Use of Deep Learning methods such as LSTM and GRU in polyphonic music generation

MSc Research Project
Data Analytics

Nipun Kulshrestha
Student ID: X18190758

School of Computing
National College of Ireland

Supervisor: Manaz Kaleel

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Nipun Kulshrestha
Student ID:	X18190758
Programme:	Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Manaz Kaleel
Submission Due Date:	17/08/2020
Project Title:	Use of Deep Learning methods such as LSTM and GRU in polyphonic music generation
Word Count:	5817
Page Count:	16

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	16th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Use of Deep Learning methods such as LSTM and GRU in polyphonic music generation

Nipun Kulshrestha
X18190758

Abstract

Music is an essential part of everyone's life and plays a very important role in many of the media and entertainment industries such as movies, games, television etc. These fields, especially music industry, have an extensive need of an integration of technology and a system that can assist artists in creating better music with ease. This is where the long-term structure creating capabilities of LSTM and GRU be used to create a polyphonic musical piece which can be used to come up with unique ideas for songs by musicians. It can also be used by non-musicians in case they want to create something personalised but do not have the right tools or the underlying theory knowledge of music to create. In this study, two separate neural network models created with LSTM and GRU respectively, are trained on music files and made to come up with patterns based on that pattern knowledge. Those patterns were then evaluated on parameters such as how close to a human composition can the networks predict notes and number of other conditions such as creating repetitive pattern, dissonant notes etc. The study concluded that most people could identify which is actual human composition and which is a machine generated composition, and rated the machine generated compositions at around 70% likeable. The LSTM model was able to learn the song structure such as chorus and verses, and was able to recreate those in its predictions, and GRU had more of repetitive and dissonant notes in the composition.

1 Introduction

In today's world, media industry plays a very vital and substantial part in people's lives. It is the largest growing industry at the moment with most investments being made in every sector of this industry. With the advancements in certain areas of Information and Technology such as Machine Learning, the use of deep learning algorithms has been of significant help in the analysis of data generated in this industry and has led to an overall profit maximisation. An example of this statement would be how directors of movies can estimate the total revenue their movie is going to generate ahead of the release using machine learning so that they can optimise their release parameters for maximum profit. This would help them to plan forward more efficiently and with estimated results already established for resource planning.

Not only movie industry, but also all the other industries in media such as games, news, advertising and music are utilising the full capabilities of Machine Learning and Deep learning algorithms. For example, news industry employs deep learning models on social media for stance detection on current topics. Advertising industry can utilise these algorithms for boosting and predicting their ads, how their performance would be, and perform product placements using analytics and find product correlations. Machine Learning is extensively being used in game development. Artificial Intelligence is certainly the most powerful tool that a game needs to employ to increase their overall user experience, playability and design more robust and dynamic Non-Playable Characters or NPCs But out of all these industries, an often-overlooked industry is the music industry. Although this industry is the backbone of many of the mentioned media industries, there has not been a lot of improvements and utilisations of these modern-day algorithms to solve some common problems that occur in this industry. Music is used in most of the entertainment industries to provide more relevance to a visual or scene. It is also used sometimes to create a reference or a connection to a character or product. An example to this is that how people can recognise a movie or a television series just by listening to the background music or theme. A good soundtrack or background score can highly increase the quality of content produced, and a right background score can always increase the energy and power of a scene. It is an essential need in games as well to produce more relevant sound effects and to enhance the current scene such as a boss battle.

In the recent years, this field and the little development that had been going on in employing machine learning technology in this field has been observed by many big companies and they have started engaging and investing in this field. Companies such as Google have launched a separate project named "Google Magenta" [1] where the teams from google brain work on the combination of Artificial Intelligence and creative arts and music generation. Spotify has its own lab working on the same field named Spotify CTRL Labs. Sony launched a tool named flow-machines [2] for AI assisted music, and IBM has tool named IBM Watson [3] just to name a few. This concept of creating music or art using deep learning is being termed as "Augmented Creativity". The aim here is not to replace an artist, but to augment them. It is an experiment to change the way people make art using Artificial Intelligence.

Music generation using neural networks is a field which is yet to be explored and is not extensively researched, and a great deal of the research remains to be done. With ample amount of media requiring soundtracks and background scoring, there has been an oversaturation of the music that can be created that would perfectly fit the type of visual. So from an artist's perspective, coming up with something unique every time can be quite a challenging task some times. There can be times when the artist cannot come up with good ideas. This is termed as a "creative block" in an artist's context, where it can take days, weeks or even months to come up with something as perfect as the artist desires. This research aims majorly on creating models that assist musicians and composers in coming up with a unique melody and idea to start composing during this creative block. The model can also be used by non-musicians wishing to create something personalised but have no knowledge of tools or any underlying music theory. Another motivation for a research in this field is that music is a combination of mathematical rules and structures with human creativity. And therefore, as deep learning models excel in this field, it seems a logical option to employ it in this research.

To proposed research is focussed on training the neural networks using LSTM and GRU networks in creating theory aware and more human like sounding music files called MIDI files which is an acronym for Musical Instrument Digital Interface. It will also

[1] Google Magenta: `https://magenta.tensorflow.org/`

[2] flow-machines: `https://https://www.flow-machines.com`

[3] IBM Watson: `https://www.ibm.com/case-studies/ibm-watson-beat/`

outline the major differences on how same architecture on different models such as GRU and LSTM effects the music generated and long term structure. The model intends to learn the song structure as well as the theoretical music concept known as the “circle of fifths” which deal with the note interactions in music theory to create compositions which are more closer to being composed by humans.

