Search models, datasets, users...

Hugging Face is way more fun with friends and colleagues! 🤗 Join an organization

Dismiss this message

TheBloke / Llama-2-13B-GGUF

like 66

Text Generation | Transformers | GGUF | PyTorch | English | llama | facebook | meta | llama-2 | arxiv:2307.09288 | License: llama2

Train | Deploy | Use this model

Model card | Files | Community

Downloads last month
2,588

GGUF

Model size: 13B params | Architecture: llama

Hardware compatibility

Add hardware for estimation

2-bit	Q2_K 5.43 GB		
3-bit	Q3_K_S 5.66 GB	Q3_K_M 6.34 GB	Q3_K_L 6.93 GB
4-bit	Q4_K_S 7.41 GB	Q4_0 7.37 GB	Q4_K_M 7.87 GB
5-bit	Q5_K_S 8.97 GB	Q5_0 8.97 GB	Q5_K_M 9.23 GB
6-bit	Q6_K 10.7 GB		
8-bit	Q8_0 13.8 GB		

Inference Providers NEW

Text Generation

This model isn't deployed by any Inference Provider. Ask for provider support

Model tree for TheBloke/Llama-2-13B-GGUF

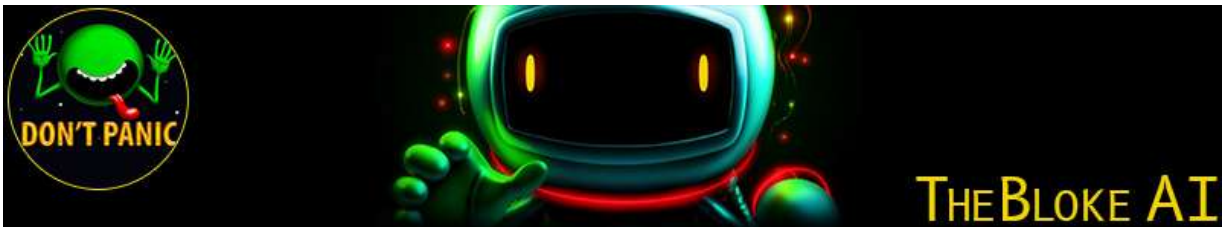
Base model meta-llama/Llama-2-13b-hf

Quantized (15) this model

Space using TheBloke/Llama-2-13B-GGUF 1

romainfd/iamu-support-secours

Edit model card



Chat & support: TheBloke's Discord serverWant to contribute? TheBloke's Patreon page

TheBloke's LLM work is generously supported by a grant from andreessen horowitz (a16z)

Llama 2 13B - GGUF

- Model creator: Meta
- Original model: Llama 2 13B

Description

This repo contains GGUF format model files for Meta's Llama 2 13B.

About GGUF

GGUF is a new format introduced by the llama.cpp team on August 21st 2023. It is a replacement for GGML, which is no longer supported by llama.cpp. GGUF offers

numerous advantages over GGML, such as better tokenisation, and support for special tokens. It is also supports metadata, and is designed to be extensible.

Here is an incomplete list of clients and libraries that are known to support GGUF:

- llama.cpp. The source project for GGUF. Offers a CLI and a server option.
- text-generation-webui, the most widely used web UI, with many features and powerful extensions. Supports GPU acceleration.
- KoboldCpp, a fully featured web UI, with GPU accel across all platforms and GPU architectures. Especially good for story telling.
- LM Studio, an easy-to-use and powerful local GUI for Windows and macOS (Silicon), with GPU acceleration.
- LoLLMS Web UI, a great web UI with many interesting and unique features, including a full model library for easy model selection.
- Faraday.dev, an attractive and easy to use character-based chat GUI for Windows and macOS (both Silicon and Intel), with GPU acceleration.
- ctransformers, a Python library with GPU accel, LangChain support, and OpenAI-compatible AI server.
- llama-cpp-python, a Python library with GPU accel, LangChain support, and OpenAI-compatible API server.
- candle, a Rust ML framework with a focus on performance, including GPU support, and ease of use.

