



Rensselaer

why not change the world?®

A Guide into Open-Source Large Language Models and Fine-Tuning Techniques

Inwon Kang, Tripp Lyons | 10/18/2023

Closed Source

- Requires payment per token.
- Difficult to determine model architecture/training data for comparison.
- Usually performs better.
- Notable Sources: OpenAI (GPT-4, GPT-3.5), Anthropic (Claude-2, Claude-1)

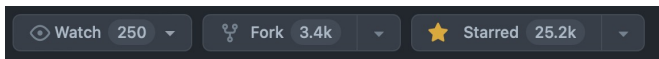
Open Source

- Weights are freely available.
- Only need infrastructure to use.
- May have restrictions for commercial use. (LLaMA)
- Weights can be modified (fine-tuned) for no cost.
- Notable sources: Meta (LLaMA), Google (T5, UL2), BigScience (Bloom)

Oobabooga

<https://github.com/oobabooga/text-generation-webui>

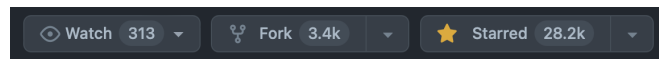
- Clean UI, lots of options.
- Supports many kinds of LLMs.
- Supports OpenAI style API and more customized API.
- 25.2K start on Github.



FastChat

<https://github.com/lm-sys/FastChat/tree/main>

- Built by creators of Vicuna.
- Includes “Chatbot Arena”.
- Supports OpenAI style API.
- More scalable.
- 28.2K stars on Github.



WHY USE A SERVER?

- Scalable for multi-user case
 - If multiple users need to query the model, it will not be scalable.
 - Model will take space on GPU even for inference.
- Easy to switch out to OpenAI
 - If using the OpenAI API, the main code stays the same, only the endpoint needs to be switched out.

OOBABOOGA/TEXT-GENERATION-WEBUI

Training Session

Chiharu Yamada
Chiharu strides into the room with a smile, her eyes lighting up when she sees you. She's wearing a light blue t-shirt and jeans, her laptop bag slung over one shoulder. She takes a seat next to you, her enthusiasm palpable in the air.
Hey! I'm so excited to finally meet you. I've heard so many great things about you and I'm eager to pick your brain about computers. I'm sure you have a wealth of knowledge that I can learn from. *She grins, eyes twinkling with excitement* Let's get started!

You
What is your message to the world?

Chiharu Yamada
Ooh, that's a deep question! Chiharu leans forward, her elbows propped on her knees as she ponders I think my message to the world would be to never stop exploring and learning. Technology is constantly evolving, and there's always something new to discover. Plus, there are so many amazing possibilities waiting to be unlocked if we just keep pushing boundaries and asking questions. *Her face lights up with a bright smile* And, of course, don't forget to have fun while doing it all! Life is an adventure, after all.

You
What is the best text editor and why is it vim?

Chiharu Yamada
Oh my gosh, you're totally right! laughs Hands down, no contest—it's gotta be Vim! It's like the Swiss Army Knife of text editors. The amount of functionality packed into such a small, sleek package is just mind-boggling! *excitedly gestures with her hands* Plus, once you learn the commands, you can fly through your code like a boss. No other text editor even comes close! *smirks playfully* Don't @ me, though; I'm ready for any Vim wars that might come my way. 😊

Input

Stop Generate Continue

Impersonate Regenerate Remove last

Chat Default Notebook Parameters Model Training Session

Generation Character Instruction template Chat history Upload character

Filter by loader: All

simple-1

max_new_tokens: 200

temperature: 0.7

top_p: 0.9

top_k: 20

typical_p: 1

epsilon_cutoff: 0

eta_cutoff: 0

ifs: 1

top_a: 0

repetition_penalty: 1.15

repetition_penalty_range: 0

encoder_repetition_penalty: 1

no_repeat_ngram_size: 0

min_length: 0

Seed (-1 for random): -1

do_sample

guidance_scale: 1

Negative prompt

mirostat_mode: 0 (mode=1 is for llama.cpp only)

mirostat_tau: 5

mirostat_eta: 0.1

penalty_alpha: 0

num_beams: 1

length_penalty: 1

early_stopping

Truncate the prompt up to this length: 4096

Custom stopping strings: "!", "nYou"

Learn more

Expand max_new_tokens to the available context length. auto_max_new_tokens

Forces the model to never end the generation prematurely. Ban the eos_token

Disabling this can make the replies more creative. Add the bos_token to the beginning of prompts

Some specific models need this unset. Skip special tokens

Activate text streaming

CURRENT OPEN SOURCE “STATE OF ART”

Model	License	Commercial use?	Pretraining length [tokens]	Leaderboard score
Falcon-7B	Apache 2.0	✓	1,500B	47.01
MPT-7B	Apache 2.0	✓	1,000B	48.7
Llama-7B	Llama license	✗	1,000B	49.71
Llama-2-7B	Llama 2 license	✓	2,000B	54.32
Llama-33B	Llama license	✗	1,500B	*
Llama-2-13B	Llama 2 license	✓	2,000B	58.67
mpt-30B	Apache 2.0	✓	1,000B	55.7
Falcon-40B	Apache 2.0	✓	1,000B	61.5
Llama-65B	Llama license	✗	1,500B	62.1
Llama-2-70B	Llama 2 license	✓	2,000B	*
Llama-2-70B-chat*	Llama 2 license	✓	2,000B	66.8

PARAMETER-EFFICIENT FINE-TUNING (PEFT)

- Represent changes to a model's weights using less weights.
- Libraries: PEFT, Adapter-Transformers.
- Uses less memory during training.
- Uses less disk space to store the changes.
- Most common method is LoRA (Low-Rank Adaptation):
 - Uses two small matrices to generate a large but low-rank matrix that is added to the weights

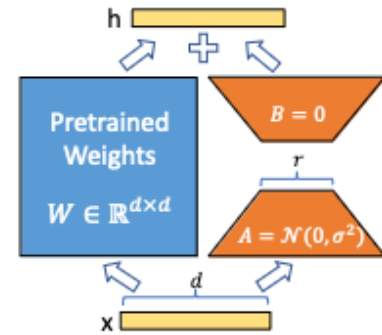


Figure 1: Our reparametrization. We only train A and B .

