TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN

MẠNG NORON NHÂN TẠO và GIẢI THUẬT DI TRUYỀN

Neural Network & Genetic Algorithm

Biên soạn: ThS.Phạm Đình Tài
pdtai@ntt.edu.vn
0985.73.39.39

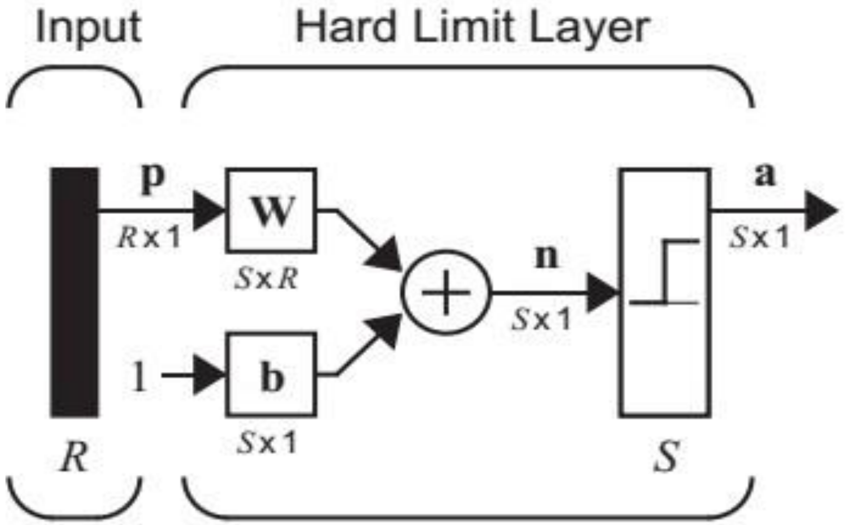
CHƯƠNG 3

Mạng Perceptron

MỤC TIÊU

- ✓ **Kiến trúc mạng perceptron**
- ✓ **Thảo luận về ưu nhược điểm của mạng Perceptron một lớp**

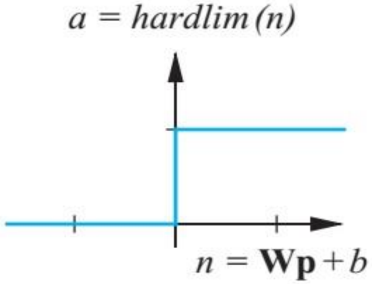
KIẾN TRÚC MẠNG PERCEPTRON



$$\mathbf{a} = \mathbf{hardlim}(\mathbf{Wp}+\mathbf{b})$$

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \vdots & \vdots & & \vdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

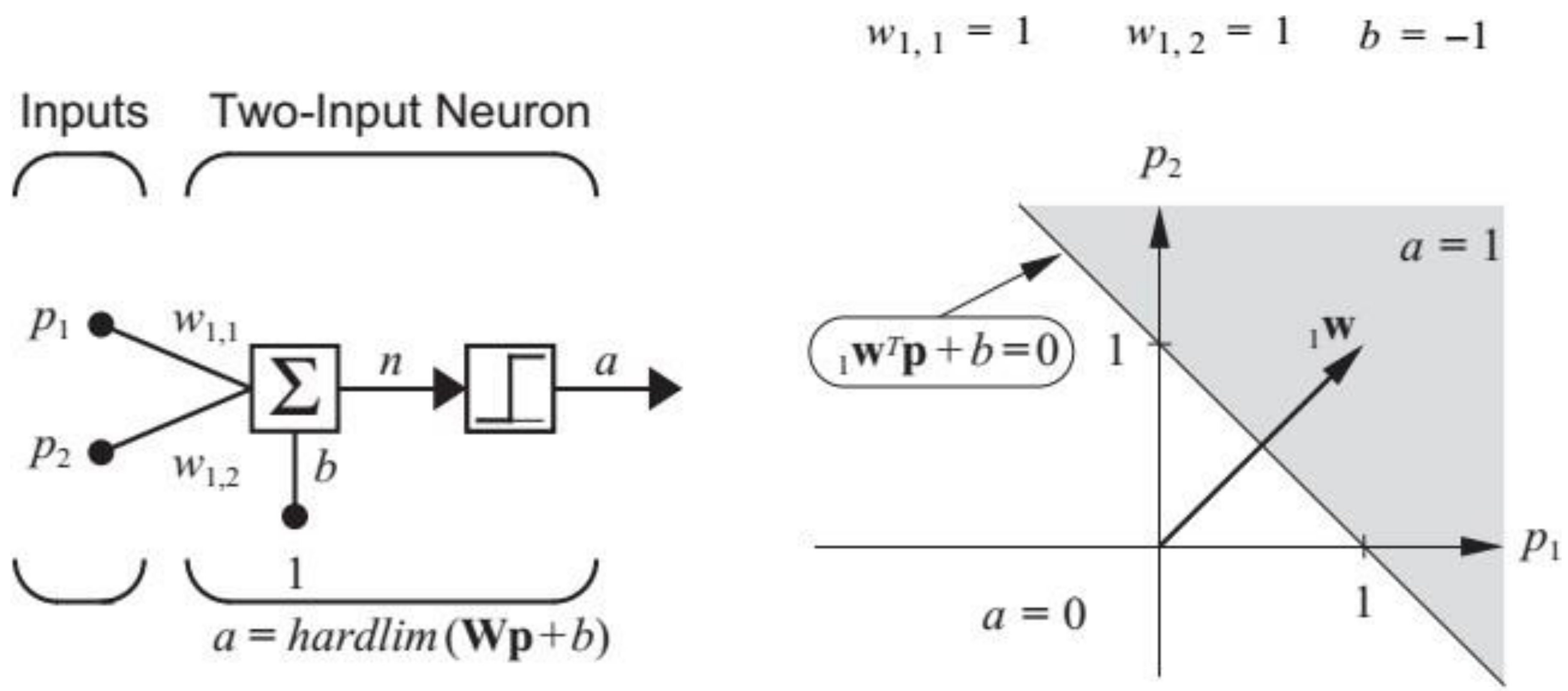
$${}_{i}\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix} \qquad \mathbf{W} = \begin{bmatrix} {}_{1}\mathbf{w}^{T} \\ {}_{2}\mathbf{w}^{T} \\ \vdots \\ {}_{S}\mathbf{w}^{T} \end{bmatrix}$$



