1/8

# 大模型]GLM-4-9B-chat FastApi 部署调用





大模型 专栏收录该内容

10 订阅 105 篇

### 境准备

AutoDL 平台中租一个 3090 等 24G 显存的显卡机器,如下图所示镜像选择 PyTorch -> 2.1.0 -> 3.10(ubuntu22.04) -> 12.1。

434机 24	4GB		1 / 12		内存: 60GB AMD EPY	
框架名称		1.5.1		>	Python版本	Cuda版本
PyTorch	>	1.6.0		>	3.10(ubuntu22.04) >	12.1
TensorFlow	>	1.7.0		>		
Miniconda	>	1.8.1		>		
JAX	>	1.9.0		>		
PaddlePaddle	>	1.10.0		>		
TensorRT	>	1.11.0		>		
Gromacs	>	2.0.0		>		
Jittor	>	2.1.0		>		
		2.3.0		>		

下来打开刚刚租用服务器 的 JupyterLab ,并且打开其中的终端开始环境配置、模型下载和运行 demo 。

换源和安装依赖包。

创建完成后仍然可以更换其他镜像

```
1 # 升级pip
 2 | python -m pip install --upgrade pip
    # 更换 pypi 源加速库的安装
 4
   pip config set global.index-url https://pypi.tuna.tsinghua.edu.cn/simple
 5
    pip install fastapi==0.104.1
 6
    pip install uvicorn==0.24.0.post1
    pip install requests==2.25.1
    pip install modelscope==1.9.5
    pip install transformers==4.37.2
    pip install streamlit==1.24.0
12 | pip install sentencepiece==0.1.99
13 pip install accelerate==0.24.1
14 | pip install tiktoken==0.7.0
                                                 第 FL1623863129 关注
                                                                                                                       12 📭 🖠
```

考虑到部分同学配置环境可能会遇到一些问题,我们在 AutoDL 平台准备了 GLM-4 的环境镜像,该镜像适用于本教程需要 GLM-4 的部署环境。点击下方 AutoDL 示例即可。(vLLM 对 torch 版本要求较高,且越高的版本对模型的支持更全,效果更好,所以新建一个全新的镜像。)https://www.codewithgpu.com/i/datawhalechina/self-llm/GLM-4

### 型下载

用 modelscope 中的 snapshot\_download 函数下载模型,第一个参数为模型名称,参数 cache\_dir 为模型的下载路径。

/root/autodl-tmp 路径下新建 download.py 文件并在其中输入以下内容,粘贴代码后记得保存文件,如下图所示。并运行 python /root/autodl-tmp/dow ,模型大小为 18 GB,下载模型大概需要 10~20 分钟。

```
1 import torch
2 from modelscope import snapshot_download, AutoModel, AutoTokenizer
3 import os
4 model_dir = snapshot_download('ZhipuAI/glm-4-9b-chat', cache_dir='/root/autodl-tmp', revision='master')
```