0.15	-0.14	-0.21	0.612	0.3	-0.14	0.1	-0.44	0.04	1.42
-0.22	0.204	0.308	-0.86	-0.42	0.201	-0.92	0.1	1.62	-1.33
-0.30	-0.16	0.634	0.147	0.46	0.38				
-0.07	-0.2	0.246	0.523	0.5	0.14				
ΔW				B		A			
Shape: (200, 200)				Shape: (200, 2)		Shape: (2, 200)			

(SCALAR) QUANTIZATION

- Use smaller data types than float32 (int8, int4, or even 1 bit).
- Libraries: bitsandbytes, AutoGPTQ, ExLlama.
- Pros:
 - Less memory during inference.
 - Less memory during training (using QLoRA)
- Cons:
 - Lower model accuracy

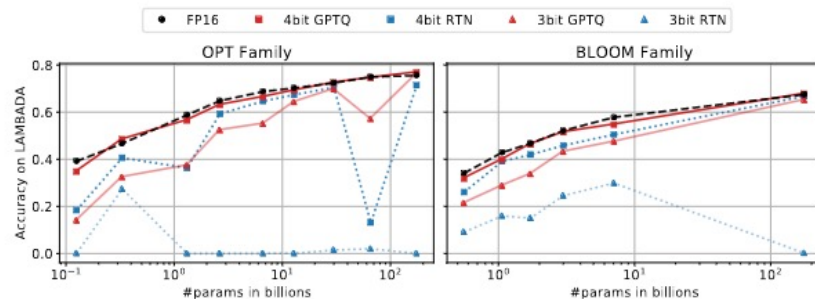


Figure 3: The accuracy of OPT and BLOOM models post-GPTQ, measured on LAMBADA.

<https://arxiv.org/abs/2210.17323>

COMMON INFERENCE PARAMETERS

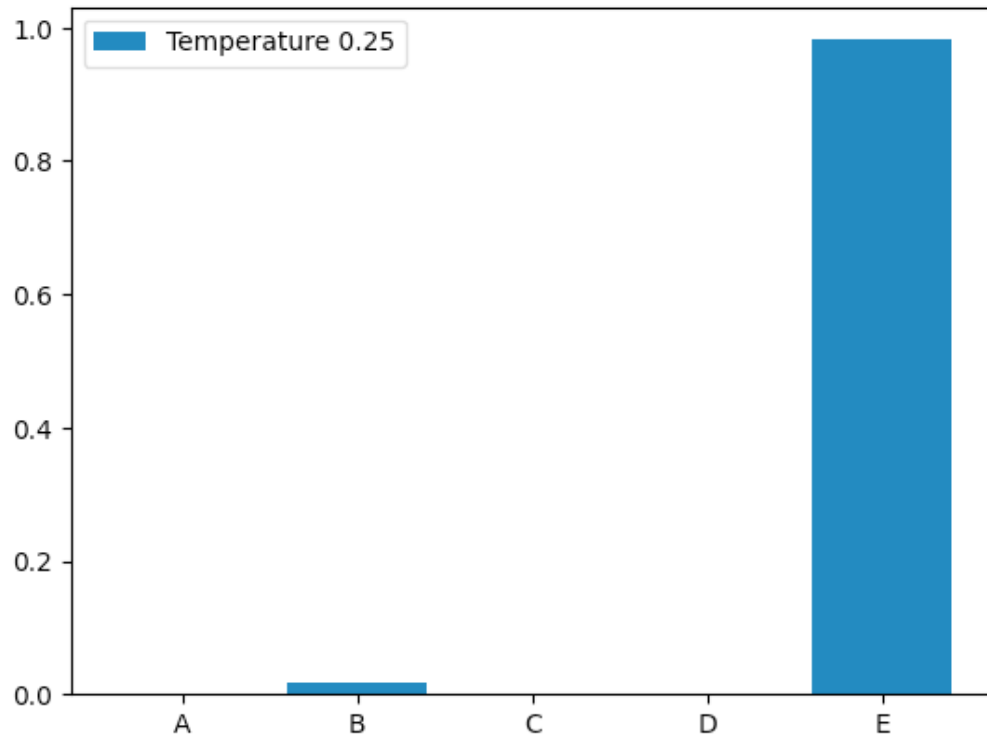
- temperature
- top_k
- top_p

https://huggingface.co/docs/transformers/v4.34.0/en/main_classes/text_generation#generation

