SongNet: Real-time Music Classification

Chi Zhang
Stanford University
czhang94@stanford.edu

Yue Zhang
Stanford University
yzhang16@stanford.edu

Chen Chen
Stanford University
chen2@stanford.edu

Abstract

In this work, we implemented and trained an end-to-end deep neural network, SongNet, to perform real-time music genre classification. Music can be represented in various forms: time-series decimals, spectrum in frequency domain and spectrograms, etc. The spectrogram stands out as the most popular choice since it incorporates time and frequency information. In this project, we used the convolutional recurrent neural network (C-RNN) to classify music. The convolutional network extracts features of spectrogram before feeding them into recurrent network which then performs classification considering both transient and overall characteristics of music. Taking only raw audio as input, the C-RNN achieved 65.23% accuracy on `fma-small` dataset, beating the best baseline by 41%.

1. Introduction

With the enormous growth of music released online, managing music library manually has become more and more challenging not only for users but also audio streaming service companies, such as Spotify and iTunes. Fast and accurate music classification is in high demand while it is non-trivial for machines to perform the task automatically at human level.

Besides, music genre classification is an essential backbone for music recommendation and unknown soundtrack recognition, which will benefit music service platforms a lot. Building a robust music classifier using machine learning techniques is essential to automate tagging unlabled music and improve users' experience of media players and music libraries.

In recent years, convolutional neural networks (CNNs) have brought revolutionary changes to computer vision community[9]. Meanwhile, CNNs have been widely used for music information retrieval, especially music genre classification[3]. Recently, it became increasingly popular to combine CNNs with recurrent networks (RNNs) to process audio signals, which introduce time sequential information to the model. In convolutional recurrent networks (C-RNNs), the CNN component is used to extract feature while the RNN plays the role of summarizing temporal features. The inputs of C-RNNs are soundtrack spectrograms and the outputs are probabilities of each genre at each timestep when performing real-time classification. In the training process, the genre is predicted as mean of all transient predictions over time.

2. Related work

Music genre classification has been actively studied since the early days of the Internet. Tzanetakis and Cook [7] used k-nearest neighbor classifier and Gaussian Mixture models with a comprehensive set of features. Those features could be summarized into three categories: rhythm, pitch and temporal structure. Support vector machine (SVM) was used by Mandel and Ellis [6] to classify music genre. Deshpande *et al.* [4] compared k-nearest neighbor, Gaussian Mixtures, and SVM to classify the music into three genres, which are rock, piano, and jazz.

In recent years, using audio spectrogram has become mainstream for music genre classification. Spectrogram encodes time and frequency information of a given music as a whole. Wyse [8] used spectrogram as input to train convolutional neural networks. Li *et al.* [5] built a CNN to classify music genre by using Mel-frequency cepstral coefficients (MFCCs) as features.

This work aims to train a C-RNN model with mel-spectrogram as the only feature, and compare this model with the traditional machine learning classifiers that need to be trained with hand-crafted features and metadata.

3. Dataset and Features

3.1. Free Music Archive [2]

The dataset used for this project is the Free Music Archive (FMA), an interactive library of high-quality, legal audio downloads direct by WFMU. Furthermore, it provides music's associated information including pre-computed features, user-level metadata, *etc.* To ensure data is balanced among different genres, we only use a small subset `fma-small` for the scope of this project. It con-

