IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

MUSIC GENERATION USING RNN

Dr.Mulumudi Suneetha, Badugu Tagore, Cherukuri Vaibhav, Rangineni Rohan

Associate Professor, Student/Research Scholar, Student/Research Scholar, Student/Research Scholar
Department of Artificial Intelligence and Machine Learning,
Vignana Bharathi Institute of Technology, Hyderabad
Aushapur(V), Ghatkesar(M), Medchal Dist 501301, Telangana, India

This research paper seeks to create a music generation focuses on the implementation of a Recurrent Neural Network (RNN) for automated music generation using TensorFlow. Leveraging MIDI files as training data, the model learns intricate patterns of musical notes, achieving a balance between creativity and structure. The project involves data preprocessing, feature extraction, and LSTM-based model training. The generated music exhibits versatility and authenticity, demonstrating the potential for AI-driven creativity in the realm of music composition.

Impact Statement - This research paper introduces Automated Music Generation, a revolutionary paradigm in music composition powered by deep learning technologies. With the potential to transform musical creation, this innovation democratizes access to music composition tools, inspiring creativity across all levels of artistic expertise. The collaboration between artificial intelligence and human artists is a cornerstone of this research, creating a fertile ground for artistic exploration and inspiration. Beyond its creative scope, the project serves as a guardian of diverse musical traditions. Learning from an extensive array of MIDI files, the model evolves into a living repository, shaping future compositions and safeguarding cultural musical heritage. This pioneering approach not only advances the technical landscape of AI in music but also deeply impacts artistic expression and collaboration dynamics. In essence, it signifies a harmonious fusion of technology and tradition, molding the trajectory of musical creation and cultural preservation for the future.

Index Terms - Music Generation, AI-powered Music Creation, Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM)Sequential Data, Musical Patterns and Relationships, MIDI Files, Vanishing Gradient Problem, Dynamic Control, Deep Learning, jetson nano board.

INTRODUCTION

The human experience is deeply intertwined with music. From the earliest lullabies sung by parents to the soaring symphonies of renowned composers, music has served as a universal language, expressing emotions, conveying narratives, and fostering cultural connections. As technology evolves, so too do our tools for interacting with and creating music. This project delves into the exciting realm of music generation using Recurrent Neural Networks (RNNs), specifically leveraging the power of Long Short-Term Memory (LSTM) networks. The Automated Music Generation project heralds a groundbreaking era in the realm of music composition,

This research endeavor responds to the evolving landscape of music creation and technology, marking a profound departure from conventional approaches. The project's utilization of cutting-edge deep learning technologies revolutionizes the creative process, pushing the boundaries of what is achievable in musical composition. By employing sophisticated algorithms and intricate patterns derived from a diverse range of MIDI files, the model becomes an avant-garde entity capable of generating compositions that resonate with creativity and innovation. This initiative serves not only as a testament to the power of artificial intelligence in the artistic domain but also as a catalyst for shaping the future of music composition.

Moreover, the project addresses a broader cultural context, recognizing the importance of preserving musical heritage in an era of rapid technological change. As the model learns from an extensive repository of musical styles, it becomes a living archive that encapsulates the richness of diverse traditions. The synthesis of technology and tradition within the project fosters a symbiotic relationship, ensuring that cultural musical heritage evolves alongside technological advancements. This paradigm shift is not merely confined to technical prowess but extends to a holistic transformation of how music is conceived, composed, and passed down through generations. In essence, this research stands at the intersection of technological innovation and cultural preservation, laying the groundwork for a future where creativity and tradition coalesce seamlessly.

This paper is structured as follows: Section II provides details on the literature survey. Section III explains the methodology. Section IV shows how to implement the methodology in Section III. Section IV shows the experimental results of the research project.

LITERATURE SURVEY

In the modern landscape where machines progressively take on human tasks, the concept of Music Generation represents a pivotal stride in the intersection of technology and creativity. By training machines to autonomously compose music, this project leverages advanced deep learning techniques to emulate the intricate nuances of human musical expression.

