机器学习纳米学位

毕业项目 文档分类

关力宁 2018.05.16

1. 问题概述

1.1 项目概述

自然语言处理（NLP）是机器学习技术重要应用范畴之一，而文档分类应该是自然语言处理中最普遍的一个应用，例如文章自动分类、邮件自动分类、垃圾邮件识别、用户情感分类等等，因此一个好的文档分类程序有非常大的应用意义。

选择这个项目的主要原因是自己曾经参与过一个新闻推送系统，系统中对新闻进行分类的模块当时是由大数据和算法部门的同事负责的，自己那时对这个分类的模块就很好奇，现在自己也可以用学到的知识来实现一个文档分类程序了。

本项目将使用使用经典的 20 类新闻包作为训练和测试数据集，该新闻包大约有 20000 条新闻，比较均衡地分成了 20 类，是比较常用的文本数据之一，可利用 sklearn 包下载。

1.2 问题陈述

本项目要解决的问题是如何利用 20 类新闻包设计好一个文档多分类程序，即经过针对输入的文档，程序可以预测出该文档的类别标签，并且要保证一个较高的预测准确率。

文本分类的基本流程是读取数据、清洗数据、特征提取、模型训练、模型评估，难点主要在特征提取和模型训练环节。我将利用 tf-idf 词袋子模型和 Word2Vec 模型提取文档特征，利用监督学习中决策树、朴素贝叶斯、支持向量机和深度学习中 CNN 四种模型进行分类训练，并以词袋子+决策树作为基准模型比较训练结果。我希望最后能在上述分类方法中找到一个预测准确率达到 80%以上的模型。

1.3 评价指标

20 类新闻包数据集是一个分类平衡的数据集，因此我将采用准确率作为评价指标。。准确率是指模型预测正确的结果所占的比例，对于本项目，计算公式为：

```
Accuracy = total_valid_predicition / total_sample
```

total_valid_predicition 是指分类预测正确文档数之和；total_sample 是指所有样本文档数之和。

2. 分析

2.1 数据的探索

分类文本数据使用经典的 20 类新闻包，利用 *sklearn* 工具包下载，里面大约有 18846 条新闻，比较均衡地分成了 20 类，是比较常用的文本数据之一，此外 sklearn 工具包下载的数据集已经按 60%和 40%的比例分成了训练集和测试集。

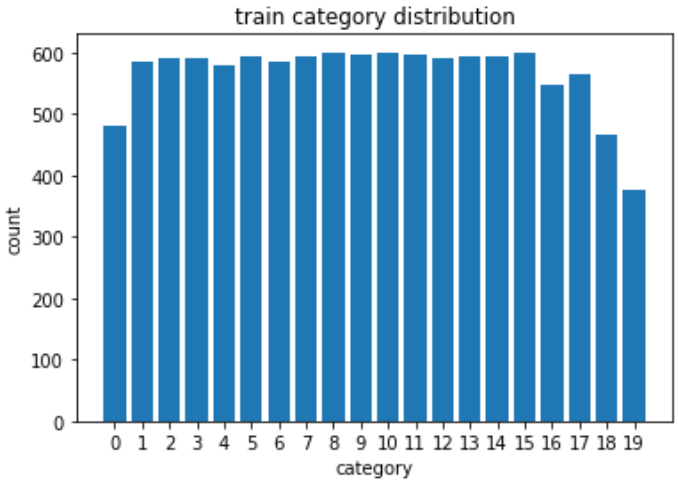
下面具体分析一个新闻文档的结构，该文档是训练集 baseball 分类下 99971 号：

