



CMC UNIVERSITY

Aspire to Inspire the Digital World

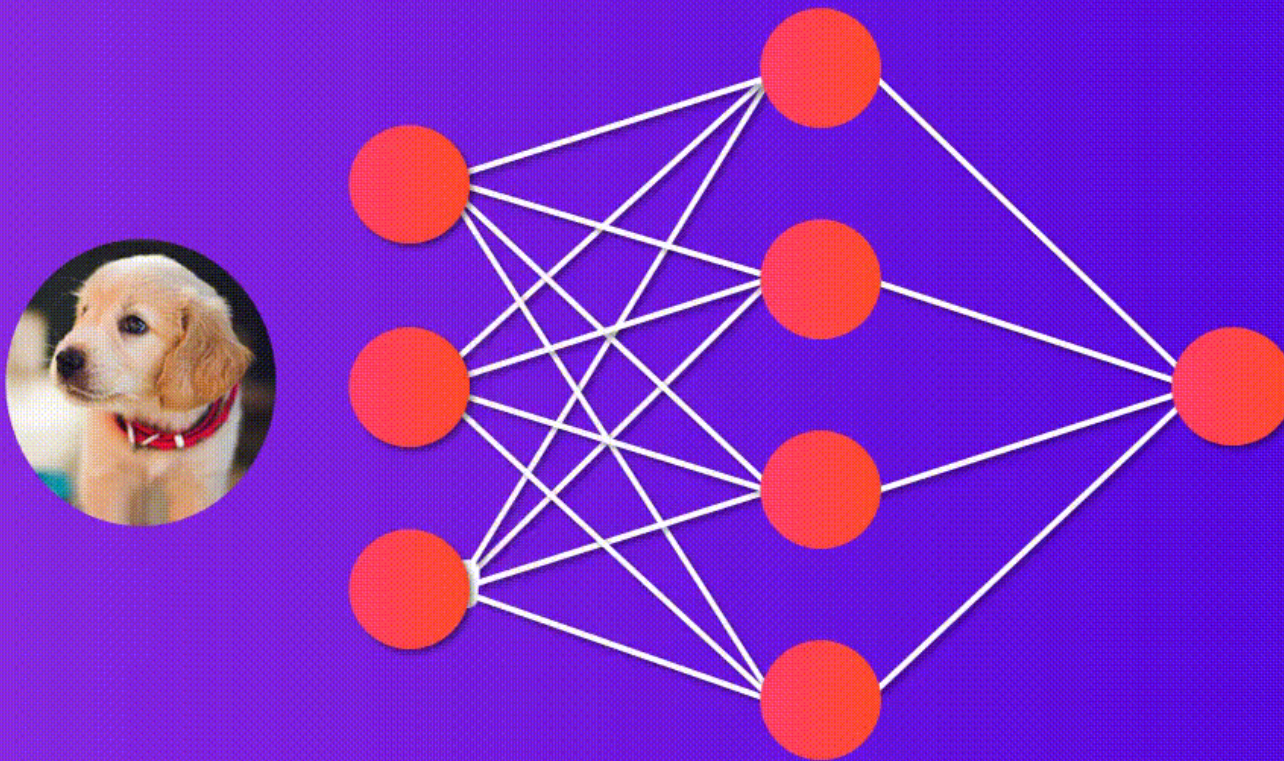
TRÍ TUỆ NHÂN TẠO

Mạng nơ ron nhân tạo (Artificial Neural Network)

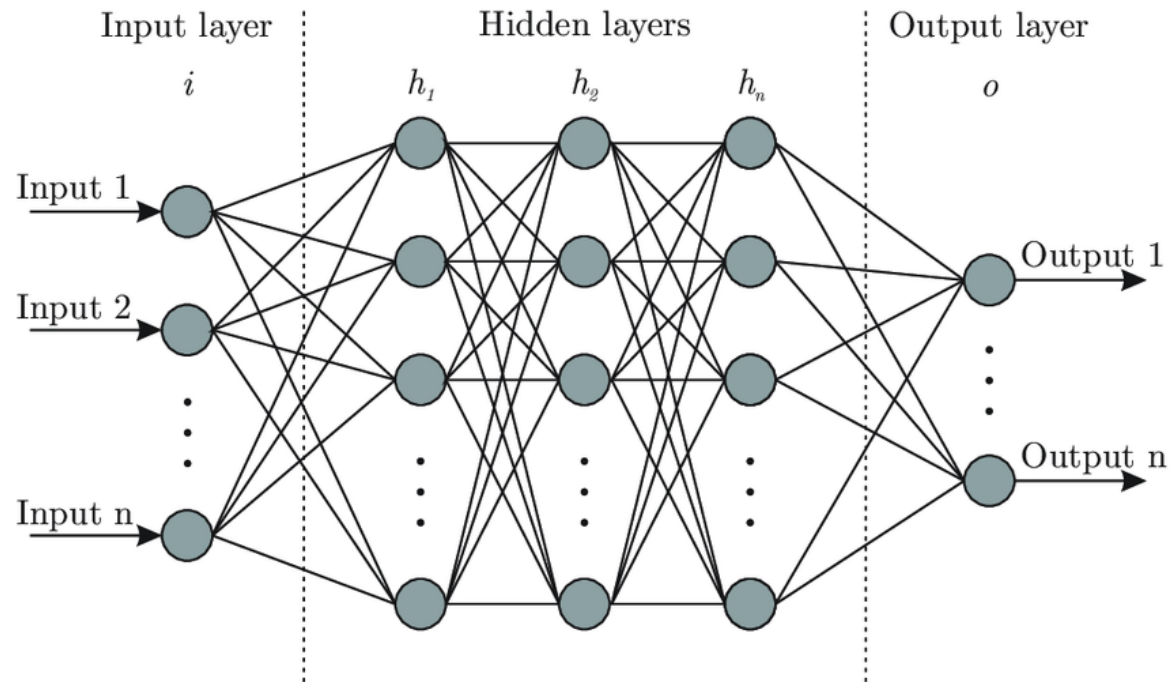
Hà Nội, tháng 09 năm 2024

1. Mạng Nơ ron nhân tạo
2. Kiến trúc mạng nơ ron
3. Học trong mạng nơ ron
4. Triển khai sử dụng mạng nơ ron

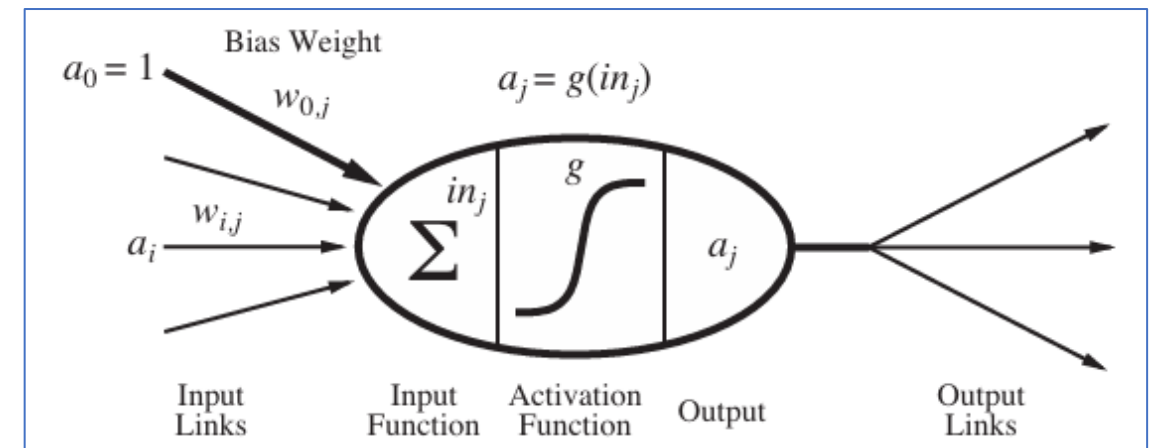
- Nó là một hệ thống tính toán mà được lấy cảm hứng từ cấu trúc và chức năng của não người.
- Nó gồm các lớp (layer) tế bào thần kinh nhân tạo được kết nối với nhau, mà xử lý và truyền thông tin thông qua các kết nối có trọng số
- Nó được thiết kế để học các mẫu và các quan hệ trong dữ liệu, làm cho chúng đặc biệt hiệu quả cho các nhiệm vụ:
 - Nhận dạng ảnh,
 - Xử lý ngôn ngữ tự nhiên
 - Nhận dạng tiếng nói,...



