Communications for Statistical Applications and Methods
2023, Vol. 30, No. 5, 501–516

https://doi.org/10.29220/CSAM.2023.30.5.501
Print ISSN 2287-7843 / Online ISSN 2383-4757

Area-wise relational knowledge distillation

Sungchul Cho[a], Sangje Park[a], Changwon Lim[1,a]

[a]Department of Applied Statistics, Chung-Ang University, Korea

Abstract

Knowledge distillation (KD) refers to extracting knowledge from a large and complex model (teacher) and transferring it to a relatively small model (student). This can be done by training the teacher model to obtain the activation function values of the hidden or the output layers and then retraining the student model using the same training data with the obtained values. Recently, relational KD (RKD) has been proposed to extract knowledge about relative differences in training data. This method improved the performance of the student model compared to conventional KDs. In this paper, we propose a new method for RKD by introducing a new loss function for RKD. The proposed loss function is defined using the area difference between the teacher model and the student model in a specific hidden layer, and it is shown that the model can be successfully compressed, and the generalization performance of the model can be improved. We demonstrate that the accuracy of the model applying the method proposed in the study of model compression of audio data is up to 1.8% higher than that of the existing method. For the study of model generalization, we demonstrate that the model has up to 0.5% better performance in accuracy when introducing the RKD method to self-KD using image data.

Keywords: deep learning, model compression, knowledge distillation, audio signal processing, image processing

1. Introduction

In recent years, deep learning has shown epoch-making performance in various fields such as image recognition (Tan and Le, 2019), speech recognition (Saon *et al*., 2017), and natural language processing (Edunov *et al*., 2018). Deep learning models with many hidden layers appeared every year. In 2015, the residual neural network (ResNet) (He *et al*., 2016) with more than 100 hidden layers won the ImageNet large scale visual recognition challenge (ILSVRC) (Deng *et al*., 2009). Figure 1 shows the winner algorithms in ILSVRC from 2010 to 2015. We can see from the figure that as the model descended deeper, the accuracy improved, but the model became excessively large accordingly.

Despite these outstanding performances, the computational cost of the deep learning models due to their complexity and large number of parameters made them difficult to deploy on devices such as smartphones (Han *et al*., 2015). Therefore, in the field of machine learning, the compression of models is one of the most important factors because it can create a more user-friendly environment for models to be widely used. As a result, research on compressing deep learning models into relatively small models has continued to overcome high computational costs (Buciluǎ *et al*., 2006).

Many techniques have been studied and developed to solve the inverse problem of accuracy and computation time in deep learning. Recently, one of the most popular techniques for model compression is knowledge distillation (KD). The basic idea of KD is to compress the model by transferring

[1]Corresponding author: Department of Applied Statistics, Chung-Ang University, 47 Heukseok-ro, Dongjak-Gu, Seoul 06974, Korea. E-mail: clim@cau.ac.kr

Published 30 September 2023 / journal homepage: http://csam.or.kr
© 2023 The Korean Statistical Society, and Korean International Statistical Society. All rights reserved.

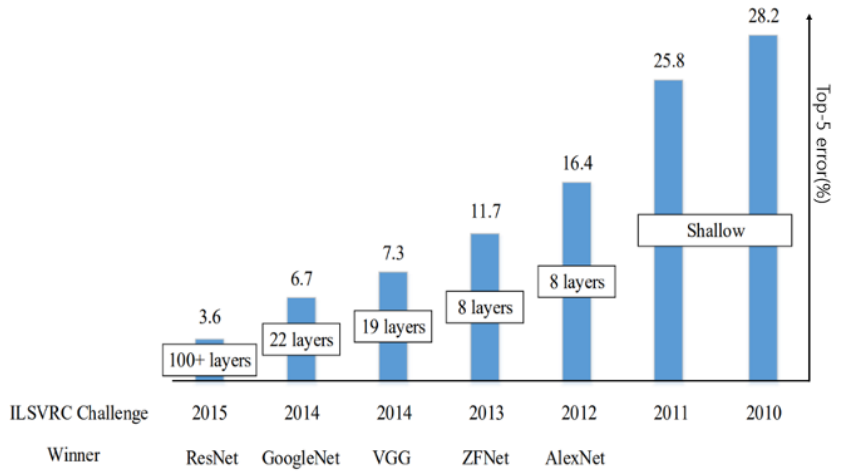

Figure 1: *The winner algorithms in ILSVRC (Zhang et al., 2017).*

the knowledge from the pre-trained, large, and complex model, the Teacher model, to the relatively small model, the student model (Hinton *et al*., 2015).

Hinton *et al*. (2015) performed model compression using the logit value, which is a pre-activated value in the output layer of the teacher model. Recently, Furlanello *et al*. (2018) studied how to improve the model's generalization performance through KD. They set the teacher model and the student model to the same structure. Afterward, the student model built through KD was set as a new teacher model and KD was performed again to increase the generalization performance of the model. Park *et al*. (2019) proposed relational KD (RKD), which additionally extracts relational information of embedded vectors from data during the training process. They showed that RKD, which adds structural relationships between embedded vectors from data points, outperforms conventional KD.

When a well-performing teacher model is successfully compressed, computers or small devices such as smartphones can learn the features of the data with less computational cost and make fine decisions just as well as before compression. Since the main idea of RKD lies in reducing the structural differences in the vectors between the teacher model and the student model, there will be further improvements if we convey more structural information about the vectors. Perhaps there is a limit to extracting more structural information when comparing a pair of vectors. However, when converting the insight of a pair of vectors to an area-wise approach, we can consider the structural information in the form of a triangle by additionally considering zero vector.