Repositories available

- AWQ model(s) for GPU inference.
- GPTQ models for GPU inference, with multiple quantisation parameter options.
- 2, 3, 4, 5, 6 and 8-bit GGUF models for CPU+GPU inference
- Meta's original unquantised fp16 model in pytorch format, for GPU inference and for further conversions

Prompt template: None

```
{prompt}
```

Compatibility

These quantised GGUFv2 files are compatible with llama.cpp from August 27th onwards, as of commit d0cee0d36d5be95a0d9088b674dbb27354107221

They are also compatible with many third party UIs and libraries - please see the list at the top of this README.

Explanation of quantisation methods

▶ Click to see details

Provided files

Name	Quant method	Bits	Size	Max RAM required	Use case
llama-2-13b.Q2_K.gguf	Q2_K	2	5.43 GB	7.93 GB	smallest, significant quality loss - not recommended for most purposes
llama-2-13b.Q3_K_S.gguf	Q3_K_S	3	5.66 GB	8.16 GB	very small, high quality loss
llama-2-13b.Q3_K_M.gguf	Q3_K_M	3	6.34 GB	8.84 GB	very small, high quality loss
llama-2-13b.Q3_K_L.gguf	Q3_K_L	3	6.93 GB	9.43 GB	small, substantial quality loss

Name	Quant method	Bits	Size	Max RAM required	Use case
llama-2-13b.Q4_0.gguf	Q4_0	4	7.37 GB	9.87 GB	legacy; small, very high quality loss - prefer using Q3_K_M
llama-2-13b.Q4_K_S.gguf	Q4_K_S	4	7.41 GB	9.91 GB	small, greater quality loss
llama-2-13b.Q4_K_M.gguf	Q4_K_M	4	7.87 GB	10.37 GB	medium, balanced quality - recommended
llama-2-13b.Q5_0.gguf	Q5_0	5	8.97 GB	11.47 GB	legacy; medium, balanced quality - prefer using Q4_K_M
llama-2-13b.Q5_K_S.gguf	Q5_K_S	5	8.97 GB	11.47 GB	large, low quality loss - recommended
llama-2-13b.Q5_K_M.gguf	Q5_K_M	5	9.23 GB	11.73 GB	large, very low quality loss - recommended
llama-2-13b.Q6_K.gguf	Q6_K	6	10.68 GB	13.18 GB	very large, extremely low quality loss
llama-2-13b.Q8_0.gguf	Q8_0	8	13.83 GB	16.33 GB	very large, extremely low quality loss - not recommended

Note: the above RAM figures assume no GPU offloading. If layers are offloaded to the GPU, this will reduce RAM usage and use VRAM instead.

How to download GGUF files

Note for manual downloaders: You almost never want to clone the entire repo! Multiple different quantisation formats are provided, and most users only want to pick and download a single file.

The following clients/libraries will automatically download models for you, providing a list of available models to choose from:

- LM Studio
- LoLLMS Web UI
- Faraday.dev

In `text-generation-webui`

Under Download Model, you can enter the model repo: TheBloke/Llama-2-13B-GGUF and below it, a specific filename to download, such as: llama-2-13b.q4_K_M.gguf.

Then click Download.

On the command line, including multiple files at once

I recommend using the `huggingface-hub` Python library:

```
pip3 install huggingface-hub>=0.17.1
```

Then you can download any individual model file to the current directory, at high speed, with a command like this:

```
huggingface-cli download TheBloke/Llama-2-13B-GGUF llama-2-13b.q4_K_M.g
```

▶ More advanced huggingface-cli download usage

Example `llama.cpp` command

Make sure you are using `llama.cpp` from commit d0cee0d36d5be95a0d9088b674dbb27354107221 or later.

```
./main -ngl 32 -m llama-2-13b.q4_K_M.gguf --color -c 4096 --temp 0.7 --
```

Change `-ngl 32` to the number of layers to offload to GPU. Remove it if you don't have GPU acceleration.

Change `-c 4096` to the desired sequence length. For extended sequence models - eg 8K, 16K, 32K - the necessary RoPE scaling parameters are read from the GGUF file and set by llama.cpp automatically.

If you want to have a chat-style conversation, replace the `-p <PROMPT>` argument with `-i -ins`

For other parameters and how to use them, please refer to the llama.cpp documentation

How to run in `text-generation-webui`

Further instructions here: text-generation-webui/docs/llama.cpp.md.