- Mô phỏng mô hình giống như hoạt động của Nơ ron thần kinh

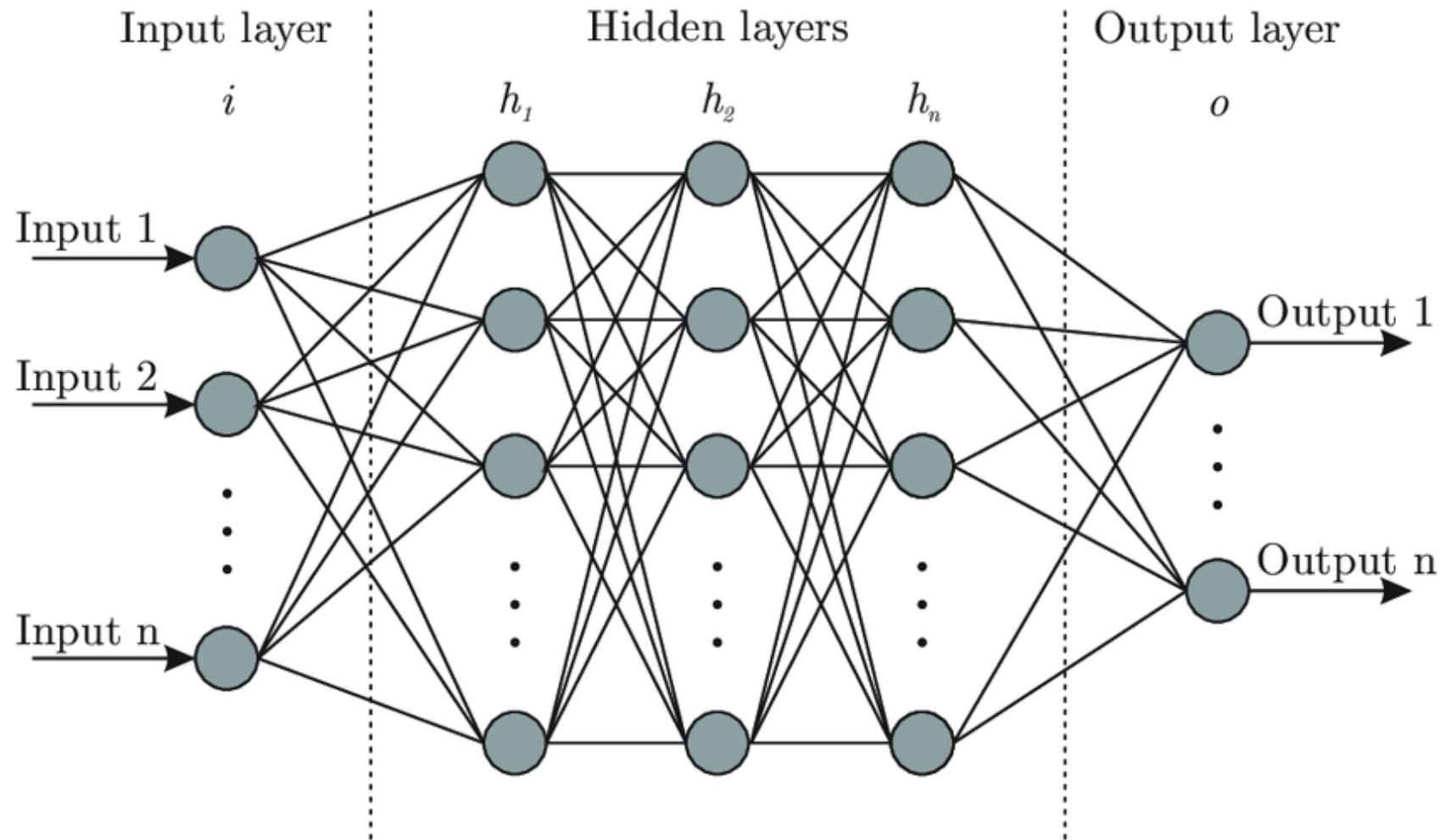


Ví dụ về mạng Nơ ron



Cấu trúc của một nơ ron

- Ví dụ về mạng Nơ ron



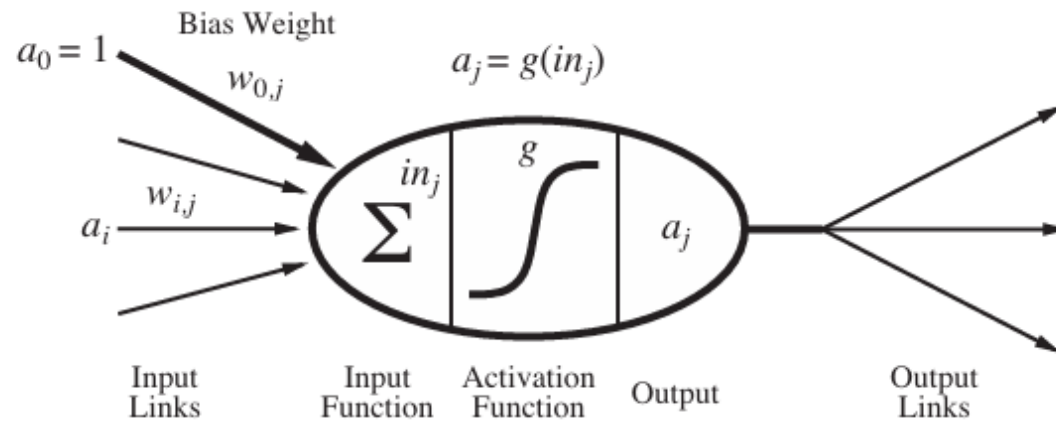
- Nó học bằng việc điều chỉnh các trọng số của các kết nối dựa vào các sai số nó mắc phải trong quá trình học (Quá trình này gọi là lan truyền ngược – backpropagation)
- Khi mạng nơ ron trở nên sâu hơn với nhiều lớp và nhiều nơ ron hơn, nó có thể học được các hàm phức tạp hơn, dẫn đến các mô hình học sâu cho kết quả vượt trội trong nhiều ứng dụng của AI.

- Các công ty công nghệ như Google, Facebook, và Amazon sử dụng các mạng nơ ron cho nhiều ứng dụng:
 - search engines,
 - social media algorithms,
 - recommendation systems.
- Công nghiệp chăm sóc sức khỏe: Các bệnh viện, các nhà nghiên cứu, và các nhà máy thiết bị y tế sử dụng mạng nơ ron cho các nhiệm vụ
 - medical imaging,
 - diagnostics,
 - personalized treatment planning.

- Các tổ chức tài chính: Các ngân hàng, công ty đầu tư và công ty bảo hiểm sử dụng mạng lưới thần kinh để:
 - phát hiện gian lận,
 - đánh giá rủi ro,
 - quản lý danh mục đầu tư và
 - dự báo tài chính.
- Thương mại điện tử và bán lẻ: Các nhà bán lẻ và nền tảng thương mại điện tử tận dụng mạng lưới thần kinh để
 - tối ưu hóa chuỗi cung ứng,
 - phân khúc khách hàng,
 - tiếp thị mục tiêu và
 - dự báo doanh số.

