Genre-detection with Deep Neural Networks

Matt Jones, Daniel Way, and Yasaman Shirian

Abstract

With the emergence of online music availability, came a vast change in the preferences and distinguishability of listener taste with respect to musical genres. Pop music, for example, has seen a number of songs such as Marc Ronson's "Uptown Funk" borrowing 1970's funk sensibilities, or Avicii's "Wake Me Up" which fuses singer/songwriter and country instrumentation into a modern dance hit. The popularity of these songs demonstrate how genres can creatively borrow from each other to create invigorating songs. While much previous work has been put into music genre classification, very little have explored this emerging phenomenon more deeply.

Our model was able to represent constituent stylistic features of songs from spectographic representations of musical data from several sources. After analyzing and fine-tuning a previously built neural network architecture, we were able to create a novel convolutional neural network model that achieves 95% accuracy upon training for 10 distinct musical genres, with 82% accuracy of genre detection achieved on test data, providing an accurate depiction of genre influence in the songs examined.
(The code for this project is available at https://github.com/sniper-wolf-N7/cs230-ProjectSpace.git)

1 Introduction

Music genre classification is the process where a piece of music is recognized, understood, and differentiated by a conventional category as belonging to a shared tradition or set of conventions (Cohen and Lefebvre, 2005; Sadie, 1980). In this project, we were interested in working on genre-detection as a probabilistic distribution of various genres, as they pertain to the underlying styles of songs, albums, and artists. We start by using a neural network model created by [1] which has accuracy of 77% for tagging 10 genres and fine tune it to achieve higher accuracy. We aim to fine-tune model to detect 7 genres : pop, rock, classical, jazz, country, hip hop, and blues. After testing the previously described model, we then proceeded to build our own model encompassing best-practices we learned in studying the previous iteration, and incorporating 3 additional genres (reggae, metal, and disco ironically) in order to compare the accuracy achieved by both classifiers.

2 Dataset

For our dataset, we used multiple open source music datasets, including selected works from GTZAN [1], FMA(free music archive) [2]. The GTZAN dataset consists of 1000 audio tracks each 30 seconds long. It contains 10 genres, each represented by 100 tracks. The tracks all have a sample rate of 22050 Hz in Mono, 16-bit .wav format audio files. The FMA dataset consists of audio tracks with eclectic mix of genres beyond the genre features we were hoping to analyze for the project. This led to the FMA dataset being harder to train on for accurate analysis of genre-features represented in our

[1] http://marsyas.info/downloads/datasets.html

[2] http://freemusicarchive.org/

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

model. Input data for both models are the preprocessed mel-spectogram representation of the music data. These spectograms are vector sequences derived from the Fast Fourier Transform (FFT) of the raw audio signals in mel-scale. Recent works have been shown that with mel-spectograms better results is achieved in genre-detection.

3 Related Work

In the paper written by [1], an analysis was conducted to determine what type of architecture would best characterize genre amongst songs in the GTZAN dataset. Convolutional neural networks were desirable for their ability to efficiently analyze spectographic features of the audio files with a relatively low computational budget. We found this method to be favorable for feature extraction as well, as recurring patterns within the audio are represented well via intensity and lateral position within the spectographic images.

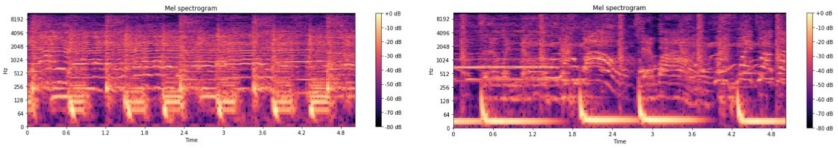

Figure 1: Comparison of Mel Spectograms in the rock (left) and hiphop (right) genres

Table 1 provides some methodologies and accuracies to compare to across the GTZAN dataset. After reviewing these works, we believe that aiming for an accuracy of $80 - 85\%$ would demonstrate an ideal method for genre-classification.

