

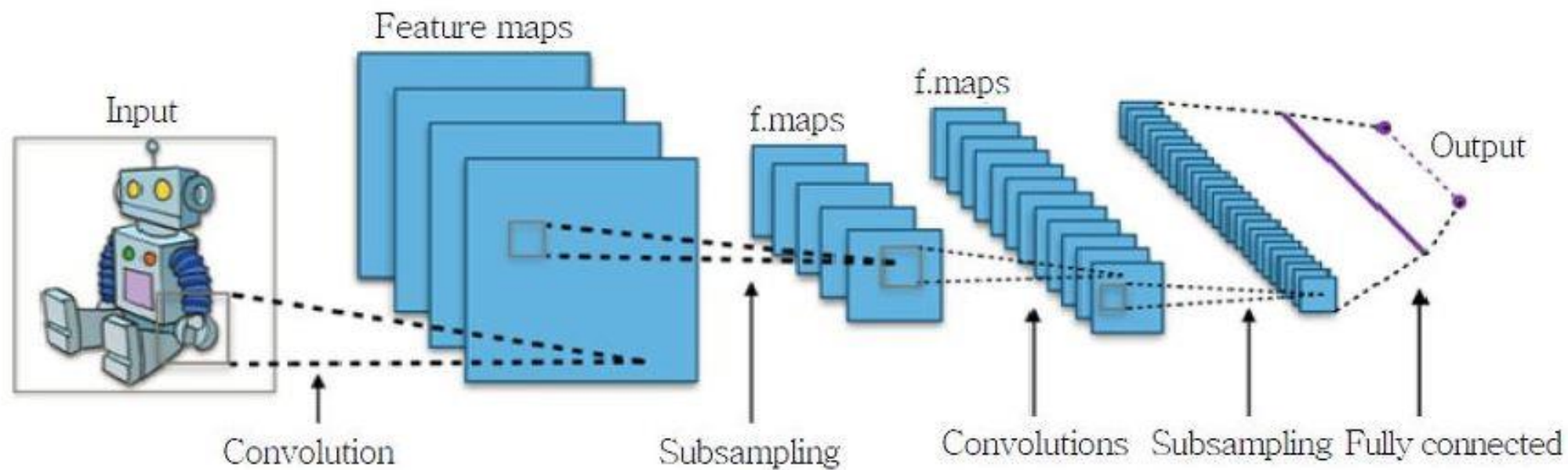
가천대 회화·조소과 AI 특강

2021-06-19

조형래

CNN(Convolutional Neural Network)

