

# KHAI PHÁ DỮ LIỆU

Trường Đại học Nha Trang  
Khoa Công nghệ thông tin  
Bộ môn Hệ thống thông tin  
Giáo viên: TS.Nguyễn Khắc Cường

# CHỦ ĐỀ 4

## PHÂN LỚP

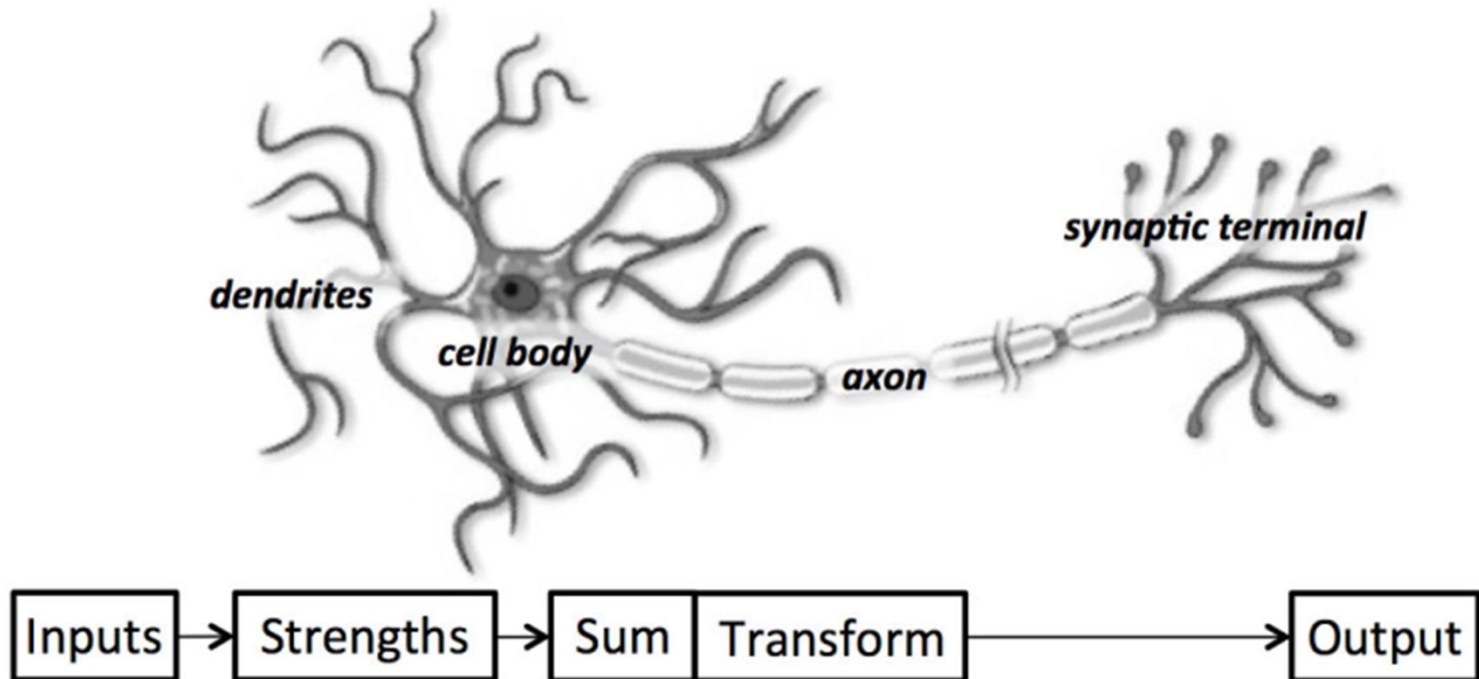
(Neural network

Deep learning

Convolutional Neural Network)

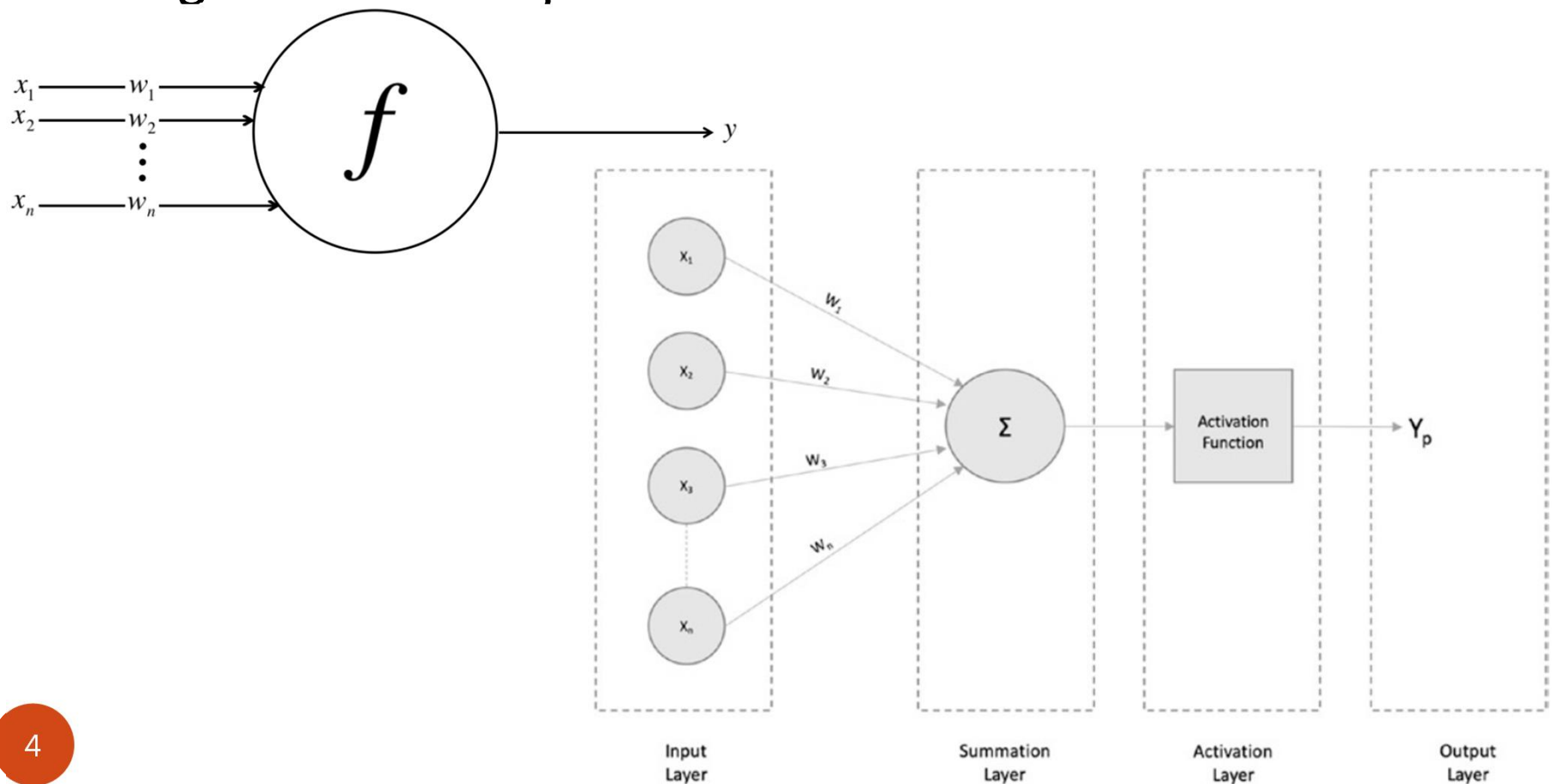
# Neural network

- Neuron
  - Là đơn vị xử lý của bộ não con người
  - Là nguồn gốc của ý tưởng xây dựng giải thuật artificial neural net



# Neural network

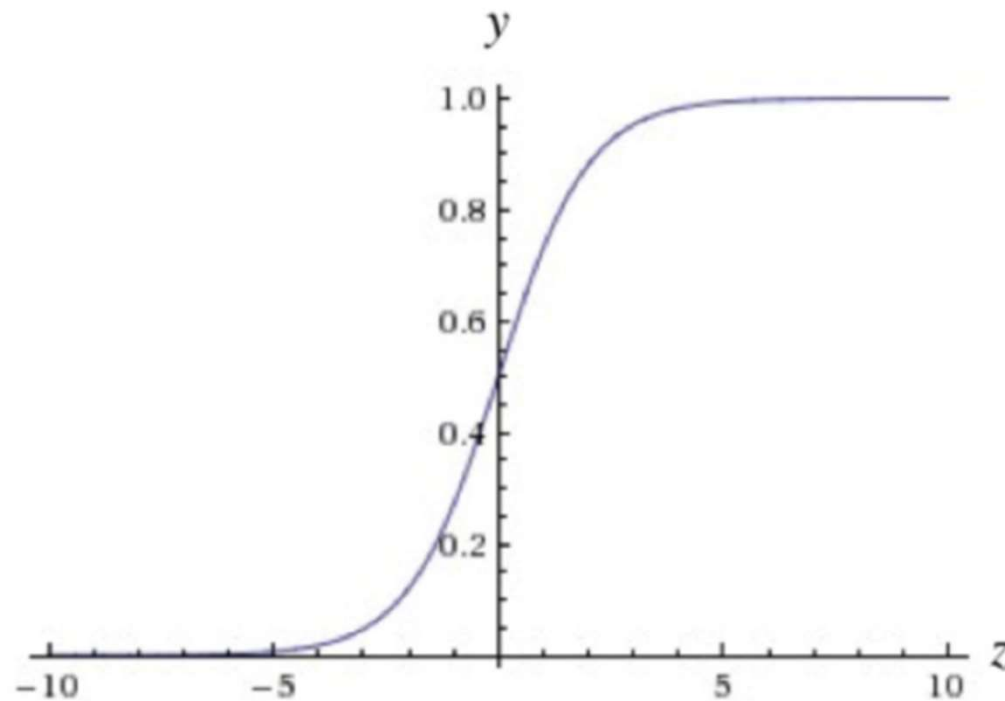
- Cấu trúc của một “neuron” trong artificial neural net  
→ gọi là: Perceptron