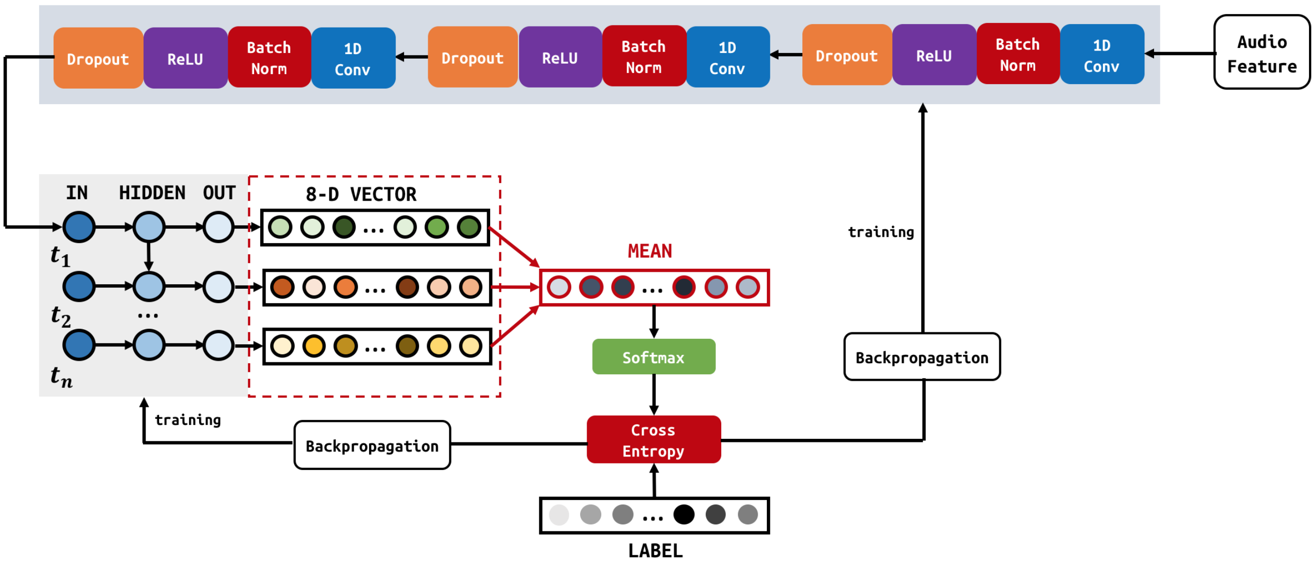

Figure 1. Convolutional Recurrent Neural Network (C-RNN)

tains 8000 tracks of 30-second clips with 8 balanced genres. Compared to the GTZAN Genre Collection released around 18 years ago, FMA is more updated and suitable in terms of genre completeness and audio quality.

The full FMA dataset includes 161 genres, unbalanced with 1 to 38, 154 tracks per genre and up to 31 genres per track. Considering limited computational resources and it is tricky to overcome class imbalance, we only used a small subset of FMA organized and selected by Michal Defferrard *et al.* [2], `fma_small`. The small subset contains 8000 30-second clips from top 8 genres, with 1000 clips per genre. The 8 genres are showing as follow:

Electronic	Pop
Experimental	Rock
Folk	Instrumental
Hip-pop	International

The FMA provided fine genre information for each track with built-in genre hierarchy, which is claimed by the artists themselves. In each of the track table, the ids of all the genres indicated by artists are included, and the root genres are provided in `genre_top` column.

The preprocessed dataset is split into 70% training, 20% validation, 10% test sets, respectively.

3.2. Features

A popular representation of sound is the spectrogram which captures both time and frequency information. In this study, we used mel-spectrogram as the only input to train our nerual network. A mel-spectrogram is a spectrogram transformed to have frequencies in mel scale, which basically is a logarithmic scale, more naturally representing how human actually senses different sound frequencies. It is simple to implement thanks to Librosa[1].

Aside from the music features extracted by Librosa, FMA also provides music metadata such as release year, number of listens, composers, durations, *etc*. There are 140 features in total that could be used for training.

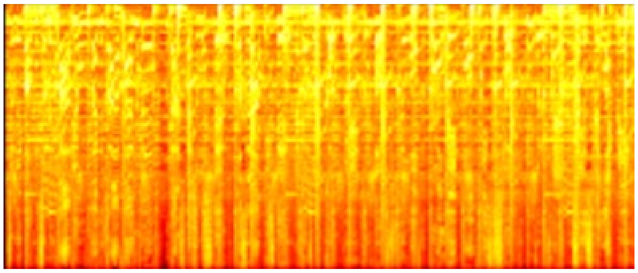

Figure 2. Spectrogram. Orange dots stand for the peaks of power over time (horizontal axis) and frequency (vertical axis). The brighter dots are, the more powerful.

