

[Week15]_문가을

딥러닝 4단계 : 합성곱 신경망 네트워크 (CNN)

1. 합성곱 신경망

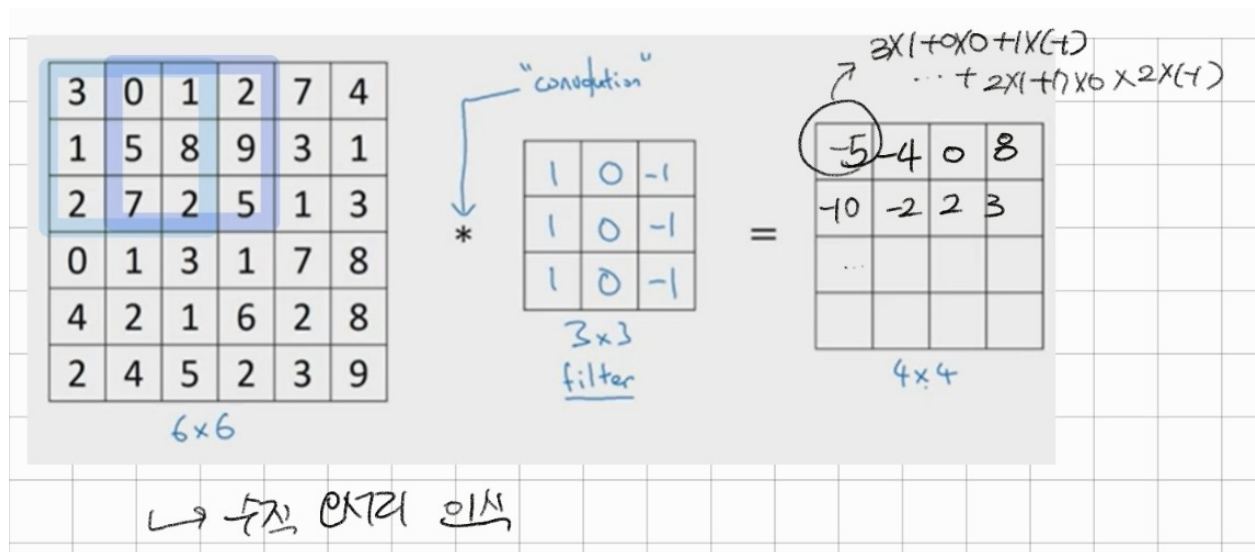
컴퓨터 비전

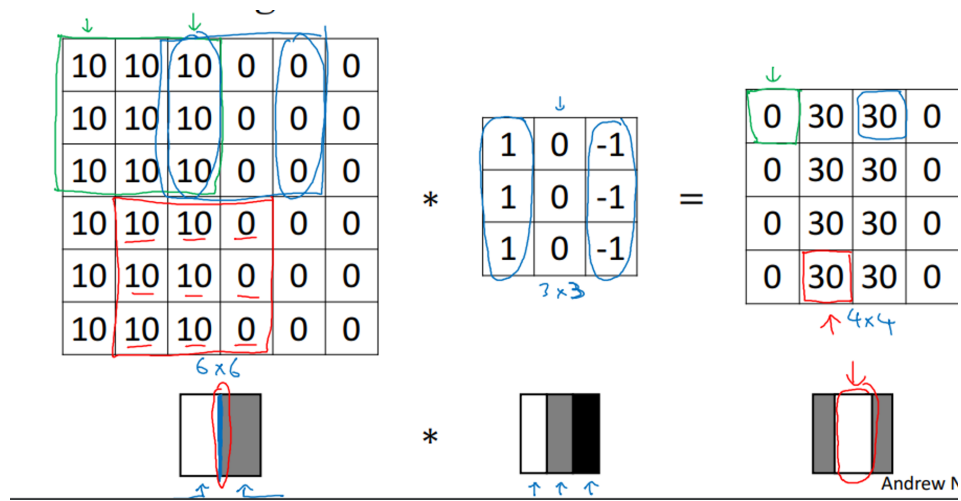
컴퓨터 비전에서 가장 큰 장애는 입력이 매우 클 수 있다는 것임.

→ 각 픽셀이 특성이 됨.

→ 합성곱 연산을 통해 해결.

