Music Classification with Neural Network

Bai Ke
Department of Physics, Tsinghua University
baik13@mails.tsinghua.edu.org

Wang Hao
Department of Physics, Tsinghua University
haowang13@mails.tsinghua.edu.cn

Abstract

Music classification can be applied to many areas. The conventional methods mainly based on feature extraction. For those methods. The better feature, the better performance. However, as a kind of naturally produced process, there may exists hidden features which Convolutional neural network can find out and do the latter classification tasks. In this paper, we introduced LSTM, CNN and RCNN as new tools for music classification. Although this network hasn't been carefully tuned up to now, it has already reach the average level. We use two different evaluation criterion to evaluate our model.

1. Introduction

The conventional methods for music classification can be concluded into three parts: High-level, which is the conceptual meanings perceived by human-beings, Mid-level, which is the combination of low levels, like Mel-scale, Bark-scale. These features have similarities to human's hearing. Low levels is a Statistical description of signal that can be learnt by machines, including Fourier Transform, STFT Spectrogram, Spectral Centroid. In this paper, we use the low level feature "STFT" Spectrogram.

We introduced two different methods, LSTM and CNN for solving this problem. Using small window length to extract information from the dataset, we divided each sample into several parts. Rely on this, we enlarge our database to 10000 30000 samples. Our max accuracy is 50 80.4% now. Although our accuracy doesn't reach the highest level, but we have to mention that the length of our sample is 1s 2s. This task is much harder than single entire song classification. If we prepare to merge the regional feature to a vector and use statistical methods to judge a complete sample. 100% accuracy can be reached.

Another problem worth discussing is that if the music can be classified. One piece of music usually includes many genres and styles. In our task, we only pay attention to one-to-one correspondence.

The following content was divided into three parts. In the first section, we'll give a quick review of two models we use in paper. The second part explains the data transference. Then we give a detailed explanation of the two model: one is the LSTM network, the other is CNN network. In each section, we first introduce our model and data preparation, followed by a discussion of our final result. At the end of the article, we have a comparison of these two models.

2. Backgrounds

2.1. DataSet

The dataset we choose is GTZAN, 23% researches used this dataset before 2014[5].This dataset has 10 classes. Each class owns 100 samples. Each sample lasts 30s. Up to now, the highest classification rate is 92.24%.[4]

2.2. LSTM

LSTM(Long-short Term Memory) model is widely used to process time sequence. It is used to substitute the usual neurons in RNN. The structure is shown in Fig.1.

The neuron is controlled by three gates and a memory unit(C_t in Fig.1). The input gate(i_t), forget gate(f_t), output gate(o_t), are decide by both input and the memory unit by a sigmoid function. The speciality of these gate is that the gate is used to multiply the signal in there processes. The there are input(using input to update the memory unit), forget(using the previous value of c_t to update the memory unit) and output(using c_t to calculate h_t). When the gate is set 1, the signal will pass the gate totally and when the gate is set 0, the signal will not pass.

Different neurons may perform differently. If input gate is 1 and forget gate is 0, the neuron loss all its memory. Otherwise, if input gate is 0 and forget gate is 1, the output is constant. This features can keep its "memory" longer compared to traditional RNN. The network is able to "remember" both long term and short term information.

Music is one of the best material to apply LSTM. The works using LSTM on music focus on many different tasks, including Blues improvisation [2], note transcription [1], vocal gender recognition [6]. The last research dealt with s similar classification task.

