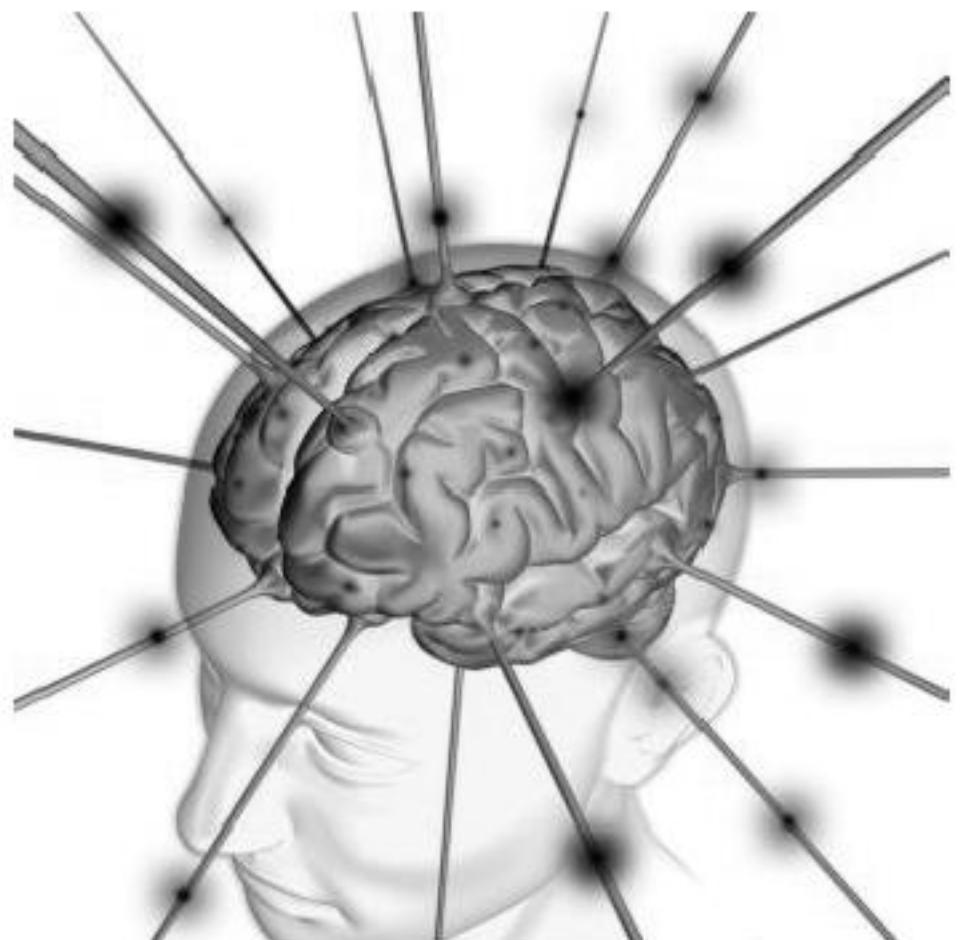


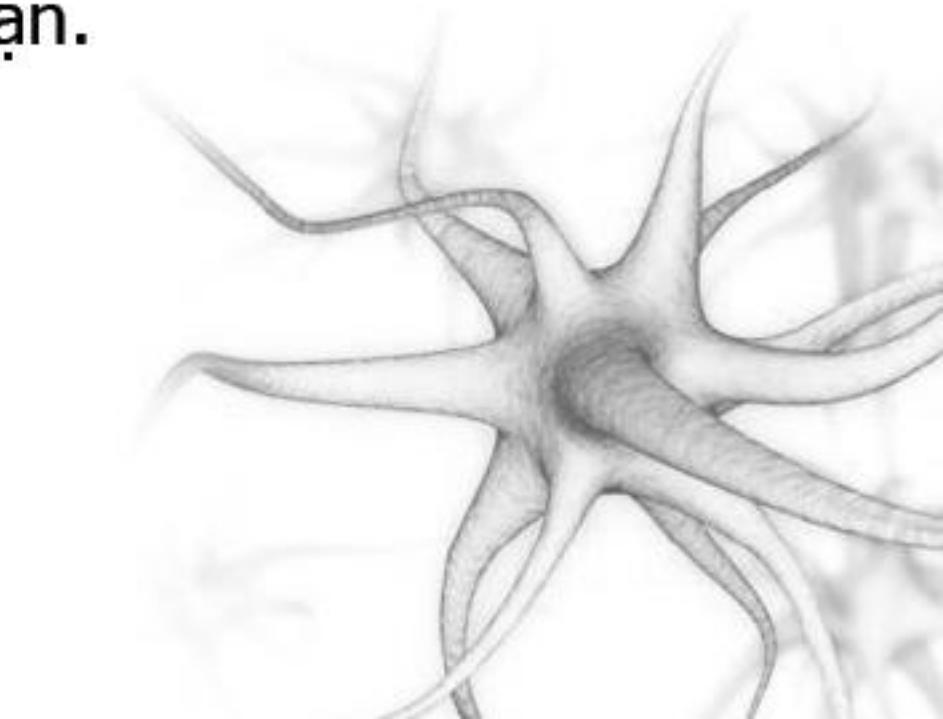
Mạng Neural

GIỚI THIỆU HỆ THỐNG ANN

ĐỘNG LỰC TỪ SINH HỌC



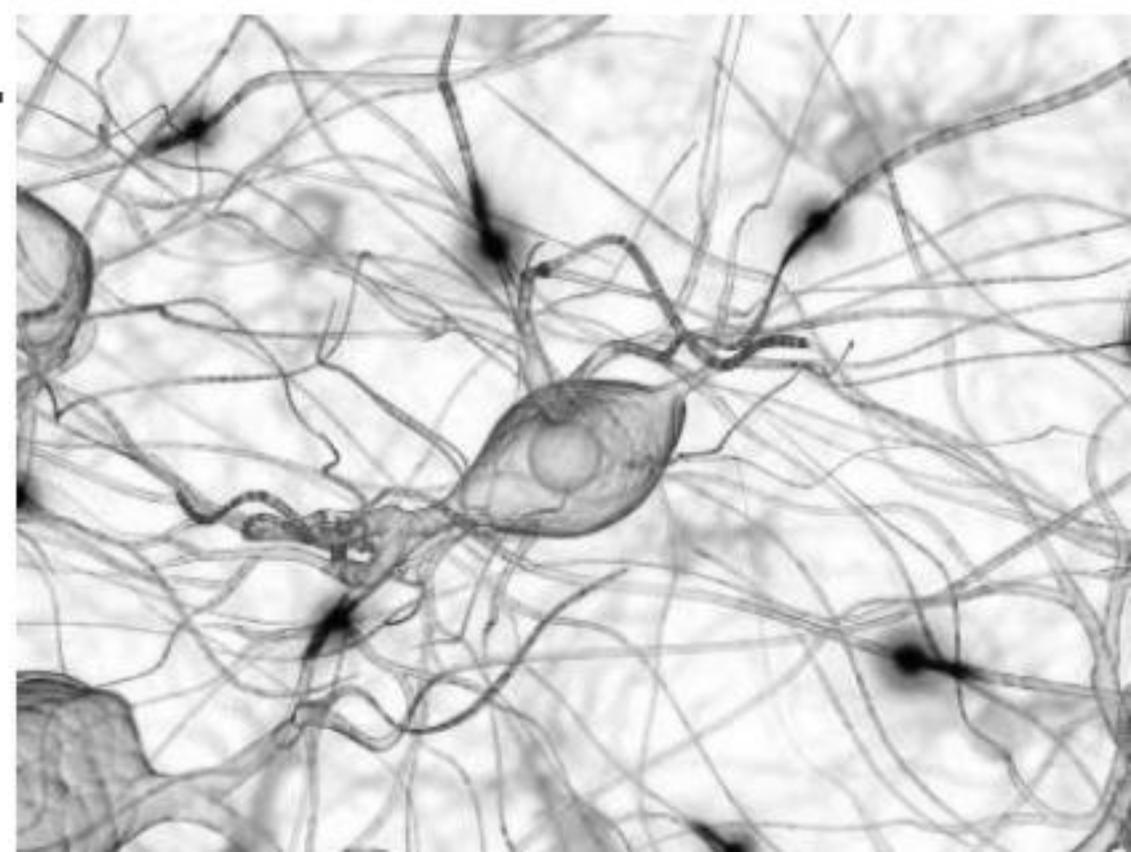
- Bộ não: cỗ máy học hoàn hảo
- Tính toán số học chậm hơn máy tính
- Chỉ mất 10^{-1} giây để nhận ra người thân của bạn.



GIỚI THIỆU HỆ THỐNG ANN

ĐỘNG LỰC TỪ SINH HỌC

- Cấu tạo bộ não: mạng liên kết các nơ-ron thần kinh.
- Mỗi nơ-ron vừa là đơn vị lưu trữ vừa là đơn vị xử lý.
- Nhận vào các tín hiệu thần kinh, lan truyền và xuất kết quả dưới dạng tín hiệu thần kinh.



GIỚI THIỆU HỆ THỐNG ANN

HỆ THỐNG ANN TIÊU BIỂU

ALVINN - *Pomerleau (1993)*

- Hệ thống tự động điều khiển xe trên đường
- Dựa vào ảnh từ camera trước xe
- Đưa ra 30 cấp độ điều khiển lách xe về phía bên trái hoặc phải, hoặc đi thẳng.



