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

1.目的

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 。

2.数据的准备

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。
准备好数据后，对数据进行探索：包括数据集的分布情况，每篇文档的单词书和词频；然后进行预处理：包括标点符号等。

3.算法的实施

通过实现算法，初步得到精度后，根据结果进行算法的调优，并评判算法的稳健性[4]。

4.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：

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

y表示标签的集合，$\hat{y}$ 表示表示测试结果的集合，n_{sample}表示样本总数
如果值为1表示全部预测正确，值为0表示全部都错。

5.基准模型

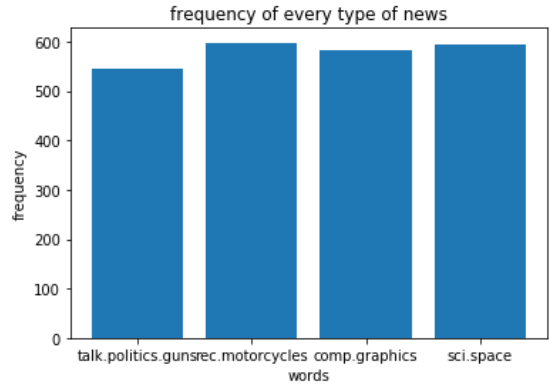
我挑选了一个基准模型：在Sklearn中给出了一个对文本分类的示例，精度是0.88，我的目标就是高于这个结果。
基准模型使用的是MultinomialNB算法，用TF-IDF提取文本特征，使用的是默认参数。

分析

数据探索

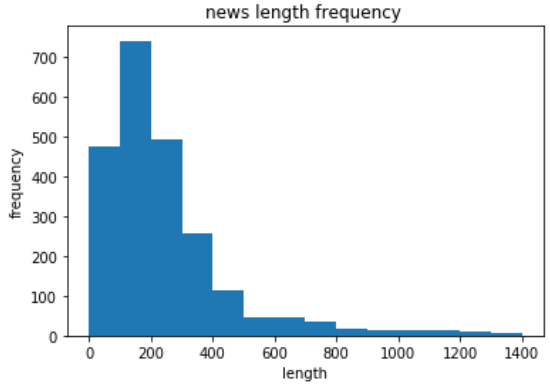
我从20个分类中挑选4个我感兴趣的作为项目的数据，基准模型也是4个分类。这4个分类分别是'talk.politics.guns', 'rec.motorcycles', 'comp.graphics', 'sci.space'。

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20 Newsgroup官网的介绍，数据的训练集和测试集分配是60%和40%。

文档的单词数量分布情况如下(只统计1500个单词一下的文档)：

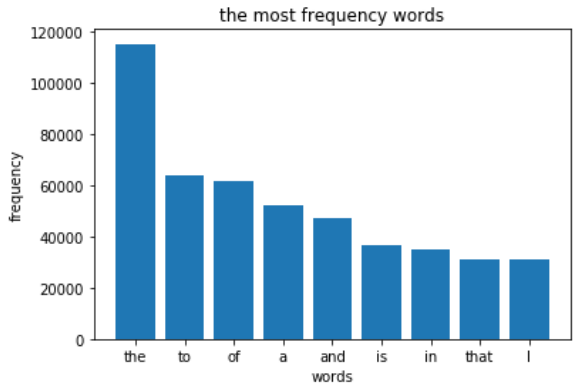


可以看出大部分文档的单词数量集中在500以下。其中单词数最多有9555个，最少的是11个单词。

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

文本的表征

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。

1. 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，用于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[5]。

2. 使用词向量表征文档：用向量来表示词。常见的可以用One-Hot的方法，如[0 0 1 0 ...]，这种表示方法有大量的0，而只有一个1。改进的方法就是稀疏的方式存储：给每一个词对应唯一的数字ID，但是这一种表示方式存在“词汇鸿沟”现象：任意两个词之间都是孤立的。因此为了克服这些缺点，可以使用分布式词向量(Distributed representation)。分布式词向量的来源是1986年Hinton的论文[6]。

机器学习算法的选择

1. 我首先考虑的是朴素贝叶斯分类器。朴素贝叶斯分类器是一系列以假设特征之间强（朴素）独立下运用贝叶斯定理为基础的简单概率分类器[7]。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. 其次考虑卷积神经网络。传统的全连接神经网络，输入层到隐藏层的神经元都是全部连接的，这样做将导致参数量巨大，而卷积神经网络则通过共享参数等方法避免这一困难。在输入是图片或者是序列的情况下能得到较好的结果，但是也有计算量大，要调的参数多等缺点。

