EENG582
Artificial Neural Networks

Problems & Solutions

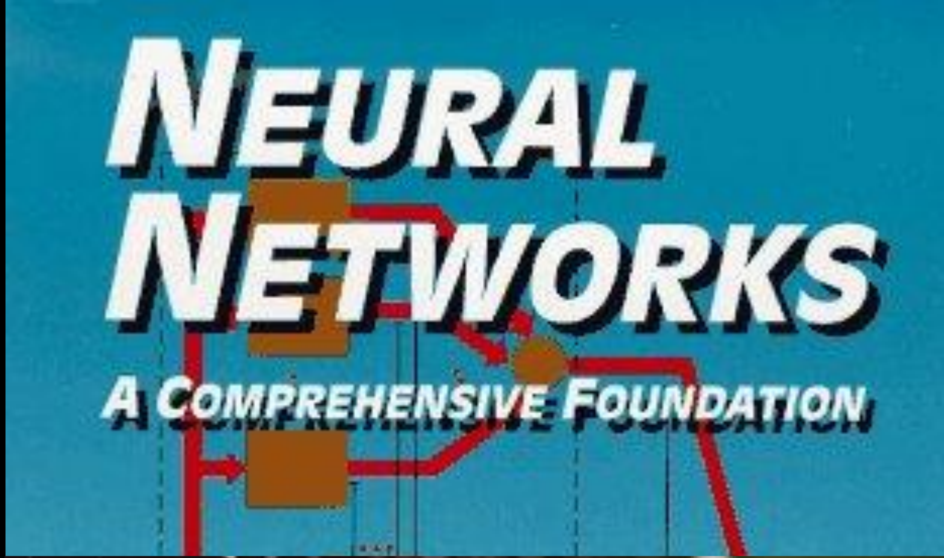
Problems 2.a

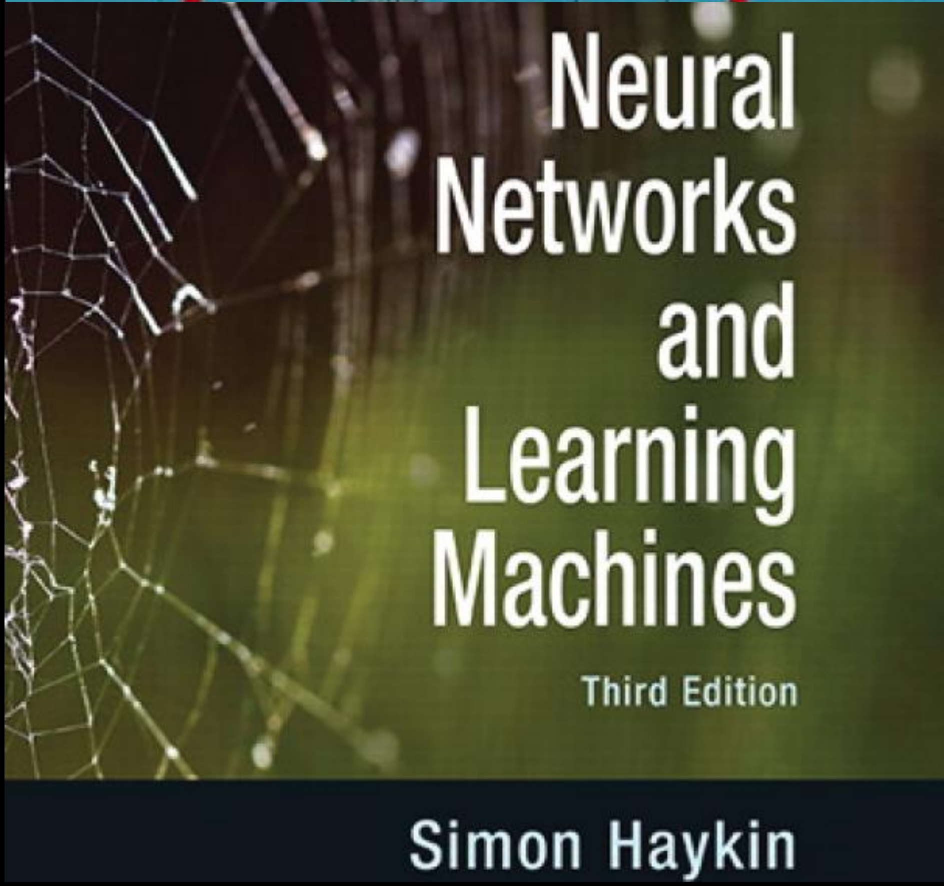
Artificial Neural Networks

Prof. Dr. Hasan AMCA

Electrical and Electronic Engineering Department
(ee.emu.edu.tr)

Eastern Mediterranean University
(emu.edu.tr)





Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

Equations (1.19)–(1.22) summarizing the perceptron convergence algorithm.

$$\mathrm{sgn}(v) = \begin{cases} +1 & \text{if } v > 0 \\ -1 & \text{if } v < 0 \end{cases} \tag{1.19}$$

$$y(n) = \mathrm{sgn}[\mathbf{w}^T(n)\mathbf{x}(n)] \tag{1.20}$$

$$d(n) = \begin{cases} +1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_1 \\ -1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_2 \end{cases} \tag{1.21}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n) \tag{1.22}$$

The algorithm for adapting the weight vector of the elementary perceptron are (1.5) and (1.6).

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) > 0 \text{ and } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_1 \\ \mathbf{w}(n+1) &= \mathbf{w}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) \le 0 \text{ and } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_2 \end{aligned} \tag{1.5}$$

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \eta(n)\mathbf{x}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) > 0 \text{ and } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_2 \\ \mathbf{w}(n+1) &= \mathbf{w}(n) + \eta(n)\mathbf{x}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) \le 0 \text{ and } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_1 \end{aligned} \tag{1.6}$$

Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

Soln. 1.1. For the perceptron to function properly, the two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ must be linearly separable. This, in turn, means that the patterns to be classified must be sufficiently separated from each other to ensure that the decision surface consists of a hyperplane. This requirement is illustrated in Fig. 1.4 for the case of a two-dimensional perceptron.

In Fig. 1.4a, the two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ are sufficiently separated from each other for us to draw a hyperplane (in this case, a straight line) as the decision boundary. If, however, the two classes $\mathcal{C}_1$ and $\mathcal{C}_2$ are allowed to move too close to each other, as in Fig. 1.4b, they become nonlinearly separable, a situation that is beyond the computing capability of the perceptron.