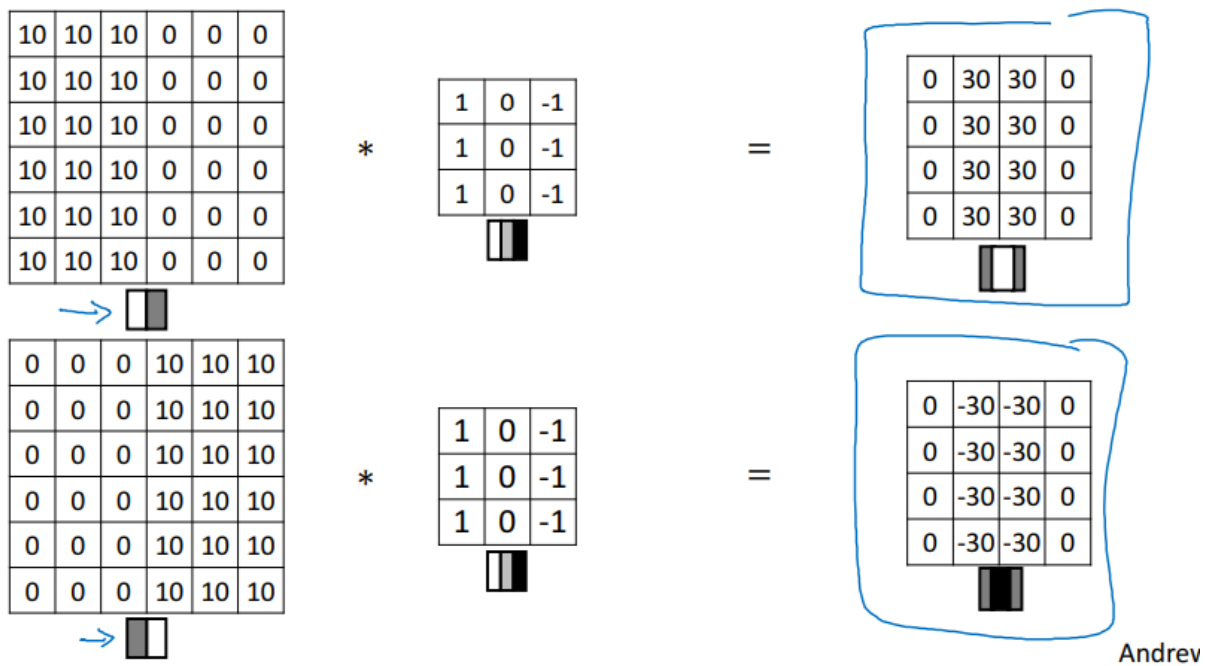
모서리 감지 예시



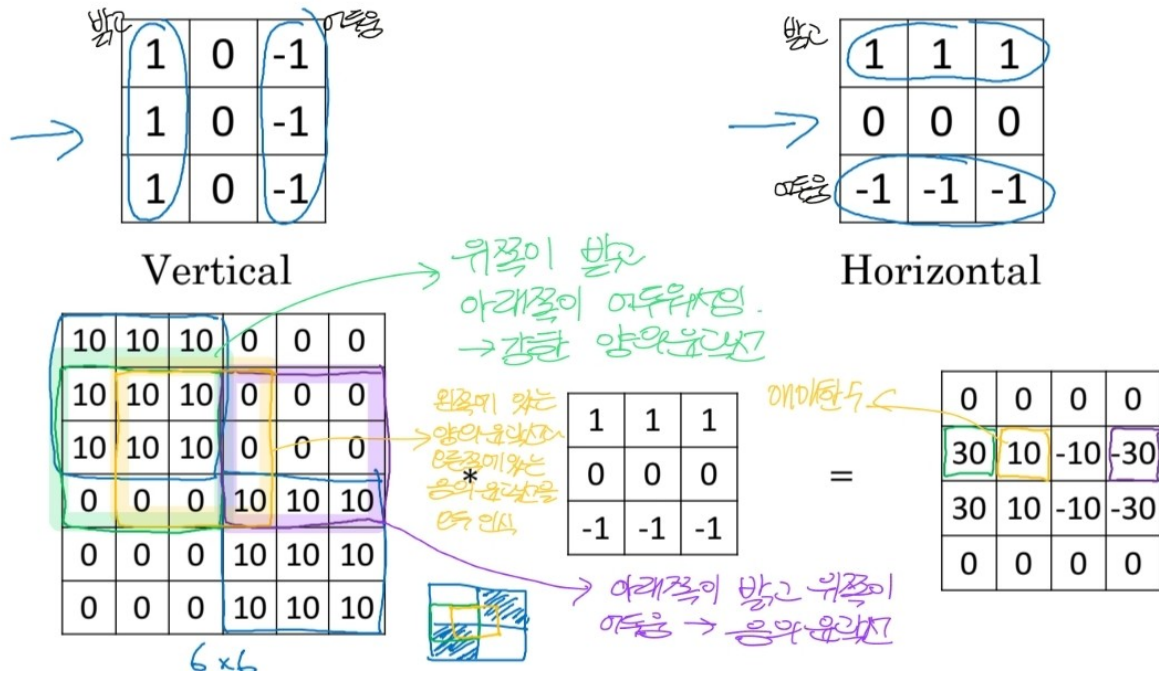


→ 결과 이미지의 가운데 밝은 영역이 이미지 가운데 강한 경계선이 있음을 알려줌.

더 많은 모서리 감지 예시



→ 밝은 곳에서 어두운 곳으로 가는 것과 그 반대를 나타내줌.



→ 서로 다른 필터는 세로 또는 가로의 윤곽선을 검출해냄.

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

vertical

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

sobel

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

scharr

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

horizontal

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

sobel

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

scharr

- sobel filter : 중간 부분의 픽셀에 더 중점을 둠.

filter 숫자를 지정하지 않고, 역전파를 통해 알아서 학습시켜서 적합한 필터를 만드는 방법을 사용함.

패딩 (padding)

- convolution

