设计文档

> 附件说明:
>
> - model.py为模型主体，含：模型定义、模型训练过程；
> - uitls.py为训练模型的过程中可能用到的辅助工具类 API，如“加载训练集”、“打印进度”等；
> - rnn_256_512_2.rar为模型权重参数。

一、Requirements

- Python: 3.7.3
- PyTorch: 1.4.0
- NumPy: 1.17.3
- matplotlib: 3.1.1

二、算法设计思路

使用神经网络模型，由于赛题为“变长文本序列相似/不相似的分类”问题——

1. 首先将词向量序列做 embedding 映射为嵌入向量；
2. 对嵌入的变长文本使用循环神经网络 RNN（LSTM）进行特征提取，保留最后一刻的隐状态作为 RNN 提取的最终特征，即将文本映射到同一语义空间；
3. 对于分别映射到 RNN 输出特征空间的 text_a 和 text_b，将二者合并为单一张量，使得 text_a 和 text_b 分别为该张量的 2 个通道，即 text_a 和 text_b 分别为下一层网络输入的两个 feature map。
4. 将合并后的输入依次经过 3 层 CNN layer，进行信息融合、细粒度特征匹配；
5. 将上述 CNN 的输出输入到全连接的 Linear layer 分类器，输出最终的后验概率预测值，即 text_a 和 text_b 的相似概率 $P(y=1|x)$；
6. 设定分类阈值，输出分类结果。

三、 `model.py` API 说明

`class RNN(nn.Module)`：神经网络层定义，其中 `forward(*args)` 方法为前向计算函数。

`prepare_data()`：准备好训练集和测试集（线下）数据待用。其通过将 *train.txt* 按照一定比例 `rate`（训练数据占总样本的比例）分割成训练集 `train_set` 和测试集 `eval_set`，返回待用。若 `sample=True`，则

还将从 *corpus.txt* 采样部分数据并添加 0 标签作为不相似样本，以缓解赛题训练数据集中的正负样本不平衡问题。

`eval_rnn()`：评价当前模型状态在测试集上的分类效果，打印测试集损失值及分类准确率。若准确率达到一定阈值，还将保存模型参数。

`train_rnn()`：进行模型训练。

四、 `utils.py` API 说明

`WORDCOUNT`：51158，词表的词汇数量

`POSITIVE`：143229，*train.txt* 中正例的数量

`NEGATIVE`：104771，*train.txt* 中负例的数量

`load_data()`：加载并提取数据集 *train/test/corpus.txt* 为所需格式。

`sample_corpus()`：从 *corpus.txt* 中采样部分数据（真正使用时的调用过程被集成在 `model.py` 的 `prepare_data()` 接口中）。

`data_set()`：将上述加载的数据封装为 TensorDataset 类型，方便 PyTorch 训练时的数据调用。

`predict()`：输入训练好的模型，对 *test.txt* 中的无标签数据进行预测。

`plotACC()`：对训练时准确率 ACC 的变化过程进行绘图。

`class Timer`：计时器，负责记录/打印训练耗时。

`class ThresholdTester`：二分类预测时，$[0,1]$ 阈值划分的测试器，测试模型在不同阈值划分下（如 0.5, 0.6, 0.7, 0.8, 0.9）的准确率。