Setting up the llama2-13b-chat model on a Mac with an M1/M2/M3 chip.

Author: Ganesh Kolandaivelu **LinkedIn:** www.linkedin.com/in/ganesh-kolandaivelu

Credit: This document consolidates information and methodologies from various blogs, discussion forums, and other resources. Although various sources have provided guides for installing the Llama model on Apple's Mx chip, I have faced challenges when following their instructions. To address this, I have simplified and refined the process, including cloning the Llama and llama.cpp repositories from Facebook's GitHub. I experimented different models and utilized the 13B model with suitable quantization parameters and provided comprehensive step-by-step instructions, complete with screenshots and troubleshooting tips. My guide also details the pre-installation necessities for a seamless setup and uniquely offers guidance on running the model with Anaconda, a topic seldom discussed elsewhere. I acknowledge the contributions of numerous tech bloggers and forum participants, and I believe this document offers a thorough guide for a complete, hassle-free installation.

Please refer to the Note section at the end of the document for storage requirements prior to starting the installation process.

Required Installations Before Proceeding

For guidance on installing the prerequisite software, please consult online resources. You have the option to pre-install these tools or install them as needed. Online materials are also available to assist with installation and troubleshooting.

Anaconda Navigator: Anaconda Navigator is a desktop graphical user interface (GUI) included in the Anaconda distribution, which is a popular Python distribution for data science and machine learning. This GUI makes it easier to use Anaconda’s capabilities without needing to manage command line commands. We use the Anaconda environment to create and deploy a Jupyter Notebook, enabling us to write Python code that integrates with the llama model. (This is required only if you want to use the Anaconda environment to develop ML applications; otherwise, you may skip it and use a simple Python program to run the model.)

XCode: We are using XCode for its essential command line developer tools.

Brew: Homebrew, commonly known as "brew," is a package management system for macOS and Linux that simplifies installing, updating, and managing software.

pip: pip is the package installer for Python. You can use pip to install packages from the Python Package Index and other indexes.

wget: wget is a command-line utility for downloading files from the web. With wget, you can download files using HTTP, HTTPS, and FTP protocols.

Step 1: Request access to Facebook llama models, download and install

a) Request access to Facebook llama models

[Commands or programs that need to be executed are highlighted in yellow.]

https://llama.meta.com/llama-downloads/

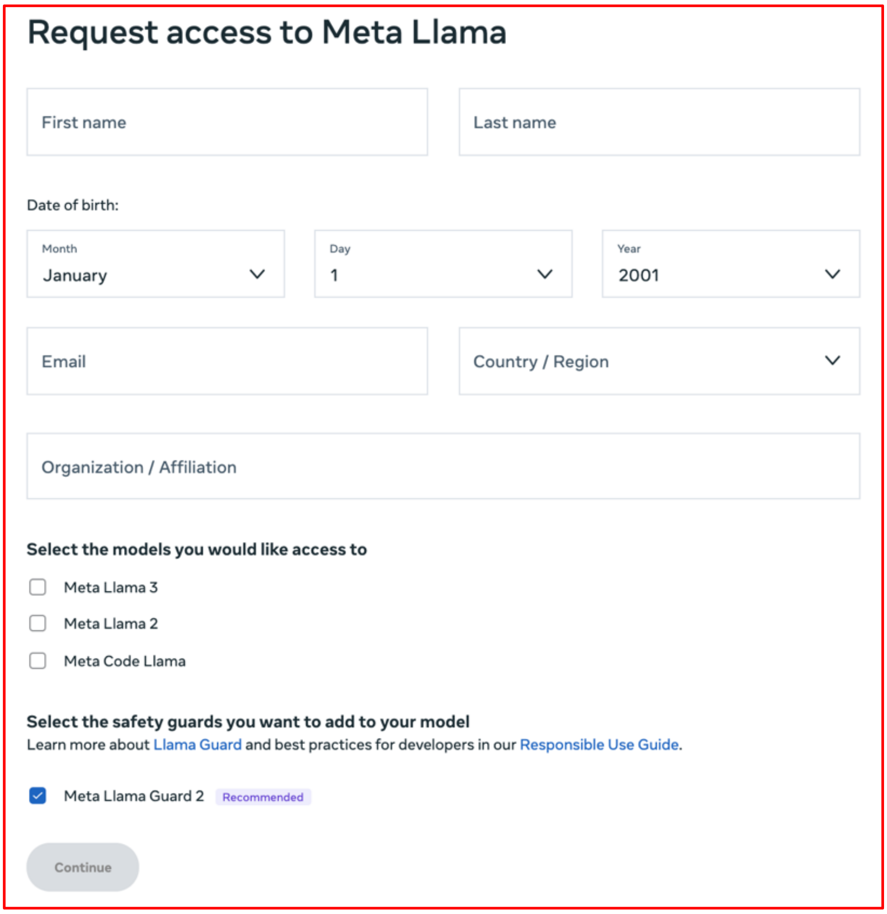
The URL directs you to a form to submit your details and request the model's download.

You have a choice among three options:

- Meta Llama 3
- Meta Llama 2
- Meta Code Llama

While further details about each model are available online for a better understanding, for my project, I'm opting for the 'Meta Llama 2' model, which is the second option on the list.

Fill out and submit the form, and you will get an email from Meta with the download instructions for the model, similar to the one provided below. I have simplified the download and installation steps for you, as outlined below. Typically, a response from Meta should arrive in a matter of seconds and the download URL valid for 24 hours.



Request access to Meta Llama

First name

Last name

Date of birth:

Month: January

Day: 1

Year: 2001

Email

Country / Region

Organization / Affiliation

Select the models you would like access to

□ Meta Llama 3

□ Meta Llama 2

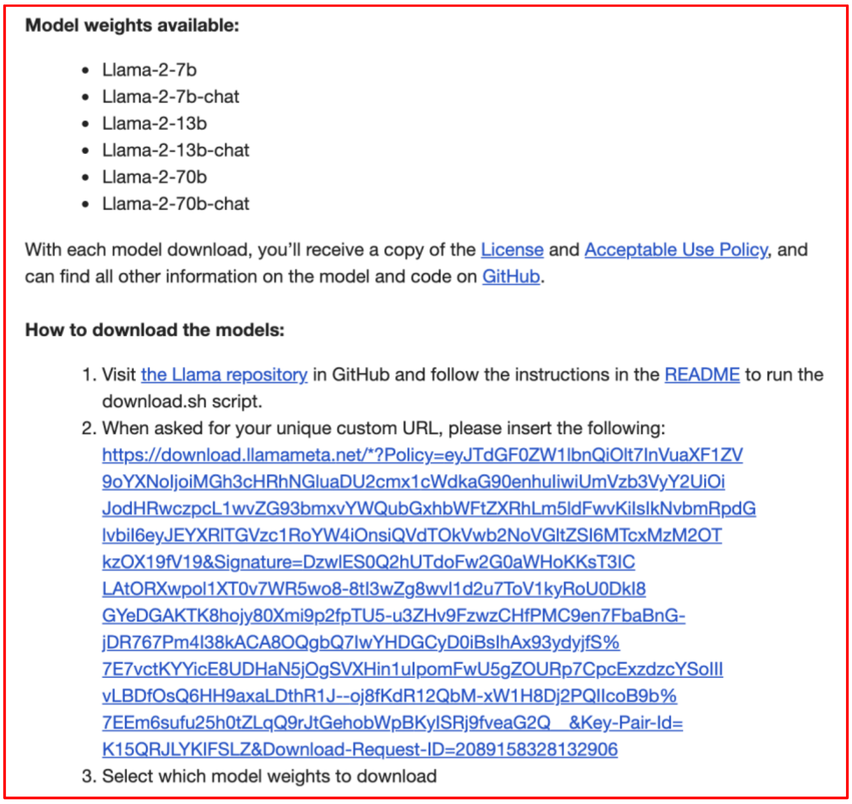
□ Meta Code Llama

Select the safety guards you want to add to your model

Learn more about Llama Guard and best practices for developers in our Responsible Use Guide.

