Project 4: 基于预训练 Transformer 模型的 WNLI 分类

环境配置

考虑到预训练 Tranformer 模型参数量不小，建议拥有 NVIDIA GPU 的同学配置 CUDA 环境。

NVIDIA GPU

Windows 系统，打开任务管理器，你能在“性能”页面中看到一个名称以 NVIDIA 开头的 GPU。

第一步：检查驱动支持的 CUDA 版本

请在终端中输入 `nvidia-smi`，查看输出中右上角的 CUDA Version，推荐大于 11.8。如果你的 CUDA 版本过低，可以考虑更新 GPU 驱动。

第二步：安装 CUDA Toolkit

前往 CUDA 官方网站 https://developer.nvidia.com/cuda-toolkit 下载 CUDA Toolkit，你选择下载安装版本应当小于等于你在第一步里看到的 CUDA Version。

第三步：安装依赖

以 CUDA 12.1 为例。

```
pip install -r requirements.txt --extra-index=https://download.pytorch.org/whl/cu121
```

`extra-index` 参数最后的 “cu121” 根据你的 CUDA 版本在 cu118、cu121 和 cu124 三个选项中选择。这个版本需要小于等于你安装的 CUDA Toolkit 的版本。

CPU

还有很多别的加速设备，如 Apple Silicon、AMD GPU、Intel GPU 和 Ascend NPU 等可以用于加速训练，但使用这些设备进行深度学习并非主流。持有这些设备且感兴趣的同学可以自行调研尝试。

> 一些关键词：
> Apple: MLX, MPS
> AMD: ROCm
> Intel: itrex
> Ascend: torch_npu

其他同学可以安装 CPU 版环境。

```
pip install -r requirements.txt
```

模型训练

进行训练

```
python -u train.py --model=prajjwal1/bert-tiny --lr=1e-4 --epoch=1 --bs=1 --
output=./output/v1
```

- model 是训练使用的预训练模型的名称，你可以在 Huggingface Hub 中寻找任何一个可以用于 Text Classification 任务的模型来进行实验。
- lr 是学习率
- epoch 是训练的轮数
- bs 是批大小，请根据你的显存大小来设置这个参数
- accum 是梯度累积步数。如果你的显存大小不足以支撑一个较大的 batch size，你可以通过调整这个参数来获得等效的更大的批大小。优化最终等效的批大小为 bs*accum。
- output 是模型的保存位置。

首次运行时，程序会自动下载数据集和模型。如遇网络原因，请设置环境变量 `HF_ENDPOINT=https://hf-mirror.com`，环境变量的设置方法请自行了解。

训练代码讲解

加载数据集

`load_dataset("SetFit/wnli")` 使用 Huggingface 提供的 `datasets` 库加载了 wnli 数据集。他在 Huggingface 上的链接为：https://huggingface.co/datasets/SetFit/wnli

`AutoTokenizer.from_pretrained(args.model, trust_remote_code=True)` 加载了预训练 Tokenizer。

`preprocess_function` 使用 Tokenizer 对数据集进行了预处理。

`AutoModelForSequenceClassification.from_pretrained` 加载了预训练模型用于序列分类任务。这个模型是一个预训练的 Transformer 后接一个随机初始化的 classifier。

`compute_metrics` 帮助我们在训练过程中随时评测模型性能。这里只观察了 Accuracy 指标。

`Trainer` 帮我们包装了整个训练循环。我们仅需为其提供训练需要的超参数即可。

探索

为了获得更高的 Accuracy，请参考以下方法：

- 尝试调节已提供的命令行参数。
- 参考 TrainingArguments 调节学习率调节器、Warm Up 策略、标签平滑等参数。
- 尝试修改数据集预处理函数 `preprocess_function`，增强模型的输入。
- 尝试其他基础模型。

评测

```
python -u test.py --model=PATH/TO/YOUR/CHECKPOINT
```

PATH/TO/YOUR/CHECKPOINT 是你模型保存的位置，你可以选择 Validation 集上 Accuracy 最高的那个 Checkpoint。

评分标准

你需要将你在测试集上的预测结果提交到 https://gluebenchmark.com/ 来获取正确率。请将测试结果设置为 Public，并在提交的文档中附上测试结果的链接。

- 完成 prajjwal1/bert-tiny 和任意一个自选模型的微调，并提交评测结果，即可获得基础分数。
- 暂定测试集正确率 0.7 为基本标准。达到 0.7 即可获得本次作业的大部分分数。最终基本正确率分数线将视同学们的整体正确率调整。
- 在报告中分析各项参数对正确率的影响，尤其是换不同模型对正确率的影响。

答疑

助教：郭虹麟
邮箱：hlguo24@m.fudan.edu.cn