Advanced Production and Industrial Engineering
R.M. Singari and P.K. Kankar (Eds.)

doi:10.3233/ATDE220793

Music Generation Using LSTM and Its Comparison with Traditional Method

Piyush Kumar Arya[a,1], Pranshu Kukreti[b] and Nilabh Jha[b]
[a] *Electrical Department, DTU, New Delhi, India*
[b] *Electrical Department DTU, New Delhi, India*

Abstract. The term music comes from the Greek translation "mousike" that stands for "the art of muses". In layman terms, it is a sequential string of sounds that people like for the purpose of entertainment, rejuvenation etc. The intriguing part of this research lies in the ability of a machine to manipulate music. The aim is to use a part of given composition (MIDI File) to generate further music via extrapolation from Long short-term memory (LSTM) which is a which is a neural network used in deep learning an d artificial intelligence, then increasing the efficiency of the model using the relation between training loss and number of epochs and finally comparing our LSTM model to traditional RNN model.

Keywords. Deep learning, neural network, recurring neural network, long short-term memory, MIDI files

1. Introduction

Melody is a sequence of notes and rest. Notes comprises of Pitch and duration. Pitch is the indicator of how high/low the note is. Pitch Notation is a combination of note name and octave. Octaves are basically different frequency of the same notes. MIDI Note Notation MIDI is a protocol to play, edit, record music. It maps the notes name to a numeric value. For encoding MIDI Inputs, each chord and note in the data with same pitch is encoded as a same variable.

There has been a lot of work going on generating music using deep learning, couple of algorithms used for this process are RNN and LSTM. Traditional method having issues like vanishing gradient and exploding gradient this paper compares both old and new algorithm used, thus finding a better algorithm for music generation.

2. Literature Review

The last years have witnessed the boom of music information retrieval software like Shazam and artificial intelligence music generation software such as AIVA and JukeDeck. This shows a step in a different direction with respect to how the general population sees music. In this section we will discuss about various works in this field that will help in future research. Eck and Schmidhuber's paper discussed the LSTM model and how did it overcome the problems faced by normal RNN cells[1]. We then

[1] Corresponding Author.

investigated various comparisons between different methods of generating music from Akanksha Dawande, Uday Chourasia and Priyanka Dixit's which helped us in analyzing the performance of different models implemented by different researchers[2]. A very common issue with RNN[3] is exploding and vanishing of gradient problem[4]. Next step to this paper in Danial Johnson's terms could be called as "Bi-axial" configuration where time axis has one LSTM cell and note axis has another LSTM cell[5].

3. Methodology

Notes are nothing but sound waves with specific combination of frequency and wavelength. We have used music21 library which was invented by people at MIT which helped us in fetching the frequency, wavelength & duration of notes. In the data pre-processing, we performed the following steps: Created a dictionary, the notes and their indices were mapped using a dictionary. The note's name was stored in the Dataset as a string. Those names were only symbols to the computer. As a result, we developed a dictionary to assign a number to each unique note in our Corpus. To retrieve the values at the time of prediction, reverse of it was done. This was used to encrypt and decode the data that entered and exited the RNN model. Encoded and segmented the dataset: Encoded and segmented the corpus into smaller, equal-length sequences: The dataset contained notes at this stage. Corpus was encoded, and tiny sequences of features and their matching targets was created. The unique characters in the dictionary were mapped such that each feature and target represented was included. Assigned X and Y: In this part we resized and standardized the labels with targets one Hot encoded. Data was now ready to be inputted in the RNN/LSTM model. Split Training and Test Dataset, we then split the data into two parts to train our model and then predict the values.

LSTM MODEL For this project research, we used LSTM. First, we initialized the model, defined the adding layer with compiling it and then trained the model. The LSTM model is trained in such a way to learn the probability of a musical note occurring at the current time.

The network's output at some time t is sent in the input again, which is based on the prior notes state until time step t-50 for retaining the past details and structure of notes[6]. The LSTM layer is dependent on the input. Training method does not apply to all notes. Only a few carefully chosen and specified notes are chosen whose occurrence is more as compared to the others for training this LSTM model, these notes are extremely useful for tweaking the model and achieving highly efficient information gain. The LSTM layer learns the mapping and connection between notes and their projection using these inputs. The Dropout layer, in addition to the LSTM layer is utilized to produce generalizations in this model.

We then checked how our model performed after it's been trained on MIDI files of piano music and plotted the learning curve and the melody generated sheet (Learning Curves are used to measure the model's performance).

As previously said, music data is complicated and difficult to express because it is made up of several qualities. Single instrumental music or tracks, especially Piano instrumental tracks, are significantly simpler to understand and process. MIDI files also aid in pre-processing since they contain metadata that may be utilized to convert the

tones into a format that is compatible with our model. As a result, we chose a piano music corpus to train our models.

