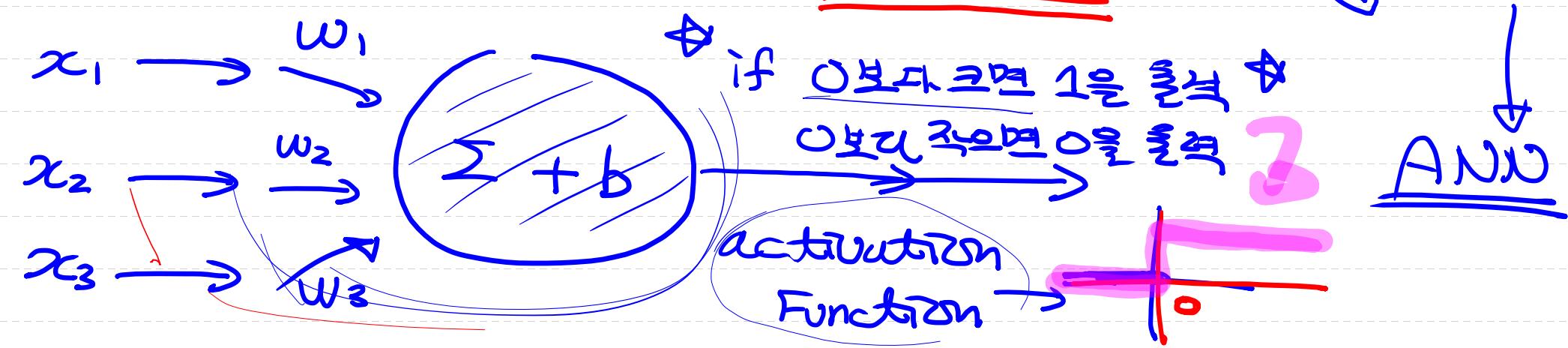
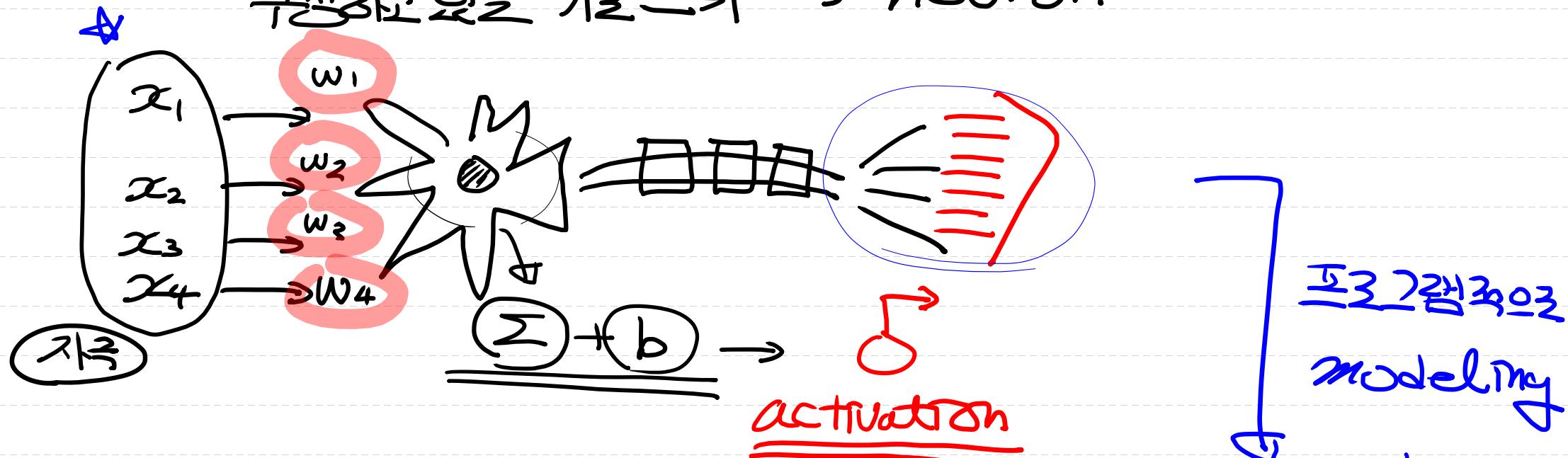


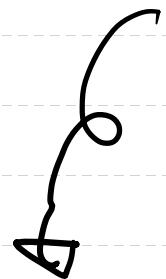
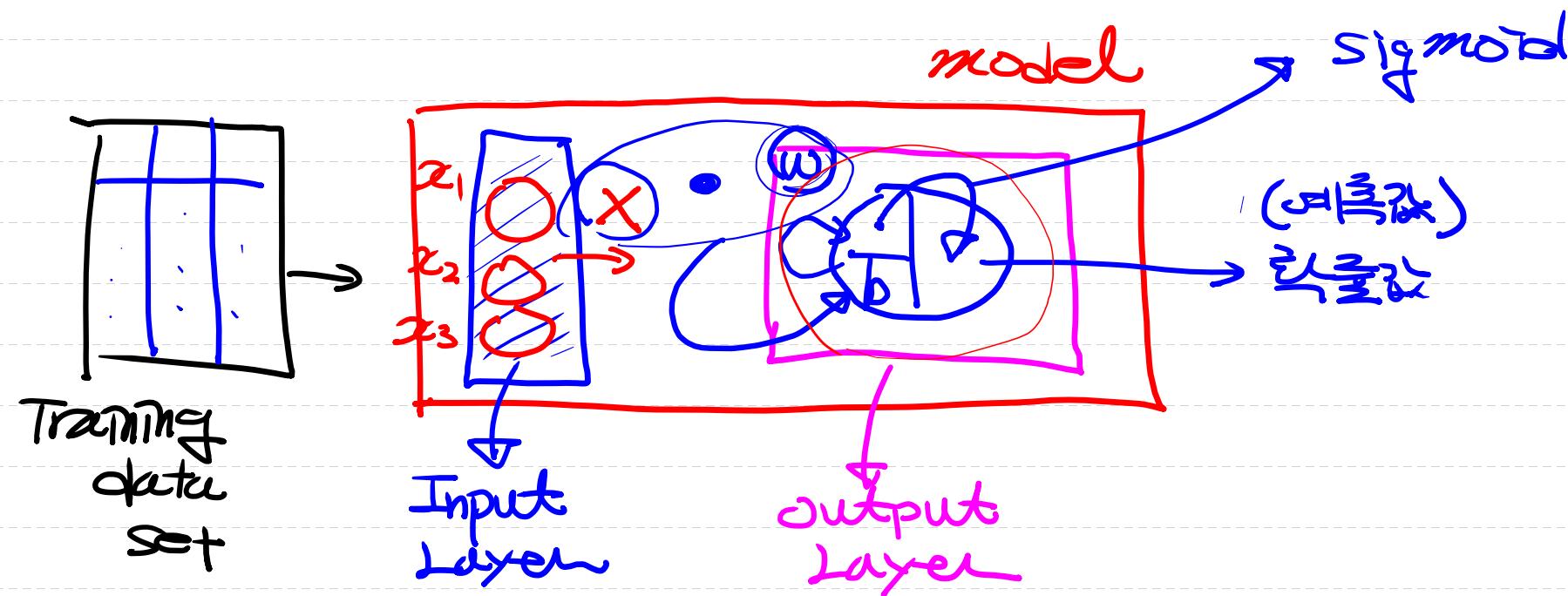
• 12/14

• Neural Network (신경망)

