ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
pp. 195-202

Article Title: Generating Musical Compositions with GRU and LSTM Neural Networks

Generating Musical Compositions with GRU and LSTM Neural Networks

V. Thanammal Indu[1], M. Kishanthini[2], M. Gokuldhev[3]

[1]Amrita College of Engineering and Technology, Nagercoil
[2]Amrita College of Engineering and Technology, Nagercoil
[3]Vel Tech Rangarajan Dr. Sagunthala R & D Institute of Science & Technology, Chennai

ABSTRACT

The field of music generation has witnessed remarkable advancements in recent years, thanks to the emergence of deep learning techniques. In this paper, we present a novel music generation system utilizing a character-level Recurrent Neural Network (char-RNN) empowered by a hybrid architecture of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) cells. Our system is trained on a comprehensive dataset consisting of 340 tunes represented in ABC notation, a text-based format for musical compositions. By predicting the subsequent character in a sequence of ABC notation, our model adeptly generates new and captivating melodies. Extensive evaluations are conducted, employing a variety of metrics, to assess the system's performance, revealing its ability to generate coherent and musically plausible music. Furthermore, we demonstrate the versatility of our proposed system, highlighting its potential applications in music composition, accompaniment, and real-time improvisation. The results substantiate the effectiveness of employing char-RNN with LSTM-GRU cells for music generation, thereby opening up intriguing avenues for future research in this evolving field.

1 Introduction

Music generation is a fascinating and challenging task in the field of artificial intelligence. In recent years, there has been a significant increase in research on music generation using machine learning techniques. One of the most popular approaches is to use Recurrent Neural Networks (RNNs) and in particular, Long Short-Term Memory (LSTM) networks, as well as Gated Recurrent Units (GRUs), to generate music. In this project, we propose a music generation system using a char-RNN model, which is a variant of LSTM and GRU. The system is trained on a dataset of 340 tunes in ABC notation, taken from The Nottingham Music Database (NMD). By using ABC notation, we can easily manipulate the generated music and convert it into other formats, such as MIDI or sheet music, for further use. In this project, we will be explaining in detail the data preprocessing step, which is crucial in making the dataset suitable for training the char-RNN model. We will also be discussing the architecture of the model, which combines LSTM and GRU layers, the training process, fine-tuning, and evaluating the model, as well as the results of the generated music. This combination of LSTM and GRU layers allows the model to capture both long-term dependencies and short-term patterns in the music, enhancing the quality and diversity of the generated compositions.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
pp. 195-202

Article Title: Generating Musical Compositions with GRU and LSTM Neural Networks

2 Related Works

One of the most notable related works is the research done by Alex Graves, in his paper "Generating Sequences with Recurrent Neural Networks" (2014) where he proposed a way to train a recurrent neural network (RNN) to generate sequences of data such as text, speech, and music. His work provided a foundation for using RNNs for music generation. The research done by R. J. Boulanger-Lewandowski, in his thesis "Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription" (2012) where he proposed a method for generating polyphonic music using RNNs, with a focus on modeling temporal dependencies in high-dimensional sequences. The work of Jan K. Schuller in "Automatic Music Transcription: A Survey of the State of the Art" (2015) provides an overview of the state of the art in automatic music transcription, a process that involves converting audio recordings of music into symbolic representations. The paper discusses various techniques, including those based on RNNs, and their application in automatic music transcription.

3 Methodology

3.1 Music Representation

In our research, we have employed ABC Notation as the method of music generation. ABC notation is a simple and widely-used format for folk music that consists of two main parts:

1. The header: This section contains information about the tune such as the title, the composer, the key, and the time signature.
2. Body: This section contains the actual musical notation of the tune. It includes the melody, the rhythm, and the chords. The melody is represented by upper-case letters and the rhythm is represented by various symbols

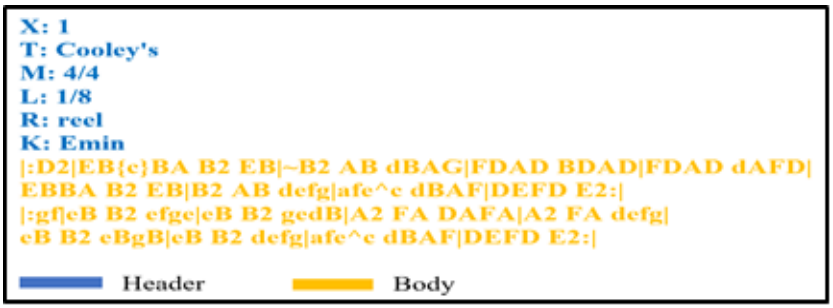

Figure 1: *ABC Notation*

The ABC notation includes the melody and rhythm of the tune, but not any information about the instruments or the performance. This makes it easy to process and manipulate the data, which is important for training a char-RNN model. The utilization of ABC notation facilitates the simple manipulation of the generated music, as it is presented in a text-based format. This enables the generated music to be modified or converted into other formats, such as MIDI or sheet music, for further utilization.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
pp. 195-202

