Using Deep Learning to Categorize Music through Spectrogram Analysis

Evan Otero, Ribhi El-Zaru, Nicholas Loeper, Andrew Heimerman

December, 9th 2017

1 Abstract

Humans have an uncanny ability to distinguish music genre almost instantaneously after hearing a piece. Even individuals with almost no musical training or expertise possess this ability. This proposes the question as to what defines a music genre. Music can be broken down into many pieces of information that lend themselves to analysis; tempo, rhythm, harmonic structure and melody to name a few, which can be consistently captured through spectrograms. By generating spectrograms from a large library of sample song data across many unique genres, we use supervised learning and a recurrent convolutional recurrent neural network (CRNN) to build a accurate classifier that asserts an unknown song's genre using cross referential spectrogram analysis. This classification will assess the probability of a song's genre within our respective target class (song genres). Classification accuracy will be used to measure the performance of our genre classifier. We improve upon previous similar experiments which used spectrograms to classify genre (Deepsound) by attempting to use a larger data set with more genre classifications.

2 Introduction

The overall goal of our project was to create a machine learning model that was capable of continuously predicting the probability distribution of songs being certain genres. Our project was inspired from a former genre classification project by Google's DeepSound team [**2**]. The DeepSound team established the basis of our project by providing referenced source code in their open source repository.

As a key differentiator between our project and theirs, the dataset and target classes allowed us to expand upon DeepSound's previous works. Our dataset was extracted from the Free Music Archive [**1**]. In comparison to DeepSound's dataset, our dataset was much larger, approximately 25,000 more songs. Additionally, we nearly doubled the amount of target classes, in this case genre classifications.

From our pool of many songs, we converted the audio files into mel-spectrograms (2-dimensional representations of spectral frequency). We used a squashing function (mel-scale) to convert the audio frequencies into something a human was more able to understand. The mel-scale is a better representation of how humans perceive certain frequencies. Spectrograms provide rich information about each song that not only is ideal for training machine learning models, but also significantly decreases the size of data. When discussing what features to use in training our model, spectrograms seemed like the appropriate fit since they accurately describe frequencies in the song and are represented as arrays of floating integers.

Our choice of model was a convolutional recurrent neural network. Since data representing songs is time sensitive, we needed our model to institute a "memory" mechanism to remember what has occured over the course of the song. We trained our model by randomly shuffling our data, splitting it into training and test sets, then training over 100 epochs. The architecture of our model allowed us to identify short term and long term structures in the song frequencies.

3 Methods

3.1 Dataset and Mel-spectrogram Generation

To obtain song sample data, we accessed the Free Music Archive (FMA) as described and provided by Defferrard, et al [**1**]. After comparing with DeepSound, who used the GTZAN dataset, we wanted to be able to provide the network with a more comprehensive dataset. The full FMA dataset offers 161 genre classifications as well as a database of 106,574 songs [**1**]. Because of the complexity of this full dataset, we opted to reduce to 25,000 songs over 16 genres, as provided in the FMA medium dataset. GTZAN consists of only 1000 songs over 10 genres. With FMA we wanted to be able to increase the number of samples by a multiple of 25 and the number of genres by 6. With such a significant increase in sample size, we believed that we could train our machine to have a higher classification accuracy than DeepSound's 67%, even with an increase in the classification difficulty through the addition of genres.

To represent the song data for learning, we converted the MP3 data to mel-spectrograms using the Python library, Librosa, while consulting the work done by DeepSound to ensure that we were processing the data similarly. Due to CRNN's ability to recognize images, representing music as spectrogram images fits our network model better than the MP3 data. Additionally, mel-spectrograms provide other unique benefits over regular spectrograms. Mel-spectrograms produced by Librosa are scaled by a log function. This maps the sound data to the normal logarithmic scale used to determine loudness in decibels (dB) as it relates to the human-perceived pitch. This provides a spectrogram that better maps the human hearing spectrum [**3**].

We decided to keep the same arguments for the creation of the mel-spectrograms as DeepSound. Each mel-spectrogram has 2048 samples per song for the fourier transform on the song, and the number of mels is set to 128 (default). This corresponds to the average human ability to hear bursts of sound (10ms or longer, which is derived from the song length/sample-window) [**2**].