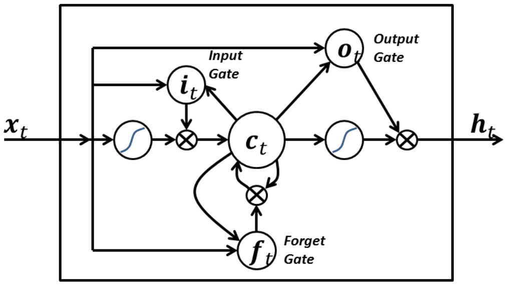

Figure 1. The structure of LSTM unit

2.3. CNN and RCNN

CNN has reached a good performance in image classification, audio classification. It can automatically extract the hidden information from data. Comparing to the conventional methods, it can abstract the high level feature from the raw data directly without manually extract. Convolutional Neural Networks (CNN) are biologically-inspired variants of MLPs. From Hubel and Wiesels early work on the cats visual cortex, we know the visual cortex contains a complex arrangement of cells. These cells are sensitive to small sub-regions of the visual field, called a receptive field. The sub-regions are tiled to cover the entire visual field. Although the assemble cell is not found in our ear. But it doesn't bother that we use this strategy to do our classification.

For improving the performance. We also imply the R-CNN model to this task. Comparing to CNN methods, the RCNN introduces recurrent units in a single layer. RCNN was proved to have better performance than conventional CNN [3]. It aims to simulate the information process better. Since the higher feature effect the lower fewer classifier in a special way, which may combine the advantages of CNN and LSTM.

3. Methods

3.1. Data

Conventional Convolutional neural network use picture as its input. But the sound is a time-sequential array. If we use the amplitude of the sound, the sample exists many noise and it's hard to train. We use a Fourier transform to transfer the data into frequency domain. The fourier transform on the entire song loss the sequential information. So we adapt SFTF.

3.2. Data Preparation

STFT is used to process the original .wav files of the dataset, trasforming them into spectrum for the following reasons. First, the features such as tunes and beats can be easily extracted from the spectrum. Second, the spectrum can be easily changed to adapt CNN and RCNN. Third, by changing the parameter of STFT, data can be compressing to any size, which simplifies subsequent training.

3.2.1 For CNN and RCNN

The window of STFT is set to be 1024 points, which is about 46ms. Compared to 25ms in traditional vocal analysis, music focus more in melody and chord. And a larger window can also extract the features while compress the data at the same time. And the overlay between windows is 3/4, that is, distance between each nearby points is about 11ms.

The number of filters is set 33 which is large enough to recognized the genre and also fit the network. The whole piece(about 30 seconds long) is cut to 15 smaller pieces, each around 2 seconds.

The input data contains 15000 sample, 1500 for each genre. Every sample is transformed into a 33×166 matrix.

3.2.2 For LSTM

The original .wav files is first transformed into spectrograms by using STFT with 2048 points. Thus, the length of time of the spectrograms in shortened to 600 700. we only use 600 points.

And then the frequency is compressed from 1025 points into 72 points by using the method below: The first 16 points in F is preserved, the $2^k + 1 \sim 2^{k+1}$th points $i = 3 \sim 9$ is compressed into 8 points by separate the points into 8 groups uniformly and use their average value. This process is used to compress the high-frequency part but also keep the high-frequency part unchanged, and preserved the information of the both.

Now each file is saved in a 600×72 matrix. The each piece is divided into 30 pieces with same length. And the final form of data is 20×72 matrix, and 3000 matrix for each genre, 30000 in total.

3.3. Model Construction

3.3.1 CNN

Two models are trained to compare the efficiency, one is a light one, the other is heavy, comparative speaking. On the one hand, If the model is too simple, convergence is hard to be met and it has poor performance. On the other hand, if the model is too large, overfitting phenomenon is serious and I have to enlarge the dropout rate several times.

Based on the frequency data, two groups data are prepared for training, one is the integer save as .lmdb, the other group is double type. It turns out to be the type of data has no obvious distinction in the following training task. Two models were adopted to train this net.

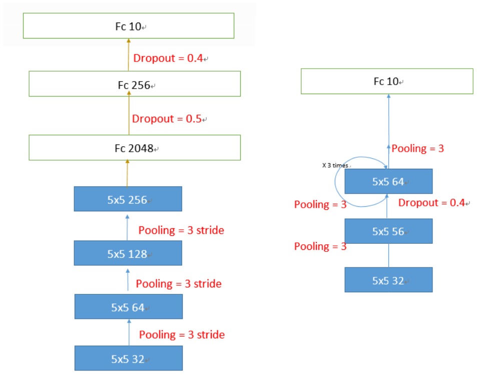

Figure 2. The left one is the complex CNN model, the blue lines means relu and pool, the red lines exists dropout. The parameters are marked beside the line. The right one is RCNN model. In this model, 3 recurrent layers are used with 2 convolution layer.

