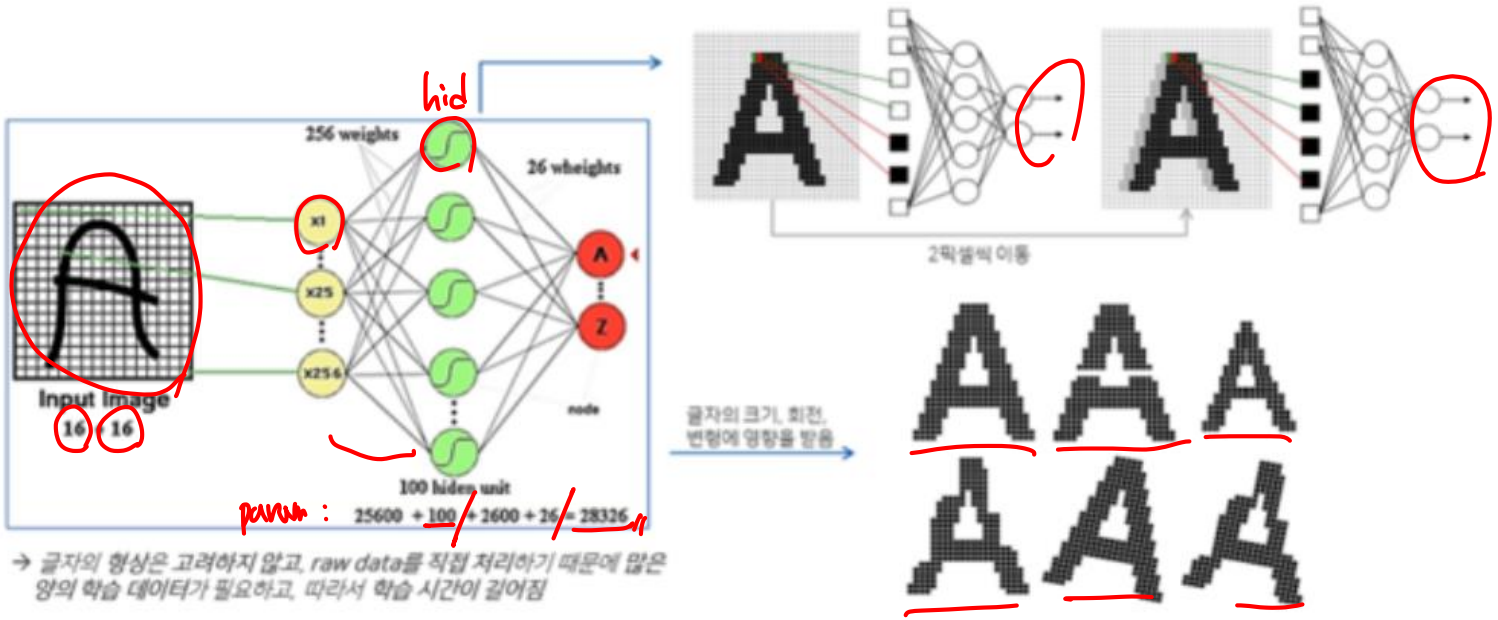
Deep Learning Basic

Jaewon Kim, Dankook Univ.

