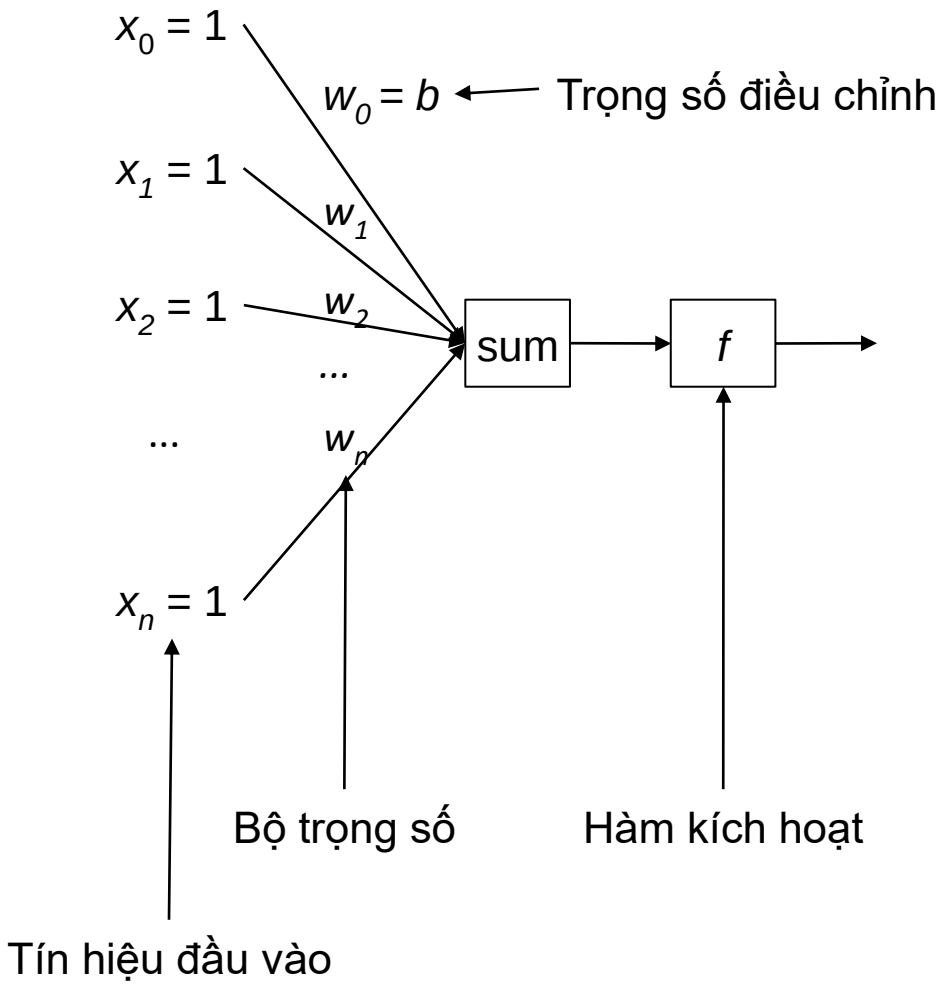
BÀI 2: HỌC MÁY (TIẾP)

7. Mạng nơ-ron tiến

- Mạng nơ-ron nhân tạo (ANN) mô phỏng hệ thống nơ-ron sinh học của bộ não người, là một mạng lưới bao gồm các nơ-ron nhân tạo liên kết với nhau. ANN có thể coi là một kiến trúc tính toán phân tán và song song
- Mỗi nơ-ron nhận tín hiệu đầu vào, thực hiện tính toán cục bộ tạo thành tín hiệu đầu ra. Giá trị đầu ra phụ thuộc vào đặc tính của mỗi nơ-ron và các liên kết của nó với các nơ-ron khác trong mạng
- ANN thực hiện việc học, ghi nhớ, và tổng quát hóa thông qua việc cập nhật giá trị trọng số của các liên kết giữa các nơ-ron
- Hàm mục tiêu phụ thuộc vào kiến trúc mạng, đặc tính của mỗi nơ-ron, chiến lược học, và dữ liệu học

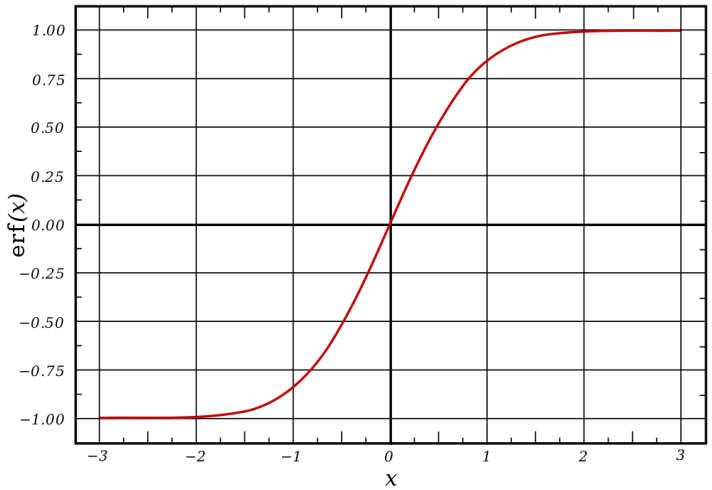
Nơ-ron nhân tạo (perceptron)



Hàm kích hoạt sigmoid

$$f(u) = \frac{1}{1 + e^{-\alpha(u + \theta)}}$$

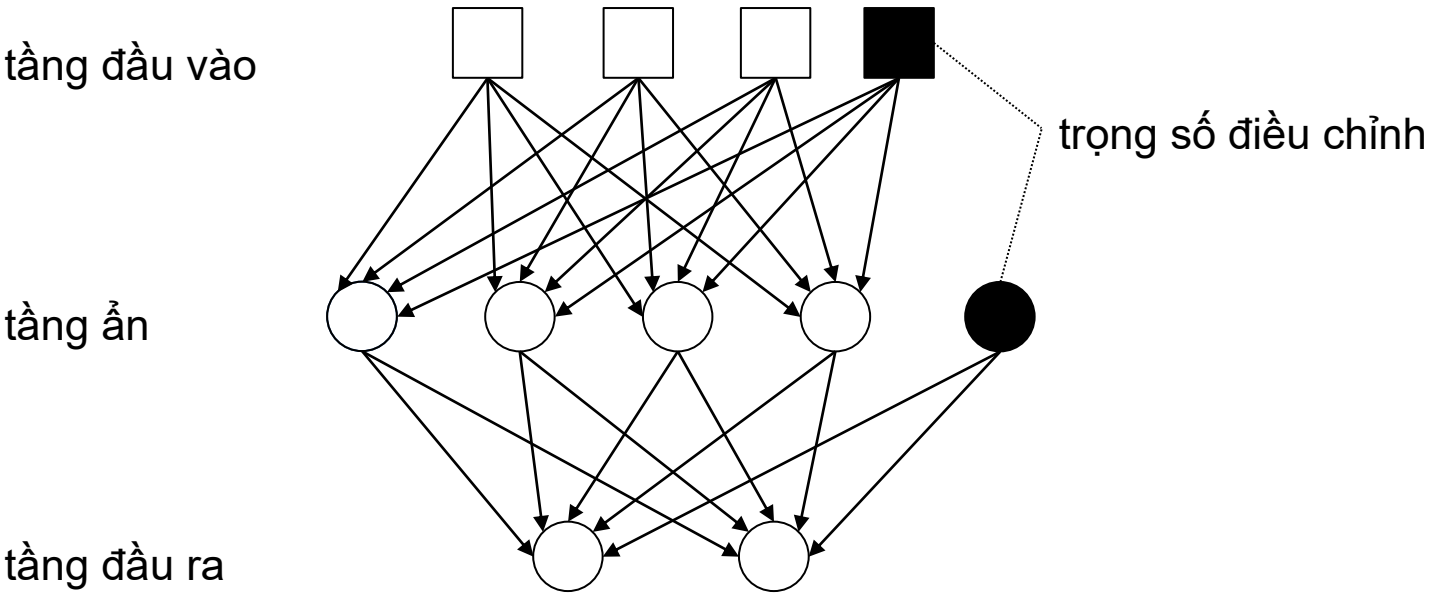
- Được dùng phổ biến
- Tham số α xác định độ dốc
- Giá trị trong khoảng (0,1)
- Hàm và đạo hàm liên tục