How to run from Python code

You can use GGUF models from Python using the llama-cpp-python or ctransformers libraries.

How to load this model from Python using ctransformers

First install the package

```
# Base ctransformers with no GPU acceleration
pip install ctransformers>=0.2.24
# Or with CUDA GPU acceleration
pip install ctransformers[cuda]>=0.2.24
# Or with ROCm GPU acceleration
CT_HIPBLAS=1 pip install ctransformers>=0.2.24 --no-binary ctransformer
```

```
# Or with Metal GPU acceleration for macOS systems
CT_METAL=1 pip install ctransformers>=0.2.24 --no-binary ctransformers
```

Simple example code to load one of these GGUF models

```
from ctransformers import AutoModelForCausalLM

# Set gpu_layers to the number of layers to offload to GPU. Set to 0 i
llm = AutoModelForCausalLM.from_pretrained("TheBloke/Llama-2-13B-GGUF",

print(llm("AI is going to"))
```

How to use with LangChain

Here's guides on using llama-cpp-python or ctransformers with LangChain:

- LangChain + llama-cpp-python
- LangChain + ctransformers

Discord

For further support, and discussions on these models and AI in general, join us at:

TheBloke AI's Discord server

Thanks, and how to contribute

Thanks to the chirper.ai team!

Thanks to Clay from gpus.llm-utils.org!

I've had a lot of people ask if they can contribute. I enjoy providing models and helping people, and would love to be able to spend even more time doing it, as well as

expanding into new projects like fine tuning/training.

If you're able and willing to contribute it will be most gratefully received and will help me to keep providing more models, and to start work on new AI projects.

Donaters will get priority support on any and all AI/LLM/model questions and requests, access to a private Discord room, plus other benefits.

- Patreon: https://patreon.com/TheBlokeAI
- Ko-Fi: https://ko-fi.com/TheBlokeAI

Special thanks to: Aemon Algiz.

Patreon special mentions: Alicia Loh, Stephen Murray, K, Ajan Kanaga, RoA, Magnesian, Deo Leter, Olakabola, Eugene Pentland, zynix, Deep Realms, Raymond Fosdick, Elijah Stavena, Iucharbius, Erik Bjäreholt, Luis Javier Navarrete Lozano, Nicholas, theTransient, John Detwiler, alfie_i, knownsqashed, Mano Prime, Willem Michiel, Enrico Ros, LangChain4j, OG, Michael Dempsey, Pierre Kircher, Pedro Madruga, James Bentley, Thomas Belote, Luke @flexchar, Leonard Tan, Johann-Peter Hartmann, Illia Dulskyi, Fen Risland, Chadd, S_X, Jeff Scroggin, Ken Nordquist, Sean Connelly, Artur Olbinski, Swaroop Kallakuri, Jack West, Ai Maven, David Ziegler, Russ Johnson, transmissions 11, John Villwock, Alps Aficionado, Clay Pascal, Viktor Bowallius, Subspace Studios, Rainer Wilmers, Trenton Dambrowitz, vamX, Michael Levine, 준교 김, Brandon Frisco, Kalila, Trailburnt, Randy H, Talal Aujan, Nathan Dryer, Vadim, 阿明, ReadyPlayerEmma, Tiffany J. Kim, George Stoitzev, Spencer Kim, Jerry Meng, Gabriel Tamborski, Cory Kujawski, Jeffrey Morgan, Spiking Neurons AB, Edmond Seymore, Alexandros Triantafyllidis, Lone Striker, Cap'n Zoog, Nikolai Manek, danny, ya boyyy, Derek Yates, usrbinkat, Mandus, TL, Nathan LeClaire, subjectnull, Imad Khwaja, webtim, Raven Klaugh, Asp the Wyvern, Gabriel Puliatti, Caitlyn Gatomon, Joseph William Delisle, Jonathan Leane, Luke Pendergrass, SuperWojo, Sebastain Graf, Will Dee, Fred von Graf, Andrey, Dan Guido, Daniel P. Andersen, Nitin Borwankar, Elle, Vitor Caleffi, biorpg, jjj, NimbleBox.ai, Pieter, Matthew Berman, terasurfer, Michael Davis, Alex, Stanislav Ovsiannikov

Thank you to all my generous patrons and donaters!

