

INSTRUMENT CLASSIFICATION

Chen Zhang, and Ye He

University of Rochester
Department of Electrical and Computer Engineering

ABSTRACT

The main job of this article is constructing an effective instruments classification system to classify three instruments: Clarinet, Flute and Trumpet. This article is based on KERAS platform and MATLAB. The main work of this project is extracting the MFCC feature vectors of audio pieces and constructing LSTM model to learn the regularity of those feature vectors to classify those three instruments. We go through algorithms of MFCC and compare different ways of calculating the MFCC. Meanwhile, we also go through the basic structure of LSTM and we analyze the advantages of LSTM compared with simple RNN. The 85.6% accuracy is obtained as a result.

Index Terms — MFCC, KERAS, LSTM, Classification

1. INTRODUCTION

1.1. Overview

By the development of Internet, people are more easily to acquire huge amount of music data. Sometimes people are interested in some instruments' sounds but they do not know exactly what instrument it is. So based on the audio features of sounds and the development of machine learning technology, we decide to design an instrument classification model.

For an audio piece the most common way

2. METHOD

2.1 MFCC

Mel-Frequency Cepstral Coefficient (MFCC) is

to make a label is extracting the MFCC feature vectors of a piece of audio. Obtaining MFCC feature matrix for training model is the first step. The second step is constructing LSTM model. Then data will be fed into the network. Keeping adjusting parameters of the model to enhance the accuracy of the model is the last part.

1.2. Data Collection

In this project a large amount of data need to be collected to produce training dataset. Because the output of MFCC coefficients of an around 100ms to 200ms length of piece for each frame is essential, a long piece of music sound has to be divided into several small pieces to increase the independence of the training data and avoid breath influence to the training process.

1.3. Feature Extraction

After processing by MFCC, a 5*900 matrix raw data to represent each instrument's record is obtained. Then this training dataset will be used to train the model. In the training process, the construction of model and corresponding parameters is adjusted all the time to enhance the performance.

1.4. Goals

Our goals are achieving a relatively higher accuracy of classification of those three instruments and making the model more robust to classify some audio piece with noise.

the kind of coefficient that makes up Mel-Frequency Cepstral (MFC) which was created in 1980 by Davis and Mermelstein. It provides very successful features in tasks of speech recognition and speaker verification. Meanwhile, MFCCs are also finding uses in

music information retrieval applications.

To get the MFCCs, we follow the steps listed below:

1. Do Short-Time Fourier Transform on frames of audio samples to get the power spectrum of the sound.

Firstly, we divide our audio samples into several frames. In practice, we use 2048 samples long frame, around 50ms. Then we add hamming window and apply Fourier transform on each frame.

2. Apply Mel filterbank on each frame of the power spectrum and sum up the energy. Mel filterbank is a series of triangular band-pass filters equally distributed on Mel-frequency axis. Mel-frequency is used to approximately simulate the human's ear perception using Eq.(1) to obtain. For each frame, if we have K filters, we can get a K dimensional vector.

$$M(f) = 1125 \ln(1 + f/700) \quad (1)$$

3. Take logarithm on the values and do Discrete Cosine Transform. The 1st coefficient of MFC will be ignored for its unreliability, so we will get a (K-1) dimensional vector as the feature of this frame.

In our model, we choose 3 kinds of instrument: flute, clarinet and trumpet. For each one, a database of 180 notes is used to train the RNN. Part of the notes is collected from *The University of Iowa Musical Instrument Samples* while the other part is collected from Youtube videos.

The recordings from *The University of Iowa Musical Instrument Samples* have been made with single Neumann KM84 cardioid condenser microphone at 16/44.1. They were edited into chromatic scales played note-by-note at *pp*, *mf*, and *ff* dynamic levels. Some instruments were played with more than one technique, including arco, pizzicato, vibrato and non-vibrato. In practice, we choose *mf* vibrato, *mf* non-vibrato, *ff* vibrato and *ff* non-vibrato for each instrument.

For each note, we choose the 11th frame to 15th frame to calculate MFCC. As a result, a 21*5 matrix is created to show the features.

The performance of MFCC may be affected by several factors: the number of the

filters, the shape of the filters, the way the filters distributed and the way the power spectrum is warped.

According to our test, too many or too few filters do not result in better accuracy. The best number of filters should be chosen by experiment. In this case, we use 22 filters.

Traditionally, the shape of filters should be triangular. The shape can also be rectangular. Hermansky creates a particular shape of the critical-band curve given by (2) and illustrated in Fig.1(c).

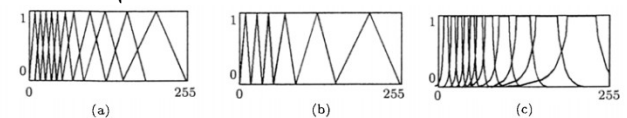
$$\Psi(B) = \begin{cases} 0, & \text{for } B < -1.3 \\ 10^{2.5(B+0.5)}, & \text{for } -1.3 \leq B \leq -0.5 \\ 1, & \text{for } -0.5 < B < +0.5 \\ 10^{-1.0(B-0.5)}, & \text{for } +0.5 \leq B \leq +2.5 \\ 0, & \text{for } +2.5 < B \end{cases} \quad (2)$$


Fig. 1

According to the previous research [2], there is no significant difference of the results with the 3 kinds of filters.

Meanwhile, there are two different kinds of distribution of the filters. One is overlapped filters (shown in Fig. 1(a)), the other is not overlapped (shown in Fig. 1(b)). After experiment, overlapped filters always lead to better result. In our model, we use 22 triangular overlapped Mel filters.

