HW3

id: r12922192

name: 邱冠珅

Q1: LLM Tuning

Describe:

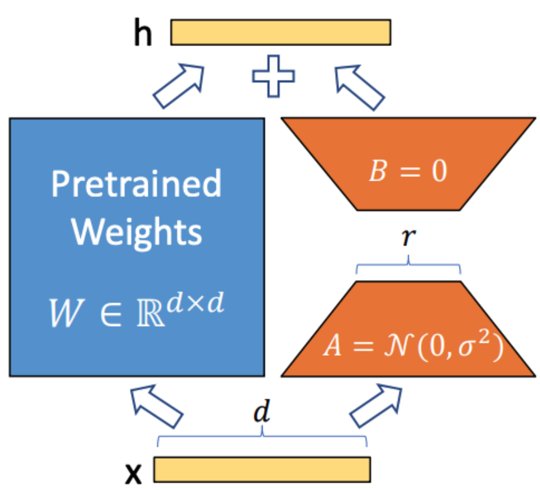
- How much training data did you use? (2%)

  I use all the data provided by the TA, which consists of about 10,000 data.

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

  Reference from the QLoRA: Efficient Finetuning of Quantized LLMs:

  To begin, load the pretrained model that has been quantized by NF4 (4-bit Normal float). This quantization reduces the size of the model. Next, add smaller A and B parameters (as shown in the graph below) compared to the pretrained weights. Finally, fine-tune this assembled model by forwarding the model and adding the output of these two components. During backpropagation, freeze the original parameters and only propagate the A and B parameters. This process equips the assembled model with both the language ability from the pretrained model and specific knowledge to convert Classical Chinese and vernacular Chinese.



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

```
//training
"auto_mapping": null,
"bias": "none",
"fan_in_fan_out": false,
"inference_mode": true,
"init_lora_weights": true,
"layers_pattern": null,
"layers_to_transform": null,
"lora_alpha": 16,
"lora_dropout": 0.0,
"modules_to_save": null,
"peft_type": "LORA",
```

```
"r": 64,
"revision": null,
"target_modules": [
  "k_proj",
  "gate_proj",
  "q_proj",
  "down_proj",
  "up_proj",
  "v_proj",
  "o_proj"
],
"task_type": "CAUSAL_LM",
"load_in_8bit": false,
"load_in_4bit": true,
"llm_int8_threshold": 6.0,
"llm_int8_skip_modules": None,
"llm_int8_enable_fp32_cpu_offload": false,
"llm_int8_has_fp16_weight": false,
"bnb_4bit_quant_type": nf4,
"bnb_4bit_use_double_quant": true,
"bnb_4bit_compute_dtype": float32,
"gradient_accumulation_steps": 16,
"num_train_epochs": 5,
"weight_decay": 0.0,
"learning_rate": 0.0002,
"max_grad_norm": 0.3,
"gradient_checkpointing": True,
"lr_scheduler_type": 'constant',
"warmup_ratio": 0.03,
```

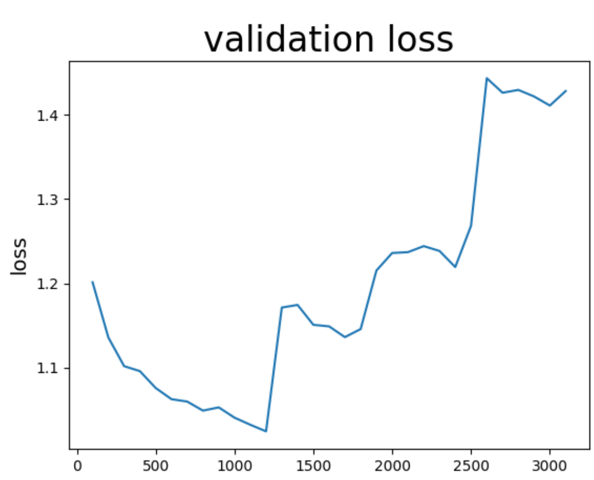
Show your performance:

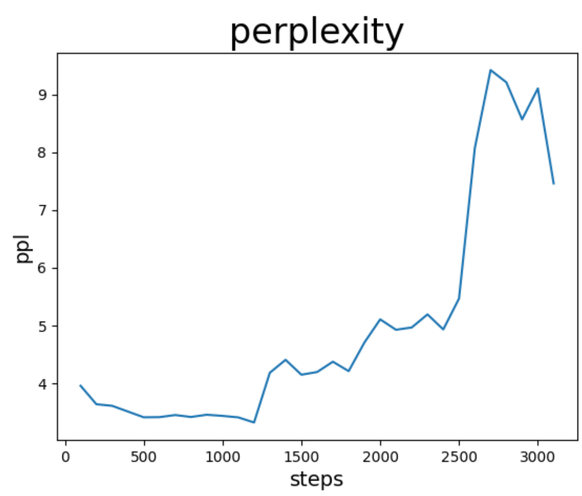
- What is the final performance of your model on the public testing set? (2%)

  public testing test

  Mean perplexity: 3.323672766685486.

