

사람의 시각을 닮은 신경망

0
0
1
1
1

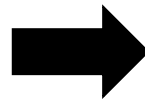
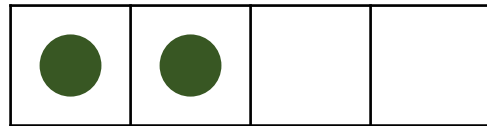
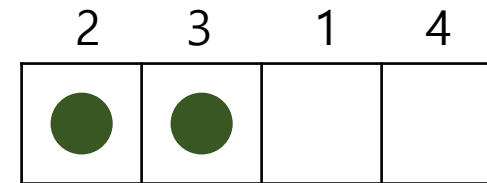
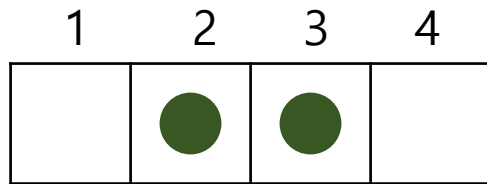
다차원 데이터 표현

- 지금까지 살펴본 신경망은 데이터를 구성하고 있는 특징들을 고려하지 않음
- 데이터들의 차원은 서로 독립적이라고 보았음

	(70 , 20)
(국어, 영어)	(80 , 40)
	(90 , 90)
	.
	.
	.
	(20 , 70)
(영어,국어)	(40 , 80)
	(90 , 90)