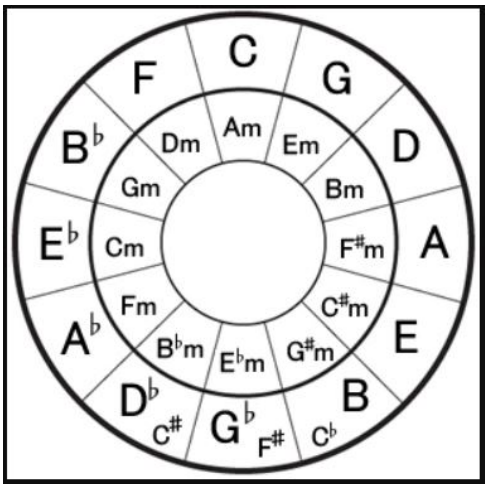


Figure 1: Circle of Fifths

So, with that being said, the main aim in question here with this research is that how well can deep learning algorithms such as LSTM and GRU be employed in generating theory based polyphonic MIDI tracks with long term structure?

The rest of the paper is divided into following sections: Section 2 is the Related work which covers a brief overview of already done research on the topic. Section 3 is the Methodology which is used in the research, followed by section 4 and 5 which are the actual Design Specification and Implementation. The next section, section 6 is the Evaluation and section 7 concludes the paper with some scope for future works.

2 Related Work

Some of the recent researches have been done in the field of neural network music generation. The researches have shown promising results, as well as there are many advanced methods explored which require much more complicated analysis of the dataset and computation resources. As many of these researches were conducted in the field of neural networks, they even took up to 1 month of continuous training even on a GPU.

One of the researches by Sigtia et al. (2015) used hybrid RNNs for generating models. By including the output of a hidden Markov model in the superimposition of a simple RNN output, they were able to calculate the joint probabilities that decide which notes get generated. This method went on to be a state of the art transcription system for the time when they researched this field. But as the problem with RNNs is that they cannot generate longer patterns, their model started getting a hit in the accuracy started showing inaccuracy.

Jaques et al. (2016) worked on a method to infuse reinforcement learning in a network of recurrent layers and then use it in the production of music. This approach they made was also recognised by a team in Google Brain, and therefore was given a chance in their project Google Magenta. This was the introduction of the concept Note RNN inclusion in the LSTM generation systems. This method was adapted to rectify a behaviour of network when using reinforcement learning where it produced a single repetitive pattern or note over and over due to getting a reward. This was a model that achieved a validation accuracy of 92 percent. This model had a log perplexity score of .2536. But as the results section pointed out, this model indeed had an improved overall score, but it was unclear that the model being used are retaining the information from the training data or not.

Another research at Google Magenta project by Dinculescu et al. (2019) introduced a new way of creating melodies by neural networks. They added a feature where a MIDI file was uploaded by user and the model generates a sample that sound similar to the one they fed input as. The team used the concept of Variational autoencoders that helped it in learning and predicting a pattern that is in the MIDI and alter it with a little variation on the analysis of overall notes and structure. The first problem with this approach was that it only had melody and no chords, and second the output would be too much similar and therefore not unique.

Members from Google Brain team Hawthorne et al. (2019) used the same Maestro dataset that I used in this study, and transcribed more than 172 hours of piano audio performances to MIDI files with a high level of accuracy. This method of transcribing was termed as Wave2Midi2Wave and used Wavenet, which is an autoregressive model and transcribed this audio into MIDI. As we can see, this study was dependent on the input being in audio waveforms, which are more time consuming to train and are also computationally expensive.

Another example of audio synthesis can be seen in the work done by Engel et al. (2019) in creating a state of the art model which was trained on Google's nSynth dataset. This model used Generative Adversarial Networks for generative modelling of audio, and used it to synthesize audio faster than the Wavenet baseline.

Roberts et al. (2018) had a good approach, where they had a hierarchical decoder that seeded the outputs from the first sequences into the next sub sequences which were independent and thereby avoided the problem of "posterior collapse" that appeared in recurrent Variational AutoEncoders.

Hori et al. (2017) used audio synthesis in their approach, but this method suggests a good way that assisted in making the model more theory aware. They used bidirectional encoder and decoder to do this. Therefore, taking an idea by this concept, the study can include LSTMs and GRUs in the model to make it more theory aware.

The research led by Jiang et al. (2019) included the use of bidirectional LSTMs in creation of less number of dissonant sounding notes getting grouped together. Instead of using bidirectional LSTMs in our system, I will be using sequential LSTM and GRU models in this proposed study.

The problem of long term structure in music can be compared to some problems that occur in Natural Language Processing. Research by Wang et al. (2019) implemented a solution which is applied in NLP. They generated the sequence of notes in order of transition probability and keep parsing it similar to any spoken language according to music grammar. This approach can be included in the model to make it create more theory based chord combinations and predictions.

Ebrahimi et al. (2019) were another researchers who created a new approach in RNN music generation. They read the notes from physical images of sheet music, which was then fed into a network of LSTMs to generate classical Persian music. As this model only fed limited theory knowledge into the system, there would be more chances of creation of dissonant notes.

Nadeem et al. (2019) in their research worked on a more advance method of MIDI generation. They took 2 LSTMs and these LSTMs were trained separately. One was trained on chords and other was trained only on melodies. This output was then combined in another LSTM dense layer and combined. The output was then written into a MIDI file. This model was evaluated by objective tests by human participants who rated that 67% of them liked the music that is being generated.