SCIENCEphotolibrary

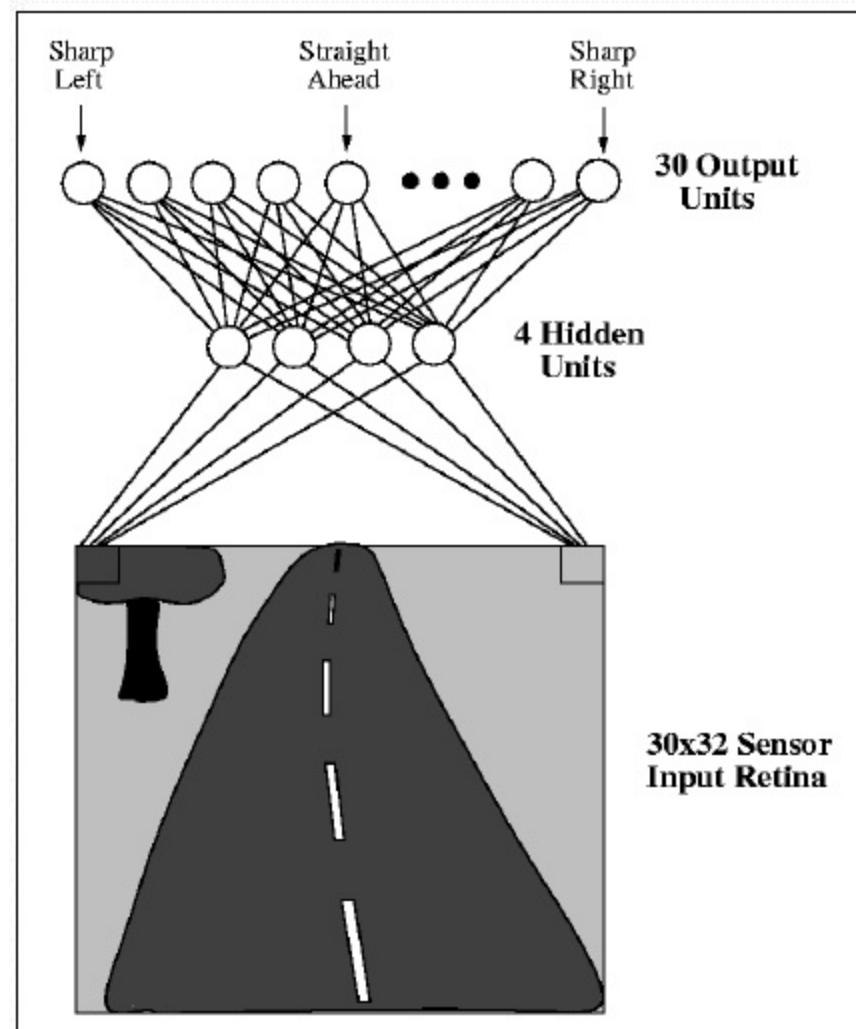


FANFOR D

HỆ THỐNG ANN TIÊU BIỂU

Cơ chế hoạt động của ALVINN

- Mạng nơ-ron truyền thẳng 3 lớp
- Lớp nhập có 30×32 giá trị tương ứng với mức xám các điểm ảnh.
- Lớp ẩn có 4 nút.
- Lớp xuất có 30 nút tương ứng với 30 cấp độ điều khiển.

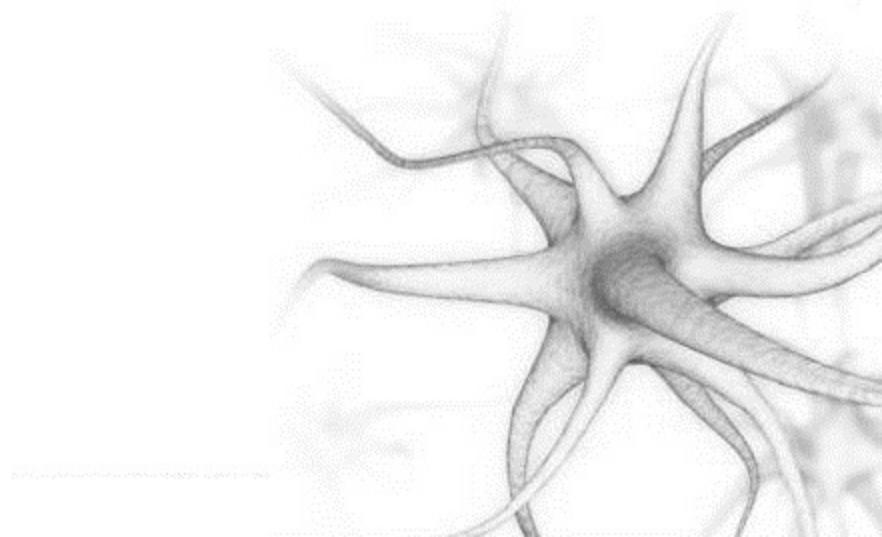


GIỚI THIỆU HỆ THỐNG ANN

HỆ THỐNG ANN TIÊU BIỂU

Cơ chế hoạt động của ALVINN

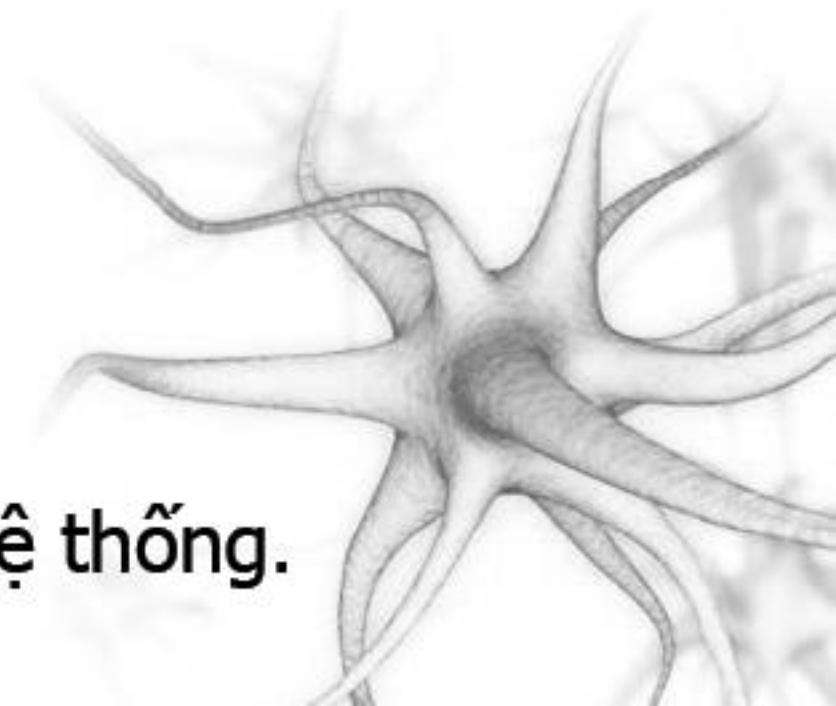
- Huấn luyện: Dữ liệu thực tế từ tài xế điều khiển xe trong phòng 5 phút.
- Kết quả: Xe chạy đúng tuyến và an toàn trên 90 dặm đường thực tế với tốc độ 70 dặm/giờ.



VẤN ĐỀ TƯƠNG THÍCH CỦA ANN

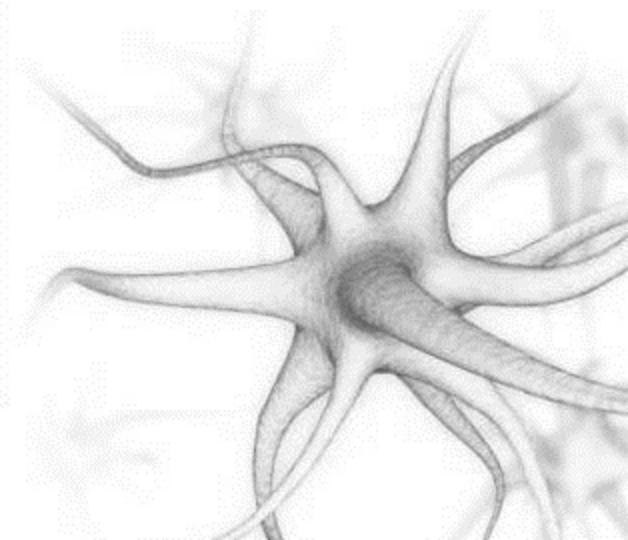
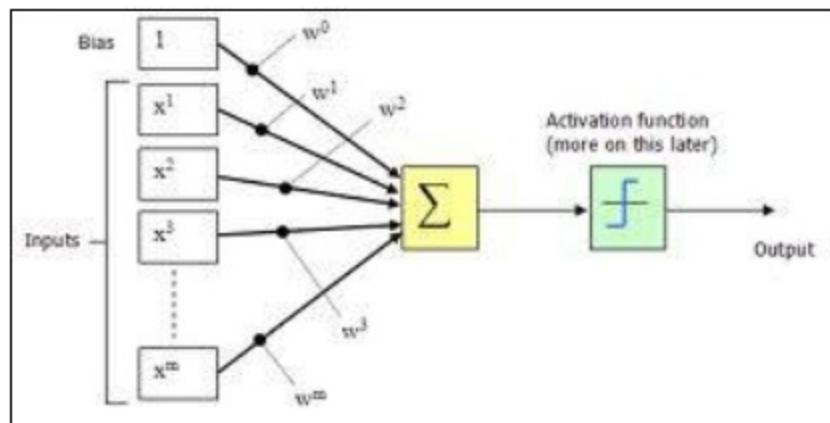
Khi nào thì có thể sử dụng ANN?

- Mẫu thử input có thể xấp xỉ bằng vector.
- Kết quả output có thể xấp xỉ bằng vector.
- Mẫu có thể chứa lỗi, nhiễu.
- Chấp nhận thời gian huấn luyện lâu.
- Thời gian thực thi rất nhanh.
- Không cần quan tâm đến bộ trọng số của hệ thống.



