

CONVOLUTIONAL NEURAL
NETWORK(CNN)

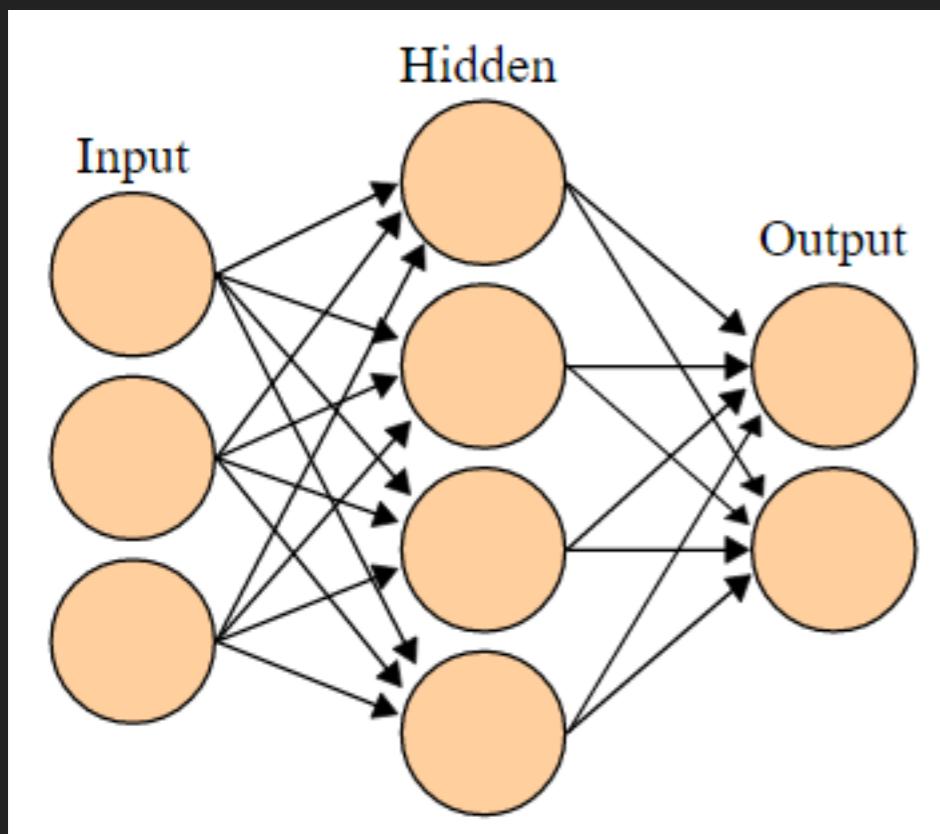
WHAT IS CNN?

GOAL

- ▶ FFNN(Feed-Foward Neural NetworK)을 이해합니다.
- ▶ FFNN와 CNN의 차이를 파악합니다.
- ▶ 컨벌루션의 의미를 이해하기 위한 노력을 해봅니다.
- ▶ CNN의 동작 프로세스를 이해합니다.
- ▶ 왜 CNN이 이미지, 음성에 쓰이는지 이해합니다.
- ▶ CNN의 각각의 동작이 어떻게 동작하는 이해합니다.

FFNN(FEED-FOWARD NEURAL NETWORK)

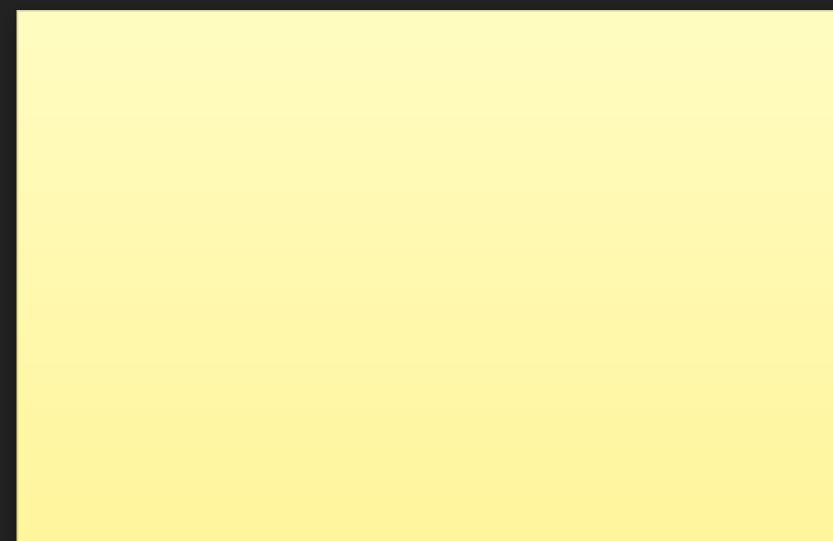
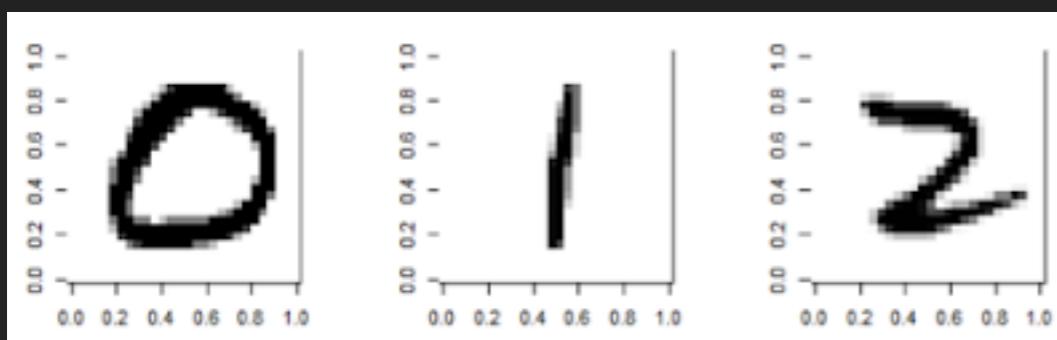
1. Input x
2. 이를 $y = Wx + b$ 와 유사한-
3. Activation Function (sigmoid, ReLu) 를 적용 ($a = b(y)$)
4. output layer를 쌓아 올린 것이 바로 Multilayer Perceptron(MLP) 이때 softmax와 같이 분류할 수 있는 함수를 Activation function으로 활용한다.



CNN의 경우에는...

1. Input x
2. 이를 $y = Wx + b$ 계산한 뒤
3. Activation Function (sigmoid, ReLu) 를 적용 ($a = b(y)$)
4. output layer를 쌓아 올린 것이 바로 Multilayer Perceptron(MLP) 이때 softmax와 같이 분류할 수 있는 함수를 Activation function으로 활용한다.