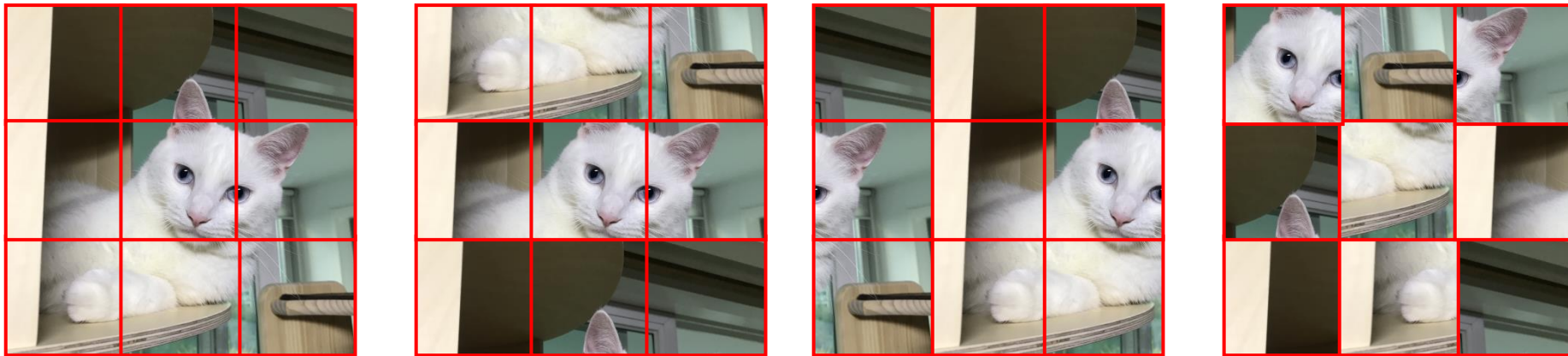
위치 정보가 중요한 다차원 데이터 표현

- 시각적으로 다르게 인식할 수 있는 패턴임에도 신경망 알고리즘에서는 구분하지 않았음



영상 데이터 표현

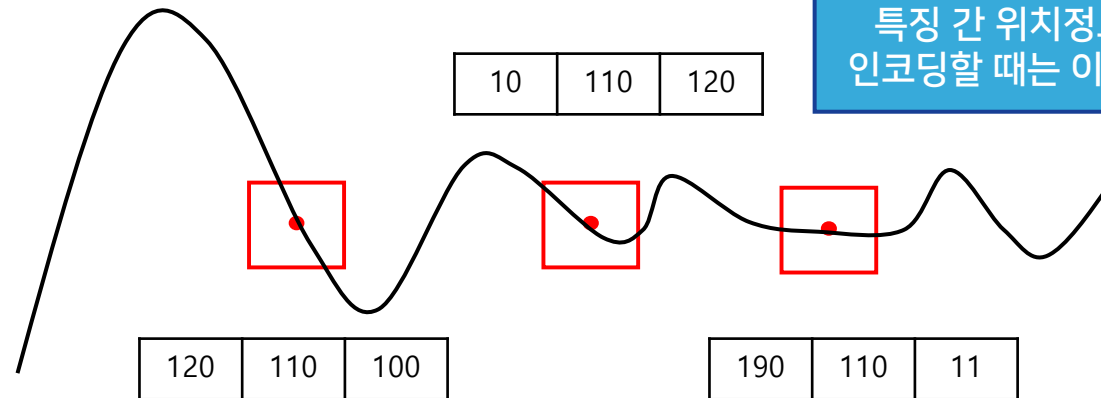
- 인간의 시각 정보 체계를 신경망에서 반영하고 있지 않음



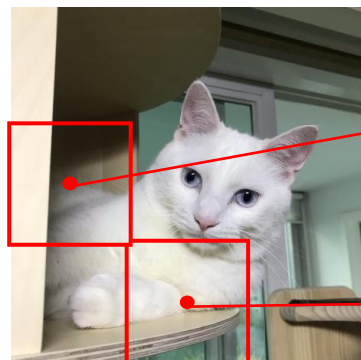
주변 정보를 고려한 데이터 표현

- 주변 특징 정보를 포함한 표현 혹은 인코딩 방법 필요

1차원 신호



2차원 신호

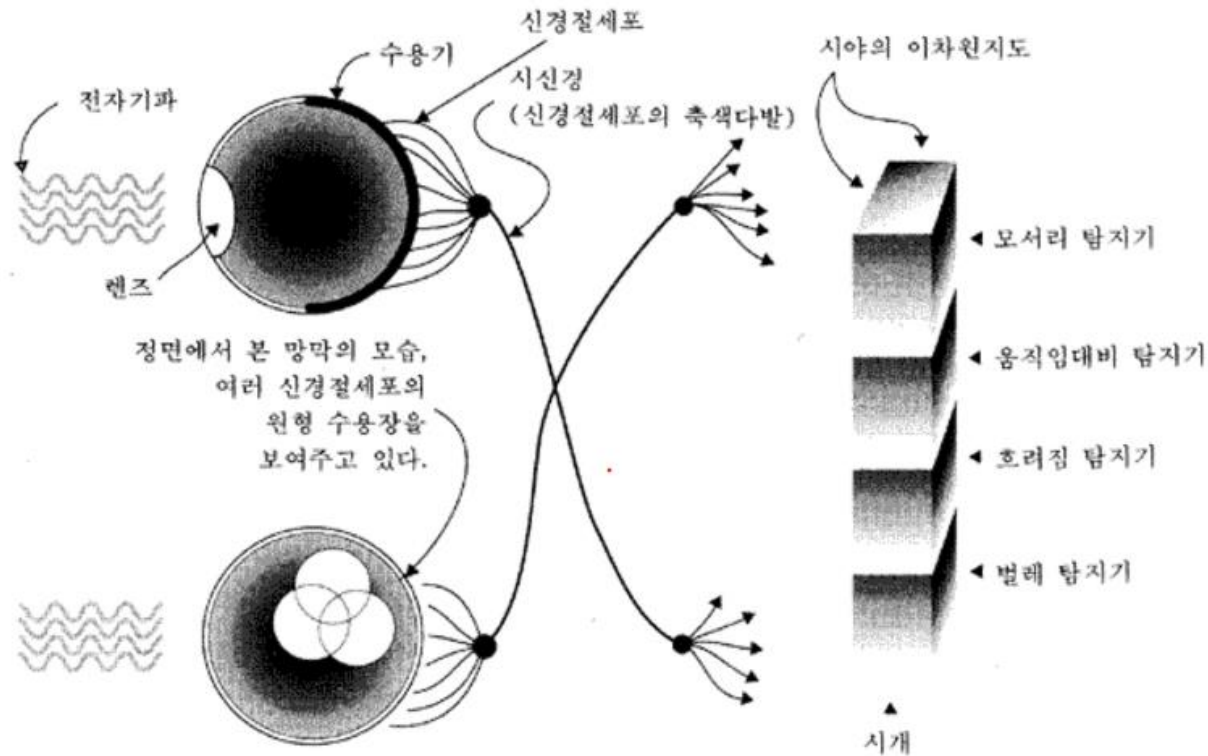


200	200	225
200	225	210
150	8	80

190	0	0
190	225	150
190	200	200

생물학적 영감

● 시신경에는 단순한 특정 자극에 반응하는 필터가 존재



http://www.aistudy.com/cognitive/pattern_martindale.htm

공간 정보 인코딩

- 주변의 특징 정보를 추가해 데이터의 특징 차원을 늘림



3차원

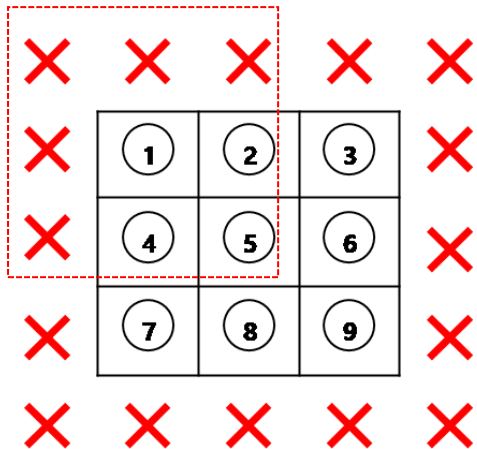


9차원

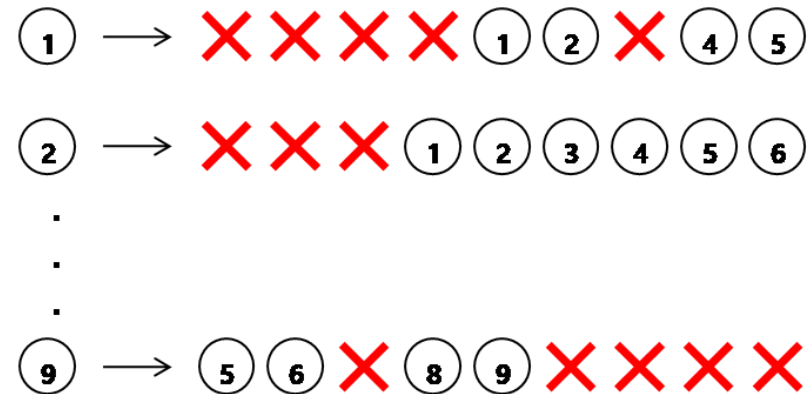
선형 함수 조합의 한계

- 특징 수 증가 → 신경망 크기 증가 → 최적화된 학습 결과를 얻기 어렵고, 충분한 양의 데이터 확보의 어려움 발생

9차원



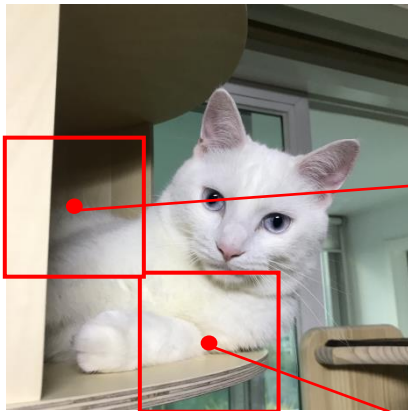
81차원



예) 128X128 영상 데이터 인코딩 -> 16384 X 9 = 147456 차원

효율적인 인코딩 #1

- 한 개의 특징에 주변 정보를 포함해서 정의: 주변 정보 평균
 - > 특징의 수를 유지하면서 같은 밝기 값을 갖는데 주변의 밝기 정보에 따라 값이 다르게 나옴



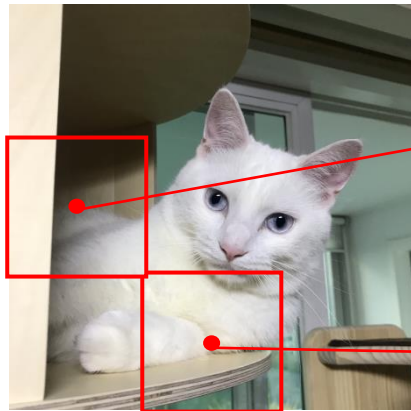
200	200	225
200	225	210
150	8	80

평균 :
 $200+200+225\cdots / 9$
 $= 166$

190	220	220
190	225	150
190	200	200

평균:
 $190+200+220\cdots / 9$
 $=198$

효율적인 인코딩 #2



200	200	225
200	225	210
150	8	80

190	220	220
190	225	150
190	200	200

상하좌우 4개

200	200	225
200	225	210
150	8	80

168

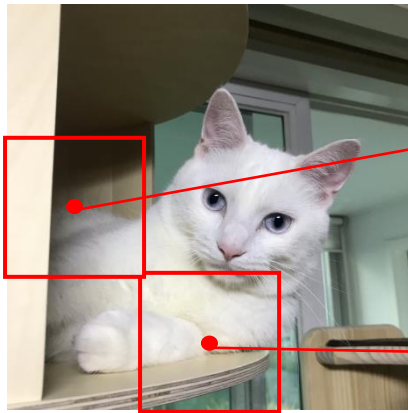
수직 2개

190	220	220
190	225	150
190	200	200

208

효율적인 인코딩 #3

- 한 개의 특징에 주변 정보를 포함해서 정의 : 뱀셈 연산 사용



200	200	225
200	225	210
150	8	80

190	100	220
190	225	150
190	200	200

위아래의 차

200	200	225
200	225	210
150	8	80

$$200 - 8 = 192$$

위아래의 차

190	100	220
190	225	150
190	200	200

$$100 - 200 = -100$$

2D 컨볼루션

- 컨볼루션 : 주변 정보를 인코딩하는 방법

I_1	I_2	I_3
I_4	I_5	I_6
I_7	I_8	I_9

(a) 입력 영상

M_1	M_2	M_3
M_4	M_5	M_6
M_7	M_8	M_9

(b) 컨볼루션 필터

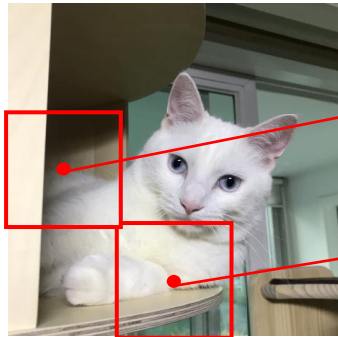
출력 픽셀 값 : $I_1 \times M_1 + I_2 \times M_2 + I_3 \times M_3 + I_4 \times M_4 + I_5 \times M_5 +$
 $I_6 \times M_6 + I_7 \times M_7 + I_8 \times M_8 + I_9 \times M_9$

입력 화소와 이웃한 각 화소에 가중치를 곱한 합을 출력 화소로 생성

$$\text{output} = \sum_{k=1}^K I_K \times M_K$$

2D 컨볼루션

- 평균을 이용한 주변 정보 인코딩은 전체 데이터가 가중치 1/9로 구성된 행렬을 사용해 표현할 수 있음



200	200	225
200	225	210
150	8	80

190	0	0
190	225	150
190	200	200

200	200	225
200	225	210
150	8	80

(*)

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

➡ 192

190	0	0
190	225	150
190	200	200

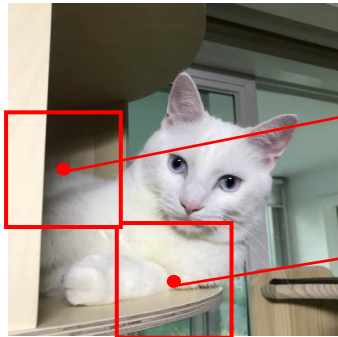
(*)

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

➡ 123

2D 컨볼루션

- 밝기차이를 인코딩하는 필터



200	200	225
200	225	210
150	8	80

190	0	0
190	225	150
190	200	200

200	200	225
200	225	210
150	8	80

(*)

