IFA-305 Sistem Cerdas (Intelligent System)
Lecture 4

Rosenblatt's Perceptron – Part 1: Forward Computation

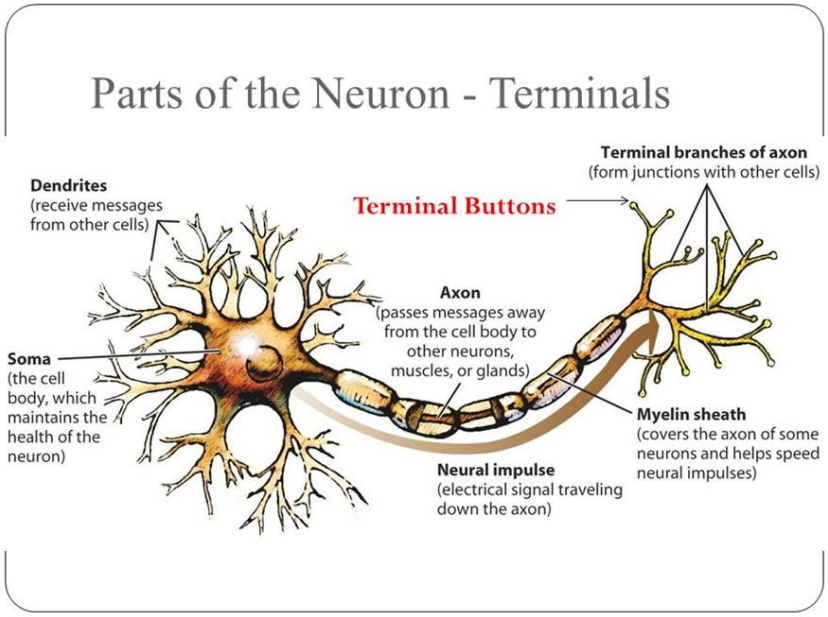
Nur Uddin, PhD.