Table 1: Previous work for comparison to feature-detection CNN

Methodology	**Accuracy**
Non-machine learning:	
ADABOOST (Bergstra et al. 2006)	
	82.5%
Machine learning:	
CRNN (Jiménez and Ferran, 2017)	77.89%

4 Experiments

4.1 Jimenez Architecture

We have used the pre-trained weights from git [1]. This network is based on the primary architecture by [2]. Their network consists of 4 stacked CNN layers comprised of [CONV - Batch Normalization - MaxPool -Dropout] followed by 2 GRU layers with size 32 and a softmax layer at the end. However, after using the pre-tuned weights, we do not observe high accuracy as it is expected (60% on our personal music library). We focused on optimization methods (Adam vs SGD), number of freeze layers (freezing first 3 conv layers versus no freezing), and architecture (adding GRU versus LSTM). We provide 6 different models:

- Adam optimization with learning rate 0.001, batch size 16:
  - No freezing layer
  - first 3 Conv layer freezed + 4 GRU layers at the end
  - first 3 Conv layer freezed + 1 GRU + 3 LSTM layers at the end
- SGD optimization with learning rate 0.001, batch size 16:
  - No freezing layer
  - first 3 Conv layer freezed + 4 GRU layers at the end

- first 3 Conv layer freezed + 1 GRU + 3 LSTM layers at the end

Batch size of 16 was optimal in order to fit the data in memory. We also did explorations with learning rate(0.1, 0.01,0.001), which proved to have an insignificant effect. The assumption underlying this model is that CNNs on input side are useful for local feature extraction and then RNN is more useful for temporal pattern aggregation of the sound.
Input audio for fine-tuning process is 30 seconds long from both FMA and GTZAN, preprocessed to get 1366 frames for 96 mel bins. Note that for fine-tuned Jiménez model, we used 7 genres instead of 10 here for the sake of time and computational efficiency. However, the new proposed model in this project is able to tag 10 different genres.

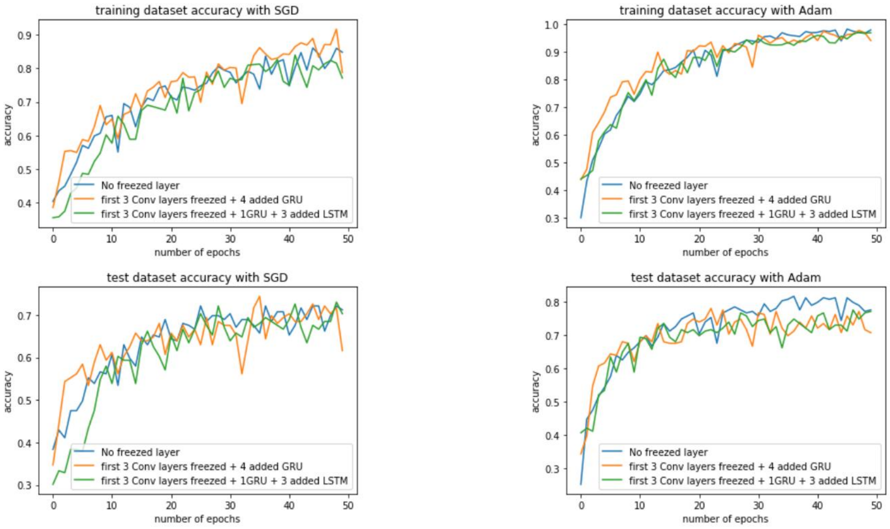

Figure 2: Accuracy versus number of epochs for different architectures and optimization methods

