International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-2, December, 2019

Artificial Music Generation using LSTM Networks

Hemalatha Eedi

Abstract: ***Advancements in machine learning have minimized the gap of variation between human and algorithm composed music. This paper realizes a music generation system using evolutionary algorithms. The music generation is fully automated with no requirement of human intervention. Multiple music sample from a single dataset were used to the neural network. Software has been constructed to exhibit the results over various datasets. The proposed model is based on recurrent neural network with the input layer represents a measure at time T, and the output layer represents the measure at time T+1. The approach results in generation of new music composition by the system. Composition rules are used as constraints to evaluate the melodies generated by the novel neural network. Thus, the results are expected to evolve to satisfy the defined constraints. The proposed system of work would be capable of music generation without human intervention.***

Keywords: Machine Learning, Music Generation, Recurrent Neural Networks.

I. INTRODUCTION

This with constant attention and meaningful research breakthroughs, Deep Learning in recent times has emerged as a prime domain in Computer Science. The enormous availability of data, portable computing power and development of efficient learning algorithms has surged the demand for Machine Learning and Deep Learning [1]. Deep Learning has become a popular field in Computer Science in recent times. The current availability of huge amount of data, computing power and development of efficient learning algorithms leads the areas of Machine Learning and Deep Learning become high demand [2]. Popular Application of Deep Learning has been Machine Learning tasks such as Classification, Prediction, Translation etc. But a rapidly growing segment of Deep Learning has been generation of content. The generated content can be of any type such as images, texts, music etc [3].

Classification, Prediction, Translation etc. have been a primary task achieved using Deep Learning. Generation of Content has been a rapidly growing segment of Deep Learning. Content creation and management demands intensive manual involvement. Ranging from a couple of minutes to hour, days, weeks and even years; building an imagination into an audio visual or tangible medium needs manually planned physical contribution by diverse set of individuals with varying and relevant skill stack.

Employing comes at cost. With steep competition in the market, organizations invest massively into man power procurement. Automated content generation and management substitutes hold the potential to significantly cut expenses and improve margins made by these organizations. If not delivering the on-par quality, at least such systems can manage laborious, repeated and non-productive but essential activities.

Music is one such commonly utilized content by exponentially large number of populations across the world during their daily commute, leisure or work [4]. A simple 4 minutes track requires months of planning, trained professionals for creating and editing the final product. With diverse variety in genres of music available, every rhythm has its own niche appeal and unique set of audience appreciating the art. Considering the fact that there are more than 97 million songs in existence, there is high possibility that many of them use a generic background score, theme or rhythm. Although being relevant and repetitive to the genre, hours of effort are put into assembling the tone for the remaining project. If this can be automated with automated generation of music based on defined set of sample tunes without any interrupt to be handle and acknowledged by human, the hours and expenses spend can be utilized for more productive activities of the same project.

The current endeavor emphasizes on this very attempt to generate uninterrupted music without any human intervention. Computer generated Artificial Music first appeared in 1957. The five facets of Objective are Type, Destination, Use, Mode, and Style [5].

II. RELATED WORK

The strongest motivation for Deep Learning is its generality. As opposed to rule-based system, these machine learning systems are agnostic thereby generating an arbitrary corpus of music.

The advantages of Deep Learning systems over Rule Based Systems are:

- It can be used to generate music when the target application is too complex to formulate rules to generate music.
- Deep Learning algorithms produce more generalized music than Rule Based Systems [6].

One of the primary works in automated music generation came from veteran professor David Cope with his automated music composer designed over the course of 8 years.

Revised Manuscript Received on December 30, 2019.
* Correspondence Author
Hemalatha Eedi*, Assistant Professor, Department of Computer Science and Engineering, JNTUH College of Engineering Hyderabad, Hyderabad, India. E-mail: hemamorarjee@jntuh.ac.in

Believing all music is a form of inspired plagiarism recombining melodies from previous music work, David Cope's attention turned on an experience-based system that could hum melodies based on random notes picked from multiple tracks with some balance and understanding of music. Initial results were synthetic and noise more than music. With introduction to an analytics engine adding some randomness to the predictability of next note with huge chunks of legendary hit music as sample inputs, the system started to output rhythms that could be passed as a human playing rather than generated from an algorithm.

