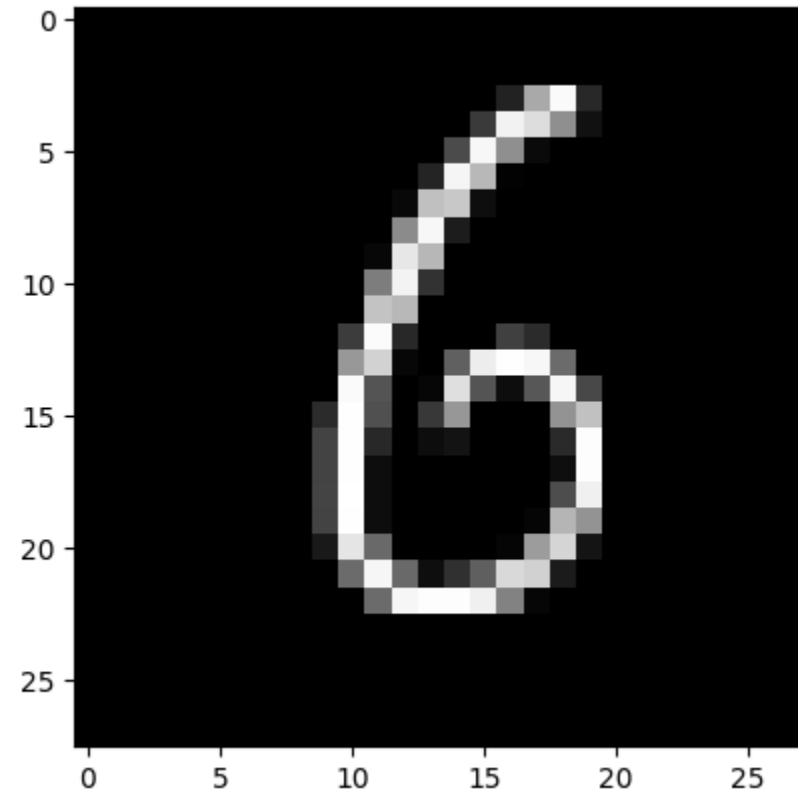
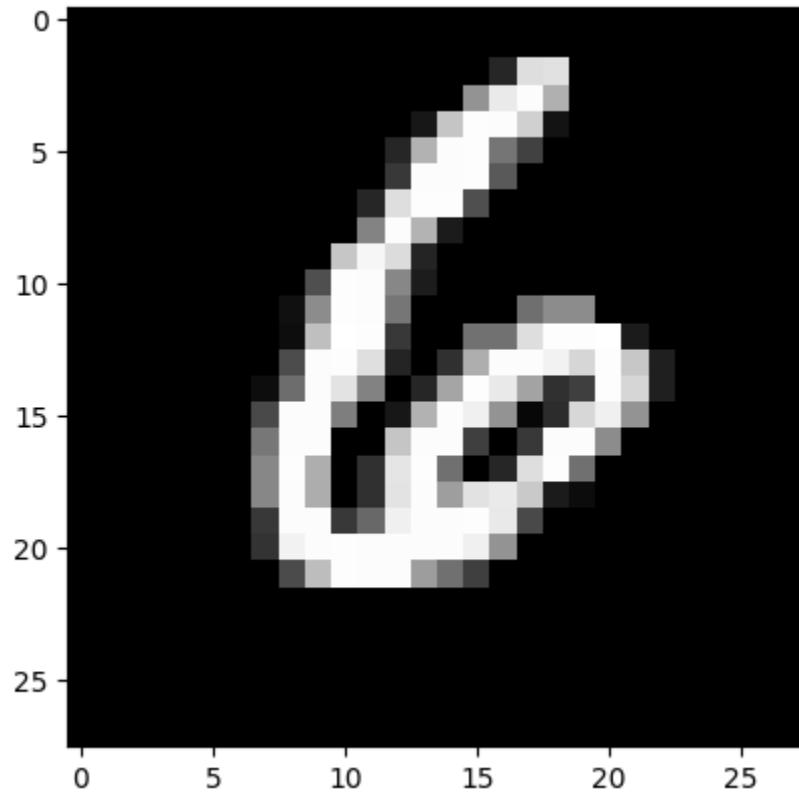
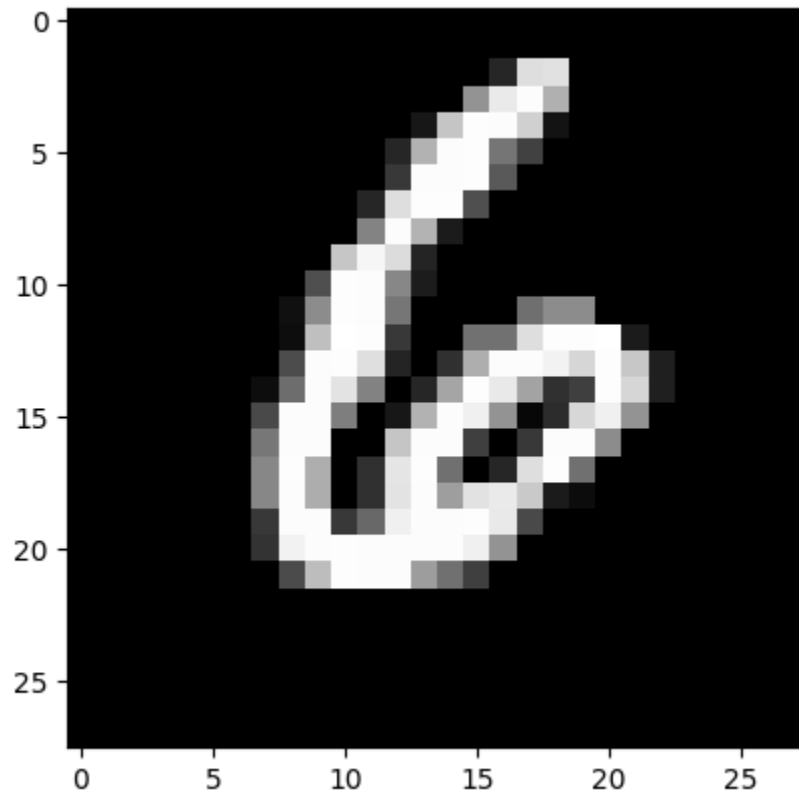


