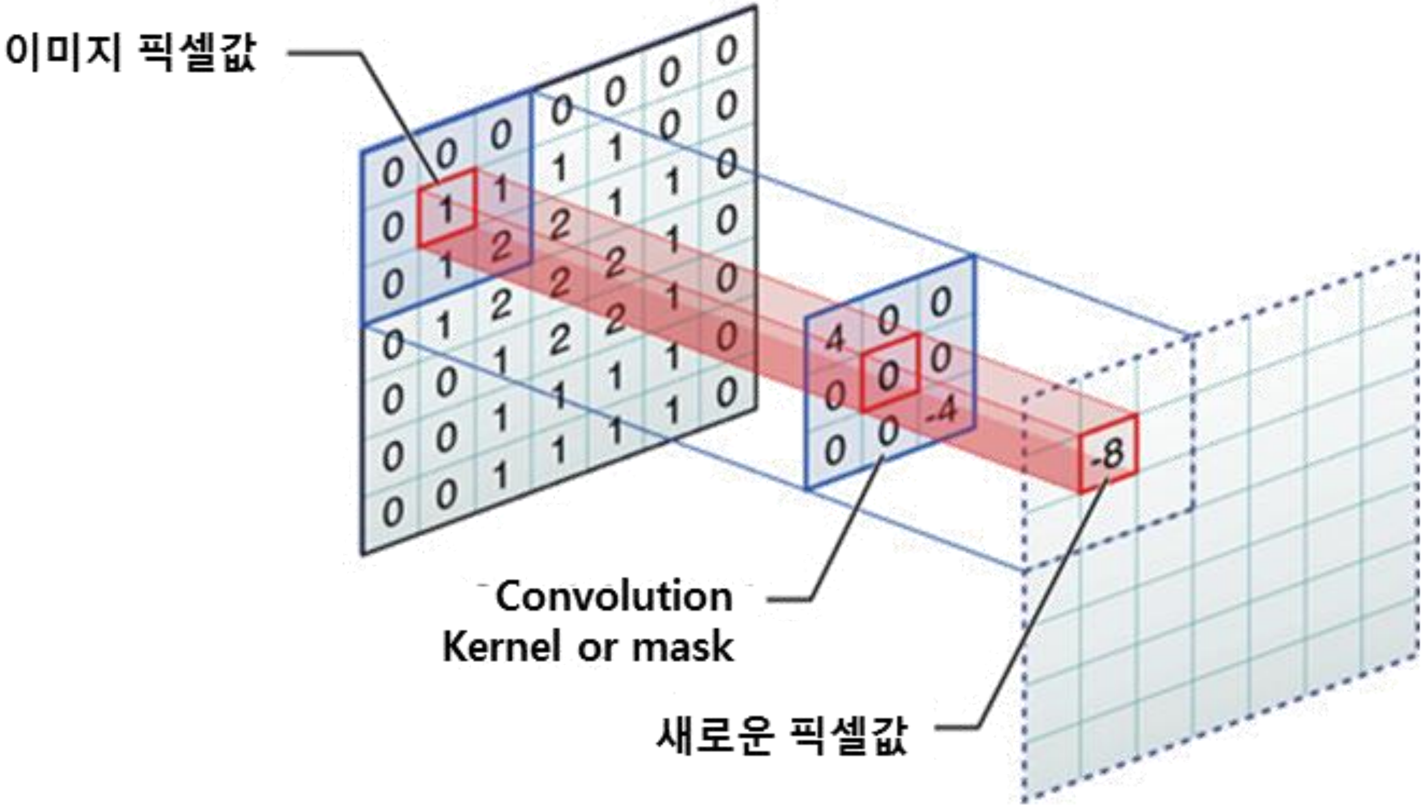
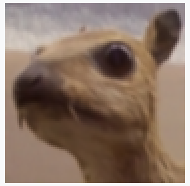
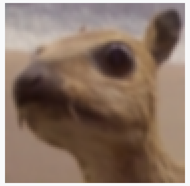
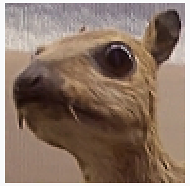
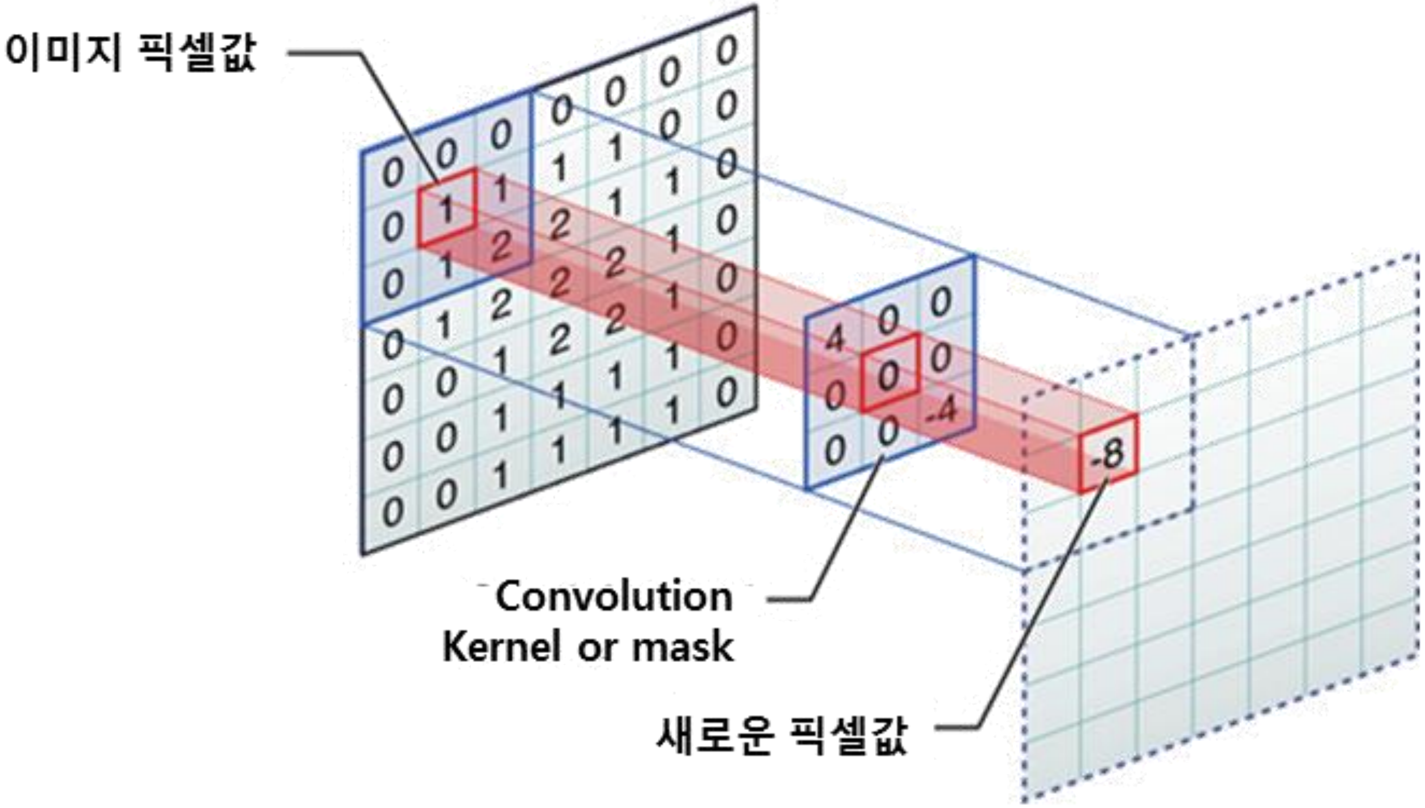
Convolution and Pooling

