Exercise Sheet 3: ICL, Summarization, and Evaluation

Relevant code: ICL, Continued_training, Summarization:CO5 repo LINK

In this exercise, you will experiment with several methods to adapt LLMs for specific use cases. You will work through three sets of exercises:

1. In-Context Learning (ICL)

2. Continue pretraining an LLM

3. Patient report summarization and generative evaluation

Overview

Exercise 1: In-Context Learning (ICL)

- Experiment with **Tree of Thought (ToT)** and **Graph of Thought (GoT)** methodologies to prompt LLMs for answering complex questions.

Exercise 2: Continue Pretraining an LLM

- Fine-tune an LLM to generate text similar to characters in **Shakespeare plays**.

Exercise 3: Patient Report Summarization and Generative Evaluation

- Fine-tune an LLM to generate summaries of PubMed articles in the form of abstracts and **objectively evaluate** them using **Bert-Score**.

- **Dataset:** SumPubMed (~32,000 abstracts and corresponding short abstracts).

- Steps:
  1. Preprocess the dataset and create train (100 samples) and test sets (10 samples).

  2. Fine-tune an LLM on the training data.

  3. Generate summaries for the test set and compute Bert-Score between generated and true summaries. Report the **mean Bert-Score**.

Exercise 1 — In-Context Learning (ICL)

1. Use **Ollama**. Install locally or request an API endpoint from your GPU cluster maintainers.

2. Update the `.env` file with:

```
OLLAMA_HOST="http://x.x.x.x:x/api/chat"
OLLAMA_MODEL="x"
```

3. Run the `ToT.py` and `GoT.py` scripts and experiment with their behavior.

Exercise 2 — Continue Pretraining an LLM

1. Use the provided `.txt` files to generate training samples.

2. Modify the function `format_dataset_lm` according to the instructions in the code.

3. When the dataset object is created (line 49), fine-tune the selected LLM.

4. Generate text in **Shakespearean style**.

5. Experiment with hyperparameters and observe their effects on generated text.

Exercise 3 — Patient Report Summarization and Generative Evaluation

1. Use the code in the `Summarization` folder in the course GitHub repo.

2. Download a quantized LLM model:

```
huggingface-cli download unsloth/Llama-3.2-3B-Instruct-bnb-4bit --local-dir ./models/Llama-3
```

You can check all available model HERE

3. Clone the dataset:

```
git clone https://github.com/vgupta123/sumpubmed
```

- Update `sum_pubmed_base_path` accordingly.

- Dataset folders:
  - `abstract` → source text

  - `short_abstract` → target text

4. Sample **without replacement**: 100 samples for training, 10 for test (seed=1000).

5. Update `summary.py` to:
   - Train the model on the training data.
   - Generate summaries for the test set.
   - Compute **Bert-Score** between generated and true summaries.
6. Experiment with hyperparameters:
   - Rank of Low Rank Matrix (r)
   - `lora_alpha`
   - Number of epochs
   - Base LLM with more parameters

Congratulations

You successfully adapted a general-purpose LLM for **bioinformatics, medical, and playwright text genres**. While commercial LLMs like GPT-5 often outperform smaller models, fine-tuning small models and using ICL are competitive methods when **privacy** or **limited infrastructure** is a concern.