One of the most serious problems in this task is nonconvergent. For different situation, iteration step size should be appropriate to adapt to the model itself. Since the range of raw data is large. The value of blobs is easily to be out of bound. When training, the step size should be kept small at first for convergence, large later for better accuracy.

3.3.2 Simple CNN

This net only have 2 convolution layer with 2 fc-layer. This network have poor performance. This tells us the net's proper size is also necessary.

3.3.3 Complex CNN

The model uses 4 convolution layer, each with a 5x5filter. At the end, this model seems to perform well. However, this model is an over-fitting model. Drop out layers are put inside the fc layer. The drop out rate is normal(0.4). After 3000 iterations, the overfitting phenomenon begin to show its bad influence. The gap between the training lose and testing lose show up. Then I change the drop out rate(0.6 even larger). The over-fitting effect disappear, the accuracy begin to increase. Then it's overfitting again. In my first experiment, I change the drop out rate manually. In theory, it seems not to be a good method. I shouldn't use such a complex model for this data(the data size is about 1/2 of cifar 10 data, if we treat a pixel as an unit). Teacher suggests me to begin trying on a small size network. However, after many useless attempts on the small size network. I gave up. It seems that only large amounts of parameters can give the net's generalization ability.

The large amounts of parameters is not only the dominator to lead the performance, but also the culprit of over fitting. In my case, every time I change the dropout rate. The training loss has a temporary decrease and back to the norm value swiftly, which means the fc layer has a good performance fighting against the noise. Accordingly, the testing right increase. Maybe the connection weight of testing sample are trapped into the local minimum. The giant change of weights inspire the better connection.

After 10000 iteration, the right rate reaches 80%.

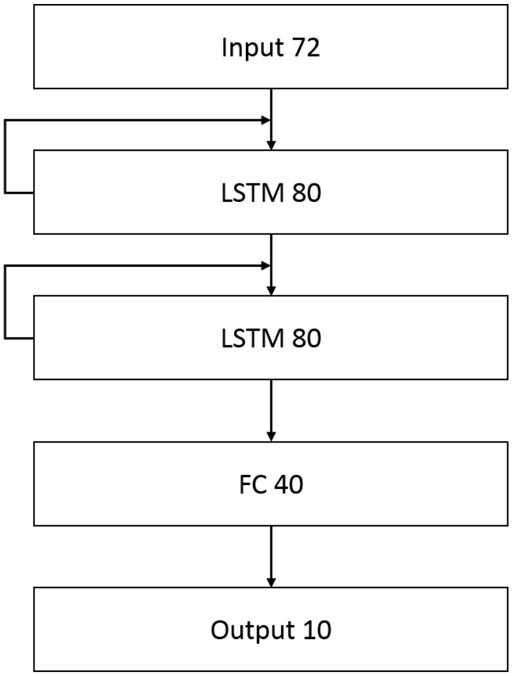

Figure 3. The structure of LSTM network
72 input neurons, followed by 2 LSTM layers with 80 neurons, a 40-neuron hidden layer and output layer for classification

3.3.4 RCNN

This is a Recurrent Convolutional neural network. In this net, recurrent units are put into this net. The proposer of the net hope to use this structure help the information exchange between the higher layer and lower layer. However, unfortunately. Although this network seems to be simple. Overfitting still exists in this network. After 200 000 iterations, the training right rate reaches 99%, the testing right rate is still 82.45%.

3.4. LSTM

The network is composed by 2 LSTM layers and 2 fully-connected layers, as is shown fig 3.