Khái niệm perceptron

- Là thành phần cơ bản của mạng nơ-ron
- Perceptron nhận tín hiệu từ dữ liệu nhập, xử lý và truyền kết quả ra bên ngoài



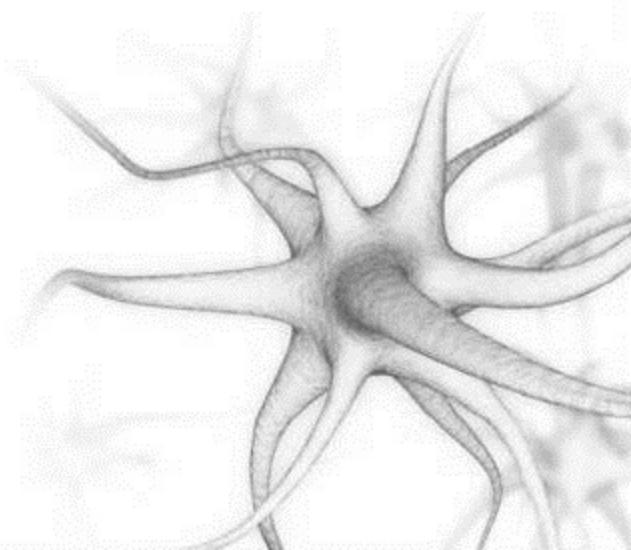
Perceptron (2)

- Giá trị đầu ra của perceptron được tính như sau:

$$o(x_1, \dots, x_n) = \begin{cases} 1 & \text{nếu } w_0 + w_1x_1 + \dots + w_nx_n > 0 \\ -1 & \text{nếu ngược lại} \end{cases}$$

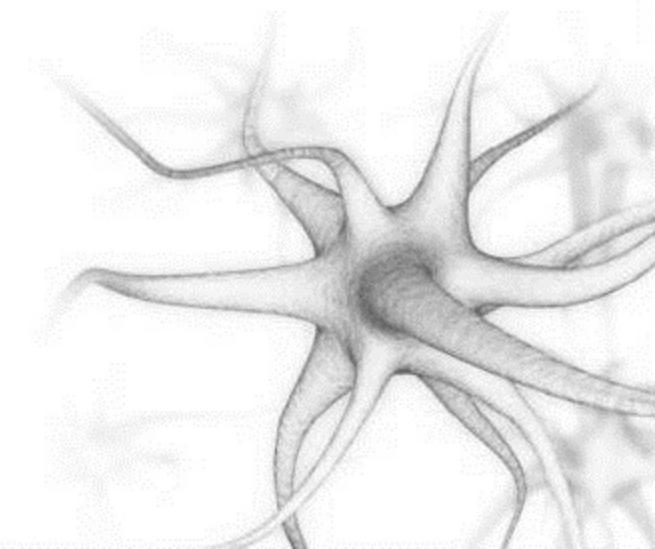
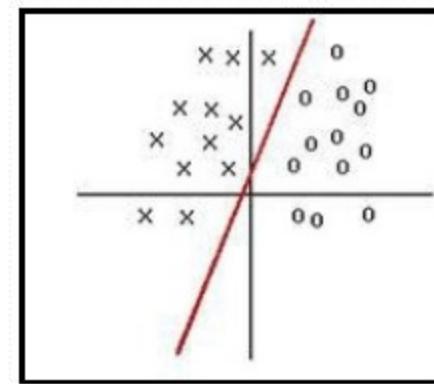
- Vec-tơ hóa công thức:

$$o(\vec{x}) = sng(\vec{w} \cdot \vec{x}) = \begin{cases} 1 & \text{nếu } \vec{w} \cdot \vec{x} > 0 \\ -1 & \text{nếu ngược lại} \end{cases}$$



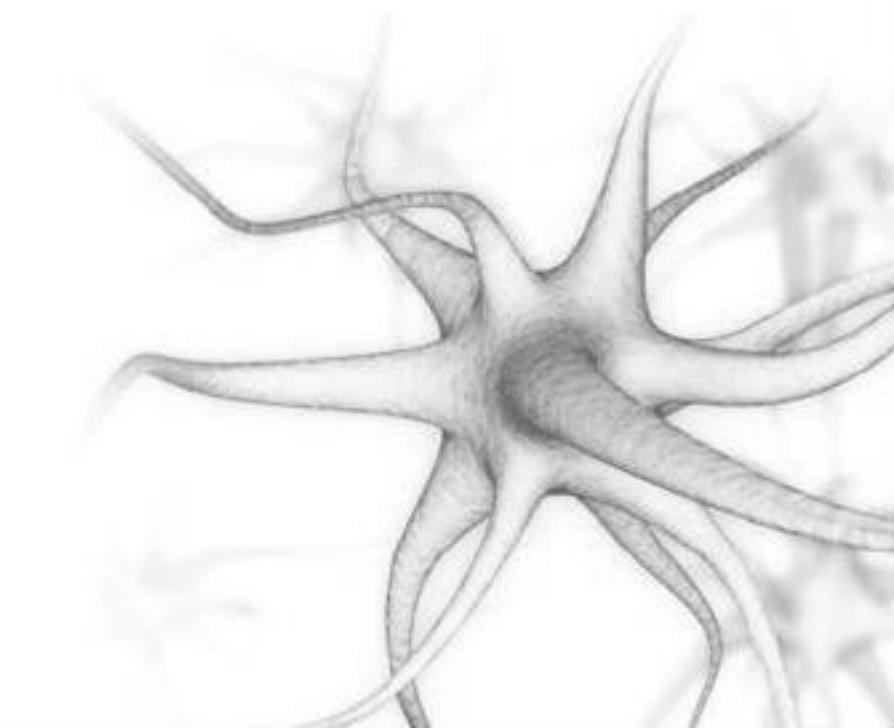
Khả năng của perceptron

- Xem như một siêu phẳng trong không gian n chiều, phân lớp tuyến tính dữ liệu.
- Có thể thể hiện cho một số hàm bool cổ sở: AND, OR, NAND, NOR,...



Học trọng số cho perceptron

- Thiết lập bộ trọng số sao cho perceptron xác định đúng tất cả các mẫu trong tập huấn luyện. Có 2 phương pháp:
 - Luật huấn luyện perceptron
 - Gradient giảm và luật huấn luyện delta

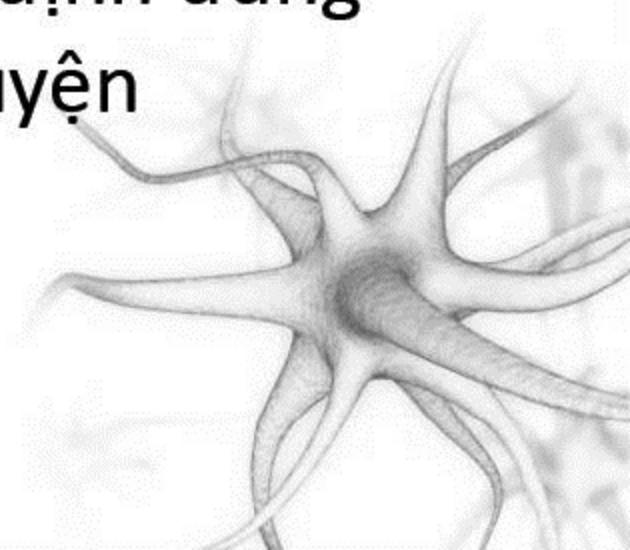


Luật huấn luyện perceptron

- Khởi tạo ngẫu nhiên bộ trọng số
- Áp dụng bộ trọng số cho toàn mẫu học
- Cập nhật trọng số cho perceptron nếu xác định sai mẫu
- Lặp cho đến khi perceptron xác định đúng tất cả các mẫu trong tập huấn luyện

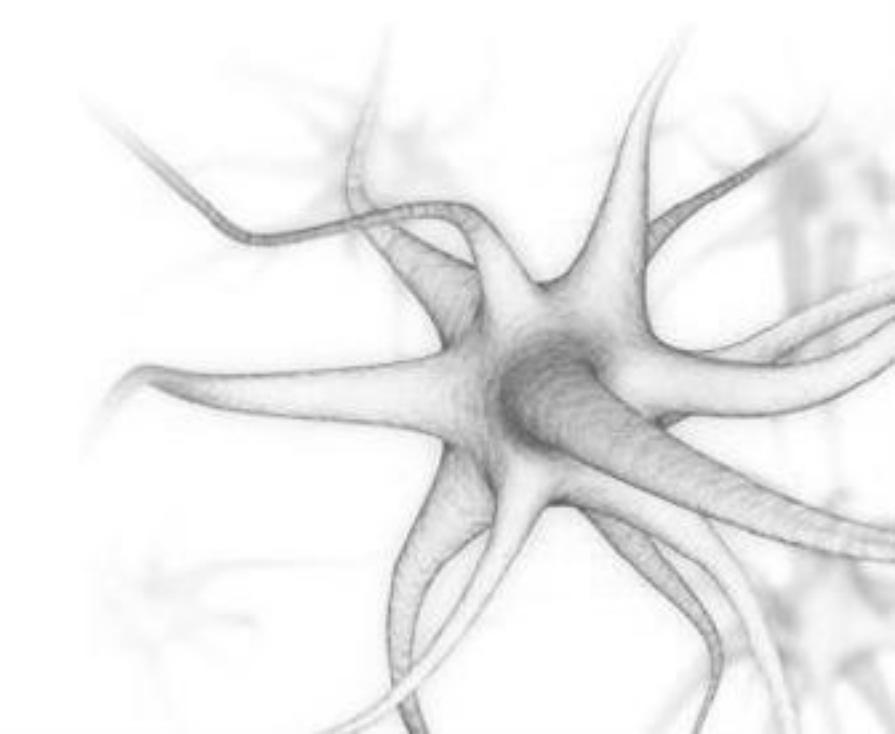
$$\mathbf{w}_i \leftarrow \mathbf{w}_i + \Delta \mathbf{w}_i$$

$$\Delta \mathbf{w}_i = \eta(t - o)\mathbf{x}_i$$



Luật huấn luyện perceptron (2)

- Điều kiện để thuật toán hội tụ là tập huấn luyện phải phân lớp tuyến tính.



