NTU ADL hw3 Report

Q1: LLM Tuning

Describe

1. **How much training data did you use?**

> I used 3000 samples for training.
> I actually tried for all samples training, but the training time was too long, it took me more than one day even with fp16 qlora.
> So then I emailed TA for reference, and finally used 3000 samples for training.
> It turns out that 3000 samples already enough for lora.

2. **How did you tune your model?**

> I used qlora training code to train this homework.
> I slighly modify the code to run it successfully.
> I trained the model for 3 epochs, however, the model started overfit just after 400 stpes.

3. **What hyper-parameters did you use?**

> Here’s the hyper paramters I used.

```
--model_name_or_path Taiwan-LLM-7B-v2.0-chat \
--output_dir ./output/Taiwan-LLM-7B-v2.0-chat \
--logging_steps 10 \
--save_strategy steps \
--data_seed 42 \
--save_steps 400 \
--save_total_limit 40 \
--evaluation_strategy steps \
--eval_steps 100 \
--eval_dataset_size 1000 \
--max_eval_samples 500 \
--per_device_eval_batch_size 1 \
--max_new_tokens 32 \
--dataloader_num_workers 8 \
--group_by_length \
--logging_strategy steps \
--remove_unused_columns False \
--do_train \
--do_eval \
--lora_r 64 \
--lora_alpha 16 \
--lora_modules all \
--double_quant \
--quant_type nf4 \
--fp16 \
--bits 4 \
--warmup_ratio 0.03 \
--lr_scheduler_type constant \
--gradient_checkpointing \
--dataset ./data \
--dataset_format alpaca \
--source_max_len 256 \
--target_max_len 64 \
--per_device_train_batch_size 1 \
--gradient_accumulation_steps 32 \
--max_steps 2000 \
--eval_steps 150 \
--learning_rate 0.0002 \
--adam_beta2 0.999 \
--max_grad_norm 0.3 \
--lora_dropout 0.05 \
--weight_decay 0.0 \
--seed 0
```

> Some are derived from the original qlora repo, some are modified by myself after some experiments.

Show your performance

1. **What is the final performace of your model on the public testing set?**

> At step 400, the model reached public testing ppl score of 3.523.
> The eval loss reached the lowest value of 1.028 at step 600.

2. **Plot the learning curve on the public testing set**

> Below is the eval/loss graph plotted with wandb
>
> 
>

Q2: LLM Inference Strategies

Zero-Shot

1. **What is your setting? How did you design your prompt?**

I used beam search with beam size 4 for generation config and Taiwan-LLM-7B-v2.0-chat as the PLM.
The prompt is quite simple. Original I used english prompt and find out that it didn't perform well.
So then I translated the prompt into traditional-chinese with the translation task specified.
The fine-grained zero-shot result I got is ppl score of 4.966. The result shows that PLM doesn't fully understand how to do this job.
I tried two prompts, one specified the translation task, one does not. The former performs better by a margin.
P1: f"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。"
P2: f"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文言文與白話文之間的翻譯回答。"

Few-Shot (In-context learning)

1. **What is your setting? How did you design your prompt?**

I used beam search with beam size 4 for generation config and Taiwan-llama-2 as the PLM.
In the prompt, I added few samples to let the PLM learn whats the 文言文 and how is it different from 白話文.
The prompt is showed below.

```
def get_prompt(instruction: str, few_shot: bool) -> str:
  '''Format the instruction as a prompt for LLM.'''
  # p = f"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
  p = f"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
  if few_shot:
      p += f"文言文又稱古文。現代文又稱白話文。"
      p += f"文言文：雅裏怒曰： 昔畋於福山，卿誣獵官，今復有此言。 現代文：雅
  p += f"USER: {instruction} ASSISTANT:"
  return p
```

2. **How many in-context examples are utilized? How you select them?**

I tried out one/two in-context samples and got the result 4.75/ ppl score for one sample/two sapmles respectively.
Based on this result,

Comparison

1. **What’s the difference between the results of zero-shot, few-shot, and LoRA?**

> For the unfinetuned methods, one-shot performs best. however among these three methods, all of them cannot complete the task effectively.
> As below table dipicts lora reached ppl score of 3.46, which is the best of all these methods.
> For the running time it only increased nearly 1% from 122 seconds to 129 seconds, but the score is from 4.9 down to 3.46.
> In conclusion, for this find of downstream task, using lora can effectively and efficiently helped.

method	ppl
zero-shot	4.96
one-shot	4.75
two-shot	4.8
lora	3.46
lora w/one-shot	3.47

