**基于酒店评价的文本情绪分类**

1. 选题原因

本身关于python和机器学习的知识有限，自然语言处理知识的了解大多来源于课上的理论以及代码实践的学习，情绪分类是我之前很感兴趣的研究方向，但目前拟定研究方向与情绪分类关系不大。本科毕业设计中采取U-Net实现CT图像的自动分割，由于U-Net是由CNN逐渐发展来的，较为熟悉，因此选取TextCNN来进行文本情绪分类。

1. 数据集说明

由于本人电脑无法进行GPU加速，故在选择数据集时排除了较为复杂或者较为庞大的数据集。实践中使用的是一个酒店评价的数据集，是从网上搜索到的情绪分类数据集汇总中的一个,共有7766\*2条数据，有label和review两个属性，review即顾客对酒店的具体评价内容，label即代表对应评论内容的情绪分类标签，分为0和1，0表示消极情绪，1表示积极情绪，并且前5322条数据均为积极情绪标签及评论内容，之后的2444条数据则为消极情绪标签及评论内容。

数据集链接：

https://raw.githubusercontent.com/SophonPlus/ChineseNlpCorpus/master/datasets/waimai\_10k/waimai\_10k.csv

1. 实践过程

## 3.1实践环境

操作系统:macOS Monterey 12

处理器：2.3 GHz 四核Intel Core i5

编程语言:python

编程软件：Pycharm

## 3.2数据划分

首先借用csv包内的方法读取初始数据集train.csv,遍历数据集中标签label列，统计出积极情绪评论个数并得出消极情绪评论，由于积极性评论集中在数据集前面部分，消极评论集在数据集后面部分，将数据集按照大概4:1的比例划分，以积极与消极的分界线为界，向前取4800，向后取1999共5999条数据为训练集，剩余1767条数据划分为测试集数据并存储至对应的train或test文件夹中。

## 3.3分词

使用jieba分词以及哈工大停用词表hit\_stopwords（删去极个别特殊符号）分别对训练集和测试集中的评论内容进行分词，将分词结果及对应的标签分别存入train或test文件夹中。

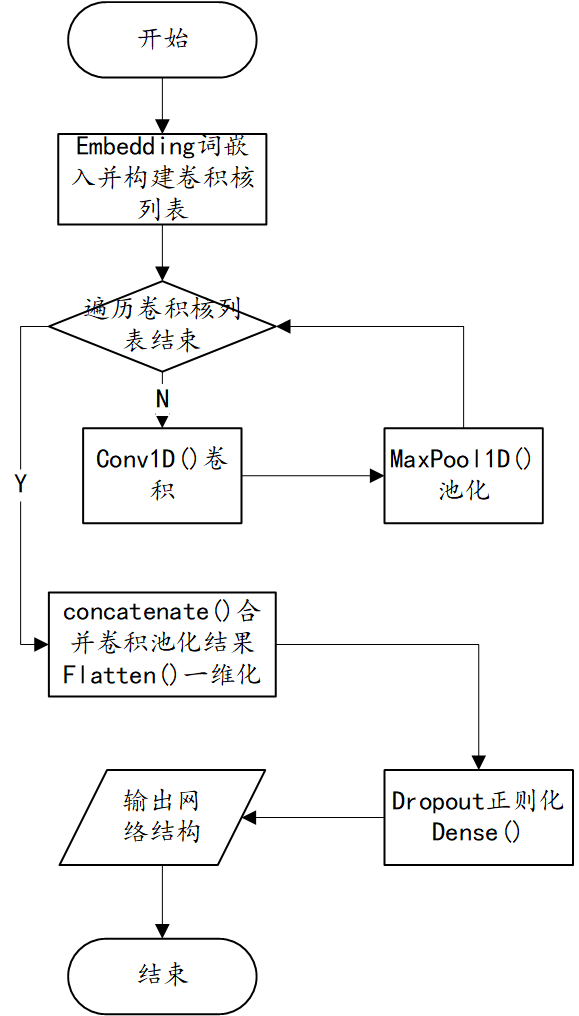
## 3.4编码

编码主要分为两部分：标签编码和分词结果编码。

首先是标签编码。第一步读取分词结果以及准确的标签数据转换成连续的数值型变量，第二步利用LabelEncoder()中的fit\_transform方法将数据集的标签数据编码，第三步借助OneHotEncoder()中的fit\_transform方法将数值型标签转换成独热编码，由于只划分为积极和消极两种标签，编码结果即[0,1]或[1,0]。

