Python期末实验报告

实验题目：基于WeFEND数据集的虚假新闻检测

学号：2012174

姓名：林语盈

学院：计算机学院

专业：计算机科学与技术

目录

[一、 实验名称 4](#_Toc21617)

[二、 实验目的 4](#_Toc29828)

[问题描述 4](#_Toc25376)

[实验目的 4](#_Toc30784)

[三、数据集说明 4](#_Toc28089)

[四、 方法介绍与分析 4](#_Toc10161)

[（一）代表性结果 4](#_Toc16335)

[（二）问题分析 6](#_Toc9684)

[五、技术细节与部分代码 7](#_Toc10070)

[（一）以预训练word2vec+cnn为例 7](#_Toc3088)

[1. 获取数据 7](#_Toc22100)

[2. 获取预训练word2vec 7](#_Toc12704)

[3. jieba分词与词向量构建 7](#_Toc18207)

[4. 模型输入构建 8](#_Toc20330)

[5. 模型构建 8](#_Toc24419)

[6. 模型训练 9](#_Toc27480)

[7. 模型测试与评估 9](#_Toc12426)

[（二） 其它部分代码 9](#_Toc4015)

[1、文件操作 9](#_Toc26688)

[2、将Ofiicial Account Name转成数字特征 9](#_Toc13822)

[3、jieba+tfidf生成词向量 10](#_Toc23072)

[4. 机器学习模型构建 10](#_Toc19094)

[5. rnn相关模型构建（以crnn举例） 10](#_Toc5076)

[六、运行截图 10](#_Toc5289)

[七、 总结与展望 11](#_Toc14126)

1. 实验名称

基于WeFEND数据集的虚假新闻检测

1. 实验目的

问题描述

定义：给定一个信息的标题、出处、相关链接以及相关评论，尝试判别信息真伪。

• 输入：信息来源、标题、相关超链接、评论

• 输出：真伪标签（0: 消息为真，1: 消息为假）

实验目的

测试与分析基于WeFEND数据集的多种方法的结果差异

探究普适的虚假新闻检测模型与方法

三、数据集说明

数据集名称：WeFEND

数据集介绍：包括来自微信公众号的文章标题、文章URL、图片URL、评论、真假标签

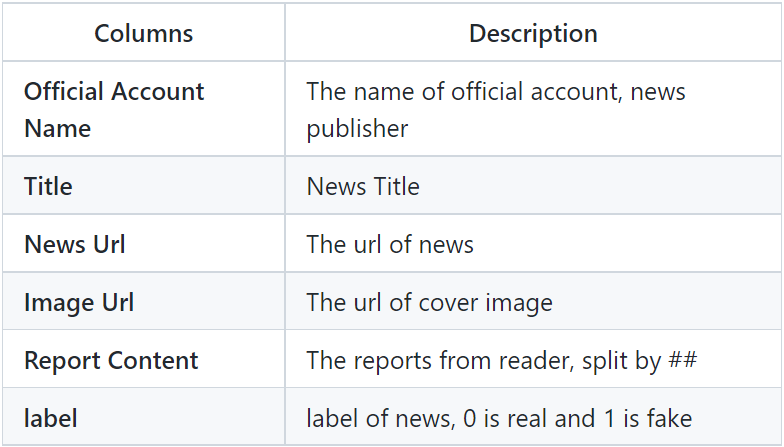
数据集大小：共20728条带标签数据

数据集目标值分布：

假新闻有63条

真新闻有16,503条

数据集成分：



例：环球人物,中国反腐风刮到阿根廷，这个美到让人瘫痪的女总统，因为8个本子摊上大事了, http://mp.weixin.qq.com/s?\_\_biz=MTAzNDI4MDc2MQ==&mid=2651677896&idx=1&sn=87f17336a5aad5eacf12dc1edfc1e7de&chksm=0e63ec9e39146588ba8187a5a45d7ae1aa9b4f4c47c06f9b5f23250937a214f5c9961a838691#rd,http://mmbiz.qpic.cn/mmbiz\_jpg/hpcO6kWnPm6cX3MhPyCmgCMpvJ175oDIIQQ9I3wRkRvTnvuOBwz5ZzbZGpYyyyGun4BoAeXrLL9J9RLiaxkibxng/0?wx\_fmt=jpeg,内容不符,0

