情感分析作业报告

吴宸昊 2019011338 计96

一、流程分析

1. 数据预处理

本次实验使用ISEAR2数据。对于数据集中的每一项，提取出句子与对应的情感序列（一个七元组，包含6个0与1个1）。之后利用`tensorflow.keras.preprocess.text`中的Tokenizer工具。

```
tokenizer = Tokenizer(num_words=None,
                      filters='!"#$%&()*+,-./:;<=>?@[\]^_`{|}~\t\n',
                      lower=True,
                      split=' ',
                      oov_token="oov")
tokenizer.fit_on_texts(training_sentences)
training_sequences = tokenizer.texts_to_sequences(training_sentences)
```

处理训练集中所有的语句，得到词典集，并利用该词典集将每一条语句根据该词典映射为唯一整数序列号。

```
training_padding = pad_sequences(training_sequences, padding='post', truncating='post',
maxlen=200)
```

将得到的所有序列号进行填充处理，用0填充为最大长度为200的整数序列。

```
training_motions = np.array(training_motions).astype('int32')
```

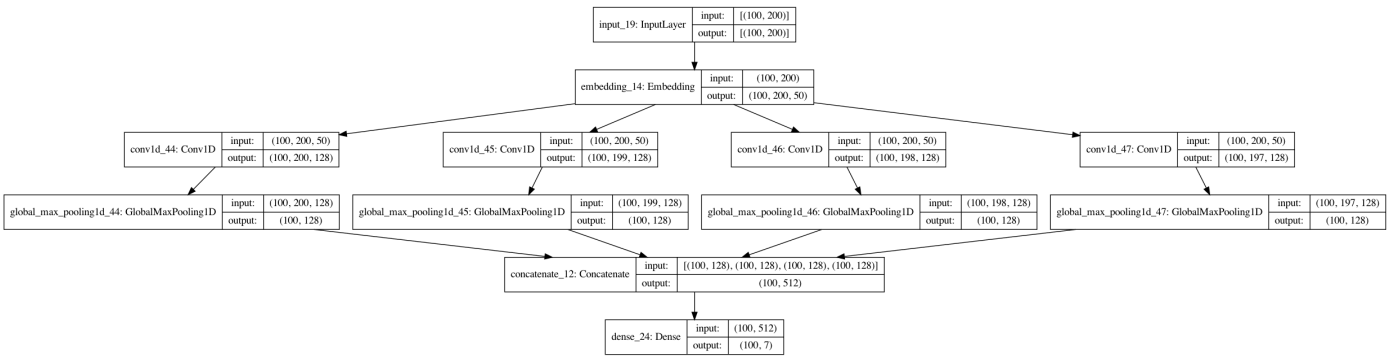
将情感序列转为numpy类型。这样我们就得到了(4600,200)的句数组与(4600,7)的情感数组。

对测试集做类似的处理。

2. 构建神经网络

1）CNN卷积神经网络

神经网络结构如图所示（详情可见/model/CNN.png）。



Embedding

输入句子首先经过embedding层，句子的每一个词被映射为一个长为DIM的向量，实验过程中DIM取作50；故整个句子映射为$LEN \times DIM$的矩阵，实验过程中该矩阵的大小为200*50.

卷积层

通过Embedding层后的矩阵会通过一系列并列的卷积层。卷积层的作用是提取局部特征信息。卷积核的大小在实验过程中被调整为[1,2,3,4]，表示相邻的几个词之间有联系，大小为200*50的矩阵经过一个大小为k*50的卷积核后得到一个长为(200-k+1)的向量，由于一个卷积核设置为128个频道，所以得到输出为(200-k+1)*128的矩阵。

