

YOLO를 이용한 객체 인식

02 Tensorflow로 배우는 CNN

Confidential all rights reserved

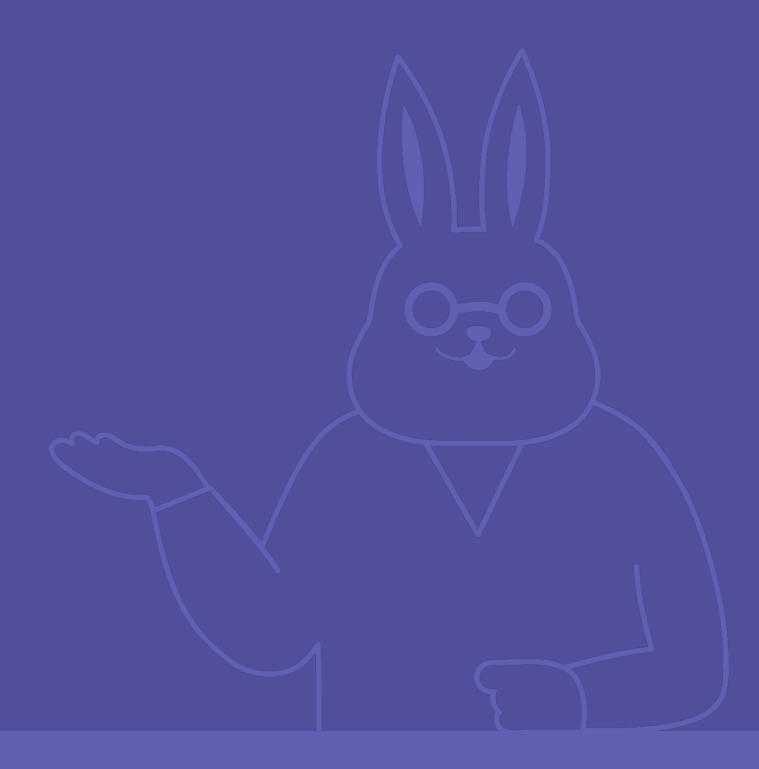
/* elice */

목차

- 01. 영상의 특징량
- 02. 컨볼루션, 패딩, 피처맵
- 03. 스트라이드, 풀링
- 04. CNN개요
- 05. FCN, 활성화 함수, 손실 함수
- 06. CNN의 첫번째 적용 모델 LeNet
- 07. 텐서플로우와 케라스 소개
- 08. 셔플링

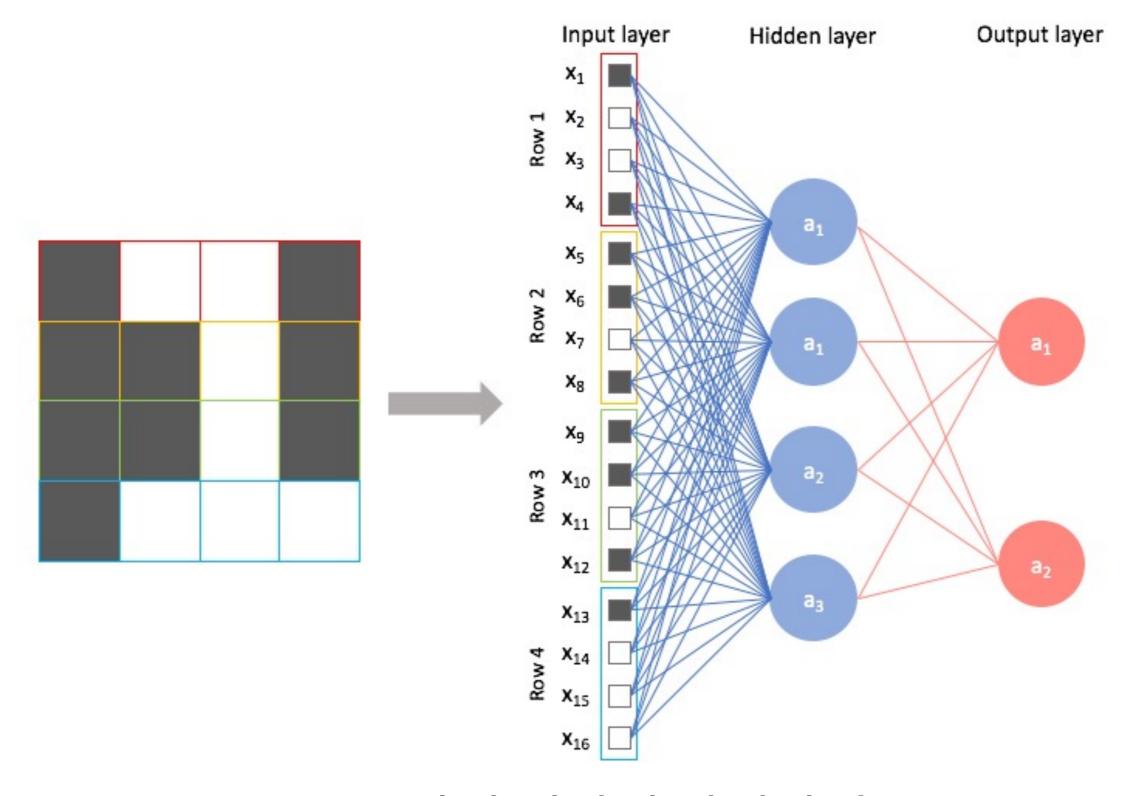
01

영상의 특징량



Confidential all rights reserved

❷ 전통적인 이미지 분류의 접근



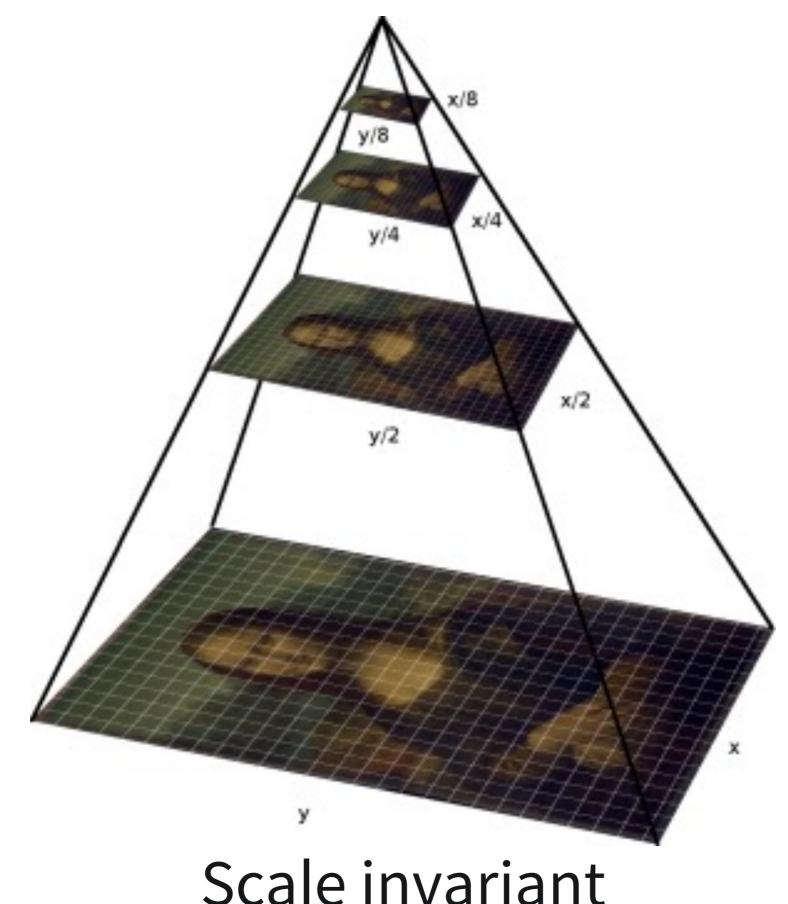
만일 영상이 바뀐다면? 영상의 해상도 엄청 크다면?

❷ 영상의 특징량의 정의

영상을 수치적 값의 관점에서 추출한 정보

영상의 특징량을 추출하는 이유는 영상의 정보를 낮은 차원으로 처리할 수 있기 때문

❷ 데이터의 관점에서 영상의 특징량을 이용해서 분류하는 이유



Scale invariant

01 영상의 특징량

❷ 데이터의 관점에서 영상의 특징량을 이용해서 분류하는 이유



intra-class variation

inter-class variation

intra-class variation

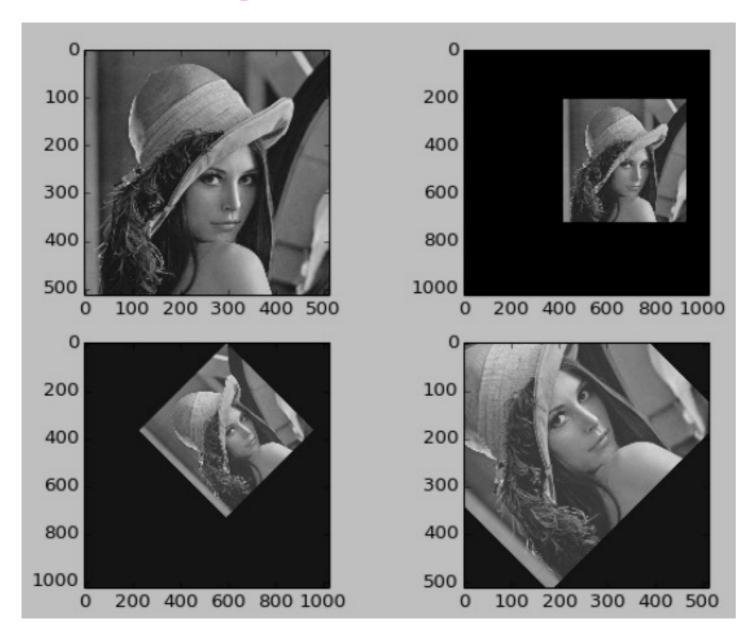
☑ 데이터의 관점에서 영상의 특징량을 이용해서 분류하는 이유



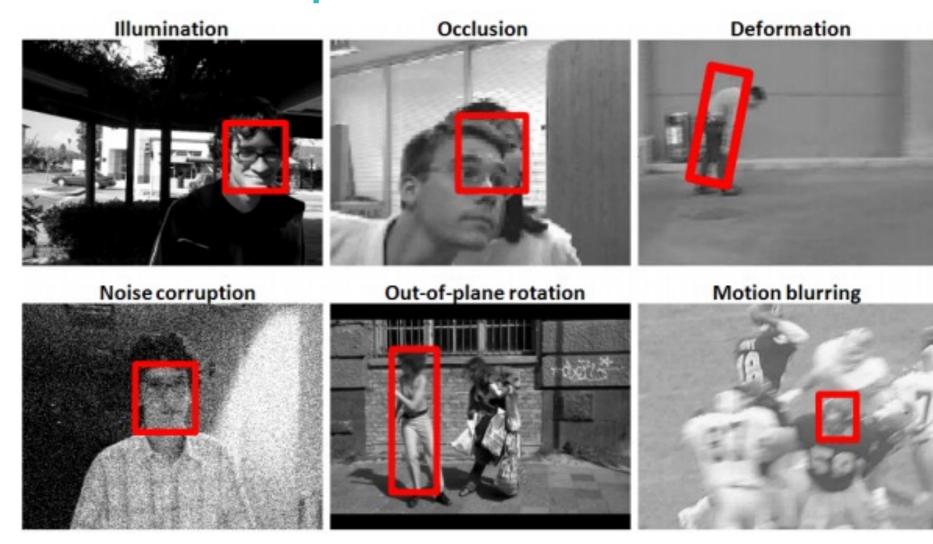
Deformation, Occlusion, Illumination change

좋은 특징량의 조건1 − Repeatability

geometric



photometric



geometric과 photometric 변환에도 불구하고 똑같은 영상에서 동일한 특징이어야 함

좋은 특징량의 조건2 – Saliency and Locality



영상의 interesting point를 포함하여야 하며 영상 내의 작은 영역을 차지해야 함

❷ 영상의 특징량의 종류 – Local VS Global

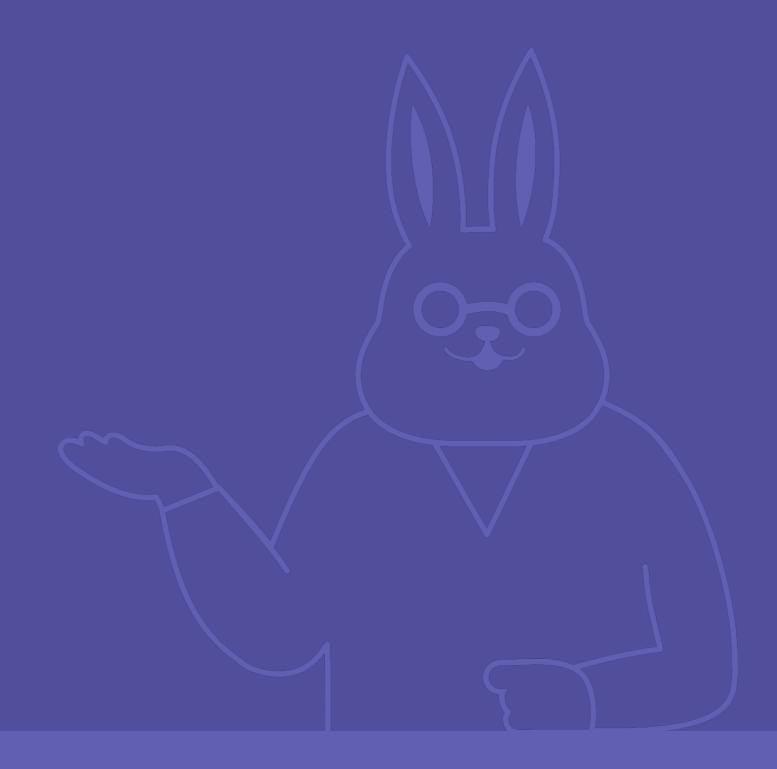




Global

Local

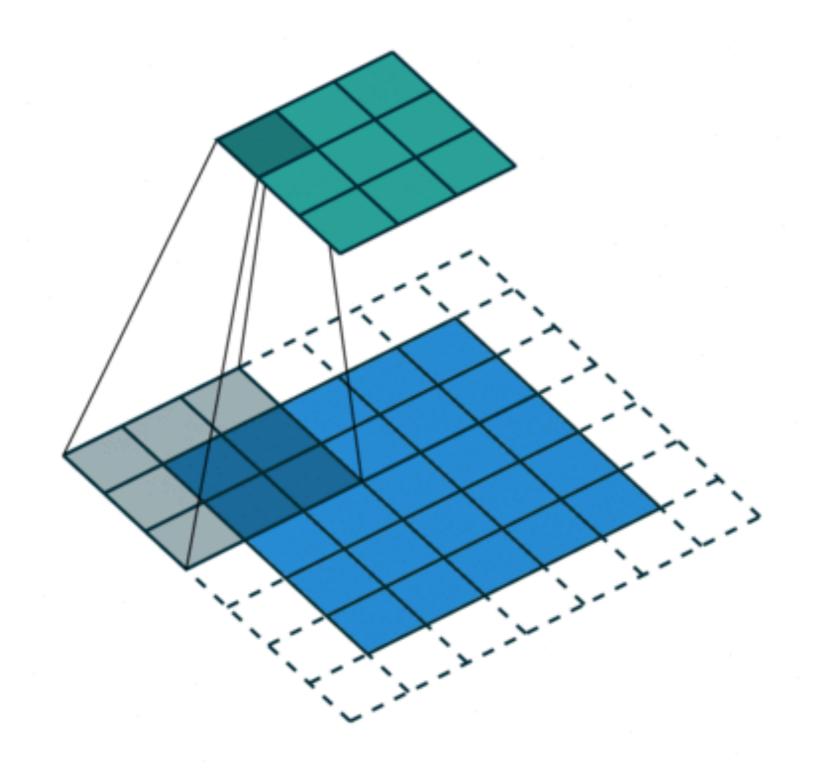
컨볼루션과 패딩, 피처맵



Confidential all rights reserved

02 컨볼루션과 패딩, 피처맵

❷ 컨볼루션



1 _{×1}	1,0	1 _{×1}	0	0
0 _{×0}	1 _{×1}	1 _{×0}	1	0
0 _{×1}	0,×0	1 _{×1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

