

## 7. CNN

---

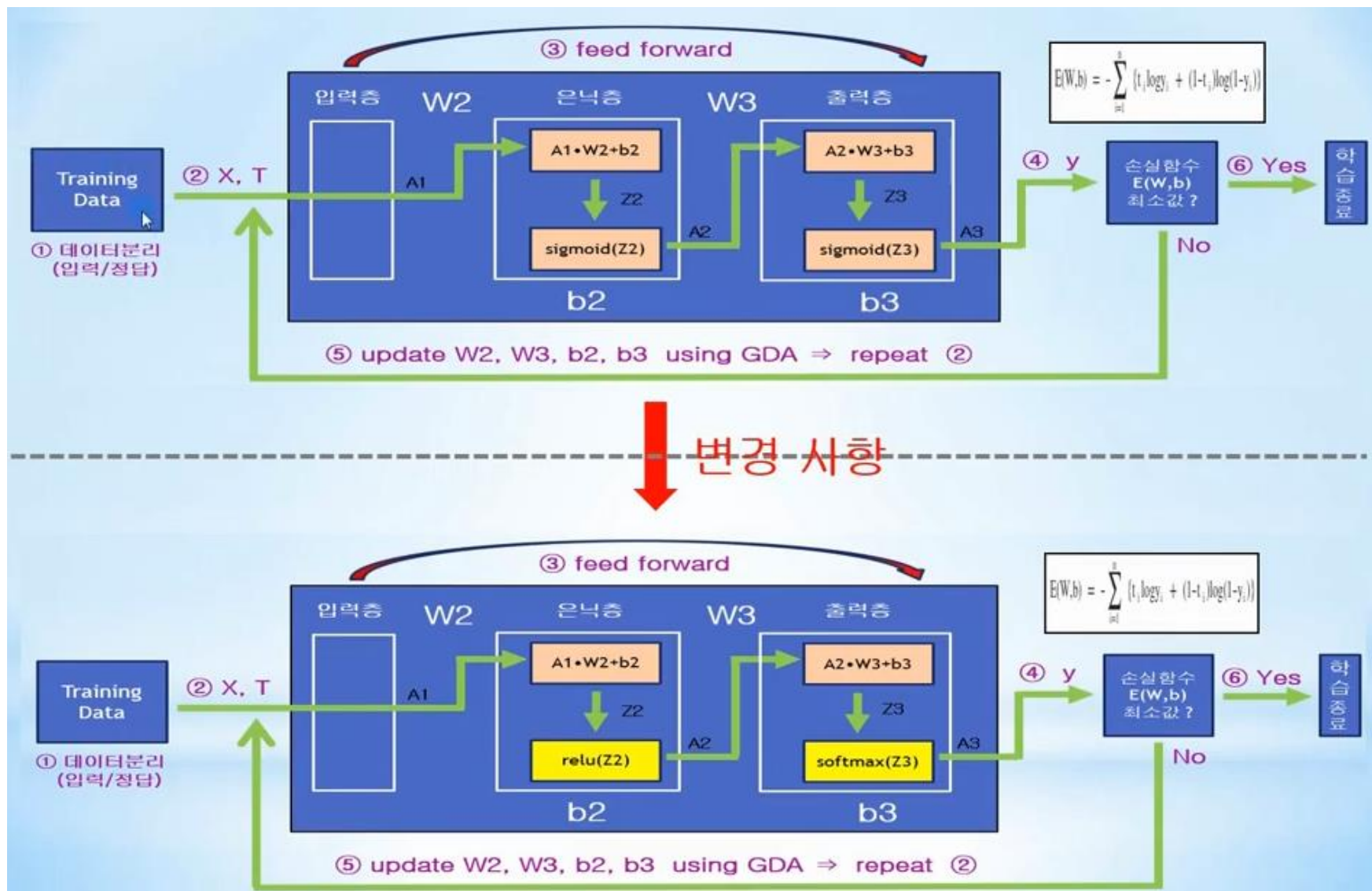
박경미

# 목차

---

- ❖ tensorflow Neural Network
- ❖ relu/ one-hot encoding/softmax
- ❖ 컨보루션 연산. 풀링. 패딩
- ❖ 필터를 통해 데이터 특징을 추출하는 원리

# tensorflow Neural Network



# relu/ one-hot encoding/softmax

은닉층

A1•W2+b2

↓ Z2

sigmoid(Z2)

은닉층

A1•W2+b2

↓ Z2

relu(Z2)

출력층

A2•W3+b3

↓ Z3

sigmoid(Z3)

출력층

A2•W3+b3

↓ Z3


softmax(Z3)

sigmoid 출력값: 0~1

relu 출력값: max(0, Z2)


