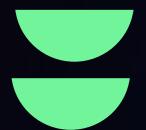
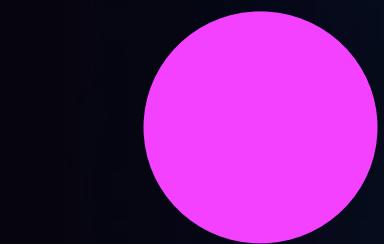




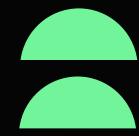
ARTIFICIAL NEURAL NETWORK



GVHD: Th.S Trần Trọng Bình
Nhóm 1: SUONG SUONG



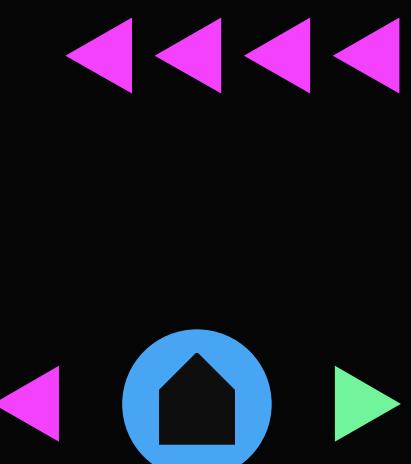
Thành viên nhóm



Nguyễn Chí Tâm	-	20133087
Huỳnh Hạo Nghị	-	20133006
Võ Hoài Thương	-	20133012
Trương Thị Thu Sương	-	20133086



NỘI DUNG BAO GỒM

- 
- 01 Giới thiệu chung về ANN
 - 02 Các khái niệm cơ bản về ANN
 - 03 Thuật toán huấn luyện ANN
 - 04 Các bước xây dựng trong mạng ANN
 - 05 Ứng dụng
- 