$$a_i = hardlim(n_i) = hardlim({}_{i}\mathbf{w}^{T}\mathbf{p} + b_i)$$

$$a = hardlim(n) = \begin{cases} 1 & if \;\; n \geq 0 \\ 0 & otherwise. \end{cases}$$

MẠNG PERCEPTRON HAI ĐẦU VÀO



- Output: $a = hardlim(n) = hardlim(\mathbf{W}\mathbf{p} + b)$

$$= hardlim({_1\mathbf{w}}^T\mathbf{p} + b) = hardlim(w_{1,1}p_1 + w_{1,2}p_2 + b)$$

MẠNG PERCEPTRON HAI ĐẦU VÀO

- Ranh giới quyết định được xác định bởi các vectơ đầu vào mà đầu vào n xác định bằng 0:

$$n = {}_1\mathbf{w}^T\mathbf{p} + b = w_{1,1}p_1 + w_{1,2}p_2 + b = 0.$$

- **Ví dụ:** Cho các giá trị

$$w_{1,1} = 1, w_{1,2} = 1, b = -1$$

MẠNG PERCEPTRON HAI ĐẦU VÀO

- Ranh giới quyết định sau đó có kết quả:

$$n = {}_1\mathbf{w}^T\mathbf{p} + b = w_{1,1}p_1 + w_{1,2}p_2 + b = p_1 + p_2 - 1 = 0$$

- Định nghĩa cho giá trị đầu vào. Ta vẽ 1 đường thẳng, 1 phía của đường thẳng sẽ có giá trị 0, ở phía bên kia nữa còn lại của đường thẳng, kết quả sẽ là 1. Có thể tìm các điểm mà nó giao với trục p_1 và trục p_2. Để tìm điểm p_2, chúng ta thiết lập $p_1 = 0$:

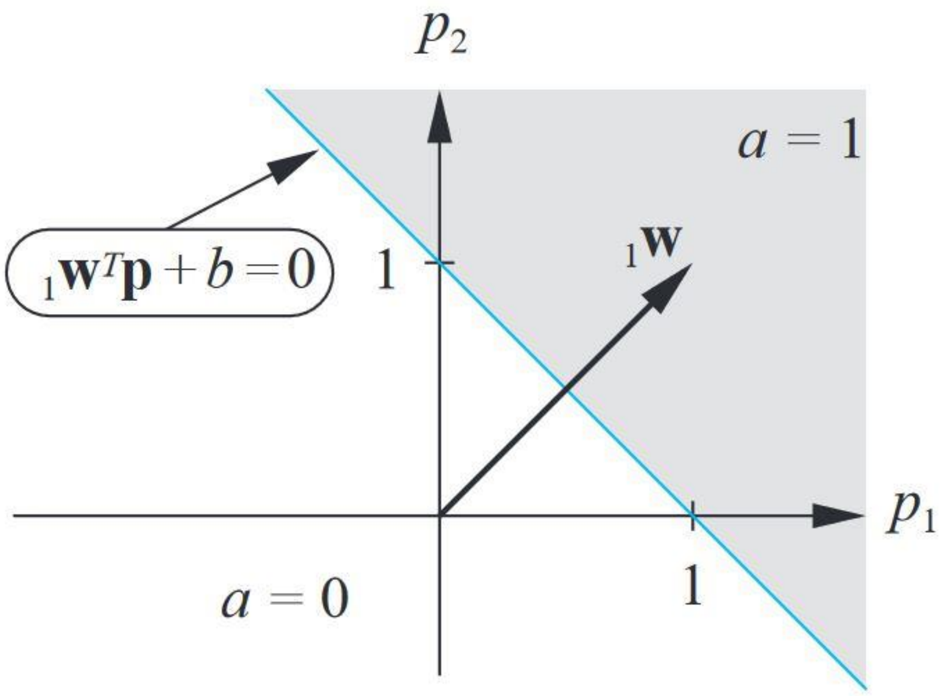
$$p_2 = -\frac{b}{w_{1,2}} = -\frac{-1}{1} = 1 \qquad \text{if } p_1 = 0$$

MẠNG PERCEPTRON HAI ĐẦU VÀO

- Để tìm điểm p_1, chúng ta thiết lập $p_2 = 0$:

$$p_1 = -\frac{b}{w_{1,1}} = -\frac{-1}{1} = 1 \qquad \text{if } p_2 = 0.$$

- Kết quả ranh giới quyết định như hình. Để tìm ra phía nào của ranh giới tương ứng với kết quả đầu ra là 1, chúng ta chỉ cần kiểm tra một điểm. Đối với đầu vào với giá trị p $[2\ 0]^T$, đầu ra mạng sẽ là:

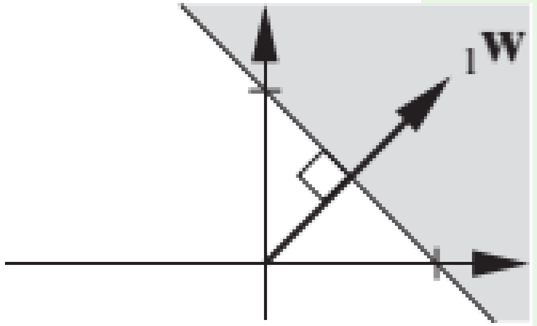


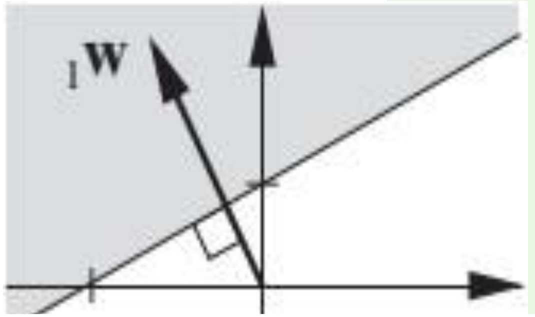
$$a = hardlim({}_1\mathbf{w}^T\mathbf{p} + b) = hardlim\left(\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}2 \\ 0\end{bmatrix} - 1\right) = 1$$

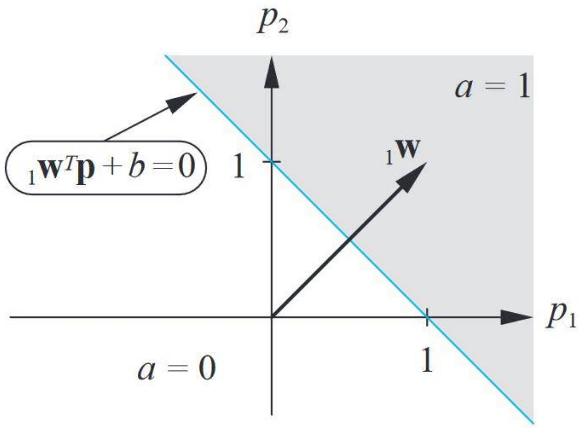
VÙNG RANH GIỚI QUYẾT ĐỊNH

$${}_1\mathbf{w}^T\mathbf{p} + b = 0 \qquad\qquad {}_1\mathbf{w}^T\mathbf{p} = -b$$

- Đối với tất cả các điểm trên biên, tích trong của vectơ đầu vào với vectơ trọng số là như nhau. Điều này ngụ ý rằng tất cả các vectơ đầu vào này sẽ có cùng một hình chiếu lên vectơ trọng số, vì vậy chúng phải nằm trên một đường thẳng trực giao với vectơ trọng số
- Ngoài ra, bất kỳ vectơ nào trong vùng tô bóng sẽ có tích bên trong lớn hơn -b và vectơ trong vùng không tô bóng sẽ có tích bên trong nhỏ hơn -b. Do đó vectơ trọng số sẽ luôn hướng về vùng có đầu ra nơron là 1.



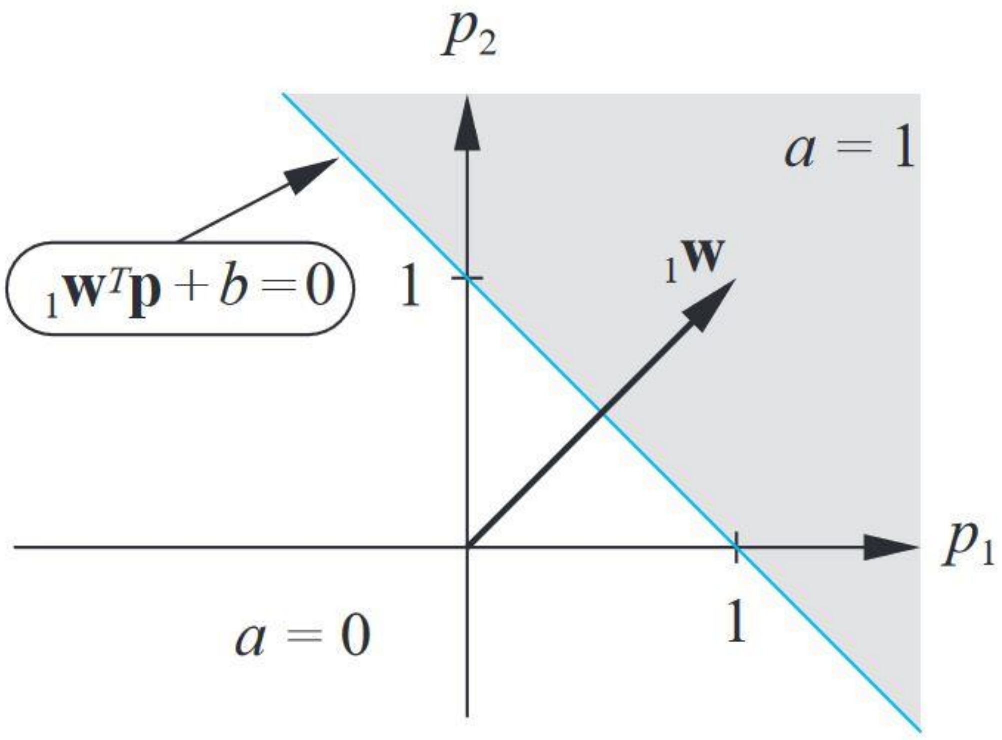




VÙNG RANH GIỚI QUYẾT ĐỊNH

- Sau khi chúng ta đã chọn một vectơ trọng số với hướng góc chính xác, giá trị độ lệch có thể được tính bằng cách chọn một điểm trên đường biên và thỏa mãn

