**Automatic Music Generation**

A Project Report

submitted in partial fulfillment of the requirements

AIML FUNDAMENTALS WITH CLOUD COMPUTING

&

GEN AI

by

**VELMURUGAN.K**

**NM ID: au22me305**

[velmurugan25kl10@gmail.com](mailto:velmurugan25kl10@gmail.com)

Under the Guidance of

**P.Raja, Master Trainer**

**ACKNOWLEDGEMENT**

We would like to take this opportunity to express our deep sense of gratitude to all individuals who helped us directly or indirectly during this thesis work.

Firstly, we would like to thank my supervisor, P. Raja master trainer, for being a great mentor and the best adviser I could ever have. His advice, encouragement and the critics are a source of innovative ideas, inspiration and causes behind the successful completion of this project. The confidence shown in me by him was the biggest source of inspiration for me. It has been a privilege working with him for the last one year. He always helped me during my project and many other aspects related to the program. His talks and lessons not only help in project work and other activities of the program but also make me a good and responsible professional.

**TABLE OF CONTENTS**

**Abstract** 04

**List of Figures** 03

**Chapter 1, Introduction** 05

1.1 Problem Statement

1.2 Motivation

1.3 Objectives

1.4. Scope of the Project

**Chapter 2, Literature Survey** 06

**Chapter 3, Proposed Methodology** 13

**Chapter 4, Implementation and Results** 15

**Chapter 5, Discussion and Conclusion** 23

**References**  24

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURES** | **DESCRIPTION** | **Page No.** |
|  | Training and Validation Data for Debugging | **17** |
|  | WaveNet and LSTM Validation | **18** |
|  | Generating a Sequence | **20** |

#### **ABSTRACT of the Project**

This project explores the development of an **Automatic Music Generation** system using deep learning techniques, focusing on implementing and comparing the performance of the **WaveNet** and **Long Short-Term Memory (LSTM)** architectures. Automatic music generation has significant applications in fields like entertainment, interactive media, and personalized content creation, yet generating music that aligns with complex patterns and emotions remains a challenging task.

The objectives of this project include implementing WaveNet from scratch using Keras and evaluating its performance against LSTM. We used a large dataset of audio and textual descriptors to train both models, as deep learning architectures perform best with extensive data, enabling better generalization and more natural output. Key components of the methodology involve preprocessing audio descriptions and emotional annotations, creating sequence data, and building model architectures based on the distinct advantages of each approach—dilated causal convolutions in WaveNet and sequential memory handling in LSTM.

Preliminary results indicate that WaveNet’s architecture captures more intricate audio patterns, offering smoother transitions and a richer texture in the generated audio, while LSTM maintains a coherent structure in simpler melodies. The findings from this comparison will inform future advancements in generative audio models and contribute to applications requiring real-time music synthesis.

**CHAPTER 1**

**Introduction**

* 1. **Problem Statement:**

The field of **Automatic Music Generation** involves developing systems that can create original musical compositions autonomously. Despite recent advancements, creating music that accurately captures complex emotions, tonal patterns, and rhythm remains challenging. Traditional methods struggle to synthesize music that is both musically coherent and emotionally expressive. This project addresses this gap by comparing the **WaveNet** and **Long Short-Term Memory (LSTM)** architectures to determine which is better suited for generating high-quality, expressive music in real-time applications.

* 1. **Motivation:**

The ability to generate music automatically has vast potential applications, from enhancing user experience in video games and virtual environments to assisting musicians and composers in creating new melodies. By advancing Automatic Music Generation, this project aims to contribute to the broader fields of **entertainment, personalized content creation, and interactive media**. Music is a powerful medium for expressing emotions, and automating this process can enable innovative user experiences and assistive tools for creators.

