山东大学计算机科学与技术学院

大数据分析与实践课程实验报告

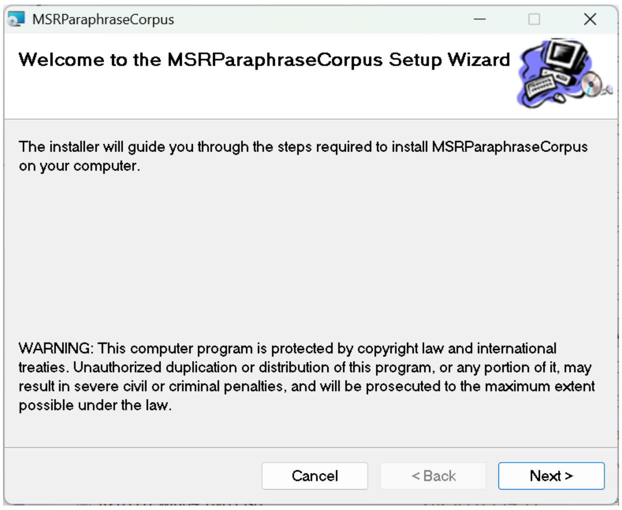
学号：202300130063	姓名：邱珺	班级：23 数据
实验题目：实验六 Bert 实践		
实验学时：2	实验日期：2025/11/7	

实验目标：本次实验主要围绕 MRPC 句子对数据集展开，旨在通过构建一个基于 BERT + 全连接层分类器的模型，学习自然语言处理中句子语义相似度判断（同义句识别）的基本流程。实验内容包括：数据集的加载与预处理、BERT 模型的调用与微调、全连接分类层的构建、模型训练及准确率计算。通过本实验，我们能够掌握利用 PyTorch 和 Hugging Face Transformers 库搭建文本分类模型的完整过程，理解预训练语言模型在下游任务中的应用方法。

实验过程：

1. 数据加载与预处理

首先下载数据集：



通过 MRPCDataset.py 读取并解析 MRPC 数据集的 train.txt 文件，将每条

数据转换为句子对 (sent1, sent2) 与对应标签 label。使用自定义的 collate_fn 函数整理批次数据，使其适配 BERT 的输入格式。

```
成功加载train集：4076条数据
数据载入完成（句子对格式）
使用设备：cuda
```

2. 模型加载与设备配置

在 main.py 中检测系统可用设备（CUDA、MPS 或 CPU），并自动选择最优运行环境。离线加载本地预训练的 bert-base-uncased 模型和对应分词器（BertModel 与 BertTokenizer），避免联网依赖问题。

```
已从本地加载 BERT 模型：c:/Workspace/SDU/Experiments/BigDataAnalysis/bert\model (当前设备：cuda
全连接分类模型创建完成
```

3. 模型结构设计

使用预训练的 BERT 模型提取句子对的语义特征（pooler_output）。构建自定义的 全连接分类器 FCModel，包含两层线性映射、ReLU 激活、Dropout 正则与 Sigmoid 输出，用于将语义特征映射到相似度概率（0~1）。

4. 训练流程与优化策略

采用 二分类交叉熵损失函数（BCELoss） 作为优化目标。使用两个优化器分别更新 BERT 参数（学习率 2e-5）与分类层参数（学习率 1e-3），防止过度破坏预训练特征。在每个批次中，将句子对送入 BERT 得到特征，再输入 FCModel 得到预测结果，计算损失与准确率，执行反向传播与参数更新。