Convolution and Pooling

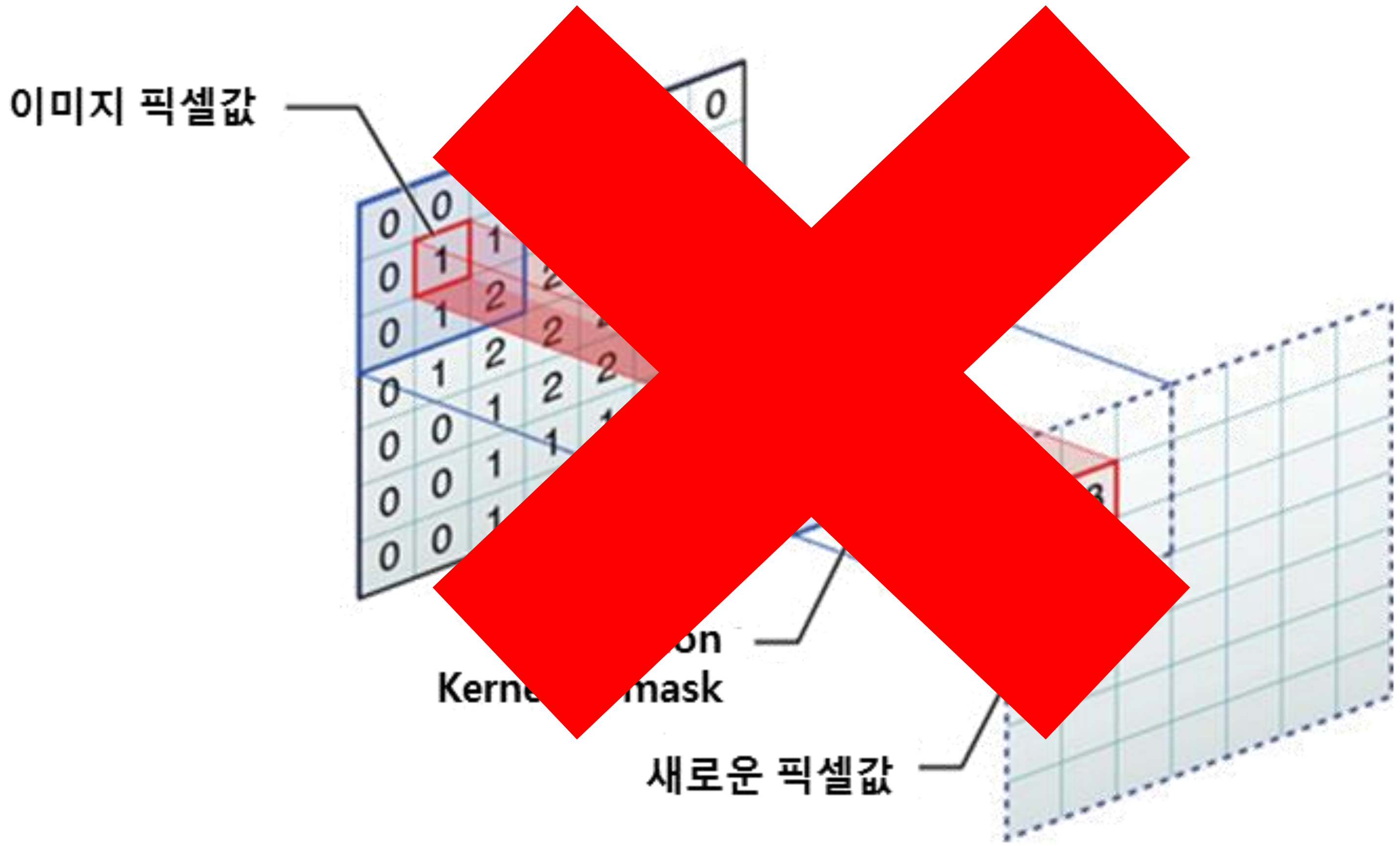
- Convolution
- Channel
- Stride
- Padding
- Pooling


이미지 픽셀값
Convolution
Kernel or mask
새로운 픽셀값

Operation	Kernel	Image result
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur 3 × 3 (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	
Gaussian blur 5 × 5 (approximation)	$\frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	
Unsharp masking 5 × 5 Based on Gaussian blur with amount as 1 and threshold as 0 (with no image mask)	$\frac{-1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & -476 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$	