端出现下图结果表示下载成功。

## 码准备

/root/autodl-tmp 路径下新建 api.py 文件并在其中输入以下内容,粘贴代码后记得保存文件。下面的代码有很详细的注释,大家如有不理解的地方,欢

```
1
   from fastapi import FastAPI, Request
   from transformers import AutoTokenizer, AutoModelForCausalLM
   import uvicorn
   import json
5
   import datetime
   import torch
6
7
   # 设置设备参数
8
   DEVICE = "cuda" # 使用CUDA
9
   DEVICE ID = "0" # CUDA设备ID, 如果未设置则为空
10
   CUDA DEVICE = f"{DEVICE}:{DEVICE ID}" if DEVICE ID else DEVICE # 组合CUDA设备信息
11
12
   # 清理GPU内存函数
13
14
   def torch_gc():
       if torch.cuda.is_available(): # 检查是否可用CUDA
15
           with torch.cuda.device(CUDA_DEVICE): # 指定CUDA设备
16
17
              torch.cuda.empty_cache() # 清空CUDA缓存
18
              torch.cuda.ipc_collect() # 收集CUDA内存碎片
19
20
   # 创建FastAPI应用
21
   app = FastAPI()
22
   # 处理POST请求的端点
23
   @app.post("/")
24
25
   async def create_item(request: Request):
26
       global model, tokenizer # 声明全局变量以便在函数内部使用模型和分词器
27
       json_post_raw = await request.json() # 获取POST请求的JSON数据
       json_post = json.dumps(json_post_raw) # 将JSON数据转换为字符串
28
29
       json_post_list = json.loads(json_post) # 将字符串转换为Python对象
30
       prompt = json_post_list.get('prompt') # 获取请求中的提示
31
       history = json_post_list.get('history')
                                                   FL1623863129 ( 关注 )
                                                                                                                  12
32
       max_length = json_post_list.get('max_leng
33
```

```
top_p = json_post_list.get('top_p') # 获取请求中的top_p参数
34
       temperature = json_post_list.get('temperature') # 获取请求中的温度参数
35
       # 调用模型进行对话生成
36
       response, history = model.chat(
37
           tokenizer,
38
           prompt,
39
           history=history,
40
           max_length=max_length if max_length else 2048, # 如果未提供最大长度,默认使用2048
41
           top_p=top_p if top_p else 0.7, # 如果未提供top_p参数,默认使用0.7
42
           temperature=temperature if temperature else 0.95 # 如果未提供温度参数,默认使用0.95
43
       )
44
       now = datetime.datetime.now() # 获取当前时间
45
       time = now.strftime("%Y-%m-%d %H:%M:%S") # 格式化时间为字符串
46
       # 构建响应JSON
47
       answer = {
48
           "response": response,
49
           "history": history,
50
           "status": 200,
51
           "time": time
52
53
       # 构建日志信息
54
       log = "[" + time + "] " + '", prompt:"' + prompt + '", response:"' + repr(response) + '"'
55
       print(log) # 打印日志
56
       torch_gc() # 执行GPU内存清理
57
       return answer # 返回响应
58
59
    # 主函数入口
60
    if __name__ == '__main__':
61
        # 加载预训练的分词器和模型
62
       tokenizer = AutoTokenizer.from_pretrained("/root/autodl-tmp/ZhipuAI/glm-4-9b-chat", trust_remote_code=True)
63
       model = AutoModelForCausalLM.from_pretrained(
64
           "/root/autodl-tmp/ZhipuAI/glm-4-9b-chat",
65
           torch_dtype=torch.bfloat16,
           trust_remote_code=True,
67
           device_map="auto",
68
69
       model.eval() # 设置模型为评估模式
70
       # 启动FastAPI应用
71
       # 用6006端口可以将autodl的端口映射到本地,从而在本地使用api
72
       uvicorn.run(app, host='0.0.0.0', port=6006, workers=1) # 在指定端口和主机上启动应用
```

#### si 部署

终端输入以下命令启动 api 服务。

```
1 cd /root/autodl-tmp
2 python api.py
```

端出现以下结果表示启用 api 服务成功。

以部署在 6006 端口,通过 POST 方法进行调用,可以重新开启一个终端使用 curl 调用,如下所示:

```
1 curl -X POST "http://127.0.0.1:6006" \
2 -H 'Content-Type: application/json' \
3 -d '{"prompt": "你好", "history": []}'
```

到的返回值如下所示:

**刊示例结果如下图所示**:

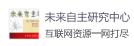
```
428-65e58bb8: /autodl-tmp# curl -X POST "http://127.0.0.1:6006" \
                  "你好", "history": []}'
我是人工智能助手,很高兴见到你,有什么可以帮助你的吗?","history":[{"role":"user","content":"你好"},{"role":"assistant",
stus":200."time":"2024-06-05 14:01:23curl -X POST "http://127.0.0.1:6006" \58bb8:"/autodl-tmp# <mark>curl -X POST "http://127.</mark>0
                          自哪里?","history"。[]}"
智能助手,没有物理意义上的"集自"集个地方。我是由清华大学 KBC 实验室和智谱 AI 公司共同开发的。我的"存在"是基于大量的数据和算法,运行在服务器上。","history"。[["role":
"metadata":"","content":"我是一个人工智能助手,没有物理意义上的"集自"某个地方。我是由清华大学 KBC 实验室和智谱 AI 公司共同开发的。我的"存在"是基于大量的数据和算法
15 14:01:58*}root@autodl-contaicurl -X POSI "http://127.0.0.1:6006" \
                                                      \n10. **0penAI**: 虽然0penAI是一家非营利组
各有其优势和特点。"}],"status":200,"time"
```

可以使用 python 中的 requests 库进行调用,如下所示:

```
import requests
1
 2
    import json
 3
    def get_completion(prompt):
 4
 5
       headers = {'Content-Type': 'application/json'}
 6
        data = {"prompt": prompt, "history": []}
 7
       response = requests.post(url='http://127.0.0.1:6006', headers=headers, data=json.dumps(data))
 8
       return response.json()['response']
 9
10
   if __name__ == '__main__':
      print(get_completion('你好,讲个幽默小故事'))
11
```

#### 用结果如下图所示:

```
autodl-container-48b1458428-65e58bb8: "/autodl-tmp# python api-request.py
     - 只猫和一只狗在公园里相遇了。猫看到狗正拿着一根树枝,于是好奇地问:"哎呀,狗哥,你这是要干嘛呢?"
狗得意洋洋地说:"峨,这可是我新学的魔法,能变出好吃的肉骨头!"
猫眼睛一亮,兴奋地问: '真的吗?那快教教我,我也想试试!"
狗点了点头,开始传授秘诀:"首先,你得把树枝放在地上,然后围着它跳三圈,再念一句咒语: 骨骨骨,肉肉肉,给我骨头吃个够!""
猫听完,立刻按照狗的指示,围着树枝跳了三圈,然后大声念道:"骨骨骨,肉肉肉,给我骨头吃个够!"
突然,树枝上真的掉下一根肉骨头来。猫高兴地捡起来,准备享用。
这时,一只乌龟慢悠悠地从旁边走过,看到这一幕,好奇地问:"猫哥,你在干嘛呢?"
猫得意洋洋地回答: '我学会了狗哥的魔法,现在能变出肉骨头了!"
乌龟听了,眨了眨眼睛,说: '蛾?那我也来试试,说不定我还能变出个金箍棒呢!"
说完,乌龟围着树枝跳了三圈,然后念道:"金金金,箍箍箍,给我金箍棒一根!"
奇迹发生了,树枝上竟然掉下一根金箍棒来。乌龟拿着金箍棒,兴奋地挥舞起来,嘴里还念叨:"这可是我的神器,谁敢惹我,我就用它打他!
猫和狗看着乌龟,惊讶得说不出话来。这时,一只小鸟飞过,笑着说:"哎呀,你们这三个,都是'幻<u>想家'啊!</u>GSDN @FL1623863129
```



微信公众号>

#### 大模型-使用 FastChat 部署ChatGLM3

田介绍如何使用 FastChat 本地部署ChatGLM3模型 (支持仅CPU环境)

#### M-4本地部署的实战教程 热门推荐

文主要介绍了<mark>GLM-4-9B</mark>本地<mark>部署</mark>的实战教程,希望对学习和使用大<mark>模型</mark>的同学们有所帮助。 文章目录 1. 前言 2. 配置环境 2.1 安装虚拟环境 2.2 安装依赖库 2.3 下载<mark>模型</mark>文件 3

系评论



半夜删你代码Orz 热评请问GLM4的stream输出,您试过吗。和GLM2的stream方法好像不一样...

