Interpretable Embedding Procedure Knowledge Transfer via Stacked Principal Component Analysis and Graph Neural Network

Seunghyun Lee, Byung Cheol Song
Department of Electronic Engineering, Inha University, Incheon 22212, South Korea
lsh910703@gmail.com, bcsong@inha.ac.kr

Abstract

Knowledge distillation (KD) is one of the most useful techniques for light-weight neural networks. Although neural networks have a clear purpose of embedding datasets into the low-dimensional space, the existing knowledge was quite far from this purpose and provided only limited information. We argue that good knowledge should be able to interpret the embedding procedure. This paper proposes a method of generating interpretable embedding procedure (IEP) knowledge based on principal component analysis, and distilling it based on a message passing neural network. Experimental results show that the student network trained by the proposed KD method improves 2.28% in the CIFAR100 dataset, which is higher performance than the state-of-the-art (SOTA) method. We also demonstrate that the embedding procedure knowledge is interpretable via visualization of the proposed KD process. The implemented code is available at https://github.com/sseung0703/IEPKT.

Introduction

Convolutional neural networks (CNNs) have been adopted by a variety of areas because of their outstanding performance. However, CNNs require a huge amount of computation and memory cost, which makes it hard to mount on embedded and mobile systems. Knowledge distillation (KD) is one of the solutions to build light-weighted CNNs (Hinton, Vinyals, and Dean 2015). The main function of KD is to create and deliver a certain knowledge so that a student network behaves similarly to a teacher network. Since KD can be applied to various machine learning areas such as semi-supervised learning and zero-shot learning, KD has been received a lot of attention recently. Conventional KD algorithms defined information from several locations of CNN, e.g., intermediate feature maps (Romero et al. 2014; Zagoruyko and Komodakis 2016a) and embedded feature vectors at the output end of CNN (Hinton, Vinyals, and Dean 2015; Park et al. 2019; Kim, Park, and Kwak 2018), as the knowledge of CNN.

Note that CNN's ultimate goal is to map high dimensional data such as images and audio to low dimensional space for easy analysis. However, the knowledge proposed so far has been far from the purpose of CNNs. In order to improve the student network's embedding performance, it is important to accurately convey the information about the embedding process of CNN, which analyzes a dataset in order from low-level features to high-level features. Therefore, we insist that the knowledge of CNN should represent the embedding procedure and be able to interpret human insight during the distillation process. In order to define and distill such interpretable embedding procedure (IEP) knowledge, we have gained a few insights from the following previous works.

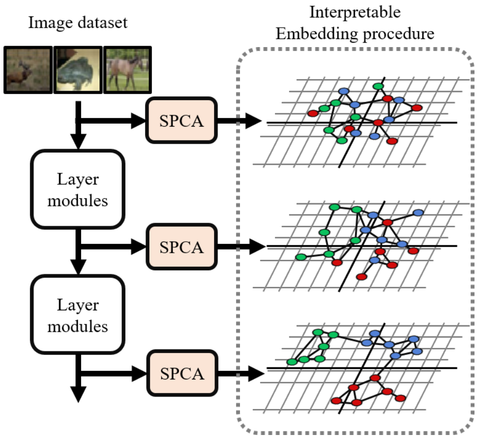

Figure 1: The conceptual illustration of the neural network's embedding procedure knowledge with the proposed stacked principal component analysis (SPCA).

Principal component analysis (PCA) has been one of the effective tools for visualizing and analyzing embedded feature distribution (Härkönen et al. 2020). On the other hand, since graph-structure can effectively represent inter-data relations, many graph neural networks (GNNs) have been developed recently. For example, GNN is attracting attention as a reliable solution to represent embedding space (Meng et al. 2018).

Based on insights from the previous works, this paper presents a new KD method for distilling IEP knowledge. First, to analyze the embedding procedure of the teacher network, a stacked PCA (SPCA), which performs PCA twice, is proposed. SPCA allows the feature map dimension to be shallow, enabling analysis and visualization of embedding procedures at relatively low cost. Here, the graph-structure is employed. Figure 1 illustrates this concept. Next, in order to distill IEP knowledge with minimal information loss, a new distillation method using a message passing neural network (MPNN) is proposed. The MPNN distills IEP knowledge by estimating the embedding procedure of each stage from the previous stage's graph.

Our contribution points are as follows: First, student networks trained by the proposed IEP knowledge provide SOTA performance. Second, IEP knowledge is interpretable via visualization, which represents the embedding procedure of CNN and coincides with human insight.

Related Works

Knowledge Distillation

Conventional KD methods defined various knowledge. Some of them defined the neural response of the last or several interim feature maps of CNN as knowledge (Ahn et al. 2019; Hinton, Vinyals, and Dean 2015; Romero et al. 2014; Zagoruyko and Komodakis 2016a). Also, there have been attempts to distill embedding knowledge so as to overcome the problem of lack of knowledge in existing KD techniques (Ge et al. 2019; Lee and Song 2019; Park et al. 2019). Such approaches were mainly intended to find inter-data relations in the embedding space of the last stage of CNN. Only some of the approaches distilled embedding procedure information directly (Lee and Song 2019). However, the knowledge defined by the above-mentioned methods was not interpretable, which means that the previous methods could lose significant information during the distillation process. To distill the teacher network's knowledge with no or minimal information loss, we propose a new method for distilling IEP knowledge.

