RhythmGenius - Melody generation using RNN-LSTM

Submitted in partial fulfilment of the requirements of the degree

BACHELOR OF ENGINEERING IN COMPUTER ENGINEERING

By

Pratik Kithani - 2003094

Kishan Kokal - 2003095

Shubham Mandal - 2003105

Harsh Punjabi - 2003136

Supervisor

Prof. Anagha Durugkar

(Associate Professor, Department of Computer Engineering, TSEC)

Computer Engineering Department

Thadomal Shahani Engineering College

Bandra (W), Mumbai - 400 050

University of Mumbai

(AY 2023 - 24)

Certificate

This is to certify that the project entitled **"RhythmGenius - Melody generation using RNN-LSTM"** is a bonafide work of

2003094 Pratik Kithani

2003095 Kishan Kokal

2003105 Shubham Mandal

2003136 Harsh Punjabi

submitted to the University of Mumbai in partial fulfilment of the requirement for the awarded degree of **"Bachelor of Engineering"** in **"Computer Engineering".**

Prof. Anagha Durugkar
Guide

Dr. Tanuja Sarode
Head of Department

Dr. G. T. Thampi
Principal

Project Approval for BE

The Project entitled **"RhythmGenius - Melody generation using RNN-LSTM"** by

2003094 Pratik Kithani

2003095 Kishan Kokal

2003105 Shubham Mandal

2003136 Harsh Punjabi

is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

_____

(Internal Examiner Name & Sign)

_____

(External Examiner Name & Sign)

Date:
Place: Mumbai

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. we also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Pratik Kithani (2003094)

Kishan Kokal (2003095)

Shubham Mandal (2003105)

Harsh Punjabi (2003136)

Date:

Abstract

In the realm of music composition, this project aims to create a music generation system leveraging Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM). The objective is to train this system on a dataset of folk melodies and enable it to generate novel melodies that exhibit similarities to those found in the training data. This project seeks to delve into the intricacies of melody creation by treating it as a time-series prediction problem, predicting the progression of musical events in time. Through LSTM's capacity to capture long-term temporal dependencies, this project aims to replicate the structural patterns found in melodies, which often involve repeated patterns with variations in pitch content, stretching, or shrinking. The project's foundation rests on the ESAC dataset, comprising over 20,000 folk melodies from various parts of the world. The core tools and libraries employed in this project include Keras/Tensorflow for deep learning, Music21 for symbolic music data processing, and MuseScore for music notation. This collaborative effort involves team members Pratik Kithani, Kishan Kokal, Shubham Mandal, and Harsh Punjabi, under the guidance of mentor Dr. Anagha Durugkar from Thadomal Shahani Engineering College.

Contents

List of Figures

Chapter 1

Introduction

1.1 Introduction

Music composition is a profoundly artistic pursuit, where the harmonious interplay of musical elements such as notes, rests, and their intricate temporal relationships come together to craft captivating melodies. It is an ancient form of human expression, transcending cultural boundaries, and evoking emotions that words alone cannot capture. In this project, we embark on a creative journey into the realm of music composition, wielding the power of deep learning techniques, particularly Recurrent Neural Networks with Long Short-Term Memory (RNN-LSTM) architecture. Our primary objective is to harness the potential of these advanced algorithms to generate music compositions, drawing inspiration from a rich dataset of folk melodies.

Folk music, with its roots deeply embedded in the collective heritage of diverse cultures, offers a remarkable tapestry of melodies and rhythms. Its distinct patterns and unique structural elements have been cherished and passed down through generations. By delving into the realm of folk melodies, our endeavour is to not only appreciate the cultural significance of this musical genre but also to contribute to its evolution. Our ultimate goal is to develop a sophisticated system, driven by artificial intelligence, which possesses the capability to autonomously compose melodies that bear a striking resemblance to the structural nuances found in folk music.

As we embark on this musical exploration, we'll delve into the intricacies of RNN-LSTM, a powerful neural network architecture that excels in capturing sequential dependencies within data. With this tool at our disposal, we aim to extract the essence of folk music and infuse it into our compositions, thereby bridging the gap between traditional artistry and cutting-edge technology. In doing so, we aspire to extend the boundaries of creativity, enrich the world of music, and offer a fresh perspective on the potential of artificial intelligence in the realm of the arts. This project serves as a testament

to the harmonious synergy between human creativity and machine intelligence, opening up new horizons for musical expression.

