Fall 2024, CS 4973 / CS 6983

Homework 1
Instructor: Alina Oprea
TA: Pravin Anand Pawar

Due date: September 24, at 11:59pm

Instructions

- Submit a PDF writeup named ”%LASTNAME%_HW1.pdf” in Gradescope. You can save your notebook as PDF, but please make sure that you include text answer to the questions. Include a link to your code in the PDF.
- For the models, you can use the HuggingFace library, and for your code you can use PyTorch or TensorFlow. We are also open about other Python frameworks, but please discuss with the instructors first if you plan to use different libraries.
- You can find some starter code and the datasets for this assignment at: https://drive.google.com/drive/folders/1q2d11l0FUZrVhUhGuRNYh1RekgMGEgi2?usp=sharing.

Course policy on collaboration and cheating:

- You may discuss the concepts with your classmates, but write up the answers entirely on your own.
- You cannot share your code with your classmates.
- You can use generative AI tools for assistance, but not for code generation.
- You cannot use code from the Internet or generative AI tools for your assignment.
- You can post questions on Piazza and are encouraged to come to the TA and Instructor office hours.

Problem 1 [Comparing multiple LLMs] 30 points

In this problem, you will evaluate multiple versions of the HuggingFace GPT2 model (small, medium, and large) according to the perplexity metric on the sample of the wikitext dataset provided.

(a) Load a sample of the wikitext test dataset we provided and print some statistics on the number of samples and average number of characters per sample. Print the first 5 samples.

(b) Compute the perplexity metric by sentence using a window length of 1024, for each of the three models.

Output the 25th and the 75th percentiles of the perplexity, the midhinge and the median perplexity. Plot a histogram of perplexities.

(c) Write some observations on how the perplexities of the three models compare.

For the GPT-2 large model, find 3 sentences with the lowest and highest perplexity, print them and explain why these sentences have high or low perplexity.

Problem 2 [Generating text with LLMs] 35 points

In this problem, you will try and compare several strategies for text generation using the GPT2 large model. You will consider the following text generation methods:

- Greedy decoding: This strategy selects the word with the highest probability as its next word given the context.
- Top-k: This strategy samples from the top-k tokens according to their predicted probabilities given the context. Use a value of $k = 10$.
- Beam search: This strategy maintains a number of hypotheses (called beams) and selects the one with the highest probability. Set the number of beams to 3.

(a) For the dataset we provided, select the first 10 tokens for each sentence as prefix and run the generation algorithm for 50 tokens.

(b) The MAUVE score is a statistical measure of two distributions. It can be used to compare how far the text written by a model is from the distribution of human text. You can find more details at: `https://huggingface.co/spaces/evaluate-metric/mauve`.

Compute the MAUVE score between the generated sentences and the ground truth for the three generation strategies. Discuss your observations on how the 3 strategies compare.

(c) Select the sentence of the highest and lowest MAUVE score for each generation strategy and print it. Discuss your observations.

Problem 3 [Fine-tuning BERT for sentiment classification] - 35 points

In this problem you will use a small pre-trained BERT model and fine-tune it for a sentiment classification task. You will use the DistilBERT model and a sample from the IMDB reviews dataset.

(a) Fine-tune the pre-trained DistilBERT model on 1000 samples selected from the IMDB reviews dataset. Freeze all layers, except the last classification layer when fine-tuning.

(b) Output the following metrics on 1000 samples selected from the testing set:

1. Train and test cross-entropy loss;
2. Accuracy, Error;
3. Precision, Recall, F1 score;
4. Plot a ROC curve of the classifier and report the AUC metric.

(c) Fine-tune all the model parameters with the same 1000 samples. Output the same metrics and compare them with the model fine-tuned on the last layer only.