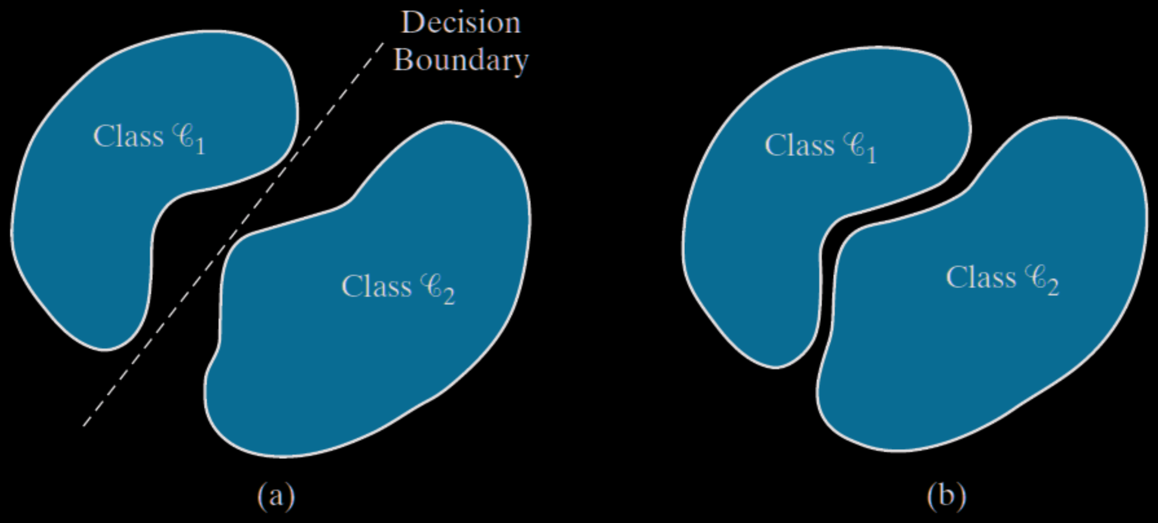


FIGURE 1.4 (a) A pair of linearly separable patterns. (b) A pair of non-linearly separable patterns.

Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

Soln. 1.1.

Suppose then that the input variables of the perceptron originate from two linearly separable classes. Let $\mathcal{H}_1$ be the subspace of training vectors $\mathbf{x}_1(1)$, $\mathbf{x}_1(2)$, ... that belong to class $\mathcal{C}_1$, and let $\mathcal{H}_2$ be the subspace of training vectors $\mathbf{x}_2(1)$, $\mathbf{x}_2(2)$, ... that belong to class $\mathcal{C}_2$. The union of $\mathcal{H}_1$ and $\mathcal{H}_2$ is the complete space denoted by $\mathcal{H}$.

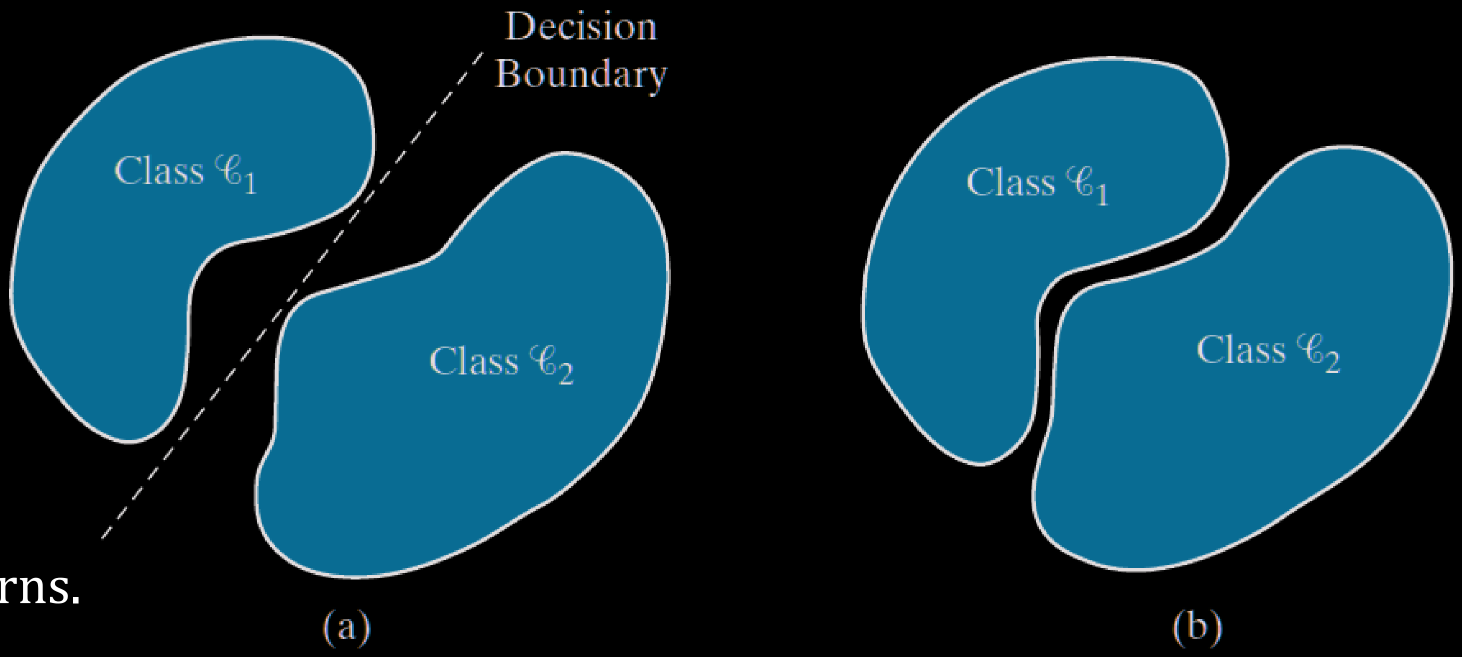


FIGURE 1.4
(a) A pair of linearly separable patterns.
(b) A pair of non-linearly separable patterns.

Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

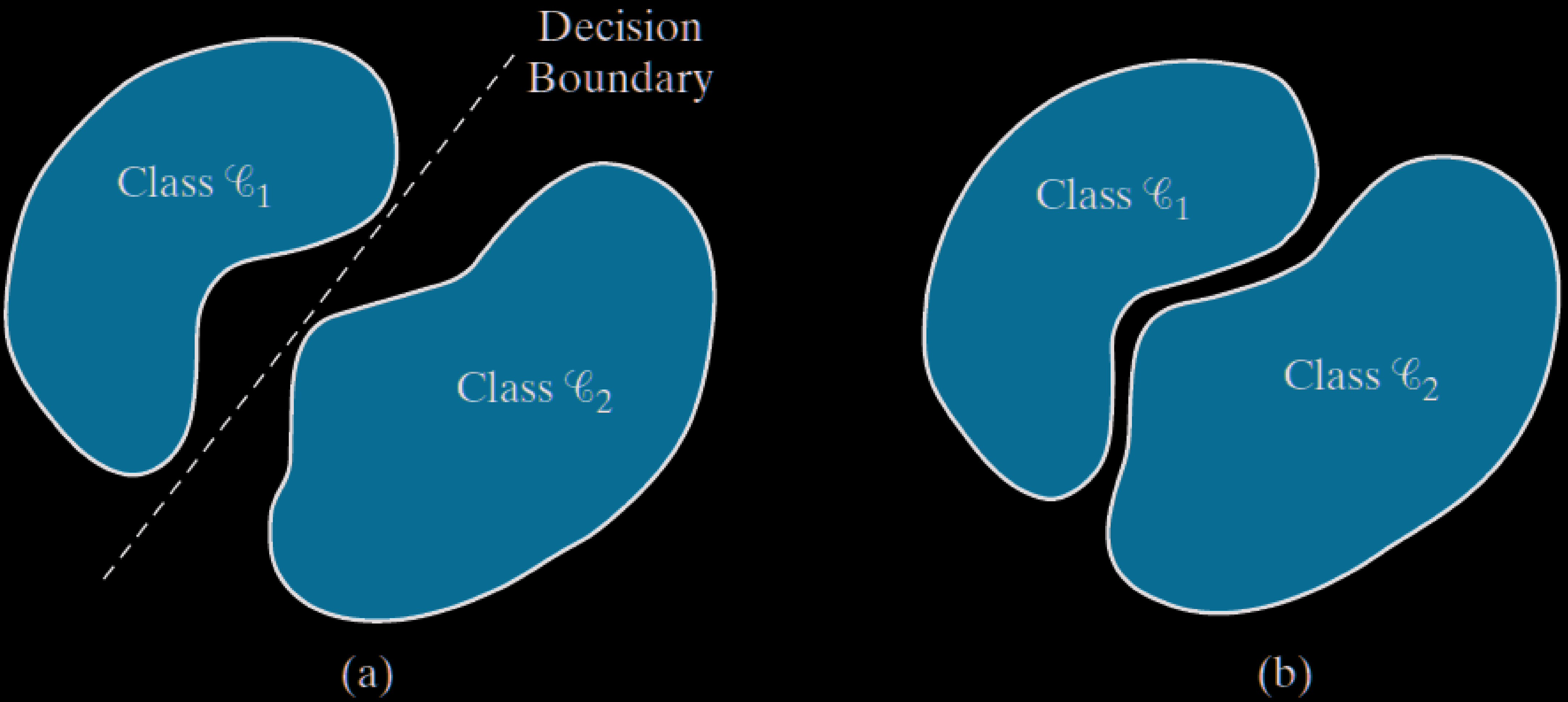


FIGURE 1.4 (a) A pair of linearly separable patterns. (b) A pair of non-linearly separable patterns.

Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

Soln. 1.1.

Recall the equations (1.19)-(1.22) and recalling Figure 1.4: for linearly separable and nonlinearly separable pattern plots for reference.

$$\operatorname{sgn}(v) = \begin{cases} +1 & \text{if } v > 0 \\ -1 & \text{if } v < 0 \end{cases} \tag{1.19}$$

$$y(n) = \operatorname{sgn}[\mathbf{w}^T(n)\mathbf{x}(n)] \tag{1.20}$$

$$d(n) = \begin{cases} +1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_1 \\ -1 & \text{if } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_2 \end{cases} \tag{1.21}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta[d(n) - y(n)]\mathbf{x}(n) \tag{1.22}$$

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) > 0 \text{ and } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_1 \\ \mathbf{w}(n+1) &= \mathbf{w}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) \le 0 \text{ and } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_2 \end{aligned} \tag{1.5}$$

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \eta(n)\mathbf{x}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) > 0 \text{ and } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_2 \\ \mathbf{w}(n+1) &= \mathbf{w}(n) + \eta(n)\mathbf{x}(n) \quad \text{if } \mathbf{w}^T(n)\mathbf{x}(n) \le 0 \text{ and } \mathbf{x}(n) \text{ belongs to class } \mathscr{C}_1 \end{aligned} \tag{1.6}$$

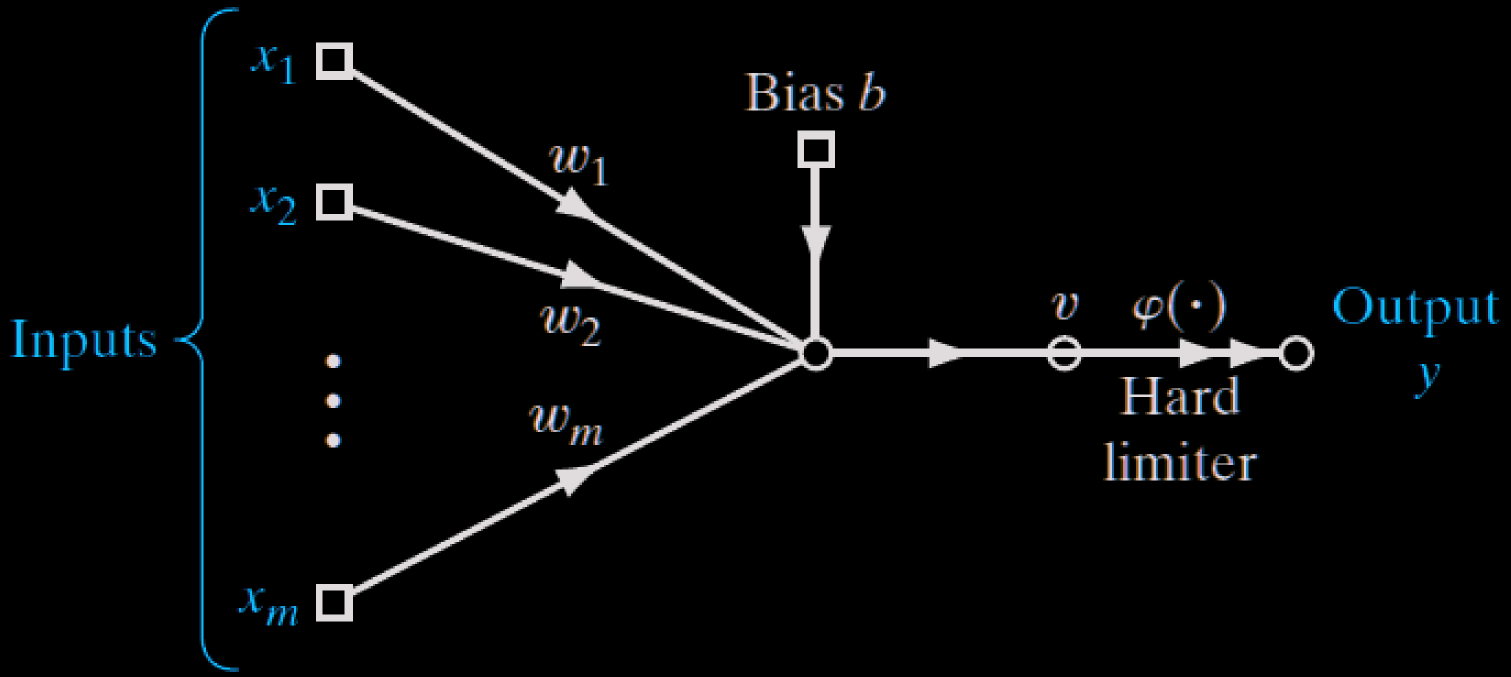
Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

- The input to the nonlinear device $v = b + w^T(n)x(n)$

$$\text{sgn}(v) = \begin{cases} +1 & \text{if } v > 0 \\ -1 & \text{if } v < 0 \end{cases} \tag{1.19}$$

$$y(n) = \text{sgn}[\mathbf{w}^T(n)\mathbf{x}(n)] \tag{1.20}$$

FIGURE 1.1 Signal-flow graph of the perceptron.



Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

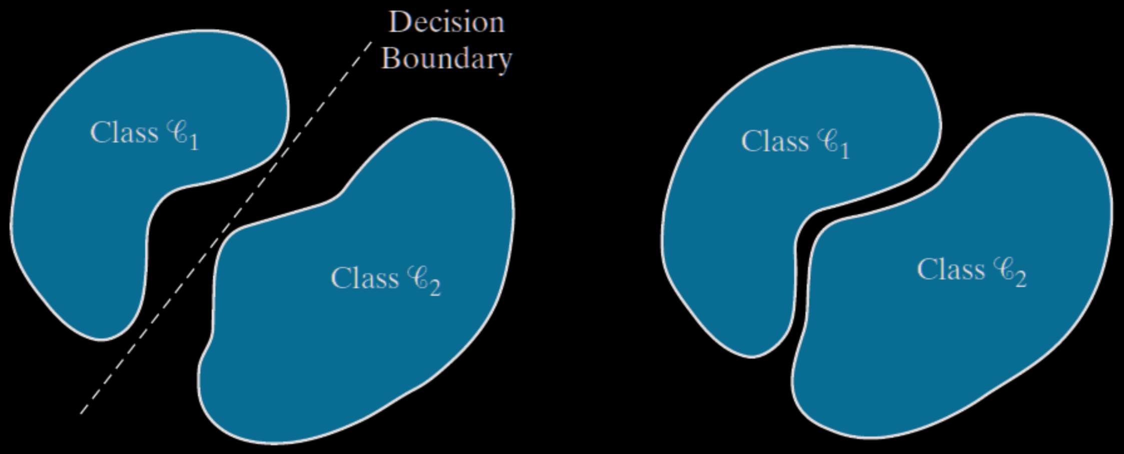
1. From (1.20) Recalling the behavior of the signum function we observe that if $\mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n) > 0$, then $y(n) = +1$.

If also $\mathbf{x}(n)$ belongs to $\mathcal{C}_1$, then $d(n) = +1$.

Under these conditions, the error signal is $e(n) = d(n) - y(n) = 0$.

And from (1.22), we get $\mathbf{w}(n+1) = \mathbf{w}(n) + \eta e(n)\mathbf{x}(n) = \mathbf{w}(n)$,

This final result is same as line 1 of Eq.(1.5).



(a) (b)

Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

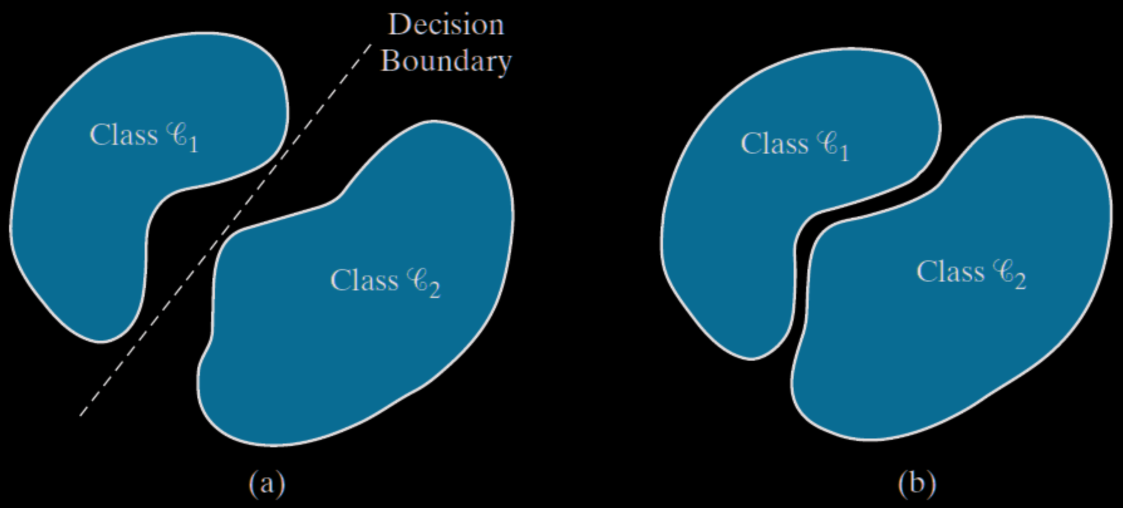
2. If $\mathbf{w}^{T}(n)\mathbf{x}(n) < 0$, then $y(n) = -1$.
If also $\mathbf{x}(n)$ belongs to $\mathcal{C}_2$, then
$d(n) = -1$.
Under these conditions, the error signal remains to be

$$e(n) = d(n) - y(n) = 0$$

and from (1.22) we have

$$\mathbf{w}(n+1) = \mathbf{w}(n)$$

This result is same as line 2 of Eq.(1.5).



Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

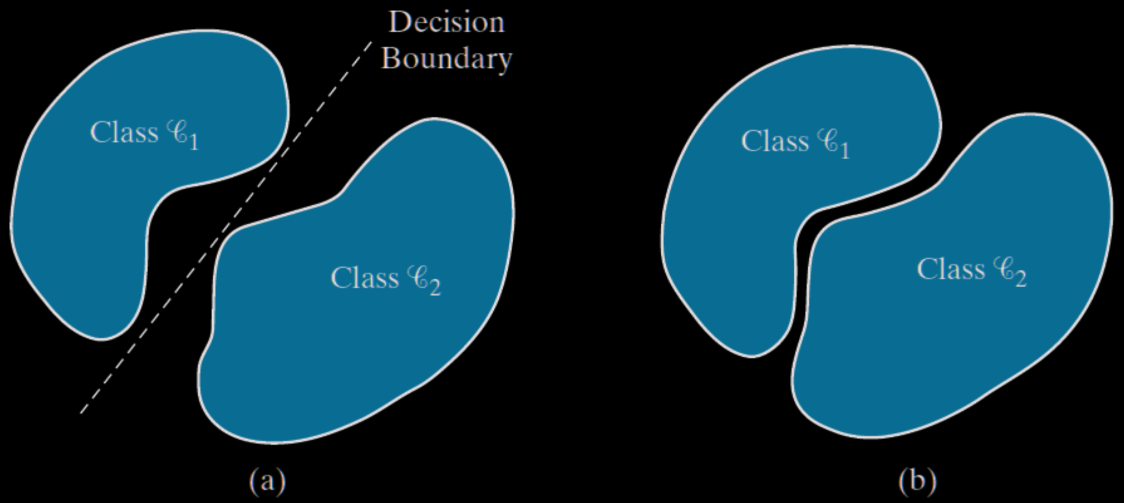
3. If $\mathbf{w}^{\mathrm{T}}(n)\mathbf{x}(n) > 0$ $\mathbf{x}(n)$ belongs to $\mathcal{C}_2$, we have

