Microsoft

Neuron With Steady Response Leads to Better Generalization

Haitao Mao[2]

Joint work with Qiang Fu[1], Lun Du[1], Xu Chen[1], Wei Fang[3], Shi Han[1], Dongmei Zhang[1]

1. Microsoft Research Asia
2. Michigan State University
3. Tsinghua University

Existing Regularizations

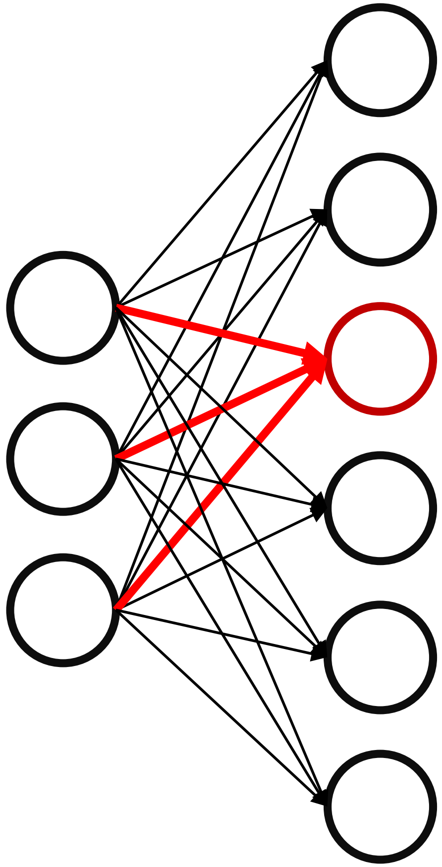
Inductive bias	Example	Principle	Leveraged information
Scale	L2	Penalize large norms of model weights	Weights
Sparseness	L1	Reward zero neuron response	Collective neuron responses
Smoothness	Jacobian	Penalize big change with small perturbation	Mapping function derivatives
Diversity	Orthogonalization	Reduce feature correlations	Weight correlations

Limitations:

- Individual neurons are lack of “global view” of its response distribution on different classes
- Neurons are only aware of responses of current mini-batch, which may contain noise and be instable

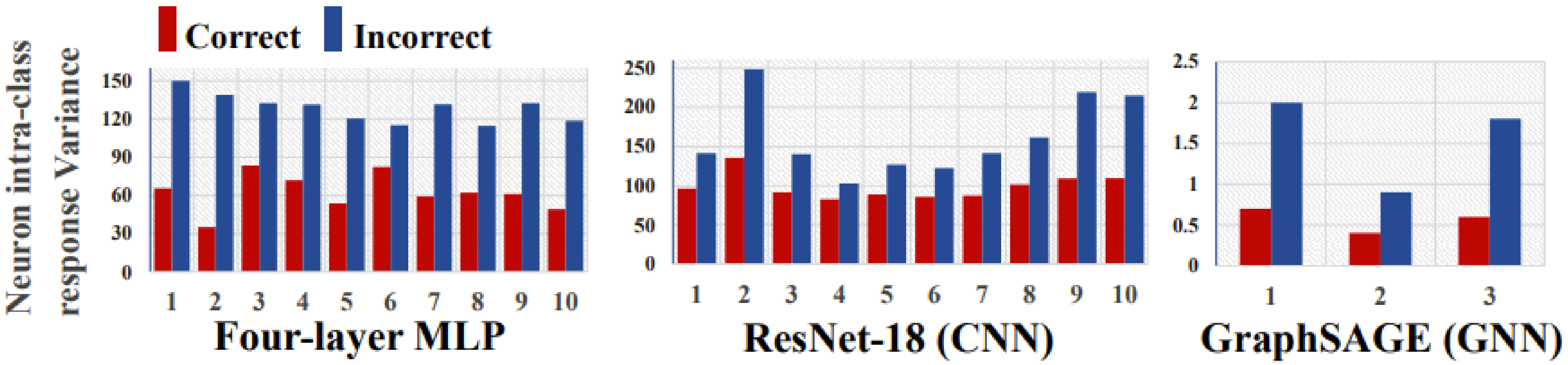
Observations

--from neuron perspective



Observation 1

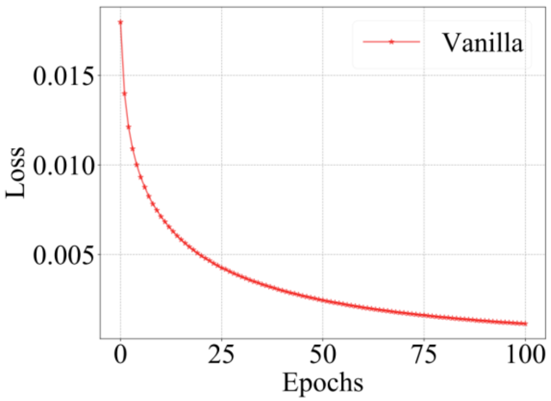
- Intra-class response variance of correctly classified samples is smaller than that of misclassified ones on arbitrary class
- Smaller intra-class response variance leads to better generalization



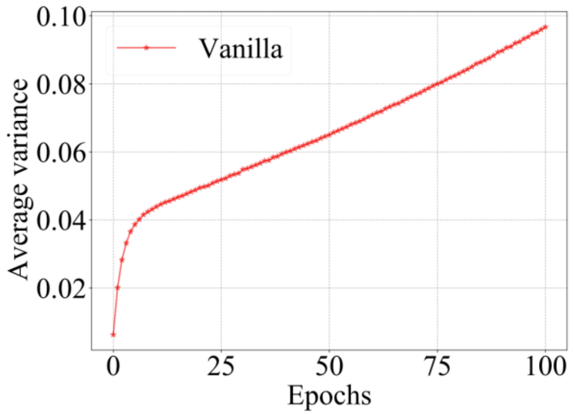
The horizontal axis and the vertical axis represent class indexes and the value of intra-class response variance, respectively. Each bar represents the intra-class response variance aggregated from all neurons in the penultimate layer.