池化层

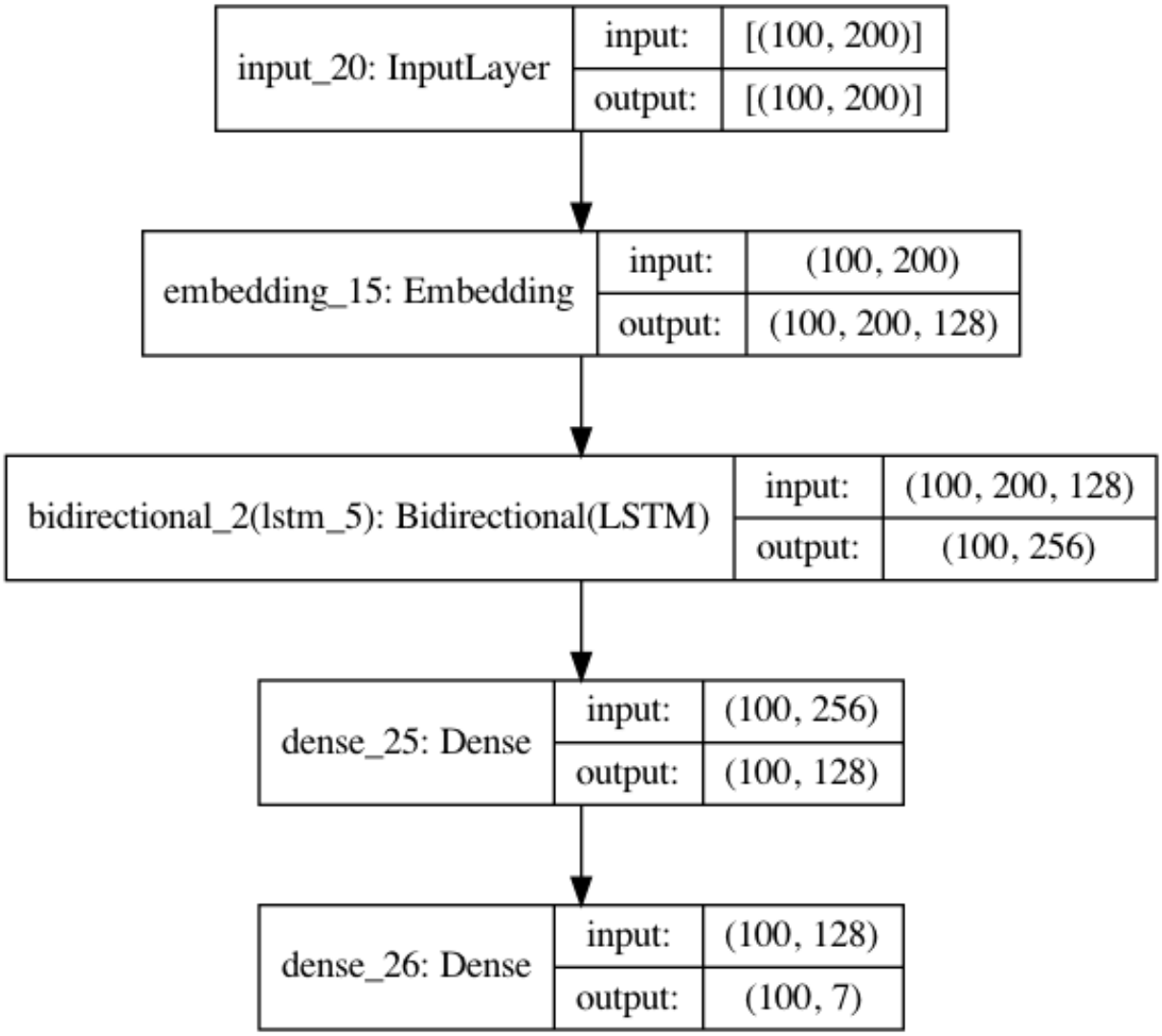
卷积层不同大小的卷积核得到的向量长度不同，需要在池化层中通过`1DMaxPooling`抽取每个特征向量的最大值，从而将其转变为一个值。所以四个卷积层结果经过各自的池化层后得到长为128的向量，将4个向量连接起来得到长为512的特征向量。

