

# | Convolution Neural Network 2

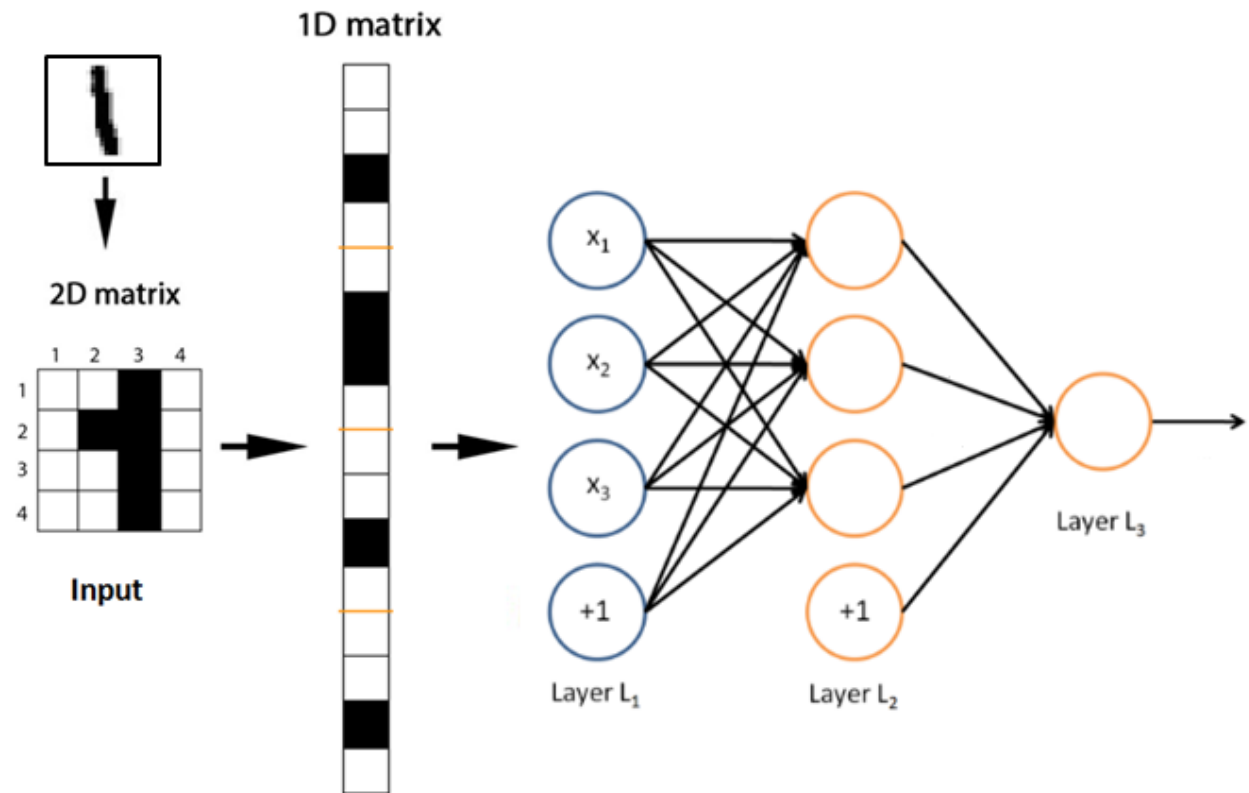


명지대학교  
MYONGJI UNIVERSITY

# CNN

## CNN으로 학습을 해야 하는 이유?

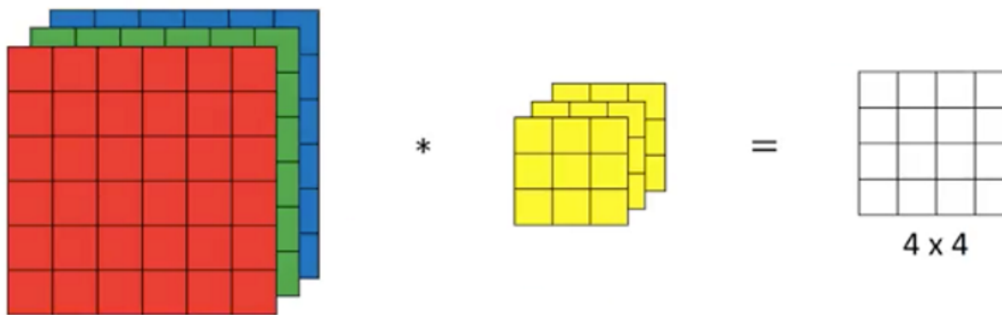
- 이미지와 같은 2D 이상의 차원을 갖는 데이터는 MLP(다층 퍼셉트론)으로 데이터 속성 자체를 그대로 처리 하지 못함