# Neural network

- Một số activation phổ biến
  - Sigmoid

$$f(z) = \frac{1}{1 + e^{-z}}$$

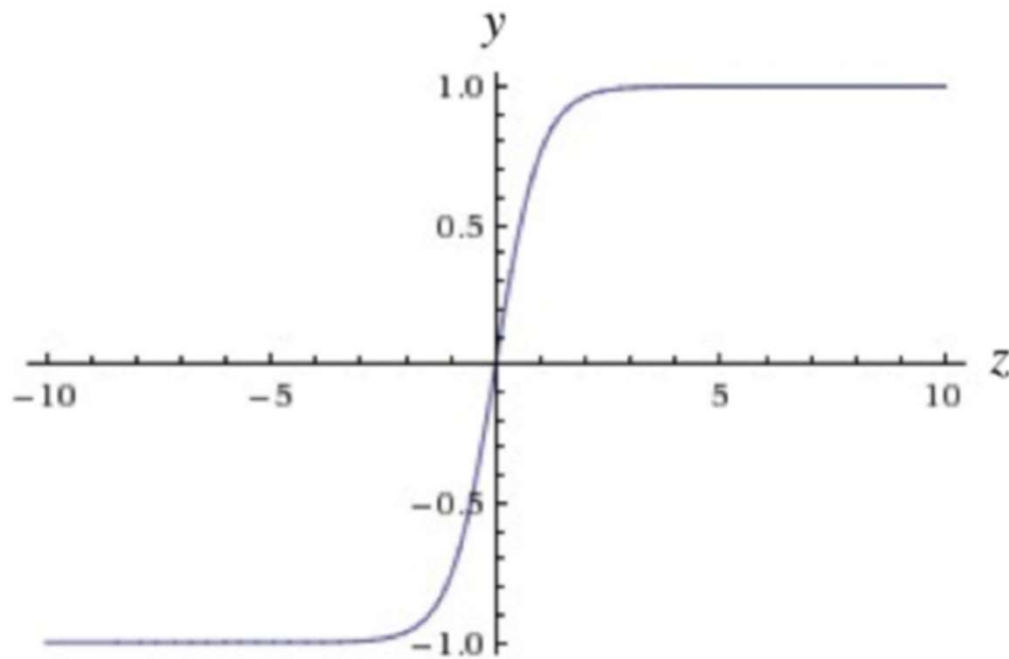


# Neural network

- Một số activation phổ biến

- Tanh

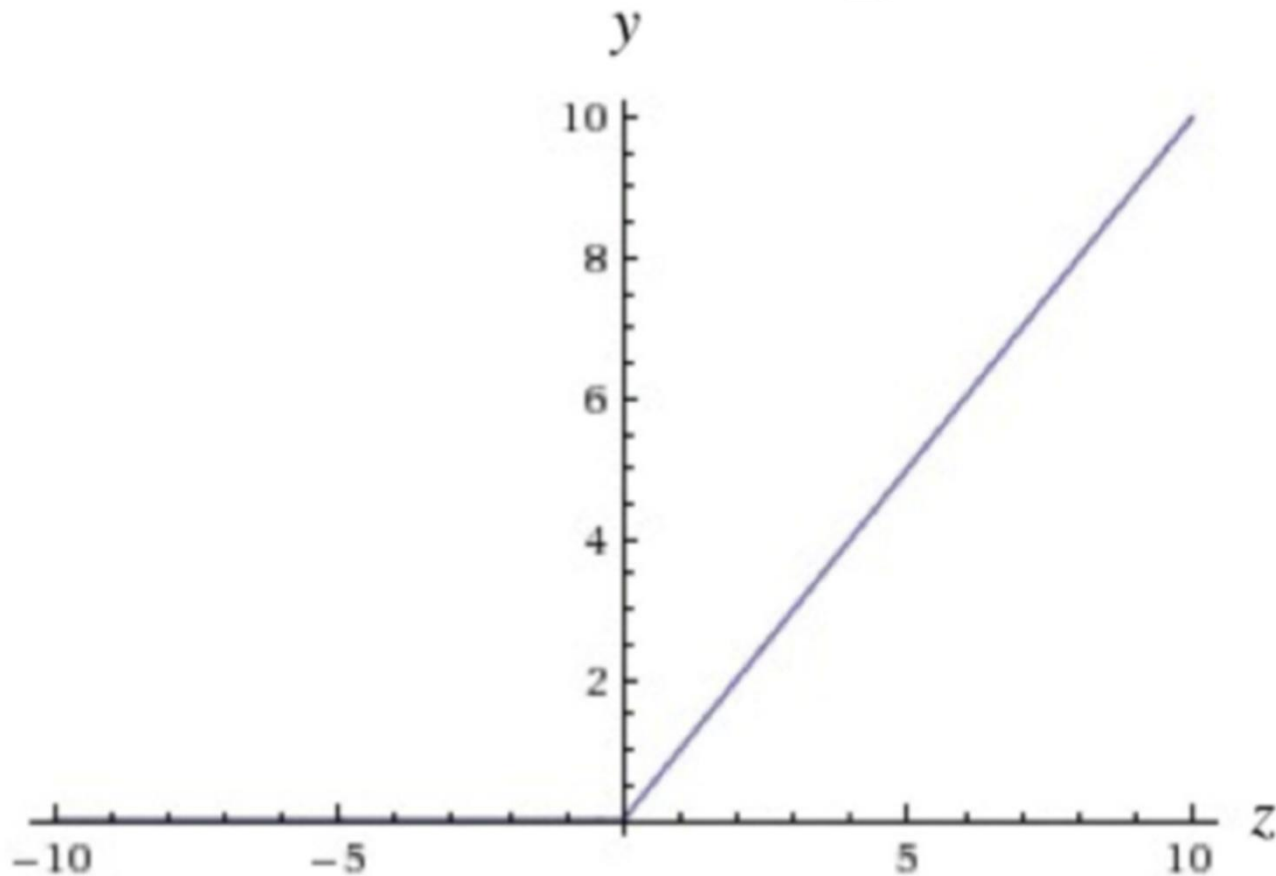
$$f(z) = \tanh(z).$$



# Neural network

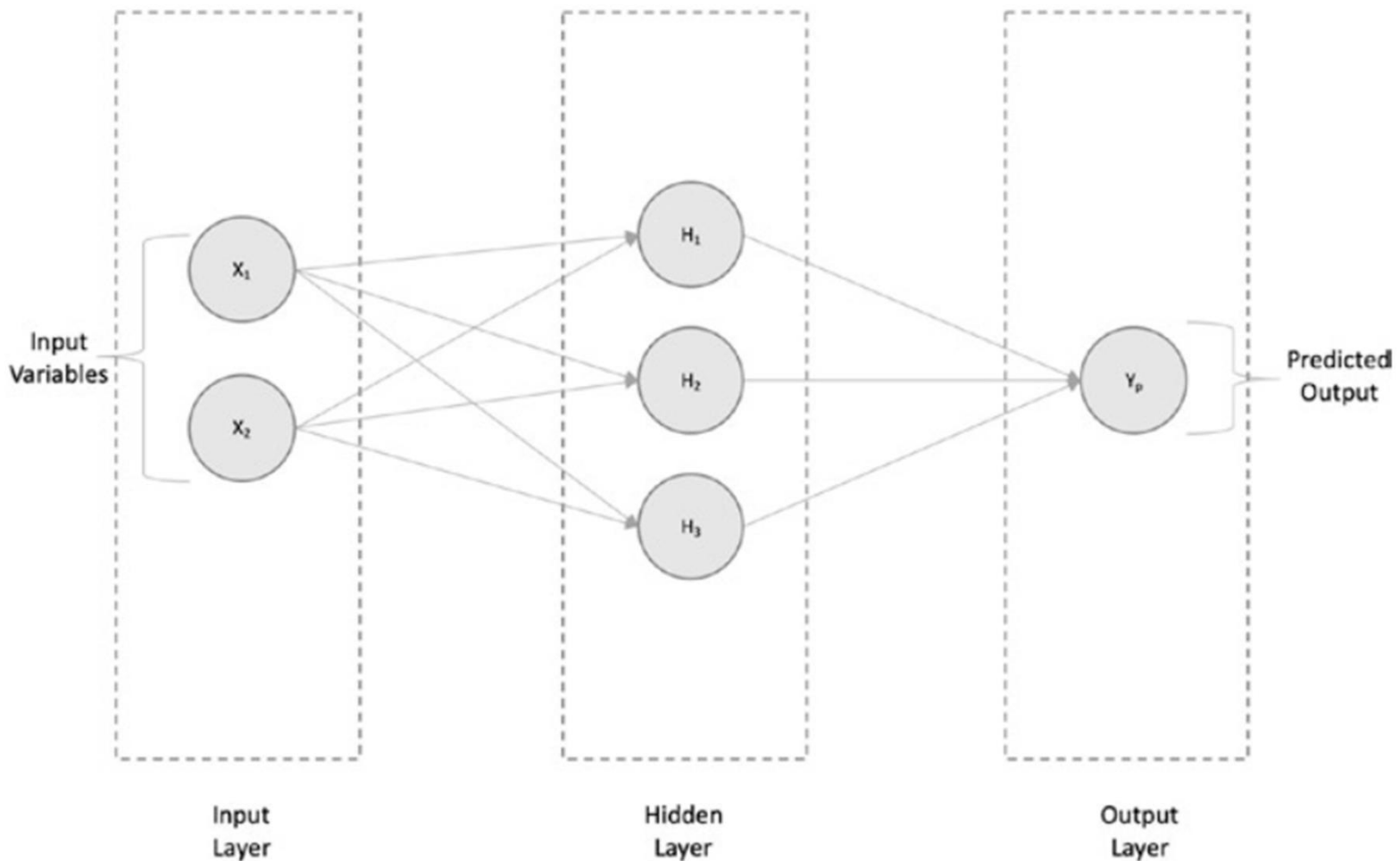
- Một số activation phổ biến
  - ReLU (restricted linear unit)

$$f(z) = \max(0, z),$$



# Neural network

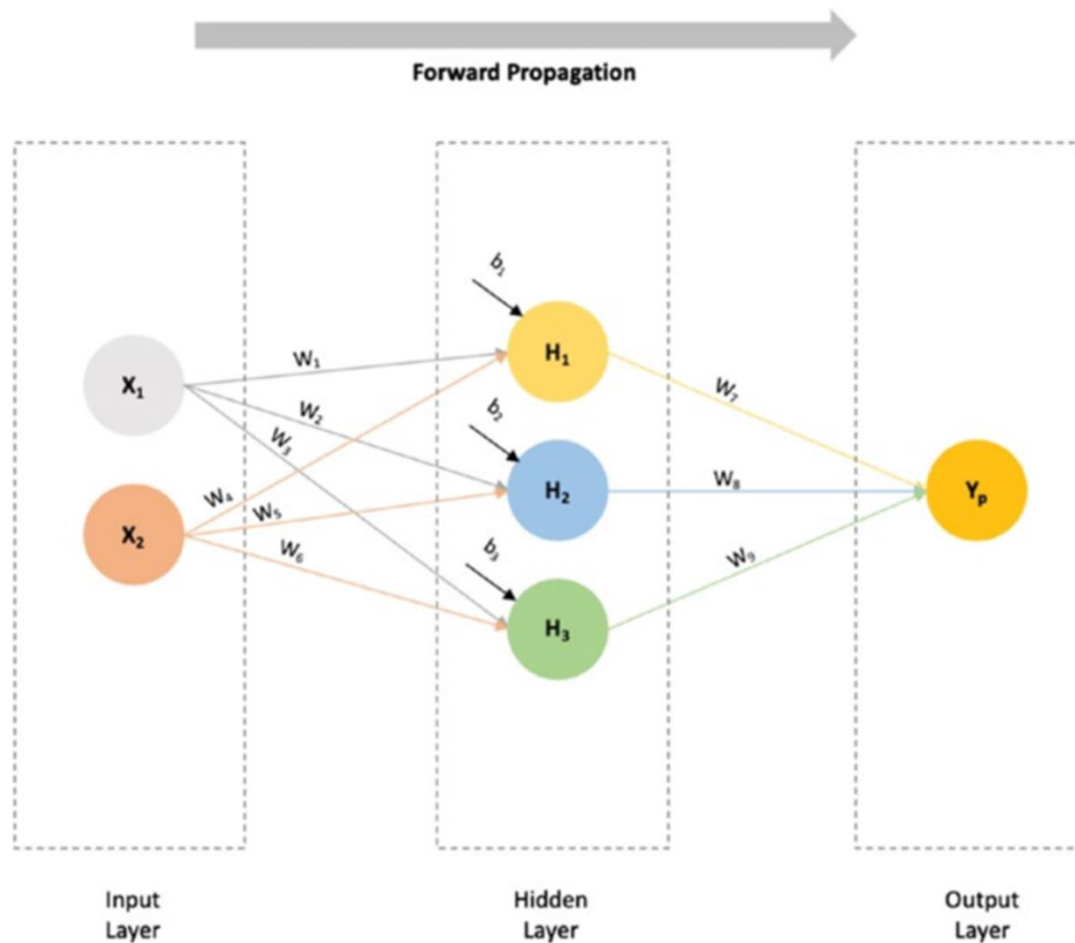
- Kiến trúc của một neural network đơn giản