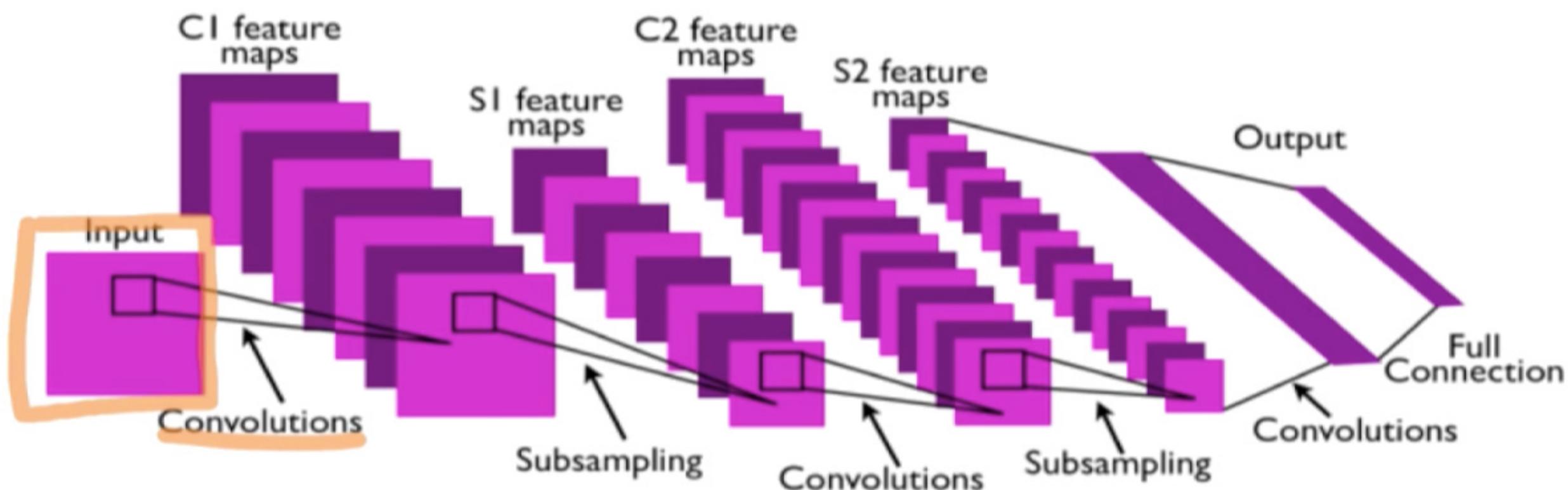
2번의 과정이 다르다.



그럼 다시 CNN으로 돌아가자.

CNN

Convolutional Neural Network



1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

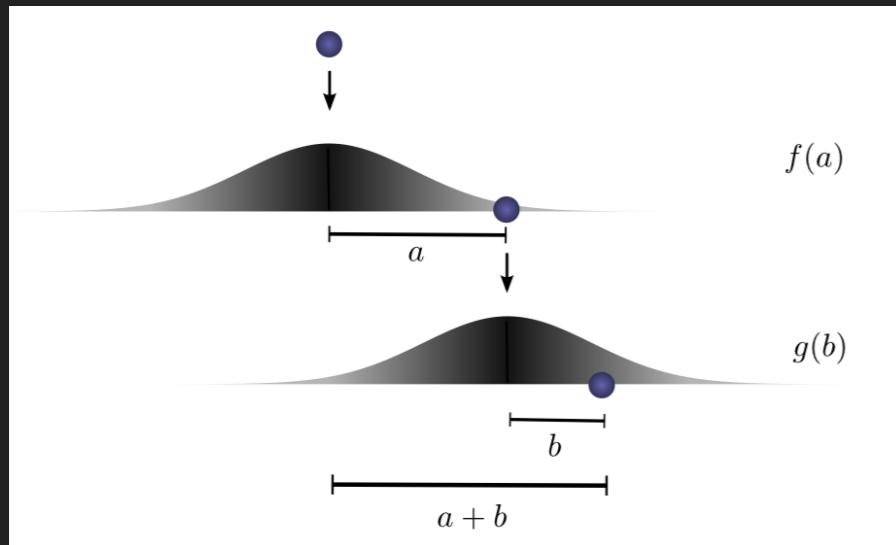
Convolved Feature

Convolutions + Subsampling + Full Connection

CONVOLUTION

- ▶ 하나의 함수와 또 다른 함수를 반전 이동한 값을 곱한 다음, 구간에 대해 적분해서 새로운 함수를 구하는 수학 연산자.
- ▶ 내가 가진 컨벌루션 필터(커널)와 그 위치가 수치적으로 얼마나 비슷한가?
- ▶ 주위 값들을 반영해 중앙의 값을 변화시키는 것
: 그럼 어떤 쪽으로 변화? 바로 목적하는 작업의 성공률이 높도록 이미지를 변형.

CONVOLUTION

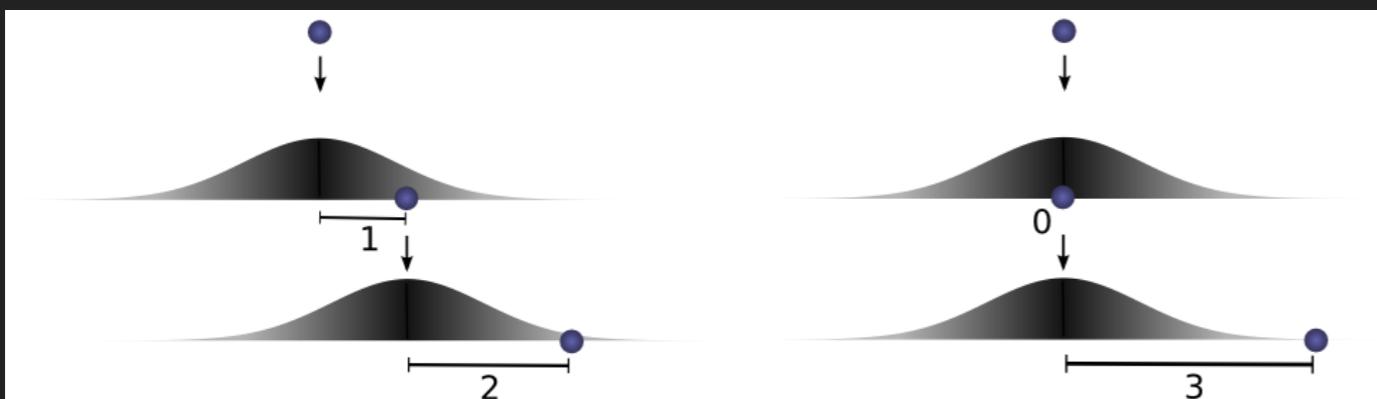
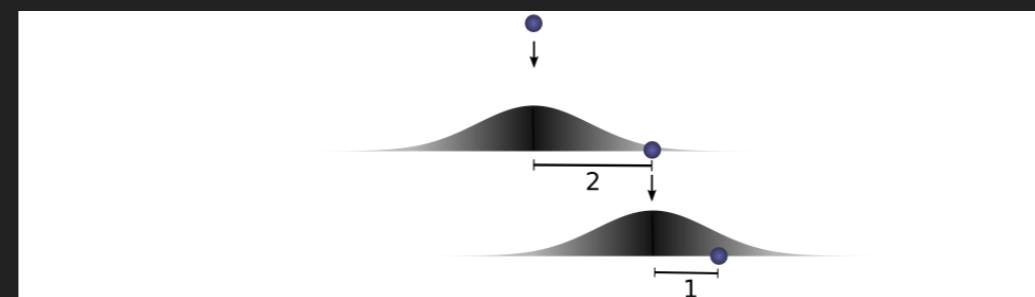


