

The background features a series of overlapping, wavy, ribbon-like shapes in various shades of green and white, creating a sense of motion and depth. The shapes flow from the left side towards the right, with some appearing more prominent than others.

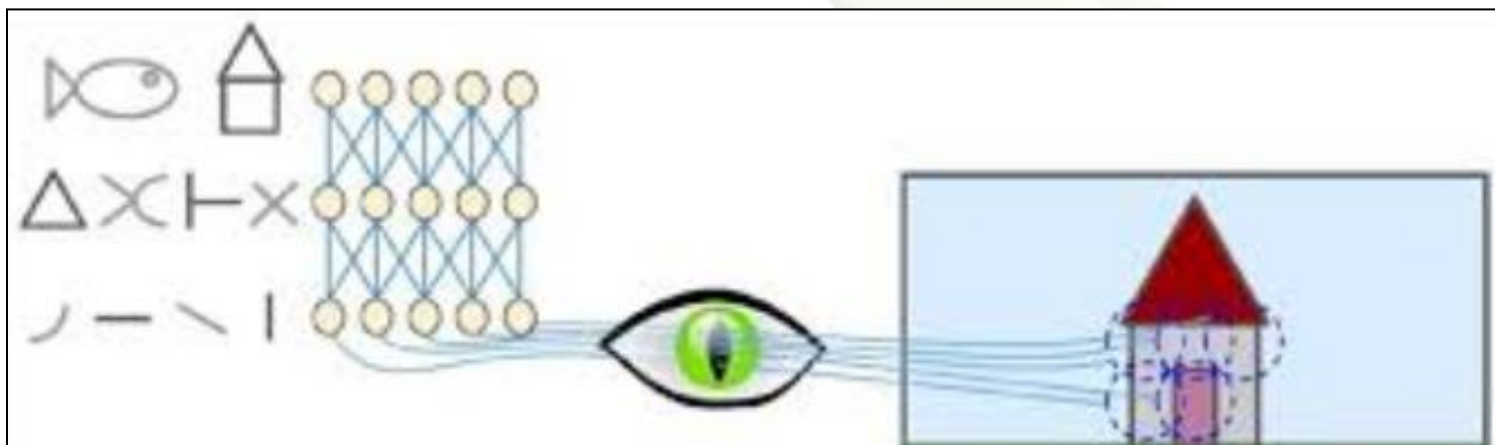
Introduction to CNN

합성곱 신경망(Convolutional Neural Network)

- 비교적 최근까지 컴퓨터를 사용해서 이미지 또는 영상에 포함된 사물 감지, 음성 인식 등의 작업을 수행하는 것은 그다지 효과적이고 안정적이지 못함
 - » 이는 주로 감각기관의 작동원리가 반영되지 않았기 때문
- 감각의 정보가 인간의 의식에 도달할 때 이미 고수준의 특성으로 채워진 상태
 - » 지각 대부분은 뇌의 특별한 시각, 청각 등의 감각기관에서 처리됨
- 합성곱 신경망은 대뇌의 시각 피질 연구에서 시작
 - » 1980년대부터 이미지 인식 분야에 사용
 - » 컴퓨터 성능 향상 및 심층 신경망 훈련 기술 발전에 힘입어 성능이 크게 향상됨
 - » 이미지 검색 서비스, 자율주행 자동차, 영상 자동 분류 시스템 등에 크게 기여
 - » 음성 인식, 자연어 처리 등의 작업에도 사용

합성곱 신경망 (Convolutional Neural Network)

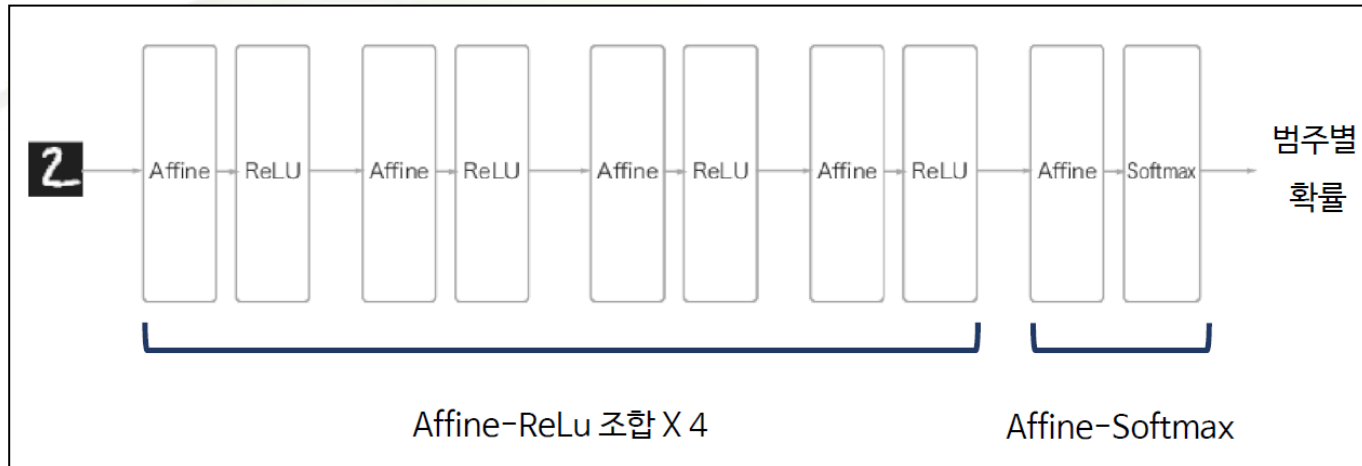
- 1958, 1959년 데이비드 허블과 토르스텐 비셀은 다양한 실험을 통해 시각 피질의 구조에 대한 주요한 사실 증명
 - » 뉴런들이 시야의 일부 범위 안에 있는 시각 자극에만 반응
 - » 뉴런의 수용장들은 서로 겹칠 수 있어서 합치면 전체 시야를 포괄
 - » 고수준 뉴런은 이웃한 저수준 뉴런의 출력에 기반



