Chat & support: my new Discord server

Want to contribute? TheBloke's Patreon page

Concept of Mind's Flan Open Llama 7B GGML

These files are GGML format model files for Concept of Mind's Flan Open Llama 7B.

GGML files are for CPU + GPU inference using llama.cpp and libraries and UIs which support this format, such as:

- text-generation-webui
- KoboldCpp
- LoLLMS Web UI
- llama-cpp-python
- ctransformers

Repositories available

- 4-bit GPTQ models for GPU inference
- 2, 3, 4, 5, 6 and 8-bit GGML models for CPU+GPU inference
- Unquantised fp16 model in pytorch format, for GPU inference and for further conversions

Compatibility

Original llama.cpp quant methods: `q4_0, q4_1, q5_0, q5_1, q8_0`

I have quantized these 'original' quantisation methods using an older version of llama.cpp so that they remain compatible with llama.cpp as of May 19th, commit `2d5db48`.

These are guaranteed to be compatbile with any UIs, tools and libraries released since late May.

New k-quant methods: `q2_K, q3_K_S, q3_K_M, q3_K_L, q4_K_S, q4_K_M, q5_K_S, q6_K`

These new quantisation methods are compatible with llama.cpp as of June 6th, commit `2d43387`.

They are now also compatible with recent releases of text-generation-webui, KoboldCpp, llama-cpp-python and ctransformers. Other tools and libraries may or may not be compatible - check their documentation if in doubt.

Explanation of the new k-quant methods

The new methods available are:

- GGML_TYPE_Q2_K - "type-1" 2-bit quantization in super-blocks containing 16 blocks, each block having 16 weight. Block scales and mins are quantized with 4 bits. This ends up effectively using 2.5625 bits per weight (bpw)
- GGML_TYPE_Q3_K - "type-0" 3-bit quantization in super-blocks containing 16 blocks, each block having 16 weights. Scales are quantized with 6 bits. This end up using 3.4375 bpw.
- GGML_TYPE_Q4_K - "type-1" 4-bit quantization in super-blocks containing 8 blocks, each block having 32 weights. Scales and mins are quantized with 6 bits. This ends up using 4.5 bpw.
- GGML_TYPE_Q5_K - "type-1" 5-bit quantization. Same super-block structure as GGML_TYPE_Q4_K resulting in 5.5 bpw
- GGML_TYPE_Q6_K - "type-0" 6-bit quantization. Super-blocks with 16 blocks, each block having 16 weights. Scales are quantized with 8 bits. This ends up using 6.5625 bpw
- GGML_TYPE_Q8_K - "type-0" 8-bit quantization. Only used for quantizing intermediate results. The difference to the existing Q8_0 is that the block size is 256. All 2-6 bit dot products are implemented for this quantization type.

Refer to the Provided Files table below to see what files use which methods, and how.

Provided files

Name	Quant method	Bits	Size	Max RAM required	Use case
flan-openllama-7b.ggmlv3.q2_K.bin	q2_K	2	2.87 GB	5.37 GB	New k-quant method. Uses GGML_TYPE_Q4_K for the attention.vw and feed_forward.w2 tensors, GGML_TYPE_Q2_K for the other tensors.
flan-openllama-7b.ggmlv3.q3_K_L.bin	q3_K_L	3	3.60 GB	6.10 GB	New k-quant method. Uses GGML_TYPE_Q5_K for the attention.wv, attention.wo,