然后是分词编码。先是统计训练集中出现次数最多的分词，统计数目可自行设置，保存训练好的数据文件，每个词都有对应的编码。由于训练集和测试集中的评论内容长短不一致，后续训练与测试时输入向量的规格不同造成程序运行失败。考虑到文本长度不同，方便后续模型训练，将文本编码后的数字序列设置为统一长度，此实践中设置为500。将以上所有编码后的结果保存到train或test文件夹中。

3.5 TextCNN模型



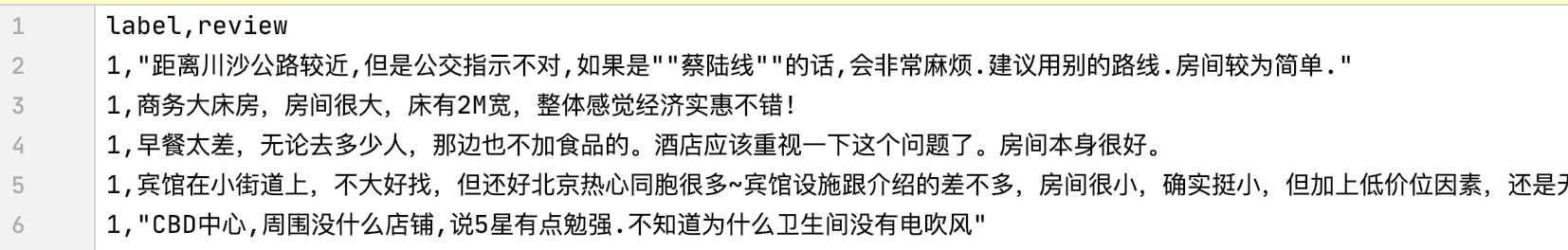
卷积神经网络在cv图像处理上有很好的效果，但是早在2014年就有人提出了TextCNN模型应用于文本分类领域内，与CNN相比较，网络结构没什么修改，主要不同就在模型的输入不同，文本分类的输入数据是一维的，而图像输入数据为二维的。

TextCNN主要分为嵌入层、卷积层、池化层和全连接层。三层卷积和池化，卷积核仅在一个方向上的移动，利用concatenate()将三次卷积结果合并，Flatten将输入一维化过渡到全连接层，Drouput层的作用是防止数据过拟合，最后Dense层设置输出结果维度以及激活函数并得出结果，summary查看网络结构。选择Adam优化器，二元交叉熵函数，评估函数选择二分类准确率。