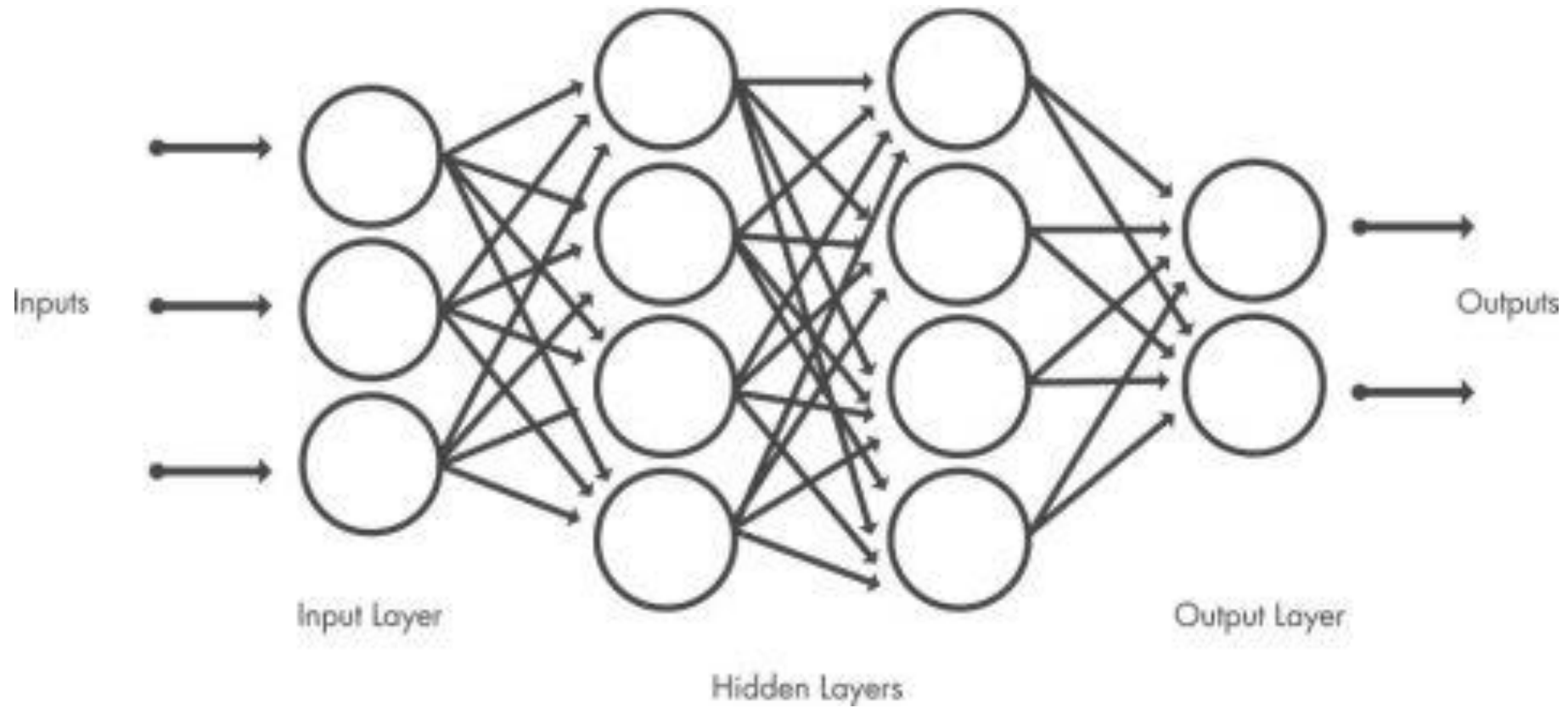
CNN



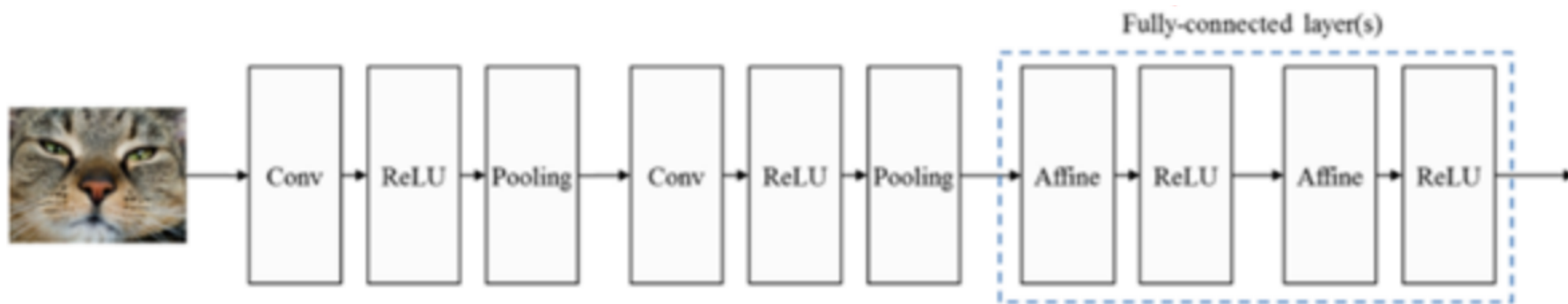
그림출처: <https://address83.tistory.com/164>