PCA in Deep Learning

As derivative functions of singular value decomposition (SVD) and eigendecomposition (EID) have been defined recently (Ionescu, Vantzos, and Sminchisescu 2015), several studies to fuse SVD and EID with deep neural networks have been published (Lee, Kim, and Song 2018; Valmadre et al. 2017). For example, SVD was used to compress feature maps (Valmadre et al. 2017), and principal components themselves were employed as compressed feature vectors (Lee and Song 2019; Lee, Kim, and Song 2018). As one of the ways to reduce the PCA's memory cost, incremental PCA (IPCA) incrementally estimated dataset's principal components and predicted mean vectors on a mini-batch basis (Ross et al. 2008). Recently, applying PCA to deep learning have been proposed (Hao and Zhang 2016). From the previous studies, we got an insight that embedding procedure knowledge can be obtained from the teacher network through IPCA.

Graph Neural Network

GNN has been studied in various fields as a tool for obtaining inter-data relations. In particular, MPNN, one of the GNNs, has emerged as a core technology in areas where edge information is important, such as physics (Cranmer et al. 2019; Henrion et al. 2017) and chemistry tasks (Gilmer et al. 2017; Nguyen, Maeda, and Oono 2017). Recently, Meng et al. proposed a technique to apply MPNN to embedding task (Meng et al. 2018). The method was able to interpret and estimate complex relationships between data by defining inter-data relations as edges. From the previous works, we gained another insight that MPNNs can capture relational information better than the attention networks that become recently popular more and more.

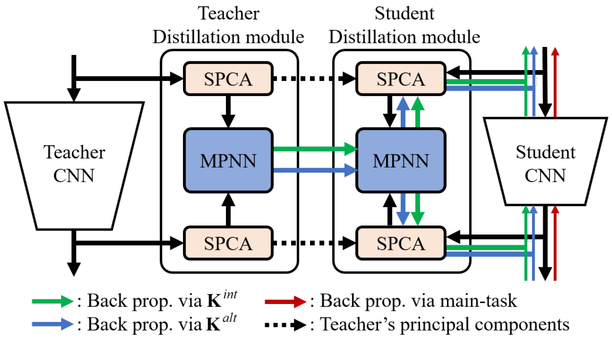
Teacher Distillation module
Student Distillation module
Teacher CNN
Student CNN
SPCA
SPCA
SPCA
SPCA
MPNN
MPNN
: Back prop. via $\mathbf{K}^{int}$
: Back prop. via $\mathbf{K}^{alt}$
: Back prop. via main-task
: Teacher's principal components

Figure 2: Conceptual diagram of the proposed knowledge distillation algorithm. $\mathbf{K}^{int}$ and $\mathbf{K}^{alt}$ mean knowledge of interim embedding stages and alteration of them, respectively.

Methods

This section defines knowledge of embedding procedure, i.e., the goal of neural networks, and suggests how to distill the knowledge in an interpretable form. The block diagram of the proposed method is shown in Fig. 2. First, compress the set of feature maps of the teacher network with SPCA, and calculate the embedding procedure, and define it as knowledge. MPNN distills this knowledge in a form that can be transferred to the student network. Next, the student network's IEP knowledge is distilled through a distillation module trained with the teacher network in advance. Finally, the student network's target task and teacher's knowledge are trained via multi-task learning. In addition, this paper presents a method for integrating the proposed method with multi-head graph distillation (MHGD), one of the latest KD techniques, to accomplish complete neural network knowledge.

Producing IEP Knowledge via SPCA

In order to produce embedding procedure knowledge of CNN, we need to analyze the inter-data relation of feature maps sensed in the middle of CNN, which corresponds to the intermediate stage of embedding. However, finding relations between feature maps usually requires high computation cost. Therefore, SPCA is applied to effectively reduce the dimension of feature maps without crucial information loss. SPCA also contributes to visualizing a set of feature maps for better understanding. The conceptual diagram of the SPCA is shown in Fig. 3.

Since feature maps generally have high spatial correlation, some feature vectors can have similar information to

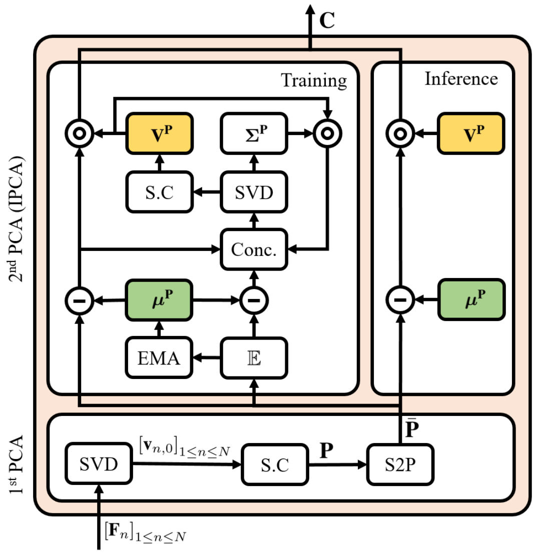

Figure 3: The block diagram of the stacked PCA (SPCA). Here, S.C is sign correction, S2P is sphere to plane mapping, EMA is an exponential moving average, and conc. means concatenation.