1. 方法介绍与分析

（一）代表性结果

|  |  |  |  |
| --- | --- | --- | --- |
| 代表模型 | Accuracy(%） | 参数数量 | 使用数据类别 |
| tfidf整句+传统机器学习方法(朴素贝叶斯) | 87 |  | Title |
| jieba分词+tfidf+传统机器学习方法（SVM） | 97 |  | Title |
| jieba分词+tfidf+传统机器学习方法（SVM） | 88 |  | Title+Ofiicial Account Name |
| Word2vec+传统机器学习方法（SVM） | 96 |  | Title |
| 预训练word2vec+全连接神经网络（结构见下表） | 96.6 | 2,269,922 | Title |
| 预训练word2vec+cnn（结构见下表） | **97.44** | 222,146 | Title |
| 预训练word2vec+Rnns（BiLSTM）（结构见下表） | 97.05 | 86,322 | Title |
| 预训练word2vec+CRNN（结构见下表） | 97.17 | 1,068,546 | Title |

代表模型结构细节：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型 | 预训练word2vec+cnn | 预训练word2vec+Rnns（BiLSTM） | 预训练word2vec+CRNN | 预训练word2vec+全连接 |
| 模型细节 |  |  |  |  |
| 准确度 |  | 97.05 | 97.19 |  |

机器学习模型准确度细节：

|  |  |  |
| --- | --- | --- |
| tfidf整句+传统机器学习方法(朴素贝叶斯) | jieba分词+tfidf+SVM Title | jieba分词+tfidf+线性回归 |
|  |  |  |
| jieba分词+tfidf+传统机器学习方法（SVM）Title+Ofiicial Account Name | Jieba+Word2vec+Svm | Jieba+Word2vec+线性回归 |
|  |  |  |

（二）问题分析

1. **首先使用基于词频的tfidf进行特征提取，将整个Title直接作为输入，使用基本机器学习方法进行训练**。

预期：根据问题，若假设词频特征足以区分真假新闻，朴素贝叶斯算法的独立性假设在词频特征中基本可以成立，而词频特征对应的多维空间也许存在某种线性划分依据，故个人认为使用朴素贝叶斯或SVM也许能有更好的结果，而决策树、随机森林等算法对于类别很少、维度过高且没有显著含义的数据应该效果不佳。

结果：实验中，使用sklearn库，共测试了SVM、朴素贝叶斯、线性回归、决策树、随机森林、K近邻等不到10种算法，结果是正确性相差不大，均在90%以下。

分析：由此结果得知，应当是因为以整句提取的词频特征不足以很好的表示和区分数据。

1. **其他特征的使用，使用jieba分词，仍使用tfidf进行文本特征提取，使用基本机器学习方法进行训练。**

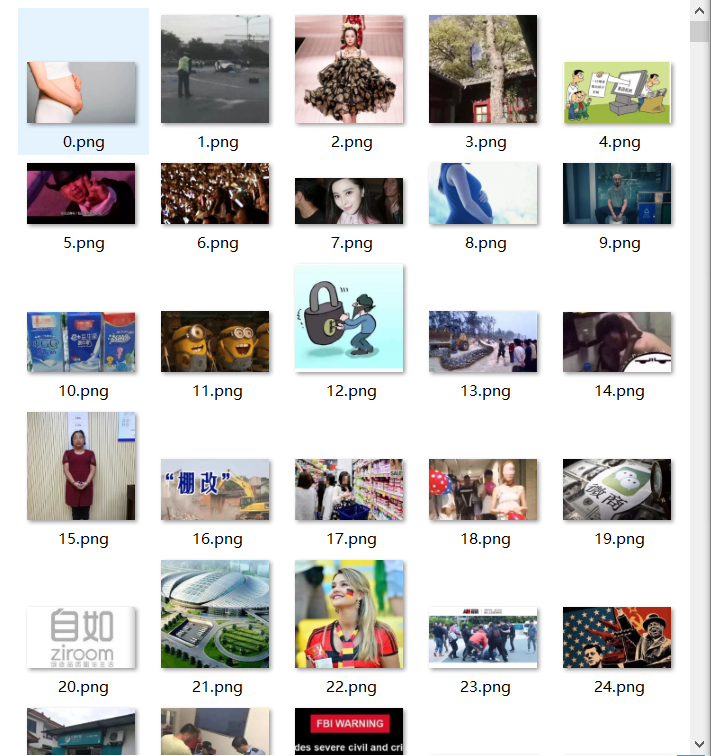
预期：使用jieba分词后机器学习算法的准确度出现差异，与上一预期相同。而使用其他输入，结果应有所提高。