In this paper, we propose area-wise RKD as an extension of RKD. After generating the embedding vectors through the model, we can extract relational knowledge through a function that measures the structural differences between the vectors. The proposed structural relationship information of embedding vectors is the area of a triangle composed of two embedding vectors and a zero vector. A classification model is constructed to reduce the difference in relational knowledge obtained from the teacher model and the student model. By adding the area-wise relational information to the distillation, we achieve improvement in model compression and generalization, two main goals of KD.

We apply the proposed method to build classification models using image and audio data to compare the performance of the models with that of the existing methods. In the experiment using audio data, we compare the proposed method with existing ones in terms of model compression. In the ex-

periment using image data, we set teacher and student models to have an identical structure to check that the proposed area-wise RKD can increase the generalization performance more than the existing methods.

Overall, the main contributions of this paper are:

1) We propose area-wise RKD to convey additional structural relational information between vectors when training the model. We achieve improvements in KD for the student model by using the proposed method combined with the existing RKD.

2) We propose a loss function that can adjust the ratio of the relational information between the teacher model and the student model.

3) We provide extensive experiments comparing the performance of different methods for model compression and generalization on both image and audio data.

The rest of this paper is organized as follows. Section 2 reviews some related works in detail. Section 3 presents the proposed method with a new loss function. Section 4 provides the details of the datasets, the preprocessing process, the models used as teacher and student models, and the experimental results. Finally, in Section 5, we draw conclusions while discussing future research directions.

2. Related work

2.1. Knowledge distillation

In KD, we first train a large and complex deep neural network model, and then when training a small model, we pass the hidden or output layer values of the teacher model to a relatively small model and set those values as the target values. For example, a large and complex model is a single large model with strong regulation, such as an ensemble of multiple trained models or a model with dropout methods. By training large and complex models, we can build models more efficiently for deployment by filtering out redundant information from training data and transferring information to smaller models (Hinton *et al*., 2015).

In general, the loss function used to train a model reflects as closely as possible whether the data belong to true classes. But during the training process, we also need to consider how good the predictive performance of the model is, i.e., how well it generalizes to make predictions to new data. This requires information about whether the generalization process is going in the right way, but usually, this kind of information is difficult to obtain. However, in the KD process, since the classification probability can be estimated using the value obtained during the training of a large and complex model using data, the generalization method can be transferred to a smaller model using the value (Hinton *et al*., 2015).

Hinton *et al*. (2015) proposed a model compression method using logit values, which are pre-activated values in the output layer. They proposed dividing the logit value by the temperature, which is a real number greater than 1, and applying the softmax function to obtain the soft target. They set the temperature values so that the distribution of values in the output layer of the teacher model is properly flattened. The same temperature is applied to the logit values from the student model and the distillation is performed by reducing the soft target difference between the teacher model and the student model. The following is how the soft targets are obtained:

$f(x,\theta_1^*)$	y	
0.01	0	1
0.01	0	2
0.01	0	3
0.02	0	4
0.8	**1**	**5**
0.1	0	**6**
0.01	0	7
0.01	0	8
0.01	0	9
0.02	0	0

Figure 2: *MNIST example.*

$$p_i = \frac{\exp(z_i/T)}{\sum_{j=1}^{n} \exp(z_j/T)}, \quad i = 1, 2, \ldots, n, \tag{2.1}$$

where z_i is the logit, and T is the temperature. If T is set to 1, p_i is equal to the value passed through the softmax function.

Figure 2 shows an example of training results on images labeled 5 on the MNIST data set (LeCun *et al*., 1998). A neural network model denoted by is trained using the image data and the output layer values. In this case, the probability of classifying the image as 6 is 0.1, which is relatively higher than the rest except for 5. This value provides important information that the image belongs to class 5 but is also similar to class 6. However, this kind of information may become obsolete when using loss functions such as cross-entropy. Thus, by introducing temperature to obtain a soft target, we can obtain a new flattened class distribution of the data, so the values can convey rich information about the class.

2.2. Born again neural networks

Furlanello *et al*. (2018) proposed born again neural networks (BANs). Rather than viewing KD as a methodology for model compression, they showed that it is possible to improve the generalization performance. In BAN, teacher and student models are set up to have the same structure. At the same time, the classification probability of the teacher model is used as the target value of the student model. Once the student model is trained, the trained model becomes a new teacher model for the next KD process. They showed that student models obtained over multiple generations can improve the performance of model generalization and outperform teacher models. In the image classification task, the goal is to find a neural network function $f : X \to Y$ that generalizes well to unseen data when

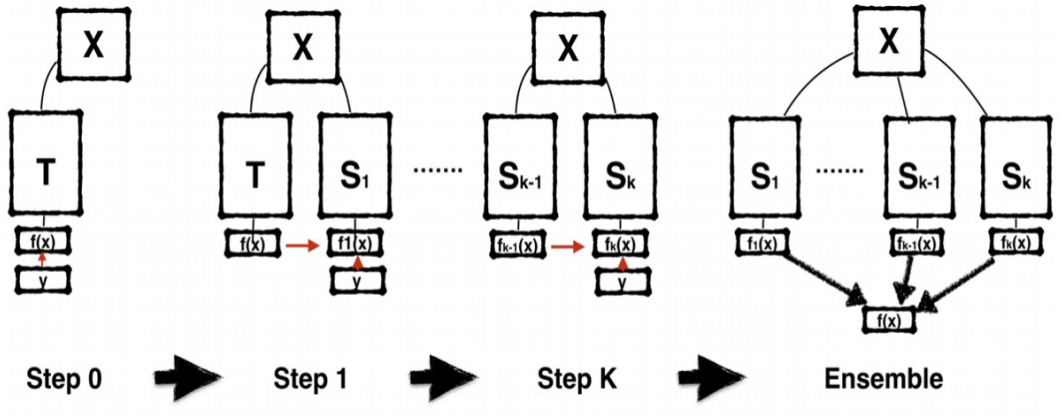

Figure 3: *The structure of born-again neural network (Furlanello et al., 2018).*