3.1. Extract Notes and Chords from MIDI File.

We extracted the components from these MIDI file streams. Only the piano is included in these midi files. As a result, it is made up of either piano chords or notes. Musical notes are the fundamental elements of music, involving pitch and frequency. When a series of notes are played in a row it forms a chord. We extracted both chords and notes from the music21 stream and obtained a series of notes existing inside the musical composition. Music21 is a collection of Python-based tools for in-depth analysis and creation of musical tones and ragas. It is based on existing frameworks.

Now we have all the notes present inside our inputted midi files extracted in list of strings where each string denotes a musical note.

Figure 1. List of string denoting a musical note.

3.2. Exploring Dataset.

Now we must explore our dataset of notes. In this we examined the dataset of notes and cleaned the dataset to get the relevant notes to build our model. Firstly, we looked at the first 50 values in our loaded dataset and printed it.

```
First fifty values in the Corpus: ['F#5', 'F5', 'F#5', 'G#5', 'F#5', 'E5', 'E-5', '1.4',
```

Figure 2. First fifty values in corpus.

We then created a count dictionary and found the unique notes in the dataset by using the function Counter() which gives us the unique notes in our dataset/corpus and then using notes dictionary we found out the average recurrence of notes, maximum and minimum recurrence of notes.

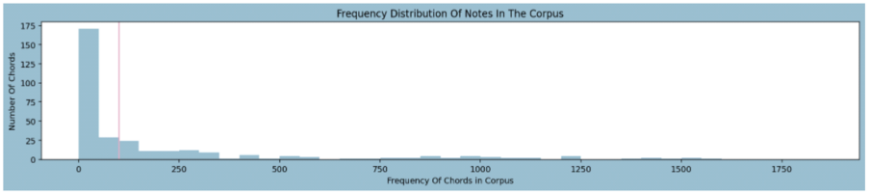

Figure 3. Frequency Distribution of Notes in the corpus.

It came out that there are some very rare notes in the melody created which were played only once in whole dataset. For tackling this, we plotted the frequency of all the notes in dataset and using that we eliminated the least occurring notes so that we don't run into some error or anomalies

4. Solution to traditional method

Being in trend this field has attracted many researches and we have got many different types of algorithms and models for generating sheet music after taking songs as a input in our case which is MIDI files. Every algorithm has some positive and some negative points. LSTM model is one step ahead of traditional RNN method. RNN model had few issues which have been improved by using LSTM cells for implementing RNN[7].

RNN model uses two inputs, first the input provided currently and the result of previous computations acts as second input. This contains three layers the input layer, the hidden layer and the output layer. Though this system successfully generates the music but still this system suffers majorly from two issues vanishing gradient problem and exploding gradient problem. After RNN predicts the output we use Back Propagation Through Time algorithm (known as BPTT) after calculating the error for calculation the gradient which updates the model[8]. Now let's say that we have T time steps for learning then the gradient of the error on the kth time step is given by

$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial h_k}\frac{\partial h_k}{\partial C_k}(\pi_{t=2}^{k}\sigma'(W_{rec}.\varepsilon_{t-1} + W_{in}.x_t).W_{rec})\frac{\partial C_1}{\partial W} \quad (1)$$

Whenever the derivative of tanh activation function is less than 1 this expression tends to vanish.

Second issue that arises is problem of exploding gradients this occurs if the weights are big enough to overpower and exceed the smaller tanh derivatives then the product of derivatives will explode

$$\pi_{t=2}^{k}\sigma'(W_{rec}.c_{t-1} + W_{in}.x_t).W_{rec} \to 0 \quad (2)$$

Here the error derivative at some kth step becomes 0 and our error gradient will vanish so no significant learning is done.

These problems are solved by LSTM. It performs the task of decision making on the basis of previous memory, current input and previous output. It has three gates that control and update the state of the cells that are forget gate, input gate and output gate. These gates use hyperbolic tangent and sigmoid activation functions. Forget gate plays a crucial role and it determines the information that is required to forget given the new information entering the network.

LSTM has a property that the cell state gradient function is an additive function which is made up from four elements A(t), B(t), C(t) and D(t).

$$\frac{\partial E_k}{\partial W} = \frac{\partial E_k}{\partial h_k}\frac{\partial h_k}{\partial C_k}(\pi_{t=2}^{k}[A_t + B_t + C_t + D_t])\frac{\partial C_1}{\partial W} \quad (3)$$

This property makes balancing of the gradient values better during backpropagation. LSTM updates and balances the four values, this additive property makes it very less likely for the gradient to vanish

5. Results

This was our trained LSTM model.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 40, 512)	1052672
dropout (Dropout)	(None, 40, 512)	0
lstm_1 (LSTM)	(None, 256)	787456
dense (Dense)	(None, 256)	65792
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 228)	58596

Total params: 1,964,516
Trainable params: 1,964,516
Non-trainable params: 0

Figure 4. Trained LSTM Model.