☑ Meta Llama Guard 2 Recommended

Email Response from Meta:



Model weights available:

- Llama-2-7b
- Llama-2-7b-chat
- Llama-2-13b
- Llama-2-13b-chat
- Llama-2-70b
- Llama-2-70b-chat

With each model download, you'll receive a copy of the License and Acceptable Use Policy, and can find all other information on the model and code on GitHub.

How to download the models:

1. Visit the Llama repository in GitHub and follow the instructions in the README to run the download.sh script.
2. When asked for your unique custom URL, please insert the following:
   https://download.llamameta.net/*?Policy=eyJTdGF0ZW1lbnQiOlt7InVuaXF1ZV
   9oYXNoIjoiMGh3cHRhNGluaDU2cmx1cWdkaG90enhuIiwiUmVzb3VyY2UiOi
   JodHRwczpcL1wvZG93bmxvYWQubGxhbWFtZXRhLm5ldFwvKiIsIkNvbmRpdG
   lvbiI6eyJEYXRlTGVzc1RoYW4iOnsiQVdTOkVwb2NoVGltZSI6MTcxMzM2OT
   kzOX19fV19&Signature=DzwlES0Q2hUTdoFw2G0aWHoKKsT3IC
   LAtORXwpol1XT0v7WR5wo8-8tl3wZg8wvl1d2u7ToV1kyRoU0DkI8
   GYeDGAKTK8hojy80Xmi9p2fpTU5-u3ZHv9FzwzCHfPMC9en7FbaBnG-
   jDR767Pm4I38kACA8OQgbQ7IwYHDGCyD0iBsIhAx93ydyjfS%
   7E7vctKYYicE8UDHaN5jOgSVXHin1uIpomFwU5gZOURp7CpcExzdzcYSolII
   vLBDfOsQ6HH9axaLDthR1J--oj8fKdR12QbM-xW1H8Dj2PQIIcoB9b%
   7EEm6sufu25h0tZLqQ9rJtGehobWpBKyISRj9fveaG2Q__&Key-Pair-Id=
   K15QRJLYKIFSLZ&Download-Request-ID=2089158328132906
3. Select which model weights to download

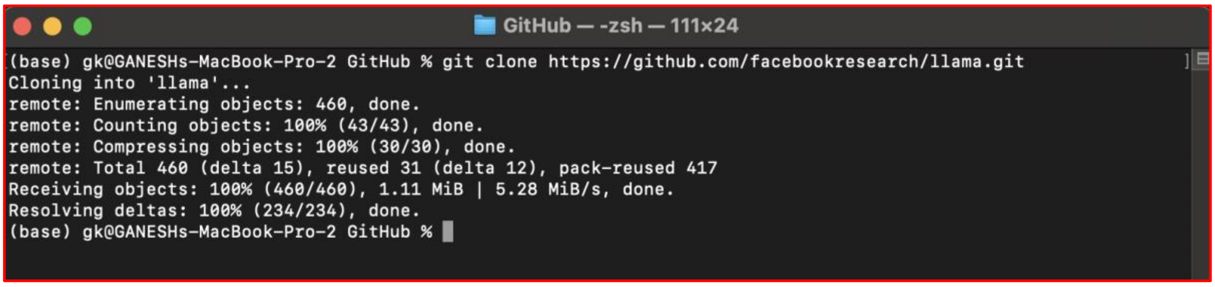
Create a folder named "GitHub" within the Documents directory. Feel free to choose your own folder names, but ensure to adjust the references in my document accordingly to match your chosen folder structure.

b) <u>Clone the llama repository</u>

Navigate to the GitHub folder and run the command

git clone https://github.com/facebookresearch/llama.git

This operation will replicate the Facebook GitHub llama repository, resulting in a new "llama" subfolder within the local GitHub directory.



```
(base) gk@GANESHs-MacBook-Pro-2 GitHub % git clone https://github.com/facebookresearch/llama.git
Cloning into 'llama'...
remote: Enumerating objects: 460, done.
remote: Counting objects: 100% (43/43), done.
remote: Compressing objects: 100% (30/30), done.
remote: Total 460 (delta 15), reused 31 (delta 12), pack-reused 417
Receiving objects: 100% (460/460), 1.11 MiB | 5.28 MiB/s, done.
Resolving deltas: 100% (234/234), done.
(base) gk@GANESHs-MacBook-Pro-2 GitHub %
```

Two methods exist for cloning: SSH and HTTPS. SSH requires an SSH key pair linked to your GitHub account, while HTTPS is often more user-friendly, especially on networks where SSH is restricted or when SSH keys are not configured. Cloning with HTTPS simplifies the process by bypassing SSH key setup, though GitHub may ask for your account credentials. For simplicity and to steer clear of key-related complications, this document employs HTTPS for cloning.

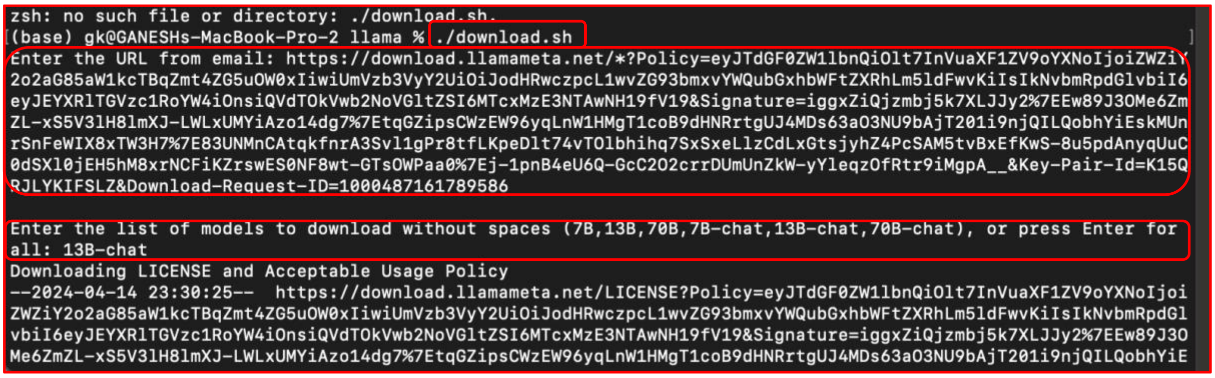
c) Download the model

Navigate into the newly created "llama" folder.

