ART AND MACHINE LEARNING
CMU 2019 SPRING
Final Project

Baroque Revival

Edward Goebel, Ananya Chandra

DESCRIPTION

Concept:

Music is an integral part of society and culture. Traditionally, music is associated with its creator and the time period in which it was created. One can often hear a melody and a slew of associations come to mind. Songs with lyrics typically evoke an idea from the artist. We attempt to blur those associations, using ML techniques to create an original song featuring novel lyrics in the style of pop music with a melody reminiscent of classical music from the Baroque era. Since style transfer is common in the intersection of art and machine learning, we chose to create a literal style transfer of musical eras. Our project is an exploration in the meaning behind a song's context. We also continue our semester-long exploration of the question of whether creativity can be automated.

Additionally, we wanted to understand what makes a song "modern." We want to gage our audience reaction to see if they assess the result as a modern work or as a baroque work. We desire to see if the content of the lyrics is what gives music the modern feel or whether the content of the background music associates the song with the baroque world. In fact, as we further explored these questions, we found that there was no melodic recipe for modern music. This inspired us to create a musical piece that would force us to ask ourselves these precise questions.

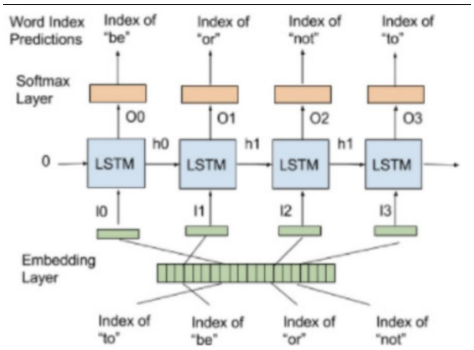
In our preparation for this project, we came across the research of Feynman Liang et al. of Cambridge University in collaboration with researchers at Microsoft. The team published the paper *Automatic Stylistic Composition of Bach Chorales with Deep LSTM*[1]*,* and created *BachBot*[2] as a proof of concept. *BachBot* is an online application which has users attempt to distinguish between real Bach compositions and music generated by their LSTM trained on Bach's music. In practice, they have found users' guessing abilities to essentially be no better than random guessing. Guided by this research, the Magenta project created Polyphony RNN[3], an LSTM which, like the LSTM behind *BachBot*, is able to model multiple simultaneous notes. We chose to use this model to generate the classical music melody for our project. For the lyrical portion of our work, we wanted to understand pop culture music so we wanted to create a great dataset full of the trendiest pop music to train on, and we wanted to generate results that could be tied into our generated Bach music.

Technique:

The inputs to the Polyphony RNN are Midi files that separate the simultaneous notes into different layers. Polyphony RNN operates by first converting the MIDI files into NoteSequences, a data format which is more efficient for the network to work with than the MIDI files. From these NoteSequences, Polyphony RNN generates SequenceExamples, which contain a sequence of inputs and a label which represents a polyphonic sequence. The SequenceExamples are what are ultimately fed into the neural network during training and evaluation.

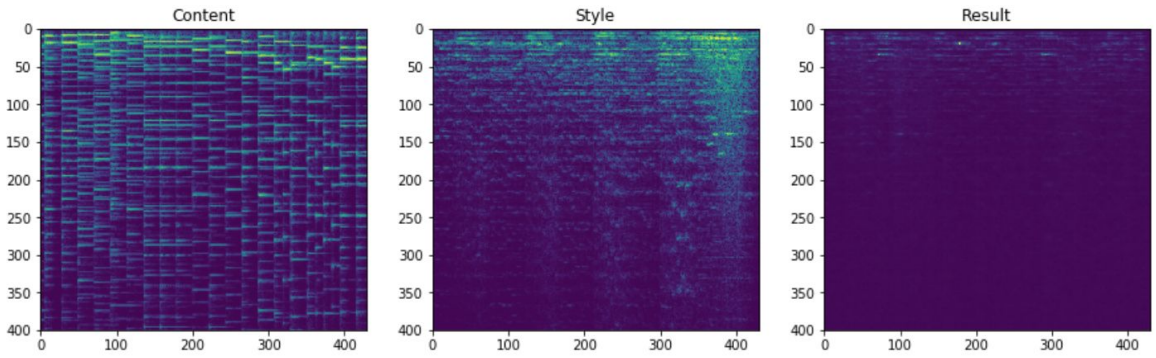
To gather the datasets we used to train Polyphony RNN, we searched the web for large amounts of MIDI files from Baroque artists such as Johann Sebastian Bach, Arcangelo Corelli, and Georg Philipp Telemann. We found a large MIDI file source list curated by Albert Meronyo[4] which was useful for compiling our datasets. After many trials and errors we produced a midi file that we believes sounded Baroque but would also mesh with some of our generated lyrics.

