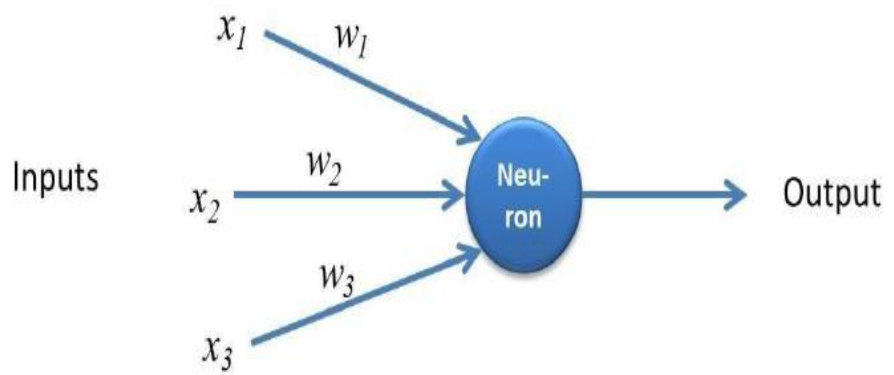
Simple model of Neural Network – The Perceptron

The perceptron learning algorithm is the simplest model of a neuron that illustrates how a neural network works. The perceptron is a machine learning algorithm developed in 1957 by Frank Rosenblatt and first implemented in IBM 704.



Perceptron formulae

$$y\_in = b + \sum_i x_i w_i;$$

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

$$w_i(\text{new}) = w_i(\text{old}) + \alpha t x_i,$$

$$b(\text{new}) = b(\text{old}) + \alpha t.$$

- $\Delta w = t(x_1, x_2, 1)$

Perceptron

- A Perceptron for the AND function: Binary inputs, bipolar targets

Solution

- Initializing $\alpha = 1$; $w_1, w_2, b = 0$ and considering

$\theta = 0.2$

- The weight change is $\Delta w = t(x_1, x_2, 1)$ if an error has occurred and zero otherwise

- Presenting the first input we have,

INPUT $(x_1$	x_2	$1)$	NET	OUT	TARGET	WEIGHT CHANGES			WEIGHTS $(w_1$	w_2	$b)$
									(0	0	0)
(1	1	1)	0	0	1	(1	1	1)	(1	1	1)