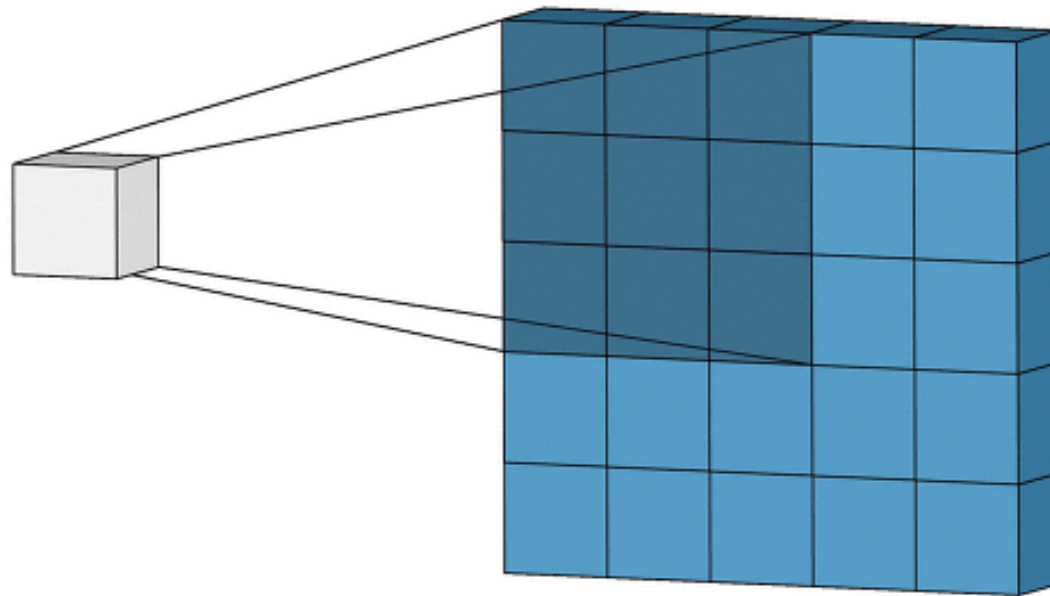
# CNN

- 컬러 이미지의 경우 가로, 세로 말고 '채널' (R,G,B) 세개의 채널 존재
- 이미지의 공간 정보란 말 그대로 이미지가 가지는 '공간적인 정보'

Convolutions on RGB image



# 합성곱(Convolution)



# 패딩

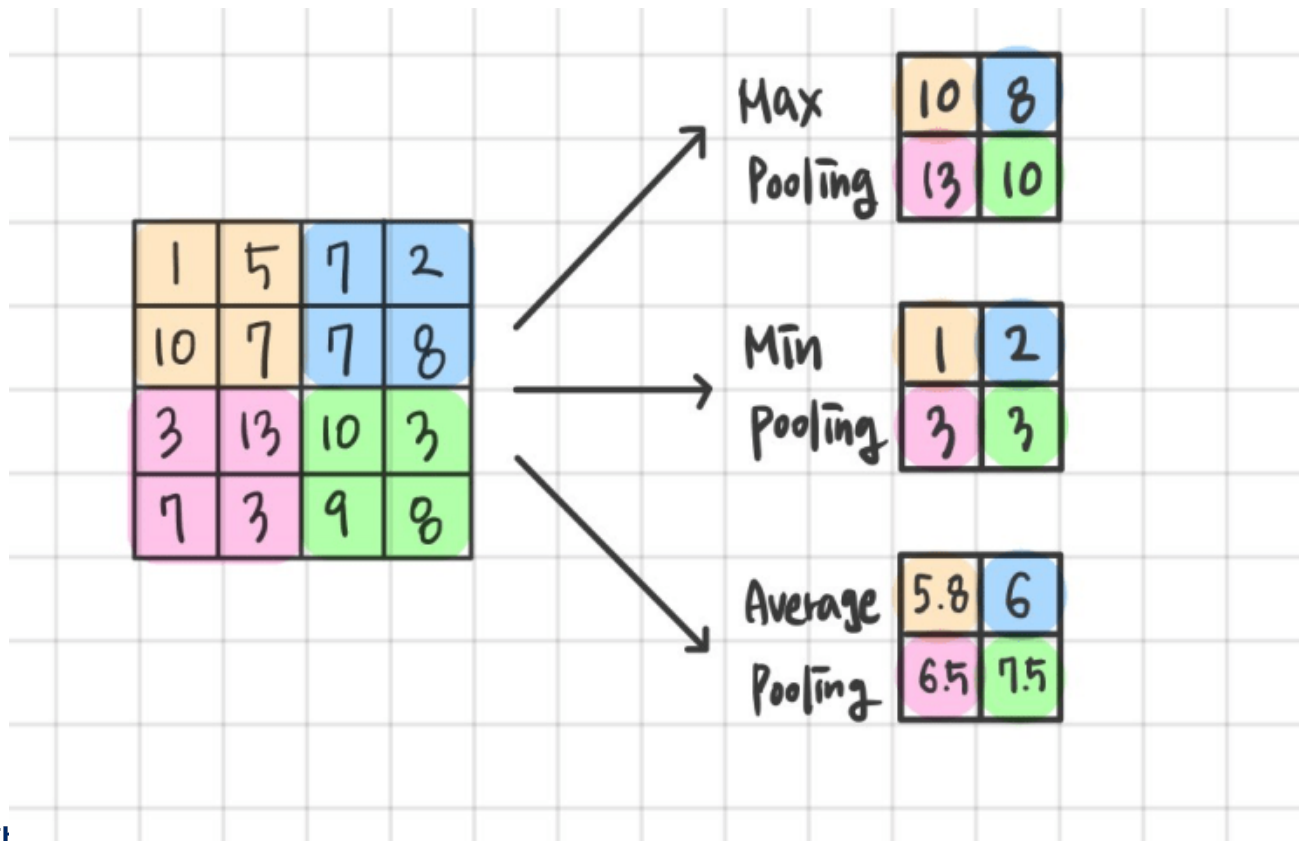
- 패딩이란 값을 0과 같은 임의의 값으로 채워서 텐서의 사이즈를 유지하거나 키우는 기법

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

Image  
(+zero Padding)

# 풀링

패딩과는 다르게 이미지의 크기를 줄이는 방법



# 이미지 사이즈

## ■ 식1. 출력 데이터 크기 계산

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

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

- 입력 이미지 높이 : H
- 입력 이미지 너비 : W
- 필터 높이 : FH
- 필터 너비 : FW
- Stride 크기 : S
- 패딩 사이즈 : P

