Setting up LLMs on CURC Resources

Research Computing
UNIVERSITY OF COLORADO BOULDER

Setting up LLMs on CURC Resources

Instructor: Brandon Reyes

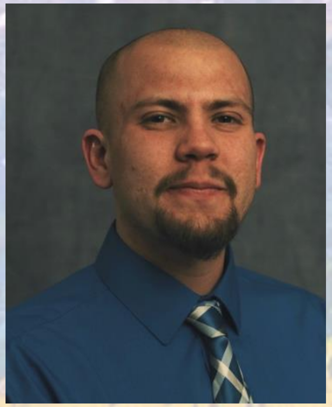
  - Research Computing

- Website: www.rc.colorado.edu
- Documentation: https://curc.readthedocs.io
- Helpdesk: rc-help@colorado.edu
- Survey: http://tinyurl.com/curc-survey18

Research Computing
UNIVERSITY OF COLORADO **BOULDER**

Meet the User Support Team

Layla
Freeborn


Brandon
Reyes

Andy
Monaghan

Michael
Schneider

John
Reiland

Dylan
Gottlieb

Mohal
Khandelwal

Ragan
Lee

Slides

https://github.com/ResearchComputing/setting_up_llms_on_curc_resources_rc_short_course

Session Overview

- What are LLMs?
- What is an LLM framework?
- Accessing CURC hosted LLMs
- Ollama
- Transformers by Hugging Face

What are LLMs?

Large Language Models (LLMs) are types of artificial intelligence models that are trained on large amounts of data, which enables them to “understand” and generate natural language. They are great at:

- Summarizing material
- Constructing code (if they are trained for coding tasks)
- Conversational tasks
- Generating text

When you break it down, LLMs are made possible by multiplication and non-linear transformations. This allows them to run very quickly on GPUs.

Important Considerations for LLMs

- Not all LLMs are equal, some are trained for specific tasks, such as code generation
- The type of data the LLMs are trained on determines what they can do
- General models (ChatGPT, Gemini) are great for common knowledge tasks, but can fall short in certain technical areas
- Bigger is not always better
  - You can have small LLMs that are extremely performant for a certain task
- LLMs can “hallucinate”. Put simply, this means they can generate incorrect answers or content that may sound plausible
  - Don’t treat LLMs as if they are all-knowing! They are not … yet

Proprietary vs Open-Source LLMs

- Models such as ChatGPT are proprietary and require that you run the model on their dedicated hardware
  - Pros: these companies have a lot of resources, allowing for quicker access and their models are usually state-of-the-art
  - Cons: **Data privacy and cost**
- Open-source LLMs can be ran anywhere
  - Pros: you can run them for free on our resources, your data stays on our system, and there are some very capable open-source models
  - Cons: proprietary models are not going to be available and complex LLMs can consume a lot of GPU memory

What is an LLM framework?

Now that we know why we would like to run an LLM locally, how do we do that? In the past, running an LLM locally took quite a bit of effort. However, now there are several LLM frameworks available.

An LLM framework allows for a user-friendly way to access LLMs and utilize them

- Can enable access to proprietary or local LLMs
- Can enable running the LLM from the command line or within a script

Common LLM Frameworks

There are many LLM frameworks out there and some that include additional functionality. However, in this talk we will focus on two extremely popular LLM frameworks that enable you to run models locally:

- Ollama
- Transformers by Hugging Face

Before we jump into things …

Before running or working with LLMs on CURC resources, please be sure that you are adhering to all CURC User Policies and CU's policies and guidance around Artificial Intelligence outlined in the pages Artificial Intelligence at CU Boulder and Resources & Guidance.

Accessing CURC hosted LLMs

To streamline access and reduce redundant storage of LLMs, CURC has created a shared space for common LLMs

- This is a CURC managed space, you cannot store models in this space
  - If you do think a model should be here, reach out to us!
- All models can be accessed using the environment variable CURC_LLM_DIR
  - You must be on a compute node to get access to this variable and the provided LLMs

What is Quantization and Instruct?

- Quantization
  - When training, most LLMs use either single or double precision. This can make those models consume a lot of memory.
  - Quantization allows you to reformat the model into a lower-precision format, which makes the model consume less memory.

- Instruct
  - Instruct or instruction models are ideal for specifying tasks for the LLM to perform and are the models used in common chat interfaces.

CURC hosted Ollama Compatible LLMs

- gpt-oss:20b
    - An open-source model from OpenAI (ChatGPT folks)
    - Requires about 14 GB of GPU memory
- gemma3:12b
    - An open-source model from Google (a derivative of Gemini)
    - Requires around 8 GB of GPU memory
- llama3.1:8b
    - An open-source model from Meta
    - Requires about 5 GB of GPU memory

All of these models are quantized, instruct models, and available at:

$CURC_LLM_DIR/ollama

CURC hosted Transformer Compatible LLMs

- gpt-oss-20b
  - This model has been quantized
  - Requires about 14 GB of GPU memory
- gemma-3-12b-it
  - This model has not been quantized
  - Without quantization, it requires more than 20 GB of GPU memory
- Llama-3.1-8B-Instruct
  - This model has not been quantized
  - Without quantization, it requires more than 20 GB of GPU memory

All of these models are instruct models and available at:

$CURC_LLM_DIR/hf-transformers

Ollama

Ollama is an LLM framework maintained by Meta.

- Open-source and extremely beginner friendly
- Enables users to run LLMs locally and retrieve models

Ollama is slightly different from other frameworks

- Easily installable models are maintained by Ollama, this means they choose what is available (you can use other models, but you need to convert them to a compatible format)
- You first have to establish an Ollama server first. An Ollama server is what allows you to install, manage, and run LLMs

Using CURC’s Ollama install

Our modules are only available on NVIDIA GPUs, we only provide the most up-to-date version, and these versions will be updated during each planned maintenance

Installing Ollama

Check out the “Self-install instructions” tab of file:///Users/brre2566/CURC_work/RC-Documentation/build/html/software/llms.html#ollama

Running Ollama from the command line

Before proceeding with these instructions, be sure that the Ollama server is up and running. Once the Ollama server is up and running, we can interact with an LLM from the command line! Let’s run a very simple model, say the 8 billion parameter Llama 3.1 model:

Explain that Ollama needs to load the model into the GPU memory first

Using Ollama in a Python Script

Before proceeding with these instructions, be sure that the Ollama server is up and running and the Ollama Python API is available. To begin, we need to create a Python script that specifies the model we want to use and input we want to provide to that model. Let's create a script called llama_query.py with the following content:

Explain that Ollama needs to load the model into the GPU memory first

Transformers by Hugging Face

Large Language Models (LLMs)

Thank you!

Survey and feedback

http://tinyurl.com/curc-survey18

Slides

https://github.com/ResearchComputing/setting_up_llms_on_curc_resources_rc_short_course