a 만큼가는 확률을 $f(a)$

a 위치에서 다시 공을 낙하 시켰을 때 간 거리 $g(b)$

그러면 $a+b = c$ 라면 거리 c 까지 가기위한 확률은 $f(a)g(b)$

예를 들어 총 거리가 3이라면 1만큼가고 2만큼 || 0만큼 3만큼



$$\dots f(0) \cdot g(3) + f(1) \cdot g(2) + f(2) \cdot g(1) \dots$$

CONVOLUTION

$$\sum_{a+b=c} f(a) \cdot g(b)$$

Turns out, we're doing a convolution! In particular, the convolution of f and g , evaluated at c is defined:

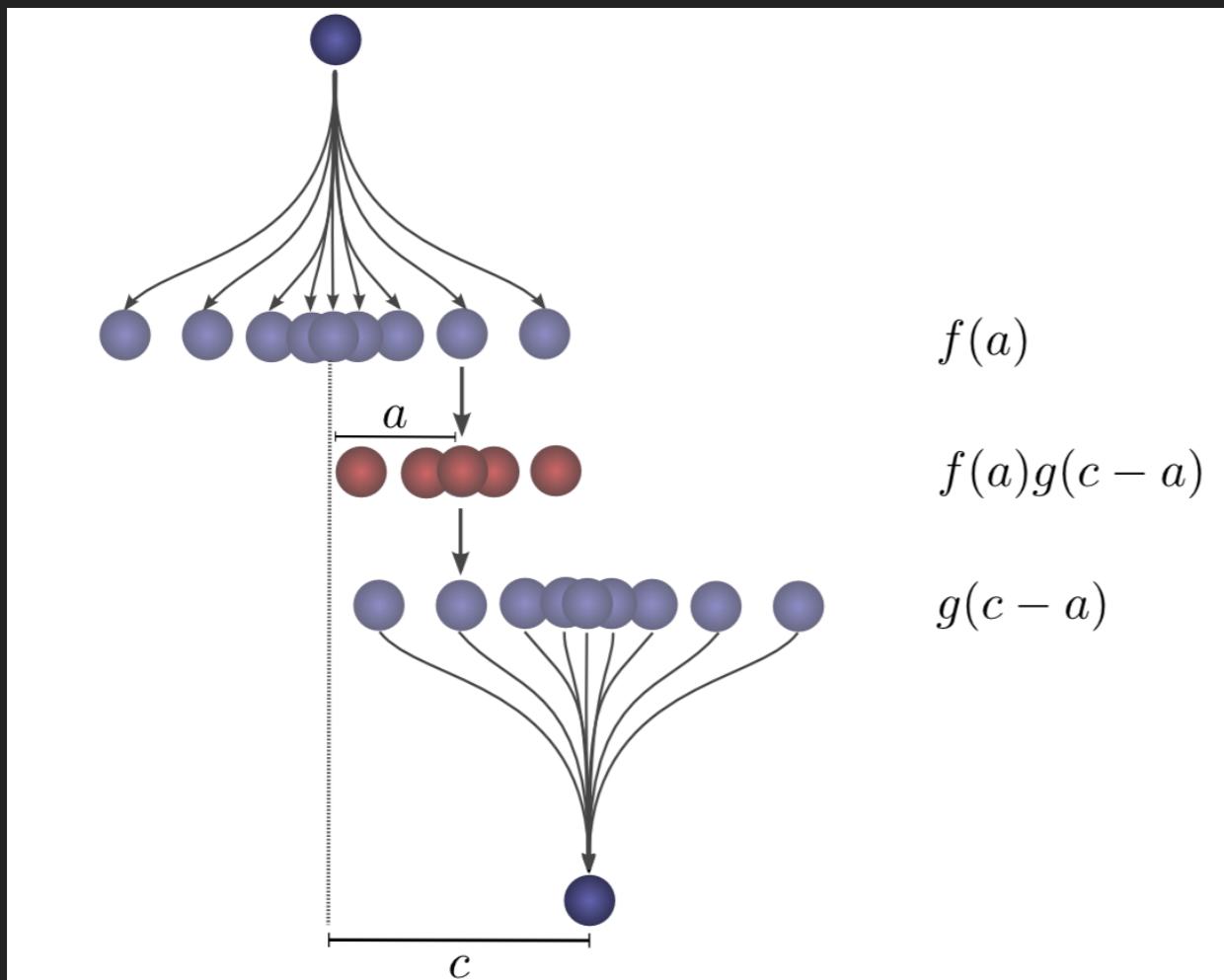
$$(f * g)(c) = \sum_{a+b=c} f(a) \cdot g(b)$$

If we substitute $b = c - a$, we get:

$$(f * g)(c) = \sum_a f(a) \cdot g(c - a)$$

This is the standard definition² of convolution.

