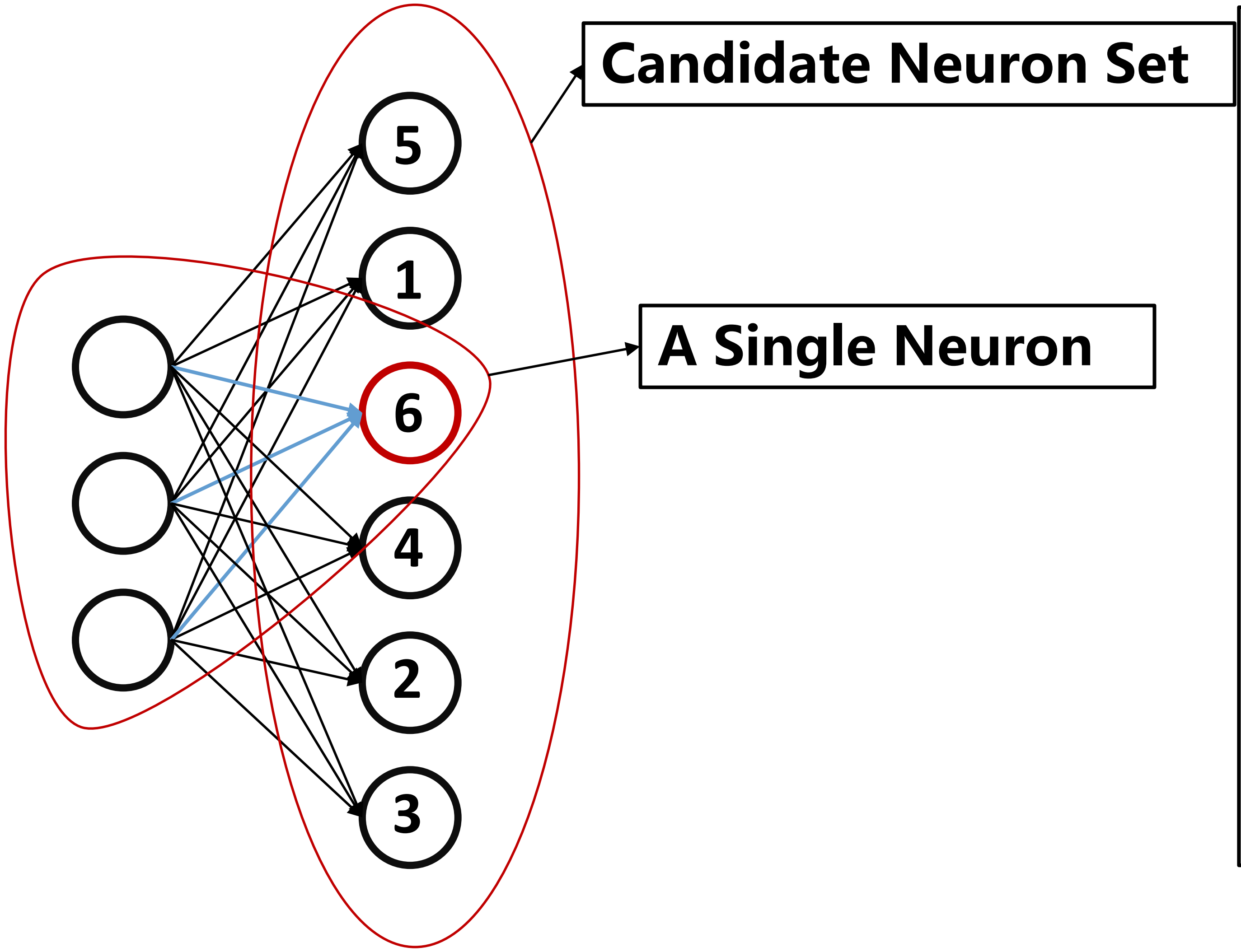
- $I(Z_i;Y)$ target-related maintenance criterion

$$tr\left(\widehat{H}\widehat{H}^T\right) - \frac{1}{N}\sum_{j=0}^{N} tr\left(\hat{Z}_i^T \Pi^j \hat{Z}_i\right)$$

Inter-class variance

Intra-class variance

Neuron Campaign Initialization algorithm



Output: weight with size [3, 3]

- Pre-Initialize a large neuron set with size [3, 6]
- Select neuron with well-designed criteria (based on IB) and **orthogonal** score
- Combine neurons as initial weight