$$y(n) = +1 \text{ and } d(n) = -1$$

the error signal $e(n) = d(n) - y(n) = -2$. So, (1.22) yields

$$\mathbf{w}(n+1) = \mathbf{w}(n) - 2\eta\mathbf{x}(n)$$

Which has the same form as the first line of Eq.(1.6), except for the scaling factor 2.



Pr. 1.1: Verify that Eqs. (1.19)–(1.22), summarizing the perceptron convergence algorithm, are consistent with Eqs. (1.5) and (1.6).

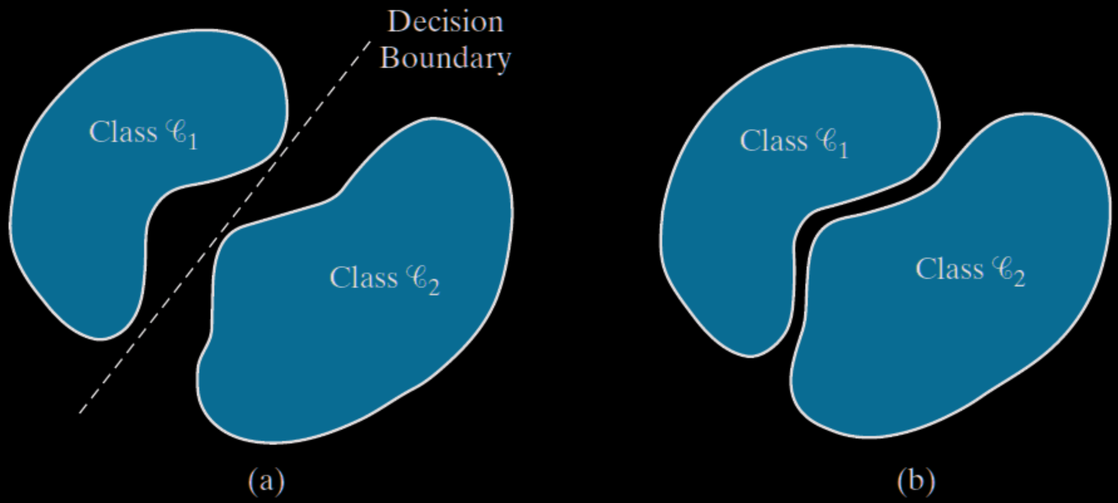
4. Finally, if $\mathbf{w}^T(n)\mathbf{x}(n) > 0$ and $\mathbf{x}(n)$ belongs to $\mathcal{C}_1$, then

$$y(n) = -1 \text{ and } d(n) = +1$$

In this case, the use of Eq.(1.22) yields, the error signal $e(n) = d(n) - y(n) = -2$. So, (1.22) yields

$$\mathbf{w}(n+1) = \mathbf{w}(n) + 2\eta\mathbf{x}(n)$$

Which has the same form as the second line of Eq.(1.6), except for the scaling factor 2.



Pr. 1.2: Suppose that in the signal-flow graph of the perceptron shown in Fig. 1.1, the hard limiter $\varphi(v) = \tanh\left(\frac{v}{2}\right)$ where v is the induced local field. The classification decisions made by the perceptron are defined as follows: Observation vector $\mathbf{x}$ belongs to class $\mathcal{C}_1$ if the output $y > \xi$, where ξ (pronounced as xi) is a threshold; otherwise, $\mathbf{x}$ belongs to class $\mathcal{C}_2$. Show that the decision boundary so constructed is a hyperplane.

