Manual for installing and running LLaMA2 on:

Windows 11:

Note: This guide will walk you through setting up the Text Generation Web UI using Anaconda and Hugging Face Transformers on a Windows 11 system without a GPU.

Step 1: Install Anaconda

1. Download Anaconda from https://www.anaconda.com/download and choose the appropriate installer for Windows.
2. Run the installer and follow the on-screen instructions to complete the installation.

Step 2: Open Anaconda PowerShell

1. Open the Start menu and search for "Anaconda PowerShell." Launch it.

Step 3: Create and Activate a Virtual Environment

1. In the Anaconda PowerShell, create a new virtual environment named textgen2 and install Python 3.10.9:

```
conda create -n textgen2 python=3.10.9
```

2. Activate the newly created virtual environment:

```
conda activate textgen2
```

Step 4: Install Dependencies

1. Install PyTorch and related packages (CPU version):

```
python -m pip install torch torchvision torchaudio --index-url
https://download.pytorch.org/whl/cpu/
```

Step 5: Clone and Prepare the Web UI

1. Clone the Text Generation Web UI repository from GitHub:

```
git clone https://github.com/oobabooga/text-generation-webui
```

2. Navigate to the cloned repository:

```
cd text-generation-webui
```

3. Install the required Python packages:

```
pip install -r requirements.txt
```

Step 6: Start the Web Server

1. Start the web server:

```
python server.py
```

2. Note the URL displayed in the command prompt (e.g., an IP address). You'll use this URL to access the Text AI generator in your browser.

Step 7: Access the Text Generation UI

1. Open your web browser (e.g., Chrome, Firefox).
2. Enter the URL you noted from the command prompt into the browser's address bar and press Enter.
3. The Text AI generator interface should now be visible.

Step 8: Choose and Start a Model

1. Within the Text AI generator interface, navigate to the "Chat" tab.
2. Choose a model you want to use from the provided Hugging Face models. You can find the list of available models at https://huggingface.co/TheBloke .
3. For example, if you want to use the "Llama-2-7B-Chat-GGML" model, enter TheBloke/Llama-2-7B-Chat-GGML in the model selection box.
4. Click the "Start" button to begin the chat. Re-running the Environment:

1. Open the Anaconda PowerShell.
2. Activate the textgen2 virtual environment:

```
conda activate textgen2
```

3. Navigate to the text-generation-webui directory:

```
cd path\to\text-generation-webui
```

4. Start the web server:

```
python server.py
```

> *Please replace path|to|text-generation-webui in the "Re-running the Environment" section with the actual path to the text-generation-webui directory on your system.*

MacOS:

Step 1: Prepare the Environment

1. Create a folder named llama2 on your local machine to store the necessary projects.
2. Navigate into the llama2 folder.

Step 2: Clone LLaMA

1. Clone the official LLaMA repository to your local machine using this link: llama (This is official llama project powered by Meta).
2. Follow the repository's instructions to set up and get the project ready.

Step 3: Clone LLaMA.cpp

1. Clone the LLaMA.cpp repository to your local machine using this link: llama2.cpp

> *LLaMA.cpp employs the int4 numerical format, significantly reducing memory requirements, and its performance is heavily impacted by memory constraints on most hardware. LLaMa.cpp is a project that rewrites the inference code of LLaMa (Long Format Language Model) using pure C++. This allows for local execution of LLaMa on various hardware setups.*

Step 4: Obtain Model Resources

1. Visit the LLaMA repository page.
2. Navigate to Meta AI page.
3. Fill in the form and submit it. Check your email for a message from Meta containing the models link.
4. Create a folder named **meta_models** within the llama2 directory.

Step 5: Download Models

1. In the meta_models folder, execute the download.sh script to download model:

```
sh ./download.sh
```

> *download.sh is used to fetch model resources through link.*

Step 6: Build LLaMA.cpp

1. Navigate to the llama.cpp folder.
2. Build the C++ project by running the command:

```
make
```

Step 7: Prepare the 7B Model

1. Ensure your Python version is **3.11** or above before going further.

2. Run the command install required libraries:

```
python3 -m pip install -r requirements.txt
```

3. Run the command to convert the 7B model to ggml FP16 format:

```
python3 convert.py --outfile models/7B/ggml-model-f16.bin --outtype f16
../../llama2/meta_models/llama-2-7b-chat
```

> *This command is for converting the 7B model to ggml FP16 format. After running convert.py script, it'll convert the original pth extension model to smaller size of bin extension model and store it in the 7B folder of models folder in llama.cpp, so you need to make sure you've create 7B folder before running convert.py.*

Step 8: Quantize the Model

1. In the llama.cpp folder, run the command to quantize the model to 4-bits using the q4_0 method. Quantize the model to 4-bits (using q4_0 method):

```
./quantize ./models/7B/ggml-model-f16.bin ./models/7B/ggml-model-q4_0.bin q4_0
```

Step 9: Run LLaMA

1. In the llama.cpp folder, run LLaMA with custom arguments using the 7B model. Customize the setup as needed. An example command:

```
./main -m ./models/7B/ggml-model-q4_0.bin -n 1024 --repeat_penalty 1.0 --color -i -r
"User:" -f ./prompts/chat-with-bob.txt
```

> *Custom arguments using a 7B model, for more custimized setup details, plese check flag usuage*