Chapter 5

Part 1
Convolutional Neural Network

Problem of MLP

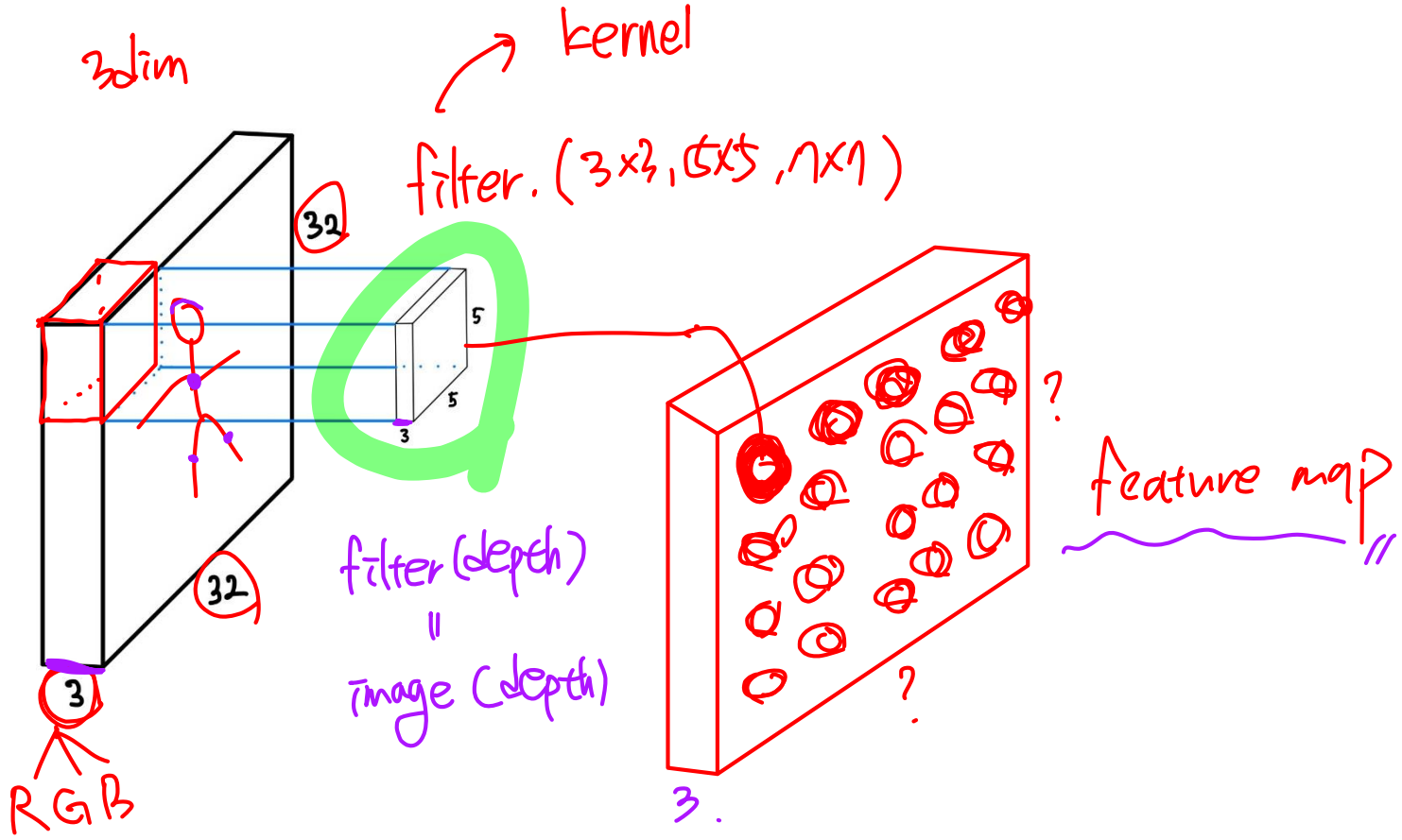


→ 글자의 형상은 고려하지 않고, raw data를 직접 