结果：经测试，不同机器学习算法的准确度出现差异，其中，SVM已经达到97%，朴素贝叶斯达到94%，线性回归90%，k近邻83%，决策树运行时间过长没有得到结果。使用了文章来源，将其对应成编号，作为额外输出进行测试，发现测试结果普遍下降。

分析：这与预期结果类似，也说明的分词后，仅用词频已经可以较好的表示和区分数据。令人惊讶的是SVM的结果竟然非常高，达到了97%，于是，重新划分数据集，进行再次验证，发现这不是偶然现象，SVM的结果稳定于97%，这只能说明该数据集的特征十分明显，也就是说，仅通过标题用词的频度就可以分辨整篇文章的真假，这时尚未考虑标题用词的含义与标题用词的顺序特征，更未考虑文章内容以及作者等特征。也由此可知，接下来的模型准确度仅是大致参考，该数据集上的准确度也许并不能适用于普适的虚假新闻检测。

另外，使用了文章来源，将其对应成编号，作为额外输出进行测试，发现测试结果普遍下降，这个结果出乎意料，也许可能是文章来源比较凌乱难以形成有意义的特征或者是直接对应编号增加维度的表示方法反而对原来的词频特征产生了影响，这里有待进一步探讨。

此外，我通过图片URL抓取了全部的图片，但经过观察，我认为对新闻真假情况贡献可能小于文本内容，也因时间所限，未进行测试。



最后，关于评论部分，发现无论真假新闻，评论均为负面，故亦未采用。

而接下来的实验中，仅使用Title文本作为特征。

1. **使用word2vec进行文本特征提取，使用基本机器学习方法进行训练。**

预期：使用与训练词向量，以增加词义的表示，应在准确度上有所提高。

结果：考虑到自己训练word2vec较难获得足够且有效的训练数据，故直接使用预训练好的词向量。预训练的词向量来自于Chinese-Word-Vectors（https://github.com/Embedding/Chinese-Word-VectorsShen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, Xiaoyong Du, [Analogical Reasoning on Chinese Morphological and Semantic Relations](http://aclweb.org/anthology/P18-2023), ACL 2018.）根据微信文章数据集的特点，选择使用搜狗新闻训练的词 + 字 + N元组的300维词向量。分词后得到对应的词向量，将其压平成2d使用SVM分类，准确度为96%。

分析：与直接采用tfidf相比有所下降，这可能是因为将词向量构建的问题，一是构建的特征向量可能比较稀疏，二是若有预训练词向量中未出现的词则将跳过，这丢失了部分信息。

1. **使用word2vec进行文本特征提取，使用全连接网络进行训练。**

预期**：**数据集如果特征明显，在运算条件允许的情况下全连接网络应该可以达到还不错的效果。

结果：直接使用3层的全连接网络，准确率为96.6%

分析：这符合预期，如果数据集数量更多，它可能会有更好的表现。可以看到，仅用三层，网络中的参数已经达到2,269,922之多，但因其可以被并行化计算，训练的代价是可以接受的，但因数据量所限，全连接网络可以获得的信息也很有限，相比于其代价，它的学习效率是不高的，这也告诉我们其他深度学习架构出现的必要性。

1. **使用word2vec进行文本特征提取，使用CNN进行训练。**

预期：应当有更小的训练代价、更好的结果。

结果：这是本次实验中得到的最好结果，准确度达到97.44%并稳定在97%以上。

分析：其输入是word2vec后已经embedding的数据，共有三个卷积层三个池化层，一个dropout层以防止过拟合，与最后的二分类全连接层。其参数数量为222,146，相比仅有3层的全连接网络，参数仅为十分之一，同时可并行，训练速度很快。这种方式考虑了一定的前后单词组合的特征，通过word2vec加入了标题的含义特征，也取得了较好的结果。

1. **使用word2vec进行文本特征提取，使用RNNs进行训练。**

预期：增加考虑了文本前后次序的特征，应该有更好的结果，同时其不可并行导致其训练速度应当较慢。

结果：这里，分别测试了标准RNN、LSTM、GRU的单向、双向、1层、2层，显著特点是不可并行、训练速度极慢，以单层双向LSTM举例，其训练参数为86,322，却远远慢于三层全连接网络。其accuracy差别不大，都不是非常理想。

