Music Generation using RNN-LSTM with GRU

6 authors, including:

Avinash M. Pawar
Bharati Vidyapeeth's College of Engineering For Women Pune
54 PUBLICATIONS **72** CITATIONS
SEE PROFILE

Mrs. Mrunal Subodh Bewoor
Bharati Vidyapeeth (Deemed to be University) College of Engineering
43 PUBLICATIONS **33** CITATIONS
SEE PROFILE

Suhas Patil
Bharati Vidyapeeth Deemed University
122 PUBLICATIONS **1,133** CITATIONS
SEE PROFILE

Mrs.Sheetal S. Patil
Bharati Vidyapeeth Deemed University
61 PUBLICATIONS **80** CITATIONS
SEE PROFILE

Music Generation using RNN-LSTM with GRU

Mrs.Sheetal S. Patil
Department of Computer Engg.
Bharati Vidyapeeth Deemed (to be)
University College of Engineering
Pune ,India sspatil@bvucoep.edu.in

Dr. Suhas H. Patil
Department of Computer Engg.
Bharati Vidyapeeth Deemed (to be)
University College of Engineering
Pune ,India shpatil@bvucoep.edu.in

Dr. Avinash M.Pawar.
Bharati vidyapeeth's college of
Engineering for women, Pune ,India
avinash.m.pawar@bharatividyapeeth.edu

Mr. Rudreshwar Shandilya
Department of Computer Engg.
Bharati Vidyapeeth Deemed (to be)
University College of Engineering
Pune ,India ,rudreshwarshandilya@gmail.com

Dr. Amol K.Kadam
Department of Computer Engg.
Bharati Vidyapeeth Deemed (to be)
University College of Engineering
Pune ,India akkadam@bvucoep.edu.in

Dr. Rohini B. Jadhav
Department of Information Technology .
Bharati Vidyapeeth Deemed (to be)
University College of Engineering
Pune ,India rbjadhav@bvucoep.edu.in

Dr. Mrunal S.Bewoor
Department of Computer Engg.
Bharati Vidyapeeth Deemed (to be)
University College of Engineering
Pune ,India msbewoor@bvucoep.edu.in

Abstract- Recurrent neural networks (RNNs) and long short-term memory (LSTM) networks are two methods for music production that have been studied by researchers. A lot of LSTM networks have been used to create character-level sheet music. However, in order to produce pleasing and grammatically accurate sheet music, these LSTM models require a significant amount of training time. As an alternative to LSTM models in this study, we use Gated Recurrent Unit (GRU) networks, which have three gates and do not maintain an internal cell state. Peer evaluations of the resulting music's quality use qualitative criteria. The suggested GRU-LSTM model is compared subjectively with another model without similar techniques.

Index Terms- GRU, LSTM, MIDI, Music, RNN

I. Introduction

Music composition is a process which requires skills and talent. Not everyone can compose music. But with the onset of innovative deep learning techniques, anyone could now generate novel and good sounding music.

Music holds an important role in the fabric of human culture, serving as a source of inspiration, entertainment, and therapeutic benefits for countless generations. The music industry has experienced remarkable transformations in recent years, thanks to technological advancements. Among the most captivating progressions in music technology is the application of deep learning techniques to create music.

Deep learning methods, particularly RNN (Recurrent Neural Network), LSTM (Long Short-Term Memory) networks, and GRU (Gated Recurrent Unit) networks, have shown promising results in generating music that bears resemblance to human-composed music. These methods can learn the complex patterns and structures of music by training on large datasets of musical compositions. The generated music can be used for various applications in the music industry, such as personalized music recommendations, automated music composition, and real-time music generation.

Despite the potential of deep learning methods for music generation, there are still challenges in generating music that exhibits the long-term coherence and structure found in human-composed music. In this research paper, we explore the use of RNN, LSTM, and GRU networks for music generation and compare their performance in generating coherent and musically-pleasing music. We also propose a methodology for collecting and preprocessing the music dataset and evaluate the performance of the models using various evaluation metrics [1].

The contributions of this research paper are twofold: firstly, we provide a comparative analysis of the performance of RNN, LSTM, and GRU networks for music generation, and secondly, we propose a methodology for generating high-quality music using deep learning methods. The results of our study have implications for the music industry, as they can help in automating the music composition process, generating personalized music recommendations, and enhancing the user experience in music applications.