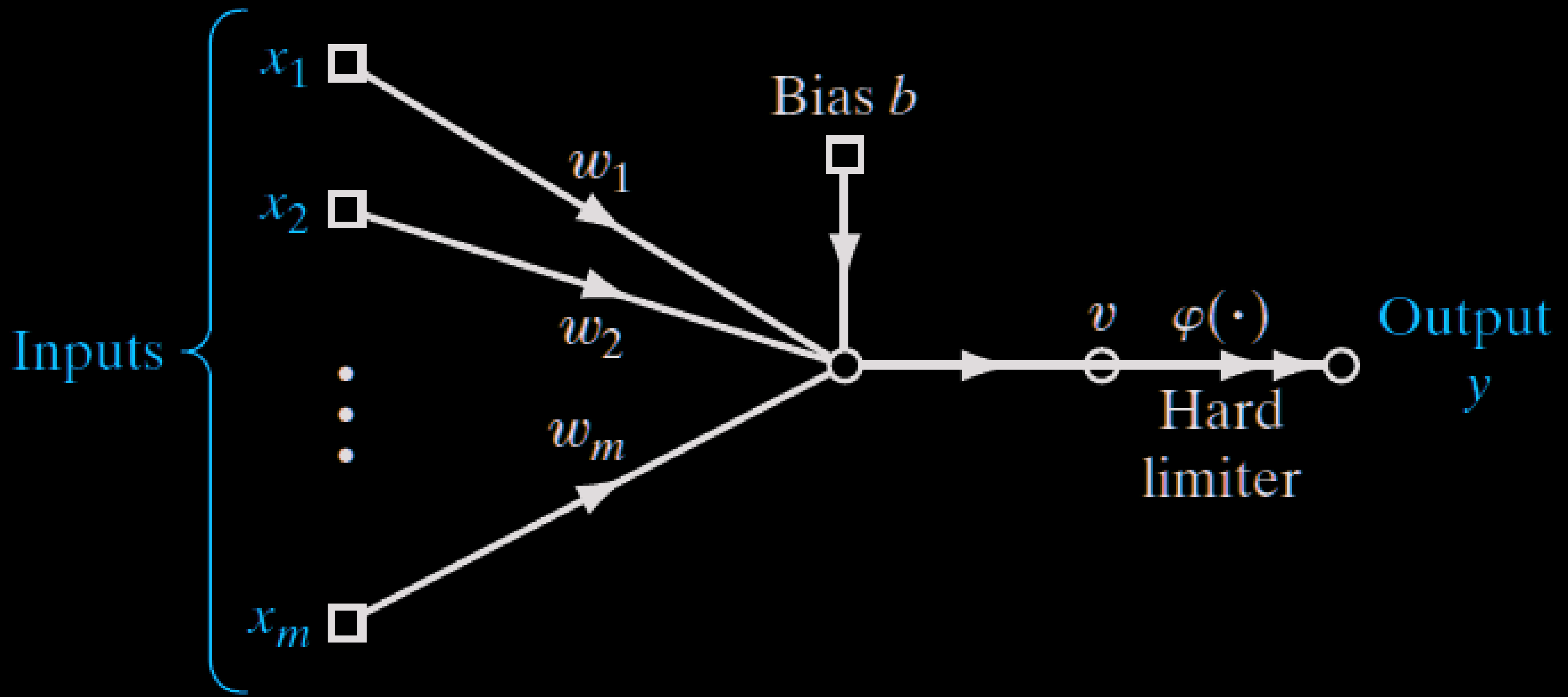
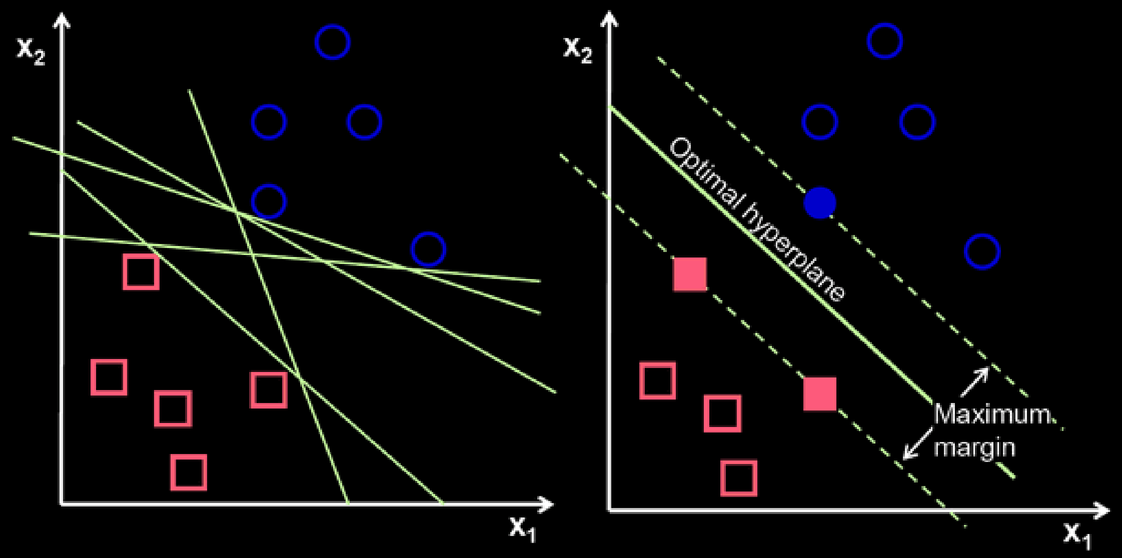


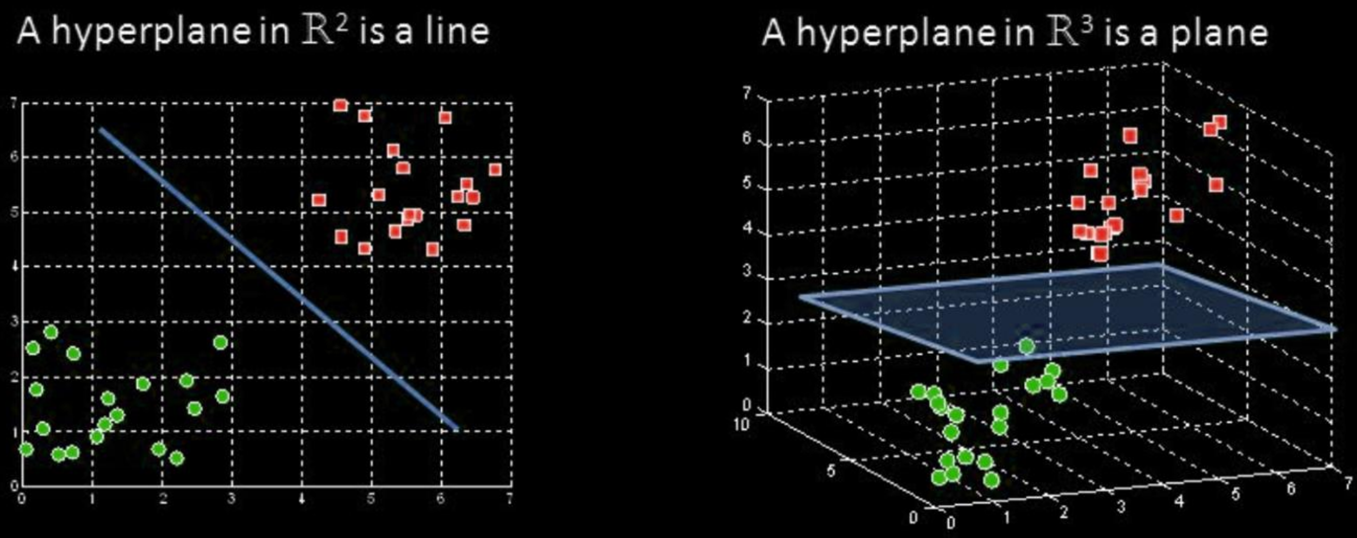
FIGURE 1.1 Signal-flow graph of the perceptron.

Pr. 1.2: Background for Hyperplanes and Python Codes for examples.

- Hyperplanes are decision boundaries that help classify the data points.
- Data points falling on either side of the hyperplane can be attributed to different classes.
- Also, the dimension of the hyperplane depends upon the number of features.
- Using these support vectors, we maximize the margin of the classifier.



