CNN

Convolution Neural Network

BOAZ 16기 분석 이다인

Computer Vision and Deep learning

컴퓨터 비전은 딥러닝의 발전 영역에 있어서 가장 두각을 나타내는 분야 중 하나

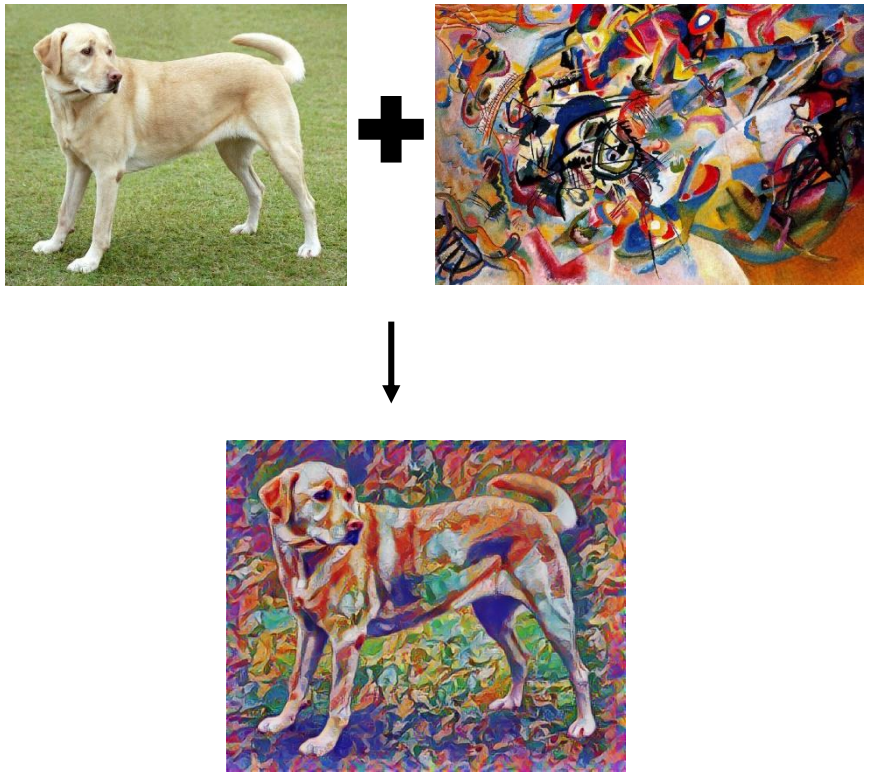
EXAMPLE

- Image Classification

Cat ?
(0/1)

- Object Detection

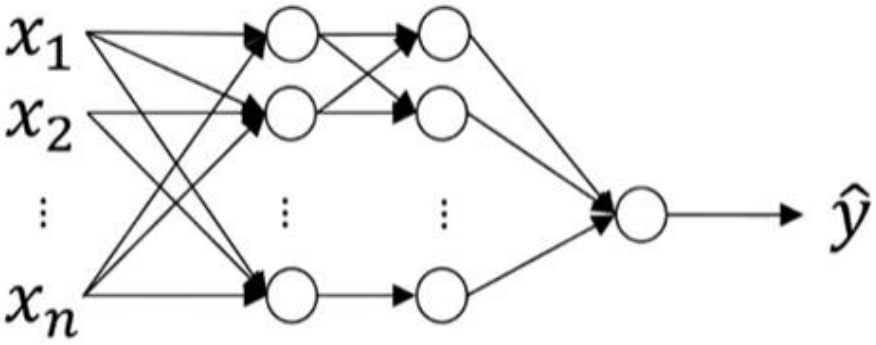
- Neural style Transfer



Computer Vision and Deep learning

- 64 x 64 x 3
- 입력 값 x의 크기 : 12288

Neural Network



- 1000 x 1000 x 3
- 입력 값 x의 크기 : 3백만

Convolution Operation

- 사람과 컴퓨터가 이미지를 받아들이고 해석하는 과정은 유사
- 컴퓨터는 Convolution 메커니즘으로 이러한 과정을 구현해 낸다

보다 High level feature를 인식해 나감

Detect Edge
(신경망 하위 층)

가능성 있는 물체 탐지

온전한 물체의 부분을 탐지
(신경망 상위 층)

