Module 6

Programming Assignment

Programming assignments are due by 11:59:59 PM on the day of the next lecture. All programming assignments are graded for clarity and functionality. Your code should be well-organized and include helpful comments where appropriate. Please upload your completed assignment to canvas as a zip file.

In this assignment, you will implement and execute a training routine for a small GPT model. Our dataset will be wikitext-103, a scrape of over 100 million tokens from good-quality wikipedia articles.

This is a large dataset, and GPT training can be cumbersome. You should train whatever size model you can comfortably run on your machine, on as much data as is needed to show progress. A GPU is highly recommended, but not necessary. In this assignment we are interested in the fidelity of the training code itself, not a high-quality final model (which would require lots and lots of compute to really work well).

For this assignment you will need the transformers and datasets libraries:

```
pip install transformers
pip install datasets
```

To assist in this assignment are several helper files, which you do not need to edit:

- **download_data.py** - downloads the wikitext-103 dataset and saves it as a text file, one sample per line. A sample is typically a paragraph from a wikipedia article or a title.
- **hftokenizer.py** - trains a GPT2-style tokenizer on the dataset.
- **gpt.py** - An implementation of a GPT model as a torch module. See this file for details on how to pass in appropriate model parameters. You are also encouraged to use your model from previous assignments, if you feel it is correct. If you choose to do this, please include it as an alternative torch module in your submission and explain how to use it (i.e. my_gpt.py).

Before starting, you should run **download_data.py** and then **hftokenizer.py** to download the data and train a tokenizer. This tokenizer will use a vocabulary of size 10,000 and an end-of-sequence token <|endoftext|>. These may each take several minutes to run.

For your assignment you should use the above files to complete:

construct_dataset.py - a script which converts the dataset into packed sequences of token_ids with a desired length. You will need to append samples with the end-of-sequence token. This script should shuffle the packed sequences and save them, to be used later in the training script.

train_model.py - a script which uses the tokenized dataset from *construct_dataset.py* to train a small GPT model. Your script should save a loss curve that demonstrates training progress (easiest to save this out every N batches).

You do not need to submit model weights.

You do not need to use gradient accumulation, weight decay, or weight tying. Weights are already initialized by gpt.py.

Your deliverables for this assignment are:

- A completed *construct_dataset.py* script
- A completed *train_model.py* script
- A loss curve showing the training progress (loss) as a function of tokens processed. Expect losses to start in the 8-12 range and end in the 3-5 range.
- A text file or comments in the code indicating the parameters you selected (model sizing, training hyperparameters, learning rates, initialization, etc).

If training over an entire epoch is not computationally feasible, train over whatever fraction of the dataset you are able to.