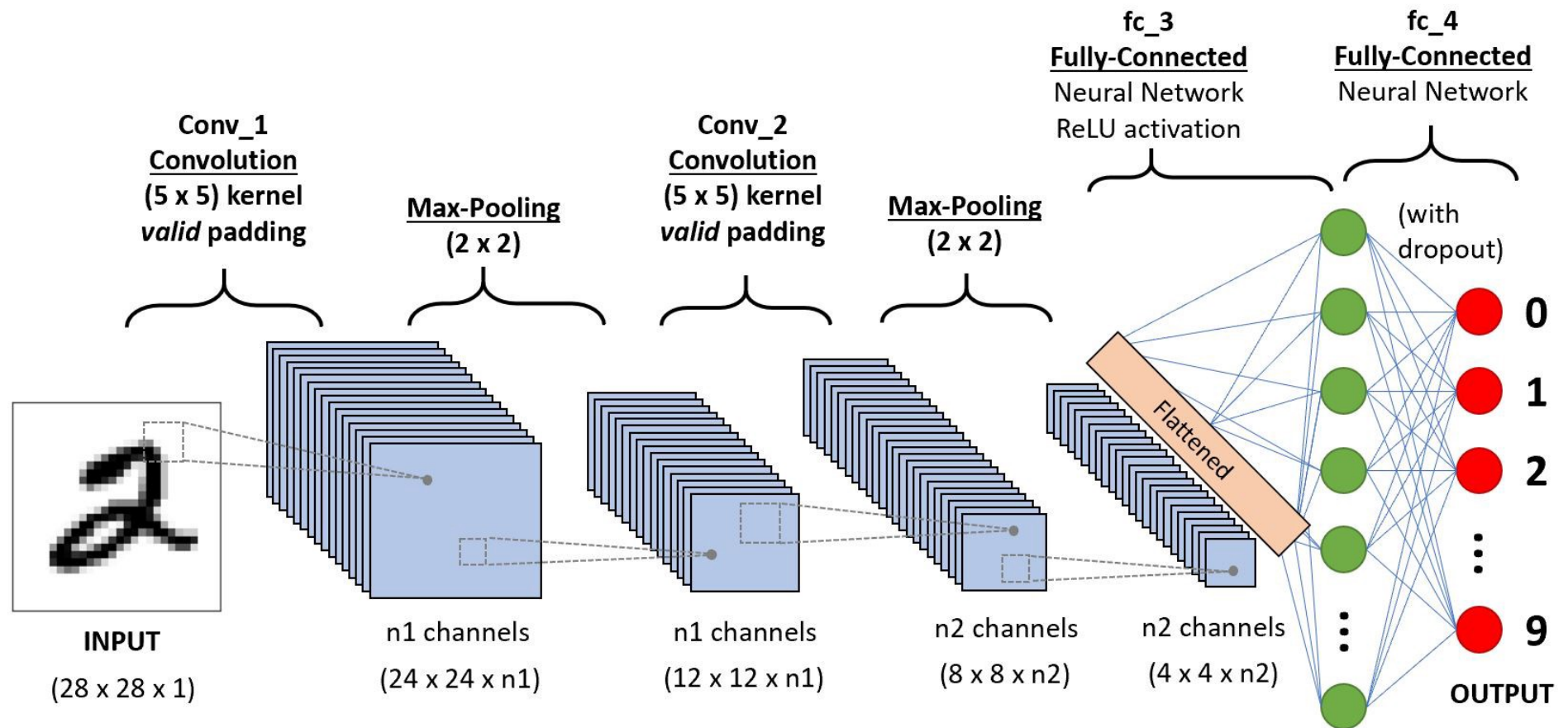
1 <sub>x1</sub>	1 <sub>x0</sub>	1 <sub>x1</sub>	0	0
0 <sub>x0</sub>	1 <sub>x1</sub>	1 <sub>x0</sub>	1	0
0 <sub>x1</sub>	0 <sub>x0</sub>	1 <sub>x1</sub>	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