01

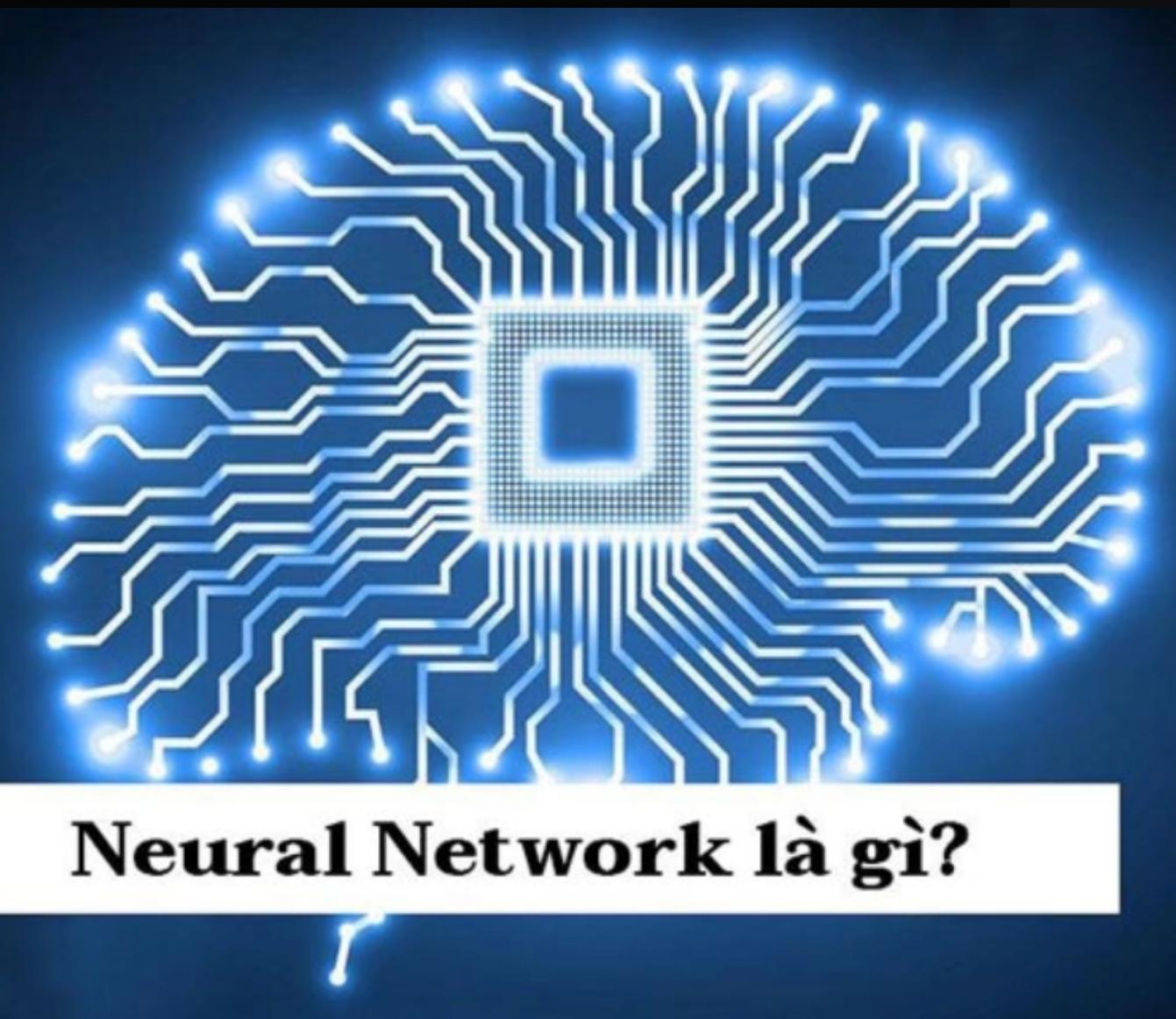
Giới thiệu chung về ANN

Định nghĩa
Các thành phần cơ bản



Định nghĩa

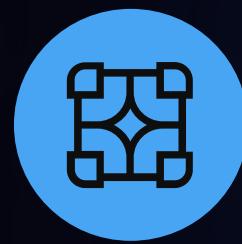
Artificial Neural Network (ANN) là mạng neural nhân tạo và là mô hình toán học được lấy ý tưởng từ các neural sinh học.



