

# Một số mạng nơ ron nhân tạo thông dụng

---

TS. NGUYỄN ĐÌNH HÓA

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

# Nội dung

---

Một số cấu trúc mạng nơ ron cơ bản

Mạng RBF

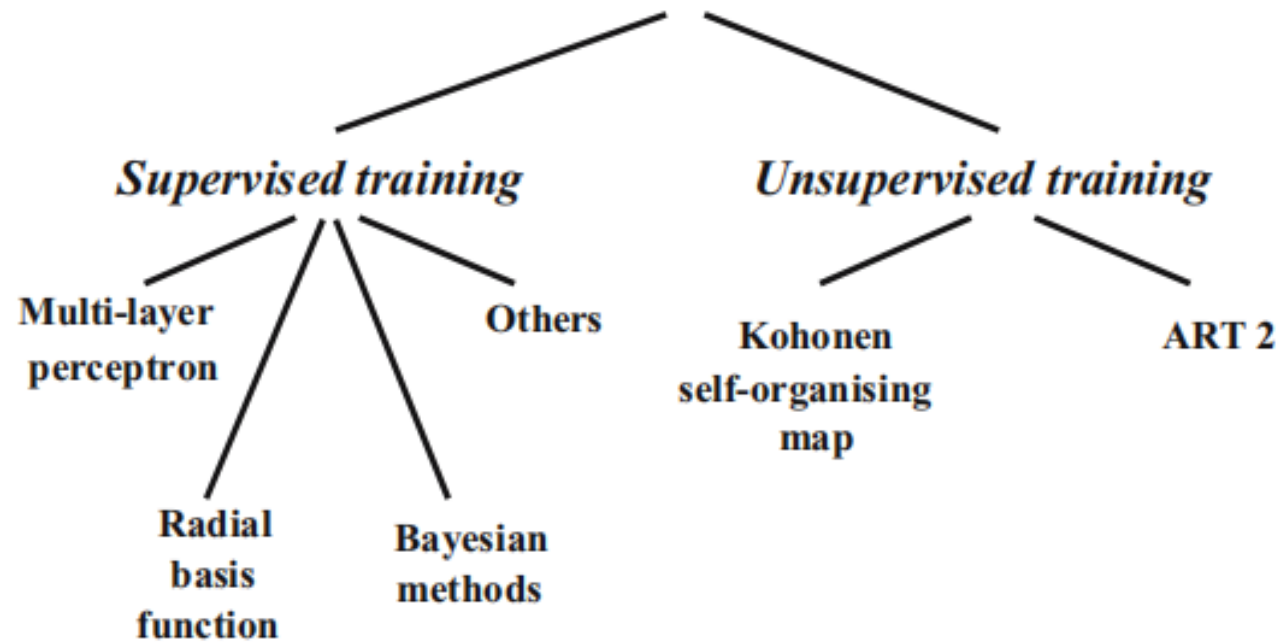
Mạng Self-Organizing Maps (SOM)

Mạng nơ ron có phản hồi

Mạng nơ ron tích chập

# Một số cấu trúc mạng nơ ron cơ bản

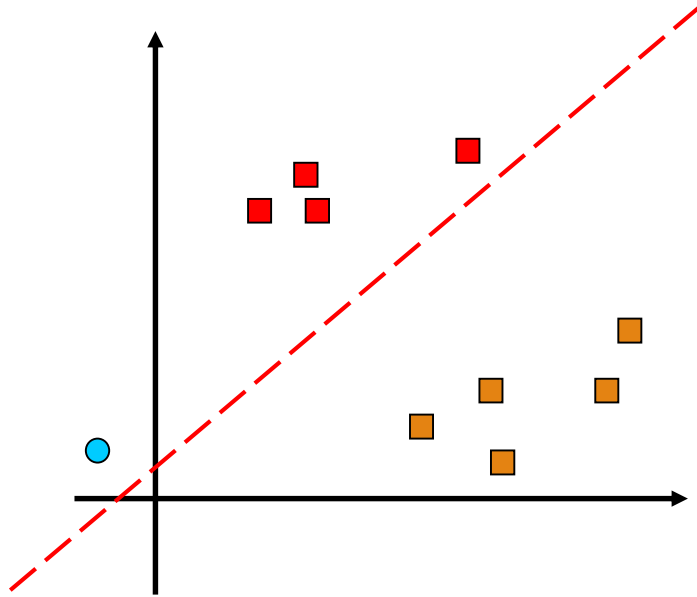
---



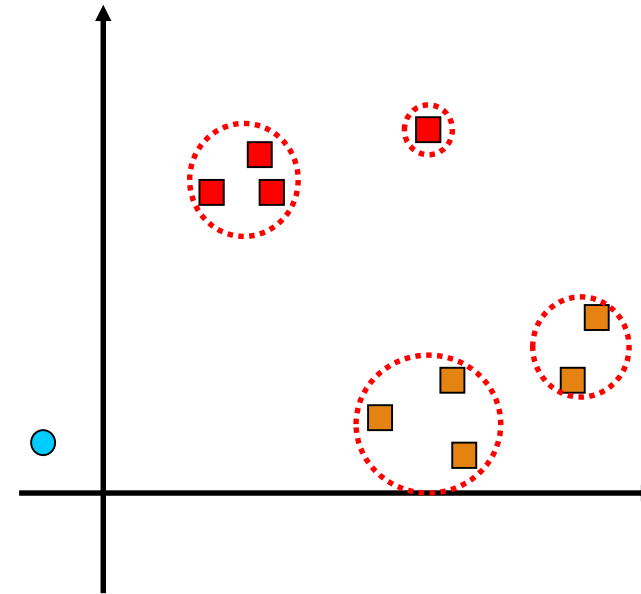
# Radial Basic Function (RBF)

---

MLP:



RBF:



# RBF

---

RBF là một mạng nơ ron có giám sát

Huấn luyện mạng RBF:

- Là quá trình tìm một mặt cong phi tuyến trong không gian đa chiều để khớp nhất với dữ liệu huấn luyện

Tổng quát hóa:

- Sử dụng mặt cong đa chiều này để nội suy ra các điểm dữ liệu mới trong tập kiểm tra

Hàm truyền của lớp đầu ra là hàm tuyến tính, kết hợp các kết quả của các hàm RBF trước đó.

$$F(x) = \sum w_i h(x)$$

Hàm truyền của lớp ẩn  $h(x)$  thường được chọn là hàm Gaussian.

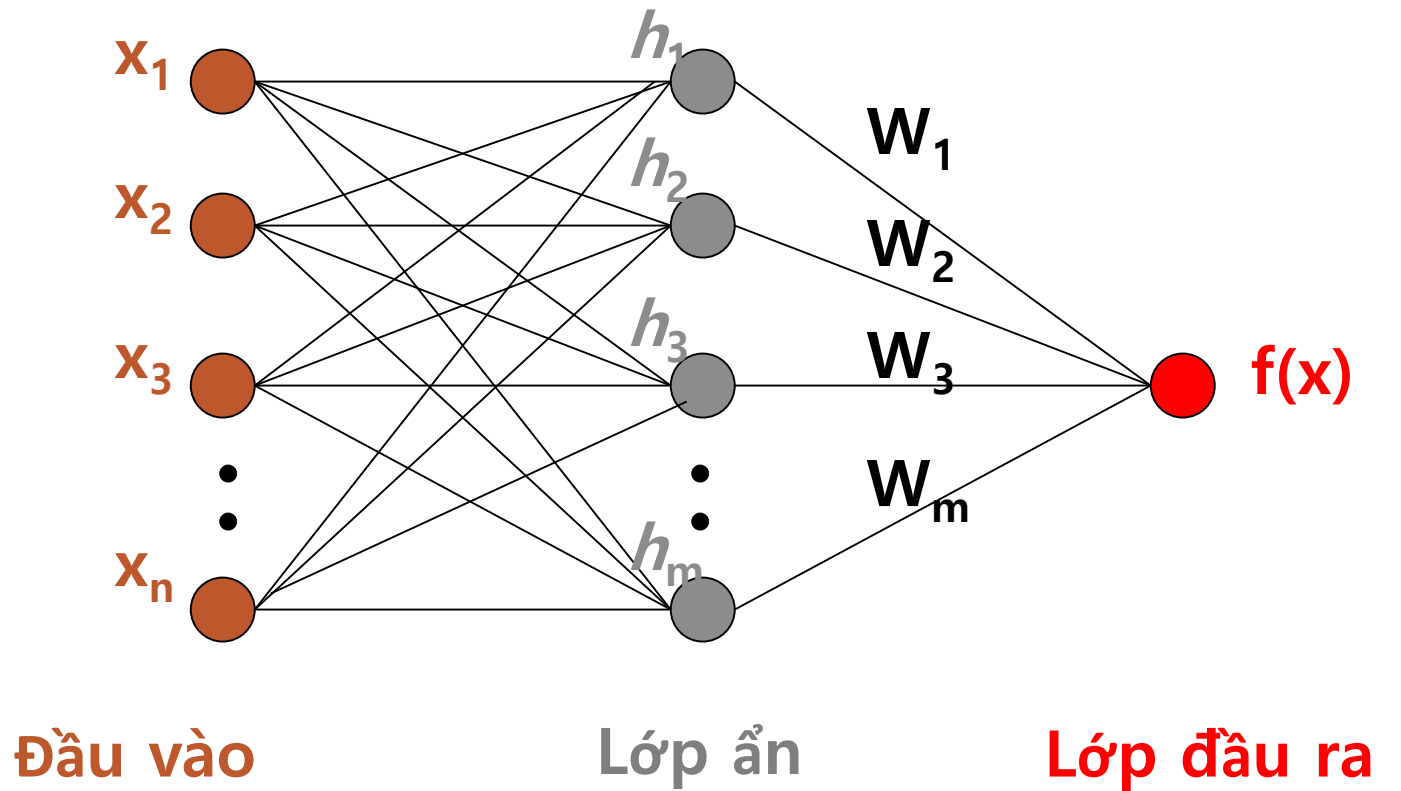
# Cấu trúc mạng RBF

Mạng RBF có số lượng rất lớn các nơ ron ở lớp ẩn.

Hàm rbf của lớp ẩn thường là hàm Gaussian:

$$h(x) = e^{-\frac{(x-c)^2}{r^2}}$$

Trong đó  $c$  là tâm của vùng dữ liệu;  
 $r$  là độ rộng của vùng dữ liệu đầu vào



# Huấn luyện RBF

---

Hai tầng huấn luyện:

- Tìm tâm và độ rộng dữ liệu
- Tìm trọng số của lớp đầu ra

Các chiến lược huấn luyện:

- Chọn tâm ngẫu nhiên rồi cố định chúng, chỉ cần huấn luyện các trọng số lớp đầu ra
- Chọn tâm dựa trên mạng tự tổ chức, sau đó huấn luyện các trọng số lớp đầu ra
- Chọn tâm có giám sát: tất cả các tham số của mạng RBF đều được huấn luyện dựa trên dữ liệu.

