

**523414 :**  
**Artificial Neural Networks (ANN)**  
**trimester 1/2561**

---

**Lecture 02:**  
**Perceptron Learning**

Institute of Engineering  
School of Computer Engineering  
Suranaree University of Technology

# Single-Neuron Perceptron

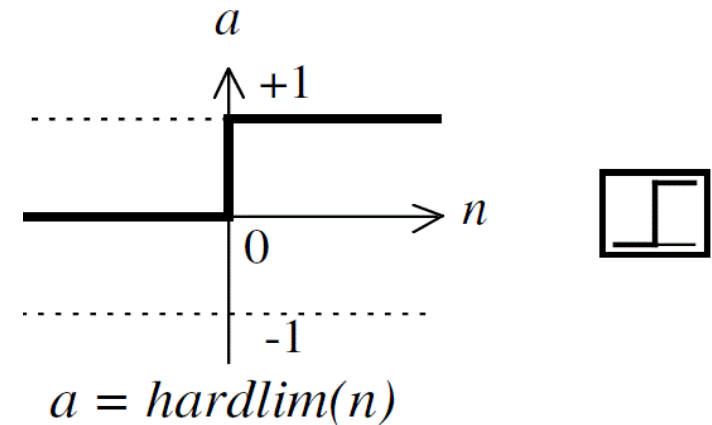
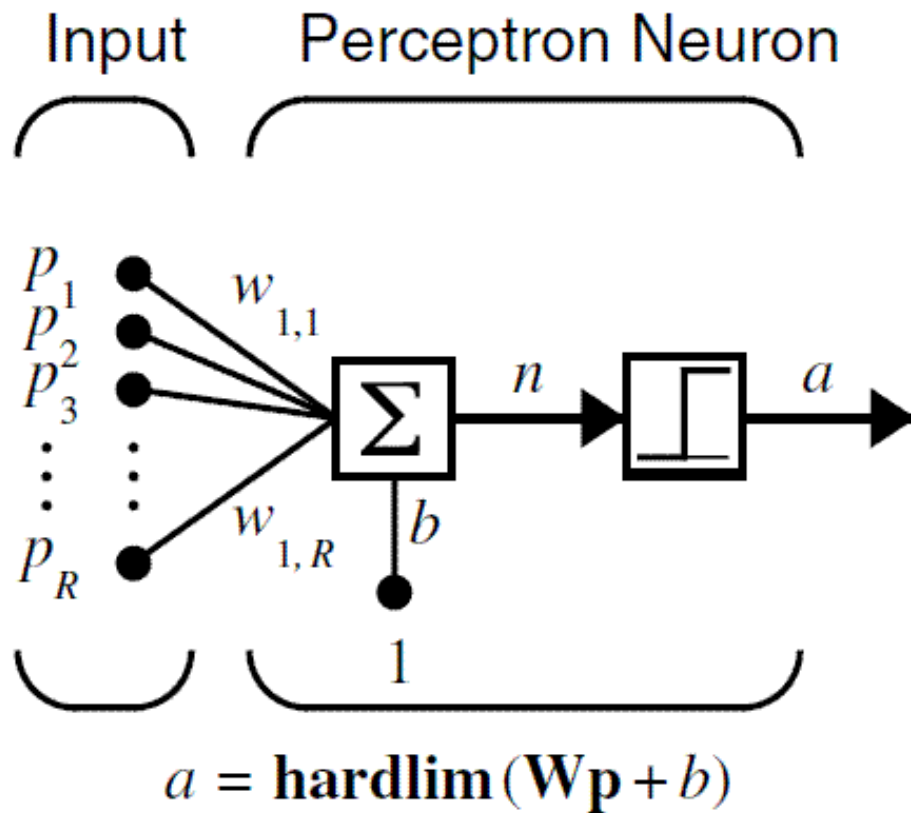
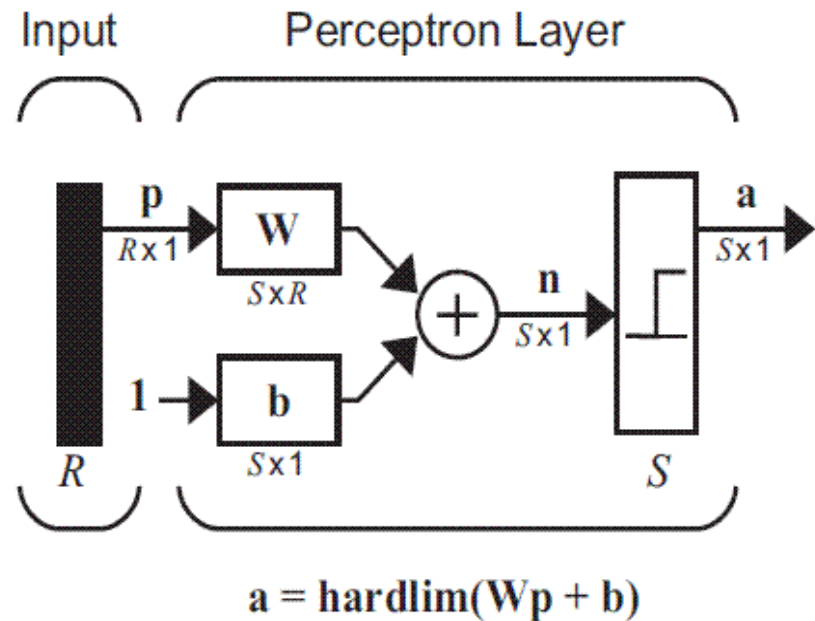
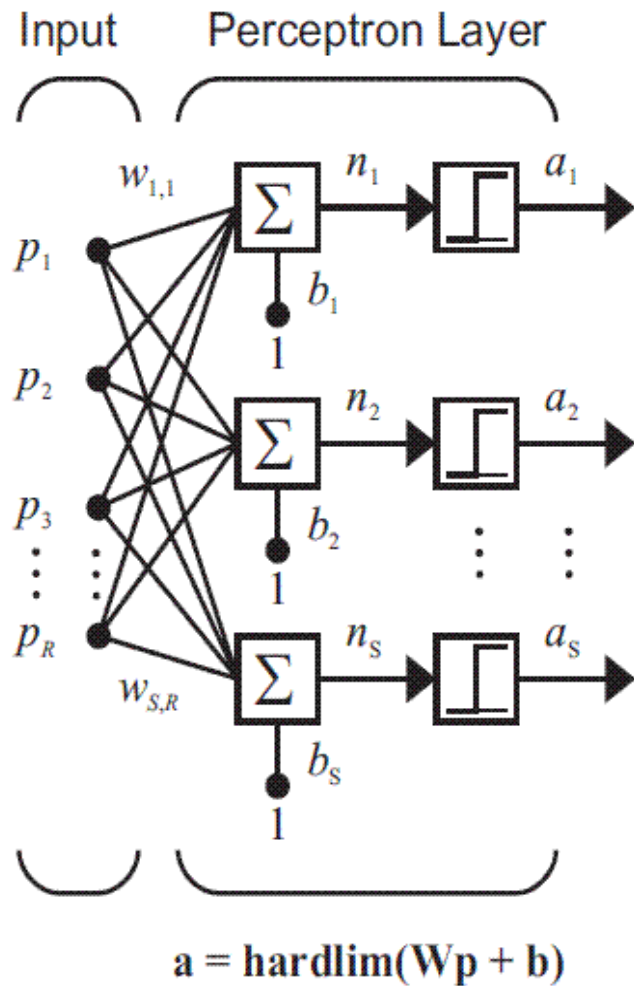


