

# 情感分析实验报告

高敬越 计 92 2019011230

June 6, 2021

## 目录

<b>1 模型结构与流程分析</b>	<b>1</b>
1.1 句子处理 . . . . .	1
1.2 模型结构 . . . . .	2
1.2.1 MLP . . . . .	2
1.2.2 CNN . . . . .	2
1.2.3 RNN . . . . .	3
<b>2 实验结果</b>	<b>3</b>
<b>3 参数选择</b>	<b>4</b>
3.1 词向量维度 . . . . .	4
3.2 learning rate . . . . .	4
3.3 卷积核 . . . . .	4
3.4 dropout 概率 . . . . .	4
<b>4 baseline 和 CNN, RNN 的效果差异</b>	<b>5</b>
<b>5 问题思考</b>	<b>5</b>
<b>6 总结和改进</b>	<b>7</b>

## 1 模型结构与流程分析

### 1.1 句子处理

采用 V2 数据集, 利用 Word2Vec 模型, 在通过神经网络前, 先通过神经网络中的 embedding 层, 将一个句子转换为 [len, word\_vec\_len] 的二维张量。对于 rnn 来说还需要记录每个句子的长度, mlp 与 cnn 可以统一转换为最长的句子长度。

对于词向量, 可以选择随机初始化, 在训练时通过反向传播更新词向量; 也可以选择用 gensim 库提前根据语料训练词向量, 再到训练中继续进行微调。在实际应用中, 两者效果差不多, 甚至随机初始化效果更好一些, 可能是因为在训练词向量时数据量太小, 且迭代数不够造成的。在后面的实验中都采用随机初始化词向量的方法。

## 1.2 模型结构

### 1.2.1 MLP

以 mlp 网络作为实验的 baseline，实现了一层隐含层的 MLP 网络，并在其间设置了 dropout，并选择 softplus 作为隐含层的激励函数。输出后取 Softmax，得到分别为 7 种情感的概率，并计算交叉熵损失函数。输入词向量构成的矩阵转化为一维向量通过 mlp。

```
MLP(  
    (embedding): Embedding(6998, 40)  
    (mlp): Sequential(  
        (0): Linear(in_features=4000, out_features=128, bias=True)  
        (1): Softplus(beta=1, threshold=20)  
        (2): Dropout(p=0.5, inplace=False)  
        (3): Linear(in_features=128, out_features=7, bias=True)  
    )  
    (loss_function): CrossEntropyLoss()  
)
```

### 1.2.2 CNN

选择 CNN 网络，在得到句子的词向量矩阵后，通过卷积层，卷积层的卷积核大小和个数可以列表形式自定义。通过卷积核后的向量连在一起，通过全连接层得到 7 种情感的概率，并计算 loss。

```
CNN(  
    (loss_function): CrossEntropyLoss()  
    (embedding): Embedding(6998, 40)  
    (conv): ModuleList(  
        (0): Sequential(  
            (0): Conv1d(40, 128, kernel_size=(3,), stride=(1,), padding=(1,))  
            (1): ReLU()  
            (2): Dropout(p=0.4, inplace=False)  
        )  
        (1): Sequential(  
            (0): Conv1d(40, 128, kernel_size=(4,), stride=(1,), padding=(2,))  
            (1): ReLU()  
            (2): Dropout(p=0.4, inplace=False)  
        )  
    )  
    (mlp_network): Sequential(  
        (0): Linear(in_features=256, out_features=128, bias=True)  
        (1): ReLU()  
        (2): Dropout(p=0.4, inplace=False)  
        (3): Linear(in_features=128, out_features=7, bias=True)  
        (4): Softmax(dim=1)  
    )  
)
```

)

### 1.2.3 RNN

实现了 LSTM 模型的 RNN，选择双向 LSTM 的模型。得到句子对应的词向量矩阵后，通过转化放入 lstm 中，取出句尾的结果，拼接后通过 mlp 和 Softmax 层得到 7 种情感的概率。

```
RNN(  
    (embedding): Embedding(6998, 40)  
    (loss_function): CrossEntropyLoss()  
    (rnn): LSTM(40, 128, batch_first=True, bidirectional=True)  
    (mlp): Sequential(  
        (0): Linear(in_features=256, out_features=128, bias=True)  
        (1): ReLU()  
        (2): Dropout(p=0.5, inplace=False)  
        (3): Linear(in_features=128, out_features=7, bias=True)  
    )  
)
```

