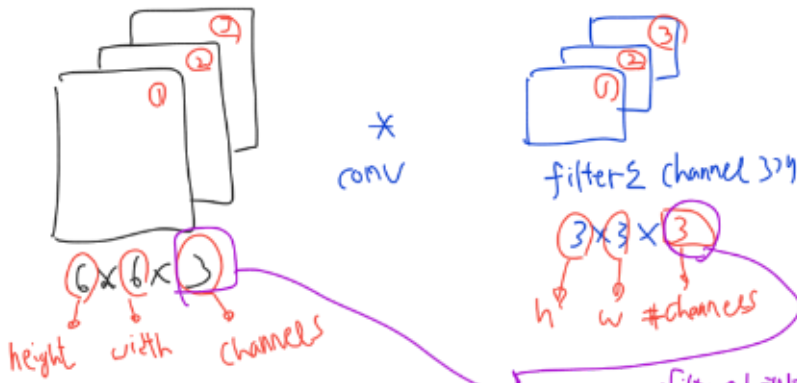


\* CNN (입체형 이미지에서의 합성곱)

↳ 합성곱 (convolution), 채널 (channel), 필터 (filter)

3D convolution RGB images → ex)  $6 \times 6 \times 3$  image

↳ R, G, B = color 채널



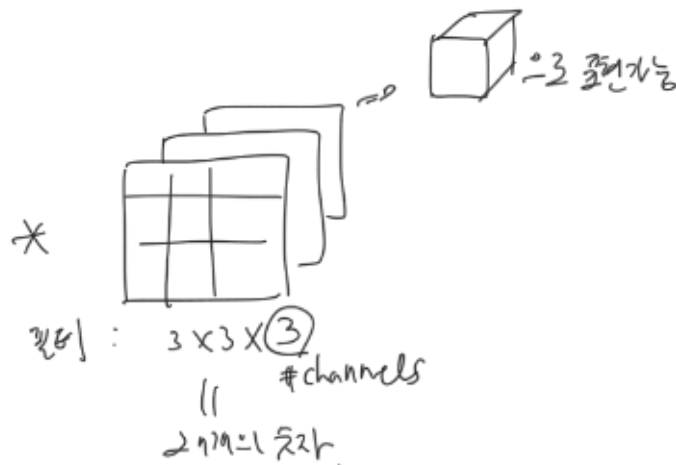
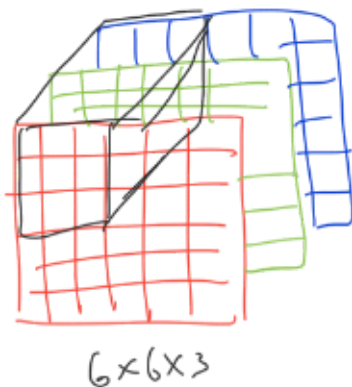
image와 filter의 채널을 맞게 맞추어야 함.

$$= \frac{4 \times 4 \times 1}{s=1}$$

채널  $s=1$

$$\text{conv 결과} = \left\lfloor \frac{h+f-s}{s} + 1 \right\rfloor$$

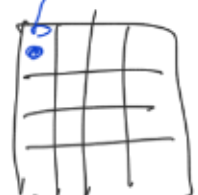
$$\frac{6+3-3}{1} = 3+1=4$$



합성곱 계산방법 : 필터 f:  $3 \times 3 \times 3 = 27$ 개 숫자를 갖고, 이를 R, G, B 각각의 채널의 9개의 숫자마다 곱함.

