딥러닝 스터디 3주차

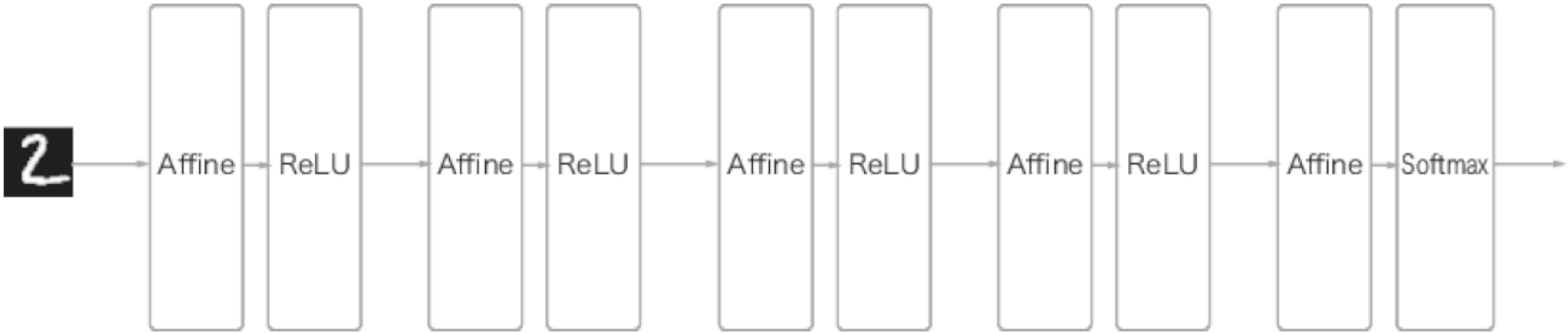
합성곱 신경망 Part 1

박소현

합성곱 신경망의 구조

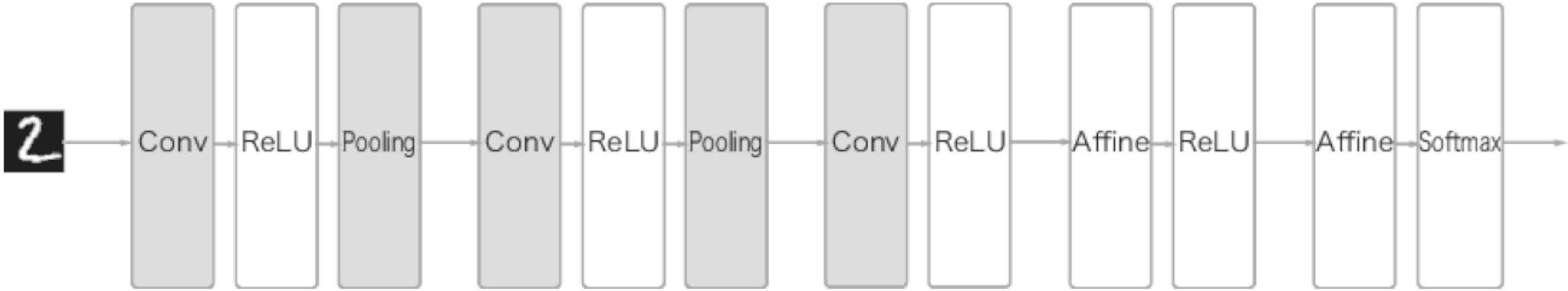
기존 신경망과의 비교

기존 신경망
- 완전 연결 계층 형태
- 인접하는 모든 계층의 뉴런이 서로 연결되어 있는 형태



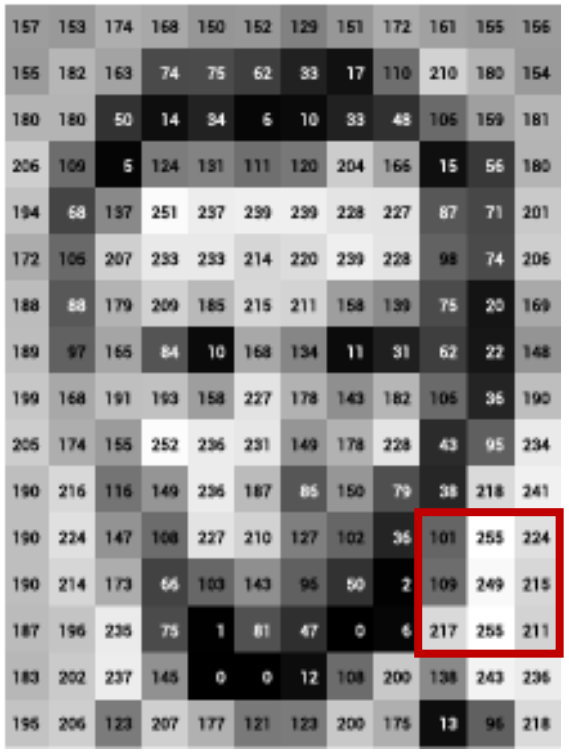
기존 신경망과의 비교

합성곱 신경망
- 기존 신경망의 Affine-ReLU를 Conv-ReLU-(Pooling)으로 바꾼 신경망



