CSC 7343 Homework 2

In this homework, your goal is to train a transformer model, called piano music composer, to generate piano music.

Data

The piano data (in midi format) can be downloaded from:

https://storage.googleapis.com/magentadata/datasets/maestro/v1.0.0/maestro-v1.0.0-midi.zip

By default, when unzipped, the data will be put into a directory named “maestro-v1.0.0”.

The file “midi2seq.py” contains a set of functions that help to process the midi data and convert the data to sequences of events.
The file “model_base.py” contains the base classes that you should inherit when implementing the following model classes.

Composer

Implement a class called “Composer”. It should be a subclass of the class ComposerBase. You must use the exact class name.

In this homework, you should implement a **language model** as a composer. You should explore different transformer model architecture to obtain a good composer. When the “compose” member function is called, it should return a sequence of events that can be translated into piano music:

- ***The music should play for at least 20 seconds.***
- ***Randomness is required in the implementation of the compose function such that each call to the function should generate a different sequence.***

The function “seq2piano” in “midi2seq.py” can be used to convert the sequence into a midi object, which can be written to a midi file and played on a computer. Train the language model using the downloaded piano plays.

Submit your work

Develop and train your model so that your model can compose reasonable piano music pieces at least 20 seconds long. Do not modify the file “model_base.py”, instead, put all

your code in a single file named “hw2.py” (*you must use this file name*) and import the “ComposerBase” class in “hw2.py”. Submit only the file “hw2.py” in moodle. We will test your implementation using code similar to the following:

```
from hw2 import Composer
piano_seq = torch.from_numpy(process_midi_seq())
loader = DataLoader(TensorDataset(piano_seq), shuffle=True, batch_size=bsz)

cps = Composer()
for i in range(epoch):
    for x in loader:
        cps.train(x[0].cuda(0).long())

cps2 = Composer(load_trained=True)
midi = cps2.compose()
midi = seq2piano(midi)
midi.write('piano1.midi')
```

We will do the testing in google colab. Make sure code can run in colab on a GPU node.

Note that the above code trains your model from scratch. In addition, you should provide trained weights for your model. We may create your models by calling the constructor with “load_trained=True”. In this case, your class constructor should download the trained weights from your google drive and load the trained weights into the model class object. In the above code, cps2 should be a Composer model with the trained weights loaded. We should be able to call cps2.compose without training it and obtain a piano sequence from the downloaded trained model.