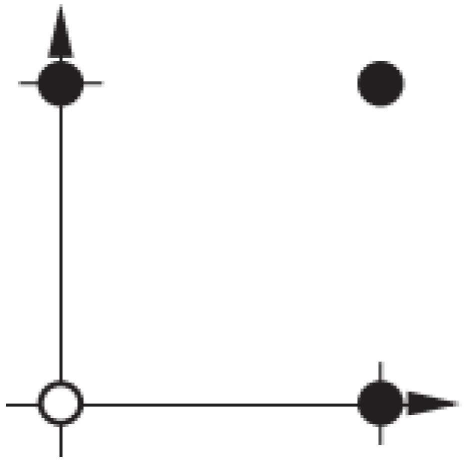
$$ {}_1\mathbf{w}^T\mathbf{p} + b = 0 $$



Toán tử OR

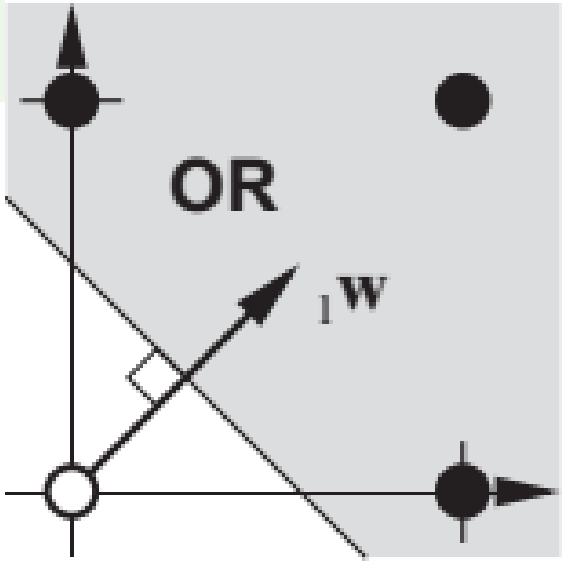
- **Ví dụ:**

$$\left\{\mathbf{p}_1 = \begin{bmatrix}0\\0\end{bmatrix}, t_1 = 0\right\} \quad \left\{\mathbf{p}_2 = \begin{bmatrix}0\\1\end{bmatrix}, t_2 = 1\right\} \quad \left\{\mathbf{p}_3 = \begin{bmatrix}1\\0\end{bmatrix}, t_3 = 1\right\} \quad \left\{\mathbf{p}_4 = \begin{bmatrix}1\\1\end{bmatrix}, t_4 = 1\right\}$$



Toán tử OR

- **Ví dụ:**



- Vector trọng số phải vuông góc với đường thẳng trong phần ranh giới

$$_1\mathbf{w} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}$$

- Lấy một điểm trên vùng ranh giới, để tìm **bias** chúng ta áp dụng

$$_1\mathbf{w}^T\mathbf{p} + b = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} 0 \\ 0.5 \end{bmatrix} + b = 0.25 + \mathrm{b} = 0 \quad \Rightarrow \quad b = -0.25$$

MẠNG PERCEPTRON NHIỀU NEURONS

- Mỗi neuron có một ranh giới quyết định riêng

$$ {}_i\mathbf{w}^T \mathbf{p} + b_i = 0 $$

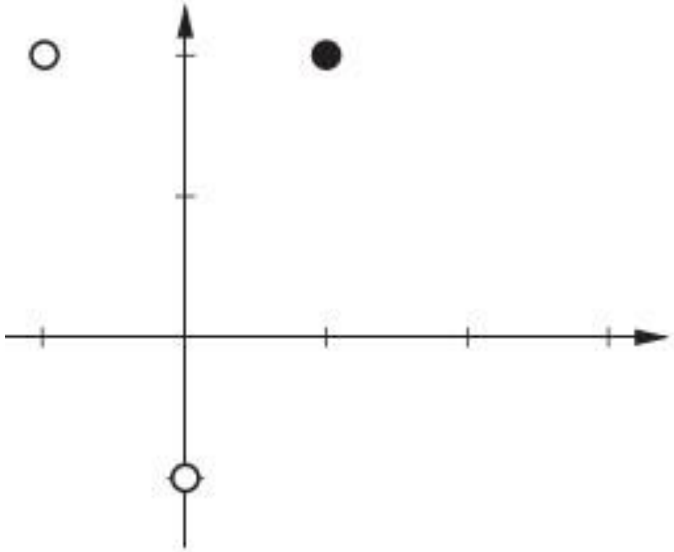
- Mỗi neuron có thể phân các vector đầu vào thành hai loại
- Như vậy một mạng S neuron có thể phân các vector đầu vào thành 2^S