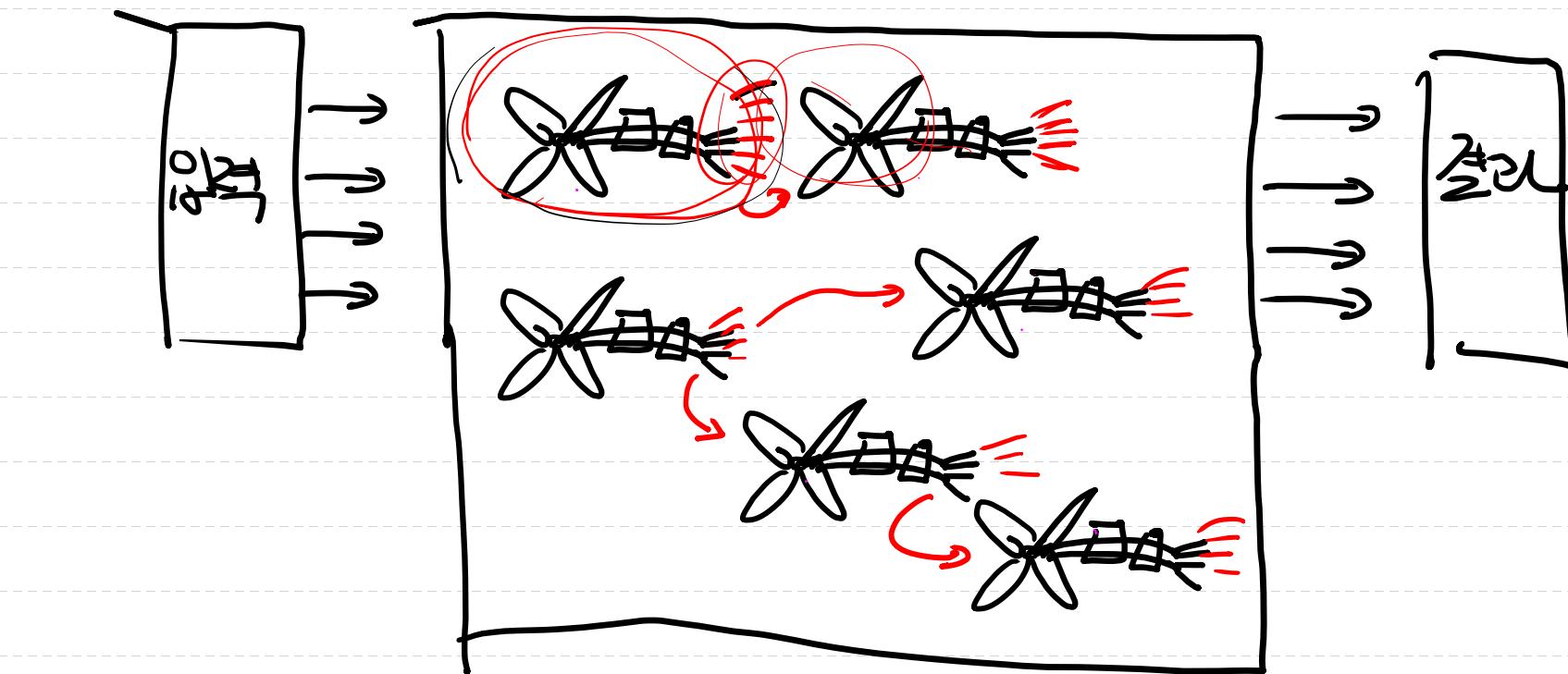
구성하고 있는 기본 един위 → "neuron"



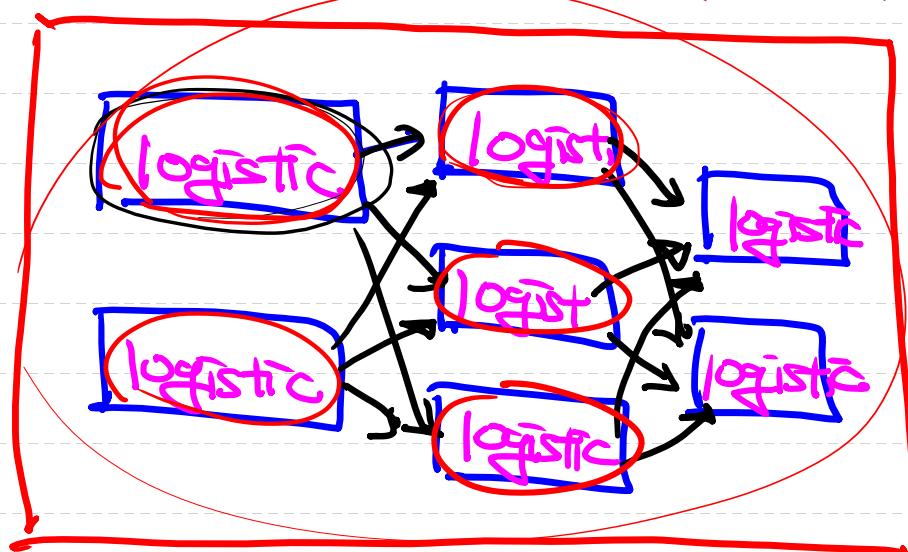
우리가 배운 logistic regression과 퍼셉트론 구조



Neural Network



ANN (Artificial Neural Network)



→ Deep Learning
구조

Deep Learning

1개의 logistic regression을 나누어주는 node가
서로 연결되어 있는 신경망 구조.

입력층, 1개 이상의 은닉층, 출력층을 구축하고 있나요!
→ layer로 구성되어 있나요

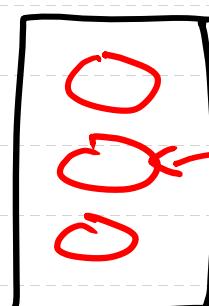
출력층의 오차를 기반으로 layer의 각 node가 가지는
가중치를 학습하는 machine learning 기법

1개 이상의 은닉층 → 끌거를 사용하는지 풀었는지 문제마다

풀어야겠는데

hidden layer가 존재하면

"Overfitting"



몇 개?

일반적으로

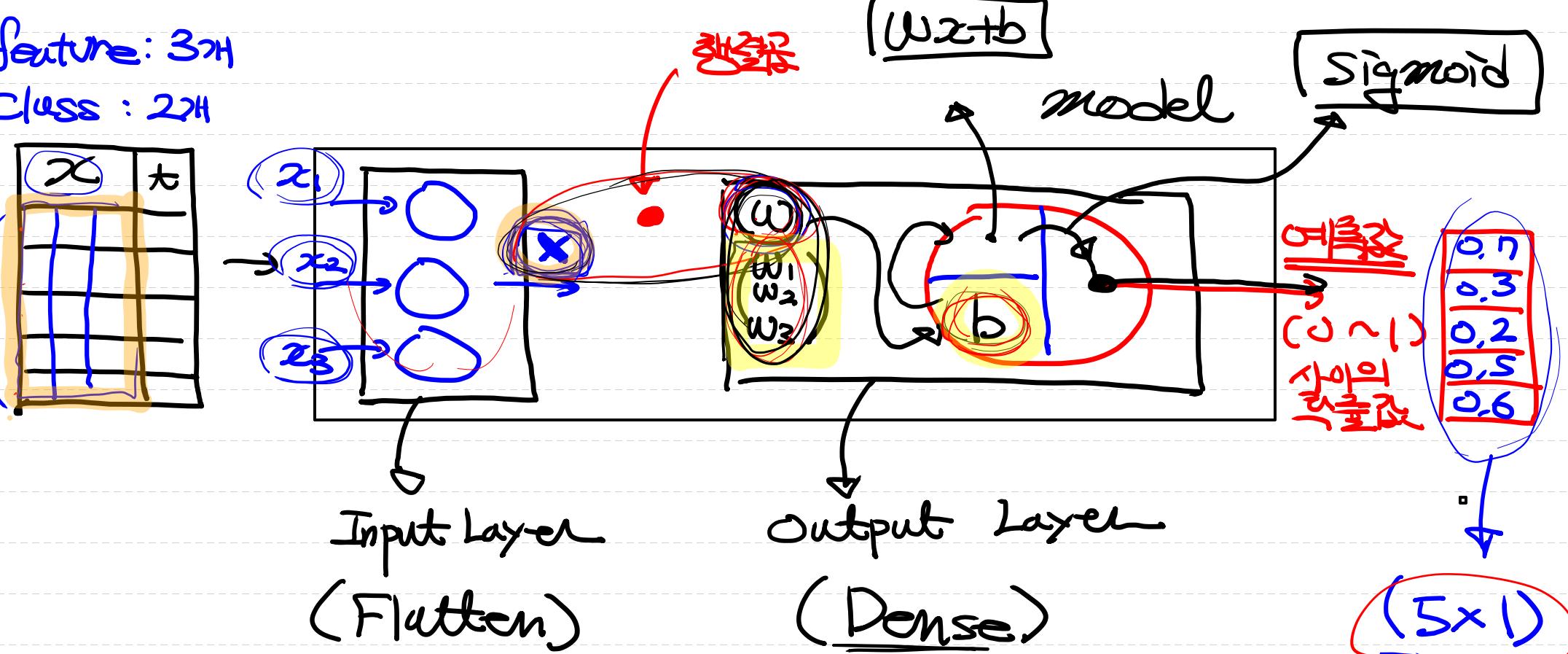
몇 개?