0	+1	0
0	0	0
0	-1	0

➡ 192

190	0	0
190	225	150
190	200	200

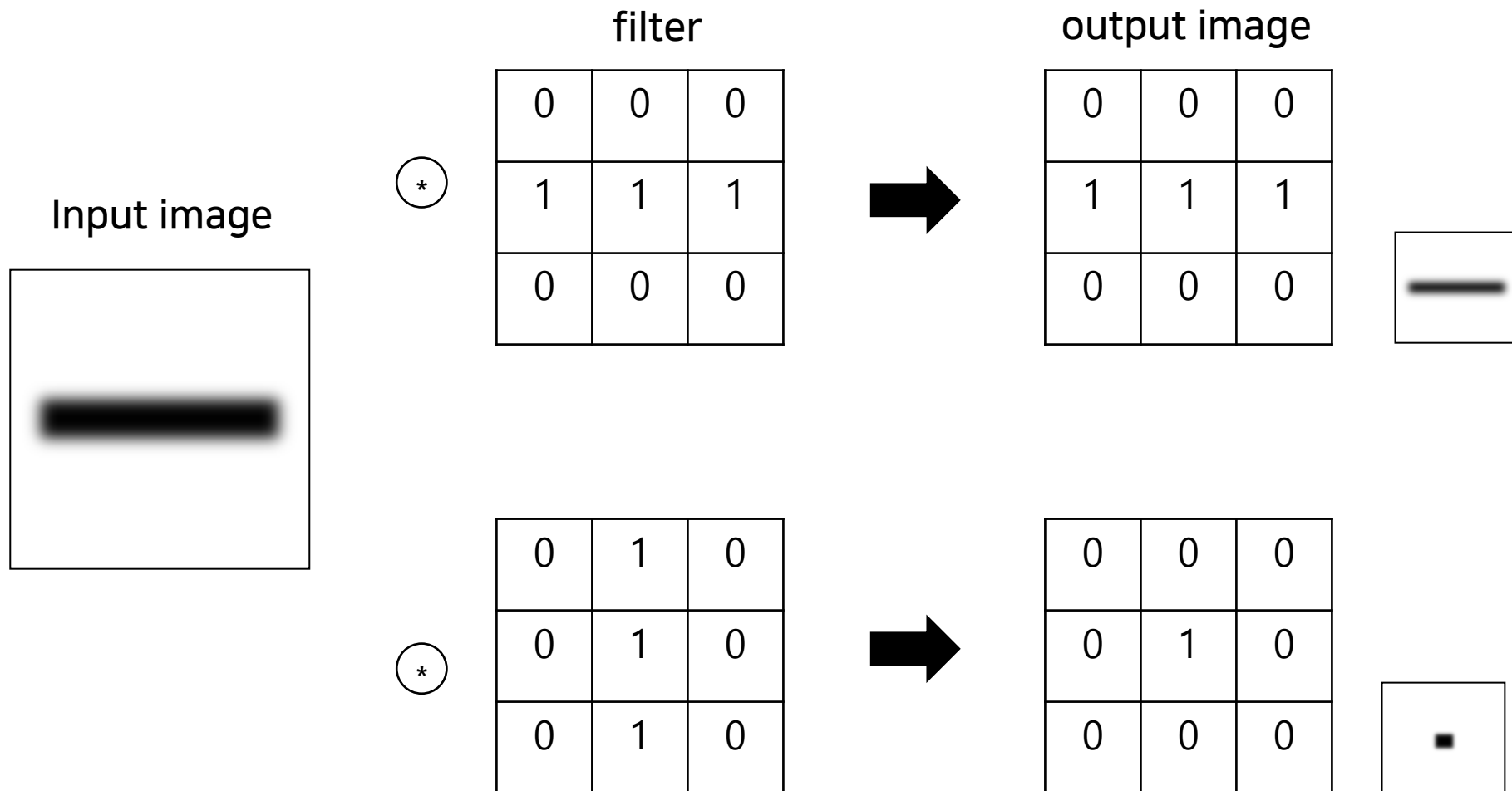
(*)

0	0	0
+1	0	-1
0	0	0

➡ -10

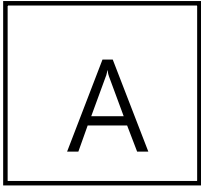
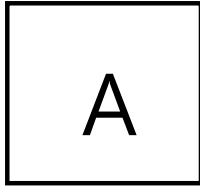
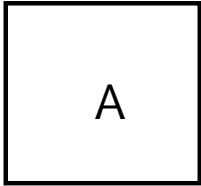
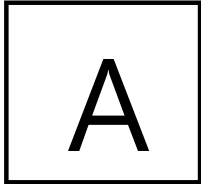
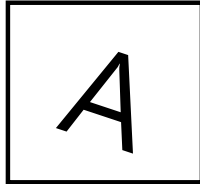
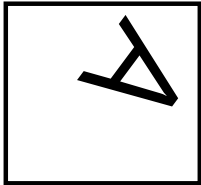
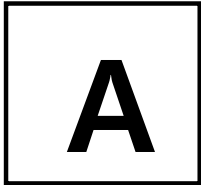
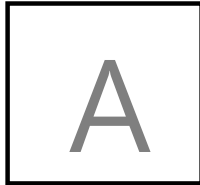
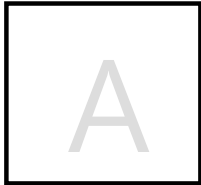
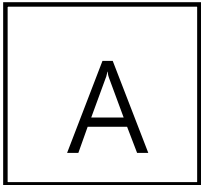
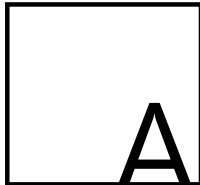
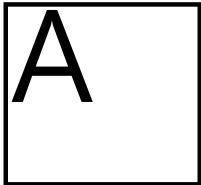
2D 컨볼루션을 이용한 특징 검출

