Instructor: Mrs Sumayah Zahid

Member1: Syed Munib ur Rehman 21K4603

Member2: Asim Khan 21k4685

Music Generation using GANs and Transformers

Objective

The primary goal of this project is to generate expressive and coherent piano music by leveraging deep generative models. Specifically, we explore and compare two distinct architectures: Generative Adversarial Networks (GANs) and Transformers, both applied on the MAESTRO dataset. The objective is to evaluate the quality, diversity, and realism of the music generated through these models.

Problem Statement

Music generation is a challenging task due to its sequential, hierarchical, and highly structured nature. Traditional models often fail to capture long-range dependencies and musical patterns. This project seeks to address the question: *Can modern deep learning techniques such as GANs and Transformers generate musically coherent and stylistically rich piano performances from scratch?* We aim to analyze the strengths and limitations of each approach by applying them individually on the MAESTRO dataset.

Methodology

Dataset

The **MAESTRO dataset** is used, which contains over 200 hours of virtuosic piano performances aligned with MIDI and audio recordings. For this project, we use the MIDI representations, which were preprocessed into piano roll sequences to be used for training.

Generative Adversarial Networks (GANs)

The GAN framework consists of two models: a **Generator**, which learns to produce realistic music sequences from random noise, and a **Discriminator**, which attempts to distinguish between real sequences from the dataset and fake sequences generated by the Generator.

Generator Architecture

The Generator begins with a latent vector of dimension 256, which is transformed through a linear projection and then passed through several 1D transposed convolution layers. The architecture expands the latent vector into a sequence of shape `[131, 480]`, corresponding to the MIDI feature representation.

Key components:

- Linear + ReLU block to reshape the latent input.
- Three layers of `ConvTranspose1d` with increasing spatial resolution and decreasing channel depth.
- Final activation using `Sigmoid` to bound the output between [0, 1].

Discriminator Architecture

The Discriminator uses spectral normalization to stabilize training and includes three `Conv1d` layers followed by dropout to avoid overfitting. The output is pooled and passed through a dense layer to classify sequences as real or fake.

Key components:

- Spectral normalization in each convolutional layer.
- LeakyReLU activations.
- Dropout (0.3) after each conv layer.
- Adaptive mean pooling across the time dimension.
- Final dense layer with sigmoid for binary classification.

Training Details and Improvements

- **Loss Function**: Binary Cross Entropy Loss (BCELoss).
- **Optimizers**: Adam with betas `(0.5, 0.999)`.
- **Learning Rates**: Generator = `0.0001`, Discriminator = `0.0002`.
- **Label Smoothing**: Real = `0.9`, Fake = `0.1`.
- **Training Schedule**: Generator trained once per epoch; Discriminator trained more aggressively in earlier experiments.
- **Other Tweaks**: Tried increasing dropout, altering learning rates and betas, and using spectral normalization.

Transformers

The Music Transformer is an autoregressive sequence model tailored for symbolic music generation. It uses the Transformer encoder architecture to model long-term dependencies in

MIDI token sequences. Instead of random noise, it takes previously generated or real musical tokens as input and predicts the next token in the sequence.

Model Architecture

The Music Transformer is built using an encoder-only Transformer architecture, optimized for sequence modeling of REMI-tokenized MIDI data. The model begins with an embedding layer that transforms token IDs into dense vectors of size 512. These embeddings are combined with learnable positional encodings to maintain temporal structure.

The core consists of 6 stacked Transformer encoder layers, each composed of multi-head self-attention (8 heads) and a position-wise feedforward network. Dropout is applied after embeddings and within the attention and feedforward blocks to regularize training. A final normalization layer is followed by a projection head that maps the hidden states to vocabulary logits for next-token prediction..

Key components:
• Token embedding + positional encoding to capture sequence information.
• Multi-head self-attention with 8 heads.
• Feedforward layer with dimension 2048 and ReLU activation.
• Layer normalization and residual connections throughout.
• Output layer projects back to vocabulary size for token prediction.

Training Details and Improvements:
• Loss Function: CrossEntropyLoss for next-token prediction.
• Optimizer: Adam with learning rate 1e-4.
• Batch Size: 16; Sequence Length: 512 tokens.
• Token Sampling: Top-k (k=5) used during inference for diversity.
• Trained on Google Colab CPU with periodic checkpoint saving.
• Sequence reconstruction handled via inverse REMI tokenization to MIDI.

Results

The GAN model was able to generate sequences that were musically plausible and structurally coherent to a limited degree. Early in training, the outputs resembled noise, but as training progressed, generated sequences began exhibiting basic rhythmic and pitch structures. The best results were achieved using:

- Spectral normalization in the discriminator

- Balanced learning rates with label smoothing
- Dropout to regularize discriminator behavior

Subjective listening tests indicated that while the GAN-generated sequences captured rhythmic motifs, they lacked long-term harmonic progression and global structure and, in general, were not able to mimic human style sequences. This could be mainly due to the limitations of Vanilla Gan, as well as the lack of Temporal Learning in the Generator

The Music Transformer was able to generate sequences that were musically coherent and rhythmically consistent, showing clear improvement over the GAN model. The generated music maintained a logical flow and recognizable motifs, with better handling of timing and note transitions.

The best results were achieved using:
• Learnable positional embeddings
• Dropout regularization and layer normalization
• Longer sequence lengths (up to 2048 tokens)

Subjective listening tests indicated that while the outputs were not yet on par with professional human compositions, they showed promising structure and musicality. Overall, the Transformer demonstrated a stronger ability to capture temporal dependencies and produce listenable sequences compared to the GAN approach.

References

1. Hawthorne, C., et al. (2019). *Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset*. arXiv:1810.12247
2. Goodfellow, I., et al. (2014). *Generative Adversarial Nets*. Advances in Neural Information Processing Systems.
3. Radford, A., Metz, L., & Chintala, S. (2016). *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*.
4. Miyato, T., & Koyama, M. (2018). *Spectral Normalization for Generative Adversarial Networks*.