The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Knowledge Distillation with Adversarial Samples Supporting Decision Boundary

Byeongho Heo,[1*] **Minsik Lee,**[2*] **Sangdoo Yun,**[3] **Jin Young Choi**[1]
{bhheo, jychoi}@snu.ac.kr, mleepaper@hanyang.ac.kr, sangdoo.yun@navercorp.com
[1]Department of ECE, ASRI, Seoul National University, Korea
[2]Division of EE, Hanyang University, Korea
[3]Clova AI Research, NAVER Corp, Korea

AAAI 2019

Contents

- Knowledge Distillation
- Adversarial Samples
- Method
- Experiments

https://zhuanlan.zhihu.com/p/215734661

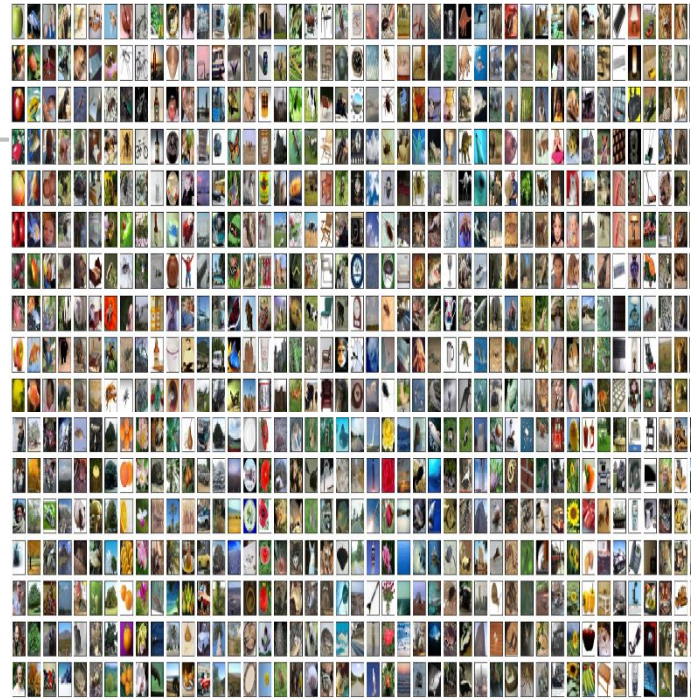
Knowledge Distillation

- In order to get better performance, we can use
  - More complex neural network (over-parameterization)
  - Ensemble learning

huge computation cost!

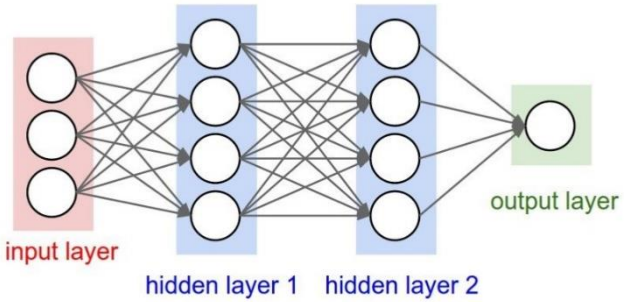
- In real-world tasks, we want
  - Small model with low complexity
  - Comparable performance