简单起见，一种文档表示方法只使用一种机器学习算法：在朴素贝叶斯方法上使用TF-IDF，在卷积神经网络上使用词向量。

处理过程

数据预处理

我对数据进行了三个地方的处理：

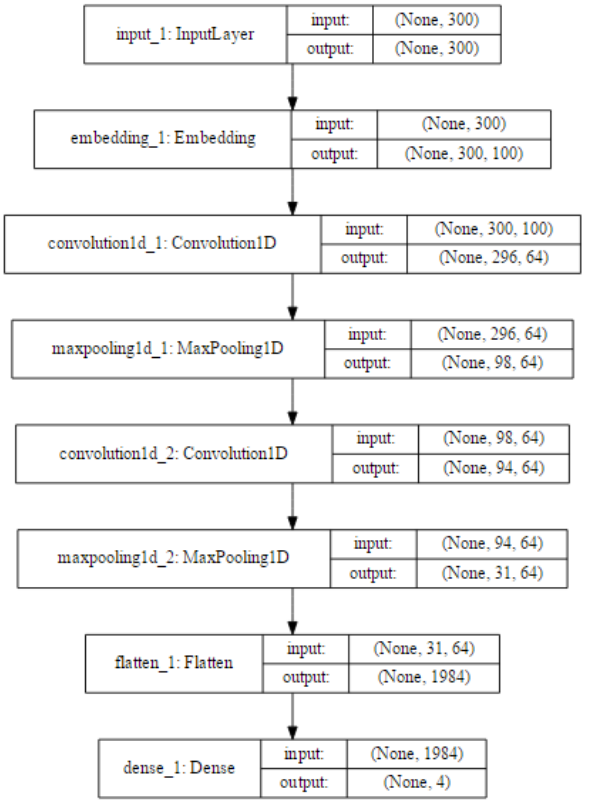
- 去掉文本符号，单词统一为小写。
- 忽略英文单词中的那些不携带信息的词汇。如在数据探索中，频率超高的'the'等。
- 设置词汇频率的最低限度，这可以过滤掉一些低频居有偶然性的词汇。

算法的实现

Sklearn中朴素贝叶斯封装为三个类型，根据文档[7]的描述Multinomial Naive Bayes适合文本分类，通过Sklearn导入了MultinomialNB，默认参数给出的结果是0.9644。

我使用Keras来搭建卷积神经网络。Keras是一个高层的神经网络库，使用简单快捷。
直接使用Keras自带的Embedding层训练词向量。词库的长度通常是10^5~10^6量级，词向量的量级通常是10~10^2。在数据探索中可以看到文本长度在300~500以下的文档占了大部分，所以在Embedding层我设置了最大文本长度是300，词库的最大单词数是10000，词向量的维度是100。

卷积神经网络的输出维度都为64，卷积核的大小为5，步长为1；池化层的大小和步长均为3。模型的结构:Input-Conv1D-Maxpool-Conv1D-Maxpool-Flatten,最后连接一层全连接层用作分类:

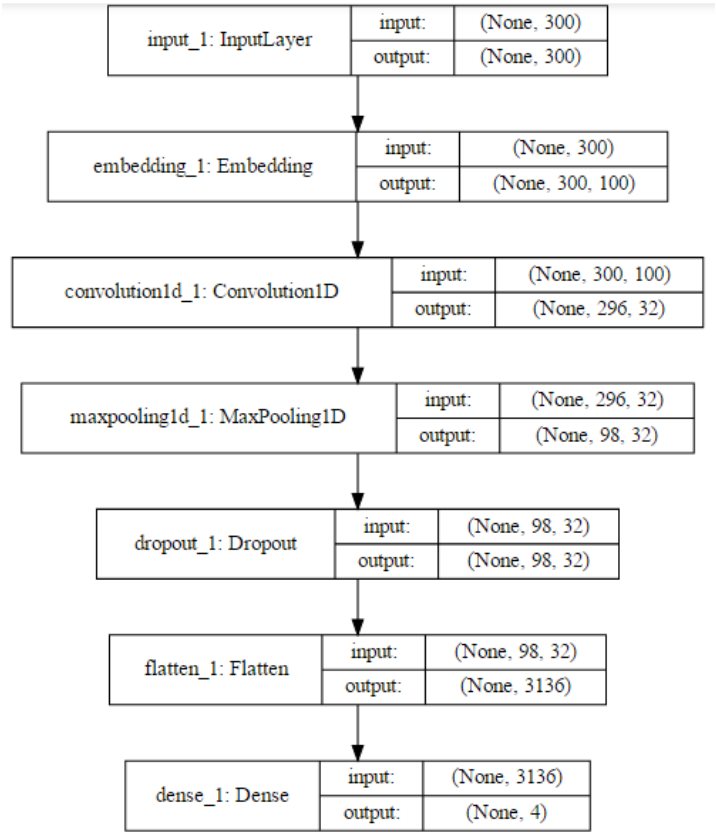


因为SGD训练时间长，容易陷入局部最优解等缺点，我选择RMSprop的优化方法。为了让神经网络在精度不在提高的时候停下来，使用Keras中的early_stop策略。经过3次的迭代，训练精度为1，测试精度为0.72。