CONVOLUTION



$$\sigma(w_0x_0 + w_1x_1 + w_2x_2 \dots + b)$$

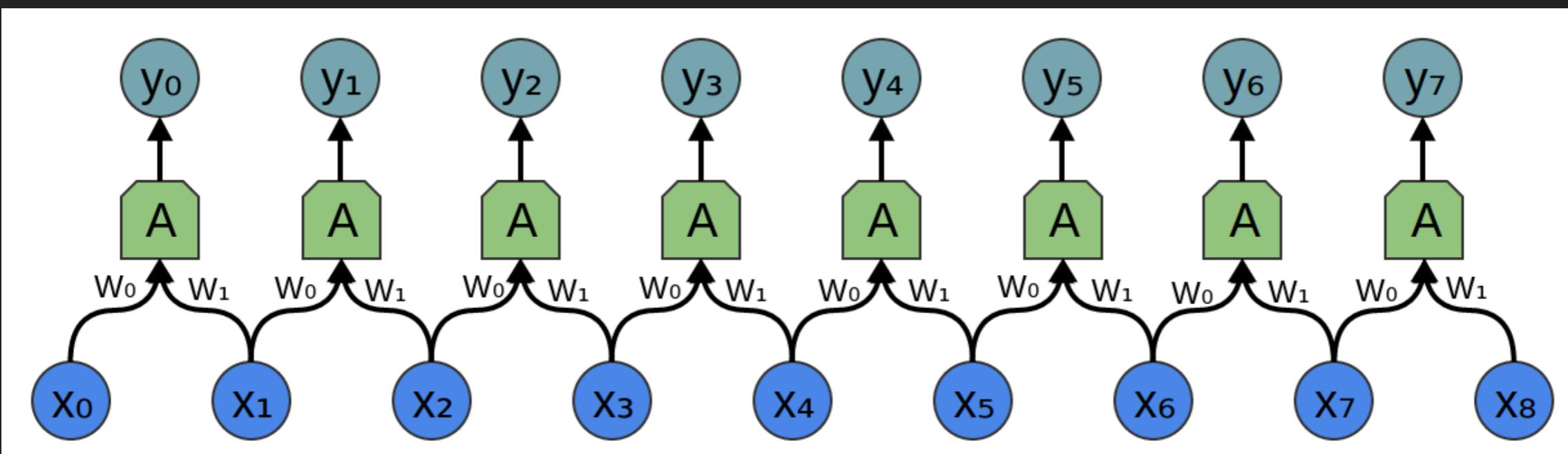
$f(a)$

$f(a)g(c-a)$

$g(c-a)$

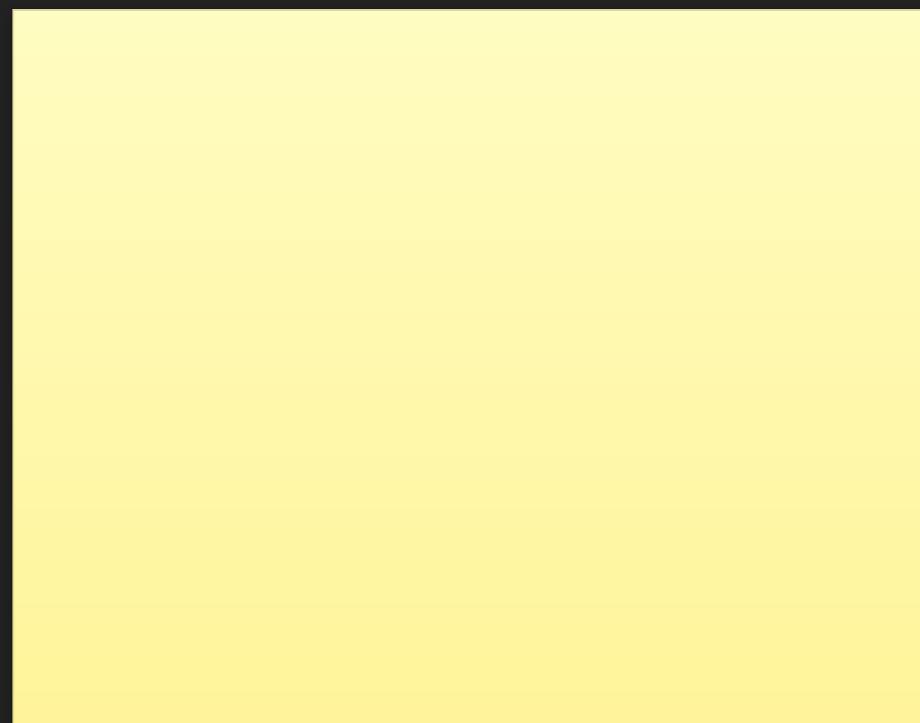
$$y_0 = \sigma(W_{0,0}x_0 + W_{0,1}x_1 + W_{0,2}x_2\dots)$$

$$y_1 = \sigma(W_{1,0}x_0 + W_{1,1}x_1 + W_{1,2}x_2\dots)$$



결론

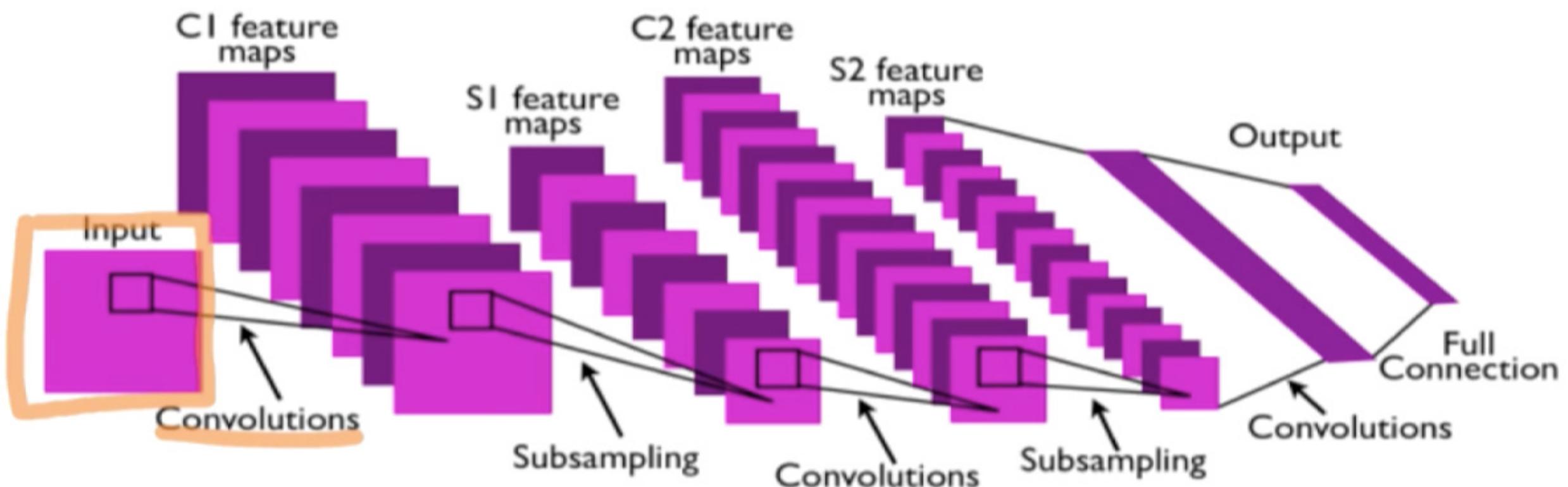
- ▶ 네트워크의 배선을 설명하는 데 매우 강력한 언어
- ▶ 주위 값들을 반영해 중앙의 값을 변화시키는 것
: 그럼 어떤 쪽으로 변화? 바로 목적하는 작업의 성공률이 높도록 이미지를 변형.



그럼 다시 CNN으로 돌아가자.