each other. Thus, a feature map can be approximated with several principal components (PCs). In detail, a feature map of $H \times W \times D$ is converted into a matrix $\mathbf{F}$ of $HW \times D$ by PCA. Here, H, W, D indicate the width, height, and depth of a feature map, respectively. We only adopt the first PC to compress the feature map as much as possible. Next, the sign flipping, which makes the largest absolute value positive, is used to remove the sign ambiguity of singular vectors (S.C in Fig. 3). Finally, assuming a mini-batch data, a set of PCs $\mathbf{P}$ is represented as follows.

$$\mathbf{P} = [\mathbf{p}_n]_{1\leq n\leq N} = [s_n \mathbf{v}_{n,0}]_{1\leq n\leq N}\,,\ \mathbf{P} \in \mathbb{R}^{N\times D} \tag{1}$$

$$s_n = \text{sign}\left(\max\left(\mathbf{v}_{n,0}\right) + \min\left(\mathbf{v}_{n,0}\right)\right) \tag{2}$$

$$\mathbf{V}_n = [\mathbf{v}_{n,k}]_{1\leq k\leq \min(HW,D)}\,,\ \mathbf{F}_n = \mathbf{U}_n \mathbf{\Sigma}_n \mathbf{V}_n^* \tag{3}$$

where * is the Hermitian function and N is the batch size.

Since $\mathbf{p}_n$ is HW times smaller than $\mathbf{F}_n$, it is possible to obtain the relation with very low cost. But the dimension of $\mathbf{P}$ is still too high to be interpreted by humans. So we apply PCA once more to find the PCs of the embedding space (the 2nd PCA in Fig. 3). Additional dimension reduction by the second PCA not only enables visualization but also eliminates unnecessary information. However, since $\mathbf{p}_n$ exists on the hypersphere as a unit vector, it is difficult to obtain a linear least squares solution. Therefore, as shown in Eq. (4), we apply stereographic projection (Apostol 1964) for mapping $\mathbf{P}$ to a plane space.

$$\bar{\mathbf{P}} = [\bar{\mathbf{p}}_n]_{1\leq n\leq N} = \left[\frac{\mathbf{p}_n + \mathbf{o}}{\cos\left(\cos^{-1}\left(\mathbf{p}_n^* \cdot \mathbf{o}\right)/2\right)^2} - 2\mathbf{o}\right]_{1\leq n\leq N}, \tag{4}$$

where $\mathbf{o}$ is the center vector of the space where PCs exist, that is $\left[1/\sqrt{D}, ..., 1/\sqrt{D}\right]$.

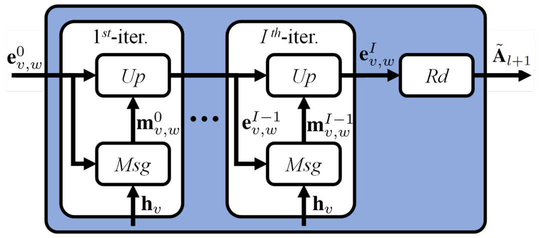

Figure 4: The block diagram of the proposed MPNN to distill interpretable embedding procedure knowledge.

Now, since $\bar{\mathbf{P}}$ is on the plane space, PCA can be applied. However, only mini-batch data is available for learning, and if the batch size is smaller than the feature dimension, full-rank PCs cannot be obtained. Also, it is not reasonable to assume that the PCs of the mini-batch data match those of the dataset. To solve these problems, we employ IPCA (Ross et al. 2008) which produces approximated PCs $\mathbf{V}^{\bar{\mathbf{P}}}$ and mean vector $\boldsymbol{\mu}^{\bar{\mathbf{P}}}$ by iteratively updating them. The IPCA in Fig. 3 is identical to the conventional IPCA, except that the mean vector calculation is replaced with an exponential moving average (EMA). The detailed description is given in the supplementary material. Since only the top half of the total PCs are used, the dimension of a set of compressed PCs $\mathbf{C}$ is reduced to half than before, as in Eq. (5) and Eq. (6).

$$\mathbf{C} = [\mathbf{c}_n]_{1\leq n\leq N} = \left[\left(\bar{\mathbf{p}}_n - \boldsymbol{\mu}^{\bar{\mathbf{P}}}\right) \cdot \mathbf{V}^{\bar{\mathbf{P}}}\right]_{1\leq n\leq N} \mathbf{\Sigma}^{\bar{\mathbf{P}}} \tag{5}$$

$$\mathbf{C} \in \mathbb{R}^{N\times D/2} \tag{6}$$

Finally, based on Eq. (5) and definition of cosine similarity, an affinity matrix $\mathbf{A}$ is defined by

$$\mathbf{A} = \left[\frac{1}{\|\mathbf{c}_v\|_2 \|\mathbf{c}_w\|_2} \mathbf{c}_v^* \cdot \mathbf{c}_w\right]_{1\leq v,w\leq N} \tag{7}$$

Note that $\mathbf{A}$ has intermediate embedding information at a sensing point of CNN. Assuming that feature maps sensed in L points, IEP knowledge is obtained by observing a set of intermediate embedding informations $[\mathbf{A}_l]_{1\leq l\leq L}$ and a set of their alterations $[\mathbf{A}_{l+1} - \mathbf{A}_l]_{1\leq l\leq L-1}$. Also, when we extract the top three components in $\mathbf{C}$ for three-dimensional visualization, it can be easily understood through visualization (see the supplementary material). The next section describes how to distill IEP knowledge in a transferable form.