처리하기 때문에 많은 양의 학습 데이터가 필요하고, 따라서 학습 시간이 길어짐

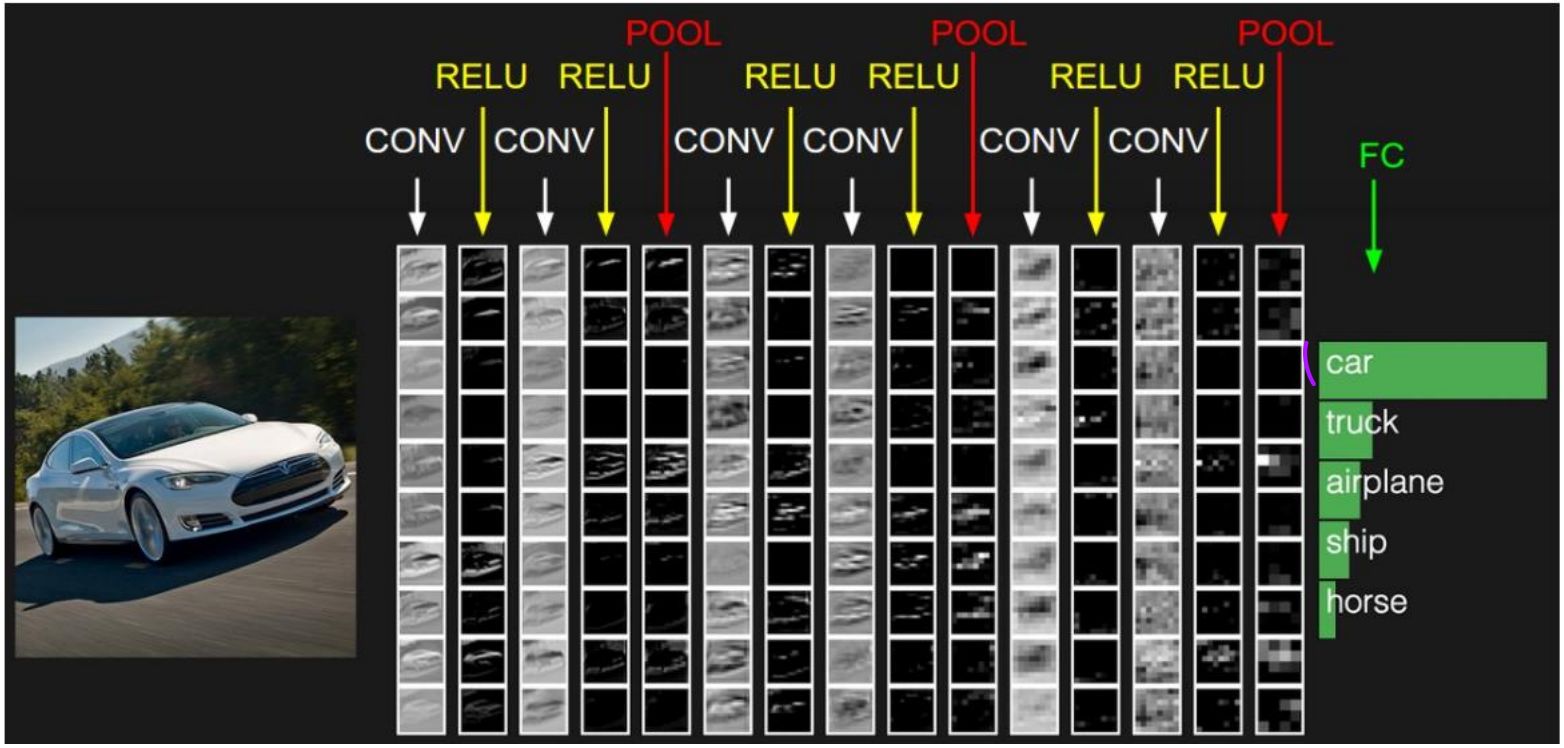
Number of parameter, Training Time, Magnitude of Net