1.2 Aim and Objectives

The primary aim of this project is to leverage deep learning techniques, specifically the RNN-LSTM architecture, to create a system capable of autonomously generating music compositions inspired by folk melodies. This project seeks to:

1. Explore Folk Music Heritage: Investigate the rich and diverse world of folk melodies from various cultures to gain a deeper understanding of their structural patterns, rhythms, and melodic elements.

2. Develop an AI-Driven Music Composer: Design and implement a robust and creative AI system that can generate original melodies, drawing from the stylistic and structural characteristics of folk music.

3. Preserve and Evolve Tradition: Aim to preserve the cultural significance of folk music by infusing it with fresh compositions, thus contributing to the evolution of this musical genre while respecting its historical roots.

4. Capture Sequential Dependencies: Utilize the capabilities of RNN-LSTM to accurately capture and reproduce the intricate temporal relationships within musical sequences, resulting in compositions that exhibit coherence and familiarity.

5. Demonstrate the Synergy of Art and Technology: Showcase the potential of artificial intelligence as a complementary tool for artists and composers, bridging the gap between traditional artistry and modern technology.

6. Enrich Musical Expression: Envision the project as a means to enrich the world of music by offering novel, inspiring compositions that serve as a testament to the limitless possibilities when creativity and AI converge.

Ultimately, this project aims to provide a valuable platform for both musicians and AI enthusiasts, demonstrating the power of technology to extend the boundaries of creativity while paying homage to the timeless beauty of folk music traditions.

1.3 Scope

The scope of this project encompasses a comprehensive exploration of the application of deep learning techniques, specifically the RNN-LSTM architecture, in the realm of music composition, with a specific focus on folk melodies. The project scope includes, but is not limited to, the following key elements:

1. Data Collection and Analysis: Gathering a diverse and extensive dataset of folk melodies from various cultural backgrounds and analyzing them to identify common structural patterns, rhythms, and melodic elements.

2. Algorithm Development: Designing, implementing, and fine-tuning a deep learning model based on the RNN-LSTM architecture for music composition. This model should have the ability to understand and generate music sequences that align with the identified characteristics of folk melodies.

3. Training and Validation: The project involves the training of the RNN-LSTM model using the acquired folk music dataset and ensuring that the generated compositions exhibit the desired folk music attributes. Validation will be carried out through various metrics and human evaluation.

4. Musical Output: Generating original music compositions autonomously using the developed AI system. These compositions should not only showcase the characteristic elements of folk music but also offer variations and creativity in line with the project's objectives.

5. User Interface: Depending on the project's complexity, the development of a user-friendly interface for musicians and composers to interact with the AI system for music generation can be considered.

6. Cultural Sensitivity: Ensuring that the project respects and appreciates the cultural diversity of folk music, adhering to ethical considerations when using cultural material in the training dataset.

7. Documentation and Evaluation: Comprehensive documentation of the project's methodology, development, and outcomes, accompanied by a thorough evaluation of the AI-generated compositions for their alignment with the folk music attributes.

8. Future Extensions: Exploring possibilities for future extensions, such as integrating additional AI models, expanding the dataset, or incorporating user feedback to further enhance the AI composer's capabilities.

9. Dissemination: Sharing the project's findings and compositions with the wider community, potentially through music performances, academic papers, or online platforms.

Chapter 2

Review of Literature

2.1 Domain Explanation

Basics of Melody and Musical Sequences:

A melody, in the context of music composition, is a captivating sequence of musical events that encompass both notes and rests. These events are integral to the composition, contributing to the overall rhythm and emotional resonance of the music. When we visualize a melody, we often represent it graphically, with pitch (frequency) depicted on the vertical, or y-axis. The position along this axis is indicative of the pitch's elevation, with higher positions corresponding to higher pitch values. On the horizontal, or x-axis, we delineate time, essentially charting the progression of musical events.