($n \times n$) 이미지를 ($f \times f$) 필터에 적용시키면 출력되는 차원은 ($n - f + 1, n - f + 1$)임.

→ 이미지가 줄어듦.

→ 가장자리에 있는 픽셀은 결과 이미지에 단 한번만 사용됨. 이미지 가장자리의 정보들을 버리게 됨.

두 문제를 해결하기 위해 padding 사용

→ 가장자리에 p 픽셀 만큼 추가로 더해줌.

결과 이미지는 ($n + 2p - f + 1$, $n + 2p - f + 1$)

<valid(유효) and same(동일) convolutions>

- valid convolution : 패딩이 없음.
- same convolution : 패딩을 한 뒤 결과 이미지가 인풋 사이즈와 같음.

→ $p = (f - 1) / 2$

일반적으로 CV에서는 f는 홀수임.

- f가 짝수라면, 패딩이 비대칭이되기 때문. 왼쪽과 오른쪽을 다르게 패딩해줘야 함. 홀수여야 합성곱에서 동일한 크기로 패딩을 더해줄 수 있음.
- 홀수면, filter에서 중심 위치가 존재함.

스트라이드(stride)

filter가 이동하는 칸 수

$n \times n$ image $f \times f$ filter

padding p stride s

Output Size:

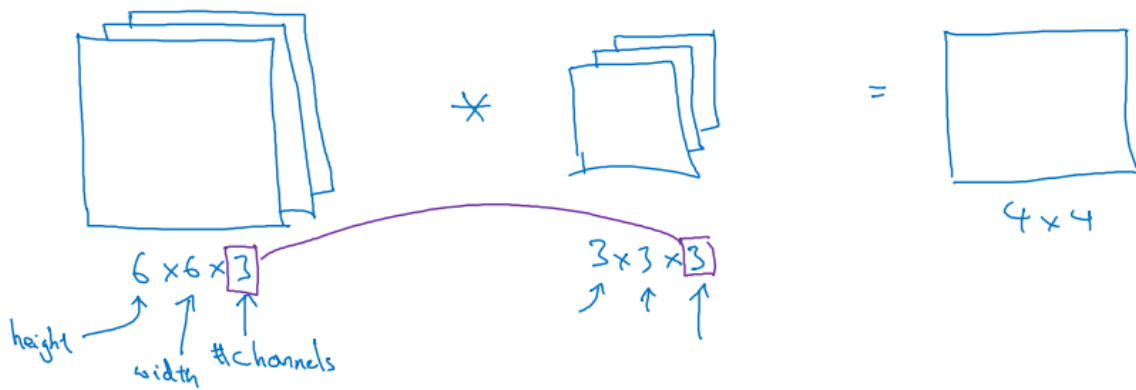
$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

→ 분수꼴이 정수가 아니라면 내림 해줌.