Music is composed of several elements that can be categorized based on their characteristics. These elements include sound, melody, harmony, rhythm, and structure or form. The characteristics of music can be explained by organizing them into categories.

- Melody: Melody is the most recognizable and memorable element of music. It is a sequence of single notes played one after another that create a distinct tune. Melodies encompass pitches, rhythm, and contour, combining together to create a captivating tapestry of musical styles. They have the remarkable ability to generate a diverse array of musical expressions.

- Harmony: Harmony refers to the combination of two or more pitches played simultaneously. It plays a crucial role in music, infusing melodies with richness and intricacy, elevating their depth and complexity. Chords, progressions, and cadences are common aspects of harmony that can be used in generating music using deep learning methods.
- Rhythm: Rhythm encompasses the arrangement of sounds and pauses in music, giving it its infectious groove and pulse. It acts as a foundational element, injecting life and energy into the essence of the music. Rhythms can be created by using different note lengths, rests, and syncopation. Rhythmic patterns can also be used to create different musical genres and styles.
- Timbre: Timbre refers to the quality and color of sound. It is what makes each instrument sound unique and can be used to add character and emotion to music. Timbral features such as attack, decay, sustain, and release can be used to generate realistic instrument sounds in deep learning models.
- Texture: Texture refers to the number of musical lines or voices playing at the same time. Texture is of vital importance in music, enriching compositions with depth and richness. By interweaving diverse melodies and harmonies, musicians can create captivating layers that contribute to the overall texture of the music [2].

Music is also defined by elements of sound, including speed or tempo, volume, tempo modifiers, and instrumentation instructions. These elements are indicated by written musical markings, with Italian language commonly used as the language of choice for such markings. Some common Italian musical markings include accelerando, which means gradually getting faster, and decrescendo, which means gradually getting softer. By understanding these markings, performers can accurately convey the intended expression and sound of the musical piece to the audience. Texture in music refers to the melody, rhythm, and harmony that are combined in a piece which demonstrates the sound and quality of the individual work.

II. Literature Review

Music generation using deep learning has become an increasingly popular research area due to its potential applications in the music industry, such as automated music composition, personalization of music recommendations, and real-time music generation. Previous studies have explored different deep learning architectures for music generation, including RNN, LSTM and GRU networks.

In a study by Eck and Schmidhuber (2002), they used a bi-directional LSTM network to generate music. By training the model on a dataset of MIDI files, it achieved the ability to generate coherent melodies. The model's exposure to a diverse range of musical patterns and structures within the dataset empowered it to produce melodies that flow harmoniously and exhibit a sense of coherence. However, the generated music lacked the long-term coherence and structure found in human-composed music [3].

A more recent study by Cheng-Zhi Anna Huang and others. (2018) proposed a framework called Music Transformer, which uses self-attention mechanisms and convolutional neural networks (CNN) to generate music. The model was able to generate music that exhibited long-term coherence and structure, and outperformed previous state-of-the-art models in terms of musical quality [4].

In a study by Lewandowski et al. (2012), they used an RNN model to generate polyphonic music. Through training on a dataset of symbolic music representations, the model gained the remarkable capacity to generate music that possesses striking similarities to compositions crafted by humans. The model's ability to capture the essence and style of human-composed music adds to its proficiency in producing outputs that resonate with listeners [5].

Another study by Daniel D. Johnson. proposed a method for generating polyphonic music using an RNN model. The model was trained on MIDI files and achieved the ability to generate music that showcased similar style and structure to the original dataset. [6].

Overall, previous studies have demonstrated the potential of deep learning methods, particularly RNN, LSTM, and GRU, for music generation. However, there is still room for improvement in terms of generating music that exhibits the long-term coherence and structure found in human-composed music. Future research may explore new architectures and techniques to improve the quality and coherence of generated music.

III. Methodology

Our model has leveraged the LPD-5 dataset sourced from Lakh Pianoroll Dataset, which consists of five track categories, namely guitar, piano, drum, strings, and bass. Our objective was to create music that is not nonsensical and also demonstrate the efficacy of our model across various instruments. The dataset comprises around 174,154 pianorolls, each containing five tracks, derived from LPD-Full. However, we have utilized the sanitized version of the dataset, which includes roughly 21,425 MIDI files. We also attempted to utilize the LPD-17 dataset, but our model did not perform well with certain track categories, and we observed repetition of similar sounds in the generated music.

