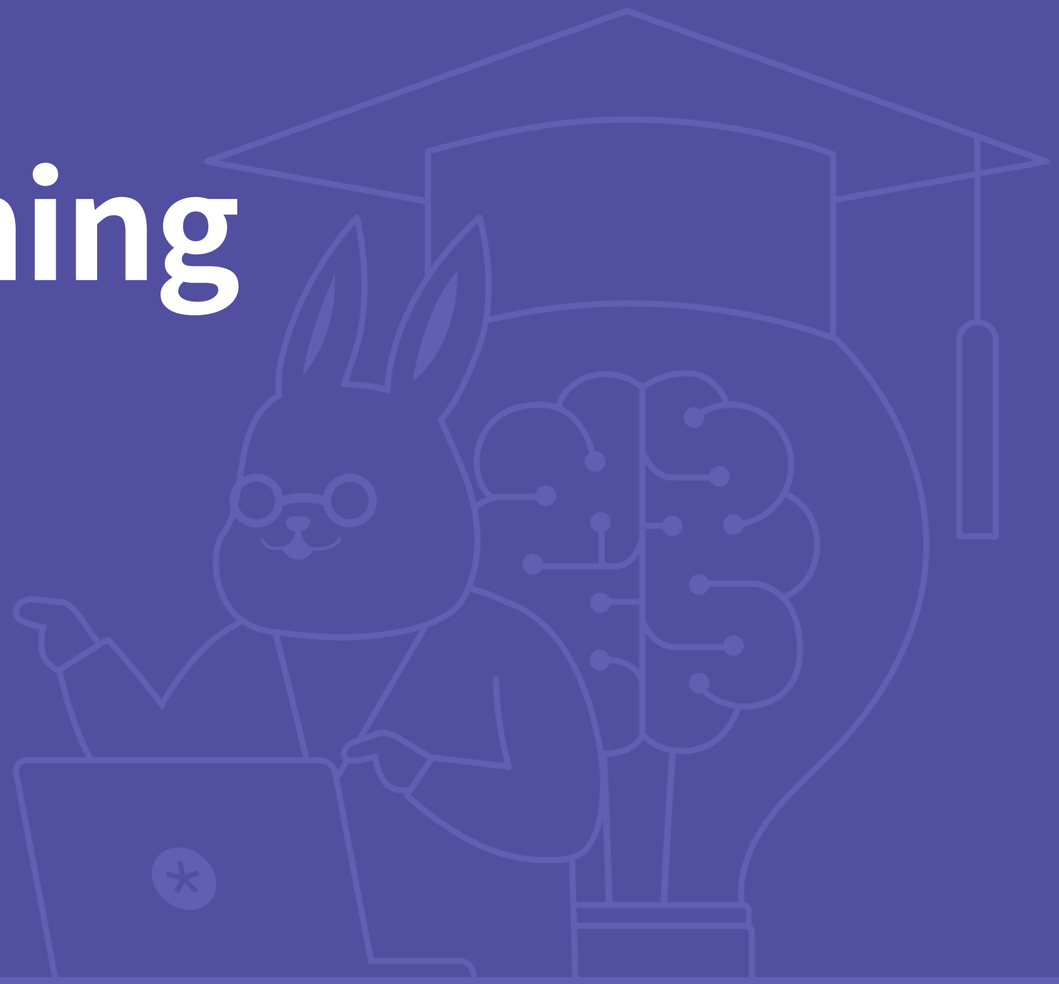
IMAGE Deep Learning

12장. MNIST 부터 CNN 까지

임성국



/* elice */

Contents

Curriculum

Curriculum 01

이미지에 대한 기초 지식을 학습하고, 이미지 처리에 필요한 양자화와 샘플링을 이해

Curriculum 02

간단한 구조인 MNIST 로 이미지 처리 실습

Curriculum 03

Convolution 에 대한 이해를 통해 CNN 을 학습

Curriculum

Curriculum 04

CIFAR-10 을 이용한 CNN 실습 진행

Curriculum 05

VGGNet 에 대한 이해와 실습

Curriculum 06

IMAGE 처리 동향을 이해한다.

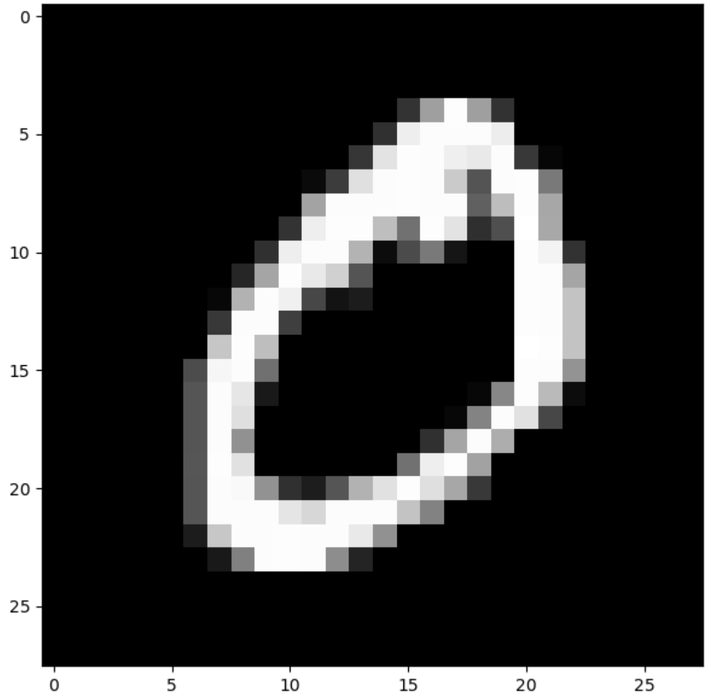
01

이미지 개요

01 이미지 개요

WHY 0?

왼쪽 그림이 0 인 이유는 뭘까?



01 이미지 개요

사물을 본다는 것의 의미

- 영국, Gregory

인지심리학자인 그레고리는 영국에서 선천적 시각장애를 가진 52 환자의 안구기증으로 인한 개인 수술 결과에 대한 보고서를 썼다. 환자는 수술전 자건거도 타고, 개인 작업실에서 물건도 만드는 활발한 활동을 하였지만 수술 후 우울증에 시달리다 3년만에 사망하였다.

01 이미지 개요

인간과 컴퓨터

```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 52 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 89 41 92 36 54 22 40 40 28 66 33 13 80
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 35 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 35 71 89 07 05 44 44 37 44 60 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 57 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

/* elice */

디지털과 아날로그

사진에 아날로그 필름을 사용했었다.
지금은 디지털 방식으로 기록한다.
아날로그와 디지털의 차이는 무엇인가?

디지털의 특징

1. 양자화

모든 기록은 정해진 단계를 가진다. 8비트로 저장되는 영상은 2^8 개의 단계만을 표현한다.

2. 샘플링

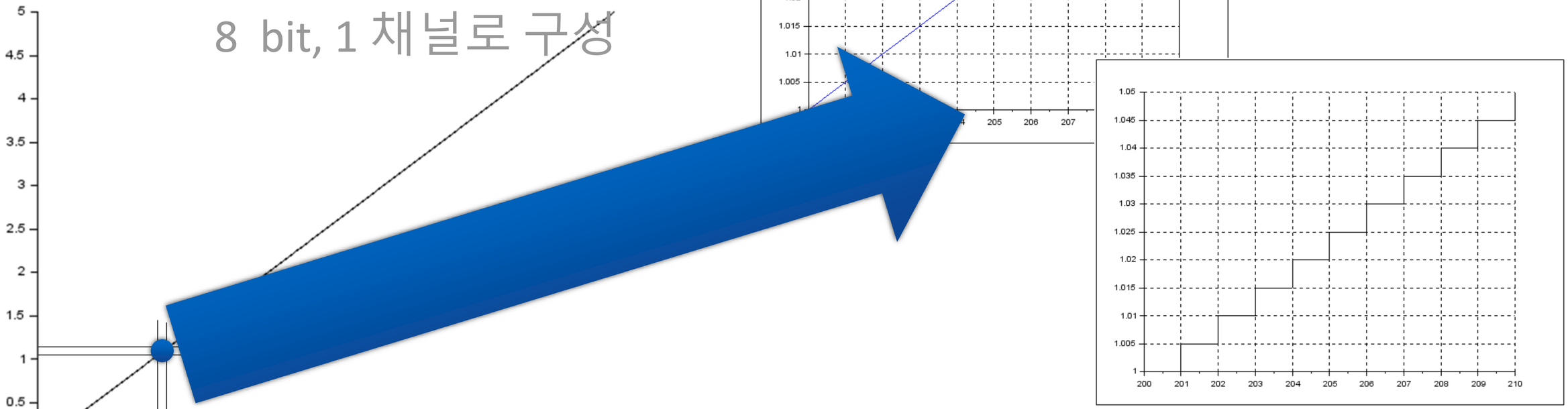
영상으로 저장되는 데이터는 기계가 가지고 있는 저장영역 이상을 표현할 수 없다.

01 이미지 개요

양자화

MNIST : 흑백 256 단계 (0 .. 255)

8 bit, 1 채널로 구성



/* elice */

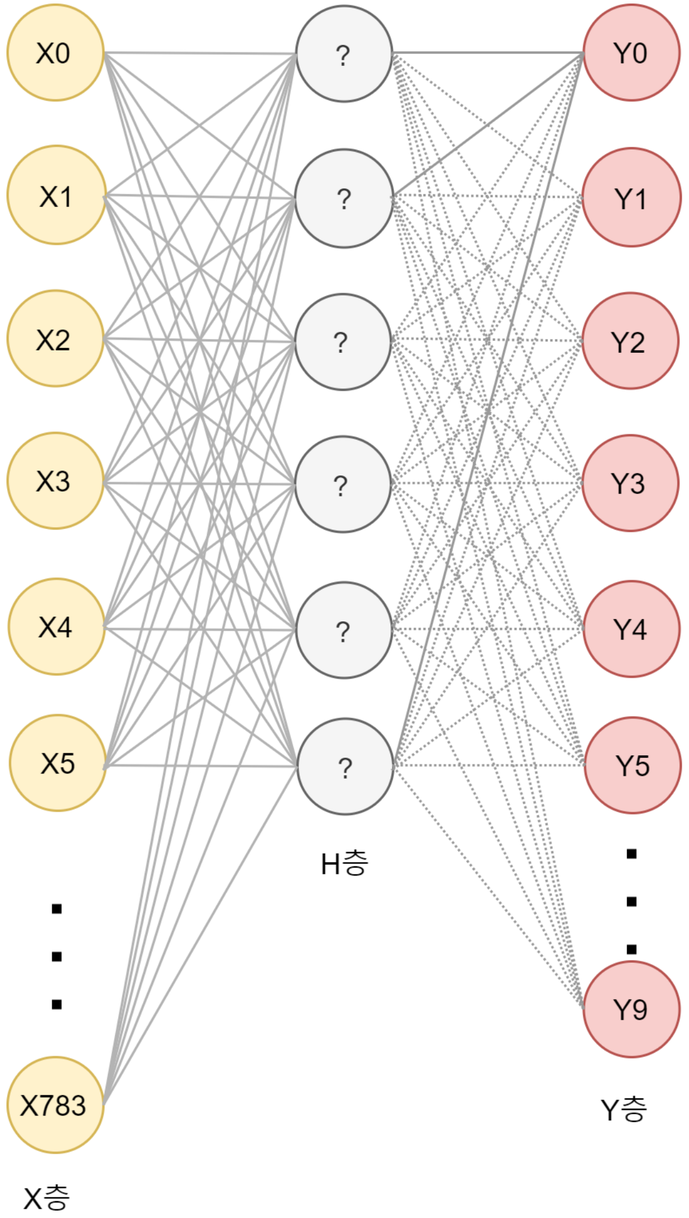
02

MNIST 실습

02 MNIST 실습

MNIST

28 * 28 크기의 이미지로 구성된
MNIST 를 784 크기의 배열로 놓고
신경망에 넣어서 결과를
확인해보자.



/* elice */

02 MNIST 실습

텐서가 아닌 배열

Example

```
import numpy as np
import time
from keras.datasets import mnist

(x_train, t_train), (x_test, t_test) = mnist.load_data()
t_trainlbl, t_testlbl = t_train, t_test

# 28X28 을 784 로 변환
x_train = x_train.reshape(   )     # 28*28 을 784로 변환
x_test = x_test.reshape(   )
```

02 MNIST 실습

수치미분

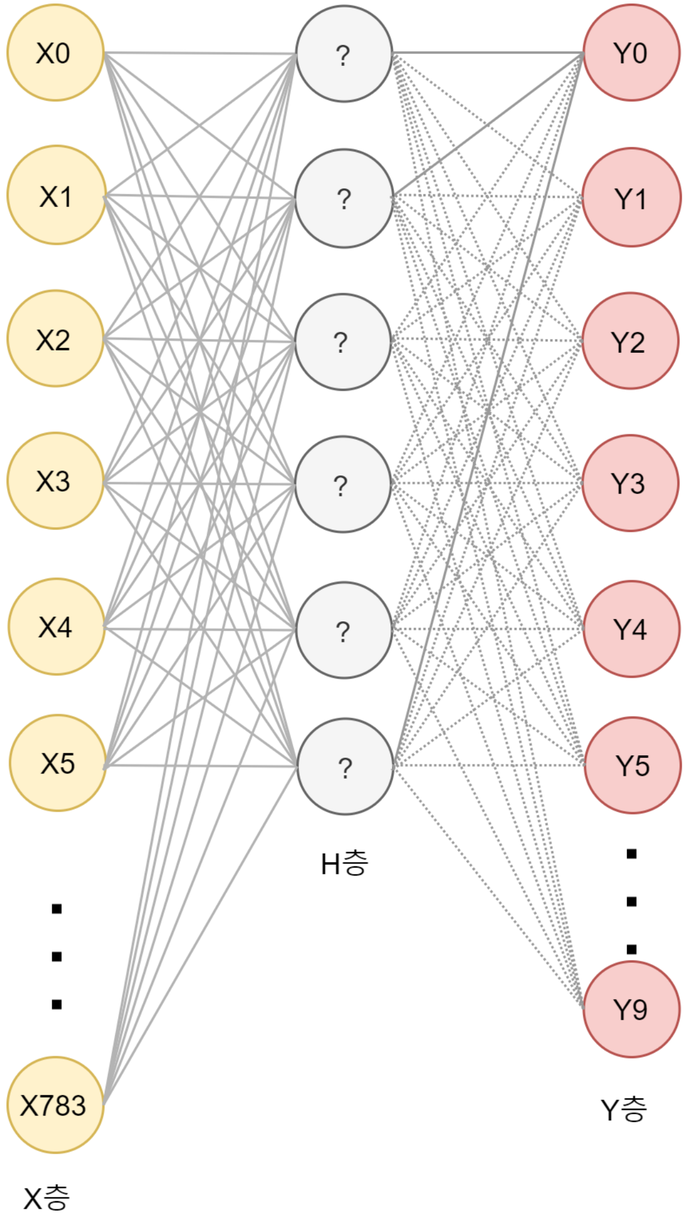
Example