# Neural network

- Forward Propagation



$$H_1 = W_1 * X_1 + W_4 * X_2 + b_1$$

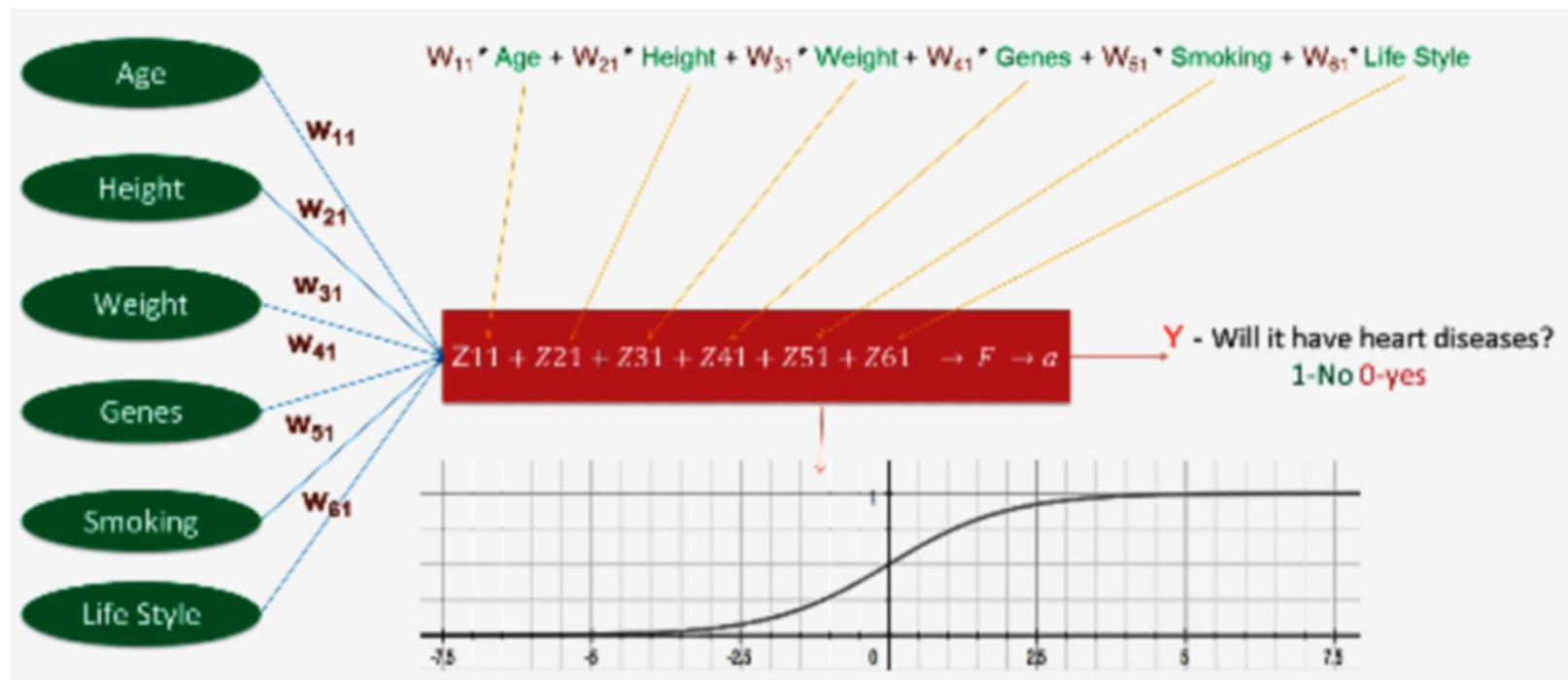
$$H_2 = W_2 * X_1 + W_5 * X_2 + b_2$$

$$H_3 = W_3 * X_1 + W_6 * X_2 + b_3$$

$$Y_p = W_7 * H_1 + W_8 * H_2 + W_9 * H_3$$

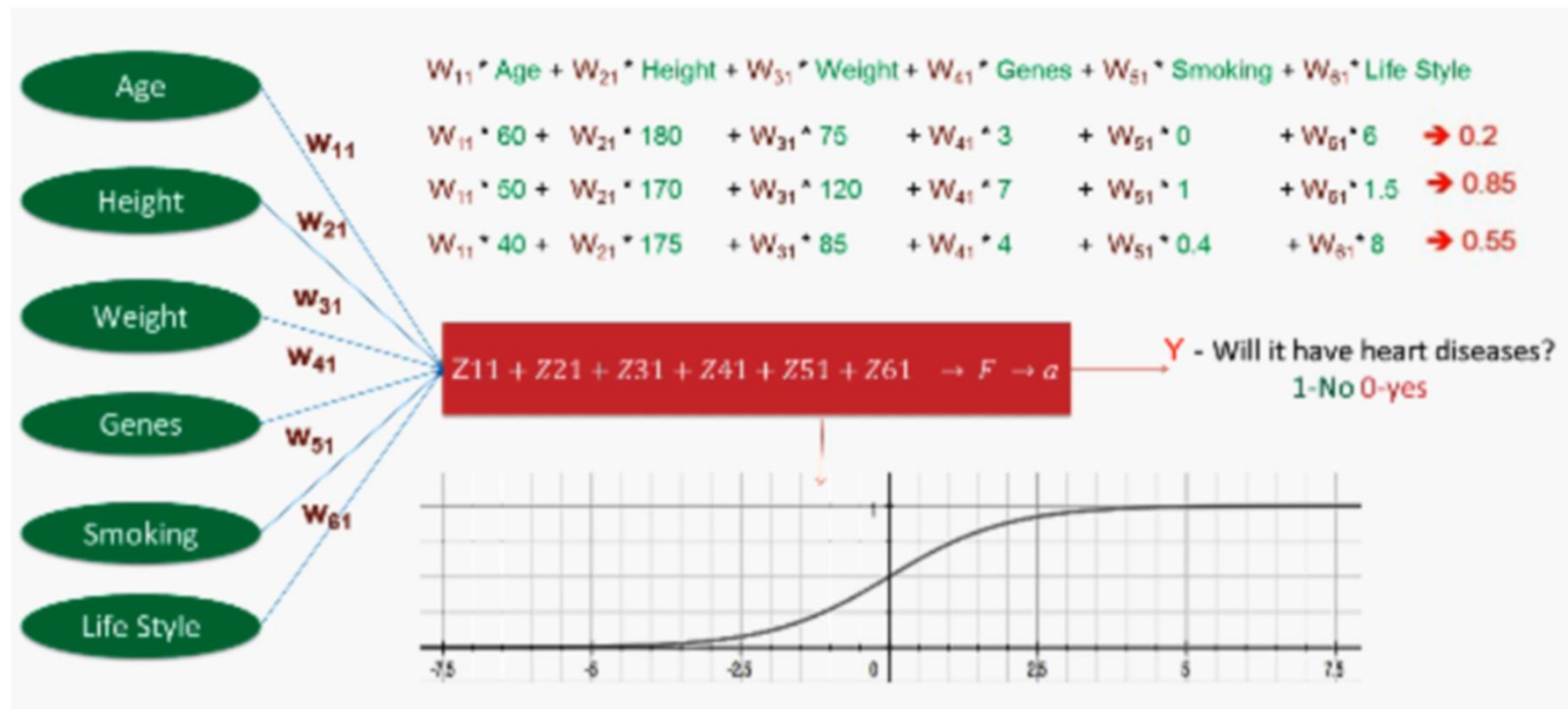
# Neural network

- Minh họa cách neural network “học”



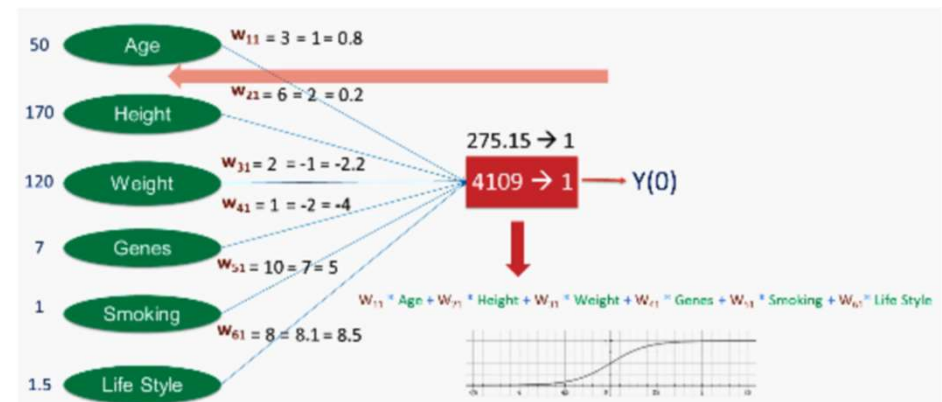
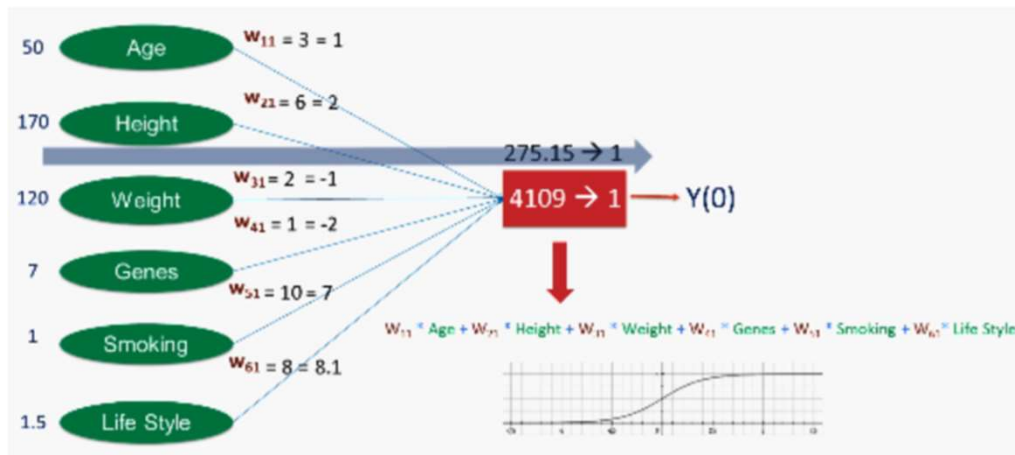
# Neural network

- Minh họa cách neural network “học”
  - Học từ các training data khác nhau



# Neural network

- Minh họa cách neural network “học”
  - Điều chỉnh weights (Back propagation)