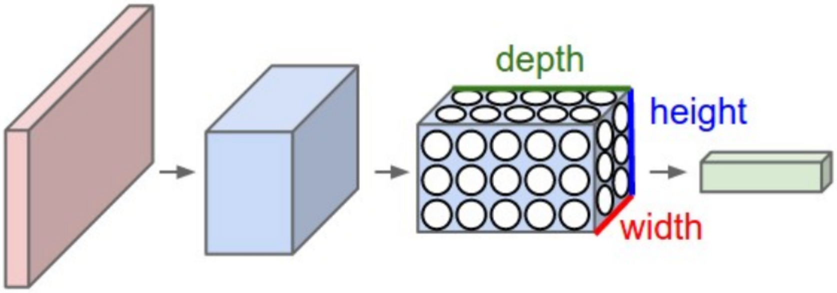
The song data set was processed into spectrograms using the metadata provided with the FMA dataset, which contained a mapping of all MP3 samples to an ID, title, artist, genre, and several other features related to the original tracks (FMA). For our purposes we trimmed this mapping down to be a mapping of '$track_id'$ to '$genre_top'$. We then removed all rows of songs that were not in our 16 genre dataset. We then iterated by genre for each of the 16 genres and created a 3-dimensional array of the mel-spectrogram data array, the one-hot encoded genre array, and the mp3 path. For each genre this data was outputted as a pickled file. These files were then concatenated and any empty rows removed (the metadata contained all song IDs, while the dataset contained a subset of the full dataset, so there were rows with zeroed arrays for each index). We also removed the genres International, Easy Listening, and Experimental. These were removed because of their ambiguous genre definitions and their small song count. The rows of the final converted song set were then randomly shuffled to ensure that the learning was receiving a random sample for each row pulled from the dataset.

The final converted song dataset was reduced to 21,695 songs and resulted in a total size reduction from approximately 23GB to 7.156GB. This size reduction was critical to allowing our algorithms to access songs at a much faster rate. We reserved 70% of this set for training and the remaining 30% as a test set[**3**].

3.2 Model

3.2.1 Motivation

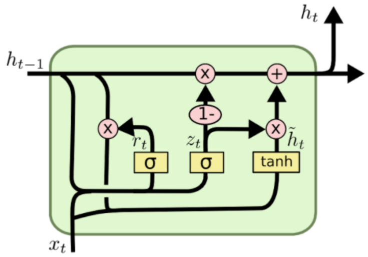
Convolutional Neural Networks (CNNs) are used in many tasks, such as segmentation 3D volumetric images, scene parsing, action recognition from videos, or objection detection. They are designed for processing data that has a known grid-like topology and have been tremendously successful in practical applications. Essentially, Convolutional Neural Networks are simply neural networks that use convolution, a specialized kind of linear operation, in place of general matrix multiplication in at least one of their layers. Thus, given that CNNs would be able to take advantage of the 2D structure of an input image and that this study is using visual representations of frequency distribution over time (mel-spectrograms) to solve the genre recognition problem, a CNN would be the ideal candidate for feature extraction.



A

Convolutional Neural Network (CNN)

Recurrent Neural Networks (RNNs) are a family of neural networks for processing sequential data. Much as a convolutional network is a neural network that is specialized for processing a grid of values X such as an image, a recurrent neural network is a neural network that is specialized for processing a sequence of values x(1), ..., x(n). One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task. Sometimes, we only need to look at recent information to perform the present task; however, there are also cases where we need more context and have long-term dependencies [**5**]. Unfortunately, as the gap grows between contexts, RNNs become unable to learn to connect the information [**4**]. Long short-term memory (LSTM) cells are explicitly designed to avoid the long-term dependency problem. The cell is responsible for "remembering" values over arbitrary time intervals; hence the word "memory" in LSTM. Using Long Short-Term Memory (LSTM) architecture, the network would then have the ability learn over many time steps, making them well-suited to classify and process time series given time lags of unknown size and duration between important events. Relative insensitivity to gap length gives an advantage to LSTM over alternative RNNs, hidden Markov models, and other sequence learning methods in numerous applications.