## 2 实验结果

在训练时使用 Adam 算法进行迭代，得到结果：

	准确率	Macro	Micro
MLP	0.4843546	0.48176845	0.484354615
CNN	0.5573663	0.55390837	0.557366362
RNN	0.4863103	0.48655899	0.486310299

表 1: 实验结果

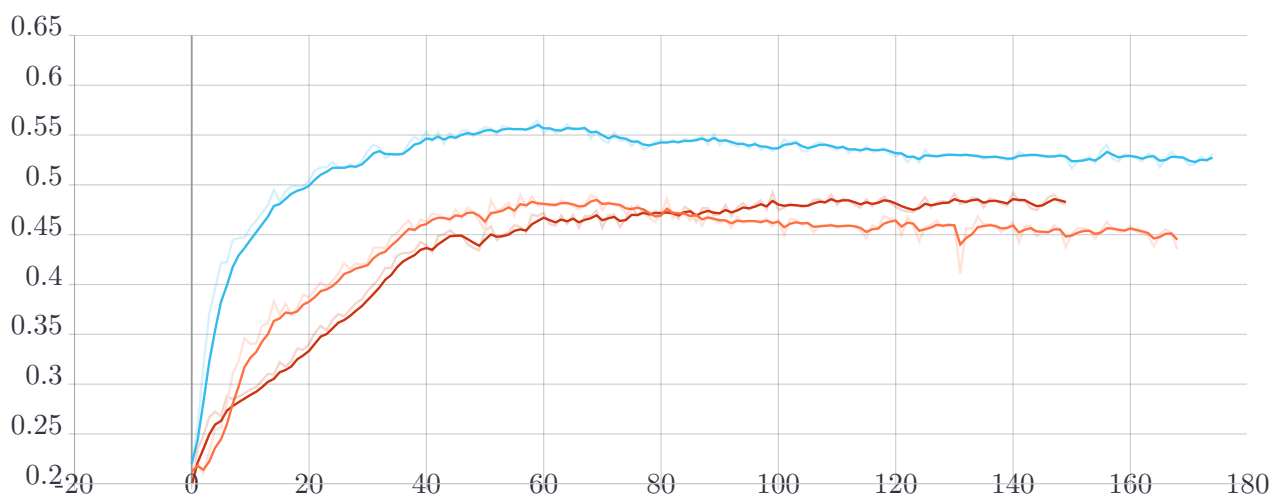


图 1: 验证集准确率，蓝：cnn，橙：rnn，红：mlp

### 3 参数选择

#### 3.1 词向量维度

根据理论分析，词向量维度越大表示能力越强，但是限于词库，如果过大可能会对词向量出现欠拟合，且训练时长增加；而词向量维度较小则表示能力较弱，准确率较低。在 CNN 网络上对不同大小的词向量维度 30、40、100 进行测试，可以得到结果：

	准确率	Macro	Micro	平均 epoch 用时
30	0.5410691	0.53945715	0.541069100	11.5s
40	0.5573663	0.55390837	0.557366362	12s
100	0.5541068	0.553989340	0.554106910	28s

表 2: 词向量维数实验结果

最终选择 40.

#### 3.2 learning rate

根据理论，learning rate 越大则收敛的越快，但是可能会由于 learning rate 过大而无法收敛到最优；学习率越小则耗费时间越长，且有可能落入局部最优。

	准确率	Macro	Micro
1e-2	0.5254237	0.52115490	0.525423728
1e-3	0.5573663	0.55390837	0.557366362
1e-4	0.4680573	0.46522443	0.468057366