완전연결 신경망의 단점

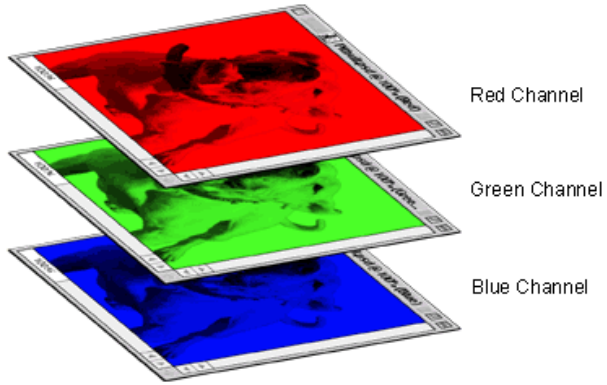
완전연결 신경망의 가장 큰 단점은 **원래 데이터(ex. 이미지) 자체의 형태가 무시된다는 것**



인접한 픽셀의 값이 비슷하다

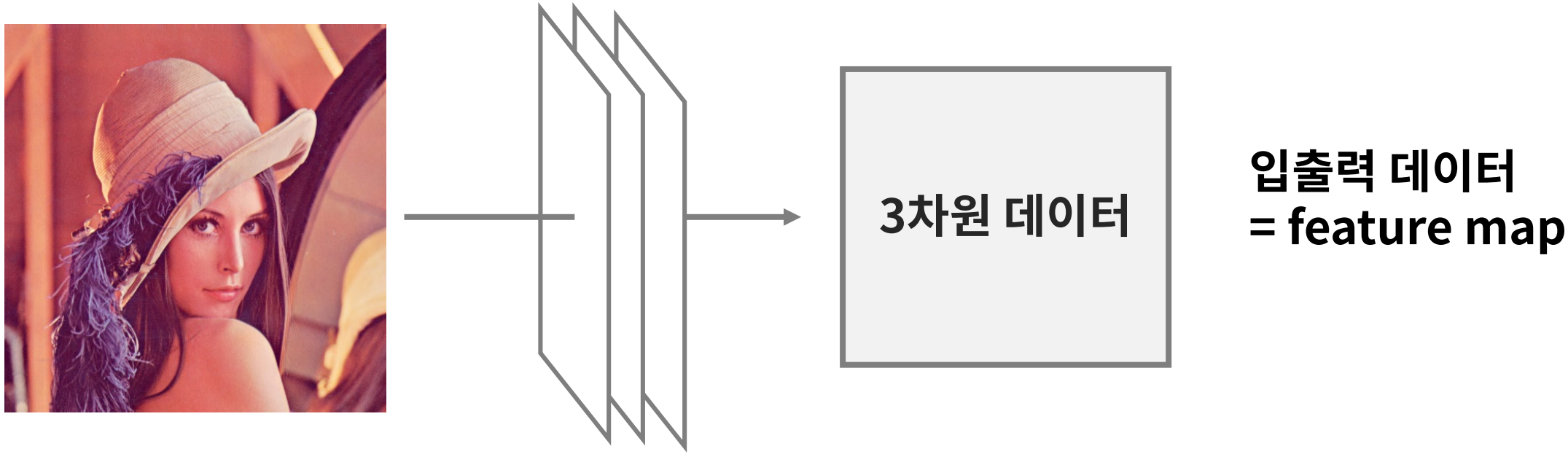
[157, 153, 174, …, 13, 96, 218]

RGB 채널끼리 밀접한 관련이 있다



그럼 합성곱 신경망은?

평탄화된 1차원 데이터가 아닌 3차원 데이터를 입력 받아서 3차원 데이터를 출력한다
→ **이미지의 특징**을 살릴 수 있다!



