**CIS731 – Final Project Report**

**Automatic music generation using neural network architecture**

**1 group member (Aneesh Phatak)**

**0.Abstract**

**Deep Learning has a variety of applications. One of many such is the field of automatic audio and music generation. There is less research in this field as compared to other very popular areas like NLP and CV. In this project I implemented a neural network(NN) architecture to produce a piece of music automatically. My project is heavily influenced by prior research in this field and needs to be looked at as a recreation or borrowal of the existing development in this field. We will explore how the dataset was generated, transformed to a format which was readable by the NN and how output i.e. music was generated. We will also explore how the potential architectures for NN were compared and evaluated.**

**1.Introduction**

For any deep learning (DL) problem, it is important that the size of input is manageable and not too huge so as to overburden the machine with longer train time and processing power. In view of this, a MIDI file is the most appropriate type of file holding music data and has low processing time and file size. A MIDI file is not directly playable by popular audio playing software unlike mp3. Also, I made use ofmusic21 which is a Python toolkit used for computer-aided musicology. With music21, we can obtain musical notation of MIDI files through a logical interface. We can make our own MIDI files by creating note and chords. We will also use this library to compile the notes and chords generated by NN.

**2.Problem description**

Our problem at the very basic level includes loading the MIDI files into our environment, extracting the number of unique tones and their distribution, setting a frequency threshold for the notes to drop unimportant notes, creating input sequence for NN by reshaping it into a format supported by NN, modifying NN architecture to meet our needs and creating output sequence in a playable format. Music at its core is a bunch of tones having different frequencies. We could randomly assign frequencies to varying tones and then compile it to create a sequence of audio, but that could be more of a cacophony rather than melodious and soothing music. That is why it is necessary to have some sort of an intuition for how music should be composed and there is hardly a better way than to take inspiration from the greatest of the great in the world of classical Western music.

**3.Approach**

I collected classical music MIDI files which used piano tones and which were composed by Bach, Mahler, Mozart and other greats. These were used as gold standard that our generated music learnt from. In NLP, a language model is used to predict the next word after a certain list of words is fed to it. Similarly, in automatic music generation, a sequence of tones will be fed to the NN and it will smartly generate the next tone. The new sequence, with the new tone will be recursively fed to the NN until our desired length of tones is created. Two main approaches are quite popular. One is Wavenet and the other is LSTM **[1]**

Wavenet is a Deep Learning-based generative model developed by Google. The name generative model indicates that given a distribution of data, the model generates brand new samples.

“Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!” [2]

In addition to these two,a GRU-based model was also used in experimenting with the best combination of NN layers.

A subjective evaluation was done by opinions of 5 volunteers who evaluated the music generated from different models. The volunteers were folks with classical music interest and/or who had been amateur performers or composers. Each volunteer was given a short explanation of how music is generated and the chance to tweak models. The music style was piano-based, with tempo four notes per beats. Each volunteer listened to 16 pieces of music whose arrangement had been randomized. In conclusion, a LSTM-based model was unanimously chosen as the go-to architecture for our purpose.

After the architecture was finalized, a function that reads the array of notes and chords present in the musical file was used to load MIDI files into our environment.

**4.Data**

As mentioned earlier, the dataset used in this problem consisted of a collection of MIDI files. These contain the instructions rather than the actual audio. Hence, it occupies very little memory. That’s why it is usually preferred while processing music data.

Due to lack of a single data source which could provide me with training set, I had to download individual sets of MIDI files and then compile them. Thus, a dataset with more than 120 compositions was created. I restricted the training data to only piano-based music.

First, to get the dataset in a readable format, I got a list of all the notes and chords in each file.

After putting all the notes and chords into a sequential list and keeping only the topmost occurring frequencies, I created the sequences that will serve as the input of our network. After assigning a unique integer to every note, the integer sequence for input data is created. Finally, after the model is trained and used to create an output sequence, I converted the integers back into the notes and then compiled the predictions into a MIDI file.

**5.Results**

The music generated using the LSTM network had a robust structure but at the same time while the musical experts were less than impressed with other architectures, the LSTM-generated music could actually be used for future compositions, albeit through extensive re-formatting and improvements. This is because there are odd-sounding notes here and there in the generated music. Hence, creating the perfect piece of music retains immense scope. There is always room for improvement which I will consequently demonstrate.

**Appendix: Future Scope**

Since the implementation submission, I have added different offsets between notes. I also added more classes for each varying duration and rest classes that constitute the rest period between notes. I also experimented with added layers to the LSTM network.[3]

### 

## References

[1] <https://www.analyticsvidhya.com/blog/2020/01/how-to-perform-automatic-music-generation/> Approach attributed to Aravind Pai, downloaded on 10/22/2020

[2]

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/> , downloaded on 12/2/2020

[3]

<https://towardsdatascience.com/how-to-generate-music-using-a-lstm-neural-network-in-keras-68786834d4c5> downloaded on 10/22/2020

[4]

<https://medium.com/datadriveninvestor/music-generation-using-deep-learning-85010fb982e2> downloaded on 10/22/2020