CS230

Music Tagging With Convolutional Neural Network

Xiao Fei Yu
SCPD Student
Stanford University
xfy@stanford.edu

Abstract

In recent years, the interest of using Deep Learning for Music Information Retrieval has drastically increased while traditional MIR techniques remain difficult, non-universal and sometimes proprietary. Genre classification using CNN has been widely studied with positive results. This paper aims to further that line of research by simultaneously predicting the genre as well as valence (mood) of the audio by using a multi-output CNN to learn the features of mel-spectrograms generated from the audio. Results show that having mood share the same architecture as genre can work quite well.

1. INTRODUCTION

With the ever increasing musical data in the world, the need for automatic Music Information Retrieval (MIR) is always growing. MIR not only helps in classifying and organizing music data but is also crucial in recommending new songs with less usage history to users. Traditional MIR techniques require extracting hand-crafted features from the original songs. This process requires a high level of expertise in audio engineering. It is often times difficult and time-consuming to design features, as different features are needed for different tasks. These traditional MIR models also lack universality and extensibility as features need to be designed differently and calculated separately for different tasks. Deep learning methods have become more popular in MIR research recently, as it allows for end-to-end models, more automated classifications, and the results can be quite effective in different fields. This paper looks to further that line of research by simultaneously predicting the genre as well as valence (mood) of the audio by using a multi-output CNN to learn the features of mel-spectrograms generated from the audio.

2. RELATED WORK

There have been several papers that focus on using deep learning for MIR. Almost all of these work use images generated from audio as inputs. The images generated are either Short Time Fourier Transforms (STFT) or mel-spectrograms. Zhang et al (2016) studied genre classification using a CNN model with three convolutional layers and three dense layers. They also employed average between max-pooling and average-pooling to provide more information to the higher level statistical layers. Their model takes

in a rectangular image as input while most others use square. Choi et al (2016) used between 4-5 convolutional layers and one dense layer in their model for genre classification and took square images as inputs. These two papers both reach around 85% in accuracy, which is about state-of-the-art. Choi et al also explored the use of using Recurrent Neural Networks which marginally increased the accuracy from 85 % to 87%. Not all studies can achieve this accuracy of course; Dong (2017) achieved 70% accuracy using only two convolution layers. Choi et al (2018) touched on the reason why mel-spectrograms should be preferred over STFT. Evidently, Mel-spectrograms are optimized for human auditory perception, where STFT data are compressed in the frequency axis, therefore are more efficient in size while preserving the most perceptually important information. However, this compression makes Mel-spectrograms not invertible back to audio while STFT is. Very few papers have studied mood classification using CNN. Liu et al (2017) used CNN to predict the emotions of the song over 18 different multi-label emotions and achieved 71% F1. This lack of studies might be due to the fact that mood labels are hard to collect and rely on. This paper relies on the Spotify's Valence metric for mood classification.

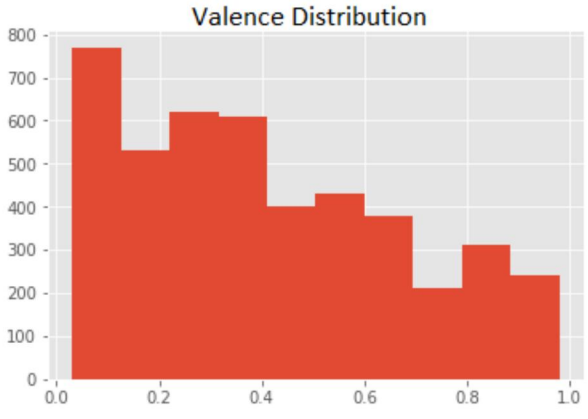
3. DATASET AND FEATURES

a. Collecting Raw Data:

This paper uses the data from FreeMusicArchives (FMA) as well as data queried via Spotify API. The FMA data set has 30 second audio samples for a variety of genres. The data is a lot less organized than the traditional GTZAN dataset that most MIR practitioners work with. The reason why I've chosen FMA data is because FMA data indicates the song name and artist for each song while the GTZAN data do not. Using the song title and artist name, I can query the valence metric from Spotify through their API. I focused on the following labels:

Rock, Pop, Hip-Hop, Instrumental, Electronic, Folk

These genres were chosen because they were the most abundant data in the FMA dataset. For example, Country and Blues had less than 50 songs each in the FMA dataset. I took 75 songs from each of the 6 genres randomly to ensure a balanced dataset. For each song I then look up their valence metric from Spotify. Valence ranges from 0 to 1 where 0 is the saddest and 1 is the happiest. Measures were taken to ensure that all the songs used actually existed in Spotify. The data set had the following Valence distribution:

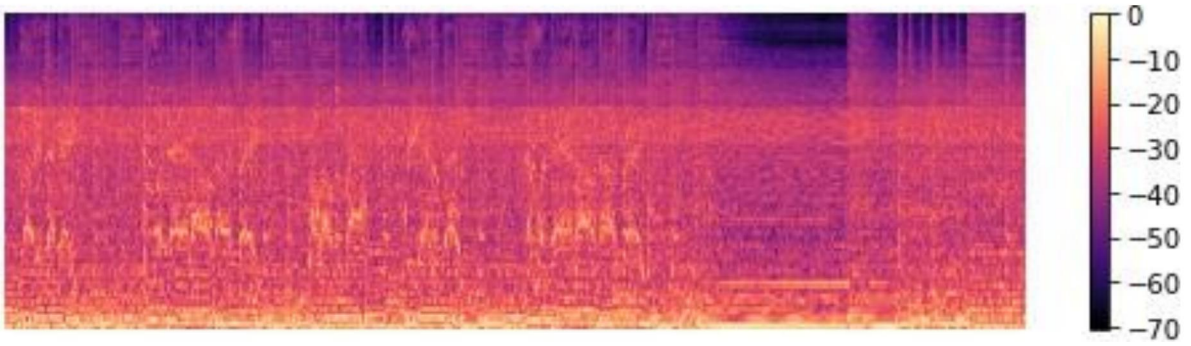


b. Data Processing

I decided to label the valence data into "Sad" where valence <.3, "Neutral" where valence >=.3 and <.6, and "Happy" where valence >=.6. I slightly skewed their division so there are more songs in the "Happy" label as songs in this data set tends to have lower valence scores. So now each song has a genre label as well as a mood label. The reason

why I labeled these is because running a multi-output CNN where both are categorical cross entropy is much easier to train than a CNN with one categorical and one regression output. Also the results would be easier to interpret. A categorical accuracy of X% is more intuitive for mood classification rather than a R Squared.

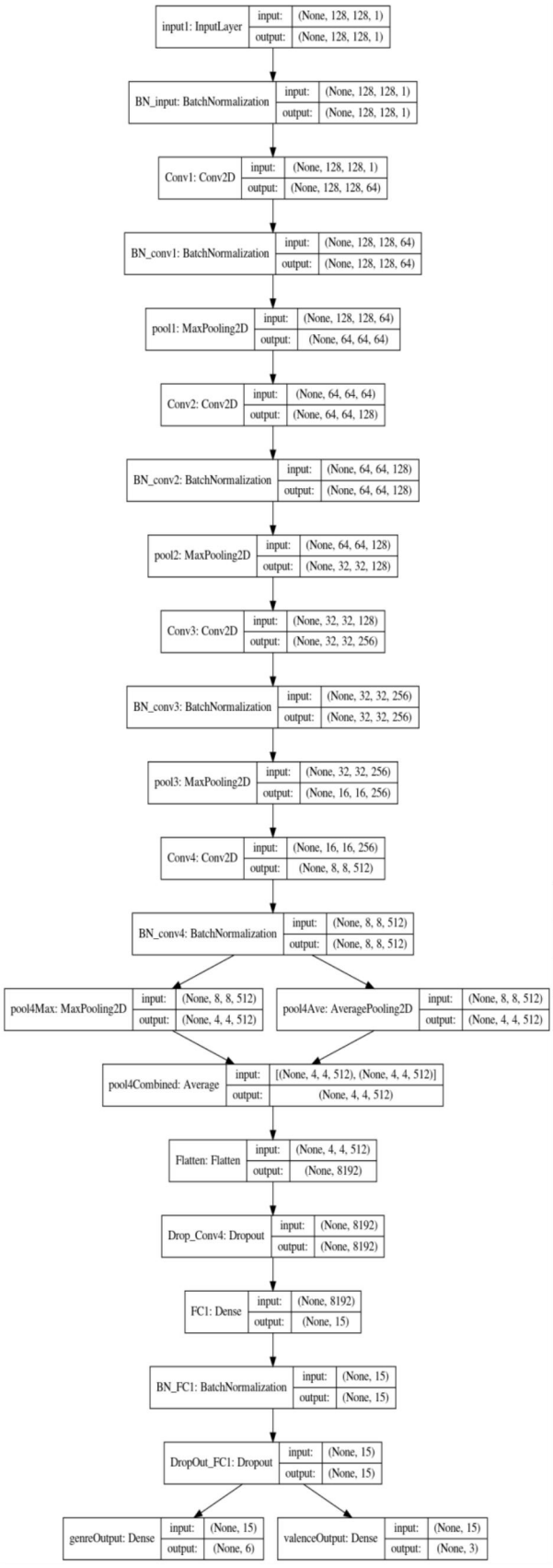
The audios are 30 seconds samples. I converted them into Mel-spectrograms as they are more perceptually intuitive and size efficient. Using the Librosa library, the Mel-spectrograms look like the following:



Where the X axis is the time dimension and Y axis is the frequency bins, there are 128 of them. These images have colour purely for visual presentation, the actual data is grayscale as colours don't carry additional information. I then divide these into 10 slices of 3 seconds each, which happens to turn the images into 128x128x1. This set up seems to work the best. I have previously tried 3 slices of 10 seconds each, believing that you need at least 10 seconds to determine the mood/genre of a song, but the model seemed to be overfitting, leaving me with 99% training accuracy, but 60% dev accuracy. The 3 second data set helps a lot with the over fitting problem, making the accuracy a lot more balanced in and out of sample. This leaves 4500 data samples. I used random number generator to randomly pick out 225 dev samples and 225 test samples (5% each). Leaving the dataset 4000 training, 225 dev, 225 test.

4. METHODS AND EXPERIMENTS

I used Choi et al and Zhang et al's models as baseline models as those performed the best in genre classification.I found that when applied to my dataset, Zhang et al's model was underfitting while Choi et al's was overfitting. This made sense as Zhang et al used 3 convolution layers while Choi et al used 5. So the ideal number of convolution layers I figured was 4. Trying out various numbers for the filter sizes, the ones that worked best were 64, 128, 256 and 512. As for kernel sizes, Choi et al suggested 3x3 kernels while Zhang suggested 1x1. Having tried various sizes, it turns out that 3x3 performed badly but 1x1 and 2x2 worked equally well, so I chose to use 2x2. For strides I tried various combinations and in the end chose 1, 1, 1, and 2 for the 4 convolutional layers. Using strides of all 1 performed the same and using strides of 2 for all layers was underfitting. Choi et al's model did something similar. I used one fully connect (dense) layer as any more would have been overfitting. Batch Norm was applied to all layers as it drastically increased training speed. I used a drop out rate of 50% for flattened layer and fully connected layer to help regularize the model. The two output layers both share the fully connected layer and each have nodes equal to the number of outputs.



I used averaging max pooling and average pooling as it increases accuracy.Please see the figure on the left for the detailed view of the model. A grid search was done to find the optimal learning rate, regulariation parameter and batch size.

LR	BS	L2	genre_acc	valence_acc
0.01	75	0.001	0.840	0.760
0.01	75	0.002	0.756	0.729
0.01	75	0.003	0.796	0.702
0.01	150	0.001	0.804	0.742
0.01	150	0.002	0.756	0.702
0.01	150	0.003	0.796	0.676
0.001	75	0.001	0.871	0.782
0.001	75	0.002	0.858	0.773
0.001	75	0.003	0.849	0.796
0.001	150	0.001	0.884	0.804
0.001	150	0.002	0.876	0.791
0.001	150	0.003	0.862	0.764

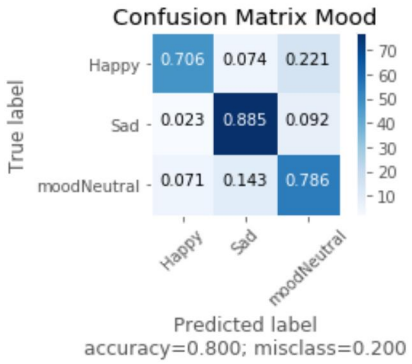
For optimal performance, I use learning rate of .001, batch size of 150 and L2 regularization of .001. For comparison against the base models in the test set:

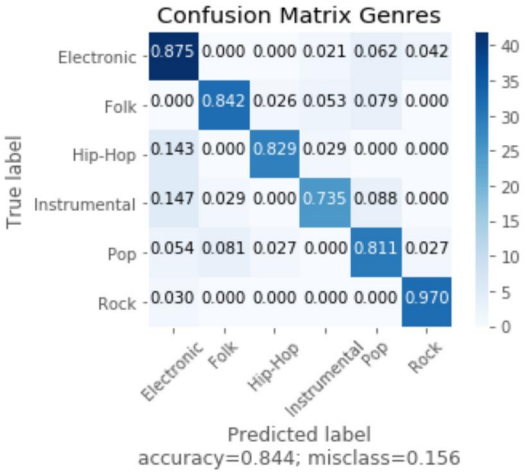
	Genre		Mood	
Model	**TrainAcc**	**TestAcc**	**TrainAcc**	**TestAcc**
Zhang et al	0.476	0.333	0.478	0.420
Choi et al	0.968	0.790	0.953	0.702
This Project	0.908	0.844	0.907	0.800

This project's model achieves a much better result than the base models.

