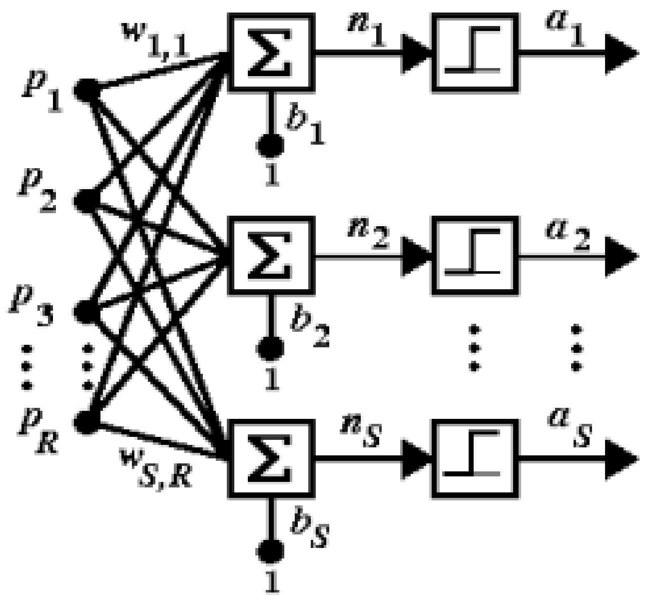
BÀI 13
THỰC NGHIỆM MẠNG NƠ-RON

MẠNG NƠ-RON MỘT LỚP

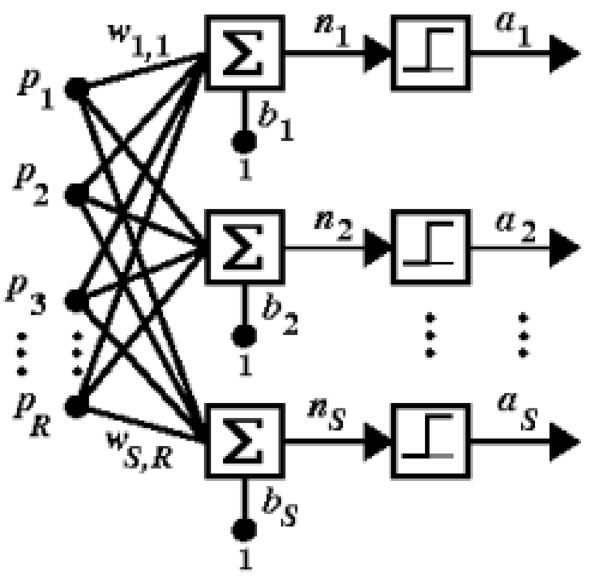
- **Mạng Perceptron**:
  - Cuối những năm 1950, Frank Rosenblatt và các cộng sự phát triển một lớp mạng nơ-ron được gọi là Perceptron.

  - Perceptron là mạng 1 lớp và
được ứng dụng cho bài toán phân lớp nhị phân



MẠNG NƠ-RON MỘT LỚP

- **Mạng Perceptron:**
  - Kiến trúc mạng



- $p = (p_1, p_2, \dots, p_R)^T$ là véctơ tín hiệu vào
- Ma trận trọng số W kích thước $S \times R$:

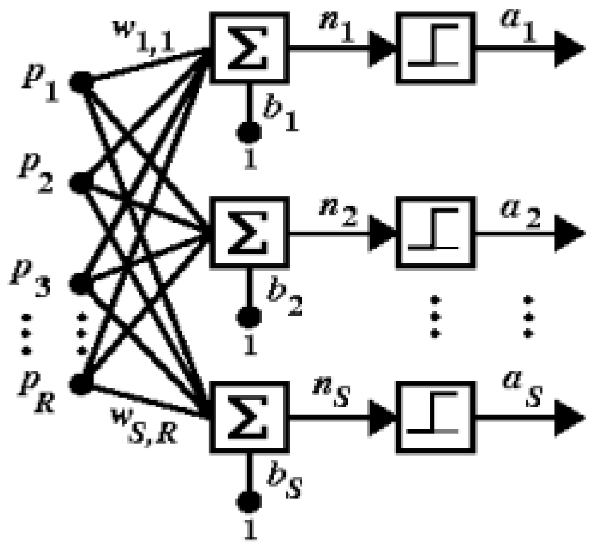
$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ \cdots & \cdots & \cdots & \cdots \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

