WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Automated Music Beats Generation Using RNN

Shruti Srivastava [1], Suryansh Kapil [2], Ujjawal Tyagi [3] , Tanishka [4], Mr. Rahul Sharma [5]

[1,2,3,4] Student, Department of CSE, Raj Kumar Goel Institute of Technology Ghaziabad, U.P., India
[5] Assistant Professor Department of CSE, Raj Kumar Goel Institute of Technology Ghaziabad, U.P., India
ujjawal2002tyagi@gmail.com, suryanshkapil405@gmail.com, shrutishreya222@gmail.com, rastogitanishka216@gmail.com, sharma.rahul9944@gmail.com
DOI: https://doi.org/10.55248/gengpi.5.0524.1373

A B S T R A C T

In this paper we worked on generating beats using RNN based neural network like Long Short-Term Memory. The data used represents notes and chords of music in the form of coded files that drive musical and other digital instruments on what to play. It is build using framework like Electronjs, Axios to make it cross platform desktop app along with Python and it's libraries like M21, and framework like Flask, TensorFlow using GPU based Jupyter Notebook. The objective of this paper is to train and improve the neural network so that we can generate automated music beats composition of different types through the network.

Keywords - Music Beats Generation, Recurrent Neural Network, Interface Integration, LSTM

1. INTRODUCTION

In the digital age, machine learning has revolutionized various fields, including music processing, where it plays a pivotal role in the creation and generation of music. As technology continues to advance, musicians and composers seek innovative tools to enhance their creative processes. One such tool is the use of Long Short-Term Memory (LSTM) based recurrent neural networks (RNNs) for music generation.

This research focuses on developing an LSTM-based RNN to identify relationships within MIDI files, capturing musical chords and notes to create diverse compositions. By leveraging the Python Keras framework and the Music21 library, built on the TensorFlow framework, we aim to provide a robust and versatile solution compatible with various electronic devices, enabling musicians to seamlessly generate music.

The primary objective of this study is to train a neural network to facilitate automatic music composition. By examining patterns in MIDI files and using LSTM architecture, the network can retain and process musical information over extended periods, mimicking the creative process of human composers.

This research aims to demonstrate the potential of integrating advanced machine learning techniques into music generation, offering a new avenue for creativity and innovation in the music industry. By utilizing LSTM networks, we seek to enhance the ability of musicians to produce high-quality compositions, ultimately contributing to the evolving landscape of music creation and production.

2. LITERATURE REVIEW

Bob Sturm developed an LSTM-based character model [10] for generating textual representations of songs. His model consists of three distinct hidden layers, each containing 512 units. This architecture allows the network to capture complex patterns in song lyrics, contributing to the generation of coherent and contextually relevant musical text. Despite its effectiveness, one notable limitation of this model is its inability to accurately reconstruct notes from the sequences it generates, indicating a need for further refinement in music-specific tasks.

In 2016, researchers introduced the WaveNet, a deep generative model that processes raw audio files to generate music and voice. This network, developed by DeepMind, surpasses traditional text-to-speech systems by producing more natural-sounding results. WaveNet's architecture employs a series of convolutional layers that enable the model to learn intricate details in audio waveforms, making it a significant advancement in the field of audio generation. However, its high computational demands and complexity pose challenges for real-time applications. [11]

Doug Eck explored music composition using LSTMs. [11] His approach involves the network selecting chords present in a given sequence, with each note predicted based on the probability of its occurrence. While this method shows promise in generating musical compositions, it struggles with accurately reconstructing notes, leading to occasional incoherence in the generated music. This highlights the need for more sophisticated models that can handle the nuances of musical structures.

Dong et al. (2018) developed MuseGAN, a generative adversarial network (GAN) designed for multi-track music generation. MuseGAN uses a combination of convolutional and recurrent layers to generate music with multiple instrument tracks, ensuring harmony and coherence among the different parts. The model allows for both unconditioned music generation and conditioned generation based on user input, providing flexibility in creative applications. Despite its success, MuseGAN requires extensive computational resources and a large dataset for training, which can be a barrier to its adoption.