- Công nghiệp ô tô: các nhà sản xuất ô tô và các nhà phát triển ô tô tự lái sử dụng mạng nơ ron cho các nhiệm vụ:
 - Phát hiện vật thể,
 - Lập kế hoạch đường đi
 - Ra quyết định trong ô tô tự lái
- Các nhà nghiên cứu và học thuật: sử dụng mạng nơ ron để nâng cao tri thức trong các lĩnh vực:
 - Trí tuệ nhân tạo
 - Ngôn ngữ học
 - Khoa học thần kinh
 - Tâm lý học

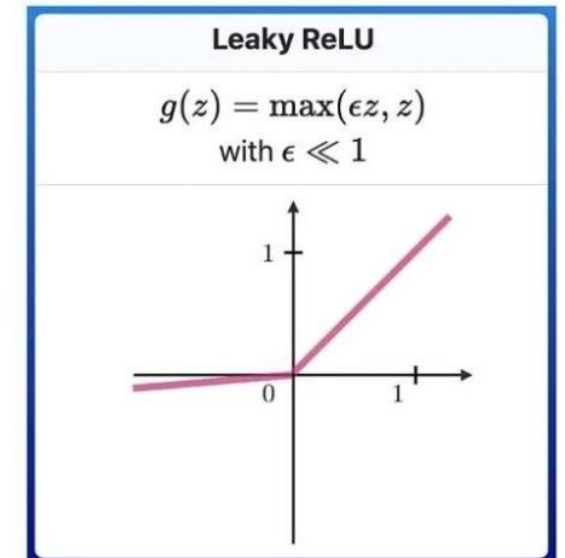
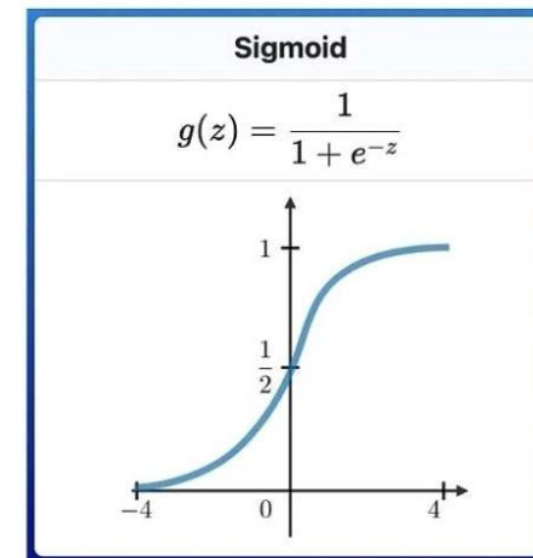
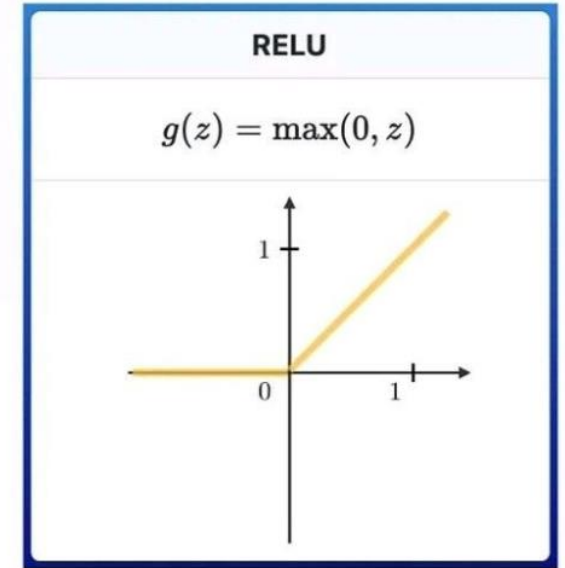
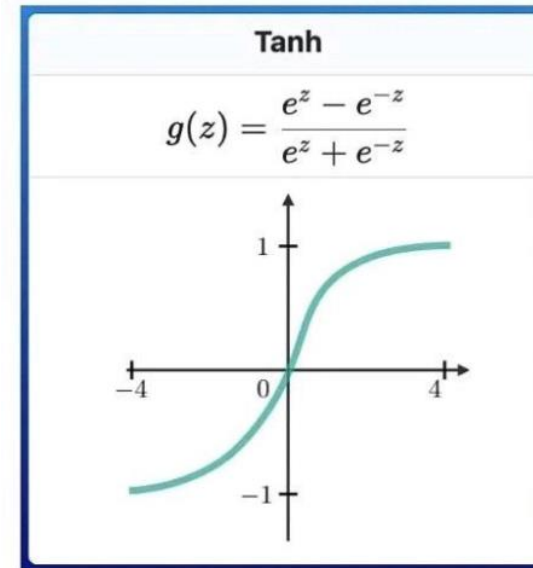
- Quá trình tính toán:



$$in_j = \sum_{i=0}^n w_{i,j} a_i$$

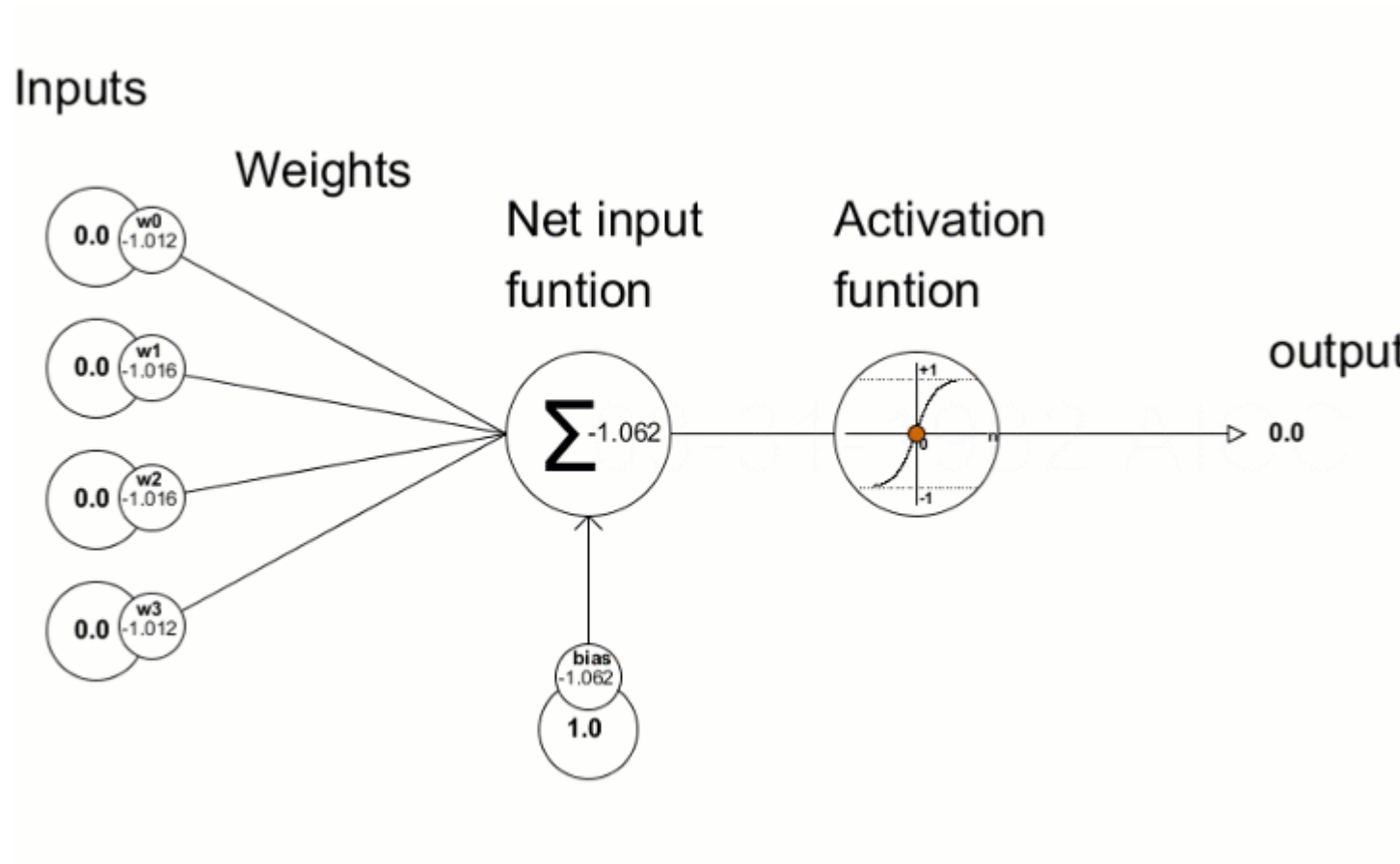
$$a_j = g(in_j) = g \left(\sum_{i=0}^n w_{i,j} a_i \right)$$