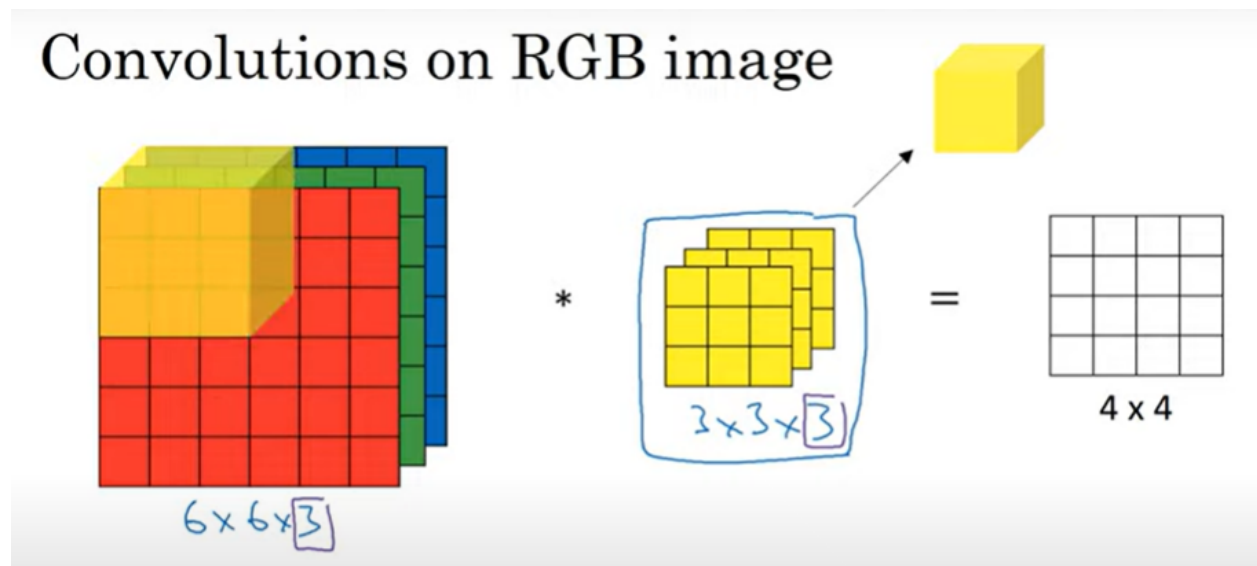
- 일반적으로 수학에서 정의하는 합성곱은 합성곱을 하기 전에 필터를 가로축과 세로축으로 뒤집는 연산을 해줘야 함.
- 지금까지 배운 합성곱은 사실 교차상관 이지만 딥러닝에서는 관습적으로 합성곱이라고 함.
- 딥러닝에서는 뒤집는 연산을 생략함. 이 뒤집는 과정은 신호처리에서는 유용하지만 심층 신경망 분야에서는 아무런 영향이 없기 때문에 생략하게 됨.

입체형 이미지에서 합성곱

3D 이미지 (RGB 이미지)



이미지의 채널 수는 필터의 채널 수와 일치해야 함.



filter의 27개의 숫자와 곱해주고 더함.