Kiến trúc mạng ANN

- Kiến trúc mạng ANN được xác định bởi:
  - Số lượng tín hiệu đầu vào/đầu ra
  - Số lượng tầng
  - Số nơ-ron trong mỗi tầng
  - Sự liên kết của các nơ-ron
- Một tầng gồm một nhóm các nơ-ron
  - Tầng đầu vào nhận tín hiệu đầu vào
  - Tầng đầu ra trả về tín hiệu đầu ra
  - (Các) tầng ẩn nằm giữa tầng đầu vào và đầu ra
- Trong mạng lan truyền tiến (FNN), đầu ra của một nơ-ron không liên kết ngược trở lại làm đầu vào của nơ-ron khác trong cùng tầng hoặc một tầng trước đó

Ví dụ FNN



VD: FNN có 3 tầng
- Tầng đầu vào gồm có 4 tín hiệu
- Tầng ẩn có 5 nơ-ron
- Tầng đầu ra có 2 nơ-ron ứng với 2 tín hiệu đầu ra

Số tham số: 4 x 4 + 5 x 2 = 26
(các mạng nơ-ron trong thực tế có ~10^6 tham số)

Hàm lỗi

- Xét một ANN có 1 giá trị đầu ra
- Đ/v một ví dụ ($\boldsymbol{x}$, y), hàm lỗi

$E_{\boldsymbol{x}}(\boldsymbol{w}) = 1/2(y-y')^2$

trong đó y' là giá trị đầu ra của ANN

- Hàm lỗi đối với tập dữ liệu D

$E_D(\boldsymbol{w}) = 1/|D| \sum_{\boldsymbol{x}\in D} E_{\boldsymbol{x}}(\boldsymbol{w})$

Suy giảm gradient

- Gradient của E (ký hiệu ∇E) là một véc-tơ hướng lên trên có độ dài tỉ lệ với độ dốc của E
- ∇E xác định hướng gây ra việc tăng nhanh nhất giá trị E

$$\nabla E(\boldsymbol{w}) = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_n} \right]$$

trong đó n là tổng số trọng số liên kết trong mạng

- Cần tìm hướng làm giảm nhanh nhất giá trị E

$$\Delta \boldsymbol{w} = -\eta . \nabla E(\boldsymbol{w})$$

trong đó η là tốc độ học

- Các hàm kích hoạt trong mạng phải liên tục và có đạo hàm liên tục

Thuật toán suy giảm gradient

Algorithm Gradient_descent_incremental((*D*, *η*))
1 Khởi tạo ***w*** ($w_i \leftarrow$ một giá trị ngẫu nhiên nhỏ)
2 **do**
3 **for** mỗi ví dụ huấn luyện ($\boldsymbol{x}$, *d*) $\in D$ **do**
4 Tính toán đầu ra của mạng
5 **for** mỗi thành phần trọng số w_i **do**
6 $w_i \leftarrow w_i - \eta(\partial E_x / \partial w_i)$
7 **endfor**
8 **endfor**
9 **until** (thỏa mãn điều kiện dừng)
10 **return** ***w***

Giải thuật lan truyền ngược

- Perceptron chỉ biểu diễn được hàm phân tách tuyến tính
- Mạng nơ-ron nhiều tầng có thể biểu diễn được các hàm phân tách phi tuyến
- Giải thuật lan truyền ngược:
  - Lan truyền tiến (tín hiệu): Các tín hiệu đầu vào được lan truyền qua các tầng đến tầng đầu ra
  - Lan truyền ngược (lỗi):
    - Tính toán lỗi dựa trên đầu ra mong muốn
    - Lỗi được lan truyền từ tầng đầu ra đến các tầng trong mạng cho đến tầng đầu vào
    - Giá trị lỗi được tính toán hồi quy dựa trên giá trị lỗi cục bộ tại mỗi nơ-ron

Khởi tạo trọng số liên kết

- Khởi tạo ngẫu nhiên
- Nếu giá trị khởi tạo lớn, hàm kích hoạt sigmoid cho giá trị lớn dẫn đến tình trạng bão hòa sớm khiến hệ thống dừng lại ở điểm cực tiểu cục bộ hoặc ở đường nằm ngang gần điểm khởi đầu
- Với trọng số liên kết w^0_{ab} (liên kết từ nơ-ron b tới nơ-ron a)
  - $w^0_{ab} \in [-1/n_a, 1/n_a]$ trong đó n_a là số nơ-ron cùng tầng với a
  - $w^0_{ab} \in [-3/k_a^{1/2}, 3/k_a^{1/2}]$ trong đó k_a là số nơ-ron ở tầng trước liên kết với a

Tốc độ học

- Tốc độ học lớn đẩy nhanh quá trình học nhưng có thể bỏ qua điểm tối ưu toàn cục và rơi vào điểm tối ưu cục bộ
- Tốc độ học nhỏ làm chậm quá trình học
- Tốc độ học thường được lựa chọn dựa trên thực nghiệm
- Nên thay đổi tốc độ học trong quá trình học

Số lượng nơ-ron ở tầng ẩn

- Kích thước (số lượng nơ-ron) của tầng ẩn thường được xác định qua thực nghiệm
- Bắt đầu với kích thước nhỏ (so với số lượng tín hiệu đầu vào)
- Tăng dần kích thước nếu mạng không thể hội tụ
- Xem xét giảm dần kích thước nếu mạng hội tụ

Giới hạn học của ANN

- ANN một tầng ẩn có thể học bất kỳ hàm nhị phân nào
- ANN một tầng ẩn có thể học bất kỳ hàm liên tục bị chặn nào
- ANN hai tầng ẩn có thể học bất kỳ hàm liên tục nào

Ưu, nhược điểm

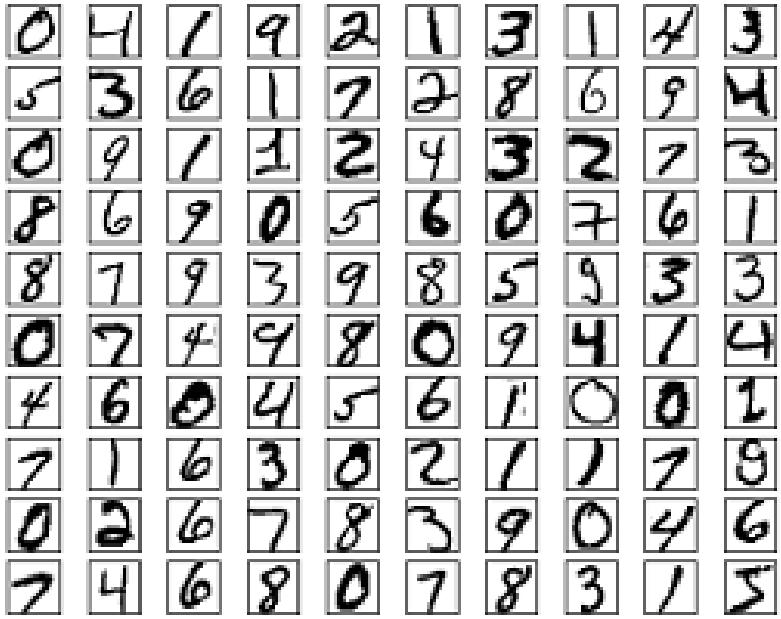
- Ưu điểm:
  - Hỗ trợ tính toán song song
  - Khả năng chịu nhiễu/lỗi
  - Tự thích nghi
- Nhược điểm:
  - Không có quy tắc xác định cấu trúc mạng và siêu tham số học cho một lớp bài toán nhất định
  - Không có phương pháp để đánh giá hoạt động bên trong của mạng
  - Khó đưa ra giải thích cho người dùng

Ứng dụng của ANN

- Ví dụ chứa nhiều thuộc tính rời rạc và liên tục
- Giá trị đầu ra có kiểu số thực, rời rạc, hoặc véc-tơ
- Dữ liệu có thể chứa nhiễu/lỗi
- Không cần thiết phải giải thích kết quả
- Chấp nhận thời gian huấn luyện lâu
- Yêu cầu thời gian phân loại/dự đoán nhanh