# Self-Organizing Maps (SOM)

---

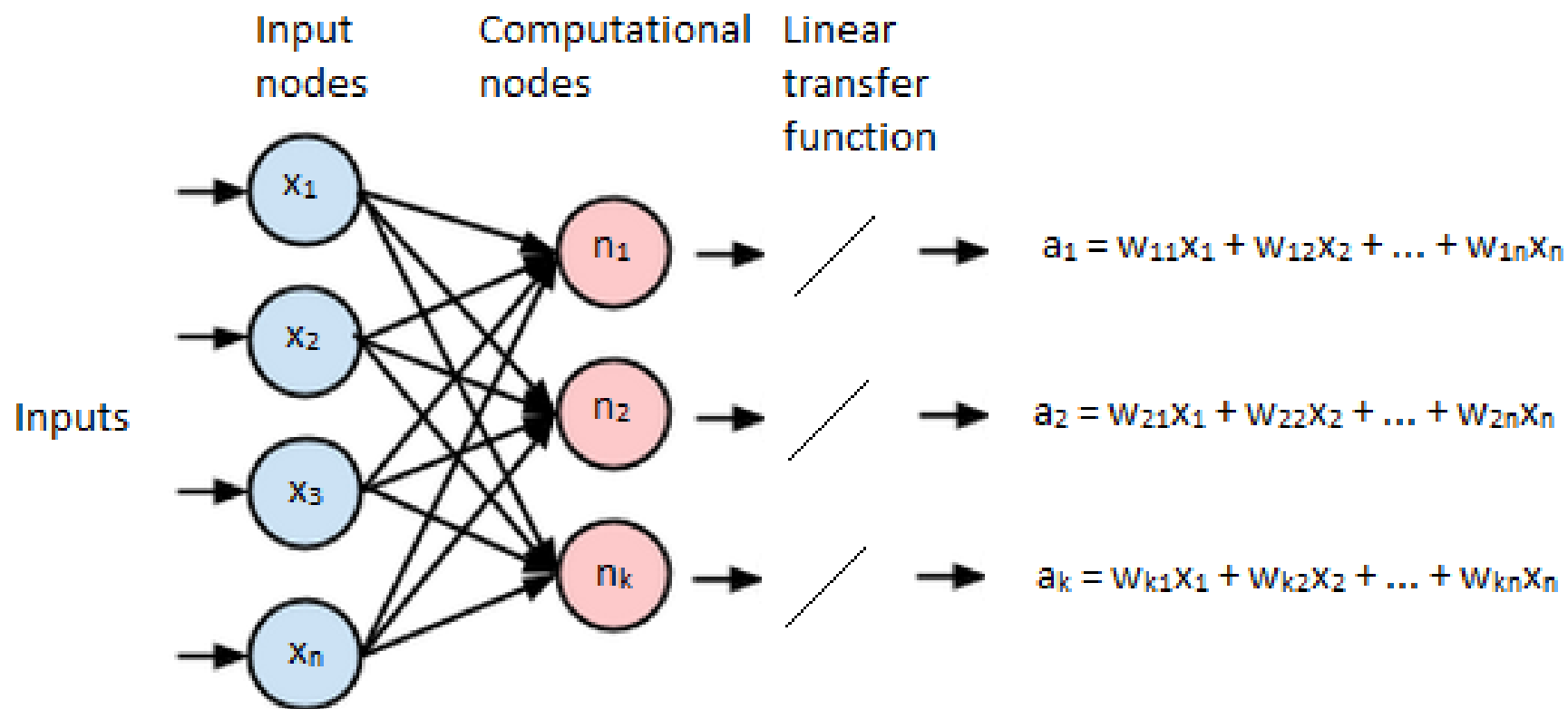
SOMs là mạng nơ ron cho phép biểu diễn dữ liệu đa chiều trong không gian ít chiều hơn, thông thường là không gian 2 chiều.

“Self-organizing” có nghĩa là không cần giám sát

SOMs làm việc dựa trên quá trình huấn luyện không giám sát



# Cấu trúc SOMs



# Cấu trúc SOMs

---

SOMs là mạng nơ ron có cấu trúc gồm 2 lớp:

- Lớp đầu vào
- Lớp tính toán

Mỗi nơ ron ở lớp đầu vào kết nối đầy đủ với tất cả các nơ ron ở lớp tính toán

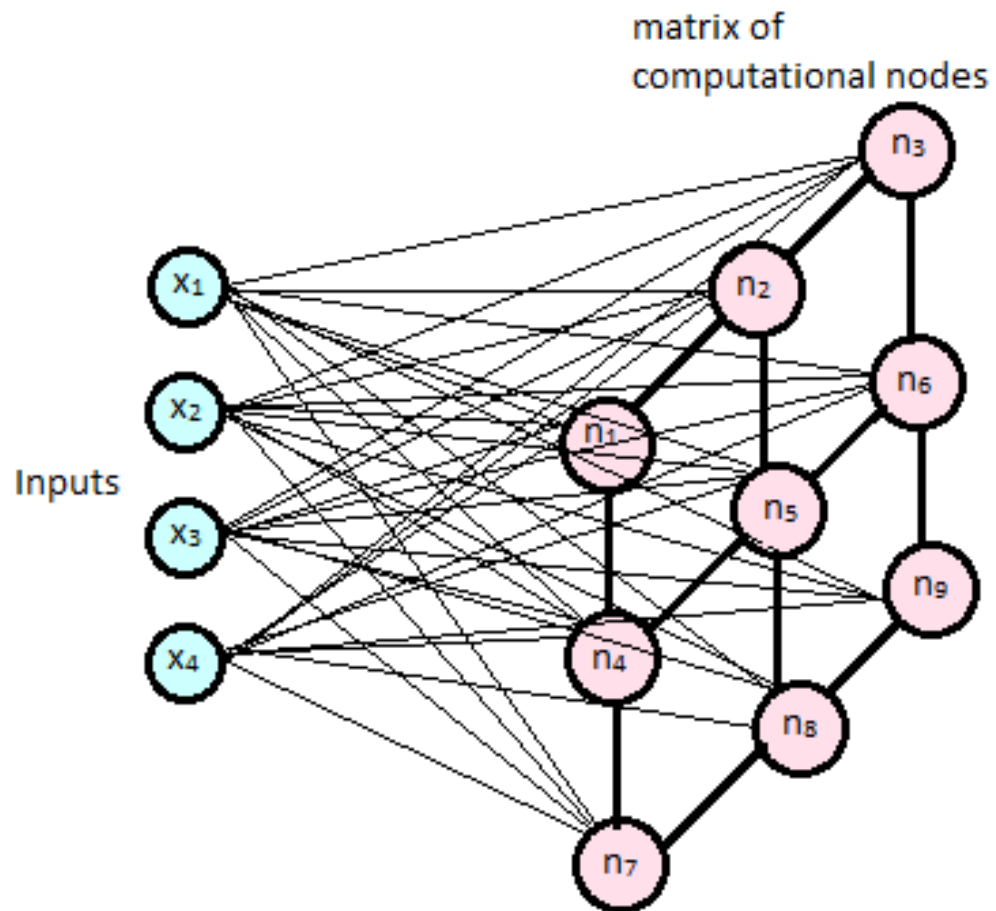
Các nơ ron ở lớp tính toán không liên kết với nhau

Mỗi nơ ron lớp tính toán gồm một vector trọng số có độ dài bằng số thuộc tính của dữ liệu đầu vào.

Các nơ ron lớp tính toán được sắp xếp theo mặt phẳng 2 chiều, hoặc đường thẳng 1 chiều, biểu diễn dữ liệu ánh xạ ở dạng 2 chiều hoặc 1 chiều

# Cấu trúc SOMs

---



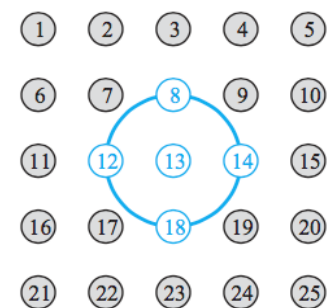
# Thuật toán huấn luyện SOM

1. Khởi tạo:
  - Tất cả các trọng số của các nơ ron được khởi tạo ngẫu nhiên.
  - Thiết lập quy tắc xác định láng giềng của các nơ ron, chọn tốc độ cập nhật trọng số
2. Chọn ngẫu nhiên một mẫu dữ liệu  $i$  từ bộ dữ liệu huấn luyện
3. Tính khoảng cách từ từng nơ ron đến mẫu dữ liệu đã chọn, tìm nơ ron  $j$  có khoảng cách ngắn nhất với mẫu dữ liệu, gọi là nơ ron chiến thắng:  $Y_j = \arg \min_j (\sum \|X_i - W_{ij}\|)$
4. Cập nhật lại trọng số của nơ ron  $j$  cùng trọng số của các nơ ron láng giềng của nó.

$$W_{ij}(t+1) = W_{ij}(t) + \alpha h_{ij}(X(t) - W_{ij}(t))$$

5. Cập nhật lại tốc độ cập nhật trọng số  $\alpha$ :  $\alpha(t) = \frac{\alpha_0}{1 + \text{decay\_rate} * t}$
6. Giảm diện tích vùng láng giềng của mỗi nơ ron
7. Lặp lại từ bước 2 đến khi điều kiện dừng của thuật toán được thỏa mãn