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

LSTM Cell

3.2.2 Architecture

In this study, we used CNNs and LSTM cells to categorize music as time progresses through mel-spectrogram analysis. First, we extract features from the spectrograms using convolutional layers, where 1-dimension convolutions across the time axis were performed in order to measure changes across time. The features are translation-invariant, where the same (pooled) feature will be active even when the image undergoes small translations, only in time domain, since higher and lower frequencies need to be distinguishable. A rectified linear united (ReLU), with the purpose of helping the network train faster and alleviating the vanishing gradient problem, and 1-D max pooling, which helps control overfitting and reduces spatial dimension of the input volume, were added after each convolutional layer.

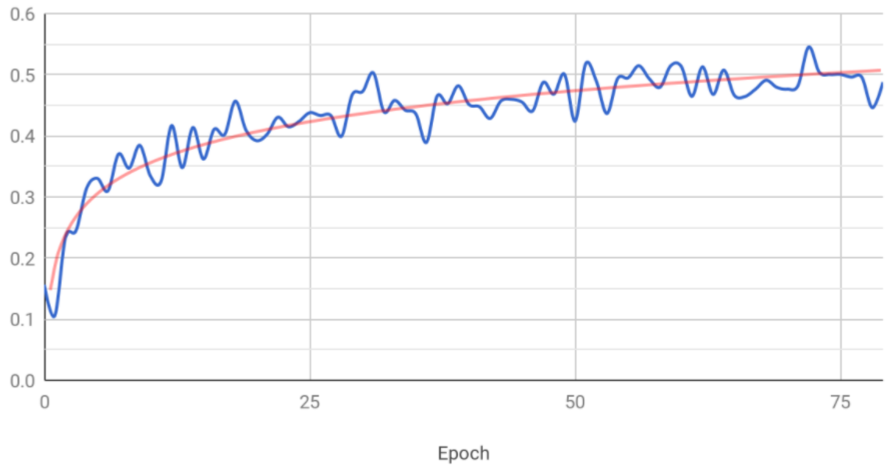
The resulting sequence of features is then fed to an LSTM layer, which should find both dependencies across a short period of time, and a long term structure of the song. To regularize, dropout layers were used. The idea is to "drop out" a random set of activations in that layer by setting them to zero. This forces the network to be redundant, where that the network should be able to provide the right classification or output for a specific example even if some of the activations were dropped out.

After the LSTM and the following dropout, the resulting sequence goes through a time-distributed fully connected layer with softmax activation, essentially producing a sequence of 16-dimensional vectors for each timestep. This vector contains the probability distribution at that point in time for each of the 16 genres.

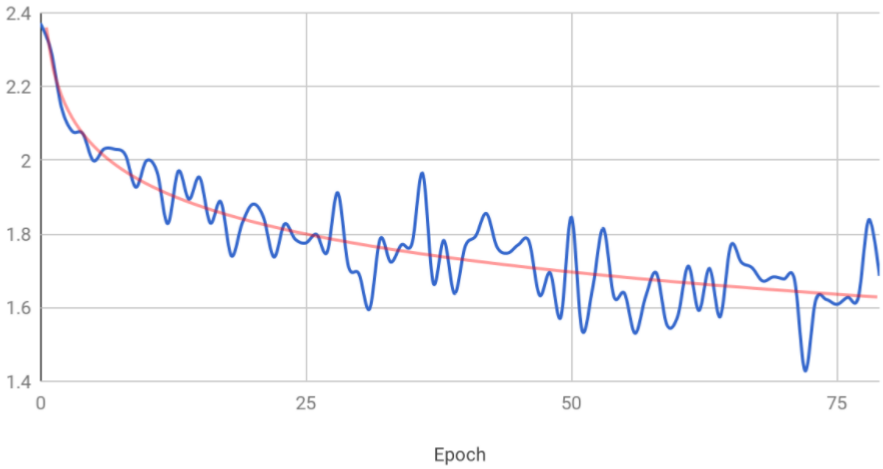
3.2.3 Evaluation

A loss function suitable for multi-label classification is categorical cross-entropy. It takes two probability distributions: the real one and the predicted one; however, the predicted sequence of the probability distribution must first be aggregated into just one probability distribution. A simple solution here is to the arithmetic mean across time of all the predicted distributions and return it as a final answer. The arithmetic mean of vectors, each representing a probability distribution, returns a vector that is also a valid distribution. The rationale here is that, for example, it is rather expected for a pop song to play in a pop genre for a majority of the time. Thus, if most of the song is classified as pop yet it sounds like classical, the classification is most likely incorrect.