이미지 픽셀값
Convolution
Kernel or mask
새로운 픽셀값


이미지 픽셀값
Kerne mask
새로운 픽셀값

$$
\begin{array}{c}\text{image}\\ \begin{array}{|c|c|c|}\hline 1 & 2 & 3\\ \hline 4 & 5 & 6\\ \hline 7 & 8 & 9\\ \hline\end{array}\end{array}
\quad \otimes \quad
\begin{array}{c}\text{kernel}\\ \begin{array}{|c|c|c|}\hline A & B & C\\ \hline D & E & F\\ \hline G & H & I\\ \hline\end{array}\end{array}
$$

$$(1 * A) + (2 * B) + (3 * C) + (4 * D) + \ldots + (9 * I)$$

image

1	2	3
4	5	6
7	8	9

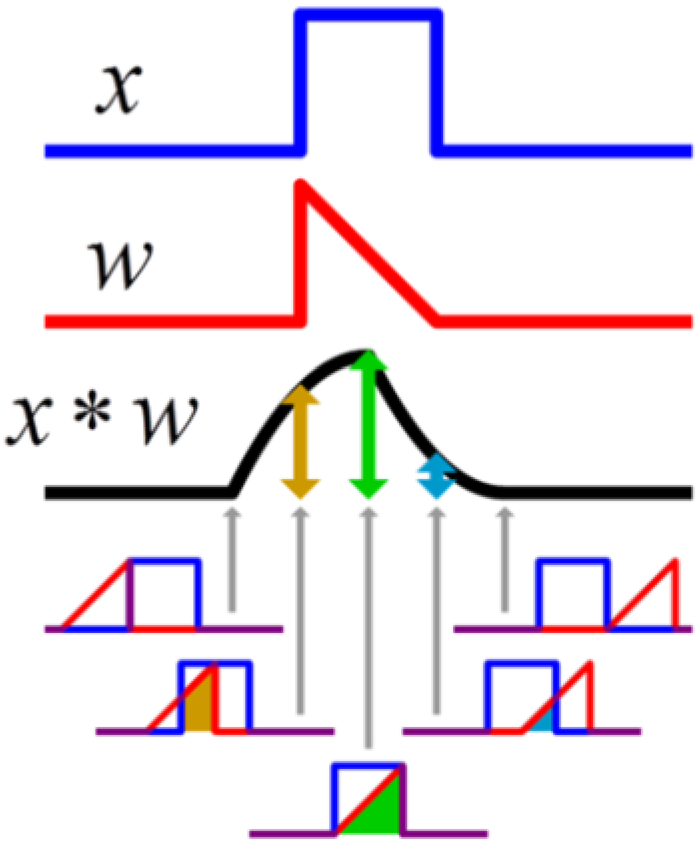
*

kernel

A	B	C
D	E	F
G	H	I

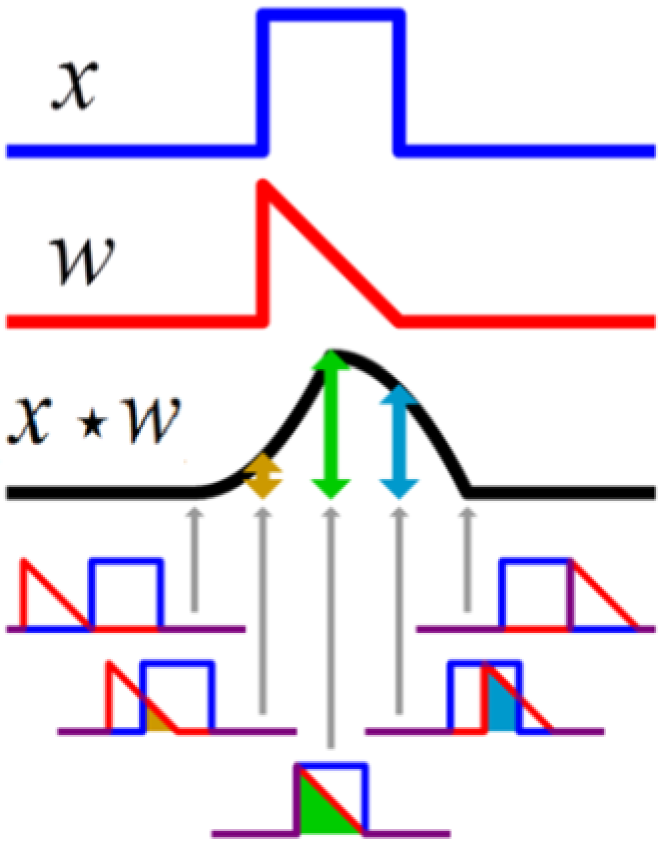
$$(1 * I) + (2 * H) + (3 * G) + (4 * F) + \ldots + (9 * A)$$

Convolution



$$x * w$$

Cross-Correlation



$$x \otimes w$$

합성곱

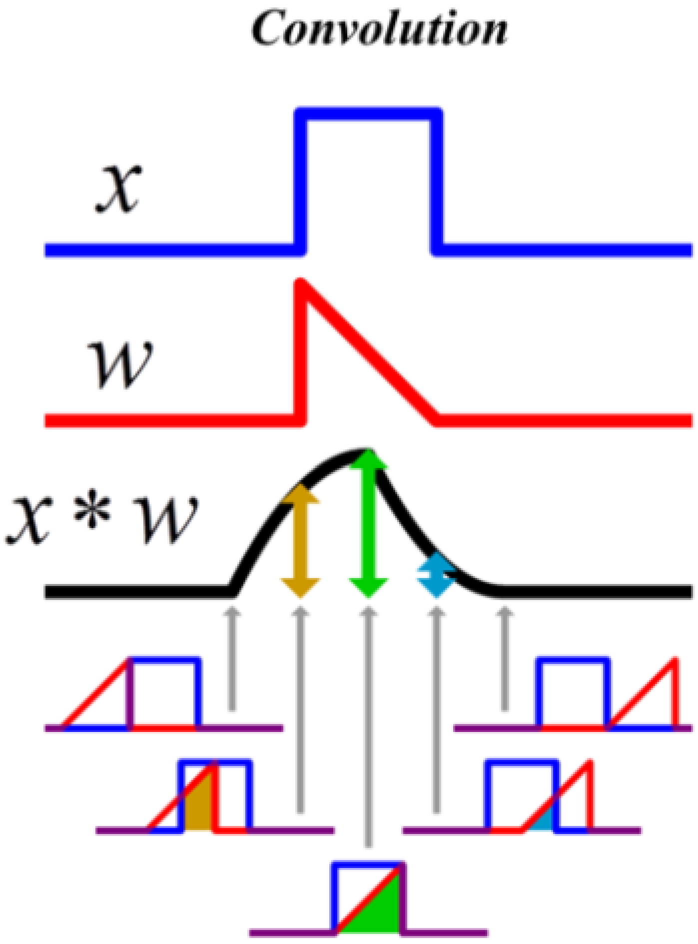
위키백과, 우리 모두의 백과사전.