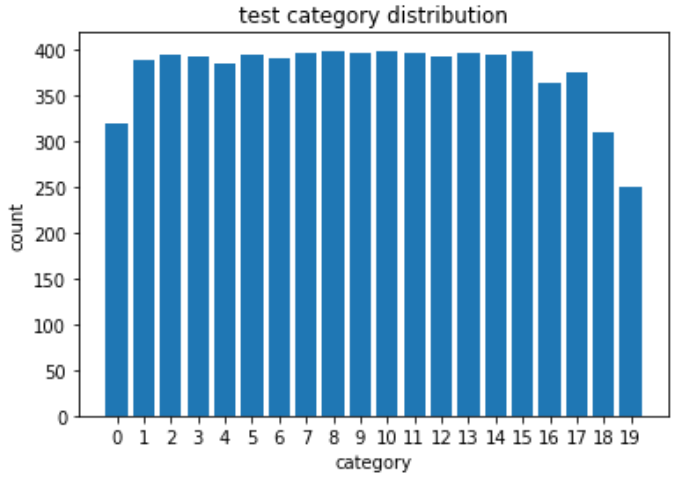
```
From: admiral@jhunix.hcf.jhu.edu (Steve C Liu)
Subject: Baseball Stats
Organization: Homewood Academic Computing, Johns Hopkins University, Baltimore, Md, USA
Lines: 17
Distribution: usa
Expires: 5/5/93
NNTP-Posting-Host: jhunix.hcf.jhu.edu
Summary: 1992 EWB II Stats wanted

        Hello, my friends and I are running the Homewood Fantasy Baseball
League (pure fantasy baseball teams). Unfortunely, we are running the league
using Earl Weaver Baseball II with the Comm. Disk II and we need the stats
for the 1992 season. (Preferably the 1992 Major League Stat Disk) We have
the '92 total stats but EWB2 needs the split stats otherwise we have 200
inning games because the Comm. Disk turns total stats into vs. L's stats
unless you know both right and left -handed stats.

        So, if anyone has the EWB2 '92 Stat Disk please e-mail me!

|Admiral Steve C. Liu        Internet Address: admiral@jhunix.hcf.jhu.edu|
|"Committee for the Liberation and Intergration of Terrifying Organisms  |
|and their Rehabilitation Into Society" from Red Dwarf - "Polymorph"      |
|****The Bangles are the greatest female rock band that ever existed!****|
|   This sig has been brought to you by... Frungy! The Sport of Kings!    |
```

可以看到文档结构主要分为三部分：文档头、文档内容、文档尾，有些文档没有包含尾信息，如 baseball 分类下 102585。文档头信息中有作者、主题、机构等信息，而主题信息和分类相关性十分高，其他信息又与类别标签几乎没有相关性，因此文档头信息需要被移除。同样文档尾信息中包含的邮箱信息和其他信息与新闻正文关系不大，而判断新闻所属分类主要和新闻内容相关，因此文档尾信息也需要被移除。

2.2 探索性可视化

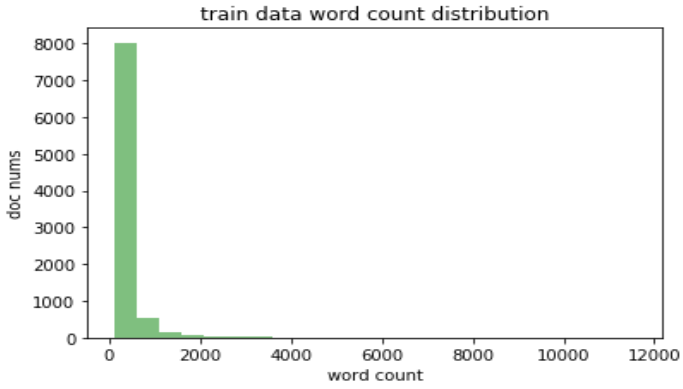
首先，利用可视化探究 20 类新闻包是否是一个分类平衡的数据集，因为这影响我是否可以将准确率作为最终评价指标。