Fig. Ref. NN Toolbox User's Guide Version 6

# Multi-Neuron Perceptron



Where

$R$  = number of elements in input

$S$  = number of neurons in layer

# Multi-Neuron Perceptron

พิจารณาเมตริกซ์น้ำหนักประสาท

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1R} \\ w_{21} & w_{22} & \cdots & w_{2R} \\ \vdots & \vdots & & \vdots \\ w_{S1} & w_{S2} & \cdots & w_{SR} \end{bmatrix}$$

เพื่อความสะดวกในบางครั้ง เราจะเขียนแยกให้อยู่ในรูปเมตริกซ์แถวที่  $i$  ของ  $\mathbf{W}$  คือ

$${}_i\mathbf{W} = \begin{bmatrix} w_{i1} & w_{i2} & \cdots & w_{iR} \end{bmatrix}$$

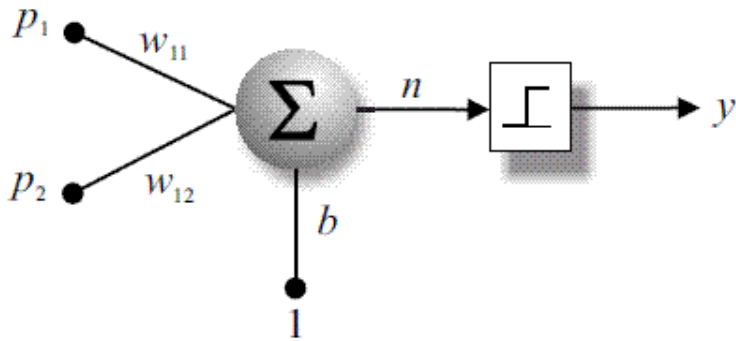
ดังนั้นสามารถเขียนเมตริกซ์น้ำหนักประสาท  $\mathbf{W}$  ในรูปของเมตริกซ์แถว  ${}_i\mathbf{W}$  ได้ดังนี้

$$\mathbf{W} = \begin{bmatrix} {}_1\mathbf{W} \\ {}_2\mathbf{W} \\ \vdots \\ {}_S\mathbf{W} \end{bmatrix}$$