#### Ib-chat及使用NextChat调用其API\_glm-4-9b-chat 硬件配置

clone https://www.modelscope.cn/ZhipuAl/glm-4-9b-chat.git 1 2 我的模型放在:/home/GLM-4-main/glm-4-9b-chat vscode切换目录为代码目录,bitsandbytes照着我这样改低版本,

at-GLM 详细部署(GPU显存>=12GB) chatglm at-GLM 详细部署(GPU显存>=12GB) 建议配置: (Windows OS



FL1623863129 关注





#### 等了! 速来体验 GLM-4-9B-Chat

Оре

大态变为「运行中」后,将鼠标移动至「API 地址」后,复制该地址并在新标签页打开,即可跳转至 GLM-4-9B-Chat Demo 页面。官方给出的数据显示,对比训练量更多的 Llar

#### GLM-4部署实战】GLM-4-9B-Chat模型本地部署实践指南

寻道码路,探索编程之路

人工智能的浪潮中,深度学习<mark>模型</mark>的<mark>部署</mark>已成为技术研究和实践的热点。自然语言处理(NLP)领域,尤其是对话系统,正迅速成为智能应用的核心。GLM-4-9B-Chat模型以其。

#### 地容器化快速部署GLM-4-9B-chat教程

YELLO'

1-4-9b-chat 容器化快速部署

#### 新开源对话大模型glm-4-9b-chat本地部署使用

M-4-9B 是智谱 AI 推出的最新一代预训练<mark>模型 GLM-</mark>4 系列中的开源版本。在语义、数学、推理、代码和知识等多方面的数据集测评中,GLM-4-9B 及其人类偏好对齐的版本 GL

#### stAPI部署GLM-4-9B-Chat遇到的坑

2401\_83

主是按照开源项目self-llm中的,主要遇到了两个问题,记录下~

#### 则langchain-ChatGLM API调用客户端(及未解决的问题)

Raine

gchain-ChatGLM是一个基于本地知识库的LLM对话库。其基于text2vec-large-Chinese为Embedding模型,ChatGLM-6B为对话大模型。原项目地址:https://github.com/chatcha

#### atGLM-6B部署、实战与微调

m0 478

atGLM-6B 是一个开源的、支持中英双语问答的对话语言模型,基于 General Language Model (GLM) 架构,具有 62 亿参数。结合模型量化技术,用户可以在消费级的显卡上达

### 零开始学ChatGLM3-6b大模型在本地平台的部署推理

weixin 446

atGLM3-6B 的基础<mark>模型 ChatGLM3</mark>-6B-Base 采用了更多样的训练数据、更充分的训练步数和更合理的训练策略。在语义、数学、推理、代码、知识等不同角度的数据集上测评!

#### 扩安全大模型: 微调internIm2模型实现针对煤矿事故和煤矿安全知识的智能问答

广安全大模型: 微调internIm2模型实现针对煤矿事故和煤矿安全知识的智能问答

#### GLM-4部署实战】GLM-4-9B-Chat模型之对话机器人部署测试

寻道码路,探索编程之路

人工智能的浪潮中,对话机器人作为人机交互的重要形式,正逐渐渗透到我们生活的方方面面。GLM-4-9B-Chat模型,以其强大的语言处理能力,为构建智能对话系统提供了坚多

### 模型]GLM4-9B-chat Lora 微调

FL16238

aConfig这个类中可以设置很多参数,但主要的参数没多少,简单讲一讲,感兴趣的同学可以直接看源码。task\_type:模型类型:需要训练的模型层的名字,主要就是attention是

#### 速调用 GLM-4-9B-Chat 语言模型

2,选择哪种后端进行推理取决于您的具体需求,包括任务类型、性能要求、资源限制等因素。性能: transformers后端使用硬件加速(如GPU、TPU等)进行推理,因此在处理

### .M4-9B-chat模型微调 单记录了对于新开源的GLM4-9B-Chat模型进行LoRA微调的全部过程。

hap

#### 模型]GLM-4-9B-Chat vLLM 部署调用

FL16238

息到部分同学配置环境可能会遇到一些问题,我们在 AutoDL 平台准备了 GLM-4 的环境镜像,该镜像适用于本教程需要 GLM-4 的部署环境。(vLLM 对 torch 版本要求较高,且