One of the oldest papers on music generated from deep learning, written by Chun [7], generates one music. This music has only one melody without any harmony. Dotted notes, rests and all chords were also ignored by the author. The lack of global structure in the music was cited as one of the major problems. Thus, it opens scope to address two major directions one is to create music utilizing all elements of musical rhythms and another is to create a model having the potential to learn a long-term model. Liu et al. [8] tackle the same problem but are unable to overcome either of the challenge. Eck et al. [9] and Boulanger-Lewandowski et al. [10] try to address the challenge of learning complex polyphonic structure in music.

III. PROPOSED MODEL

The proposed model generates artificial music generation using Long Short-Term Memory (LSTM) and its architecture shown in Fig. 1.

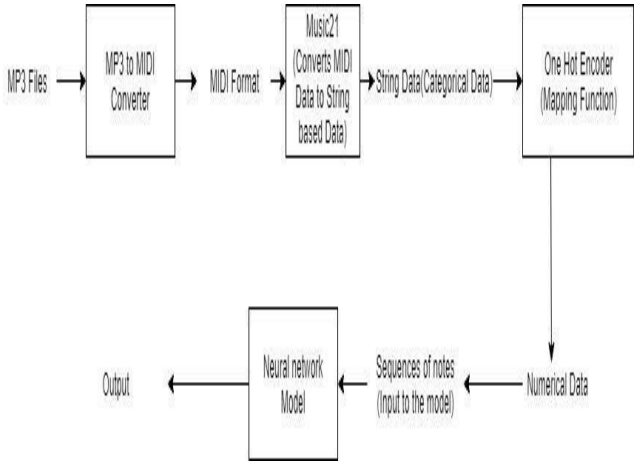

Fig. 1. Artificial Music Generation with LSTM

Artificial Composition has been a research interest for a long time. Composition of music by machines involves prediction of the notes and chords according to certain set of rules. Advances in Machine learning have enabled algorithms to compose music that is comparable to music composed by humans. Neural Networks are a natural choice for making such a prediction.

3.1 Recurrent Neural Networks

A recurrent neural network [6] is a class of artificial neural networks that make use of sequential information as shown in Fig. 2.

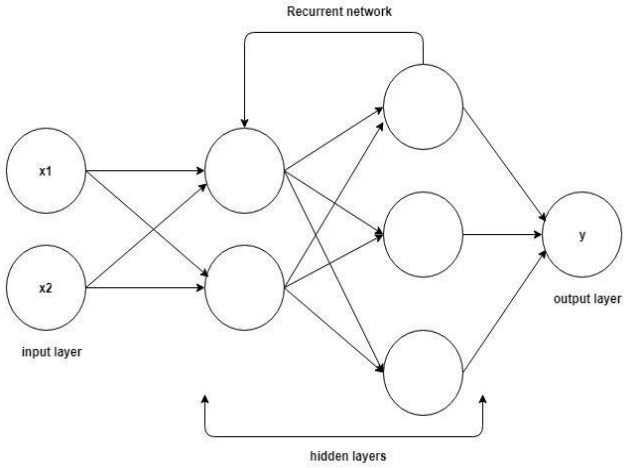

Fig. 2. Recurrent Neural Network

LSTM networks are used to generate music in this particular model.

3.1.1 Long Short-Term Memory (LSTM) Networks

LSTM has become the de facto standard for recurrent networks as shown in Fig. 3.

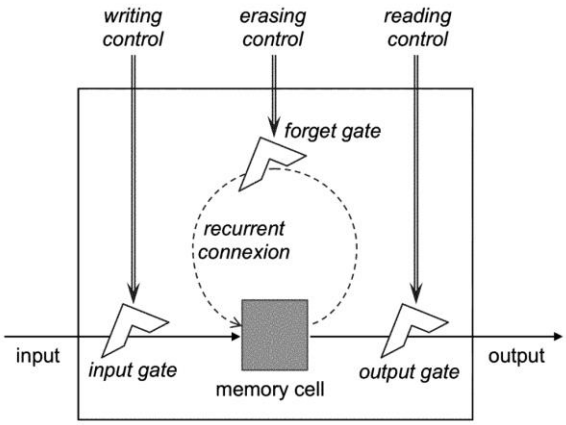

Fig. 3. LSTM Architecture

3.2 Music21

Music21 is a Python toolkit used for computer-aided musicology. It allows us to teach the fundamentals of music theory, generate music examples and study music. The toolkit provides a simple interface to acquire the musical notation of MIDI files [11]. Additionally, it allows to create Note and Chord objects so that model can make our own MIDI files easily.