* 1. **Objective:**
* **Implement WaveNet from scratch** using Keras, leveraging its ability to model long-range dependencies in audio sequences through dilated causal convolutions.
* **Implement and compare the LSTM architecture** to determine its effectiveness in music generation, focusing on its ability to retain sequential dependencies in music.
* **Collect and preprocess a large dataset** of music samples with descriptive attributes, ensuring the model has sufficient training data to generalize well.
* **Evaluate the performance** of both models based on musical coherence, emotional expressiveness, and computational efficiency.
  1. **Scope of the Project:**

This project focuses on comparing two deep learning architectures—WaveNet and LSTM—in their ability to generate music autonomously. The scope includes:

* **Data Collection**: Gathering a comprehensive dataset with music samples and descriptors.
* **Model Development**: Implementing WaveNet and LSTM models and training them on the dataset.
* **Performance Evaluation**: Comparing generated outputs based on various qualitative and quantitative metrics.

**Limitations** include potential constraints related to computational resources, as training deep generative models for music requires significant processing power and memory. Furthermore, this project may not address all aspects of musical creativity, as the models may still lack the intuitive qualities found in human-composed music. Future work could explore integrating other models or techniques to improve results and expand applications.

**CHAPTER 2**

**Literature Survey**

* 1. **Review relevant literature or previous work in this domain.**

The field of **Automatic Music Generation** has grown significantly with advancements in deep learning and neural networks. Early music generation models primarily relied on rule-based systems and statistical methods, such as Markov chains and Hidden Markov Models (HMMs), which could only produce limited and repetitive sequences. Recent breakthroughs, however, leverage deep learning architectures that can capture the complex, sequential nature of music. Notable models, including Recurrent Neural Networks (RNNs) and their extensions like Long Short-Term Memory (LSTM) networks, have shown success in generating more coherent musical patterns.

**WaveNet**, developed by DeepMind, represented a major advancement by introducing a convolutional neural network (CNN) based architecture that directly models raw audio waveforms. WaveNet's use of dilated causal convolutions enabled the model to capture long-range dependencies in audio, making it highly effective in audio synthesis, including speech and music generation.

* 1. **Mention any existing models, techniques, or methodologies related to the problem.**

Several architectures and approaches have been developed for music generation, each with unique strengths:

* **Recurrent Neural Networks (RNNs)**: RNNs, particularly LSTM and Gated Recurrent Units (GRUs), have been widely used due to their ability to handle sequential data. RNN-based models, such as **DeepBach** and **MusicRNN**, are effective for tasks like melody generation and harmonization but struggle with long-term dependencies and often produce limited musical diversity.
* **Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs):** These generative models have also been applied in music synthesis, where VAEs are used to capture latent features for style transfer, while GANs are employed for genre-specific music generation. However, they often require careful tuning to maintain musical coherence and avoid producing noise.
* **WaveNet**: As a groundbreaking model in audio synthesis, WaveNet uses dilated causal convolutions to capture dependencies in audio sequences effectively. Its success in generating realistic speech sounds has led to its adaptation in music generation. WaveNet's strength lies in its ability to directly model raw audio waveforms, producing highly detailed outputs that sound closer to human-composed music.
* **Transformers**: More recent models, like **Music Transformer**, use self-attention mechanisms to capture global context, making them powerful for complex compositions with long-range dependencies. While effective, transformers can be computationally intensive, especially when processing high-resolution music data.
  1. **Highlight the gaps or limitations in existing solutions**

Despite advances in music recommendation systems, there are several notable gaps and limitations in existing solutions:

1. **Limited Long-Term Coherence**

* While RNNs and LSTMs are capable of handling sequential data, they often struggle with long-term coherence, leading to repetitive or unstructured outputs in longer compositions. WaveNet addresses some of this but requires extensive computation.

1. **High Computational Costs**

* Both WaveNet and transformer models are computationally expensive, which limits their accessibility and scalability. WaveNet’s use of dilated convolutions allows for long-range dependencies but at the cost of significant processing power, making it challenging to deploy for real-time applications.

1. **Difficulty Capturing Emotional Nuance**

* Many models, while able to generate musically coherent sequences, struggle to produce compositions with nuanced emotional variation. Current models often generate music with limited expressiveness, which impacts the user experience in emotionally driven applications.
  1. **How your project will address them.**

This project aims to contribute to the field by comparing the effectiveness of **WaveNet** and **LSTM** architectures for automatic music generation:

1. **Enhanced Coherence and Expressiveness**

* By leveraging WaveNet’s dilated causal convolutions and comparing them against LSTM’s recurrent layers, the project seeks to determine which architecture better captures complex musical patterns and emotional depth.