分析：这应该是因为Title长度较短，且词语前后的依赖关系可能对新闻虚假情况贡献较小。

1. **使用word2vec进行文本特征提取，使用CRNN进行训练。**与RNN们类似，训练速度慢且效果不是非常理想。
2. **Transformer的尝试。**

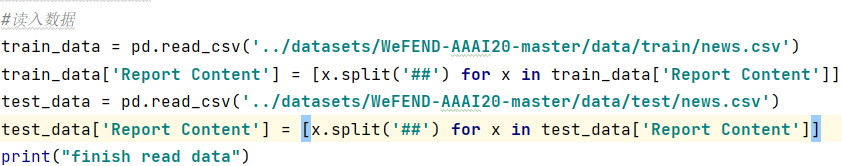
分析：Transformer，如BERT模型，在文本分类任务中取得了很好的效果，根据前面的结果合理推测，即使不加入masking与位置表示，仅用自注意力机制（多头注意力）应当也很适合本次的任务，这有待进一步测试。

五、技术细节与部分代码

（一）以预训练word2vec+cnn为例

1. 获取数据

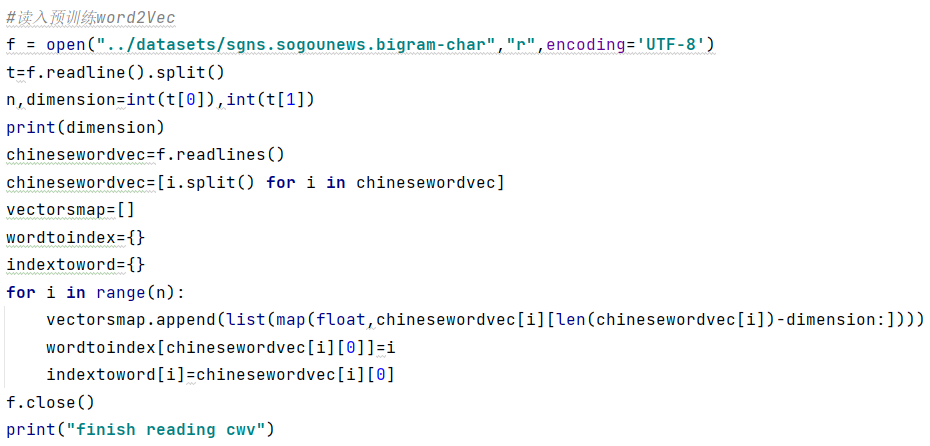
根据已有数据集，利用padas的read\_csv函数读取文本数据，保留需要的部分（这里是Title和label）。