Vùng láng giềng

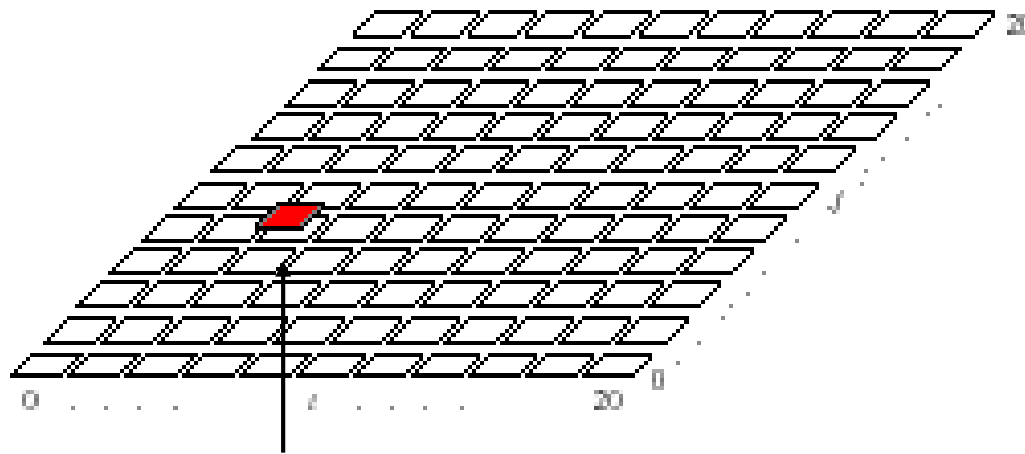


# Huấn luyện SOM

---

Tìm nơ ron có trọng số gần với mẫu dữ liệu đầu vào nhất

$$Y_j = \arg \min_j (\sum \| X_i - W_{ij} \|)$$

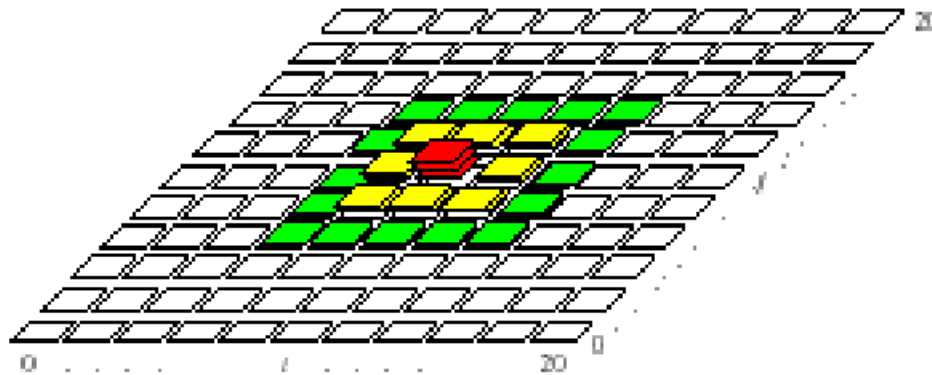


Nơ ron chiến thắng

# Cập nhật trọng số của nơ ron chiến thắng

---

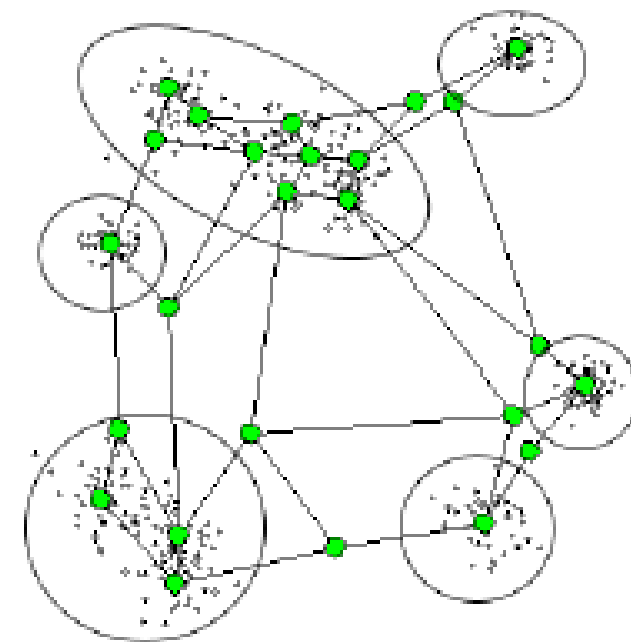
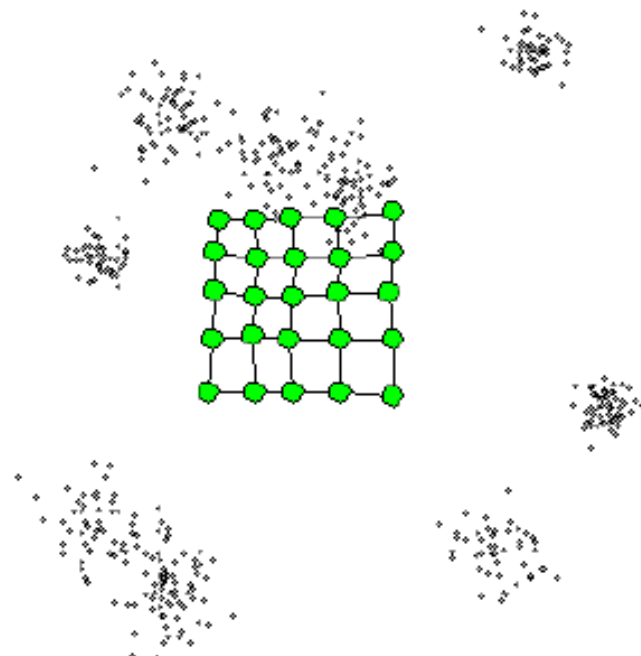
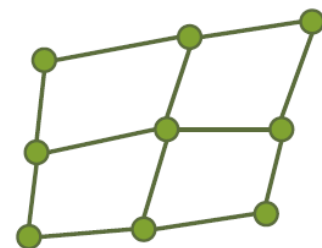
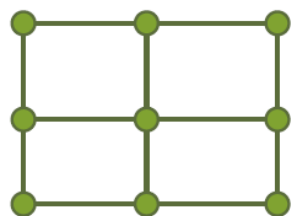
$$W_{ij}(t+1) = W_{ij}(t) + \alpha h_{ij}(X(t) - W_{ij}(t))$$



$$h_{ij} = \exp(-d_{ij}^2 / 2\sigma^2)$$

# Huấn luyện SOM

---



# Mạng nơ ron phản hồi (RNN)

---

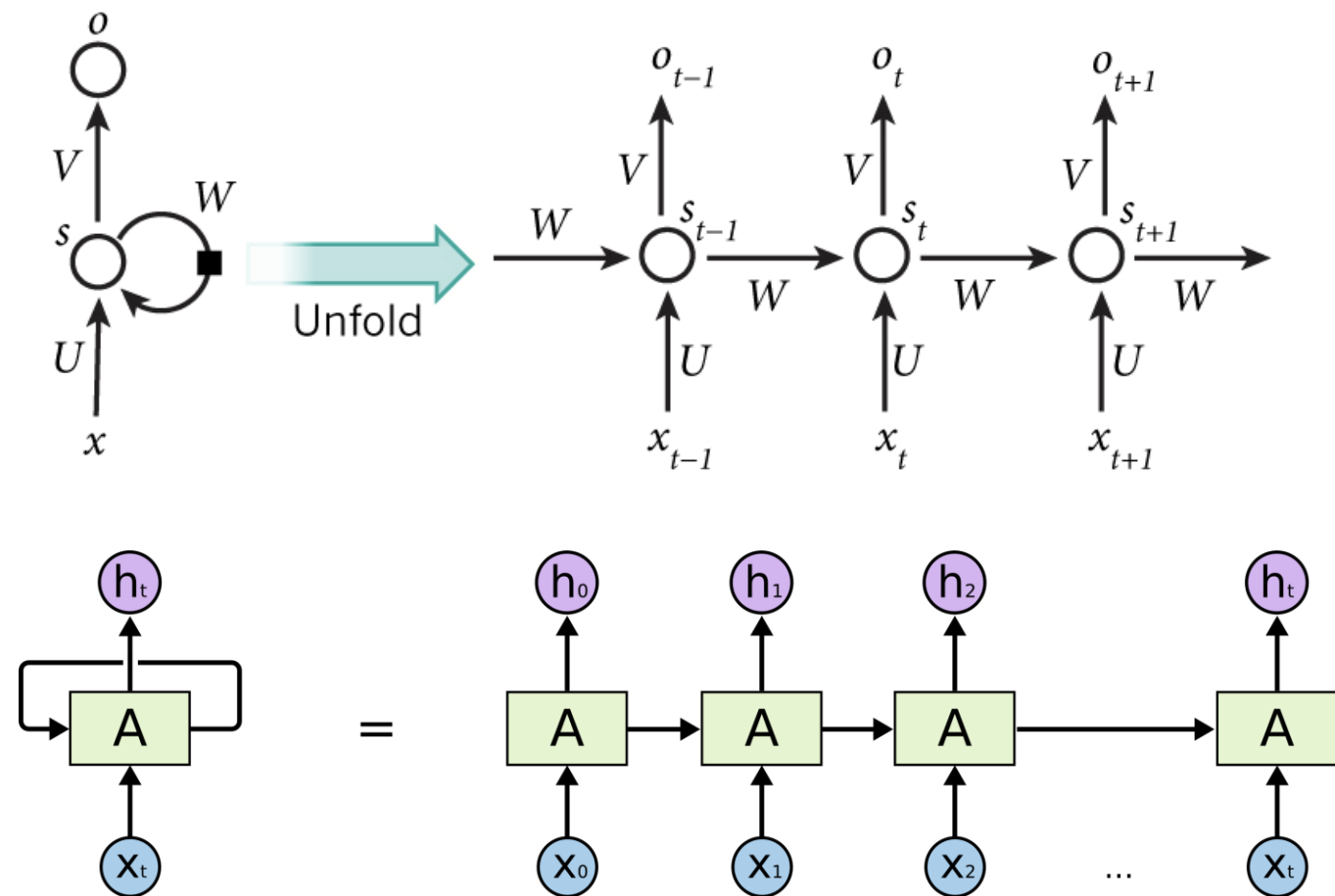
RNN là mạng nơ ron có đầu ra quay ngược trở lại làm một thông tin cho đầu vào

RNN cho phép mạng nơ ron có thể học được các thông tin về mối quan hệ thời gian giữa các mẫu dữ liệu (thông tin về chuỗi dữ liệu)

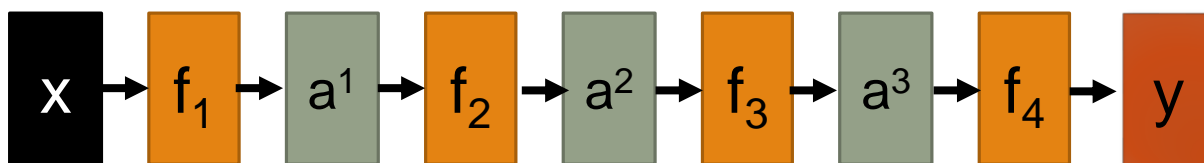
RNN khó huấn luyện hơn các mạng nơ ron thông thường do tốn tài nguyên tính toán hơn



