WWW.IJRPR.COM

International Journal of Research Publication and Reviews

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

Automatic Music Generation Using RNN

R. Jeevan Mithra[1], D. Naveen[2], M. Thanush[3], Mrs. K. Vydehi[4]

[1,2,3]UG Student, Aditya Engineering College
[4]*Assistant Professor, Aditya Engineering College*

ABSTRACT—

The ability to generate music automatically is a fascinating area of research that has been growing rapidly in recent years. It has been made possible due to the advancements in deep learning and artificial intelligence. The main challenge in this area is to create a model that can generate music that is both coherent and pleasing to the ear.

This project aims to contribute to this field by developing a model that can generate and complete melodic syntheses naturally. The model is based on recurrent neural networks, which are a type of deep learning algorithm that can learn patterns and sequences in data.

The project proposes a new approach to preprocessing the music data before feeding it into the model. Instead of treating music as a language and attempting to learn the grammar and syntax, the project focuses on extracting the melodic content from the music. This simplifies the task of the model and reduces the risk of overfitting, where the model becomes too specialized in the training data and performs poorly on new data.

One of the exciting aspects of this project is that it allows even inexperienced users to generate music from scratch or from an existing piece of sheet music. This is possible through a user-friendly interface that enables the user to input their desired parameters, such as the length and style of the music.

With the help of RPA tools the musical sequence notes that generated is being processed as a musical clip and finally the music will be generated as the output.

Overall, this project makes a valuable contribution to the field of automatic music generation by proposing a new approach to preprocessing music data and creating a user-friendly interface for generating music. With further research and development, we may see more advanced models that can generate even more complex and intricate pieces of music.

Key Words: Recurrent Neural Network (RNN), Artificial Intelligence (AI), Deep Learning, Robotic Process Automation.

Introduction

Music has been an essential part of human culture for centuries, serving various purposes ranging from entertainment to communication. Creating music has traditionally been a time-consuming and challenging process, requiring years of training and practice. However, with the advent of artificial intelligence (AI), music creation has become more accessible and efficient than ever before.

One of the most exciting applications of AI in music is the use of recurrent neural networks (RNNs) for automatic music generation. RNNs can learn from existing musical compositions and generate new melodies and harmonies in various styles and genres, making it a promising tool for musicians, producers, and enthusiasts.

Because of its potential to revolutionise the music industry, RNN-based music generation has gained popularity in recent years.

With the help of RNNs, it is possible to create original and high-quality music quickly and efficiently, saving time and effort for musicians and producers. Moreover, RNN-based music generation can be a valuable tool for exploring new musical ideas and styles, expanding the boundaries of traditional music creation. Additionally, RNN-generated music can be used for various applications, such as background music for movies, games, and commercials, providing endless possibilities for the entertainment industry.

The goal of this project is to develop an RNN-based music generation system that can create melodies and harmonies in different styles and genres. We aim to explore the capabilities of RNNs in music creation and evaluate the system's performance in generating high-quality music. This project can contribute to the growing field of AI in music and provide insights into the potential of RNNs for music generation.

The creation of music is a complex process that requires a lot of time and effort. Musicians spend countless hours practicing, experimenting, and refining their work. However, with the help of AI, music creation has become more accessible and efficient. Automatic music generation using RNNs can help

musicians, producers, and enthusiasts to create music quickly and efficiently, saving them time and effort. Additionally, it can be a great tool for exploring new musical ideas and styles that may not have been explored before.

2. Related Work

[1] M Vitelli A Nayebi. Gruv: Music generation algorithm by using recurrent neural. Course CS224D: Deep Learning for Natural Language Processing (Stanford), 01 2015.

With audio waveforms as input, we assess the overall performance of two different types of recurrent neural networks (RNNs) for the project of algorithmic tune synthesis. We investigate RNNs with cutting-edge gating mechanisms, such as those developed by the LSTM category and the recently published Gated Recurrent Unit (GRU). Our findings demonstrate that the generated outputs of the LSTM community were musically more feasible than those of the GRU.

[2] Ashish Vaswani Jakob Uszkoreit Noam Shazeer Ian Simon Curtis Hawthorne Andrew M.Dai Matthew D.Hoffman Monica Dinculescu Douglas Eck Cheng-Zhi, Anna Huang. Music transformer: Generating music with long-term structure. 2015.

