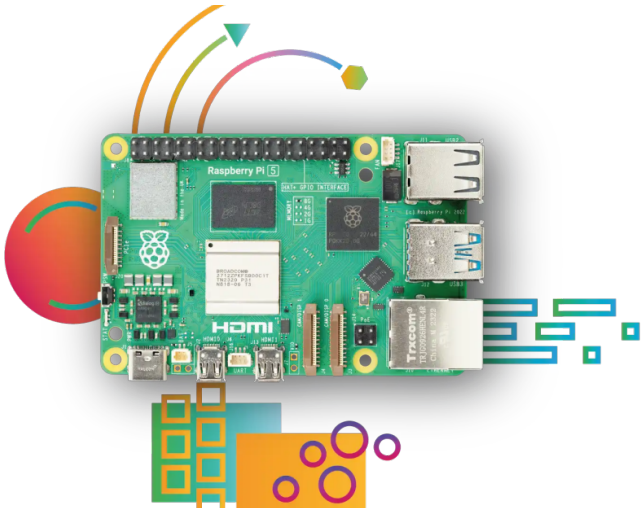


Running LLM (LLaMA) on Raspberry Pi 5

Marek Żelichowski
TM1 / IBM Planning Analytics Expert

December 12, 2023

I finally got my hands on Raspberry Pi 5 and went straight into extensive testing of it's capabilities. I got the more powerful 8GB version (4GB is unfortunately to litte, but it should also work on Raspberry Pi 4), to which specification can be found here.

What is needed to run LLM on Raspberry Pi?

- 8GB Raspberry Pi with power supply
- at least 32 GB memory card, preinstalled with a 64-bit raspbian (or any other linux distro)
- Linux PC with at least 16 GB of RAM
- Pendrive (around 22GB needed)

I will skip the usual basic-setup part, but if You need help on that, it's available easily i.e. here

We need to start on the linux-based PC, this is because we need to convert the raw model to GGPL format, which requires more memory that raspberry has. If we actually try to it on raspi it will throw an illegal instruction error.

First we need to make sure that we are euqiped with all the needed tools, that's why we will try to update our system and check if we have git installed:

```
sudo apt update && sudo apt install git
```

Next we need some more tools: torch, numpy and sentencepiece. Please note that this command may vary a bit on a different linux distributions, but if you are here, you probably will be able to get this solved:

```
python3 -m pip install torch numpy sentencepiece
```

and on the last step of the preparation we need to ensure that we have G++ and build essential installed:

```
sudo apt install g++ build-essential
```

Ok, now let us really start the journey. We need to clone the github repository of the LLaMA model, change the directory to the newly downloaded one, and build the project files

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

cd llama.cpp

make
```

In the meantime we need to start downloading the model, we will be using a 7B model, because it's the biggest that Raspberry can currently handle:

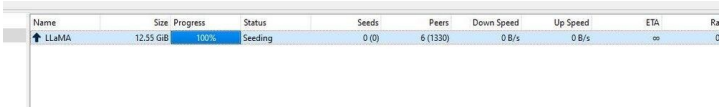
Model	Original size	Quantized size (4-bit)
7B	13 GB	3.9 GB
13B	24 GB	7.8 GB
30B	60 GB	19.5 GB
65B	120 GB	38.5 GB

Comparison of AI model size after quantization

The model can be downloaded from any real source, but I've used this one:

```
magnet:?
xt=urn:btih:ZXXDAUWYLRUXXBHUYEMS6Q5CE5WA3LVA&dn=LLaM
A
```

It needs to be downloaded through some torrent client - for linux I recommend qBittorrent. Make sure that you are selecting only the 7B model as well as tokenizer.model and tokenizer_checklist.chk



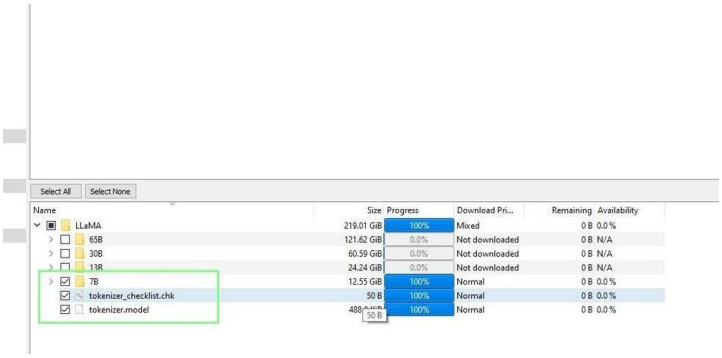


Image credit: Tom's Hardware

copy the downloaded files to /llama.cpp/models/

At this point you need to open the model folder /llama.cpp/models/7B and there should be a .json file (named params.json). Open the json file with any text editor and set the "vocab_size": to 32000 from -1. I'm not really sure why meta has this "-1" value in the file, but it will not work like that.

Now we need to convert the model to GGML FP16 format. Depending on your PC spec this might take some time (and remember, it will not work on the raspberry). To do this, we can use built in utility:

```
python3 convert.py models/7B/
```

after the step is complete, we need to "qantizate" the model. This basically means that all the neural network weights used in the model will be changed from float16 to init8 which will make it much easier for the not-so-powerful machines to handle. I strongly recommend (but it's not needed) to read more about it **here.**

To do the quantization, we also use what's already available:

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

Once the quantization is complete we are almost done with using our PC - we just need to move whole content of "/llama.cpp/models/" to some Pendrive that we would use to transfer it to our Raspberry Pi

Now we need to boot ours raspberry to desktop

Open the terminal and make sure that git is installed

```
sudo apt update && sudo apt install git
```

Clone the repository that we used above

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

Install the same modules that we used on laptop

```
python3 -m pip install torch numpy sentencepiece
```

and also making sure that we have G++ and build essential installed

```
sudo apt install g++ build-essential
```

after having all this done, we need to build the project files by running

```
cd llama.cpp
make
```

now we move all the content of the pendrive to /models/ inside our llama.cpp folder.

all that is left is to run the model. We can use example given by the repo author

```
./examples/chat.sh
```

It takes some time to run, as even the smallest of the LLaMA models is quite big for Raspberry Pi

If you are successful, you should see something like this:

Congratulation, you are now having your own personal AI assistant, running on one of the smallest computers in the world!

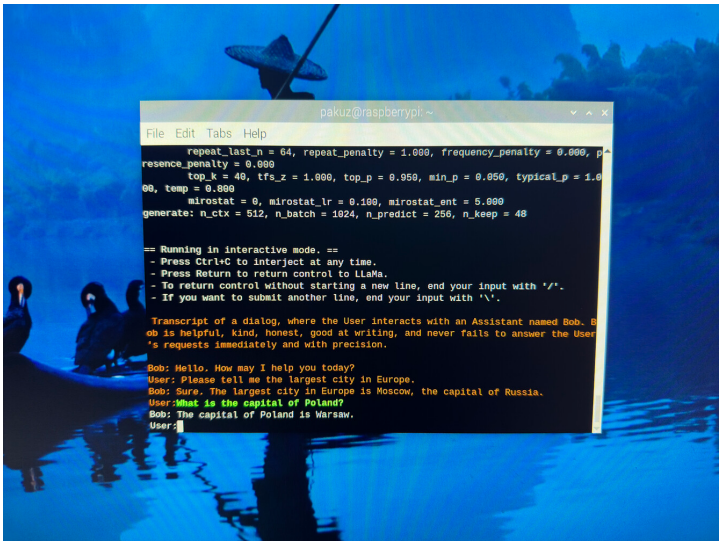


Image credit: Marek Żelichowski

Disclaimer: This is work of many people stitched together

Disclaimer 2: This model is very limited and the computational power of Raspberry Pi is not great - it's done for presentation/educational purposes and running it on PC that we used to convert models would probably be much faster

Report this article

Comments

28 · 4 comments · 1 repost

Like Comment Share

Add a comment...

Most relevant

Marcin Nowak-Liebiediew · 3rd+ 1y
Software Engineer | Generative AI, Rust, Blockchain

How many tokens per second?

Like | Reply · 1 Reply

Marek Żelichowski Author 1y
TM1 / IBM Planning Analytics Expert

Marcin Nowak-Liebiediew around 2.5 - a lot depends on cooling (I have a active cooled case)

Like | Reply

Dmitrii Osipov · 3rd+ (edited) 1y
Sr. Specialist App/Prod Support in AT&T

I was able to run llama2 13b q4_0 on rpi 5 8G, and have to confirm that it runs avg 2.6 tok/s, I'm using a passive northbridge radiator from old M/B (idle 37, max 72 deg.C), but ofc active cooling will be way better. For model I'm using --ctx_size 1152 and -t 4, and it is quite close to maximum perf. of rpi 5 without fine tuning. ...more

Like · 2 | Reply

Jakub Strnad · 3rd+ 1y
Penetration tester | AI enthusiast

i just tried to run 7B Q4M model on rpi5 8GB and it runs quite nice, about 2.3tok/s

Like | Reply

Enjoyed this article?

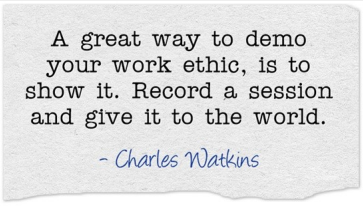
Follow to never miss an update.

Marek Żelichowski

TM1 / IBM Planning Analytics Expert

Follow

More articles for you



How To Create a Dual monitor Timelapse PIP in Linux

Charles W.

My challenge in understanding how 'ls *.c' works

John Freddy Garcia Capera

Published on LinkedIn

Linux Tutorial Series – 113 – Review – revisiting top's columns

Mislav Jurić