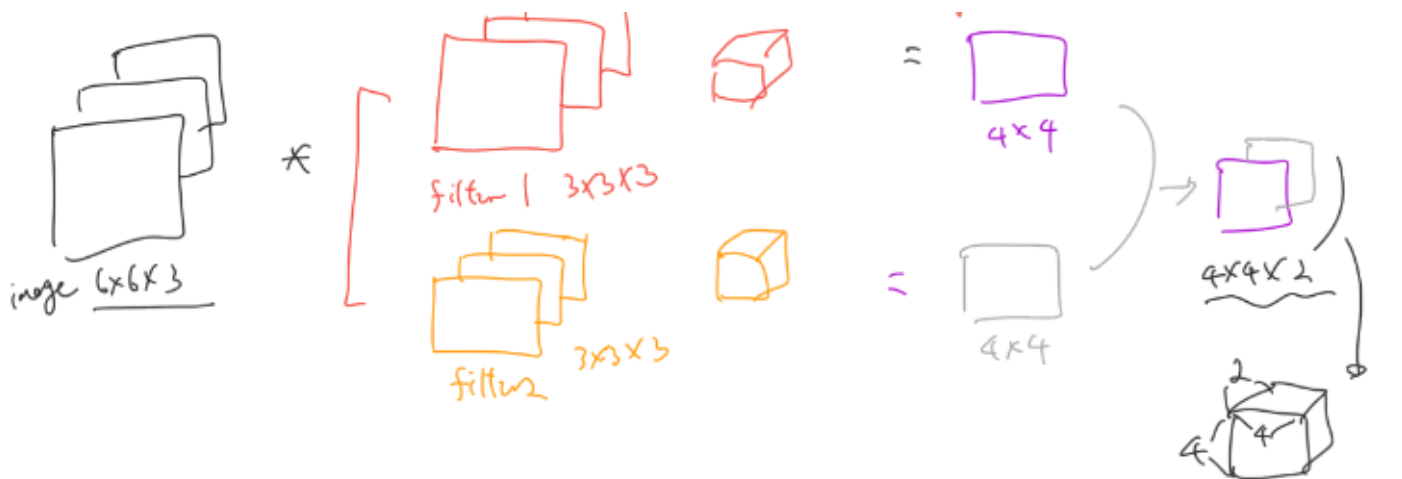
$$R \times f_{\#1} + G \times f_{\#2} + B \times f_{\#3} = \text{한 칸 계산값}$$

$3D \times 3D \rightarrow 2D$ 가 나옴. (R, G, B channel)이 2개의 칸칸에 합쳐져서 나오함.



\* 수직, 수평 움직임을 동시에 안고 싶다면?





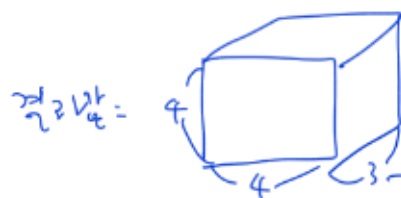
☆ 6/5 정리

image:  $n \times n \times n_c$  (channel)

filter:  $f_x \times f_y \times n_c$

$$) * = (n-f+1) \times (n-f+1) \times n'_c$$

☆)  $n'_c = 3, n = 6, f = 3$



$n'_c =$  필터의 갯수

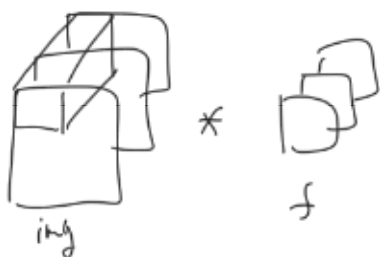
서로 다른 필터 3개를  
img에 합성하면 화소  
합친 결과이다.

+ stride와 padding이 존재시

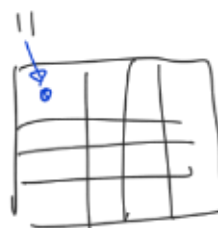
$$\Rightarrow \text{결과값} \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times n'_c$$

