Music Generation Using Transformer-based Models

**Abstract**

This project explores the application of Transformer-based neural networks for generating diverse and musically coherent sequences in MIDI format. By training on a rich dataset of piano compositions, the model learns the temporal dependencies and patterns of musical notes. Leveraging advanced techniques such as top-k sampling and temperature scaling, the model generates unique pieces, showcasing the potential of AI in creative domains like music composition.

**Introduction**

**Motivation**

Music is a universal language that resonates with human emotions. The advent of AI in creative fields has opened possibilities for automating music composition, providing tools for composers, and creating novel musical experiences. This project seeks to harness the power of Transformers to generate music, demonstrating how deep learning can contribute to creative endeavors.

**Problem Statement**

Generating music that is both coherent and expressive is a challenging task. It requires understanding long-term dependencies, maintaining a sense of rhythm, and adhering to musical rules—all of which are inherently sequential problems.

**Objective**

The primary objective of this project is to design, train, and evaluate a Transformer-based model capable of generating musically coherent sequences in MIDI format. The generated sequences should reflect the diversity and structure present in the training data.

**Dataset**

**Dataset Description**

The project utilizes the **MAESTRO Dataset**, a large collection of MIDI files of piano performances. Key attributes include:

* **Source**: MAESTRO (MIDI and Audio Edited for Synchronous Tracks and Organization).
* **Contents**: MIDI files with precise timing and expressive dynamics.
* **Size**: Over 1,200 hours of annotated piano performances.

**Preprocessing**

* **Tokenization**: MIDI files are converted into sequences of tokens representing notes, velocities, and durations.
* **Positional Encoding**: A positional encoding scheme is applied to capture the temporal order of notes.
* **Tools Used**: Libraries such as pretty\_midi and midiutil facilitate parsing and encoding MIDI data.

**Model Architecture**

**Overview**

The Transformer model, originally designed for natural language processing tasks, excels at sequence modeling due to its self-attention mechanism. This project adapts the Transformer architecture to generate music by treating MIDI sequences as analogous to text.

**Detailed Architecture**

* **Input Representation**:
  + Token embeddings represent musical events.
  + Positional encodings capture the order of tokens.
* **Transformer Layers**:
  + Multi-Head Attention layers to model relationships between notes.
  + Feed-Forward Neural Networks for feature transformation.
  + Layer Normalization for stability.
* **Output Layer**:
  + Dense layer with softmax activation for token prediction.

**Model Parameters**:

* Embedding size: 256
* Number of Transformer layers: 4
* Attention heads: 8
* Vocabulary size: 307

**Training Process**

**Data Splitting**

The dataset is divided into:

* **Training Set**: 80%
* **Validation Set**: 10%
* **Test Set**: 10%

**Training Details**

* **Framework**: TensorFlow/Keras
* **Hyperparameters**:
  + Batch size: 64
  + Learning rate: 0.001
  + Epochs: 10
* **Loss Function**: Sparse categorical cross-entropy.

**Visualization**

Training and validation loss over epochs:

* The training loss decreased steadily, indicating effective learning.
* Validation loss plateaued, showing the model’s ability to generalize.

**Music Generation**

**Generation Process**

1. **Seed Sequence**: Start with a short sequence of tokens.
2. **Token Prediction**: Use the trained model to predict subsequent tokens.
3. **Sampling Techniques**:
   * **Top-k Sampling**: Restrict predictions to the top-k most likely tokens.
   * **Temperature Scaling**: Adjust the randomness of predictions.
   * **Excluding Top Predictions**: Avoid overused patterns.

**Saving Outputs**

* Generated sequences are converted back to MIDI format using midiutil.
* Each piece is saved as a standalone MIDI file.

**Generated Samples**

The model generated diverse and musically coherent pieces, capturing rhythmic and melodic patterns seen in the training data. Files are available in the generated\_music folder for review.

**Results and Analysis**

**Quantitative Results**

* **Training Loss**: Steadily decreased from 5.4 to 5.212.
* **Validation Loss**: Plateaued indicating stable generalization.

**Qualitative Results**

* **Strengths**:
  + Captures long-term dependencies.
  + Generates diverse musical structures.
* **Weaknesses**:
  + Occasionally produces abrupt transitions.
  + Limited dynamics in some sequences.

**Comparison with Baselines**

Compared to simpler models like RNNs or LSTMs, the Transformer demonstrated superior performance in capturing temporal dependencies.

**Applications and Future Work**

**Applications**

* **Creative Tools**: Assist composers in generating ideas.
* **Game and Film Music**: Automated background score generation.
* **Educational Tools**: Aid in teaching music composition.

**Future Improvements**

* Expand the dataset to include multi-instrument compositions.
* Introduce style conditioning for genre-specific music generation.
* Develop a real-time interface for user interaction.

**Conclusion**

This project successfully demonstrates the potential of Transformer models for music generation. By leveraging advanced techniques in sequence modeling, the system generates coherent and expressive musical sequences, showcasing AI’s capacity to contribute to creative fields.

**References**

1. Vaswani, A., et al. "Attention Is All You Need." Advances in Neural Information Processing Systems. 2017.
2. Hawthorne, C., et al. "Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset." arXiv preprint arXiv:1810.12247.
3. Libraries: TensorFlow, Keras, pretty\_midi, midiutil.

**Appendices**

* **Model Architecture**:

A screen shot of a computer program

Description automatically generated

* **Visualizations**
  + A graph with orange lines and numbers

    Description automatically generated