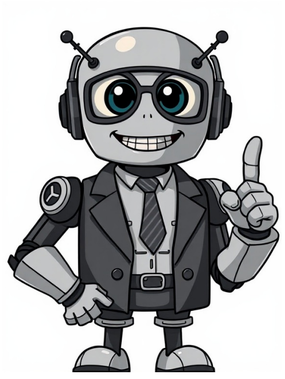
Click to verify



selltomorrow 2018-08-07 11:31:33 定位出错代码为vector vec_com_temp(vecLHSEat.size());该代码会反复调用，调用第六次就出错最，很奇怪的问题，哪位有经验可以指导一下，分全送了 ...全文 gray_wkl 2019-02-18 12:28:19 看到一个面试题，问字符串常量“A”的长度？ 我在网上看到的答案是2，也就是把\0也算作长度了，但是我在一本书上看到说“hello girl”的长度是10（空格也是一个字符），这里又没有把\0算作长度..... 那么究竟该不该算呢 只是看到了，想纠结一下，可能没有什么实际意义 ...全文 acwinder 2010-11-17 10:10:58 如题，CString的大小是不是能通过sizeof来获取，而长度是不是能通过getlength来获取？ ...全文 cjy0000 2015-09-24 11:55:21 因为是大文件，肯定要循环接收，那么recv的buffer设置多少合适呢？这个有没有个经验值或者标准什么的？谢谢！ ...全文 鹿鹿´_>` 2019-12-09 10:58:43 def cross_entropy_error(y, t): if y.ndim == 1: t = t.reshape(1, t.size) y = y.reshape(1, y.size) if t.size == y.size: t = t.argmax(axis=1) batch_size = y.shape[0] return -np.sum(np.log(y[np.arange(batch_size), t] + 1e-7)) / batch_size log(y[1，2]).....)这是怎么弄的，也就是y[1，2]咋算的？？ 最近本人来开始摸索微调训练，因为需要部署在ollama上面，所以需要转换成.gguf格式，但是部署上了ollama后，发现结果和预期差异很大，所以一步一步往回追溯，最后发现使用convert_hf_to_gguf.py转换后的.gguf模型，使用llama-cli运行时的输出，和预期的差异很大，使用训练数据来测试，结果完全不对。另外部署上ollama后相同的输入数据，答案和上一步直接运行.gguf差异也很大 首先是转成.gguf前最后一步，验证adapter合并后模型模型的代码： import torch from transformers import (AutoModelForCausalLM, AutoTokenizer, pipeline, BitsAndBytesConfig) from peft import PeftModel, PeftConfig from datasets import load_dataset import json import time # 1. 设置路径 merge_model_path = "/home/ps/pyCode/merged_model" # 原始基础模型 test_data_path = "/home/ps/文档/trainingData" # 测试数据集 # 2. 加载基础模型和分词器 tokenizer = AutoTokenizer.from_pretrained(merge_model_path, trust_remote_code=True, padding_side="right") tokenizer.pad_token = tokenizer.eos_token # 确保设置填充token # 配置量化（可选，减少显存占用） bnb_config = BitsAndBytesConfig(load_in_4bit=True, bnb_4bit_quant_type="nf4", bnb_4bit_compute_dtype=torch.bfloat16) merge_model = AutoModelForCausalLM.from_pretrained(merge_model_path, quantization_config=bnb_config, device_map="auto", trust_remote_code=True, torch_dtype=torch.bfloat16) # 4. 创建文本生成管道 text_generator = pipeline("text-generation", model=merge_model, #device=model.device, max_new_tokens=256, # 最大生成长度 do_sample=True, # 启用随机采样 temperature=0.7, # 控制随机性 (0-1) top_p=0.9, # 核采样参数 tokenizer=tokenizer, eos_token_id=tokenizer.eos_token_id) # 加载测试数据集 #test_data = load_dataset("json", data_files=test_data_path+ "/*.json", split="train") outputs = text_generator([{"role":"system","content":"根据输入的症状描述或诊断症状名，告诉我对应的症状名，对应的贴敷方案"},{"role":"user","content":f"开放性脏器，经常反复感染，是妇科的常见病，多发病。