# 4주차 발표자료

이지환



위와 같은 두 이미지의 경우,  
사람은 두 이미지 모두 6이라는 숫자라고 파악할 수 있지만  
기계는 서로 다른 입력으로 인식하며, 특히 다층 퍼셉트론은 적은 수의 값이  
달라져도 분류/예측에 큰 영향을 미침.



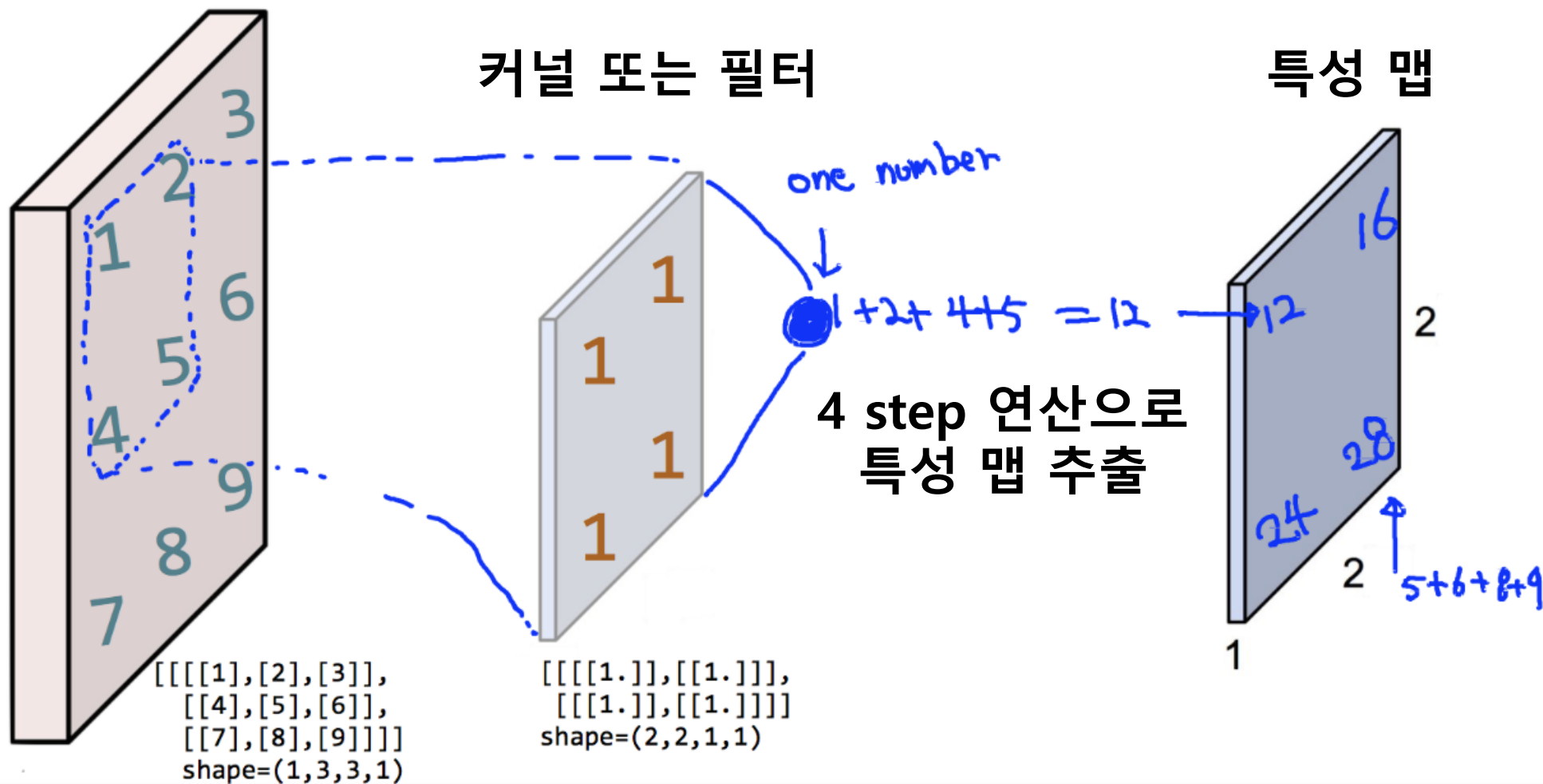
만약 이를 완전연결 계층인 다층 퍼셉트론으로 분류한다면 이미지를 1차원 텐서(벡터)로 변환하여 입력층으로 사용함.  
그런데, 1차원으로 변환하면 데이터가 어떠한 구조로 구성되어 있는지에 대한 공간적 정보가 사라지기 때문에 이러한 정보를 보존하며 학습하는 방법이 필요해짐 -> **합성곱 신경망 탄생 배경**