Gradient giảm và luật delta

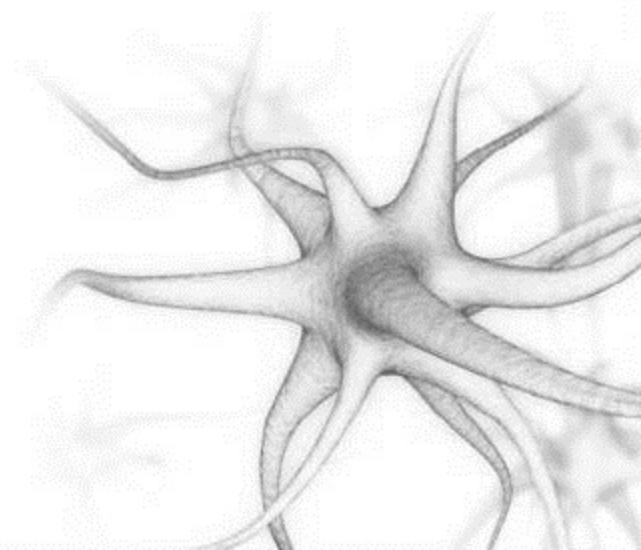
- Áp dụng tốt cho những dữ liệu không phân lớp tuyến tính
- Nếu dữ liệu không phân lớp tuyến tính, luật sẽ hội tụ đến xấp xỉ tối ưu nhất
- Sử dụng hàm lỗi và phương pháp gradient giảm



Hàm lỗi

- Thể hiện lỗi huấn luyện của một bộ trọng:

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$



Phương pháp Gradient giảm

- Ý tưởng chính của luật học delta là phương pháp gradient giảm, gồm các bước:
 - Chọn ngẫu nhiên một điểm x_0 trong không gian trọng số
 - Tính độ dốc của mặt lồi tại x_0
 - Cập nhật trọng số theo hướng dốc nhất của mặt lồi.
 - Xem điểm này như điểm x_0 mới, và quay về bước đầu tiên



Đạo hàm riêng trọng số

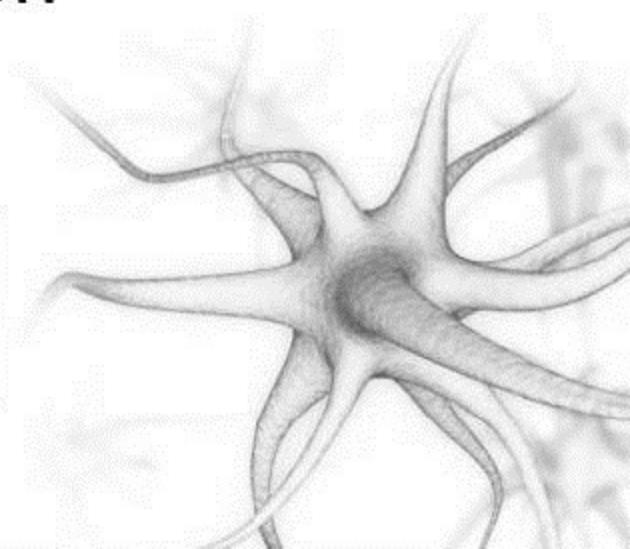
- Dùng đạo hàm riêng để tính độ dốc nhất trên mặt lồi

$$\nabla E(\vec{w}) = \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right]$$

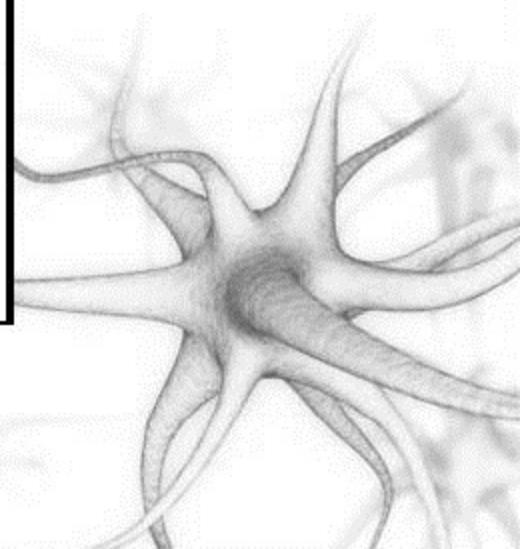
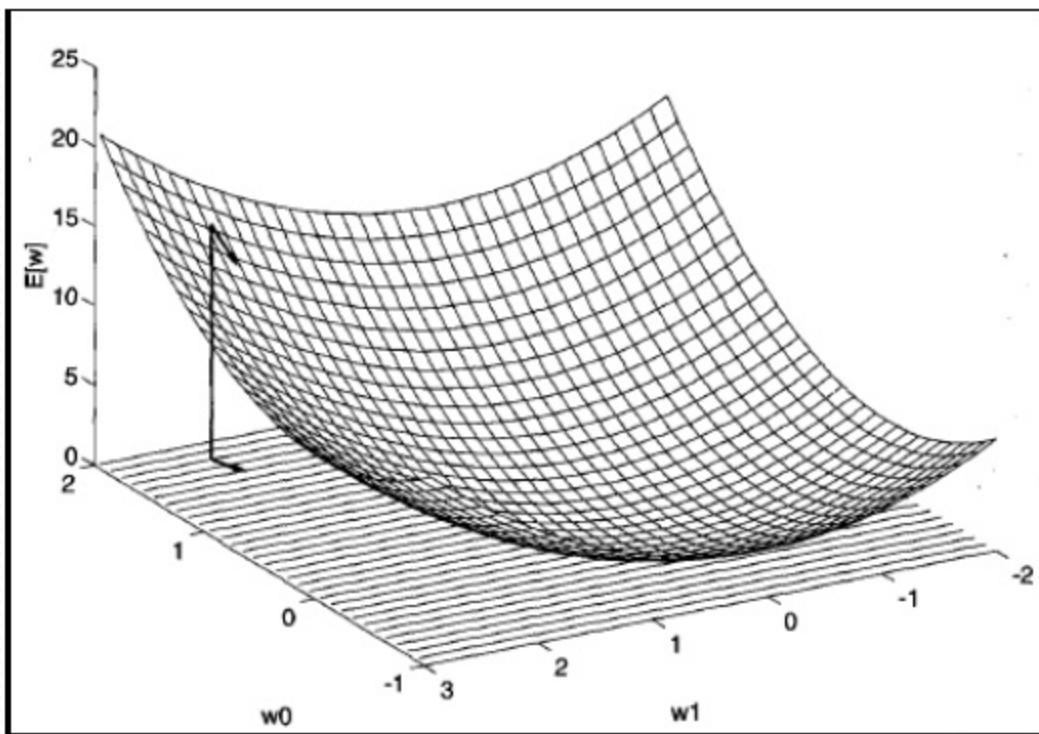
- Cập nhật trọng số cho perceptron

$$w_i \leftarrow w_i + \Delta w_i$$

$$\Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{id}$$



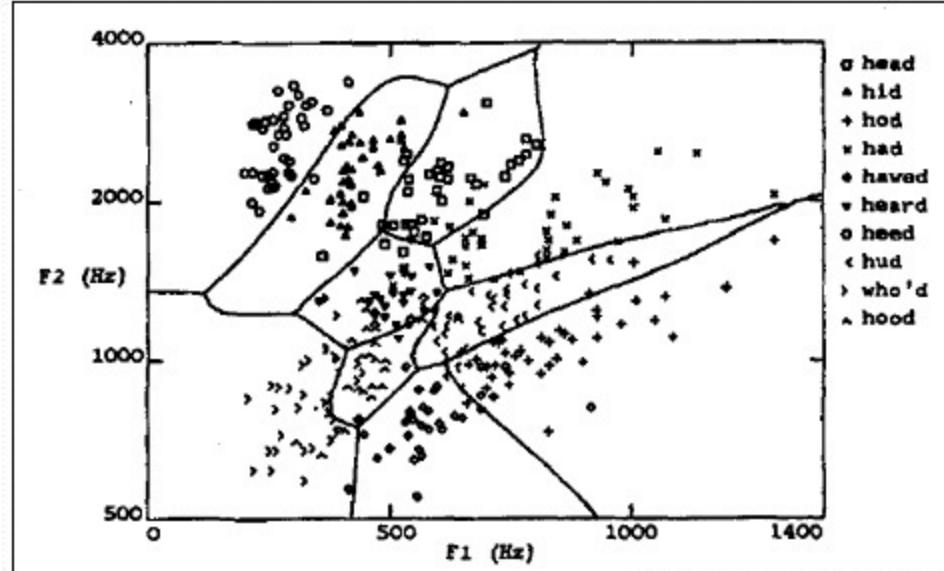
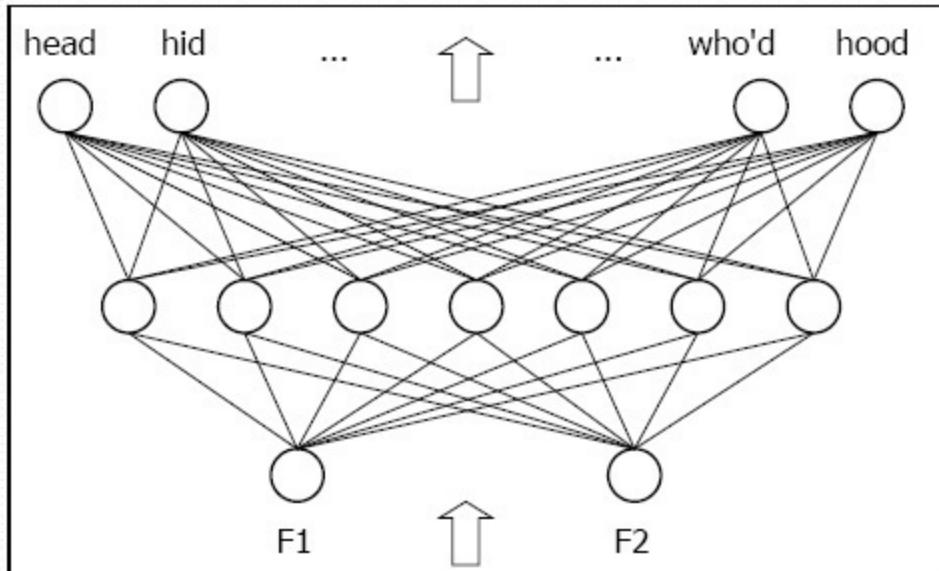
Không gian giả thuyết