node의 수보다는 layer의 개수가 중요!

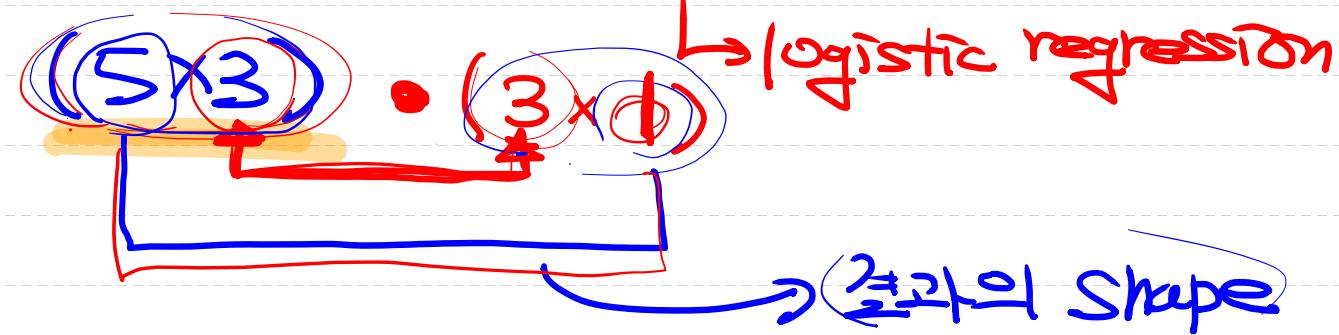
풀까요

• logistic regression 1개를 이용 → binary classification

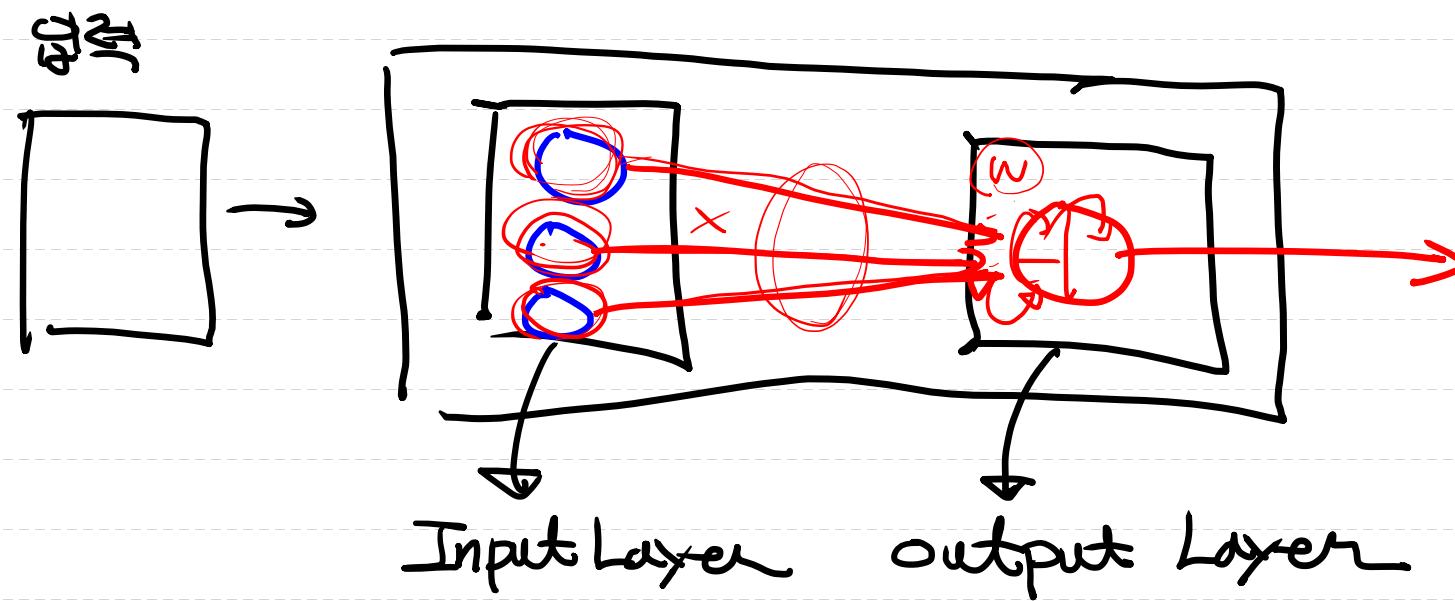
- ✓ feature: 3개
- ✓ class : 2개



우2개
계산해야하는
parameters
→ 4개



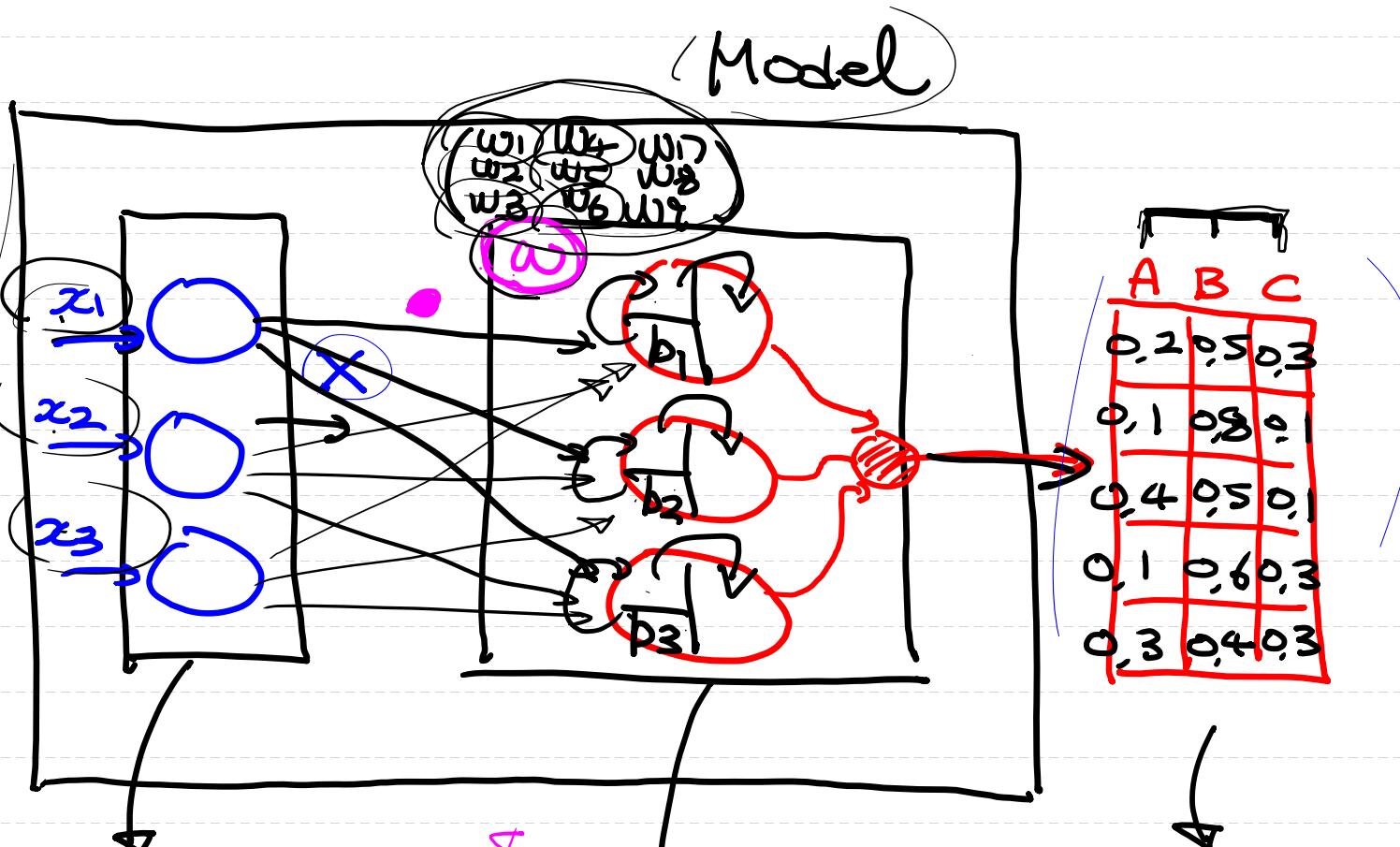
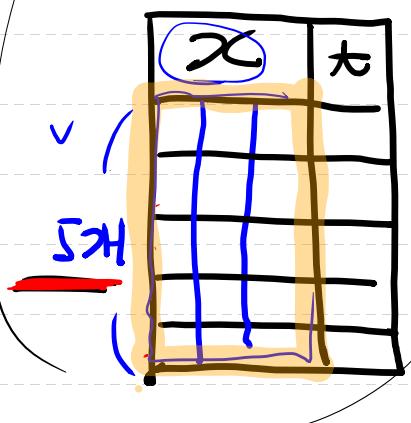
간단한 2진 분류 (binary classification)



○ Logistic regression은 2 분류가 아닙니다 → Multinomial Classification

feature: 3개

Class: 3개



	A	B	C
0, 1	0, 5	0, 3	0, 1
0, 1	0, 8	0, 1	0, 1
0, 4	0, 5	0, 1	0, 1
0, 1	0, 6	0, 3	0, 3
0, 3	0, 4	0, 3	0, 3