For generating lyrics, we manually compiled a dataset with lyrics from the top 40 pop music in America over the course of the last eight years. We felt that this would most accurately represent modern music. We used a multi-payer Recurrent Neural Network (LSTM) for training and sampling from character-level language models. The model inputted a large text file that represented our dataset and learned to predict the next character in the sequence.



Source:https://www.kdnuggets.com/2017/08/deeper-recurrent-networks-sequence-bag-words-model.html

We also attempted to style transfer a Baroque style orchestra onto our chosen Midi file result. This occurs through analysis of the spectogram:



Essentially, the style of a modern-day orchestra is transferred onto the content by using the spectogram images as content images and style images. Additionally, we changed Alpha several times in order to get a final result. Alpha controlled the intensity of the style versus the content. We were not too happy with the results after playing around extensively with the various parameters and layers.

Instead, we moved forward by porting our music onto Musescore, an application where we were able to create direct sheet music and play around with the instrumentation of the music itself. After conducting research into Baroque style orchestras we used cellos, violins, and french horns to create the most authentic Baroque sound. After this, we ported our results into music editing software where one of our team members, Ananya, sang the lyrics over the music by creating the most aptly fitting melody over the generated background music. We also received help from a sound editor who created an appropriate pop beat as well as worked on the sound mixing of the result.

Process

Our first trial was meant to be a small proof of concept, so we trained on only 23 MIDI files for a very short training period. The results generated were interesting, but sounded a bit much like random notes being played. The first trial results can be found here :
https://github.com/ananyachandra/BaroqueRevival/tree/master/Trial1

Next, we tried training the model for several more hours to see how the results were affected.
The second trial results can be found here: https://github.com/ananyachandra/Baroque Revival/tree/master/Trial2
We found these results to be a step in the right direction, but we decided that a larger dataset would be needed to achieve better results.

Our next model included a larger dataset than our previous trials along with a change in the number of epochs and the number of hidden layers.
The third trial results can be found here:
https://github.com/ananyachandra/BaroqueRevival/tree/master/Trial3

After training for about 16 hours, we came to :
https://github.com/ananyachandra/BaroqueRevival/tree/master/Trial4

This fourth trial was among the best results we were able to achieve for Project 3. In order to achieve even better results, we set up our AWS environment to run with the *magenta-gpu* environment which uses the *tensorflow-gpu* package instead of the normal *tensorflow* package. This allowed us to make better use of the GPU on AWS and train Polyphony RNN on about 700 MIDI files, a much larger dataset than we used previously. Indeed, this led to an improved result, which we felt embodied the style of Baroque-era music better than our previous trials. The results sounded slower and more methodical, as opposed to the more chaotic melodies in our previous trials. After playing with several parameters, and editing the number of layers, we generated audio we found to be the best result and used to proceed with our project can be found here:
http://www.andrew.cmu.edu/user/egoebel/generated_baroque.mp3

In terms of lyric generation, we created a dataset by running a script that took the lyrics of top 40 music for the last eight years. After playing around with several parameters and running with a varied amount of epochs using Shakespeare-LSTM but replacing the dataset. We played with a variety of loss functions for the model including mean squared error, mean absolute error, categorical cross entropy, sparse categorical cross entropy, and binary cross entropy. Some of the lyrics we generated include:

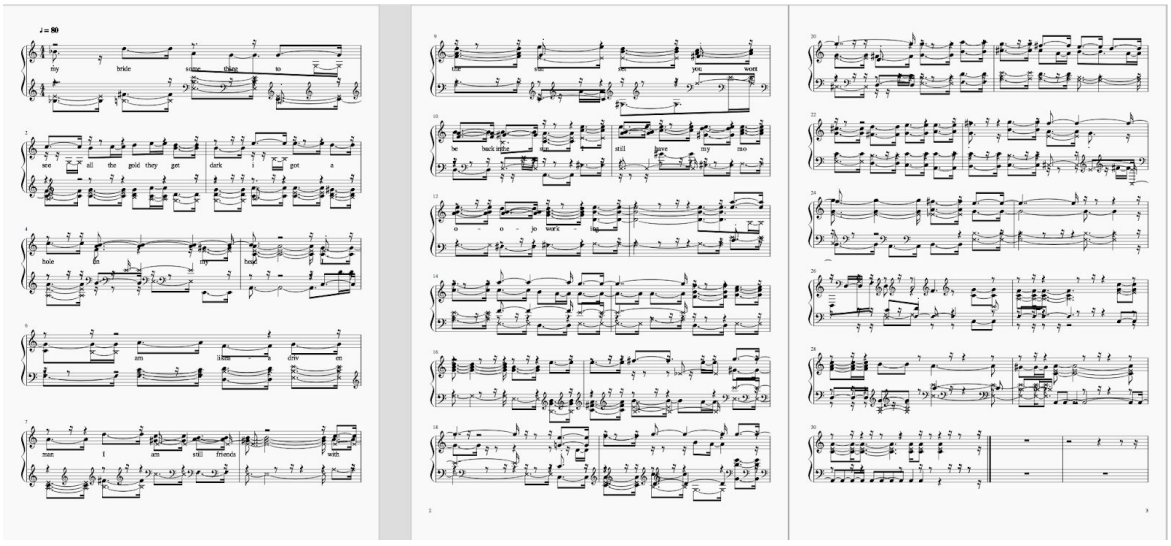
i won't let go
i don't know what i wanna hold you hard
i wanna be your end
but that's all right my darling
i have to be loved
i don't know what i wanna hold you down
and i hope that you love me
too much, come on now
oh, darling
BINARY CROSS ENTROPY

if it's loving that you need, i got it right here, baby, now
i got a feeling in my arms
i won't be a bridge one to black
i've been reading look before it all
and i was a friend to the same
what i said that you're gone
SPARSE CATEGORICAL CROSS ENTROPY

Our final lyrics are the following:

My bride something to see, All the gold, they get dark
I got a hole in my head, I am like a driven man,
I'm still friends with the sunset, that you won't be back to the sun.
I still got my mojo working but I said I'm gonna go home.
Yeah, I've got to say I love you and I can't help but drive about you.
you know I'm hard, I've got to go on, I can see the way you make me feel.

When we were satisfied with the classical-style music and pop-song lyrics generated by the RNN models, we focused on combining the two components into a single, coherent song. At this stage in the process, we leveraged our own creativity, as it required determining how the generated lyrics should be sang with the generated melody, recording Ananya singing the lyrics over the melody, and utilizing post-processing effects to further improve the result.



Result:

The result can be downloaded at http://www.andrew.cmu.edu/user/egoebel/BaroqueRevivalFinal.m4a. As mentioned, the result combines our best result from generating classical music and lyric generation. Ananya recorded herself singing the lyrics alongside the classical tune, which had been edited to sound like typical Baroque-era instruments. A pop beat was added to make the song sound more natural and flow more. In this way, our result combines results from machine learning algorithms with our own creativity.