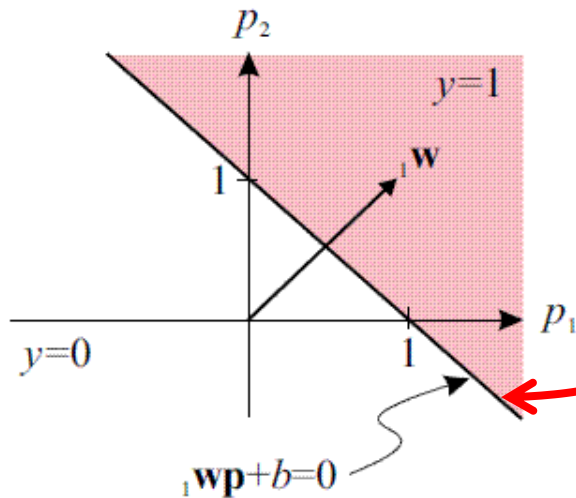
**ดังนั้น**  $y_i = \text{hardlim}(n_i) = \text{hardlim}({}_i\mathbf{w}\mathbf{p} + b_i)$

หรือ  $a_i$

# Single –Neuron Perceptron : Example



รูปที่ 9.2: เพอร์เซ็ปตรอนนิวรอนเดี่ยวแบบ 2 อินพุต



รูปที่ 9.3: ตัวอย่างเส้นแบ่งพื้นที่ในกรณี  ${}_1\mathbf{w} = [1 \ 1]$

$$\begin{aligned} y &= \text{hardlim}(n) = \text{hardlim}(\mathbf{W}\mathbf{p} + b) \\ &= \text{hardlim}({}_1\mathbf{w}\mathbf{p} + b) \\ &= \text{hardlim}(w_{11}p_1 + w_{12}p_2 + b) \end{aligned}$$

$$n = {}_1\mathbf{w}\mathbf{p} + b = w_{11}p_1 + w_{12}p_2 + b = 0$$

**When**  $w_{11} = 1$   $w_{12} = 1$  และ  $b = -1$

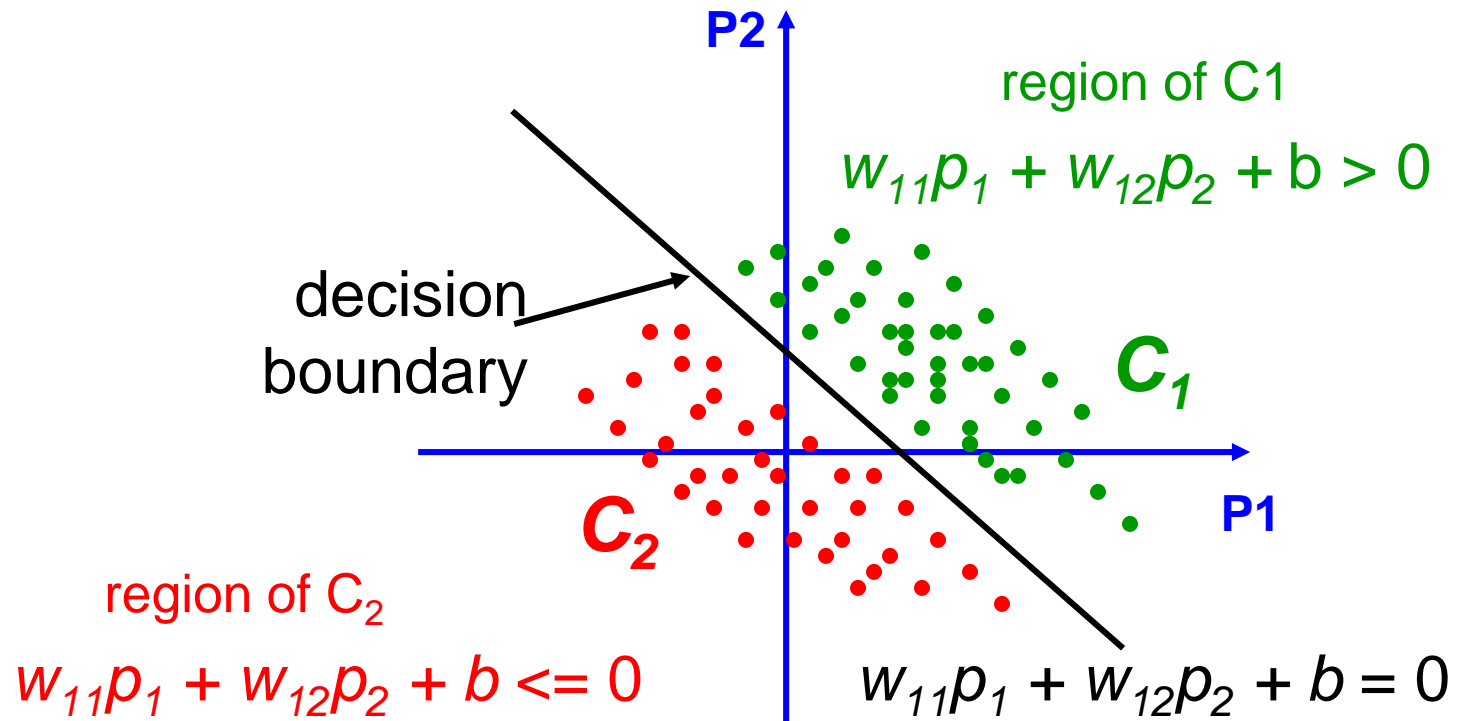
$$n = {}_1\mathbf{w}\mathbf{p} + b = p_1 + p_2 - 1 = 0$$

