

# Convolutional Neural Networks



VietAI Teaching Team

VietAI ML Foundation Class 5 - Lecture 9



## Nội Dung

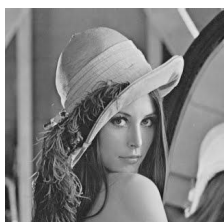
1. Giới thiệu
2. Convolutional neural networks (CNN)
3. CNN Tensorflow

## Nội Dung

1. Giới thiệu
2. Convolutional neural networks (CNN)
3. CNN Tensorflow

## Giới Thiệu - Các loại ảnh phổ biến

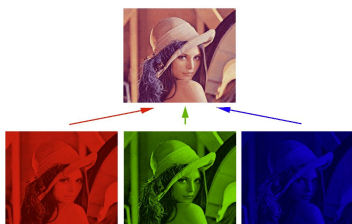
Ảnh xám



0	120	3	0	6
100	0	43	2	128
2	1	255	0	1
34	0	4	5	7
1	1	56	0	1

height

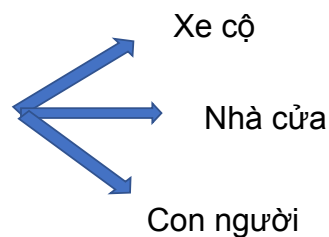
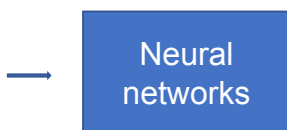
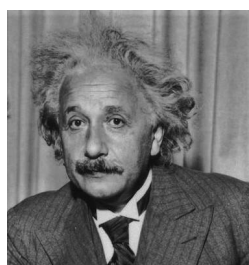
Ảnh RGB



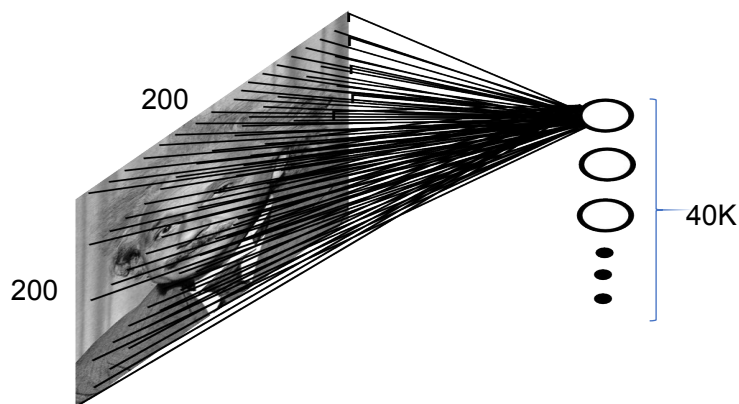
0	120	3	0	6		
100	0	120	3	0	6	
2	100	0	120	3	0	6
34	2	100	0	43	2	128
1	34	2	1	255	0	1
	1	34	0	4	5	7
		1	1	56	0	1

depth

## Giới Thiệu - Bài toán phân loại cho ảnh



## Giới Thiệu - Mạng neuron fully connected

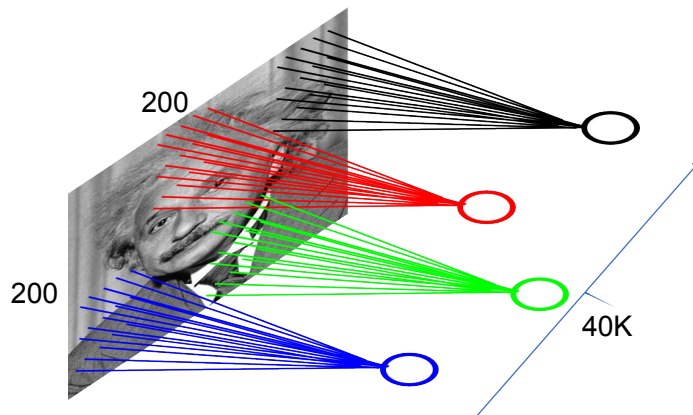


Mỗi hidden node kết nối với tất cả pixel  
Số lượng tham số của tầng này :