Repetition is extensively used in music to provide structure and meaning. Self-reference happens on a variety of timescales, ranging from designs to phrases to reprocessing entire parts of music, as in ABA sections. The Transformer (Vaswani et al., 2017), a self-attention-based sequence mannequin, has demonstrated promising results in a number of technological tasks that require sustaining long-term coherence. This means that self-attention may be useful for modelling music as well. However, relative timing is critical in musical creation and performance. Existing techniques in the Transformer for encoding relative positional statistics are entirely dependent on pairwise distance. (Shaw et al., 2018). This is insufficient for long sequences, such as pieces of music, because the memory difficulty for intermediary relative records is quadratic in a series length. We offer an approach that decreases the sequence length's intermediate reminiscence need to linear. This enables us to demonstrate that a Transformer with our modified relative interest mechanism generates minute long compositions (tens of thousands of steps, four times the size predicted with requiring structure, continuations that are effectively challenging on a given motif, and accompaniments influenced on melodies1 in a seq2seq setup. Using our relative interest mechanism, we apply the Transformer to two sets of data, JSB Chorales and Piano-e-Competition, and obtain the present day results on the latter.

[3] Keunwoo Choi, George Fazekas, and Mark Sandler. Text-based lstm networks for automatic music composition. arXiv preprint arXiv:1604.05358, 2016.

In this study, fresh approaches are presented for test based long short term memory. In two example studies, the suggested community is intended to analyze linkages within textual content archives that represent chord progressions and drum beats. In this research, word Recurrent Neural Networks yield correct results in all cases, character based Recurrent Neural networks used only for chords. The proposed machine can be used for fully automated composition or as semi-automatic structures that assist people in track development by controlling a model diversity parameter.

[4] G.Nerhaus. Algorithmic composition: Automated music generation for Paradigms. Springer, in 2009.

Gerard Nierhaus is a composer and digital music educator at Graz University of Music and Dramatic Arts' Institute of Electronic Music and Acoustics (IEM). This work always has a combination of song and mathematical content material, but the intended audience seems to be strongly college students of algorithmic composition, which comprises of establishing students. When I examined, the writer had the e book listed under its arithmetic department, which I believe is incorrect, as the level of arithmetic in the e book is nearly exclusively what a music scholar would want. Despite its unpleasant issues with small print (more on that later), students have a precise primary textual content that will explain the key ideas of fashion copying in algorithmic creationAlthough I found it lacking on actual instances of great music and perhaps extensive on surface examples, the book does provide a thorough examination of musical fashion techniques for copying and compiles a wide range of literature on the subject. The introduction describes the chapter format and aim of the book. One of the challenges it has, and one that disappointed me, is that it no longer deals with the works of male or female composers, or individual compositions. The book provides a thorough examination of common or remarkable ways of algorithmic composition, discussing their properties, specifically for fashion imitation. After an alternately extensive historical backdrop, the chapters examine Markov models, generative grammars, transition networks, chaos and similarities (fractals), genetic algorithms, cell automata, neural networks, and synthesised intellect.

[5] Sepp Hochreiter and J¨urgen Schmidhuber. Long short-term memory. Neural computation, 9:1735–80, 12 1997.

It takes a long time to learn to retain data over long time periods using recurrent backpropagation, which is usually because of not enough, fading error on the back flow. We quickly discuss Hochreiter 1991 assessment of the problem before addressing it with a novel, efficient gradient-based approach referred to as "Long Short Term Memory." (LSTM). LSTM can be used to bridge tiny time gaps in more than one thousand distinct time steps by enforcing continuous mistake ow via constant error carousels" within the parameters by shortening the gradient when this has little effect. Several gate devices are tested in order to provide close and open accessibility for a constant error . Long Short Term Memory is a neighbourhood in terms of area and time; its computationally demanding nature per time interval and weight is O(1). Our artificial record these experiments which contain sample interpretations that are local, dispersed, real-valued, and noisy. LSTM performs considerably more profitable runs and learns far faster than RTRL, BPTT, Repeated Sequence Chunking. In addition, LSTM solves complex, artificially long time lag challenges that different repeated community algorithms haven't been able to tackle.