the inputs and labels of the training data are $(x, y)\ \epsilon\ X \times Y$ (Furlanello *et al.*, 2018). In general, we try to get $\theta^*$by minimizing the loss function L for the weight parameter θ in some space as follows:

$$\theta^* = \text{argmin}_\theta\, L\,(y, f(x, \theta))\,. \tag{2.2}$$

In BAN, the student model is trained using the values of the output layer of the teacher model, and the output layer of the student model seeks to obtain optimized parameters. Experiments have shown that student models obtained over multiple generations can improve the performance of model generalization and outperform teacher models. Moreover, a better model was obtained when the average ensemble was applied to the student model.

Figure 3 shows the structure of a BAN, where X is the training data, T and S are the teacher model and the student model, respectively, and $f(x)$ and y are the output layer value and the target value, respectively. According to the algorithm of BAN, $S_{(t-1)}$ becomes a teacher model of S_t and this process can be performed k times. During the training process, we can optionally introduce the loss function as follows:

$$\theta_i^* = \text{argmin}_{\theta_i}\, \left[L(y, f(x, \theta_i)) + L\left(f(x, \theta_{i-1}^*), f(x, \theta_i)\right)\right]. \tag{2.3}$$

The final ensemble model is based on the average of the probability values for each student model to classify the data:

$$f^k(x) = \sum_{i=1}^{k} \frac{f(x, \theta_i^*)}{k}, \tag{2.4}$$

where $f^k(x)$ is the average of the values of the output layer of the student models after training k generation, and θ_i^* is the optimized weight parameter after the training process of the i^{th} student model.

2.3. Relational knowledge distillation

Park *et al.* (2019) proposed RKD, a method to distill relational information of values embedded by a model. In order to extract relational information, specific hidden layers are selected from the teacher

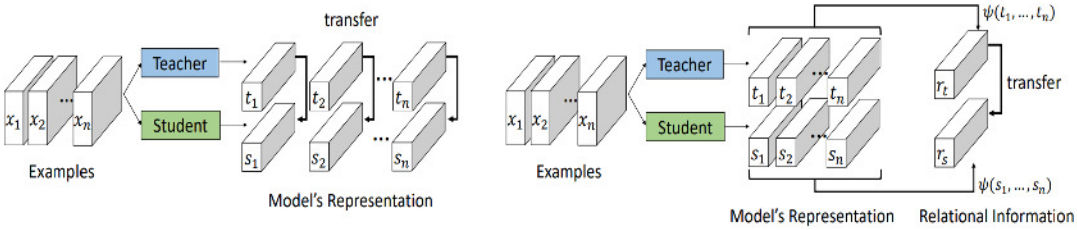

Figure 4: *Conventional knowledge distillation (left) and relational knowledge distillation (right) (Park et al., 2019).*

model and the student model, and the structural information of the values of the hidden layer is defined as a specific function. Afterward, they proposed a method to further utilize the training method in a way to reduce the structural difference between the teacher model and the student model. Then, they showed that RKD can improve the performance of models in the fields of metric learning, image classification, and few-shot learning.

Figure 4 shows how the conventional KD and RKD techniques are applied, where t_i and s_i are embedding vectors obtained after the data x_i passes through the teacher model and the student model, respectively. t_i and s_i are dependent because they are outputs of input x_i. In the conventional method, the knowledge generated from a single data point is transferred individually from the teacher model to the student model. However, in RKD, the knowledge containing relational information generated from more than a single data point is transferred. The two functions of RKD measuring relational information are defined:

$$\psi_{Distance}(t_i, t_j) = \frac{1}{\mu} \left\| t_i - t_j \right\|_2, \tag{2.5}$$

$$\psi_{Angle}(t_i, t_j, t_k) = \cos \angle t_i t_j t_k, \tag{2.6}$$

where $\psi_{Distance}$ measures the Euclidean distances between two vectors and μ represents the normalization factor. And ψ_{Angle} measures the angle formed by the three vectors obtained from the model output.

After obtaining relational information from the teacher and student models, the student model is trained to reduce the information gap using Huber loss:

$$\delta(x, y) = \begin{cases} \frac{1}{2}(x-y)^2 & \text{for } |x-y| \le \delta, \\ \delta |x-y| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases} \tag{2.7}$$

Huber loss is commonly used for robust regression, and the smaller the δ is, the more robust the model is against outliers. Typically, δ is set between 1 and 2 in robust regression.

Distance-wise distillation loss L_{RKD-D} and Angle-wise distillation loss L_{RKD-A} are now defined as:

$$L_{RKD-D} = \sum_{(x_i, x_j) \in \chi^2} l_\delta(\psi_{Distance}(t_i, t_j),\ \psi_{Distance}(s_i, s_j)), \tag{2.8}$$

$$L_{RKD-A} = \sum_{(x_i, x_j, x_k) \in \chi^2} l_\delta(\psi_{Angle}(t_i, t_j, t_k),\ \psi_{Angle}(s_i, s_j, s_k)), \tag{2.9}$$

where l_δ is the Huber loss and δ is set to 1 the same as in RKD.

