Multimedia Tools and Applications (2022) 81:5179–5189
https://doi.org/10.1007/s11042-021-11881-1

1193: INTELLIGENT PROCESSING OF MULTIMEDIA SIGNALS

Attentional networks for music generation

Gullapalli Keerti[1] · A N Vaishnavi[1] · Prerana Mukherjee[2] · A Sree Vidya[1] · Gattineni Sai Sreenithya[1] · Deeksha Nayab[1]

Received: 1 January 2021 / Revised: 9 August 2021 / Accepted: 23 December 2021 /
Published online: 21 January 2022

Abstract
Realistic music generation has always remained as a challenging problem as it may lack structure or rationality. In this work, we propose a deep learning based music generation method in order to produce old style music particularly JAZZ with rehashed melodic structures utilizing a Bi-directional Long Short Term Memory (Bi-LSTM) Neural Network with attention. Owing to the success in modelling long-term temporal dependencies in sequential data and its success in case of videos, Bi-LSTMs with attention serves as a natural choice and early utilization in music generation. We validate in our experiments that Bi-LSTMs with attention are able to preserve the richness and technical nuances of the music performed.

Keywords Recurrent neural network (RNN) · Long short term memory (LSTM) · Attention · Bidirectional LSTM · MIDI format

✉ Prerana Mukherjee
prerana@jnu.ac.in

Gullapalli Keerti
keerti.g17@iiits.in

A N Vaishnavi
vaishnavi.a17@iiits.in

A Sree Vidya
sreevidya.a17@iiits.in

Gattineni Sai Sreenithya
saisreenithya.g17@iiits.in

Deeksha Nayab
deeksha.n17@iiits.in

[1] Indian Institute of Information Technology, Sri City, Andhra Pradesh, India

[2] Jawaharlal Nehru University, Delhi, India

1 Introduction

Artistic skills made musicians to incorporate various modern computer tools in order to make their music more better and versatile. They can thus create a variety of expressive styles that are appealing. For some imaginative purposes, gifted artists utilize conventional media or current PC apparatuses to make an assortment of expressive styles that are exceptionally engaging yet issue happens when they arrive at the bottleneck of making it more realistic. Fine subtleties is especially significant ingredient of music age. Tuning in to fascinating music and if there is some approach to produce music naturally, especially good quality music at that point, it's a major jump in the realm of music industry.

Fortunately enough, neural networks applied to music had an alternate confidence during the AI winter in 1970s. During the period from 1988 to 2009, a significant progression led to the traction in this field. AI winter refers to the era (1974–1980 and 1987–1993) when there was heavy crunch in funding and research interests in AI due to insufficient computing ability, institutional funding limitation, economic crisis, high expectations and subsequent crash observed in stock-markets, failure to adapt into next generation machines (such as collapse of LISP machine market, repulsion to new expert systems deployment etc.). The term was first coined in the annual meeting of American Association of Artificial Intelligence in 1984. The connectionist approach was being replaced by the computationalism in 1969 with the advent of high computing machines. In 2009 when the AI winter ended the deep learning works started influencing and impacting the field of music and audio AI industry. In [16], the seminal work utilized deep convolutional neural networks for music genre classification where the high level semantic concepts were extracted from melspectrograms. In [6], first end-to-end music classifier was built. It extended the idea of directly utilizing waveforms for the task of music audio tagging. It was pioneered by the work of Lewis and Todd [17, 21] in the 80s to the work of Eck and Schmidhuber [9] where we have traced a long way. Their work first utilized LSTMs in music generation. In [15], authors stated that LSTMs are able to capture the medium-scale melodic structure in music pretty well. When trained on sufficient audio data, they are also able to generate novel melodies. Due to the recent success in speech synthesis models, particularly with WaveNet [19] raw audio files are increasingly used in music generation. More recent generative models such as generative adversarial networks (GANs) or variational auto-encoders (VAEs) can even generate novel timbral spaces as well as render novel songs while directly working in the waveform domain.

