Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 3 (2024)

Indian Music Generation and Analysis using LSTM

Jyoti Lele[1], Aditya Abhyankar[2]
[1] Department of Electrical and Electronics Engineering, Dr. Vishwanath Karad, MIT World Peace University, Pune, 411038, India
[2] Department of Technology, Savitribai Phule Pune University, Pune,411007, India

Abstract:- While writing a song with a distinctive melody and appropriate chords for accompaniment, a musician must experiment a lot. This work describes a machine-learning technique that uses LSTM to automatically generate chords and note sequences from a collection of Indian songs that are stored as .wav files. We propose to employ machine learning as a tool to manipulate and understand massive amounts of data generated from .wav files of various Indian songs. A brief introduction about music and its components is provided in the paper which includes notes, instrument, chords and durations. This analysis is very useful while generating sequestered musical constructs to synthesize Indian songs of any language.

Keywords*:* LSTM, Indian Music, chords, melody, music generation.

1. Introduction

For a very long time, automated music creation has been a computational challenge. Beginning in the 1980s, a number of academics attempted to produce music using grammar- and rule-based systems as well as constraint satisfaction issues [2]. A lot of experimentation has been carried out on Western music for music generation as compared to Indian music. Indian music is a rich and diverse musical tradition that has evolved over thousands of years. The Indian cultural, religious, and linguistic variety is reflected in its vast array of styles, instruments, and regional variants. Indian music is characterized by its rich melodic and rhythmic traditions which is a blend of western harmony and Indian melody. A few salient characteristics and elements of music are melody, harmony, rhythm, dynamics, timbre which work together harmoniously to produce the songs. The wide variety of musical genres and styles that are present in the world are produced by the way these elements interact in different ways. So we propose here a methodology to extract these elements from the song and use them to synthesize a song. It will help the music learners to understand different parts of music and how they are integrated into each other as well as the music arrangers to produce a song. A song can be reproduced synthetically once these parts are placed properly. Hence we have proposed here a machine learning approach to generate a song using long short term memory. Deep learning has been more and more popular for sequence modelling during the last few years [1]. In order to accomplish this, LSTM network architectures have shown to be highly effective in predicting the subsequent output in a series.

2. literature survey

Music generation with different machine learning approaches is proposed by many authors. Different methodologies like Long short term memory (LSTM), generative AI (GAN), recurrent neural network are used to generate music automatically. Few of the approaches are discussed below.

François Pachet has used LSTM for generating music. He has presented an LSTM-based model that can create expressive and coherent musical compositions by learning long-term dependencies in musical sequences. A variational LSTM-autoencoder architecture is proposed by Shih-Lun Wu et al. for generating polyphonic music with complex structures [4]. Keunwoo Choi et al. have used deep learning approach to chord detection in polyphonic music, using a dataset of over 1,200 songs. The authors use a convolutional neural network (CNN) to extract features from audio signals and a recurrent neural network (RNN) to predict the chords [9]. Satoru

Fukayama et al. have proposed a neural drum machine for creating drum tracks based on LSTM networks. The system demonstrates the versatility and adaptability of LSTM-based music generating systems by enabling users to interactively manage the generation process by modifying characteristics like style and complexity [5], [12]. While generating music the appropriate chords should be arranged for the set of melody notes given in one bar or meter. So automatic chords progression is also one of the challenging problems while generating the music. This research introduces a novel approach that uses bidirectional long short-term memory (BLSTM) networks trained on a lead sheet database to generate chord sequences from a symbolic melody [9]. They have claimed that proposed model achieves 23.8% and 11.4% performance increase as compared to HMM and DNN-HMM approach. S. D. You and P. Liu [7] have used genetic algorithm to find suitable variations of chords while generating song from given set of melody. Real-time Music Accompaniment Generation without Logical Latency nor Exposure Bias shows the potential for song creation and software development [13]. K. Zhao et al., have used Biaxial LSTM networks to generate polyphonic music, and LookBack cocept which is heart of LSTM is used in the architecture to improve the long-term structure [14]. Y. Huang, X. Huang, and Q. Cai have combined convolutional neural network and LSTM for music generation [15].

3. Methodology

3.1 Music21 library

To automate song generation the melody notes, chords, instruments should be extracted from recorded song to train the neural network. So to find these elements we used music21 library with its functions like note, pitch, chord, etc

as explained with one example blow. The only limitation of music21 is, it requires midi file to find these elements.

Note- From the 12 notes C, C#, D, D#, E, F, F#, G, G#, A, Bb, B; middle C is considered as C4 i.e. C note in 4th octave. One octave above can be denoted as C5 while below is denoted as C3. The sharp notes are denoted by lowercase or uppercase letter while flat notes are denoted by '-'. B-2 is flat note in 2nd octave. Merely note indicates the pitch.

