Innovation Center for Education

TRUSTED DELIVERY PARTNER for IBM ICE

University of Petroleum and Energy Studies

Internship - Low Level Design

on

Design a deep learning algorithm for novel music generation

Team members:
Brihad Gunapu (R214220354)
Chehek Narula (R214220386)
Chirag (R214220370)
Chirag Pawadia (R214220368)
Desh Iyer (R214220360

Guided by:
Sumit Kumar Shukla

Table of Contents

1. INTRODUCTION

1. Scope of the Document

The primary goal of this project is to develop an advanced deep learning algorithm capable of creating original music compositions. The process involves curating a diverse dataset of music compositions, selecting a suitable deep learning architecture, and training the model to understand musical patterns and structures. It is important to note that the algorithm's purpose is to enhance human creativity rather than replace musicians. To assess the quality and musicality of the generated compositions, objective evaluations and user feedback will be utilized. The project will also address ethical considerations, including copyright issues, and focus solely on generating music in an AI-assisted manner, excluding audio synthesis.

2. Intended Audience

he project's target audience includes the following:

1. Musicians and Music Enthusiasts: Musicians, composers, and music enthusiasts will be interested in exploring the algorithm's capabilities to generate novel musical ideas and find inspiration for their own compositions. They will be able to interact with the model through a user-friendly interface, enabling them to explore various musical styles and themes.

2. AI Researchers and Developers: AI researchers and developers with an interest in music generation and deep learning will find this project relevant. They can gain valuable insights into the application of deep learning techniques for creative tasks and understand the challenges and opportunities in generating original music compositions.

3. Music Industry Professionals: Professionals in the music industry may be curious about the potential of AI-assisted music generation and how it can be integrated into music production processes.

4. General Public: The project's outcomes, particularly the user interface for interactive music creation, may appeal to the general public interested in AI technology and music.

3. System Overview

The Automatic Music Generation System comprises the following main components:

a. Data Collection: The system starts by collecting musical notes from a diverse collection of MIDI files, chosen for their compact representation and the availability of various musical instruments.

b. Data Preprocessing: The gathered musical notes undergo preprocessing to convert them into a suitable format for training the LSTM model. This process involves encoding musical notes, representing time durations, and organizing the data into sequences.

c. LSTM-based Model: The core of the system is the LSTM-based model, selected for its ability to remember and capture long-range dependencies in sequential data, making it ideal for generating coherent musical sequences. The model is trained using the preprocessed musical notes, where it learns to predict the next set of notes based on the input sequence.

d. Model Training: The LSTM model is trained on a dataset of musical note sequences. During training, the model's weights are optimized by minimizing the loss function, which quantifies the difference between the predicted notes and the actual notes from the training data.

e. Music Generation: After training, the LSTM model is utilized to generate new musical notes. The process begins with a seed sequence, which can be either random or user-defined. The model then iteratively predicts subsequent notes, considering its internal memory and the sequence of previously generated notes.

f. MIDI File Creation: The final step involves transforming the generated musical notes into a MIDI file format. The MIDI format stores musical information, such as pitch, duration, and velocity, making it suitable for representing the newly composed music piece.

2. LOW LEVEL SYSTEM DESIGN

4. Sequence Diagram:

```
Title: Music Composition Sequence Diagram

Actor: User

User -> UI: Select music style, mood, and constraints
UI -> Model: Request to generate music with given inputs
Model -> Data: Retrieve preprocessed music dataset
Data -> Model: Provide preprocessed dataset
Model -> Model: Initialize deep learning model
Model -> Model: Load pre-trained weights (optional)
Model -> Model: Generate music based on inputs
Model -> UI: Return generated music
UI -> User: Display generated music
User -> UI: Provide feedback and customization
```

5. Navigation Flow/UI Implementation:

The Navigation Flow and User Interface (UI) are essential components of the Automatic Music Generation System. The UI allows users to interact with the system seamlessly and control various functionalities. The Navigation Flow guides users through the different steps of the music generation process. The UI's key features include:

1. Music Genre and Mood Selection: Users can select a music genre and mood to influence the composition's style and tone.

2. Generate Music Button: Clicking "Generate Music" initiates the AI-powered music generation process.

3. Loading Animation/Progress Bar: While the music is being generated, a loading animation or progress bar is displayed to indicate ongoing processing.

4. Music Preview and Playback Controls: The UI presents the generated music for preview, with playback controls for play, pause, and stop to assess the composition.

5. Saving Options: Users can save the composition as MIDI or audio files for offline use.

6. Feedback Collection: Optional feedback collection through ratings or text input to gather user insights and improve the model.

