3-28-23 I will not be updating this guide anymore. If someone wants to make a better guide going forward, please be my guest. I've heard the one-click installers might work now, making this all much simpler. A guide for CPU inference would be good as well. --v2-anon

3-26-23 New weights are required as of March 26th called "LLaMA-HF (3-26-23)". The old "LLaMA-HFv2" weights no longer work 16bit version: 3-26 16bit Torrent, 7B 4bit version: 3-26 4bit Torrent for 7B, 13B-65B 4bit version 3-26 4bit 128g Torrent for 13B, 30B, & 65B (You probably want this last one!) [Good news! These weights are 2-5x faster on GPU!]

LLaMA ~~Int8~~ 4bit ChatBot Guide v2

Want to fit the most model in the amount of VRAM you have, if that's a little or a lot? Look no further.

FAQ

Q: Doesn't 4bit have worse output performance than 8bit or 16bit?
A: No, **GPTQ 4bit has effectively NO output quality loss** compared to baseline uncompressed fp16. Additionally, GPTQ 3bit (coming soon) has negligible output quality loss which goes down as model size goes up!
Q: How many tokens per second is 2it/s?
A: Tokens per "iteration" (it) depends on the implementation. In ooba's webUI 1 "it" is 8 words/tokens. So 2it/s is 16 tokens/words per second!

Table of Contents

8-bit Model Requirements for LLaMA

Model	VRAM Used	Minimum Total VRAM	Card examples	RAM/Swap to Load*
LLaMA-7B	9.2GB	10GB	3060 12GB, RTX 3080 10GB, RTX 3090	24 GB
LLaMA-13B	16.3GB	20GB	RTX 3090 Ti, RTX 4090	32GB
LLaMA-30B	36GB	40GB	A6000 48GB, A100 40GB	64GB
LLaMA-65B	74GB	80GB	A100 80GB	128GB

**System RAM (not VRAM) required to load the model, in addition to having enough VRAM. NOT required to RUN the model. You can use swap space if you do not have enough RAM.*

4-bit Model Requirements for LLaMA

Model	Model Size	Minimum Total VRAM	Card examples	RAM/Swap to Load*
LLaMA-7B	3.5GB	6GB	RTX 1660, 2060, AMD 5700xt, RTX 3050, 3060	16 GB
LLaMA-13B	6.5GB	10GB	AMD 6900xt, RTX 2060 12GB, 3060 12GB, 3080, A2000	32 GB
LLaMA-30B	15.8GB	20GB	RTX 3080 20GB, A4500, A5000, 3090, 4090, 6000, Tesla V100	64 GB
LLaMA-65B	31.2GB	40GB	A100 40GB, 2x3090, 2x4090, A40, RTX A6000, 8000, Titan Ada	128 GB

**System RAM (not VRAM) required to load the model, in addition to having enough VRAM. NOT required to RUN the model. You can use swap space if you do not have enough RAM.*

Choosing 8bit or 4bit

8bit: Easier setup, lower output quality (due to RTN), **recommended for first-timers**
4bit: Faster, smaller, higher output quality (due to GPTQ), but more difficult setup

It's recommended to start with setting up 8bit. Once 8bit is working you can come back to read "BONUS 3" on setting up 4bit.

To continue with 8bit setup, just keep reading.

8bit LLaMA Installation (start here)

Install text-generation-webui

All you need to get started is to install https://github.com/oobabooga/text-generation-webui using **"Installation option 1: conda"**.

Acquiring the CORRECT ~~HFv2~~ "3-26-23" Model Weights

But wait, there's one more thing. You need the MODEL WEIGHTS. But you don't need just any LLaMA model weights.

The original leaked weights won't work. You need the "3-26-23" (HuggingFace Safe Tensor) converted model weights.
You can get them by using this torrent or this magnet link
**If you have the old weights and really want to convert them yourself, scroll to the bottom of this guide for instructions.*

How to tell if you have the "3-26-23" Converted Weights

If you already have some weights and are not sure if they're the right ones, here's how you can tell.