합성곱(合成-, convolution, 콘벌루션)은 하나의 함수와 또 다른 함수를 반전 이동한 값을 곱한 다음, 구간에 대해 적분하여 새로운 함수를 구하는 수학 연산자이다.

$$y(t) = (x * w)(t)$$

$$= \int_{-\infty}^{\infty} x(a)w(t-a)da$$

$$y[n] = \sum_{a=-\infty}^{\infty} x[a]w[n-a]$$



$$y(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} image[m,n]\ kernel[i-m, j-n]$$

image

1	2	3
4	5	6
7	8	9

*

kernel

A	B	C
D	E	F
G	H	I

(1 * I) + (2 * H) + (3 * G) + (4 * F) + ... + (9 * A)

2-D Convolution

$$y(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} image[m,n]\ kernel[i-m, j-n]$$

image

1	2	3
4	5	6
7	8	9

*

kernel

A	B	C
D	E	F
G	H	I

-1,-1	0,-1	1,-1
-1,0	0,0	1,0
-1,1	0,1	1,1

*

i-1, j-1	i, j-1	i+1, j-1
i-1, j	i, j	i+1, j
i-1, j	i, j+1	i+1, j+1

Ex) m = -1, n = -1

-1,-1	0,-1	1,-1
-1,0	0,0	1,0
-1,1	0,1	1,1

i-1, j-1	i, j-1	i+1, j-1
i-1, j	i, j	i+1, j
i-1, j	i, j+1	i+1, j+1

-1,-1	0,-1	1,-1
-1,0	0,0	1,0
-1,1	0,1	1,1

i-1, j-1	i, j-1	i+1, j-1
i-1, j	i, j	i+1, j
i-1, j	i, j+1	i+1, j+1

Ex) m = 1, n = -1

$$y(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} image[m,n]\ kernel[i+m, j+n]$$