Considering all the approaches we saw earlier, it is evident that LSTMs are the most abundantly used algorithm for music generation due to its long term structure generation property. Therefore, motivated by these researches I had a question in my mind about the comparison of a neural network architecture which is like LSTM in long term memory and that brought me down to GRU. Main motive was to compare if GRU had capability to surpass the compositions created by use of LSTM so that it could be used in future to create better models for music generation.

3 Methodology

The original Google Magenta project was implemented using RNNs. The main problems that can be noted were that the compositions that were created did not have a good long-term structure capability. Therefore, the use of LSTM for this purpose is justified if a good long-term structure has to be generated. Another big aspect in music generation to be considered is the size of dataset that has to be used, as the training time and computation resources needed for the processing of deep learning networks on huge amounts of data can be a big challenge. In most of the cases where audios were used to train model for synthesis of a song, computation issues have been mentioned where even TPUs were used to train on the audio datasets for months at a time to get accurate results.

So, to tackle the computation issues, it seems logical to use MIDI files, as they have all the necessary information about the notes being played, their tempo and time signature etc, all of that in a comparatively small amount of space. This is achieved as the MIDI files do not use actual audio, but just the information about notes which makes it a lot faster and less complex to deal with. It can be dropped on any virtual instrument that supports MIDI playback and the results can instantly be generated which is why it is termed as polyphonic. A MIDI file may also contain multiple instrument tracks on a single file, and that would be played on separate tracks, all at once with different or same instruments on different tracks. Thus, MIDI can make the training and working on music a lot easier.

Basically, in this study, the study will be using LSTMs or Long Short Term Memory and GRUs or Gated Recurrent Units to tackle the problem of long-term structure in music. Both networks contain a memory gate which can store historical data. An LSTM contains 4 gates, namely input, update, forget and delete gates whereas a GRU uses only 2 gated which are update gate and reset gate to tackle long term memory problems in sequences. Using a conjunction of these properties in both the networks, we can employ them in creating more dynamic notes without forgetting the context of the ongoing scale and compare them on the basis of dissonant sounding notes and repetition of notes.

Music terminology To understand a bit more about the terms that we will be using ahead, a basic music terminology is as follows.

- Note – This is the single and the most basic unit in a MIDI file. This can be thought of as the value of the exact timestamp and the duration till when the sound from the instrument has to be played. The note decides the pitch and time for an instrument.

- Octave – This is the difference between 2 notes which are placed 11 notes apart from each other, and the interval between the pitch and its frequency of the higher note is doubled. A physical piano and piano roll typically contain 8 octaves.
- Bar – This is the most basic unit of measure in music. Every bar would contain a time segment corresponding to the number of beats.
- Chord – It is a group of harmonic sounding notes which are played together and complement each other to sound like a distinct note when played together.
- Tempo – This is the most basic setting in any Digital Audio Workstation and defines the speed with which the notes are played in it.
- Scale – This is the set of notes which are ordered in the sequence corresponding to their frequency. There are multiple scales that are created in the combinations of 7 notes chosen from the 12 in the octave.
- Velocity – Velocity is the amount of pressure that is applied on each hit of a note, and the force with which the instrument is played. Usually, it is the difference between the 2 velocities of notes which create a more human feel in music.
- Piano roll – This is a representation in a Digital Audio Workstation or MIDI software, which is laid out like a physical piano and contains octaves. A note is placed on a piano roll to be played at a particular time signature.

3.1 Dataset

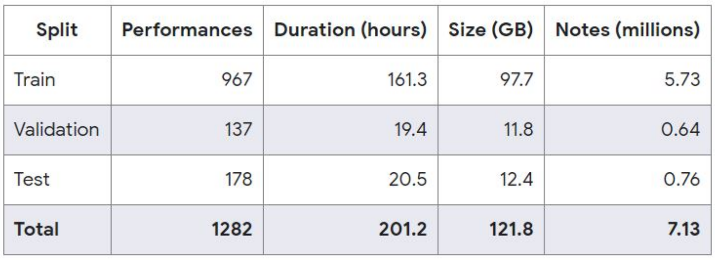
The obvious choice of suitable data for the proposed methodology would be to go with MIDI files. This choice can significantly affect in the study, as it directly reduces a lot of overhead involved in using audio datasets. This in turn, also reduces all the computational overheads and space complexity, as MIDI files as smaller than actual audio files. Another advantage of using a MIDI file for training the network is that it is not bound to a single instrument like audio and can be transferred to sound like any instrument of choice.

A dataset from Google Magenta's official website [1] was a perfect match for the type of data needed for this study. A dataset named "Maestro" [2] which is an acronym for MIDI and Audio Edited for Synchronous TRacks and Organization was used, which is a dataset of actual audio virtuosic piano performances of more than 200 hours of audio, which are finely synced with precision up to 3 milliseconds between their note labels and audio waveforms in their respective MIDI files introduced by Hawthorne et al. (2018) in the paper.

