



VietAI

Neural Network



VietAI teaching team

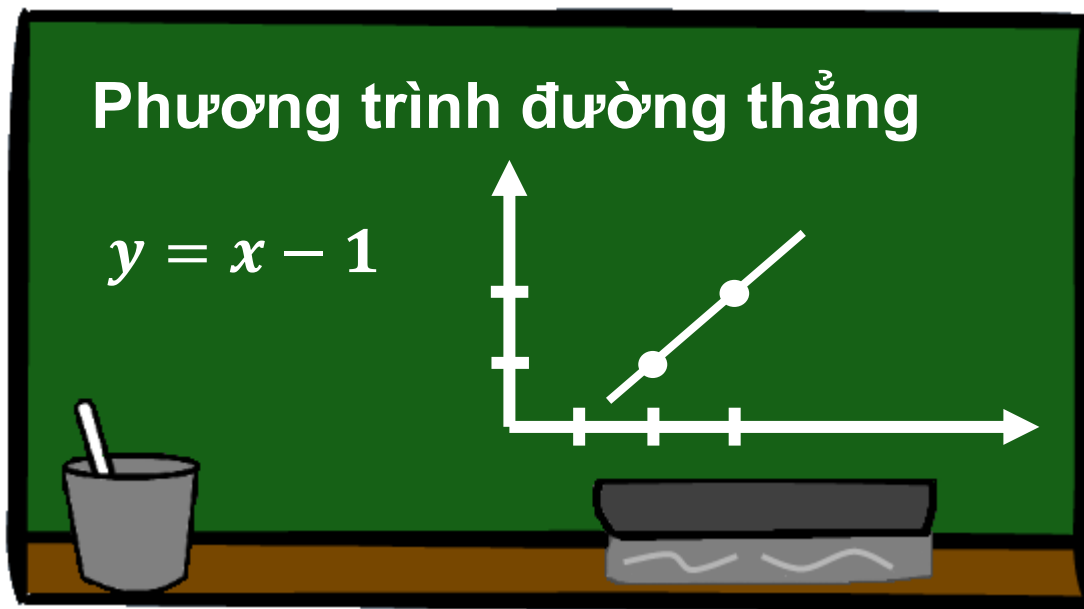
Nội dung

1. Giới thiệu bài toán
2. Giới thiệu Mạng neuron
3. Kiến trúc Mạng neuron
4. Lan truyền thuận
5. Các hàm kích hoạt (activation functions)
6. Lan truyền ngược
7. Bài tập ví dụ
8. Một số vấn đề khi huấn luyện neural network

Nội dung

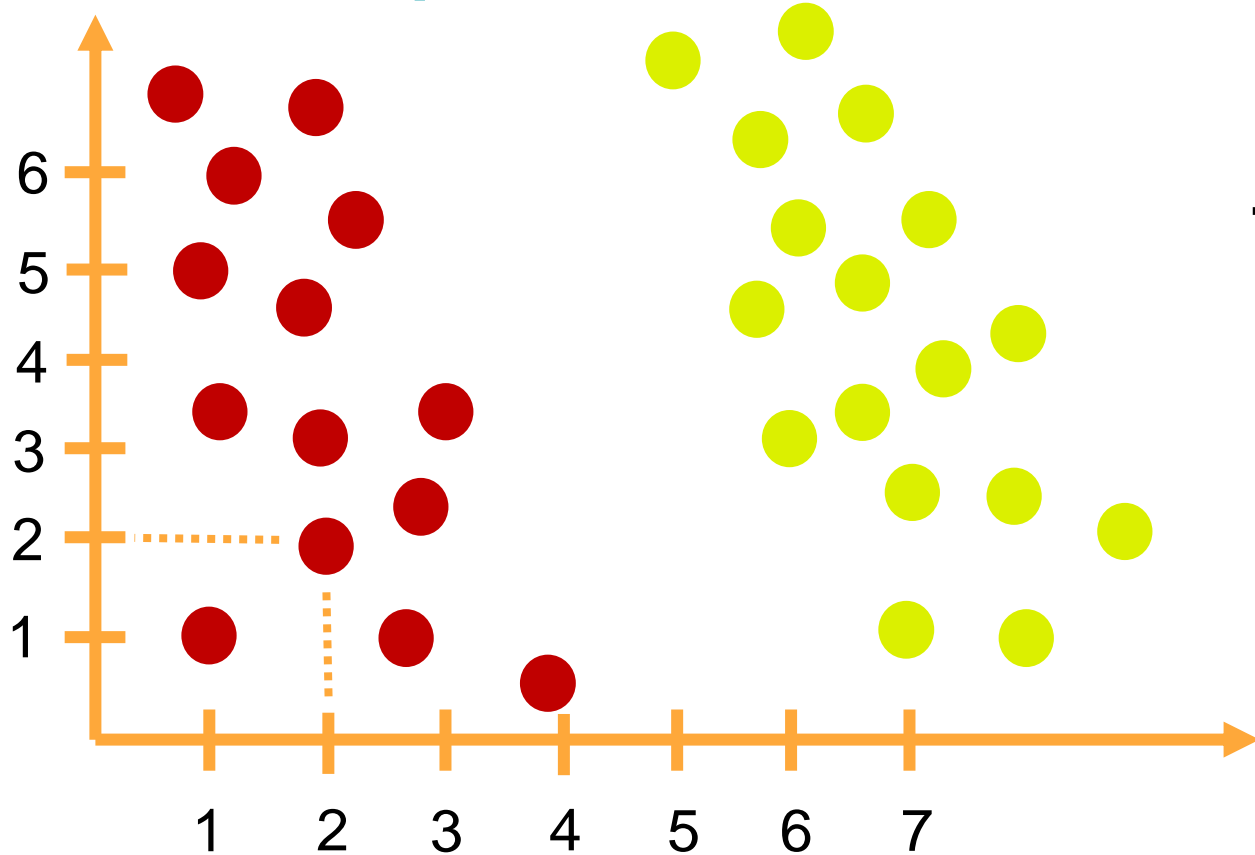
1. **Giới thiệu bài toán**
2. Giới thiệu Mạng neuron
3. Kiến trúc Mạng neuron
4. Lan truyền thuận
5. Các hàm kích hoạt (activation functions)
6. Lan truyền ngược
7. Bài tập ví dụ
8. Một số vấn đề khi huấn luyện neural network

1 Giới thiệu bài toán



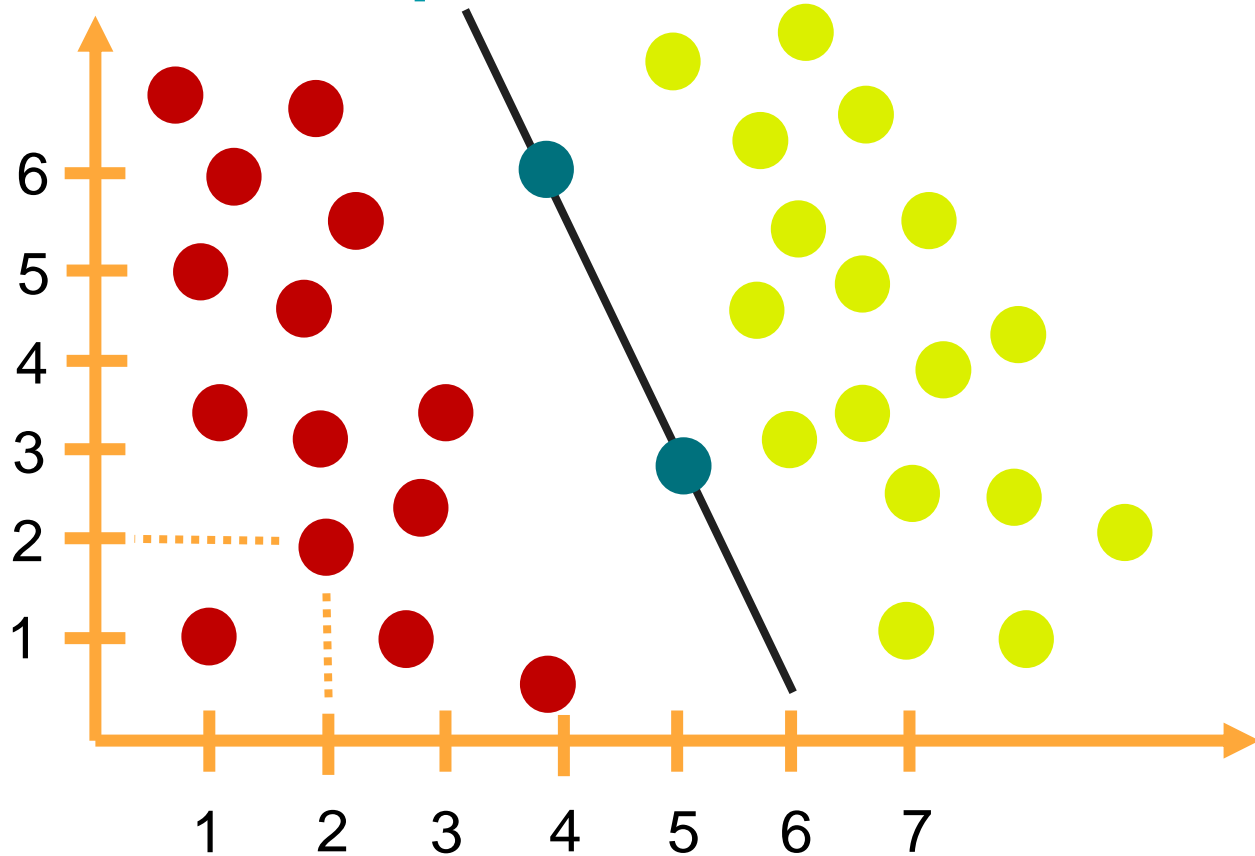
Bảng giá trị:

1 Giới thiệu bài toán



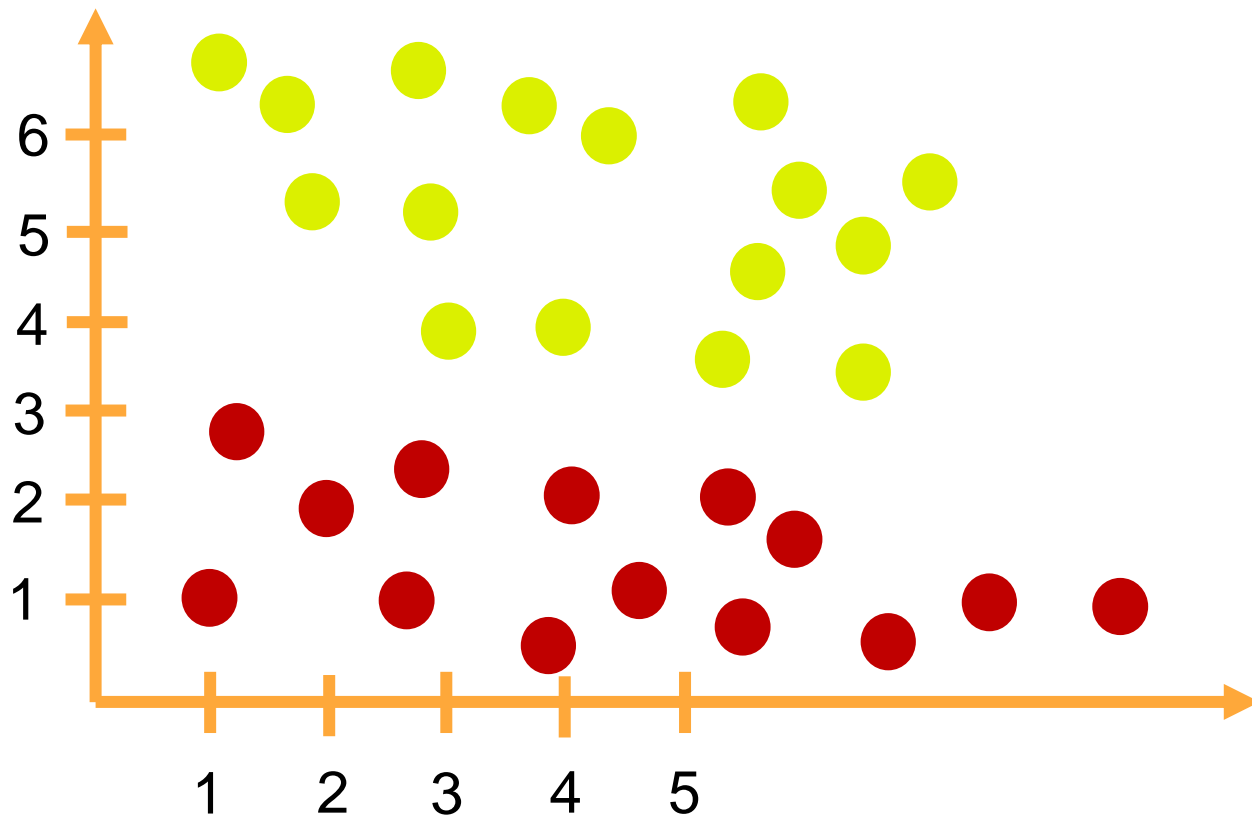
**Tìm đường thẳng
phân cách
điểm đỏ
và điểm xanh**

