International Journal of Scientific Research in Computer Science, Engineering and Information Technology
ISSN : 2456-3307 (www.ijsrcseit.com)
doi : https://doi.org/10.32628/CSEIT2173145

A Deep Learning Approach for Generating Pleasant Music

Dax Jain*, Diya Mistry, Dr. Nidhi Arora
Inferenz Pvt. Ltd., Ahmedabad, Gujarat, India

Article Info
Volume 7, Issue 3
Page Number: 641-649

Publication Issue :
May-June-2021

Article History
Accepted : 10 June 2021
Published : 16 June 2021

ABSTRACT

Advancement in deep neural networks have made it possible to compose music that mimics music composition by humans. The capacity of deep learning architectures in learning musical style from arbitrary musical corpora have been explored in this paper. The paper proposes a method for generated from the estimated distribution. Musical chords have been extracted for various instruments to train a sequential model to generate the polyphonic music on some selected instruments. We demonstrate a simple method comprising a sequential LSTM models to generate polyphonic music. The results of the model evaluation show that generated music is pleasant to hear and is similar to music played by humans. This has great application in entertainment industry which enables music composers to generate variety of creative music.

Keywords: Music Generation, Deep Learning, LSTM, RNN, Polyphonic Music, MIDI

I. INTRODUCTION

Sounds are all around us. We hear different types of sounds during the day, some are good to ears while some are only noise. From birds chirping to waves lapping against a coastline are tuneful and can be considered as a form of music of nature. Humans are capable of organizing sounds together in thoughtful ways to create a specific atmosphere. The lyrical sounds are also a strong means of expressions and emotions. These organized sounds put together forms man-made music.

Music comprises of sounds, vibrations, and silent moments, and it isn't always pleasant to listen. Blacking [3] perceived music as an “organized sound into a socially acceptable pattern.” Technically, music is a collection of coordinated sounds. It can be composed by putting sounds and tones in some particular order. Music composers get into the process of music creation by organizing various sounds creatively to obtain music of their choice. With the rise in machine learning tools, deep neural networks have seen lot of acceptance in variety of application domains. It has also seen applications in arts and creativity. From writing novels like Shakespeare to convert photos into paintings, neural networks are everywhere. Music is considered to be an area of art which connects feelings to various types of sounds. Depending on the type of music, there are number of emotions which can be attached to it. Music is capable of not only conveying emotions, but also gives environmental experiences. Music composition is also one of the most creative domains. In this paper, we propose a pleasantly new system based on deep neural networks to generate polyphonic music.

A dataset consisting of MIDI files is used to produce new music because MIDI files are found to be efficient in producing better sounding, less noisy music. Another reason for using midi files instead of raw audio data is that the raw audio data requires more computation power. We have used Windows 10 on Ubuntu 8.1 with Intel core i5 4th gen CPU and Intel core i7 9th gen which included Nvidia GTX 1650 4 GB GPU. The implementation was done in Python language using some popular libraries for music generation like TensorFlow, Keras, mido, pretty_midi etc. We have used deep learning networks which accept existing music data and generate new polyphonic music from it. The pre-processed dataset is passed through deep network to develop new good quality music. We have used LSTMs as it is one of the variations of recurrent neural networks and are widely used in sequence-based models.

We propose a model to generate the chord progression as a structural guide for the music. Our goal is to generate music that is pleasant to hear but not necessarily one that resembles how humans play music. Since there is only 1 chord for every 8-time steps of our polyphonic model, the chord structures last for a longer time frame, which is not possible to be processed with only one LSTM. The chord structure is given as input in a polyphonic LSTM which generates the actual music. In contrast to other work, our polyphonic LSTM is free to generate any note, not just chord notes. The LSTM is provided information about the chords. It is thus easier to model this as a learning problem where composed music is used as training data to extract useful musical patterns. An additional objective of the model is to generate music which sounds pleasant.

II. EXISTING WORK

Music is generated in a two-step process. First step is to generate a chord progression and in second step the chord progression is used to generate polyphonic music. Polyphony is a type of musical texture which consists of two or more simultaneous lines of independent melody instead of just one voice as in case of monophony. Lot of researchers have worked in this direction. In the early 1950s, Xenakis [16] used the concepts of statistics and probability to compose music. He perceived music as a sequence of sounds that occur by chance. So, he composed music using stochastic theory. He randomly selected sound elements depending on mathematical concepts and used Wave Net and LSTM models. Early efforts by Hochreiter & Schmidhuber [13] shows that LSTM learns much faster than RTRL, BPTT, Recurrent Cacade-Correlation, Elman nets and Neural Sequence Chunking and can solve complex problems which were not previously solved by recurrent networks. Liu & Ramakrishnan [17] proposed a framework based on resilient propagation and long short term memory for music composition. They showed that LSTM network is able to recreate music.

