NLP-Beginner

Task1: 基于机器学习的文本分类

姓名: 王沛晟　　学号: 15307130021

一、 数据集划分与模型建立

将提供的数据集 train.tsv 打乱后，90%作为训练集，10%作为验证集。使用提供的测试集 test.tsv 作为最终模型评估的标准。

考虑到影评有 0,1,2,3,4 五种结果，且有递进关系，使用 logistic regression 模型，采用交叉熵作为损失函数，训练四个分类器，第 k 个分类器以影评结果是否大于等于 k 作为分类依据。每条影评均由若干单词构成，采用 bag-of-words 和 n-gram 分别进行测试。

二、 词袋模型

使用 batch, mini-batch 和 shuffle 进行梯度下降，得到的结果对比如下：

Method	Batch size	Learning rate	Epoch	Time	Accuracy
Shuffle	1	3	5	1176.89	0.5428
Mini-batch	16	3	5	1054.24	0.5722
Mini-batch	64	3	5	976.15	0.5768
Mini-batch	128	3	5	1041.26	0.5732
Batch	140453	3	5	966.89	0.5250

由此可见，在相同 epoch 数量的情况下，采用 mini-batch 的效果最好。Shuffle 由于不能很好的应用并行计算能力，因此每个 epoch 的时间更长，loss 收敛所需要的时间也更长。Batch 能够很好的应用并行计算能力，但是本次训练集有 14 万余行，每次迭代的空间代价太大。

使用 mini-batch(batch size=64)，改变 learning rate 和 epoch，在验证集（共计 15607 条）上得到的结果对比如下：

（注：Distribution 表示预测的结果中 0,1,2,3,4 的分布情况，真实情况为 [745, 3124, 7617, 3218, 903]）

Learning rate	Epoch	Time	Accuracy	Distribution
0.1	1	338.34	0.5027	[0, 271, 14640, 681, 15]
0.1	2	521.07	0.5102	[7, 408, 14382, 794, 16]
0.1	3	711.76	0.5171	[11, 509, 13965, 1100, 22]
0.3	1	329.65	0.5189	[11, 635, 13874, 1067, 20]
0.3	2	525.36	0.5278	[18, 820, 13546, 1178, 45]
0.3	3	682.60	0.5359	[30, 937, 13137, 1431, 72]
1	1	331.10	0.5394	[38, 985, 12998, 1443, 143]
3	5	976.15	0.5768	[203, 1391, 11605, 2068, 340]
3	10	1791.80	0.5890	[157, 1871, 10616, 2638, 325]
5	5	979.45	0.5901	[202, 1920, 10551, 2677, 257]
5	10	1799.05	0.5858	[216, 2269, 10130, 2439, 553]
7	5	985.48	0.5852	[182, 2505, 10443, 2042, 435]
7	10	1795.02	0.5892	[264, 1837, 10529, 2410, 567]
5	20	3346.71	0.5854	[344, 1969, 10078, 2605, 611]
5	100	16133.09	0.5737	[344, 2338, 8844, 3622, 459]