Knowledge Distillation



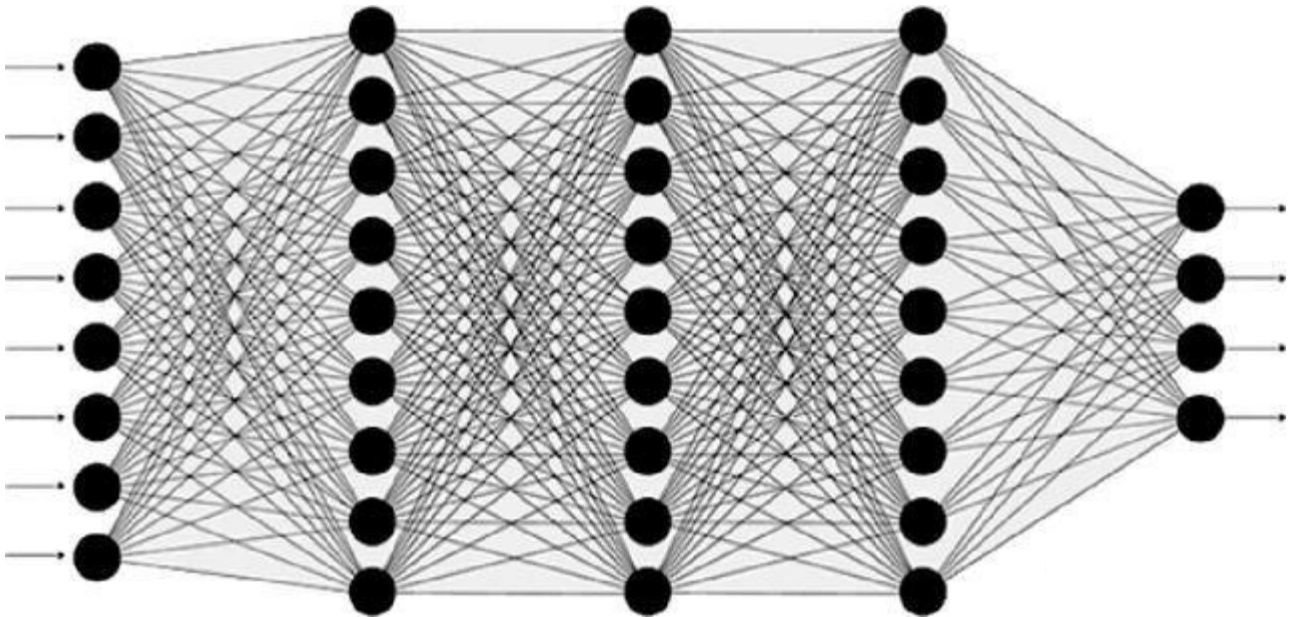
train

67%

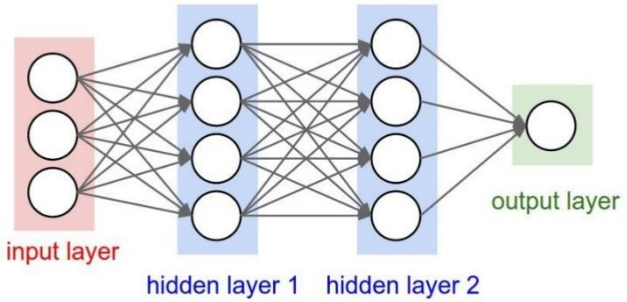


X

98%



95%



✓

Knowledge Distillation

- How to distillation? [1]
  - A transfer set
  - Minimize the cross-entropy of the distribution produced by two models

$$min - p * \log(q)$$

- Why is it called distillation?

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

$T \to 0, q_i \to one\ hot$
$T \to \infty, q_i \to softer$

  - T is a temperature that is normally set to 1. Using a higher value for T produces a softer probability distribution over classes.
  - When the temperature T is increased, knowledge is transferred to the distilled model. After it has been trained, it uses a temperature of 1.

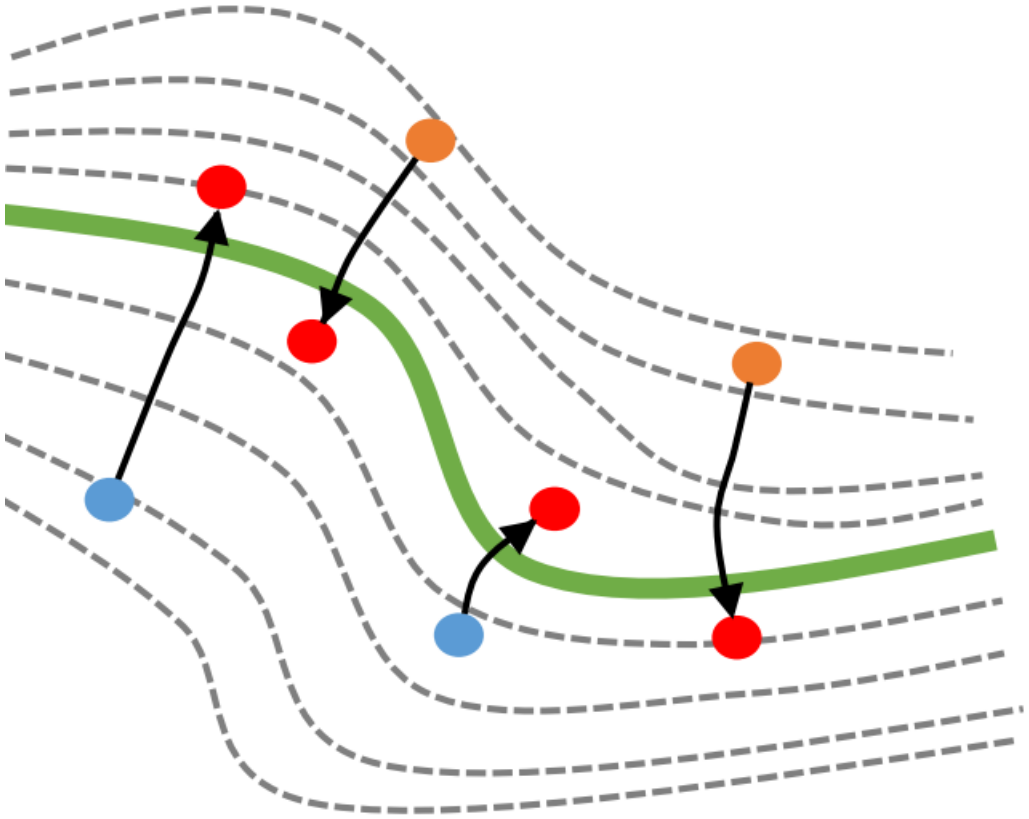
[1] Distilling the Knowledge in a Neural Network. (NIPS'15)