Execute the command - ./download.sh

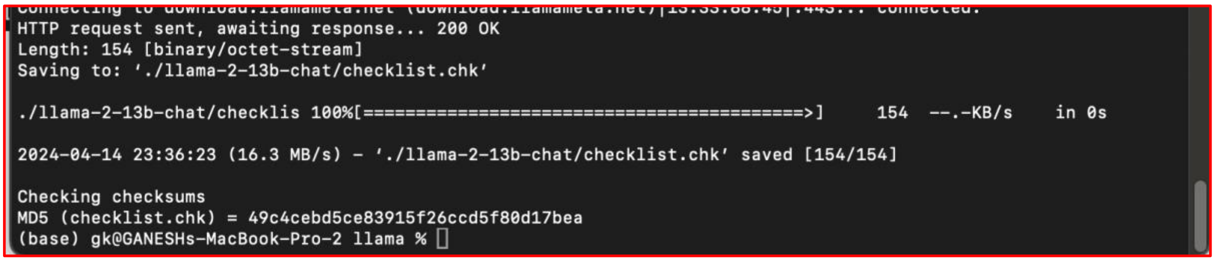
Once you run the download.sh script, you will be prompted to input the URL, which you can retrieve from the email sent by Facebook (valid for 24 hours). The same URL is also depicted in the email screenshot provided in the earlier section.

Following that, the script will request which model to download. From the list of available models, select the '13B-chat' model. Please see the screenshot below for reference on these steps.



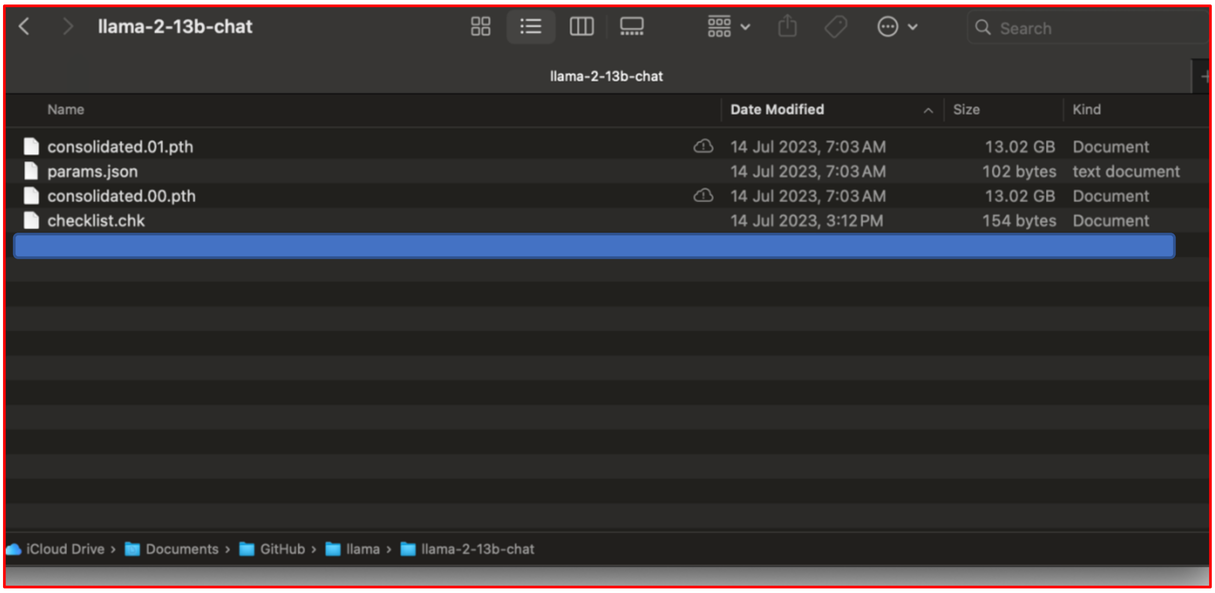
```
zsh: no such file or directory: ./download.sh.
(base) gk@GANESHs-MacBook-Pro-2 llama % ./download.sh
Enter the URL from email: https://download.llamameta.net/*?Policy=eyJTdGF0ZW1lbnQiOlt7InVuaXF1ZV9oYXNoIjoiZWZiY
2o2aG85aW1kcTBqZmt4ZG5uOW0xIiwiUmVzb3VyY2UiOiJodHRwczpcL1wvZG93bmxvYWQubGxhbWFtZXRhLm5ldFwvKiIsIkNvbmRpdGlvbiI6
eyJEYXRlTGVzc1RoYW4iOnsiQVdTOkVwb2NoVGltZSI6MTcxMzE3NTAwNH19fV19&Signature=iggxZiQjzmbj5k7XLJJy2%7EEw89J3OMe6Zm
ZL-xS5V3lH8lmXJ-LWLxUMYiAzo14dg7%7EtqGZipsCWzEW96yqLnW1HMgT1coB9dHNRrtgUJ4MDs63aO3NU9bAjT201i9njQILQobhYiEskMUn
rSnFeWIX8xTW3H7%7E83UNMnCAtqkfnrA3Svl1gPr8tfLKpeDlt74vTOlbhihq7SxSxeLlzCdLxGtsjyhZ4PcSAM5tvBxEfKwS-8u5pdAnyqUuC
0dSXl0jEH5hM8xrNCFiKZrswES0NF8wt-GTsOWPaa0%7Ej-1pnB4eU6Q-GcC2O2crrDUmUnZkW-yYleqzOfRtr9iMgpA__&Key-Pair-Id=K15Q
RJLYKIFSLZ&Download-Request-ID=1000487161789586

Enter the list of models to download without spaces (7B,13B,70B,7B-chat,13B-chat,70B-chat), or press Enter for
all: 13B-chat
Downloading LICENSE and Acceptable Usage Policy
--2024-04-14 23:30:25--  https://download.llamameta.net/LICENSE?Policy=eyJTdGF0ZW1lbnQiOlt7InVuaXF1ZV9oYXNoIjoi
ZWZiY2o2aG85aW1kcTBqZmt4ZG5uOW0xIiwiUmVzb3VyY2UiOiJodHRwczpcL1wvZG93bmxvYWQubGxhbWFtZXRhLm5ldFwvKiIsIkNvbmRpdGl
vbiI6eyJEYXRlTGVzc1RoYW4iOnsiQVdTOkVwb2NoVGltZSI6MTcxMzE3NTAwNH19fV19&Signature=iggxZiQjzmbj5k7XLJJy2%7EEw89J3O
Me6ZmZL-xS5V3lH8lmXJ-LWLxUMYiAzo14dg7%7EtqGZipsCWzEW96yqLnW1HMgT1coB9dHNRrtgUJ4MDs63aO3NU9bAjT201i9njQILQobhYiE
```