In this work, we utilize an end to end pipeline based on Bi-LSTM network with an attention module to produce old style Jazz music with rehashed melodic structures automatically without any human intervention. The input to the network is the audio file in MIDI format. The key goal is to develop a model which can learn from a set of musical notes, analyze them and then generate a pristine set of musical notes. This task is a real challenge because the model should be able to register prior knowledge and generate structure of musical notes for future projection into a learning sequence. It is preprocessed and converted into the musical notes. This is then fed to a Bi-LSTM network with 512 hidden units with attention which is again followed by an LSTM layer and flattened to generate the consecutive musical notes. This can be appended to the original MIDI file. Attention mechanism gives more importance to certain parts of music which are required so that the generated output will be more synchronized with the input. Attention mechanism can be defined as the process which allows us to select relevant information from all available data. This is inspired by the cognitive ability of the human visual system to focus on salient features rather than

entire information. Global (soft) attention aims to derive a context vector based on all hidden states of the BiLSTM network. Entire input state space is utilized to reallocate weights and retrieve some critical details, motivated by this we use attention in the proposed music generation framework. The system is then evaluated by cross-entropy and MSE between input and predicted sequences. Figure 1a shows the music sheet of the song "The Last Farewell" in MIDI format. Figure 1b gives a list of musical notes in ASCII characters of the corresponding music file. Table 1 shows the batch construction for the used dataset.

2 Background and related work

The pioneering work on profound learning based music is done by Chen et al. [5], where the authors produce a music with just a single tune. The authors perform preprocessing steps in the data such as they removed speckled notes, rests, and off-tune harmonies. They addressed one of the principle issues which is the absence of structure in the generated music with machine learning based methods. In order to circumvent this issue, the two possible solutions are as follows: 1. to construct music with melodic beat, increasingly complex structure, and using a wide range of notes counting speckled notes, longer harmonies and rests. or 2. to construct a model equipped for adapting to encode long- term dependency structure and having the capacity to assemble new tune.

Liu et al. [18] also addressed a similar issue. They suggested that the music generated by their approach didn't appropriately recognize the song and different fragments of the piece and thus not able to capture the entire essence that most of the old style music pieces have. Eck et al. [10] utilized two distinctive LSTM based networks – i) to learn harmony structure and nearby note structure, ii) to realize longer dependency conditions so as to attempt to become familiar with a song and hold it all through the piece. This enabled the authors to generate music that never deviates a long way from the first harmony progression song. However, this method could just handle limited number of harmonies and cannot make a progressively assorted blend of notes. Boulanger-Lewandowski et al. [3] attempted to manage the test of learning complex polyphonic structure in music. They utilized a Recurrent Temporal Restricted Boltzmann machine (RTRBM) so as to demonstrate unconstrained polyphonic music. Utilizing the RTRBM modelling enabled them to speak to a confounded dispersion over each time step as opposed to a solitary token as in most character language

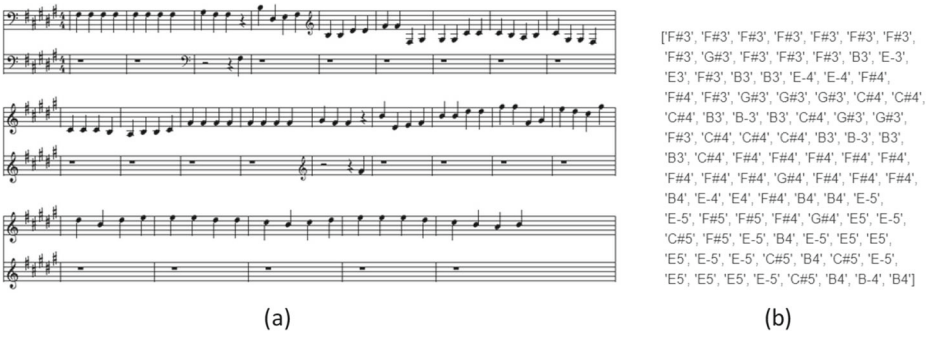

Fig. 1 a) Sheet Music of the song: "The Last Farewell" by Roger Whittaker b) Musical Notes (Extracted from MIDI file) of the song: "The Last Farewell"

Table 1 Batch construction for the JAZZ ML ready MIDI dataset: Batch size 64 characters

	Batch-1	Batch-2	...	Batch-150	Batch-151
0	0...63	64...127	...	9536...9599	9600...9663
1	9701...9764	9765...9829	...	19237...19300	19301...19364
.	...	...	...	...	...
.	...	...	...	...	...
.	...	...	...	...	...
14	135814...135877	135878...135941	...	145350...145413	145414...145477
15	145515...145578	145579...145642	...	155051...155114	155115...155178
.	...	...	...	...	...

models. This model is also able to handle the issue of polyphony in the music generated. Drewes et al. [8] proposed a strategy to utilize algebra to create music in a linguistic way with the help of tree-based models. Markov chains [20] and Markov hidden units can also be utilized to devise a numerical model to produce music.

