Music Generation using Recurrent Neural Network

Dr.R.Poorvadevi[1], Gunturu Tejasvy[2], Golla V R V N Sai Tharun[3]

[1]*Assistant Professor, Dept. Of CSE, SCSVMV (Deemed to be University), Kanchipuram, TamilNadu, India*
[2]*Student, Dept. Of CSE, SCSVMV (Deemed to be University), Kanchipuram, TamilNadu, India*
[3]*Student, Dept. Of CSE, SCSVMV (Deemed to be University), Kanchipuram, TamilNadu, India*

---***---

Abstract - *The upward thrust in computational assets and as much as the current increase in recurrent neural network architectures, music technology could currently be realistic for large-scale use of data. The most common recurrent network used for modeling long-run dependencies is the long short-time memory (LSTM) network.*

Recently. Gated Recurrent Units (GRU) had been used to successfully model the lengthy-run dependencies in a very shape of well-known collection modeling tasks. It's instead higher to say that by the use of LSTM and GRU networks for the undertaking of algorithmic music generation, it will higher produce the model in the lengthy-run artistic structure of musical dots and convey compositional art of music that sounds distinctive and musically coherent by connecting those dots.

Key Words: **Char-RNN, LSTM, GRU, MIDI, ABC Notation**

1. INTRODUCTION

Advancement in technology has led to improved methods of catering to the needs of the target users. At earlier times, people used to interact and access websites to download music. But as time passed, they became impatient to even go to a particular website that allows them to download for free. Rather they choose streaming apps that require payment on a yearly or monthly which provides them all sorts of music under one platform. But here, keeping in mind the target users, the main point is the users are satisfied if they arrive at a one-stop-shop for all their music needs. That one-stop shop is provided under the roof of Deep learning.

People have different tastes in music which can be in the sense of Genres, Composers, Instruments, Lyricists, and background scores. In this study, the generation of music is developed based on these factors using Deep learning.

1.1 Scope of the Project

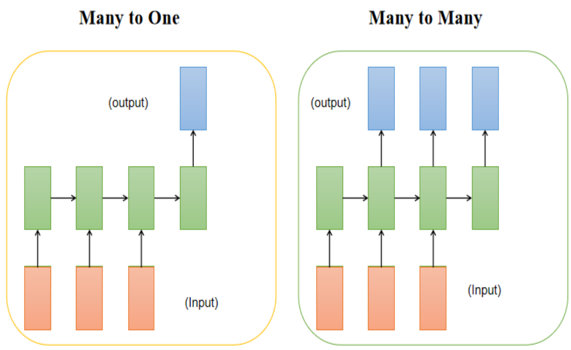
The main aim of this project is on generating music automatically employing a Recurrent Neural Network (RNN). It doesn't necessarily get to be a music expert so on getting music. Even an individual with no prior knowledge of music can generate decent quality music using RNN.

Our task here is to get some existing music data then train a model the model has to understand the patterns in music that we humans enjoy. Once it learns this, the model should be ready to generate new-music for us. It cannot simply copy-paste from the training data. It has to analyze the patterns of music to generate a new set. We here aren't expecting our model to get new music that is of professional quality, but we like it to generate decent quality music that should be melodious and good to listen to.

1.2 Char-RNN

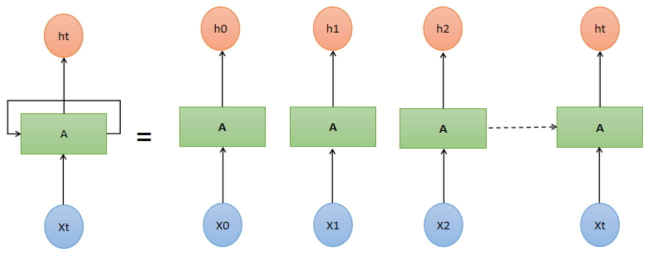
Since music is a string of characters just like any other sentence therefore the common choice will be using RNN or schemes of RNN like "LSTMs or GRUs which can process sequence information by knowing the patterns in the input".

Char RNN which is specific type of RNN is brought to use here. Now the music is a string of characters. One after the other characters of the string sequence are fed to RNN and the produced output should be the following character in the sequence. So, therefore, the number of outputs will be equal to the number of inputs. Hence, Many to-Many RNN is used.



In the above figure, Many to May RNN is the type which has equal number of inputs an outputs as shown. Here each green cell corresponds to RNN unit which has recurring mechanism

RNN would be trained in a way it predicts the output i.e. next character when provided with a first character as input. This is the main principle on how it will grasp the whole sequence and master a new sequence by itself.