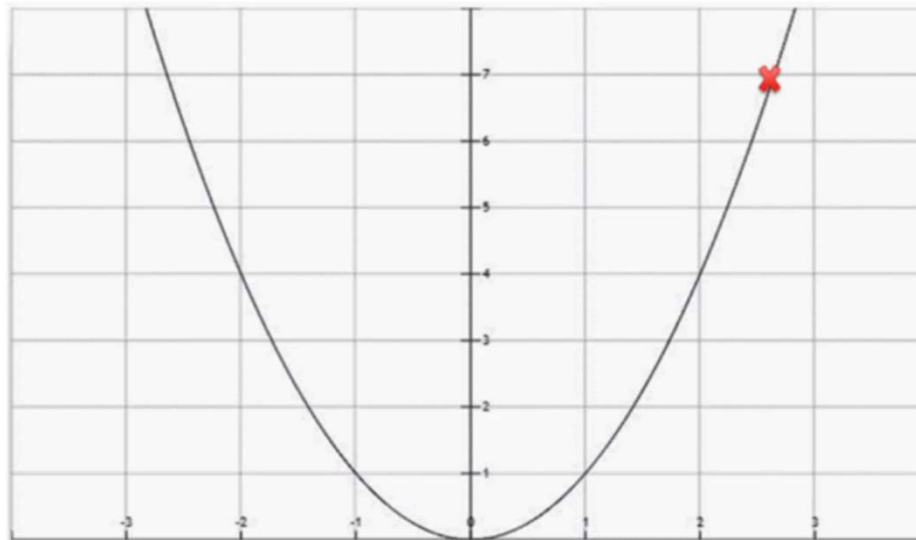


# Neural network

- Minh họa cách neural network “học”
  - Điều chỉnh weights (Back propagation)
    - Cần Loss function (tự thiết kế), ví dụ:

$$J = 1/m * \sum (H(w) - Y)^2 - LossFunction$$

→ Dùng để đánh giá mức độ khác nhau giữa giá trị thật sự (label) của từng training data so với giá trị mà designed ANN dự đoán



# Neural network

- Minh họa cách neural network “học”
  - Điều chỉnh weights (Backward propagation)
    - Mục tiêu của điều chỉnh weights:
      - Làm cho sự sai khác là dần ít nhất (gradient descent)
    - Idea:
      - Trừ weights cũ cho đạo hàm của nó như sau

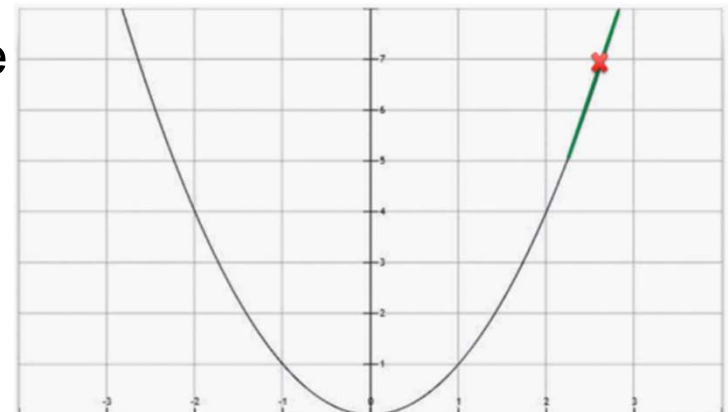
$$J = 1/m * \text{sum}(H(w) - Y)^2 - \text{LossFunction}$$

$$W_{11} = W_{11} - \alpha \frac{dJ}{dW_{11}},$$

$$W_{21} = W_{21} - \alpha \frac{dJ}{dW_{21}},$$

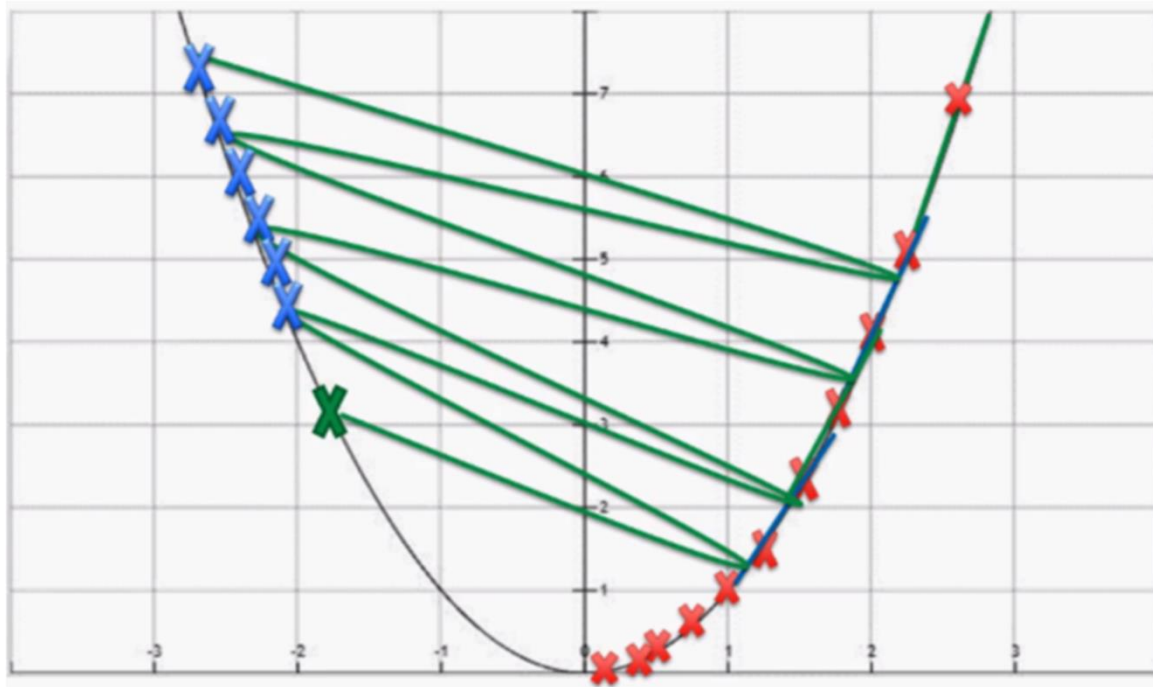
$$W_{31} = W_{31} - \alpha \frac{dJ}{dW_{31}},$$

$\alpha$ : learning rate



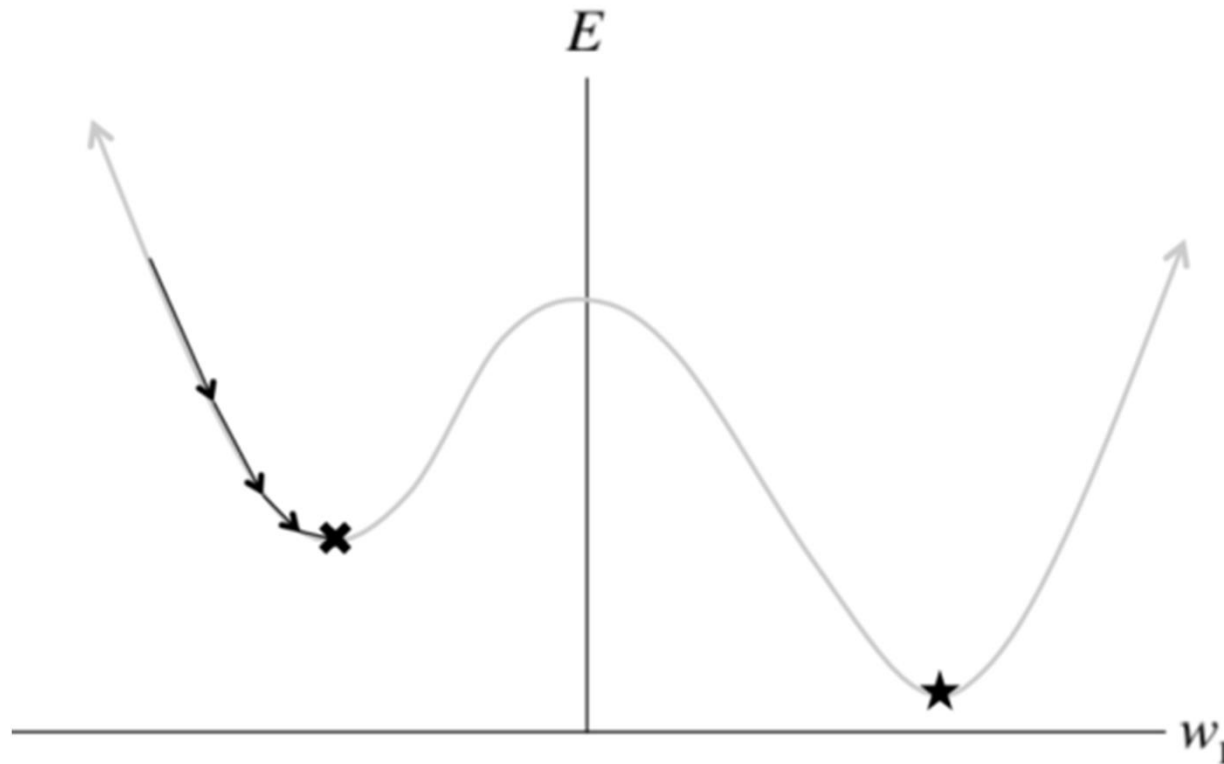
# Neural network

- Minh họa cách neural network “học”
  - Bản chất của quá trình học:
    - Lặp đi lặp lại Forward propagation và Backward propagation
    - Thu được bộ weights giúp cho sai số dần giảm xuống



# Neural network

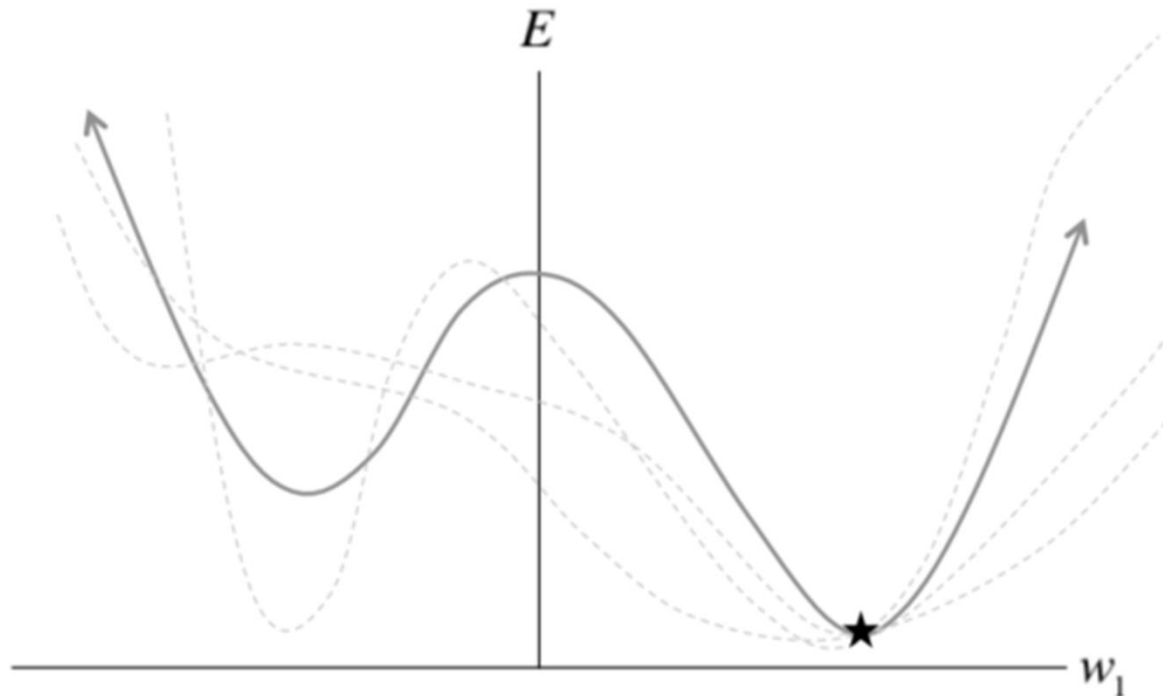
- Minh họa cách neural network “học”
  - Vấn đề cần lưu ý của gradient descent
    - rất dễ nhầm lẫn giữa local minima và global minima