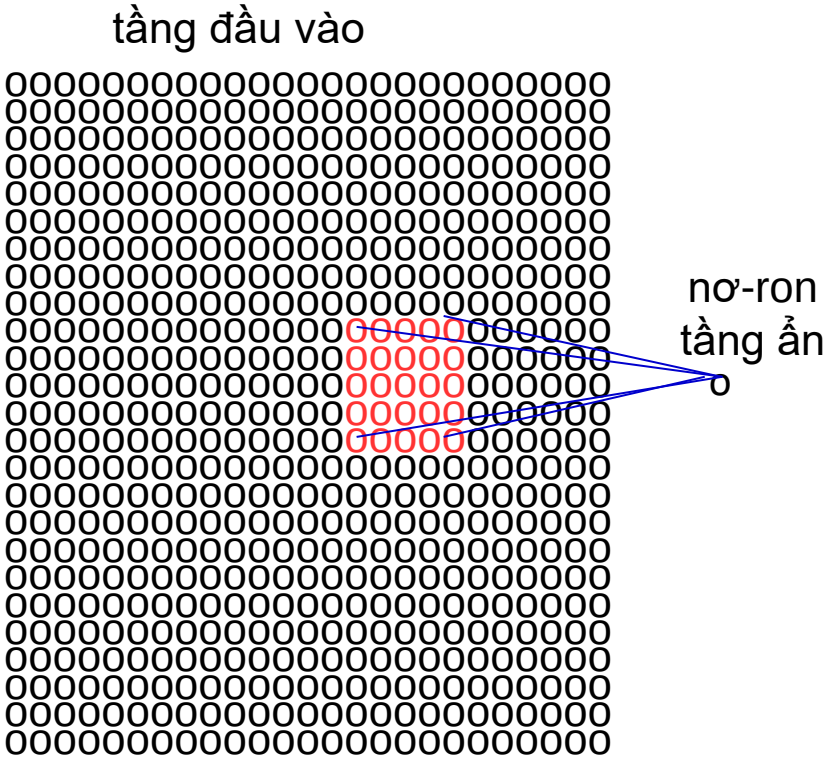
8. Mạng nơ-ron tích chập

- Bài toán nhận dạng chữ số [0..9]
  - Đầu vào: Ảnh chứa một số
  - Đầu ra: Phân loại [0..9]
- Tập DL MNIST:
  - Kích thước ảnh 28 x 28
  - DL huấn luyện: 60K
  - DL kiểm thử: 10K
- FNN không tận dụng được thông tin về không gian giữa các pixel trong ảnh

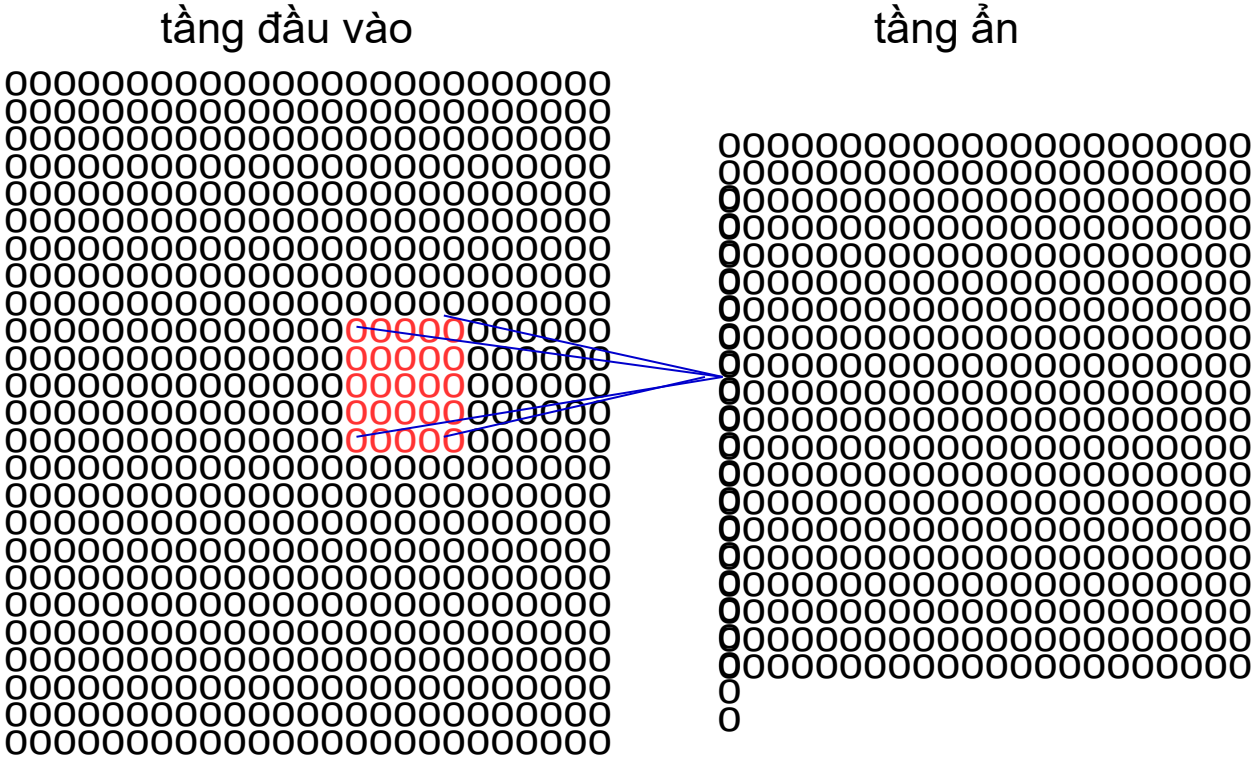


Vùng cảm thụ cục bộ

- Mạng nơ-ron tích chập (CNN) mô phỏng hoạt động của thị giác
- Biểu diễn tín hiệu đầu vào dưới dạng ma trận 28 x 28
- Mỗi nơ-ron trong tầng ẩn chỉ liên kết với các tín hiệu đầu vào trong vùng 5 x 5 (tương đương 25 điểm ảnh)
- Cho vùng cảm thụ 'trượt' trên ảnh đầu vào, mỗi vị trí liên kết với một nơ-ron ở tầng ẩn
- Tầng ẩn có 24 x 24 nơ-ron