The dataset in itself is of a very large size as it contains the MIDI files from 2004 to 2018, and each piano performance would contain at least 3000 notes in a single performance. This would be an issue for a deep learning network as the size of the vocabulary would be too large for the network to train on. I have also been fascinated by the music

[1] `https://magenta.tensorflow.org/`

[2] `https://magenta.tensorflow.org/datasets/maestro`



Split	Performances	Duration (hours)	Size (GB)	Notes (millions)
Train	967	161.3	97.7	5.73
Validation	137	19.4	11.8	0.64
Test	178	20.5	12.4	0.76
Total	**1282**	**201.2**	**121.8**	**7.13**

Figure 2: Properties of Dataset

background music from Final Fantasy series, which is a science fantasy media franchise. The music of each and every part is excellently written as well as composed with the correct number of notes and emotion each of the composition. So, I decided to include some of the openly available MIDI files. Therefore, some midi files which are freely available on Bitmidi [4] were also included in the training. Due to computational limits, only a subset of the combination of this dataset from 2018 Maestro performances and Final Fantasy was used in the training. 10% of the files from the dataset was divided for test dataset to be used in evaluation.

3.2 Pre-processing

The pre-processing of MIDI files is not too difficult and daunting task like most of the text and image datasets, as there are no missing values procedures to follow. Each MIDI file will contain a timestamp, tempo and a group of notes being played together or a single melody line. With the help of libraries available from MIT on MIDI processing such as music21, the MIDI file can be flattened out and parsed directly. But some MIDI files can also contain drum parts, which are not necessary in his study as the focus of this study is not on percussion-based instruments. In order to do that, drums were removed from each midi file, and only the first instrument that appears in the MIDI file was considered as a feature in feature selection. Drums are generally on the tenth track in MIDI files. But using only the first track on the MIDI ensured that only the piano parts will be selected and not the drums.

So, after the piano track was chosen for all the MIDI files, the notes and timestamps from the sheet was parsed and flattened using music21 and saved to a pickle file for faster access. The pickle file was appended with note information on each of the MIDI files from the dataset.

3.3 Workflow

After getting the pre-processing done on the data, the cumulative data is stored in a pickle file. The pickle file contains all the notes, their pitch, duration and timestamps which are flattened out using music21. As there are multiple pitched such as C0, C1, C2 etc and these are for all the notes throughout the octave on whole piano roll. For the neural network to process easily, these notes are converted into number. A number is assigned to each note on the piano roll and mapped in a dictionary. Then input and output sequences are created and mapped to corresponding inputs and outputs.

[4] Bitmidi: `https://bitmidi.com/`

The final step is to reshape this sequence into the format of LSTM and GRU layers of 512 units. So, an array of these values is normalised and fed into the LSTM and GRU networks. 2 separate models are built and trained separately to compare the performance of both algorithms in creating sequences.

The final trained models from both networks are then used to generate a similar sequence on which the model is trained on a limit of number of notes to be generated. These generated notes are again converted back to normalised output and finally written back to a new MIDI file using music21.

3.4 Evaluation

Most of the music generative models suffer from a problem of not enough methods for objective evaluation. As music is more of a theory-based concept, it can be better evaluated only using human listeners. Very few algorithms for objective evaluations have been successfully implemented, but they require altogether different kind of output and approaches in music generation than the one used in this study. Initially, my choice of evaluation for the study was to implement a customised BLEU score with customised n-grams. This is an evaluation method used in Natural Language Programming to evaluate the quality of sentences generated by a deep learning model. Essentially, something similar is implemented where there is a need of quality of a pattern generated is to be ranked. But that method failed, as the method involved checking the n-grams with the corpus of words generated which is a large number, and in this case, the number of notes appearing in a composition would be out of 12 notes only. Therefore, BLEU score would not be a good choice in this case.

So, for this study, subjective evaluation must be considered to assess the quality of compositions generated. A questionnaire containing the audios from MIDI files as well as the test dataset is included in the form and the participants are asked to rank the compositions. The participants were a mix of musician and non-musicians to keep it non-biased.

4 Design Specification

As the model to be created will be a deep learning model, the framework to be utilised is Keras and Tensorflow for the model creation. The model employs the LSTM and GRU implementations from the Tensorflow library. As the model created for the task is complex in nature with around 512 hidden units and with multiple layers, the training time for this would take a long time and many epochs for an acceptable model on a normal CPU, so for the training a free GPU provided on a Google Colab notebook was used for reducing the training time by 75%.

The types of files that are being worked on are MIDI files, or Musical Instrument Digital Interface files, which will contain note and velocity information, these are not normal text or image files and will be needing to be flattened out using some libraries from python. Music21 is an open source library toolkit developed by MIT for working on MIDI files. This library is an all in one package and contains all the major functions that are needed for working on a MIDI file. This library can plot the piano rolls, provide an analysis on the track, set tempo and time signature for the tracks as well as create multitrack instrument MIDI files to include multiple instruments in a single track.

Additional library that was required for this research was PyGame, which is a library to create games. This library was used as this can play our generated MIDI file within the console.

5 Implementation

For the implementation of the study, the language used was python. The reason for choosing python was the extensive amounts of library support for working with midi files. The library music21 made the flattening easier. The inclusion of pickling made sure that the files can be opened easily after flattening. The models were implemented using Keras framework. Libraries GRU and LSTM were used to include the models in the final network. Both the models had exactly same number of units as well as other configurations. For the type of model, a sequential model was used with 512 units. The LSTM layers were followed by a batch normalization layer and a dropout layer of value of 0.3. This dropout layer provided the input to a fully connected dense layer. This dense layer had an activation function relu, and then the outputs were fed into a dropout and dense for normalization. The final output was collected through a softmax activation layer.

The flow diagram of both the models are given below.