Input Layer

Output Layer

Input Layer Output Layer

X의 shape (5×3)

(3×3)

(5×3)

$$9+3=12\text{H}$$

Deep Learning 03 표현

hidden layer - 2개

각 hidden layer의 node

hidden Layer

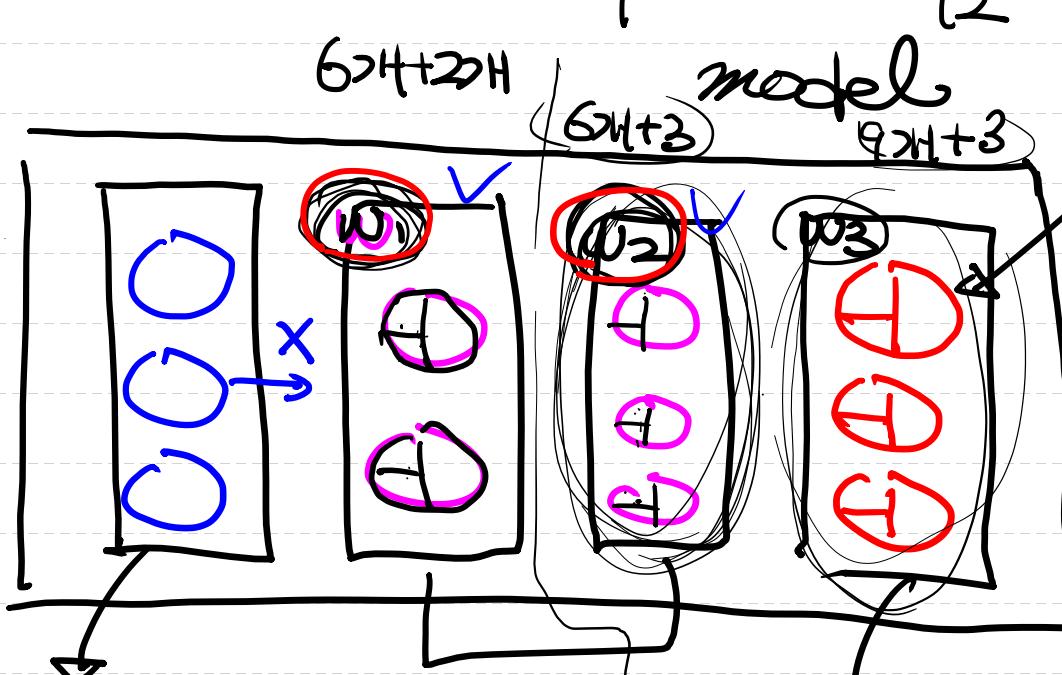
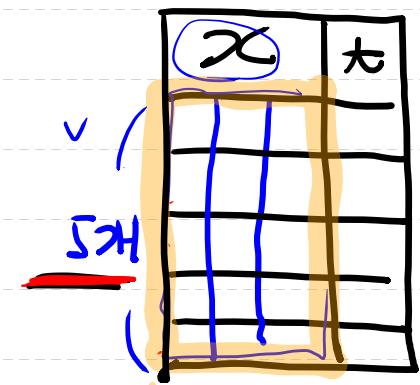
n=0

feature: 3개

class : 3개

(2, 3)

logistic regression



activation
→ softmax

A	B	C
0, 2	0, 5	0, 3
0, 1	0, 8	0, 1
0, 4	0, 5	0, 1
0, 1	0, 6	0, 3
0, 3	0, 4	0, 3

Input
Layer

hidden
Layer

outputs
Layer

우리가 계산해야

될는 parameter

는 몇개?? → 12개

X의 shape

(5, 3)

(Dense)

3×2

(5, 2)