- 다수의 필터를 정의해 입력 영상의 기하학적 특징 파악



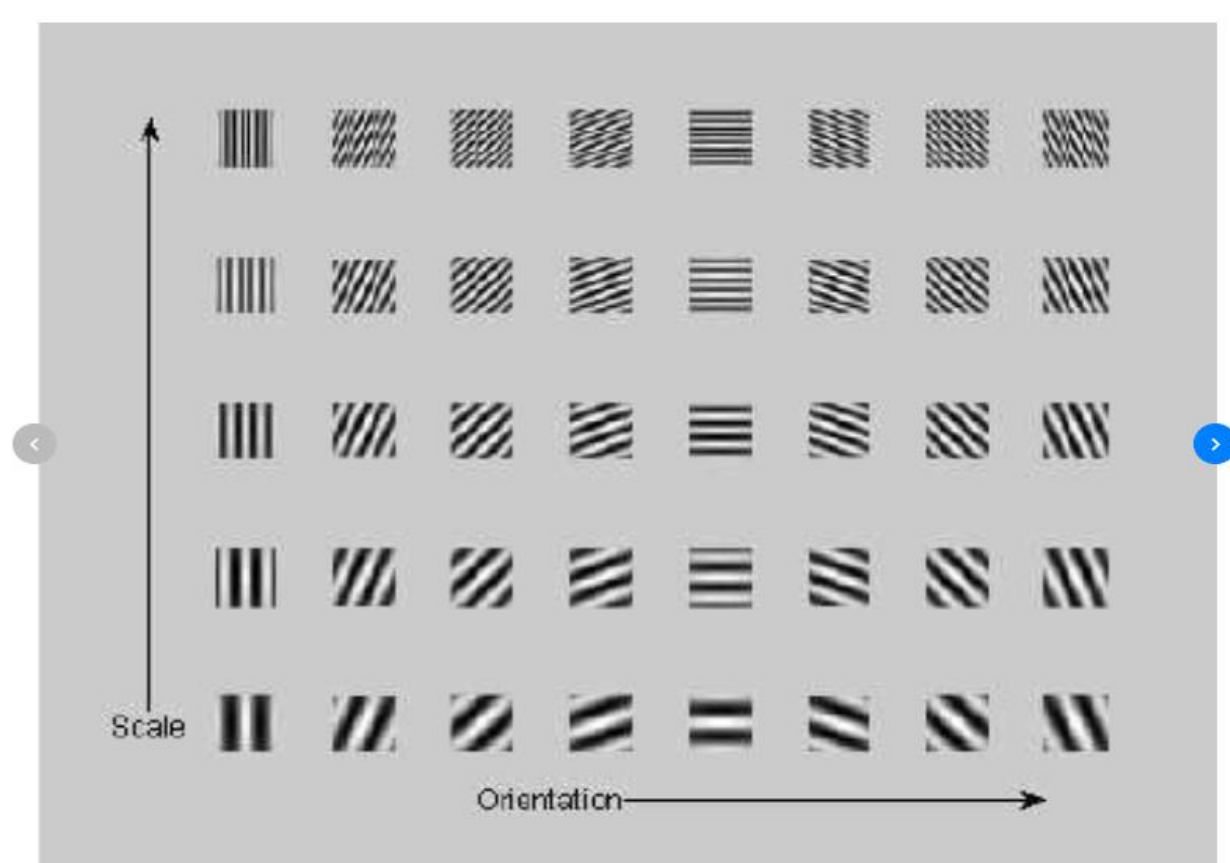
영상을 위한 컨볼루션 필터 설계의 어려움

- 영상에 대한 기하학적 변화에도 강인하도록 필터가 설계되어야 함

크기			
각도			
밝기			
위치			

영상을 위한 컨볼루션 필터 설계의 문제점

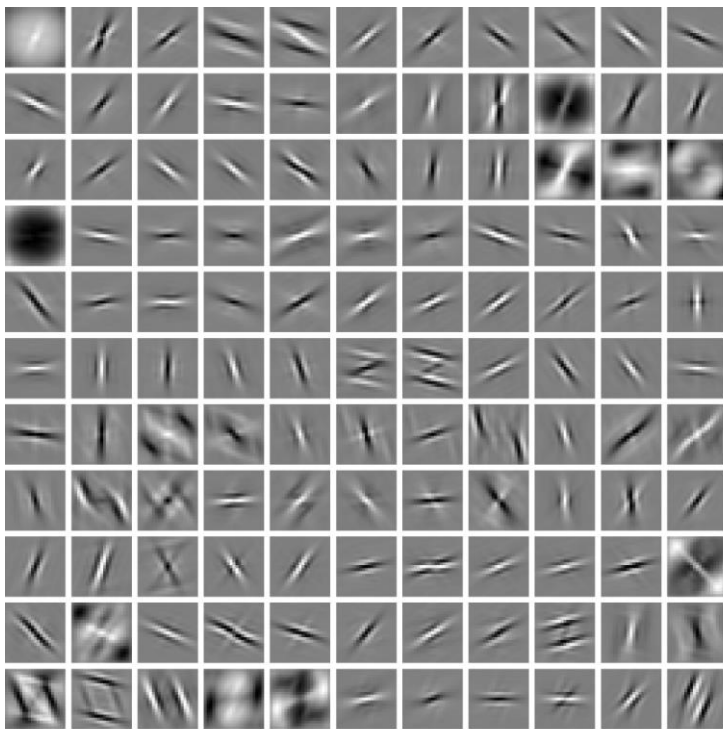
- 필터 수가 많아지면 차원 수가 늘어나 계산 복잡도가 급격히 높아짐



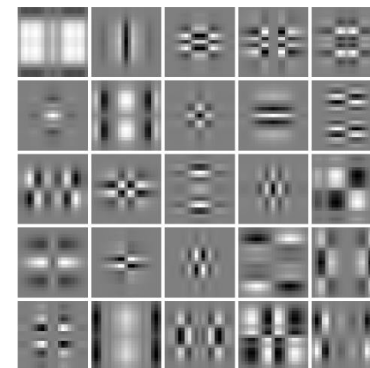
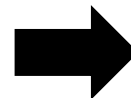
Schematic representation of the Gabor filter banks in 5 scales and 8 orientations

영상을 위한 컨볼루션 필터 설계의 문제점

- 컨볼루션 신경망 : 학습을 통해 필터를 설계



Full- rank filters

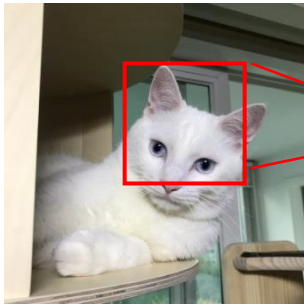


Approximating
Separable filters

R. Rigamonti; V. Lepetit; P. Fua : **Learning Separable Filters**. 2012.

컨볼루션 신경망이란

- 컨볼루션 신경망 : 물체의 방향과 크기가 바뀌어도 어려움 없이 물체를 인식할 수 있다는 생물학적 근거에 기반
- 이미지로부터 직접 공간적 특징을 학습



