

Tutorial 2: Train your own LLMs

CSC 6203 Large Language Models

Ke Ji | Oct. 11, 2024

Context

- 1. Train Your Own LLMs**
- 2. Assignment 2 of Our Course**
- 3. Colab Practice**

First of All, a Reminder

1. Please visit our course website more frequently. We will update the course materials and the latest updates on our website.

Course website: <https://llm-course.github.io>

CSC 6203 Large Language Models
Teaching Complex B201, Friday 13:30-16:20, Sep. 4th - Dec. 13, 2024
Autumn 2024

This course offers a comprehensive study of Large Language Models (LLMs). We'll explore architecture engineering, training techniques, efficiency enhancements, and prompt engineering. Students will gain insights into the application of LLMs in various domains, tools integration, privacy and bias issues, as well as their limitations and alignment. The curriculum includes guest lectures on advanced topics and in-class presentations to stimulate practical understanding. This course is ideal for anyone seeking to master the use of LLMs in their field.
本课程提供对大语言模型 (LLM) 的全面学习。我们将探索大模型的架构工程、提示工程、训练技术、效率提升。学生将深入了解大语言模型在各个领域的应用、工具集成、隐私和偏见问题及其局限性和对齐。该课程包括高级主题的客座讲座和课堂演示，以激发实践理解。本课程对于任何想要掌握大语言模型在其领域的使用的人来说都是理想的选择。

2. **Join our WeChat Group:** Please join our course WeChat group to stay updated and connect with classmates. This will also help ensure that assignments are submitted on time :)



First of All, a Reminder

3. Note that the **deadline of assignment 1** is **Oct. 18th** (11.59pm).

ASSIGNMENT INFORMATION

Due Date Friday, October 18, 2024 11:59 PM	Points Possible 100
---	-------------------------------

For Assignment 1, please consult the "**Assignment_1_Guideline.pdf**" for detailed instructions. This assignment involves selecting a task and applying **Prompt Engineering** techniques to complete it.

A Colab notebook has been prepared to facilitate your start:

https://colab.research.google.com/drive/1JFtkSnT_Sik8vIqvAXB8iXDKwMiQrHyf?usp=sharing.

Key points:

1. Submit a brief report using the template at <https://www.overleaf.com/read/jpctgzjdry#36db1a>.
2. Deadline: October 18, 2024.

[Assignment_1_Guideline.pdf](#)

[Tutorial_1_Prompt_Engineering_CSC6023.ipynb](#)

1. Train Your Own LLMs

Overall Process

Typically, training your own LLMs contains the following 5 main steps:

- Load pre-trained model and tokenizer (or from scratch?)
- Data preparation
- Start training
- Save your trained model
- Evaluate your trained model on a given test dataset

Training Tricks

You can use other parameter-efficient fine-tuning methods to be able to fine-tune large models on limited GPU resources.

Here, we introduce some useful open source github project that may help.

- *PEFT: <https://github.com/huggingface/peft>
- OpenDelta: <https://github.com/thunlp/OpenDelta>
- QLoRA: <https://github.com/artidoro/qlora>

Training Frameworks

Using notebooks for long-term model training is inconvenient in some cases. You can also try the following model training frameworks:

- *LLaMA-Factory: <https://github.com/hiyouga/LLaMA-Factory>
- stanford_alpaca: https://github.com/tatsu-lab/stanford_alpaca
- FastChat: <https://github.com/lm-sys/FastChat>
- DeepSpeed-Chat: <https://github.com/microsoft/DeepSpeedExamples/tree/master/applications/DeepSpeed-Chat>
- LLMZoo: <https://github.com/FreedomIntelligence/LLMZoo>

2. Assignment 2 of Our Course

Assignment 2

You will be asked to **train your own LLMs**. We offer six optional training tasks, or you can choose one that personally interests you. The given tasks are across two languages and three domain. We encourage you to exercise creativity and adapt the training process to engage in interesting explorations, especially during the instruction fine-tuning process.

We will provide a detailed PDF file with assignment instructions:

<https://llm-course.github.io/Assignments/Assignment2/Assignment 2 Train Your Own LLMs.pdf>

Assignment 2 Deadline

- **Deadline:** 2024. 11. 15, 11:59pm

Take it easy, but don't forget to submit on time.

3. Colab Practice

Practice of Train your own LLMs

If you're new to using LLMs, don't worry; we've prepared a [Colab notebook](#) for you:

https://colab.research.google.com/drive/19ZfC6roqmBggKvhMfyILJkIVkYqoCrmD#scrollTo=UpU2_tmDcmT2

This will guide you on

1. Utilizing model, tokenizer, and dataset loading function from Hugging Face.
2. Performing basic data cleaning.
3. Training the model with basic modeling techniques, including quantization, such as qlora in this instance.
4. Evaluating the model's performance on test set.
5. Saving your custom model and preparing it for deployment.

Try to deploy your own LLMs

Gradio: <https://www.gradio.app/guides/quickstart>

Building Your First Demo


You can run Gradio in your favorite code editor, Jupyter notebook, Google Colab, or anywhere else you write Python. Let's write your first Gradio app:

```
import gradio as gr

def greet(name, intensity):
    return "Hello, " + name + "!" * int(intensity)

demo = gr.Interface(
    fn=greet,
    inputs=["text", "slider"],
    outputs=["text"],
)

demo.launch()
```

 **Tip:** We shorten the imported name from `gradio` to `gr`. This is a widely adopted convention for better readability of code.

Ollama: <https://ollama.com/download>

Thanks

That's all for today's class, and you are now free to leave!