3.3 Training

Training phase uses data, Music Therapy Technology, builds model and generates music.

3.3.1 Data

In this model, Piano Music is used to train the data. Soundtrack of Final Fantasy and Pokemon is used to train the data. Any set of MIDI files consisting of single instrument can be used to train the Network.

Before using any piece of music, the mp3 file has to be converted to MIDI Format. Following is an excerpt from MIDI File that has been read using Music21 as shown in Fig. 4.

```
{0.0} <music21.stream.Part 0x1ffd5f89f60>
    {0.0} <music21.instrument.Piano Piano>
    {0.0} <music21.tempo.MetronomeMark animato Quarter=120.0>
    {0.0} <music21.stream.Voice 0x1ffd6523588>
        {0.0} <music21.note.Note C>
        {0.0} <music21.note.Note B>
        {0.25} <music21.note.Note B->
        {0.5} <music21.note.Note C>
        {0.6667} <music21.note.Note B>
        {1.0} <music21.note.Note C>
        {1.0} <music21.note.Note B>
        {1.0} <music21.note.Note B->
        {1.25} <music21.note.Note C>
        {1.3333} <music21.note.Note B>
        {1.5} <music21.note.Note B->
        {1.5} <music21.note.Note B>
        {2.5} <music21.note.Note C>
        {2.75} <music21.note.Note B->
        {3.25} <music21.note.Note C>
```

Fig. 4. Sample MIDI F using Music21

3.3.2 Music Theory Terminology

- Notes: In music, a note is the pitch and duration of a sound
- Pitch refers to the frequency of the sound, or how high or low it is and is represented with the letters [A, B, C, D, E, F, G], with A being the highest and G being the lowest.
- Offset refers to where the note is located in the piece.

To generate the music accurately, proposed model has to predict which chord or note is the next. The prediction Array will have to contain every note and object that is encountered in training set.

Notes usually have varying intervals between them. There can be many notes in quick session and then followed by a rest period where no note is played for a while.

```
<music21.note.Note B> 72.0
<music21.chord.Chord E3 A3> 72.0
<music21.note.Note A> 72.5
<music21.chord.Chord E3 A3> 72.5
<music21.note.Note E> 73.0
<music21.chord.Chord E3 A3> 73.0
<music21.chord.Chord E3 A3> 73.5
<music21.note.Note E-> 74.0
<music21.chord.Chord F3 A3> 74.0
<music21.chord.Chord F3 A3> 74.5
<music21.chord.Chord F3 A3> 75.0
<music21.chord.Chord F3 A3> 75.5
<music21.chord.Chord E3 A3> 76.0
<music21.chord.Chord E3 A3> 76.5
<music21.chord.Chord E3 A3> 77.0
<music21.chord.Chord E3 A3> 77.5
<music21.chord.Chord F3 A3> 78.0
<music21.chord.Chord F3 A3> 78.5
<music21.chord.Chord F3 A3> 79.0
```

Fig. 5. Notes of Music21

As can be seen in the above Fig. 5., the most common interval between the notes is 0.5. Therefore, for simplification, the varying offset is disregarded and a constant interval is chosen as offset.

3.3.3 Preparing the Data

Before using the notes and chords as the input and output of Neural Network, data has to be prepared. First, the data has to be loaded into the array.

Each file is loaded into Music21 Stream object using converter. Parse(file) function. Using that stream object, first get a list of all the notes and chords in the file. Then append the pitch of every note object using its string notation since the most significant parts of the note can be recreated using the string notation of the pitch. Append every chord by encoding the id of every note in the chord together into a single string, with each note being separated by a dot. These encodings allow to easily decode the output generated by the network into the correct notes and chords.