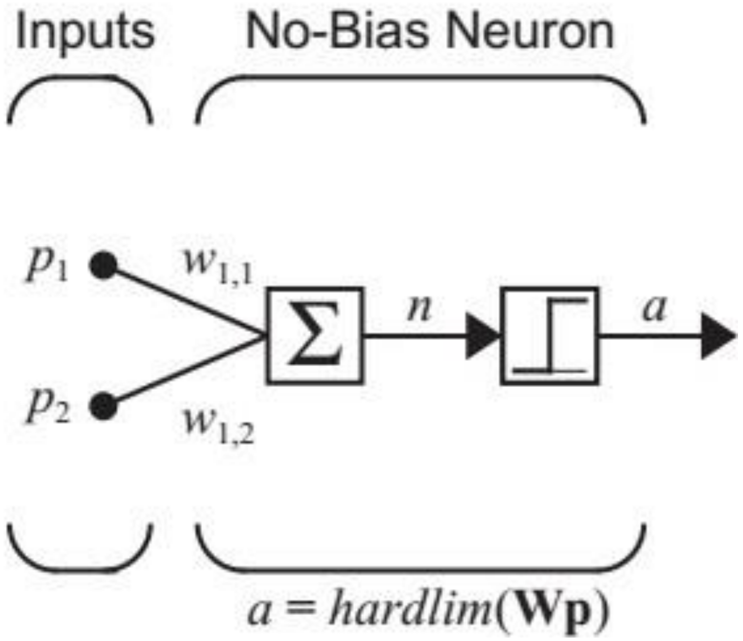
MẠNG PERCEPTRON NHIỀU NEURONS

- **Ví dụ:**

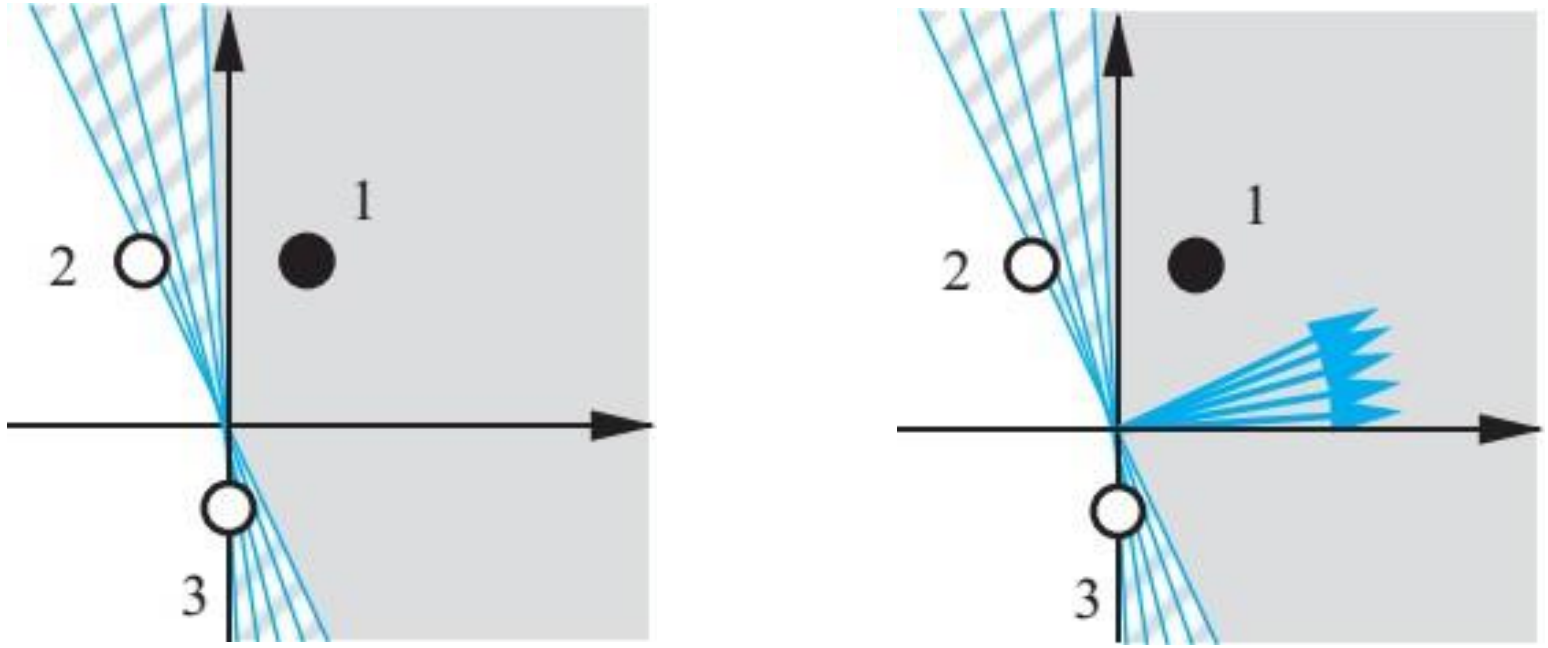
$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \ldots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

$$\left\{\mathbf{p}_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, t_1 = 1\right\} \quad \left\{\mathbf{p}_2 = \begin{bmatrix} -1 \\ 2 \end{bmatrix}, t_2 = 0\right\} \quad \left\{\mathbf{p}_3 = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, t_3 = 0\right\}$$