4. Method

4.1. Baseline Classifiers

We trained four traditional classification models on the dataset as baseline classifiers, including k-nearest neighbors, logistic regression, multilayer perception, and linear support vector machine. It was found that baseline models could achieve no higher than 50% accuracy. Since these models were merely used for comparison, we adopted the default implementation and parameters in scikit-learn library. The input features include all 140 features provided by FMA.

4.2. SongNet Architecture

As shown in Fig.1, SongNet consists of three-layer convolutional neural network (CNN) which is followed by a recurrent neural network (RNN). The one obvious decision, which is present across related literature and our work is using convolutional layers to extract features from a song. The reason is straightforward: the network should not be constrained to use hand-crafted features but extract useful features as it needs. The output of CNN is a pretty long sequence in which every timestep strongly relies on both the immediate predecessors and long term structure of the entire song. To capture both transient and overall characteristics of a song, RNN becomes a natural choice. We initially used LSTM but found that `TimeDistributedLayer` in Keras worked better.

To start, features are extracted from the spectrograms using convolutional layers. It is important to point out that the features are translation-invariant only in time domain: frequencies do matter and needs to be distinguished. Thus, 2-D convolution seems unsuitable in this case: we are interested in changes across time - every convolutional layer should look at a small period of time as a whole, extract the most valuable information and create a feature map that is still a sequence over time. Then one dimensional convolutions across the time axis were adopted. Each convolution is followed by ReLU activation and 1-D max pooling. To regularize the model, we added Dropout to every convolutional layers.

The CNN outputs a sequence of features and it is then fed to RNN represented by a time-distributed fully connected layer with softmax activation, essentially giving us a sequence of 8-dimensional vectors (8 is the number of genres in `fma-small`) at each timestep. The RNN part is designed to find both dependencies across short period of time, and a long term structure of a song. These vectors are interpreted as the networks belief of the music genre at the particular point of time, *i.e.* probability distributions. To reduce the time series of 8-D probability vectors into a single one genre probability distribution, we simply take the mean. It is the most intuitive way to tackle the disproportion problem of inferring music genre per timestep versus just one label for the whole song, but it turns out to very effective.

5. Results and Discussion

5.1. Performance

The accuracies of baseline classifiers and SongNet are reported in the table below. It can be observed that our C-RNN model outperforms the best baseline by 41%. The validation set was used to help us tune hyperparameters of SongNet. During training, the learning rate was initially set to 0.001 and further decayed subject to `ReduceLROnPlateau` scheduler. The reported numbers are accuracies with respect to the test set.

Model	**Accuracy**
Random Guessing	0.1250
K Nearest Neighbors	0.3638
Logistic Regression	0.4225
Multilayer Perceptron	0.4488
Support Vector Machine	0.4638
C-RNN	**0.6523**

It is worth mentioning that all of our baseline models were trained and tested with “rich” features including music metadata (year, artist, *etc.*). However, in the current C-RNN model setting we decided not to incorporate metadata for simpler training setup. The fact that C-RNN model still beats the best baseline by a significant amount even without metadata demonstrates the power of deep learning models on classification tasks.

5.2. Error Analysis

To further interpret the results and guide future work, we plotted the confusion matrix (Fig.3) of SongNet. It is shown that 6 out of the 8 genres can be classified accurately. However, the model does not perform well on Experimental and Pop. Of particular interest is the Experimental genre. By definition, Experimental music is a general label for any music that pushes existing boundaries and genre definitions, be it in rock, jazz, modern composition or any other style. Thus it inherently contains features of many different genres. This can make it difficult to be classified correctly based on raw audio. Similarly, Pop has misclassification issues as well. By definition, pop music is a genre of music that is often regarded as the softer alternative to rock. We can then infer that these two genres must share features as well. Thus the model might be confused as well. One possible solution to these issue is to include music metadata for reasons to be discussed in Future Work section.

