

# 인공지능 기초학습 2

2017010698  
수학과 오서영

## **CNN (Convolutional Neural Network)**

: 모델이 직접 이미지, 비디오, 텍스트  
또는 사운드를 분류하는  
딥러닝에 가장 많이 사용되는 알고리즘

## **CNN이 Fully connected Neural Network와 비교하여 갖는 차별성**

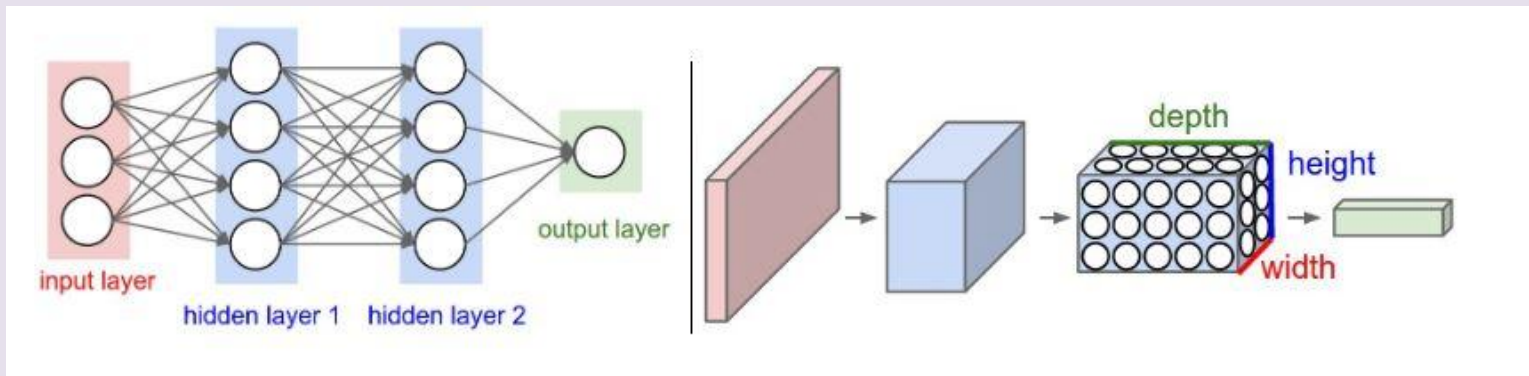
1. 각 레이어의 입출력 형상 유지
2. 이미지의 공간 정보를 유지하면서 인접 이미지와의 특징을 효과적으로 인식
3. 복수의 필터로 이미지의 특징 추출 및 학습
4. 추출한 이미지의 특징을 모으고 강화하는 Pooling 레이어
5. 필터를 공유 파라미터로 사용하기 때문에, 일반 인공 신경망과 비교하여 학습 파라미터가 매우 적음

# FC layer의 문제점

**FC : 1차원 데이터만 입력 받을 수 있기 때문에,  
3차원 데이터를 평탄화 해서 입력해야 한다.  
여기서 3차원 데이터의 공간적 정보가 소실된다는 문제가 발생한다**

MNIST 이미지는 형상이 (1채널, 가로 28픽셀, 세로 28픽셀)인 3차원  
이 3차원 데이터에는 공간적으로 가까운 픽셀은 값이 비슷하다거나,  
RGB의 각 채널은 서로 밀접하게 관련되어 있다든가 하는  
공간적 정보가 들어있다.

이를 Affine layer에 입력할 때, (1, 784)의 1차원 데이터로  
평탄화 해서 넘기기 때문에 이런 공간적 정보가 소실된다.  
**반면 CONV layer는 형상을 유지한다. 입/출력 모두 3차원  
데이터로 처리하기 때문에 공간적 정보를 유지할 수 있다.**