Convolution Operation with Edge Detection

- 즉, 컴퓨터가 Input Image를 인식할 때 처음으로 할 일은 Vertical/ Horizon Edge를 찾는 것

Input Image

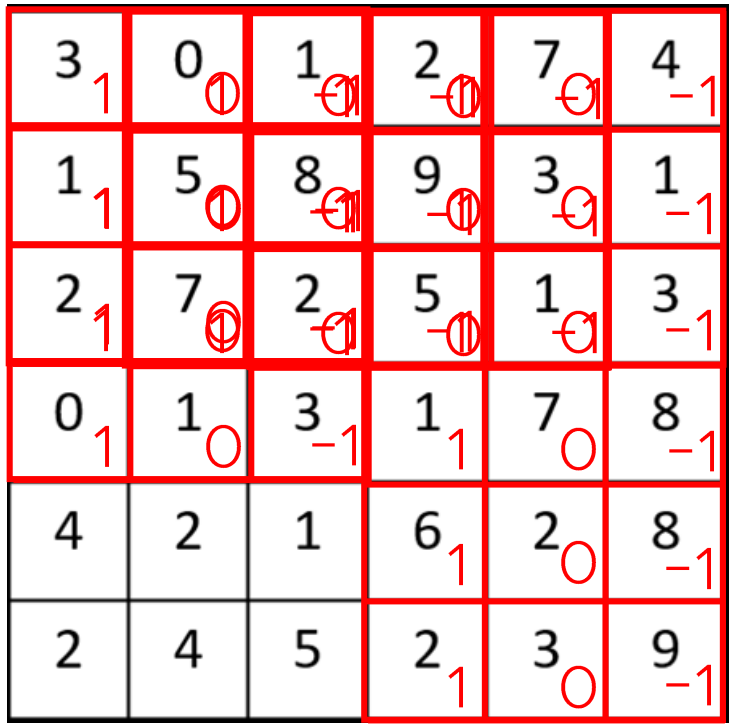
Vertical Edges

Horizontal Edges

Filter

- Filter(of Kernel)을 적용해 Convolution operation(합성곱 연산)을 수행함으로써 Edge를 찾아낸다

3x1 + 1x1 + 2x1 + 0x0 + 5x0 + 7x0 + 1x−1 + 8x−1 + 2x−1 =−5



6x6x1 Input Image

* convolution

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

3x3x1 Filter
for vertical edge

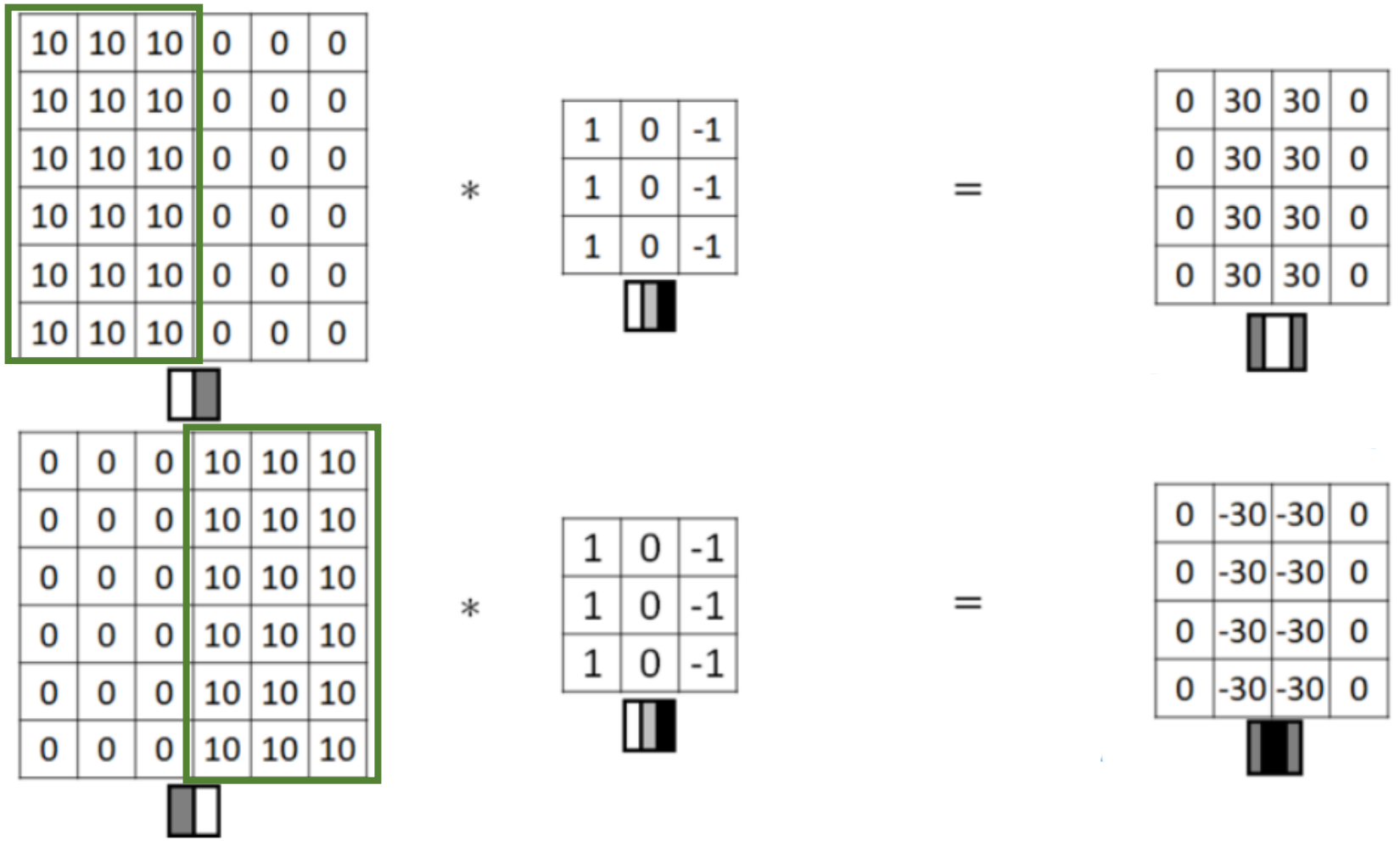
=

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

4x4x1 Output Image

Filter

- Vertical Edge



Filter

- Vertical and Horizontal Edge Detection

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

Vertical

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

Horizontal

10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
0	0	0	10	10	10
0	0	0	10	10	10
0	0	0	10	10	10

*

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

=

0	0	0	0
30	10	-10	-30
30	10	-10	-30
0	0	0	0

Learning to detect Edges

- Image가 주어졌을 때, 그 이미지에서 우리가 윤곽선을 검출하기 위해서 filter를 직접 설정해줄 필요는 없다
- CNN에서는 filter의 원소를 parameter로 설정한 뒤, **Backpropagation**을 이용해 스스로 학습하게해서 최적의 filter를 찾는다