表 3: 不同 lr 实验结果

#### 3.3 卷积核

测试了两种选择，选取的卷积核分别为 [3,4] 和 [3,4,5]。根据理论，卷积核越大即为考虑的词语上下文越多，越小越能关注到词语本身的意思。因此卷积核越多，覆盖越大更能全面的提取信息，但计算量也会增加。因此选择 [3,4] 和 [3,4,5] 两种卷积核的选择，经过实验发现符合预期，且 [3, 4, 5] 时长也可以接受，因此选择 [3,4,5] 作为卷积核。

	准确率	Macro	Micro
[3,4]	0.5541069	0.55286013	0.554106910
[3,4,5]	0.5573663	0.55390837	0.557366362

表 4: 卷积核个数实验结果

#### 3.4 dropout 概率

根据理论分析，dropout 概率越大则越能解耦，防止过拟合，但是同时训练难度增大，训练时间变长且更难收敛；而 dropout 概率过小则会造成过拟合到训练集上，训练集准确率高，但在测试集和验证集上表现不佳。

	准确率	Macro	Micro
0.1	0.5332464	0.53489534	0.533246414
0.4	0.5573663	0.55390837	0.557366362
0.9	0.5091264	0.50881046	0.509126466

表 5: 不同 dropout 实验结果

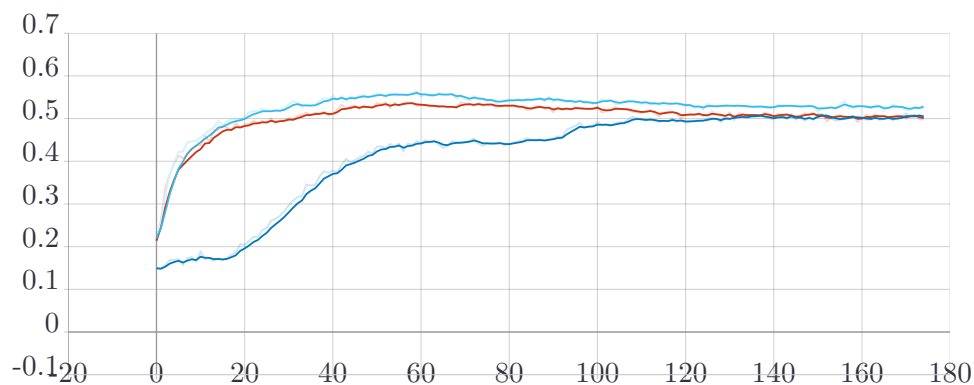


图 2: 验证集准确率, 深蓝: 0.9, 浅蓝: 0.4, 红: 0.1

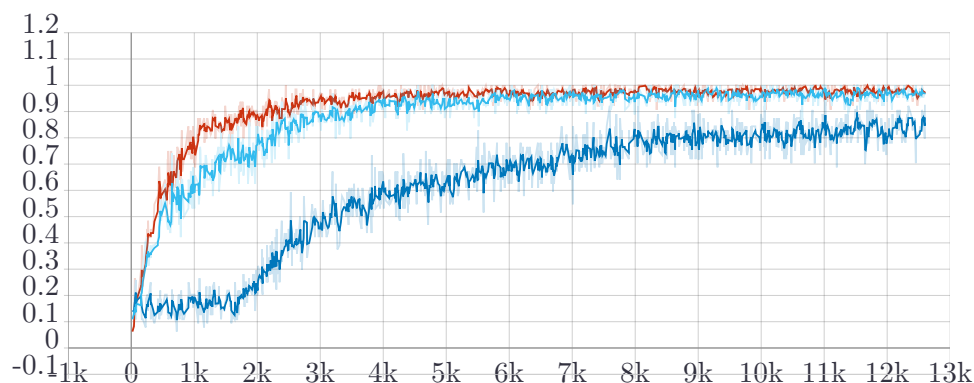


图 3: 训练集准确率, 深蓝: 0.9, 浅蓝: 0.4, 红: 0.1

## 4 baseline 和 CNN, RNN 的效果差异

经过实验来看, CNN 的效果最好, RNN 和 MLP 的效果次之。RNN 和 MLP 相比准确率稍高, 但对参数的设置较为敏感, 需要精确调参。这次实验设计的 CNN 和 RNN 在最后都采用了和 MLP 相同的网络结构, 只是参数有所不同, 说明卷积网络和循环网络对句子都有着辅助提取特征的作用。但可能是因为句子不够多且长度较短, RNN 的记忆效应效果不佳, 和 MLP 直接分类相近。