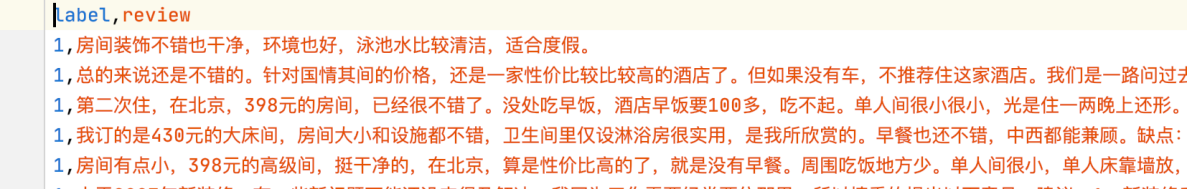
1. 实验结果

## 4.1数据集划分

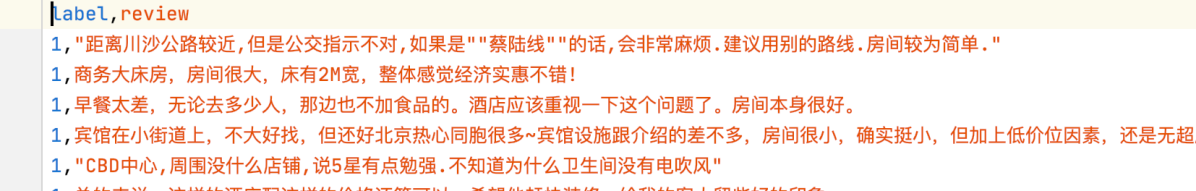
原始数据



训练集数据

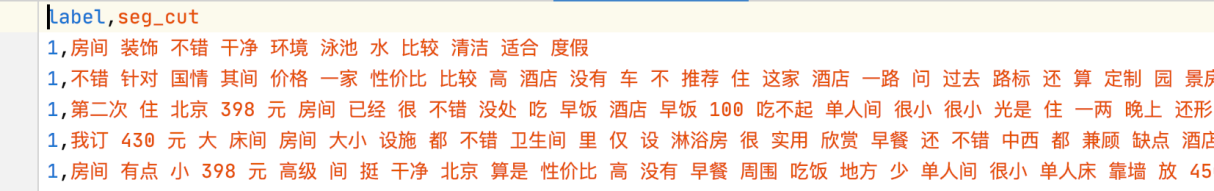


测试集数据

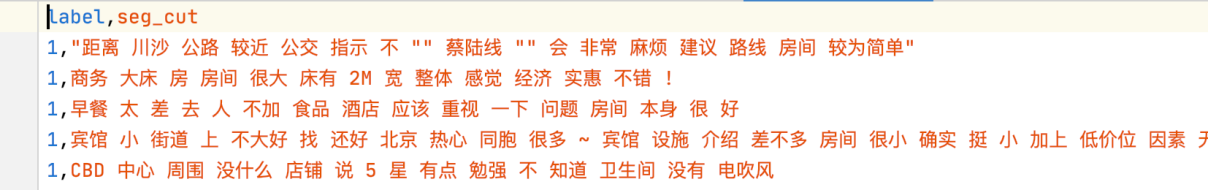


## 5.2分词结果

训练集

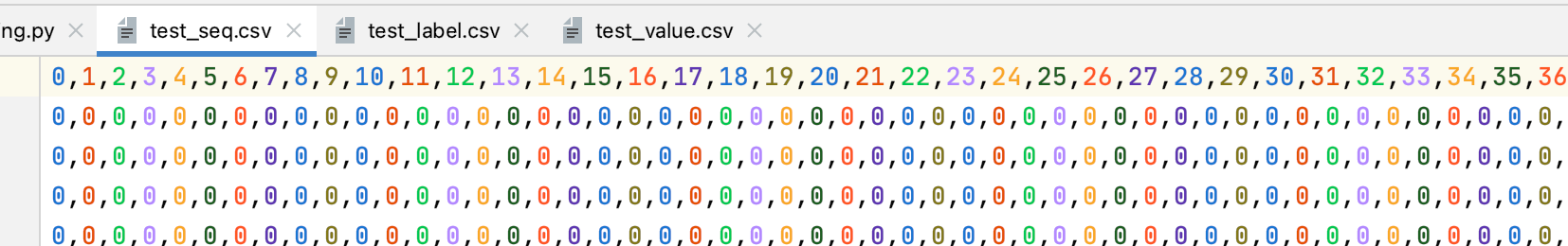


测试集

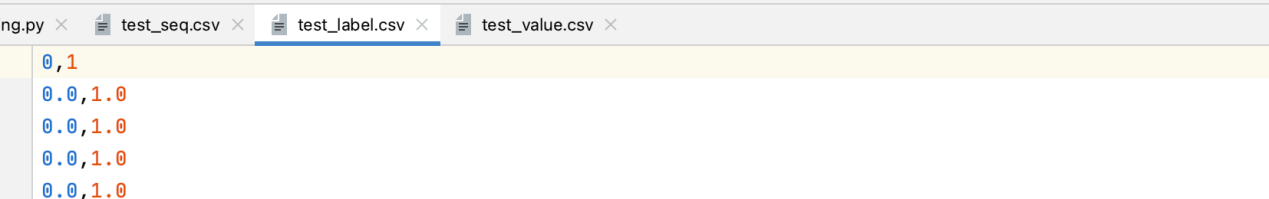


5.3编码（以测试集为例）

