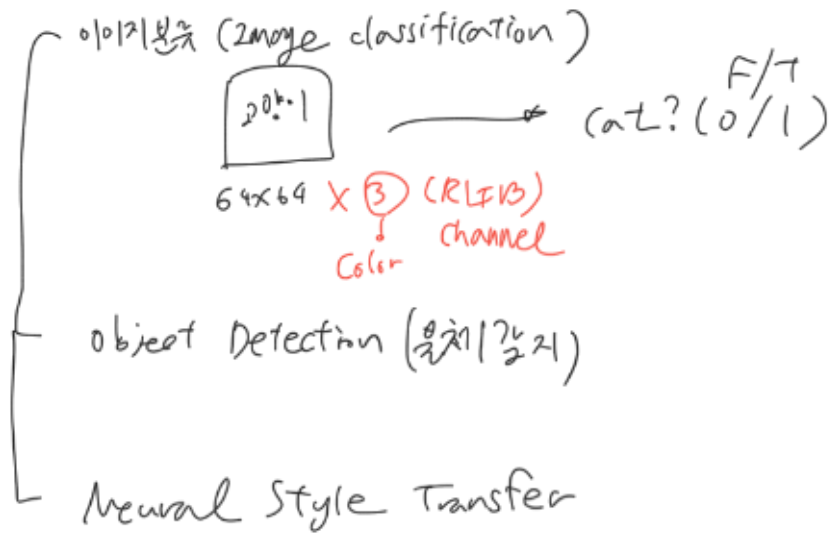
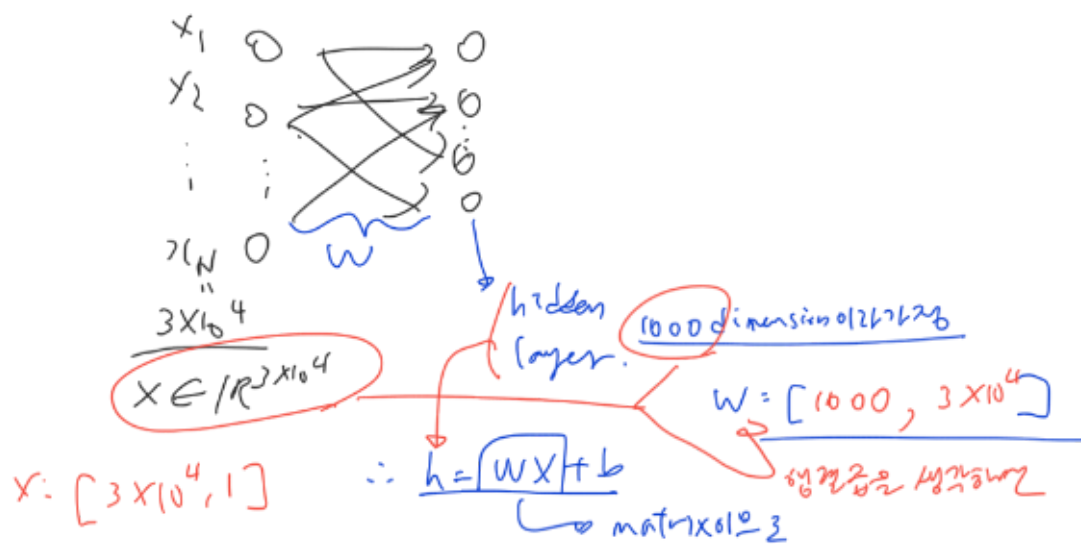


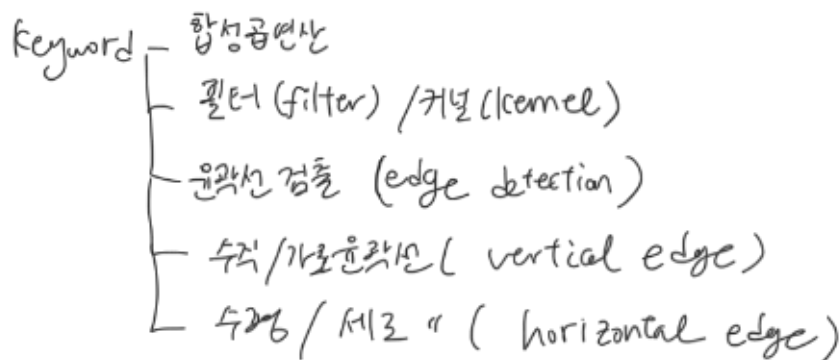
딥러닝 ① 컴퓨터비전



ex) Image가 100x100 이면 RGB 채널 3
총 100x100x3 = 30,000 차원



② 모서리 감지 \rightarrow 합성곱 신경망과 모서리 감지 필터를 배운다.