5. 模型训练与评估

设置多轮（8 个 epoch）训练循环，每轮输出平均损失与准确率。通过连续训练与评估，观察模型在语义相似度识别任务上的收敛趋势与性能变化。

```
===== Epoch 1/8 =====
batch 10 | loss: 0.5549 | acc: 0.8125
batch 20 | loss: 0.5416 | acc: 0.7500
batch 30 | loss: 0.5410 | acc: 0.7500
batch 40 | loss: 0.5828 | acc: 0.7500
batch 50 | loss: 0.8597 | acc: 0.5625
batch 60 | loss: 0.4984 | acc: 0.8125
batch 70 | loss: 0.4767 | acc: 0.8750
batch 80 | loss: 0.5993 | acc: 0.6250
batch 90 | loss: 0.5513 | acc: 0.6875
batch 100 | loss: 0.4295 | acc: 0.7500
batch 110 | loss: 0.5481 | acc: 0.6875
batch 120 | loss: 0.5756 | acc: 0.6875
batch 130 | loss: 0.4850 | acc: 0.8125
batch 140 | loss: 0.4644 | acc: 0.8125
batch 150 | loss: 0.3728 | acc: 0.8750
batch 160 | loss: 0.3553 | acc: 0.8125
batch 170 | loss: 0.3958 | acc: 0.7500
batch 180 | loss: 0.4562 | acc: 0.7500
batch 190 | loss: 0.4570 | acc: 0.7500
batch 200 | loss: 0.7987 | acc: 0.6250
batch 210 | loss: 0.2436 | acc: 1.0000
batch 220 | loss: 0.5097 | acc: 0.7500
batch 230 | loss: 0.3538 | acc: 0.7500
batch 240 | loss: 0.3390 | acc: 0.8125
batch 250 | loss: 0.6124 | acc: 0.7500
Epoch 1 结束 | 平均损失: 0.5232 | 平均准确率: 0.7446
```

```
===== Epoch 2/8 =====
batch 10 | loss: 0.4415 | acc: 0.7500
batch 20 | loss: 0.1609 | acc: 0.9375
batch 30 | loss: 0.0699 | acc: 1.0000
batch 40 | loss: 0.2544 | acc: 0.9375
batch 50 | loss: 0.1774 | acc: 0.8750
batch 60 | loss: 0.7458 | acc: 0.8125
batch 70 | loss: 0.3444 | acc: 0.8750
batch 80 | loss: 0.2303 | acc: 0.8750
batch 90 | loss: 0.1806 | acc: 1.0000
batch 100 | loss: 0.1690 | acc: 1.0000
batch 110 | loss: 0.1557 | acc: 0.9375
batch 120 | loss: 0.1592 | acc: 0.9375
batch 130 | loss: 0.2400 | acc: 0.8750
batch 140 | loss: 0.4075 | acc: 0.8750
batch 150 | loss: 0.2788 | acc: 0.8750
batch 160 | loss: 0.2212 | acc: 0.8750
batch 170 | loss: 0.3354 | acc: 0.8125
batch 180 | loss: 0.3523 | acc: 0.8750
batch 190 | loss: 0.2539 | acc: 0.8750
batch 200 | loss: 0.2728 | acc: 0.9375
batch 210 | loss: 0.2213 | acc: 0.9375
batch 220 | loss: 0.2835 | acc: 0.8750
batch 230 | loss: 0.0989 | acc: 1.0000
batch 240 | loss: 0.5629 | acc: 0.7500
batch 250 | loss: 0.2678 | acc: 0.9375
Epoch 2 结束 | 平均损失: 0.2858 | 平均准确率: 0.8852
```

```
===== Epoch 3/8 =====
batch 10 | loss: 0.1443 | acc: 0.9375
batch 20 | loss: 0.0544 | acc: 1.0000
batch 30 | loss: 0.0140 | acc: 1.0000
batch 40 | loss: 0.0050 | acc: 1.0000
batch 50 | loss: 0.3593 | acc: 0.9375
batch 60 | loss: 0.0163 | acc: 1.0000
batch 70 | loss: 0.0490 | acc: 1.0000
batch 80 | loss: 0.1309 | acc: 0.9375
batch 90 | loss: 0.3242 | acc: 0.9375
batch 100 | loss: 0.0944 | acc: 0.9375
batch 110 | loss: 0.0728 | acc: 0.9375
batch 120 | loss: 0.0512 | acc: 1.0000
batch 130 | loss: 0.0704 | acc: 0.9375
batch 140 | loss: 0.0152 | acc: 1.0000
batch 150 | loss: 0.0393 | acc: 1.0000
batch 160 | loss: 0.1800 | acc: 0.8750
batch 170 | loss: 0.0732 | acc: 0.9375
batch 180 | loss: 0.0139 | acc: 1.0000
