Programming Assignment Summary Report
Machine Learning
CSCI-B 555 Spring 2017

Rahul Raghatate

April 25, 2017

Problem Statement

Implement a two-layer perceptron with the backpropagation algorithm to solve the parity problem.

Parity Problem definition:

Desired output =1 if (Nos. of 1’s in input pattern are odd)

= 0 otherwise

Implementation

Neural Net:

Input Layer

One Hidden layer - 4 neurons

Output layer – 1 neuron

Biases are considered only for the input layer.

Activation Function:

Sigmoid Function with a = 1 is used.

INPUT Patterns:

```
  [[0,0,0,0,1],
            [0,0,0,1,1],
        [0,0,1,0,1],
        [0,0,1,1,1],
        [0,1,0,0,1],
        [0,1,0,1,1],
        [0,1,1,0,1],
        [0,1,1,1,1],

        [1,0,0,0,1],

        [1,0,0,1,1],

        [1,0,1,0,1],

        [1,0,1,1,1],

        [1,1,0,0,1],

        [1,1,0,1,1],

        [1,1,1,0,1],

        [1,1,1,1,1]]
```

This is the input data matrix; the input has 4 binary bits.

Bias 1 added at end therefore total 5 bits for the input and total of 16 training examples.
Input:(16*5)

wt_l1 is the weights at hidden layers (synapse from input layer to hidden layer).

wt_l1 is a matrix of size (5*4). In this matrix, the first row is the weights of the bias for respective neurons in hidden layer.

xw1: is the input at hidden layer neuron before the activation is applied. It is matrix of size (16*4) where each row is the input on the specific neuron for given training examples.

l1_o: It is the output of hidden layer neurons after applying the activation. It is matrix of size (16*4).

l1w2: is the input at second layer neuron before the activation is applied.

wt_l2: is weight at output layer (synapse from hidden layer to output layer). There are 4 neurons in hidden layer and one output layer, Hence wt_l2 is a matrix of size (4*1).

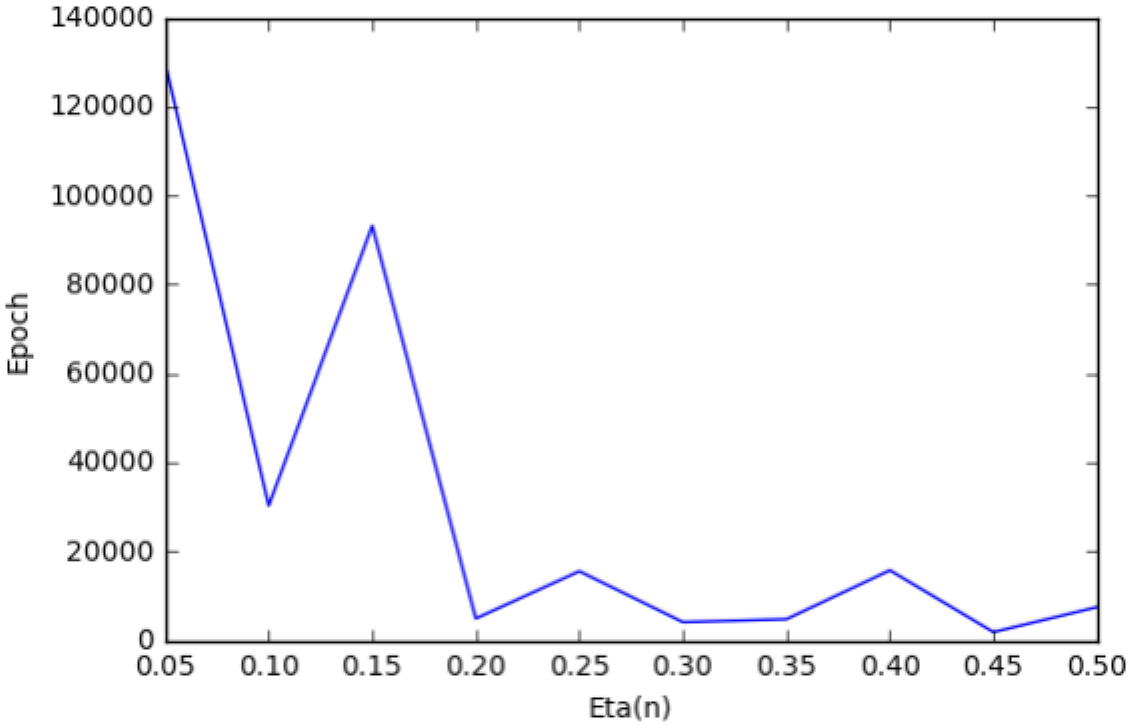
l2_o:
yhat is the output of the output layer and it is of the size (16*1)
d_o:
This is the desired output for the 16 training examples

Execution and Results

Vary the value of η from 0.05 to 0.5 with increment 0.05, and report the number of epochs for each choice of η

Include a momentum term in weight update with = 0.9 and report its effect on the speed

of training for each value of η



Conclusion:

The graph clearly shows that number of epochs decreases as learning rate increases. But as learning rate increases, there is a chance that the system might not converge as error might keep oscillating around the minima.The momentum term is not only added to avoid local minima, but higher value of momentum, will increase the speed of convergence of system. However, setting the momentum parameter too high can create a risk of overshooting the minimum, which can cause the system to become unstable.

Attachments

Please find attached code.