The **WRONG** original leaked weights have filenames that look like:
`consolidated.00.pth`
`consolidated.01.pth`
OR
`pytorch_model-00001-of-00033.bin`
`pytorch_model-00002-of-00033.bin`

The **CORRECT** "HF Converted" weights have filenames that look like:
`model-00001-of-00033.safetensors`
`model-00002-of-00033.safetensors`

So you got the right weights, now what?

Put them in `text-generation-webui/models/LLaMA-7B`

Install bitsandbytes for 8bit support (skip this on Linux)

Install bitsandbytes (Windows only)

1. Download these 2 dll files:
https://github.com/DeXtmL/bitsandbytes-win-prebuilt/raw/main/libbitsandbytes_cpu.dll
https://github.com/DeXtmL/bitsandbytes-win-prebuilt/raw/main/libbitsandbytes_cuda116.dll
2. Move those files into `C:\Users\xxx\miniconda3\envs\textgen\lib\site-packages\bitsandbytes\`
3. Now edit `bitsandbytes\cuda_setup\main.py` with these:
4. Change `ct.cdll.LoadLibrary(binary_path)` to `ct.cdll.LoadLibrary(str(binary_path))` **two times** in the file.
5. Then replace `if not torch.cuda.is_available(): return 'libsbitsandbytes_cpu.so', None, None, None, None`
with `if torch.cuda.is_available(): return 'libbitsandbytes_cuda116.dll', None, None, None, None`

Load the webUI

Now, from a command prompt in the text-generation-webui directory, run:
`conda activate textgen`
`python server.py --model LLaMA-7B --load-in-8bit --no-stream` * and GO!

**Replace `LLaMA-7B` with the model you're using in the command above.*

Okay, I got 8bit working now take me to the 4bit setup instructions.

Troubleshooting

I'm getting CUDA errors

Install bitsandbytes (Windows only)

1. Download these 2 dll files:
https://github.com/DeXtmL/bitsandbytes-win-prebuilt/raw/main/libbitsandbytes_cpu.dll
https://github.com/DeXtmL/bitsandbytes-win-prebuilt/raw/main/libbitsandbytes_cuda116.dll
2. Move those files into:
`KoboldAI\miniconda3\python\Lib\site-packages\bitsandbytes` for Kobold
or `C:\Users\xxx\miniconda3\envs\textgen\lib\site-packages\bitsandbytes\` for ooba's text-generation-webui
3. Now edit `bitsandbytes\cuda_setup\main.py` with these:
4. Change `ct.cdll.LoadLibrary(binary_path)` to `ct.cdll.LoadLibrary(str(binary_path))` **two times** in the file.
5. Then replace `if not torch.cuda.is_available(): return 'libsbitsandbytes_cpu.so', None, None, None, None`
with `if torch.cuda.is_available(): return 'libbitsandbytes_cuda116.dll', None, None, None, None`

After that you should be able to load models with 8-bit precision.

Help I got an OOM error (or something else)

If you run into trouble, ask for help at https://github.com/oobabooga/text-generation-webui/issues/147

BONUS: KoboldAI Support for LLaMA

KoboldAI GitHub: https://github.com/KoboldAI/KoboldAI-Client

KoboldAI also requires the HFv2 converted model weights in the torrent above.
Simply place the weights in `KoboldAI/models/Facebook_LLaMA-7b/` (or `13b` `30b` `65b` depending on your model)
Until KoboldAI merges the patch to support these weights you'll have to patch it yourself. Follow the steps below to do that.

How to patch KoboldAI for LLaMA support

Install KoboldAI 8bit
Get KoboldAI 8bit from: https://github.com/ebolam/KoboldAI/tree/8bit
Install it using `git clone -b 8bit https://github.com/ebolam/KoboldAI/`
(You cannot use the windows installer or zip file. You must install using git clone or it will not work.)