CNN

Convolutional Neural Network



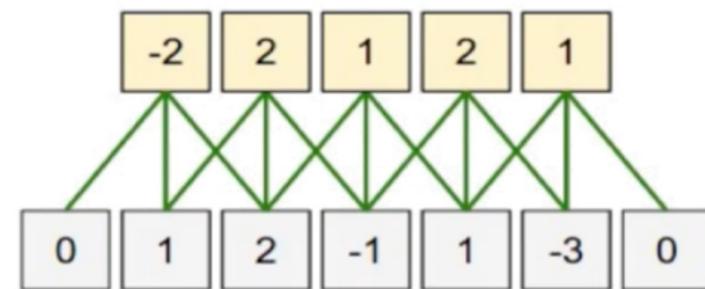
Convolutions + Subsampling + Full Connection

ZERO-PADDING

중간에서 하는 컨벌루션은 문제 없으나, 맨 처음에는 컨벌루션을 할 수 없을 때,

이 때 0을 넣어서 컨벌루션할 수 있도록 만들어 주는 것을 말한다.

Zero-padding



What is the size of the input?

$$n_{in} = 5$$

What is the size of the output?

$$n_{out} = 5$$

What is the size of the filter?

$$n_{filter} = 3$$

What is the size of the zero-padding?

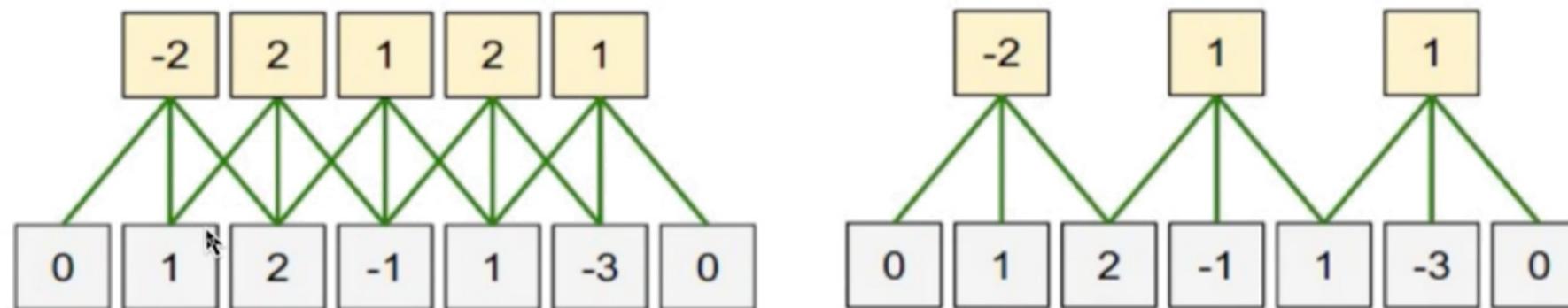
$$n_{padding} = 1$$

$$n_{out} = (n_{in} + 2 * n_{padding} - n_{filter}) + 1 \quad 5 = (5 + 2 * 1 - 3) + 1$$

STRIDE

점프뛰기를 연상합니다.

Stride



(Left) Stride size: 1

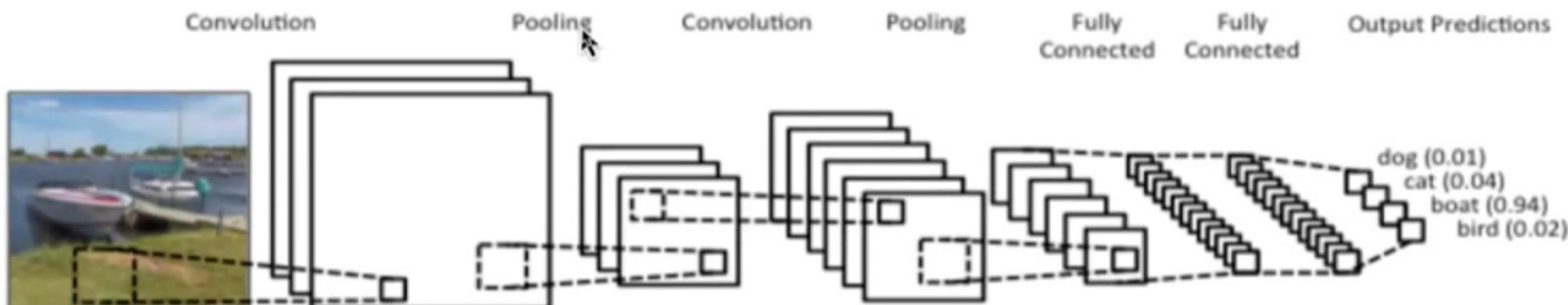
(Right) Stride size: 2

If stride size equals the filter size, there will be
no overlapping.