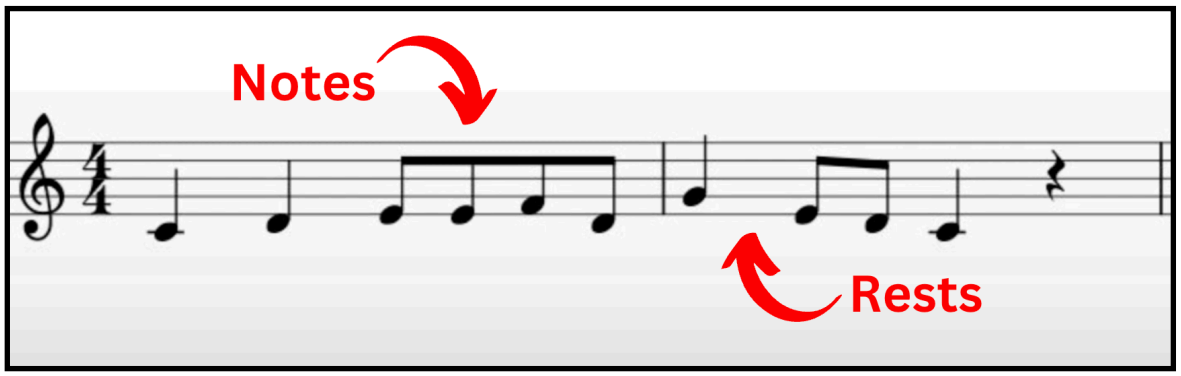

Figure 2.1: Visualisation of Musical Events in a Melodic Sequence

In essence, when we analyze a melody, we are meticulously tracking the evolution of pitch values over time, recording when specific notes or rests occur. Consequently, this perspective allows us to perceive a melody as a temporal sequence, akin to a narrative unfolding in time. This perspective is essential for the purposes of our project, where we endeavor to create AI-driven compositions that encapsulate the essence of melodies.

As we delve deeper into this exploration, we encounter the notion of a time series. In data science and analysis, a time series is a specific data structure that aligns seamlessly with our melodic interpretation. It involves the collection of samples at equally spaced

intervals in time. In the case of melody, this translates into capturing the musical events at distinct, evenly distributed moments throughout the composition. Hence, the process of melody generation can be aptly framed as a time-series prediction problem, where our primary aim is to forecast the upcoming musical events, or in other words, predict the next sample within this evolving time series.

However, before embarking on the journey of generating melodies, we face the preliminary task of reducing the complexity of musical notation. To do this, we must distill all the various musical events into a standardized vocabulary. This vocabulary should include accepted elements such as pitches, which represent distinct musical notes, and rests, which signify moments of silence or musical pause. By undertaking this reduction, we streamline the composition process, allowing for a more coherent and systematic approach to generating melodies. It also paves the way for the AI system to understand and work with these fundamental building blocks, facilitating the creation of harmonious and stylistically consistent compositions. In essence, the simplification of the musical language into a limited set of elements sets the stage for the subsequent AI-driven melody generation process.

2.1 Review of Existing Systems

In the pursuit of developing an AI-driven system for the generation of music compositions, particularly focusing on folk melodies, it is imperative to survey and assess the landscape of existing systems and technologies. The review of existing systems not only offers valuable insights but also provides a foundation for positioning our project within the broader context of AI-generated music and melody composition. This section discusses and evaluates some of the notable existing systems and approaches in the field:

Magenta Project by Google:

Description: Google's Magenta Project is a notable initiative in AI-generated music. It encompasses a range of tools and models, including LSTM-based recurrent neural

networks, to compose music. Magenta has successfully generated various music genres, but its focus on folk music is limited.

Strengths: Well-established, open-source platform with extensive resources.

Limitations: May require customization to cater specifically to folk melodies.

OpenAI's MuseNet:

Description: MuseNet is an AI music generation platform developed by OpenAI. It employs deep learning techniques to compose music across multiple genres and styles, including classical and contemporary music. However, its specialization in folk music is relatively unexplored.

Strengths: Versatile, capable of generating music in multiple styles.

Limitations: Lack of dedicated support for folk music and cultural nuances.

DeepBach:

Description: DeepBach is an AI system that specializes in generating Baroque-style music. It utilizes deep learning techniques and generative models to create intricate compositions in this genre.

Strengths: Highly specialized for a specific musical genre.

Limitations: Limited adaptability to folk music, which often follows different structural patterns.

Amper Music:

Description: Amper Music is a commercial AI music composition platform. It focuses on producing music for media, including film and advertising. While versatile, its primary emphasis is not on folk music.

Strengths: User-friendly and commercially available.

Limitations: Limited focus on folk music and potential cost considerations.

AI-Generated Folk Music Research:

Description: Several research projects have explored AI-generated folk music. However, these are often experimental and not as widely accessible as commercial platforms.

Strengths: Tailored to folk music, capturing cultural nuances.

Limitations: Limited availability and may lack user-friendly interfaces.

2.3 Limitations of Existing System

Existing systems and projects in the field of AI-generated music composition offer a range of capabilities and specializations, but there is a gap when it comes to dedicated systems for folk music. Our project aims to address this gap by not only utilizing deep learning techniques, like the systems mentioned above but also by focusing on the cultural nuances and structural intricacies specific to folk melodies. It also seeks to create a system that is both accessible and adaptable for musicians and composers interested in folk music, thereby contributing to the enrichment and preservation of this diverse musical tradition.