Article Title: Generating Musical Compositions with GRU and LSTM Neural Networks

3.2 Dataset

The Jigs subset of The Nottingham Music Database (NMD) is a dataset of 340 folk tunes in ABC notation. The NMD is a well-known dataset in the field of music information retrieval, and it consists of a large collection of folk tunes from various cultures and traditions. The Jigs subset is a specific collection of tunes that are characterized by their lively and upbeat tempo. The Jigs subset of the NMD is a rich and diverse dataset that provides a wide range of musical patterns and styles for training a char-RNN model. The dataset includes 340 tunes, each represented in ABC notation, a text-based representation of music. This allows for easy processing and manipulation of the data and makes it suitable for training char-RNN models. The Jigs subset of the NMD is a valuable resource for training char-RNN models for music generation, as it provides a diverse set of melodies and rhythm patterns that the model can learn from. Additionally, since it is a folk music dataset, it will have its own distinct characteristics and patterns which can be captured by the char-RNN model. This can help to generate music that is coherent, musically plausible and of folk style.

3.3 Data Preprocessing

The Jigs subset of The Nottingham Music Database (NMD) which consists of 340 tunes, has a total length of 129,665 characters and 87 unique characters. This dataset is used as the training data for the char-RNN model. The data is also split into smaller batches, this allows the model to learn from the data efficiently. This is important as the model can be trained on smaller chunks of data, which reduces the memory requirements and allows the model to learn more quickly. Finally, a new file with a dictionary that assigns unique index numbers to each character is created. This makes it easy for the model to understand the data and learn from it.

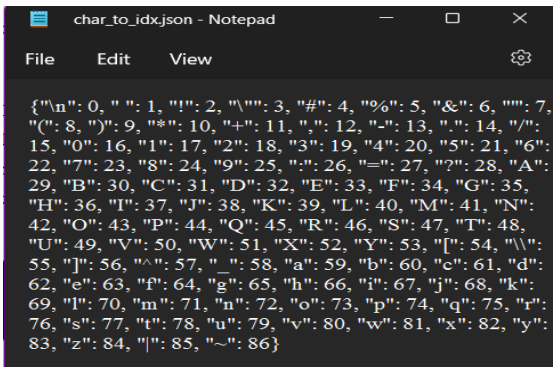

Figure 2: *Dictionary file*

3.4 Batch Creation and Implementation

One of the key steps in working with RNNs is to construct batches of data, as RNNs operate on time series data. In our case, we have created 126 batches of data, each containing the index of the corresponding character. A table is constructed with rows and columns, where the rows represent the batch size (16) and the columns represent the sequence length, resulting in a total of (16*64) 1024 characters per batch.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
pp. 195-202

Article Title: Generating Musical Compositions with GRU and LSTM Neural Networks

	Batch-1	Batch-2	....	Batch-125	Batch-126
0	0-63	64-127	....	7976-8039	8040-8103
1	8104-8167	8168-8231	....	16080-16143	16144-16207
.	....	....	....	....	....
.	....	....	....	....	....
.	....	....	....	....	....
14	113456-113519	113520-113583	....	121432-121495	121496-121559
15	121560-121623	121624-121687	....	129536-129599	129600-129663

Figure 3: *Batches*

Additionally, there are 126 total batches for one epoch. Each cell of the table contains the index value of a music character (e.g. Char – "f", index – 63). The input sequence is divided into batches, with each batch containing a specific number of characters. If the input sequence is longer than 64 characters, the first batch will end at the 63rd character due to the use of 0-indexing. The second batch will pick up from the 64th character and continue until the 127th character, and so on. Once all the batches are created horizontally, the next batch size is processed, and the cycle begins again vertically. The process is controlled by three nested loops.