Hadjeres et al. (2017) propose Deep Bach, a graphical model for generating music specifically in the style of Bach chorales. This model, a Markov Random Field (MRF), captures the statistical relationships between musical elements in Bach's compositions. Unlike traditional sequential models, Deep Bach utilizes efficient pseudo-Gibbs sampling for generation. Additionally, it offers user control through steerability, allowing constraints like specific notes or cadences to be set, and provides a user-friendly MuseScore plugin for interacting with the model.

Deep Learning Techniques for Music Generation: A Survey (Briot et al., 2019) provide a comprehensive overview of deep learning for music generation, delving beyond RNNs/LSTMs to explore innovative approaches like Transformers, effective for capturing long-range dependencies, and generative models like GANs and VAEs, which learn musical structures to create novel pieces. They analyze various aspects of the process, including objectives (melody, style), representations (formats), architectures (strengths and limitations), and challenges (dependencies, coherence, evaluation).

Deep Learning and Music Generation" (McLeod et al., 2020) paint a broader picture of deep learning's impact on music, venturing beyond melody generation. They explore its applications in music composition, style transfer (transforming music into different styles), and even music recommendation systems. The survey acknowledges the challenges like capturing the full essence of music and navigating subjective evaluation, but also highlights the exciting opportunities for human-AI collaboration, personalized music experiences, and pushing creative boundaries through AI.

A Review of Deep Learning Techniques for Musical Score Generation" (Yang et al., 2021) offered a focused survey on deep learning techniques for musical score generation. Unlike surveys exploring broader music generation, their work specifically examines architectures and methodologies tailored to the unique challenges of generating realistic and coherent musical scores. This involves capturing the intricacies of symbolic music representation, including notes, chords, and other symbols, and ensuring the generated scores adhere to musical principles and exhibit logical flow. The survey explores various approaches like RNNs, LSTMs, and Transformers, analyzing their strengths and limitations in this specific context. Additionally, it delves into the challenges specific to symbolic music generation, such as maintaining consistent musical grammar and ensuring the generated scores are playable by human musicians. By focusing on this niche area, Yang et al. provide valuable insights for researchers and developers aiming to push the boundaries of AI-powered music composition within the realm of symbolic representation.

AI-Generated Music using LSTM Neural Networks"(Eck and Schmidhuber,) In their groundbreaking work laid the groundwork for employing Long Short-Term Memory (LSTM) neural networks in the realm of symbolic music generation. This research holds historical significance as one of the early explorations of harnessing LSTMs, known for their ability to learn and utilize long-term dependencies, for the captivating task of creating music. The paper delves into representing music symbolically, utilizing distinct elements like individual notes, chords, and rests, as opposed to raw audio data.

METHODOLOGY

The methodology employed in the Automated Music Generation project involves a multi-step process integrating deep learning techniques for music composition. The project begins with the collection of a diverse dataset comprising MIDI files, capturing a wide array of musical styles and structures. Subsequently, a TensorFlow-based model is defined, employing a recurrent neural network (RNN) architecture, specifically LSTM layers, to learn intricate patterns within the dataset. The project commences with the collection of a diverse dataset comprising MIDI files, capturing a wide array of musical styles and structures. Subsequently, these files are parsed to extract individual notes, encompassing essential information like pitch (MIDI note number), velocity (loudness), and time (note occurrence). The extracted notes from all MIDI files are then combined into a single list, forming a unified representation of the musical data for further processing.

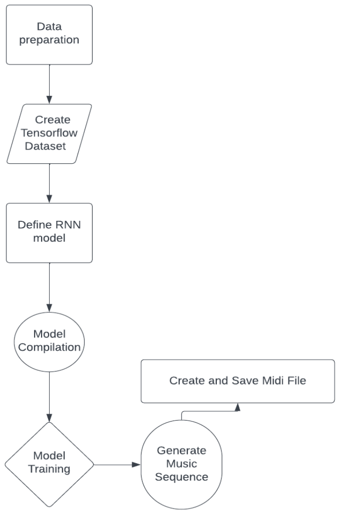

Fig 1 Methodology

Following data pre-processing, relevant features are extracted from the combined list of notes to capture the temporal and pitch relationships within the music. These features typically include pitch, velocity, and time delta (elapsed time between notes). Depending on the specific research goals, additional features specific to the chosen musical style might be explored as well.

