Genre Classification and Generation of Music Samples via Neural Networks

Laurel Doyle and Rebecca Alexandra Lopez

Abstract

Much exploration has been done into the application of neural networks to imaging tasks, but quite a bit less has been done in the field of understanding audio. This paper discusses the development of two deep learning models designed for the related purposes of generating music audio and classifying that audio into appropriate music genres. Both recurrent and convolutional network architectures are employed in these tasks. In this project, we utilize the GTZAN music data set for generating audio files of specified genres as well as for training of the genre classification network.

1 Introduction

Audio data, specifically music, presents an exciting challenge for neural network research. Due to the high-dimensional and artistic nature of music, applying machine learning techniques to the tasks of generating and classifying music results in incredibly complex models. However, understanding the nature of music data is a highly desirable goal for many businesses who make use of music recommendation algorithms, such as Spotify, YouTube, or Pandora. Thus, although music is difficult to model and classify using neural networks, it is still worthwhile to attempt this task.

In this work, we attempt to explore the features of music data that distinguish genres from one another. This is done utilizing the GTZAN dataset, which is freely available on Kaggle [3]. This dataset includes 1,000 thirty second audio clips of songs belonging to 10 different genres as well as feature tables and spectrogram image files for these clips. In our work, we create models using the WAV audio clips themselves, processed with the Librosa software package.

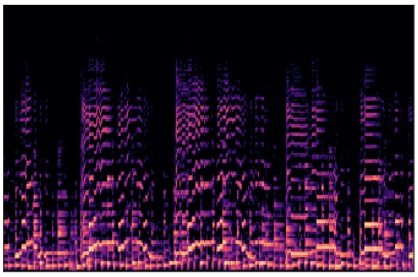


Figure 1: Example of mel spectrogram for a jazz song sample included with the GTZAN dataset

2 Methods

All neural network architectures described in the following were constructed using the PyTorch software library.

2.1 Music Generation Network

To accomplish the music generation task, we wanted to employ a recurrent neural network architecture directly to sequential audio data, similar to a prediction or forecasting task. Much of the research presented in other works utilizes spectrogram or wave plot images as the music generation mechanism rather than sequential audio data [2] [1]. However, we hoped to create a network that could generate audio of unspecified length, which would be a task best accomplished by a recurrent neural network.

The network architecture to generate music is based on the gated recurrent unit (GRU) architecture. This setup was intended to capture the nature of the entire sequential audio clips and thus produce the best approximation for the provided song. An LSTM layer was also tested for the task, but training proceeded more slowly than with the GRU implementation. In addition, the GRU structure is bidirectional in the music generation network. The intention behind this was to ensure information later in the sequence was utilized in the early steps of the song prediction.

Training of this recurrent neural network layer was aided by a one-dimensional convolution layer applied to the sequence before input to the GRU. This reduced the number of parameters in the model and thus sped up training.

The basic structure of training involved selecting 90 song samples from the 100 presented for each genre in the dataset. These 90 samples were then split into training and validation clips all of length 1 second. 29 seconds of each audio clip were used for training and 1 second was used for validation. The RNN model took the first 0.9 seconds of audio sample as input and produced 0.1 seconds of audio as output. This output was then compared to the following 0.1 seconds of real music in the training sample clip, and a mean squared error loss function was applied. All audio had a sample rate of 10,000 Hz, which is lower than the original sample rate of 22,050 Hz.

A separate network was trained for each genre, in order to only capture the important features of that genre and hopefully make outputs distinguishable during the classification task. Each network was trained for 1,000 epochs. A notebook containing the code for one of the genre's generation task can be found in additional materials. All other genres used the same parameters for the networks and training schema.

2.2 Music Classification Network

In order to classify the generated music clips, we convert our generated audio files into visual files as images are widely accepted and often used for classification purposes. Specifically, this network was trained on mel spectrograms depicting randomly selected clips of 5 seconds length from each of the 100 thirty second samples from each genre.

For this part of the project, we implement a convolution neural network. Specifically, our network feature fives convolution layers, each followed by a max pooling layer to keep the parameter count manageable. The setup can be viewed in Figure 3.

In terms of parameters, we found the Adam optimizer to work best with a learning rate of 0.0001. Higher learning rates resulted in overall lower accuracy, and smaller learning rates had resulted in requiring too high of an epoch count for our computing limits. Additionally, we introduce batch normalization to increase the stability and speed of our network. We utilized a dropout probability of 0.5 in order to counteract the over fitting done on the training data.