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

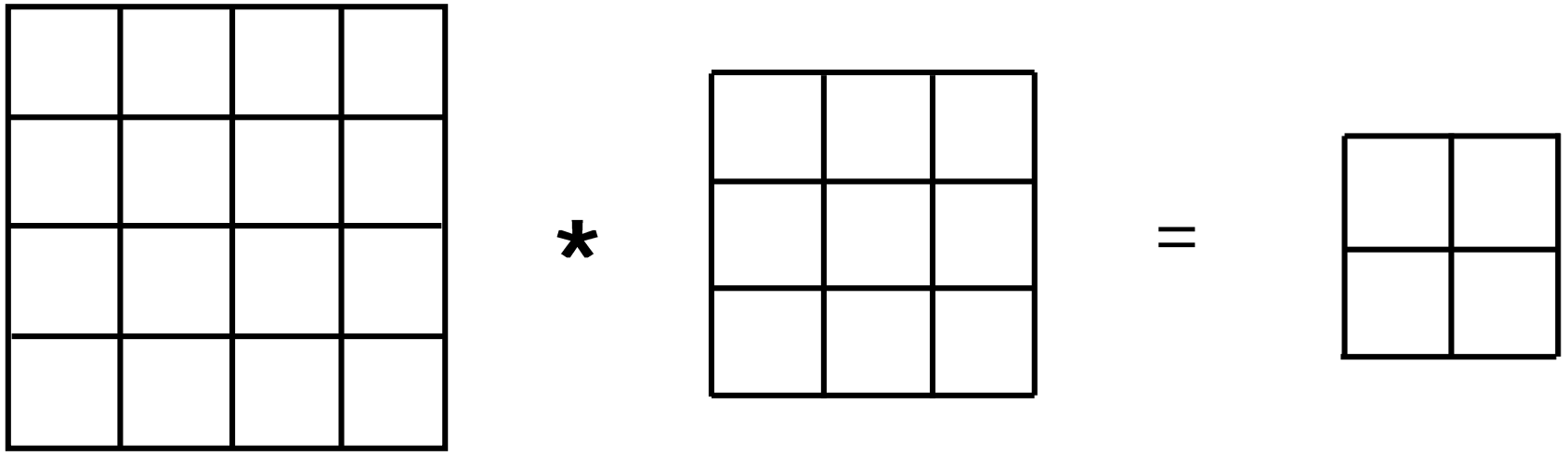
$*$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$=$?

Padding

- n x n image를 f x f filter로 convolution 연산을 할 때마다 image size가 축소
- 또한 가장자리에 위치한 pixel이 output image에 덜 사용되어서 정보가 손실된다는 문제가 발생

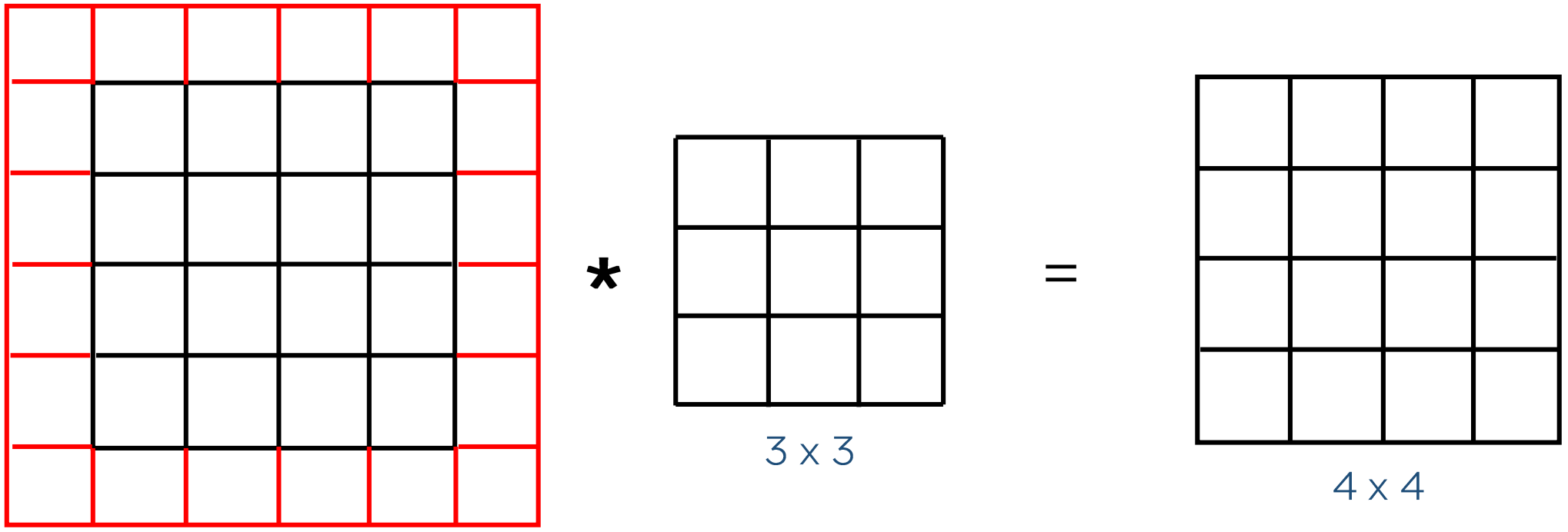


4 x 4 * 3 x 3 = 2 x 2

- Padding을 사용하지 않았을 때 output image의 크기 : $(n - f + 1) \times (n - f + 1)$

Padding

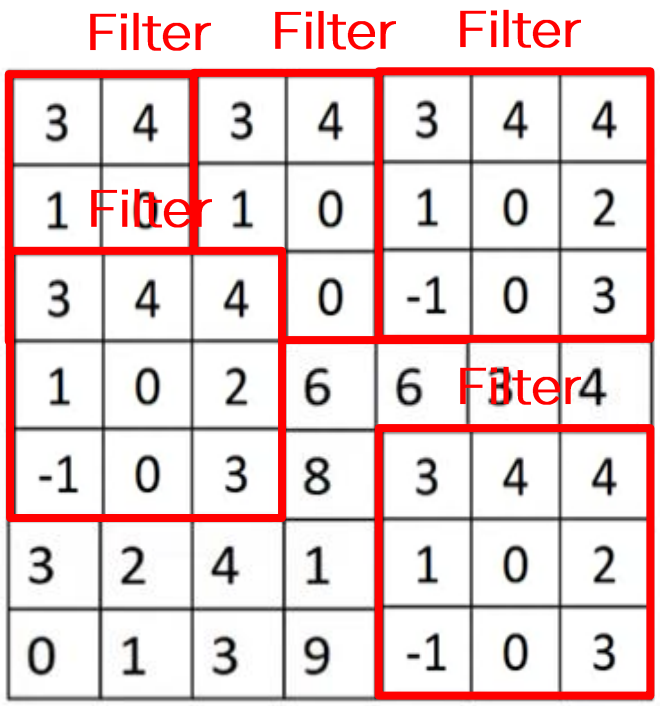
p = 1



- 합성곱 연산을 수행하기 전에 입력 데이터(이미지) 주변을 특정 값 (예컨대 0)으로 채우는 기법
- Padding을 사용했을 때 output image의 크기 : (n + 2p – f + 1) x (n + 2p – f + 1)
- output과 input의 크기가 같게 해주는 p = (f – 1)/2