After the leap forward in AI, numerous new models and techniques were proposed in the field of music age. Depiction of different AI empowered procedures can be found in [3, 4, 13] including a probabilistic model utilizing RNNs, Anticipation RNN and Recursive Artificial Neural Networks (RANN), an adaptation of artificial neural networks [1] for creating the consequent notes, resulting note duration and rhythm. Generative Adversarial Networks (GANs) [11] are also effectively utilized in generating melodic notes where the model consists of two networks, generator that is responsible for generating random information and discriminator that is responsible for assessing created arbitrary information for realness against the original data. MuseGAN [7] is a generative adversarial network that creates representative multi-track music. Next, in the subsequent sections we provide the background on different components relevant to this work.

2.1 RNN

Recurrent Neural Networks (RNNs) include intermittent associations inside the hidden layers between past and current states in the neural network. This capacity of memory storage makes it extremely helpful in applications such as discourse handling and music composition. The primary issue with a standard RNN is that it stores the data of just the previously attended state; this implies the setting expands just a single strand back. This isn't extremely helpful in music composition where the start of the tune may be quite significant than in the center and the end too.

2.2 LSTM

Long Short Term Memory networks – generally called "LSTMs" – are an extraordinary sort of RNN, equipped for adapting to long term conditions. They were presented by Hochreiter and Schmidhuber, and were refined and promoted by numerous individuals. They work colossally well on a huge assortment of issues, and are currently broadly utilized.

All intermittent neural networks have the type of a chain of rehashing modules of neural networks. In standard RNNs, this rehashing module will have an extremely basic structure, for example, a solitary tanh layer. Regularly used architecture of LSTM units have a cell and three regulators. Cell is the memory part of the LSTM unit. Regulators comprises of input,

output and forget gate. The dependencies between the subsequent input notes is taken care by the cell. The sigmoid layer (activation function of LSTM) yields numbers somewhere in the range of zero and one, depicting the amount of every part ought to be let through. An estimation of zero signifies "let nothing through," while an estimation of one signifies "let everything through".

2.3 Attention based LSTM

Attention is a later advancement that really takes care of our center issue. This enables to take care of specific segments of the contribution at some random moment and utilize those segments to help produce portions of the yield as opposed to simply the last output of the LSTM layer.

$$c_i = \sum_{j=1}^{n} \alpha_{ij} h_j \tag{1}$$

where α_{ij} are weights that define the consideration of hidden states in each output. c_i is the i^{th} element of the context vector for output y_i which is the weighted sum of attention weights of hidden states of input sequence. h_j is the encoder network's hidden state.

$$\alpha_{ij} = align(y_i, x_j) \tag{2}$$

Here α_{ij} tells how well subsequent notes y_i and x_j are aligned.

$$\alpha_{ij} = \frac{exp(score(s_{i-1}, h_j)))}{\sum_{j'=1}^{n} exp(scores_{i-1}, h'_j))} \tag{3}$$

The α_{ij} gives the softmax of predefined alignment score.

$$score(s_i, h_j) = {v_a}^T tanh(W_a[s_i, h_j]) \tag{4}$$

where v_a and W_a are weight matrices that are learned in the alignment model. s_i is the decoder network's hidden states.

2.4 Bidirectional LSTM

Bidirectional LSTMs are an expansion of conventional LSTMs that can improve model execution on arrangement order issues. In issues where all timesteps of the information arrangement are accessible, Bidirectional LSTMs train two rather than one LSTM on the information grouping.

2.5 MIDI

The Musical Instrument Digital Interface format (MIDI or .mid) is used to store message rules which contain note pitches, their volume, speed, start and end timestamp, phrases and so forth. It doesn't store songs like sound formats, however it stores data that is equipped for producing future melodic notes. These rules can be deciphered by a sound card which uses a wavetable (table of recorded sound waves) to make an understanding of the MIDI messages into genuine stable information. It very well may be deciphered by midi player studio, for example, Fruity Loops (FL) Studio or standard sequencers like Synthesia. Musical notation software like MuseScore or Finale can make a translation of midi into an editable sheet music; this empowers customers to make music in regular music documentation on their PCs and they may listen to it by MIDI players.

