Train a perceptron to classify according to:

- (4, 5) yes
- (6, 1) yes
- (4, 1) no
- (1, 2) no

There will be three weights (w_0, w_1, w_2) where is the w_0 threshold, corresponding to phantom input -1.

Start with "random" weights, say (0, +1, -1)

Choose $\eta = 1$.

EPOCH 1

weights	input	desired	actual	error	new weights
(0, 1, -1)	(-1, 4, 5)	yes	no	1	(-1, 5, 4)
(-1, 5, 4)	(-1, 6, 1)	yes	yes	0	no change
(-1, 5, 4)	(-1, 4, 1)	no	yes	-1	(0, 1, 3)
(0, 1, 3)	(-1, 1, 2)	no	yes	-1	(1, 0, 1)

EPOCH 2

weights	input	desired	actual	error	new weights
(1, 0, 1)	(-1, 4, 5)	yes	yes	0	no change
(1, 0, 1)	(-1, 6, 1)	yes	? no	1	(0, 6, 2)
(0, 6, 2)	(-1, 4, 1)	no	yes	-1	(1, 2, 1)
(1, 2, 1)	(-1, 1, 2)	no	yes	-1	(2, 1, -1)

EPOCH 3

weights	input	desired	actual	error	new weights
(2, 1, -1)	(-1, 4, 5)	yes	no	1	(1, 5, 4)
(1, 5, 4)	(-1, 6, 1)	yes	yes	0	no change
(1, 5, 4)	(-1, 4, 1)	no	yes	-1	(2, 1, 3)
(2, 1, 3)	(-1, 1, 2)	no	yes	-1	(3, 0, 1)

EPOCH 4

weights	input	desired	actual	error	new weights
(3, 0, 1)	(-1, 4, 5)	yes	yes	0	no change
(3, 0, 1)	(-1, 6, 1)	yes	no	1	(2, 6, 2)
(2, 6, 2)	(-1, 4, 1)	no	yes	-1	(3, 2, 1)
(3, 2, 1)	(-1, 1, 2)	no	yes	-1	(4, 1, -1)

EPOCH 5

weights	input	desired	actual	error	new weights
(4, 1, -1)	(-1, 4, 5)	yes	no	1	(3, 5, 4)
(3, 5, 4)	(-1, 6, 1)	yes	yes	0	no change
(3, 5, 4)	(-1, 4, 1)	no	yes	-1	(4, 1, 3)
(4, 1, 3)	(-1, 1, 2)	no	yes	-1	(5, 0, 1)

EPOCH 6

weights	input	desired	actual	error	new weights
(5, 0, 1)	(-1, 4, 5)	yes	no	1	(4, 4, 6)
(4, 4, 6)	(-1, 6, 1)	yes	yes	0	no change
(4, 4, 6)	(-1, 4, 1)	no	yes	-1	(5, 0, 5)
(5, 0, 5)	(-1, 1, 2)	no	yes	-1	(6, -1, 3)

EPOCH 7

weights	input	desired	actual	error	new weights
(6, -1, 3)	(-1, 4, 5)	yes	yes	0	no change
(6, -1, 3)	(-1, 6, 1)	yes	no	1	(5, 5, 4)
(5, 5, 4)	(-1, 4, 1)	no	yes	-1	(6, 1, 3)
(6, 1, 3)	(-1, 1, 2)	no	yes	-1	(7, 0, 1)

EPOCH 8

weights	input	desired	actual	error	new weights
(7, 0, 1)	(-1, 4, 5)	yes	no	1	(6, 4, 6)
(6, 4, 6)	(-1, 6, 1)	yes	yes	0	no change
(6, 4, 6)	(-1, 4, 1)	no	yes	-1	(7, 0, 5)
(7, 0, 5)	(-1, 1, 2)	no	yes	-1	(8, -1, 3)

EPOCH 9

weights	input	desired	actual	error	new weights
(8, -1, 3)	(-1, 4, 5)	yes	yes	0	no change
(8, -1, 3)	(-1, 6, 1)	yes	no	1	(7, 5, 2)
(7, 5, 2)	(-1, 4, 1)	no	yes	-1	(8, 1, 3)
(8, 1, 3)	(-1, 1, 2)	no	no	0	(8, 1, 3)

