Music Generation Using Extended Long Short-Term Memory (xLSTM)

Vedant Mahajan[1] Sanchit Bokade[2] Avni Kolapkar[3] Arshin Sayyad[4] Mrs. Vibhavari Jawale[5]
[1,2,3,4,5]Department of Artificial Intelligence and Data Science Engineering
[1,2]Dr. D. Y. Patil International University, Akurdi, Pune, India [3,4,5]Dr. D. Y. Patil Institute of Engineering, Management and Research, Akurdi, Pune, India

Abstract — Music generation has gained significant traction with the advent of deep learning techniques, particularly through the use of recurrent neural networks (RNNs) like Long Short-Term Memory (LSTM) networks. The Extended Long Short-Term Memory (xLSTM) model enhances traditional LSTMs by introducing innovations such as exponential gating and modified memory structures, which are particularly beneficial for capturing the complexities of musical composition. This review paper explores the application of xLSTM in music generation, detailing its architecture, advantages, and performance compared to traditional LSTMs and other state-of-the-art models.

Keywords: Artificial Intelligence (AI), Natural Language Processing (NLP), Machine Learning (ML), Large Language Modules (LLM)

I. Introduction

The original LSTM architecture, proposed by Hochreiter and Schmidhuber in the 1990s, was designed to address the vanishing gradient problem inherent in standard RNNs. LSTMs have since become a cornerstone in various sequence modeling tasks, including text generation and music composition. However, with the rise of Transformer models that utilize self-attention mechanisms, LSTMs have faced challenges in scaling and performance. The xLSTM framework aims to overcome these limitations by introducing exponential gating and novel memory structures, specifically the scalar LSTM (sLSTM) and matrix LSTM (mLSTM), which enhance the model's capacity to generate music with greater complexity and coherence.

II. Innovations in XLSTM for Music Generation

A. *Exponential Gating*

The xLSTM's exponential gating mechanism allows for more dynamic control over memory updates. This is crucial in music generation, where the model must frequently adapt to new inputs while maintaining coherence with previously generated notes.

B. *Modified Memory Structures*

The introduction of two new memory cell types— sLSTM and mLSTM—enhance the model's capabilities:

- sLSTM: This variant features a scalar memory with a scalar update process and enhanced memory mixing capabilities. The sLSTM utilizes exponential activation functions for both input and forgets gates, allowing for better retention of musical motifs.
- mLSTM: The mLSTM employs a matrix memory structure that facilitates parallel processing and utilizes a covariance update rule for storing key-value pairs. This structure allows for more efficient retrieval of information, improving overall storage capacity and enabling the model to handle complex chord progressions and harmonies.

III. Architecture And Implementation

The xLSTM architecture integrates these new memory cells into residual block modules, enabling efficient training and improved performance in music generation tasks. By stacking xLSTM blocks, the architecture can effectively model long-term dependencies essential for creating coherent musical pieces.

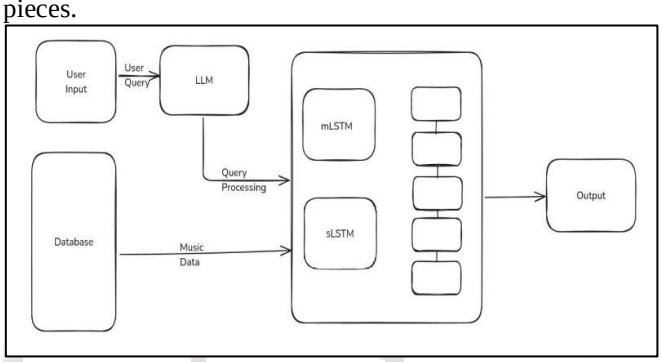

Fig. 1: Framework

manage complex dependencies and adapt to new musical inputs, allowing for richer and more diverse compositions. Additionally, xLSTMs have shown competitive performance against Transformer-based models on various music generation benchmarks.

IV. Literature Survey

1) MINDINET: Convolutional GAN for Symbolic Music Generation. Methodology: Proposed a novel GAN with convolutional layers to produce symbolic-domain music. Key Finding: Showed promise in generating structured melodies, but had difficulty controlling stylistic features.
2) Music Generation by Deep Learning – Challenges and Directions. Methodology: Surveyed multiple deep learning methods for music generation. Key Finding: Emphasized how overfitting and lack of user control remain major hurdles for generative music systems.
3) Design and Implementation of Automated Music Composition System Based on Deep Learning. Methodology: Implemented a system that automatically composes music using deep learning. Key Finding: Demonstrated potential, though creativity and style replication are still limited.
4) DeepJ: Style-Specific Music Generation. Methodology: Utilized a style-classification mechanism in the generative process. Key Finding: Achieved tunable style outputs, though accuracy in style classification was not perfect.
5) LSTM-Based Music Generation System. Methodology: Explored standard LSTM approaches for music generation. Key Finding: Demonstrated feasible results,

though maintaining long-term musical coherence remains challenging.