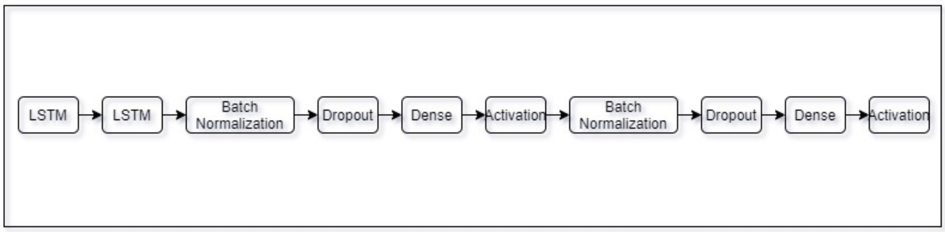


Figure 3: LSTM Model

The figure below shows the sequential model of GRU.

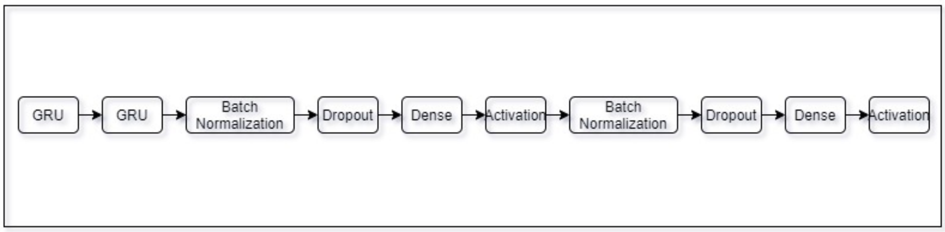


Figure 4: GRU Flow

The models were trained on the same dataset and run for a total of 200 epochs for training. The learning rate was kept to a default of 0.01 and the batch size was 128 for the training. As the training must be done on a large number of notes, the normal system would take a long time to execute. So a free machine was used on Google Colab to train as it provides a free GPU to use for training. Using a GPU, the total training time was

7 days for both the models. The weights from each epoch was saved in a file and can be reloaded. The logs are recorded for each epoch and loss is monitored.

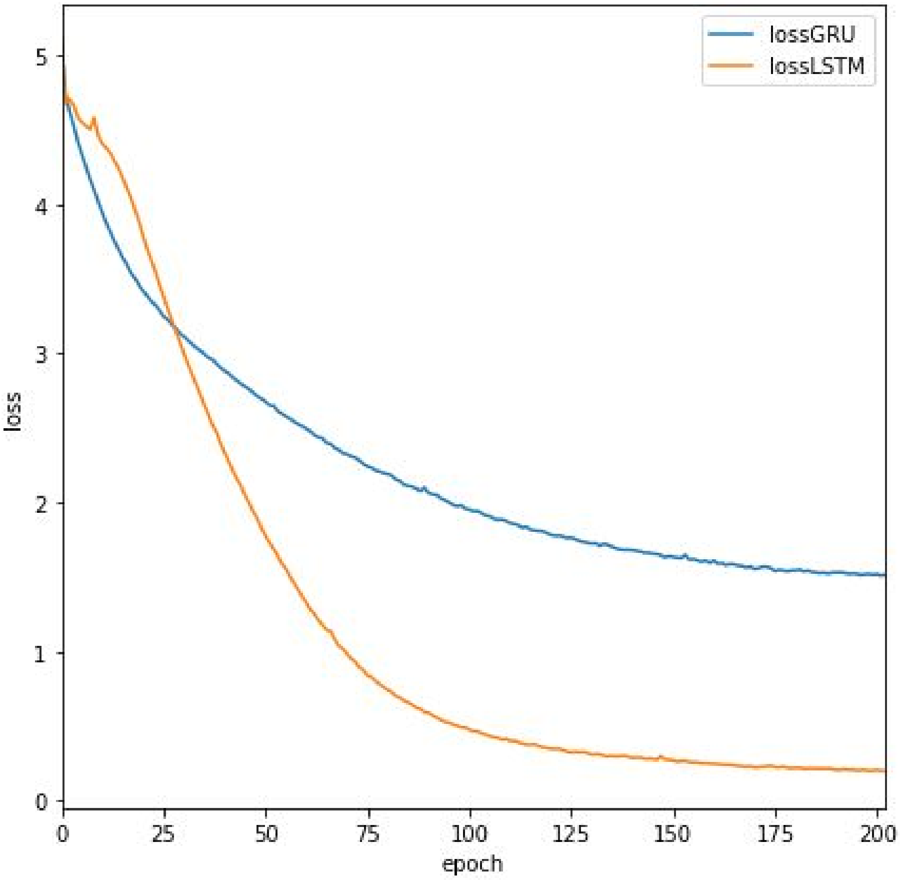


Figure 5: Loss graph - LSTM vs GRU

After the models completed on the training, a pattern with a fix number of notes, which was 500 in this study, was generated using each of the LSTM and GRU models. This output was then written to a MIDI file after parsing it back to notes and using music21 library, it was parsed back to a MIDI file. This process of generating patterns was repeated multiple times using both models. A total of 20 MIDIs were generated in total.

These MIDI files were then converted to actual audio for evaluation questionnaire. The parts of the songs generated were cropped and put together for the questions to participants. Audio 1 was from LSTM, audio 2 was from GRU and audio 3 was an actual human composition from the test dataset. The participants were asked multiple questions based on the audios. They were asked to identify which of the songs were more likely to be composed by human. Then, out of the only 2 options from LSTM and GRU, they were asked which of them was closer to a human composition. They were asked to rate melody created by both models individually. The next question was based on bad sounding notes and which audio had more dissonant and bad sounding notes. They were asked which of the compositions had more repetitive notes and were asked to rate the long-term structure of the song. They were finally asked if they were a musician or a non-musician to check for non-bias.