image

1	2	3
4	5	6
7	8	9

$\otimes$

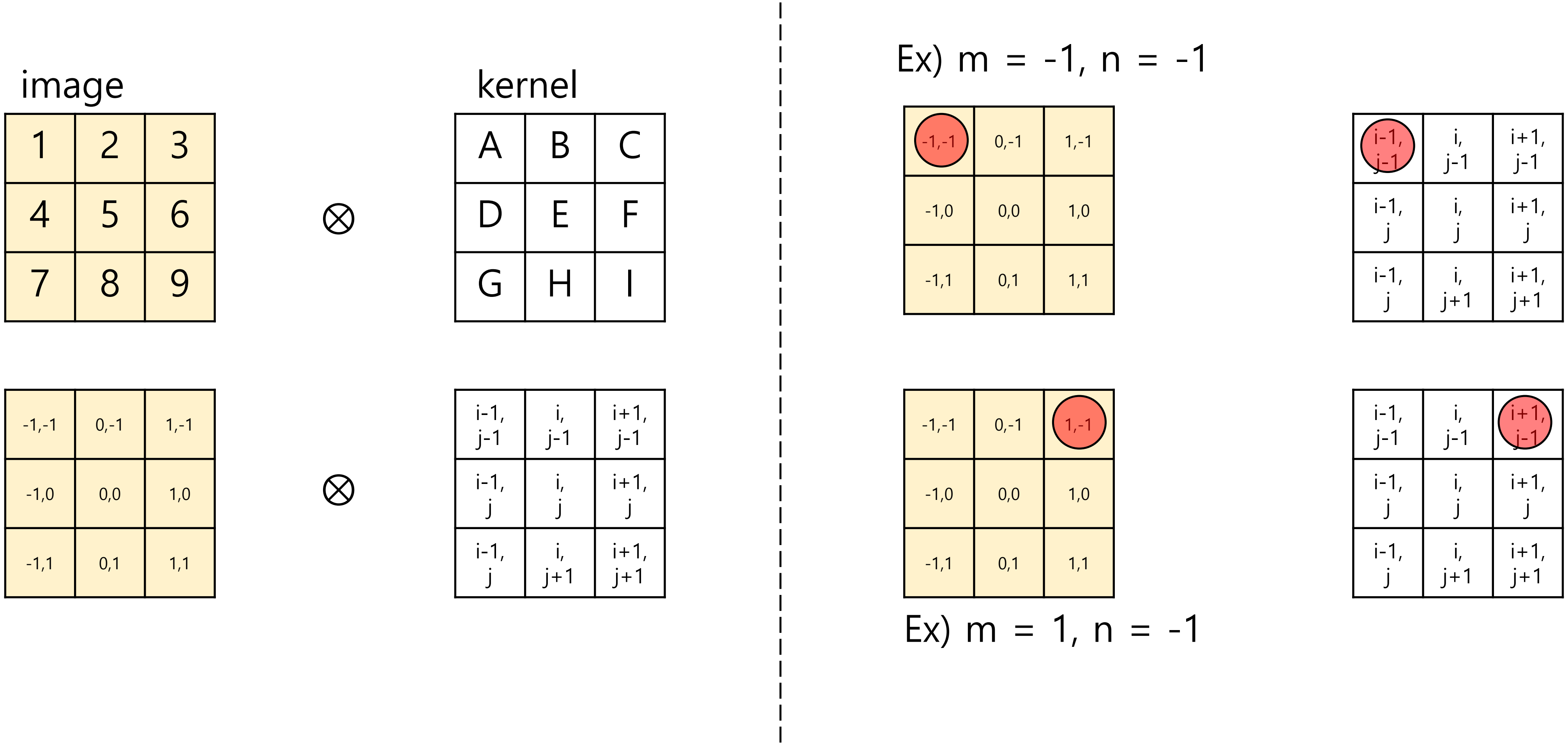
kernel

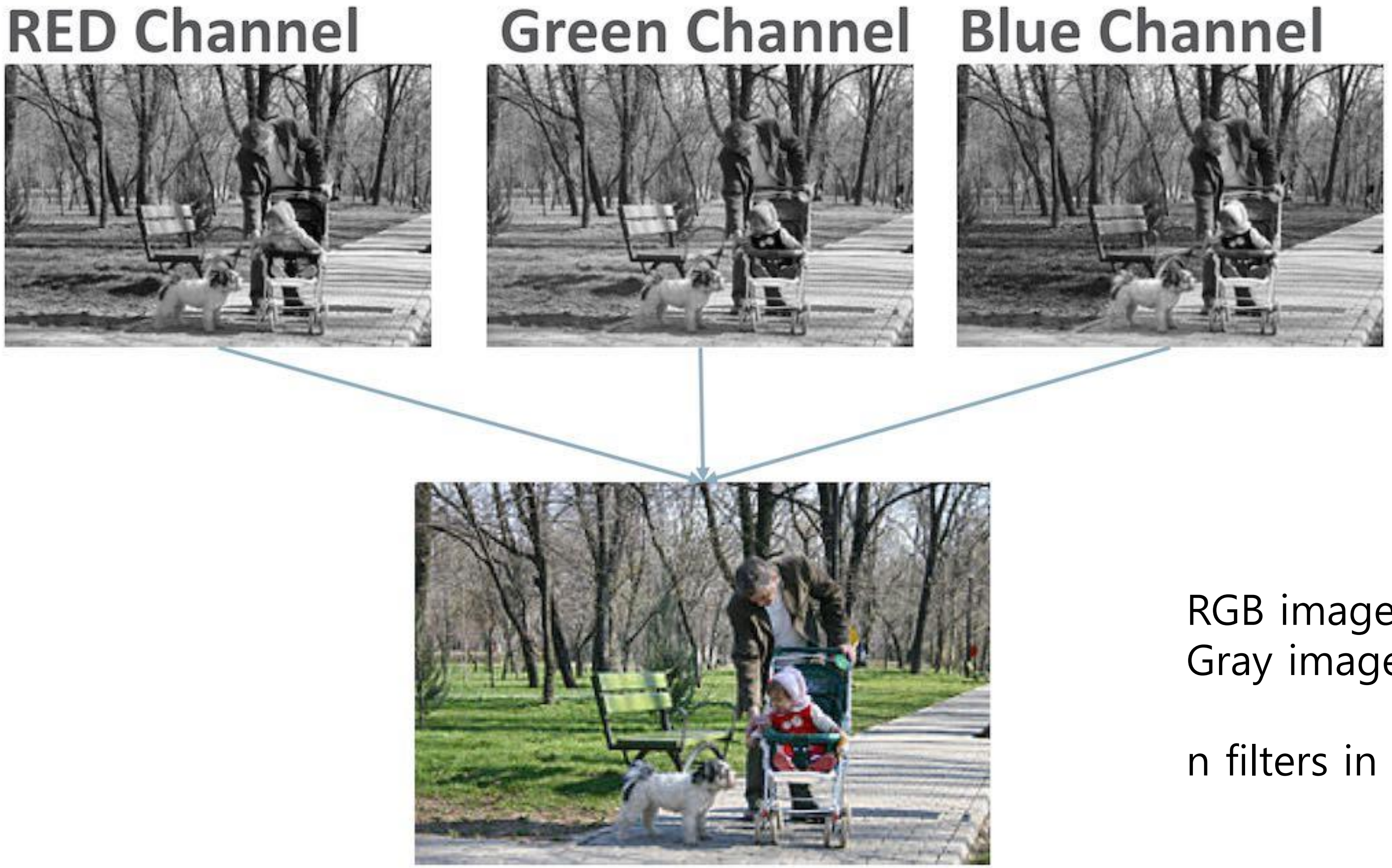
A	B	C
D	E	F
G	H	I

(1 * A) + (2 * B) + (3 * C) + (4 * D) + … + (9 * I)

2-D Cross-Correlation

$$y(i,j) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} image[m,n]\ kernel[i+m, j+n]$$