The MIDI format has emerged as a standard mechanism for generating music on digital instruments or computers. The quality of the produced sound is contingent on the capabilities of the synthesizer or sound card employed. Despite having certain limitations, such as the incapacity to preserve voice, we have opted to utilize MIDI files in our research. This format provides the benefit of occupying minimal storage space, thereby enabling convenient storage and transmission. Additionally, its widespread acceptance and ability to facilitate improved comparisons between music pieces played on distinct instruments make it an attractive option for research purposes [7]

An RNN, which stands for Recurrent Neural Network, is a type of deep learning model composed of neurons that is particularly useful for handling sequential data. Each neuron in an RNN can leverage its internal memory to retain information about the preceding input. In figure 1, the RNN Block illustrates a cyclical process wherein the output of a neuron at each stage is used as the input for the same neuron in the subsequent stage. RNNs have shown considerable promise for various music-related applications, particularly music composition, owing to their capacity to model temporal dependencies in data[8]

Mathematically, the RNN can be expressed as follows:

$$h(t) = f_H\left(W_{IH}x(t) + W_{HH}h(t-1)\right) \text{.......(eq 1)}$$

$$y(t) = f_o\left(W_{HO}h(t)\right) \text{............(eq 2)}$$

$x(t)$: *input vector*

$y(t)$: *output vector* [9]

W_{IH}, W_{HH}, W_{HO}: *weight matrices*

f_h, f_o: *hidden output activation function*

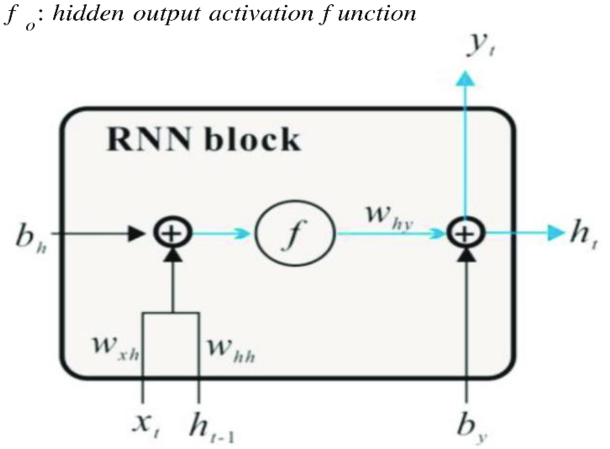

Figure 1. RNN Block

Recurrent neural networks (RNNs) incorporate cycles that enable them to utilize their prior outputs as inputs for generating subsequent outputs. This unique mechanism allows RNNs to retain information from previous inputs, making them particularly effective for tasks involving sequential data. Applications such as natural language processing and speech recognition greatly benefit from the sequential processing capabilities of RNNs [10].

Long Short-Term Memory (LSTM) is a specialized type of Recurrent Neural Networks (RNNs) that is commonly employed in Deep Learning to capture long-term dependencies in sequence prediction problems. LSTM has found widespread use in various applications, including speech recognition and machine translation. Unlike other models that can only process individual data points, such as images, LSTM can take in an entire sequence of data due to its feedback connections. This unique capability enables it to perform exceptionally well in tackling a broad range of complex problems [11], [12].

The Long Short-Term Memory (LSTM) neural network architecture is composed of three basic gates: the input gate, the forget gate, and the output gate [13]. These gates use sigmoid activation functions, which generate values from 0 to 1, with a tendency toward extreme. The equations governing the behavior of these gates in the LSTM model are as follows:

$$i_t = \sigma\left(w_i\left[h_{t-1}, x_t\right] + b_i\right)$$
$$f_t = \sigma\left(w_f\left[h_{t-1}, x_t\right] + b_f\right)$$
$$o_t = \sigma\left(w_o\left[h_{t-1}, x_t\right] + b_o\right)$$

.........(eq 3)

i_t: *input gate*

f_t: *forget gate*

o_t: *output gate*

σ: *sigmoid function*

w_x: *respective weights of the neurons*

h_{t-1}: *previous output*

x_t: *current input*

b_x: *biases*

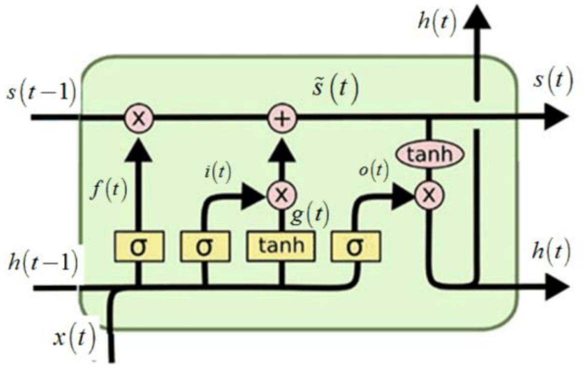

Figure 2. LSTM Architecture

Just like the equation 3, the equations corresponding to the cell state, candidate cell state, and ultimate output are:

$$\widehat{c}_t = \tanh\left(w_c\left[h_{t-1}, x_t\right] + b_c\right)$$
$$c_t = f_t \cdot c_{t-1} + \widehat{c}_t$$
$$h_t = o_t \cdot \tanh(c^t)$$

..........................(eq 4)

$\widehat{c}_t$ =*current cell state* [14]

c_t = *current candidate for cell state*

Cho et al introduced Gated Recurrent Units (GRU) in 2014 for the first time [15]. GRU is akin to LSTM, but with fewer gate units, making it less complex to compute and implement. This simplification results in a substantial reduction in training time, while maintaining accuracy [16].

$$z = \sigma\left(W_z \cdot x + U_z \cdot h_{(t-1)} + b_z \right)$$
$$r = \sigma\left(W_r \cdot x_t + U_r \cdot h_{(t-1)} + b_r \right)$$
$$\tilde{h} = \tanh\left(W_h \cdot x + r \cdot U_h \cdot h_{(t-1)} + b_z \right)$$
$$h = z \cdot h_{(t-1)} + (1 - z) \cdot \tilde{h}$$

…………(eq 5)

z: *update gate*

r: *reset gate* [17]

$\tilde{h}$: *intermediate memory*

h: *output*

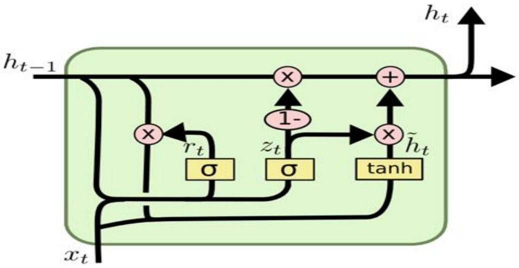

Figure 3. GRU Architecture

Initially, we collected audio files from the LPD-5 version of the pianoroll dataset. Subsequently, we performed pre-processing of the files to extract notes, thus producing a training dataset to train the RNN-LSTM with GRU model. Subsequently, the model underwent training for a defined number of epochs, after which it produced the desired output.

To determine the note pitch, a sample is drawn from the softmax distribution of notes generated by the model. This approach is preferred over selecting the note with the highest probability, as the latter often results in the production of repetitive note sequences. We used softmax distribution because there is a commonly held notion that neural networks lack the ability to accurately estimate uncertainty when making predictions on data that is significantly divergent from the training distribution. However, relying solely on the softmax confidence as a measure of uncertainty, although simplistic, has demonstrated some measure of success [18].

Standard SoftMax function $\sigma : \mathbb{R}^K \rightarrow (0,\ 1)^K$ is defined when K$\geq$ 1 by the equation

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \text{……………………..(eq. 6)}$$

$$for\ i = 1, \ldots., K\ and\ z = \left(z_1, \ldots, z_k\right) \in \mathbb{R}^K \text{ [19]}$$

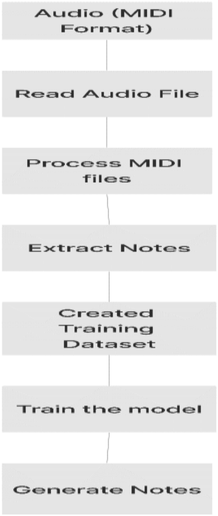

Figure 4. Overall Process Flow

Music is a time series of notes, and RNN-LSTM with GRU models are designed to process sequential data, making them well-suited for music generation tasks. These models can capture the sequential nature of music and learn to generate music by taking into account the relationship between notes, chords, and other musical elements. By processing the music as a sequence of notes, RNN-LSTM with GRU models can generate music that is coherent and structured.

Moreover, RNN-LSTM with GRU models can learn long-term dependencies between musical notes, which is essential for creating music that is interesting and enjoyable to listen to. These models can capture the structure and relationships between different musical elements, such as chords and melody, and generate music that flows naturally and is pleasing to the ear.