# Cấu trúc RNN



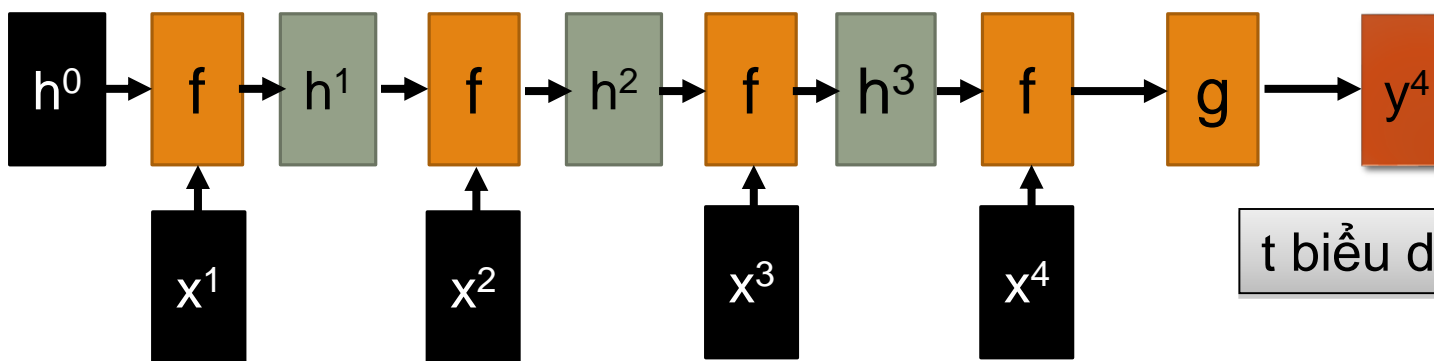
# Mạng truyền thẳng Vs Mạng phản hồi



Mạng truyền thẳng có các tham số khác nhau cho các lớp khác nhau

$$a^t = f_t(a^{t-1}) = \sigma(W^t a^{t-1} + b^t)$$

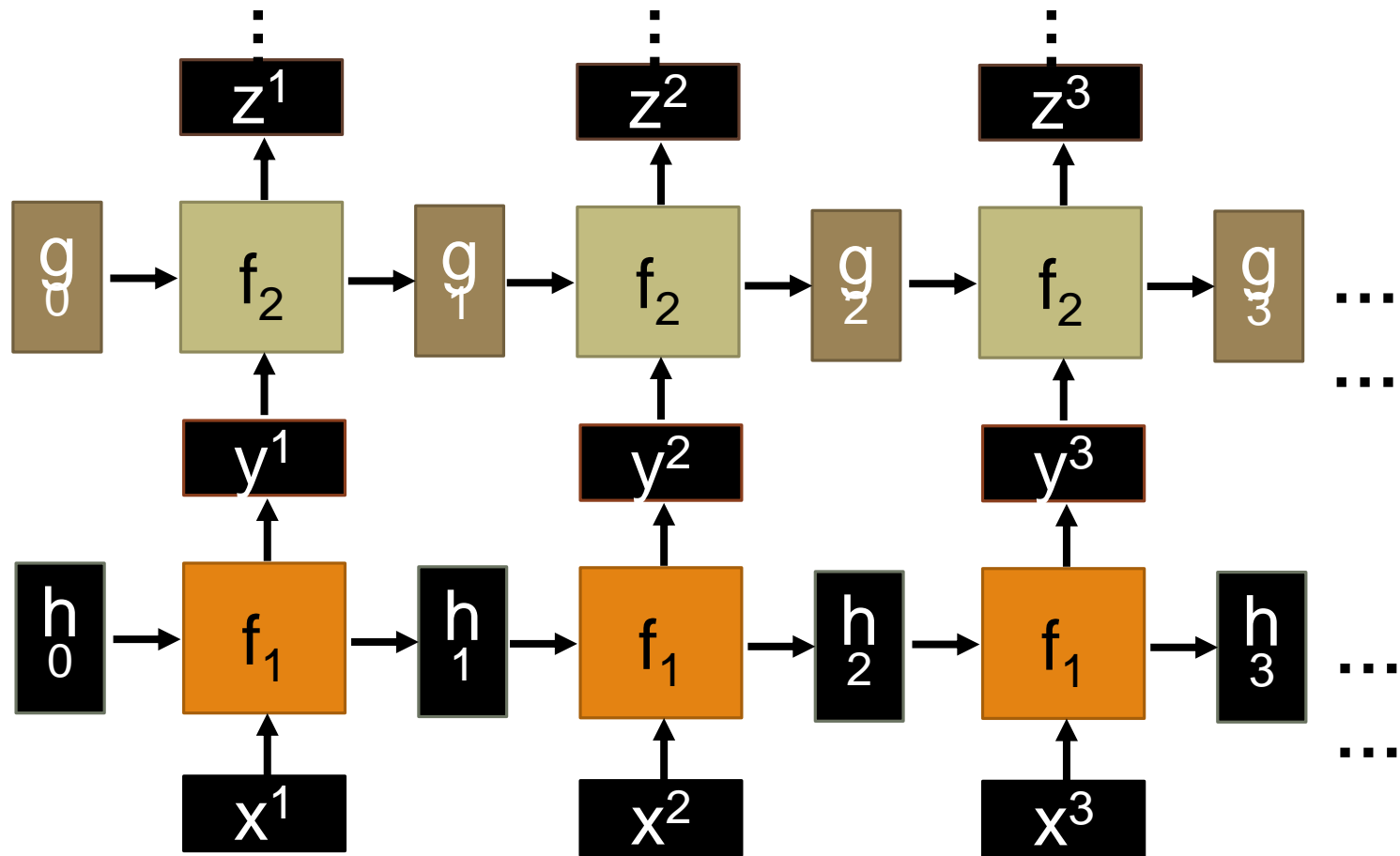
$t$  biểu diễn số lớp



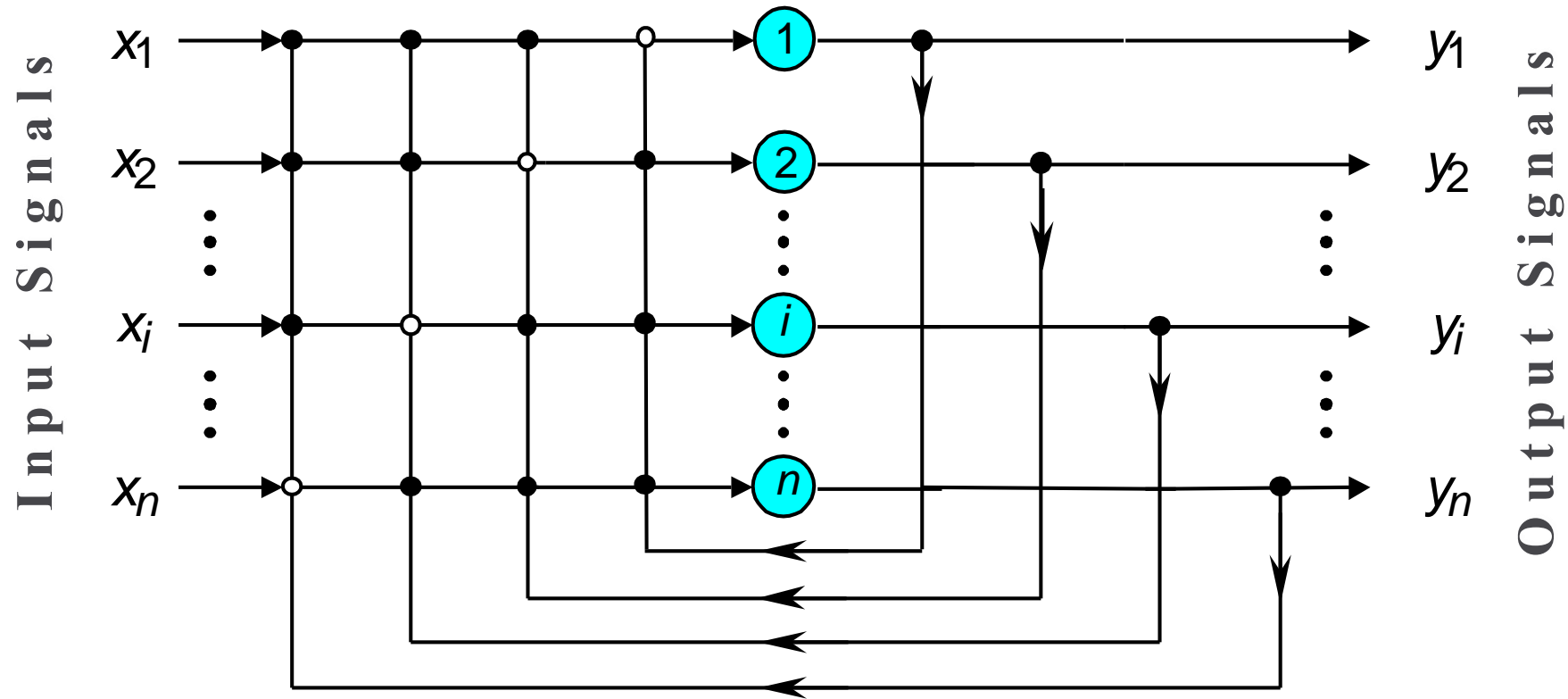
$t$  biểu diễn bước thời gian

$$a^t = f(a^{t-1}, x^t) = \sigma(W^h a^{t-1} + W^i x^t + b^i)$$

# Deep RNN (mạng phản hồi sâu)



# RNN – mạng Hopfield một lớp



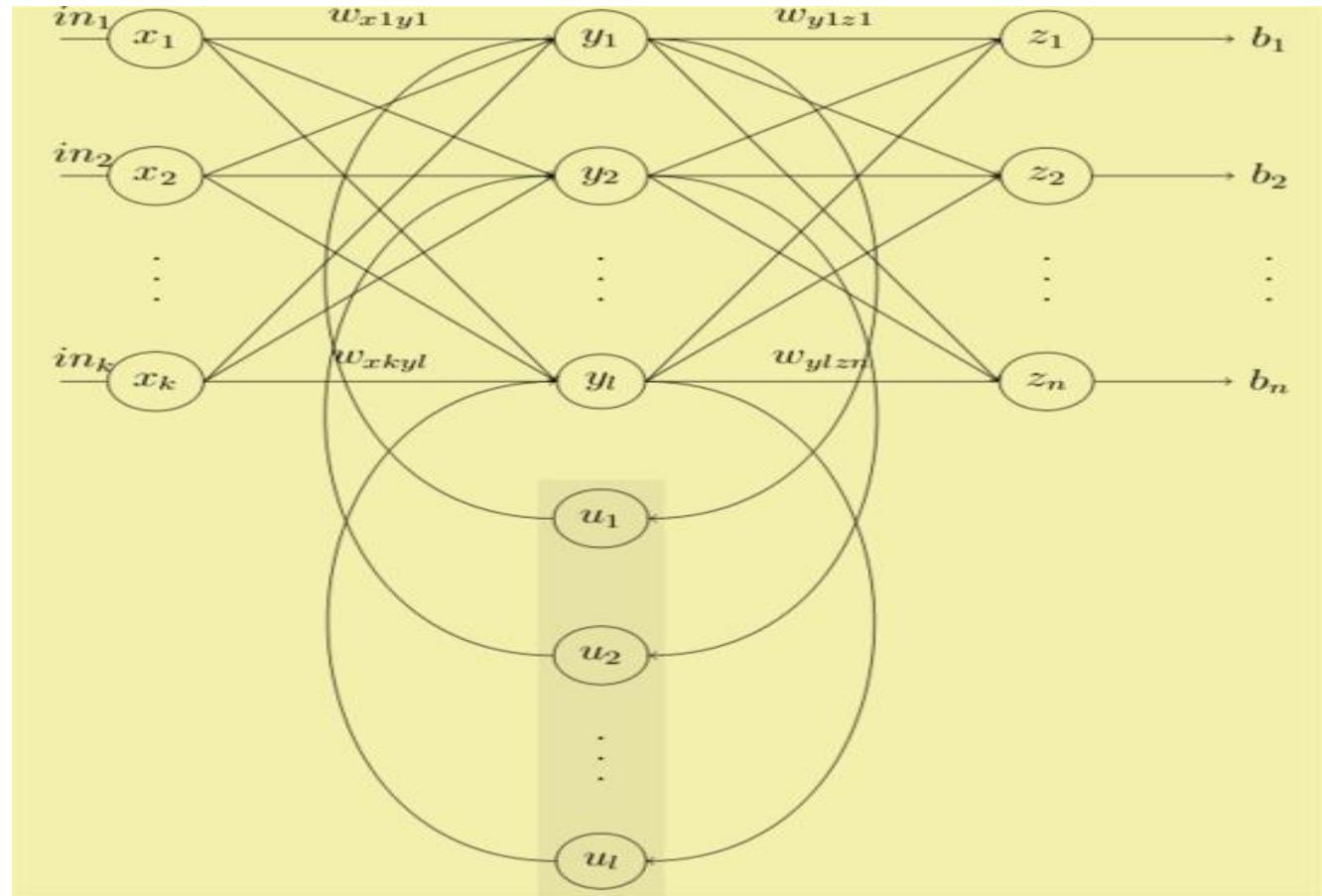
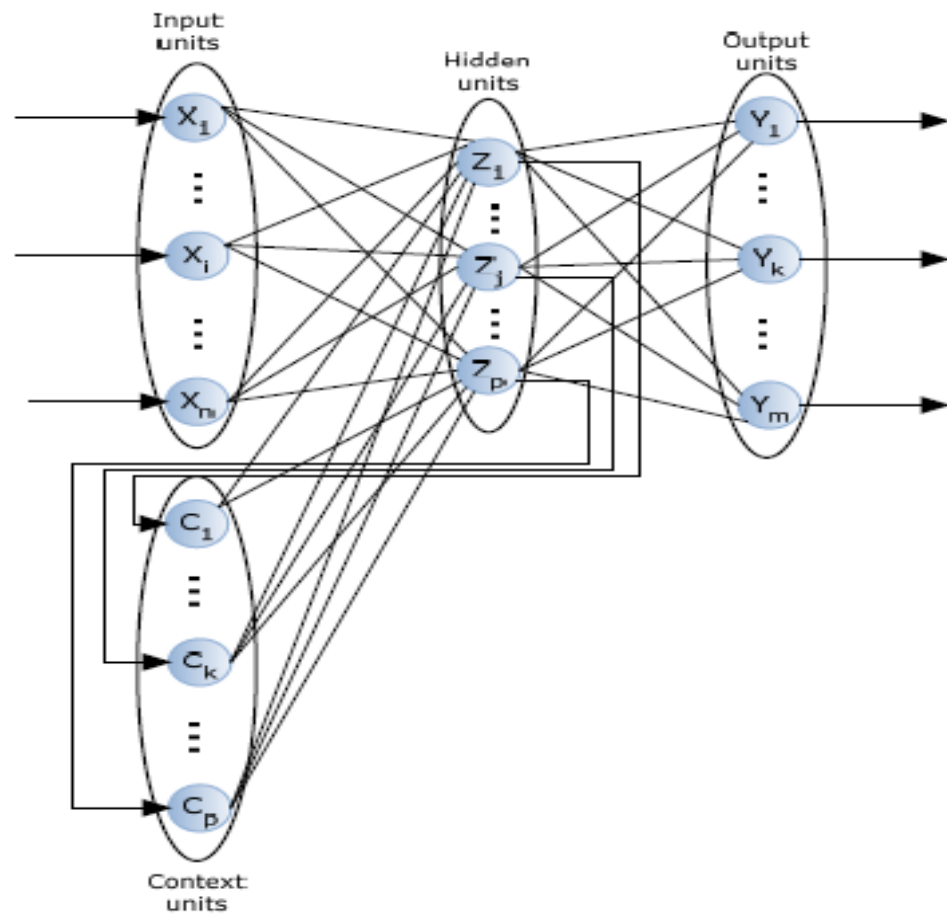
# RNN – Mạng Hopfield