6 Evaluation

The melody generated by neural networks can be individually visualised and analysed. A short analysis of one of the MIDI generated by LSTM is given below. The actual midi generated is presented in the figure below. The blue bars are the notes and the black and white lines on the left are the notes of piano on the piano roll.

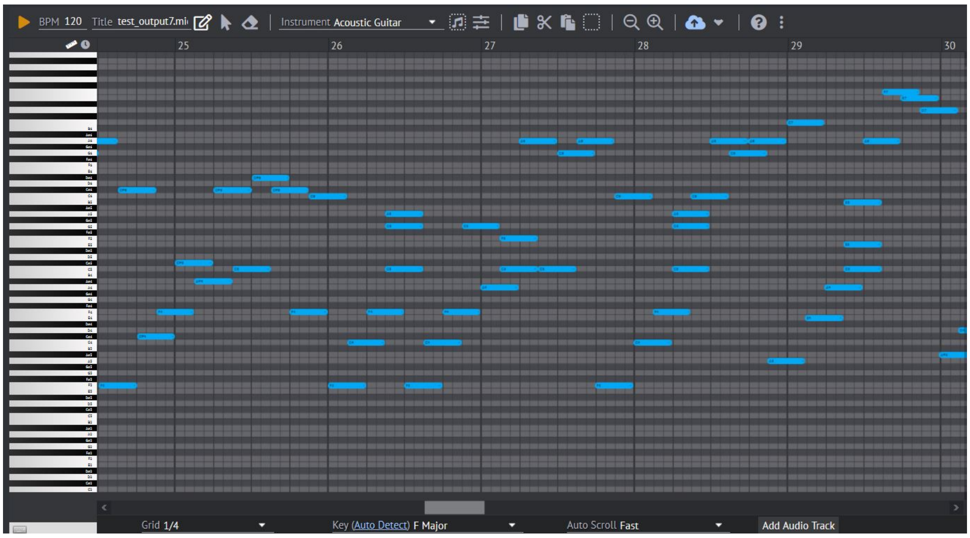

BPM 120 Title test_output7.mi
Instrument Acoustic Guitar
Grid 1/4
Key (Auto Detect) F Major
Auto Scroll Fast

Figure 6: Piano roll of the produced MIDI visualised on DAW

This is the start of the midi which is visualised in a Digital Audio Workstation. As we can see, the notes look arranged clearly and has less repetitive notes or structure.

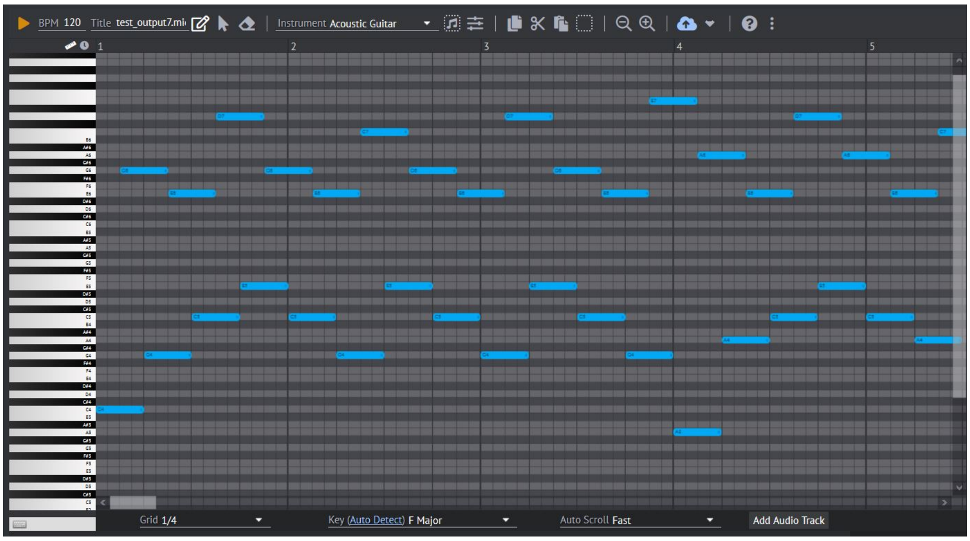

BPM 120 Title test_output7.mi
Instrument Acoustic Guitar
Grid 1/4
Key (Auto Detect) F Major
Auto Scroll Fast

Figure 7: Intro of produced output on DAW

6.1 Analysis of MIDI produced

We can conduct a collective analysis of the whole MIDI to find out the specific information about the created composition by the number of notes and find out the music key of the piece by estimation of maximum number of notes as present in the MIDI. The graph of notes is displayed below. As we see, the maximum number of notes are C, F and A which are constituents of an F major chord.