CNN

입력 계층, 은닉 계층, 출력 계층으로 구성



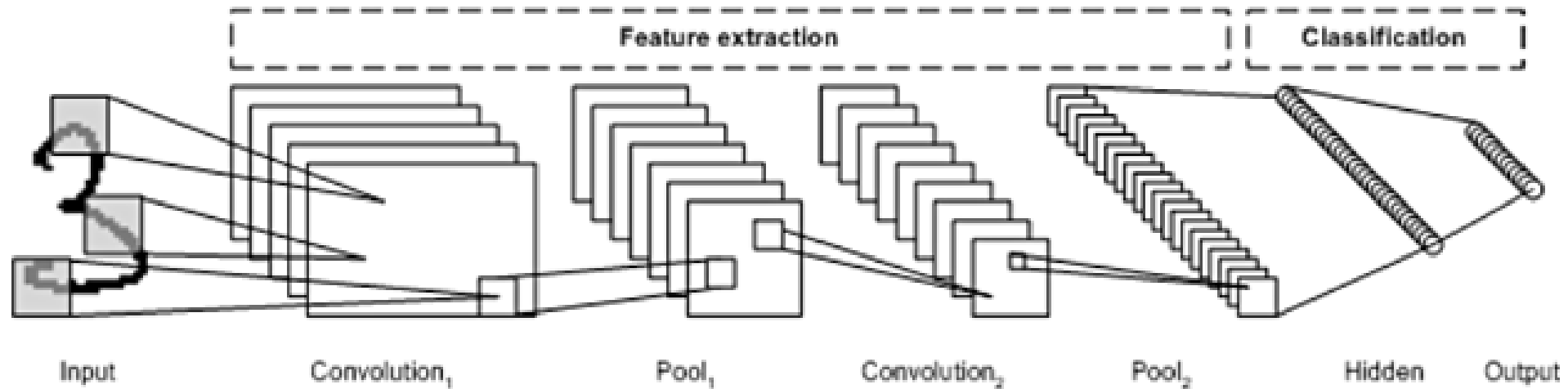
CNN



CNN 구조

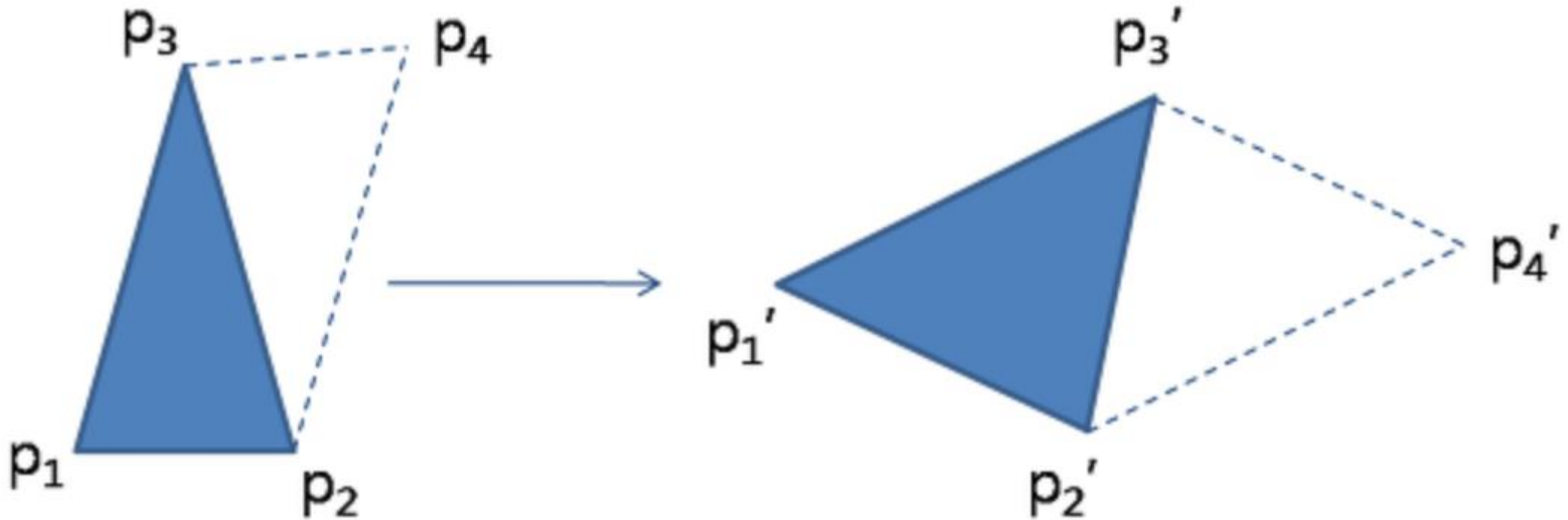
CNN

CNN은 입력 데이터에 대해서 필터를 통해 이미지의 특징을 추출하고 (Convolution Layer), 특징을 강화하고 이미지의 크기를 축소(Pooling Layer)한다. 이 과정을 반복하여 처리된 결과를 출력한다.