$b = (b_1, b_2, \dots, b_S)^T$

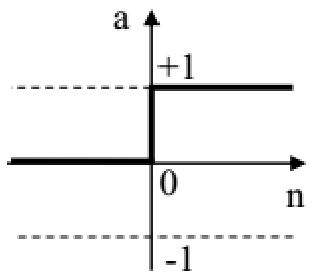
$n_i = w_i^T p + b_i$

MẠNG NƠ-RON MỘT LỚP

- **Mạng Perceptron:**
  - Kiến trúc mạng:



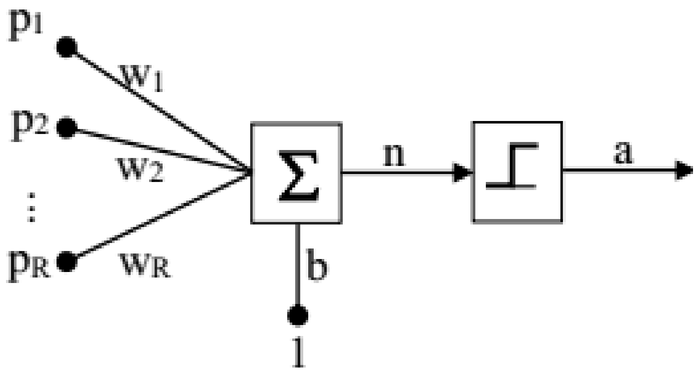
Hàm kích hoạt



Đầu ra thứ i được tính theo a_i = hardlim(n_i), với hardlim(n) = 1 nếu n ≥ 0 và bằng 0 trong các trường hợp còn lại

MẠNG NƠ-RON MỘT LỚP

- **Thuật giải huấn luyện mạng Perceptron:**
  - **Xét trường hợp mạng có 1 nơron**: Trong trường hợp này mạng có nhiều đầu vào và 1 đầu ra

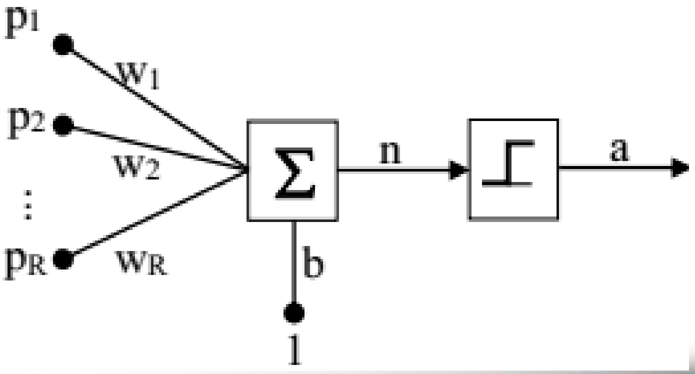


MẠNG NƠ-RON MỘT LỚP

- **Thuật giải huấn luyện mạng Perceptron:**
  - Xét tập $\{p_1, t_1\}, \{p_2, t_2\}, \ldots, \{p_Q, t_Q\}$, trong đó cặp $\{p_i, t_i\}$ là tín hiệu vào và tín hiệu ra mong muốn

  tương ứng với $i = 1, 2, \ldots, Q$.

  - *Ý tưởng thuật toán*: Với mỗi tín hiệu vào p_i sẽ tính được đầu ra thực tế a, giá trị a sẽ được so sánh với t_i, dựa trên kết quả này mà ta sẽ điều chỉnh W và b sao cho **a =**



MẠNG NƠ-RON MỘT LỚP

- **Thuật giải huấn luyện mạng Perceptron**
  - **Input**: $\{p_1, t_1\}, \{p_2, t_2\}, \ldots, \{p_Q, t_Q\}$,
  - **Output**: W, b

Bước 1: Khởi tạo ngẫu nhiên W, b.