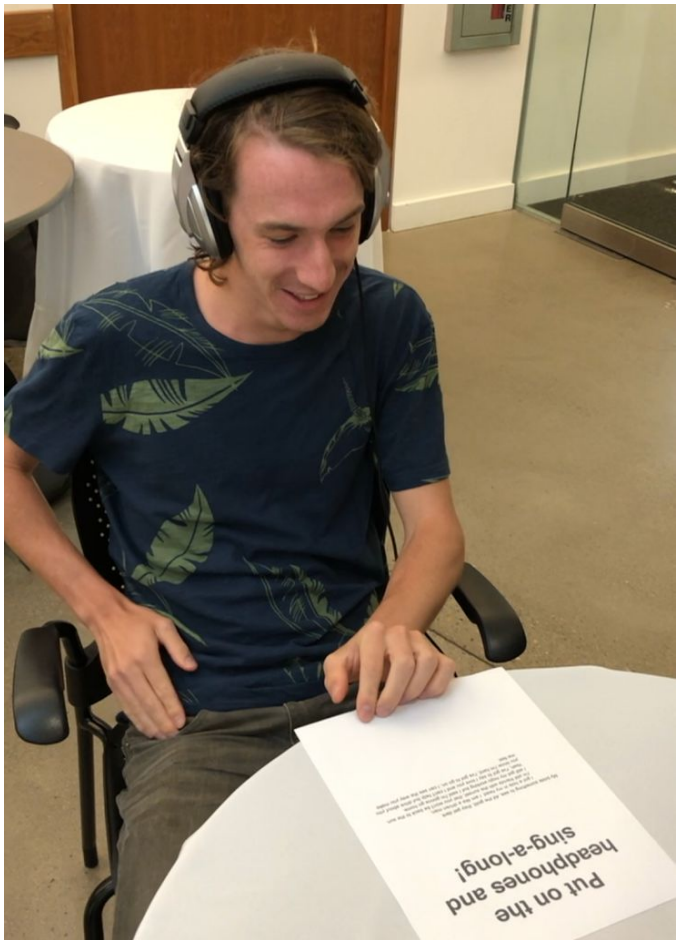
Reflection:

The classical music melody and pop song lyrics we used as a basis for our project were chosen through an iterative process of gathering and changing datasets, training the RNNs, and carefully examining the results. Then, we worked with these results, combining them into a single song. We found this project to be a fascinating exploration into combining multiple machine learning algorithms as well as our own creativity in order to create a novel artistic work.

In terms of our initial exploration we found that many members of our audience at the exhibition believed the song to be a modern song. Their subconscious mind immediately connected the lyrics to a popular song and thought the song had a baroque feel. This led us to understand that modern day pop music may not necessarily have a melodic formula. We can see this with the success of a song like “Old Town Road” by Lil Nas X, where the lyrics are very modern and forward-thinking but are set to a very old-sounding country track. In fact, we can maybe understand from our project that what makes “modern” day music “modern” is the fact that every piece is unique and that there is no true definition of what will hit the top charts and what will not. We could see our listeners enjoying our final result and nodding their head to the rhythm. Some even commented that they would like to listen to this song on repeat.

One lesson we learned is that training and evaluating results from neural networks is an iterative process, and to achieve good results, it is vital to gather a large enough dataset and allow the network to train for a sufficient amount of time. Through the sub-par results of some of our early trials and the improvements of our later trials, we came to appreciate the human component in evaluating the output of machine learning algorithms for novelty and artistic value.

We believe that our project is also a solid exploration into the questions which inspired it regarding algorithmically-generated music. We find artistic value in the result we have produced. At the same time, we think of it in a different light than other songs with the knowledge that its core components were created with the use of machine learning techniques. Our result is a unique blend of techniques and styles, befitting of its union of such different genres being brought together into one work.



We enjoyed hearing feedback from individuals who attended the exhibition. They remarked upon the novelty of the project and enjoyed reading the lyrics. Some were intrigued and were curious about how the melody and lyrics were generated.

References:

[1] Feynman Liang, et al. Automatic Stylistic Composition of Bach Chorales with Deep LSTM.
https://ismir2017.smcnus.org/wp-content/uploads/2017/10/156_Paper.pdf

[2] http://bachbot.com

[3] Polyphony RNN GitHub.
https://github.com/tensorflow/magenta/tree/master/magenta/models/polyphony_rnn

[4] Albert Meronyo's curated MIDI source list
https://github.com/albertmeronyo/awesome-midi-sources?fbclid=IwAR05BypikNFXncSk8M4UZbw6l82sEU31lhvyVoQk5x-WDIv7c2-z9pZHtqM

[5] Shakespeare LSTM
https://github.com/kangeunsu/ArtML/blob/master/text-generation-LSTM/shakespeare-LSTM/

CODE

GitHub Link: https://github.com/ananyachandra/FinalProject

RESULT

Link to final song: http://www.andrew.cmu.edu/user/egoebel/BaroqueRevivalFinal.m4a