4	

Convolved Feature

♥ 컨볼루션

5	5	5	5	5	5
5					
5	5	5			
		5	5	5	5
					5
	5	5	5	5	

0	0	0
1	1	1
0	0	0

0	0	0			
5	5	5	5	5	5
6	0	0			
5	5	5			
		5	5	5	5
					5
	5	5	5	5	

0	0	0
1	1	1
0	0	0

10	15		

	0	0	0		
5	5	5	5	5	5
5	0	0	0		
5	5	5			
		5	5	5	5
					5
	5	5	5	5	

0	0	0
1	1	1
0	0	0

10	15	15		

				0	0
5	5	5	5	5	5
5				0	0
5	5	5			
		5	5	5	5
					5
	5	5	5	5	

0	0	0
1	1	1
0	0	0

10	15	15	15	15	10

0	6	6	5	5	5	5
1	5	1				
0	5	6	5			
			5	5	5	5
						5
		5	5	5	5	

0	0	0
1	1	1
0	0	0

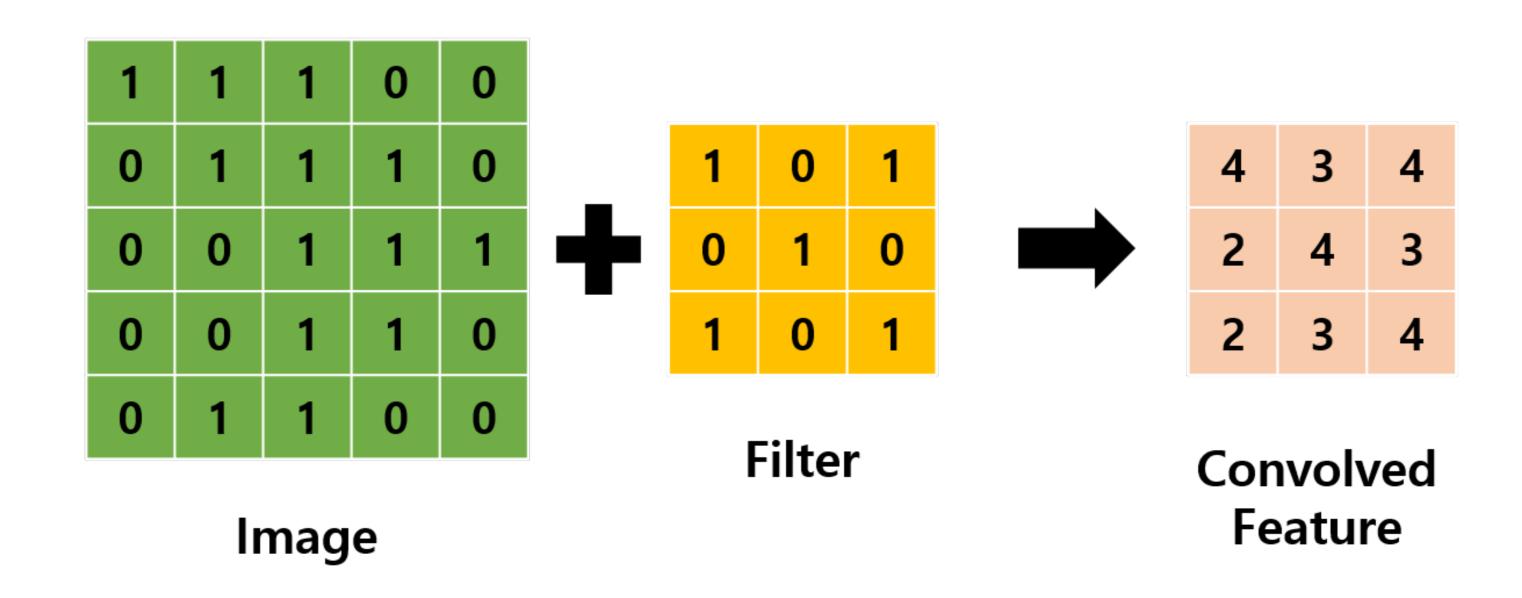
10	15	15	15	15	10
5					

5	5	5	5	5	5
5					
5	5	5			
		5	5	5	5
					5
	5	5	5	5	

0	0	0
1	1	1
0	0	0

10	15	15	15	15	10
5	5	0	0	0	0
10	15	10	5	0	0
0	5	10	15	15	10
0	0	0	0	5	5
5	10	15	15	10	5





컨볼루션을 계속 하다보면 점점 이미지가 작아짐



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

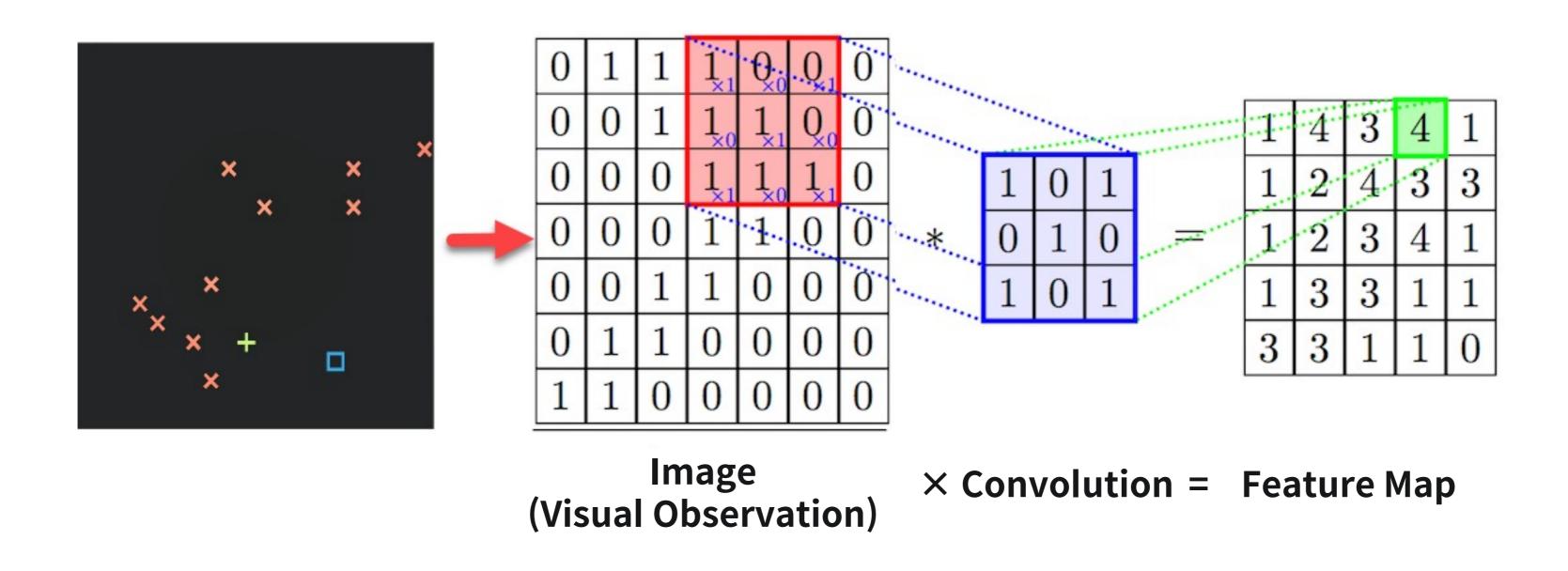
0	0	0	0	0	0	0
0	60	113	56	139	85	0
0	73	121	54	84	128	0
0	131	99	70	129	127	0
0	80	57	115	69	134	0
0	104	126	123	95	130	0
0	0	0	0	0	0	0

Kernei					
0	-1	0			
-1	5	-1			
0	-1	0			

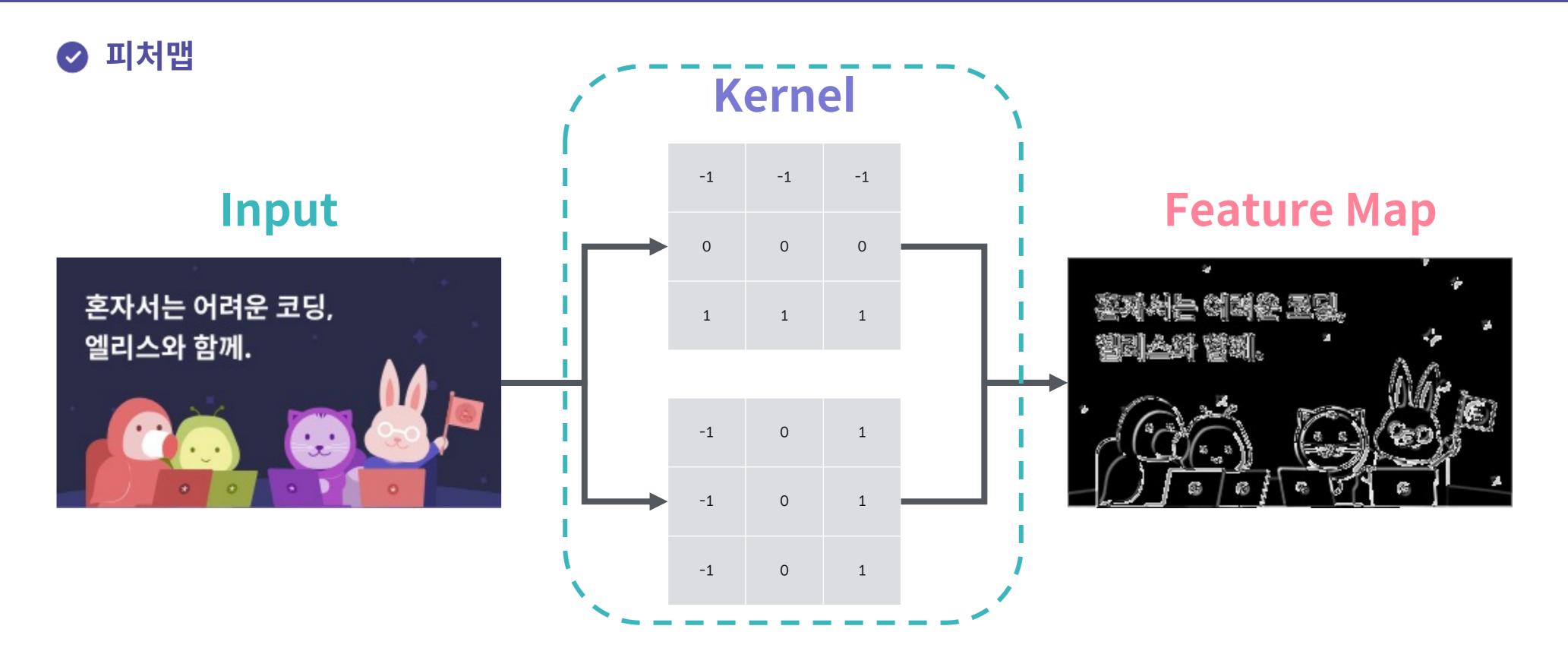
114		

Image (+zero Padding)

☑ 피처맵

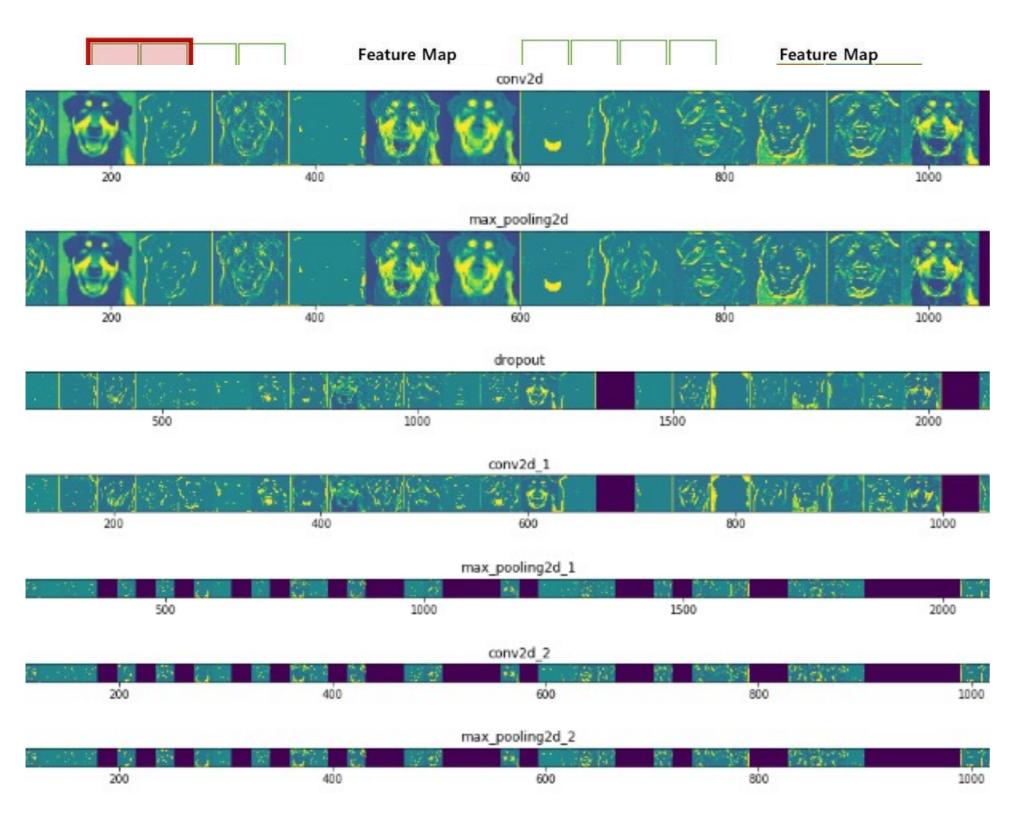


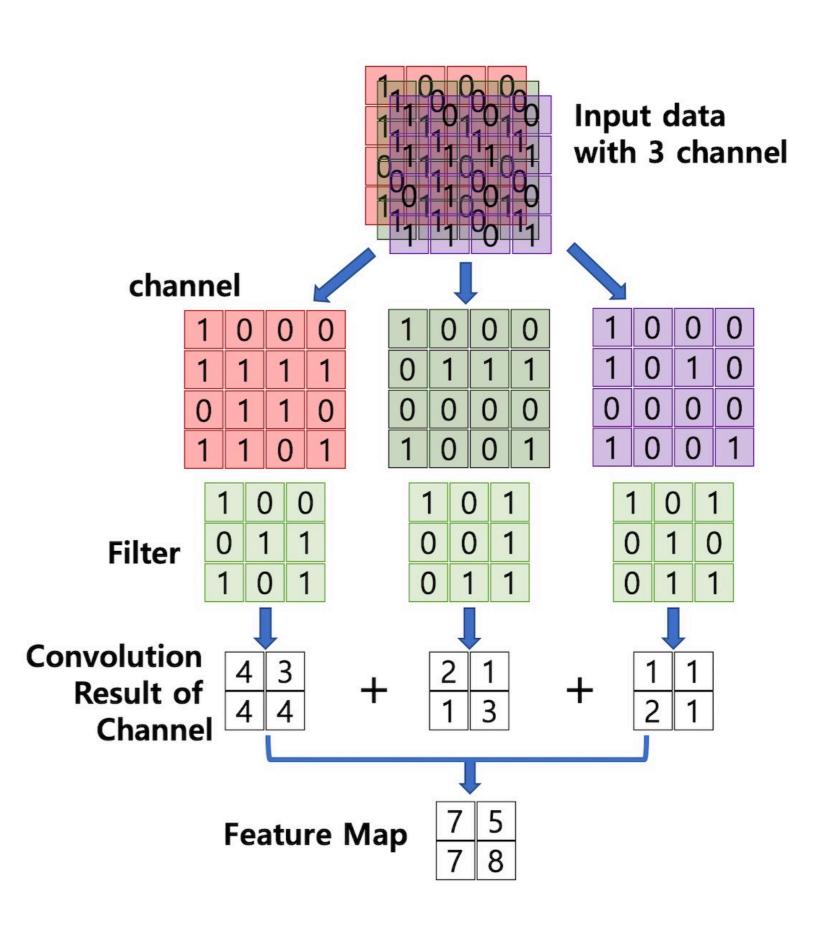
피처맵이란 컨볼루션의 결과를 의미 즉, 영상이 필터를 통과한 결과 행렬이 Feature Map 02 컨볼루션과 패딩, 피처맵



