文档分类项目总结报告

项目概述

背景

- 人工智能是指使用计算机实现与人一样，甚至比人更好的执行某种任务的的技术。人工智能的应用领域广泛。目前主要应用在：金融，安防，电商销售，医疗健康，个人助理，教育和自动驾驶等[1]。
- 机器学习则是实现人工智能的一种方法。机器学习最基本的做法，是使用算法来解析数据、从中学习，然后对真实世界中的事件做出决策和预测。分为4类：监督学习，无监督学习，半监督学习和增加学习。
- 自然语言处理（Natural Language Processing，简称NLP）就是用计算机来处理、理解以及运用人类语言(如中文、英文等)。自然语言处理是计算机科学领域与人工智能领域中的一个重要方向，具体应用实例如手机上的智能语音助理,机器翻译等。现代NLP算法是基于机器学习，特别是统计机器学习。

所以这个项目-文档分类，是属于人工智能领域，具体的研究方向是自然语言处理-因为需要从文本中提取特征，最后使用机器学习算法来进行训练模型与预测新的文档类型。

项目描述

目的就是构建一个机器学习的算法，根据文本内容去预测其所属的类型。这是属于机器学习中的监督学习范畴 。训练的数据来自20 Newsgroups[2]。这收集了差不多20000个文档，20个不同的类型，数据最初最初来自en Lang。现在20 Newsgroup已经变成了机器学习中一个热门的数据，如应用在文本分类上。数据可以通过官网下载，也可以利用Sklearn工具包[3]下载.Sklearn已经帮我们分好了训练集和测试集。
数据利用sklearn工具包[2]下载.sklearn已经帮我们分好了训练集和测试集。可以直接通过调用

```
newsgroup_train = fetch_20newsgroups(subset='train')
```

获取训练集，其中参数subset的值还可以是'test'获取代表测试集，'all'代表获取所有数据。

- 获取数据特征：

```
X_train = newsgroup_train.data
```

- 获取数据标签：

```
y_train = newsgroup_train.target
```

指标

评估指标

评估是通过计算模型正确预测的次数在全部测试样本上的比例，公式如下：

$$\texttt{accuracy}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} 1(\hat{y}_i = y_i)$$

如果值为1表示全部预测正确，值为0表示全部都错。

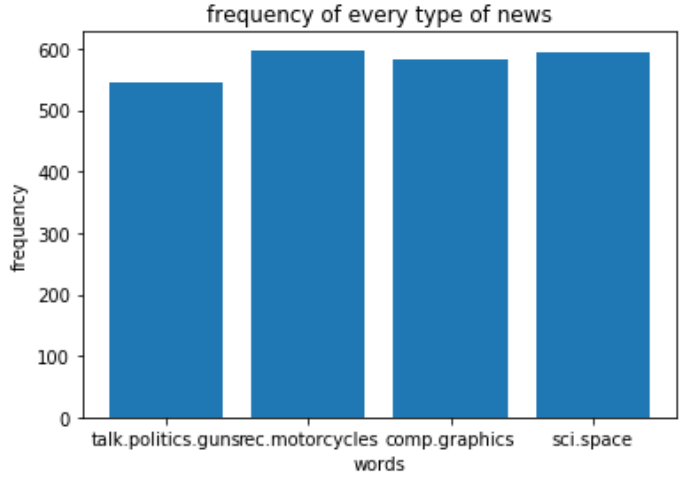
基准模型

我挑选了一个基准模型：在Sklearn中给出了一个对文本分类的示例，精度是0.88我的目标就是高于这个结果。基准模型使用的是MultinomialNB算法，用TF-ID提取F文本特征，使用的是默认参数。