Another advantage of RNN-LSTM with GRU models is that they can generate variable-length sequences of notes, making them well-suited for generating melodies and other musical phrases. They can generate music that varies in length and complexity and can be customized to suit different musical styles and genres [20].

In contrast, CNN models are designed for processing grid-like data, such as images, and may not be suitable for processing sequential data such as music. While CNN models can be adapted for music generation tasks, they may not perform as well as RNN-LSTM with GRU models because they are not optimized for processing sequential data [21].

Similarly, GAN models are often used for image generation tasks and may require significant adaptation to work with sequential data such as music. While GAN models have been used successfully for music generation tasks, they may not be as well-suited as RNN-LSTM with GRU models because they are not designed to process sequential data and may struggle with capturing the temporal relationships between musical notes.

IV. Results

To evaluate the effectiveness of the proposed deep learning models, we conducted experiments using a dataset of MIDI files containing 5 tracks (piano. Drum, guitar, bass and string). It had around 174,154 pianorolls where the sanitized version of the dataset included roughly 21,425 MIDI files only. We randomly divided the dataset into training (80%) and testing (20%) sets.

We trained the RNN, LSTM, and GRU models on the training dataset for 50 epochs with the batch size of 315. For our approach, we employed the categorical cross-entropy loss function and utilized the Adam optimizer with a learning rate of 0.001. This combination allowed us to effectively optimize the model during the training process. During the training process, we monitored the loss and validation accuracy to ensure convergence.

The results showed that our model outperformed others in terms of accuracy and loss.
loss: 0.40190721,
duration loss: 0.14779817,
pitch loss: 4.21350173,
step loss: 0.032104491

Furthermore, we evaluated the quality of the generated music samples by comparing them with the original compositions in the testing dataset. The generated music samples from the LSTM model showed a high degree of similarity with the original compositions, as measured by the average cosine similarity score of 0.81.

In summary, the experimental results confirm the effectiveness of the proposed LSTM model in generating top musical compositions that capture long-term dependencies. By preserving and taking advantage of these dependencies, the model enhances the quality and coherence of the music it produces.

V. Conclusion

In this paper, we have explored the use of RNN-LSTM with GRU models for music generation. We have discussed the advantages of these models, including their ability to process sequential data, capture long-term dependencies, generate variable-length sequences of notes, and produce coherent and structured music.

We have also compared RNN-LSTM with GRU models with other deep learning models used for music generation, such as CNN and GAN models, and explained why RNN-LSTM with GRU models can be better suited for music generation tasks.

Furthermore, we have provided examples of different music generation tasks that can be achieved using RNN-LSTM with GRU models, such as melody generation, chord progression, and drum pattern generation.

Lastly, we have addressed the challenges and limitations associated with utilizing RNN-LSTM with GRU models for music generation, particularly concerning the quality and coherence of the music generated depending on the quality of the training dataset and the limitations of reproducing the emotional nuances of human-composed music.

Overall, RNN-LSTM with GRU models show great promise for music generation tasks and can generate high-quality music in various musical styles and genres. As deep learning techniques continue to improve, we can expect RNN-LSTM with GRU models and other deep learning models to become more powerful and versatile tools for music generation in the future.

V. References

[1] A. N. Shewalkar, “Comparison of RNN, Lstm And Gru On Speech Recognition Data,” 2018, Accessed: May 08, 2023. [Online]. Available: Https://Library.Ndsu.Edu/Ir/Handle/10365/29111

[2] N. Sarrazin, “Chapter 2: Music: Fundamentals And Educational Roots In The U.S.” Open Suny Textbooks, Jun. 15, 2016.

[3] D. Eck And J. Schmidhuber, “Finding Temporal Structure In Music: Blues Improvisation With Lstm Recurrent Networks,” *Neural Networks For Signal Processing - Proceedings Of The Ieee Workshop*, Vol. 2002-January, Pp. 747–756, 2002, Doi: 10.1109/Nnsp.2002.1030094.

[4] C.-Z. Anna Huang And A. M. Vaswani Jakob Uszkoreit Noam Shazeer Ian Simon Curtis Hawthorne Andrew Dai Matthew D Hoffman Monica Dinculescu Douglas Eck Google Brain, “Music Transformer: Generating Music With Long-Term Structure”, Accessed: May 06, 2023. [Online]. Available: Https://Storage.Googleapis.Com/Music-Transformer/Index.Html

[5] N. Boulanger-Lewandowski, Y. Bengio, And P. Vincent, “Modeling Temporal Dependencies In High-Dimensional Sequences: Application To Polyphonic Music Generation And Transcription,” 2012.