---

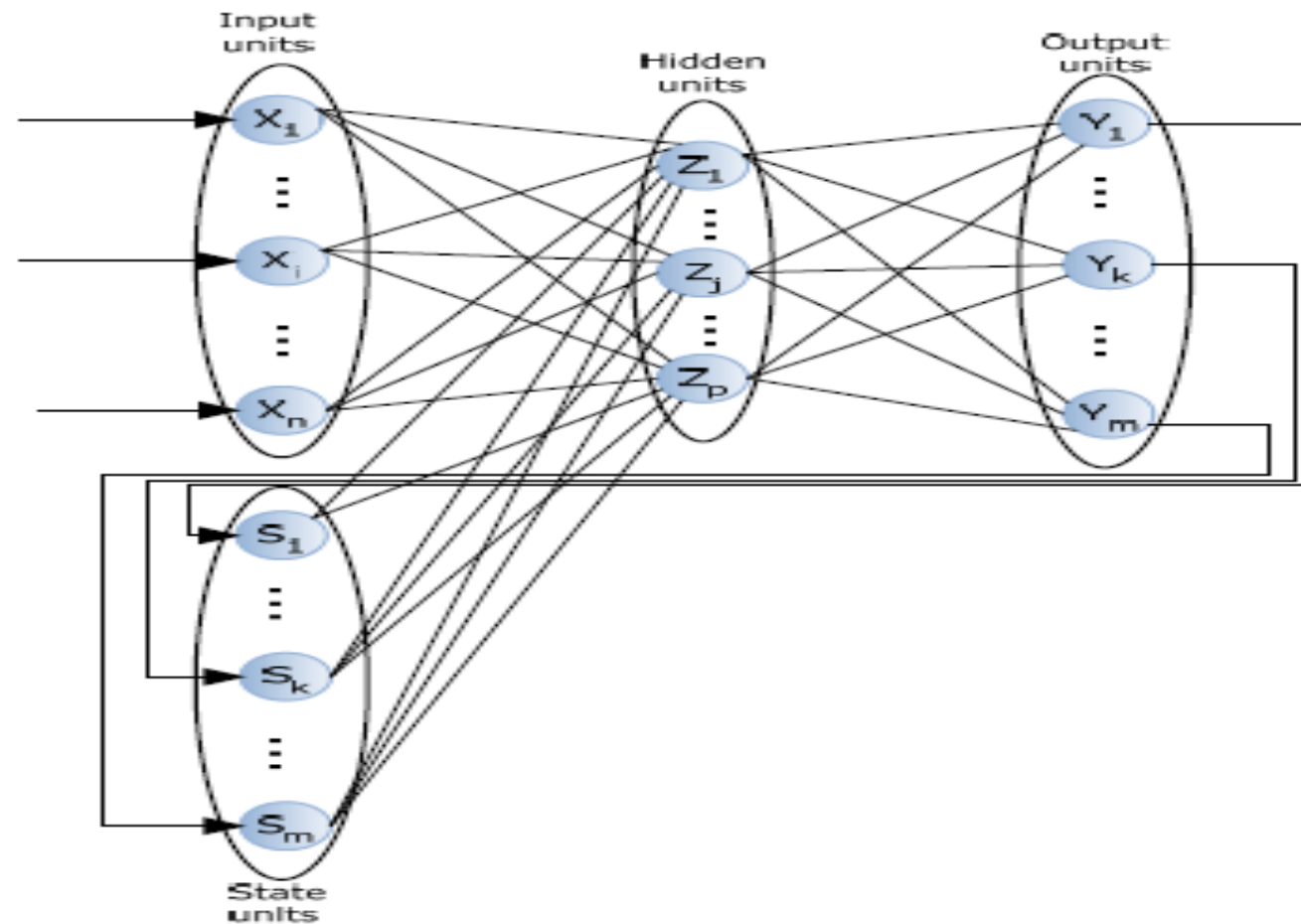
■ Weight Matrix:  $W = \begin{bmatrix} 0 & w_{12} & w_{13} & \dots & w_{1n} \\ w_{12} & 0 & w_{23} & \dots & w_{2n} \\ w_{13} & w_{23} & 0 & \dots & w_{3n} \\ \vdots & \vdots & & \ddots & \vdots \\ w_{1n} & w_{2n} & w_{3n} & \dots & 0 \end{bmatrix}$

■ Inputs:  $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$  where  $x_j$  is either 1 or -1

# RNN – mạng Elman

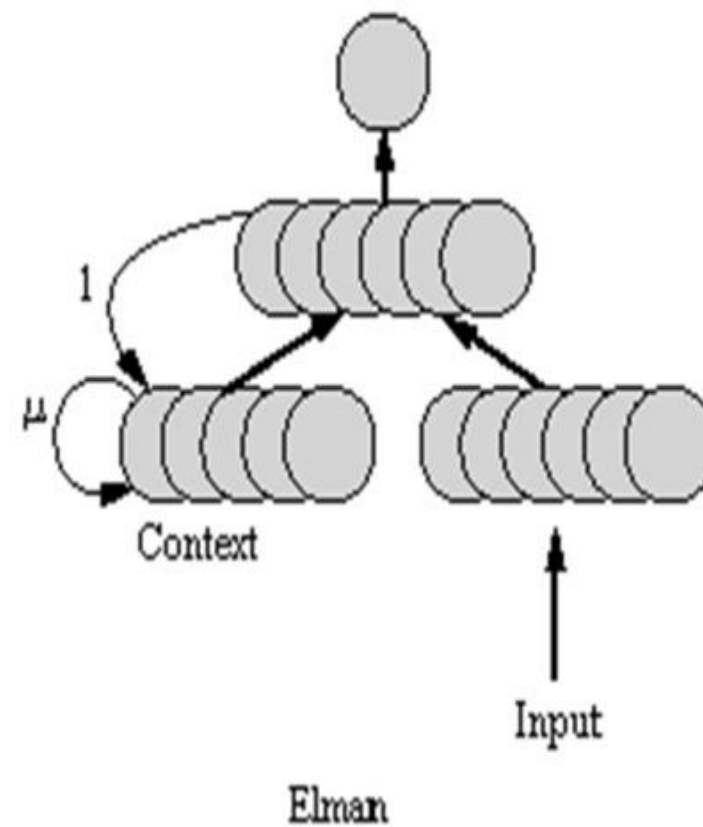
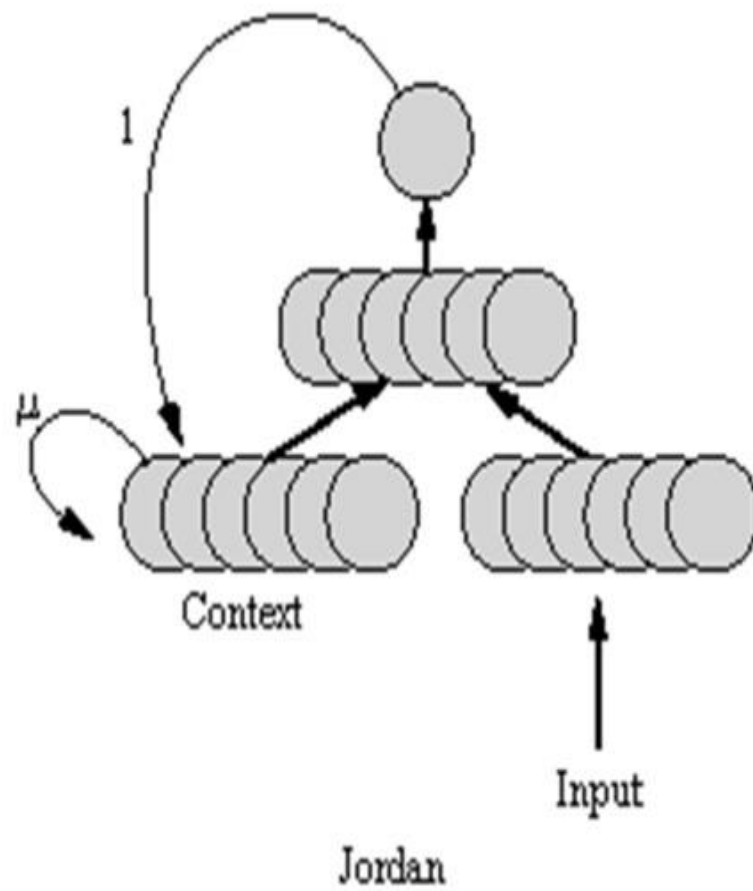


# RNN – mạng Jordan



# RNN

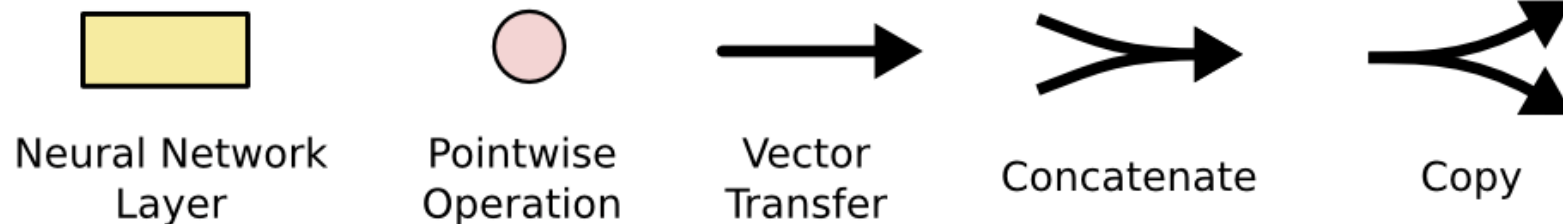
---





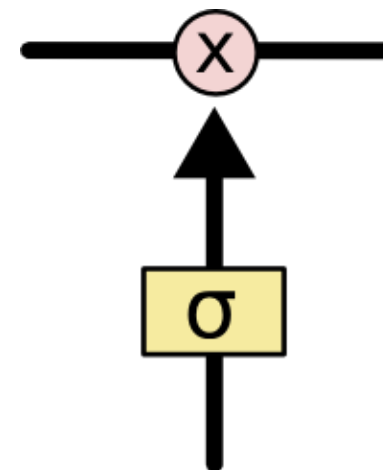
# Long Short Term Memory (LSTM)

Một số ký hiệu cần nhớ



Lớp sigmoid với đầu ra mang giá trị nằm giữa 0-1 cho phép xác định có bao nhiêu thông tin được truyền qua.

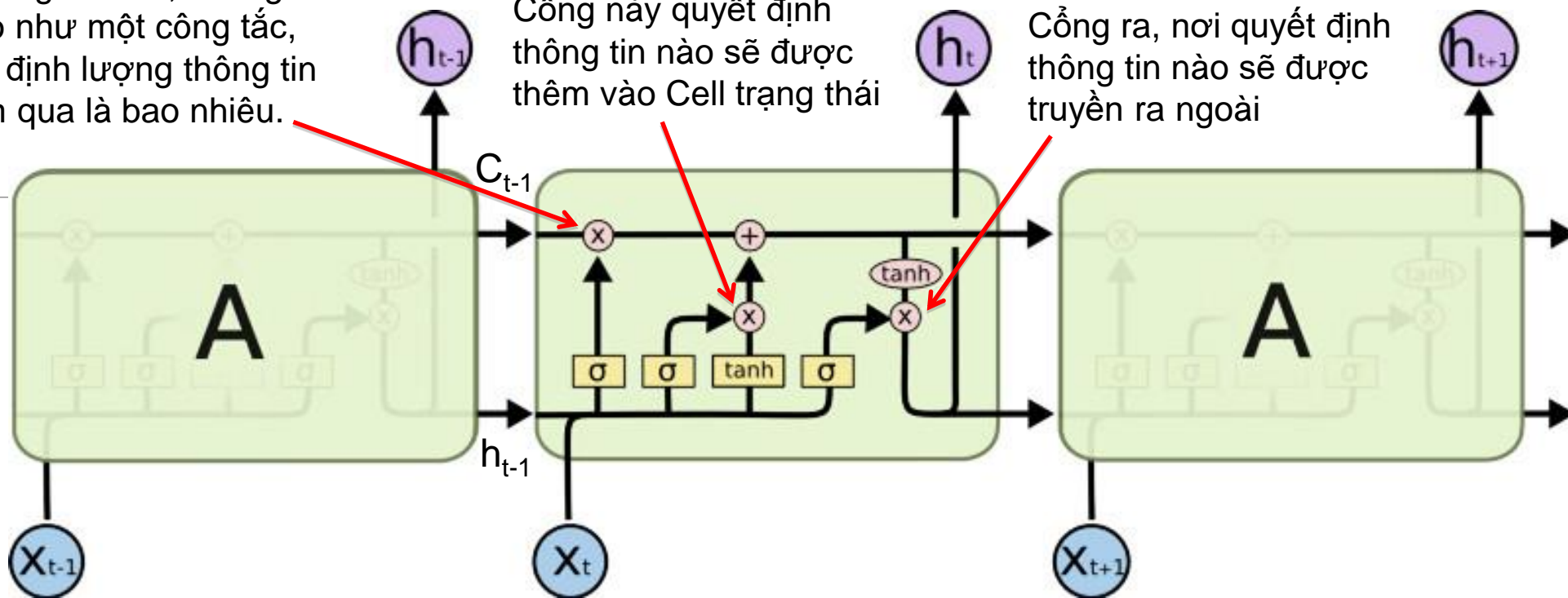
Cổng X màu hồng biểu diễn phép nhân từng điểm.