1. 获取预训练word2vec

考虑到自己训练word2vec较难获得足够且有效的训练数据，故直接使用预训练好的词向量。

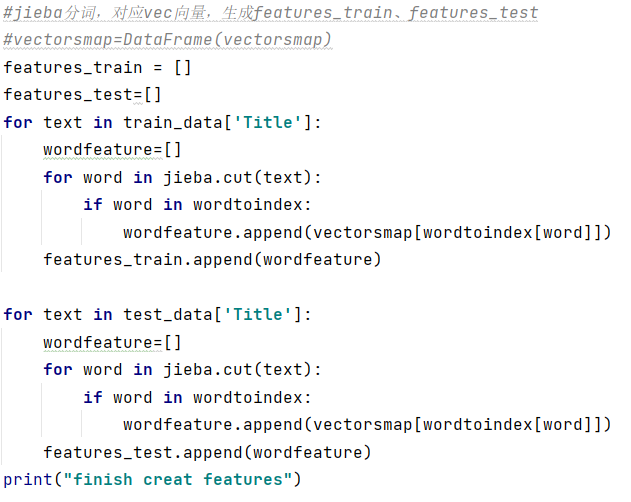
使用搜狗新闻训练的词 + 字 + N元组的300维词向量，并构建其对应词表。



1. jieba分词与词向量构建

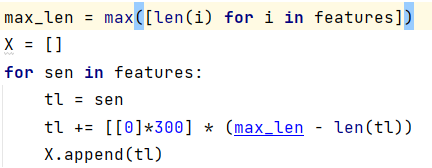
①使用jieba库对中文文本进行分词。

②按分词结果依次找到该词对应的词向量。（这里若找不到则忽略该词）



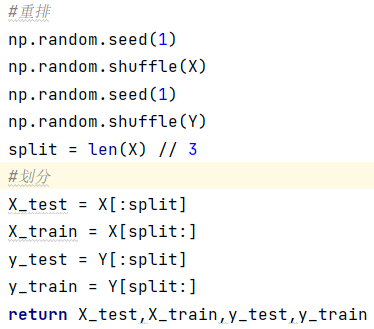
1. 模型输入构建

①数据维度补齐（这里在末尾补0），得到维度为（20728，59，300）的矩阵，其中20728是数据总条数，59是一句话中最大词数，300是一个词的词向量维度。



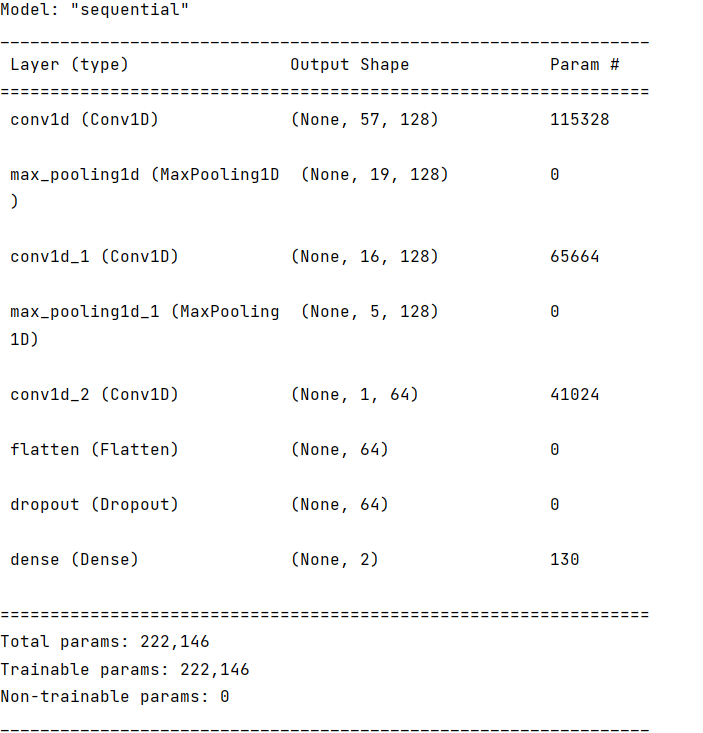
②数据随机重排与划分。

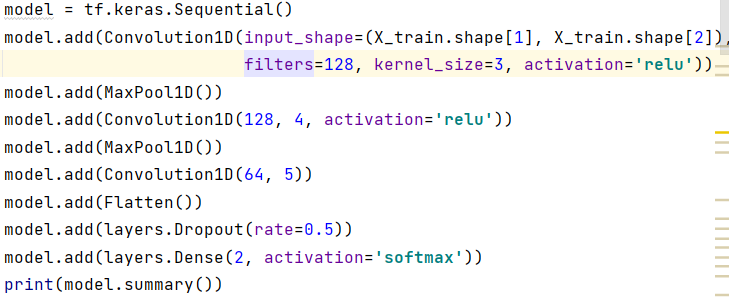
考虑到原划分的不确定性，将train与test合并，重新打乱，再按训练集：测试集=2：1的比例重新划分。



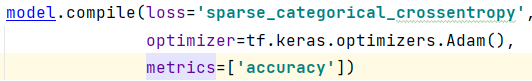
1. 模型构建

使用tf.keras的api进行cnn的构建，其网络结构如下所示。首先的三个卷积核大小为3、4、5的卷积层，各加一个最大池化层，激活函数使用relu；接下来是dropout层以防止过拟合，最后是用于分类的全连接层。

