**作业报告**

**一、处理流程**

系统环境：Ubuntu-16.04

Python版本：python 2.7

1.原始数据：train\_data, valid\_data, test\_data （原始数据中每一行是一个dictionary，含有两个条目:”label”和:”text”，分别对应标签和文本）

2.预处理数据：train\_text, train\_label, valid\_text, valid\_label, test\_text, test\_label （即简单的将原数据中的标签和文本分开存储，一行代表一篇文章）[对应python代码是preprocess.py]

3.再次处理数据：train\_coefs, valid\_coefs, test\_coefs（这里读取了glove已经训练好词向量数据，将glove文档中的单词进行标号，然后读取以上第②步所得到的\*\_text文档，分句后为每一句生成一个list，list元素是句中单词的标号）[对应python代码是word\_to\_coefs.py, load\_glove.py,其中word\_to\_coef.py是主程序，load\_glove.py负责定义一个读取glove的词向量文档]

4.训练数据：my\_model.h5 （读取第③步得到的train\_coefs和第②步得到的train\_label文档，放入深度学习模型中进行训练，保存训练模型）[对应python代码是train.py, read\_data.py, load\_glove.py,其中train.py是主程序，read\_data.py负责读取之前处理好的数据到内存中]

5.预测数据：predict（加载第④步得到的my\_model.h5，读取第②步和第③步的到的valid\_coefs, test\_coefs, valid\_label文档进行预测，并分析验证集上的准确率，predict存储着测试集的预测结果）[对应的python程序是predict.py,read\_data.py,其中predict.py是主程序]

[关于测试集上的预测结果文件predict：每行为一个数字(0或1)加上’\n’]

**二、提取特征**

对一篇段落的特征提取很简单，提取结果是一个刻画段落特征的list，list的每个元素是刻画句子特征的list，代表句子的list的元素是句子中单词的标号。例如一个段落的特征是：[[1, 2], [3, 4]]表示该段落有两个句子[1, 2], [3, 4],第一个句子中的单词是标号为1的单词和标号为2的单词。

提取该特征的时候必然要考虑文档的段落中句子数、句子中单词数的分布情况，这里用analyze.py程序读取了所有数据，对每个段落进行分句，计算每个段落的句子数分布情况，对每个句子进行分词，计算每个句子的单词数分布情况。最终得出结论是90%的段落包含句子数不多于16，90%的句子包含单词数不多于40。因此在word\_to\_coef.py程序的特征提取处理中，对每个段落只提取了前16个句子，每个句子又只提取前40个单词。

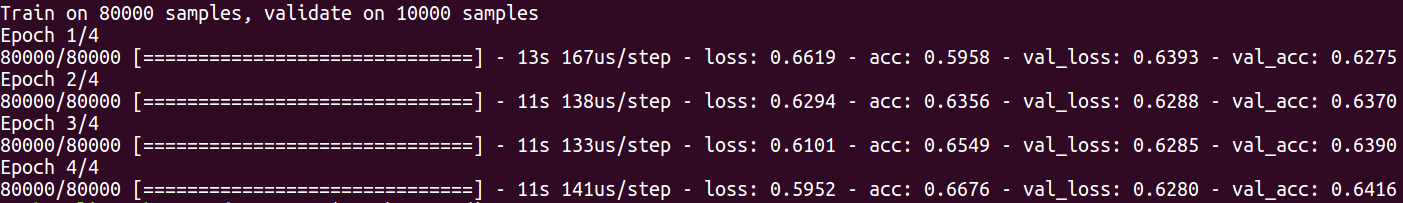
**三、模型框架**

这里采用了Keras深度学习框架。输入层是数组x，x包含四个数组，代表一篇段落的四个句子，每个数组中的元素是一个句子中单词的标号。输出层is\_coherent只有一个单元，激活函数是sigmoid，故取值范围是0~1，这描述了相似程度，可以代表预测标签（四舍五入后为0或1）。

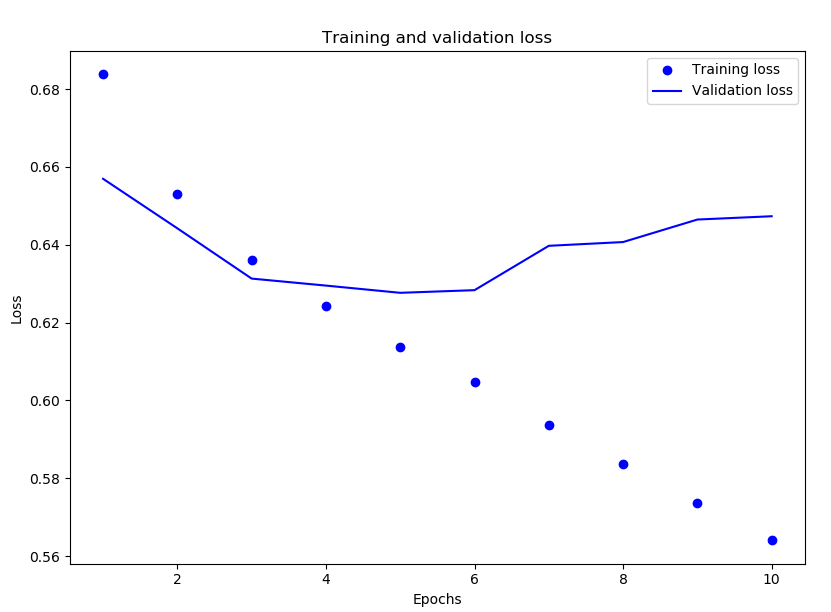
除了输入层和输出层，中间包含8个隐藏层。第一个隐藏层有4个节点，是嵌入层embedding，它将4个句子中单词的标号映射为单词的词向量；第二个隐藏层有4个节点，是卷积层，它对embedding每个节点分别进行一维卷积，激活函数是relu；第三个隐藏层有4个节点，是池化层，它对卷积层每个节点分别进行全局最大池化；第四个隐藏层有3个节点，是拼接层，它将第一、二句子，第二、三句子，第三、四句子，拼接起来，借此以刻画相邻句子间的关系。第五个隐藏层有3个节点，是全连接层，它对第四个隐藏层每个节点都进行全连接处理，激活函数是relu。第六个隐藏层是拼接层，有1个节点，它将第五层的节点都拼接起来。第七个隐藏层是全连接层，它对第六个隐藏层的节点进行全连接处理，激活函数是relu。第八个隐藏层是dropout层，有1个节点，它用来进行正则化处理，目的是减少过拟合。

**四、结果分析**

代码最终对每篇段落只取了4个句子，每个句子只取5个单词，这是为了减少训练时间和减少过拟合。训练结果如下图所示：



验证集上的正确率是64.16%，而在训练过程中训练集相应的正确率是66.76%。如果采取10个epoch，则训练集和验证集的损失函数值的结果如下：



这说明模型在训练的过程中已经产生了过拟合，如果继续训练下去反而会降低验证集上的正确率。

在减少过拟合的过程中，采取了减少特征数（即减少提取句子数目、减少每个句子提取单词的数目）、以及增加Dropout率等尝试，但是最终训练的验证集正确率上限都是64%左右，并且依然达到了过拟合的结果。