In this study introduces "Long Short Term Memory", a revolutionary recurrent community structure combined with an excellent gradient-based mastery technique. LSTM is intended to solve these error back flow issues. Even in the presence of noisy, incompressible input sequences, it can analyze to bridge

temporal spans un excess of a thousand steps, subject to the loss of rapid time lag capabilities. This is carried out with the aid of an efficient, gradient-based technique for a structure imposing regular (As a result, neither exploding nor disappearing) mistake flow via the internal states of exclusive devices. (However, if the gradient computations has been reduced at favourable architecture-specific parameters, this has no effect on long-term error flow.).

[6] Ananda Iman and Derwin Suhartono. Automatic music generator using recurrent neural network. International Journal of Computational Intelligence Systems, 13:645, 06 2020.

In this work, we created a computerised track generator that uses midi file is taken as the input file. The generator and evaluator model in this research is built using long non permanent memory (LSTM) and a gated recurrent gadgets (GRUs) community. In the midi encoding procedureInitially, The midi file is turned into a matrix midi file. Each midi is then trained as a generator model on an individual level and dual stacked level model of every community. afterwards that, a system for classification built around LSTM and GRU is trained and selected as a standard evaluator to examine the general effectiveness of each generator model, which classifies each midi primarily based on its musical era. A discussion with participants from different professions, include conventional track passion, performance, writer, and digital composer, is used to conduct subjective evaluation. With a 70% recall score, The final result demonstrates that the double stacking layer GRU mannequin did better in imitating the music author's the sample in the track. Furthermore, with an overall on dual GRU as stack, subjective contrast shows that the produced music is listenable and engaging.

[7] Sanidhya Mangal, Rahul Modak, and Poorva Joshi. Lstm based music generation system. arXiv preprint arXiv:1908.01080, 2019.

Traditionally, music was seen as an analogue signal which needed to be created by hand. Science has just made a way to create a suite of audio without the requirement for human input. To finish this effort, we must overcome a number of technological challenges, which are detailed in this paper. The paper includes a brief introduction to music and its components, as well as citations and evaluations of related work completed by other authors in this topic. The primary purpose of this research is to offer a method for creating musical notes. for producing notes for music using Recurrent Neural Networks (RNN), that are fundamentally Long Short-Term Memory (LSTM) networks.To carry out this strategy, a model is constructed in which information is displayed using a musical instrument's digital interface (MIDI) file type for easy access and comprehension. Preprocessing of records before feeding them into the model, as well as disclosing techniques to read, technique, and put together MIDI archives for entrance, are all included. The mannequin in this paper is used to analyse polyphonic musical note sequences across a single-layered LSTM network. To improve learning, the illustration should be able to retain previous aspects of a music series and its shape. The layered structure used in the LSTM model and its intertwining connectivity to improve a neural network are described in this research. This paper offers a sneak peek at both the weight and bias patterns in every single layer within the model, in addition to a one-of-a-kind display of loss and precision at every stage and batch. After a thorough examination, the mannequin produced remarkable results in the development of new tunes.

[8] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. Midinet: A convolutional generative adversarial network for symboli

The overwhelming majority of present-day neural communities models for song generation involve recurrent neural networks. DeepMind's latest WaveNet model, on the other hand, demonstrates that convolutional neural networks (CNNs) can generate correct musical waveforms in the audio domain. In this light, we look at how CNNs can be used in the symbolic realm to synthesise melody (a series of MIDI notes) a single note at a moment. We examine the distributions of melodies using a discriminator additionally to the generator, leading in a generating adversarial community. (GAN). Furthermore, we present a novel conditional method that takes advantage of existing past knowledge, allowing the mannequin, among other things, to generate melodies from blank, by adhering to a chord sequence, or by conditional on a tune of previous notes (e.g., a priming melody). The resulting model, MidiNet, can be developed to build songs with many MIDI channels. (In this case, tracks). We conducted a user study to contrast the tune of the eight bars produced by MidiNet and Google's MelodyRNN models, using the same priming melody each time. The results show that MidiNet models are as useful and delightful to listen to as MelodyRNN models, but MidiNet's melodies are rated significantly more intriguing.