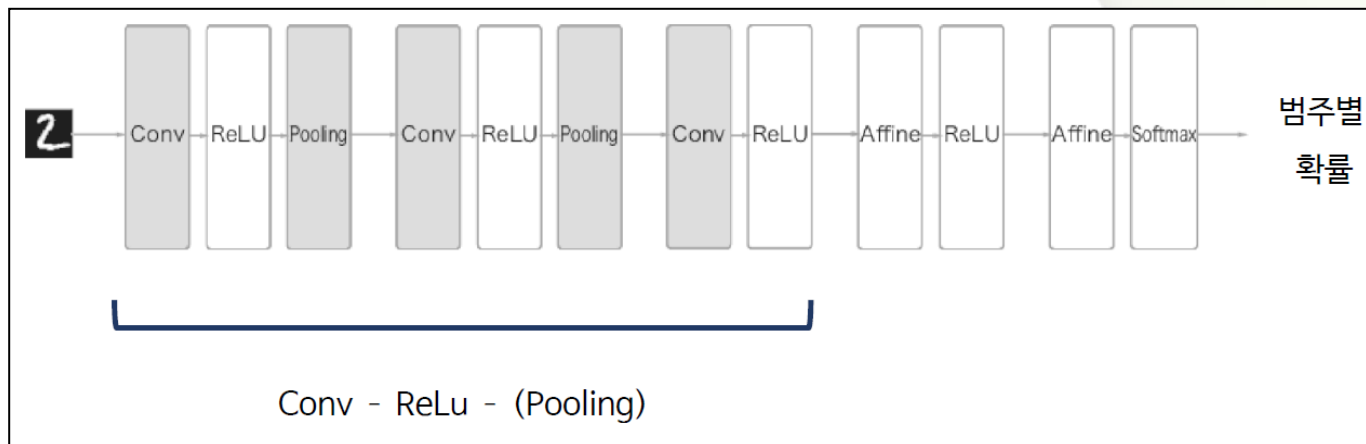
- 시각 피질에 대한 연구는 1980년에 소개된 신인식기(neocognitron)에 영향을 주었고 점진적으로 합성곱 신경망으로 진화
 - » 1998년 얀 르쿤, 레옹 보토, 요슈아 벤지오, 패트릭 하프너의 논문이 중요한 전환점

합성곱 신경망 구조

■ 기본 신경망 구조 (완전 연결 구조)