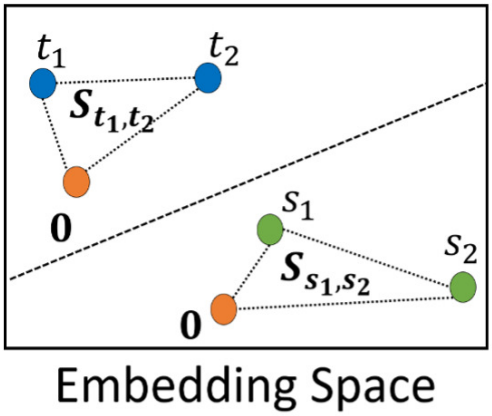

Figure 5: *Area formed by two vectors in embedding space.*

3. Proposed methodology

RKD has been proposed as a new method to extract structural relationship information for output. However, since the distance and angle information are converted through a loss function, we believe that additional information is needed to preserve structural relationships.

In this paper, we propose area-wise RKD as an extension of RKD. The loss function of the area-wise RKD is defined as:

$$L_{Area} = \sum_{(x_i,x_j)\in\chi^2} l_\delta(\psi_{Area}(t_i,t_j),\ \psi_{Area}(s_i,s_j)), \tag{3.1}$$

where t_i and s_i are the embedding vectors obtained from teacher and student models, respectively. These vectors can be set within the range of $(x_i, x_j)\epsilon\chi^2$ by creating pairs of input data during training. The function l_δ is to calculate the difference between the relational information of the teacher model and the student model. We use the Huber loss function with $\delta = 1$. The function ψ_{Area} defines the relational information and is used to calculate the area of the triangle formed by the two embedding vectors and the zero vector after the data is fed into the model as follows:

$$\psi_{Area}(t_i,t_j) = \frac{1}{2\mu_t}\left|t_i \| t_j\right| \sin\theta = \frac{1}{\mu_t} S_{t_i,t_j}, \tag{3.2}$$

where θ is the angle between the two vectors t_i and t_j, and μ_t is the average of all area values obtained during training and is defined as:

$$\mu_t = \frac{1}{\left|\chi^2\right|} \sum_{(x_i,x_j)\in\chi^2} S_{t_i,t_j}. \tag{3.3}$$

Then, training is conducted by reducing the size difference of each triangle obtained from the teacher model and the student model. Figure 5 shows an illustration in which the embedding vectors obtained after the training data has passed through the teacher model and the student model form a triangle in the embedding space. In conventional RKD, structural differences in embedding values are learned by reducing distance differences or angular differences, respectively. By adding the area differences to them, it is expected to provide richer information on the structural differences between the teacher model and the student model and build a better classification model.

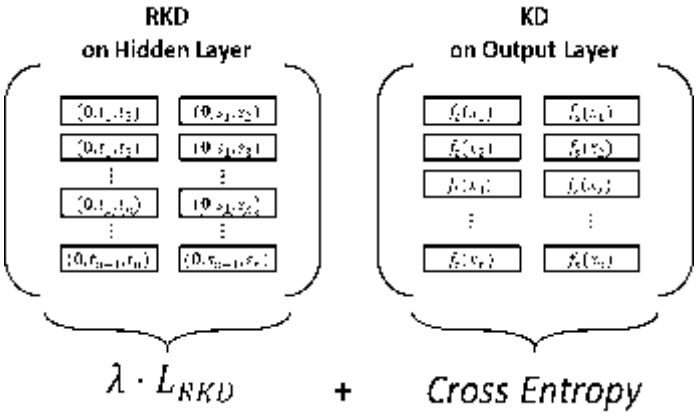

Figure 6: *Structure of the loss function for the student model using RKD.*

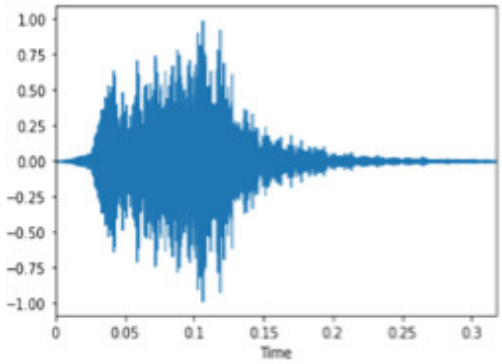

Figure 7: *Time-amplitude graph of dog bark.*

To construct the student model, the output layer values are used to convey knowledge about the probabilities of the classes using cross-entropy as the loss function. Relational information is additionally conveyed using the embedding value of the flattened layer before the output layer. Relational information can be scaled by multiplying λ. Finally, the loss function is defined as the sum of the loss function for RKD and the loss function for class probabilities and can be used in combination with distance, angle, and area-wise RKD methods:

$$L_{Total} = \sum_{i}^{n} \text{CrossEntropy}\left(f_t(x_i), f_s(x_i)\right) + \lambda_{RKD} \times \sum_{(x_i, x_j, \ldots, x_n) \in \chi^n} l_\delta\left(\psi(t_1, t_2, \ldots, t_n), \psi(s_1, s_2, \ldots, s_n)\right),$$

where $f_t(x_i)$ and $f_s(x_i)$ are obtained from the output layer of the teacher and student models, respectively. Figure 6 shows the structure of the loss function for the student model.