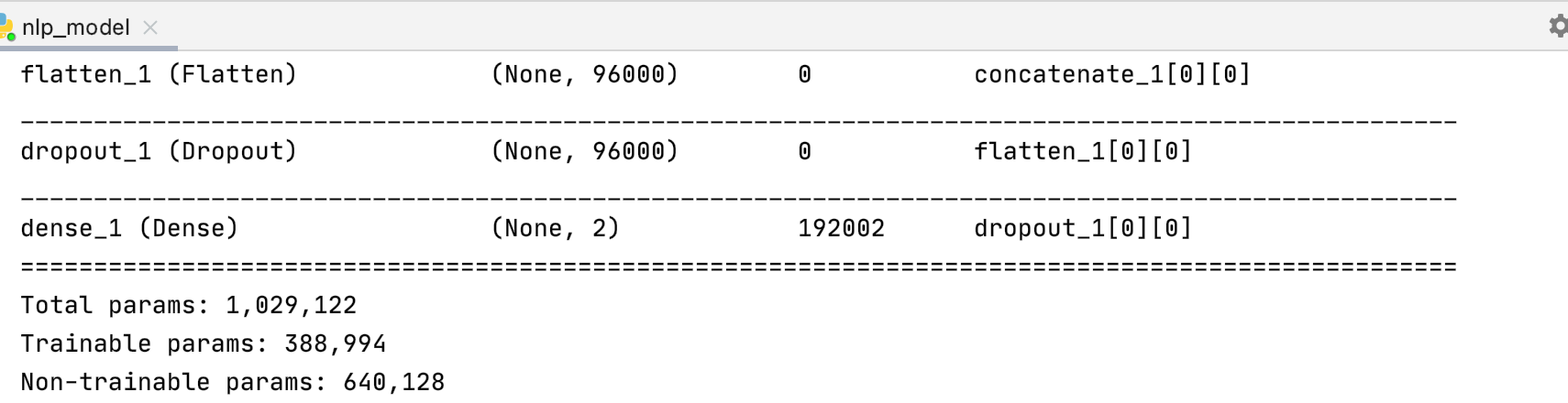
测试集分词编码

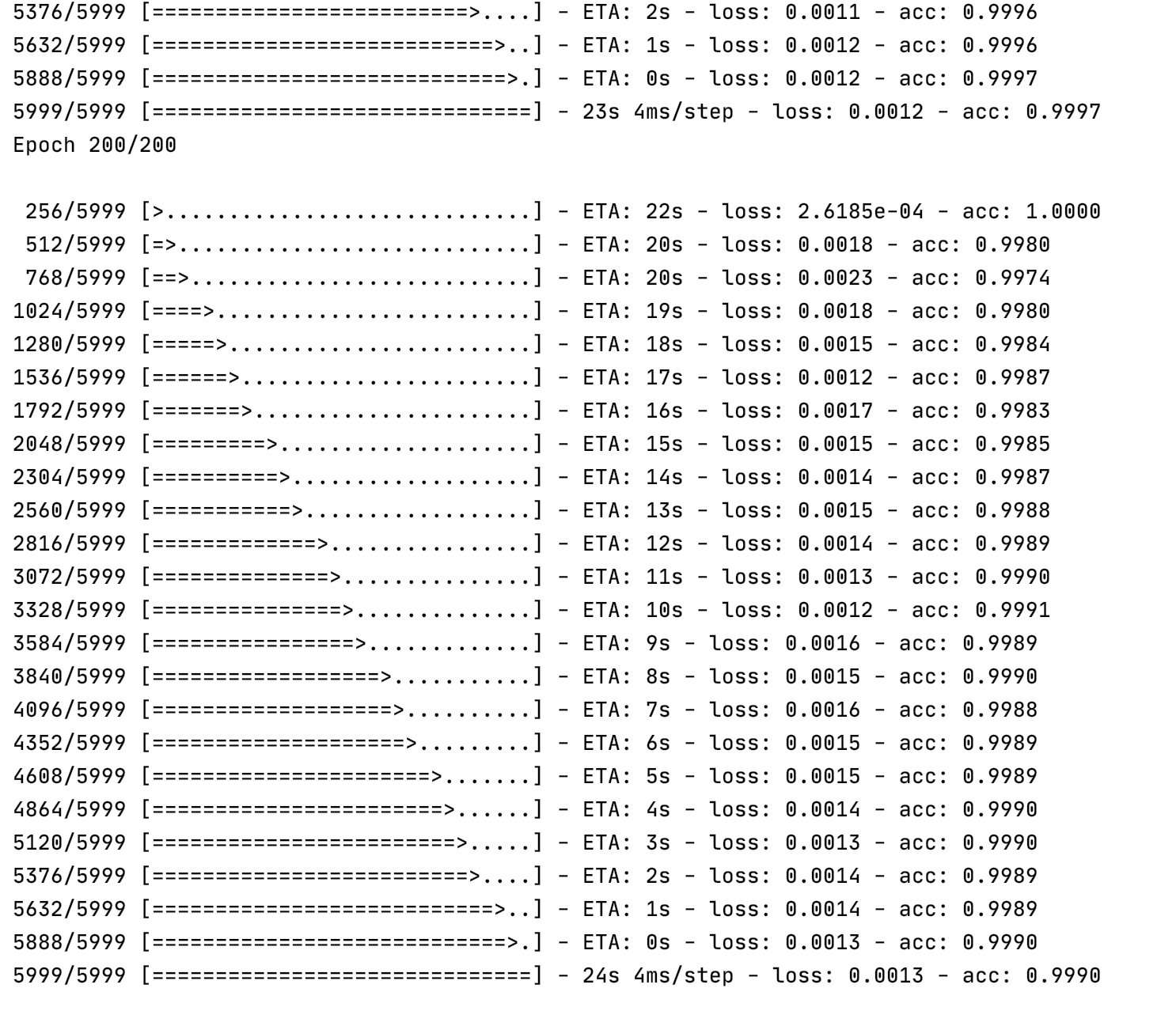


测试集标签编码

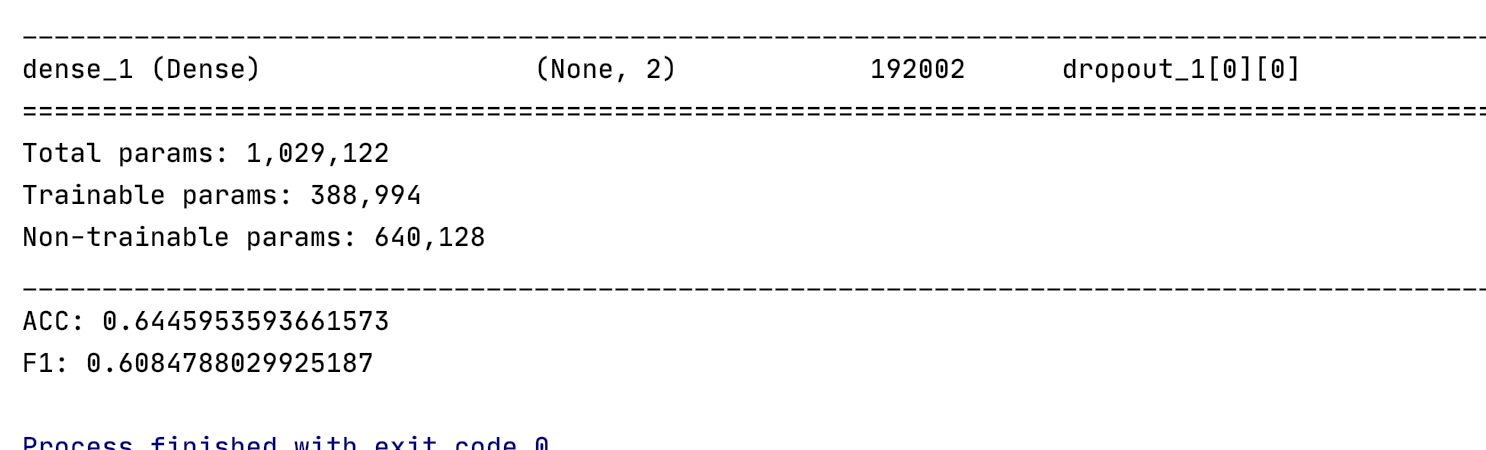


5.4 训练





5.5 测试



1. 遇见的问题

在实践过程中，使用anaconda安装tensorflow时选择较低版本的，2.X版本的有些方法在构建模型时有些无法使用，h5py版本过高导致测试时在加载训练结果text\_Cnn\_train.h5文件时会报错AttributeError，对出现问题的库包降低版本以解决问题。

刚开始忘记把训练集和测试集编码设置相同的长度，仅以训练集的编码维数来构建模型，由于训练集和测试集长度不同，因此不设置相同编码长度时，在使用模型训练结果预测标签时会出现问题，后设置相同长度后问题得以解决。

编码后由于训练集和测试集转码后的csv文件少一行类别，总比设定的训练和测试项目数量各少1，结果发现是在保存分词编码结果以及标签编码结果时少一个类别行，后来在编码保存的代码部分都插入一行类别行，无实际意义，不影响后续训练与测试。

1. 改进方向

通过实践我对自然语言处理的基本流程有了大致了解，显然从最后的评价指标数值来看，TextCNN情绪分类的能力一般，以后可以了解更多高级的模型，bilstm-attention、transformer、bert等，或者也可以将TextCNN与其他模型结合，去解决更复杂的情绪分类问题或者自然语言处理其他分支的任务。