

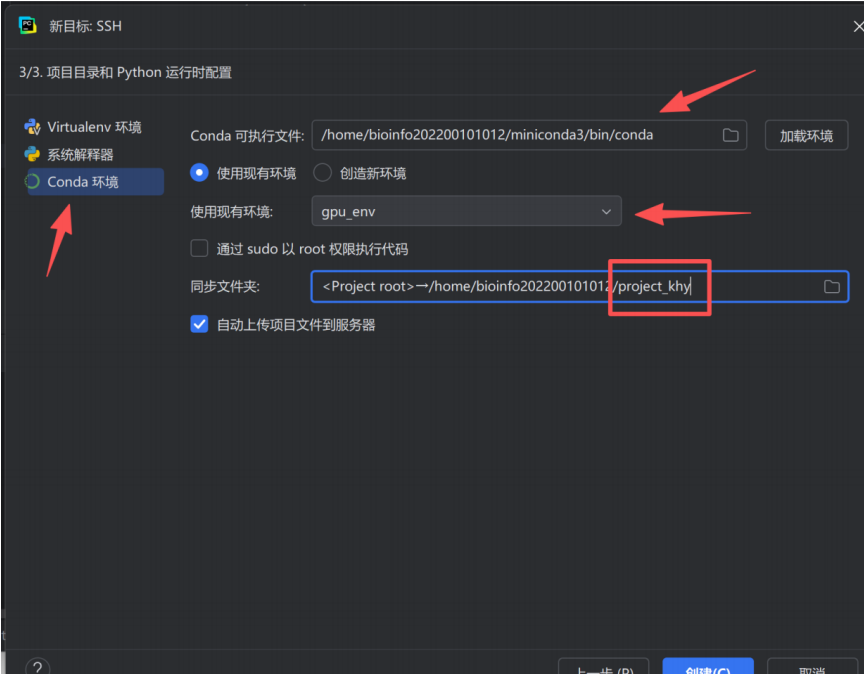
山东大学计算机科学与技术学院

大数据分析实践课程实验报告

学号：202200101012	姓名：康海洋	班级：数据23
实验题目：Bert实践		
实验学时：2	实验日期：2025.11.06	
实验目标： 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.		

实验步骤：

Pycharm连接远程服务器



The screenshot shows the 'SSH' configuration window in PyCharm, specifically the '3/3. 项目目录和 Python 运行时配置' (Project directories and Python runtime configuration) tab. On the left, 'Conda 环境' (Conda environment) is selected. The main area shows 'Conda 可执行文件' (Conda executable) set to '/home/bioinfo202200101012/miniconda3/bin/conda'. Below this, '使用现有环境' (Use existing environment) is selected, and the dropdown menu shows 'gpu_env'. The '同步文件夹' (Sync folders) section shows '<Project root> -> /home/bioinfo202200101012/project_khy'. Red arrows point to the 'Conda 环境' selection, the 'Conda 可执行文件' field, the 'gpu_env' dropdown, and the project path field. A red box highlights the project path field.

目录层级

代码块

```
1 project_khy/
2 |
```

```
3 |— bert.py
4 |— FCModel.py
5 |— MRPCDataset.py
6 |— msr_paraphrase_train.txt
7 |
8 |— distilbert-base-uncased/
9     |— config.json
10     |— vocab.txt
11     |— tokenizer.json
12     |— tokenizer_config.json
13     |— pytorch_model.bin
14
```

1. bert.py — 训练 BERT 的主程序

作用：

- 加载数据集
- 加载 tokenizer
- 加载 BERT (DistilBERT)
- 加载你的 FCModel (分类头)
- 定义优化器
- 执行训练 (for epoch)
- 打印 loss / accuracy
- 使用 GPU

这是整个实验的“主入口”。

2. FCModel.py — 分类头 (MLP)

作用：

- 接收 BERT 输出的句向量 (768 维)
- 通过 1~2 层全连接神经网络分类
- 输出一个 0~1 的概率 (同义 or 不同义)

训练的就是这个部分 + 微调 BERT 的参数。

3. MRPCDataset.py — 自定义数据集类

作用：

- 读取 MRPC 数据文件 (msr_paraphrase_train.txt)
- 提取：
 - sentence1

- sentence2

- label (0/1)

- 构造成 PyTorch 能读取的 dataset
- 返回给 DataLoader 供训练使用

4. msr_paraphrase_train.txt —— 训练数据集（微软 MRPC）

作用：

- 每行是一个句子对
- 带有它们是否是“同义句”的标签（0 或 1）
- 你训练用的数据来源

结果图片：

预处理完成：

```
/home/bioinfo202200101012/miniconda3/bin/conda run -n gpu_env --no-capture-output python /home/bioinfo202200101012/project_khy/bert.py
✅ 数据加载完成：总共有效样本 4076 条
✅ 数据载入完成，共 4076 条数据
✅ 设备配置完成，当前设备：cuda
Some weights of the model checkpoint at ./distilbert-base-uncased were not used when initializing DistilBertModel: ['vocab_layer_norm.bias', 'vocab_layer_norm.weight', 'vocab_proj.bias', 'vocab_proj.weight']
- This IS expected if you are initializing DistilBertModel from the checkpoint of a model trained on another task or with another architecture (e.g. initializing a BertForSequenceClassification model from a BertForPreTraining model).
- This IS NOT expected if you are initializing DistilBertModel from the checkpoint of a model that you expect to be exactly identical (initializing a BertForSequenceClassification model from a BertForSequenceClassification model).
✅ BERT 模型加载完成
✅ 全连接层 FCModel 创建完成
GPU 已使用显存 bytes: 267377152
Batch 0: loss=0.6866, acc=0.5625
GPU 已使用显存 bytes: 1092224000
Batch 1: loss=0.6162, acc=0.6875
GPU 已使用显存 bytes: 1092965376
Batch 2: loss=0.4882, acc=0.8125
GPU 已使用显存 bytes: 1092668416
Batch 3: loss=0.6881, acc=0.6875
GPU 已使用显存 bytes: 1092470784
Batch 4: loss=0.6418, acc=0.6875
```

实验过程：

```
Batch 10: loss=0.6018, acc=0.7500
GPU 已使用显存 bytes: 1092570112
Batch 11: loss=0.5946, acc=0.6875
GPU 已使用显存 bytes: 1093261312
Batch 12: loss=0.7212, acc=0.6250
GPU 已使用显存 bytes: 1093360640
Batch 13: loss=0.5844, acc=0.6875
GPU 已使用显存 bytes: 1092816896
Batch 14: loss=0.6309, acc=0.6250
GPU 已使用显存 bytes: 1092767744
Batch 15: loss=0.6907, acc=0.5625
GPU 已使用显存 bytes: 1093261312
Batch 16: loss=0.8214, acc=0.3750
GPU 已使用显存 bytes: 1093360640
Batch 17: loss=0.8028, acc=0.3750
GPU 已使用显存 bytes: 1092470784
Batch 18: loss=0.6212, acc=0.6875
GPU 已使用显存 bytes: 1092965376
Batch 19: loss=0.6248, acc=0.6250
```

运行完成:

```
Batch 251: loss=0.9072, acc=0.6875
GPU 已使用显存 bytes: 1092421632
Batch 252: loss=0.4484, acc=0.8125
GPU 已使用显存 bytes: 1092470784
Batch 253: loss=0.4617, acc=0.8125
GPU 已使用显存 bytes: 1092372480
Batch 254: loss=0.3886, acc=0.6667
✅ EPOCH 1 finished: loss=0.5342, acc=0.7309
```

进程已结束, 退出代码为 0