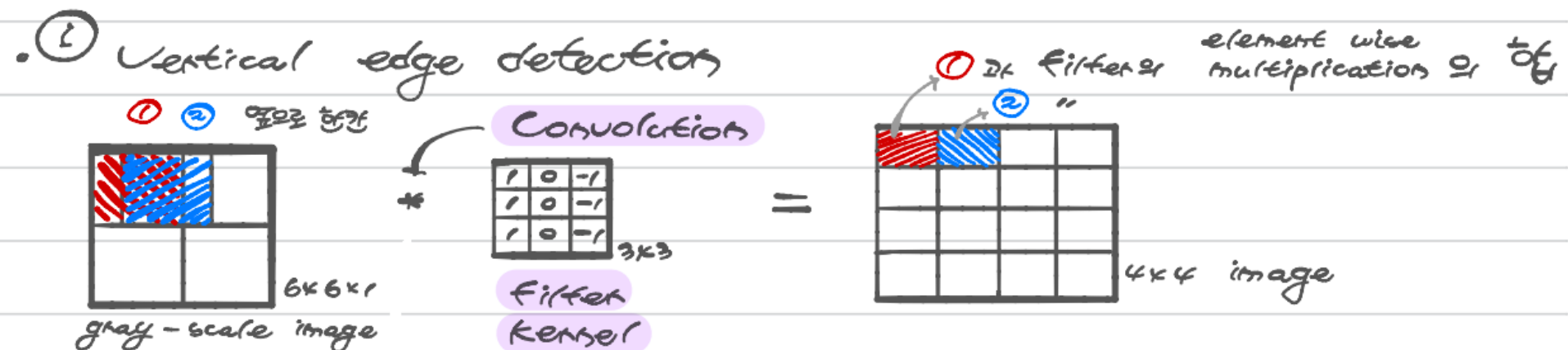
# Convolution (합성곱)

## 2. Edge detection example

• Computer Vision problem

— first, detect edges in image

→ how? ) ① detect vertical edges  
② detect horizontal edges



⇒ why use?

모두 10	모두 0
----------	---------

\*

밝음 → 어둡음과 차이를  
만드는 역할을  $\overline{7}$

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

=

0		0
0	4	0
0	30	0
0		0



보여줄 수 있는 vertical edge

detect

output

: looks like strong vertical edge

필터 : 이미지의 특징을 찾아내기 위한 공용 파라미터

### 3. More edge detection

- Vertical and horizontal detection

1) vertical

/	0	-/
/	0	-/
/	0	-/

2) horizontal

/	/	/
0	0	0
-/	-/	-/

— ex)

①

10	0
0	0

②

\*

/	/	/
0	0	0
-/	-/	-/

=

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



( ① : strong positive edge

② : strong negative edge



# Channel (채널)

RED Channel



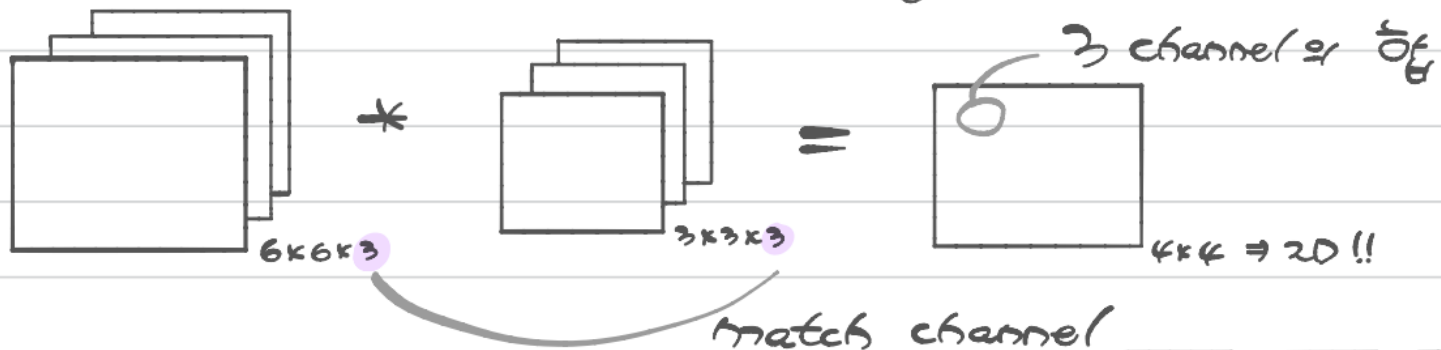
Green Channel



Blue Channel



• convolutions on RGB images






— 객체 추출을 위해서만 detect 하고 싶다 :