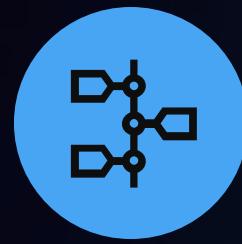
Các thành phần chính



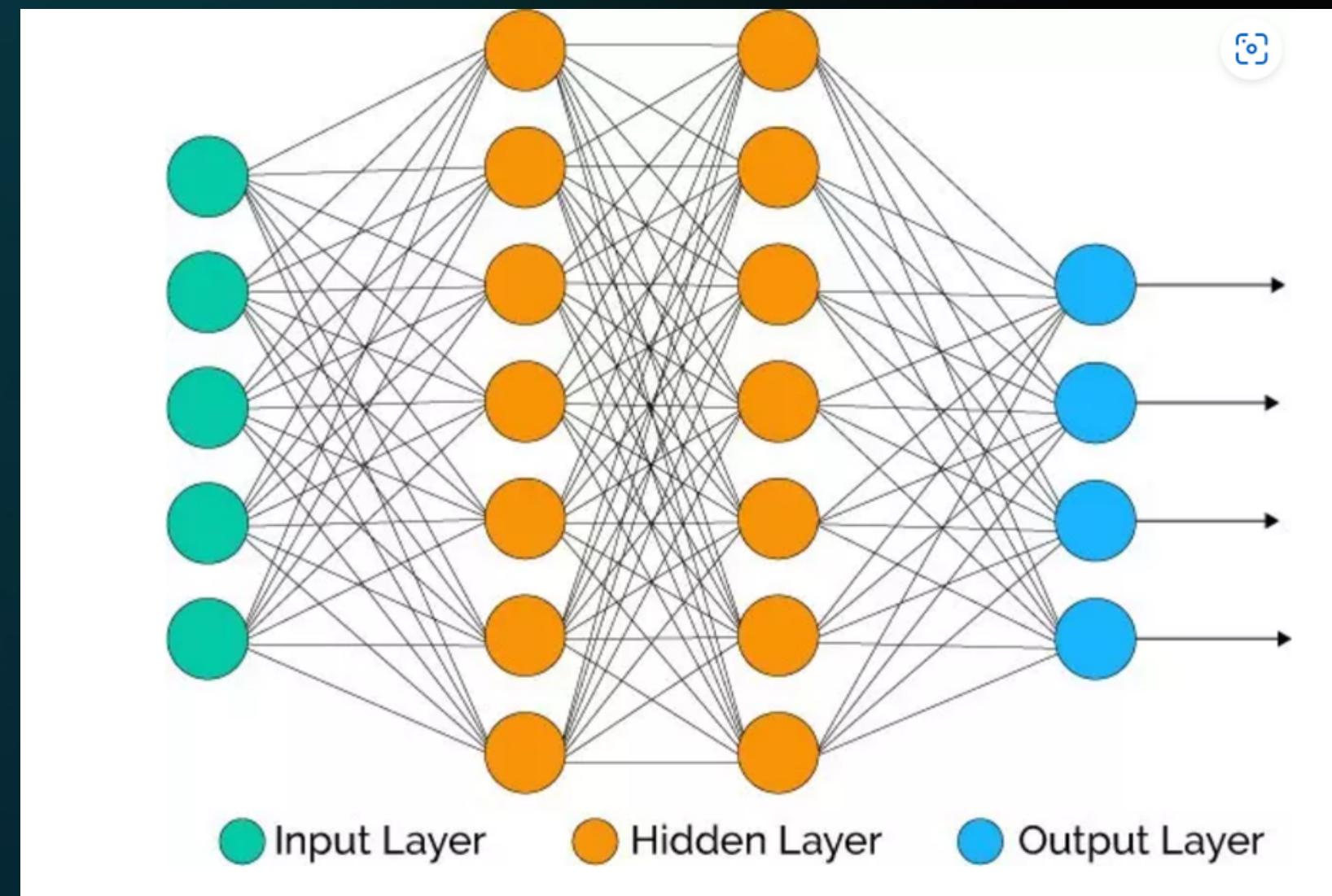
Input layer



Output layer



Hidden layer





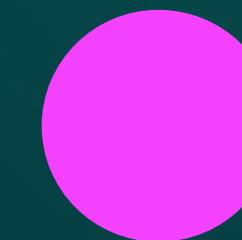
Một số khái niệm cơ bản



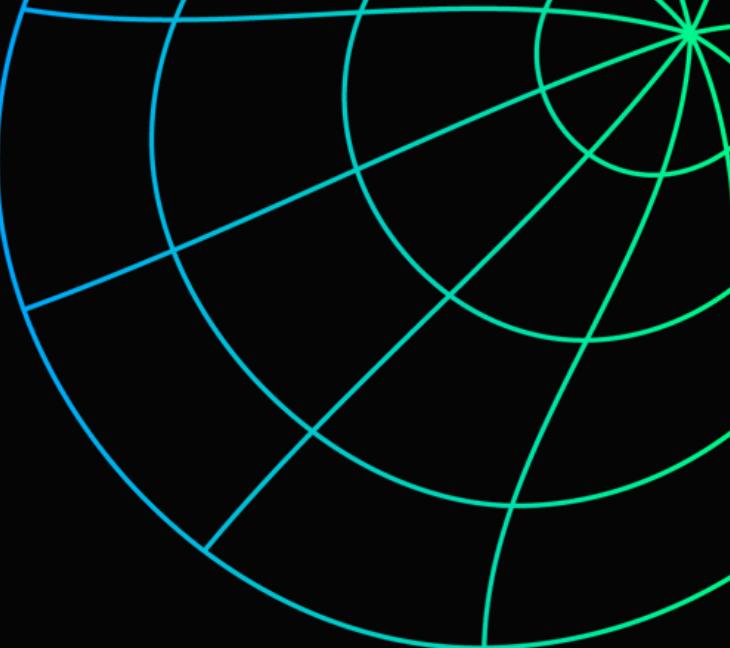


Hàm kích hoạt

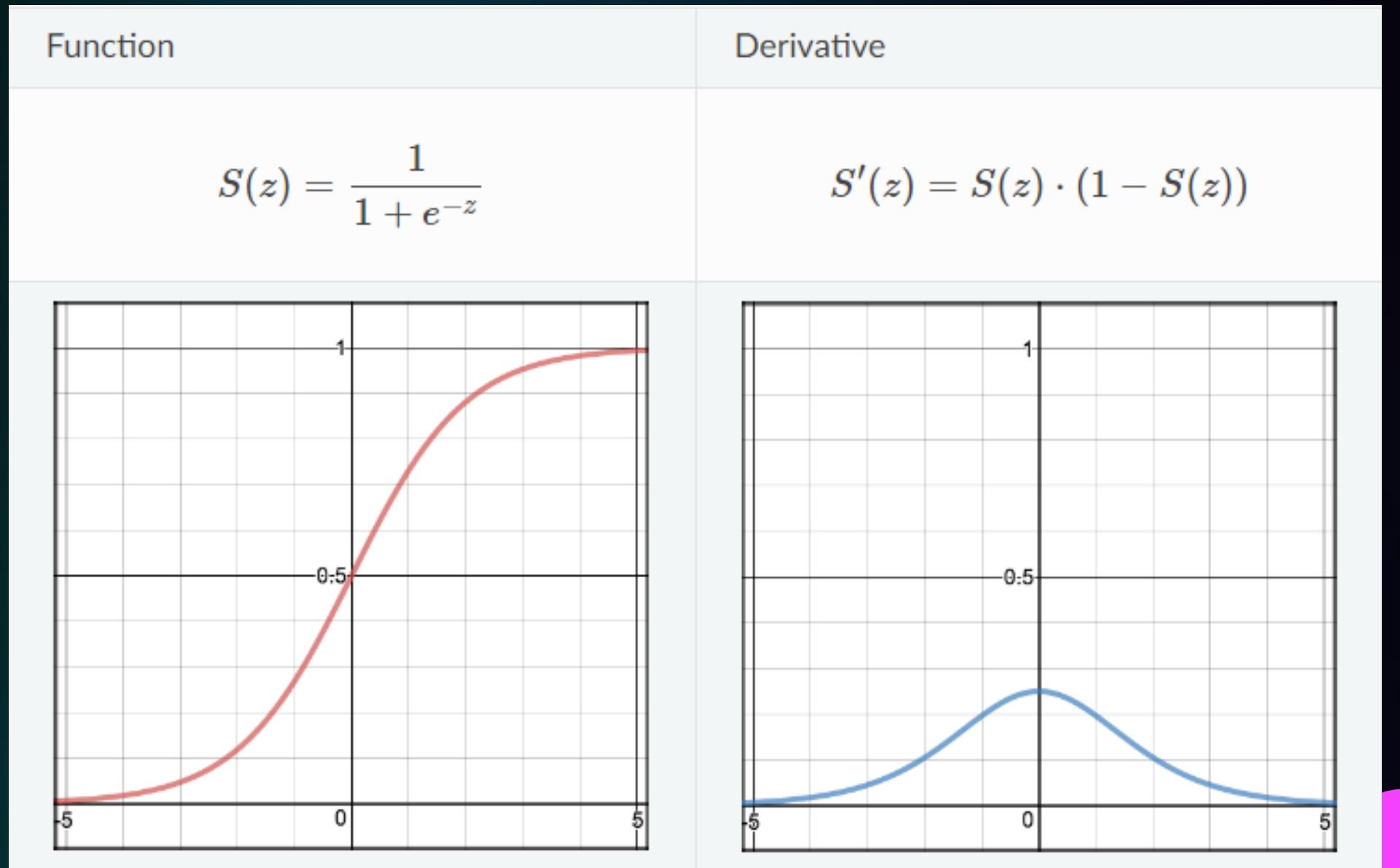
Hàm kích hoạt thường là một hàm phi tuyến, nó quyết định một nơ-ron có được kích hoạt hay không bằng cách tính tổng có trọng số và cộng thêm hệ số điều chỉnh.