```
def numerical_diff(f, x):
    .. .. ..
    while not it.finished:
        index = it.multi_index
        tmp = float(x[index])
        x[index] = tmp + h
        fxh2 = f()     # f(x+h)
        x[index] = tmp - h
        fxh1 = f()     # f(x-h)
        nd_coef[index] = (fxh2 - fxh1) / (2*h)
        x[index] = tmp
        it.iternext()
    return nd_coef
```

02 MNIST 실습

MNIST

미분을 사용하는 경우와 역전파를
사용하는 경우의 시간을
경험해보자.

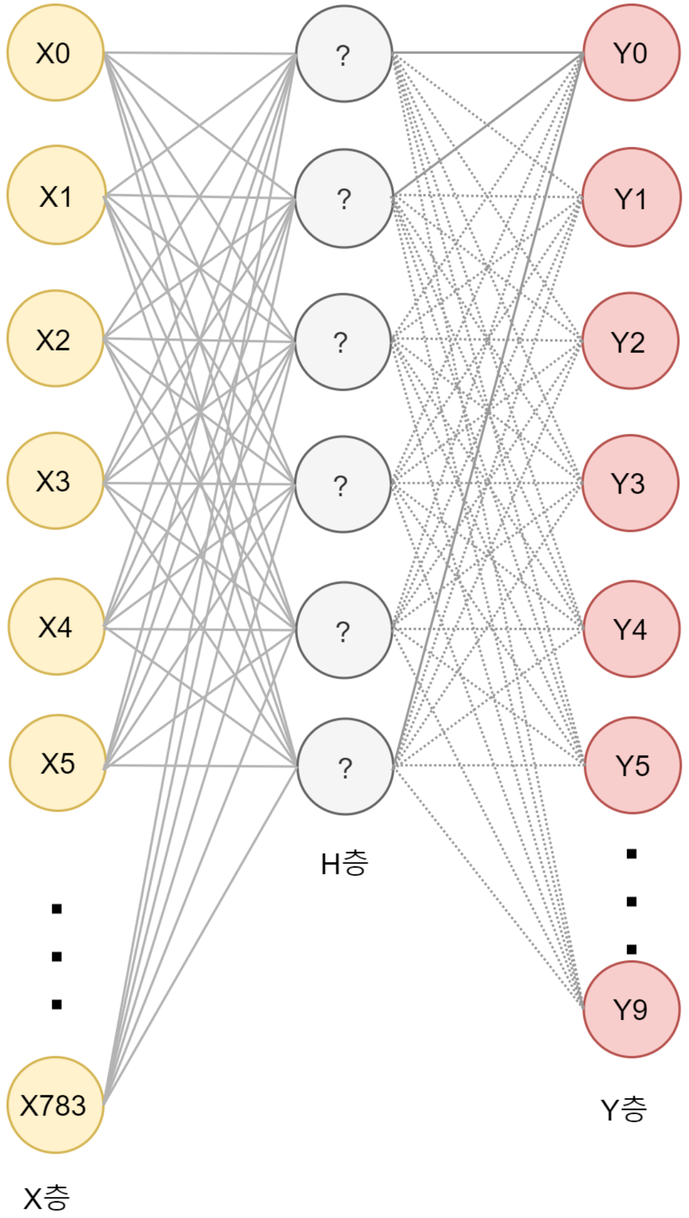


/* elice */

02 MNIST 실습

MNIST

Tesnorflow 를 사용한 MNIST 실습



/* elice */

02 MNIST 실습

모델생성

- 28*28 크기의 입력을 납작하게(784) 만든다.
- 입력과 히든레이어를 연결한다. 히든레이어는 128 개의 텐서를 가진다. 이때 활성함수는 ReLu 를 사용한다.
- 히든레이어와 출력을 연결한다. 출력은 10개다. 소프트맥스를 사용한다.

Example

```
model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dense(10, activation='softmax')
])
```

02 MNIST 실습

모델컴파일

- 최적화를 위해 Adam 을 사용한다.
- 크로스엔트로피로 손실함수를 계산한다.
- 정확도로 기준한다.

Example

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

02 MNIST 실습

학습진행

- 만들어진 모델에 학습용 데이터와 정답을 넣고, 10 에포크(전체 데이터를 10번 반복)진행

Example