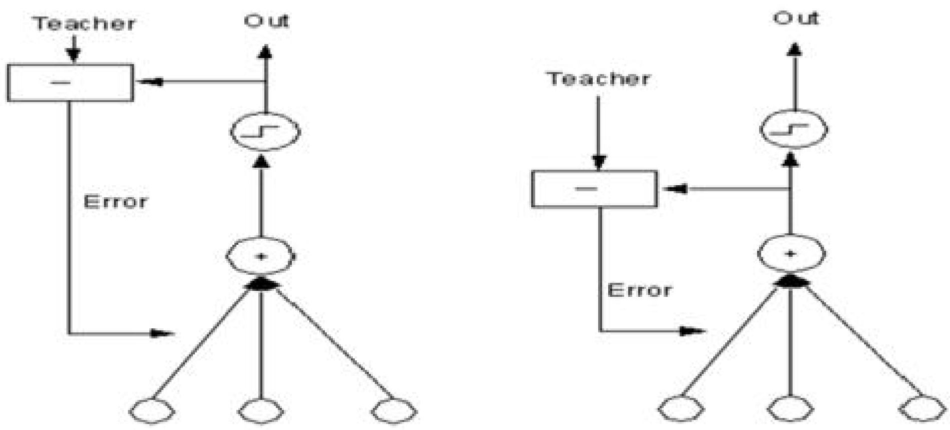
EPOCH 10

weights	input	desired	actual	error	new weights
(8, 1, 3)	(-1, 4, 5)	yes	yes	0	no change
(8, 1, 3)	(-1, 6, 1)	yes	yes	0	no change
(8, 1, 3)	(-1, 4, 1)	no	no	0	no change
(8, 1, 3)	(-1, 1, 2)	no	no	0	no change

Adaline Example

ADAptive LInear NEuron

Example – AND Function

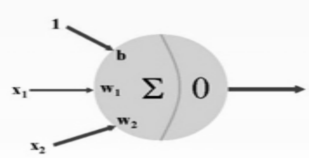


Perceptron Learning

Adaline

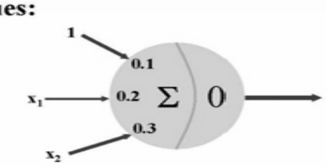
Example – AND Function

- Construct an AND function for a ADALINE neuron
  - let $\eta = 0.1$



x1	x2	bias	Target
1	1	1	1
1	−1	1	−1
−1	1	1	−1
−1	−1	1	−1

Initial Conditions: Set the weights to small random values:



First Training Run

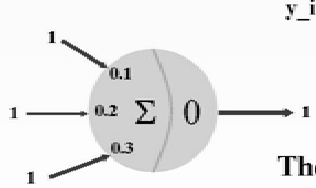
- Apply the input (1,1) with output 1

Neuron input

$y\_in = b + \sum x_i w_i$

Delta rule

$b(new) = b(old) + \eta(t - y\_in)$
$w_i(new) = w_i(old) + \eta(t - y\_in)x_i$



The net input is:

$y\_in = 0.1 + 0.2*1 + 0.3*1 = 0.6$

The new weights are:

$b = 0.1 + 0.1(1-0.6) = 0.14$
$w_1 = 0.2 + 0.1(1-0.6)1 = 0.24$
$w_2 = 0.3 + 0.1(1-0.6)1 = 0.34$

The largest weight change is 0.04

Second Training Run

- Apply the second training set (1 -1) with output -1

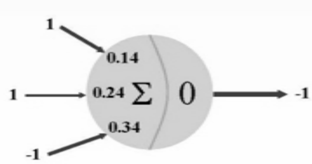
The net input is:

$y\_in = 0.14 + 0.24*1 + 0.34*(-1) = 0.04$

The new weights are:

$b = 0.14 - 0.1(1+0.04) = 0.04$
$w_1 = 0.24 - 0.1(1+0.04)1 = 0.14$
$w_2 = 0.34 + 0.1(1+0.04)1 = 0.44$

The largest weight change is 0.1

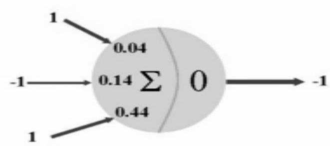


Third Training Run

- Apply the third training set (-1 1) with output -1

The net input is:

$$y\_in = 0.04 - 0.14*1 + 0.44*1 = 0.34$$



The new weights are:

$$b = 0.04 - 0.1(1+0.34) = -0.09$$
$$w_1 = 0.14 + 0.1(1+0.34)1 = 0.27$$
$$w_2 = 0.44 - 0.1(1+0.34)1 = 0.31$$

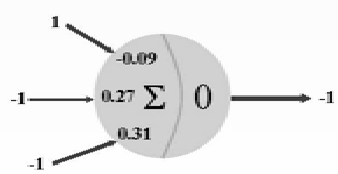
The largest weight change is 0.13

Fourth Training Run

- Apply the fourth training set (-1 -1) with output -1

The net input is:

$$y\_in = -0.09 - 0.27*1 - 0.31*1 = -0.67$$



The new weights are:

$$b = -0.09 - 0.1(1+0.67) = -0.27$$
$$w_1 = 0.27 + 0.1(1+0.67)1 = 0.43$$
$$w_2 = 0.31 + 0.1(1+0.67)1 = 0.47$$

The largest weight change is 0.16

The final solution

- Continue to cycle through the four training inputs until the largest change in the weights over a complete cycle is less than some small number (say 0.01)
- In this case, the solution becomes
  - $b = -0.5$
  - $w_1 = 0.5$
  - $w_2 = 0.5$