$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$
R	G	B

**Stride** : 지정된 간격으로 필터를 순회하는 간격

## 5. Strided Convolutions

•  \*  = 

if stride = 2, (\*)가 두 칸씩 이동

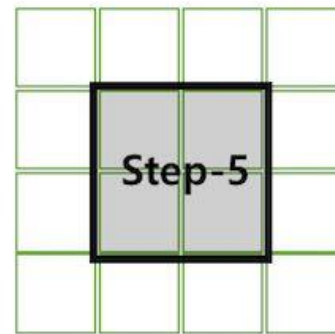
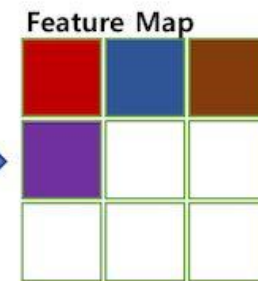
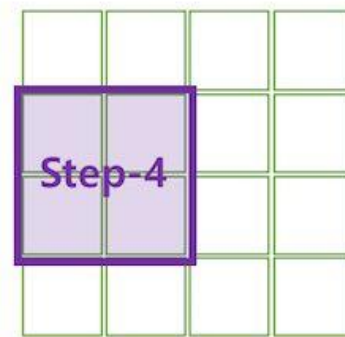
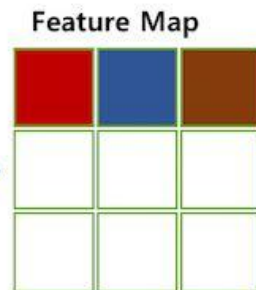
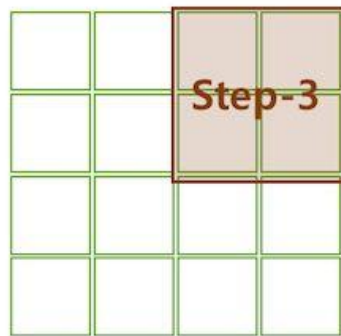
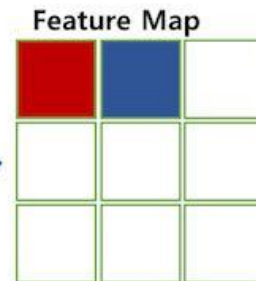
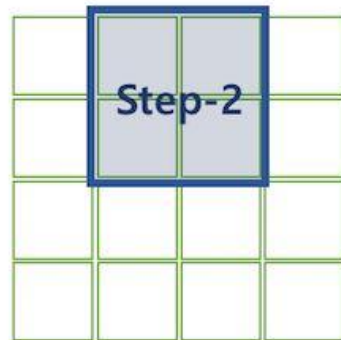
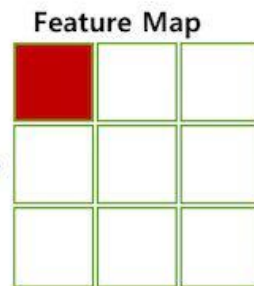
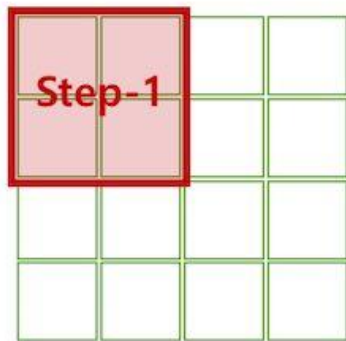
i.e.  $n \times n * f \times f = \frac{n+2p-f}{s} + 1 \times \frac{n+2p-f}{s} + 1$

