

机器学习课程报告：作业 3

一、实验简介

该实验任务是对新闻文本进行分类，并使用五折交叉验证结果。本次实验使用的是卷积神经网络（CNN），并使用了 glove 词向量作为预处理词向量，使用了三层 CNN 构建模型，最终模型的 accuracy 达到了 96.4%。除此之外，使用了朴素贝叶斯分类器作为对比方法。实验证明，CNN 的效果优于使用朴素贝叶斯分类器的方法。





















二、使用环境

本实验所使用环境如下：

Ubuntu16.04、16GB 内存、NVIDIA GTX1070（8GB 显存）、编程语言 Python 3.6、
框架：TensorFlow 1.8.0 ， Keras 2.0.4 ， Sklearn 0.19.1、IDE 为 Pycharm2018。

三、数据集

本实验使用的数据集是 20_newsgroups 数据集（下载地址：
<http://www.qwone.com/~jason/20Newsgroups/>），此数据集共有 20 个分类，如下图所示：

-  alt.atheism
-  comp.graphics
-  comp.os.ms-windows.misc
-  comp.sys.ibm.pc.hardware
-  comp.sys.mac.hardware
-  comp.windows.x
-  misc.forsale
-  rec.autos
-  rec.motorcycles
-  rec.sport.baseball
-  rec.sport.hockey
-  sci.crypt
-  sci.electronics
-  sci.med
-  sci.space
-  soc.religion.christian
-  talk.politics.guns
-  talk.politics.mideast
-  talk.politics.misc
-  talk.religion.misc

其中，每一个文件夹代表一个新闻分类，每一个新闻分类的文件夹中含有 1000 个新闻文件（soc.religion.christian 有 997 个），共有 19997 个新闻文本文件。每个文件中的内容为文本数据，下面是编号为 82758 的文本文件的内容：

```
1 Newsgroups: talk.religion.misc
2 Path:
  cantaloupe.srv.cs.cmu.edu!crabapple.srv.cs.cmu.edu!fs7.ece.cmu.edu!europa.eng.gtefsd.com!howlan
  d.reston.ans.net!noc.near.net!uunet!quack!pharvey
3 From: pharvey@quack.kfu.com (Paul Harvey)
4 Subject: Re: 5 Apr 93 . . . God's Promise in Psalm 85: 8
5 Message-ID: <flwtP5p@quack.kfu.com>
6 Organization: The Duck Pond public unix: +1 408 249 9630, log in as 'guest'.
7 References: <C50KDr.Duz@acsu.buffalo.edu>
8 Date: 5 Apr 1993 18:04:59 UTC
9 Lines: 21
10
11 In article <C50KDr.Duz@acsu.buffalo.edu>
12 psyrobtw@ubvmsb.cc.buffalo.edu (Robert Weiss) writes:
13 > I will hear what God the LORD will speak:
14 > for he will speak peace
15 > unto his people, and to his saints:
16 > but let them not turn again to folly.
17
18 Psalm85 (JPS): For the leader. Of the Korahites. A psalm. O LORD, You
19 will favor Your land, restore Jacob's fortune; You will forgive Your
20 people's iniquity, pardon all their sins; Selah; You will withdraw all Your
21 anger, turn away from Your rage. Turn again, O God, our helper, revoke
22 Your displeasure with us. Will you be angry with us forever, prolong
23 Your wrath for all generations? Surely You will revive us again, so that
24 Your people may rejoice in You. Show us, O LORD, Your faithfulness;
25 grant us Your deliverance. Let me hear what God, the LORD, will speak;
26 He will promise well-being to His people, His faithful ones; may they
27 not turn to folly. His help is very near those who fear Him, to make His
28 glory dwell in our land. Faithfulness and truth meet; justice and
29 well-being kiss. Truth springs up from the earth; justice looks down
30 from heaven. The LORD also bestows His bounty; our land yields its
31 produce. Justice goes before Him as He sets out on His way.
```