```
model.fit(x_train, y_train, epochs=10)
```

/* elice */

03

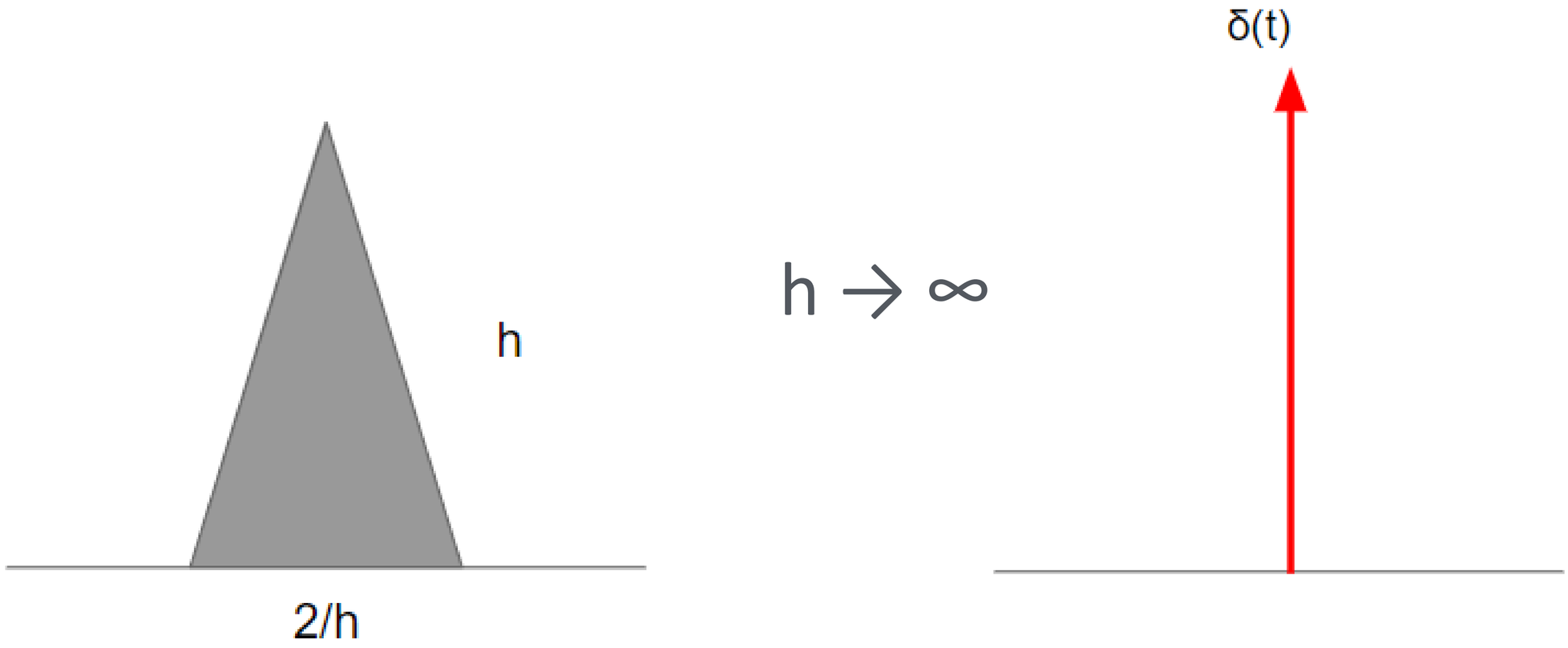
CNN

아날로그 신호처리

선형 시불변 시스템 (Linear Time Invariant System , LTI System)

선형, 시간의 영향을 받지 않는 시스템

Dirac Delta



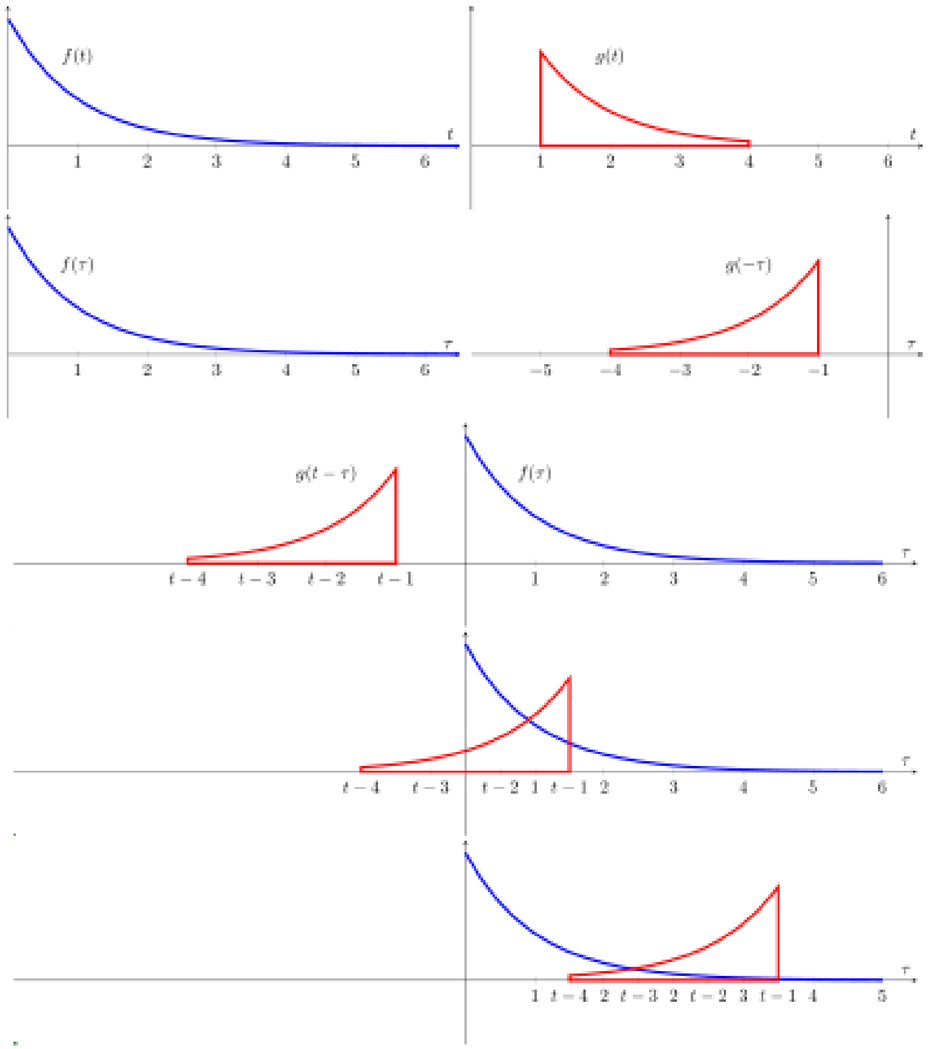
임펄스응답

Convolution, 합성곱

f(t) 와 g(t) 의 convolution

f(t) * g(t)

두 함수가 겹치는 부분의 적분값

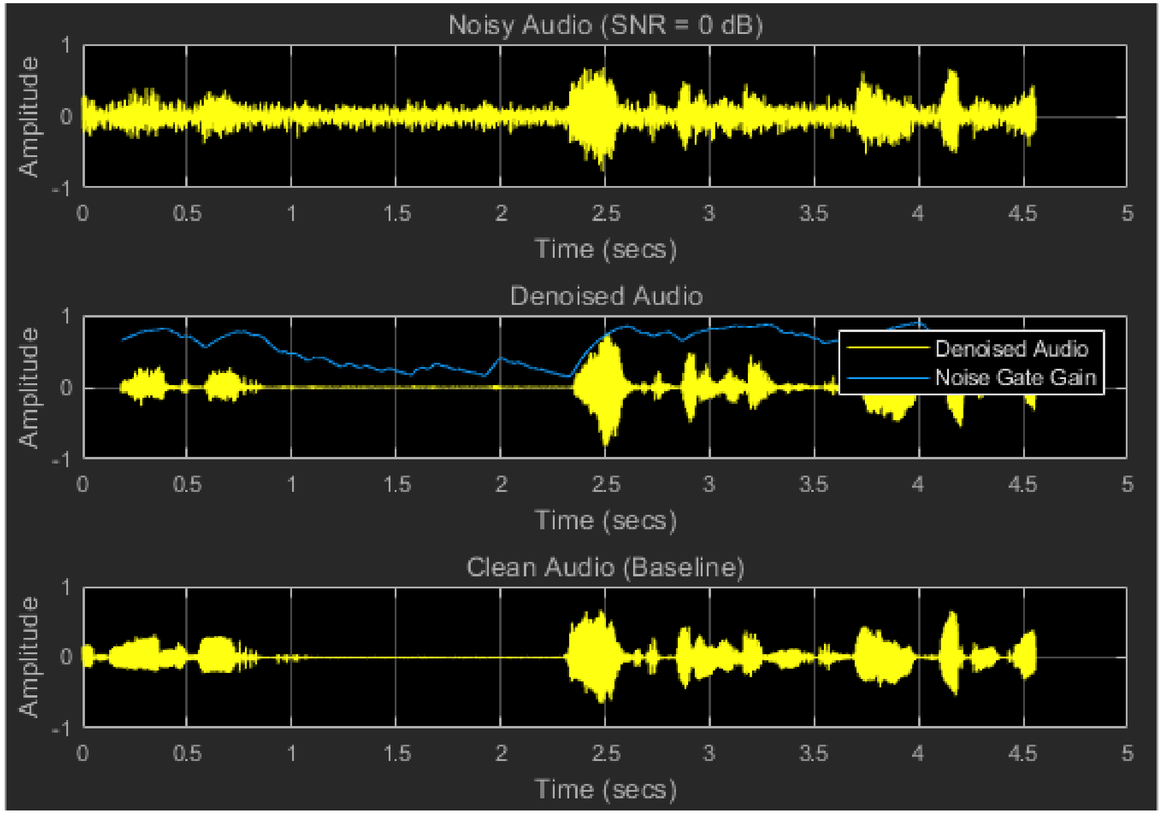


Convolution, Filter

f(t) : (화이트노이즈가 포함된) 음성신호

g(t) : 잡음 제거 필터

f(t) * g(t) : 노이즈가 제거된 음성신호



Convolution 의미

$$f(t) * g(t)$$

f(t) 안에 있는 g(t)

또는 g(t) 안에 있는 f(t)

Lena

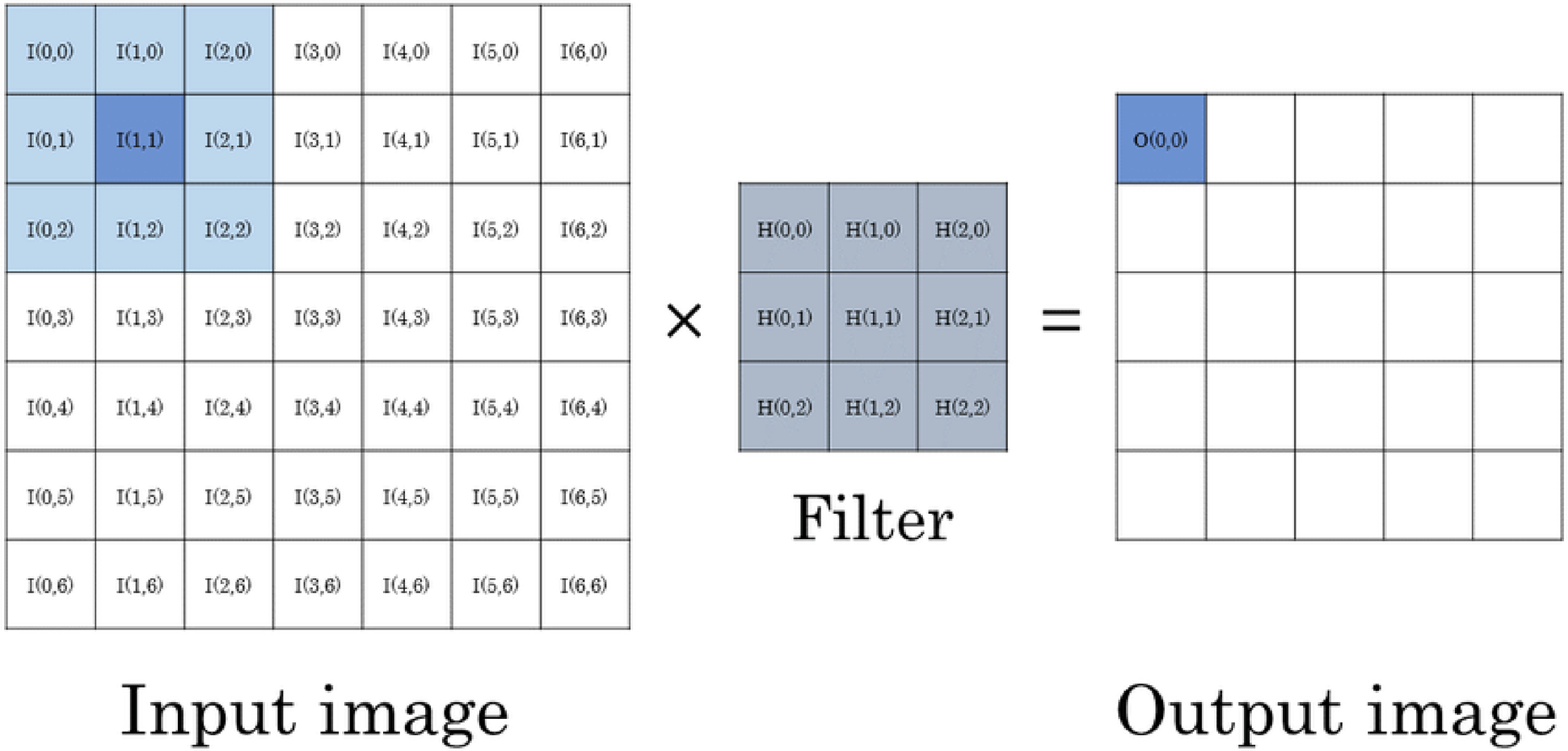
이미지처리에 많이 사용되는 사진

RED

GREEN

BLUE

이미지와 Convolution



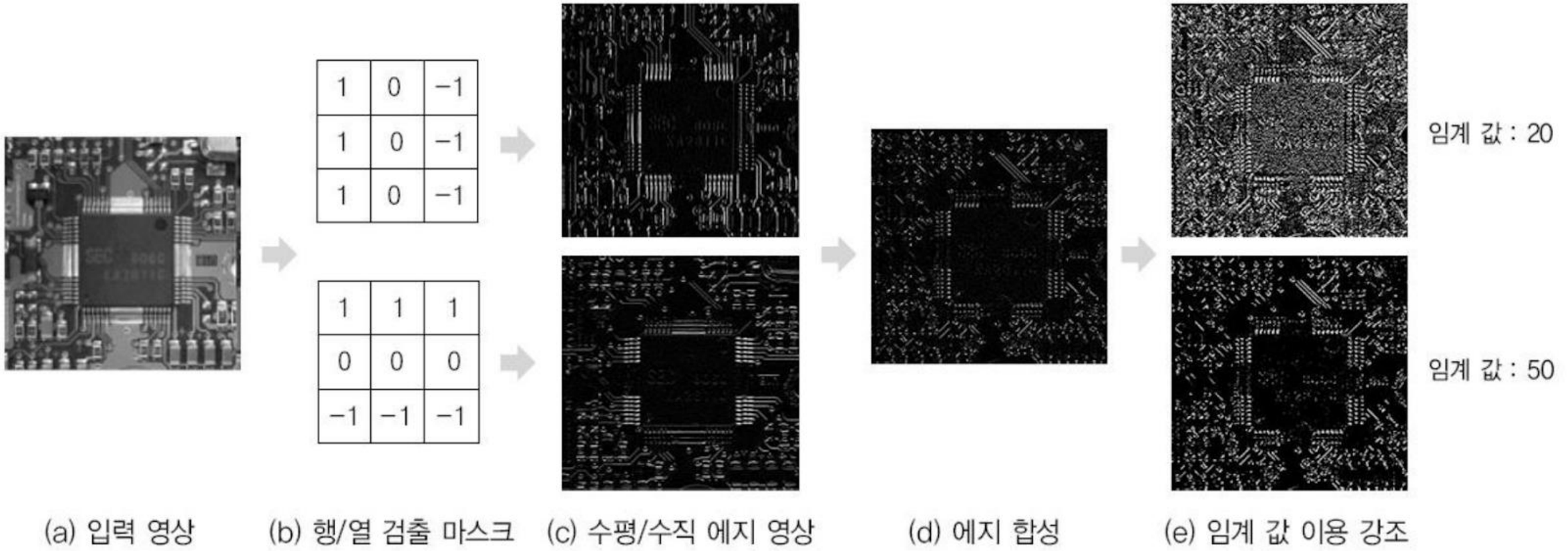
/* elice */

가우시안 필터

*

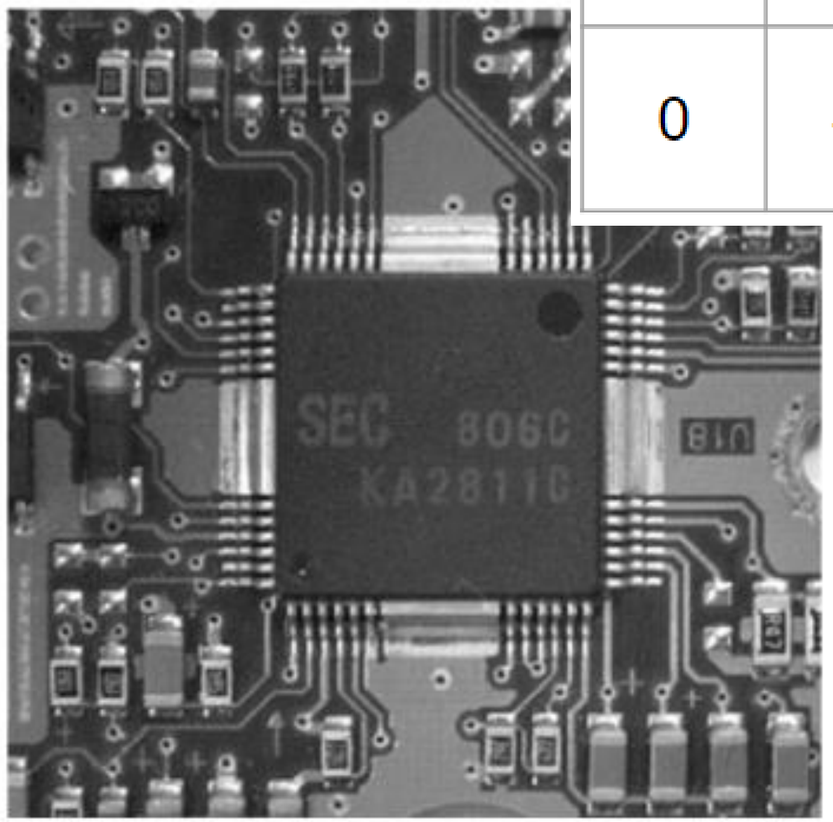
1	4	8	4	1
4	16	32	16	4
8	32	64	32	8
4	16	32	16	4
1	4	8	4	1

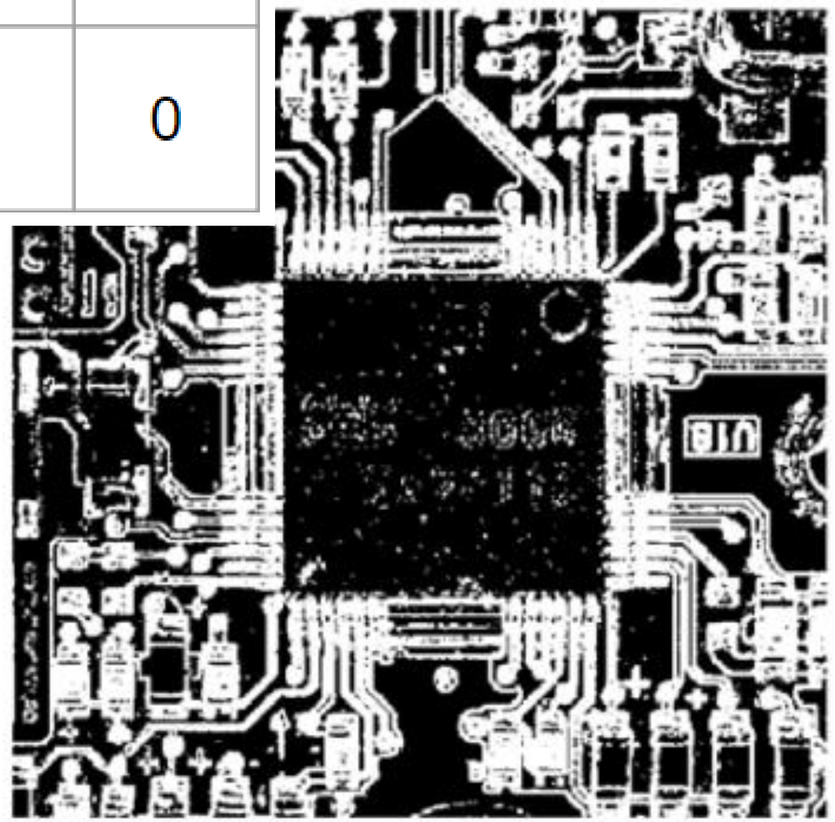
미분 필터


1 0 −1
1 0 −1
1 0 −1
1 1 1
0 0 0
−1 −1 −1
임계 값 : 20
임계 값 : 50
(a) 입력 영상
(b) 행/열 검출 마스크
(c) 수평/수직 에지 영상
(d) 에지 합성
(e) 임계 값 이용 강조

