CSIE5431 ADL HW3

陳禹翰 B10705050

November 5, 2024

Q1: LLM Tuning

Describe:

- How much training data did you use? (2%)

  I used 5000 samples for training. The training data were sampled randomly to avoid only good performance on `public_test.json` yet poor performance on `private_test.json`.

- How did you tune your model? (2%)

  1. **4-bit Quantization with BitsAndBytes:**
     I applied 4-bit quantization using the BitsAndBytes library to reduce GPU memory usage while maintaining model performance. Specifically, I used the "nf4" quantization type and enabled double quantization with computation in `bfloat16`. This allowed me to efficiently work with the large model ('zake7749/gemma-2-2b-it-chinese-kyara-dpo').

  2. **Low-Rank Adaptation (LoRA):**
     Instead of full model fine-tuning, I applied Low-Rank Adaptation (LoRA), which allowed me to fine-tune only a subset of parameters. The LoRA configuration included:

     * Rank (r) of 8, which determines the dimensionality of the low-rank adaptation.
     * LoRA alpha of 32 to adjust the learning rate scaling.
     * Dropout of 0.1 to help prevent overfitting during adaptation.

     This approach significantly reduced the number of trainable parameters, making fine-tuning computationally efficient, especially for a large language model.

  3. **Dataset Subsampling for Training:**
     I selected 5000 random examples from the training dataset to reduce training time and focus on a smaller but representative set of data. This subset was created using randomly selected indices, providing a balance between training efficiency and model generalization.

  4. **Data Collator for Sequence-to-Sequence Tasks:**
     I used the `DataCollatorForSeq2Seq` from the `transformers` library to handle data preparation. This collator takes care of padding the sequences dynamically, ensuring that each batch has consistent dimensions while minimizing padding tokens. This helps to make training more efficient.

5. **Learning Rate Scheduling:**
I employed a linear learning rate scheduler with no warm-up steps to adjust the learning rate over the training process. The learning rate started at a fixed value (5×10^{-5}) and decayed linearly, which helps to achieve more stable convergence by reducing the learning rate gradually.

6. **Optimizer Setup:**
I used AdamW as my optimizer, which is commonly used for training transformer models due to its ability to apply weight decay, thereby helping to prevent overfitting. The learning rate was set to 5×10^{-5} to balance between fast convergence and stability.

7. **GPU Memory Management:**
After each epoch, I called `torch.cuda.empty_cache()` and `gc.collect()` to clear GPU memory and release resources. This was crucial for handling the large model and preventing out-of-memory (OOM) errors during training and validation.

8. **Batch-Wise Progress Tracking:**
I used `tqdm` progress bars to track the progress of both training and validation steps. This provided visibility into the current state of training, including the batch loss, making it easier to identify issues and monitor convergence.

9. **Validation and Perplexity Evaluation:**
After each training epoch, I evaluated the model on a public test dataset. I calculated the validation loss to monitor overfitting and computed the perplexity score on the validation data to gauge the quality of the language model's predictions.

10. **Model Checkpointing:**
I saved the model and tokenizer checkpoints after each epoch to ensure that I could resume training if needed and evaluate different versions of the model. This also allowed me to later use the best checkpoint for inference.

- What hyper-parameters did you use? (2%)

  * **Quantization Configuration:**
    - Load in 4-bit quantization: `True`
    - Quantization type: `nf4`
    - Use double quantization: `True`
    - Compute data type: `bfloat16`
  * **LoRA Configuration:**
    - Rank (r): 8
    - LoRA Alpha: 32
    - LoRA Dropout: 0.1
    - Bias: `none`
    - Task Type: `CAUSAL_LM`
  * **Training Parameters:**
    - Learning Rate: 5×10^{-5}
    - Batch Size: 4

· Number of Epochs: 1

* **Learning Rate Scheduler:**

  · Scheduler Type: `linear`

  · Number of Warm-up Steps: 0

* **Optimizer:**

  · Optimizer Type: `AdamW`

Show your performance:

- What is the final performance of your model on the public testing set? (2%)

  My model has a performance of approximately 16.80 perplexity on the public testing set.

