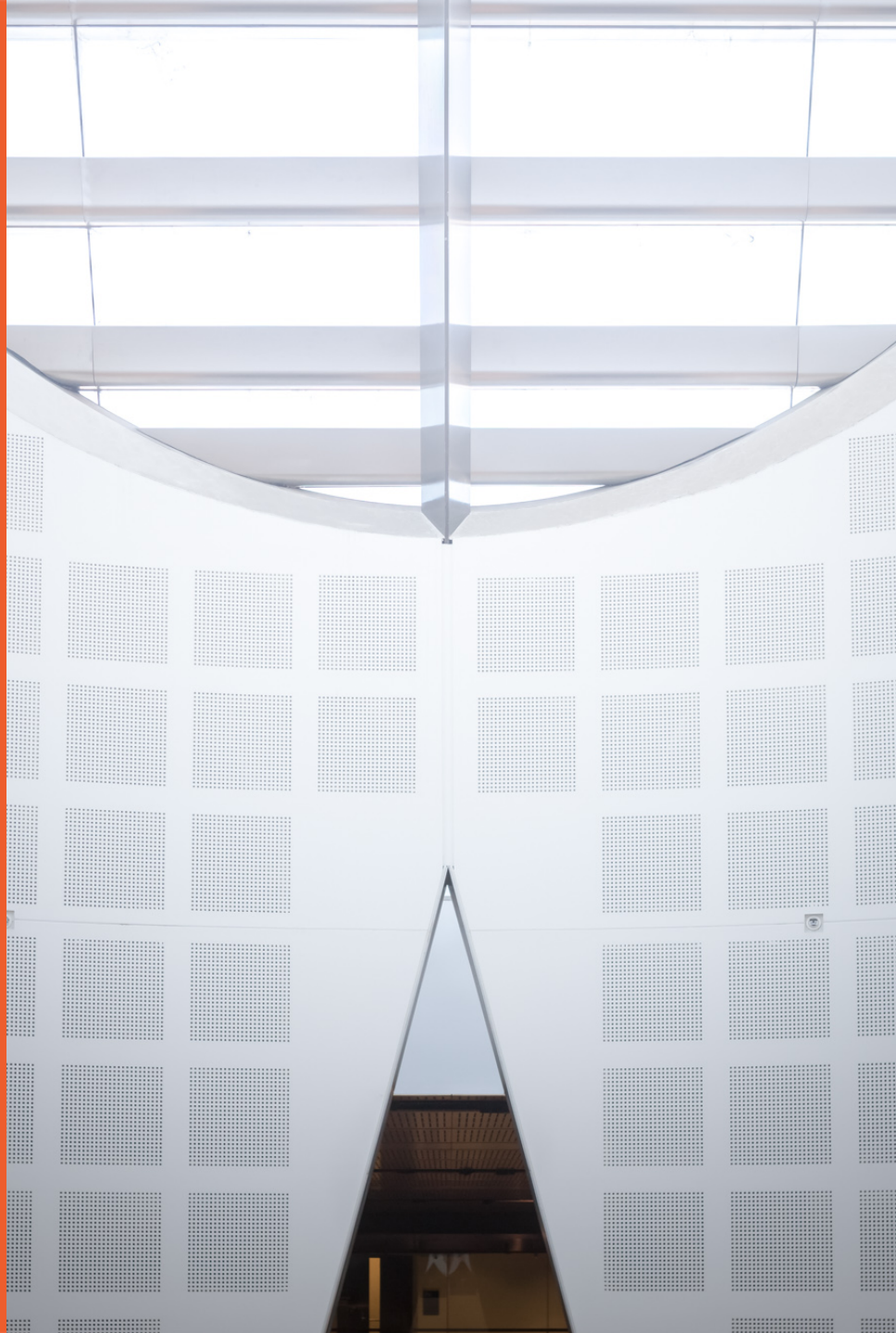
Learning Student Network with Few Data

Shumin Kong
School of Computer Science
University of Sydney

THE UNIVERSITY OF SYDNEY



Agenda

- Knowledge Distillation
- Teacher-student learning

Network Compression – Why?

- Nowadays, deep learning models are getting more and more difficult to train, take BigGAN as an example.
- It takes time.
  - Training BigGAN requires 15 days of training with 8 Nvidia Tesla v100.
- It needs a lot of data.
  - The training set of ImageNet is around 120 gigabytes.
- It costs money.
  - US$4406 to run 8 Nvidia Tesla v100 for 15 days on AWS.
- Is deep learning research becoming a pay2win game?
  - No!

Knowledge Distillation

- Idea: to distill the knowledge from a larger (teacher) network to a smaller (student) network.
- Teacher networks:
  - Large in terms of # of parameters.
  - Deep in terms of # of layers.
  - Usually take a long time to train.
- Student network:
  - Small in terms of # of parameters.
  - Shallow in terms of # of layers.