Một số hàm kích hoạt thường gặp



Hàm sigmoid

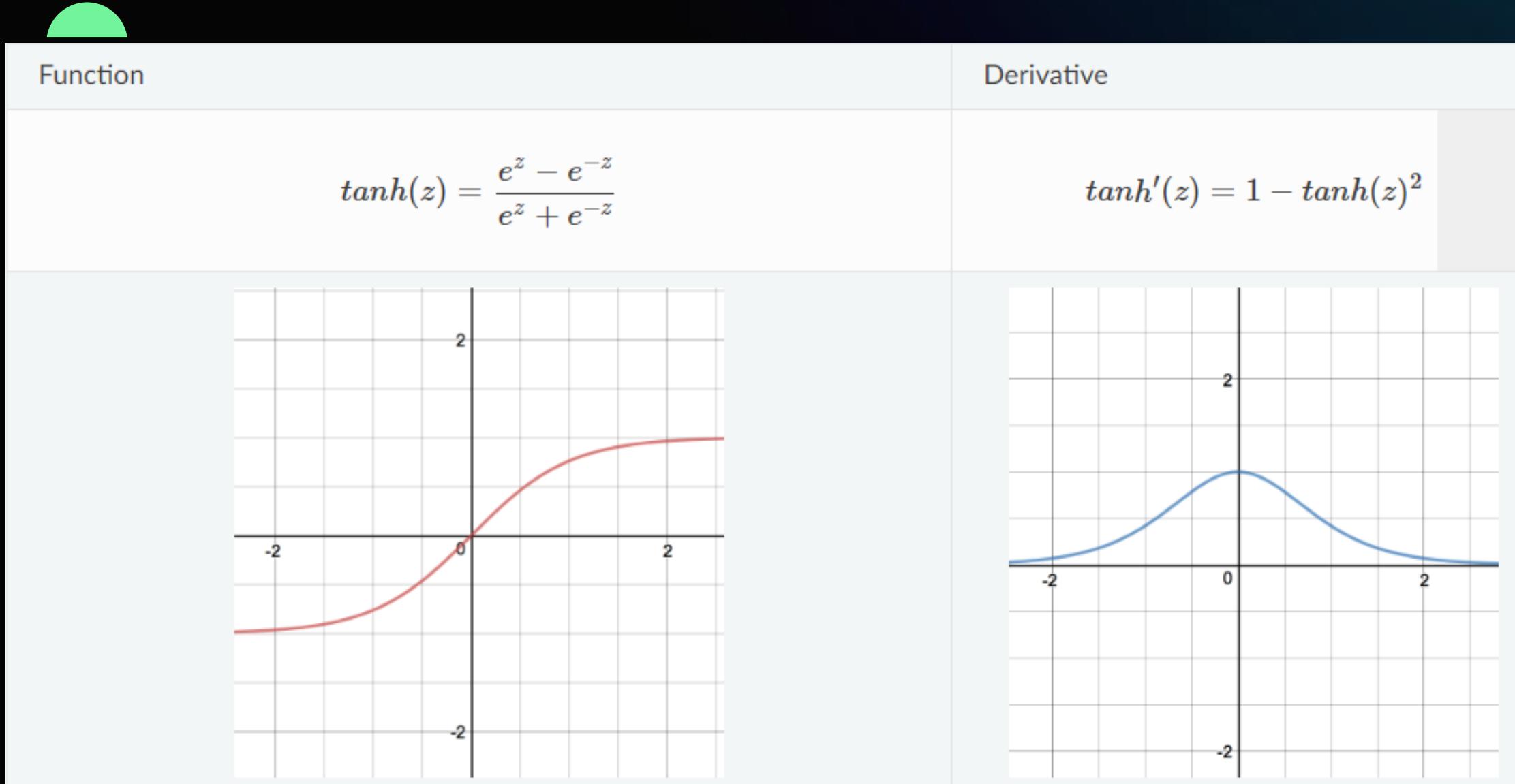


- **Ưu điểm:**
 - Cho kết quả tốt cho bài toán phân loại (phân loại nhị phân)
 - Đầu ra của hàm kích hoạt luôn nằm trong khoảng (0,1) so với (-inf, inf) của hàm tuyến tính. Giá trị của nó ràng buộc trong phạm vi nhất định.
- **Nhược điểm:**
 - Nó làm nảy sinh vấn đề “Vanishing gradient”.