출처 : https://m.blog.naver.com/laonple/221193389981

Convolution Neural Network (CNN)



Convolution Neural Network (CNN)



Convolution Neural Network (CNN)

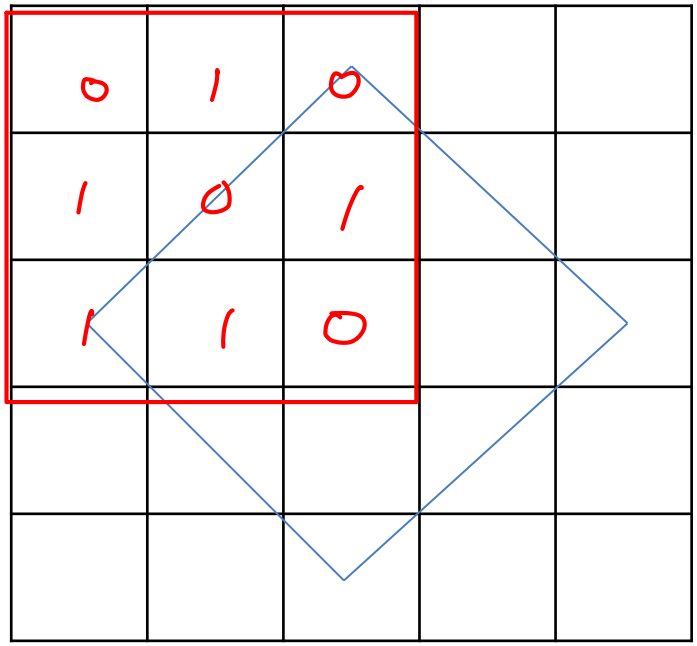
Filter (kernel)

stride = 1

3x3 filter

0	1	0
1	0	1
1	1	0

Image



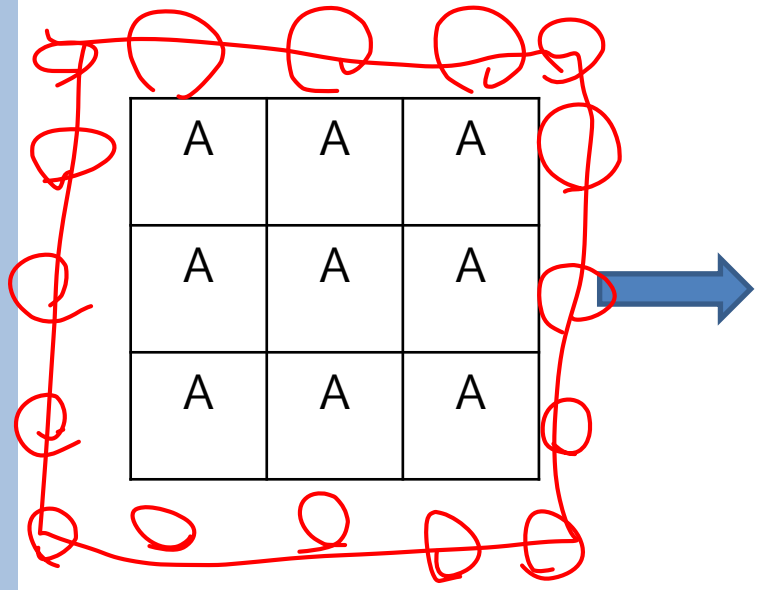
feature (곱셈합)

1 3 0
3 1 3
0 3 1

Padding

이미지의 크기를 유지하기 위함.

padding 이미지 크기가 계속 줄어듦.



