Triplet Loss for Knowledge Distillation

Hideki Oki, Motoshi Abe, Jyunichi Miyao and Takio Kurita
Department of Information Engineering
Hiroshima University
Higashi-Hiroshima, Japan
{hidebon9105, i13abemotoshi}@gmail.com, {miyao, tkurita}@hiroshima-u.ac.jp

***Abstract*—In recent years, deep learning has spread rapidly, and deeper, larger models have been proposed. However, the calculation cost becomes enormous as the size of the models becomes larger. Various techniques for compressing the size of the models have been proposed to improve performance while reducing computational costs. One of the methods to compress the size of the models is knowledge distillation (KD). Knowledge distillation is a technique for transferring knowledge of deep or ensemble models with many parameters (teacher model) to smaller shallow models (student model). Since the purpose of knowledge distillation is to increase the similarity between the teacher model and the student model, we propose to introduce the concept of metric learning into knowledge distillation to make the student model closer to the teacher model using pairs or triplets of the training samples. In metric learning, the researchers are developing the methods to build a model that can increase the similarity of outputs for similar samples. Metric learning aims at reducing the distance between similar and increasing the distance between dissimilar. The functionality of the metric learning to reduce the differences between similar outputs can be used for the knowledge distillation to reduce the differences between the outputs of the teacher model and the student model. Since the outputs of the teacher model for different objects are usually different, the student model needs to distinguish them. We think that metric learning can clarify the difference between the different outputs, and the performance of the student model could be improved. We have performed experiments to compare the proposed method with state-of-the-art knowledge distillation methods. The results show that the student model obtained by the proposed method gives higher performance than the conventional knowledge distillation methods.**

***Index Terms*—Convolutional Neural Network, Metric Learning, triplet loss, knowledge distillation**

I. Introduction

After the deep Convolutional Neural Network (CNN) proposed by Krizhevsky et al. [1] won the ILSVRC 2012 with higher score than the conventional methods, it became very popular for image classification and object recognition. Later, with the increase in computer performance, deeper and larger models such as VGG [2] and ResNet [3] were proposed. It is believed that deeper models with more parameters perform better performance than the shallow models. However, the calculation cost becomes enormous as the size of the models becomes larger. Various techniques for compressing the size of the models have been proposed to improve performance while reducing computational costs.

One of the methods to compress the size of the models is knowledge distillation (KD). Knowledge distillation is a technique for transferring knowledge of deep or ensemble models with many parameters (teacher model) to smaller shallow models (student model). Specifically, the learning of the student model is accelerated by using the output of the trained teacher model. For example, Ba et al. [4] used the square error between the teacher model and the student model as the student model loss. Hinton et al. [5] used the KL-divergence of the softmax output of both models as the loss of the student model.

In knowledge distillation, the knowledge of the teacher model is transferred to the student model by making the student model learn to imitate the output of the teacher model. In other words, knowledge distillation can also be considered as a method for increasing the similarity between the outputs of the teacher model and the student model. So, in this paper, we propose to introduce metric learning into knowledge distillation.

In metric learning, the researchers are developing the methods to build a model that can increase the similarity of outputs for similar samples. Representative methods for deep metric learning include the Siamese Network [6]–[8] and the Triplet Network [9], [10]. These are learned so that the Euclidean distance between outputs for similar samples is reduced. In addition, learning is performed so that the Euclidean distance between outputs for dissimilar samples becomes large. That is, there is a function to clarify the difference between dissimilar outputs. The model will try to get the same output for the same object, and will be able to clearly distinguish the difference between different objects. With this feature, the metric learning model has performed extremely well in areas such as face verification and human re-identification.

In this paper, we propose to introduce this metric learning concept into knowledge distillation. The functionality of the metric learning to reduce the differences between similar outputs can be used for the knowledge distillation to reduce the differences between the outputs of the teacher model and the student model. Since the outputs of the teacher model for different objects are usually different, the student model needs to distinguish them. We think that metric learning can clarify the difference between the different outputs, and the performance of the student model could be improved.

The effectiveness of the proposed approach is experimentally confirmed for image classification tasks. Experimental results show that our method dramatically improves the per-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

formance of the student model. Also, we have performed experiments to compare the proposed method with state-of-the-art knowledge distillation methods. The results show that the student model obtained by the proposed method gives higher performance than the conventional knowledge distillation methods.

