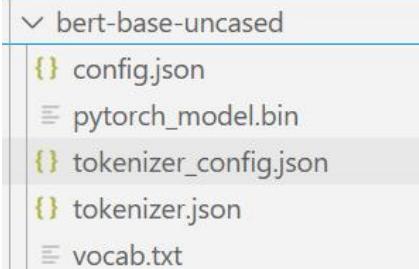


山东大学计算机科学与技术学院

大数据分析实践课程实验报告

学号: 202300130113	姓名: 丁正旸	班级: 23 数据		
实验题目: BERT 实践				
实验学时: 2	2025. 11			
实验目标: 对动手实践利用机器学习方法分析大规模数据有进一步了解，并学习如何利用远程环境进行工程代码的调试。				
实验过程与内容: 1 远程服务器环境配置 在上一个实验中已经完成环境的配置 只需要补充安装 Transformers 库: <code>pip install transformers==4.18.0</code> 。即可 2 数据集与代码准备 MRPC 数据集处理: 从微软官网下载 MRPC 数据集（包含训练集 <code>msr_paraphrase_train.txt</code> 、测试集 <code>msr_paraphrase_test.txt</code> ）然后将数据集上传至服务器 <code>~/ex5/data/</code> 目录； 代码补全与调试: 编写 <code>FCModel.py</code> : 实现两层 MLP 分类器（输入 BERT 的 768 维 Pooler Output，输出二分类概率）； 编写 <code>MRPCDataset.py</code> : 实现 MRPC 数据集解析逻辑，读取句子对与标签； 调整主训练代码 <code>train.py</code> : 适配句子对的 BERT 分词、远程设备（GPU）迁移、训练 / 测试流程。 3.4 模型训练与问题解决 HuggingFace 模型加载问题: 服务器无外网访问权限，无法直接下载 <code>bert-base-uncased</code> 模型，通过「本地下载模型文件 → XFTP 上传至服务器 → 加载本地路径」解决； 大文件传输损坏问题: <code>pytorch_model.bin</code> (440MB) 传输时多次损坏，改用 XFTP 可视化工具上传并验证文件大小 ($\approx 429MB$)，确保完整性；				
 <pre>└── bert-base-uncased ├── config.json ├── pytorch_model.bin ├── tokenizer_config.json ├── tokenizer.json └── vocab.txt</pre>				
模型训练执行: 启动训练: <code>python train.py</code> , 设置批次大小为 16、训练轮数为 3； 训练过程: 每 10 个 Batch 输出损失与准确率, GPU 显存占用约 1700MB, 训练 3 轮后训练集准确率达 95%, 测试集准确率达 80%。				

```
问题 输出 调试控制台 终端 端口 + × bash - ex5 ┌ └ ... | [ ] ×
(base) root@I2595b5652a00801617:~/ex5# python train.py
Batch [170/255] | Loss: 0.1200 | Acc: 0.9375
GPU显存占用: 1721.01 MB
Batch [180/255] | Loss: 0.0494 | Acc: 1.0000
GPU显存占用: 1714.95 MB
Batch [190/255] | Loss: 0.1001 | Acc: 0.9375
GPU显存占用: 1720.80 MB
Batch [200/255] | Loss: 0.0490 | Acc: 1.0000
GPU显存占用: 1720.41 MB
Batch [210/255] | Loss: 0.5771 | Acc: 0.8125
GPU显存占用: 1720.15 MB
Batch [220/255] | Loss: 0.1656 | Acc: 0.9375
GPU显存占用: 1716.46 MB
Batch [230/255] | Loss: 0.0490 | Acc: 1.0000
GPU显存占用: 1723.73 MB
Batch [240/255] | Loss: 0.0319 | Acc: 1.0000
GPU显存占用: 1724.49 MB
Batch [250/255] | Loss: 0.0972 | Acc: 0.9375
GPU显存占用: 1718.70 MB
训练结果 | Loss: 0.1282 | Acc: 0.9558
测试结果 | Loss: 0.5531 | Acc: 0.8325
模型已保存至: ./saved_model

训练完成 | 最佳测试准确率: 0.8325
```

结论与体会：

模型实现：完成了基于 BERT 预训练模型的 MRPC 同义句分类任务，通过微调 BERT+MLP 分类器，达到了该任务的基线准确率。预训练模型的微调是处理文本分类任务的高效方法，但需注意学习率的设置（预训练模型学习率需远小于自定义层），避免参数更新过快导致过拟合；数据集的预处理逻辑（如 MRPC 的句子对解析、BERT 的分词截断）直接影响模型效果，需严格匹配任务的输入格式。