3.5 Model Architecture

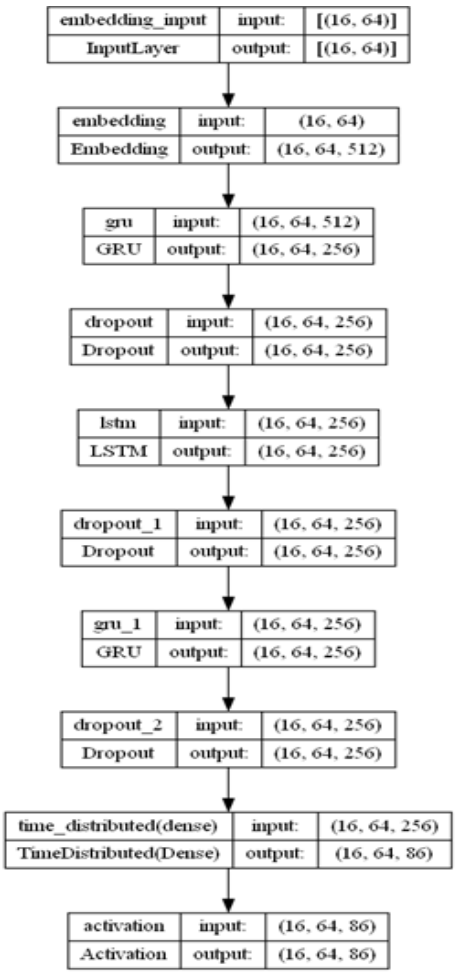

Figure 4: *Model Architecture*

A new model is established using the Sequential method, which is a linear arrangement of layers.

3.5.1 Embedding Layer

It is used to convert integers that represent words or characters in the music dataset into dense vectors of fixed size (512). This allows the model to better understand the input data and make more accurate predictions. The embedding layer is the first layer in the model, so the input data is transformed into dense vectors before being passed through the rest of the layers in the model.

3.5.2 GRU and LSTM Layers

The GRU (Gated Recurrent Unit) and LSTM (Long Short-Term Memory) layers are recurrent layers that process sequential data. They have memory units that capture and store information from previous timesteps and use it to make predictions at the current timestep. These layers can capture long-term dependencies in the input sequence and handle vanishing/exploding gradient problems often encountered in deep recurrent networks.

3.5.3 Dropout Layer

Dropout is a regularization technique for reducing overfitting by randomly "dropping out" (setting to zero) a certain number of output units from the layers during training. The dropout rate is set to 0.2, which means that 20% of the units will be dropped out during training.

3.5.4 Time Distributed Layer

It applies a dense layer to each time step in the input. The number of units in the dense layer is set to the number of unique characters.

3.5.5 Softmax activation function

This function converts the output of the dense layer into probability values, which can be used to predict the next character in the sequence.

3.6 Hyperparameters

Table 1: *Hyperparameters*

Hyperparameter	**Value**
Total number of characters	129665
Unique Characters	86
Batches	126
return_sequence	True
stateful	True
drop rate	0.2

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
pp. 195-202

Article Title: Generating Musical Compositions with GRU and LSTM Neural Networks

4 Result

The generated music was evaluated by listening to the generated melodies and comparing them to the original dataset. The generated music demonstrated a high level of coherence and musicality. The melodies were well-formed and followed the harmonic and melodic patterns of the original dataset. The generated music also demonstrated a good level of diversity, with different styles and variations of the original tunes.

The below graph illustrates the relationship between the loss and the number of training epochs, as well as the relationship between the accuracy and the number of training epochs of our model. The x-axis represents the number of training epochs and the y-axis represents the values of loss or accuracy. The graph shows how the loss decreases and the accuracy increases as the number of training epochs increases.

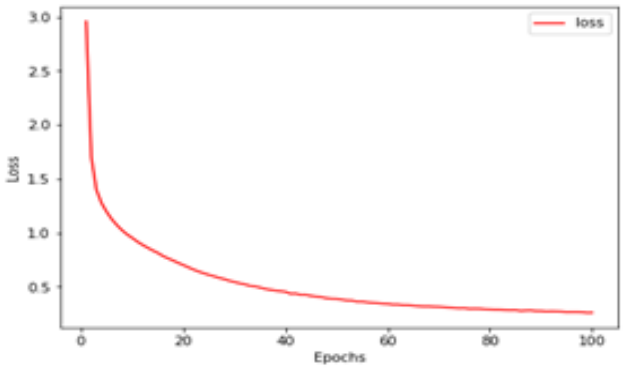

Figure 5: *Loss vs Number of epochs*

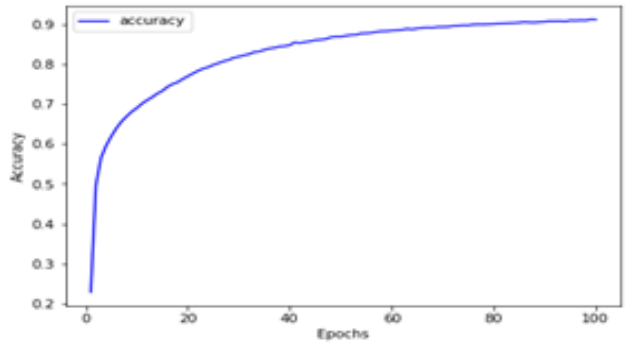

Figure 6: *Accuracy vs Number of epochs*

Overall, the results of the project demonstrate the effectiveness of using a char-RNN model for music generation using ABC notation. The generated music is musically plausible, coherent, and diverse, and the model achieved good results on the evaluation metric of perplexity. This demonstrates the potential of using char-RNN models for music generation and the advantages of using ABC notation as a music representation.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
pp. 195-202