Bước 2: Với mỗi p_i tính đầu ra thực tế:

$$a = \text{hardlim}(Wp_i + b)$$

Bước 3: So sánh a với t_i

Nếu $a = t_i$ thì $W^{new} = W^{old}$

Nếu $a \neq t_i$ thì $W^{new} = W^{old} + (t_i - a)p_i^T$

$$b^{new} = b^{old} + (t_i - a)$$

Bước 4: Lặp bước 2, bước 3 cho đến khi $a = t_i, \forall i = 1, 2, \ldots, Q$.

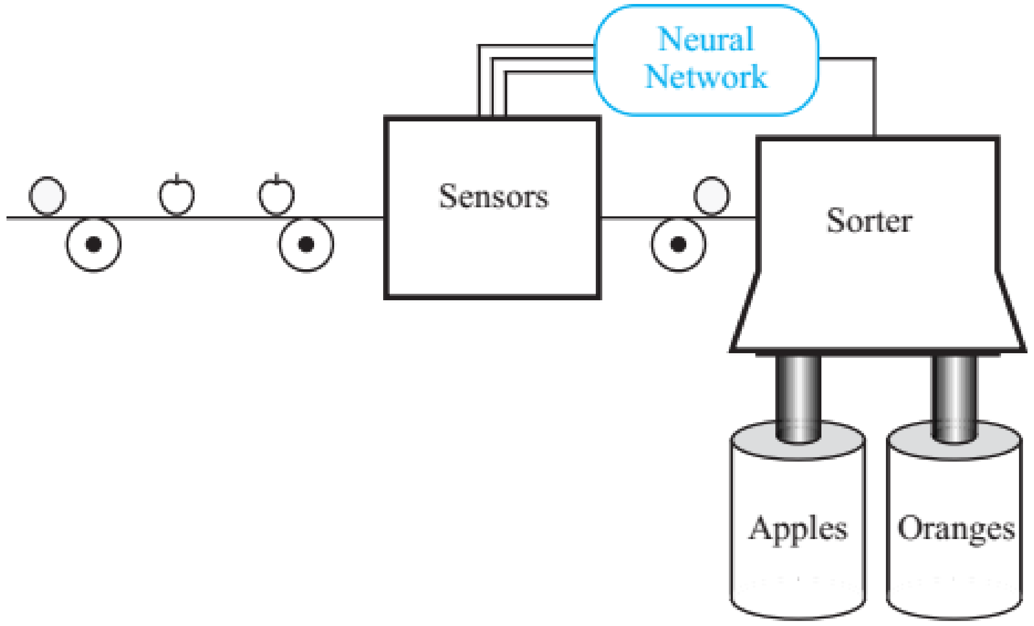
MẠNG NƠ-RON MỘT LỚP

- **Ví dụ minh họa**:

Giả sử có một kho chứa hoa quả:

Cam và táo để lẫn lộn. Thiết kế mô hình

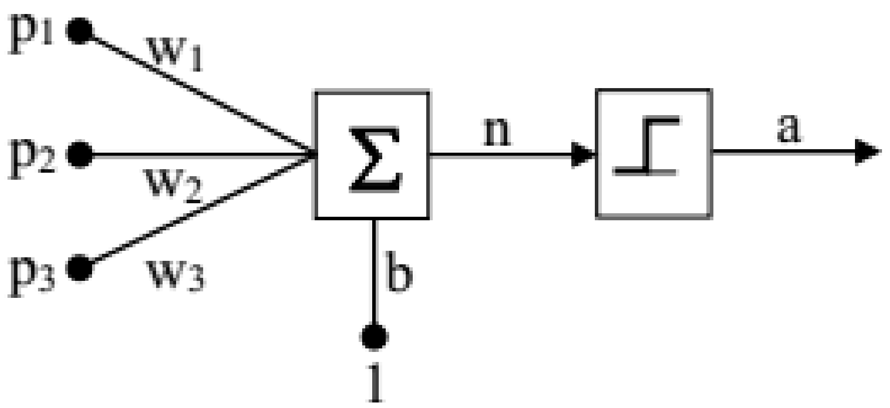
cho phép phân 2 loại quả này.