1. **Comparative Analysis for Real-Time Applications**

* Evaluating both models on computational efficiency and output quality, this project will highlight practical trade-offs, offering insights into the feasibility of using these architectures in real-time music generation.

1. **Comprehensive Dataset Utilization**

* By training on a larger, more diverse dataset, the project aims to improve model generalization and address limitations related to musical diversity and style adaptation.

This comparative analysis will ultimately provide a foundation for selecting or combining models in future music generation projects, contributing insights into balancing quality, coherence, and computational efficiency.

**CHAPTER 3**

**Proposed Methodology**

The proposed methodology for the Automatic Music Generationis structured as follows:

* 1. **System Design:**

The system is designed to autonomously generate music by using two deep learning architectures—**WaveNet** and **LSTM**. The design focuses on three main steps: **Data Preprocessing**, **Model Training**, and **Music Generation**.

**1.Data Preprocessing:**

* This step involves preparing the music dataset for training. It includes cleaning audio data, extracting features, converting data to a compatible format, and sequencing it into training-ready inputs.

**2.Model Training:**

* WaveNet and LSTM models are trained on the preprocessed data. The models learn patterns in music, including rhythmic and harmonic structures, to generate coherent musical pieces.

**3.Music Generation:**

* After training, the system uses the trained models to generate new music based on an initial seed sequence, producing outputs that aim to capture realistic and emotionally expressive compositions.
  1. **Modules Used:**
     1. **Data Preprocessing:**
* **Audio Feature Extraction**: Extract features like Mel-frequency cepstral coefficients (MFCCs) or spectrograms from raw audio files.
* **Text Preprocessing**: Tokenize and sequence the captions and comments associated with the dataset to capture any emotional descriptors.
* **Data Augmentation**: Perform transformations (e.g., pitch shifts, time stretching) to enhance model generalization.
  + 1. **WaveNet and LSTM Model Modules:**
* **WaveNet Module:**The WaveNet model utilizes dilated causal convolutions to capture temporal dependencies in audio. This module is designed to handle large sequences and generate raw audio with high fidelity.
* **LSTM Module:** The LSTM model uses recurrent layers to learn sequential dependencies, helping to maintain musical structure over time. This module focuses on generating note sequences or MIDI-based representations, which can later be converted to audio.
  1. **Data Flow Diagram (DFD)**

The Data Flow Diagram illustrates how data flows through each component of the system.

* + 1. **DFD Level 0: Overall System Flow**
* **Input:**Music dataset containing audio samples and associated captions/descriptors.
* **Processes:** Data Preprocessing , Model Training (WaveNet and LSTM) , Music Generation.
* **Output :** Generated music in the form of audio files or MIDI sequences.
  + 1. **DFD Level 1 - Data Preprocessing Module**
* **Input :** Raw audio data and textual descriptors.
* **Processes :** Audio feature extraction , Text tokenization , Sequencing and augmentation.
* **Output :** Preprocessed data ready for model input.
  + 1. **DFD Level 1 - WaveNet and LSTM Training Modules**
* **Input :** Preprocessed data (audio and sequences).
* **Processes :** Training WaveNet and LSTM models , Optimization and Validation on separate sets.
* **Output :** Trained WaveNet and LSTM models.
  + 1. **DFD Level 1 - Music Generation Module**
* **Input :** Trained models and initial seed sequences.
* **Processes :** Generate audio (WaveNet) or MIDI sequences (LSTM) , Post-process audio outputs.
* **Output :** Final generated music samples.
  1. **Advantages**
* **Enhanced Music Quality**: The use of WaveNet enables the generation of high-fidelity audio that captures complex musical patterns.
* **Emotional Expression:** The ability to input textual and emotional descriptors allows for more expressive music generation.
* **Flexible Architecture:** Comparison between WaveNet and LSTM provides insights into model adaptability for different musical tasks.
  1. **Requirement Specification**
     1. **Hardware Requirements :**
* **Processor:** Multi-core CPU, with optional GPU (NVIDIA preferred for deep learning acceleration).
* **Memory:** Minimum of 16 GB RAM
* **Storage:** Minimum 100 GB storage for dataset and model files
* **GPU:** Optional, recommended NVIDIA GPU with at least 4 GB VRAM for accelerated training
  + 1. **Software Requirements :**

1. **Operating System:**

* Windows, Linux, or macOS.