1 Giới thiệu bài toán

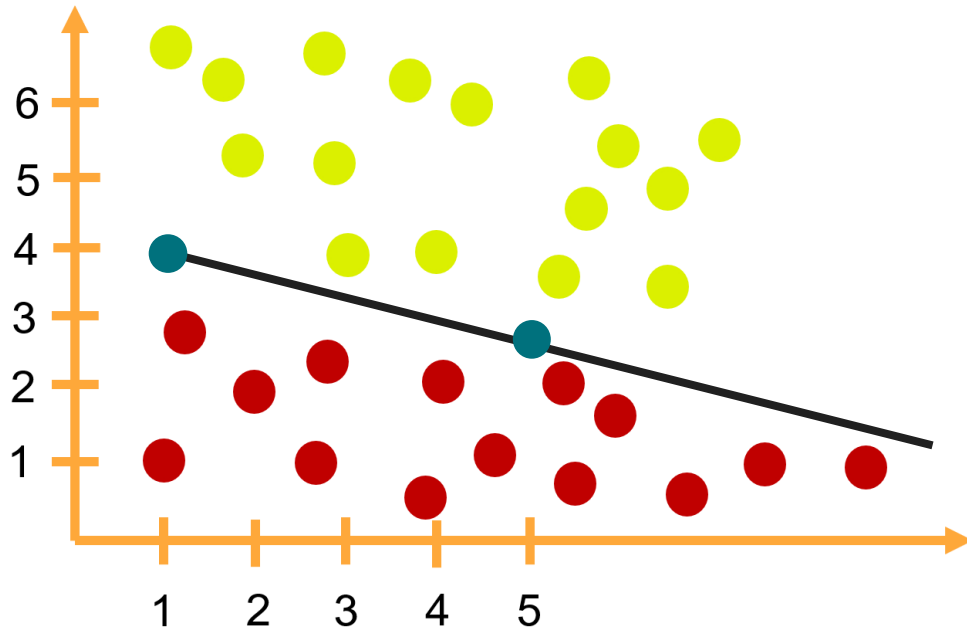


$$y = -3x + 18$$

1 Giới thiệu bài toán



1 Giới thiệu bài toán



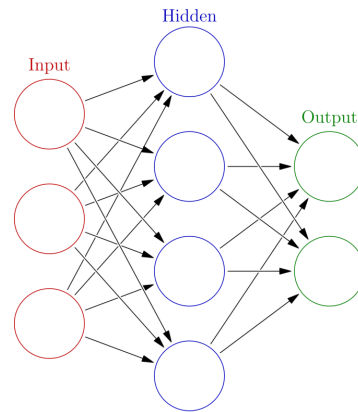
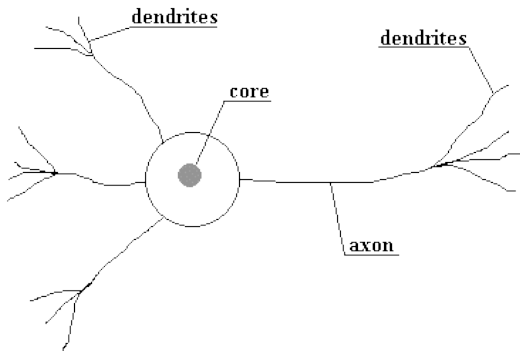
$$y = -\frac{1}{4}x + \frac{17}{4}$$

Nội dung

1. Giới thiệu bài toán
2. Giới thiệu
3. Kiến trúc Mạng neuron
4. Lan truyền thuận (dự đoán kết quả)
5. Các hàm kích hoạt (activation functions)
6. Lan truyền ngược (thuật toán học gradient descent)
7. Bài tập ví dụ
8. Một số vấn đề khi huấn luyện neural network

2 Neural Network

- Mô hình cơ bản trong machine learning/deep learning
- Tên gọi khác:
 - Mạng neuron truyền thẳng (feedforward neural network - FNN)
 - Mạng neuron nhân tạo
- Lấy ý tưởng từ cách xử lý thông tin của não người
- Được áp dụng trên nhiều bài toán khác nhau



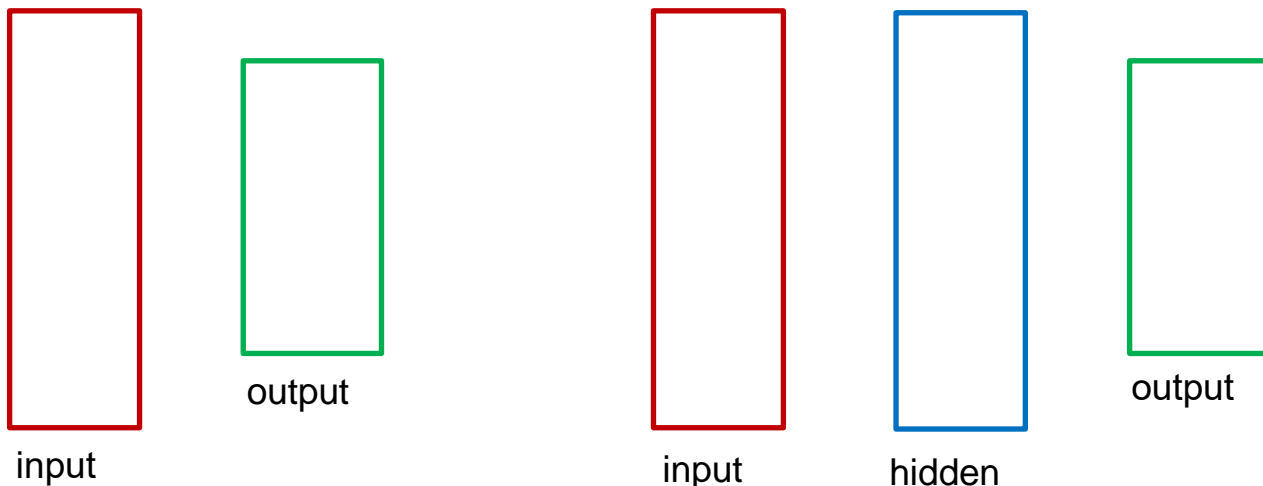
3 Kiến trúc

- Khái niệm:
 - Lớp (layer)
 - Nơ-ron (neuron)
 - Trọng số (weight)
 - Đơn vị điều chỉnh (bias)
- Tính chất:
 - Kết nối đủ (fully connected)
 - Số lượng hidden layer
 - Số lượng neuron mỗi hidden layer

- Khái niệm:
 - Lớp (layer)
 - Nơ-ron (neuron)
 - Trọng số (weight)
 - Đơn vị điều chỉnh (bias)
- Tính chất:
 - Kết nối đủ (fully connected)
 - Số lượng hidden layer
 - Số lượng neuron mỗi hidden layer

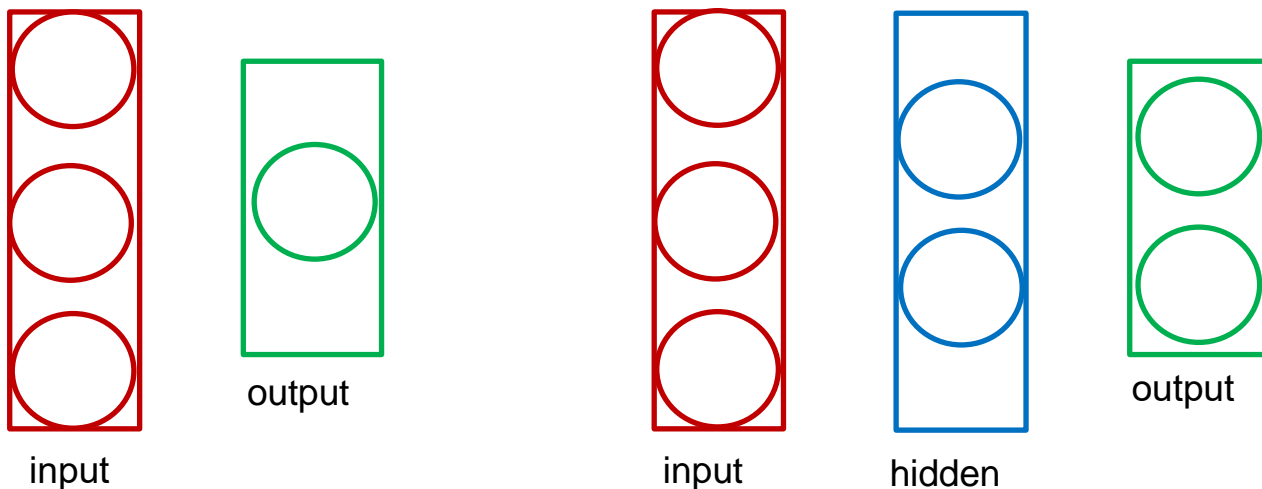
3 Lớp (layer)

- Lớp (layer) là tập hợp chứa các neuron theo chiều dọc
- Mỗi neural network có 3 layer chính:
 - Lớp đầu vào (input layer)
 - Lớp ẩn (hidden layer) → (optional)
 - Lớp đầu ra (output layer)



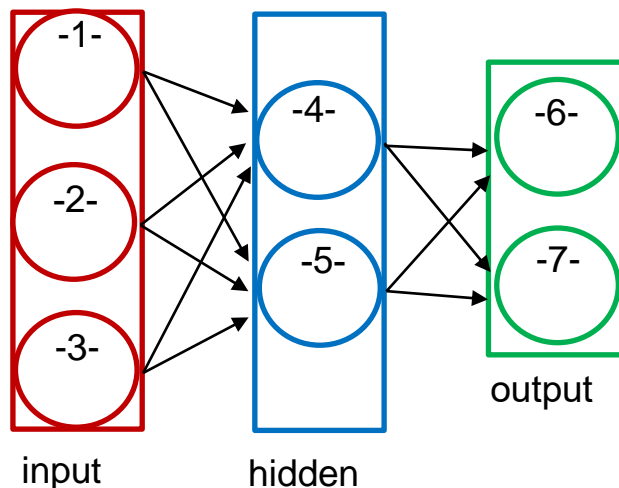
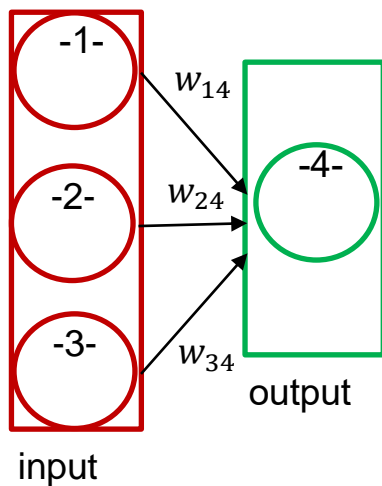
3 Nơ-ron (neuron)

- Nơ-ron (neuron) là đơn vị chứa kết quả tính toán tại một thời điểm
- Mỗi neuron chứa một giá trị số thực