Sigmoid

$\sigma(x) = \frac{1}{1+e^{-x}}$



ReLU

$\max(0, x)$



[머신러닝 강의 19] 딥러닝 (I) 참조

sigmoid 출력값: 0~1

softmax 출력값  $\frac{\exp(z_i^{(3)})}{\sum_i \exp(z_i^{(3)})}$

index	0	1	2	3	4	5	6	7	8	9
Z3	0.1	0.5	1.0	1.5	0.1	2.1	1.1	0.9	1.8	0.2

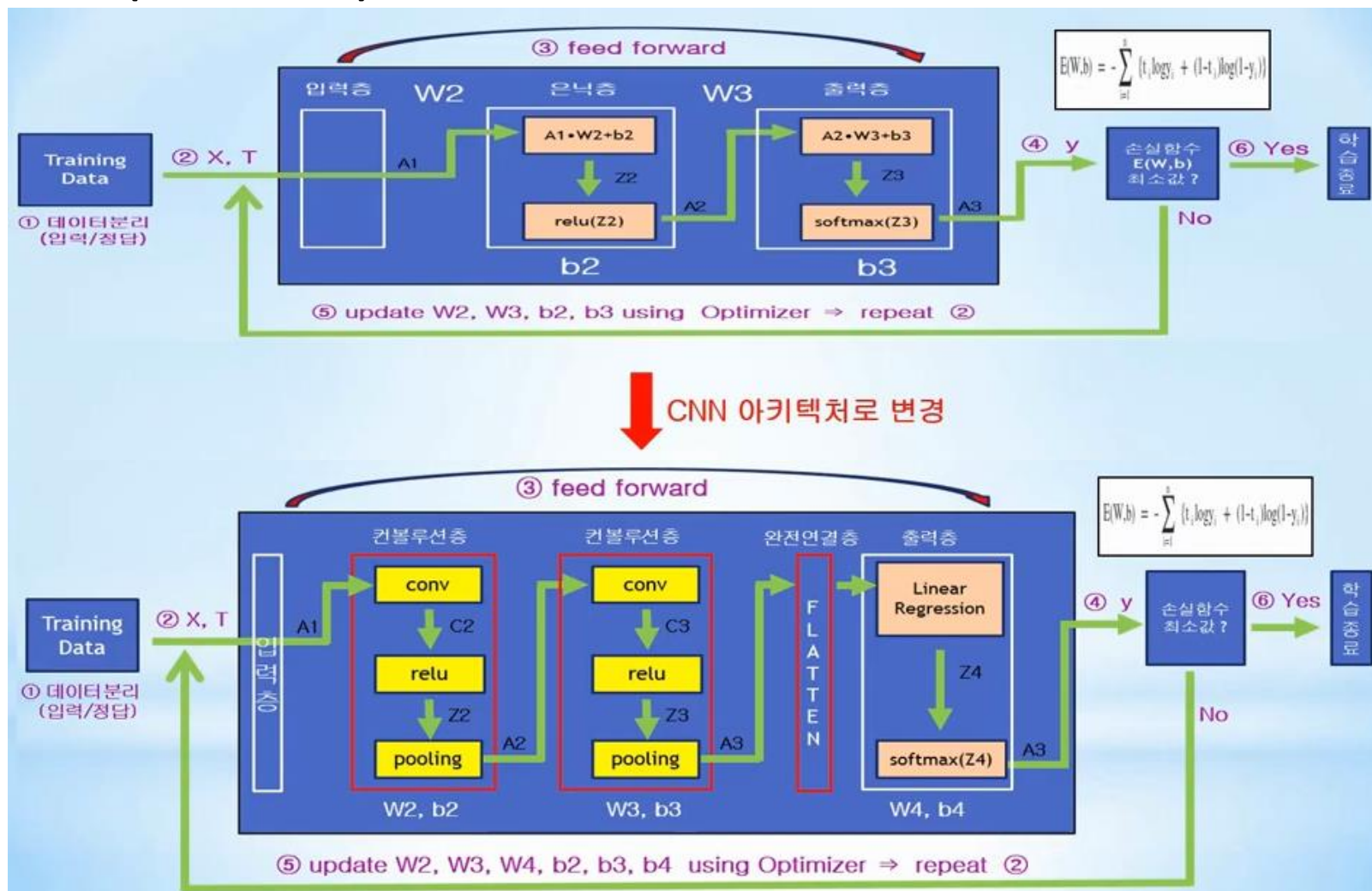
softmax

index	0	1	2	3	4	5	6	7	8	9
softmax(Z3)	0.03	0.05	0.09	0.14	0.03	0.26	0.09	0.08	0.19	0.04

softmax 함수 출력은 0 과 1사이 값이며 모두 더하면 1이 됨.  
즉, 출력 총합이 1이라는 것은 각각의 값이 정답일 확률을 나타낸다고 해석할 수 있음.  
[예] 인덱스 5의 값이 0.26, 즉 26% 확률로 가장 높기 때문에, one-hot encoding 에 의해 정답을 5 로 예측하고 있음

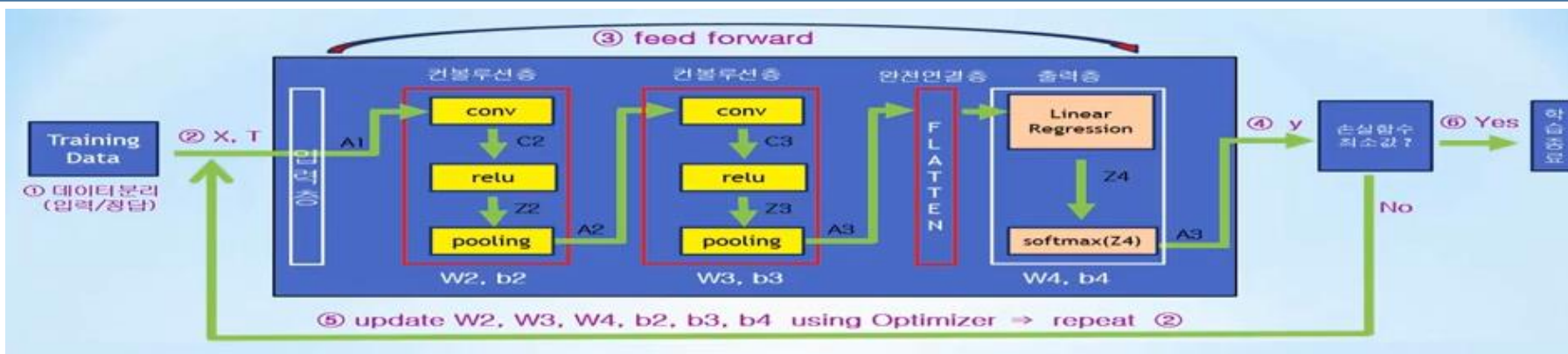
# 컨보루션 연산. 풀링. 패팅

## ❖ 아키텍처 비교(NN vs. CNN)





# 컨볼루션층 개요 conv / pooling



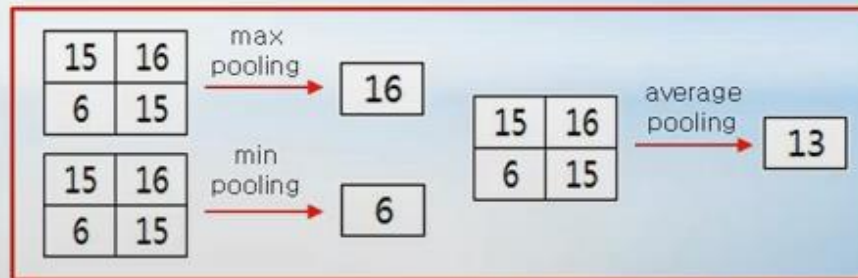
## ➤ conv (컨볼루션, convolution)