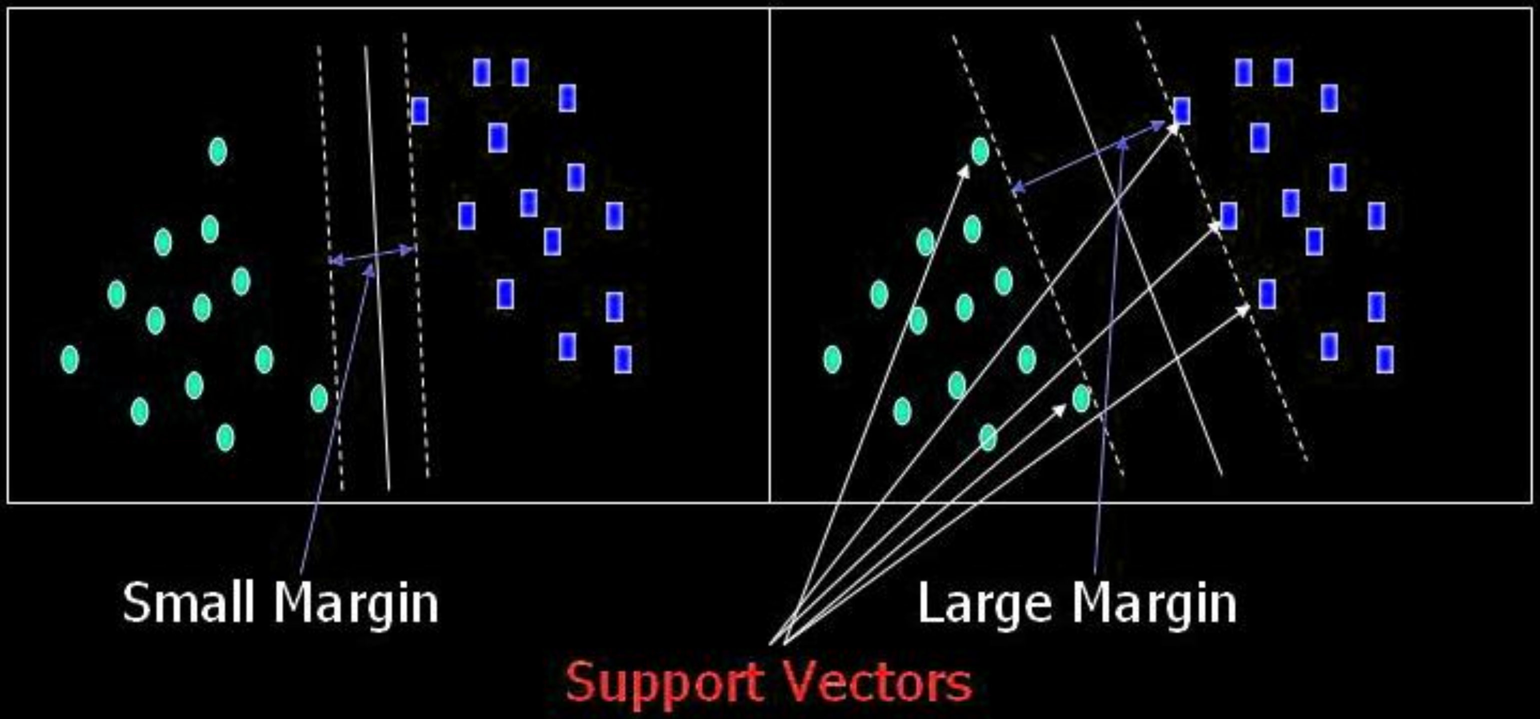
Possible hyperplanes



Hyperplanes in 2D and 3D feature space

Pr. 1.2: Background for Support Vector Definition and examples.

- Support vectors are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane.
- Using these support vectors, we maximize the margin of the classifier.
- Deleting the support vectors will change the position of the hyperplane.
- These are the points that help us build our Support Vector Machine (SVM)



https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

Soln. 1.2:

The output signal is given in the question as

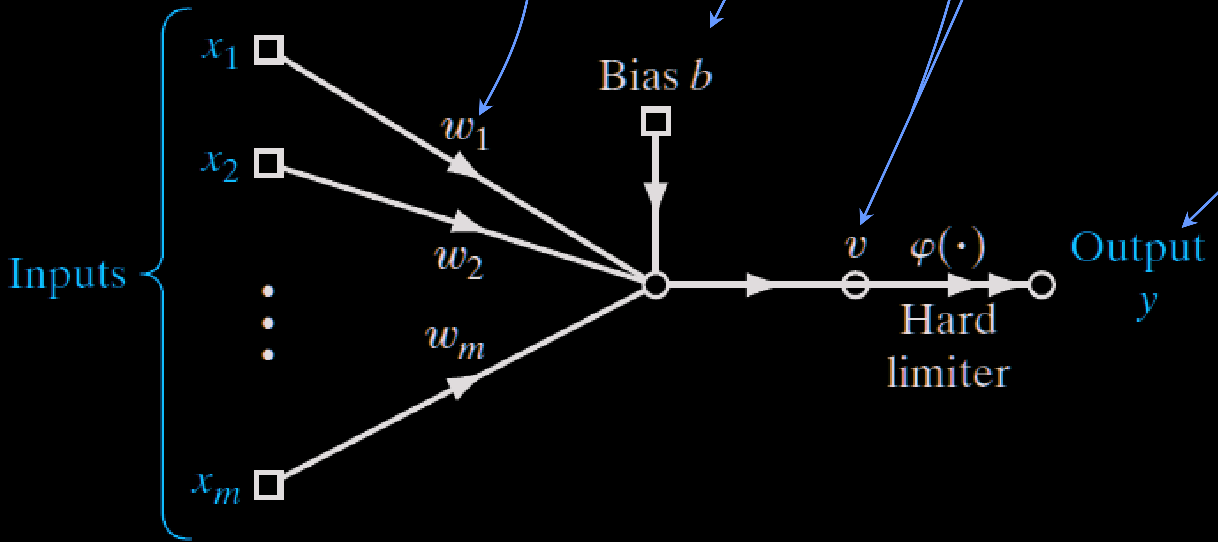
$$y = \tanh\left(\frac{v}{2}\right) = \tanh\left(\frac{b}{2} + \frac{1}{2}\sum_i w_i x_i\right)$$

Equivalently,

$$v = b + \frac{1}{2}\sum_i w_i x_i \qquad (1)$$

where, $y' = 2\tanh^{-1} y$. Equation (1) is the equation of a hyperplane.

FIGURE 1.1 Signal-flow graph of the perceptron.



Pr. 1.3: (a) The perceptron may be used to perform numerous logic functions. Demonstrate the implementation of the binary logic functions AND, OR, and COMPLEMENT.

(b) A basic limitation of the perceptron is that it cannot implement the EXCLUSIVE OR function. Explain the reason for this limitation....