입출력 데이터
= feature map

합성곱 계층

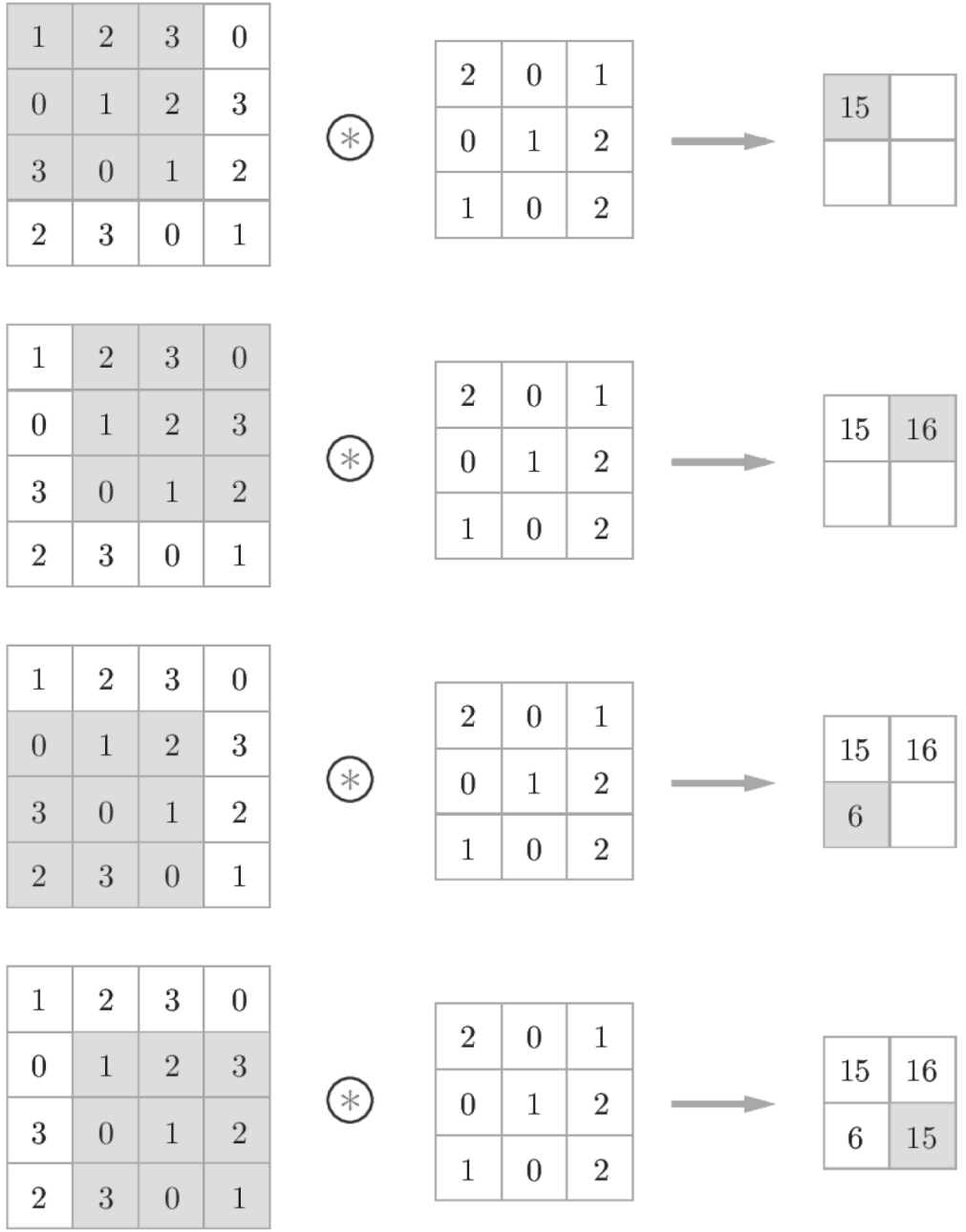
합성곱 연산

필터 연산

필터를 씌우고 왼쪽에서 오른쪽, 위쪽에서 아래쪽으로 움직이면서 연산을 하는 것

어떻게 연산을 할까?

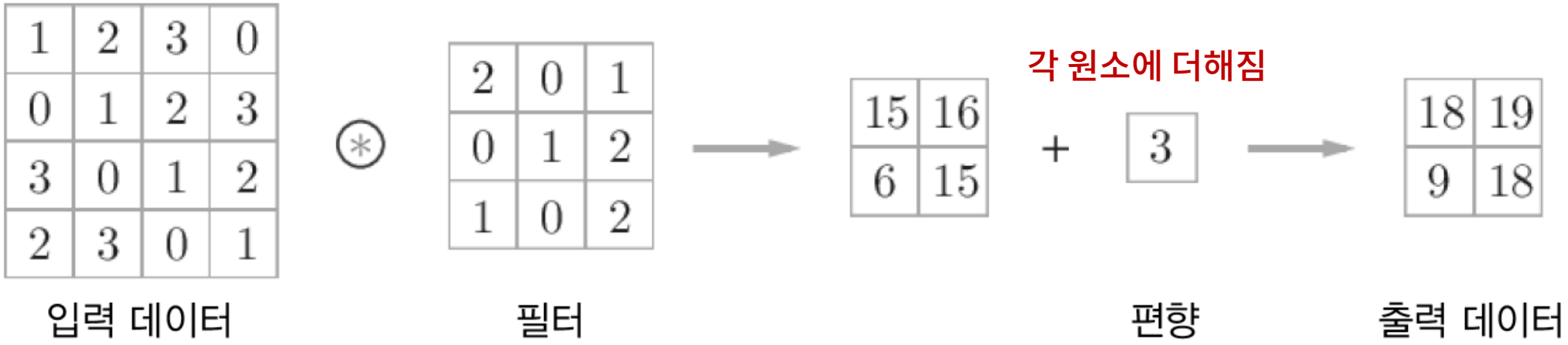
입력 데이터와 필터에서 대응하는 원소끼리 곱한 후에 그 합을 결과 데이터에 저장



합성곱 연산

그럼 편향은?