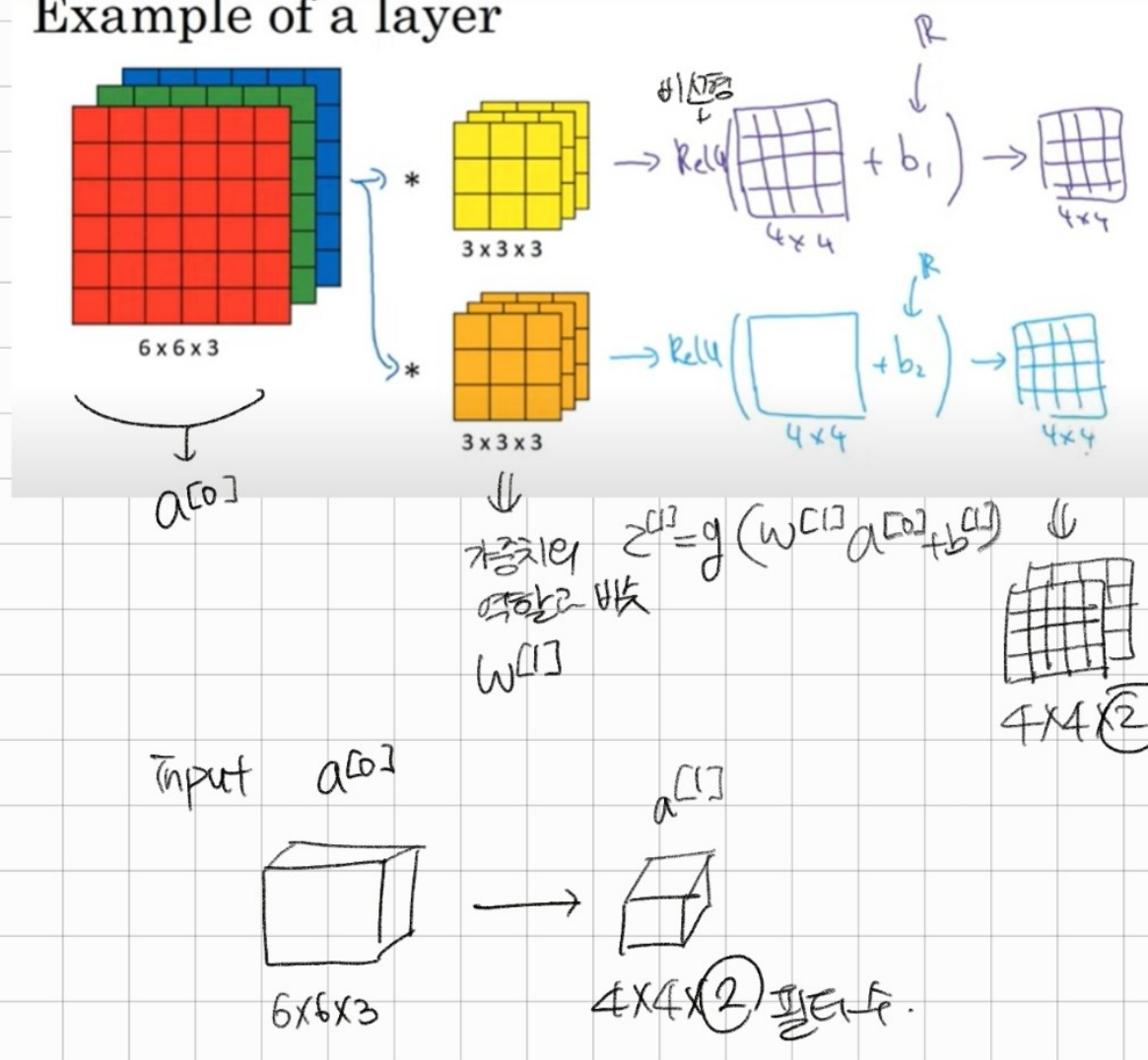
필터가 빨간색이나 초록색 또는 파란색의 채널 단 하나만 보는 것도 가능함.

multiple filters

- 패딩과 스트라이드가 없다고 가정했을 때, 최종 출력으로
 $(n \times n \times n_c) * (f \times f \times n_c) = (n - f + 1) \times (n - f + 1) \times n_{c'}$ 형태가 됩니다.
 - n : 이미지의 크기
 - n_c : 채널의 개수
 - f : 필터의 크기
 - $n_{c'}$: 사용된 필터의 개수
-

합성곱 네트워크의 한 계층 구성하기

Example of a layer



< Number of parameters in one layer >

If you have 10 filters that are $3 \times 3 \times 3$ in one layer of a neural network, how many parameters does that layer have?

→ 하나의 필터 당 파라미터 수 : $3 \times 3 \times 3 + 1 = 28$

→ 10개의 필터 : $28 \times 10 = 280$

image의 크기에 상관 없이 파라미터 수는 고정됨.