Adversarial Samples

$$x + \epsilon \operatorname{sign}\left(\nabla_x \mathcal{L}_{\mathrm{CE}}(f(x), y)\right) = x_{adv}$$

$$\max_{\|x' - x\| \leq \epsilon} \quad \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} [\mathcal{L}(f_\theta(x'), y)]$$

Adversarial Samples



Adversarial attack

$$x_{adv} = x + \epsilon \operatorname{sign}\left(\nabla_x \mathcal{L}_{\mathrm{CE}}(f(x), y)\right)$$

$$\max_{\|x'-x\| \le \epsilon} \quad \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}} [\mathcal{L}(f_\theta(x'), y)]$$

Method

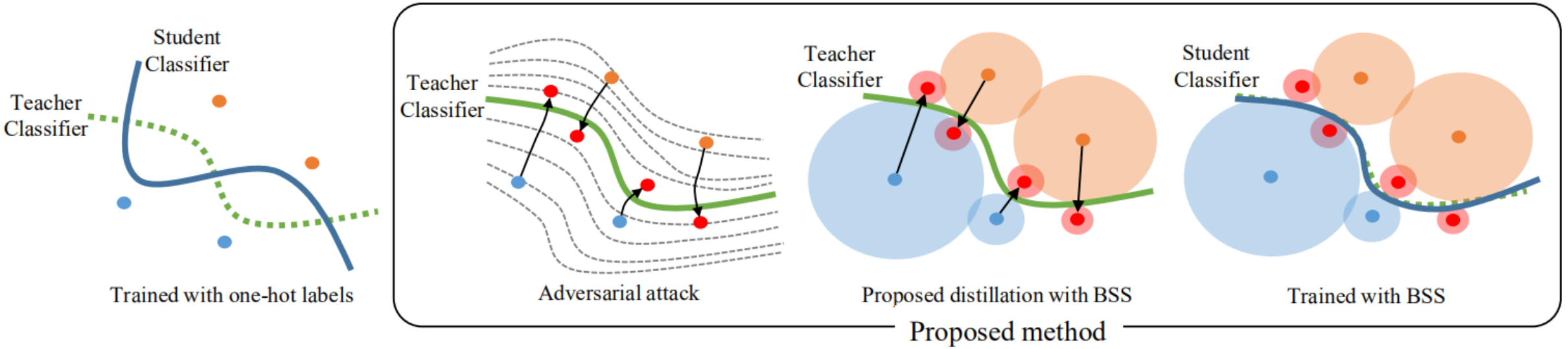


Figure 1: The concept of knowledge distillation using samples close to the decision boundary. The dots in the figure represent the training sample and the circle around a dot represents the distance to the nearest decision boundary. The samples close to the decision boundary enable more accurate knowledge transfer.

- The generalization performance of a classifier highly depends on how well the classifier learns the true decision boundary between the actual class distributions
- The key idea is about using adversarial samples near a decision boundary to transfer the knowledge related with the boundary.

Method

- Iterative Scheme to find a BBS.
  - BBS: boundary supporting samples.

- Knowledge distillation using BBS

Method

- Iterative Scheme to find a BBS.
  - The classifier f can produce classification scores for all classes, $f_b(x)$ and $f_k(x)$ denote the classification score for the base class and target class, respectively.
  - The adversarial attack is to decrease $f_b(x)$ while increasing $f_k(x)$.
  - The loss function for adversarial attack

$$\mathcal{L}_{\mathrm{k}}(\mathrm{x}) = f_b(x) - f_k(x)$$

This loss becomes **zero** at a point on the decision boundary, **positive** at a point within the base class, and **negative** at an adversarial point within the target class.