0	0	0	0	0
0	A	A	A	0
0	A	A	A	0
0	A	A	A	0
0	0	0	0	0

Padding의 크기 : $\frac{Filter-1}{2}$ ($stride$가 1일 때)

Calculation the output size

$$(OH, OW) = \left(\frac{H + 2P - FH}{S} + 1, \frac{H + 2P - FW}{S} + 1\right)$$

in hight, padding, filter, stride

Ex)

Image

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

5 (세로), 5 (가로)

3x3 filter

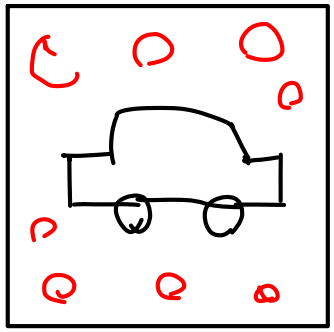
2	1	0
0	0	1
1	2	2

P=1,
S=1

$$OH, OW = \left(\frac{5+2-3}{1} + 1, \frac{5+2-3}{1} + 1\right)$$

$$= (5, 5)$$

Pooling 이미지의 크기를 줄인 샘플링



1	3	3	2
2	1	3	5
7	2	6	2
2	2	4	1

$\frac{1+3+2+1}{4} = \frac{7}{4}$

3	3	5
7	6	6
7	6	6

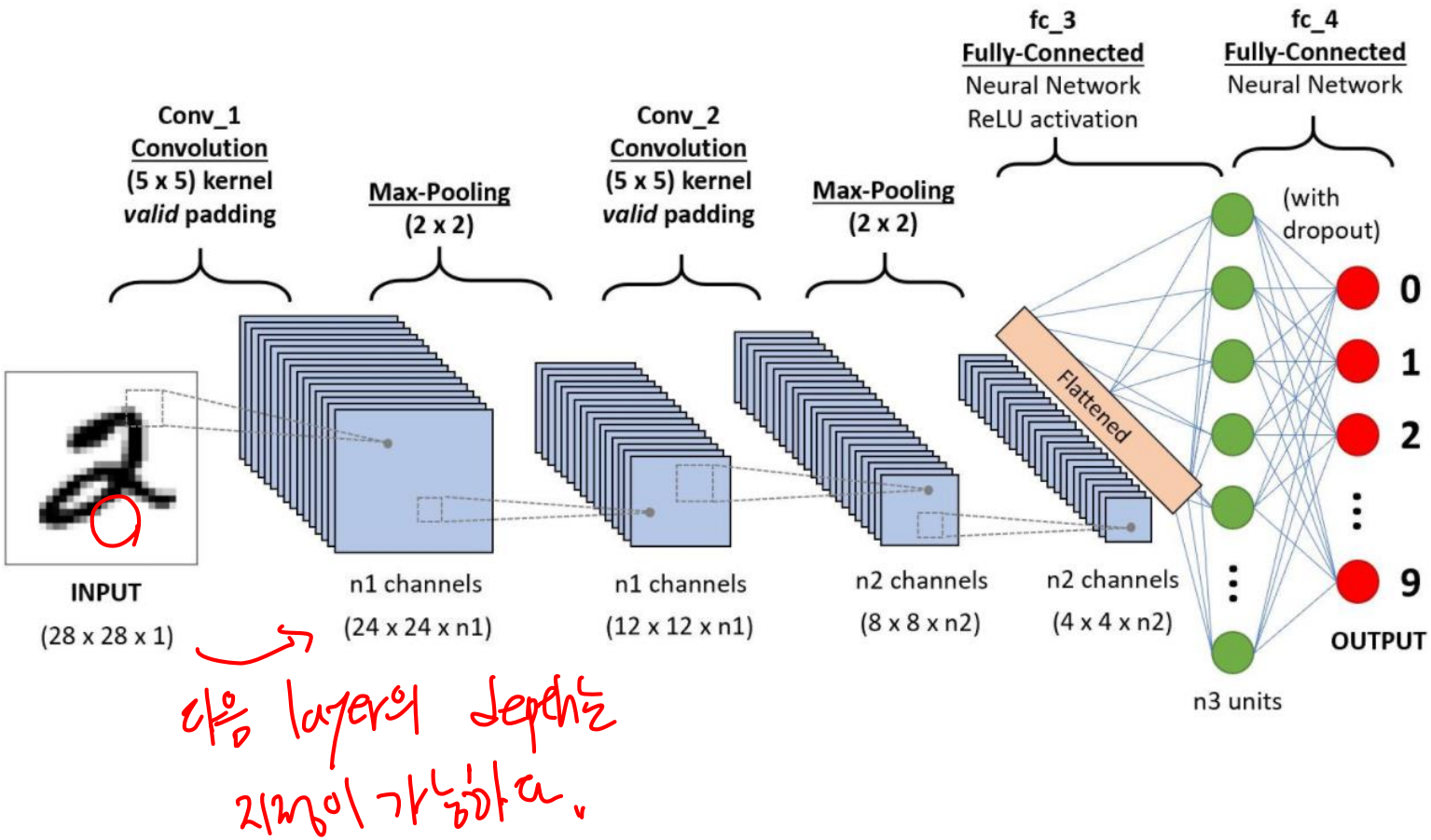
2x2 Maxpooling

1.75	2.5	3.25
3	3	4
3.25	3.5	3.25

2x2 Avgpooling

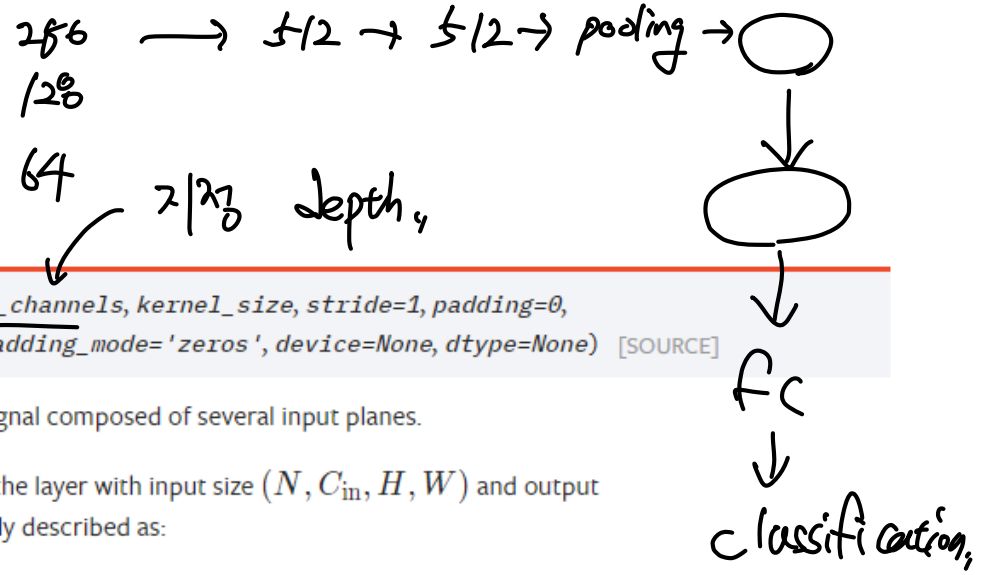
CNN Architecture

MNIST



다음 layer의 depth는
2개씩 감소한다.

Channels (Depth)



CONV2D

CLASS `torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1, padding=0, dilation=1, groups=1, bias=True, padding_mode='zeros', device=None, dtype=None)` [SOURCE]

