# Homework 4

陈春祺 5140379022

## Problem 1

###### 参数设置

使用TensorFlow实现了一个单层的LSTM模型，参数设置如下：

**LSTM cell**：tf.nn.rnn\_cell.BasicLSTMCell

**输入层节点数**：310

**隐藏层节点数**：256

**输出层节点数**：3

**Time step**：5

**损失函数**：预测值与真实值的交叉熵

**Batch size**：由于数据量较少，所以一次训练输入全部的数据作为一个Batch；由于15段数据最短的为185秒，所以统一取每段数据的前面185秒，而time step为5，所以每段数据被分为37条子数据，每条子数据包含5个长度为310的向量；故9段训练数据共有333条子数据，6段测试数据共有222条子数据，训练时的batch size为333，测试时batch size为222，最终的训练数据维度为333\*5\*310，测试数据维度为222\*5\*310.

**学习率**：0.01

**训练轮数**：1000

**参数初始化**：节点权值随机初始化，偏移值初始化为0.1

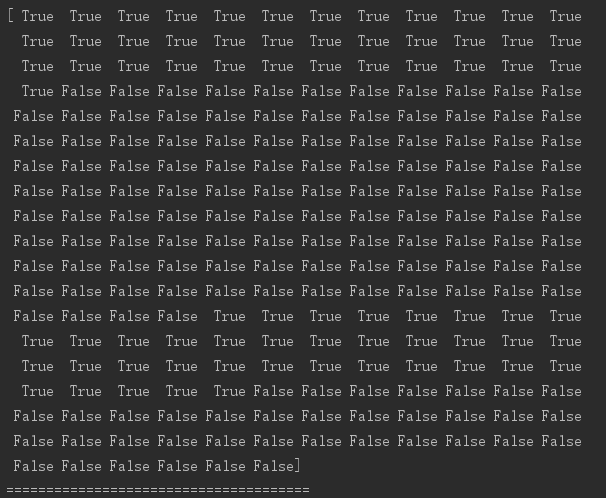
###### 训练数据：01.npz

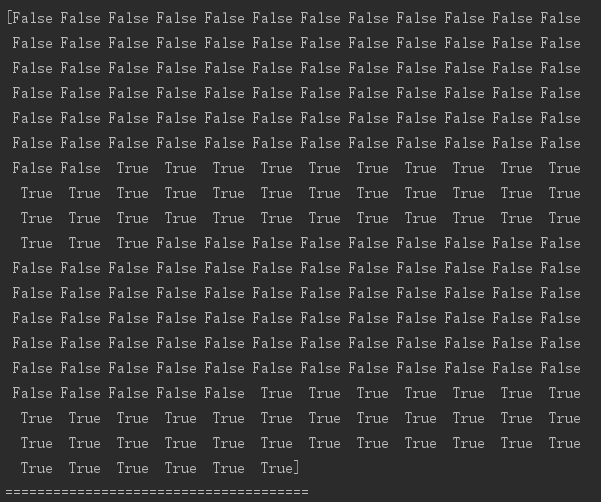
###### 训练数据：02.npz

###### 训练数据：03.npz

###### 结果分析

可以看出，在三份数据上进行的训练和测试结果基本一致，损失度的下降和准确率等等基本是相同的结果，损失度收敛很快，在50轮左右就趋近收敛，但在1左右的时候就会停止继续下降，所以准确度也基本上全程保持不变，因为之后损失度就不会在下降了。其中数据集1和2的预测准确度在0.33，或者说非常准确的三分之一，而数据集3预测准确率在0.5，或者说非常准确的二分之一，为什么会出现这样的情况呢？为了寻找原因，选择了训练1轮以后和训练50轮以后的两次预测结果来比较，如下图所示，第一张图为训练1轮以后的结果，第二张为训练50轮以后的结果：





可以看到，由于原始的每段数据都被分为了37段子数据，而图中每次预测，来自同一段数据的37段子数据都得到相同的预测结果，而原始测试数据只有6段，所以测试准确率就会是1/6、2/6、3/6、4/6、5/6、6/6。尝试修改学习率、隐藏层节点数、参数初始化方式、time step，都没有太大的影响。据推测，之所以会出现这样的结果，可能是因为数据量太少，导致训练得到的模型缺少泛化性，所以效果较差。

## Problem 2

这一题需要结合所有数据训练一个通用的模型，所以将上一题使用的数据集1、2、3和对应的label连接起来作为lstm模型的输入，其中训练数据的大小为999\*5\*310，测试数据的大小为666\*5\*310，其余参数不变。

结果如下图所示：

可以看出来，将所有数据连接起来训练，仍然得到与上一题相似的结果，准确率几乎稳定在0.33，推测原因也与之前的情况一致，数据量太少，缺少泛化性。

###### 代码

代码文件为lstm.py;

各种超参数的定义在文件开始处；

节点权值、偏移值定义在weights、biases；

Lstm的结构定义在函数rnn(X, weights, biases, batch\_size)；

第一题的运行过程定义在函数run\_lstm();

第二题的运行过程定义在函数run\_lstm\_concat();