II. Related Works

A. Deep Convolutional Neural Network

The deep CNN has been proven to be effective and has been applied many applications such as for image classification [1], [3], object detection [11], image segmentation [12] and so on.

The computation within the convolution layers is regarded as a filtering process of the input image as

$$f^{(c)}_{p,q} = h\Big(\sum_{r=0}^{convy-1} \sum_{s=0}^{convx-1} w^{(c)}_{r,s} f^{(c-1)}_{p+r,q+s} + b^{(c)} \Big) \ , \tag{1}$$

where $w^{(c)}_{r,s}$ is the weight of the neuron indexed as (r, s) in the c-th convolution layer and $b^{(c)}$ is the bias of the c-th convolution layer. The size of the convolution filter is given as $convx \times convy$. The activation function of each neuron is denoted as h. Usually, pooling layers are added after the convolution layers. The pooling layer performs downsampling for reducing computational costs and enhancing against micro position changes. Fully-connected layers like multi layer perceptron is connected to the convolution layers which is used to construct the classifier.

B. Metric Learning

Metric learning is a method of learning embedding for metric (Euclidean distance, cosine similarity, etc.). Metric learning is applied to a wide range of tasks such as image search, biometric authentication, and abnormality detection.

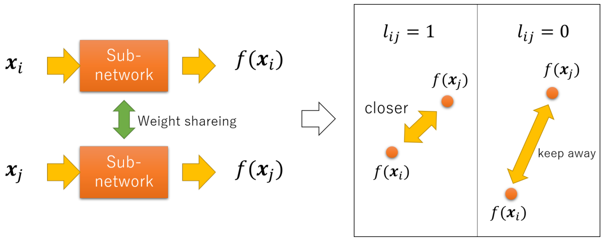

Fig. 1. The structure of Siamese Network. The distance metric is learned depending on the label l_{ij} given to each sample pair $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

A typical model for deep learning embedding is the Siamese Neural Network [6]–[8]. The Siamese Network consists of two identical networks joined at their outputs. The two networks extract feature vectors from two different samples. Usually, the weights of the two networks are shared. The objective function of the optimization for training the parameters of the networks is defined by using these extracted feature vectors as

$$\begin{aligned} E = \frac{1}{2|\chi^2|} & \sum_{(i,j)\in\chi^2} l_{ij}(D_{ij})^2 \\ & + (1 - l_{ij}) max(m - D_{ij},\ 0)^2 \ , \end{aligned} \tag{2}$$

where D_{ij} represents the distance between the pair of the outputs $f(\boldsymbol{x}_i)$ and $f(\boldsymbol{x}_j)$ of each network for the sample pair $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The distance D_{ij} is defined by

$$D_{ij} = ||f(\boldsymbol{x}_i) - f(\boldsymbol{x}_j)||_2 \ , \tag{3}$$

where m is a parameter indicating the distance between clusters and χ^2 is an index set of sample pairs randomly generated from the samples in the mini-batch. A label l is assigned for each sample pair such that label is $l_{ij} = 1$ when the pair i and j is similar and label is $l_{ij} = 0$ when the pair i and j is dissimilar. The scheme of Siamese Network is shown in Fig. 1.

After the training of Siamese Network, the distance between the outputs for dissimilar pair will be far while the distance between the outputs for similar pair become close. The Siamese Network can only consider the metric between samples pairwise. For this reason, the Siamese Network have to uniquely determine the concept of similarity. For example, if there are two different male images, in the case of the concept of "gender", they should be judged to be similar. However, in the case of the concept of "individuals", they should be judged not to be similar. It is difficult to express these multiple concepts in the Siamese Network.

Wang et al. [9], Hoffer et al. [10] proposed the Triplet Network (Figure 2), an extension of the Siamese Network to triplet-wise.

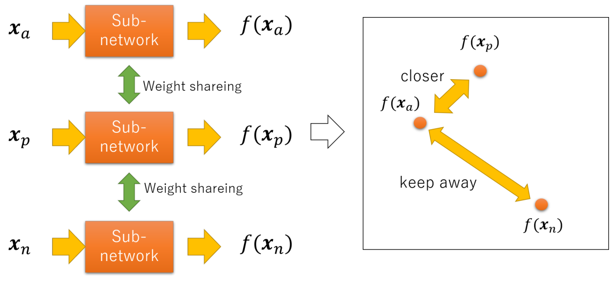

Fig. 2. The structure of Triplet Network. The distance metric is learned by using three networks with the shared weights from triplet $\boldsymbol{x}_a$, $\boldsymbol{x}_p$ and $\boldsymbol{x}_n$ which are called "anchor", "positine", and "negative". Learning progresses as closer "anchor-positive", and keep away "anchor-negative".