피처맵이란 컨볼루션의 결과를 의미 즉, 영상이 필터를 통과한 결과 행렬이 Feature Map

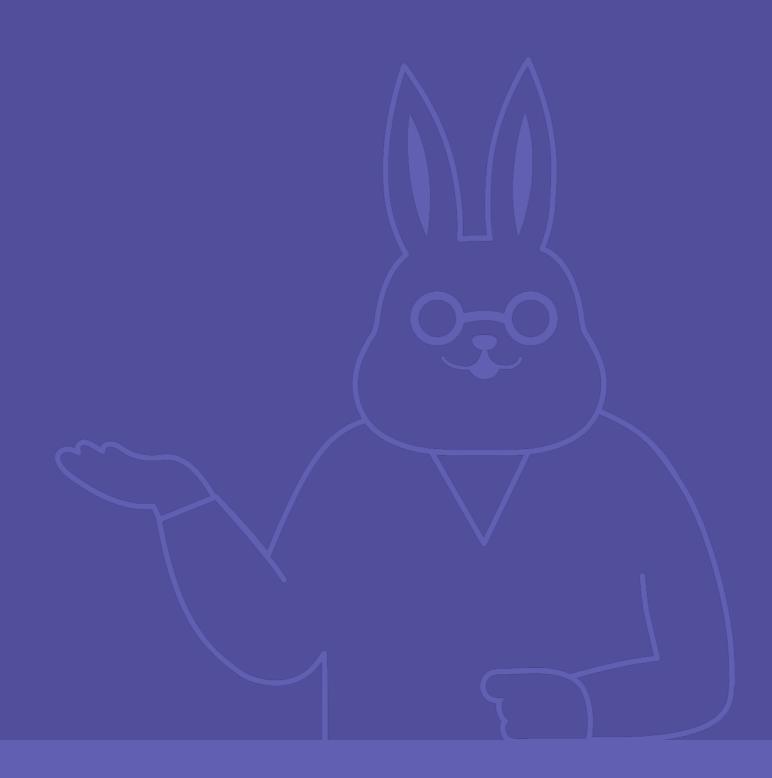
❷ 컬러 영상의 피처맵





컬러 채널별로 컨볼루션 수행 후 채널별 결과를 합산

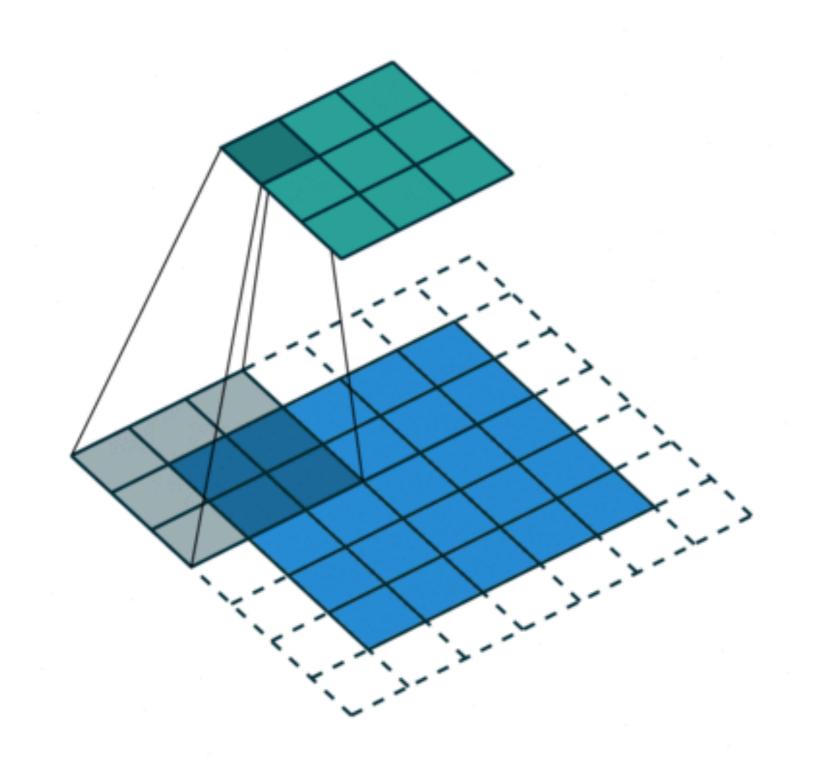
스트라이드와 풀링



Confidential all rights reserved

03 스트라이드와 풀링

❷ 스트라이드



02	00	0,	0	0	0	0
0,	2_0	2_0	3	3	3	0
00	0,	1,	3	0	3	0
0	2	3	0	1	3	0
0	3	3	2	1	2	0
0	3	3	0	2	3	0
0	0	0	0	0	0	0

1	6	5
7	10	9
7	10	8

스트라이드란 이미지를 횡단할 때 커널의 스텝 크기 기본값은 1임

10	15	15	15	15	10
5	5	0	0	0	0
10	15	10	5	0	0
0	5	10	15	15	10
0	0	0	0	5	5
5	10	15	15	10	5

0	0	0
1	1	1
0	0	0

영상

커널

10	15	15	15	15	10
5	5	0	0	0	0
10	15	10	5	0	0
0	5	10	15	15	10
0	0	0	0	5	5
5	10	15	15	10	5

0	0	0
1	1	1
0	0	0

10	

커널

피처맵



Stride: 2

10	15	15	15	15	10
5	5	0	0	0	0
10	15	10	5	0	0
0	5	10	15	15	10
0	0	0	0	5	5
5	10	15	15	10	5

0	0	0
1	1	1
0	0	0

10	0	

커널

피처맵

Stride: 2

10	15	15	15	15	10
5	5	0	0	0	0
10	15	10	5	0	0
0	5	10	15	15	10
0	0	0	0	5	5
5	10	15	15	10	5

0	0	0
1	1	1
0	0	0

10	0	X
15		

커널

피처맵

Stride: 2

10	15	15	15	15	10
5	5	0	0	0	0
10	15	10	5	0	0
0	5	10	15	15	10
0	0	0	0	5	5
5	10	15	15	10	5

0	0	0	
1	1	1	
0	0	0	

10	0	X
15	40	

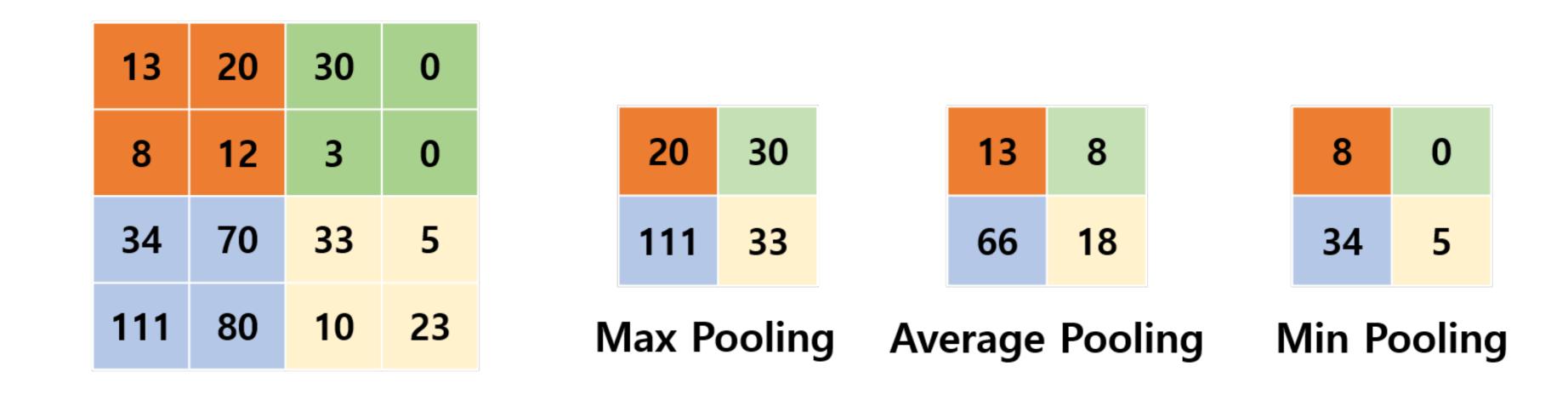
커널

피처맵

Activation Map



풀링



영상에서 특징점만 강조하면서도 차원을 낮추기 위해서 사용

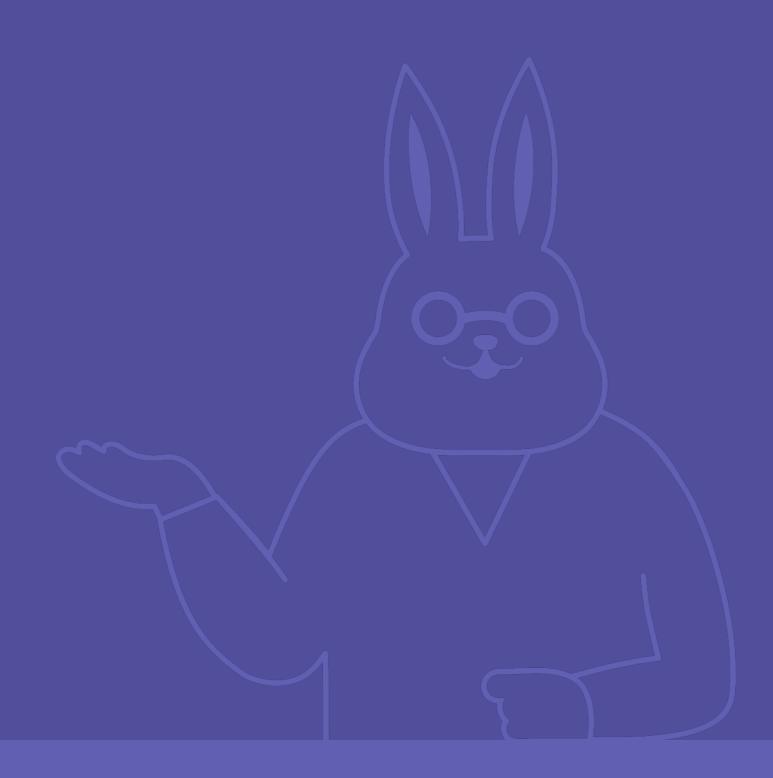
❷ 스트라이드와 풀링의 사용

영상을 다운 샘플링(표본화)을 하기 위해서 주로 사용 정보의 수를 줄여 추론속도를 높이기 위함

스트라이드는 단순히 다운 샘플링만 하지만 풀링은 다운 샘플링만 하면서 **특징점을 강조하는 효과가 있음** 그러나 스트라이드가 처리속도에 이점이 있음

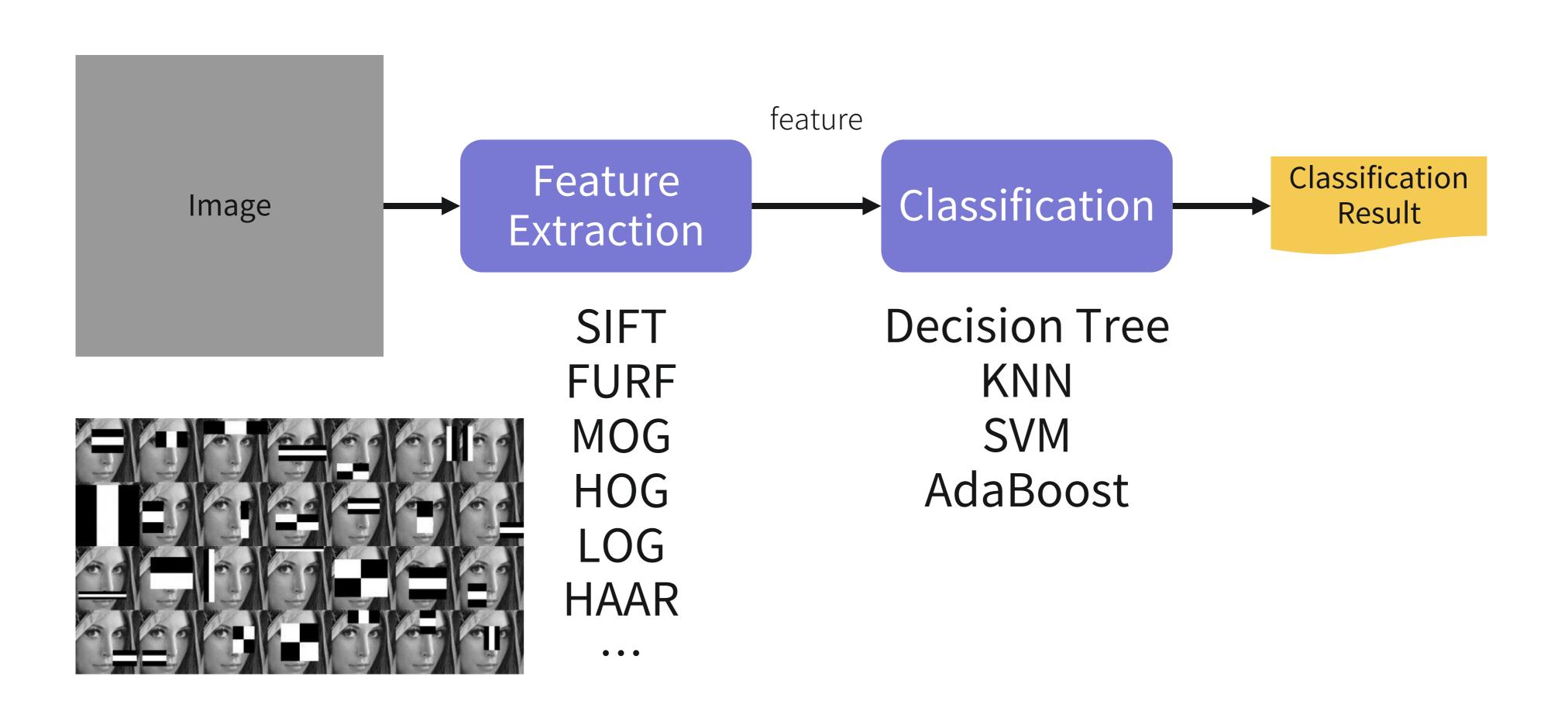
따라서 풀링과 스트라이드를 속도와 인식률 측면에서 적절히 트레이드 오프하여 사용 04