The important metrics being evaluated here are the validation accuracy and validation loss. Validation accuracy is the number of correct prediction made as a ratio of all predictions made on the test dataset (30% of the original dataset). The test dataset is data that the neural network has not been trained on. The validation loss is a performance metric for evaluating the predictions of probabilities of membership to a given class. Predictions that are correct or incorrect are rewarded or punished proportionally to the confidence of the prediction.



Validation Accuracy



Validation Loss

As a baseline for this study, random guessing with 13 genres would be an accuracy of 7.69%. State of the art music genre recognition on GZTAN (a different dataset than ours) using deep learning was 84% accuracy [**6**]. However, our model solves a slightly different problem - we don't just want a single prediction for each track, but a continuous output containing the network's belief of the genre in every point of time. Thus, some accuracy was sacrificed in order to achieve that. The results of this model are discussed in the following section.

4 Results

After training, the convolutional neural network had 54.52% classification accuracy. Though we were disappointed with being unable to beat Deepsound's classification accuracy of 70%, this result is not entirely surprising due to our introduction of genres and our unbalanced dataset. We tied the predictive model generated by the CRNN to front end in order to visualize our model's predictive ability. A demo can be used by following the instructions on our repository.

Being that the 54.52% accuracy rate was on predicting the genres of the training set data, we wanted to see how our model would do compared to our own categorization. During this experimentation, it was interesting to see exactly how our model differed from our traditional definitions of genre, and we gained much insight into how music progresses over time with respect to what its genre seems to be.

Firstly, we noticed that the model often disagreed with what we considered to be "Hip-Hop" and "Rock" in the modern day. After testing our model on modern-day rock songs, such as "Last Nite" by the Strokes, and modern-day hip-hop we found that the model was significantly more prone to error than with traditional hip-hop and rock, such as Nas and ACDC. It would often identify these genres as either Pop, Soul/RnB, Instrumental or Electronic. This goes to show just how important data is in our model's training. The model's definition of Hip-Hop, Rock and so on, is ultimately determined by what we give it as said genres. Thus, when some genres have very few entries, the model is unable to build a deep enough understanding of said genre. Thus, the few amount of songs for country, Soul/RnB and blues meant that the model didn't have an encompassing understanding of said genres and overvalued the few attributes that it attributed to each of these genres.

Furthermore, we felt that the changing probability distribution at time that our model generated provided a very unique perspective into music. Many genres seemed to be very correlated in our model. For instance, Instrumental and Electronic, Classical and Jazz, and Soul/RnB and Pop seemed closely correlated with each other respectively. Thus, an interesting task for future work would be to calculate and demonstrate the correlations of said genres over time. By understanding what genres are most likely a specific one, we would gain a strong understanding of the relationships and similarities between genres.

Lastly, we noticed that by training our predictive model on spectrograms generated from mp3 files, it was unable to detect attributes of songs such as lyrics that would eliminate the chance of a song being a genre like Classical or Jazz music. For instance, when we tested our model on Blues songs, the model would predict the song to be Jazz music at very high probabilities even when there were vocals. This is understandable, as the two genres do sound very similar, but low probability of lyrics in Classical Music ideally should have made the model weigh classical music very low in songs with lyrics. Since spectrograms are unable to train the model on how to explicitly detect lyrics, the model was unable to do this task that comes logically to humans, showing a minor issue in this approach to the problem.

5 Conclusions

Mel-spectrograms proved to be a reliable method to compress song data in a format that is also easy to train upon using CRNN. Because spectrograms are two dimensional visualizations of songs, common features and differences are visible between songs and the model was able to achieve a significant classification accuracy. However, we believe that we could have achieved a higher rate of success and fewer misclassifications with a more evenly distributed song dataset. Future modifications would be to expand the dataset to the full FMA dataset, which has a near even distribution across all genres. Also, calculating the expected value of the genre over the whole song instead of at timeslices could offer a higher overall classification accuracy. Another improvement of our model would be to analyze the confusion matrix that was produced during our learning. By understanding how the genres are perceived by the model to be related, modifications could be made to reduce the overall confusion of genres.