$$200 \times 200 \times 40K = 1.6B$$

**Quá nhiều tham số!!!**

## Giới Thiệu - Mạng neuron locally connected

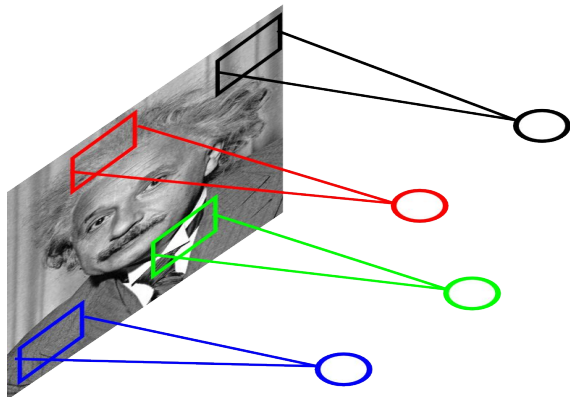


Mỗi hidden node chỉ kết nối với một vùng pixel **cố định** có kích thước 10x10  
Số lượng tham số của tầng :

$$200*200*10*10 = 4M$$

Số lượng tham số giảm nhiều so với fully-connected

## Giới thiệu - Mạng neuron locally connected



**Hạn chế!!!:** Chỉ phù hợp với ảnh **có bố cục rõ ràng**, chẳng hạn như ảnh khuôn mặt người.

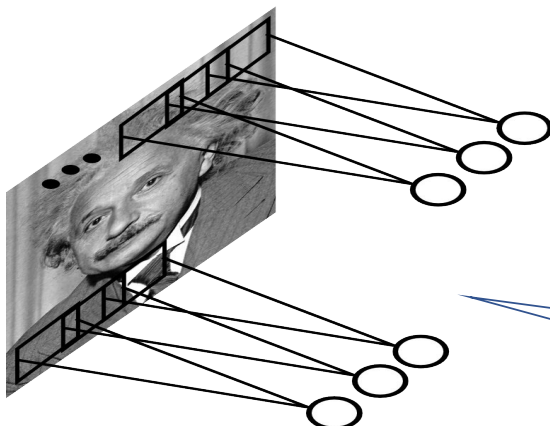
Làm thế nào để áp dụng cho những ảnh thông thường?