Upon successful download of the 13B-chat model, you should see a screen similar to the one shown in the screenshot provided below.



```
Connecting to download.llamameta.net (download.llamameta.net)|13.33.88.45|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 154 [binary/octet-stream]
Saving to: './llama-2-13b-chat/checklist.chk'

./llama-2-13b-chat/checklis 100%[=========================================>]     154  --.-KB/s    in 0s

2024-04-14 23:36:23 (16.3 MB/s) - './llama-2-13b-chat/checklist.chk' saved [154/154]

Checking checksums
MD5 (checklist.chk) = 49c4cebd5ce83915f26ccd5f80d17bea
(base) gk@GANESHs-MacBook-Pro-2 llama % 
```

The "llama-2-13b-chat" folder should now appear inside the "llama" folder, containing the files as depicted in the screenshot below.


llama-2-13b-chat
Search
llama-2-13b-chat

Name	Date Modified	Size	Kind
consolidated.01.pth	14 Jul 2023, 7:03 AM	13.02 GB	Document
params.json	14 Jul 2023, 7:03 AM	102 bytes	text document
consolidated.00.pth	14 Jul 2023, 7:03 AM	13.02 GB	Document
checklist.chk	14 Jul 2023, 3:12 PM	154 bytes	Document

iCloud Drive › Documents › GitHub › llama › llama-2-13b-chat

d) Clone llama.cpp repository

Utilize llama.cpp for the conversion and quantization of the obtained models.

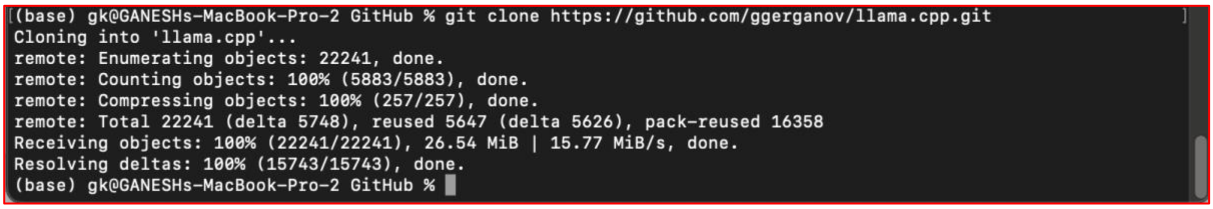
LLaMa.cpp is an efficient C/C++ implementation of Meta's LLaMa architecture, designed by Georgi Gerganov. It stands out in the landscape of large language models (LLMs) by offering a lighter, more portable alternative that is suitable for environments with limited computing resources. The project focuses on optimizing a specific model architecture, leading to notable performance improvements and efficiency in processing LLaMa models through specialized formats like GGML and GGUF.

LLaMa.cpp is particularly useful in industries requiring efficient data processing without the overhead of extensive hardware dependencies. Its adaptability makes it ideal for developing customer service chatbots, sophisticated data analysis tools, and other digital interaction applications where quick response times and resource efficiency are crucial.

Navigate back to the GitHub directory and execute the following command to clone llama.cpp:

```
git clone https://github.com/ggerganov/llama.cpp.git
```

The screen should resemble the provided screenshot after the completion.



```
(base) gk@GANESHs-MacBook-Pro-2 GitHub % git clone https://github.com/ggerganov/llama.cpp.git
Cloning into 'llama.cpp'...
remote: Enumerating objects: 22241, done.
remote: Counting objects: 100% (5883/5883), done.
remote: Compressing objects: 100% (257/257), done.
remote: Total 22241 (delta 5748), reused 5647 (delta 5626), pack-reused 16358
Receiving objects: 100% (22241/22241), 26.54 MiB | 15.77 MiB/s, done.
Resolving deltas: 100% (15743/15743), done.
(base) gk@GANESHs-MacBook-Pro-2 GitHub %
```

A new directory named llama.cpp will now be present within the GitHub folder.

Change to the llama.cpp directory by entering:

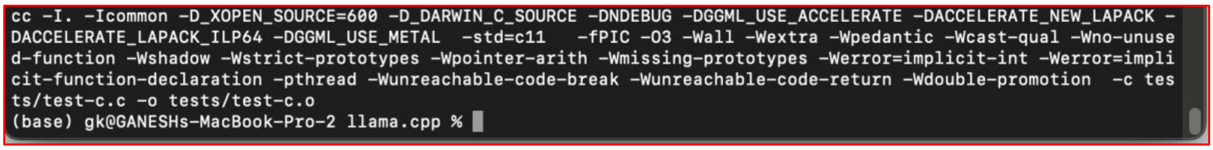
cd llama.cpp

Then compile the code using the below code with the METAL flag set:

LLAMA_METAL=1 make

It appears that setting LLAMA_METAL=1 may trigger the use of GPU resources for computation. Presumably, this environment variable, when set to 1, could activate certain features or optimizations tailored for macOS systems where the Metal API is used to leverage the GPU.

The command will take several minutes to execute, so please be patient until it finishes. The screen should resemble the provided screenshot after the completion.



```
cc -I. -Icommon -D_XOPEN_SOURCE=600 -D_DARWIN_C_SOURCE -DNDEBUG -DGGML_USE_ACCELERATE -DACCELERATE_NEW_LAPACK -
DACCELERATE_LAPACK_ILP64 -DGGML_USE_METAL  -std=c11   -fPIC -O3 -Wall -Wextra -Wpedantic -Wcast-qual -Wno-unuse
d-function -Wshadow -Wstrict-prototypes -Wpointer-arith -Wmissing-prototypes -Werror=implicit-int -Werror=impli
cit-function-declaration -pthread -Wunreachable-code-break -Wunreachable-code-return -Wdouble-promotion  -c tes
ts/test-c.c -o tests/test-c.o
(base) gk@GANESHs-MacBook-Pro-2 llama.cpp %
```

e) Create a Python virtual environment

Navigate to the llama.cpp folder and execute the following commands one after another:

First, set up a virtual environment with the command python3 -m venv .env

The python3 -m venv .env command creates a virtual environment named .env. A virtual environment is an isolated Python environment that allows you to manage dependencies for different projects separately without installing them globally. This helps avoid conflicts between project dependencies. Ensure to use .env as the virtual environment name.

Then activate the virtual environment by typing source .env/bin/activate

Activating it means that any Python packages you install next will only affect this virtual environment, and the Python interpreter used will be the one inside this environment.