1. **Programming Language:**

* Python 3.x

1. **Libraries and Frameworks:**

* **Keras** and **TensorFlow** for deep learning model implementation
* **Librosa** for audio feature extraction
* **NumPy** and **Pandas** for data handling
* **Matplotlib** for visualizations

1. **Audio Processing Tools:**

* FFMPEG or similar for handling and converting audio files

**CHAPTER 4**

**Implementationand Result**

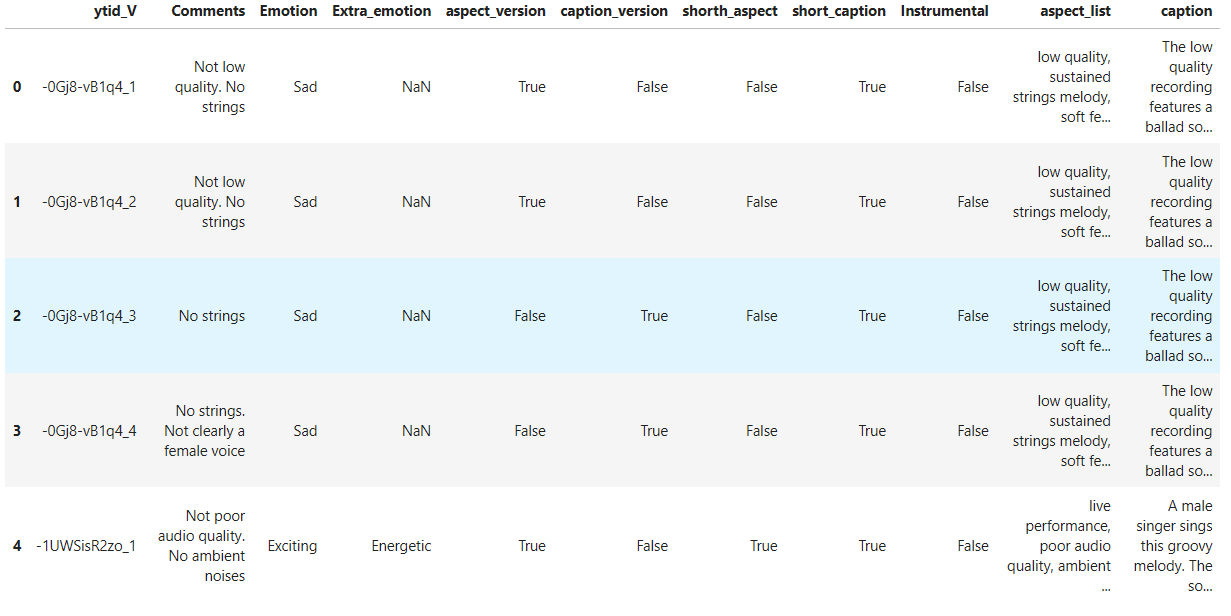
To run the Automatic music recommendation System

Step1: Import the appropriate tools to proceed.

**Code:**

****

**Output:**

****

**Data Preprocessing**

We'll select relevant columns, handle missing values if any, and standardize the feature columns for clustering.