→ Mỗi hidden node cần kết nối nhiều vùng pixel trên một bức ảnh

→ **Chia sẻ tham số**



## Giới thiệu Mạng neuron weight-sharing locally connected



Tham số **được chia sẻ** cho các vùng ảnh khác nhau

Phép convolution

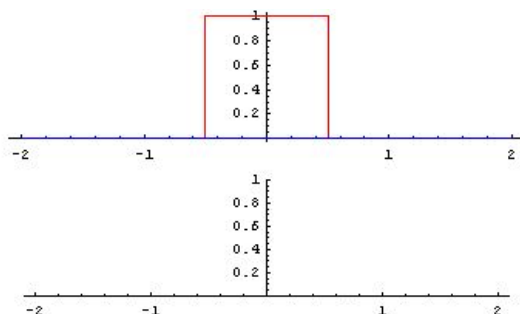


## Giới Thiệu - Mathematical convolution cho hàm liên tục

f

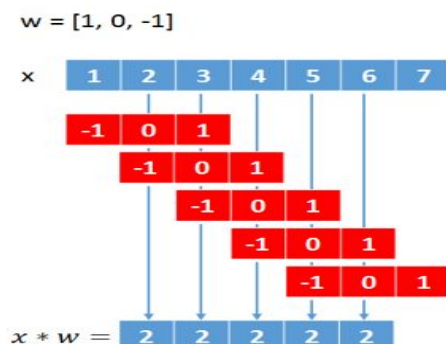
g

$f * g$



Công thức:  $(f * g)(t) = \int_{-\infty}^{+\infty} f(m)g(t - m) dm$

## Giới Thiệu - Mathematical convolution cho hàm rời rạc



## Giới Thiệu - Phép convolution trong xử lý ảnh

- Edge detection



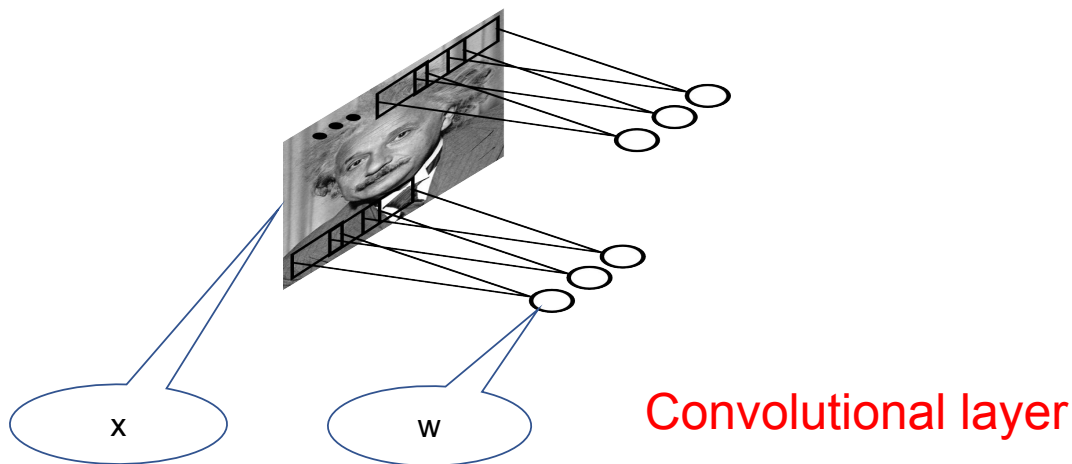
\*

1	0	-1
2	0	-2
1	0	-1

Sobel filter



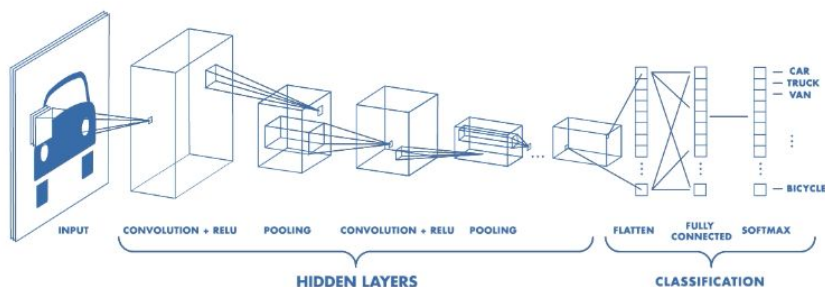
## ● Giới Thiệu - Phép convolution trong thị giác máy tính



## ● Nội dung

1. Giới thiệu
2. Convolutional neural networks (CNN)
3. CNN Tensorflow

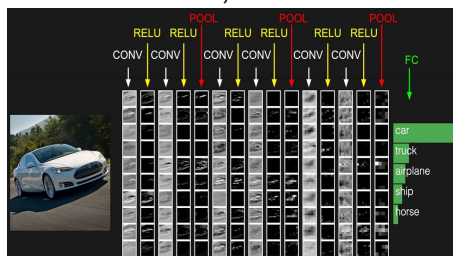
## CNN - Kiến trúc phổ dụng



## CNN - Kiến trúc phổ dụng

- Tầng convolutional
  - Tầng activation (ReLU)
  - Tầng pooling (optional)
- } Lặp k lần
- } Lặp n lần
- Tầng fully connected

$k = 2, n = 3$



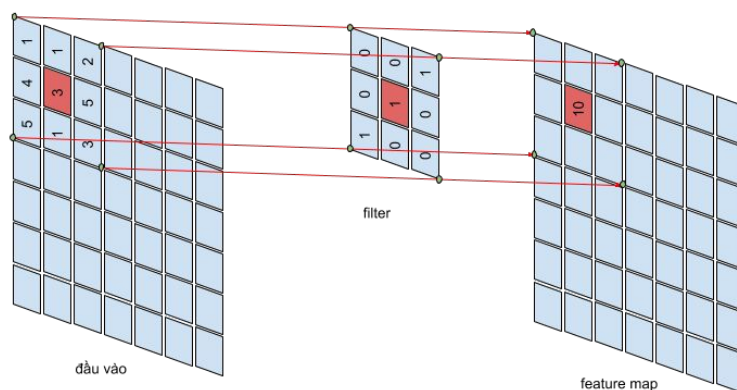


## CNN - Kiến trúc phổ dụng

- Tầng convolutional
  - Tầng activation (ReLU)
  - Tầng pooling (optional)
- } Lặp k lần
- } Lặp n lần
- Tầng fully connected