### .M之GLM-4: GLM-4-9B的简介、安装和使用方法、案例应用之详细攻略

近期请国内外头部出版社可尽快私信博主

M之GLM-4-9B: GLM-4-9B的简介、安装和使用方法、案例应用之详细攻略 目录 GLM-4的简介 GLM-4-9B的安装和使用 GLM-4-9B的案例应用 GLM-4的简介 GLM-4-9B

#### 机器学习】GLM4-9B-Chat大模型/GLM-4V-9B多模态大模型概述、原理及推理实战

人工

文首先对GLM4-9B的模型特点及原理进行介绍,接着分别对GLM4-9B-Chat语言大模型和GLM-4V-9B多模态大模型进行代码实践。排了很多坑,推荐阅读和收藏。

#### n-4-9b-chat微调 最新发布

1-4-9b-chat微调是一种基于语言<mark>模型</mark>的微调技术。在人工智能领域,语言<mark>模型</mark>是用来理解和生成自然语言的重要工具,它能够根据大量语料库训练得到对语言的泛化理解。所谓(

关于我们 招贤纳士 商务合作 寻求报道 ☎ 400-660-0108 ☎ kefu@csdn.net ⇨ 在线客服 工作时间 8:30-22:00 公安备案号11010502030143 京ICP备19004658号 京网文 [2020] 1039-165号 经营性网站备案信息 北京互联网违法和不良信息举报中心 家长监护 网络110报警服务 中国互联网举报中心 Chrome商店下载 账号管理规范 版权与免责声明 版权申诉 出版物许可证 营业执照 ◎1999-2024北京创新乐知网络技术有限公司









私信

关注



!博主文章

Q

#### 文章

0-11全版本下载地址MSDN纯净版ISO-20217更新 ① 124245

常见60种野生中草药 ① 70434

manager打不开闪退问题完美解决2017 新方法 ① 65884

商品类目查询方法怎样查看别人商品的 淘宝类目查询工具软件 ① 40625

·][原创]VSCode C++怎么让运行的时候 cmd窗口,而不是在VSCode调试输出 30896

#### 专栏

). 环境	竟配置	254篇
数据	居集	445篇
深度	<b></b> 達学习	216篇
软件	牛工具	43篇
Pytl	hon	201篇
Pyte	orch	15篇

# 评论

集][目标检测]猪数据集VOC-2856张 云端: 优质好文, 博主的文章细节很到 文章思路清晰,图文并茂,排版整: ... yolov8的辣椒缺陷检测系统python源... 奇才李先生: 基于yolov8的辣椒缺陷检 统python源码 onnx模型 评估指标曲 ... 引集][目标检测]海上红外目标检测检测... 云端: 优秀, 干货就是干货, 字字精辟 ,已收藏,博主的文章总是如一盏。... 〕集][目标检测]Udacity交通目标检测... 云端: 阅读这篇博文真是一种享受! 作 文字流畅自然,吸引了我的目光。 3 ... yolov5的中国交通标志TT100K检测... 黯然.: <a><a><a><a></a></a></a>The explanation of 基于yolo 中国交通标志TT100K检测系统pyth ...

## 在看

空间安全基础(三) ① 708 ensp的telnet登录防火墙具体配置 VA开源】基于Vue和SpringBoot古典舞 交流平台



#### 2024/9/29 12:10

code刷题]面试经典150题之9python哈 详解 (知识点+题合集) 💿 673 者怎样在游戏业生存?

### 文章

tensorflow报错InternalError: libdevice ound at ./libdevice.10.bc解决方法

集][目标检测]猪数据集VOC-2856张

[学习]基于YOLO高质量项目源码+模型 II界面汇总

4			
9月	08月	07月	
5篇	114篇	55篇	
i月	05月	04月	ı
4篇	124篇	136篇	
3月	02月	01月	
3篇	51篇	98篇	
3年 46	4篇	2022年 180篇	
1年 180年		2020年 133年	•

### 准备

下载

准备

邹署