2D 신경망



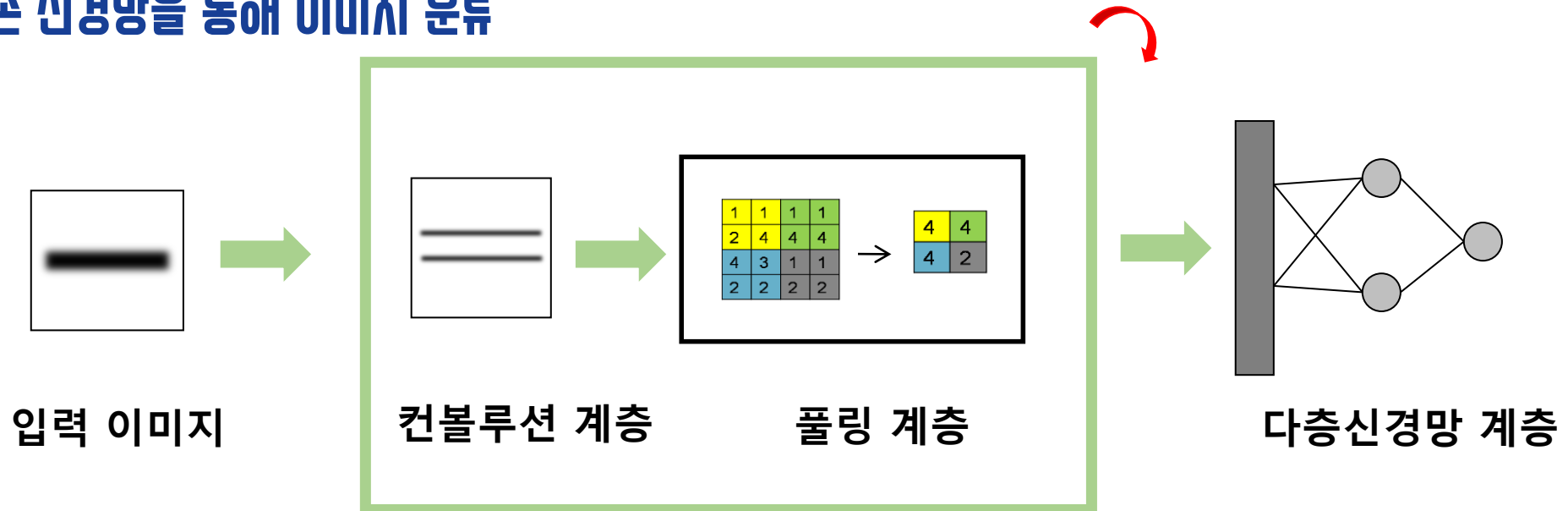
필터 학습



신경망 기반 분류

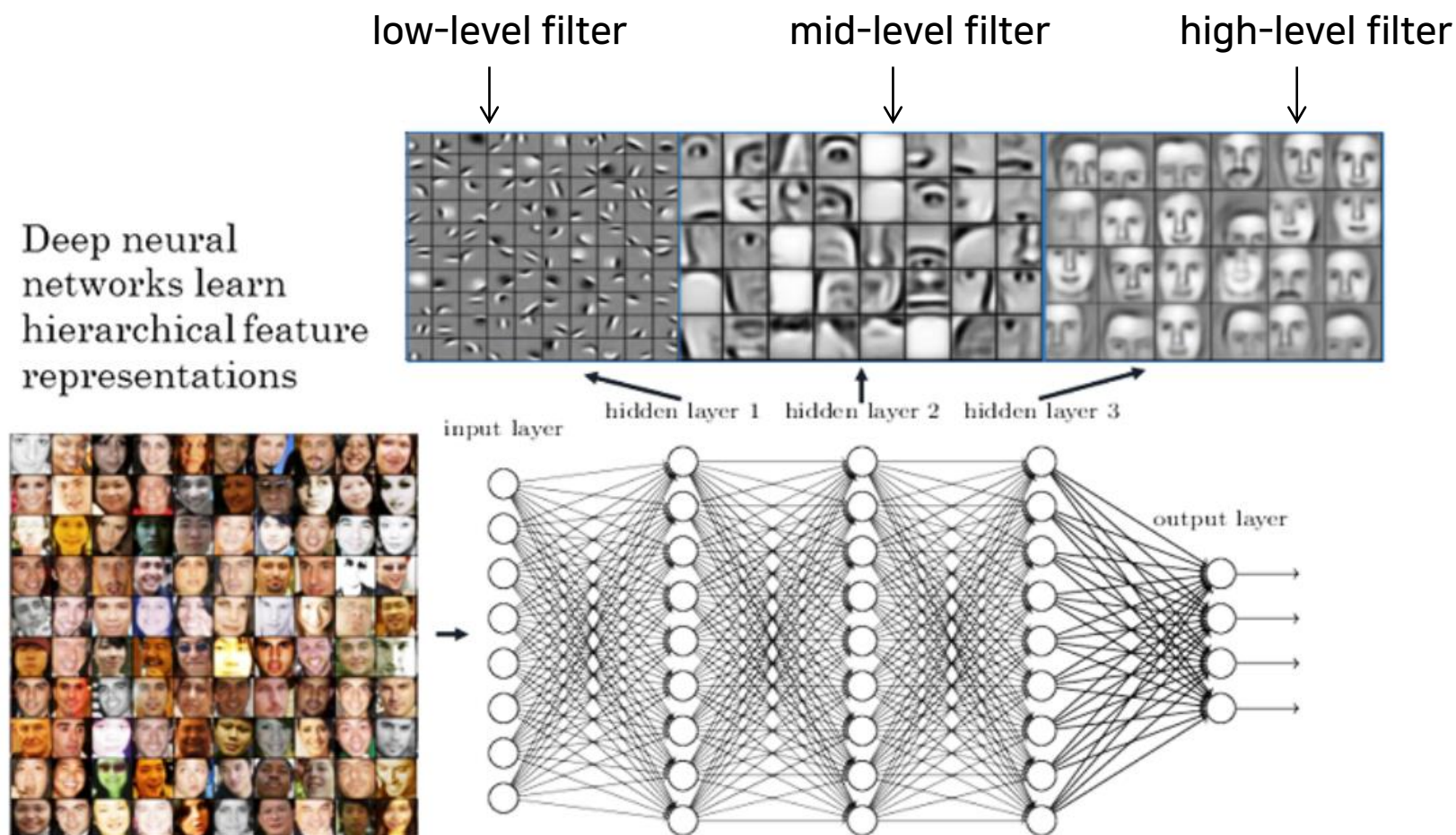
컨볼루션 신경망 구조

- 컨볼루션 계층 : 이미지 필터링
- 풀링 계층 : 이미지 크기 축소
- 기존 신경망을 통해 이미지 분류

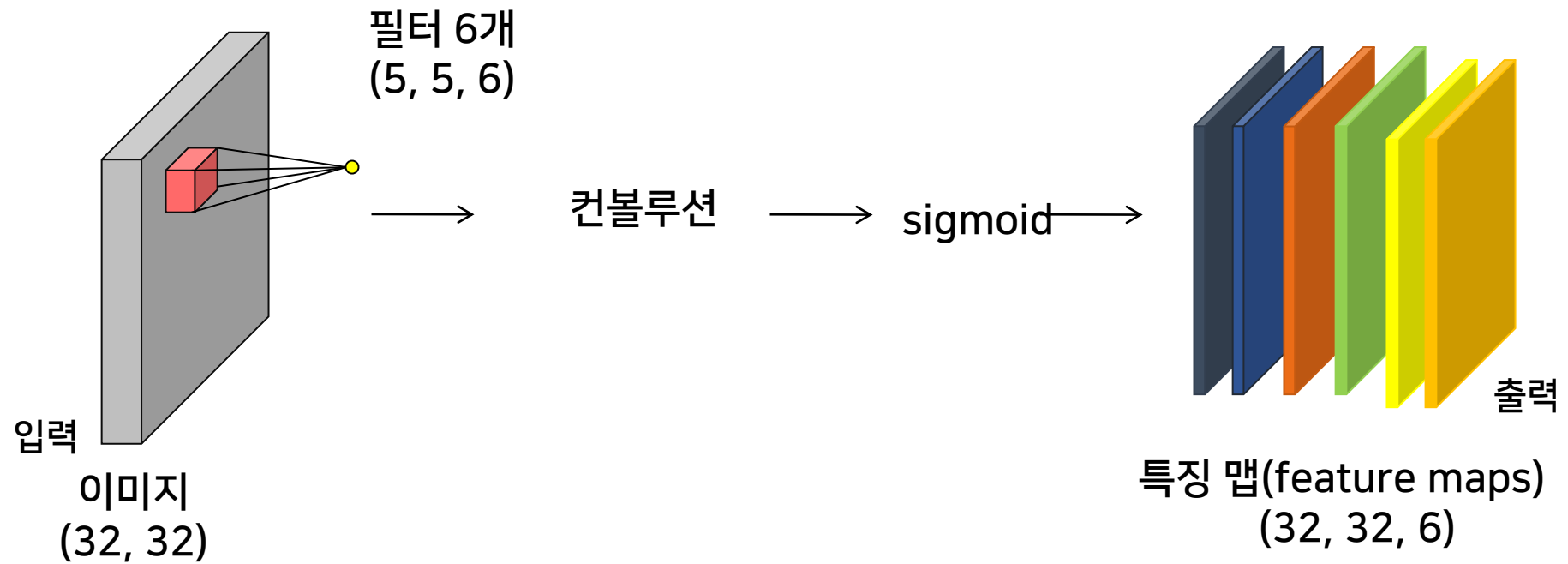


컨볼루션 신경망에서의 필터 학습

- 컨볼루션 신경망은 일반적으로 다층의 필터로 구성



컨볼루션 계층



필터 연산 시 외곽 픽셀 처리

- 가장자리 픽셀들은 이웃하는 픽셀 정보가 없기 때문에 컨볼루션 연산을 수행할 수 없음
- 컨볼루션 계층을 여러 번 거쳐도 영상의 크기를 유지할 수 있는 방법 필요

1	10	1	10	1
10	50	0	9	5
10	50	50	80	10
0	50	20	0	0
1	1	0	0	50

5X5 입력이미지



0	1	0
0	1	0
0	1	0

3X3 필터

$$\begin{aligned} &1 \times 0 + 10 \times 1 + 1 \times 0 + \\ &10 \times 0 + 50 \times 1 + 0 \times 0 + \\ &10 \times 0 + 50 \times 1 + 50 \times 0 \\ &= 110 \end{aligned}$$

110		

3X3 출력 영상

패딩을 이용한 외각 픽셀 처리

- 패딩(padding) : 데이터의 외부에 일정한 값의 계층을 덧대는 것
 - > 입력 영상의 가장자리에 0으로 padding 수행
- 패딩 수행 시 여러 번의 컨볼루션 계층을 사용해 신경망을 구성해도 영상 크기가 줄어들지 않음

padding 1

0	0	0	0
0	225	255	0
0	50	50	0
0	0	0	0

padding 2

0	0	0	0	0	0
0	0	0	0	0	0
0	0	225	255	0	0
0	0	50	50	0	0
0	0	0	0	0	0
0	0	0	0	0	0

padding 3

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	225	255	0	0	0
0	0	0	50	50	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

출력 영상 크기 = $(2p+n) \times (2p+n)$

스트라이드(Stride)

- 스트라이드 : 컨볼루션 연산 시 이동하는 x방향과 y방향의 이동량(보폭)
 > 일반적으로 1을 사용

Stride 1

1	10	1	10	1
10	50	0	9	5
10	50	50	80	10
0	50	20	0	0
1	1	0	0	50

$$1 \times 0 + 10 \times 1 + 1 \times 0 + 10 \times 0 + 50 \times 1 + 0 \times 0 + 10 \times 0 + 50 \times 1 + 50 \times 0 = 110$$