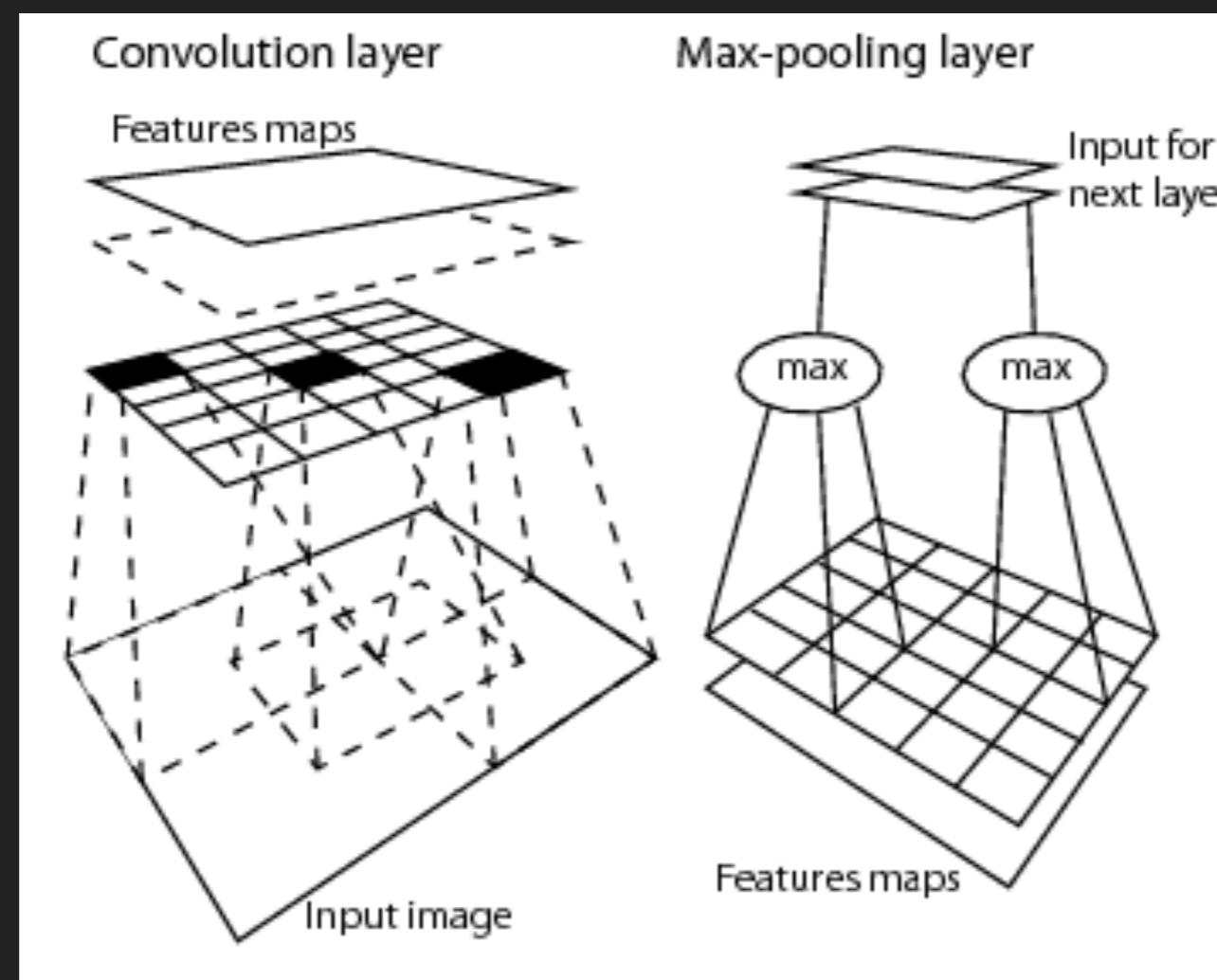
CNN

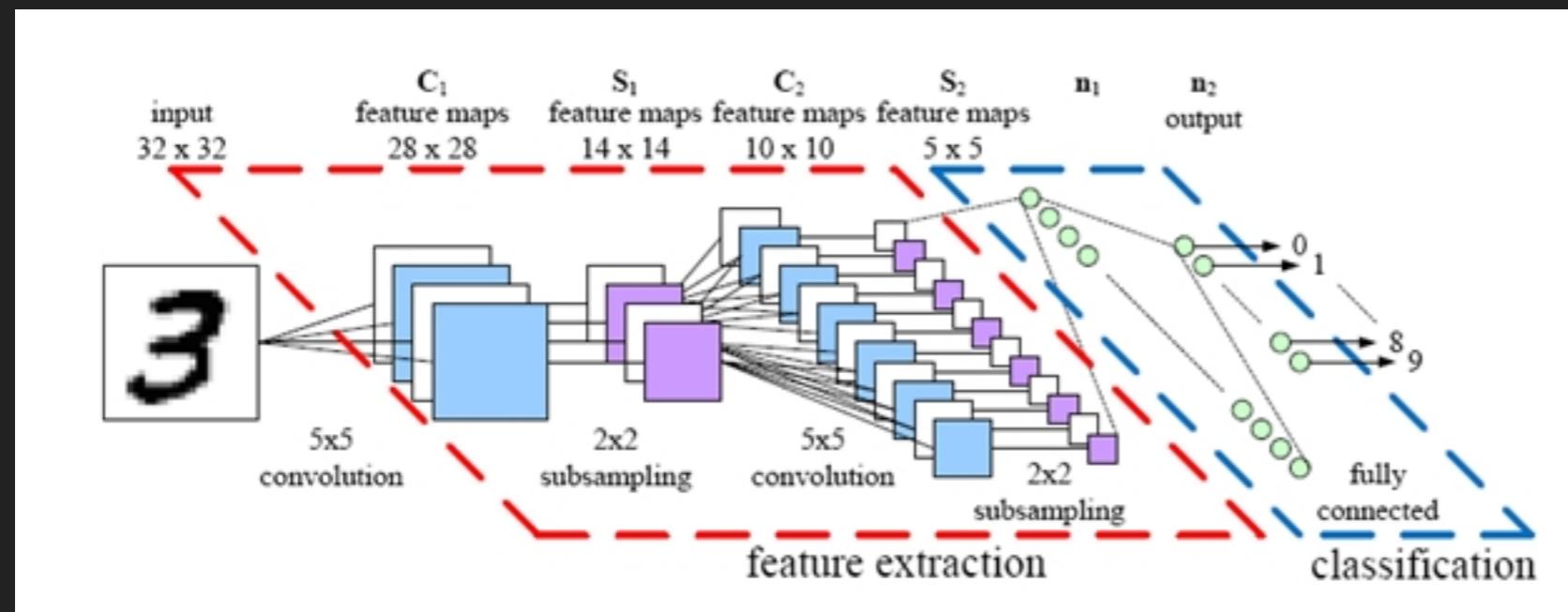
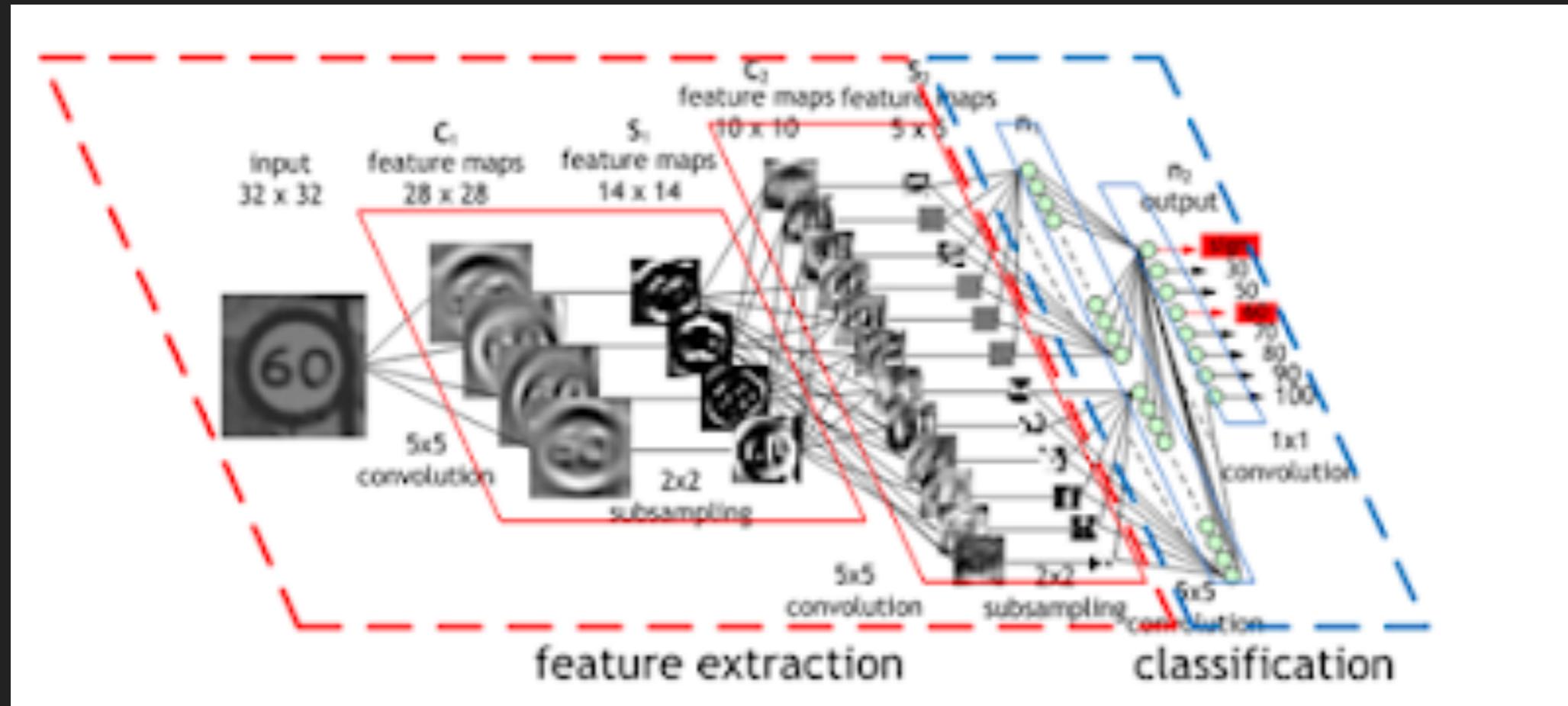
CNNs are basically layers of **convolutions** followed by **subsampling** and **fully connected layers**.