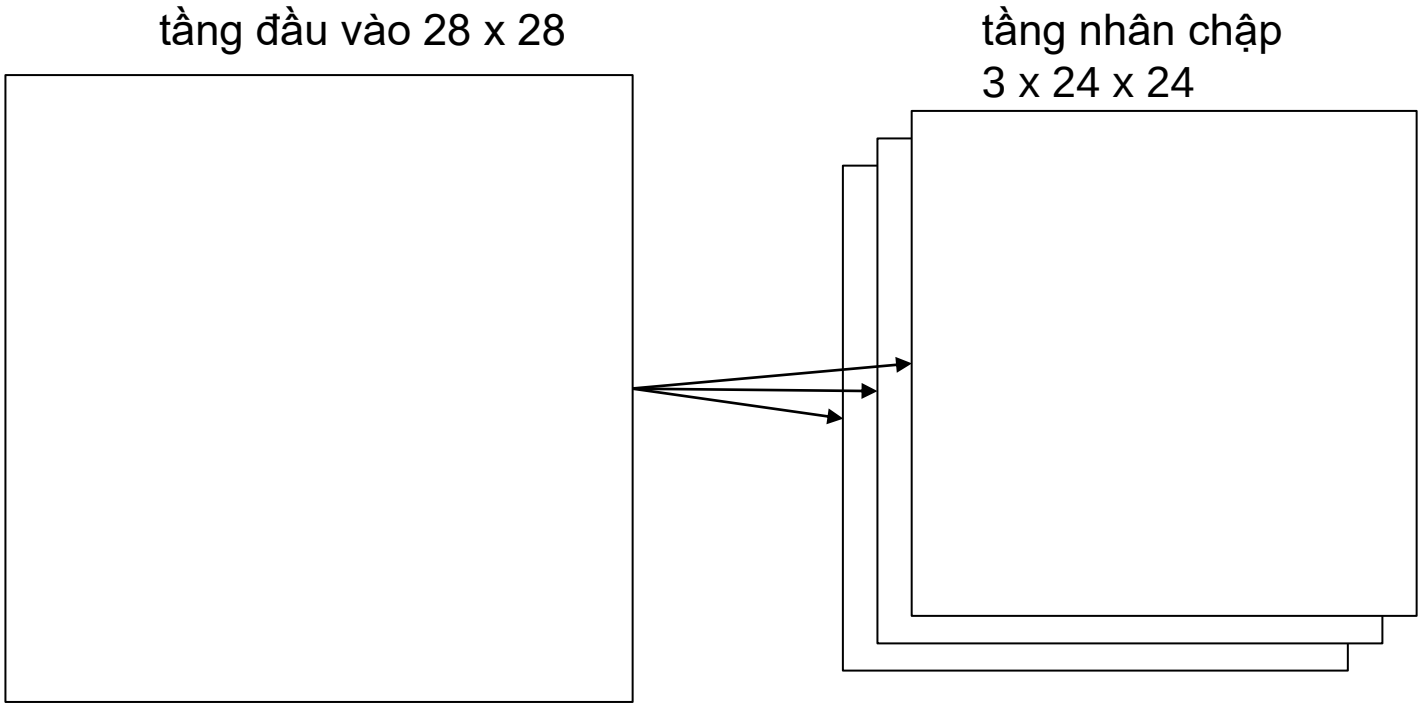
VD với bộ lọc 5 x 5



Chia sẻ trọng số

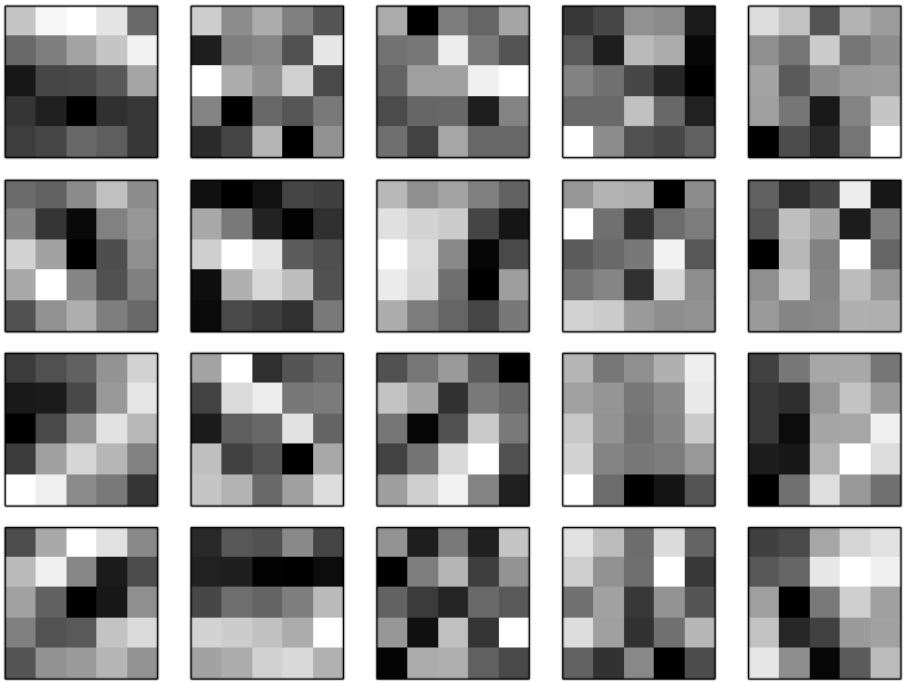
- Các trọng số (và trọng số điều chỉnh) của các vùng cảm thụ cục bộ được chia sẻ với nhau
- Giả thiết: Tầng ẩn có vai trò phát hiện ra cùng một đặc trưng thị giác (vd: nét móc lên trên sang phải) ở các vị trí khác nhau của ảnh do tính chất bất biến tịnh tiến (translational invariance) của ảnh
- Trọng số liên kết giữa tầng đầu vào và tầng ẩn được gọi là nhân (kernel) hoặc bộ lọc (filter)
- Giá trị tại nơ-ron của tầng ẩn được gọi là bản đồ đặc trưng (feature map)
- Mỗi tầng ẩn có nhiều bản đồ đặc trưng ứng với các nhân khác nhau

VD với 3 bộ lọc



VD trọng số của bộ lọc

- Ô trắng thể hiện giá trị trọng số nhỏ
- Ô đậm thể hiện giá trị trọng số lớn
- Các bộ lọc 'học' được các mẫu không gian



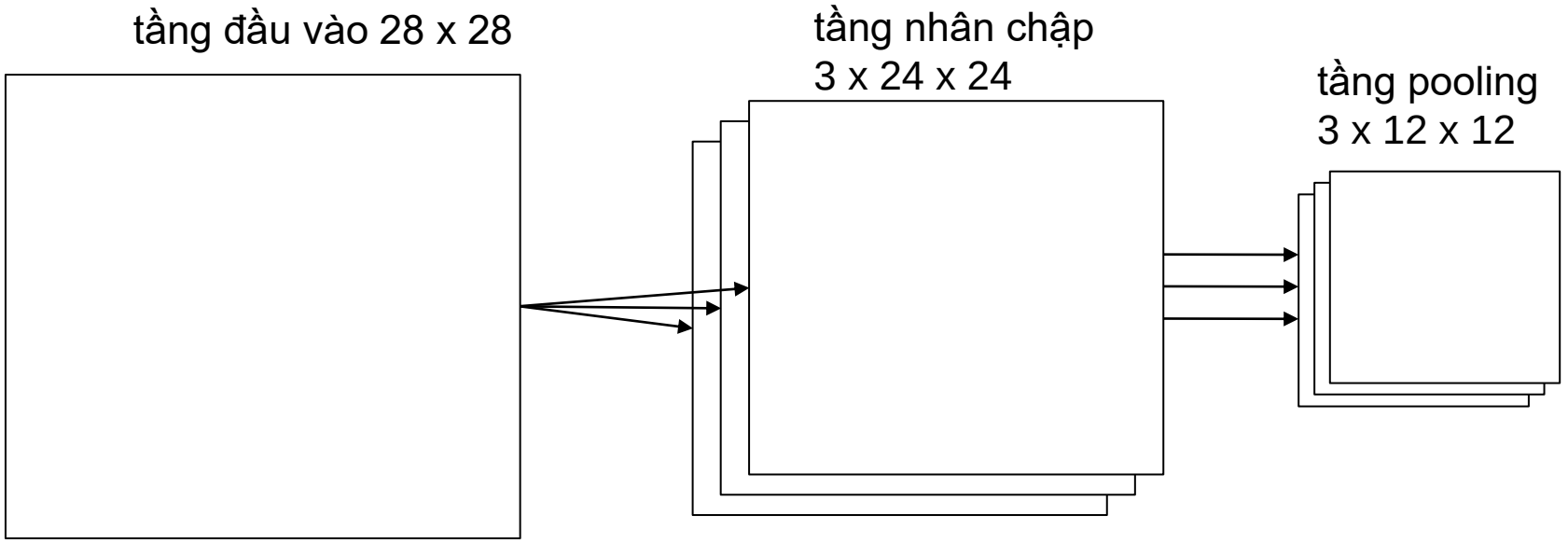
Số lượng tham số

- Số lượng tham số nhỏ hơn hàng chục lần so với mạng liên kết đầy đủ tương ứng
- Một bộ lọc cần 5 x 5 = 25 trọng số + 1 trọng số điều chỉnh
- 20 bộ lọc cần 20 x 26 = 520 tham số
- Một FNN có kích thước tầng ẩn = 30 cần 30 x 28 x 28 (liên kết) + 30 (điều chỉnh) = 23,550