- 편향은 필터를 적용한 후에 더하면 됨
- 편향은 항상 1×1 크기인데 브로드캐스팅으로 각 원소에 더해짐



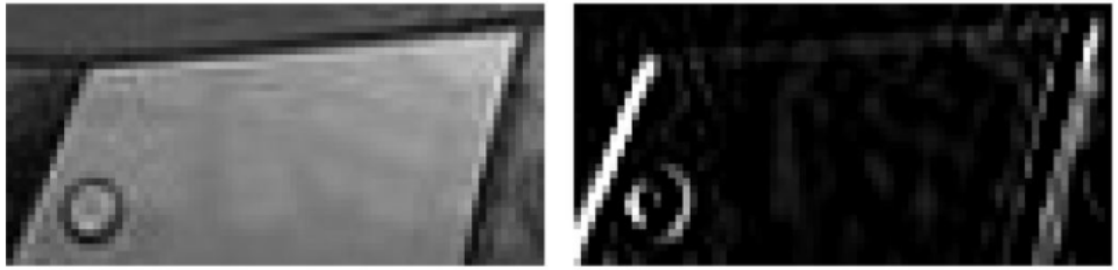
그럼 합성곱 연산을 통해서 무엇을 하는데?

이미지의 특징을 잘 골라낼 수 있도록 하는 연산이다!

- 신경망 학습을 반복하면서 이 필터의 값(= 매개변수)을 업데이트 함
- 이미지의 특징을 잘 찾는 필터가 되도록 함

$$\text{Convolution filter} = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

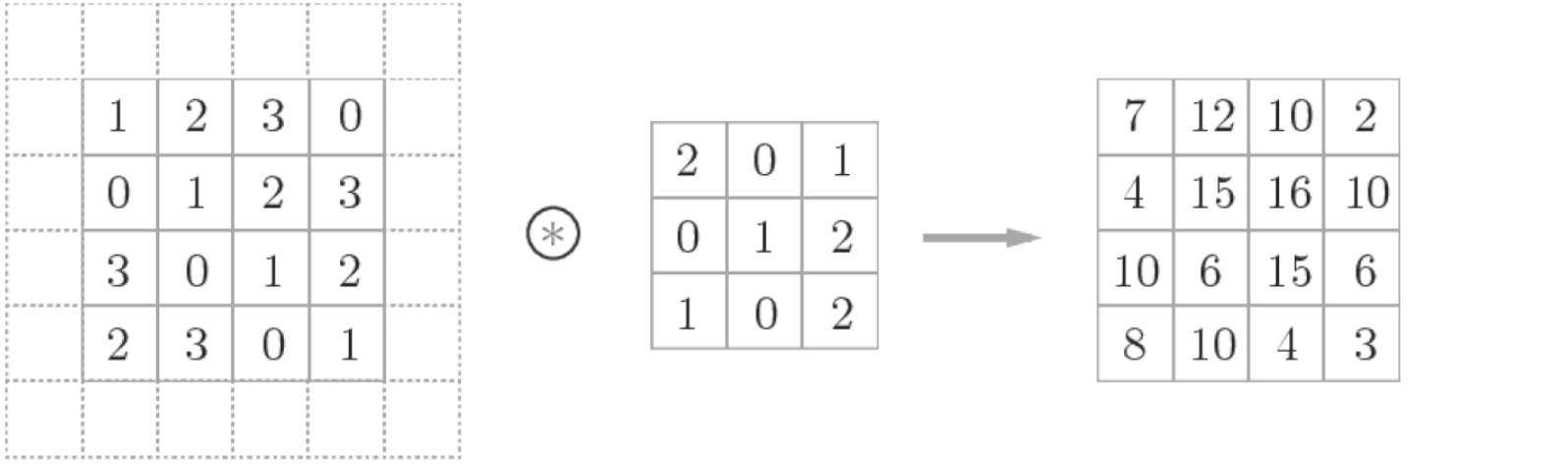
$$\text{Convolution filter} = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$



패딩

패딩(Padding)은 **합성곱 연산으로 인해 원 데이터의 크기가 줄어드는 것을 방지**하기 위한 것

- 합성곱 연산을 계속 하다가 결국에 출력 데이터의 크기가 1이 되어버리면 더이상 이 연산을 할 수 없음
- 우리가 추울 때 패딩을 입는 것처럼 특정 값(ex. 0)으로 원 데이터를 죽 둘러싸주는 것이다.