Chapter 3

Proposed System

3.1 Analysis

Functional Requirements:

1. Pre-processing:

- It should perform data pre-processing tasks, including cleaning, format conversion, vocabulary creation, and sequence alignment to ensure data uniformity.

2. Model Training:

- The system should train a Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM) architecture using the pre-processed folk melody dataset.

- The model should learn to recognize the temporal dependencies, pitch variations, and structural patterns within folk melodies.

3. Melody Generation:

- The system must be able to generate new and original folk melodies.

- It should take a seed melody as input and provide predictions for subsequent notes and rests in the sequence.

4. Cultural and Stylistic Sensitivity:

- The system should be designed to respect and reflect the cultural and stylistic nuances of folk music.

- It should generate melodies that align with the characteristics of the training dataset, including pitch content, rhythms, and structural patterns specific to folk melodies.

5. User Interface:

- The system should provide a user-friendly means for musicians and composers to interact with the system.

6. Integration with Music21 and MuseScore:

- If integrated, the system should work seamlessly with the Music21 Python library for processing symbolic music data.

- It may also allow for the translation of AI-generated melodies into traditional musical notation using MuseScore.

Non-Functional Requirements:

1. Performance:

- The system must exhibit robust and efficient performance, ensuring that melody generation is timely and responsive.

- It should handle large datasets and complex music generation tasks without significant delays.

2. Scalability:

- The system should be designed with scalability in mind, allowing for future enhancements, such as the incorporation of additional AI models or expansion of the dataset.

3. Accuracy:

- The model's accuracy in generating melodies must be a focal point, ensuring that generated melodies closely resemble the style and structure of folk melodies from the training dataset.

4. Cultural Sensitivity:

- The system should prioritize the preservation and respectful representation of the cultural significance of folk music in generated melodies.

5. Security and Data Privacy:

- Any personal or sensitive data used in the project, including user information, must be protected and handled in compliance with data privacy regulations.

6. Usability and Accessibility:

- If a user interface is implemented, it should be user-friendly and accessible to musicians and composers of varying technical backgrounds.

7. Documentation and Reporting:

- Comprehensive documentation should be provided, outlining the system's architecture, functionality, and usage.

- The project report should include detailed descriptions of the model's training process, results, and validation methods.

8. Collaboration and Team Dynamics:

- Effective communication and collaboration among team members, as well as with the project mentor, are essential for the successful completion of the project.

9. Adaptability and Maintenance:

- The system should be designed to adapt to changing requirements and updates in deep learning libraries or other tools.

- Maintenance procedures and plans for updates should be established.

10. Ethical Considerations:

- Ethical guidelines, such as respecting copyright and cultural sensitivity, should be upheld throughout the project.

- Ethical practices should be documented and adhered to.

11. Resource Management:

- Efficient utilization of computational resources, including memory and processing power, is essential for the system's stability and performance.

By addressing these functional and non-functional requirements, the proposed music generation system will aim to create a harmonious blend of technology and artistic expression, preserving the cultural heritage of folk music while pushing the boundaries of AI in the domain of music composition.

3.2 Design Details

The Music Generator (Training):

During the training phase, the neural network is exposed to the folk melody dataset to learn the intricate patterns and structures present in the music. The training process unfolds as follows:

Data Chunking: To facilitate learning, the folk melodies are divided into manageable data chunks. These chunks represent coherent segments of music and serve as the input to the neural network.

LSTM Model Training: The core of the system, the LSTM (Long Short-Term Memory) model, is trained on these data chunks. The model is designed to understand and internalize the temporal dependencies and nuances within the melodies. It learns to predict the subsequent notes or rests in the music based on the patterns it discerns during this phase.

Pattern Recognition: As the model is exposed to the training data, it gradually recognizes the recurring musical patterns and variations that define the folk melodies. These patterns include melodic sequences, rhythms, and pitch content, which are crucial for creating harmonious compositions.

Parameter Optimization: Throughout the training process, various model parameters, such as the number of layers, units per layer, and learning rates, are fine-tuned to ensure optimal performance. This parameter optimization aims to maximize the model's ability to generate melodies that closely adhere to the folk music genre.

The Music Generator (Inference):

The inference phase represents the system's creative ability to generate new melodies inspired by the training data. This phase functions as follows:

Seed Melody Initialization: The process begins with the selection of a seed melody. This seed melody serves as the starting point for generating a new composition. Users or composers can input a specific seed melody, defining the initial character of the music to be generated.