Duration- That number represents how many quarter notes long it is. So half note Duration can be denoted as 2 or 2.0. we can also create Durations that aren't exactly "whole", "half", "quarter", etc. Let's create a dotted quarter note, which is 1.5 quarter notes long:

With all such functions we can do the analysis of given song and find different elements like melody notes, chords, durations and instruments present in a song in sequence.

3.2 Recurrent Neural Network (RNN)

RNN were created because of some limitations in the feed-forward neural network like, it Can not handle sequential data, Considers only the current input and Cannot memorize previous inputs.

Recurrent neural networks contain cycles that feed the network activations from a previous time step as inputs to the network to influence predictions at the current time step. These activations are stored in the internal states of the network which can in principle hold long-term temporal contextual information. This mechanism allows RNNs to exploit a dynamically changing contextual window over the input sequence history [6].

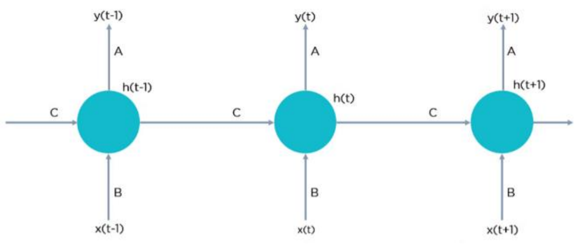

Figure 1. RNN Architecture

"x" is the input layer, "h" is the hidden layer, and "y" is the output layer. A, B, and C are the network parameters used to improve the output of the model. At any given time t, the current input is a combination of input at x(t) and x(t-1).

h(t) = fc(h(t-1), x(t))   --------------(1)

where h(t) is new state, h(t-1) is old state and nx(t) is input vector at time step t as shown in equation (1).

Recurrent neural network instead of creating multiple hidden layers, creates one and loops over it as many times as required. This makes it different from feed forward neural network. In a feed-forward neural network, the decisions are based on the current input. It doesn't memorize the past data. RNNs have a memory of past inputs, which allows them to capture information about the context of the input sequence. So they are best suitable for time series applications.

The middle layer 'h' can consist of multiple hidden layers

3.3 Long short term memory (LSTM)

Traditional RNNs performance can be restricted due to vanishing gradient problem which may occur while training. So Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture, designed to overcome the vanishing gradient problem of RNNs. LSTMs are particularly effective for processing and predicting sequences of data, such as time series data, natural language text, and audio signals. While generating a song given a set of melody notes, the corresponding chords and the next set of notes can be predicted with LSTM. Because of its capacity to successfully handle time-varying patterns and capture long-range dependencies, LSTMs are a very useful tool for modelling and analyzing sequential data.

The LSTM network architecture consists of memory cells and gates. The flow of information coming inside and going out of the memory cell is controlled by the gates namely Forget get, Input gate and output gate as shown in figure 2.

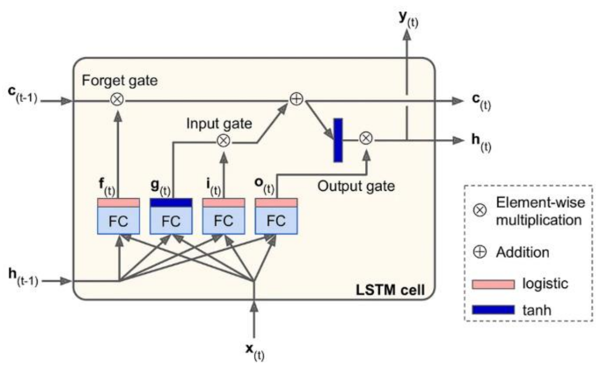

Figure 2. LSTM Architecture [10]

Forget Gate: Determines which information from the previous cell state should be discarded.

Input Gate: Decides which new information from the current input should be stored in the cell state.

Output Gate: Determines which information from the current cell state should be output as the prediction.

Since LSTM handles sequence of data, it can be used for predicting notes of a piece of music for given set of melody notes and chords.

4. Results and Discussion

The experimentation was carried out with two LSTM models; one trained with .mid file while the other trained with .wav file format. Both approaches are explained below:

4.1 Approach1

The analysis of a song in midi File format was carried out with music21. The song was 'Gulabi aankhe' from the film 'The Train' was considered for the analysis and tried to synthesize the same.

Notes = ['E6', '4.9', 'F6', 'E6', 'F6', 'E6', 'D6', 'C6', '2.7', 'B5',….]

