112-1 ADL HW3 Report

`b10902138 陳德維`

Q1: LLM Tuning

- Describe

- ### Training Data:
  - I randomly picked out `4000` training data out from `train.json` using `./preprocess.py` with seed `1006` and use `5%` of them to do evaluation check with seed `1006`.
- ### Finetuning Method:
  - I use `qlora.yml` from `OpenAccess-AI-Collective/axolotl` to tune my model. `QLoRa`, `Quantized LLMs with Low-Rank Adapters`, it uses these techniques to save memory without sacrificing the performance including `4-bit NormalFloat Quantization`, `Double Quantization`, and `Paged Optimizers`.
- ### Hyper-parameters:

```
base_model: ./Taiwan-LLM-7B-v2.0-chat
model_type: LlamaForCausalLM
tokenizer_type: LlamaTokenizer
is_llama_derived_model: true

load_in_8bit: false
load_in_4bit: true
strict: false

seed: 1006
datasets:
        - path: ./data/random_train.json
          ds_type: json
          type: alpaca
val_set_size: 0.05
output_dir: ./trained_model

adapter: qlora
```

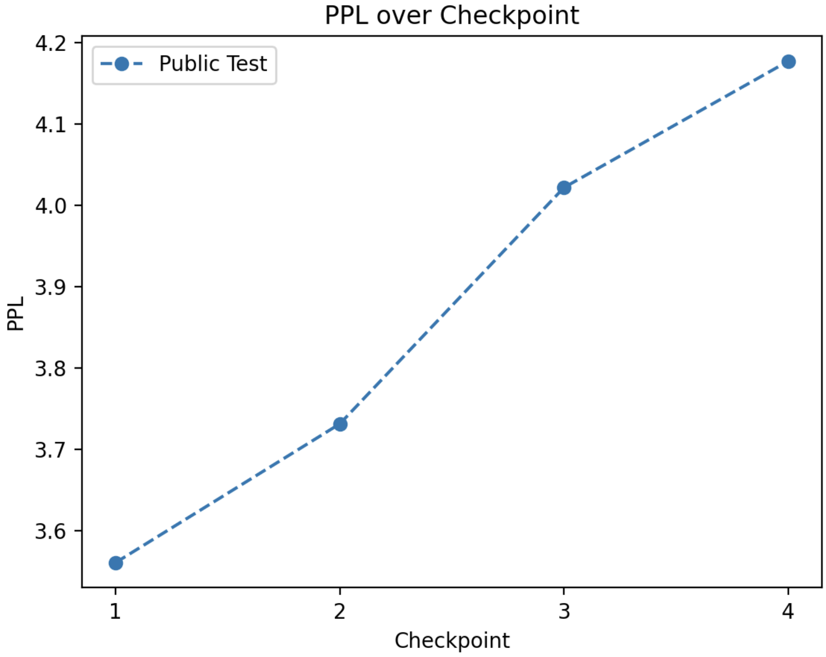
```
sequence_len: 2048
sample_packing: true
pad_to_sequence_len: true

lora_r: 4
lora_alpha: 16
lora_dropout: 0.05
lora_target_linear: true

gradient_accumulation_steps: 4
micro_batch_size: 2
num_epochs: 5
optimizer: paged_adamw_32bit
lr_scheduler: cosine
learning_rate: 0.0002

train_on_inputs: false
group_by_length: false
bf16: true
fp16: false
tf32: false

gradient_checkpointing: true
logging_steps: 1
flash_attention: true

warmup_steps: 10
eval_steps: 0.05
weight_decay: 0.0

special_tokens:
        bos_token: "<s>"
        eos_token: "</s>"
        unk_token: "<unk>"
```

- Performance

- ### Inference 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。請進行文言文到現代文或現代文到文言文的翻譯。USER: {instruction} ASSISTANT:`
- ### BNB Config:

```
config = BitsAndBytesConfig(
        load_in_4bit=True,
        bnb_4bit_quant_type="nf4",
        bnb_4bit_compute_dtype=torch.bfloat16,
        bnb_4bit_use_double_quant=True,
)
```

- **Public Testing Test:**
  - Best performance: `Mean perplexity: 3.561242261886597 (From checkpoint 1)`
- **Learning Curve:**



```
- Checkpoint 1: `Mean perplexity: 3.561242261886597`
- Checkpoint 2: `Mean perplexity: 3.731644229412079`
- Checkpoint 3: `Mean perplexity: 4.02179647838592`
- Checkpoint 4: `Mean perplexity: 4.176944113254547`
```