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





Q2: LLM Inference Strategies

Zero-Shot

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

▼ setting:

```
'num_beams': 1,
'num_beam_groups': 1,
'penalty_alpha': None,
'use_cache': True,
'temperature': 1.0,
'top_k': 50,
'top_p': 1.0,
'typical_p': 1.0,
'diversity_penalty': 0.0,
'repetition_penalty': 1.0,
'length_penalty': 1.0,
'no_repeat_ngram_size': 0,
```

- prompt_1:

  你是人工智慧助理，以下是用戶和人工智能助理之間的對話。你要對用戶的問題提供有用、安全、詳細和禮貌的回答。接下來要做翻譯任務，請根據USER指令進行文言文及白話文的轉換，其中文言文是在中國古代語言模式，白話文又稱為現代文是現代的語言模式。USER: {instruction} ASSISTANT:”

- result_1:

  Mean perplexity: 5.141525848388672

- prompt_2:

你是人工智慧助理，以下是用戶和人工智能助理之間的對話。你要對用戶的問題提供有用、安全、詳細和禮貌的回答。USER: {instruction} ASSISTANT:

- result_2:

  Mean perplexity: 5.454097266197205

▼ Comparison:

Use the two prompts mentioned above. Prompt_2 is the original prompt from the TA, and Prompt_1 is created by me to inform the model about the ongoing task. By analyzing the results, we can determine that the prompt which provides detailed instructions can achieve a lower perplexity.

Few-Shot (In-context Learning)

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

▼ setting:

```
'num_beams': 1,
'num_beam_groups': 1,
'penalty_alpha': None,
'use_cache': True,
'temperature': 1.0,
'top_k': 50,
'top_p': 1.0,
'typical_p': 1.0,
'diversity_penalty': 0.0,
'repetition_penalty': 1.0,
'length_penalty': 1.0,
'no_repeat_ngram_size': 0,
```

- prompt:

  你是人工智慧助理，以下是用戶和人工智能助理之間的對話。你要對用戶的問題提供有用、安全、詳細和禮貌的回答。接下來要做翻譯任務，請根據USER指令進行文言文及白話文的轉換， 其中文言文是在中國古代語言模式，白話文又稱為現代文是現代的語言模式 \
  請學習以下翻譯例子進行轉換 \
  將“正月，甲子朔，鼕至，太後享通天宮；赦天下，改元”翻譯成現代文。 答案：聖曆元年正月，甲子朔，鼕至，太後在通天宮祭祀；大赦天下，更改年號。 \
  將“雅裏惱怒地說： 從前在福山田獵時，你誣陷獵官，現在又說這種話。”翻譯成文言文。 答案：雅裏怒曰： 昔畋於福山，卿誣獵官，今復有此言。\
  將下麵句子翻譯成文言文：令、錄、簿、尉等職官有年老病重的人允許彈勃。 答案：令、錄、簿、尉諸職官有耄耋篤疾者舉劾之。 \
  翻譯成現代文：\n士匄請見，弗內。答案：士匄請求進見，荀偃不接見。 \

  USER: {instruction} ASSISTANT:

- result:

Mean perplexity: 4.686557821273803

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

  I use 4 in-context examples to ask the model to generate an answer based on them. First, I select two examples in vernacular and Classical Chinese respectively. Then, I observe that the testing data has very short sentences to convert, so I choose the other two short examples to demonstrate the conversion for short sentences.

  After testing different numbers of in-context examples, here is the comparison table.

numbers of examples	perplexity
1	4.94330560874939
2	4.834038286209107
3	4.736983351707458
4	4.686557821273803
5	4.738565093517304

Comparison:

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

  The **zero-shot** instruction model is used to convert vernacular and Classical Chinese without any examples or performance measures, relying solely on the pretrained model's comprehension.