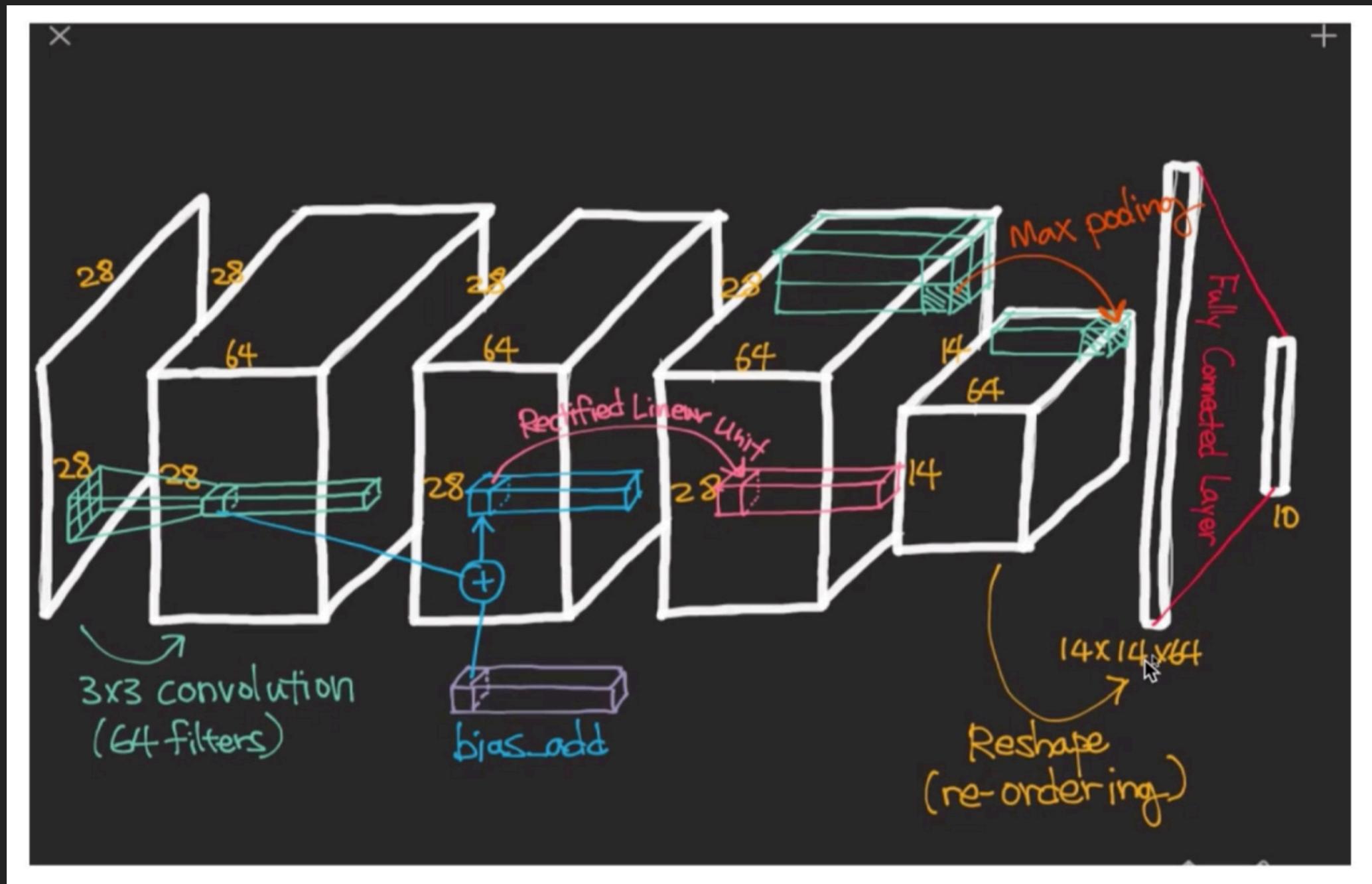
Intuitively speaking, **convolutions** and **subsampling** layers work as **feature extraction** layers while a **fully connected layer** **classifies** which category current input belongs to using extracted features.