Q3: Bonus - Other methods

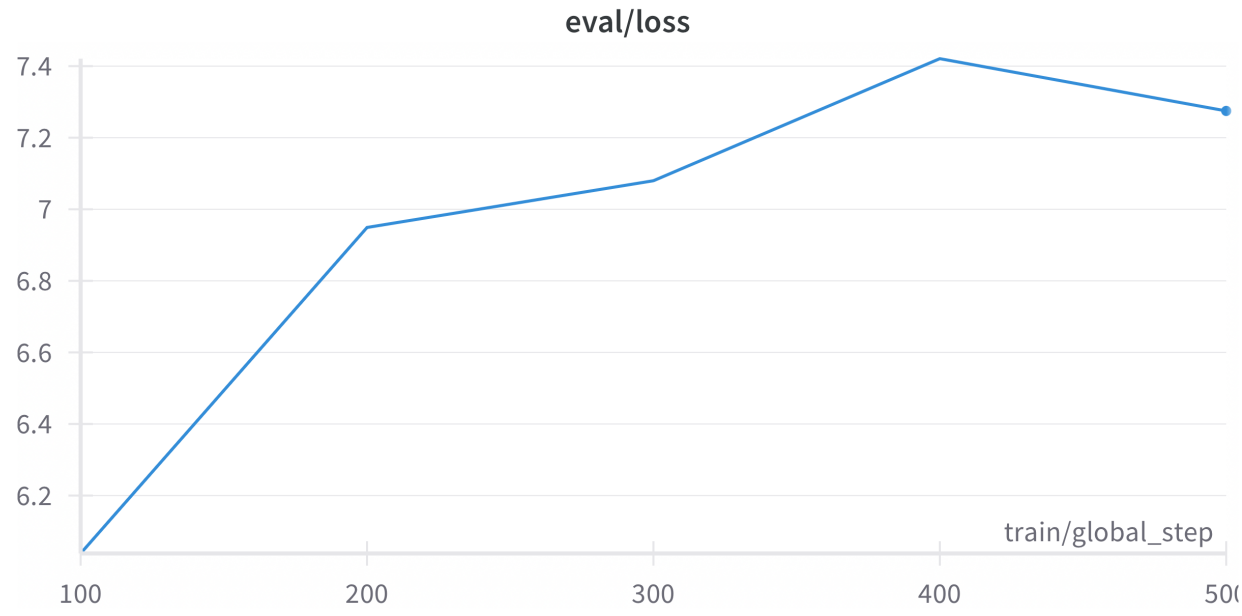
Choose one of the following tasks of implementation

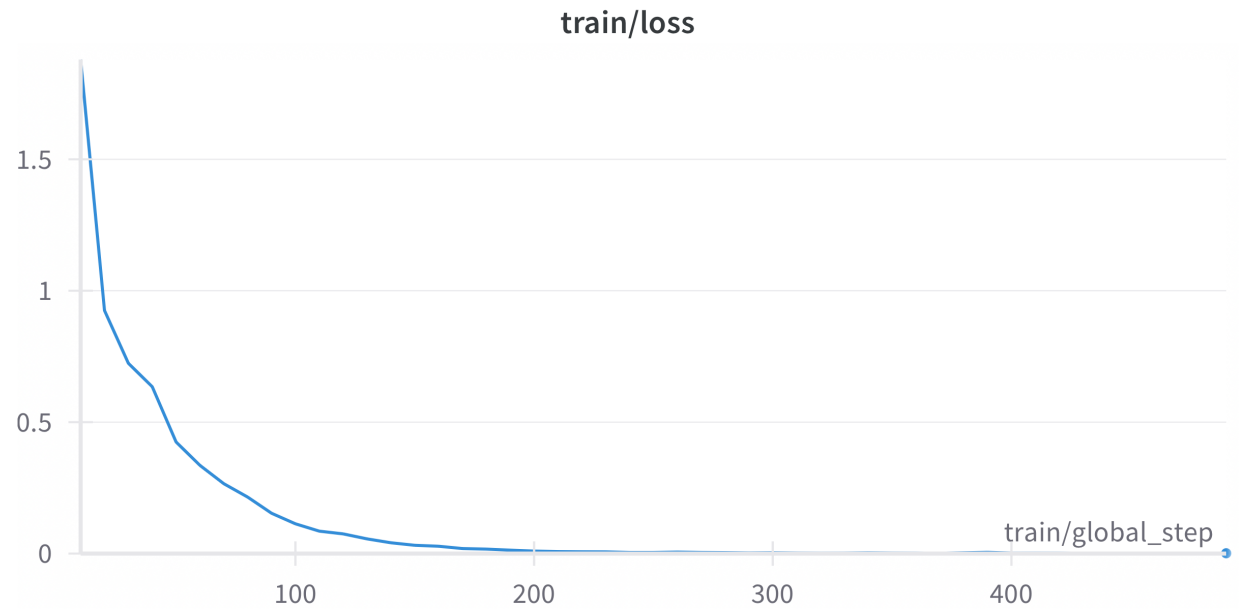
1. **Experiments with different PLMs**

> I tried out some different PLMs, including ckip/bert-base-han-chinese and ckip/bert-case-chinese.
> Both of them are the PLM from Academia Sinica.
> They were trained for fill-mask task on traditional chinese and ancient traditional chinese.

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

Unfortunately, I kept encountered tokenizer problems with ckip/bert-base-han-chinese PLM, but after changing it to ckip/bert-base-chinese the training was ran successfully.
The training settings I used was same as the using Taiwan-LLM-chat PLM.
The training loss decreased very fast, but the eval loss does not.





I think its because the ckip model is not trained for generating texts, it does not has the ability to understand the question and generate correspond answers.