- 입력데이터(A1, A2...)와 가중치들의 집합체인 다양한 필터(filter)와의 **컨볼루션 연산**을 통해 **입력데이터의 특징(feature)**을 **추출**하는 역할을 수행함 → [다음 강의에서 자세히 설명]

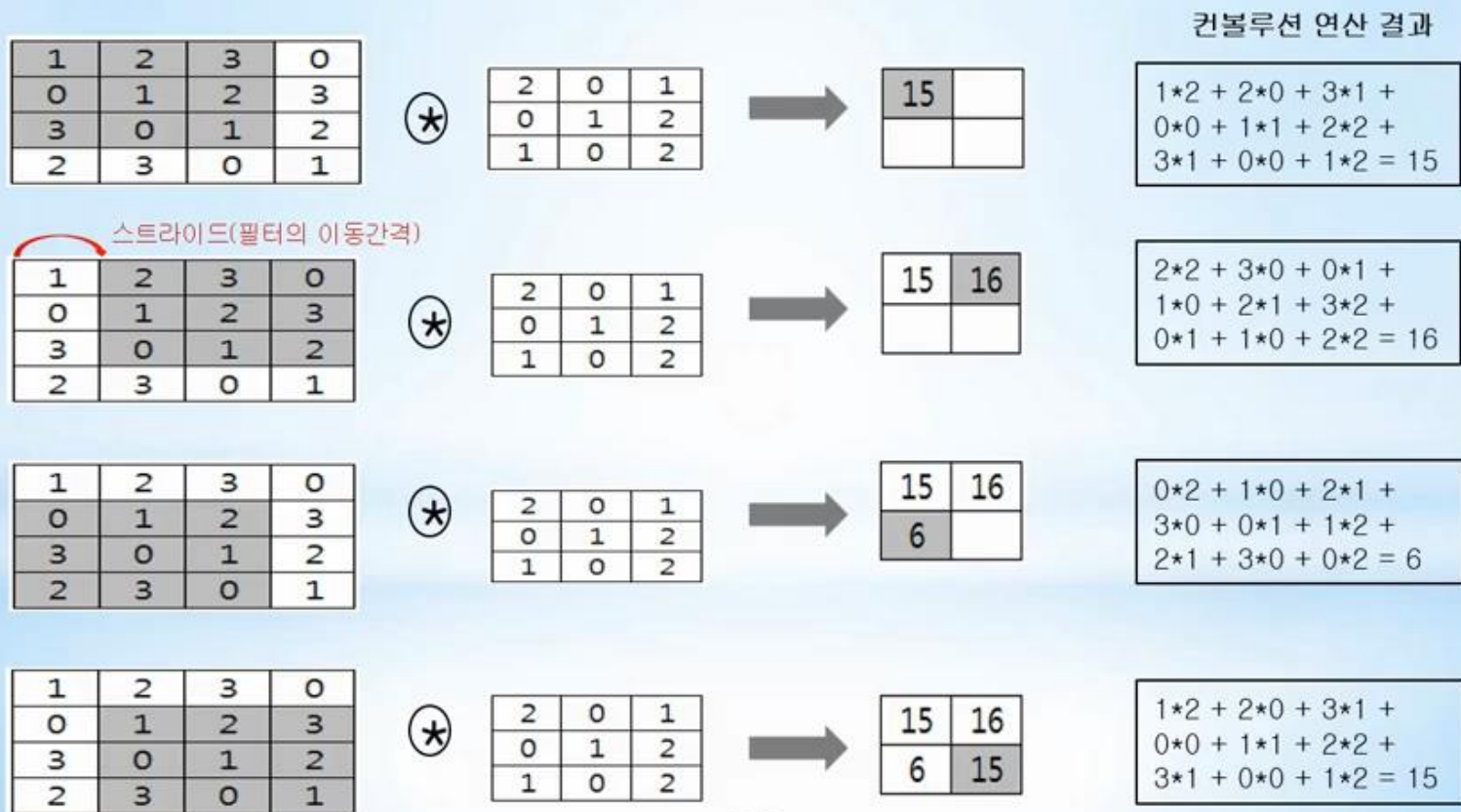
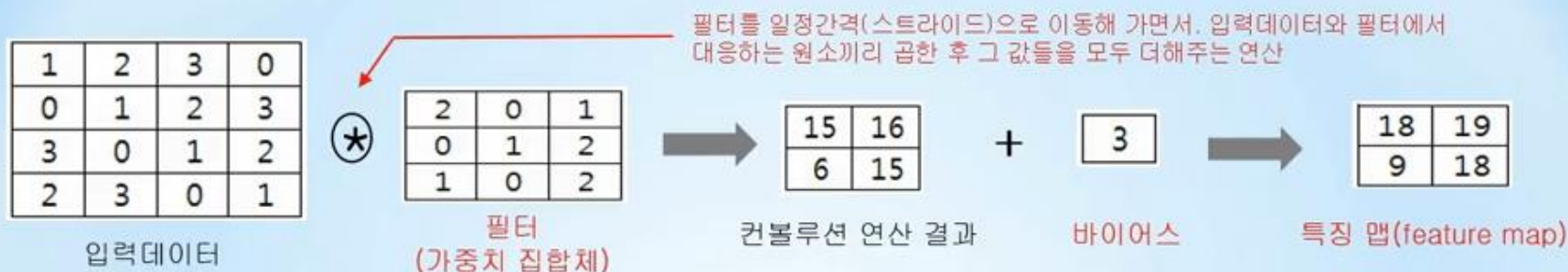
$$\begin{aligned} A1 & \otimes \text{filter}_1 + b2 \longrightarrow \text{입력데이터 A1 특징(feature) 추출} \\ A2 & \otimes \text{filter}_2 + b3 \longrightarrow \text{입력데이터 A2 특징(feature) 추출} \end{aligned}$$

