Title: GenAI Assignment 6

Your Name: Michael Wood

Date: 12/6/24

Purpose

- In this project, I developed a small-scale music generation model using **Transformers**, specifically designed to generate new musical compositions in the style of **Bach's Cello Suites**.
- By utilizing MIDI files of the cello suites, I trained a Transformer model to predict musical notes and their corresponding durations.
- The model learns the underlying patterns in Bach's music, enabling it to generate new sequences of notes and durations that imitate the structure and style of the original works.
- The main objective here is to explore the application of Transformer models to music generation and understand the steps necessary for preparing, training, and evaluating such a model.
- Note: Much of the code for this project was borrowed from https://github.com/bforoura/GenAI/tree/main/Module7.

Tasks

1. **Data Preparation**: Convert MIDI files into sequences of notes and durations.
    a. This part was done with the transformers_util functions including the code to parse the midi files. The dataset of notes and durations was then created.
    b. I also loaded the initial data into my Google Drive and mounted my Drive in Colab.
2. **Tokenization**: Convert the notes and durations into tokens that the **Transformer** model can understand.
    a. This is where the training set for the model was created with help from the causal_attention_mask function.
3. **Model Architecture**: Build a Transformer-based architecture that processes musical notes and durations to generate new sequences.
    a. The model has input, embedding, and output layers for the notes and durations of the training set with one Transformer Block layer.
4. **Training**: Train the model on the processed data to learn the relationships between notes, durations, and musical structure.
    a. The initial model was trained on 100 epochs with the first 5 generated outputs.
    b. After those initial 5 generations, the training was moved up to 1000 epochs on the same dataset and model.
5. **Music Generation**: Use the trained model to **generate** new musical sequences that reflect the style of Bach's Cello Suites.
    a. The model generates a new musical sequence and outputs it to a file in my Google Drive in the transformer_output folder.

Discussion Points

- I found the results of this model to be quite interesting. I want to first mention that I loaded the output files in Musescore and changed the default instrument of piano to cello to more accurately reflect the Bach cello the model was trained on. Once I did this, the output seemed to at least match the sound of the training data. The model seemed to love starting on rests and ending with a staccato (a note with a dot above or below it). One thing that I found very intriguing was when a note in a generated output of the model (output-20241126-033555.mid) had a different color in Musescore. I believe this is to note a change in tone in Musescore. I am not sure if this was Musescore itself marking that note in particular or the generator actually stating the tone of that note, but it was a unique thing to see nonetheless.
- The model seemed to almost mimic Bach's style at times. It did not learn all of music theory with just thirty-six pieces of Bach, but that was to be expected. I found it strange to see a lot of sixteenth and eighth notes at first, but this is at least more sensible for a cello piece rather than a piano piece, which is an instrument I am more familiar with. At first, I thought this was a result of the max number of tokens the model was given to generate. However, when I changed the number of tokens around, it seemed as though it was more of a result of the model actually mimicking Bach's style. Looking at the initial training dataset, it seemed as though Bach did use a lot of those notes for the cello, so the model was able to see this style within those thirty-six pieces. As for how the generated output sounded, I am not too familiar with Bach's work, so I cannot say for certain how well it tonally matched his style. However, I do know a little music theory, and I can say that the generated output did at least follow certain rules at certain different times. I cannot say that the model fully learned all of the rules or identified certain music patterns completely, but with more training and configuration, a better model may be able to fully learn the rules of music and find ways to break them like an actual human.

Extra Credit

- I trained a second model on a new training set of midi files of Scott Joplin's piano pieces.
- I changed the batch size to 64 and the number of epochs to 500 for this model.
- I generated several outputs with varying numbers of tokens and temperatures and saved them to the output_joplin folder in my Google Drive.
- The results were very different compared to Bach. This was to be expected since this data was based on a different instrument with more complex notes and chords being played. Since this model was only meant to output one string of individual notes, the model had a hard time trying to make sense of chords and multiple notes being played at the same time. However, the ways it still tried to replicate this were interesting. For example, in output-20241203-180439.mid, there appeared what seemed to be notes tied closely together. This made the notes sound as if they were rolled across the piano, which resulted in a unique and interesting sound to the piece.
- The model was not able to fully mimic Joplin's style of ragtime piano, but it tried its best with lots of sixteenth notes gliding back and forth across the piano. Once again, the model does not have a great understanding of music theory, but it certainly learned something new and different with this different training set compared to training on Bach.