합성곱은 어떻게 공간적 정보를 보존하는가?

-> 특성 맵(feature map)

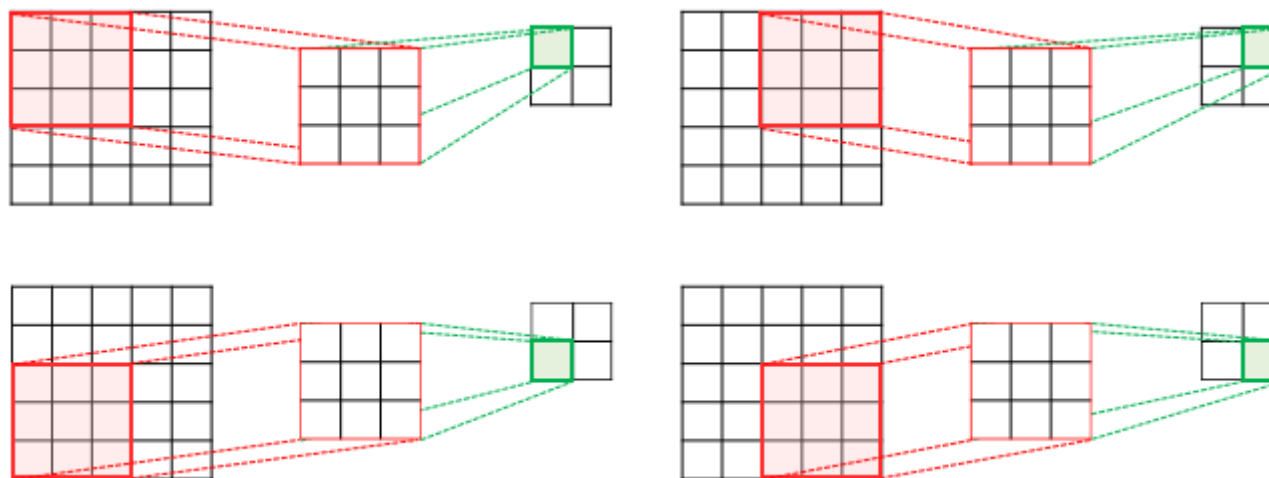
## Simple convolution layer

Image: 1,3,3,1 image, Filter: 2,2,1,1, Stride: 1x1, Padding: VALID



## 스트라이드(Stride)

- 연산 내 step마다 커널의 이동 범위
- 이전 슬라이드의 스트라이드는 1이었으며,  
아래 그림은 스트라이드가 2일 경우 5 x 5 이미지에 3 x 3 커널로 합성곱 연산하는 과정