MPNN for Distilling IEP Knowledge

The IEP knowledge obtained by the SPCA coincides with the purpose of CNN. But in the case of a simple student network, this knowledge is so sharp and complex, which can give over-constraint if transferred as is. So, in order to distill the IEP knowledge, we employ MPNN which can interpret inter-data relation effectively and give a task that estimates the next interim embedding stages from current stages. The overall structure of the proposed MPNN is illustrated in Fig. 4.

First, the node feature $\mathbf{h}_v$ and the initial edge feature $\mathbf{e}^0_{v,w}$ are defined using the two set of compressed PCs $\mathbf{C}_l$ and $\mathbf{C}_{l+1}$

which are sensed at two adjacent points. $\mathbf{h}_v$ uses $\mathbf{c}_{l+1,v}$ as it is, and $\mathbf{e}^0_{v,w}$ is obtaining by linear mapping (*LM*) of $\mathbf{c}_{l,v}$ and dimension-wise relation like Eq. (8).

$$\mathbf{e}^0_{v,w} = \bar{\mathbf{c}}_{l,v} \odot \bar{\mathbf{c}}_{l,w},\ \bar{\mathbf{c}}_{l,v} = \frac{LM\left(\mathbf{c}_{l,v}\right)}{\left\|LM\left(\mathbf{c}_{l,v}\right)\right\|_2}, \tag{8}$$

where $\odot$ stands for Hadamard product, and *LM* consists of a fully connected (FC) layer and a batch normalization (Ioffe and Szegedy 2015). Next, to update the edge feature, the message function *Msg* and the edge update function *Up* operate as follows.

$$\mathbf{m}^i_{v,w} = Msg\left(\mathbf{h}_v, \mathbf{h}_w, \mathbf{e}^i_{v,w}\right) = GLU\left(\left[\mathbf{h}_v - \mathbf{h}_w, \mathbb{E}\left(\mathbf{e}^i_{v,w}\right)\right]\right) \tag{9}$$

$$\mathbf{e}^{i+1}_{v,w} = Up\left(\mathbf{e}^i_{v,w}, \mathbf{m}^i_{v,w}\right) = \mathbf{e}^i_{v,w} + \mathbf{m}^i_{v,w}, \tag{10}$$

where $\mathbb{E}$ is a function that returns the average of all components of an input. *Msg* uses a gated linear unit (GLU) (Dauphin et al. 2017), and *Up* simply adds edge features and messages. Finally, the edge feature $\mathbf{e}^I_{v,w}$ is obtained by repeating the above process *I* times, and it is input to the readout function *Rd*, and the $(l+1)^{th}$ affinity matrix $\tilde{\mathbf{A}}_{l+1}$ is returned. This final process is expressed as follows.

$$Rd\left(\mathbf{e}^I_{v,w}\right) = \mathbb{E}\left(\mathbf{e}^I_{v,w}\right) \tag{11}$$

$$\tilde{\mathbf{A}}_{l+1} = \left[Rd\left(\mathbf{e}^I_{v,w}\right)\right]_{1\leq v,w\leq N} \tag{12}$$

We use Kullback-Leibler divergence (KLD) (Kullback 1997) as a loss function for learning the proposed MPNN.

$$\mathcal{L}^{MPNN} = KLD\left(\sigma\left(\mathbf{A}_{l+1}\right) \| \sigma\left(\tilde{\mathbf{A}}_{l+1}\right)\right), \tag{13}$$

where σ stands for the softmax function. The details of learning are described in the supplementary material.

In the proposed MPNN, each edge and message have clear meanings. The initial edge indicates the l^{th} interim embedding stage, and MPNN updates it to the $(l+1)^{th}$ interim embedding stage using messages, which indicate alteration of embedding stage. Therefore, we define them as an intermediate embedding knowledge $\mathbf{K}^{int}$ and alteration of embedding knowledge $\mathbf{K}^{alt}$ to transfer IEP knowledge, which are shown in Eq. (14) and Eq. (15), respectively.

$$\mathbf{K}^{int} = \left[\tilde{\mathbf{A}}_{l+1}\right]_{1\leq l\leq L-1} \tag{14}$$

$$\mathbf{K}^{alt} = \left[\mathbf{m}^i_{l,v,w}\right]_{1\leq l\leq L-1,\ 1\leq v,w\leq N,\ 1\leq i\leq I} \tag{15}$$

Since this knowledge is smoothened by neural layers, it can be more easily learned by the student network. Also, the proposed knowledge can interpret the embedding procedure through visualization, which can be verified in the Experiment Section.

Transferring IEP Knowledge to the Student

To transfer the IEP knowledge distilled from the teacher network, a distillation module must be applied even to the student network. First, SPCA is applied to the matrix-formed feature map $\mathbf{F}^S_l$ which is sensed in the student network. Next, $\mathbf{C}^S_l$ is obtained using the information produced by the SPCA of the teacher network as follows.

$$\mathbf{P}^S_l = \left[\mathbf{p}^S_{l,n}\right]_{1\leq n\leq N} = \left[\frac{s_n\left(\mathbf{u}^S_{l,n}\right)^* \cdot \mathbf{F}^S_n}{\left\|\left(\mathbf{u}^S_{l,n}\right)^* \cdot \mathbf{F}^S_n\right\|_2}\right]_{1\leq n\leq N} \tag{16}$$

$$\mathbf{C}^S_l = \left[\left(\mathbf{p}^S_{l,n} - \boldsymbol{\mu}^{\mathbf{P}}\right) \cdot \mathbf{V}^{\mathbf{P}}\right]_{1\leq n\leq N} \tag{17}$$