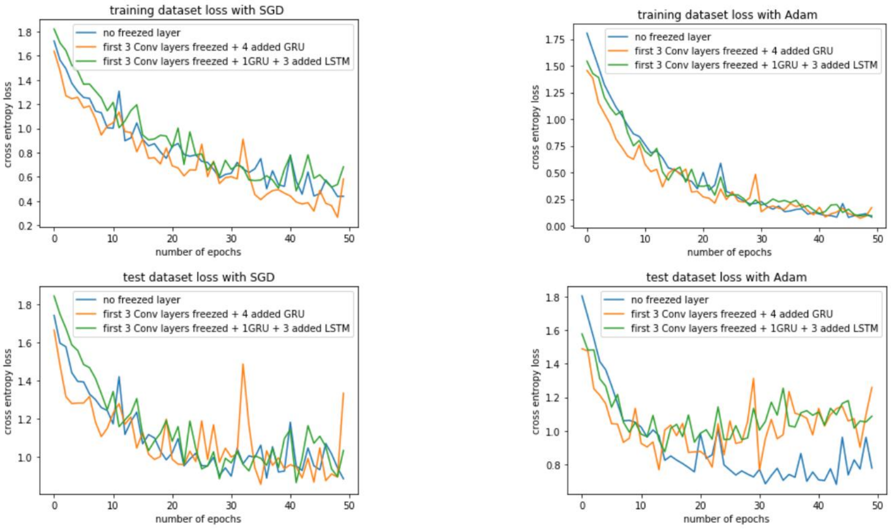

Figure 3: Loss versus number of epochs for different architectures and optimization methods

4.2 Novel Architecture

For our second experiment we built a CNN to predict genre using the GZTAN data. Our first goal was to overfit a small subset of the data to ensure we had a functional model and tune the hyperparameters to meet this goal. We started with an architecture of 3 stacked CNN layers comprised of [CONV

- Batch Normalization - MaxPool] followed by a 10 unit fully connected layer with a softmax for predicting the genre and Adam for optimization. We used an initial training set of 190 music samples and validation set of 30 samples (19 from each genre for training and 3 for validation). During this initial overfitting experimentation the key hyperparameter choices we made were related to normalization and the learning rate. Results can be see in Table (2).

Learning Rate	Normalization	Training Accuracy	Validation Accuracy
.0005	No	20%	20%
.0001	No	99.5%	43.3%
.00005	No	99.5%	33%
.0005	Yes	13.7%	13.3%
.0001	Yes	99.5%	33%
.00005	Yes	99.5%	40%

Table 2: Accuracy based on different learning rate

Following the overfitting experiment we decided to move forward with a learning rate of 0.00005 and to normalize the data. Next, our goal was to get the model to generalize better rather than overfitting the training data. Our first tactic was to use more of the dataset available. We gradually increased this to see the impact but even with including the full data set of 900 samples we only achieved 46% validation accuracy. We then decided to augment our dataset by cutting each 30-second sample into 3 10-second samples. This had two positive effects for reducing the high variance problem: first, it tripled our dataset size and second, it significantly decreased the number of parameters in our model in the fully connected layer (about 3x reduction). With this we were able to achieve 76% accuracy. (Table 3)

Clip Length	Train / Val Size	Paremeters	Training Acc.	Validation Acc.
30 sec	500 / 50	5,104,714	99%	38%
30 sec	900 / 50	5,104,714	99%	46%
10 sec	2700 / 150	1,664,714	96%	76%

Table 3: Effect of data augmentation

With 76% accuracy on validation without a full overfitting on the training data we turned our attention to once again creating a slightly more powerful model and then using hyperparameter tuning and architectural decisions to improve performance. Ultimately we were able to obtain 82% accuracy with a final architecture of 3 [CONV - Batch Normalization - MaxPool] layers, a 100 hidden-unit layer with 50% dropout and tanh activation, and a final 10-unit softmax layer. The full set of experiments in this last phase of experimentation can be seen in Table (4).

