

Assignment 3

Perceptron Learning Algorithm (PLA)

Please go through the complete assignment problem statement and wiki pages.

Part 1

- Implement a **perceptron** with following requirements (<https://en.wikipedia.org/wiki/Perceptron>)-
 - Two input functions (x1 and x2)
 - Weights (w0, w1 and w2)
 - w0 is bias
- Use the **sigmoid** function as the activation function. (Since the output node uses a sigmoid activation function that returns a number between 0 and 1, associate any output greater than 0.5 to one class and any output less than or equal to 0.5 to the other class)

$$g(h) = \frac{1}{1+e^{-h}}$$

- Calculate the error using following formula -

$$\sum_{j=0}^n (t_j - h_j)^2$$

- Use the the Delta rule to update the weights (wi) . Here is the link to wiki page on Delta Rule https://en.wikipedia.org/wiki/Delta_rule

$$\Delta w_{ji} = \alpha (t_j - y_j) g'(h_j) x_i$$

- Derivative of sigmoid function-

$$g'(h) = g(h)(1-g(h))$$

- Train and test the neural network with random dataset.
- Iterate the PLA for at least 1000 iterations. (this is **epoc**)
- Use the value of **learning rate** as 0.001.
- Here is the algorithm -

```

loop until error < 0.0001 or 1000 iterates
    delta weight = 0
    weights = random(0,1) // w0 = 1, w1, w2
    loop through each training instance
        calculate delta weight
        apply delta weight to weights

```

- Create a graph of epoc (x axis) vs. average error (y-axis)
- Use neural network to learn the function and test it using the **10 fold cross validation method**.

Part 2 -

- Train the two input perceptron to build an AND, OR, NAND, NOR. Also, verify that XOR cannot be trained using perceptron since it is a non linear separable function.