☆ 사용한 필터의 수  
+ channel 수를 의미

☆ 채널 = 색상 또는 영상 이미지의 깊이



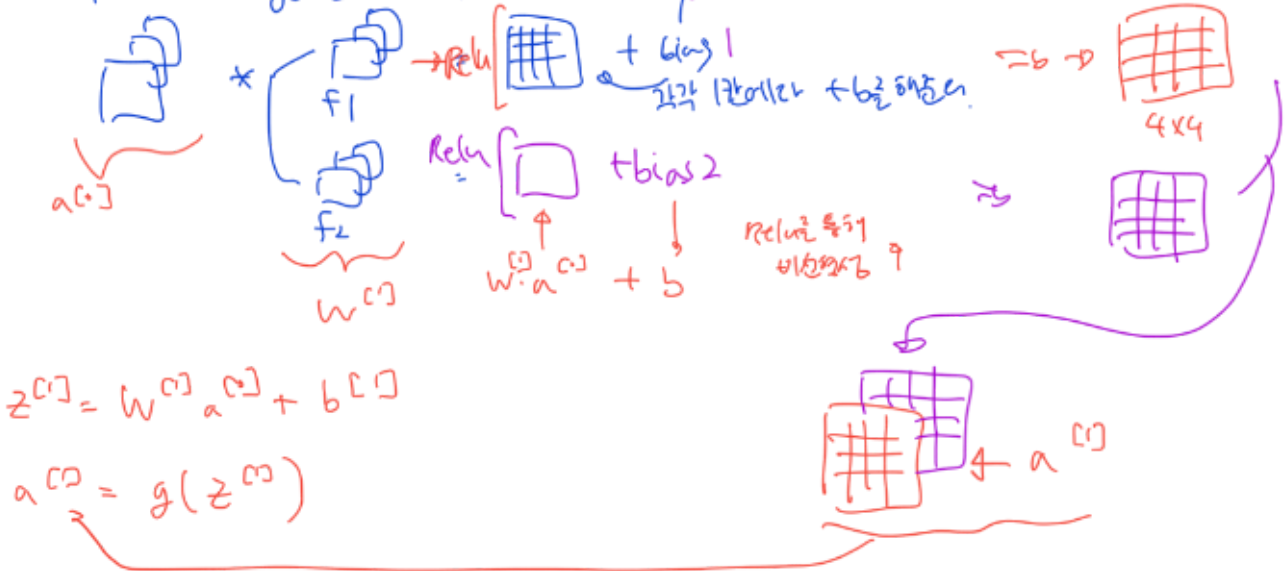
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} \times \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \times \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \times \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$



\* 합성곱 네트워크의 각 계층 수행하기.

- 합성곱 (convolution)
- 편향 (bias)
- 활성화 함수 (activation)

CNN의 한 계층 (layer)을 한눈에 보기 위해서는  $b$ 를 다 이해해야 한다.



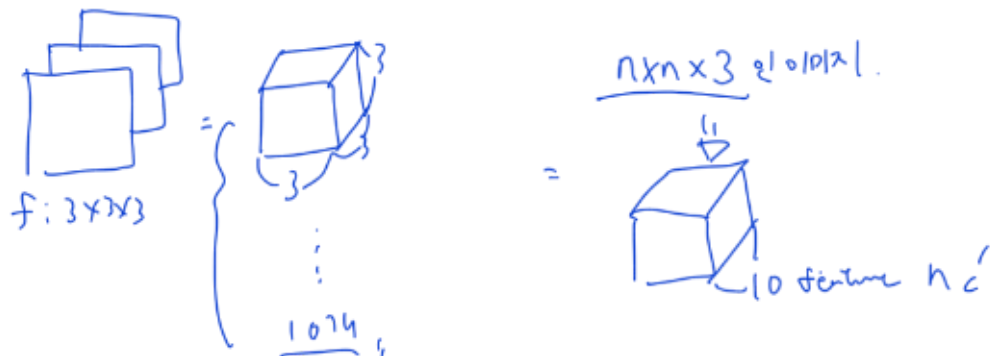
$$z^{[0]} = W^{[0]} a^{[0]} + b^{[0]}$$

$$a^{[0]} = g(z^{[0]})$$

결과:  $6 \times 6 \times 3$  의  $a^{[0]}$  에서 filter들의 합성곱을 통해

다음 layer  $4 \times 4 \times 2$  의  $a^{[1]}$  이 되었다.

~~연습문제~~ 연습문제! If you have 1674개의 ( $3 \times 3 \times 3$  인) in one layer of neural network, then how many parameters does that layer have?