Algorithm details

$X{:}[60000, 784] \xrightarrow{W} Z{:}[60000, 1000] \longrightarrow s{:}[1000]$

$X{:}[60000, 100] \xrightarrow{W} Z{:}[60000, 200] \longrightarrow s{:}[200]$

Algorithm 1 Neuron Campaign initialization algorithm

Input: Candidate weight matrix $\mathbf{W}$ [784, 1000]

1: **for** t=1 to T **do** T = 100

2: Update generalized orthnormalization matrix at t steps: $\mathbf{A}_t = (\mathbf{A}_{t-1}, \mathbf{a}_t^T)^T$ [784, t]

3: Calculate the null space projection by $\mathbf{P}_t = \mathbf{P}_{t-1} - \mathbf{a}_t\mathbf{a}_t^T\mathbf{W}$

4: Select optimal neuron whose index is chosen by $i = \max_i s_i \frac{||\mathbf{p}_t^i||}{||\mathbf{W}_i||}$

5: Update $\mathbf{w}^* = \mathbf{W}_{\cdot,i}$

6: Normalize basis of the generalized orthnormalization matrix as $\mathbf{a}_{\mathbf{t+1}} = \mathbf{p}_t^i/||\mathbf{p}_t^i||$

7: **end for**

Output: Winning neurons formed weight matrix $\mathbf{W}'$ [784, 100]

Ensure orthogonality of the selected neurons

Select the neuron with largest score

Evaluation

Experimental details

Table 1: minimal error rate and corresponding epoch comparison of IBCI with baseline methods on MNIST.

Strategy	Layers	Vanilla	LSUV	**IBCI**
Xavier	2	2.04 ± 0.03 (75)	2.05 ± 0.06 (51)	$\mathbf{1.93 \pm 0.06}$ **(60)**
	3	1.82 ± 0.05 (52)	1.80 ± 0.07 (63)	$\mathbf{1.71 \pm 0.09}$ **(36)**
	5	2.83 ± 0.16 (98)	3.13 ± 0.17 (69)	$\mathbf{2.53 \pm 0.09}$ **(78)**
He	2	2.03 ± 0.03 (65)	2.00 ± 0.04 (70)	$\mathbf{1.93 \pm 0.07}$ **(57)**
	3	1.83 ± 0.05 (54)	1.86 ± 0.07 (71)	$\mathbf{1.73 \pm 0.04}$ **(35)**
	5	2.76 ± 0.07 (80)	2.90 ± 0.12 (77)	$\mathbf{2.62 \pm 0.08}$ **(73)**

Hidden layer dimension setting

layers	Hidden Layer Dimension
2	784, 100, 10
3	784, 256, 100, 10
5	784, 32, 32, 32, 32, 10

Ablation Study

Table 2: minimal error rate and corresponding epoch comparison of IBCI with methods with only one criterion.

Strategy	Layers	IBCI	TIE	IIM
Xavier	2	**1.93 ± 0.06 (60)**	2.04 ± 0.07 (58)	2.07 ± 0.09 (84)
	3	**1.71 ± 0.09 (36)**	1.82 ± 0.03 (43)	1.82 ± 0.05 (52)
	5	**2.53 ± 0.09 (78)**	2.68 ± 0.05 (82)	2.57 ± 0.09 (84)
He	2	**1.93 ± 0.07 (57)**	2.07 ± 0.06 (59)	2.034 ± 0.09 (62)
	3	**1.73 ± 0.04 (35)**	1.83 ± 0.07 (42)	1.856 ± 0.05 (55)
	5	**2.62± 0.08 (73)**	2.89 ± 0.11 (74)	2.67 ± 0.12 (86)

Target Information Enhancement, i.e., IBCI without IIM

Input Information Maximization , i.e., IBCI without TIE

Microsoft

Conclusion & Future work

Conclusion & Future Work

Conclusion

- Introduce the Information Bottleneck Theory into practice use.
- Propose a novel and interesting neuron campaign initialization algorithm.

Future work

- Introduce to broader neural network architectures.
- Can we help to understand the recent popular initialization with pretrain?