CNN 7H A

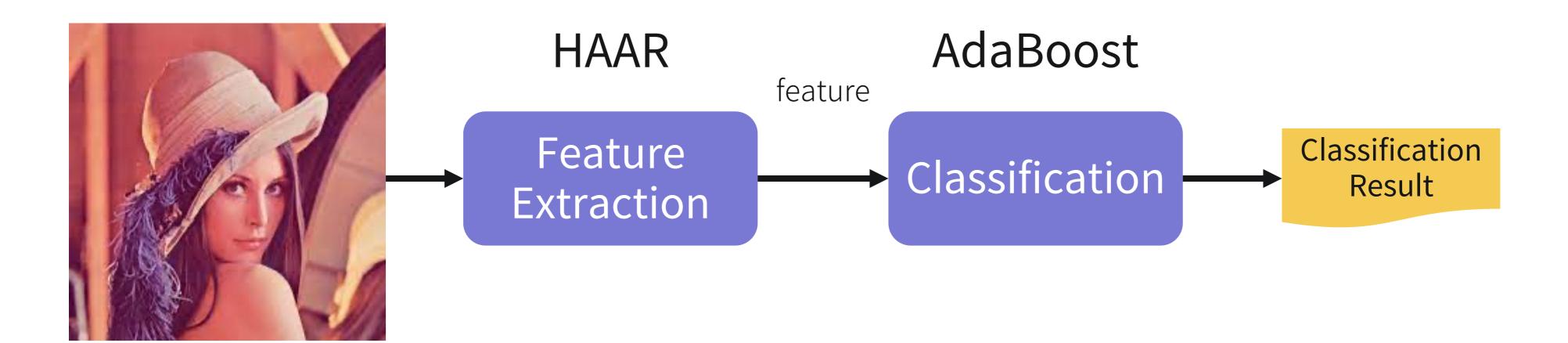


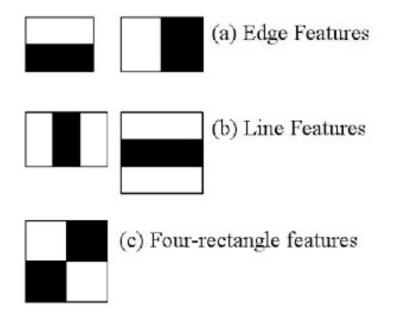
Confidential all rights reserved

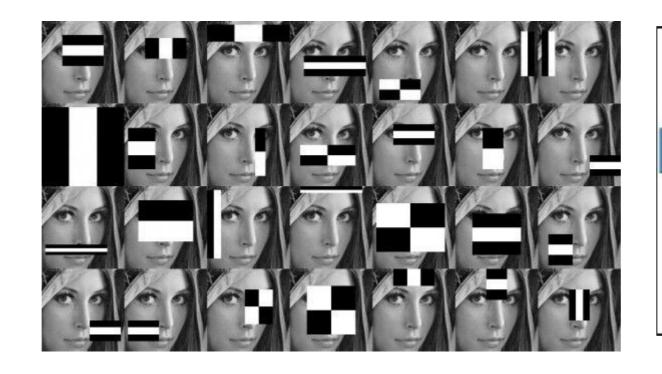
❷ 영상 분류 시스템

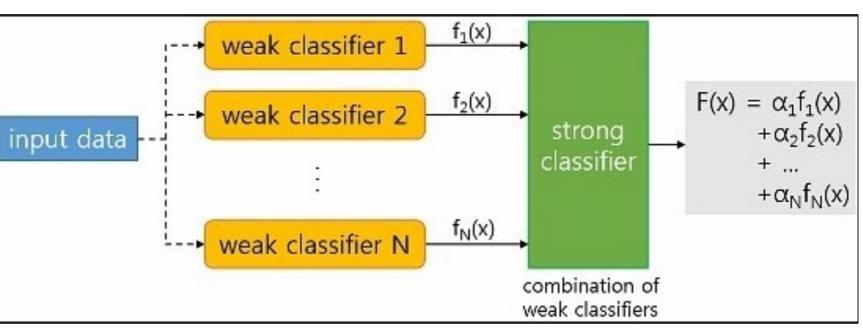


❷ 영상 분류 시스템 예시

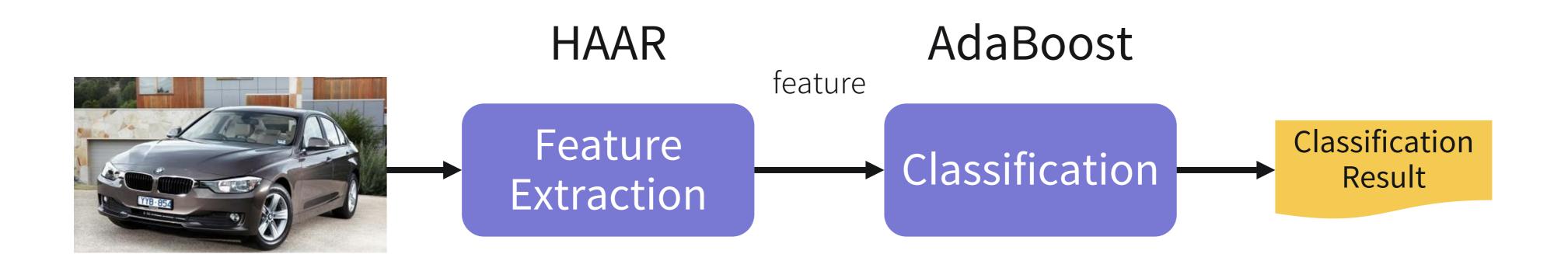


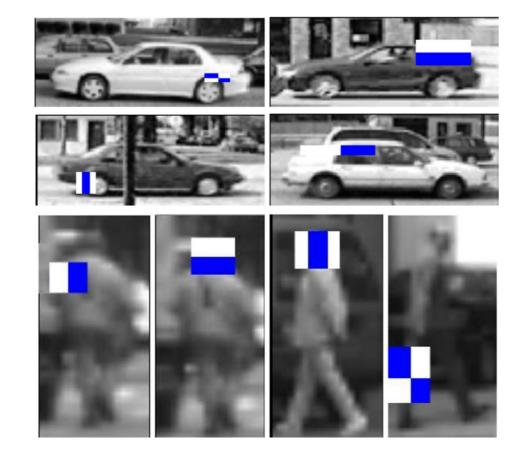


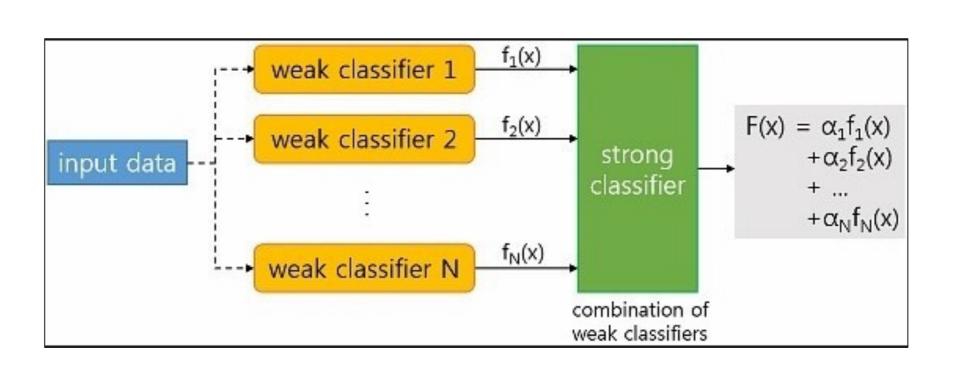




❷ 영상 분류 시스템 예시 – 차로 바뀌면?

