

## 实验要求：

使用 pytorch 或者 tensorflow 的相关神经网络库，编写 RNN 的语言模型，并基于训练好的词向量，编写 RNN 模型用于文本分类。

具体来说，在本次实验中，需要通过 RNN 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)。

## 实验步骤

1. **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。
2. **数据准备**：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：

```
├── test
├── train
├── imdb.vocab
├── imdbEr.txt
└── README
```

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。

3. **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。
4. **语言模型**：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。
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 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，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 是分类 loss，如交叉熵 loss。
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，重新训练模型，并在测试集上测试（注意，这理应是你的实验中**唯一**一次在测试集上的测试），并记录测试的结果。我们并不强制要求使用验证集，但如果你要使用验证集，必须从原始数据的训练集中划分而不是测试集，并且在实验报告中要描述你的具体设置。**本次实验使用准确率（ACC）作为评价指标。**

## 实验提交

本次实验截止日期为 **12 月 12 日 23:59:59**，需在 bb 系统提交代码源文件及实验报告，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-作业 2.zip
2. 代码仅包含.py 文件，请勿包含数据集和实验中间结果（例如中间保存的模型等），如果有多个文件，放在 src/文件夹内
3. 实验报告要求 pdf 格式，要求包含姓名、学号，内容包括简要的**实验过程**和关键代码展示，对数据集的**处理设置**超参数的**实验分析**以及测试集上的**实验结果**。