Some researchers also worked on approaches based on chord progressions. Choi et al. [14] created a text-based LSTM that learns relationships within text documents to represent chord progressions. Chu et al. [12] presented hierarchical recurrent neural network which generated monophonic melody first and then based on the melody chords, he added drums. Experiment by Huang & Wu [1] focused on learning embeddings for musical notes. The embeddings show that the model can learn to distinguish between low and high pitches. Chaun et al present an approach for music generation that incorporates domain knowledge in its representation [6]. They used a deep network structure consisting of multi-layered convolutional neural networks for learning the knowledge representation and recurrent neural networks with long short-term memory cells for capturing temporal dependencies in music sequences.

Some approaches in literature use raw audio data, whereas some other use MIDI files. Generally, systems that use MIDI files produce better sounding, less noisy music. Moreover, training on raw audio data requires more computing power, and is often infeasible with current approaches. Oord et al. created Wave Net, a text-to-speech model based on CNNs that is trained on raw audio data [2]. They show that their model can also be used to generate music. Mehri et al. train hierarchical RNNs on raw audio data [23]. Eck et al. use two different LSTM networks – one to learn chord structure and local note structure and one to learn longer term dependencies in order to try to learn a melody and retain it throughout the piece [9]. This allows the authors to generate music that never diverges far from the original chord progression melody. The only downside of this approach was that this architecture considers a predefined number of chords and thus it is not able to create a diverse combination of notes.

Back in 2011, when the field of memetic computing was emerging, lot of researchers started using memetic algorithms to address real world problems and music generation was one of those. Wells & ElAarag [25] suggested a memetic algorithm to produce quality musical compositions and they found that the generated music was very promising and conformed with MIDI protocol standards. Boulanger-Lewandowski et al. [5] try to deal with the challenge of learning complex polyphonic structure in music. They used a recurrent temporal restricted Boltzmann machine (RTRBM) to model unconstrained polyphonic music. RTRBM architecture allowed them to represent a complicated distribution over each time step instead of considering a single token like other models. By using this approach, they were able to tackle the problem of polyphony in generated music. Lyu et al. [15] also used Restricted Boltzmann Machine (RBM) to exploit high dimensional data modelling along with Long Short-Term Memory (LSTM) to memorize and retrieve useful information for music generation. They claim that their model is more generalized for various musical styles and can generate polyphonic music better than other models. Sigtia et al [22] combined probabilistic graphical model with acoustic and language model predictions. Their acoustic model is a neural network based model which they used for estimating the probabilities of pitches in a frame of audio. On the other hand, a recurrent neural network is chosen to be the language model to understand the correlations between pitch combinations over time.

According to Kim & Andre [19] as opposed to composing music traditionally, each single component of a composition is selected and combined in a user-specific manner. This does not need a human being to continuously provide inputs on the music. Work of Salas et al. [21] was to model the musical composition process of machines in expressing music. Their model was based on a linguistic approach. They considered music as a language which is composed of various sequences of symbols to form melodies and lexical symbols which produce sounds and silences. Their work was around determination of functions to describe the probability distribution of musical notes for the purpose of using them for automatic music generation.

Correa et al. [8] presented two approaches for computer aided music composition. One of the approaches which is based on supervised learning, the back propagation network is made to learn the musical structure from melodies of the training set. On the other hand, in unsupervised learning approach, they used self-organizing maps which learns to compose new melodies from input by extracting the landscapes contours. In similar work, Boenn et al. [4] described an automated system to compose melodic, harmonic and rhythmic music. Their system was also found to be capable of finding

errors in music composed by humans. In another work by Unehara & Onisawa [24] a method of music composition was introduced based on Genetic Algorithm. They took one chromosome to correspond to 4-bar musical information to compose 16-bar musical works reflecting users feeling. On the other hand, Chun & Miikkulainen [7] worked on an evolutionary algorithm to find a neural network which can maximize the chances of generating good melodies. They used the rules comprising of tonality and rhythm as a fitness function for the evolution.