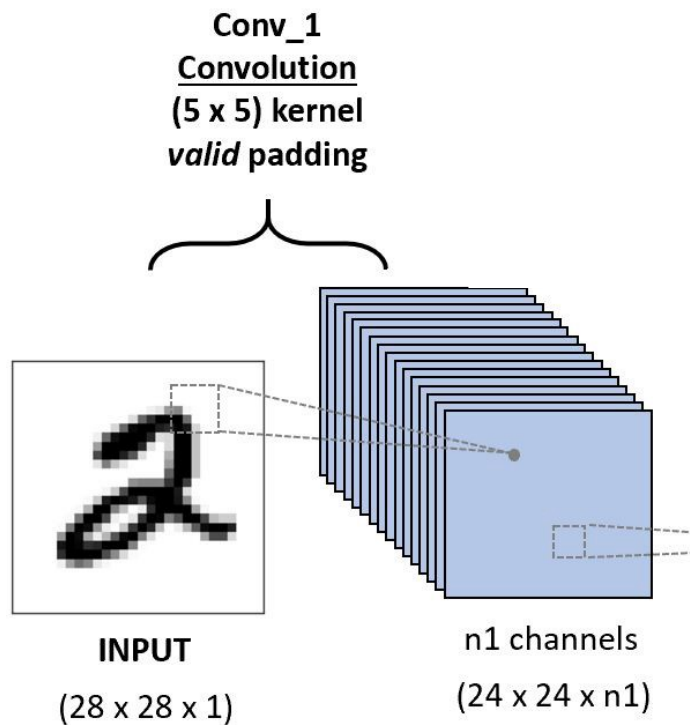
Convolved  
Feature

# 합성곱 신경망 (Convolutional Neural Network)





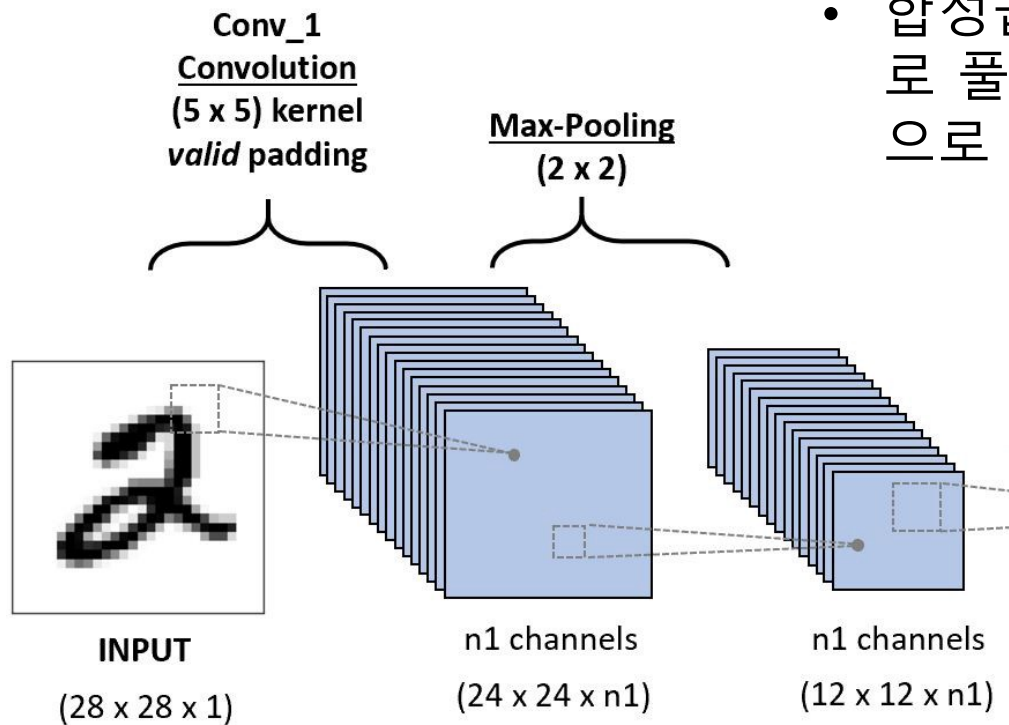
# 합성곱 신경망 (Convolutional Neural Network)



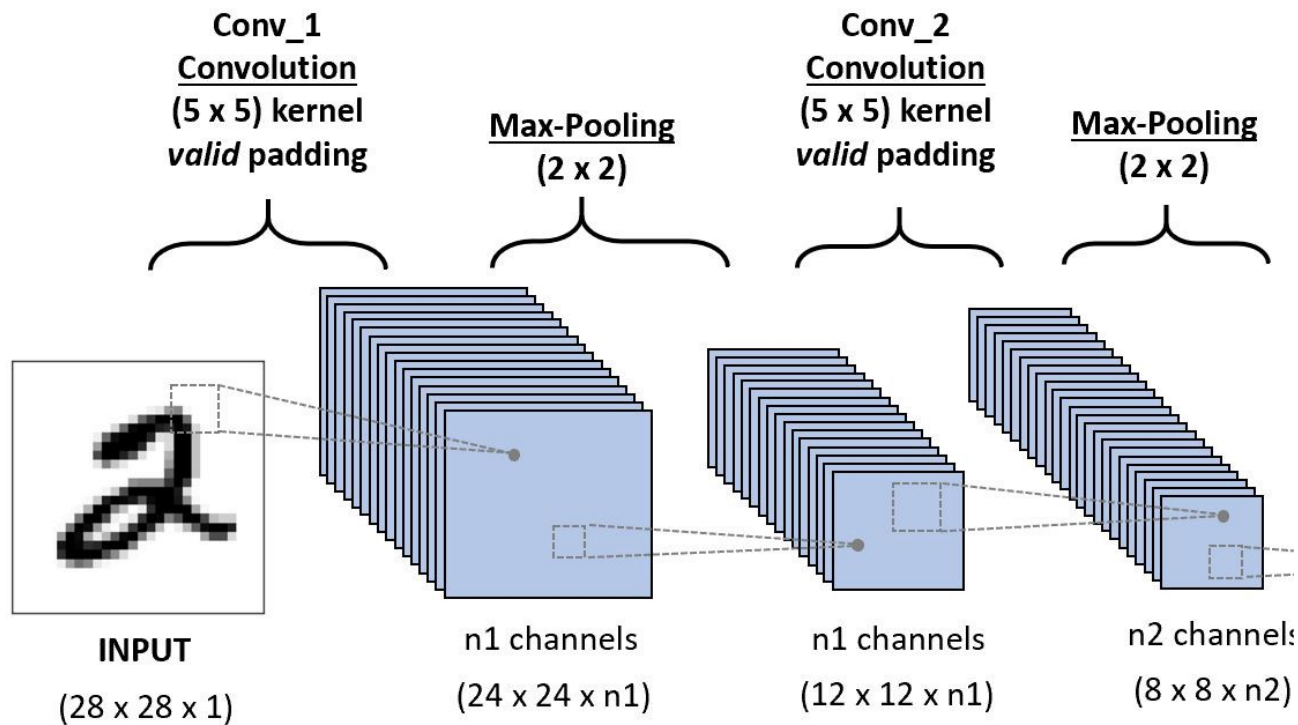
- 가장 먼저 28 x 28 x 1의 흑백 입력이미지에 5x5의 필터를 합성곱
- 패딩은 0, 스트라이드는 1, 필터사이즈는 5이기 때문에 이미지 사이즈는  $28-5+1=24$
- 하나의 필터를 합성곱하면 하나의 출력 이미지가 나오므로 여러장의 필터(5x5xn1)를 만들어서 여러장의 이미지(n1장, 24x24xn1)로 만들기

# 합성곱 신경망 (Convolutional Neural Network)

- 합성곱 이후에 이미지의 크기가 너무 크므로 풀링을 수행해서 이미지의 크기를 절반으로 줄이기 ( $12 \times 12 \times n1$ )



