Graph-based Knowledge Distillation by Multi-head Attention Network

Seunghyun Lee*, Byung Cheol Song

lsh910703@gmail.com, bcsong@inha.ac.kr

Department of Electronic Engineering, Inha University, Republic of Korea

BMVC CARDIFF 2019

Introduction

Knowledge Distillation

- **Achieve optimal performance** from a small student network (SN) by distilling the knowledge of a large teacher network (TN) and transferring the distilled knowledge to the small SN.
- **Distilled knowledge can be applied for other purposes** such as semi-supervised learning and pruning.

Contribution Points

- **Analyze previous knowledge distillation methods** and point out their fundamental issue.
- Propose a **novel algorithm to extract embedding procedure knowledge** via attention networks.

Problem Statement

Limitations of Previous Approaches

- Most of the previous methods focus on **How** to distill knowledge, not **What** to distill.
- **All type of knowledge is not still acceptable** as a neural network's knowledge.
  - Neural response & Multi-connection : Too naïve.
  - Shared-representation : Cannot find inter-data relation.
  - Inter-data relation knowledge : Only focus on the last embedded space.

Problem Definition

- **Find the knowledge which coincides with neural network's purpose.**
- Embed high-dimensional data into low-dimension for easier analysis.
- A good teacher teaches not only answer but how to solve.
  → **Embedding procedure** is the real knowledge of the network.

Method

Training Multi-head Attention to Distill Knowledge

- **Estimator** which estimates set of back-end singular vector ($\mathbf{V}^B$) using set of front-end singular vector ($\mathbf{V}^F$).

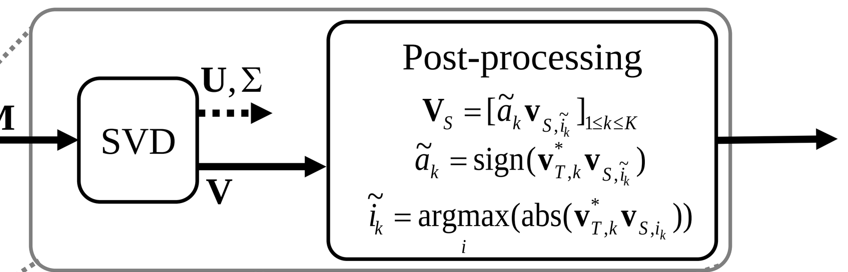
$$\overline{\mathbf{V}}^B = f_2\left(\mathbf{G}, f_1\left(\mathbf{V}^F\right)\right) \qquad L_{MHAN} = \sum_{m=1}^{M} \frac{1}{N} \mathbf{V}_m^B \overline{\mathbf{V}}_m^B$$

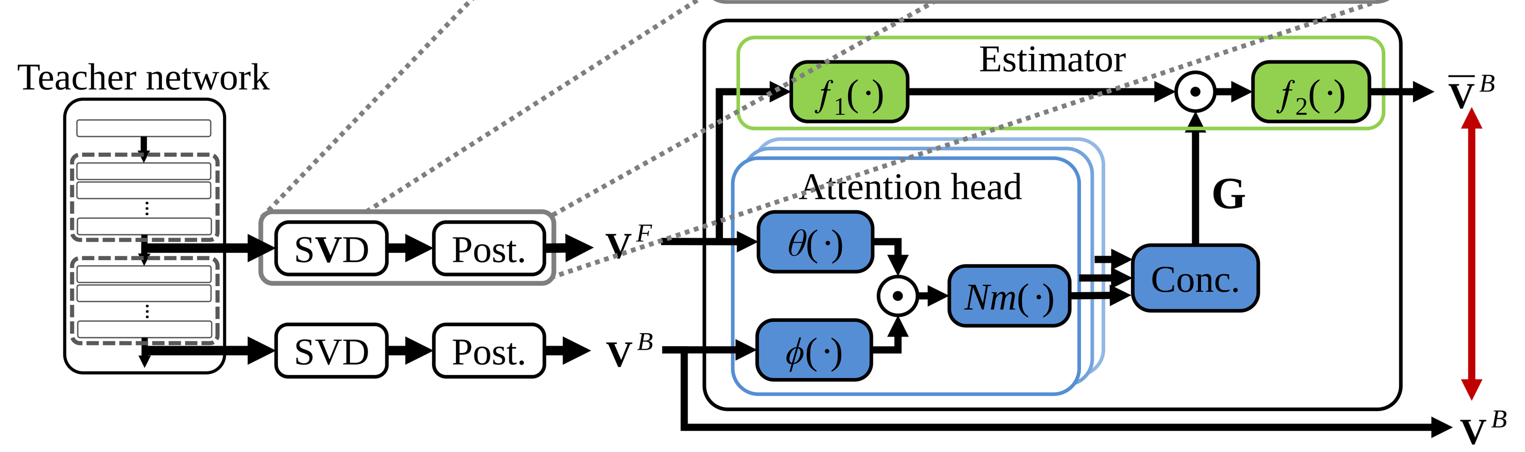
- **Attention head** which enhances the estimator's feature vector to make it easy to estimate $\mathbf{V}^B$.