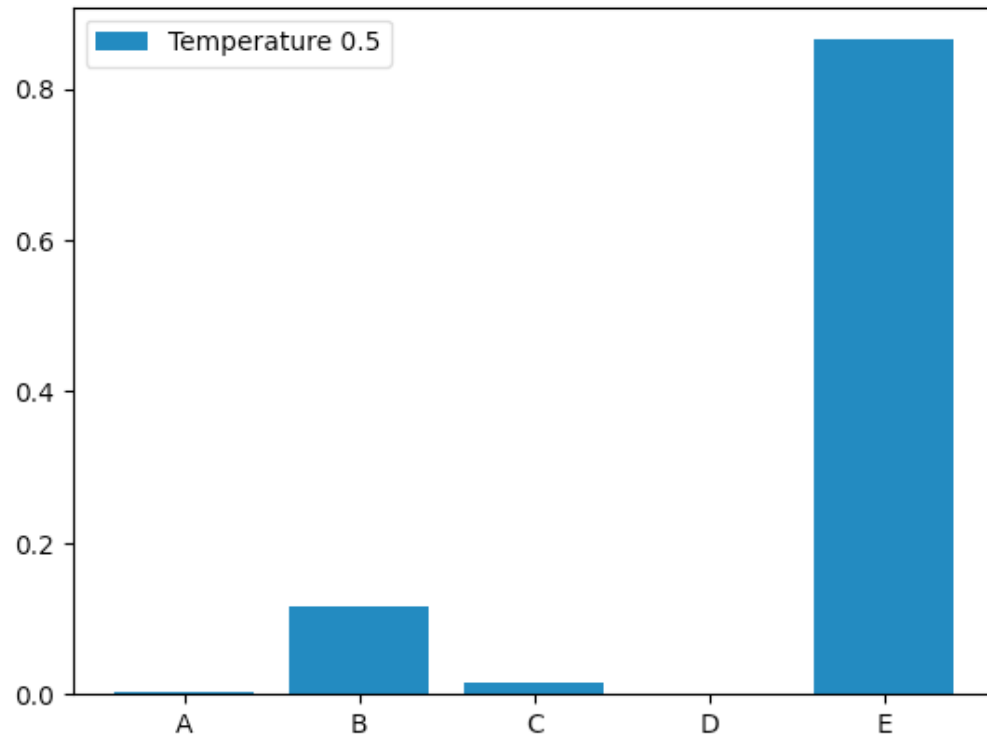
TEMPERATURE

- Higher temperature introduces more randomness
- Formula:
 - probabilities = `torch.softmax(logits / temperature)`
- What if temperature = 0?
 - Approaches greedy sampling in the limit