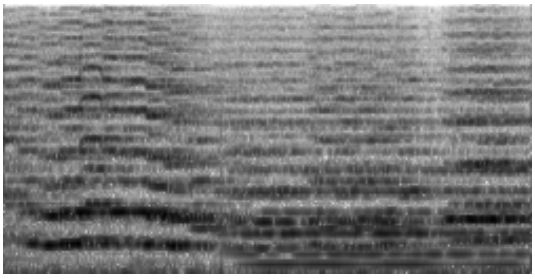


Figure 2: Example of mel spectrogram for a five second classical music sample utilized in training

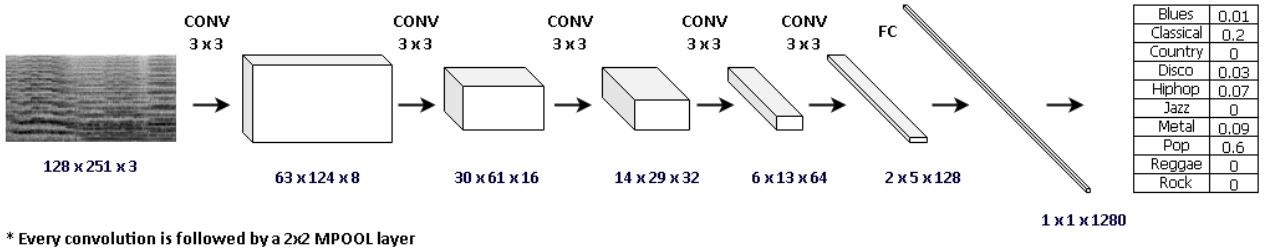


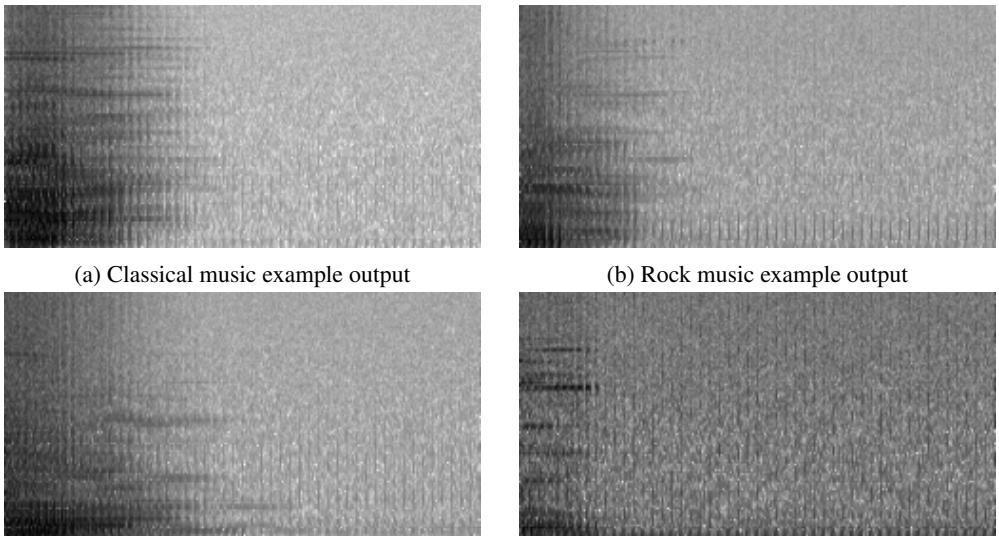
Figure 3: CNN architecture used for classification of mel spectrograms into their appropriate genres

3 Results

3.1 Music Generation Network

The primary products of the trained music generation neural networks are the mel spectrograms produced for each genre, which would be classified using the classification network described in our methods. These were generated using a 10 element test set selected from each genre of the GTZAN dataset, and left out of training. The clips generated are 5 seconds long, and were created using only a 0.9 second clip from each test file. The 5 second length was accomplished by repetitive generation of 0.1 second of music from the preceding 0.9 seconds of music (either already generated or from the seed sample).

Examples of these 5 second mel spectrograms are shown here:



(a) Classical music example output

(b) Rock music example output

(c) Metal music example output

(d) Jazz music example output

Figure 4: Resulting mel spectrograms from recurrent neural networks for music generation from each genre

One notices the choppiness in the spectrogram due to the iterative nature of the generation scheme. The auditory volume of the music-like structures in these spectrograms is low. This is best demonstrated using a wave plot as shown in Figure 5.