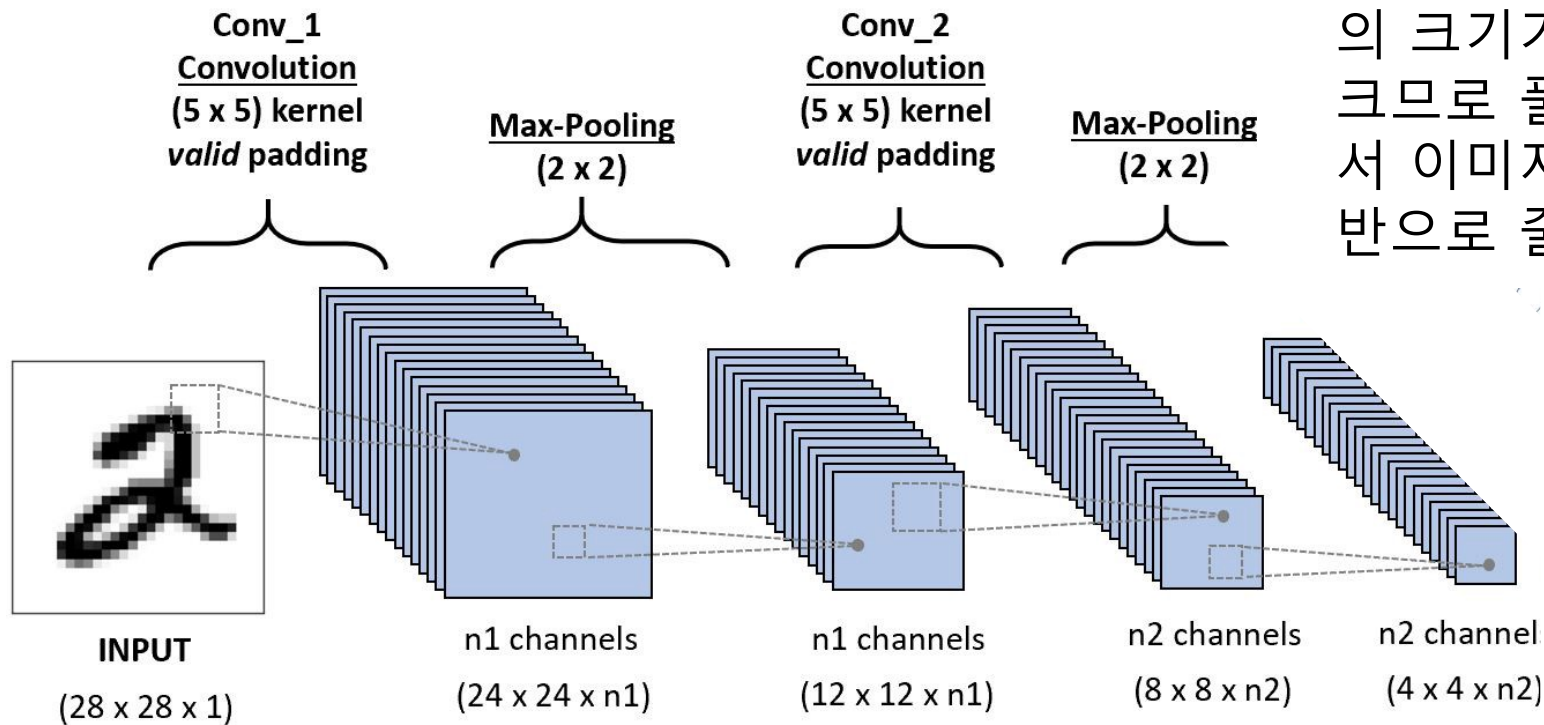
# 합성곱 신경망 (Convolutional Neural Network)



- 다시 합성곱을 수행
- $12 \times 12 \times n1$ 장의 이미지들에  $5 \times 5$ 필터를 합성곱합니다. 패딩은 0, 스트라이드는 1, 필터사이즈는 5이기 때문에 이미지의 사이즈는  $12 - 5 + 1 = 8$
- $n2$  장의 필터를 사용해서 이미지 사이즈를  $8 \times 8 \times n2$ 로 만들기

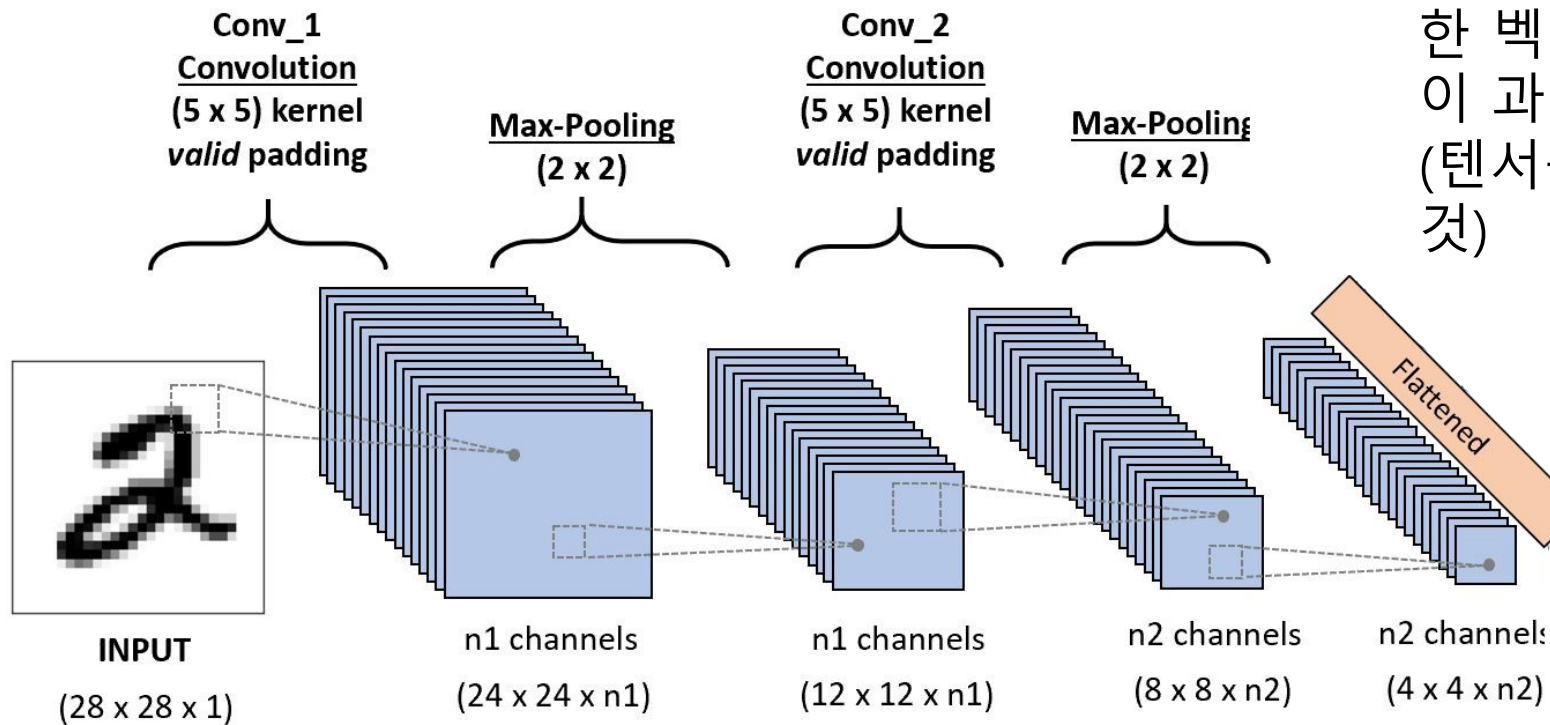
# 합성곱 신경망 (Convolutional Neural Network)