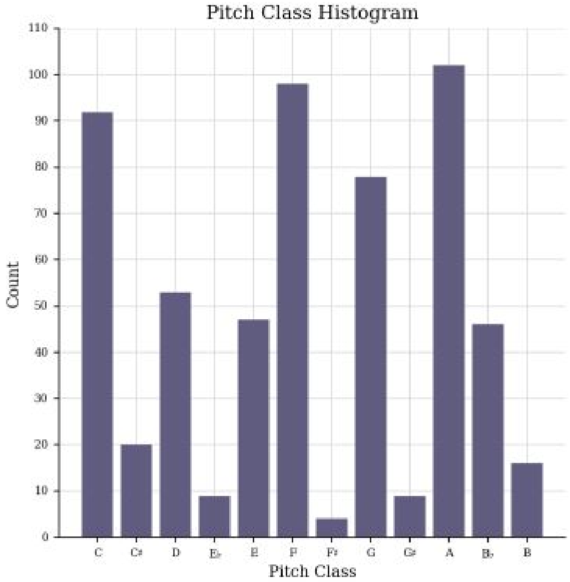
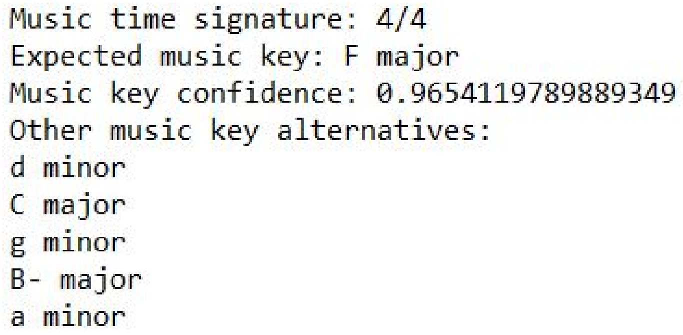


Figure 8: Notes Histogram

We can analyse this MIDI by using python for analysing alternate music key alternatives as well. Like in this case, it can also be d minor.

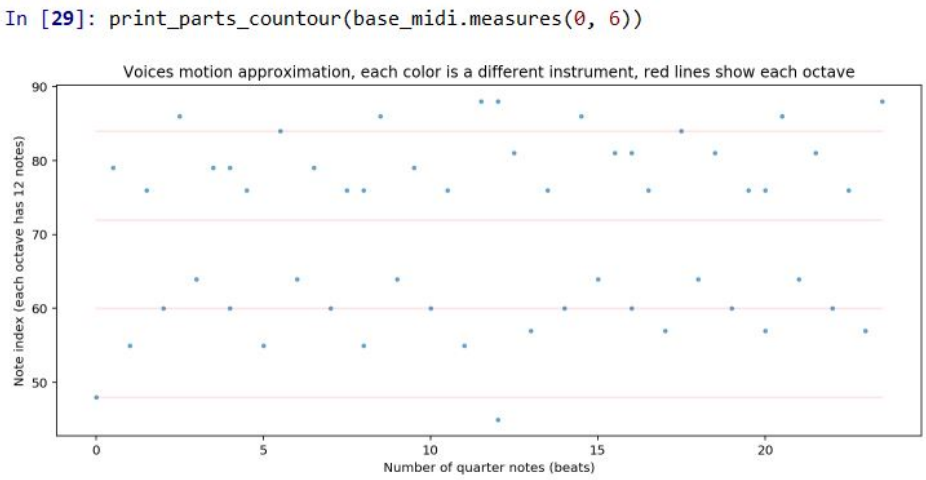


```
Music time signature: 4/4
Expected music key: F major
Music key confidence: 0.9654119789889349
Other music key alternatives:
d minor
C major
g minor
B- major
a minor
```

Figure 9: Python analysis of MIDI

6.2 Visualisation through Python

The MIDI can also be visualised using python. The figure 7 of piano roll is the same MIDI in the figure below of the first 6 bars.



```
In [29]: print_parts_countour(base_midi.measures(0, 6))
```

Voices motion approximation, each color is a different instrument, red lines show each octave
Note index (each octave has 12 notes)
90
80
70
60
50
0
5
10
15
20
Number of quarter notes (beats)

Figure 10: Visualising pianoroll in python

We can also check how the notes are scattered in the MIDI by the scatter plot below.

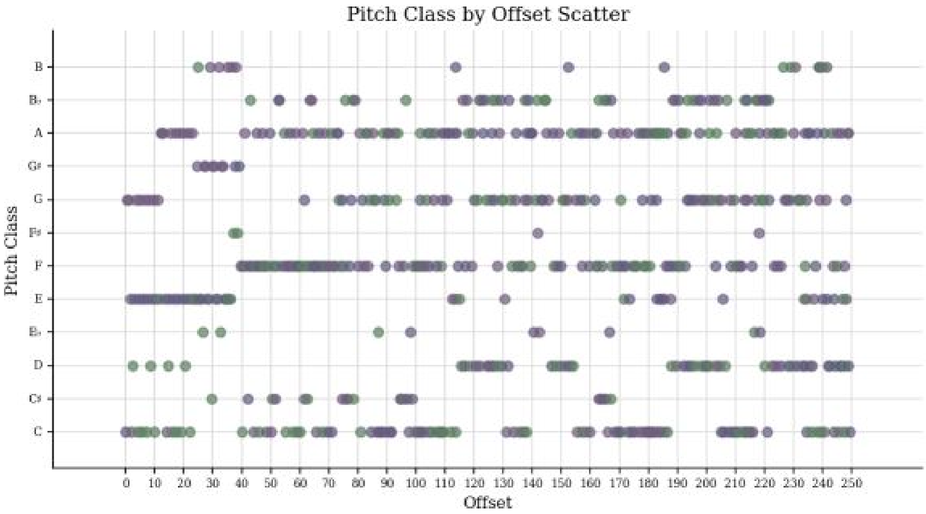


Figure 11: Scatterplot of notes in python

6.3 Discussion

There were 20 responses on the questionnaire. The results from the questionnaire were interpreted using the charts and graphs plotted by Microsoft forms based on the answers from participants. The audios in the form were as follows – Option 1 was melody generated by LSTM, option 2 was generated by GRU and option 3 was from the test dataset and original human composition.