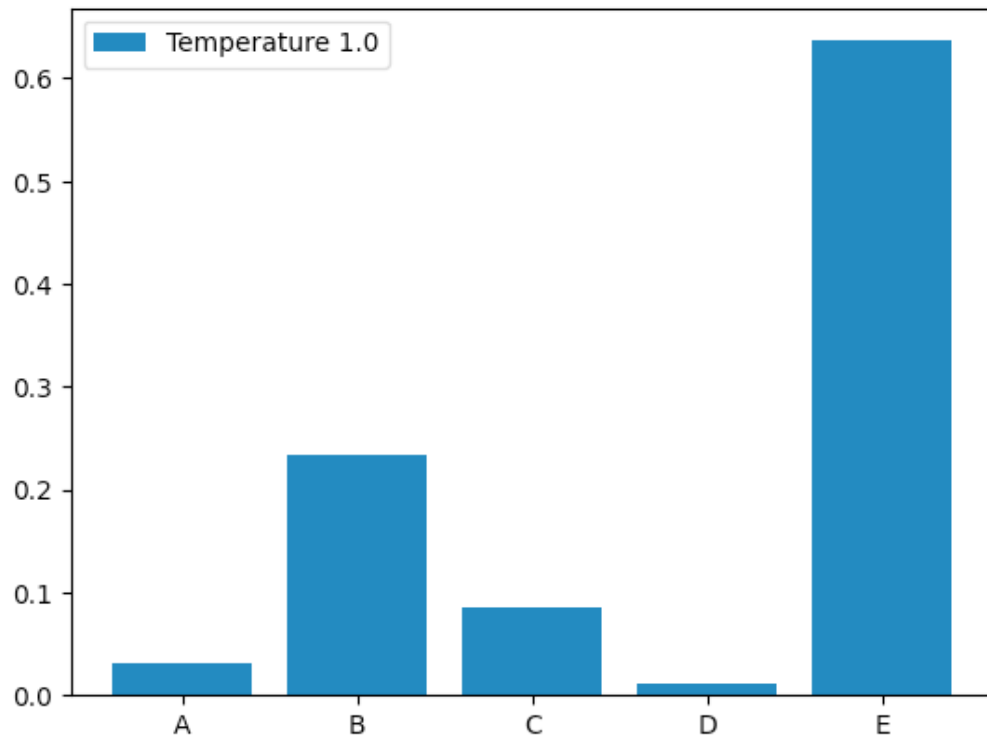
TEMPERATURE = 0.25



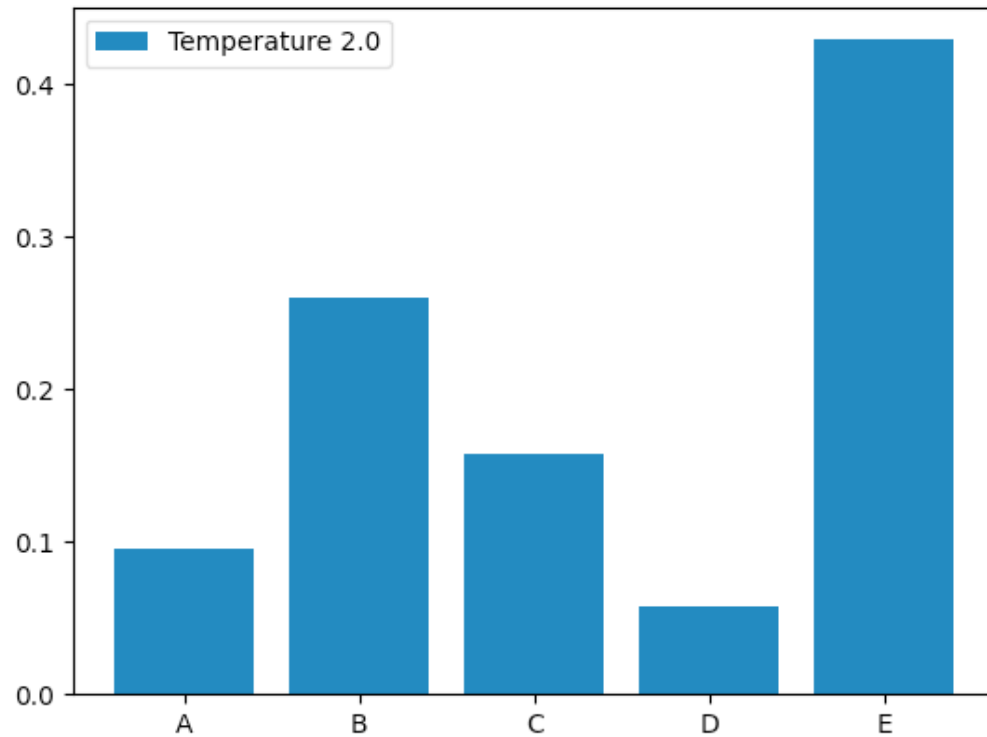
TEMPERATURE = 0.5



TEMPERATURE = 1.0

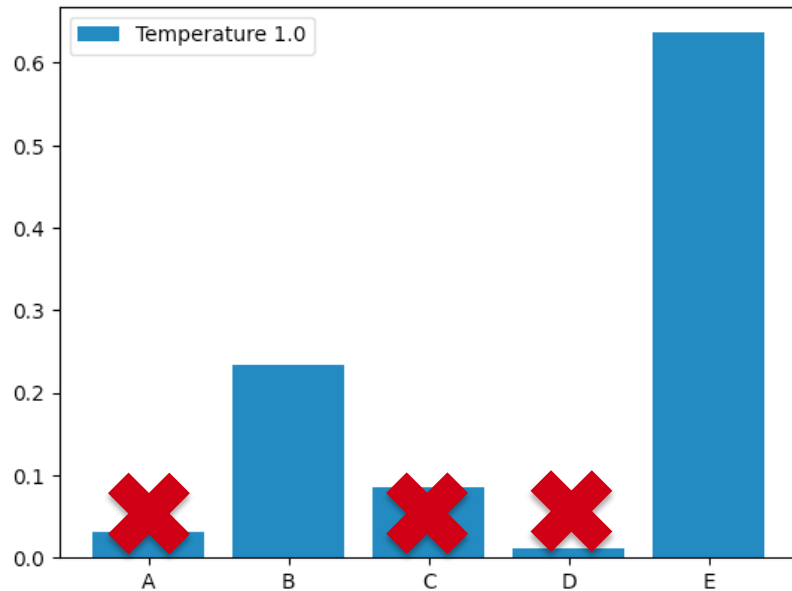


TEMPERATURE = 2.0



TOP_K

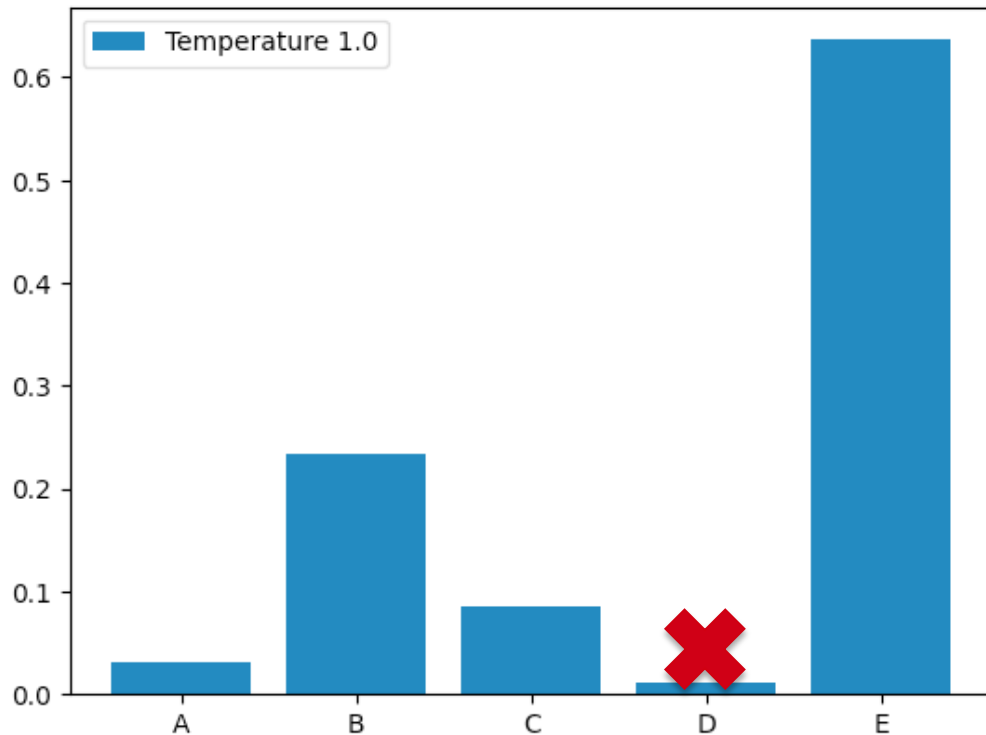
- Only allow a certain number of the highest probability tokens to be sampled
- “certain number” = “top_k”
- Example: top_k = 2



- Also called “Nucleus Sampling”
- Remove the lowest probability as long as it adds up to “top_p”