각 filter 마다  $3 \times 3 \times 3 = 27$ 개의 변수를 갖고

+ bias 변수를 더하면  $27 + 1 = 28$ 개의 parameter를 1 필터가 갖는다.

총 16개 필터이므로  $28 \times 10 = 280$ , Parameter 문제.

$\therefore 280$  parameters

🌀 질문! input 이미지의 크기가 커짐에 따라 parameter와 관련된 문제도 생김!

$\therefore$  input 이미지 크기와 parameter의 수는 관련없으므로  
Overfitting을 방지할 수 있는 방법 중 하나 = parameter 수를 줄인다.

\* Summary : 합성곱 표현 if layer 'l' 가 합성곱 계층이면

Each filter:  $f^{[l]} = f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$   $f^{[l]} = \text{filter size (필터 크기)}$

Activation:  $a^{[l]} \rightarrow n_H^{[l]} \times n_W^{[l]} \times n_C^{[l]}$

$p^{[l]} = \text{'l' layer 에서의 패딩의 양.}$

$s^{[l]} = \text{'l' layer 에서의 stride 크기}$

$n_C^{[l]} = \text{number of filters}$

Input:  $n_H^{[l-1]} \times n_W^{[l-1]} \times n_C^{[l-1]}$  필터 수 (이전 layer를 통해 얻어진)

$= n_H^{[l-1]} \times n_W^{[l-1]} \times n_C^{[l-1]}$  : l-1인 이유는 (l-1) layer의 합성곱을 통해 얻어진 값이므로

output:  $n_H^{[l]} \times n_W^{[l]} \times n_C^{[l]}$

'l' layer를 통해 얻어진 결과

$$= \left\lfloor \frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor \times \left\lfloor \frac{n_W^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \right\rfloor \times n_C^{[l]}$$

필터의 갯수

'l' layer의 parameter 수 =  $(f^{[l]} \times f^{[l]} \times n_C^{[l-1]} + \underbrace{b}_{\text{bias}}) \times n_C^{[l]}$

필터 수

\* Weight:  $f^{[l]} \times f^{[l]} \times n_C^{[l-1]} \times n_C^{[l]}$   
 (개 필터) #f = 필터의 수 'l' layer의 필터 수

\* bias =  $n_C^{[l]} - (1, 1, 1, n_C^{[l]})$   
 $\downarrow$   
 필터마다 다른 값을 가지니까 상수라서

Activation:  $a^{[l]}$ 의 경우 단일 mini batch 등을 사용한다면

m개의 example에 대해서  $a^{[l]}: m \times n_H^{[l]} \times n_W^{[l]} \times n_C^{[l]}$

input 값은 이전 layer에서 나온 것이라  $n_H, n_W, n_C$ 가  $[l-1]$ 차.

output은  $l$  layer에서 계산 값이니까  $n_H, n_W, n_C$ 가  $[l]$ 차.

filter는  $f^{[l]}$ 로 해서 현재 layer의 크기를 갖되

channel 수는 input의 channel과 맞아야 하므로  
각 filter는  $\Rightarrow f^{[l]} \times f^{[l]} \times \underline{n_C^{[l-1]}}$