Triplet Network is designed to learn embedding from a triplet of samples called "anchor", "positive" and "negative". The triplet network learns embedding such that the distance between "anchor" $\boldsymbol{x}_a$ and "positve" $\boldsymbol{x}_p$ is smaller than the distance between "anchor" $\boldsymbol{x}_a$ and "negative" $\boldsymbol{x}_n$. Various losses have been proposed for the Triplet Network. The triplet loss

$$\begin{aligned} E = \sum_{(a,p,n)\in\Theta} & max(0, m + ||f(\boldsymbol{x}_a) - f(\boldsymbol{x}_p)||_2^2 \\ & - ||f(\boldsymbol{x}_a) - f(\boldsymbol{x}_n)||_2^2) \end{aligned} \tag{4}$$

is generally used, where $f(\cdot)$ denotes the output of the model, and m is a parameter indicating the distance between clusters. Θ is an index set of "anchor", "positive", and "negative". The triplet network learns so that the anchor-positive distance is relatively closer than the anchor-negative distance. Therefore, multiple similar concepts can be considered without depending on one similar concept. Triplet Network can overcome the drawbacks of Siamese Network.

C. Knowledge Distillation

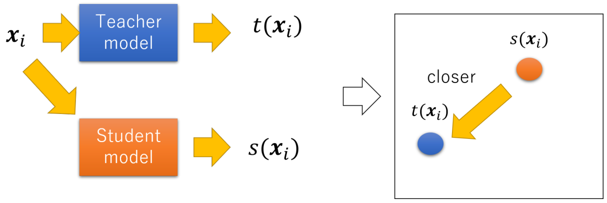

Fig. 3. The structure of conventional knowledge distillation. The student model tries to mimic the teacher model.

Knowledge distillation is a technique for transferring knowledge of deep or ensemble model with many parameters (teacher model) to smaller shallow model (student model). Various approaches have been proposed for transferring knowledge.

Ba et al. [4] considered the difference between the input vectors of the softmax activation function (logits) of the teacher model and the student model as loss (Equation(5)), and proposed to train the student model so that the feature vectors of the student model got close to the teacher model (BKD). The loss to measure the similarity between the teacher model and the student model is defined by the mean squared errors between the feature vector $t(\cdot)$ of the teacher model and the feature vector $s(\cdot)$ of the student model as

$$E_{BKD} = \frac{1}{2}\sum_{i\in\chi} ||t(\boldsymbol{x}_i) - s(\boldsymbol{x}_i)||_2^2, \tag{5}$$

where χ is the set of training samples. As a result, the authors found that the performance of the student model was higher than the network trained by the student model alone. They claimed that the uncertainty of the feature vectors of the teacher model is useful for learning the student model.

Hinton et al. [5] proposed training the student model so that the softmax outputs of the teacher model and the softmax outputs (probability) of the student model are close (HKD). They used the KL-divergence of the softmax outputs of both models as the loss for training the student model. The loss based on the KL-divergence is defined as

$$E_{HKD} = \sum_{i\in\chi} KL(\text{softmax}(\frac{t(\boldsymbol{x}_i)}{T}), \text{softmax}(\frac{s(\boldsymbol{x}_i)}{T})), \tag{6}$$

where softmax$(\cdot)$ is the softmax function, $KL(\cdot)$ is KL-divergence between the softmax outputs of the teacher model $\{p_i\}$ and the softmax outputs $\{q_i\}$ of the student model defined by

$$KL(\boldsymbol{p}, \boldsymbol{q}) = \sum_i p_i log(\frac{p_i}{q_i}) \ , \tag{7}$$

and T is a temperature parameter. They argued that negative values in logits could have a positive effect in some cases or negative effect in some cases, on learning.