3. TECHNOLOGY USED

To stay competitive in today's digital world, leveraging advanced technology is essential. In the realm of music processing, machine learning techniques have become increasingly important for innovation. One significant development in this field is the application of Long Short-Term Memory (LSTM) networks for music generation. This article explores the technological architecture behind developing an LSTM-based recurrent neural network for music composition. The project utilizes the Python Keras framework and the Music21 library, built on TensorFlow, to design and implement a system capable of generating music by capturing and analyzing musical patterns in MIDI files.

1. ELECTRON JS : A framework that was developed and later open sourced by GitHub, and is used to make cross platform desktop apps effortlessly with common web technologies. It is based on chromium. We are using it to build the UI of our music player application. HTML, CSS, SASS Years old web structuring and designing technologies, that can be used to replace XAML if we were to use .NET in this scenario. It provides additional functionalities and adds syntactic sugar on top of CSS for smoother development.
2. M21 : Using Music21 to read MIDI files allows us to examine the time intervals between notes. These intervals are typically small, frequently 0.5. We may safely ignore such little offsets for our experiment because they do not affect the music's primary melodic structure.
3. RNN : It was developed and later open sourced by GitHub, and is used to make cross platform desktop apps effortlessly with common web technologies. It is based on chromium. We are using it to build the UI of our music player application. HTML, CSS, SASS Years old web structuring and designing technologies, that can be used to replace XAML if we were to use .NET in this scenario. It provides additional functionalities and adds syntactic sugar on top of CSS for smoother development. [12]
4. Keras : Working with TensorFlow is made easier with Keras, a high-level API built on top of TensorFlow. Using the Keras framework, we have designed and trained our LSTM model-based network. We were able to improve the overall efficiency of our implementation and expedite the coding process by leveraging the high-level TensorFlow APIs that Keras supplied.
5. AbcJs : A JavaScript module called abcjs is used to display sheet music in the ABC notation format on websites. It enables interactive music applications and webpages by letting developers to dynamically display music notation.
6. LameJs : It is a JavaScript package that enables web browsers to perform MP3 encoding. It makes it easier for programmers to create browser-based multimedia apps and audio processing tools by enabling them to encode audio data into the MP3 format directly within a web application.
7. Yt-search : This Node.js module lets developers do programmatic searches for YouTube videos. It offers an interface through which programs can access YouTube's search capabilities to get URLs and video metadata according to predefined search parameters.
8. Yt-dl-core : Yt-dl-core is a Node.js module that offers a streamlined interface for downloading videos from YouTube and other online video platforms. It makes it simple for developers to incorporate video downloading features into their apps by using the youtube-dl library to manage the downloading process.
9. TensorFlow : Google created the open-source machine learning framework TensorFlow[8]. It offers a whole ecosystem of libraries and tools for creating and implementing machine learning models at a large scale. Numerous tasks, such as neural networks, deep learning, and reinforcement learning, are supported by TensorFlow, and are widely used in both research and production environments.
10. Python : Python is a well-liked programming language that is well-known for being straightforward and adaptable. It is extensively utilized in several fields, including as scientific computing, data science, web development, and artificial intelligence. Python is a popular choice for machine learning and artificial intelligence (AI) applications because of its many libraries and frameworks, including TensorFlow and Keras.
11. Docker : Docker is a platform that uses containerization technology to build, ship, and operate programs. It enables programmers to bundle dependencies and applications into portable, light containers that function reliably in a variety of settings. Docker enhances resource efficiency, scalability, and deployment process simplicity.

4. PROPOSED METHODOLOGY

A. Input Format: To implement the neural network, we must first understand the input format to the network and input multiple midi files which splits into two types of objects, Chords and Notes. A note object obtained from Music21 contains information about three things of music, offset, pitch values and octave. Below is an excerpt of the input format.