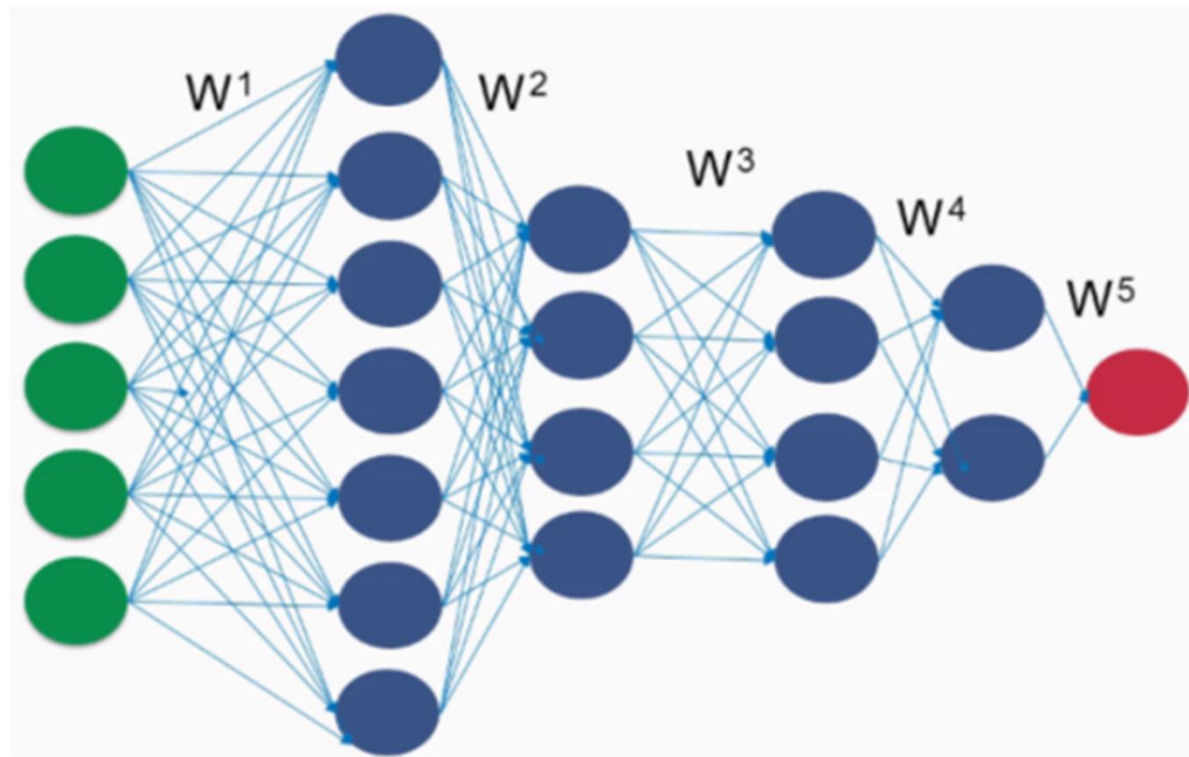
# Neural network

- Minh họa cách neural network “học”
  - Idea khắc phục vấn đề của gradient descent
    - Thay đổi (thiết kế / dùng) nhiều kiểu function đánh giá sai số  $\rightarrow$  so sánh  $\rightarrow$  tìm ra thiết kế nào tránh được nhiều nhất nhàm lẫn giữa local minima và global minima



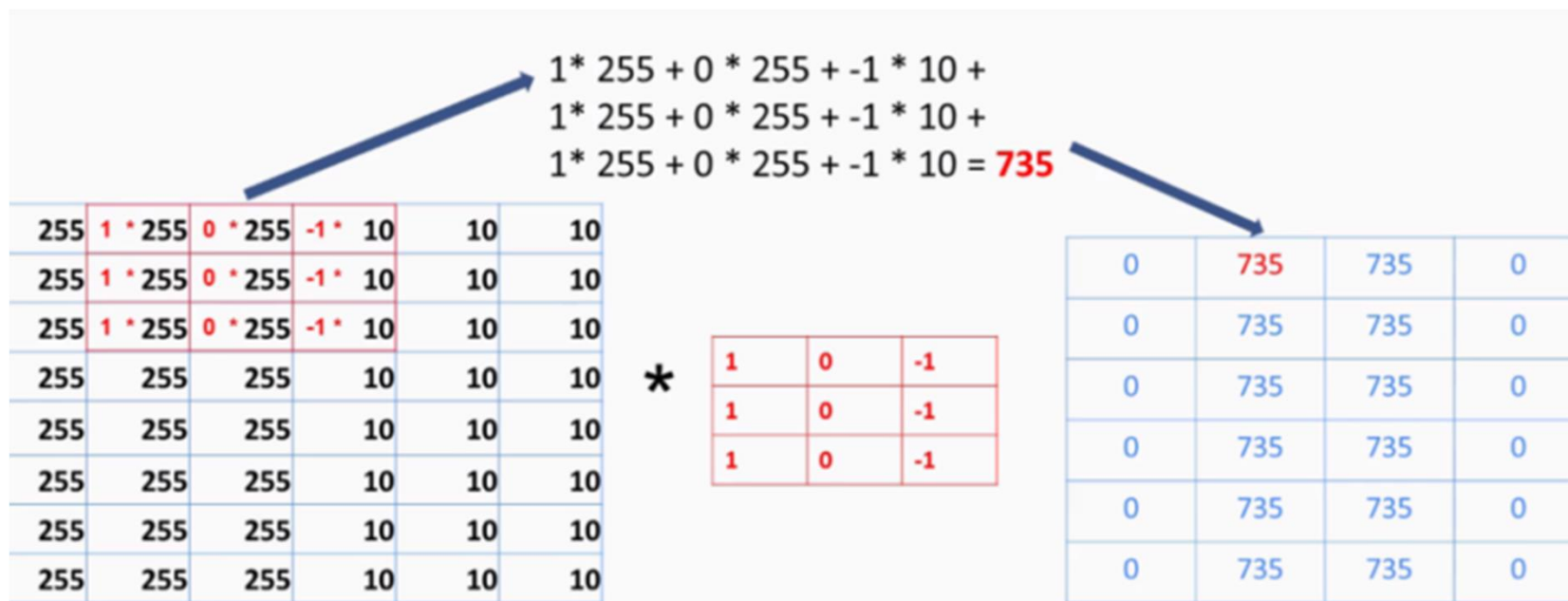
# Deep learning

- Deep learning
  - Nhiều layer hơn
  - Mỗi layer nhiều node hơn



# Convolutional Neural Network

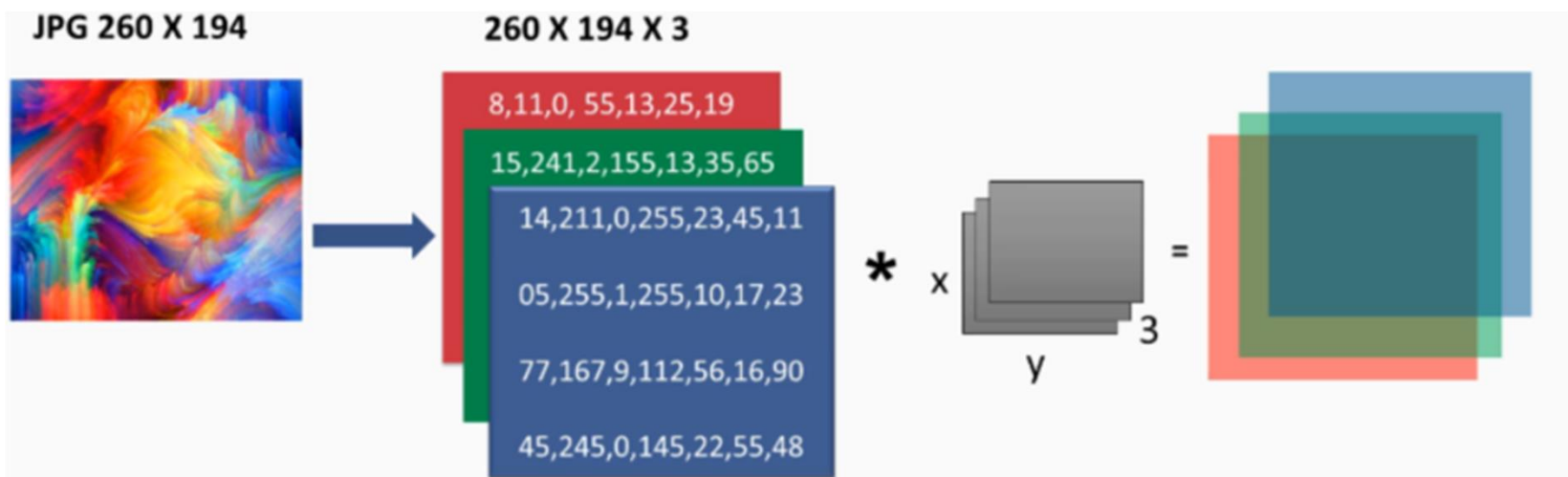
- CNN = Convolutional Neural Network architectures
- Là một phiên bản của Deep learning
- CNN thường dùng với data là image
- Convolution?