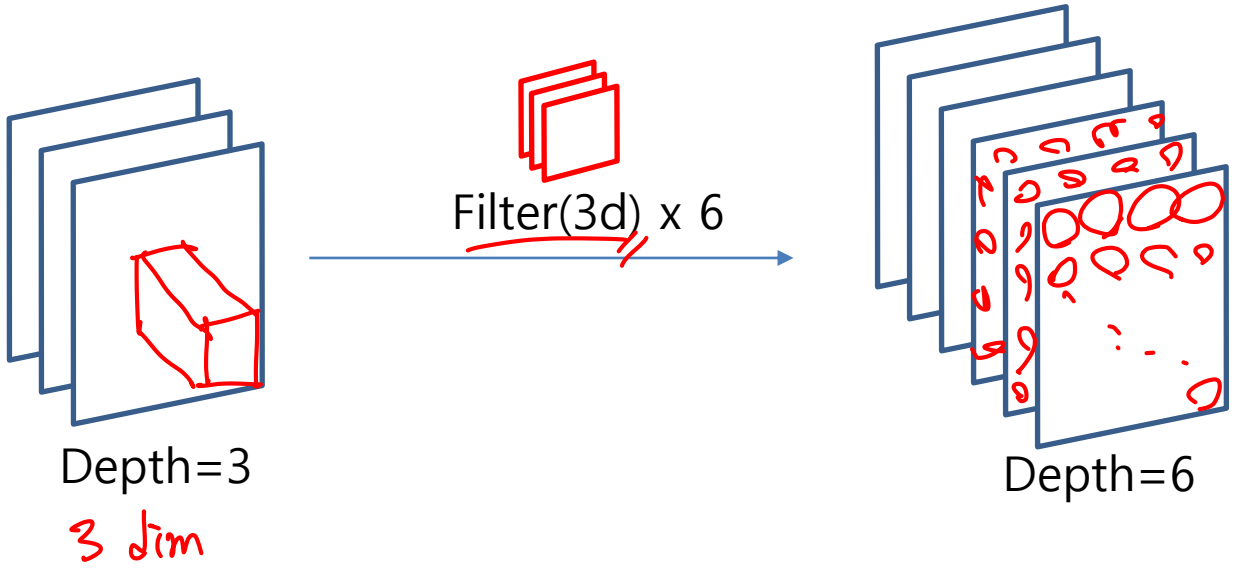
Applies a 2D convolution over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C_{in}, H, W) and output $(N, C_{\text{out}}, H_{\text{out}}, W_{\text{out}})$ can be precisely described as:

$$\text{out}(N_i, C_{\text{out}_j}) = \text{bias}(C_{\text{out}_j}) + \sum_{k=0}^{C_{\text{in}}-1} \text{weight}(C_{\text{out}_j}, k) \star \text{input}(N_i, k)$$

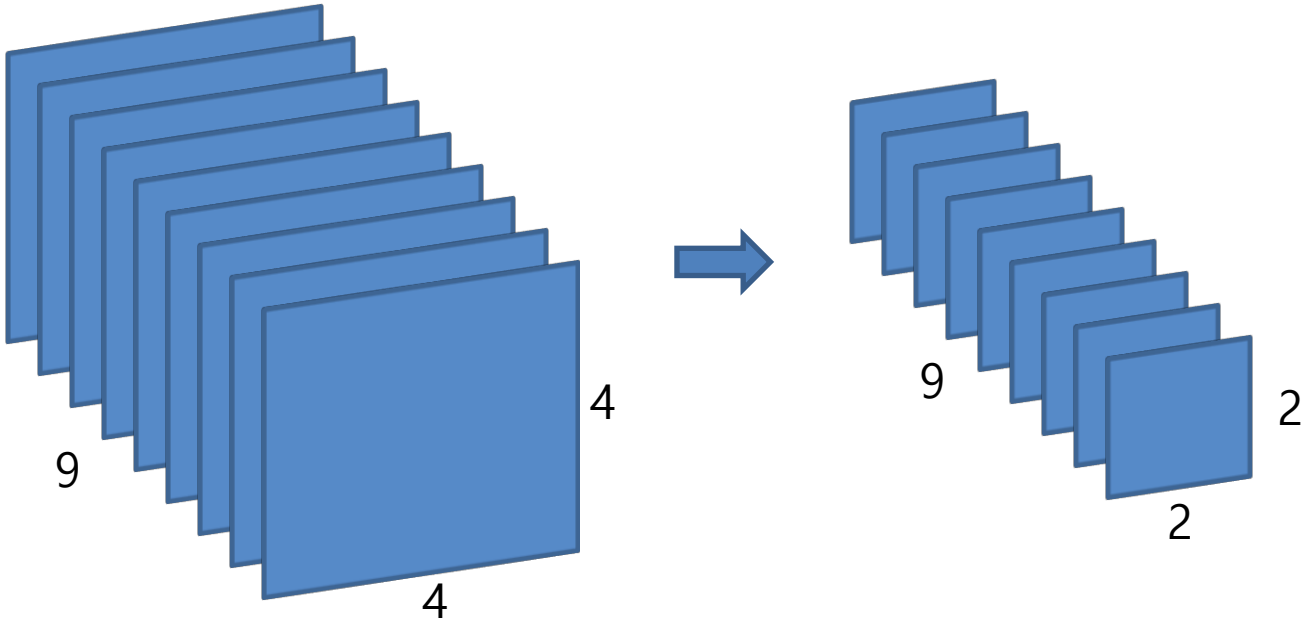
출처 : https://pytorch.org/docs/1.9.1/generated/torch.nn.Conv2d.html

Channels (Depth)