全连接层

使用全连接层将池化层中得到的高维向量转为我们需要的7维向量。可以使用L2正则项来防止过拟合。

2）RNN循环神经网络

神经网络结构如图所示（详情可见/model/RNN.png）。实验中使用到的LSTM为Bi-LSTM，是为了改进单向RNN无法处理词语在句子中前后顺序的问题。

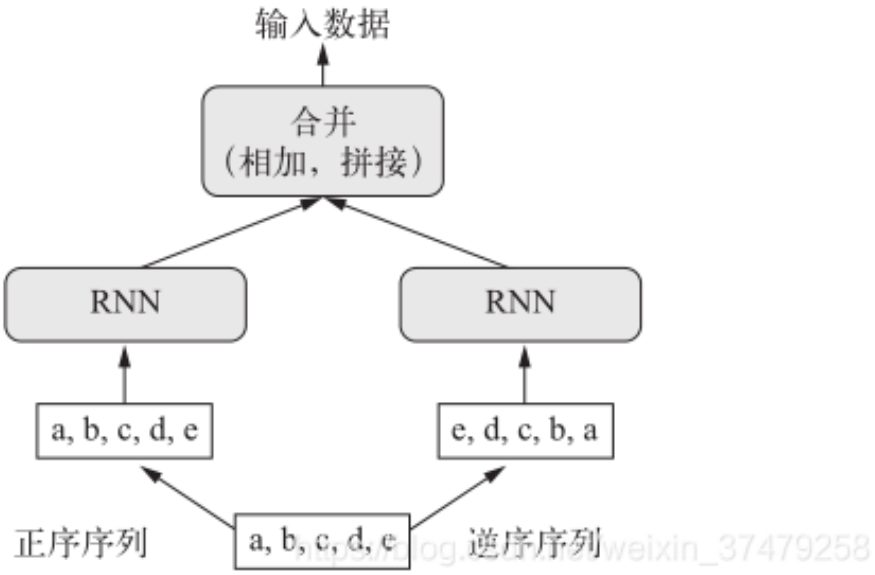


Embedding

输入句子首先经过embedding层，句子的每一个词被映射为一个长为DIM的向量，实验过程中DIM取作128；故整个句子映射为$LEN \times DIM$的矩阵，实验过程中该矩阵的大小为200*128.