1	10	1	10	1
10	50	0	9	5
10	50	50	80	10
0	50	20	0	0
1	1	0	0	50

$$10 \times 0 + 1 \times 1 + 10 \times 0 + 50 \times 0 + 0 \times 1 + 9 \times 0 + 50 \times 0 + 50 \times 1 + 80 \times 0 = 51$$

110	51	

Stride 2

1	10	1	10	1
10	50	0	9	5
10	50	50	80	10
0	50	20	0	0
1	1	0	0	50

$$1 \times 0 + 10 \times 1 + 1 \times 0 + 10 \times 0 + 50 \times 1 + 0 \times 0 + 10 \times 0 + 50 \times 1 + 50 \times 0 = 110$$

1	10	1	10	1
10	50	0	9	5
10	50	50	80	10
0	50	20	0	0
1	1	0	0	50

$$1 \times 0 + 10 \times 1 + 1 \times 0 + 0 \times 0 + 9 \times 1 + 5 \times 0 + 50 \times 0 + 80 \times 1 + 10 \times 0 = 99$$

110	99

풀링 계층(Pooling Layer)

- 풀링 계층 : 입력 영상의 크기를 줄이는 역할 (대략적인 정보 특성만 남김)
 - > 영상의 크기를 반감해 계산량을 줄이고, 데이터에 치중된 과적합 방지



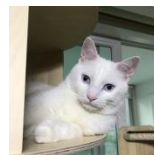
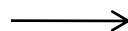
128 X 128

100개 필터 사용
 $128 \times 128 \times 100$
 $= 163840$



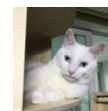
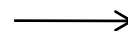
64 X 64

100개 필터 사용
 $64 \times 64 \times 100$
 $= 409600$



32 X 32

100개 필터 사용
 $32 \times 32 \times 100$
 $= 102400$

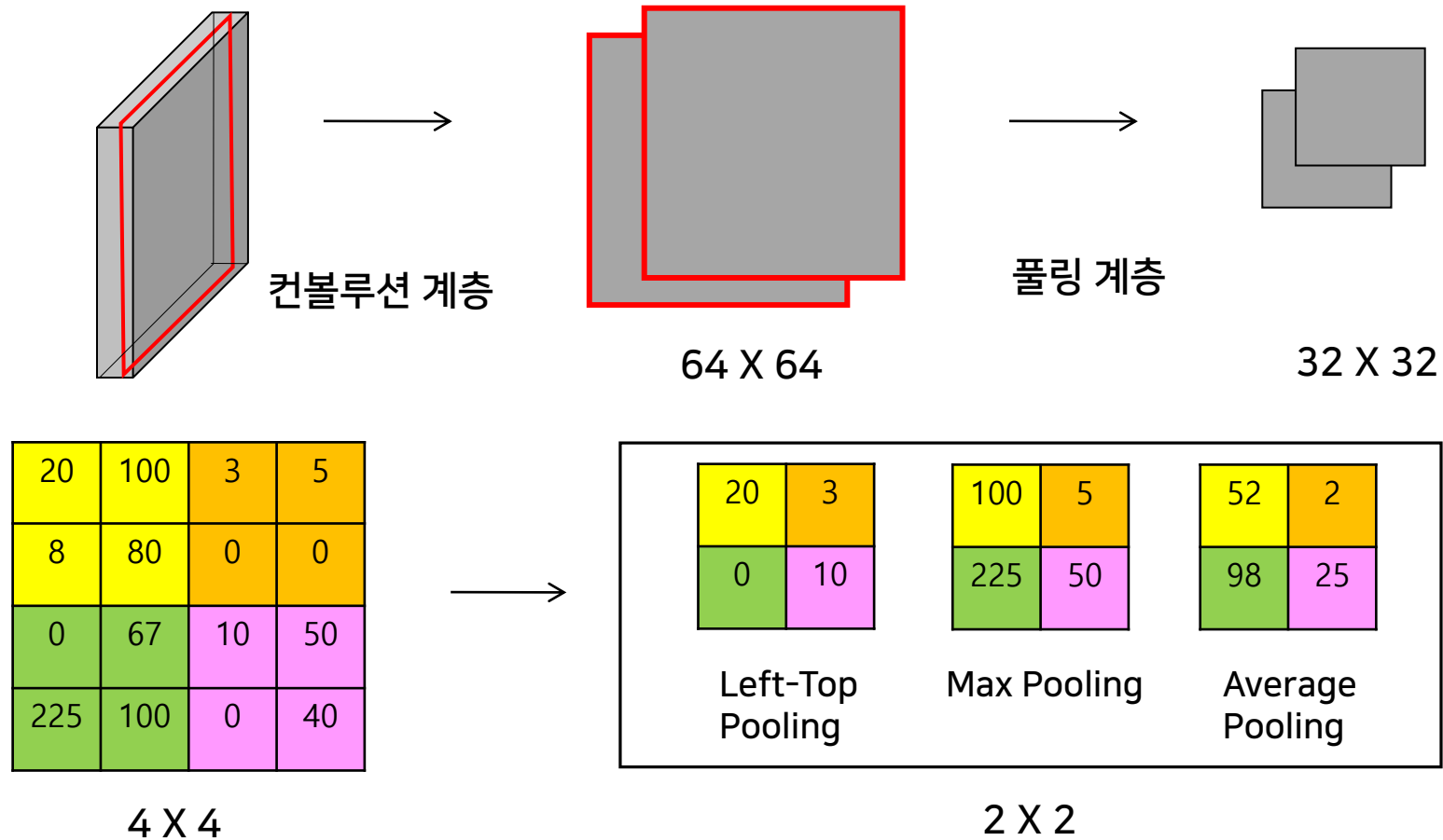


16 X 16

100개 필터 사용
 $16 \times 16 \times 100$
 $= 25600$

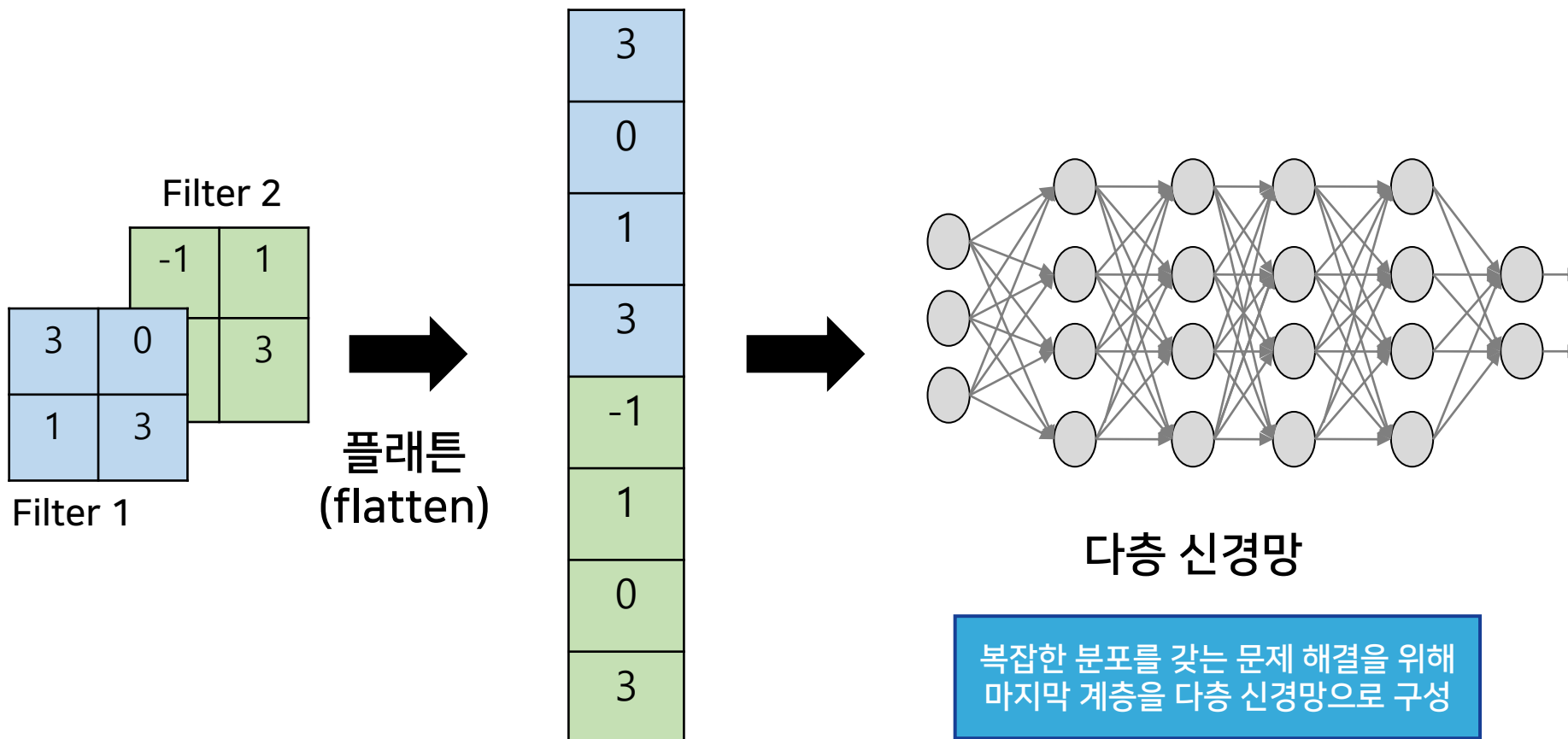
최대 풀링(Max Pooling)

