ENSEMBLE OF CONVOLUTIONAL NEURAL NETWORKS WITH TRANSFER LEARNING FOR AUDIO CLASSIFICATION

Bongjun Kim

Dept. of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL, USA
bongjun@u.northwestern.edu

ABSTRACT

This technical report describes models from our submissions for *Making Sense of Sounds Data Challenge*. The task is to build a system to classify audio files into 5 different classes. Our model is a Convolutional Neural network (CNN)-based model which consists of 6 convolutional layers and 3 fully-connected layers. We applied transfer learning to training the model by utilizing VGGish model that has been pre-trained on a large scale of a dataset. To boost the performance, we used ensemble techniques by combining results from models that are trained on a different set of training data.

***Index Terms*—** Audio classification, Convolutional neural networks, ensemble, transfer learning

1. INTRODUCTION

This technical report describes our approach to the *Making Sense of Sounds Data Challenge* [1]. The task of this challenge is to build a system to classify audio files into 5 different classes: Nature, Human, Music, Effects, and Urban. As a training set, 1,500 audio files are provided with their labels. Since the 1,500 is a quite small number for model training, we perform transfer learning by using a part of an existing pre-trained CNN model. We also use ensemble technique to boost the performance of the system given the small number of training examples.

2. DATASET

The training dataset contains 1,500 audio files of 5 different classes: Nature, Human, Music, Effects, and Urban. On top of the 5 classes, they also have fine-grained labels which represent specific sound type information. However, we did not use the additional sound type labels to build our models. All audio files are 5 seconds long and they are single-channel 44.1 kHz, 16-bit WAV files.

3. MODEL ARCHITECTURE

Table 1 shows the architecture of our model. It consists of 6 convolutional layers (Conv) and 3 fully-connected layers (FC). It takes a mel-spectrogram of a 5-second audio as an input representation and its output is class probabilities of 5 audio classes. In the table, filter sizes and the number of channels of convolutional layers are represented as *Conv (width × height, the number of channels)*. MP indicates Max-Pooling operation with a kernel size of 2 and a stride of 2. The layer-1 to layer-6 were taken from VGGish model [1] which has been trained on 8M-YouTube dataset. The pre-trained model is publicly available [2]. At the end of convectional layers, Soft-max pooling operator [2] is performed over time-axis of the feature map from layer-6. Then its output is fed into a set of fully-connected layers.

Table 1: Model architecture. *MP: 2D-Max Pooling (kernal size: 2 × 2, stride: 2), *SP: Soft-max Pooling over time-axis

Layers	Components	Output shape
Input	Mel-spectrogram	498×64
Layer-1	Conv (3×3, 64) → Relu → MP	249×32, 64
Layer-2	Conv (3×3, 128) → Relu → MP	124×16, 128
Layer-3	Conv (3×3, 256) → Relu	124×16, 256
Layer-4	Conv (3×3, 256) → Relu → MP	62×8, 256
Layer-5	Conv (3×3, 512) → Relu	62×8, 512
Layer-6	Conv (3×3, 512) → Relu → *SP	1×8, 512
Layer-7	FC (1024) → Relu	1024
Layer-8	FC (128) → Relu	128
Layer-9	FC (5) → Softmax	5

4. TRAINING AND TESTING PROCEDURE

Audio files for training and testing are resampled to 16kHz mono. Each sample is represented by a log-scale Mel-spectrogram with 64 Mel bins, a window size of 25 ms and hop size of 10 ms. Given a 5-second audio file, the size of the input representation is 498×64.

We applied transfer learning when training our models. Layer-1 to layer-6 are initialized with parameters from convolutional layers of the pre-trained VGGish model. The three FC layers are randomly initialized. When training, the first three convolutional layers (Layer-1 to 3) are fixed (not updated) and rest of layers are fine-tuned on the training set. Cross-entropy loss and Adam Optimizer with learning rates of 0.001 were used.

[1] `http://cvssp.org/projects/making_sense_of_sounds/site/challenge/`

[2] `https://github.com/tensorflow/models/tree/master/research/audioset`

Since the size of the given training set is not large, we train models with 10-fold cross validation setup. We divided the training set into 10 folds where each fold has 150 examples (30 examples per class). We held out a fold as a validation set and trained a model on the remaining 1350 examples. A model was trained for 50 epochs and the model that shows the best classification accuracy on the validation set was chosen. This process was repeated over the 10 folds, which builds 10 different models.

5. SUBMISSION SYSTEMS

In order to predict labels of evaluation set, we used outputs from all of the 10 different models. We used two simple ensemble techniques. Given an evaluation audio example, 1) class probabilities from the 10 models are added up and the class with the highest probability is chosen as a predicted label, 2) the label predicted by a majority of the models is chosen (i.e.majority voting). We submit both models for this challenge.

6. REFERENCES

[1] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, *et al.*, "Cnn architectures for large-scale audio classification," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 131–135.

[2] J. Salamon, B. McFee, P. Li, and J. P. Bello, "Dcase 2017 submission: Multiple instance learning for sound event detection," Tech. Rep., 2017.