Multisystem fusion model based on tag relationship

Technical Report

Zhuangzhuang Liu, Junyan Fang, Xiaofeng Hong, Gang Liu

Beijing University of Posts and Telecommunications
Beijing, China
{ liuzhuangzhuang2345, fangjunyan, hongxiaofeng, liugang}@bupt.edu.cn

ABSTRACT

Audio tagging aims to assign one or more labels to the audio clip. In this paper, we proposed our solutions applied to our submission for DCASE2020 Task5 It focus on predicting whether each of the sources of noise pollution is present or absent in a 10-second scene [1]. And we should consider the spatiotemporal context(STC) in our work. We used VGG as our basic model and we regarded Multi-task learning as a method to train our models. We introduced the relationship between fine labels and coarse labels in our system. Finally, the coarse-grained and fine-grained taxonomy results are obtained on the Micro Area under precision-recall curve (AUPRC), Micro F1 score and Macro Area under precision-recall curve (AUPRC).

Index Terms— Audio tagging, VGG, Multi-task learning, relationship between labels

1. INTRODUCTION

The sounds in our everyday environment carry a lot of information about events happening nearby, but the machine sound processing is still far behind. Audio tagging plays an important role in multimedia understanding. The Sound of New York City (SONYC)[2] is a system for monitoring, analyzing and mitigating urban noise pollution. One of its goals is to map the spatiotemporal distribution of noise in real time and over the years in large cities such as New York. In order to reduce noise pollution, citizen participation is crucial, but some residents are unlikely to file a complaint with the city officials. Therefore, the goal of the DCASE 2020 task 5 is to predict whether there are 29 kinds of noise pollution in the 10 second scene recorded by the acoustic sensor network. Differently, this year's task provides identifiers for the New York City block (location) where the recording was taken as well as when the recording was taken, quantized to the hour. We need to consider the relationship between spatiotemporal context (STC) metadata and our labels' prediction.

2. PROPOSED FRAMEWORK

Compared to the single label audio tagging task, we put stronger focus on considering the relationship between fine labels and coarse labels. In the sections to follow, we describe the audio features, loss functions and network architectures. We also will introduce how we deal with the labels offered by the official.

2.1. Input

Recordings are resampled to 32000 Hz and to generate mel spectrogram with a Hanning window size of 1024 and hop length of 500 samples. Mel filters which band is 64 are used to transformed STFT spectrogram to mel spectrogram, and frequencies lower than 50 Hz and beyond 14000 Hz are removed.

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 [3] convolutional network, and the model structure is shown in table 1. After that, we also used two VGG to achieve the Multi-task learning. Both VGGs are same with the VGG mentioned before.

2.3. Loss Function

We regarded the multi-class and multi-label task as many binary problems. So we used binary cross loss function, which is offered by torch.

$$\text{loss} = -\sum_{i=1}^{n} \hat{y}_i log y_i + (1-\hat{y}_i)\log(1-y_i) \quad (1)$$

2.4. One-hot targets

In the train dataset and validate dataset, each data will be commented by at least three annotators. And they may give different labels to the same data. In our system, we used one-hot targets as our target. We calculate the average of each classes of noise for every data. And we set a threshold to judge whether the data include those kinds of events. If the average is more than the threshold, we think it includes. And if threshold is more than the average, we think it doesn't include. And Our processing of these data may result in some audio' label being 0 for each class. And we will delete these data and don't train them in our model.

3. EXPERIMENT

In this part we will introduce how we conduct our experiment. We also will show how we used the STC data.

Table 1: Description of convolutional neural network architecture

Input 1×640×64
3×3 Conv(stride-1, pad-1)-64-BN-RELU
3×3 Conv(stride-1, pad-1)-64-BN-RELU
2×2 AvgPool(stride-2)
3×3 Conv(stride-1, pad-1)-128-BN-RELU
3×3 Conv(stride-1, pad-1)-128-BN-RELU
2×2 AvgPool(stride-2)
3×3 Conv(stride-1, pad-1)-256-BN-RELU
3×3 Conv(stride-1, pad-1)-256-BN-RELU
2×2 AvgPool(stride-2)
3×3 Conv(stride-1, pad-1)-512-BN-RELU
3×3 Conv(stride-1, pad-1)-512-BN-RELU
1×1 AvgPool(stride-2)
FC(512+16, 512+16,bias=true)
FC(512+16, class_num,bias=true)

Table 2: Training hyper-parameters

Hyper-parameters	Values
Batch-size	32
Learning rate (LR)	1e-3
LR decay factor	0.9

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 [4] was used as the gradient descent algorithm. Refer to table 2 for the values.