Convolutional layer increases depth

Channels (Depth) - Pooling



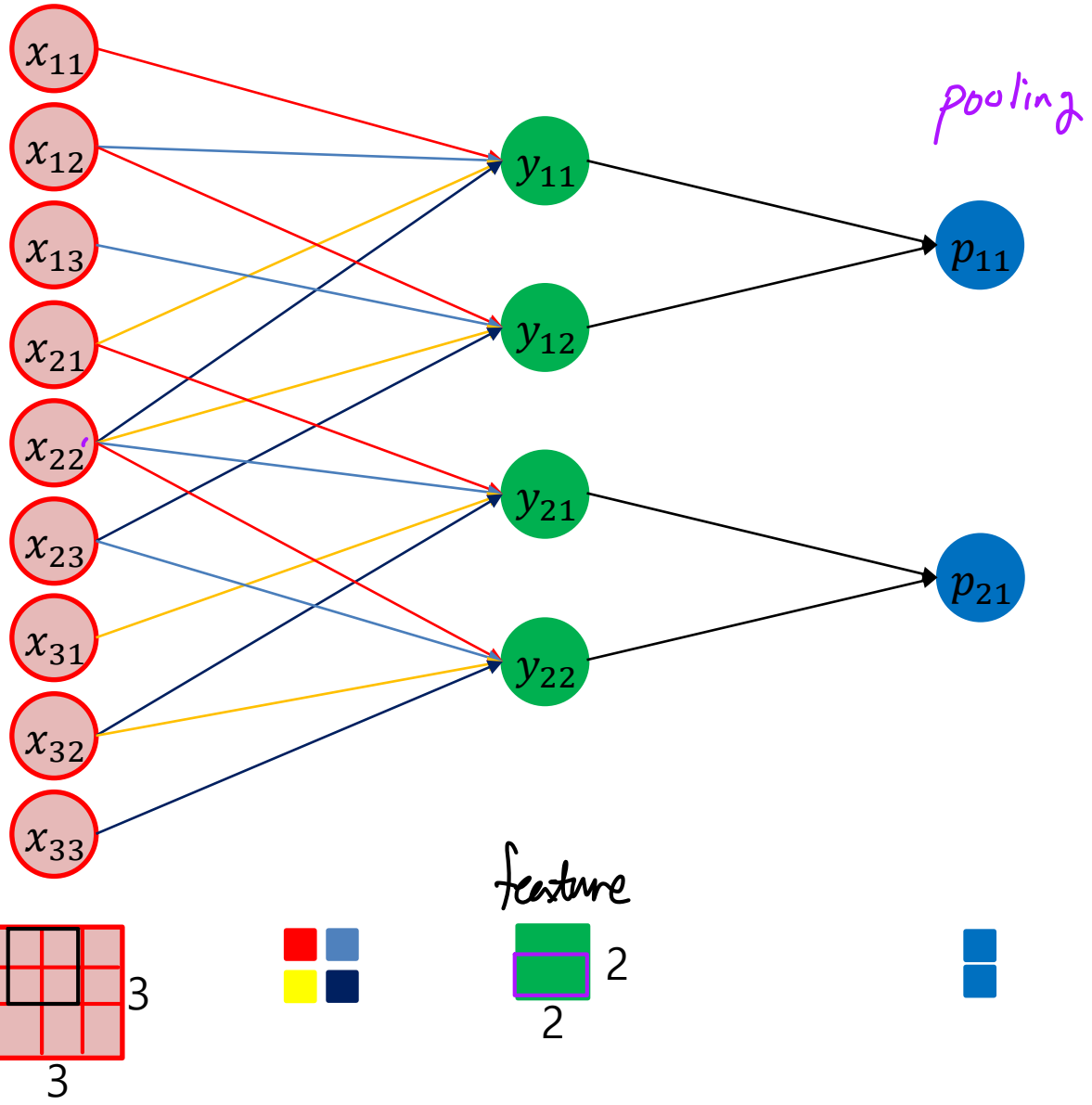
CNN Backpropagation

input

$x_{11}\ x_{12}\ x_{13}$
$x_{21}\ x_{22}\ x_{23}$
$x_{31}\ x_{32}\ x_{33}$

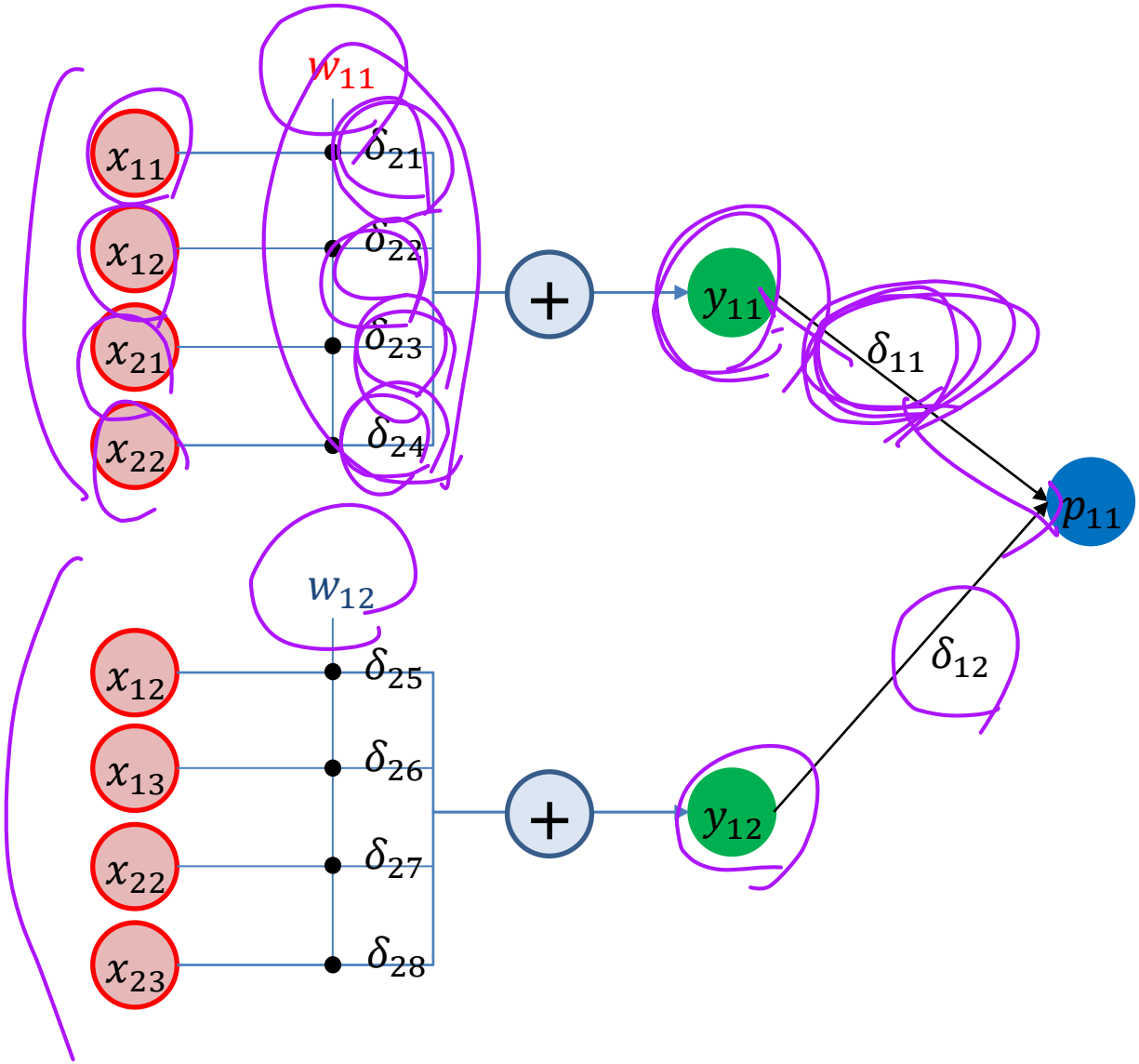
feature

$y_{11}\ y_{12}$
$y_{21}\ y_{22}$



출처:https://ratsgo.github.io/deep%20learning/2017/04/05/CNNbackprop/

CNN Backpropagation



CNN Backpropagation(x_{11})

$$f(x) = maxpool(\delta_{11}, \delta_{12})$$

$$\delta_{11} = \delta_{21} + \delta_{22} + \delta_{23} + \delta_{24}$$

$$\delta_{12} = \delta_{25} + \delta_{26} + \delta_{27} + \delta_{28}$$

$$\frac{\partial P_{11}}{\partial w_{11}} = \frac{\partial P_{11}}{\partial \delta_{11}} \times \frac{\partial \delta_{11}}{\partial w_{11}}$$

$$P_{11} = \delta_{11} + \delta_{12} \longrightarrow \frac{\partial P_{11}}{\partial \delta_{11}} = 1$$

$$\delta_{11} = \delta_{21} + \delta_{22} + \delta_{23} + \delta_{24}$$

$$= x_{11}w_{11} + x_{12}w_{11} + x_{21}\ w_{11} + x_{22}\ w_{11}$$

$$\frac{\partial \delta_{11}}{\partial w_{11}} = x_{11} + x_{12} + x_{21} + x_{22}$$

$$\therefore \frac{\partial P_{11}}{\partial w_{11}} = 1 \times (x_{11} + x_{12} + x_{21} + x_{22})$$

Thank you..!!!