## ➤ pooling (풀링)

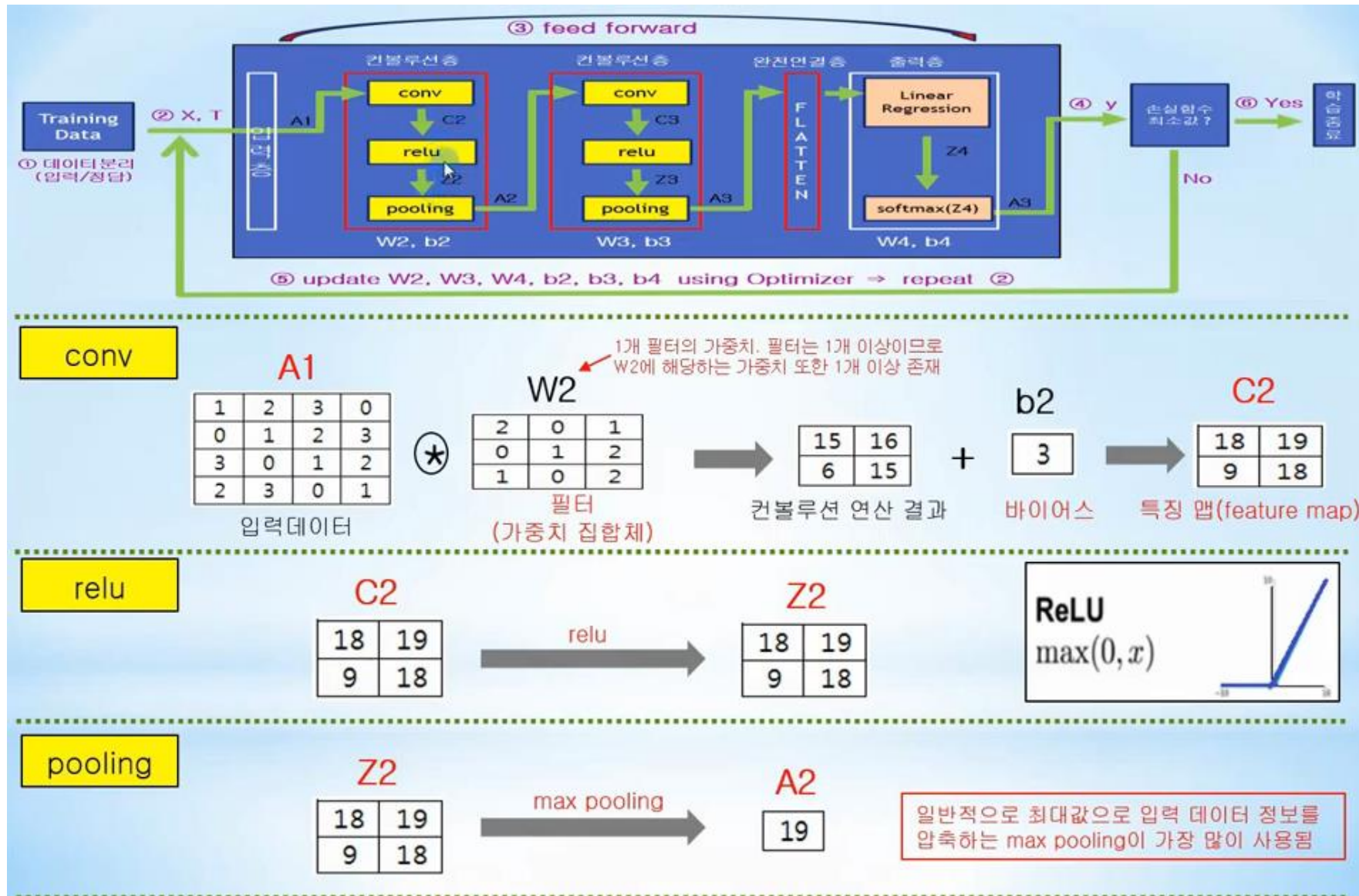
- 입력 정보를 최대값 • 최소값 • 평균값 등으로 압축하여 데이터 연산량을 줄여주는 역할 수행



# 컨볼루션(convolution) 연산- 특징 추출



# rel 연산 / pooling 연산





# 패딩(Padding)

- ❖ 컨볼루션 연산을 수행하기 전에 **입력 데이터 주변을 특정 값(예를 들면 0)으로 채우는 것**, 컨볼루션 연산에서 자주 이용되는 방법
  - 컨볼루션 연산을 수행하면 데이터 크기(shape)이 줄어드는 단점을 방지



# 컨볼루션 연산을 통한 출력 데이터 크기 계산

❖ 입력 데이터 크기(H,W), 필터 크기(FH,FW), 패딩 P, 스트라이드 s일 때 **출력 데이터 크기 (OH, OW)**

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

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

[예1] 입력 (4, 4), 필터 (3, 3), 패딩 1, 스트라이드 1 ⇒ 출력 (4, 4)

$$OH = \frac{4 + 2*1 - 3}{1} + 1 = 4$$

$$OW = \frac{4 + 2*1 - 3}{1} + 1 = 4$$

[예2] 입력 (7, 7), 필터 (3, 3), 패딩 0, 스트라이드 2 ⇒ 출력 (3, 3)