In recent years, various approaches have been proposed for knowledge distillation. Park et al. [13] expressed the relationship between the outputs of the teacher model as the Euclidean distance between the two outputs, and transferred it to the student model (RKD-D). They expressed the similarity between the pair of the outputs $t(\boldsymbol{x}_i)$ and $t(\boldsymbol{x}_j)$ as

$$\psi_D(t(\boldsymbol{x}_i), t(\boldsymbol{x}_j)) = \frac{||t(\boldsymbol{x}_i) - t(\boldsymbol{x}_j)||_2}{\sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\chi^2} ||t(\boldsymbol{x}_i) - t(\boldsymbol{x}_j)||_2}, \tag{8}$$

where χ^2 is a set of sample pairs randomly generated from the samples in the mini-batch. They optimized the student model by Huber loss so that the similarity between the outputs of the student model and the similarity of the outputs of the teacher model got closer. The loss function of RKD-D is defined by using the similarity ψ_D as

$$E_{RKD-D} = \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\chi^2} l(\psi_D(t(\boldsymbol{x}_i), t(\boldsymbol{x}_j)), \psi_D(s(\boldsymbol{x}_i), s(\boldsymbol{x}_j))), \tag{9}$$

where

$$l(p, q) = \begin{cases} \frac{1}{2}(p-q)^2 & (|p-q| \leq 1) \\ |p-q| - \frac{1}{2} & (otherwise) \end{cases} \tag{10}$$

is Huber loss.

They also used the cosine of the angle formed by the three outputs as the similarity of the model outputs (RKD-A). The cosine of the three outputs $t(\boldsymbol{x}_i)$, $t(\boldsymbol{x}_j)$, and $t(\boldsymbol{x}_k)$ is defined by

$$\psi_A^{(t_{ijk})} = \cos \angle t(\boldsymbol{x}_i)t(\boldsymbol{x}_j)t(\boldsymbol{x}_k), \tag{11}$$

where

$$\cos \angle t(\boldsymbol{x}_i)t(\boldsymbol{x}_j)t(\boldsymbol{x}_k) = \langle \boldsymbol{e}^{(t_{ij})}, \boldsymbol{e}^{(t_{kj})} \rangle \tag{12}$$

$$\begin{aligned} \boldsymbol{e}^{(t_{ij})} &= \frac{t(\boldsymbol{x}_i) - t(\boldsymbol{x}_j)}{||t(\boldsymbol{x}_i) - t(\boldsymbol{x}_j)||_2}, \\ \boldsymbol{e}^{(t_{kj})} &= \frac{t(\boldsymbol{x}_k) - t(\boldsymbol{x}_j)}{||t(\boldsymbol{x}_k) - t(\boldsymbol{x}_j)||_2} \ . \end{aligned} \tag{13}$$

Loss of RKD-A is also defined by using Huber loss [14] as

$$E_{RKD-A} = \sum_{(\boldsymbol{x}_i,\boldsymbol{x}_j,\boldsymbol{x}_k)\in\chi^3} l(\psi_A^{(t_{ijk})}, \psi_A^{(s_{ijk})}) \ , \tag{14}$$

where χ^3 is a set of triplet samples randomly generated from the samples in the mini-batch.

They also argued that using both angles and Euclidean distance for knowledge transfer would further improve the performance of the student model (RKD-DA). The loss for this case is defined as

$$E_{RKD-DA} = \lambda_{RKD-D} E_{RKD-D} + \lambda_{RKD-A} E_{RKD-A} \ , \tag{15}$$

where λ_{RKD-D} and λ_{RKD-A} are the hyper-parameters.

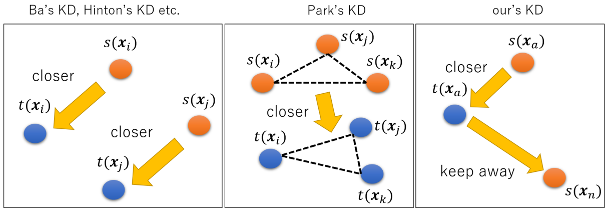

Fig. 4. The summary of knowledge distillation methods in terms of the loss function.

The pair-wise or triplet-wise similarity between the output of the teacher model to the output of the student model is considered in RKD-D, RKD-A, or RKD-DA while the point-wise transfer was performed in the knowledge distillation methods such as Ba's KD [4] or Hinton's KD [5]. Thus the knowledge distillation methods such as RKD-D, RKD-A, or RKD-DA succeeded in achieving top performance.