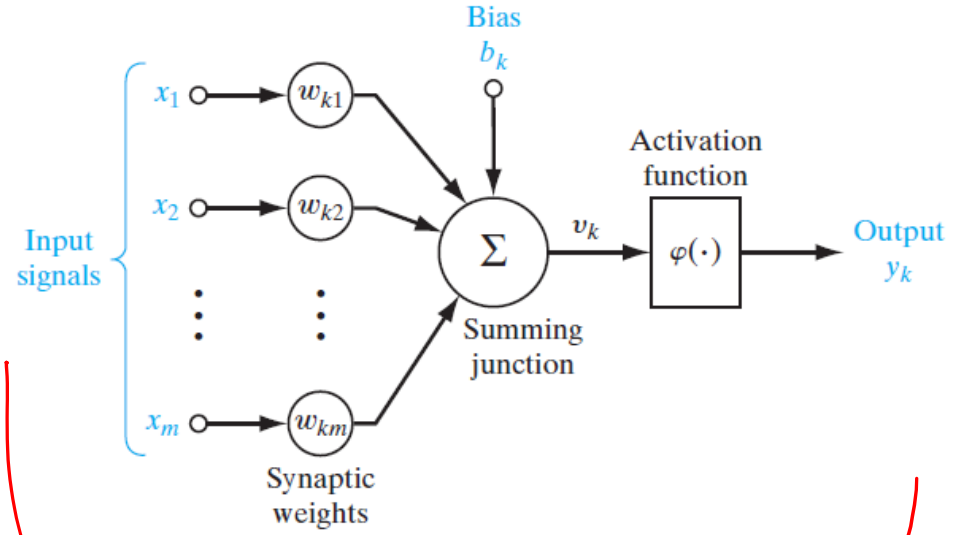
Program Studi Informatika

Universitas Pembangunan Jaya

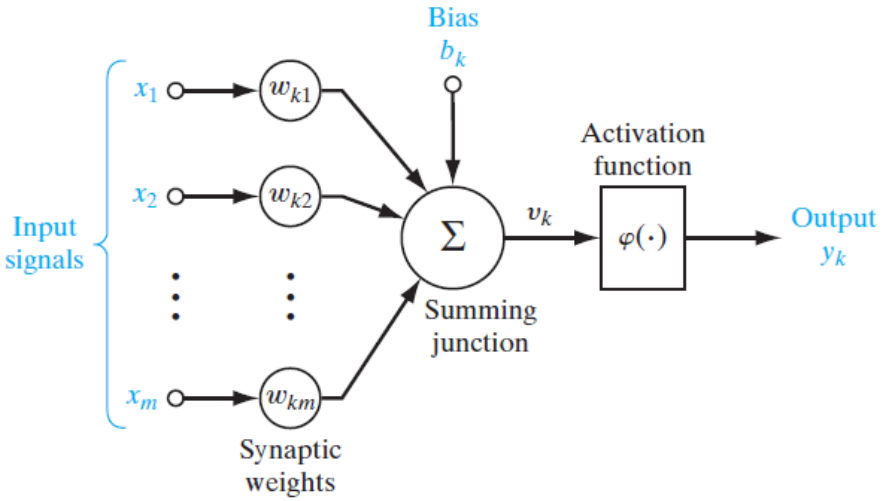
Tangerang Selatan

Model of Neuron



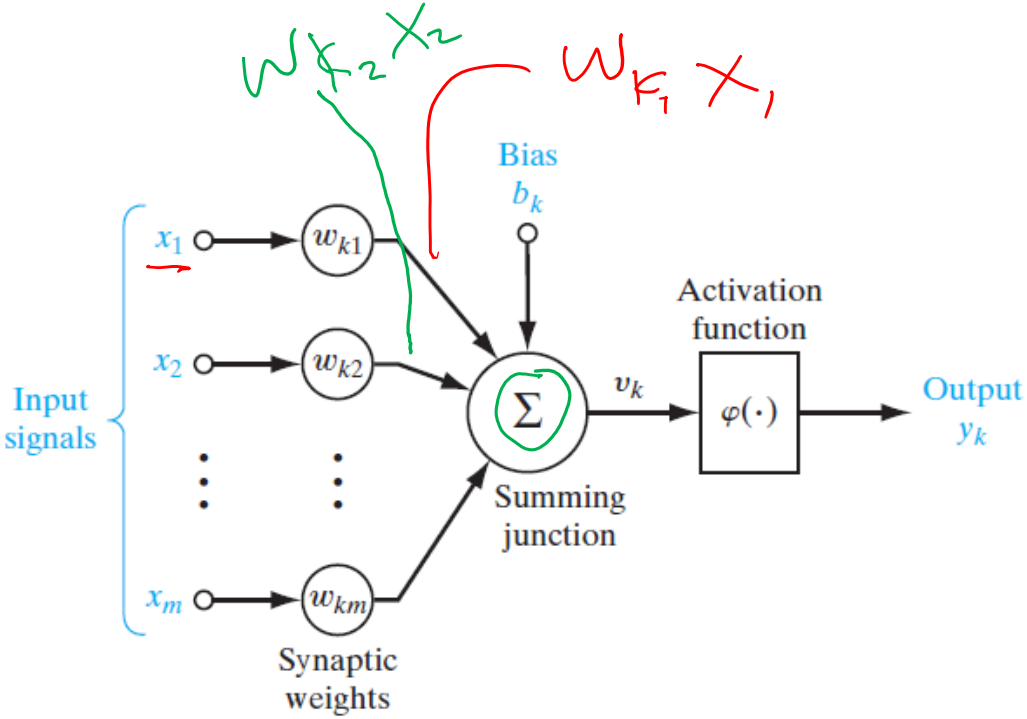


Elements of Neuron Model



- A set of *synapses*, each synapse is characterized by a *weight* to strength the input signals.
- An *adder* for summing the weighted input signals.
- An *activation function* for limiting the amplitude of the output of a neuron
- A *bias* for increasing or lowering the net input of the activation function.

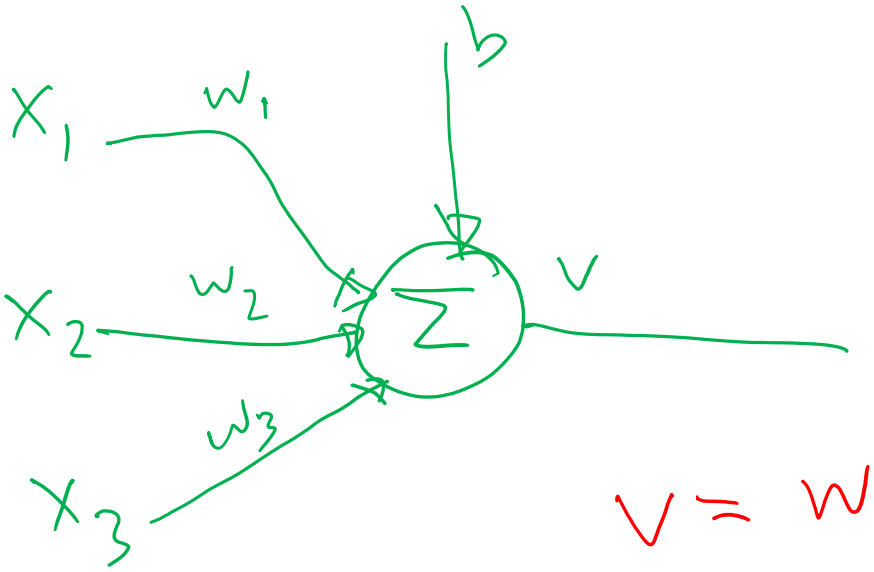
Mathematical Model of Neuron



$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k$$

$$y_k = \varphi(v_k)$$

Exercise:



$w_1 = 2$

$w_2 = 3$

$w_3 = -1$

$$v = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

$$= 2x_1 + 3x_2 - x_3 + b$$

$$v = \sum_{i=1}^{3} w_i x_i + b$$

Matrix Representation

$$v = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

$$= 2x_1 + 3x_2 - x_3 + b \quad \checkmark$$

$$v = \sum_{i=1}^{3} w_i x_i + b$$

$$v = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

$$= \begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b$$

$$\boxed{v = wx + b}$$

$w_1 = 2$
$w_2 = 3$
$w_3 = -1$

$$w = \begin{bmatrix} 2 & 3 & -1 \end{bmatrix}$$

$x_1 = 2$
$x_2 = -1$
$x_3 = 3$

$b = 1$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix}$$

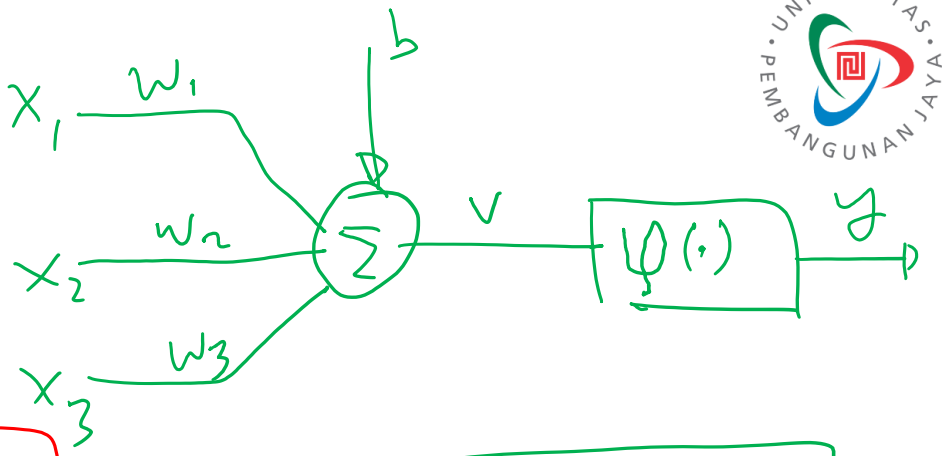
$$v = wx + b$$

$$= \begin{bmatrix} 2 & 3 & -1 \end{bmatrix} \begin{bmatrix} 2 \\ -1 \\ 3 \end{bmatrix} + 1$$

$$= 4 + (-3) + (-3) + 1$$

$$= -1$$

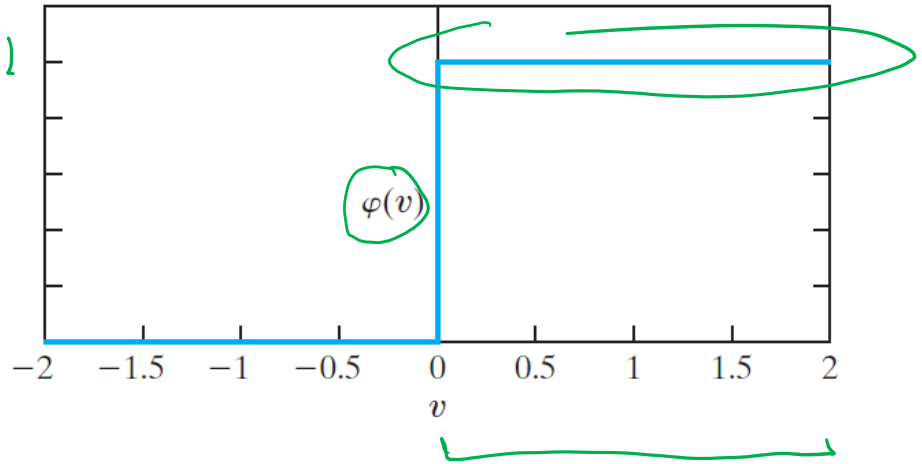
Activation Function



1. Threshold function

$$y = \varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

$$y_k = \begin{cases} 1 & \text{if } v_k \geq 0 \\ 0 & \text{if } v_k < 0 \end{cases}$$



$v = -1$

$y = \varphi(-1) = 0$

Activation Function (Cont'd)

