

# Convolutional Neural Networks



VietAI Teaching Team

# Nội dung

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

# Nội dung

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

# Giới thiệu - Raster images

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

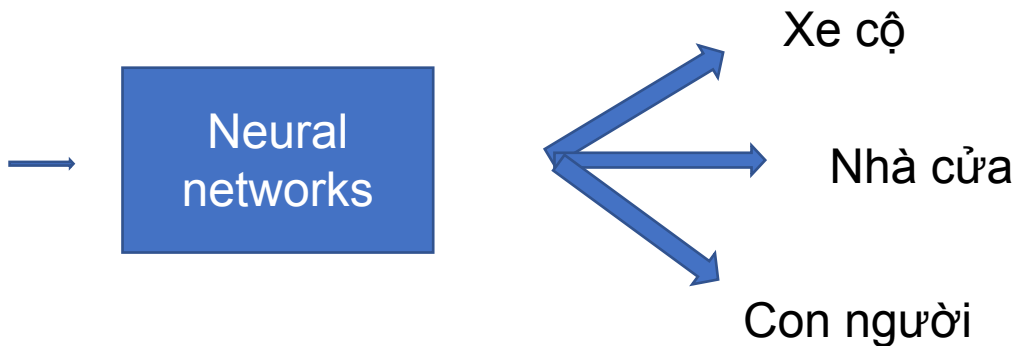
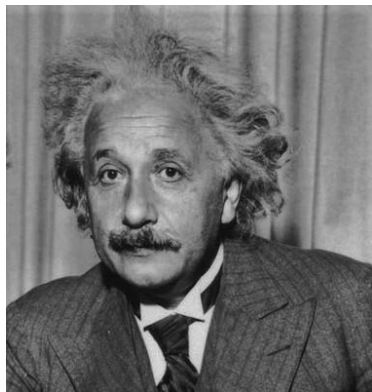
Ảnh RGB



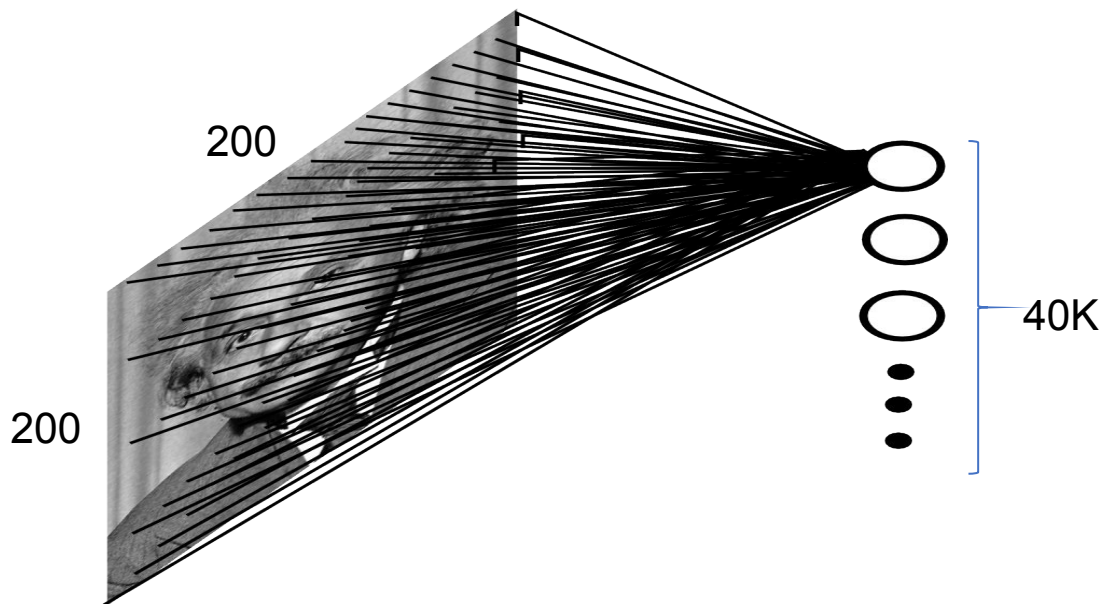
0	120	3	0	6		
100	0	120	3	0	6	
2	100	0	120	3	0	6
34	2	100	0	43	2	128
1	34	2	1	255	0	1
	1	34	0	4	5	7
		1	1	56	0	1