- Plot the learning curve on the public testing set (2%)

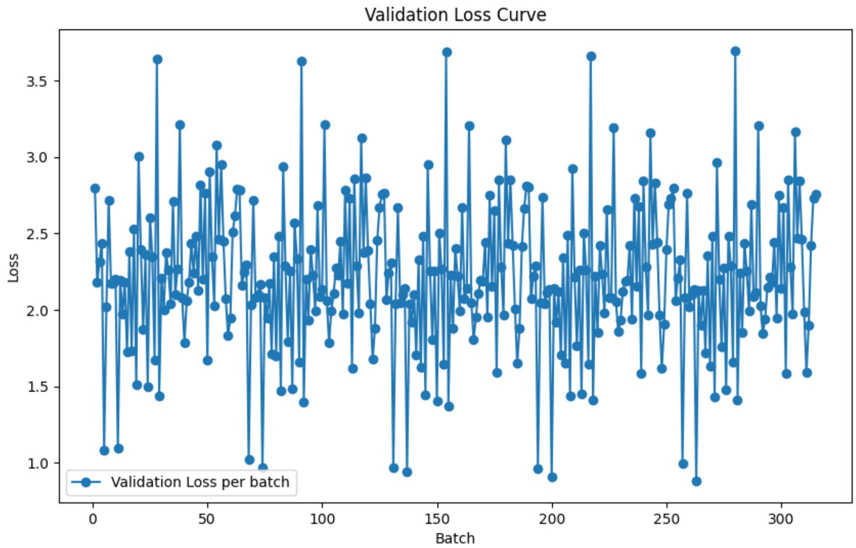

Figure 1: Validation Loss Per Batch

Q2: LLM Inference Strategies

Note: Please conduct zero-shot and few-shot experiments on Orginal Model that has not been fine-tuned with QLoRA

Zero-Shot

- What is your setting? How did you design your prompt? (1%)

  – **Setting:**
  In the zero-shot experiment, I used the original model (`zake7749/gemma-2-2b-it-chinese-kyara-dpo`) without any further fine-tuning or training. The model was evaluated on 10 randomly selected instructions from the

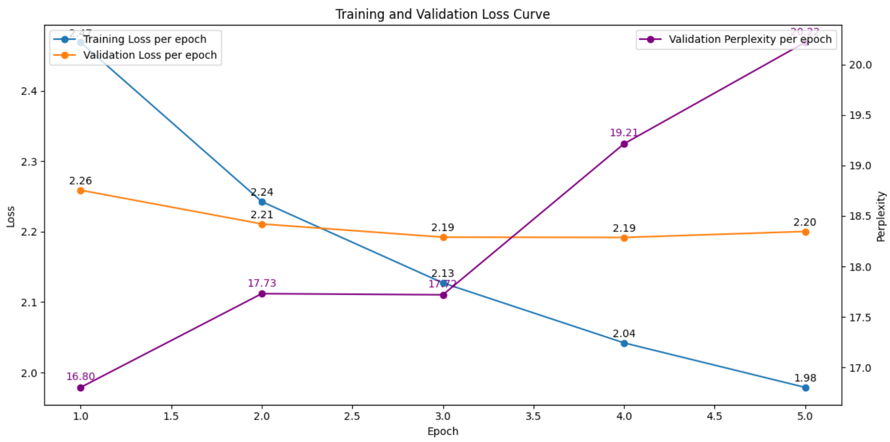

Figure 2: Validation Loss and Perplexity Per Epoch

private_test.json dataset to assess its ability to generalize from pre-trained knowledge.

- **Prompt Design:**
  The prompt was designed to clearly instruct the model to translate between Classical and Modern Chinese in both directions. The prompt was structured as follows:

  請你扮演一個人工智慧國文助理，幫助用戶在白話文和文言文之間轉換
  你被給予的指示中會有明確要求要轉換成文言文還是白話文
  如果這個要求在句首，請用這種格式回答：答案：文言文答案/白話文答案
  如果這個要求在句尾，請直接回答：文言文答案/白話文答案
  以下是你被給予的指示：
  USER:{instruction}
  ASSISTANT:

  This prompt explicitly defined the task for the model and set clear expectations for the response format, ensuring that the model knew whether to translate to Classical or Modern Chinese.

Few-Shot (In-context Learning)

- What is your setting? How did you design your prompt? (1%)
  - **Setting:** For the few-shot experiment, I utilized the original model in an in-context learning setup. The prompt included several examples of translations between Classical and Modern Chinese, followed by the new instruction to help the model understand the expected task better. The few-shot experiment was also conducted on the same 10 randomly selected instructions as the zero-shot experiment.
  - **Prompt Design:** The prompt contained three in-context examples of input-output pairs, illustrating the conversion between Classical and Modern Chinese. The examples included both directions (Modern to Classical and Classical to Modern) to provide a comprehensive context for the model. The prompt was structured as follows:

    請你扮演一個人工智慧國文助理，幫助用戶在白話文和文言文之間轉換
    你被給予的指示中會有明確要求要轉換成文言文還是白話文
    如果這個要求在句首，請用這種格式回答：答案：文言文答案/白話文答案