3 Proposed methodology

Figure 2 outlines the proposed architecture for music generation. In our work, we work with the MIDI file format. This representation is a protocol which enables the digital music tools and musical instruments to be able to interconvert the music. We utilized the Music21 toolkit to extract crucial information from these MIDI files. We convert the midi file into a stream object. It is then used to extract notes and chords present in the file. The intermediate representation then consists of pitches and chords represented as a list of integers separated by a dot. Each note is the notation representation which may further consist of pitch (frequency of sound), octave (interval between 1 musical pitch and its fundamental frequency) and an offset (denoting the note location). Chord is mainly constituted when few notes are played simultaneously. So, this index representation is the dictionary to map pitches to integers. Thus, it intrinsically embeds the sounds such as tones, chords appropriately. Inside that list generated, any new chord is assumed to be a new pitch. It is similar to the analogy that letters are combined to generate meaningful words which when placed in contextual order forms a meaningful sentence. Similarly the music vocabulary is also utilized to generate music which is uniquely defined by the pitches in the notes list. Since, the input feed to the BiLSTM with Attention network should be in real values, we provide an index representation (integer values) to each unique pitch in the notes list which are originally in string format. Attention mechanism gives more importance to certain parts of music which are required so that the generated output will be more synchronized with the input. If the previous notes are more melodious and weighted more by attention then we can generate an output that is melodious (so that it does not deviate from input). The normalized input stream of first 100 notes is fed to the network to predict the 101^{st} corresponding note. The output vector is then converted to one-hot vector representation. Each BiLSTM cell is fed the sampled index. The index with maximum probability is chosen as the output. This index is then mapped to the actual note (reverse mapping from dictionary) and added to the predicted output list, which is a list of strings of notes and chords. Finally, this encoding is converted back to MIDI format.

3.1 Preprocessing

To prepare the model, the MIDI records should have to be changed over into a structure that can be encoded as numeric information to feed the Bidirectional LSTM network. The typical length of each MIDI file in the dataset ranges from 1 to 4774. We have utilized a software package Music21[1] to preprocess the data. We included the rests and duration (Rhythm) as opposed to just having notes and chords. We utilized the Music21 library to take our MIDI records and convert them into a stream object which consists of rhythms, and notes, with a related instrument and time duration. It parses all the MIDI files and annexed each note/harmony/rest-duration combination to a vector representation which is subdivided into 100-dimensional note samples. Those subdivided samples are then fed into the Bidirectional LSTM for training in order to generate the consecutive notes.

The data consists of two parts: i) Notes and ii) Chords. Pitch, octave, and offset of the Note are also covered as note objects. Pitch refers to the recurrence of the sound, or how high or low it is and is indicated with the letters [A, B, C, D, E, F, G], with A being the most elevated and G being the least. In order to transpose any note to a level higher octave it is