(4, 4)
입력 데이터(패딩 : 1)

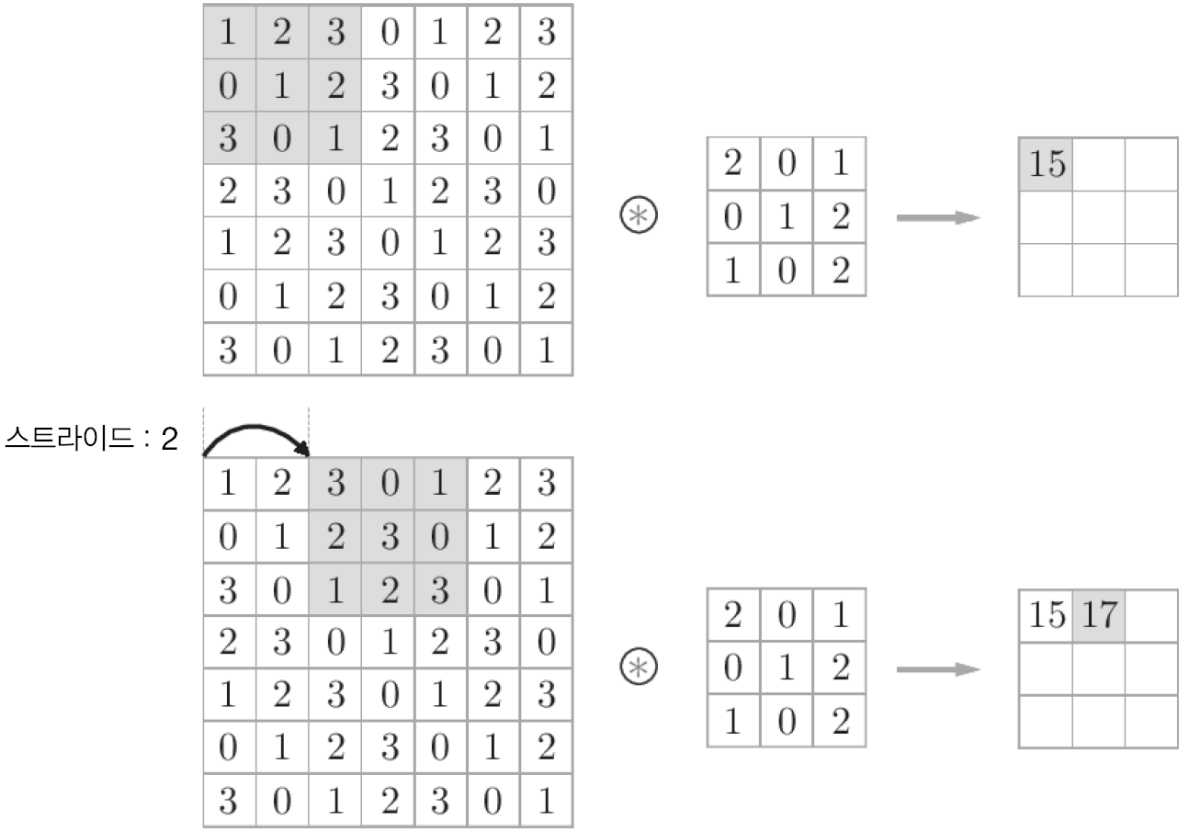
(3, 3)
필터

(4, 4) **크기가 유지**
출력 데이터

스트라이드

스트라이드(Stride)는 **필터를 하는 윈도우를 몇 칸씩 이동할지를** 말한다.

ex. 스트라이드 2이면 윈도우가 두 칸씩 이동을 한다.



패딩과 스트라이드를 적용한 출력 데이터

(H, W) : (입력 데이터의 높이, 입력 데이터의 너비)

(FH, FW) : (필터의 높이, 필터의 너비)

(OH, OW) : (출력 데이터의 높이, 출력 데이터의 너비)