เส้นแบ่งพื้นที่ตั้งฉากกับเมตริกซ์น้ำหนักประสาทเสมอ

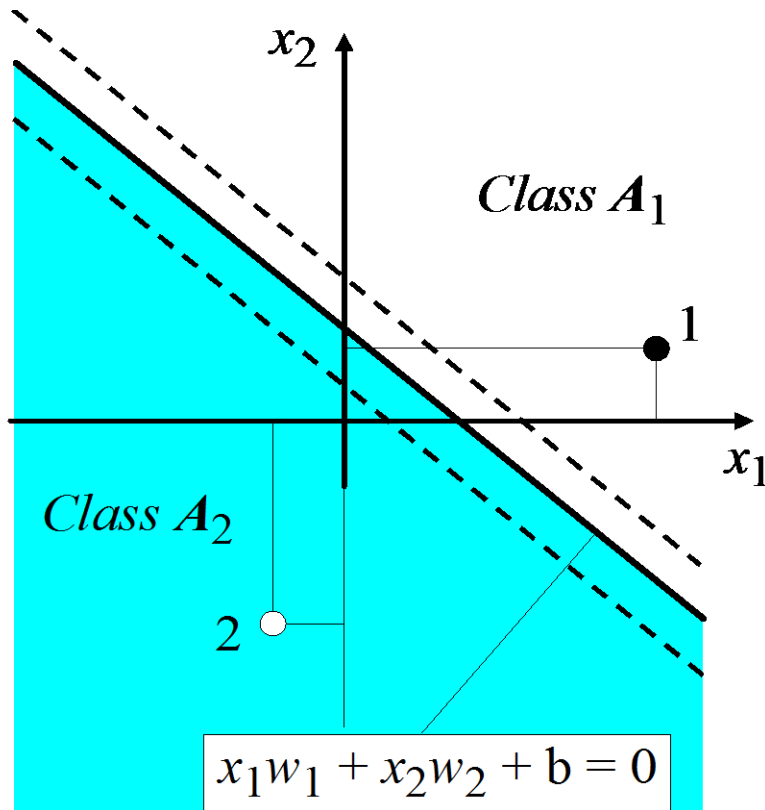
Fig. Ref. ปัญญาเชิงคำนวณ, ผศ.ดร. อาทิตย ศรีแก้ว

