Grenze International Journal of Engineering and Technology, January Issue

Classical Music Generation using LSTM-based Models

Dhyey Shah[1], Nimish Tiwari[2], Shyamal Virnodkar[3], Akshit Savaliya[4] and Jiya Pancholi[5]
[1-5]Computer Engineering Department, K. J. Somaiya Institute of Technology, Mumbai, India
Email: {dhyey.shah, nimish.tiwari, shyamal, akshit.savaliya, jiya.pancholi}@somaiya.edu

***Abstract*—AI generated music has witnessed a lot of innovative methods in the past. Most of the recent generation techniques revolve around machine learning, especially deep learning. This paper explores music generation using Long Short-Term Memory (LSTMs), which are well-suited for handling sequential data. Since music typically consists of a sequential set of notes and chords, LSTMs can be used to extract important features such as notes, duration, and offset present in the input music. Considering these features is essential for generating a smooth and coherent output. A note accuracy of 86.7% and training loss of 0.56 was achieved with the method described in this paper. Thus, the ability of LSTMs to capture distinct features in the music, and to generate new music based on these features is investigated.**

***Index Terms*— AI generated music, autoencoders, deep learning and neural network.**

I. Introduction

The field of deep learning (DL) has experienced rapid growth in recent years and has now become a standard tool for a wide range of applications including tasks like image classification, speech recognition, and language translation. One interesting application of DL is for generating musical melodies based on the symbolic representation of some input music. Existing methods, employing techniques such as Variational Autoencoders [1] (VAE), Generative Adversarial Networks [2] (GANs), Transformers [3], and Recurrent Neural Networks [4] (RNNs) have shown great potential for this task. Yet, as per our knowledge, the automatic mu-sic composition field still lags in terms of producing a nuanced, coherent output that flows naturally, and sounds as expressive and rich as human performances.

A significant challenge with music generation lies in translating music theory knowledge to a rigorous mathematical representation, because of the subjective and subtle nature of music. This deficiency often leads to undesirable behavior such as inappropriate pauses and musical irregularities in the compositions. Thus, a question arises: Can incorporating music theory knowledge enhance the capabilities of these models? Addressing this question necessitates not only an understanding of the problem, but also the exploration of effective solutions.

This paper explores music generation using LSTMs, in order to achieve a melodic and coherent output. The model is trained with an emphasis on compositions of the classical era, in order to better understand and analyze the note and chord variations. The performance of this model is analyzed with the help of metrics such as accuracy, training loss, etc. in this paper.

II. Literature Review

Various DL techniques for automated generation of music have been explored in the past decade.

Grenze ID: 01.GIJET.11.1.505_1

However, there has been a significant gap in applying these models to sequential data, particularly in the domain of music. MusicVAE, a model proposed by Roberts et al. [1], addresses this gap by introducing a hierarchical decoder for modelling sequences with long-term structures. generation in lower layers and drums/chords in higher levels. Dong et al. [2] introduce MuseGAN, that uses GANs to generate multi-track symbolic music.
In contrast to these methods, recent work by Wu and Yang [3] addresses the shortcomings of AI-composed music using Transformers, with a focus on the jazz genre. Chu et al. [4] propose a novel hierarchical RNN designed specifically for pop music generation, delineating melody. Another technique proposed by Kalingeri et al. [5], generates music by working with raw audio data in the frequency domain. A hybrid method presented by Wang et al. [6] builds an intelligent music generation system using a combination of GAN and VAE within a DL network to create music in adherence to music theory rules.
Huang et al. [7] introduced a modified Transformer model using a relative attention mechanism for music generation, overcoming memory complexities in representing relative positional information. Muhamed et al. [8] perform symbolic music generation through Transformer-GANs, enhancing long composition quality by incorporating adversarial losses and optimizing discrete sequences via the Gumbel-Softmax trick to outperform baseline models and the state-of-the-art Music Transformer in human evaluations.
The method proposed by [9] introduces Piano Inpainting Application (PIA), which is based on a Linear Transformer architecture trained on Structured MIDI Encoding to efficiently restore missing segments in piano performances. Another recent paper [10] focuses on Sichuan unvoiced music, devising a melody generation algorithm that amalgamates RNN-LSTM from DL and genetic algorithms (GA). It is evident that a considerable amount of work has been done in this direction, and recently too. Progress in this field has been particularly boosted with the advent of new DL techniques and architectures.

