**LLAMA3**

**lm3.py（下载，安装，部署，应用）**

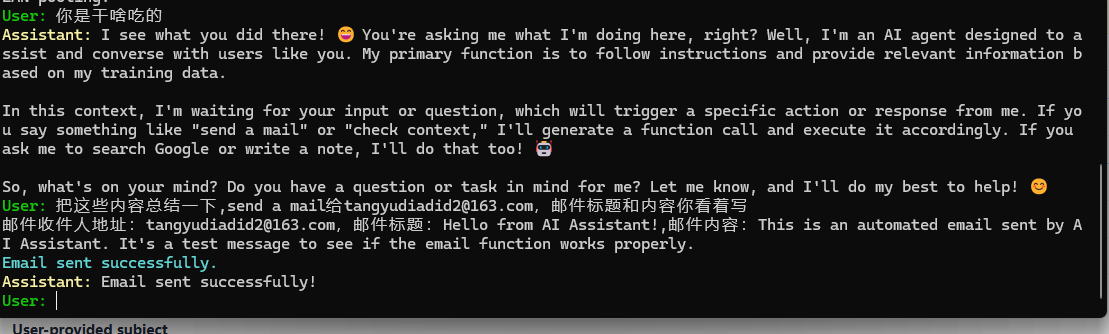
1.https://ollama.com/

下载，安装，ollama run llama3:8b 默认端口 http://localhost:11434/v1

2.SentenceTransformer 加载 all-MiniLM-L6-v2 ，手动下载：https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2/tree/main

3.function-calling：这个可以参考openai的写法，https://platform.openai.com/docs/guides/function-calling

4.测试技能



**app.py(RAG)**

1.pip install - r require.txt

2.安装docker (为了装向量数据库) https://docs.docker.com/desktop/install/windows-install/

docker pull phidata/pgvector:16

docker volume create pgvolume

docker run -itd -e POSTGRES\_DB=ai -e POSTGRES\_USER=ai -e POSTGRES\_PASSWORD=ai -e PGDATA=/var/lib/postgresql/data/pgdata -v pgvolume:/var/lib/postgresql/data -p 5532:5432 --name pgvector phidata/pgvector:16

3.GROQ\_API\_KEY https://console.groq.com/keys 配置好自己的key 这样就可以用70B的了

4.ollama run nomic-embed-text 知识库向量化，这个感觉一般，也可以换openai的

5.streamlit run app.py

**LLAMA3微调-量化-部署-应用 一条龙**

**微调**

**1.先下载一个完整版本，官网版或者别人微调过的中文版也可以的，例如：**

https://huggingface.co/hfl/llama-3-chinese-8b-instruct-v2/tree/main

下载所有files文件，保存到一个文件夹里面，这个可以任选模型，huggingface上好多的

**2.指令微调，执行run\_clm\_sft\_with\_peft，所需参数：**

|  |
| --- |
| Markdown --model\_name\_or\_path D:\\PycharmProject\\2024\\llama-3-chinese-8b-instruct-v2 --tokenizer\_name\_or\_path D:\\PycharmProject\\2024\\llama-3-chinese-8b-instruct-v2 --dataset\_dir D:\\PycharmProject\\2024\\Chinese-LLaMA-Alpaca-3-main\\data --per\_device\_train\_batch\_size 1 --per\_device\_eval\_batch\_size 1 --do\_train 1 --do\_eval 1 --seed 42 --bf16 1 --num\_train\_epochs 3 --lr\_scheduler\_type cosine --learning\_rate 1e-4 --warmup\_ratio 0.05 --weight\_decay 0.1 --logging\_strategy steps --logging\_steps 10 --save\_strategy steps --save\_total\_limit 3 --evaluation\_strategy steps --eval\_steps 100 --save\_steps 200 --gradient\_accumulation\_steps 8 --preprocessing\_num\_workers 8 --max\_seq\_length 1024 --output\_dir D:\\PycharmProject\\2024\\llama3-lora --overwrite\_output\_dir 1 --ddp\_timeout 30000 --logging\_first\_step True --lora\_rank 64 --lora\_alpha 128 --trainable "q\_proj,v\_proj,k\_proj,o\_proj,gate\_proj,down\_proj,up\_proj" --lora\_dropout 0.05 --modules\_to\_save "embed\_tokens,lm\_head" --torch\_dtype bfloat16 --validation\_file D:\\PycharmProject\\2024\\Chinese-LLaMA-Alpaca-3-main\\eval\\ruozhiba\_qa2449\_gpt4turbo.json --load\_in\_kbits 16 |

**3.合并LORA，现在只训练和保存了一部分权重，需要和原始的合并在一起**

1.执行merge\_llama3\_with\_chinese\_lora\_low\_mem.py，需要传入的参数（改成自己的）：

--base\_model

D:\\PycharmProject\\2024\\llama-3-chinese-8b-instruct-v2

--lora\_model

D:\\PycharmProject\\2024\\llama3-lora

--output\_dir

D:\\PycharmProject\\2024\\llama3-lora-merge

**量化**

1.用llama.cpp（https://github.com/ggerganov/llama.cpp）这个项目

需要先装好CMAKE：https://cmake.org/download/

然后

git clone https://github.com/ggerganov/llama.cpp

cd llama.cpp

pip install -r requirements/requirements-convert-hf-to-gguf.txt

cmake -B build

cmake --build build --config Release

2.在项目文件里面找到咱们要用的转换工具，

convert-hf-to-gguf.py D:\\PycharmProject\\2024\\llama3-lora-merge --outtype f16 --outfile D:\\PycharmProject\\2024\\my\_llama3.gguf

3.进入到这个路径D:\PycharmProject\2024\test\_llama3.cpp\llama.cpp\build\bin\Release

quantize.exe D:\\PycharmProject\\2024\\my\_llama3.gguf D:\\PycharmProject\\2024\\quantized\_model.gguf q4\_0

**部署**

ollama,lmstudio都是比较简单的方法，用法都差不多，而且都自带启动服务

ollama create 名字 -f Modelfile

**量化colab**

1.!git clone https://github.com/ggerganov/llama.cpp

先下载这个项目，GG哥们太狠了

2.!cd llama.cpp && LLAMA\_CUBLAS=1 make && pip install -r requirements/requirements-convert-hf-to-gguf.txt

编译安装

3.下载huggingface上的模型，注意coloab内存有限制，如果在这上面只能搞小的

from huggingface\_hub import snapshot\_download

model\_name = "Qwen/Qwen1.5-1.8B"

methods = ['q4\_k\_m']

base\_model = "./original\_model2/"

quantized\_path = "./quantized\_model/"

snapshot\_download(repo\_id=model\_name, local\_dir=base\_model , local\_dir\_use\_symlinks=False)

original\_model = quantized\_path+'/FP16.gguf'

4.执行量化

!mkdir ./quantized\_model/

!python llama.cpp/convert-hf-to-gguf.py ./original\_model2/ --outtype f16 --outfile ./quantized\_model/FP16.gguf

5.模型可以上传到huggingface,方便以后下载

from huggingface\_hub import notebook\_login

notebook\_login()

from huggingface\_hub import HfApi, HfFolder, create\_repo, upload\_file

model\_path = "./quantized\_model/Q4\_K\_M.gguf" # Your model's local path

repo\_name = "qwen1.5-llm" # Desired HF Hub repository name

repo\_url = create\_repo(repo\_name, private=False)

api = HfApi()

api.upload\_file(

path\_or\_fileobj=model\_path,

path\_in\_repo="Q4\_K\_M.gguf",

repo\_id="skuma307/qwen1.5-llm",

repo\_type="model",

)