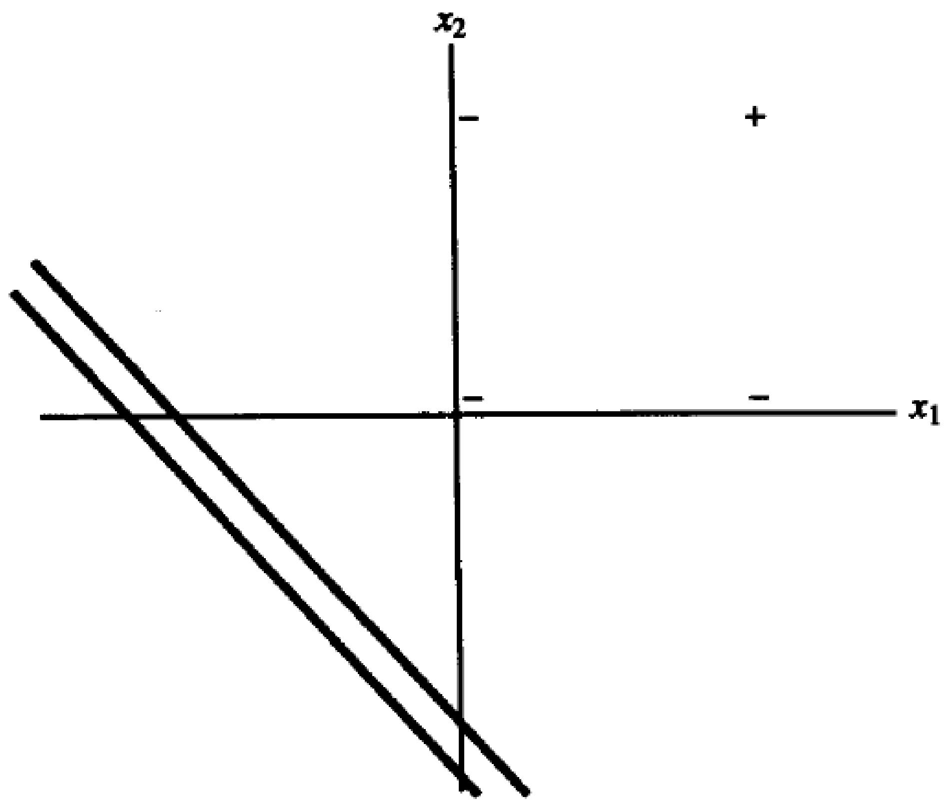
The separating lines become

$$x_1 + x_2 + 1 = .2$$

and

$$x_1 + x_2 + 1 = -.2.$$

- Decision boundary for logic function AND after first training input



- Presenting the second input yields the following,

INPUT			NET	OUT	TARGET	WEIGHT CHANGES			WEIGHTS		
(x_1	x_2	1)							(w_1	w_2	b)
									(1	1	1)
(1	0	1)	2	1	−1	(−1	0	−1)	(0	1	0)

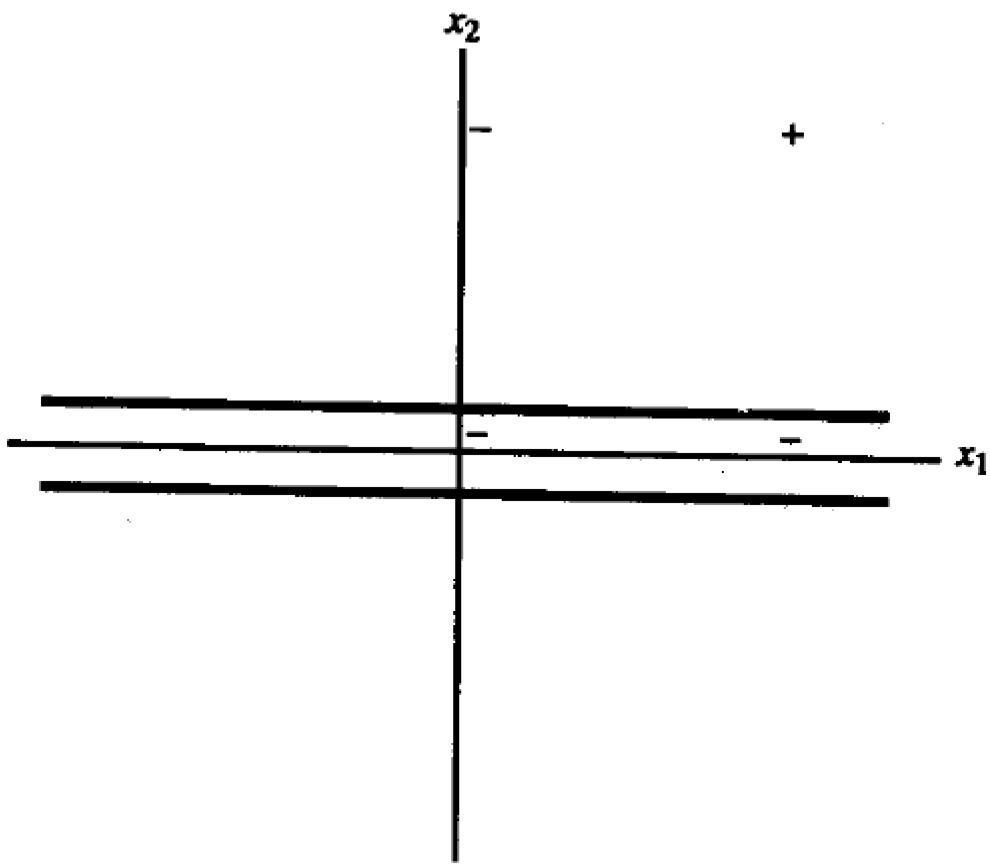
The separating lines become

$$x_2 = .2$$

and

$$x_2 = -.2$$

- Decision boundary after second training input,



- For the third input we have,

INPUT			NET	OUT	TARGET	WEIGHT CHANGES			WEIGHTS		
(x_1	x_2	1)							(w_1	w_2	b)
									(0	1	0)
(0	1	1)	1	1	−1	(0	−1	−1)	(0	0	−1)

- For fourth input pattern of first epoch,

INPUT			NET	OUT	TARGET	WEIGHT CHANGES			WEIGHTS		
$(x_1$	x_2	$1)$							$(w_1$	w_2	$b)$
									(0	0	−1)
(0	0	1)	−1	−1	−1	(0	0	−(0)	(0	0	−1)