# Single –Neuron Perceptron : Example

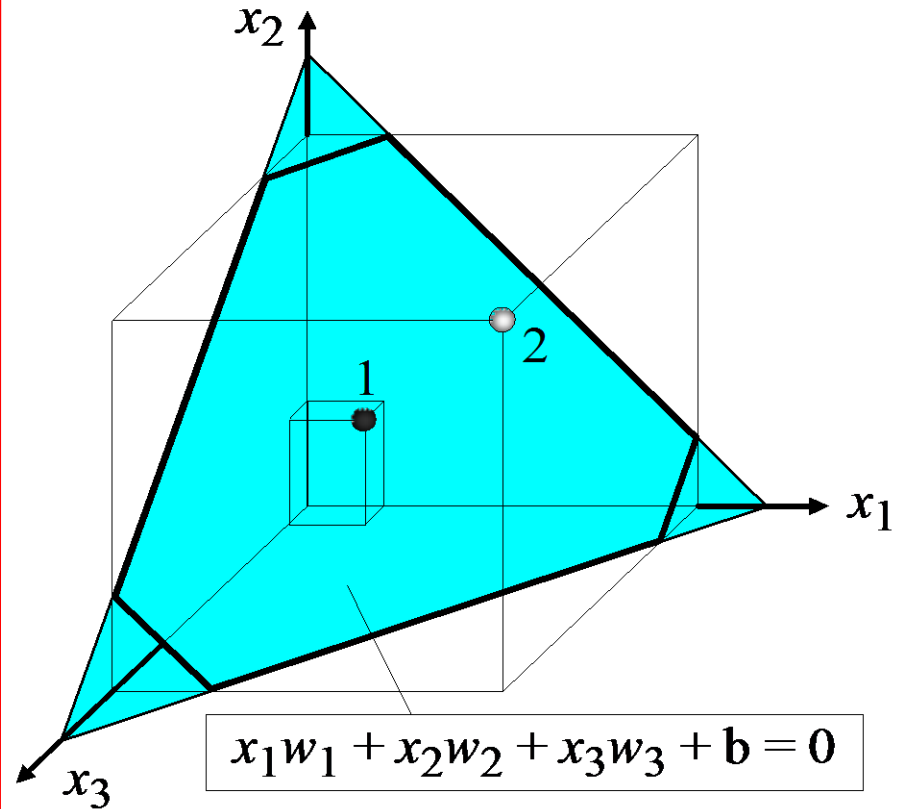
- If input patterns are linear separable, perceptron can use only 1 neuron to separate input into 2 class as follow



# Single Neuron--Linear Separable Problem



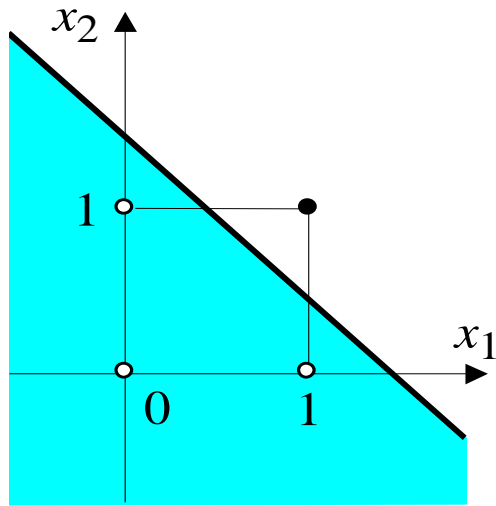
(a) Two-input perceptron.



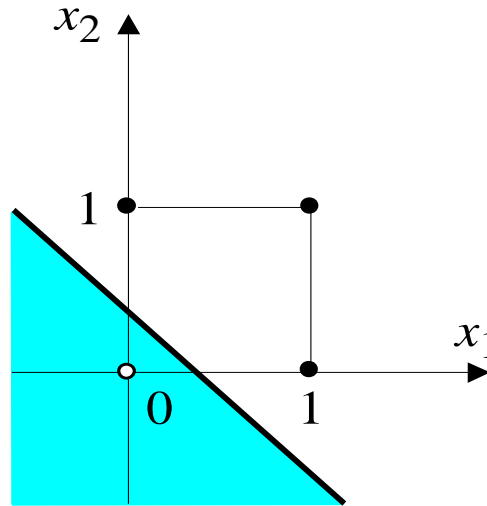
(b) Three-input perceptron.

Fig. Ref. Negnevitsky, Pearson Education, 2005

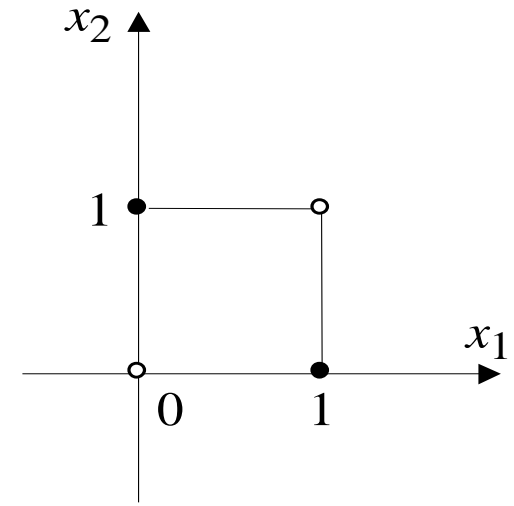
# Basic Logical Operation



(a) *AND* ( $x_1 \cap x_2$ )