Pr. 1.3: (a) AND operation: Truth Table 1

Inputs		Output
x_1	x_2	y
1	1	1
0	1	0
1	0	0
0	0	0

This operation may be realized using the perceptron of Fig. 1

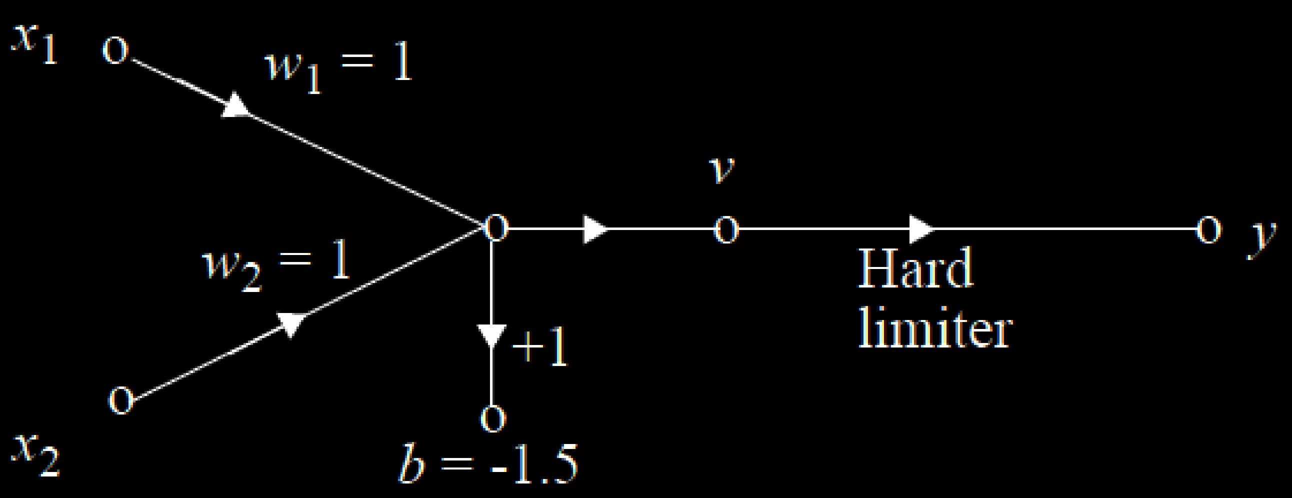


Figure 1: Problem 1.3

Pr. 1.3:

The hard limiter input is

$$
\begin{aligned}
v &= w_1 x_1 + w_2 x_2 + b \\
&= x_1 + x_2 - 1.5
\end{aligned}
$$

If $x_1 = x_2 = 1$, then $v = 0.5$, and $y = 1$

If $x_1 = 0$, and $x_2 = 1$, then $v = -0.5$, and $y = 0$

If $x_1 = 1$, and $x_2 = 0$, then $v = -0.5$, and $y = 0$

If $x_1 = x_2 = 0$, then $v = -1.5$, and $y = 0$

Pr. 1.3:

These conditions agree with truth table 1.

OR operation: Truth Table 2

Inputs		Output
x_1	x_2	y
1	1	1
0	1	1
1	0	1
0	0	0

Pr. 1.3: The OR operation may be realized using the perceptron of Fig. 2:

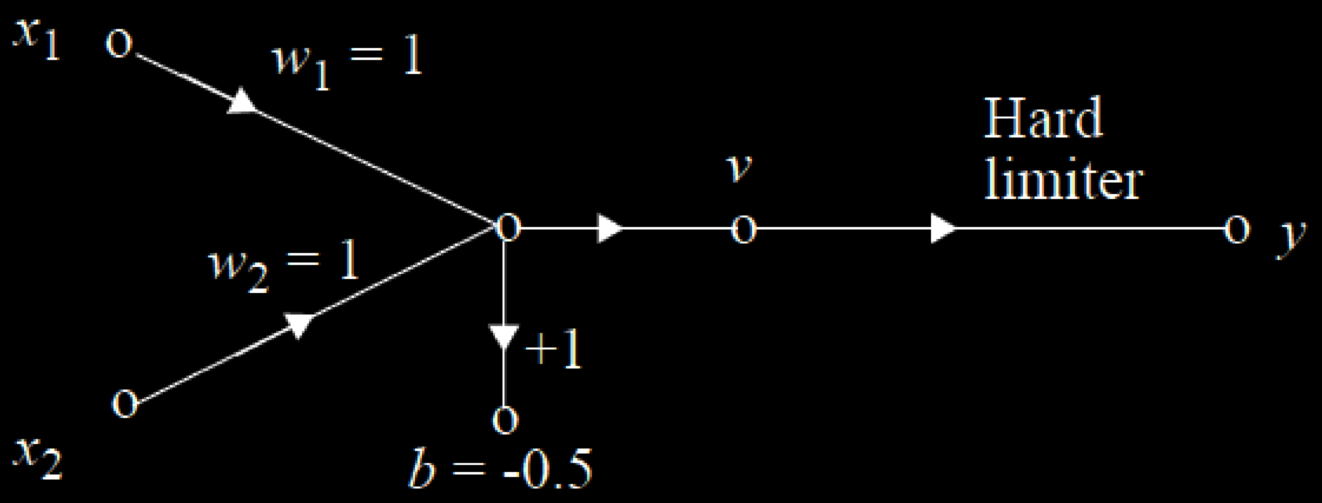


Figure 2: Problem 1.3

In this case, the hard limiter input is

$$v = x_1 + x_2 - 0.5$$

If $x_1 = x_2 = 1$, then $v = 1.5$, and $y = 1$

If $x_1 = 0$, and $x_2 = 1$, then $v = 0.5$, and $y = 1$

If $x_1 = 1$, and $x_2 = 0$, then $v = 0.5$, and $y = 1$

If $x_1 = x_2 = 0$, then $v = -0.5$, and $y = -1$

Pr. 1.3:

These conditions agree with truth table 2.

COMPLEMENT operation: Truth Table 3

Input x,	Output, y
1	0
0	1

The COMPLEMENT operation may be realized as in Figure 3::

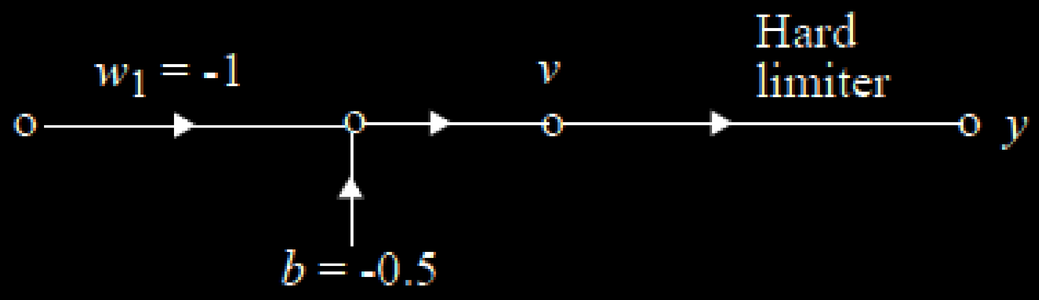


Figure 3: Problem 1.3

The hard limiter input is

$$v = wx + b = -x + 0.5$$

If $x = 1$, then $v = -0.5$, and $y = 0$
If $x = 0$, then $v = 0.5$, and $y = 1$

These conditions agree with truth table 3.

Pr. 1.3:

(b) EXCLUSIVE OR operation: Truth table 4

Inputs		Output
x_1	x_2	y
1	1	0
0	1	1
1	0	1
0	0	0

This operation is nonlinearly separable, which cannot be solved by the perceptron.