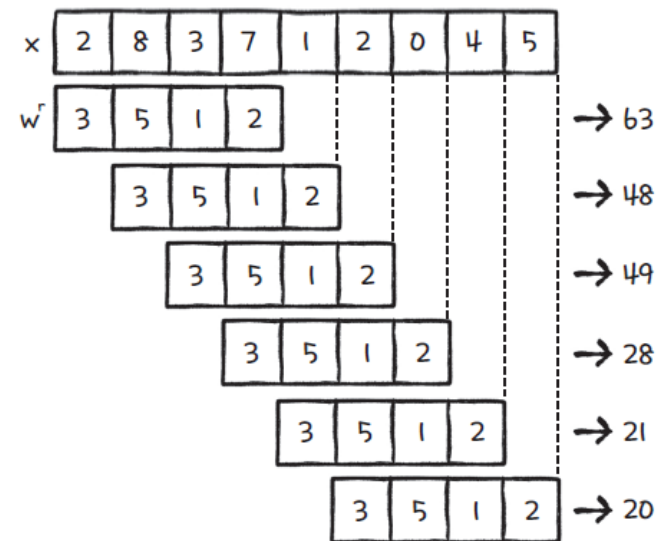
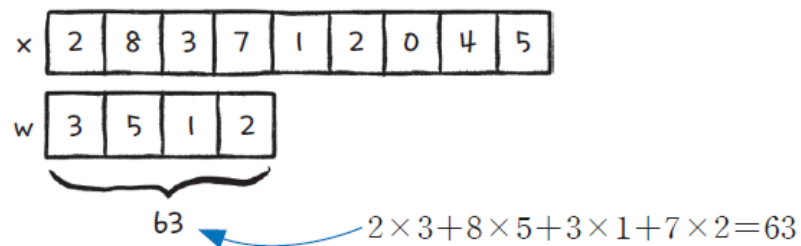
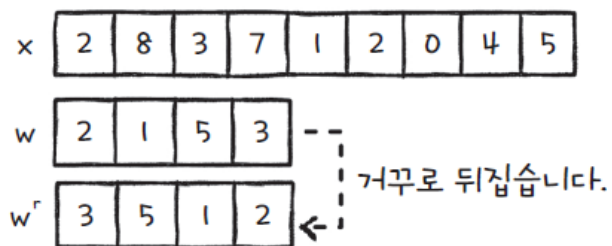


# 08장 이미지를 분류합니다 — 합성곱 신경망

## 08-1 합성곱 연산에 대해 알아봅시다

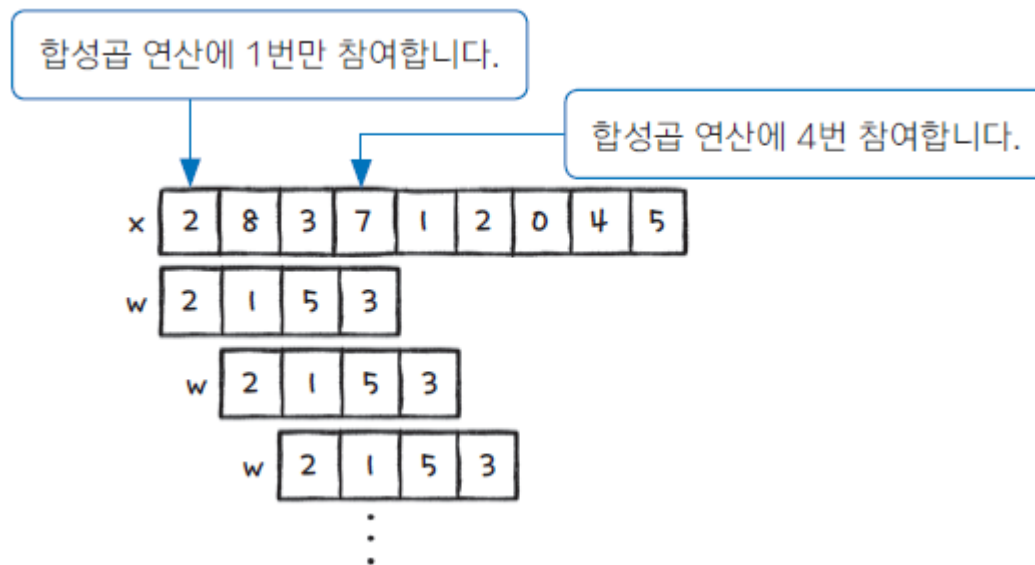
### 합성곱

곱하는 배열을 뒤집은 다음 원본 배열에 미끄러지듯 점 곱을 수행하여 더함



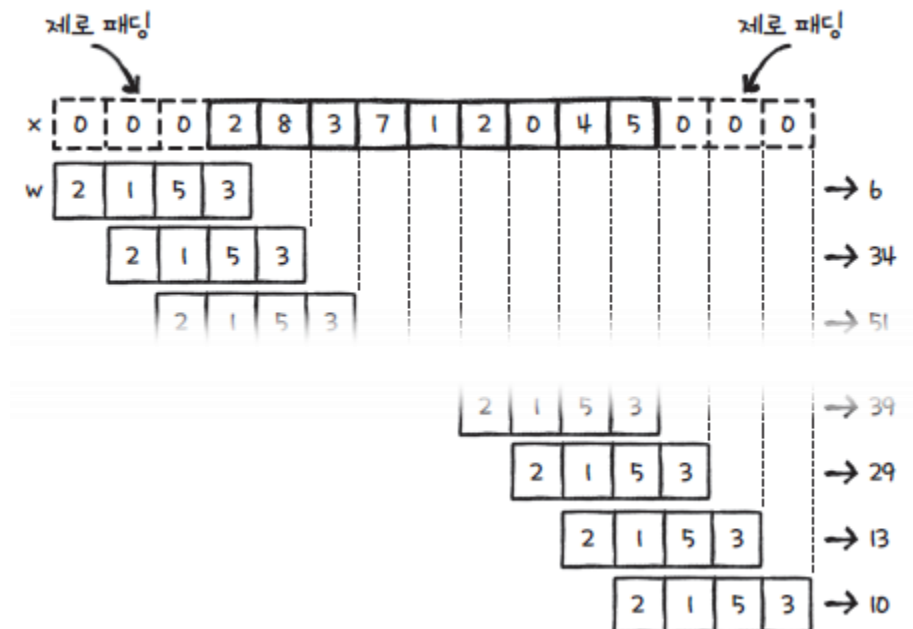
## 08-1 합성곱 연산에 대해 알아봅시다

원본 배열이 합성곱 연산에 참여하는 정도?