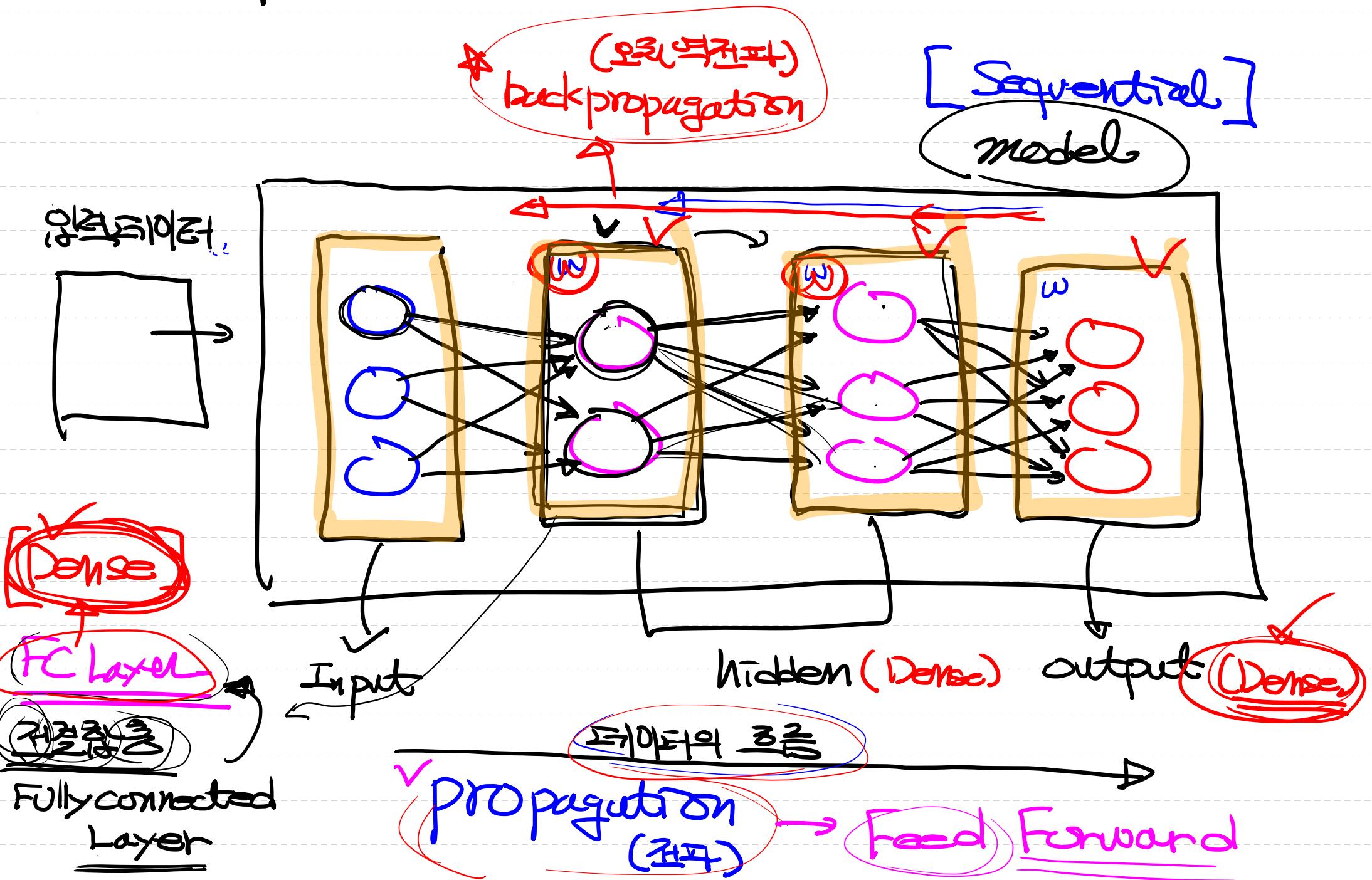
(2, 3)

→ softmax

(Dense)

$(5 \times 3) \times (3 \times 3)$

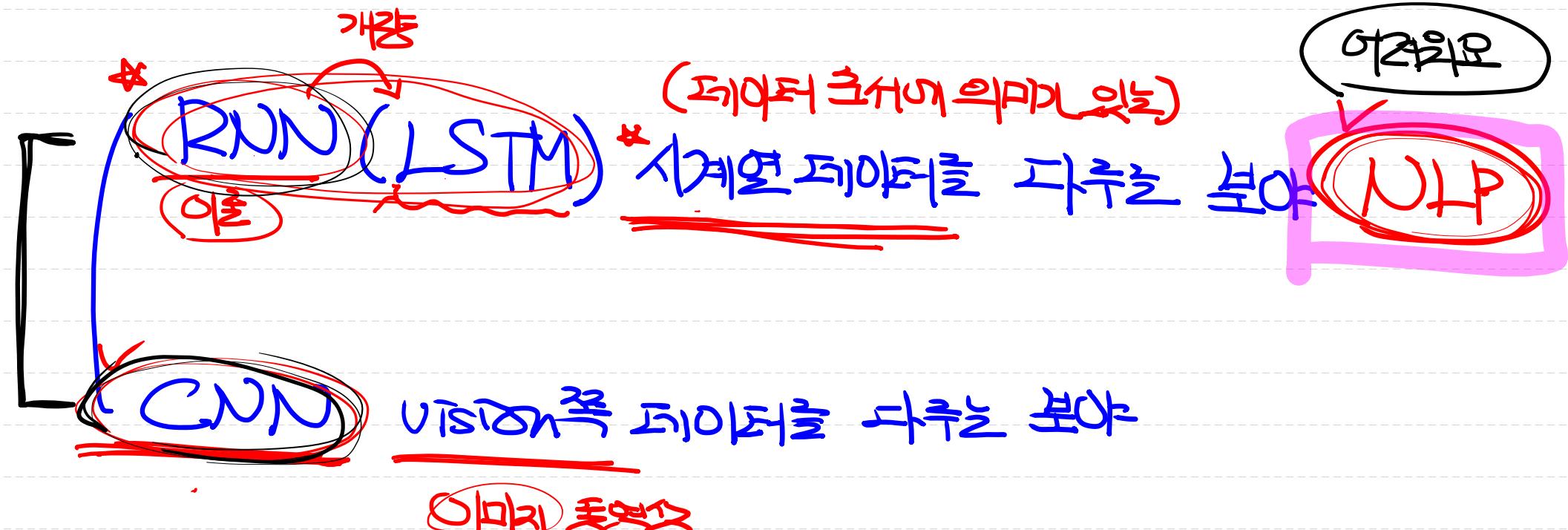
Simple 흐 형태의 그림. (레이어가 연결되어 연결되어 있지요!)



Dense (FC Layer)로 구성된 Neural Network을

→ Fully Connected Network (전결합 신경망)

• Deep Learning → DNN (Deep Neural Network)



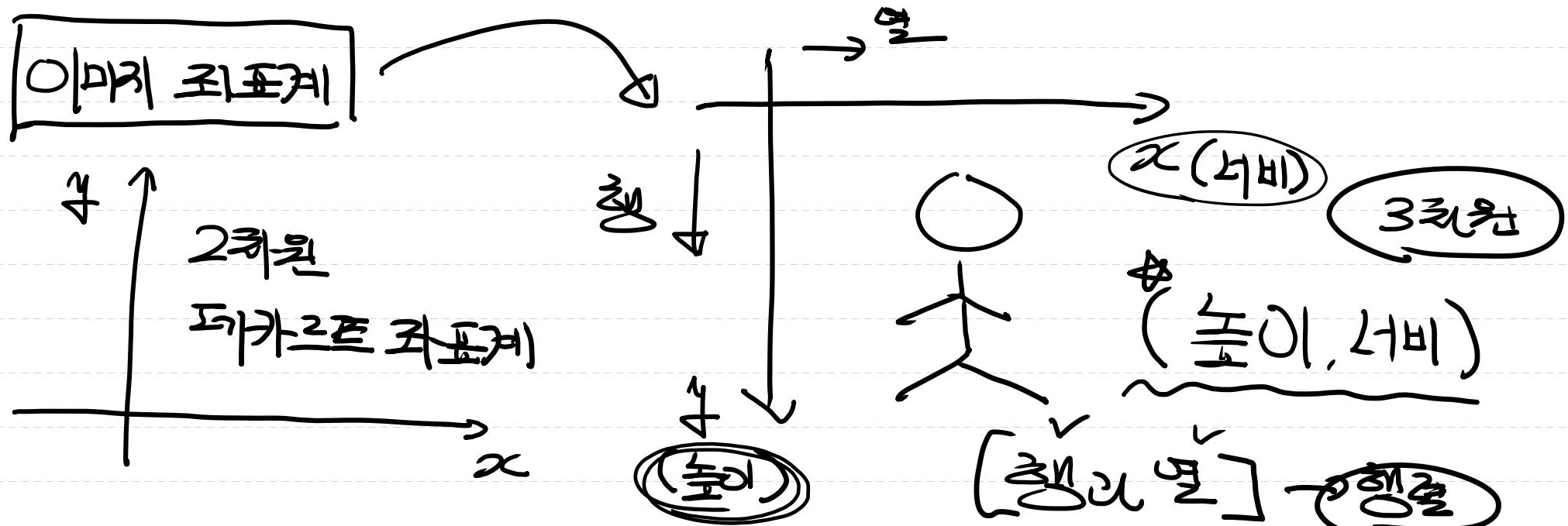
• CNN → Image에 대한 기반적 이해 + 용어.