Table 1: Class information of UrbanSound8

Class ID	Sound Class
0	air conditioner
1	car horn
2	children playing
3	dog bark
4	drilling
5	engine idling
6	gun shot
7	jackhammer
8	siren
9	street music

Table 2: The number of samples per fold

Fold	# of Samples
1	873
2	888
3	925
4	990
5	936
6	823
7	838
8	806
9	816
10	837

4. Experimental study

4.1. Datasets

4.1.1. UrbanSound8K

UrbanSound8K is a dataset of 8,732 sounds collected from cities (Salamon *et al*., 2014). Each data is less than 4 seconds long and the data set is divided into 10 classes. Figure 7 shows data corresponding to a dog barking as a time-amplitude graph. This data is divided into 10 sets for cross-validation, and detailed descriptions are given in Tables 1 and 2.

4.1.2. CIFAR-100

CIFAR-100 is essentially the same as CIFAR-10 with 60,000 color images. However, CIFAR-100 is again divided into 100 classes. Each class has 600 images, and the data for each class is divided into 500 training data and 100 test data. Figure 8 shows some images of different classes.

4.2. Feature extraction from the unstructured data

In this study, image data are used for experiments to improve the generalization performance of the model and audio data are used for model compression experiments.

4.2.1. Feature extraction from the audio data

Audio recognition tasks require extracting valid features from an input signal. Acoustic characteristics of audio can be obtained using the Python package LibROSA (McFee *et al*., 2015). Various

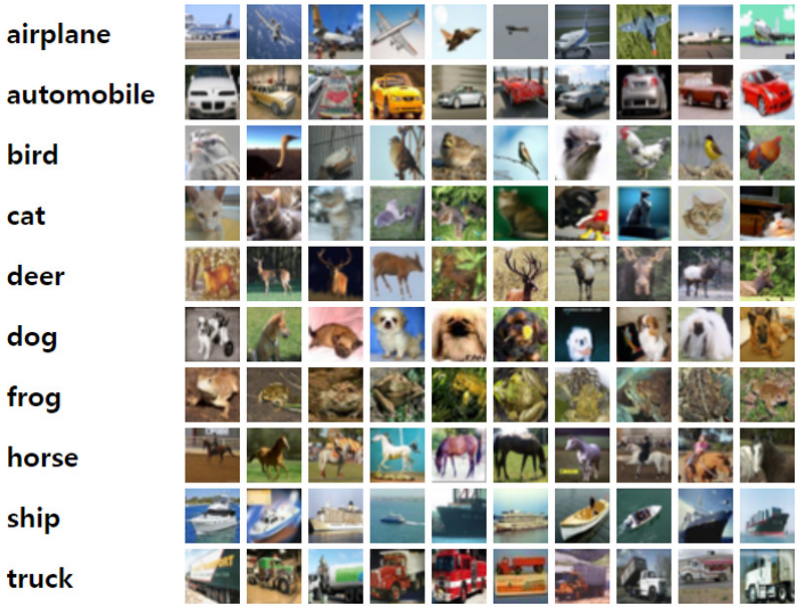

Figure 8: *CIFAR-10 data set (Krizhevsky and Hinton, 2009).*

features can be extracted, such as mel-frequency cepstral coefficients (MFCC) (Logan, 2000), Mel-spectrogram, Chromagram (Müller and Ewert, 2011), Constant-Q chromagram, and Chroma energy normalized statistics (CENS) (Shepard, 1964).

A spectrogram is a method to visualize an acoustic signal. The sound signal is divided into several sections and the values obtained through FFT are converted into frequency and amplitude values over time. Chromagram is a method for visualizing pitch and is widely used to analyze music. It has the advantage of being able to capture melody and harmony characteristics from sound data. After applying the local Fourier transform of the sound signal, a Constant-Q chromagram or CENS can be obtained by the further conversion process. Figure 9 shows an example of audio features extracted using the above method.

The feature values extracted from the audio are stacked in multiple layers and used as input to a convolutional neural network (CNN) (Piczak, 2015). Figure 10 shows the structure of the CNN model. For audio data less than 4 seconds in length, zero padding is applied to match the input length of the data. Data augmentation is done by adding Gaussian noise before extracting acoustic features to double the amount of data. Figure 11 shows an example of data augmentation seen in a time-amplitude graph before and after adding Gaussian noise.

4.2.2. Feature extraction from image data

In this paper, we use a vanilla CNN and residual neural network (ResNet) model (He *et al*., 2016). CNNs are used to extract features from images and are effective in classifying, predicting, and recognizing images. CNNs extract local information from data through filters. In addition, down-sampling is performed through max pooling for the part showing a large signal among the extracted information. The features of the data are learned through a series of convolutional and max pooling layers and then used in combination with specific output layers corresponding to the objectives of the model