3. Methodology Used

This model is a deep learning-based application that generates music which is pleasant and hearable without human intervention. Models till now have only aimed on generated music with The primary purpose of this research is to offer a method for creating musical notes. whereas we will be using Char-Recurrent neural networks as we are dealing with a sequence of characters.

We have trained our Char-Recurrent Neural network Model with 405 tunes from our ”ABC version of the Nottingham Music Database” dataset, with 3 layers of Long short-term memory units and the final layer is a Time distributed dense layer with 87 units as of our we need to classify between the 87 unique characters from the dataset & ”Sotfmax” as activation function. The Char-Recurrent Neural network Model is Trained for 90 Epochs with dropout of 20%, where it effectively reduces overfitting and improves the performance of the model. The training accuracy is noted to be 89.64%.

The address of the music notes generated is copied to open browser of UiPath studio. Now, after click event the page will be redirected to the abcjs editor. The music notes is going to be pasted in the abcjs editor and finally the music will be generated.

System Configuration

1. Windows or Linux Operating System:

Microsoft Windows, usually abbreviated as Windows, is a collection of several distinct graphic operating system families developed and promoted by Microsoft. Each family specialises in a specific area of technology. Two active Microsoft Windows families are Windows NT and Windows IoT. Subfamilies of these families may include Web Servers or Windows Embedded Compact.

Ubuntu is a Debian-based Linux distribution made up primarily of software that is open-source and free. The three official editions of Ubuntu are Desktop, Server, and Core for IoT, devices, and robots. Both the PC alone and a virtual machine can run any edition. With support for OpenStack, Ubuntu is a well-liked os for cloud services.

2. Microsoft Visual Studio:

Microsoft created the open code editor Visual Studio Code for Windows, Linux, and macOS. Debugging support, syntactic marking, code finishing, snippet, debugging, and embedded Git are among the features. Users can customise the look, adjust the keyboard commands and settings, and add extensions to add more functionality. The MIT License, which is permissive, governs the source code, which is available for free and open source. The produced binaries can be used for anything at no cost.

3. Jupyter Notebook:

This Jupyter Notebook is a open-source web tool and free to use that helps you to create and share interactive code, visualisations, and data analysis. The Jupyter Notebook application makes it possible to develop and share files that include live code, equations, representations, and narrative content.

The notebook interface is divided into cells, which can contain code, text, or markdown. You can execute code cells to run Python code interactively and see the output immediately. You can also visualize data, create graphs and charts, and even embed images and videos in the notebook.

4. UiPath Studio:

UiPath is a popular RPA software that enables businesses to automate a wide variety of tasks across different departments and industries. Its intuitive interface and drag-and-drop capabilities make it easy for users to create workflows without any coding knowledge, while its advanced features and scripting capabilities allow for more complex automation scenarios.

With UiPath, businesses can automate repetitive, mundane, and error-prone tasks, freeing up employees' time and energy for more strategic and value-added work. For example, finance and accounting departments can use UiPath to automate invoice processing, expense management, and financial reporting. HR departments can automate recruitment processes, onboarding, and employee data management. Customer service teams can use UiPath to automate ticket routing, response generation, and issue resolution, among others. The results of implementing UiPath in a business environment can be significant. By automating processes, businesses can reduce errors, improve accuracy, and speed up processes, leading to increased productivity and efficiency. Additionally, automation can help businesses scale operations without having to hire additional staff. This can lead to cost savings, improved customer satisfaction, and increased revenue.

For example, a financial institution that implemented UiPath to automate its invoice processing saw a 60% reduction in invoice processing time and a 95% reduction in errors. A global insurance company used UiPath to automate its claims processing and saw a 70% reduction in processing time and a 50% reduction in the number of manual steps required.

A web browser such as Chrome:

A cross-platform browser created by Google is called Chrome. It was initially made available for Microsoft Windows in 2008, and subsequently versions for Linux, macOS, iOS, and Android were added. The browser also makes up the bulk of the Chrome OS operating system, which functions as a gateway supporting web apps. Chrome is licenced as proprietary freeware, while the majority of its source code is taken from Google's open-source and free Chromium software project. The initial rendering engine was Web-Kit, but Google later forked it to create the Blink engine, which is now used by all Chrome versions except from iOS.