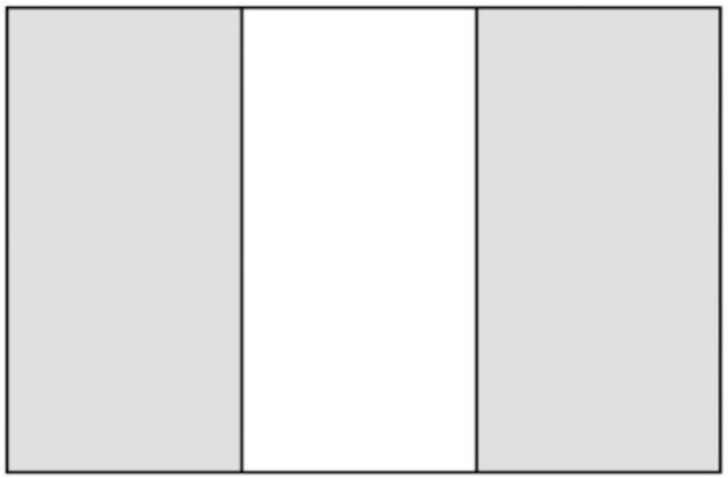
2차 미분 필터, 라플라스

0	-1	0
-1	4	-1
0	-1	0

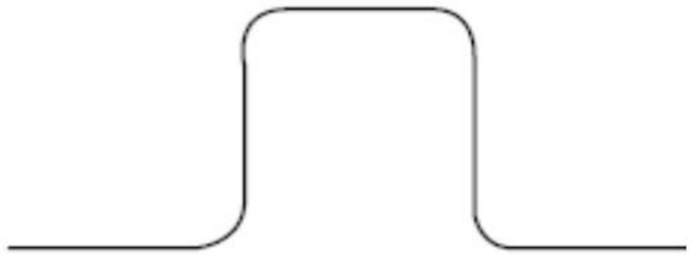




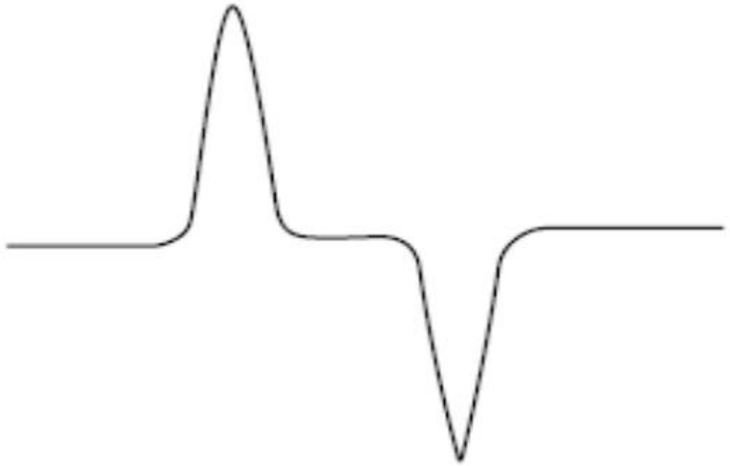
명암변화와 미분값



원래 영상

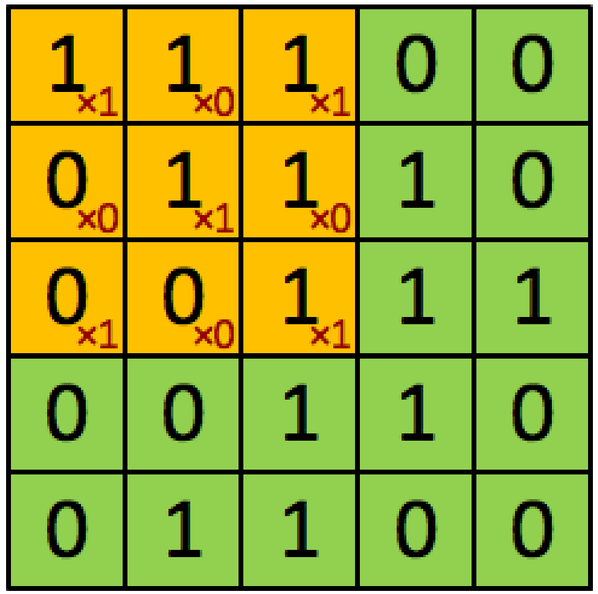


명암도 변화

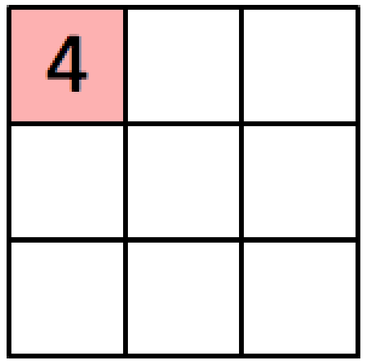


미분값 변화

합성곱의 정의



Image



Convolved Feature

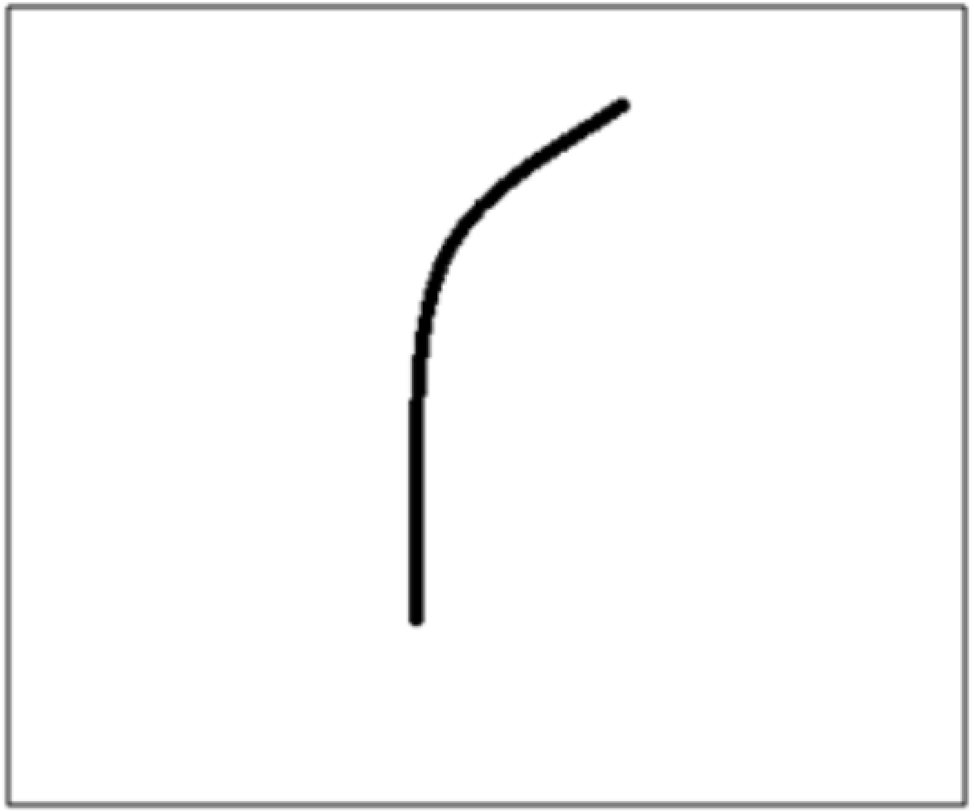
합성곱 연산은 두 함수 f, g 가운데 하나의 함수를 반전(reverse), 전이(shift)시킨 다음, 다른 하나의 함수와 곱한 결과를 적분하는 것을 의미

- 위키

Filter

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

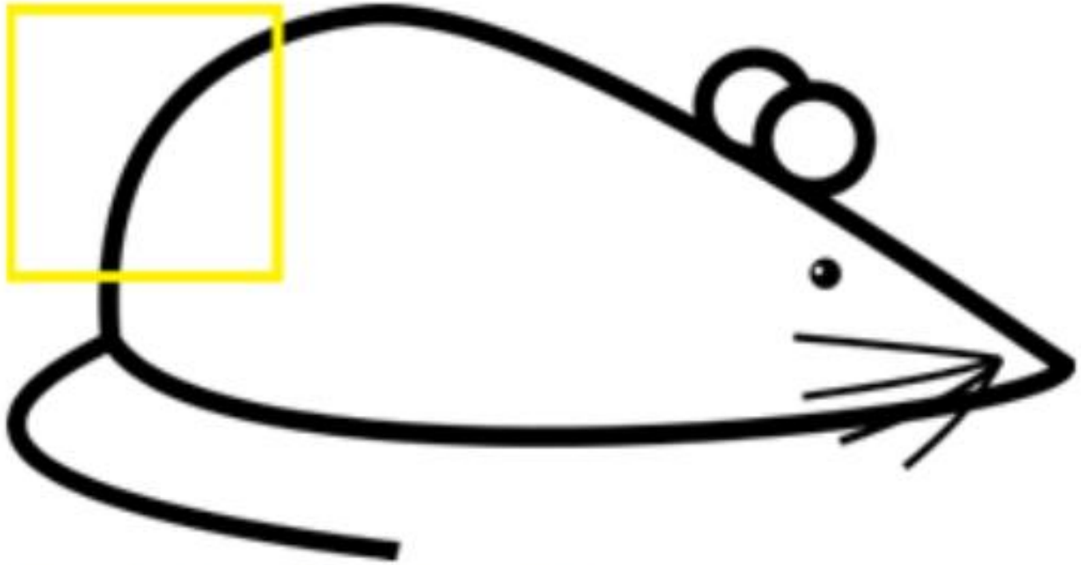
Pixel representation of filter



Visualization of a curve detector filter

Original Image

Original image



Visualization of the filter on the image

Original Image 1 * filter(Convolution)

Visualization of the filter on the image

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Original Image 1 * filter(Convolution)

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

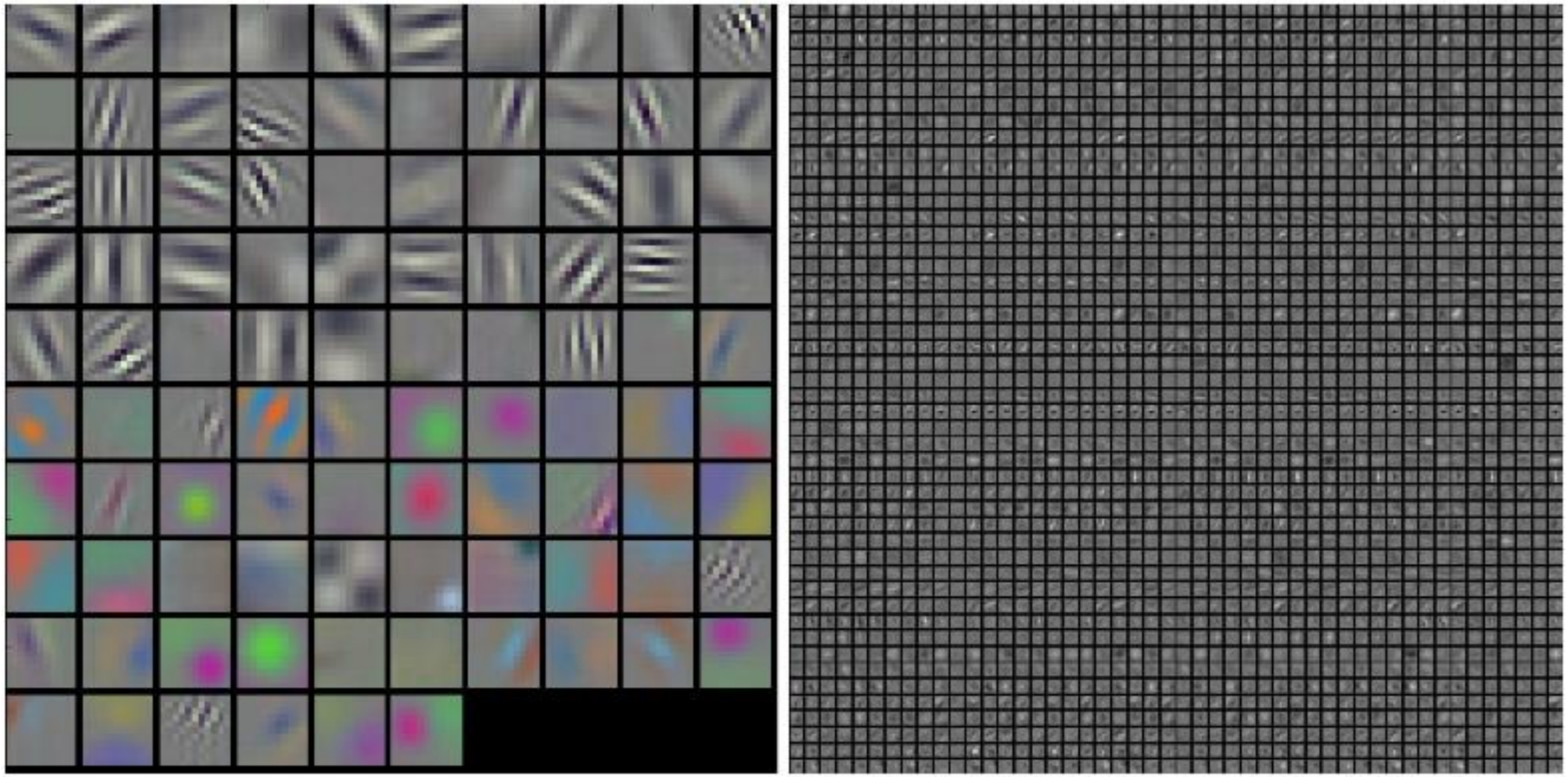
