Creating music using Machine Learning

Dehit Trivedi
Department of Electronics and Communications
Institute of Technology , Nirma University
Ahmedabad , India
18bec119@nirmauni.ac.in

***Abstract*—Music is not only just an art, but it is an expression of human emotion. When an artist is creating music we can often hear the emotions, experiences and the energy they have in that particular moment. Music connects people all over the globe and is shared across cultures. People listen to music mainly due to the emotions it evokes. Any task a person performs generates emotions. Our goal is to build a generative model from neural network architecture to try to create music which has both melody and harmony and is as similar to music composed by a composer. "If I were not a physicist , I would probably be a musician. I often think in music. I live my daydreams in music. I see my life in terms of music." -Albert Einstein. We may not be a physicist like Mr. Einstein but we totally agree with his thought on music. In this report we are going to develop an end-to-end model for Automatic Music Generation. In particular the work is based on Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM).**

***Index Terms*—Music, Emotion, Model, RNN, LSTM**

I. Introduction

Music is not just an art but an ultimate language. Many of the artist throughout the history have composed pieces that were both creative and deliberate. We define music as a collection of tones of frequencies. So, the Automatic Music Generation is a process of composing a short and sweet piece of music without any human intervention. It all started by a random selection of sounds and mixing them to form a piece of music. In 1787, Mozart discovered a Dice Game for these random sound selections. He managed to compose nearly 272 tones manually by dice game. Then, he selected a tone based on the sum of the both dice.

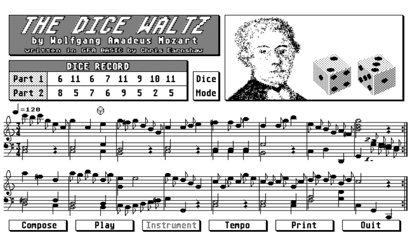

Fig. 1. Random Note from dice game

Automated Music composition has been a widely researched method of composing new music for many years now. Prior to machine learning one of the most popular approach to generate music was generative grammars. These old techniques produced simplified music pieces with non-complex melodies and inadequacies like predictable repeating melodies. Currently the modern tool for generating music is neural networks. Neaural Network [6] can trained with music to generate its own melody without human intervention. The NNs learns the pattern of the inputted music and then uses that to generate its own.

II. Neural Network

A Neural Network is defined as 'an interconnected assembly of a simple processing element. The processing in the network relies on the number or the weights of connections between those nodes which are trained by or adapt to the training data set and this process is often known as training the network or learning. Neural Networks are the ideal tool for classification problems. The benefits of Neural Network in classification are due to several things. The NNs are very adaptive as they adjust according to the data inputted. They are capable of identifying the functions which relate attribute to categories with some degree of accuracy. NNs are too flexible to model complex real world classification problems like prediction , image and speech recognition etc. Not all NNs are the same however, and the most common types of NNs are split into two wide categories of recurrent neural network (RNN) and feed forward (FF). In FF NNs the activation flows via the input layer to output layer without any form of recursion. There is only one hidden layer between the that is 'deep' FF NNs means that there are more hidden layers but with the same concept of FF NNs. The nature of FF NNs is unsuitable to music generation because there is no context or memory. No past state of the NN can affect the future states of the weights in the network. The musical melodies are essentially contextual and the depend on the memory of the earlier notes in order to generate new notes that are coherent and pleasant sounding. This is why FF NNs are completely unsuitable for automatic music composition.

A. Recurrent NNs

RNNs have recurrent connections within the hidden layer between previous and current states in the NN.This means that it is storing information in the form of activations thereby offering a kind of memory. This quality of storing information in the form of activations thereby offering a kind of memory.

This capability of storage makes it very useful for things like image and speech processing, predictions and music compositions. One of the major drawback with the RNN is that it stores the information of only previous state. This means the context extends only one generation back. This is not very

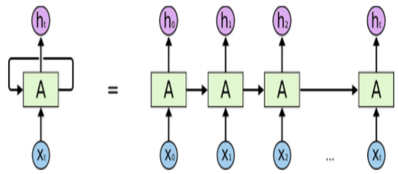

Fig. 2. unfolded RNN

useful in music composition where the beginning of the song plays a role in the middle and end as well.

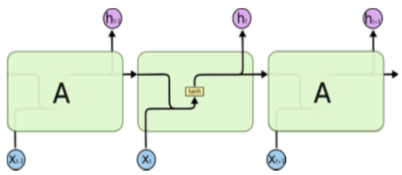

Fig. 3. RNN architecture

B. LSTM NNs

LSTM NNs [8] solve the problem of a lack of long term memory in Regular RNNs. LSTM networks add a different type of cells know as the memory cells. These cells consist of three gates i.e. input, output and forget. Input gates control the amount of data which is inputted into the memory, output gate controls the data passed to next layer and forget gate controls the loss or tearing in the store memory.

