



TRƯỜNG ĐẠI HỌC VINH

# Mạng nơ ron cơ bản

Phan Anh Phong, PhD.  
Vinh University

1

## Nội dung



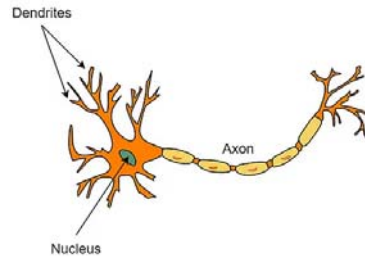
- Mạng nơ ron sinh học và ANN
- Mạng Perceptron
- Các mô hình mạng nơ ron thông dụng
- Kiến trúc mạng nơ ron 1 lớp (1 tầng)
- Mạng nơ ron nhiều lớp (nhiều tầng)
- Thảo luận

2

## Một nơ ron sinh học vs nhân tạo



- Mạng nơ-ron sinh học là mạng lưới các tế bào thần kinh (nơ-ron). Các nơ-ron sinh học kết nối thông qua các điểm tiếp xúc chuyên biệt (**khớp thần kinh**).
- Các tín hiệu truyền từ nơ-ron này sang nơ-ron khác mang theo các **hướng dẫn** cho hành động, suy nghĩ và học tập

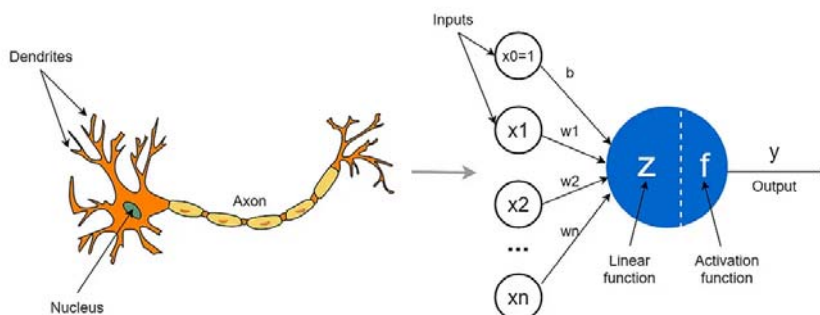


3

## Một nơ ron sinh học vs nhân tạo



- Một nơ-ron có thể nhận nhiều đầu vào và cho ra một kết quả duy nhất (đầu ra)