P : 패딩, S : 스트라이드

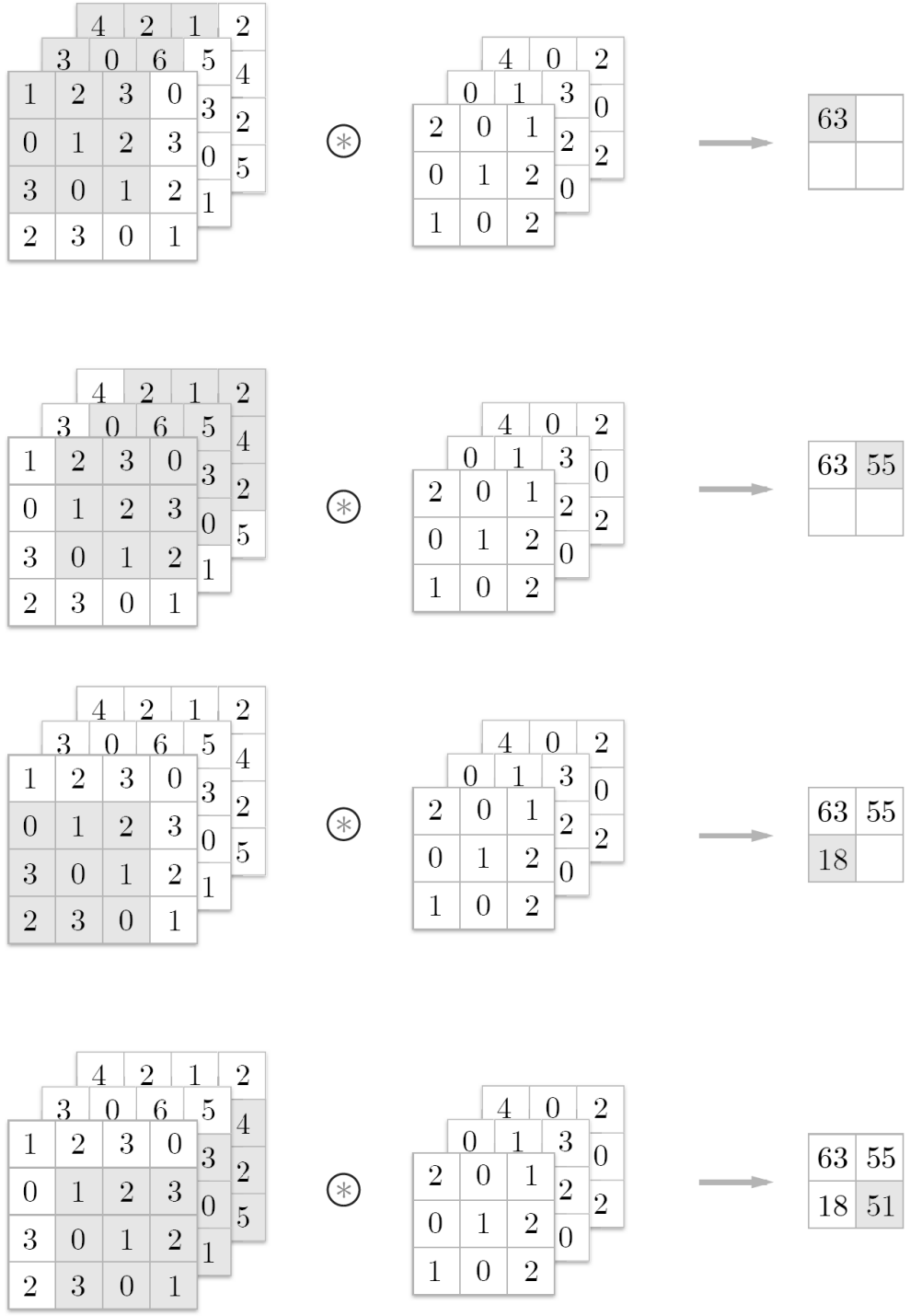
$$OH = \frac{H + 2P - FH}{S} + 1 \qquad OW = \frac{W + 2P - FW}{S} + 1$$

정수로 나누어 떨어져야함

3차원 데이터의 합성곱 연산

3차원 데이터는 (세로, 가로, 채널) 로 구성

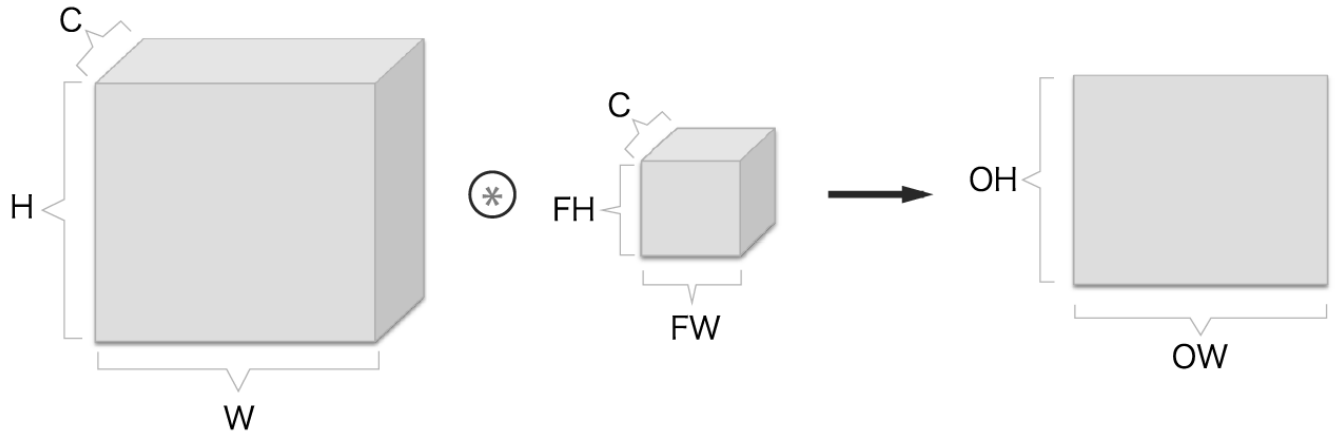
- 합성곱 연산시 **채널의 크기(=개수)만큼 필터가 존재**
- 각 채널에 맞추어진 필터가 있는데 각각 씌워서 연산 값을 다 합하는 거라 생각