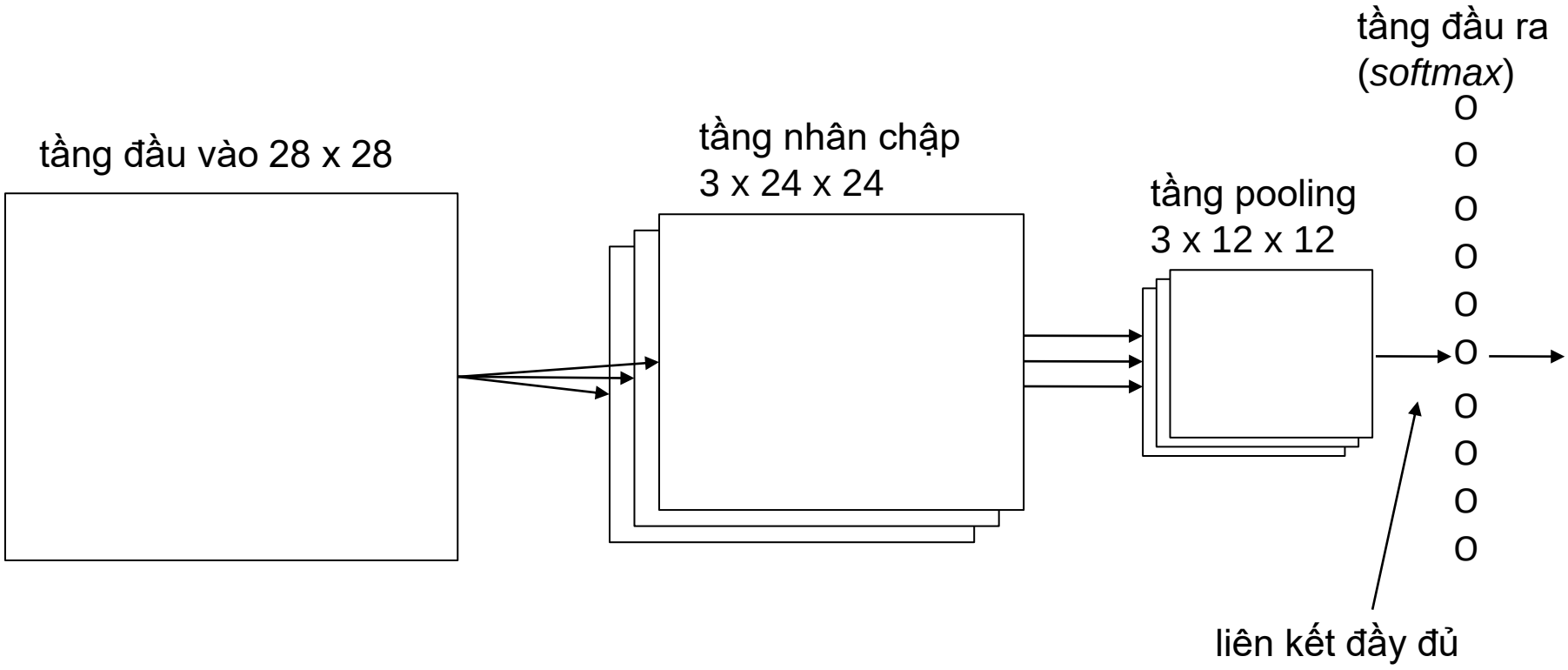
Tầng pooling

- Tầng pooling nằm kế tiếp tầng nhân chập
- Tầng pooling có vai trò tổng hợp những thông tin quan trọng nhất từ tầng nhân chập
- VD: bản đồ đặc trưng của được chia thành các vùng 2x2, mỗi vùng được đưa đến tầng pooling và trả về giá trị lớn nhất (max pooling)
- Pooling được áp dụng với mỗi bộ lọc
- Giả thiết: pooling có tác dụng xóa đi thông tin vị trí và chỉ giữ lại các đặc trưng
- L2 pooling: Tính căn bậc hai của tổng bình phương các giá trị

VD tầng pooling



VD mạng đầy đủ



Hàm kích hoạt softmax

- Giải quyết vấn đề "học chậm" của hàm lỗi bình phương
- z_j^L là đầu vào của nơ-ron j ở tầng đầu ra L
- a_j^L là giá trị đầu ra của nơ-ron j ở tầng đầu ra L sau khi qua hàm kích hoạt *softmax*
- Mô hình sử dụng hàm lỗi *log-likelihood*

$$a_j^L = \frac{e^{x_j^L}}{\sum_k e^{z_k^L}}$$

$$\sum_j a_j^L = \frac{\sum_j e^{z_j^L}}{\sum_k e^{z_k^L}} = 1.$$

$$C \equiv -\ln a_j^L$$

Khả năng tổng quát hóa của CNN

- Sử dụng nhiều bước nhân chập - pooling để học được các đặc trưng trừu tượng mức cao
- Lựa chọn các hàm kích hoạt (vd *sigmoid*, *tanh*, *relu)* dựa trên thực nghiệm
- Lựa chọn hàm kích hoạt - hàm lỗi: sigmoid - cross entropy vs softmax - log likelihood
- Áp dụng các kĩ thuật tránh overfit (vd drop out)

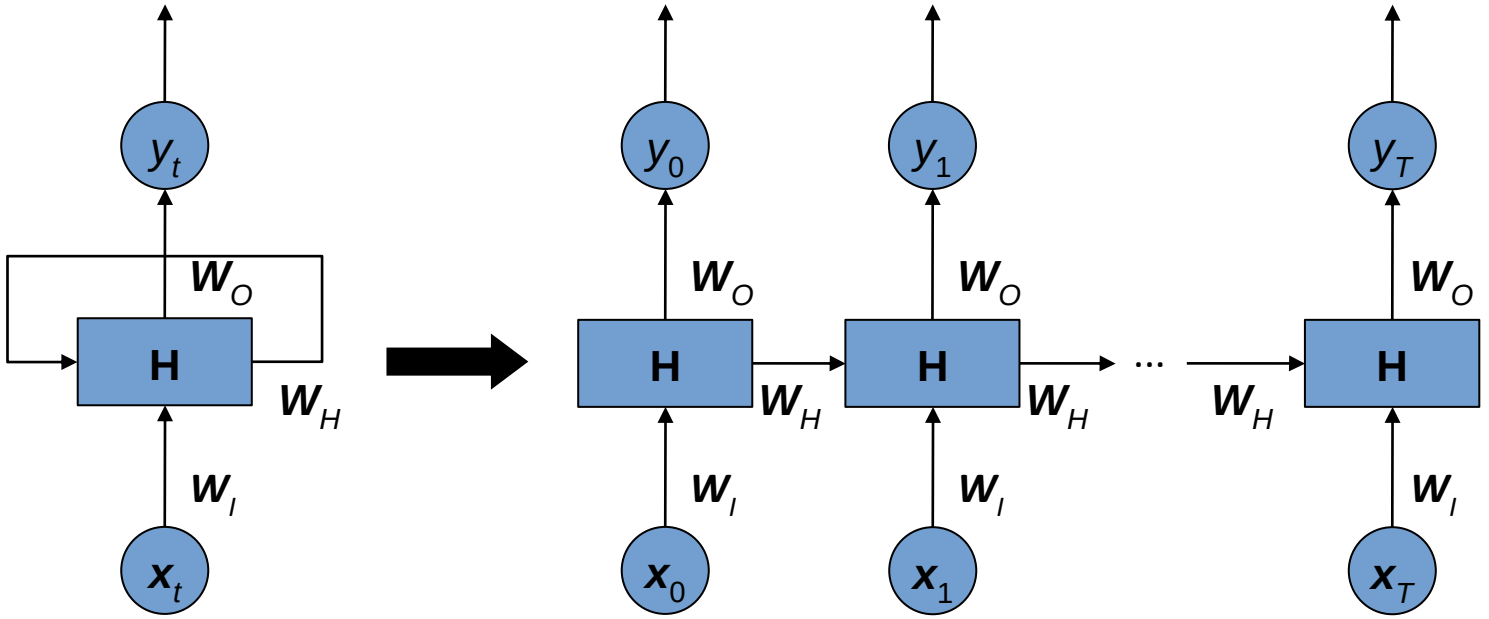
9. Mạng nơ-ron hồi quy

- Trong nhiều trường hợp, đầu ra ở thời điểm hiện tại không chỉ phụ thuộc vào tín hiệu ở thời điểm hiện tại mà còn phụ thuộc vào tín hiệu (và đầu ra) ở các thời điểm trước đó
- Các bài toán chuỗi thời gian (dự báo giá cả, dự báo thời tiết, dự báo chỉ số chứng khoán….)
- Các bài toán trong xử lý tiếng nói và ngôn ngữ (nhận diện tiếng nói, gán nhãn từ loại, phân loại cảm xúc)

