DEEP LEARNING FOR MUSIC GENERATION

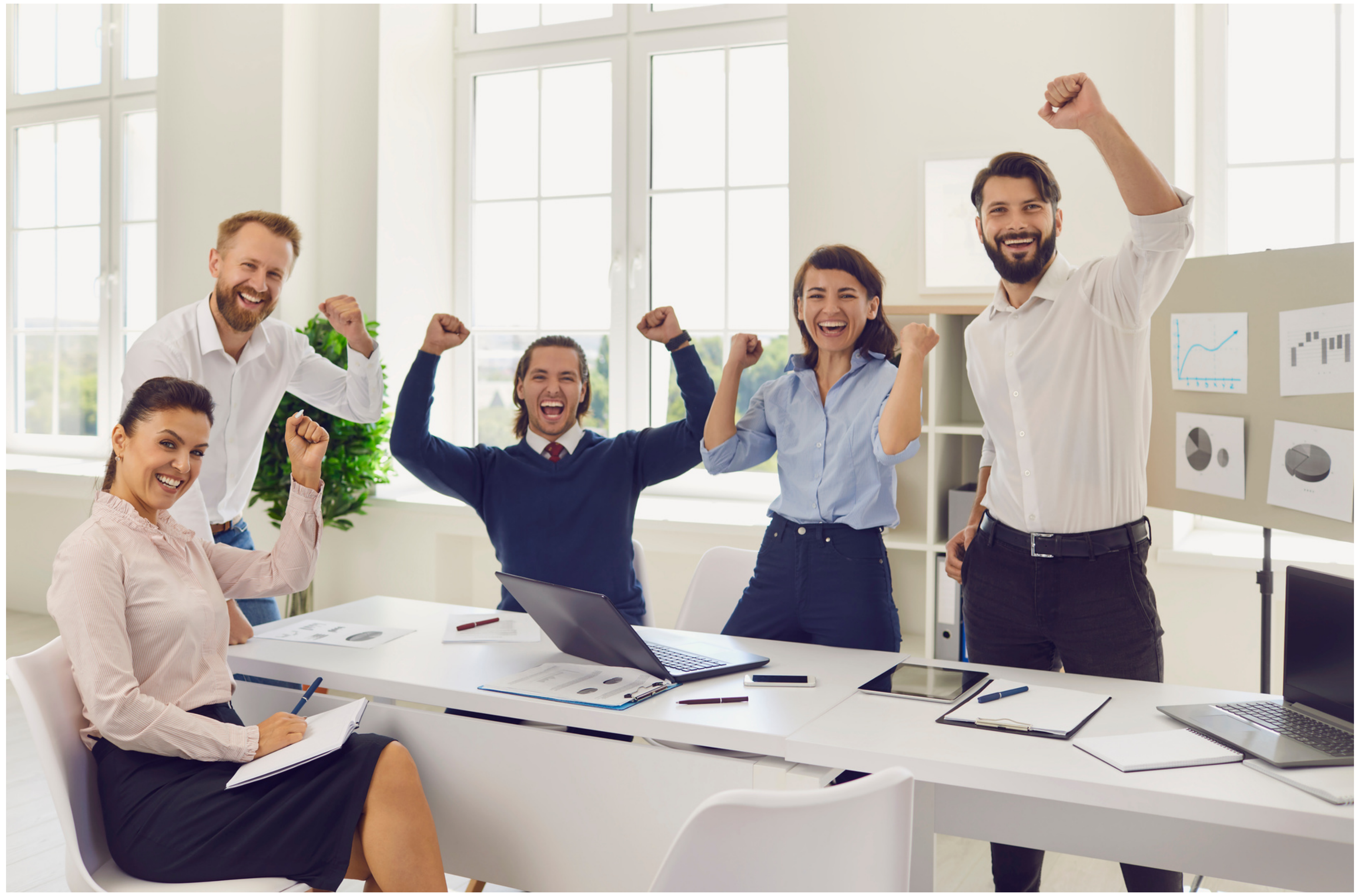
OUR TEAM

- JASEEM ALI T

  MSC DATA ANALYTICS WITH SPECIALIZATION IN COMPUTATIONAL SCIENCE

- MOHAMMED HAFEEZ KK

  MSC DATA ANALYTICS WITH SPECIALIZATION IN COMPUTATIONAL SCIENCE



INTRODUCTION

In the dynamic landscape of music composition, the fusion of human creativity and artificial intelligence (AI) has given rise to Automatic Music Generation (AMG). Leveraging machine learning, particularly Long Short-Term Memory (LSTM) networks, this project explores the potential of AI to enhance human creativity in crafting original music. Beyond AI applications, the endeavor contributes to the evolution of musical expression by unraveling the nuances of algorithmic creativity. The goal is to deepen the conversation between AI and music, exploring collaborative possibilities in the creative process.

MILESTONE

Automatic Music Generation (AMG) has generated significant interest in the fields of artificial intelligence and music composition. Numerous approaches to autonomous music generation have been proposed, including rule-based systems and machine learning techniques.

Long Short-Term Memory (LSTM) networks, a type of RNN, have gained popularity in AMG due to their ability to capture long-range dependencies while mitigating the vanishing gradient problem.