Một số hàm kích hoạt thường gặp

Tanh



Ưu điểm:

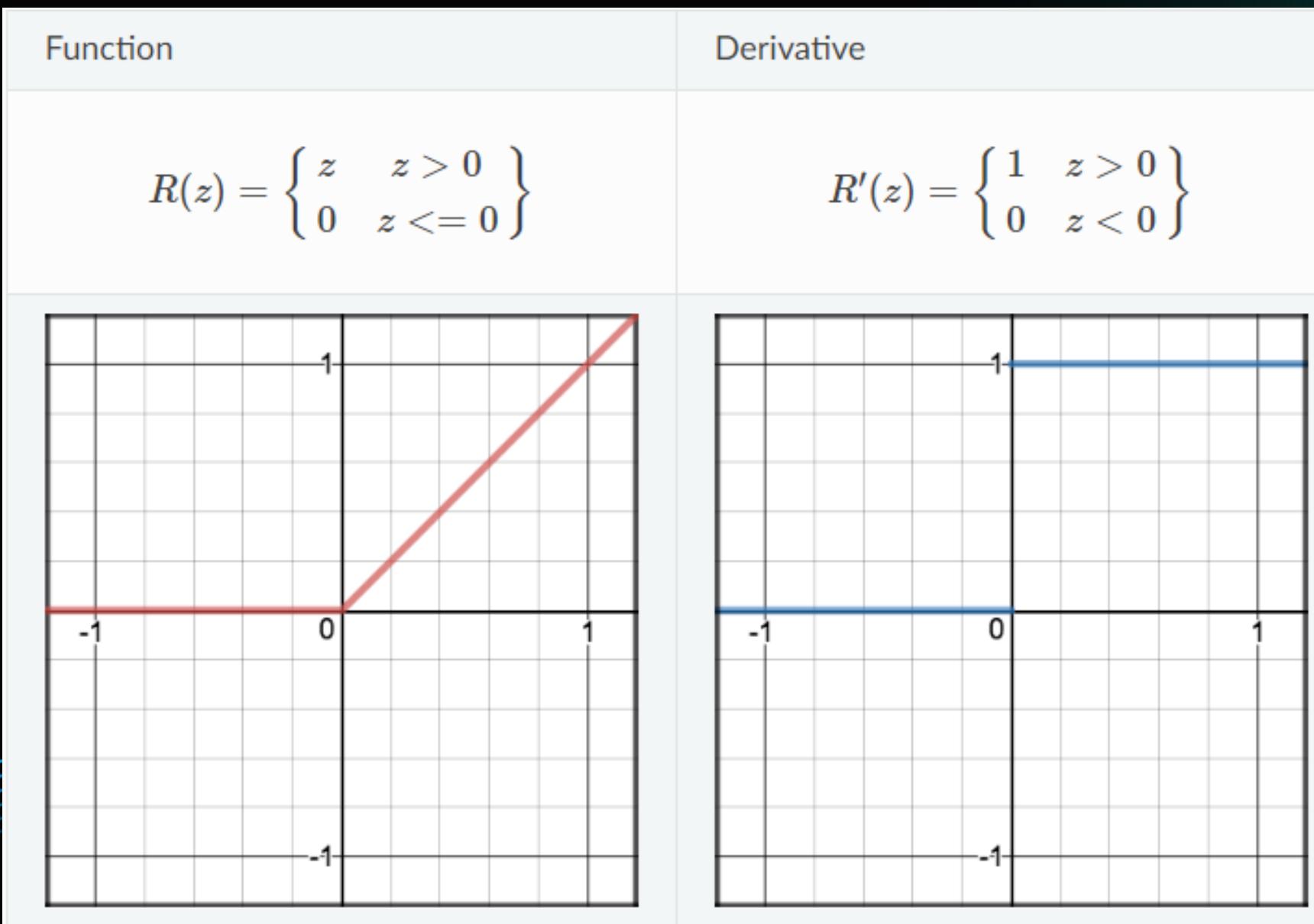
- Giá trị nằm trong đoạn $[-1, 1]$
- Không giống như Sigmoid, có giá trị trung tâm là 0. Trong thực tế, hàm tanh luôn được ưu tiên hơn so với hàm sigmoid.