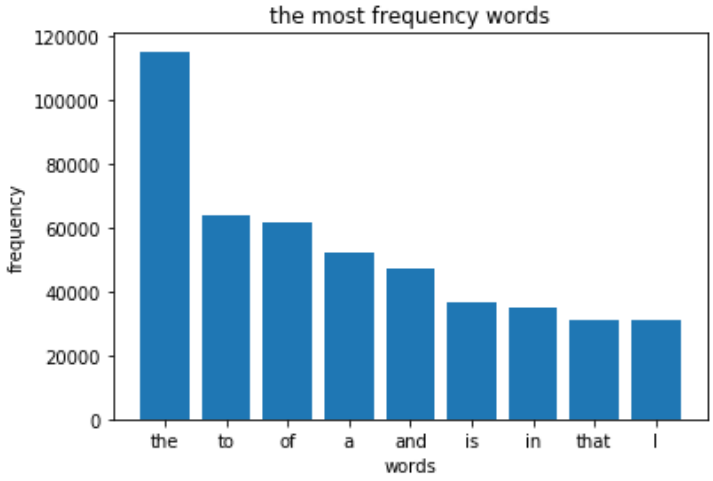
分析

数据探索

我从20个分类中挑选4个作为项目的数据。这四个分类分别是
'talk.politics.guns', 'rec.motorcycles', 'comp.graphics', 'sci.space'。
各个数据训练集的分布情况如下图：



可以看出各个分类训练集分布差不多，根据官网的介绍，数据的训练集和测试集分配是60%和40%。因此测试机的数据分布也是差不多平均的。
下面显示一个具体文本内容:
'Subject: Re: Shaft-drives and Wheelies\nFrom: Stafford@Vax2.Winona.MSUS.Edu (John Stafford)\nDistribution: world\nOrganization: Winona State University\nNntp-Posting-Host: stafford.winona.msus.edu\nLines: 19\n\nIn article 1r16ja$dpa@news.ysu.edu, ak296@yfn.ysu.edu (John R. Daker)\nwrote:\n> \n> \n> In a previous article, xlyx@vax5.cit.cornell.edu () says:\n> \n> Mike Terry asks:\n> \n> >Is it possible to do a "wheelie" on a motorcycle with shaft-drive?\n> >\n> No Mike. It is imposible due to the shaft effect. The centripital effects\n> of the rotating shaft counteract any tendency for the front wheel to lift\n> off the ground.\n\n\tThis is true as evinced by the popularity of shaft-drive drag bikes.\n\n==========\nJohn Stafford Minnesota State University @ Winona\n All standard disclaimers apply.\n'
文中有很多与我们预测这篇文章主题无关的单词与符号，如'\n\n'，单词'the'等内容。



从图中可以看出，像定冠词'the',介词'to','of'等没有携带信息的单词数量最大，这也可以理解，但是它们却不携带信息，不利于机器学习算法从数据中学习。

算法分析

这是监督学习问题，算法需要特征(feature)和标签(label)。标签已经给出来了，现在需要考虑的是从文本中提取特征。我使用了词袋模型来表征文本内容。
假定一篇文档中包含的信息，可以只由其中包含的词语来描述，并且与词语在文档中的位置没有关系，这便是词袋模型，英文为bag of words，意为单词的袋子。
其中关于单词的一个统计方法有几种。首先可以考虑直接统计单词的词频。但是这有个问题，有一些单词在所有文档都有出现，甚至次数差不多，这样的单词并不利于区别不同的文档。
所以我考虑使用TF-IDF的方法。TF-IDF是一种统计方法，用以评估一字词对于一个文件集或一个语料库中的其中一份文件的重要程度。字词的重要性随着它在文件中出现的次数成正比增加，但同时会随着它在语料库中出现的频率成反比下降。[4]

机器学习算法的选择：

1. 我首先考虑的是朴素贝叶斯方法。它只需要少量的数据就可以得到一个较好的分类结果，训练速度快，而且经常被用来文本分类。但是它需要假设特征之间是相互独立的。

2. 其次我考虑SVM。SVM在分类问题是表现比较优秀的，特别是每一个类区别明显的时候。SVM的训练和测试时间相对朴素贝叶斯算法长，对缺失数据敏感等。

3. 最后考虑全连接神经网络，深度学习这几年发展比较快，而且可以应用到很多领域，分类预测准确度高。但是神经网络我们无法观测到其学习的过程，给出的输出的结果难以解释等，全连接神经网络训练时间较长，也容易过拟合。

处理过程

数据预处理

我对数据进行了三个地方的处理，调用Sklearn中的TfidfVectorizer[5]实现。使用TfidfVectorizer其中的三个属性：preprocessor，stop_words，min_df进行文本处理：