$$\mathbf{G} = \left[Nm\left(\mathbf{S}_a\right)\right]_{1\le a\le A} \quad Nm(\mathbf{S}) = \left[\frac{\exp(\mathbf{S}_{i,j})}{\sum_k \exp(\mathbf{S}_{i,k})}\right]_{1\le i,j\le N} \quad \mathbf{S} = \left[\theta(\mathbf{v}_i^B)\cdot\phi(\mathbf{v}_i^F)\right]_{1\le i,j\le N}$$

Post-processing[1]

- Remove bad properties of singular vector

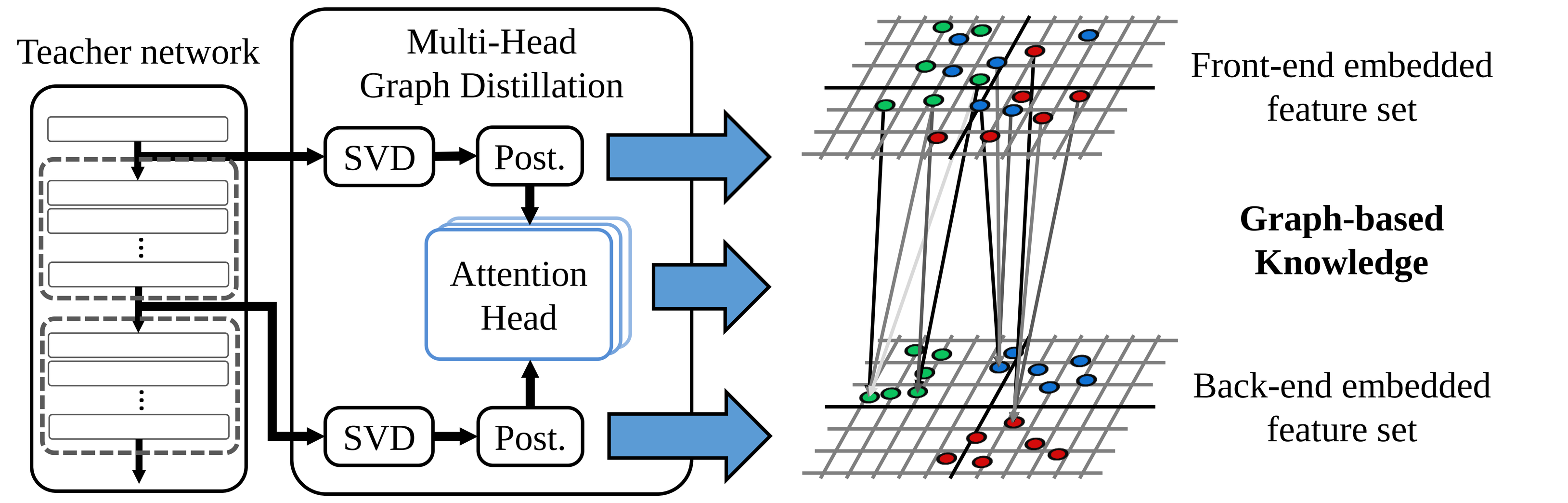




Attention map as Graph-based Knowledge

- Attention heads extract the relation between $\mathbf{V}^F$ and $\mathbf{V}^B$ to enhance to estimate $\mathbf{V}^B$ easily.
  → Give more attention to the $\mathbf{V}^F$ embedded into similar points.
  → **Embedding procedure** is expressed by **graph-form.**