Name	Quant method	Bits	Size	Max RAM required	Use case
					and feed_forward.w2 tensors, else GGML_TYPE_Q3_K
flan-openllama-7b.ggmlv3.q3_K_M.bin	q3_K_M	3	3.28 GB	5.78 GB	New k-quant method. Uses GGML_TYPE_Q4_K for the attention.wv, attention.wo, and feed_forward.w2 tensors, else GGML_TYPE_Q3_K
flan-openllama-7b.ggmlv3.q3_K_S.bin	q3_K_S	3	2.95 GB	5.45 GB	New k-quant method. Uses GGML_TYPE_Q3_K for all tensors
flan-openllama-7b.ggmlv3.q4_0.bin	q4_0	4	3.79 GB	6.29 GB	Original llama.cpp quant method, 4-bit.
flan-openllama-7b.ggmlv3.q4_1.bin	q4_1	4	4.21 GB	6.71 GB	Original llama.cpp quant method, 4-bit. Higher accuracy than q4_0 but not as high as q5_0. However has quicker inference than q5 models.
flan-openllama-7b.ggmlv3.q4_K_M.bin	q4_K_M	4	4.08 GB	6.58 GB	New k-quant method. Uses GGML_TYPE_Q6_K for half of the attention.wv and feed_forward.w2 tensors, else GGML_TYPE_Q4_K
flan-openllama-7b.ggmlv3.q4_K_S.bin	q4_K_S	4	3.83 GB	6.33 GB	New k-quant method. Uses GGML_TYPE_Q4_K for all tensors
flan-openllama-7b.ggmlv3.q5_0.bin	q5_0	5	4.63 GB	7.13 GB	Original llama.cpp quant method, 5-bit. Higher accuracy, higher resource usage and slower inference.
flan-openllama-7b.ggmlv3.q5_1.bin	q5_1	5	5.06 GB	7.56 GB	Original llama.cpp quant method, 5-bit. Even higher accuracy, resource usage and slower inference.
flan-openllama-7b.ggmlv3.q5_K_M.bin	q5_K_M	5	4.78 GB	7.28 GB	New k-quant method. Uses GGML_TYPE_Q6_K for half of the attention.wv and feed_forward.w2 tensors, else GGML_TYPE_Q5_K
flan-openllama-7b.ggmlv3.q5_K_S.bin	q5_K_S	5	4.65 GB	7.15 GB	New k-quant method. Uses GGML_TYPE_Q5_K for all tensors
flan-openllama-7b.ggmlv3.q6_K.bin	q6_K	6	5.53 GB	8.03 GB	New k-quant method. Uses GGML_TYPE_Q8_K - 6-bit quantization - for all tensors
flan-openllama-7b.ggmlv3.q8_0.bin	q8_0	8	7.16 GB	9.66 GB	Original llama.cpp quant method, 8-bit. Almost indistinguishable from

Name	Quant method	Bits	Size	Max RAM required	Use case
					float16. High resource use and slow. Not recommended for most users.

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 run in `llama.cpp`

I use the following command line; adjust for your tastes and needs:

```
l -p "### Instruction: Write a story about llamas\n### Response:"
```

If you're able to use full GPU offloading, you should use `-t 1` to get best performance.

If not able to fully offload to GPU, you should use more cores. Change `-t 10` to the number of physical CPU cores you have, or a lower number depending on what gives best performance.

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

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

How to run in `text-generation-webui`

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

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!

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: Luke from CarbonQuill, Aemon Algiz, Dmitriy Samsonov.

Patreon special mentions: Mano Prime, Fen Risland, Derek Yates, Preetika Verma, webtim, Sean Connelly, Alps Aficionado, Karl Bernard, Junyu Yang, Nathan LeClaire, Chris McCloskey, Lone Striker, Asp the Wyvern, Eugene Pentland, Imad Khwaja, trip7s trip, WelcomeToTheClub, John Detwiler, Artur Olbinski, Khalefa Al-Ahmad, Trenton Dambrowitz, Talal Aujan, Kevin Schuppel, Luke Pendergrass, Pyrater, Joseph William Delisle, terasurfer , vamX, Gabriel Puliatti, David Flickinger, Jonathan Leane, Iucharbius , Luke, Deep Realms, Cory Kujawski, ya boyyy, Illia Dulskyi, senxiiz, Johann-Peter Hartmann, John Villwock, K, Ghost , Spiking Neurons AB, Nikolai Manek, Rainer Wilmers, Pierre Kircher, biorpg, Space Cruiser, Ai Maven, subjectnull, Willem Michiel, Ajan Kanaga, Kalila, chris gileta, Oscar Rangel.

Thank you to all my generous patrons and donaters!

Original model card: Concept of Mind's Flan Open Llama 7B

No original model card was provided.