CS-461, Winter 2025
Homework #2: Transformer-based Language Model
Due Date: Monday, February l7th @ 11:59PM
Total Points: 10.0

In this assignment, you will work with your group to build a GPT2-style autoregressive language model. A Transformer codebase is provided along with some starter code. The Transformer code base is derived from a blog by Samuel Lynn-Evans (see https://towardsdatascience.com/how-to-code-the-transformer-in-pytorch-24db27c8f9ec). This codebase is a complete Transformer architecture (e.g., encoder and decoder) that performs English-to-French translation.

You will modify this architecture to form a decoder-only autoregressive language model. Modules related to dataset preprocessing, beam search and translation have already been eliminated. The starter code includes modules to tokenize the training and test corpora, call the training loop and call the evaluation of the final model. You may discuss the homework with other groups but do not take any written record from the discussions. Also, do not copy any additional source code from the Web. You are encouraged to perform this assignment in a command-line Linux environment. MiniConda and PyTorch are compatible with this codebase.

Steps to complete the homework

1. (3.0 points) Modify the codebase to create a GPT2-style autoregressive language model. You will need to eliminate cross-attention, tie the source and target vocabularies into a single vocabulary, build a data feeder capable of handling arbitrary batch sizes, and construct your own training and test modules, among other tasks. You are encouraged to code your own multi-class cross-entropy loss function, but it is okay to use F.cross_entropy(). You should use loss.backward() and optimizer.step() to train your model. You may also change the command line parameters for the model. Discuss all changes you make to the Transformer codebase, referencing line numbers where possible.
2. (5.0 points) Train your GPT2-style autoregressive language model on the Wikitext-2 corpus for 20 epochs and provide learning curves for the training and validation sets, as well as the final perplexity of the test set. You should report all metrics as perplexity. You should use a model with an embedding dimension of 512, 8 attention heads, 6 layers, a sequence length of 512, a dropout rate of 0.1 and tied embedding spaces. This configuration results in a model of approximately 71M parameters, trains for about 40 minutes and attains a training and test perplexity of about 400 and 200, respectively. You may use additional hyper-parameters as you see fit. Please describe all hyper-parameters used for training and provide a brief justification for each. Lastly, please give the training script and final weights that we can use to run your model. Hint: Start with a much smaller model when debugging.
3. (1.0 points) Modify the model built in Step #2 to replace the dot-product in the attention head with a different distance metric (e.g., Euclidean distance). Retrain your model and provide new learning curves and final test set perplexity. Comment on your results.
4. (1.0 points) Modify the model built in Step #2 to eliminate the residual connect accompanying the feed-forward module. Retrain your model and provide new learning curves and final test set perplexity. Comment on your results.

Submission Instructions

Turn in your homework as a single zip file in Canvas. The zip file should include the code for your GPT2-style autoregressive language model, the scripts used to run them and a PDF file of your results/write-up.

Good luck, and have fun!