Đáp ứng theo thời gian

- Mạng nơ-ron hồi quy (RNN) có khả năng lưu trữ các thông tin trong quá khứ
- Trạng thái của một nơ-ron ở tầng ẩn tại thời điểm (*t*-1) được sử dụng làm giá trị đầu vào tại thời điểm *t*
- Trọng số liên kết được chia sẻ theo thời gian

VD mạng RNN



Số lượng tham số

- I: Số nơ-ron của tầng đầu vào
- H: Số cell RNN (số nơ-ron của tầng ẩn)
- K: Số nơ-ron của tầng đầu ra
- Số tham số: I x H + H x H + H + H x K + K
  - Số liên kết giữa tầng đầu vào và tầng ẩn: I x H
  - Số liên kết hồi quy của tầng ẩn: H x H
  - Số liên kết của tầng ẩn và tầng đầu ra: H x K
  - Số tham số điều chỉnh của tầng ẩn: H
  - Số tham số điều chỉnh của tầng đầu ra: K

Lan truyền ngược: Pha tiến

$$a_h^t = \sum_{i=1}^{I} w_{ih} x_i^t + \sum_{h'=1}^{H} w_{h'h} b_{h'}^{t-1}$$

trong đó:

- x^t_i là tín hiệu đầu vào tại nơ-ron *i* tại thời điểm *t*
- w_i^h là trọng số liên kết của nơ-ron *i* của tầng đầu vào và nơ-ron *h* của tầng ẩn
- $w_{h'h}$ là trọng số liên kết của nơ-ron *h'* và nơ-ron *h* của tầng ẩn
- $b_{h'}^{t-1}$ là giá trị đầu ra của nơ-ron *h'* tại thời điểm *(t-1)*
- α^t_h là đầu vào của nơ-ron *h* của tầng ẩn tại thời điểm *t*

Pha tiến (tiếp)

$$b_h^t = \theta_h(a_h^t)$$

trong đó:

- b^t_h là giá trị đầu ra của nơ-ron h của tầng ẩn tại thời điểm t
- θ_h là hàm kích hoạt tại nơ-ron h của tầng ẩn
- α^t_h là đầu vào của nơ-ron h của tầng ẩn tại thời điểm t

Pha tiến (tiếp)

$$a_k^t = \sum_{h=1}^{H} w_{hk} b_h^t$$

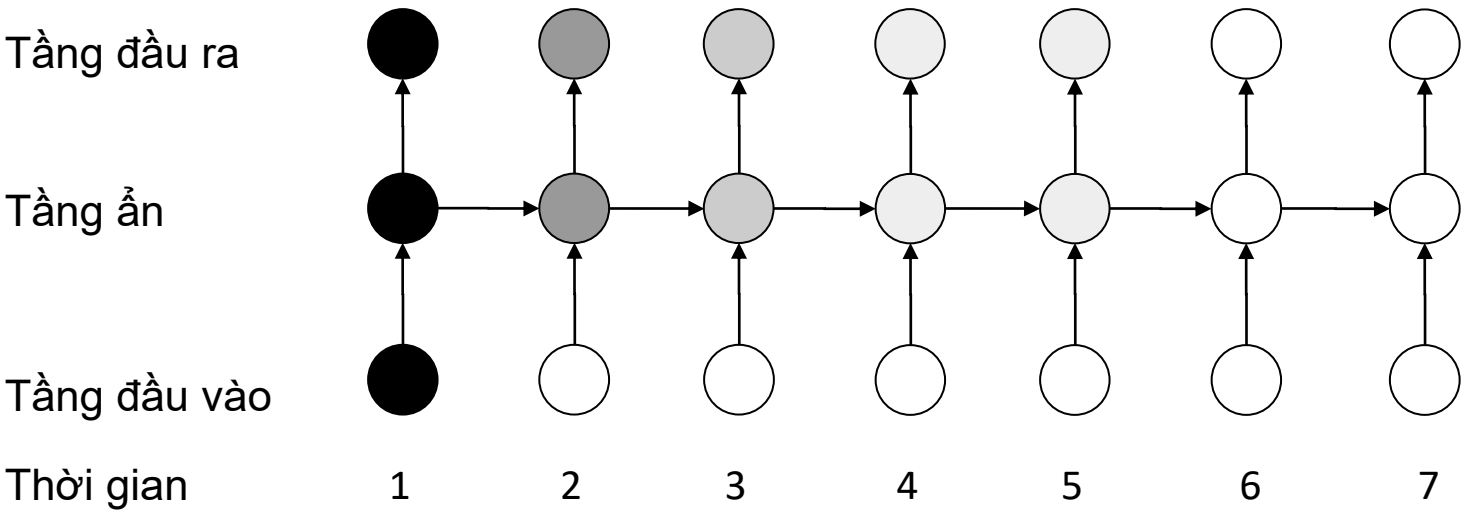
trong đó:

- α^t_k là giá trị đầu vào của nơ-ron k của tầng đầu ra tại thời điểm t
- w_{hk} là trọng số liên kết giữa nơ-ron h của tầng ẩn và nơ-ron k của tầng đầu ra
- b^t_h là giá trị đầu ra của nơ-ron h của tầng ẩn tại thời điểm t

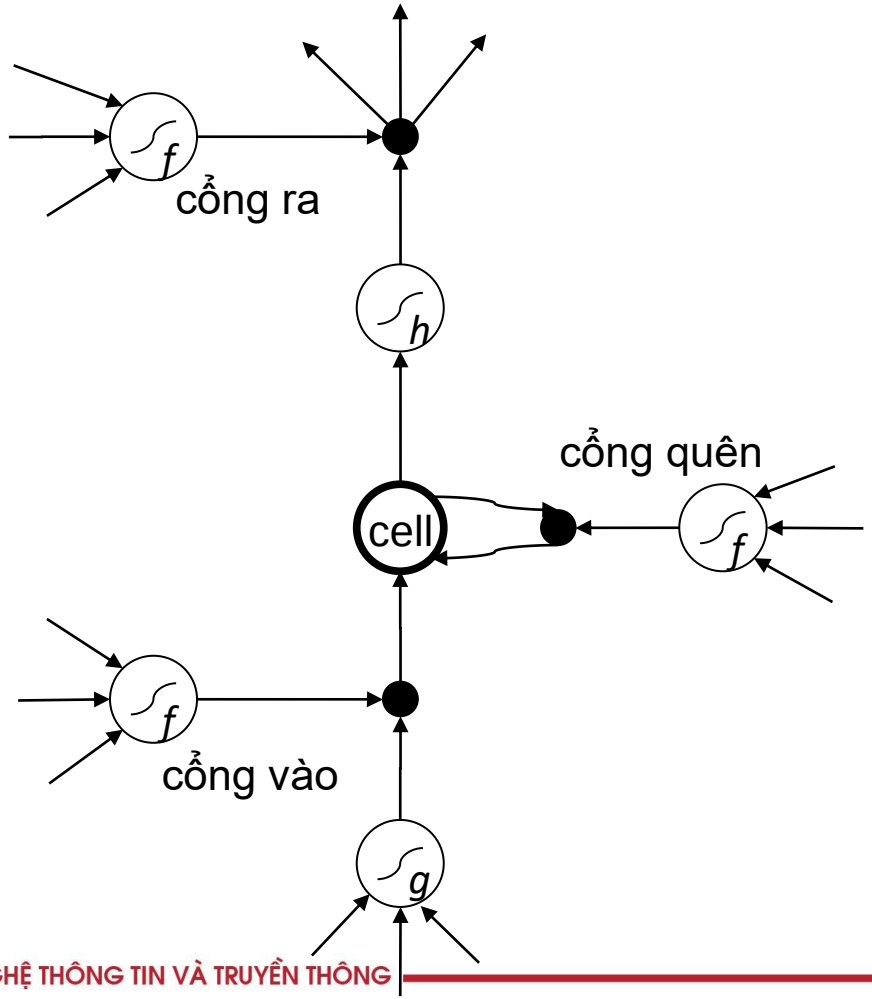
Gradient triệt tiêu

- Trong mạng nơ-ron nhiều tầng, các tầng đầu có "tốc độ học" chậm hơn các tầng cuối
- Gradient bị triệt tiêu khi lan truyền ngược từ tầng cuối về tầng đầu
- Điều này xảy ra với RNN theo chiều thời gian
- Một cách giải quyết vấn đề triệt tiêu gradient là dùng cell LSTM