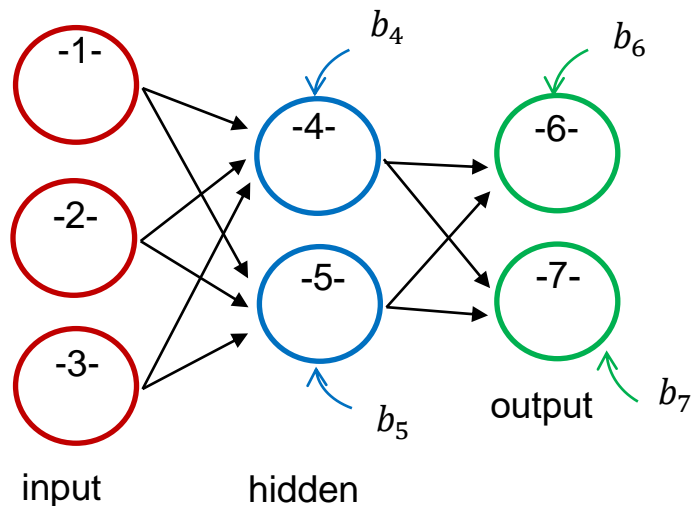
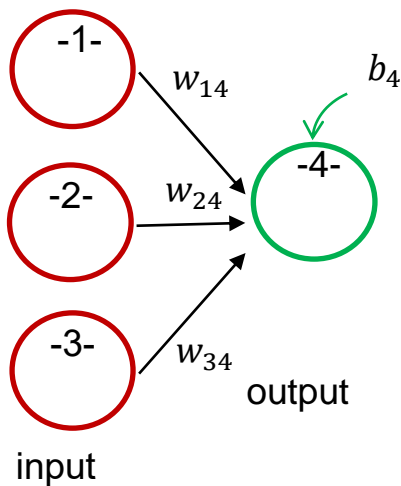
3 Trọng số (weight)

- Trọng số (weight) là giá trị nằm trên mỗi cạnh nối các neuron với nhau
- Mỗi giá trị trọng số là một số thực
- Giá trị trọng số được xem là “bộ nhớ dài hạn” và được sử dụng tại nhiều thời điểm



3 Đơn vị điều chỉnh (bias)

- Đơn vị điều chỉnh (bias) là giá trị đi theo mỗi neuron ở tất cả các layer, ngoại trừ input layer
- Mỗi giá trị bias là một số thực
- Giá trị bias giúp cho mô hình phân lớp “linh hoạt” hơn



- Khái niệm:
 - Lớp (layer)
 - Nơ-ron (neuron)
 - Trọng số (weight)
 - Đơn vị điều chỉnh (bias)
- Tính chất:
 - Kết nối đủ (fully connected)
 - Số lượng hidden layer
 - Số lượng neuron mỗi hidden layer

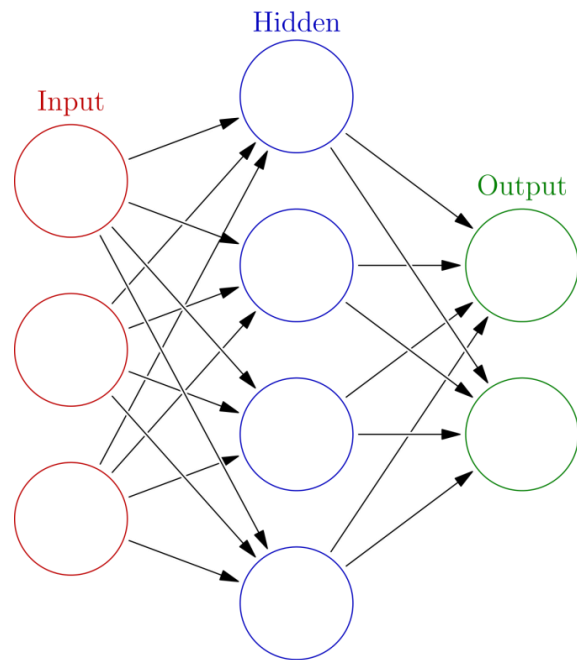
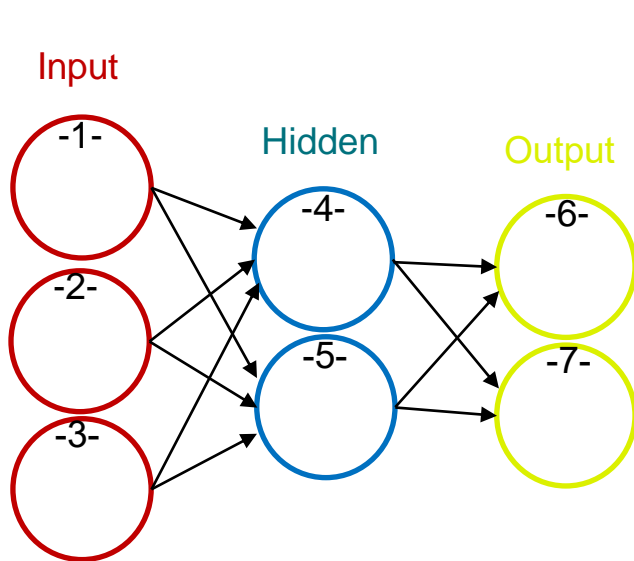
} parameters

3 Kiến trúc

- Khái niệm:
 - Lớp (layer)
 - Nơ-ron (neuron)
 - Trọng số (weight)
 - Đơn vị điều chỉnh (bias) } parameters
- Tính chất:
 - Kết nối đủ (fully connected)
 - Số lượng hidden layer
 - Số lượng neuron mỗi hidden layer

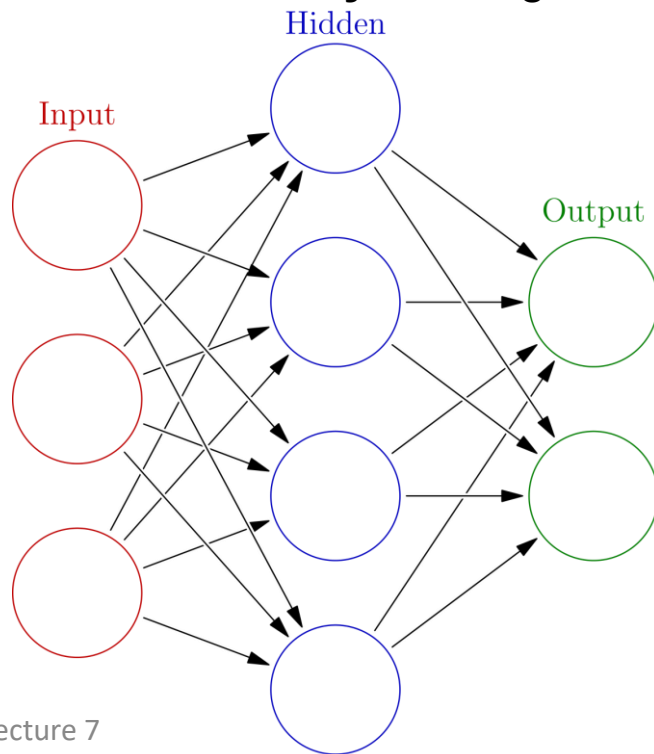
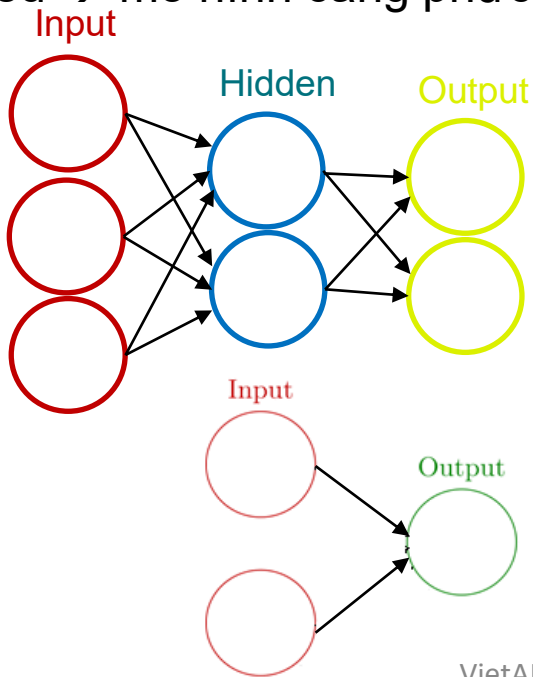
3 Tính chất

- **Kết nối đủ (fully connected):** Ngoài lớp output, mỗi neuron tại lớp thứ k sẽ được kết nối với **tất cả** các neuron ở lớp thứ $k + 1$



3 Tính chất

- **Số lượng neuron ở input layer và output layer:** tùy thuộc vào bài toán
- **Số lượng hidden layer và số lượng neuron mỗi hidden layer:** càng nhiều → mô hình càng phức tạp



3 Kiến trúc

- Khái niệm:
 - Lớp (layer)
 - Nơ-ron (neuron)
 - Trọng số (weight)
 - Đơn vị điều chỉnh (bias)

} parameters
- Tính chất:
 - Kết nối đủ (fully connected)
 - Số lượng hidden layer
 - Số lượng neuron mỗi hidden layer

} hyper-parameters

3 Kiến trúc

Bài tập 1: Hãy vẽ mạng neuron (bao gồm đầy đủ các ký hiệu **parameters**) với các thông số (**hyper parameters**) sau:

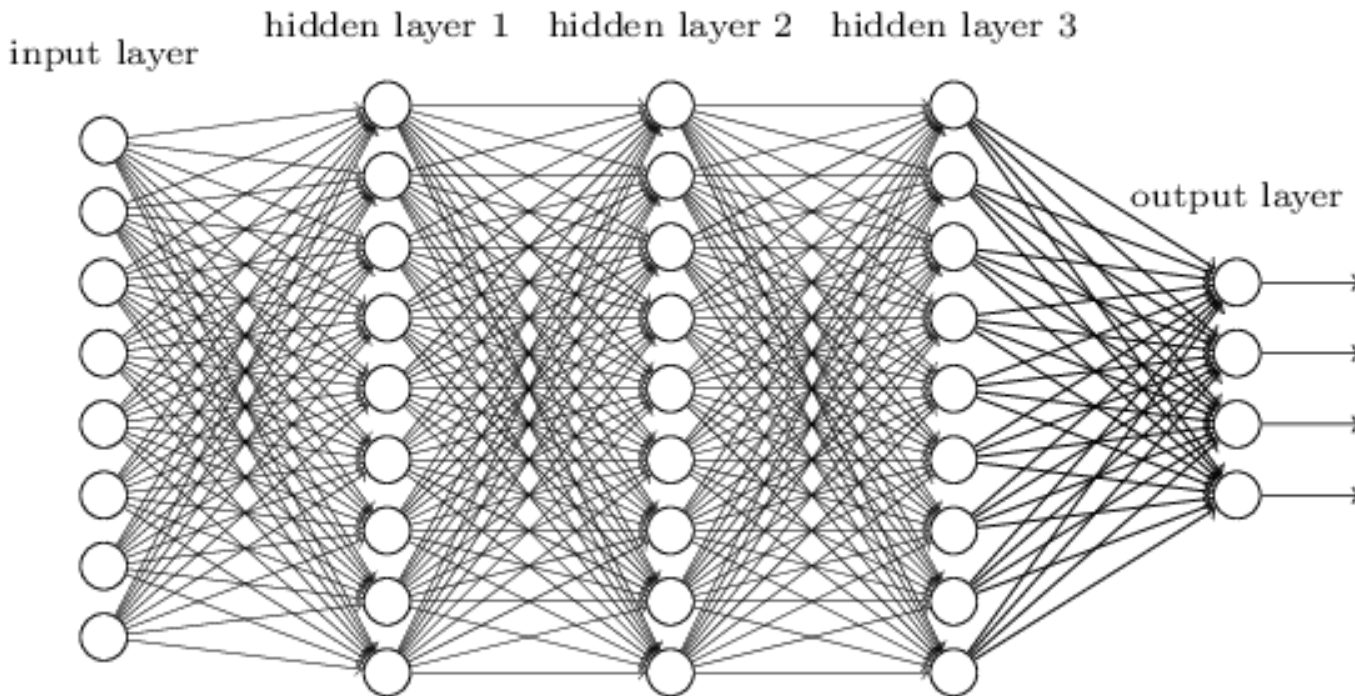
- Input layer: gồm 4 neuron
- Hidden layer: có 2 hidden layer
 - Layer đầu tiên có 3 neuron
 - Layer thứ hai có 2 neuron
- Output layer: có 2 neuron



Giải:

3 Kiến trúc

Bài tập 2: Hãy cho biết hyper parameters của mạng neuron sau:

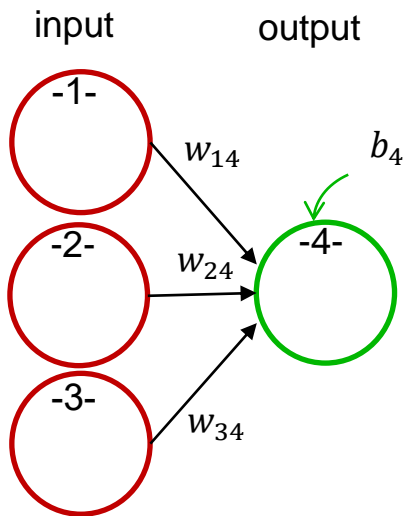


Nội dung

1. Giới thiệu bài toán
2. Giới thiệu Mạng neuron (Neural Network)
3. Kiến trúc Mạng neuron
4. **Lan truyền thuận (dự đoán kết quả)**
5. Các hàm kích hoạt (activation functions)
6. Lan truyền ngược (thuật toán học gradient descent)
7. Bài tập ví dụ
8. Một số vấn đề khi huấn luyện neural network

4 Lan truyền thuận

- Ví dụ:** Xét mạng neuron có cấu trúc và parameters sau



- Giá trị đầu vào:

$$x_1 = 5$$

$$x_2 = 3$$

$$x_3 = 2$$

- Quá trình tính output:

$$\hat{y}_4 = x_1 \times w_{14} + x_2 \times w_{24} + x_3 \times w_{34} + b_4$$

$$= 5 \times 0.1 + 3 \times 0.6 + 2 \times 0.5 + 0.8$$

$$= 0.5 + 1.8 + 1 + 0.8$$

$$= 4.1$$

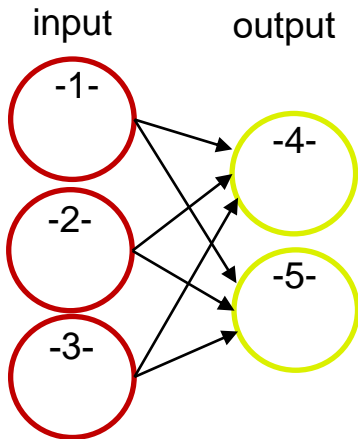
$$w_{14} = 0.1$$

$$w_{24} = 0.6 \quad b_4 = 0.8$$

$$w_{34} = 0.5$$

4 Bài tập: Lan truyền thuận 1

- Bài tập:** Xét mạng neuron có cấu trúc và parameters sau



- Giá trị đầu vào:

$$x_1 = 1$$

$$x_2 = 8$$

$$x_3 = -6$$

- Quá trình tính output:

$$\hat{y}_4 =$$

$$\hat{y}_5 =$$

$$w_{14} = -0.71$$

$$w_{24} = 0.23$$

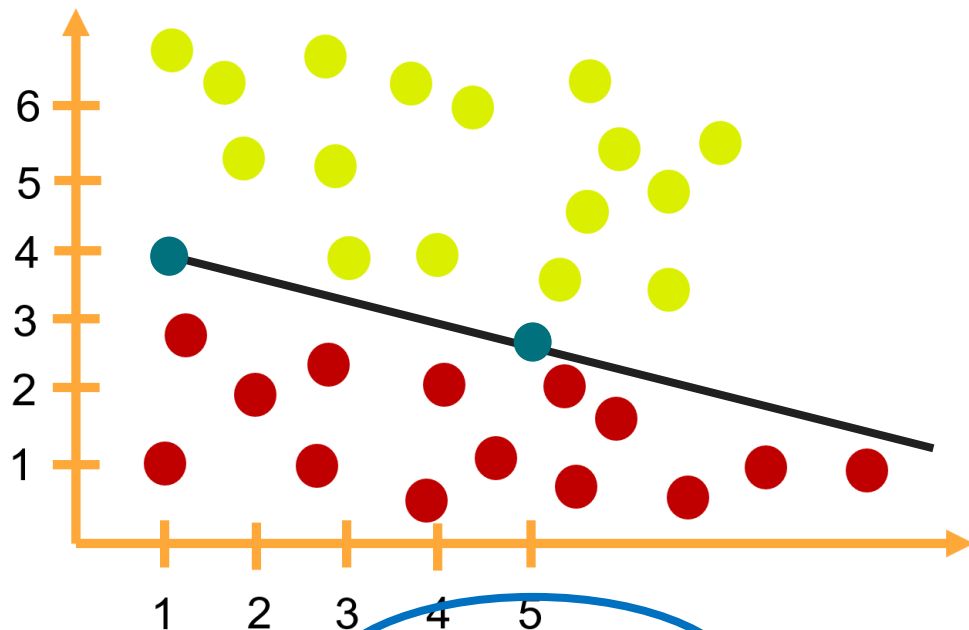
$$w_{34} = 0.14 \quad b_4 = 0.56$$

$$w_{15} = 0.42 \quad b_5 = -0.15$$

$$w_{25} = 0.30$$

$$w_{35} = -0.10$$

4 Bài toán cũ

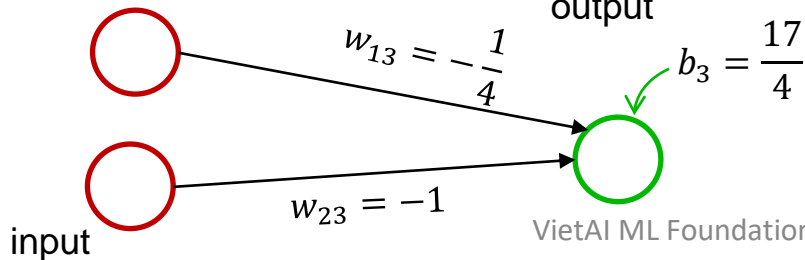
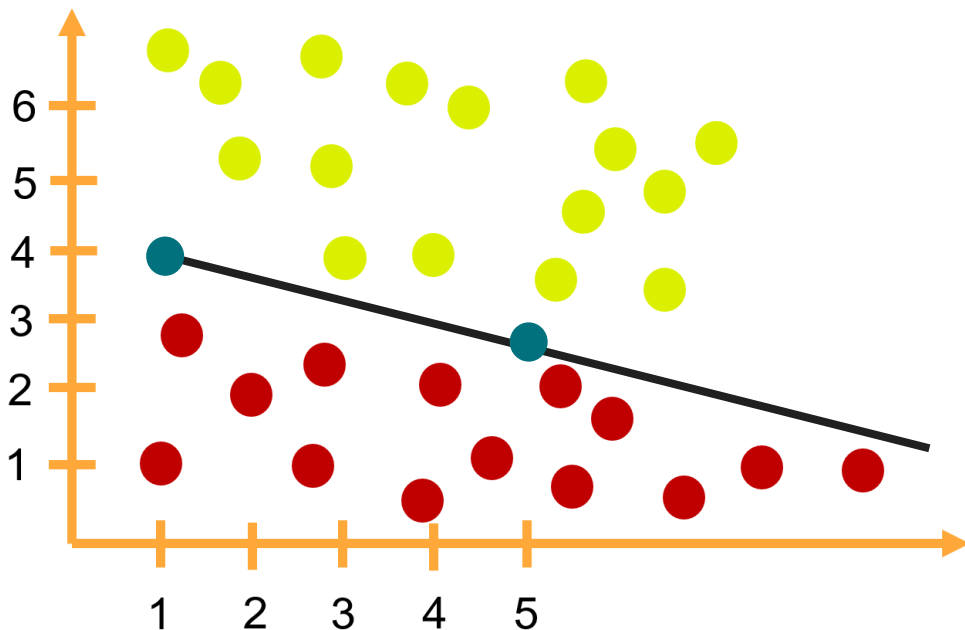


$$y = -\frac{1}{4}x + \frac{17}{4}$$

Điểm(2; 2) $-\frac{1}{4}x - y + \frac{17}{4} = -\frac{1}{4} \cdot 2 - 2 + \frac{17}{4} = \frac{7}{4} > 0 \rightarrow \text{Màu đỏ}$

Điểm (4; 4): $-\frac{1}{4}x - y + \frac{17}{4} = -\frac{1}{4} \cdot 4 - 4 + \frac{17}{4} = \frac{-3}{4} < 0 \rightarrow \text{Màu xanh}$

4 Bài toán cũ



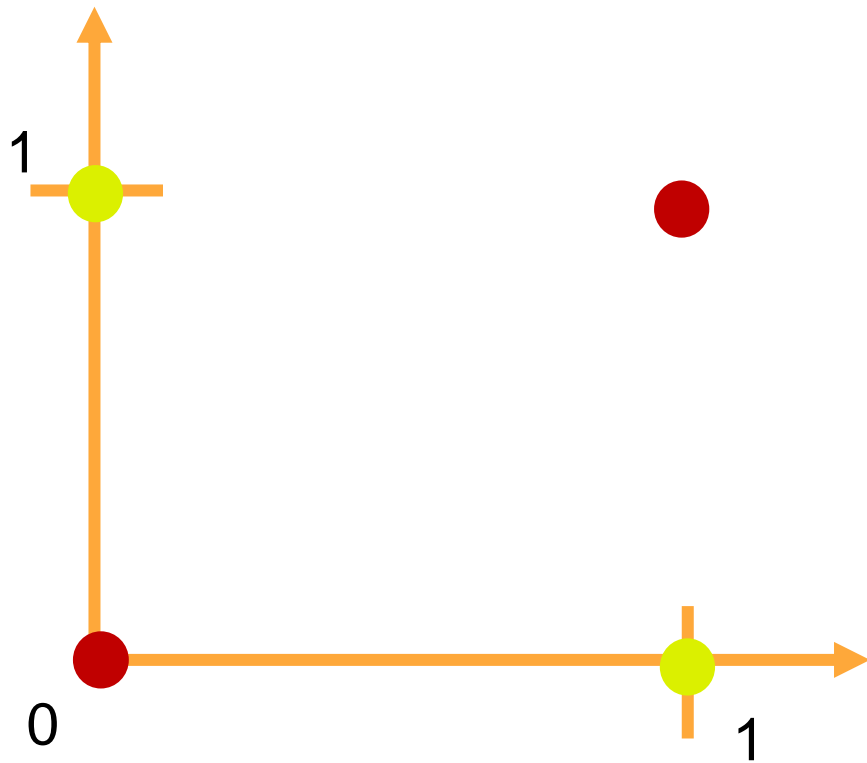
Giá trị input

$$-\frac{1}{4}x - y + \frac{17}{4}$$

$$\equiv -\frac{1}{4}x_1 - x_2 + \frac{17}{4}$$

Phương trình này
tương đương với
neural net sau:

4 Bất khả phân tuyến tính



Bài “khó”

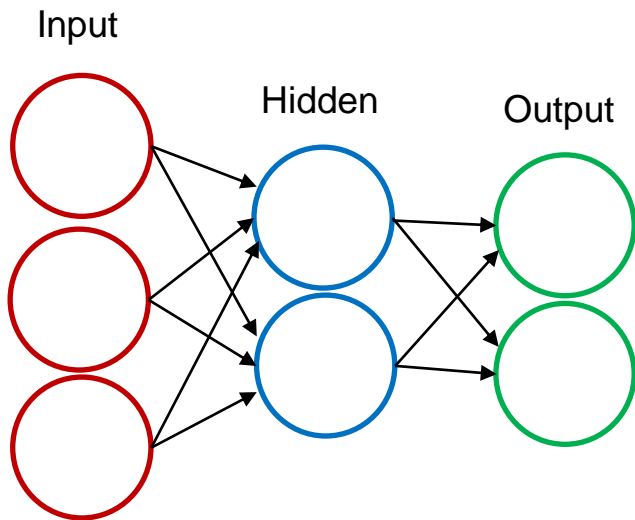
A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

Nội dung

1. Giới thiệu bài toán
2. Giới thiệu Mạng neuron (Neural Network)
3. Kiến trúc Mạng neuron
4. Lan truyền thuận (dự đoán kết quả)
5. **Các hàm kích hoạt (activation functions)**
6. Lan truyền ngược (thuật toán học gradient descent)
7. Bài tập ví dụ
8. Một số vấn đề khi huấn luyện neural network