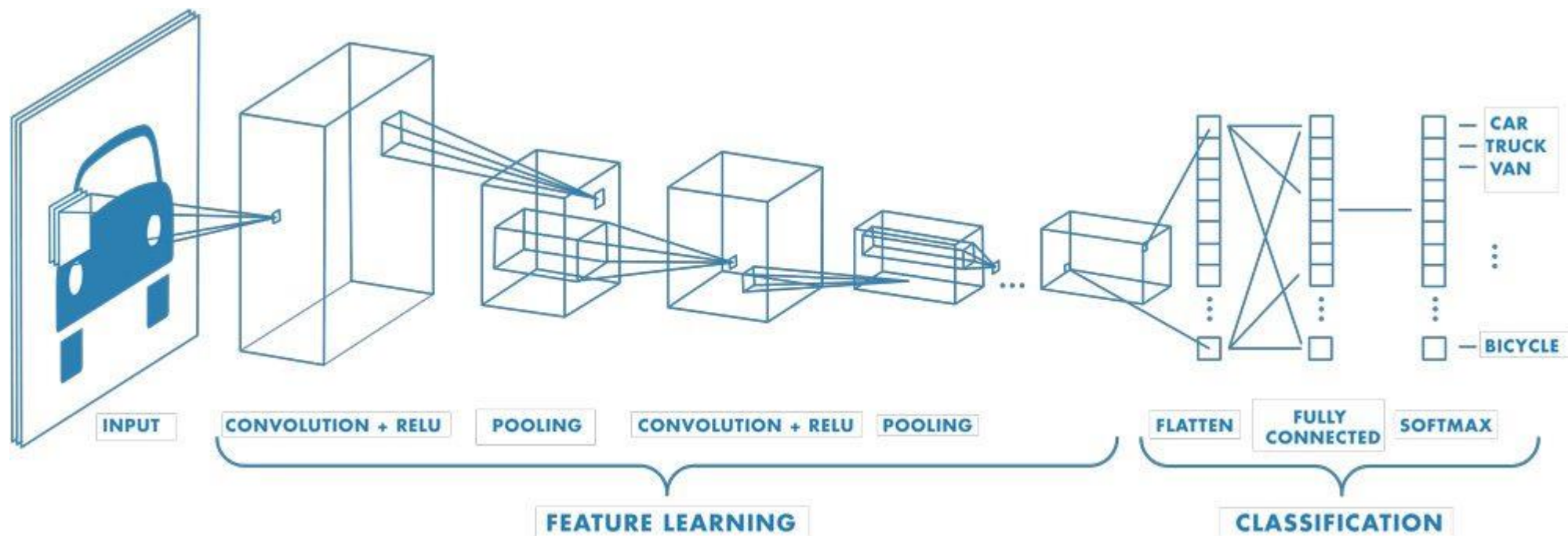
Affine

2D 평면에서의 affine 변환, 임의의 삼각형을 임의의 삼각형으로 매핑시킬 수 있는 변환(평행성을 보존)



