

심층학습

05 합성곱 신경망

SW융합학부 양희경

주교재: 오렐리앙 제롱, 핸드온 머신러닝(사이킷런과 텐서플로를 활용한 머신러닝, 딥러닝 실무), 한빛미디어, 2018.04

학기 내용

1. 심층학습 소개 Deep learning
2. 신경망 Neural network
3. 역전파 Backpropagation
4. 심층 신경망 훈련
- 5. 합성곱 신경망 Convolutional neural network(CNN)**
6. 오토인코더 Auto encoder(AE)
7. 적대적 생성 네트워크 Generative adversarial network(GAN)

내용

5.1 시각 피질의 구조

5.2 합성곱층

5.3 풀링층

5.4 CNN 기본 구조

5.5 CNN 구조

5.1 시각 피질의 구조

- 허블Hubel 과 비셀Wiesel 의 시각 체계 연구(고양이 실험)
 - 뉴런들은 일부 시야 범위 내의 시각 자극에만 반응
(국부 수용장을local receptive field 가짐)
 - 단순한 형태에 자극 받는 뉴런이 있는 반면, 어떤 뉴런은 저수준 패턴이 조합된 더 복잡한 패턴에 반응
→ 고수준 뉴런이 이웃한 저수준 뉴런의 출력에 기반하는 구나!



Hubel, David H. "Single unit activity in striate cortex of unrestrained cats." *The Journal of physiology* 147.2 (1959): 226-238.

Hubel, David H., and Torsten N. Wiesel. "Receptive fields of single neurones in the cat's striate cortex." *The Journal of physiology* 148.3 (1959): 574-591.

Hubel, David H., and Torsten N. Wiesel. "Receptive fields and functional architecture of monkey striate cortex." *The Journal of physiology* 195.1 (1968): 215-243.

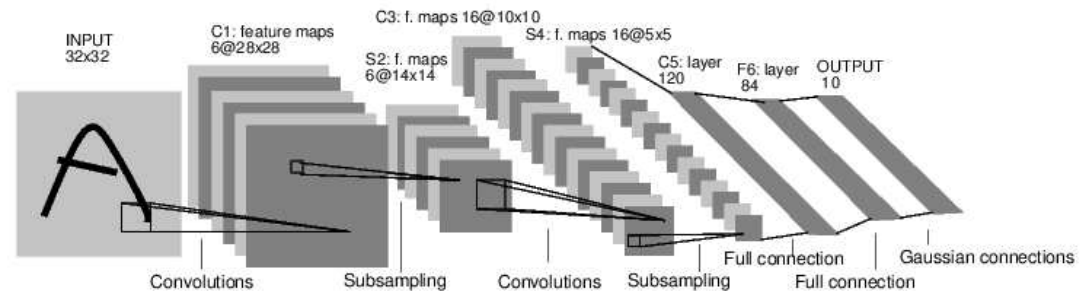
5.1 시각 피질의 구조

- Convolutional Neural Network (CNN)으로 진화
 - LeNet-5[LeCun98]
 - 수표의 손으로 쓴 숫자 인식
 - 합성곱층 Convolution layer, 풀링층 pooling layer 제안



Yann LeCun

Facebook AI & New York Univ.



LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

내용

5.1 시각 피질의 구조

5.2 합성곱층

5.3 풀링층

5.4 CNN 기본 구조

5.5 CNN 구조

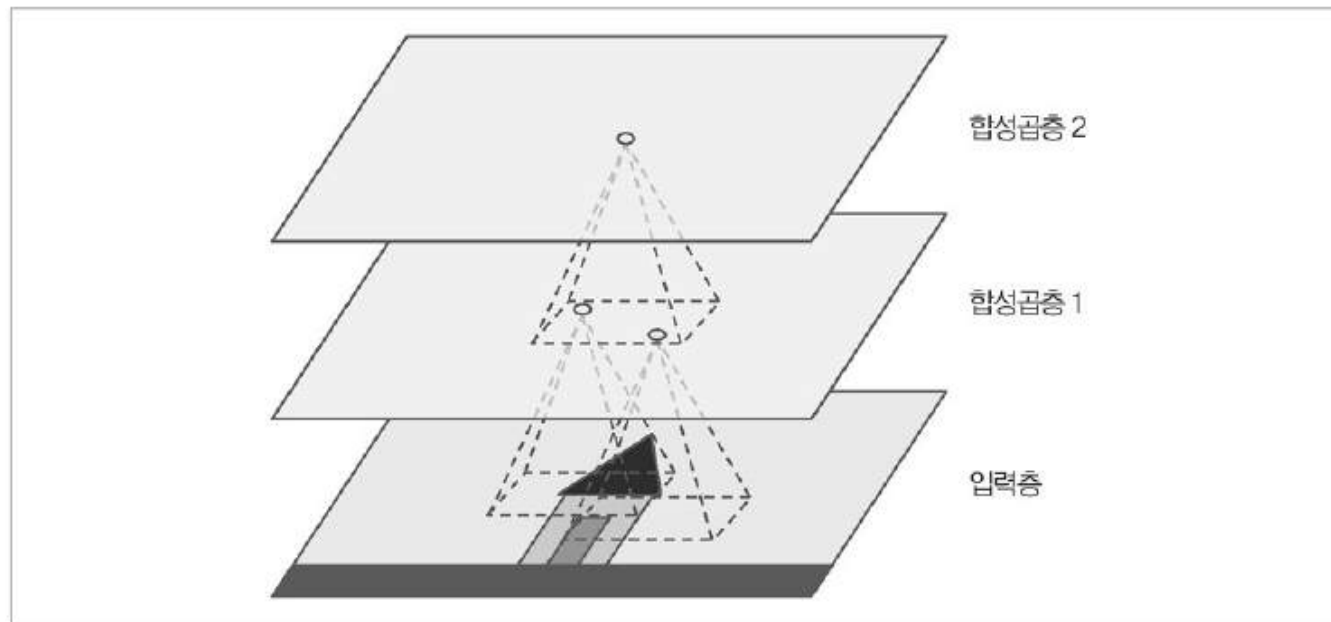
5.2 합성곱층

- CNN 의 구조Convolutional neural network
- (배경지식) 합성곱 연산
- (배경지식) 패딩과 스트라이드Padding & stride
- 합성곱 식

CNN 의 구조

- CNN 의 주요 구성요소: 합성곱층(convolutional layer)
 - 수용장 안에 있는 영역만 상위 레이어 유닛에 연결
 - 첫 번째 히든 레이어는 저수준 특성에 집중, 두 번째 히든 레이어는 고수준 특성으로 조합해 나감

그림 13-2 사각 형태의 국부 수용장을 가진 CNN 층



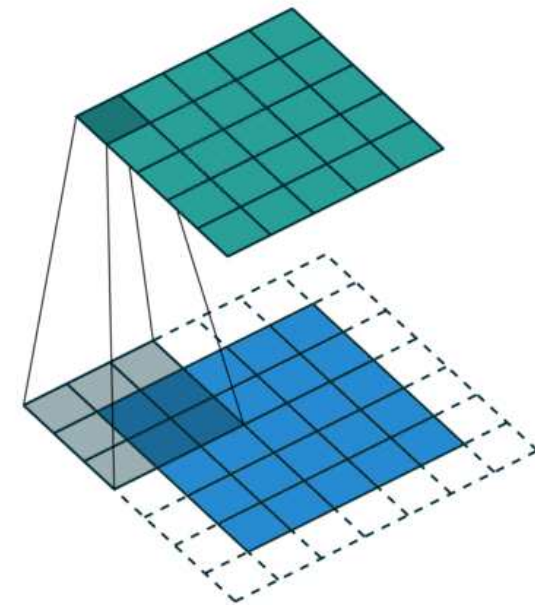
(배경지식) 합성곱 연산 Convolution

- 한 함수 g 가 다른 함수에 대해 얼마나 겹치는 지 계산
 - Kernel(or filter) : 함수 g
 - Stride : 커널의 이동 보폭
 - Padding : 출력 크기를 조정하기 위해 입력 가장자리에 셀을 추가

