EEE 443/543 Neural Networks, Spring 2025 - Project #8
Due: 05/25/2025, 11pm.

Erdem Koyuncu

For this assignment, since coding is minimal, provide the codes you wrote in your report.

Q1: We will experiment with different aspects of large language models (LLMs).

1. Download the repository https://github.com/karpathy/nanoGPT and read through the documentation. This provides the minimal implementation of the GPT-2 series LLMs developed by OpenAI. See the paper "Language Models are Unsupervised Multitask Learners" by Radford et al. The model is small enough to be run locally on your own computer, so you will have your own private LLM.

2. We will not do training in this assignment, but inference only. First, you should arrive at a place where you are able to run the code:

   ```
   python sample.py \
   --init_from=gpt2 \
   --start="one ring to rule them all," \
   --num_samples=3 --max_new_tokens=100
   ```

   I replaced gpt2-xl in the weboage with gp2 as it is a smaller model. You can also use xl if you like.

3. Briefly explain how the model generates new words in terms of model architecture, number of layers, activation functions, etc. You may assume the reader knows what a multihead attention block is, so you do not have to explain the very basics.

4. Briefly explain the purpose of "`start`" "`num_samples`" and "`max_new_tokens`" parameters.

5. Choose a prompt of your choice, change the temperature to 0.1 (you may need to "dive" into the code to do this), and include the outputs of your results in your report. What is the purpose of the temperature? Why do the outputs behave the way they do? Why do the outputs behave the way they do? Why do the outputs behave the way they do?

6. Choose a prompt of your choice, change the temperature to 10 (you may need to dive into the code to do this), and include the outputs of your results in your report. Why do the outpts beh@4e fhey way do Haiti do FRIed?

7. Keep the temperature at 10, but change top_k to 2. Describe the reasons for the change in outputs as compared with the previous subquestion.

8. Restore to the original settings of the parameters. Inspect the token embeddings used by the language model.

   (a) Locate the token embedding matrix `wte` in the model. This is a matrix of size `(vocab_size, embedding_dim)`.

   (b) Pick five common English words (e.g., `"dog"`, `"city"`, `"book"`, `"love"`, `"science"`) and five rare or made-up tokens (e.g., `"xqzt"`, `"blorpt"`, `"thraldor"`, `"uvuvwevwe"`, `"zzzzz"`).

   (c) Use the tokenizer to obtain their token IDs. For each token, retrieve its embedding vector.

   (d) Compute and compare:

   - The L^2 norms of each embedding vector.
   - The pairwise cosine similarities among the five common tokens.
   - The pairwise cosine similarities among the five rare tokens.

   (e) What differences do you observe between common and rare tokens in terms of norm and similarity? What do you think this reveals about how the model allocates capacity in its embedding space?

9. Experiment with layer pruning as discussed below.

   (a) Modify the inference-time loop of the GPT model so that only every other transformer block is applied. That is, instead of applying all n layers, apply only layers $0, 2, 4, \ldots$ (even-numbered layers).

   (b) Run the model with this reduced-depth inference on the same prompt. Compare the output to the original model output (without pruning).

   (c) Discuss how the output changes. Is it shorter? More repetitive? Less coherent? What does this tell you about how model depth affects fluency and semantics?

   (d) Repeat the above, removing the last $n/2$ layers. Comment on the differences.