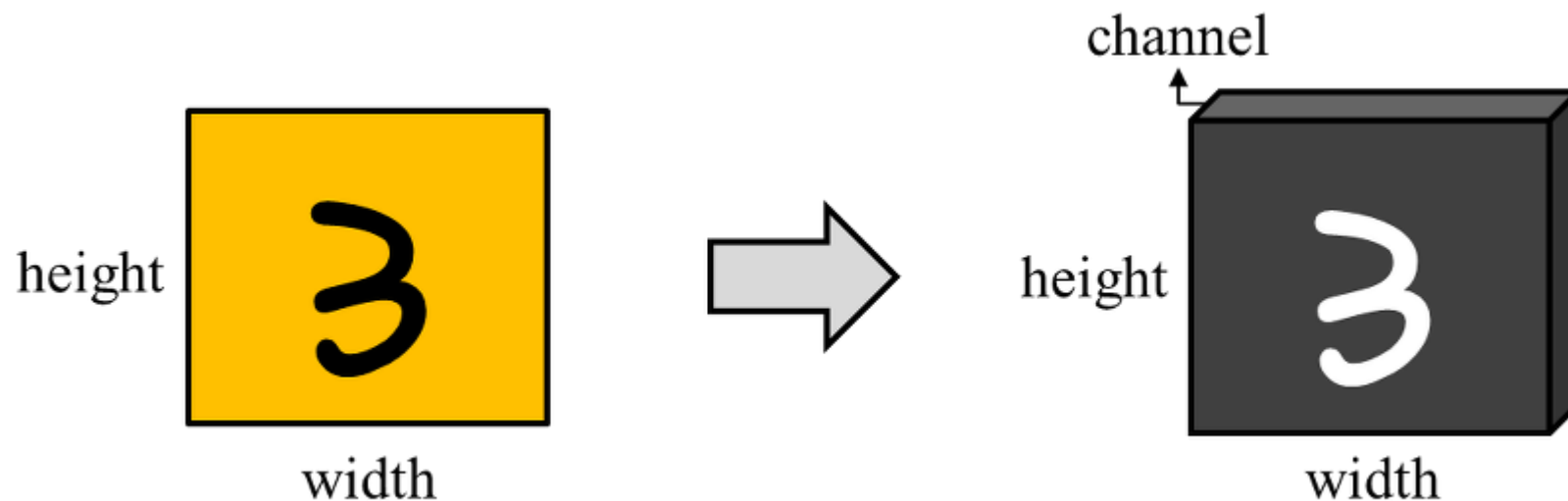
CNN

- **컨벌루션 계층**; 입력 영상을 컨벌루션 필터에 통과시킨다. 각 필터는 영상에서 특정 특징을 활성화한다.
- **ReLU(Rectified Linear Unit) 계층**; 음수 값은 0에 매핑하고 양수 값은 그대로 두어서 더 빠르고 효과적인 훈련이 이루어지도록 한다.
- **풀링 계층**; 비선형 다운샘플링을 수행하여 신경망이 학습해야 하는 파라미터의 개수를 줄임



합성곱 계층 (Convolutional Layer)

이미지 데이터는 높이X너비X채널의 3차원 텐서 (tensor)로 표현될 수 있다.
이미지의 색상이 RGB 코드로 표현되었다면, 채널의 크기는 3이 되며
각각의 채널에는 R, G, B 값이 저장된다.



컨벌루션 레이어(Convolutional Layer)

높이X너비X채널이 4X4X1 인 텐서 형태의 입력 이미지에 대해 3X3 크기의 필터를 적용하는 합성곱 계층에서는 이미지와 필터에 대한 합성곱 연산을 통해 2X2X1 텐서 형태의 이미지가 생성된다.