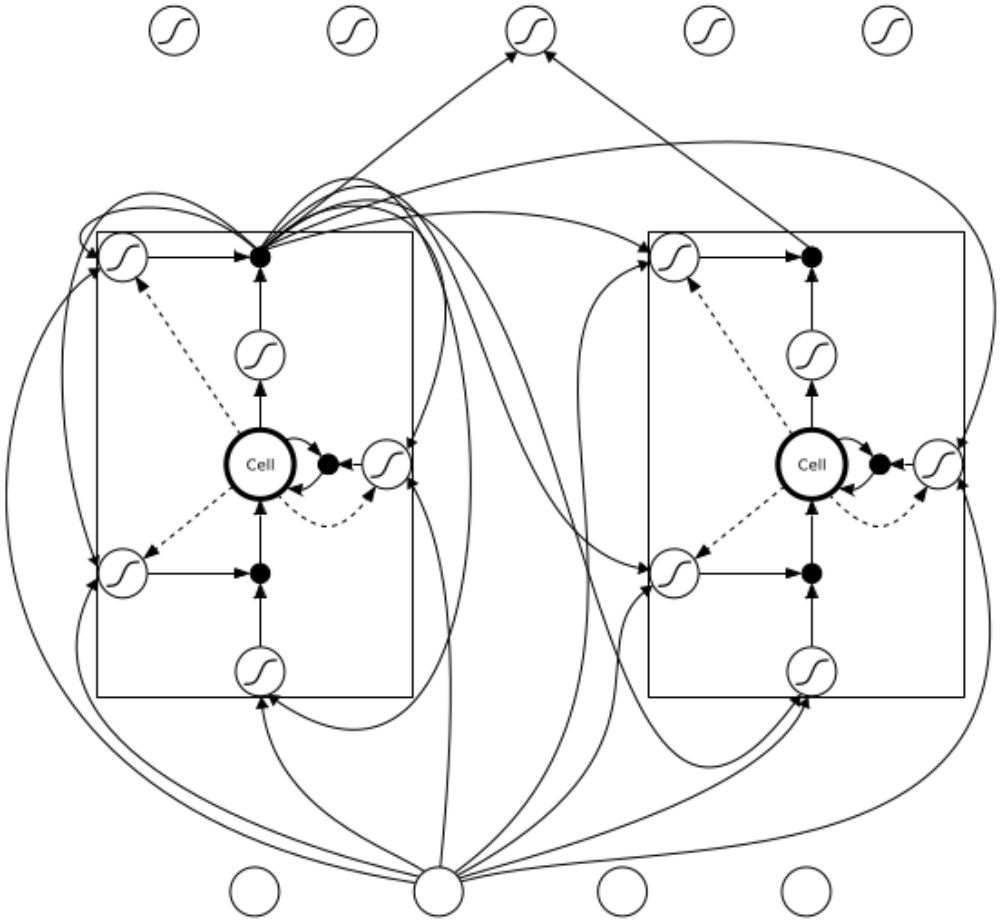
Gradient triệt tiêu trong RNN



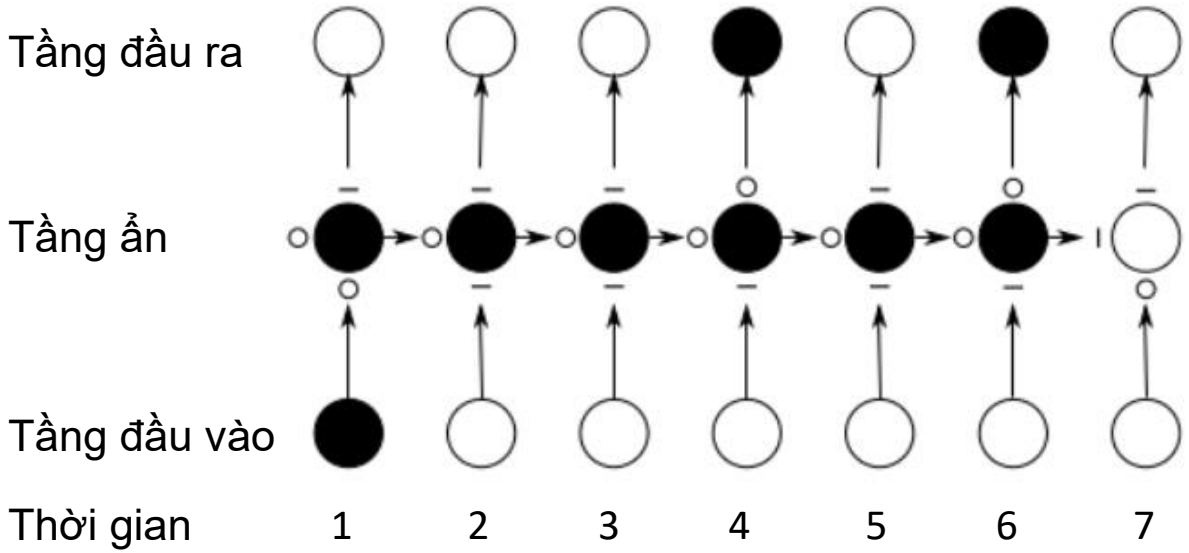
Long-short Term Memory (LSTM)



LSTM (tiếp)



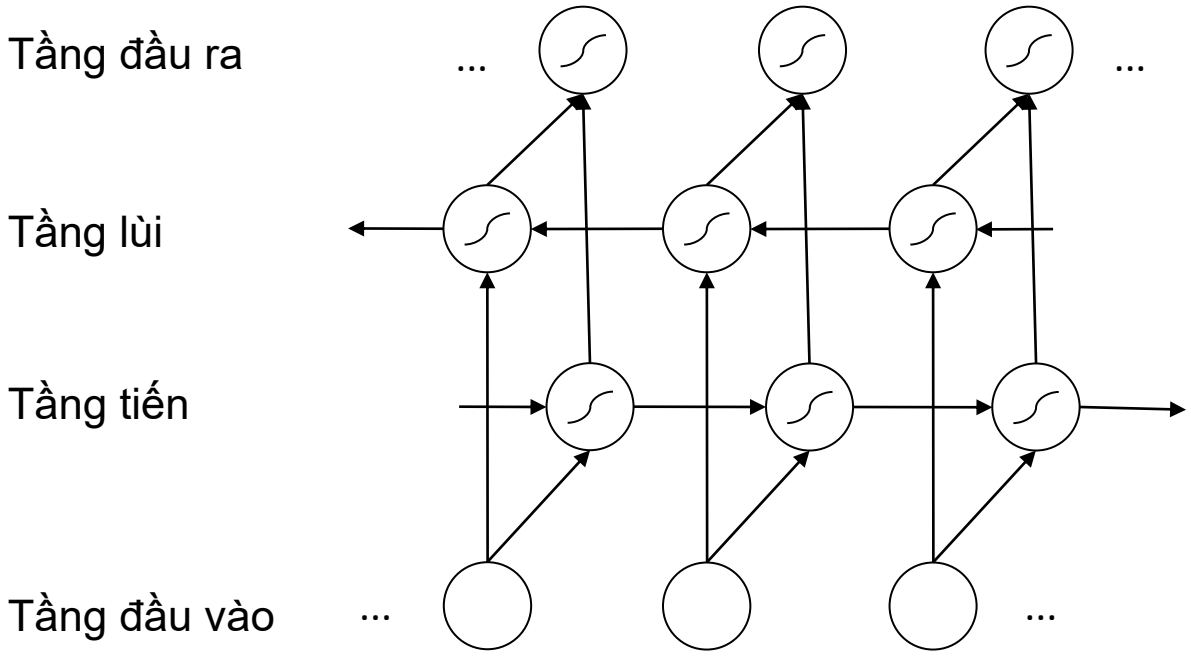
LSTM (tiếp)