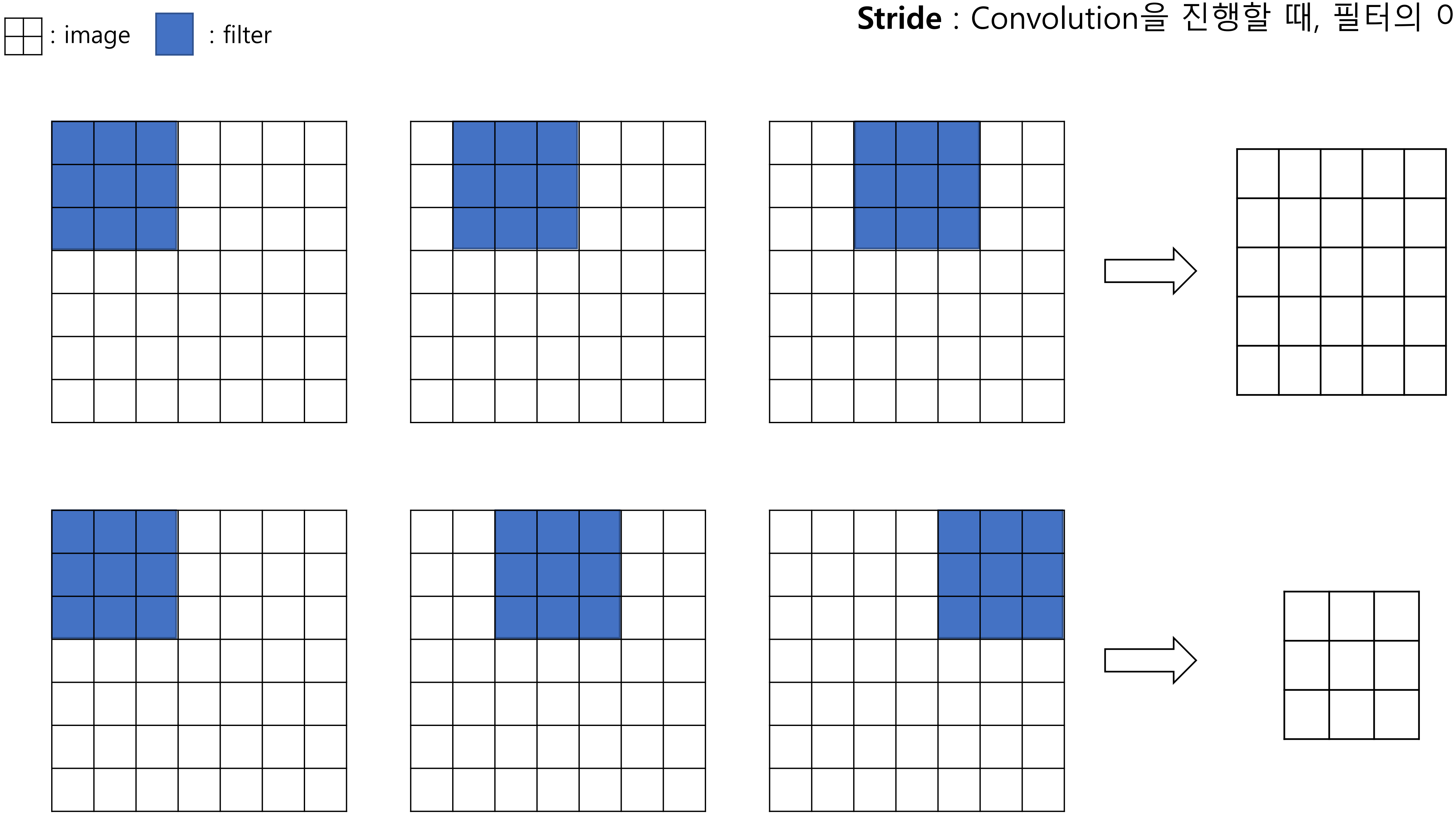




RGB image -> 3 channels
Gray image -> 1 channel

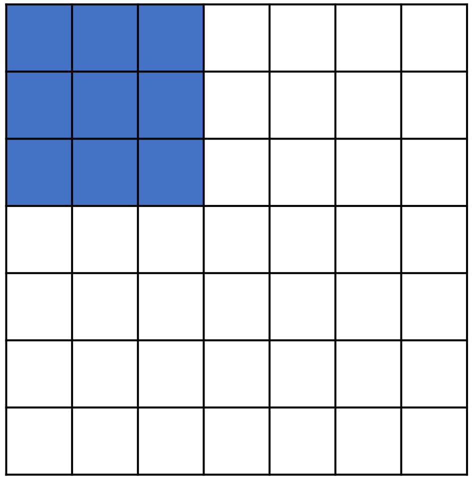
n filters in conv. layer -> n channels output

Stride : Convolution을 진행할 때, 필터의 이동 간격.