After the environment is active, install the necessary dependencies using the command

pip install -r requirements.txt

pip install -r requirements.txt is the command to install all the Python dependencies listed in the requirements.txt file. This file contains a list of pip packages with their respective versions, which are necessary for the project to run.

f) Generate ggml model format

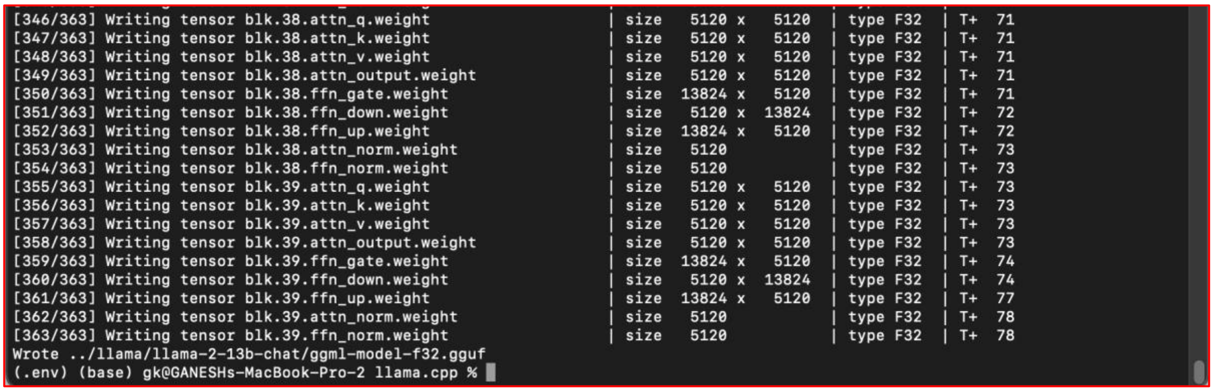
Lastly, run the conversion script with the command

python convert.py ../llama/llama-2-13b-chat

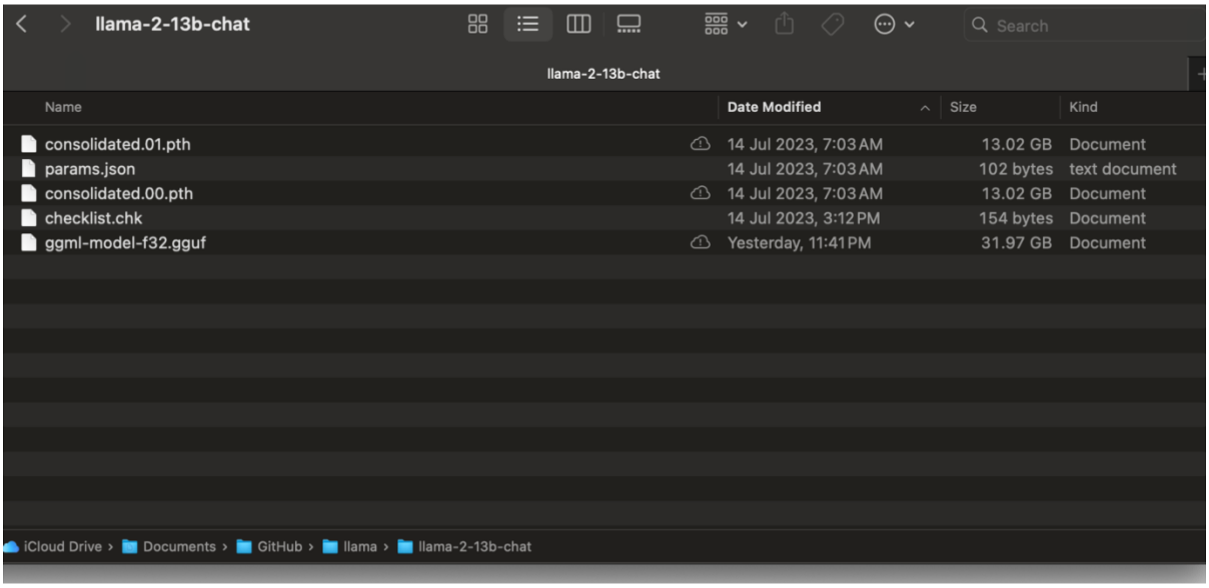
The command that generates the ggml-model-f32.gguf file in the llama-2-13b-chat folder is part of optimizing the llama model for efficient execution on an M1/M2/M3 chip-based Mac OS system. The M1 chip, in my case, being an ARM-based processor with specific machine learning accelerators, benefits from having models that are tailored to its architecture.

The conversion to the ggml-model-f32.gguf format ensures that the model uses floating-point 32 precision, which strikes a balance between computational precision and performance.

By executing this conversion step, you are essentially transforming the machine learning model into a form that is ready for deployment and execution on Mac OS devices with the M1 chip, ensuring that the application can run with the high efficiency and speed that the hardware is capable of providing.



```
[346/363] Writing tensor blk.38.attn_q.weight                   | size   5120 x   5120  | type F32  | T+  71
[347/363] Writing tensor blk.38.attn_k.weight                   | size   5120 x   5120  | type F32  | T+  71
[348/363] Writing tensor blk.38.attn_v.weight                   | size   5120 x   5120  | type F32  | T+  71
[349/363] Writing tensor blk.38.attn_output.weight              | size   5120 x   5120  | type F32  | T+  71
[350/363] Writing tensor blk.38.ffn_gate.weight                 | size  13824 x   5120  | type F32  | T+  71
[351/363] Writing tensor blk.38.ffn_down.weight                 | size   5120 x  13824  | type F32  | T+  72
[352/363] Writing tensor blk.38.ffn_up.weight                   | size  13824 x   5120  | type F32  | T+  72
[353/363] Writing tensor blk.38.attn_norm.weight                | size   5120           | type F32  | T+  73
[354/363] Writing tensor blk.38.ffn_norm.weight                 | size   5120           | type F32  | T+  73
[355/363] Writing tensor blk.39.attn_q.weight                   | size   5120 x   5120  | type F32  | T+  73
[356/363] Writing tensor blk.39.attn_k.weight                   | size   5120 x   5120  | type F32  | T+  73
[357/363] Writing tensor blk.39.attn_v.weight                   | size   5120 x   5120  | type F32  | T+  73
[358/363] Writing tensor blk.39.attn_output.weight              | size   5120 x   5120  | type F32  | T+  73
[359/363] Writing tensor blk.39.ffn_gate.weight                 | size  13824 x   5120  | type F32  | T+  74
[360/363] Writing tensor blk.39.ffn_down.weight                 | size   5120 x  13824  | type F32  | T+  74
[361/363] Writing tensor blk.39.ffn_up.weight                   | size  13824 x   5120  | type F32  | T+  77
[362/363] Writing tensor blk.39.attn_norm.weight                | size   5120           | type F32  | T+  78
[363/363] Writing tensor blk.39.ffn_norm.weight                 | size   5120           | type F32  | T+  78
Wrote ../llama/llama-2-13b-chat/ggml-model-f32.gguf
(.env) (base) gk@GANESHs-MacBook-Pro-2 llama.cpp %
```



llama-2-13b-chat
Search
llama-2-13b-chat

Name	Date Modified	Size	Kind
consolidated.01.pth	14 Jul 2023, 7:03 AM	13.02 GB	Document
params.json	14 Jul 2023, 7:03 AM	102 bytes	text document
consolidated.00.pth	14 Jul 2023, 7:03 AM	13.02 GB	Document
checklist.chk	14 Jul 2023, 3:12 PM	154 bytes	Document
ggml-model-f32.gguf	Yesterday, 11:41 PM	31.97 GB	Document

iCloud Drive › Documents › GitHub › llama › llama-2-13b-chat

Step 2: Quantization of the model

The subsequent phase involves the quantization of the model. Quantization in the context of machine learning models, such as the one being referenced in the command below, is a process that reduces the precision of the numbers used to represent model parameters. This is often done to reduce the computational resources required to store and process the model, which can be especially beneficial for deploying models on devices with limited memory and processing power, such as mobile phones or embedded systems.