Image의 기본

- 이미지를 구성하는 가장 작은 단위 \rightarrow pixel

우리가 흔히 해상도를 말할 때 이 해상도가 pixel의 개수!

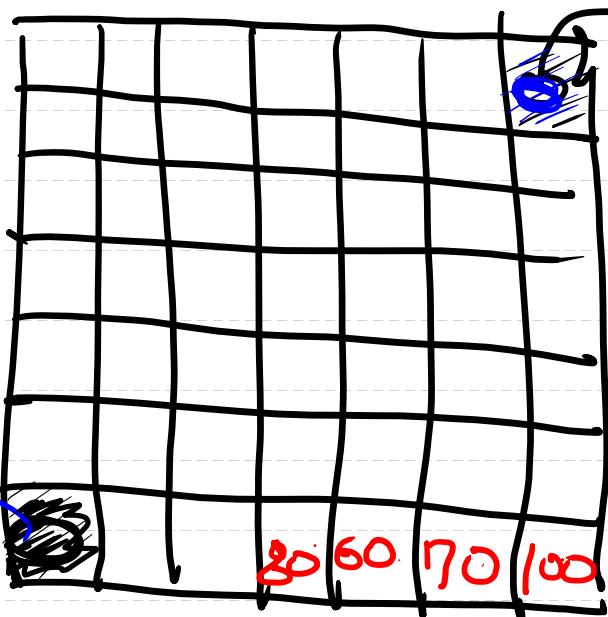
하나의 pixel은 기본적으로 3개의 값으로 구성 + 1개의 깊이(밝기) \rightarrow R, G, B (24bit) <투명도>



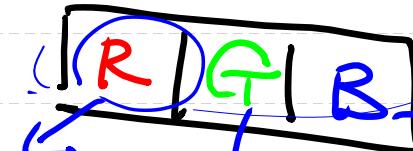
① Pillow library 사용 → Image를 pixel data(ndarray)로 쉽게 변환이 가능!

(컬러, 흑백)

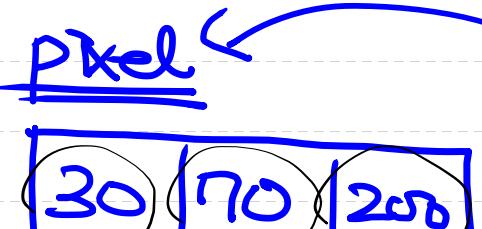
② B는 이미지는 3차원 ndarray를 구성되어 있어요



Pixel (1차원으로 보면)



0~255 0~255 0~255



(만약 흑백 이미지를 처리면
RGB의 풍선은 RGB값을
제거)

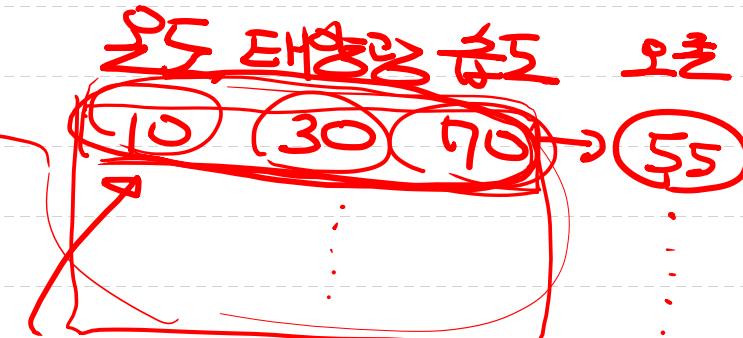


- 이미지 흐름의 문제점.

①

DNN

(feature)
인식데이터가 2차원



→ 디아이터자체는 1차원

이전 디아이터가 여러개 흐름

그럼에도 이미지를 경우

디아이터자체가 3차원 or 2차원

이전 이미지가 여러개 일렬로 사용

4차원 or 3차원

일렬으로 사용하기 위해 이미지를 → 1차원으로 바꿈

"인식데이터가 윗줄"

② pixel로 썩어서 흐름하기 때문에 딱히 어려운 현상이
발생.

우리가 보는 이미지



컴퓨터가 보는 이미지

08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	56	77	91	08
49	49	99	00	17	81	18	57	60	87	17	40	98	43	69	46	04	55	62	00
81	49	31	71	55	79	14	29	93	71	40	67	53	99	30	03	49	13	26	65
52	90	96	24	04	60	11	42	69	24	48	56	01	32	54	71	37	02	34	91
22	31	14	71	51	67	43	59	41	92	34	54	22	40	40	28	44	33	13	70
24	47	32	60	99	03	45	02	44	75	33	53	78	36	64	20	35	09	12	80
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
47	24	10	68	02	62	12	20	95	63	94	39	63	04	49	91	44	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	94	83	14	88	34	89	63	70
21	36	23	09	75	00	74	44	20	45	35	14	00	41	33	97	34	31	33	90
78	17	33	28	22	75	31	67	15	94	03	80	04	42	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	34	29	85	57
84	56	00	48	35	71	89	07	05	44	44	37	44	40	21	58	51	54	17	48
19	80	61	68	05	94	47	49	28	73	92	13	86	52	17	77	04	09	55	40
04	52	08	53	97	35	99	14	07	97	57	32	16	26	26	79	33	17	93	41
04	36	68	81	57	62	20	72	03	16	33	67	46	55	12	32	43	93	53	59
04	42	14	73	38	25	39	11	24	94	72	18	06	46	29	32	40	62	74	36
20	49	34	41	72	30	33	88	34	62	99	69	82	47	59	81	74	14	34	20
20	23	35	29	78	31	90	01	74	31	49	71	48	86	81	14	23	57	05	44
01	70	54	71	83	51	54	49	16	92	33	48	61	43	51	31	89	19	07	43

크게 조정
close 하기



우리가 보는 이미지

pixel 3 버전

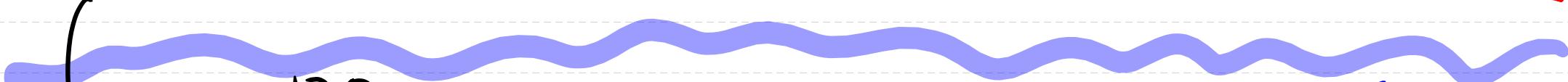
온전히 떨어져요

* 드라이버들이 많이 풀요!

이전 문제점과 같이 Image 흐름은 상당히 결과가 좋지 않아요!

MNIST → 28x28 Size가 적구요
이미지가 복잡하지 않아요.)

but 솔바이어리를 이용하면 상당히 높아요
(DNN을 이용) 50~60%.

