Module 5

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 a full GPT style language model using the components from previous assignments. You are encouraged to use your own implementations from previous weeks if you feel comfortable, but working implementations will also be provided. You only need to implement the architecture (you do NOT need to train the model).

Optional helper files include:

- **linear.py** - a linear layer implementation
- **embedding.py** - an embedding layer implementation. You will use two embedding instances in your model, one for tokens and one for position.
- **mha.py** - An implementation of multihead attention without masking.

Before building the full model you need to add a causal mask to multihead attention. You may use your implementation from last week or start with mha.py. Note that some implementations (i.e. PyTorch) separate this out from the attention mechanism, but since we are only considering GPT-style models we can bake the causal mask directly into our layer.

You should then complete two classes in gpt.py:

TransformerDecoderBlock - one "layer" of our model which includes a variety of subcomponents. See class lecture notes or the file itself for details.

GPTModel - a module which uses embeddings, transformer blocks, and any other modules you wish to design for convenience of implementation. For a batch of integer inputs representing token ids, this should output a corresponding batch of logits representing a distribution over the vocabulary.

NOTE: You do not need to train your model! Only implement the architecture. We will train our models next week.

Your deliverables for this assignment are:

- An updated *mha.py* that applies a causal mask.
- A completed *gpt.py* containing the TransformerDecoderBlock class and the GPTModel class.