By reducing the precision of the parameters, quantization can lead to a smaller model size and faster inference times, at the potential cost of a slight reduction in model accuracy. There are different levels of quantization, with the most common being converting 32-bit floating-point numbers (which are typically used in the training of machine learning models) to 16-bit integers or 8-bit integers or 4-bit integers.

Command:

```
./quantize ../llama/llama-2-13b-chat/ggml-model-f32.gguf ../llama/llama-2-13b-chat/ggml-model-f32_Q4_1.bin Q4_1
```

Let us break down the parameters used:

./quantize: This is the executable command that initiates the quantization process.

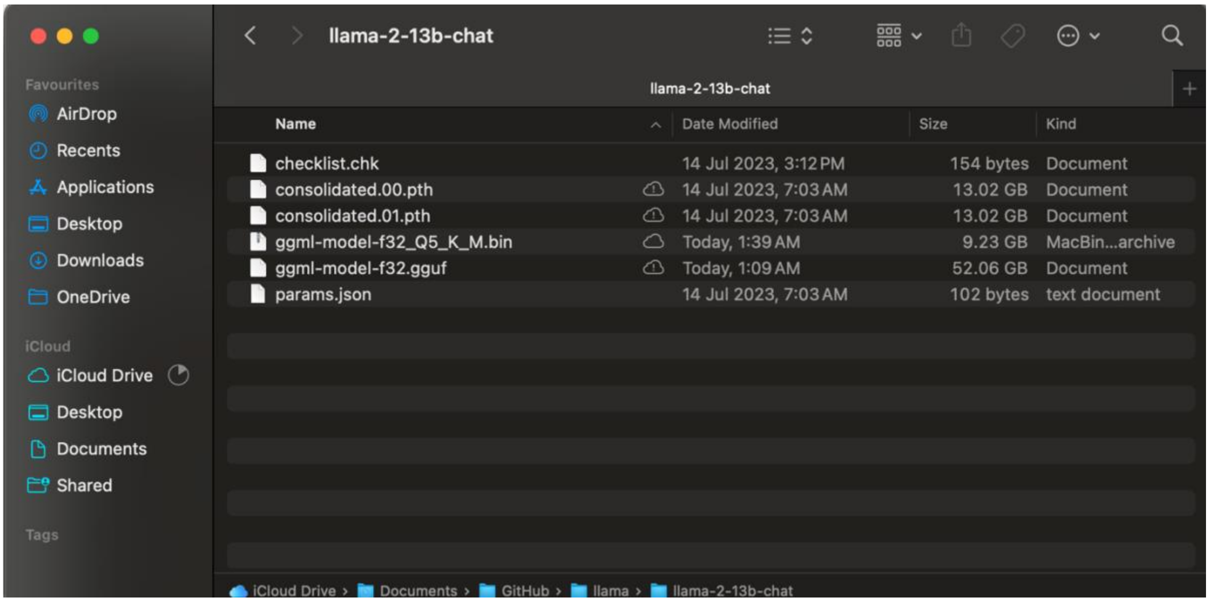
../llama/llama-2-13b-chat/ggml-model-f32.gguf: This is the path to the input file that is to be quantized. The ggml-model-f32 indicates that the original model uses 32-bit floating-point numbers.

../llama/llama-2-13b-chat/ggml-model-f32_Q4_1.bin: This is the path where the quantized model will be saved. ggml-model-f32_Q4_1.bin is the name of the quantized model.

Q4_1: This is an argument to the quantize command specifying the type of quantization to apply. It means "quantize to 4 bits with a specific quantization schema (like scheme 1)". The model parameter Q4_1 achieves greater accuracy compared to Q4_0, although it does not reach the accuracy level of Q5_0. Nevertheless, it offers faster inference times than the Q5 models.

One can explore online resources to discover additional model parameters. Deciding involves weighing the trade-offs between model size, inference, speed, and accuracy to find a balance that suits the specific requirements of the application.

Within the llama-2-12b-chat folder resides the ggml-model-f32_Q4_1.bin, the quantized version of the model optimized for balance between size, speed, and accuracy, suitable for deployment on consumer hardware like the M1 Mac OS systems.



llama-2-13b-chat

Name	Date Modified	Size	Kind
checklist.chk	14 Jul 2023, 3:12 PM	154 bytes	Document
consolidated.00.pth	14 Jul 2023, 7:03 AM	13.02 GB	Document
consolidated.01.pth	14 Jul 2023, 7:03 AM	13.02 GB	Document
ggml-model-f32_Q5_K_M.bin	Today, 1:39 AM	9.23 GB	MacBin...archive
ggml-model-f32.gguf	Today, 1:09 AM	52.06 GB	Document
params.json	14 Jul 2023, 7:03 AM	102 bytes	text document

iCloud Drive › Documents › GitHub › llama › llama-2-13b-chat

Step 3: Running the model using a Jupyter Notebook within the Anaconda environment.

Open a new terminal and run the following commands in the sequence mentioned:

```
conda create --name llamaMac python=3.8 -c conda-forge
```

Using the -c conda-forge flag in our commands directs the installation of packages from the Conda-Forge repository instead of the default Anaconda repository. It is important to note that not using Conda-Forge resulted in compatibility issues and package mismatches on Arm64 architecture in my experience, which Conda-Forge helped to resolve.

Activate the newly created conda environment:

```
conda activate llamaMac
```

Install numpy, pandas and jupyter from conda-forge

```
conda install numpy pandas jupyter -c conda-forge
```

Execute the below commands in the sequence mentioned:

```
export CMAKE_ARGS="-DLLAMA_METAL=on"
```

```
export FORCE_CMAKE=1
```