batch 190 | loss: 0.0804 | acc: 0.9375
batch 200 | loss: 0.0514 | acc: 1.0000
batch 210 | loss: 0.0048 | acc: 1.0000
batch 220 | loss: 0.1379 | acc: 0.9375
batch 230 | loss: 0.0286 | acc: 1.0000
batch 240 | loss: 0.0273 | acc: 1.0000
batch 250 | loss: 0.0422 | acc: 1.0000
Epoch 3 结束 | 平均损失: 0.0984 | 平均准确率: 0.9684
```

```
===== Epoch 4/8 =====
batch 10 | loss: 0.0018 | acc: 1.0000
batch 20 | loss: 0.0117 | acc: 1.0000
batch 30 | loss: 0.0065 | acc: 1.0000
batch 40 | loss: 0.0248 | acc: 1.0000
batch 50 | loss: 0.0703 | acc: 0.9375
batch 60 | loss: 0.0032 | acc: 1.0000
batch 70 | loss: 0.0073 | acc: 1.0000
batch 80 | loss: 0.0102 | acc: 1.0000
batch 90 | loss: 0.0298 | acc: 1.0000
batch 100 | loss: 0.0004 | acc: 1.0000
batch 110 | loss: 0.0071 | acc: 1.0000
batch 120 | loss: 0.0097 | acc: 1.0000
batch 130 | loss: 0.0022 | acc: 1.0000
batch 140 | loss: 0.0283 | acc: 1.0000
batch 150 | loss: 0.0023 | acc: 1.0000
batch 160 | loss: 0.0047 | acc: 1.0000
batch 170 | loss: 0.0080 | acc: 1.0000
batch 180 | loss: 0.0393 | acc: 1.0000
batch 190 | loss: 0.0218 | acc: 1.0000
batch 200 | loss: 0.0313 | acc: 1.0000
batch 210 | loss: 0.0232 | acc: 1.0000
batch 220 | loss: 0.0047 | acc: 1.0000
batch 230 | loss: 0.0148 | acc: 1.0000
batch 240 | loss: 0.3686 | acc: 0.9375
batch 250 | loss: 0.0292 | acc: 1.0000
Epoch 4 结束 | 平均损失: 0.0455 | 平均准确率: 0.9872
```

```
===== Epoch 5/8 =====
batch 10 | loss: 0.0192 | acc: 1.0000
batch 20 | loss: 0.0071 | acc: 1.0000
batch 30 | loss: 0.0242 | acc: 1.0000
batch 40 | loss: 0.2278 | acc: 0.9375
batch 50 | loss: 0.0003 | acc: 1.0000
batch 60 | loss: 0.0047 | acc: 1.0000
batch 70 | loss: 0.0305 | acc: 1.0000
batch 80 | loss: 0.0054 | acc: 1.0000
batch 90 | loss: 0.0225 | acc: 1.0000
batch 100 | loss: 0.0247 | acc: 1.0000
batch 110 | loss: 0.0013 | acc: 1.0000
batch 120 | loss: 0.0037 | acc: 1.0000
batch 130 | loss: 0.0021 | acc: 1.0000
batch 140 | loss: 0.0070 | acc: 1.0000
batch 150 | loss: 0.0118 | acc: 1.0000
batch 160 | loss: 0.0174 | acc: 1.0000
batch 170 | loss: 0.0327 | acc: 1.0000
batch 180 | loss: 0.0031 | acc: 1.0000
batch 190 | loss: 0.0339 | acc: 1.0000
batch 200 | loss: 0.0945 | acc: 0.9375
batch 210 | loss: 0.0028 | acc: 1.0000
batch 220 | loss: 0.0054 | acc: 1.0000
batch 230 | loss: 0.0312 | acc: 1.0000
batch 240 | loss: 0.0008 | acc: 1.0000
batch 250 | loss: 0.0004 | acc: 1.0000
Epoch 5 结束 | 平均损失: 0.0431 | 平均准确率: 0.9877
```

```
===== Epoch 6/8 =====
batch 10 | loss: 0.0004 | acc: 1.0000
batch 20 | loss: 0.0015 | acc: 1.0000
batch 30 | loss: 0.0004 | acc: 1.0000
batch 40 | loss: 0.0429 | acc: 1.0000
batch 50 | loss: 0.0008 | acc: 1.0000
batch 60 | loss: 0.0012 | acc: 1.0000
batch 70 | loss: 0.4349 | acc: 0.9375
batch 80 | loss: 0.0191 | acc: 1.0000
batch 90 | loss: 0.0064 | acc: 1.0000
batch 100 | loss: 0.0071 | acc: 1.0000
batch 110 | loss: 0.0010 | acc: 1.0000
batch 120 | loss: 0.0001 | acc: 1.0000