In above piece the melody notes and chords both are present which can be separated as shown in Table 1 below:

Table 1: Melody notes and chords Analysis

Melody notes	'E6', 'F6', 'E6', 'F6', 'E6', 'D6', 'C6', 'B5'	
Chords	'4.9', '2.7'	
Melody note analysis	E6	Note - E
		Octave - 6
Chord Analysis	4.9	Notes – A4, E5
		Octave- 4,5

Chords are generally classified according to the number of notes in them, like triads, dyads, tetrads etc. Out of these the triad is the most common chord type. For a set of melody notes in a meter the same chords are played simultaneously. So while defining durations, if the duration of every melody note in a meter (with 4 notes) is 'Half note' or 0.5; the chords should be played for 2.0 i.e. 0.5*4.

4.2 LSTM logic while training:

We can select the sequence length of our choice. We set it to 100. So for every note of notes array, traced with index I, the LSTM is trained using the following logic.

seq_in = notes [i:i + sequence_length]

seq_out = notes [i + sequence_length]

The seq_in is a set of notes from notes[i] to notes [i+sequence length]. For this set seq_in, the predicted note should be next immediate note and hence seq_out will be notes [i + sequence_length]. These pairs of seq_in and corresponding seq_out are kept in memory of LSTM and training with 200 epochs, batch size of 128 and dropout 0.3 was carried out. The magnitude spectrum of the audio file generated as output of LSTM and the original file is shown in figure 3. It clearly shows the problem of overfitting and the magnitude spectrums are not matching properly. One of the probable reason for it may be: the LSTM outputs only one note/chord at a time and hence the duration and meter wise arrangement of notes and chords fails.

Another reason can be in one meter the melody notes and chords are played concurrently and not sequentially. So the output which is created sequentially will not match.

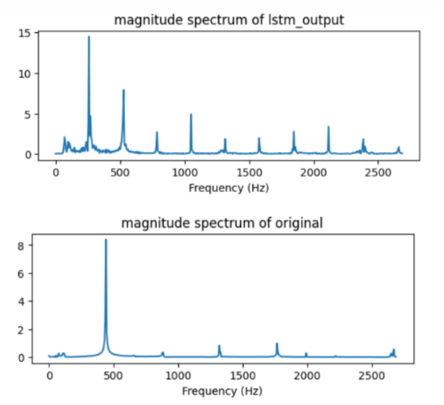

Figure 3. Magnitude spectrum of LSTM output and original song

4.3 Approach2

The another approach we tried with RNN with same logic as explained in LSTM approach but used .wav files instead of .mid files. Also the network architecture was totally different.

The first layer of LSTM model with 100 neurons and activation function as Relu was designed. It was followed by 4 dense layers each having 50, 25,12,1 neurons respectively. The network was trained with 20 epochs. The reproduced song was same like that of the original one when tested by listening. It can be shown from figure 4 and figure 5 that the magnitude spectrum calculated with Fast Fourier Transform and log-frequency power spectrogram calculated with Short Time Fourier Transform of both original song and the one generated through RNN were same.

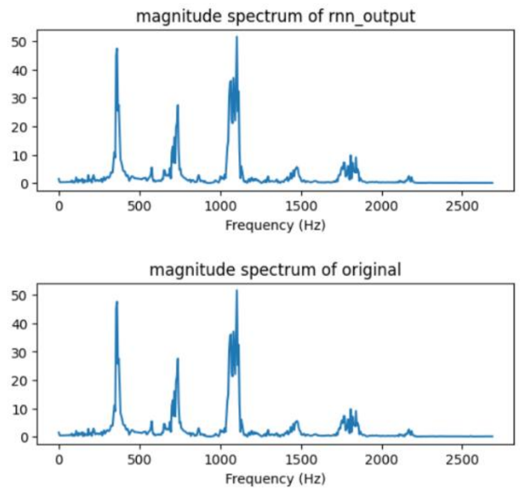

Figure 4. Magnitude spectrum of RNN output and original song

From FFT magnitude spectrum plot we can observe frequencies present in original and RNN output signal. While Spectrogram gives spectrum of frequencies varying with time.