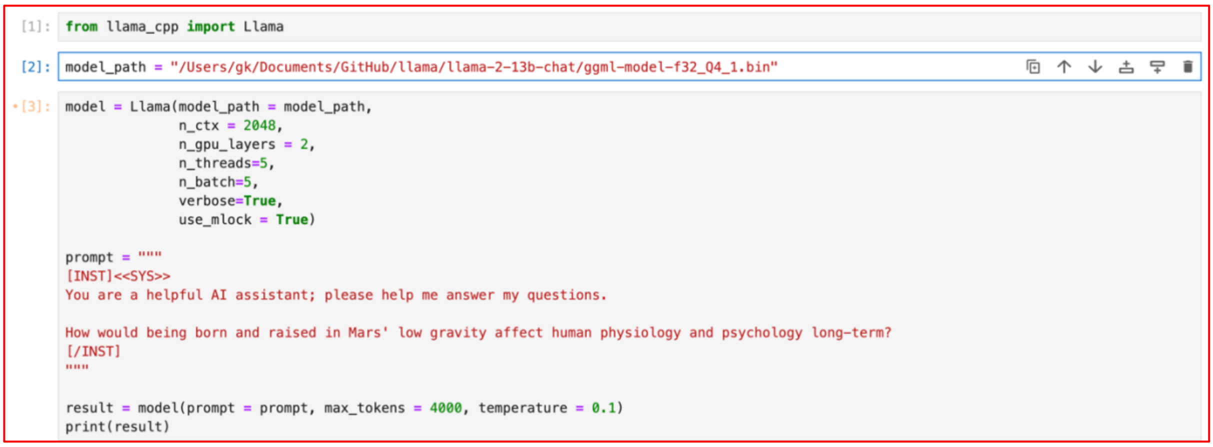
```
pip install llama-cpp-python==0.2.27
```

from the conda environment – llamaMac, invoke jupyter notebook.

```
Jupyter notebook
```

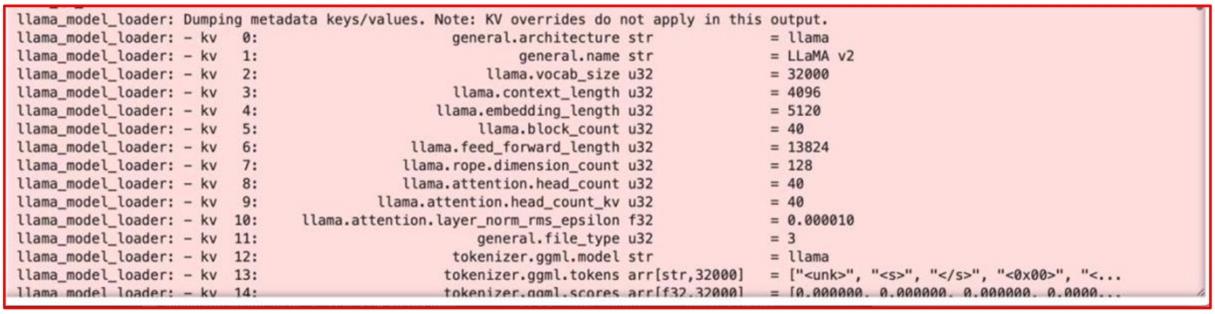
Enter the below code in a jupyter notebook and execute it. Modify the model_path in the code as required.

```
from llama_cpp import Llama
model_path = "/Users/gk/Documents/GitHub/llama/llama-2-13b-chat/ggml-model-f32_Q4_1.bin"
model = Llama(model_path = model_path,
            n_ctx = 2048,
            n_gpu_layers = 2,
            n_threads=5,
            n_batch=5,
            verbose=True,
            use_mlock = True)
prompt = """
[INST]<<SYS>>
You are a helpful AI assistant; please help me answer my questions.
How would being born and raised in Mars' low gravity affect human physiology and psychology long-term?
[/INST]
"""
result = model(prompt = prompt, max_tokens = 4000, temperature = 0.1)
print(result)
```

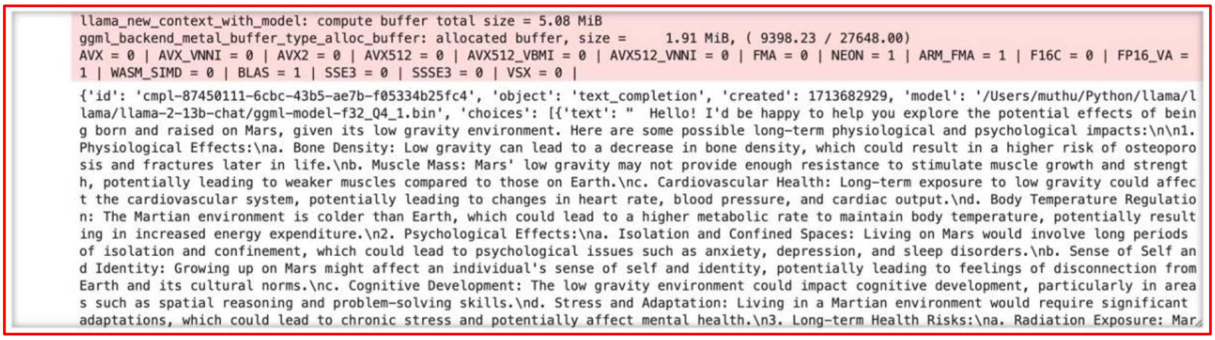


```
[1]: from llama_cpp import Llama

[2]: model_path = "/Users/gk/Documents/GitHub/llama/llama-2-13b-chat/ggml-model-f32_Q4_1.bin"

•[3]: model = Llama(model_path = model_path,
                   n_ctx = 2048,
                   n_gpu_layers = 2,
                   n_threads=5,
                   n_batch=5,
                   verbose=True,
                   use_mlock = True)

      prompt = """
      [INST]<<SYS>>
      You are a helpful AI assistant; please help me answer my questions.

      How would being born and raised in Mars' low gravity affect human physiology and psychology long-term?
      [/INST]
      """

      result = model(prompt = prompt, max_tokens = 4000, temperature = 0.1)
      print(result)
```