Nhược điểm:

- Gặp vấn đề "vanishing gradient"

Một số hàm kích hoạt thường gặp

ReLU (Rectified Linear Units)



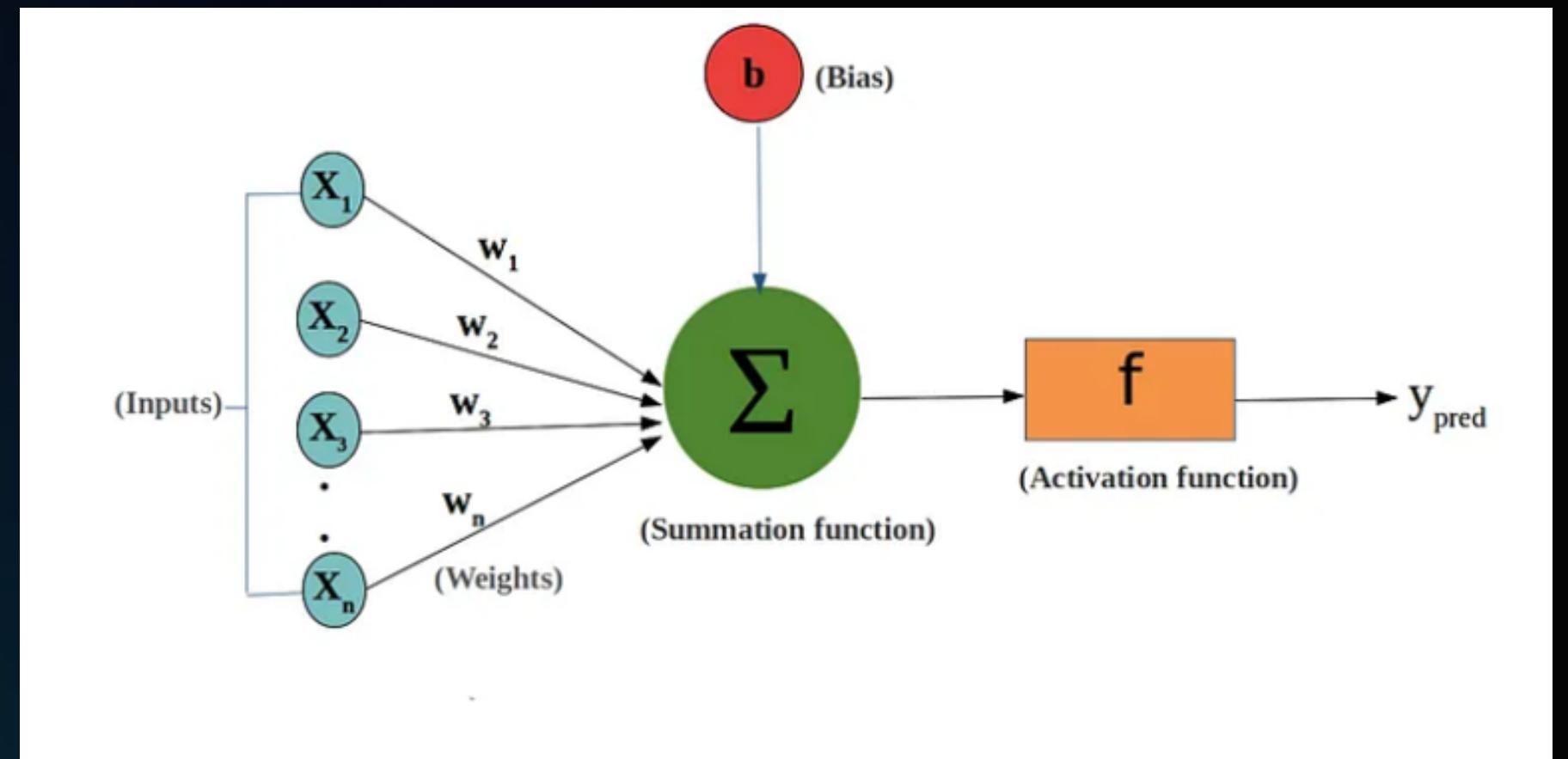
Ưu điểm:

- ReLu ít tốn kém hơn về mặt tính toán so với tanh và sigmoid vì nó phép toán đơn giản của nó.
- Khắc phục được vấn đề "vanishing gradient".