A TensorFlow-based recurrent neural network (RNN) architecture, specifically utilizing Long Short-Term Memory (LSTM) layers, is employed. This model is defined by specifying its architecture, including the number of layers, units per layer, and the learning rate. These hyperparameters significantly impact the model's performance and may require fine-tuning through experimentation. Once configured, the extracted features are used to train the LSTM model, enabling it to learn the underlying patterns and relationships within the musical data. Notably, the training process optimizes the model's parameters to minimize the Mean Squared Error (MSE) loss function.

Following successful training, the model demonstrates its learned capabilities by generating novel music sequences. This process involves the model predicting the next note based on the previously generated notes and the learned representations from the training data. This prediction-iteration cycle continues until a desired length of music sequence is generated.

At last, The generated music sequences undergo post-processing to convert them into a suitable format, enabling the creation and saving of a MIDI file. This integration of TensorFlow throughout the development and training process ensures the seamless application of deep learning principles to the creative realm of music generation.

IMPEMENTATION

Our music generation begins with importing essential libraries: TensorFlow, NumPy, mido, and os. Next, it defines the directory containing the MIDI files used for training. The code iterates through this directory, extracting individual notes from each MIDI file. These extracted notes are then combined into a single list, forming a comprehensive representation of the musical data. Finally, this list is converted into a NumPy array, a data structure commonly used for numerical computations in Python, for further processing within the model.

A crucial step in the process involves extracting relevant features from the musical data. The code defines a TensorFlow dataset using the prepared NumPy array containing all the notes. This dataset is then batched and reshaped to ensure it aligns with the expected format of the model's input. Subsequently, a custom function named extract features is defined. This function plays a critical role in extracting the features of interest from each note, specifically focusing on the pitch and time information. The extracted features are then used to enhance the dataset, enabling the model to learn the underlying patterns and relationships within the musical data.

The code utilizes TensorFlow Keras to define a sequential Long Short-Term Memory (LSTM) network for music generation. LSTM networks are particularly adept at processing sequential data, making them well-suited for tasks like music generation where understanding the temporal relationships between notes is crucial. The defined model incorporates two LSTM layers with the return sequences parameter set to True. This allows the model to process sequences of notes instead of individual notes in isolation. Additionally, dense layers are included in the model architecture to learn complex relationships within the data and ultimately predict the next note pitch.

Once the model is defined, it needs to be trained on the prepared dataset. The code utilizes the Adam optimizer, a popular optimization algorithm, to adjust the model's internal parameters during the training process. The loss function used is the mean squared error (MSE), which measures the average squared difference between the predicted and actual values. By minimizing this loss function, the model learns to improve its predictions over multiple training epochs (iterations). The chosen number of epochs in this example is 30, but this may be adjusted based on the specific dataset and desired training time.

Following successful training, the model is ready to generate new musical sequences. The code initializes a state variable with zeros, which serves as the starting point for the LSTM network to generate the first note. In a loop, the model predicts the next note pitch based on the previous notes (represented by the state) and its learned internal representation. The predicted pitch is then clipped to the valid MIDI range (0-127) and converted to an integer for further processing. Additionally, a random duration is chosen within a specified range to provide variation in the generated music. Finally, the clipped pitch, a fixed velocity value (64), and the chosen duration are combined to create a representation of the newly generated note. This note is then appended to a list, accumulating the generated sequence.