$g()$: hàm kích hoạt

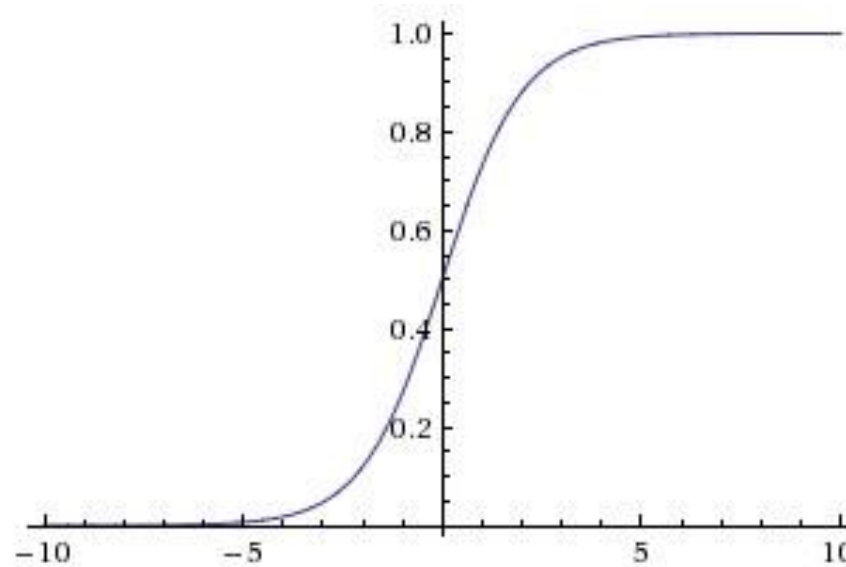


Một số hàm kích hoạt phổ biến

- Hoạt động của 1 nơ ron với hàm kích hoạt Tanh

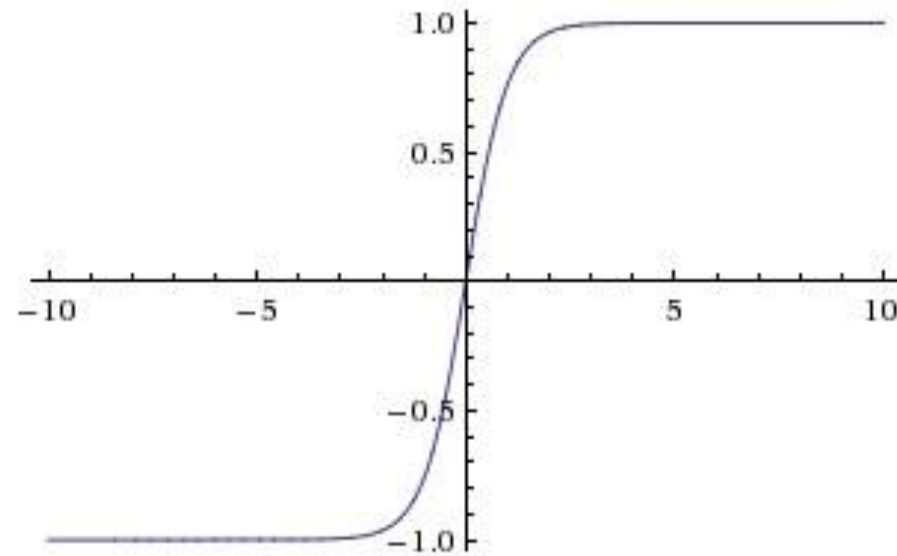


- Một số hàm kích hoạt (activation function)
Hàm sigmoid()



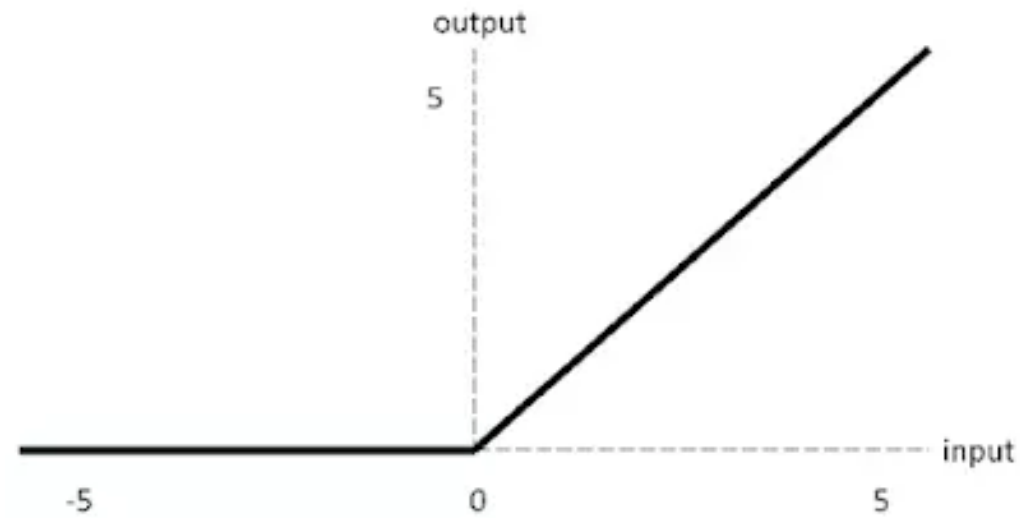
$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