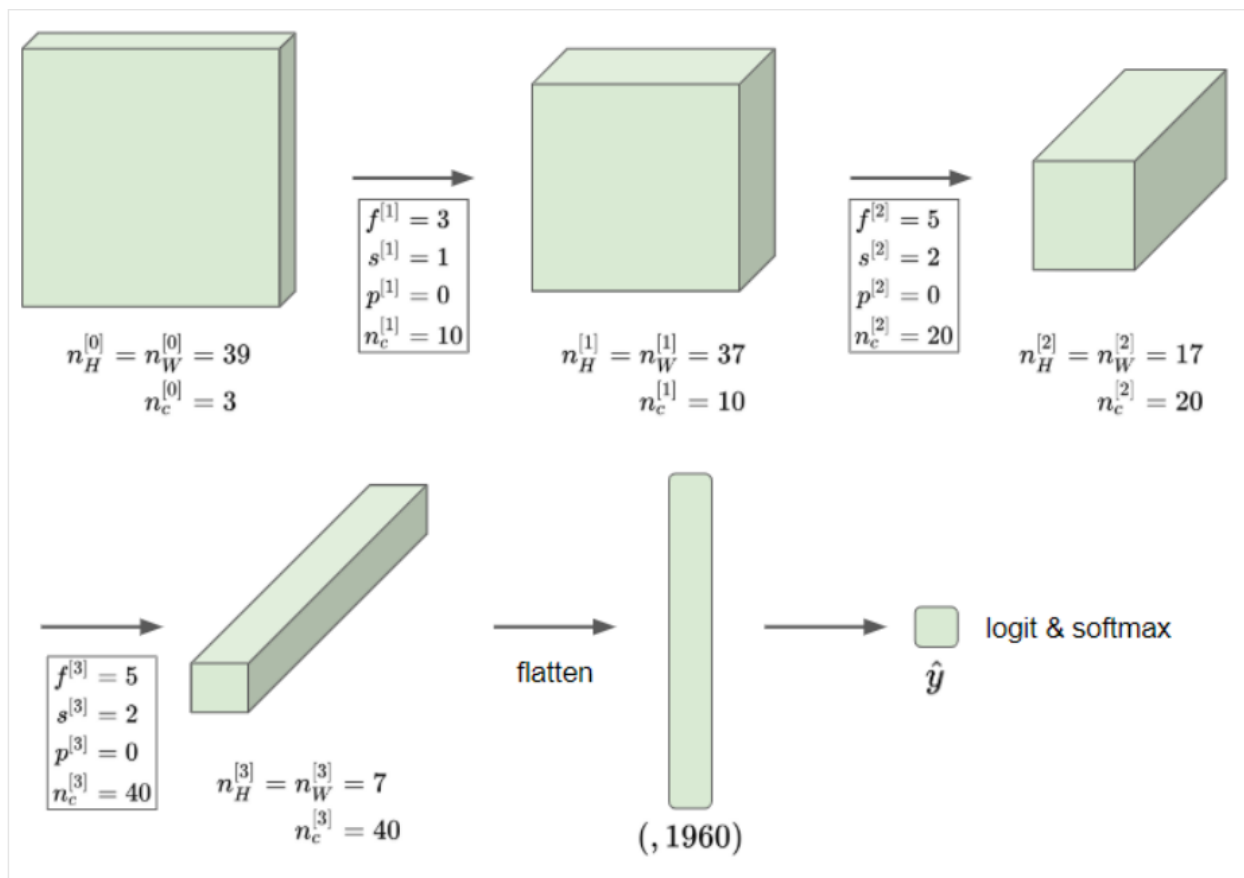
→ 과대 적합을 방지하는 성질임.

<notation>

- l : l 번째 계층
- $f^{[l]}$: 필터의 크기
- $p^{[l]}$: 패딩의 양
- $s^{[l]}$: 스트라이드 크기
- n_H : 이미지의 높이
- n_W : 이미지의 넓이
- n_c : 채널의 수