The generated $\mathbf{C}^S_l$ is then inputted into the MPNN, which shares parameters with the teacher network, to distill the knowledge of the student network. The knowledge should be transferred in a way that minimizes the difference between the distilled knowledge of the teacher and student networks. So, the loss functions $\mathcal{L}^{int}$ and $\mathcal{L}^{alt}$ are defined by Eq. (18) and (19).

$$\mathcal{L}^{int} = KLD\left(\sigma\left(\mathbf{K}^{S,int}\right) \| \sigma\left(\mathbf{K}^{T,int}\right)\right) \tag{18}$$

$$\mathcal{L}^{alt} = \frac{1}{N^2}\left\|\mathbf{K}^{S,alt} - \mathbf{K}^{T,alt}\right\|_1 \tag{19}$$

KLD is adopted to avoid putting too strong constraints on $\mathbf{K}^{int}$ (the 2nd term of Eq. (18)). Since $\mathbf{K}^{alt}$ is a key information, strong constraint using L_1-norm is given to $\mathbf{K}^{alt}$ (the 1st term of Eq. (18)). So, the proposed method transfers two kinds of knowledge to the student network. Thus, the student network learns totally three tasks simultaneously, including a target task.

However, when transferring knowledge in the middle of CNN, the gradient of the transfer task can be much larger than that of the target task. This becomes over-constraint on the student network. Therefore, an appropriate multi-task learning technique is required to prevent such a phenomenon.

(Lee, Kim, and Song 2018) proposed a gradient clipping based on the norm of the target task's gradient. Inspired from (Lee, Kim, and Song 2018), we propose to clip the gradient obtained by knowledge based on the norm of the target task's gradient. Specific details are as follows.

$$\frac{\partial\Theta}{\partial\mathcal{L}^{Total}} = \frac{\partial\Theta}{\partial\mathcal{L}^{Target}} + clip\left(\frac{\partial\Theta}{\partial\mathcal{L}^{int}}\right) + clip\left(\frac{\partial\Theta}{\partial\mathcal{L}^{alt}}\right) \tag{20}$$

$$clip\left(z\right) = \max\left(1, \left\|\frac{\partial\Theta}{\partial\mathcal{L}^{Target}}\right\|_2 / \left\|z\right\|_2\right) z \tag{21}$$

As a result, the knowledge of the teacher network can be transferred as much as possible without over-constraint.

Black-box Knowledge Distillation via Multi-head Graph Distillation

The proposed method focuses on distilling CNN's IEP knowledge. However, in fact, CNN has black-box knowledge that cannot be interpreted due to its inherent characteristic. So distilling black-box knowledge is required to deliver complete knowledge of CNN. We generate black-box knowledge $\mathbf{K}^{BB}$ by fusing the multi-head graph distillation (MHGD) (Lee and Song 2019) and linguistic-informed self-attention (LISA) (Strubell et al. 2018). MHGD is a method of distillation of arbitrary relations interpreted by

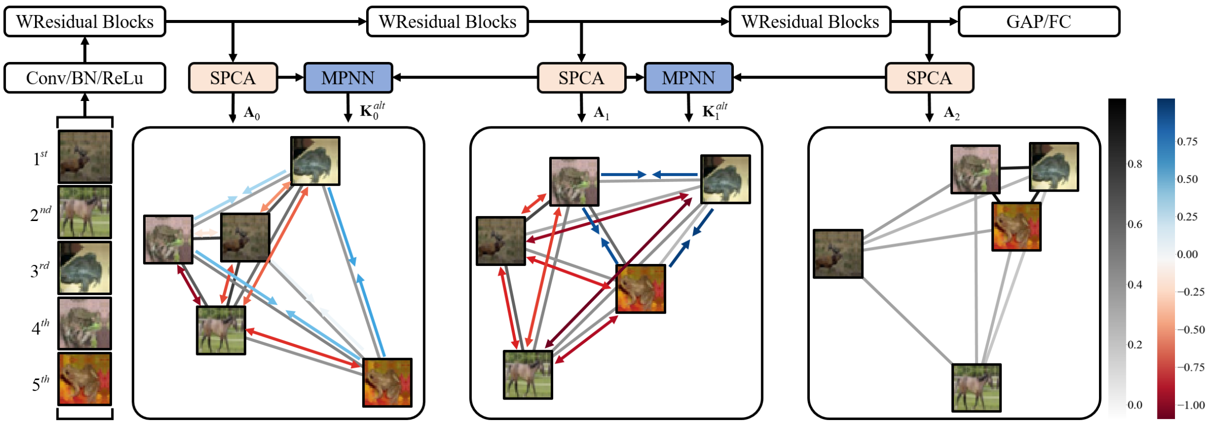

Figure 5: Visualization of $\mathbf{A}$ and $\mathbf{K}^{alt}$ obtained through the proposed distillation module. $\mathbf{A}$ and $\mathbf{K}^{alt}$ contain information of interim embedding stages and their alterations, respectively. For clear visualization, $\mathbf{A}$ is represented by gray scale and $\mathbf{K}^{alt}$ by RdBu colormap.