- Hàm Tanh



$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

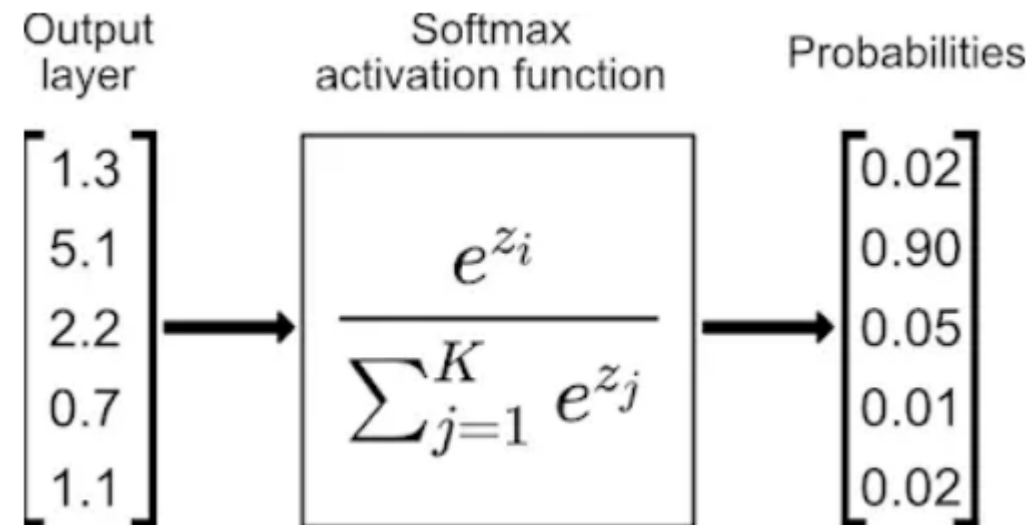
- Hàm ReLu

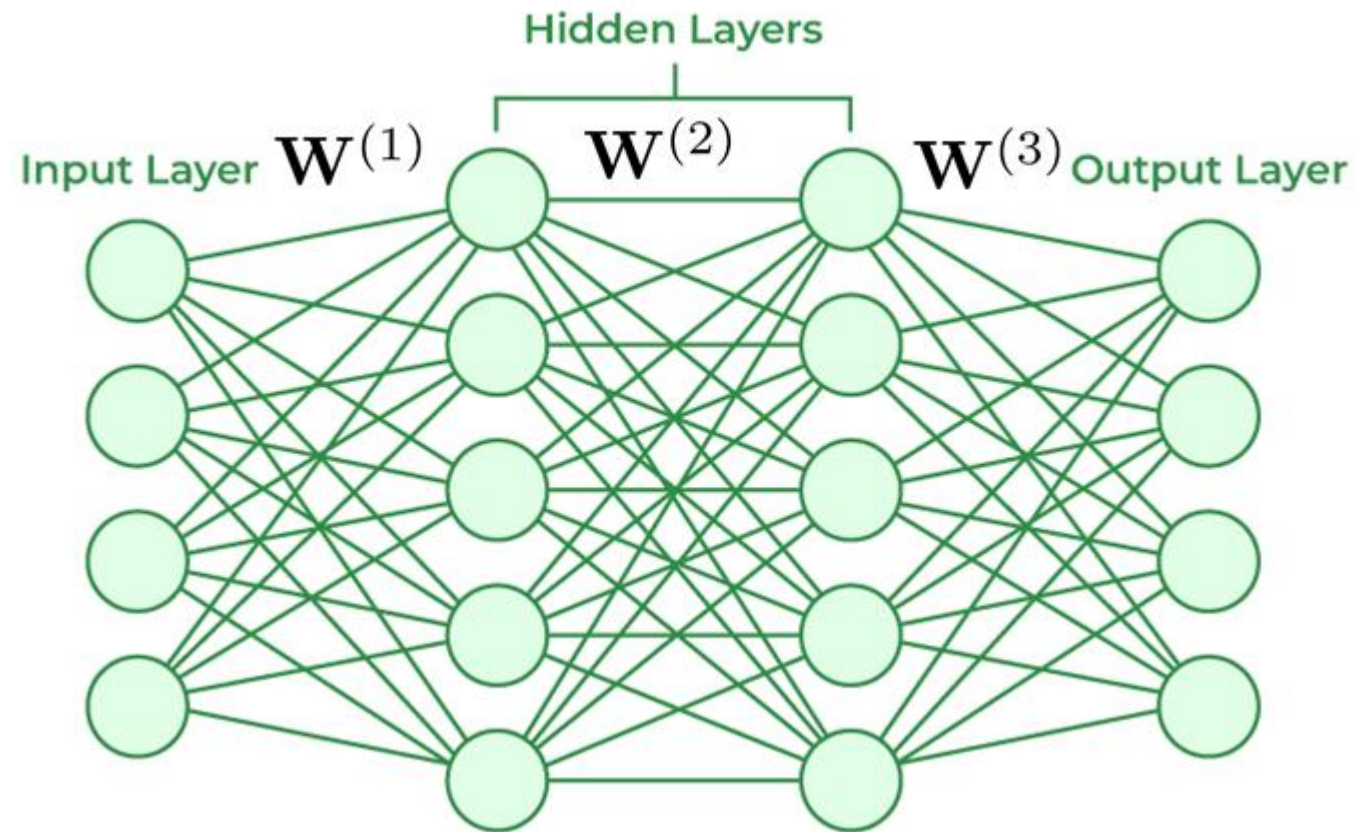


$$\text{ReLU}(x) = \max(0, x)$$

- Một số hàm kích hoạt

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_{j=1}^K \exp(z_j)}$$





- Mỗi vòng lặp của quá trình huấn luyện bao gồm hai bước:
- Tính toán đầu ra dự đoán , là **feedforward**
- Cập nhật các weight và bias, là **backpropagation**

- Sai số tổng bình phương

$$L(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

- Cross-entropy

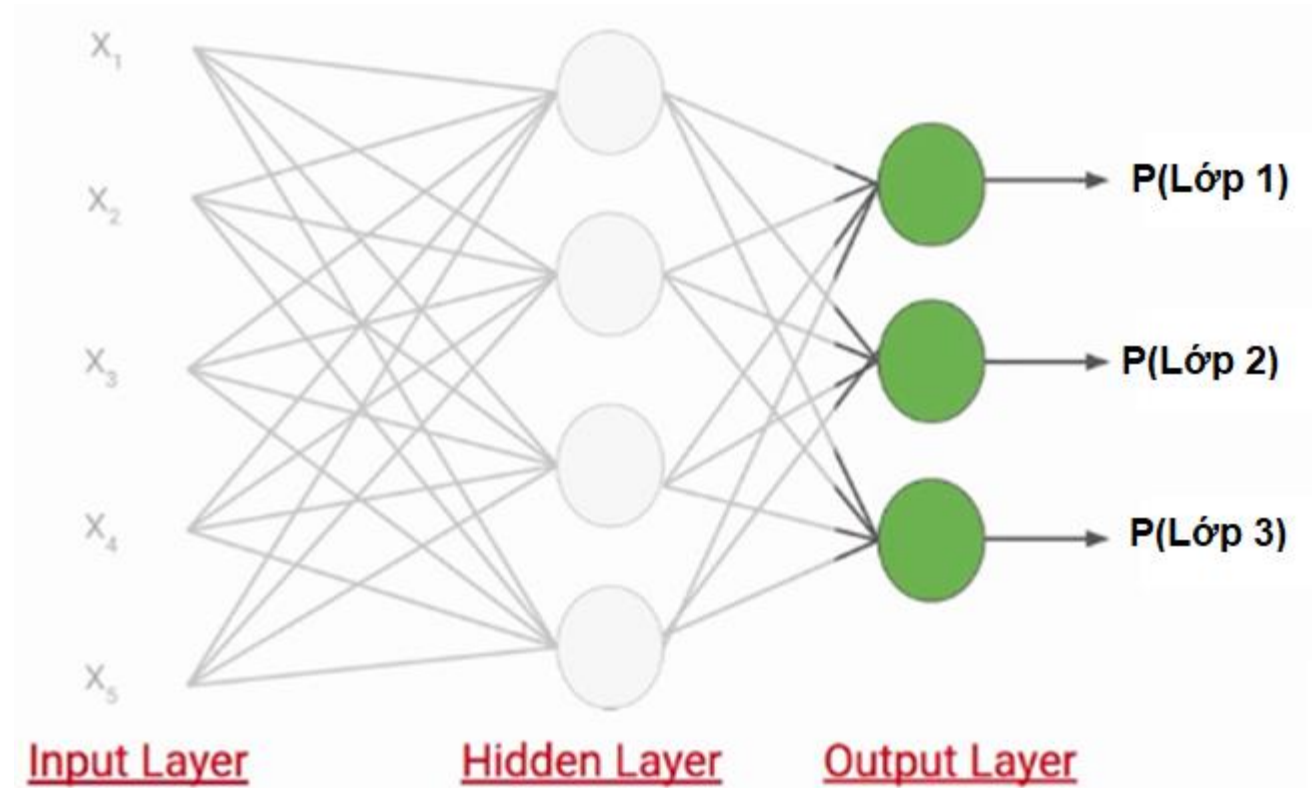
$$L(\mathbf{w}) = - \sum_{i=1}^N y_i \cdot \log \hat{y}_i$$

- Cho tập dữ liệu:

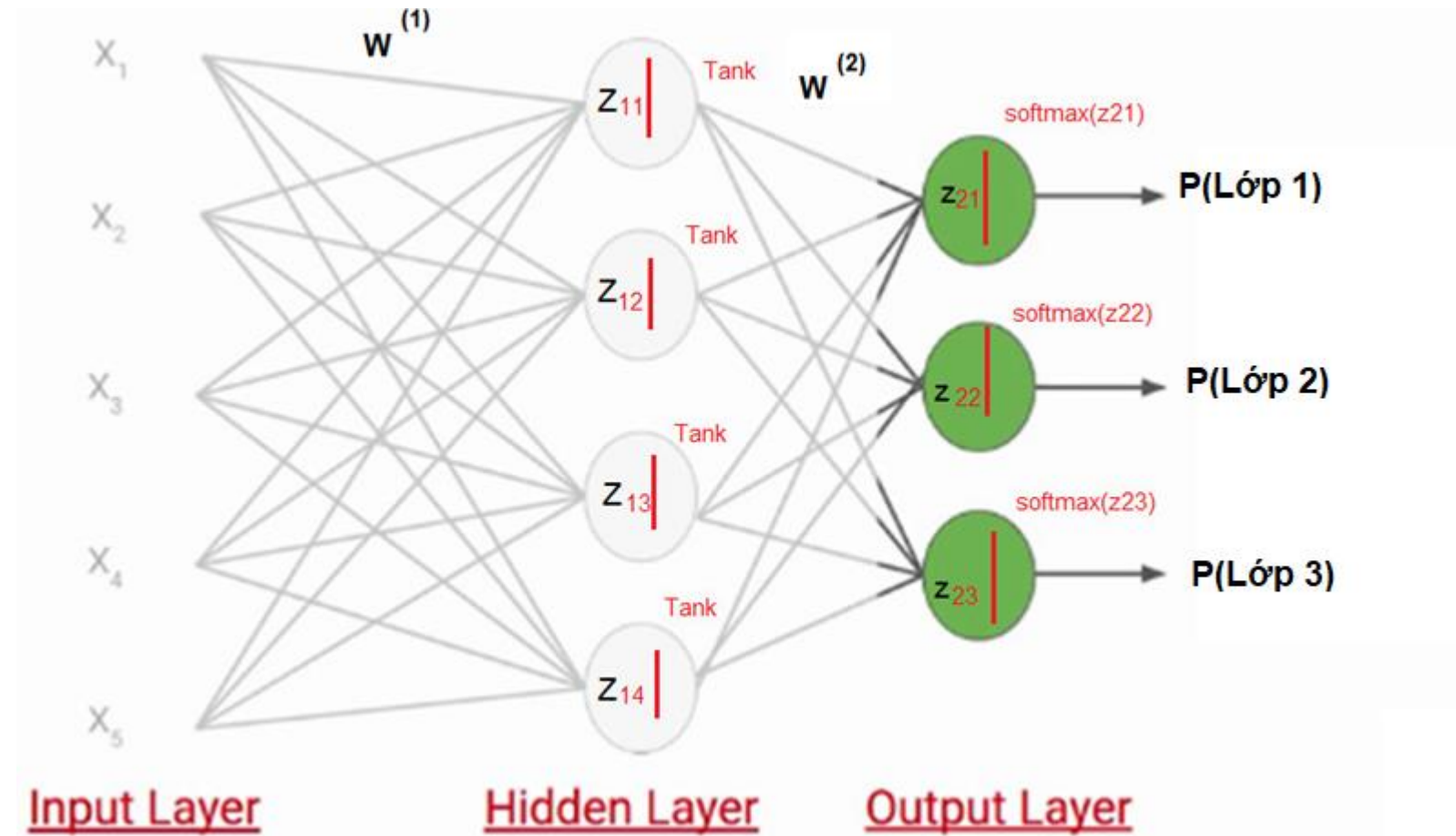
Đặc trưng x1	Đặc trưng x2	Đặc trưng x3	Đặc trưng x4	Đặc trưng x5	Nhãn
1	22	569	35	0	Lớp 1
1	7	351	75	1	Lớp 2
1	45	451	542	1	Lớp 2
1	5	572	8	0	Lớp 1
0	22	565	44	1	Lớp 3
0	24	243	546	1	Lớp 3
1	78	953	42	0	Lớp 2

- Kiến trúc mạng

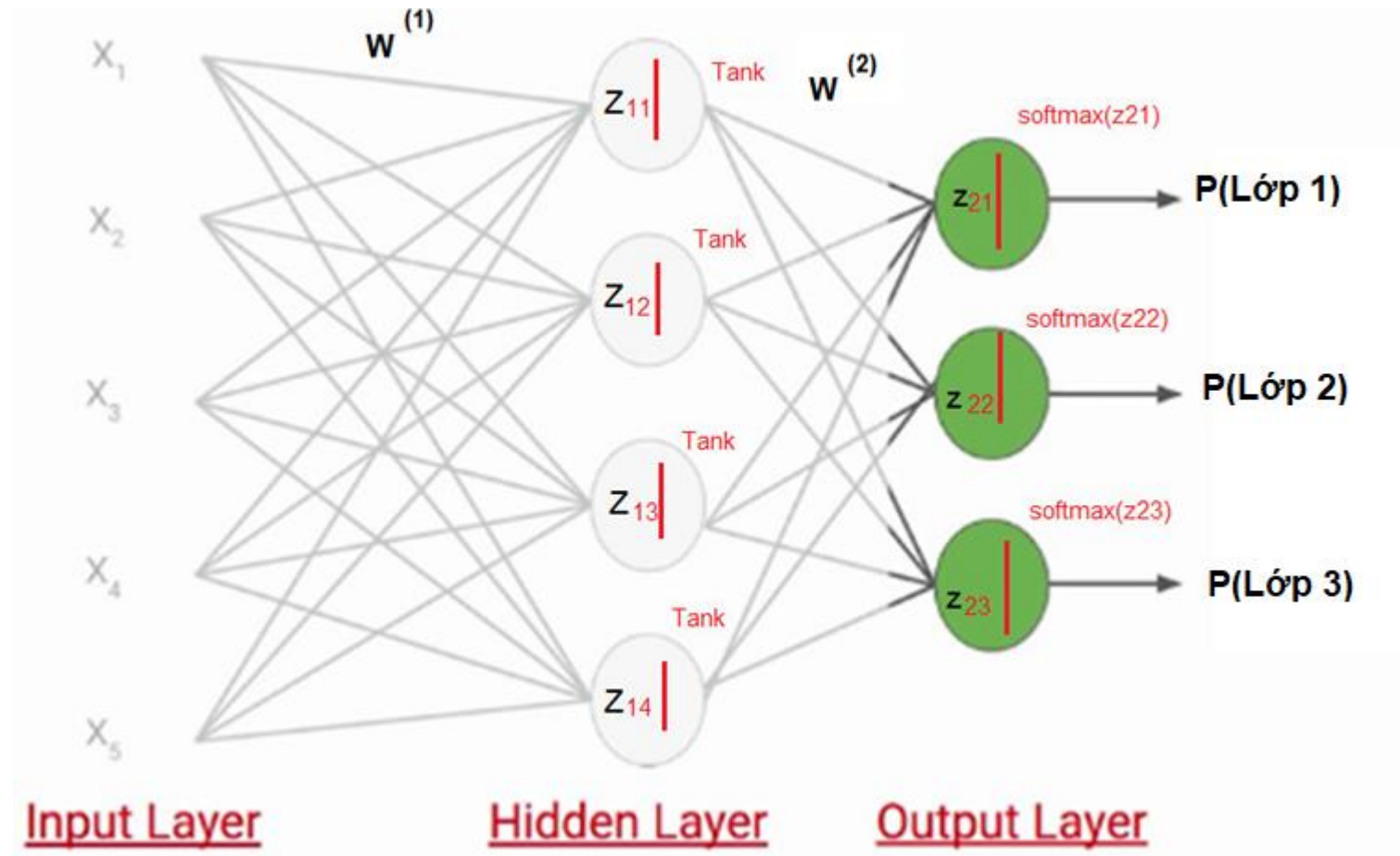
- Chúng ta có 1 lớp đầu vào gồm 5 nơ-ron bởi vì chúng ta có 5 đặc trưng trong tập dữ liệu.
- Chúng ta chọn lớp ẩn gồm 4 nơ-ron (do mình chọn)
- Chúng ta chọn lớp đầu ra gồm 3 nơ-ron bởi tập dữ liệu có 3 lớp