5 Giới thiệu hàm kích hoạt

- Xét neural net sau



Activation function

Vẫn là mô hình phân lớp tuyến tính dù thêm nhiều hidden layer

$$\hat{y} = \left(\sum x \cdot w \right) + bias$$

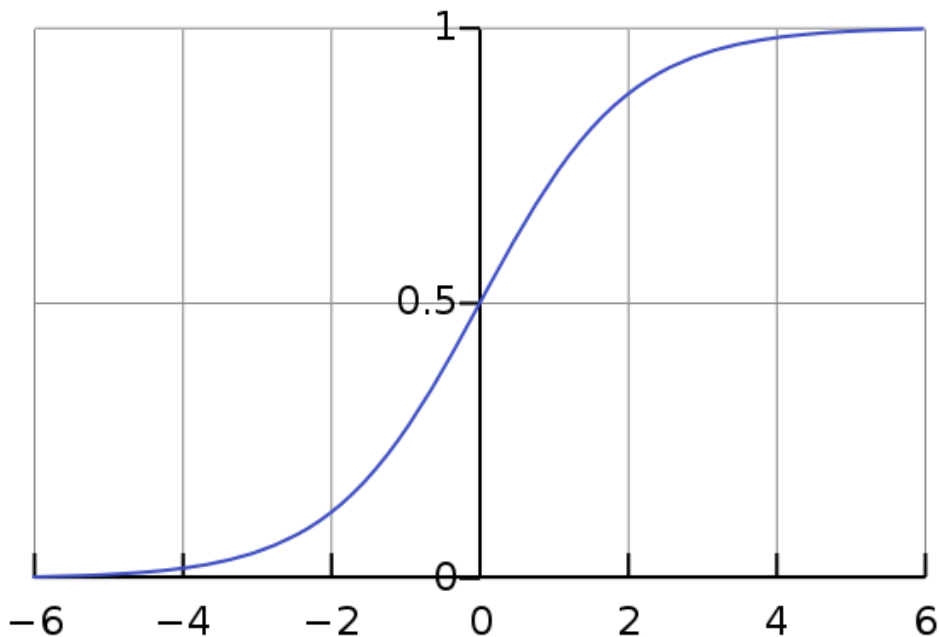
- $\hat{y} \in (-\infty; +\infty)$
- Không thể biết neuron nào cần “activate”

5

Hàm kích hoạt

Hàm sigmoid:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$



Hàm kích hoạt

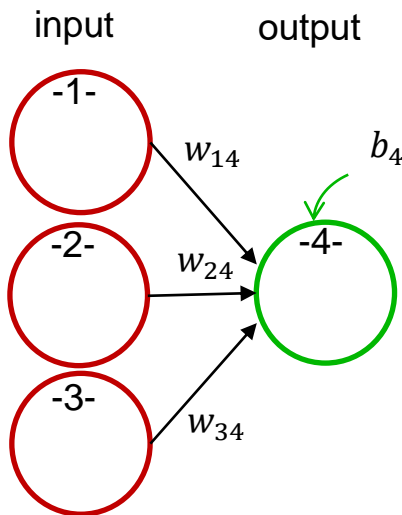
- Kết quả mỗi phép tính sẽ được đưa vào hàm activation, trước khi được dùng để tính tiếp ở các layer tiếp theo

~~$$\hat{y} = \left(\sum x \cdot w \right) + bias$$~~

$$\hat{y} = \text{sigmoid} \left(\left(\sum x \cdot w \right) + bias \right)$$

4 Lan truyền thuận

- Ví dụ:** Xét mạng neuron có cấu trúc và parameters sau



- Giá trị đầu vào:

$$x_1 = 5$$

$$x_2 = 3$$

$$x_3 = 2$$

- Quá trình tính output:

$$\text{tạm } \hat{y}_4 = x_1 \times w_{14} + x_2 \times w_{24} + x_3 \times w_{34} + b_4$$

$$= 5 \times 0.1 + 3 \times 0.6 + 2 \times 0.5 + 0.8$$

$$= 0.5 + 1.8 + 1 + 0.8$$

$$= 4.1$$

$$\hat{y}_4 = \text{sigmoid}(\text{tạm}) = \frac{1}{1 + e^{-4.1}} = 0.983$$

$$w_{14} = 0.1$$

$$w_{24} = 0.6 \quad b_4 = 0.8$$

$$w_{34} = 0.5$$

5 Công thức lan truyền thuận

Với **input** và **parameters** cho trước, công thức Lan truyền thuận trong mạng neuron được định nghĩa như sau:

$$x_j^k = \text{activation_function} \left(\left(\sum_{i=1}^{n^{k-1}} x_i^{k-1} \cdot w_{ij} \right) + bias_j \right)$$

k : Lớp đang xét

x : Giá trị của neuron

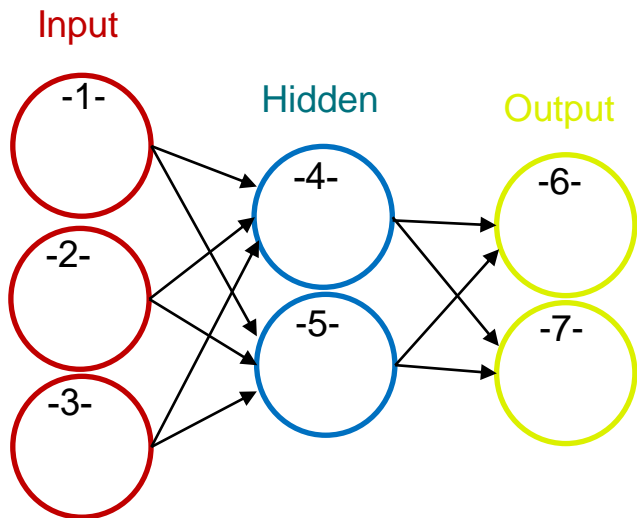
n^{k-1} : Tổng số neuron ở lớp trước đó

w_{ij} : Trọng số nối neuron thứ i và neuron thứ j

$bias_j$: Bias của neuron thứ j

5 Bài tập: Lan truyền thuận 2

Tính giá trị của output neuron \hat{y}_6 và \hat{y}_7 tại thời điểm input $X = \{x_1 = 0.2, x_2 = 0.5, x_3 = -0.4\}$, sử dụng hàm sigmoid làm activation function. Biết rằng cấu trúc và parameters của mạng như sau:



$$w_{14} = 0.2$$

$$w_{24} = 0.4$$

$$w_{34} = 0.5$$

$$w_{15} = 0.2$$

$$w_{25} = 0.3$$

$$w_{35} = 0.6$$

$$w_{46} = 0.1$$

$$w_{47} = 0.6$$

$$w_{56} = 0.5$$

$$w_{57} = 0.3$$

$$b_4 = 0.8$$

$$b_5 = 0.2$$

$$b_6 = 0.1$$

$$b_7 = 0.5$$

5 Giải:

5 Giải:

Nội dung

1. Giới thiệu bài toán
2. Giới thiệu Mạng neuron (Neural Network)
3. Kiến trúc Mạng neuron
4. Lan truyền thuận (dự đoán kết quả)
5. Các hàm kích hoạt (activation functions)
6. Lan truyền ngược (thuật toán học gradient descent)
7. Bài tập ví dụ
8. Một số vấn đề khi huấn luyện neural network

6 Lan truyền ngược

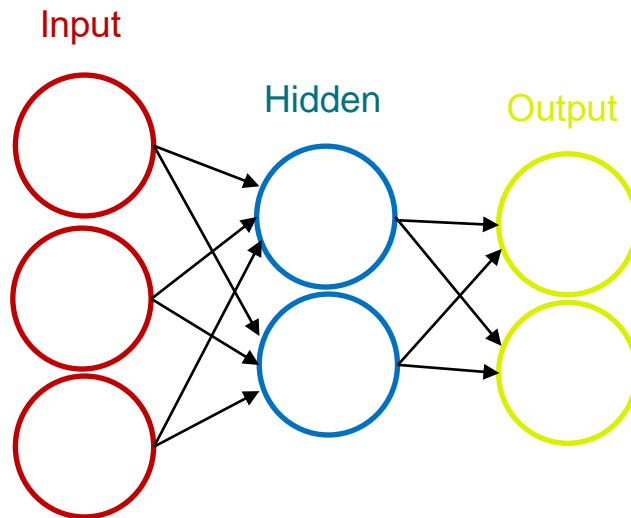
- Lan truyền ngược (back propagation) là cách gọi chung của những thuật toán giúp neural network có thể học và “sửa sai” sau quá trình Lan truyền thuận
- Có nhiều thuật toán thường được sử dụng

Gradient descent

AdaGrad

RMSProp

Adam



6

Khởi tạo

- Bước 0: Tạo cấu trúc mạng, chuẩn bị tập dữ liệu huấn luyện, gồm đầu vào x và nhãn y , tốc độ học α
- Bước 1: Khởi tạo ngẫu nhiên tất cả các giá trị weights
- Bước 2: Thực hiện Lan truyền thuận đến output layer, được các giá trị h trong hidden neuron và \hat{y} trong output neuron

- **Bước 3:**

Đối với output layer

- **3.1:** Tính độ lỗi: $e_k = y_k - \hat{y}_k$
- **3.2:** Tính sai số gradient: $\delta_k = \hat{y}_k \times (1 - \hat{y}_k) \times e_k$
- **3.3:** Tính giá trị cần thay đổi: $\Delta w_{jk} = h_j \times \delta_k$
- **3.4:** Cập nhật trọng số: $w_{jk} = w_{jk} + \alpha \times \Delta w_{jk}$
- **3.5:** Cập nhật bias: $b_k = b_k + \alpha \times \delta_k$

Đối với hidden layer, bỏ **3.1** và thay đổi **3.2**:

- **3.2:** Tính sai số gradient: $\delta_k = h_k \times (1 - h_k) \times \sum_{i=1}^n \delta_i \times w_{ki}$

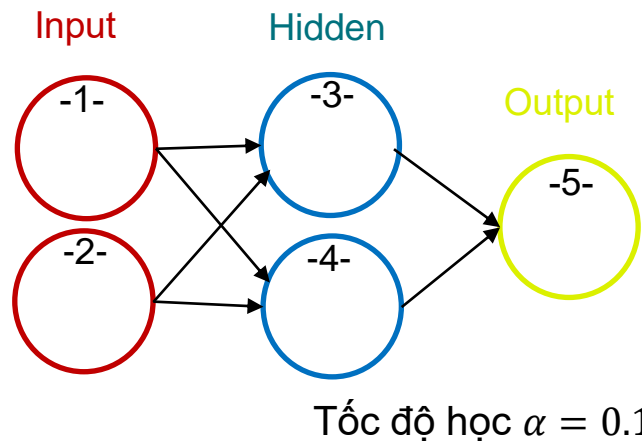
- Quá trình lan truyền ngược được thực hiện 1 lần sau mỗi lần Lan truyền thuận
- Thực hiện lặp **Bước 2** và **Bước 3** nhiều lần cho đến khi mô hình hội tụ (converge)
 - Đặt ra số lần
 - Đặt ra độ chính xác mong muốn
- Quá trình lặp **Bước 2** và **Bước 3** được gọi là quá trình huấn luyện (training process) và được thực hiện trên toàn bộ tập huấn luyện
- Sau khi dừng quá trình huấn luyện, ta sẽ có được tập các giá trị weights tối ưu cho bài toán
- Tập weights này sẽ được giữ nguyên (freeze) và mang vào sử dụng

6 Quá trình học

- Ví dụ:** Hãy tạo một mạng neuron và huấn luyện mạng để có thể thực hiện được phép XOR

Input 1 Input 2 output
 x_1 x_2 y

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



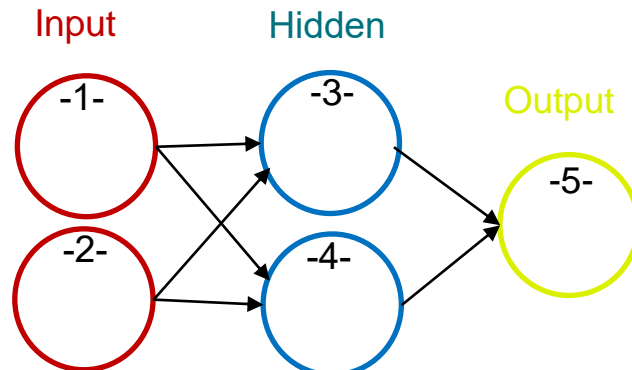
Xong **Bước 0** 😊

6 Quá trình học

- Ví dụ:** Hãy tạo một mạng neuron và huấn luyện mạng để có thể thực hiện được phép XOR