We plotted a training loss vs no of epochs graph to see how training loss decreased with number of epochs and for increasing the performance of our model.

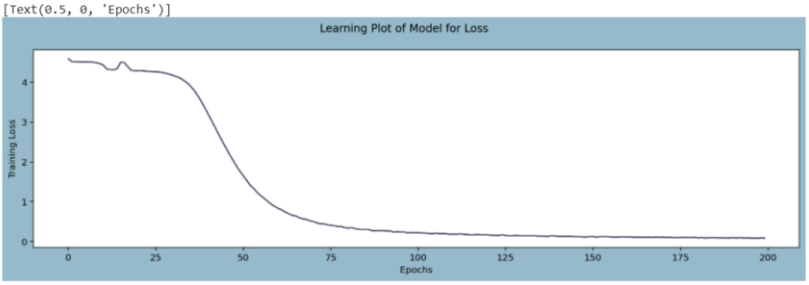

Figure 5. Learning plot of Model for Loss.

This is the music sheet that was generated as an output that we can further play to hear the generated music

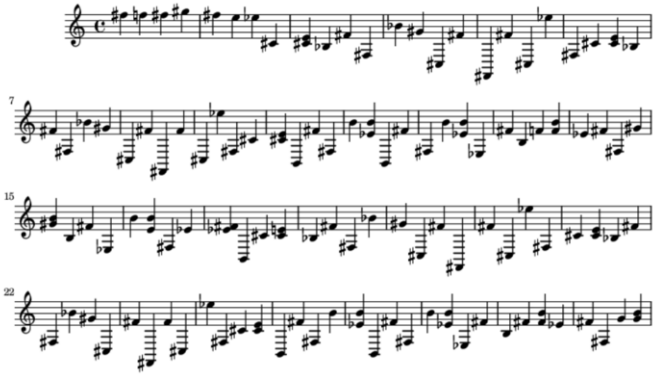

Figure 6. Sheet Music generated as an output.

6. Conclusion

Music Generation is a hot topic and many researches are in progress in this field. There are many ways to generate music and every algorithm has its negatives and positives. We have used LSTM model in our research paper, giving MIDI files as input to train our model. We saw the relation between training loss and number of epochs, the performance of model increased with number of epochs as the training loss decreased. This model generated sheet music which could be used to either play or to further generate music. Then comparing our algorithm of using LSTM cells to normal RNN cells we find out that our method solved two major issues occurring in normal RNN first being the vanishing gradient problem and second the exploding gradient problem. Thus this paper says that using LSTM cell produces better results as compared to normal RNN cells

References

[1] D. Eck and J. Schmidhuber, "A First Look at Music Composition using LSTM Recurrent Neural Networks," *Ist. Dalle Molle Di Stud. Sull Intelligenza Artif.*, vol. 103, p. 48, 2002.

[2] A. Dawande, "Music Generation and Composition Using Machine Learning," vol. 10, no. 12, pp. 151–155, 2021.

[3] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," *30th Int. Conf. Mach. Learn. ICML 2013*, no. PART 3, pp. 2347–2355, 2013.

[4] N. H. Kumar, P. S. Ashwin, and H. Ananthakrishnan, "MellisAI - An AI Generated Music Composer Using RNN-LSTMs," *Int. J. Mach. Learn. Comput.*, vol. 10, no. 2, pp. 247–252, 2020, doi: 10.18178/ijmlc.2020.10.2.927.

[5] N. Kotecha and P. Young, "Generating Music using an LSTM Network," 2018, [Online]. Available: http://arxiv.org/abs/1804.07300.

[6] D. Eck and J. Schmidhuber, "Learning the long-term structure of the blues," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2415 LNCS, pp. 284–289, 2002, doi: 10.1007/3-540-46084-5_47.

[7] M. Dua, R. Yadav, D. Mamgai, and S. Brodiya, "An Improved RNN-LSTM based Novel Approach for Sheet Music Generation," *Procedia Comput. Sci.*, vol. 171, pp. 465–474, 2020, doi: 10.1016/j.procs.2020.04.049.

[8] R. Cascade-correlation and N. S. Chunking, "Previous work," *Routledge Libr. Ed. Linguist. Mini-Set A Gen. Linguist.*, vol. 2–11, no. 8, pp. 13–35, 2013, doi: 10.3138/9781487583064-002.