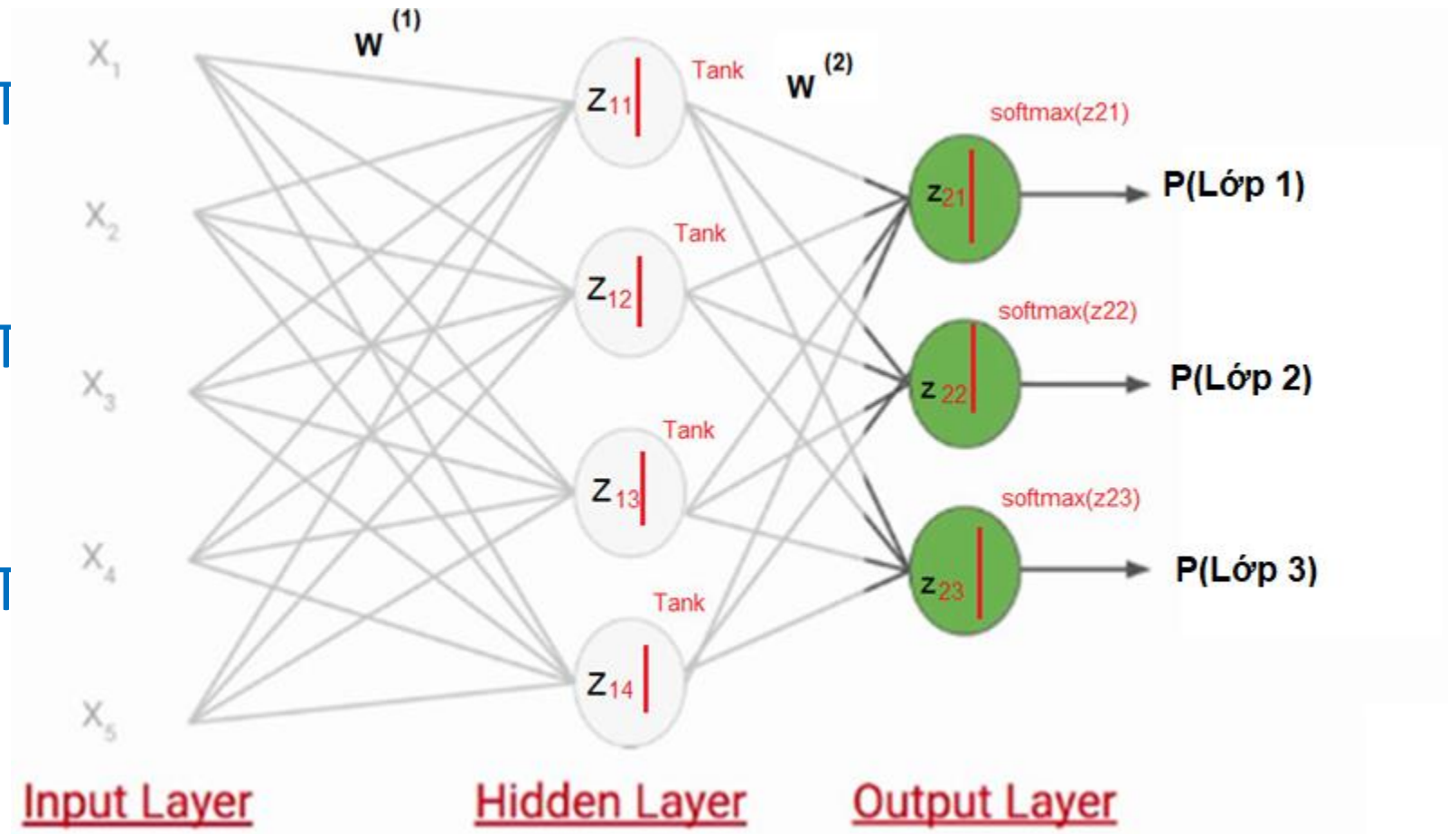
- Mỗi nơ-ron lớp ẩn sử dụng inputs, weights, và biases để tính tổng ra một giá trị (ký hiệu là z_{11} , z_{12} , z_{13} , z_{14})



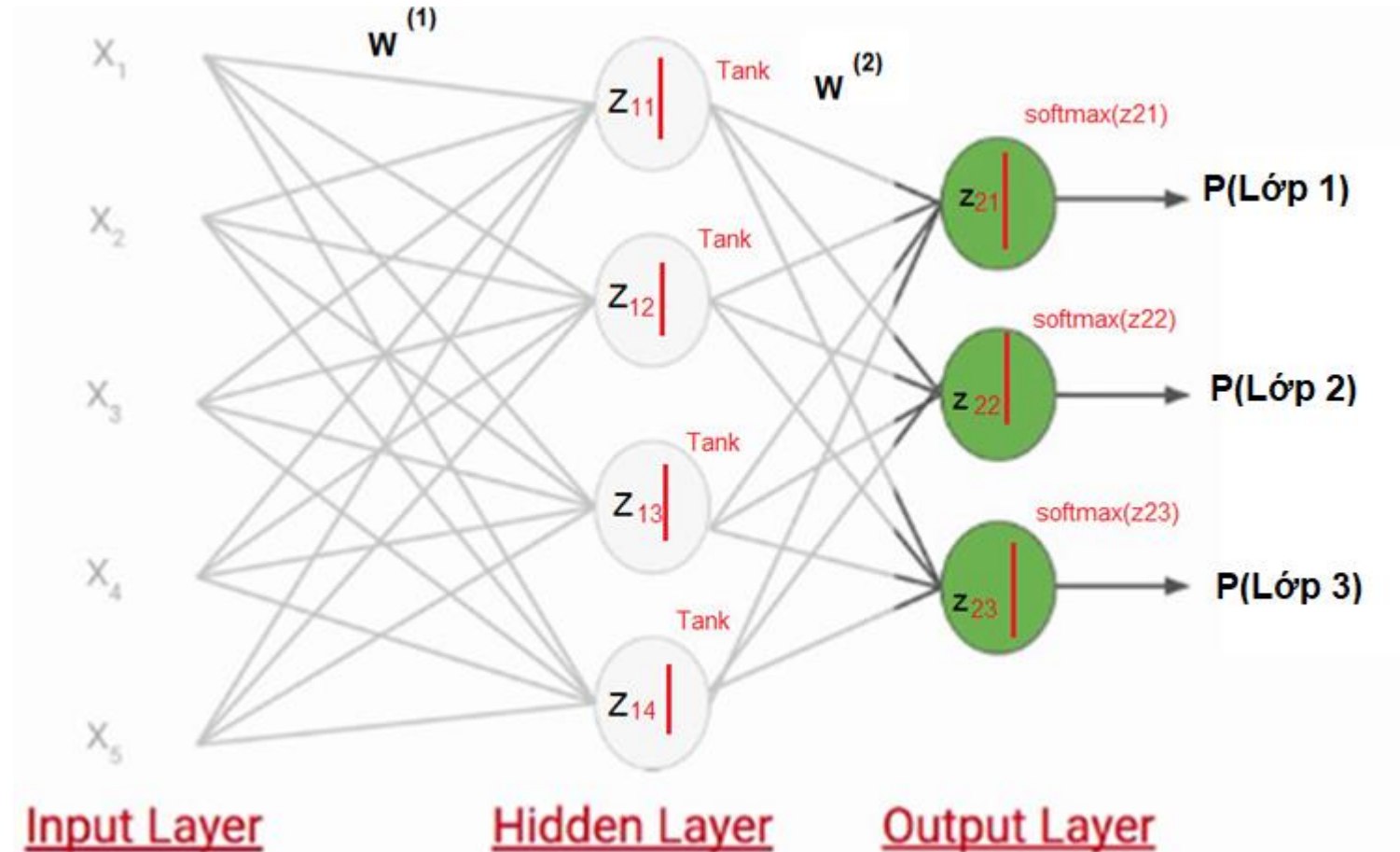
- Trên mỗi z_{1j} , ta áp dụng hàm Tanh:
 $\text{Tanh}(z_{11})$, $\text{Tanh}(z_{12})$,
 $\text{Tanh}(z_{13})$, $\text{Tanh}(z_{14})$
- Gửi các giá trị
 $\text{Tanh}(z_{11})$, $\text{Tanh}(z_{12})$,
 $\text{Tanh}(z_{13})$, $\text{Tanh}(z_{14})$
đến lớp đầu ra