$$OH = \frac{7 + 2*0 - 3}{2} + 1 = 3$$

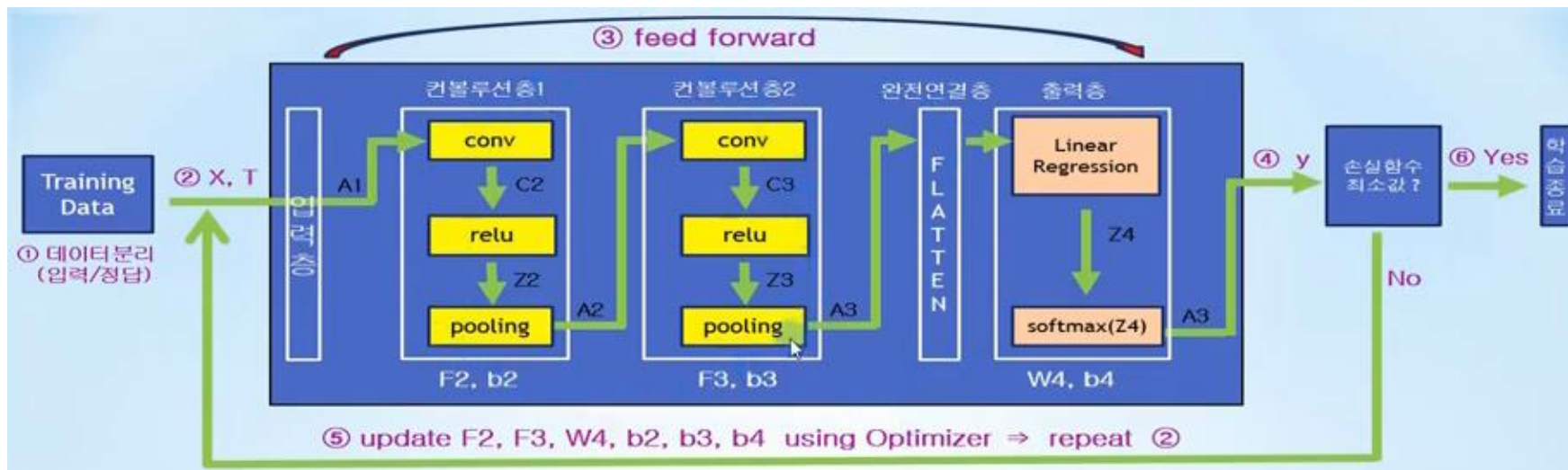
$$OW = \frac{7 + 2*0 - 3}{2} + 1 = 3$$

[예3] 입력 (28, 31), 필터 (5, 5), 패딩 2, 스트라이드 3 ⇒ 출력 (10, 11)

$$OH = \frac{28 + 2*2 - 5}{3} + 1 = 10$$

$$OW = \frac{31 + 2*2 - 5}{3} + 1 = 11$$

# 컨볼루션 층(convolution layer) 역할



	conv 출력	relu 출력	pooling 출력	컨볼루션 층 역할
컨볼루션층1	$A1 \otimes F2 + b2 = C2$ 입력 데이터    필터 (가중치 집합체)    바이어스    특징 맵 (feature map)	$C2 > 0, C2$ $C2 \leq 0, 0$	$A2$ max pooling	입력데이터 $A1$ 과 가중치들의 집합체인 1개 이상의 필터 $F2$ 와 컨볼루션 연산을 통해 <b>입력데이터 <math>A1</math>의 특징(feature)을 추출</b> 하는 역할을 수행함
컨볼루션층2	$A2 \otimes F3 + b3 = C3$ 입력 데이터    필터 (가중치 집합체)    바이어스    특징 맵 (feature map)	$C3 > 0, C3$ $C3 \leq 0, 0$	$A3$ max pooling	입력데이터 $A2$ 과 가중치들의 집합체인 1개 이상의 필터 $F3$ 와 컨볼루션 연산을 통해 <b>입력데이터 <math>A2</math>의 특징(feature)을 추출</b> 하는 역할을 수행함

# 컨볼루션 층(convolution layer) 역할

---

- ① 입력데이터와 1개 이상의 필터들과 컨볼루션 연산을 통해서
- ② 입력데이터 특징(feature)을 추출하여 특징맵(feature map)을 만들고
- ③ 특징맵에서 최대 값을 뽑아내서 다음 층으로 전달

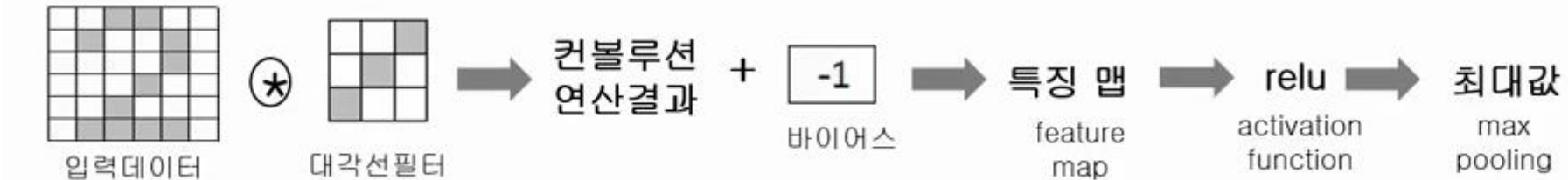
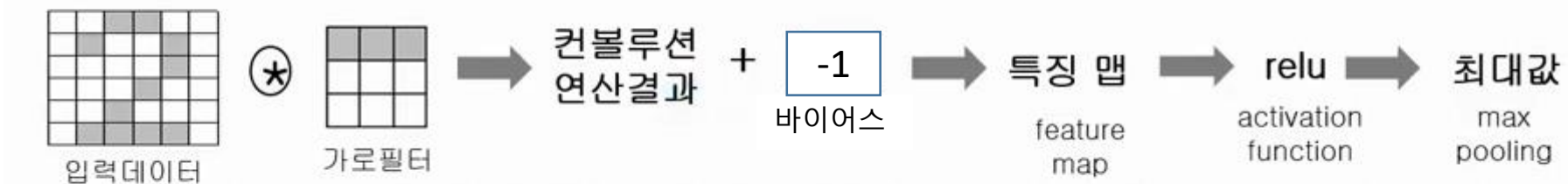


필터를 통해 데이터 특징을 추출 ?



