Perceptron Learning

BY

DR. ANUPAM GHOSH

Perceptron-Learning

- The input vectors are allowed to be either binary or bipolar. However, the outputs must be in bipolar form
- The bias W0 is adjustable but the threshold θ used in the activation function is fixed
- Learning takes place only when the computed output does not match the target output.
- The threshold θ of the activation function may be interpreted as a separating band of width 2θbetween the region of positive response and negative response.
- The band separating the regions of positive response from that of the negative response is defined by the pair of lines

$$w_0x_0 + w_1x_1 + w_2x_2 = \theta$$

$$w_0x_0 + w_1x_1 + w_2x_2 = -\theta$$

Algorithm

Procedure *Perceptron-Learning*

Step 1. Initialize all weights, $w_0, \ldots, w_m$.

Step 2. Set learning rate η such that $0 < \eta \le 1$, and threshold θ.

Step 3. For each training pair $s : t$ do Steps 4–8.

Step 4. Activate the input units, $x_i = s_i$, for $i = 0, \ldots, m$.

Step 5. Compute the net input to the output unit

$$y\_in = \sum_{i=0}^{m} w_i x_i$$

Step 6. Compute the activation of the output unit using the function

$$y\_out = \begin{cases} 1, & \text{if } y\_in > \theta \\ 0, & \text{if } -\theta \le y\_in \le \theta \\ -1, & \text{if } y\_in < -\theta \end{cases}$$

Step 7. If there is an error, *i.e.*, $y\_out \ne t$, then adjust the weights as follows

$$w_i \text{ (new)} = w_i \text{ (old)} + \eta \times t \times x_i$$

If, however, no error has occurred, the weights are kept unchanged.

Step 8. If there were no error, *i.e.*, $y\_out = t$, for the entire set of training pairs, then stop. Otherwise go to Step 3.

Perceptron Training Rule

- Weights modified for each example
- Update Rule:

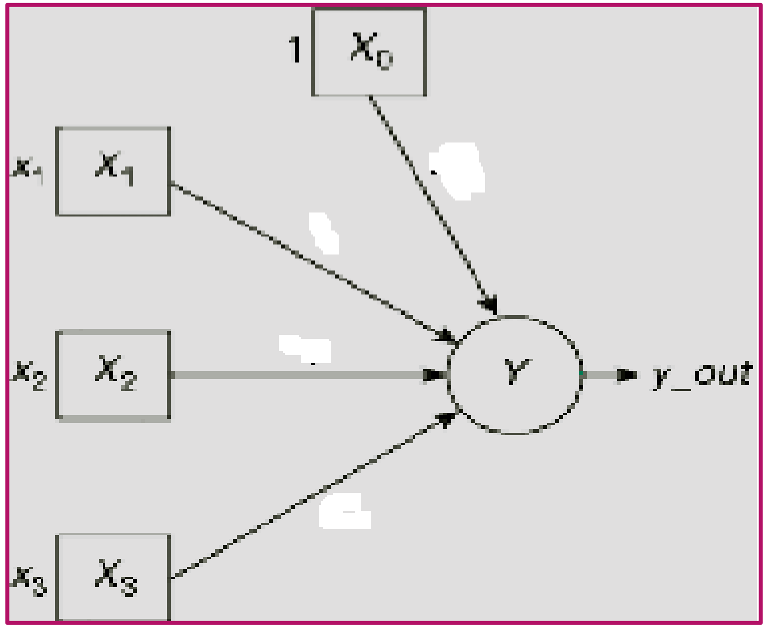
$$w_i \leftarrow w_i + \Delta w_i$$

where

$$\Delta w_i = \eta(t - o)x_i$$

learning rate | target value | perceptron output | input value

Sample Demonstration



Input pattern			Target output (t)
x_1	x_2	x_3	
–1	1	1	–1
1	–1	1	–1
1	1	–1	–1
1	1	1	1

the initial weights are all kept at 0 and the learning rate is set to $\eta = 1$. Both the inputs and the outputs are presented in bipolar form.

