Simple Convolutional Networks Attempting Acoustic Scene Classification Cross Devices

Technical Report

Chi Zhang[1], Hanxin Zhu[2], Ting Cheng[3]

University of Electronic Science and Technology of China, Chengdu, China
[1]18zc@std.uestc.edu.cn
[2]2018011212014@std.uestc.edu.cn
[3]citrus@uestc.edu.cn

ABSTRACT

This technical report describes our submission for task1a (Acoustic Scene Classification with Multiple Devices) of the DCASE 2020 Challenge. The results of the DCASE 2019 show that the convolution neural networks (CNNs) can acquire excellent classification accuracies. Our work will still be based on the convolution neural networks. We consider two feature extraction methods that are provided by OpenL3 library. Finally, our method improves the accuracy of classification by 2% as compared to the baseline system.

Index Terms— DCASE 2020, convolution neural network (CNN), acoustic scene classification

1. INTRODUCTION

Sound contains a lot of information [1]. We can judge the source of sound by the sound we receive, so as to judge different acoustic scenes and acoustic events. With the rapid development of artificial intelligence, machines can also replace us to make the above judgments with high accuracy.

The DCASE challenge focuses on acoustic scene recognition and classification. This year the challenge is divided into six different tasks. Task 1 remains focusing on the classification of acoustic scenes. There are 2 subtasks in task 1. In the subtask 1a, the development set contains data from 10 cities and 9 devices: 3 real devices and 6 simulated devices.

So far, the development of CNN has been relatively mature, and it has been widely used in computer vision, with outstanding advantages in speech recognition and image processing [2-4]. Our work is still based on the convolutional neural network. In this report, we describe the feature extraction methods and CNN structures used to get our submissions.

2. DATA PERPROCESSING

In this section, we will describe the method of data preprocessing. We use OpenL3 library for audio embedding. The size of analysis window is 1 second, and size of hop are set to 100ms and 50ms in turn to get different information about the audio. The embedding size is 512, and the input representation is Mel spectrum with 256 frequency banks.

Input (hop-size = 50ms)
2*2 Conv2D-32-BN-ReLU 5*5 MaxPooling2D Dropout-0.5
2*2 Conv2D-64-BN-ReLU 4*100 MaxPooling2D Dropout-0.5
Dense(512,ReLU)
Dense(10,softmax)

Figure 1 Network structure “N1”.

Input (hop-size = 100ms)
3*3 Conv2D-32-BN-ReLU 5*5 MaxPooling2D Dropout-0.5
3*3 Conv2D-64-BN-ReLU 4*100 MaxPooling2D Dropout-0.3
Dense(512,ReLU)
Dense(10,softmax)

Figure 2 Network structure of “N2”.

3. NETWORK STRUCTURE

A conventional CNN consists of several convolutional layers, and each layer contains filters to convolve with the output from the previous layer. The filters can capture local patterns of the input feature maps. It also includes pooling layers and dropout layers to prevent model overfitting, and the last layers of the network usually consist of a full connection layer.

The architecture of the CNN we use is adapted from the baseline system. We adopt three kinds of CNNs with different depths: two 3-layer CNNs and a 7-layer CNN. Figures 1-3 summary the network structure in this report.

The network structure “N1” is inspired by baseline system. It consists of two convolutional layers with kernel size of 2*2 to extract more detailed information, and the number of filters per layer is 32 and 64 to get more features, the pooling layer adopts 5*5 max-pooling and 4*100 max-pooling to get different sensory

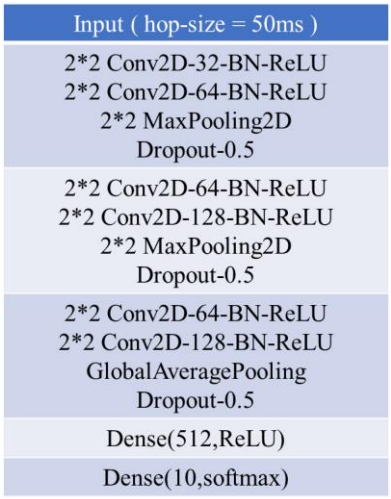

Figure 3 Network structure "N3".

fields. After each pooling layer, there is a drop out layer to prevent model overfitting, and the rate of dropout is 0.5. The hop size of input data is 50ms to get more features. The output size of full connected layer is 512.