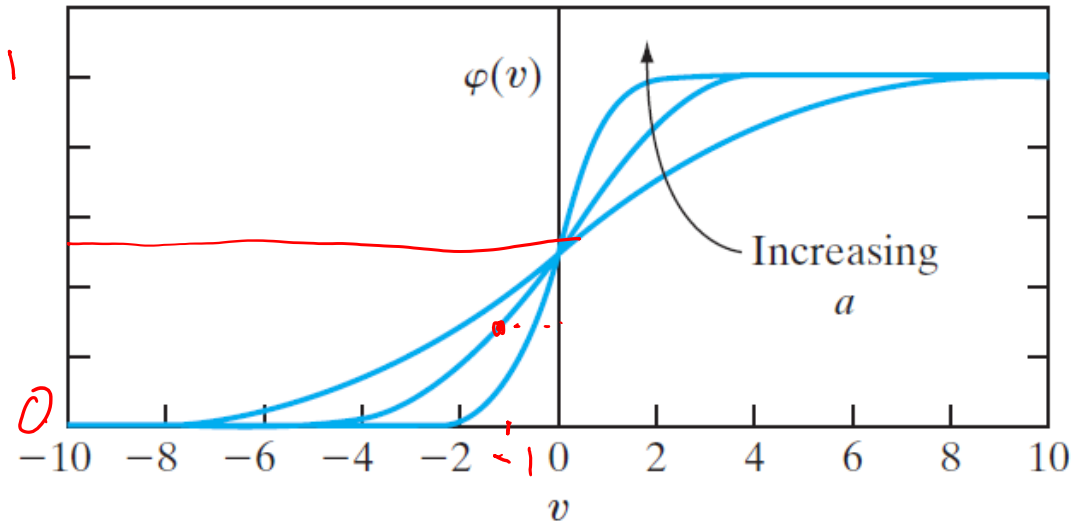
2. Sigmoid function

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

$a = 5$

$v = -1$

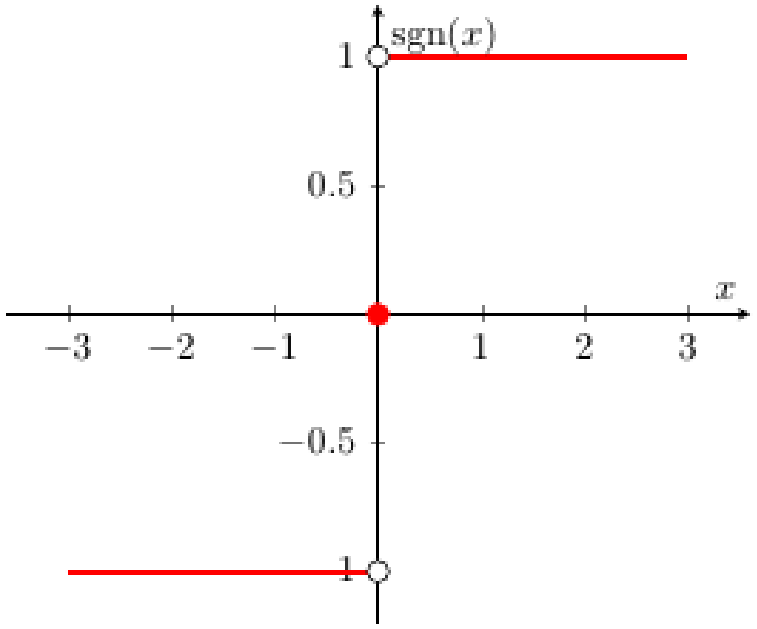
$$\varphi(-1) = \frac{1}{1 + \exp(-5)}$$



Activation Function (Cont'd)

3. Signum function

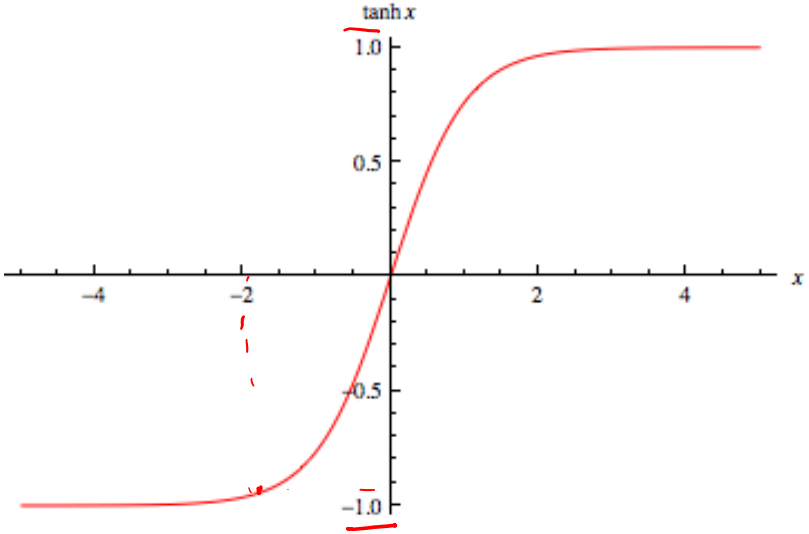
$$\varphi(v) = \begin{cases} 1 & \text{if } v > 0 \\ 0 & \text{if } v = 0 \\ -1 & \text{if } v < 0 \end{cases}$$