RANH GIỚI PHÂN TÁCH

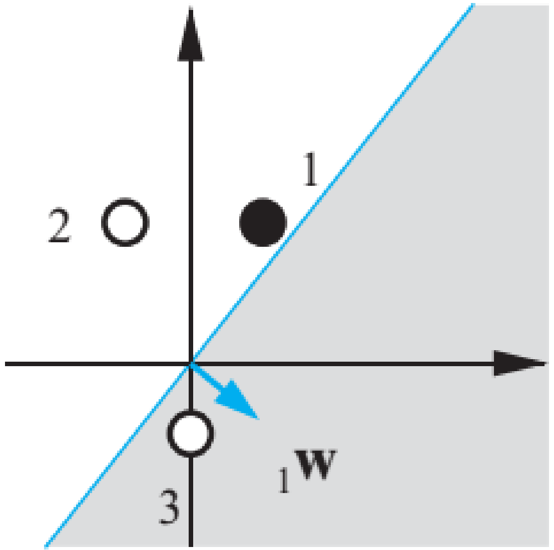


- Khi **bias** = 0, ranh giới đi qua gốc tọa độ
- Tồn tại nhiều đường ranh giới cho phép phân tách các điểm
- Độ dài của vector trọng số không quan trọng

KHỞI TẠO MẠNG NEURONS

- Mạng không có **bias** (b = 0)
- Khởi tạo ngẫu nhiên trọng số

$$_1\mathbf{w} = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix}$$



- Đưa đầu vào vào mạng => phân loại sai

$$a = hardlim({}_1\mathbf{w}^T\mathbf{p}_1) = hardlim\left(\begin{bmatrix} 1.0 & -0.8 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \end{bmatrix}\right)$$

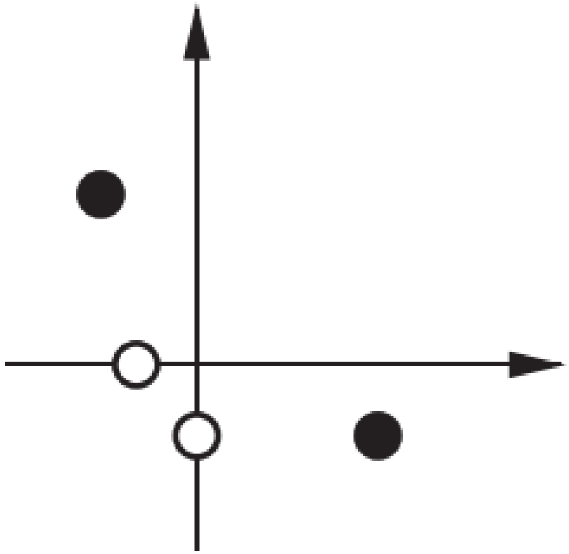
$$a = hardlim(-0.6) = 0$$

- ***“Cần phải thay đổi trọng số để nó chỉ đến gần p_1 hơn”***

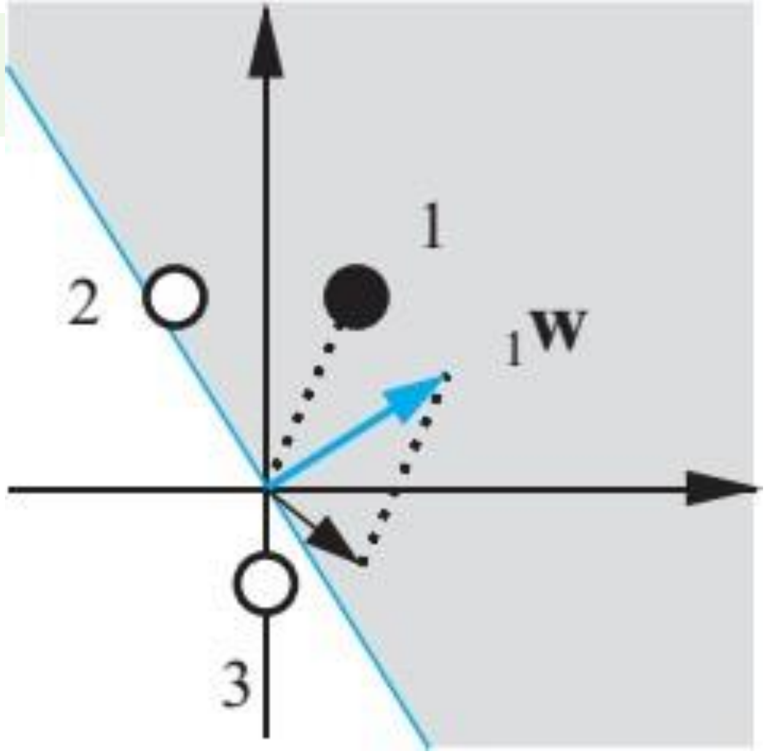
Ý TƯỞNG

- Cho w = p_1: khi đó đảm bảo là sẽ phân loại đúng vector p_1
- Tuy nhiên hoàn toàn có thể xảy ra các trường hợp mà đó không phải là đáp án đúng
- Nếu cộng thêm p_1 vào w, điều này làm cho w hướng đến gần p_1 hơn

If $t = 1$ and $a = 0$, then ${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}$

