

山东大学计算机科学与技术学院

大数据分析课程实验报告

学号：202300130028

姓名：苗雨健

班级：数据 23

实验题目：BERT 环境配置

实验学时：2

实验日期：2025/11/10

实验目标：
尝试采用远程服务器运行代码

实验步骤与内容：
云服务器配置：继续使用小组作业配置的服务器

显卡类型

A30-24G

A10-24G

PRO 6000-96G

H100-80GB

A800-80GB

A100-80GB

A100-40GB

5880Ada-48G

A6000-48G

L40S-48G

8000-48G

5000-16G

V100-16G

P100-16G

5090-32G

4090-24G

4090-48G

4090D-24G

M40-24G

4080-16G

4070 Ti-12G

3090 Ti-24G

4070-12G

4060 Ti-16G

3090-24G

3080 Ti-12G

3080-10G

3080-20G

4060-8G

3070-8G

3060 Ti-8G

3060-12G

2080 Ti-22G

2080 Ti-11G

2070 SUPER-8G

1080 Ti-12G

5060 Ti-16G

V100-32G

P40-24G

T4-16G

P4-8G

TITAN X-12G

如何选卡?

GPU数量

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

多机多卡需求请联系客服开通

GPU类型	地区	数量	显存	显卡驱动版本	最高CUDA版本	CPU 型号	CPU配置	内存	实例磁盘	网络	价格	到期时间	可用性
3090-24G	华北	1	24 GB	535.161.08	12.2	Intel(R) Xeon(R) CPU E5-2682 v4	32核	64G	系统盘：20G 数据盘：50GB NVME (关机保留24小时/可扩容至800 GB)	U: 240 Mbps/s D: 800 Mbps/s	¥0.88/小时	2025-12-21	-

数据盘：免费 50GB ☐ 需要扩容

实例镜像

官方镜像

备份镜像

镜像市场

PyTorch / 2.0.0 / 11.7.0 / 3.8

没有需要的镜像?

Pycharm 的 ssh 链接：
链接成功，唤起 conda 环境

```
(pythonProject) PS C:\Users\33566\Desktop\202300130028-苗雨健\实验四> conda activate chinatravel
(chinatravel) PS C:\Users\33566\Desktop\202300130028-苗雨健\实验四> python test_cuda.py
```

测试 cuda

```

import torch
# 检查CUDA是否可用
cuda_available = torch.cuda.is_available()
print(f"CUDA 是否可用: {cuda_available}")
if cuda_available:
    # 检查当前可用的GPU数量
    gpu_count = torch.cuda.device_count()
    print(f"可用的 GPU 数量: {gpu_count}")
    # 获取当前GPU名称
    for i in range(gpu_count):
        print(f"GPU {i} 名称: {torch.cuda.get_device_name(i)}")
    # 检查当前设备
    current_device = torch.cuda.current_device()
    print(f"当前使用的设备索引: {current_device}")
else:
    print("未检测到可用的 CUDA 设备")

```

结果:

```

(chinatravel) PS C:\Users\33566\Desktop\202300130028-苗雨健\实验四> python test_bert.py
CUDA 是否可用: True
可用的 GPU 数量: 1
GPU 0 名称: NVIDIA GeForce RTX 3060
当前使用的设备索引: 0

```

测试 bert

```

def predict_sentiment(text): # 1个用法
    # 编码输入
    inputs = tokenizer(
        text,
        return_tensors='pt',
        padding=True,
        truncation=True,
        max_length=512
    )
    inputs = {k: v.to(device) for k, v in inputs.items()}

    # 推理
    with torch.no_grad():
        outputs = model(**inputs)

    logits = outputs.logits
    probs = torch.softmax(logits, dim=-1)
    sentiment = torch.argmax(probs, dim=-1).item()
    return "积极" if sentiment == 1 else "消极", probs

# 测试
text = "The weather is great today, I feel very happy!"
sentiment, probability = predict_sentiment(text)
print(f"文本: {text}")

```

结果:

```

(chinatravel) PS C:\Users\33566\Desktop\202300130028-苗雨健\实验四> python test_bert.py
Warning: Model is not fine-tuned, using pre-trained base model, prediction results may not be accurate
Warning: transformers library version 4.21.0 is outdated, consider upgrading to the latest version for better performance and features

文本: The weather is great today, I feel very happy!
预测的情感: 积极
概率: tensor([[0.1523, 0.8477]], device='cuda:0')

```

结论分析与体会：

本次实验成功搭建了兼容 BERT 模型的远程环境，PyTorch、CUDA、transformers 等组件版本匹配，GPU 调用正常，通过本地加载 bert-base-uncased 模型完成测试，验证了全流程可用性。体会到版本兼容性和国内镜像源是环境配置的关键，远程服务器与 VSCode 的结合既利用了高性能 GPU，又方便调试，本地模型加载则提升了效率、降低了网络依赖，这些技巧为后续机器学习任务提供了实用参考。