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 \odot

1	0	1
1	2	0
3	0	1

 =

40	

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 \odot

1	0	1
1	2	0
3	0	1

 =

40	32

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 \odot

1	0	1
1	2	0
3	0	1

 =

40	32
26	

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 \odot

1	0	1
1	2	0
3	0	1

 =

40	32
26	25

CNN

- 행렬 형태로 필터를 처리

Input image



Convolution
Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map

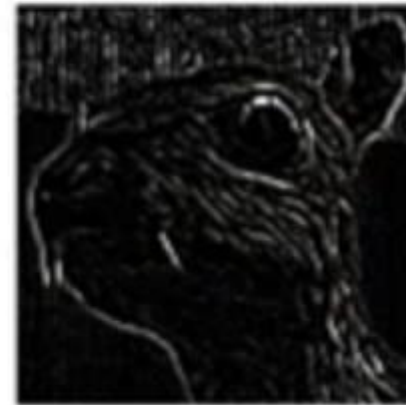
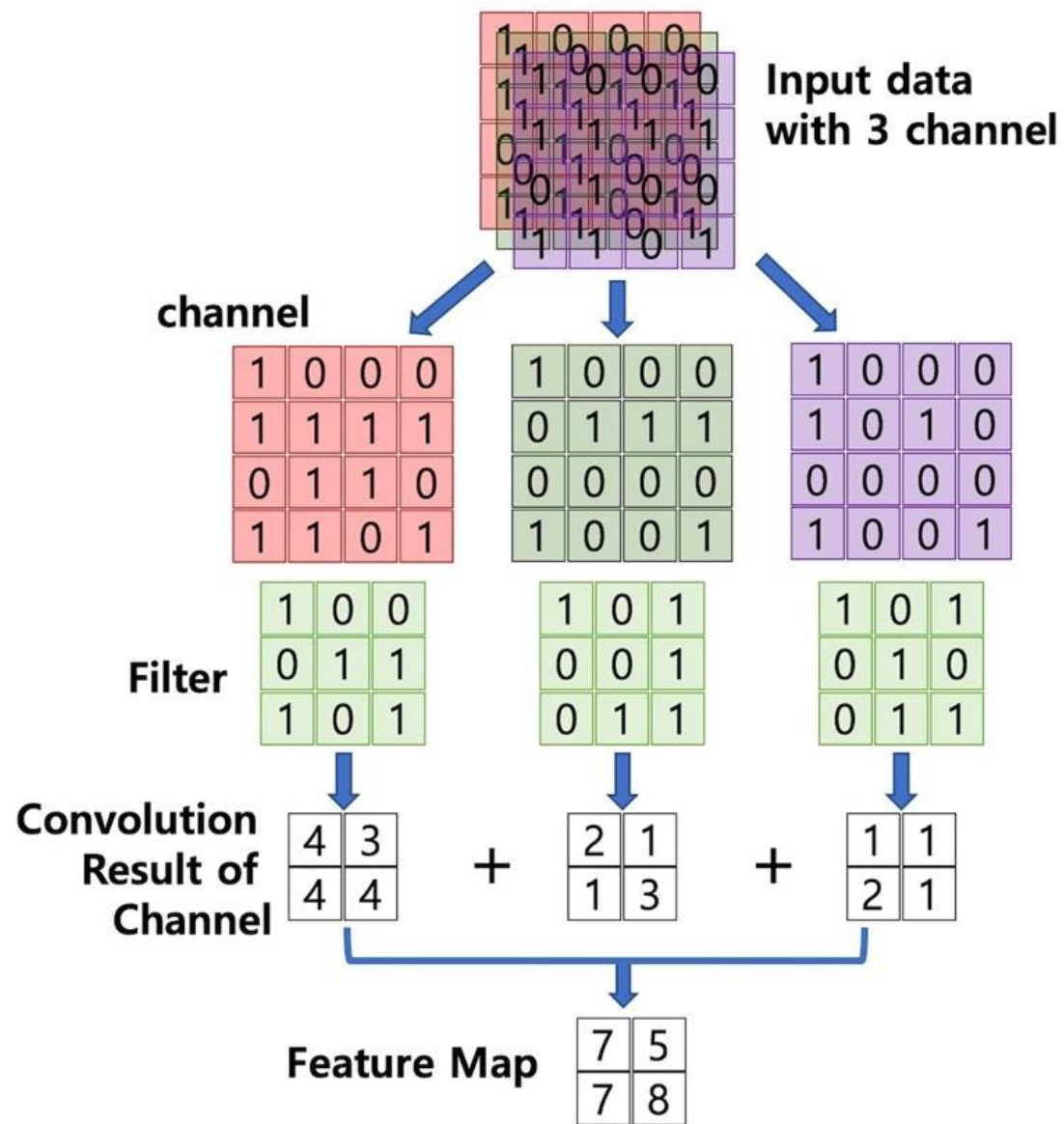


그림 출처; <https://m.blog.naver.com/jevida/221841296542>

채널, 필터, 컨볼루션, 피쳐맵



패딩(Padding)

합성곱 계층을 거치면서 이미지의 크기는 점점 작아지게 되고 이미지의 가장자리에 위치한 픽셀들의 정보는 점점 사라지게 된다. 이러한 문제점을 해결하기 위해 이용되는 것이 패딩 (Padding)이다.

패딩은 이미지의 가장자리에 특정 값으로 설정된 픽셀들을 추가함으로써 입력 이미지와 출력이미지의 크기를 같거나 비슷하게 만드는 역할을 수행한다.