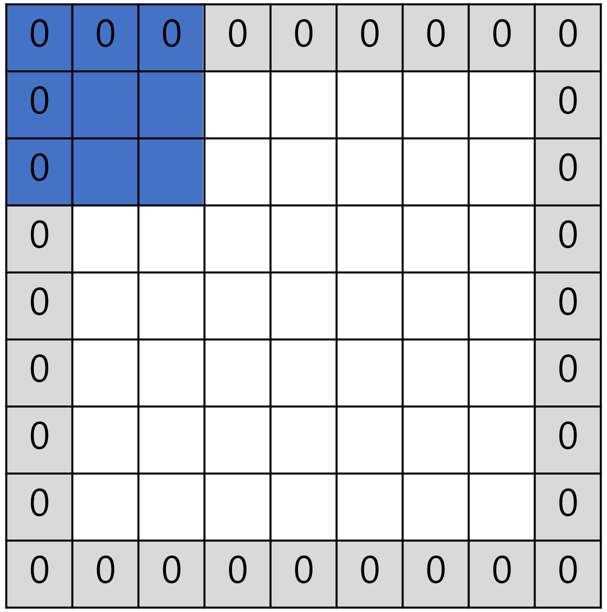
Padding : image의 모서리 부분 정보 손실 방지

Zero Padding = X

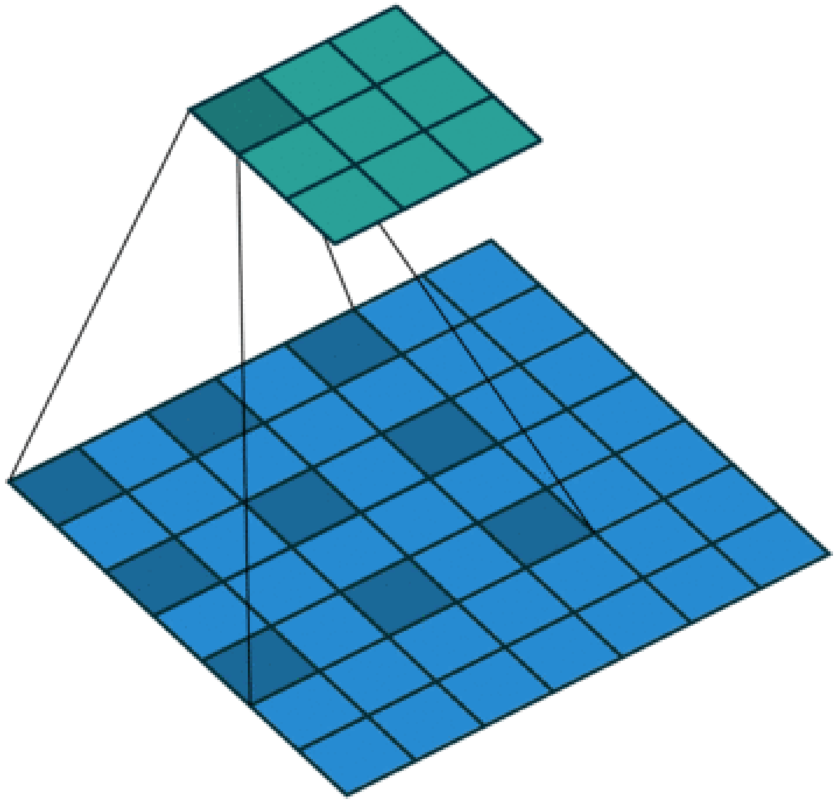


7*7 $\xrightarrow[\text{conv}]{}$ 5*5

Zero Padding = O



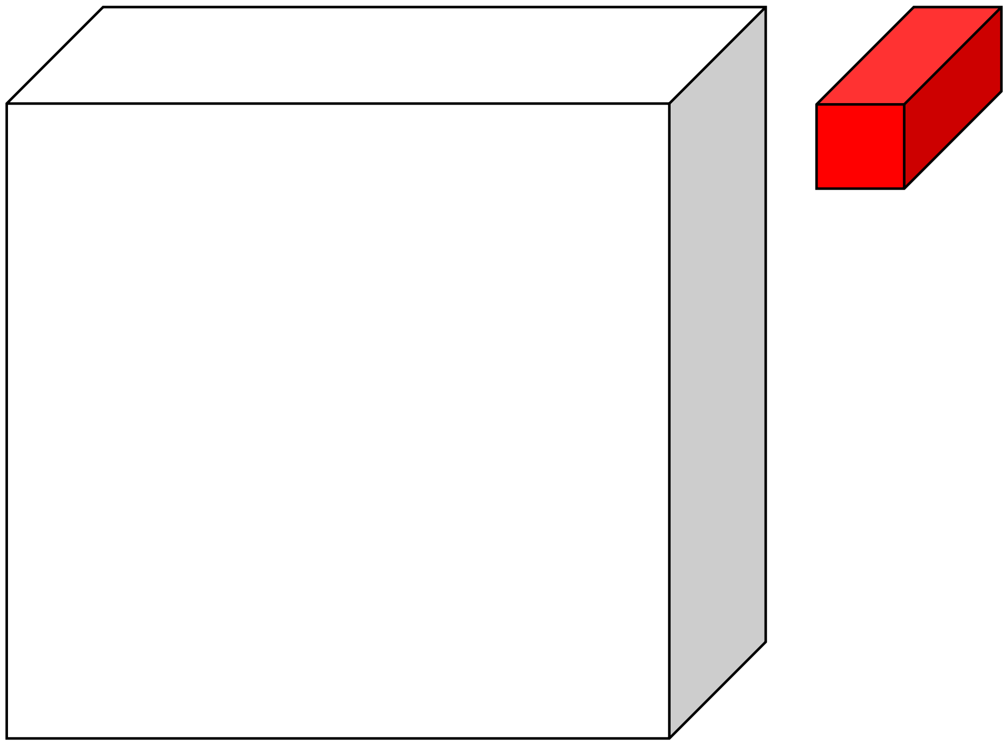
7*7 $\xrightarrow[\text{conv}]{}$ 7*7