Further Exploration

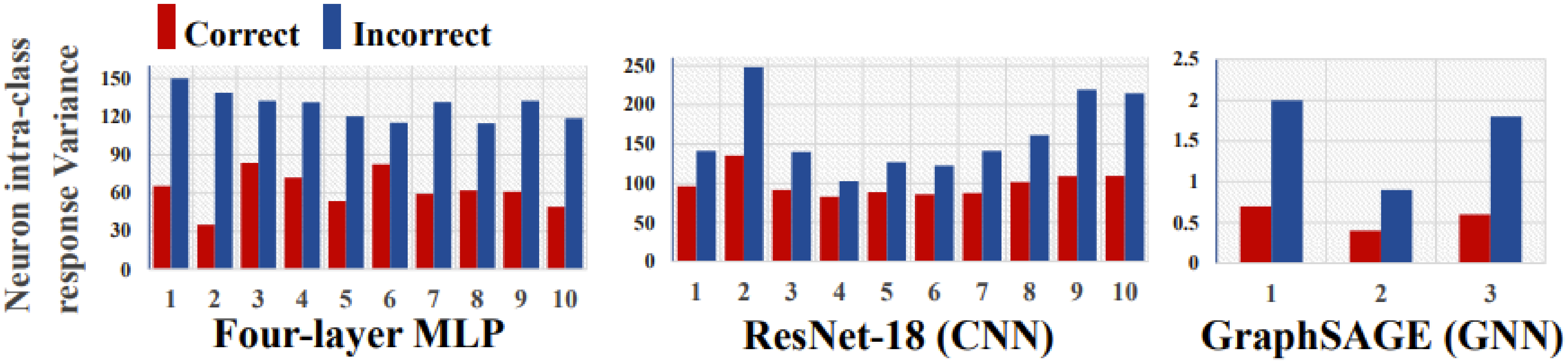
Neuron with Steady Response Leads to Better Generalization

Qiang Fu[1*], Lun Du[1*], Haitao Mao[1,2*], Xu Chen[1,3], Wei Fang[1,4] ,Shi Han[1], Dongmei Zhang[1]

1. Microsoft Research Asia
2. University of Electronic Science and Technology of China
3. Peking University
4. Tsinghua University

Observation 1

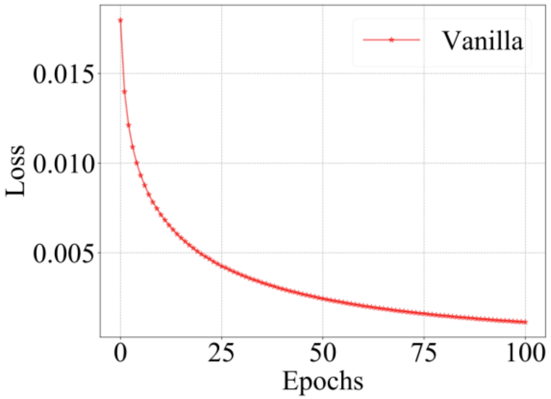
- Intra-class response variance of correctly classified samples is smaller than that of misclassified ones on arbitrary class
- Smaller intra-class response variance leads to better generalization



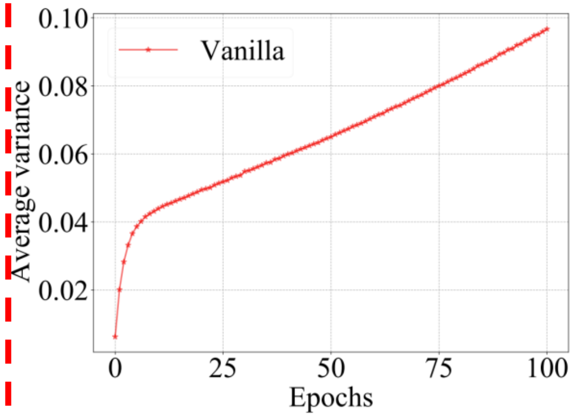
The horizontal axis and the vertical axis represent class indexes and the value of intra-class response variance, respectively.
Each bar represents the intra-class response variance aggregated from all neurons in the penultimate layer.

Observation 2

- Does cross entropy control intra-class response variance well? No!
- The ascending intra-class response variance shows the potential improvement space for the regularization



(a) Training cross-entropy loss



(b) Training intra-class variance

Ascending Variance!

Training procedure of vanilla four-layer MLP on MNIST

Key Insight

- Neuron with small intra-class responses variance can lead to better generalization
- Cross entropy can NOT control intra-class response variance well

Regularization on intra-class response variance is needed!