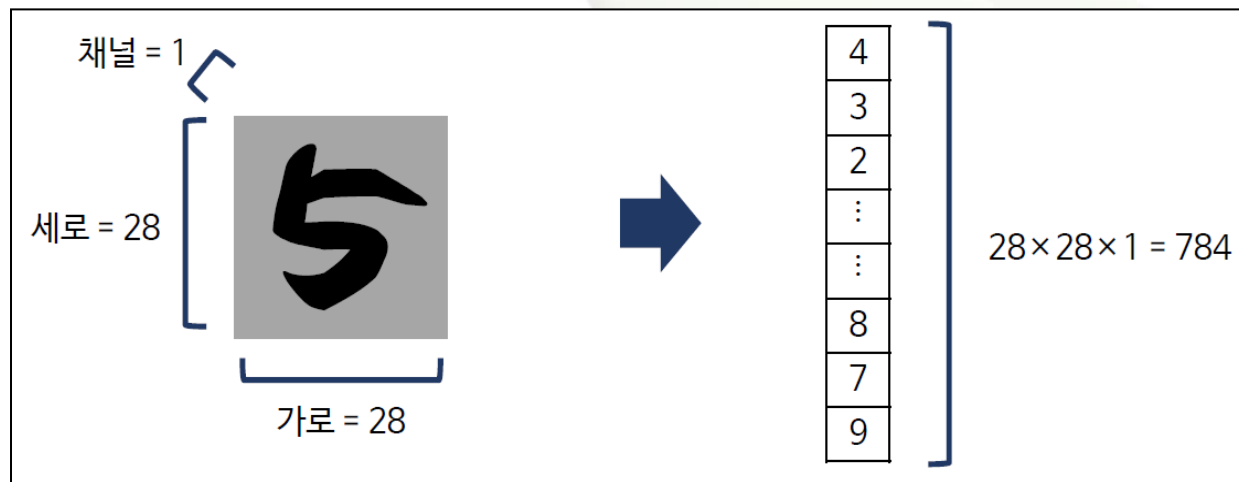
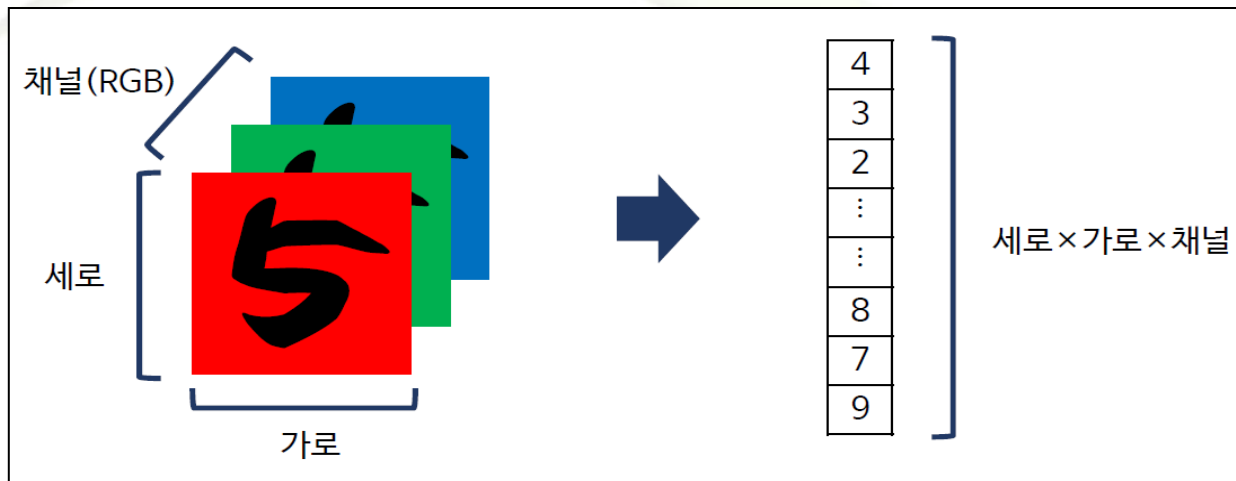