Stride

S = 2



7x7x1 Input Image

*

3	4	4
1	0	2
-1	0	3

3x3x1 Filter

=

91	100	83
69	91	127
44	72	74

3x3x1 Output image

- Filter의 적용 위치 간격
- output image : $(n + 2p - f) / s + 1 \times (n + 2p - f) / s + 1$

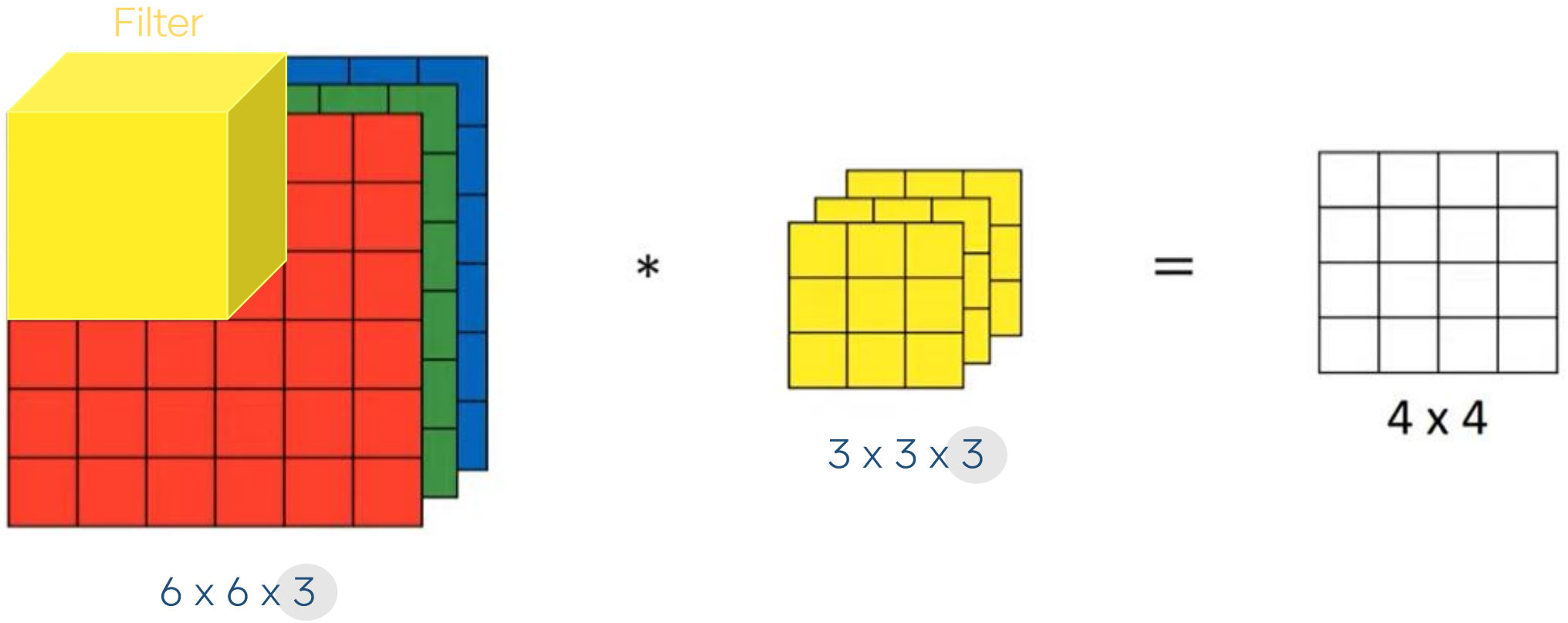
Summary of convolutions

n x n image	f x f filter
padding p	stride s

$$\text{output image} : \left\lfloor \frac{n + 2p - f}{s} \right\rfloor + 1 \times \left\lfloor \frac{n + 2p - f}{s} \right\rfloor + 1$$

