Statistical Theories for Brain and Parallel Computing

Assignment No.7

17M18819

ZHU LICHENG

## Description of Assignment

In assignment 7, I need to construct a probabilistic and binary model and apply gradient descent to calculate the weights of the RNN instead of calculating output given weights in the previous assignments.

The stochastic phenomenon that generates three binary values and probability is given below

|  |  |  |  |
| --- | --- | --- | --- |
| X1 | X2 | X3 | q(x1,x2,x3) |
| 0 | 0 | 0 | 0.25 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0.25 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0.25 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0.25 |

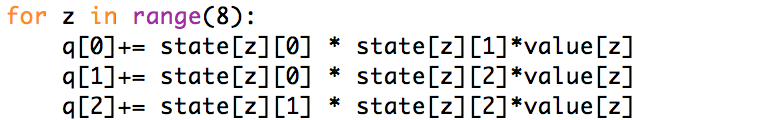
## Solution

In this assignment, we need to apply gradient descent to reduce the error between estimated value from our RNN and real value. I will list the procedure of learning as follows:

1. Give initial values to weights. (In our case, we only have three weights W12,W13 and W23)
2. For every n and m, repeat updating this RNN by the order of X1🡪X2🡪X3 for 1000 times and compute the average of Xn\*Xm.
3. Then we can update Wnm = Wnm – \*alpha\*(p-q) where p(mn) = and q(mn) = .
4. Repeat 2 and 3 until G = is small enough.

In my program, I assigned the initial value of my weights as [0,0,0] which are W12, W13, W23 separately.

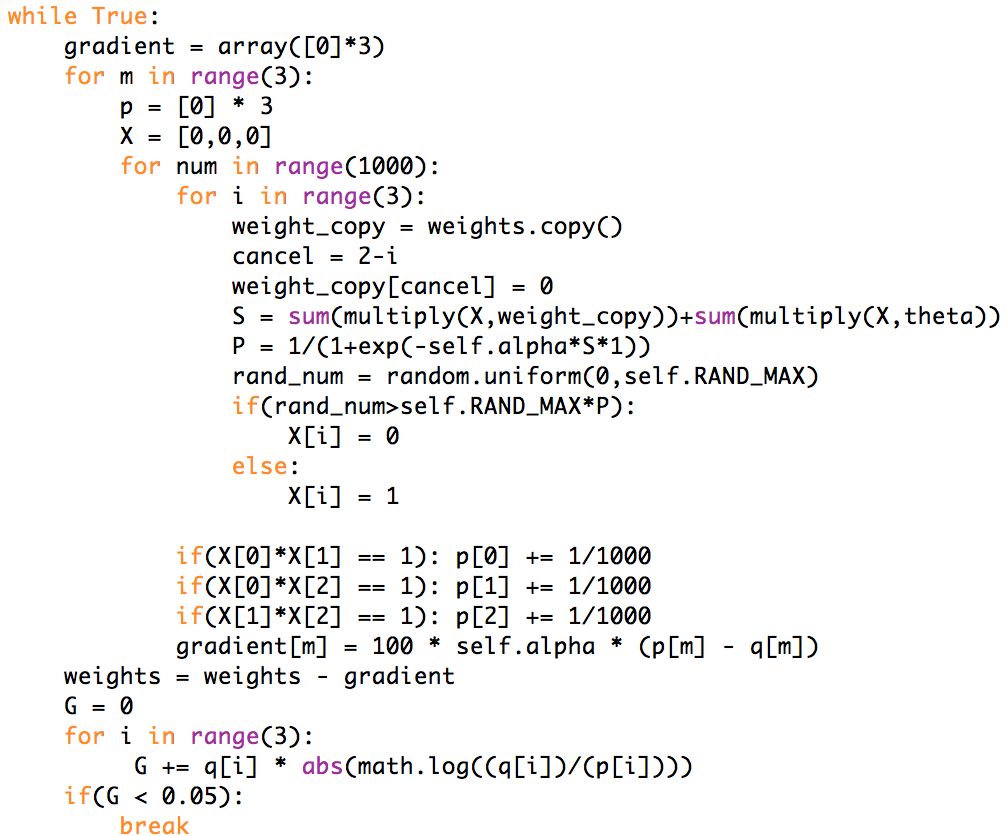
In the following graph, I calculated the value of q for every m and n



I repeated calculating RNN until G is smaller than 0.05. 0.05 is observed manually by myself since I found that when G is smaller than 0.05, p would be very close to q:

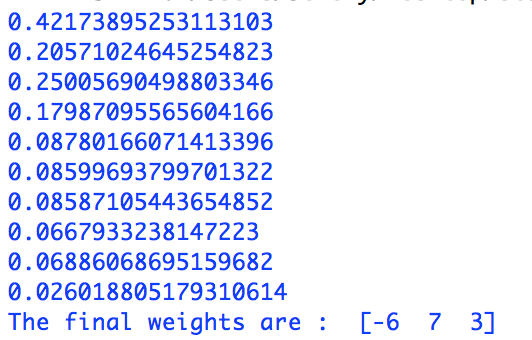


The following graph shows the updating and ending condition of the procedure of learning:



***Result***

Here I output the value of G every time the W is updated until G is smaller than 0.05



Therefore, the final weights are [-6, 7, 3] for W12, W13, W23 separately although actually this answer could fluctuate a little bit, but this should be precise enough to mimic the stochastic phenomenon.

## Running command

**ass7().gen\_weight()**