  Similar to zero-shot, **few-shot** models also rely on pretrained models. However, few-shot models use examples as instructions and ask the model to generate answers based on these examples as references.

  Unlike zero-shot and few-shot models, **LoRA** fine-tunes the pretrained model by freezing the original parameters and adding a smaller number of trainable parameters to match the task. As a result, LoRA is able to achieve higher performance.

	pretrained model	example	fine tune
zero shot	v		
few shot	v	v	
LoRa	v	v	v

Q3: Bonus: Other methods (2%)

Choose one of the following tasks for implementation.

- Experiments with different PLMs

  PLM: FlagAlpha/Llama2-Chinese-7b-Chat

  Similar to TA's model, the FlagAlpha/Llama2-Chinese-7b-Chat model is based on Meta/LLaMa-2. It is fine-tuned using data crawled from the Internet, Chinese

Wikipedia, WuDao, Clue, and MNBVC to enhance its ability in the Chinese language.

Unlike TA's model, this model utilizes a significant amount of Simplified Chinese training data.

Describe your experimental settings and compare the results to those obtained from your original methods.

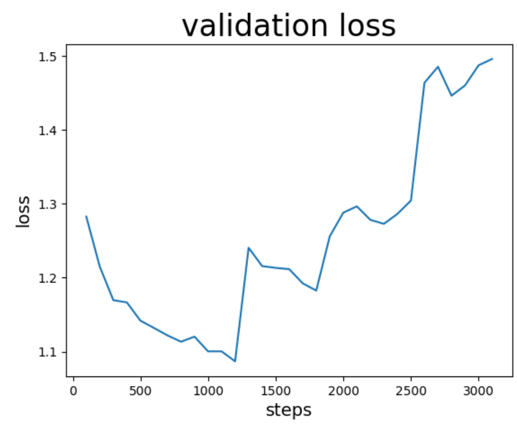
▼ Setting

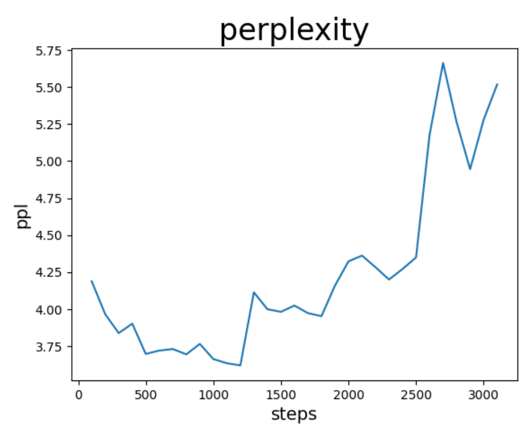
```
//training
"auto_mapping": null,
"bias": "none",
"fan_in_fan_out": false,
"inference_mode": true,
"init_lora_weights": true,
"layers_pattern": null,
"layers_to_transform": null,
"lora_alpha": 16,
"lora_dropout": 0.0,
"modules_to_save": null,
"peft_type": "LORA",
"r": 64,
"revision": null,
"target_modules": [
  "k_proj",
  "gate_proj",
  "q_proj",
  "down_proj",
  "up_proj",
  "v_proj",
  "o_proj"
],
"task_type": "CAUSAL_LM",
"load_in_8bit": false,
"load_in_4bit": true,
"llm_int8_threshold": 6.0,
"llm_int8_skip_modules": None,
"llm_int8_enable_fp32_cpu_offload": false,
"llm_int8_has_fp16_weight": false,
"bnb_4bit_quant_type": nf4,
"bnb_4bit_use_double_quant": true,
"bnb_4bit_compute_dtype": float32,
"gradient_accumulation_steps": 16,
"num_train_epochs": 5,
"weight_decay": 0.0,
"learning_rate": 0.0002,
"max_grad_norm": 0.3,
"gradient_checkpointing": True,
"lr_scheduler_type": 'constant',
"warmup_ratio": 0.03,
```

▼ Comparison

FlagAlpha/Llama2-Chinese-7b-Chat(QLoRA)

smallest mean perplexity at 1200 steps: 3.6209425139427185





(recall): Taiwan LLaMa Mean perplexity: 3.323672766685486

Compared to Taiwan LLaMa, FlagAlpha/Llama2-Chinese-7b-Chat has lower performance. I speculate that the Simplified Chinese data used to fine-tune LLama 2 and obtain FlagAlpha/Llama2-Chinese-7b-Chat is the main cause of the performance gap.