Input 1 Input 2 output
 x_1 x_2 y

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0



Tốc độ học $\alpha = 0.1$

$$w_{13} = 0.5$$

$$w_{14} = 0.9$$

$$w_{23} = 0.4$$

$$w_{24} = 1.0$$

$$w_{35} = -1.2$$

$$w_{45} = 1.1$$

$$b_3 = 0.8$$

$$b_4 = -0.1$$

$$b_5 = 0.3$$

Xong **Bước 1** 😊

6 Quá trình học

- Bước 2:** Xét mẫu huấn luyện $x_1 = 1, x_2 = 1$ và $y = 0$

$$h_3 = \text{sigmoid}(x_1 w_{13} + x_2 w_{23} + b_3)$$

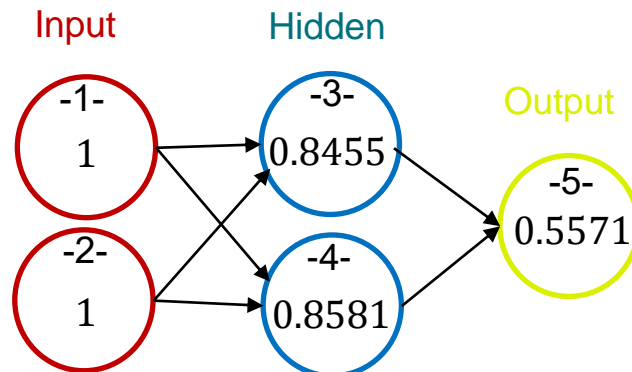
$$= \frac{1}{1 + e^{-(1 \times 0.5 + 1 \times 0.4 + 0.8)}} = 0.8455$$

$$h_4 = \text{sigmoid}(x_1 w_{14} + x_2 w_{24} + b_4)$$

$$= \frac{1}{1 + e^{-(1 \times 0.9 + 1 \times 1.0 - 0.1)}} = 0.8581$$

$$\hat{y}_5 = \text{sigmoid}(h_3 w_{35} + h_4 w_{45} + b_5)$$

$$= 0.5571$$



Tốc độ học $\alpha = 0.1$

$$w_{13} = 0.5$$

$$w_{14} = 0.9$$

$$w_{23} = 0.4$$

$$w_{24} = 1.0$$

$$w_{35} = -1.2$$

$$w_{45} = 1.1$$

$$b_3 = 0.8$$

$$b_4 = -0.1$$

$$b_5 = 0.3$$

6 Quá trình học

● Bước 3 Thực hiện huấn luyện mô hình

Lan truyền ngược tại output layer

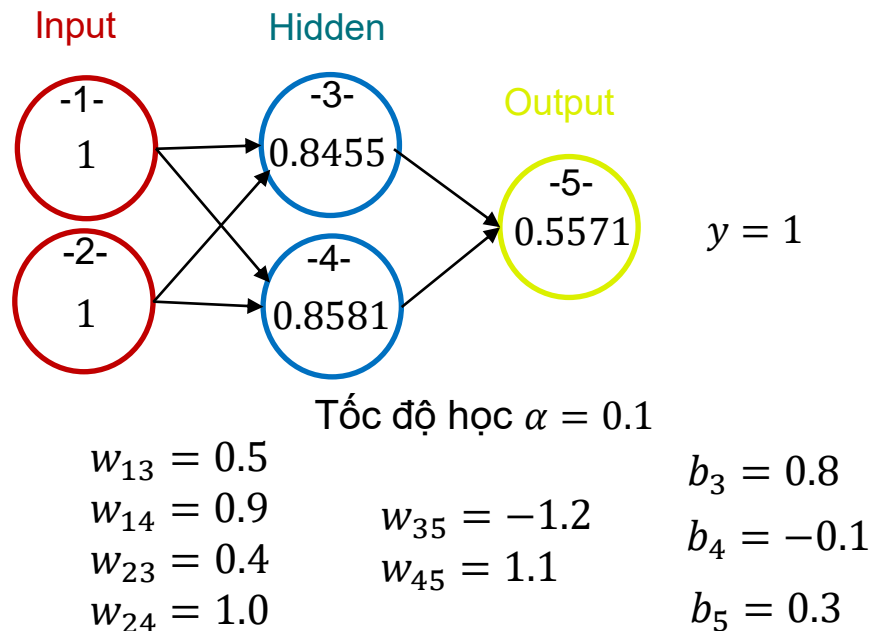
$$e_5 = y_5 - \hat{y}_5 = 0 - 0.5571 = -0.5571$$

$$\begin{aligned}\delta_5 &= \hat{y}_5 \times (1 - \hat{y}_5) \times e_5 \\ &= 0.5571 \times (1 - 0.5571) \times (-0.5571) \\ &= -0.1375\end{aligned}$$

$$\begin{aligned}\Delta w_{35} &= h_3 \times \delta_5 \\ &= 0.8455 \times (-0.1375) \\ &= -0.1163\end{aligned}$$

$$\begin{aligned}\Delta w_{45} &= h_4 \times \delta_5 \\ &= 0.8581 \times (-0.1375) \\ &= -0.1180\end{aligned}$$

- 3.1: $e_k = y - \hat{y}$
- 3.2: $\delta_k = \hat{y}_k \times (1 - \hat{y}_k) \times e_k$
- 3.3: $\Delta w_{jk} = h_j \times \delta_k$



6

Quá trình học

● Bước 3: Thực hiện huấn luyện mô hình

Cập nhật tại output layer

$$\Delta w_{35} = -0.1163$$

$$\delta_5 = -0.1375$$

$$\Delta w_{45} = -0.1180$$

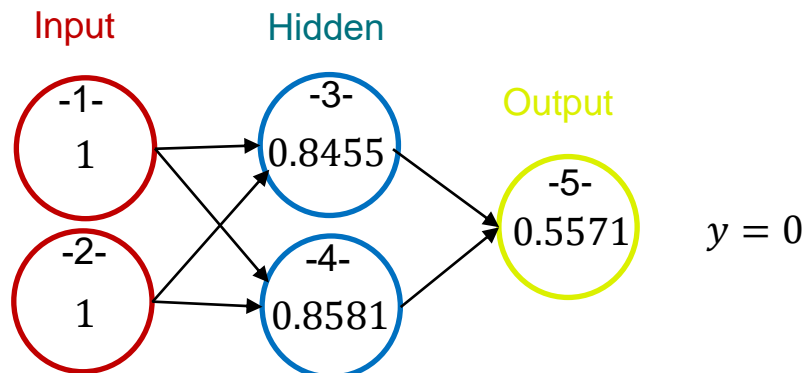
$$\begin{aligned} w_{35} &= w_{35} + \alpha \times \Delta w_{35} \\ &= -1.2 + 0.1 \times (-0.1163) \\ &= -1.21163 \end{aligned}$$

$$\begin{aligned} w_{45} &= w_{45} + \alpha \times \Delta w_{45} \\ &= 1.1 + 0.1 \times (-0.1180) \\ &= 1.0882 \end{aligned}$$

$$\begin{aligned} b_5 &= b_5 + \alpha \times \delta_5 \\ &= 0.3 + 0.1 \times (-0.1375) \\ &= 0.28625 \end{aligned}$$

$$\circ \mathbf{3.4:} \quad w_{jk} = w_{jk} + \alpha \times \Delta w_{jk}$$

$$\circ \mathbf{3.5:} \quad b_k = b_k + \alpha \times \delta_k$$



Tốc độ học $\alpha = 0.1$

$$w_{13} = 0.5$$

$$w_{14} = 0.9$$

$$w_{23} = 0.4$$

$$w_{24} = 1.0$$

$$w_{35} = -1.2$$

$$w_{45} = 1.1$$

$$w_{35} = -1.21163$$

$$w_{45} = 1.0882$$

$$b_3 = 0.8$$

$$b_4 = -0.1$$

$$b_5 = 0.3$$

$$b_5 = 0.28625$$

6

Quá trình học

● Bước 3: Thực hiện huấn luyện mô hình

Lan truyền ngược tại hidden layer

- neuron h_3

$$\delta_5 = -0.1375$$

$$\begin{aligned}\delta_3 &= h_3 \times (1 - h_3) \times \delta_5 \times w_{35} \\ &= 0.8455 \times (1 - 0.8455) \times (-0.1375) \\ &\quad \times (-1.2) = 0.0216\end{aligned}$$

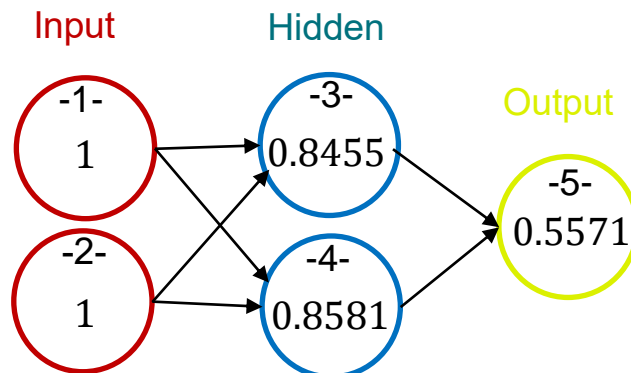
$$\begin{aligned}\Delta w_{13} &= x_1 \times \delta_3 \\ &= 1 \times 0.0216 \\ &= 0.0216\end{aligned}$$

$$\begin{aligned}\Delta w_{23} &= x_2 \times \delta_3 \\ &= 1 \times 0.0216 \\ &= 0.0216\end{aligned}$$

○3.1: (không có)

○3.2: $\delta_k = h_k \times (1 - h_k) \times \sum_{i=1}^n \delta_i \times w_{ki}$

○3.3: $\Delta w_{jk} = x_j \times \delta_k$



Tốc độ học $\alpha = 0.1$

$$w_{13} = 0.5$$

$$w_{14} = 0.9$$

$$w_{23} = 0.4$$

$$w_{24} = 1.0$$

$$w_{35} = -1.2$$

$$w_{45} = 1.1$$

$$w_{35} = -1.21163$$

$$w_{45} = 1.0882$$

$$b_3 = 0.8$$

$$b_4 = -0.1$$

$$b_5 = 0.3$$

$$b_5 = 0.28625$$

6

Quá trình học

- Bước 3:** Thực hiện huấn luyện mô hình

Cập nhật tại hidden layer - neuron h_3

$$\delta_3 = \Delta w_{13} = \Delta w_{23} = 0.0216$$

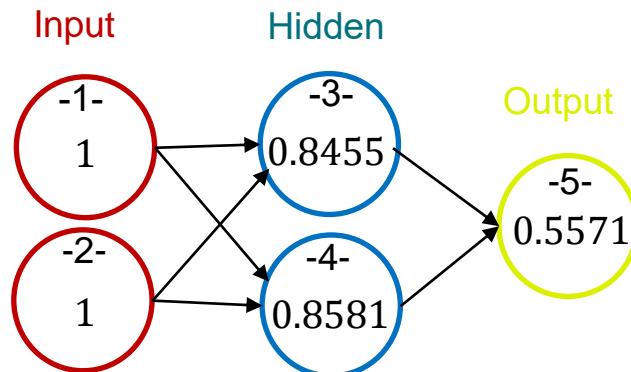
$$\begin{aligned} w_{13} &= w_{13} + \alpha \times \Delta w_{13} \\ &= 0.5 + 0.1 \times 0.0216 \\ &= 0.50216 \end{aligned}$$

$$\begin{aligned} w_{23} &= w_{23} + \alpha \times \Delta w_{23} \\ &= 0.4 + 0.1 \times 0.0216 \\ &= 0.40216 \end{aligned}$$

$$\begin{aligned} b_3 &= b_3 + \alpha \times \delta_3 \\ &= 0.8 + 0.1 \times 0.0216 \\ &= 0.80216 \end{aligned}$$

$$\circ 3.4: w_{jk} = w_{jk} + \alpha \times \Delta w_{jk}$$

$$\circ 3.5: b_k = b_k + \alpha \times \delta_k$$



Tốc độ học $\alpha = 0.1$

$$\begin{aligned} w_{13} &= 0.5 \\ w_{14} &= 0.9 \\ w_{23} &= 0.4 \\ w_{24} &= 1.0 \\ w_{13} &= 0.50216 \\ w_{23} &= 0.40216 \end{aligned}$$

$$\begin{aligned} w_{35} &= -1.2 \\ w_{45} &= 1.1 \\ w_{35} &= -1.21163 \\ w_{45} &= 1.0882 \end{aligned}$$

$$\begin{aligned} b_3 &= 0.8 \\ b_4 &= -0.1 \\ b_5 &= 0.3 \\ b_5 &= 0.28625 \\ b_3 &= 0.80216 \end{aligned}$$