(b) *OR* ( $x_1 \cup x_2$ )

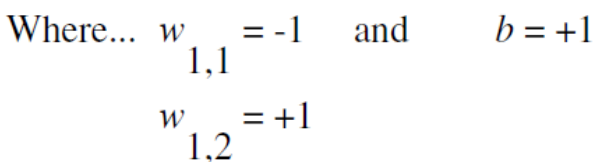


(c) *Exclusive-OR*  
( $x_1 \oplus x_2$ )

- A perceptron can learn the operations *AND* and *OR*, but not *Exclusive-OR (XOR)*.
- *XOR* is non-linear separable problem



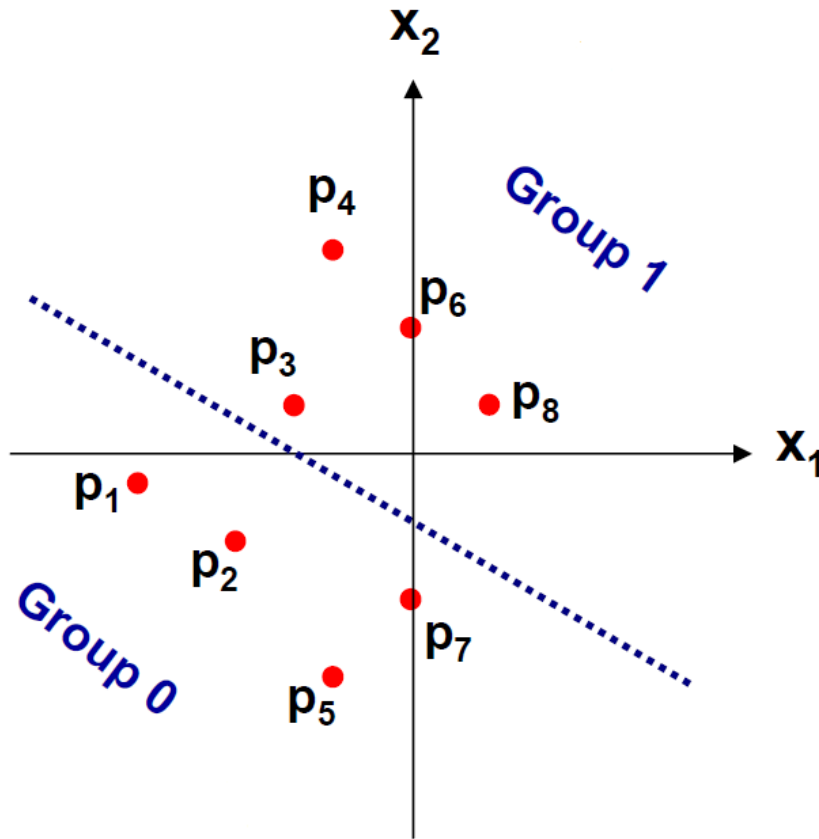
\_\_\_\_\_



- The decision boundary at line L is always perpendicular to the weight matrix **W** and shifted according to the bias **b**