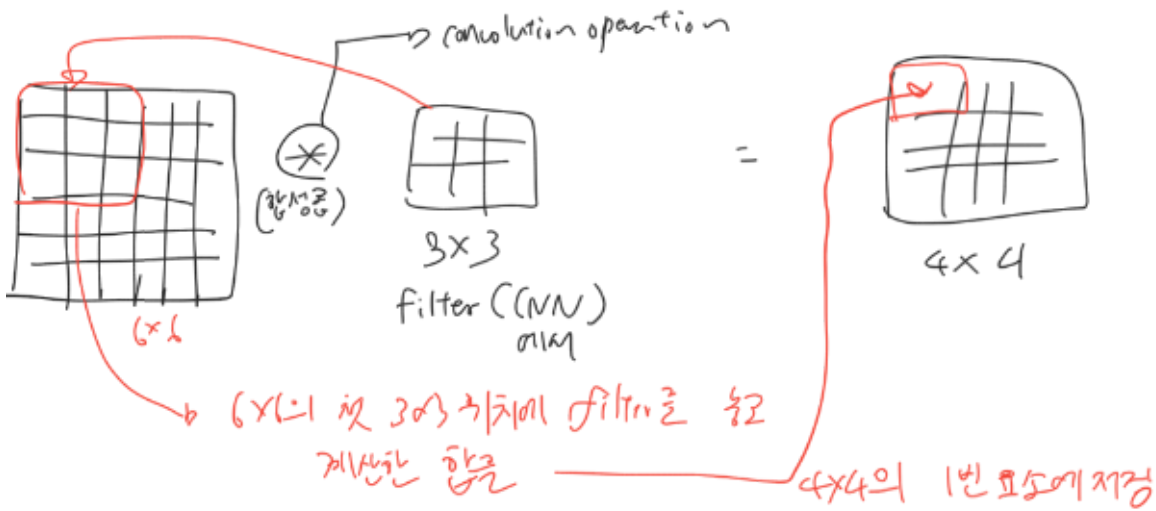


① 이미지 보정 \rightarrow 수직 / 수평 / 세로 / 회전 (image rectification)

수평모리찾기 (가로선만 찾아냄.)

* Vertical edge detection

gray scale channel 이미.