Activation Function (Cont'd)

4. Hyperbolic tangent function

$$\varphi(v) = \tanh(v)$$



$$tanh\ x = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

Rosenblatt’s Perceptron

History

In the formative years of neural networks (1943–1958), several researchers stand out for their pioneering contributions:

- McCulloch and Pitts (1943) for introducing the idea of neural networks as computing machines.
- Hebb (1949) for postulating the first rule for self-organized learning.
- Rosenblatt (1958) for proposing the perceptron as the first model for learning with a teacher (i.e., supervised learning).

History

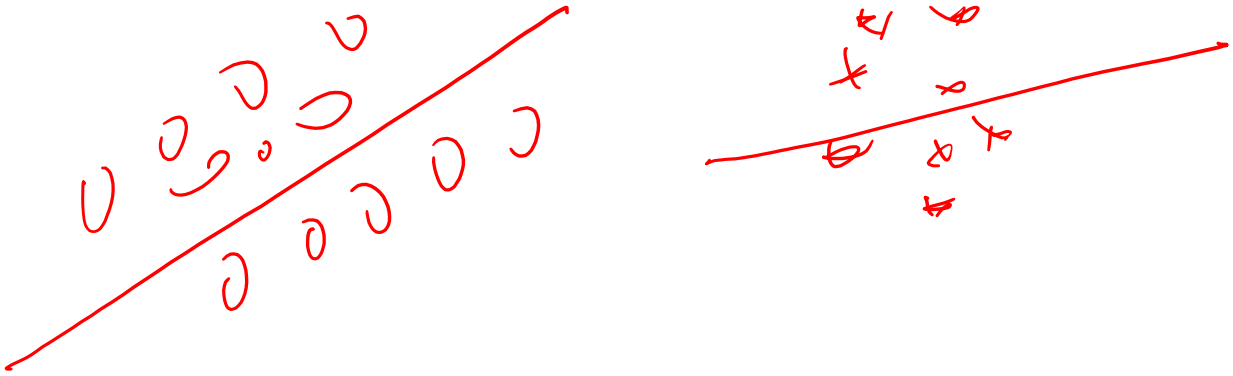
Pioneering work on neural network:

- McCulloch and Pitts (1943) for introducing the idea of neural networks as computing machines.
- Hebb (1949) for postulating the first rule for self-organized learning.
- Rosenblatt (1958) for proposing the perceptron as the first model for learning with a teacher (i.e., supervised learning).

Perceptron

= model dari neuron

- The perceptron is the simplest form of a neural network.
- It is used to classify **linearly separable** patterns.
- The learning algorithm was developed by Rosenblatt (1958, 1962) for his perceptron brain model.