Số lượng tham số của LSTM

- I: Số nơ-ron của tầng đầu vào
- H: Số cell LSTM (số nơ-ron của tầng ẩn)
- K: Số nơ-ron của tầng đầu ra
- Số tham số: 4 x (I x H + H x H + H) + H x K + K
  - Số liên kết giữa tầng đầu vào và tầng ẩn: I x H
  - Số liên kết hồi quy của tầng ẩn: H x H
  - Số liên kết của tầng ẩn và tầng đầu ra: H x K
  - Số tham số điều chỉnh của tầng ẩn: H
  - Số tham số điều chỉnh của tầng đầu ra: K

RNN hai chiều



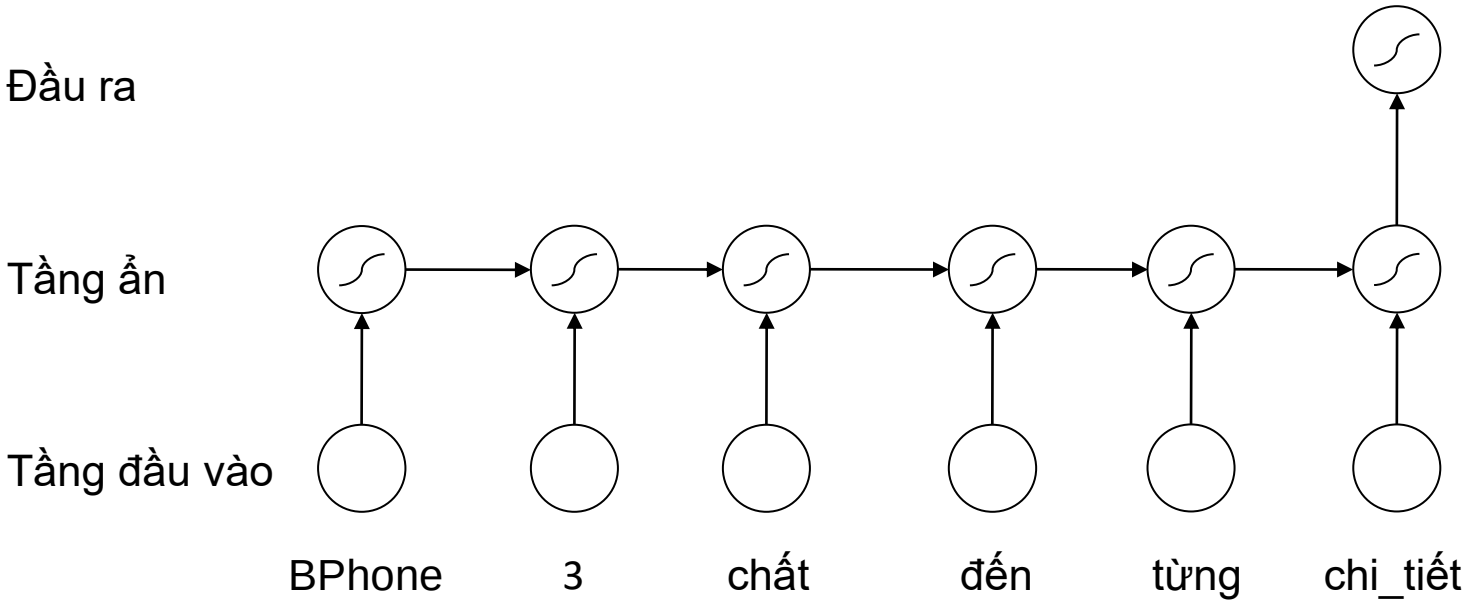
RNN cho phân loại cảm xúc

- Bài toán phân loại:
  - Đầu vào là văn bản trong đó $\boldsymbol{x}_t$ là từ thứ t trong văn bản đầu vào
  - Đầu ra là cảm xúc của văn bản (tích cực, trung tính, tiêu cực)
- Sử dụng đầu ra của từ cuối cùng $\boldsymbol{x}_T$ để phân loại: $\boldsymbol{x}_T$ được coi là biểu diễn của cả văn bản (!)

RNN cho phân loại cảm xúc (tiếp)

- Tầng đầu vào có V nơ-ron trong đó V là kích thước của từ điển
- Biểu diễn *one-hot*: Mỗi từ $\boldsymbol{x}_t$ ứng với một từ t_i trong từ điển tương ứng với nơ-ron thứ i có giá trị bằng 1, các nơ-ron khác có giá trị bằng 0
- Tầng đầu ra có 3 nơ-ron biểu diễn 3 lớp cảm xúc

RNN cho phân loại cảm xúc (tiếp)



10. Kết hợp các bộ phân loại
10.1 Bagging

- (Boostrap Aggregating)
- Cho một tập DL huấn luyện D gồm n ví dụ và một giải thuật học cơ sở
- Huấn luyện:

1. Tạo ra k mẫu $S_1, S_2, \ldots, S_k$ bằng cách lấy mẫu có thay thế từ n ví dụ trong D (một ví dụ có thể không xuất hiện hoặc xuất hiện nhiều lần trong một mẫu). Trung bình, S_i chứa 63.2% ví dụ trong D
2. Xây dựng một bộ phân loại cho mỗi mẫu S_i. Giải thuật học cơ sở được sử dụng cho cả k bộ phân loại

Baggin (tiếp)

- Kiểm tra: Phân loại mỗi ví dụ kiểm thử bởi *k* bộ phân loại và sử dụng cơ chế bầu cử với trọng số cân bằng để chọn ra lớp tốt nhất
- Bagging cho kết quả cải thiện với các giải thuật học **không ổn định** (cây quyết định)
- Tuy nhiên, bagging có thể làm giảm kết quả với các giải thuật **ổn định** (Naive Bayes, kNN)

10.2 Boosting

- Huấn luyện:
  - Boosting tạo ra một chuỗi các bộ phân loại (cái sau phụ thuộc vào cái trước)
  - Sử dụng cùng một giải thuật học cơ sở
  - Các ví dụ phân loại sai của bộ phân loại trước được tập trung bằng cách tăng trọng số
- Kiểm tra: Với mỗi ví dụ kiểm thử, kết quả của k bộ phân loại ở mỗi bước được tổng hợp lại để cho ra kết quả cuối cùng
- Boosting hiệu quả hơn với các giải thuật học không ổn định

Algorithm AdaBoost(D, Y, BaseLeaner, k)

1 Khởi tạo $D_1(w_i) \leftarrow 1/n$ với mọi i; // khởi tạo trọng số

2 **for** t = 1 to k **do**

3 $f_t \leftarrow$ BaseLearner(D_t); // xây dựng bộ phân loại mới f_t

$$e_t \leftarrow \sum_{i:f_t(D_t(\mathbf{x}_i)) \neq y_i} D_t(w_i);$$

4 // tính lỗi của f_t

5 **if** $e_t > 1/2$ **then** // nếu lỗi quá lớn

6 $k \leftarrow k - 1$; // bỏ qua lần lặp

7 exit-loop; // và thoát

8 **else**

9 $\alpha_t \leftarrow e_t / (1 - e_t)$;

$$\alpha_t \quad \text{if } f_t(D_t(\boldsymbol{x}_i)) = y_i$$

10 $D_{t+1}(w_i) \leftarrow D_t(w_i) \times$ 0 nếu không // cập nhật trọng số

$$D_{t+1}(w_i) \leftarrow \frac{D_{t+1}(w_i)}{\sum_{i=1}^{n} D_{t+1}(w_i)}$$

11 // hiệu chỉnh trọng số

12 **endif**

13 $$f_{final}(\mathbf{x}) \leftarrow \underset{y \in Y}{\operatorname{argmax}} \sum_{t:f_t(\mathbf{x})=y} \log \frac{1}{\beta_t}$$

14 // bộ phân loại cuối cùng

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict