MULTI-LABEL AUDIO TAGGING SYSTEM FOR FREESOUND 2019: FOCUSING ON NETWORK ARCHITECTURES, LABEL NOISY AND LOSS FUNCTIONS

Technical Report

Xiaofeng Hong, Gang Liu

Beijing University of Posts and Telecommunications
Beijing, China
{hongxiaofeng, liugang}@bupt.edu.cn

ABSTRACT

In this technical report, we propose our solutions applied to our submission for DCASE2019 Task2. We focus on the model architectures which can efficiently tag the audio with multi-label and noisy label. Furthermore, we use multi-label models based on convolutional network and recurrent network to unify detection of audio events. Graph representation is also utilized to take the audio event co-occurrence into account which is reflected in the loss functions. We also tried Semi-Supervised Learning to use the noisy data. Finally, we tried an ensemble of CNNs and CRNN, trained by using cross validation folds. Compared to the baseline score of 0.537, we achieved a score of 0.700 on the public leaderboard.

Index Terms— Audio event, CNN, CRNN, graph representation, Semi-Supervised Learning

1. INTRODUCTION

Due to lack of large-scale audio datasets, applying machine learning methods to the field of audio events classification and detection is challenging. The Detection and Classification of Acoustic Scenes and Events 2019 (DCASE2019) is held to encourage people to explore novel approaches which can applied to specific audio challenges. There are five tasks in DCASE2019 challenge, this technical report describes our submission of task 2 [1, 2], audio tagging with noisy labels and minimal supervision.

In freesound audio tagging 2019, officials provided two data sets, a small set of manually-labeled data and a lager set of noisy-labeled data. How to use noisy data effectively is the key point to get a high score.

Recently, neural networks are widely used to audio classification tasks [3, 4]. We also used two types of deep learning networks: CNNs and CRNNs. In order to combine their performance, we use an ensembling method to get the final prediction.

2. PROPOSED FRAMEWORK

Compared to the single label audio tagging task, we put stronger focus on considering the multi-label and noisy label. In the sections to follow, we describe the audio features, loss functions and network architectures.

2.1. Input

The audio signals, sampled at 44.1KHz, are converted into log-mel spectrograms. A window size of 46ms and 40ms with 50% overlap are used. The first feature extraction scheme is 128 mel-scale filters which is used to train the CNN model. The second feature extraction scheme is 40 mel-scale filters which is used to train the CRNN model. When the lengths of the audios in the curated datasets are different, we used an audio segment of T = 128 (3 channels) and T = 512 (one channel) respectively, where the T is the time dimension of the feature. The lengths of most audios in the noisy datasets are 15s. We also stacked the one channel feature to 3 RGB channels to train a CNN model.

The figure below shows the number of audio clips in curated datasets.

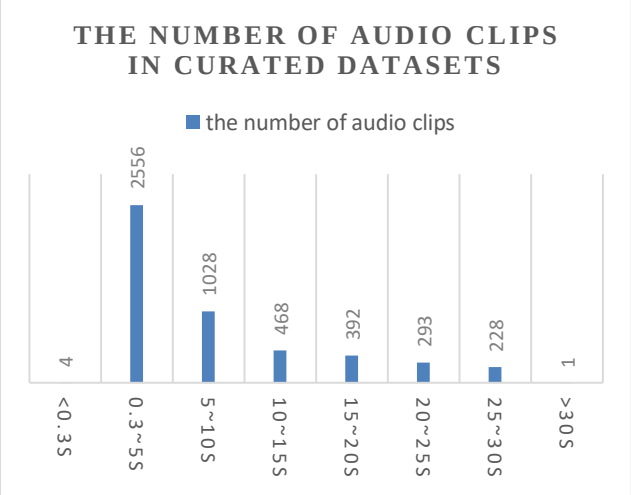

Figure 1: the number of audio clips

2.2. Network Architecture

We experimented with two different network architectures. CNN has achieved many excellent results in the field of image recognition, so we first adopted the *VGG-style* [5] convolutional network, and the model structure is shown in table 1.

Table 1: Description of convolutional neural network architecture

Input 3×128×128 (1×128×512)
3×3 Conv(stride-1, pad-1)-64-BN-RELU
3×3 Conv(stride-1, pad-1)-64-BN-RELU
2×2 MaxPool(stride-2)
3×3 Conv(stride-1, pad-1)-128-BN-RELU
3×3 Conv(stride-1, pad-1)-128-BN-RELU
2×2 MaxPool(stride-2)
3×3 Conv(stride-1, pad-1)-256-BN-RELU
3×3 Conv(stride-1, pad-1)-256-BN-RELU
2×2 MaxPool(stride-2)
3×3 Conv(stride-1, pad-1)-512-BN-RELU
3×3 Conv(stride-1, pad-1)-512-BN-RELU
2×2 MaxPool(stride-2)
512×80 FC(dropout-0.4)

We also tried other convolutional networks, including ResNet18 [6], SEResNet [7] and Densenet [8]. Due to the differences between audio and image, deeper model did not get better results.

Because of the strong temporal correlation of audio, we tried to model time series. So we used the convolutional layers and Gated Recurrent Unit (GRU) [9] to form the network, convolutional layers learn effectives features and higher recurrent layers perform sequential modelling. The CRNN architecture is shown in table 2.

Table 2: Description of convolutional recurrent neural network architecture

Input 1×40×512
5×5 Conv(stride-1×1, pad-SAME)-256-BN-RELU
5×1 MaxPool(stride-5×1)
5×5 Conv(stride-1×1, pad-SAME)-256-BN-RELU
4×1 MaxPool(stride-4×1)
5×5 Conv(stride-1×1, pad-SAME)-256-BN-RELU
2×1 MaxPool(stride-2×1)
CNN output: x
BiGRU-2-layers (dropout-0.2)
GRU output: a
a + x (Residual)
512×80FC(dropout-0.4)

2.3. Loss Function

Since the audio tagging task is a multi-label problem. So, we use the MultiLabelSoftMarginLoss in PyTorch [10]. It creates a criterion that optimizes a multi-label one-versus-all loss based on max-entropy between input x and target y of size (N, C).

$$\text{loss}(x,y) = -\frac{1}{C} * \sum_i y[i] * \log((1+\exp(-x[i]))^{-1}) + (1-y[i]) * \log(\frac{\exp(-x[i])}{(1+\exp(-x[i]))}) \quad (1)$$

To count the co-occurrence of audio evens, we also introduce the graph Laplacian regularization as described in section 2.4.

2.4. Graph Laplacian Regularization

Conventional methods cannot consider the co-occurrence of the audio events. In this case, we leverage the power of Laplacian matrix to take events co-occurrence into account [11]. In our experiment, it can significantly reduce the training time but with increased performance. The graph Laplacian matrix L [12] is defined as

$$\mathrm{L} = \Delta - \mathrm{A} \quad (2)$$

where Δ is degree matrix which is diagonal. A is called adjacency matrix. The adjacency matrix was calculated by counting the number of co-occurring audio events in each clip over the curated training datasets.

$$\Delta_{ii} = \sum_j A_{i,j} \quad (3)$$

So, our loss function is finally given as

$$\text{Loss} = \text{MultiLabelSoftMarginLoss} + \beta \text{Tr}\{(\sum_{t=1}^{T} y_t)^T L \ (\sum_{t=1}^{T} y_t)\} \quad (4)$$

In our work, the regularization weight β is 1e-5.

2.5. Semi-Supervised Learning (SSL)

One of the features of this task is that it contains a lot of noisy-labeled data. In the past year's Freesound challenge, many solutions were used to handle noisy label, such as pseudo-labeling, which was to relabel the weakly labeled data using the pre-trained models.

In our experiment, we used a Semi-Supervised Learning method called Interpolation Consistency Training (ICT), proposed in [13]. In order to verify the effectiveness of ICT, we split the curated datasets into two parts, training set and evaluation set. The training set is 20% of the total and the evaluation set is 80% of the total. In addition, we also used full noisy-labeled data to train the model. But we removed the noisy label.

We used the *VGG-style* model and trained for 200 epochs. Finally, we achieved a score of 0.612 on the evaluation set. The model also can make our public leaderboard score higher .

Furthermore, due to time constraints, we did not get better results using ICT. In the future, we will go further with ICT or other SSL methods.

3. EXPERIMENT

3.1. Experiment Settings

In our experiment, we used the same learning rate to train all CNNs, which is different from CRNN. But we used the same learning rate adjustment strategy, and Adam [14] was used as the gradient descent algorithm. Refer to table 3 for the values.

Table 3: Training hyper-parameters

Hyper-parameters	Values
Batch-size	32
Learning rate (LR)	3e-3 (CNNs) / 1e-4 (CRNN)
LR decay factor	0.9
LR decay rate	2

3.2. Data Augmentation

Mix up [15] is a data augmentation method used during training with the curated and noisy datasets. It linearly mixes two training data and then inputs into the model. Let x_i and x_j are two samples from the train loader, y_i and y_j are the corresponding one-hot label, then the mix up generates an augmentation data $\hat{x}$ and its label $\hat{y}$ as follows:

$$\hat{x} = \alpha x_i + (1 - \alpha)x_j \quad (5)$$
$$\hat{y} = \alpha y_i + (1 - \alpha)y_j \quad (6)$$

where $\alpha \in (0, 1)$. In our work, we set α to be a variable of Beta(0.2, 0.2).