3.2. Data Augmentation

Mix up [5] 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:

$$\hat{x} = \alpha x_i + (1-\alpha)x_j \quad (2)$$

$$\hat{y} = \alpha y_i + (1-\alpha)y_j \quad (3)$$

where $\alpha \in (0,1)$. In our work, we set α to be a variable of Beta(1.0, 1.0).

In addition, we also tried another audio data augmentation method called SpecAugment, proposed in [6]. 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

The UST challenge is a task of multilabel classification. The area under the precision-recall curve (AUPRC) is the classification metric to evaluate. And for coarse-grained and fine-grained AUPRC, micro-auprc and macro-auprc are both computed, Fscore is used for analysis as well.

Table 3: Significance of the STC tensor

year	tensor[0]-tensor[3]
week	tensor[4]-tensor[7]
day	tensor[8]-tensor[9]
hour	tensor[10]-tensor[13]
location	tensor[14]-tensor[15]

Table 4: Models of the submissions

Model name	One-hot data	Loss monitor	Data augmen-tation	Multi-task
Model1	no	no	no	yes
Model2	yes	no	no	yes
Model3	non	no	yes	yes
Model4	yes	no	yes	yes
Model5	no	no	yes	yes
Model6	yes	no	yes	yes
Model7	yes	yes	yes	yes
Model8	no	no	no	yes
Model9	yes	no	no	yes
Model10	no	no	yes	no
Model11	yes	no	yes	no

3.4. STC Data

In our system, the STC data will be added after the convolutional layers and before the full connected layer. To decrease the number of the parameters, we set a 1×16 tensor for each STC data. the significance of the tensor is showed in table 3. The tensor will be added to the first FC's input after the last convolutional layer. And we don't normalize the STC tensor.

3.5. Loss Monitor

Because the train data is unreliable, we used a method to reduce the influence brought by them. If some audios' loss is proved to be too large, we will delete them and don't let them to be back-forward. By this way we improved our statistics successfully.

4. RESULTS

In table 4 we will show models we used to get our 4 submissions. We mixed some models' result up to get our 4 submissions. Our mix followed the formula below:

$$Y_{en} = \exp(\sum_n \mu_n \log(y_n)) \quad (4)$$

4.1. Submission1

We mixed all models up in this submission. And the μ of each model is follow the following list: (0.08, 0.08, 0.08, 0.08, 0.04, 0.08, 0.04, 0.04, 0.32, 0.08, 0.08)

4.2. Submission2

We mixed model2, model4, model6, model7, model9, model11 in this submission. And the μ of each model is follow the following list: (0.15, 0.15, 0.15, 0.05, 0.35, 0.15)

4.3. Submission3

We mixed model2, model8 and model9 in this submission. And the μ of each model is follow the following list: (0.4, 0.2, 0.4)

4.4. Submission4

We mixed model2, model4, model6, model9, model11 in this submission. And the μ of each model is follow the following list: (0.2, 0.2, 0.2, 0.2, 0.2)

5. REFERENCE

[1] http://dcase.community/workshop2020/.
[2] Cartwright, M., Mendez, A.E.M., Cramer, J., Lostanlen, V., Dove, G., Wu, H., Salamon, J., Nov, O., Bello, J.P. "SONYC Urban Sound Tagging (SONYC-UST): A Multilabel Dataset from an Urban Acoustic Sensor Network", Proceedings of the Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE) , 2019.
[3] 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.E. G. Williams, Fourier Acoustics: Sound Radiation and Nearfield Acoustic Holography, London, UK: Academic Press, 1999.
[4] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in 3rd Int. Conf. Learn. Repr. (ICLR), San Diego, CA, 2015.
[5] 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.
[6] 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.