0	0	0	0	0	0
0	0	1	7	5	0
0	5	5	6	6	0
0	5	3	3	0	0
0	1	1	1	2	0
0	0	0	0	0	0

 \otimes

1	0	0
1	2	1
1	2	3

 $=$

26	42	55	35
34	41	33	28
18	25	23	14
3	9	8	8

ReLU

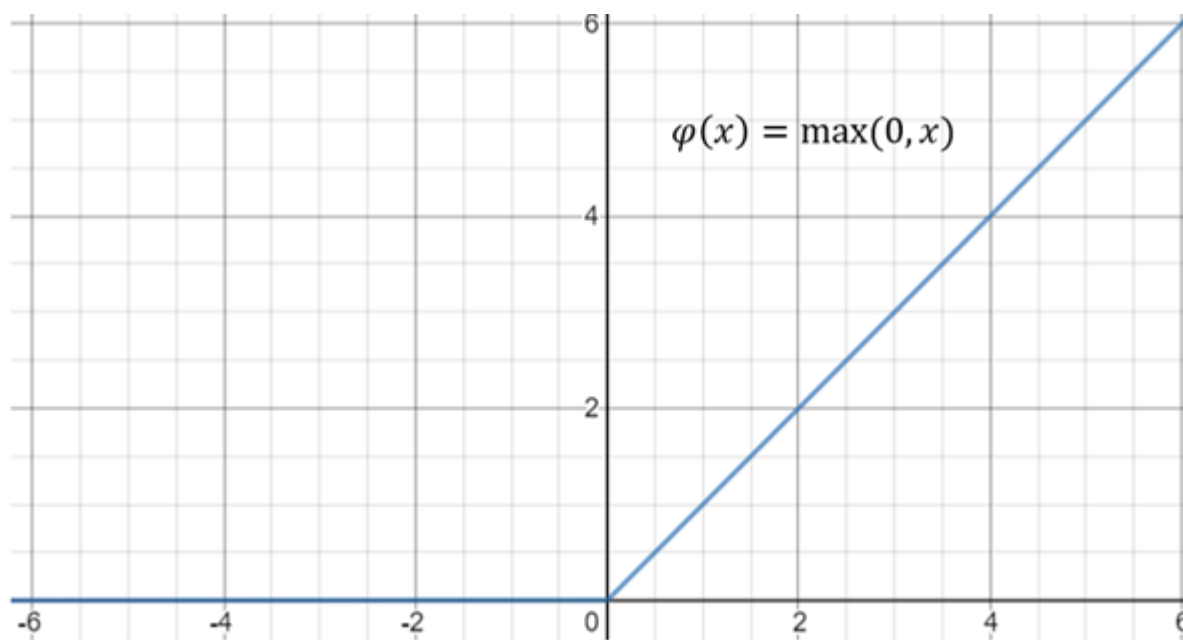
입력이 음수이면 0을 출력하고, 양수이면 입력 값 그대로 출력하는 함수

$$\varphi(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} = \max(0, x)$$

ReLU formula

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

ReLU derivative



Activation Function;
입력 신호의 총합을 출력 신호로 변환하는 함수

풀링(Pooling)

- Pooling Layer는 이미지의 국소적인 부분들을 하나의 대표적인 스칼라 값으로 변환함으로써 이미지 크기를 줄이는 기능을 수행
- Pooling = sub sampling 레이어
- 컨볼루션 레이어의 출력 데이터를 받아서 출력 데이터(Activation Map)의 크기를 줄이거나 특정 데이터를 강조하는 용도로 사용
- Max Pooling, Min Pooling, Average Pooling
- 특징: Pooling 레이어를 통과하면 행렬의 크기 감소, Pooling 레이어를 통해서 채널 수 변경 없음

13	20	30	0
8	12	3	0
34	70	33	5
111	80	10	23

Activation Map

20	30
111	33

Max Pooling

13	8
66	18

Average Pooling

8	0
34	5

Min Pooling

텐서

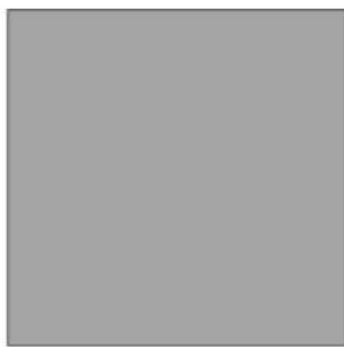
- 데이터의 배열
- 텐서의 Rank는 몇 차원 배열인가를 의미
- 스칼라는 일반적으로 존재하는 값(1개)
- 벡터는 스칼라가 여러 개 모인 것