*Fig. Ref. NN Toolbox User's Guide Version 6*

# Linear Separable Problem

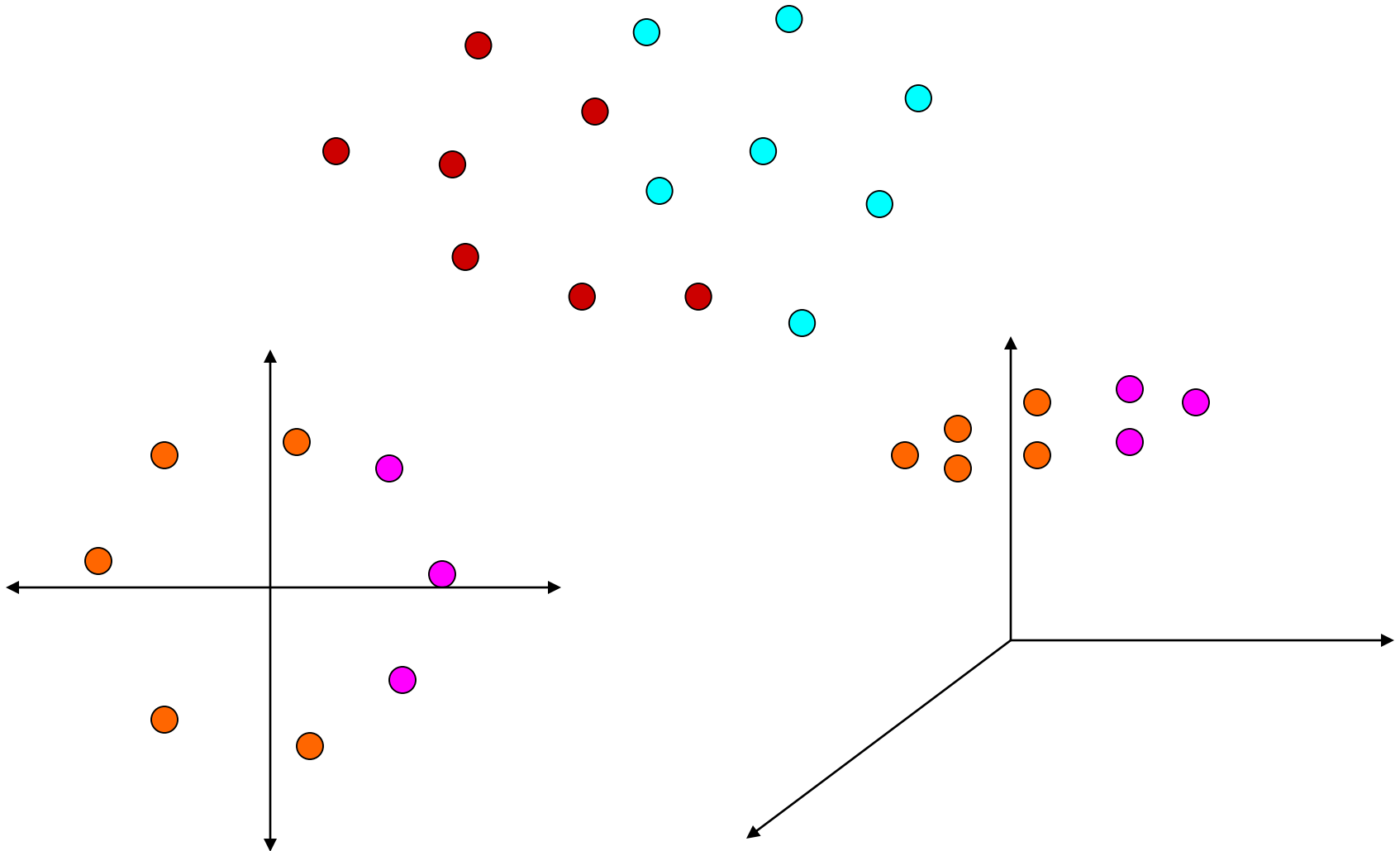


Data	$x_1$	$x_2$	Group
$p_1$	-3	-0.5	0
$p_2$	-2	-1.2	0
$p_3$	-1.5	0.7	1
$p_4$	-1	3	1
$p_5$	-1	-3.5	0
$p_6$	0	2	1
$p_7$	0	-2.5	0
$p_8$	1	0.7	1

Fig. Ref. 2005 Systems Sdn. Bhd.

# Linear or Non-Linear Separable

---



# Perceptron Learning Rule

Input : each pair of  $\{p_1, t_1\} \{p_2, t_2\} \dots \{p_Q, t_Q\}$

Output :  $y$

(Error)  $e = t - y$

$$\mathbf{W}^{new} = \mathbf{W}^{old} + e\mathbf{p}$$

$$\mathbf{b}^{new} = \mathbf{b}^{old} + e$$

นั่นคือ

$$\text{ถ้า } e = 1 \quad \text{แล้ว } \mathbf{W}^{new} = \mathbf{W}^{old} + \mathbf{p}$$

$$\text{ถ้า } e = -1 \quad \text{แล้ว } \mathbf{W}^{new} = \mathbf{W}^{old} - \mathbf{p}$$

$$\text{ถ้า } e = 0 \quad \text{แล้ว } \mathbf{W}^{new} = \mathbf{W}^{old}$$

## Example (9.1 from student ref.)

พิจารณาปัญหาการฝึกสอนเพื่อแยกแยะของตัวปฏิบัติการ **OR** โดยมีเวกเตอร์อินพุตและเวกเตอร์เป้าหมายดังนี้

$$\left\{ \mathbf{p}_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \mathbf{t}_1 = [0] \right\}$$

$$\left\{ \mathbf{p}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \mathbf{t}_2 = [1] \right\}$$