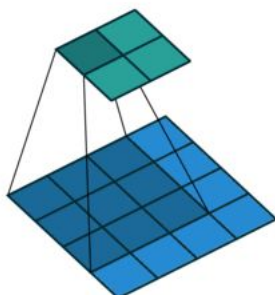
## Tầng Convolutional - 2D convolution

- Phép convolution tại một vị trí của bức ảnh



## Tầng Convolutional - 2D convolution

- Trượt filter theo 2 chiều width và height của ảnh



## Tầng Convolutional - 2D convolution

- Trượt filter theo 2 chiều width và height của ảnh

Ảnh 5x5

2	0	3	0	1
1	0	4	2	1
3	1	1	0	1
1	0	4	5	0
0	1	2	0	1

\*

Filter 3x3

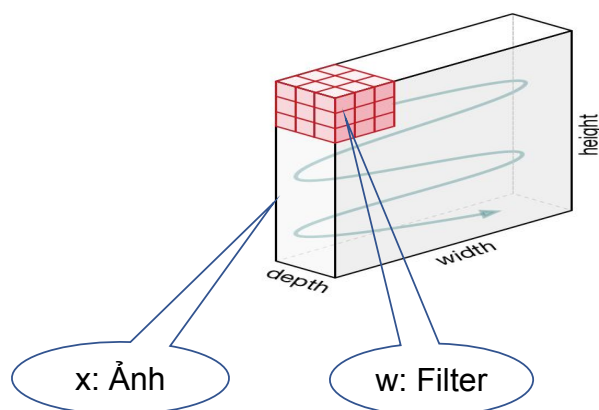
0	1	0
0	1	0
0	1	0

=

Feature map 3x3

1	8	2
1	9	7
2	7	5

## Tầng convolutional - 2D Convolution cho ảnh RGB



## Tầng Convolutional - Stride

- Stride (S) là độ dài bước trượt của filter

Ảnh 5x5

2	0	3	0	1
1	0	4	2	1
3	1	1	0	1
1	0	4	5	0
0	1	2	0	1

Filter 3x3

0	1	0
0	1	0
0	1	0

\*

=

Feature map 3x3

1	8	2
1	9	7
2	7	5

S = 1

Feature map 2x2

1	2
2	5

=

S = 2

## Tầng Convolutional - Padding

- Khi dùng convolution, thông tin ở biên bức ảnh bị mất...

Ảnh 3x3                      Filter 3x3                      Feature map 1x1

0	4	2
1	1	0
3	4	5

\*

0	1	0
0	1	0
0	1	0

=

9
---

...và kích thước của ảnh giảm nhanh → **Padding**

## Tầng Convolutional - Padding

- Thêm **các số 0** vào **biên** của bức ảnh

Ảnh 3x3 được thêm pad                      Filter 3x3                      Feature map 3x3