下面是数据集文件的读取和处理，主要是将每一个文件夹的文件及其分类对应起来，将其和输入对应存入相应的数据结构中，代码如下，主要将数据集的文件存储 texts 中，将其对应的标签文件存入 labels 中：

```
texts = [] # list of text samples
labels_index = {} # dictionary mapping label name to numeric id
labels = [] # list of label ids
for name in sorted(os.listdir(TEXT_DATA_DIR)):
    path = os.path.join(TEXT_DATA_DIR, name)
    if os.path.isdir(path):
        label_id = len(labels_index)
        labels_index[name] = label_id
        for fname in sorted(os.listdir(path)):
            if fname.isdigit():
                fpath = os.path.join(path, fname)
                f = open(fpath, encoding='latin-1')
                texts.append(f.read())
                f.close()
                labels.append(label_id) # labels 和 texts 是一一对应的
```

四、预训练的词向量

本模型在表示文本时使用的是预训练的词向量，使用的预训练词向量来自斯

斯坦福大学的公开数据 (<https://github.com/stanfordnlp/GloVe>)，此练习中使用的词向量是使用 2014 年维基百科数据和 Gigaword 数据进行训练的，在下载的训练数据集中有 4 个文件，分别是 glove.6B.50d.txt, glove.6B.100d.txt, glove.6B.200d.txt 和 glove.6B.300d.txt，即最终生成的词向量分别是 50 维、100 维、200 维和 300 维。因为实验环境的制约，这里使用 50 维的预训练向量，其内容如下：

```
1 the 0.418 0.24968 -0.41242 0.1217 0.34527 -0.044457 -0.49688 -0.17862 -0.00066023 -0.6566 0.27843 -0.14767 -0.55677 0.14658 -0.0095095 0.011658
2 , 0.013441 0.23682 -0.16899 0.40951 0.63812 0.47709 -0.42852 -0.55641 -0.364 -0.23938 0.13001 -0.063734 -0.39575 -0.48162 0.23291 0.090201 -0.1
3 . 0.15164 0.30177 -0.16763 0.17684 0.31719 0.33973 -0.43478 -0.31086 -0.44999 -0.29486 0.16608 0.11963 -0.41328 -0.42353 0.59868 0.28825 -0.115
4 of 0.70853 0.57088 -0.4716 0.18048 0.54449 0.72603 0.18157 -0.52393 0.10381 -0.17566 0.078852 -0.36216 -0.11829 -0.83336 0.11917 -0.16605 0.061
5 to 0.68047 -0.039263 0.30186 -0.17792 0.42962 0.032246 -0.41376 0.13228 -0.29847 -0.085253 0.17118 0.22419 -0.10046 -0.43653 0.33418 0.67846 0.
6 and 0.26818 0.14346 -0.27877 0.016257 0.11384 0.69923 -0.51332 -0.47369 -0.33075 -0.13834 0.2702 0.30938 -0.45012 -0.4127 -0.09932 0.038085 0.0
7 in 0.33042 0.24995 -0.60874 0.10923 0.036372 0.151 -0.55083 -0.074239 -0.092307 -0.32821 0.09598 -0.82269 -0.36717 -0.67009 0.42909 0.016496 -0
8 a 0.21705 0.46515 -0.46757 0.10082 0.10135 0.74845 -0.53104 -0.26256 0.16812 0.13182 -0.24909 -0.44185 -0.21739 0.51004 0.13448 -0.43141 -0.0312
9 " 0.25769 0.45629 -0.76974 -0.37679 0.59272 -0.063527 0.20545 -0.57385 -0.29009 -0.13662 0.32728 1.4719 -0.73681 -0.12036 0.71354 -0.46098 0.65
10 'a 0.23727 0.40478 -0.20547 0.58805 0.65533 0.32867 -0.81964 -0.23236 0.27428 0.24265 0.054992 0.16296 -1.2555 -0.086437 0.44536 0.096561 -0.16
11 for 0.15272 0.36181 -0.22168 0.066051 0.13029 0.37075 -0.75874 -0.44722 0.22563 0.10208 0.054225 0.13494 -0.43052 -0.2134 0.56139 -0.21445 0.07
12 - 0.16768 1.2151 0.49515 0.26836 -0.4585 -0.23311 -0.52822 -1.3557 0.16098 0.37691 -0.92702 -0.43904 -1.0634 1.028 0.0053943 0.04153 -0.018638
```

官网上提供的文件便是文本文件，非二进制文件，看起来比较容易理解，并且很容易处理，从图中可以看出，每一行表示一个词向量的训练结果，如第一行表示 “the”，其对应的词向量便是后面对应的 50 个数据。