- 다음 연산이 l 번째 층의 연산이면, 이전 층 ($l-1$) 의 이미지의 크기는 $n_H^{[l-1]} \times n_W^{[l-1]} \times n_c^{[l-1]}$ 가 됩니다. 그 결과로 나오는 이미지의 크기는 $n_H^{[l]} \times n_W^{[l]} \times n_c^{[l]}$ 입니다.
 - l 번째 층의 높이 혹은 넓이의 크기연산 공식은 아래와 같습니다.
 - $n_H^{[l]} = \frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1$
 - $n_W^{[l]} = \frac{n_W^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1$
- $n_c^{[l]}$ 개의 크기가 $f^{[l]} \times f^{[l]} \times n_c^{[l-1]}$ 인 필터가 합성곱 연산을 진행하게 됩니다. 그리고 활성화 함수를 거쳐 l 번째 층의 결과값이 계산됩니다. 합성곱 연산에 사용된 변수는 총 $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]}$ 개 입니다. 추가된 편향까지 더하면 한 층의 합성곱 신경망에 필요한 변수는 $f^{[l]} \times f^{[l]} \times n_c^{[l-1]} \times n_c^{[l]} + n_c^{[l]}$ 개가 됩니다.

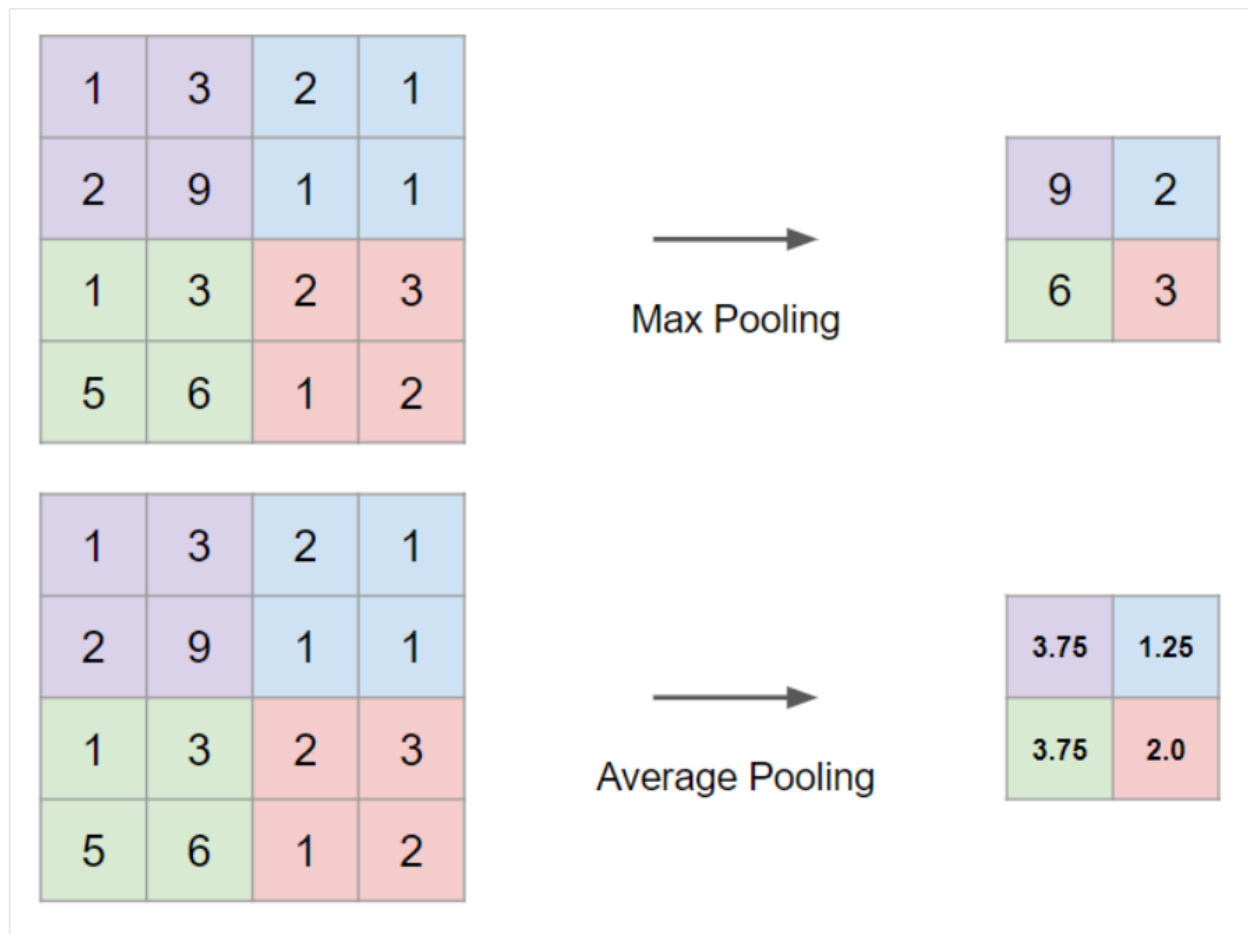
간단한 합성곱 네트워크 예시



<Types of layers in a convolutional network>

- Convolution (CONV, 합성곱층)
- Pooling (POOL)
- Fully connected (FC)

풀링(Pooling)층



$f = 2, s = 2$

Max pooling

- 가장 큰 수가 특정 특성을 의미할 수 있기 때문에 사용.
- 학습할 수 있는 변수가 없음. $\rightarrow f, s$

