

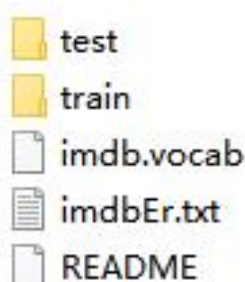
深度学习导论 DS2001.01.2023SP——实验四

实验要求

使用 `pytorch` 或者 `tensorflow` 分别编写基于 RNN (LSTM) 和 BERT 的语言模型，并基于训练好的词向量，实现文本情感分类 (Text Sentiment Classification)：输入一个句子，输出是 0 (负面) 或 1 (正面)，**并对两个不同模型的实验结果展开对比分析**。

实验步骤

- 网络框架：**要求选择 `pytorch` 或 `tensorflow` 其中之一，依据官方网站的指引安装包。本次实验强烈推荐大家安装 GPU 版本，否则将严重影响实验进度。如果你没有 GPU 机器，可考虑使用一些云资源，例如 Google Colab。
- 数据集：**本次实验统一使用指定的 IMDB 公开数据集 “Large Movie Review Dataset”。该数据集分别包含 25,000 条电影评论作为训练集和测试集。为使用该数据集，首先你需要下载原始数据 `aclImdb_v1.tar.gz` (地址：<https://ai.stanford.edu/~amaas/data/sentiment/>)。解压后你将会看到如下目录结构：



请仔细阅读 README 文件以了解每个文件的具体结构和数据格式。

- 数据预处理：**你需要通过 `pytorch` 或 `tensorflow` 所提供的标准数据接口，将原始数据处理为方便模型训练脚本所使用的数据结构，如 `torch.utils.data.Dataset` 等。由于该数据集是非常常见的公开数据集，你可以参考一些公开代码片段。

若要生成验证集要求只能从**训练集**中划分。训练只能在训练集上完成，实验调参只能在验证集上完成。

提示:对 `pytorch` 的 `Dataset` 类数据集划分可以选择直接调用 `torch.utils.data.random_split` 进行相关操作

4. 语言模型：

- (1) RNN: 搭建语言模型并使用训练集语料进行训练，得到一个词向量编码器。可以使用 NLP 封装库，如 `Torchtext`、`Gensim` 等，也可以自己编写语言模型。
 - (2) BERT: 搭建 BERT 模型并加载大语料库上预训练的模型参数（与 RNN 不同，**无须手动训练**）。推荐的预训练参数来源为 `huggingface` (https://huggingface.co/docs/transformers/model_doc/bert)。
5. **情感分类：**基于训练好的语言模型（固定参数），编写一个情感分类模型，包含一个 RNN/BERT 模型和一个分类器（如 MLP）。首先，将一个句子中的每个单词对应的词向量输入 RNN/BERT，得到句子的向量表征。然后将句向量

作为分类器的输入,输出二元分类预测,同样进行 loss 计算和反向梯度传播训练,这里的 loss 选用**交叉熵损失 (CrossEntropyLoss)**。

提示:此次实验中不能固定 BERT 模型的参数,而须随分类器一同训练以达到微调的目的。

6. 测试性能: 选择你认为最合适的(例如,在验证集上表现最好的)一组超参数,重新训练 RNN&BERT 模型,并在**测试集上测试(注意,此处应是你的实验中唯一一次在测试集上的测试)**,评价指标选用准确率(Accuracy),记录测试的结果(Loss&Acc)。**并将两个模型实验结果进行对比分析,写在实验报告中。**

提示:为便于直观比较,建议将验证集 loss 和 acc 变化的曲线利用 **matplotlib** 绘图进行可视化。

实验提交

实验四截止时间:6月29日23:59:59,线下完成代码检查(关键代码讲解+运行展示+结果展示),并需在 bb 系统提交源代码及实验报告,具体要求如下:

1. 全部文件打包在一个压缩包内,压缩包命名为:学号-姓名-ex4.zip
2. 代码仅包含.py文件,请勿包含实验中间结果(例如中间保存的数据集等);如果有多个代码文件,放在src/文件夹内
3. 实验报告提交为.pdf格式,包含学号、姓名,内容包括简要的实验过程、关键代码展示、**对两个模型的对比分析**以及测试集上的实验结果。