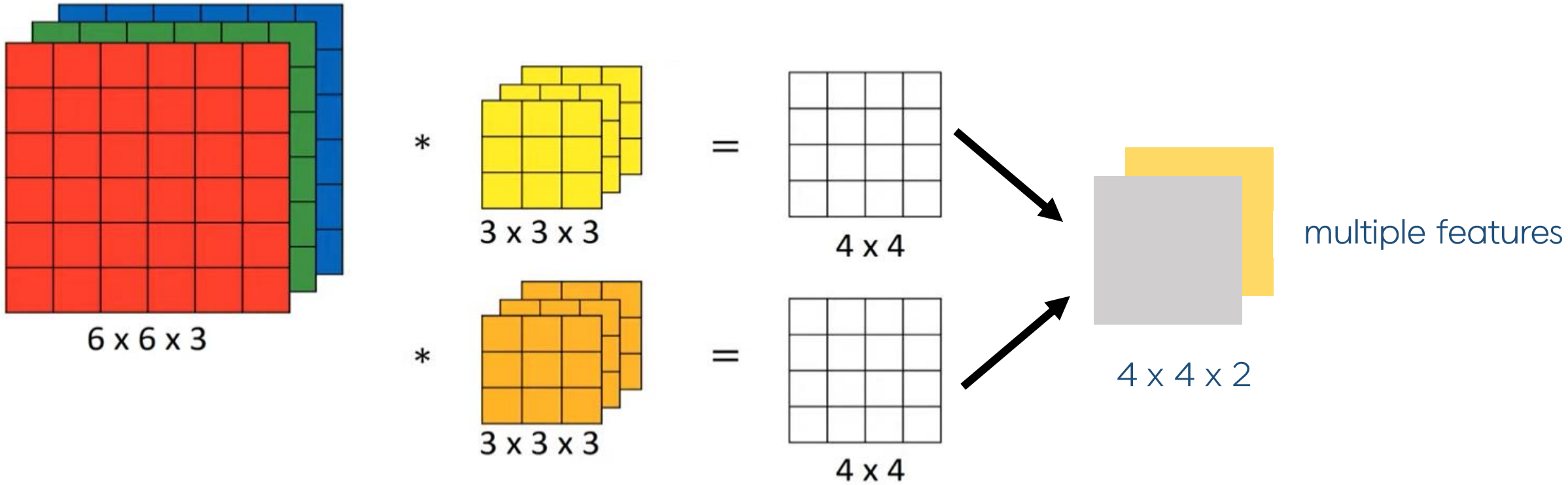
Convolutions over volumes

- convolutions on RGB images



Convolutions over volumes

- Multiple Features



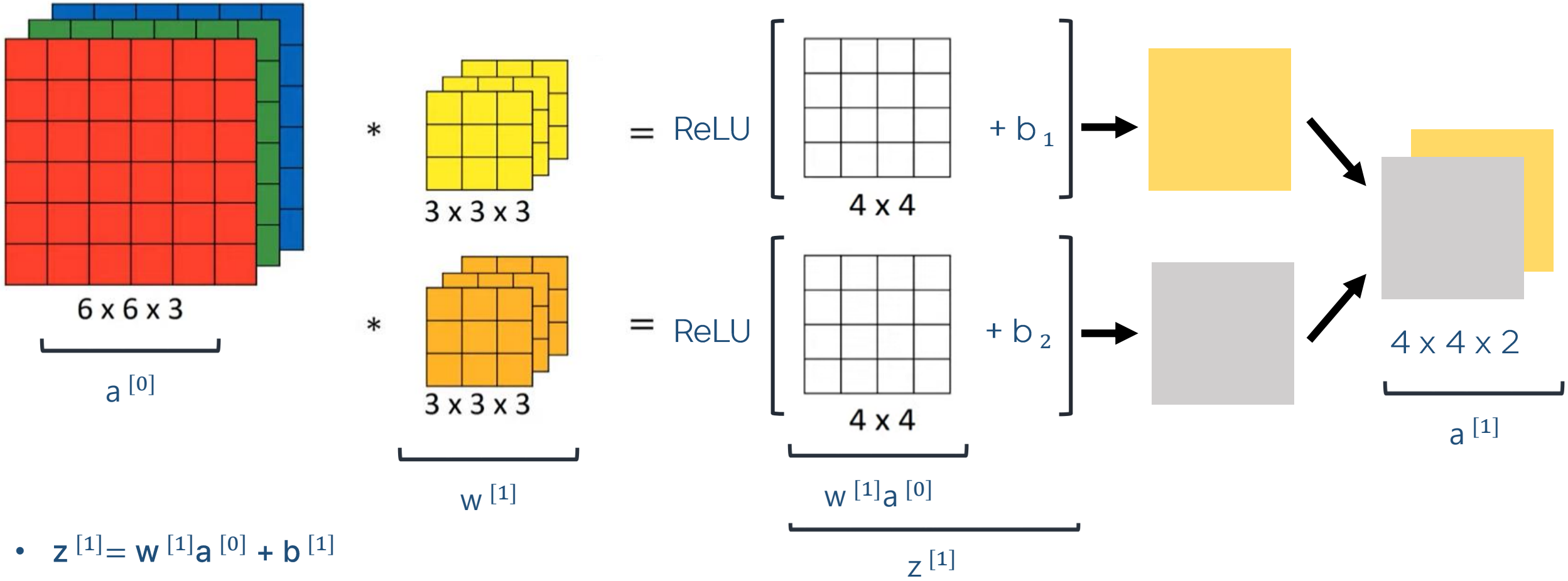
If, stride = 1 & No padding

output image : $n - f + 1 \times n - f + 1 \times n'_c$

사용한 filter의 개수

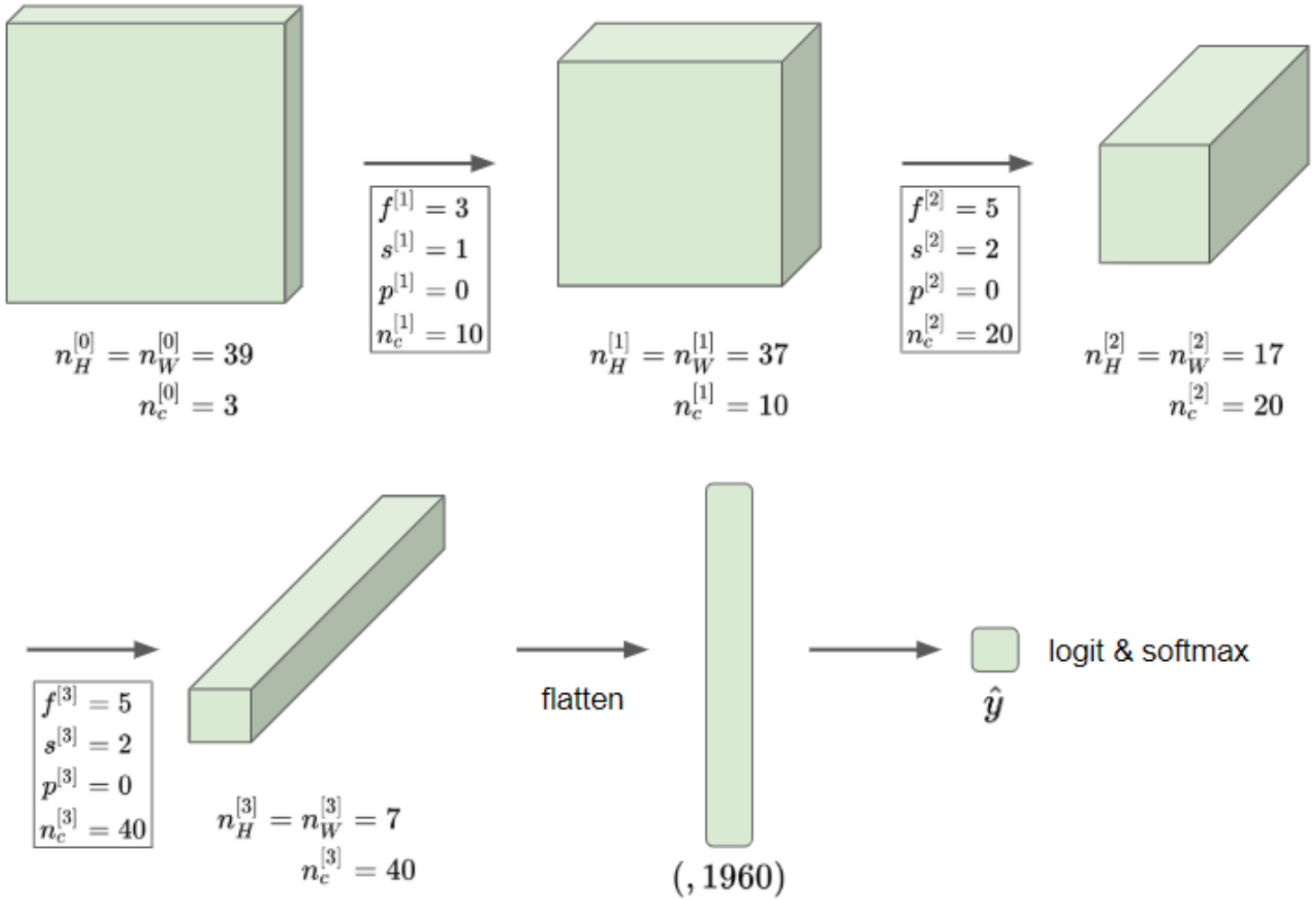
One layer of a convolutional Network

- Example of a layer

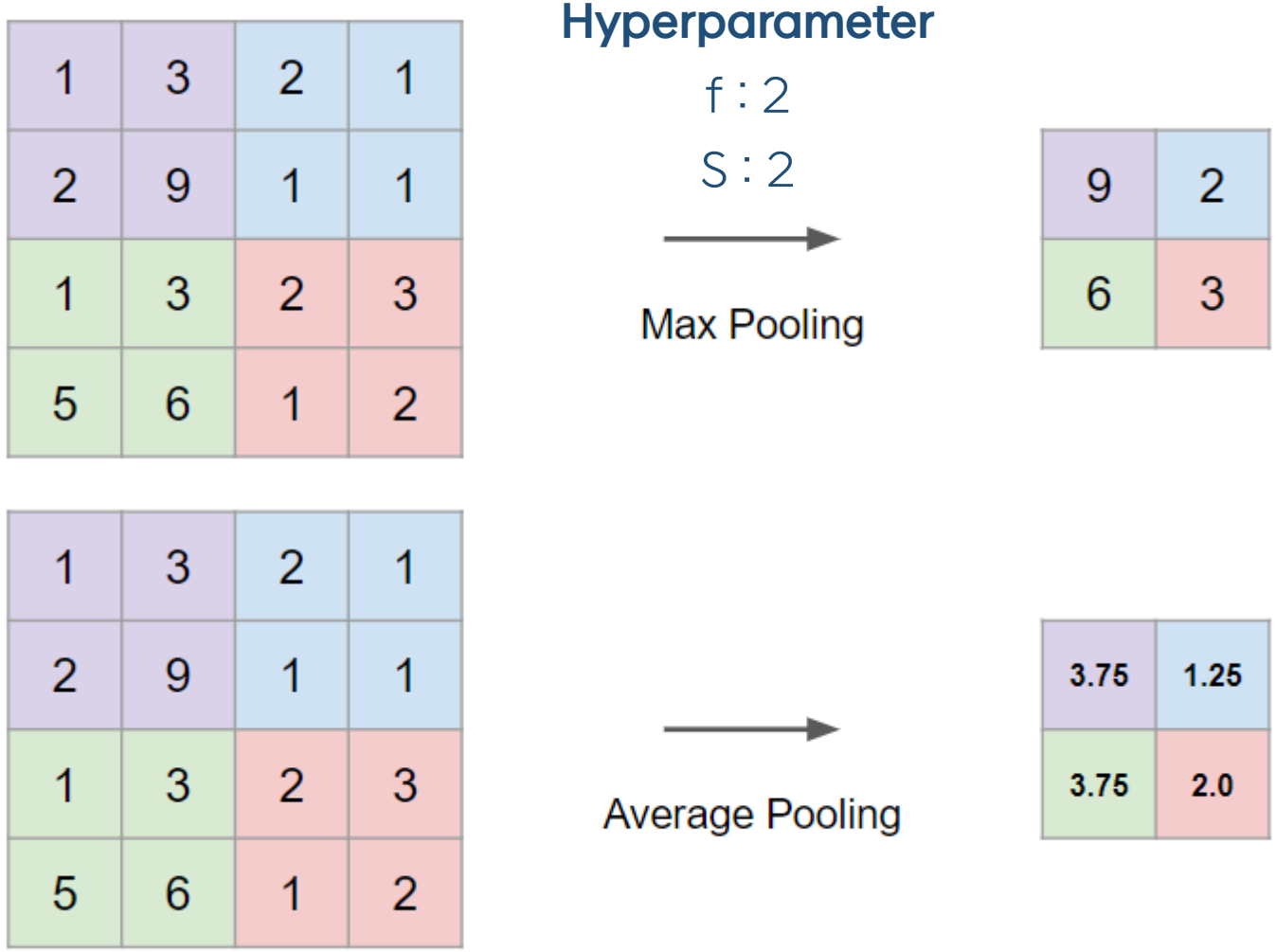


- $z^{[1]} = w^{[1]} a^{[0]} + b^{[1]}$