[1] http://web.mit.edu/music21/doc/moduleReference/moduleConverter.html

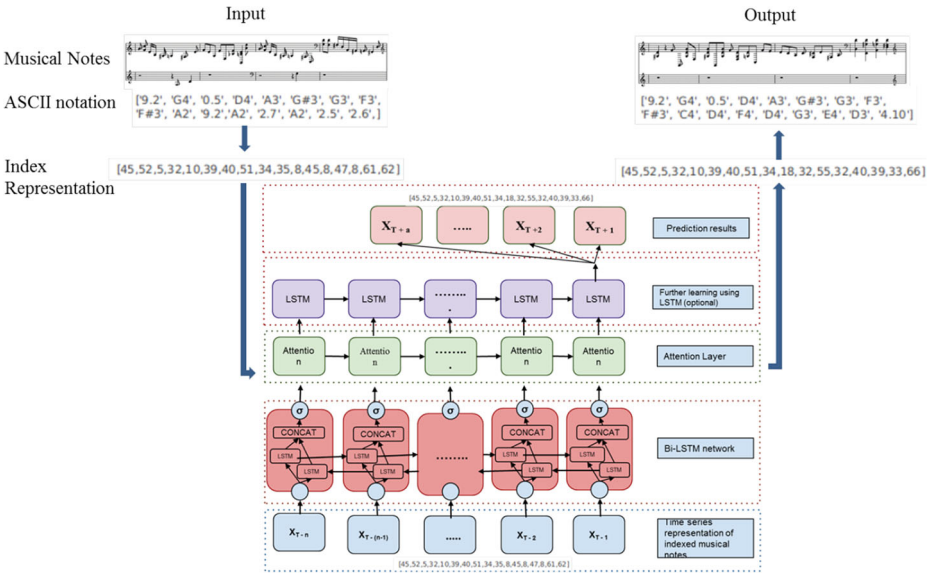

Fig. 2 Architecture diagram

required to add 12 to its pitch value. This transposition is quite easy in MIDI format as it is simply by doing arithmetic operations such as addition and subtraction to a fixed value. Octave refers to the set of pitches one may use on a piano. Offset refers to where the note is situated in the musical piece. Similarly, Chord objects are basically a placeholder for few notes that are played simultaneously. Intuitively, we may observe that to generate music precisely our neural network should have the ability to anticipate the upcoming note or harmony. Bi-LSTMs networks can mimic that properly. In our experimentation, the training set comprises of various notes and harmonies. The notes generally have shifting interims between them. We can have numerous notes with hardly a pause in between and afterward followed by a rest period where no note is played for a brief timeframe.

3.2 Music generation using attention based LSTM

Recurrent neural networks (RNNs) are quite widely used to process sequential data information. The seminal work by Bengio et al. [2] proposed the standard RNN model. It has limitations such as it suffers from gradient vanishing and explosion issue. In order to address this and incorporate long-term dependencies, Hochreiter and Schmidhuber [14] proposed Long-Short Term Memory networks (LSTMs). As LSTMs only account for the forward information flow, Bidirectional Long Short-Term Memory networks were thus proposed by Graves and Schmidhuber [12] a variant of LSTM networks, which is composed of a forward LSTM network and a reverse LSTM network, capturing the context knowledge of time series. To achieve our objective of generating old style music with rehashed melodic structure, we have utilized Bi-LSTM network with attention layer [22]. We have utilized

1x100 dimensional input notes sample size into a Bidirectional LSTM with 512 cells, followed by an attention layer, subsequent LSTM layer with 512 nodes. Finally we have a 3400 node Dense layer with softmax predictions, 3400 denotes the number of possible unique note/chord/rest-duration combinations in the input data. We have utilized dropout to reduce over-fitting issues. Further, we utilized categorical cross entropy loss and rmsprop as the optimizer function. The second LSTM enables further learning these interdependencies between the notes and harmonies.

4 Experimentation and results

4.1 Dataset discription

We used Jazz ML ready MIDI dataset[2] to train our model. The dataset comprises of 818 diverse Jazz music melodies. There are 804 distinct notes which are annexed in a unique way i.e. each distinct character which is represented in ASCII is mapped to unique numerical value. The dataset comprises of:

- The list of notes extracted from the midi file,
- Number of notes and
- List of unique notes for each midi file.

We have divided the dataset into training/validation and test splits in the ratio 80:20. All the parameter tuning, training was performed on the training/validation set while the final accuracy has been evaluated on the test set.

4.2 Experimentation results

The output generated file is compared with the original file to find out the deviations from the input sequence. As seen in Table 2, the cross entropy loss and mean square error (MSE) between the subsequent notes is compared for three variants: LSTM, LSTM with Attention, Bidirectional-LSTM with Attention and LSTM. The categorical cross entropy loss ablation study is done on the training set. LSTM performs the least than other two as it gives high error rates. In order to improve the learning capability, attention layer is added to the LSTM and it is found that it perfoms better than vanilla LSTM network. MSE metric is more correlated to the inherent harmony structures between the original melody and the music generated by the network. Finally, we observe that in all cases Bidirectional-LSTM with Attention and followed by stacked LSTM gives the best results. Figure 3 shows the categorical cross entropy loss for the three variants of LSTMs considered in this work. The learning curves are plotted in Fig. 4. Once the model is trained, we utilize the trained model to generate music for the test set. We objectively quantify the goodness of the musical notes generation and since the MSE is calculated with respect to the original notes it also validates that the realism of the generated musical notes is not lost. In Fig. 5, we show qualitative results demonstrating the generated music segments along with the original music sequences. It can be observed that the note structures in the generated music sheet by the proposed model and input sheet is similar in nature indicating that by providing the input sheet notes upto 100 notes, every 101^{th} note is predicted accurately.