5.3. Kernel Clips

In computer vision, convolutional layers are used to extract features from images. Low level kernels can detect edges or corners and higher level kernels can capture more sophisticated structures. In our setting, we also expect convolutional layers to do similar things. SongNet has 3 convolutional layers so we expect kernels to extract different levels of music genre kernels. It would be straightforward and much clearer if kernel numbers are converted to music clips. After “listening” to some of kernels, we found that kernel clips in the first convolutional layer are mainly basic beats and elements of music. The clips from the last convolutional layer, however, are already human-listenable syn-

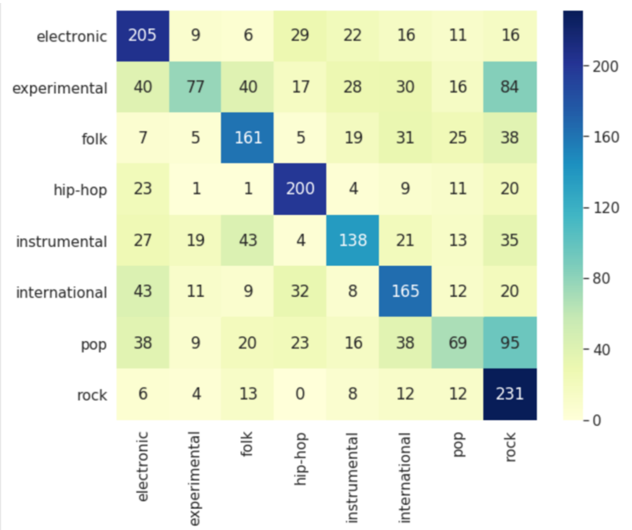

Figure 3. Confusion matrix on test set. Horizontal axis represents predicted labels of SongNet. Vertical axis represents the ground-truth. Diagonal numbers indicate correctly classified samples.

thesized music clips of certain genres. We demonstrated the kernel clips during the poster session, and uploaded them to Google Drive (link) for grading purposes.

5.4. Real-time

The ultimate goal of SongNet is real-time genre classification as the soundtrack plays. This is the reason why we combined recurrent network with convolutional neural network in our architecture. As discussed in the architecture section, for each timestep, the model outputs a probability distribution vector among 8 different genres so it enables real-time classification. It is better to show this functionality with a GUI. Due to limited timeline, we did not implement it yet, but it could be an interesting extension in future work.

6. Future Work

Following our discussion above, we conclude two possible extensions of current work.

1. To further increase the test accuracy, it is essential to solve the Experimental genre issue because it contributes a lot to the loss. It is worth trying to incorporate music metadata. We expect the metadata to help increase the performance because even though this music itself shares some features with other genres such as rock and electronic, additional information such as artists and album years will be able to help the model better classify this genre.

2. Build a graphical user interface to allow users upload a music clip and then visualize the real-time classification. This is fun as well as beneficial for further tuning the model. It's fun because users can have a better way of interaction with the model. As users upload more songs, we could collect more data to improve the model.

References

[1] Librosa. `https://librosa.github.io/librosa/`. Accessed: 2018-12-11.

[2] K. Benzi, M. Defferrard, P. Vandergheynst, and X. Bresson. FMA: A dataset for music analysis. *CoRR*, abs/1612.01840, 2016.

[3] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho. Convolutional recurrent neural networks for music classification. *CoRR*, abs/1609.04243, 2016.

[4] H. Deshpande and R. Singh. Classification of music signals in the visual domain. 2001.

[5] T. L. Li, A. B. Chan, and A. H. Chun. Automatic musical pattern feature extraction using convolutional neural network. In *In Proc. IMECS*, 2010.

[6] M. Mandel, M. I. M, and D. Ellis. Song-level features and support vector machines for music classification, 2005.

[7] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[8] L. Wyse. Audio spectrogram representations for processing with convolutional neural networks. *CoRR*, abs/1706.09559, 2017.

[9] Z. Zuo, B. Shuai, G. Wang, X. Liu, X. Wang, B. Wang, and Y. Chen. Convolutional recurrent neural networks: Learning spatial dependencies for image representation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 18–26, June 2015.