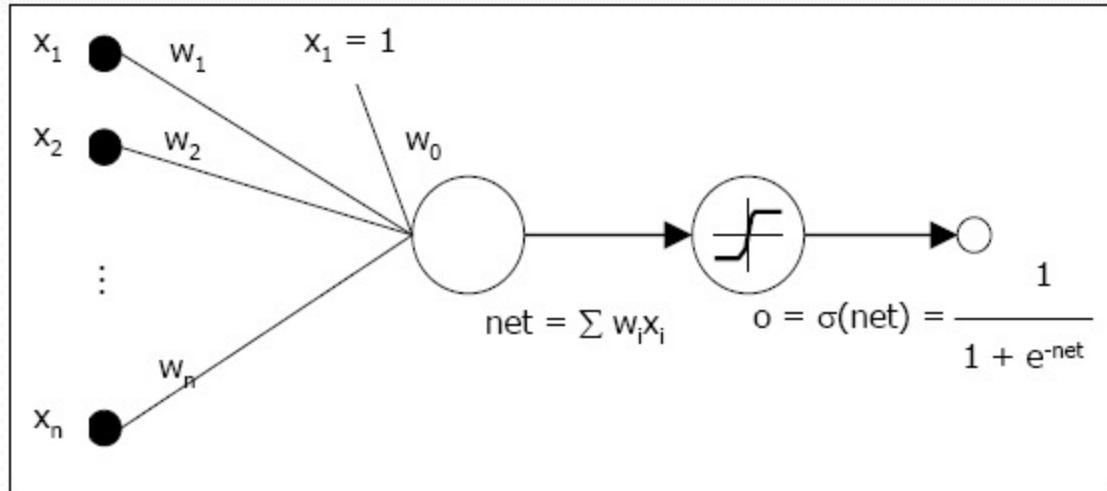
MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

MẠNG ĐA LỚP

- Perceptrons đơn chỉ có thể mô tả được các mặt quyết định tuyến tính.
- Mạng đa lớp thì dễ dàng mô tả các mặt quyết định phi tuyến.

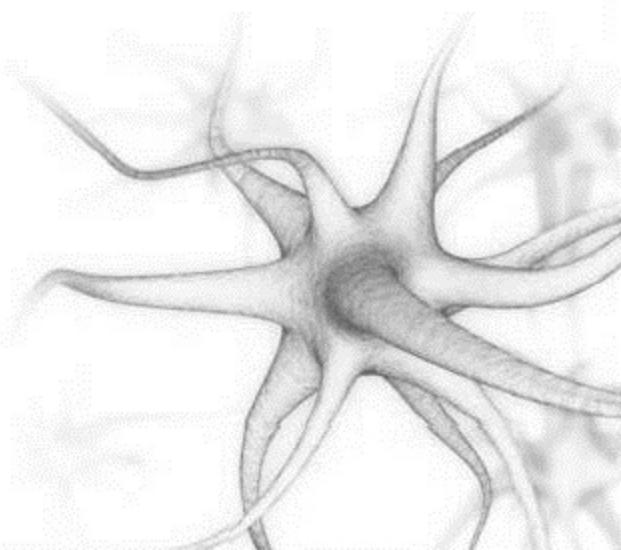


MẠNG ĐA LỚP VÀ THUẬT TOÁN TRUYỀN NGƯỢC NGƯỜNG SIGMOID



$$\sigma(y) = \frac{1}{1 + e^{-y}}$$

$$\frac{d\sigma(y)}{dy} = \sigma(y) \cdot (1 - \sigma(y))$$



MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

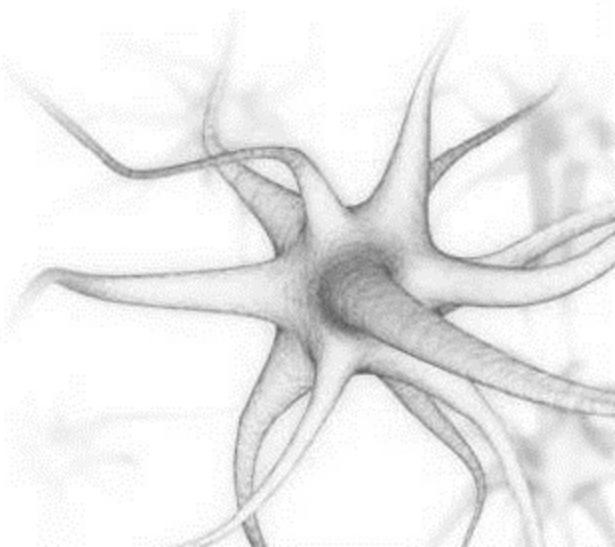
THUẬT TOÁN LAN TRUYỀN NGƯỢC

Hàm lỗi:

$$E(w) = \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{outputs}} (t_{kd} - o_{kd})^2$$

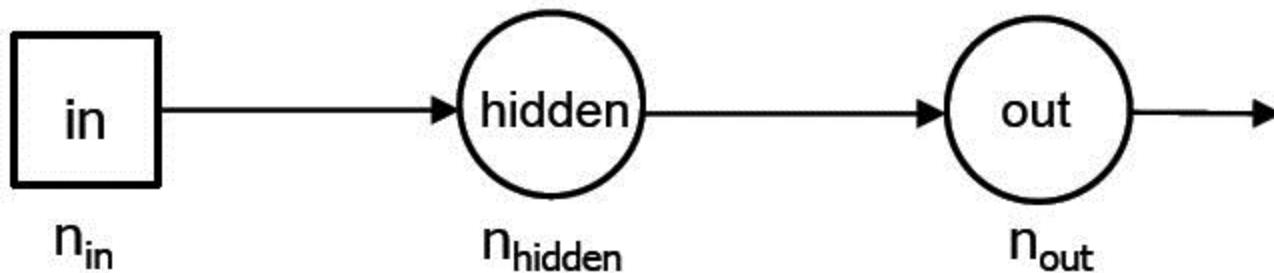
outputs: tập tất cả các nơ-ron đầu ra của mạng

t_{kd}, o_{kd} : là giá trị đích và giá trị đầu ra tương ứng của nơ-ron thứ k ứng với mẫu học d

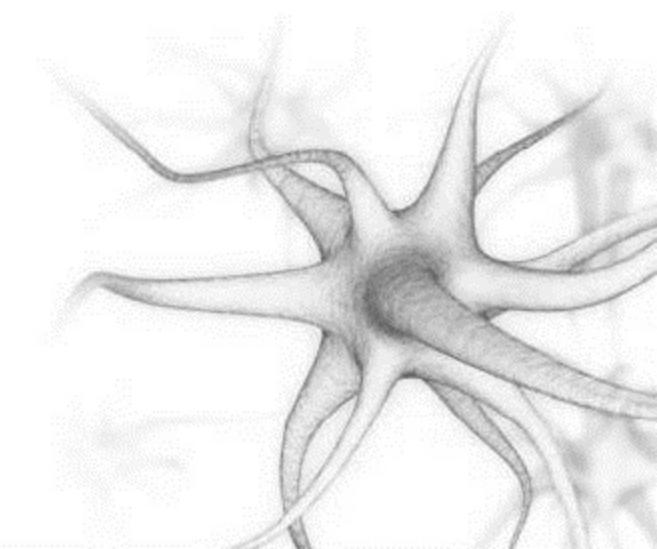


THUẬT TOÁN LAN TRUYỀN NGƯỢC

BackPropagation (training_examples, η , n_{in} , n_{out} , n_{hidden})



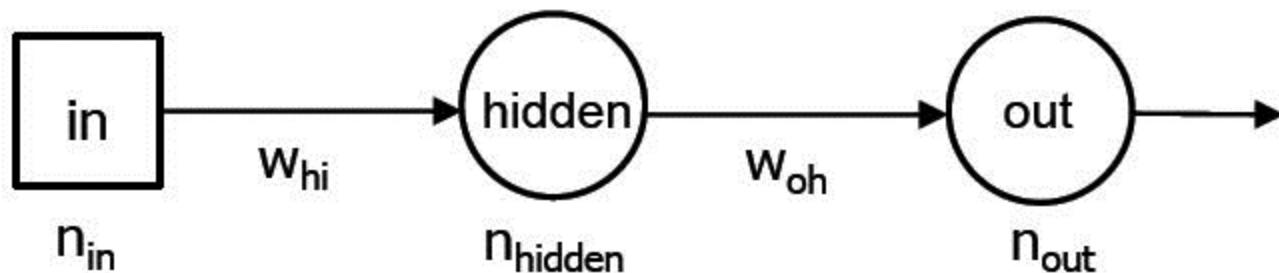
- Tạo mạng truyền thẳng gồm:
 n_{in} đầu vào,
 n_{hidden} nơ-ron tầng ẩn,
 n_{out} nơ-ron đầu ra.