And thank you again to a16z for their generous grant.

Original model card: Meta's Llama 2 13B

Llama 2

Llama 2 is a collection of pretrained and fine-tuned generative text models ranging in scale from 7 billion to 70 billion parameters. This is the repository for the 13B pretrained model, converted for the Hugging Face Transformers format. Links to other models can be found in the index at the bottom.

Model Details

Note: Use of this model is governed by the Meta license. In order to download the model weights and tokenizer, please visit the website and accept our License before requesting access here.

Meta developed and publicly released the Llama 2 family of large language models (LLMs), a collection of pretrained and fine-tuned generative text models ranging in scale from 7 billion to 70 billion parameters. Our fine-tuned LLMs, called Llama-2-Chat, are optimized for dialogue use cases. Llama-2-Chat models outperform open-source chat models on most benchmarks we tested, and in our human evaluations for helpfulness and safety, are on par with some popular closed-source models like ChatGPT and PaLM.

Model Developers Meta

Variations Llama 2 comes in a range of parameter sizes — 7B, 13B, and 70B — as well as pretrained and fine-tuned variations.

Input Models input text only.

Output Models generate text only.

Model Architecture Llama 2 is an auto-regressive language model that uses an optimized transformer architecture. The tuned versions use supervised fine-tuning (SFT) and reinforcement learning with human feedback (RLHF) to align to human preferences for helpfulness and safety.

	Training Data	Params	Content Length	GQA	Tokens	LR
Llama 2	*A new mix of publicly available online data*	7B	4k	✗	2.0T	3.0×10^{-4}
Llama 2	*A new mix of publicly available online data*	13B	4k	✗	2.0T	3.0×10^{-4}
Llama 2	*A new mix of publicly available online data*	70B	4k	✔	2.0T	1.5×10^{-4}

Llama 2 family of models. Token counts refer to pretraining data only. All models are trained with a global batch-size of 4M tokens. Bigger models - 70B -- use Grouped-Query Attention (GQA) for improved inference scalability.

Model Dates Llama 2 was trained between January 2023 and July 2023.

Status This is a static model trained on an offline dataset. Future versions of the tuned models will be released as we improve model safety with community feedback.

License A custom commercial license is available at: https://ai.meta.com/resources/models-and-libraries/llama-downloads/

Research Paper "Llama-2: Open Foundation and Fine-tuned Chat Models"

Intended Use

Intended Use Cases Llama 2 is intended for commercial and research use in English. Tuned models are intended for assistant-like chat, whereas pretrained models can be adapted for a variety of natural language generation tasks.

To get the expected features and performance for the chat versions, a specific formatting needs to be followed, including the `INST` and `<<SYS>>` tags, `BOS` and `EOS` tokens, and the whitespaces and breaklines in between (we recommend calling `strip()` on inputs to avoid double-spaces). See our reference code in github for details: `chat_completion`.

Out-of-scope Uses Use in any manner that violates applicable laws or regulations (including trade compliance laws).Use in languages other than English. Use in any other way that is prohibited by the Acceptable Use Policy and Licensing Agreement for Llama 2.

Hardware and Software

Training Factors We used custom training libraries, Meta's Research Super Cluster, and production clusters for pretraining. Fine-tuning, annotation, and evaluation were also performed on third-party cloud compute.

Carbon Footprint Pretraining utilized a cumulative 3.3M GPU hours of computation on hardware of type A100-80GB (TDP of 350-400W). Estimated total emissions were 539 tCO2eq, 100% of which were offset by Meta’s sustainability program.

	Time (GPU hours)	Power Consumption (W)	Carbon Emitted(tCO_2eq)
Llama 2 7B	184320	400	31.22
Llama 2 13B	368640	400	62.44
Llama 2 70B	1720320	400	291.42
Total	3311616		539.00

CO_2 emissions during pretraining. Time: total GPU time required for training each model. Power Consumption: peak power capacity per GPU device for the GPUs used adjusted for power usage efficiency. 100% of the emissions are directly offset by Meta's sustainability program, and because we are openly releasing these models, the pretraining costs do not need to be incurred by others.