Neuron Steadiness Regularization (NSR)

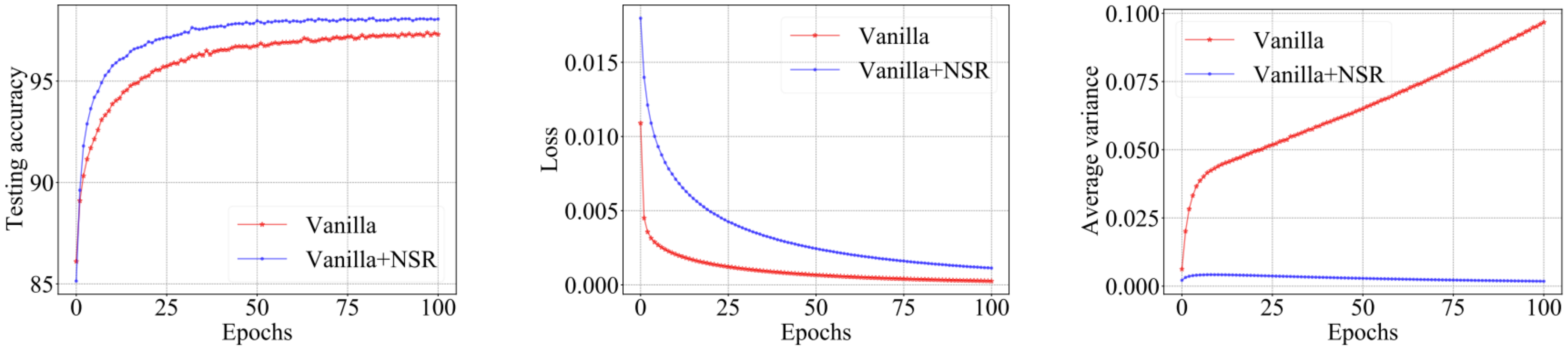
- We propose a new regularization method called NSR
  - NSR is the first work to encode inductive bias from the perspective of **class-dependent response distribution** of individual neurons
- NSR improves generalization by controlling neuron intra-class response variance
  - Significant improvement on MLP, CNN, and GNN
  - Bigger improvement than typical regularizations like L1/L2/Jacobian
  - Further gain when combining with Batch Normalization and Dropout
- NSR has low overhead on both memory and computation

Evaluation Setting

Network Architecture	Vanilla model	Dataset	Optimization
Multiplayer Perceptron	MLP-3,4,6,8,10	MNIST	SGD
Convolutional Neural Network	ResNet-18	CIFAR-10	Momentum
	VGG-19		
	ResNet-50	ImageNet	Adam
Graph Neural Network	GraphSAGE	WikiCS, PubMed, Amazon-Photo, Computers	Adam
	GCN		

Dynamics of Training & Testing

- Neuron intra-class response variance is growing larger in vanilla model
- NSR could control neuron intra-class response variance well
- NSR has higher testing accuracy although its cross-entropy loss is even larger



(a) Testing accuracy (b) Training cross-entropy loss (c) Training intra-class variance

Training procedure of vanilla four-layer MLP and four-layer MLP with our Neuron Steadiness Regularization on MNIST

Some advice for undergraduate Research

- 培养感恩和善良的心
- 失败是常态, 学会去面对
- 打好机器学习基础&写一写博客，对社区有一些贡献
- shoot low, aim high
- 关注心理健康, 学会自我调节
  - 书籍：活出心花怒放的人生
  - Up主： 是慢慢丫

实习生、PHD position

微软亚洲研究院, DKI 数据分析组

- 导师
  - 杜仑 Researcher
  - 付强 Principal Researcher
- 联系方式
  - lun.du@microsoft.com
- 研究方向
  - 神经网络基础研究
  - 图神经网络
  - 推荐系统
  - 代码理解
  - 表格数据分析

马耀 新泽西理工

Yao.ma@njit.edu

Tyler Derr 范德堡大学

tyler.derr@vanderbilt.edu

范文琦 香港理工大学

wenqifan@polyu.edu.hk

赵翔宇 香港城市大学

zhaoxi35@msu.edu

Microsoft

Thanks & QA

Reference

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification.
- Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 249–256
- Ravid Shwartz-Ziv and Naftali Tishby. 2017. Opening the black box of deep neural networks via information. arXiv Prepreint arXiv:1703.00810 (2017)
- Dmytro Mishkin and Jiri Matas. 2015. All you need is a good init. 2017 IEEE Conference on Computer Vision and Pattern Recognition.