LSTM

使用`Bidirectional`，双向循环LSTM蕴含其中，200*128的矩阵经LSTM层得到长为256的输出，后接全连接层。该层的结构可表示如下。

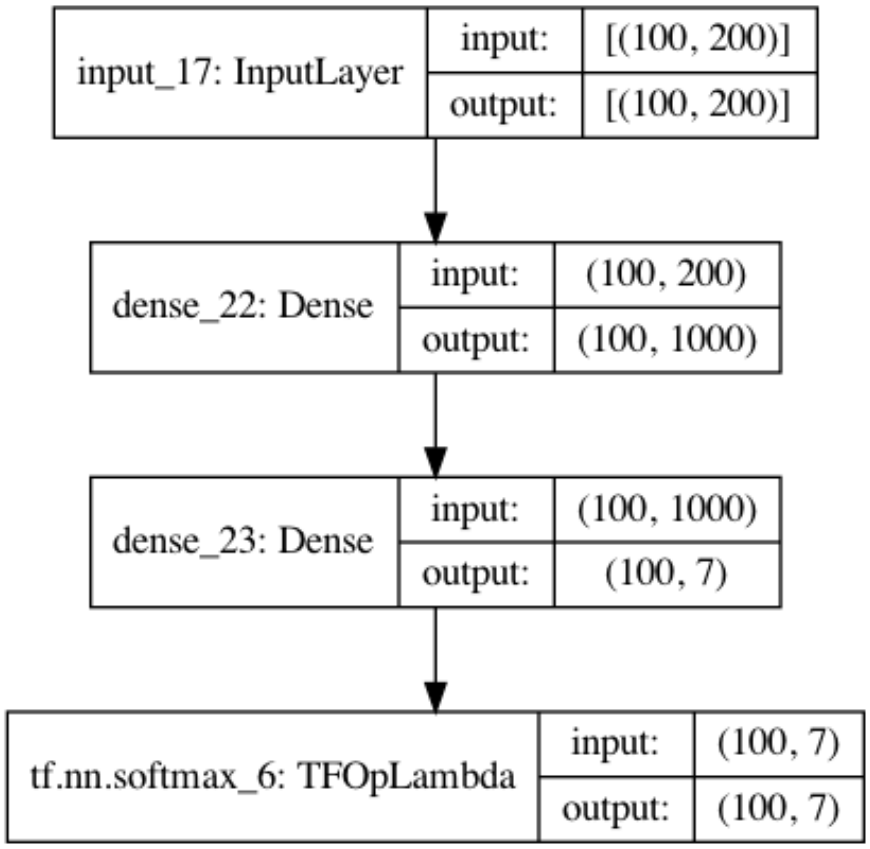


全连接层

第一层全连接层使用`Relu()`作为激活函数，输出向量长度为128；第二层全连接层使用`softmax()`解决分类问题，输出长为7的向量。

3）Baseline：MLP

用效果更差的全连接层做比对。中间层的参数设置为了1000。全连接层比较简单，不赘述。



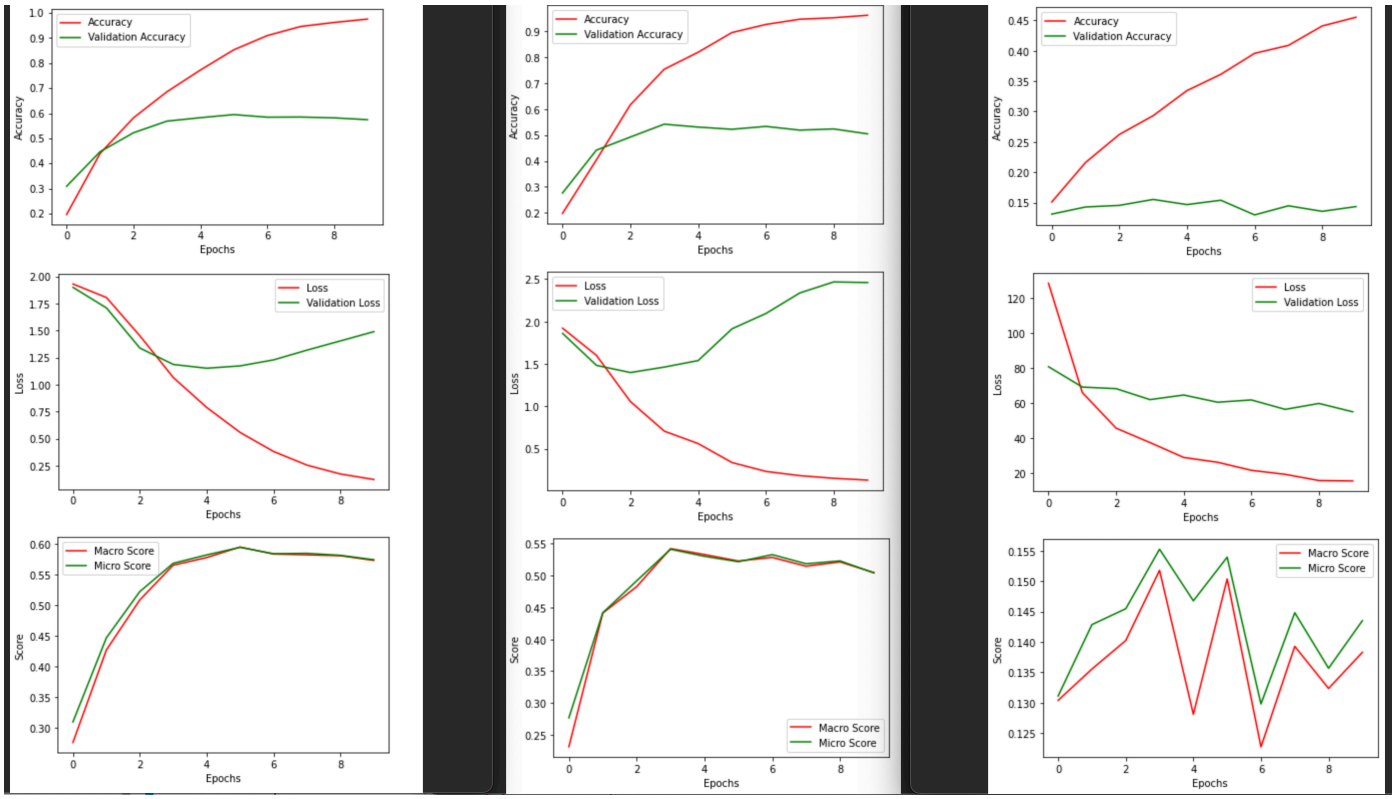
二、实验结果

	准确率	宏平均	微平均
CNN	59.49%	0.5954	0.5949
RNN	54.14%	0.5424	0.5414
MLP	15.53%	0.1518	0.1553

实验结果如下：

从上到下依次为准确率、训练损失、F-score

从左到右依次为cnn,rnn与mlp



三、参数调节

首先说明如何根据loss信息判断学习状态。

> 1、train loss 不断下降，test loss不断下降，说明网络仍在学习。
> 2、train loss 不断下降，test loss趋于不变，说明网络过拟合。
> 3、train loss 趋于不变，test loss不断下降，说明数据集有问题。
> 4、train loss 趋于不变，test loss趋于不变，说明学习遇到瓶颈，需要减小学习率或批量数目。
> 5、train loss 不断上升，test loss不断上升，说明网络结构设计不当，训练超参数设置不当，数据集经过清洗等问题。