- 写了一个去掉文本符号，并将单词全部转化为小写的函数preprocessor。

- stop_words代表TF-IDF忽略的词汇。我给它赋值'english'，这表示忽略英文单词中的那些不携带信息的词汇。如在数据探索中，频率超高的'the'等。

- 通过min_df设置词汇频率的最低限度，这可以过滤掉一些低频居有偶然性的词汇。
  最后单词数从38843降低到13315

算法的实现

1.MultinomialNB

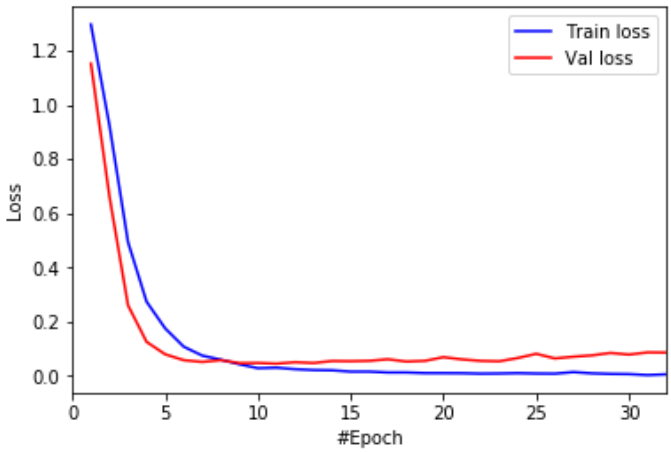
默认参数给出的结果是0.9612。alpha的默认值是1，改为0.2后得到精度0.9644。

2.SVC

默认参数结果是0.2576，我首先考虑改变核函数，默认的核是'rgb',改为''linear'后得到了0.9644的精度。

3.神经网络

我使用Keras来搭建神经网络。Keras是一个高层的神经网络库，使用简单快捷。
设置了迭代次数是800次，优化器是SGD。为了让神经网络在精度不再提高的时候停下来，Keras提供了回调函数EarlyStopping。
最后发现过拟合现象比较严重。迭代100次训练精度到了1，验证精度也不再提高。然后我删除了一层激活层(Activation)还有一层全连接层(Dense)发现提高不多。过拟合问题我考虑使用dropout，或者regularization解决。dropout是指在两个神经网络层数据传输过程中中，随机设置一部分的值为0。而regularization是指在损失函数中加一个额外的值，目的就是削弱权重的影响。实践中发现dropout比较好。经过400多次迭代后收敛，测试集的精度是0.96。
因为SGD训练时间长，容易陷入局部最优解等缺点。我改用了RMSprop的优化方法。最后经过了30多次的迭代就收敛，精度为0.97。效率还是很高的。



总结

三个算法的精度都到了0.96以上，比之前例子(基准)0.88高了0.8，表现还不错。
接下来看看其中MultinomialNB给出的一个结果，对于主题重要意义的单词：
[people，edu，com，gun，guns，stratus，fbi，batf，don，weapons]，可以看出与'talk.politics.guns'有关的主题，算法还是抓取到了关键词汇-如people(人)，gun(枪),weapons(武器)，FBI(美国联邦调查局)

后续的提高考虑

- 从上面的输出可以看出单词的单复数并没有处理如：'gun','guns'等。同理，还可以考虑关于时态等的转化，如'running'，'ran'等统一为'run'等。
- 文本中邮件地址部分的'com'，'edu'并没有去掉，可以考虑设置一个频率上限。
- 前面的讨论都是基于词袋模型进行文本体征提取的，这有一个缺点，就是词与词之间没有联系。因此我们可以使用词向量表征文档,使用一个向量来表征一个词，这会拉近相近词之间的距离。词向量的来源是1986年Hinton的论文[6]。

参考

1. 2016网易科技全球人工智能发展报告
2. 20 newsgroups官网：http://www.qwone.com/~jason/20Newsgroups/
3. sklearn 中20news的地址：http://scikit-learn.org/stable/datasets/twenty_newsgroups.html

4. 来自中文wiki对TF-IDF的介绍：https://zh.wikipedia.org/wiki/Tf-idf

5. sklearn的TfidfVectorizer:http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

6. http://www.cogsci.ucsd.edu/~ajyu/Teaching/Cogs202_sp12/Readings/hinton86.pdf