The network structure "N2" is inspired by baseline system too. It consists of two convolutional layers with kernel size of 3*3 to extract more detailed information, and the number of filters per layer is 32 and 64 to get more features, the pooling layer adopts 5*5 max-pooling and 4*100 max-pooling to get different sensory fields. After each pooling layer, there is a drop out layer to prevent model overfitting, and the rate of dropout is 0.3. The hop size of input data is 100ms which is the same as baseline system. The output size of full connected layer is 512.

The network structure "N3" is inspired by Q. Kong's network [5]. It consists of six convolutional layers with kernel size of 2*2 to extract more detailed information, and the number of filters per layer is 32, 64, 64, 128, 128 and 256 to get more features. Each two layers of convolution is followed by a pooling layer and the pooling mode is 2*2 max-pooling ,2*2 max-pooling and global-average-pooling. After each pooling layer, there is a drop out layer to prevent model overfitting, and the rate of dropout is 0.5. The hop size of input data is 50ms to get more features. The output size of full connected layer is 512.

For all the above models, the data is activated by the ReLU function. Batch normalization is included in all the models to accelerate the learning process and improve the baseline level by regularization terms [6]. Batch normalization can also prevent overfitting. Moreover, padding is consistently used in all the models to make the input and output with the same dimensions. In the judgment layer, the audio is classified into 10 acoustic scenes by softmax, because of its outstanding performance in multi-class classification tasks.

4. EXPERIMENTS

4.1. Datasets

The development set contains data from 10 cities. The total amount of audio in the development set is 40 hours. The development set contains data from 10 cities and 9 devices: 3 real devices (A, B, C) and 6 simulated devices (S1-S6). Data from devices B, C and S1-S6 consists of randomly selected segments from the simultaneous recordings, therefore all overlap with the data from device A, but not necessarily with each other. The total amount of audio in the development set is 64 hours.

The dataset is provided with a training/test split in which 70% of the data for each device is included for training, 30% for testing. In 'eval' mode, all development sets are used for training the model, and there is a special evaluation set for model evaluation. Some devices appear only in the test subset. In order to create a perfectly balanced test set, a number of segments from various devices are not included in this split.

4.2. Train Procedure

After audio preprocessing, the data will be sent to convolutional neural network for learning. After each development data set is learned, the test set is used for model assessment. In the process of assessment, the loss function adopts categorical cross entropy. After the assessment is completed, the next round of learning takes place. The optimizer uses Adam algorithm, whose learning rate and batch size are set to 0.001 and 16. And model performance after each epoch is evaluated on the validation set, and best performing model is selected.

5. RESULTS

Table 1 shows the output results of the above three CNNs, from which we can see that "N1" and "N2" have outperformed the baseline system by about 2% and 1%, respectively. The result of "N3" in 'dev' mode is slightly worse than baseline.

Based on the comprehensive model architecture and testing results, it appears that both smaller convolution kernel and smaller hop size can help to extract more audio information. Global average pooling shows better performance and the dropout rate has a great influence on the classification accuracies.

Table 1 : Classification accuracies of different models

Network structure	Classification accuracy
"N1"	0.574
"N2"	0.561
"N3"	0.537
Baseline	0.554

6. CONCLUSIONS

In this report, we have introduced audio preprocessing, CNN structures and training process in our submission. Although the proposed model is simple, it can still improve the accuracy of acoustic scene classification to some extent.

7. REFERENCES

[1] D. Turnbull, L. Barrington, D. Torres, and G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," IEEE Transactions on Audio, Speech, and Language Processing, vol. 16, no. 2, pp. 467–476, 2008.

[2] A. Mesaros, T. Heittola, and T. Virtanen, “Acoustic scene classification: an overview of DCASE 2017 challenge entries,” in 16th International Workshop on Acoustic Signal Enhancement (IWAENC), Tokyo, Japan, 2018.

[3] Y. Han and K. Lee, “Convolutional neural network with multiple-width frequency-delta data augmentation for acoustic scene classification,” IEEE AASP Challenge on DCASE 2016 Technical Report, 2016.

[4] Y. Le Cun and Y. Bengio, “Convolutional networks for images, speech, and time series,” in The Handbook of Brain Theory and Neural Networks, M. A. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 255–258.

[5] Q. Kong, Y. Cao, T. Iqbal, W. Wang, and M. D. Plumbley, “Cross-task learning for audio tagging, sound event detection and spatial localization: DCASE 2019 baseline systems,” arXiv:1904.03476, 2019.

[6] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in the 2015 International Conference on Machine Learning, Lille, France, July 2015, pp. 448-456.