According to the results, the votes were from participants where 5 were musicians and 15 were non musicians. When given the 3 audios, 45% of the participants could identify the actual human composition, while 35% voted for LSTM composition and 20% voted for GRU composition. Participants also voted the melody of LSTM to be 74.5% likeable and GRU to be 60%. Half of the participants voted for LSTM and half voted for GRU when asked which composition out of the 2 is closer to human composition.

On the page of repetitive structures and notes, 75% of the people voted that GRU had more repetitiveness and only 25% voted that LSTM had more repeating notes. Participants voted the long-term structure of LSTM at 73.5% and 65% for GRU. 75% participants voted that GRU had more dissonant and unpleasant sounding notes and 20% voted for LSTM, while only 5% participants voted for the actual human composition. The original detailed actual results can be viewed on this form at this link [5].

It is clear after these results that LSTM is overall better in generating melodies with less dissonance and repetitive notes than GRU. But not all patterns generated from LSTM were good and sometimes the generated melody would only contain a pattern of a few notes repeated over and over. The loss after training came down to 0.19 in 200 epochs while on the same number of epochs on GRU, the loss could only be minimised to 1.5 which could be the reason for more repetitive notes in the GRU compositions. Sometimes, the melodies would jump on from scale to scale without any scale relevance and sounded odd.

The overall achievement of the models was that they were actually able to analyse song structure and also repeated the group of notes from the first quarter of the song in the later parts of the song which could be termed as the chorus and verses. Models also identified music theory to an extent. They could not perfectly relate the "circle of fifths" theory, which defines note interactions, but were able to identify what are the notes that can generally be related and used with each other.

7 Conclusion and Future Work

The main aim of the study was to implement LSTM and GRU on the same parameters and training data and find out if neural networks are capable of creating polyphonic musical compositions that are actually closer to what a human would compose that could assist a musician or non-musician in creating a unique composition. To an extent, the LSTM is better at analysing a song structure and creating a composition with a smaller number of dissonant notes and overall a more pleasant-sounding composition than GRU on the same training provided. Most people rated the compositions created by neural networks combined (LSTM and GRU) than original human composition.

For future work, this model can be trained on the full MAESTRO dataset instead of a subset, on a system with better GPU and overall configuration as the training takes a lot of time. Another addition to the study can be the addition of multiple instruments in a single midi file to make it more efficient. Sequential LSTM model can be switched out as parallel LSTM and trained individually on chords separately and then the melody generated separately on basis of chords generated to minimise total number of dissonant notes created and better song structure.

This model can partially be used by musicians and non-musicians, as this is not totally melodic, and a sometimes created only a bunch of repetitive notes instead of anything meaningful. It can be used as a very basic melody generator and then be trained on larger

[5] Microsoft forms: https://forms.office.com/Pages/AnalysisPage.aspx?id=wUnbbnK_6k6LP6f9CiW2jIkoJwPfo01Hi89u05SR1YNUNkhTVFpUSFNSOEYzNVJNNTM1TE5FNkdUWS4u&AnalyzerToken=7AdnOtUnr6ajHwsWGnSAiy63j2d1fCBd

dataset instead of small subset due to lower computation power and longer training times which can increase the structure mapping and theory knowledge in the model.

References

Dinculescu, M., Engel, J. and Roberts, A. (eds) (2019). *MidiMe: Personalizing a MusicVAE model with user data.*

Ebrahimi, M., Majidi, B. and Eshghi, M. (2019). Procedural composition of traditional persian music using deep neural networks, *2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)*, pp. 521–525.

Engel, J., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C. and Roberts, A. (2019). Gansynth: Adversarial neural audio synthesis.
URL: *https://openreview.net/pdf?id=H1xQVn09FX*

Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, A., Dieleman, S., Elsen, E., Engel, J. and Eck, D. (2019). Enabling factorized piano music modeling and generation with the maestro dataset.
URL: *https://arxiv.org/pdf/1810.12247*

Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C. A., Dieleman, S., Elsen, E., Engel, J. H. and Eck, D. (2018). Enabling factorized piano music modeling and generation with the MAESTRO dataset, *CoRR* **abs/1810.12247**.
URL: *http://arxiv.org/abs/1810.12247*

Hori, T., Nakamura, K. and Sagayama, S. (2017). Music chord recognition from audio data using bidirectional encoder-decoder lstms, *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1312–1315.

Jaques, N., Gu, S., Turner, R. E. and Eck, D. (2016). Generating music by fine-tuning recurrent neural networks with reinforcement learning, *Deep Reinforcement Learning Workshop, NIPS.*

Jiang, T., Xiao, Q. and Yin, X. (2019). Music generation using bidirectional recurrent network, *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, pp. 564–569.

Nadeem, M., Tagle, A. and Sitsabesan, S. (2019). Let's make some music, *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–4.

Roberts, A., Engel, J., Raffel, C., Hawthorne, C. and Eck, D. (2018). A hierarchical latent vector model for learning long-term structure in music, *International Conference on Machine Learning (ICML).*
URL: *http://proceedings.mlr.press/v80/roberts18a.html*

Sigtia, S., Benetos, E., Boulanger-Lewandowski, N., Weyde, T., d'Avila Garcez, A. S. and Dixon, S. (2015). A hybrid recurrent neural network for music transcription, *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2061–2065.

Wang, J., Wang, X. and Cai, J. (2019). Jazz music generation based on grammar and lstm, *2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Vol. 1, pp. 115–120.