Predictive Model Output: The LSTM model, which has learned from the training data, starts to provide predictions. These predictions are the model's educated guesses regarding the next set of notes in the music sequence, given the seed melody.

Melody Expansion: The predicted notes are appended to the seed melody, creating an extended sequence. This extended sequence is then re-introduced to the model for another round of prediction. This iterative process continues, with each iteration building upon the previous one.

Progressive Melody Generation: As the system proceeds through iterations, it generates a complete composition step by step. Each iteration augments the melody further, and the model's predictions lead to the formation of the entire piece.

Fig 3.1: The music generator (training)

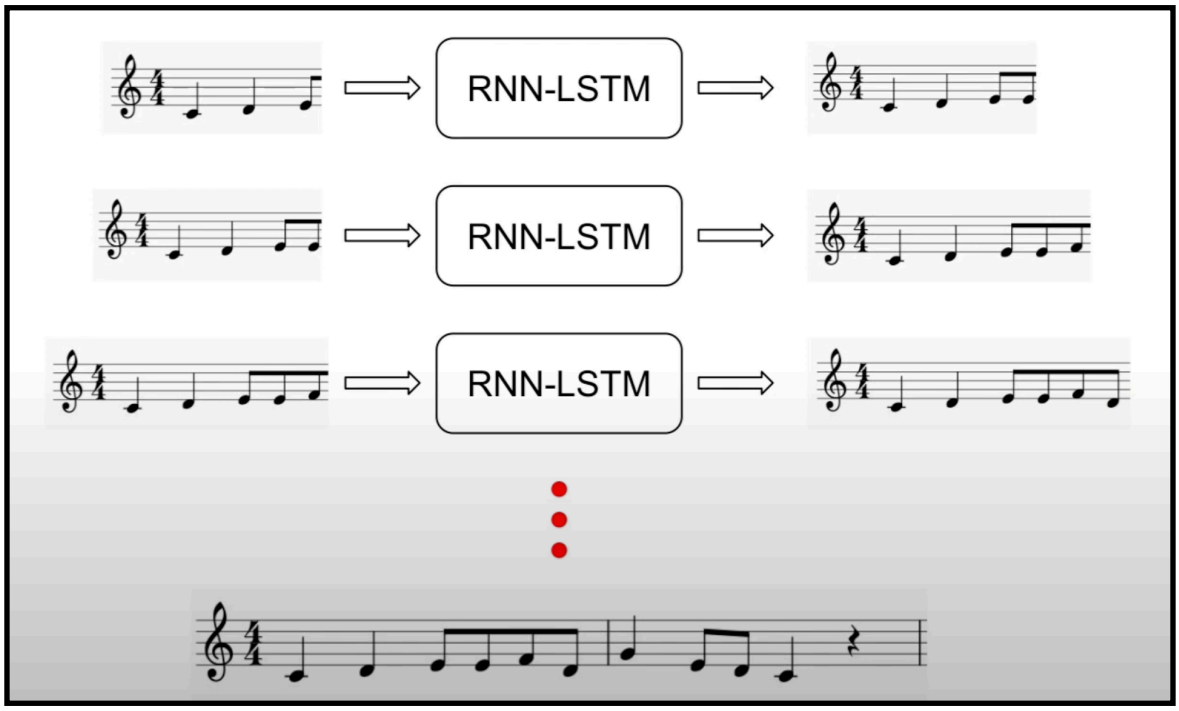

Fig 3.2: The music generator (inference)

By presenting the training and inference phases in this detailed manner, you provide readers with a comprehensive understanding of how the system learns from the training data and leverages that knowledge to generate coherent and culturally resonant folk melodies during the inference phase.

Proposed System:

Here's an explanation of the flow of a user interacting with the music generation system, which includes inputting a seed melody, converting it to time series data, using the model to generate melodies, and finally obtaining the generated music in MIDI format:

1. Input Seed Melody:

The user initiates the music generation process by inputting a seed melody into the system. This seed melody serves as the starting point for the AI to generate a new composition based on the user's input. The seed melody can be created directly within the application or uploaded from an existing MIDI file.

2. Convert Seed Melody to Time Series Data using Music21:

The user's seed melody, in musical notation or MIDI format, is processed and converted into time series data using the Music21 Python library. This conversion involves encoding the musical events, notes, rests, and their temporal positions in a structured format suitable for input to the AI model.

3. Input Time Series Data to the Model:

The processed time series data, representing the user's seed melody, is fed into the AI model. The model, based on its training with a dataset of folk melodies, interprets the input data and predicts the next sequence of notes and rests to extend the composition.