<music21.note.Note F>

<music21.chord.Chord B-2 F3>

<music21.note.Note E>

<music21.chord.Chord B-2 F3 >

<music21.note.Note D>

The chord objects are like container for set of notes that will be played at same time. The degree of sound which determines its extent (highness or lowness) depends, which is basically the frequency of the sound is pitch. It is represented with the letters from A to G. Octave is the interval between one musical pitch.

Another thing is the intervals between notes, as the notes can have varying intervals. Notes can occur in any form where there can be swift series for a time period followed by a pause where no note is being played. MIDI files read using Music21 provide the interval between two successive notes. We get to see it's generally 0.5 therefore we can ignore such small offset for our experiment and it will not have much effect on the melodies of the music.

<music21.chord.Chord E3 A3> 77.5

<music21.chord.Chord F3 A3> 78.0

<music21.chord.Chord F3 A3> 78.5

B. Preparing the data: We handle MIDI files from various musical genres in our data processing pipeline. We use the features of the Music21 library to process these MIDI files, reading and parsing them to produce note and chord objects. The input for our LSTM network consists of these note and chord objects. We use the Music21 converter to convert every MIDI file, producing stream objects composed of all the individual notes and chords in the song. The pitch number is encoded into string notation to be represented. To record the ID of Consistent music can be produced using this encoding strategy, making it easier to decode the network's output into intelligible notes and chords. [Insert Image] This categorical data is converted to integer-based values using one hot encoding since our network performs significantly better with integer-based values. The mapping function in the following image functions essentially as a one-hot encoder, one-hot encoding categorical variables like "orange" and "apple" into integer values 0 and 1, respectively. Machine learning performs more effectively with these encoded values than with string-based category variables.

C. Architecture of Networks: These layers comprise our model:

- Long Short-Term Memory: The way the traditional human brain learns is through repeated task performance.[9] We acquire abilities that expand upon our prior knowledge rather than having to start from scratch every time. A traditional neural network is unable to do this. For example, to understand what is happening at every frame in a video. Recurrent networks are networks because they have loops in them.
- Dropout layers: We use the Dropout approach to minimize the overfitting in neural networks.
- In our method, we input several MIDI files, which are separated into two different kinds of objects: notes and chords.
- Three fundamental elements of music are included in a note object that is obtained from Music21: offset, pitch values, and octave.

Here is an excerpted sample that illustrates the format of the input: Contrarily, chord objects act as holding spaces for simultaneous note sets that represent chords in a musical composition. Pitch: A sound's characteristic that establishes how high or low it is; it is directly associated with its frequency. Pitch is denoted by the letters A through G in musical notation. Every letter has a unique pitch and corresponds to a certain musical note. An octet is the space between two notes in a melody. Our neural network has to predict the next note or chord to be played with high accuracy to produce harmonious music. The intervals between notes also need to be taken into account because they might differ greatly. 4. Challenges and Opportunities Rapid note transitions are frequently followed by quiet intervals in musical compositions where no notes are played. Using Music21 to read MIDI files allows us to examine the time intervals between notes. These intervals are typically small, frequently 0.5. We may safely ignore such little offsets for our experiment because they do not affect the music's primary melodic structure.

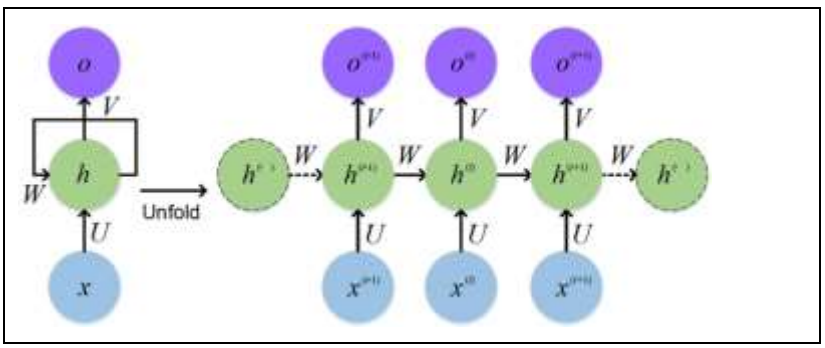

Fig. 1 – Standard RNN and Unfolded RNN

D. Generating Music: For crafting the soundtrack, we'll utilize the identical code employed during the model's training. The only change is to put the weights file into the model rather than the notes and chord objects. The network will be built using the same methodology as before. While the initial point for our musical composition is randomly chosen from the list, individuals desiring control over the starting point can implement a function to replace the existing random function. When composing music, the number of notes is flexible. We decided to use 1000 notes, which would result in about 4 minutes of music. We feed the corresponding sequence into the network to make each note. Expanding the length of the music is feasible by selecting more notes, though this will extend the creation time.

