语句情感分类模型报告

姓名：陈莉

环境：python3.6, tensorflow-gpu==1.12.0, keras==2.2.4

一 模型 1

预处理：

（1） 读入所有数据，通过正则表达式将句子中常见的带有缩写的英语转换成独立单词，如句子中的 don't 拆分成 do not，并将所有的标点符号去除。

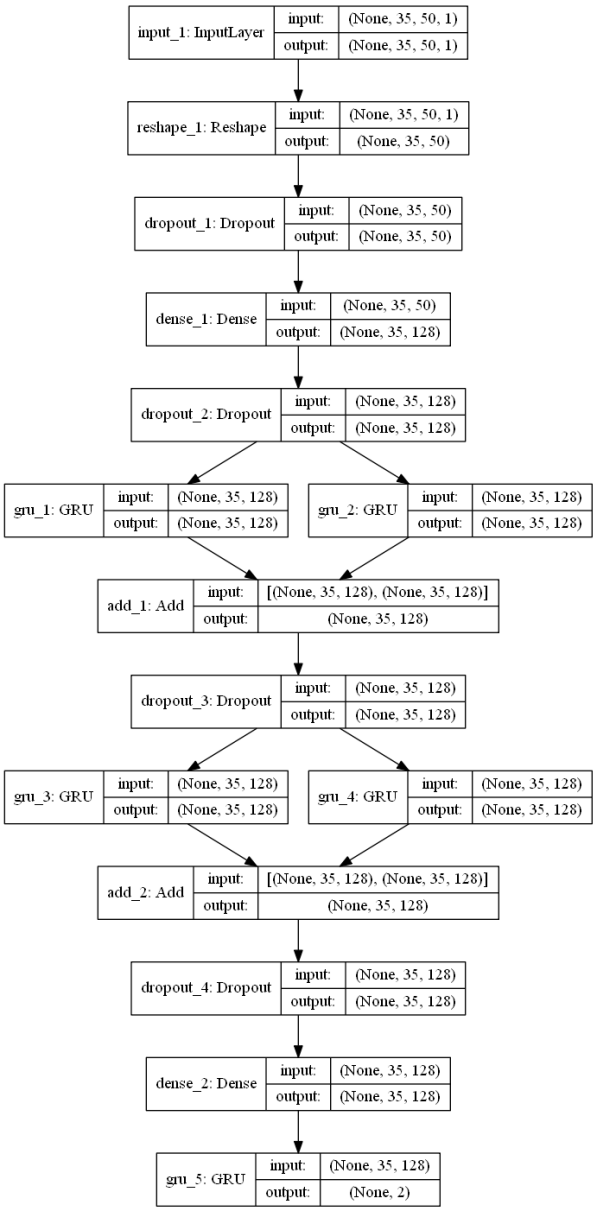
（2） 载入预训练好的词向量字典，将句子中的每一个字 embedding 成对应字典中的向量在本次作业中使用了 GloVe(Global Vectors for Word Representation)库中的 Wikipedia 2014 + Gigaword 5 的 50 维字典库其中包含有 60 亿个 tokens。

（3） 将输入句子进行 padding 操作，经过统计发现超过 99%的训练、测试句子单词数量少于 35，因此设定句子长度为 35，长于 35 个单词的句子进行裁切选择前 35 个单词作为输入。根据文献表示在句子前进行补 0 操作比在句子后面进行补 0 操作在 RNN 网络中的识别率更高，因此短于 35 个单词的句子在句子前面进行 padding 操作，padding 的向量为 50 维的 0 向量，输出的句子的维度是(35, 50 ,1)。

（4） 读取有标签数据的标签，将标签进行 one_hot 格式处理。

建模：

在本次作业中两个模型都是用的是 Bi-GRU 网络，在尽可能减少模型参数的前提下保证句子前后关系可以被学习到，模型结构参数如图所示，使用 Adam 梯度下降算法进行优化，初始学习率设置为 0.05，一阶矩估计的指数衰减率设置为 0.9，二阶矩估计的指数衰减率设置为 0.999，学习率衰减系数设置为 0.01。损失函数使用交叉熵损失函数，分类器使用 SoftMax 分类器输出概率。所有激活函数使用 relu 函数。



训练：

首先将有标签数据进行训练，batch 设置为 20，输入的数据的格式为(batch,sentence_maxlen,word_dim,1)即(20,35,50,1)。Epoch 大小设置为 20，将所有带标签数据打乱训练，每个 epoch 需要 25 分钟训练。最终训练精度约为 0.7 并保存。

识别：

将没有标签的训练数据传入训练好的模型中，每个 batch 传入 20 条句子，输出为标签(0,1)的概率分布，将其与向量(0,1)相乘可以获得句子情感为积极的概率，当模型判断句子为积极的概率大于 0.7 时则认为它是积极情感的句子，写入 1。

再训练：

将没有标签的数据集写入标签后同初训练相似，使用相同的优化算法，batch 大小和损失函数对现有的模型进行训练，这里的 epoch 设置为 10，每次学习时间约为 2 小时。最终准确率可以达到 0.85。

再识别：

将使用再训练后保存的模型对没有标签的数据进行再次识别，当识别概率大于 0.5 时认为句子情感为积极，输出 1，否则输出 0 即完成作业。下图是模型参数量。

```
Total params: 418,578
Trainable params: 418,578
Non-trainable params: 0
```

二 模型 2

预处理：

（1） 读入所有数据，载入预训练分词模型库 tokenizer_lstm.pickle，这是一个开源的基于双层 LSTM 网络训练好的的分词工具包，利用 tokenizer 对所有句子进行分词，产生一个所有此的词典，这里只统计出现频率最高的前 20000 个词。

（2） 将句子中的词兑换成词典中的索引。

（3） 将输入句子进行 padding 操作，这里设定句子长度为 50，长于 50 个单词的句子进行裁切选择前 50 个单词作为输入。同样的短于 50 个单词的句子在句子前面进行 padding 操作，padding 后输出句子维度是(maxlen, indux_in_dic)即(50,1)。

（4） 读取有标签数据的标签。

建模：

模型 2 中输入维度和特征较少，可以构建较为简单的神经网络，如图所示使用 Adam 梯度下降算法进行优化，初始学习率设置为 0.01，一阶矩估计的指数衰减率设置为 0.9，二阶矩估计的指数衰减率设置为 0.999，没有学习率衰减。损失函数使用二进制交叉熵损失函数，分类器使用全连接层用激活函数 sigmoid 分类器输出情感类别。其他激活函数使用 relu 函数。

初训练：

此模型参数较少，特征较少，训练速度较快，训练时使用顺序训练，使用最后 2000 条带标签数据作为验证集验证，batch 设置为 500，epoch 设置为 10，最终模型精度达到 0.87，损失值为 0.06，验证集精度为 0.80，有过拟合现象出现。

识别：

将没有标签的训练数据依次传入训练好的模型中，输出结果为 0 或 1 分类，不能设置阈值，将结果保存。

再训练：

将没有标签的数据同识别中产生的标签一同放入初训练好的模型中进行再训练，训练设置 epoch 为 10 次，保留 1/10 的数据作为验证集测试，最终训练集精度为 0.95，验证集精度为 0.93，效果较好。

再识别：

将使用再训练后保存的模型对没有标签的数据进行再次识别，输出结果即为 2 值分类，即完成作业。右图是模型参数量。