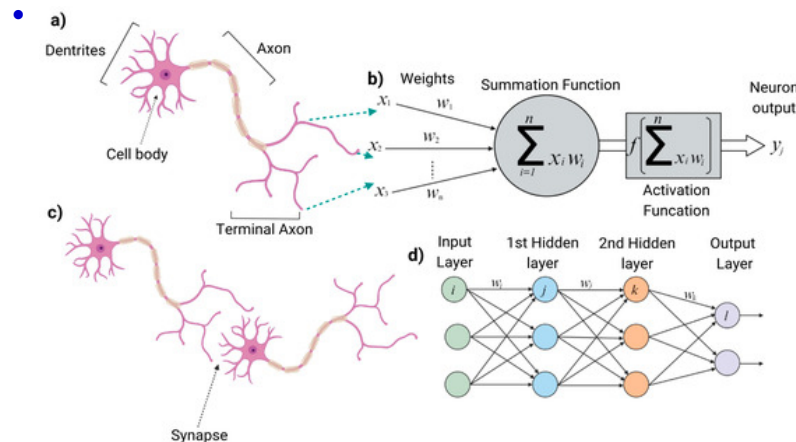


4

## Mạng nơ-ron



- Một nơ-ron và một mạng nơ-ron



5

## Mạng nơ-ron nhân tạo



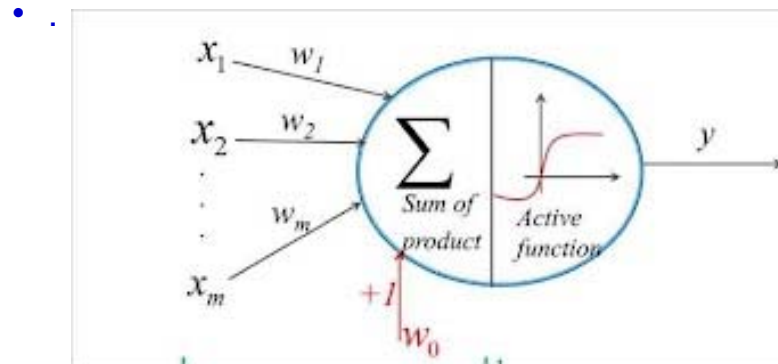
- Mạng nơ-ron nhân tạo (ANN) là một chương trình máy tính hoặc mô hình học máy ra quyết định theo cách tương tự như não người.
- Mạng nơ-ron nhân tạo có kiến trúc và hoạt động dựa vào mô phỏng kiến trúc và cách các nơ-ron sinh học hoạt động cùng nhau.
- ANN đơn giản hơn rất nhiều so với mạng nơ-ron sinh học!

6

## Một nơ ron nhân tạo



- Một nơ-ron có thể nhận nhiều đầu vào và cho ra một kết quả duy nhất (đầu ra)

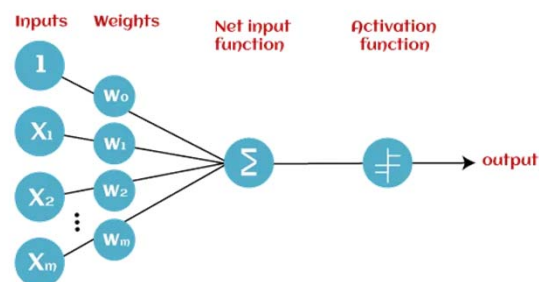


7

## Perceptron



- Perceptron là một trong những thuật toán học có giám sát đơn giản nhất cho bài toán phân lớp nhị phân (2 lớp).
- Perceptron là nền tảng của mạng neural hiện đại.



8

## Perceptron



- Perceptron là mạng nơ ron đơn giản nhất, được phát triển bởi Frank Rosenblatt vào năm 1958.
- Gồm các thành phần:
  - Các đầu vào (inputs): Nhận dữ liệu vào  $x_1, x_2, \dots, x_n$
  - Trọng số (weights): Mỗi đầu vào được nhân với trọng số tương ứng  $w_1, w_2, \dots, w_n$
  - Ngưỡng (bias): Một giá trị  $b$
  - Hàm kích hoạt (activation function): có nhiều lựa chọn (giả sử chọn step function)

9

## Một số hàm kích hoạt



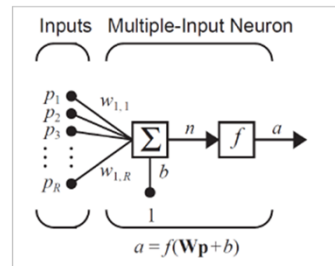
Hàm kích hoạt	Vào/Ra
Step (Hard Limit/Heaviside)	$a = f(n) = \begin{cases} 1, & \text{nếu } n \geq 0 \\ 0, & \text{nếu } n < 0 \end{cases}$
Linear	$a = f(n) = n$
Exponential Linear Unit	$a = f(n) = \begin{cases} n & \text{nếu } n \geq 0 \\ \alpha(e^\alpha - 1) & \text{nếu } n < 0 \end{cases}$
Scaled Exponential Linear Unit	$a = f(n) = \alpha \begin{cases} n & \text{nếu } n \geq 0 \\ \alpha(e^\alpha - 1) & \text{nếu } n < 0 \end{cases}$
<b>Softplus</b>	<b><math>a = f(n) = \log_n(1 + e^n)</math></b>
<b>Rectified Linear Unit (ReLU)</b>	<b><math>a = f(n) = \max(0, n)</math></b>
<b>Tanh</b>	<b><math>a = f(n) = \frac{2}{1 + e^{-2n}} - 1</math></b>
<b>Sigmoid</b>	<b><math>a = f(n) = \frac{1}{1 + e^{-n}}</math></b>