After putting all the notes and chords into sequential list, sequences can be created which will serve as input for the network.

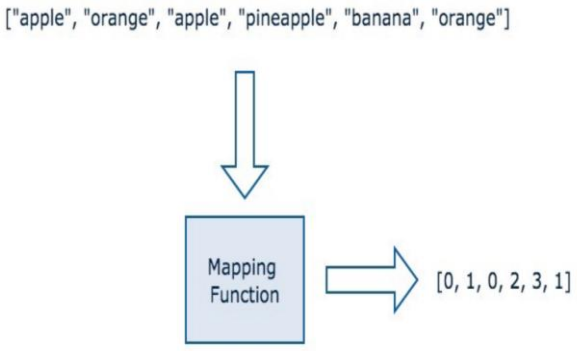

Fig. 6. Mapping Function

First, a mapping function is created to map string based categorical data to integer based numerical data as shown in Fig. 6. This is done because neural network performs much better with integer-based numerical data than string-based categorical data.

Next, the input sequences for the network and their respective output sequences have to created. output for each input sequence will be the first note or chord that comes after the sequence of notes in the input sequence in the list of notes. The length of each sequence to be 100 notes/chords. This means that to predict the next note in the sequence the network has the previous 100 notes to help make the prediction.

The final Step in preparing the data is normalize the input using One Hot Encoding.

3.4 Model

The proposed model uses LSTM layers, Dropout layers Dense layers and The Activation layer.

For each LSTM, Dense, and Activation layer the first parameter is how many nodes the layer should have.

For the Dropout layer the first parameter is the fraction of input units that should be dropped during training.
For the first layer model have to provide a unique parameter called input_shape. The purpose of the parameter is to inform the network of the shape of the data it will be training.
The last layer should always contain the same number of nodes as the number different outputs our system has. This assures that the output of the network will map directly to classes.
Once the architecture of the network has been determined, the network can be trained. The *model.fit ()* function in Keras is used to train the network. The first parameter is the list of input sequences and the second is a list of their respective outputs.

To make sure that one can stop the training at any point in time without losing work, this model uses model checkpoints. Model checkpoints provide us with a way to save the weights of the network nodes to a file after every epoch. This allows us to stop running the neural network once satisfied with the loss value without having to worry about losing the weights.

3.5 Generating Music

To be able to use the neural network for generating music, it has to be put in the same state as before. First load the weights generated from the training phase into the model
Since the full list of note sequences are at disposal, pick a random index in the list as a starting point, this allows model to rerun the generation code without changing anything and get different results every time.

Then create a mapping function to decode the output of the network. This function will map from numerical data to categorical data.

The process choses to generate 500 notes using the network since that is roughly two minutes of music and gives the network plenty of space to create a melody. For each note that to generate to submit a sequence to the network. The first sequence submit is the sequence of notes at the starting index. For every subsequent sequence that use as input will remove the first note of the sequence and insert the output of the previous iteration at the end of the sequence as shown in Fig. 7.

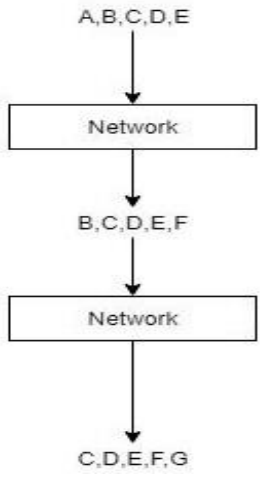

Fig. 7. Inputting Sequences to the network

To determine the most likely prediction from the output from the network, extract the index of the highest value. The value at index *X* in the output array correspond to the probability that *X* is the next note.
All the outputs from the network are collected into a single array. Sine all the encoded representations of the notes and chords are available in an array, process can start decoding them and creating an array of Note and Chord objects.
First, model has to determine whether the output decoding is a Note or a Chord.

If the pattern is a **Chord**, model has to split the string up into an array of notes. Then loop through the string representation of each note and create a Note object for each of them. Then create a Chord object containing each of these notes.

