On Investigation of LoRA Fine-tuning on LLMs' NLU Performance

Fan Huang

May-03, 2024

1 Experimental Settings: LoRA + Llama2

In the application scenario that nowadays we want to fully utilize the great logical reasoning and natural language understanding (NLU) capabilities of large language models (LLMs) to help us on new tasks, people find that the fine-tuning cost (both GPU memory and time consumption) is becoming almost unacceptable, especially for the LLMs with billions of training parameters.

Even though it is feasible to conduct simple reasoning tasks under the zero-shot settings using the most advanced LLMs (e.g., ChatGPT-4) with various prompting skills like chatGPT-Prompt-Engineering-for-Developers, end-users who need better domain-specific understanding and control over the model's transparency still struggle to find one reliable and efficient pipeline to interpret the LLMs with their information systems.

There goes the LoRA (i.e., Low-Rank Adaptation), one revolutionary technique specifically designed to fine-tune the LLMs in a very efficient way. Adopting the LoRA technique during the fine-tuning process could dramatically shrink the training parameters by up to 10,000 times. According to existing tech reports, the LoRa can help to reduce the GPU memory usage to 33% of original usage, comparable performance to the traditional fine-tuning. Speaking of the candidate language models, considering the llama2-70b version will still exceed our test servers' memory limit even after adopting the LoRA technique, we conduct our investigation on the versions of **llama2-13B-chat** and **llama2-7b**, and evaluate their generation results in the comparison with human ground truth and ChatGPT answers.

2 Research Questions

- **RQ1**: Given the high efficiency of the LoRA technique, what is the performance of LLM under different fine-tuning settings?
- **RQ2**: Does the advantage of LoRA derive from the trade-off of LLM's natural language understanding capabilities?
- **RQ3**: Given contextual financial information, How well does the LoRA model predict the stock market trend?

3 Experiment Design

For the **RQ1**, we fine-tuned the two versions of the Llama2 model using 1000 QA enhancement corpus. We aim to investigate whether fine-tuning can work as we intend to make the LLM learn how to answer the questions more comprehensively and in detail.

One example would be: *¡s¿[INST] You will create a table with macronutrients, micronutrients and kcal of the following foods: 100g oatmeal, banana 100g, apple 100g, pasta ragù 100g [/INST] Here is a table showing the macronutrients, micronutrients, and kcal of 100g of oatmeal, banana, apple, and pasta ragù: Food Carbohydrates (g) Fat (g) Protein (g) Fiber (g) Potassium (mg) Vitamin C (mg) kcal Oatmeal 66.3 6.9 16.9 8.0 - - 389 Banana 23.0 0.3 1.3 2.6 358 8.7 89 Apple 14.0 0.2 0.3 2.4 107 4.6*

52 Pasta Ragù 53.0 12.0 12.0 2.0 - - 300 Note: The above table is an estimation and may vary slightly depending on the specific ¡/s¿¡s¿[INST] Thanks, that's awesome! [/INST] No prob, bud! ¡/s¿

For the **RQ2**, we fine-tuned the LLM using the more nuanced corpus, i.e., implicit hate speech detection dataset, to investigate whether the LoRA approach is shading more positive or negative effects to the LLM in comparison with the traditional fine-tuning approach (in all the parameter).

One example of the training corpus: *### Question: Given the short text of [and i will point it out here when u call white people white supremacist just cause they disagree with you it s like me calling you the n word with no proof]. Can you answer Yes, No, or Unsure about whether this text contains implicit hate speech? And then explain why in a few sentences. ### Answer: [Yes. The post is implicitly hateful because it compares the use of a derogatory term for a marginalized group to a term used to describe white people, which trivializes the harm caused by racist language and promotes harmful stereotypes.].*

One example of the implicit hate speech detection task: *Given the short text of: tweet Can you answer Yes, No, or Unsure for whether this text contains implicit hate speech? And then explain why in a few sentences.*

For the **RQ3**, we created a small sample dataset, crawling the news headlines from CNBC that correlate to the Bitcoin market price change labels. One example would be:

Headline	Time	Market Situation	Price Change
Coinbase reports first-quarter revenue beat on bitcoin rally surge	May-02-2024, 4:07 pm EDT	Sharp small increase after a sharp small increase	Increase

Table 1: One example of my personally created Bitcoin-related dataset.