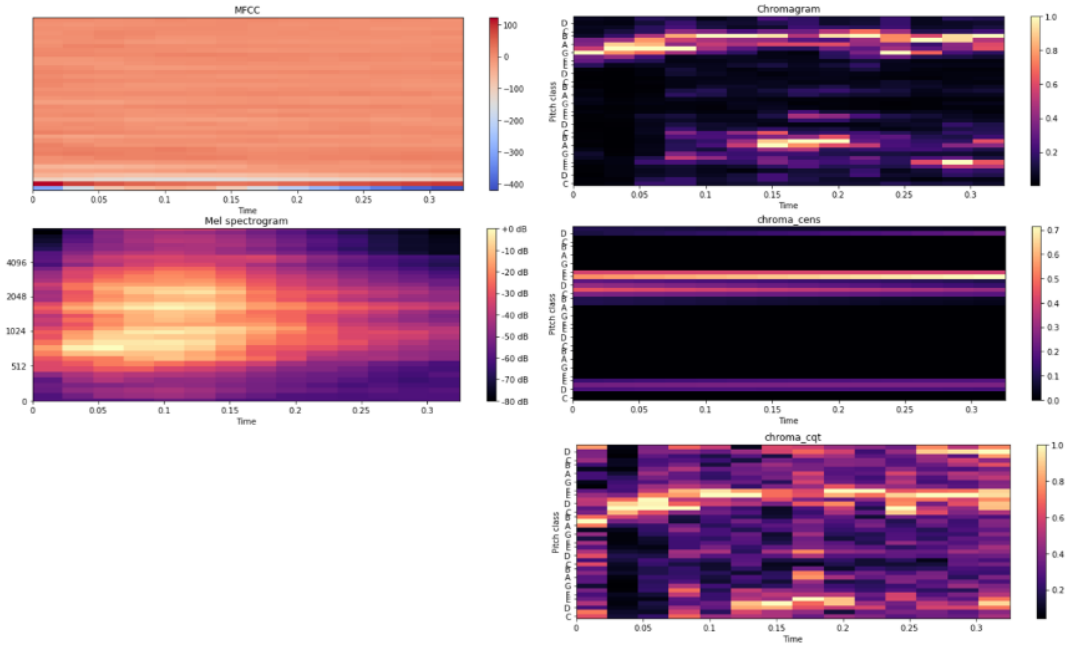

Figure 9: *Audio features extracted from dog bark (clockwise from top right, chormagram, constant-Q chromagram, chroma energy normalized, mel-spectrgram, MFCC).*

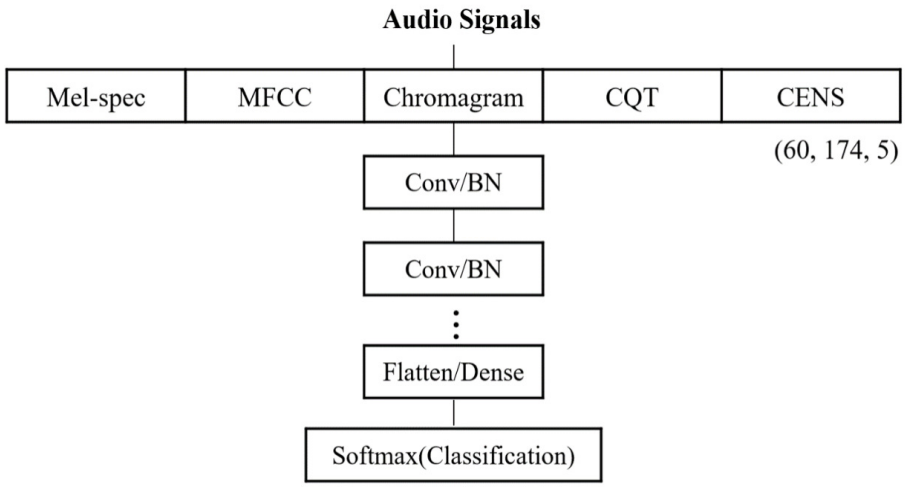

Figure 10: *The structure of a CNN model to use the stacked feature of audio as input.*

(e.g., classification and object detection). Convolution reduces the amount of model computation, and max pooling enables learning of invariant transformations such as position, size, and rotation of data (LeCun *et al*., 1998).

ResNet introduces skip connections between hidden layers. If we assume that $H(X)$ is an output

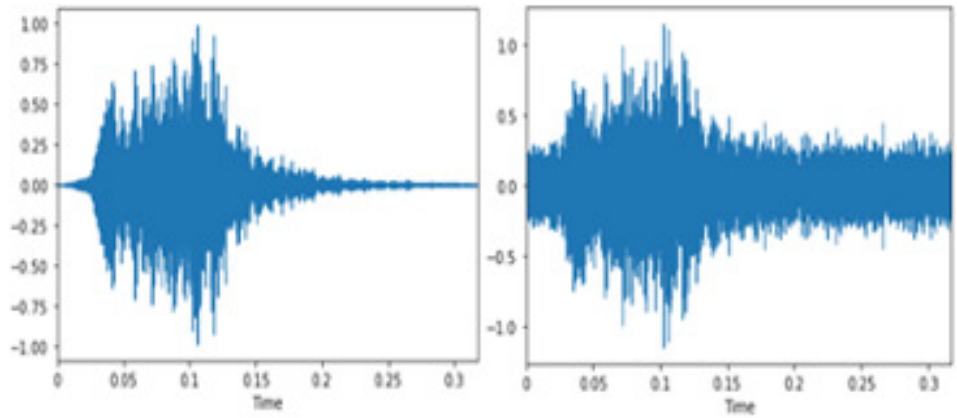

Figure 11: *Time-amplitude graph of dog bark before and after adding Gaussian noise.*

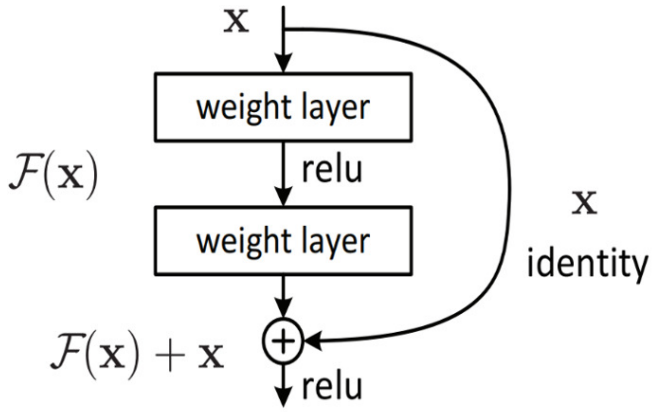

Figure 12: *The structure of shortcut in ResNet (He et al., 2016).*

value obtained through two hidden layers in a CNN, then the input is a value that has been multiplied by weights and passed through an activation function. ResNet, on the other hand, assumes that $H(X)$ is a value that passes through the two hidden layers but is not activated via an activation function. If you set $H(X) = F(X) + X$, we can learn $F(X)$, which is the core of ResNet. This allows us to introduce a much deeper model, as the value of X is preserved even if $F(X)$ is poorly trained during training, as shown in Figure 12. That is, it has the advantage of preventing gradient vanishing problems (He *et al.*, 2016).