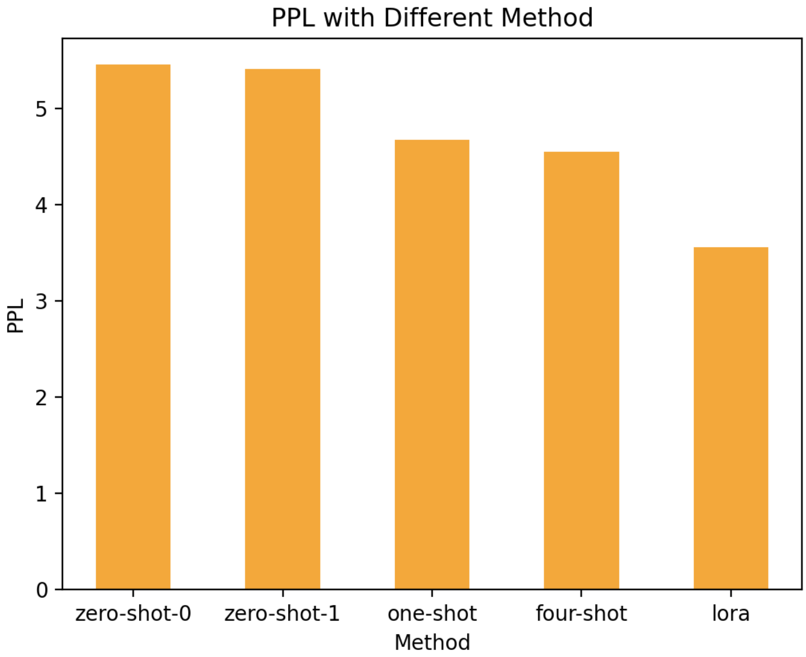
Q2: LLM Inference Strategies:

- **- Zero-Shot**
  - **Setting 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。USER: {instruction} ASSISTANT:`
    - **How I design?**
      - This is from the sample code.

    - **Performance**:
        - `Mean perplexity: 5.452863416671753`
  - **Setting 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。請進行文言文到現代文或現代文到文言文的翻譯。USER: {instruction} ASSISTANT:`
    - **How I design**?
        - I simply add a little bit hint about what is going to happen.
    - **Performance**:
        - `Mean perplexity: 5.412987493515015`
- **- Few-Shot**
  - **Setting 1 (One Shot)**
    - If currently doing `文言文 -> 白話文`:
        - `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。請使用以下文本作為少量示例，指導模型進行文言文到現代文。USER: 辛未，命吳堅為左丞相兼樞密使，常楙參知政事。\n把這句話翻譯成現代文。 ASSISTANT: 初五，命令吳堅為左承相兼樞密使，常增為參知政事。 USER: {instruction} ASSISTANT:`
    - else doing `白話文 -> 文言文`:
        - `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。請使用以下文本作為少量示例，指導模型進行白話文到文言文。USER: 她不僅手巧，擅長女紅，而且體態輕盈，相貌皎潔。\n幫我把這句話翻譯成文言文 ASSISTANT: 善工巧，體貌輕潔。 USER: {instruction} ASSISTANT:`
    - **How I design?**
        - First, I try to determine that this instruction wants me to do which way of translation, and I sort out some prompt including:
            - 文言文 -> 白話文
                - `翻譯成現代文`
                - `翻譯成白話文`
                - `文言文翻譯`
            - 白話文 -> 文言文
                - `翻譯成文言文`
                - `翻譯成古文`
                - `中國古代怎麼說`
        - According to this, I use the corresponding prompt from the top.
    - **Performance**:

        - `Mean perplexity: 4.6702891345024105`
- **Setting 2 (4 shot)**
    - If currently doing `文言文 -> 白話文`, append these to the prompt:
        - `USER: 辛未，命吳堅為左丞相兼樞密使，常楙參知政事。\n把這句話翻譯成現代文。 ASSISTANT: 初五，命令吳堅為左承相兼樞密使，常增為參知政事。`
        - `USER: 翻譯成白話文：\n王申，以保忠為定難軍節度使。\n答案：ASSISTANT: 十六日，任命趙保忠為定難軍節度使。`
        - `USER: 文言文翻譯：\n賈逵、張衡、蔡邕、王蕃、陸績皆以北極紐星之樞，是不動處。 ASSISTANT: 答案：賈逵、張衡、蔡邕、王蕃、陸續都認為北極紐星的樞紐，是不移動的地方。`
        - `USER: 將下麵句子翻譯成現代文：\n公正嗟服。還，具言之於武帝，帝大欽重之。 ASSISTANT: 尹公正非常佩服，迴國後把這些事情都告訴瞭周武帝，周武帝十分欽敬看重熊安生。`
    - else doing `白話文 -> 文言文`:
        - `USER: 翻譯成文言文：\n有鄰跟隨差役去見閻王說： 有人告你的狀說，不待殺死，就活生生的取齣它的腎。 ASSISTANT: 有鄰隨吏見王，王雲： 有訴君雲，不待殺瞭，生取其腎。`
        - `USER: 她不僅手巧，擅長女紅，而且體態輕盈，相貌皎潔。\n幫我把這句話翻譯成文言文 ASSISTANT: 善工巧，體貌輕潔。`
        - `USER: 唐朝元和年間，博陵人崔玨，從汝鄭來，僑居在長安延福裏。\n翻譯成古文： ASSISTANT: 元和中，博陵崔玨者，自汝鄭來，僑居長安延福裏。`
        - `USER: 於是對二公說： 祥瑞應該依德而至，災異也因政而生。\n這句話在中國古代怎麼說： ASSISTANT: 乃言於二公曰： 夫瑞應依德而至，災異緣政而生。`
    - **How I design?**
        - Same as above, instead for this case, I give 4 examples. I choose these examples from `train.json` which all have different way to ask. (ex. Ask in prefix/suffix, the way it ask...)
    - **Performance**:
        - `Mean perplexity: 4.547110088825226`

