CONVOLUTIONAL NEURAL NETWORKS AND X-VECTOR EMBEDDING FOR DCASE2018 ACOUSTIC SCENE CLASSIFICATION CHALLENGE

Technical Report

Hossein Zeinali, Lukas Burget and Honza Cernosky

Brno University of Technology, Czech Republic

ABSTRACT

In this report, the BUT team submissions for Task 1 (Acoustic Scene Classification, ASC) of the DCASE-2018 challenge is described. Also, the analysis of different method performance on the development set is provided. The proposed approach is a fusion of two different Conventional Neural Network (CNN) topologies. The first one is the common two-dimensional CNNs which mainly is used in image classification task. The second one is one dimensional CNN for extracting embeddings from the neural network which is too common in speech processing, especially for speaker recognition. In addition to the topologies, two types of features were suggested to be used in this task, Mel-spectrogram in log domain and CQT features which explained in detail in the report. Finally, the outputs of different systems are fused using a weighted average.

Index Terms— audio scene classification, i-vectors, convolutional neural networks, deep learning

1. INTRODUCTION

Sorry, I didn't have enough time to write the introduction.

2. DATASET

In this work, the DCASE2018 acoustic scene classification challenge data was used. The dataset consists of 10 scene classes and was recorded in six large European cities and also different environments in each city. In the development set of the dataset, each acoustic scene has 864 segments which mean 8640 audio segments in the development set. The evaluation set also was collected in the same cities and has 3600 audio segments. Each segment has an exactly 10-second duration which achieved by splitting the larger recorded audio from each environment. The dataset includes a predefined fold. Each team can also create its own folds, but we used the single original fold for evaluation. The audio segments are 2-channels recording at 48000 Hz sampling rate.

3. FEATURES

In this work different features are used in single and multichannel modes. All features are extracted using zero mean audio signal. The main features are Mel-spectrogram. For extracting this feature, first short time Fourier transform is computed on 40 millisecond Hamming windowed waves with 20-millisecond overlap using 2048 point FFT. Next, the calculated power spectrum is converted to 80 bands Mel scales features and finally is transformed to the logarithmic scale. The second used feature is 84-dimensional constant-Q transform of the audio signal. This feature is extracted using librosa toolbox [1].

We used the features in two modes, single channel and 4 channels. In single channel mode, the audio signal first converted to mono and a single channel feature is extracted from it. In the 4 channel mode, 4 features are extracted from the signal similar to [2]. Two features from left and right channels, one feature from the summation of both channels (i.e. $L + R$) and one feature from the subtraction of both channels (i.e. $L - R$). Here, in contrast to [2] we use these 4 features as a single input to the CNNs.

4. METHODS

We have suggested two different CNN topologies for this challenge. The first one is the common two dimensional CNN which mainly is used in image processing tasks and the second one is a one dimensional CNN which mostly is used in speech processing fields for extracting neural network embedding which called *x-vector*, especially used in speaker recognition. Both networks are explained in more details in the following.

4.1. Two-Dimensional CNN

We followed the common network proposed in [3] with some modifications. Table 1 shows the network architecture. In general, the network contains 3 CNN blocks. The first layer is a two-dimensional convolutional layer with 32 kernels with $7 * 11$ kernel size and unitary depth and stride in both dimensions. This layer follows with a batch-normalization and *Rectified Linear Unit (ReLU)* activation. The next layer is a max-pooling layer operating over $2 * 10$ non-overlapping rectangles which feed a dropout layer at the end of CNN block. The output of this blocks feeds to the next block and so on. At the end of the third CNN block, there is a 2-dimensional global average pooling which by a batch-normalization layer. Finally, the last layer of the network is a Dense layer (fully connected) with 10 nodes and softmax activation function.

4.2. One-Dimensional CNN: x-vector Topology

The x-vector topology consists of one-dimensional CNNs in the time direction. Table 2 shows the network architecture for this case. The network has three subparts. The first one just works in frame level. The second part is statistic pooling compressing frame level layer to one statistic layer. In contrast to CNN used in image processing, here both mean and variance are used as extracted statistics. The last subpart of the network is segment-level consisting two Dense layer which follows with a Dens softmax layer like the previous topology. This network was used in two manners. First,

Table 1: 2-Dimensional CNN topology. BN: Batch Normalization, ReLU:Rectifier Linear Unit.

Input $80 \times 500 \times 1$ or $80 \times 500 \times 4$
(7×11) Conv2D(pad=1, stride=1)-32-BN-ReLU (2×10) MaxPooling2D Dropout (0.3)
(7×11) Conv2D(pad=1, stride=1)-64-BN-ReLU (2×5) MaxPooling2D Dropout (0.3)
(7×11) Conv2D(pad=1, stride=1)-128-BN-ReLU (5×10) MaxPooling2D Dropout (0.3)
GlobalAveragePooling2D BatchNormalization
Dense-10-SoftMax

Table 2: 1-Dimensional CNN topology - x-vector. BN: Batch Normalization, ReLU:Rectifier Linear Unit.

Input $500 \times 80 \times 1$
(3×1) Conv1D(pad=1, stride=1)-128-ReLU-BN Dropout (0.15)
(3×1) Conv1D(pad=1, stride=1)-128-ReLU-BN Dropout (0.15)
(5×1) Conv1D(pad=1, stride=1)-128-ReLU-BN Dropout (0.15)
(1×1) Conv1D(pad=1, stride=1)-128-ReLU-BN Dropout (0.15)
(1×1) Conv1D(pad=1, stride=1)-256-ReLU-BN
Statistic Pooling, Mean and Variance
Dense-128-ReLU-BN Dropout (0.15)
Dense-128-ReLU-BN
Dense-10-SoftMax

the softmax output of the network is used like before. In the second case, the embeddings which extracted from the first segment level affine transform are used for training another classifier. Here, Cosine distance is used which is too common in speaker verification. So, for each class, the average of embeddings of each class is used as class representation and cosine distance is used to measure the similarity of each test vector with different classes. For more detail about x-vector we kindly refer the reader to original paper [4].

5. SYSTEMS AND FINAL FUSION

In this challenge, we used different systems to make the final fusion results. For each feature type, one two-dimensional CNN is trained with the single channel features. Also, one more CNN is trained using 4-channels features. So, for each feature type we have two networks. In addition to these, one x-vector CNN is trained for each feature by using one-channel features. So, we have 6 networks with softmax classifier. The x-vector CNNs also are used to extract neural network embeddings, which means two more systems which used cosine distance classifier.

We trained these systems in two scenarios, the first one using the data without any augmentation and the second one using augmented data. Finally, we have 16 systems (8 for each scenario) should be fused to form the final submission. The combination of data-augmentation and also using cosine distance based system or not, constructs 4 different final submission. The output of different systems is fused by weighted average using FoCal Multi-class toolbox [5].

6. REFERENCES

[1] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*, 2015, pp. 18–25.

[2] Y. Han and J. Park, "Convolutional neural networks with binaural representations and background subtraction for acoustic scene classification," DCASE2017 Challenge, Tech. Rep., September 2017.

[3] R. Hyder, S. Ghaffarzadegan, Z. Feng, and T. Hasan, "BUET bosch consortium (B2C) acoustic scene classification systems for DCASE 2017," DCASE2017 Challenge, Tech. Rep., September 2017.

[4] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 165–170.

[5] N. Brümmer, "Focal multi-class: Toolkit for evaluation, fusion and calibration of multi-class recognition scorestutorial and user manual," *Software available at http://sites. google. com/site/nikobrummer/focalmulticlass*, 2007.