Guide to install LLMs locally using Ollama

Author: Jayanta Adhikary

Introduction

Large Language Models or LLMs are machine learning models trained on huge sets of data and can be used to recognize and generate texts. One of its most common use cases is Generative AI, ie, when given a prompt or a question, it can provide text in reply.

Running open-source LLMs in our system locally can be quite troublesome but thanks to a few tools like Ollama, it can be very straightforward.

Installation

1. Visit Ollama's website and click on the download button.

Ollama
Get up and running with large language models, locally.
https://ollama.com/

2. Choose your operating system on the downloads page and follow the steps to install Ollama.

Run Ollama

1. After installing Ollama, you can go to your terminal, and use the `ollama` command to check if it has been installed properly. You will get the available commands and flags for Ollama.

2. You can use `ollama list` to view all the downloaded models. It should be empty if you are using Ollama for the first time.
3. You can download models using `ollama pull model-name`, where model-name is the name of the model you wish to download which are available at the ollama library.

   There are ways you can use models that are not directly available at Ollama, and I will discuss them later in this guide.

library
Get up and running with large language models, locally.
https://ollama.com/library

4. After a model is downloaded, you can run it using the run command.

   Example: `ollama run llama2`
5. The model runs and allows you to send a prompt. It replies with an answer. You can quit the prompt session by typing `/bye`

Ollama Rest API

Ollama has a REST API for running, generating responses and managing large language models. It is hosted on `localhost 11434` . If you visit the site when ollama is running, it should show the following text: *Ollama is running*.

Lets try sending a request to the Ollama API using curl in the terminal to generate responses.

```
curl http://localhost:11434/api/generate -d '{
  "model": "llama2",
  "prompt":"Why is the sky blue?",
    "stream": false
}'
```

The optional parameter stream in the above example when set to false returns only a single json object as a response (by default it is true)

You can check out Ollama's API Documentation if you want to use the other optional parameters or learn more on how to use the api to run and manage models.

Next, I will show how you can use Python to generate responses through the Ollama API.

Using Python to Generate Responses

There are different ways to generate responses through ollama using python. I will be using the Ollama Python Library to make the process simple and easy.

Install the library using `pip install ollama`

```
import ollama
response = ollama.chat(model='llama2', messages=[
  {
    'role': 'user',
    'content': 'Why is the sky blue?',
        'stream': False
  },
])
print(response['message']['content'])
```

This is how simple it is to use the ollama python library to generate responses. The traditional way of generating responses which I used before was very time consuming and difficult. This library totally changed the way I used to interact with models using python.

Exposing Ollama to Local Network

1. Lets use Environment Variables to configure the Ollama server and change the host from 127.0.0.1 to 0.0.0.0

   For Mac: `launchctl setenv OLLAMA_HOST "0.0.0.0"`

   For Linux:

   a. `systemctl edit ollama.service`

   b. This will open up an editor

   c. For each environment variable, add a line `Environment` under section `[Service]`

   ```
   [Service]
   Environment="OLLAMA_HOST=0.0.0.0"
   ```

2. Reload the Ollama application if mac, or if linux type the commands below to reload systemd and ollama.

   ```
   systemctl daemon-reload
   systemctl restart ollama
   ```

3. Now to access the Ollama api on other devices in the local network, we can use our device ip address and the port in this format: `ipaddress:port` to check if it is accessible. If it is not working, that might mean the port must be blocked by your device's firewall.

   a. For Ubuntu:

      a. If you are using Debian or Ubuntu based devices you can use the Uncomplicated Firewall (ufw) to manipulate the firewall.

      b. You can type `sudo ufw enable`

      c. If you want to enable the port 11434, you can type `sudo ufw enable 11434`

      d. Optional: If you are using ssh, don't forget to type `sudo ufw enable ssh` else you might not be able to ssh into the system later.

      e. Use `sudo ufw reload` to apply the changes.

      f. Now you should be able to open the ollama api on your local network.