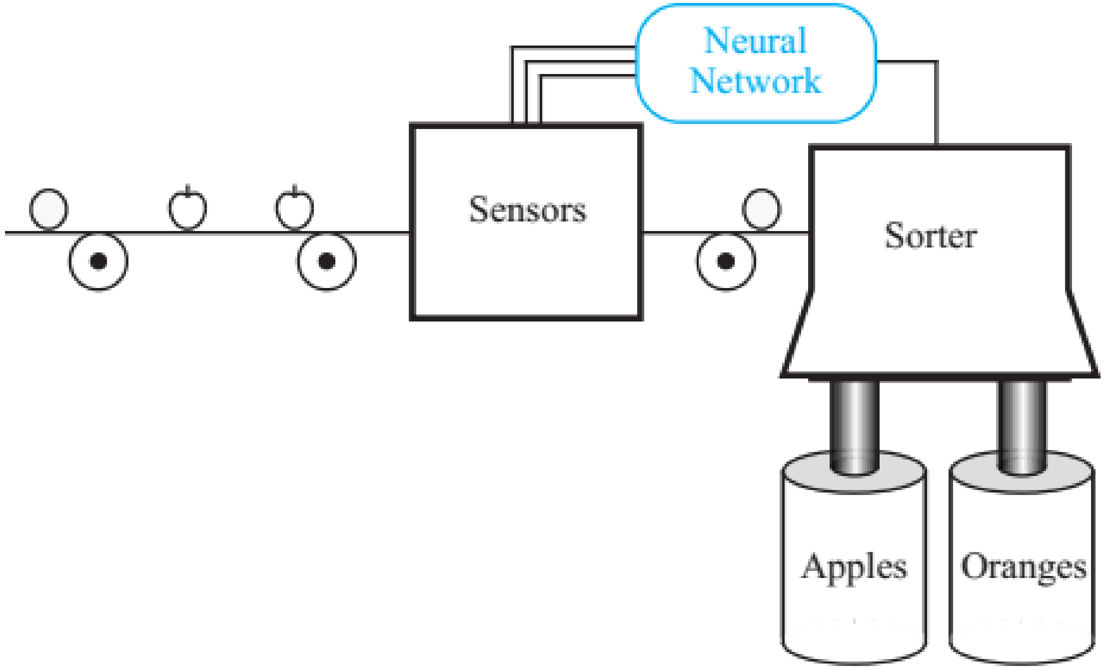


- Mỗi loại quả x(Hình dạng, vỏ, cân nặng)

MẠNG NƠ-RON MỘT LỚP

- **Ví dụ minh họa**: Xét cặp tín hiệu vào/ra: $\{p_1 = (1\ \ -1\ \ -1)^T, t_1 = 0\}$, $\{p_2 = (1\ \ 1\ \ -1)^T, t_2 = 1\}$





MẠNG NƠ-RON MỘT LỚP

- **Ví dụ minh họa**: Xét cặp tín hiệu vào/ra:

$\{p_1 = (1\ \ -1\ \ -1)^T, t_1 = 0\}, \{p_2 = (1\ \ 1\ \ -1)^T, t_2 = 1\}$

Khởi tạo bộ trọng số: W = [0,5 -1 -0,5], b = 0,5.

Lần lặp thứ 1:

$$a = \text{hardlim}(Wp_1 + b) = \text{hardlim}\left([0,5 - 1 - 0,5]\begin{bmatrix}1\\-1\\-1\end{bmatrix} + 0,5\right)$$

$$= \text{hardlim}(2,5) = 1$$

Do $a \neq t_1$ nên cập nhật ma trận trọng số:

$W^{new} = W^{old} + (t_1 - a)p_1^T = [0,5\ \ -1\ \ -0,5] + (-1)(1\ \ -1\ \ -1) = [-0,5\ \ 0\ \ 0,5]$

$b^{new} = b^{old} + (t_1 - a) = 0,5 + (-1) = -0,5$

MẠNG NƠ-RON MỘT LỚP

- **Ví dụ minh họa**: Xét cặp tín hiệu vào/ra:

$\{p_1 = (1\ \ -1\ \ -1)^T,\ t_1 = 0\}, \{p_2 = (1\ \ 1\ \ -1)^T,\ t_2 = 1\}$

Lần lặp thứ 1:

$$a = \text{hardlim}(Wp_2 + b) = \text{hardlim}\left([-0{,}5\ 0\ 0{,}5]\begin{bmatrix}1\\1\\-1\end{bmatrix} - 0{,}5\right)$$

$$= \text{hardlim}(-0{,}5) = 0$$

Vì $a \neq t_2$ nên cập nhật ma trận trọng số:

$W^{new} = W^{old} + (t_2 - a)p_2^T = [-0{,}5\ \ 0\ \ 0{,}5] + (1)(1\ \ 1\ \ -1) = [0{,}5\ \ 1\ -0{,}5]$

$b^{new} = b^{old} + (t_2 - a) = -0{,}5 + 1 = 0{,}5$

MẠNG NƠ-RON MỘT LỚP

- **Ví dụ minh họa**: Xét cặp tín hiệu vào/ra:

$\{p_1 = (1 \;\; -1 \;\; -1)^T, t_1 = 0\}, \{p_2 = (1 \;\; 1 \;\; -1)^T, t_2 = 1\}$

Lần lặp thứ 2:

$$a = \text{hardlim}(Wp_1 + b) = \text{hardlim}\left([0{,}5 \; 1 - 0{,}5] \begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix} + 0{,}5\right)$$

$$= \text{hardlim}(0{,}5) = 1 \neq t_1$$

$$W^{new} = W^{old} + (t_1 - a)p_1^T = [0{,}5 \;\; 1 \;\; -0{,}5] + (-1)(1 \;\; -1 \;\; -1) = [-0{,}5 \;\; 2 \;\; 0{,}5]$$

$$b^{new} = b^{old} + (t_1 - a) = 0{,}5 + (-1) = -0{,}5$$

$$a = \text{hardlim}(Wp_2 + b) = \text{hardlim}\left([-0{,}5 \; 2 \; 0{,}5] \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} - 0{,}5\right) = \text{hardlim}(0{,}5)$$

$$= 1$$

Do $a = t_2$ nên giữ nguyên trọng số.

MẠNG NƠ-RON MỘT LỚP

- **Ví dụ minh họa**: Xét cặp tín hiệu vào/ra:

$\{p_1 = (1\ \ -1\ \ -1)^T, t_1 = 0\}, \{p_2 = (1\ \ 1\ \ -1)^T, t_2 = 1\}$

Lần lặp thứ 3:

$$a = \text{hardlim}(Wp_1 + b) = \text{hardlim}\left([-0{,}5\ \ 2\ 0{,}5]\begin{bmatrix}1\\-1\\-1\end{bmatrix} - 0{,}5\right) = 0 = t_1$$

$$a = \text{hardlim}(Wp_2 + b) = \text{hardlim}\left([-0{,}5\ 2\ 0{,}5]\begin{bmatrix}1\\1\\-1\end{bmatrix} - 0{,}5\right) = 1 = t_2$$

MẠNG NƠ-RON MỘT LỚP

- **Ví dụ minh họa**: Xét cặp tín hiệu vào/ra:

$\{p_1 = (1 \;\; -1 \;\; -1)^T, t_1 = 0\}, \{p_2 = (1 \;\; 1 \;\; -1)^T, t_2 = 1\}$

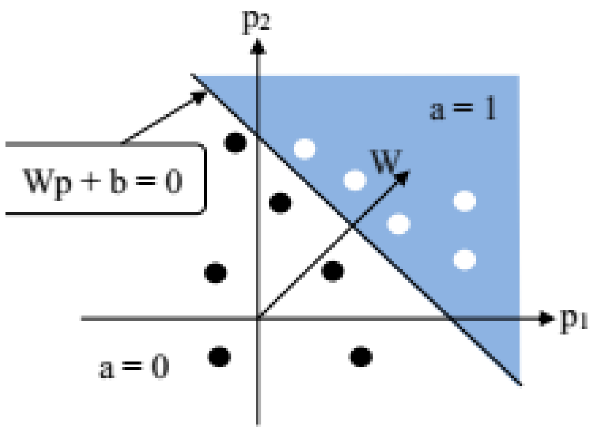
Đến đây thuật toán dừng và ta nhận được bộ trọng số: W = [-0,5 2 0,5], b = -0,5.