Knowledge Distillation – Founding work

- Idea: Train the student network such that its soften output is similar to that of the teacher network.
- Denote z_i as the logits, i.e. the un-activated outputs of the final layer, of a network, we train the student network s.t. its

$$q_i = \mathrm{softmax}(\frac{z_i}{T})$$

- is similar to the teacher network's, where T is a constant denoting the temperature of the distillation, which is usually set to one.

Hinton, G., Vinyals, O., & Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Knowledge Distillation - FitNet

- Idea: Train the student network such that the output one of its middle layer is similar to that of the teacher network.
- Denote h_S and h_T as the output of middle layer of students and teachers, respectively. We pre-train the student network such that

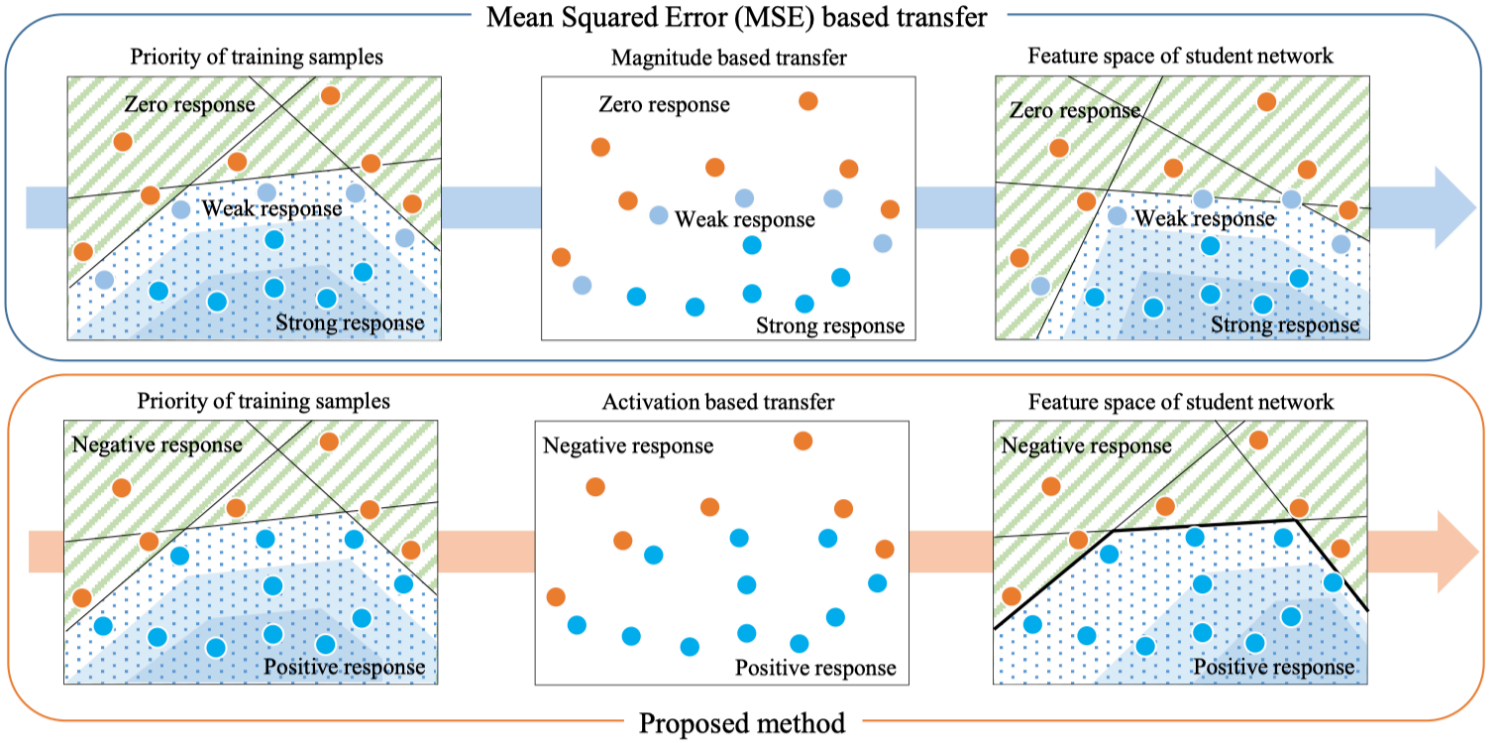
$$h'_S = f(W^*, h_S)$$

- is similar to h_T, where $f(W^*, \cdot)$ is an additional layer that can make sure the shape of h'_S and h_T is consistent.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., & Bengio, Y. (2014). Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550.

Knowledge Distillation – Activation Boundary

- Idea: Train the student network such that the activation boundary of its middle layer is similar to that of the teacher network.



Heo, B., Lee, M., Yun, S., & Choi, J. Y. (2019, July). Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 3779-3787).

Problems

- Previous work still assume that the entire dataset used to train the teacher network is also available for the student network.

- This assumption may not hold for some real-world scenarios.
  - many applications do not open their large training dataset completely but only supply a fraction for verification purposes;
  - When initializing voice-assistant apps, user only need to speak a few words instead of the whole dictionary for the app to recognize the voice of user.

- In other words, the generalization ability of the student network is overlookded.