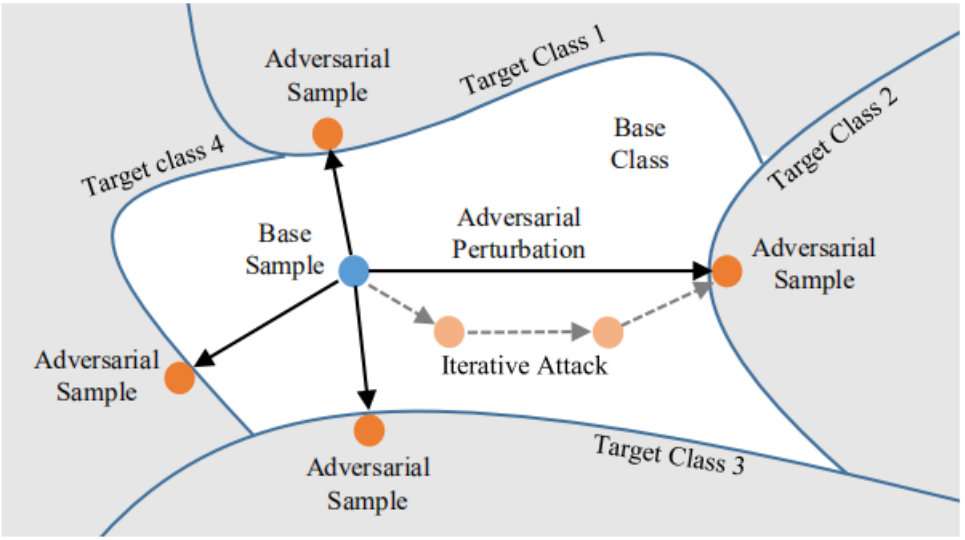


Figure 2: Iterative scheme to find BSSs for a base sample

Method

- Knowledge distillation using BBS

$$J(\boldsymbol{a}, \boldsymbol{b}) = -\boldsymbol{a}^T \log \boldsymbol{b}$$

$$\mathcal{L}(n) = \mathcal{L}_{cls}(n) + \alpha \mathcal{L}_{KD}(n) + \beta \sum_{k}^{K} p_n^k \mathcal{L}_{BS}(n, k).$$

$$\mathcal{L}_{cls}(n) = J(\boldsymbol{y}_n^{true}, \sigma\left(f_s(\boldsymbol{x}_n)\right)),$$

$$\mathcal{L}_{KD}(n) = J(\sigma\left(\frac{f_{\mathrm{t}}(\boldsymbol{x}_n)}{T}\right), \sigma\left(\frac{f_s(\boldsymbol{x}_n)}{T}\right)),$$

$$\mathcal{L}_{BS}(n, k) = J(\sigma\left(\frac{f_{\mathrm{t}}(\mathring{\boldsymbol{x}}_n^k)}{T}\right), \sigma\left(\frac{f_s(\mathring{\boldsymbol{x}}_n^k)}{T}\right)).$$

Experiments

Table 1: Comparison on CIFAR-10 dataset

Student	Original	Hinton (2015)	FITNET (2015) +Hinton	AT (2016) +Hinton	FSP (2017) +Hinton	Proposed	FSP (2017) +Proposed
ResNet 8	86.02%	86.66%	86.73%	86.86%	87.07%	87.32%	**87.52%**
ResNet 14	89.11%	89.75%	89.72%	89.84%	89.92%	**90.34%**	90.13%
ResNet 20	90.16%	90.77%	90.46%	90.81%	90.27%	**91.23%**	90.19%

Table 2: Comparison on ImageNet 32×32

	Original	Hinton (2015)	FITNET (2015) +Hinton	AT (2016) +Hinton	FSP (2017) +Hinton	Proposed	FSP (2017) +Proposed
Top1 acc	31.94%	32.43%	32.60%	32.61%	32.66%	32.69%	**32.72%**
Top5 acc	56.21%	56.99%	57.02%	57.14%	57.14%	57.17%	**57.27%**

Table 3: Comparison on TinyImageNet

	Original	Hinton (2015)	FITNET (2015) +Hinton	AT (2016) +Hinton	FSP (2017) +Hinton	Proposed	FSP (2017) +Proposed
Top1 acc	50.68%	52.35%	53.52%	52.74%	53.43%	52.99%	**53.86%**
Top5 acc	76.14%	77.67%	78.10%	77.82%	78.15%	78.38%	**78.49%**

Experiments

Table 4: Comparison of knowledge distillation using various types of adversarial samples.

	Baseline	Random noise	L2 minimize	FGSM (2015)	DeepFool (2016)	Proposed
Accuracy	86.66%	86.73%	86.94%	87.06%	86.95%	**87.32%**

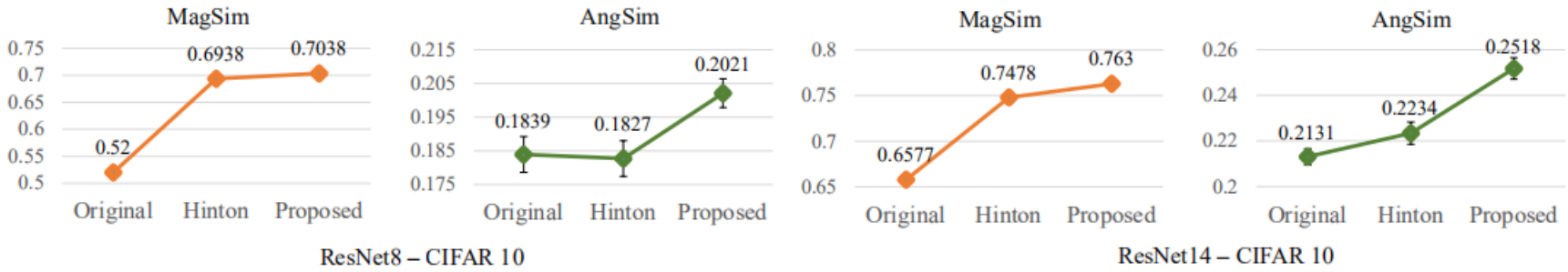


Figure 3: Evaluation of proposed method for decision boundary similarities (*MagSim*, *AngSim*).