如果這個要求在句尾，請直接回答：文言文答案/白話文答案
以下是幾個範例：
範例一：
USER:翻譯成文言文：\n於是，廢帝讓瀋慶之的堂侄、直將軍瀋攸之賜瀋慶之毒藥，命瀋慶之自殺。
ASSISTANT:帝乃使慶之從父兄子直閣將軍攸之賜慶之藥。
範例二：
USER:文言文翻譯：\n靈鑒忽臨，忻歡交集，乃迴燈拂席以延之。
ASSISTANT:答案：靈仙忽然光臨，趙旭歡欣交集，於是他就把燈點亮，拂拭乾淨床席來延請仙女。
範例三：
USER:希望您以後留意，不要再齣這樣的事，你的小女兒病就會好。\n這句話在古代怎麼說：
ASSISTANT:以後幸長官留意，勿令如此。
現在是你被給予的指示：
USER:{instruction}
ASSISTANT:

This prompt structure helped the model understand the expected format of the response and the nuances of translating different sentence structures.

- How many in-context examples are utilized? How you select them? (1%)

  I utilized three in-context examples in each few-shot prompt. These examples were carefully chosen from the public testing set to:

  * Cover both translation directions (Modern to Classical and Classical to Modern).
  * Include varied sentence structures to represent the diversity of the language.
  * Provide enough context to improve the model's understanding of the task without exceeding token limits.

  By selecting examples that represented typical translation tasks, I ensured that the model could generalize well to the new instructions provided in the prompt.

Comparison

- What's the difference between the results of zero-shot, few-shot, and LoRA? (2%)

  I have tested 10 samples from the private testing set to compare the results from the three different strategies.

  - **Case 1: d573ddd1-7bb9-468d-b906-e392223d9579**

    Instruction: "建武元年，拜為禦史中丞。\n把這句話翻譯成文言文："

Strategy	Output
LoRA Output	建武元年，拜禦史中丞。
Zero-Shot Output	建武元年，拜禦史中丞。
Few-Shot Output	建武元年，拜禦史中丞。

Table 1: Comparison of Outputs for Case 1

  - **Case 2: e3c475ca-f2b2-4450-af6d-675e646c2488**

Instruction: “文言文翻譯：\n上嘉納之。”

Strategy	Output
LoRA Output	答案：上嘉納之。
Zero-Shot Output	答案：上嘉之。
Few-Shot Output	答案：上嘉納之。

Table 2: Comparison of Outputs for Case 2

– **Case 3: 7efea98b-646a-4bd8-b85c-0118d3493506**

Instruction: “翻譯成文言文：\n親自耕作的原因，就是錶示重視祭祀的供品。\n答案：”

Strategy	Output
LoRA Output	親自耕作，以示重視祭品。
Zero-Shot Output	親自耕作，以示重視祭祀。
Few-Shot Output	親自耕作，以示重視祭祀的供品。

Table 3: Comparison of Outputs for Case 3

– **Case 4: bdb5da27-91b9-428d-ba77-921a53e174f4**

Instruction: “翻譯成現代文：\n四十五年，五大夫賁攻韓，取十城。”

Strategy	Output
LoRA Output	四十五年，五大夫賁攻韓，奪取十座城池。
Zero-Shot Output	四十五年，五大夫賁攻韓，取十城。
Few-Shot Output	答案：四十五年，五大夫賁攻韓，取十城。

Table 4: Comparison of Outputs for Case 4

– **Case 5: 7cfccb6e-ab89-4ae6-b38a-065ed249b12d**

Instruction: “陳後主說：那就把毛喜安置在一個小郡中，不許他再在朝廷參預政事。\n這句話在中國古代怎麼說：”

Strategy	Output
LoRA Output	陳後主曰：遂置毛喜於小郡，不許參政。
Zero-Shot Output	陳後主曰：將毛喜安置於小郡，不許參政。
Few-Shot Output	陳後主曰：遂將毛喜安置於小郡，不許參政。

Table 5: Comparison of Outputs for Case 5

– **Case 6: 134dca67-ff20-4582-a6aa-7610a0c00822**

Instruction: “此一難忍也。\n翻譯成白話文：”

Strategy	Output
LoRA Output	這是難以忍受的。
Zero-Shot Output	難以忍耐。
Few-Shot Output	希望您以後留意，不要再齣這樣的事，您的小女兒病就會好。

Table 6: Comparison of Outputs for Case 6

– **Case 7: 1c494ae3-1df0-4ce8-ae4f-3857864dec25**

Instruction: "到崇政殿受冊就夠瞭！元宵節燈宴，太後的母親應當入宮，太後製止說：夫人登樓，皇上一定增加禮儀，這樣由於我的原因而超越典章製度，於心很不安啊！隻令賜給她燈燭，於是每年以此為常例。\n這句話在古代怎麼說："

Strategy	Output
LoRA Output	到崇政殿受冊就足矣！元宵節燈宴，太後母親應入宮，太後製止曰：夫人登樓，皇上必增加禮儀，因我而超越典章，於心不安。隻賜燈燭，每年以此為常例。
Zero-Shot Output	到崇政殿受冊就足矣！元宵節燈宴，太後母親應入宮，太後製止說：夫人登樓，皇上必增加禮儀，故因我的原因而超越典章製度，於心不安！隻令賜燈燭，每年以此為常例。
Few-Shot Output	元宵節燈宴，太後的母親應當入宮，太後製止說：夫人登樓，皇上一定增加禮儀，這樣由於我的原因而超越典章製度，於心很不安啊！隻令賜給她燈燭，於是每年以此為常例。

Table 7: Comparison of Outputs for Case 7

– **Case 8: 2c0caaf9-904a-425f-b516-de4dd438d66c**

Instruction: "以己之心為根據來要求彆人，把自己的好惡也用到彆人身上。\n幫我把這句話翻譯成文言文"

Strategy	Output
LoRA Output	以己之心為根據，要求彆人，把自己的好惡也用彆人。
Zero-Shot Output	以己之心為根據，要求彆人，把自己的好惡也用於彆人。
Few-Shot Output	以己心為之，以己惡為之。

Table 8: Comparison of Outputs for Case 8

– **Case 9: 364fee8a-45bc-4f58-92d9-cafc8085fe09**

Instruction: "人或鑿崖取之，即風雷為變。\n翻譯成白話文："

Strategy	Output
LoRA Output	有的人鑿崖取之，即風雷為變。
Zero-Shot Output	人或鑿崖取之，即風雷為變。
Few-Shot Output	人或鑿崖取之，即風雷為變。

Table 9: Comparison of Outputs for Case 9

– **Case 10: 91650fd4-6b91-474a-b301-9e0187d4f9a3**

Instruction: "翻譯成文言文：\n她高壽多少？\n答案："

Strategy	Output
LoRA Output	高壽何？
Zero-Shot Output	高壽多少？
Few-Shot Output	高壽多少？

Table 10: Comparison of Outputs for Case 10

The performance of the language model varied significantly across the three different inference strategies: Zero-Shot, Few-Shot, and LoRA fine-tuning. Below is a detailed analysis of the differences in results for these three approaches:

* **Zero-Shot**
    - In the zero-shot setting, the model generated outputs that were often similar to the original instruction without any adaptation. This method lacked context-specific adjustments, resulting in less nuanced translations.
    - The output was frequently concise but did not always capture all the subtleties of Classical or Modern Chinese, especially when more complex or indirect phrases were used. Some translations appeared less formal or lacked precision, which indicated a limitation in handling nuanced translation tasks.
* **Few-Shot (In-Context Learning)**
    - By providing several in-context examples, the model was able to generate translations that better reflected the expected structure and style. The few-shot approach showed notable improvement over the zero-shot approach, particularly in maintaining linguistic formality and capturing more nuanced expressions.
    - The few-shot outputs demonstrated that the examples provided significantly influenced the model's translation accuracy. However, the model was still limited by the quality and diversity of the examples. In some cases, the output appeared slightly inconsistent when compared to the expected answer.
* **LoRA Fine-Tuning**
    - LoRA fine-tuning yielded the best results among the three strategies. The model, having been trained specifically for the task, provided consistently high-quality translations that accurately reflected both Classical and Modern Chinese, retaining the intended meaning and style.
    - The LoRA model's outputs were more concise and formal, which is characteristic of Classical Chinese. Additionally, the model was able to correctly convert both simple and complex sentences, even in cases where zero-shot or few-shot settings produced incomplete or inaccurate results.

- · LoRA fine-tuning allowed the model to adapt more effectively to the specifics of the task, demonstrating a clear improvement in accuracy, fluency, and overall translation quality compared to the other approaches.

Summary of Observed Differences

- ∗ **Accuracy:**
  The LoRA approach significantly improved the accuracy of translations compared to zero-shot and few-shot, with fewer mistakes and omissions in the translations.
- ∗ **Contextual Understanding:**
  Few-shot learning provided better contextual understanding than zero-shot due to in-context examples. However, it still fell short of the performance achieved by the LoRA fine-tuned model.
- ∗ **Formal Style and Nuance:**
  LoRA fine-tuning effectively captured the formal and nuanced style required for Classical Chinese, whereas zero-shot often struggled, and few-shot only partially succeeded.