Automated Music Composition using LSTM

5 authors, including:

Piyush Agrawal
Bharati Vidyapeeth College of Engineering, Delhi
1 PUBLICATION **4** CITATIONS
SEE PROFILE

Nitish Pathak
Bhagwan Parshuram Institute of Technology
51 PUBLICATIONS **251** CITATIONS
SEE PROFILE

Sajal Kaushik
North Carolina State University
3 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Automated Music Generation using LSTM

Training LSTM based RNN to generate automated music

Piyush Agrawal
Information Technology
Bharati Vidyapeeth’s College of Engineering
New Delhi, India
me@ipiyush.com

Sajal Kaushik
Computer Science and Engineering
Bharati Vidyapeeth’s College of Engineering
New Delhi, India
sajalkaushik17@gmail.com

Subham Banga
Information Technology
Bharati Vidyapeeth’s College of Engineering
New Delhi, India
subhambanga26@gmail.com

Nitish Pathak
BVICAM
New Delhi, India

Shivendra Goel
BVICAM
New Delhi, India

***Abstract*— Music composition has been an interesting and active area of research in Machine Learning. In this paper we worked on generating music through LSTM (Long Short-Term Memory) based Recurrent neural network. The proposed network is built to learn relationships within MIDI (Musical Instrument Digital Interface) files that represents chords and notes of the music. The network built can be used for fully automatic compositions that can us the humans to compose different musics. The work has been implemented using Keras framework of Python which is built on the top of Tensorflow framework, Music21 library for generating objects files of music which has to feed in our architecture. The objective of this paper is to train our neural network such that we can have automated music composition through the network.**

Keywords—LSTM, RNN, MUSIC GENERATION

I. INTRODUCTION

A. RNN

A recurrent neural network is a specific type of neural network which is based on sequential information and data. Reason behind calling them recurrent because same set of weights are applied recursively over a differential graph-like structure. RNN has great application in Natural Language Processing [16].

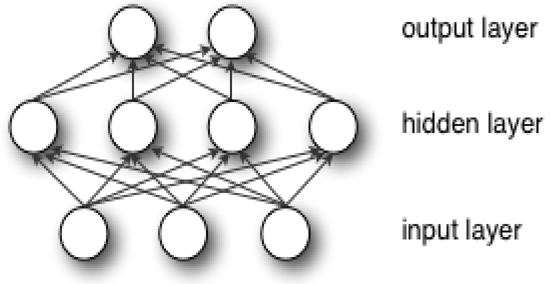


Fig. 1. Architecture of Neural Network

The decision made by RNN at *i-1* step will affect the result at ith step as well. Therefore, RNN works by taking inputs from two sources and result is produced by combining both of them.

In RNN, an information just like human brain is preserved in hidden layers. The hidden information state at any time is the function taking the present input xt and previous information in ht-1 of hidden layer both multiplied by their weight matrices.

The process of memory forwarding can be represented mathematically as :

$$\mathbf{h}_t = \phi\left(W\mathbf{x}_t + U\mathbf{h}_{t-1}\right),$$

Fig. 2. Equation for memory forwarding. [8]

B. LSTM

In this paper, a Long Short-Term Memory (LSTM) model based neural network is created. A specific type of Recurrent Neural Network is used that has the capability to avoid long term dependency problems. LSTMs are extremely useful in the situation where the neural network is required to collect and retain information for a long duration of time as it can be seen in the case of music and text generation.

The key to LSTM is the cell state which is like a conveyor belt. Only slight interactions are made through the entire chain that is running straight down and it is very simple for information to just flow through it unchanged. The LSTM [4] model has the ability to delete or join information to the cell state by regulating carefully via structures called gates.

I. Literature review

1. A LSTM comprising of character-based model [14] created by Bob Sturm which create a text based presentation of a song. This LSTM was made up of three different hidden layers and was trained with each layer consisting of 512 units.
2. Recently in 2016, A work on Wavenets which are able to generate music and speech and is based on using raw audio format files to learn and understand the complex music structure. The sounds generated through this network was more natural than any pre-existing text-to-speech recognitions. [15].
3. Doug Eck, in A First Look at Music Composition using LSTM [15]. A same set of chords present in sequences are chosen and for each note as input only one single node was present in the network so as to get the probability of note chosen being played each time. The main limitation of this neural network was incapability to rearticulate notes.

