电影评论情感分类

杨安东

目录

摘要

本实验的目的是在提供的电影评论数据集上进行情感识别，情感分为两类，正面和负面。按照实验说明文档提示使用 Text_CNN 网络来实现电影评论情感分类。但在搭建好网络后遇到了准确率无法提升，句子长度确定困难的等问题，无法获得理想结果。在修改了网络结构，不断尝试超参后，最终获得了满足要求的结果，接下来将对实验过程与遇到的问题进行介绍。本实验主要使用 pytorch 框架完成。

1 问题描述

根据输入的一句影评来判断为正面评价还是负面评价。输入数据已经经过分词处理，不再需要分词，但是需要自行建立词到编号的字典。同时实验数据集也包含了一个预训练的词向量。

1. 输入数据例子：如果 我 无聊 时 网上 乱逛 偶尔 看到 这部 电影 我 可能 会 给 它 打 四星 但是 TNND 姐是 花 大洋 去 电影院 看 姐在 电影院 睡着 姐 现在 非常 心疼 电影票 钱 一部 无聊 至极 电影
2. 输出标签：0-正面，1-负面

2 解决方案

按照实验说明的描述，直接使用 Text_CNN 来完成本实验。

2.1 网络结构设计

网络结构使用典型的 text_cnn，其结构如图 1所示。网络最初的部分是一个 Embedding 层，负责将词语转换为固定长度的词向量，这是一个可以训练的网络。

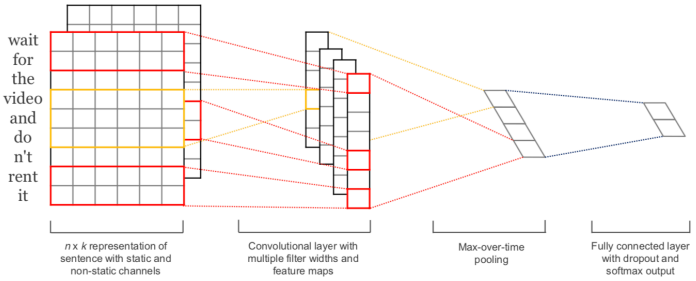

图1 Text_CNN 网络结构

2.2 损失函数设计

损失函数的选择没有太多考虑，使用较为常用的交叉熵损失函数。

2.3 超参设计

在数据集中最长的文本具有 690 多个字符，如果统一使用常用的 75，大部分句子都会被截断，可能会丢失过多的信息，在经过反复测试后，选取了 666 作为文本最大长度。最终使用的其他超参如下：

1. lr=0.001
2. batch_size=50
3. epoch=3

4. embedding_dim=50
5. kernel_num=256
6. dropout=0.5

其中学习率使用了 pytorch 提供的 lr_schedule 函数，其可以根据 epoch 动态调节学习率。在本文中，其定义如下：lr_scheduler = torch.optim.lr_scheduler.StepLR(optimizer, step_size=10, gamma=0.8)。

3 实验分析

3.1 数据预处理

数据的处理方法主要是提取出词到编号的字典。数据集组成部分有：

1. 训练集: 包含 2W 条左右中文电影评论，其中正负向评论各 1W 条左右.
2. 验证集: 包含 6K 条左右中文电影评论，其中正负向评论各 3K 条左右。
3. 测试集: 包含 360 条左右中文电影评论，其中正负向评论各 180 条左右.
4. 预训练词向量: 中文维基百科词向量 word2vec。

由于数据只有标签与已经分词的语句，并没有处理为词语对应的编号。因此需要自己编写数据集读取与编号过程，处理 word2id。

pytorch 自定义数据读取主要使用 torch.utils.data.DataLoader 函数，其需要 Dataset 类型作为参数。torch.utils.data.Dataset 是抽象类，可以通过继承 Dataset 类并重写 ___len___ 与 ___getitem___ 方法实现自定义的数据读取。在本实验中即为读取一条语句，遇到超过规定句子长度的文本就截断，遇到长度不足规定句子长度的就使用在 word2id 数据中定义好的‘_PAD_’ 字符。之后按照 word2id 字典将词语转换为编号进行返回。

3.2 实验结果与分析

训练过程中损失函数变化如图 2所示。

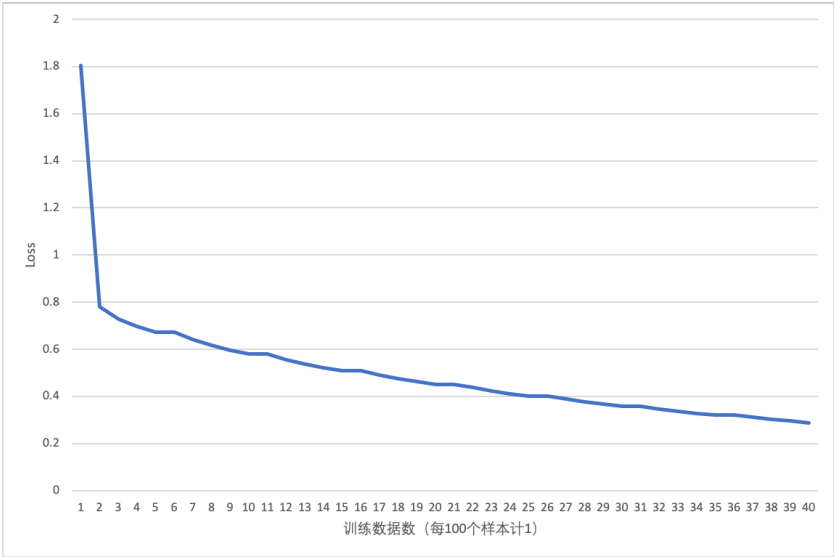

图2 Loss 随着训练过程的变化

训练进行 3 个 epoch，就可以获得 84% 的平均准确率，其他统计数据如召回率，F1，精度等如图 3所示。

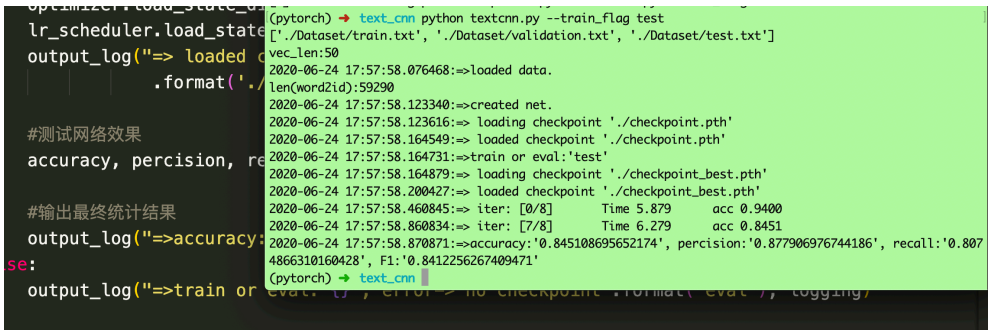

图3 实验在测试集上的结果