Franklin [10] believed that the state of the musical system depends on history of past states of music. He presented that LSTM network can be used to represent music knowledge and can learn musical tasks and are also capable of learning to reproduce long songs. Huang & Wu [1] build a generative model from a deep neural network architecture to create music which shows both the behaviors; harmony and melody. To do that they performed end-to-end learning of deep neural nets alone. Chuan et al [6] transformed music into a 2D representation to graphically encode musical relationships between pitches. They incorporated this representation in Convolutional Neural Networks for learning the abstract encoding. They also used Recurrent Neural Networks with LSTM for capturing temporal dependencies in music sequences.

III. EXPERIMENTS

Looking at the widespread use and acceptance of LSTM in music generation, we have also presented two deep networks to generate music. Experiments performed to generate new music from the representative music dataset are discussed in this section.

A. Dataset Used

A clean MIDI dataset which is the subset of the Lakh MIDI Dataset [20] has been used in this work. The dataset consists of approximately one hundred thousand songs in the MIDI data format. MIDI files do not contain any sounds, rather it consists of a series of messages like “note on”, “note off”, and “change tempo”. The MIDI messages are interpreted by a software on MIDI instrument which then produces the sound. These MIDI messages can be sent to different channels with different sounding instruments. For example, channel 0 can represent a piano, whereas channel 1 may corresponds to a guitar. Because MIDI files only contain a score of the song and no actual sound, a song usually takes much less storage space than other audio files such as WAV or MP3. This is also beneficial when training neural networks. Since the dataset is smaller, one can incorporate more songs during training. Moreover, it is simple to change the instrument with which the music is played.

Furthermore, the MIDI format provides a basic representation of music, whereas a raw audio file is more difficult to interpret, for humans as well as machine learning algorithms. MIDI files are smaller in size as they do not contain any audio data. For example, a MIDI file can explain what notes are played, when they're played, and how long or loud each note should be. Files in this format are basically instructional files that explain how the sound should be produced once attached to a playback device or loaded into a particular software program that knows how to interpret the data. This is the reason why MIDI files are considered to be a preferred way for sharing musical information between similar applications and for transferring over low-bandwidth internet connections. The small size also allows for storing on low-capacity devices.

Data pre-processing is the first step to make sure that the data used in the study is clean and in usable format. All songs in the dataset have different tempo through which makes it difficult to be used in model training. So, the tempo of the songs is subjected to normalization. A histogram of chords extracted from the data is as shown in figure 1.

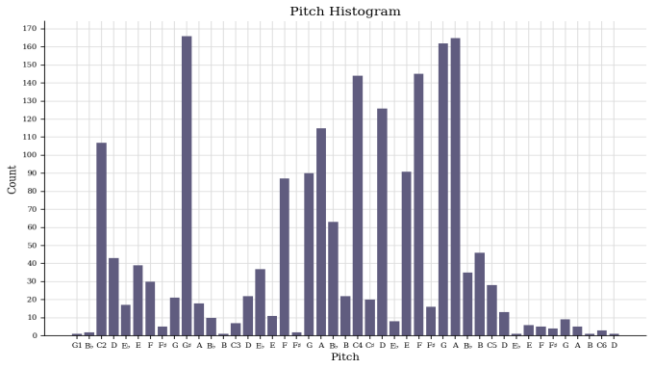

Figure 1. Histogram of Notes

B. Deep Networks for Generating Music

When you listen to a song, lyrics dependencies are very important to make it understandable. In the same way, when we read a book, we understand each paragraph based on our understanding of the context. This is also true when we read a small story. Each word is dependent on previous words. Classical Neural Networks (NN), fail to grab the behavior of context dependency. This is the reason why Recurrent Neural Networks (RNN) are preferred over NNs.

A key feature of generating music is to make it pleasing and this can be achieved by long-term dependencies in the music. Hence, an LSTM (Long Short-Term Memory) Network [18] is used which is an improvement over the architecture of RNN with simple memory cells and a gating architecture. The gates in LSTM decide whether LSTM cells should forget or persist the previous state in each loop and thus make LSTMs capable of learning useful dependencies within a long sequence. Two types of models namely Chord LSTM and Polyphonic LSTM have been used in the proposed work.

i. Chord LSTM

The chords of the songs are epitomized by the technique from the natural language processing. In Machine learning applications that deal with language, words are often substituted with integer ids and the word-id pairs are stored in a dictionary. The most happening words of a corpus receive a unique id, because the remaining words do not occur often enough for the algorithms to learn anything meaningful from them. The rarely occurring words receive the id of unknown tag. We use the same technique for the chord LSTM. The chords are substituted with ids and the chord pairs are stored in a dictionary. So, the chord LSTM only sees the ids of the chords and has no knowledge of the notes that make up the chords. Before feeding the chords into chord LSTM the chord id's have to be encoded in the form of vectors, which is done by one-hot encoding.