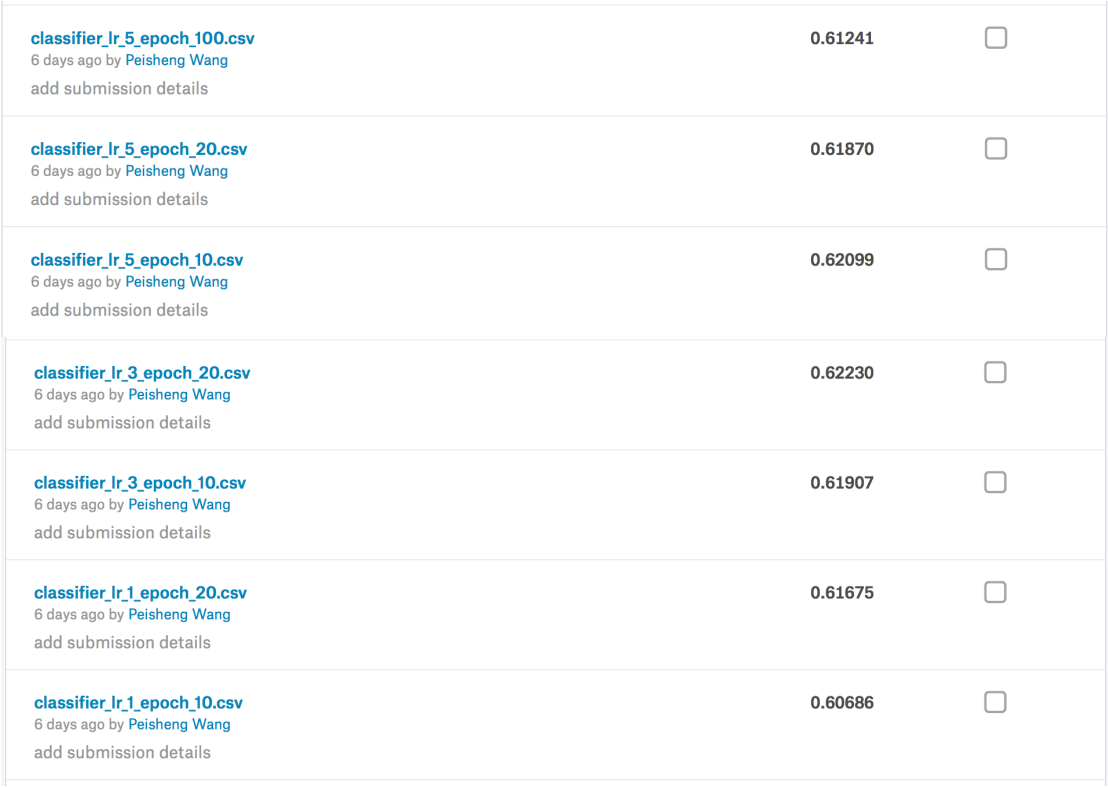
Learning rate	Iterations	Accuracy
10	100	0.4504
10	1000	0.5405
10	5000	0.5642
10	10000	0.5559
15	100	0.2769
15	1000	0.4686
15	5000	0.5484
15	10000	0.5710
20	100	0.2444
20	1000	0.3550
20	5000	0.4904

20	10000	0.5619
30	100	0.4926
30	1000	0.4687
30	5000	0.5599
30	10000	0.5608

可以得出的结论大致有：

1. 相同 learning rate 时，随着 epoch 次数增加，accuracy 先提升后降低（从欠拟合到过拟合）。
2. Learning rate 越小，需要达到拟合的 epoch 次数也越多。
3. Learning rate 太大会使得模型无法达到局部最优解，从而预测效果不佳。
4. Epoch 次数越多，预测的分布情况越接近真实的分布情况。
5. 在本次实验中，效果最好的 learning rate 大约在 5 左右，10 个 epoch 拟合程度就已经不错。

由此，我们将 learning rate 取[1,3,5]，将 epoch 取[10,20,100]，在测试集上测试对应的模型，结果如下：



可以看到，learning rate 为 1 或 3 时，需要 20 个 epoch 达到的效果更好。而 learning rate 为 5 时，只需要 10 个 epoch 的效果最好。虽然 loss 收敛得更快，但最终结果并没有 learning rate 为 3 时的好。

最终 learning rate 为 3，epoch 为 20 的模型在测试集上表现最好，accuracy 为 0.62230。

三、 N-gram 模型

使用 ngram 模型，取 ngram range 为 1 到 3，并使用 mini-batch 方法（batch size 为 64）进行梯度下降，得到如下结果：

注： ngram 后产生的词向量大约为十八万维，受限制于内存大小及训练时间（大致与训练集大小的平方成正比），只能将训练集调整为前 30000 条数据。

Learning rate	Epoch	Time	Accuracy
1	2	358.72	0.5127
1	5	572.83	0.5238
1	10	928.71	0.5287
3	2	358.55	0.5178
3	5	571.61	0.5371
3	10	927.65	0.5394
5	2	358.88	0.5272
5	5	570.48	0.5388
5	10	928.53	0.5472
10	2	357.67	0.5418
10	5	571.09	0.5392
10	10	928.60	0.5465

可以看到，在相同 epoch 次数的情况下，ngram 模型表现不如词袋模型。可能是由于词向量维数大幅度增加，训练的开销也大幅增加，在短时间内很难取得很好的效果。

四、 结语

本次实验通过对一个影评语料库进行情感评价，我主要学习了如何实现基于 logistic/softmax regression 的文本分类。

作为机器学习的第一个入门任务，我学习到了一些基础知识，如：数据集的划分、文本的特征表示：Bag-of-Word 和 N-gram 模型、损失函数、梯度下降的不同方法，并分析了不同的特征、损失函数、学习率对最终分类性能的影响。