The loss using the output distribution of the teacher model (soft target) can be used alone or combined with the loss using the label of the original training data (hard target). The combined loss can be defined as

$$E_{KD} = E_{hard} + \lambda_{soft} E_{soft}, \tag{16}$$

where E_{hard} is hard target loss and E_{soft} is soft target loss. In this case, a hyper-parameter λ is introduced to keep the balance between the hard target and the soft target. It is also possible to combine multiple soft target losses. For example, in Park's KD, they are λ_{RKD-D} and λ_{RKD-A}.

III. Proposed Method

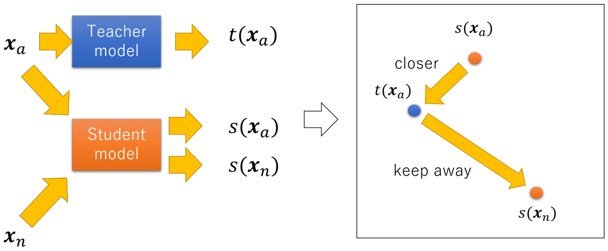

Fig. 5. The structure of our method. The student model is optimized so that its own output for sample $\boldsymbol{x}_a$ and the output of the teacher model for sample $\boldsymbol{x}_a$ are closer. At the same time, the student model is optimized so that its own output for sample $\boldsymbol{x}_n$ which is the different classes in the soft target and the output of the teacher model for sample $\boldsymbol{x}_a$ are keeping away.

Previously proposed methods of knowledge distillation focused on minimizing the difference between the teacher model and the student model in order to transfer knowledge of the teacher model. For example, Ba et al. [4] considered the difference in model output as loss and Hinton et al. [5] considered the difference in model output distribution as loss. In other words, they are solving the optimization problem in which the similarity between the function of the teacher model and the function of the student model is maximized. There is a possibility that knowledge of the teacher model can be transferred to the student model by using the metric learning method to embed the neighboring relations of the teacher model in the output space of the student model.

Triplet [10], one of the deep metric learning methods, is a powerful method for learning similarity and dissimilarity (Equation (4)). Triplet loss has a function to reduce the output distance of "anchor-positive" and a function to increase the output distance of "anchor-negative". We propose to apply this technique for knowledge distillation.

We define the triplet loss for knowledge transfer as

$$E_{ourKD} = \sum_{(a,n)\in\Omega} max(0, m + ||t(\boldsymbol{x}_a) - s(\boldsymbol{x}_a)||_2^2 - ||t(\boldsymbol{x}_a) - s(\boldsymbol{x}_n)||_2^2) \ , \tag{17}$$

where $t(\cdot)$ is the output of the teacher model, $s(\cdot)$ is the output of the student model, and m is the margin. Further, $\boldsymbol{x}_a$ is a training sample extracted at random, and $\boldsymbol{x}_n$ is a sample classified into a different class from $\boldsymbol{x}_a$ in the case of a soft target. Ω is the index set of each corresponding sample. So, the output of the teacher model and the output of the student model for the same sample are considered as "anchor" and "positive", respectively. Also we consider the output of the student model for samples of different classes in the case of soft target as "negative". Since we are training the student model, the weight of the teacher model is not updated. That is, $t(\cdot)$ is treated as a constant during training process.

For our loss, there is a term that makes the outputs of the teacher model and the student model closer. This is realized by making the square error of the outputs of the teacher model and the student model a loss. This is similar to Ba et al. [4] proposed function. It is easy to define the loss by using KL-divergence similar with Hinton et al. [5]. In addition, there are terms that increase the distance between outputs of different classes. In other words, in addition to the conventional loss functions, it is possible to add a constraint that the student model makes the outputs of other classes dissimilar. This will allow the student model to clarify differences in output between classes. We have shown experimentally that this proposal contributed to the performance improvement of the student model. We will also seek further performance improvements in combination with other distillation losses.

IV. Experiment

We experimentally investigated the effectiveness of the proposed method with the image classification task. CIFAR-10 [15] and Tiny ImageNet [16] are used as datasets.

A. Experiments using CIFAR-10

CIFAR-10 is a dataset that includes color images of 10 kinds of objects such as "automobile" and "dog." The size of each image in CIFAR-10 is 32 × 32. The numbers of the training samples and the test samples are 50,000 and 10,000. We conducted comparative experiments with Ba's KD [4] (BKD), Hinton's KD [5] (HKD), and Park's RKD-DA [13].