CNN through an attention network. LISA also adds extrinsic information to the attention network so that the attention network can extract an arbitrary relation of data. Next, similarly to LISA, the black-box knowledge $\mathbf{K}^{BB}$ is distilled by injecting the IEP knowledge into MHGD. The detailed structure is explained in the supplementary material.

The proposed method can transfer all kinds of embedding procedure knowledge from the teacher network, i.e., black-box knowledge as well as IEP knowledge, to the student network. Therefore, the student network receives clear guidance about the embedding procedure from the teacher network, which leads to additional performance improvement.

Experimental Results

This section shows the results of three kinds of experiments. First, we visualize the information of the proposed IEP knowledge. Second, we evaluate the basic performance of the proposed method through experiments on small network enhancement and transfer learning. Third, the effect of each component of the proposed method on the overall performance is verified. Additional evaluation results to show the effectiveness of the proposed method can be found in the supplementary material. We used four neural network architectures for implementing the proposed method: WResNet (Zagoruyko and Komodakis 2016b), ResNet (He et al. 2016), MobileNet-V2 (Sandler et al. 2018), and VGG (Simonyan and Zisserman 2014). Also, one of the most popular methods, attention transfer (AT) (Zagoruyko and Komodakis 2016a) as well as four SOTA methods: factor transfer (FT) (Kim, Park, and Kwak 2018), activation boundary (AB) (Heo et al. 2018), relational knowledge distillation (RKD) (Park et al. 2019), and comprehensive overhaul (CO) (Heo et al. 2019). were adopted for comparison with the proposed method. In addition, MHGD (Lee and Song 2019), which has a similar concept to the proposed method, was compared. We implemented all the techniques to be compared by ourselves. Detailed information such as network structures and hyper-parameters is described in the supplementary material.

Visualization of Embedding Procedure

This section visualizes the proposed IEP knowledge that is configured by $\mathbf{A}$ and $\mathbf{K}^{alt}$. For this experiment, WResNet40-4 trained on CIFAR10 dataset (Krizhevsky and Hinton 2009) was used. First, color mapping was employed to clearly display the numerical values of the each knowledge. Next, the edge lengths were manually adjusted according to the strength of the relation. Figure 5 shows the visualization result.

Note that through the neural layers of CNN, a given data is embedded as it is analyzed from low-level features to high-level features. If the embedding procedure of CNN is visualized, it is expected that data can be clustered according to the similarity of low-level features in the early stage of CNN and the similarity of high-level features in the late stage. In fact, Fig. 5 shows that the changes in the affinity matrices of the proposed knowledge are consistent with the above expectation. The feature maps of the initial layer have a strong relation to visually similar data, i.e., the 1^{st} to 4^{th} feature maps. On the other hand, since high-level features are analyzed as the later layers progress, we can observe that feature maps has a strong relation to the context, that is, the data of the same class, e.g., the 3^{rd}, 4^{th}, and 5^{th} feature maps. Accordingly, $\mathbf{K}^{alt}$ obtained by the proposed method has a positive value for data of the same class and a negative value for data of different classes.

Thus, the proposed IEP knowledge can be a tool for describing the embedding procedure, which is consistent with human intuition. Additional visualization results about full data in CIFAR10 can be found in the supplementary material.

Dataset	Rate	Student	AT	FT	AB	RKD	MHGD	CO	IEP	IEP+Black-box
CIFAR100	Full	76.09	76.98	77.14	77.29	77.02	77.45	78.21	78.12	**78.37**
	0.50	69.77	71.13	72.41	72.28	69.57	73.32	74.33	74.22	**74.53**
	0.25	59.28	63.07	63.70	66.79	53.57	67.27	67.90	68.57	**69.02**
	0.10	40.65	47.66	48.29	57.38	23.27	54.58	40.80	55.89	**59.04**
TinyImageNet	Full	59.71	60.92	55.61	60.19	61.12	62.26	63.56	63.29	**63.73**
	0.50	52.53	54.50	55.81	54.41	54.09	56.56	59.14	58.56	**59.27**
	0.25	43.56	46.54	39.19	48.99	42.19	50.59	52.56	53.20	**53.68**
	0.10	28.44	32.38	34.08	42.18	20.90	38.28	34.73	43.00	**45.01**

Table 1: Small network enhancement performance comparison of several KD methods for CIFAR100 and TinyImageNet datasets with various sample rates.

Dataset	Rate	Student	AT	FT	AB	RKD	MHGD	CO	IEP	IEP+Black-box
CUB200-2011	Full	52.21	58.87	59.96	56.80	52.54	55.77	60.83	60.13	**61.35**
	0.50	30.58	39.51	42.94	39.77	29.72	34.02	37.61	42.24	**43.06**
	0.25	14.25	19.68	21.18	20.52	14.15	18.41	14.29	22.00	**22.60**
	0.10	5.87	8.05	8.04	7.03	6.60	5.97	4.61	8.74	**9.69**
MIT-scene	Full	51.00	56.32	60.07	59.52	53.50	47.90	57.72	59.32	**60.94**
	0.50	36.83	42.43	46.53	46.80	39.18	36.48	35.16	45.83	**47.85**
	0.25	21.59	28.54	31.96	33.13	25.39	25.51	21.14	33.83	**34.28**
	0.10	10.59	14.44	14.39	19.79	12.17	10.07	6.07	18.44	**19.94**

Table 2: Transfer learning performance comparison of several KD methods for CUB-200-2011 and MIT-scene datasets with various sample rates.