■ 합성곱 신경망 구조



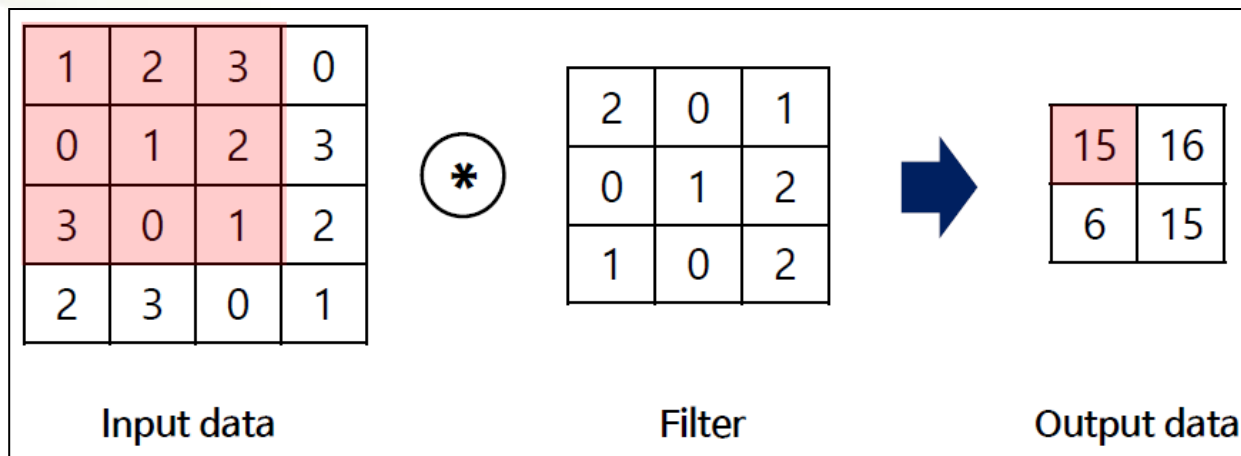
완전 연결 계층 문제

- 3차원 데이터인 이미지가 완전연결 계층에 입력될 때 1차원 데이터로 변형되면서 공간 정보 손실

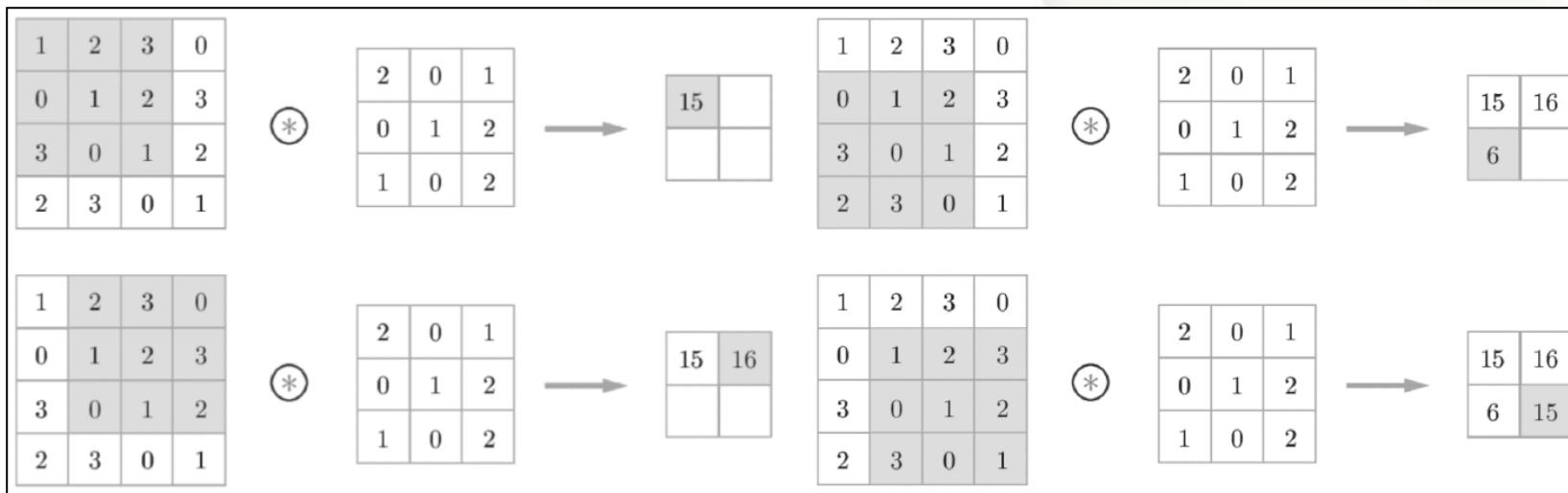


합성곱

- 특정 높이와 너비의 필터(Kernel Filter)를 일정 간격(Stride)으로 이동하면서 입력 데이터에 적용