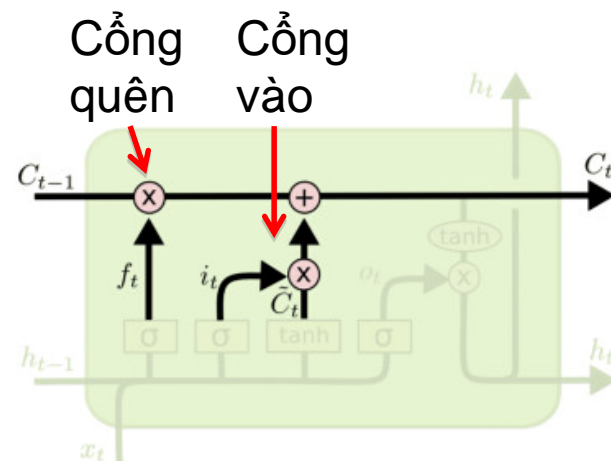
Cổng Sigmoid: 0,1 đóng vai trò như một công tắc, quyết định lượng thông tin truyền qua là bao nhiêu.

Cổng này quyết định thông tin nào sẽ được thêm vào Cell trạng thái

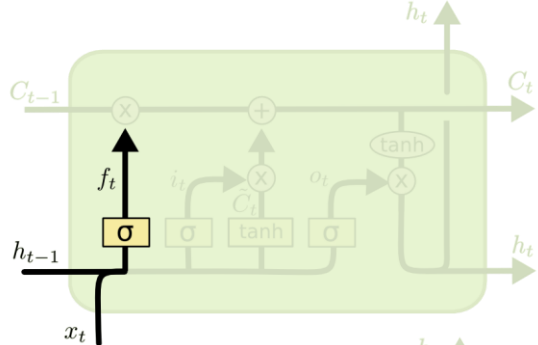
Cổng ra, nơi quyết định thông tin nào sẽ được truyền ra ngoài



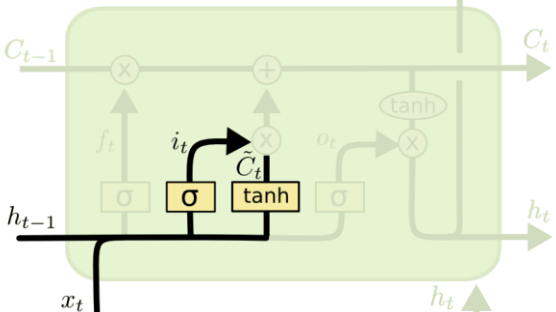
Cell trạng thái  $C_t$  đóng vai trò quan trọng nhất trong LSTM, nó được thay đổi rất chậm, cho phép truyền thông tin qua nó một cách dễ dàng và ít bị thay đổi.



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

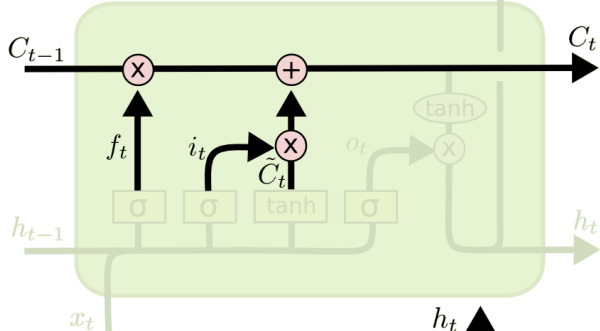


$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

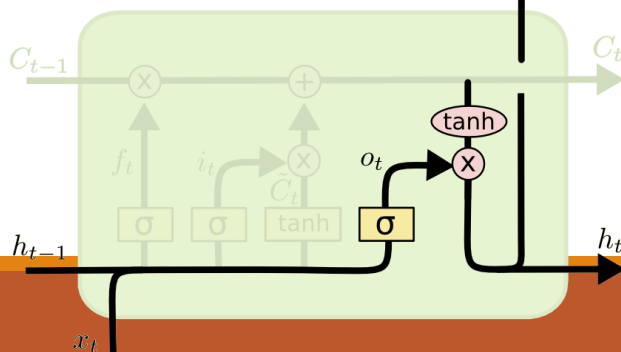


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

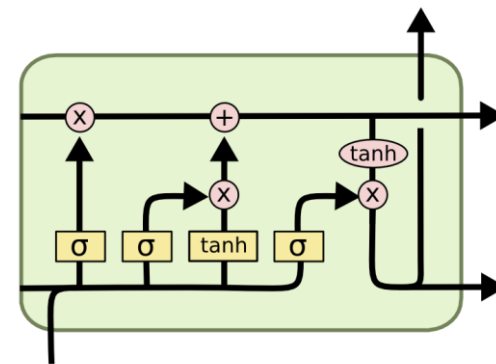


$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

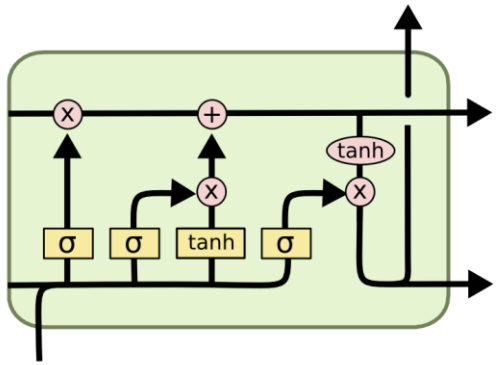


$i_t$  xác định thành phần nào trong mạng sẽ được cập nhật.  
 $C'_t$  thực hiện thay đổi nội dung

Cập nhật Cell trạng thái

Xác định phần nào của Cell trạng thái được đưa thông tin ra đầu ra

4 ma trận trọng số này cần được cập nhật đồng thời



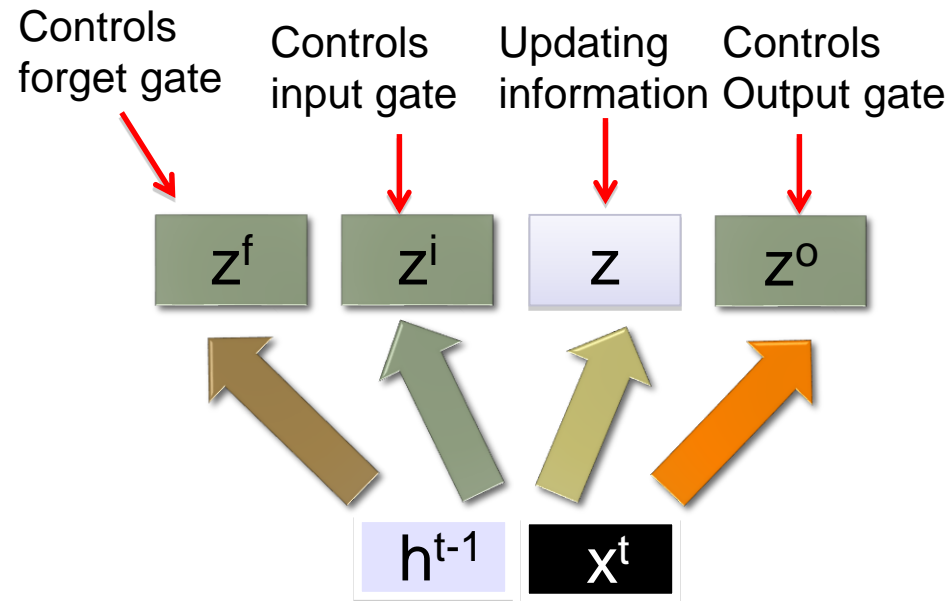
$c^{t-1}$

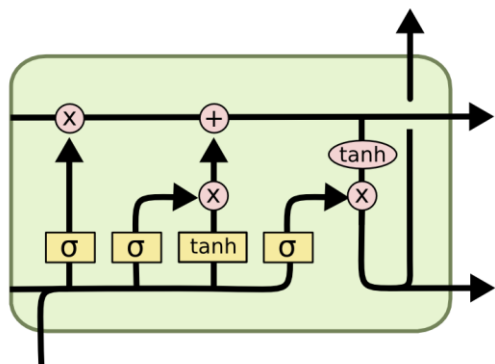
$$z = \tanh(W \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix})$$

$$z^i = \sigma(W^i \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix})$$

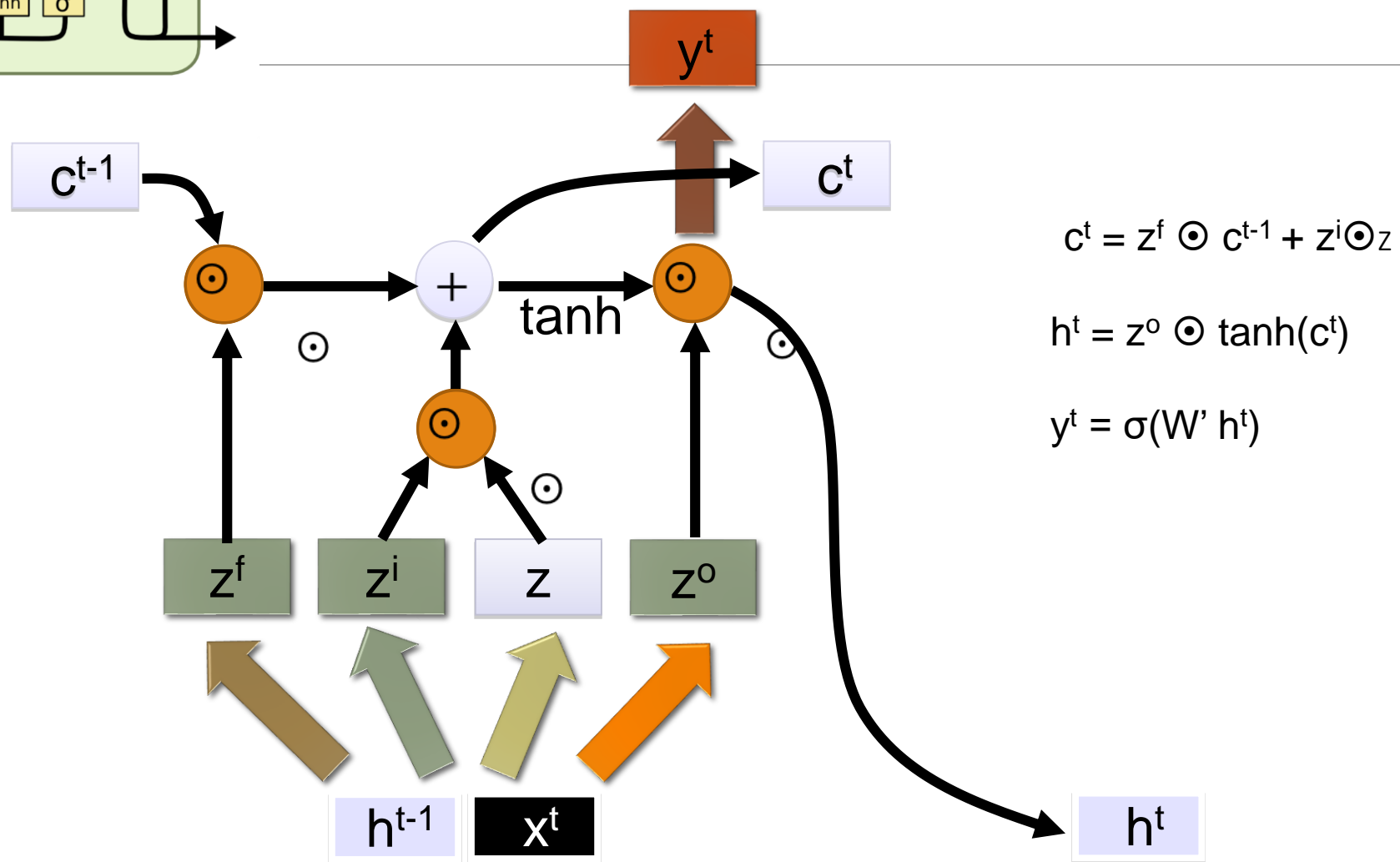
$$z^f = \sigma(W^f \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix})$$