4. Model Recurrently Predicts the Next Melody:

The AI model begins generating the next part of the melody, iteratively predicting and appending new musical events to the input data. This process is recurrent and continues for multiple iterations, allowing the melody to evolve and expand as more predictions are made.

5. Convert the Time Series Data Back to MIDI using Music21:

The generated music, now in the form of time series data, is converted back into a MIDI format using the Music21 library. This conversion includes translating the structured data into a MIDI file that can be understood by music notation software, playback applications, or MIDI-compatible instruments.

6. Output File is Generated in MIDI:

The final output of the AI-generated composition is presented to the user in MIDI format. The user can download or access the MIDI file, which encapsulates the entire composition that has been generated by the AI system, starting from the user's seed melody.

This flow empowers the user to actively participate in the music generation process by providing a starting point (the seed melody) and witnessing the AI's creative contributions as it generates new melodies based on the input. The ability to convert between time series data and MIDI format ensures compatibility with a wide range of music software and playback devices, allowing the user to further explore and utilize the generated compositions.

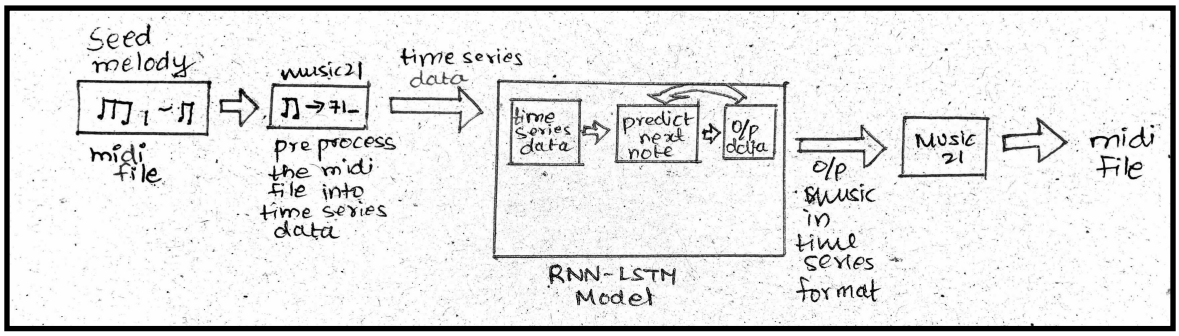

Fig 3.3: Flow Diagram

Chapter 4

Implementation Details

4.1 Experimental Setup

Pre-processing

1. Filtering Non-Acceptable Durations:

The preprocessing begins with filtering out songs that contain non-acceptable durations. These non-acceptable durations range from a sixteenth note value to a quarter note value, along with intermediate durations such as eighth notes, dotted notes, dotted quarter notes, half notes, and whole notes. This step simplifies the dataset and ensures that all musical events adhere to specific duration guidelines. To achieve this, attributes provided by the Music21 library are utilized to analyze each note and rest within the songs.

2. Transposition to a Common Key:

The next crucial preprocessing step involves transposing the songs to a common key, such as C major or A minor, depending on the original key of each song. This transposition process reduces the variability in keys across the dataset, simplifying the model's learning process and reducing the amount of data required for training. The transposition is achieved by finding the interval between the tonic note of the current key and the desired key, and then applying this interval to transpose the entire song.

3. Using Music21 for Data Processing:

The Music21 package is instrumental in the preprocessing workflow. It enables symbolic music analysis and conversion, and its capabilities include estimating the key of a piece of music, representing music in an object-oriented manner, and converting between different music formats. In the context of this project, Music21 is used for various tasks, such as estimating the key of the songs, extracting information required for transposition, and encoding the songs with Music21 methods and objects.

4. Encoding Songs in Time Series Representation:

As part of our preprocessing efforts, we established an "encoded song list" to efficiently store notes and rests in a time series format. To achieve this, we developed a dedicated function called "encode song." This function takes a song in Music21 object format and transforms it into a string, effectively encoding the song in a time series representation.

In the encoding process, we employed a for loop to distinguish between notes and rests within the song. For notes, we recorded their pitch and duration within the time series representation. Rests were processed in a similar manner, separately from the notes.

5. Converting the Encoded Song to a String and Saving to Text Files:

Once the songs were encoded into time series format, we converted them into string representations. We utilized the "map" function to cast the items in the encoded song as strings, facilitating further processing of the data.

We also documented the step of saving the converted encoded songs to text files. To do this, we created a structured directory within our project's file system. Each song in the dataset received a unique file name and was stored in the designated directory for ease of access and reference.

