TRAINING METHOD USING CLASS-FRAME PSEUDO LABEL FOR WEAKLY LABELED DATASET ON DCASE2019

Technical Report

Sakiko Mishima, Yu Kiyokawa, Takahiro Toizumi, Kazutoshi Sagi, Reishi Kondo and Toshiyuki Nomura

Data Science Research Laboratories, NEC Corporation
1753 Shimonumabe, Nakahara-ku, Kawasaki 211-8666, Japan
{s-mishima@cb, y-kiyokawa@cq, t-toizumi@ct, ksagi@ah, kondoh@ct, t-nomura@da}.jp.nec.com

ABSTRACT

We propose a training method using class-frame pseudo label for weakly labeled datasets given by the IEEE AASP challenge on detection and classification of acoustic scenes and events 2019 (DCASE2019) task 4. Our model is constructed based on a residual network (ResNet) and trained by datasets including strong and weak labels. The strong label has event classes and their presences at each frame, and the weak label has only event classes. In order to train the model effectively, we propose class-frame pseudo labels for weakly labeled datasets. The class-frame pseudo label contributes to improvement of the event presence prediction at each frame by avoidance of overfitting to strongly labeled datasets. A result shows that F1-scores by our proposed method are 25.9% and 62.0% in the event-based and segment-based evaluations, respectively.

Index Terms— acoustic event detection, residual network, pseudo labeling, DCASE

1. INTRODUCTION

An audio event detection (AED) task using weakly labeled datasets has been proposed as a part of the detection and classification of acoustic scenes and events 2019 (DCASE2019) challenge [1]. The DCASE2019 task 4 evaluates AED systems which estimate the event classes and their presence at each frame. Datasets for the system development consists of strongly and weakly labeled data. A strong label has the event classes and the event presences at each frame, while a weak label has only the event classes. The DCASE2019 task 4 raises a question about the utilization of the datasets for a model training.

We propose a training method for weakly labeled datasets using class-frame pseudo labels. Our method firstly trains the model based on a residual network (ResNet) [2] using both strongly and weakly labeled datasets for event classes, and strongly labeled dataset for event presences. The method secondary applies the prediction of event presences at each frame on weakly labeled data as the class-frame pseudo label and re-trains the network. The class-frame pseudo label contributes to improvement of the event presence prediction at each frame by avoidance of overfitting to strongly labeled datasets.

2. MODEL ARCHITECTURE

Our model architecture is based on the ResNet. The output layer of our model is modified to output both the event classes and the event presences at each frame. The model is trained by two-stage training using the given labels and the class-frame pseudo labels.

Figure 1 shows the model architecture for the proposed method. An input is the acoustic feature map extracted from an audio clip. The model outputs two kinds of probability. One is a probability matrix (128×10) corresponding to the event presence at each frame. The other is a 10 dimensional probability vector which shows the event class in the audio clip. Our model consists of three blocks, a convolution block, a residual convolution block and a fully connection block.

2.1. Preprocessing

The given raw-waveform audio clip data are resampled by 44.1 kHz and preprocessed by extracting log-mel spectrogram features within the limits of 0 to 16 kHz. The dimension number of the log-mel feature is 64. After the log mel representations are extracted using 20 ms windows with 8 ms steps, they are normalized from 0 to 1 with min-max normalization. When an audio clip is shorter than 10 seconds, it is zero-padded to equalize the length. The output of this processing is the feature map, whose size is 1024×64.

2.2. Convolution block

The convolution block expands an input acoustic feature (1024×64) into 64 channel feature ($64 \times 1024 \times 64$). The convolution layer uses 64 channel 3×3 filters with stride size 1. The convoluted features are fed into batch normalization and rectified linear unit (ReLU) layers.

2.3. Residual convolution block

The residual convolution block consists of 4 residual blocks. Residual structure proposed by He et al. [2] makes the training of deep structure model easy by adding shortcut connection. The residual convolution block expands the input feature in the channel axis while repeating a process with changing parameters. The feature size is changed from $64 \times 1024 \times 64$ to $512 \times 128 \times 8$.