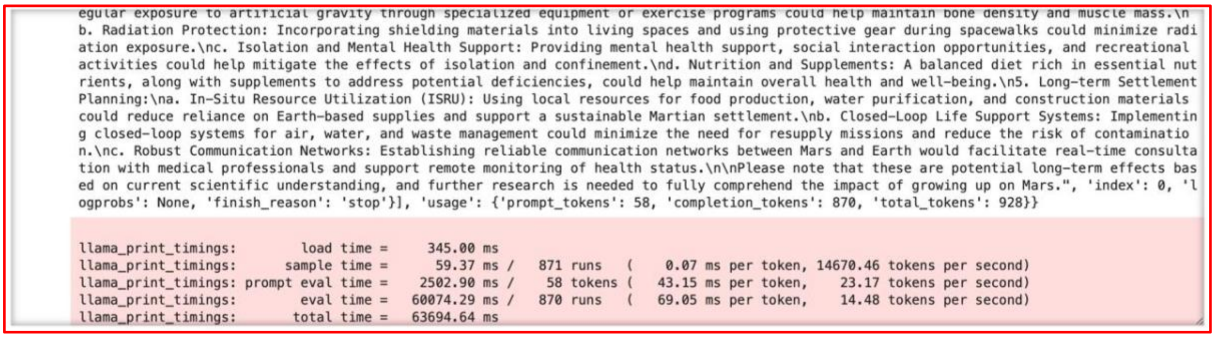
The processing of the results and the appearance of the final output are depicted in the screenshots below.



```
llama_model_loader: Dumping metadata keys/values. Note: KV overrides do not apply in this output.
llama_model_loader: - kv   0:                       general.architecture str              = llama
llama_model_loader: - kv   1:                               general.name str              = LLaMA v2
llama_model_loader: - kv   2:                           llama.vocab_size u32              = 32000
llama_model_loader: - kv   3:                       llama.context_length u32              = 4096
llama_model_loader: - kv   4:                     llama.embedding_length u32              = 5120
llama_model_loader: - kv   5:                          llama.block_count u32              = 40
llama_model_loader: - kv   6:                  llama.feed_forward_length u32              = 13824
llama_model_loader: - kv   7:                 llama.rope.dimension_count u32              = 128
llama_model_loader: - kv   8:                 llama.attention.head_count u32              = 40
llama_model_loader: - kv   9:              llama.attention.head_count_kv u32              = 40
llama_model_loader: - kv  10:     llama.attention.layer_norm_rms_epsilon f32              = 0.000010
llama_model_loader: - kv  11:                          general.file_type u32              = 3
llama_model_loader: - kv  12:                       tokenizer.ggml.model str              = llama
llama_model_loader: - kv  13:                      tokenizer.ggml.tokens arr[str,32000]   = ["<unk>", "<s>", "</s>", "<0x00>", "<...
llama_model_loader: - kv  14:                      tokenizer.ggml.scores arr[f32,32000]   = [0.000000, 0.000000, 0.000000, 0.0000...
```



```
llama_new_context_with_model: compute buffer total size = 5.08 MiB
ggml_backend_metal_buffer_type_alloc_buffer: allocated buffer, size =     1.91 MiB, ( 9398.23 / 27648.00)
AVX = 0 | AVX_VNNI = 0 | AVX2 = 0 | AVX512 = 0 | AVX512_VBMI = 0 | AVX512_VNNI = 0 | FMA = 0 | NEON = 1 | ARM_FMA = 1 | F16C = 0 | FP16_VA =
1 | WASM_SIMD = 0 | BLAS = 1 | SSE3 = 0 | SSSE3 = 0 | VSX = 0 |
{'id': 'cmpl-87450111-6cbc-43b5-ae7b-f05334b25fc4', 'object': 'text_completion', 'created': 1713682929, 'model': '/Users/muthu/Python/llama/l
lama/llama-2-13b-chat/ggml-model-f32_Q4_1.bin', 'choices': [{'text': "  Hello! I'd be happy to help you explore the potential effects of bein
g born and raised on Mars, given its low gravity environment. Here are some possible long-term physiological and psychological impacts:\n\n1.
Physiological Effects:\na. Bone Density: Low gravity can lead to a decrease in bone density, which could result in a higher risk of osteoporo
sis and fractures later in life.\nb. Muscle Mass: Mars' low gravity may not provide enough resistance to stimulate muscle growth and strengt
h, potentially leading to weaker muscles compared to those on Earth.\nc. Cardiovascular Health: Long-term exposure to low gravity could affec
t the cardiovascular system, potentially leading to changes in heart rate, blood pressure, and cardiac output.\nd. Body Temperature Regulatio
n: The Martian environment is colder than Earth, which could lead to a higher metabolic rate to maintain body temperature, potentially result
ing in increased energy expenditure.\n2. Psychological Effects:\na. Isolation and Confined Spaces: Living on Mars would involve long periods
of isolation and confinement, which could lead to psychological issues such as anxiety, depression, and sleep disorders.\nb. Sense of Self an
d Identity: Growing up on Mars might affect an individual's sense of self and identity, potentially leading to feelings of disconnection from
Earth and its cultural norms.\nc. Cognitive Development: The low gravity environment could impact cognitive development, particularly in area
s such as spatial reasoning and problem-solving skills.\nd. Stress and Adaptation: Living in a Martian environment would require significant
adaptations, which could lead to chronic stress and potentially affect mental health.\n3. Long-term Health Risks:\na. Radiation Exposure: Mar
```



```
egular exposure to artificial gravity through specialized equipment or exercise programs could help maintain bone density and muscle mass.\n
b. Radiation Protection: Incorporating shielding materials into living spaces and using protective gear during spacewalks could minimize radi
ation exposure.\nc. Isolation and Mental Health Support: Providing mental health support, social interaction opportunities, and recreational
activities could help mitigate the effects of isolation and confinement.\nd. Nutrition and Supplements: A balanced diet rich in essential nut
rients, along with supplements to address potential deficiencies, could help maintain overall health and well-being.\n5. Long-term Settlement
Planning:\na. In-Situ Resource Utilization (ISRU): Using local resources for food production, water purification, and construction materials
could reduce reliance on Earth-based supplies and support a sustainable Martian settlement.\nb. Closed-Loop Life Support Systems: Implementin
g closed-loop systems for air, water, and waste management could minimize the need for resupply missions and reduce the risk of contaminatio
n.\nc. Robust Communication Networks: Establishing reliable communication networks between Mars and Earth would facilitate real-time consulta
tion with medical professionals and support remote monitoring of health status.\n\nPlease note that these are potential long-term effects bas
ed on current scientific understanding, and further research is needed to fully comprehend the impact of growing up on Mars.", 'index': 0, 'l
ogprobs': None, 'finish_reason': 'stop'}], 'usage': {'prompt_tokens': 58, 'completion_tokens': 870, 'total_tokens': 928}}

llama_print_timings:        load time =     345.00 ms
llama_print_timings:      sample time =      59.37 ms /   871 runs   (    0.07 ms per token, 14670.46 tokens per second)
llama_print_timings: prompt eval time =    2502.90 ms /    58 tokens (   43.15 ms per token,    23.17 tokens per second)
llama_print_timings:        eval time =   60074.29 ms /   870 runs   (   69.05 ms per token,    14.48 tokens per second)
llama_print_timings:       total time =   63694.64 ms
```

Note:

a) To avoid any issues during installation, please ensure that you remove any extraneous spaces when copying commands from the document. Extra spaces can cause errors in the command-line environment.

b) Ensure you have at least 100GB free before starting the installation. The downloaded 13B-chat model takes up about 30GB, the ggml format around 50GB, and the final quantized model an additional 8GB. Choosing llama guard may increase the size of the downloaded model. If storage issues arise, pause the installation and delete the downloaded model to reclaim space. The specific files you delete will depend on your progress in the installation process. For instance, once you have converted the model to the ggml format, you would not need the original model files you downloaded earlier. To locate large files, particularly in cache folders, you can use tools like DaisyDisk on macOS.

c) After successfully operating the model, you can delete the temporary and initial model files, such as consolidated.00.pth, consolidated.01.pth, and ggml-model-f32.gguf. Retain only the final quantized model file, ggml-model-f32_Q4_1.bin, to free up approximately 80 GB of storage space.

d) I conducted this experiment on an Apple M3 chip equipped with 36 GB of RAM and a 12-core processor (6 performance cores and 6 efficiency cores). The test yielded results in approximately 60 seconds. The performance can vary based on the machine's configuration and the complexity of the query inputted.

e) You could explore various models and experiment with different quantization parameters to find the optimum balance that suits your needs for performance, size, inference speed, and resource utilization. While the steps for the models I tested were consistent, you will likely need to adjust the quantization parameters to match each specific model, which you can find online.