METHODOLOGY

The project began with the collection of a diverse dataset of classical piano MIDI files. The MIDI files were obtained from a variety of online repositories and libraries, and included compositions by renowned classical composers.

1. DATA COLLECTION
2. DATA PROCESSING
3. MODEL DEVLOPMENT

METHODOLOGY

4 MODEL DEVELOPMENT

5 TRAINING

6 EVALUATION AND DEPLOYMENT

DATA PROCESSING

Once the dataset had been compiled, the MIDI files were preprocessed to extract relevant musical features. This entailed parsing MIDI files to identify notes, chords, and other musical components.

MIDI File Parsing: The first step in data processing is parsing the MIDI files to extract the musical elements necessary for training the LSTM model. MIDI files contain information about notes, chords, tempo, and other musical attributes encoded in a standardized format.

Encoding Musical Tokens: After extracting the musical elements, they need to be encoded into a format suitable for input to the LSTM model

Sequence Generation: Once the musical elements are encoded, sequences of tokens are generated to represent the input data for the LSTM model.

DATA PROCESSING

Padding and Truncation: To ensure uniformity in sequence length, padding or truncation may be applied to the generated sequences

Data Splitting: Finally, the processed data is split into training and validation sets for model training and evaluation.

MODEL DEVELOPMENT

Architecture Design:

The first step in model development is designing the architecture of the LSTM-based AMG model. The architecture determines the structure and connectivity of the neural network, including the number of LSTM layers, the number of units (neurons) in each layer, and the connections between layers

Input Representation:

The input data, consisting of sequences of encoded musical tokens, needs to be formatted into tensors suitable for processing by the LSTM layers.

MODEL DEVELOPMENT

LSTM Layer Configuration:
The LSTM layers constitute the core of the model and are responsible for capturing temporal dependencies and patterns within the input sequences

Output Layer Design:
After the LSTM layers, an output layer is added to the model to generate predictions based on the learned representations

.

MODEL DEVELOPMENT

Model Compilation:

Once the architecture is defined, the model is compiled by specifying additional training parameters such as the loss function, optimizer, and evaluation metrics

Hyperparameter Tuning

Finally, hyperparameter tuning is performed to optimize the model's performance and generalization capabilities

Training

The LSTM-based model was trained on the preprocessed MIDI dataset using a combination of training and validation sets. The training process involved minimizing a chosen loss function, typically categorical cross-entropy or mean squared error, through gradient-based optimization techniques such as stochastic gradient descent (SGD) or Adam optimization

Evaluation

Following training, the performance of the LSTM-based AMG model was measured using a variety of metrics and qualitative assessments. Quantitative metrics like accuracy and loss provided information about the model's predictive capabilities and generalisation performance

Key Steps in Code

- Installed necessary libraries: NumPy, Music21, TensorFlow, Scikit-learn
- Loaded required libraries
- Defined a function to read MIDI files and extract piano notes
- Read MIDI files, parsed them, and extracted piano notes
- Filtered out infrequent notes and created a new dataset
- Preprocessed the data by converting notes to numerical indices
- Split the data into training and testing sets
- Created a sequential LSTM model with two stacked layers
- Compiled the model using the Adam optimizer and sparse categorical cross-entropy loss function
- Trained the model on the training data and validated on the testing data
- Saved the trained model for future predictions
- Loaded the saved model for generating new music sequences
- Generated a new music sequence by predicting notes based on a random input sequence
- Converted the predicted notes into a MIDI file and saved it

RESULT

Model Training and Evaluation:

- LSTM-based AMG model trained on classical piano MIDI files for 80 epochs.
- Batch size of 128 used during training with Adam optimizer..
- Final validation accuracy approximately 85%, indicating good generalization.

Music Generation:

- LSTM model used to generate new musical compositions.
- Random sequence from the testing dataset used as the initial input pattern.
- Model iteratively predicted the next note or chord, repeated 200 times.
- Generated musical sequence saved as "jas.midi."

LIMITATIONS AND CHALLENGES

Computing Resources:

- Significant need for computational resources during complex LSTM model training.
- Limited access to high-performance computing infrastructure impacted experimentation scale and speed.

Data Availability and Quality:

- Dataset limitations in size and diversity may have affected model generalization.
- Inconsistencies or inaccuracies in MIDI files could introduce noise or bias into training data.

Model Complexity and Tuning:

- Designing and optimizing LSTM-based AMG model involved navigating a complex landscape.
- Extensive experimentation and tuning needed for selecting appropriate hyperparameters.

CONCLUSION

- DEMONSTRATED MACHINE LEARNING'S POTENTIAL IN COMPOSING CLASSICAL PIANO-LIKE PIECES THROUGH METICULOUS DATA PREPROCESSING AND MODEL DEVELOPMENT
- ACKNOWLEDGED LIMITATIONS IN COMPUTATIONAL RESOURCES, DATA QUALITY, AND SUBJECTIVE EVALUATIONS.
- DESPITE CHALLENGES, OUR PROJECT MARKS A STEP TOWARDS UNLOCKING AI'S CREATIVE POTENTIAL IN MUSIC COMPOSITION, EMPHASIZING THE NEED FOR ONGOING RESEARCH AND INNOVATION FOR BROADER ARTISTIC EXPRESSION.

THANK YOU!