VIMAL JYOTHI
ENGINEERING COLLEGE
JYOTHI NAGAR, CHEMPERI - 670632, KANNUR D.T., KERALA
An ISO 9001:2008 Certified Institution

Assignment Cover Page

Name of the student :- Nandana Sathyanandan

PRN ....VML22CS131........ Admission No....1982....

Subject Name :- Soft Computing | Subject Code:- CST444

Assignment Title/No : 1

Name of the faculty:- Asuathi .T.V

Assignment Submitted on ....20|2|26....

Late submission rules : Max mark will reduced to 50% for 1-5 working day's delay, no mark will be awarded thereafter.

I am hereby confirming that this assignment is my own and I haven't adopted any unfair means in any steps of its preparation to enhance my performance in this assignment.

Date :....20|2|26....

Nandana

Sign with Name

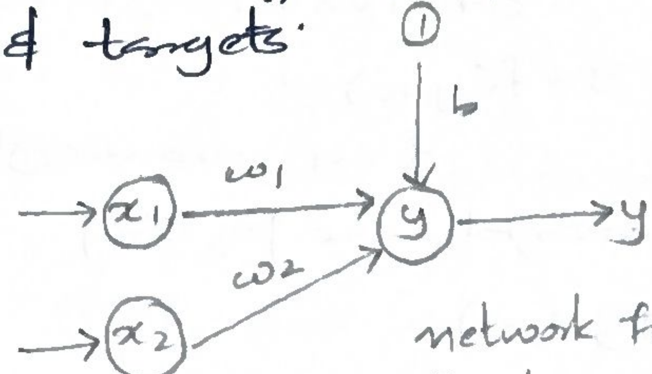
Assignment subdivision	Maximum Mark	Marks awarded	Remarks
A			
B			
C			

Feed back/suggestions :

Name and sign of the faculty

) Find the weights using the perceptron network for ANDNOT function when all the i/ps are presented only once. Use bipolar i/ps & targets.

x_1	x_2	t
1	1	-1
1	-1	1
-1	1	-1
-1	-1	-1



network for ANDNOT function

Let $\alpha = 1$ $\quad \theta = 0$

(1, 1, -1)

$y_{in} = b + x_1 w_1 + x_2 w_2 = 0 + 1 \times 0 + 1 \times 0 = 0$

Applying activation function over the net i/p:

$$y = f(y_{in}) = \begin{cases} 1 & \text{if } y_{in} > 0 \\ 0 & \text{if } -0 \le y_{in} \le 0 \\ -1 & \text{if } y_{in} < -0 \end{cases}$$

$w_1(new) = w_1\,old + \alpha t x_1 = 0 + 1 \times -1 \times 1 = -1$

$w_2(new) = -1$

$b\,new = b\,old + \alpha t = 0 + 1 \times -1 = -1$

$w = [-1, -1, -1]$

(1, -1, 1)

$y_{in} = b + x_1 w_1 + x_2 w_2$

$= -1 - 1 + 1 = -1$

the output $y = f(y_{in})$ $\quad y = -1 \quad t = 1 \quad y \neq t$

$w_1(new) = w_1\,old + \alpha t x_1 = -1 + 1 \times 1 \times 1 = 0$

$w_2(new) = w_2(old) + \alpha t x_2 = -1 + 1 \times 1 \times -1 = -2$

$b\,new = -1 + 1 \times 1 = 0$

$w = [0, -2, 0]$

(-1, 1, -1)

$w_1(new) = -1 + 1 \times 1 \times 1 = 0$

$w_2(new) = \cancel{-1} \times 1 \times \cancel{1} \times -1 = -2$

$b(new) = -1 + 1 \times 1 = 0$

$y_{in} = b + x_1 \omega_1 + x_2 \omega_2$

$= 0 + -1 \times 0 + 1 \times -2 = -2$

$y = f(y_{in}) = -1$

$t = y$ no weight changes

weights are $[0, -2, 0]$

$(-1, -1, -1)$

$y_{in} = 0 + -1 \times 0 + -1 \times -2 = 2$

$y = f(y_{in}) = 1 \quad t \neq y$

$\omega_1(new) = \omega_1(old) + \alpha t x_1 = 0 + 1 \times -1 \times -1 = 1$

$\omega_2(new) = \omega_2(old) + \alpha t x_2 = -2 + 1 \times 1 \times -1 = -1$

weights after presenting fourth i/p sample are

$\omega = [1, -1, 1]$

one epoch of training for ANDNOT function using

INPUT			Target	Net i/p	Calculated output	Weights	
x_1	x_2	b	(t)	y_{in}	(y)	ω_1	ω_2
						(0	0
1	1	1	-1	0	0	-1	-1
1	-1	1	1	-1	-1	0	-2
-1	1	1	-1	-2	-1	0	-2
-1	-1	1	-1	2	1	1	-1

8b) How is the training algorithm performed in back propagation neural network.

step 0: Initialize weights and learning rate

step 1: perform step 2-9 when stopping condition is false.

step 2: perform step 3-8 for each training pairs.

Feed forward phase (phase I):

step 3: Each i/p unit receives input signal x_i and sends it to the hidden unit $(i = 1$ to $n)$

step 4: Each hidden unit z_j $(j = 1$ to $p)$ sum, its weighted i/p signals to calculate net input.

$$z_{inj} = V_{oj} + \sum_{i=1}^{n} x_i v_{ij}$$

Calculate the output of the hidden unit by applying activation function.

$z_j = f(z_{inj})$

step 5: For each output unit y_k ($k=1$ to m)
calculate the net input:

$y_{ink} = w_{ok} + \sum_{j=1}^{p} z_j w_{jk}$ and apply the activation function to compute the output signal.

$y_k = f(y_{ink})$

Back propogation of error (Phase II):
step 6: Each output unit (y_k)($k=1$ to m) receives a target pattern corresponding to the input training pattern and computes the error correction term:

$\delta_k = (t_k - y_k) f'(y_{ink})$

Then update the weights and bias:

$\Delta w_{jk} = \alpha \delta_k z_j$

$\Delta w_{ok} = \alpha \delta_k$.

δ_k to the hidden layer backwards.

step 7: Each hidden unit z_j ($j=1$ to p) sums its delta inputs from the output units:

$\delta_{inj} = \sum_{k=1}^{m} \delta_k w_{jk}$

The term δ_{inj} gets multiplied with the derivative of $f(z_{inj})$ to calculate the error term:

$\delta_j = \delta_{inj} f'(z_{inj})$

Then update the weights and bias.

$\Delta v_{ij} = \alpha \delta_j x_i$ $\qquad \Delta v_{oj} = \alpha \delta_j$

Weight and bias updation (phase III)

step 8: Each output unit y_k($k=1$ to m) updates the bias and weights:

$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk}$

$w_{ok}(new) = w_{ok}(old) + \Delta w_{ok}$

Each output unit z_j ($j=1$ to p) updates the bias and weights:

$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij}$

$v_{oj}(new) = v_{oj}(old) + \Delta v_{oj}$

Step 9: Check for the stopping condition maybe certain number of epochs reached or when the actual output equals to target output.