- Response of all of the input pattern is negative for the weight derived but since the response for input pattern (1,1) is incorrect, we are not finished

- The second epoch of training yields the following weight updates for the first input:

INPUT			NET	OUT	TARGET	WEIGHT CHANGES			WEIGHTS		
(x_1	x_2	1)							(w_1	w_2	b)
									(0	0	−1)
(1	1	1)	−1	−1	1	(1	1	1)	(1	1	0)

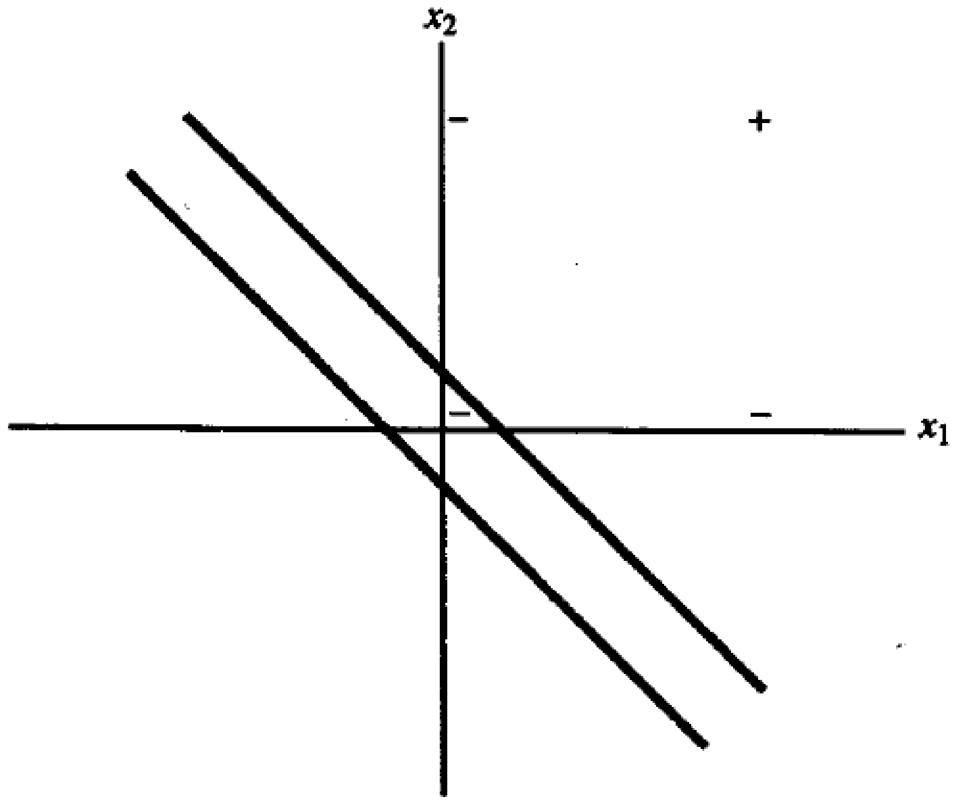
The separating lines become

$$x_1 + x_2 = .2$$

and

$$x_1 + x_2 = -.2.$$

- Boundary after first training input of second epoch,



- For the second input in second epoch,

INPUT			NET	OUT	TARGET	WEIGHT CHANGES			WEIGHTS		
(x_1	x_2	1)							(w_1	w_2	b)
									(1	1	0)
(1	0	1)	1	1	−1	(−1	0	−1)	(0	1	−1)