Small Network Enhancement with Sampled Dataset

This section examines the small network enhancement performance of the proposed method on CIFAR100 (Krizhevsky and Hinton 2009) and TinyImageNet (Deng et al. 2009). In this experiment, the teacher and student networks were trained on the same dataset, but the student network used sampled datasets. Four sampling rates (full, 0.5, 0.25, and 0.1) were considered. WResNet40-4 was used as the teacher network. The teacher network provides the performance of 77.52% on CIFAR100 and 62.30% on TinyImageNet. We used WResNet16-4 as the student network, and set the hyper-parameters such that we can get the best performance when using the full dataset (rate = full).

Since the proposed method transfers embedding procedure knowledge, high performance can be expected thanks to the excellence of knowledge if the teacher and student networks are trained with the same dataset. Table 1 proves this assumption by comparing the proposed method with the existing KD methods. We can find that the proposed method outperforms other KD schemes at all sampling rates. The larger the sampling rate, that is, the lower the rate, the better the proposed method. For example, when the rate is 10%, the IEP+Black-box at CIFAR100 performed 1.66% higher than AB that is the best of SOTA techniques. Also, it provided 2.83% higher performance than AB for TinyImageNet. This is because the proposed KD method can transfer the most accurate knowledge that the student network needs to perform the target task.

Transfer Learning

The next experiment is about transfer learning. As the teacher network, ResNet32 which was pre-trained with ImageNet-2012 (Russakovsky et al. 2015) was employed. As the student network, ResNet14 which was learned with MIT-scene (Quattoni and Torralba 2009) and CUB200-2012 (Wah et al. 2011) datasets was used. Also, the student network was pre-trained with teacher knowledge and fine-tuned with the target dataset. The experimental results are shown in Table 2. We can observe that the proposed method, IEP+Black-box, is always superior to conventional techniques. However, because the target datasets of the teacher and student networks do not match, the performance improvement in Table 2 tends to be somewhat lower than that in Table 1. Therefore, to apply the proposed method most effectively, it is important to use the teacher network trained on the same dataset as the target dataset.

Knowledge Transfer to Heterogeneous Student Network

In this section, we tried to transfer teacher knowledge to several heterogeneous student networks that have different architectures from the teacher network to verify knowledge's dependency on network architecture. As a dataset for this experiment, 25% sampled CIFAR100 was used, and the teacher network was set to WResNet40-4. WResNet16-2 and WResNet16-1 were used as student networks that have different dimensional feature maps. Also, we employed MobileNet-V2 which is one of the famous light-

Architecture	Student	AT	FT	AB	RKD	MHGD	CO	IEP+Black-box
WResNet16-2	56.61	59.42	57.28	62.53	54.27	59.29	60.21	**63.78**
WResNet16-1	51.88	53.01	50.95	55.01	48.46	50.72	52.67	**56.09**
MobileNet-V2	56.96	59.04	57.48	61.35	58.17	61.80	62.72	**64.82**
VGG	47.76	49.88	48.13	N/A	N/A	47.40	45.18	**55.82**

Table 3: Small network enhancement performance comparison for different architecture or feature depth with teacher network.

Dataset	Student	$\mathbf{K}^{int}$	$\mathbf{K}^{alt}$	IEP	$\mathbf{K}^{BB}$
CIFAR100	59.28	61.77	68.14	68.57	67.75
TINY	43.56	45.90	52.62	53.26	51.92

Table 4: Ablation study for each proposed knowledge. TINY denotes TinyImageNet.

Sample rate	100%	50%	25%	10%
PCA-IPCA	78.12	74.22	68.57	55.89
PCA-PCA	77.92	73.66	66.58	51.88

Table 5: Performance comparison of PCA and IPCA with various batch sizes.

Iteration	1	2	3	4
CIFAR100	67.91	68.57	68.44	66.63
TinyImageNet	52.81	53.20	53.28	51.55

Table 6: Performance according to the number of message passing iterations.

weight architecture that has different layer modules, and finally adopted VGG having a quite different architecture with WResNet. Table 3 shows that the proposed method outperforms others for all of the network architectures. For example, in the case of VGG, the proposed method noticeably improved the performance of the student network while the other methods failed or were less effective. This result experimentally proves that the proposed method can distill the network's authentic knowledge that is independent of network architecture.

Ablation Study

CIFAR100 and TinyImageNet datasets were used for ablation study. For a clear comparison, we experimented with the 25% sampled datasets showing significant performance improvements. First, Table 4 shows how the three types of knowledge, $\mathbf{K}^{int}$, $\mathbf{K}^{alt}$, and $\mathbf{K}^{BB}$, can help improve network performance. It is worth noting that even the student network that has only one knowledge transferred delivers significant performance. In particular, $\mathbf{K}^{alt}$ and $\mathbf{K}^{BB}$, i.e., knowledge related to alteration, can outperform the SOTA techniques.