A random character is given as input after the Char-RNN model completes its training. This random character is taken from a bunch of unique characters which is already fed into the Char-RNN during the training phase. Now, the model starts generating characters of the sequence automatically which relies on the sequence and pattern data that the model learns previously at the time of training.

The single RNN layer is constructed similarly where it has 256 LSTM Units in a single layer of RNN. At each time, almost all of the RNN units involve in generating outputs which will be inputted to the following layer, and also the same output will again act as the input to the same RNN unit like a repeated structure.

In Keras library in LSTM, there's a parameter referred to as "return_sequences'. It's False by default". But if it's true, then every RNN unit can generate output for every character means that at every time step.

2. Methodology

2.1 Construction of Batches

Creating batches is one of the most crucial stage in the model execution. It involves a complex use of code and understanding the underlying fundamentals linked to the character RNN. The whole model is effective with batches as the reason that makes it stand out of other models that are previously built. In this the batches are constructed based on three parameters:

- Batch Size (16)-defines the number of batches that needs to be used.
- Sequence Length (64) gives the size of the sequence that needs to be send as input.
- Number of unique characters. The distinct characters present in the music sequence in ABC format.

The first two parameters are usually user side inputs and could be seen as variables. To find the unique characters in the sequence that is fed into the RNN model, the total number of characters present initially as a whole is needed to be found. Being the most fundamental step, assignment of numerical indices to those unique characters comes the next step. This is done by initializing a mapping function, dictionary, in which key maps to a character and value maps to its index

	Batch-1	Batch-2	…	Batch-150	Batch-151
0	0.….63	64….127	….	9536….9599	9600…9663
1	9701…9764	9765…9829	…	19236…19300	19301…19364
.			….		
.			….		
14	135814..335877	125878…135941	….	145350..145413	145414…145477
15	145515..145578	145579…145642	….	155051..155114	155115…155178

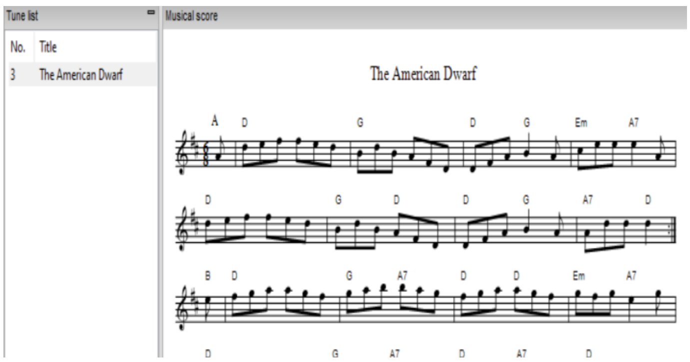
2.2 Implementation of Batches

The rows and columns table is created which is the following step once the assignment is over. The rows here indicates the first parameter used to construct butches i.e. Batch Size and the columns indicates the batch numbers allocated to it. Each value in a particular row and column indicates the music in its sequence length. If the sequence that is given as input is greater than 64, Batch-1 ends at 63 because construction of batches follow 0-indexing and Batch-2 takes it from here until twice the sequence length i.e. up to 127 and Batch-3 takes it from here until thrice the sequence length i.e. up to 191 and so on, Once the sequence gets all the batches horizontally, the next batch size comes up so the cycle starts going vertically.

Implementing them in the source involves three nested loops out of which first loop is run the numbers until the batch number which processes each time and allocates memory to a new batch. Second loop is taken in as rows in a batch and third loop is taken as columns in a batch.

2.3 Notation of Music

The notion about representing the music as string of tones where RNN is used which takes sequence as an input.



Musical Sheet

Above diagram represents sheet music notation which is a familiar category for musicians. Music here, is denoted as a string of musical chords. Each chord is delimited by a space. The advantage of sheet music explaining in this scenario is that it helps in denoting both single and multi-instrument music.

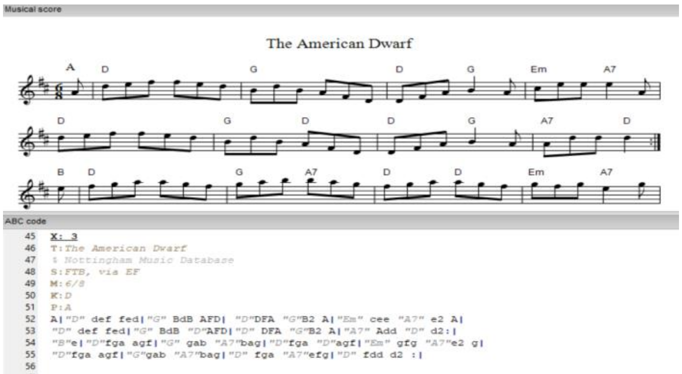
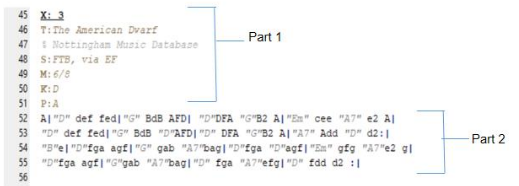