In the case of CIFAR-10, the 8-layers CNN model shown in Table I was used as a teacher model, and the 5-layers CNN

TABLE I
MODEL STRUCTURES FOR CIFAR-10

student model(5 layers)	teacher model(8 layers)
conv1(channel:32, filter:3)	conv1(channel:32, filter:3)
max pooling(2*2)	batch normalization(32)
conv2(channel:32, filter:3)	max pooling(2*2)
max pooling(2*2)	conv2(channel:32, filter:3)
conv3(channel:64, filter:3)	batch normalization(32)
max pooling(2*2)	max pooling(2*2)
fully-connected(128)	conv3(channel:64, filter:3)
output(10)	batch normalization(64)
	conv4(channel:64, filter:3)
	batch normalization(64)
	conv5(channel:128, filter:3)
	batch normalization(128)
	max pooling(2*2)
	fully-connected(512)
	dropout
	fully-connected(128)
	dropout
	output(10)

model was used as a student model. The number of trainable parameters of the student model is 161,130, and the number of trainable parameters of the teacher model is 1,256,106. The number of parameters in the student model is about 12.8% of the number of parameters in the teacher model.

An activation function ReLU defined by

$$h(x) = max(0, x) \tag{18}$$

is used for the outputs of each convolution layer and the outputs of each fully-connected layer except for the last output layer.

In the training, the optimization is done by using the stochastic gradient descent (SGD) with momentum. The learning rate of SGD is initially set at 0.01 and changed by multiplying 0.1 at every 100 epochs. The momentum parameter is set to 0.9. We also introduced weight decay to prevent over-learning. We set hard target loss as cross entropy with softmax and combined it with soft target loss. In addition, we introduced a parameter λ as shown in Equation (16) to maintain the balance between hard target loss and soft target loss. We set $\lambda_{BKD} = 2$, $\lambda_{HKD} = 16$, $\lambda_{RKD-D} = 10$, $\lambda_{RKD-A} = 20$ and $\lambda_{oursKD} = 2$. The temperature parameter of Hinton's KD [5] was set to 4, and the hyper-parameter m of triplet loss was set to 5.

TABLE II
CLASSIFICATION ACCURACY FOR CIFAR-10

method	accuracy
Student model	74.66%
Ba's KD	80.40%
Hinton's KD	79.16%
RKD-DA	79.21%
Ours KD	**81.14**%
Teacher model	83.42%

The Table II shows the classification accuracy of the student model for test samples in each method. From the table, it can be seen that in the case of CIFAR10, the proposed method achieves higher performance than the other methods.

B. Experiments using Tiny ImageNet

Also, we have performed experiments using the Tiny ImageNet dataset. Tiny ImageNet dataset includes color images of 200 kinds of objects. The image size is 64×64, and it has 500 train images and 50 test images per class. For the case of Tiny ImageNet, VGG19 with batch normalization and VGG11 were used as the teacher model and the student model, respectively. The numbers of trainable parameters of the teacher model and the student model are 46,028,808 and 35,213,896, respectively. This means that the number of parameters of the student model is about 76.5% of those of the teacher model. We conducted comparative experiments with Ba's KD [4] (BKD), Hinton's KD [5] (HKD), and Park's RKD-DA [13].

The weights in the convolution layers were pre-trained by using ImageNet, and they were used as the initial weights of the training. Similarly, the SGD with momentum was used for the optimization. The learning rate of SGD with momentum is initially set at 0.001 and multiplied by 0.9 every 3 epochs. The momentum parameter is set to 0.9. We also used mixup [17] to prevent over-learning. Cross-entropy with softmax was used as the hard target loss and was combined with the soft target loss. Also, we introduced a parameter λ as shown in Equation (16) to maintain the balance between hard target loss and soft target loss. We set $\lambda_{BKD} = 2$, $\lambda_{HKD} = 16$, $\lambda_{RKD-D} = 25$, $\lambda_{RKD-A} = 50$ and $\lambda_{oursKD} = 2$. The temperature parameter of Hinton's KD [5] was set to 4, and the parameter m of triplet loss was set to 5.