*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

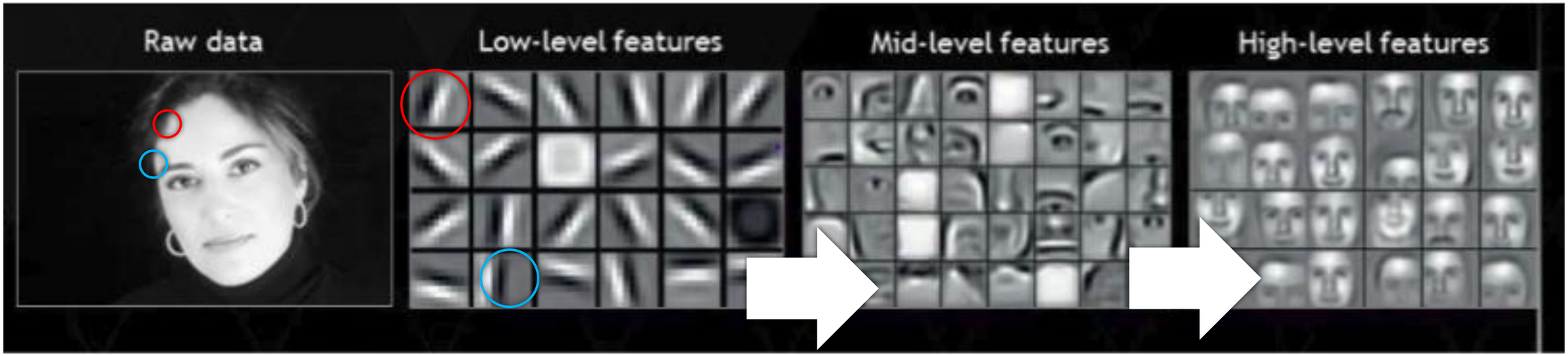
Multiplication and Summation = 0

AlexNet 에서 나온 1, 2차 Filter



Convolution 의미 : 'F 안에 G 있다'

Edge 성분검출 -> 위치값 확인



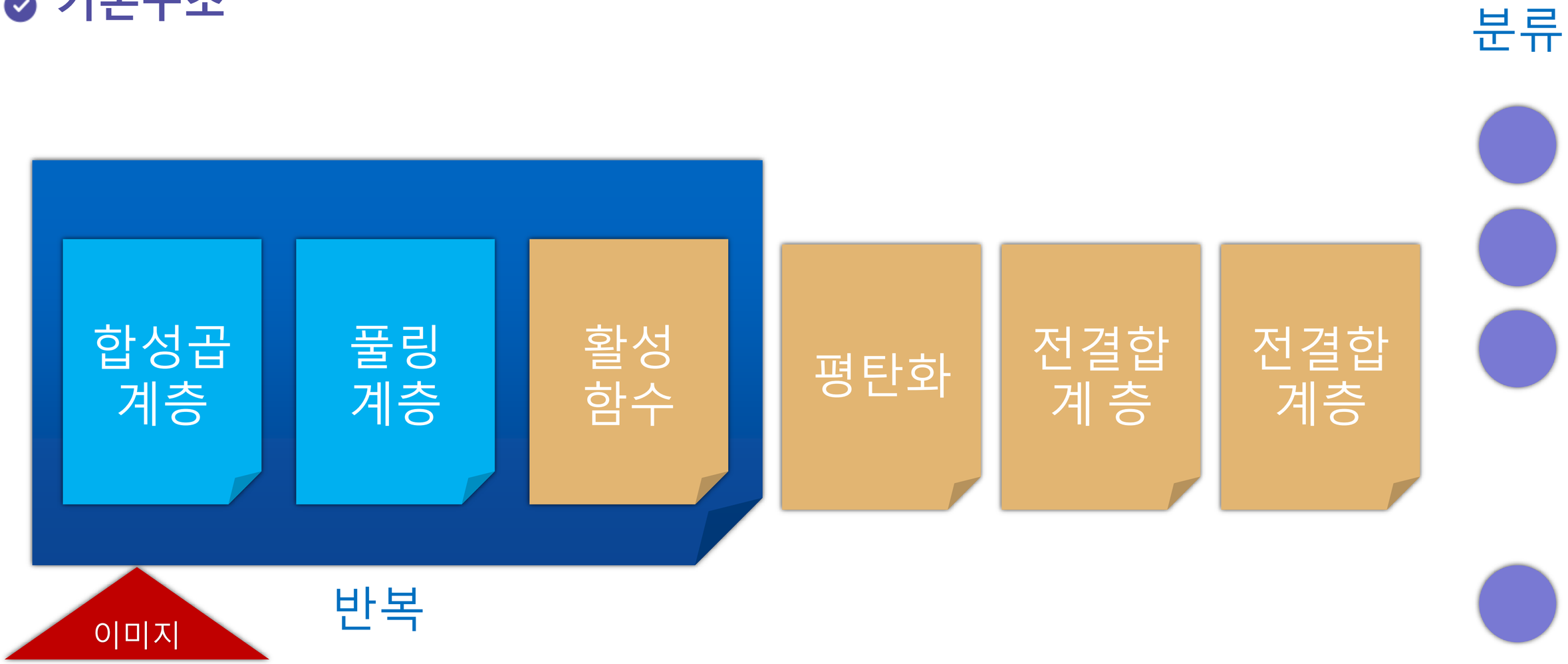
층이 올라갈수록 특징들은 결합되고,

이미지 크기는 줄어듬

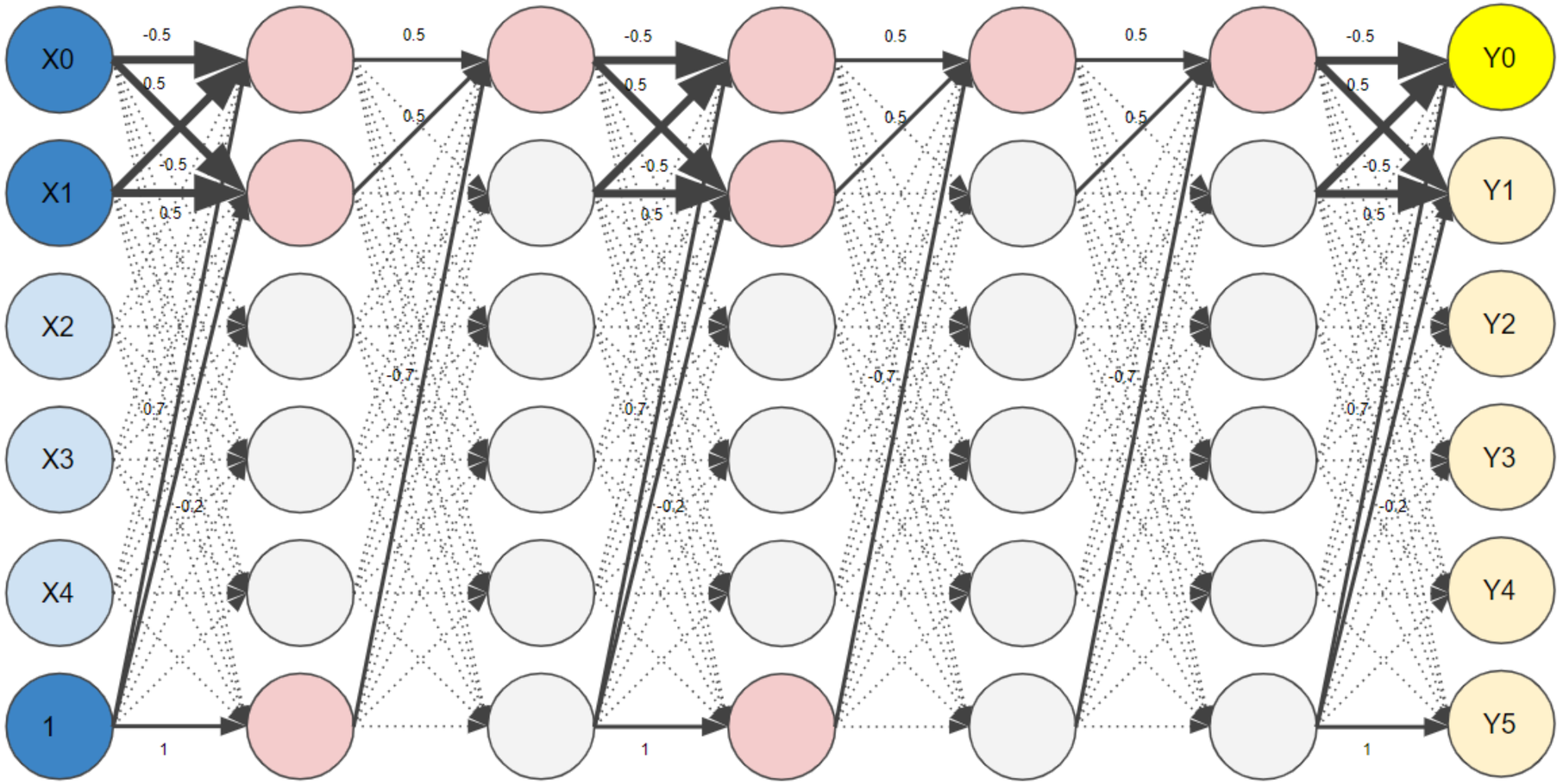
03-2

Convolution Layer

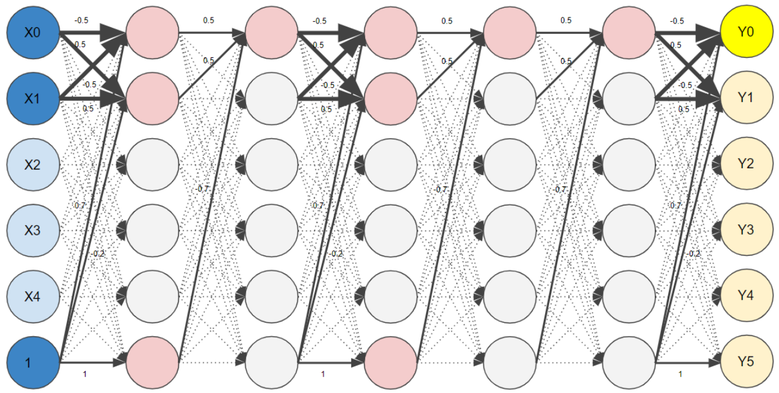
기본구조



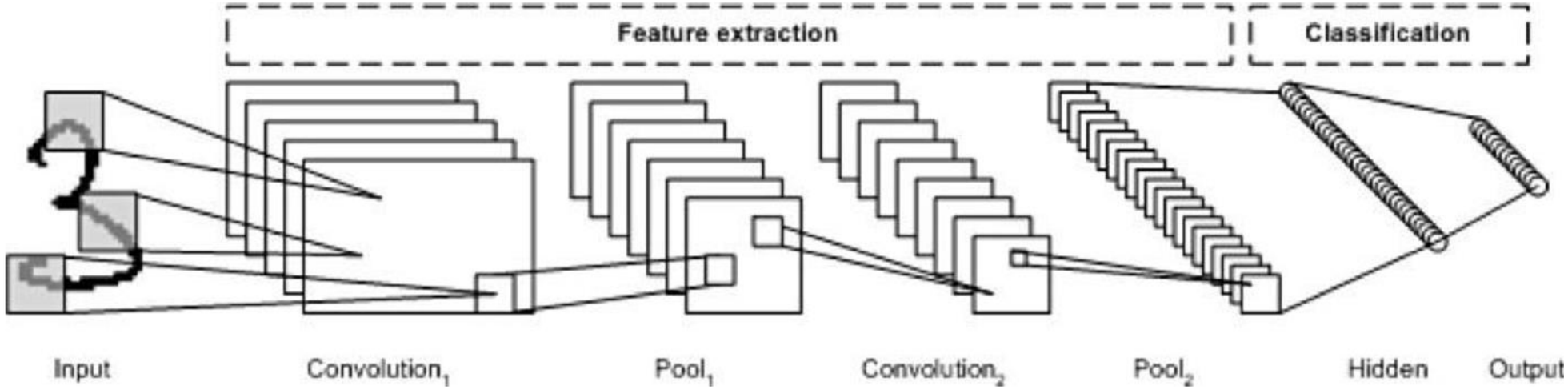
Fully Connected Layer



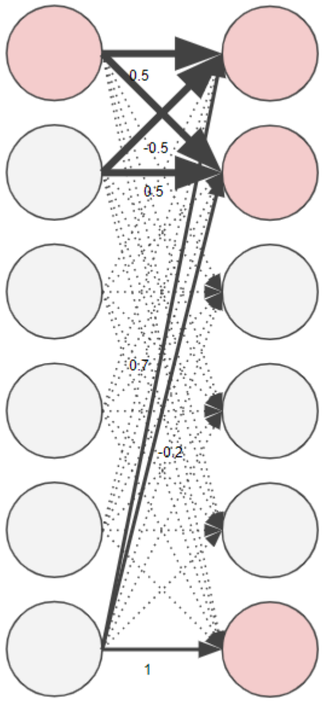
CNN 이 Fully Connected Layer 와 다른점

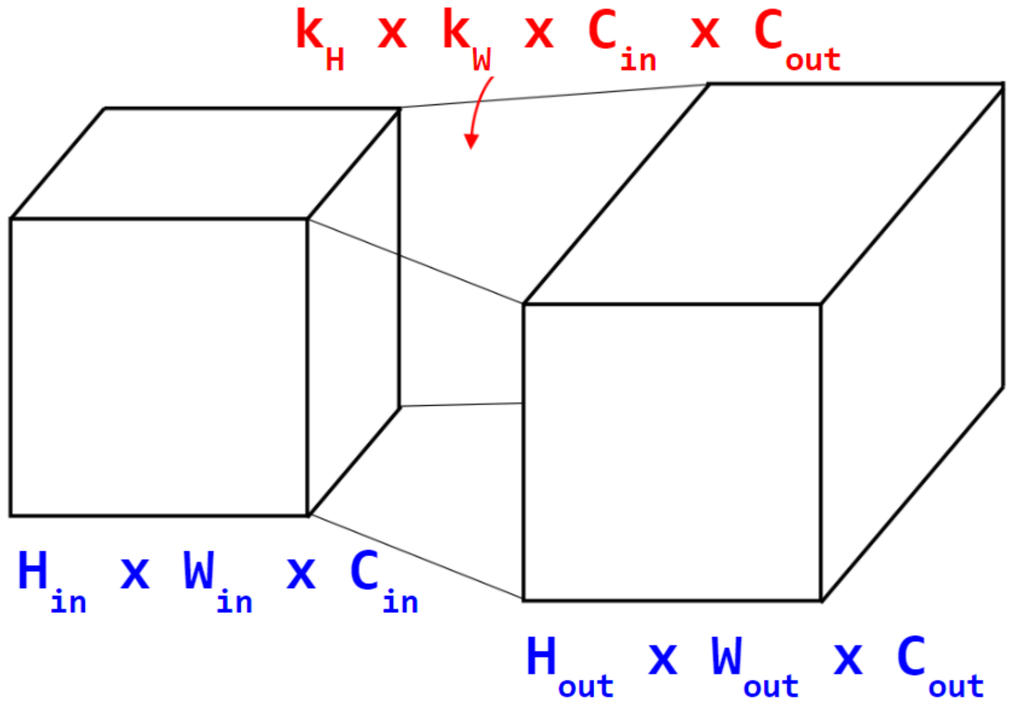


기존의 모양에서 많은 변화가 생긴다. 개별 입력값이 아닌 영상이 입력이 되고, 가중치 대신 Convolution Filter (Kernel), 일반 곱셈 대신 합성곱을 사용.

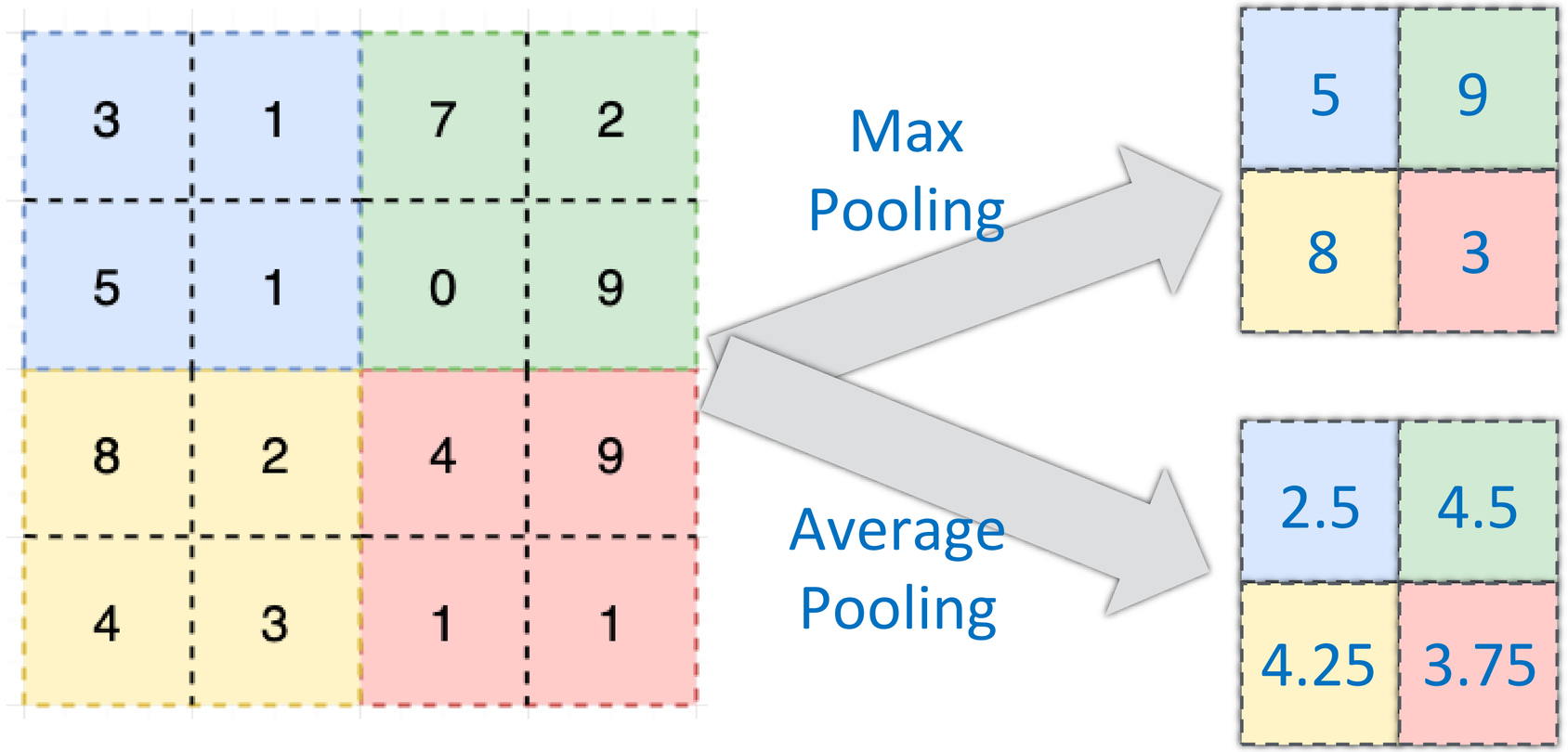


Fully Connected Layer <-> Convolution Layer

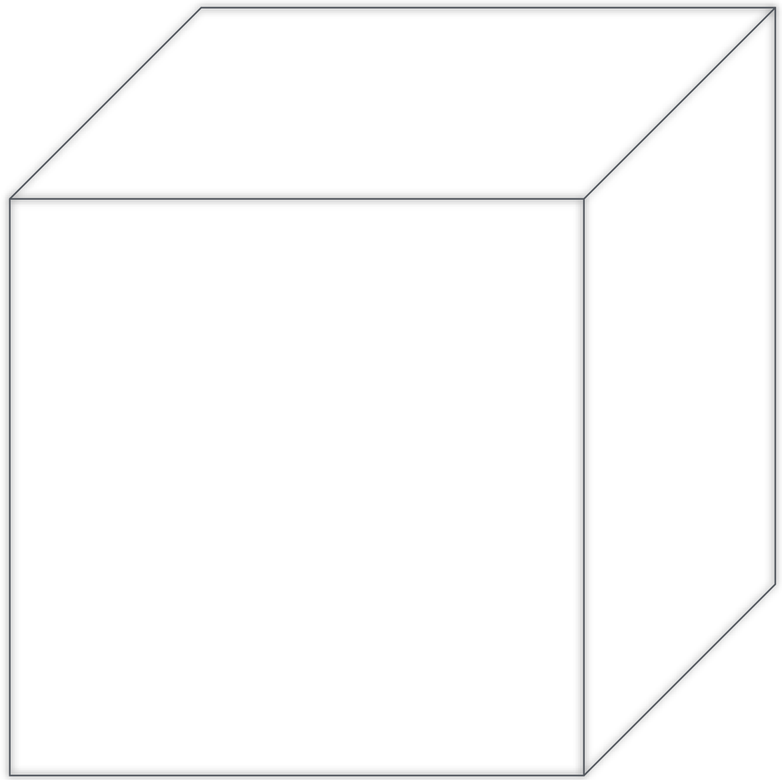




Pooling Layer (Sub Sampling)



Flatten, 평탄화



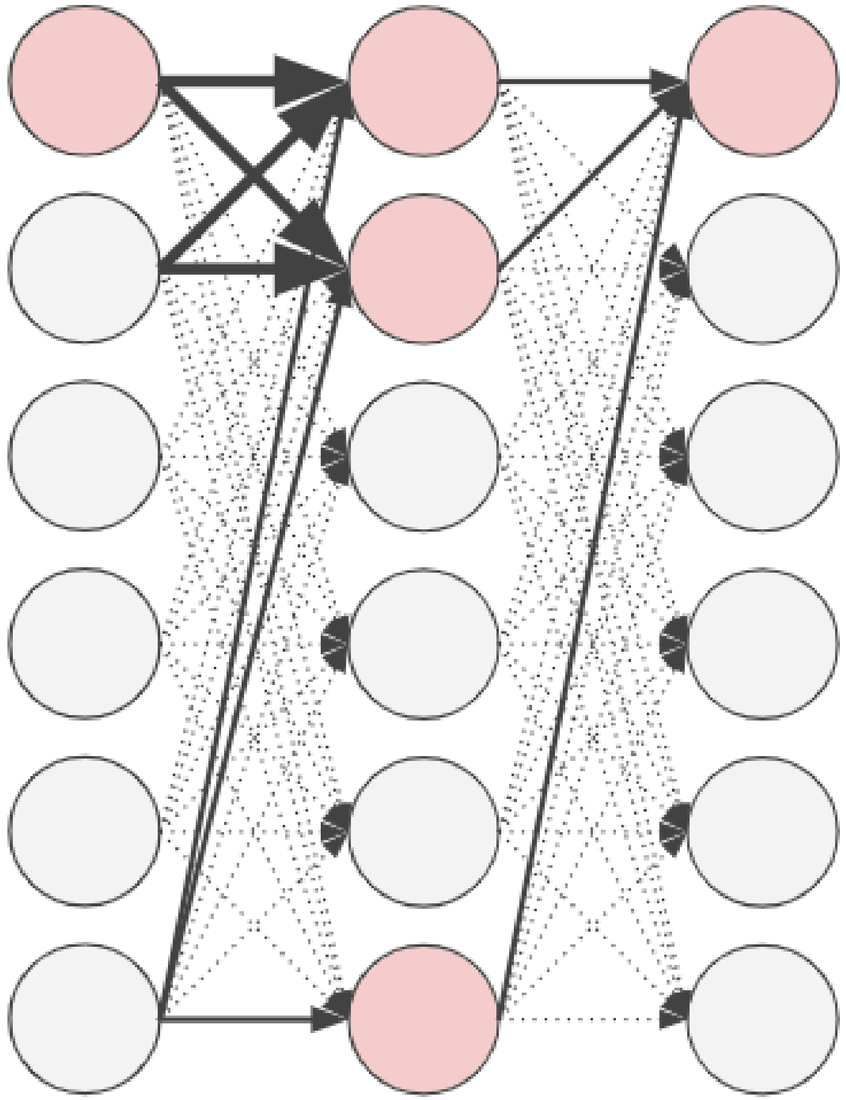
$H_{in} \times W_{in} \times C_{in}$

$H_{in} \times W_{in} \times C_{in}$