0	0	0	0	0
0	0	4	2	0
0	1	1	0	0
0	3	4	5	0
0	0	0	0	0

\*

0	1	0
0	1	0
0	1	0

=

1	5	2
4	9	7
4	5	6

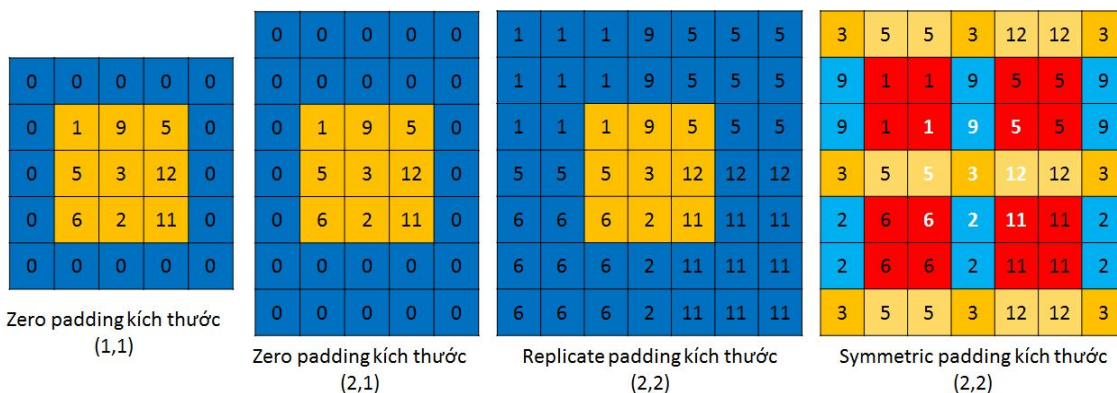
Ảnh  $W \times H$ , filter  $F \times F$ , stride  $S$ , padding  $P$

Feature map  $W' \times H'$ , trong đó:

$$W' = \left\lfloor \frac{W + 2P - F}{S} \right\rfloor + 1, H' = \left\lfloor \frac{H + 2P - F}{S} \right\rfloor + 1$$

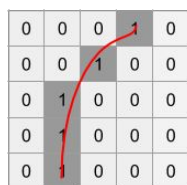
## Tầng Convolutional - Padding

- Một số loại padding khác

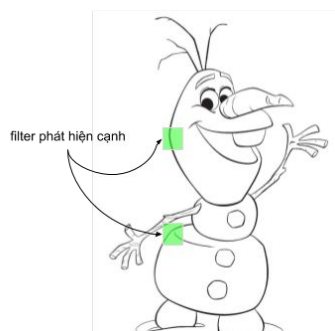


## Tầng Convolutional - Feature detector

- Convolution được xem như là feature detector



filter phát hiện cạnh



## Tầng Convolutional - Tổng kết

- Đầu vào: Tensor có kích thước  $H \times W \times D$
- Siêu tham số:
  - Số filter:  $K$
  - Kích thước của một filter:  $F \times F \times D$
  - Stride của filter:  $S$
  - Padding:  $P$
- Tham số học:
  - $K \times F \times F \times D$
- Đầu ra: Tensor có kích thước  $H' \times W' \times K$ 
  - $H' = \left\lfloor \frac{H+2P-F}{S} \right\rfloor + 1$
  - $W' = \left\lfloor \frac{W+2P-F}{S} \right\rfloor + 1$

## Tầng Convolutional - Bài tập

- Cho ảnh RGB 5x5x3 qua tầng convolutional có 2 filter 3x3x3, với padding  $P = 1$  và stride  $S = 2$ . Tính output.

Ảnh

0	0	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	1	1	0
0	0	1	1	0

0	1	1	1	0
0	1	2	2	0
0	1	1	1	0
0	1	2	2	0
0	1	1	1	0

1	1	1	1	1
0	1	1	1	0
0	0	1	0	0
0	1	1	1	0
1	1	1	1	1

Filter 1

1	-1	0
1	-1	0
1	-1	0

0	1	-1
0	1	-1
0	1	-1

-1	0	1
-1	0	1
-1	0	1

Filter 2

1	1	1
-1	-1	-1
0	0	0

0	0	0
1	1	1
-1	-1	-1

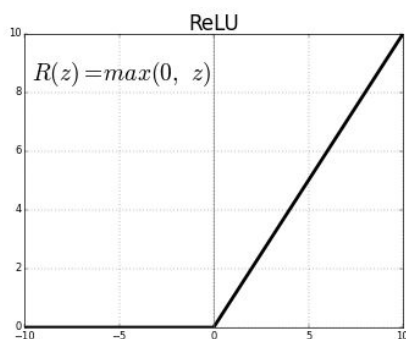
-1	-1	-1
0	0	0
1	1	1

## CNN - Kiến trúc phổ dụng

- Tầng convolutional
  - Tầng activation (ReLU)
  - Tầng pooling (optional)
- } Lặp k lần
- } Lặp n lần
- Tầng fully connected

## Tầng Activation - Hàm ReLU

- Trong cấu trúc của CNN, tầng ReLU thường theo sau tầng Convolutional
- Ngoài hàm ReLU, ta có thể dùng các hàm activation khác như sigmoid, tanh...