In addition, we also tried another audio data augmentation method called SpecAugment, proposed in [16]. But we did not get a significant improvement. So, in order to save the inference time on the stage 2, we did not use this method.

3.3. Evaluation Metric

Label-weighted label-ranking average precision is the primary metric of this task. Formally, given a binary indicator matrix of the ground truth labels $y \in \{0,1\}^{n_{samples} \times n_{labels}}$ and the score of each label $\hat{f} \in \mathbb{R}^{n_{samples} \times n_{labels}}$, the average precision is defined as

$$\mathrm{LRAP}(y, \hat{f}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \frac{1}{\|y_i\|_0} \sum_{j:y_{ij}=1} \frac{|L_{ij}|}{rank_{ij}} \quad (7)$$

where $L_{ij} = \{k: y_{ik} = 1, \hat{f}_{ik} \geq \hat{f}_{ij}\}$, $rank_{ij} = |\{k: \hat{f}_{ik} \geq \hat{f}_{ij}\}|$, $|\cdot|$computes the cardinality of the set, and $\|\cdot\|_0$ is the L_0 "norm".

4. RESULTS

We used 5-fold cross validation to select the weight best model. Each fold has the same number of categories. The table below shows the cross-validation score of each fold.

Table 4: Stratified folds of curated datasets

Fold	1	2	3	4	5	total
Clips	991	1001	993	982	1003	4970

Table 5: The cross-validation score of each fold

	Fold1	Fold2	Fold3	Fold4	Fold5
VGG-style	0.851	0.839	0.837	0.860	0.840
CRNN	0.839	0.809	0.821	0.848	0.824

Finally, we submitted 2 submissions with different model ensemble. We fused the CNNs and CRNN by using weighted geometric mean as follows:

$$Y_{en} = \exp\left(\frac{1}{N}\sum_n \mu_n \log(y_n)\right) \quad (8)$$

Where N denotes the number of models and μ denotes the weight of different models. In our work, μ is 0.4 for the outputs from CRNN and 0.6 for CNNs. The N is 27 and 26:
5(fold) × (VGG × 2 + ResNet18 + Densenet + SEResNet) + CRNN + ICT_VGG and 5(fold) × (VGG × 2 + ResNet18 + Densenet + SEResNet) + ICT_VGG.

Table 6: Score of our submissions

Number of models	Public leaderboard
26	0.699
27	0.699

5. REFERENCES

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

[2] Eduardo Fonseca, Manoj Plakal, Frederic Font, Daniel P. W. Ellis, and Xavier Serra. "Audio tagging with noisy labels and minimal supervision". *Submitted to DCASE2019 Workshop, 2019. URL: https://arxiv.org/abs/1906.02975*

[3] Y. Xu, Q. Kong, W. Wang, and M. D. Plumbley, "Large-scale weakly supervised audio classification using gated convolutional neural network," *arXiv preprint arXiv:1710.00343*, 2017.

[4] E. Fonseca, M. Plakal, F. Font, D. P. W. Ellis, X. Favory, J. Pons, and X. Serra, "General-purpose tagging of Freesound audio with AudioSet labels: Task description, dataset, and baseline," *arXiv preprint arXiv:1807.09902*, 2018.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd Int. Conf. Learn. Repr. (ICLR)*, San Diego, CA, 2015.

[6] 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.

[7] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, vol. 7, 2017.

[8] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks." in *CVPR*, vol. 1, no. 2, 2017, p. 3.

[9] K. Cho, B. van Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proc. EMNLP*, 2014, pp. 1724–1734.

[10] https://pytorch.org/

[11] Keisuke Imoto, Seisuke Kyochi, "Sound Event Detection Using Graph Laplacian Regularization Based on Event Co-occurrence," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.

[12] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, 2013.

[13] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. "Interpolation consistency training for semi-supervised learning," *arXiv preprint arXiv:1903.03825*, 2019.
[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. Learn. Repr. (ICLR)*, San Diego, CA, 2015.
[15] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *6th Int. Conf. Learn.Repr. (ICLR)*, Vancouver, Canada, 2015.
[16] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, Quoc V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," *arXiv preprint arXiv: 1904.08779*, 2019.