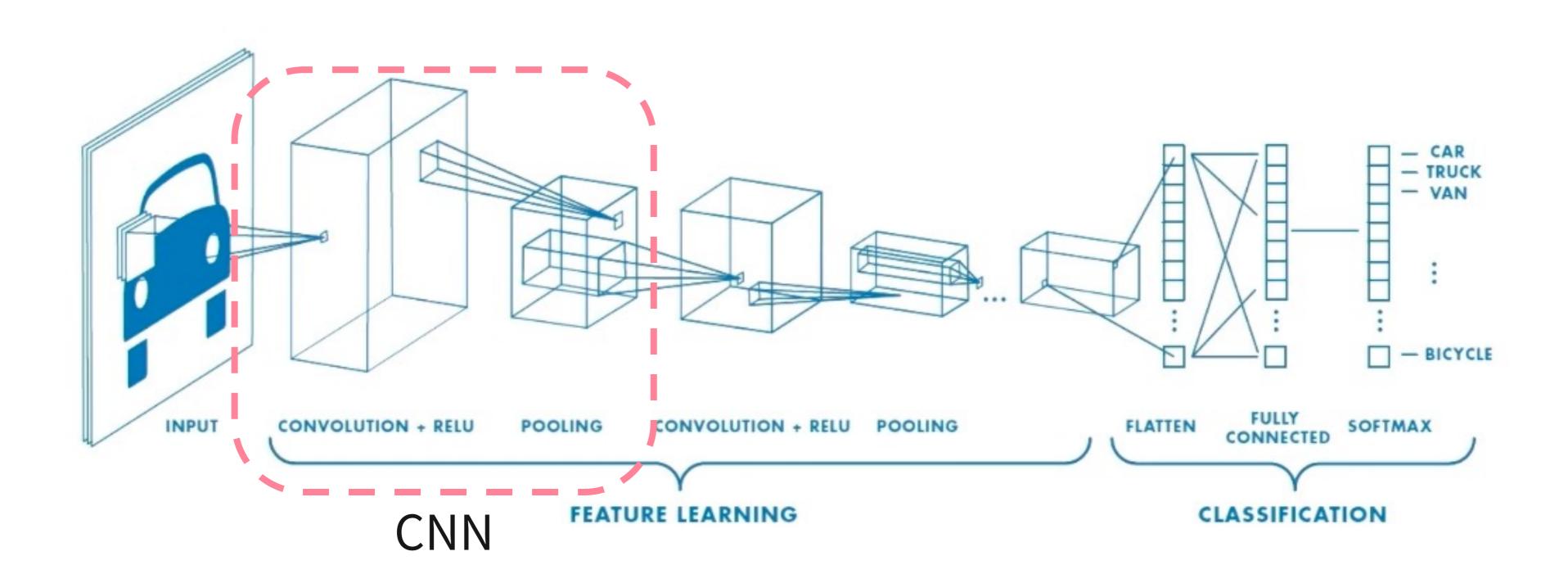






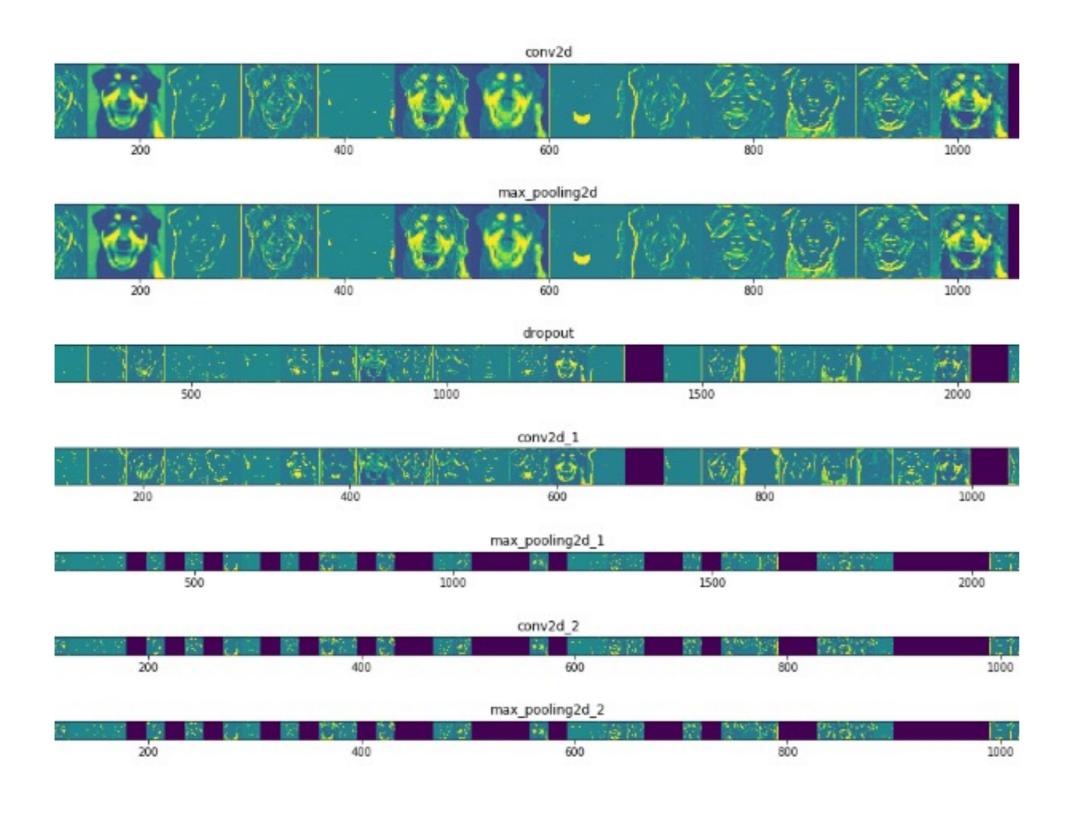
04 CNN 개요

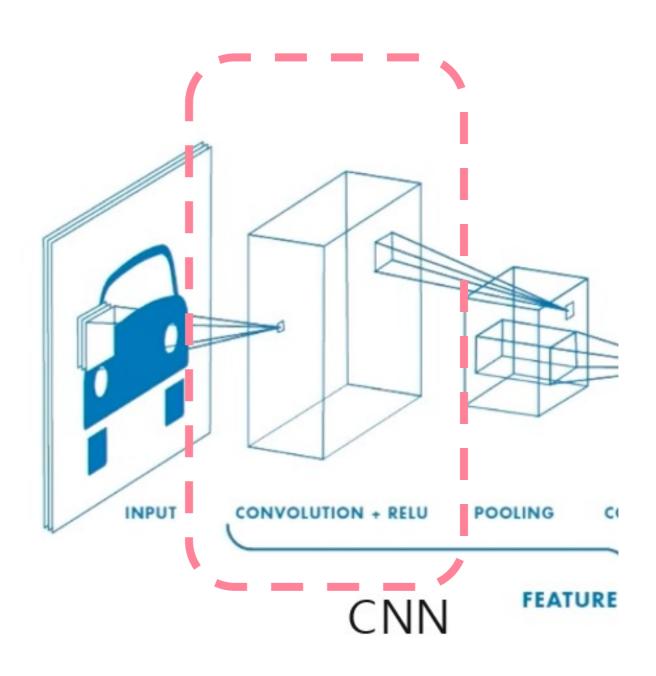
♥ CNN 네트워크 디자인



04 CNN 개요

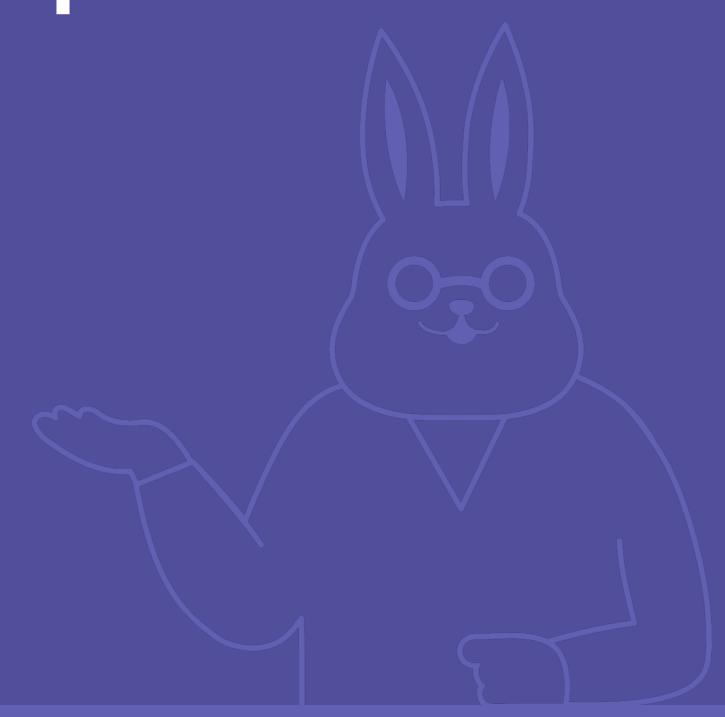
◎ 여러 개의 컨볼루션 필터를 조합하여 다양한 피처맵 추출





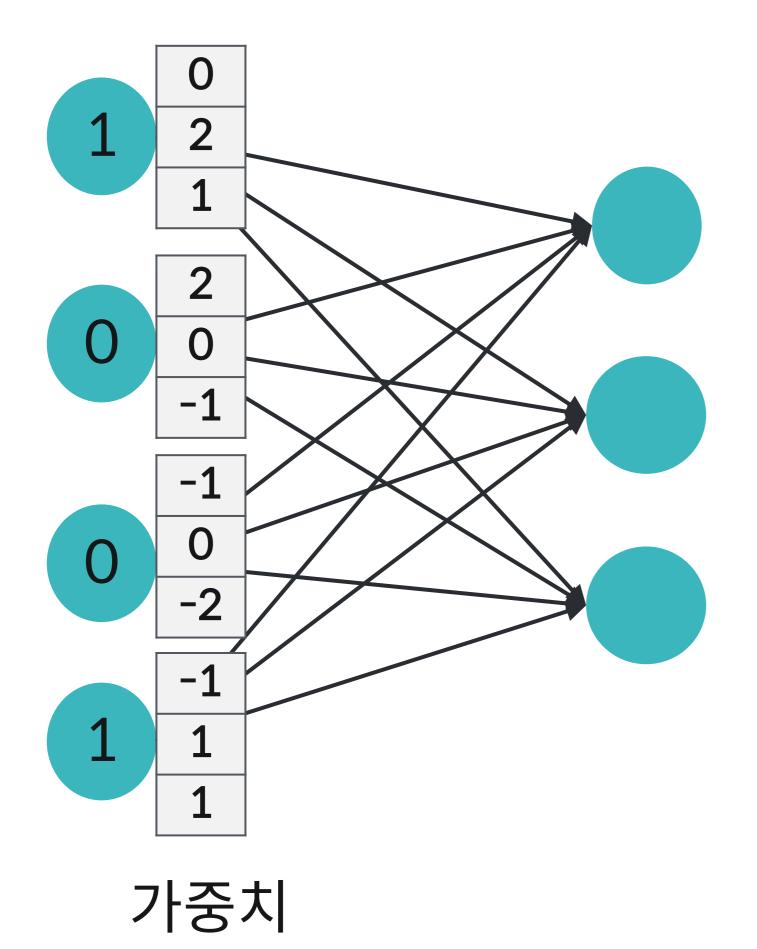
컨볼루션 필터를 여러 개를 사용할 수 있음

FCN, 활성화 함수, 손실 함수

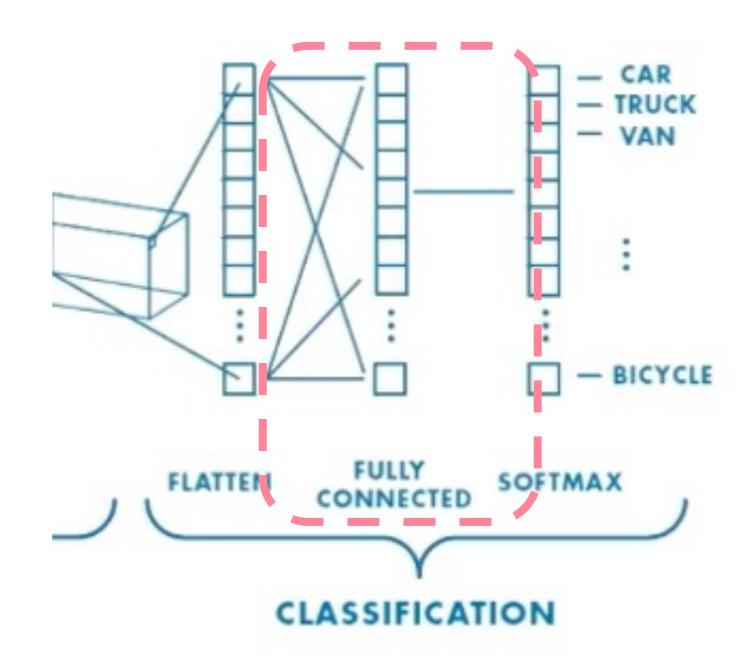


Confidential all rights reserved

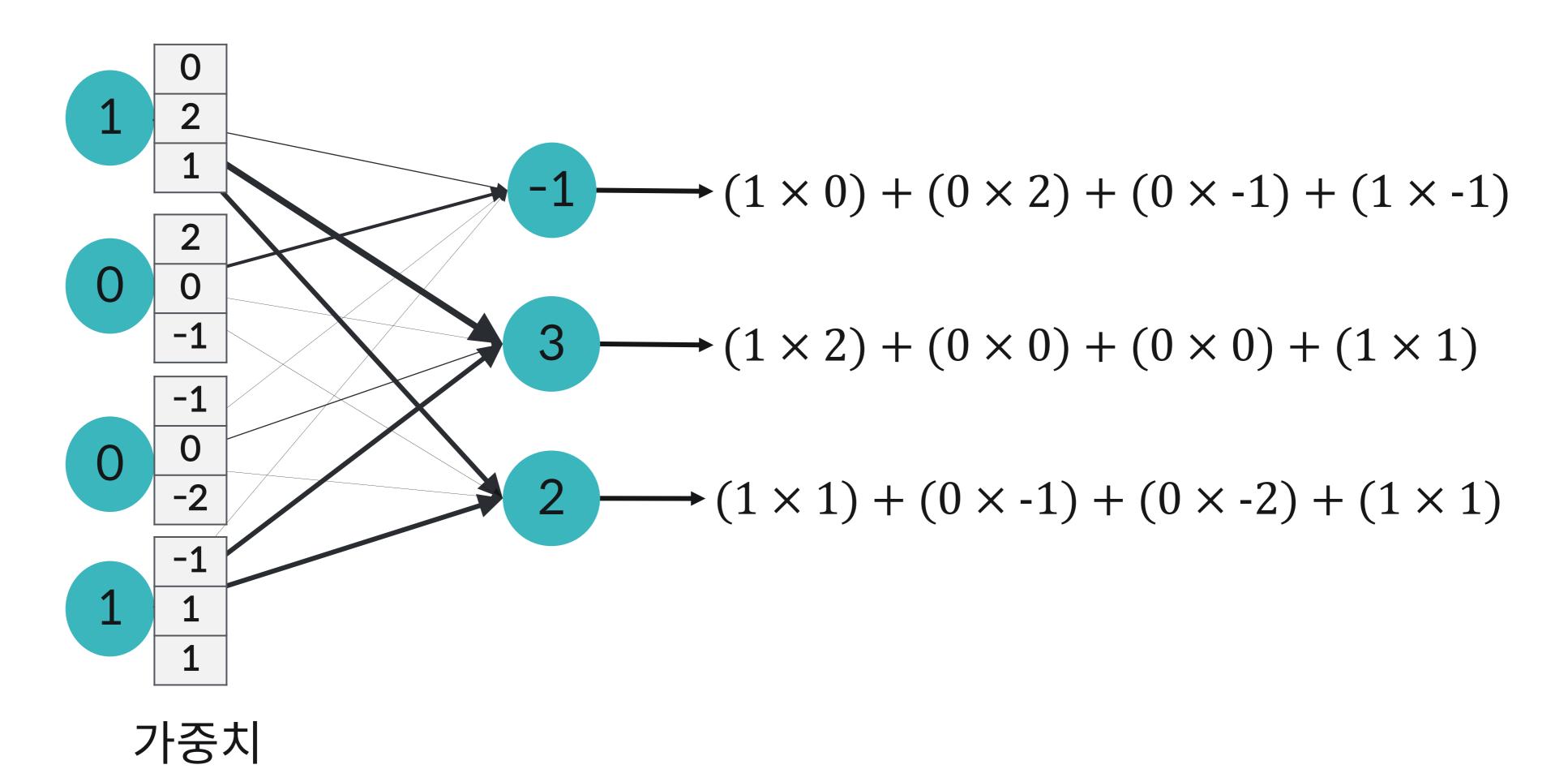
❷ Fully connected layer란



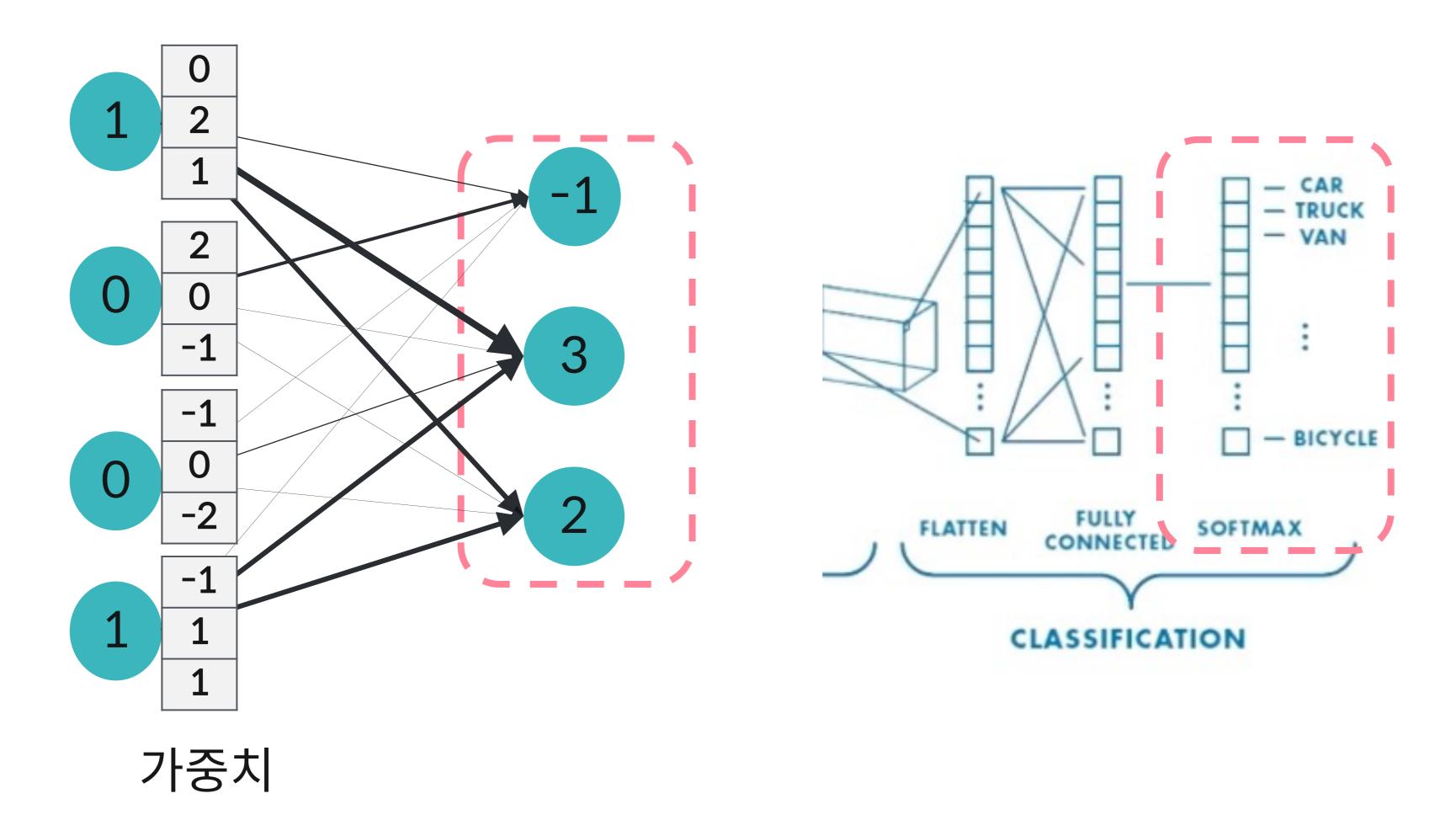
Class 분류 네트워크 마지막에 사용됨 모든 뉴런들이 연결되는 네트워크