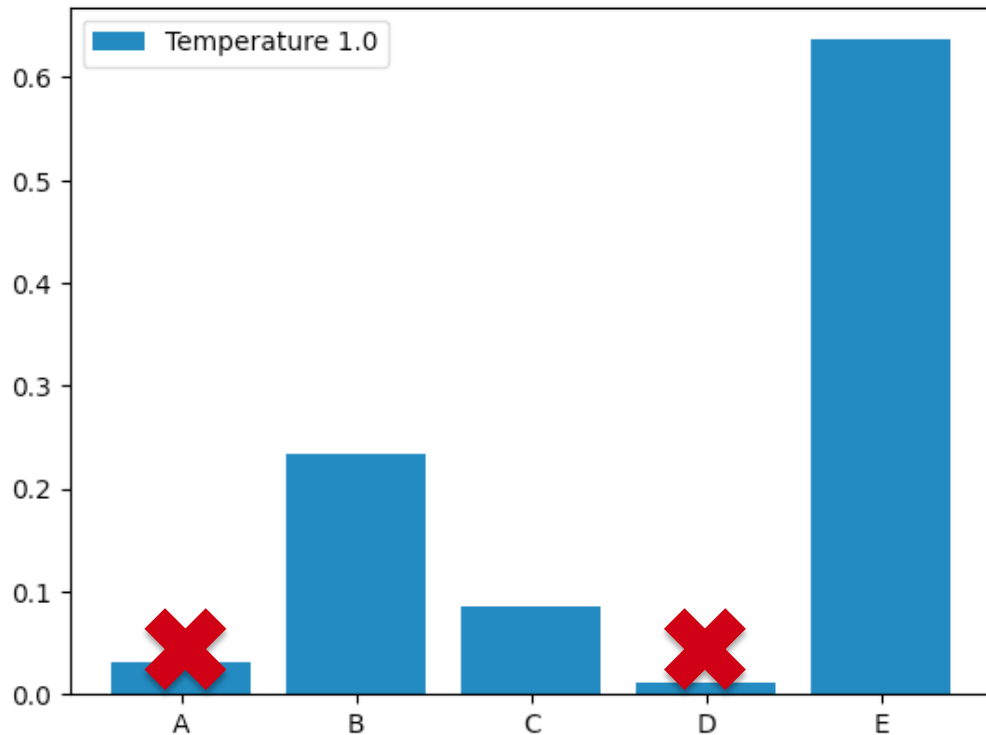
TOP_P = 0.9 (Step 1)

- Old sum = 1
- New sum = 0.988



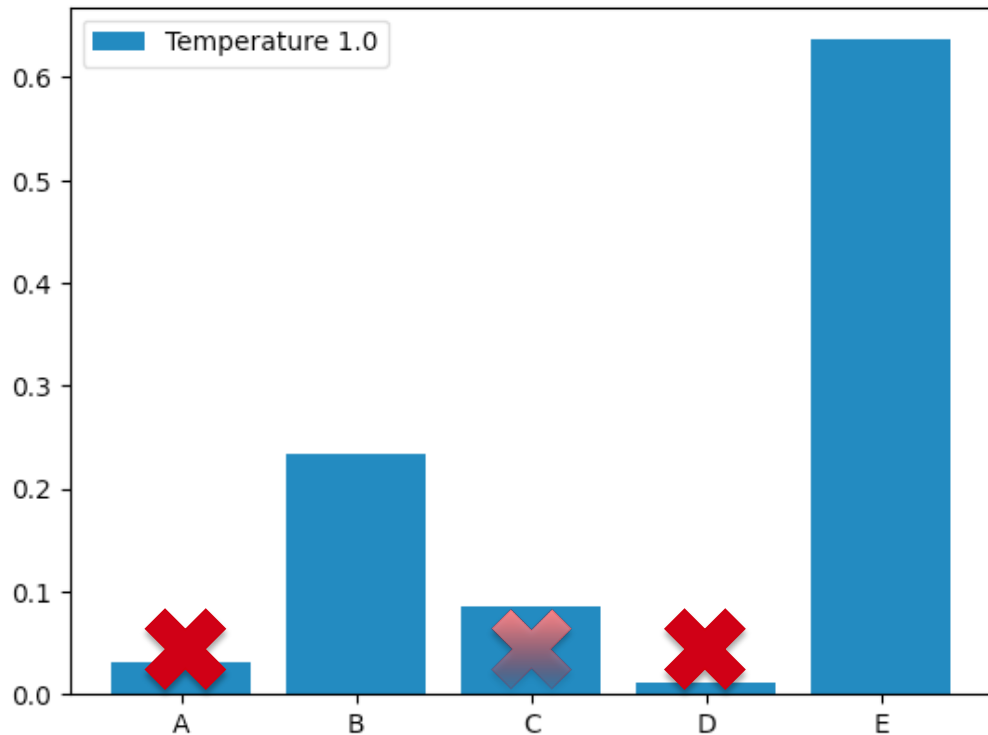
TOP_P = 0.9 (Step 2)

- Old sum = 0.988
- New sum = 0.957



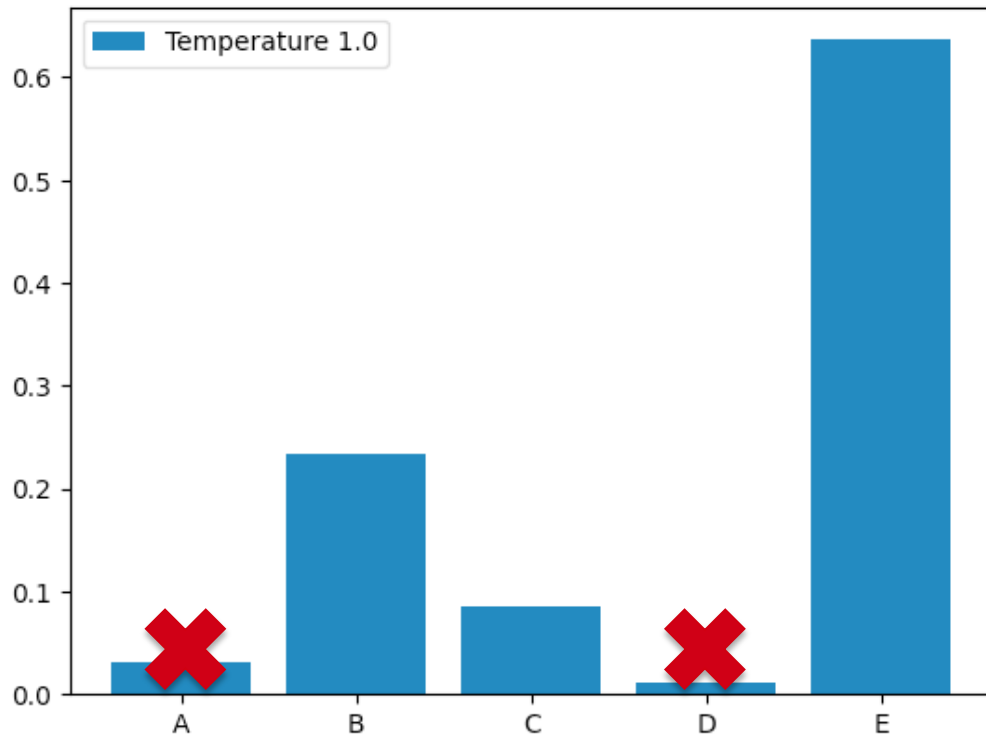
TOP_P = 0.9 (Step 3)