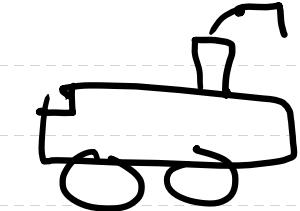
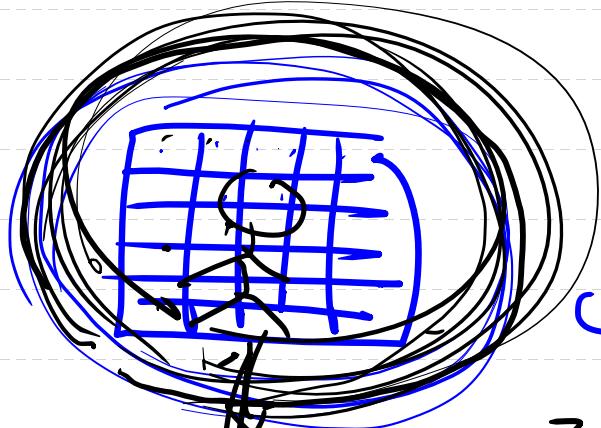
(
어려운
이전 이미지 흐름을 가능토록 한 알고리즘이 등장 → CNN)

Convolutional Neural Network

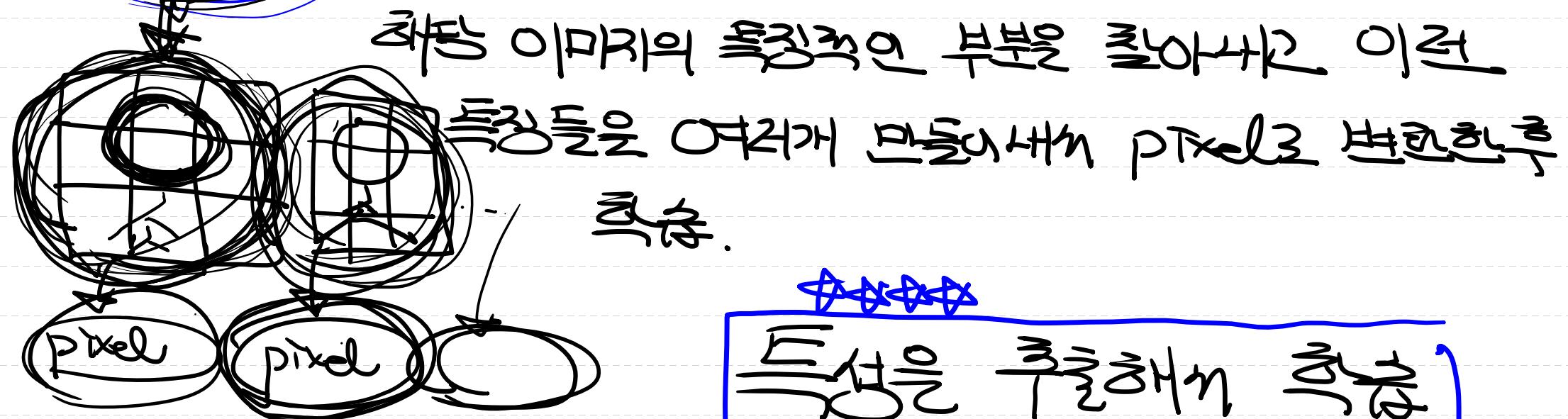
(특성곱 신경망) ↳ 또가요??

CNN의 개념 → 사람이 이미지를 인식하는 것처럼 웃사하게

특징을 추출하자!



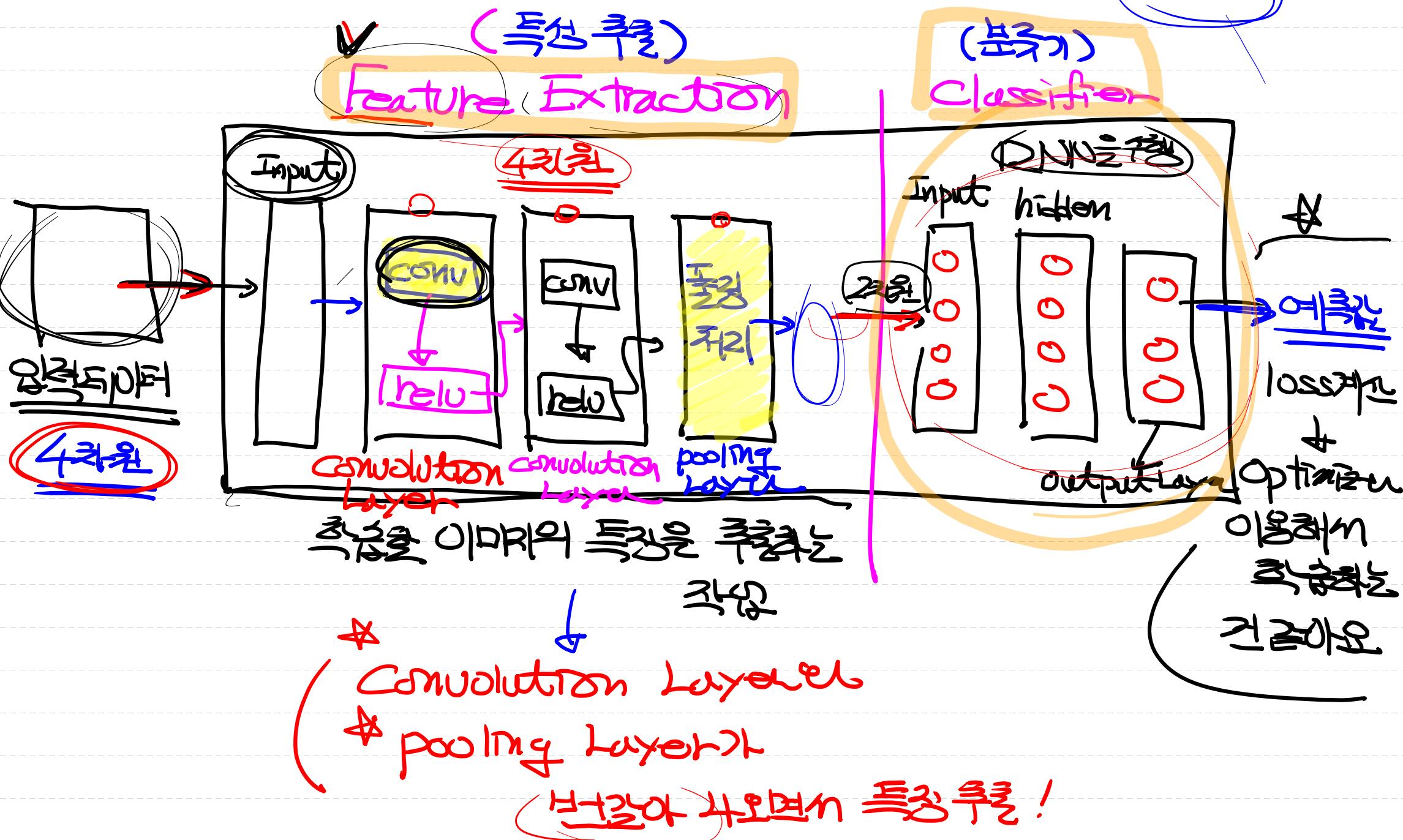
이미지의 pixel을 추출하는 대신 X



특성을 추출해가기

* * *

CNN의 Architecture



① convolution 연산 [conv.]