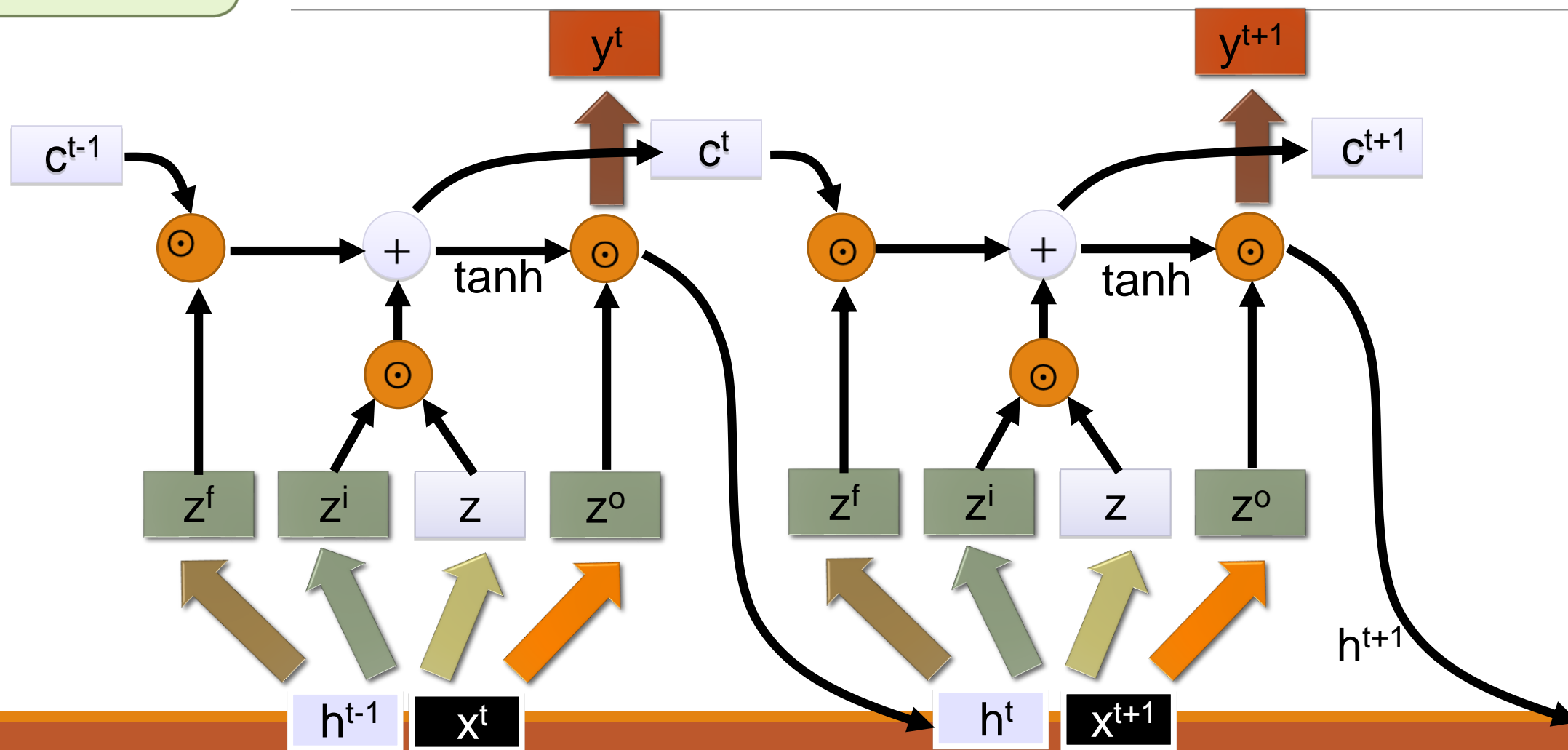
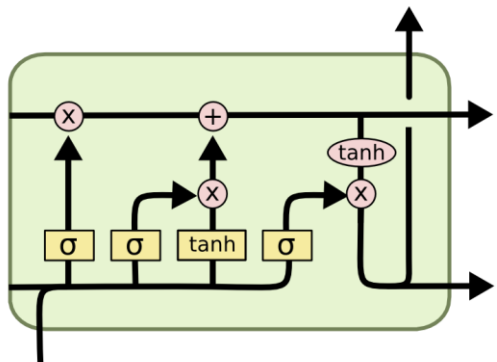
$$z^o = \sigma(W^o \begin{bmatrix} x^t \\ h^{t-1} \end{bmatrix})$$



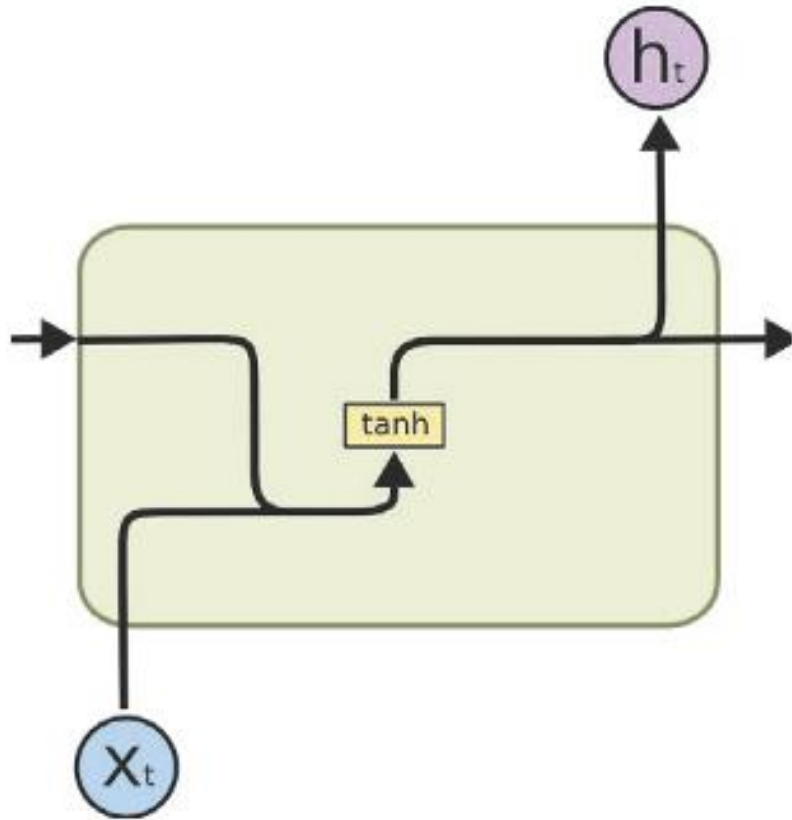


⊙ Element-wise multiply

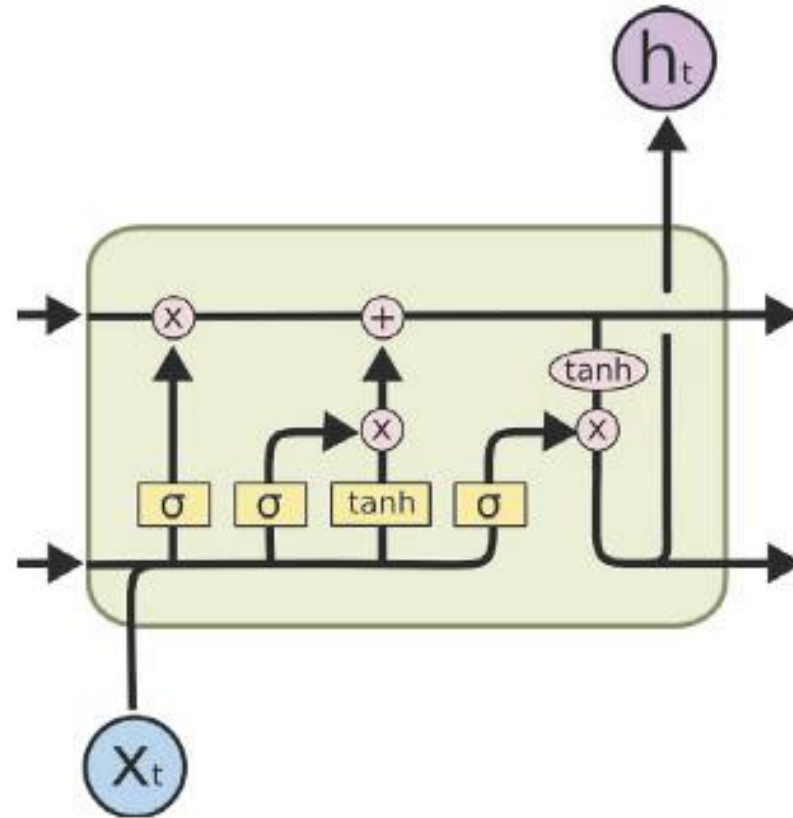




# RNN so với LSTM



(a) RNN

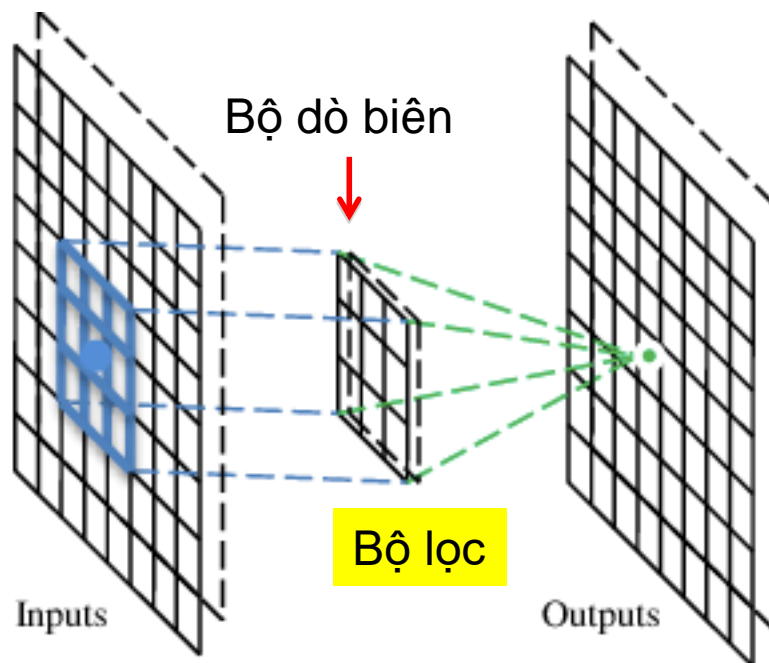


(b) LSTM

# Mạng nơ ron tích chập (CNN)

Convolutional Neural Networks (CNN) là một mạng nơ ron gồm một số tầng tích chập (và một số các tầng nơ ron khác).