（图3）

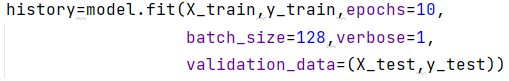


编译时，使用多分类损失函数，优化方法使用Adam，度量y\_和y都是数值给出

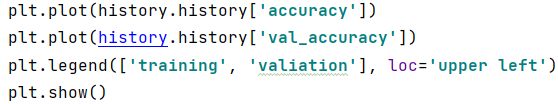


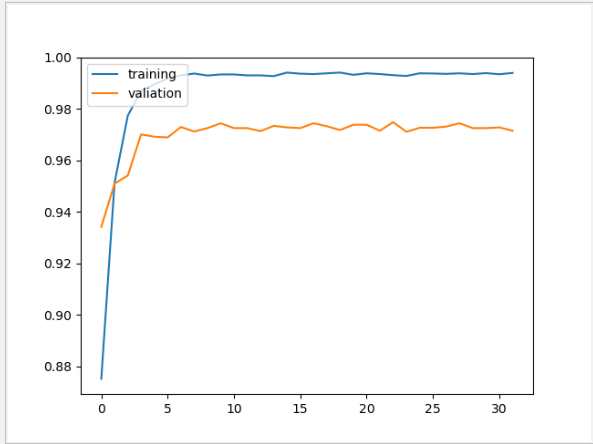
1. 模型训练

经调整，迭代次数epochs取10，batch\_size取128，验证数据采用之前统一划分好的数据，verbose=1表示显示中间过程。



图示化训练和验证准确率变化，试验后发现epochs=10比较合适



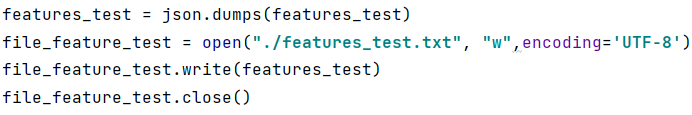


1. 模型测试与评估

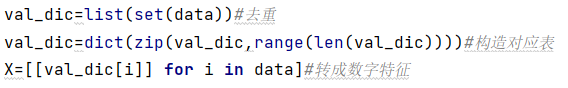
这里在DL方法部分仅采用accuracy进行评估, 每次采用相同的测试数据，减少其他干扰。