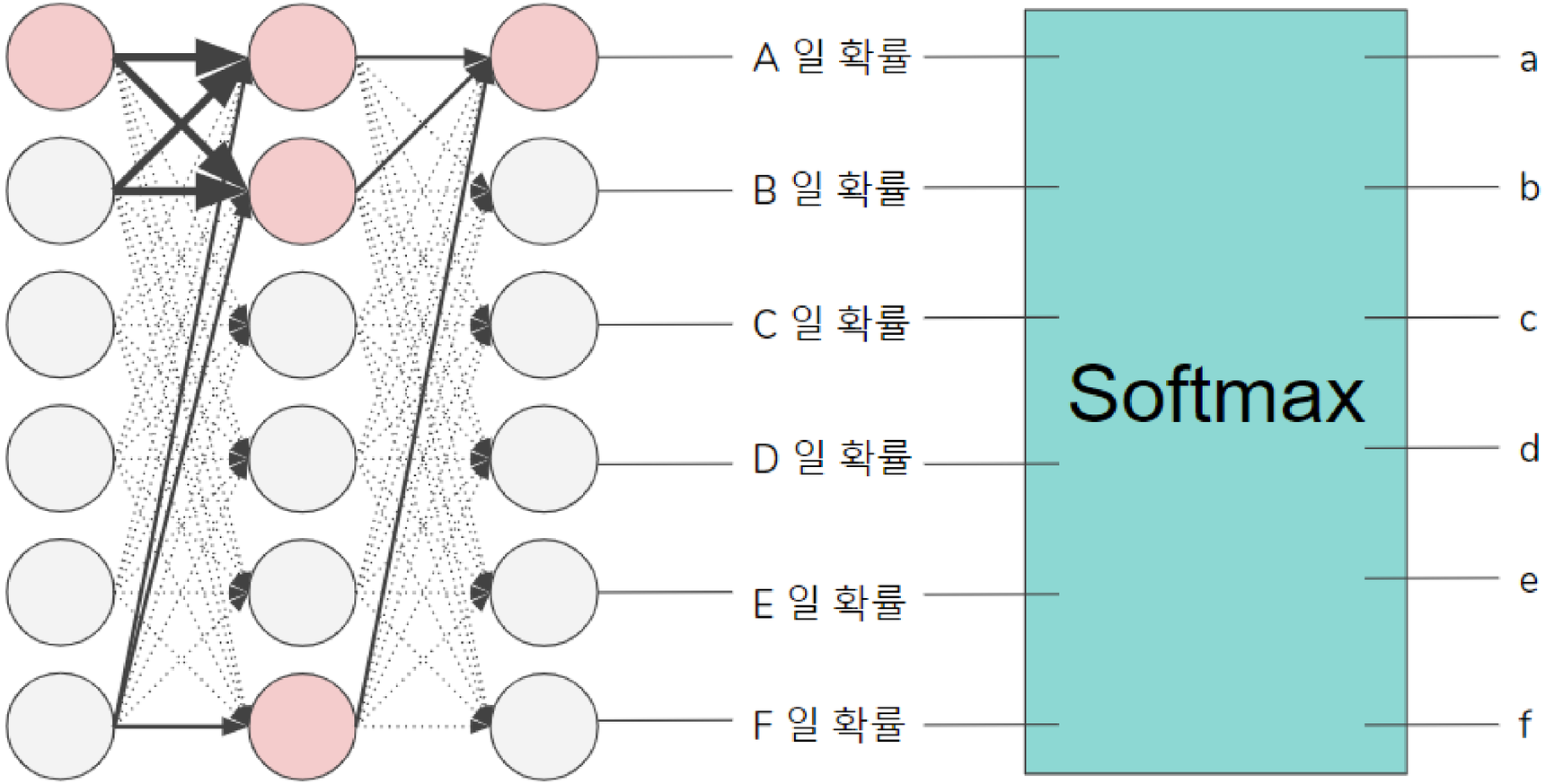
전결합 계층

일반적으로 평탄화를 마친 후

2개의 전결합 계층을 사용한다.



Softmax 함수



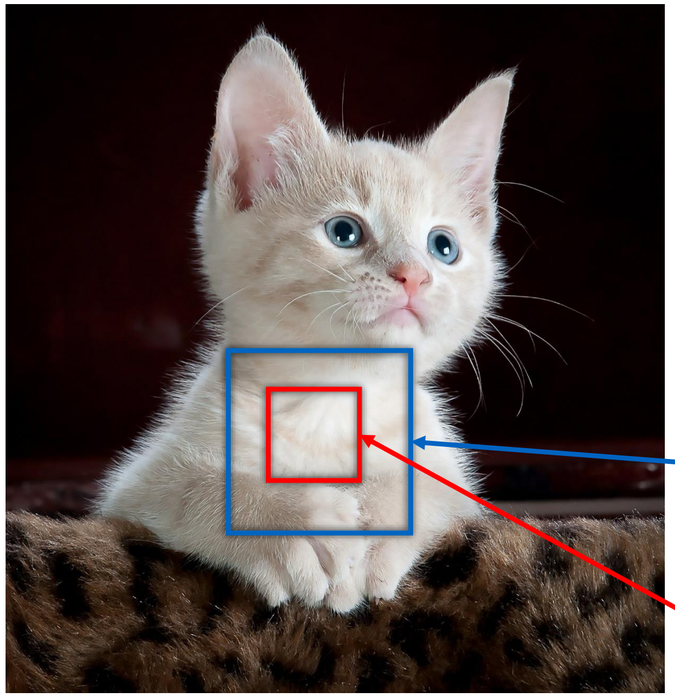
마지막 계층에 Softmax 활성함수 사용

a+b+c+d+e+f = 1, a,b,c,d,e,f >=0

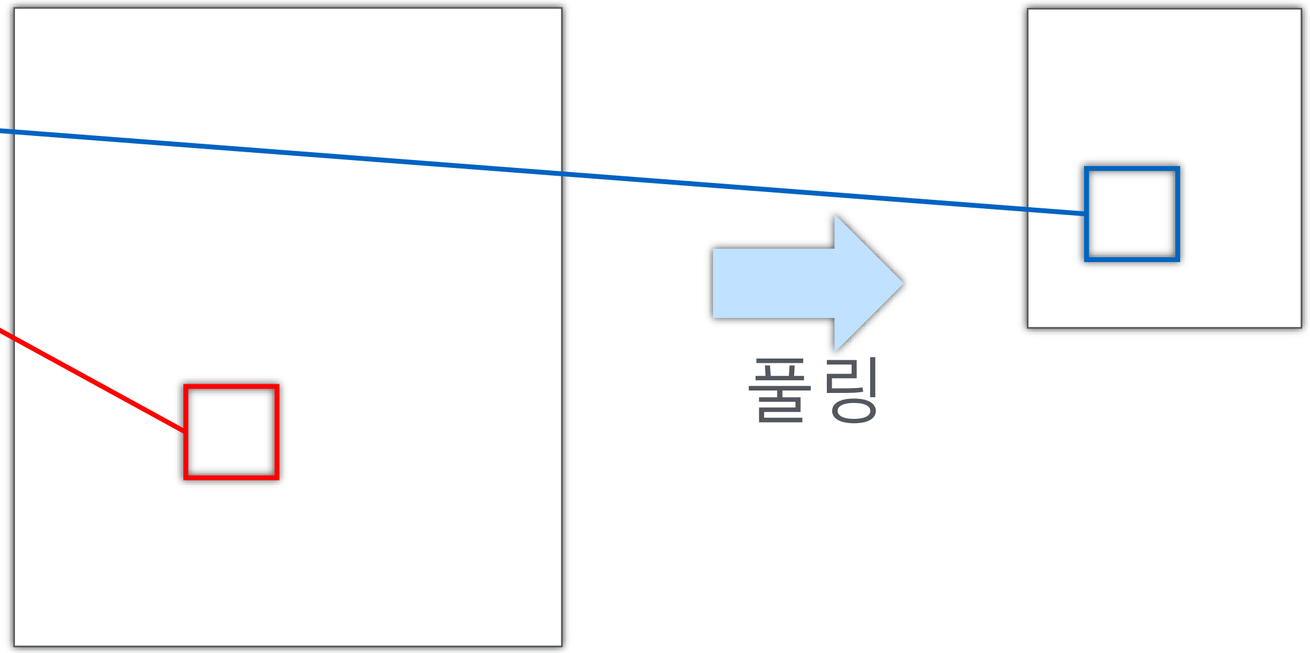
CNN 기본구조의 이해

Convolution Layer 는 특징을 찾아내고, Pooling Layer 는 처리할 맵(이미지) 크기를 줄여준다. 이를 N 번 반복한다. 반복할 때마다 줄어든 영역에서의 특징을 찾게 되고, 영역의 크기는 작아졌기 때문에 빠른 학습이 가능해진다.

Receptive Field



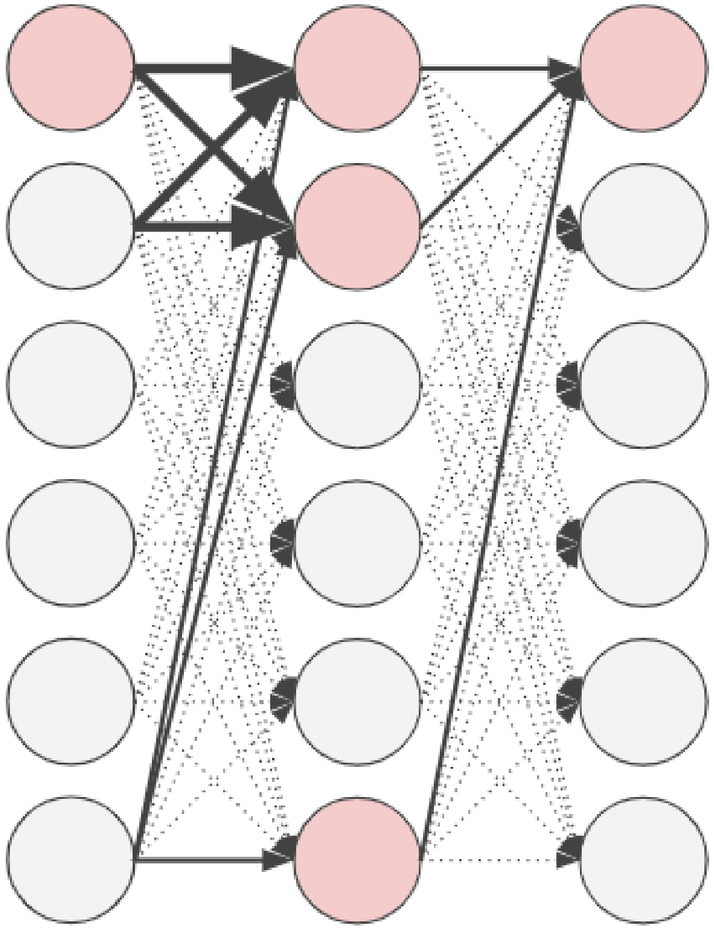
필터의 크기는 동일하지만 풀링에 의해 처리할 영역이 줄어들기 때문에 실제로 원본영상에서 필터가 차지하는 범위가 넓어진다. 이 범위를 Receptive Field 라 부른다.



03 - 3

Convolution Layer 의 수식이해

전결합 계층의 수학적 표현



$$y_0 = a(\boldsymbol{w}_0^T \boldsymbol{x} + b_0)$$

$$y_1 = a(\boldsymbol{w}_1^T \boldsymbol{x} + b_1)$$

$$\vdots$$

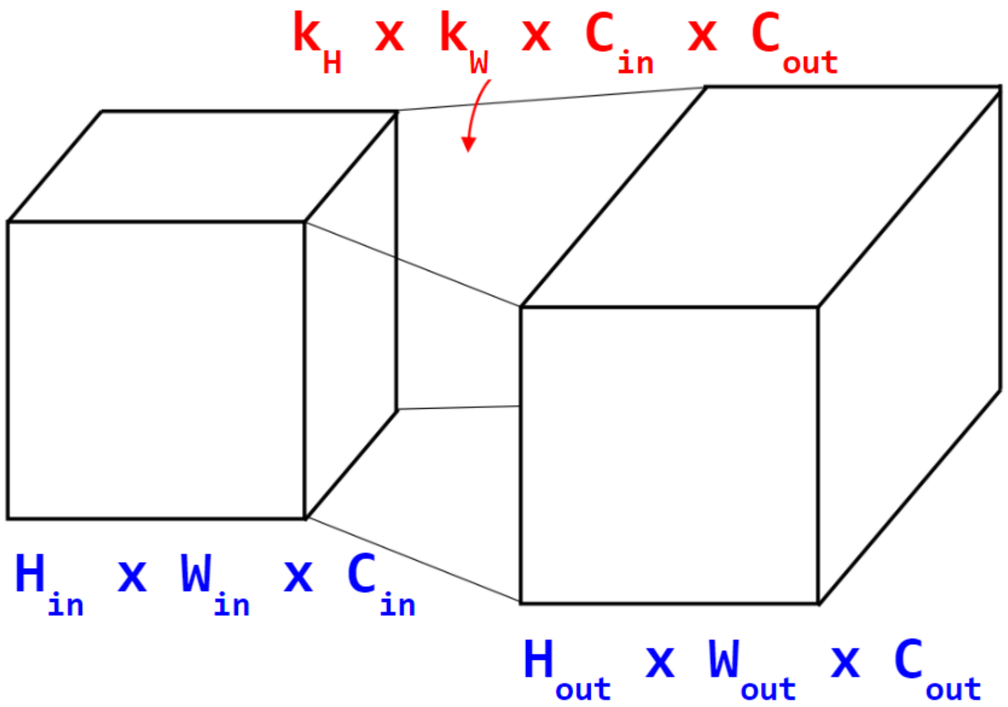
$$y_{M-1} = a(\boldsymbol{w}_{M-1}^T \boldsymbol{x} + b_{M-1})$$

$$W = [\boldsymbol{w}_0, \boldsymbol{w}_1, \dots, \boldsymbol{w}_{M-1}]^T$$

$$\boldsymbol{b} = [b_0, b_1, \dots, b_{M-1}]^T$$

$$y = a(Wx + b)$$

합성곱 계층의 수학적 표현



