Abstract

This project sought to apply and explore Machine Learning and AI techniques to remix songs. The project is inspired from https://deepart.io/ which recombines content and style of two arbitrary input images together using convolutional neural networks to produce artistic images that resemble the original inputs.

To accomplish the project’s goal, two different approaches have been attempted. The first approach uses a deep neural network that learns features of wave-based sound files to reconstruct a new sound file that has the properties of two different songs. The second approach uses a graph-based probability model that is built from sample music files. The second approach generates a new song by sequencing notes based on the model’s probability function. Unfortunately, the neural network has not been implemented successfully because of poor training performance; however, the graph-based model has been successful at generating songs.

Architecture

Both approaches have been implemented using the Python programming language. For the neural net approach, experimentation was done using a convolutional neural net with 2-3 convolution layers followed by 1-2 fully connected layers. Various number of features were tried for convolution layers, but they were typically between 32 and 128, whereas the number of nodes in the fully-connected layers were either 256 and 512. Various learning rates were also experimented with, ranging from .1 to 1e-5 as well as dropout rates, which ranged from .25 to .75. Additionally, different filter sizes were tried, as well as max vs. average pooling.

For the graph-based probability model, a framework was implemented that provides functionality for building NGrams, and generating new songs using the NGrams. It contains interfacing modules and configuration options for implementing custom song generators and NGram models. Flask was used for the web application, and MIDI.js for playing midi files on the client side. The website is hosted on an AWS server.

Impact

Casual Users
- Recreate music from favorite songs
- The generated song will contain the notes and certain “themes” or “motifs” from the original inputs
- The duration of the the generated song can be significantly extended and varied, although the song becomes monotonous and repetitive as its duration is increased
- Modifying the n-size of the NGram changes the variation of the output song.

Developers
- Plug and Play Modularity and Interfacing
- Implement custom generation algorithms using NGrams
- Implement custom modeling algorithms

Results

For the deep neural net approach, a neural net was successfully built for predicting the genre of the input music. Unfortunately, due to limitations of time and resources, the neural net could not be trained to be sophisticated enough to produce accurate predictions. The first filter layer of the convolutional neural net was supposed to encapsulate most of the features from training music. The layer was planned to be used for generating music; however, since training the neural net had not yielded results, further work with this approach was stopped.

A modular framework has been built for the graph-based model approach, which uses midi files to generate music. The framework provides multiple options and configurations, some of which are available on the website. For example, the user can choose what type of generation method he wants to be used to produce the final song. Or, the user can choose from which midi music sources he wants the generator to sample music. The generator can either sample one song or a corpus of songs. Additionally, multiple options are available for mixing music, which include: mixing instruments or adding instruments from different sources.

Summary

Although different implementation methods were investigated, a deep neural network could not be developed to learn style from music in the same way that https://deepart.io/ does for music, because of an inadequate amount of time to do enough modeling and experimentation with the neural network.

The graph-based probability model method worked successfully, but required significant implementation and design efforts due to inherent complexity in processing midi songs, and the generated song aesthetics can be quite poor for certain inputs. The generation algorithm can be further improved by adding mp3 support, which provides higher music playback quality, and applying various techniques and rules such as Fuch’s counterpoint to refine the generated song’s aesthetics.

The team would like to thank our sponsors, Josh Cai and Sihui Huang, and faculty advisor, Dr. Vincent Ng, for lending us their time and guidance.