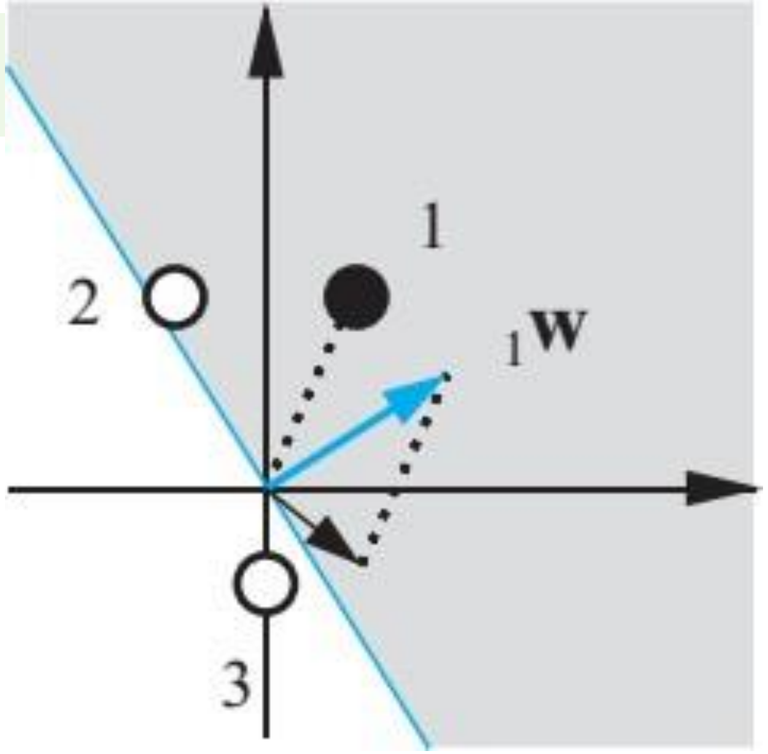


Ý TƯỞNG



$${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}_1 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix}$$

Ý TƯỞNG



$$ {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}_1 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix} $$

$$ a = hardlim({}_1\mathbf{w}^T \mathbf{p}_2) = hardlim\left(\begin{bmatrix} 2.0 & 1.2 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \end{bmatrix} \right) $$

$$ = hardlim(0.4) = 1 . $$

Ý TƯỞNG

$$_1\mathbf{w}^{new} = {_1\mathbf{w}^{old}} + \mathbf{p}_1 = \begin{bmatrix} 1.0 \\ -0.8 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix}$$

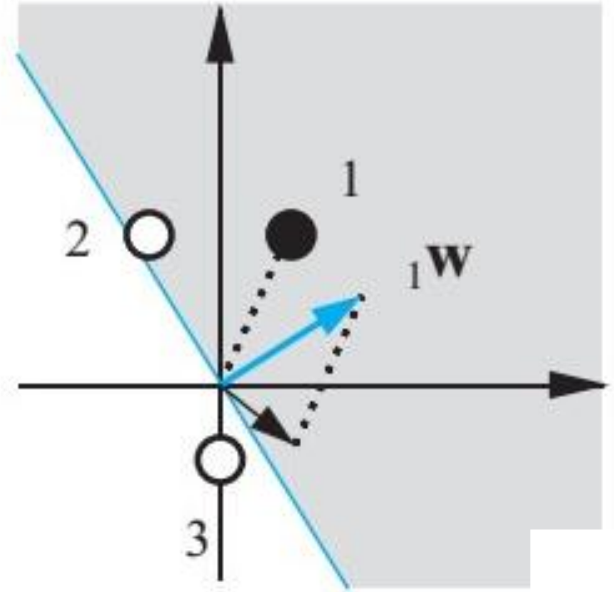
If $t = 0$ and $a = 1$, then $_1\mathbf{w}^{new} = {_1\mathbf{w}^{old}} - \mathbf{p}$

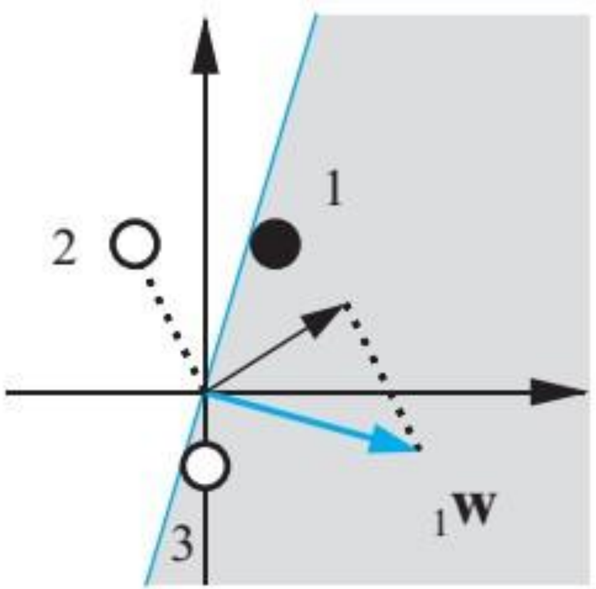
$$_1\mathbf{w}^{new} = {_1\mathbf{w}^{old}} - \mathbf{p}_2 = \begin{bmatrix} 2.0 \\ 1.2 \end{bmatrix} - \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix}$$

$$a = hardlim({_1\mathbf{w}^T}\mathbf{p}_3) = hardlim\left(\begin{bmatrix} 3.0 & -0.8 \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix}\right)$$

$$= hardlim(0.8) = 1 .$$

$$_1\mathbf{w}^{new} = {_1\mathbf{w}^{old}} - \mathbf{p}_3 = \begin{bmatrix} 3.0 \\ -0.8 \end{bmatrix} - \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 3.0 \\ 0.2 \end{bmatrix}$$





CÁC KHẢ NĂNG

$$\text{If } t = 1 \text{ and } a = 0, \text{ then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}.$$

$$\text{If } t = 0 \text{ and } a = 1, \text{ then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} - \mathbf{p}.$$

$$\text{If } t = a, \text{ then } {}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old}.$$

TỔNG QUÁT HÓA

$$\boxed{e = t - a}$$

If $e = 1$, then ${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + \mathbf{p}$.