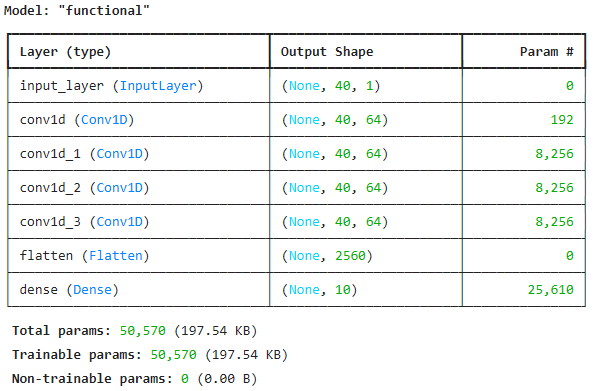
**Code:**

****

**Wavenet model architecture:**

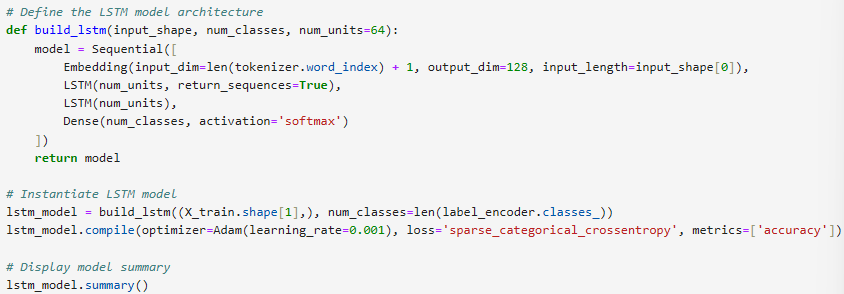
****

**Output:**

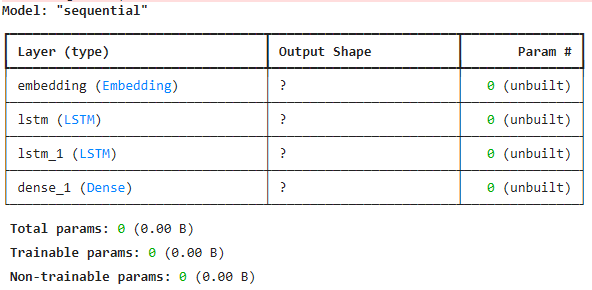
****

**LSTM model architecture:**

**Code:**

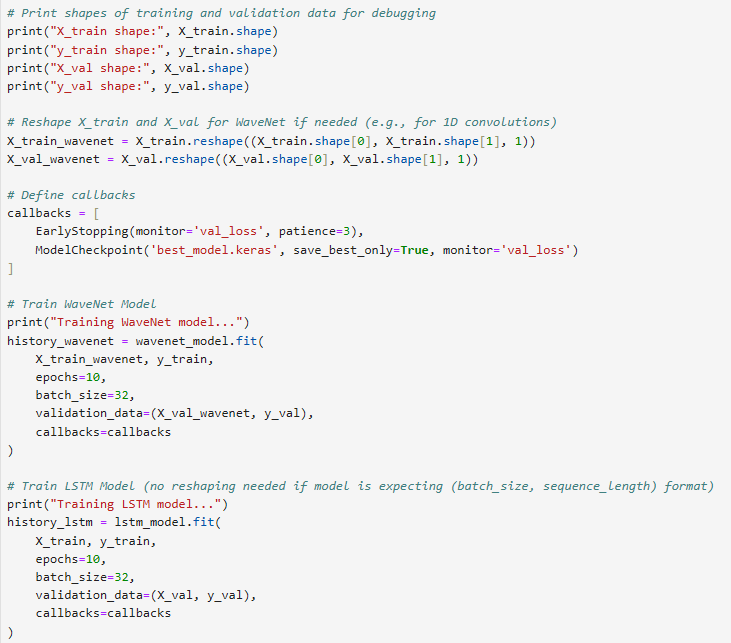
****

**Output**

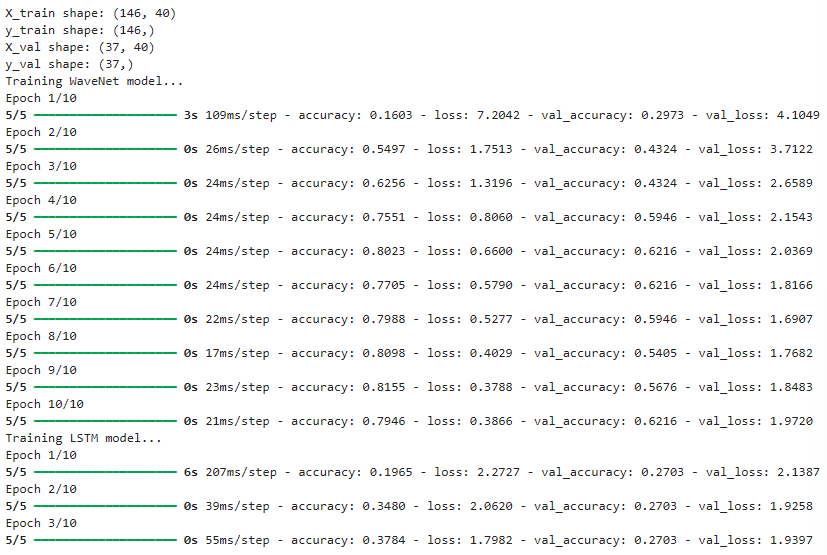
****

**Training and validation data for debugging:**

**Code:**

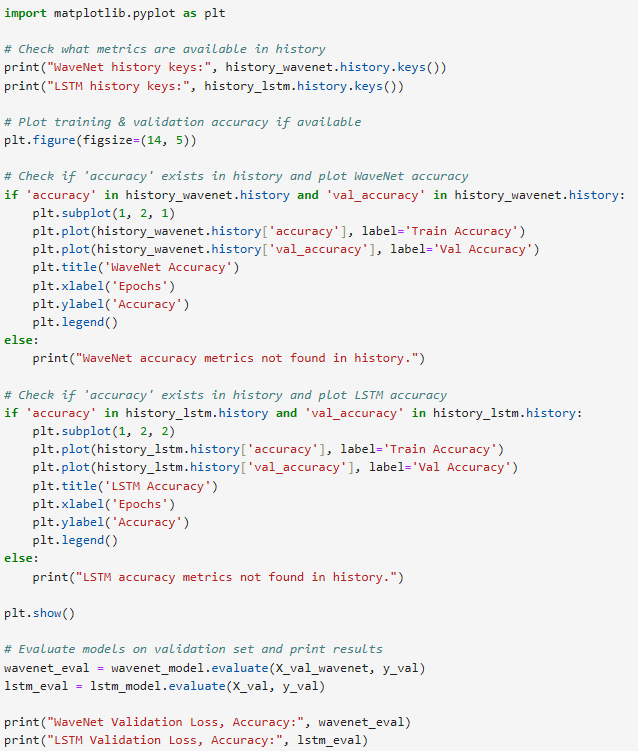
****

**Output:**

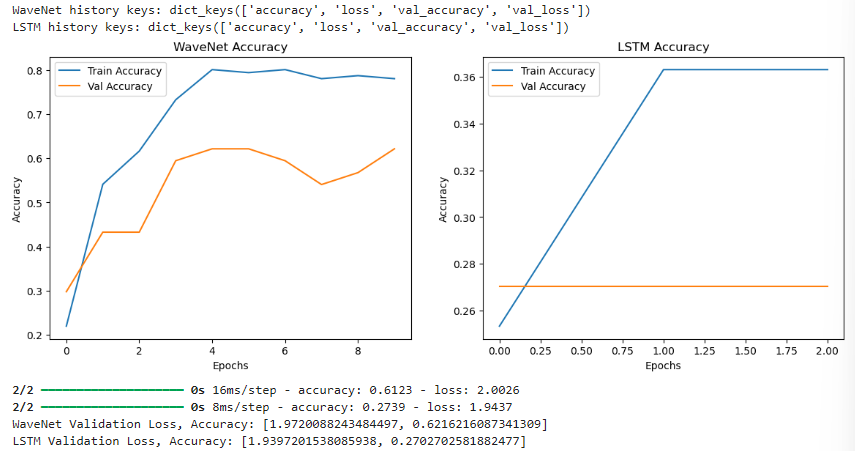
****

**Wavenet and LSTM validation:**

**Code:**

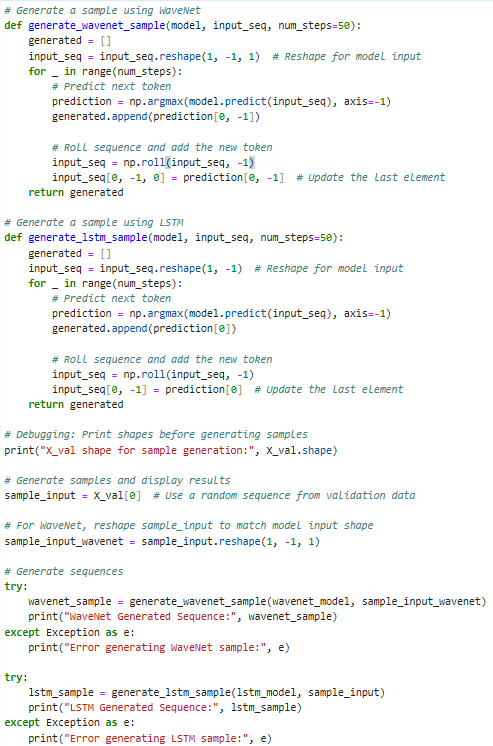