# 특징추출과장

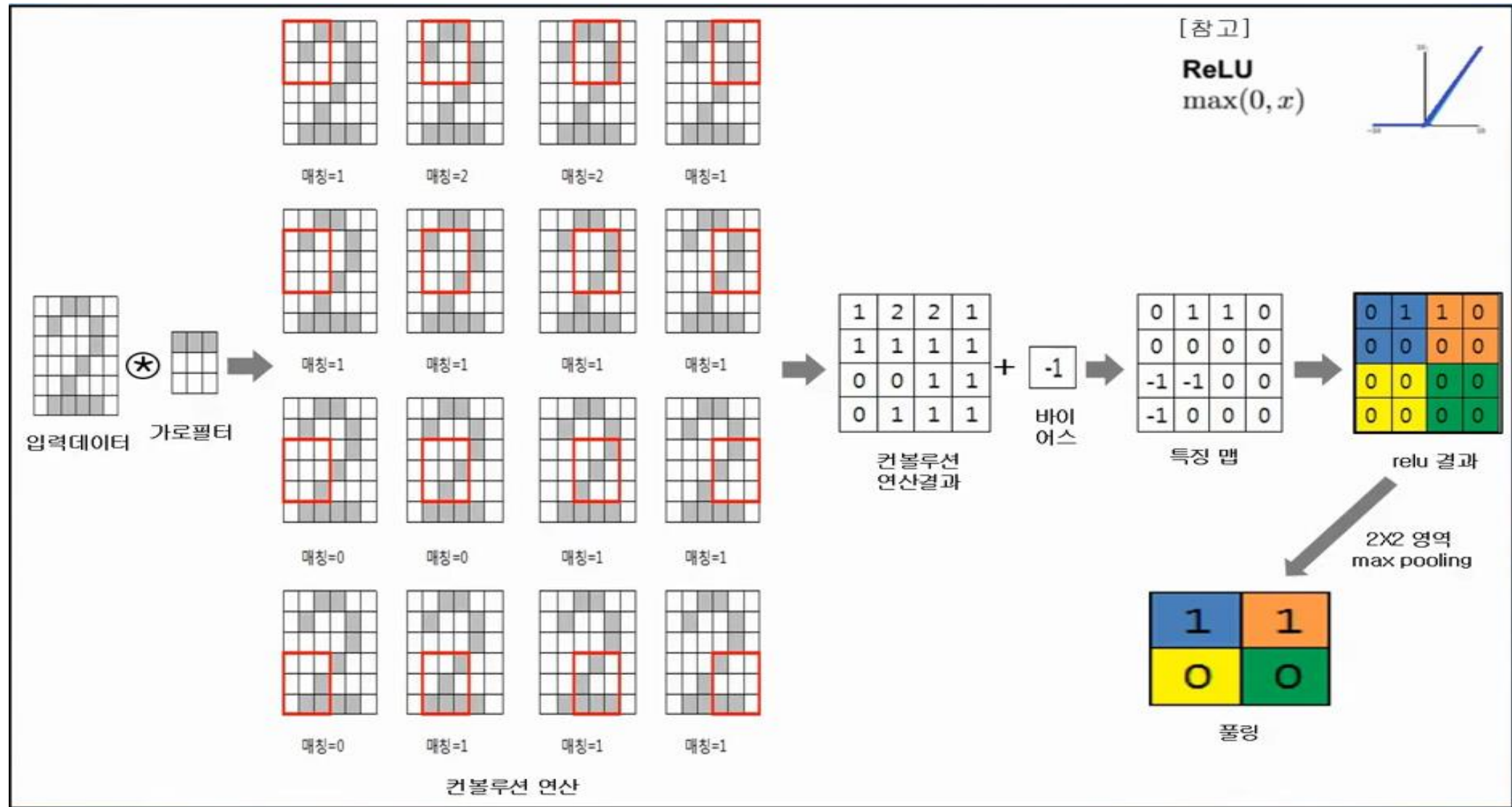
➤ 입력데이터 1개 (숫자 2)에 필터 3개 (가로, 대각선, 세로 필터) 적용 (계산 편의를 위해 패딩 적용하지 않음)