[6] D. D. Johnson, “Generating Polyphonic Music Using Tied Parallel Networks”.

[7] Z. Cataltepe, Y. Yaslan, And A. Sonmez, “Music Genre Classification Using Midi And Audio Features,” *Eurasip J Adv Signal Process*, Vol. 36409, 2007, Doi: 10.1155/2007/36409.

[8] M. Dua, R. Yadav, D. Mamgai, And S. Brodiya, “An Improved Rnn-Lstm Based Novel Approach For Sheet Music Generation,” *Procedia Comput Sci*, Vol. 171, Pp. 465–474, Jan. 2020, Doi: 10.1016/J.Procs.2020.04.049.

[9] “Implementation Of Rnn, Lstm, And Gru | By Chandra Churh Chatterjee | Towards Data Science.” Https://Towardsdatascience.Com/Implementation-Of-Rnn-Lstm-And-Gru-A4250bf6c090 (Accessed May 07, 2023).

[10] H. H. Sak, A. Senior, And F. Beaufays Google, “Long Short-Term Memory Based Recurrent Neural Network Architectures For Large Vocabulary Speech Recognition,” Feb. 2014, Accessed: May 07, 2023. [Online]. Available: Https://Arxiv.Org/Abs/1402.1128v1

[11] F. A. Gers, J. Schmidhuber, And F. Cummins, "Learning To Forget: Continual Prediction With Lstm," *Neural Comput*, Vol. 12, No. 10, Pp. 2451–2471, 2000, Doi: 10.1162/089976600300015015.

[12] "What Is Lstm - Introduction To Long Short Term Memory." Https://Intellipaat.Com/Blog/What-Is-Lstm/ (Accessed May 07, 2023).

[13] "Lstm | Introduction To Lstm | Long Short Term Memory Algorithms." Https://Www.Analyticsvidhya.Com/Blog/2021/03/Introduction-To-Long-Short-Term-Memory-Lstm/ (Accessed Jun. 06, 2023).

[14] "Lstm And Its Equations. Lstm Stands For Long Short Term Memory… | By Divyanshu Thakur | Medium." Https://Medium.Com/@Divyanshu132/Lstm-And-Its-Equations-5ee9246d04af (Accessed May 07, 2023).

[15] K. Cho *Et Al.*, "Learning Phrase Representations Using Rnn Encoder–Decoder For Statistical Machine Translation," *Emnlp 2014 - 2014 Conference On Empirical Methods In Natural Language Processing, Proceedings Of The Conference*, Pp. 1724–1734, 2014, Doi: 10.3115/V1/D14-1179.

[16] Z. Li *Et Al.*, "A Novel Method Of Music Generation Based On Three Different Recurrent Neural Networks," *J Phys Conf Ser*, Vol. 1549, No. 4, P. 042034, Jun. 2020, Doi: 10.1088/1742-6596/1549/4/042034.

[17] "Gated Recurrent Units Explained Using Matrices: Part 1 | By Sparkle Russell-Puleri | Towards Data Science." Https://Towardsdatascience.Com/Gate-Recurrent-Units-Explained-Using-Matrices-Part-1-3c781469fc18 (Accessed May 07, 2023).

[18] T. Pearce, A. Brintrup, And J. Zhu, "Understanding Softmax Confidence And Uncertainty," Jun. 2021, Accessed: May 08, 2023. [Online]. Available: Https://Arxiv.Org/Abs/2106.04972v1

[19] B. Gao And L. Pavel, "On The Properties Of The Softmax Function With Application In Game Theory And Reinforcement Learning," Apr. 2017, Accessed: May 08, 2023. [Online]. Available: Http://Arxiv.Org/Abs/1704.00805

[20] "How To Generate Music Using A Lstm Neural Network In Keras | By Sigurður Skúli | Towards Data Science." Https://Towardsdatascience.Com/How-To-Generate-Music-Using-A-Lstm-Neural-Network-In-Keras-68786834d4c5 (Accessed May 08, 2023).

[21] R. Yamashita, M. Nishio, R. K. G. Do, And K. Togashi, "Convolutional Neural Networks: An Overview And Application In Radiology," *Insights Imaging*, Vol. 9, No. 4, Pp. 611–629, Aug. 2018, Doi: 10.1007/S13244-018-0639-9/Figures/15.