Assignment 4

Parameter Efficient Supervised Fine-Tuning of a Pretrained Language Model (Total: 100 Points)

Objectives

In this assignment, you will explore the process of supervised fine-tuning on a small pretrained language model originally trained with the next-token prediction (autoregressive) and not adapted with any post-training modifications or chat-specific templates. You will research parameter-efficient fine-tuning approaches, select an appropriate dataset from Hugging Face, and evaluate how fine-tuning influences the model's behavior.

1. Background and Supervised Fine-Tuning

- Research and describe the concept of supervised fine-tuning.
- Explain the characteristics of a base model that has been pretrained with next-token prediction and has not undergone additional post-training adjustments. Demonstrate your claim with an example.
- Compare standard next-token prediction training with supervised fine-tuning methodologies.

2. Parameter Efficient Fine-Tuning

- Investigate parameter-efficient fine-tuning methods (e.g., techniques like low-rank adaptation or similar strategies).
- Summarize the key ideas behind these methods in your own words, focusing on their benefits and potential limitations.
- Discuss why a parameter-efficient approach is particularly beneficial for fine-tuning language models in this assignment.

3. Dataset Selection and Preparation

- Choose a dataset that is specifically suitable for supervised fine-tuning of conversational models. Examples include datasets like HuggingFaceTB/smol-smoltalk, bigcode/the-stack-smol, or similar collections.
- Provide an overview of the chosen dataset, emphasizing its conversational nature. Include details on how the data incorporates chat templates and how the tokenizer is expected to handle conversation-specific formatting.
- Outline the necessary preprocessing procedures, such as tokenization tailored to conversational data, encoding strategies for dialogue context, and any adjustments required to maintain chat template structure.

4. Model Fine-Tuning with Supervised Learning

- Select a small base language model (with 1B parameter or less). For instance, consider models such as smollm or qwen 2.5 available from Hugging Face.
- Detail how you will integrate a parameter-efficient fine-tuning method (e.g., LoRA) into the supervised fine-tuning process.
- Describe any modifications made to accommodate the conversational format, including adjustments to model input handling and ensuring that the tokenizer effectively processes chat templates.

5. Training and Evaluation

- Outline the supervised fine-tuning procedure using the chosen dataset and parameter-efficient approach. Include training configurations such as learning rate, batch size, and any additional hyperparameters.
- Evaluate the fine-tuned model by comparing its performance before and after fine-tuning. Focus on metrics that capture conversational quality and response accuracy.
- Provide an in-depth analysis of the model's behavior in conversation scenarios. Discuss improvements in handling chat templates and how the tokenizer's treatment of conversational inputs changes the model's outputs. Highlight any observed limitations or challenges in adapting the base model to a supervised fine-tuning setting.

Submission Guidelines

- Submit all code as a Google Colab notebook, ensuring the link is shared in the submission description with **Anyone with the link** access. Additionally, upload the notebook file and include detailed explanations and visualizations, either within the notebook as markdown cells or in a separate PDF report. Avoid sharing .py or ZIP files for this assignment.
- Clearly label each part and question in your submissions.
- Do not attach any datasets; instead, provide the link to them on Hugging Face in your notebook.
- **Deadline:** April 2th, 2025

Rubric and Expectations

- **Code Quality and Functionality:** Code should be well-organized, modular, and thoroughly commented. Demonstrate effective use of the Hugging Face ecosystem, ensuring that the integration of parameter-efficient fine-tuning techniques (e.g., LoRA) is clear and reproducible. Ensure that the preprocessing pipeline correctly handles conversational data, including chat templates and tokenizer-specific adjustments.
- **Performance Metrics and Benchmarking:** Accurately report evaluation metrics that reflect conversational quality and model performance before and after fine-tuning. Benchmark results against appropriate baselines (e.g., the performance of the base model without fine-tuning) and, where applicable, reference published work.
- **Visualization and Interpretability:** Provide clear visualizations (such as attention heatmaps, response comparisons, or other interpretable representations) to illustrate model behavior and improvements. Interpret these visualizations to explain how parameter-efficient fine-tuning has affected the handling of conversational inputs and chat templates.
- **Analysis and Reflection:** Offer a comprehensive analysis discussing the methodology, challenges encountered during fine-tuning, and insights gained from the experiments. Reflect on the differences in model behavior pre- and post-fine-tuning, including the impact on conversation handling and the tokenizer's treatment of chat templates. Propose potential improvements or alternative strategies for further enhancing model performance.
- **Adherence to Guidelines:** Submissions must strictly follow the provided guidelines and formatting instructions. Ensure all required sections are addressed, and the work is submitted by the deadline.