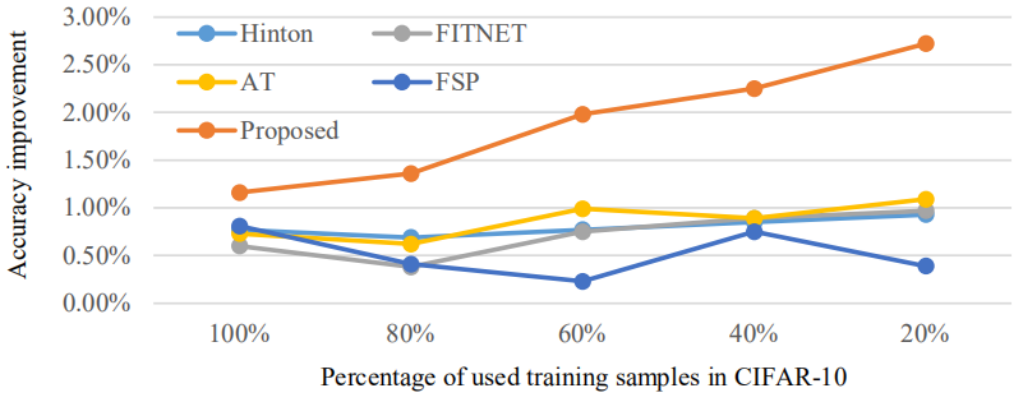


Figure 4: Generalization of the classifier. The smaller the number of training samples are, the larger improvement the proposed method shows.

The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons

Byeongho Heo,[1] **Minsik Lee,**[2] **Sangdoo Yun,**[3] **Jin Young Choi**[1]
{bhheo, jychoi}@snu.ac.kr, mleepaper@hanyang.ac.kr, sangdoo.yun@navercorp.com
[1]Department of ECE, ASRI, Seoul National University, Korea
[2]Division of EE, Hanyang University, Korea
[3]Clova AI Research, NAVER Corp, Korea

AAAI 2019

Method

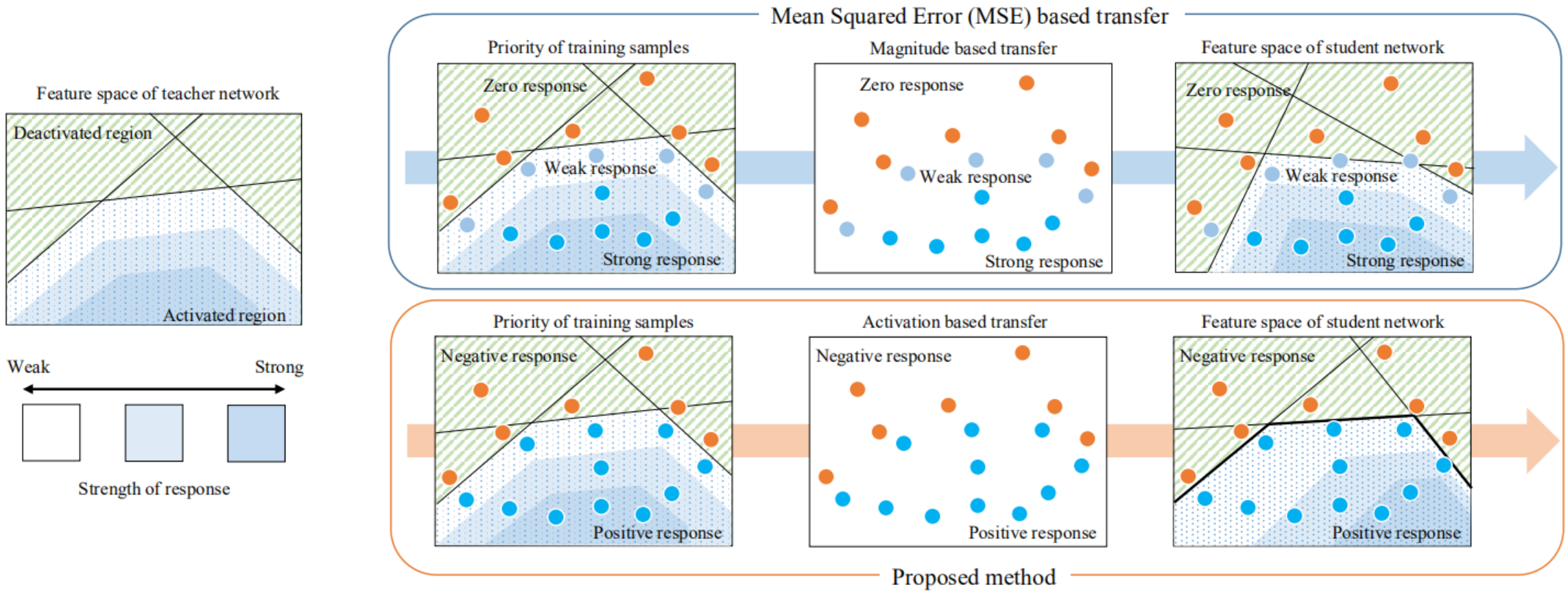


Figure 1: The concept of the proposed knowledge transfer method. The proposed method concentrates on the activation of neurons, not the magnitude of neuron responses. This concentration enables more precise transfer of the activation boundaries.