The sample has 72 points in frequency, and that is the input of the network. The first 2 LSTM layers deal with that frequency 20 times. The two layers each has 80 hidden neurons. Then the results pass the two fully-connected layers with 40 and 10 neurons. The first FC layer Rectified Linear

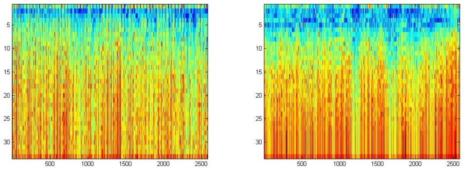

Figure 4. The left picture is the spectrum of classical music, the right one is disco's. The red squares in these two different spectrum seem to have the same feature

function as the activation function and the second is followed by Softmax loss function. The labels is transformed into 10-dimension vectors with 1 and 0 and calculate Softmax loss with results of the network.

3.5. Another Judgment Criteria

Worry about the poor performance for a long time. Finally, when I glance over the original raw data picture. I find out the answer to my puzzle. This problem start when I cut the whole music into pieces.

The left picture is the spectrum of classical music, the right one is disco's. Although this two kind of music have different music style. In detailed places. Their features may be in a same pattern as the picture show.

At first, we treat each 2s' clip as a sample. Now, each sample become a voter to the judgement of the whole piece music.(We divide the music into 15 pieces before). Here we make a rule: when half of neurons have the same label, then the music can be recognized as this kind of music.

4. Result

4.1. LSTM

Train the network 500 rounds and the result in illustrated in fig.5. It is shown that the accuracy on test data stop growing around 0.55 and that is the accuracy we finally got.

The details of result is shown in Table.1. Three of the genres (Classical, Metal, Pop) can be well recognized while other genres are not. This imbalance in accuracy reveal that the network it self cannot divide these genre in its mapping space. Another choice is to add more neurons into the network, but actually it will not help, the accuracy stay at the same level. So this result is the best this kind of the network can do.

The results also show that similar kind of music turn out to be classified into the same genre, for example, Rock and Metal, Hiphop and Pop. The similarity will reflect on little difference in spectrum. The results of LSTM network illustrate that only when genres have large differences, can the features be obvious enough to be recognised by the network. If the classification task is simpler, for example, 3 or 4 total different genre is to be labeled, the network will perform much better. However, it is not enough for this task.

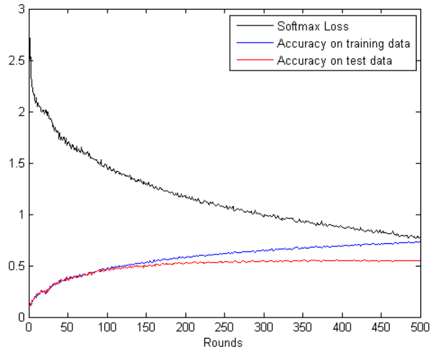

Figure 5. The result of training of the LSTM network

Black line is loss function in last Softmax layer. Blue line and Red line are accuracy on training data and test data.

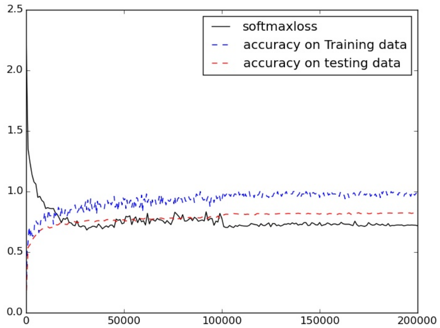

Figure 6. The result of training of the RCNN network the blue color means the training accuracy and loss, the red line means the testing loss and accuracy

4.2. RCNN

As we can see, in picture 6, this net convergent quickly. After 25000 iterations, it keeps steady. When it meets 100000 iterations, a little step shows up. At this point, we change our step size from 0.0001 to 0.00001. But there isn't significantly change.

But we can learn from both the RCNN and LSTM model, these data seems to quickly arrive the stage where they have 50% classification accuracy. It seems that the data itself causes this result. So we try a new judge criterion.