Linearly Separable Patterns

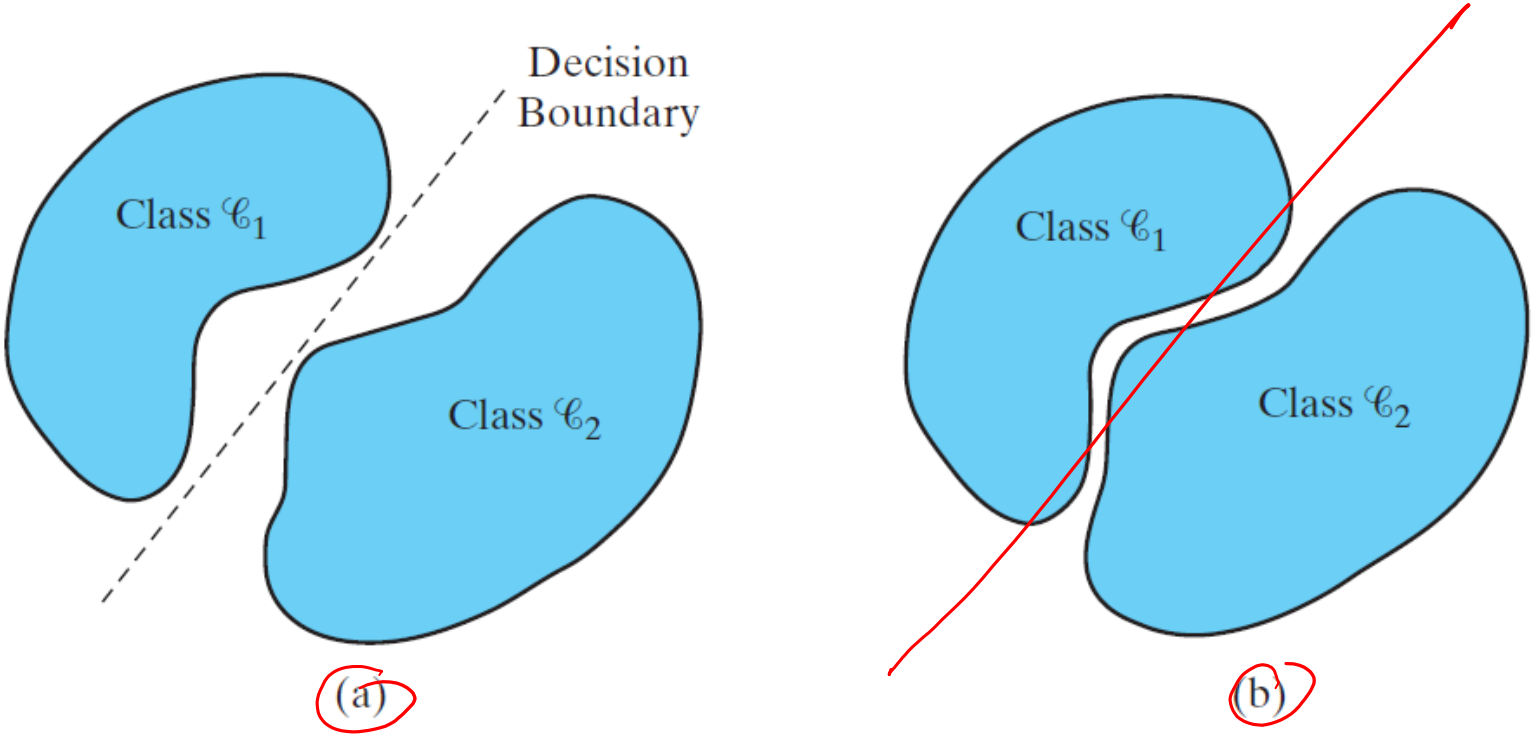
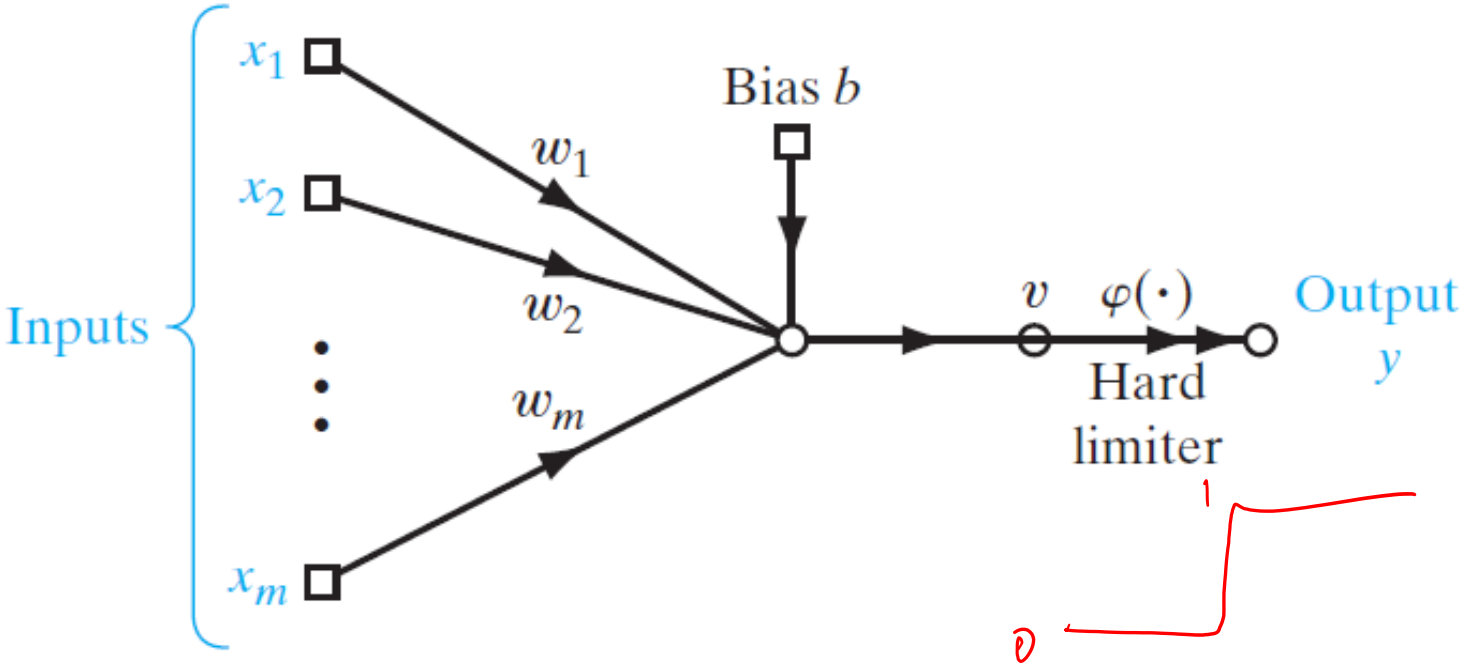