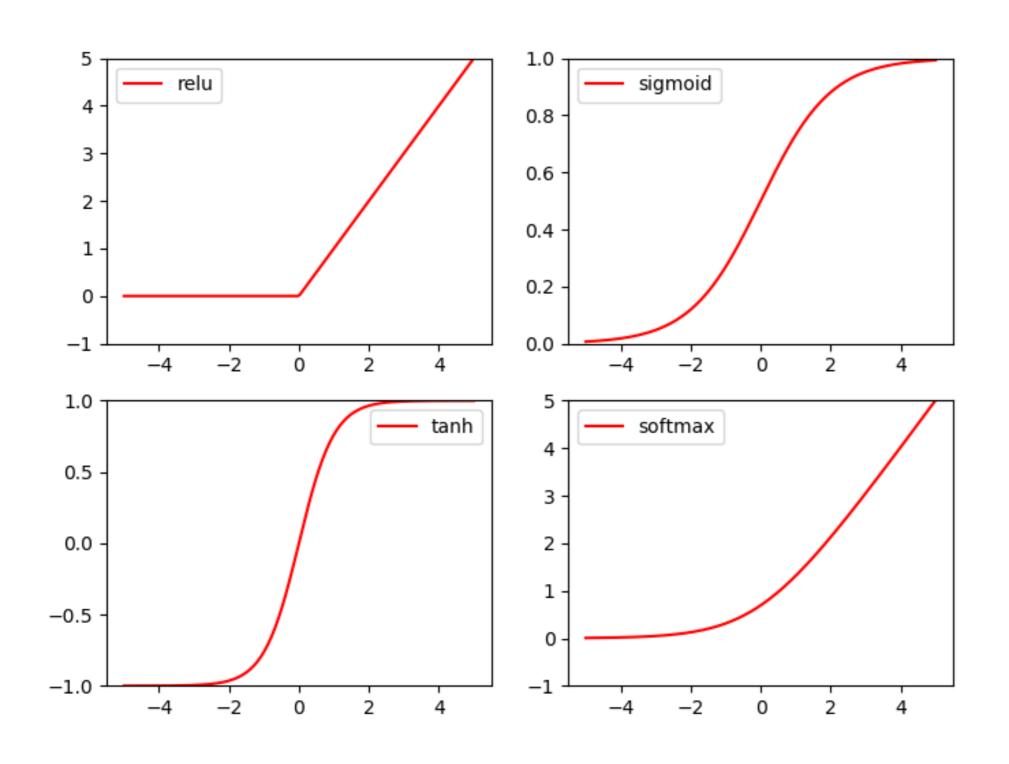
Fully connected layer



❷ 활성화 함수



❷ 활성화 함수란

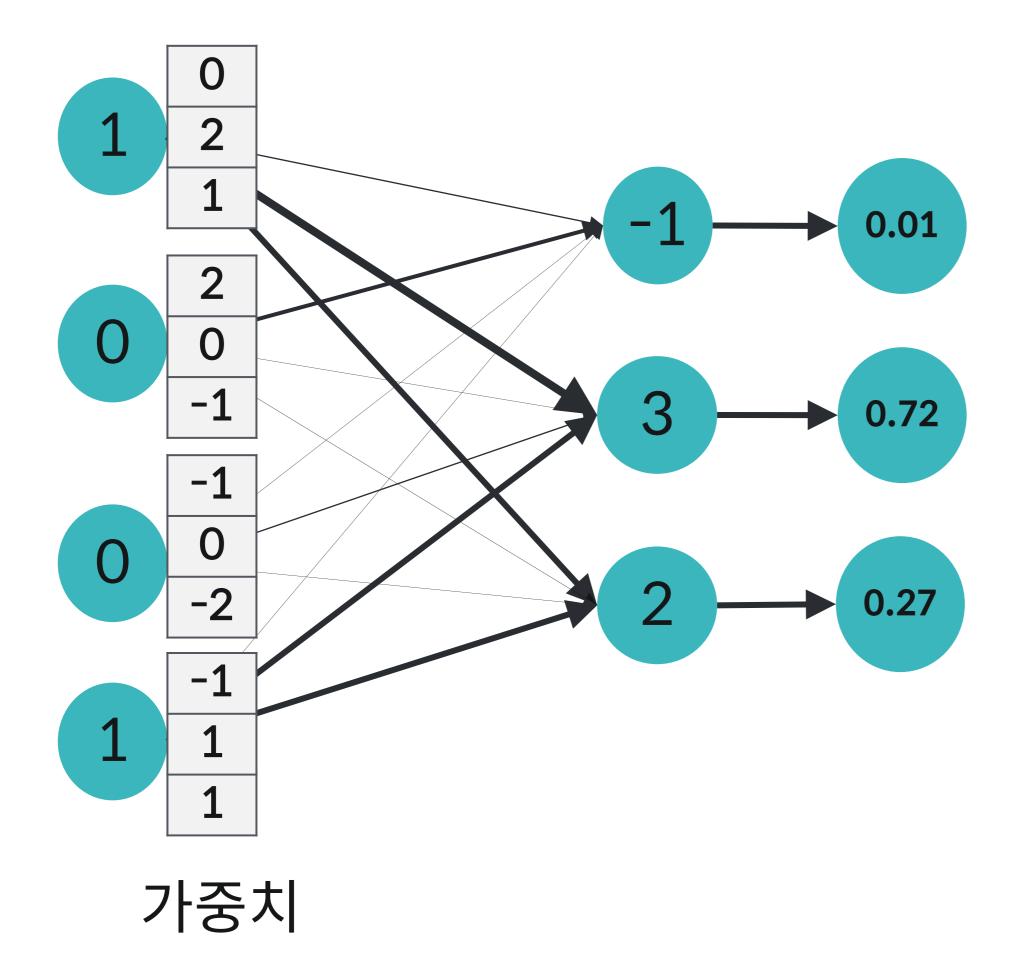


입력 노드를 On / Off 로 표현할 수 있도록 이진 분류해주는 함수

자극의 정도에 따라 On과 Off를 구분해주는 tanh Sigmoid

특정 세기 이상의 자극이 오면 활성화가 되는 Relu, softmax

SoftMax

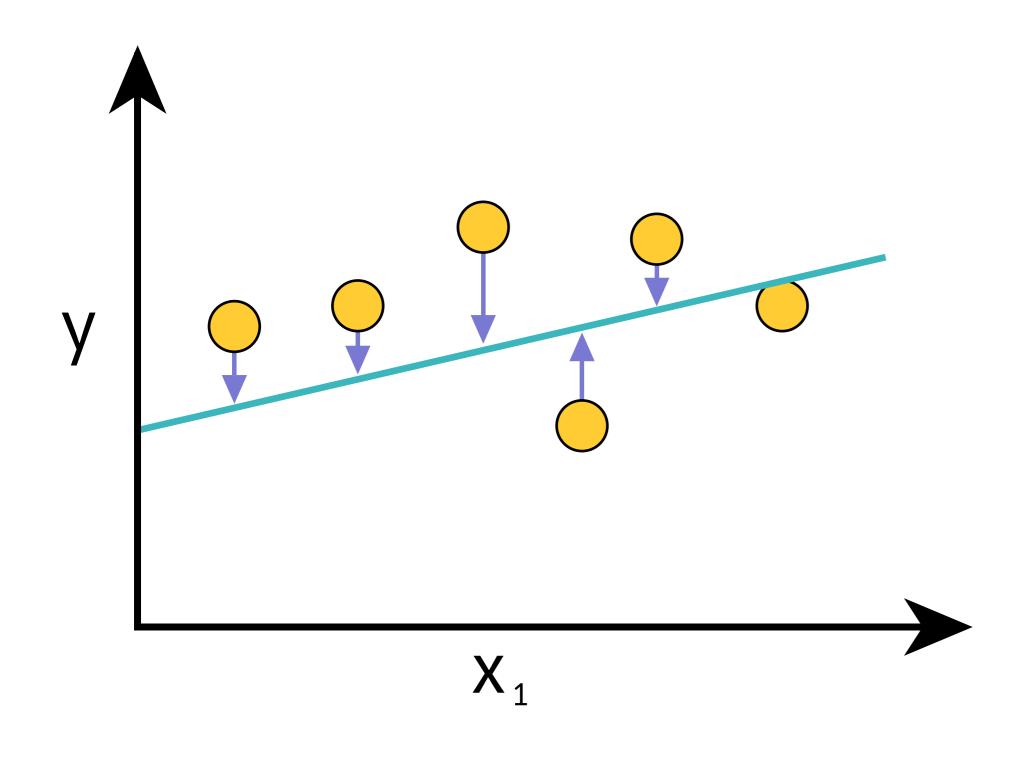


SoftMax

Class 분류 네트워크에서 결과 레이어 값이 일정하지 않을 수 있음 이를 0~1사이 값으로 일반화 하기 위한 함수

각 Class로 분류될 확률을 0~1 벡터로 표현 각 벡터의 합이 1이 되는 활성 함수

❷ 손실 함수



손실 함수 = 비용 함수 = 목적 함수

모델을 학습할 때 쓰이는 함수로 예측값과 기준값의 차이를 수치화

손실 함수 결과가 클수록 차이가 큰 것 손실 함수 결과가 작을수록 오차가 적은 것

MSE =
$$\frac{1}{n} \sum_{i=1}^{n} (y_i - \tilde{y}_i)^2$$

❷ 손실 함수: MSE

정답	0	0	1	0	0
예측	0.1	0.1	0.6	0.1	0.1

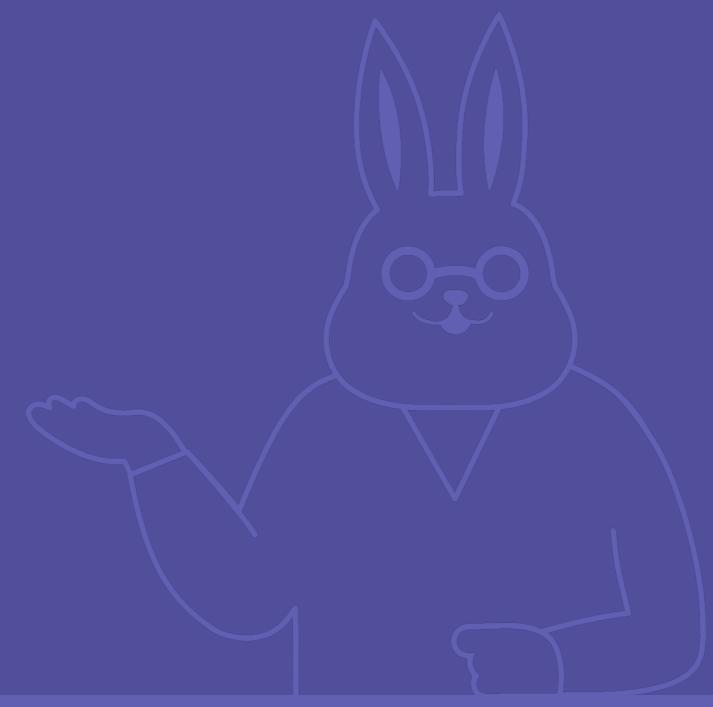
차이	-0.1	-0.1	-0.4	-0.1	-0.1
제곱	0.01	0.01	0.16	0.01	0.01