depth

# Giới thiệu - Bài toán phân loại ảnh



# Giới thiệu - Mạng neuron fully connected



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

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

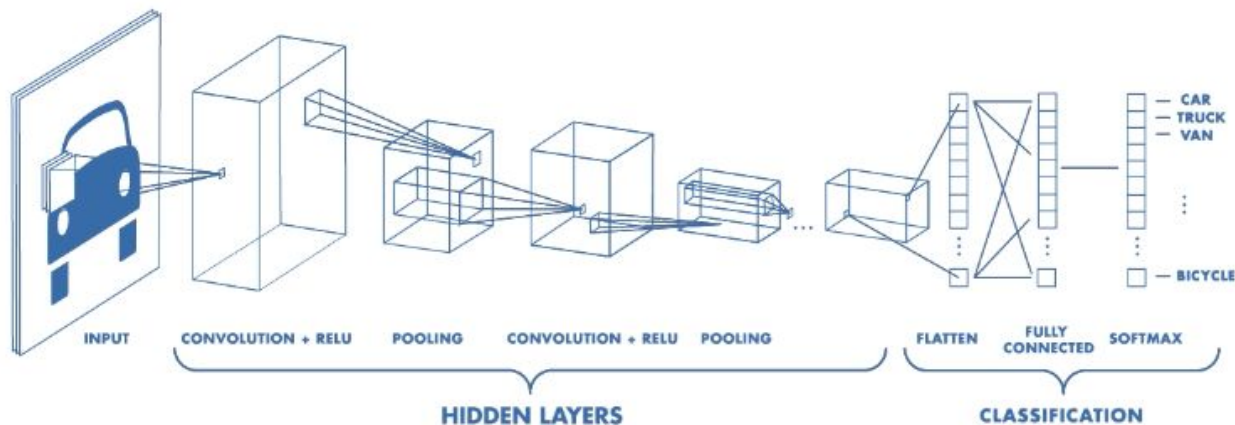
Quá nhiều tham số!!!

Giải pháp? CNN!

# Nội dung

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

# CNN - Kiến trúc phổ dụng



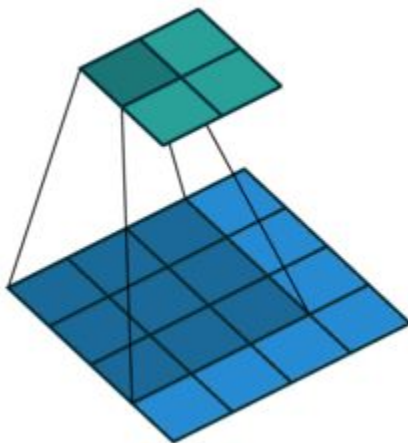


# CNN - Kiến trúc phổ dụng

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

# Tầng convolutional - 2D convolution

- Trượt filter theo 2 chiều width và height của ảnh grayscale



# Tầng convolutional - 2D convolution

- Ví dụ:

1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

Convolved  
Feature

Filter/Kernel 3x3

1	0	1
0	1	0
1	0	1

# Tầng convolutional - 2D convolution

- Ví dụ

Ảnh 5x5 input

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/Kernel 3x3

0	1	0
0	1	0
0	1	0

=

Feature map 3x3

1	8	2
1	9	7
2	7	5

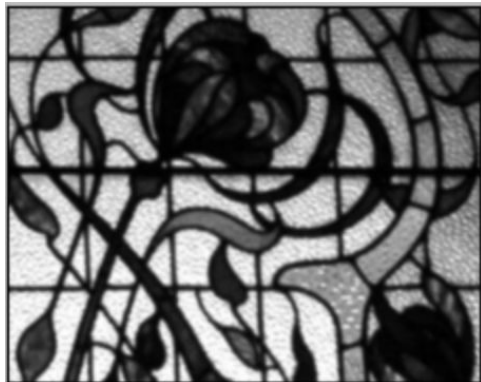
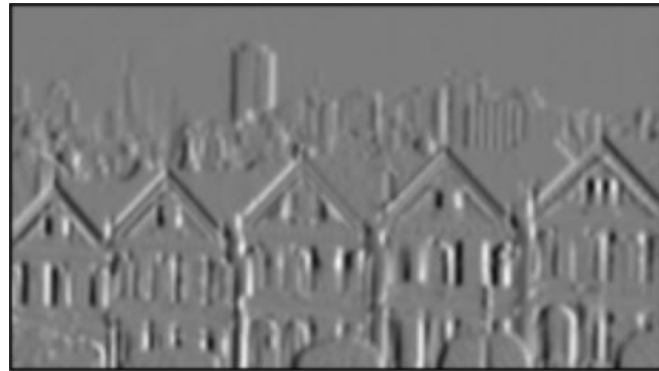
# Ý nghĩa của filter