7. Ethical Considerations: Information and disclaimers about responsible use of AI-generated content.

8. Additional Sections: Project details, help, and support information may be included for users' convenience.

1. Configurations/Settings

The Automatic Music Generation System relies on specific configurations and settings to implement the LSTM-based model for music composition. Key configurations include:

a. LSTM Architecture: Configuring the LSTM model with an appropriate number of LSTM layers and hidden units to balance model complexity and overfitting. The depth of the network influences its ability to capture intricate musical patterns.

b. Sequence Length: Defining the sequence length used for training the LSTM model to capture long-range dependencies in music data. Longer sequences offer more context but may demand more memory and computational resources.

c. Batch Size: Determining the batch size, which affects training efficiency and dataset representation during each optimization step.

d. Learning Rate: Setting the learning rate to control optimization step size and achieve efficient convergence during training.

e. Loss Function: Choosing an appropriate loss function, such as categorical cross-entropy, to guide the model's accurate note predictions.

f. Training Epochs: Determining the number of training epochs to prevent overfitting or underfitting and employing early stopping strategies if needed.

g. Seed Sequence: Choosing the initial seed sequence for music generation, influencing the output's musical composition.

h. Temperature: Adjusting the temperature parameter to control the diversity of generated notes, affecting creativity and variability.

2. DATA DESIGN

The data design for the music composition project includes managing various data types used throughout the system:

1. Music Dataset: A diverse and preprocessed music dataset containing MIDI files or other suitable formats for model training, organized based on genres, artists, and relevant metadata.

2. User Preferences: Structured storage of user-selected preferences for music generation, such as genre and mood.

3. Generated Compositions: Storage of AI-generated music compositions in a standardized format, along with associated metadata like genre, mood, and composition length.

4. Evaluation Metrics and Feedback: Records of evaluation metrics and user feedback to assess the quality and originality of generated compositions.

5. Real Musicians' Input (Potential Integration): Storage of contributions made by real musicians to hybrid compositions, including relevant information about collaborating musicians and their input.

6. User Accounts (Potential Integration): If user accounts are implemented, secure storage of user credentials and user-specific data, such as saved compositions and preferences.

7. Ethical Considerations and Disclaimers: Data related to ethical considerations, copyright information, and disclaimers for responsible use of AI-generated content.

3. DETAILS OF OTHER FRAMEWORKS BEING USED

The data design for the music composition project includes managing various data types used throughout the system:

1. Music Dataset: A diverse and preprocessed music dataset containing MIDI files or other

suitable formats for model training, organized based on genres, artists, and relevant metadata.

2. User Preferences: Structured storage of user-selected preferences for music generation, such as genre and mood.

3. Generated Compositions: Storage of AI-generated music compositions in a standardized format, along with associated metadata like genre, mood, and composition length.

4. Evaluation Metrics and Feedback: Records of evaluation metrics and user feedback to assess the quality and originality of generated compositions.

5. Real Musicians' Input (Potential Integration): Storage of contributions made by real musicians to hybrid compositions, including relevant information about collaborating musicians and their input.

6. User Accounts (Potential Integration): If user accounts are implemented, secure storage of user credentials and user-specific data, such as saved compositions and preferences.

7. Ethical Considerations and Disclaimers: Data related to ethical considerations, copyright information, and disclaimers for responsible use of AI-generated content.

DETAILS OF OTHER FRAMEWORKS BEING USED

The project utilizes the following frameworks to achieve its objectives:

a. TensorFlow (Version 2.3.1): An open-source deep learning framework by Google, suitable for building and training machine learning models, including LSTM-based models. It supports GPU acceleration, beneficial for training complex models.

b. music21 (Version 5.5.0): A Python library designed for music composition, analysis, and processing. It aids in handling music data, including parsing MIDI files, working with musical notations, and generating MIDI files.

c. NumPy (Version 1.19.5): A fundamental library for numerical computing in Python, helpful for processing numerical data, such as musical notes, efficiently.

d. scikit-

learn (Version 0.24.2): A popular machine learning library offering tools for data preprocessing, model selection, and evaluation. It may be used for tasks such as data splitting, evaluation metrics, and post-processing of LSTM predictions.

Caching:

Caching plays a crucial role in the music generation model to avoid redundant computations during the prediction phase. Caching can be effectively applied in the following components:

a. Preprocessing Phase: Caching preprocessed data, such as note sequences and numerical representations, to speed up training and generation.

b. Training Phase: Caching gradients of the model's parameters during training to avoid redundant calculations during each epoch.

c. Prediction Phase: Caching intermediate states of the model during music generation to optimize subsequent note predictions.

d. MIDI File Generation: Caching the mapping between numerical representations and MIDI notes to avoid recalculating MIDI notes for each generated sequence.

REFERENCES

1. Magenta Project: Google's Magenta project offers pre-trained models and tools for music generation using machine learning techniques.
https://magenta.tensorflow.org/

2. DeepBach and Music Transformer: Research papers explore deep learning models like DeepBach and Music Transformer for generating coherent musical compositions.
https://arxiv.org/abs/1612.01010

3. Jazz Transformer: A model capable of generating jazz melodies and collaborating with human musicians.
https://arxiv.org/abs/2104.00270

4. Symbolic Music Genre Transfer with Transformer: A model that can transfer music genres in symbolic music data.
https://arxiv.org/abs/2101.00611

5. Survey and Tutorial Papers: There are survey and tutorial papers available that provide insights into various deep learning techniques used in music generation tasks.
https://arxiv.org/abs/2006.00702

Innovation Center for Education

TRUSTED DELIVERY PARTNER for IBM ICE