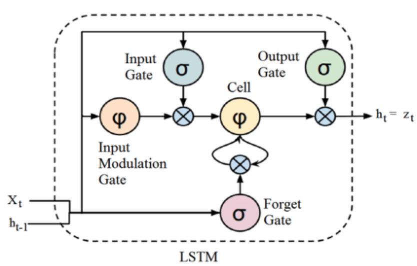

Fig. 4. LSTM Memory Cell

These gates consist of sigmoid and hyperbolic tangent functions with a low computational complexity. which means it will not be slow during training. The structure of the memory cell can be seen in Figure 4. Each gate consists of a function, the middle circle is the memory stored and to the right is the sigmoid activation function.

The function of these memory cells is that they extend the memory of the RNN and allow the network to read, write and delete the stored information based on the importance. This allows the networks to store long-term memory which is especially useful in speech recognition and music composing, making it ideal for most of the music generators. It will help us identify the melody structure and proper note sequencing. Figure 5 shows the basic architecture of LSTM network

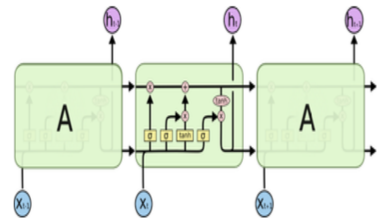

Fig. 5. LSTM Architecture

structure where each 'h' node represents the memory cell and each column represents the a stage of NN. In this network, the output of each stage is mapped to the input of the next stage, and the arrow between the memory cells represents the stored data kept between stages.Therefore this structure is ideal for music generators where a random note is entered, and the predicted output note's used as input for the next stage thereby sequencing the note and at the end composing a sweet song.

III. Methodology

The objective of the project is to create a model that is able to generate certain sequence of notes and chord, where the next node is predicted factoring the previous noted and chord. Hence a model should be created which takes a random note or chord as input and outputs a note/chord. The project is divided into three major task;

- Preparing Data
- Training
- Testing

A. Preparing Data

The music files are available in midi format which cannot be directly used into the model as model operated on numerical data only. the midi is special music format whose structure is as shown in figure 6. Hence data in midi file is to be prepared which can be used for computation by the network.

MIDI File	Chunk				
	Type	Length	Data		
	MThd	6	<format>	<tracks>	<division>
	MTrk	<length>	<delta_time> <event> ...		

Fig. 6. Structure of midi

Python provides module called "music21" [7] which can be used for converting data of midi files into notes and chords. The notes and chords are stored in categorical data which can be mapped to numerical data. Hence this numerical data can be fed to the model. The following flow diagram illustrate the process;

B. Training

The model requires training in order to generate music which is neither highly correlated nor completely random.

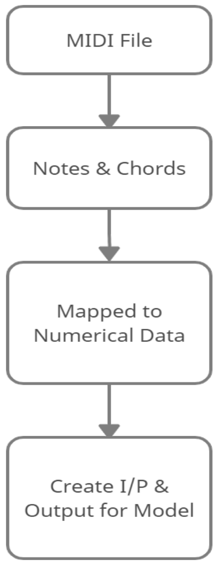

Fig. 7. Block diagram for Preparing Data

The prepared data will be divided Following block diagram illustrate the training;

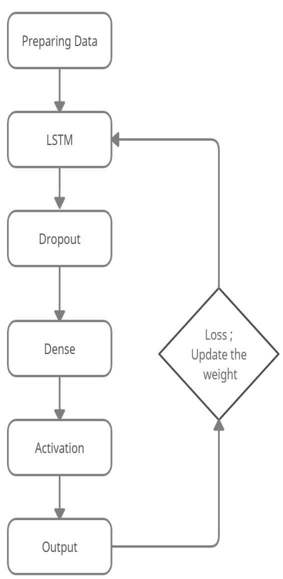

Fig. 8. Block diagram for Training

The model comprises of ;

- 3 LSTM layers
- 3 Dropout layers
- 2 Dense Layers
- 1 Activation layer

Important hyper-parameter for the model;

- Hidden Layers : They are the layers between the input and output layer. Increasing the number of hidden layer would increase the models capacity to understand complex functions.
- Number of units : It is the number number of nodes in the specified layer. More the number of units more complex can be realized by the model however it might lead to over-fitting to the training data.
- Dropout Rate : The value of the dropout rate ranges from 0 to 1 and it randomly decides the node that has to be dropped out.
- Activation Function : It computes the weighted sum of inputs and the biases. the output of the activation functions decide the neuron can be "activated" or not and can also be used to control the output of the neural network.