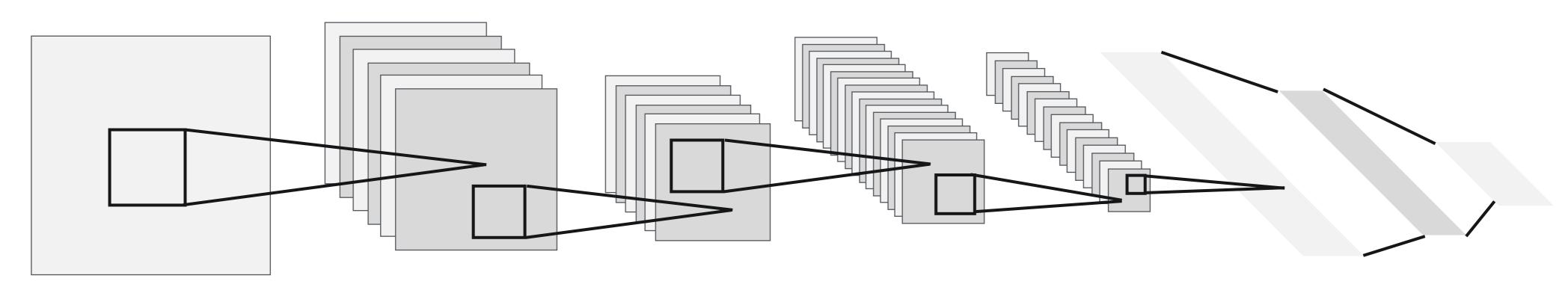
정답	0	0	1	0	0
예측	0.2	0.2	0.2	0.2	0.2

차이	-0.2	-0.2	0.8	-0.2	-0.2
제곱	0.04	0.04	0.64	0.04	0.04

평균	0.16
병관	0.16

CNN의 첫번째 적용 모델, LeNet





Conv

Subsampling

Conv

Subsampling

Conv

FC

FC

Input 32x32

6@28x28

6@14x14

16@10x10

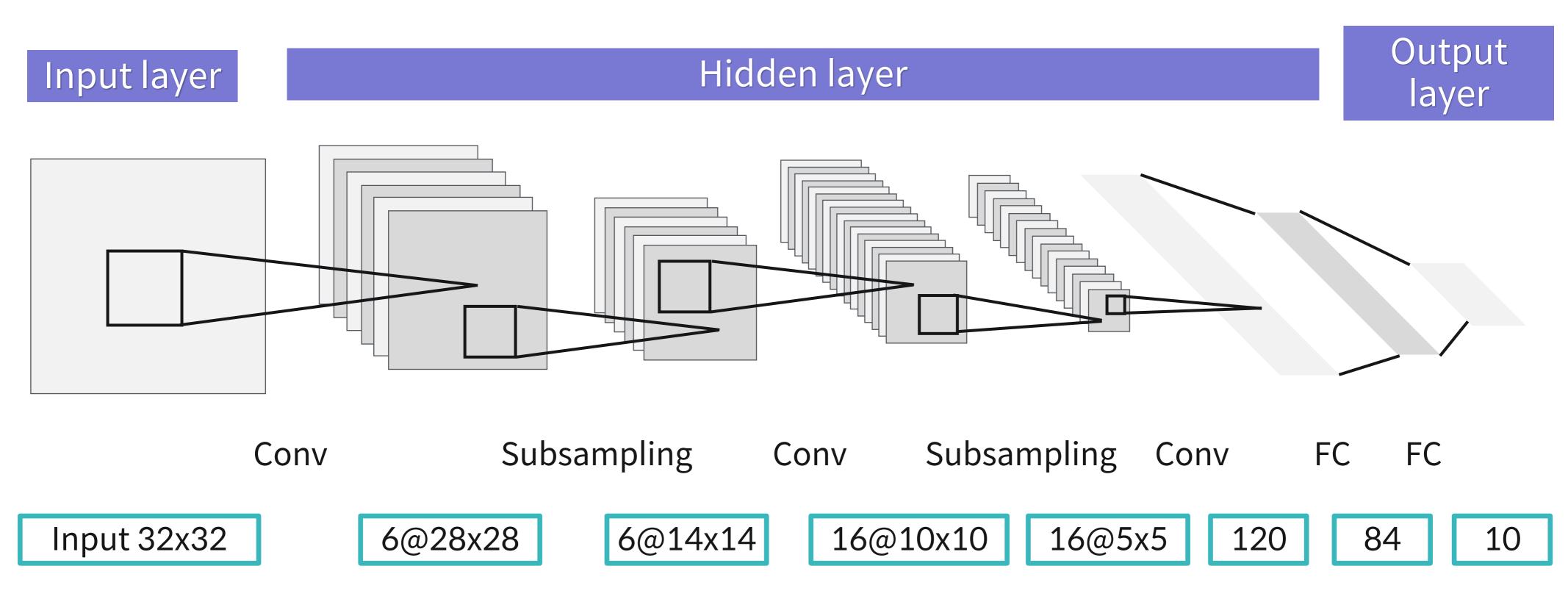
16@5x5

120

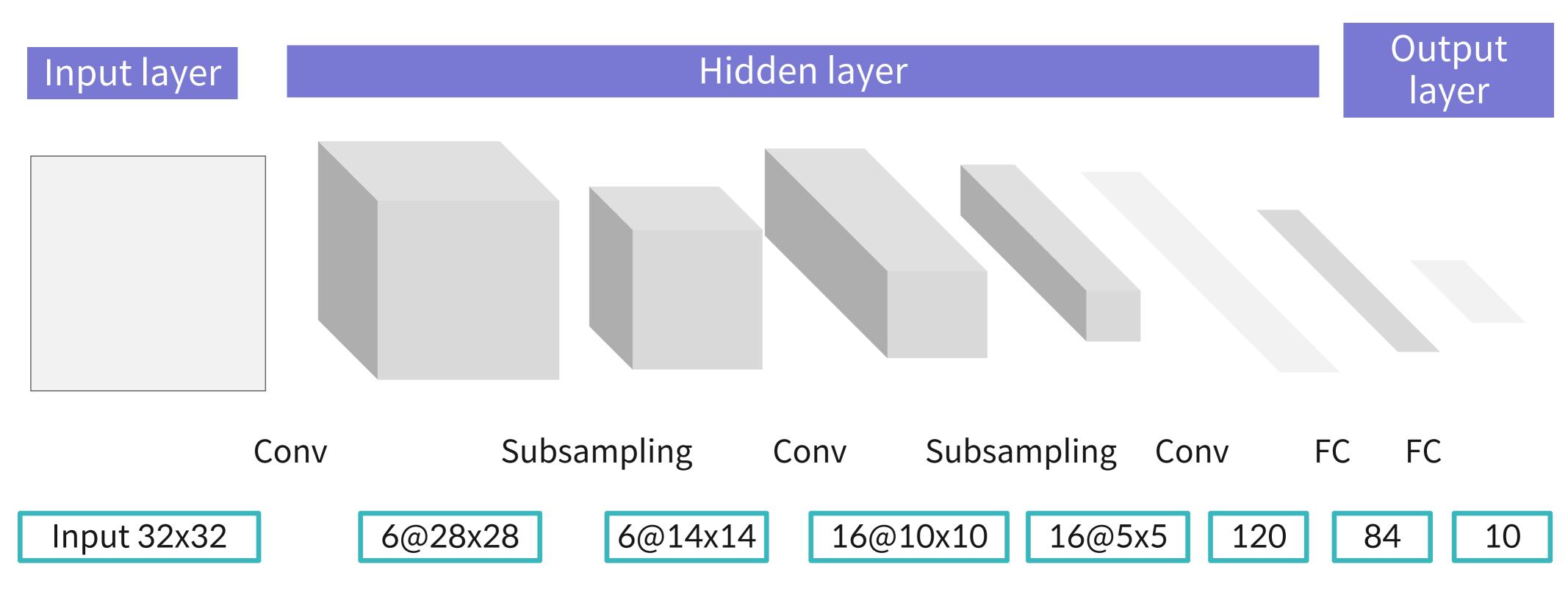
84

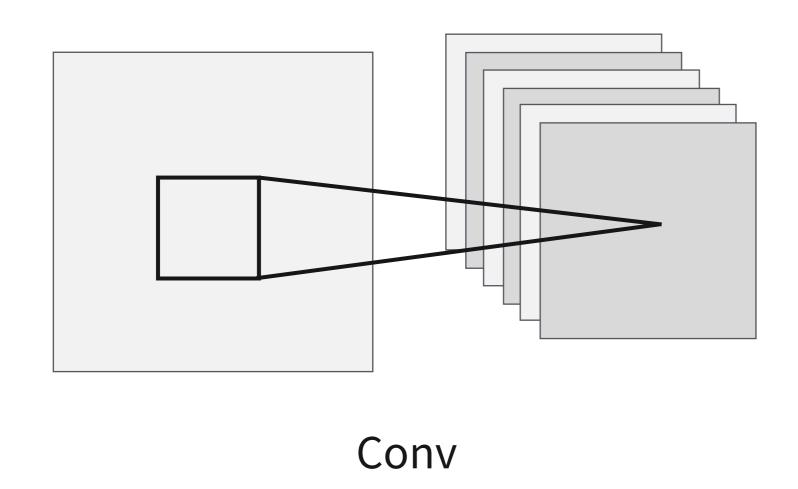
10









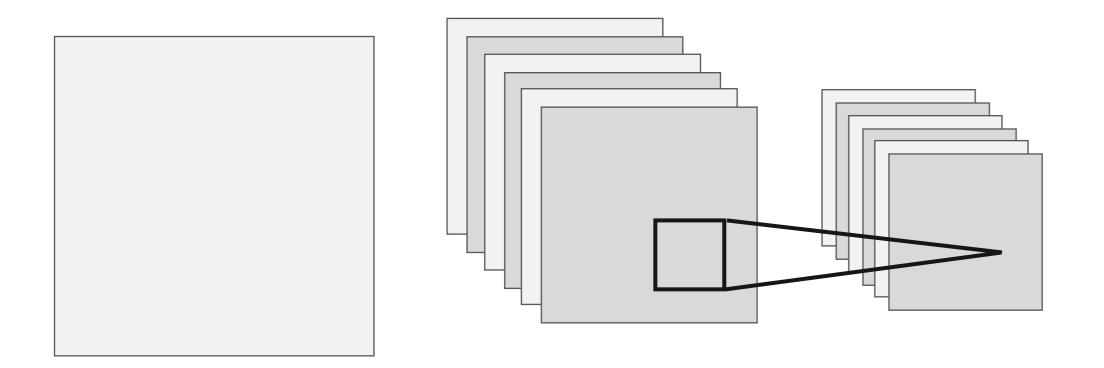


Input 32x32

6@28x28

1단계: 컨볼루션

입력값을 5X5 커널 6채널로 컨볼루션 입력값의 6가지 특징을 파악할 수 있음 5x5 커널은 3x3 커널보다 많은(넓은) 특징을 파악할 수 있음



Conv

Subsampling

Input 32x32

6@28x28

6@14x14

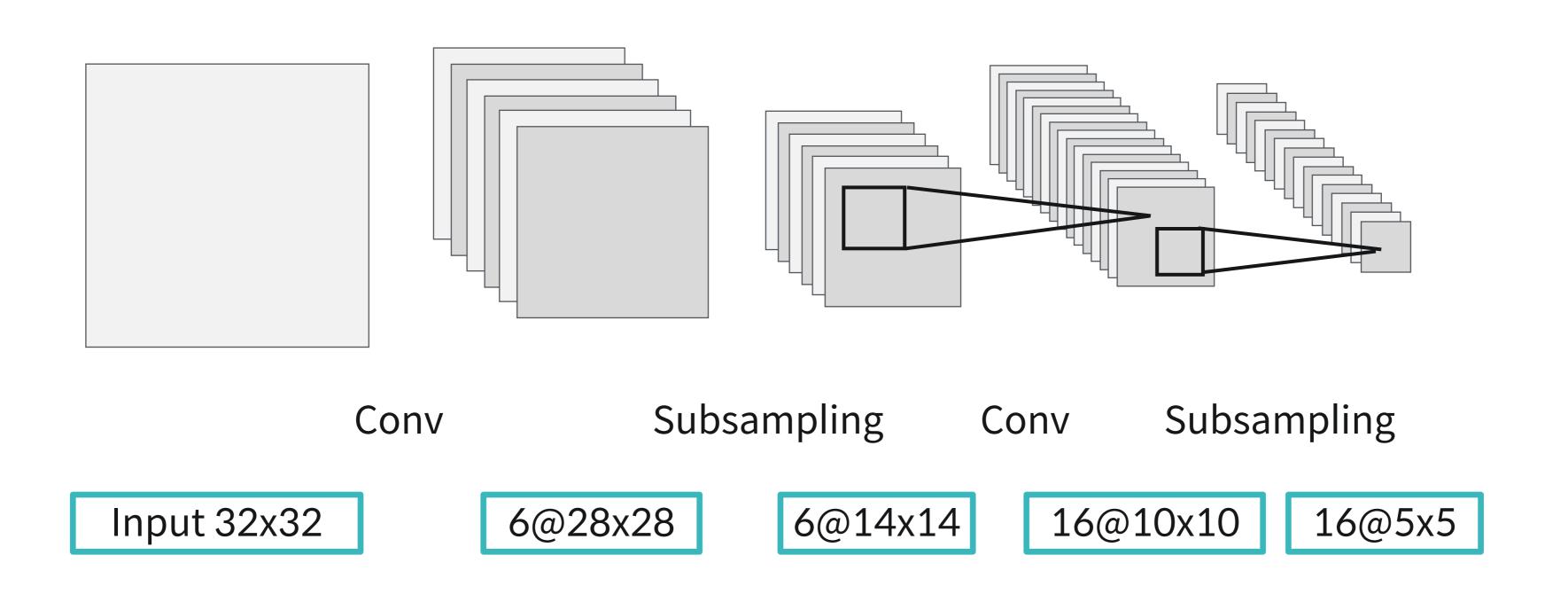
2단계: Subsampling (Average Pooling)

1단계 컨볼루션의 결과에

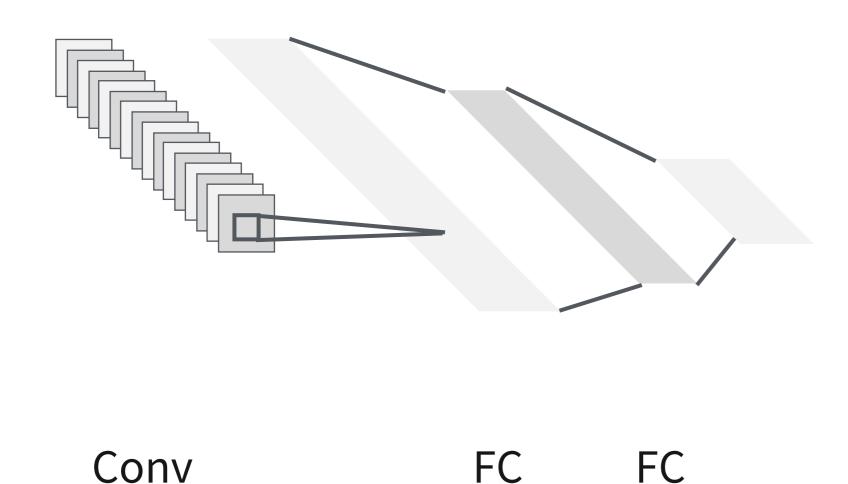
2x2로 Average POOLING

2x2 크기 중에 대푯값을 정하기 때문에

특징 정보를 ¼ 크기로 압축



특징 정보를 파악하기 위해 반복



3단계: 1x1 컨볼루션 레이어와 FC레이어

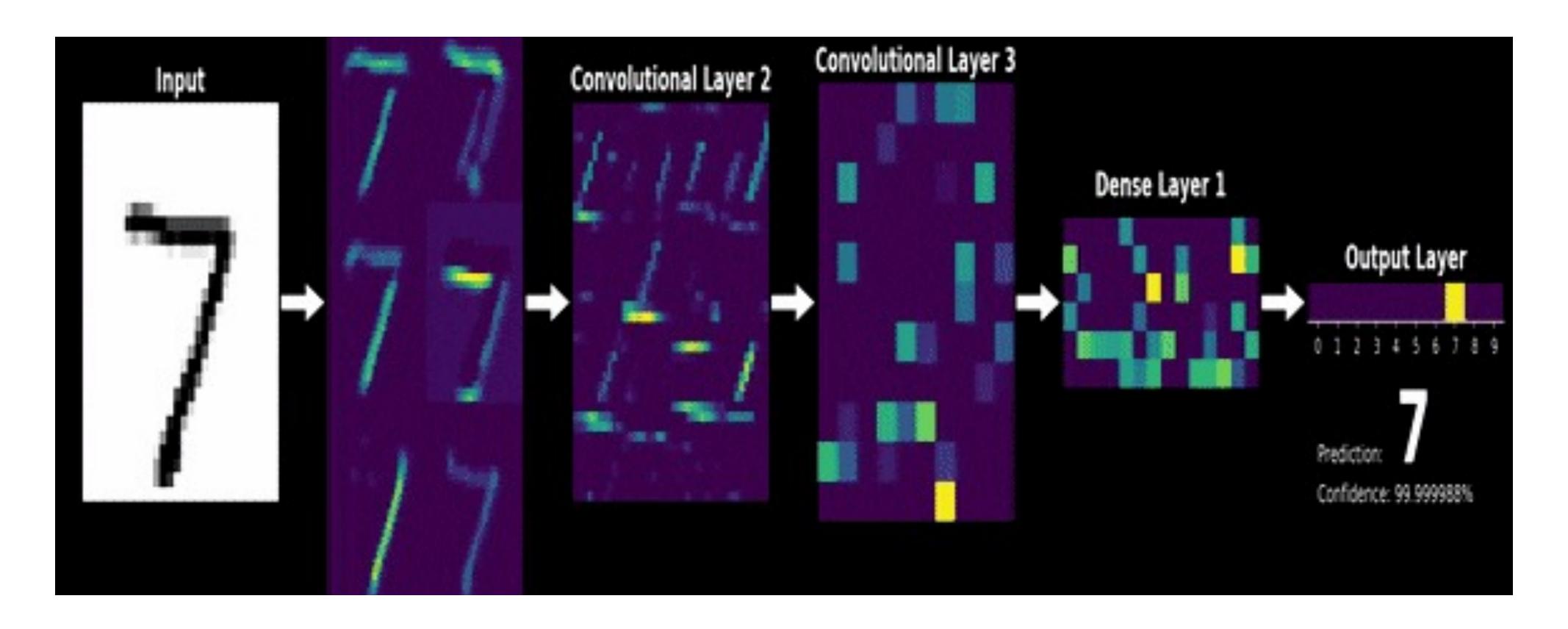
16@5x5 결과를 120 채널 1x1 컨볼루션

-> Flattening 하기 위해

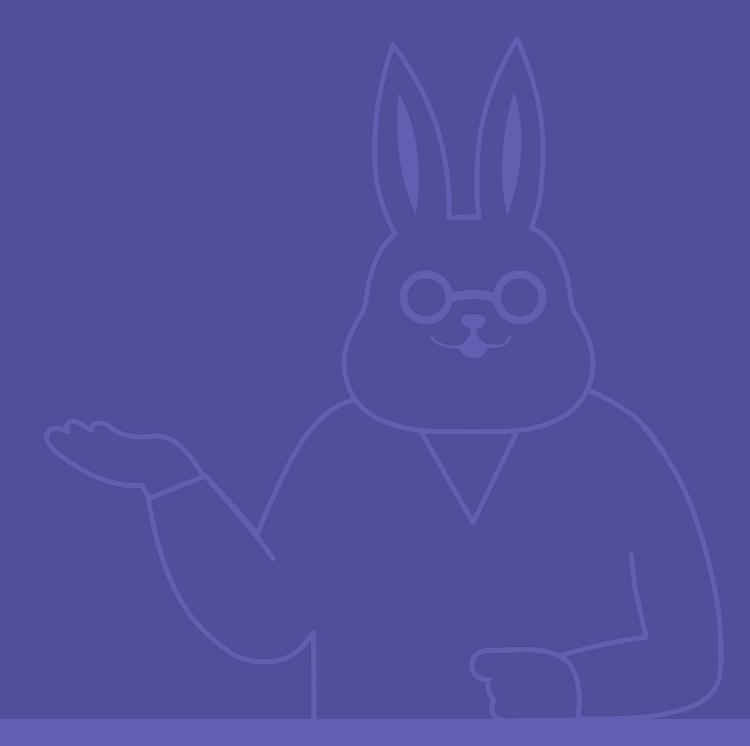
120개의 FC레이어, 84개 FC레이어

10개의 결과 레이어를 거쳐 10가지 분류

L	ayer	Feature Map	Size	Kernal Size	Stride	Activation	
Input	Image	1	32X32	_	_	_	
1	Convolution	6	28X28	5X5	1	tanh	
2	Average Pooling	6	14X14	2X2	2	tanh	
3	Convolution	16	10X10	5X5	1	tanh	
4	Average Pooling	16	5X5	2X2	2	tanh	
5	Convolution	120	1X1	5X5	1	tanh	
6	FC	_	84	_	_	tanh	
Output	FC	-	10	_	_	softmax	