- $a^{[1]} = g(z^{[1]})$

CNN example

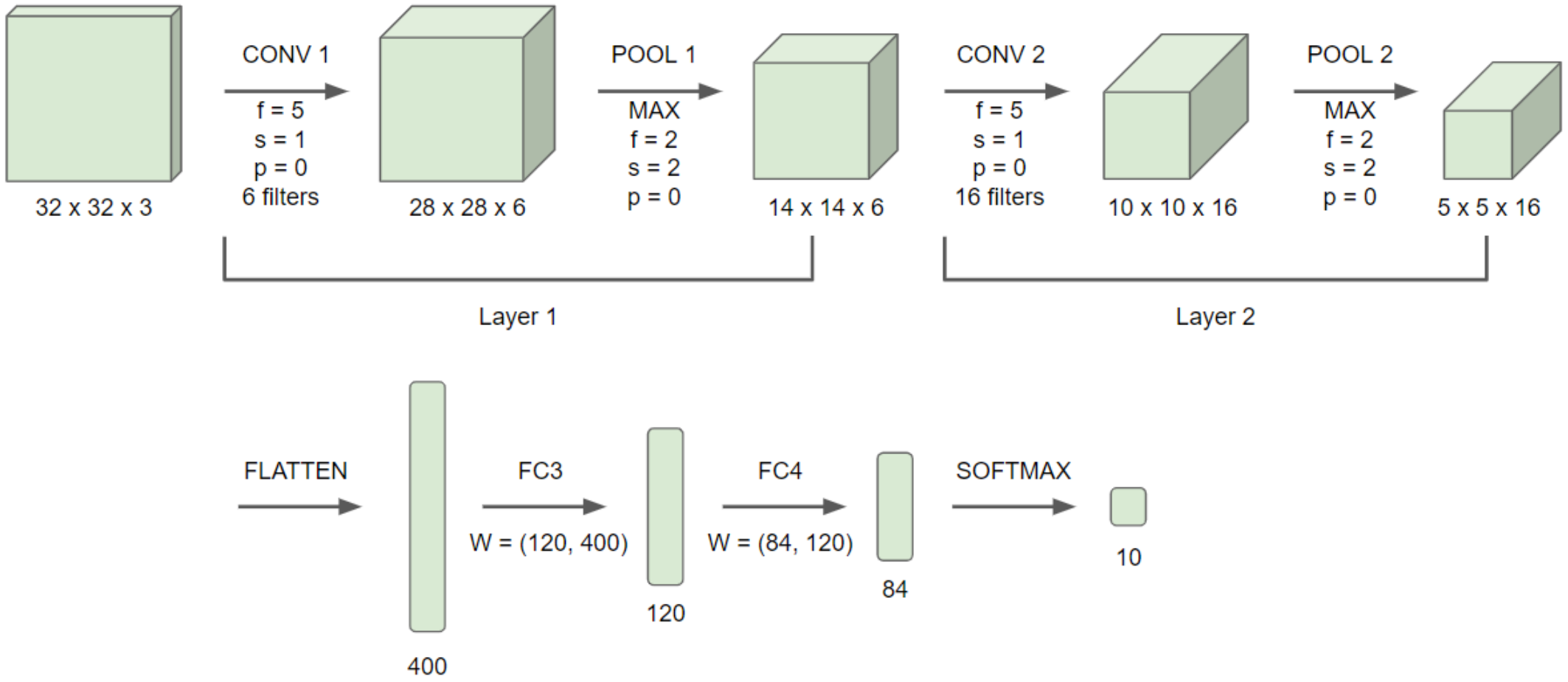

$n_H^{[0]} = n_W^{[0]} = 39$
$n_c^{[0]} = 3$
$f^{[1]} = 3$
$s^{[1]} = 1$
$p^{[1]} = 0$
$n_c^{[1]} = 10$
$n_H^{[1]} = n_W^{[1]} = 37$
$n_c^{[1]} = 10$
$f^{[2]} = 5$
$s^{[2]} = 2$
$p^{[2]} = 0$
$n_c^{[2]} = 20$
$n_H^{[2]} = n_W^{[2]} = 17$
$n_c^{[2]} = 20$
$f^{[3]} = 5$
$s^{[3]} = 2$
$p^{[3]} = 0$
$n_c^{[3]} = 40$
$n_H^{[3]} = n_W^{[3]} = 7$
$n_c^{[3]} = 40$
flatten
$(, 1960)$
$\hat{y}$
logit & softmax

Pooling



- Pooling은 합성곱 계층과 달리 대상 영역에서 최댓값이나 평균을 취하는 명확한 처리이므로 학습해야 할 매개변수가 없음
- Pooling을 사용해도 채널 수가 변하지 않음
- 입력데이터가 조금 변해도 Pooling의 결과는 잘 변하지 않음
- 주로 f=2, s=2 가 자주 사용되고 이러한 Fooling은 높이와 너비를 절반 정도만큼 줄어들게 하는 효과가 있음

CNN


32 x 32 x 3
CONV 1
f = 5
s = 1
p = 0
6 filters
28 x 28 x 6
POOL 1
MAX
f = 2
s = 2
p = 0
14 x 14 x 6
Layer 1
CONV 2
f = 5
s = 1
p = 0
16 filters
10 x 10 x 16
POOL 2
MAX
f = 2
s = 2
p = 0
5 x 5 x 16
Layer 2
FLATTEN
400
FC3
W = (120, 400)
120
FC4
W = (84, 120)
84
SOFTMAX
10