II. About Libraries and frameworks used

A. Music21

Music21 is a Python based tool used for operating on music. It allows to explain various aspects of music. The toolkit gives a musical notes of various journals. Additionally, it nodes can be created.

Our use for Music21 includes extracting the dataset which is basically MIDI files then getting the objects of notes and chord to feed into network. Further, our neural network predicted output is converted into musical notations with the help of Music21.

B. Keras

Keras is a high level API built on top of tensorflow such that it simplifies the interaction made with Tensorflow [9].

Keras library is used for creating and training our LSTM model based network. This framework's APIs simplifies the whole code by provide high level APIs to Tensorflow.

III. Method

At a high level, we feed MIDI files of music, mostly consists of Final Fantasy soundtracks. We train our tracker offline with the midi files which is used to generate music later on.

A. Input format

To implement the neural network we must first understand the input format to the network.

- We input multiple midi files which splits into two types of objects, Chords and Notes.
- A note object obtained from Music21 contains information about three things of music, offset, pitch values and octave..

Below is an excerpt of the input format.

```
<music21.note.Note F>
<music21.chord.Chord B-2 F3>
<music21.note.Note E>
<music21.chord.Chord B-2 F3 >
<music21.note.Note D>
```

and the chord objects are like container for set of notes that will be played at same time.

- Pitch: The degree of sound which determines its extent (highness or lowness) depends, which is basically the frequency of the sound. It is represented with the letters from A to G.

Octave: It is the interval between one musical pitch.

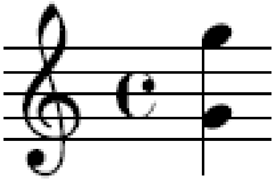


Fig. 3. An example of an octave, from G4 to G5 [11]

For music to originate the crisp work our network has to do is to be able to anticipate which note or chord has to be played next.

Another thing is the intervals between notes, as the notes can have varying intervals. Notes can occur in any form where there can be swift series for a time period followed by a pause where no note is being played.

MIDI files read using Music21 provide the interval between two successive notes. We get to see it's generally 0.5 therefore we can ignore such small offset for our experiment and it will not have much effect on the melodies of the music.

```
<music21.chord.Chord E3 A3> 77.5
<music21.chord.Chord F3 A3> 78.0
<music21.chord.Chord F3 A3> 78.5
```

B. Preparing the data

We know the format of input which is basically the MIDI files of different music which is read by Music21 to generate objects of notes and chords. This data of chords and notes is then fed into our LSTM[x4] network.

Each midi is parsed through the Music21 converter.
On parsing MIDI files we get stream objects which consists of all notes and chords. The pitch value is encoded into string notation and appended. We encode id of all notes in a chord separating them by a dot in a single string. Output of the network can be easily decoded into notes and chords due to such encoding.

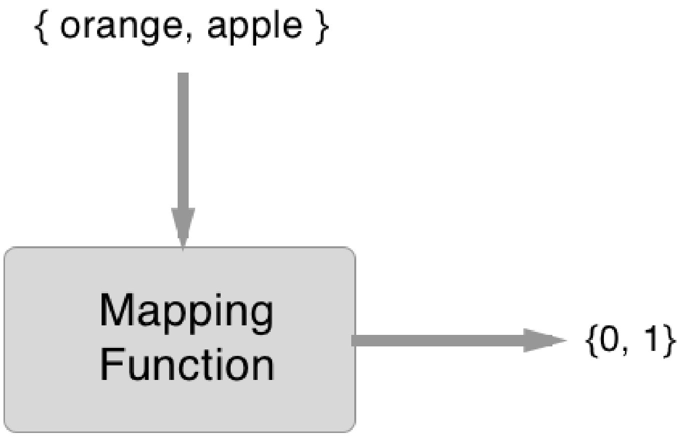


Fig. 6. Converting Categorical to Numerical data

Since our network much better with integer based value, therefore we convert this categorical data to integer-based values using one hot encoding. [12].