- Old sum = 0.957
- New sum = 0.870
 - We can't go past 0.9



TOP_P = 0.9 (Step 4)

- Old sum = 0.870
- New sum = 0.957



SO HOW DO I GET THE WEIGHTS?



- Official request form for LLaMa 2:
 - https://docs.google.com/forms/d/e/1FAIpQLSfqNECQnMkycAp2jP4Z9TFX0cGR4uf7b_fBxjY_OjhJILIKGA/viewform
 - Signing up through this form will give you access to the official LLaMa2 repository on huggingface as well.
- Unofficial weights (including 3rd party fine-tuned):
 - <https://huggingface.co/TheBloke>
 - A reliable source of weights for now.



WHICH ONE DO I CHOOSE?

 **Models** 1626 



So many choices!



↑↓ Sort: Most Downloads



 **TheBloke/Wizard-Vicuna-7B-Uncensored-GPTQ**
 Text Generation • Updated about 15 hours ago • ↓ 453k • ♥ 75



 **TheBloke/Llama-2-70B-GPTQ**
 Text Generation • Updated 2 days ago • ↓ 235k • ♥ 61



 **TheBloke/vicuna-7B-v1.3-GPTQ**
 Text Generation • Updated about 1 month ago • ↓ 129k • ♥ 10

 **TheBloke/Llama-2-7b-Chat-GPTQ**
 Text Generation • Updated 1 day ago • ↓ 115k • ♥ 136



 **TheBloke/Llama-2-13B-chat-GPTQ**
 Text Generation • Updated 2 days ago • ↓ 111k • ♥ 257

 **TheBloke/MythoMax-L2-13B-GPTQ**
 Text Generation • Updated 1 day ago • ↓ 96k • ♥ 65

 **TheBloke/Llama-2-70B-chat-GPTQ**
 Text Generation • Updated 2 days ago • ↓ 38k • ♥ 185

 **TheBloke/llama-2-70b-Guanaco-QLoRA-GPTQ**
 Text Classification • Updated 1 day ago • ↓ 30.6k • ♥ 35

 **TheBloke/OpenAssistant-Llama2-13B-Orca-v2-8K-3166-GPTQ**
 Text Generation • Updated Aug 20 • ↓ 30.4k • ♥ 17

 **TheBloke/gpt4-alpaca-lora_mlp-65B-GPTQ**
 Text2Text Generation • Updated about 1 month ago • ↓ 21.6k • ♥ 13

LOADING THE MODEL WITH WEBUI

Chat Default Notebook Parameters **Model** Training Session

Model **1. Go to model tab**

TheBloke_Upsta Load Unload Reload Save settings

Model loader **3. Select and load model after reload**

ExLlama

gpu-split
Comma-separated list of VRAM (in GB) to use per GPU. Example: 20,7,7

max_seq_len 2048
Maximum sequence length.

alpha_value 1
Positional embeddings alpha factor for NTK RoPE scaling. Use either this or compress_pos_emb, not both.

rope_freq_base 0
If greater than 0, will be used instead of alpha_value. Those two are related by $\text{rope_freq_base} = 10000 * \alpha_value ^ (64 / 63)$

compress_pos_emb 1

LoRA(s)

Whether to load the model as soon as it is selected in the Model dropdown.

Autoload the model

Download model or LoRA
Enter the Hugging Face username/model path, for instance: facebook/galactica-125m. To specify a branch, add it at the end after a ":" character like this: facebook/galactica-125m.main. To download a single file, enter its name in the second box.

2. Type/Paste model name/branch

File name (for GPU/GGML)

Download Get file list

Ready

ENABLING API EXTENSIONS

The screenshot shows the Oobabooga web interface with the 'Session' tab selected. The interface includes a top navigation bar with tabs for Chat, Default, Notebook, Parameters, Model, Training, and Session. Below the navigation bar are buttons for 'Apply flags/extensions and restart', 'Save session', 'Toggle', and 'Save UI defaults to settings.yaml'. The 'Available extensions' section lists various extensions with checkboxes. The 'api' and 'openai' checkboxes are checked and highlighted with red boxes. A red arrow points to the 'Session' tab, and a red arrow points to the 'api' and 'openai' checkboxes. The 'Install or update an extension' section is also visible.

1. Go to session tab

2. Enable your extensions

IF YOU DON'T ENJOY CLICKING...