블록으로 생각하기

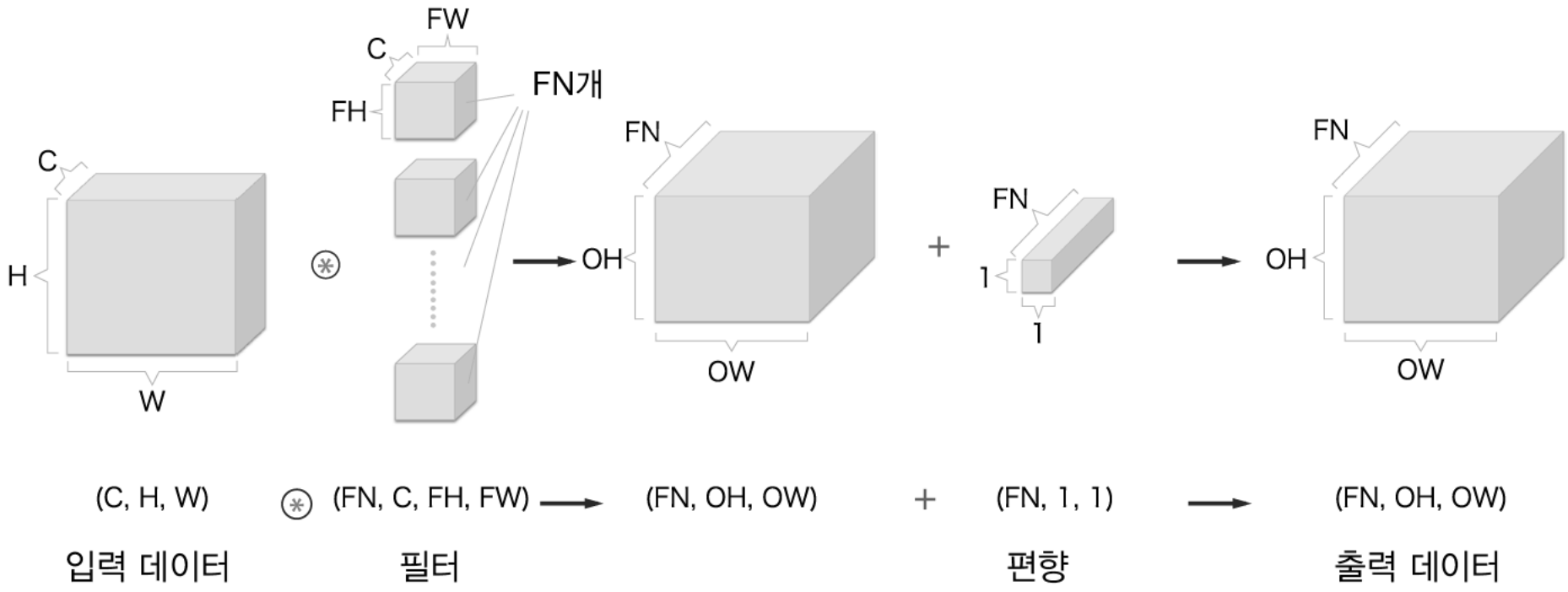
아까처럼 여러 겹의 종이를 겹친 형상을 생각하지 말고 **블록**으로 생각하면 쉽다. 이런 식으로 **구한 출력 데이터 하나가 이미지를 대표하는 특징인 feature map**이 되는 것이다.

[입력데이터가 1개이고, 이 입력데이터에 대한 필터도 1개인 경우]