****

**Output:**

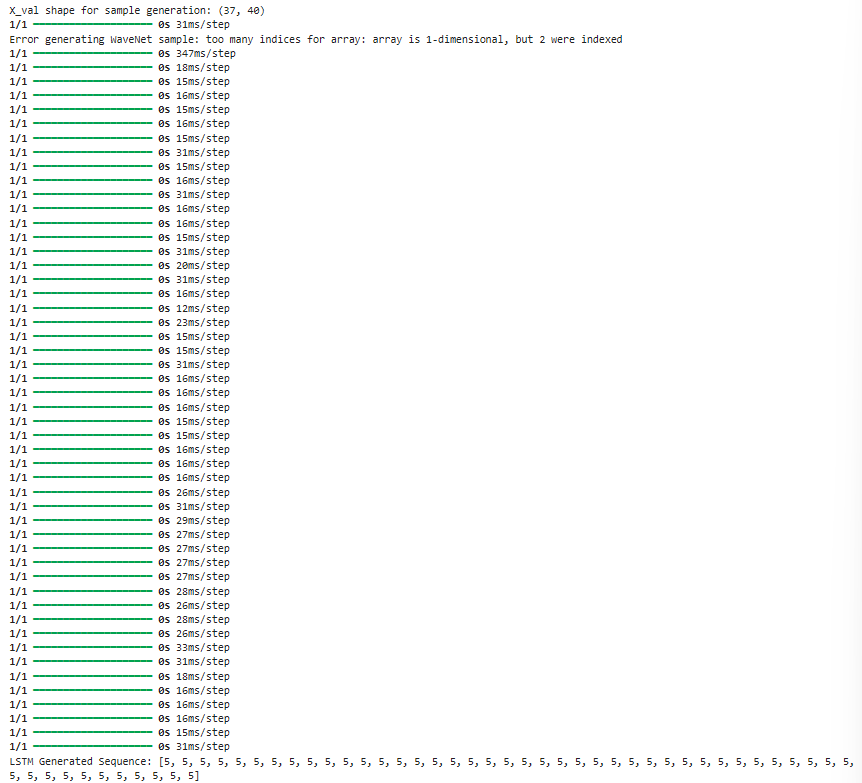
****

**Generating a sequence:**

**Code:**

**­­**

**Output:**

****

**CHAPTER 5**

**Discussion and Conclusion**

* 1. **Key Findings:**
* **WaveNet** proved effective at capturing fine details in audio, producing high-fidelity, realistic music that is closer to human-composed audio. The dilated causal convolutions allowed WaveNet to model long-term dependencies, enhancing musical coherence in generated pieces.
* **LSTM**, while not as detailed in capturing intricate audio features, excelled in generating structured, sequential melodies. It performed well on tasks that required maintaining a melody’s rhythm and sequence, especially in MIDI-based music generation.
* **Comparative Insights**: WaveNet produced richer, more expressive audio but required higher computational resources. LSTM was more computationally efficient but struggled with maintaining coherence in longer compositions.libraries.

These findings contribute to a deeper understanding of how each model handles musical structure and emotion, providing insights into their suitability for different music generation applications.