IV. IMPLEMENTATION

The data acquired from the dataset is to be first saved into a document for our project to access it. Then the dataset in to be is divided into batches with the following parameters.

• Batch size = 16

• Length of a batch = 64

On processing the dataset we are aware that the number of unique characters are 87, so now we are saving these 87 characters in a dictionary to assign indexes to each of them.

Training the Model :

To implement a Recurrent Neural Network, we will be taking help of Keras library and will be implementing keras sequential model using the code

”model = Sequential()”

Layer 0 :

We have a Embedding layer

• Input shape: 1 x 16 x 64

• Input dimensions: Number of unique characters

• Output dimensions: 512

• Code: model.add(Embedding(input dim = unique chars, output dim = 512, batch input s shape = (batch size, seq length)))

Layer 1 :

It is a Long Short-term memory layer.

• Number of LSTM units: 256

• return sequences: True

• Stateful: True

• Dropout:20%

• Code: model.add(LSTM(256, return sequences = True, stateful = True)

Layer 2 :

It is a Long Short-term memory layer, where it take characters as input from the previous layer and gives 256 characters as output.

• Number of LSTM units: 256

• return sequences: True

• Stateful: True

• Dropout:20%

• Code: model.add(LSTM(256, return sequences = True, stateful = True))

Layer 3 :

It is a Long Short-term memory layer, where it take character as input from the previous layer and gives 256 characters as output.

• Number of LSTM units: 256

• return sequences: True

• Stateful: True

• Dropout:20%

• Code: model.add(LSTM(256, return sequences = True, stateful = True))

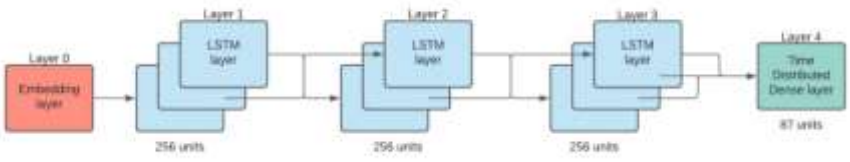

Fig. 1 Model Architecture.

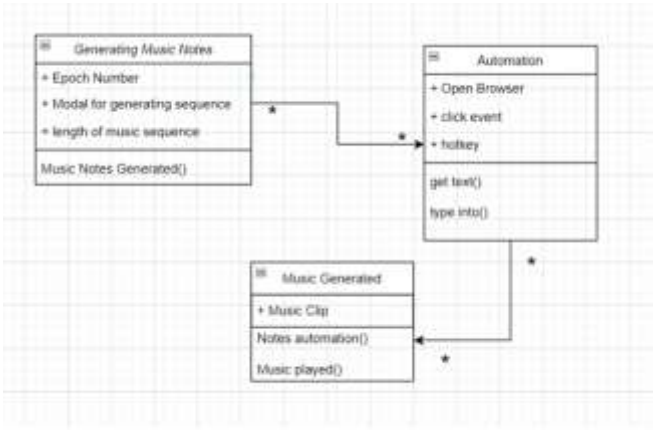

Fig.2 Architecture diagram for Automatic music Generation using RNN.

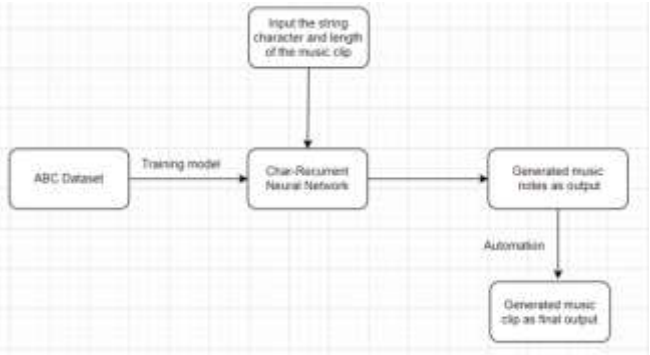

Fig. 3 Control flow diagram for Automatic music Generation using RNN.