- $Z_{21} = w_{11}^{(2)} \text{Tanh}(z_{11}) + w_{21}^{(2)} \text{Tanh}(z_{12}) + w_{31}^{(2)} \text{Tanh}(z_{13}) + w_{41}^{(2)} \text{Tanh}(z_{14})$
- $Z_{22} = w_{12}^{(2)} \text{Tanh}(z_{11}) + w_{22}^{(2)} \text{Tanh}(z_{12}) + w_{32}^{(2)} \text{Tanh}(z_{13}) + w_{42}^{(2)} \text{Tanh}(z_{14})$
- $Z_{23} = w_{13}^{(2)} \text{Tanh}(z_{11}) + w_{23}^{(2)} \text{Tanh}(z_{12}) + w_{33}^{(2)} \text{Tanh}(z_{13}) + w_{43}^{(2)} \text{Tanh}(z_{14})$



- $\text{Softmax}(Z_{21})$ sẽ cho xác suất mà đầu vào thuộc về lớp 1,....
- $\text{Softmax}(Z_{22})$ sẽ cho xác suất mà đầu vào thuộc về lớp 2,....
- $\text{Softmax}(Z_{23})$ sẽ cho xác suất mà đầu vào thuộc về lớp 3,....



- Tính Softmax(Z_{21}), Softmax(Z_{22}), và Softmax(Z_{23})

$$\text{softmax}(z_{21}) = \frac{\exp(z_{21})}{\exp(z_{21}) + \exp(z_{22}) + \exp(z_{23})}$$

Z_{21} 2.33	\rightarrow P(Lớp 1)	$= \frac{\exp(2.33)}{\exp(2.33) + \exp(-1.46) + \exp(0.56)}$	$= 0.83827314$
Z_{22} -1.46	\rightarrow P(Lớp 2)	$= \frac{\exp(-1.46)}{\exp(2.33) + \exp(-1.46) + \exp(0.56)}$	$= 0.01894129$
Z_{23} 0.56	\rightarrow P(Lớp 3)	$= \frac{\exp(0.56)}{\exp(2.33) + \exp(-1.46) + \exp(0.56)}$	$= 0.14278557$

Sử dụng thuật toán

Back-propagation

+ Với mỗi phần tử dữ liệu huấn luyện: cho đi qua mạng + điều chỉnh các tham số của mạng.

```
function BACK-PROP-LEARNING(examples, network) returns a neural network
  inputs: examples, a set of examples, each with input vector  $\mathbf{x}$  and output vector  $\mathbf{y}$ 
           network, a multilayer network with  $L$  layers, weights  $w_{i,j}$ , activation function  $g$ 
  local variables:  $\Delta$ , a vector of errors, indexed by network node

  repeat
    for each weight  $w_{i,j}$  in network do
       $w_{i,j} \leftarrow$  a small random number
    for each example  $(\mathbf{x}, \mathbf{y})$  in examples do
      /* Propagate the inputs forward to compute the outputs */
      for each node  $i$  in the input layer do
         $a_i \leftarrow x_i$ 
      for  $\ell = 2$  to  $L$  do
        for each node  $j$  in layer  $\ell$  do
           $in_j \leftarrow \sum_i w_{i,j} a_i$ 
           $a_j \leftarrow g(in_j)$ 
      /* Propagate deltas backward from output layer to input layer */
      for each node  $j$  in the output layer do
         $\Delta[j] \leftarrow g'(in_j) \times (y_j - a_j)$ 
      for  $\ell = L - 1$  to  $1$  do
        for each node  $i$  in layer  $\ell$  do
           $\Delta[i] \leftarrow g'(in_i) \sum_j w_{i,j} \Delta[j]$ 
      /* Update every weight in network using deltas */
      for each weight  $w_{i,j}$  in network do
         $w_{i,j} \leftarrow w_{i,j} + \alpha \times a_i \times \Delta[j]$ 
  until some stopping criterion is satisfied
  return network
```

Chọn kiến trúc mạng nơ ron

- Lựa chọn kiến trúc mạng nơ ron thích hợp cho bài toán:
 - feedforward networks
 - Recurrent networks
 - or convolutional networks.
- Sự lựa chọn phụ thuộc vào:
 - loại dữ liệu đầu vào,
 - độ phức tạp của bài toán
 - Đầu ra mong muốn

Chuẩn bị dữ liệu

- Thu thập và tiền xử lý dữ liệu để đảm bảo nó phù hợp với khuôn dạng cho mạng nơ ron
- Bao gồm chuẩn hóa, mở rộng quy mô hoặc tăng cường dữ liệu để cải tiến hiệu năng mạng và khả năng tổng quát hóa.

Khởi tạo **Weights** và **Biases**

- Khởi tạo ngẫu nhiên các **Weights** và **Biases** của mạng nơ ron.
- Khởi tạo phù hợp là quan trọng cho huấn luyện hiệu quả, do nó có thể ảnh hưởng đến tốc độ hội tụ và hiệu năng cuối cùng của mạng

Xác định Loss Function

- Chọn **Loss Function** mà đo sự khác nhau giữa các giá trị thực tế và dự đoán.
- Loss functions gồm:
 - mean squared error cho regression task
 - cross-entropy cho bài toán phân lớp

Huấn luyện mạng

- Huấn luyện mạng nơ ron sử dụng thuật toán tối ưu như gradient descent or Adam or RMSprop.
- Bộ tối ưu điều chỉnh các weights and biases để cực tiểu loss function, lặp lại việc cải tiến hiệu năng mạng trên tập huấn luyện

Validate và Fine-tune

- Đánh giá mạng nơ ron được huấn luyện trên một tập validation tách biệt để xác định khả năng tổng quát của nó.
- Fine-tune các siêu tham số của mạng như learning rate, batch size, hoặc số các lớp ẩn, để tối ưu hiệu năng của nó trên cả training và validation data

Yêu cầu dữ liệu

- Mạng nơ ron đòi hỏi lượng lớn dữ liệu có nhãn cho huấn luyện, nó có thể khó khăn cho việc thu thập, gán nhãn, và tiền xử lý.
- Dữ liệu kém chất lượng hoặc không đủ có thể dẫn đến overfitting hoặc hiệu năng dưới tối ưu

Phức tạp tính toán

- Huấn luyện mạng nơ ron đặc biệt mạng học sâu đòi hỏi năng lực phần cứng như GPUs or TPUs.

Khả năng giải thích mô hình

- Mạng lưới thần kinh thường được coi là "hộp đen" do cấu trúc bên trong phức tạp của chúng, khiến việc hiểu cách chúng đưa ra những dự đoán cụ thể trở nên khó khăn.

Điều chỉnh siêu tham số

- Lựa chọn các siêu tham số đúng như learning rate, network architecture, và activation functions, có thể là thách thức và tốn nhiều thời gian
- Chọn các siêu tham số không hiệu quả có thể dẫn đến hiệu năng thấp hoặc hội tụ chậm trong quá trình huấn luyện

Overfitting

- Các mạng nơ ron có xu hướng quá khớp, đặc biệt khi huấn luyện trên các tập dữ liệu nhỏ hoặc nhiều.
- Quá khớp xuất hiện khi mô hình thu nhiều trong dữ liệu, dẫn đến hiệu năng cao trên tập huấn luyện nhưng tổng quát kém đối với dữ liệu mới

Cực tiểu cục bộ và biến mất đạo hàm

- Trong khi huấn luyện, mạng nơ ron có thể mắc kẹt ở cực tiểu cục bộ, ngăn cản chúng đạt được cực tiểu toàn cục của hàm mất mát.
- Vấn đề biến mất đạo hàm (các đạo hàm quá nhỏ để cập nhật các trọng số hiệu quả) có thể cản trở trong mô hình học sâu.



CMC UNIVERSITY

Aspire to Inspire the Digital World



THANK YOU