10

## Perceptron



- Mô tả một nơ-ron nhân tạo, trong đó:
  - $p$  là vector đầu vào
  - $W$  là vector trọng số
  - $b$  là độ lệch (bias)
  - $f$  là hàm kích hoạt (activation function)
  - $a$  là đầu ra của nơ-ron

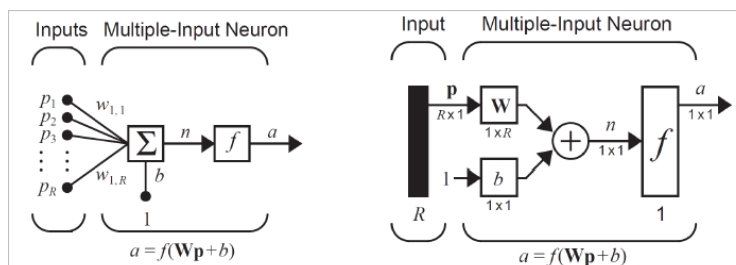


11

## Perceptron



- Vào,  $p = [p_1, p_2, \dots, p_R]^T$  là một vec-tơ cột.
- Trọng số,  $W = [w_{1,1}, w_{1,2}, \dots, w_{1,R}]$  là một vec-tơ hàng.
- Độ lệch,  $b$ , là một số vô hướng.
- Hàm TỔNG,  $n = w_{1,1}p_1 + \dots + w_{1,R}p_R + b = Wp + b$ .
- Hàm kích hoạt  $f$  tạo ra đầu ra của nơ-ron:  $a = f(n) = f(Wp + b)$ .

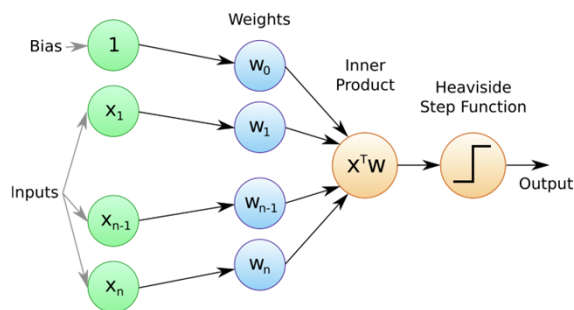


12

## Perceptron



- Cách hoạt động của perceptron cụ thể:
  - Tính tổng có trọng số:  $z = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$
  - Giả sử hàm kích hoạt được chọn là hàm step:  $y = f(z)$ 
    - Nếu  $z > 0$ , output = 1
    - Nếu  $z \leq 0$ , output = 0



13

## Perceptron



- Ví dụ: Một perceptron nhận 3 đầu vào ( $x_1, x_2, x_3$ ) với các trọng số tương ứng là  $w_1 = 0.5$ ,  $w_2 = -0.2$ ,  $w_3 = 0.8$  và ngưỡng (bias)  $b = -0.3$ .
- Mô hình toán học:
  - Đầu vào:  $x_1 = 1$ ,  $x_2 = 0$ ,  $x_3 = 1$
  - Trọng số:  $w_1 = 0.5$ ,  $w_2 = -0.2$ ,  $w_3 = 0.8$
  - Ngưỡng:  $b = -0.3$
  - Tổng có trọng số:  $z = w_1x_1 + w_2x_2 + w_3x_3 + b$
  - Hàm kích hoạt: hàm step:
    - $f(z) = 1$  nếu  $z > 0$
    - $f(z) = 0$  nếu  $z \leq 0$

14

## Perceptron



- Mô hình toán học:
  - Đầu vào:  $x_1, x_2, x_3$
  - Trọng số:  $w_1 = 0.5, w_2 = -0.2, w_3 = 0.8$
  - Ngưỡng:  $b = -0.3$
  - Tổng có trọng số:  $z = w_1x_1 + w_2x_2 + w_3x_3 + b$
  - Hàm kích hoạt (hàm step với  $t = 0$ ):
    - $f(z) = 1$  nếu  $z > 0$
    - $f(z) = 0$  nếu  $z \leq 0$
- Tính toán: Giả sử có bộ đầu vào  $x_1 = 1, x_2 = 0, x_3 = 1$ 
  - Tính tổng có trọng số:  $z = 0.5(1) + (-0.2)(0) + 0.8(1) + (-0.3) z = 0.5 + 0 + 0.8 - 0.3 z = 1.0$
  - Áp dụng hàm kích hoạt:  $f(z) = f(1.0) = 1$  (vì  $1.0 > 0$ )