4.3. Experimental procedure and composition

4.3.1. Experimental procedure

In the experiment using CIFAR-100, both the teacher model and the student model are set to ResNet-56 to classify images. After training the teacher model, the softmax function value of the output layer is set as the target value of the student model to perform RKD.

In the experiment with UrbanSound8K, the teacher model is set to ResNet-18 and the student model uses a vanilla CNN with 2 convolutional layers. RKD is further applied using the flattened

Table 3: Results of model compression experiments using UrbanSound8K

Model	Accuracy(%)
ResNet-18 (teacher)	68.16
2 Conv CNN (baseline student)	63.70
CNN+angle (50)	64.46
CNN+distance+angle (40, 10)	65.12
CNN+distance+area (20, 20)	65.13
CNN+angle+area (40, 20)	65.28
CNN+distance+angle+area (40, 20, 40)	64.27
CNN+Hinton (T=1.5)	64.37
CNN+Hinton+angle+area (10, 50)	65.53

The values in parentheses refer to weights of loss functions for RKD.

layer before the output layer. We also introduce Hinton KD to the output layer together with RKD to see if the model compression performance is improved.

For CIFAR-100, we randomly split the 50,000 training data into 45,000 training data and 5,000 validation data, and use 10,000 test data to measure test accuracy. And for UrbanSound8K, we average the validation accuracies after performing 10-fold cross-validation.

4.3.2. Experimental configuration

In the experiment using CIFAR-100, Nesterov momentum is used as an optimizer (Nesterov, 1983). The batch size is 256 and the weight decay is 0.0001. The initial learning rate is set to 0.001, and for the 60^{th}, 120^{th}, and 160^{th} epochs, the learning rate is multiplied by 0.1. The delta of the Huber loss function is set to 1. In the experiment using Hinton KD, the temperature is set to 1.5.

In the experiment using UrbanSound8K, Nesterov momentum is used as the optimizer, the batch size is set to 8, and the weight decay is 0.0001. The initial learning rate is set to 0.1, and the learning rate is multiplied by 0.1 every 10 epochs. The delta of the Huber loss function is set to 1.

All models are constructed using Tensorflow v.1.14 (Abadi *et al*., 2016).

4.4. Results

4.4.1. UrbanSound8K

In the experiment using audio data, KD was applied to compress the model. The results are summarized in Table 3. The teacher model is ResNet-18, the student model is a CNN with 2 convolutional layers, and it shows that the teacher model with 18 hidden layers has been successfully compressed into a student model with 2 convolutional layers. The number of weight parameters used in the teacher model is 178,466 and the number of parameters used in the student model is 116,538. That is, model compression through KD reduced the number of weight parameters by about 35%. We built a classification model using ResNet-18 and achieved a 10-fold cross-validation accuracy of 68.16%. In addition, when ResNet-18 was used as the teacher model and a CNN with two convolutional layers was used as the student model, the student model was able to achieve 63.70% accuracy when compressing the model using the conventional KD. We introduced several combination methods of RKD. First, when only angle-wise RKD was used, the accuracy was 64.46%, which was about 0.7% better than the baseline model. When the two methods of RKD were used together, the accuracy of the classification model was higher than that when only one method was used. The accuracies of the model using distance and angle, distance and area, and angle and area were 65.12%, 65.13%, and 65.28%, respectively. When all three methods (distance, angle, and area) were applied together, the accuracy improved to 64.27%, but it was no better than applying only the relational knowledge of angle and

Table 4: Results of model compression experiments using CIFAR-100

Model	Accuracy (%)
ResNet-56	59.72
BAN ensemble (baseline)	64.30
BAN+distance ensemble	64.44
BAN+angle ensemble	65.36
BAN+area ensemble	64.28
BAN+distance+angle ensemble	64.32
BAN+distance+area ensemble	63.86
BAN+angle+area ensemble	64.82
BAN+distance+angle+area ensemble	64.50

area together. Additional experiments were also conducted to compare the model compression performance when RKD was applied with Hinton KD. When the temperature was set to 1.5, the accuracy of the student model was 64.37%. In addition, when angle-wise and area-wise RKD were applied together with Hinton KD, the accuracy was the highest at 65.53%.

4.4.2. CIFAR-100

The experimental results using CIFAR-100 are shown in Table 4. ResNet-56 showed an accuracy of 59.72%. After performing BAN with 3 generations, the ensemble of 3 student models has an accuracy of 64.30%, which is much higher than that of ResNet-56. Several RKDs were introduced in the BAN to improve the performance of the model. At this time, all weights of the loss function related to RKD were selected as 20.

We first considered one RKD for model training and then examined the performance of the averaged ensemble model. When each of RKDs based on distance, angle, and area was applied, the accuracies of the ensemble models were 64.44%, 65.36%, and 64.28%, respectively. Therefore, in this case, the model using angle-wise RKD improved the accuracy by more than 1% and was the best among the three methods.

When two types of RKD were applied together, the accuracies of the model learned using RKD based on distance and angle, distance and area, and angle and area were 64.32, 63.86, and 64.82%, respectively. Finally, when all three RKDs were applied simultaneously, the accuracy was 64.50%.