TABLE III
CLASSIFICATION ACCURACY FOR TINY IMAGENET

method	accuracy
Student model	58.63%
Ba's KD	59.80%
Hinton's KD	59.80%
RKD-DA	59.73%
Ours KD	**60.00**%
Teacher model	63.17%

Table III shows the classification accuracy of the student model for the Tiny ImageNet dataset in each method. The results show that the proposed method gives slightly better performance than the other methods.

C. Experiments on the combined loss

Park et al. [13] succeeded in achieving even higher performance by combining their method (RKD-DA) with Hinton's method [5]. Here we investigate the effectiveness of the combination of different loss functions. The loss of each method is combined with Hinton's loss as

$$E = E_{hard} + \lambda_{soft} E_{soft} + \lambda_{HKD} E_{HKD} \tag{19}$$

. Depending on the datasets CIFAR-10 and Tiny ImageNet, the same network architectures with the previous subsections were used. The student models were trained, and the classification performances of the trained student models are calculated.

TABLE IV
CLASSIFICATION ACCURACY FOR COMBINED WITH HKD

method	CIFAR10	Tiny ImageNet
Student model	74.66%	58.63%
Hinton's KD	79.16%	59.80%
Ba's KD + HKD	80.33%	60.19%
RKD-DA + HKD	79.65%	59.89%
Ours KD + HKD	**80.93%**	**60.62%**
Teacher model	83.42%	63.17%

Table IV shows the classification performance of the test samples for each dataset. From this table IV, it is noticed that the classification accuracy of Park's KD [13] was improved by combining with the Hinton's loss for CIFAR-10 dataset, but the classification accuracy of the proposed method and Ba's KD [4] was not improved. However the proposed method still gives the best classification accuracy. For the Tiny ImageNet dataset, all methods showed an increase in the classification accuracy when they are combined with Hinton's KD [5]. Also, the proposed method gives the best accuracy for this case.

Also we consider a combination with Park's KD [13]. We investigate the effects of combining RKD-DA loss with our loss and we investigate combinations with both Hinton's KD [5] and RKD-DA. Depending on the datasets CIFAR-10 and Tiny ImageNet, the same network architectures with the previous subsections were used.

TABLE V
CLASSIFICATION ACCURACY OF OURS KD FOR COMBINED WITH RKD-DA

method	CIFAR10	Tiny ImageNet
Student model	74.66%	58.63%
RKD-DA	79.21%	59.73%
ours KD	**81.14%**	60.00%
ours KD + RKD-DA	80.17%	**60.32%**
ours KD + HKD + RKD-DA	80.30%	**60.65%**
Teacher model	83.42%	63.17%

Table V shows the classification accuracy of the student model in each method. In the case of the CIFAR10 dataset, no performance improvement was obtained even when combined with RKD-DA. In the case of the Tiny ImageNet dataset, we succeeded in improving performance by combining it with RKD-DA. In addition, using both RKD-DA and Hinton's KD in combination with our method resulted in further performance improvements. Although the combination of our method and Hinton's KD had improved the performance, the combination of both Hinton's KD and RKD-DA achieved even higher performance.

V. CONCLUSION

In this paper, we applied the concept of metric learning to knowledge distillation. By introducing the concept of metric learning, we transfer not only the teacher model knowledge to be imitated but also the knowledge that should not be imitated to the student model. Our experimental results showed that this approach can significantly improve the performance of the student model. Also, the combination with Hinton's KD [5] and Park's KD [13] succeeded in giving the student model even better performance. This fact indicates that metric learning contributes to transferring the knowledge of the teacher model to the student model.

On the other hand, the concept of metric learning, such as Triplet loss, has a problem with hard sampling. In hard sampling, we intentionally select samples that are difficult to learn to get consistent training results. If you use only samples that are easy to learn, the performance of learning decreases. In the proposed approach, it is necessary to use hard sampling because the proposed method is based on the Triplet loss. Thus we have to introduce a way to address this problem, like Hermans et al. [18].

ACKNOWLEDGMENT

This work was partly supported by JSPS KAKENHI Grant Number 16K00239.

REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[4] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pages 2654–2662, 2014.

[5] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[6] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[7] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, pages 539–546, 2005.

[8] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.

[9] Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1386–1393, 2014.

[10] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92. Springer, 2015.

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[13] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[14] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics*, pages 492–518. Springer, 1992.

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Citeseer*, 2009.

[16] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015.

[17] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[18] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.