Noah Masur, Ali Woodward, and Aidan Wall

CPSC 393- Machine Learning

May 19, 2021

Predicting the Next Song in a Playlist

CPSC 393 Machine Learning Final Project

**Abstract**

Music curation and discovery has been revolutionized by the technology industry in the last 10 years by large tech companies like Spotify. In this study, we aimed to use a deep learning Long-Short Term Memory (LSTM) architecture in order to generate our own model of music recommendation. After training our model on 103 playlists to recognize the sequential patterns of playlists, the goal was to be able to generate a song recommendation after giving the model 5 songs to come before it. Our model ended up being able to generate the specific attributes of a new song and using a search function it was possible to hear examples of songs with said attributes. Ultimately in further research we would hope to fine tune our model and work towards generation of actual playlists instead of singular songs.

**Introduction**

The world of music that we live in today is a different experience than any time before ours. Music is more accessible than ever with the help of machine learning algorithms that recommend songs, curate playlists, give you statistics about your listening and everything in between.

Some of the leading companies in music streaming services are Spotify and Apple Music, both having millions of monthly listeners all around the globe. While these dearly loved apps have made discovering music a more pleasant experience than manually searching for songs, understanding the technicalities behind music recommendation and playlist curation is a fun way to better understand how exactly we are getting our music.

Spotify has made huge sums of their data accessible to the public as well as their API that allows users and creators to use Spotify’s built in functions to do their own music testing and generating. For our final project, we wanted a better understanding of how we might use deep learning that we have learned about throughout the course of this semester to generate our own music instead of relying on the apps we use daily. More specifically, we decided to create a model that understands the sequential patterns of playlists and can produce the metrics of a newly generated song based on 5 songs that it is given before it.

**Data**

The data that we used was accessed through Spotify’s API. This dataset consisted of 103 playlists. The original dataset was significantly larger, however, we had to take into account that many of Spotify’s curated playlists are made based upon one particular artist. Using artist specific playlists would be detrimental to our study because it is not a representation of a playlist that is based upon a specific theme or genre other than the artist themself. Instead of our model learning about the progression of a genre-based playlist, an artist specific playlist may only generate songs that are extremely similar to one artist’s style. After removing all artist specific playlists, we were then left with the 103 playlists that we used for training. Each row of the dataset included a playlist ID, track ID, and seven key song attributes. The “mood’ attributes included in the dataset consist of danceability, valence, and energy. The “properties” attributes were loudness, speechiness, and instrumentalness. Lastly, the “context” attributes included liveness and acousticness. In total, the dataset included 8,025 songs, with each playlist ranging from 49 – 99 songs in length, resulting in a dataset of 9 columns and 8,025 rows.

**Methods**

Since our data is sequential, the model we decided to use is a Recurrent Neural Network (RNN). To combat issues that would affect accuracy, including the vanishing gradient problem, we used a specialized RNN, a Long Short Term Memory network (LSTM). To input the data into the LSTM there was some pre-processing required. First, we converted the 2-dimensional data frame of the tracks by their attributes into 3-dimensional data set by separating the tracks by playlist. We then removed the playlist name as a feature because it was not a feature we were predicting on. For our model, we decided to predict based on five songs through testing and because this is the maximum number of seed tracks allowed by the Spotify API function to get recommendations. To implement this, we went through each playlist and sequentially stored sections of five songs as explanatory variables (X) and the next song as the response variable (Y). Although this could pose a training issue for shorter playlists, all the playlists we used were almost 50 songs long, so the model is able to understand the patterns of playlists based on five songs. The data we inputted into our model was X-data that had 7,313 separate chunks of 5 songs with 7 attributes, and Y-data that had 7,313 separate chunks of 1 song with 7 attributes.

To begin training our model, we started off with a network of 2 LSTM layers of size 64 and 32, respectfully, with a dropout layer of 0.2 separating them. The initial model was trained using a batch size of 16, Adam as the optimizer, and Mean Squared Error (MSE) as the loss function over 50 epochs. Over the course of many iterations in which we changed the amount of layers, size of layers, optimizer, loss function, and number of epochs, the effectiveness of the model did not change. Therefore, we decided to focus on quickness of training and simplicity, so our final model is similar to our initial model with an additional dropout layer and less epochs (because it stabilized much sooner than 50 epochs).

**Results**

The results of our project is a model that given the last 5 songs, will predict the features of the next song. On the training data, the model does very well, with the training loss decreasing over time. The validation loss did not do as well.

Graphical user interface

Description automatically generated with low confidence

This could possibly mean that our model is overfit. One reason this could be is that it is only training on 8,000 songs across ~100 playlists. If we had the ability to train it on 50,000+ songs over many more diverse playlists, it might be less over fit. This is also in part because the data it is being trained on is not too similar. There are thousands of genres of music, and it is only being trained on 100 playlists with genres of music that do not show up more than once and genres that do not appear at all. Our model would probably perform better if we implemented more measures to combat overfitting, but with the nature of the data this is kind of challenging, and because it is so subjective, two songs may not be too similar in characteristics but in sound and perception might be much more similar. Even if the similarity between the features is not super close according to the metrics of the model, they may very well be similar songs. The only way to truly test this is to listen to the songs and use that as a metric. We also ran our data through a clustering algorithm, to see if there is separation between tracks, by genre. There was distinctness, but not to the human eye in 2 dimensions. If we had the ability to graph and draw clusters in 6 dimensions, it would show separation between all the different types of music.

We also learned a lot about pre-processing data, and a lot of general knowledge about API’s. There are a lot of features within Spotify’s API that helped us complete our project. Our results leave us with desires for improvement, such as allowing individual users to have tracks generated for them based on their playlists on their Spotify account.

**Discussion**

Our process and results taught us a lot that we did not initially have knowledge about at the beginning of this project. While it is so useful that Spotify has made their API public and we began the project very excited to use its features, there was definitely a big learning curve that came with learning to implement the API in the ways we had initially conceptualized. Through thorough data exploration and understanding of the data that we were training our model with, it is understandable why our model may be overfit. The world of music is so vast that trying to accurately train a model without certain time or resource elements is extremely difficult to truly encapsulate the entire music space. Our results showed us how much there is to be learned in the area of music recommendation as well as how deep learning architectures can be implemented in a way that we had not had a chance to experiment with yet. Ultimately, our results were not as conclusive as we had hoped but the knowledge that we gained through the exploration of music data and LSTM implementation with partnered use of Spotify’s API was an extremely beneficial learning experience.

**Conclusion**

In conclusion, our project ended up looking different than how we initially anticipated but truly taught us so much in the process. The LSTM model that we created was able to generate attributes of a song following the previous five and by searching these attributes and analyzing by ear, we were extremely pleased with how closely related some of the songs were. There is much work to continue to be done in this area for both tuning of the current model as well as large scale improvements for future usage.

**Future Works**

There are many exciting ways that we could move forward with this project in the future. For this project, we trained our model on 103 playlists from Spotify’s dataset, however, with more time and processing power we would love to train the model on Spotify’s “million playlists” dataset. Further extension of the study could be focused on the hyper-tuning of parameters to perfect what we would like to succeed. Lastly, with the basis that we have, future work can be done to be able to generate full playlists using our sequential model instead of just one song. This could allow for a new basis for playlist generation that not only focuses on the overall genre and mood of songs to put them together in a playlist, but truly understands the nature and progression in which a playlist is put together in a specific order.