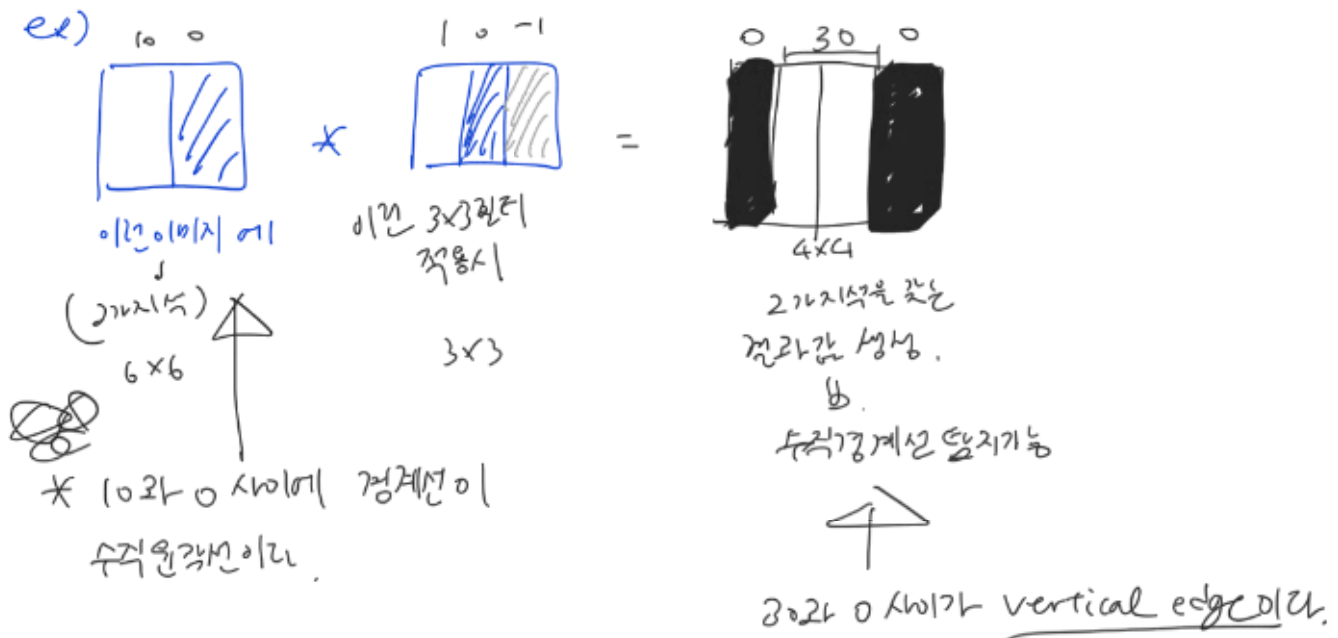
결과값은 2byte 크기 $N \times M$ 이고

filter = $f \times f$ 2byte

$(n-f+1) \times (m-f+1)$ matrix이다.

* Convforward라함

tf.nn.conv2d 3 합성곱 계산 가능..



* 3강 : 더 많은 모서리감지

양(+), 음(-) 원곡선 차이 = 밝기 차이

어두운 dark

filter \rightarrow

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

\downarrow
세로 (vertical) 방향의
filter



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

\downarrow
가로 (horizontal) 방향의
filter



1	0	1
0	1	0



이것이 (가)
 \downarrow
filter가 이 위치에 있게 되면
어두운 값이
이런 색이 나올

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

Sobel filter

3	0	-3
10	0	-10
3	0	-3

Scharr filter

세로 방향의 필터



Image



back propagation

convolution operation

\star filter를 특정한 변수로 두기

w_1	w_2	w_3
\vdots	\vdots	\vdots
\vdots	\vdots	w_9

w_i 로 두기 학습시켜서 다양한
필터를 만들어 내기.

\star filter를 변수로 두고 여러개의 filter를 통해 계속 학습시켜 최적의 filter를 만들어 내고
이를 통해 여러 Image Classification을 가능케 한다.

\Rightarrow 컴퓨터 비전에서 매우 중요.

\star

\rightarrow 최근 들어 많은 filter 조합 학습시켜 Image Classification에 사용할 최적 filter를
사용

\star 4강 패딩 (Padding)

\Rightarrow 합성곱 (convolution operation) 신경망을 변경하는 방식 중 하나.

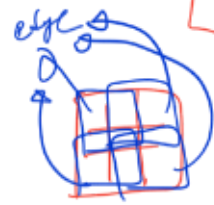
결과

계산식

$n \times n$ (Image)를 $\times f \times f$ (filter)로 합성곱 \Rightarrow

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

단점)



Img가 축소됨 $6 \times 6 \rightarrow 4 \times 4$... 반복시 \rightarrow 2mg 축소

\rightarrow 가장자리 픽셀이 경라 이미지 계산에 각각 1번 밖에 사용이 안됨,

반면 중간 영역값 픽셀은 여러번 필터에 걸림 (치이발성)

이로 인해 가장자리 근처 정보들이

喪失됨으로 덜 쓰임.

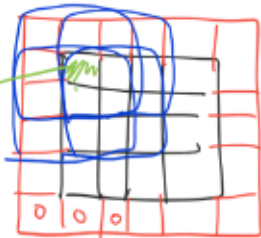
Why?) 이미지 축소가 deep 신경망에서 치명적인가?

만약 수백층을 거쳐 나온다면 결국 아주 작은 2mg만 남게된다.

+

2mg 가장자리 정보가 손실

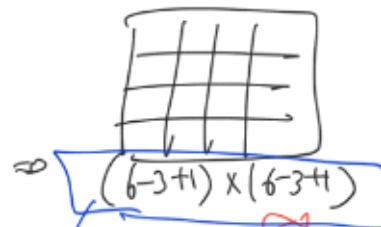
해결책 \Rightarrow 이미지를 pad로 덧대기 (테두리에 1픽셀만큼 0으로 채우기)
padding



$4 \times 4 \rightarrow 6 \times 6$ 이미지

\otimes
합성곱

3×3 filter



결과 = $6 \times 6 \Rightarrow$ 2mg 크기 유지 가능

padding으로 채우는 값은 0으로 채우기 \Rightarrow Img에 영향없음.