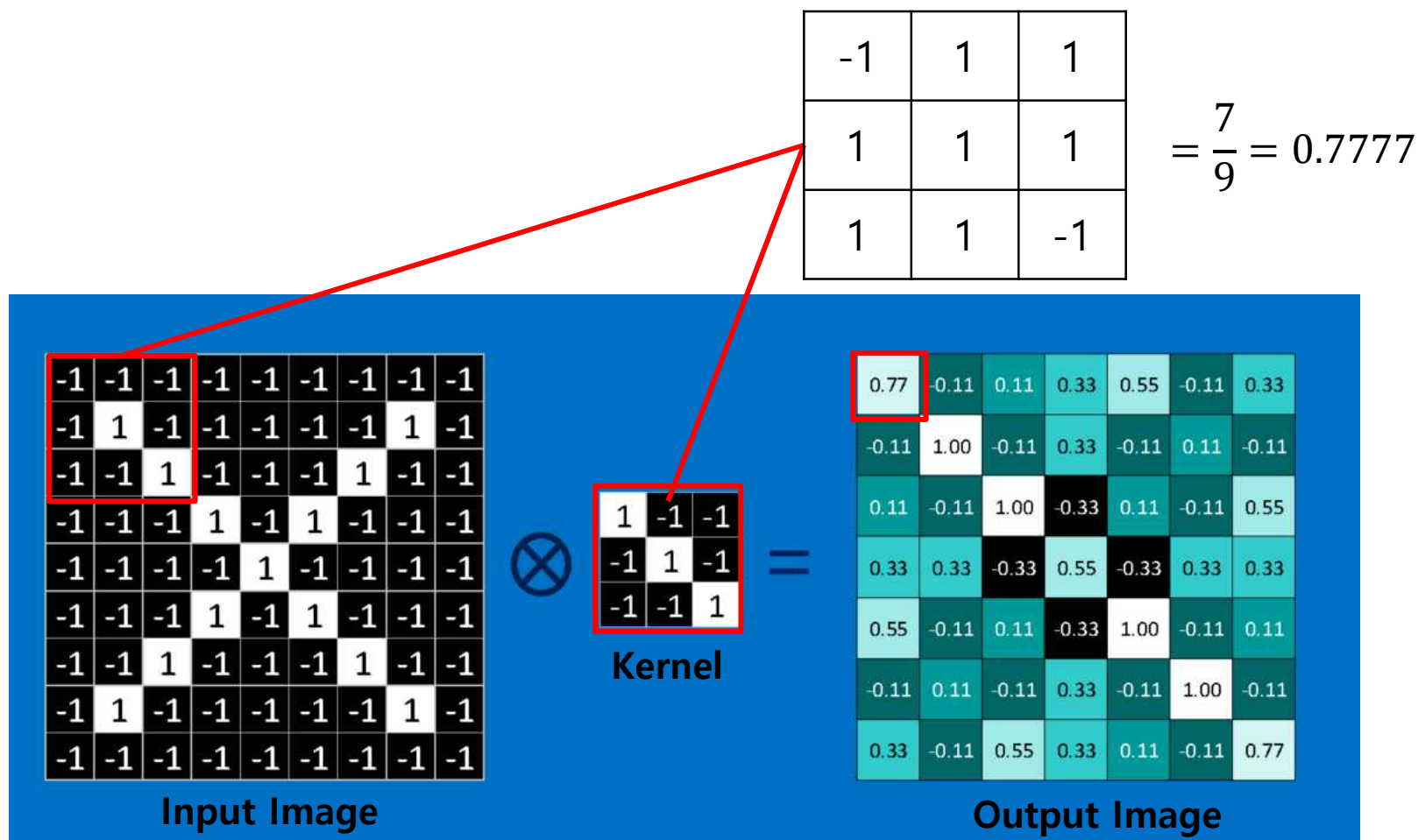


Stride

© adventure.howstuffworks.com

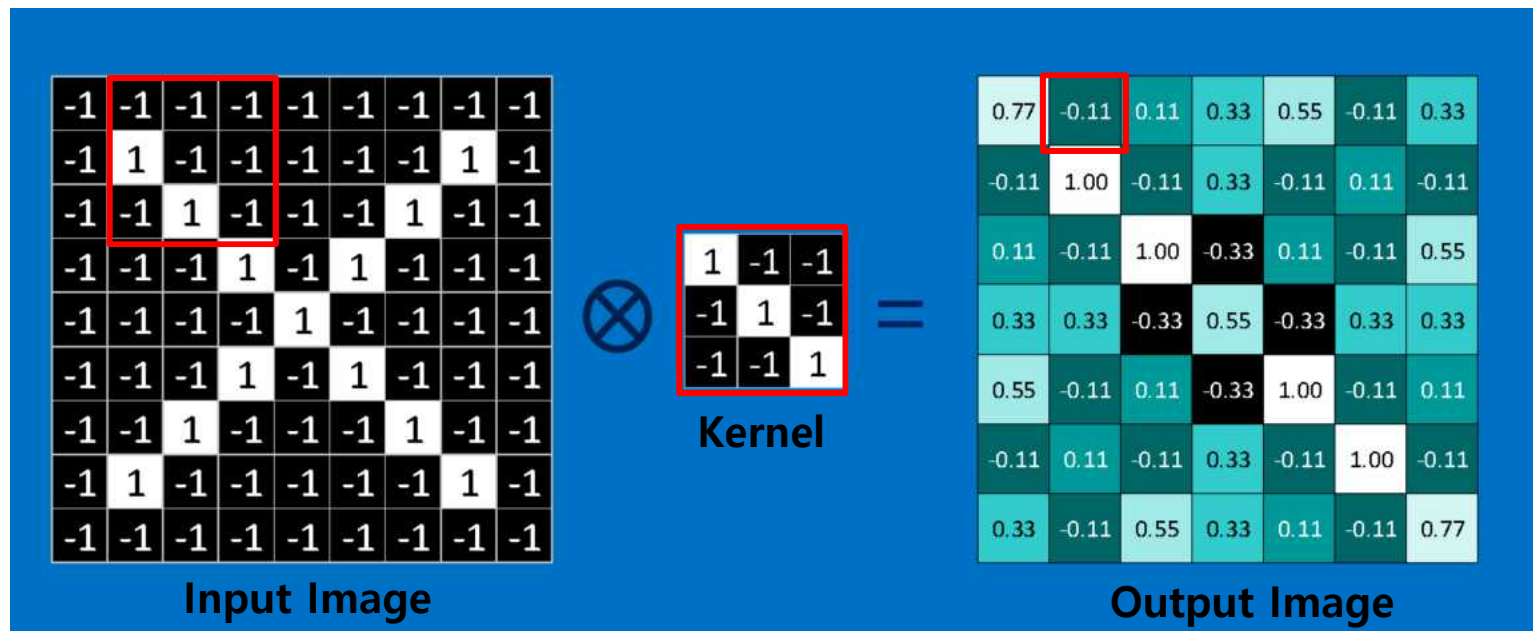
https://github.com/vdumoulin/conv_arithmetic

(배경지식) 합성곱 연산

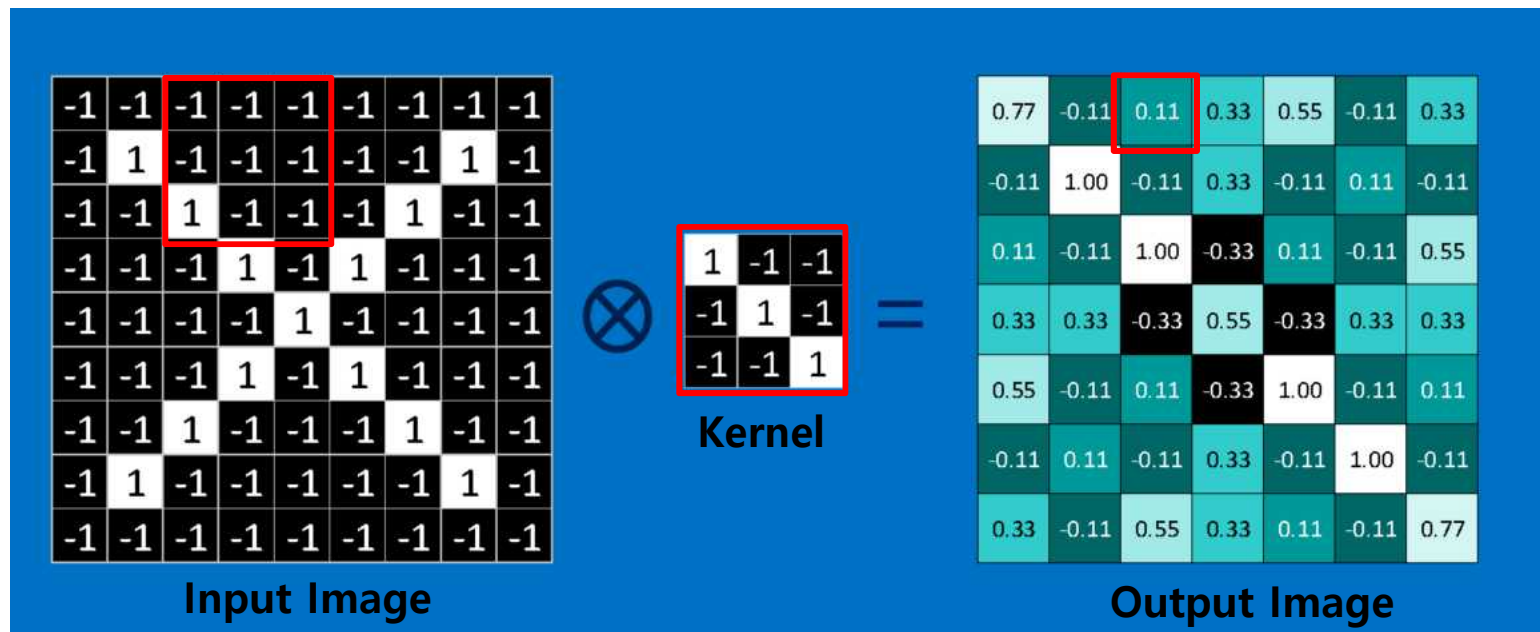


https://brohrer.github.io/how_convolutional_neural_networks_work.html

(배경지식) 합성곱 연산

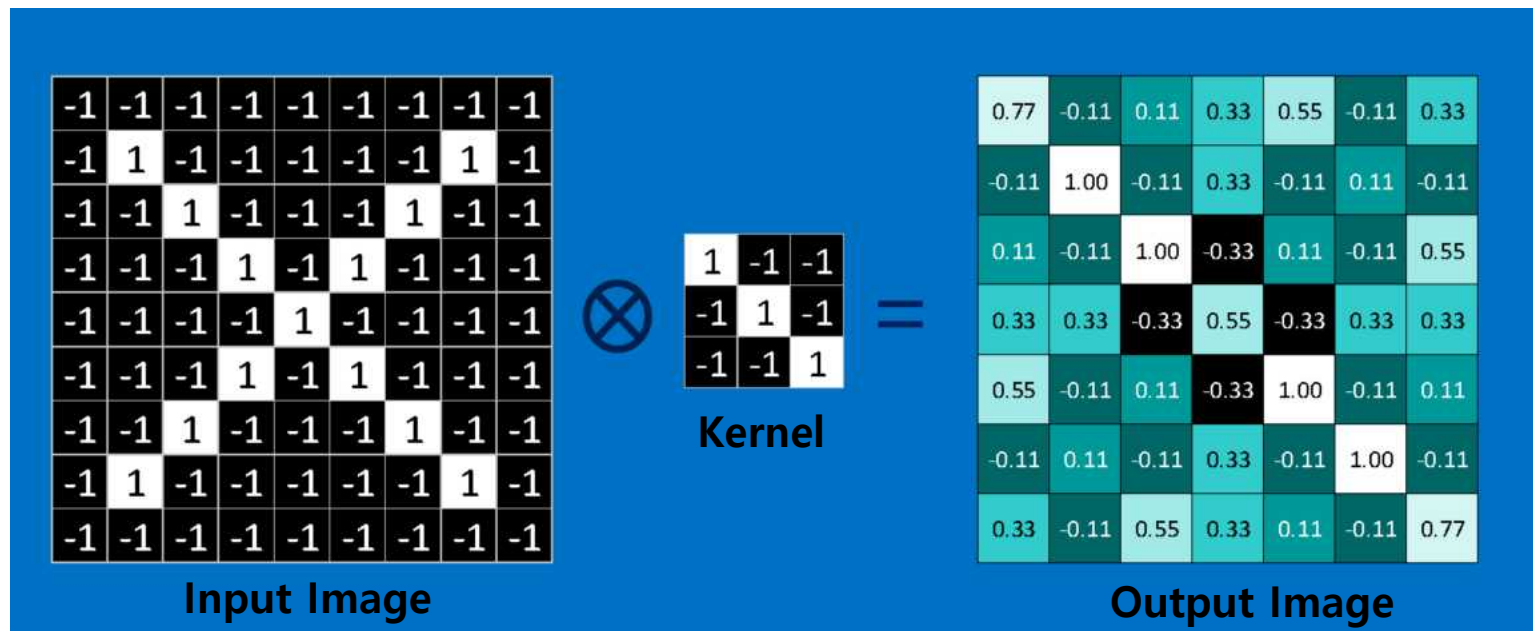


(배경지식) 합성곱 연산



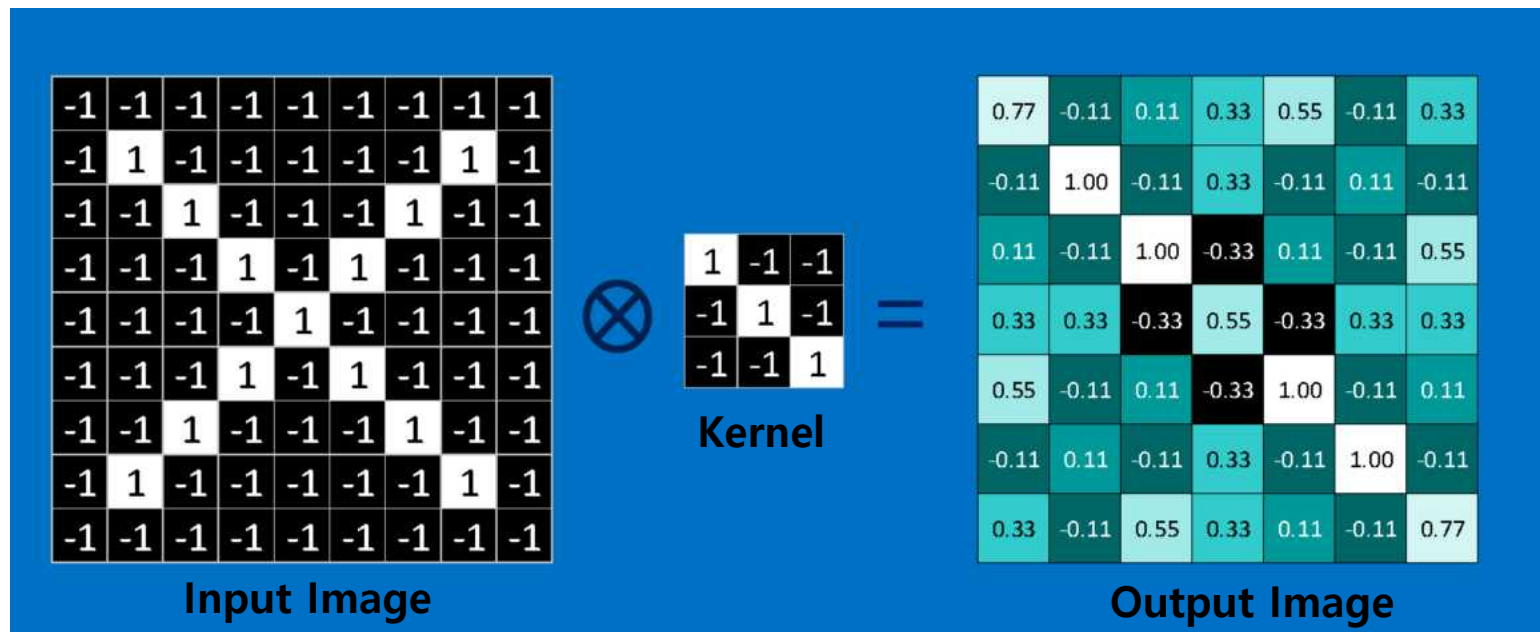
(배경지식) 합성곱 연산

- 출력 영상 Output Image 의 크기: $\text{floor}((I - F)/S + 1)$
 - I: 입력 영상 Input image 의 크기
 - F: 커널의 크기
 - S: 스트라이드



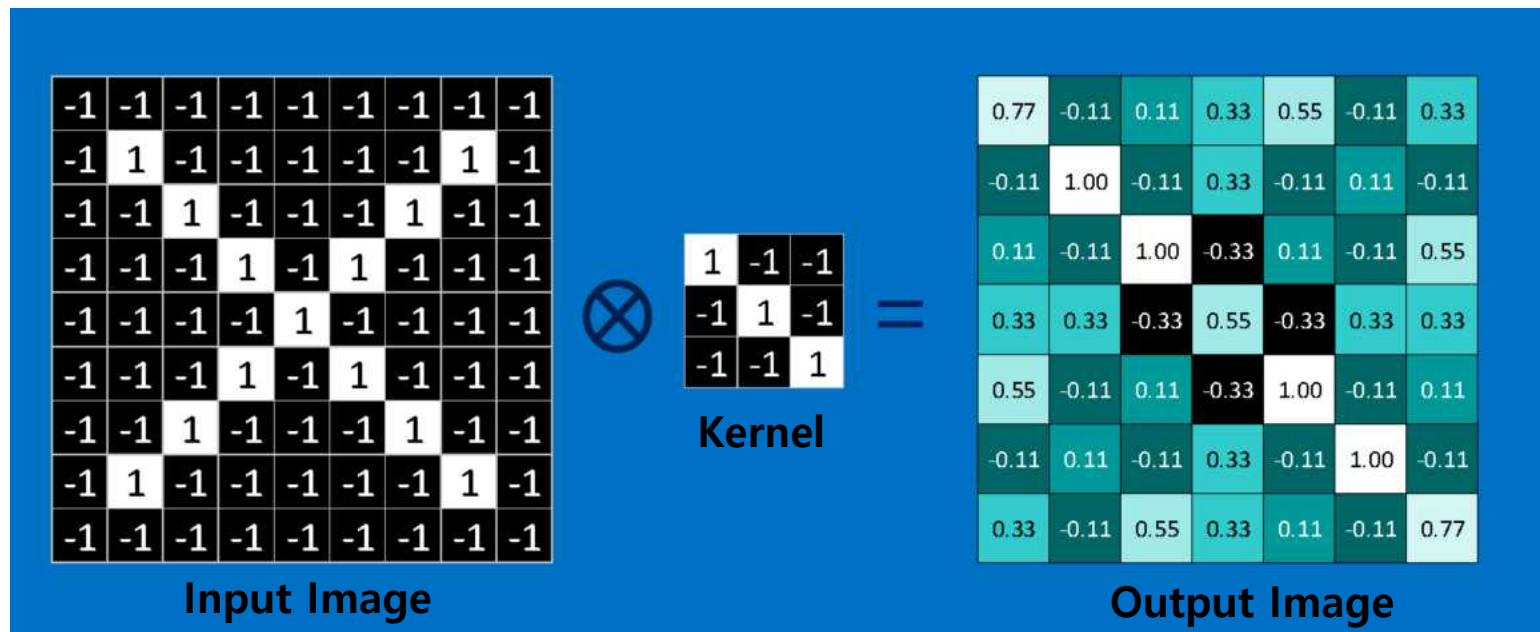
(배경지식) 합성곱 연산

- 커널을 1번 적용한 경우 Output의 크기
– 9x9 → 7x7



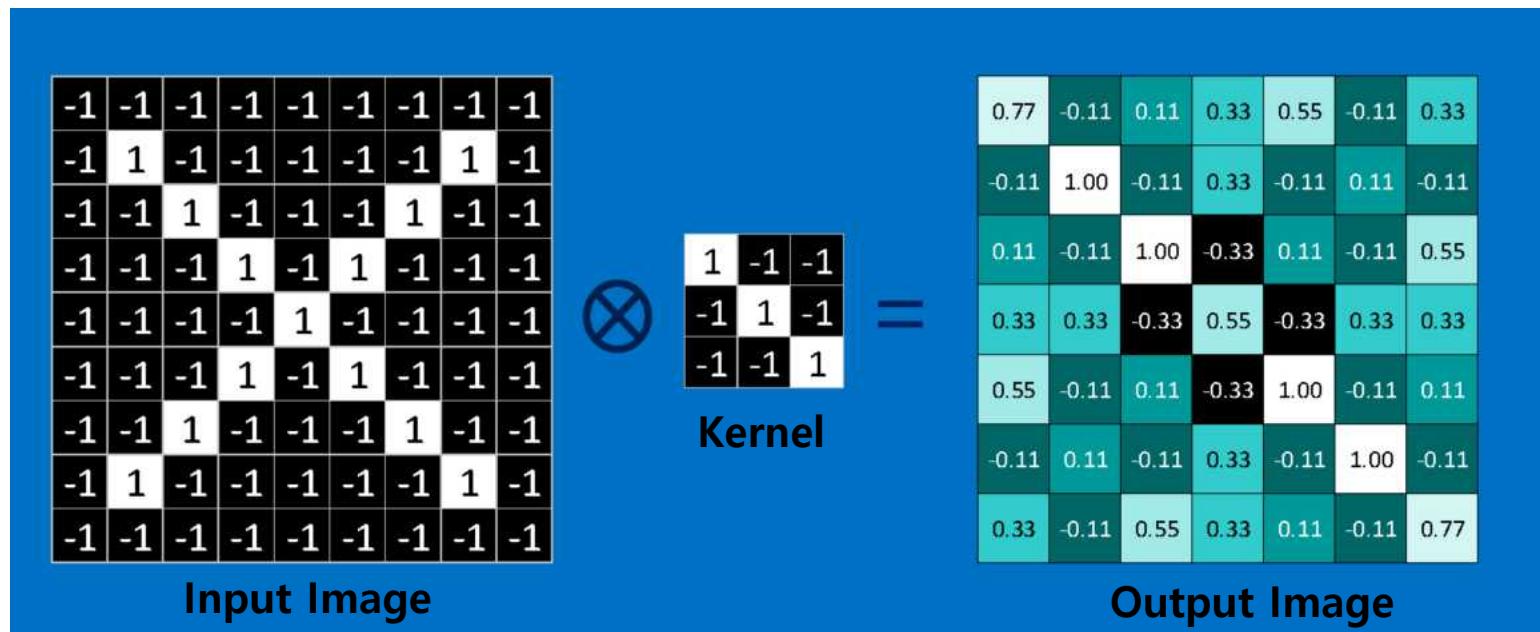
(배경지식) 합성곱 연산

- 커널을 2번 적용한 경우 Output의 크기
– $9 \times 9 \rightarrow 7 \times 7 \rightarrow 5 \times 5$



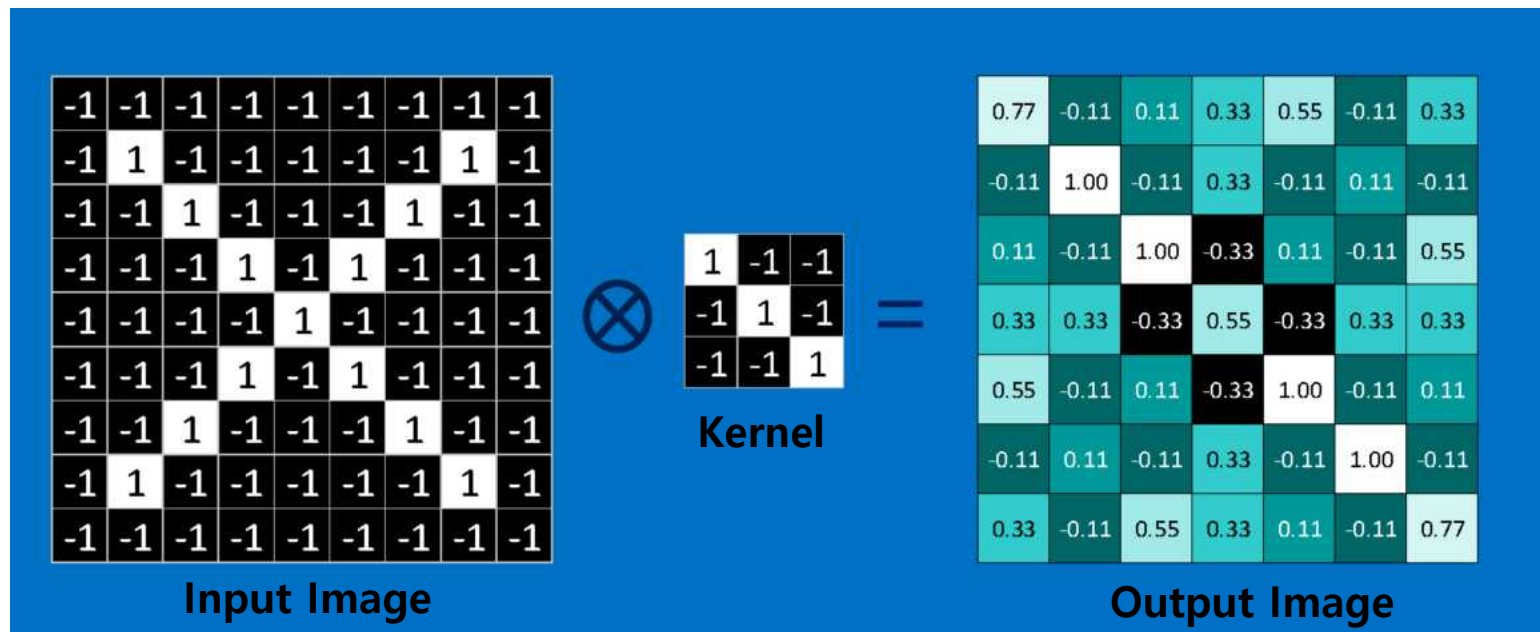
(배경지식) 합성곱 연산

- 커널을 3번 적용한 경우 Output의 크기
– $9 \times 9 \rightarrow 7 \times 7 \rightarrow 5 \times 5 \rightarrow 3 \times 3$



(배경지식) 합성곱 연산

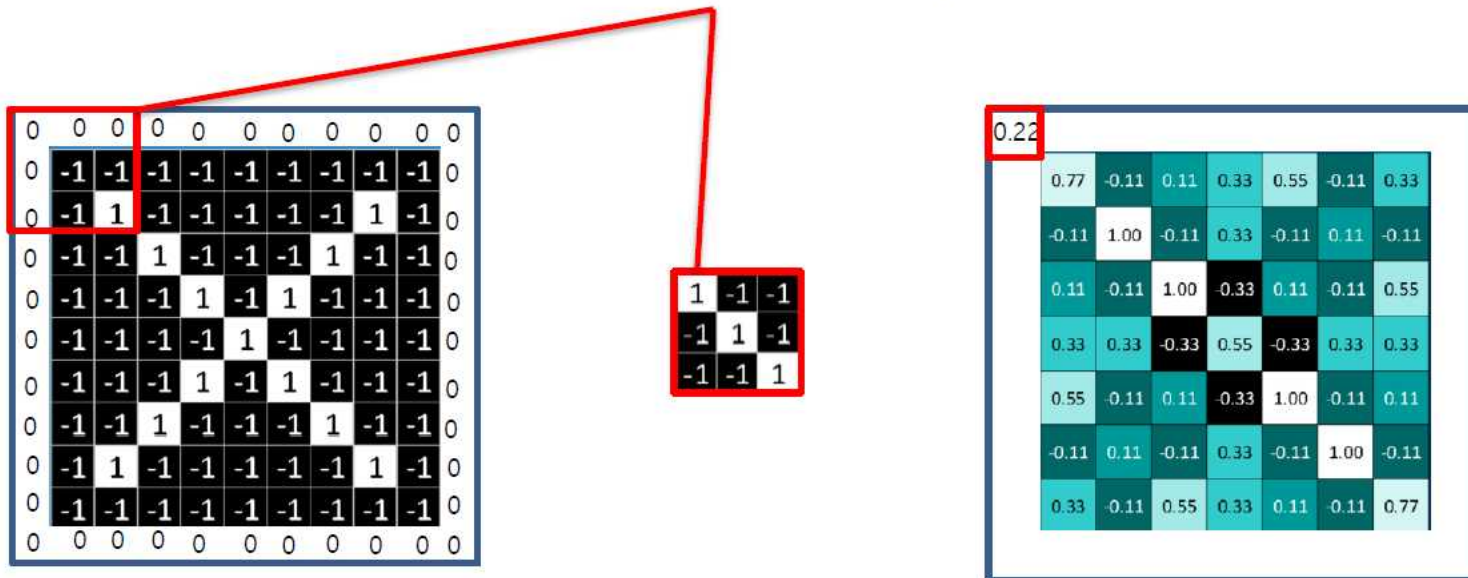
- 커널을 4번 적용한 경우 Output의 크기
 – $9 \times 9 \rightarrow 7 \times 7 \rightarrow 5 \times 5 \rightarrow 3 \times 3 \rightarrow 1 \times 1$



(배경지식) 합성곱 연산

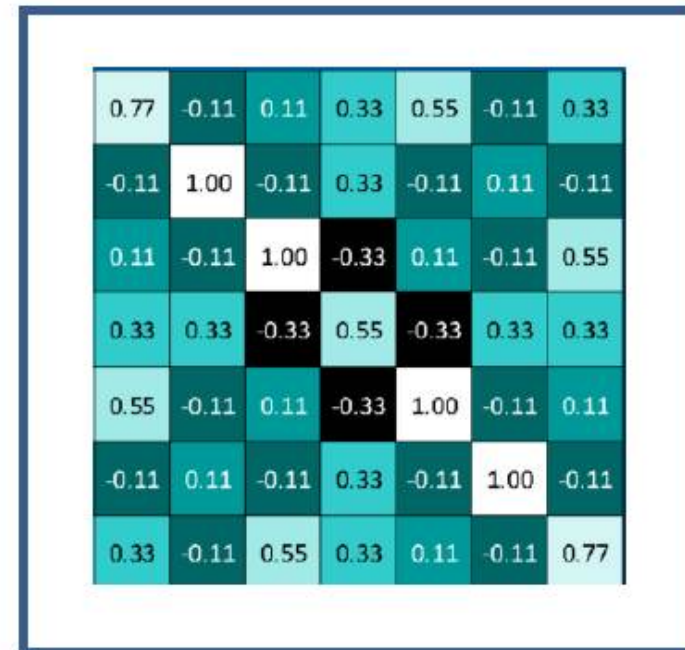
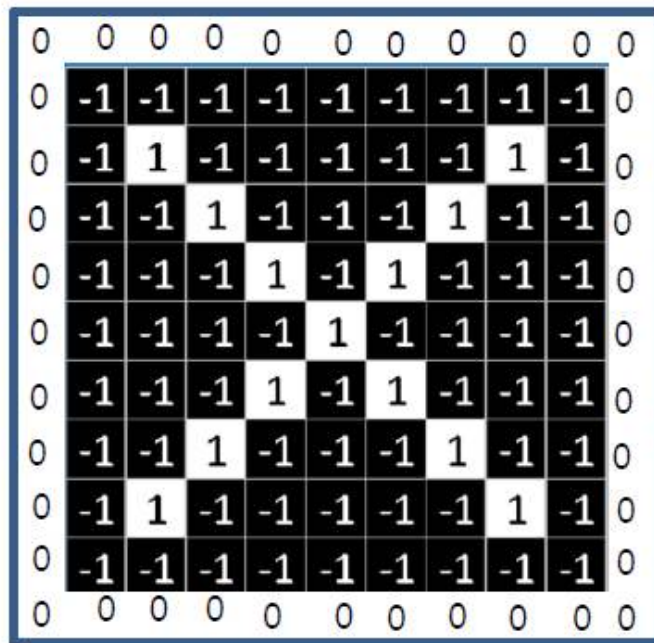
- 패딩^{Padding}: 가장 자리에 공간을 추가하여 크기를 조절
 - 제로 패딩^{Zero padding}: 추가된 공간을 0으로 채움

$$(-1 + 1 + 1 + 1) / 9 = 0.2222$$



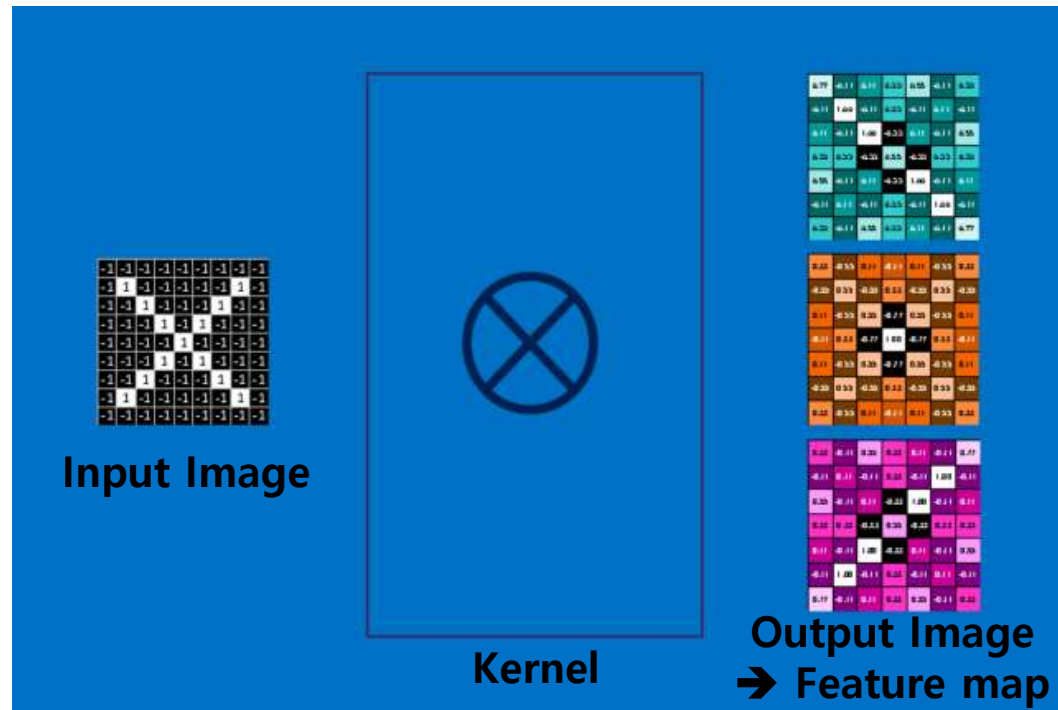
(배경지식) 합성곱 연산

- 출력 영상의 크기: $\text{floor}((I - F + 2P)/S + 1)$
 - I: 입력 영상 Input image 의 크기
 - F: 커널의 크기
 - S: 스트라이드
 - P: 패딩으로 추가된 공간의 크기



(배경지식) 합성곱 연산

- 하나의 영상에 여러 종류의 필터를 적용할 수 있음



(배경지식) 합성곱 연산

- 다양한 고전적인 필터들

입력 영상



Outline 필터

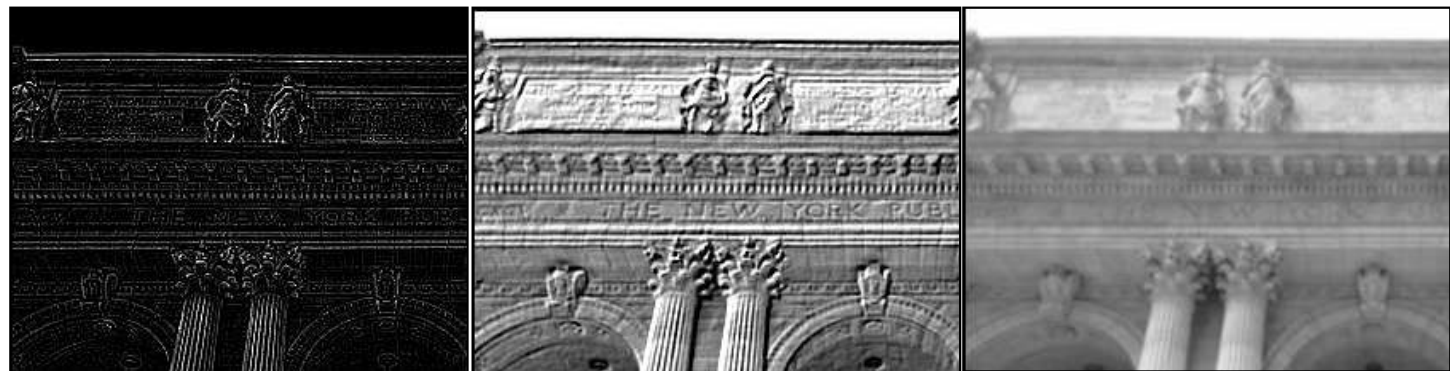
-1	-1	-1
-1	8	-1
-1	-1	-1

Emboss 필터

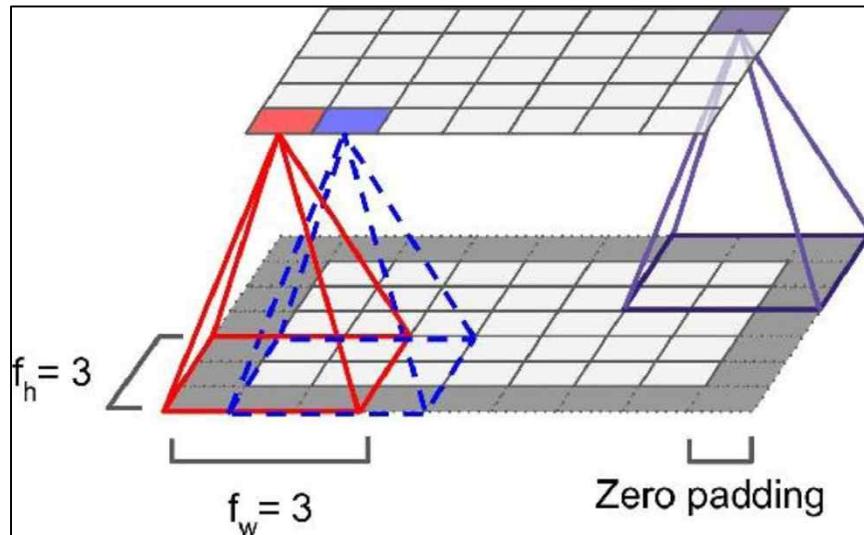
-2	-1	0
-1	1	1
0	1	2

Blur 필터

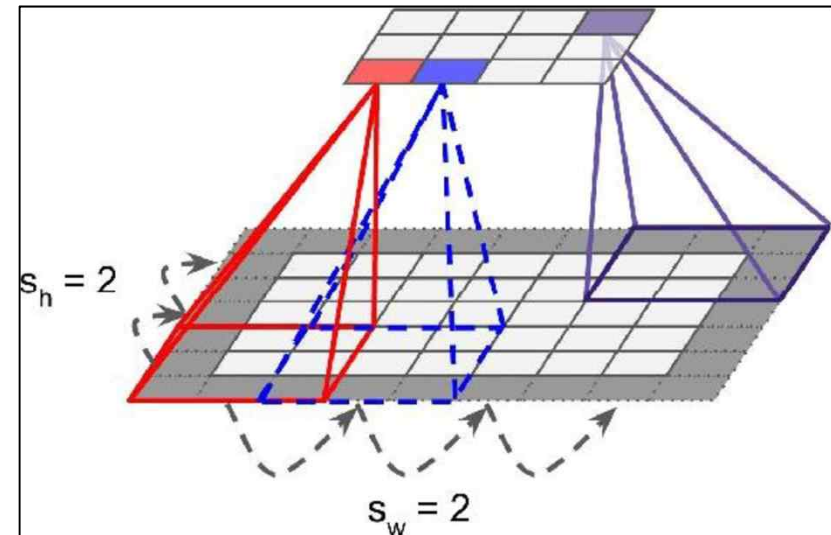
1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16



(배경지식) 패딩과 스트라이드



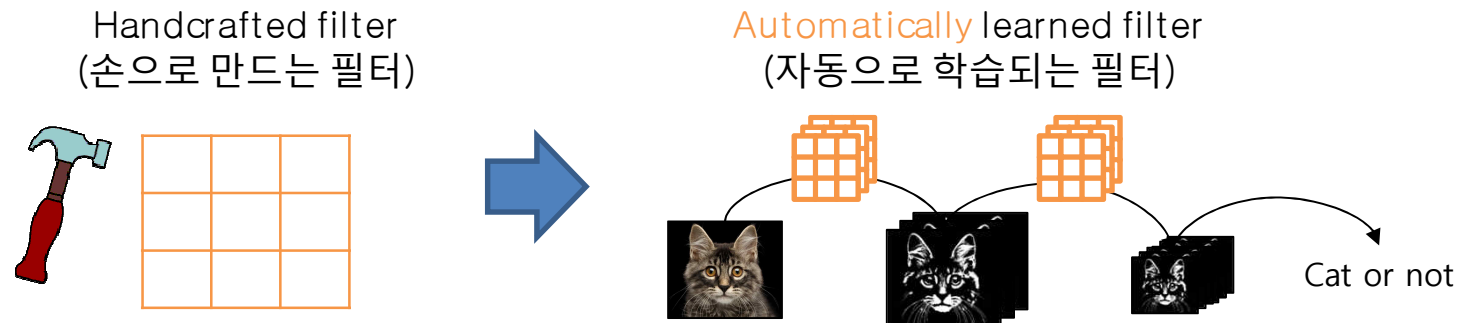
레이어 사이의 연결 모습과 제로 패딩



스트라이드로 특성맵 크기 축소시키기

(배경지식)

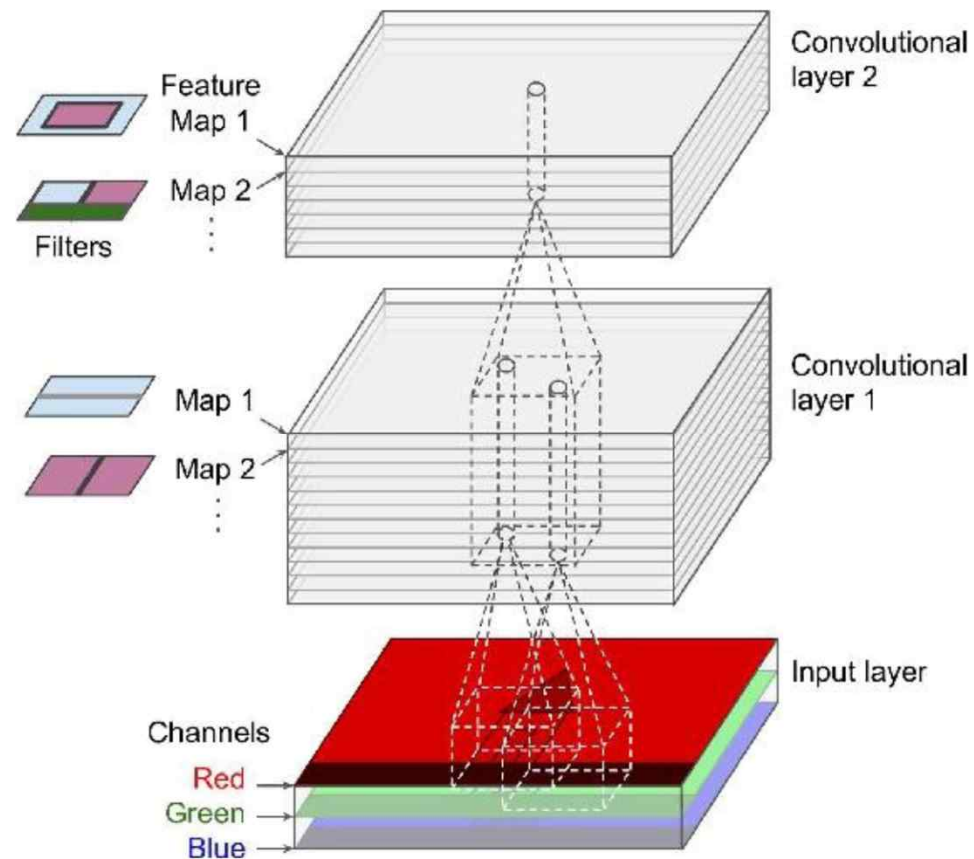
- 딥러닝 이전 vs. 이후 합성곱 연산



합성곱 식

- 합성곱 레이어에 있는 유닛의 출력 계산

$$- z_{i,j,k} = b_k + \sum_{u=0}^{f_h-1} \sum_{v=0}^{f_w-1} \sum_{k'=0}^{f_{n'}-1} x_{i',j',k'} \times w_{u,v,k',k}, \text{ where } \begin{cases} i' = i \times s_h + u \\ j' = j \times s_w + v \end{cases}$$



합성곱 식과 nn.Conv2d()

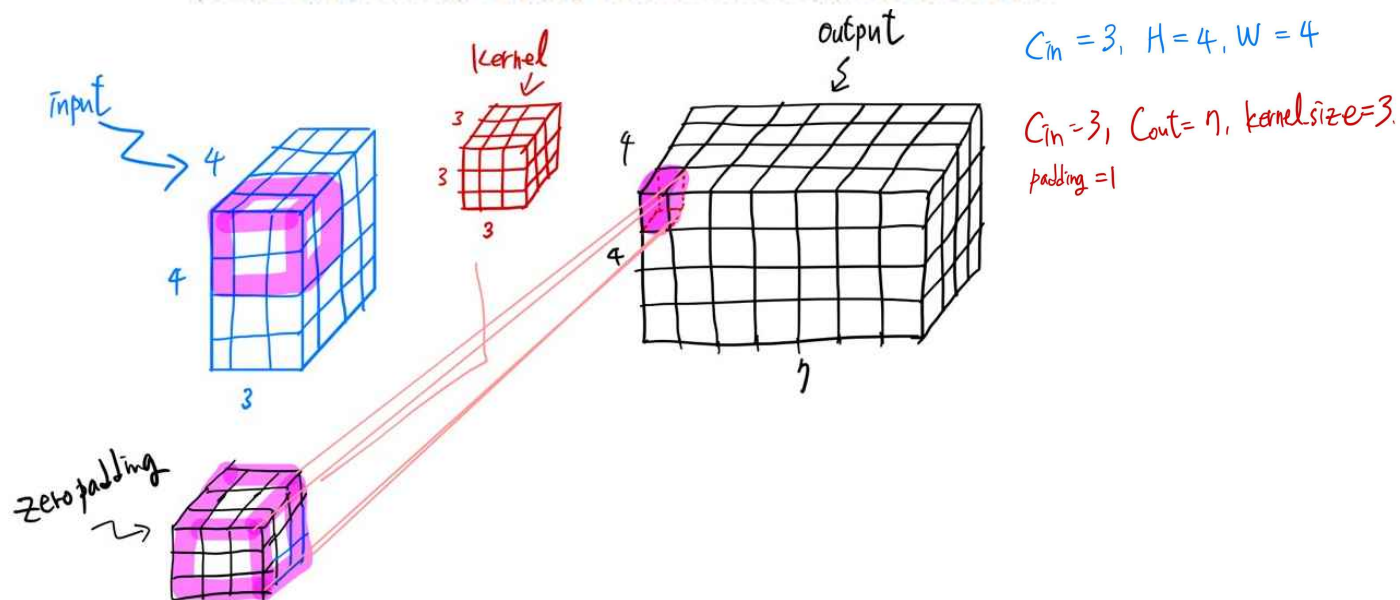
- Conv2d

```
torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1,  
padding=0, dilation=1, groups=1, bias=True, padding_mode='zeros')
```

[SOURCE]

Applies a 2D convolution over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C_{in}, H, W) and output $(N, C_{out}, H_{out}, W_{out})$ can be precisely described as:



합성곱 식과 nn.Conv2d()

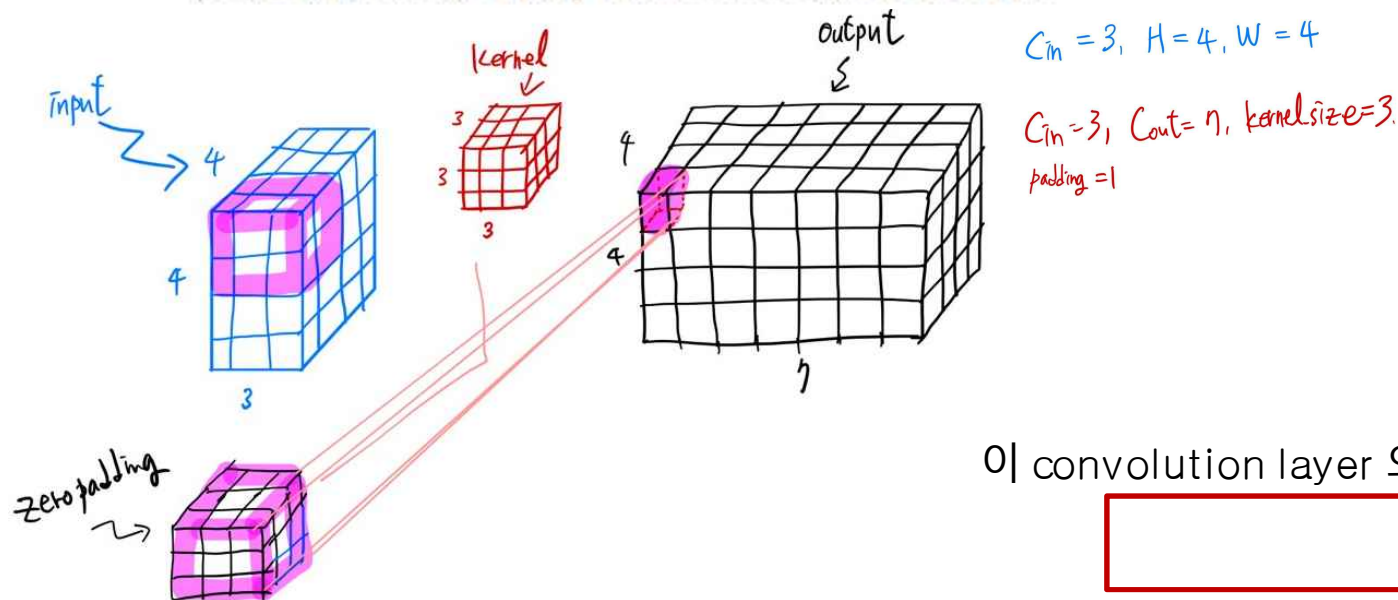
- Conv2d

```
torch.nn.Conv2d(in_channels, out_channels, kernel_size, stride=1,  
padding=0, dilation=1, groups=1, bias=True, padding_mode='zeros')
```

[SOURCE]

Applies a 2D convolution over an input signal composed of several input planes.

In the simplest case, the output value of the layer with input size (N, C_{in}, H, W) and output $(N, C_{out}, H_{out}, W_{out})$ can be precisely described as:



이 convolution layer 의 파라미터 수는?

내용

5.1 시각 피질의 구조

5.2 합성곱층

5.3 풀링층

5.4 CNN 기본 구조

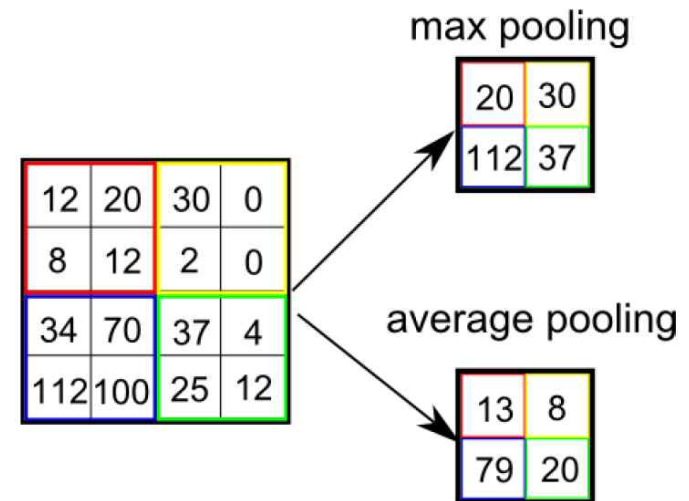
5.5 CNN 구조

5.3 풀링층

- 풀링(pooling)(부표본(subsampling))
 - 정보의 압축, 축소
 - 중요한 정보 또는 평균 정보만 전달
 - 보통 입력 채널 수 = 출력 채널 수

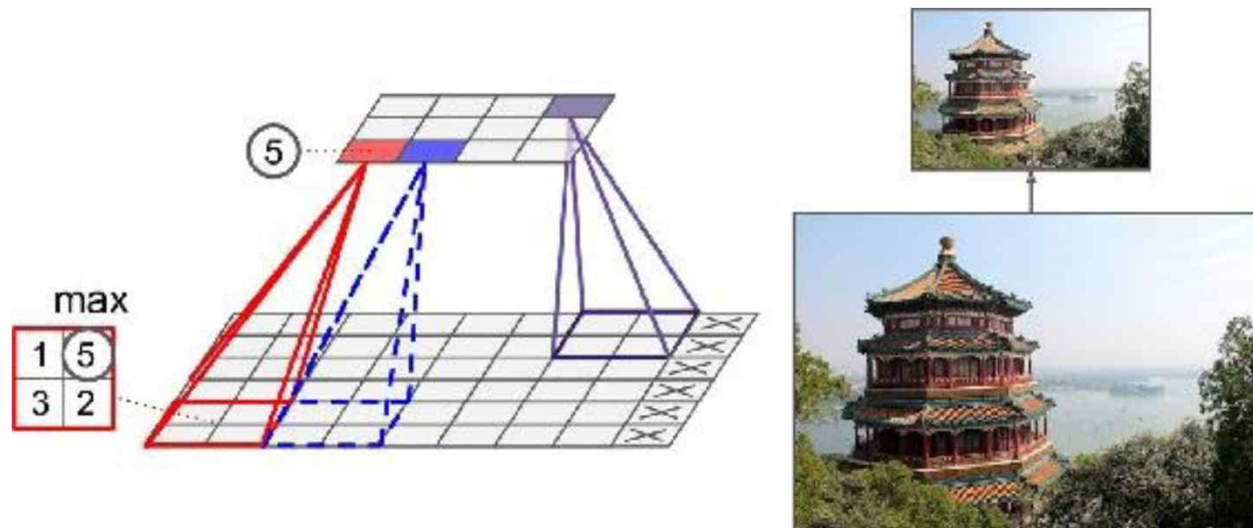


- 종류
 - 최대 풀링(Max pooling)
 - 강한 자극만 기억
 - 평균 풀링(Average pooling)
 - 평균 자극을 기억



5.3 풀링층

- 목적
 - 계산량과 메모리 사용량 ↓
 - 파라미터 수 ↓ (과적합 방지)
 - 위치 불변성 location invariance



최대 풀링층(2x2 풀링 커널, 스트라이드2, 패딩 없음)

내용

5.1 시각 피질의 구조

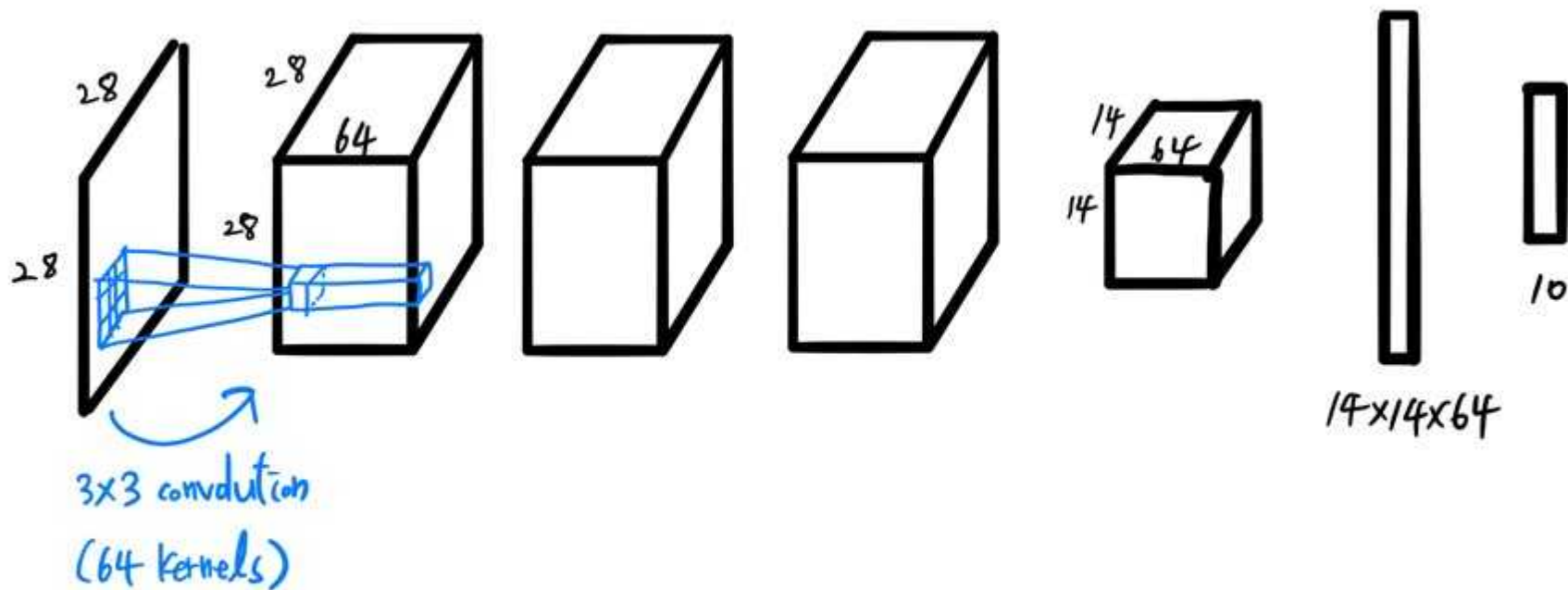
5.2 합성곱층

5.3 풀링층

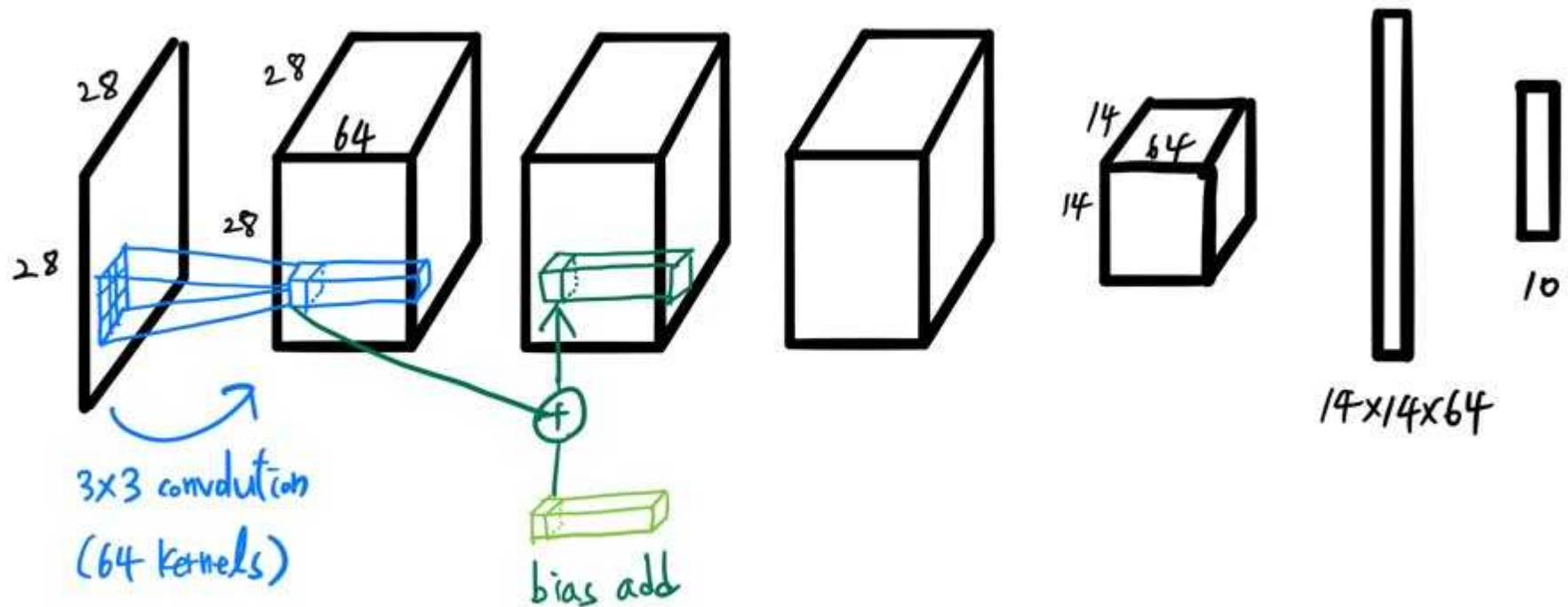
5.4 CNN 기본 구조

5.5 CNN 구조

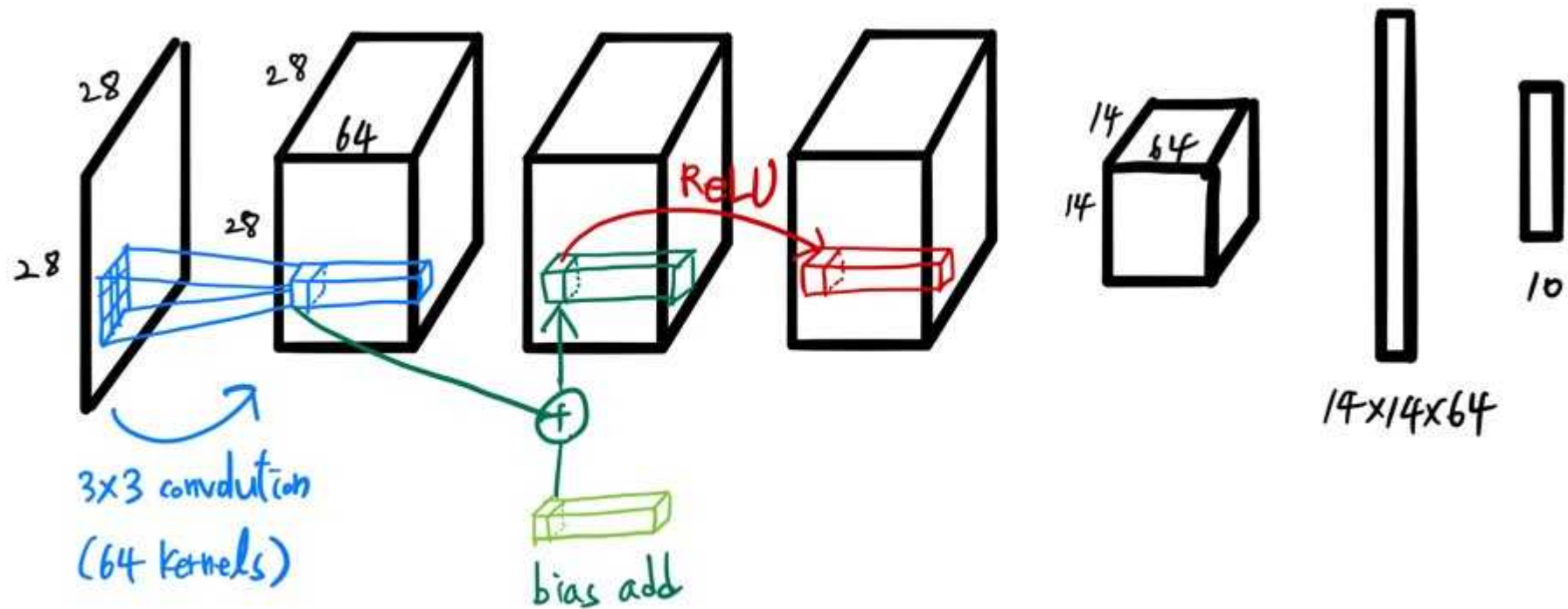
5.4 CNN 기본 구조



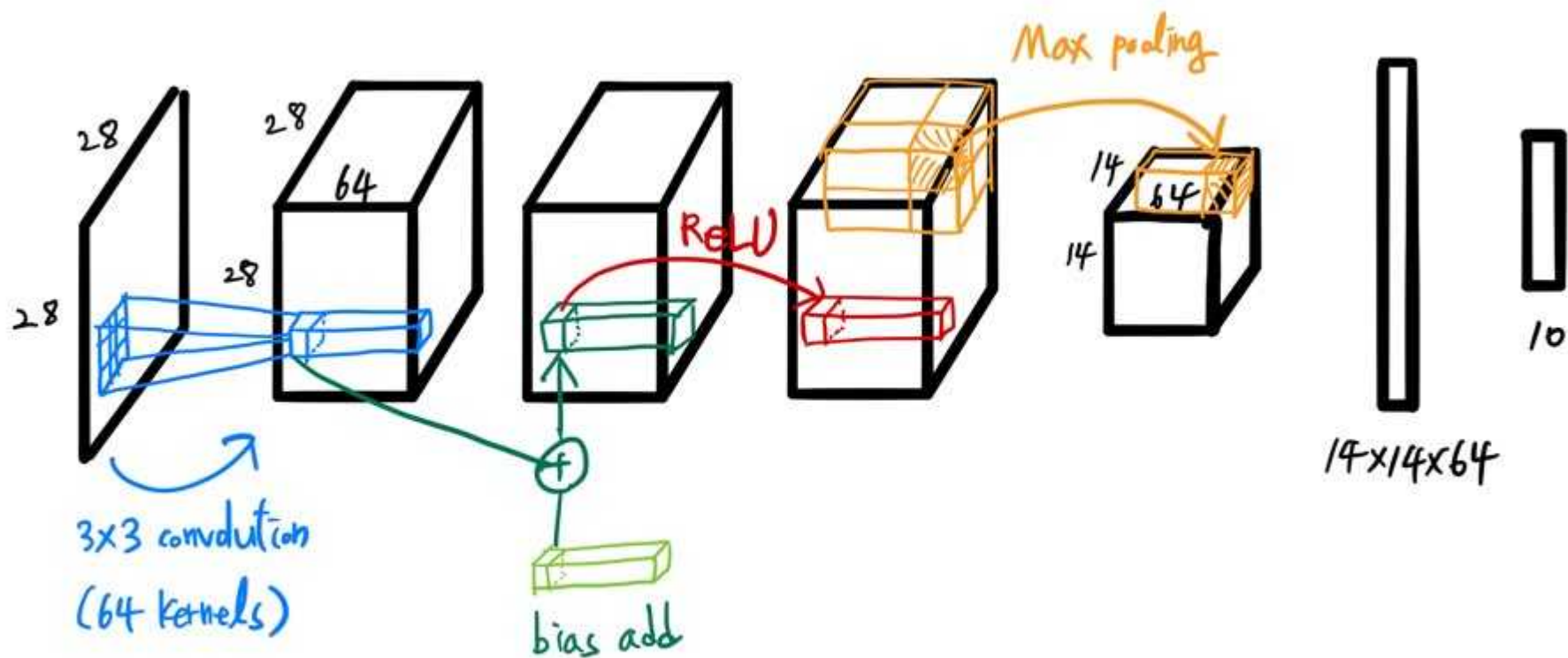
5.4 CNN 기본 구조



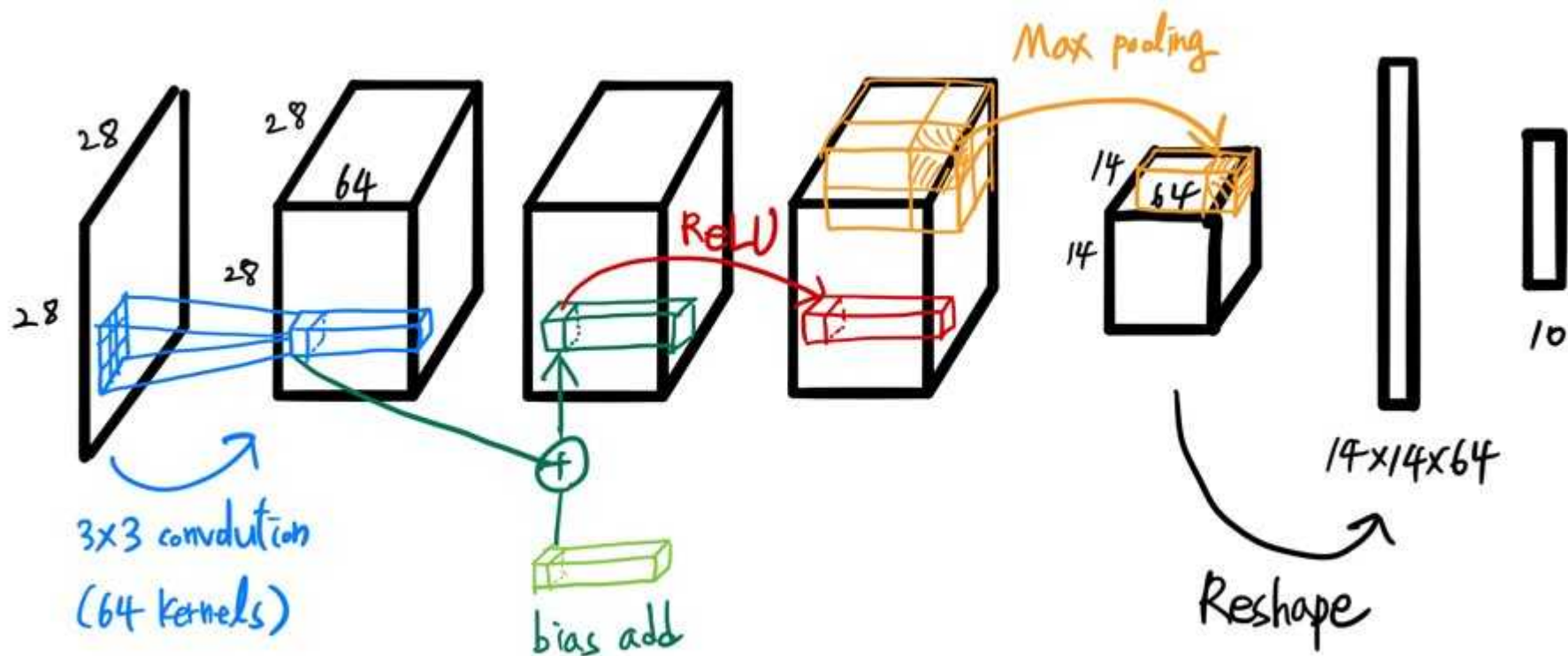
5.4 CNN 기본 구조



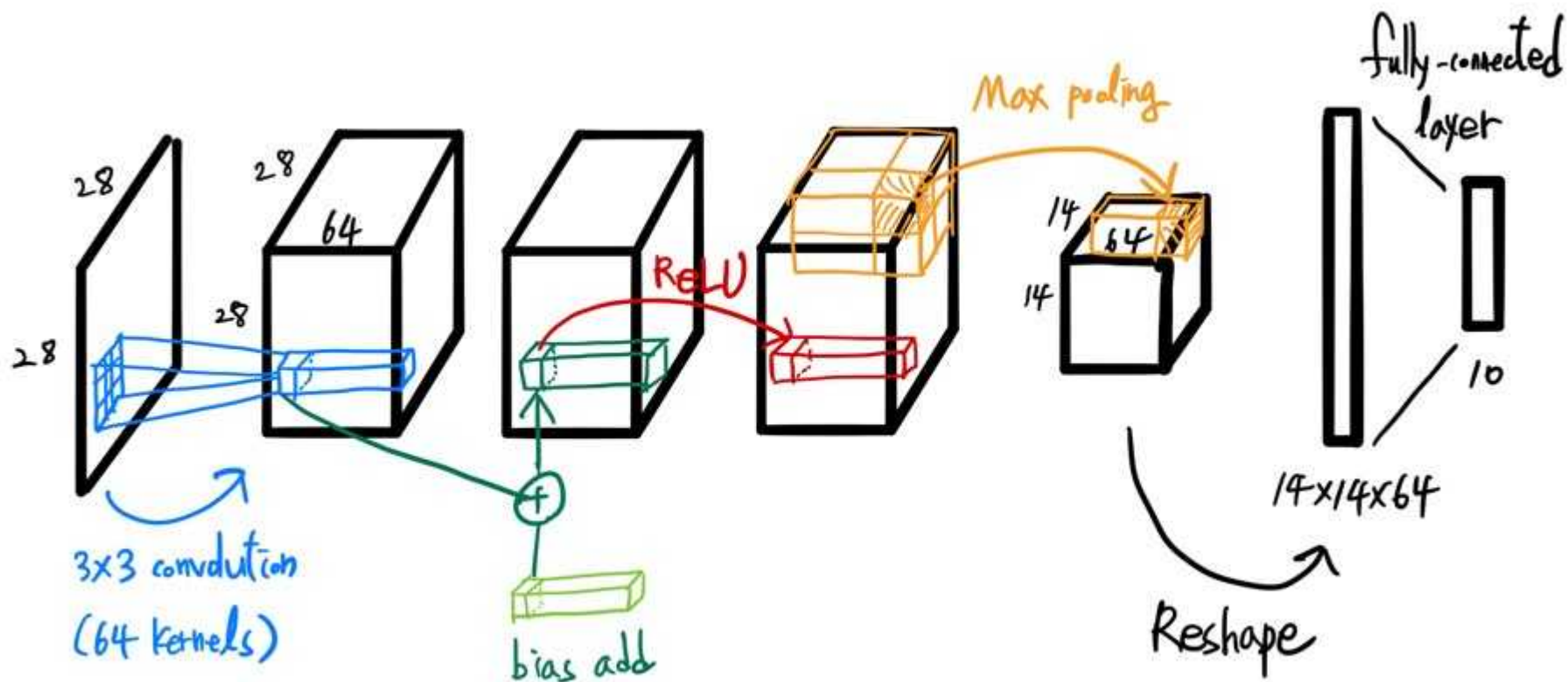
5.4 CNN 기본 구조



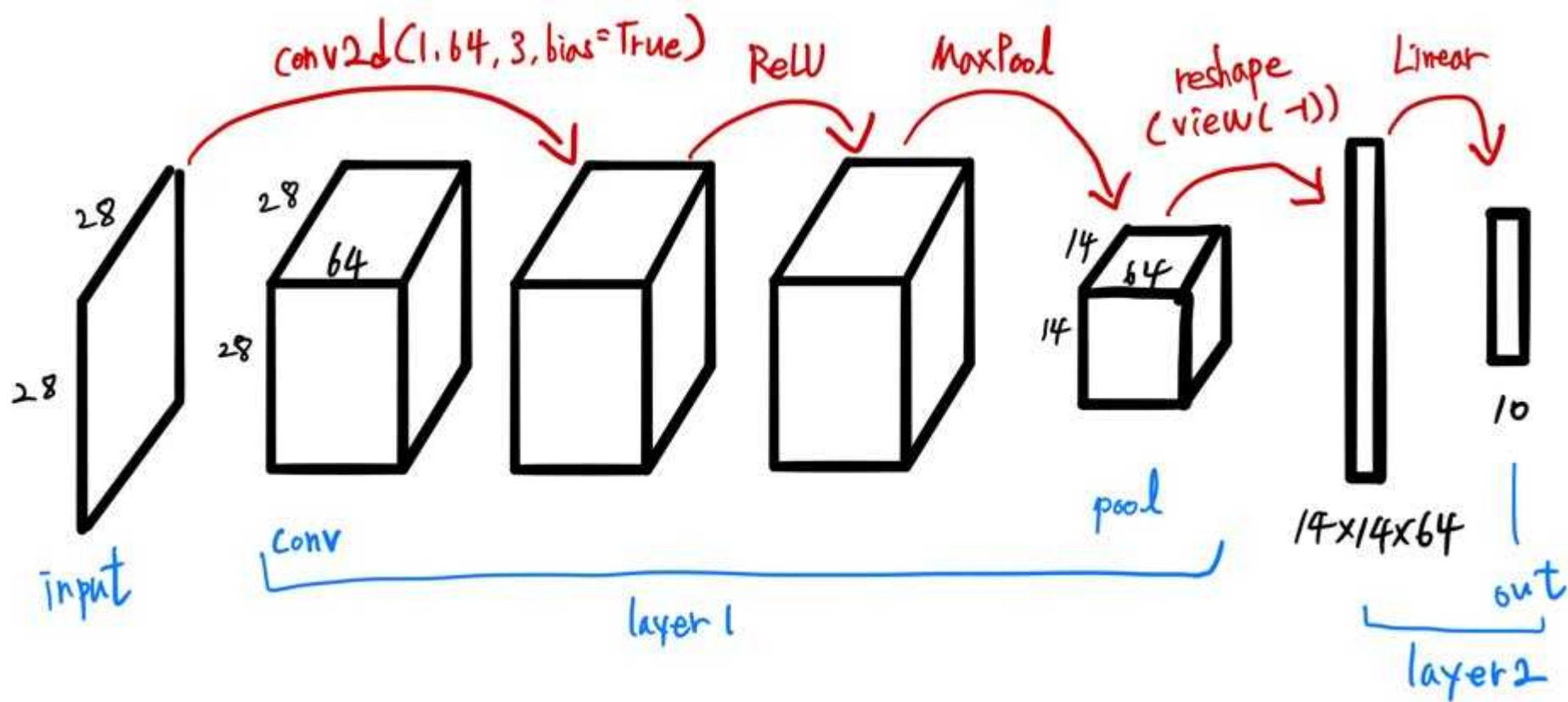
5.4 CNN 기본 구조



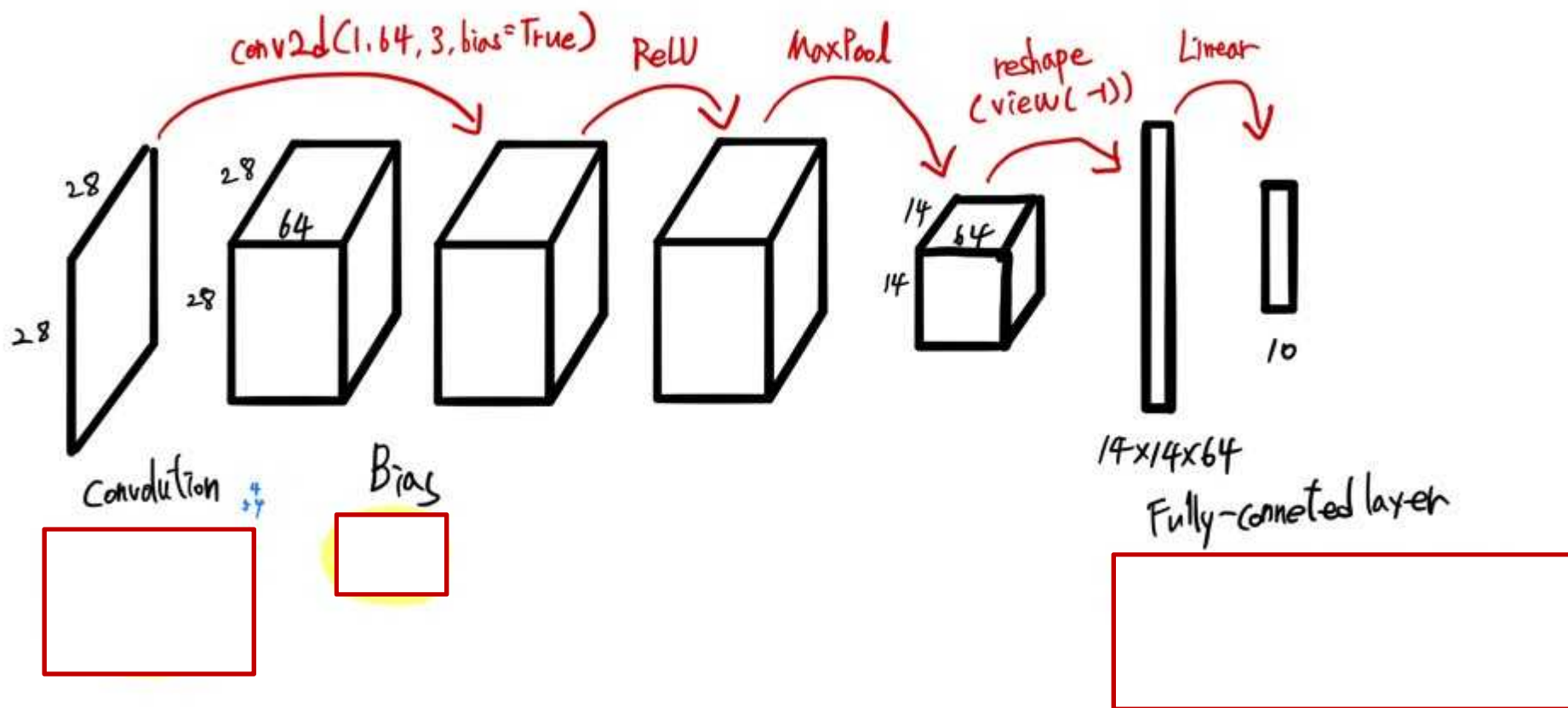
5.4 CNN 기본 구조



5.4 CNN 기본 구조

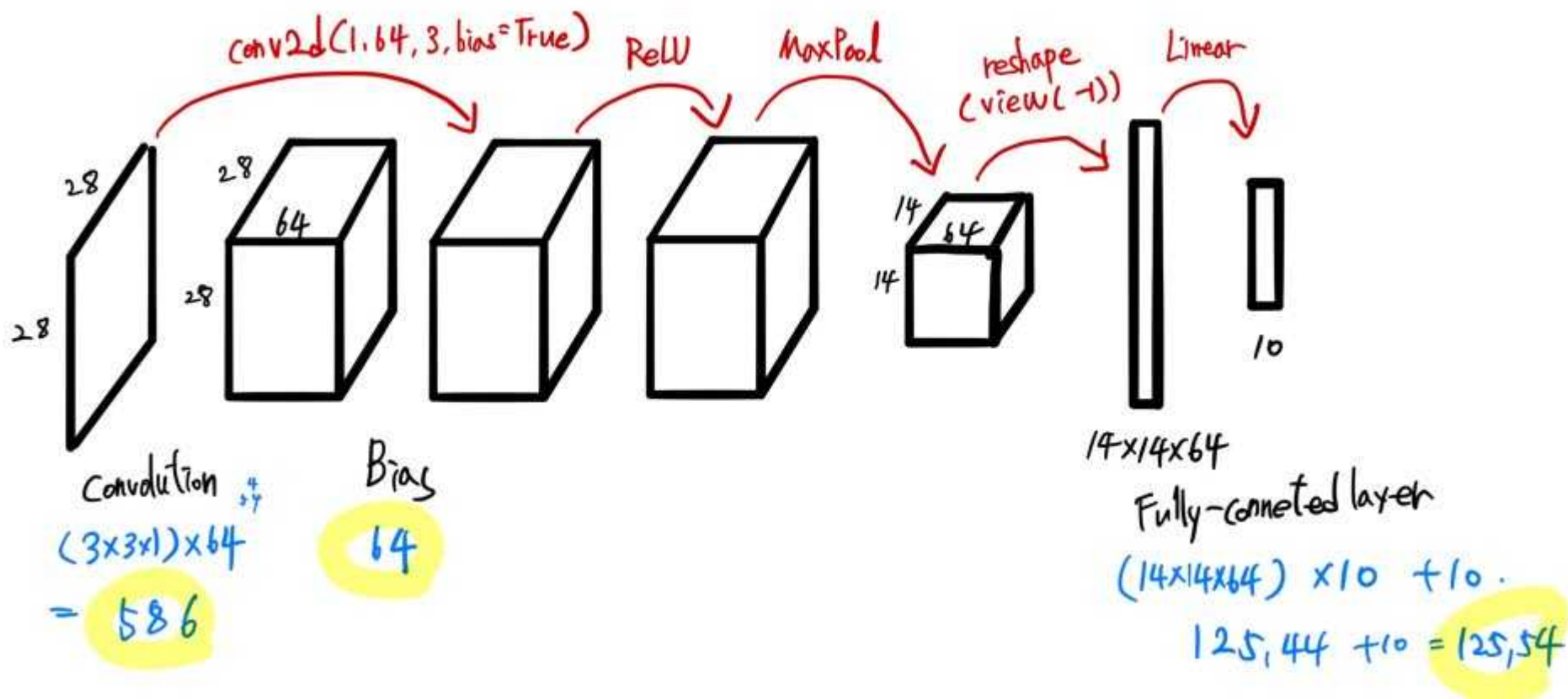


5.4 CNN 기본 구조



파라미터 수는?

5.4 CNN 기본 구조



내용

5.1 시각 피질의 구조

5.2 합성곱층

5.3 풀링층

5.4 CNN 기본 구조

5.5 CNN 구조