- 합성곱 이후에 이미지의 크기가 여전히 너무 크므로 풀링을 수행해서 이미지의 크기를 절반으로 줄이기( $4 \times 4 \times n_2$ )



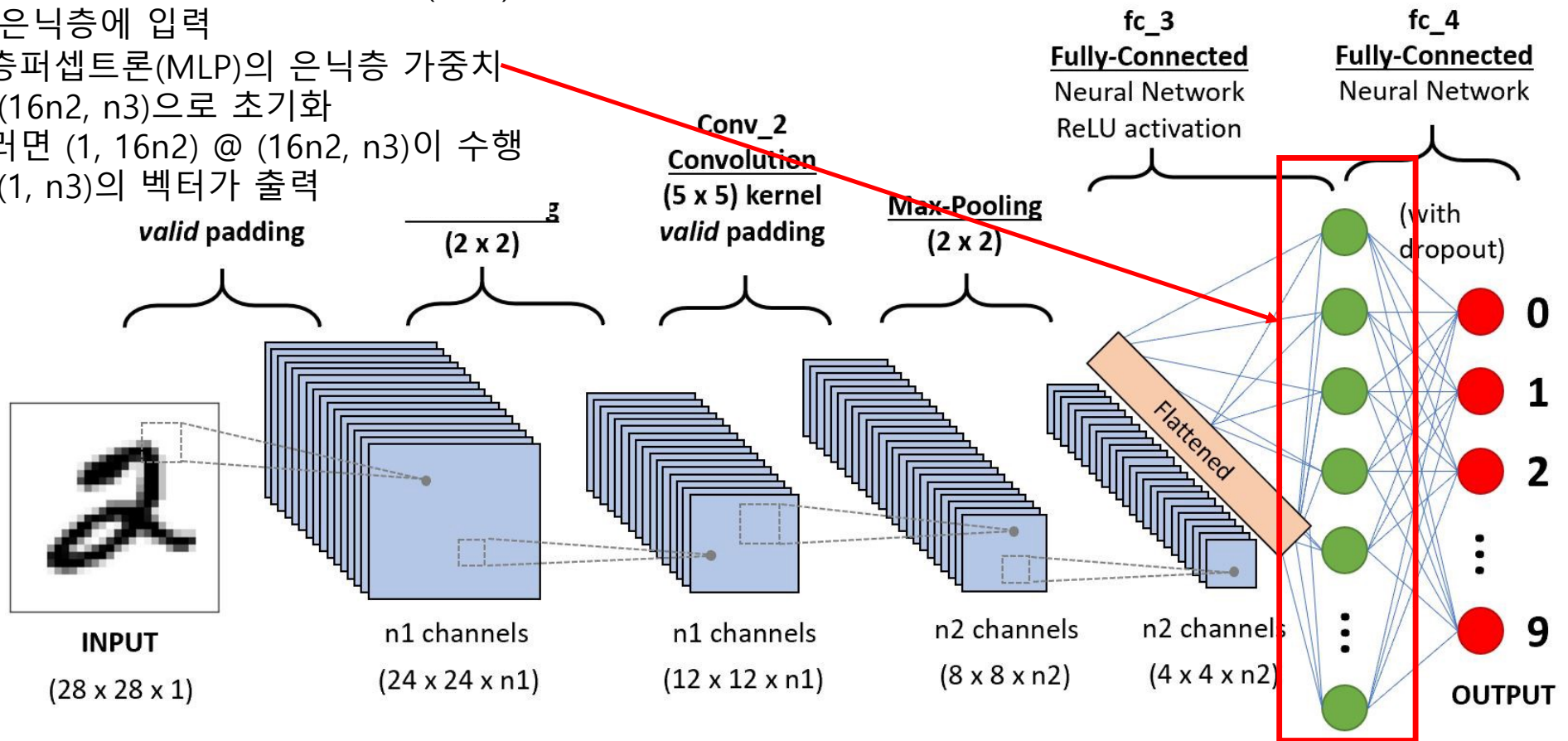
# 합성곱 신경망 (Convolutional Neural Network)

- $4 \times 4 \times n_2$ 의 텐서를 reshape( $1, 16n_2$ )하면 길쭉한 벡터( $1 \times 16n_2$ ) 생성이 과정을 flatten 과정 (텐서를 벡터로 쭉 펴는 것)