padding  $p$ , stride  $s$

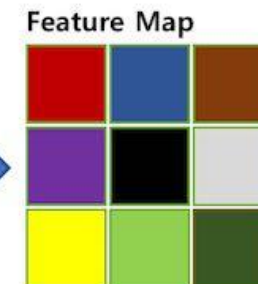
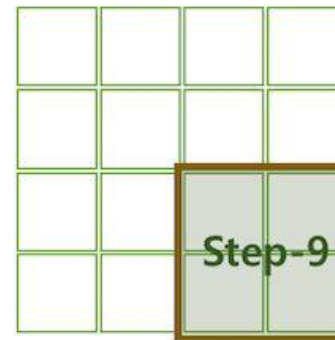
⊕ 결과가 아닐 경우  $\text{floor}(\lfloor \cdot \rfloor)$  사용

i.e.  바깥에 잘라  
제거





.....

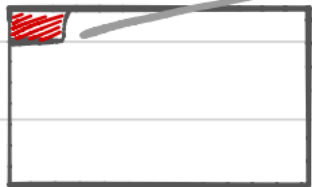


# Padding

패딩 : 입력 데이터의 외각에 지정된 픽셀만큼 특정 값을 채워 넣는 것

## 4. Padding

모니터는 여기서 영향



$n \times n$



$f \times f$



$(n-f+1)(n-f+1)$

— or 원본의 단점 ① your image shrink

② pixels on corners are used much less in output  
가장자리 정보를 많이 버리게 됨

## ⇒ padding

more influence



\*



$f \times f$



더 큼  
 $(n+2p-f+1)(n+2p-f+1)$

가장자리와  
영향 ↓

$p = * \text{padding} = 1$

# Pooling

풀링 : 데이터의 크기를 줄이거나 특정 데이터를 강조하는 용도

## 9. Pooling Layers

• Pooling layers : "Max pooling"

1	3	2	1
2	9	1	1
1	3	2	3
5	6	1	2

 $\xrightarrow{\text{max}}$ 

9	2
6	3

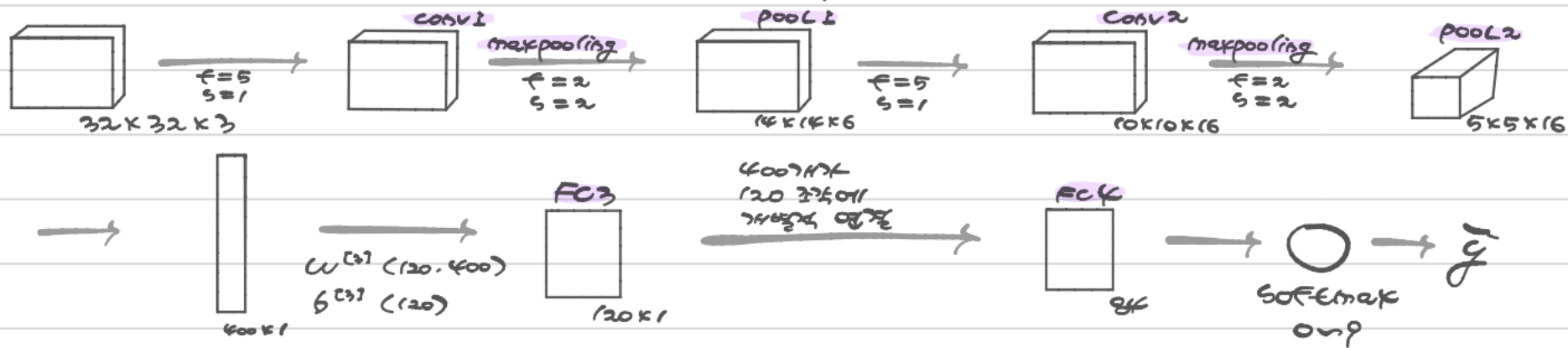
$f = 2$   
 $s = 2$

hyperparameter

so no GD!  
 no learning!

데이터 크기를 줄이기 위해 (memory ↓, overfitting)  
 특징의 값이 큰 값이 다른 특징들을 대표한다는 개념

• NN ex (MNIST)  $\Rightarrow 0 \sim 9$



# Reference

- [1] 합성곱 신경망 (CNN, Convolution Neural Network),  
<https://machine-geon.tistory.com/46>