\*

Sobel filter

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



# Tầng convolutional - 2D convolution

- Ví dụ:

Ảnh 5x5 input

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/Kernel 3x3

0	1	0
0	1	0
0	1	0

\*

=

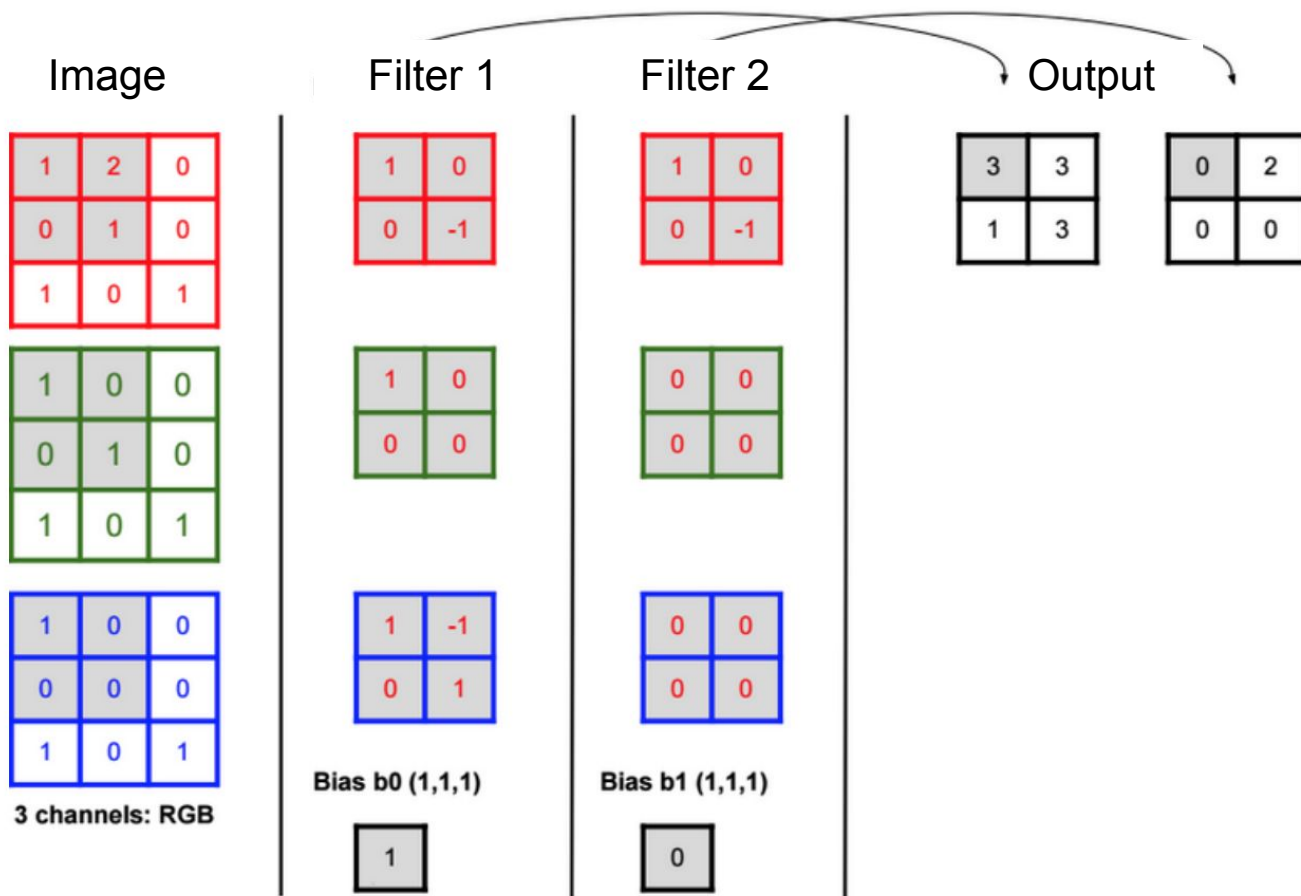
Feature map 3x3

1	8	2
1	9	7
2	7	5

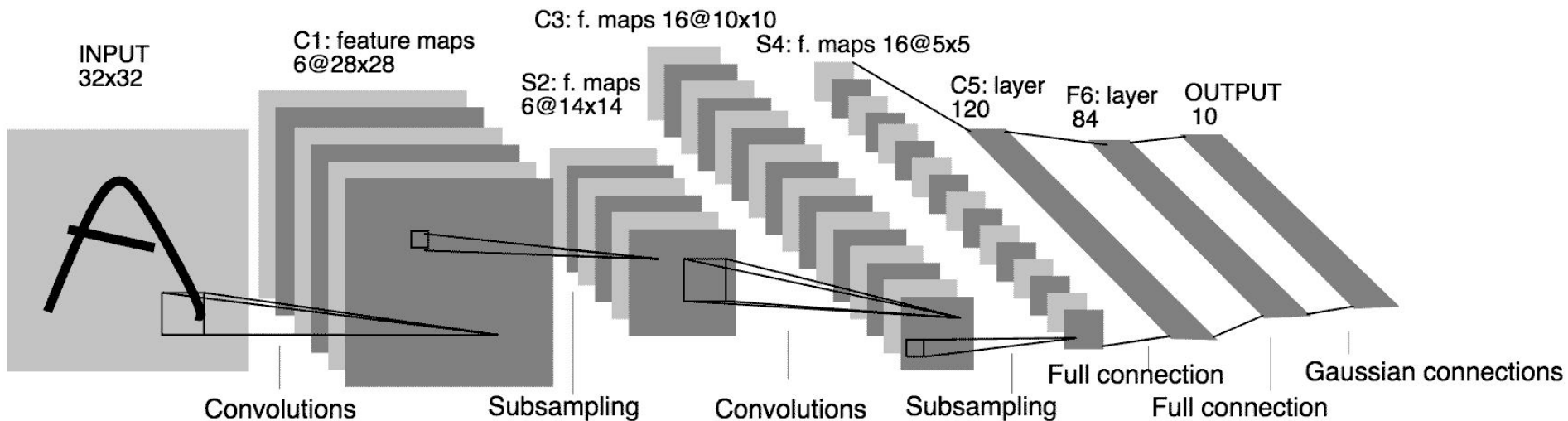
Bias

0
---

# Tầng convolutional - 2D convolution



# Tầng convolutional - Multiple layers





# 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, kích thước của feature map giảm so với input...

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
---

... để giữ nguyên kích thước

→Padding

# Tầng convolutional - Padding

- Thêm các số 0 vào biên của bức ảnh

Ảnh 3x3 được thêm pad

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

Filter 3x3

0	1	0
0	1	0
0	1	0

\*

=

Feature map 3x3

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

0	0	0	0	0
0	1	9	5	0
0	5	3	12	0
0	6	2	11	0
0	0	0	0	0

Zero padding kích thước  
(1,1)

0	0	0	0	0
0	0	0	0	0
0	1	9	5	0
0	5	3	12	0
0	6	2	11	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Zero padding kích thước  
(2,1)

1	1	1	9	5	5	5
1	1	1	9	5	5	5
1	1	1	9	5	5	5
5	5	5	3	12	12	12
6	6	6	2	11	11	11
6	6	6	2	11	11	11
6	6	6	2	11	11	11

Replicate padding kích thước  
(2,2)

3	5	5	3	12	12	3
9	1	1	9	5	5	9
9	1	1	9	5	5	9
3	5	5	3	12	12	3
2	6	6	2	11	11	2
2	6	6	2	11	11	2
3	5	5	3	12	12	3

Symmetric padding kích thước  
(2,2)

# Tầng convolutional - Tính chất và Ưu điểm

- Tính chất:
  - Locally connected
  - Shared weights
- Ưu điểm:
  - Giữ được kết cấu của bức ảnh so với việc duỗi ma trận thành vector
  - Giảm số tham số phải học
  - Equivariance to translation