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 (**steadiness**) 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)

- Final regularized loss function

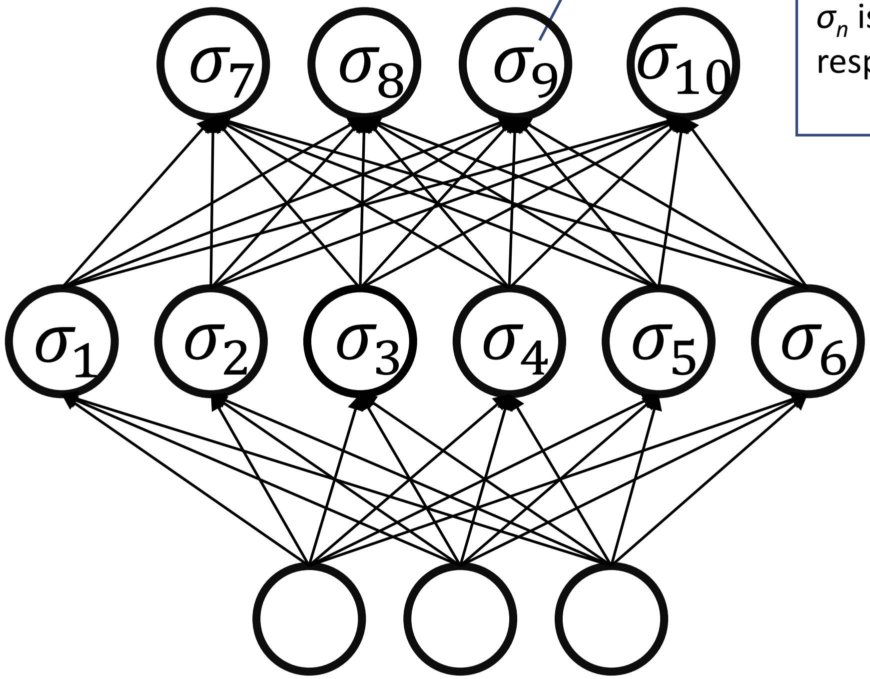
$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_S$$

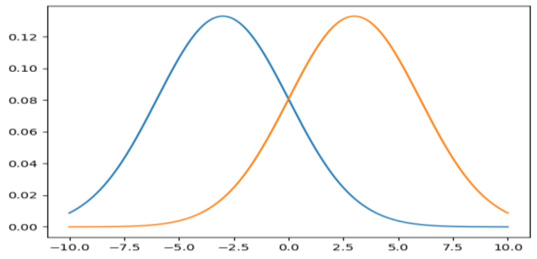
L_C is cross entropy loss

L_S is NSR loss of the model

- NSR loss of the model

$$\mathcal{L}_S = \sum_{n=1}^{N} \lambda_n \sigma_n$$





σ_n is calculated by the n_{th} neuron's response distribution of classes

Evaluation Setting

Multiple datasets and architectures

<table>
<tr><th>Network Architecture</th><th>Vanilla model</th><th>Dataset</th><th>Optimization</th></tr>
<tr><td>Multiple Layer Perceptron</td><td>MLP-3,4,6,8,10</td><td>MNIST</td><td>SGD</td></tr>
<tr><td rowspan="3">Convolutional Neural Network</td><td>ResNet-18</td><td rowspan="2">CIFAR-10</td><td rowspan="2">Momentum</td></tr>
<tr><td>VGG-19</td></tr>
<tr><td>ResNet-50</td><td>ImageNet</td><td>Adam</td></tr>
<tr><td rowspan="2">Graph Neural Network</td><td>GraphSAGE</td><td rowspan="2">WikiCS, PubMed, Amazon-Photo, Computers</td><td rowspan="2">Adam</td></tr>
<tr><td>GCN</td></tr>
</table>

Performance of NSR on MLP & CNN

Model	MLP-3	MLP-4	MLP-6	MLP-8	MLP-10	ResNet-18	VGG-19
Vanilla	3.09 ± 0.10	2.29 ± 0.07	2.44 ± 0.09	2.87 ± 0.09	3.06 ± 0.06	7.96 ± 0.12	10.57 ± 0.17
Vanilla+NSR	2.80 ± 0.08	1.64 ± 0.04	1.76 ± 0.06	1.98 ± 0.09	1.72 ± 0.14	7.20 ± 0.09	8.77 ± 0.10
Gain of NSR	9.39%	28.38%	27.87%	30.87%	43.79%	9.55%	17.03%

Model	ResNet-18	VGG-19	ResNet-50
Vanilla	4.22 ± 0.07	9.19 ± 0.18	7.82 ± 0.09
Vanilla+NSR	3.74 ± 0.08	8.09 ± 0.17	6.98 ± 0.08
Gain of NSR	11.37%	11.97%	10.74%

Thanks!