15

## Cách học từ dữ liệu của perceptron



- Khởi tạo: Bắt đầu với các trọng số ( $w_1, w_2, \dots, w_n$ ) và ngưỡng ( $b$ ) được gán giá trị ngẫu nhiên hoặc 0.
- Quá trình học:
  - Đưa một mẫu đầu vào  $x = (x_1, x_2, \dots, x_n)$  vào mạng
  - Tính đầu ra dự đoán:  $\hat{y} = f(w_1x_1 + w_2x_2 + \dots + w_nx_n + b)$
  - So sánh với đầu ra mong muốn  $y$
  - Điều chỉnh trọng số nếu có lỗi
- 

16



## Cách học từ dữ liệu của perceptron



- Quy tắc cập nhật trọng số:
  - $w_i(\text{mới}) = w_i(\text{cũ}) + \eta(y - \hat{y})x_i$
  - Trong đó:
    - $w_i$ : trọng số thứ  $i$
    - $\eta$ : tốc độ học (learning rate) - một hằng số dương nhỏ
    - $y$ : đầu ra mong muốn (0 hoặc 1)
    - $\hat{y}$ : đầu ra dự đoán của perceptron
    - $x_i$ : giá trị đầu vào thứ  $i$
- Cập nhật ngưỡng:
  - $b(\text{mới}) = b(\text{cũ}) + \eta(y - \hat{y})$
- Thuật toán dừng khi:
  - Đạt đến số lần lặp tối đa, hoặc
  - Tất cả các mẫu huấn luyện được phân loại chính xác, hoặc
  - Thay đổi trọng số dưới một ngưỡng nhất định

17

## Cấu trúc chung của mạng nơ ron



- Các lớp:
  - Lớp đầu vào : Nhận dữ liệu đầu vào từ bên ngoài
  - Lớp ẩn : Xử lý thông tin, có thể có nhiều lớp ẩn khác nhau
  - Lớp đầu ra : Cung cấp kết quả cuối cùng của mạng
- Nơ-ron, mỗi nơ-ron, bao gồm:
  - Các đầu vào, mỗi đầu vào có một trọng số tương ứng
  - Trọng số: Xác định mức độ ảnh hưởng của mỗi đầu vào đến nơ-ron
  - Hàm tổng: để tính tổng theo trọng số của các đầu vào
  - Bias - một tham số để điều chỉnh ngưỡng kích hoạt
  - Hàm kích hoạt biến đổi tổng đầu vào thành đầu ra
- Hàm kích hoạt
  - Chuyển đổi tín hiệu đầu vào thành đầu ra

18

## Cấu trúc chung của mạng nơ ron



- Quá trình học:
  - Lan truyền xuôi (Forward propagation): Dữ liệu được truyền từ lớp đầu vào qua các lớp ẩn đến lớp đầu ra
  - Lan truyền ngược (Backpropagation): Cập nhật trọng số và bias dựa trên sai số giữa kết quả dự đoán và kết quả thực tế
  - Hàm mất mát (Loss function): Đo lường độ chính xác của mạng, đo sai số thực tế với sai số dự đoán.
  - Thuật toán tối ưu hóa (Optimization): Thuật toán giúp cập nhật trọng số để giảm thiểu hàm mất mát (như Gradient Descent)

19

## Các mô hình mạng nơ ron



1. Perceptron (Mô hình đơn giản nhất)
  - Ưu điểm:
    - Đơn giản, dễ hiểu và triển khai.
    - Hiệu quả với dữ liệu phân loại tuyến tính.
  - Nhược điểm:
    - Không giải quyết được bài toán phi tuyến (ví dụ: XOR).
    - Không có tầng ẩn.
  - Ứng dụng:
    - Ít được dùng trong thực tế do quá đơn giản.
    - Cơ sở để xây dựng các mô hình phức tạp hơn.