- Comparison



- **Performance**:
  - Zero-Shot 1:
    - `Mean perplexity: 5.452863416671753`
  - Zero-Shot 2:
    - `Mean perplexity: 5.412987493515015`
  - One-Shot:
    - `Mean perplexity: 4.6702891345024105`
  - Four-Shot:
    - `Mean perplexity: 4.547110088825226`
  - LoRa:
    - `Mean perplexity: 3.561242261886597`
- As we can see, the performance of zero-shot to four-shot has improved as expected. However, there are still some gap between fine-tuning with QLoRa and in-context learning.

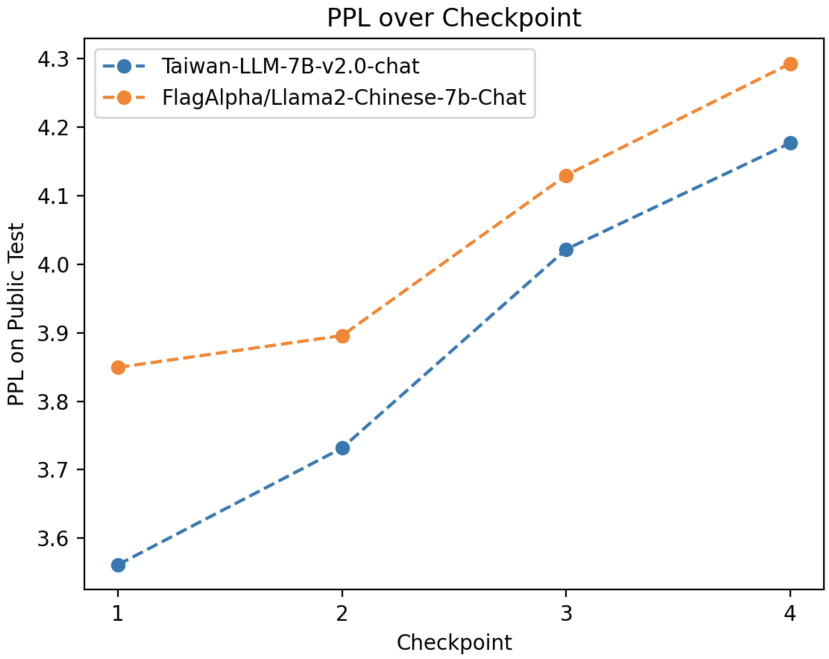
Q3: Bonus

- I choose another **PLM**:
  - `FlagAlpha/Llama2-Chinese-7b-Chat`
- **Training hyper-parameters:**

```
base_model: FlagAlpha/Llama2-Chinese-7b-Chat
model_type: LlamaForCausalLM
tokenizer_type: LlamaTokenizer
is_llama_derived_model: true
```

```yaml
load_in_8bit: false
load_in_4bit: true
strict: false

seed: 1006
datasets:
        - path: ./data/random_train.json
          ds_type: json
          type: alpaca
val_set_size: 0.05
output_dir: ./trained_model

adapter: qlora
sequence_len: 2048
sample_packing: true
pad_to_sequence_len: true

lora_r: 4
lora_alpha: 16
lora_dropout: 0.05
lora_target_linear: true

gradient_accumulation_steps: 4
micro_batch_size: 2
num_epochs: 5
optimizer: paged_adamw_32bit
lr_scheduler: cosine
learning_rate: 0.0002

train_on_inputs: false
group_by_length: false
bf16: true
fp16: false
tf32: false

gradient_checkpointing: true
logging_steps: 1
flash_attention: true

warmup_steps: 10
eval_steps: 0.05
weight_decay: 0.0

special_tokens:
        bos_token: "<s>"
        eos_token: "</s>"
        unk_token: "<unk>"
```

- **Inference Settings:**
    - Inference 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。請進行文言文到現代文或現代文到文言文的翻譯。USER: {instruction} ASSISTANT:`
    - BNB config:
        - Same as Q1.
- **Performance:**



- Checkpoint 1:
- `Mean perplexity: 3.849330807685852`
- Checkpoint 2:
- `Mean perplexity: 3.8958204884529115`
- Checkpoint 3:
- `Mean perplexity: 4.129890115261078`
- Checkpoint 4:
- `Mean perplexity: 4.292995451927185`

- As we can see, the original model preforms better, but both model pass the baseline!