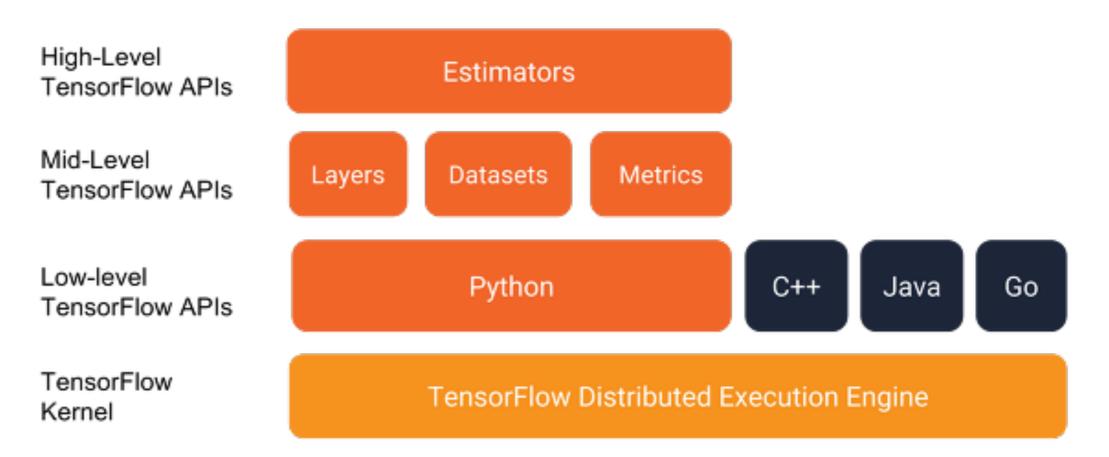
텐서플로우와 케라스 소개



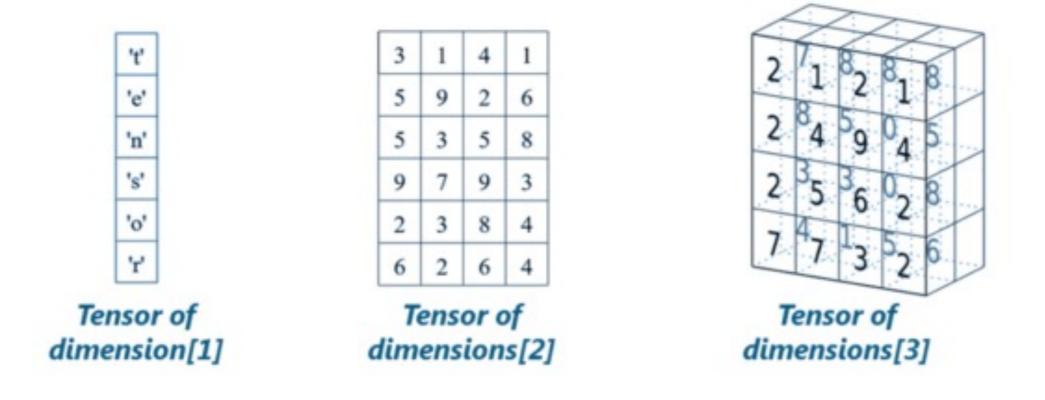
Confidential all rights reserved

❷ 텐서플로우

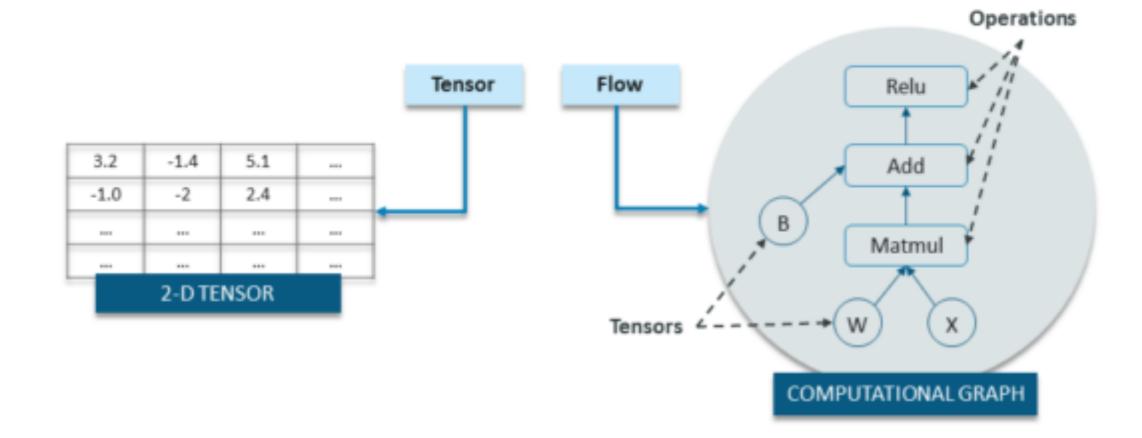




❷ 텐서플로우

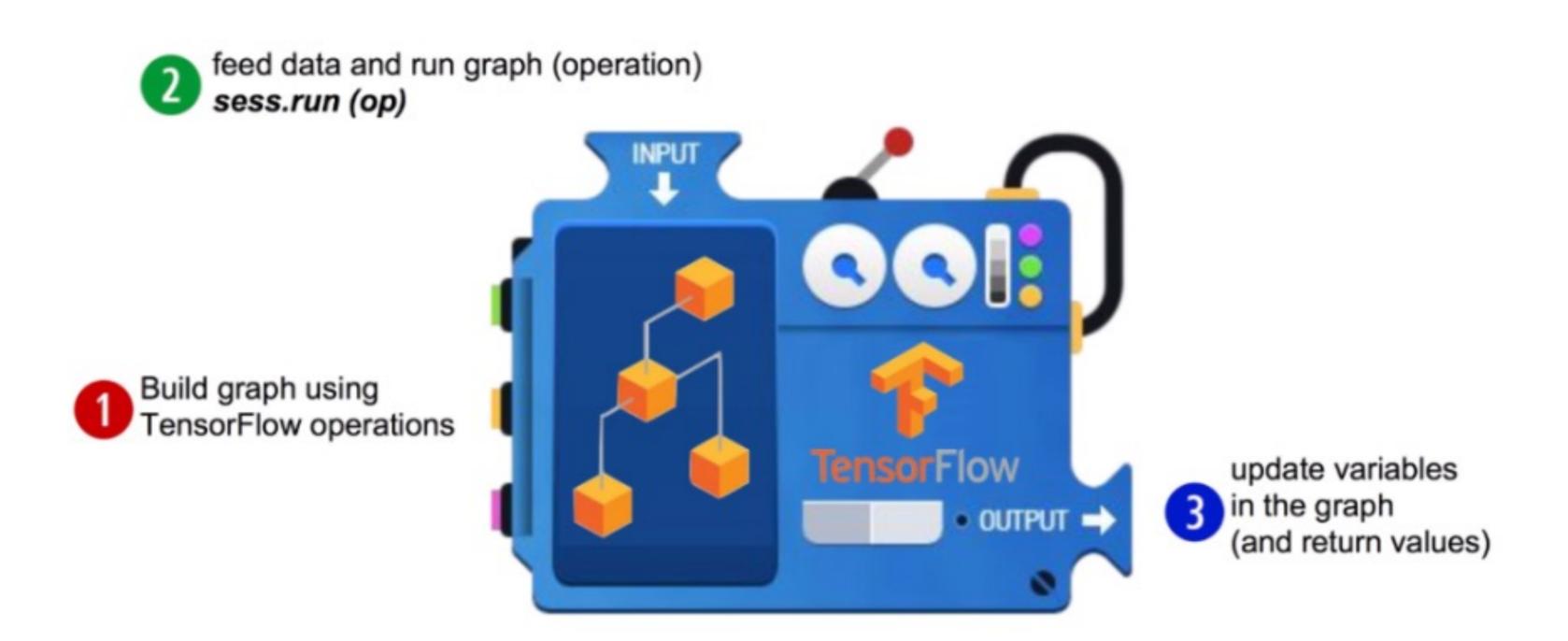


텐서의 표현



텐서와 텐서플로우

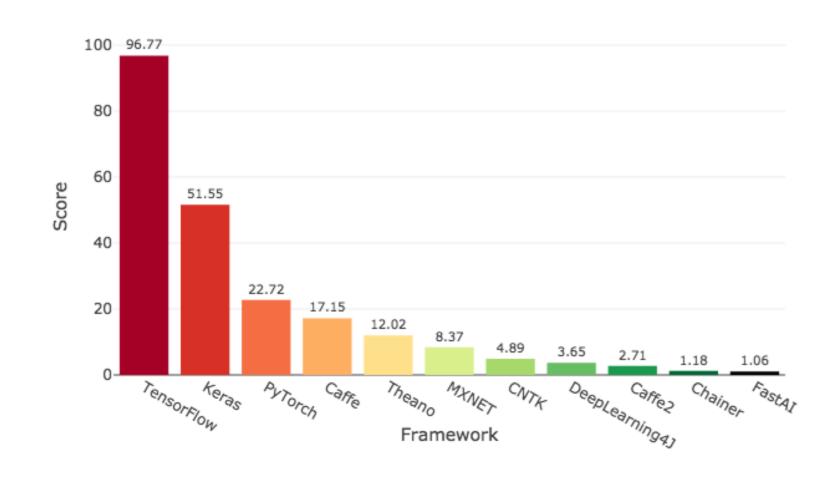
❷ 텐서플로우



☑ 케라스



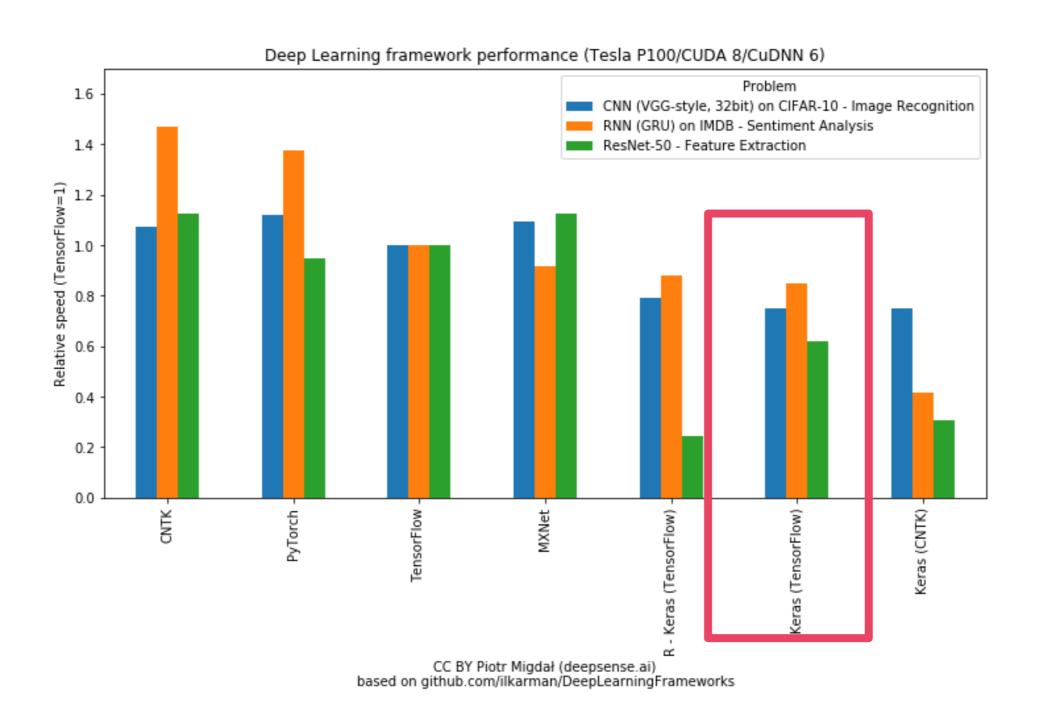
Deep Learning Framework Power Scores 2018



장점

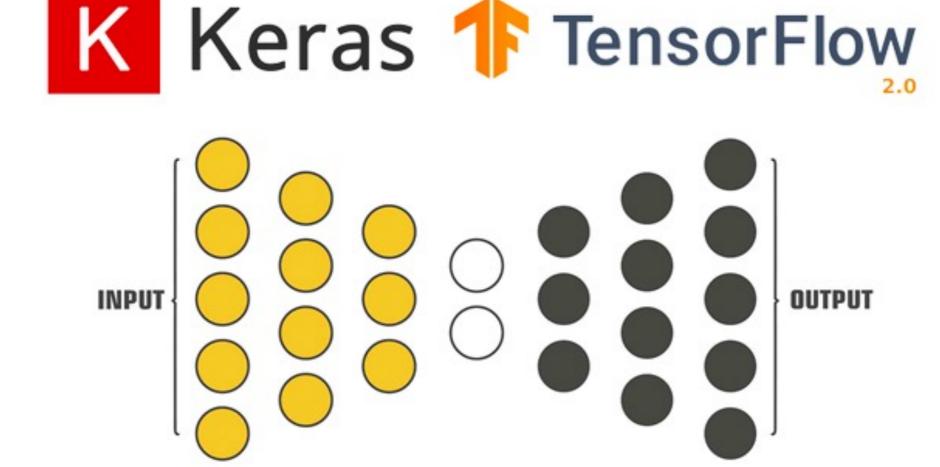
- 사용자 친화적
- 빠른 구현이 가능
- 다양한 프레임워크들의 벡엔드 호환
- 글로벌 기업들의 지원을

❷ 케라스 – 유일한 단점, 속도



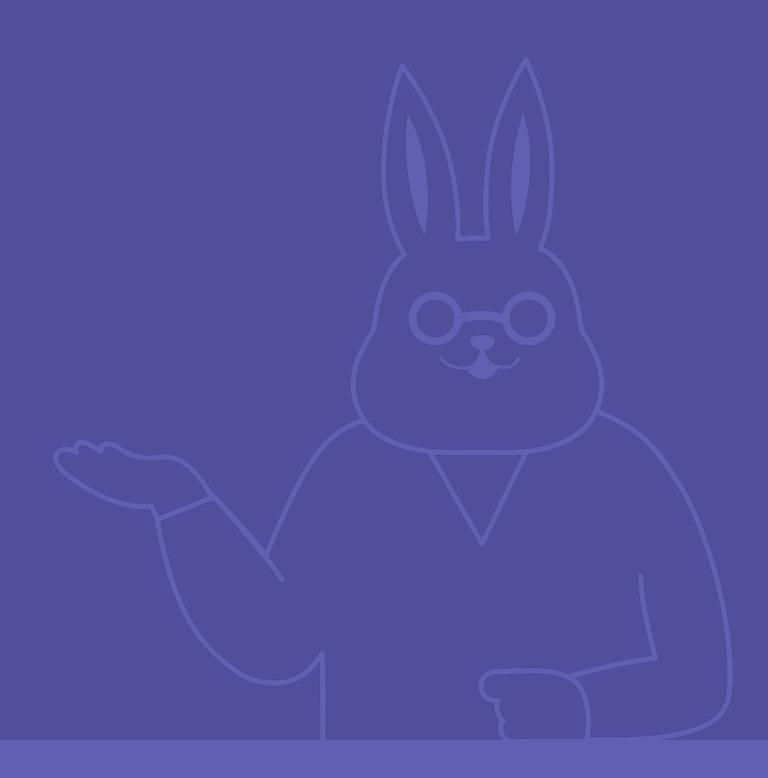
프레임워크 별 CNN, RNN, ResNet-50 추론 속도 성능 비교

☑ 텐서플로우 + 케라스



- Tensorflow 2.0부터 케라스가
 Tensorflow에 통합
- 모델의 틀은 케라스로 정의하고 실제 세션의 오퍼레이션은 텐서플로우로 하는 형태의 개발

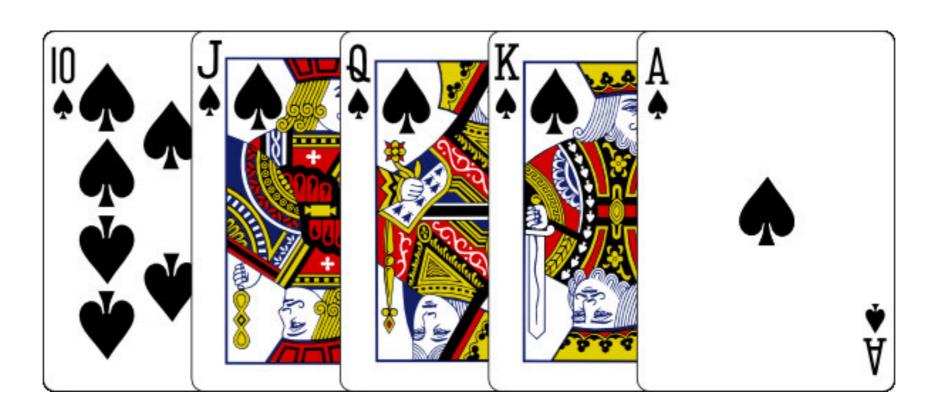
셔플링



Confidential all rights reserved

08 셔플링

❷ 셔플링이란



데이터 학습에서 카드를 섞는 것처럼 데이터를 섞는 것을 의미함 Mini Batch에서 활용

❷ 셔플링이란

Iteration

1	1	2	3	4	5	6	7	8	9	• •	• •	N-1	N
2	1	2	3	4	5	6	7	8	9	• •	• •	N-1	N
3	1	2	3	4	5	6	7	8	9	• •	• •	N-1	N
• •	1	2	3	4	5	6	7	8	9	• •	• •	N-1	N

❷ 셔플링이란

Iteration

1	1	2	3	4	5	6	7	35	9	• •	• •	N-1	N
2	567	344	983	84	75	29	442	546	34	• •	• •	77	90
3	485	234	200	299	857	82	345	77	32	• •	• •	35	21
• •	349	122	879	696	44	37	21	2	90	• •	• •	987	1

♥ 셔플링효과

셔플을 안 할 경우,

최적의 가중치로 수렴하기까지 오래 걸림 과적합이 될 가능성이 있음

셔플을 할 경우,

그래디언트가 다양한 방향으로 탐색하게 됨 모델이 좀더 좋은 최적해를 찾거나 더 빨리 최적해로 수렴될 수 있음

크레딧

/* elice */

코스 매니저 이재성

콘텐츠 제작자 최지수

강사 최지수

감수자 최지수

디자이너 강혜정

연락처

TEL

070-4633-2015

WEB

https://elice.io

E-MAIL

contact@elice.io