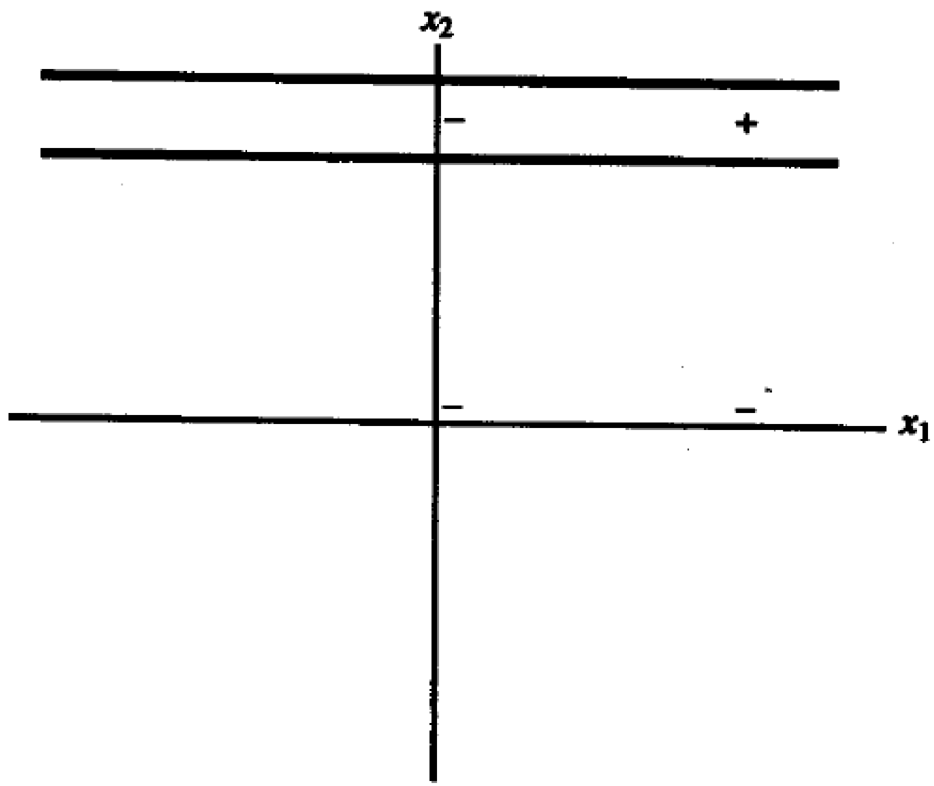
The separating lines become

$$x_2 - 1 = .2$$

and

$$x_2 - 1 = -.2.$$

- Boundary after second input of second epoch,



- In the second epoch, the third input yields:

INPUT			NET	OUT	TARGET	WEIGHT CHANGES			WEIGHTS		
$(x_1$	x_2	$1)$							$(w_1$	w_2	$b)$
									(0	1	−1)
(0	1	1)	0	0	−1	(0	−1	−1)	(0	0	−2)

- To complete the second epoch of training, we present the fourth training pattern:

INPUT			NET	OUT	TARGET	WEIGHTS CHANGE			WEIGHTS		
(x_1	x_2	1)							(w_1	w_2	b)
									(0	0	−2)
(0	0	1)	−2	−1	−1	(0	0	0)	(0	0	−2)

- The result of third epoch will be,

INPUT			NET	OUT	TARGET	WEIGHT CHANGES			WEIGHTS		
(x_1	x_2	1)							(w_1	w_2	b)
									(0	0	−2)
(1	1	1)	−2	−1	1	(1	1	1)	(1	1	−1)
(1	0	1)	0	0	−1	(−1	0	−1)	(0	1	−2)
(0	1	1)	−1	−1	−1	(0	0	0)	(0	1	−2)
(0	0	1)	−2	−1	−1	(0	0	0)	(0	1	−2)

The results for the fourth epoch are:

(1	1	1)	−1	−1	1	(1	1	1)	(1	2	−1)
(1	0	1)	0	0	−1	(−1	0	−1)	(0	2	−2)
(0	1	1)	0	0	−1	(0	−1	−1)	(0	1	−3)
(0	0	1)	−3	−1	−1	(0	0	0)	(0	1	−3)

Finally, the results for the tenth epoch are:

			net	y	t						
(1	1	1)	1	1	1	(0	0	0)	(2	3	−4)
(1	0	1)	−2	−1	−1	(0	0	0)	(2	3	−4)
(0	1	1)	−1	−1	−1	(0	0	0)	(2	3	−4)
(0	0	1)	−4	−1	−1	(0	0	0)	(2	3	−4)

Thus, the positive response is given by all points such that

$$2x_1 + 3x_2 - 4 > .2,$$

with boundary line

$$x_2 = -\frac{2}{3}x_1 + \frac{7}{5},$$

and the negative response is given by all points such that

$$2x_1 + 3x_2 - 4 < -.2,$$

with boundary line

$$x_2 = -\frac{2}{3}x_1 + \frac{19}{15}$$

- Final decision boundaries for AND function in perceptron learning