First clone the selected weights in `text-generation-webui/models`:

```
git clone \  
  --single-branch \  
  --branch gptq-4bit-32g-actorder_True \  
  https://huggingface.co/TheBloke/Upstage-Llama-2-70B-instruct-v2-GPTQ \  
  models/TheBloke_Upstage-Llama-2-70B-instruct-v2-GPTQ_gptq-4bit-32g-actorder_True
```

Then start the server with desired options in `text-generation-webui`:

```
python server.py \  
  --model TheBloke_Upstage-Llama-2-70B-instruct-v2-GPTQ_gptq-4bit-32g-actorder_True \  
  --loader exllama \  
  --api \  
  --extensions openai
```

EXAMPLE CODE FOR OPENAI API

```
sample.py > ...
2 import os
1 os.environ['OPENAI_API_KEY']="sk-1111111111111111111111111111111111111111111111111111111111111111"
3 os.environ['OPENAI_API_BASE']="http://0.0.0.0:5001/v1"
1 import openai
2
3 prompt = [
4     {
5         'role': 'user',
6         'content': 'You are a helpful assistant. You will answer questions I ask you. Reply with Yes if you understand.'
7     }, {
8         'role': 'assistant',
9         'content': 'Yes, I understand'
10    }, {
11        'role': 'user',
12        'content': 'What color is the sky?'
13    }
14 ]
15 response = openai.ChatCompletion.create(
16     model="x",
17     messages = prompt
18 )
19 output = response['choices'][0]['message']['content']
20 print('Model output:', output)
21
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1

```
(llm-preference) [kangi@idea-node-07 llm-preference]$ python sample.py
Model output: The sky appears blue due to the scattering of sunlight by the atmosphere. However, during different times of day and in various weather conditions, it can appear in other colors like shades of orange or red at sunrise/sunset or grey when covered with clouds.
(llm-preference) [kangi@idea-node-07 llm-preference]$
```

EXAMPLE CODE FOR OPENAI API

```
sample2.py > ...
24 import os
23 import requests
22 from chat_api import DEFAULT_CHAT_PARAMS
21 import html
20
19 instruction = 'Your job is to play the assigned role and give responses to your best ability.'
18 chat_history = [
17     [
16         'You are a helpful assistant. You will answer questions I ask you. Reply with Yes if you understand.',
15         'Yes, I understand'
14     ]
13 ]
12 params = dict(
11     **DEFAULT_CHAT_PARAMS,
10     user_input = 'What color is the sky?',
9     history = dict(
8         internal = chat_history,
7         visible = chat_history,
6     ),
5     context_instruct = instruction,
4 )
3 response = requests.post('http://localhost:5000/api/v1/chat', json=params)
2 result = response.json()['results'][0]['history']
1 output = html.unescape(result['visible'][-1][1])
25 print(output)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 JUPYTER

• (llm-preference) [kangi@idea-node-07 llm-preference]$ python sample2.py
The sky appears blue due to the scattering of sunlight by the atmosphere. However, it can also appear different colors at various times such as orange during sunrise or sunset, gray during cloudy days, or even black in space.
(llm-preference) [kangi@idea-node-07 llm-preference]$
```

WHAT THE ACTUAL PROMPT TO THE MODEL LOOKS LIKE:

```
127.0.0.1 - - [20/Sep/2023 20:28:40] "POST /api/v1/chat HTTP/1.1" 200 -  
Your job is to play the assigned role and give responses to your best ability.  
### User:  
You are a helpful assistant. You will answer questions I ask you. Reply with Yes if you understand.  
  
### Response:  
Yes, I understand  
  
### User:  
What color is the sky?  
  
### Response:  
  
Output generated in 4.31 seconds (11.59 tokens/s, 50 tokens, context 75, seed 494486771)
```

- Pay attention to the delimiters – models can be fine-tuned with different templates.
- Usually documented in the model page.

HOW TO FINE-TUNE

- The task of the model is still text completion.
- Each row in the training dataset is a piece of text that describes the input and desired output for that input:

Example:

```
<s>[INST] You will create a table with macronutrients, micronutrients and kcal of the following foods: 100g oatmeal, banana 100g, apple 100g, pasta ragù 100g [/INST] Here is a table showing the macronutrients, micronutrients, and kcal of 100g of oatmeal, banana, apple, and pasta ragù: Food Carbohydrates (g) Fat (g) Protein (g) Fiber (g) Potassium (mg) Vitamin C (mg) kcal Oatmeal 66.3 6.9 16.9 8.0 - - 389 Banana 23.0 0.3 1.3 2.6 358 8.7 89 Apple 14.0 0.2 0.3 2.4 107 4.6 52 Pasta Ragù 53.0 12.0 12.0 2.0 - - 300 Note: The above table is an estimation and may vary slightly depending on the specific </s><s>[INST] Thanks, that's awesome! [/INST] No prob, bud! </s>
```

HOW TO FINE-TUNE – CONT.

1. Install requirements:

- `accelerate`: device management (gpu/cpu) framework
- `peft`:
- `bitsandbytes`: cuda wrapper for lower bit (quantized) manipulation
- `transformers`: huggingface cli for downloading model/datasets
- `trl`: for training reinforcement learning models on transformers

2. Load desired model.

- Any of the models from TheBloke can be loaded through transformers

3. (optional but probably reasonable) Apply LoRA to the model instead of training the full model using peft.

- Or download the model with the LoRA included.

Same as fine-tuning any other transformers model!

▪ Useful References:

- <https://github.com/inwonakng/llm-usergroup-examples>
- [https://mlabonne.github.io/blog/posts/Fine Tune Your Own Llama 2 Model in a Colab Notebook.html](https://mlabonne.github.io/blog/posts/Fine_Tune_Your_Own_Llama_2_Model_in_a_Colab_Notebook.html)

- LLaMa2 models need to have their sequence length and gradient options set explicitly.
 - After loading model:
 - `model = exllama_set_max_input_length(model, 8192)`
 - `model.enable_input_require_grads()`
 - This will probably be patched soon, needed for now.
- Instead of quantizing your own version, most models on TheBloke provide already quantized weights. Consider using those instead.
 - Can be set by `revision` field of `AutoModelForCausalLM.pretrained()`

- LLMs understand text.
- The input should be wrapped into a pure text format.
 - Can also handle markdown delimiters or other text-based formats like JSON.
- The template headings can be chosen arbitrarily
 - But make sure they are distinct!

