KNOWLEDGE DISTILLATION WITH SPECIALIST MODELS IN ACOUSTIC SCENE CLASSIFICATION

Technical Report

Jee-weon Jung[*], *Hee-Soo Heo*[*], *Hye-jin Shim, and Ha-Jin Yu*[†]

School of Computer Science, University of Seoul, South Korea

ABSTRACT

In this technical report, we describe our submission for the Detection and Classification of Acoustic Scenes and Events 2019 task1-a competition which exploits knowledge distillation with specialist models. Different acoustic scenes that share common properties are one of the main obstacles that hinder successful acoustic scene classification. We found that confusion between scenes, sharing the common properties, causes most of the errors in the acoustic scene classification. For example, the confusing scene pairs such as airport-shopping mall and metro-tram have caused the most errors in various systems. We applied knowledge distillation based on the specialist models to address the errors from the most confusing scene pairs. Specialist models where each model concentrates on discriminating a pair two similar scenes are exploited to provide soft-labels. We expected that knowledge distillation from multiple specialist models and a pre-trained generalist model to a single model could train an ensemble of models that gives more emphasis on discriminating specific acoustic scene pairs. Through knowledge distillation from well trained model and specialist models to single model, we report improved accuracy on the validation set.

Index Terms— Acoustic scene classification, Specialist models, Knowledge distillation, Teacher-student learning, Deep neural networks

1. SYSTEM DESCRIPTION

In this technical report, we describe out submission for the Detection and Classification of Acoustic Scenes and Events (DCASE) 2019 competition task 1-a [1, 2]. Our submission exploits a score-level ensemble where one uses a convolutional neural network (CNN) that inputs raw waveforms and the other one uses a CNN that inputs log Mel-energy features. Details regarding the idea, hypothesis and other academical points of view will be dealt in the workshop paper which we will submit for DCASE 2019 workshop.

2. FEATURE EXTRACTION

For the raw waveform model, we apply pre-emphasis and no other pre-processing methods such as utterance or global mean and standard deviation normalization. We use raw waveform from both channels (left and right, stereo) without modification, making the shape of our input to the CNN as ($\#sample$, 2) where $\#sample$ would be 480,000 when using the whole audio segment. At training phase we randomly crop about 3 s from each audio segment for data augmentation effect (because different part is cropped from the identical audio segment every epoch) and boosting training speed. At evaluation phase, all 10 s segment is input. This procedure follows that of [2].

For the log Mel-energy feature model, we extract an 256 dimensional log Mel-energy features with an window length of 100 ms and shift size of 40 ms. No delta nor delta-delta coefficients are used. Table 1 shows the details of Mel-energy feature extraction.

Table 1: Hyper-parameters for Mel-energy feature extraction.

Hyper-parameter	
frame length	100ms
shift size	40ms
number of FFT bins	4800
number of filters	256
pre-emphasis	with 0.97 coefficient
normalization	global mean & variance normalization

Table 2: DNN architecture of raw waveform model with input sequence shape: (479999×2). At training phase, input sequence shape is (239999×2) where 239999 samples are randomly selected.

Layer	Output shape	Kernel size	Stride
Conv1	39999×64	12	12
Res1	13333×64	3	1
Res2	4444×128	3	1
Res3	1481×128	3	1
Res4	493×128	3	1
Res5	164×128	3	1
Res6	54×128	3	1
Res7	18×128	3	1
GlobalPool	128	-	-
Dense1	64	128×64	-
Output	10	64×10	-

3. NETWORK ARCHITECTURE

The raw waveform and Mel-energy models use CNN architectures with residual connections and pooling layers [3, 4], which is identical to that used in [5]. Table 2 and 3 describe overall architectures of raw waveform and Mel-energy model, respectively.

[*] Equal contribution.