五、基于 TensorFlow 框架训练神经网络

下面代码是没有加入五折交叉验证情况下的神经网络框架，本架构是基于预训练的词向量进行处理的。处理文本数据需要将文本数据序列化，本实验设置的最大序列长度是 1000。对于句子长度的限制是 2000，比 2000 长的文本后面会进行截断，比 2000 短的文本会自动补 0。其步骤如下：

Step1: 读入新闻文本数据并将其与标签对应存储。

Step2: 文本数据标准化、序列化处理，并将其转化成 tensor 格式。

Step3: 生成词向量矩阵。

Step4: 将词向量矩阵载入 Keras Embedding 层，并设置之后不再改变参数。

Step5: 设置 CNN 参数，并进行训练预测。

网络结构的设置如下：

```
x = Conv1D(128, 5, activation='tanh')(embedded_sequences) # (?, 996, 128)
x = MaxPooling1D(5)(x) # (?, 199, 128)
x = Conv1D(128, 5, activation='tanh')(x) # (?, 195, 128)
x = MaxPooling1D(5)(x) # (?, 39, 128)
x = Conv1D(128, 5, activation='tanh')(x) # (?, 35, 128)
x = MaxPooling1D(35)(x) # (?, 1, 128)
x = Flatten()(x) # 128
x = Dense(128, activation='tanh')(x)
preds = Dense(len(labels_index), activation='softmax')(x)
```

网络使用了 3 层卷积层池化，设置的维度如上所示，向量维度的变化如右侧注释，使用的激活函数是 tanh。

六、使用验证集判断模型效果

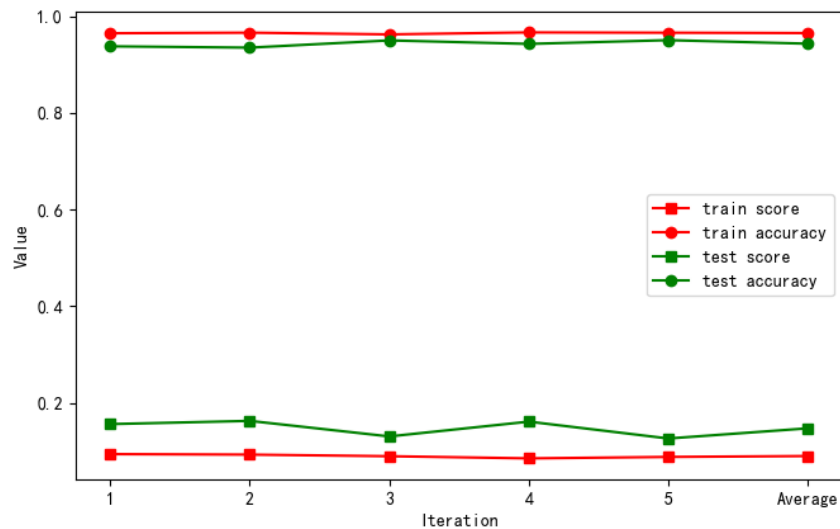
1、使用 CNN 模型效果

使用五折交叉验证，参数设置和第三节一致的情况下的性能如下表：

Iteration1	Iteration2	Iteration3	Iteration4	Iteration5	Average
------------	------------	------------	------------	------------	---------

Train score	.09405	.09305	.08970	.08525	.08821	.09005
Train accuracy	.96445	.96569	.96218	.96625	.96562	.96485
Test score	.15613	.16276	.13062	.16123	.12635	.14742
Test sccuracy	.93749	.93473	.94974	.94249	.95024	.94294

