CONVOLUTIONAL NEURAL NETWORKS SYSTEM FOR THE MIREX 2017 AUDIO CLASSIFICATION(TRAIN/TEST) TASKS

Huanmin Xu
Sichuan University
xuhuanminzj@hotmail.com

Jiancheng Lv
Sichuan University
lvjiancheng@scu.edu.cn

Bijue Jia
Sichuan University
bijue_jia@163.com

ABSTRACT

In this submission system, we used the Melspectrogram transform to preprocess the original music audio. We proposed a randomized segmentation approach to generate training samples. Randomly dividing the audio into the segmented data and randomly combining them can help us get a large number of non-repeating new data. For the training model, we utilized a three-layer convolution neural network(CNN). After running a series of experiments, the CNN parameters such as kernel sizes and learning rate were set carefully.

1. INTRODUCTION

Audio auto classification is one kind of problems in the field of music information retrieval. It consists of various research topics, attracting a large number of scholars studying it. In contrast to traditional machine learning methods, deep learning is becoming more and more popular in the study of such problems. In recent years, Convolution neural network (CNN) has excellent performance. Because of its high degree of freedom, good robustness, superior performance, CNN is widely used in all areas of deep learning.

Music classification problems can be divided into two modules to discuss, one is the preprocessing of audio data, the other is the selection of training models. Many articles have studied these two issues from various aspects [1, 2]. Also in the previous MIREX tasks, a variety of new ideas and techniques has been tested.

In our system, except the above modules, we find a more effective way to generate the data. Our training model is no longer limited by the amount of data and become more robust. This idea can help the system get better result without changing the training model and data preprocessing method. We describe in detail in the following sections.

2. OUR SYSTEM

Our system is based on a convolutional neural network and Mel spectrograms. The following subsections introduce our system in details.

2.1 Audio Preprocessing

With the knowledge of signal processing and the experience of others dealing with music signals, we used the mel spectral transform to preprocess the original audio. Because our data is monaural, we can directly do the Fourier transform. After getting the spectrogram, we applied a Mel filterbank to it. This can make the sound spectrum more in line with the human ear hearing habits. Finally, we took the logarithm, converted the unit into decibels. The *Librosa* library was adopted to implement the above steps.

2.2 Train Data Generation

We used a new randomization method to generate training samples. For the same audio, we randomly intercepted several 120-dimensional data on the time axis of the spectrum. Then we concatenated these data segments as a new piece of data sample and put into our training set. After the experiment, we believed that 120-dimensional data can almost represent the music genre, and the combination of multi-segment data can also increase the error-tolerant rate of the preprocessing. By doing this, our system is not limited by the amount of raw audios and can get enough different data for training. A large number of randomized elements also greatly enhance the robustness of our training samples.

We think this is a very good way to create training samples. It can also have more improvement such as combine same genre songs rather than one song, add more fragments to combine and so on.

2.3 Convolutional Neural Networks

Our model is a three-layer CNN. We used three convolutional layers, two pooling layers, two full-connection layers. The convolution kernel shapes of the three convolution layers are 10×12, 3×4 and 2×5, and the numbers of convolution kernel are 30, 30, 60 respectively. We used max-pooling in two pooling layers, in which filter shape are 2×8 and 2×2 respectively. On the boundary of the matrix, zero-padding are adopted.

In all convolutional layers and full-connection layers, we applied Exponential linear units (ELUs) [3] . We also added Dropout [4] in the full-connection layer in order to avoid over-fitting.

Through experiments, we set a learning rate of 0.0002 and it will exponential decay during the training. The batch size and train step were also set carefully.

2.4 Testing

A testing sample was gotten as same as the training sample and we randomly generated 15 samples in each songs. We finally voted for the genre which was selected the most.

3. IMPLEMENTATION

Our system is implemented in *Python 3.5.2* using libraries such as *librosa*, *numpy* and *datetime*.*Tensorflow 1.0.0-rc2* is used for deep neural networks designing.

4. REFERENCES

[1] Kereliuk C, Sturm B L, Larsen J.: Deep Learning and Music Adversaries[J]. *IEEE Transactions on Multimedia*, 2015, 17(11):2059-2071.

[2] Sigtia S, Dixon S.: Improved music feature learning with deep neural networks[C]. *IEEE International Conference on Acoustics, Speech and Signal Processing.*, IEEE, 2014:6959-6963.

[3] Clevert D A, Unterthiner T, Hochreiter S.: Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)[J] *Computer Science*, 2015.

[4] Hinton G E, Srivastava N, Krizhevsky A, et al: Improving neural networks by preventing co-adaptation of feature detectors[J]. *Computer Science*, 2012, 3(4):pgs. 212-223.