If the pattern is a **Note**, model create a Note object using the string representation of the pitch contained in the pattern. At the end of each iteration model increase the offset by 0.5 (as decided in a previous section) and append the Note/Chord object created to a list.

Since model have a list of Notes and Chords generated by the network, proposed model can create a Music21 Stream object using the list as a parameter. Then finally to create the MIDI file to contain the music generated by the network use the *write* function in the Music21 toolkit to write the stream to a file.

IV. RESULTS

After the network is trained, a random sequence is picked and is given as input. For each node model want to generate, model have to submit a sequence to the network. The first sequence model submit is the sequence of notes at the starting index. The following sequences, model will remove the first node of the sequence and insert the output of previous iteration. A mapping function is created to decode the output of the network. The number of notes is generated according to the required time duration of the output music peace. The output is determined if it is a note or a chord. Proposed model set the offset 0.5 between the notes. The output is converted to MIDI format by using music21 as shown in Fig.8.

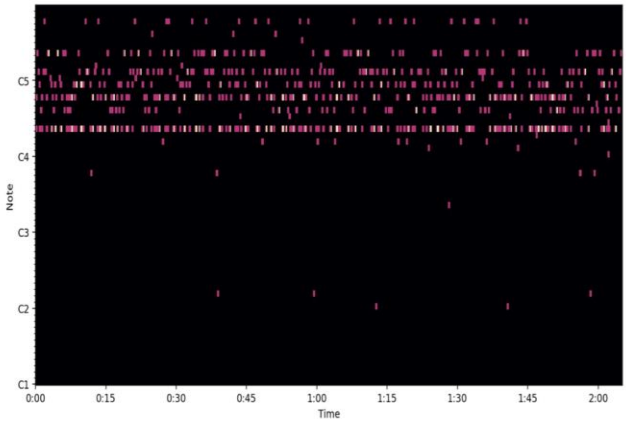

Fig. 8. Output with Time Vs Note

V. CONCLUSION AND FUTURE WORK

The proposed model uses LSTM Network to generate music. In this work, the music generated by the software is comparable to music generated by an artist. The proposed model was able to learn musical structures and was able to generate music using those structures.

For further study, Other neural networks like Boltzmann machines can also be used to generate music. Music generated by these models can be compared to music generated by LSTM networks.

REFERENCES

1. Allen Huang and Raymond Wu. Deep learning for music, June 2016. arXiv:1606.04930v1.
2. Tom M. Mitchell. Machine Learning. McGraw-Hill, 1997.
3. David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. Nature, 323(6088):533–536, October 1986.
4. Jeff Hao. Hao staff piano roll sheet music, Accessed on 19/03/2017. http://haostaff.com/store/index.php?main page=article.
5. Frank Rosenblatt. The Perceptron – A perceiving and recognizing automaton. Technical report, Cornell Aeronautical Laboratory, Ithaca, NY, USA, 1957. Report 85-460-1.
6. David Cope. The Algorithmic Composer. A-R Editions, 2000.
7. Chun-Chi J. Chen and Risto Miikkulainen. Creating melodies with evolving recurrent neuralnetworks.Proceedings of the 2001 International Joint Conference on Neural Networks, 2001.
8. I-Ting Liu and Bhiksha Ramakrishnan. Bach in 2014: Music composition with recurrent neural network. Under review as a workshop contribution at ICLR 2015, 2015.
9. Douglas Eck and Jurgen Schmidhuber. A first look at music composition using lstm recurrentneural networks.Technical Report No. IDSIA-07-02, 2002.
10. Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal de-pendencies in high-dimensional sequences: Application to polyphonic music generation and transcription. Proceedings of the 29th International Conference on Machine Learning, (29),2012.
11. MIDI Manufacturers Association (MMA). MIDI Specifications, Accessed on 14/04/2017. https://www.midi.org/specifications.

AUTHORS PROFILE

Ms. Hemalatha Eedi, working as Assistant Professor in the Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Hyderabad College of Engineering Hyderabad since 2006. Her areas of interest include Parallel and Distributed Computing, Cloud Computing, IoT, Machine Intelligence and Deep Learning.