In the figure shown above we have a mapping function which primarily act as one-hot encoder where the categorical values like orange, apple are one-hot encoded into integer values 0 and 1 respectively. This encoded values make the Machine learning to work efficiently than the string based categorical values.

```
encoded =
to_categorical(data)
```

Fig. 8. One-hot encoding using keras

C. Network Architecture

Our model consists of these layers.

- LSTM: A Recurrent Neural Network
  A traditional human brain learns by persisting things. We do not start learning by scratch every time instead we learning things above what we have already learned.
  A traditional neural network cannot do this. For example, to understand the events happening at every point in a movie. Recurrent Network are networks with loop in them.

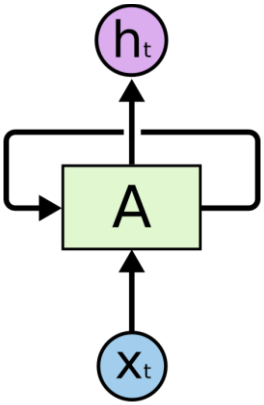


Fig. 7. A RNN's loop [13]

- *Dropout layers*
  To reduce overfitting in neural network we are using Dropout technique.
- *Dense layers*
  is nothing but a fully connected neural layer connecting each input node to output node.
- *The Activation layer*.

D. Generating Music

We will reuse the code we used for training the model to generate the music. The only difference is that instead of loading the notes and chords objects we will load the model with weights file. We will setup the network the same way as we did before.
For our music, the beginning point is selected randomly from the list but in any case if one wants to control the starting point then a function can be created to replace the current random function.

You can choose any number of notes in generated music. We chose 1000 notes. which roughly generates 4 minutes of music We provide the sequence to the network for every note we want to generate. Selecting higher number of notes will require more time to generate the music but will lead to longer music length.

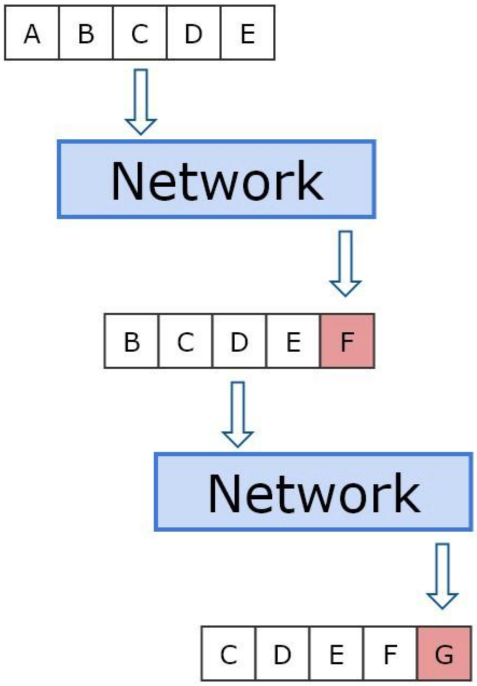
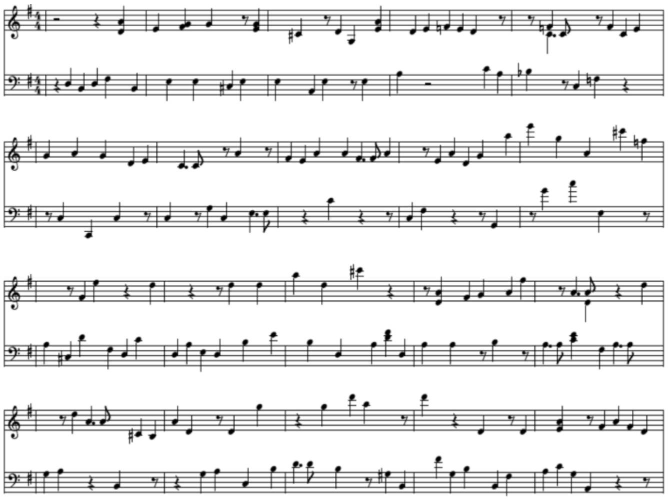


