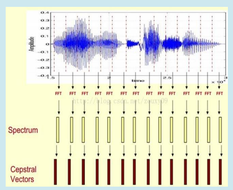
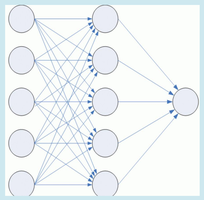
Instruments Classification

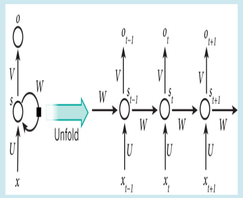
Chen Zhang Ye He
University of Rochester

Based on MFCC feature extraction of audio record and RNN neural network technology, we finish the instruments calssfication task. Here instrumetns include Trumpet, Flute and Clarinet. We collect several music pieces of those instruments and divide them into different frames for which we do the MFCC analysis to obtain feature vectors .And after data preprocessing to modify the data format, we feed dataset to LSTM model.

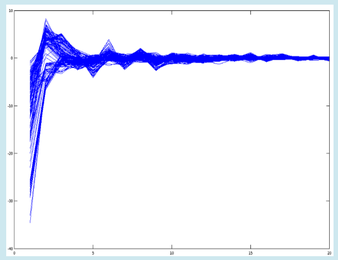
We train the model and evaluate the model with tuning dataset splitted from our raw data. Finally we obtain a relativly ideal result. For our testing data, we achieve a high accuracy. There is still room for us to imporve our result and we will do more research in the future.

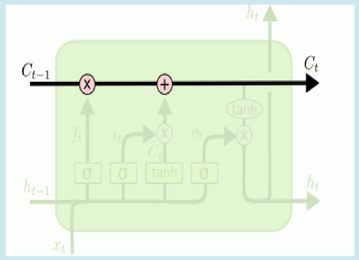




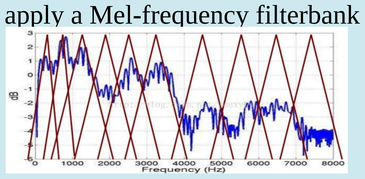


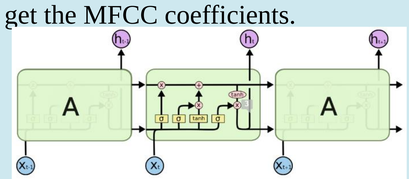
Our goal is classifying different instruments sound pieces by our model. In the end, if we have some music pieces of these three instruments, after inputing them to our model we can classify them into different groups which represents and shows corresponding instruments name and image.





We use MFCC coefficients to extract the feature vector of music pieces we collected. We take 1 frame of audio to do Fourier transform, and apply a Mel-frequency filterbank to get the MFCC coefficients.





Then we do the feature engineering for raw data coming from MFCC processing. After that, we construct our LSTM model to do the training ,use the tuning dataset to evaluate our model and feed the testing dataset to see the result. In the last part we keep adjusting parameters of the whole model to try to obtain a higher accuracy.

MFCC part:

In our experiment, we choose 3 kinds of instruments: flute, clarinet and trumpet. For each one, a database of 180 notes is used to train the RNN. Parts of the notes are collected from *The University of Iowa Musical Instrument Samples* while the other parts are collected from Youtube videos.For each note, we choose the 11th frame to 15th frame to do MFCC. As a result, a 21*5 matrix is created to show the features.

Learning part:

Here is the evaluating result after multiple attempets. The model finally stablize in an accuracy of 85.5556%. And the test cost stablize in around 0.744

Test accuracy means when we feed our tune data to our model, the number of ratio of results after classification to the real results. And the tune cost is value of loss function.

```
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
```

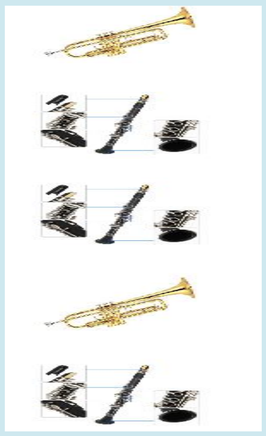
This is the test result:

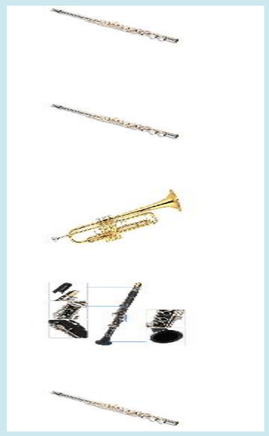
Here we just obtain a batch as test data to verify our model and the real label of this batch in sequence is:

(denote Trumpet as T; Clarinet as C; Flute as F)

T C C T C F F T C F

As shown below the test rsult is perfect.





Difficulty and Conclusion

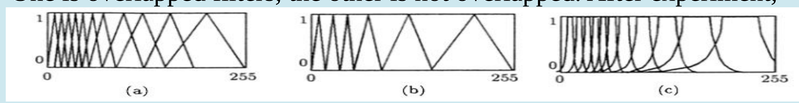
We have compared different MFCC methods:

The performance of MFCC may be affected by several factors: the number of the filters, the shape of filters, and the way the filters distributed.

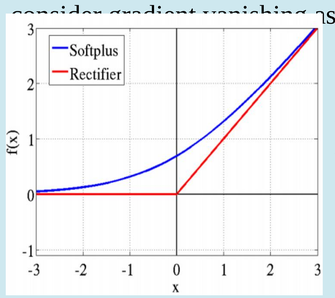
Too many or too few filters do not result in better accuracy. The best number of filters should be chosen by experiment.

Traditionally, the shape of filters should be triangular. The shape can also be rectangle. Hermansky creates a particular shape of the critical-band curve based on Bark frequency. There's no difference between the results.

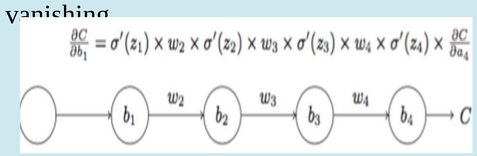
Meanwhile, there are two different kinds of distribution of the filters. One is overlapped filters, the other is not overlapped. After experiment,



Preprocessing the raw data can improve the accuracy and make the code run more efficiently,which including, normalization, one hot key, radom distribution and splitting data.We also have compared SimpleRNN model with LSTM model. LSTM has better performance in classifying. For LSTM, its memory function has obvious benefit in training process. Though it is a little bit time-comsuming but, it really improves the accuracy. Thus in the later resaerch of this kind of project, we have to consider gradient vanishing as a important problem.



Relu function can keep the gradient as a costant number ; using LSTM is another way to minimize the affect of gradient vanishing



Future Work

After reading some thesis about the structure of Network , for our project, if we want to imporve the accuracy besides collecting more data to train our model there is an another method to realize the goal --- change the structure. In the future study we will try to perfect the model.

And we will try different MFCC algorithms to try to obtain more effective methods of feature exatraction to enhance the accuracy.