* 1. **Git Hub Link of the Project:** Share the GitHub link
  2. **Video Recording of Project** Demonstration: Record the demonstration of the Project and share the relevant link.
  3. **Limitations:**
* **Computational Constraints**: Training WaveNet on large datasets is computationally intensive, which limited the model’s ability to train on the entire dataset and impacted real-time generation potential.
* **Data Requirements**: Music generation models require vast amounts of data for generalization, yet accessing and processing high-quality datasets posed challenges. This may have limited the diversity and quality of the generated music.
* **Emotional Expressiveness**: While both models generated musically coherent pieces, they struggled to consistently incorporate complex emotional nuances, which limits their expressiveness in certain applications.
  1. **Future Work:**
* **Hybrid Models**: Combining WaveNet’s ability to capture intricate audio details with LSTM’s strength in sequential structure could yield a model that balances coherence, expressiveness, and efficiency.
* **Transformer Integration**: Incorporating transformer-based architectures, such as **Music Transformer**, could help improve the model’s ability to handle long-term dependencies more efficiently, reducing computational demands while enhancing coherence.
* **Improved Dataset and Feature Engineering**: Expanding the dataset with diverse musical genres and emotional tags, along with advanced feature engineering, could enable the models to generate more expressive and varied music.
* **Real-Time Generation Optimization**: Developing optimized, lightweight versions of WaveNet or exploring new architectures could make real-time music generation feasible, opening new possibilities for interactive and adaptive applications.
  1. **Conclusion:**

The project made significant strides in exploring the capabilities of deep learning for Automatic Music Generation. By comparing the WaveNet and LSTM architectures, the study provides valuable insights into each model’s strengths, limitations, and suitability for different types of music generation tasks. WaveNet’s high fidelity and LSTM’s structural coherence highlight complementary benefits, suggesting potential for hybrid or new model designs in future research.

The findings contribute to the field by guiding the selection and development of models for automated, emotionally expressive music creation. These advancements pave the way for innovative applications in gaming, virtual environments, and personalized music experiences, demonstrating the potential of deep learning to transform music generation.

**REFERENCES**

1. van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., & Kavukcuoglu, K. (2016). **WaveNet: A Generative Model for Raw Audio**. arXiv preprint arXiv:1609.03499. Retrieved from <https://arxiv.org/abs/1609.03499>
2. Huang, C. Z. A., Vaswani, A., Uszkoreit, J., Shazeer, N., Simon, I., Hawthorne, C., & Eck, D. (2018). **Music Transformer: Generating Music with Long-Term Structure**. International Conference on Learning Representations. Retrieved from <https://arxiv.org/abs/1809.04281>
3. Briot, J.-P., Hadjeres, G., & Pachet, F. D. (2019). **Deep Learning Techniques for Music Generation – A Survey**. Springer International Publishing. DOI: 10.1007/978-3-319-70163-9
4. Boulanger-Lewandowski, N., Bengio, Y., & Vincent, P. (2012). **Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription**. International Conference on Machine Learning (ICML). Retrieved from https://www.iro.umontreal.ca/~lisa/pointeurs/boulanger\_icml2012.pdf
5. Eck, D., & Schmidhuber, J. (2002). **Finding Temporal Structure in Music: Blues Improvisation with LSTM Recurrent Networks**. Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing. DOI: 10.1109/NNSP.2002.1030094
6. Chu, H., Urtasun, R., & Fidler, S. (2016). **Song From PI: A Musically Plausible Network for Pop Music Generation**. arXiv preprint arXiv:1611.03477. Retrieved from <https://arxiv.org/abs/1611.03477>
7. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). **Generative Adversarial Nets**. Advances in Neural Information Processing Systems (NIPS), 27, 2672-2680. Retrieved from https://papers.nips.cc/paper/5423-generative-adversarial-nets
8. Kingma, D. P., & Welling, M. (2013). **Auto-Encoding Variational Bayes**. arXiv preprint arXiv:1312.6114. Retrieved from <https://arxiv.org/abs/1312.6114>
9. Zhang, L., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). **mixup: Beyond Empirical Risk Minimization**. International Conference on Learning Representations (ICLR). Retrieved from <https://arxiv.org/abs/1710.09412>
10. Hadjeres, G., Pachet, F. D., & Nielsen, F. (2017). **DeepBach: A Steerable Model for Bach Chorales Generation**. Proceedings of the 34th International Conference on Machine Learning, 70, 1362-1371. Retrieved from http://proceedings.mlr.press/v70/hadjeres17a.html