입력
4x4
→
이미지 3x3
2x2
표현
4x4

이미지 3x3
2x2
표현
4x4

W

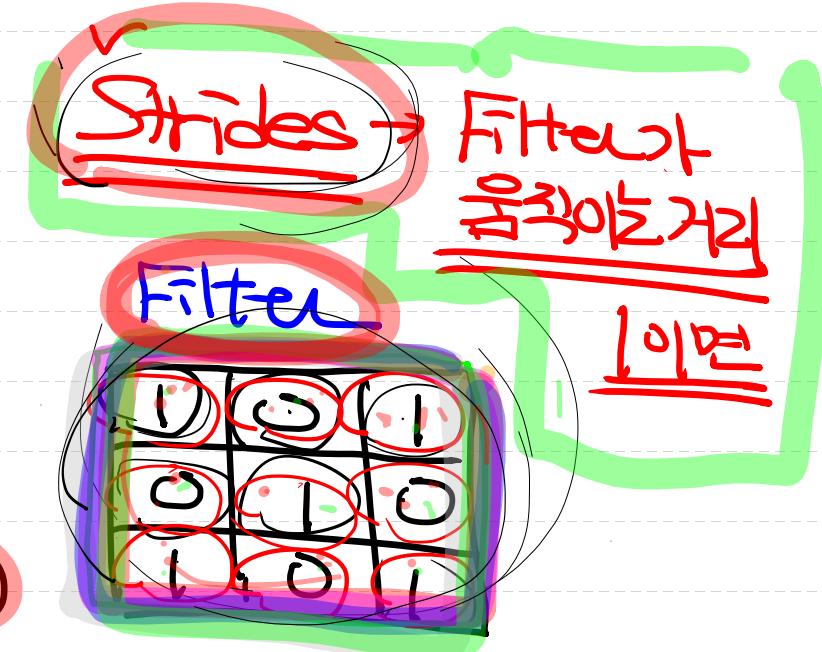
Image pixel data

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

conv

→ 노비

$$1 \times 1 + 1 \times 0 + 1 \times 1 + \\ 0 \times 0 + 1 \times 1 + 1 \times 0 + \\ 0 \times 1 + 0 \times 1 + 1 \times 1 = 4$$

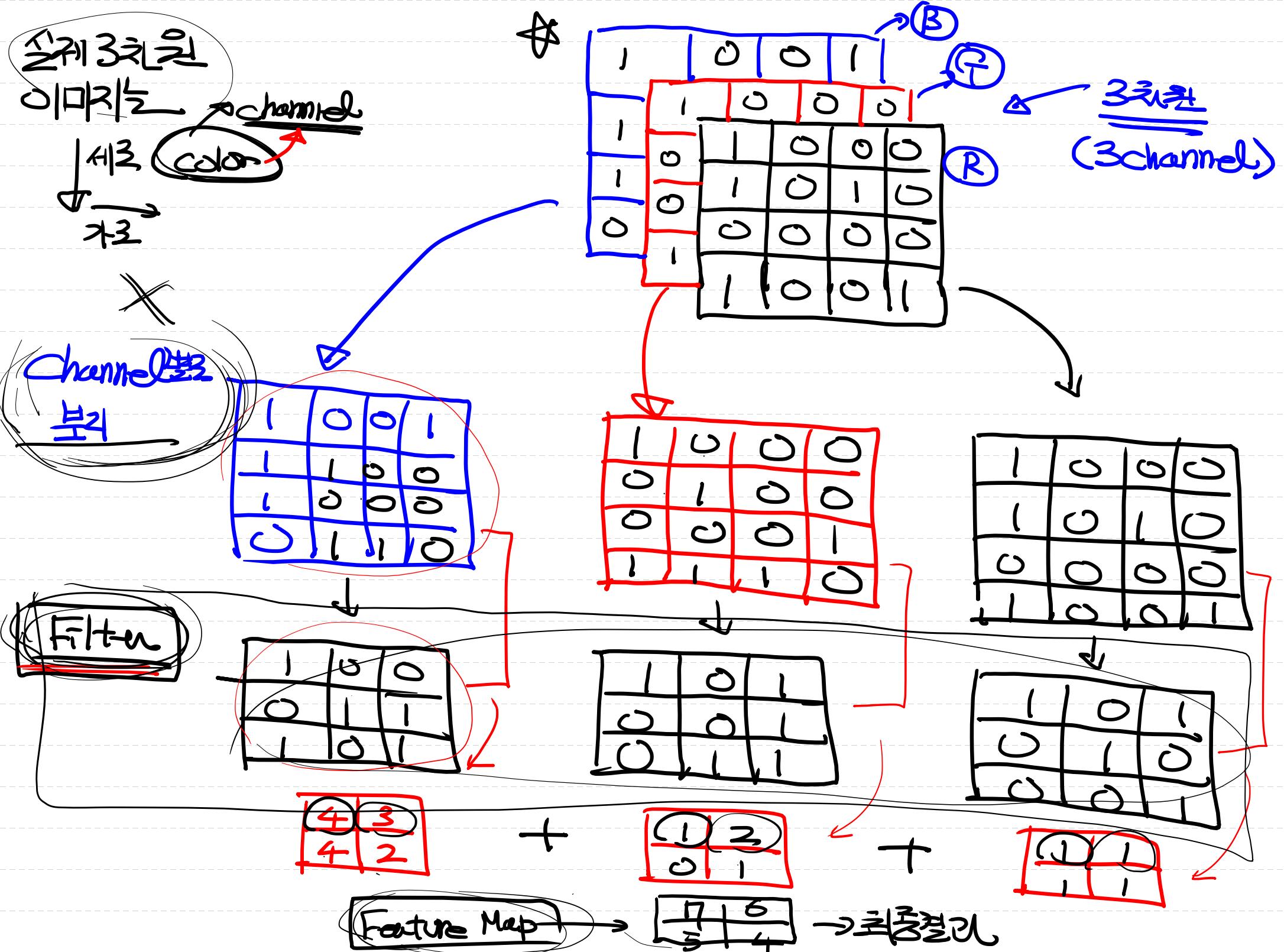


같은 위치에 있는 것끼리 합해가

됌 되해요!

4	3	4
0	0	0
0	0	0

Feature
Map

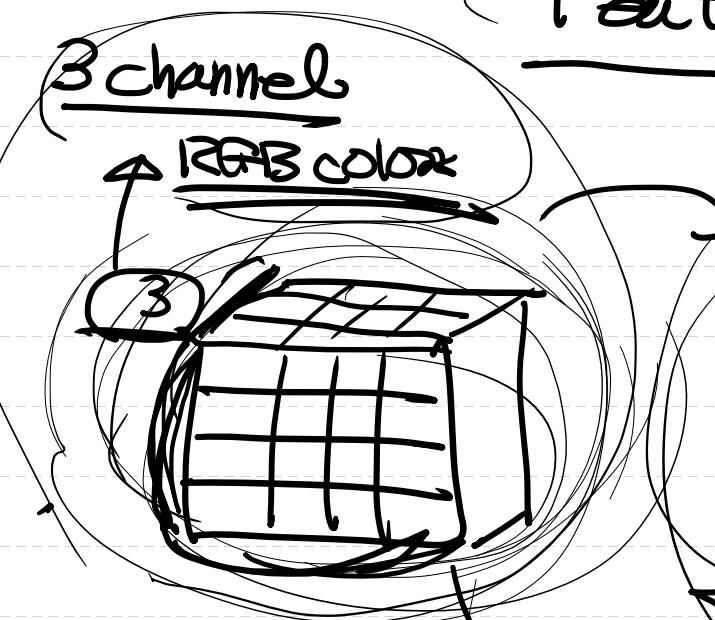


★ Channel의 수와 상관없이

Filter 1개당 Feature Map을 1개여요!!
(2차원)

그러면 Filter 2 여러개 이용하면 (10개 2차원)

Feature Map을 10개로 나눠요



→ Feature Map의
Channel은 10

이미지의 크기가
변경됩니다

Feature Map 1개가 이미지의
특정 내용이에요!!