[†] Corresponding author.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT & Future Planning(2017R1A2B4011609)

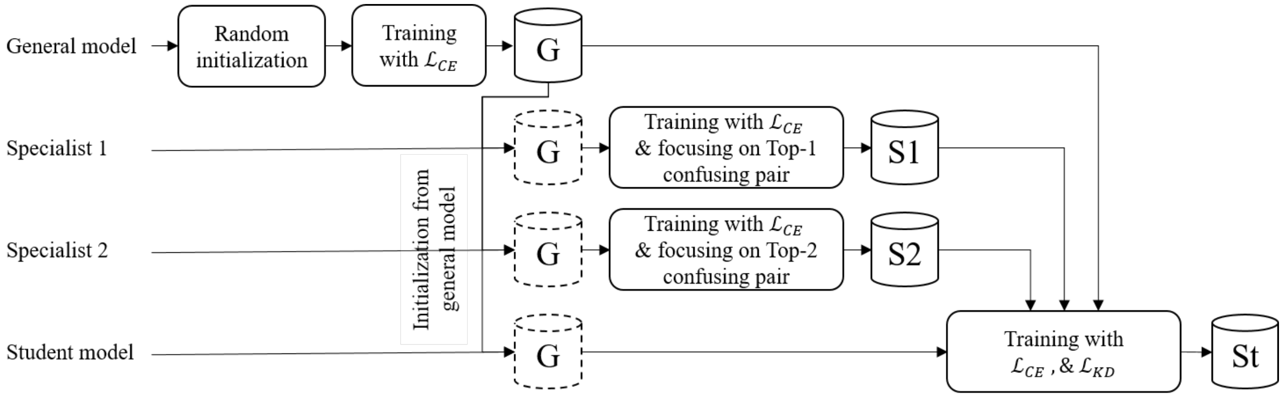
General model
Random initialization
Training with $\mathcal{L}_{CE}$
G
Specialist 1
G
Training with $\mathcal{L}_{CE}$ & focusing on Top-1 confusing pair
S1
Specialist 2
G
Training with $\mathcal{L}_{CE}$ & focusing on Top-2 confusing pair
S2
Initialization from general model
Student model
G
Training with $\mathcal{L}_{CE}$, & $\mathcal{L}_{KD}$
St

Figure 1: Workflow of the training procedure.

Table 3: CNN architecture for Mel-energy model (l: length of input sequence).

Layer	Output shape	Kernel size	Stride
Conv1	$l \times 252 \times 30$	7×7	1×1
Res1	$l \times 252 \times 30$	3×3	1×1
Res2	$(l/2) \times 126 \times 60$	3×3	2×2
Res3	$(l/4) \times 63 \times 120$	3×3	2×2
Res4	$(l/12) \times 21 \times 240$	3×3	3×3
AvgPool	240	Global	Global
Output	10	240×10	-

Table 4: Common hyper-parameters for training raw waveform model (lr_t: learning rate at the $t'th$ iteration).

Hyper-parameter	
max epoch	70
batch size	24
weight of L2 regularization	0.001
data augmentation	mixup ($\alpha = 0.1$) [7]
optimizer	AMSGrad [8]
initial learning rate	0.0001
beta1, beta2	0.9,0.999
learning rate scheduling	multiply 0.2 at 20, 50^{th} epoch

4. TRAINING PROCEDURE

Our submission comprises a three stage training phase: first we train the generalist model, second we train specialist models where each specialist model further concentrates on discriminating a pair of most confusing scenes depending on the confusion matrix of the generalist model, and the last phase of knowledge distillation (KD) training from generalist and two specialist models to a single model. Note that terms ‘generalist’ and ‘specialist’ refer to those from Hinton *et al.*’s paper that describes knowledge distillation [6].