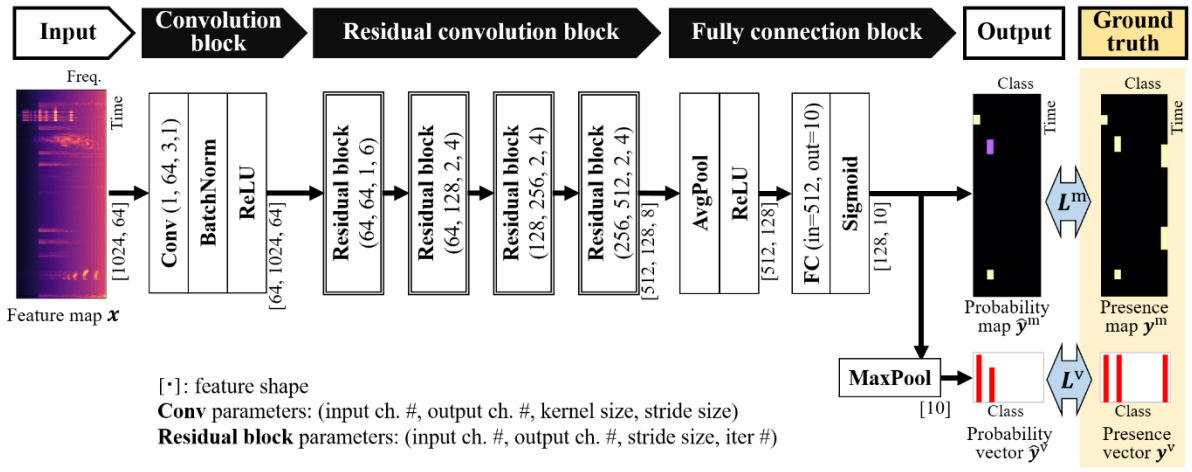

Figure 1. Model architecture for the system.

Figure 2 shows the architecture of a residual block utilized in our model. The block has 4 parameters to control the output feature size and the number of conversion process, which are the number of input and output channels, a stride size and an iteration number. In exchange for doubling the number of output channels, the block realizes contraction of feature dimensions by half. The residual block repeats processing of branches and additions. In Fig. 2, the right branch is called as a shortcut connection. The output feature is calculated by addition of the feature through shortcut connection and the converted features. In case of the first iteration, convolution and batch normalization layers are included in shortcut pass in order to adjust the channel numbers.

2.4. Fully connection block

As shown in Fig. 1, the fully connection block outputs a probability matrix and a 10 dimensional probability vector. The feature tensor ($512 \times 128 \times 8$) is converted into a feature matrix (512×128) by feature-dimension-wise average pooling layer and ReLU activation layer. A fully connected layer projects 512 dimensional features on 10 dimensional probability vectors corresponding to class presence. The output of sigmoid layer is a feature matrix and shows the probability for time wise sound event class. In order to represent the class in an audio clip using the probability vector, max pooling layer is adopted at the end of model.

3. PROPOSED TRAINING METHOD

The class-frame pseudo labels for weakly labeled datasets are utilized for the model training in order to avoid overfitting to strongly labeled datasets. The training step is divided into two stages based on labels of the weakly labeled datasets.

Figure 3 shows an overview of the proposed training method. At the first stage of training, the model is trained only using given label data of strongly and weakly labeled datasets. When the first model training is finished, weakly labeled dataset is classified by the given vector y^{v} and probability vector $\hat{y}^{v}$. The class-frame pseudo label is given to samples in which the given label and binarized probability vector match. In the second stage of training, the model is trained by the given and class-frame pseudo labels.

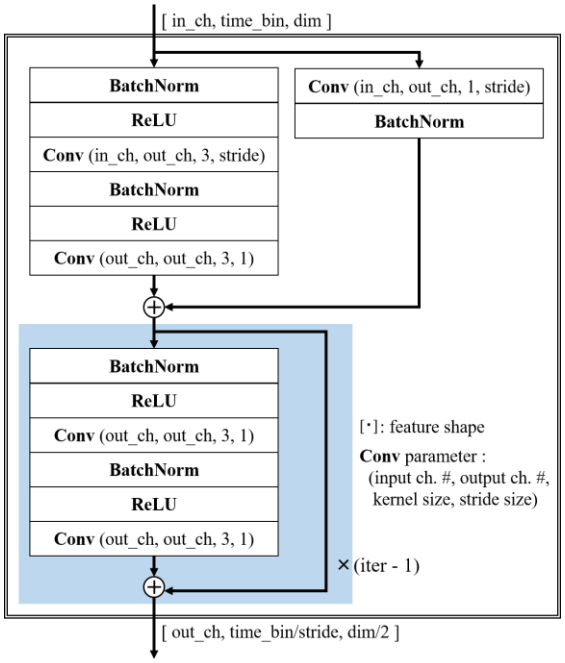

Figure 2. Architecture of the residual block.

The class-frame pseudo label is updated using the ground truth and the second model output.

3.1. Training loss

Our training method utilizes weighted binary cross entropy (BCE) loss [3]. The parameters of the network are updated by stochastic gradient descent optimizer using gradients of the loss calculated by back propagation.