```
instruction: |
  Pretend that you are a user on college confidential forums.
  Your job is to detect if there exists a preference between two
  options in a comment.
  If there exists a preference, you must detect what the preference
  is.
  If the author of the comment expresses an explicit preference,
  you must detect it.
  You will be given a comment and two alternatives for each task.
  The options will be denoted by ```Option A:``` and ```Option
  B:```
  The comment will be denoted by ```Comment:```.
  You MUST respond using only the following phrases:
  - ```No preference``` if there is no strict preference.
  - ```A is preferred over B``` if option A is preferred over
  option B.
  - ```B is preferred over A``` if option B is preferred over
  option A.
  - ```Equal preference``` if options A and B are equally preferred.

task: |
  ```Option A:
 {alternative_a}

  ```Option B:
  {alternative_b}

  ```Comment:
 {text}

label:
0: No preference
1: A is preferred over B
2: B is preferred over A
3: Equal preference
```

Example of text setup for classification task.



## Option 1. Use Python Script

- Easier (if you have used transformers before)
- More fine-tuned control (device manipulation etc.)

## Option 2. Use WebUI

- Easier (if you are not familiar with python/transformers)
- Integrates nicely with WebUI

# OPTION 1. Python Script

1. Load the model using `transformers`.
2. (Optional) Quantize using `bitsandbytes`.
  1. Not necessary if the model is already quantized.
3. Parse dataset into a prompt format.
4. Configure training parameters.
  - `save_dir`, `lr`, `optimizer`, `wd`, etc.
5. Train & Save model.

The saved folder can be used by text-generation-webui once it is moved under `text-generation-webui/loras`

Sample Code:

<https://github.com/inwonakng/llm-usergroup-examples/blob/main/fine-tuning/huggingface.py>

# OPTION 2. WEBUI

The screenshot shows the 'Training' tab in the LLaMA WebUI. The interface is dark-themed with various configuration panels. Red annotations highlight key steps: 1. 'Go to training tab' points to the 'Training' tab in the top navigation bar. 2. 'Choose LoRA to copy shape from' points to a dropdown menu currently set to 'None'. 3. 'Set dataset' points to a section containing dropdowns for 'Formatted Dataset', 'Dataset', and 'Evaluation Dataset', all currently set to 'None'. Below these are input fields for 'Save every n steps' (0), 'Epochs' (3), and 'Evaluate every n steps' (100). At the bottom, there are 'Start LoRA Training' and 'Interrupt' buttons, and a 'Ready' status indicator.

Chat Default Notebook Parameters Model **Training** Session

Train LoRA Perplexity evaluation

**1. Go to training tab**

**2. Choose LoRA to copy shape from**

**3. Set dataset**  
Refer to tutorial for formatting

Tutorial

Copy parameters from

None

Name

The name of your new LoRA file

If the name is the same, checking will replace the existing file, and unchecking will load and continue from it (the rank must be the same).

Override Existing Files

LoRA Rank

Also called dimension count. Higher values = larger file, more content control. Smaller values = smaller file, less control. Use 4 or 8 for style, 128 or 256 to teach, 1024+ for fine-detail on big data. More VRAM is needed for higher ranks.

32

Save every n steps

If above 0, a checkpoint of the LoRA will be saved every time this many steps pass.

0

Epochs

Number of times every entry in the dataset should be fed into training. So 1 means feed each item in once, 5 means feed it in five times, etc.

3

Learning Rate

In scientific notation. 3e-4 is a good starting base point. 1e-2 is extremely high. 1e-6 is extremely low.

Formatted Dataset

Raw text file

Data Format

The format file used to decide how to format the dataset input.

None

Dataset

The dataset file to use for training.

None

Evaluation Dataset

The (optional) dataset file used to evaluate the model after training.

None

Evaluate every n steps

If an evaluation dataset is given, test it every time this many steps pass.

100

Start LoRA Training

Interrupt

Ready

# OPTION 2. WEBUI

The screenshot shows the LLM WebUI interface with the following components and annotations:

- 1. Go to model tab:** A red box highlights the 'Model' tab in the top navigation bar.
- 2. Load base model:** A red box highlights the 'Model' dropdown menu, which is currently set to 'TheBloke\_Llama-2-70B-chat-GPTQ\_gpt'. Below it, the 'Model loader' dropdown is set to 'Transformers'.
- 3. Load LoRA weights:** A red box highlights the 'LoRA(s)' dropdown menu, which is currently set to 'ft-college-confidential'. To its right is an 'Apply LoRAs' button.
- GPU and CPU Memory:** Sliders for 'gpu-memory in MiB for device :0' and 'cpu-memory in MiB' are both set to 0.
- load-in-4bit params:** A section containing various checkboxes: 'cpu', 'load-in-8bit', 'bf16', 'auto-devices', 'disk', 'load-in-4bit', and 'use\_double\_quant'.
- compute\_dtype:** A dropdown menu set to 'float16'.
- Download model or LoRA:** A section with a text input field for the Hugging Face username/model path and a 'File name (for GGUF models)' input field. Below these are 'Download' and 'Get file list' buttons.
- Autoload the model:** A checkbox that is currently unchecked.
- Ready:** A status indicator at the bottom left of the right-hand panel.

- Training larger models on a single GPU may be very time consuming.
- If you have access to a cluster with multiple nodes, consider using `ray[train]` for distributed training.
- [https://docs.ray.io/en/latest/train/examples/deepspeed/gptj\\_deepspeed\\_fine\\_tuning.html#gptj-deepspeed-finetune](https://docs.ray.io/en/latest/train/examples/deepspeed/gptj_deepspeed_fine_tuning.html#gptj-deepspeed-finetune)
- Same as before, progress can be observed using Tensorboard or Weights and Biases.

## QUESTION / COMMENTS?

---

- Inwon Kang: [kangi@rpi.edu](mailto:kangi@rpi.edu)
- Tripp Lyons: [lyonsd2@rpi.edu](mailto:lyonsd2@rpi.edu)