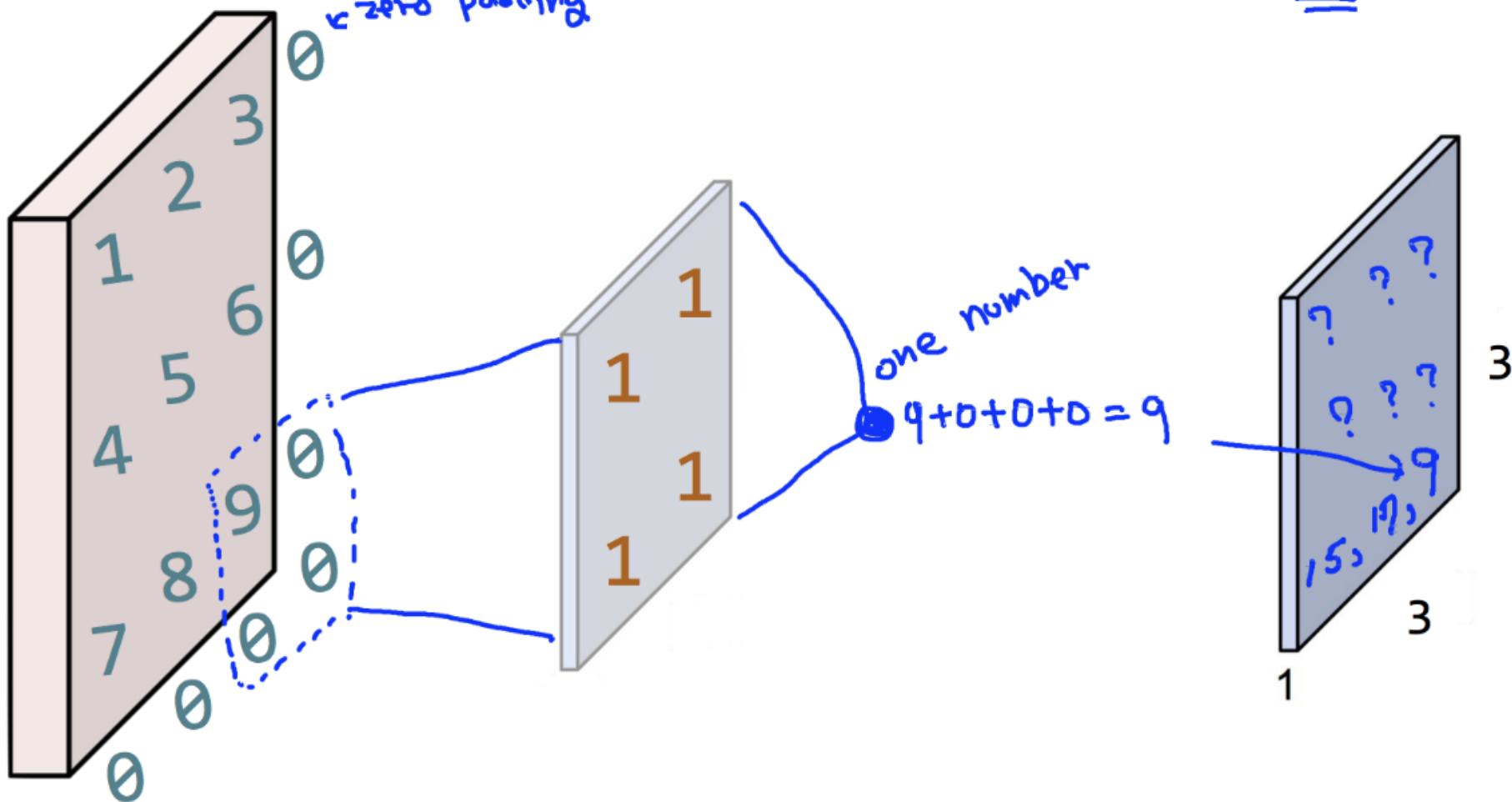
## 패딩(Padding)

- 합성곱 연산 전 입력 가장자리에 데이터를 추가해주며, 주로 0을 추가하는 **zero padding**을 사용
- 합성곱 연산 이후 특성 맵의 크기가 입력 크기에 비해 작아지는 현상을 방지 가능