At the first phase, we train the generalist model which exploits categorical cross-entropy shown in the following equation,

$$\mathcal{L}_{CE}(\theta) = -\sum_{i=1}^{N} logP(y_i|\boldsymbol{x}_i;\theta), \tag{1}$$

where N is the number of training data samples, P is function (DNN) that maps input data to a posterior distribution using the softmax output, and $\boldsymbol{x}_i$ is the input, θ is parameter set of the model, and y_i denote the $i'th$ input data and the corresponding label, respectively. In our models, based on DNN, P is defined by applying softmax function to the output layer as follow:

$$P(i|\boldsymbol{x};\theta) = \frac{exp(z_i)}{\sum_{j} exp(z_j)}, \tag{2}$$

where $\boldsymbol{z}$ is the output of the output layer.

After training the generalist model, we calculate the confusion matrix and find the two most confusing pairs of acoustic scenes which two specialist models respectively concentrates to discriminate. For training specialist model, we follow the recipe introduced in Hinton *et al.*’s study where for each mini-batch construction, half are sampled from the target confusing pair, and the other half are sampled from other acoustic scenes. We initialize specialist models with trained generalist model’s weight parameters. Categorical cross-entropy is used as the loss function for specialist training.

After training the generalist and two specialist models, we conduct knowledge distillation from these three models to one model, which is also initialized using the weight parameters of the generalist model using the following equation 3:

$$\begin{aligned}&\mathcal{L}_{KD}(\theta;\theta_g,S)\\ &\quad= -\sum_{i=1}^{N}\sum_{i=j}^{M} logQ(j|\boldsymbol{x}_i;\theta)[Q(j|\boldsymbol{x}_i;\theta_g) + \sum_{\theta_s \in S} Q(j|\boldsymbol{x}_i;\theta_s)],\end{aligned} \tag{3}$$

$$Q(i|\boldsymbol{x};\theta) = \frac{exp(z_i/T)}{\sum_{j} exp(z_j/T)}, \tag{4}$$

where Q denotes the probability function with concept of temperature T [6], θ_g is parameter set of the generalist model, θ_s is parameter set of the specialist model, and S is the set of specialist models. The loss function $\mathcal{L}_{KD}$ has been proposed to train the single model that achieves an ensemble of models with different characteristics [6].

Table 5: Common hyper-parameters for training Mel-energy model (lr_t: learning rate at the $t'th$ iteration).

Hyper-parameter	
max epoch	200
batch size	50
weight of L2 regularization	0.0001
data augmentation	mixup ($\alpha = 0.1$) [7]
optimizer	Adam [9]
initial learning rate	0.001
beta1, beta2	0.9,0.999
learning rate scheduling	$lr_t = \frac{initial\_lr}{1+0.0001t}$

Table 6: Performances of various systems on fold-1 configuration in accuracies (%) (G: generalist model, S1: $1'th$ specialist model, S2: $2'nd$ specialist model, St: student model).

System	G	S1	S2	St
Raw waveform	73.71	74.89	74.53	75.81
Mel-energy	74.33	74.12	74.48	76.15

5. RESULTS

We evaluated the systems, described in this report, by following the fold-1 configuration of DCASE2019 task1-a. Table 6 shows the performances of various models according to input features and training methods.

6. REFERENCES

[1] http://dcase.community/challenge2019/.

[2] A. Mesaros, T. Heittola, and T. Virtanen, “A multi-device dataset for urban acoustic scene classification,” in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2018 Workshop (DCASE2018)*, November 2018, pp. 9–13. [Online]. Available: https://arxiv.org/abs/1807.09840

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] ——, “Identity mappings in deep residual networks,” in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[5] H.-S. Heo, J.-w. Jung, H.-j. Shim, and H.-J. Yu, “Acoustic scene classification using teacher-student learning with soft-labels,” *arXiv preprint arXiv:1904.10135*, 2019.

[6] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.

[7] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “mixup: Beyond empirical risk minimization,” *arXiv preprint arXiv:1710.09412*, 2017.

[8] S. J. Reddi, S. Kale, and S. Kumar, “On the convergence of adam and beyond,” *arXiv preprint arXiv:1904.09237*, 2019.

[9] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.