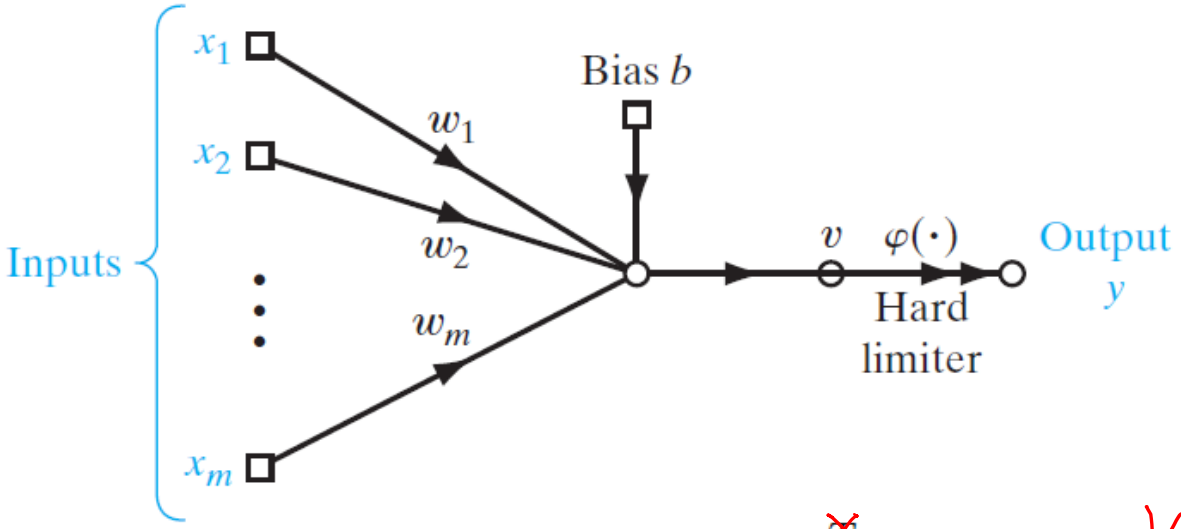
FIGURE 1.4 (a) A pair of linearly separable patterns. (b) A pair of non-linearly separable patterns.

Perceptron Model

- Rosenblatt's perceptron is built around a nonlinear neuron, namely, the McCulloch–Pitts model of a neuron.



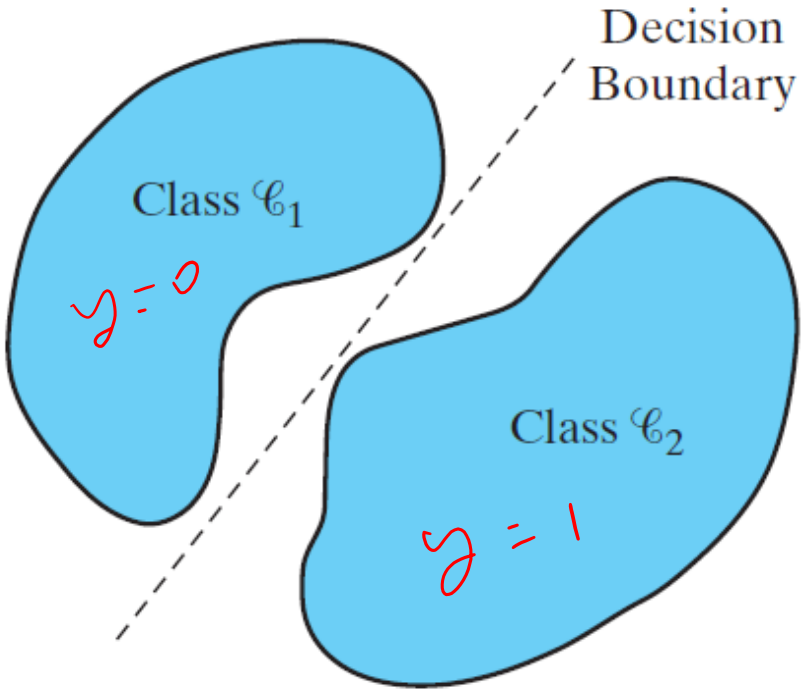
Mathematics Model of Perceptron



$$v = \mathbf{w}^T x + b = Wx + b$$

$$y = \varphi(v)$$

Classification



Question 1:
How to make classification?