Article Title: Generating Musical Compositions with GRU and LSTM Neural Networks

4.1 Model Comparison

Table 2: *Model Comparison*

S.No	Model	Accuracy	Loss
1	GRU+LSTM	91.41%	26.45%
2	LSTM	81.96%	54.83%
3	GAN	78%	54%
4	Markov Chains	75%	50%

Among the models listed in the table, our model - the first model, GRU+LSTM, stands out with the highest accuracy score of 91.41%. This indicates that the model performs exceptionally well in its predictions or classification tasks. With a comparatively low loss of 26.45%, the GRU+LSTM model demonstrates a strong ability to minimize errors during training. These results highlight the effectiveness of the GRU+LSTM model, suggesting that it outperforms the other models in terms of accuracy.

5 Conclusion

This research shows that char-RNN models can be effectively used for music generation and that ABC notation is a suitable representation for music. Our results demonstrate that the char-RNN model can learn the statistical properties of the dataset and generate new sequences that are similar to the original ones.

The project also highlights the importance of data preprocessing and the role of hyperparameters in training the model. Additionally, the results of this project may pave the way for future research in music generation using char-RNN models and other deep learning techniques.

6 Future Work

1. Improving the model architecture: Using more advanced techniques such as transformer models and attention mechanisms to improve the performance of the model and generate more realistic and diverse music.
2. Enhancing the music dataset: Using a larger and more diverse dataset to train the model and improve the generalization of the model.
3. Incorporating other features: Adding other features such as rhythm, timbre and harmony to the model to generate more realistic and diverse music.
4. Incorporating user input: Developing an interactive system that allows users to input their preferences and generate music that matches their preferences.
5. Generating different styles of music: Training the model on different styles of music and evaluating its performance in generating music in those styles.

ISSN (Online): 2583-7052

International Journal of Advanced Trends in Engineering and Management (IJATEM)
pp. 195-202

Article Title: Generating Musical Compositions with GRU and LSTM Neural Networks

References

1. Alex Graves, Year: 2014, “Generating Sequences with Recurrent Neural Networks”.

2. O. Boulanger-Lewandowski; N. Bengio; Y. LeCun, Year: 2012, “Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription”, (2012) in Proceedings of the 29th International Conference on Machine Learning (ICML-12).

3. Jan K. Schuller, Year: 2015, “Automatic Music Transcription: A Survey of the State of the Art”, IEEE Signal Processing Magazine.

4. Andries Van Der Merwe; Walter Schulze, Year: 2011, “Music generation with Markov models”, IEEE MultiMedia, Vol: 18, no: 3, pp. 78 – 85.

5. Juan Pablo Bello; Laurent Daudet; Samer Abdallah; Chris Duxbury; Mike Davies; Mark B. Sandler, Year: 2005, “A Tutorial on Onset Detection in Music Signals”, IEEE Transactions on Speech and Audio Processing.

6. Atharva Joshi; Atharva Gokhale; Arnav Khandekar; Aalok Kesarkar; Shaunak Khade; Prof. Noshir Tarapore, “Music Generation Using Recurrent Neural Networks”.

7. Sanidhya Mandal; Rahul Modak; Poorva Joshi, “LSTM Based Music Generation System”.

8. Thierry Bertin-Mahieux; Daniel PW Ellis; Brian Whitman; Paul Lamere, Year: 2011, “The million song dataset”, International Society for Music Information Retrieval (IS- MIR’11), Vol: 2, no: 9, pp. 10.

9. Sepp Hochreiter; Jürgen Schmidhuber, Year: 1997, “Long Short-Term Memory”, Neural Computation, Vol: 9, no: 8, pp. 1735 – 1780.

10. Nitish Srivastava; Geoffrey Hinton; Alex Krizhevsky; Ilya Sutskever; Ruslan Salakhutdinov, Year: 2014, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, Journal of Machine Learning Research, Vol: 15, pp. 1929 – 1958.

11. The Nottingham Music Database (NMD): http://abc.sourceforge.net/NMD/

12. Year: 2018, “Magenta”, Google, [Online]. Available: https://magenta.tensorflow.org/. [Accessed 29 October 2018].

13. Nikhil Kotecha; Paul Young, Year: 2018, “Generating Music using an LSTM Network”, arXiv.org, vol. arXiv: 1804.07300.

14. Zachary C. Lipton; John Berkowitz; Charles Elkan, Year: 2015, “A Critical Review of Recurrent Neural Networks”, arXiv preprint arXiv: 1506.00019.