其中，score 值表示损失函数值，值越小越好。图形展示为：



从结果可知，在训练集准确率达到到了 96.4%，在测试集上达到了 94.2%左右。

2、使用朴素贝叶斯效果

此贝叶斯模型使用的是 sklearn 中的 MultinomialNB()模型，使用的是五折交叉验证，主要代码如下：

```
rawData = datasets.load_files("../data/20_newsgroups/", encoding='latin-1')    # 加载文件夹中文件

vec = CountVectorizer()    # 向量化模块
X = vec.fit_transform(rawData.data)    # 归一化处理
y = rawData.target

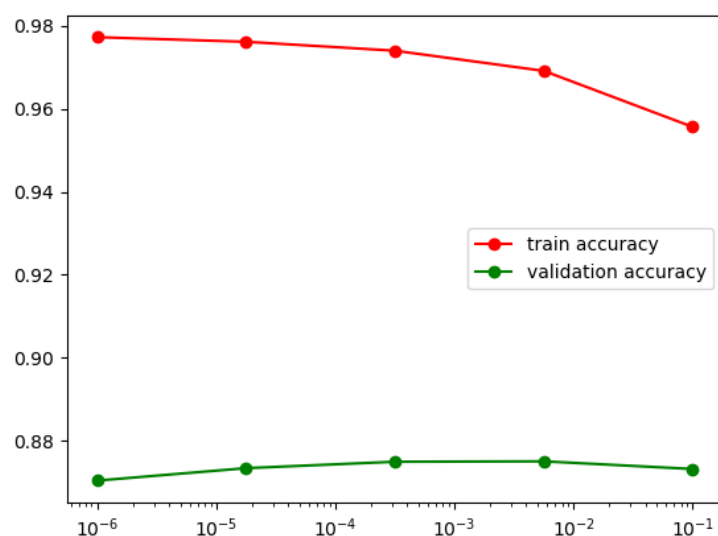
param_range = np.logspace(-6, -1, 5)
train_accuracy, validation_accuracy = validation_curve(MultinomialNB(), X, y, cv=5,
param_name="alpha", param_range=param_range, scoring="accuracy")

# by the following code to get the mean value of every cross validation step
train_accuracy_mean = train_accuracy.mean(1)
train_accuracy_std = train_accuracy.std(1)
validation_accuracy_mean = validation_accuracy.mean(1)
validation_accuracy_std = validation_accuracy.std(1)

print("train_accuracy_mean: ", train_accuracy_mean.mean(0))    # get the mean value of the
five cross validation
```

```
print("validation_accuracy_mean: ", validation_accuracy_mean.mean(0))
```

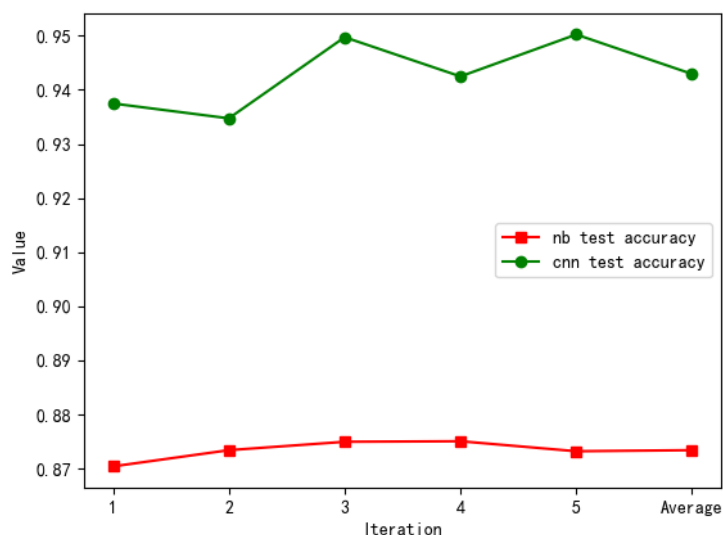
算法效果如下图所示，下图中每一个点表示对应参数的 5 折交叉验证的平均值，比如 10^{-6} 表示当 α 的值是 10^{-6} 相应的准确率值。



模型最终在训练集上的准确率是 0.9704430514900817，在测试集上的准确率是 0.8733812603150788。

七、不同模型效果比较

使用 CNN 模型和使用 NB 模型的执行结果对比如下图，从图中可以看出，相比较于 NB，CNN 具有绝对的优势。但是就执行效率来说，CNN 效率较低，需要消耗大量的硬件资源，NB 的效率更高些。



八、总结

本次实验是对新闻文本进行分类，并使用五折交叉验证结果。本文使用了

卷积神经网络（CNN）模型和朴素贝叶斯模型。在 CNN 模型中，使用了 glove 词向量作为预处理词向量，使用了三层 CNN 构建模型，最终模型的 accuracy 达到了 96.4%。本实验使用了朴素贝叶斯分类器作为对比方法，朴素贝叶斯的平均准确率是 94.2%，比 CNN 低 2.2%。

虽都是使用网络构建模型，但是相对于图片来说，一些细节还是有一些不同，图像考虑像素点，是一个 2 维或者 3 维的矩阵，而对于文本来说基于 1 维属性，且利用的词向量质量对结果会有影响。