Tầng tích chập bao gồm nhiều bộ lọc thực hiện phép nhân chập





# Nhân chập

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Bộ lọc 1

-1	1	-1
-1	1	-1
-1	1	-1

Bộ lọc 2

⋮ ⋮

Mỗi bộ lọc thực hiện dò tìm các mẫu có kích thước (3 x 3).

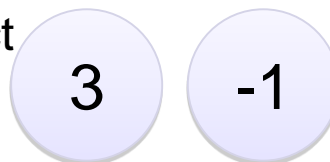
# Nhân chập

Bước nhảy = 1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Dot  
product  
→



1	-1	-1
-1	1	-1
-1	-1	1

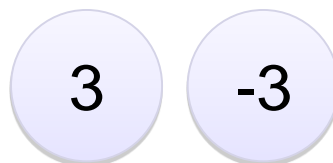
Bộ lọc 1

# Nhân chập

Bước nhảy = 2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image



1	-1	-1
-1	1	-1
-1	-1	1

Bộ lọc 1

# Nhân chập

Bước nhảy = 1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

1	-1	-1
-1	1	-1
-1	-1	1

Bộ lọc 1

# Nhân chập

Bước nhảy = 1

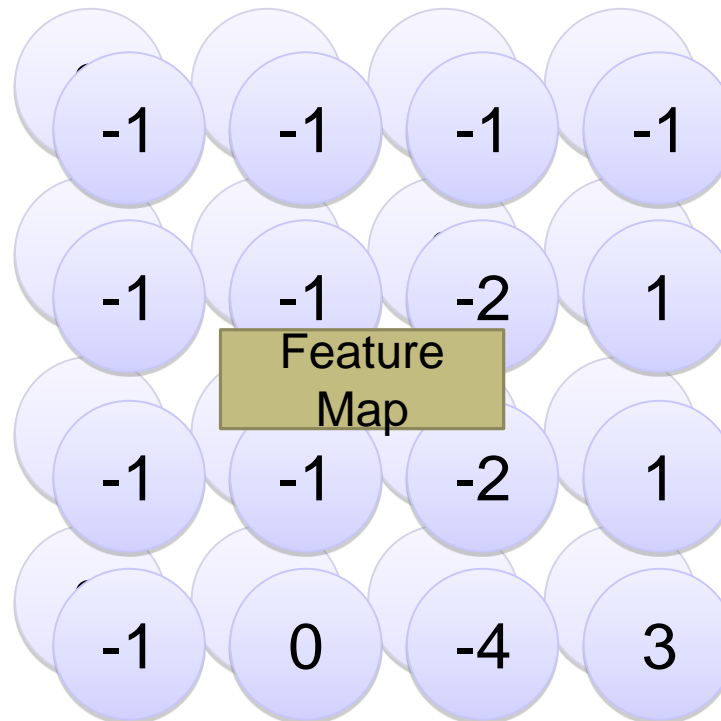
-1	1	-1
-1	1	-1
-1	1	-1

Bộ lọc 2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

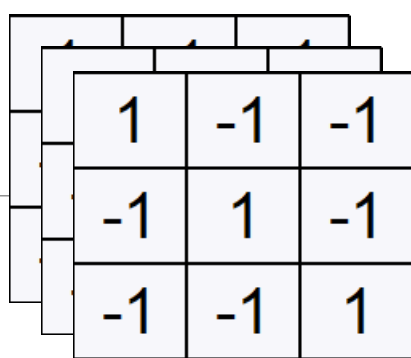
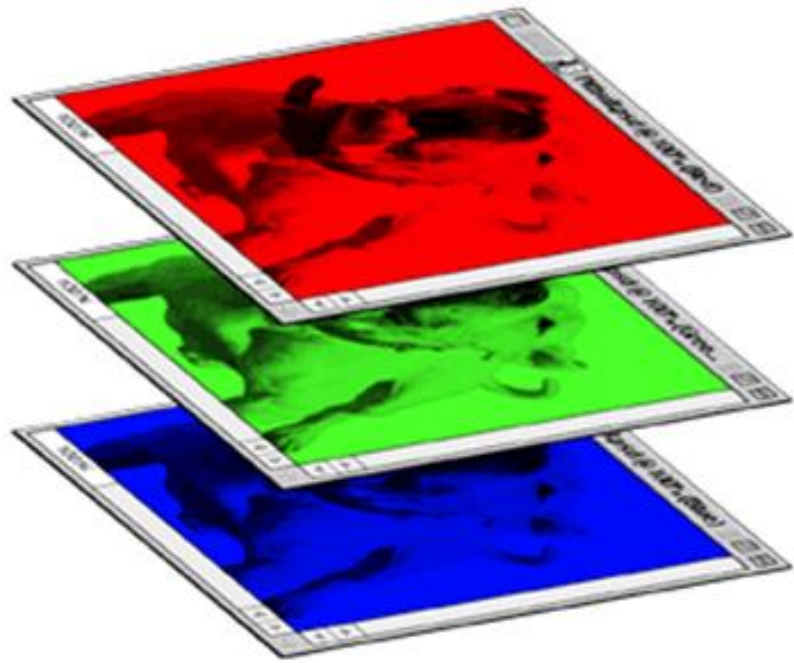
6 x 6 image

Quá trình lặp lại cho mỗi bộ lọc khác nhau

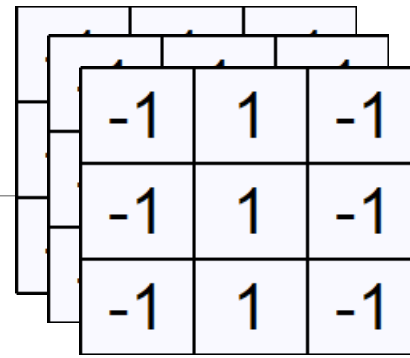


Hai ảnh 4 x 4 tạo nên ma trận  
3 chiều 2 x 4 x 4

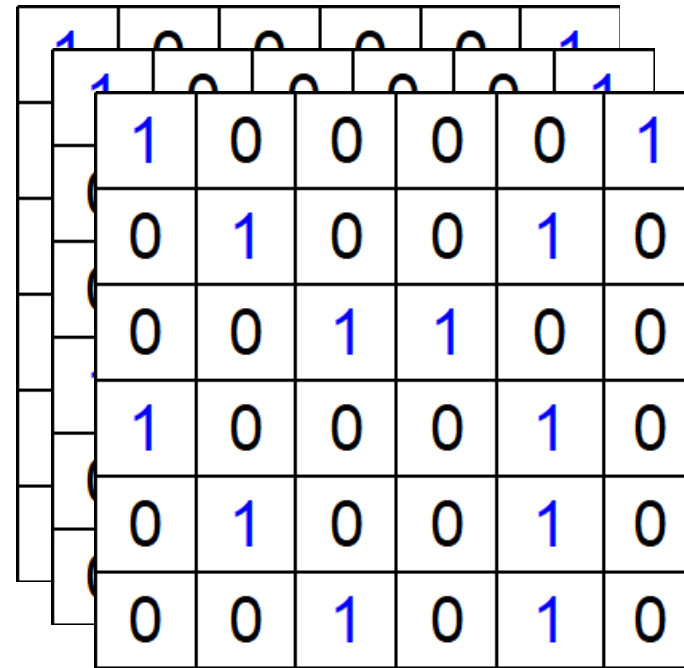
## Ảnh màu RGB



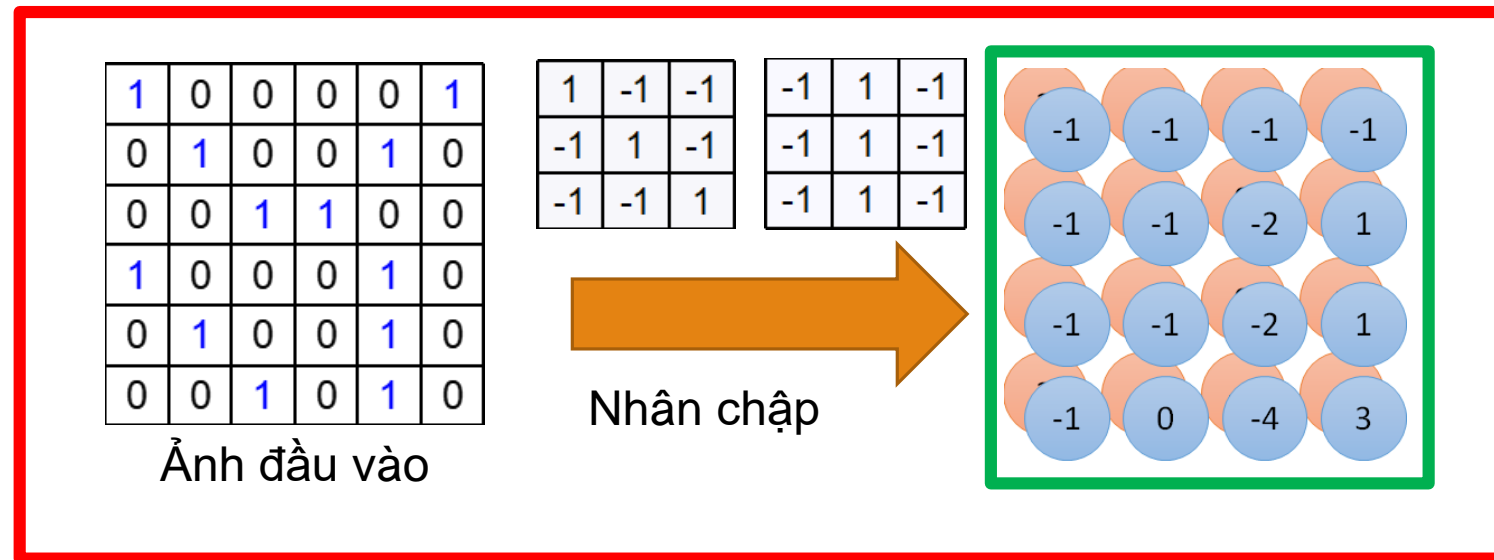
## Bộ lọc 1



## Bộ lọc 2



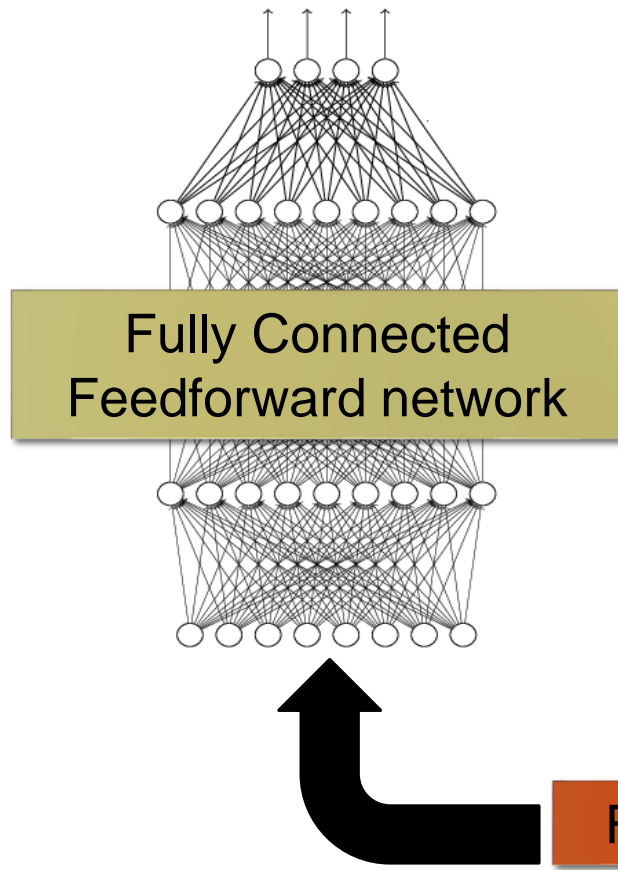
# Nhân chập



# The whole CNN



cat dog .....



Convolution

Max Pooling

Convolution

Max Pooling

Flattened

Có thể lặp lại nhiều lần các tổ hợp này