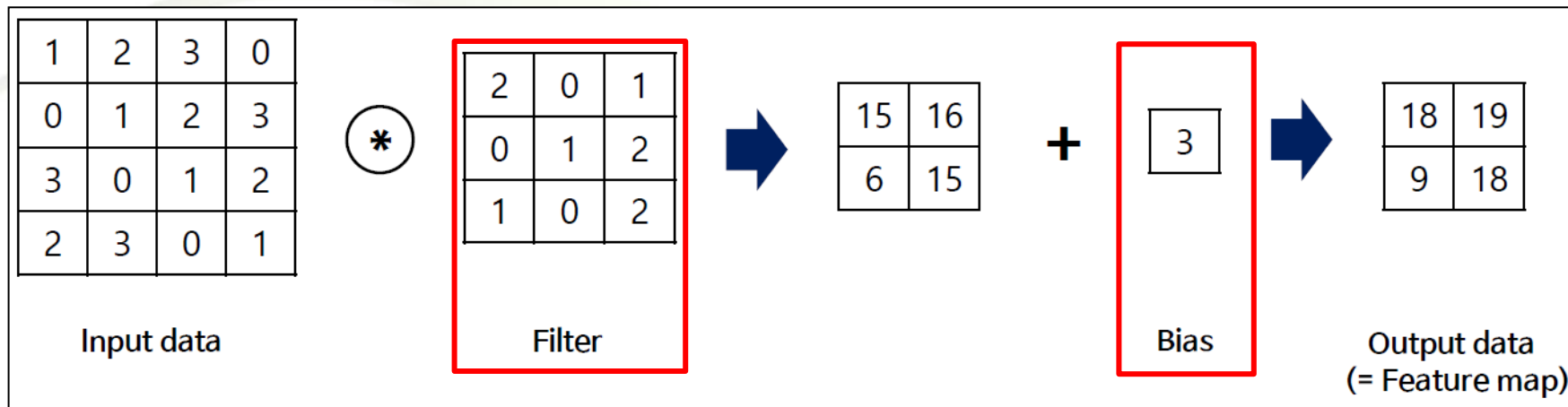


- 연산 사례



합성곱

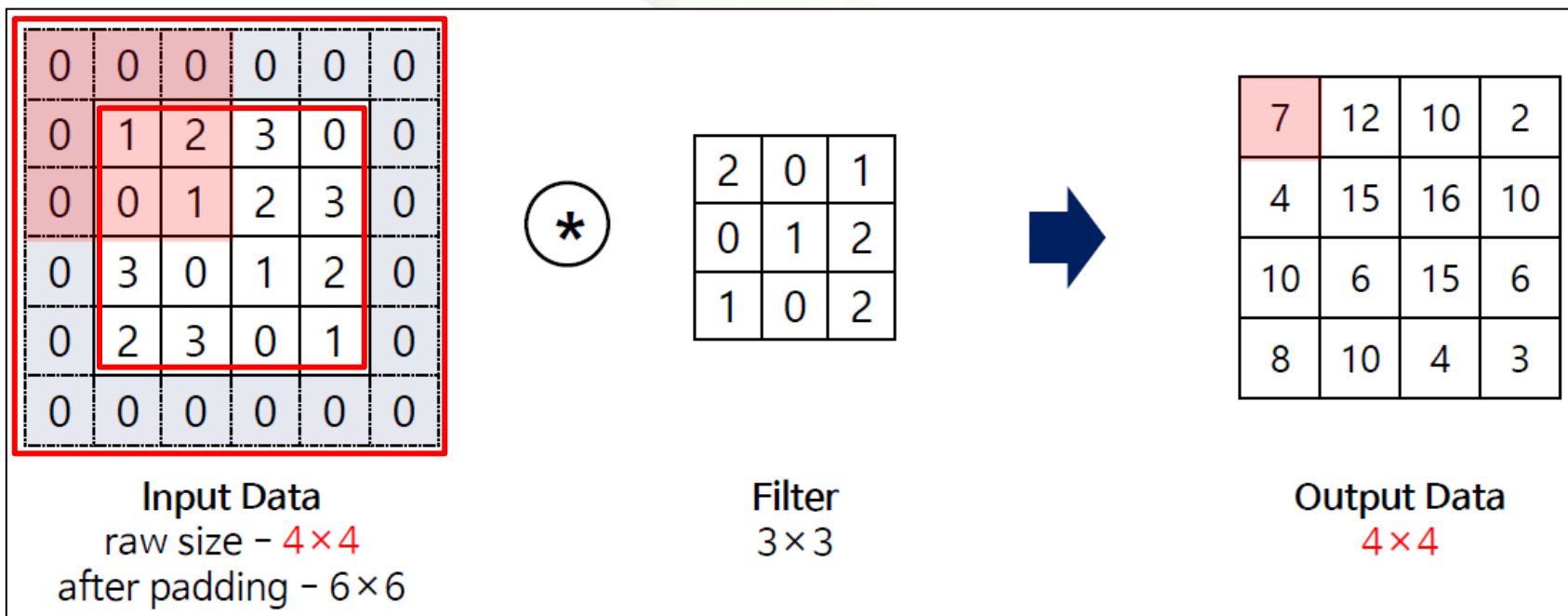
- 편향을 적용한 합성곱 연산



» 필터(Filter)와 편향(Bias)가 학습 대상 매개변수

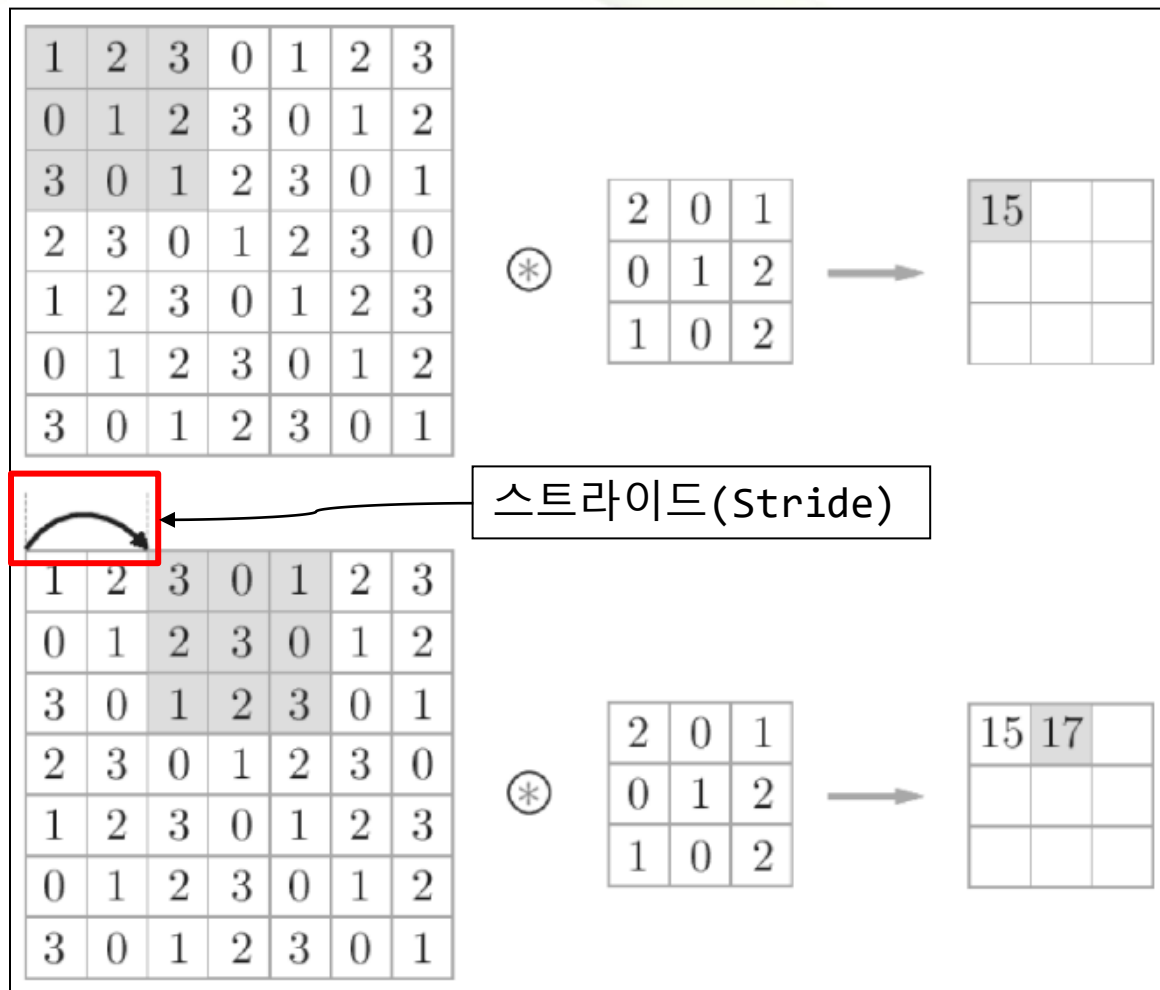
패딩

- 입력 데이터 주변에 채워진 값
 - 주로 0을 사용 → Zero padding
 - 일반적으로 입력 데이터와 출력 데이터의 크기를 맞추기 위해 사용