$$y = \begin{cases} 0, & \text{belongs to class } C_1 \\ 1, & \text{belongs to class } C_2 \end{cases}$$

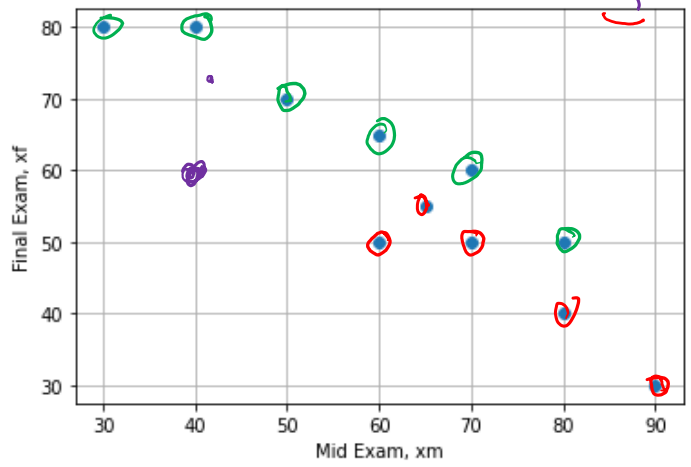
Question 2:
What is the activation function?

Example 1: Grading System

Mid Exam	Final Exam	Grade
60	50	Fail
70	60	Pass
40	80	Pass
60	65	Pass
80	50	Pass
70	50	Fail
65	55	Fail
30	80	Pass
80	40	Fail
90	30	Fail
50	70	Pass

```
import numpy as np
import matplotlib.pyplot as plt
```

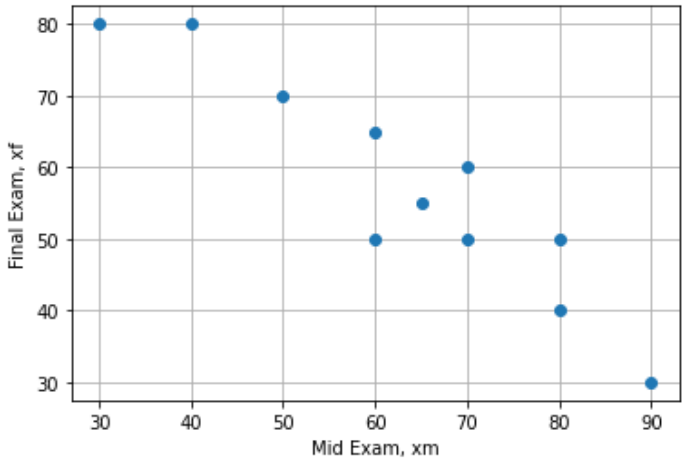
```
# array
xm=np.array([60, 70, 40, 60, 80, 70, 65, 30, 80, 90, 50])  # mid exam
xf=np.array([50, 60, 80, 65, 50, 50, 55, 80, 40, 30, 70])  # final exam
plt.xlabel('Mid Exam, xm')
plt.ylabel('Final Exam, xf')
plt.plot(xm,xf,'o')
plt.grid()
```



Mid Exam	Final Exam	Grade
60	50	Fail
70	60	Pass
40	80	Pass
60	65	Pass
80	50	Pass
70	50	Fail
65	55	Fail
30	80	Pass
80	40	Fail
90	30	Fail
50	70	Pass

```
import numpy as np
import matplotlib.pyplot as plt
```

```
# array
xm=np.array([60, 70, 40, 60, 80, 70, 65, 30, 80, 90, 50])  # mid exam
xf=np.array([50, 60, 80, 65, 50, 50, 55, 80, 40, 30, 70])  # final exam
plt.xlabel('Mid Exam, xm')
plt.ylabel('Final Exam, xf')
plt.plot(xm,xf,'o')
plt.grid()
```



Exercise (Homework): Restaurant Survey

Price	Taste	Buy ?
5	6	Yes
5	7	Yes
6	3	No
6	8	Yes
7	3	No
7	5	No
8	3	No
8	5	No
9	6	No
9	9	Yes
10	7	No