## Tầng ReLU - Bài tập

- Cho input 3x3x2 như hình bên dưới. Tính output khi áp dụng tầng activation ReLU

0	-1	0
-1	-1	1
0	-2	0

1	-1	-1
0	-2	-1
0	0	0

## CNN - Kiến Trúc

- Tầng convolutional
  - Tầng activation (ReLU)
  - Tầng pooling (optional)
- } Lặp k lần
- } Lặp n lần
- 
- Tầng fully connected
  - Tầng ReLU
- } Lặp m lần
- 
- Tầng fully connected



## Tầng Pooling - Max pooling, average pooling

- Stride thường có kích thước bằng với kích thước filter
- Filter được áp dụng độc lập trên chiều depth
- Giảm chiều feature map, vị trí tuyệt đối của các đặc trưng không còn quan trọng

2	3	1	4
1	2	4	8
8	9	6	5
9	1	7	4

đầu vào

3	8
9	7

sau khi pooling

## Tầng Pooling - Tổng kết

- Đầu vào: Tensor có kích thước  $H \times W \times D$
- Siêu tham số:
  - Kích thước của filter:  $F \times F$
  - Stride của filter:  $S$
- Tham số học: Tầng pooling **không có tham số học**
- Đầu ra: Tensor có kích thước  $H' \times W' \times D$ , trong đó
  - $H' = \frac{H-F}{S} + 1$
  - $W' = \frac{W-F}{S} + 1$

## Tầng Pooling - Bài tập

- Cho input 3x3x2. Tính output của tầng max pooling với filter 2x2 và stride  $S = 1$

0	0	0
0	0	1
0	0	0

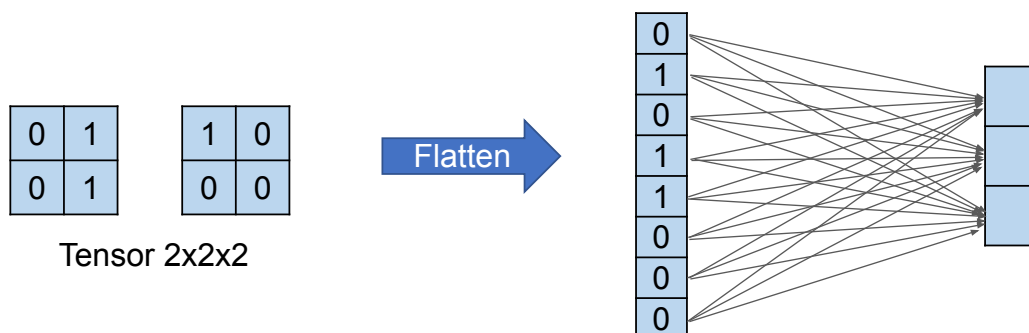
1	0	0
0	0	0
0	0	0

## CNN - Kiến trúc phổ dụng

- Tầng convolutional
  - Tầng activation (ReLU)
  - Tầng pooling (optional)
- } Lặp k lần
- } Lặp n lần
- Tầng fully connected

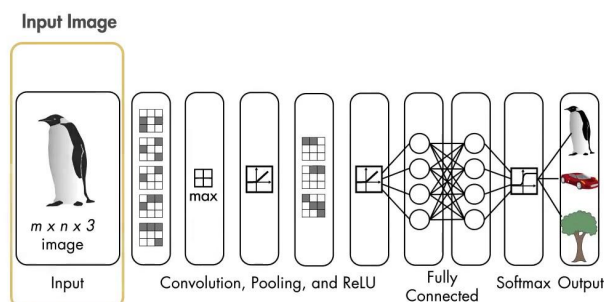
## Tầng Fully Connected - Flattening

- Flattening là quá trình “duỗi” tensor 3D thành tensor 1D để làm đầu vào cho tầng fully connected
- Ví dụ:



## Tầng Fully Connected - Bài tập

- Thiết kế cấu trúc CNN cho bài toán phân loại 3 tầng với dữ liệu ảnh 78x78x3



- Tính chất
  - Translation invariance
  - Weight sharing

- Ứng dụng