This enables 8bit/int8 support for all Kobold models, not just LLaMA.

Run KoboldAI
Run KoboldAI as normal and select `AI > load a Model from its directory > Facebook_LLaMA-7b`
Enjoy!

If you have issues with KoboldAI, go to their Discord: https://koboldai.org/discord

BONUS 2: TavernAI with LLaMA

TavernAI GitHub: https://github.com/TavernAI/TavernAI
How to connect Tavern to Kobold with LLaMA
(Tavern relies on Kobold to run LLaMA. Follow all of the KoboldAI steps first.)

1. With KoboldAI running and the LLaMA model loaded in the KoboldAI webUI, open TavernAI.
2. Ensure TavernAI's API setting is pointing at your local machine (127.0.0.1).
3. Pick a character and start chatting.

That's it! No further configuration is necessary. Enjoy!

If you have issues with TavernAI, go to their Discord: https://discord.com/invite/zmK2gmr45t

BONUS 3: 4bit LLaMA (Basic Setup)

4bit has NO reduction in output quality vs 16bit (thanks to GPTQ) while substantially reducing VRAM requirements

How to install 4bit LLaMA w/ webUI

1. Verify that you have 8bit LLaMA working in ooba's webUI per the instructions above, first.*
**(If you have under 10GB of VRAM then just skip straight to step 2)*
2. Acquire the latest 4bit weights from:
3-23-26 4bit Torrent Link (Use these for 7B only)
3-23-26 4bit Magnet Link (Use these for 7B only)
3-23-26 4bit 128g Torrent Link (Use these for 13B, 30B, 65B)
3-23-26 4bit 128g Magnet Link (Use these for 13B, 30B, 65B)
3. (Windows only) Install Visual Studio 2019 with C++ build-tools before completing 4-bit setup below, per this comment on the 4bit repo
4. Open a command line in the text-generation-webui directory and run `conda activate textgen`
5. Now continue to follow the installation instructions at https://github.com/oobabooga/text-generation-webui/wiki/LLaMA-model#4-bit-mode while running all commands from inside the (textgen) conda environment
6. Enjoy 4bit LLaMA with a webUI

Appendix

List of Torrents

You need #3 for 16bit or 8bit and BOTH #3 & #4 for 4bit!
#1 is only if you want to convert HF weights yourself

1. Original Facebook LLaMA Weights

Torrent: https://files.catbox.moe/oyy6vh.torrent
Magnet: `magnet:?xt=urn:btih:b8287ebfa04f879b048d4d4404108cf3e8014352&dn=LLaMA&tr=udp%3a%2f%2ftracker.opentrackr.org%3a1337%2fannounce`

2. Updated 3-26-23 Converted 16bit/8bit LLaMA Weights

3-26-23 weights Torrent Link
3-26-23 weights Magnet Link
~~HFv2 Torrent~~
~~HFv2 Magnet:~~

3. 4bit pre-quantized experimental GPTQ LLaMA Weights

3-26-23 4bit Torrent Link (Use these for 7B only)
3-26-23 4bit Magnet Link (Use these for 7B only)
3-26-23 4bit 128g Torrent Link (Use these for 13B, 30B, 65B)
3-26-23 4bit 128g Magnet Link (Use these for 13B, 30B, 65B)
~~Old HFv2 4bit Torrent~~
~~Old HFv2 4bit Magnet~~
~~Oldest HF 4bit Torrent~~
~~Oldest HF 4bit Magnet Link~~
~~LLaMA-7B int4 DDL: https://huggingface.co/decapoda-research/llama-7b-hf-int4/resolve/main~~
~~LLaMA-13B int4 DDL: https://huggingface.co/decapoda-research/llama-13b-hf-int4/tree/main~~
~~LLaMA-30B int4 DDL: https://huggingface.co/decapoda-research/llama-30b-hf-int4/tree/main~~
~~LLaMA-65B int4 DDL: https://huggingface.co/decapoda-research/llama-65b-hf-int4/tree/main~~