스트라이드

- 다음 번 필터 적용을 위해 이동하는 간격
 - » 스트라이드가 커지면 출력 크기가 작아진다



출력 크기 계산

$$OH = \frac{H + 2P - FH}{S} + 1$$

$$OW = \frac{W + 2P - FW}{S} + 1$$

입력 크기 - (H, W)

필터 크기 - (FH, FW)

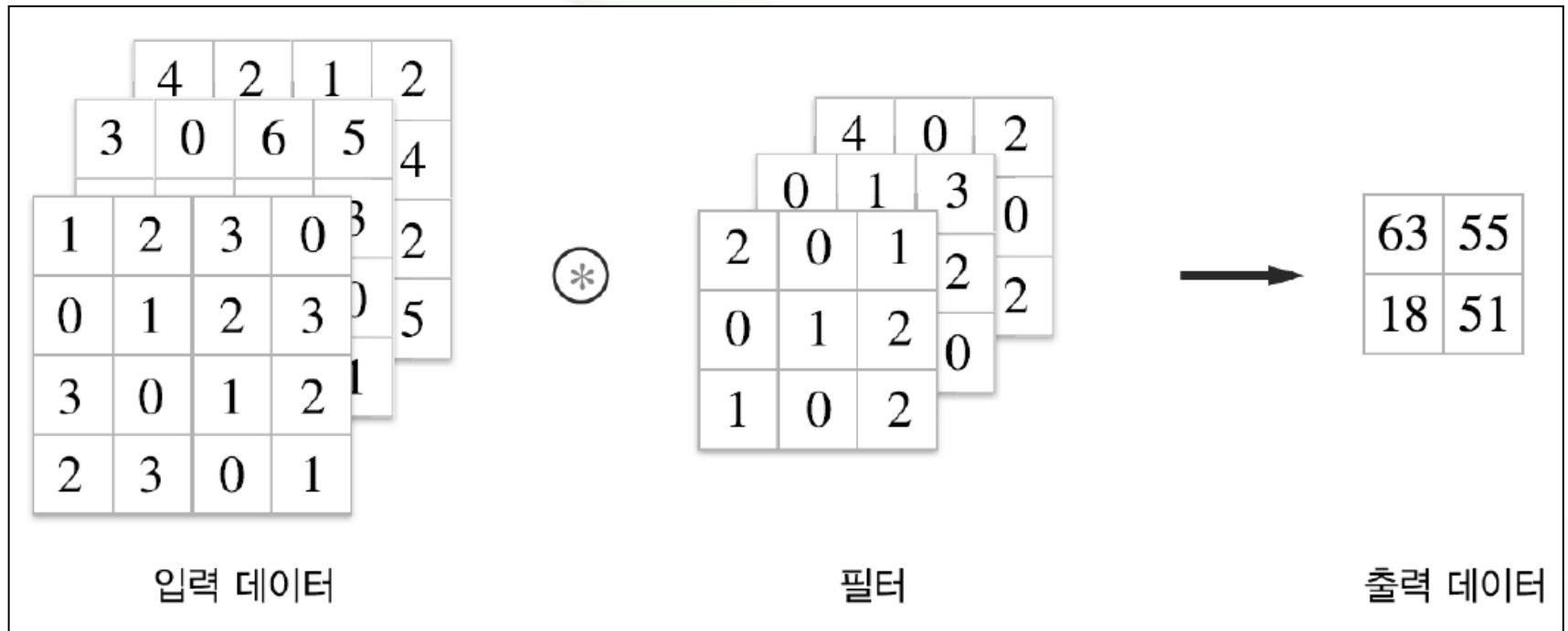
출력 크기 - (OH, OW)