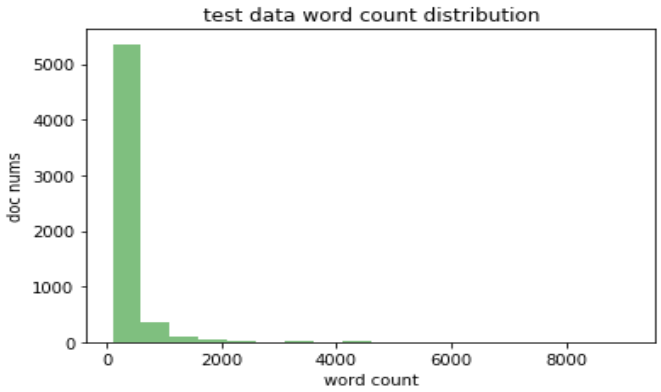




通过可视化，可以确认无论训练集还是测试集都是分类平衡的，因此可以用准确率做评价指标。

其次，因为最终是基于文档中单词进行分类预测的，所以文档中单词数也是一个很重要的因素，利用可视化来探察所有文档的单词数分布是否比较集中。





可以看出，绝大部分文档的单词数都小于 500，一小部分大于 500 小于 1000，极小比例文档单词数大于 1000。

2.3 算法和技术

- 清洗数据

首先我将去掉文档中的 headers、footers、quotes，只保留新闻内容主体，然后去掉内容主题中的标点符号、停用词，最后将单词统一为小写形式。

- 提取文档特征
  - 词袋 tf-idf 模型

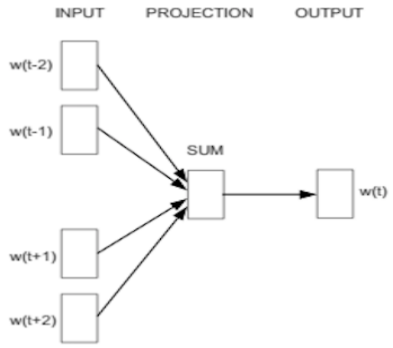
    词袋模型是一种统计某个词在一份文档中出现次数的算法。统计所得的词频数据可以用于比较文档并测量其相似性，具体应用包括搜索、文档分类、主题建模等。词频-逆文档频率（TF-IDF）是另一种根据文章中包含的词来判断文章主题的方法。TF-IDF 为词赋予权重——TF-IDF 测量的是相关性，而非频率。因此，在整个数据集中，词频都会被 TF-IDF 分值所取代。TF-IDF 会测量某一特定文档中的词的出现次数（即词频，term frequency）。出现某一个词的文档数量越多，这个词作为信号的价值就越小。这样做的目的是仅留下独特的高频词用作标记。每个词的 TF-IDF 相关性是一种标准化的数据格式，总和也是 1。

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

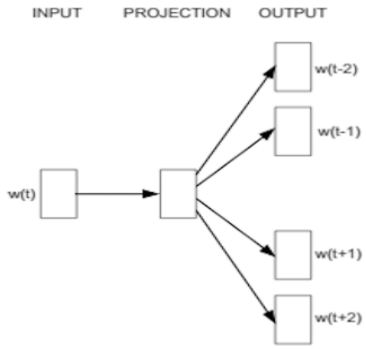
$tf_{i,j}$ = number of occurrences of i in j
df_i = number of documents containing i
N = total number of documents

  - Word2Vec 模型

    Word2Vec 是一个用于处理文本的双层神经网络。它的输入是文本语料，输出则是一组向量：该语料中词语的特征向量。虽然 Word2vec 并不是深度神经网络，但它可以将文本转换为深度神经网络能够理解的数值形式。Word2vec 的目的和功用是在向量空间内将词的向量按相似性进行分组。它能够识别出数学上的相似性。Word2Vec 神经网络的输出是一个词汇表，其中每个词都有一个对应的向量，可以将这些向量输入深度学习网络，也可以只是通过查询这些向量来识别词之间的关系。Word2Vec 的方式有两种，一种是用上下文预测目标词（连续词袋法，简称 CBOW），另一种则是用一个词来预测一段目标上下文，称为 skip-gram 方法，我们使用 skip-gram 方法。



CBOW



Skip-gram

- 分类模型
  - 决策树

    决策树是一种预测模型，代表的是一种对象特征属性与对象目标值之间的一种映射关系。决策树分为分类树和回归树两种，分类树对离散变量做决策，输出是样本的预测类别；回归树对连续变量做决策，输出是一个实数 ，本项目应采用分类树。构造决策树的关键是如何确定树中每个节点，这里主要依赖信息熵和信息增益理论，具体实现算法有 ID3、C4.5、CART，scikit-learn 使用 CART 算法的优化版本。
  - 朴素贝叶斯