# Convolutional Neural Network

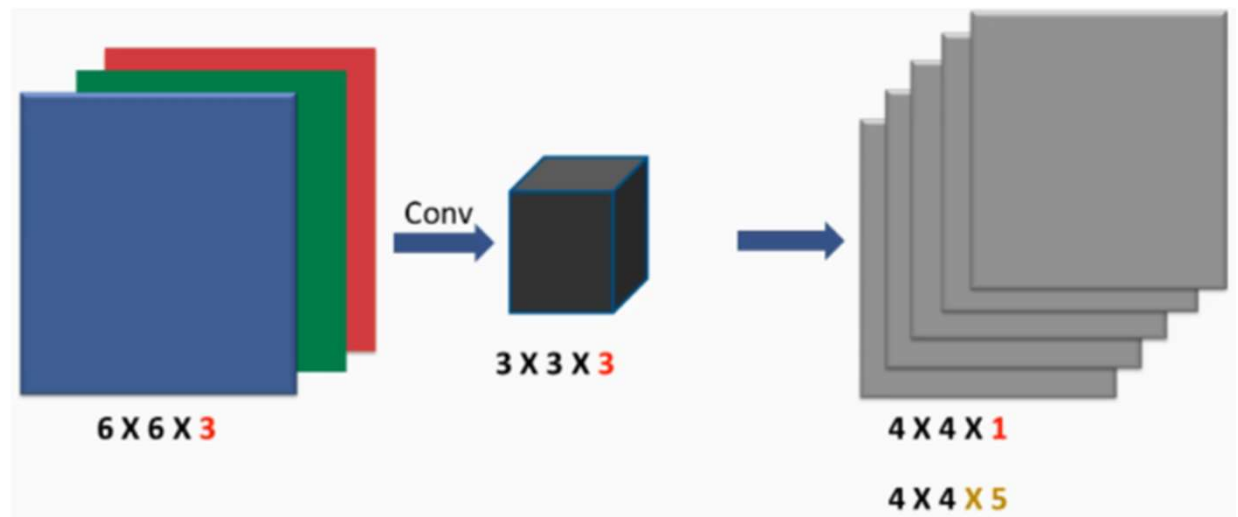
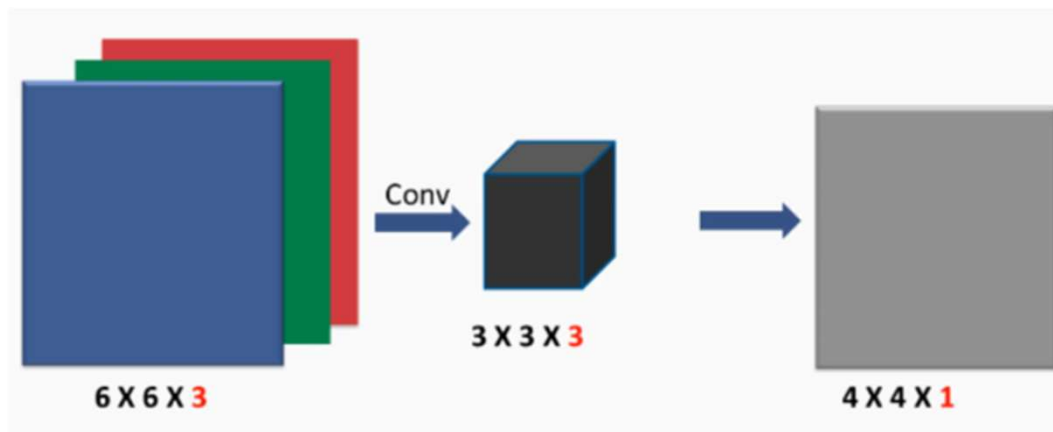
- Sự thay đổi size, shape khi thực hiện convolution?

255	255	255	10	10	10	<div>8 X 6</div> <div>*</div> <div><div><div>1</div><div>0</div><div>-1</div></div><div><div>1</div><div>0</div><div>-1</div></div><div><div>1</div><div>0</div><div>-1</div></div></div> <div>3 X 3</div> <div>=</div> <div>6 X 4</div>	0	735	735	0
255	255	255	10	10	0		735	735	0	
255	255	255	10	10	0		735	735	0	
255	255	255	10	10	0		735	735	0	
255	255	255	10	10	0		735	735	0	
255	255	255	10	10	0		735	735	0	
255	255	255	10	10	0		735	735	0	
255	255	255	10	10	0		735	735	0	



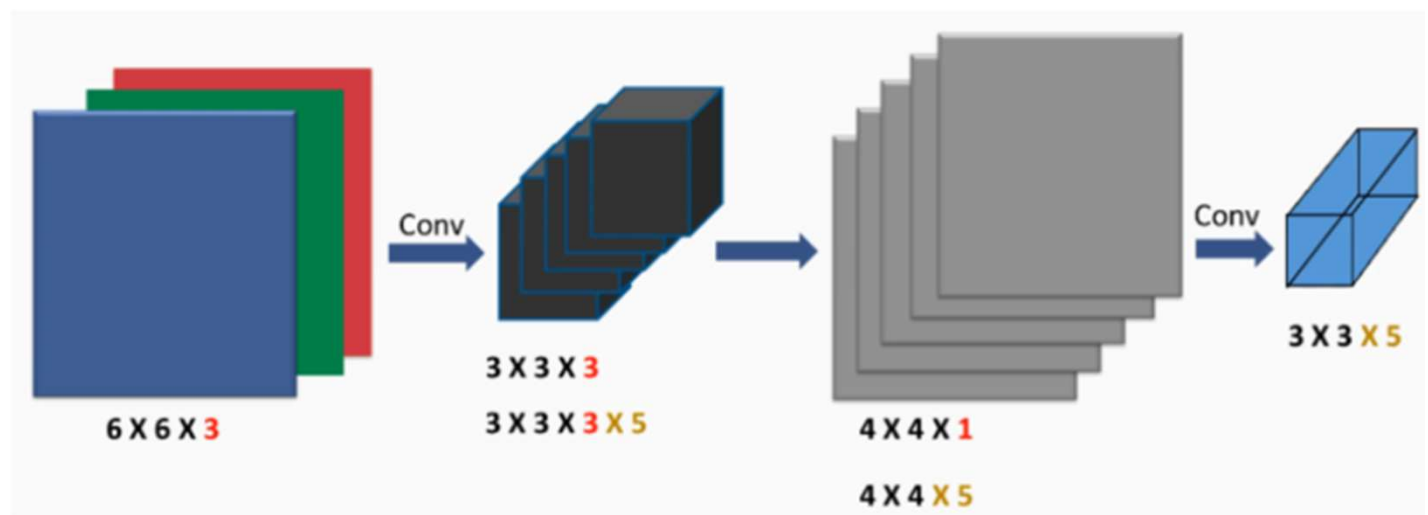
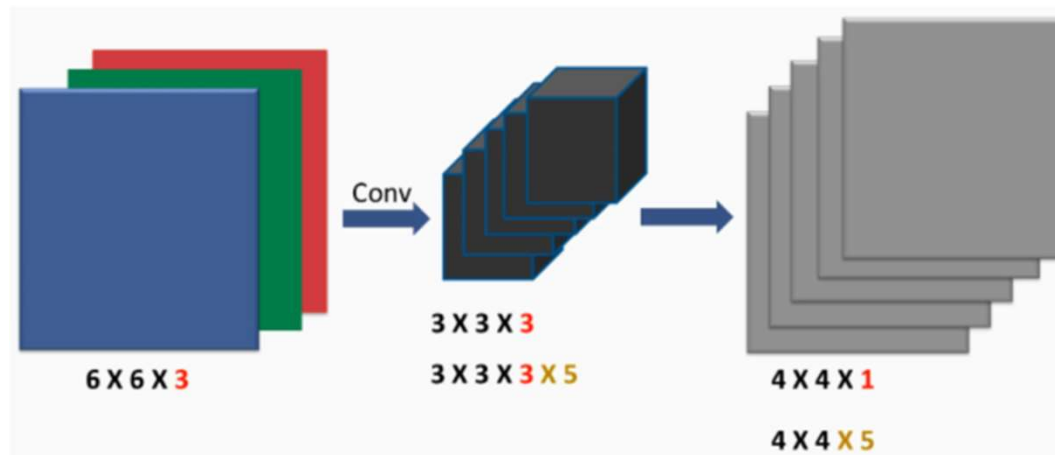
# Convolutional Neural Network

- Sự thay đổi size, shape khi thực hiện convolution?



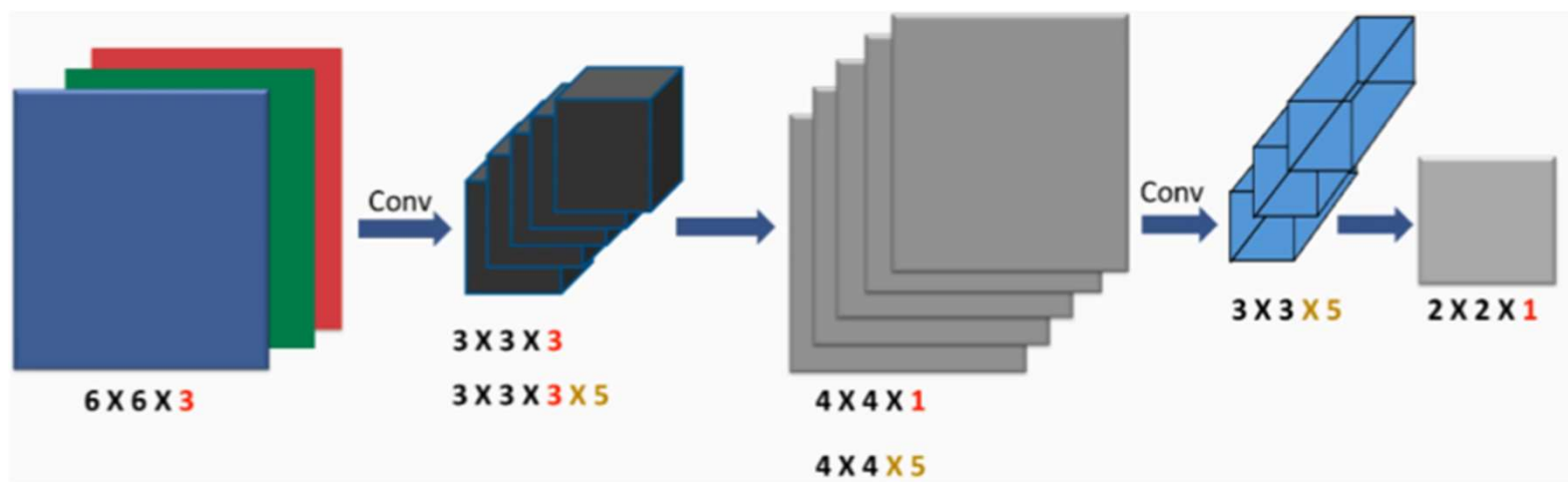
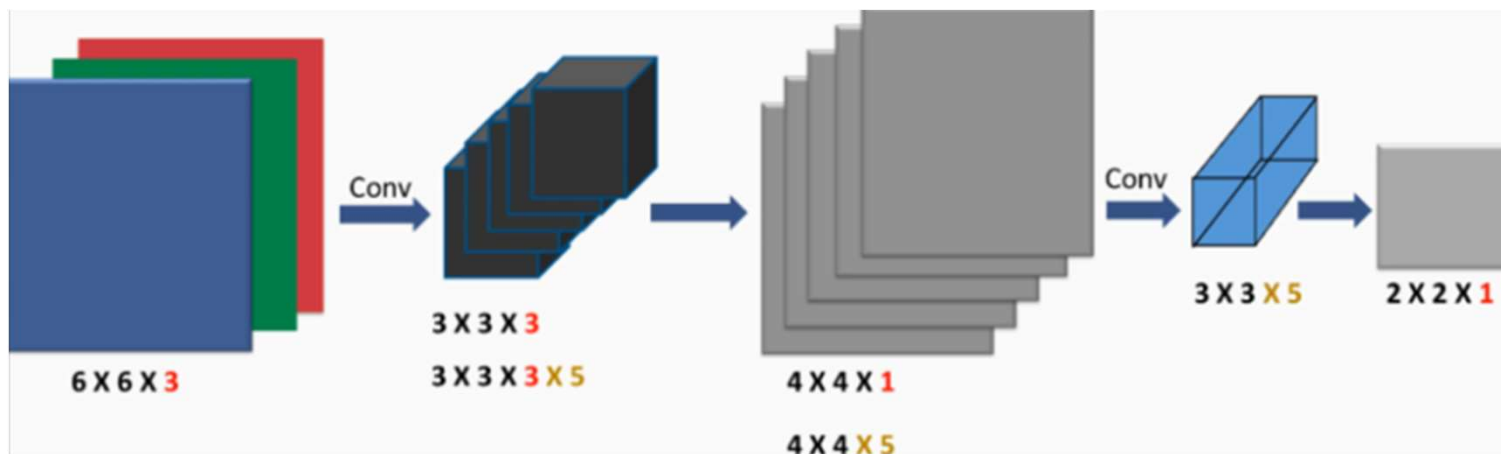
# Convolutional Neural Network

- Sự thay đổi size, shape khi thực hiện convolution?