The second approach is to transfer the neuron responses for initializing the student network before the classification training. (network initialization method)

Method

- An existing method of neuron response transfer.

$$\mathcal{L}(\boldsymbol{I}) = \|\sigma(\mathcal{T}(\boldsymbol{I})) - \sigma(\mathcal{S}(\boldsymbol{I}))\|_2^2$$

$$\text{ReLU } \sigma(x) = max(0, x)$$

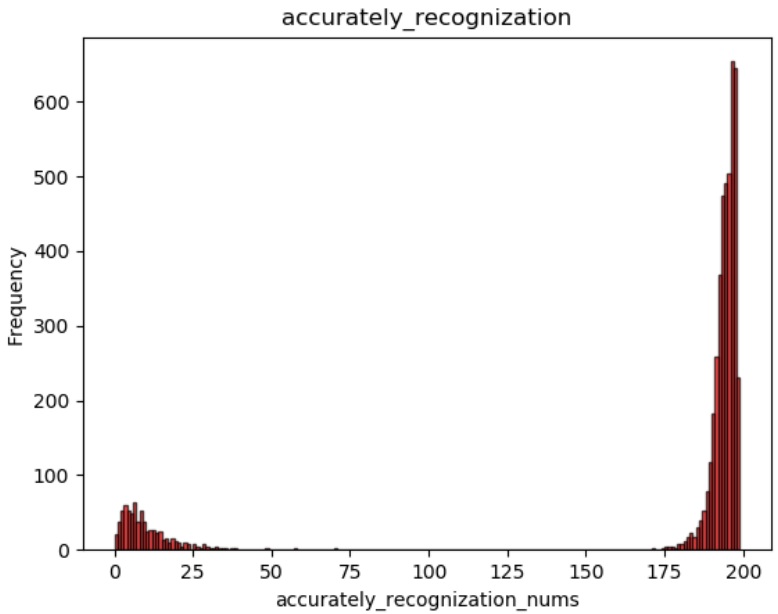
  - The part from the input image to a hidden layer is defined as function T, and a part of the student network is defined as function S.

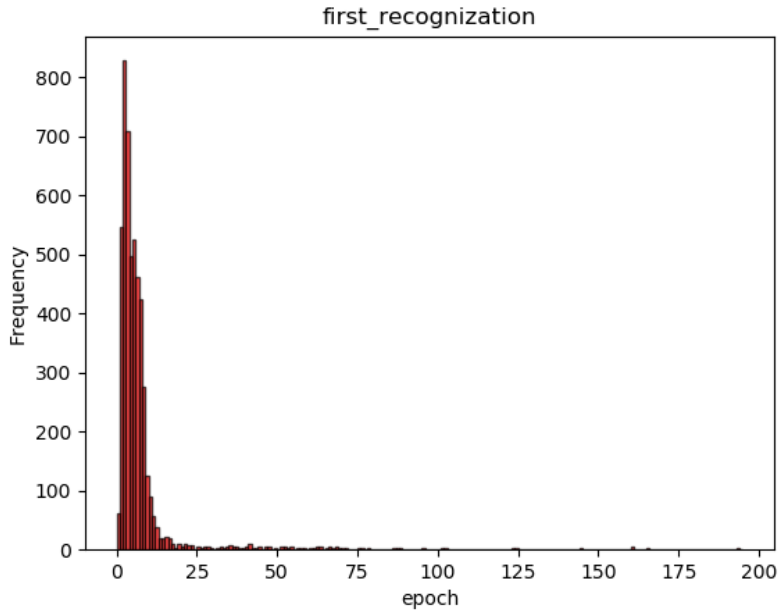
- The proposed method.