20

## Các mô hình mạng nơ ron



### 2. Mạng Nơ-ron truyền thẳng (Feedforward Neural Network - FNN)

- Ưu điểm:
  - Có thể xử lý dữ liệu phi tuyến nhờ các lớp ẩn.
  - Phù hợp với bài toán phân loại hoặc hồi quy.
- Nhược điểm:
  - Dễ bị overfitting nếu không có regularization (kỹ thuật điều chỉnh L1/L2 - phạt các trọng số lớn, Hoặc Dropout – tắt 1 số nơ ron)
  - Không phù hợp với dữ liệu sequential data, hoặc dữ liệu không gian (như ảnh).
- Ứng dụng:
  - Nhận dạng chữ viết tay (MNIST).
  - Dự đoán giá nhà, phân loại sản phẩm.

21

## Các mô hình mạng nơ ron



### 3. Multilayer Perceptron (MLP) là 1 FNN có ít nhất một lớp ẩn.

- Ưu điểm:
  - Giải quyết được bài toán phi tuyến.
  - Phổ biến và mạnh mẽ trong nhiều bài toán.
- Nhược điểm:
  - Dễ overfitting nếu không regularize tốt.
  - Cần nhiều dữ liệu và thời gian huấn luyện.
- Ứng dụng:
  - Nhận dạng chữ viết tay, phân loại hình ảnh đơn giản, dự đoán chuỗi.

22

## Các mô hình mạng nơ ron



### 4. Mạng Nơ-ron tích chập (Convolutional Neural Network - CNN)

- Xử lý ảnh

### 5. Mạng Nơ-ron hồi tiếp (Recurrent Neural Network - RNN)

- Xử lý ngôn ngữ tự nhiên

### 6. Mạng Transformer

- Mô hình ngôn ngữ lớn
- Xử lý ảnh.