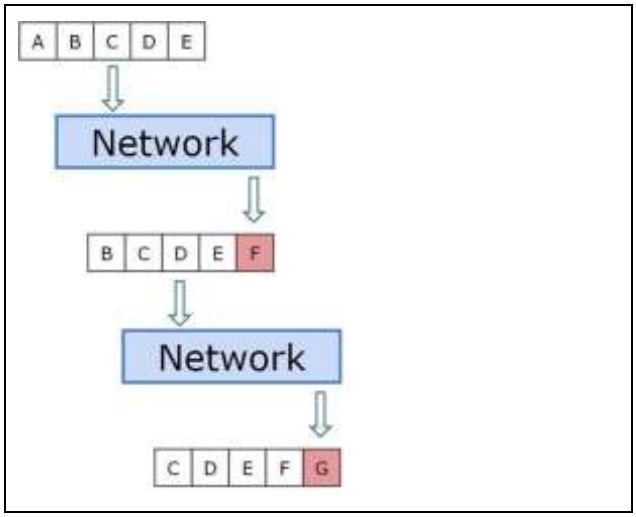

Fig. 2 – Music Generation

5. RESULTS AND DISCUSSION

The result of our research is the successful generation of music using the LSTM-based neural network model. After training the network with MIDI files, predominantly consisting of Final Fantasy soundtracks, the model produced a test_output.mid file that can be played on various music applications. The generated music showcases the model's ability to learn and replicate musical patterns from the input data.

Upon listening to the generated samples, some users noted the presence of dissonant or "weird" notes, indicating that while the model can produce coherent sequences, it does not always generate perfect melodies. This highlights an area for improvement in future iterations of the model.

Overall, the results demonstrate the feasibility of using LSTM networks for music generation, with the generated music providing a foundation for further refinement and experimentation. Future improvements could focus on better handling of timing and unknown patterns, as well as exploring the use of multiple instruments to enhance the richness and variety of the generated compositions.

Discussion:-

This study investigates the application of Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) for music generation, offering a novel method for autonomous music production. In order to create a system that can produce coherent musical sequences from MIDI files, the study emphasizes the integration of multiple technologies and frameworks. This talk explores the research's main conclusions, difficulties, and possible future paths.

Principal Results

Effective Music Generation: The main achievement of this research is the use of an LSTM-based neural network to generate music in an effective manner. After the network was trained on a dataset mainly composed of Final Fantasy soundtracks, a test output file that can be played on several music applications was produced. The music that is produced shows how capable the model is.

6. CONCLUSION AND FUTURE SCOPE

In this research, we developed an LSTM-based recurrent neural network for automatic music generation. The primary goal was to identify and model the relationships within MIDI files, capturing musical chords and notes to enable the creation of diverse musical compositions.[6] Utilizing the Python Keras framework and the Music21 library, built on TensorFlow, we successfully trained our neural network to generate music.

Our LSTM network demonstrated the capability to generate coherent musical pieces, suggesting that neural networks can indeed be used for music creation. Despite achieving good results, there are areas for improvement. The model struggled with perfect melody generation, often producing some dissonant notes. This limitation indicates the need for further refinement in handling timing and recognizing unknown patterns within the music data.

Future work could enhance the network's performance by incorporating additional classes to account for timing variations and implementing strategies to manage unrecognized inputs. Additionally, exploring the integration of multiple instruments could further diversify the generated music. Comparisons with other models, such as ensemble methods, and leveraging more powerful GPUs, could provide insights into achieving higher quality and more sophisticated music generation.

