

# 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: 202300130030	姓名: 赵汉哲	班级: 数据 23
实验题目: bert 实践		
实验学时: 4	实验日期: 2025. 11. 7	
<p><b>实验目标:</b> 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.</p>		
<p><b>实验描述:</b></p> <p><b>一. 实验步骤</b></p> <ol style="list-style-type: none"><li>1. 配置环境:<ol style="list-style-type: none"><li>a. 使用 SSH 连接远程服务器，例如可以使用 XHELL</li><li>b. 从 Anaconda 的官网下载 for Linux 的安装包，用 FTP 传输至服务器上想要的安装位置，并使用 bash 命令在该位置进行安装</li><li>c. 使用 conda create 命令创建 base 环境以外的虚拟环境</li><li>d. 使用 conda activate 命令进行虚拟环境的切换</li><li>e. 安装所需要的包</li></ol></li><li>2. 对 PyCharm 配置(该步骤参考的 PyCharm 版本为 2021.3.3 Professional Edition, 日后的版本可能操作略有不同):<ol style="list-style-type: none"><li>a. 下载 PyCharm 专业版，并打开已有的代码项目</li><li>b. 配置 SSH configuration, 连接服务器</li><li>c. 设置 deployment, 在其中设置路径映射</li><li>d. 配置 Python Interpreter, 使用已经配置好的 deployment 的配置</li><li>e. 设置 deployment 自动同步，并手动进行第一次上传已有的代码.</li></ol></li><li>3. 熟悉 PyTorch 框架下，利用预训练的 transformers 的预训练 BERT 模型对 MRPC 数据集进行同义预测的 pipeline. 尝试理解数据是如何预处理，模型是怎么读入数据，是如何进行推理，如何进行评价的. (代码逻辑: 对 BERT 进行微调，每个句子对用 BERT 指定分隔符 [SEP] 连接后，通过 BERT 得到合成句子的 representation. 再通过一个两层的多层感知机得到分类结果. 这里预训练 BERT 模型使用的是 HuggingFace 的 BERT-base-uncased)</li></ol> <p><b>二. 数据集</b></p>		

MRPC (Microsoft Research Paraphrase Corpus) 包含了 5800 个句子对，有的是同义的，有的是不同义的，是否同义由一个二元标签进行描述。

数据集下载链接:<https://www.microsoft.com/en-us/download/details.aspx?id=52398>

### 三. 代码

#### 1. 加载数据集 MRPCDataset:

按行读取文件 `msr_paraphrase_train.txt`, 提取出 `sentences` 和 `labels`。因为数据集的问题，某些行含有 tab 制表符，导致加载数据出错（包括分隔符每行应为 5 列，多余的制表符导致每行大于五列），这里将多出的列合并到 `s2` 中：

```
# 将多出的列合并到s2中
s2 = " ".join(cols[4:])
```

#### 2. 全连接层 FCModel:

根据 BERT-base-uncased 的形状定义全连接层：

```
def __init__(self):
    super(FCModel, self).__init__()
    # BERT-base-uncased的pooler_output维度为768
    self.fc1 = nn.Linear(in_features: 768, out_features: 256) # 第一层全连接
    self.fc2 = nn.Linear(in_features: 256, out_features: 1) # 输出层（二分类用sigmoid）
    self.relu = nn.ReLU()
    self.dropout = nn.Dropout(0.3) # 防止过拟合
```

#### 3. 训练代码 train.py:

使用 `MRPCDataset()` 加载数据，并定义 `bert_model` 和全连接模型 `model`，定义损失函数和优化器，使用 `bert` 和全连接层进行模型训练。

### 结果图片：

```
batch 251 loss:0.558984 accuracy:0.750000
torch.cuda.memory_allocated(): 1768050688
batch 252 loss:0.623935 accuracy:0.687500
torch.cuda.memory_allocated(): 1768546816
batch 253 loss:0.764593 accuracy:0.500000
torch.cuda.memory_allocated(): 1768546816
batch 254 loss:0.687332 accuracy:0.583333
EPOCH 1 loss:0.648986 accuracy:0.665113
```