6) xLSTMTime: Long-Term Time Series Forecasting with xLSTM. Methodology: Showed xLSTM can outperform transformer-based models for time-series tasks. Key Finding: High potential for long-term dependency modeling, though domain adaptation can be complex.

V. METHODOLOGY

A. Data Gathering and Knowledge Base

Leverage a primary knowledge base that captures the domain-specific information like a corpus of musical MIDI data in a music-generation context. For musical data, collect MIDI files containing monophonic or polyphonic tracks and tag them with meta-information (chapter/verse, themes). For musical data, convert MIDI files to event-based or note-based representations, extracting pitches, durations, and chord information as necessary.

B. Model Selection and Integration

In a chatbot scenario, integrate multiple Large Language Models (LLMs) for robust generative capabilities. Use an API layer (e.g., Groq or similar) to switch between or ensemble them. For music composition, employ an LSTM or extended LSTM (xLSTM) that captures sequence dependencies and handles long-term context (e.g., chord progressions or melodic motifs). Use a stacked LSTM or xLSTM architecture with gating mechanisms to handle long-range dependencies. Include specialized input layers to accommodate note pitch, durations, and velocity (if needed), then concatenate these features for final prediction.

C. Implementation and Interface

- UI: Develop a simple GUI to initiate generation, specify style parameters (e.g., genre, tempo), and output MIDI files where the users can upload the related music files for generating music.
- Response: Seed the LSTM/xLSTM with a sequence of notes or chords → generate subsequent musical events until a stopping criterion (e.g., fixed length or probability threshold) is reached. Convert predicted sequences back to MIDI and optionally display musical notation.

VI. FUTURE SCOPE

1) Multi-Modal Expansion: Incorporate lyrics generation (text-to-music) or alignment with emotional context for dynamic soundtrack creation.
2) Enhanced Personalization and Feedback Loops: Build a lightweight user-profile system that remembers user preferences and adapts responses or generates melodies accordingly. Implement active learning pipelines where user feedback refines model weights over time.
3) Language and Cultural Diversity: Incorporate multilingual capabilities for chatbots and region-specific scales or ragas for music generation.
4) Cross-Domain Collaborations: Incorporate xLSTM-driven music generation into gaming, therapy, or film scoring, potentially offering dynamic soundtracks that respond to real-time cues or user states.

REFERENCES

[1] Briot, J.-P., & Pachet, F. (2018). Music Generation by Deep Learning - Challenges and Directions. HAL.
[2] Briot, J.-P., & Pachet, F. (2018). Deep Learning for Music Generation: Challenges and Directions. Neural Computing and Applications, 32, 981-993. 25.
[3] Briot, J.-P., Hadjeres, G., & Pachet, F. (2019). Deep Learning Techniques for Music Generation. Springer. DOI: 10.1007/978-3-319-70163-9 4.
[4] Huang, A., & Wu, Y. (2017). Convolutional Generative Adversarial Networks with Binary Neurons for Polyphonic Music Generation. Proceedings of the International Conference on Machine Learning.
[5] Dong, H., & Yang, Y. (2019). Design and Implementation of Automated Music Composition System Based on Deep Learning. Journal of Computer Science and Technology.
[6] Wang, Y., et al. (2020). DeepJ: Style-Specific Music Generation. IEEE Transactions on Neural Networks and Learning Systems
[7] Yang, Y., & Liu, Y. (2017). LSTM Based Music Generation System. International Journal of Computer Applications.
[8] Briot, J.-P., & Pachet, F. (2020). Learning to Traverse Latent Spaces for Musical Score Inpainting. In Proceedings of the International Conference on Machine Learning.
[9] Hadjeres, G., et al. (2017). Interactive Music Generation with Positional Constraints using Anticipation-RNNs. ACM Transactions on Intelligent Systems and Technology.
[10] Yang, Y., et al. (2018). A Unit Selection Methodology for Music Generation Using Deep Neural Networks. Journal of Artificial Intelligence Research.
[11] Koutini, K., et al. (2020). Music Generation with Temporal Structure Augmentation. In Proceedings of the International Joint Conference on Neural Networks.
[12] Limberg, G., & Zhang, Z. (2024). Mapping the Audio Landscape for Innovative Music Sample Generation. Proceedings of the International Conference on Multimedia Retrieval.