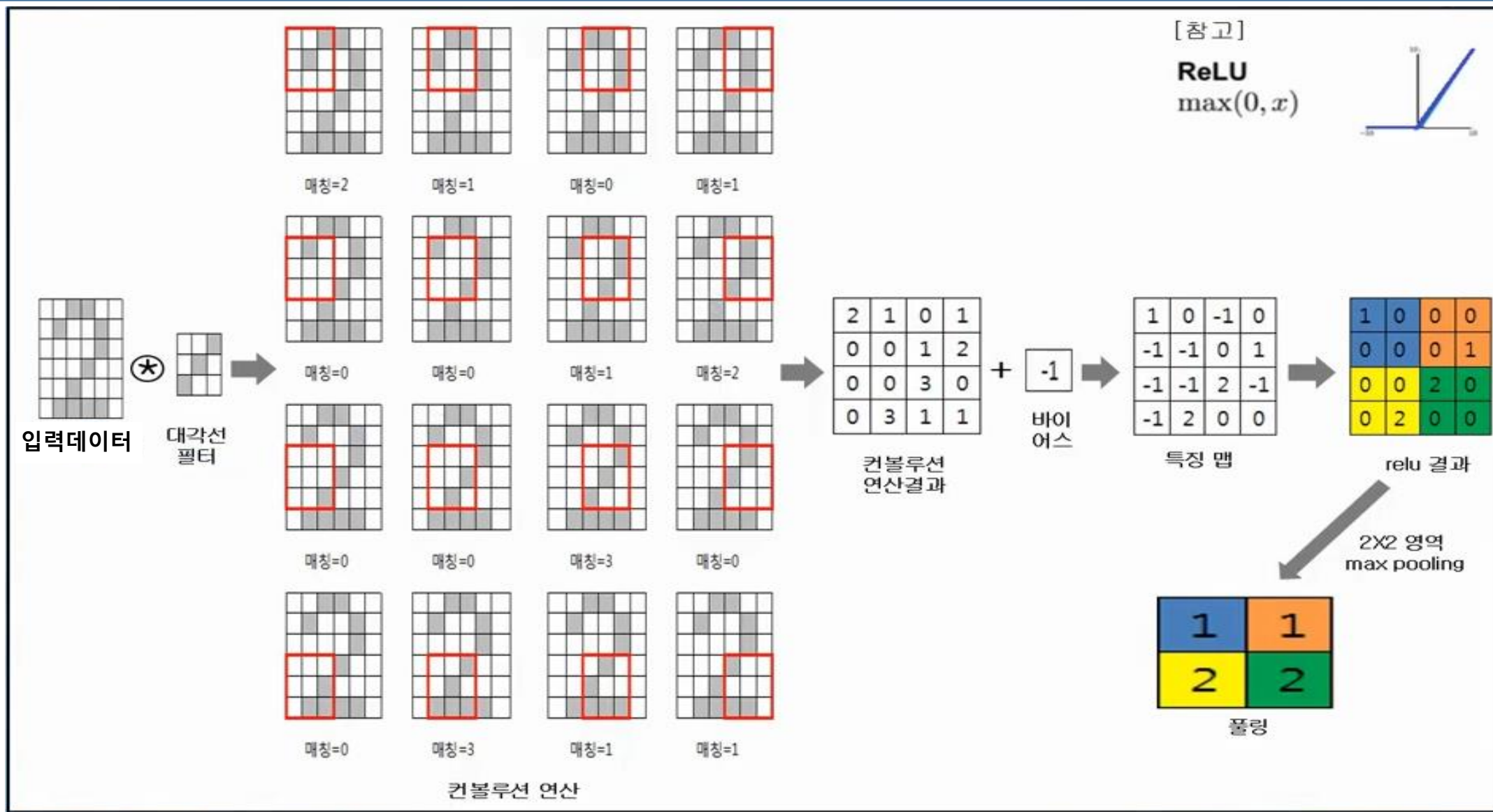
[참고] 입력데이터와 필터



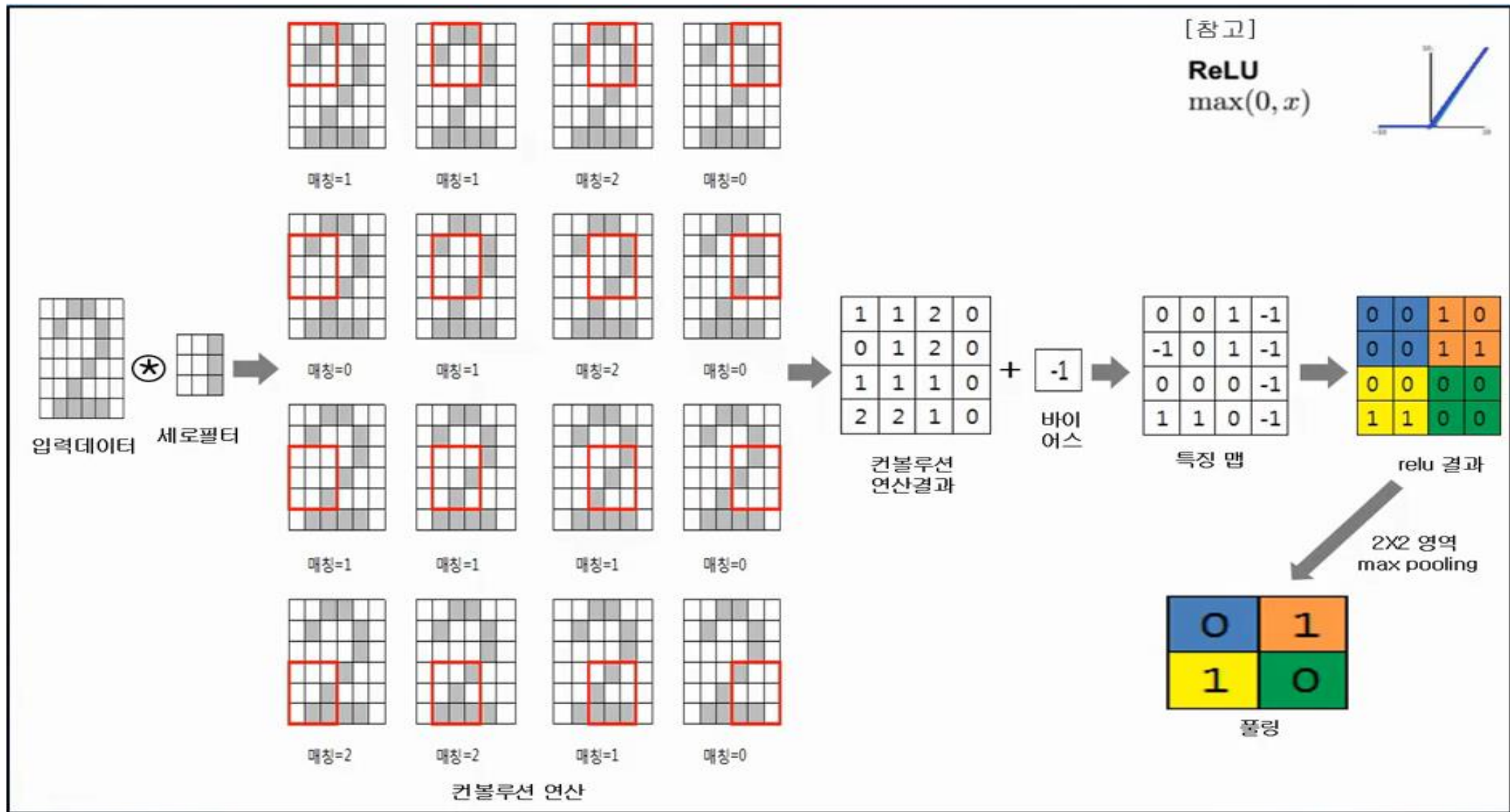
# 가로필터를 통한 입력데이터 측정추출(스트라이트1,패딩없음)



