Homework 3

11-967

Due date: 02/07/2025 11:59 PM EST

In this homework, you will experiment with generation and perplexity evaluation using an open-source model available in the HuggingFace library.

*Note: there are **two** Gradescope submission pages, one for your code and one for your `report.pdf`. Please submit your code to **Assignment** as instructed, and your `report.pdf` to **Assignment (PDF)**.*

Problem 1: Text Generation & Perplexity Analysis

In this problem, you will learn how to use your build-from-scratch model, as well as a pre-trained model from HuggingFace to do text generation. You will use your final model from HW2 and the Pythia-1.4B model from HuggingFace. Outlines of the codes for text generation are provided for you in `generate.py` and `use_pythia.py` under `src/lm/`.

[Question 1.1] (***Coding, 10 points***) To get generation working on your own model, there are two functions that you are expected to implement in `generate.py`:

1. `generate` - given a `DecoderLM` and a list of prompts, generate tokens and compute perplexity of the generated text.
2. `softmax_with_temperature` - given a set of logits and a temperature, transform them into probabilities with the softmax function and temperature annealing.

Complete the two functions. A unit test is provided in the test script `test_generate.py`.

DELIVERABLES FOR Q1.1

Upload your code to Gradescope as instructed in the starter code README and ensure all test cases pass. We may also run additional tests after the submission deadline.

[Question 1.2] (***Written, 15 points***) We have provided you code for generating continuations and calculating perplexity with Pythia-1.4B in `use_pythia.py` and `evaluate_perplexity.py` under `src/lm/`.

Choose three prefixes by taking the first sentence to paragraph from a recent news article, blog post, or other text source. You may also choose from the 5 examples we have provided in `data/prefixes.jsonl` or get creative and write your prefixes.

For each of your three prefixes, you should experiment with performing at least 32 tokens of generation using both your own model and Pythia-1.4B, using different temperature values.

DELIVERABLES FOR Q1.2

In your report, answer the following:

A. Report the prefixes you chose and the generation hyperparameter settings (`temperature` and `max_new_tokens`) you chose. Copy and paste the generations of the two models separately.

B. What happens to the generations if the temperature is near zero, or near one? Answer in at most two sentences.

C. Under the same temperature value, compare the quality and perplexity scores of generations between the two models. Report your findings and possible reasons behind them in at most three sentences.

[Question 1.3] (***Written, 10 points***) So far, you have computed perplexity of a validation set drawn from C4 and of a handful of generations. In this question, you will compute perplexity according to `Pythia-1.4B` of a few other text sources.

Start by running `data/make_ppl_example_jsonl.py` which creates a `jsonl` file with two documents in it: (1) a few paragraphs from the Wikipedia article about CMU and (2) a famous nonsense poem by English writer Lewis Carrol. Use `evaluate_perplexity.py` to compute the perplexity of each document.

DELIVERABLES FOR Q1.3

Answer the following questions in at most 3 sentences each.

A. What is the perplexity of the CMU Wikipedia paragraphs? What happens to perplexity if you remove every instance of "the" in the text? Provide an explanation for the observed perplexity change.

B. Write a mimic poem by swapping several of the nonsense word in each stanza of the Jabberwocky to a different nonsense word. For example, you might replace "mimsy" with "flimty" or "borogoves" with "kittipoos". Report the perplexity of the original Jabberwocky poem and your own version. Despite both poems being filled with made-up nonsense words, one should have significantly lower perplexity than the other. Give a reason why this could be the case.