临床上以小腹不适，活动后加重，腰酸，腿酸，白带过多，外阴瘙痒。盆腔炎性疾病包括女性生殖道器官及其周围组织（子宫、附件、盆腔腹膜等组织）。"}], return_full_text=False, # 不返回输入文本 num_return_sequences=1,) # 提取生成的文本 response = outputs[0]['generated_text'].strip() print(f"test question answer:{response}"); outputs = text_generator("根据输入的症状描述或诊断症状名，告诉我对应的症状名，对应的贴敷方案黏黄鼻涕，舌质红苔黄，眼屎多，四肢暖，头痛，汗出热不解，咽红，大椎湿。发热的3-5天。", return_full_text=False, # 不返回输入文本 num_return_sequences=1,) # 提取生成的文本 response = outputs[0]['generated_text'].strip() print(f"no 2 test question answer:{response}"); 第二次的是按照切词器格式直接输入文本来进行测试，这两次都达到了预期，和标准答案完全一致，我这个模型的system固定为："根据输入的症状描述或诊断症状名，告诉我对应的症状名，对应的贴敷方案" 然后使用llama.cpp进行了模型转换，指令："python3 convert_hf_to_gguf.py /home/ps/pyCode/merged_model --outtype f16 " 转换成功 最后运行llama-cli指令"/home/ps/gitProject/llama.cpp/build/bin/llama-cli -m Merged_Model-8.2B-F16.gguf --file prompt.txt" prompt.txt内容如下，就是验证合并模型的第二次输入 运行结果： 明显不对，这个问题可能是什么原因导致的？ 转换的详细log，运行llama-cli的详细log我会放在下面两楼 frankspy 2005-11-29 11:26:51 打印一张票证，第一张打印的票证的内容，第二张打印出来"PCL XL error Warning： IllegalMediaSize"的字样，不知改如何处理，请高手指教！！！ ...全文 完美芯片 2008-10-13 09:53:45 程序里不用包括头文件就可以直接使用size_t，到底是为什么？ ...全文 yaochunyu236 2012-10-23 05:46:31 vs编译器当数组申请过大的时候会报错：错误 25 error C2148: 数组的总大小不得超过 0x7fffffff 字节。下面问题来了，如果真的需要比这大的空间的时候应该怎么去解决这个问题？怎么解决内存的连续性？请各位大神指教。ps:不要问申请这么大空间干啥，请不要纠结这个问题。 ...全文 最近程序运行，会比较频繁的当机，查看dump发现当机点总是内存分配函数(malloc_consolidate)。 症状： 函数在malloc_consolidate中不能返回，程序收到 Segmentation fault(11)而当机。 其中一次栈如下： #0 0x007d5401 in malloc_consolidate () from /lib/libc.so.6 #1 0x007d73bd in _int_malloc () from /lib/libc.so.6 #2 0x007d93ab in malloc () from /lib/libc.so.6 #3 0x00c50aa7 in operator new () from /usr/lib/libstdc++.so.6 #4 0x00c50bdd in operator new[] () from /usr/lib/libstdc++.so.6 #5 0x085af1ef in function5 (this=0xbfdc3864, byteBufLen=1024, bitBufLen=256) at *.cpp:95 #6 0x085af962 in function6 (this=0xbfdc3864) at *.cpp:24 #7 0x0814931d in function7 (this=0x94cba48, user=0x2492f220, npcOid=2483, bAuto=true) at *p.cpp:1787 #8 0x0814af06 in function8 (this=0x94cba48, cre=0x1c6f8550, user=0x2492f220) at *.cpp:1827 #9 0x085b2dee in function9 (this=0x92e8c6c, ev=0x8775100, data=0xbfdc3a20) ... 栈底函数每次相同，但是栈内容从不重复。 网上搜索了一把，众说纷纭，莫衷一是，还请高手现身指点，谢谢。 主要想知道几个要点： 1 是栈越界还是堆越界，为什么这种越界会引起这种问题呢，小弟实在不能理解。 2 怎样从dump文件找出蛛丝马迹，以尽快将真凶缉拿归案呢？