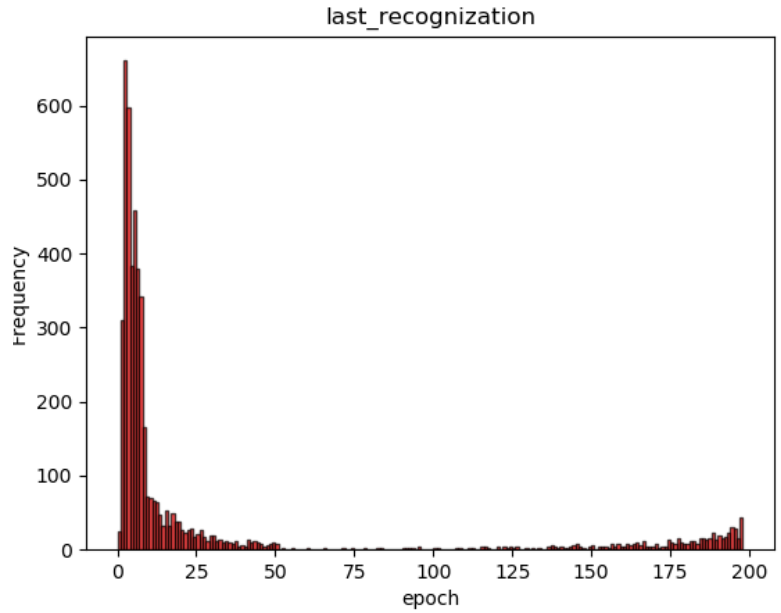
$$\mathcal{L}(\boldsymbol{I}) = \|\rho(\mathcal{T}(\boldsymbol{I})) - \rho(\mathcal{S}(\boldsymbol{I}))\|_1, \quad \rho(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise.} \end{cases}$$

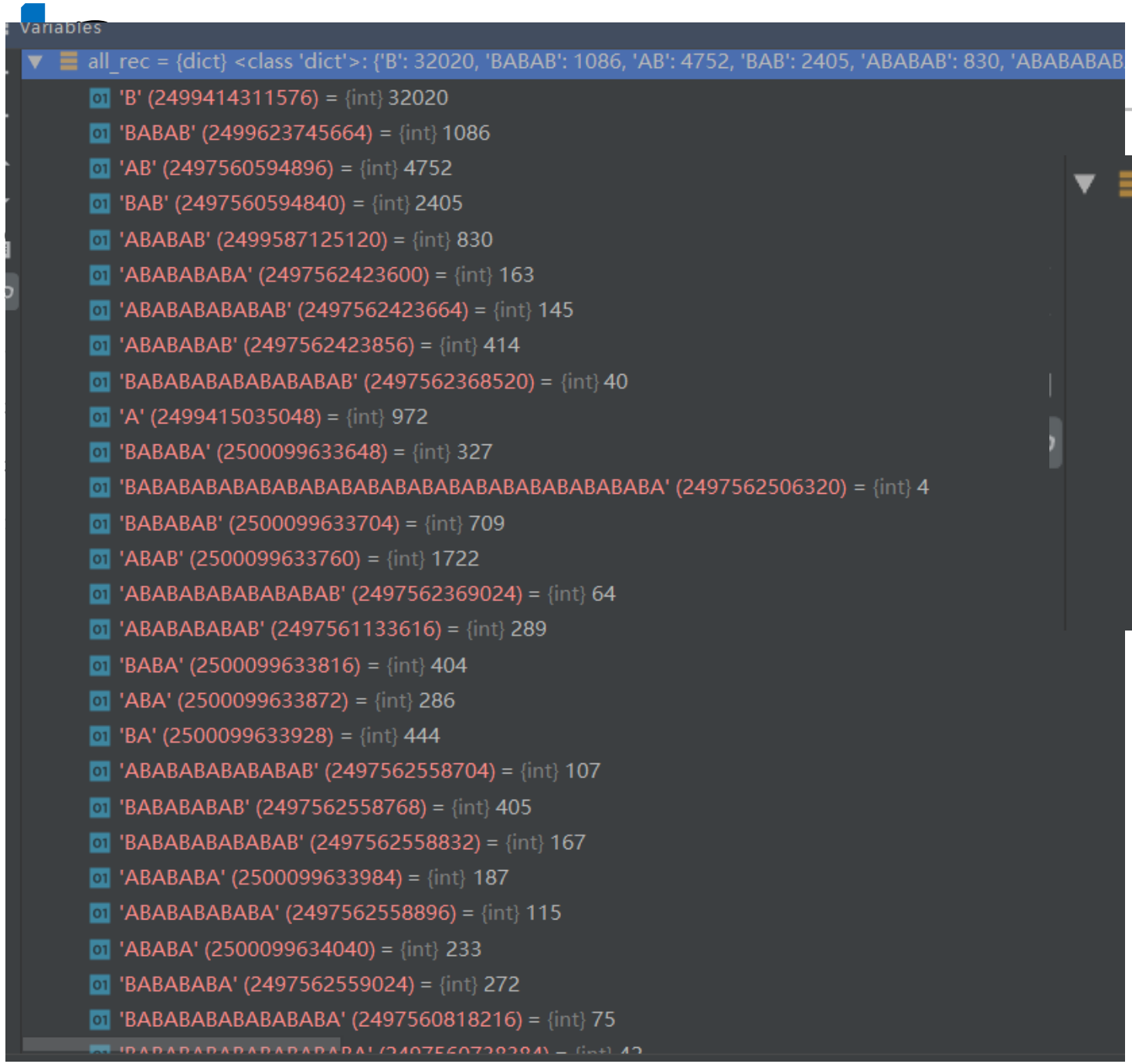
  - The activation boundaries play an important role in forming the decision boundaries for classification-friendly partitioning of the feature space in each hidden layer.
  - $\rho(x)$ expresses whether a neuron is activated or not.