# Tầng convolutional - Tổng kết

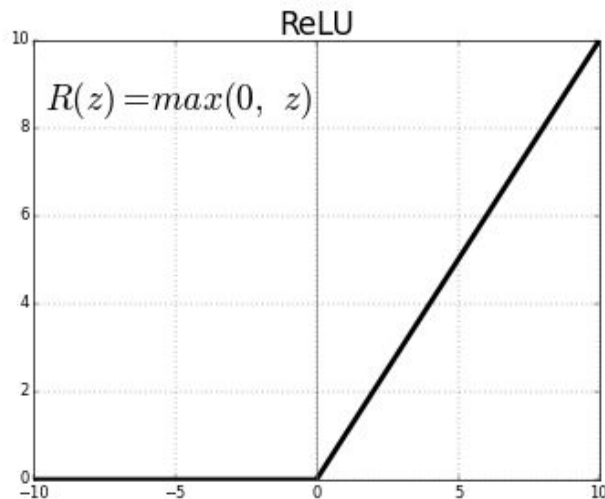
- Đầu vào: Tensor có kích thước  $H \times W \times D$
- Siêu tham số:
  - Số filter:  $K$
  - Kích thước của một filter:  $F \times F \times D$
  - Stride của filter:  $S$
  - Padding:  $P$
- Tham số học:
  - $K \times F \times F \times D$
- Đầu ra: Tensor có kích thước  $H' \times W' \times K$ 
  -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

# CNN - Kiến trúc phổ dụng

- Tầng convolutional
  - Tầng activation (ReLU)
  - Tầng pooling (optional)
  - Tầng fully connected
- Diagram illustrating the structure of a CNN architecture:
- The first three layers (convolutional, activation, and pooling) are grouped by a bracket labeled "Lặp k lần" (Repeat k times).
- The entire group of the first three layers is further grouped by a larger bracket labeled "Lặp n lần" (Repeat n times).