- 영상의 크기를 줄이는 여러 방법 중 일반적으로 최대 풀링을 활용함



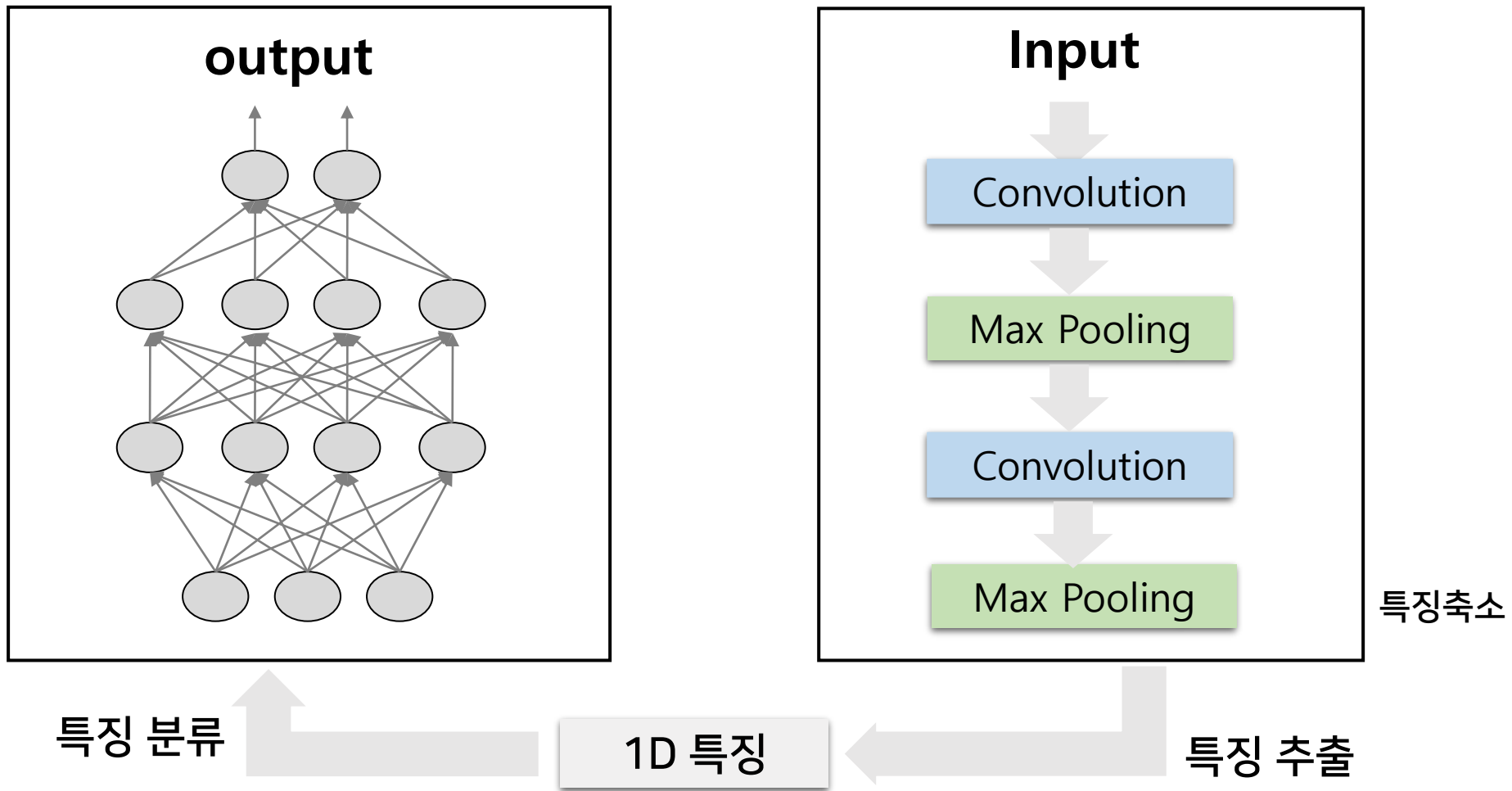
다층 신경망 계층

- 플래튼(Flatten) : 이전 계층의 2차원 형태의 데이터를 1차원 형태로 변환



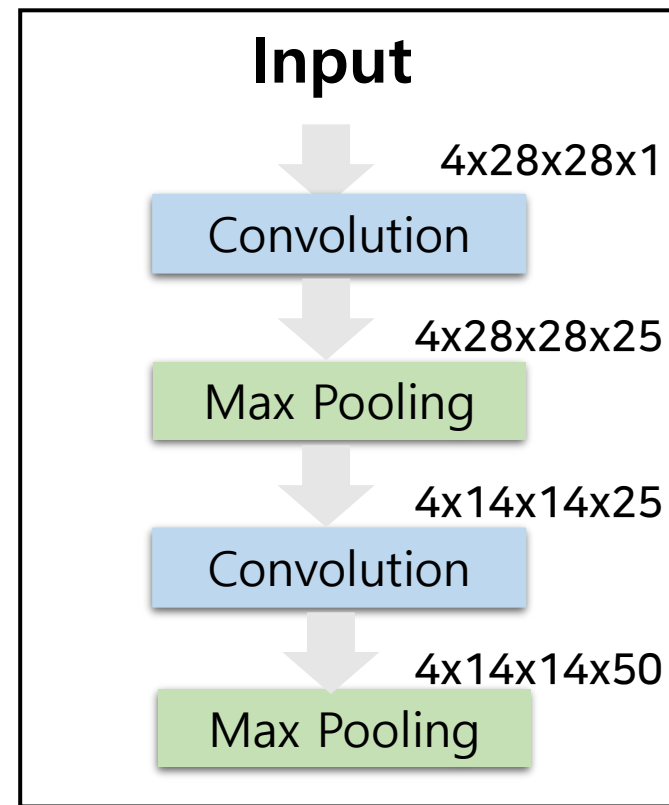
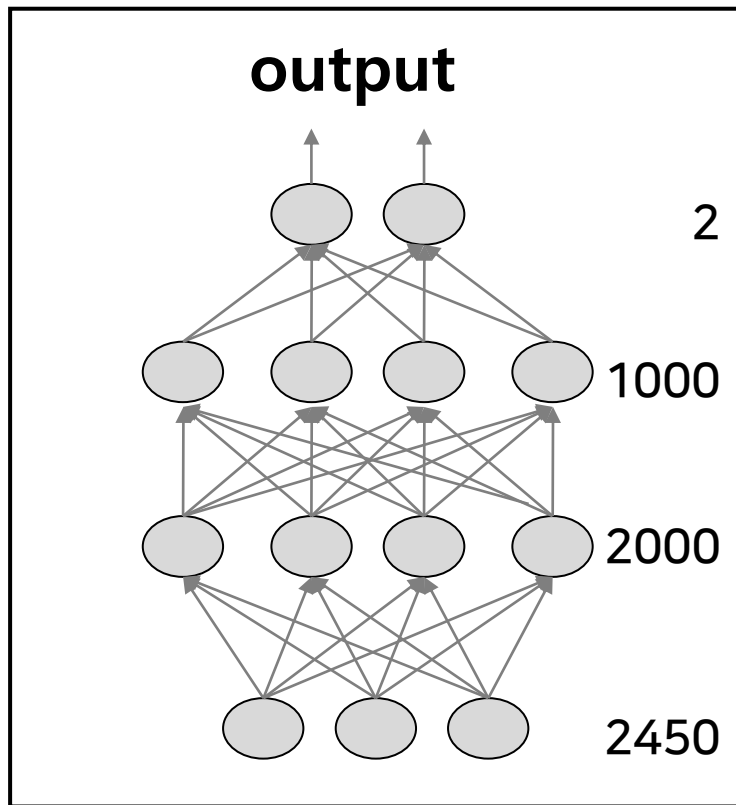
컨볼루션 신경망 정리