## 5 问题思考

1. 如果控制实验训练的停止时间? 简要陈述你的实现方式, 并试分析固定迭代次数与 early stopping 等方法的优缺点。

我控制实验训练的停止时间的方式类似固定迭代次数。首先选择足够大的 epoch 个数，根据模型在 evaluate 数据上的准确率和 loss 等指标来判断是否过拟合，例如结果中的 eval/accuracy。这样可以确定模型经过多少次迭代会过拟合，并据此设置合适的 epoch 数值，并选择其中表现最好的 epoch 的网络参数。

固定迭代次数的优点为可以更全面的掌握在不同时段的模型的表现和拟合情况，但是缺点为时间较长；

提前结束的优点为可以根据模型和任务适当调整结束的条件，并且使用灵活，时间短，但缺点为可能出现一些波动就会提前停止，未能达到最优。

**2. 过拟合和欠拟合是深度学习常见的问题，有什么方法可以解决上述问题。**

解决过拟合：

- (a) 增加数据集大小，或者适当减少网络的参数，减弱其表示能力，防止过拟合到训练集上，或者进行交叉检验；
- (b) 在神经网络中添加适当的结构，例如 Batch Normalization、Dropout 等层来解耦防止过拟合；
- (c) 还可以增加正则化 l1, l2 loss 等，防止模型过于复杂。

解决欠拟合首先也要判断模型的表达能力是否不足，适当增加新的特征和空间；同时如果有上面的 bn、dropout 层则适当减小参数；在我的实验中还发现可能学习率过高，无法收敛。

**3. 试分析梯度消失和梯度爆炸产生的原因，以及对应的解决方式。**

梯度消失和梯度爆炸是在反向传播时因为链式法则的连乘，而导致导数接近 0 或变得非常大。这可能是由于在网络较深时，选择的激活函数 (sigmoid) 等的导数接近 0 或者初始化的权重大于 1 造成的。为解决梯度消失和梯度爆炸，有多种方式以及其他没有提到的方式：

- (a) 选择适当的激活函数，如 Relu 等；
- (b) 在更新时设置一个梯度的范围，使得梯度值不要过大；
- (c) 初始化时选择恰当的初始化方法；
- (d) 使用合适的网络结构，例如在 RNN 中选用 LSTM 结构，或者选用残差结构（这个没有用上）

**4. 试分析 CNN，RNN，全连接神经网络 (MLP) 三者的优缺点与各自适用的场景。**

	MLP	CNN	RNN
优点	模型简单，易于训练与理解，并且以全局信息训练网络参数	结构清晰，利用卷积核提取句子上下文信息，准确率高；	加入记忆项进行递推，具有时序性；且句子长度可以改变
缺点	结构简单，表达能力不足，准确率低；若增加参数以增强表达能力，则参数数量过多；同时句子最大长度固定，会出现很多空词的情况。	理论上对于情感可以通过局部推断的句子更好，但对全局信息把握不够。	难以调节，经常出现梯度消失或梯度爆炸；运算较慢，难以对 mini batch 进行并行计算
适用场景	简单的分类，需要快速训练	大部分场景表现都很优秀，并且 MLP 的引入有效避免填充时过多 0 的出现	应该在有较长句子时可以更好提取情感特征

表 6

## 6 总结和改进

1. 在实现了作业要求的 3 个神经网络后，我对神经网络的使用和原理都有了更深刻的认识，也理解了 Word2Vec 等方法的原理，丰富了实现神经网络的经验；
2. 在实现过程中，遇到了很多问题，例如句子如何处理、怎么确定参数、如何确定网络结构等等，丰富了调参和设计经验；同时确定神经网络中张量维数时也遇到了很多问题，丰富了解决问题的经验；
3. 在最初实现后，发现准确率并不高。在经过学习相关代码资料和阅读相关文章后，不断调整其中的问题，最终将准确率提升到我能达到的最高水平；
4. 感谢老师和助教的讲解。