Cần gán nhãn cho mẫu mới p = (1 -1 1)^T. Với mẫu này đầu ra a =

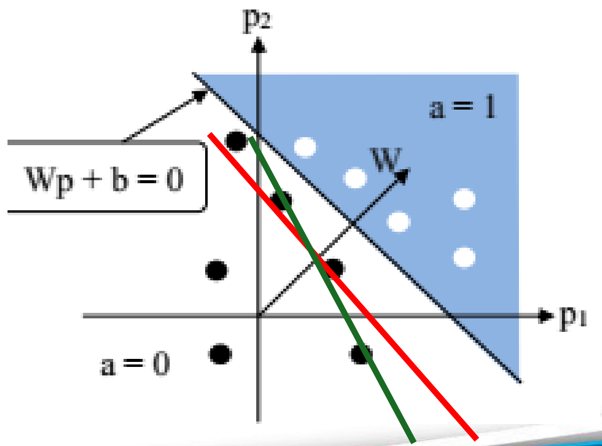
$$\text{hardlim}(Wp + b) = \text{hardlim}\left([-0{,}5\ 2\ 0{,}5]\begin{bmatrix}1\\-1\\1\end{bmatrix} - 0{,}5\right) = 0.$$

MẠNG NƠ-RON MỘT LỚP

- **Nhận xét**:
  - Quá trình huấn luyện mạng Perceptron thực chất là tìm siêu phẳng phân tách tập điểm $\{p_1, p_2, \ldots, p_Q\}$ thành hai vùng, điểm đen và điểm trắng.

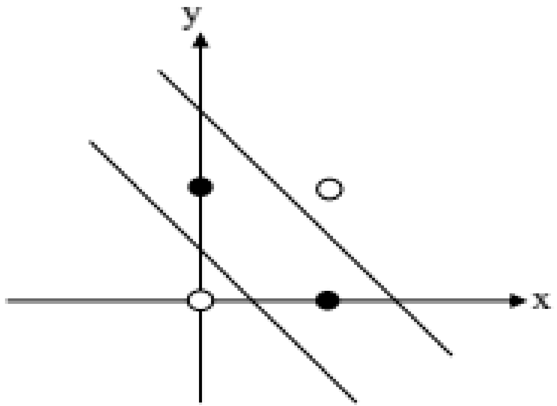


  - Có vô số đường phân tách hai vùng này.



MẠNG NƠ-RON MỘT LỚP

- **Nhận xét**:
  - Mạng này không giải quyết được bài toán XOR

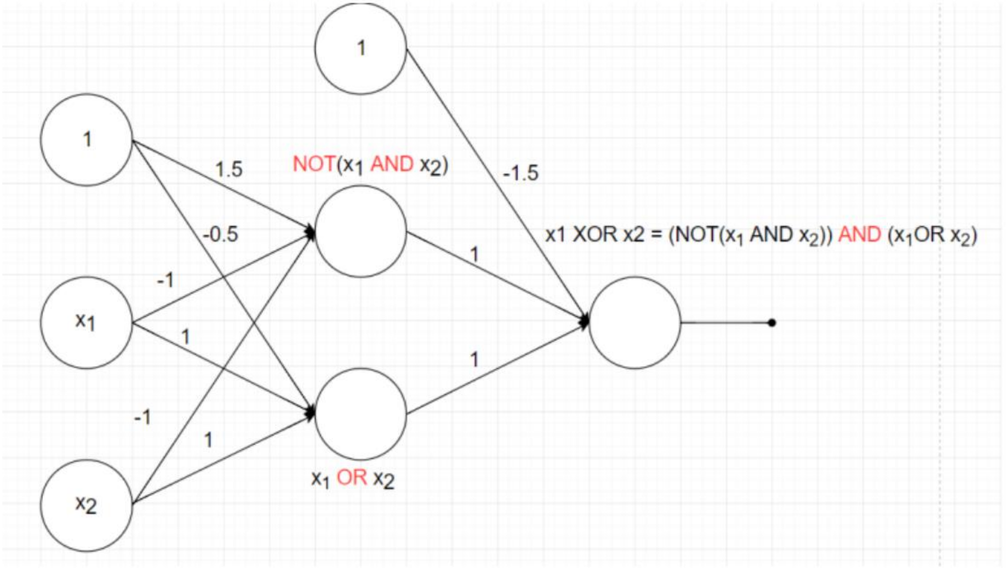


  - Thiết kế mạng nhiều lớp cùng thuật toán lan truyền ngược (backpropagation algorithm) để huấn luyện mạng này.