1. 其它部分代码

1、文件操作



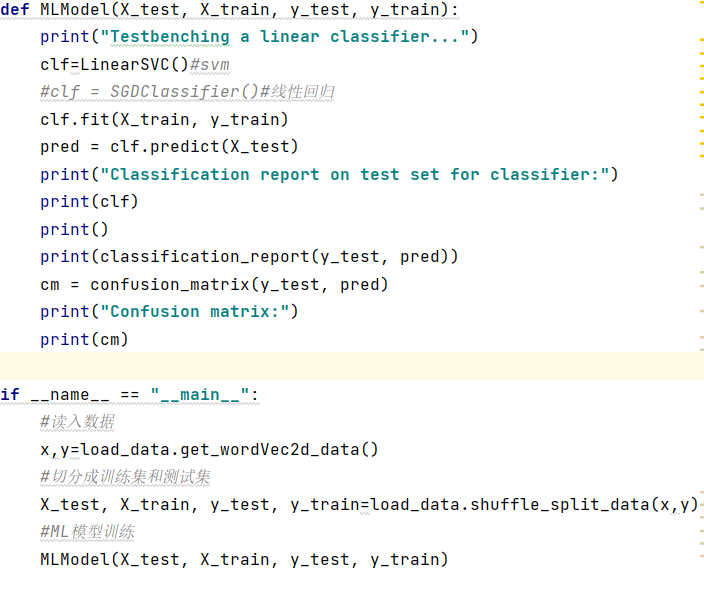
2、将Ofiicial Account Name转成数字特征



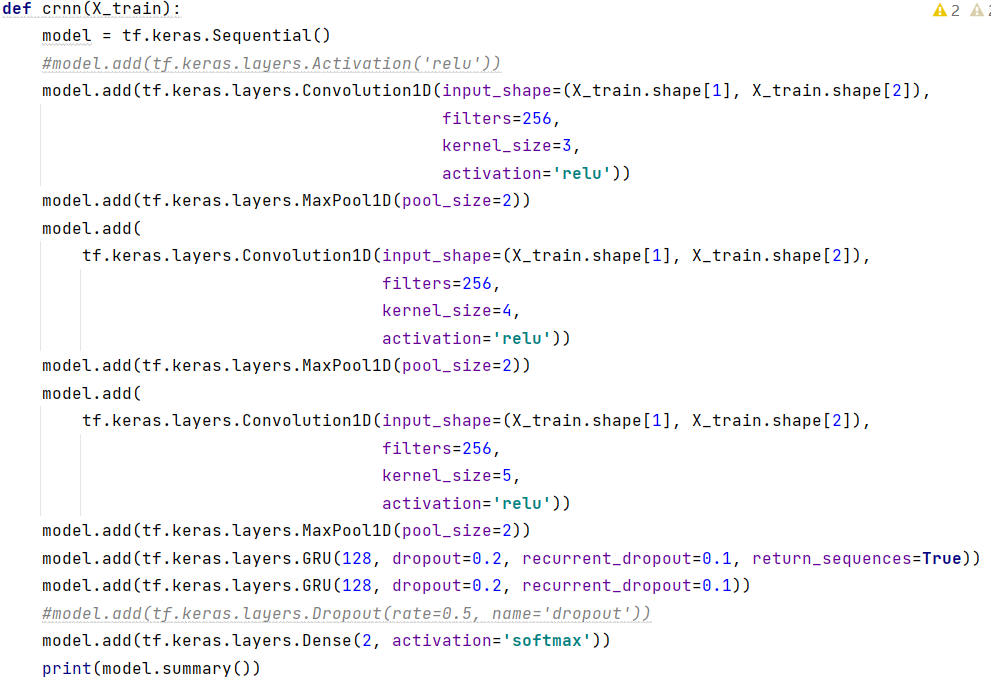
3、jieba+tfidf生成词向量



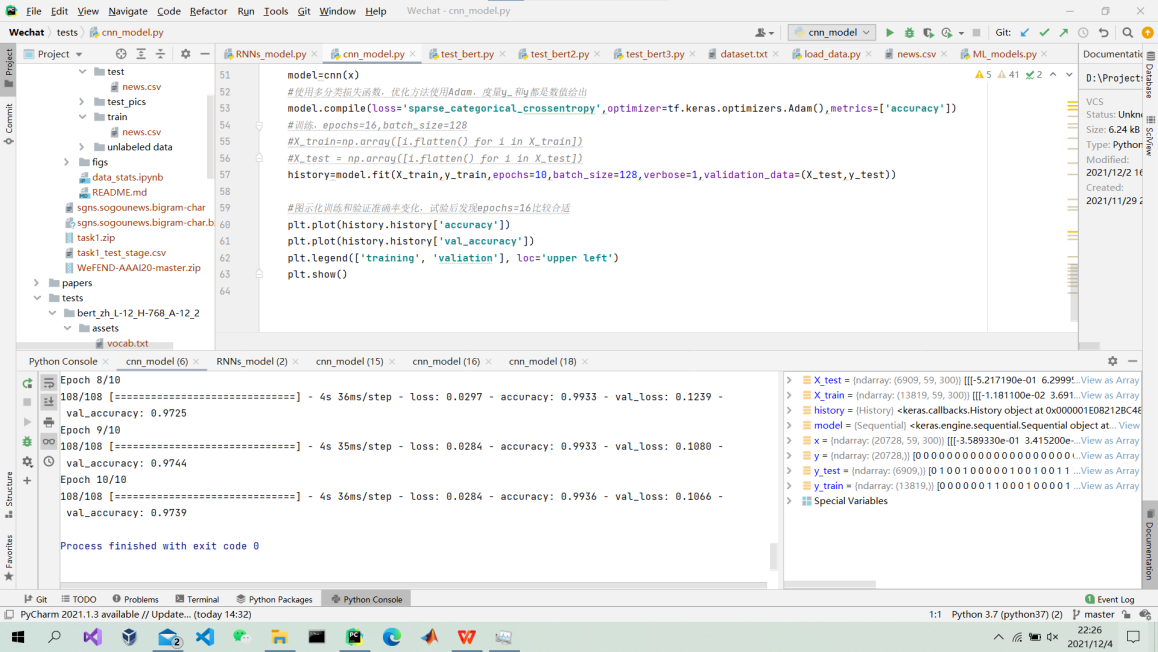
1. 机器学习模型构建



1. rnn相关模型构建（以crnn举例）



六、运行截图



1. 总结与展望

文中在WeFEND数据集中测试了ML、CNN、RNN相关方法，获得了一定结果。

但还有以下方向可以完善：

-Transformer，如BERT模型，在文本分类任务中取得了很好的效果，根据前面的结果合理推测，即使不加入masking与位置表示，仅用自注意力机制（多头自注意力）应当也很适合本次的任务，这有待进一步测试。

-其他特征的合理运用与加入，如评论的情感分析、文章内容文本的加入、图片特征提取内容识别、先验知识的加入等

-测试结果完整性的进一步完善