#	Input				Net input	Out-put	Tar-get	Weight adjustments				Weights			
	x_0	x_1	x_2	x_3	y_in	y_out	t	Δw_0	Δw_1	Δw_2	Δw_3	w_0	w_1	w_2	w_3
0												0	0	0	0
1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1
2	1	−1	1	1	2	1	−1	−1	1	−1	−1	0	2	0	0
3	1	1	−1	1	2	1	−1	−1	−1	1	−1	−1	1	1	−1
4	1	1	1	−1	2	1	−1	−1	−1	−1	1	−2	0	0	0

Epoch #1

#	Input				Net input	Out-put	Tar-get	Weight adjustments				Weights			
	x_0	x_1	x_2	x_3	y_in	y_out	t	Δw_0	Δw_1	Δw_2	Δw_3	w_0	w_1	w_2	w_3
0												−2	0	0	0
1	1	1	1	1	−2	−1	1	1	1	1	1	−1	1	1	1
2	1	−1	1	1	0	0	−1	−1	1	−1	−1	−2	2	0	0
3	1	1	−1	1	0	0	−1	−1	−1	1	−1	−3	1	1	−1
4	1	1	1	−1	0	0	−1	−1	−1	−1	1	−4	0	0	0

Epoch #2

#	Input				Net input	Out- put	Tar- get	Weight adjustments				Weights			
	x_0	x_1	x_2	x_3	y_in	y_out	t	Δw_0	Δw_1	Δw_2	Δw_3	w_0	w_1	w_2	w_3
0												−4	0	0	0
1	1	1	1	1	−4	−1	1	1	1	1	1	−3	1	1	1
2	1	−1	1	1	−2	−1	−1	0	0	0	0	−3	1	1	1
3	1	1	−1	1	−2	−1	−1	0	0	0	0	−3	1	1	1
4	1	1	1	−1	−2	−1	−1	0	0	0	0	−3	1	1	1

Epoch #3

#	Input				Net input	Output	Target	Weight adjustments				Weights			
	x_0	x_1	x_2	x_3	y_in	y_out	t	Δw_0	Δw_1	Δw_2	Δw_3	w_0	w_1	w_2	w_3
0												−3	1	1	1
1	1	1	1	1	0	0	1	1	1	1	1	−2	2	2	2
2	1	−1	1	1	0	0	−1	−1	1	−1	−1	−3	3	1	1
3	1	1	−1	1	0	0	−1	−1	−1	1	−1	−4	2	2	0
4	1	1	1	−1	0	0	−1	−1	−1	−1	1	−5	1	1	1

Epoch #4

#	Input				Net input	Out-put	Tar-get	Weight adjustments				Weights			
	x_0	x_1	x_2	x_3	y_in	y_out	t	Δw_0	Δw_1	Δw_2	Δw_3	w_0	w_1	w_2	w_3
0												−5	1	1	1
1	1	1	1	1	−2	−1	1	1	1	1	1	−4	2	2	2
2	1	−1	1	1	−2	−1	−1	−1	1	−1	−1	−4	2	2	2
3	1	1	−1	1	−2	−1	−1	−1	−1	1	−1	−4	2	2	2
4	1	1	1	−1	−2	−1	−1	−1	−1	−1	1	−4	2	2	2

Epoch #5

#	Input				Net input	Out-put	Tar-get	Weight adjustments				Weights			
	x_0	x_1	x_2	x_3	y_in	y_out	t	Δw_0	Δw_1	Δw_2	Δw_3	w_0	w_1	w_2	w_3
0												−4	2	2	2
1	1	1	1	1	2	1	1	0	0	0	0	−4	2	2	2
2	1	−1	1	1	−2	−1	−1	0	0	0	0	−4	2	2	2
3	1	1	−1	1	−2	−1	−1	0	0	0	0	−4	2	2	2
4	1	1	1	−1	−2	−1	−1	0	0	0	0	−4	2	2	2

Epoch #6

It will be Continued....