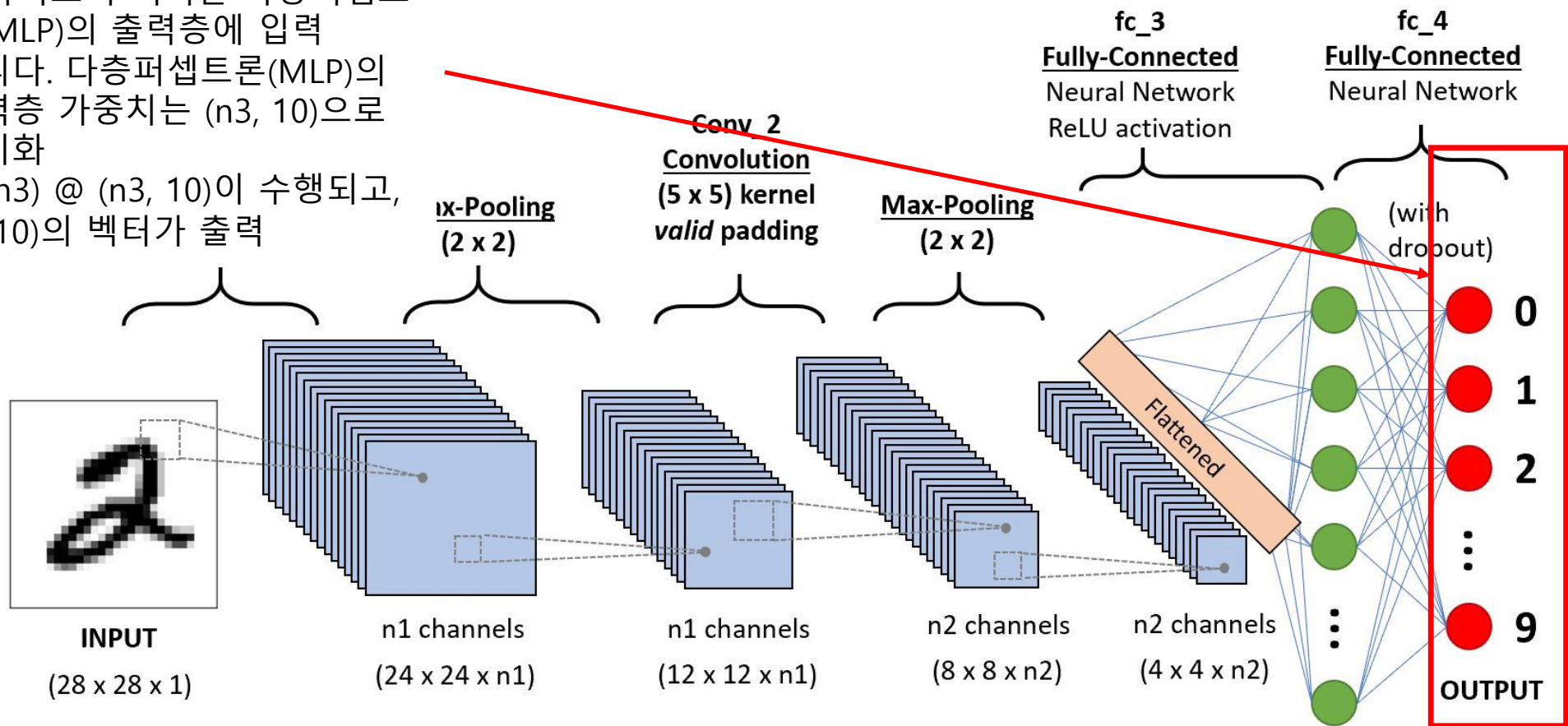
# 합성곱 신경망 (Convolutional Neural Network)

- 이 flatten 벡터를 다층퍼셉트론(MLP)의 은닉층에 입력
- 다층퍼셉트론(MLP)의 은닉층 가중치는  $(16n_2, n_3)$ 으로 초기화
- 그러면  $(1, 16n_2) @ (16n_2, n_3)$ 이 수행  
→  $(1, n_3)$ 의 벡터가 출력