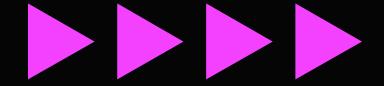
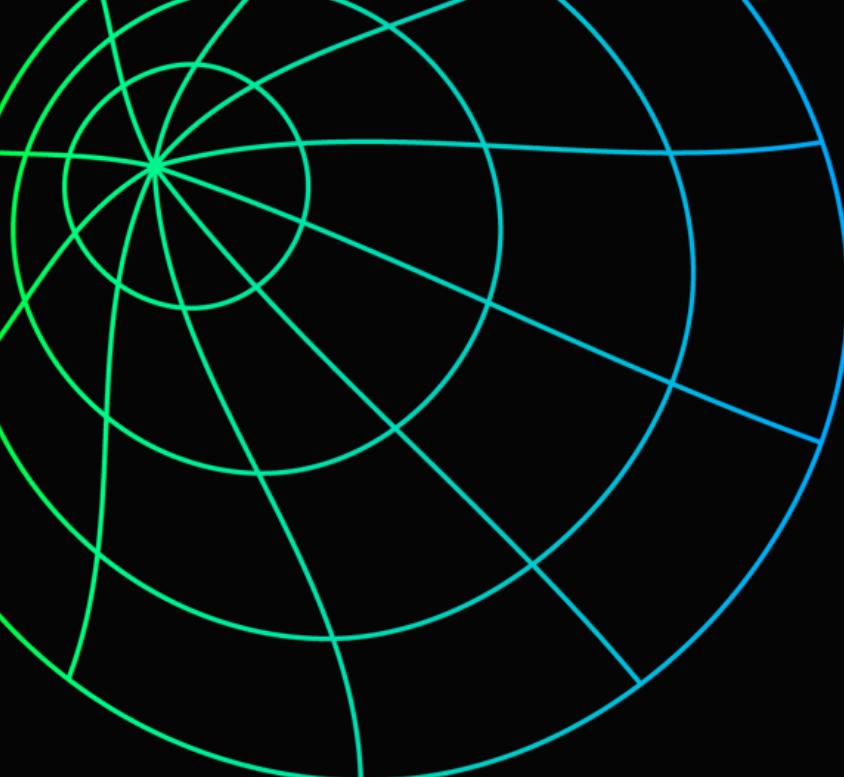
Nhược điểm:

- Chỉ được sử dụng trong các lớp ẩn của mô hình mạng thần kinh.
- Phạm vi của ReLu là $[0, \infty)$. Điều này có nghĩa là nó có thể "nổ tung" hàm kích hoạt.
- Ngưng phản ứng với các lối đầu vào do các giá trị $x < 0$ thì độ dốc sẽ bằng 0 (không có gì thay đổi)

What is weight and bias in a neural network?



- Trọng số: trọng số là các giá trị thực được đính kèm với mỗi đầu vào thể hiện tầm quan trọng của tính năng tương ứng đó trong việc dự đoán đầu ra cuối cùng.
- Bias Độ lệch chỉ đơn giản là một giá trị không đổi (hoặc một vectơ không đổi) được thêm vào tích của đầu vào và trọng số. Xu hướng được sử dụng để bù đắp kết quả.



03

Thuật toán

Gradient descent
Backpropagation



Gradient descent

THUẬT TOÁN:

Repeat:

Compute predictions ($y'[i]$, $i = 0, \dots, m$)

Get derivatives: $dW1$, $db1$, $dW2$, $db2$

Update:

$W1 = W1 - LearningRate * dW1$

$b1 = b1 - LearningRate * db1$

$W2 = W2 - LearningRate * dW2$

$b2 = b2 - LearningRate * db2$

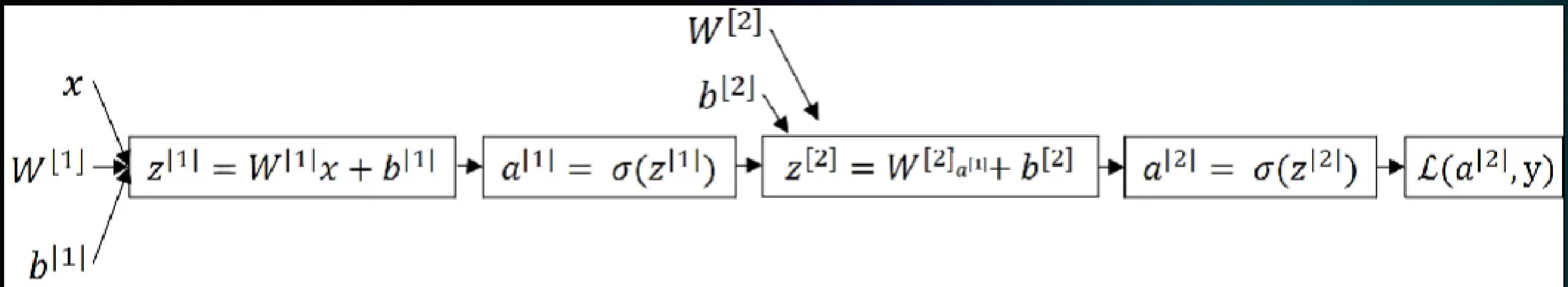
- $n[0] = Nx$
- $n[1] = Số Neuron ẩn$
- $n[2] = Số Neuron đầu ra = 1$
- $W1$ hình dạng là $(n[1], n[0])$
- $b1$ hình dạng là $(n[1], 1)$
- $W2$ hình dạng là $(n[2], n[1])$
- $b2$ hình dạng là $(n[2], 1)$