由于选择adam优化器，所以在实验过程中不需要调节learning_rate/momentum等参数。数据集也是给定的无法调整。所以我们要做的是尽可能提高准确率，同时防止过拟合。

对于**CNN**，主要调节的参数有几个：词向量长度、卷积核大小、卷积核频道数。实验中发现卷积核数量对实验结果的影响较大，在数量较少时，准确率较低。分析原因是因为频道数量少时学习到文本特征不完全，对于准确率有较大影响。词向量长度与卷积核大小我在实验过程中动态调整为了50、[1, 2, 3, 4]。

对于**RNN**，情况类似。主要的参数是LSTM中隐藏状态的数量，隐藏状态数量少时对句子信息的记忆不够充分，会导致学习效果差。

四、差异比较

参数较少的MLP实验效果较差。

实验过程发现MLP训练速度很快。根据`二、实验结果`可知，MLP在拟合早期即出现过拟合情况，表现为曲线起伏。这是因为MLP只是简单的将数据输入拼接，不能学习到数据空间信息，也不能记忆历史信息，所以实验结果很差。

最终结果显示，CNN/RNN的实验准确率是MLP的4倍左右。

五、 问题思考

1. 如何控制好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。

受数据集限制，在实验训练早期数据集就会出现明显过拟合。我选择记录明显出现过拟合现象的训练轮次，并以此为训练轮次上限。

固定迭代次数关键在于对于迭代次数的选取：如果选取次数过小，无法排出训练效果短暂变差后继续变好的可能；如果迭代次数过大，会导致训练时间不必要的浪费和过拟合的出现。

早停法主要是训练时间和泛化错误之间的权衡。一定程度上有效避免网络过拟合，保证模型准确率。

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。

过拟合

- 正则化：在损失函数中加入L1范数或者L2范数、dropout、早停、数据增强和标签平滑等
- 数据扩增：从数据源头获取更多数据、通过一定规则扩充数据、根据当前数据集估计数据分布参数等

欠拟合：

- 添加多项式特征：如将线性模型通过添加二次项或者三次项
- 使用非线性模型：核SVM 、决策树模型
- 增加新特征：可以考虑加入进特征组合、高次特征，来增大假设空间
- 减少正则化参数：正则化的目的是用来防止过拟合的，但是模型出现了欠拟合，则需要减少正则化参数
- 调整模型的容量(capacity)

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。

二者本质相同：目前神经网络基于反向传播的优化算法来指导网络权重的优化。而当前的链式规则会导致随着层数的加深，梯度呈指数级变化。在反向传播过程中需要对激活函数进行求导：如果导数大于1，那么随着网络层数的增加梯度更新将会朝着指数爆炸的方式增加，这就是梯度爆炸。如果导数小于1，随着网络层数的增加梯度更新信息会朝着指数衰减的方式减少，即梯度消失。

解决办法

梯度消失：

- 减小网络高度
- 更换激活函数如relu

梯度爆炸：

- 更新梯度时对梯度设置一个阈值以防止梯度爆炸
- 权重正则化：对网络权重做正则来限制过拟合
- 更换激活函数如relu

4. 试分析CNN，RNN，MLP三者的优缺点与各自适用的场景

	优点	缺点	应用场景
CNN	自动进行特征提取、轻松处理高维数据	池化层导致信息丢失、卷积层缺乏平移不变性	文字处理，图像处理，视频处理
RNN	可以利用历史信息	层数增多时速度慢、长时依赖	自然语言处理、语音识别、手写体识别
MLP	训练速度快	效果一般	分类任务(数字识别)

六、 心得体会

这次实验给我们提供近距离体会“炼丹”的过程，是一次结合书本知识与实际运用的过程。通过实操，我体会了CNN、RNN、MLP三种模型的优缺点，对于神经网络有了具体概念。在实验过程中，也实际碰到了过拟合等问题的解决办法。

同时，我很佩服这些为后人“植树”的开发者，他们开发出一套好用的工具简化后人的工作。在前人的基础之上，训练神经网络就类似于一个拼装的过程。敬佩之情油然而生。

最后，虽然这次实验的正确率只有60%左右，但还是很感谢老师与助教的辛勤付出。