    朴素贝叶斯分类是一种十分简单的分类算法，它的思想基础是这样的：对于给出的待分类项，求解在此项出现的条件下各个类别出现的概率，哪个最大，就认为此待分类项属于哪个类别。贝叶斯定理如下：

    $$P(B|A) = \frac{P(A|B)P(B)}{P(A)}$$

    朴素贝叶斯分类器，计算每个特征属性划分对每个类别的条件概率：

    $$P(y_i|x) = \frac{P(x|y_i)P(y_i)}{P(x)}$$

    $$P(x|y_i)P(y_i) = P(a_1|y_i)P(a_2|y_i)...P(a_m|y_i)P(y_i) = P(y_i)\prod_{j=1}^{m} P(a_j|y_i)$$
  - 支持向量机

    支持向量机（support vector machines，SVM）是一种分类模型，它的基本模型是定义在特征空间上的间隔最大的线性分类器，间隔最大使它有别于感知机；支持向量机还包括核技巧，这使它成为实质上的非线性分类器。简单的说就是把数据投影到高维空间进行生维，使得在低纬度无法可分的问题在高维空间变得可分。
  - 卷积神经网络(CNN)

可以避免传统神经网络全连接层导致的参数量巨大的问题，抗噪能力较强，分类准确率高，但调参较难，训练时间稍长。

2.4 基准模型

以词袋 tf-idf 和决策树作为基准模型，决策树易于理解和实现，是分类问题中最常见的模型，基准模型的准确率将在下面对数据进行预处理后首先进行计算。

3. 方法

3.1 数据预处理

通过设置 sklearn 的 fetch_20newsgroups 的 remove 参数来移除文档中的 header、footer；利用正则表达式来移除文档中的数字、标点符号和其他非法字符，正则\W 用于匹配非数字、字母、下划线字符，\d 用于匹配数字字符；利用 nltk 包移除掉文档中的停用词。

文档处理前：

```
From: jzawodn@bgsu.edu (Jeremy D. Zawodny)
Subject: Help needed in setting up NCSA Telnet w/ AppleTalk or Phonenet...
Summary: help me....
Organization: Bowling Green State Univ.
Lines: 21


Okay, I'm trying to install NCSA telnet on a couple (okay, a whole bunch)
of machines.  They're all true blue IBMs with either Fallon Phonenet cards
or Dastar cards.  (I belive those names are correct.)  Well, the docs for
telnet say that it'll run over an AppleTalk driver, but I've had little
success.

If anyone has succesfully installed Telnet w/ AppleTalk, I'd like some
help with the config file for Telnet...

BTW, please reply via E-mail if possible...

Thanks,

Jeremy

--
Jeremy Zawodny | Computer Science Undergrad | Bowling Green State University
--------------------------------------------------------------------------------
jzawodn@andy.bgsu.edu | Student Computer Consultant | *thrilled* OS/2 2.0 user
```

文档处理后：

```
['okay', 'trying', 'install', 'ncsa', 'telnet', 'couple', 'okay', 'whole', 'bunch', 'machines', 'true', 'blue', 'ibm
s', 'either', 'fallon', 'phonenet', 'cards', 'dastar', 'cards', 'belive', 'names', 'correct', 'well', 'docs', 'telne
t', 'say', 'run', 'appletalk', 'driver', 'little', 'success', 'anyone', 'succesfully', 'installed', 'telnet', 'w', 'a
ppletalk', 'like', 'help', 'config', 'file', 'telnet', 'btw', 'please', 'reply', 'via', 'e', 'mail', 'possible', 'tha
nks', 'jeremy', 'jeremy', 'zawodny', 'computer', 'science', 'undergrad', 'bowling', 'green', 'state', 'university']
```

