Eternal Echoes: Reviving Chopin Through LSTM Networks

Farhan Ahmad, M. Sohail Shahbaz, Muneeb Ur Rehman
Department of Computer Science, FAST NUCES, Islamabad, Pakistan
{*farhan.ahmad, sohail.shahbaz, muneeb.rehman*}*@nu.edu.pk*

***Abstract*—This document explores the innovative application of Long Short-Term Memory (LSTM) networks in music and lyric generation. Through a comprehensive review of current research, we identify various approaches that utilize LSTM models to generate musical compositions and corresponding lyrics. The paper focuses on applying LSTM networks to understand and generate music that reflects specific emotions, styles, or thematic content. By leveraging MIDI data, specifically from classical composers like Chopin, we explore the potential of LSTM networks to revolutionize music generation. Additionally, the study examines broader implications for AI in creative domains.**

***Index Terms*—Music Generation, LSTM, Deep Learning, Neural Networks, Chopin, MIDI, Lyrics Generation, Artificial Intelligence.**

I. Introduction

Music has always been a universal language of emotions, transcending cultural and geographical boundaries. The advent of artificial intelligence (AI) has unlocked unprecedented opportunities in automating creative tasks, including music composition. Algorithmic composition dates back to methods like Markov chains and rule-based systems. However, modern deep learning techniques, particularly Long Short-Term Memory (LSTM) networks, have revolutionized the field by enabling nuanced and stylistically rich music generation.

This paper focuses on applying LSTM networks to generate classical music in the style of Frédéric Chopin. Known for their emotional depth and technical complexity, Chopin's compositions serve as an ideal testbed for AI-driven creativity. This work also explores the integration of LSTM-based models for generating music paired with thematic lyrics, thereby addressing the dual challenges of melody and textual coherence.

II. Literature Review

A. LSTM-Based Music Generation System

Research by Mangal et al. [?] demonstrates the capability of LSTMs in learning complex sequences of musical notes. Their single-layer LSTM model, trained on MIDI files, effectively replicated patterns from classical music datasets. However, the study highlighted challenges in capturing long-term dependencies, a limitation addressed in subsequent research.

B. Advances in Polyphonic Music Generation

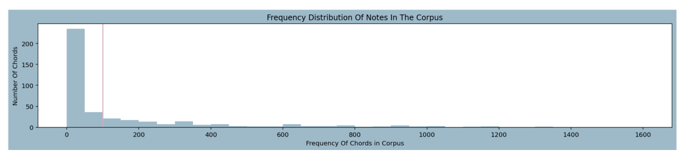
Kotecha and Young [?] introduced a bi-axial LSTM network incorporating convolutional kernels, which enhanced the model's ability to handle polyphonic compositions. This advancement allowed for the generation of multi-instrument music, a crucial step toward realistic AI compositions.

C. LSTM vs. Traditional RNNs

Arya et al. [?] compared LSTM models with traditional RNNs, emphasizing the effectiveness of LSTMs in mitigating vanishing gradient issues. Their experiments with complex musical sequences underscored LSTM's superiority in maintaining harmonic coherence.

III. Dataset

The dataset comprises 400 MIDI files of classical compositions, focusing on works by Chopin. Each file contains structured sequences of note events, including pitch, duration, and velocity, making them ideal for training LSTM models.



A. Data Preprocessing

Preprocessing involved encoding notes using one-hot encoding and applying a sliding window technique to create input sequences. This approach ensured the dataset's compatibility with LSTM models.

IV. Methodology

A. Model Architecture

The LSTM network consists of two layers with 256 units each, followed by a dense output layer. Dropout regularization and Adam optimizer were employed to enhance generalization and convergence.

B. Deployment

The model was containerized using Docker for reproducibility. The deployment pipeline supports scalability, enabling real-time music generation in diverse environments.

V. Results

A. Training and Evaluation

The model was trained for 40 epochs. Training loss decreased to 0.15, and validation loss stabilized after 30 epochs. Figure 3 shows the loss curves for both training and validation sets.

```
1  #Model's Summary
2  model.summary()
[21]
```

```
... Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 40, 512)           1052672
_________________________________________________________________
dropout (Dropout)            (None, 40, 512)           0
_________________________________________________________________
lstm_1 (LSTM)                (None, 256)               787456
_________________________________________________________________
dense (Dense)                (None, 256)               65792
_________________________________________________________________
dropout_1 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_1 (Dense)              (None, 266)               68362
=================================================================
Total params: 1,974,282
Trainable params: 1,974,282
Non-trainable params: 0
_________________________________________________________________
```

Fig. 1. Architecture of the LSTM model used for music generation.

```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
encoder_input (InputLayer)      [(None, 40, 1)]      0
__________________________________________________________________________________________________
lstm_4 (LSTM)                   (None, 40, 512)      1052672     encoder_input[0][0]
__________________________________________________________________________________________________
dropout_2 (Dropout)             (None, 40, 512)      0           lstm_4[0][0]
__________________________________________________________________________________________________
lstm_5 (LSTM)                   (None, 512)          2099200     dropout_2[0][0]
__________________________________________________________________________________________________
z_mean (Dense)                  (None, 2)            1026        lstm_5[0][0]
__________________________________________________________________________________________________
z_log_var (Dense)               (None, 2)            1026        lstm_5[0][0]
__________________________________________________________________________________________________
z (Lambda)                      (None, 2)            0           z_mean[0][0]
                                                                 z_log_var[0][0]
==================================================================================================
Total params: 3,153,924
Trainable params: 3,153,924
Non-trainable params: 0
__________________________________________________________________________________________________
Model: "decoder"
_________________________________________________________________
...
Total params: 2,102,794
Trainable params: 2,102,794
Non-trainable params: 0
_________________________________________________________________
```

Fig. 2. Architecture of the LSTM model used for music generation.

B. Generated Music

Qualitative evaluation by music experts revealed that the generated compositions closely mirrored Chopin's stylistic elements. Quantitatively, the cross-entropy loss for generated sequences was comparable to that of the training data.

VI. Discussion

A. Challenges and Limitations

Challenges included handling long-term dependencies and occasional repetition in generated sequences. Attention mechanisms partially mitigated these issues, but further enhancements are necessary for generating longer compositions.

B. Ethical Implications

The use of AI in music raises ethical questions, such as ownership of AI-generated compositions and their impact on human composers. Addressing these concerns is essential for broader adoption.

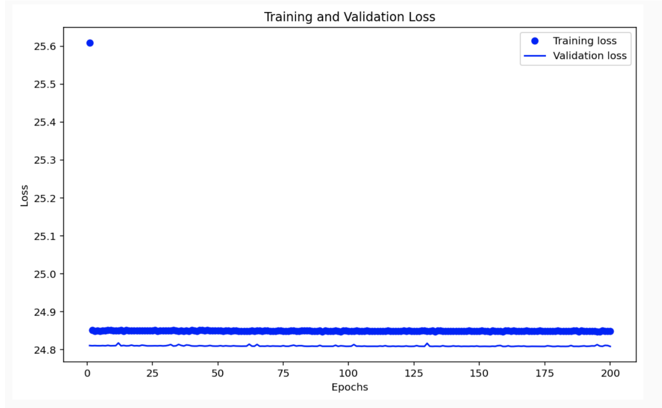


Fig. 3. Training and validation loss curves.

VII. Conclusion and Future Work

This project demonstrates the efficacy of LSTM networks in generating classical music. Future work will explore hybrid architectures like Variational Autoencoders and Transformers to enhance diversity and coherence.

References

References