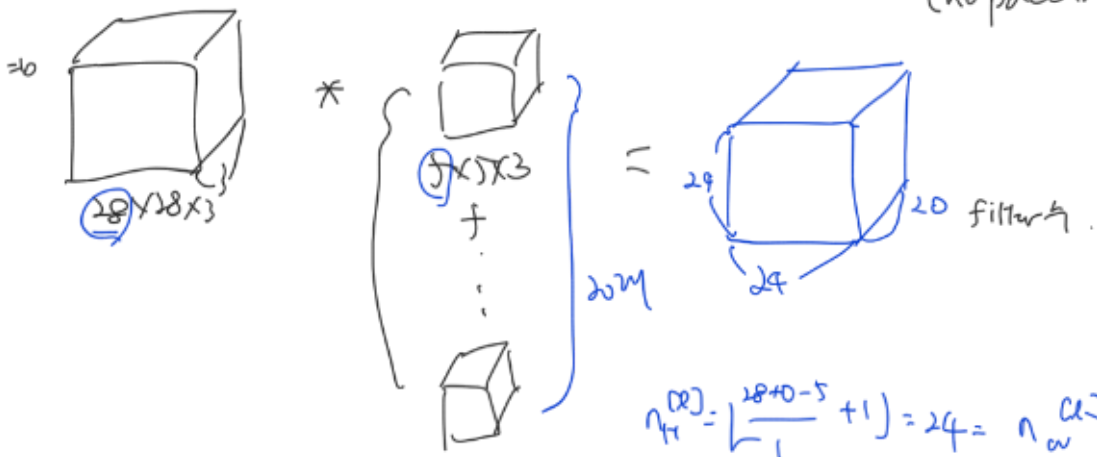
parameter 수는  $[f^{[l]} \times f^{[l]} \times n_C^{[l-1]} + B] \times \underline{n_C^{[l]}}$   
현재 layer의 필터 수

Output은  $n_H^{[l]} \times n_W^{[l]} \times n_C^{[l]}$  이고

$$= \left\lfloor \frac{n_H^{[l-1]} + 2p - f^{[l]}}{s} \right\rfloor + 1 \times \left\lfloor \frac{n_W^{[l-1]} + 2p - f^{[l]}}{s} \right\rfloor + 1 \times n_C^{[l]}$$

이때,

(예) 합성곱 연산.  $28 \times 28 \times 3$  img,  $5 \times 5$  filter 20744개.  
(no padding,  $s=1$ )



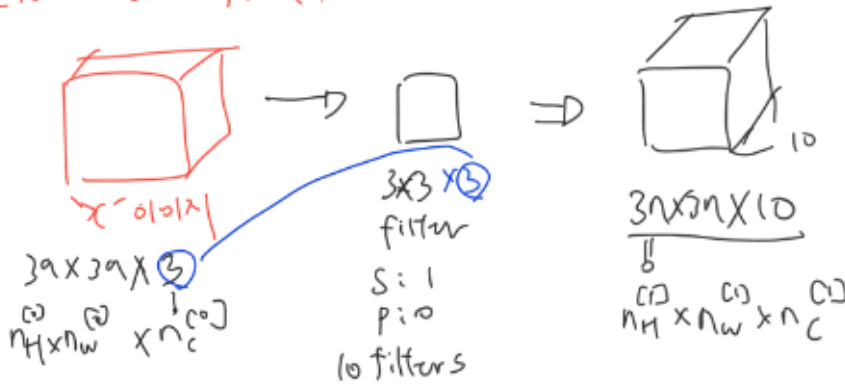
$\therefore$  output:  $24 \times 24 \times 20$

parameter 수는:  $[f^{[l]} \times f^{[l]} \times n_C^{[l-1]} + \underbrace{1}_{\text{bias}}] \times n_C^{[l]}$

$$= (5 \times 5 \times 3 + 1) \times 20$$

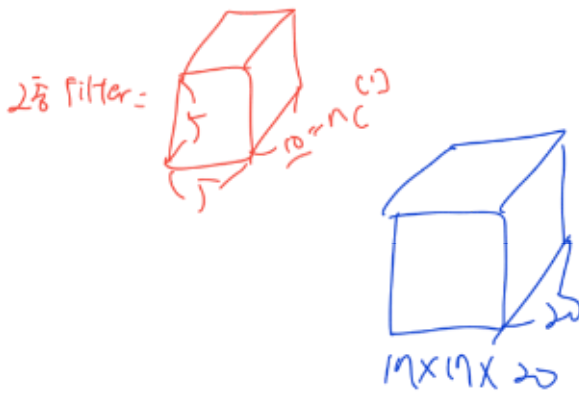
$$= 16 \times 20 = \underline{1600} \text{ parameter}$$