In the "vote model", we got the accuracy up to 98%. It seems that this network is too complex to adapt to this data. More experiments are done to give a detailed explanation. The detail is shown in Table.2

Using this evaluation criterion, we get 80.5% with 4000

LSTM Results	Original Label									
	Blues	Classical	Country	Disco	Hiphop	Jazz	Metal	Pop	Reggae	Rock
Blues	0.185	0.013	0.081	0.071	0.031	0.077	0.040	0.000	0.036	0.128
Classical	0.098	0.852	0.055	0.036	0.022	0.289	0.002	0.020	0.071	0.037
Country	0.098	0.010	0.129	0.049	0.015	0.102	0.005	0.010	0.064	0.066
Disco	0.005	0.000	0.023	0.020	0.017	0.005	0.003	0.000	0.011	0.013
Hiphop	0.042	0.002	0.060	0.111	0.181	0.026	0.039	0.021	0.096	0.083
Jazz	0.248	0.077	0.163	0.037	0.019	0.190	0.007	0.026	0.105	0.043
Metal	0.245	0.005	0.096	0.213	0.218	0.008	0.872	0.002	0.053	0.314
Pop	0.008	0.028	0.192	0.357	0.395	0.171	0.017	0.869	0.313	0.206
Reggae	0.027	0.010	0.164	0.068	0.085	0.126	0.005	0.052	0.222	0.043
Rock	0.042	0.003	0.036	0.037	0.017	0.006	0.010	0.000	0.028	0.066

Table 1. LSTM network results by genres, accuracy of each layer, row is the music itself, the column is classification outcome. Classical ,Mental ,Pop reach the best performance

iteration	clip's accuracy	whole accuracy	threshold
500	32.29%	46%	5
4000	49.12%	80.5%	5
10000	60.32%	96%	5
10000	60.23%	82%	7
60000	70.46%	whole music	–

Table 2. the 1st to 4th column is the outcome of vote model, which we cut the music into pieces firstly, then they vote for the most likely one. The threshold is used to avoid the phenomenon that the elements is random, the biggest one is random selected. In the last column, the training sample is the whole music. So it only has one accuracy.

iterations. Another possibility is the combination of features induce this effect. However, when I put the raw data ,the total 30s music, into the network. After 4000 x 15 iterations(Because in former model, we divide one music into 15 pieces, which means they have more data to adjust their net) The accuracy is only 70.46%.

The reason is that if we put all the data together, the input matrix is too huge(33 x 2500). Problem raises that all features are mixed together. However, if we cut it into pieces, the time sequences between the pieces disappear though, the local character is well preserved, the voting strategy is used only to avoid the strange noise in the music. For example, the disco music exists part which is relaxed or a break in half part. If the whole piece is considered as a whole, these part may influence the final outcome seriously. However, if we divide them into pieces, these noise's effect will decrease. Only if it doesn't meet our demand, we just reject them. They won't make a influence on other parts of "good clips".

5. discussion and conclusion

In our model, we use two kind of methods, LSTM and convolutional neural network. In convolutional part. We treat a piece of music as several overlapping fragments and use the short-time fragment as a training or testing sample. Then the combination of these fragments vote for the most possible label. As we can see, different evaluation criterion make a huge influence on the final outcome. This is mainly induced by the data we trained itself. Finally, we raise the accuracy up to 98%. In future, we will have a try on a giant music database " millions song database" to further evaluate our model.

References

[1] S. Böck and M. Schedl. Polyphonic piano note transcription with recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 121–124. IEEE, 2012.

[2] D. Eck and J. Schmidhuber. Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 747–756. IEEE, 2002.

[3] M. Liang and X. Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375, 2015.

[4] Y. Panagakis, C. Kotropoulos, and G. R. Arce. Music genre classification using locality preserving non-negative tensor factorization and sparse representations. In *ISMIR*, pages 249–254, 2009.

[5] B. L. Sturm. A survey of evaluation in music genre recognition. In *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, pages 29–66. Springer, 2014.

[6] F. Weninger, J.-L. Durrieu, F. Eyben, G. Richard, and B. Schuller. Combining monaural source separation with long

short-term memory for increased robustness in vocalist gender recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 2196–2199. IEEE, 2011.