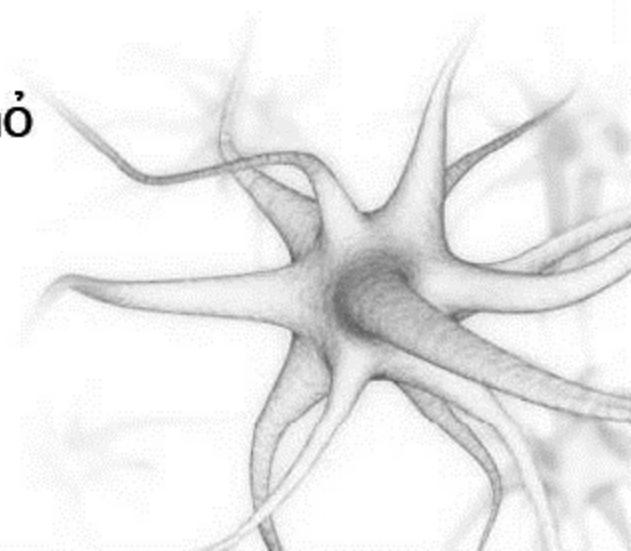
MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

THUẬT TOÁN LAN TRUYỀN NGƯỢC

BackPropagation (training_examples, η , n_{in} , n_{out} , n_{hidden})



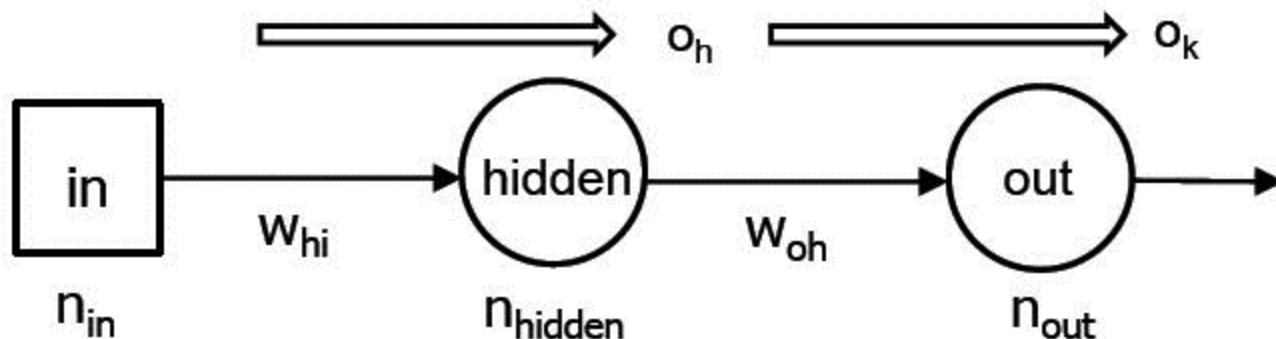
- Khởi tạo bộ trọng cho mạng với giá trị nhỏ



MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

THUẬT TOÁN LAN TRUYỀN NGƯỢC

BackPropagation (training_examples, η , n_{in} , n_{out} , n_{hidden})

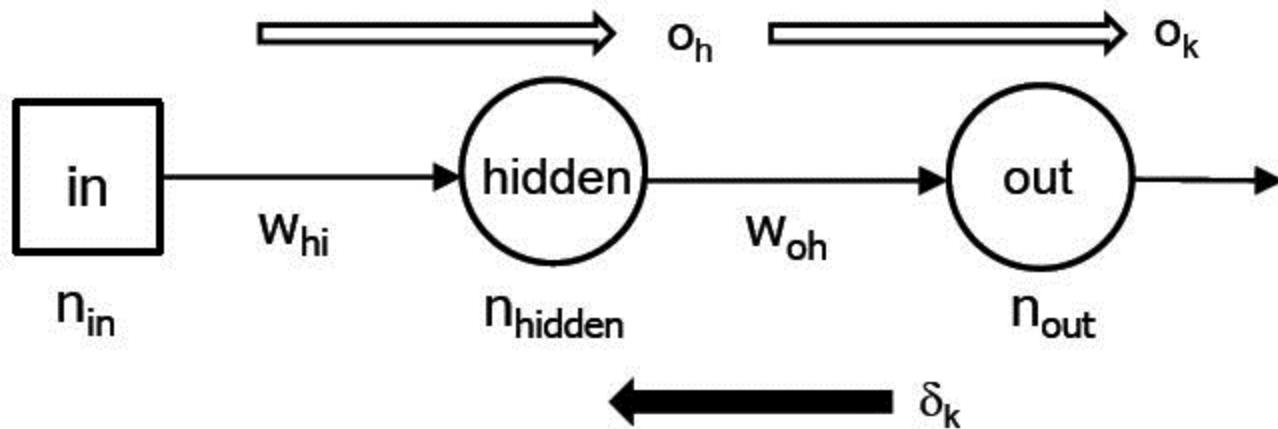


- Trong khi **<điều kiện kết thúc chưa thỏa>** làm:
Với mỗi cặp (x, t) trong không gian mẫu huấn luyện thực hiện:
 1. Đưa giá trị đầu vào \vec{x} qua mạng và tính đầu ra o_u ứng với mỗi nơ-ron u trong mạng.

MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

THUẬT TOÁN LAN TRUYỀN NGƯỢC

BackPropagation (training_examples, η , n_{in} , n_{out} , n_{hidden})

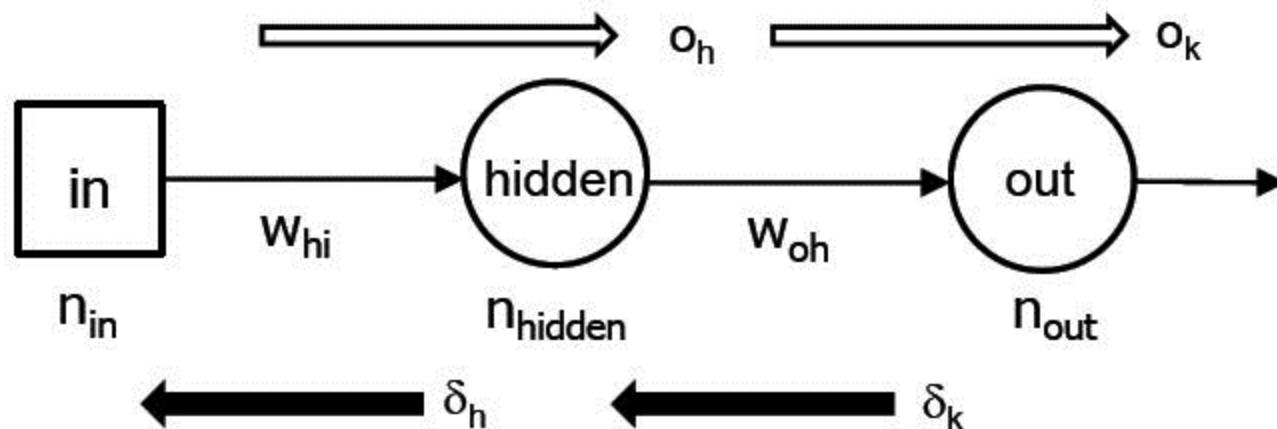


- Trong khi **<điều kiện kết thúc chưa thỏa>** làm:
Với mỗi cặp (x, t) trong không gian mẫu huấn luyện thực hiện:
 2. Với mỗi nơ-ron đầu ra k , ta tính giá trị lỗi δ_k
$$\delta_k = o_k(1 - o_k)(t_k - o_k)$$

MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

THUẬT TOÁN LAN TRUYỀN NGƯỢC

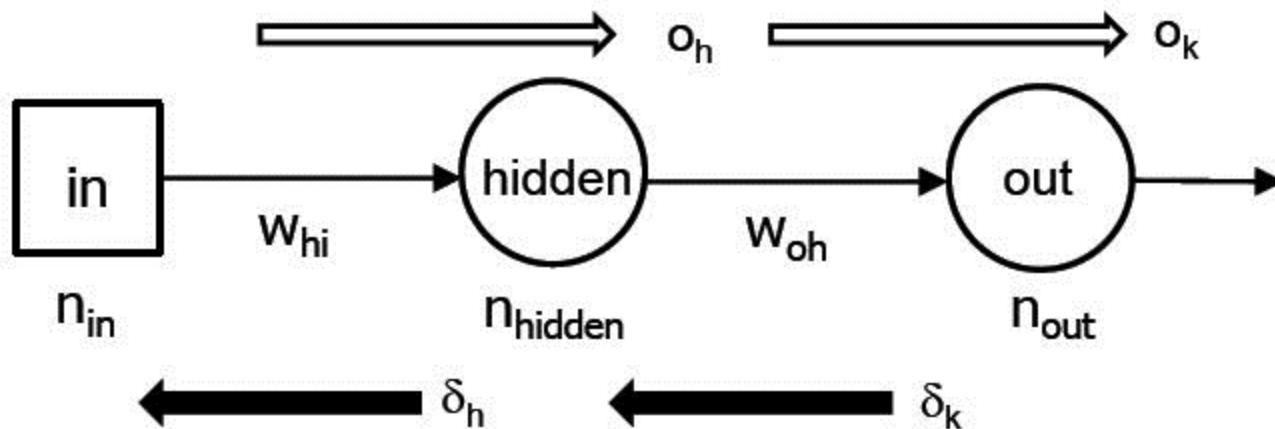
BackPropagation (training_examples, η , n_{in} , n_{out} , n_{hidden})



- Trong khi **<điều kiện kết thúc chưa thỏa>** làm:
Với mỗi cặp (x, t) trong không gian mẫu huấn luyện thực hiện:
 3. Với mỗi nơ-ron tầng ẩn h , ta tính giá trị lỗi δ_h
$$\delta_h = o_h(1 - o_h) \sum_{k \in \text{outputs}} w_{kh} \delta_k$$

THUẬT TOÁN LAN TRUYỀN NGƯỢC

BackPropagation (training_examples, η , n_{in} , n_{out} , n_{hidden})