BCE loss is widely utilized to calculate the loss between the ground truth and a predicted probability of audio clip [4, 5]. The

weighted BCE loss solves the class imbalance with a hyper parameter w_c, where c is a class index. The weighted BCE loss is defined as:

$$L = \frac{1}{N_{all}} \sum_{n=1}^{N_{all}} \sum_{c=1}^{C} w_c \, E\left(y_{n,c}, \hat{y}_{n,c}\right), \tag{1}$$

$$w_c = \frac{e^{1/N_c}}{\sum_{c=1}^{C} e^{1/N_c}}, \tag{2}$$

$$E\left(y_{n,c}, \hat{y}_{n,c}\right) = -(y_{n,c} \log \hat{y}_{n,c}) + \left(1 - y_{n,c}\right) \log\left(1 - \hat{y}_{n,c}\right), \tag{3}$$

where L is the weighted BCE loss, N_c is the sample numbers of class c, N_{all} and C are the total number of samples and classes, and $E\left(y_{n,c}, \hat{y}_{n,c}\right)$ represents the BCE loss for the ground truth $y_{n,c}$ and the predicted probability $\hat{y}_{n,c}$.

The strongly labeled datasets have two types of label, the time wise event presence matrix $\boldsymbol{y}^{\mathrm{m}}$ and the event class presence vector $\boldsymbol{y}^{\mathrm{v}}$, where m and v indicate matrix and vector, respectively. The losses of these datasets are defined as $L_{\mathrm{s}}^{\mathrm{m}}$, $L_{\mathrm{s}}^{\mathrm{v}}$, where s means strongly labeled data. In the first training, the weakly labeled datasets have only the event class presence vector $\boldsymbol{y}^{\mathrm{v}}$. However, the second training is given the class-frame pseudo label $\boldsymbol{y}'^{\mathrm{m}}$. The losses of these datasets are defined as $L_{\mathrm{w}}^{\mathrm{v}}$, $L_{\mathrm{p}}^{\mathrm{m}}$, where w means weakly labeled data and p means class-frame pseudo label data. The number of audio clips with strong and weak label are M_{s} and M_{w}, respectively.

3.2. Pseudo label

The pseudo labeling proposed by Lee [6] for semi-supervised learning regularizes the training to avoid overfitting to the given label data. We expand this method for weakly labeled data to annotate the event presences at each frame. The prediction matrix $\hat{y}_{n,c}^{\mathrm{m}}$ are binarized by a threshold θ_i as (4) and (5).

$$y'^{\mathrm{m}}_{n,c} = \begin{cases} 1 & \text{if } \hat{y}_{n,c}^{\mathrm{m}} \geq \theta_i \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

$$\theta_i = \min\left(-\delta \cdot i + \frac{3}{4}, \frac{1}{4}\right), \tag{5}$$

where $y'^{\mathrm{m}}_{n,c}$ represents the class-frame pseudo label, c and i are indices of class and training iteration, respectively. The threshold θ_i is changed according to the model training iteration i. On the assumption that a reliability of the prediction result gets higher as the training progresses, the threshold is set lower than previous iteration's one from 0.75 to 0.25.

3.3. Training on first stage

In the first stage of training, a total loss is calculated with given labels of the datasets. After losses are calculated in each dataset, the total loss $L^{\mathrm{all\_1st}}$ is computed as a sum of them:

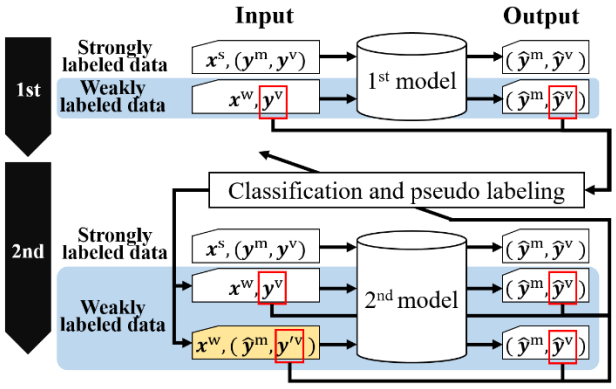

Figure 3. Overview of the proposed training method.

$$L^{\mathrm{all\_1st}} = L^{\mathrm{v}} + L^{\mathrm{m\_1st}}, \tag{6}$$

$$L^{\mathrm{v}} = \frac{M_{\mathrm{w}}}{(M_{\mathrm{w}}+M_{\mathrm{s}})} L_{\mathrm{w}}^{\mathrm{v}} + \frac{M_{\mathrm{s}}}{(M_{\mathrm{w}}+M_{\mathrm{s}})} L_{\mathrm{s}}^{\mathrm{v}}, \tag{7}$$

$$L^{\mathrm{m\_1st}} = \frac{(M_{\mathrm{w}}+M_{\mathrm{s}})}{M_{\mathrm{s}}} L_{\mathrm{s}}^{\mathrm{m}}, \tag{8}$$

where L^{v} is a vector loss of the weak and strong labels and $L^{\mathrm{m\_1st}}$ is a matrix loss of the strong label. The weights based on sample numbers in (7) and (8) are decided so that the contribution ratios to the total loss $L^{\mathrm{all\_1st}}$ are the same for the vector loss L^{v} and matrix loss $L^{\mathrm{m\_1st}}$.