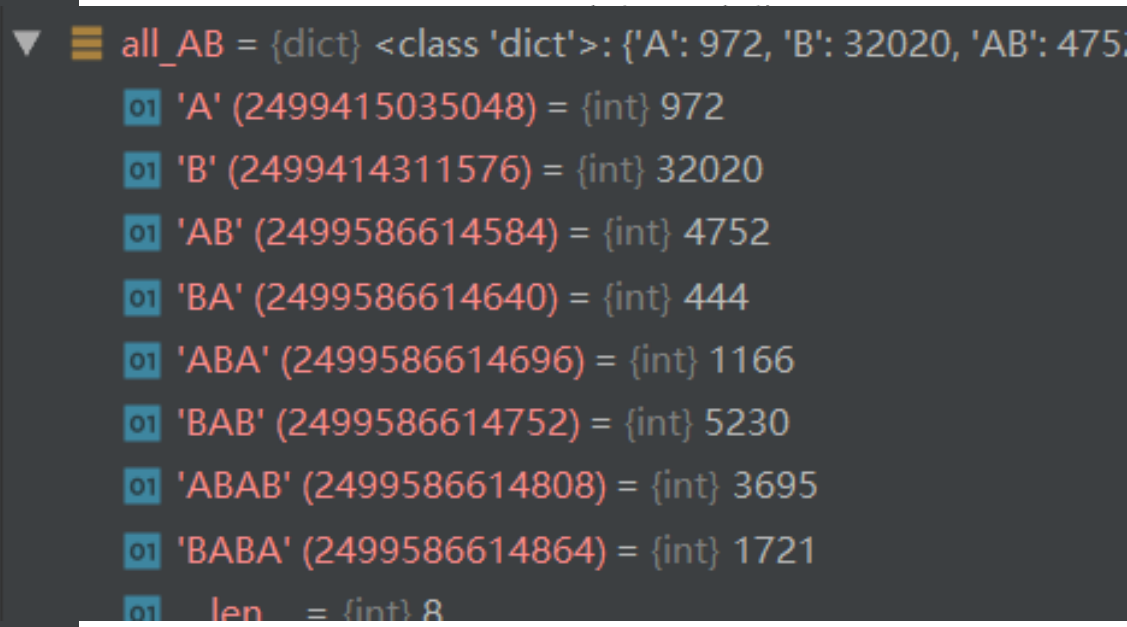
Experiments








Variables
all_rec = {dict} <class 'dict'>: {'B': 32020, 'BABAB': 1086, 'AB': 4752, 'BAB': 2405, 'ABABAB': 830, 'ABABABAB
'B' (2499414311576) = {int} 32020
'BABAB' (2499623745664) = {int} 1086
'AB' (2497560594896) = {int} 4752
'BAB' (2497560594840) = {int} 2405
'ABABAB' (2499587125120) = {int} 830
'ABABABABA' (2497562423600) = {int} 163
'ABABABABABAB' (2497562423664) = {int} 145
'ABABABAB' (2497562423856) = {int} 414
'BABABABABABABABAB' (2497562368520) = {int} 40
'A' (2499415035048) = {int} 972
'BABABA' (2500099633648) = {int} 327
'BABABABABABABABABABABABABABABABABABABABABABABA' (2497562506320) = {int} 4
'BABABAB' (2500099633704) = {int} 709
'ABAB' (2500099633760) = {int} 1722
'ABABABABABABABAB' (2497562369024) = {int} 64
'ABABABABAB' (2497561133616) = {int} 289
'BABA' (2500099633816) = {int} 404
'ABA' (2500099633872) = {int} 286
'BA' (2500099633928) = {int} 444
'ABABABABABABAB' (2497562558704) = {int} 107
'BABABABAB' (2497562558768) = {int} 405
'BABABABABABAB' (2497562558832) = {int} 167
'ABABABA' (2500099633984) = {int} 187
'ABABABABABA' (2497562558896) = {int} 115
'ABABA' (2500099634040) = {int} 233
'BABABABA' (2497562559024) = {int} 272
'BABABABABABABABA' (2497560818216) = {int} 75
Frequency


all_AB = {dict} <class 'dict'>: {'A': 972, 'B': 32020, 'AB': 475
'A' (2499415035048) = {int} 972
'B' (2499414311576) = {int} 32020
'AB' (2499586614584) = {int} 4752
'BA' (2499586614640) = {int} 444
'ABA' (2499586614696) = {int} 1166
'BAB' (2499586614752) = {int} 5230
'ABAB' (2499586614808) = {int} 3695
'BABA' (2499586614864) = {int} 1721
len = {int} 8

Thanks