[2] https://www.kaggle.com/saikayala/jazz-ml-ready-midi

Table 2 Performance analysis on music generation

Methods	Categorical cross entropy loss	RMSE	MSE
MidiNet [23]	0.1203	0.7112	0.5058
Adrien Ycart and E. Benetos [24]	0.2317	0.9904	0.9808
LSTM	0.5097	1.0919	1.1924
LSTM + Attention	0.2286	0.8924	0.7864
Bi-LSTM + Attention + LSTM	0.1069	0.6694	0.4481

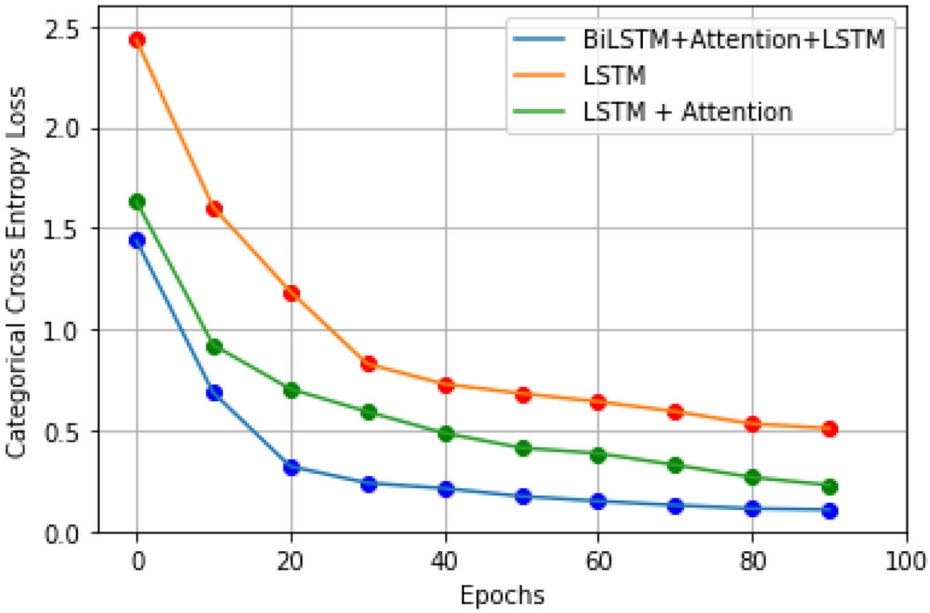

Fig. 3 Graph: Categorical Cross Entropy Loss

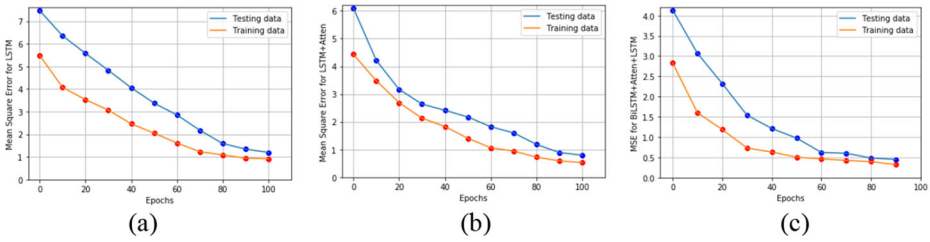

Fig. 4 Performance Evaluation Graph: (a)-(c) shows Mean Square Error for LSTM, LSTM+attention and Bi-LSTM+attention respectively

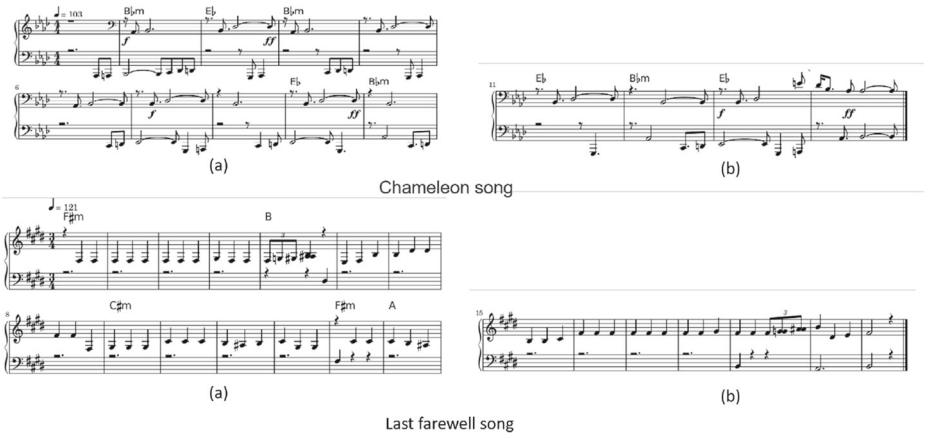

Fig. 5 a) Input music sheet b) Output music sheet generated by the proposed framework for songs Chameleon (Top Row) and Last farewell (Bottom row)