MẠNG NƠ-RON NHIỀU LỚP

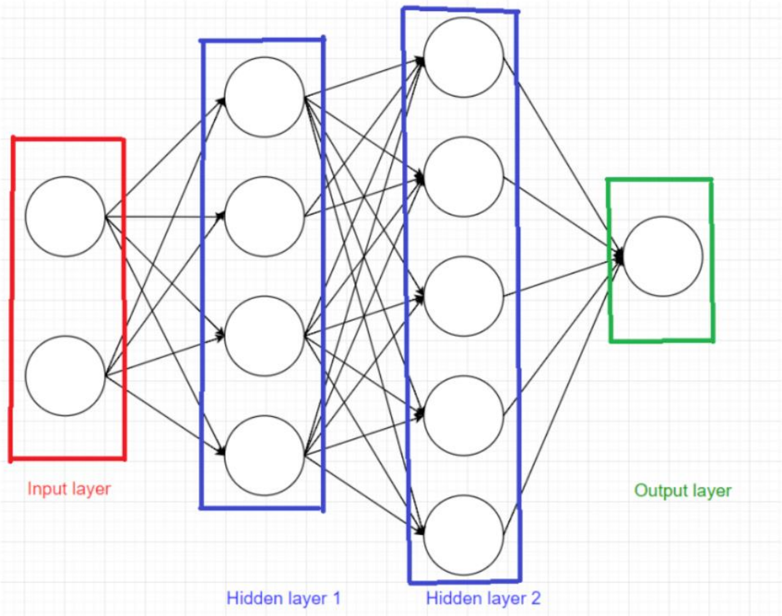
- Mạng nơ-ron hai lớp cho bài toán XOR

A XOR B = (NOT(A AND B) AND (A OR B))