6. Creating Unique Identifiers for Songs:

To enhance dataset organization and reference, we implemented a system for generating unique identifiers for each song. These identifiers serve as a means of distinguishing and locating individual songs within the dataset. Our approach involved utilizing an index derived from a list of songs and converting it into a string format. These unique identifiers were then combined with other strings for effective song management and reference.

Each unique identifier was stored as a text file in the dataset directory to ensure a systematic and organized dataset structure.

7. Merging and Separating Multiple Songs:

We recognized the importance of both merging and separating multiple songs within the dataset. This practice streamlined dataset management and made it more manageable for various operations, including training a machine learning model like "Alice-chan," which specializes in generating melodies. The consolidation of songs into a single dataset simplified model training and analysis.

4.2 Software and Hardware Setup

Software and Hardware Setup

In the pursuit of developing our generative music project using a folk song dataset, it was imperative to establish a robust software and hardware infrastructure. This section provides an overview of the key components and tools utilized in our project setup.

Hardware Setup

Our project harnessed the computational power of a dedicated hardware configuration, which included:

Computer: We utilized a high-performance computer with advanced processing capabilities to support the computational demands of deep learning tasks.

Graphics Processing Units (GPUs): To expedite the training of our neural network, we leveraged powerful GPUs, which significantly accelerated the model training process. These GPUs are particularly valuable for handling the complex computations associated with deep learning.

Memory: The hardware setup featured ample system memory to accommodate the data processing and model training requirements of our project.

Software Setup

The software environment was carefully tailored to ensure seamless development, training, and evaluation of our generative music system. The essential components of our software setup included:

Operating System: Our project was executed on a well-configured operating system, providing a stable foundation for software and hardware interactions.

Programming Languages: We primarily employed Python, a versatile programming language, for implementing our project. Python's rich ecosystem of libraries and tools, along with its suitability for deep learning, made it the language of choice.

Version Control: To maintain code integrity and facilitate collaboration among team members, we employed Git, a widely-used version control system.

Deep Learning Frameworks and Libraries: Our project leveraged the following deep learning frameworks and libraries:

Keras/TensorFlow: We relied on Keras, an open-source high-level neural networks API, integrated with TensorFlow, for building and training our RNN-LSTM model. TensorFlow provided the computational backend for efficient model training.

Music21: A powerful Python library designed for symbolic music analysis and manipulation, Music21 played a pivotal role in the preprocessing of our dataset. It enabled tasks such as estimating the key of musical pieces and representing music in an object-oriented manner.

MuseScore: For music notation and rendering, we utilized MuseScore. This software allowed us to visualize and verify the correctness of the processed musical data.

The hardware and software setup described here was meticulously crafted to meet the project's computational and data processing needs. This infrastructure formed the backbone of our generative music system, enabling the training and generation of melodies based on the folk song dataset. The use of GPUs, powerful deep learning frameworks, and specialized music processing libraries contributed to the success of our project.

Chapter 5

Implementation

5.1 Plan for implementation

The successful development of a Melody Generation system requires a well-structured and coordinated effort among various teams and individuals. This section outlines the allocation of tasks and key milestones for our project.

5.1.1 Task Allocation

Our project involves distinct teams, each assigned specific tasks to contribute to the Melody Generation system. The task allocation is as follows:

Development Team:

- Develop and implement the RNN-LSTM model for melody generation.
- Fine-tune the model for improved melody composition.
- Create the software interface for user interaction with the melody generation system.

Data Acquisition and Preprocessing Team:

- Gather and preprocess the dataset of folk melodies used for training and evaluation.
- Transform the musical data into a format suitable for input into the RNN-LSTM model.
- Perform data augmentation and quality checks to enhance the dataset's effectiveness.

User Interface and Experience Team:

- Design and develop the user interface for the melody generation system.
- Ensure a user-friendly and intuitive experience for inputting seed melodies and receiving generated melodies.
- Implement functionality for users to customize and control the generation process.

5.1.2 Milestones

The project's implementation is structured around specific milestones to guide our progress and ensure a systematic approach to achieving our objectives. The milestones are defined as follows:

Milestone 1 (End of Phase 1 - November 30, 2023):

- Complete the development and implementation of the RNN-LSTM model for melody generation.
- Gather and preprocess the dataset of folk melodies required for training and evaluation.

Milestone 2 (End of Phase 2 - December 31, 2023):

- Design and initiate the development of the user interface for the melody generation system.
- Fine-tune the RNN-LSTM model for improved melody composition, focusing on capturing the structural patterns of folk melodies.
- Continue the development of the user interface, ensuring a smooth and user-friendly experience.