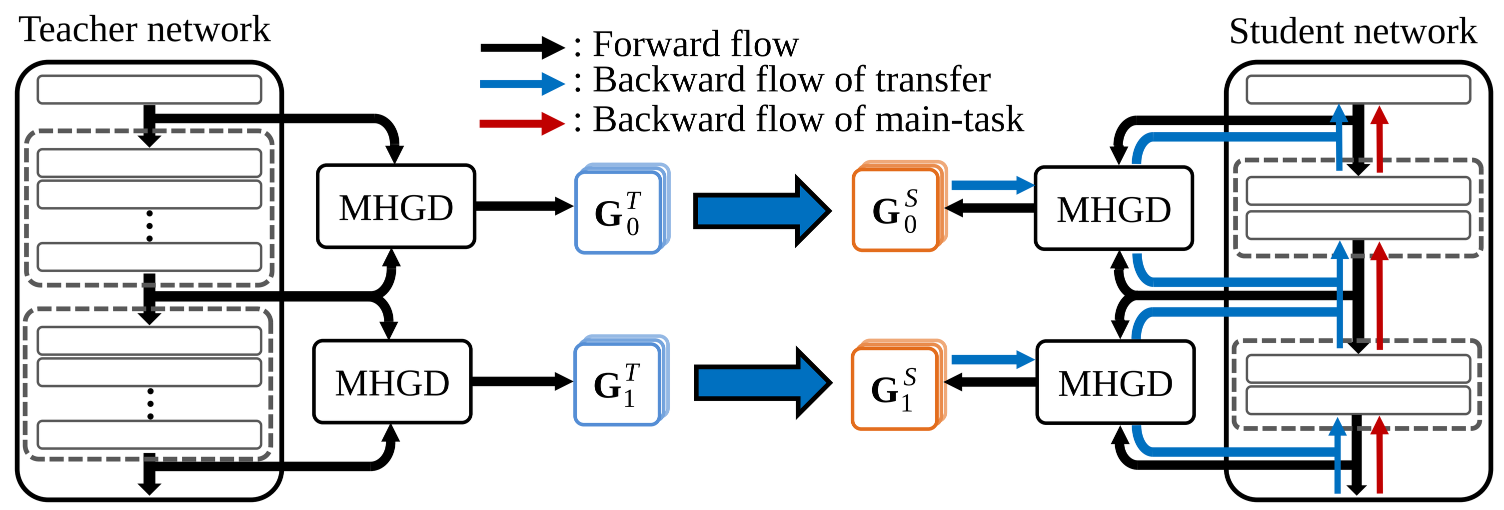
$$L_{transfer} = \sum_{m,i,j,a} \mathbf{G}^S_{m,i,j,s}\left(\log(\mathbf{G}^S_{m,i,j,s}) - \log(\mathbf{G}^T_{m,i,j,s})\right)$$



Transfer of Graph-based Knowledge

- Adaptive constraint multi-task learning via gradient clipping [1].
- Transfer the TN's knowledge as much as possible without over-regularization.

$$\left(\frac{\partial\Theta}{\partial L_{KD}}\right)_{clipped} = \frac{1}{1+\exp(-\tau)}\frac{\partial\Theta}{\partial L_{transfer}}, \text{ where } \tau = \max\left(1, \left\|\frac{\partial\Theta}{\partial L_{main}}\right\|_2 \Big/ \left\|\frac{\partial\Theta}{\partial L_{transfer}}\right\|_2\right)$$



Experimental results

Small Network Enhancement

- Network architectures

Network		FLOPS (M)	Params (M)
VGG	TN (VGG16)	143.7	11.83
	SN (VGG7)	17.6 (**12.2%**)	17.6 (**18.5%**)
WResNet	TN (WResNet22-4)	374.2	0.417
	SN (WResNet10-4)	93.2 (**24.9%**)	0.1404 (**33.7%**)

Transfer Knowledge to Different Architectures

- Even though transferring knowledge to SN that is different from TN's architecture, the proposed method outperforms the others.

Effect of Attention Head

- More heads tend to produce much knowledge.
- But too many attention heads may cause over-constraint.

Performance comparison of several KD methods for **CIFAR100**.

Method	Teacher	Student	Soft-logits	FSP	AB	KD-SVD	KD-SVDF	MHGD
VGG	67.99	59.97	60.95	61.87	64.56	64.25	64.38	**67.02**
WResNet	77.22	71.62	71.88	71.57	72.23	71.83	71.82	**72.79**

Performance comparison of several KD methods in **TinyImageNet**.

