Melody Generator with LSTM and Transformer

Keerthana Golla, Rice University

Problem Statement

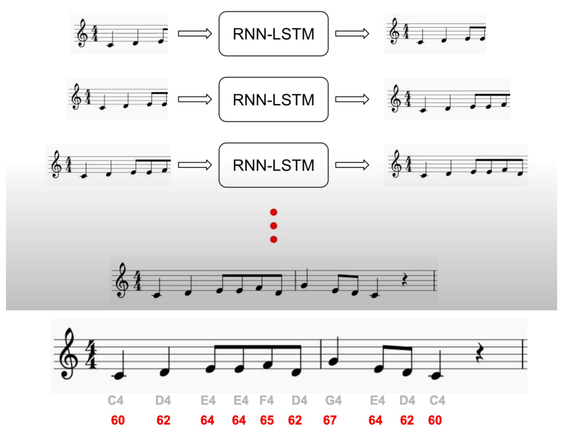
This project aims to explore the synergy of Long Short-Term Memory (LSTM) networks and Transformer models in the context of melody generation. Leveraging the music21 toolkit and diverse datasets in MIDI and Kern formats, our objective is to create a versatile music generator capable of producing music in various predefined styles, with a specific focus on melody compositions.

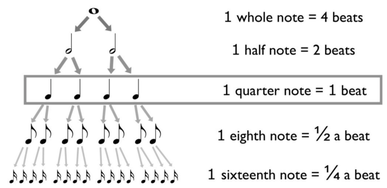
DataSets

- I leveraged a total of 2 main datasets for this Melody Generator project. The initial dataset is from Jazz Kaggle, providing a foundation for jazz compositions in MIDI format. The second dataset, collected from humdrum.org in Kern format, enriches the diversity of our training data.
- There are over 20k songs available in combined. I strategically utilized a subset of around 2100 songs from these datasets for testing purposes. These subsets were meticulously preprocessed to accommodate our computational limitations while ensuring the integrity of the music data in our analysis.

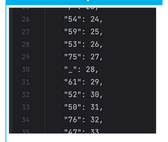
Architecture

Melody is the sequential of Notes and Rests



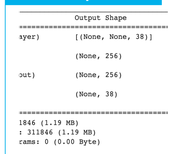


Step 1



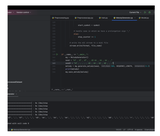
Preprocessing and converting it to a time-series prediction problem

Step 2



Model Training - LSTM or Transformer

Step 3



Generate the new sequence using the saved model

Procedure

- Preprocessing.py : In preprocessing step loads Kern format songs, filters for acceptable durations, transposes to a standardized key, and encodes them into a time-series-like format. The preprocessed songs are then combined into a single file, symbol-to-integer mappings are created, and training sequences are generated.

Model Training

- LSTM- Train.py: I configured an LSTM-based music generation neural network with specified output units and incorporates a single LSTM layer with dropout regularization. It utilizes training sequences derived from preprocessed music data, employing the Adam optimizer and sparse categorical cross entropy loss during training. Key hyperparameters such as output units and learning rate are tuned, and the resulting model is saved in HDF5 format for subsequent melody generation.
- Transformer Train.py: Here, I configured the model architecture with BERT embeddings and a pooling layer, compiles it with specified loss and optimizer, and trains it using generated training sequences. The input data for training includes BERT input_ids and attention_mask. The training progress is logged, and the trained model is saved in an HDF5 file for later use in generating melodies. The hyperparameters, including output units and learning rate, are adjustable to fine-tune the model.
- In both the cases - I have chosen Adam optimizer , Hyperparameters would be - output units , Epochs and learning rate. I have introduced Dropout layers in both LSTM and Transformer so to avoid overfitting . Additionally used early stopping in case of LSTMs by limiting the number of epochs.

Melody Generation

- Melody Generator.py- The generator takes a seed melody, extends it using start symbols, and predicts subsequent symbols based on the model. The generated melody is constructed considering note/rest durations and encoded symbols. Finally, the resulting melody is saved as a MIDI file.

Results

For LSTM – 5 EPOCHS

For LSTM 50 epochs

For Transformers

Conclusion

- Through meticulous preprocessing, model training, and the incorporation of BERT embeddings, we demonstrated the ability to capture complex musical patterns. The Melody Generator class serves as a testament to the practical application of our research. This project opens exciting possibilities for the intersection of deep learning and music composition
- Looking ahead, there are several avenues for future exploration. Fine-tuning hyperparameters, experimenting with different datasets, and enhancing the model's creativity through more sophisticated sampling techniques could elevate the melody generation process. Additionally, we can explore integrating the music field with the video field. One such interesting use case would be generating music from dance videos

Works Cited

- Hiehn, Steve. "How to Generate Music with Python: The Basics." Medium. Link
- Humdrum. "MIDI Commands." Understanding Music Representation. Link
- Valerio Velardo - The Sound of AI. "Technical Terms and In-Depth Knowledge of Music.". Link
- "Music Generation Using GANs." IEEE Xplore. Link
- Kaggle. "Music Generation with LSTM." Link