## Simple convolution layer

Image: 1,3,3,1 image, Filter: 2,2,1,1, Stride: 1x1, Padding: SAME

*want 3x3*



입력, 커널 크기 및 스트라이드에 따른 특성 맵의 높이 및 너비

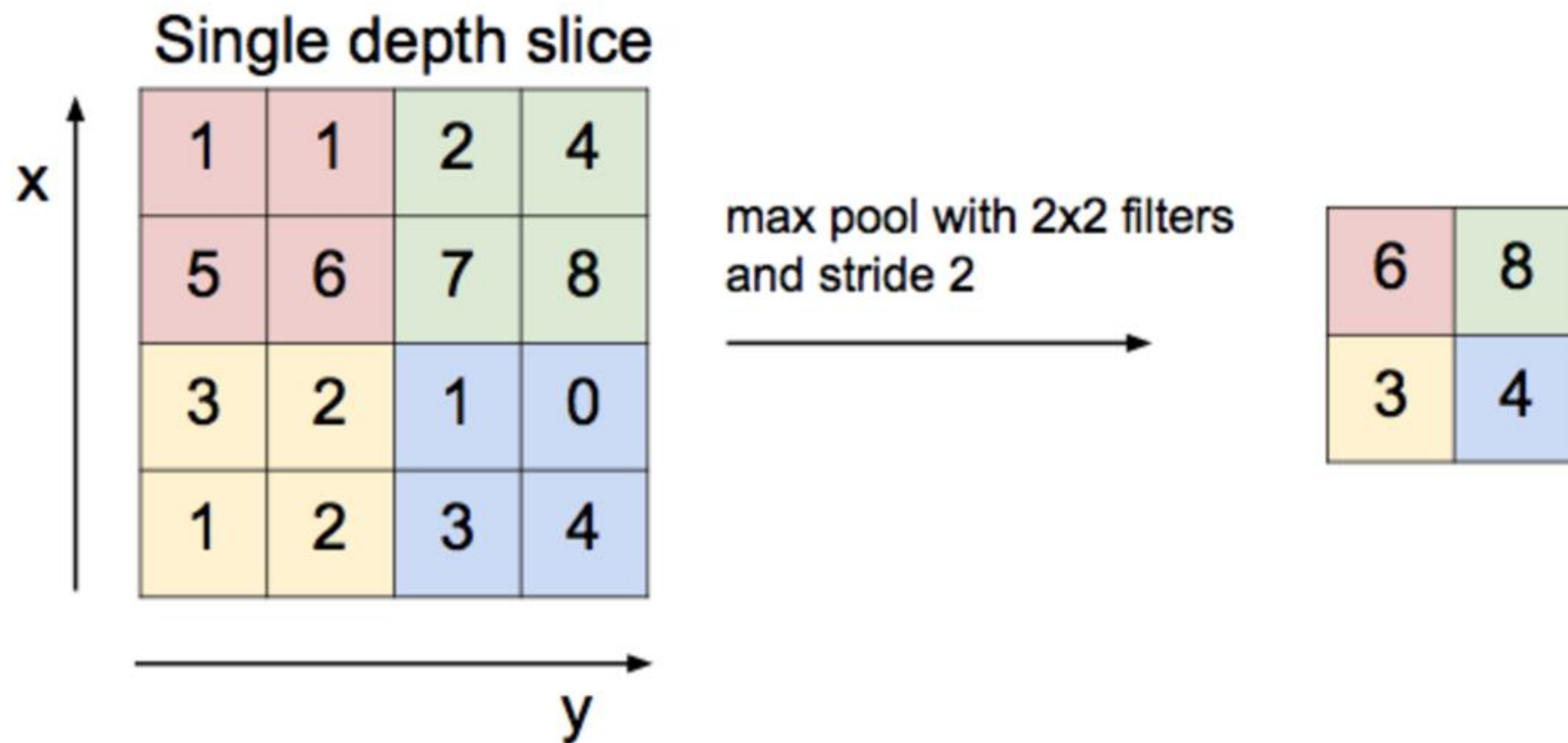
- $I_{h,w}$  : 입력의 높이 및 너비
- $K_{h,w}$  : 커널의 높이 및 너비
- $S$  : 스트라이드
- $O_{h,w}$  : 특성 맵의 높이 및 너비

$$O_{h,w} = \text{floor}\left(\frac{I_{h,w} - K_{h,w} + 2P}{S} + 1\right)$$

## 풀링(Pooling)

- 만들어진 특징 맵으로 집계 함수(평균 또는 최댓값)를 사용하여 데이터 크기를 축소하는 방법
- 아래 그림은 최댓값을 집계 함수로 채택한 Pooling 방법(MAX Pooling)을 시각화

## MAX POOLING





## 데이터 증강 (Data Augmentation)

- 갖고 있는 데이터 셋의 규모를 증강시킴으로써 오버피팅을 예방하는 방법
- 이미지 데이터 증강의 경우, Flipping(뒤집기), Gray Scale, Rotation(회전) 등의 방법이 있음

### Rotation(회전)의 예시