Milestone 3 (End of Phase 3 - March 31, 2024):

- Complete the development of the user interface, allowing users to input seed melodies and receive generated melodies.
- Perform extensive testing of the RNN-LSTM model and the user interface to ensure functionality and stability.
- Compile the project findings, including results, insights, and observations, within the project report for presentation.

These milestones serve as crucial checkpoints in our project timeline, enabling us to monitor progress, make adjustments, and ultimately achieve our goal of creating a Melody Generation system that can produce melodies with structural patterns similar to folk music.

5.2 Timeline chart for Term 1 and Term 2

Task	Duration	Start Date	End Date
Project Kick-off: • Brainstorming Ideas • Exploring research papers on brainstormed ideas • Exploring existing systems • Exploring limitations of the existing systems • Designing the architecture to overcome those limitations	1.5 months	July 2023	August 2023
Phase 1: • Gathering and preprocessing of the dataset of folk melodies required for training and evaluation.	2 months	August 2023	November 2023
Phase 2: • Design and initiation of the development of the user interface for the melody generation system. • Fine-tuning of the RNN-LSTM model for improved melody composition, focusing on capturing the structural patterns of folk melodies. • Continue the development of the user interface, ensuring a smooth and user-friendly experience.	1 month	November 2023	December 2023

Task	Duration	Start Date	End Date
Phase 3: • **Completion of the development of the user interface, allowing users to input seed melodies and receive generated melodies.** • **Extensive testing of the RNN-LSTM model and the user interface to ensure functionality and stability.**	3 months	January 2024	March 2024

Figure 5.1 Timeline Details

This timeline outlines the key milestones and their respective timeframes, allowing for a clear overview of the project's progress over the specified period. Please adjust the dates and descriptions as needed to align with your project's schedule and tasks.

Chapter 6

Conclusion

In embarking on the journey of Melody Generation, we have taken the initial steps towards our ultimate goal. While our project remains a work in progress, the completion of the preprocessing phase marks a significant milestone on our path to creating a Melody Generation system that encapsulates the essence of folk melodies. This section reflects on our achievements, the challenges we've encountered, and the promising road ahead.

6.1 Preprocessing Milestone

The successful preprocessing of the folk melody dataset has set the foundation for the subsequent stages of our project. It involved the meticulous gathering and preparation of the data, ensuring that it is ready to be ingested by our RNN-LSTM model. Data augmentation and quality checks were implemented to enhance the dataset's robustness, enabling it to capture the rich and diverse spectrum of folk melodies from around the world.

6.2 Challenges and Learnings

Our journey has not been without its challenges. Preprocessing a musical dataset, particularly one as culturally diverse as folk melodies, presented unique complexities. Ensuring the dataset's integrity and quality demanded attention to detail and a deep understanding of musical structures. However, these challenges have been invaluable, providing us with opportunities to learn and adapt as we progress.

6.3 The Road Ahead

The completion of the preprocessing phase signifies the transition to the next stages of our project. In the coming phases, we will develop and fine-tune the RNN-LSTM model for melody generation, with a particular focus on capturing the intricate structural patterns found in folk music. Simultaneously, the user interface development will proceed, ensuring a seamless and user-friendly experience for both inputting seed melodies and receiving generated compositions.

6.4 Collaborative Effort

Our project is a testament to the power of collaboration. The diverse teams involved, including development, data acquisition and preprocessing, user interface design, and quality assurance, have worked together seamlessly to advance the project. It is the collective effort and expertise of each team member that propels us forward.

6.5 Looking Ahead

While we pause to reflect on our achievements thus far, it is important to keep our eyes on the horizon. The completion of the preprocessing phase signifies the start of an exciting and challenging journey ahead. Our commitment to creating a Melody Generation system that resonates with the spirit of folk melodies remains unwavering.

As we move forward, the melodies that will emanate from our system are not just sequences of notes; they will be a testament to our dedication, our learning, and our passion for the world of music. We eagerly anticipate the melodies yet to be composed and the harmonies yet to be discovered.

In conclusion, the road ahead is filled with promise, and the completion of the preprocessing phase is but the first step towards achieving our goals. With each passing milestone, our Melody Generation project draws closer to becoming a reality.

References

[1] Conner Michael, et al. Music Generation Using an LSTM arXiv 22 Mar 2022

[2] Mangal. Sanidhya. et al. LSTM Based Music Generation System. LARJSET, vol. 6, no. 5. May 2019, pp. 47-54.

[3] Lou. "Music Generation Using Neural Networks. Accessed: Jul 26. 2023