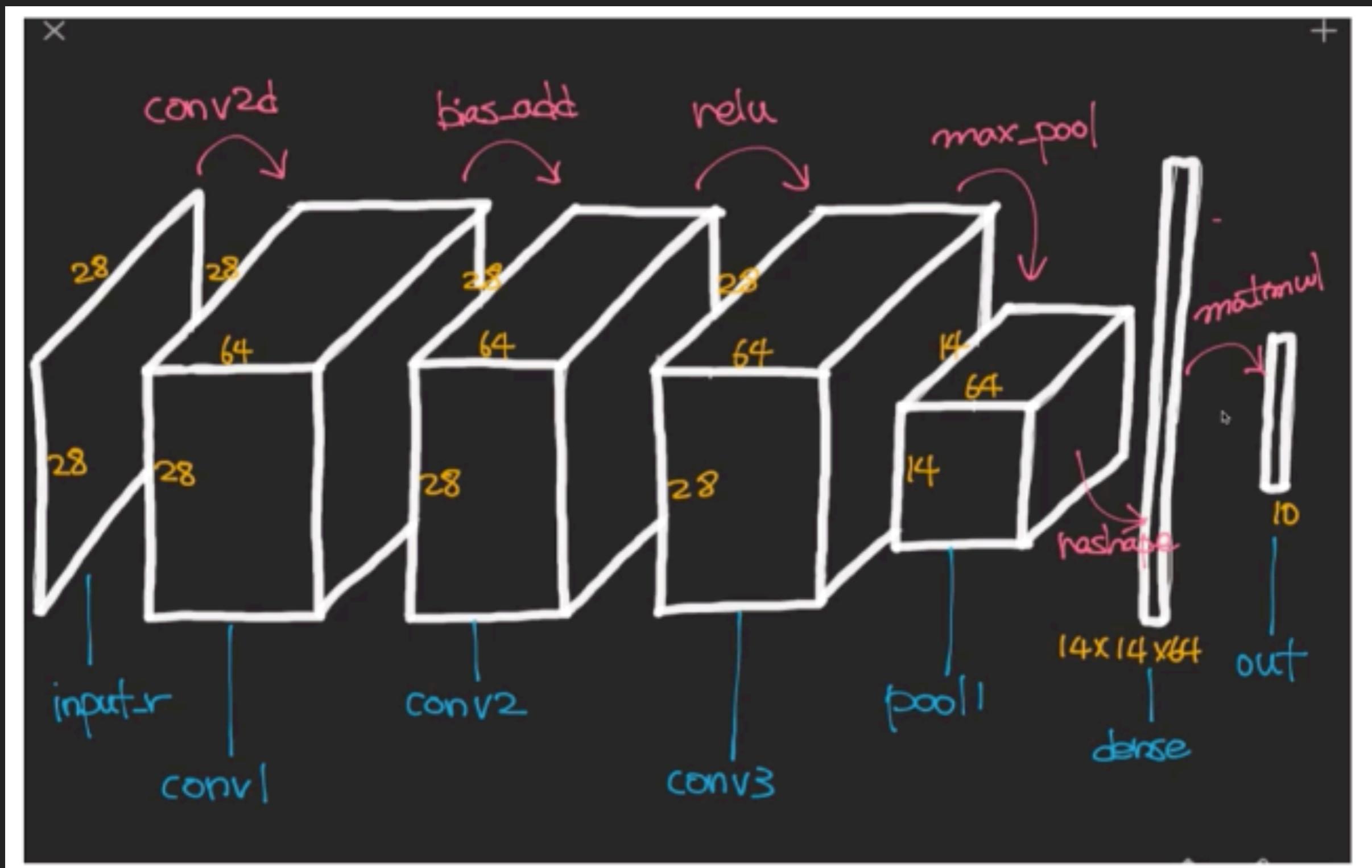
SUBSAMPLING / POOLING

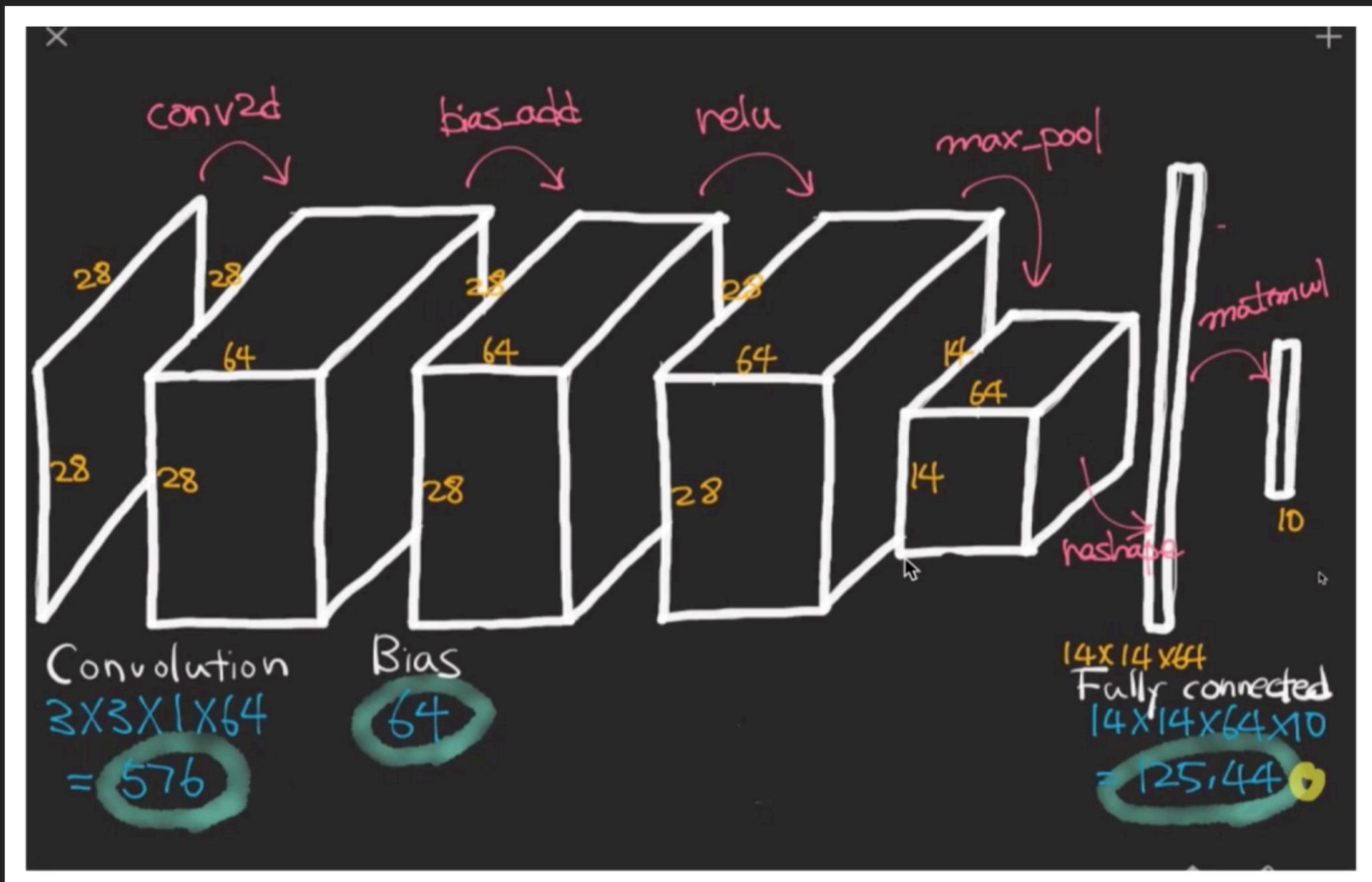






CNN





4가지 CNN 살펴보기

- ▶ AlexNET
- ▶ VGG
- ▶ GooLeNet
- ▶ ResNet