III. METHODOLOGY

In this section, the methodology employed for music generation using an LSTM based neural network is described. This approach is built upon the foundation of DL techniques. The method used is based on two main steps: (i) extracting relevant features from the input files; (ii) training an LSTM to learn and retain the features necessary for generation. The key steps involved in the proposed methodology are as follows:

A. Data Collection

The method mentioned takes input in MIDI (Musical Instrument Digital Interface) format, as they are universally used for storing musical annotation, and are well-suited for this task [11], [12]. MIDI files store musical data that describes sound instructions, including notes, pitch, duration, and other musical attributes, enabling digital instruments and software to interpret and play music.
The target genre for the dataset is music belonging to the classical era. In particular, a few compositions from prominent composers of the classical era such as Beethoven, Haydn, Liszt, Mozart, and Schubert were included. In total, there were a total of 116 compositions in the form of MIDI files in the dataset. While the number of selected data samples is not too high, note that each composition is complex and rich, where extracting the relevant features can be enough for the model to adapt to the musical style and rhythm.

B. Pre-processing and Extraction

In this phase, the files were converted to .MID format [13], to be processed by the Music21 library. Using this library, the notes and chords of each file were analysed, and relevant features such as pitch, duration and offset were extracted [14] from each file and saved into a single data object.
The reason for selecting notes, duration and offset as the primary features for training is because of their intrinsic significance in music theory [15], [16]. These features indicate crucial aspects of musical notes, their temporal characteristics, and positional placement. After the completion of this step, this data can be given as a direct input to the LSTM model.

C. Model Architecture

In this section, the underlying architecture of the model is explained in detail. The model of choice for this task is an LSTM, in order to leverage its capability to handle long sequential data. It is also well-suited for capturing patterns in sequential data, making it an excellent candidate for handling musical sequences [17], [18], which often follow a temporal structure with melodies. The LSTM in use comprises of three distinct branches corresponding to the three input features (notes, offset and duration), each of which utilizes an LSTM layer for sequence processing.

An LSTM with 256 units was used for individual branches, followed by a concatenated layer, two LSTM layers with 512 units, batch normalization, and dense layers for predicting notes, offsets, and durations, as shown in Fig. 1.

The activation function used within the dense layers after the LSTM units is the softmax activation function. This function is commonly used in multi-class classification tasks to transform the model's output into a probability distribution over multiple classes.

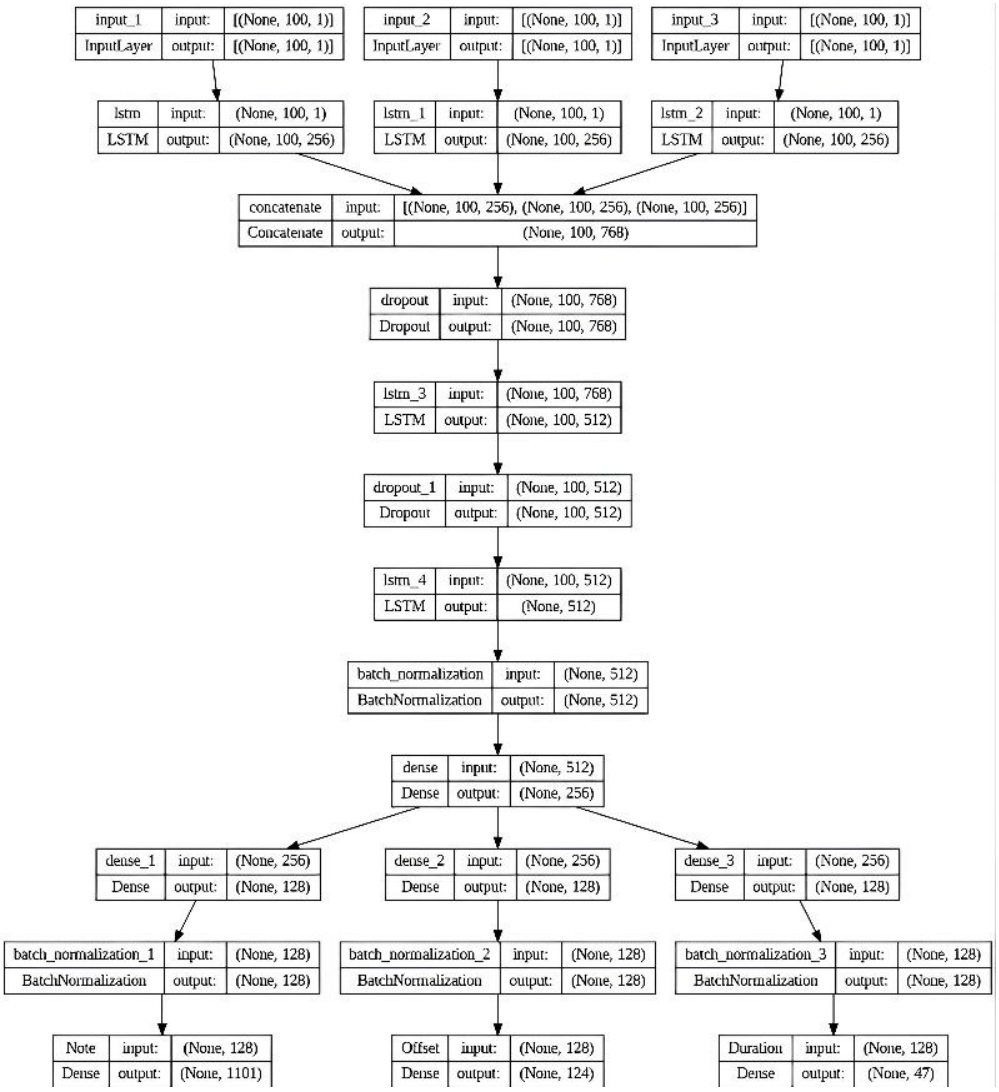

Figure 1. Architecture of the LSTM model

D. Model Training

The neural network architecture consists of three branches as mentioned earlier, each of which focuses on note, offset and duration classification. The model was compiled using the Kullback-Leibler divergence loss function, which is suitable for LSTM models [19], along with the Adam optimizer and accuracy metrics for each output branch.

The model was trained over 50 epochs and a batch size of 512 instances per iteration for rapid convergence. The total number of trainable parameters was found to be around 5.9 million. Dividing the training process into multiple epochs helped the model update the weights with each pass of the dataset.

However, it should be noted that too few or too many epochs may lead to underfitting or overfitting of the model. Thus, it is essential to choose a balanced number of epochs. For the model training, checkpoints were implemented to save the best-performing model weights at each epoch, allowing continuous learning or retraining.

IV. Result Analysis

To better understand the performance of the model during training, the progression of the loss and accuracy values is examined in this section. Table 1 shows the change in values for note loss, offset loss, duration loss, note accuracy, offset accuracy, and duration accuracy, which are abbreviated as NL, OL, DL, NA, OA, and DA respectively.

An interval of 10 epochs has been chosen to see how the loss and accuracy vary with number of epochs. It can be observed from the table that the loss values keep decreasing, and that of accuracy keep increasing with each epoch. After training for 50 epochs, the net loss comes out to be 0.56, and the final notes, offset and duration accuracies are 86.7%, 98.3% and 97.5% respectively.

Table I. Variation In Values Of Loss And Accuracy

Epoch	Net Loss	NL	OL	DL	NA	OA	DA
1	8.5327	5.5272	1.6059	1.3997	0.024	0.623	0.652
10	3.5062	2.855	0.2404	0.4108	0.29	0.924	0.868
20	1.6451	1.3818	0.0963	0.167	0.612	0.967	0.943
30	1.0034	0.8275	0.0682	0.1076	0.752	0.977	0.963
40	0.7149	0.5745	0.0556	0.0848	0.822	0.98	0.97
50	**0.5561**	0.4353	0.0482	0.0711	**0.867**	**0.983**	**0.975**

It is also evident from Fig. 2 that the training loss for each feature kept decreasing steadily. The graph for the net loss for the model is shown below in Fig. 3. The net loss can be expressed as the sum of the three individual feature losses. As expected, the loss decreases in a near logarithmic manner.

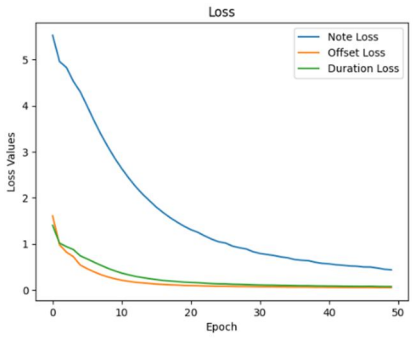

Figure 2. Graph for feature loss

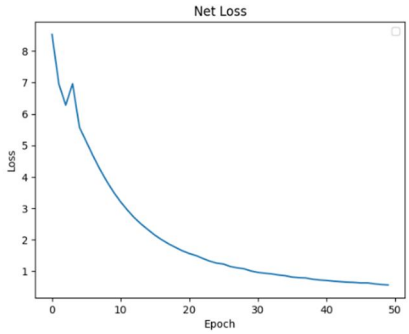

Figure 3. Graph for net loss

Similarly, Fig. 4 contains graphs for the accuracy of each feature as shown below. It is again evident that the accuracy of each feature increases steadily as the number of epochs keep increasing. As seen earlier, the initial loss value for notes was highest. Since it is inversely proportional to the accuracy, the starting accuracy for notes is considerably lower (near zero) than the other features, which is aligned with the expected behaviour.

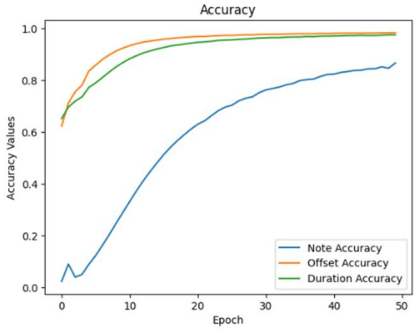

Figure 4. Graph for feature accuracy

The variation in values for training loss and accuracy is indicative of the model's adaptability and learning process throughout the training epochs. Through experimentation, it was found that the generated music closely resembles the style, structure, and musical nuances of the input compositions from composers such as Beethoven, Haydn, Liszt, Mozart, and Schubert. The model also proved to be capable of producing coherent melodies, pleasing to the ear.

Despite the seemingly satisfactory results obtained above, there are a few limitations with this approach:

- This method only works for single-track music compositions. It currently does not support multi-track compositions i.e., music containing the simultaneous arrangement of multiple instruments or voices.
- The model works best for compositions that make use of a single instrument. The quality of music may degrade in case of polyphonic music, as the notes and chords generally tend to be more complex. Note that even single-track music can be polyphonic if it contains multiple notes at the same time instance.
- There can be some kind of composer bias and repetition in the model, which can affect the authenticity and diversity of the generated music. To counter this, a larger, and more representative dataset can be selected which would help in minimizing any bias towards a particular composer or genre.

V. Conclusion And Future Scope

In this study, experimentation with LSTM models trained on music from classical composers yielded interesting results in terms of melody and rhythm. The method used in this paper can be further enhanced by collecting a larger, and more diverse dataset representative of various musical styles and artists.

It also remains of great interest to witness further advancements in this area, especially for longer pieces of music, and those that are polyphonic in nature. Incorporating vocal elements in the generated music is another significant challenge to be tackled. At the same time, there is a rising need for mitigating composer bias, which can help balance representation and diversity in the generated music.

References

[1] Roberts, A., Engel, J., Raffel, C., Hawthorne, C. & Eck, D.. (2018). A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music. Proceedings of the 35th International Conference on Machine Learning, in Proceedings of Machine Learning Research 80:4364-4373.

[2] Dong, H.-W., W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang. "MuseGAN: Multi-Track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment". Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, Apr. 2018, doi:10.1609/aaai.v32i1.11312.

[3] Wu, S. L., & Yang, Y. H. (2020). The Jazz Transformer on the front line: Exploring the shortcomings of AI-composed music through quantitative measures. arXiv preprint arXiv:2008.01307.

[4] Chu, Hang, Raquel Urtasun, and Sanja Fidler. "Song from PI: A musically plausible network for pop music generation." arXiv preprint arXiv:1611.03477 (2016).

[5] Kalingeri, V., & Grandhe, S. (2016). Music generation with deep learning. arXiv preprint arXiv:1612.04928.

[6] T. Wang, J. Liu, C. Jin, J. Li and S. Ma, "An intelligent music generation based on Variational Autoencoder," 2020 International Conference on Culture-oriented Science & Technology (ICCST), Beijing, China, 2020, pp. 394-398, doi: 10.1109/ICCST50977.2020.00082.

[7] Huang, Cheng-Zhi Anna, et al. "Music transformer." arXiv preprint arXiv:1809.04281 (2018).

[8] Muhamed, A., L. Li, X. Shi, S. Yaddanapudi, W. Chi, D. Jackson, R. Suresh, Z. C. Lipton, and A. J. Smola. "Symbolic Music Generation With Transformer-GANs". Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 1, May 2021, pp. 408-17, doi:10.1609/aaai.v35i1.16117.

[9] Hadjeres, Gaëtan, and Léopold Crestel. "The piano inpainting application." arXiv preprint arXiv:2107.05944 (2021).

[10] Dong, Ling. "Using deep learning and genetic algorithms for melody generation and optimization in music." Soft Computing (2023): 1-15.

[11] Alaeddine, M., Tannoury, A. (2021). Artificial Intelligence in Music Composition. In: Maglogiannis, I., Macintyre, J., Iliadis, L. (eds) Artificial Intelligence Applications and Innovations. AIAI 2021. IFIP Advances in Information and Communication Technology, vol 627. Springer, Cham. https://doi.org/10.1007/978-3-030-79150-6_31

[12] Emma Frid, Celso Gomes, and Zeyu Jin. 2020. Music Creation by Example. In Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems (CHI '20). Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376514

[13] Sieger, Nicholas J., and Ahmed H. Tewfik. "Audio coding for representation in MIDI via pitch detection using harmonic dictionaries." Journal of VLSI signal processing systems for signal, image and video technology 20 (1998): 45-59.

[14] G. Ozcan, C. Isikhan and A. Alpkocak, "Melody extraction on MIDI music files," Seventh IEEE International Symposium on Multimedia (ISM'05), Irvine, CA, USA, 2005, pp. 8 pp.-, doi: 10.1109/ISM.2005.77.

[15] Monahan, Caroline B., and Edward C. Carterette. "Pitch and duration as determinants of musical space." Music Perception 3.1 (1985): 1-32.

[16] Krumhansl, Carol L. "Rhythm and pitch in music cognition." Psychological bulletin 126.1 (2000): 159.
[17] M. K. Jędrzejewska, A. Zjawiński and B. Stasiak, "Generating Musical Expression of MIDI Music with LSTM Neural Network," 2018 11th International Conference on Human System Interaction (HSI), Gdansk, Poland, 2018, pp. 132-138, doi: 10.1109/HSI.2018.8431033.
[18] Mangal, Sanidhya, Rahul Modak, and Poorva Joshi. "LSTM based music generation system." arXiv preprint arXiv:1908.01080 (2019).
[19] Prokhorov, Victor, et al. "On the importance of the Kullback-Leibler divergence term in variational autoencoders for text generation." arXiv preprint arXiv:1909.13668 (2019).