# 대각선필터를 통한 데이터 특징 추출

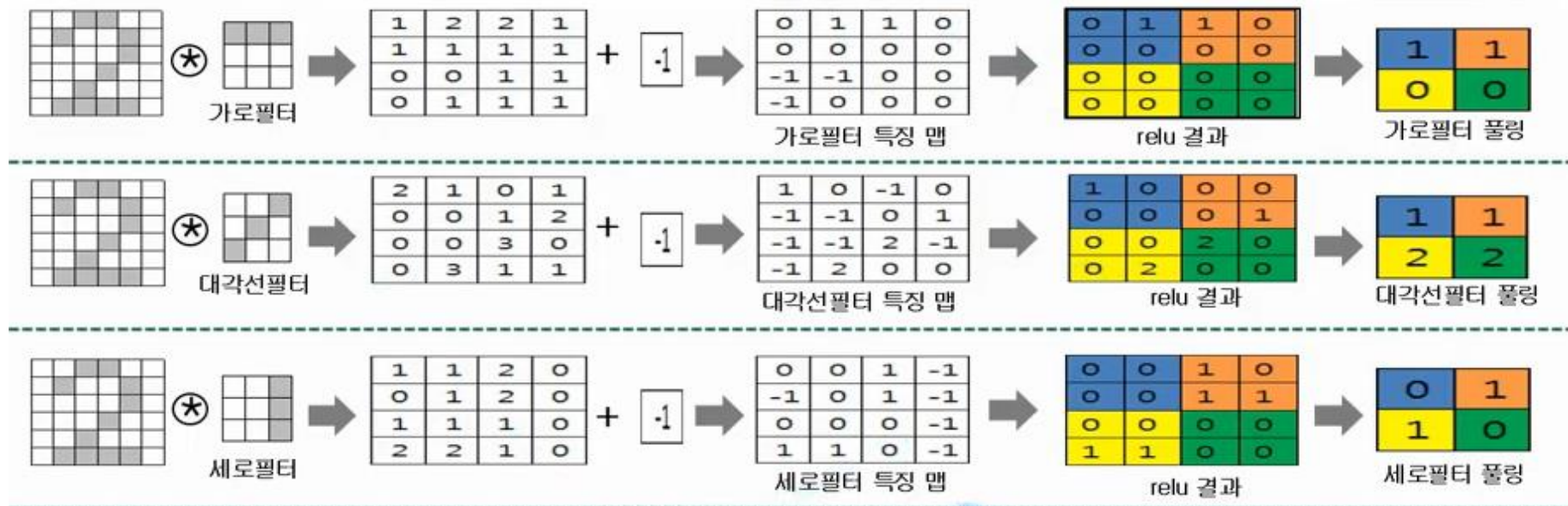


# 새로필터를 통한 입력데이터 특징추출





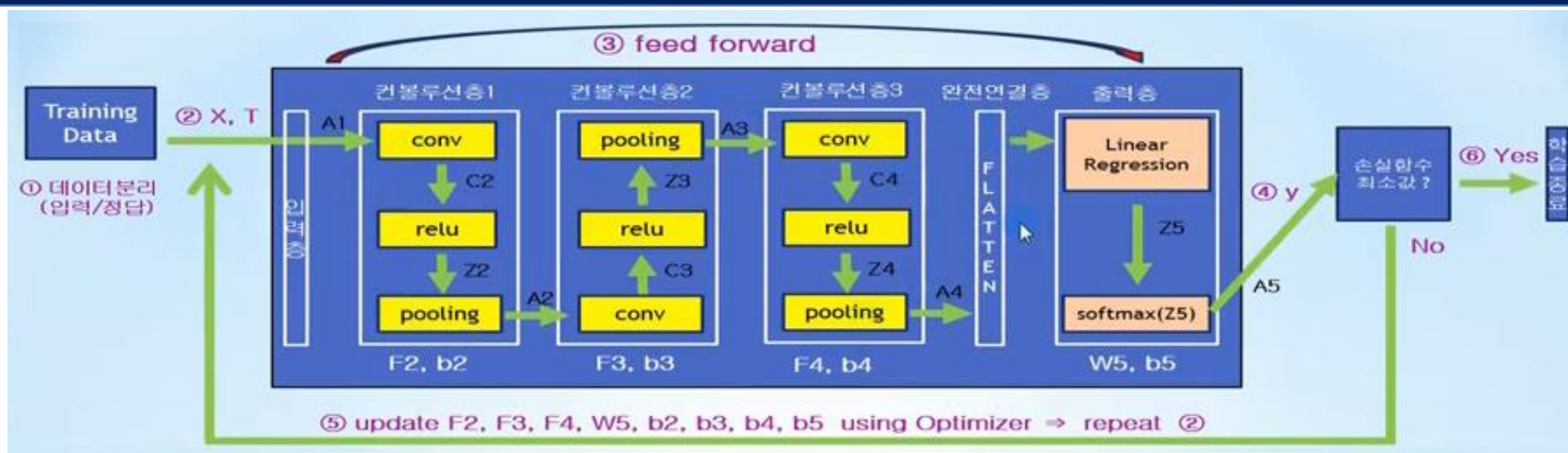
# 필터를 통한 입력데이터 특징 추출원리-특징 맵이 압축된 풀링 값



➤ 컨볼루션 연산 결과인 특징 맵(feature map) 값을 압축하고 있는 풀링 값을 보면,

- 대각선 필터에 대한 풀링 값이 가로와 세로필터의 풀링 값 보다 큰 값으로 구성되어 있는데,
- 풀링 값이 크다는 것은, 데이터 안에 해당 필터의 특징(성분)이 많이 포함되어 있는 것을 의미함. 즉, 특징 맵 값이 압축되어 있는 풀링 결과 값을 통해 데이터의 특징(성분)을 추출 할 수 있음
- 위의 예제를 보면, 입력 데이터 '2' 는 대각선 특징이 가로나 세로 특징보다 더욱 많이 포함되어 있으며 이러한 특징을 추출하는데 대각선 필터가 가로나 세로보다 유용하다는 것을 알 수 있음

# CNN(컨볼루션 신경망) 아키텍처



	conv 출력	relu 출력	pooling 출력	TensorFlow API
컨볼루션층1	$A1 \otimes F2 + b2 = C2 \rightarrow Z2 \rightarrow A2$			<b>conv</b> <code>tf.nn.conv2d(...)</code>
컨볼루션층2	$A2 \otimes F3 + b3 = C3 \rightarrow Z3 \rightarrow A3$			<b>relu</b> <code>tf.nn.relu(...)</code>
컨볼루션층3	$A3 \otimes F4 + b4 = C4 \rightarrow Z4 \rightarrow A4$			<b>pooling</b> <code>tf.nn.max_pool(...)</code>

	역할	TensorFlow API
완전연결층	컨볼루션 층의 3차원 출력 값을 1차원 벡터로 평탄화 작업 수행하여 일반 신경망 연결처럼 출력층의 모든 노드와 연결시켜주는 역할 수행	<b>FLATTEN</b> <code>tf.reshape(A4,...)</code>
출력층	입력받은 값을 출력으로 0~1사이의 값으로 모두 정규화하며 출력 값들의 총합은 항상 1이 되도록 하는 역할 수행	<b>softmax</b> <code>tf.nn.softmax(Z5)</code>