References

[1] Defferrard, Michaël, et al. “FMA: A Dataset For Music Analysis.” [1612.01840] FMA: A Dataset For Music Analysis, 5 Sept. 2017, arxiv.org/abs/1612.01840.

[2] Michalak, Piotr Kozakowski Bartosz. “Music Genre Recognition.” DeepSound, 26 Oct. 2016, deepsound.io/music_genre_recognition.html.

[3] Stevens, S. S., et al. “A Scale for the Measurement of the Psychological Magnitude Pitch.”The Journal of the Acoustical Society of America, vol. 8, no. 3, 1937, pp. 185–190., asa.scitation.org/doi/10.1121/1.1915893.

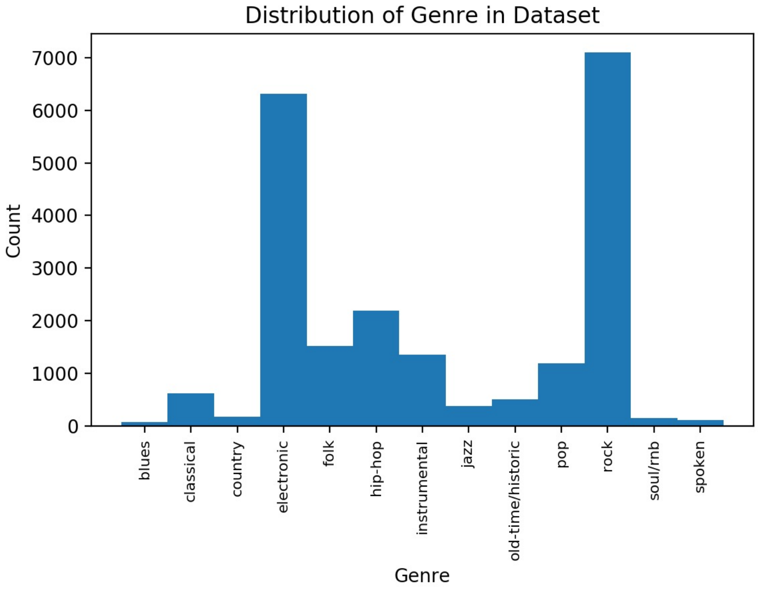
[4] Olah, Christopher. “Understanding LSTM Networks.” Colah’s Blog, 27 Aug. 2015, colah.github.io/posts/2015-08-Understanding-LSTMs/.

[5] Goodfellow, Ian, et al. “Deep Learning.” Deep Learning, MIT Press, 2016, www.deeplearningbook.org/.

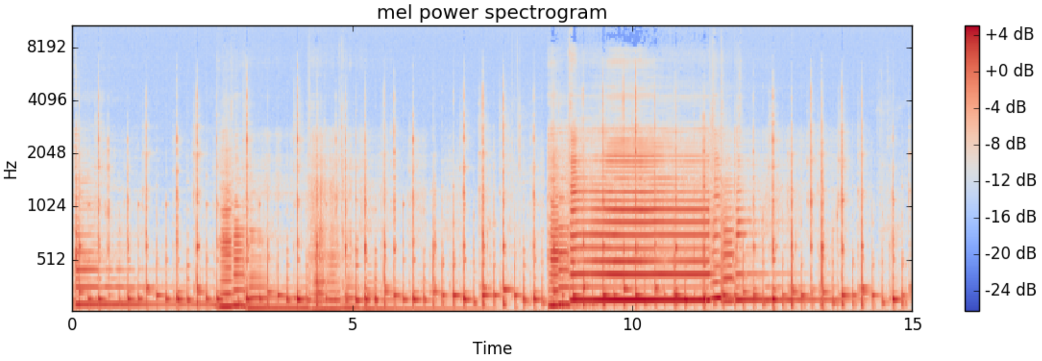
[6] Hamel, Philippe. ”Deep Learning in MIR: Demystifying the Dark. Part II: The State-of-the-Art”. 4 November 2013, https://marl.smusic.nyu.edu/wordpress/wp-content/papercite-data/pdf/ISMIR2013$_{D}eep_{L}earning_{P}art2_{H}amel.pdf$.