### 7. Mạng đối nghịch tạo sinh (Generative Adversarial Network - GAN)

- Tạo ảnh nghệ thuật, deepfake, tăng cường dữ liệu

### 8. ... (more)

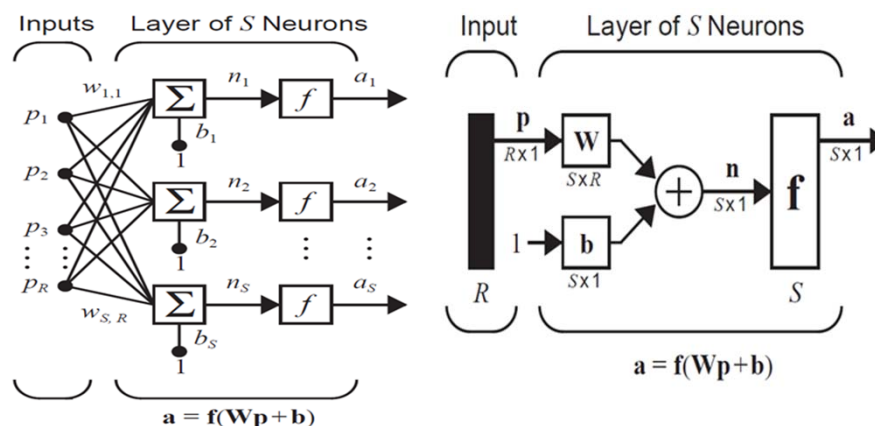
Deep Learning

23

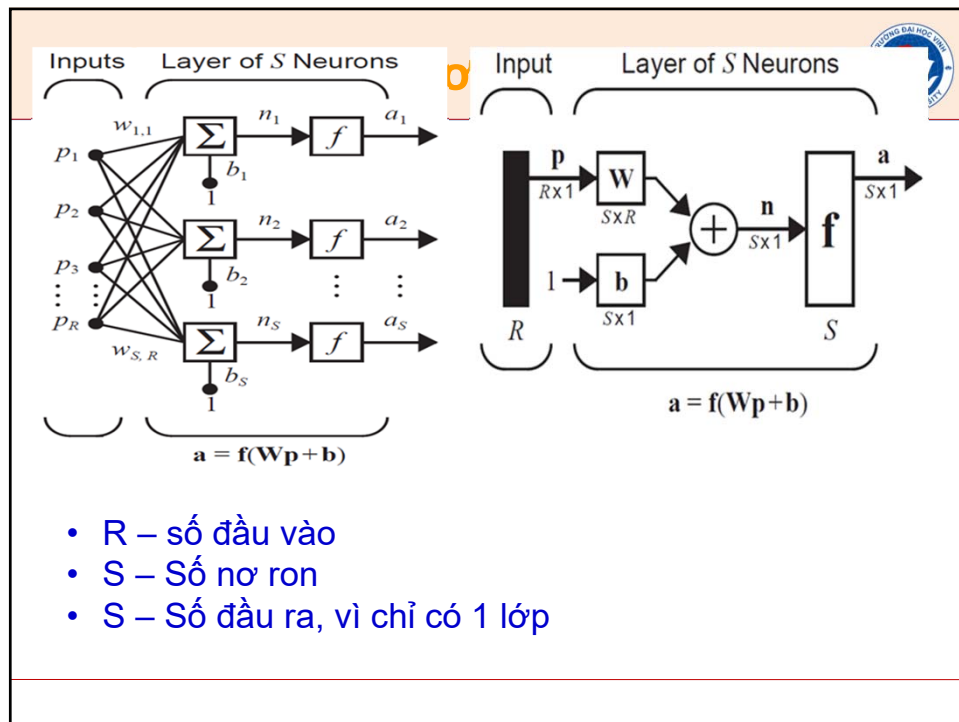
## Cấu trúc mạng nơ ron một lớp



- Cấu trúc của một mạng nơ-ron một lớp gồm  $S$  nơ-ron



24



25

## Cấu trúc mạng nơ ron một lớp

- Ví dụ về một mạng nơ-ron một lớp với 3 đầu vào ( $R=3$ ) và 2 nơ-ron đầu ra ( $S=2$ ).
- Cấu trúc mạng
  - Đầu vào:  $p_1, p_2, p_3$  (ví dụ:  $p = [0.5, -0.1, 0.8]$ )
  - Trọng số:
    - Nơ-ron 1:  $w_{1,1} = 0.2, w_{1,2} = -0.3, w_{1,3} = 0.5$
    - Nơ-ron 2:  $w_{2,1} = 0.4, w_{2,2} = 0.1, w_{2,3} = -0.6$
  - Bias:  $b_1 = 0.1, b_2 = -0.2$
  - Hàm kích hoạt:  $f(x) = \text{sigmoid}(x) = 1/(1+e^{(-x)})$

26

## Cấu trúc mạng nơ-ron 1 lớp



- Ví dụ một mạng nơ-ron một lớp với 3 đầu vào ( $R=3$ ) và 2 nơ-ron đầu ra ( $S=2$ ).
- **Biểu diễn dưới dạng ma trận:**
  - Vector đầu vào:  $p = [0.5, -0.1, 0.8]^T$
  - Ma trận trọng số:  
$$W = \begin{bmatrix} 0.2 & -0.3 & 0.5 \\ 0.4 & 0.1 & -0.6 \end{bmatrix}$$
  - Vector bias:  $b = [0.1, -0.2]^T$
  - Vector đầu ra:  $a = f(Wp + b) = [0.65, 0.38]^T$

27

## Cấu trúc mạng nơ-ron một lớp



- Ví dụ một mạng nơ-ron một lớp với 3 đầu vào ( $R=3$ ) và 2 nơ-ron đầu ra ( $S=2$ ).
- **Mạng trên thực hiện ánh xạ từ không gian 3 chiều sang không gian 2 chiều, sử dụng hàm kích hoạt sigmoid**

28

## Thảo luận