# 합성곱 신경망 (Convolutional Neural Network)

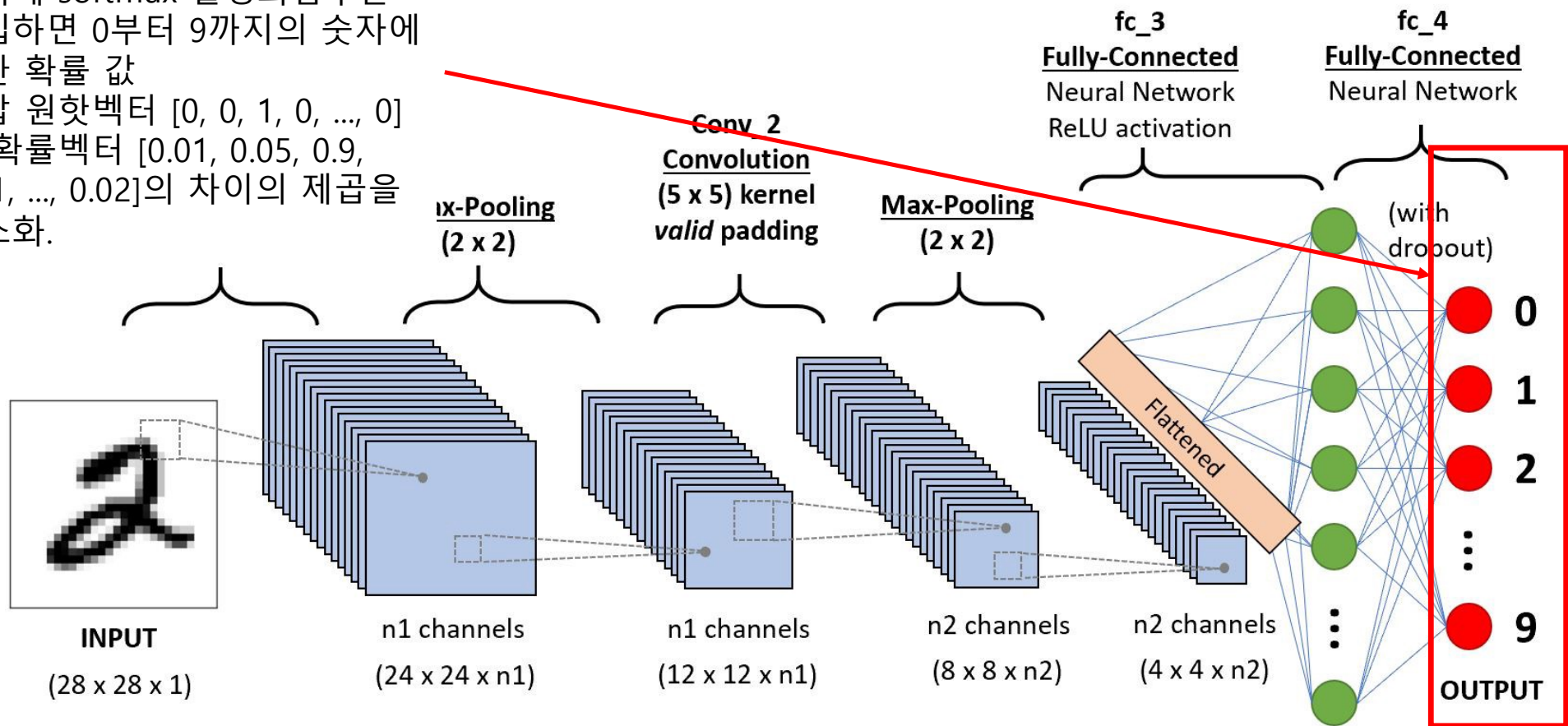
- $n_3$  사이즈의 벡터를 다층퍼셉트론(MLP)의 출력층에 입력
- 합니다. 다층퍼셉트론(MLP)의 출력층 가중치는  $(n_3, 10)$ 으로 초기화
- $(1, n_3) @ (n_3, 10)$ 이 수행되고,  $(1, 10)$ 의 벡터가 출력





# 합성곱 신경망 (Convolutional Neural Network)

- 여기에 softmax 활성화함수를 도입하면 0부터 9까지의 숫자에 대한 확률 값
- 정답 원핫벡터  $[0, 0, 1, 0, \dots, 0]$ 와 확률벡터  $[0.01, 0.05, 0.9, 0.01, \dots, 0.02]$ 의 차이의 제곱을 최소화.

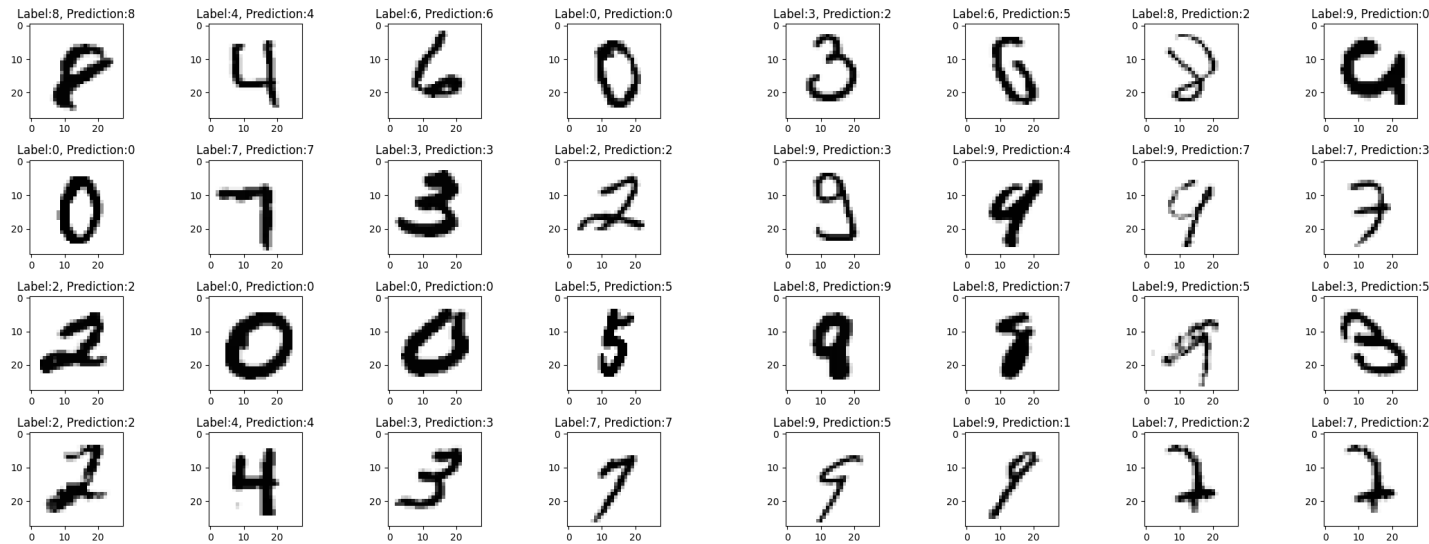




# 합성곱 신경망 (Convolutional Neural Network)

- 사용되는 필터도 역시 가중치이며, 랜덤하게 초기화
- 그리고 학습을 진행하면서 필터의 값들도 역전파 알고리즘에 의해 학습
- 즉, 합성곱층, 풀링층 모두 역전파 알고리즘에 의해 기울기가 계산되며 합성곱층의 필터는 학습되어 계속적으로 변화

# 합성곱 신경망 (Convolutional Neural Network)



잘 예측 한 예

잘 예측하지 못한 예