- 컨볼루션 계층과 Max Pooling 계층으로 묶여진 과정은 반복될 수 있음



컨볼루션 신경망 정리

- 25개 및 50개의 필터 역시 동시에 역전파 오류법을 통해 학습



4 x 2450

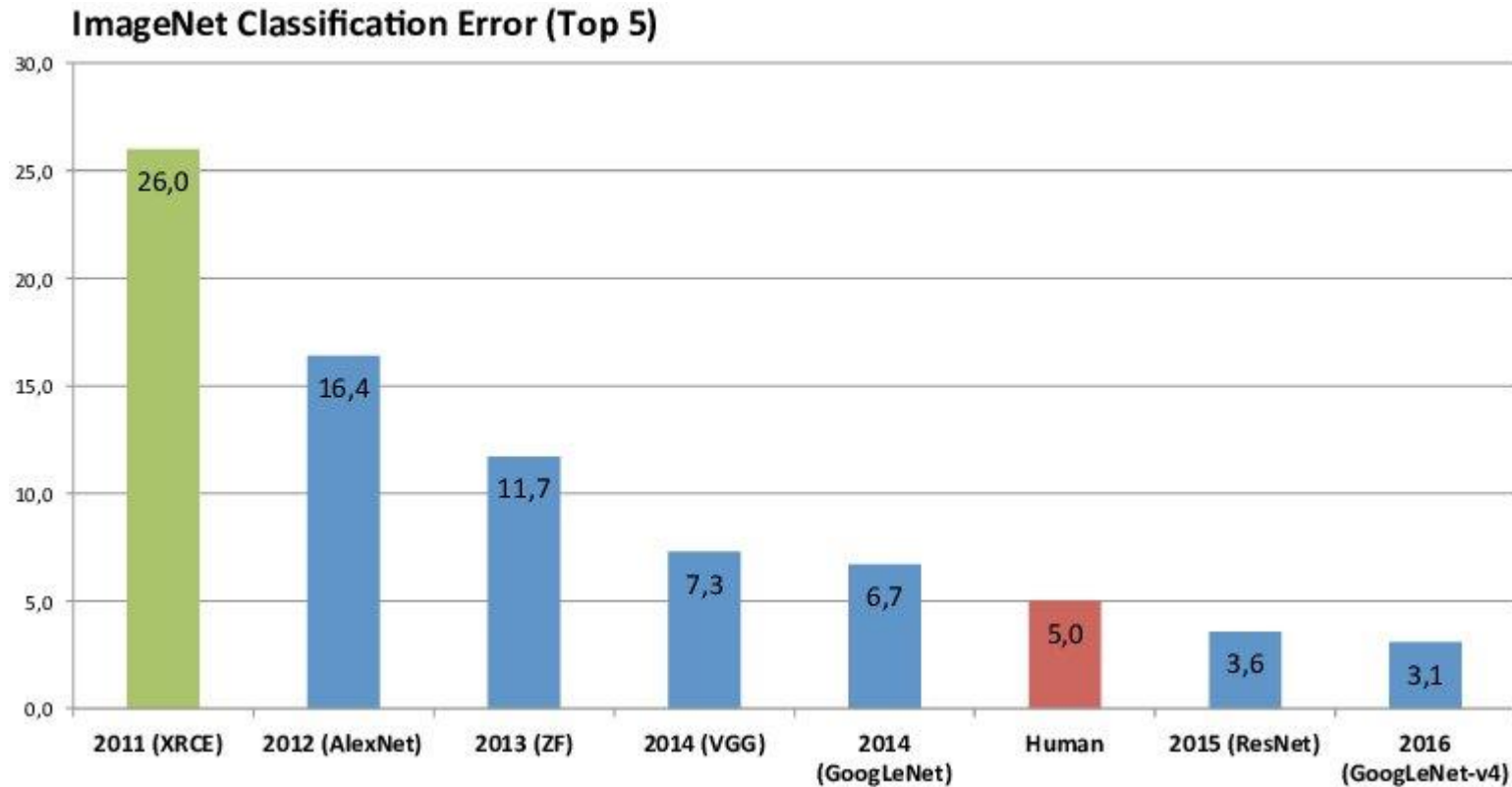
1D 특징

4 x 7 x 7 x 50

- 이미지에 레이블 정보가 부여되어 있어 특정 키워드로 정확하게 검색 가능

31

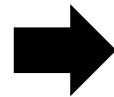
CNN 응용 분야 : ImageNet



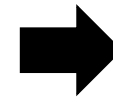
<https://www.researchgate.net/figure/Winner-results-of-the-ImageNet-large-scale-visual-recognition-challenge-LSVRC-of-the-fig7-324476862>

AlphaGo

- 주변 바둑돌의 2차원 행렬 데이터 구성 → 현 시점에서의 검은 돌, 흰 돌의 정보를 파악



Neural
Network



Next move
(19 x 19 positions)

19 x 19 matrix

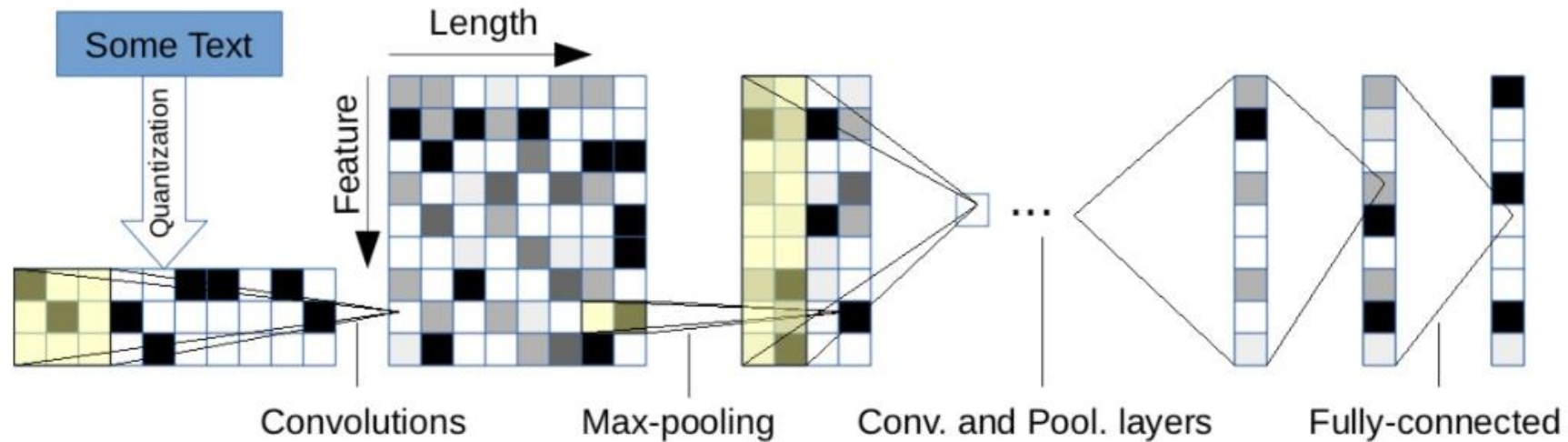
Black: 1

white: -1

none: 0

CNN in text classification

- 특정 단어들의 지역적 연관성을 통해 어떤 단어들이 서로 연관되어 있는지를 파악



출처 : Character-based Convolutional Neural Network for Text Classification
Taken from "Character-level Convolutional Networks for Text Classification", 2015