4.3 Comparison with related works

In [24], the author discusses about polyphonic midi sequences using LSTM networks. On comparing with the LSTM and LSTM+Attention it showed better results, but it showed a relatively high error rate when compared with Bi-directional LSTM with Attention and LSTM. Because music is all about learning the patterns and in order to recreate it we needed a model to understand these patterns and to be able generate subsequent notes to make a pleasant tune. Hence, attention plays a key role in our model. In [23], the error rate is relatively lower than [24] and when compared with Bidirectional LSTM with Attention and LSTM the error rates are almost similar. The differentiating factor between these two is how well the generated notes match with original one.

5 Conclusion

In this paper, we presented bidirectional LSTM with Attention and LSTM with the objective of producing music that is coherent and good to hear. Our proposed model improved the structure of the generated music by understanding the patterns in it and training them until a better accuracy and minimal error rates is achieved. Given the ongoing trends in AI in music industry, we envisage that the current work presents progressively complex models and information portrayals that successfully captures the fundamental melodic structure.

References

1. Abraham A (2005) Artificial neural networks. Handbook of measuring system design
2. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166
3. Boulanger-Lewandowski N, Bengio Y, Vincent P (2012) Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. arXiv:1206.6392
4. Browne CB (2001) System and method for automatic music generation using a neural network architecture. US Patent 6, 297, 439

5. Chen CC, Miikkulainen R (2001) Creating melodies with evolving recurrent neural networks. In: IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222), vol 3. IEEE, pp 2241–2246
6. Dieleman S, Schrauwen B (2014) End-to-end learning for music audio. In: 2014 IEEE International conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 6964–6968
7. Dong HW, Hsiao WY, Yang LC, Yang YH (2018) Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In: Thirty-second AAAI conference on artificial intelligence
8. Drewes F, Högberg J. (2007) An algebra for tree-based music generation. In: International conference on algebraic informatics. Springer, pp 172–188
9. Eck D, Schmidhuber J (2002) Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In: Proceedings of the 12th IEEE workshop on neural networks for signal processing. IEEE, pp 747–756
10. Eck D, Schmidhuber J (2002) A first look at music composition using lstm recurrent neural networks. Istit Dalle Molle Stud Sull Intell Artif 103:48
11. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y (2014) Generative adversarial nets. In: Advances in neural information processing systems, pp 2672–2680
12. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Netw 18(5-6):602–610
13. Hadjeres G, Nielsen F (2017) Interactive music generation with positional constraints using anticipation-rnns. arXiv:1709.06404
14. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
15. Johnson DD (2017) Generating polyphonic music using tied parallel networks. In: International conference on evolutionary and biologically inspired music and art. Springer, pp 128–143
16. Lee H, Pham P, Largman Y, Ng A (2009) Unsupervised feature learning for audio classification using convolutional deep belief networks. Adv Neural Inf Process Syst 22:1096–1104
17. Lewis J (1988) Creation by refinement: a creativity paradigm for gradient descent learning networks. In: International conf. on neural networks, pp 229–233
18. Liu I, Ramakrishnan B et al (2014) Bach in 2014: Music composition with recurrent neural network. arXiv:1412.3191
19. Oord A. v. d., Dieleman S, Zen H, Simonyan K, Vinyals O, Graves A, Kalchbrenner N, Senior A, Kavukcuoglu K (2016) Wavenet: A generative model for raw audio. arXiv:1609.03499
20. Schulze W, Van Der Merwe B (2010) Music generation with markov models. IEEE MultiMedia (3):78–85
21. Todd P (1988) A sequential network design for musical applications. In: Proceedings of the 1988 connectionist models summer school, pp 76–84
22. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: NIPS
23. Yang LC, Chou SY, Yang YH (2017) Midinet: A convolutional generative adversarial network for symbolic-domain music generation. arXiv:1703.10847
24. Ycart A, Benetos E et al (2017) A study on lstm networks for polyphonic music sequence modelling. ISMIR