5 Conclusion

We have explored 3 different architectures with 2 different optimization methods for fine-tuning the model proposed by Jimenez [1]. The provided plots demonstrate the cross entropy loss and accuracy versus number of epochs for training and test dataset. At the conclusion of our development, we were able to achieve 85% accuracy for test set. This is higher than the original model, as well as the non-neural network classifiers we examined at the onset of this project. We believe that the richer mix dataset of FMA and GTZAN and improved architecture were among the factors in getting better performance. Looking at our results, Adam optimization method outperforms SGD optimization method, as higher train and test accuracy is reached with Adam optimization (95% and 85% respectively). It is particularly interesting to note that among architectures utilizing Adam optimization method where none of the layers are frozen during training, we reliably receive better

Additional Hidden Layer	Activation	Dropout	Num Epoch	Train Acc.	Val Acc.
10 units	relu	0%	10	70%	69%
10 units	tanh	0%	10	73%	71%
10 units	tanh	0%	20	81%	75%
10 units	tanh	0%	30	91%	77%
100 units	tanh	20%	30	99%	80%
100 units	tanh	50%	30	99%	82%
100 units	tanh	70%	30	99%	80%

Table 4: Fully connected and epoch tuning

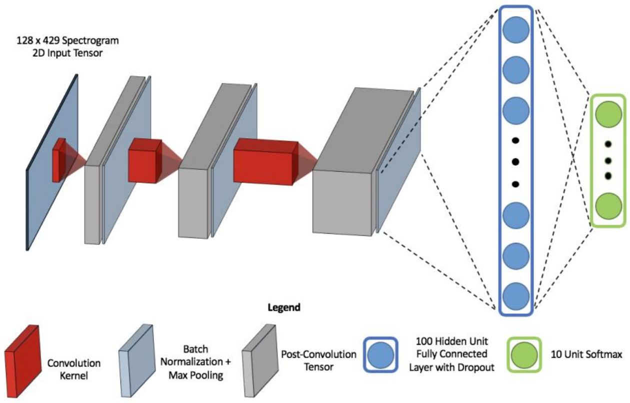

Figure 4: Final Novel Architecture

convergence in accuracy for our test set. The other interesting observation, is that for genre-detection, training better CNN layers have more impact on the accuracy of the network than adding more RNN layers the end of the architecture. Figure (3) and (2)supports this reasoning, as adding LSTM and GRU layers does not outperform the case with training the Conv layers without adding RNN.

6 Future work

With the wide variety of musical data becoming available online, research tools such as the neural network architecture developed in this project can serve more and more utility for both music researchers and the music industry as listener tastes continue to evolve, serving functions such as accurate categorization of metadata for music publishers, as well as preference matching for listeners of streaming music services. Training and testing the models on more contemporary music would better position such a tool for accurate analysis as new music continues to be released. Additionally, given larger amounts of data, larger architectures would be ideal for exploring further increases in accuracy. The project may also provide particularly useful information for transfer learning applications, where original musical works can be reimagined in novel forms.

7 Contributions

- Yasaman Shirian: Labeled FMA adatset and provided new mix dataset of GTZAN and FMA, fine-tuned the model with mixed dataset with Adam optimization method.
- Matt Jones: Developed novel model architecture, iterating through multiple versions and providing analysis on it.
- Daniel Way: Fine-tuned the model using the GTZAN dataset and SGD optimization method.

References

[1] A. Jiménez and F. José. Music genre recognition with deep neural networks.

[2] K. Choi, G. Fazekas, and M. Sandler. Automatic tagging using deep convolutional neural networks. *arXiv:1606.00298*, 2016.

[3] R. Ajoodha. Automatic genre classification. *The university of Witwatersrand, School of Computer Science*, 2014.

[4] E. Benetos and C. Kotropoulos. A tensor-based approach for automatic music genre classification. *Proceeding of the European Signal Processing Conference*, 2008.

[5] Dumitru Erhan Douglas Eck J. Bergstra, Norman Casagrande and Balázs Kégl. Aggregate features and adaboost for music classification. *Machine learning*, 2006.

[6] J. Cast, Ch. Schulze, and Ali Fauci. Music genre classification. 2014.

[7] H. Cohen and C. Lefebvre. Handbook of categorization in cognitive science. *Elseveir*, 2005.

[8] M. Ogihara T. Li and Qi Li. A comparative study on content-based music genre classification. *ACM SIGIR conference on Research and development in information retrieval*, 2003.

[9] S. Oramas et al. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval*, 1, 2018.