The results show that the best performance improvement was obtained using angle-wise RKD and sub-optimal performance was obtained using both angle-wise and area-wise RKD together.

5. Conclusions

In this study, we proposed area-wise RKD with a new loss function. This method can reduce the difference in relational information between the teacher model and the student model by obtaining the area information generated by embedding vectors of the data. In addition to the existing distance- and angle-wise RKDs, it is expected to provide richer information on the structural differences between teacher and student models and further improve the performance of the student model.

Existing methods have tried to improve the performance of the student model by transferring information about the distance between embedding vectors or the angle formed by the embedding vectors from the teacher model to the student model. On the other hand, the proposed method can be viewed as considering the information that includes the distance and angle considered in the existing methods at once by trying to convey the information about the area of the triangle created by the embedding vectors. For example, even if the distance between the embedding vectors is the same, it

can be seen that a small angle and a large angle between the vectors represent different information.

Conversely, even if the angle between embedding vectors is the same, a short distance and a long distance between vectors represent different information. However, considering the area of a triangle containing both distance information and angle information, the effect of integrating each piece of information to be delivered through the two existing methods can be expected.

To verify the performance of area-wise RKD, we conducted two experiments using audio and image data. In the experiment using audio data, model compression was performed using the proposed method along with the existing methods. The results show that the performance of model compression is best when both area- and angle-wise RKDs are applied together.

In the experiment using image data, we considered BAN to improve the generalization performance of the model by introducing various RKDs. As a result, the performance improvement was the best when angle-wise RKD was applied, and sub-optimal accuracy was shown in the experiment when area-wise and angle-wise RKD were used together.

For future research, we plan to apply the proposed method to data in other domains, such as natural language, video, and time series data. In addition, we could consider applying the proposed method to RNN-based models, Seq2Seq models, and attention-based models.

6. Acknowledgements

This research was supported by the Chung-Ang University research grant in 2020. This research was also supported by Next-Generation Information Computing Development Program through the National Research Foundation (NRF) of Korea and the NRF grant funded by the Ministry of Science, ICT (NRF-2017M3C4A7083281, NRF-2021R1F1A1056516).

References

Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, and Kudlur M (2016). Tensorflow: A system for large-scale machine learning. In *Proceedings of 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, Savannah, GA, USA, 265–283.

Buciluǎ C, Caruana R, and Niculescu-Mizil A (2006). Model compression. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 535–541.

Deng J, Dong W, Socher R, Li LJ, Li K, and Fei-Fei L (2009). Imagenet: A large-scale hierarchical image database. In *Proceedigns of 2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, USA, 248–255.

Edunov S, Ott M, Auli M, and Grangier D (2018). Understanding back-translation at scale, Available from: arXiv preprint arXiv:1808.09381

Furlanello T, Lipton ZC, Tschannen M, Itti L, and Anandkumar A (2018). Born again neural networks, Available from: arXiv preprint arXiv:1805.04770

Han S, Mao H, and Dally WJ (2015). Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, In *Proceedings of 4th International Conference on Learning Representations (ICLR 2016)*, San Juan, Puerto Rico, Available from: arXiv preprint arXiv:1510.00149

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*, Las Vegas, NV, USA, 770–778.

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

Krizhevsky A and Hinton G (2009). Learning multiple layers of features from tiny images (Technical report), University of Toronto, **1**, 7, Available from: https://www.cs.toronto.edu/ kriz/learning-features-2009-TR.pdf

LeCun Y, Bottou L, Bengio Y, and Haffner P (1998). Gradient-based learning applied to document recognition, *Proceedings of the IEEE*, **86**, 2278–2324.

Logan B (2000). Mel frequency cepstral coefficients for music modeling, *ISMIR*, **270**, 1–11.

McFee B, Raffel C, Liang D, Ellis D, McVicar M, Battenberg E, and Nieto O (2015). Librosa: Audio and Music Signal Analysis in Python. In *Proceedings of the 14th Python in Science Conference*, Austin, Texas, USA, 18–24.

Müller M and Ewert S (2011) Chroma toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *Proceedings of the International Conference on Music Information Retrieval*, Miami, Florida, USA, 215–220.

Nesterov YE (1983). A method for solving the convex programming problem with convergence rate $O(1/k^2)$, *Doklady Akademii Nauk SSSR*, **269**, 543–547.

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*, Long Beach, CA, USA, 3967–3976.

Piczak KJ (2015). Environmental sound classification with convolutional neural networks. In *Proceedings of 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing*, Boston, MA, USA, 1–6.

Salamon J, Jacoby C, and Bello JP (2014). A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando, Florida, USA, 1041–1044.

Saon G, Kurata G, Sercu T *et al.* (2017). English conversational telephone speech recognition by humans and machines, In *Proceedings of 18th Annual Conference of the International Speech Communication Association*, Stockholm, Sweden, 132–136, Available from: arXiv preprint arXiv:1703.02136

Shepard R (1964). Circularity in judgments of relative pitch, *The Journal of the Acoustical Society of America*, **36**, 2346–2353.

Tan M and Le QV (2019). EfficientNet: Rethinking model scaling for convolutional neural networks, In *Proceedings of 36th International Conference on Machine Learning*, Long Beach, CA, USA, 6105–6114, Avalilable from: arXiv preprint arXiv:1905.11946

Zhang Z, Xing F, Su H, Shi X, and Yang L (2017). Recent advances in the applications of convolutional neural networks to medical image contour detection, Available from: https://doi.org/10.48550/arXiv.1708.07281

Received March 22, 2023; Revised April 5, 2023; Accepted April 10, 2023