패딩 - P

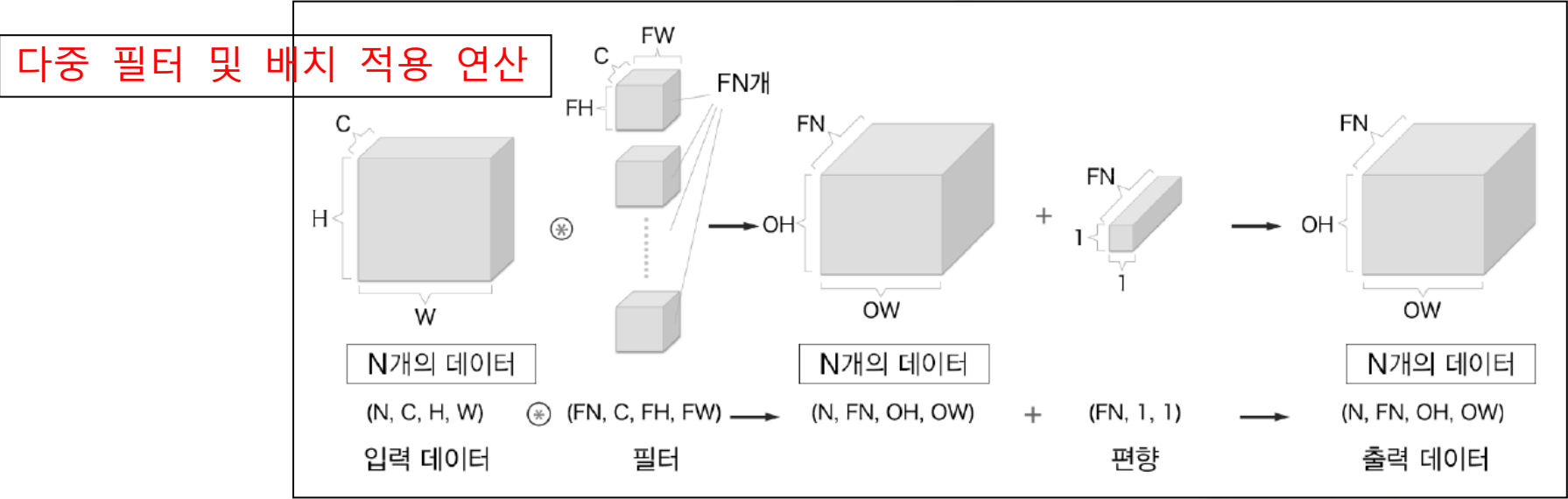
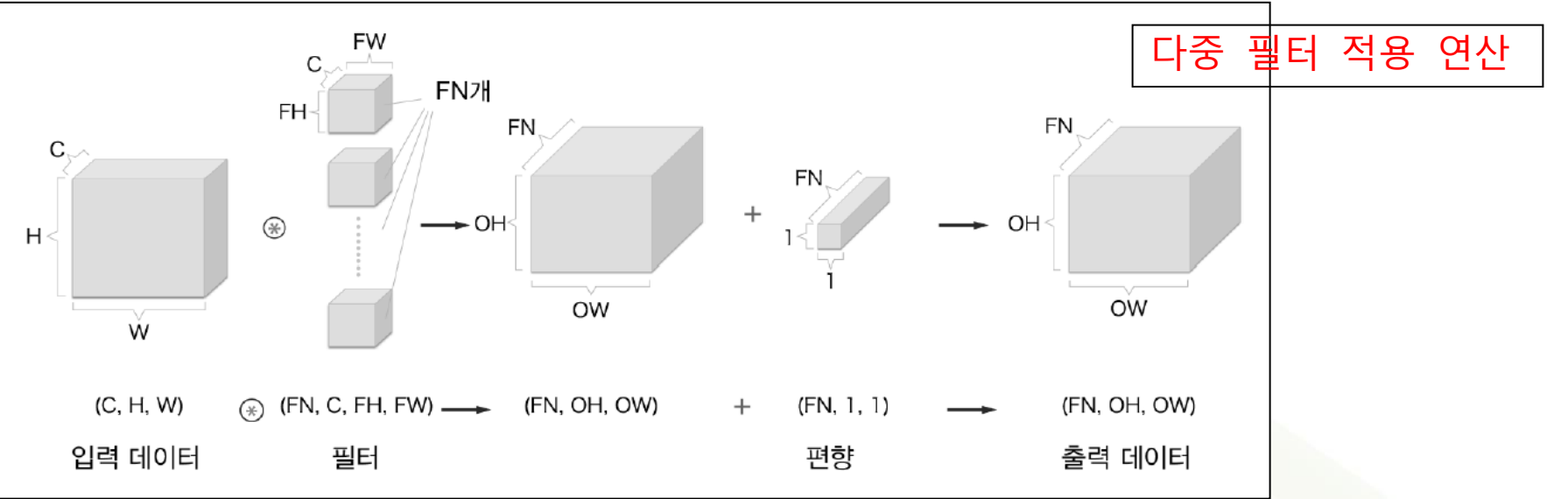
스트라이드 - S