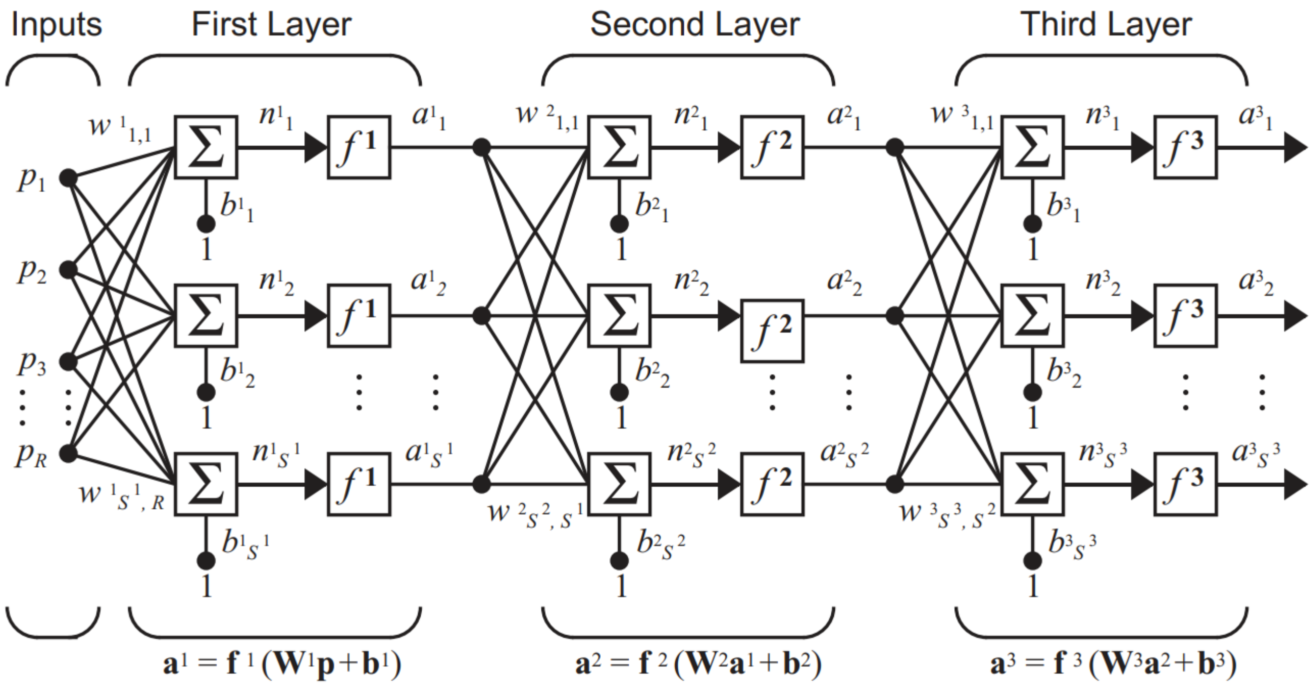
MẠNG NƠ-RON NHIỀU LỚP

- Mô hình mạng nhiều lớp:
  - Lớp input
  - Lớp ẩn
  - Lớp ra: số nơ-ron ở lớp ra phụ thuộc vào yêu cầu của bài toán. Ví dụ với bài toán nhị phân, ta chỉ cần 1 nơ-ron ở lớp ra; bài toán phân loài hoa Iris cần 3 nơ-ron vì hoa Iris có 3 loài.



MẠNG NƠ-RON NHIỀU LỚP

- Mô hình mạng 3 lớp: đầu ra của lớp trước là đầu vào của lớp sau



$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{W}^3\mathbf{f}^2(\mathbf{W}^2\mathbf{f}^1(\mathbf{W}^1\mathbf{p}+\mathbf{b}^1)+\mathbf{b}^2)+\mathbf{b}^3)$$

MẠNG NƠ-RON NHIỀU LỚP

- Huấn luyện mạng:
  - Pha 1: Lan truyền tiến, từ dữ liệu đầu vào, lan truyền qua các lớp để tính đầu ra thực tế.
  - Pha 2: Lan truyền ngược, dữ liệu từ đầu ra (lớp thứ n) được truyền ngược lại lớp thứ n-1. Quá trình này tiếp tục cho đến khi dữ liệu được chuyển qua lớp input. Ở mỗi lớp trọng số được điều chỉnh theo thuật toán Gradient Descent. Mục tiêu của pha này là điều chỉnh trong số sao cho đầu ra thực tế gần với đầu ra mong muốn nhất.

MẠNG NƠ-RON NHIỀU LỚP

- Huấn luyện mạng:
  - Pha 1: Lan truyền tiến

$$z^{(1)} = \begin{bmatrix} z_1^{(1)} \\ z_2^{(1)} \\ z_3^{(1)} \end{bmatrix} = \begin{bmatrix} a_1^{(0)} * w_{11}^{(1)} + a_2^{(0)} * w_{21}^{(1)} + a_3^{(0)} * w_{31}^{(1)} + b_1^{(1)} \\ a_1^{(0)} * w_{12}^{(1)} + a_2^{(0)} * w_{22}^{(1)} + a_3^{(0)} * w_{32}^{(1)} + b_2^{(1)} \\ a_1^{(0)} * w_{13}^{(1)} + a_2^{(0)} * w_{23}^{(1)} + a_3^{(0)} * w_{33}^{(1)} + b_3^{(1)} \end{bmatrix}$$

$$= (W^{(1)})^T * a^{(0)} + b^{(1)}$$

$$a^{(1)} = \sigma(z^{(1)})$$

$$z^{(2)} = (W^{(2)})^T * a^{(1)} + b^{(2)}, a^{(2)} = \sigma(z^{(2)})$$

MẠNG NƠ-RON NHIỀU LỚP

- Huấn luyện mạng:
  - Pha 1: Lan truyền tiến

$$
\begin{aligned}
z^{(2)} &= (W^{(2)})^T * a^{(1)} + b^{(2)} \\
a^{(2)} &= \sigma(z^{(2)}) \\
z^{(3)} &= (W^{(3)})^T * a^{(2)} + b^{(3)} \\
\hat{y} = a^{(3)} &= \sigma(z^{(3)})
\end{aligned}
$$

$$a^{(0)} \Rightarrow z^{(1)} \Rightarrow a^{(1)} \Rightarrow z^{(2)} \Rightarrow a^{(2)} \Rightarrow z^{(3)} \Rightarrow a^{(3)} = \hat{y}$$

MẠNG NƠ-RON NHIỀU LỚP

- Huấn luyện mạng:
  - Pha 2: Lan truyền ngược