Before we feed the chord ids into the chord LSTM, we have to encode them as vectors. To do so we use one-hot encoding. The input vectors are the same size as the size of the chord vocabulary. All the vector entries are 0, except for the entry at the index of the chord id which equals 1.

ii. Polyphonic LSTM

To represent the music data that is fed into the polyphonic LSTM we use a piano roll representation. Piano roll is a music format which represents a music piece by a score-like matrix. The vertical and horizontal axes in this representation show note pitch and time, respectively. The values represent the velocities of the notes. The notes that are played at each time step are represented as a vector. The length of these vectors is the number of notes. If a note is

played at that time step, the corresponding vector entry is a 1 and if the note is not played the corresponding entry is a 0. The piano rolls of the songs are created with the pretty midi library [11] for Python.

C. Parameters used by Model

The goal of chord LSTM is to learn a meaningful representation of the chords from the training data. The architecture of chord LSTM is as shown in figure 2 where the embedding matrix consists of learnable parameters. The output from the first layer is fed into two LSTM layers with 256 hidden cells with a SoftMax activation function. The output of the LSTM is in the form of a vector that corresponds to the probabilities for the chords to be played next. To generate a new chord progression, we first feed a seed of variable length into the LSTM. The next chord is then generated by sampling the output probability vector. The generated chord is then fed into the LSTM again and the next chord is again sampled and so on.

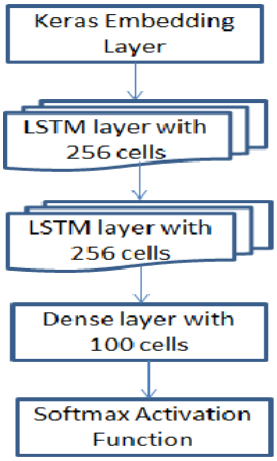

Figure 2. Architecture of Chord LSTM

The input vector to LSTM consists of the vectors from the piano rolls of the songs. The input vectors are fed into two LSTM layers with 256 nodes in the hidden layer. The sigmoid activation function has been used in the output layer. To generate a new song, a seed consisting of the piano roll and the corresponding chords are fed into the LSTM. The notes which are played at the next time step are then sampled from the output vector. The notes are sampled independently in such a way that if one note is chosen to be played, the probabilities of playing the other notes do not change. Figure 3 shows the architecture of polyphonic LSTM with 256 nodes in first two LSTM layers while 60 nodes with dense connections in the output layer.

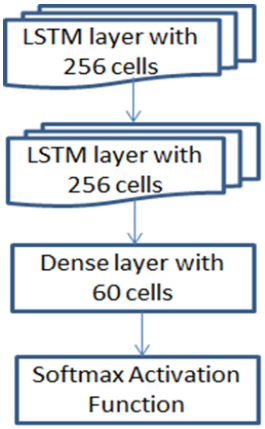

Figure 3. Architecture of Polyphonic LSTM

IV. RESULT OF EXPERIMENTS

A sample input is shown in figure 4 with time on x-axis and notes played in the music on y-axis. The colors in the figure represent the notes played by different instruments.

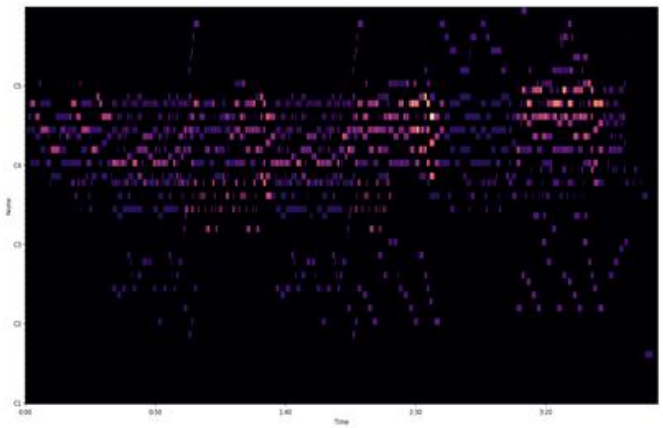

Figure 4. The Input Music

Figure 5 shows a closer look of the input music played only for 20 seconds.