3.2 执行过程

- 获取文档特征。首先我利用 sklearn 的 TfidfVectorizer 来求得单词的 tfidf 权重值。接着我利用 20 类新闻和 text8 作为语料库分别训练出 Word2Vec 模型，对于单词 post，20 类新闻的 word2vec 模型预测出的最相近单词组是

```
[('repost', 0.6051768064498901),
 ('posts', 0.5657081007957458),
 ('postings', 0.5549967288970947),
 ('followup', 0.5498017072677612),
 ('summarize', 0.5389478802680969),
 ('comment', 0.5246976017951965),
 ('posted', 0.5206242203712463),
 ('newsgroup', 0.5181717872619629),
 ('responses', 0.5161654949188232),
 ('topic', 0.5063819885253906)]
```

Text8 的 word2vec 预测出的单词组是

```
[('pre', 0.5336133241653442),
 ('during', 0.5090769529342651),
 ('after', 0.4649081826210022),
 ('audial', 0.4634260535240173),
 ('tolbiac', 0.4483736157417297),
 ('continuation', 0.4422840178012848),
 ('beginning', 0.4368570446968078),
 ('postwar', 0.4366604387760162),
 ('prior', 0.4279540479183197),
 ('dc', 0.4248111248016357)]
```

- 利用 train_test_split 将 tf-idf 表示的训练集切分为训练数据集和验证数据集，验证集比例是 0.1，并且定义了用于评估模型准确率的函数 evaluate_model。
- 利用 sklearn 包，分别用决策树、朴素贝叶斯、支持向量机三种模型进行训练。其中贝叶斯采用 MultinomialNB，原因是词向量特征是离散数据，MultinomialNB 相比于高斯朴素贝叶斯更适合；支持向量机采用 LinearSVC，原因是词向量特征维度很高，线性核函数处理效率较高。测试结果如下：

模型	验证集准确率	测试集准确率
决策树(基准模型)	0.608	0.503
朴素贝叶斯	0.823	0.770
支持向量机	0.884	0.795

- 最后我利用 keras 建立神经网络，并分别用 20 类新闻和 text8 得到的 word2vec 作为输入训练模型。对于 embedding 层，考虑到大多数文档长度在 500 以下，我将文档长度定位 300，词库数量设置为 15000，词向量维度设置为 200。接着利用一层卷积层和池化层来缩小向量长度，再通过 Flatten 层把二维向量转换为 1 维，最后用一层 Dence 将输出收缩到 20 个分类，测试准确率是 0.547 和 0.521。利用 20 类新闻得到的 Word2Vec 要比 text8 的效果更好，在词量上 20news 比 text8 少了 4 万单词，但同一单词的 most_familiar，20news 显然比 text8 更贴近 20 类新闻。

3.3 完善

- 我利用网格搜索法和 k 折交叉法来优化朴素贝叶斯和支持向量机模型，优化后朴素贝叶斯的提升较为明显，在测试集上的准确率提高到了 0.801，支持向量机并没有提升，测试集准确率仍未 0.795。
- 对于神经网络，我在 flattern 层后增加了一层 dropout 进行去拟合，在 dropout 后再增加一层 Dence 层，通过两个全连接层把输出收缩到 20 个分类，优化后准确率有一定的提高 0.652 和 0.581。

4. 结果

4.1 模型的评价与验证

下表展示了各个模型的训练结果以及优化后的训练结果：

	基准模型	朴素贝叶斯	贝叶斯优化后	SVM	优化后 SVM	CNN 20newsgroup	CNN text8
训练集准确率	0.608	0.823	0.876	0.884	0.885	0.966	0.958
测试集准确率	0.503	0.770	0.801	0.795	0.795	0.643	0.578

由上表可知，优化后的朴素贝叶斯模型分类准确率最好，且相比基准模型的分类准确率有很大提高。卷积神经网络在训练集上分类效果很好，但是在测试集上得分不高，存在过拟合现象，可能还需要再增加一些训练层，或需要更多的训练数据，并不断调优参数才能得到较高的分类准确率，但同样这也会大大增加训练时间。

虽然朴素贝叶斯的模型会失去词语之间的顺序信息，但考虑到有些独立假设在各个分类之间的分布都是均匀的所以对于似然的相对大小不产生影响；即便不是如此，也有很大的可能性各个独立假设所产生的消极影响或积极影响互相抵消，最终导致结果受到的影响不大，所以贝叶斯模型的效果确非常明显，事实证明，贝叶斯模型确实是简单、实用且强大的。