RANK	TYPE	EXAMPLE
0	scalar	[1]
1	vector	[1,1]
2	matrix	[[1,1],[1,1]]
3	3-tensor	[[[1,1],[1,1]],[[1,1],[1,1]],[[1,2],[2,1]]]
n	n-tensor	

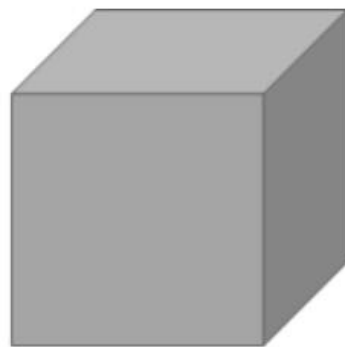
텐서



vector



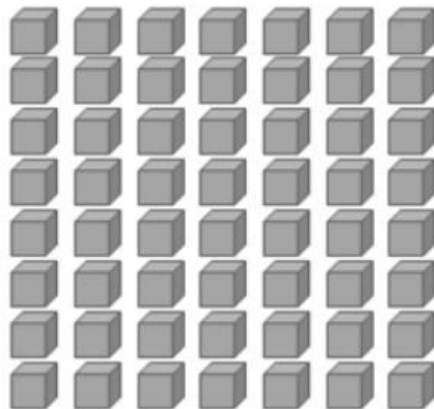
matrix



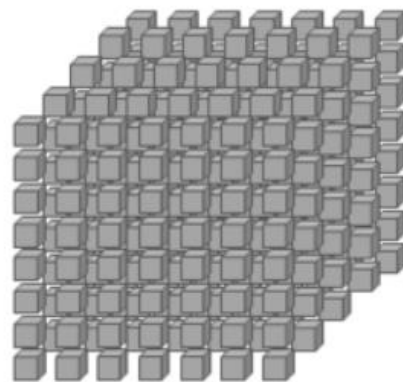
3d-tensor



4d-tensor



5d-tensor



6d-tensor

텐서 예시

WORD	INDEX	EXAMPLE
hi	0	[1,0,0,0]
king	1	[0,1,0,0]
queen	2	[0,0,1,0]
jack	3	[0,0,0,1]

단어의 vector를 기반으로 문장을 matrix로 표현=>

word	EXAMPLE
hi king	[[1,0,0,0],[0,1,0,0]]
hi queen	[[1,0,0,0],[0,0,1,0]]
hi jack	[[1,0,0,0],[0,0,0,1]]

말뭉치=>

hi king hi queen hi jack
[[[1,0,0,0], [0,1,0,0]], [[1,0,0,0], [0,0,1,0]], [[1,0,0,0], [0,0,1,0]]]

- word는 4차원
 - 각 문자는 2개의 단어
 - 총 3개의 문장
- 따라서 모델은 (3,2,4)의 3차원 Tensor

스칼라, 벡터, 매트릭스.ipynb

https://github.com/artjow/-AI-/blob/main/%EC%8A%A4%EC%B9%BC%EB%9D%BC%2C%EB%B2%A1%ED%84%B0_%EB%A7%A4%ED%8A%B8%EB%A6%ADipynb.ipynb

컨볼루션 계산

O: 출력 이미지의 크기 (폭)
I: 입력 이미지의 크기 (너비)
K: 커널
S: 스트라이드
P: 패딩 크기
N: 커널 수

출력 이미지의 채널 수는 갯수 (N)와 같음

$$O = \frac{I - K + 2P}{S} + 1$$

컨볼루션 계산; 컨볼루션 계층의 출력 텐서 크기

$$O = \frac{I - K + 2P}{S} + 1$$

문제) 입력 이미지 크기는 $227 \times 227 \times 3$

첫 번째 전환 레이어 (Conv-1)는 $11 \times 11 \times 3$ 크기의 수 96 개, stride = 4, padding = 0

$$O = \frac{227 - 11 + 2 * 0}{4} + 1 = 55$$

Conv-1의 출력 출력 텐서 크기는 $55 \times 55 \times 96$

3 채널 (RGB) 이미지에 대해 3 배가 곱 해져 총 $55 \times 55 \times 96 \times 3$ 이 됨.

따라서 출력 텐서는 871,200

CNN 네트워크 실습

<https://github.com/artjow/-AI-/blob/main/cnn.ipynb>