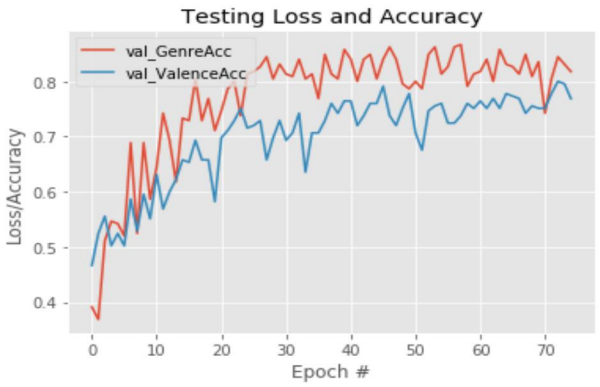
5. RESULTS AND DISCUSSION

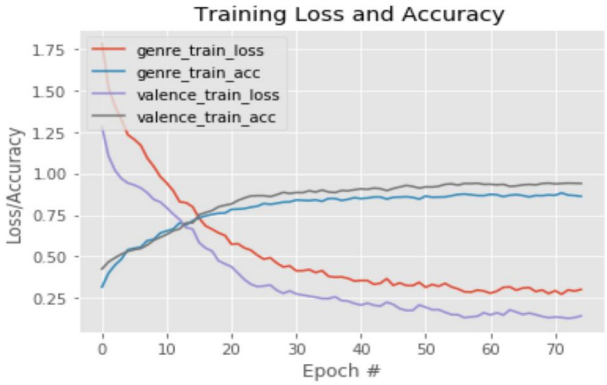
Accuracy is calculated categorically (number of songs categorized correctly divide by total number of songs). This measurement is more accurate compared to Precision and Recall for this specific task. The genre classification overall is on par with state of the art (~85%). Below we can see the prediction results in the test sample and Instrumental songs have the worst performance (73.5%) and are often mistaken as electronic. This makes sense logically as these two types of songs are similar, electronic songs tend to be





more instrumental. Happy songs and mood-neutral songs suffer with 70.6% and 78.6% accuracy respectively. This is because there are less happy songs in our data sets (only 25% are Happy while the optimal number would be 33%), expanding the dataset and balancing the mood categorization in future works can potentially fix this issue.





The loss and accuracy seems to be converging in both the training and the validation sets, which is what we want. Overall I think the prediction accuracy is satisfactory in both genre and mood.

6. CONCLUSION AND FUTURE WORK

This paper looked to expand on using deep learning for MIR, and tested the possibility of using one CNN model to produce more than one music tag by predicting the genre and mood from the same fully connected layer. Satisfactory results have been achieved in both predictions though can be improved with a better data set. One can extend on this project by extending the dataset and balancing the number of songs in each mood category to increase accuracy. Also, one can extend this model to also predict danceability, energy and other Spotify features. Having a CNN that predicts all these information end to end is much more efficient than dealing with the complexity of traditional MIR techniques and at the same time bypasses Spotify's proprietary non-public methods. It would also be interesting to examine what each of the 15 nodes of the Fully Connected layer represent, and whether this vector can be used for music recommendation, where songs with similar vectors are recommended.

7. CONTRIBUTION

All work was done by Xiao Fei Yu as there are no others on the team.

8. GITHUB LINK:

https://github.com/xfyu416/CS230

9. REFERENCES

Keunwoo Choi, Gyorgy Fazekas, Kyunghyun Cho, and MarkS, “A Tutorial on Deep Learning for Music Information Retrieval,” May 2018.

Weibin Zhang, Wekang Lei, Xiangmin Xu, and Xiaofeng Xing, “Improved Music Genre Classification with Convolutional Neural Networks,” September 2016.

Keunwoo Choi, Gyorgy Fazekas, and MarkS, “Convolutional Recurrent Neural Networks for Music Classification,” December 2016.

Xin Liu, Qingcai Chen, Xiangping Wu, Yan Liu, and Yang Liu, “CNN based music emotion classification”, April 2017

Mingwen Dong, “Convolutional Neural Network Achieves Human-Level Accuracy in Music Genre Classification”, Feb 2018

Christine Senac, Thomas Pellegrini, Florian Mouret, and Julien Pinquier, “Music Feature Maps with Convolutional Neural Network for Music Genre Classification, June 2017

Bob Sturm, “The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use”, June 2013