fig: ABC notation of music

ABC notation is a type of notation of music. Generally, it uses alphabets from A-G, to signify the musical notes i.e. sharp or smooth, inclined or plain, the length of the note, key, instrumentation, tempo etc. This type of notation start as a character set code that would facilitate the sharing of music on a line base and also used as an added and straightforward language for software system developers.



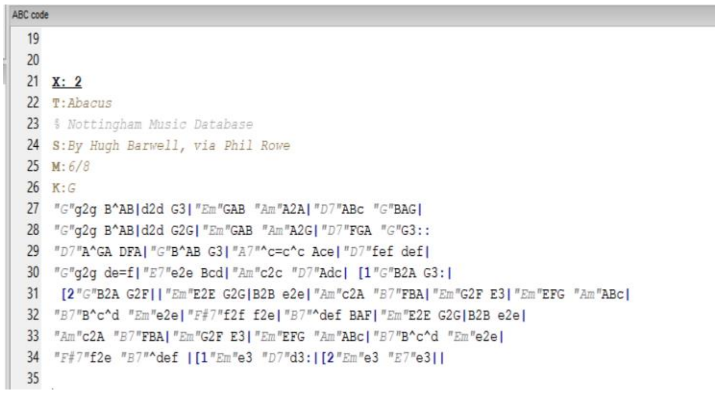
ABC notation format specifications

Parts of the notation are described below:

Part- I represents Meta data. Statements in this part often start from a single letter with semi colon after which indicates different factors of the note like:

- X: - denotes if there is multiple tunes in the file.
- T: - denotes the title of the Music.
- M: - denotes the time signature,
- L: - denotes the length of the note.
- R: - denotes the tune type.
- K: - indicates the key

Part-2 represents the tune unlike a normal music language, it shows a sequence of characters in which each character resembles a musical note just like NLP



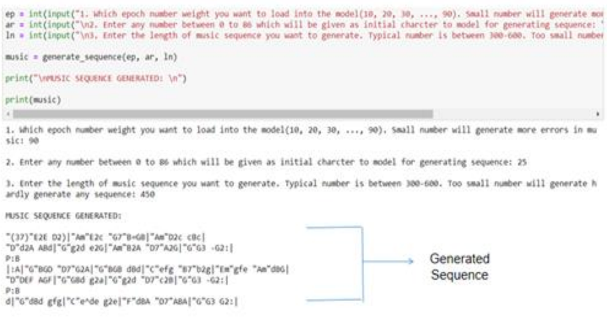
MIDI music format specifications

MIDI is simply a protocol that is one of the crucial tools for people working in the music arena. It connects various electronic gear including musical instruments, and monitors that has digitalized features to edit, cut, trim, and merge music files. The devices which use MIDI speak the same language which provides a base for communication between them. Benefits of MIDI:

- Minimized file size.
- User friendly.
- Edit performances chord by chord.
- Replace sounds.
- Control Nature.

MIDI is not something which is audible on hearing grounds, it is just a sequence of casual messages for note control, pitch control, program change, synchronization and many others. These messages generally indicate if the features are ON or OFF. An event of MIDI is a simply a message of MIDI. The devices using MIDI can be any common device that transfers MIDI information to computer over USB

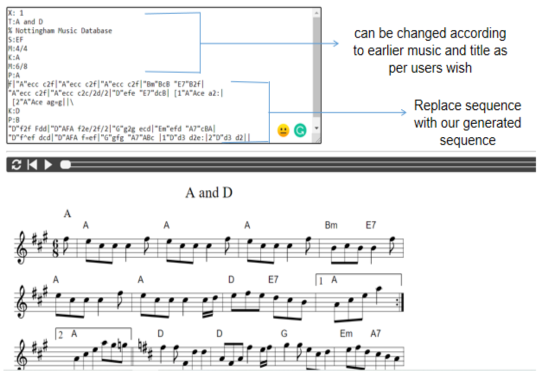
3. Results



We get three inputs from the user namely:

- Epoch number- The epoch value at the each cycle in the model during the training are stored as weight values in text file format. This is done so that no file is lost during the process and are saved at equal intervals. Larger epoch has more accuracy and thus will generate decent quality music.
- Model number: The number at which the generation has to start. Since the sequence taken has 87 unique characters, the range is therefore from 0-86.
- Length: The length of the music sequence the user wishes to have. Larger number gets a perfect sequence whereas a small number hardly generates any sequence.