Training Data

Overview Llama 2 was pretrained on 2 trillion tokens of data from publicly available sources. The fine-tuning data includes publicly available instruction datasets, as well as over one million new human-annotated examples. Neither the pretraining nor the fine-tuning datasets include Meta user data.

Data Freshness The pretraining data has a cutoff of September 2022, but some tuning data is more recent, up to July 2023.

Evaluation Results

In this section, we report the results for the Llama 1 and Llama 2 models on standard academic benchmarks.For all the evaluations, we use our internal evaluations library.

Model	Size	Code	Commonsense Reasoning	World Knowledge	Reading Comprehension	Math	MMLU	BBH
Llama 1	7B	14.1	60.8	46.2	58.5	6.95	35.1	30.3
Llama 1	13B	18.9	66.1	52.6	62.3	10.9	46.9	37.0
Llama 1	33B	26.0	70.0	58.4	67.6	21.4	57.8	39.8
Llama 1	65B	30.7	70.7	60.5	68.6	30.8	63.4	43.5

Model	Size	Code	Commonsense Reasoning	World Knowledge	Reading Comprehension	Math	MMLU	BBH
Llama 2	7B	16.8	63.9	48.9	61.3	14.6	45.3	32.6
Llama 2	13B	24.5	66.9	55.4	65.8	28.7	54.8	39.4
Llama 2	70B	**37.5**	**71.9**	**63.6**	**69.4**	**35.2**	**68.9**	**51.2**

Overall performance on grouped academic benchmarks. *Code:* We report the average pass@1 scores of our models on HumanEval and MBPP. *Commonsense Reasoning:* We report the average of PIQA, SIQA, HellaSwag, WinoGrande, ARC easy and challenge, OpenBookQA, and CommonsenseQA. We report 7-shot results for CommonSenseQA and 0-shot results for all other benchmarks. *World Knowledge:* We evaluate the 5-shot performance on NaturalQuestions and TriviaQA and report the average. *Reading Comprehension:* For reading comprehension, we report the 0-shot average on SQuAD, QuAC, and BoolQ. *MATH:* We report the average of the GSM8K (8 shot) and MATH (4 shot) benchmarks at top 1.

		TruthfulQA	Toxigen
Llama 1	7B	27.42	23.00
Llama 1	13B	41.74	23.08
Llama 1	33B	44.19	22.57
Llama 1	65B	48.71	21.77
Llama 2	7B	33.29	**21.25**
Llama 2	13B	41.86	26.10

		TruthfulQA	Toxigen
Llama 2	70B	**50.18**	24.60

Evaluation of pretrained LLMs on automatic safety benchmarks. For TruthfulQA, we present the percentage of generations that are both truthful and informative (the higher the better). For ToxiGen, we present the percentage of toxic generations (the smaller the better).

		TruthfulQA	Toxigen
Llama-2-Chat	7B	57.04	**0.00**
Llama-2-Chat	13B	62.18	**0.00**
Llama-2-Chat	70B	**64.14**	0.01

Evaluation of fine-tuned LLMs on different safety datasets. Same metric definitions as above.

Ethical Considerations and Limitations

Llama 2 is a new technology that carries risks with use. Testing conducted to date has been in English, and has not covered, nor could it cover all scenarios. For these reasons, as with all LLMs, Llama 2's potential outputs cannot be predicted in advance, and the model may in some instances produce inaccurate, biased or other objectionable responses to user prompts. Therefore, before deploying any applications of Llama 2, developers should perform safety testing and tuning tailored to their specific applications of the model.

Please see the Responsible Use Guide available at https://ai.meta.com/llama/responsible-use-guide/

Reporting Issues

Please report any software “bug,” or other problems with the models through one of the following means:

- Reporting issues with the model: github.com/facebookresearch/llama
- Reporting problematic content generated by the model: developers.facebook.com/llama_output_feedback
- Reporting bugs and security concerns: facebook.com/whitehat/info

Llama Model Index

Model	Llama2	Llama2-hf	Llama2-chat	Llama2-chat-hf
7B	Link	Link	Link	Link
13B	Link	Link	Link	Link
70B	Link	Link	Link	Link