batch 130 | loss: 0.0222 | acc: 1.0000
batch 140 | loss: 0.0001 | acc: 1.0000
batch 150 | loss: 0.0010 | acc: 1.0000
batch 160 | loss: 0.0025 | acc: 1.0000
batch 170 | loss: 0.0587 | acc: 1.0000
batch 180 | loss: 0.0209 | acc: 1.0000
batch 190 | loss: 0.0016 | acc: 1.0000
batch 200 | loss: 0.1919 | acc: 0.9375
batch 210 | loss: 0.0017 | acc: 1.0000
batch 220 | loss: 0.0267 | acc: 1.0000
batch 230 | loss: 0.0007 | acc: 1.0000
batch 240 | loss: 0.0007 | acc: 1.0000
batch 250 | loss: 0.0120 | acc: 1.0000
Epoch 6 结束 | 平均损失: 0.0364 | 平均准确率: 0.9872
```

```
===== Epoch 7/8 =====
batch 10 | loss: 0.0415 | acc: 1.0000
batch 20 | loss: 0.0269 | acc: 1.0000
batch 30 | loss: 0.0035 | acc: 1.0000
batch 40 | loss: 0.0423 | acc: 1.0000
batch 50 | loss: 0.0057 | acc: 1.0000
batch 60 | loss: 0.0015 | acc: 1.0000
batch 70 | loss: 0.0009 | acc: 1.0000
batch 80 | loss: 0.0042 | acc: 1.0000
batch 90 | loss: 0.0025 | acc: 1.0000
batch 100 | loss: 0.0007 | acc: 1.0000
batch 110 | loss: 0.0003 | acc: 1.0000
batch 120 | loss: 0.0013 | acc: 1.0000
batch 130 | loss: 0.0077 | acc: 1.0000
batch 140 | loss: 0.0022 | acc: 1.0000
batch 150 | loss: 0.0006 | acc: 1.0000
batch 160 | loss: 0.0159 | acc: 1.0000
batch 170 | loss: 0.0435 | acc: 1.0000
batch 180 | loss: 0.0062 | acc: 1.0000
batch 190 | loss: 0.0004 | acc: 1.0000
batch 200 | loss: 0.0037 | acc: 1.0000
batch 210 | loss: 0.0004 | acc: 1.0000
batch 220 | loss: 0.0004 | acc: 1.0000
batch 230 | loss: 0.0075 | acc: 1.0000
batch 240 | loss: 0.0004 | acc: 1.0000
batch 250 | loss: 0.0063 | acc: 1.0000
Epoch 7 结束 | 平均损失: 0.0226 | 平均准确率: 0.9948
```

```
===== Epoch 8/8 =====
batch 10 | loss: 0.0040 | acc: 1.0000
batch 20 | loss: 0.0038 | acc: 1.0000
batch 30 | loss: 0.0022 | acc: 1.0000
batch 40 | loss: 0.0000 | acc: 1.0000
batch 50 | loss: 0.0003 | acc: 1.0000
batch 60 | loss: 0.0118 | acc: 1.0000
batch 70 | loss: 0.0017 | acc: 1.0000
batch 80 | loss: 0.0004 | acc: 1.0000
batch 90 | loss: 0.0007 | acc: 1.0000
batch 100 | loss: 0.0002 | acc: 1.0000
batch 110 | loss: 0.0000 | acc: 1.0000
batch 120 | loss: 0.0006 | acc: 1.0000
batch 130 | loss: 0.0001 | acc: 1.0000
batch 140 | loss: 0.0154 | acc: 1.0000
batch 150 | loss: 0.0014 | acc: 1.0000
batch 160 | loss: 0.0003 | acc: 1.0000
batch 170 | loss: 0.0031 | acc: 1.0000
batch 180 | loss: 0.0030 | acc: 1.0000
batch 190 | loss: 0.0138 | acc: 1.0000
batch 200 | loss: 0.0009 | acc: 1.0000
batch 210 | loss: 0.0013 | acc: 1.0000
batch 220 | loss: 0.0047 | acc: 1.0000
batch 230 | loss: 0.0008 | acc: 1.0000
batch 240 | loss: 0.0031 | acc: 1.0000
batch 250 | loss: 0.0108 | acc: 1.0000
Epoch 8 结束 | 平均损失: 0.0200 | 平均准确率: 0.9944
训练完成！
```

结论分析：

本实验围绕句子语义相似度判定任务展开，旨在掌握基于预训练语言模型的文本语义理解与分类流程。实验以 MRPC 句子对数据集为基础，首先完成数据加载与结构探索，明确同义句识别任务目标；随后利用 BERT 模型提取句子语义特征，并构建全连接层分类器进行训练与验证。通过模型训练、优化与准确率评估的完整流程，深化了对深度语言模型在自然语言处理中的应用理解，掌握了文本预处理、特征表示及模型微调的核心方法，为后续语义分析与智能文本理解任务奠定基础。