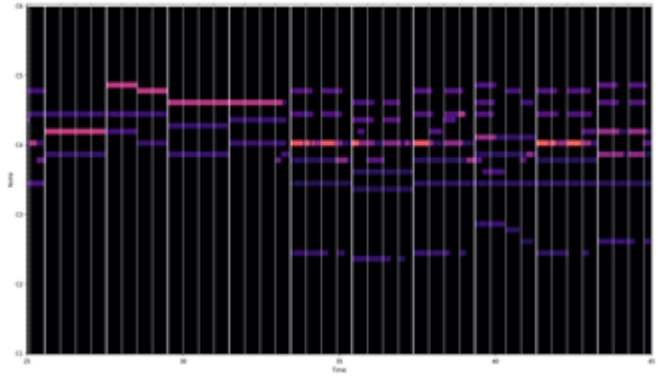

Figure 5. Part of Input Music

Figure 6 represents the piano roll of the input music in which x-axis represent the time position and y-axis represents the note. This piano roll is considered as the input for the model. The tempo of this piano roll is changed and made same for all the piano rolls. As it shows the notes in single colour, it represents a single instrument. On the other hand, figure 7 represents the piano roll obtained as a result of training our models on input in figure 6.

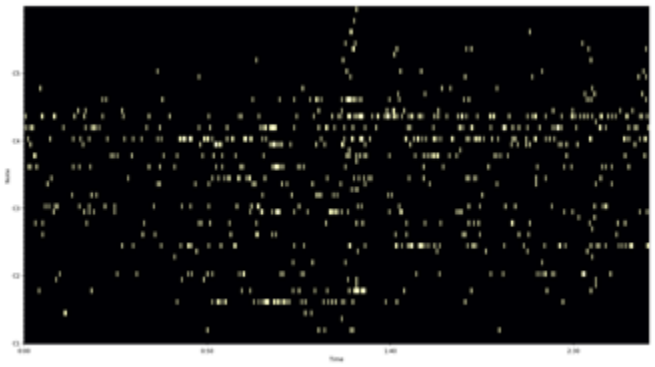

Figure 6. Input Piano roll

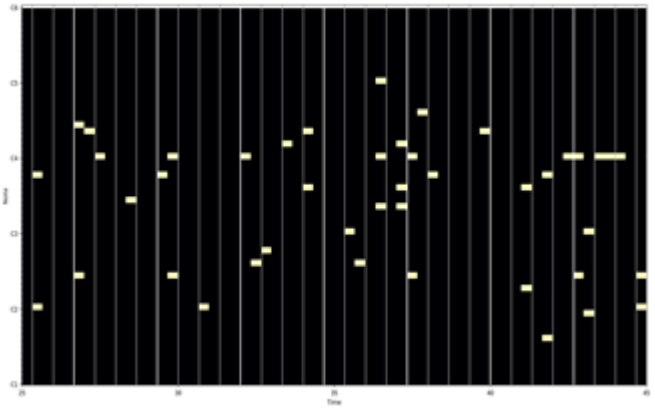

Figure 7. Output Piano roll

Similar outputs are obtained for other instruments also. The music produced is found to be pleasant to ears and is very close to the original music.

V. CONCLUSION

The paper presents the work done for training deep neural networks for generating new music from the data in the form of MIDI files. The chords and piano roll representations are extracted from the MIDI files to prepare training dataset. These extracted chords and piano rolls data is used to train the chord LSTM and polyphonic LSTM. The chord LSTM generates a chord progression which is used as input to the polyphonic LSTM to generate new music in MIDI format. The LSTM model learns the notes to be played on different types of chords.

The results obtained by these models show promising results by effectively generating music for selected

instruments. The music was found to be melodious and pleasant to listen. When listening to the music, one can freely vary tempo and instrumentation. As the network has a simple structure and is easy to implement and use, the proposed deep learning models can be used in music and entertainment industry. Another advantage of these models which can bring interest of people from entertainment is that the models take very less time to generate music if MIDI data is given as inputs instead of raw audio files.

VI. REFERENCES

[1]. Huang, A. & Wu, R. (2016). Deep Learning for Music. CoRR, abs/1606.04930.

[2]. Oord, A. van den, Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., & Kavukcuoglu, K. (2016). WaveNet: A Generative Model for Raw Audio. In The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016 (pp. 125). ISCA.

[3]. Blacking, J. (1969). The value of music in human experience. Year Book of the International Folk Music Council, pp. 33-71.