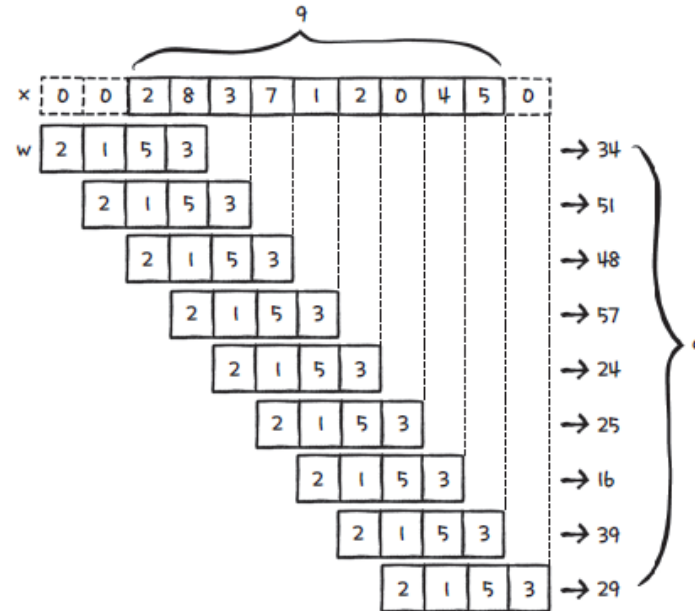
## 08-1 합성곱 연산에 대해 알아봅시다

풀 패딩(full padding)  
모든 원본 배열의 원소가 연산에 참여



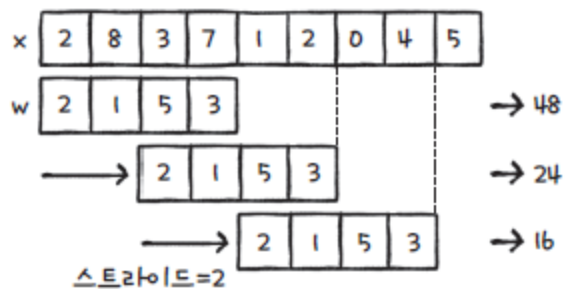
## 08-1 합성곱 연산에 대해 알아보니다

세임 패딩(same padding)  
출력 배열의 길이가 원본 배열의 길이와 같아지도록 만듦



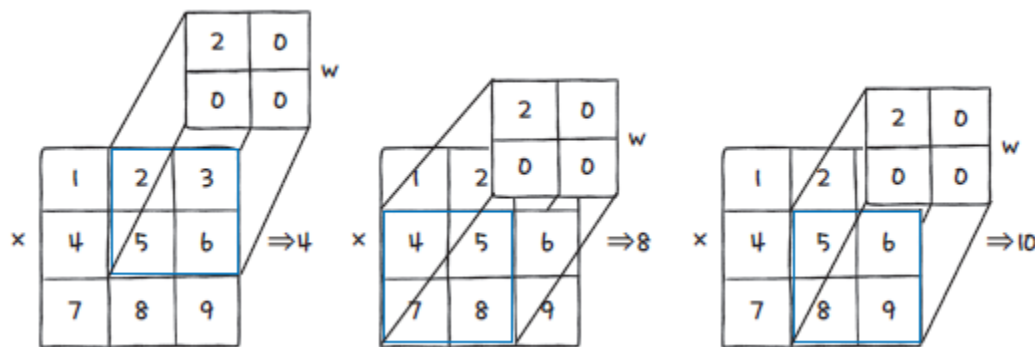
## 08-1 합성곱 연산에 대해 알아보니다

스트라이드(stride)  
곱하는 배열의 미끄러지는 정도를 조정



## 08-1 합성곱 연산에 대해 알아보니다

2차원 배열의 합성곱