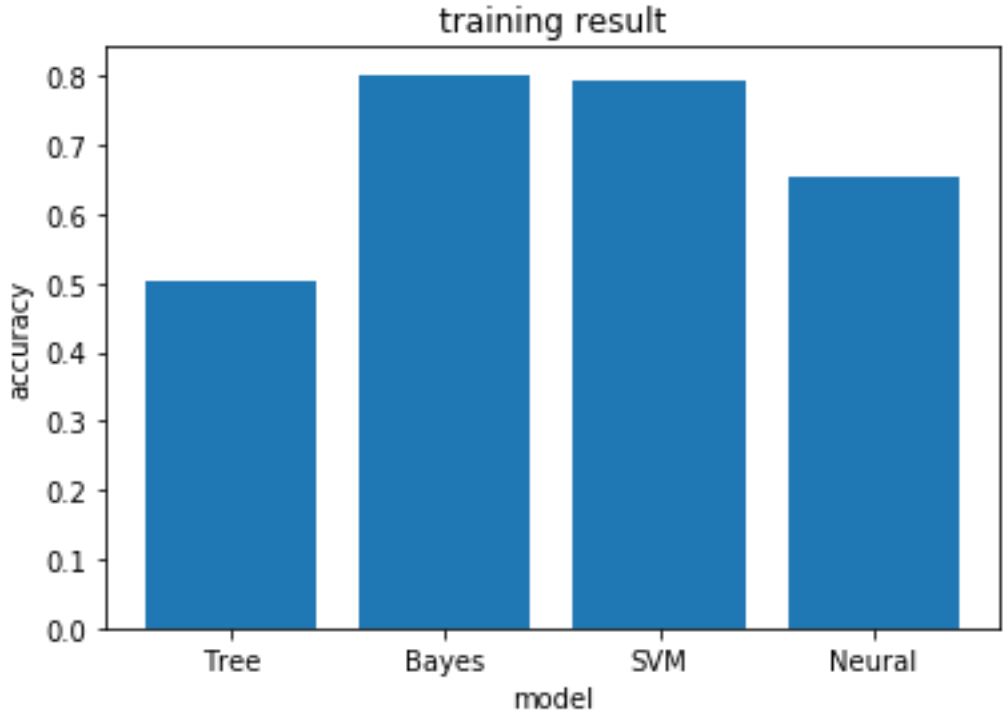
4.2 合理性分析

最终选择的朴素贝叶斯模型准确率比基准模型提高了 0.3，0.8 的准确率对于文档分类来说算一个及格的水平，可以在生产环境处理简单的文档分类要求了。支持向量机方法相比来说也不错，但是准确率稍微低了一点点。卷积神经网络的训练结果不太理想，虽然我已经把 dropout 层设置到 0.5 了，但是过拟合现象并没有消除，准确率只有 0.643，并且在层数不多的前提下，卷积神经网络的训练时间已经远远大于朴素贝叶斯和支持向量机了，随着训练数据集的增加和层数及参数的调整，训练时间是一个很大的问题。