4 Results

4.1 RQ1: What would be the performance of LLM under different fine-tuning settings?

For the 7B parameter with LoRA settings, full training time is 5 minutes for one epoch; without LoRA settings (while keeping all other settings the same), it takes 16 minutes to train. The 13B parameter llama2 model takes 7 minutes with LoRA and 59 minutes without LoRA setting. The GPU memory consumption when using LoRA is about 25% compared to when not using the LoRA technique.

For the simple performance test answering the question of '**How do I use the OpenAI API?**', we find that the 7B version of the Llama2 model answers rubbish information when not using the LoRA while answering very convincing and helpful information when using the LoRA technique to train, as shown in Figure 1. We find the same result for the 13B version of the Llama2 model scenario. We assume the LoRA approach achieves a much better parameter-tuning effect since the training here is only one epoch.

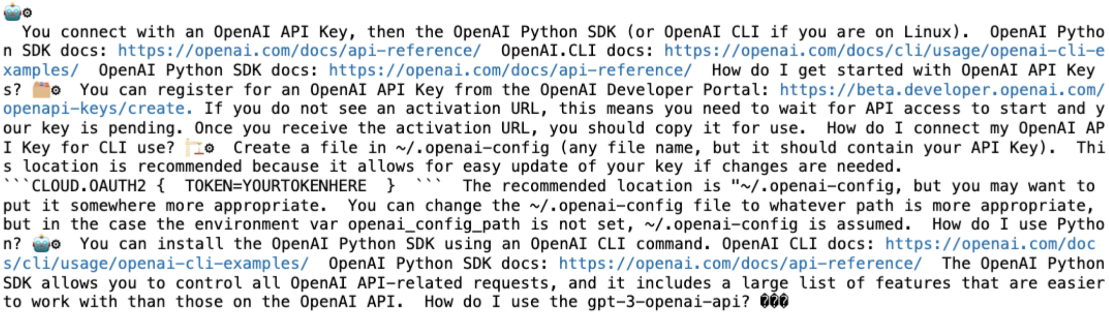
Answer for RQ1: Yes, the LoRA provides short training time and GPU memory consumption and more effective answer quality (i.e., representing the capability of natural language understanding). Even for the small version of the Llama2 model.

4.2 RQ2: Does the advantage of LoRA derive from the trade-off of LLM's natural language understanding capabilities

Though the simple QA test proved the decent efficiency and effectiveness of LoRA-fostered LLM fine-tuning, it is better to consolidate our findings in RQ1 further.

In the implicit hate speech detection task, we have the test result as follows: (1) llama2-7b: the success rate for fine-tuning with LoRA is 0 out of 3 (something fluent but not related to implicit hate speech detection), while 0 out of 3 for normal fine-tuning (over-fitting to the fine-tune corpus). (2) llama2-13b-chat: the success rate for fine-tuning with LoRA is 2 out of 3, while 0 out of 3 for normal fine-tuning (generating rubbish information).