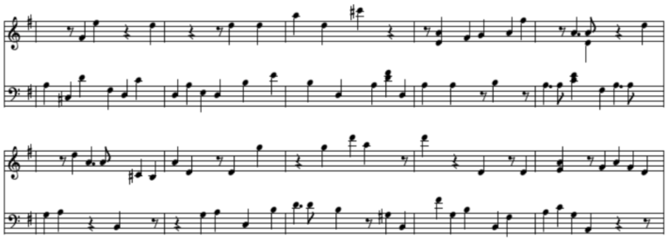
Fig. 9. Network selecting notes to generate music.

IV. Results

A test_output.mid file is generated by the network which can be played on different applications like Apple Garageband.

Some sample music generated can be listen from here https://soundcloud.com/poush12/music-lstm-1 [17]. There are some weird notes which can be seen in the output sheet. This is because of neural network incapable of making perfect melodies.





II.

Fig. 10. Notes of the music generated by network

V. Limitations

Through our LSTM network and 352 classes, we were able to achieve remarkable results. However, it can be improved in various areas.

First, we didn't considered the interval between notes to keep our network simple but to have more satisfying results. To achieve this we will have to add more classes for each and every duration and one extra class to represent to intervals.

Second, Our network need to know how to handle unknown notes. Currently, Our network would fail if encountered with a note that it is unaware of. A possible solution for this could be finding the note similar to unknown one.

More instruments can be added to the dataset to generate different types of music currently we tested it with only single instrument.

VI. Conclusions

We used simple LSTM based network to automate music generation. Results may not be perfect but are very good which shows that the neural network can be used to create music and has potential to produce higher complex musical extracts. LSTM proved to be a good model for capturing long-timescale dependencies. By providing musical note objects to our network, it was able to learn a musical style which was then used to generate the music. Future work can be done on the variants of lstm and ensemble models which will require high powered GPUs.

III. References

[1] Zaremba, W., Sutskever, I., Vinyals, O.: Recurrent neural network regularization. arXiv preprint arXiv:1409.2329 (2014)

[2] Kleedorfer, F., Knees, P., Pohle, T.: Oh oh oh whoah! towards automatic topic detection in song lyrics. In: ISMIR. pp. 287–292 (2008)

[3] Hiller, L., Isaacson, L.M.: Experimental Music. Composition with an Electronic Computer. McGraw-Hill Book Company (1959)

[4] K. Choi, G. Fazekas, and M. Sandler, "Text-based LSTM networks for Automatic Music Composition", 1st Conference on Computer Simulation of Musical Creativity, 2016

[5] Laden, B. and Keefe, D. H. (1989). The representation of pitch in a neural net model of chord classification. Computer Music Journal , 13(4):44{53.

[6] Rothstein, J. (1992). MIDI : a comprehensive introduction . Oxford University Press, Oxford.

[7] Gers, F. A. and Schmidhuber, J. (2000). Recurrent nets that time and count. In Proc. IJCNN'2000, Int. Joint Conf. on Neural Networks , Como, Italy.

[8] https://deeplearning4j.org/lstm.htm

[9] https://keras.io/

[10] https://en.wikipedia.org/wiki/Octave#/media/File:Octave_example.png

[11] https://ahmedhanibrahim.wordpress.com/2014/10/10/data-normalization-and-standardization-for-neural-networks-output-classification/

[12] http://colah.github.io/posts/2015-08-Understanding-LSTMs

[13] Sturm, Santos and Korshunova, "Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units", Late-breaking demo at the 2015 Int. Symposium on Music Information Retrieval.

[14] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” CoRR, vol. abs/1609.03499, 2016.

[15] Douglas Eck , Juergen Schmidhuber, A First Look at Music Composition using LSTM Recurrent Neural Networks, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 2002.

[16] Socher, Richard; Lin, Cliff; Ng, Andrew Y.; Manning, Christopher D., "Parsing Natural Scenes and Natural Language with Recursive Neural Networks" (PDF), 28th International Conference on Machine Learning (ICML 2011)

[17] https://soundcloud.com/poush12/music-lstm-1