Future Scope: A promising method for autonomous music composition is the suggested LSTM-based recurrent neural network (RNN) for music generation. Nonetheless, a number of directions for further study and development might be investigated to improve its functionality and performance: To manage exact timing and rhythmic variances, use more classes. This might entail giving note durations and rests more complex representations, which would enable the network to produce music with more complex rhythmic patterns. Give the model a deeper understanding of dynamics (volume changes) so that it can generate more complex and expressive music. Create reliable systems to handle new notes or inputs for the network, maybe using methods like transfer learning or data augmentation. This would enhance the model's generalization abilities and stop it from crashing when it encountered unfamiliar musical parts.

Generation of Multi-Instrument Music: Extend the approach to managing several instruments at once to produce compositions that are more intricate and multi-layered. The network may be trained on polyphonic MIDI files, which simulate an ensemble performance by including several instrument tracks.

Enhancement of Harmony and Melody Production by improve the model's capacity to produce melodies and harmonies that are both aesthetically beautiful and cohesive. This would entail feeding the network with a larger and more varied dataset for training and integrating sophisticated ideas from music theory.

To make the model more versatile, train it on a greater range of musical genres. The network may be trained to produce music that complies with distinct genre-specific guidelines and traits by exposing it to a variety of genres and traditions. Provide tools that let users have a say in how music is created. This could be deciding on desired styles, establishing the opening themes for the music, or modifying key and speed.

Real-Time Production of Music: Enhance the model to produce music in real time, allowing users to create interactive applications where they can compose music as they go. This would necessitate increasing computational effectiveness and perhaps making use of GPU hardware acceleration

Collaborative Composition of Music: Provide collaborative tools that allow for the participation of numerous people or even AI systems to enhance the composition process, resulting in a more dynamic and participatory approach to music creation.

Metrics for Evaluation: Provide more thorough and uniform measurements to assess the calibre of generated music. This could include both subjective metrics like listener enjoyment and objective metrics like following musical theory.

Future research can greatly improve the capabilities of LSTM-based music generation models by addressing these areas, making them more adaptable, user-friendly, and able to generate high-quality musical compositions in a variety of genres and circumstances.

References

[1] Alexander Agung Santaso Gunawan1, Ananda Phan Iman2, Derwin Suhartono1“ Automatic Music Generator Using Recurrent Neural Network”(2020)

[2] Kleedorfer, F., Knees, P., Pohle, T.: Oh oh oh whoah! towards automatic topic detection in song lyrics. In: ISMIR. pp. 287–292 (2008)

[3] Hiller, L., Isaacson, L.M., “Experimental Music. Composition with an Electronic Computer”. McGraw-Hill Book Company (1959)

[4] K. Choi, G. Fazekas, and M. Sandler, “Text-based LSTM networks for Automatic Music Composition” (2016)

[5] Laden, B. and Keefe, D. H., “The representation of pitch in a neural net model of chord classification” (1989)

[6] Rothstein, J., “MIDI : a comprehensive introduction” . Oxford University Press, Oxford. (1992)

[7] Sabina-Cristina Necula & Vasile-Daniel Pavaloaia, “AI-Driven Recommendations: A Systematic Review of the State of the Art in E-Commerce”

[8] Ferdin Joe John Joseph, Sarayut Nonsiri, and Annop Monsakul, “Keras and TensorFlow: A Hands-On Experience”(2021)

[9] Sturm, Santos and Korshunova, "Folk Music Style Modelling by Recurrent Neural Networks with Long Short Term Memory Units"(2015)

[10] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” CoRR, vol. abs/1609.03499, (2016)

[11] Douglas Eck , Juergen Schmidhuber, A First Look at Music Composition using LSTM Recurrent Neural Networks, Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale (2002)

[12] Socher, Richard; Lin, Cliff; Ng, Andrew Y.; Manning, Christopher D., "Parsing Natural Scenes and Natural Language with Recursive Neural Networks", 28th International Conference on Machine Learning (ICML 2011