Next, we investigated the effect of the SPCA for obtaining the proposed knowledge. As mentioned earlier, if IPCA is not used as the second PCA, accurate PCs will not be obtained, and inter-data relations will not be predicted well. So, MPNN concentrates on memorizing data rather than accurately estimating relations. As a result, network performance is degraded because the intended knowledge is not distilled. Table 5 supports this analysis. When using the normal PCA as the second PCA, the performance tends to decrease, and the lower the sampling rate, the larger the degradation. This is because the distillation module makes it easier to memorize data. Therefore, we can find that IPCA is essential for obtaining the proposed knowledge. For reference, the visualization of the embedded dataset obtained by each method can be found in the supplementary material.

Finally, we take a look at the performance according to the number of iterations of MPNN. As the number of iterations increases, more accurate IEP knowledge can be obtained. However, if MPNN uses too many layers, the capacity increases, so there is a risk of focusing on memorizing data. Table 6 demonstrates this assumption. In fact, we can see that the performance increases as the number of iterations increases, but at a certain point, the performance starts decreasing. Our experiments show that two to three iterations are most advantageous for the best performance.

Conclusion

Knowledge distillation has recently been proven to be effective in a variety of computer vision problems with significant advances. However, research on knowledge distillation is currently too focused on performance. We thought that knowledge distillation could be an essential tool for understanding the nature of deep neural networks, not just a technique. So we validated our idea by defining new interpretable knowledge and suggesting a graph neural network based on interpretable knowledge. Thus, the experimental result shows that the proposed method outperforms SOTA methods in various datasets.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government (MSIT) [2020-0-01389, Artificial Intelligence Convergence Research Center (Inha University)] and National R&D Program through the National Research Foundation of Korea(NRF) funded by Ministry of Science and ICT(2020M3H2A1078119).

References

Ahn, S.; Hu, S. X.; Damianou, A.; Lawrence, N. D.; and Dai, Z. 2019. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9163–9171.

Apostol, T. M. 1964. *Mathematical analysis*. Addison-Wesley Reading.

Cranmer, M. D.; Xu, R.; Battaglia, P.; and Ho, S. 2019. Learning Symbolic Physics with Graph Networks. *arXiv preprint arXiv:1909.05862* .

Dauphin, Y. N.; Fan, A.; Auli, M.; and Grangier, D. 2017. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 933–941. JMLR. org.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Ge, S.; Zhao, S.; Li, C.; and Li, J. 2019. Low-resolution face recognition in the wild via selective knowledge distillation. *IEEE Transactions on Image Processing* 28(4): 2051–2062.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 1263–1272.

Hao, W.-L.; and Zhang, Z. 2016. Incremental PCANet: A lifelong learning framework to achieve the plasticity of both feature and classifier constructions. In *International Conference on Brain Inspired Cognitive Systems*, 298–309. Springer.

Härkönen, E.; Hertzmann, A.; Lehtinen, J.; and Paris, S. 2020. GANSpace: Discovering Interpretable GAN Controls. *arXiv preprint arXiv:2004.02546* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Henrion, I.; Brehmer, J.; Bruna, J.; Cho, K.; Cranmer, K.; Louppe, G.; and Rochette, G. 2017. Neural message passing for jet physics. In *Proceedings of the Deep Learning for Physical Sciences Workshop at NIPS*.

Heo, B.; Kim, J.; Yun, S.; Park, H.; Kwak, N.; and Choi, J. Y. 2019. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, 1921–1930.

Heo, B.; Lee, M.; Yun, S.; and Choi, J. Y. 2018. Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons. *arXiv preprint arXiv:1811.03233* .

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

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* .

Ionescu, C.; Vantzos, O.; and Sminchisescu, C. 2015. Matrix backpropagation for deep networks with structured layers. In *Proceedings of the IEEE International Conference on Computer Vision*, 2965–2973.

Kim, J.; Park, S.; and Kwak, N. 2018. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems*, 2760–2769.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Kullback, S. 1997. *Information theory and statistics*. Courier Corporation.

Lee, S.; and Song, B. C. 2019. Graph-based Knowledge Distillation by Multi-head Self-attention Network. *arXiv preprint arXiv:1907.02226* .

Lee, S. H.; Kim, D. H.; and Song, B. C. 2018. Self-supervised knowledge distillation using singular value decomposition. In *European Conference on Computer Vision*, 339–354. Springer.

Meng, Z.; Adluru, N.; Kim, H. J.; Fung, G.; and Singh, V. 2018. Efficient relative attribute learning using graph neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 552–567.

Nguyen, H.; Maeda, S.-i.; and Oono, K. 2017. Semi-supervised learning of hierarchical representations of molecules using neural message passing. *arXiv preprint arXiv:1711.10168* .

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational Knowledge Distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3967–3976.

Quattoni, A.; and Torralba, A. 2009. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 413–420. IEEE.

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

Ross, D. A.; Lim, J.; Lin, R.-S.; and Yang, M.-H. 2008. Incremental learning for robust visual tracking. *International journal of computer vision* 77(1-3): 125–141.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115(3): 211–252. doi:10.1007/s11263-015-0816-y.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520.

Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .

Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; and McCallum, A. 2018. Linguistically-informed self-attention for semantic role labeling. *arXiv preprint arXiv:1804.08199* .

Valmadre, J.; Bertinetto, L.; Henriques, J.; Vedaldi, A.; and Torr, P. H. 2017. End-to-end representation learning for correlation filter based tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2805–2813.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Zagoruyko, S.; and Komodakis, N. 2016a. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928* .

Zagoruyko, S.; and Komodakis, N. 2016b. Wide residual networks. *arXiv preprint arXiv:1605.07146* .