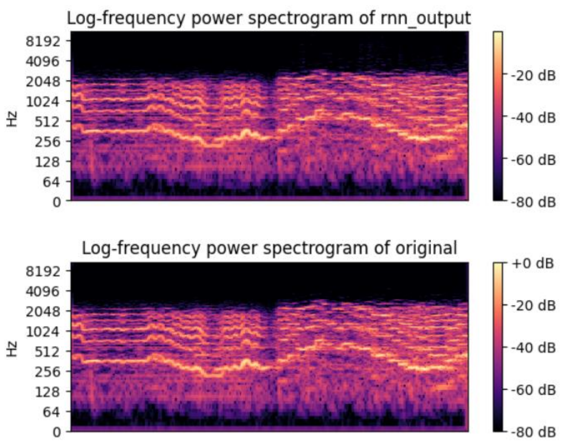

Figure 5. log-frequency power spectrogram of RNN output and original song

So this approach gave good results and the song was synthesized correctly. This approach can be utilized in future to concatenate and reproduce different songs as well to reproduce the original song. So for time series applications RNN or LSTM is the good option.

5. Conclusion

Music generation with RNN or LSTM is possible because both are best suited foe sequential data. In first approach with .mid files the analysis of a song was carried out with music21 library which gives all elements of a song like melody notes present, chords or combinations of notes like dyads, triads. Though the output of approach1 was not so promising due to overfitting and lack of proper arrangement of melody notes and chords with proper duration, this gave the direction to the research. In another approach the similar logic but with .wav format gave very good results and this methodology can be used in future to generate synthetic songs with no need of singer, instrument player or many other people involved in this process.

6. Future Work

Many improvements can be made to this project. The LSTM structure was very simple, more complex architectures of LSTM or Generative AI approach may increase the accuracy. The problem of overfitting may cause the repetition of same pattern and the original song may not be reproduced as it is for LSTM trained with midi files. As stated earlier, concurrency of chords, melody notes played by many instruments and in synchronization with time signatures is required by the music arranger. Which can be achieved by training the LSTM with a larger dataset consisting more songs but of the same kay.

References

[1] Jon Gillick, Adam Roberts, Jesse Engel, and Douglas Eck, "Music Generation Using LSTM Recurrent Neural Networks," arXiv preprint arXiv:1803.05428, 2018.

[2] E. Karagül, "Chord Progression and Music Estimation Using LSTM Networks," Senior Project Report, Bogazici University, January 7, 2019.

[3] Jesse E. C. Lee and M. W. Park: “Music Chord Recommendation of Self Composed Melodic Lines for Making Instrumental Sound,” Multimedia Tools and Applications, pp. 1-17, 2016

[4] Shih-Lun Wu, Alexander Lerch, and Yi-Hsuan Yang, "Variational LSTM-Autoencoders for Generating Complex Structures in Polyphonic Music," in Proceedings of the International Society for Music Information Retrieval Conference (ISMIR), 2019

[5] Satoru Fukayama, Ichiro Fujinaga, and Yoshihiro Miyazaki, "Neural Drum Machine: An Interactive System for Generating Drum Tracks," in Proceedings of the International Conference on New Interfaces for Musical Expression (NIME), 2019

[6] H. Sak et al., "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in Proc. of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3389-3393, 2014.

[7] S. D. You and P. Liu: “Automatic Chord Generation System Using Basic Music Theory and Genetic Algorithm,” Proceedings of the IEEE Conference on Consumer Electronics (ICCE), pp. 1-2, 2016.

[8] Y. Yi, X. Zhu, Y. Yue and W. Wang, "Music Genre Classification with LSTM based on Time and Frequency Domain Features," 2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS), Chengdu

[9] Hyungui Lim, Seungyeon Rhyu, Kyogu Lee, "Chord Generation from Symbolic Melody Using BLSTM Networks" , in Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR'2017), Suzhou, China, October 2017.

[10] https://www.oreilly.com/library/view/neural-networks-and/9781492037354/ch04.html

[11] https://www.simplilearn.com/tutorials/deep-learning-tutorial/rnn

[12] X. Liang, L. Lin, X. Shen, J. Feng, S. Yan and E. P. Xing, "Interpretable Structure-Evolving LSTM," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 2175-2184, doi: 10.1109/CVPR.2017.234

[13] Wang, Z., Liang, Q., Zhang, K., Wang, Y., Zhang, C., Yu, P., Feng, Y., Liu, W., Wang, Y., Bao, Y., & Yang, Y. (2022). SongDriver: Real-time Music Accompaniment Generation without Logical Latency nor Exposure Bias. ArXiv. https://doi.org/10.1145/3503161.3548368

[14] K. Zhao, S. Li, J. Cai, H. Wang and J. Wang, "An Emotional Symbolic Music Generation System based on LSTM Networks," 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 2019, pp. 2039-2043, doi: 10.1109/ITNEC.2019.8729266.

[15] Y. Huang, X. Huang, and Q. Cai, "Music Generation Based on Convolution-LSTM," IEEE Access, vol. 7, pp. 28743-28751, 2019.