Good evening, everyone. I'm Keerthana, a first-year master's student in computer science. Today, I'd like to share my exploration of LSTMs and Transformers, specifically BERT, in the context of generating melodies. Similarly, not just melodies but you can focus on any particularly predefined genre like JAss. I've worked with datasets containing melodies in MIDI and Kern formats, and I'll explain why I chose these formats shortly. Although I initially had 20k songs from Kaggle and humdrum.org, computational constraints led me to focus on 2k songs for this project.

I've broken down the problem into three key steps: data preprocessing, model training, and melody generation using the trained model. Each step is implemented in a separate Python file. Before diving into preprocessing, it's crucial to decide on the data format. I opted for MIDI and Kern formats because they represent melodies as sequential notes and rests. Unlike MP3 or WAV formats, which require spectrograms. these MIDI/KERN formats are ideal for time-series prediction, especially for handling long sequence relations. For this reason, I chose unidirectional LSTM and a (BERT). Bidirectional Encoder Representations from Transformers, which boosts melody creation by understanding musical notes in both directions.

In Midi/kern each note is represented to a number .and in the preprocessing step, , with the help of music21 toolkit , I have given few specified acceptable durations, excluded invalid durations. I converted all kind of different notes into C major and A minor. Lastly I mapped unique symbols generated in the process, including 'r' for rests and '/' for the end symbol. After doing the above steps for all the songs in dataset , I combined them in one single file so as to break them as sequence and input to the training model. The same preprocessed file was used for both LSTM and transformers with minute changes.

Notably, the number of input nodes varies based on the dataset. With 38 different mappings in my dataset, I had an input layer of 38 nodes and one middle layer, including drop out layer for lstm when coming to transformer a pooling layer is added to summarizes the information for the model to interpret effectively, ensuring a harmonious output in melody generation.

.

Optimizer Choice:

I opted for the Adam optimizer, a popular choice for training neural networks. Adam combines the advantages of two other extensions of stochastic gradient descent, providing efficient and effective model updates.
Handling Overfitting:

To prevent overfitting, I introduced Dropout layers in both LSTM and Transformer models. Additionally, early stopping mechanisms were employed in LSTMs, limiting the number of epochs to ensure optimal performance.

I saved both trained models in an h5 file for future use in melody generation.

After training, input a seed sequence into the model. The model predicts subsequent elements iteratively, extending the sequence, and the process continues until a defined termination condition. The final output is a new melody reflecting learned patterns and creative variations.

Lets listen to the music generated by LSTM 5 epochs , LSTM 50 epochs, Transformers 5 epochs. Lets compare each of them and you will find it interesting, atleast I found it interesting.
notable point is tho I have given the same start symbol to each model.I got a music sequence of different lengths and different melodies as a awhole.

In conclusion, my exploration of LSTM and BERT for melody generation has showcased their unique strengths. LSTM, with its sequential understanding, provides a solid foundation, while BERT, with bidirectional attention, adds a layer of complexity.
This project sets the stage for future work, such as fine-tuning hyperparameters, experimenting with diverse datasets, and enhancing creativity through advanced sampling techniques. Furthermore, the integration of the music and video domains could open up exciting possibilities, like generating music from dance videos.

I'd like to express my gratitude to the resources and references that contributed to this project.
Thank you for your attention, and I'm here to answer any questions you may have about this project.