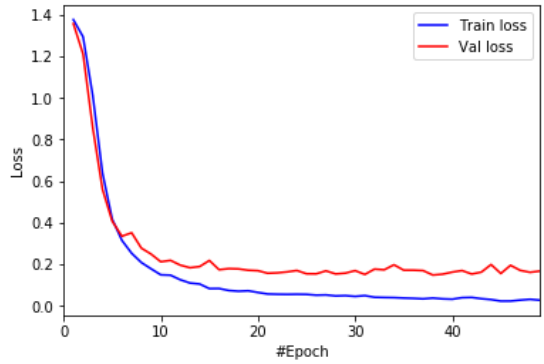
算法的优化

1. 通过使用网格搜索的方法寻找MultinomialNB参数alpha最优值。最后得出的alpha=0.055为最佳，测试精度为0.9625。这比默认的alpha得到的精度低一点。因为网格搜索方法是寻找训练集的最佳参数，可能样本数量不够，结果有一定的偶然性。

2. 卷积神经网络的过拟合比较严重。我删除了前面的一层Conv1D和一层Maxpool，效果不明显，然后通过添加Dropout层，在卷积层使用正则化，降低卷积输出维度等操作,模型结构如下：



发现过拟合现象没有明显改善。最后通过在Embedding添加值为0.5的dropout，迭代49次之后，精度到0.9825。



总结

模型稳健性

1. 我测试了MultinomialNB的不同alpha值

alpha	0.001	0.01	0.1	1	2
测试精度	0.9553	0.9608	0.9618	0.9644	0.9637

看模型的结果发现在alpha从0.001到2这么大的跨度，精度没有很大幅度的改变。

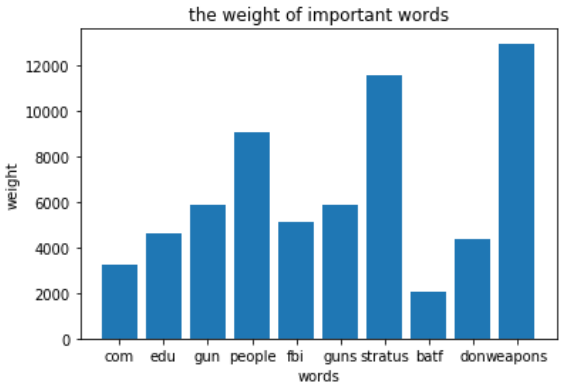
2. 卷积神经网络在不同迭代次数的表现

迭代次数	1	10	20	30	49
训练精度	0.3224	0.9752	0.9876	0.9938	0.0.9941
测试精度	0.3684	0.9567	0.9676	0.9812	0.982

在10次左右已经给出较好的测试精度，到30次以后已经有点过拟合了。

模型结果的表现

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（权重大的）



从上图可以看出，MultinomialNB模型成功抓住了'talk.politics.guns'主题一些关键词如'weapons'（武器），gun(枪)，(fbi)美国联邦调查局等

难点与提升

这个项目难点之一就是特征的提取，用什么方法提取特征，如何对特征进行预处理，这是得到好的分类结果的前提。如从上面的重要单词分布图就能得到两个提升的方向：

- 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统一，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，这也是特征提取可以优化的部分。

结束语

这个项目的目标是文档分类。对数据进行了三个方面的处理：去掉符号，统一单词为小写，去掉不携带信息的单词。文档表征分别用了TF-IDF和分布式词向量。算法选择了朴素贝叶斯分类器，卷积神经网络。对朴素

贝叶斯分类器进行了alpha值的优化，对卷积神经网络做了防止过拟合的处理。
最后结果是：在朴素贝叶斯方法中使用TF-IDF得到了0.96左右的精度；在卷积神经网络在中使用词向量得到了0.98左右的精度。这都高于了基准模型0.88的结果，模型表现还不错，当然也有很多可以提升的地方。

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. wiki稳健性词条:https://zh.wikipedia.org/wiki/%E7%A8%B3%E5%81%A5%E6%80%A7_(%E7%BB%8F%E6%B5%8E%E5%AD%A6)
5. 来自中文wiki对TF-IDF的介绍：https://zh.wikipedia.org/wiki/Tf-idf
6. Hinton的论文 http://www.cogsci.ucsd.edu/~ajyu/Teaching/Cogs202_sp12/Readings/hinton86.pdf
7. wiki朴素贝叶斯分类器词条：https://zh.wikipedia.org/wiki/%E6%9C%B4%E7%B4%A0%E8%B4%9D%E5%8F%B6%E6%96%AF%E5%88%86%E7%B1%BB%E5%99%A8
8. Sklearn中对朴素贝叶斯方法的解释：http://scikit-learn.org/stable/modules/naive_bayes.html