[4]. Boenn, G., Brain, M., De vos, M., & Ffitch, J. (2011). Automatic Music Composition Using Answer Set Programming. Theory Pract. Log. Program., 11(2–3), 397–427.

[5]. Boulanger-Lewandowski, N., Bengio, Y., & Vincent, P. (2012). Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. In Proceedings of the 29th International Coference on International Conference on Machine Learning (pp. 1881–1888). Omnipress.

[6]. Chuan, Ching-Hua, & Herremans, D., (2018). Modeling Temporal Tonal Relations in Polyphonic Music Through Deep Networks with a Novel Image-Based Representation. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018 (pp. 2159–2166). AAAI Press.

[7]. Chun-chi J. Chen, & Miikkulainen, R., (2001). Creating Melodies with Evolving Recurrent Neural Networks. In in: Proceedings of the INNS-IEEE International Joint Conference on Neural Networks, 2241–2246, IEEE, Piscataway, NJ (pp. 2241–2246).

[8]. Correa, D., Levada, A., Saito, J., & Mari, J. (2008). Neural Network Based Systems for Computer-Aided Musical Composition: Supervised x Unsupervised Learning. In Proceedings of the 2008 ACM Symposium on Applied Computing (pp. 1738–1742). Association for Computing Machinery.

[9]. Eck, D., Schmidhuber, J. (2002). A First Look at Music Composition using LSTM Recurrent Neural Networks. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale.

[10]. Franklin, J. (2006). Recurrent Neural Networks for Music Computation. INFORMS J. on Computing, 18(3), 321–338.

[11]. Brunner, G., Wang, Y., Wattenhofer, R. & Wiesendanger, J. (2017). JamBot: Music Theory Aware Chord Based Generation of Polyphonic Music with LSTMs. In 29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA, November 6-8, 2017 (pp. 519–526). IEEE Computer Society.

[12]. Chu, H., Urtasun, R. & Fidler, S. (2017). Song From PI: A Musically Plausible Network for Pop Music Generation. In 5th International

Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings. OpenReview.net.

[13]. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), pp. 1735–1780.

[14]. Choi, K., & Fazekas, G. & Sandler, M.B. (2016). Text-based LSTM networks for Automatic Music Composition. CoRR, abs/1604.05358.

[15]. Lyu, Qi, Wu, Z., Zhu, J. & Meng. H. (2015). Modelling High-Dimensional Sequences with LSTM-RTRBM: Application to Polyphonic Music Generation. Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, pp. 4138-4139.

[16]. Xenakis, I. (1971). Formalized Music Thought and Mathematics in Composition. Indiana University Press.

[17]. Liu, I-Ting & Ramakrishnan, B. (2014). Bach in 2014: Music Composition with Recurrent Neural Network. CoRR, abs/1412.3191.

[18]. Briot, J.P., Hadjeres, G. & Pachet, F.D. (2017). Deep Learning Techniques for Music Generation - A Survey. CoRR, abs/1709.01620.

[19]. Kim, S., & André, E. (2004). A Generate and Sense Approach to Automated Music Composition. In Proceedings of the 9th International Conference on Intelligent User Interfaces (pp. 268–270). Association for Computing Machinery.

[20]. Raffel, C. (2016). Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching. Columbia University.

[21]. Salas, H., Gelbukh, A., & Calvo, H. (2010). Music Composition Based on Linguistic Approach. In Proceedings of the 9th Mexican International Conference on Advances in Artificial Intelligence: Part I (pp. 117–128). Springer-Verlag.

[22]. Sigtia, S., Benetos, E., & Dixon, S. 2016. An end-to-end neural network for polyphonic piano music transcription. IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 24(5):927–939.

[23]. Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A.C. & Bengio, Y. (2016). SampleRNN: An Unconditional End-to-End Neural Audio Generation Model. CoRR, abs/1612.07837.

[24]. Unehara, M., & Onisawa, T. (2005). Music Composition by Interaction between Human and Computer. New Gen. Comput., 23(2), 181–191.

[25]. Wells, D., & ElAarag, H. (2011). A Novel Approach for Automated Music Composition Using Memetic Algorithms. In Proceedings of the 49th Annual Southeast Regional Conference (pp. 155–159). Association for Computing Machinery.

Cite this article as :

Dax Jain, Diya Mistry, Dr. Nidhi Arora, "A Deep Learning Approach for Generating Pleasant Music", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 7, Issue 3, pp.641-649, May-June-2021. Available at doi : https://doi.org/10.32628/CSEIT2173145
Journal URL : https://ijsrcseit.com/CSEIT2173145