# Ex Conv Net.



1층 parameter  
 $= (3 \times 3 \times 3 + 1) \times 10$   
 $= 280$

① layer



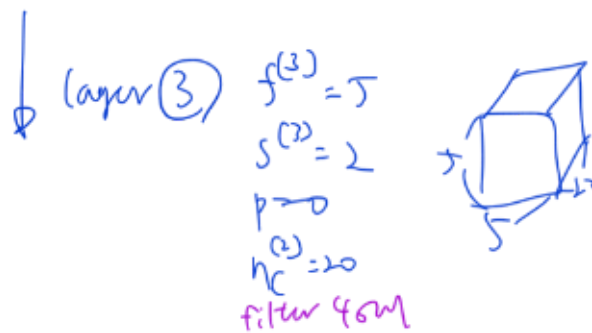
$f^{(2)} = 5$   
 $S^{(2)} = 2$   
 $P = 0$   
 20 filters  
 $= \left\lfloor \frac{17-5}{2} + 1 \right\rfloor = 13$   
 $= n_W^{(2)}, n_H^{(2)}$

$251 \times 20$

② layer  $\Rightarrow$  2층 parameter =  $(5 \times 5 \times 5 \times 20 + 1) \times 20$   
 $= 5020$

1층  $\rightarrow$  2층 parameter가 280  $\rightarrow$  5020로 증가

layer가 복잡해짐에 따라 parameter 수가 증가함을 알 수 있다.



input =  $13 \times 13 \times 20$

output (layer 3) =  $\left\lfloor \frac{13-7}{2} + 1 \right\rfloor = 7 = n_H^{(3)} = n_W^{(3)}$

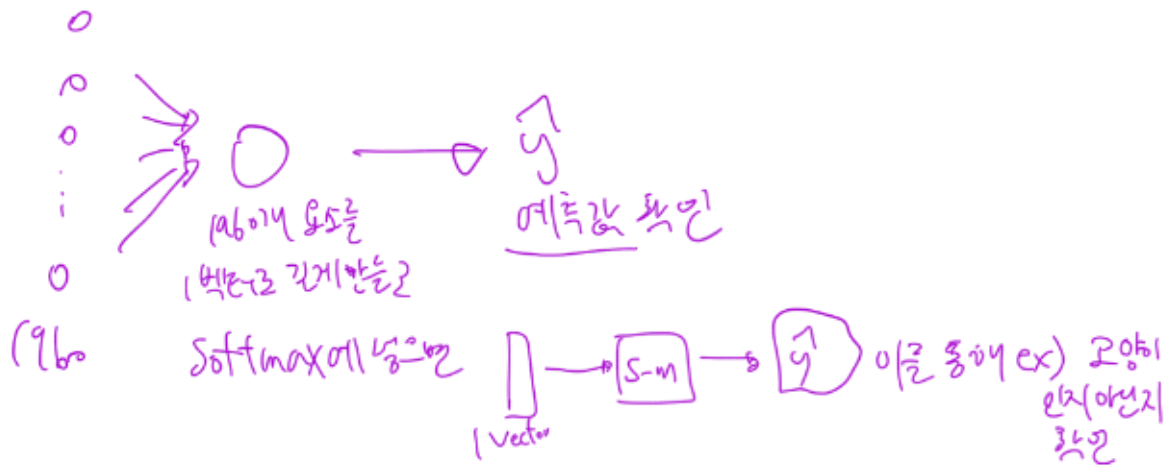
$\therefore 7 \times 7 \times 40$

$(1960)$



parameter  
 $= (7 \times 7 \times 7 \times 20 + 1) \times 40$   
 $= 20040$

이를 1 벡터로 펼쳐서 196개의 요소로 받는다.



회피커리라이터란 filter를 주나 S, p를 결정해준다.

신경망이 깊어질수록 39 → 11 → 7로 이미지가 줄어든다.

반면 채널은 3 → 10 → 20 → 40로 증가  $n_c^{(k)} \uparrow$

일반적인 Types of layer in a convolutional network.

- ① - convolution (합성곱층) = (CONV)
- ② - Pooling (POOL)
- ③ - Fully connected (FC)