# Convolutional Neural Network

- Sự thay đổi size, shape khi thực hiện convolution?



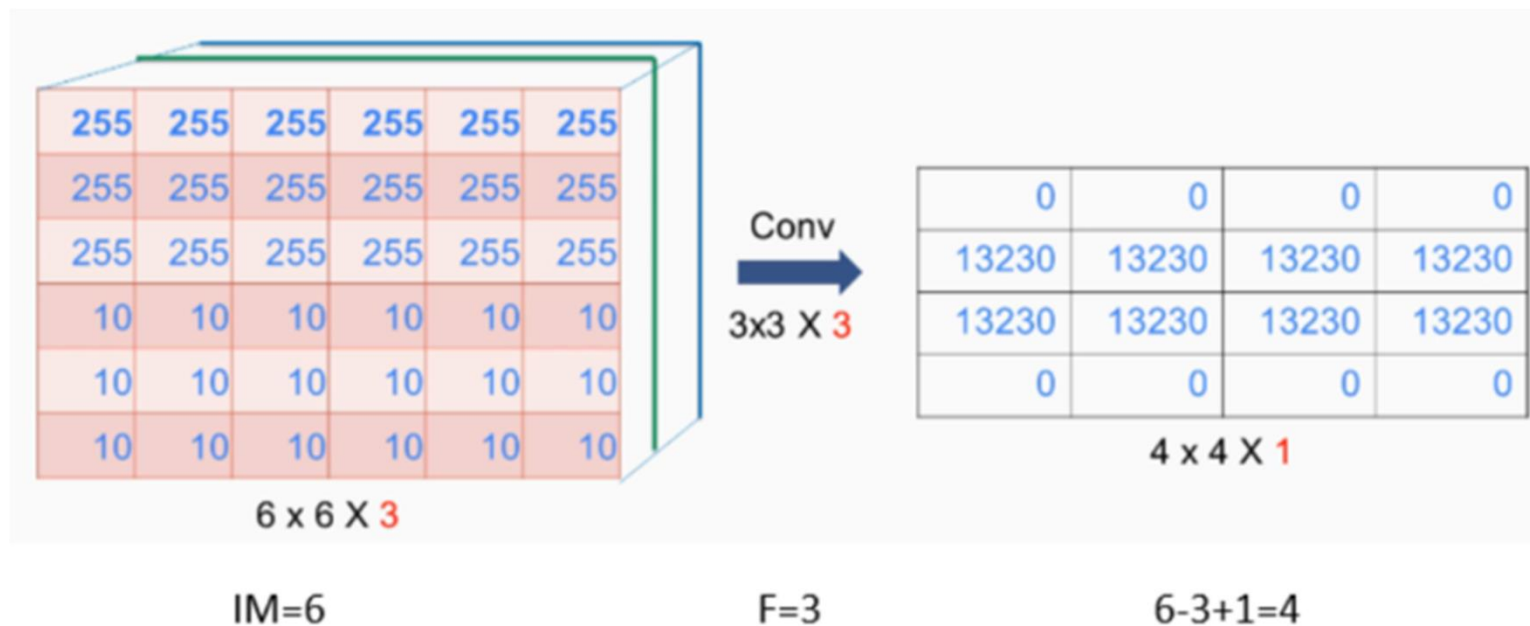
# Convolutional Neural Network

- Các tham số của convolution layer?

- Size output:

- $OM = IM - F + 1$

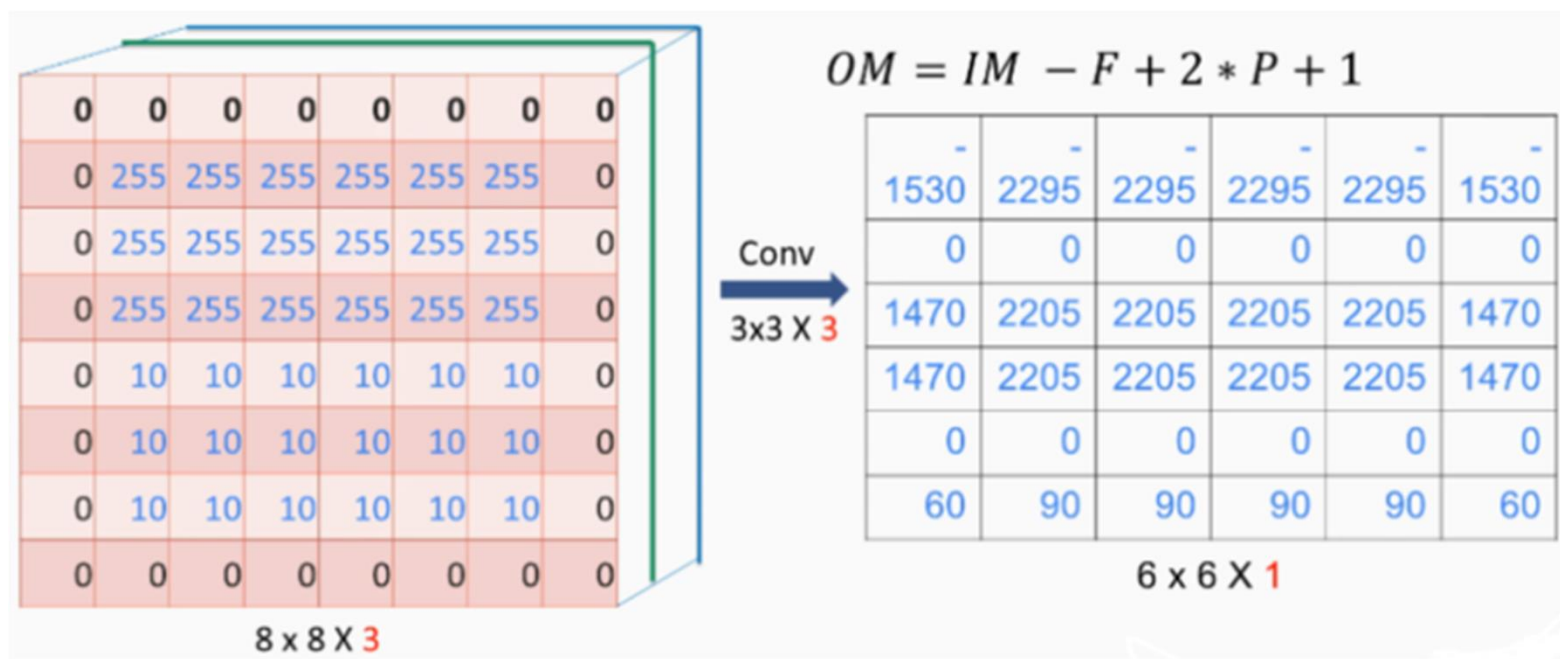
(Output matrix = input matrix – filter size + 1)





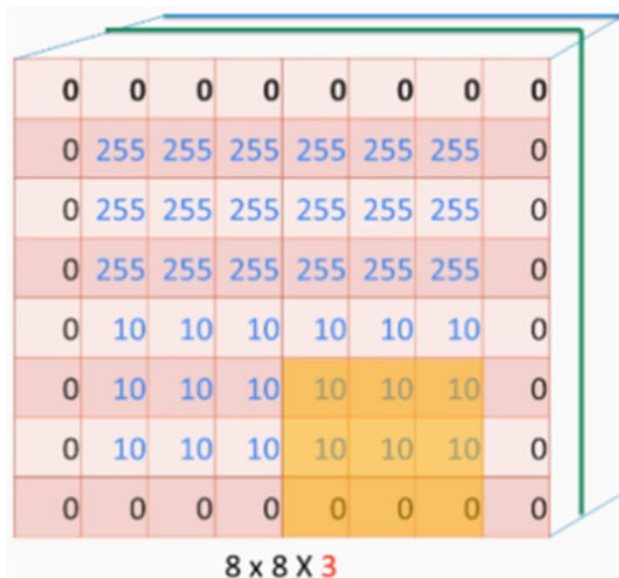
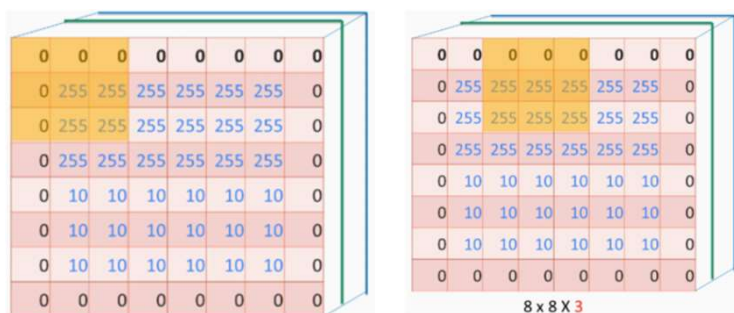
# Convolutional Neural Network

- Các tham số của convolution layer?
  - Padding:



# Convolutional Neural Network

- Các tham số của convolution layer?
- Stride:



$$OM = \frac{IM - F + 2 * P}{S} + 1$$

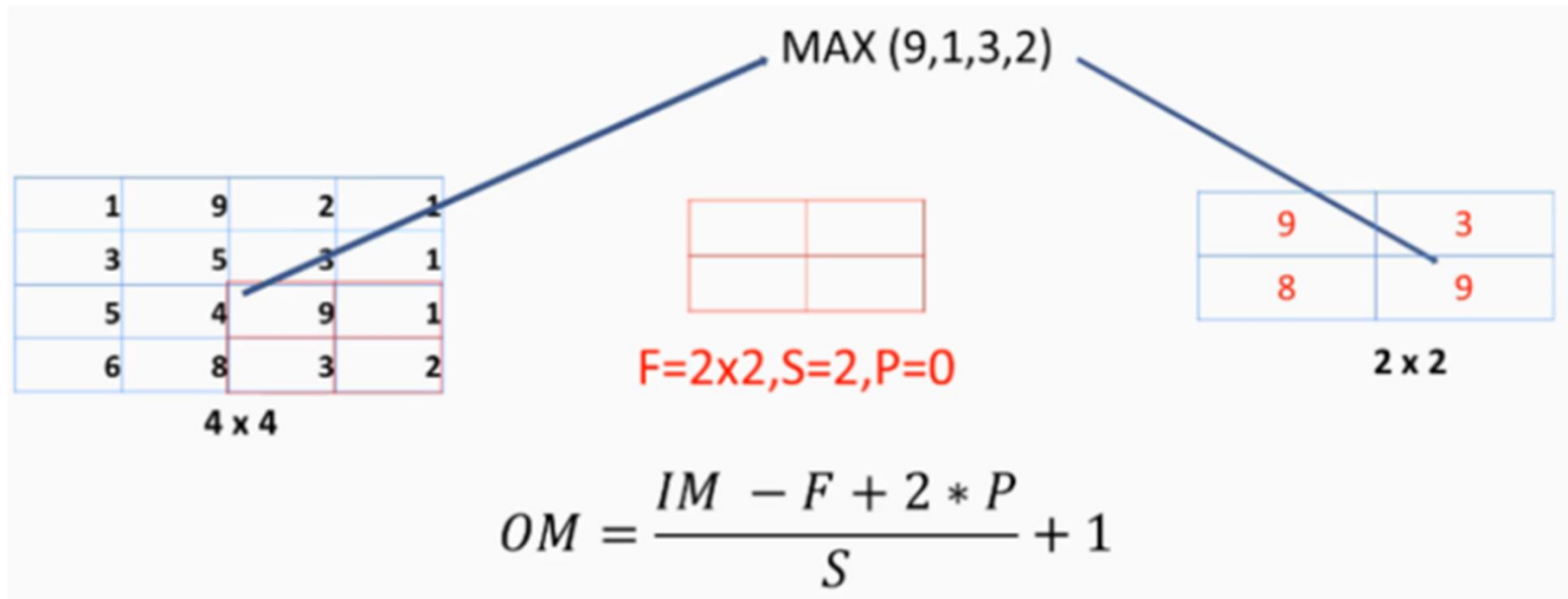
Conv  
3x3 X 3

-	-	-
1530	2295	2295
1470	2205	2205
60	90	60

3 x 3 X 1

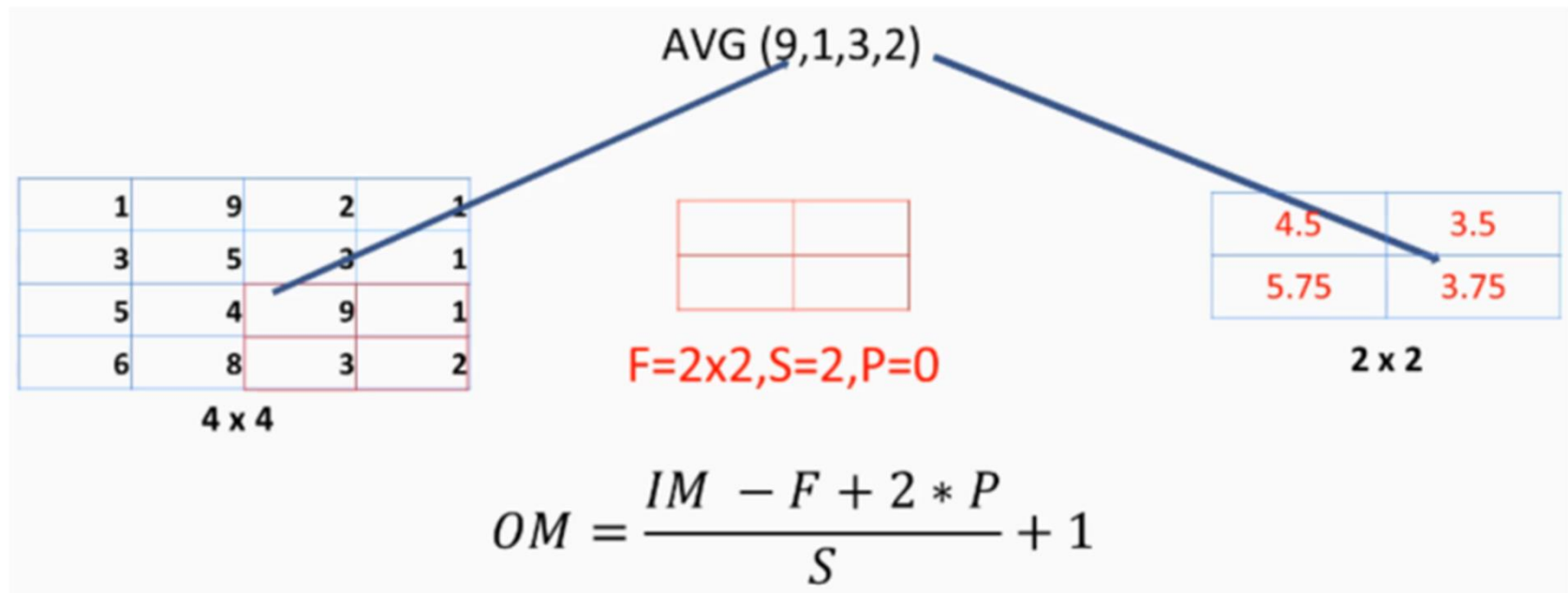
# Convolutional Neural Network

- Các tham số của convolution layer?
  - Max pooling:



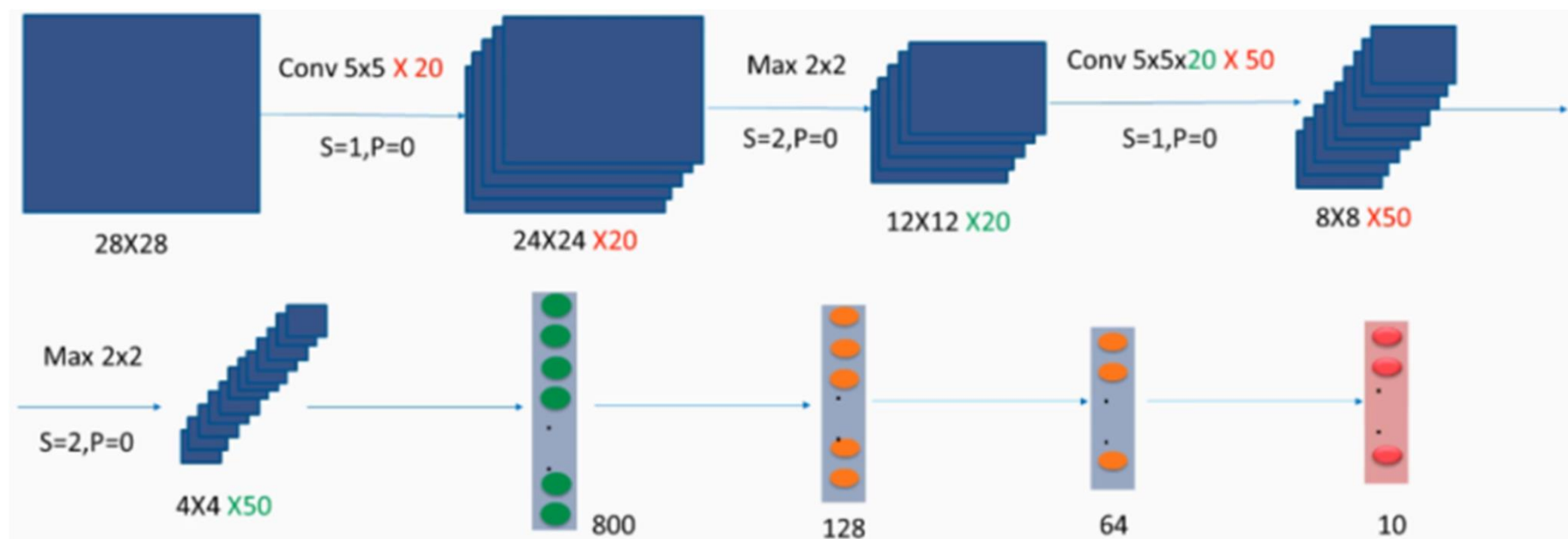
# Convolutional Neural Network

- Các tham số của convolution layer?
  - Average pooling:



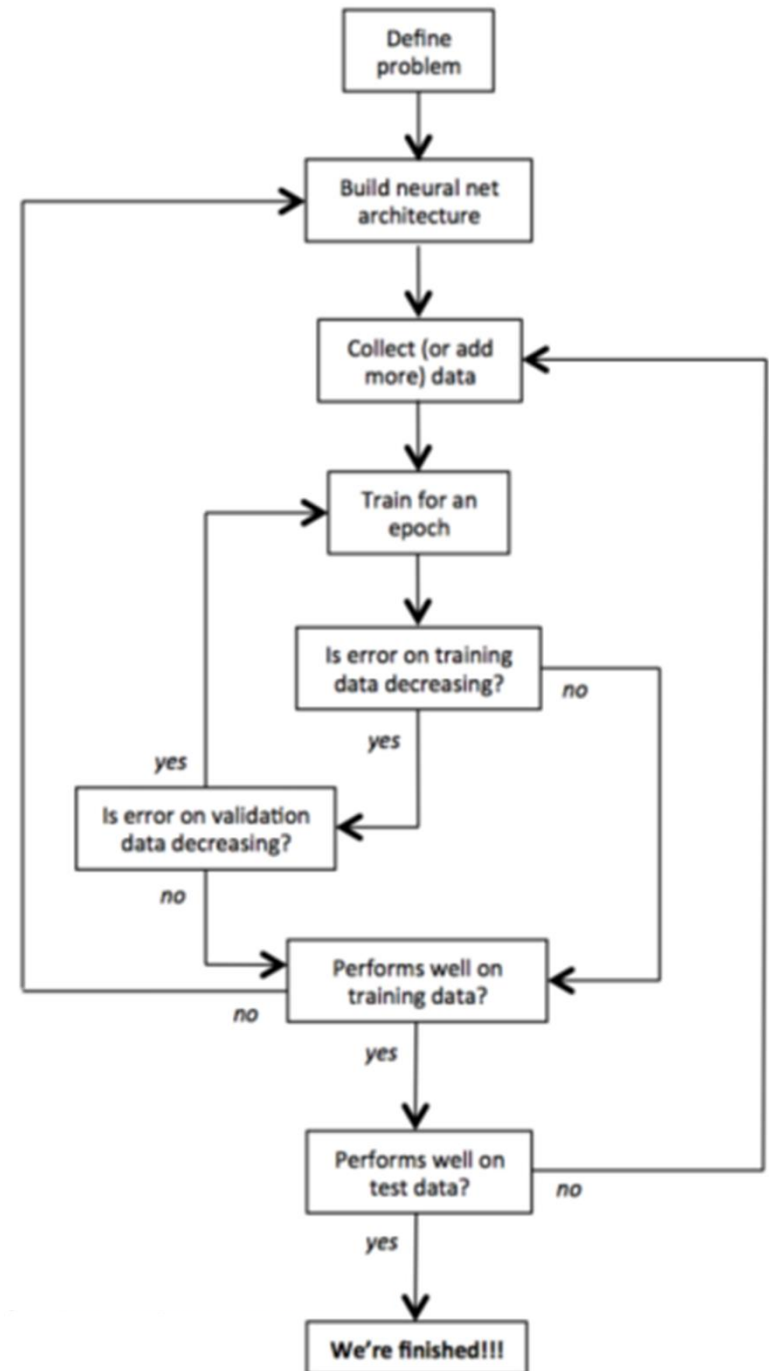
# Convolutional Neural Network

- Ví dụ neural network được thiết kế bằng CNN



# ANN

- Tóm tắt qui trình thiết kế artificial neural network



Q / A