대표적인 CNN

- LeNet

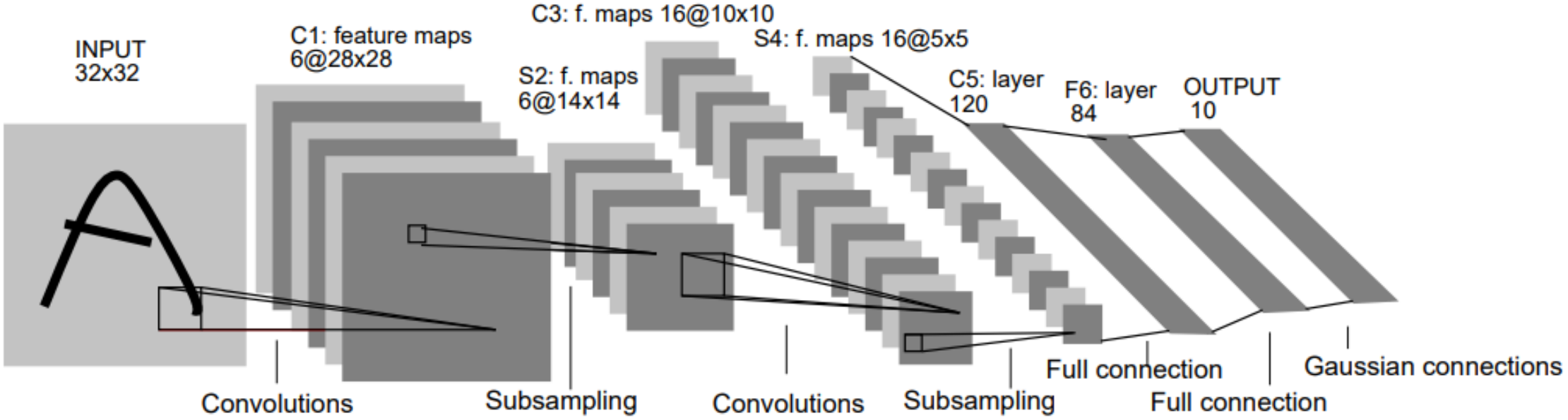
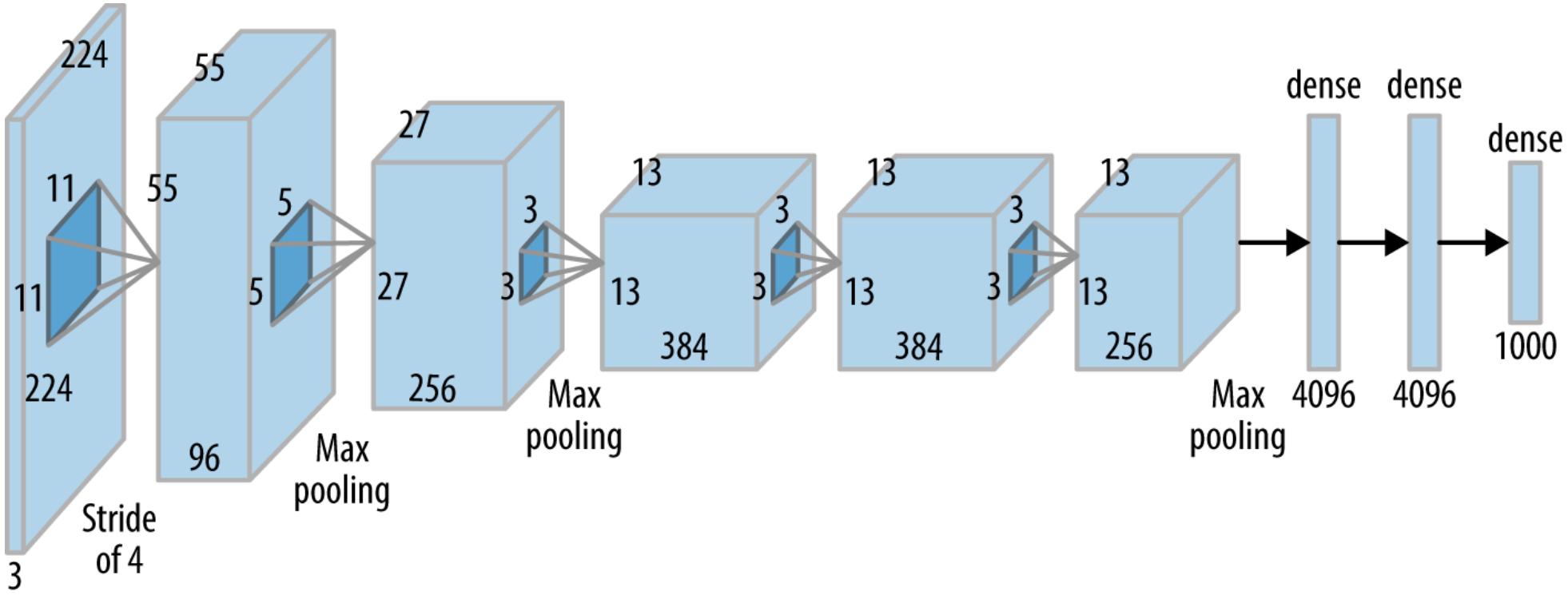


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

- LeNet은 손글씨 숫자를 인식하는 네트워크로, 1998년에 제안된 CNN의 원조
- 활성화 함수로 시그모이드 함수를 사용, 서브 샘플링을 하여 중간 데이터의 크기를 줄임

대표적인 CNN

- AlexNet



- 2012년에 발표된 AlexNet은 딥러닝 열풍을 일으키는데 큰 역할
- 활성화함수로 ReLU를 이용, 드롭아웃을 사용, LRN이라는 국소적 정규화를 실시하는 계층을 이용

감사합니다