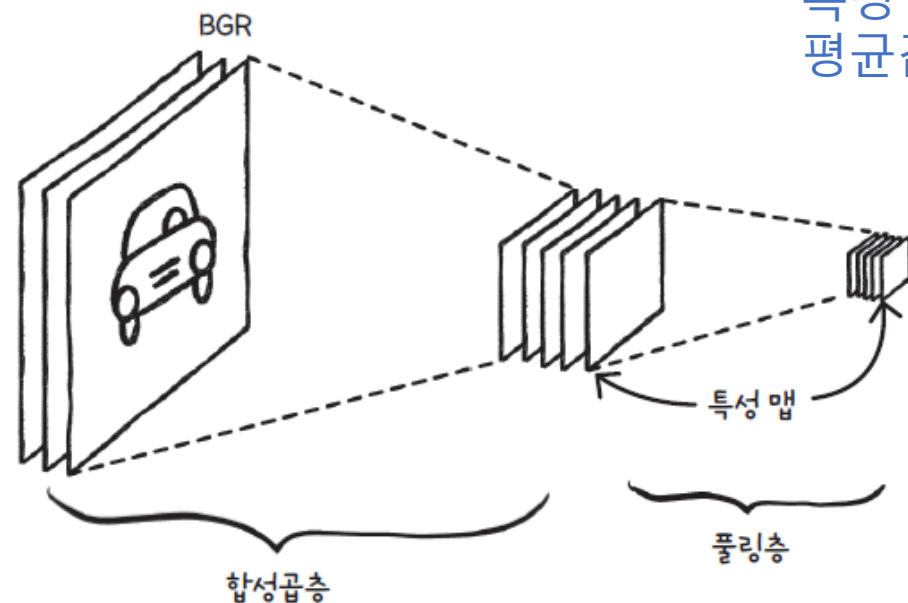


밸리드 패딩, 스트라이드 = 1

## 08-2 풀링 연산에 대해 알아보니다

합성곱 신경망의 전체 구조  
풀링층 추가

[풀링]  
특성맵을 스캔하며 최댓값을 고르거나  
평균값을 계산하는 것



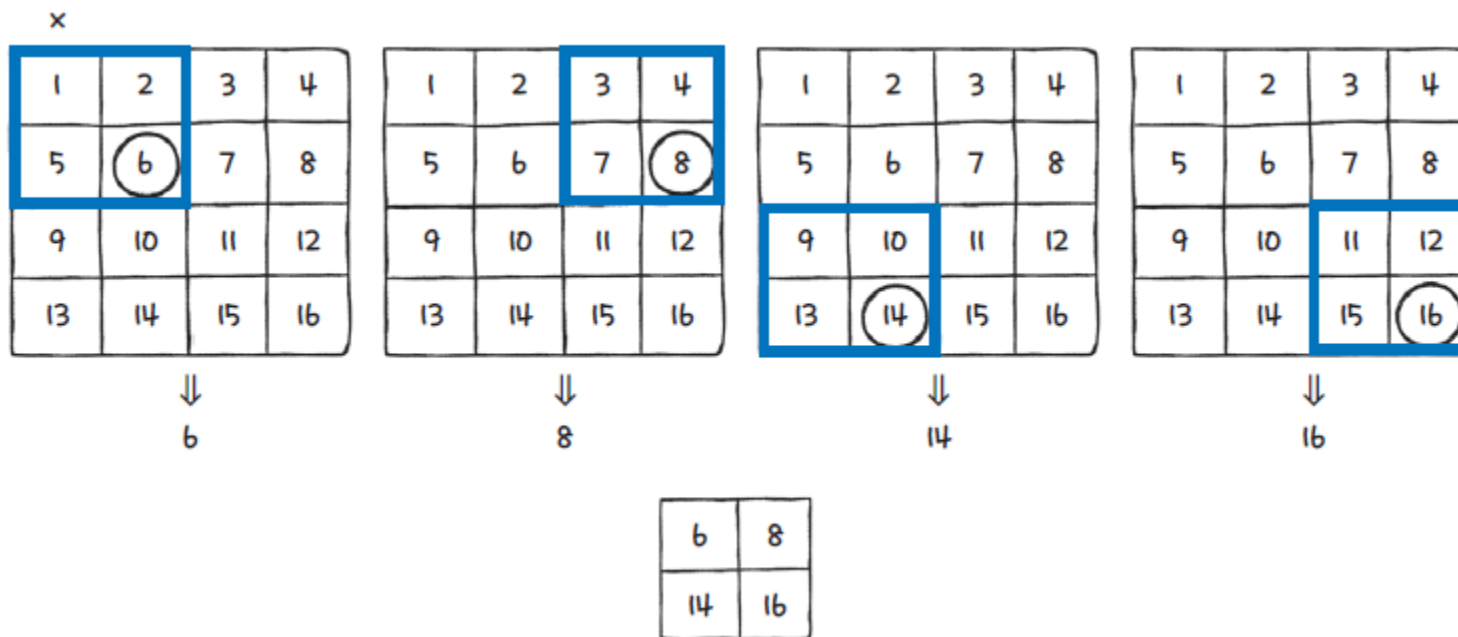
[특성맵]  
합성곱층과 풀링층에서 만들어진 결과



## 08-2 풀링 연산에 대해 알아보니다

### 풀링

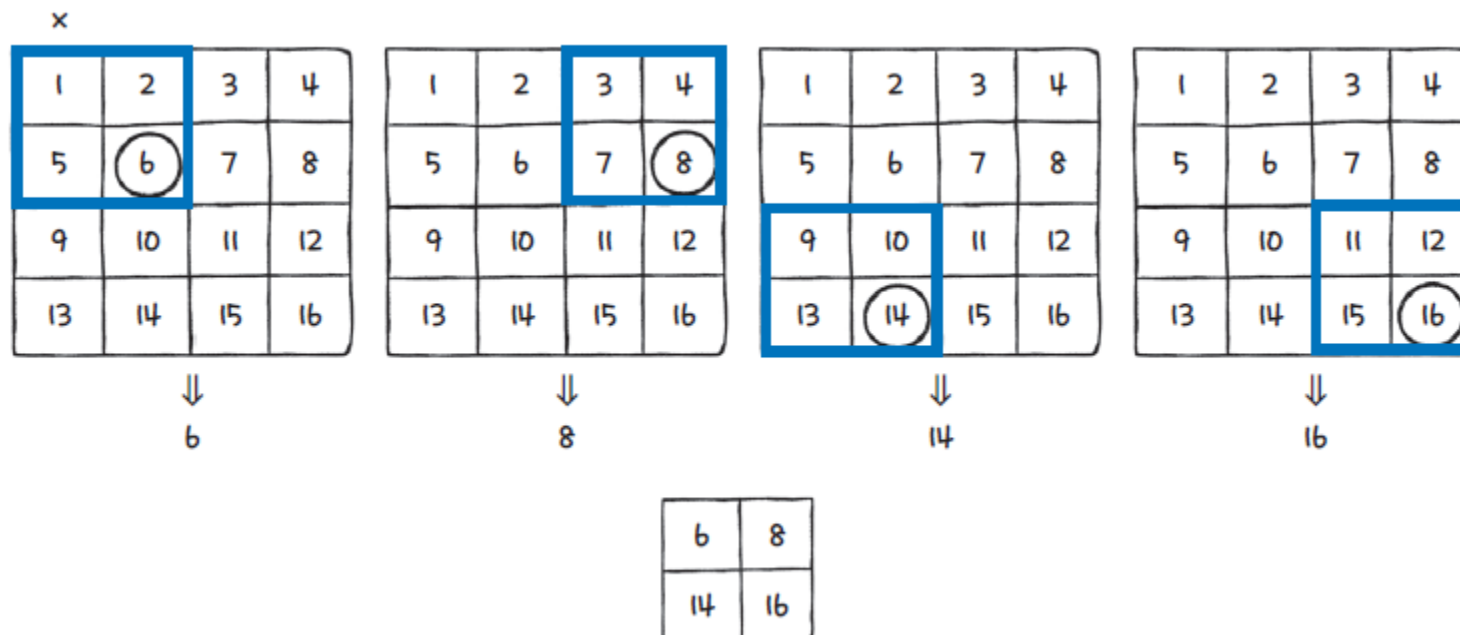
특성 맵을 스캔하며 최댓값 또는 평균값을 계산하는 것



최대 풀링

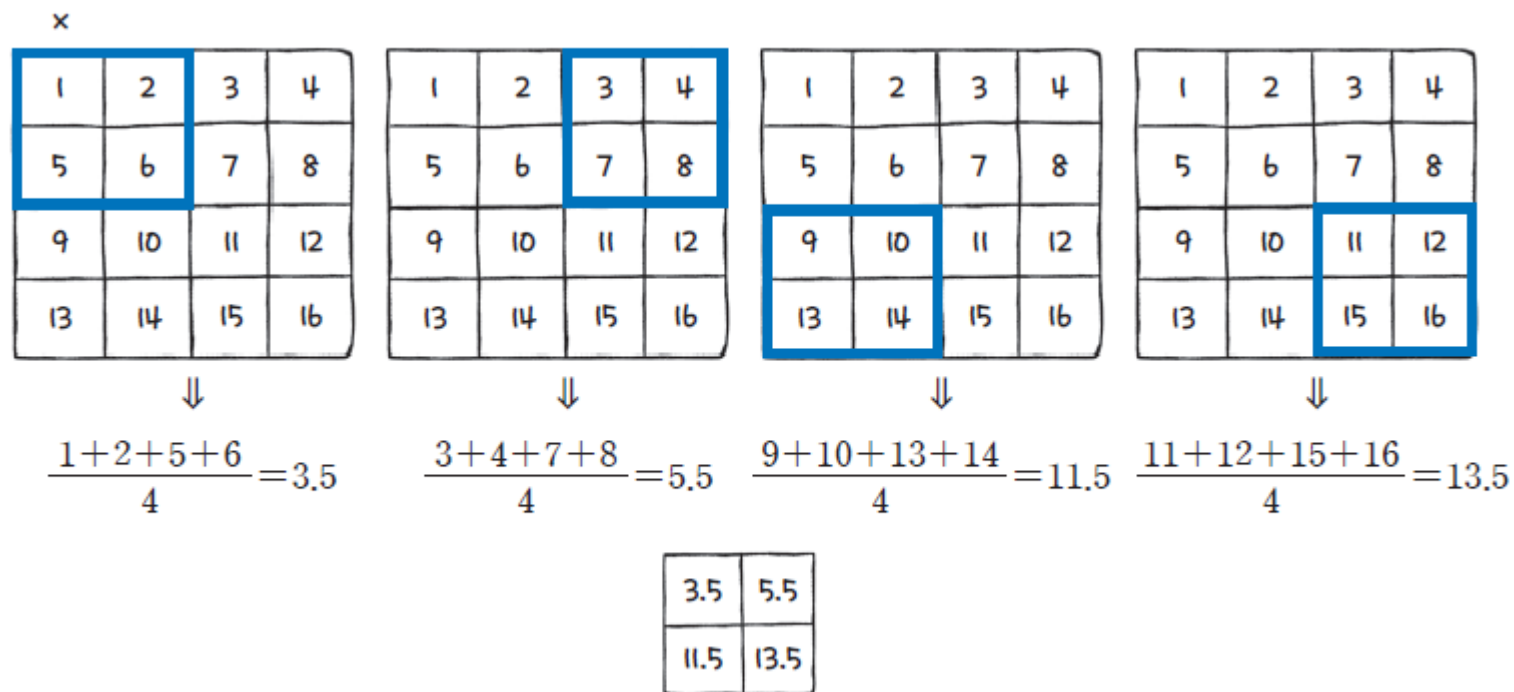
## 08-2 풀링 연산에 대해 알아보니다

최대 풀링(max pooling)  
특성 맵을 스캔하며 최댓값을 고름



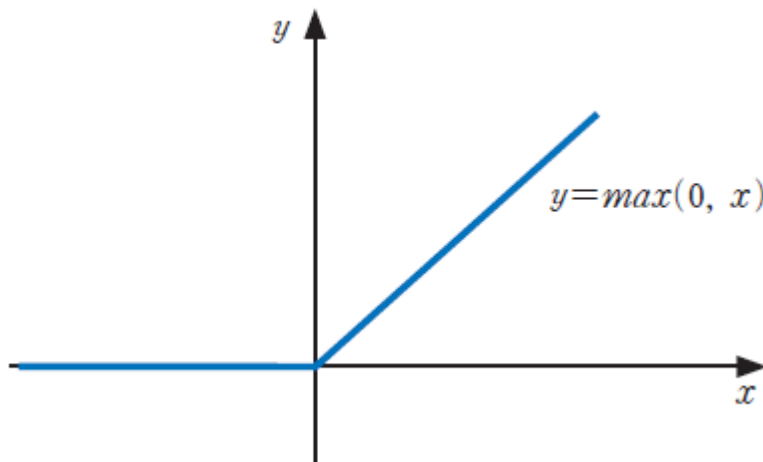
## 08-2 풀링 연산에 대해 알아보니다

평균 풀링(average pooling)  
특성 맵을 스캔하며 평균값을 계산



## 08-3 합성곱 신경망의 구조를 알아보니다

렐루 함수  
합성곱 신경망에 사용할 활성화 함수



$$y = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$

## 08-3 합성곱 신경망의 구조를 알아보니다

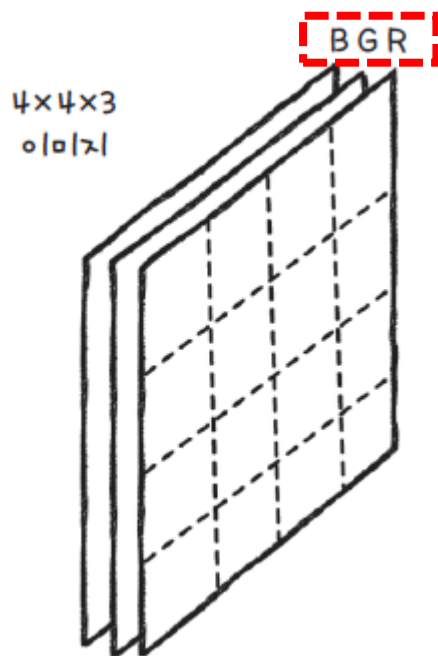
렐루 함수의 도함수

$x=0$ 일 때의 미분은 0으로 생각(그렇게 해도 실전에서 잘 동작함)

$$y = \begin{cases} 1 & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$

## 08-3 합성곱 신경망의 구조를 알아보니다

이미지의 색상을 표현하기 위한 채널  
이미지의 색상은 RGB로 표현

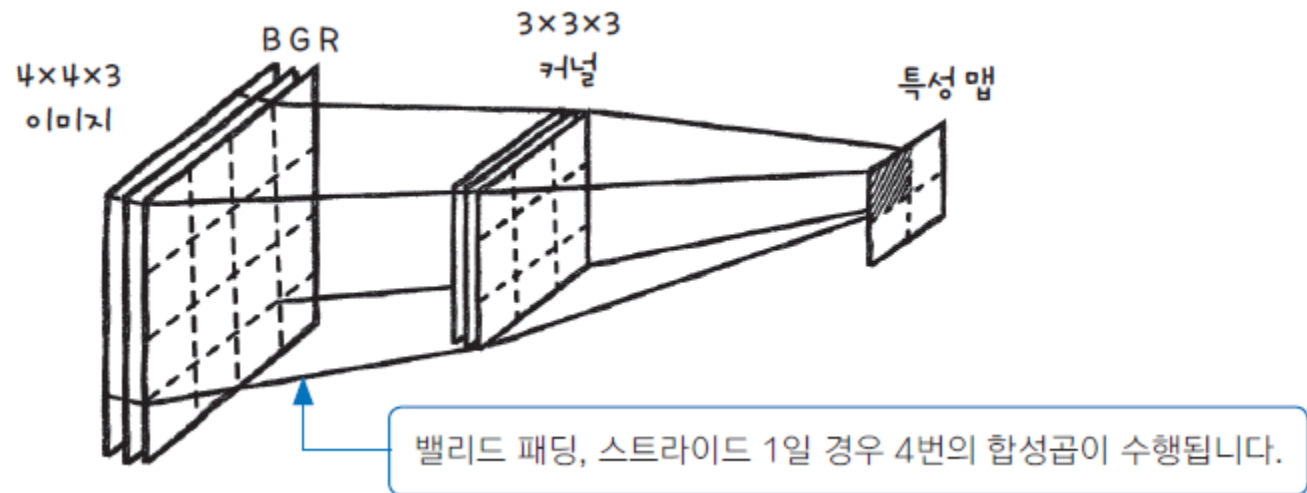


합성곱 신경망은 이미지의 2차원 형태를  
입력으로 그대로 사용

- 이미지를 한 줄로 펼칠 필요가 없음
- 이미지 정보가 손상되지 않음.

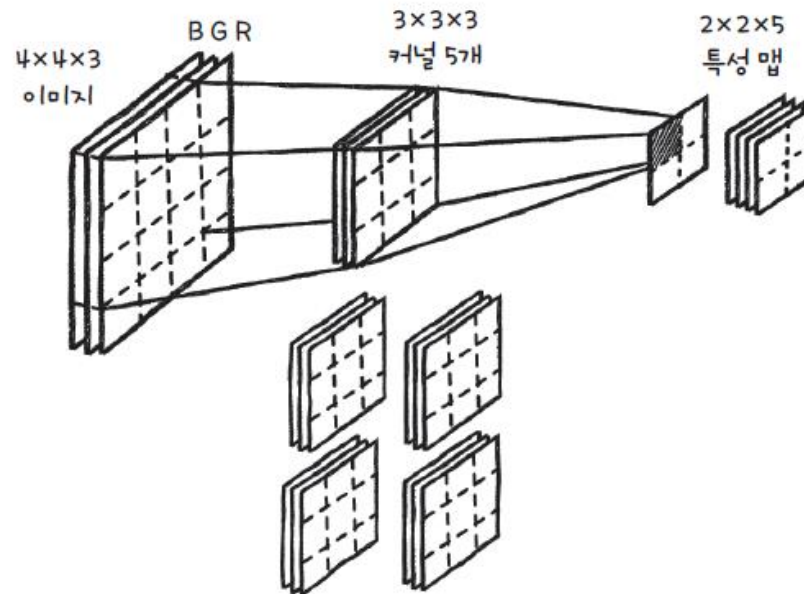
## 08-3 합성곱 신경망의 구조를 알아보니다

합성곱 층의 연산 이해  
첫 번째 합성곱



## 08-3 합성곱 신경망의 구조를 알아봅시다

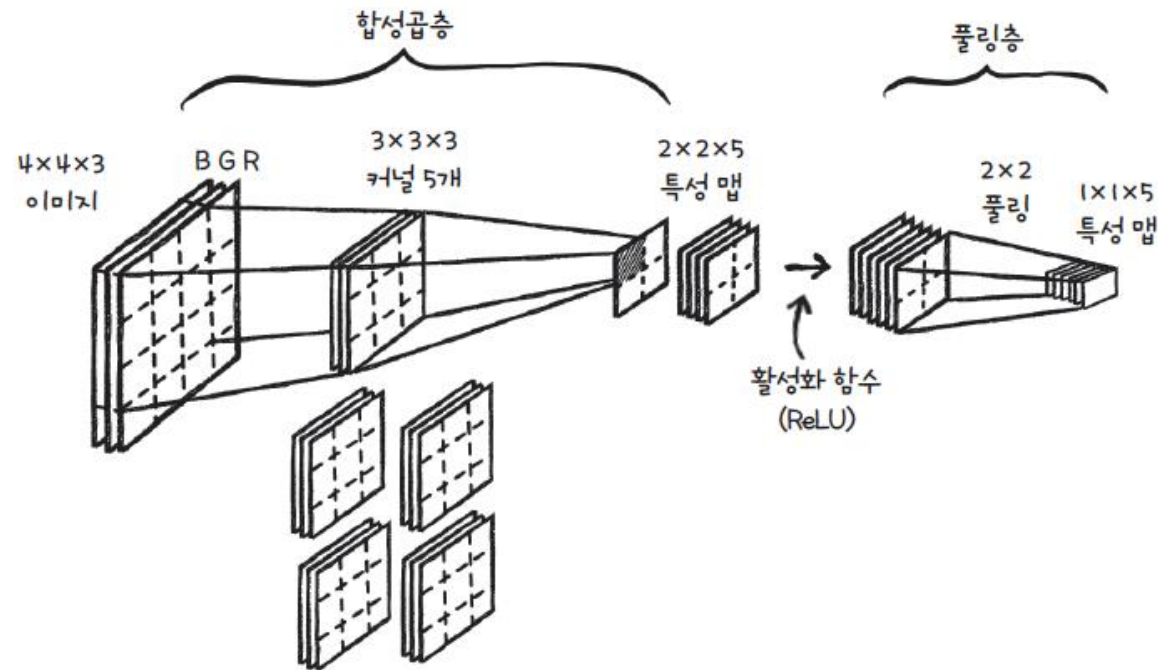
합성곱 층의 연산 이해  
전체 합성곱





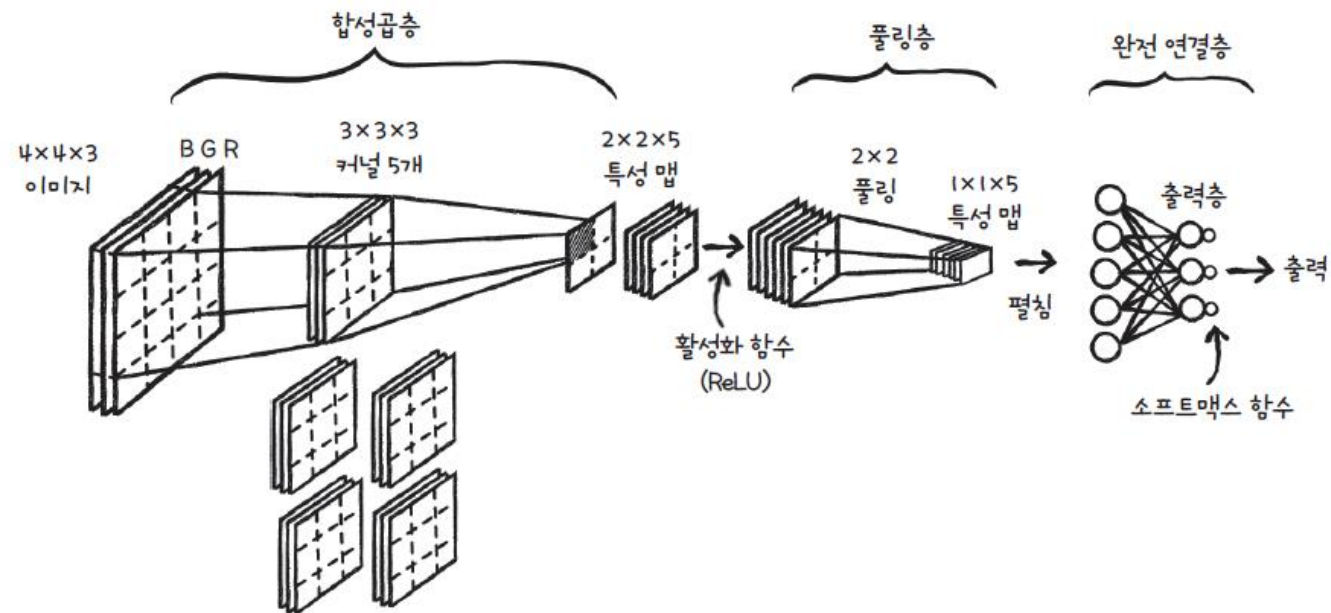
## 08-3 합성곱 신경망의 구조를 알아봅시다

풀링층을 포함한 전체 연산의 이해



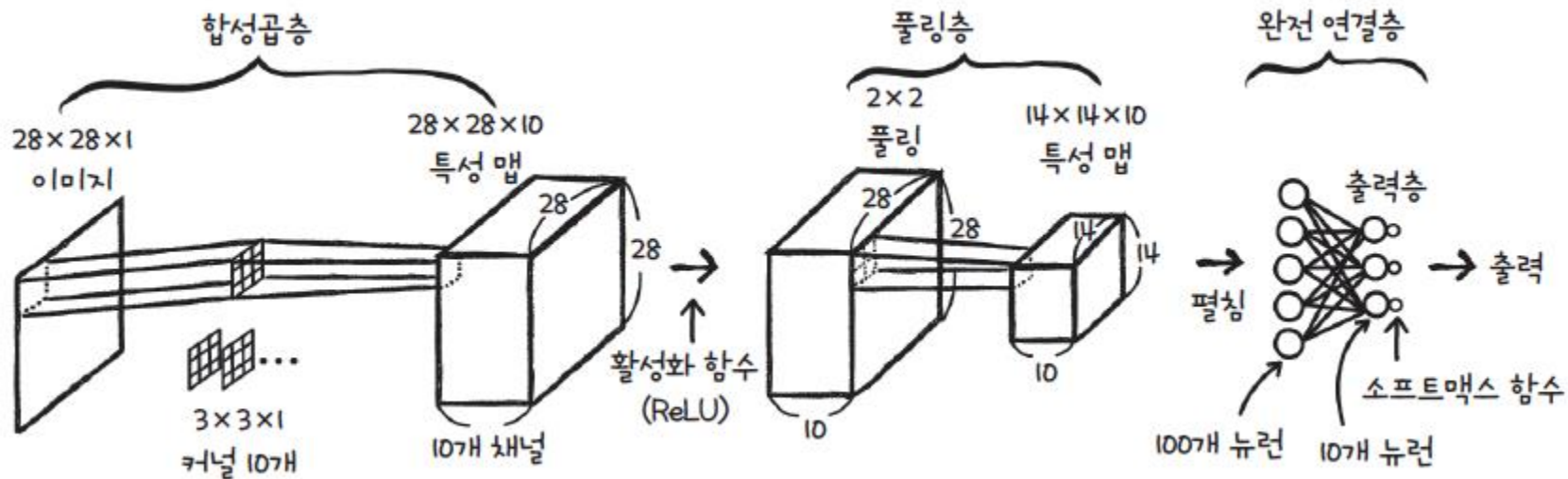
## 08-3 합성곱 신경망의 구조를 알아보니다

풀링층을 거친 특성맵을 펼쳐 완전 연결층에 주입



## 08-4 합성곱 신경망을 만들고 훈련합니다

합성곱 신경망의 전체 구조



## 08-5 케라스로 합성곱 신경망을 만듭니다

### 케라스로 합성곱 신경망 만들기

Conv2D : 케라스의 합성곱층 클래스

MaxPooling2D : 케라스의 최대풀링 클래스

Flatten : 케라스에서 특성맵을 일렬로 펼칠 때 사용

Dense 등 클래스를 임포트한다.

```
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
```

필요한 클래스 임포트

## 08-5 케라스로 합성곱 신경망을 만듭니다

케라스로 합성곱 신경망 만들기  
합성곱층, 풀링층 쌓기 > 특성 맵 펼치기

```
conv1 = tf.keras.Sequential( )  
conv1.add(Conv2D(10, (3, 3), activation='relu', padding='same', input_shape=(28, 28, 1)))
```

합성곱층 쌓기

```
conv1.add(MaxPooling2D((2, 2)))
```

풀링층 쌓기

```
conv1.add(Flatten( ))
```

특성 맵 펼치기

## 08-5 케라스로 합성곱 신경망을 만듭니다

케라스로 합성곱 신경망 만들기  
완전 연결층 쌓고 모델 구조 살펴보기

```
conv1.add(Dense(100, activation='relu'))  
conv1.add(Dense(10, activation='softmax'))
```

완전 연결층 쌓기

```
conv1.summary( )  
Model: "sequential"
```

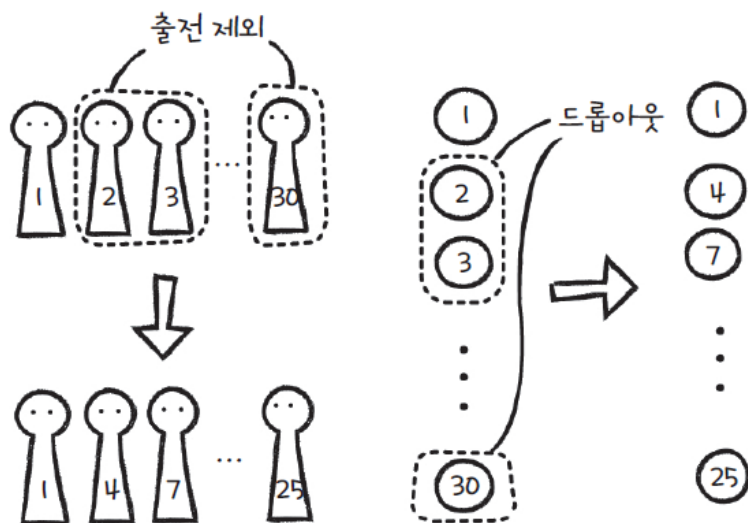
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 10)	100
max_pooling2d (MaxPooling2D)	(None, 14, 14, 10)	0

모델 구조 살펴보기

## 08-5 케라스로 합성곱 신경망을 만듭니다

### 드롭 아웃

무작위로 뉴런을 비활성화 시켜 전체 뉴런이 고르게 훈련할 수 있도록 만듦(과대적합 down)



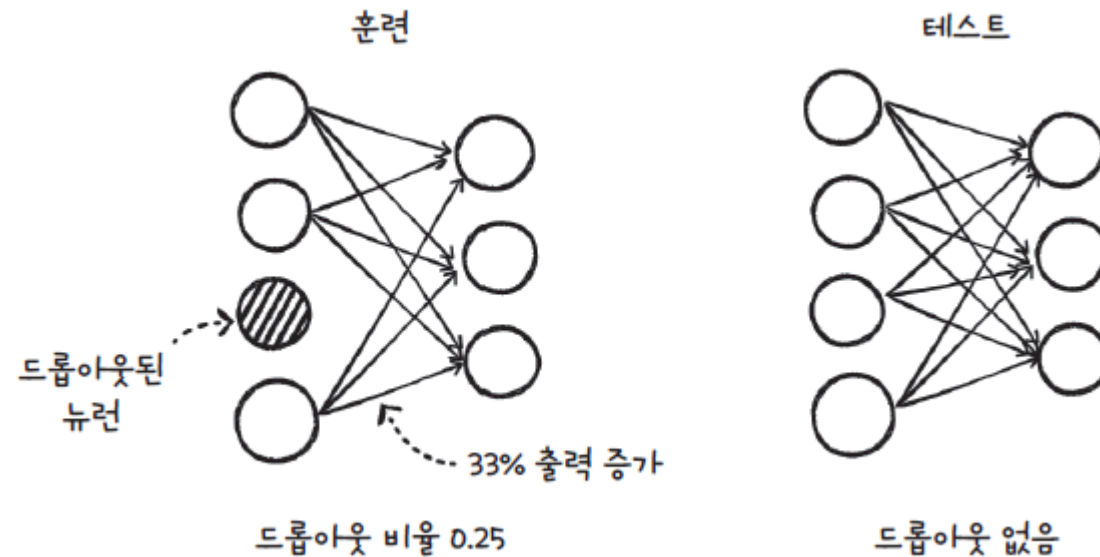
## 08-5 케라스로 합성곱 신경망을 만듭니다

### 텐서플로의 드롭 아웃

드롭아웃 비율만큼 뉴런의 출력을 높임

-드롭아웃층에는 학습되는 가중치가 없음.

-단순히 일부 뉴런의 출력을 무작위로 0으로 만들고, 나머지 뉴런의 출력을 드롭되지 않은 비율로 나누어 증가시킨다.





# 09장 텍스트를 분류합니다 — 순환 신경망

## 09-1 순차 데이터와 순환 신경망을 배웁니다

### 순차 데이터

샘플이 서로 연관되어 있는 경우(순서가 있는 경우) 순차 데이터로 분류

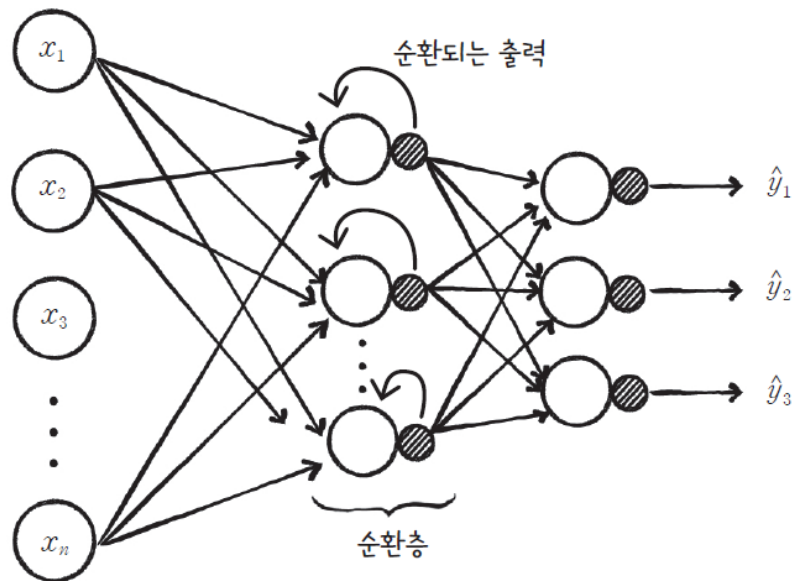
3시    4시    5시  
... 20°C   23°C   22°C ...    Hello Deep Learning

시계열 데이터(시간)

문장 데이터(순서)

## 09-1 순차 데이터와 순환 신경망을 배웁니다

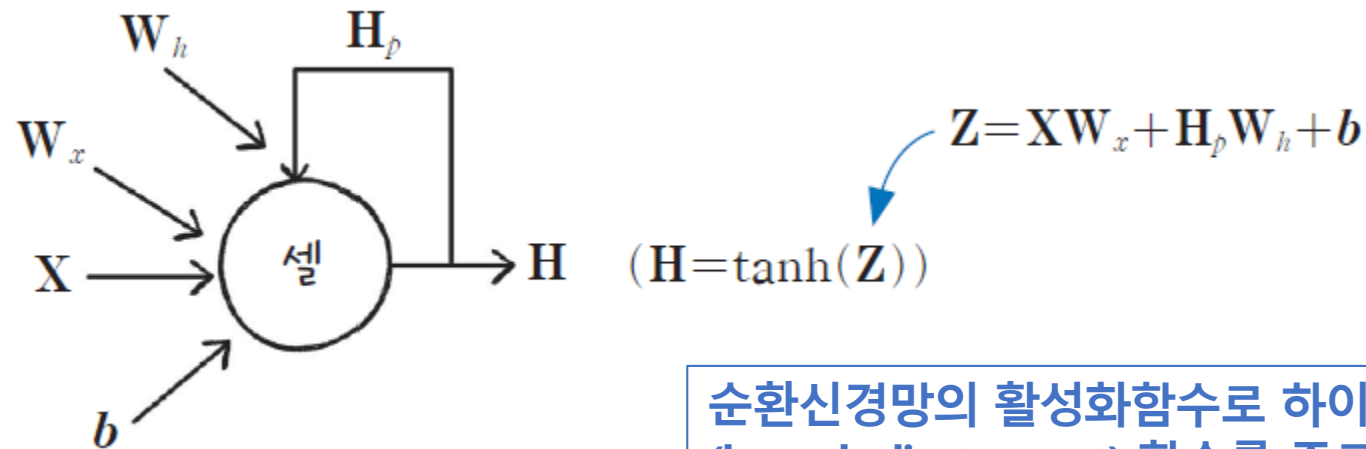
순환 신경망  
뉴런의 출력을 순환시키는 구조를 가진 신경망(순차 데이터를 잘 처리)



## 09-1 순차 데이터와 순환 신경망을 배웁니다

### 셀 = 순환 신경망의 뉴런

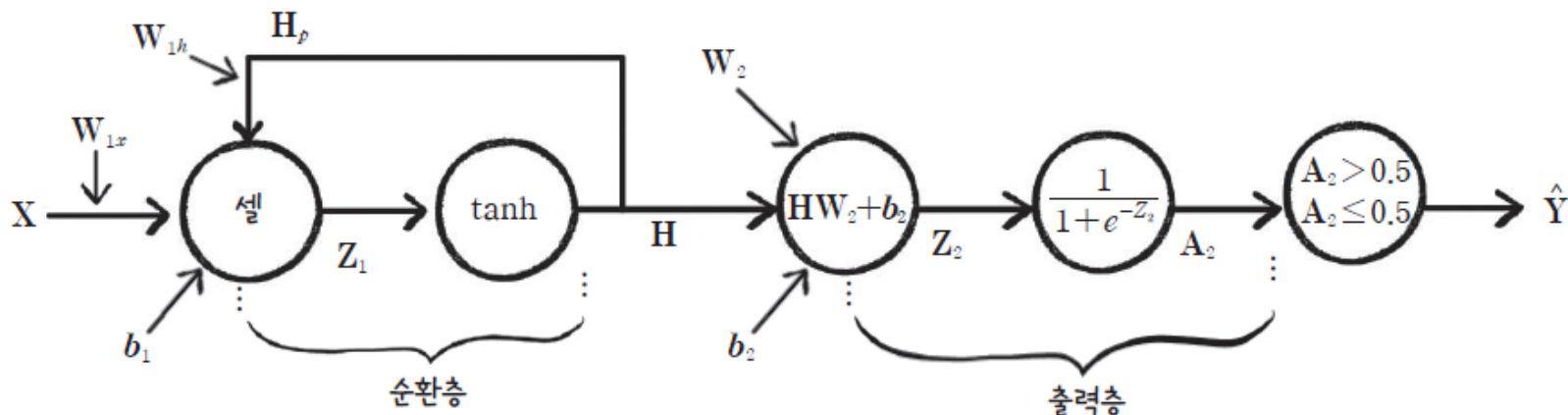
- 입력  $X$ 에 곱해지는 가중치  $W_x$
- 이전 타임 스텝의 은닉 상태  $H_p$ 에 곱해지는 가중치  $W_h$
- 그리고 절편  $b$ 를 함께 표시



순환신경망의 활성화함수로 하이퍼볼릭탄젠트 (hyperbolic tangent) 함수를 주로 사용함

## 09-1 순차 데이터와 순환 신경망을 배웁니다

순환 신경망의 정방향 계산



순환층의 정방향 계산	출력층의 정방향 계산
$Z_1 = XW_{1x} + H_pW_{1h} + b_1$ $H = \tanh(Z_1)$	$Z_2 = HW_2 + b_2$ $A_2 = \text{sigmoid}(Z_2)$