6 Quá trình học

● Bước 3: Thực hiện huấn luyện mô hình

Lan truyền ngược tại hidden layer

- neuron h_4

$$\delta_5 = -0.1375$$

$$\delta_4 =$$

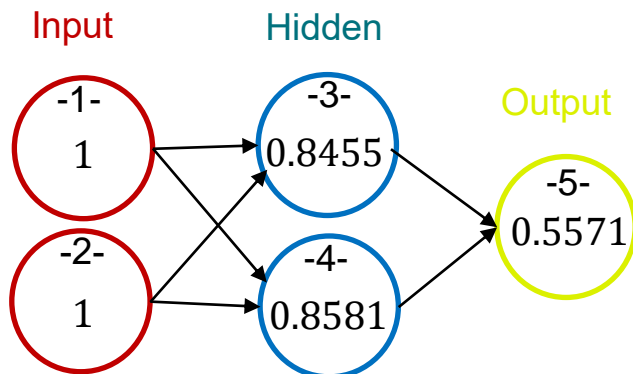
$$\Delta w_{14} =$$

$$\Delta w_{24} =$$

○ 3.1: (không có)

○ 3.2: $\delta_k = h_k \times (1 - h_k) \times \sum_{i=1}^n \delta_i \times w_{ki}$

○ 3.3: $\Delta w_{jk} = x_j \times \delta_k$



Tốc độ học $\alpha = 0.1$

$$w_{13} = 0.5$$

$$w_{14} = 0.9$$

$$w_{23} = 0.4$$

$$w_{24} = 1.0$$

$$w_{13} = 0.50216$$

$$w_{23} = 0.40216$$

$$w_{35} = -1.2$$

$$w_{45} = 1.1$$

$$w_{35} = -1.21163$$

$$w_{45} = 1.0882$$

$$b_3 = 0.8$$

$$b_4 = -0.1$$

$$b_5 = 0.3$$

$$b_5 = 0.28625$$

$$b_3 = 0.80216$$

6 Quá trình học

- **Bước 3:** Thực hiện huấn luyện mô hình

Cập nhật tại hidden layer - neuron h_4

$$\delta_4 =$$

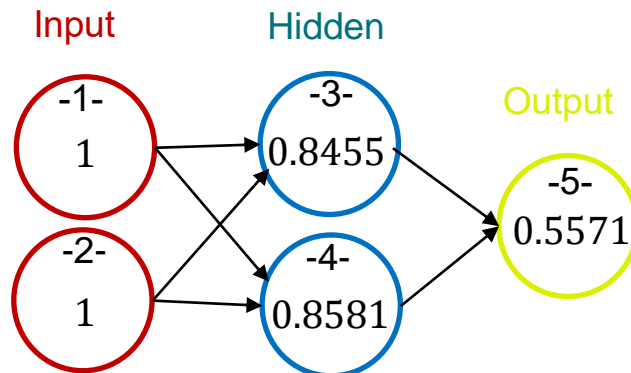
$$w_{14} =$$

$$w_{24} =$$

$$b_4 =$$

$$\circ 3.4: w_{jk} = w_{jk} + \alpha \times \Delta w_{jk}$$

$$\circ 3.5: b_k = b_k + \alpha \times \delta_k$$



Tốc độ học $\alpha = 0.1$

$$w_{13} = 0.5$$

$$w_{14} = 0.9$$

$$w_{23} = 0.4$$

$$w_{24} = 1.0$$

$$w_{13} = 0.50216$$

$$w_{23} = 0.40216$$

$$w_{35} = -1.2$$

$$w_{45} = 1.1$$

$$w_{35} = -1.21163$$

$$w_{45} = 1.0882$$

$$b_3 = 0.8$$

$$b_4 = -0.1$$

$$b_5 = 0.3$$

$$b_5 = 0.28625$$

$$b_3 = 0.80216$$

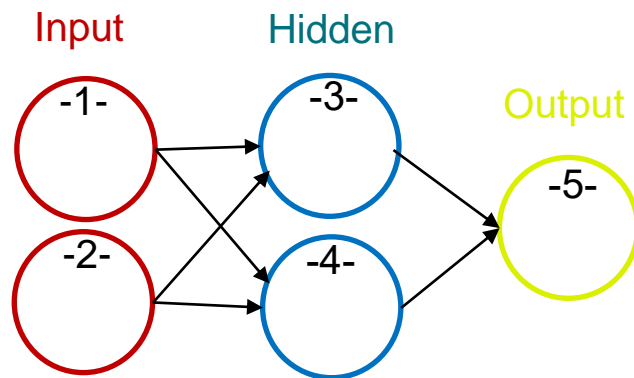
7 Bài tập ví dụ

Bài tập: Hãy tiếp tục huấn luyện mô hình ở lần lặp thứ hai, sử dụng kết quả từ lần lặp đầu tiên với mẫu huấn luyện:

$$x_1 = 1, x_2 = 1, y = 0$$

→ **Giải:**

Bước 2: Lan truyền thuận

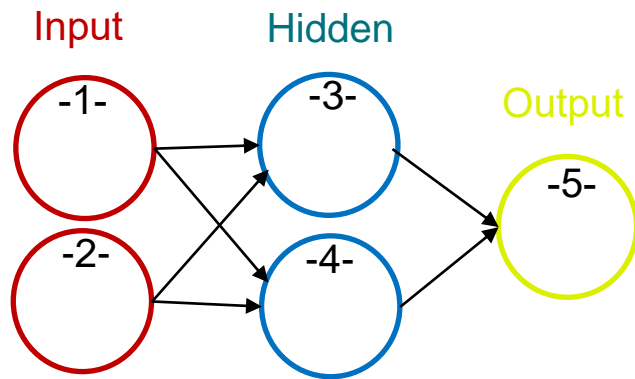


Tốc độ học $\alpha = 0.1$

7 Bài tập ví dụ

→ Giải:

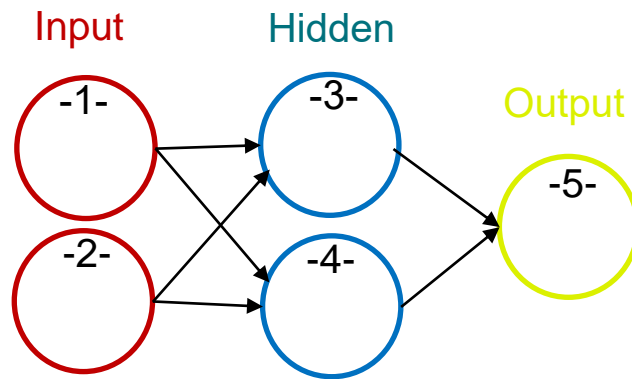
Bước 3: Lan truyền ngược tại output layer:



Tốc độ học $\alpha = 0.1$

7 Bài tập ví dụ

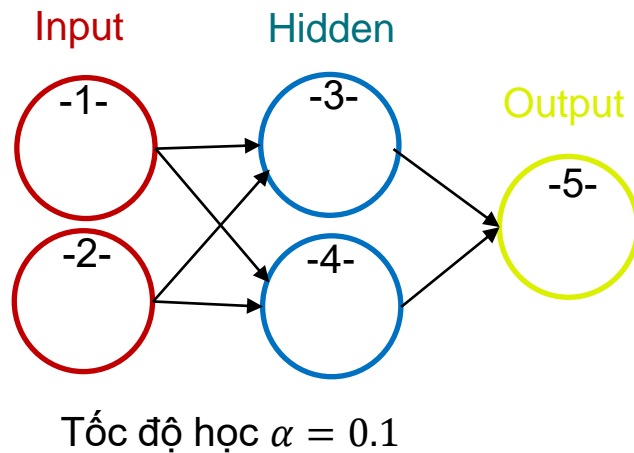
Cập nhật tại output layer



Tốc độ học $\alpha = 0.1$

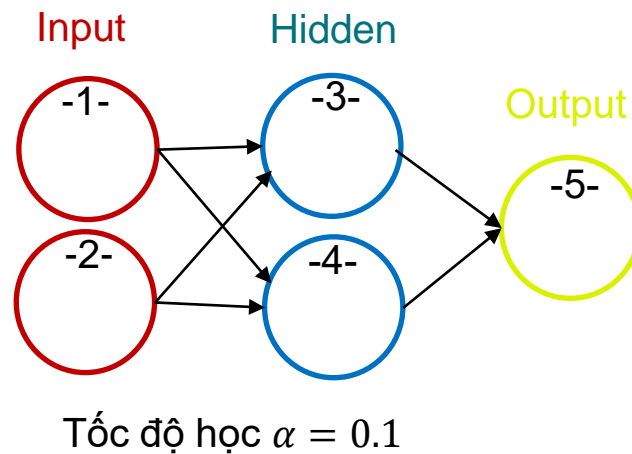
7 Bài tập ví dụ

Bước 4: Lan truyền ngược tại hidden layer – neuron h_3 :



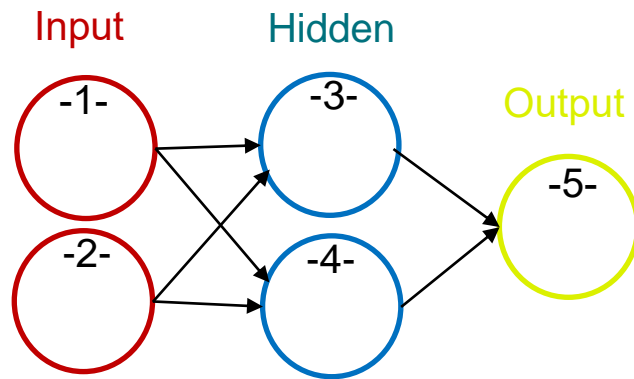
7 Bài tập ví dụ

Update tại hidden layer - neuron h_3 :



7 Bài tập ví dụ

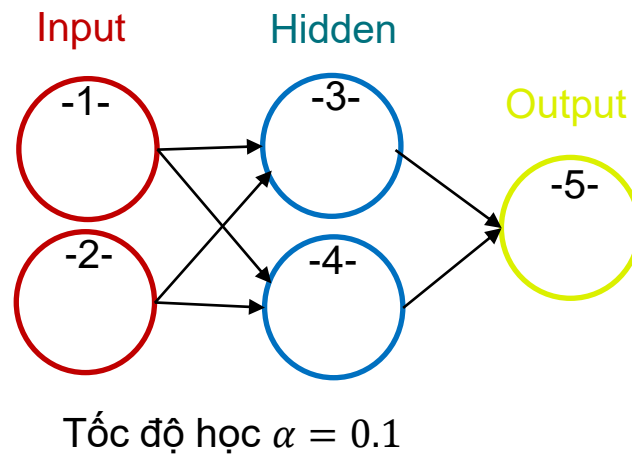
Lan truyền ngược tại hidden layer – neuron h_4 :



Tốc độ học $\alpha = 0.1$

7 Bài tập ví dụ

Update tại hidden layer - neuron h_4 :



7 Kết quả

Sau quá trình huấn luyện trải qua 10000 thế hệ (epoch) - hay 40000 vòng lặp - kết thúc:

- Tổng bình phương sai số bằng 0.01
- Tập trọng số và độ điều chỉnh tối ưu thu được:

$$w_{13} = 3.7441$$

$$w_{35} = -8.5614$$

$$w_{14} = 6.0029$$

$$w_{45} = 7.7689$$

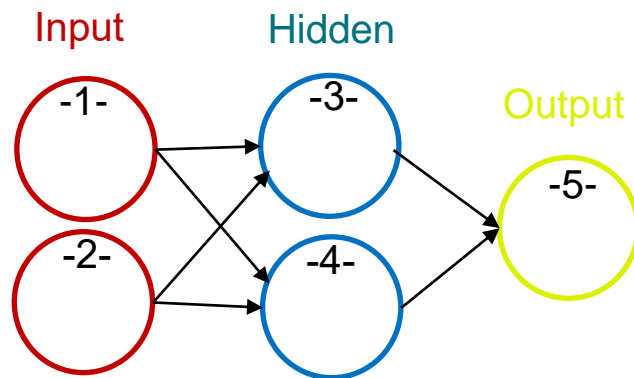
$$w_{23} = 3.7312$$

$$w_{24} = 5.9413$$

$$b_3 = -5.7384$$

$$b_4 = -2.5519$$

$$b_5 = -3.4613$$



Tốc độ học $\alpha = 0.1$

7 Kết quả

- Với giá trị trọng số và độ điều chỉnh tối ưu, kết quả dự đoán (predict) cho phép toán XOR như sau

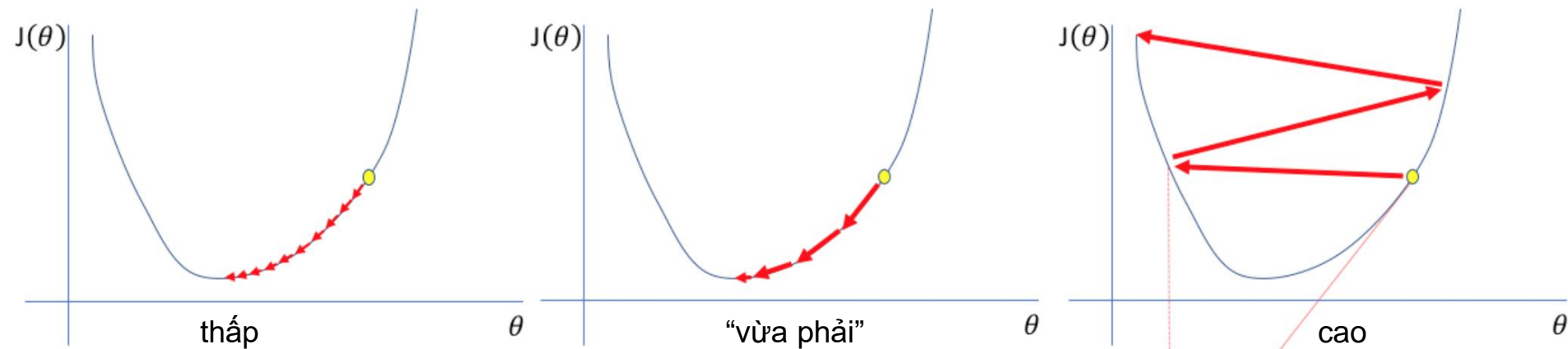
Đầu vào		Kết quả mong muốn	Kết quả từ neural network	Sai số
x_1	x_2	y	\hat{y}	e
1	1	0	0.0486	-0.0486
0	1	1	0.9543	0.0457
1	0	1	0.9544	0.0456
0	0	0	0.0508	-0.0508

Nội dung

1. Giới thiệu bài toán
2. Giới thiệu Mạng neuron (Neural Network)
3. Kiến trúc Mạng neuron
4. Lan truyền thuận (dự đoán kết quả)
5. Các hàm kích hoạt (activation functions)
6. Lan truyền ngược (thuật toán học gradient descent)
7. Bài tập ví dụ
8. Một số vấn đề khi huấn luyện neural network

8 Hyper parameters

- Số lượng neuron trong hidden layer:
 - Ít: hội tụ nhanh, phù hợp các bài toán phân lớp đơn giản
 - Nhiều: hội tụ chậm, dễ gây overfitting khi có ít dữ liệu để train. Bù lại model sẽ học được những phân phối phức tạp (nếu đủ dữ liệu)
- Hệ số học/ tốc độ học (learning rate):



8 Tính toán trên ma trận

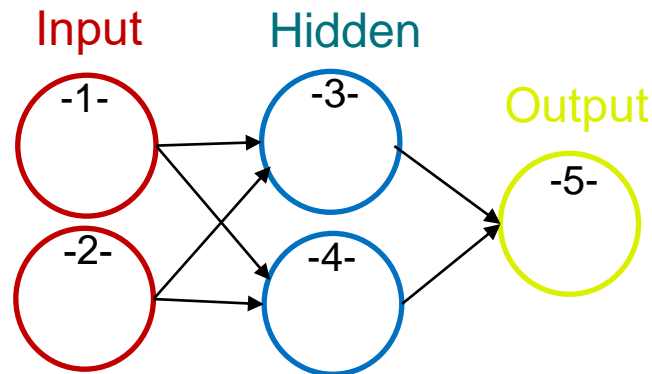
- Sử dụng phép nhân ma trận (dot product) để thực hiện các phép tính trong neural network

➤ *Tính toán trên scalar* - Input: x_1, x_2

$$h_3 = \text{activate}(x_1 \times w_{13} + x_2 \times w_{23} + b_3)$$

$$h_4 = \text{activate}(x_1 \times w_{14} + x_2 \times w_{24} + b_4)$$

$$y_5 = \text{activate}(h_3 \times w_{35} + h_4 \times w_{45} + b_5)$$



➤ *Tính toán trên ma trận*: Gộp scalar \rightarrow ma trận - Input: $\mathbf{X} = [x_1 \quad x_2]$

$$\mathbf{H} = \text{activate}(\mathbf{X} \cdot \mathbf{W} + \mathbf{b})$$

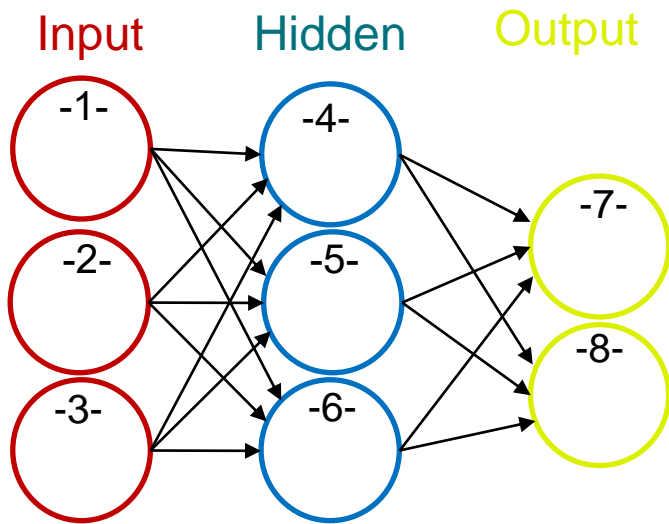
$$= \text{activate}([x_1 \quad x_2] \cdot \begin{bmatrix} w_{13} & w_{14} \\ w_{23} & w_{24} \end{bmatrix} + [b_3 \quad b_4])$$

$$= \text{activate}([x_1 \times w_{13} + x_2 \times w_{23} \quad x_1 \times w_{14} + x_2 \times w_{24}] + [b_3 \quad b_4])$$

$$= \text{activate}([h_3 \quad h_4])$$

8 Bài tập về nhà

Bài 1: Cho mạng neuron có cấu trúc sau:



$$w_{14} = 0.2$$

$$w_{15} = 0.1$$

$$w_{16} = -0.3$$

$$w_{24} = 0.4$$

$$w_{25} = 0.7$$

$$w_{26} = 0.3$$

$$w_{34} = -0.5$$

$$w_{35} = 0.1$$

$$w_{36} = 0.9$$

$$w_{47} = 0.4$$

$$w_{48} = 0.6$$

$$w_{57} = -0.2$$

$$w_{58} = -0.1$$

$$w_{67} = 0.7$$

$$w_{68} = 0.8$$

$$b_4 = -0.6$$

$$b_5 = 1.0$$

$$b_6 = -0.2$$

$$b_7 = 0.8$$

$$b_8 = 0.3$$

Với tốc độ học $\alpha = 0.1$, hàm kích hoạt là sigmoid và input $\mathbf{X} = [3, 1, -2]$.

Hãy tính kết quả lớp output \mathbf{Y} sau khi thực hiện Lan truyền thuận lần 3

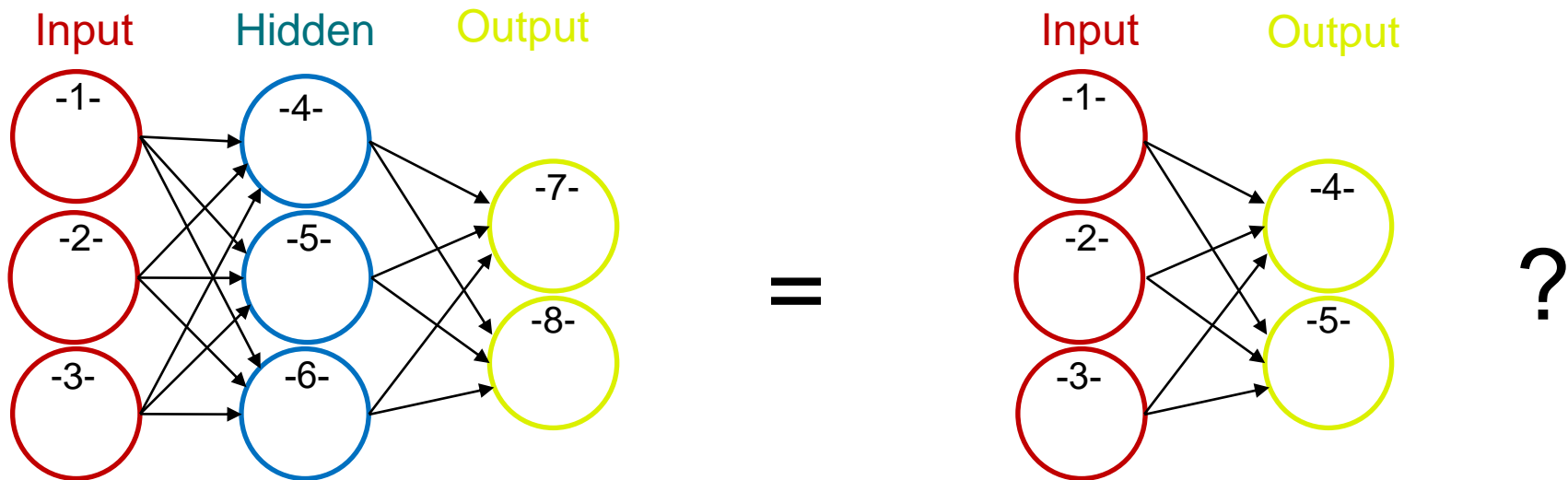
Sử dụng phương pháp nhân ma trận thay vì scalar

Forward prop \rightarrow back prop \rightarrow forward prop \rightarrow back prop \rightarrow forward prop

8 Bài tập về nhà

Bài 2 (optional): Hãy chứng minh:

Với parameters tùy ý, nếu không có activation function, thì hai models sau hoàn toàn tương tự nhau



Basic FNN formulas in a nutshell 😊

- Bước 0: Tạo cấu trúc mạng, chuẩn bị tập dữ liệu huấn luyện, gồm đầu vào x và nhãn y , tốc độ học α
- Bước 1: Khởi tạo ngẫu nhiên tất cả các giá trị weights
- Bước 2: Thực hiện Lan truyền thuận đến output layer, được các giá trị h trong hidden neuron và \hat{y} trong output neuron
- **Bước 3:** Lan truyền ngược:

Đối với output layer

- **3.1:** Tính độ lỗi: $e_k = y_k - \hat{y}_k$
- **3.2:** Tính sai số gradient: $\delta_k = \hat{y}_k \times (1 - \hat{y}_k) \times e_k$
- **3.3:** Tính giá trị cần thay đổi: $\Delta w_{jk} = h_j \times \delta_k$
- **3.4:** Cập nhật trọng số: $w_{jk} = w_{jk} + \alpha \times \Delta w_{jk}$
- **3.5:** Cập nhật bias: $b_k = b_k + \alpha \times \delta_k$

Đối với hidden layer, bỏ **3.1** và thay đổi **3.2**:

- **3.2:** Tính sai số gradient: $\delta_k = h_k \times (1 - h_k) \times \sum_{i=1}^n \delta_i \times w_{ki}$