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

Contents

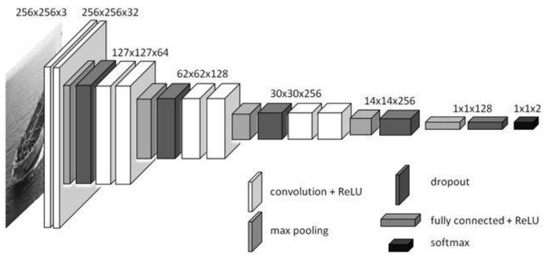
- Background
- Our paper
  - Observations
  - Methodology
  - Evaluation
- Future discussion

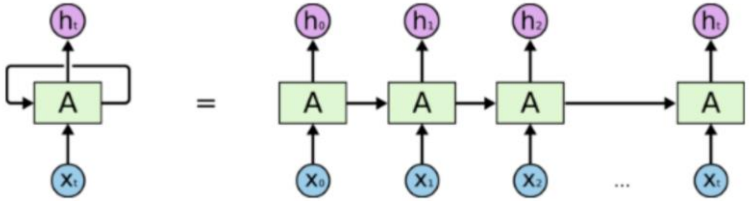
Microsoft

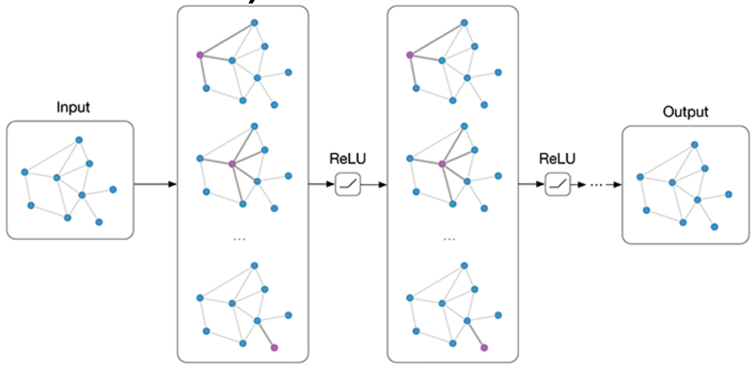
Background

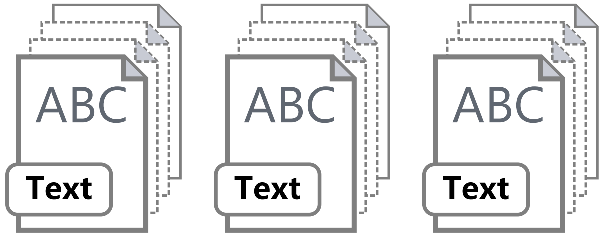
Problem Definition

- Design domain-generic architecture-agnostic regularization to improve generalization of deep neural network
  - Diverse domains: image, text, graph, time series, …
  - Different architectures: MLP, CNN, GNN, RNN, Transformer, …









Generalization Techniques

- Data augmentation
  - Image distortion
  - Noise generation
  - Adversarial examples
- Network architecture
  - Pooling
  - Weight sharing
  - Batch normalization
- Optimization procedure
  - Dropout
  - Stochastic dropout
  - Early stop
- Regularization
  - L1/L2 regularization
  - Jacobian regularization
  - Orthogonality regularizations
- Learning paradigm
  - Pre-training + fine tuning
  - Transfer learning
  - Meta learning

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

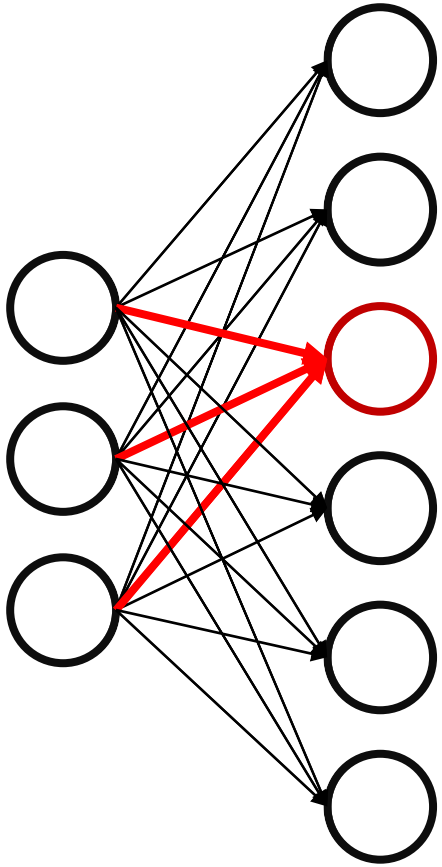
Limitations:

- 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

Microsoft

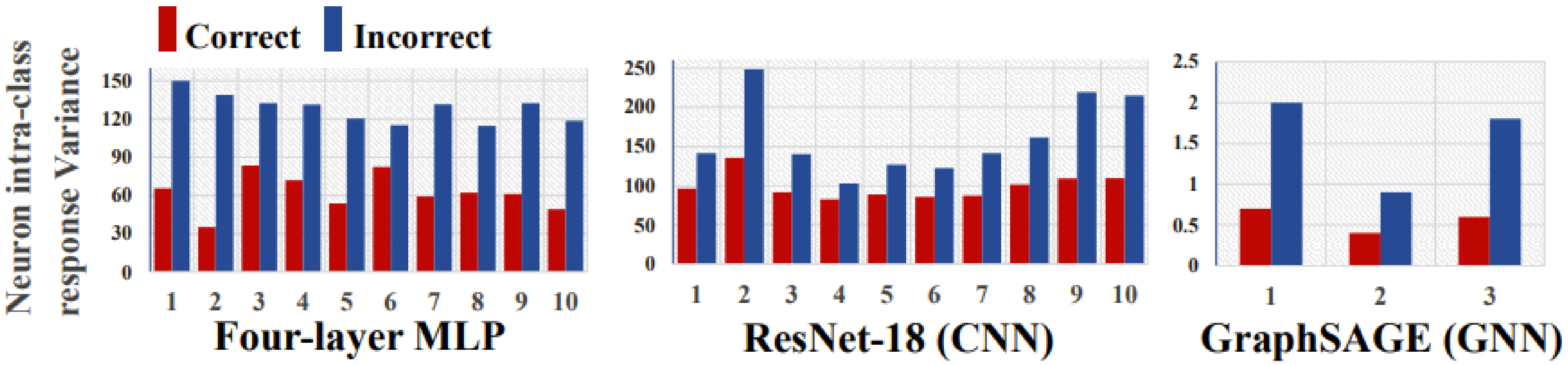
Observations

--from neuron perspective



Observation 1

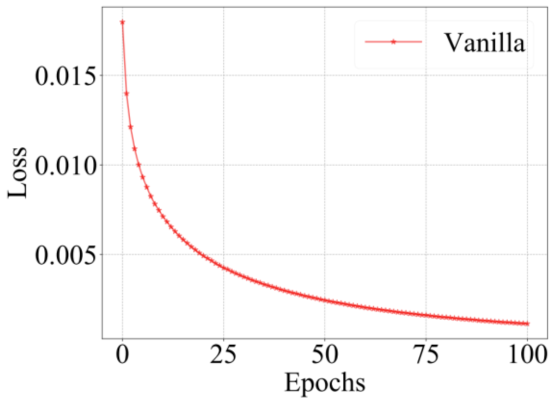
- Intra-class response variance of correctly classified samples is smaller than that of misclassified ones on arbitrary class
- Smaller intra-class response variance leads to better generalization



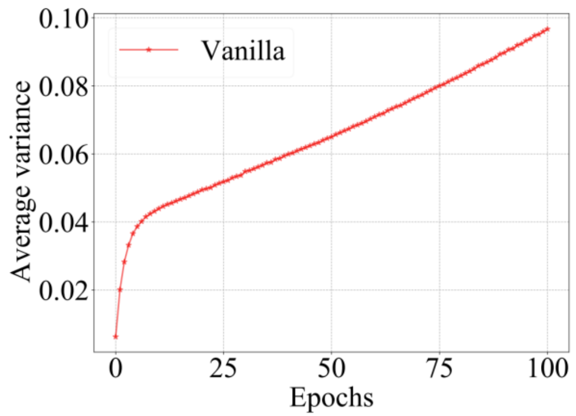
The horizontal axis and the vertical axis represent class indexes and the value of intra-class response variance, respectively. Each bar represents the intra-class response variance aggregated from all neurons in the penultimate layer.

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)

-- class-dependent response distribution of **individual neurons**

Overall Formulation

- Final regularized loss function

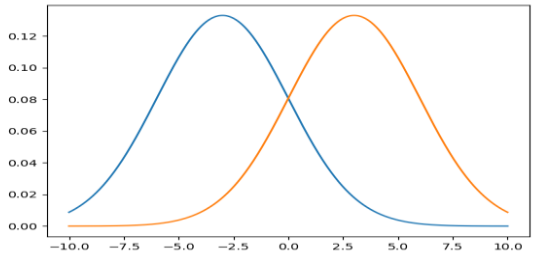
$$\mathcal{L} = \mathcal{L}_C + \mathcal{L}_S$$

L_C is cross entropy loss

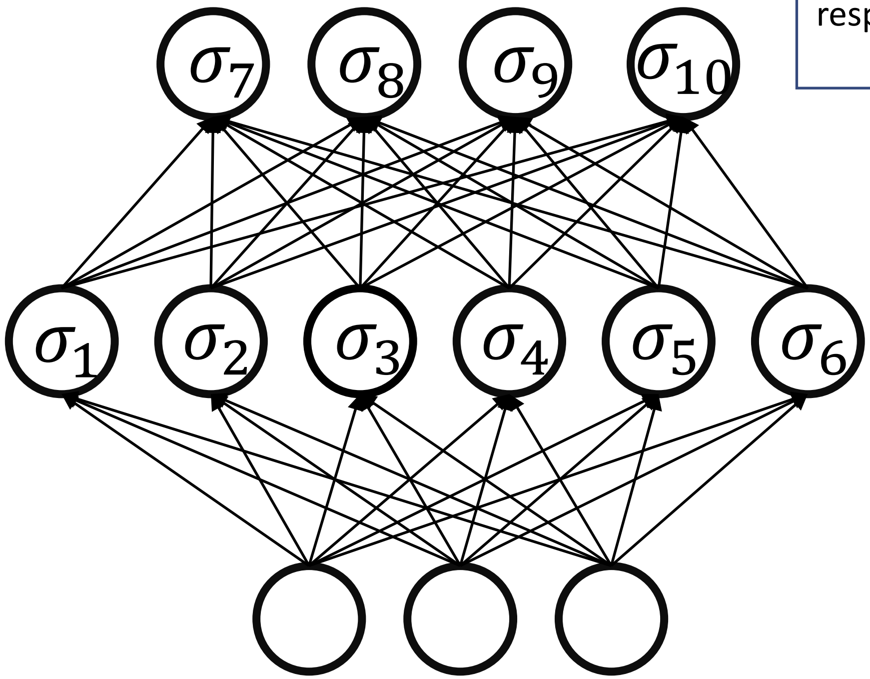
L_S is NSR loss of the model

- 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

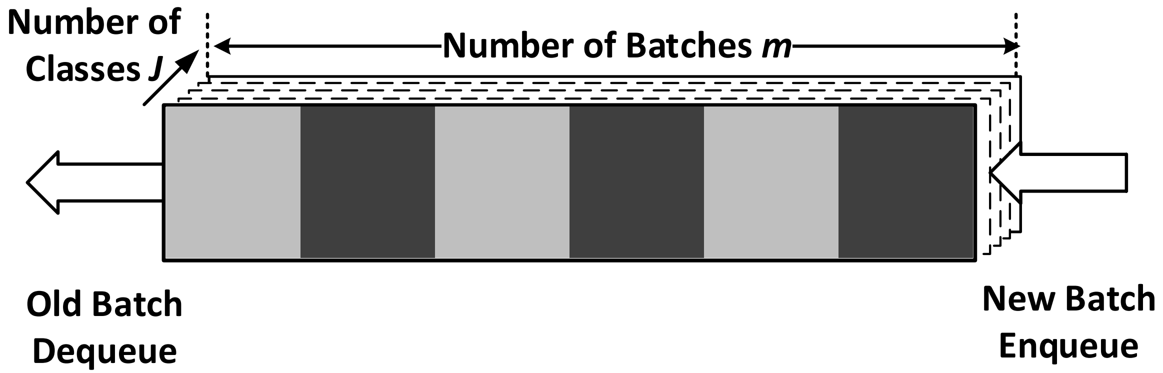


Single Neuron Implementation

$$
\begin{aligned}
\sigma_n &= \sum_{j=1}^{J} \alpha_j \cdot \mathrm{Var}\left(X_{n,j}\right) \\
&= \sum_{j=1}^{J} \alpha_j \cdot \mathbb{E}\left[\left(X_{n,j} - \mathbb{E}\left[X_{n,j}\right]\right)^2\right] \\
&= \mathbb{E}\left[\sum_{j}^{J} \alpha_j X_{n,j}^2\right] - \sum_{j}^{J} \alpha_j \mathbb{E}^2\left[X_{n,j}\right]
\end{aligned}
$$

Term 1 is calculated by current mini-batch

Term 2 is calculated by recent *m* mini-batch

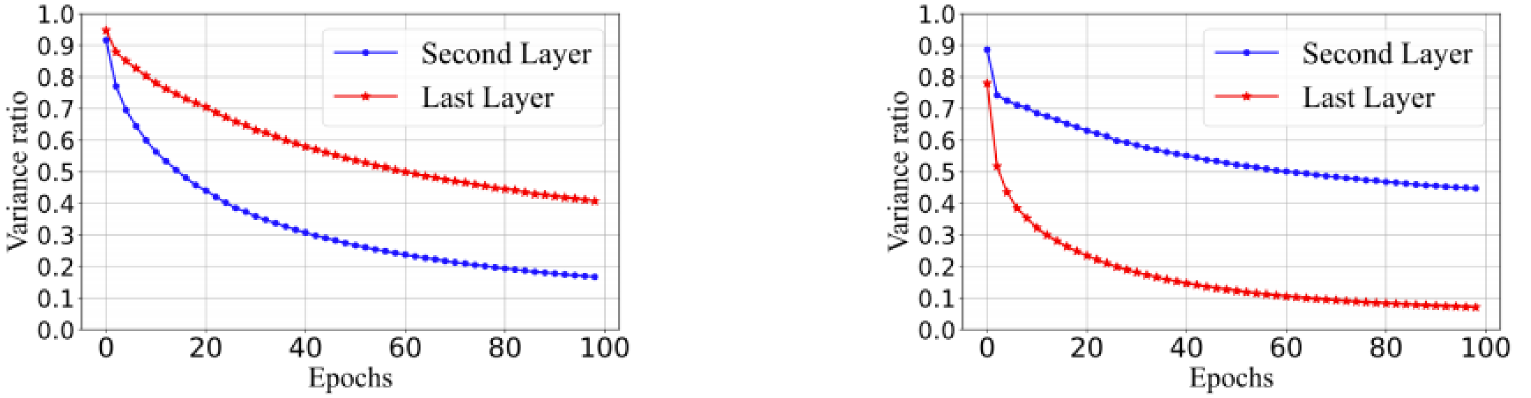


$X_{n,j}$ denotes the n_{th} neuron's response for samples of j_{th} class; α_j is the weight of j_{th} class

Simplification on applying NSR

We do simplifications for better trade-off between gain and overhead

- ❑We apply NSR to only one specific layer
- ❑We use the same value of λ for all neurons in the same layer
- ❑We select the layer with biggest aggregated variance to apply NSR



(a) NSR only applied on the second layer (b) NSR only applied on the last layer

The variance ratio is the neuron intra-class response variance of the three-layer MLP applied with NSR, (a) only on the second layer and (b) only on the last layer, divided by the corresponding variance of the vanilla three-layer MLP

Theorem Guarantee

Lemma 3.1. *For a multi-class classification problem, when (1) a deep learning model utilizing the Cross Entropy loss with an NSR regularization term on any of its intermediate layer is optimized via gradient descent and (2) the capacity of the model is sufficiently large, the Consistency of Representations Complexity Measure on this model $\mathcal{C}$ will tend to be 0. Under the same condition, when the deep learning model is only optimized by the Cross Entropy loss without an NSR regularization term, there will be infinite local minima where the complexity measure $\mathcal{C}$ will be a positive number.*

Complexity Measure: Consistency of representation:

$$\mathcal{C} = \frac{1}{J}\sum_{i=1}^{J} \max_{i \neq j} \frac{S_i + S_j}{M_{i,j}}$$

Microsoft

Evaluation

- How does NSR work on various datasets and different neural architectures?
- Does NSR outperform other classical regularization methods?
- What is the effect of combining NSR with other popular methods like Batch Normalization or Dropout?
- Which layer(s) should NSR be applied with?

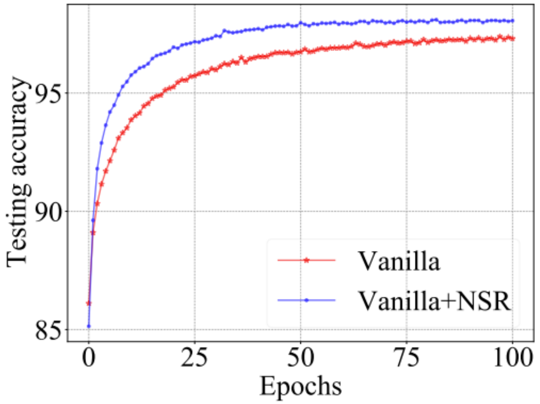
Evaluation Setting

Multiple datasets and architectures

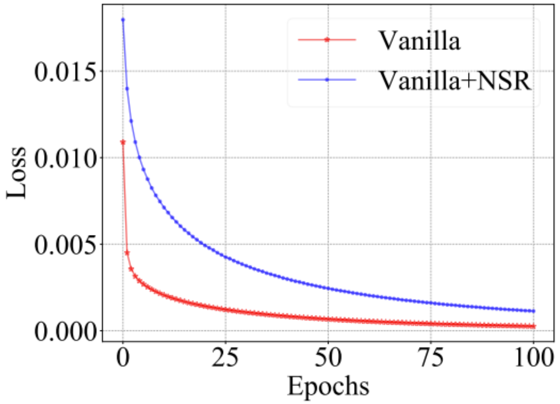
Network Architecture	Vanilla model	Dataset	Optimization
Multiple Layer Perceptron	MLP-3,4,6,8,10	MNIST	SGD
Convolutional Neural Network	ResNet-18	CIFAR-10	Momentum
	VGG-19		
	ResNet-50	ImageNet	Adam
Graph Neural Network	GraphSAGE	WikiCS, PubMed, Amazon-Photo, Computers	Adam
	GCN		

RQ1: NSR effect on training and testing

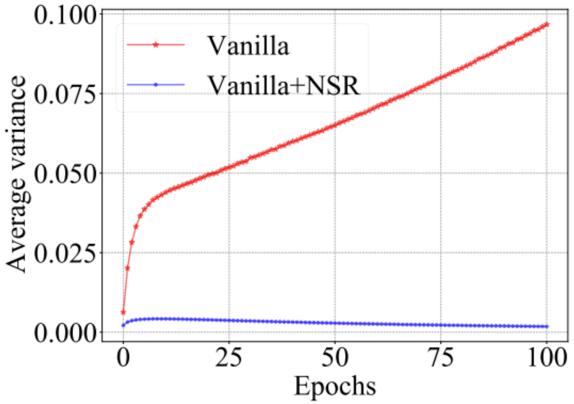
- Neuron intra-class response variance is growing larger in vanilla model
- NSR could control neuron intra-class response variance well
- NSR has higher testing accuracy although its cross-entropy loss is even larger



(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

RQ1: 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%

RQ1: Performance of NSR on GNN

Dataset	Model	GraphSAGE-2	GraphSAGE-3	GraphSAGE-4	GCN-2	GCN-3	GCN-4
PubMed	Vanilla	10.73 ± 0.06	10.20 ± 0.25	10.43 ± 0.17	12.02 ± 0.00	12.76 ± 0.18	14.01 ± 0.07
	Vanilla+NSR	9.89 ± 0.08	9.48 ± 0.12	9.79 ± 0.19	11.92 ± 0.00	12.19 ± 0.11	12.96 ± 0.08
	Gain	7.83%	7.06%	6.14%	0.83%	4.47%	7.49%
Amazon-Photo	Vanilla	5.82 ± 0.00	5.20 ± 0.14	6.37 ± 0.30	6.73 ± 0.00	8.00 ± 0.11	10.24 ± 0.14
	Vanilla+NSR	4.54 ± 0.10	4.86 ± 0.13	5.62 ± 0.59	6.27 ± 0.00	7.96 ± 0.10	9.03 ± 0.25
	Gain	21.99%	6.54%	11.77%	6.84%	0.50%	11.82%
Amazon-Computers	Vanilla	11.37 ± 0.55	11.88 ± 1.05	15.49 ± 0.90	12.17 ± 0.07	14.90 ± 0.25	18.07 ± 0.74
	Vanilla+NSR	10.47 ± 0.05	10.22 ± 0.54	12.86 ± 0.82	10.86 ± 0.03	13.66 ± 0.12	16.02 ± 0.23
	Gain	7.92%	13.97%	16.98%	10.76%	8.32%	11.34%
WikiCS	Vanilla	16.81 ± 0.21	15.97 ± 0.18	16.63 ± 0.31	18.41 ± 0.06	18.66 ± 0.23	19.21 ± 0.31
	Vanilla+NSR	16.06 ± 0.33	15.27 ± 0.21	15.43 ± 0.24	17.99 ± 0.05	18.10 ± 0.27	18.84 ± 0.26
	Gain	4.46%	4.38%	7.22%	2.28%	3.00%	1.93%
Average Gain of NSR		10.55%	7.99%	10.53%	5.18%	4.07%	8.15%

RQ2: Comparison with Other Regularization

Regularization	MLP-4	ResNet-18	GraphSAGE
Vanilla	2.29 ± 0.07	7.96 ± 0.12	11.37 ± 0.55
L1	2.27 ± 0.05	7.83 ± 0.23	10.81 ± 0.13
L2	2.27 ± 0.05	7.67 ± 0.18	10.68 ± 0.35
Jacobian	2.21 ± 0.04	7.90 ± 0.07	11.27 ± 0.45
NSR	1.64 ± 0.04	7.20 ± 0.09	10.52 ± 0.22

Hyper-parameter λ in every method is tunned with the same NNI setting

RQ3: Combination with Other Techniques

MLP-4	Vanilla	Vanilla + BN	Vanilla + BN + NSR
Error rate	2.29 ± 0.07	2.22 ± 0.04	1.62 ± 0.08

MLP-4	Vanilla	Vanilla + Dropout	Vanilla + Dropout + NSR
Error rate	2.29 ± 0.07	2.19 ± 0.04	1.64 ± 0.04

RQ4: Effect of NSR on Different Layers

- The accuracy is improved no matter which layer is applied with NSR
- Applying NSR to the layer with the biggest variance could
  - obtain the most significant gain compared with other layers
  - achieve similar accuracy compared with the best one

Method	Error rate
MLP	2.29 ± 0.07
MLP_2	2.22 ± 0.08
MLP_3	1.90 ± 0.13
MLP_4	1.64 ± 0.04
$MLP_{3,4}$	1.63 ± 0.08

Intra-class variance of layer 2,3,4 is 409, 510, and 1660

Method	Error rate
GraphSAGE	11.37 ± 0.55
$GraphSAGE_1$	10.47 ± 0.05
$GraphSAGE_2$	10.52 ± 0.22
$GraphSAGE_{1,2}$	10.30 ± 0.17

Intra-class variance of layer 1,2 is 4.15 and 2.68

Future Work

Future Work

- Explore hyper-parameter setting strategies
  - Adaptive setting λ along with training procedure
  - Setting λ according to neuron significance
- Explore other statistics based on neuron response distribution
  - Inter-class response distance
  - Adjacent class response distance
- Adapt NSR to broader architectures and tasks
  - Transformer, RNN
  - NLP tasks, time series related tasks
- Analyze on the response relationship between Neuron

Microsoft

Thanks & QA