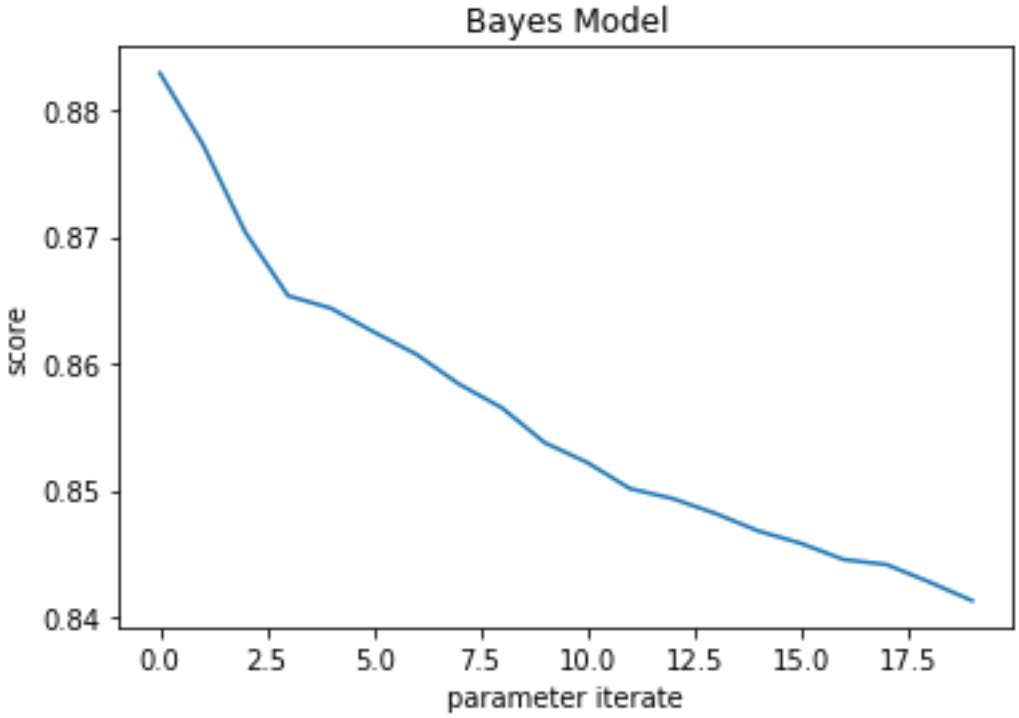
5. 项目结论

5.1 结果可视化

各个模型调优后的最高准确率如下图所示



再看看效果最好的朴素贝叶斯，不同 alpha 参数值对它的影响



可以看到随着平滑参数 alpha 的增大，准确率呈现降低趋势，在 20 类新闻分类问题上，应设置较小的平滑参数来代替默认值 1，采用平滑参数主要是用来解决在测试集上出现了训练集中没有的特征时的情况。

5.2 对项目的思考

本项目两个主要的难题是怎样表示和提取文档的特征以及选择哪种分类模型，本项目把单词作为文档特征，并采用词向量来描述这个特征，tfidf 结合了词频和逆文本频率能很好的识别出那些能有效识别某一类别文档的单词，此外还采用了 skip-gram 模式的 word2vec，word2vec 能够将稀疏、维度高的词向量矩阵进行降维，而 skip-gram 模式可以基于单词推测出近义的语句，这很符合分类新闻的需求。

分类模型分别选择了传统机器学习的决策树、朴素贝叶斯和支持向量机(决策树作为基准模型)和卷积神经网络，经优化朴素贝叶斯和支持向量机均能达到 80%左右的准确率，卷积神经网络训练结果不太理想，可以看出传统的机器学习方法在文档分类问题上还是由很好的表现的，在一些场景下，应用传统机器学习方法能更简单、快速的解决实际需求。

5.3 需要作出的改进

自己的神经网络模型未能得到很高的准确率，这可能有很多原因是自己的层和参数选择未能合理调优的原因，参考做项目过程中查阅到的文档，在有合适训练集输入以及增加诸如 LSTM、GRU 层，并不断调参后是可以得到比机器学习方法更高的准确率的。在机器性能足够的条件下，我还会继续尝试用神经网络进行训练，看能否达到更好的效果。

6. 参考资料

[1] http://www.qwone.com/~jason/20Newsgroups/

[2] http://mattmahoney.net/dc/test8.zip

[3] https://developers.google.com/machine-learning/crash-course/classification/accuracy

[4] http://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html

[5] https://deeplearning4j.org/cn/bagofwords-tf-idf

[6] https://deeplearning4j.org/cn/word2vec

[7] http://www.cnblogs.com/leoo2sk/archive/2010/09/17/naive-bayesian-classifier.html

[8] https://pythonspot.com/nltk-stop-words/

[9] http://scikit-learn.org/stable/modules/feature_extraction.html

[10] https://radimrehurek.com/gensim/models/word2vec.html

[11] https://github.com/keras-team/keras/blob/master/examples/pretrained_word_embeddings.py

[12] https://keras-cn.readthedocs.io/en/latest/