# 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...





# 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 - Giới thiệu

- Giảm chiều feature map

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

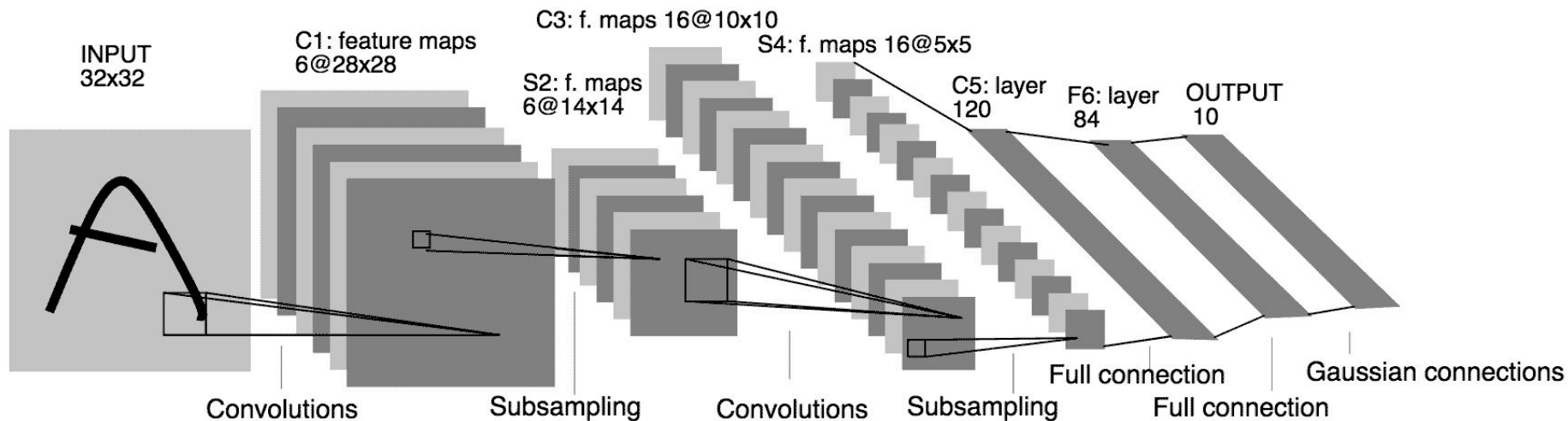
đầu vào

3	8
9	7

$S = 2$

sau khi pooling

# Tầng pooling - Ví dụ



## Tầng pooling - Tổng kết

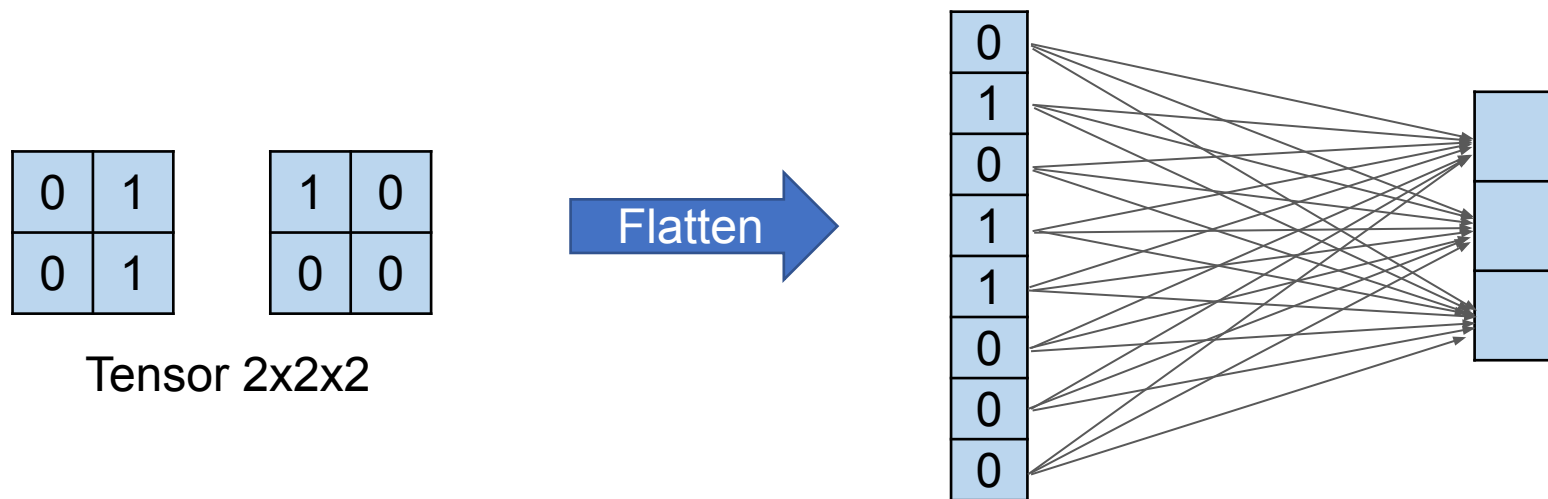
- Đầu vào: Tensor có kích thước  $H \times W \times D$
- Siêu tham số:
  - Kích thước của filter:  $F \times F$
  - Stride của filter:  $S$
- Tham số học: Tầng pooling **không có tham số học**
- Đầu ra: Tensor có kích thước  $H' \times W' \times D$ , trong đó
  - $H' = \frac{H - F}{S} + 1$
  - $W' =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

# CNN - Kiến trúc phổ dụng

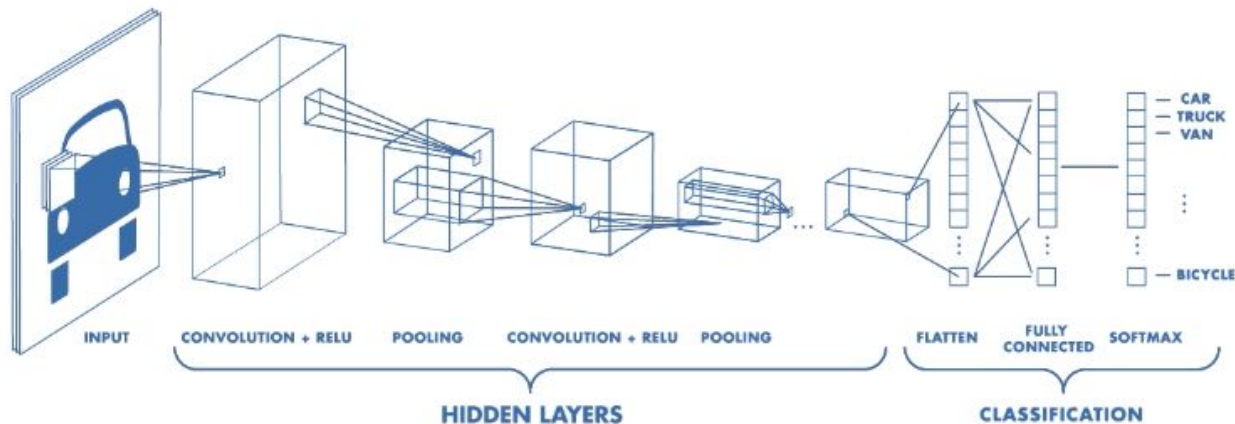
- Tầng convolutional
  - Tầng activation (ReLU)
  - Tầng pooling (optional)
- Lặp k lần
- Lặp n lần
- Tầng fully connected

## Tầng fully connected - Flattening

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



# CNN - Kiến trúc phổ dụng



# CNN - Tổng kết

- Tính chất
  - Locally connected
  - Weight sharing

## • Ứng dụng