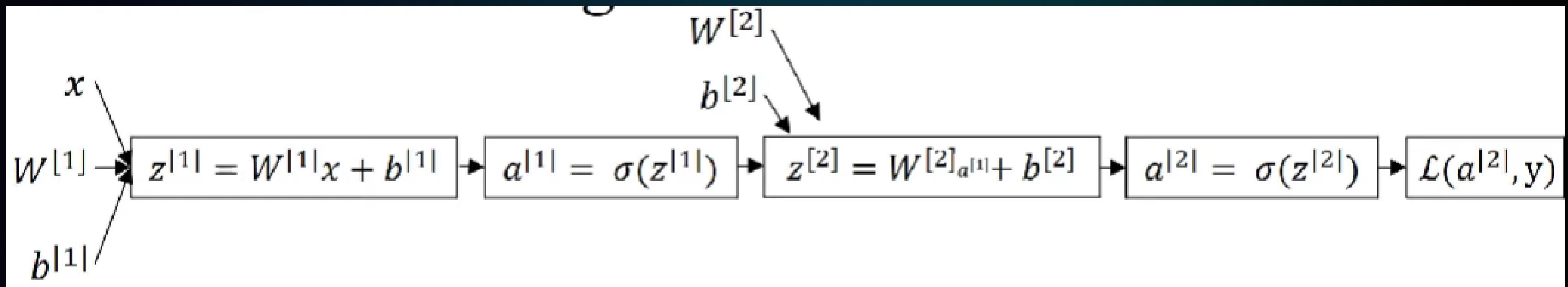
Feedforward

Feedforward là kiểu kết nối giữa các nơ-ron trong mạng neural network, trong đó các tín hiệu được truyền từ input layer qua các lớp hidden layer và output layer mà không có sự phản hồi từ output layer trở lại input layer.

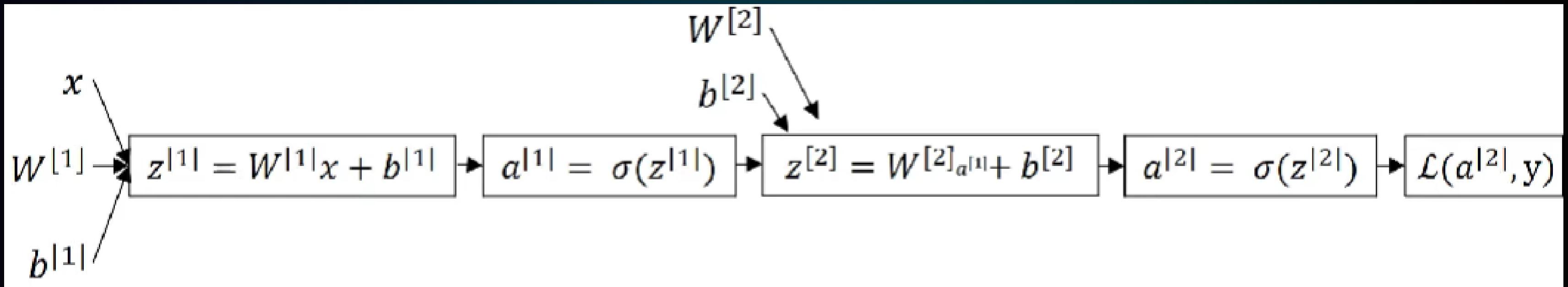


Backpropagation

Backpropagation (lan truyền ngược): Thuật toán này hoạt động bằng cách lan truyền ngược lại từ đầu ra (output) của mạng đến đầu vào (input), tính toán gradient theo từng trọng số ở mỗi lớp và sử dụng gradient này để điều chỉnh các trọng số của mạng bằng thuật toán gradient descent.



Backpropagation



$$dz^{[2]} = a^{[2]} - y$$

$$dz^{[1]} = W^{[2]T} dz^{[2]} * g^{[1]'}(z^{[1]})$$

$$dW^{[2]} = dz^{[2]} a^{[1]T}$$

$$dW^{[1]} = dz^{[1]} x^T$$

$$db^{[2]} = dz^{[2]}$$

$$db^{[1]} = dz^{[1]}$$

Forward and Backward propagation

$$Z^{[1]} = W^{[1]}X + b^{[1]}$$

$$A^{[1]} = g^{[1]}(Z^{[1]})$$

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = g^{[2]}(Z^{[2]})$$

⋮

$$A^{[L]} = g^{[L]}(Z^{[L]}) = Y$$

Forward propagation



$$dz^{[2]} = a^{[2]} - y$$

$$dW^{[2]} = dz^{[2]}a^{[1]T}$$

$$db^{[2]} = dz^{[2]}$$

$$dz^{[1]} = W^{[2]T}dz^{[2]} * g^{[1]'}(z^{[1]})$$

$$dW^{[1]} = dz^{[1]}x^T$$

$$db^{[1]} = dz^{[1]}$$

Backward propagation



04
:



Các bước xây dựng mô hình ANN



Các bước xây dựng mô hình ANN

1. Xác định tập dữ liệu và cấu trúc tập dữ liệu cần huấn luyện
2. Xác định kích thước đầu vào và kích thước đầu ra của mô hình
3. Xác định số lớp và số node của mỗi lớp
4. Xác định hàm mất mát.
5. Xác định thuật toán tối ưu
6. Xác định các tham số cần thiết: epoch, learning rate, ...





Demo



THANKS

Any questions ?

