Assignment 7: Fine-Tuning Large Language Models
Language: Python
Due: Wed 11:59am

Preliminary requirements:

- Huggingface account (free): https://huggingface.co/
- Access to Llama2 (If you decide you want to use this model): https://llama.meta.com/llama-downloads/
    - Also be sure to request access on Huggingface as well: https://huggingface.co/meta-llama/Llama-2-7b-hf
- Create a HuggingFace token
    - Go to profile -> settings -> Access Tokens
- Public ssh key:
    - https://vast.ai/docs/instance-setup/ssh
    - **You need to send me your ssh key on slack or via email jbrophy@ucdavis.edu**

PLEASE READ

- Every group will have a 6 hour window of access to a Vast.ai instance.
- I need at least a 2 hour notice before I can get your instance up and running.
- If you finish before the 6 hour window, please let me know so I can shut the instance down.
- **You will be required to add comments to your code before I will reserve your instance**.

Data:

- 4 different data sources of data for different tasks
    - StackOverflow: for coding and programming assistants
        - https://huggingface.co/datasets/jbrophy123/stackoverflow_dataset
    - Quora: Informal question-answer format suitable for a more conversational AI
        - https://huggingface.co/datasets/jbrophy123/quora_dataset
    - Alpaca: a large question-answer dataset generated by GPT-3, useful if you want the AI sound more similar to ChatGPT
        - https://huggingface.co/datasets/jbrophy123/alpaca_dataset
    - Medical: medical question-answer dataset
        - https://huggingface.co/datasets/jbrophy123/medical_dataset
- A sentiment classification dataset: https://huggingface.co/datasets/jbrophy123/imdb_sentiment_analysis
- **If you would like to use a dataset not listed here, please ask me in advance. You must provide me with a test set which you have validated has no observations in common with the train set.**

In all cases, you are limited to 2k training observations (you may use less, but not more than 2k). **Hint: You can modify my code to subset your dataset to only 2k observations, or you can edit the model training code directly**

Model: You may choose any publicly available, open-source LLM (Not ChatGPT)
Some of my personal favorites:

- Llama 2 7B (from Meta)
- Mistral 7B

Some additional considerations:

- the model must be available on HuggingFace for you to use it
- Models have two versions: chat and regular. The Chat versions have been instruction fine-tuned, which occasionally makes them difficult to fine-tune for new tasks.

You must use 7 Billion parameters or less in any model you choose

Questions:

1. Model and Data decisions
    a. What data source did you choose and why?
        i. This is obviously up for interpretation, but a good response focuses on why the data chosen will help you solve a problem
    b. What business/environmental/social/political problem will your fine-tuned model tackle?
        i. This is obviously up for interpretation, but a good response focuses on why is a model like this actually necessary? How does it solve a problem, who will use it, why do they need an AI to help them solve this problem?

2. Explain to me, in your own words:
    a. What is LoRA, and why is it necessary?
        i. LoRA, or Low Rank Adaptation, involves decomposing our pre-trained weight matrix C into smaller two matrices A and B of rank r which is lower than the rank of C. A and B approximate C such that C=AB. This is necessary because fine-tuning LLM's is expensive and time consuming, LoRA updates only a small portion of model weights and therefore saves time, money, and energy.
    b. What is QLoRA and why is it necessary?
        i. QLoRA is very similar to LoRA in the sense that we still decompose our pre-trained weight matrix into smaller matrices of lower rank. However, the main difference is that the model is quantized such that each weight and bias term now take up less space in computer memory. This is necessary because it allows us to fine-tune models for even cheaper on a single GPU
    c. What is instruction fine-tuning and how is it different from the unsupervised pre-training?
        i. Instruction Fine-tuning involves tuning a model so that it behaves like a 'chatbot' by training it on a sample of conversational data, where the

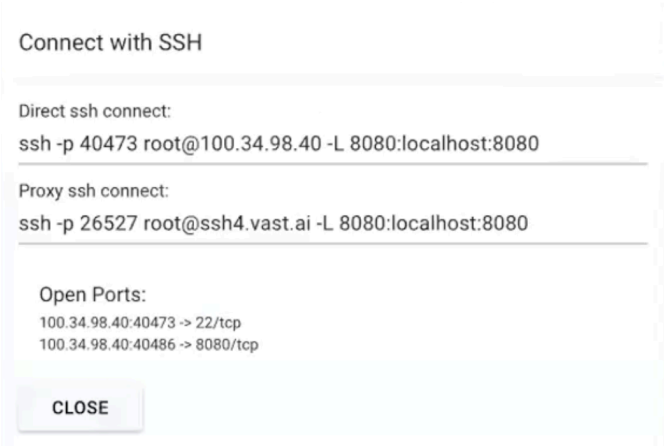
model learns to replicate the way people talk to each other. Unsupervised pre-training is where you teach the model about the world by giving it a lot of data, and then training it on the unsupervised task of repeatedly predicting the next token in a massive collection of text.

   d. Is what you are doing here considered instruction fine-tuning or unsupervised pre-training? Why?
      i. This is instruction fine-tuning because we are training the language model to follow specific conversation patterns, relating to one of the data categories students chose.

3. Coding:
   a. Note: the following is only if you decide to use my code, if you write your own code then normal BAX452 standards apply
      i. Because you have been provided with most of the code for fine-tuning your LLM, please make sure to go through the code and add comments to each line, explaining what it does.
      ii. In the script arguments file, you will notice that I have provided you with empty metadata fields, please give a short explanation of each to understand what each hyperparameter is/does.
      iii. Both of these things are required to be complete before I reserve your instance
4. Fine-tuning:
   a. Please fine-tune your model on your given task, either by writing your own shell script to run your code, or by running it directly through the Vast.ai command line
      i. Note: don't forget to .ssh into your vast.ai instance **(see tutorial on last page)**
   b. Make sure you properly specify the lora_dir argument, otherwise your work will be lost
      i. Note: Huggingface is the easiest place to save it. In my code, you will see the default location is jbrophy123/practice_llm. Change this to your own huggingface account/directory. **Point deductions will be made for those who upload their model to my account instead of their own.**
   c. Please write the name/directory of your model here:

Vast.ai tutorial

- Start by creating your own public access key: https://vast.ai/docs/instance-setup/ssh
  - Send me your key
- I will create an instance for you on vast.ai, and you will get an ssh key:



- I will send you the direct ssh connect, let's break down each part:
    - ssh: sets up the connection to the remote server
    - -p 40473 stands for the port number on the remote server
    - root@100.34.98.40 stands for the login credentials and the ip address of the remote server
- ssh will log you into the instance, but we still need to copy over the code from our local computer to the vast.ai instance
    - We do this with scp or secure copy protocol (in the video I said secure transfer protocol which is wrong, sorry it was early)
    - We can use all the parts of our ssh command to create a scp command:
        - Format: **scp -r -P <remote port number> <location of code on local computer> <username and ip address of server>:<directory on server>**
        - For this server and for my local directory, the scp command would be something like

scp -r -P 40473 /Users/jakebrophy/Desktop/llm_assignment root@100.34.98.40:/root/

Once you are logged in you can run your code either through the command line or through a shell script.

Instances:

- Patrick
- Charles
- Sushma
- Winnie
- Yingyu
- Yongjia (Will)
-