Appendix

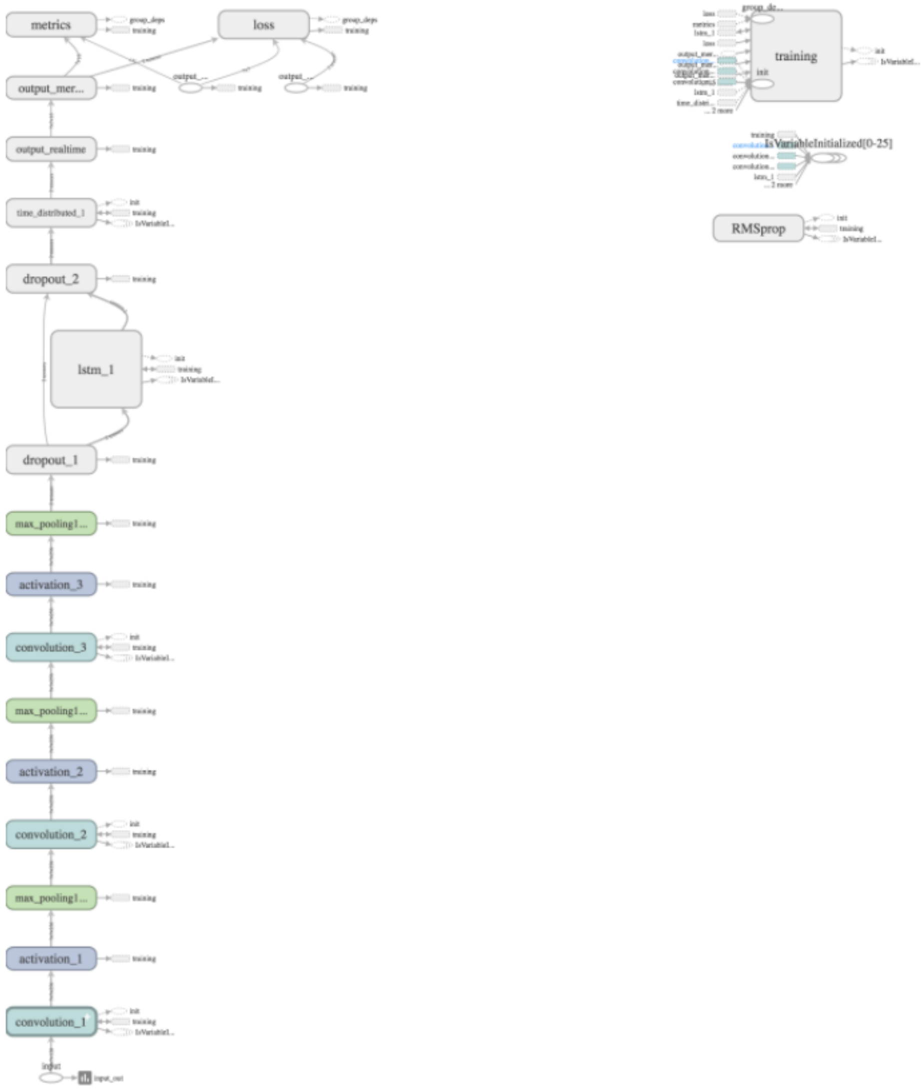
5.1 Distribution of Genres in Dataset



5.2 Sample Mel-Spectogram



5.3 Convolutional Recurrent Neural Network (CNN + LSTM)


metrics
loss
group_deps
training
output_mer...
output_...
output_realtime
time_distributed_1
init
IsVariableI...
dropout_2
lstm_1
dropout_1
max_pooling1...
activation_3
convolution_3
max_pooling1...
activation_2
convolution_2
max_pooling1...
activation_1
convolution_1
input
input_out
group_de...
training
init
IsVariableInitialized[0-25]
RMSprop