최대 풀링이 평균 풀링 보다 많이 쓰임

Hyperparameters:

f : filter size

s : stride

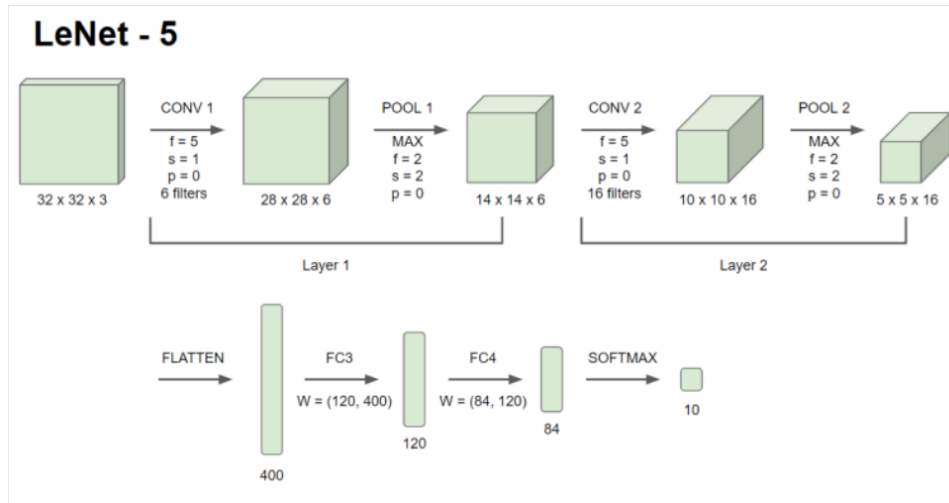
Max or average pooling

convolution과 똑같이

$$\begin{aligned} \circ n_H^{[l]} &= \frac{n_H^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \\ \circ n_W^{[l]} &= \frac{n_W^{[l-1]} + 2p^{[l]} - f^{[l]}}{s^{[l]}} + 1 \end{aligned}$$

학습하는 변수가 없음

CNN 예시



<하이퍼 파라미터 선정>

- 문헌에서 다른 사용자들에게 작동했던 하이퍼 파라미터 차용.
- 신경망이 깊어질수록 높이와 너비는 감소함.
- 채널의 수는 증가함.
- conv → pool → conv → pool → ... → FC

	Activation shape	Activation Size	# parameters
Input:	(32,32,3)	3,072 $a^{(0)}$	0
CONV1 (f=5, s=1)	(28,28,8)	6,272	208
POOL1	(14,14,8)	1,568	0
CONV2 (f=5, s=1)	(10,10,16)	1,600	416
POOL2	(5,5,16)	400	0
FC3	(120,1)	120	48,001
FC4	(84,1)	84	10,081
Softmax	(10,1)	10	841

왜 합성곱을 사용할까요?

<합성곱 층을 사용할 때의 이점>

- 변수 공유(parameter sharing) : 어떤 한 부분에서 이미지의 특성을 검출하는 필터가 이미지의 다른 부분에서도 똑같이 적용되거나 도움이 됨.
- 희소 연결 : 출력값이 이미지의 일부에 영향을 받고, 나머지 픽셀들의 영향을 받지 않기 때문에, 과대 적합 방지.
- 이동 불변성을 포착하는 데에도 용이, 이미지가 약간의 변형이 있어도 포착 가능.