3.4. Training on second stage

In the second stage of training, the model is trained with the given labels and the class-frame pseudo labels.

First, the weakly labeled data are classified by the given labels and the predicted vectors by the model trained in the first stage. When a binarized prediction vector of the data is equal to the ground truth of weak label, the data is given the class-frame pseudo label. Here, the binarizing threshold is defined by (5). The class-frame pseudo label is generated using the class-frame prediction vector of the data as shown in subsection 3.2.

Then, a total loss of the second training is computed using the given labels and the class-frame pseudo labels. The number of data with pseudo label is defined as M_{p}, which is less than or the same as that of weakly labeled data, M_{w}. The second stage training loss $L^{\mathrm{all\_2nd}}$ is:

$$L^{\mathrm{all\_2nd}} = L^{\mathrm{v}} + L^{\mathrm{m\_2nd}}, \tag{9}$$

$$L^{\mathrm{m\_2nd}} = \frac{(M_{\mathrm{w}}+M_{\mathrm{s}}) \times M_{\mathrm{s}}}{(M_{\mathrm{s}}+M_{\mathrm{p}})^2} L_{\mathrm{s}}^{\mathrm{m}} + \frac{(M_{\mathrm{w}}+M_{\mathrm{s}}) \times M_{\mathrm{p}}}{(M_{\mathrm{s}}+M_{\mathrm{p}})^2} L_{\mathrm{p}}^{\mathrm{m}}, \tag{10}$$

where $L^{\mathrm{m\_2nd}}$ is a matrix loss of the strong label and class-frame pseudo label. The vector loss L^{v} is calculated in the same manner as (7).

Table 1. Evaluation result on validation dataset.

	Event-based F1-score	Segment-based F1-score
DCASE 2019 baseline [1]	23.7 %	55.2 %
ResNet **without** pseudo label	24.1 %	61.3 %
ResNet **with** pseudo label (proposed)	25.9 %	62.0 %

After the model is trained using the loss $L^{\text{all\_2nd}}$ in the i-th iteration, the weakly labeled datasets are classified again. This process is repeated to the end of training.

4. EVALUATION

4.1. Experimental condition

The DCASE 2019 task 4 datasets consists of 10 category audio data occurring in domestic environments: speech, dog, cat, alarm/bell ringing, dishes, frying, blender, running water, vacuum cleaner and electric shaver/toothbrush. The training datasets includes strongly annotated synthetic data and weakly annotated real data.

4.2. Evaluation results

Table 1 compares F1-scores by the DCASE2019 baseline system and our systems (with and without class-frame pseudo label). The F1-scores by the ResNet without pseudo label are 24.1% and 61.3% in the event-based and segment-based evaluation, respectively. The F1-scores by the ResNet with pseudo label are 25.9% in the event-based evaluation and 62.0% in the segment-based evaluation. These results indicate that the proposed method outperforms the baseline scores.

5. CONCLUSIONS

This paper presented a training method using class-frame pseudo label for weakly labeled datasets given by the DCASE2019 task 4. Proposed method firstly trains the model based on a ResNet using both strongly and weakly labeled datasets for event classes, and strongly labeled dataset for event presences. The method secondary applies the prediction of event presences in each frame on weakly labeled data as the class-frame pseudo label and re-trains the network. The class-frame pseudo label contributes to improvement of the event presence prediction at each frame by avoidance of overfitting to strongly labeled datasets. A result shows that F1-scores by our proposed method are 25.9% and 62.0% in the event-based and segment-based evaluations, respectively.

6. REFERENCES

[1] http://dcase.community/workshop2019/.

[2] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv: 1512.03385*, https://arxiv.org/abs/1512.03385, 2015.

[3] C. Y. Hsieh, Y. A. Lin and H. T. Lin, "A Deep Model with Local Surrogate Loss for General Cost-sensitive Multi-label Learning," in *Proc. AAAI*, 2018.

[4] Y. Xu, Q. Kong, W. Wang and M. D. Plumbley, "Large-scale Weakly Supervised Audio Classification Using Gated Convolutional Neural Network," in *Proc. IEEE ICASSP*, 2018, pp. 121-125.

[5] E. Çakır, G. Parascandolo, T. Heittola, H. Huttunen and T. Virtanen, "Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection," *IEEE Trans. ASLP*, vol. 25, no. 6, pp. 1291-1303, Jun. 2017.

[6] D. H. Lee, "Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks," in *Proc. WREPL*, 2013.