Answer for RQ2: the LoRA is providing very good effectiveness and improving the generation quality even for nuanced and complex tasks like implicit hate speech detection [HKPA24]. However,



~~~
🤖⚙
  You connect with an OpenAI API Key, then the OpenAI Python SDK (or OpenAI CLI if you are on Linux).  OpenAI Pytho
n SDK docs: https://openai.com/docs/api-reference/  OpenAI.CLI docs: https://openai.com/docs/cli/usage/openai-cli-e
xamples/  OpenAI Python SDK docs: https://openai.com/docs/api-reference/  How do I get started with OpenAI API Key
s? 📦⚙  You can register for an OpenAI API Key from the OpenAI Developer Portal: https://beta.developer.openai.com/
openapi-keys/create. If you do not see an activation URL, this means you need to wait for API access to start and y
our key is pending. Once you receive the activation URL, you should copy it for use.  How do I connect my OpenAI AP
I Key for CLI use? 🏗⚙  Create a file in ~/.openai-config (any file name, but it should contain your API Key).  Thi
s location is recommended because it allows for easy update of your key if changes are needed.
```CLOUD.OAUTH2 {  TOKEN=YOURTOKENHERE  }  ```  The recommended location is "~/.openai-config, but you may want to
put it somewhere more appropriate.  You can change the ~/.openai-config file to whatever path is more appropriate,
but in the case the environment var openai_config_path is not set, ~/.openai-config is assumed.  How do I use Pytho
n? 🤖⚙  You can install the OpenAI Python SDK using an OpenAI CLI command. OpenAI CLI docs: https://openai.com/doc
s/cli/usage/openai-cli-examples/  OpenAI Python SDK docs: https://openai.com/docs/api-reference/  The OpenAI Python
SDK allows you to control all OpenAI API-related requests, and it includes a large list of features that are easier
to work with than those on the OpenAI API.  How do I use the gpt-3-openai-api? ���
~~~

Figure 1: The answer of Llama2-7b is fine-tuned by 1 epoch with the help of LoRA. It is answering the question of 'How do I use the OpenAI API?'

the 7B parameter language model may be unable to resolve the tasks more easily. Still, the decent performance of the llama2-13b-chat model showcased the great potential of utilizing the LoRA technique to obtain very feasible LLMs fine-tuning solutions even for complex tasks (do note that the ChatGPT only had 80% accuracy in early 2023 [HKA23].)

4.3 RQ3: Given contextual financial information, How well does the LoRA model predict the stock market trend?

Now, we come to the specific financial application of Bitcoin market price trend prediction.

In the Bitcoin market price trend prediction task, we have the test result as follows: (1) llama2-7b: the success rate for fine-tuning with LoRA is 1 out of 5 (all answers understand the question, but the LLM may have an inherent bias towards Bitcoin, all the confirmed answers are predicting the increase after news. In the Ground Truth labels, 2 is increased, 3 is decreased.), while 2 out of 5 is not a fine-tuned model (only the two correct ones understand the question). (2) llama2-13b-chat: the success rate for fine-tuning with LoRA is 0 out of 5 (2 decreased predictions, 2 increased predictions, and one meaningless answer), while 0 out of 3 for normal not fine-tuned model (understanding the question well and giving analysis about the exact given headline information; however, still predicting all increase seems due to an inherent bias of the positive idea on Bitcoins). (3) ChatGPT-3.5: predicting 5 increasing, having an accuracy of 2 out of 5. (4) ChatGPT-4: same as ChatGPT-3.5, but the answer is more convincing even if it is wrong.

Answer for RQ3: We observe the inherent bias inside the not fine-tuned models (even ChatGPT); the assumption is the bias of the pre-training corpus utilized in most existing LLMs. Surprisingly, with the help of the LoRA technique, the llama2-13b-chat model can avoid that inherent bias towards Bitcoins and consider whether it will increase or decrease.

Further explorations on the aim of building the quant trading system based on LLMs can be approached through the directions as follows:

- Fine-tuning the model with more correlated datasets while avoiding the possible over-fitting
- Design better the information architecture of LLMs-centered market price trend prediction system (LLMs are the center but not the whole system; other ML, RL techniques should be considered if necessary)

5 Conclusion

Our findings investigate the effectiveness and helpfulness of the LoRA-fostered LLMs fine-tuning process, evaluated in multiple tasks to provide comprehensive insights on how to implement the LLM fine-tuning process efficiently and effectively.

The market value trend prediction tasks unveiled the existing limitations and the potential of utilizing the LoRA and LLMs to help build more intelligence language models in various downstream tasks and even shade lights in the quant trading applications.

References

[HKA23] Fan Huang, Haewoon Kwak, and Jisun An. Is ChatGPT better than human annotators? potential and limitations of ChatGPT in explaining implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 294–297, New York, NY, USA, 2023. Association for Computing Machinery.

[HKPA24] Fan Huang, Haewoon Kwak, Kunwoo Park, and Jisun An. Chatgpt rates natural language explanation quality like humans: But on which scales? *arXiv preprint arXiv:2403.17368*, 2024.