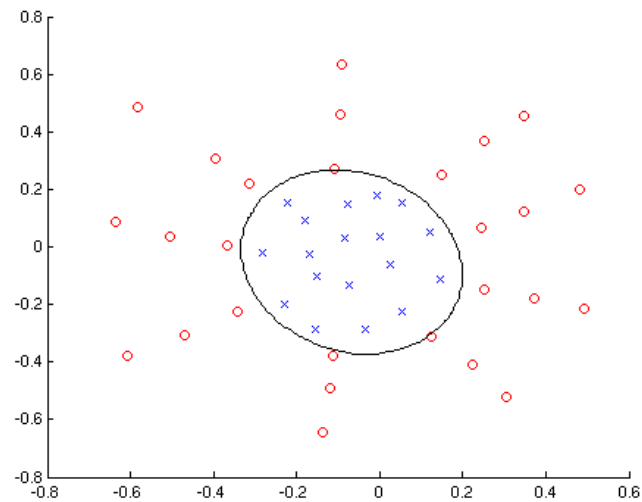
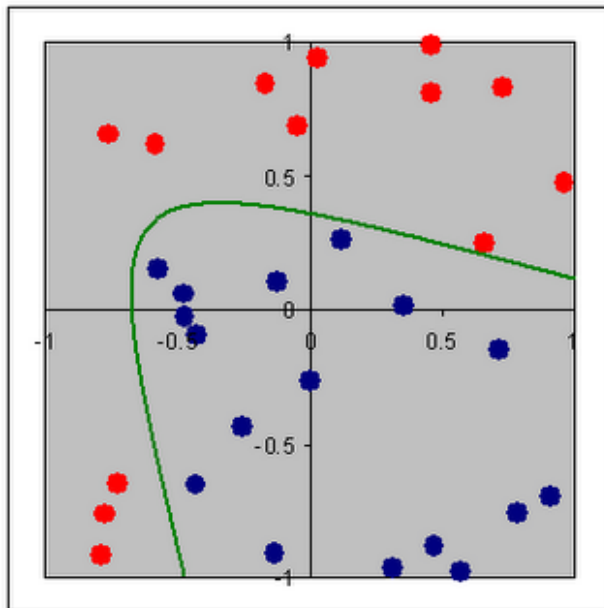
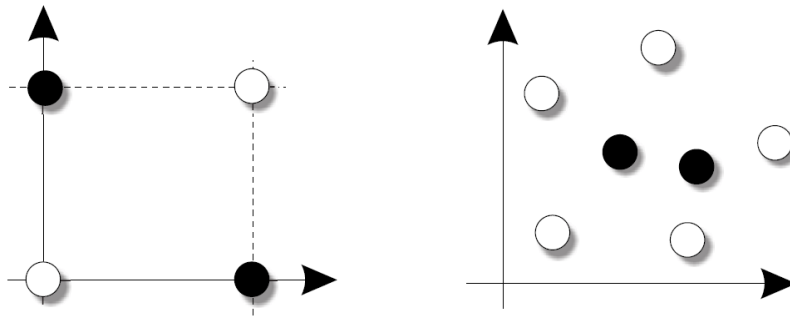
$$\left\{ \mathbf{p}_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \mathbf{t}_3 = [1] \right\}$$

$$\left\{ \mathbf{p}_4 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \mathbf{t}_4 = [1] \right\}$$

เวกเตอร์น้ำหนักประสาท  $\mathbf{w}$  มีการกำหนดค่าเริ่มต้นคือ  $\mathbf{w} = [-1.2, -0.5]^T$

และไบอัส  $b$  มีค่าเริ่มต้นเท่ากับ 1.0

# ข้อจำกัดของเพอร์เซ็ปตรอน



# การปรับแต่งกฎการเรียนรู้แบบเพอร์เซ็ปตรอน

- อัลกอริทึมการปรับส่วนเพิ่ม (increment adaptation algorithm)

$$\mathbf{w}^{new} = \begin{cases} \mathbf{w}^{old} + \alpha e \frac{\mathbf{p}}{\|\mathbf{p}\|^2} & \text{ถ้า } |n| \geq \gamma \\ \mathbf{w}^{old} + \alpha t \frac{\mathbf{p}}{\|\mathbf{p}\|^2} & \text{ถ้า } |n| < \gamma \end{cases}$$

- อัลกอริทึมปรับปรุงการผ่อนคลาย (modified relaxation)

$$\mathbf{w}^{new} = \begin{cases} \mathbf{w}^{old} & \text{ถ้า } |n| \geq \gamma \text{ และ } e = 0 \\ \mathbf{w}^{old} + \alpha \hat{e} \frac{\mathbf{p}}{\|\mathbf{p}\|^2} & \text{อื่นๆ} \end{cases}$$