Our work - Formulation

- To boost the generalization ability of the student network, we propose to explore the ground-truth data-generating distribution $\mathbb{P}$.
- To approach that, we suggest to construct a Wasserstein ball with radius of ϵ that surrounds the distribution $\mathbb{P}_N$,

$$\mathcal{B}_\epsilon(\mathbb{P}_N) \coloneqq \{\mathbb{Q} \in \mathcal{M}(\mathcal{X}) : d_W(\mathbb{P}_N, \mathbb{Q}) \leq \epsilon\}$$

- where:
  - $\mathcal{M}(\mathcal{X})$ is the set of all possible distributions $\mathbb{Q}$ over $\mathcal{X}$,
  - $d_W(\cdot,\cdot)$ is the Wasserstein distance between two probabilistic distributions.
- Since the distribution $\mathbb{P}$ lies in this ball, we suggest to optimize the risk of all possible ground-truth distributions within the $\mathcal{B}_\epsilon(\mathbb{P}_N)$.

Our work - Formulation

$$\mathcal{B}_\epsilon(\mathbb{P}_N) \coloneqq \{\mathbb{Q} \in \mathcal{M}(\mathcal{X}) : d_W(\mathbb{P}_N, \mathbb{Q}) \leq \epsilon\}$$

- Since the distribution $\mathbb{P}$ lies in this ball, we suggest to optimize the risk of all possible ground-truth distributions within the $\mathcal{B}_\epsilon(\mathbb{P}_N)$.
- This goal equals to improving the worst-case risk of all distributions in $\mathcal{B}_\epsilon(\mathbb{P}_N)$, which is equivalent to minimizing

$$\sup_{\mathbb{Q} \in \mathcal{B}_\epsilon(\mathbb{P}_N)} \mathbb{E}[\![\ell(\boldsymbol{x}; \mathcal{N}_\mathcal{S}, \mathcal{N}_\mathcal{T})]\!]. \quad (1)$$

Theoretical Analysis

$$\sup_{\mathbb{Q}\in\mathcal{B}_\epsilon(\mathbb{P}_N)} \mathbb{E}[\![\ell(\boldsymbol{x};\mathcal{N}_S,\mathcal{N}_T)]\!]. \ (1)$$

- It can be shown that an upper bound of Eq.(1) can be bounded by

$$\sup_{\mathbb{Q}\in\mathcal{B}_\epsilon(\mathbb{P}_N)} \mathbb{E}^{\mathbb{Q}}[\![\ell(\boldsymbol{x})]\!] \leq \frac{1}{N}\sum_{i=1}^{N}\ell(\boldsymbol{x}_i) + \epsilon\sup_{\boldsymbol{x}\in\mathcal{X}}\left|\left|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{x})\right|\right|$$

- if the loss function ℓ:
  - is proper, convex, lower semicontinous, and
  - is $\bar{\mathrm{L}}$ - Lipchitz continuous on $\mathcal{X}$.

Theoretical Analysis

$$\sup_{\mathbb{Q}\in\mathcal{B}_\epsilon(\mathbb{P}_N)} \mathbb{E}^{\mathbb{Q}}[\![\ell(\boldsymbol{x})]\!] \leq \frac{1}{N}\sum_{i=1}^{N}\ell(\boldsymbol{x}_i) + \epsilon \sup_{\boldsymbol{x}\in\mathcal{X}} \left|\left|\nabla_{\boldsymbol{x}}\ell(\boldsymbol{x})\right|\right|$$

- Since calculating supremum can be computationally expensive, this upper bound is approximated by training examples and we adopt a proxy of this value. Finally, our proposed loss function is

$$\mathcal{L}_W(\mathcal{N}_S) = \frac{1}{N}\sum_{i=1}^{N} l\left(\boldsymbol{o}_i^S, \boldsymbol{o}_i^T\right) + \epsilon\frac{1}{N}\sum_{i=1}^{N} ||\nabla_{\boldsymbol{x}_i} l\left(\boldsymbol{o}_i^S, \boldsymbol{o}_i^T\right)|| \quad (2)$$

P.S. inclusion of the gradient term in Eq. (2) can be explained as training the network with data augmentation by adding noise to the data.

Experiments

Table 2: Classification accuracy with respect to various methods and different number of training examples on CIFAR-10 dataset.

M	Ours	AB	FN	KD
Full	**84.66%**	82.65%	77.84%	79.72%
1000	**82.01%**	76.19%	63.84%	77.63%
500	**73.08%**	66.69%	54.51%	50.63%
100	**47.24%**	44.7%	29.68%	31.83%
50	**41.16%**	38.23%	31.65%	28.32%

Table 3: Classification accuracy with respect to various methods and different number of training examples on Fashion-MNIST dataset.

M	Ours	AB	FN	KD
Full	**94.72%**	92.1%	92.23%	92.1%
1000	**91.95%**	91.67%	91.53%	90.44%
500	**90.46%**	90.25%	82.6%	89.45%
100	**85.18%**	84.09%	81.92%	83.77%
50	**81.73%**	80.82%	81.40%	78.5%

Thank you.