2.2 Pre-processing and LSTM

There are two parts for LSTM training. First is preprocessing part and the second part is LSTM network. For the first part:

1. Raw data needs to be divided into many batches. The advantage of doing this is avoiding the influence of outliers, for when the model process every 10 batches (set by us), the values of loss function calculated by outliers will be smoothed.
2. Normalization has to be done to limited data in the same scale, which will enhance the accuracy and efficiency.
3. One hot key label should be added to comparing probability of a test instance

whether to be one of those three instruments.

After preprocessing, training data will be fed into the network.

For the second part, LSTM is a special Neural Network. Basic structure of Neural Network is -- input layer, hidden layer and output layer. For every layer there are several nodes and for each node there is an activation function which is the math foundation of how a node processes the input. Some common activation functions are “tanh”(Fig.2), “relu”(Fig.3), and “logistic”(Fig.4) function.

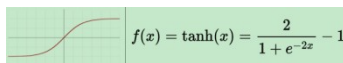


Fig.2

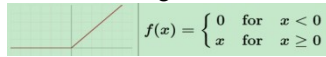


Fig.3

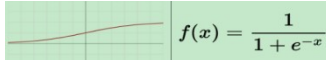


Fig.4

These activation functions are used to map data into nonlinear domain to mimic the electricity value changing in human brain. In this project, Relu is used because it will lessen the influence of gradient vanishing. As the function graph shown above, relu function can keep a constant slope which will be used in the gradient calculation later and this constant value will keep the gradient value larger than zero.

Weight is another essential element of a neural network. It is learned from training dataset by back propagation. After the network obtains a value, it will compare it with real value and calculate the result of loss function. More specifically an error value is calculated for each node, which will be fed into loss function to calculate the error gradient which finally will be fed into the optimization method to update the weight value.

For RNN nodes in hidden layer are not independent. Thus it can process input with sequential property and because of its relationship with other nodes, it was always used to process time sequence. LSTM (shown in Fig.5) is a special RNN.

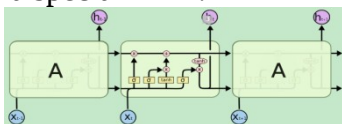


Fig.5

The main difference of LSTM is that it has a chain like structure, but the repeating module has a different structure. Comparing with RNN, Instead of having a single neural network layer, there are four, interacting in a very special way^[4] (Fig.5). In the structure LSTM can decide whether to store, update and combine data from the front nodes and how to assign different weights for each elements in the input vectors. The Fig.6 below shows the typical architecture of a single LSTM memory cell^[5]:

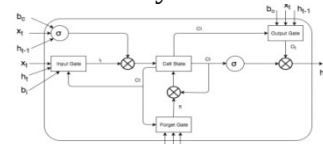


Fig.6

LSTM can avoid gradient vanishing problem which affects weights update and in the end will affect the accuracy. So that is why LSTM is so popular and has better performance than Simple RNN. Our testing result is satisfying and our accuracy is shown in Fig.7 The final result will stabilize around 85.6%.

tune cost: 1.08521640301	tune accuracy: 0.466666666667
tune cost: 0.467112421989	tune accuracy: 0.833333333333
tune cost: 0.451586902142	tune accuracy: 0.855555555556
tune cost: 0.498083919287	tune accuracy: 0.844444444444
tune cost: 0.538256168365	tune accuracy: 0.833333333333
tune cost: 0.569069623947	tune accuracy: 0.833333333333
tune cost: 0.597509026527	tune accuracy: 0.833333333333
tune cost: 0.626045763493	tune accuracy: 0.844444444444
tune cost: 0.652533650398	tune accuracy: 0.855555555556
tune cost: 0.68151307106	tune accuracy: 0.855555555556

3. CONCLUSION

In this paper, we have discussed the way of using MFCC feature matrix as the features of sound pieces to realize an instrument classification model. For the MFCC part, we have compared different ways of calculating MFCC, including changing the number of the filters, the shape of filters, and the way the filters distributed. As a result, we choose 22 triangular overlapped filters as our Mel filterbanks. To further improve the result, we could try to use the derivatives of the MFCCs to incorporate the time-domain features.

After reading some thesis about the structure of Network, for our project, if we want to improve the accuracy besides collecting more data to train our model there is an another method to realize the goal --- add an another model, which used to evaluate the derivative

MFCC coefficients and feedback the result to the original model.

In the future study we will also try different parameters of the model, extract more frames for an instance to highlight the strength

of LSTM and most importantly we will enlarge our dataset to make the training process more reliable to try to obtain a higher accuracy.

REFERENCE

- [1] Bob L. Sturm, Marcela Morvidone, and Laurent Daudet, "Musical Instrument Identification Using Multiscale Mel-Frequency Cepstral Coefficients," *18th European Signal Processing Conference*, Aalborg, Denmark, Aug. 23-27, 2010.
- [2] Zheng Fang, Zhang Guoliang, and Song Zhanjiang, "Comparison of Different Implementations of MFCC," *J. Comput. Sci. & Technol.*, Vol. 16, No.6, pp 582-589, Nov. 2001.
- [3] Music Instrument Samples, University of Iowa Electronic Music Studios, <http://theremin.music.uiowa.edu/MIS.html>
- [4] colah (Aug. 27, 2015) Understanding LSTM Networks <http://colah.github.io/posts/2015-08-Understanding-LSTMs/.html>
- [5] A. Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.