3차원 데이터 합성곱 연산

- 3차원 데이터는 채널을 고려해서 필터를 만들고 연산
 - 필터의 채널의 개수와 이미지의 채널의 개수가 같도록 만들어서 연산



합성곱 연산



풀링(Pooling)

- 세로 및 가로 방향의 공간을 줄이는 연산 (Sub-sampling)
- 특징
 - » 주로 풀링의 윈도우 크기와 스트라이드 값은 동일하게 설정

풀링 윈도우 - (2×2)

스트라이드 - 2

12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

max pooling

20	30
112	37

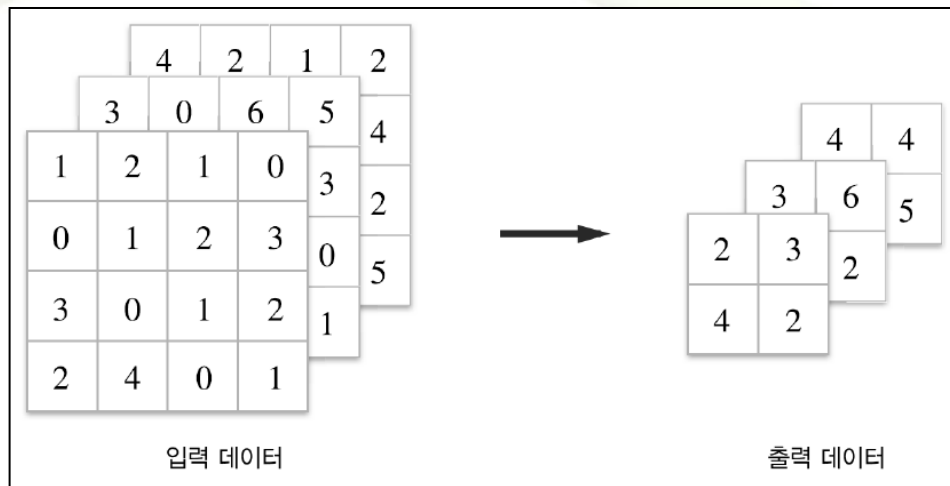
average pooling

13	8
79	20

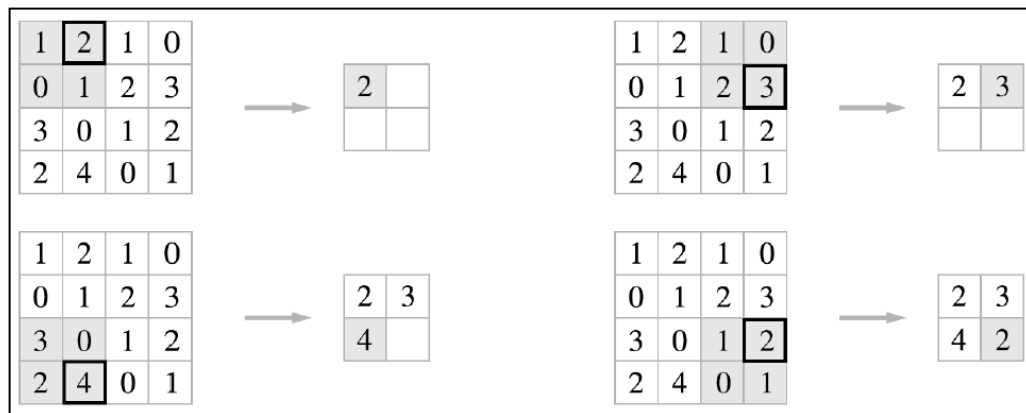
풀링(Pooling)

■ 특징 계속

» 채널마다 독립적으로 계산 → 입력 데이터 채널 수와 출력 데이터의 채널 수 동일



» 연산 과정에 학습할 대상 매개변수 없음



풀링(Pooling)

- 특징 (계속)

- » 입력 데이터의 변화에 크게 영향 받지 않음