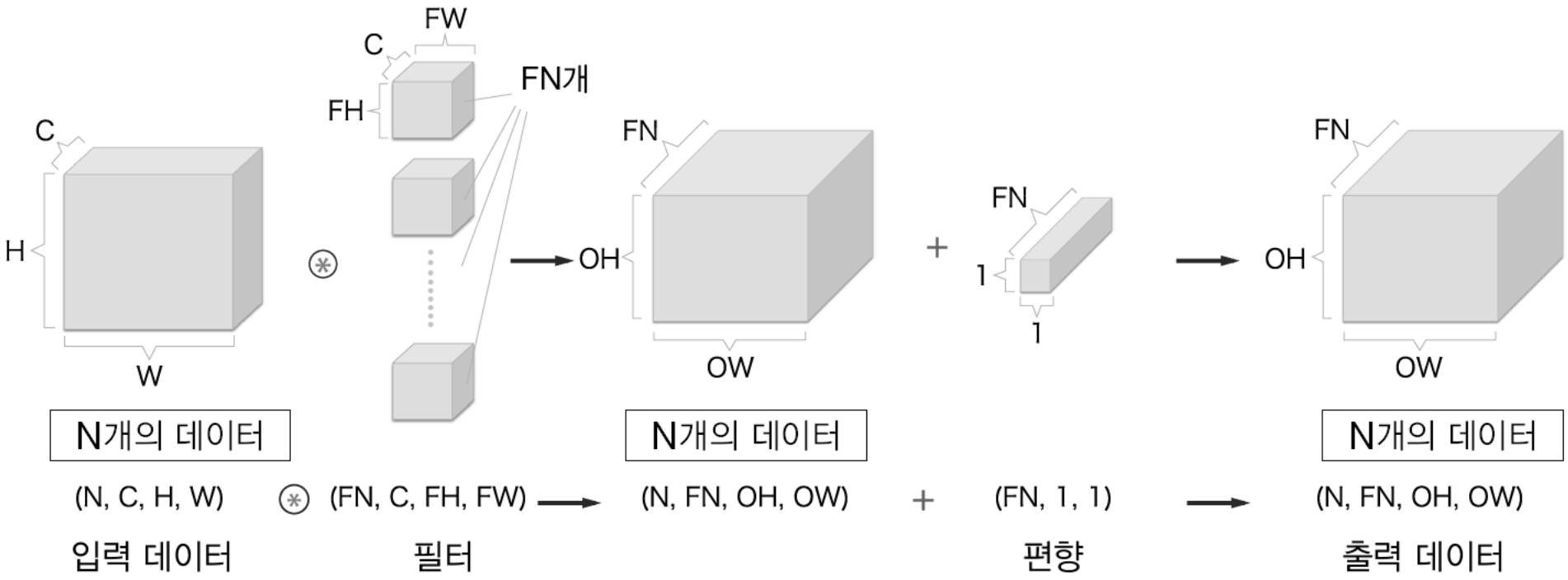
블록으로 생각하기

[입력데이터가 1개이고, 이 입력데이터에 대한 필터는 FN개인 경우]



블록으로 생각하기

[입력데이터가 N개이고, 이 입력데이터에 대한 필터는 FN개인 경우]



풀링 계층

풀링 계층은 무엇을 할까?

우리가 다루는 이미지 데이터는 생각 이상으로 매우 크고 데이터의 숫자가 많다.

엄청난 양의 연산이 필요하고 그 연산으로 인한 시간과 비용이 상당하다.

그래서 이 풀링 계층을 통해 이미지의 특징에서도 **대표적인 것**들만 쏙쏙 뽑아내는 것이다.

풀링 계층의 특징

1. 학습해야할 매개변수가 없다. (최대값 or 평균)

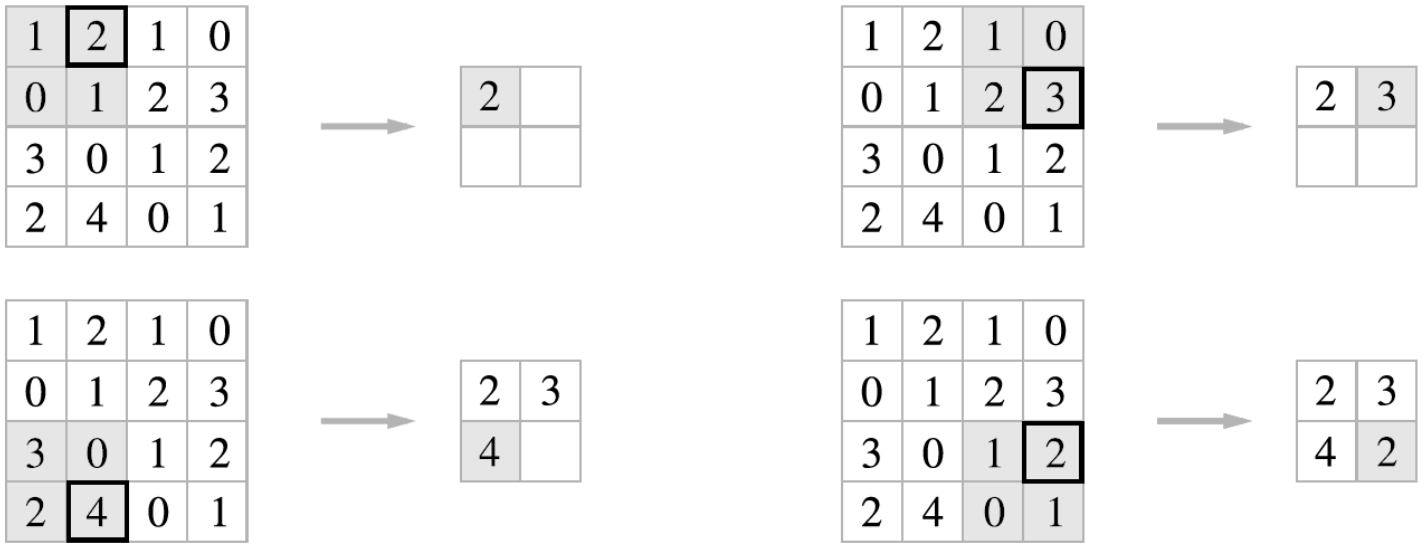
2. 채널의 개수가 변하지 않는다. 크기만 줄일 뿐 특징의 개수는 유지

3. 입력데이터가 조금 변화해도 풀링의 결과는 크게 변하지 않는다.

최대 풀링

풀링의 윈도우에 있는 값 중 가장 큰 값을 결과 값으로 하는 **최대 풀링**이 많이 쓰인다.
이 때 풀링의 윈도우 크기는 스트라이드와 같아야 한다.
왜? 윈도우가 이동을 해도 앞의 데이터와 겹치지 않게 하기 위해서이다.

[스트라이드 = 2이고 풀링 윈도우 크기 = 2일 때의 최대 풀링 예시]




QnA