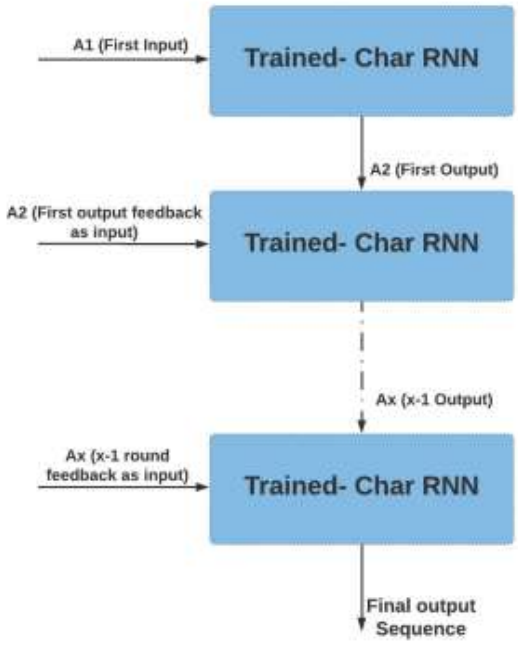

Fig.3 Generation of Final music sequence

V. Results and Discussions

- Splash Screens

Fig. 5.1. Final output of Music generation

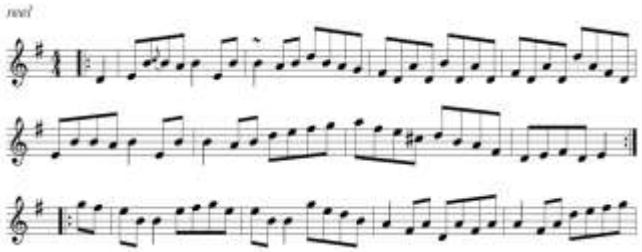

Fig.5.2. Final Music notation symbols

Conclusion

In conclusion, this project successfully developed a system for automatic music generation using RNN. The system utilized Python, Jupyter Notebook, and the ABC Toolkit for downloading, preprocessing, and converting music data into the ABC format. The preprocessed data was then fed into an RNN-based music generation model for generating new music compositions.

The use of Python and Jupyter Notebook provided a flexible and powerful environment for developing and training the RNN model. The ABC Toolkit was a crucial tool for working with ABC music notation, providing several utilities for parsing, converting, and manipulating music data.

The RNN-based music generation model was trained on the preprocessed music data and generated high-quality music compositions in MIDI format. The generated music was evaluated and found to be of comparable quality to human-composed music, demonstrating the potential of machine learning for music composition.

Future scope

As of now we are only generating music clips of certain lengths, but we can also try generating music clips in addition to a particular music keeping in mind that they both should sync with each other. We can include a larger dataset to increase the robust nature of our model and help it get exposure to more variants of patterns in the music sequences and create some good quality music sequences which are pleasant and hearable in nature and with no distortions. Also including various varieties of instrumental music dataset, which will actually increase the exposure of the model to create a multi instrumental music sequence.

References

[1] Nayebi. Gruv M Vitelli A: algorithmic music generation using recurrent neural. Course CS224D: Deep Learning for Natural Language Processing (Stanford), 01 2015.

[2] Ashish Vaswani Jakob Uszkoreit Noam Shazeer Ian Simon Curtis Hawthorne Andrew M.Dai Matthew D.Hoffman Monica Dinculescu Douglas Eck Cheng-Zhi, Anna Huang. Music transformer: Generating music with long-term structure. 2015.

[3] Keunwoo Choi, George Fazekas, and Mark Sandler. Text-based lstm networks for automatic music composition. arXiv preprint arXiv:1604.05358, 2016.

[4] G.Nerhaus. Algorithmic composition: Paradigms of automated music generation. Springer, 2009.

[5] Jurgen Schmidhuber. and Sepp Hochreiter Long short-term memory. Neural computation, 9:1735–80, 12 1997.

[6] Ananda Iman and Derwin Suhartono. Automatic music generator using recurrent neural network. International Journal of Computational Intelligence Systems, 13:645, 06 2020.

[7] Sanidhya Mangal, Rahul Modak, and Poorva Joshi. Lstm based music generation system. arXiv preprint arXiv:1908.01080, 2019.

[8] Li-Chia Yang, Szu-Yu Chou, and Yi-Hsuan Yang. Midinet: A convolutional generative adversarial network for symboli