3,5,7 사용

$$\boldsymbol{W} = \begin{bmatrix} W_{0,0} & \cdots & W_{0,M-1} \\ \vdots & \ddots & \vdots \\ W_{N-1,0} & \cdots & W_{N-1,M-1} \end{bmatrix}$$

$$\boldsymbol{b} = [b_0, b_1, \ldots, b_{M-1}]^T$$

$$Y_{i,j} = a(W_{i,j} * X_i + b_j)$$

Padding

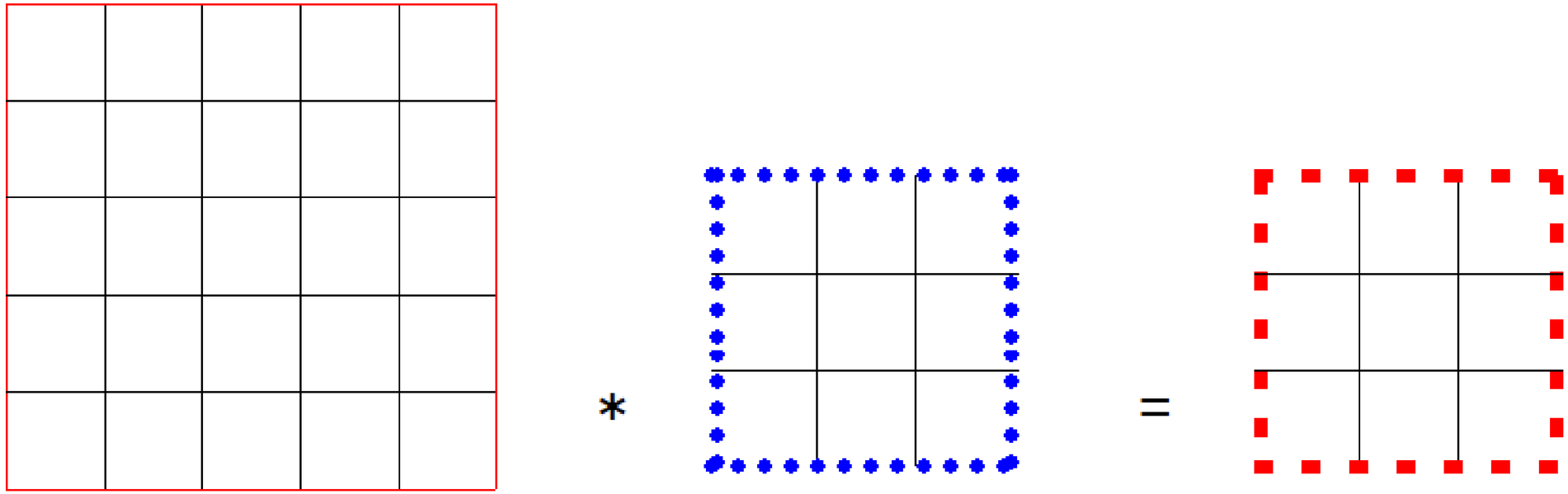
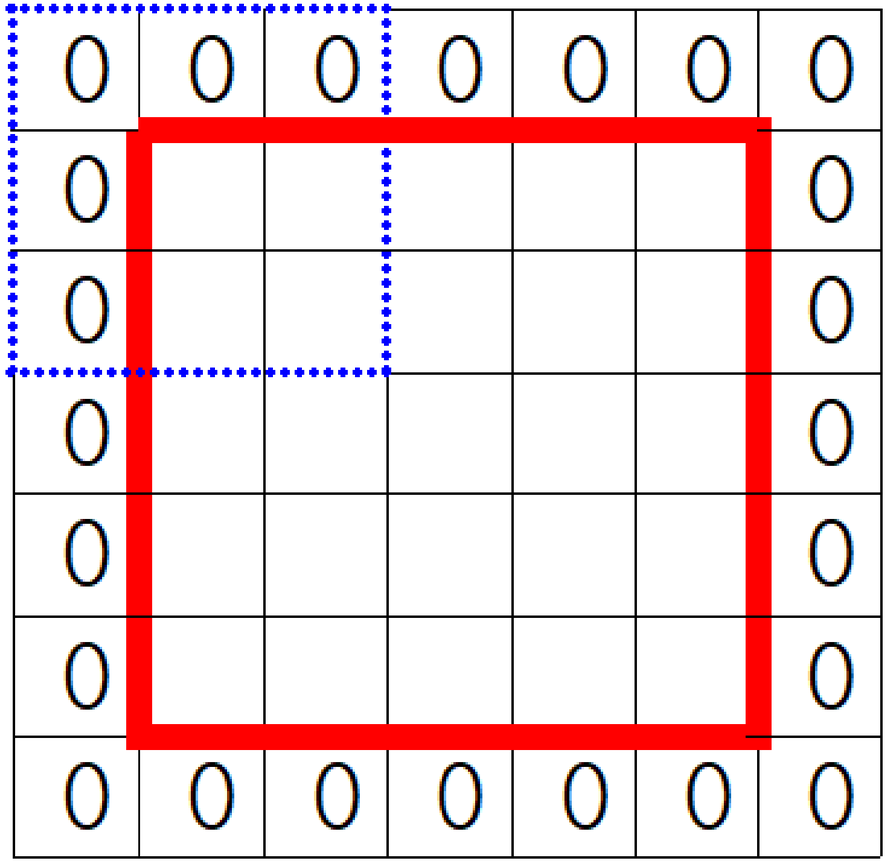


Image Filter(Kernel) Convolution Result

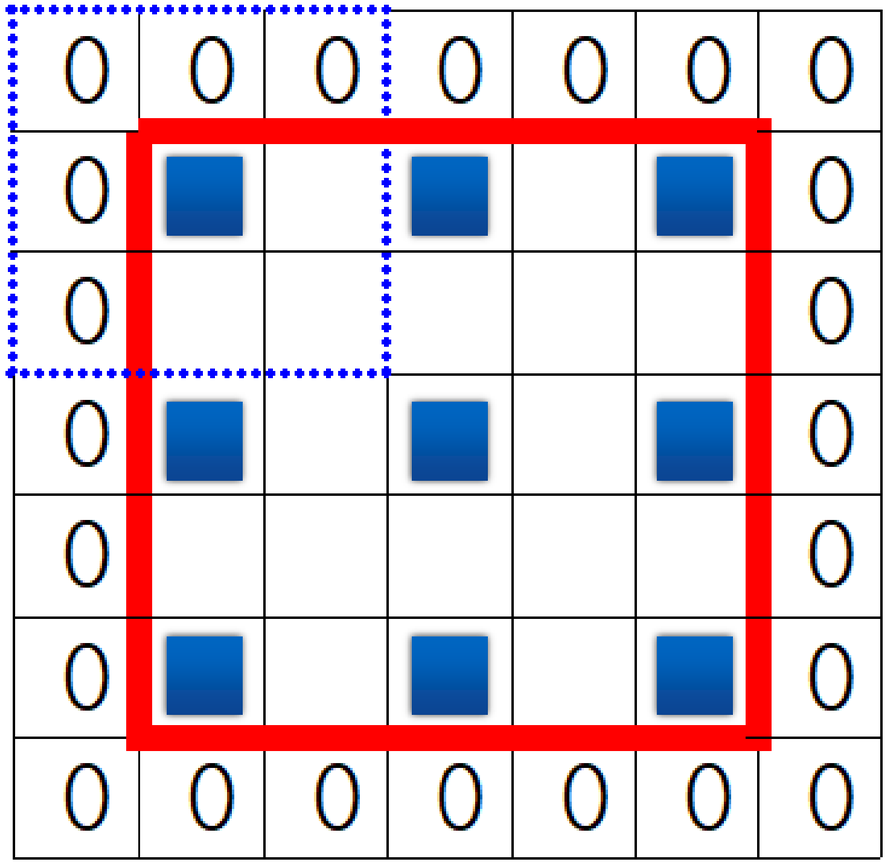
3 x 3 Filter 를 쓰면 상, 하, 좌, 우 모두 1줄씩 처리를 못하게 된다. 원본에서 가로, 세로 2줄씩 작아지는 결과

Padding



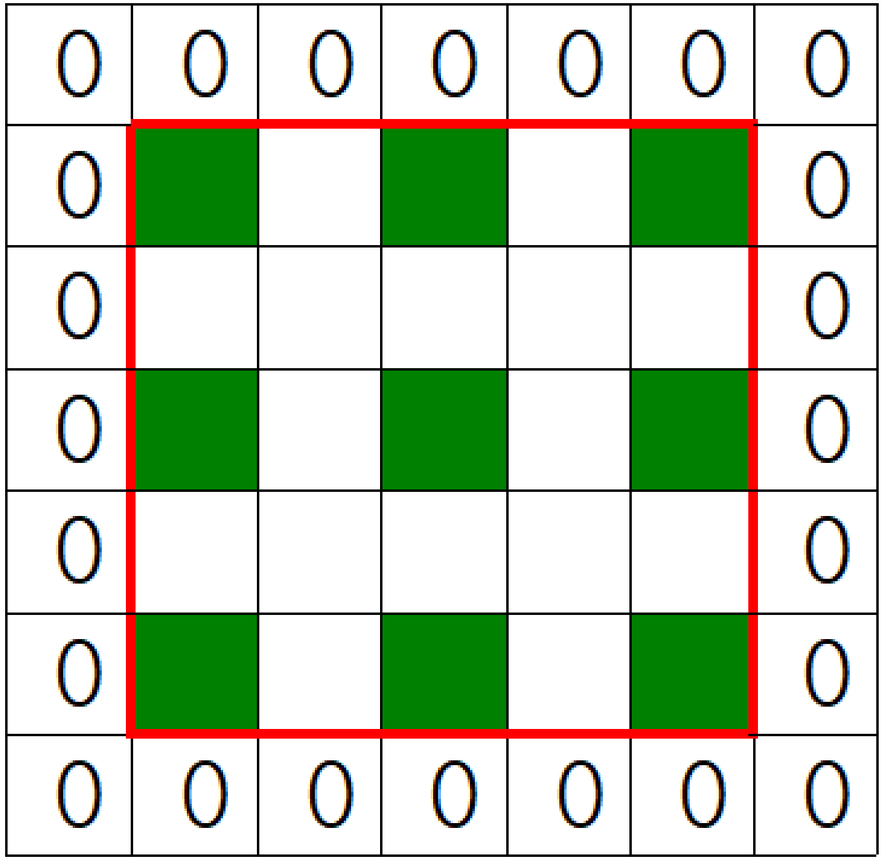
원본 이미지의 상하좌우에 한 줄씩 추가하여 0으로 채워 넣는 것을 0-Padding 이라 한다.

Stride



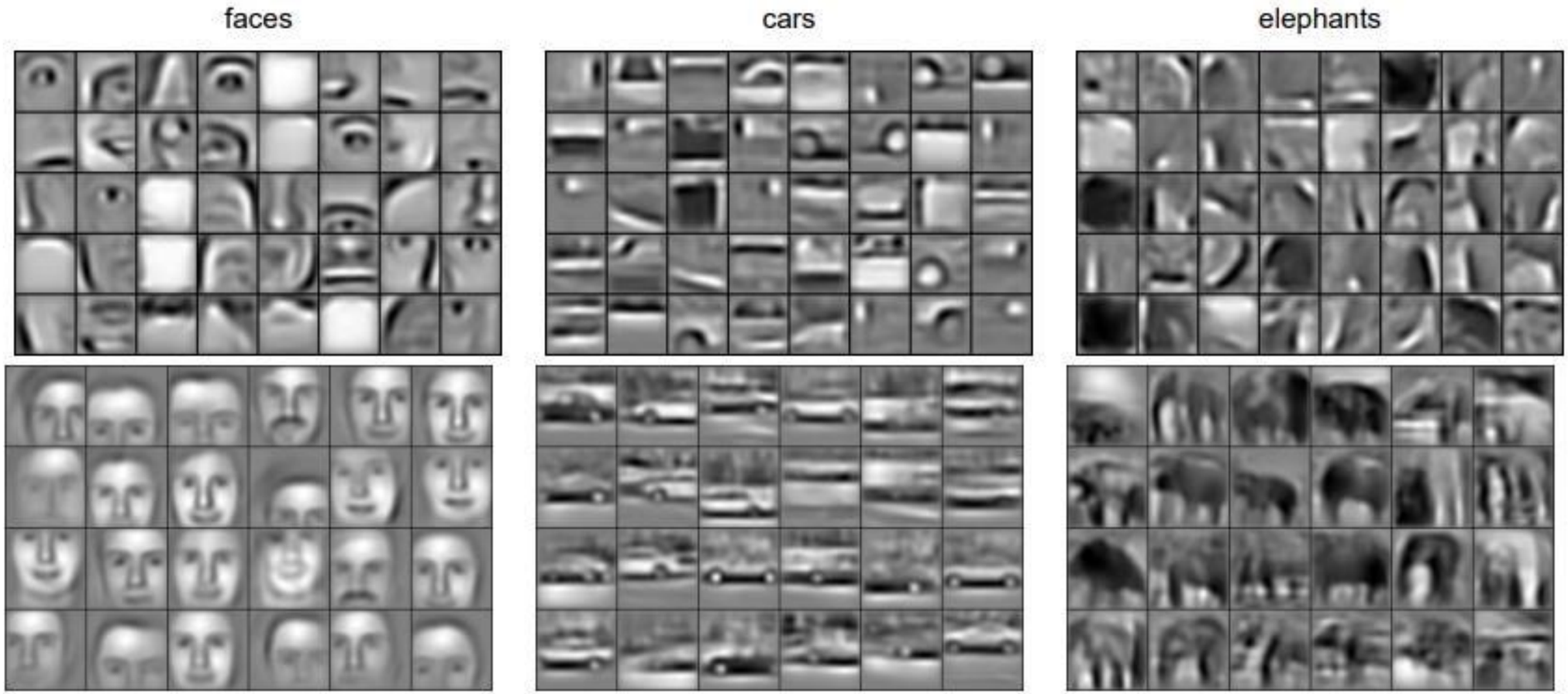
커널(Convolution Filter)을 이동시키는 거리를 Stride 라 하고, Stride 가 2 이상이면 출력의 크기가 줄어든다.

Sub Sampling



Pooling 과 거의 같은 의미로 사용, Pooling 은 최대값 혹은 평균값을 사용하는 반면 Sub-Sampling 은 일정거리에 있는 값만을 사용

특징맵



CNN 추출된 특징(feature) 맵, - 2009

04

CIFAR-10 으로 CNN 실습

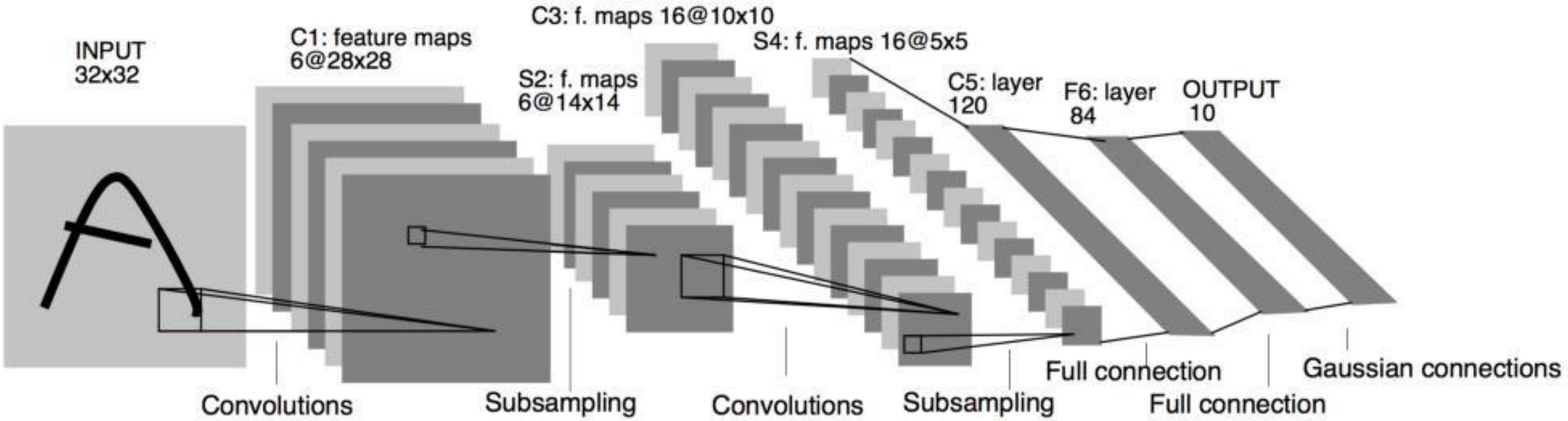
05

VGGNet 실습

06

Deep Learning 이미지 처리 역사와 동향분석