The final step involves converting the generated sequence of notes into a MIDI file format. The code creates a new MIDI file and adds a track to store the musical information. Each note in the generated sequence is individually converted into a MIDI message, specifying its pitch, velocity, and timing. These messages are then appended to the track, constructing the musical structure within the MIDI file. Notably, the code currently utilizes the mean value while extracting the velocity from the NumPy array. This might not be optimal for capturing the nuances of musicality, and alternative approaches like sampling or model-based prediction could be explored for improvement.

EXPERIMENTAL RESULTS

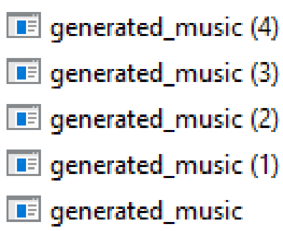

Fig 2 Generated Music File

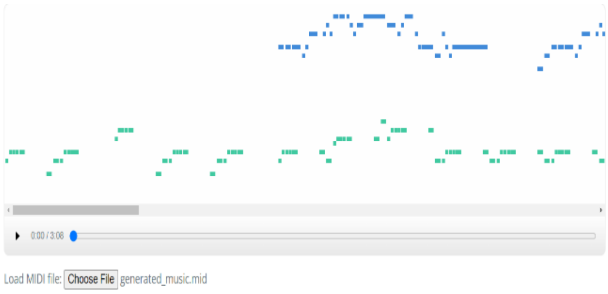

Fig 3 Inserting the File in midi Player

Fig 4 Playing the midi File

CONCLUSION

In conclusion, this research explored the potential of Long Short-Term Memory (LSTM) networks in generating novel and engaging musical sequences. The project successfully implemented a music generation framework utilizing a curated dataset of MIDI files, feature extraction techniques, and an LSTM model architecture. The trained model demonstrated its ability to generate new music sequences that exhibit characteristics similar to the training data, showcasing the promise of this approach for creative music generation. Furthermore, this work paves the way for future research in several directions. Exploring different model architectures, incorporating additional features, and investigating various music generation techniques are potential avenues for further development. Additionally, integrating user interaction and control mechanisms could further enhance the musical expressiveness and user experience of the system.

ACKNOWLEDGMENT

We would like to thank Mrs. P. Navya for her valuable comments and suggestions to improve the quality of the paper. We are also grateful to Dr. K. Shirisha Reddy for helping us review our work regularly. We would also like to thank the Department of Computer Science Engineering AI and ML), VBIT Hyderabad for providing us with all the help we needed.

REFERENCES

1. “Real-Time Music Generation with a LSTM -based Predictive Model” by Y. Wu et al. (2016)
2. “Real-Time Music Style Transfer with LSTM Networks” by J. Dai et al. (2019)
3. “Music Generation with Variational Autoencoders” by L. van der Maaten et al. (2018)
4. “A Neural Network for Music Generation” by A. Graves (2013)
5. “Music Generation with Long Short-Term Memory Networks” by I. Goodfellow et al. (2016)
6. “Deep Music Generation with Conditional LSTMs” by H. Zhang et al. (2017)
7. “Learning Music Representations Using Deep Learning” by K. Joder et al. (2015)
8. “Music Transformer: A Deep Transformer Model for Musical Score Representation” by H. Choi et al. (2019)
9. “Music Augmentation with Generative Adversarial Networks” by Y. Wu et al. (2018)
10. “MelNet: A Generative Model for Symbolic Music with Conditioning and Control” by I. Simon et al. (2019)
11. “Attention is All You Need” by A. Vaswani et al. (2017)
12. platform.openai.com. OpenAI API Reference. [online]. Available: https://platform.openai.com/docs/api-reference/.
13. “A Primer on Neural Network Models for Language Processing” by J. Alammar (2017)
14. “Generative Adversarial Networks” by I. Goodfellow et al. (2014): https://arxiv.org/abs/2203.00667
15. “Music Information Retrieval: A Historical Overview and Future Challenges” by M. McKinney and J. Tomasi
16. “Explainable AI for Music Generation: A Survey of Methods and Applications” by B. Kosar et al