3.2 Music Classification Network

Confusion matrices for the initial validation set and for the test set of generated audio are shown in Figures 6 and 7 respectively.

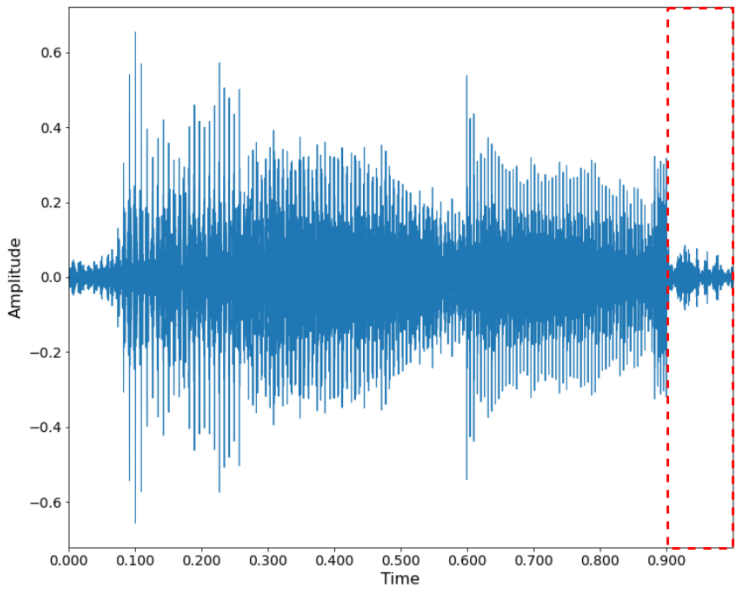


Figure 5: Wave plot of 0.9 second song clip used for training and 0.1 second generated output from the neural network, highlighted in a red box. Clearly, the amplitude of the generated output audio is much lower and this can be determined via comparison of the mel spectrograms as well.

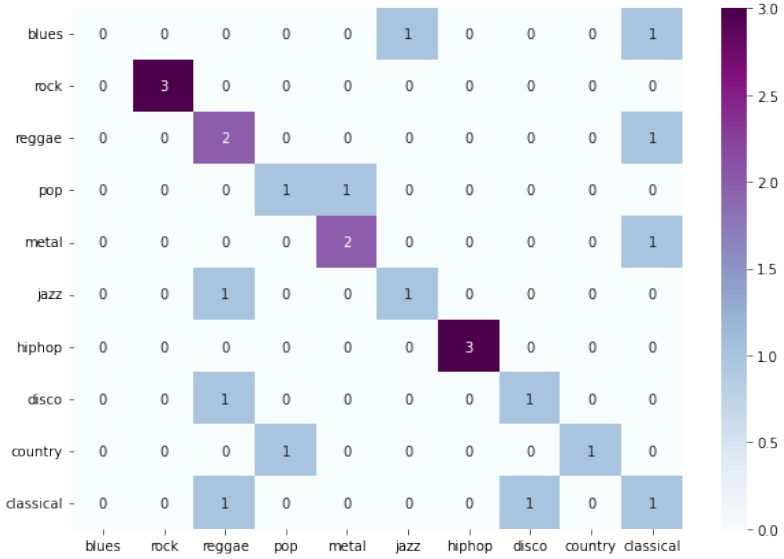


Figure 6: Confusion matrix for original test set from GTZAN database

As we can see in Figure 6, the convolutional network applied to the mel spectrograms from the GTZAN database song performs with satisfying accuracy, given these are 5 second clips.

However, the classification in Figure 7 could use some improvement. This likely due to the generation network performance issues described in Sections 3.1 and 4.

4 Discussion

The 10 networks for music generation trained with varying degrees of success. The neural network was able to produce audio with frequency features not dissimilar from those noticeable in the original

