UNIT 4

PERCEPTRON

Algorithm:

Step 0. Initialize weights and bias.

(For simplicity, set weights and bias to zero.) Set learning rate.

Step 1. While stopping condition is false, do Steps 2-6.

Step 2. For each training pair s:t, do Steps 3-5.

Step 3. Set activations of input units:

$$x_i = s_i$$

Step 4. Compute response of output unit

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

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

Step 5. Update weights and bias if an error occurred for this pattern.

$$y \ne t$$

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

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

else

$$w_i(\text{new}) = w_i(\text{old})$$

$$b(\text{new}) = b(\text{old})$$

Step 6. Test stopping condition.

If no weights changed in Step 2, stop; else, continue.

Example

A Perceptron for the **AND** function: bipolar inputs and targets

Let us consider again the AND function with binary input and bipolar target, now using the perceptron learning rule. The training data are as given in Example 2.6 for the Hebb rule. An adjustable bias is included, since it is necessary if a single-layer net is to be able to solve this problem. For simplicity, we take $\alpha = 1$ and set the initial weights and bias to 0, as indicated. However, to illustrate the role of the threshold, we take $\theta = .2$.

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

The separating lines become

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

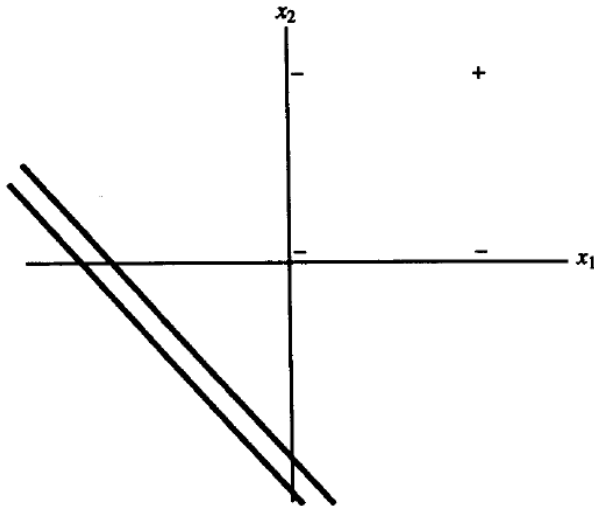and

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

Figure 2.15 Decision boundary for logic function AND after first training input.

The graph in Figure 2.15 shows that the response of the net will now be correct for the first input pattern.

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

$$x_2 = -.2$$

The graph in Figure 2.16 shows that the response of the net will now (still) be correct for the first input point.

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	$-1)$

Since the components of the input patterns are nonnegative and the components of the weight vector are nonpositive, the response of the net will be negative (or zero).
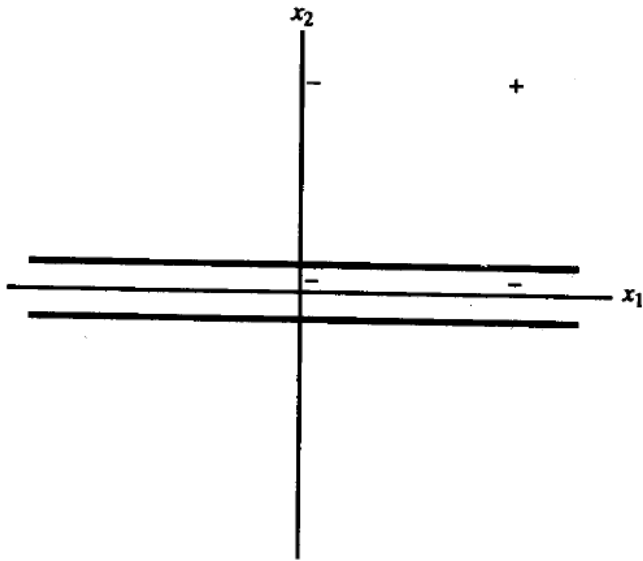
Figure 2.16 Decision boundary after second training input.

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

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

The response for all of the input patterns is negative for the weights derived; but since the response for input pattern (1, 1) is not correct, we are not finished.

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

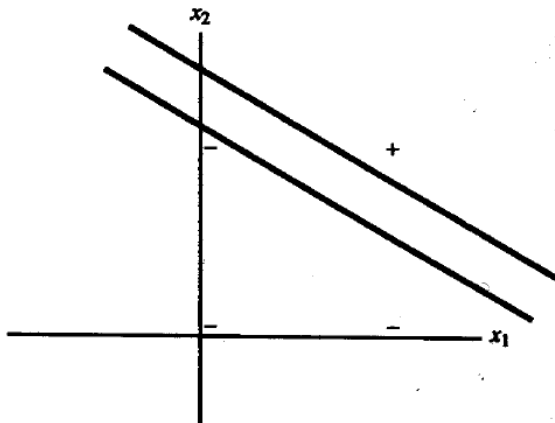$$x_2 = -\frac{2}{3}x_1 + \frac{19}{15}$$

(see Figure 2.19.)

Figure 2.19 Final decision boundaries for AND function in perceptron learning.