Method	Teacher	Student	Soft-logits	FSP	AB	KD-SVD	KD-SVDF	MHGD
VGG	56.30	52.40	53.78	54.85	54.99	55.33	55.35	**56.35**
WResNet	61.31	55.91	56.00	56.04	56.53	55.72	55.95	**56.90**

Performance comparison of various KD methods with **WResNet as the TN**.

Method	Student	Soft-logits	FSP	AB	KD-SVD	MHGD
VGG	69.76	70.51	69.44	71.24	70.31	**71.52**
MobileNet	66.18	67.35	60.35	67.84	67.03	**68.32**
ResNet	71.57	71.81	70.40	71.55	71.55	**72.74**

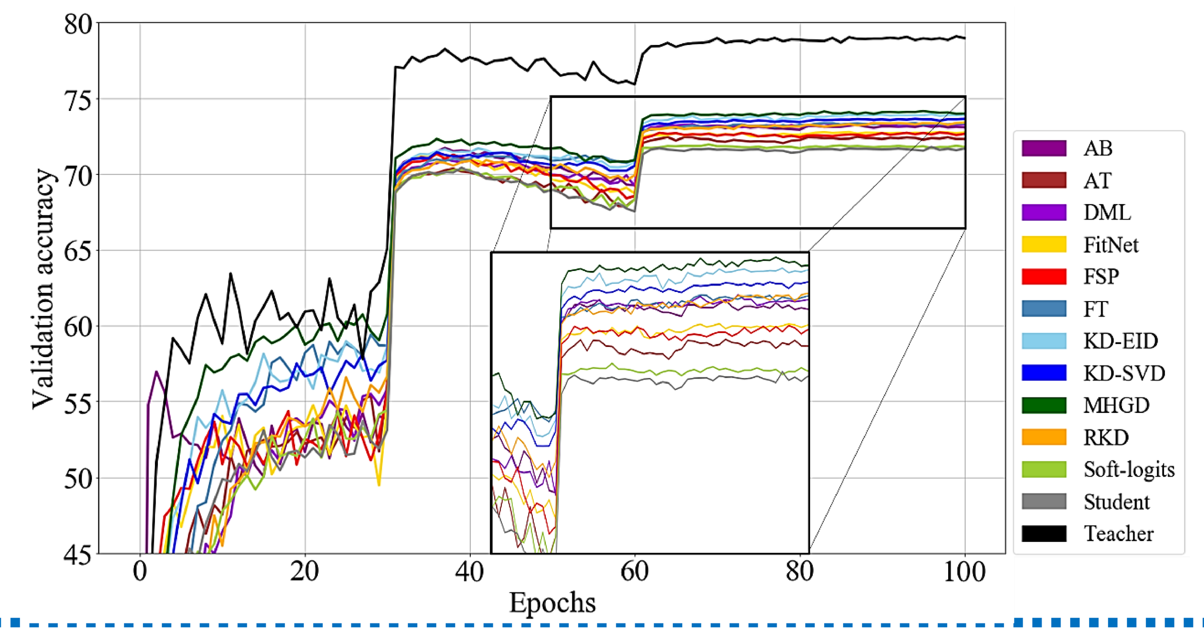
The performance change according to the **number of attention heads**.

num_head	0 (Student)	1	2	4	8	16
Accuray	59.97	65.71	66.41	67.01	**67.02**	66.70

Comparison with SOTA

- **The codes for proposed and previous methods are available at above QR code or https://github.com/sseung0703/KD_methods_with_TF**

Method	Student	Teacher	Soft-logits[2]	FitNet[5]	AT[6]	FSP[3]
Accuracy	71.76	78.96	71.79	72.74	72.31	72.65
Method	**DML[7]**	**KD-SVD[1]**	**FT[8]**	**AB[4]**	**RKD[9]**	**MHGD**
Accuracy	73.27	73.68	73.35	73.08	73.40	**73.98**



[1] Lee et al. ECCV2018
[2] Hinton et al. NIPS 2014
[3] Yim et al. CVPR2017
[4] Heo et al. AAAI2019
[5] Romeo et al. ICLR2015
[6] Sergey et al. ICLR2017
[7] Zhang et al. CVPR2018
[8] Kim et al. NeurIPS2018
[9] Park et al. CVPR2019