+ p 는 패딩의 양 ex) $p=1$ (1픽셀만큼 추가로 가장자리에 더했다는 의미)

edge가 4번
필터에 걸림
 \downarrow
영향 없음

$$(n+2p-f+1) \times (n+2p-f+1)$$

$$= (n+2p-f+1)$$

it) $p=2$

$$(n+2p-f+1) \times (n+2p-f+1) = 6 \times 6$$

$4 + 4 - 3 + 1$
 $\quad \quad \quad \downarrow$
 $\quad \quad \quad 6$

* "Valid": $(n \times n) \times (f \times f) \rightarrow (n-f+1) \times (n-f+1)$
 no-padding valid convolution (유효 합성곱) $p=0$ dimen Img

"Same": output size is the same as the input size.
 we-padding $(n+2p-f+1) \times (n+2p-f+1) = n \times n$ we-p $n \times n \rightarrow \text{filter size } n \times n$

$$\therefore n+2p-f+1 = n$$

$$\therefore p = \frac{f-1}{2}$$

따라서 패딩 양 p 을 filter 크기를
 보고 결정 가능하다.

* 일반적으로 컴퓨터 비전에서

filter는 홀수이다

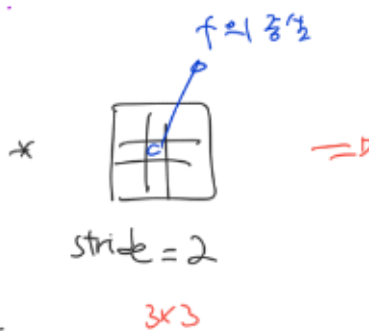
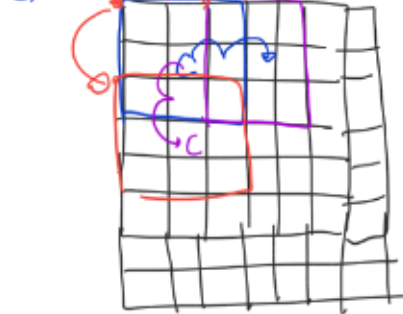
왜냐하면 짝수도 쓸 수 있다.

why? f 가 even이면 padding이 비례정비된다.
 filter에 중심 픽셀을 잡을 수가 없다.

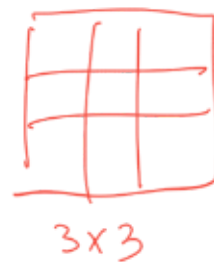
3x3 이 very common, also usually use 5x5, 7x7 and so on

* Stride \rightarrow filter를 이동시키는 칸수.

ex) 2칸씩 f 이동.



$p=0$ 이 된다. (valid)로



$n \times n$ $f \times f$
padding p stride s

$$\Rightarrow \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

$$\frac{7+2 \times 0 - 3}{2} + 1 = \boxed{2+1=3}$$

한번 $\frac{n+2p-f}{s}$ 가
정수가 아니거든?
 \downarrow
내림 해준다.

ex) 만약 $\text{Img} = 28 \times 28$ (mnist이고) filter = 3x3,

$p=1, s=2$ 라면 결과값은?

$$\Rightarrow \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor = \boxed{14 \times 14}$$

$$= \frac{28+2-3}{2} = \lfloor 14.5 \rfloor = 14$$

이후, $n \times n$ (image), $f \times f$ (filter), padding p , stride s 이라면 \Rightarrow output image은

$$= \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

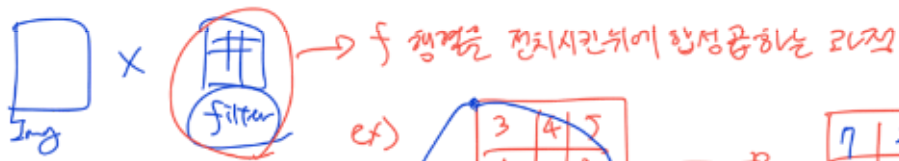
미러링 X
(요차상관)

미러링 O
(합성곱)

* Cross-correlation vs convolution

* mirroring (미러링과정)을 함께 해봐 실제 합성곱 연산이 가능해진다.

미러링이란



ex)

3	4	5
1	0	2
1	1	1

mirroring

7	2	5
9	0	4
-1	1	3

결론) 미러링 안하면 어떻게 할?

여기 끝은

위점이 없는 것 같은 느낌

* 딥러닝에서는 cross-correlation로 합성곱이라 하기로 한다. (관습적)

→ 미러링 안해도 신경망이 영향 없고 고르 단는 문제