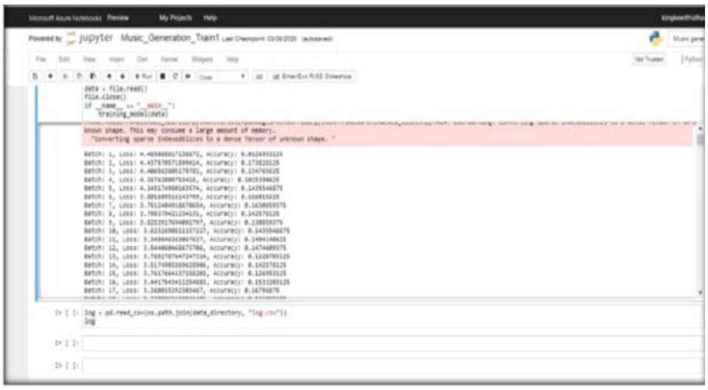
In order to hear the music, any ABC music player software needs to install on the system or any online music portal that plays ABC format music is needed. One such website is https://www.abcjs.net/abcjs-editor.html

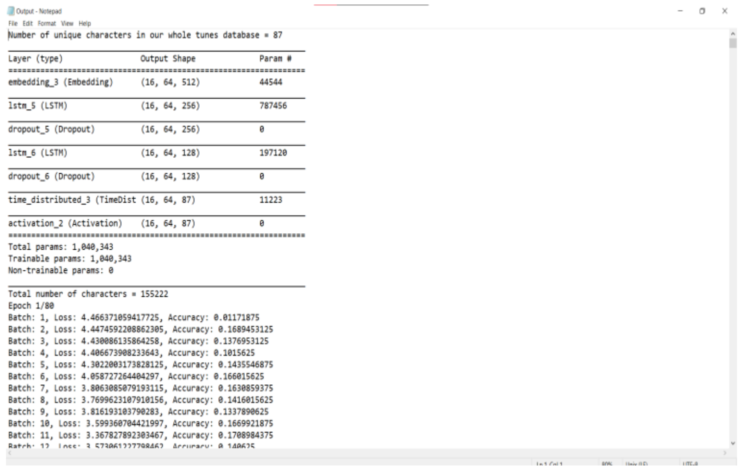


3.1 Visualization

The model is trained successfully with a better produced accuracy which is calculated taking into account the epoch values loss and accuracy. This file is saved in directory and visualized.

Output arrived when training the model is finished is shown below





4. Conclusions

Automatic Music generation with LSTM has supported lengthy sequences to execute at ease. The differed method used in this study to stand out of other writings is the use and development of batches which made memory consumption a lot less but decreases the performance of the system. One cannot simply copy paste any music so that the user can hear. It is rather a trained model which takes in mind the chords of familiar music that is heard generally among humans and executes a calculated output.

In order to make prediction, eighty seven unique characters are provided as input to the model output generates eighty seven probabilistic values by utilizing the soft max activation layer. Among the obtained eight y seven probabilistic values, next character probabilistically is chosen and not deterministically. Finally chosen character is fed back to the model and so on. The output character is concatenated and the music is generated. This is how music is generated. Larger the batch number larger the accuracy. The accuracy of the model arrived here is 89.7 'No as the most probabilistic value to predict the next character is taken to be greater than 0.5.

REFERENCES

[1] Yang, Li-Chia, Szu-Yu Chou, and Yi-Hsuan Yang. "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation." arXiv preprint arXiv .1703.10847 (2017).

[2] Briot, Jean-Pierre, Gaetan Hadjeres, and Francois-David Pachet. "Deep learning techniques for music generation--a survey." arXiv preprint arXiv .1709.0 1 620 (2017)K. Elissa, "Title of paper if known," unpublished.

[3] Kalingeri, Vasanth, and Srikanth Grand he. "Music generation with deep learning." arXiv preprint arXiv': 16 12.04928 (2016).

[4] Agarwala, Nipun, Yuki lnoue, and Axel Sly. "Music composition using recurrent neural networks." CS 224n: Natural Language Processing with Deep Learning, spring (2017).

BIOGRAPHIES

Dr.R.Poorvadevi is Assistant Professor in Computer science and Engineering department in SCSVMV (Deemed to be University).

Gunturu Tejasvy is pursuing B. Eng. from SCSVMV (Deemed to be University).

Golla V R V N Sai Tharun is pursuing B. Eng. from SCSVMV (Deemed to be University).