LeNet-5

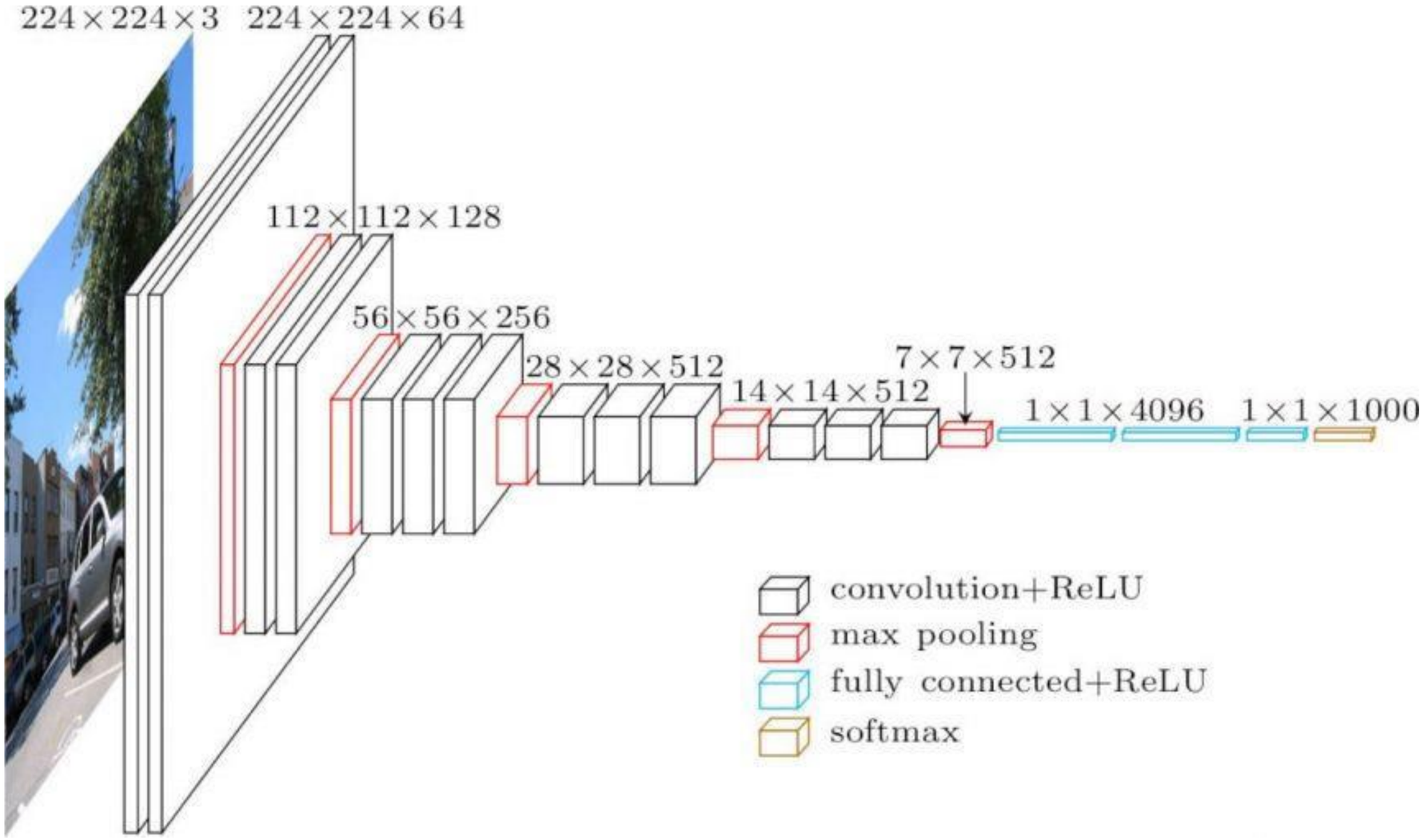


CNN 의 조상격인 LeNet 은 다양한 버전이 있으며 그중 최종 버전은 LeNet-5 이다.

LeNet 은 CNN 을 처음으로 시도한 얀 르쿤(Yann LeCunn)이 1998년 개발한 CNN 알고리즘의 이름이다.

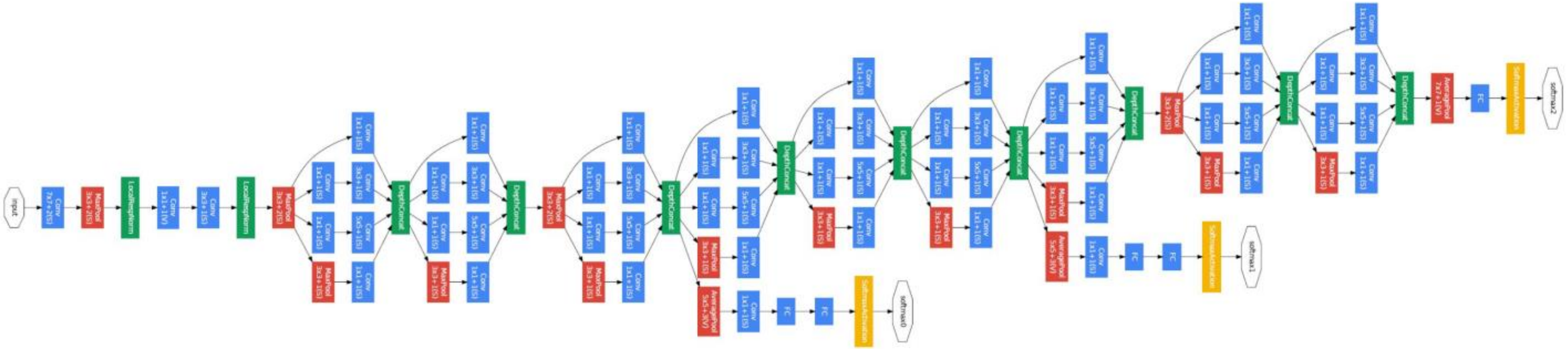
/* elice */

VGG-16



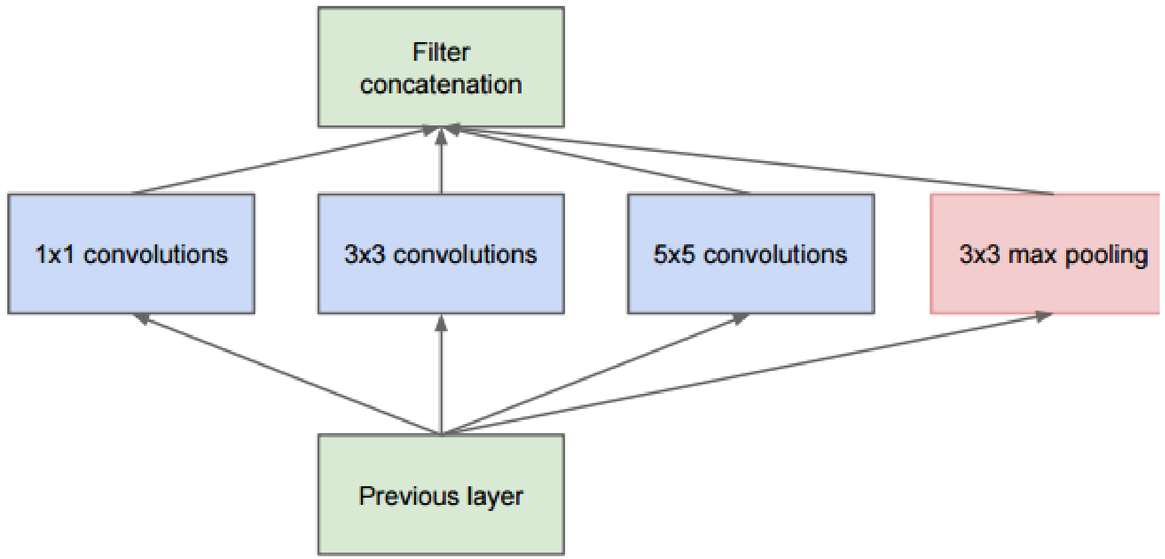
2014년 ILSVRC 에서 92.7% 의 정확도(전년 대비 4.3%p 향상)를 보인 VGG-16

GoogleNet (Inception)

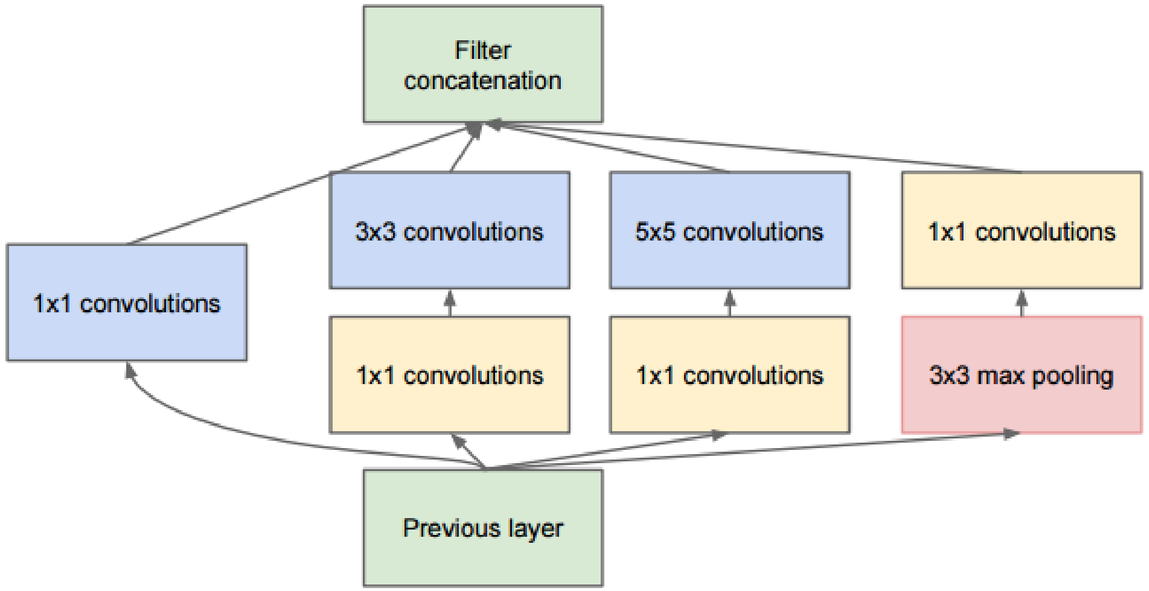


2014년 ILSVRC 에서 93.3% 의 정확도로 1위

Inception module



(a) Inception module, naïve version



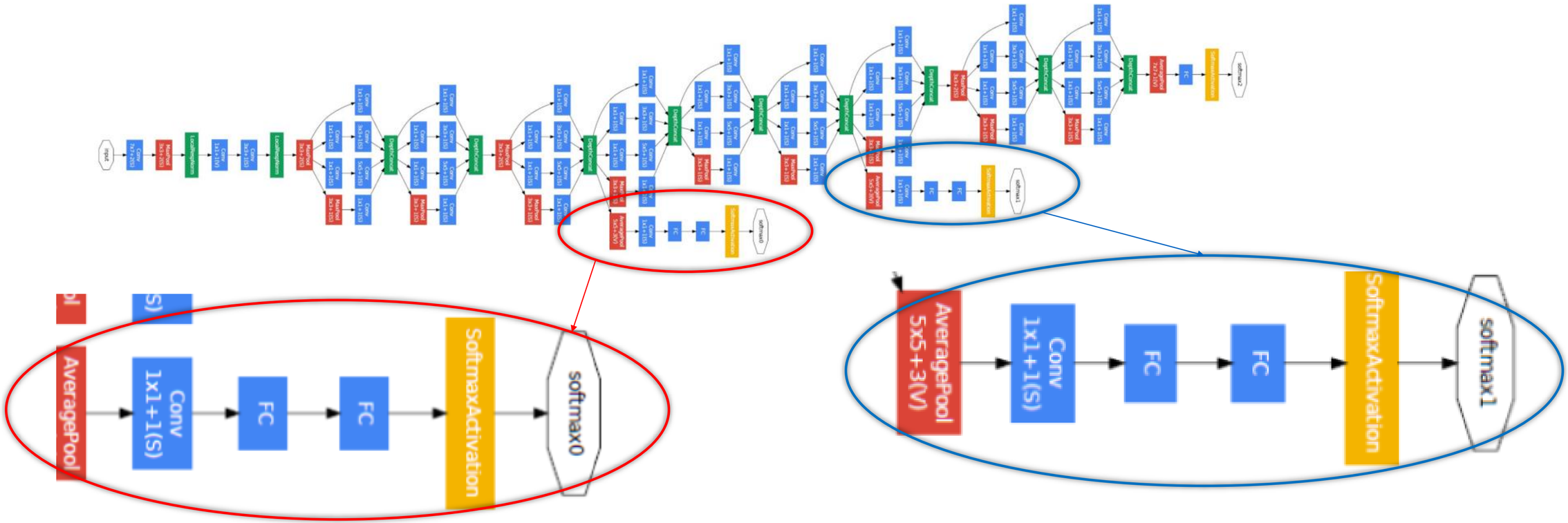
(b) Inception module with dimension reductions

다양한 크기의 합성곱 계층을 한번에 계산

연산량을 줄이기 위한 구조

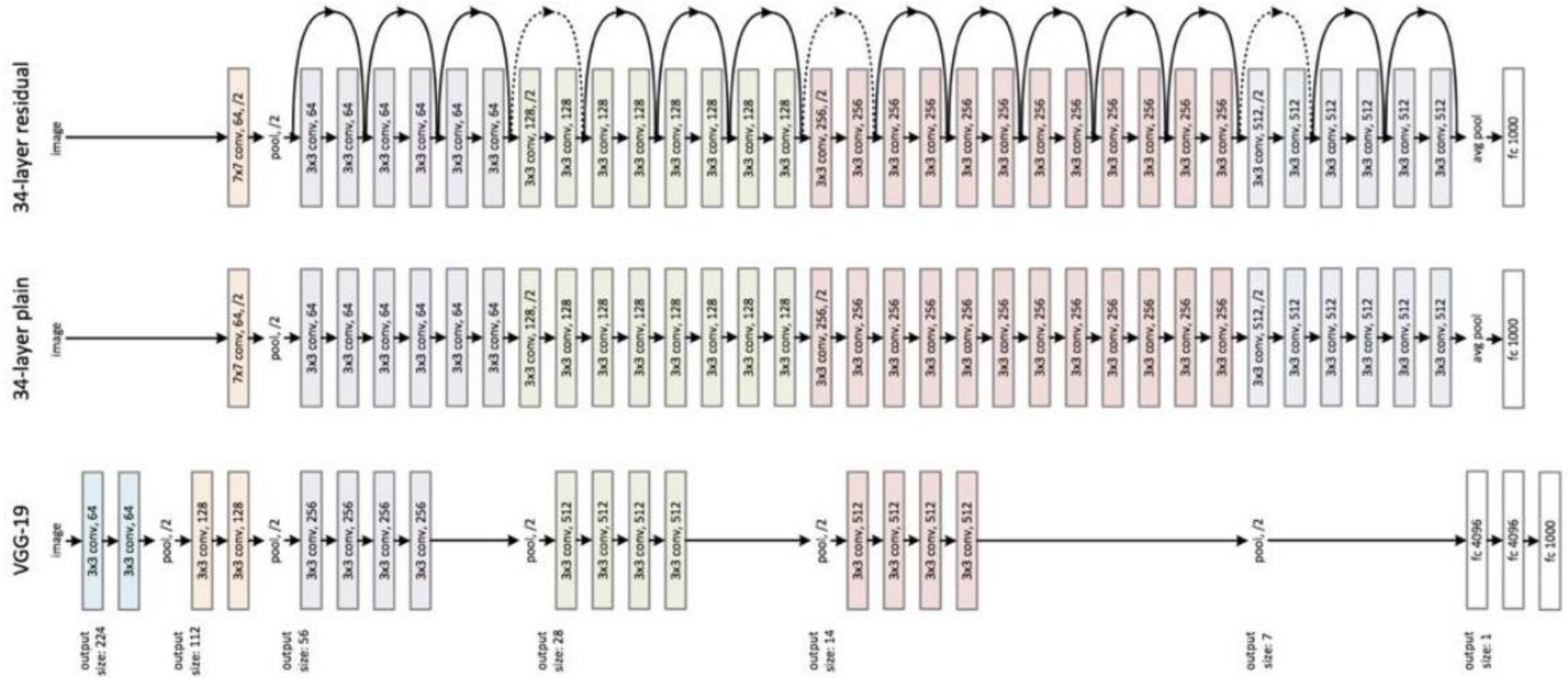
/* elice */

Inception module – 추가 분류기 사용



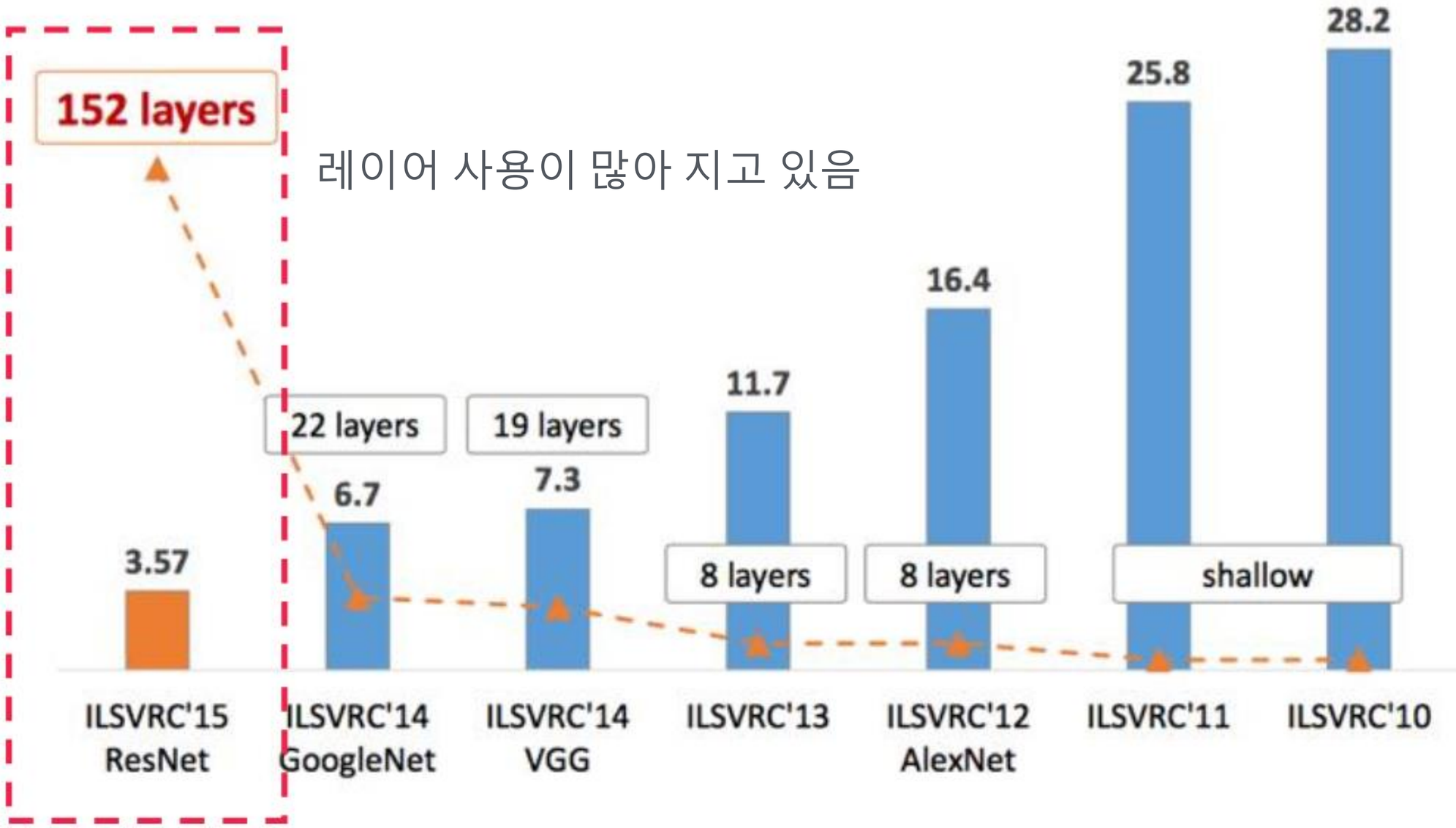
역전파 기울기소실방지를 위한 추가 분류기 사용

ResNet



Skip-Connection 과 많은 레이어 사용이 특징

ILSVRC



레이어 사용이 많아 지고 있음

Credit

/* elice */

코스 매니저
임승연

콘텐츠 제작자
임승연

강사
오혜연 교수님

감수자
김수인

디자인
박주연

Contact

TEL
070-4633-2015

WEB
https://elice.io

E-MAIL
contact@elice.io