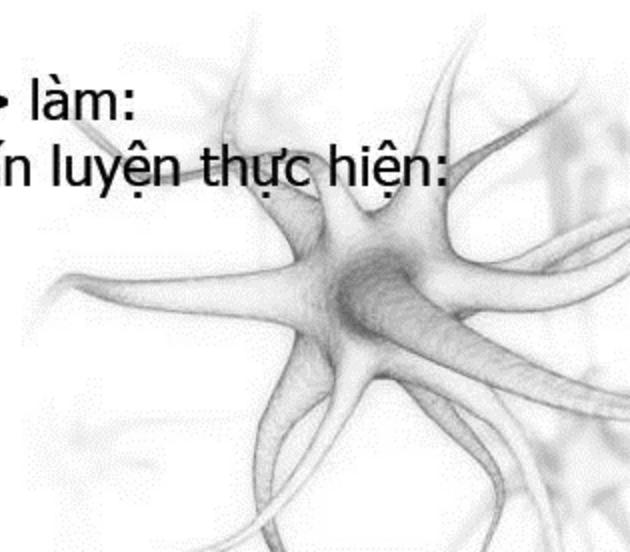


- Trong khi **<điều kiện kết thúc chưa thỏa>** làm:
Với mỗi cặp (x, t) trong không gian mẫu huấn luyện thực hiện:

4. Cập nhật lại trọng số có trong mạng w_{ji}

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

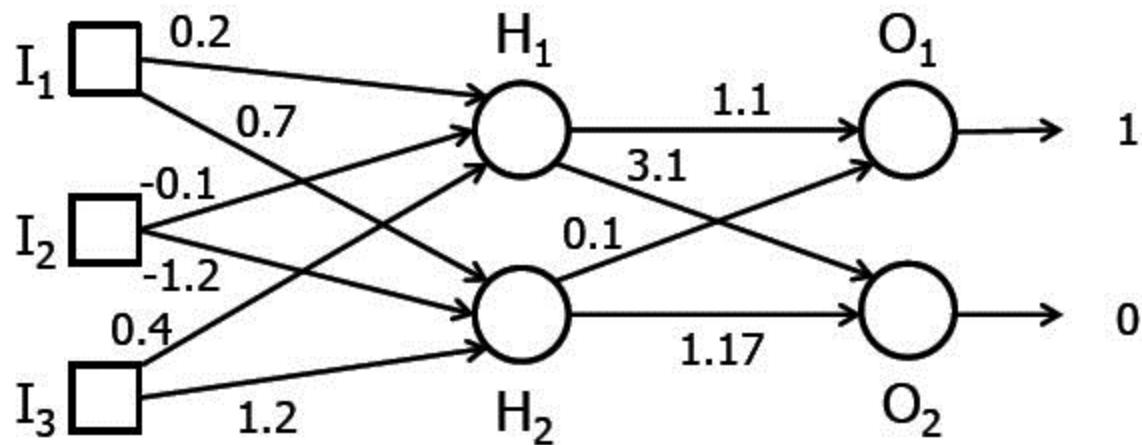
$$\Delta w_{ji} = \eta \delta_j x_{ji}$$



MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

THUẬT TOÁN LAN TRUYỀN NGƯỢC

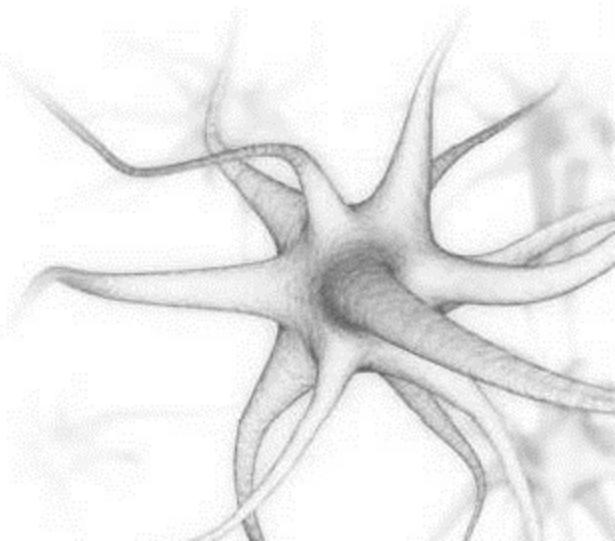
Ví dụ:



Input: $\vec{x} = (10, 30, 20)$

Target: $\vec{t} = (1, 0)$

Hệ số học: $\eta = 0.1$



THUẬT TOÁN LAN TRUYỀN NGƯỢC

Ví dụ (tt):

- Đưa giá trị đầu vào \vec{x} qua mạng và tính đầu ra o_u ứng với mỗi nơ-ron u trong mạng.

1

$$o = \sigma(\text{net}) = \frac{\text{---}}{1 + e^{-\text{net}}} \quad \text{Với net} = \sum w_{ji}x_{ji}$$

$$H_1: \text{net}_{H1} = 10 * 0.2 + 30 * (-0.1) + 20 * 0.4 = 7$$

$$o_{H1} = \sigma(\text{net}_{H1}) = 0.9990$$

$$H_2: \text{net}_{H2} = 10 * 0.7 + 30 * (-1.2) + 20 * 1.2 = -5$$

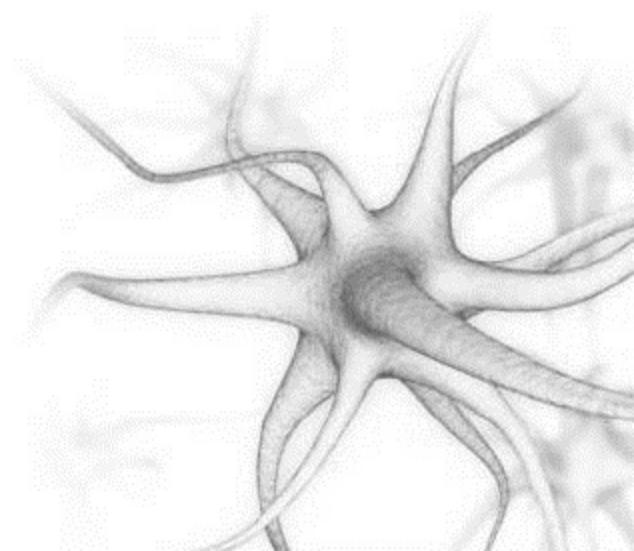
$$o_{H2} = \sigma(\text{net}_{H2}) = 0.0067$$

$$O_1: \text{net}_{O1} = 0.9990 * 1.1 + 0.0067 * 0.1 = 1.0996$$

$$o_{O1} = \sigma(\text{net}_{O1}) = 0.7501$$

$$O_2: \text{net}_{O2} = 0.9990 * 3.1 + 0.0067 * 1.17 = 3.1047$$

$$o_{O2} = \sigma(\text{net}_{O2}) = 0.9571$$



THUẬT TOÁN LAN TRUYỀN NGƯỢC

Ví dụ (tt):

2. Với mỗi nơ-ron đầu ra k, ta tính giá trị lỗi δ_k

$$\delta_k = o_k(1 - o_k)(t_k - o_k)$$

$$\delta_{O1} = o_{O1}(1 - o_{O1})(t_{O1} - o_{O1}) = 0.750(1 - 0.750)(1 - 0.750) = 0.0469$$

$$\delta_{O2} = o_{O2}(1 - o_{O2})(t_{O2} - o_{O2}) = 0.957(1 - 0.957)(0 - 0.957) = -0.0394$$

3. Với mỗi nơ-ron tầng ẩn h, ta tính giá trị lỗi δ_h

$$\delta_h = o_h(1 - o_h) \sum_{k \in \text{outputs}} w_{kh} \delta_k$$

$$\begin{aligned}\delta_{H1} &= o_{H1}(1 - o_{H1})[(w_{11} * \delta_{O1}) + (w_{21} * \delta_{O2})] \\ &= 0.999(1 - 0.999)[(1.1 * 0.0469) + (3.1 * (-0.0394))] \\ &= -0.0000705\end{aligned}$$

$$\begin{aligned}\delta_{H2} &= o_{H2}(1 - o_{H2})[(w_{12} * \delta_{O1}) + (w_{22} * \delta_{O2})] \\ &= 0.0067(1 - 0.0067)[(0.1 * 0.0469) + (1.17 * (-0.0394))] \\ &= -0.000275\end{aligned}$$

MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

THUẬT TOÁN LAN TRUYỀN NGƯỢC

Ví dụ (tt):

4. Cập nhật lại trọng số có trong mạng w_{ji}

$$w_{ji} \leftarrow w_{ji} + \Delta w_{ji}$$

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

Nơ-ron ẩn	Nơ-ron đầu ra	η	δ_o	$o_H = x_{ji}$	$\Delta = \eta \delta_o x_{ji}$	W cũ	W mới
H ₁	O ₁	0.1	0.0469	0.999	0.000469	1.1	1.100469
H ₁	O ₂	0.1	- 0.0394	0.999	-0.00394	3.1	3.09606
H ₂	O ₁	0.1	0.0469	0.0067	0.0000314	0.1	0.1000314
H ₂	O ₂	0.1	- 0.0394	0.0067	-0.0000264	1.17	1.1699736

THUẬT TOÁN LAN TRUYỀN NGƯỢC

Ví dụ (tt):

4. Cập nhật lại trọng số có trong mạng w_{ji}

Đầu vào	Nơ-ron ẩn	η	δ_H	x_I	$\Delta = \eta \delta_0 x_{ji}$	W cũ	W mới
I ₁	H ₁	0.1	-0.0000705	10	-0.0000705	0.2	0.1999295
I ₁	H ₂	0.1	-0.000275	10	-0.000275	0.7	0.699725
I ₂	H ₁	0.1	-0.0000705	30	-0.0002115	-0.1	-0.1000705
I ₂	H ₂	0.1	-0.000275	30	-0.000825	-1.2	-1.200825
I ₃	H ₁	0.1	-0.0000705	20	-0.000141	0.4	0.399859
I ₃	H ₂	0.1	-0.000275	20	-0.00055	1.2	1.19945

MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

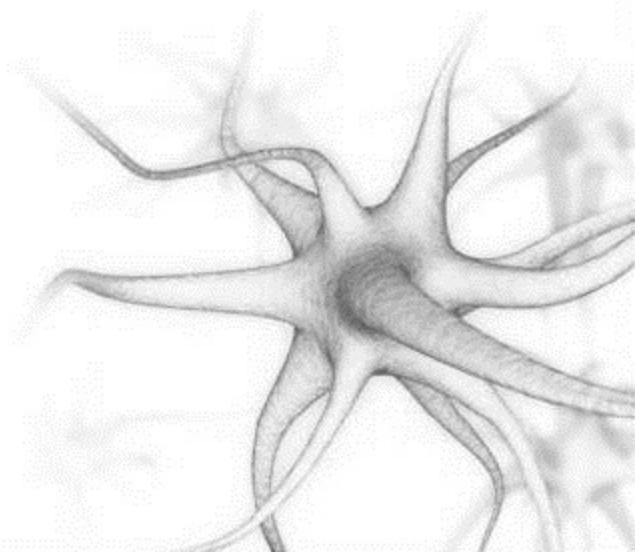
THUẬT TOÁN LAN TRUYỀN NGƯỢC

- Cập nhật trọng tăng cường

$$\Delta w_{ji}(n) = \eta \delta_i x_{ji} + \alpha \Delta w_{ji}(n-1)$$

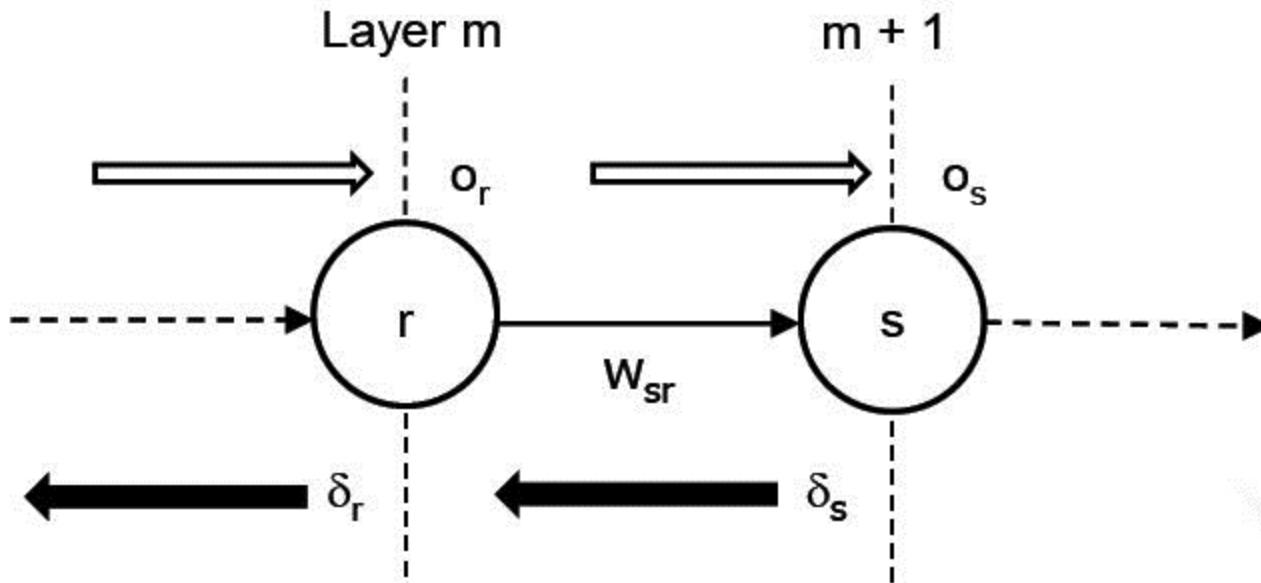
n : là lần lặp thứ n

$0 \leq \alpha < 1$: là hằng số tăng cường (momentum)

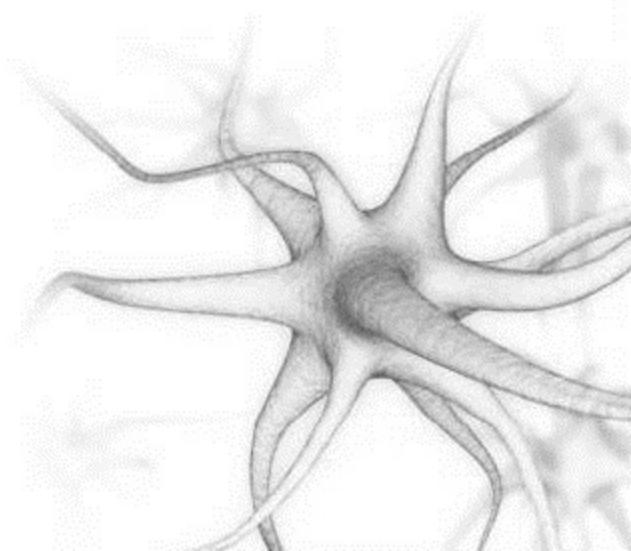


THUẬT TOÁN LAN TRUYỀN NGƯỢC

- Vấn đề học trong mạng đa lớp truyền thẳng (nhiều hơn 2 lớp)



$$\delta_r = o_r(1 - o_r) \sum_{s \in \text{layer } m+1} w_{sr} \delta_s$$



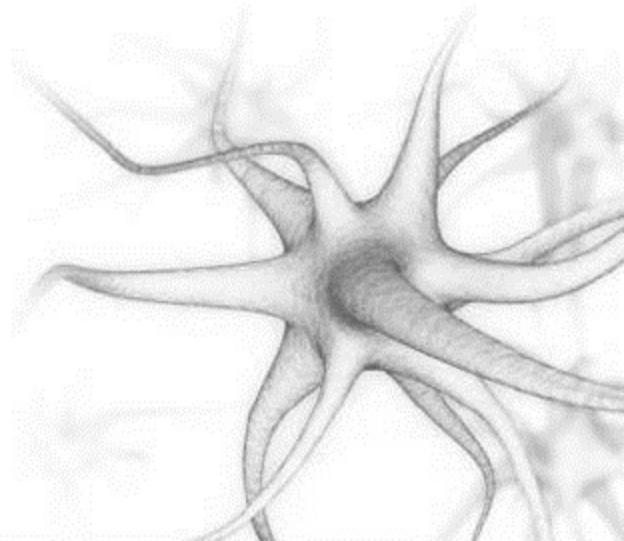
MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

CÔNG THỨC TOÁN TRONG LUẬT HỌC

- $E_d = \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2$

- $\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} \frac{\partial net_j}{\partial w_{ji}} = \frac{\partial E_d}{\partial net_j} x_{ji}$$



MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

CÔNG THỨC TOÁN TRONG LUẬT HỌC

Trường hợp 1: Luật học cho các trọng số của nơ-ron đầu ra

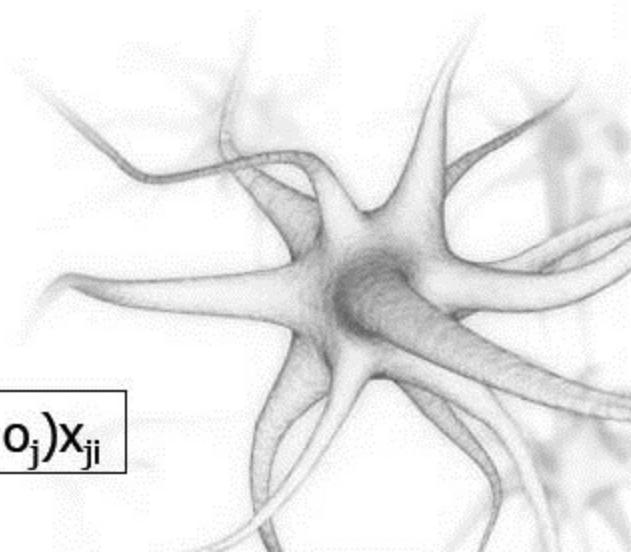
$$\frac{\partial E_d}{\partial \text{net}_j} = \frac{\partial E_d}{\partial o_j} \frac{\partial o_j}{\partial \text{net}_j}$$

$$\frac{\partial E_d}{\partial o_j} = \frac{\partial}{\partial o_j} \frac{1}{2} \sum_{k \in outputs} (t_k - o_k)^2 = \frac{\partial}{\partial o_j} \frac{1}{2} (t_j - o_j)^2 \quad (\text{Do } \frac{\partial}{\partial o_j} (t_k - o_k)^2 = 0 \text{ với } k \neq j)$$

$$= 2 * \frac{1}{2} * (t_j - o_j) \frac{\partial (t_j - o_j)}{\partial o_j} = -(t_j - o_j)$$

$$\frac{\partial o_j}{\partial \text{net}_j} = \frac{\partial \sigma(\text{net}_j)}{\partial \text{net}_j} = o_j(1 - o_j)$$

$$\frac{\partial E_d}{\partial \text{net}_j} = -(t_j - o_j) o_j(1 - o_j) \Rightarrow \boxed{\Delta w_{ji} = \eta(t_j - o_j) o_j(1 - o_j)x_{ji}}$$



MẠNG ĐA LỚP VÀ THUẬT TOÁN LAN TRUYỀN NGƯỢC

CÔNG THỨC TOÁN TRONG LUẬT HỌC

Trường hợp 2: Luật học cho các trọng số của nơ-ron tầng ẩn

$$\begin{aligned}\frac{\partial E_d}{\partial net_j} &= \sum_{k \in Downstream(j)} \frac{\partial E_d}{\partial net_k} \frac{\partial net_k}{\partial net_j} \\&= \sum_{k \in Downstream(j)} -\delta_k \frac{\partial net_k}{\partial net_j} = \sum_{k \in Downstream(j)} -\delta_k \frac{\partial net_k}{\partial o_j} \frac{\partial o_j}{\partial net_j} \\&= \sum_{k \in Downstream(j)} -\delta_k w_{kj} \frac{\partial o_j}{\partial net_j} = \sum_{k \in Downstream(j)} -\delta_k w_{kj} o_j(1 - o_j)\end{aligned}$$

Đặt $\delta_j = -\frac{\partial E_d}{\partial net_j} \Rightarrow \delta_j = o_j(1 - o_j) \sum_{k \in Downstream(j)} -\delta_k w_{kj}$

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