If $e = -1$, then ${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} - \mathbf{p}$.

If $e = 0$, then ${}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old}$.

$$\boxed{{}_1\mathbf{w}^{new} = {}_1\mathbf{w}^{old} + e\mathbf{p} = {}_1\mathbf{w}^{old} + (t - a)\mathbf{p}}$$

HUẤN LUYỆN PERCEPTRON NHIỀU NEURON

- Các luật trên cập nhật trọng số cho một neuron
- Luật này có thể được khái quát hóa cho mạng perceptron nhiều neuron
- **Ví dụ:** Cập nhật hàng thứ i của ma trận trọng số

$$ {}_{i}\mathbf{w}^{new} = {}_{i}\mathbf{w}^{old} + e_i \mathbf{p} $$

- Cập nhật phần tử thứ i của vector bias

$$ b_i^{new} = b_i^{old} + e_i $$

HUẤN LUYỆN PERCEPTRON NHIỀU NEURON

- Dưới ký pháp ma trận, luật học chuyển thành:

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

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

HUẤN LUYỆN PERCEPTRON NHIỀU NEURON

- **Ví dụ:** Nhận dạng bài toán với 02 thông số đầu vào:

$$\left\{\mathbf{p}_1 = \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}, t_1 = \begin{bmatrix} 0 \end{bmatrix}\right\} \qquad \left\{\mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, t_2 = \begin{bmatrix} 1 \end{bmatrix}\right\}$$

- **Giải:**
- Khởi tạo:
  - ✓ Các trọng số của mạng và vector bias sẽ được khởi tạo ngẫu nhiên với các giá trị nhỏ

$$\mathbf{W} = \begin{bmatrix} 0.5 & -1 & -0.5 \end{bmatrix}, \; b = 0.5$$

HUẤN LUYỆN PERCEPTRON NHIỀU NEURON

- **Bước đầu:**

$$a = hardlim(\mathbf{W}\mathbf{p}_1 + b) = hardlim\left(\begin{bmatrix}0.5 & -1 & -0.5\end{bmatrix}\begin{bmatrix}1\\-1\\-1\end{bmatrix} + 0.5\right)$$

$$= hardlim(2.5) = 1$$

- **Tính toán lỗi:** $e = t_1 - a = 0 - 1 = -1$

- **Cập nhật lại trọng số:** $\mathbf{W}^{new} = \mathbf{W}^{old} + e\mathbf{p}^T = \begin{bmatrix}0.5 & -1 & -0.5\end{bmatrix} + (-1)\begin{bmatrix}1 & -1 & -1\end{bmatrix}$

$$= \begin{bmatrix}-0.5 & 0 & 0.5\end{bmatrix}.$$

- **Cập nhật lại Bias:** $b^{new} = b^{old} + e = 0.5 + (-1) = -0.5$

- **Bước lặp thứ 2 với mẫu thứ 2:**

$$a = hardlim(\mathbf{W}\mathbf{p}_2 + b) = hardlim\left(\begin{bmatrix}-0.5 & 0 & 0.5\end{bmatrix}\begin{bmatrix}1 \\ 1 \\ -1\end{bmatrix} + (-0.5)\right)$$

$$= hardlim(-0.5) = 0$$

- **Tính toán lỗi:** $e = t_2 - a = 1 - 0 = 1$

- **Cập nhật lại** $\mathbf{W}^{new} = \mathbf{W}^{old} + e\mathbf{p}^T = \begin{bmatrix}-0.5 & 0 & 0.5\end{bmatrix} + 1\begin{bmatrix}1 & 1 & -1\end{bmatrix} = \begin{bmatrix}0.5 & 1 & -0.5\end{bmatrix}$

$$b^{new} = b^{old} + e = -0.5 + 1 = 0.5$$

HUẤN LUYỆN PERCEPTRON NHIỀU NEURON

- Bước lặp thứ 3 với mẫu thứ 1:

$$a = hardlim(\mathbf{W}\mathbf{p}_1 + b) = hardlim\left(\begin{bmatrix}0.5 & 1 & -0.5\end{bmatrix}\begin{bmatrix}1\\-1\\-1\end{bmatrix} + 0.5\right)$$

$$= hardlim(0.5) = 1$$

$$e = t_1 - a = 0 - 1 = -1$$

- Cập nhật lại $\mathbf{W}^{new} = \mathbf{W}^{old} + e\mathbf{p}^T = \begin{bmatrix}0.5 & 1 & -0.5\end{bmatrix} + (-1)\begin{bmatrix}1 & -1 & -1\end{bmatrix}$

$$= \begin{bmatrix}-0.5 & 2 & 0.5\end{bmatrix}$$

$$b^{new} = b^{old} + e = 0.5 + (-1) = -0.5$$

HUẤN LUYỆN PERCEPTRON NHIỀU NEURON

- Nếu tiếp tục các bước lặp, có thể tìm được ma trận trọng số và vector bias sao cho cả hai mẫu đầu vào đều được phân loại đúng

- Giải thuật sẽ hội tụ đến một lời giải

Thank you !