Search models, datasets, users...

TheBloke/TinyLlama-1.1B-1T-OpenOrca-GGUF

like 9

Transformers | GGUF | Open-Orca/OpenOrca | bigcode/starcoderdata | cerebras/SlimPajama-627B | English | llama | text-generation-inference | License: other

Train | Deploy | Use in Transformers

Model card | Files | Community

Downloads last month
102

Finetuned from jeff31415/TinyLlama-1.1B-1T-OpenOrca

Datasets used to train TheBloke/TinyLlama-1.1B-1T-OpenOrca-GGUF

Open-Orca/OpenOrca
Viewer • Updated Oct 21 • 31.3k • 924

bigcode/starcoderdata
Viewer • Updated May 16 • 15k • 220

cerebras/SlimPajama-627B
Viewer • Updated Jul 7 • 4.74k • 227

Edit model card

Chat & support: TheBloke's Discord server

Want to contribute? TheBloke's Patreon page

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

Tinyllama 1.1B 1T Openorca - GGUF

- Model creator: jeff zhao
- Original model: Tinyllama 1.1B 1T Openorca

Description

This repo contains GGUF format model files for jeff zhao's Tinyllama 1.1B 1T Openorca.

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.

Here is an incomplate 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
- jeff zhao's original unquantised fp16 model in pytorch format, for GPU inference and for further conversions

Prompt template: ChatML

```
<|im_start|>system
{system_message}<|im_end|>
<|im_start|>user
{prompt}<|im_end|>
<|im_start|>assistant
```

Compatibility

These quantised GGUFv2 files are compatible with llama.cpp from August 27th onwards, as of commit d0cee0d

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
tinyllama-1.1b-1t-openorca.Q2_K.gguf	Q2_K	2	0.48 GB	2.98 GB	smallest, significant quality loss - not recommended for most purposes
tinyllama-1.1b-1t-openorca.Q3_K_S.gguf	Q3_K_S	3	0.50 GB	3.00 GB	very small, high quality loss
tinyllama-1.1b-1t-openorca.Q3_K_M.gguf	Q3_K_M	3	0.55 GB	3.05 GB	very small, high quality loss
tinyllama-1.1b-1t-openorca.Q3_K_L.gguf	Q3_K_L	3	0.59 GB	3.09 GB	small, substantial quality loss
tinyllama-1.1b-1t-openorca.Q4_0.gguf	Q4_0	4	0.64 GB	3.14 GB	legacy; small, very high quality loss - prefer using Q3_K_M
tinyllama-1.1b-1t-openorca.Q4_K_S.gguf	Q4_K_S	4	0.64 GB	3.14 GB	small, greater quality loss
tinyllama-1.1b-1t-openorca.Q4_K_M.gguf	Q4_K_M	4	0.67 GB	3.17 GB	medium, balanced quality - recommended
tinyllama-1.1b-1t-openorca.Q5_0.gguf	Q5_0	5	0.77 GB	3.27 GB	legacy; medium, balanced quality - prefer using Q4_K_M
tinyllama-1.1b-1t-openorca.Q5_K_S.gguf	Q5_K_S	5	0.77 GB	3.27 GB	large, low quality loss - recommended
tinyllama-1.1b-1t-openorca.Q5_K_M.gguf	Q5_K_M	5	0.78 GB	3.28 GB	large, very low quality loss - recommended
tinyllama-1.1b-1t-openorca.Q6_K.gguf	Q6_K	6	0.90 GB	3.40 GB	very large, extremely low quality loss
tinyllama-1.1b-1t-openorca.Q8_0.gguf	Q8_0	8	1.17 GB	3.67 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/TinyLlama-1.1B-1T-OpenOrca-GGUF and below it, a specific filename to download, such as: tinyllama-1.1b-1t-openorca.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
```

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

```
huggingface-cli download TheBloke/TinyLlama-1.1B-1T-OpenOrca-GGUF
```

▶ More advanced huggingface-cli download usage

Example `llama.cpp` command

Make sure you are using `llama.cpp` from commit d0cee0d or later.

```
--temp 0.7 --repeat_penalty 1.1 -n -1 -p "<|im_start|>system\n{sys
```

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

Change `-c 2048` 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 in Python code, using ctransformers

First install the package

Run one of the following commands, according to your system:

```
# Base ctransformers with no GPU acceleration
pip install ctransformers
# Or with CUDA GPU acceleration
pip install ctransformers[cuda]
# Or with AMD ROCm GPU acceleration (Linux only)
CT_HIPBLAS=1 pip install ctransformers --no-binary ctransformers
# Or with Metal GPU acceleration for macOS systems only
CT_METAL=1 pip install ctransformers --no-binary ctransformers
```

Simple ctransformers example code

```
our system.
1.1b-1t-openorca.Q4_K_M.gguf", model_type="llama", gpu_layers=50)
```

How to use with LangChain

Here are guides on using llama-cpp-python and 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

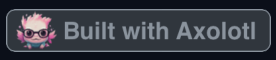
Special thanks to: Aemon Algiz.

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

Thank you to all my generous patrons and donaters!

And thank you again to a16z for their generous grant.

Original model card: jeff zhao's Tinyllama 1.1B 1T Openorca



Base model:

PY007/TinyLlama-1.1B-intermediate-step-480k-1T

Dataset:

Fine tuned on OpenOrca GPT4 subset for 1 epoch,Using CHATML format

Model License:

Apache 2.0, following the TinyLlama base model.

Hardware and training details:

Hardware: 1*RTX A5000, ~16 hours to complete 1 epoch. GPU from autodl.com, cost around $3 for this finetuning. https://wandb.ai/jeff200402/TinyLlama-Orca?workspace= for more details.

Company