C. Testing

To test the model, a random note can be assigned as starting point. when the starting note is fed to model(generation code) the next sequence will be generated. Feeding every generated sequence back to the model will keep on generating a new sequence. However this output generated is numerical data and hence it has to be mapped back to categorical data and converted back to midi file.

the below is the block diagram for Testing;

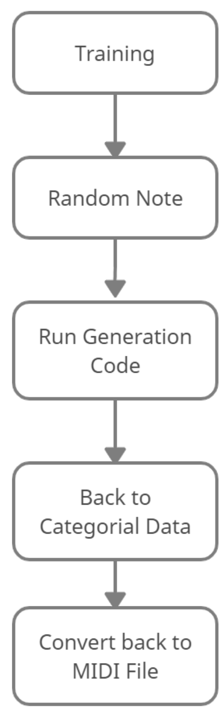

Fig. 9. Block Diagram for Testing

IV. Generative Adversarial Networks(GAN)

GAN is an alternative technique for the music generation system. It has a unique architecture where two neural networks are working against each-other. One network is called ”generator” and the other is called ”discriminator”.

The generator network creates new data instances and the discriminator networks tries to determine whether the new data instance was generated or it is another training sample. Hence it generator network will be generating new music and the discriminator network will determine if it is fake or not.

the below is the block diagram for Testing;

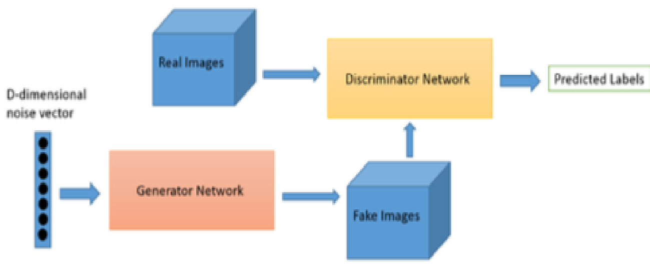

Fig. 10. GAN Architecture

From the figure 9 and the architecture of GAN it can be concluded that it is far more extensive than LSTM algorithm both computationally and complexity. If implemented with appropriate computing power and proper algorithm it is capable if producing best score created by deep learning.

V. Conclusion

Music is considered the most creative concept and a fundamental activity which set us apart as species. The algorithm discussed in this projects enable machine to create their own music. Hence narrowing the gap between human brain and CPU even more.

This project has explored the concept of Recurrent Neural Networks, Long Short Term Memory, Music Theory, MIDI files, and several literature and projects which provided a clear understanding and raised challenges which can be overcome by new innovative ideas.

There are many major projects by companies like Google, OpenAI, Microsoft and many other to develop a efficient algorithm to make machine capable of generating music. This project is not just a technical project, it has a power to bring the people from both technical and artistic background to come together and create new efficient decision rules, network models which can improve the performance of the machine to generate all kinds of music and achieve the desired goal.

References

[1] C. -. J. Chen and R. Miikkulainen, ”Creating melodies with evolving recurrent neural networks,” IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222), 2001, pp. 2241-2246 vol.3, doi: 10.1109/IJCNN.2001.938515.

[2] Eck, D. and Schmidhuber, J. (2002). A first look at music composition using LSTM recurrent neural networks. Manno, Switzerland: Instituto Dalle Molle di Studi Sull’ Intelligenza Artificiale.

[3] P. Sheikholharam and M. Teshnehlab, ”Music Composition Using Combination of Genetic Algorithms and Recurrent Neural Networks,” 2008 Eighth International Conference on Hybrid Intelligent Systems, 2008, pp. 350-355, doi: 10.1109/HIS.2008.46.

[4] Liang, F. (2016). BachBot: Automatic composition in the style of Bach chorales (M. Phil in Machine Learning, Speech, and Language Technology). Cambridge: University of Cambridge.

[5] P. Y. Nikhil Kotecha, “Generating music using an lstm network,” 2018.

[6] K. S. Geoffrey Hinton with Nitish Srivastava, “Neural networks for machine learning.”

[7] MIT. Music21. Accessed: 20/04/2019. [Online]. Available: https://web.mit.edu/music21/

[8] S. H. Schmidhuber and Jürgen, “Long short-term memory,” 1997.