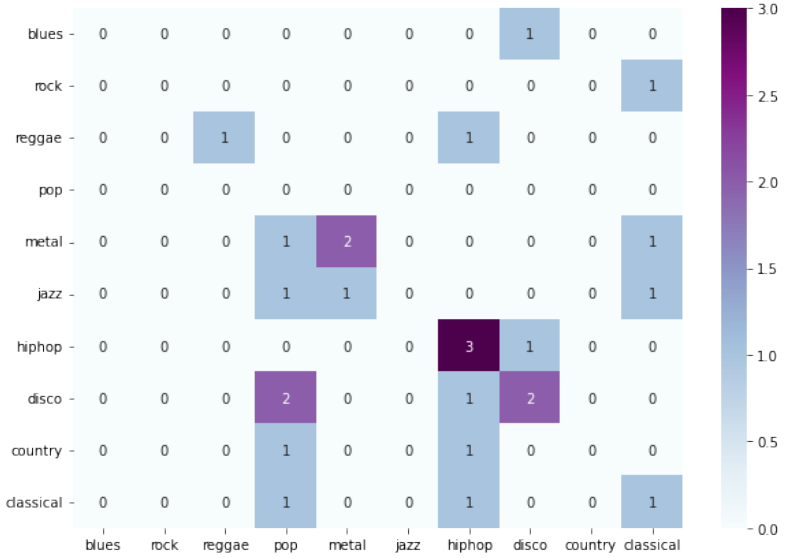


Figure 7: Confusion matrix for generated audio files from methods described in this work

spectrograms of that genre. However, in iterations past about 2 seconds of audio generation, the frequency features dissolve completely into static noise.

This is likely due to insufficient sequence length of the input and output audio as well as insufficient training. A training curve example for one of the music generation networks is shown in Figure 8. This was typical for the ten music generation networks: training loss decreased steadily while validation first increased then flatlined and eventually decreased (one network was trained beyond the standard 1,000 epochs to find this result). The initial increase in validation loss is explained by the fact that the audio sequence data has mean near zero; the initial state of the network produces a near zero output which is actually closer to the validation sequence than the later trained outputs. However, if the model truly learned information about the music genre, the validation error would again decrease.

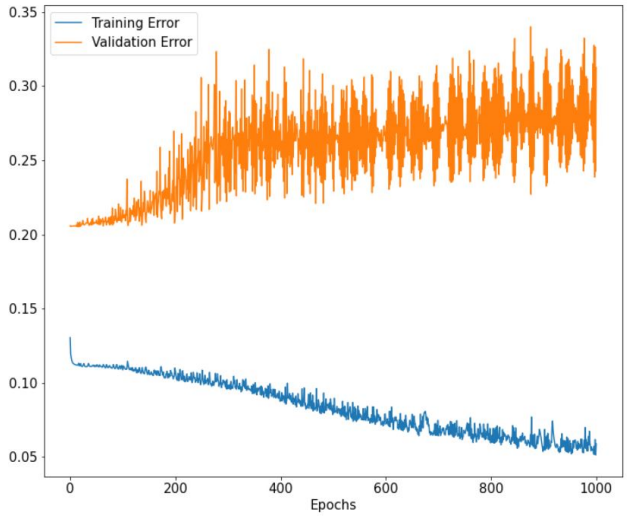


Figure 8: Training loss curves for blues music generation network

The primary issue here is insufficient computing resources, as the spectrograms and training curves indicate further training likely could have yielded improved results. Each training epoch resulted in clearer, less noisy audio and this trend would likely have continued. However, training the 10

networks was computationally expensive and within the scope of this project it was not possible to train the recurrent neural network sufficiently for the desired results.

This resulted in a confusion matrix such as that shown in Figure 7. Although some genres such as hiphop, metal, and disco show promising accuracy, there is still significant room for improvement due to the lack of musical structure in the generated audio.

5 Conclusion

In this work, we explored two neural network types for two related tasks in understanding music data. The recurrent neural networks for generating music proved difficult to train for the desired task, however with significant computing resources, it would be a worthwhile area of exploration since initial indicators are promising. The convolutional neural network designed for genre classification performed as expected given the nature of the generated music. Overall, this project presented an excellent opportunity to employ deep learning methods for an intriguing set of tasks and a worthwhile learning experience on the real challenges faced by machine learning researchers.

6 Additional Materials

Below you can find the files we utilized for data and the files containing our written code:

- Example of Music Generation Network (Reggae generation network)
- Music Classification Network
- Example of 30 second music training clip from GTZAN database reggae genre
- Example of 5 second generated song output developed from this music clip's seed input

References

[1] Victor Basu. Generate music with variational autoencoder. `https://www.kaggle.com/code/basu369victor/generate-music-with-variational-autoencoder`.

[2] Beatrix Benko. Music classification and generation with spectrograms. `https://deeplearning.neuromatch.io/projects/ComputerVision/spectrogram_analysis.html`.

[3] Andrada Olteanu. Gtzan dataset - music genre classification. `https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification`.