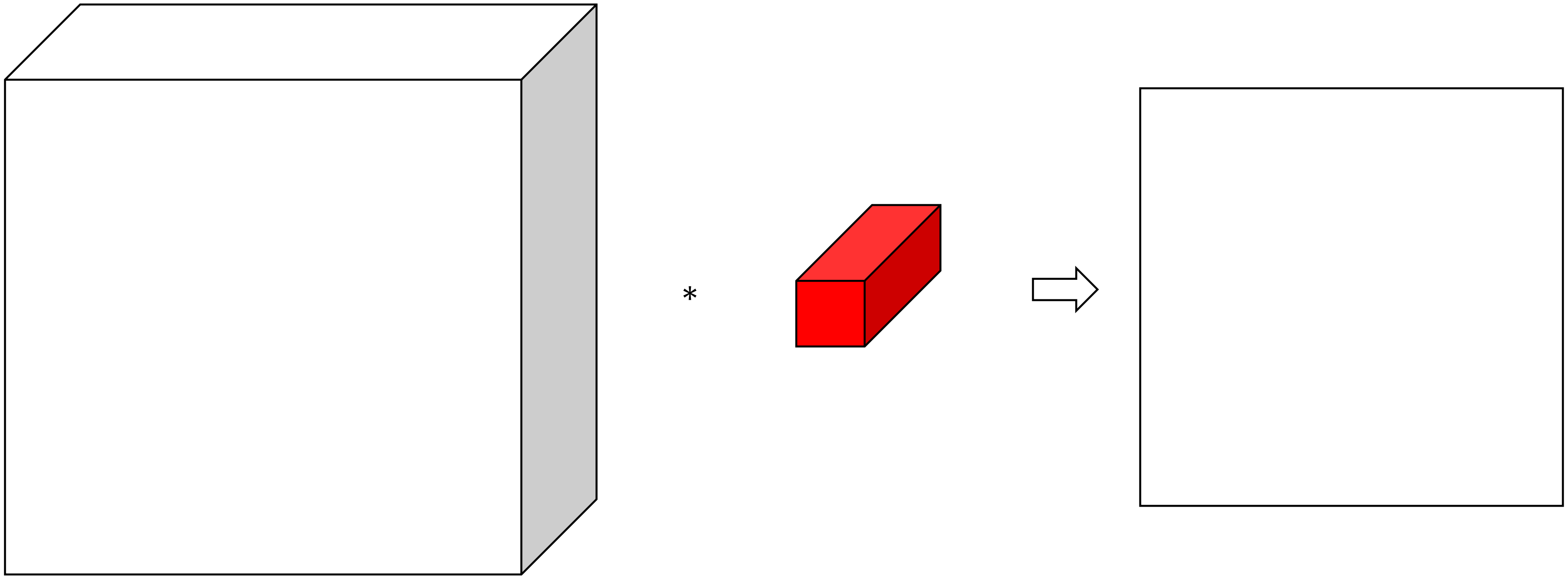


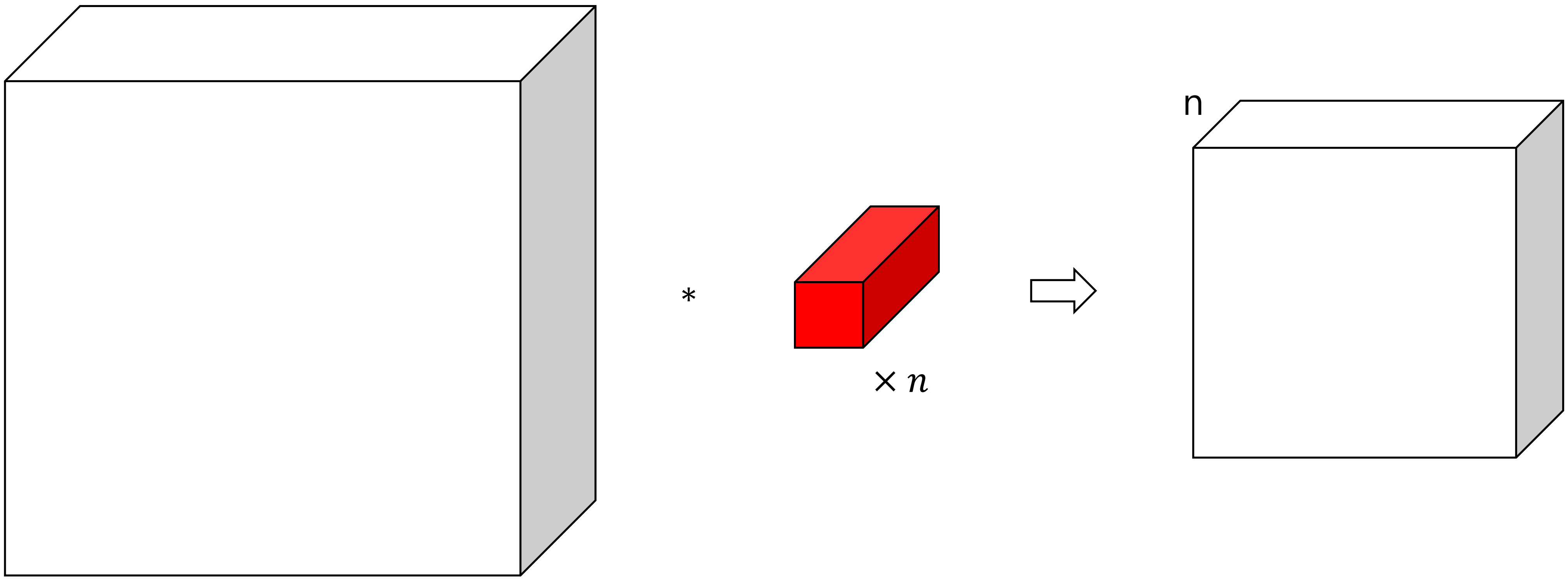
Dilated Convolutions

dilation rate : 커널 사이의 간격
ex) dilation rate = 2

넓은 시야가 필요할 때 용이
필터 내부에 zero padding을 추가
연산의 효율 좋다

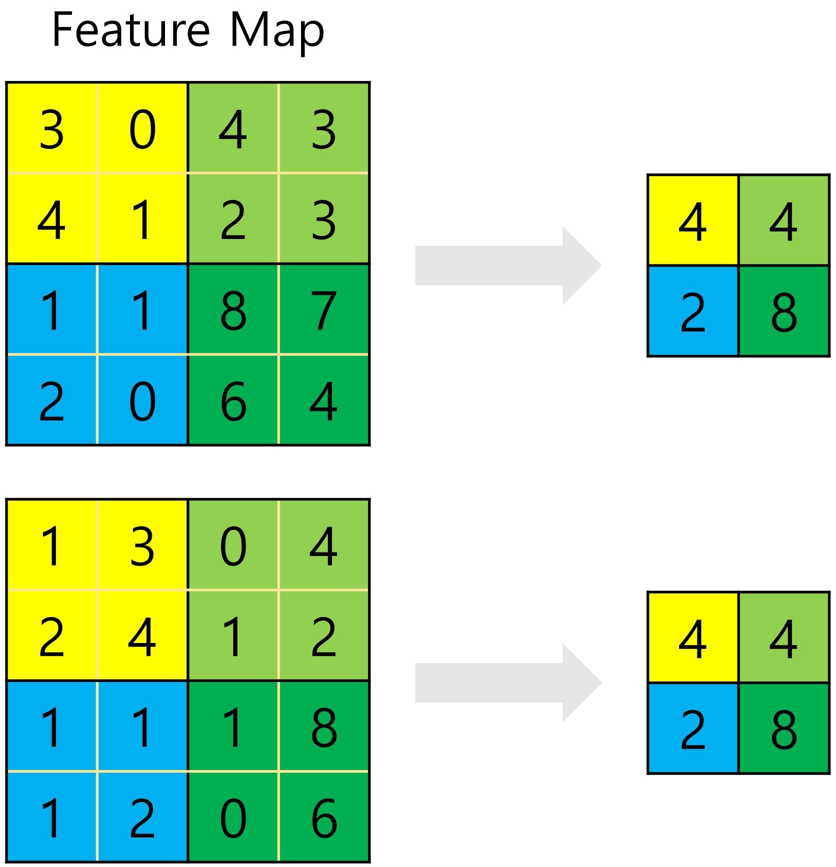




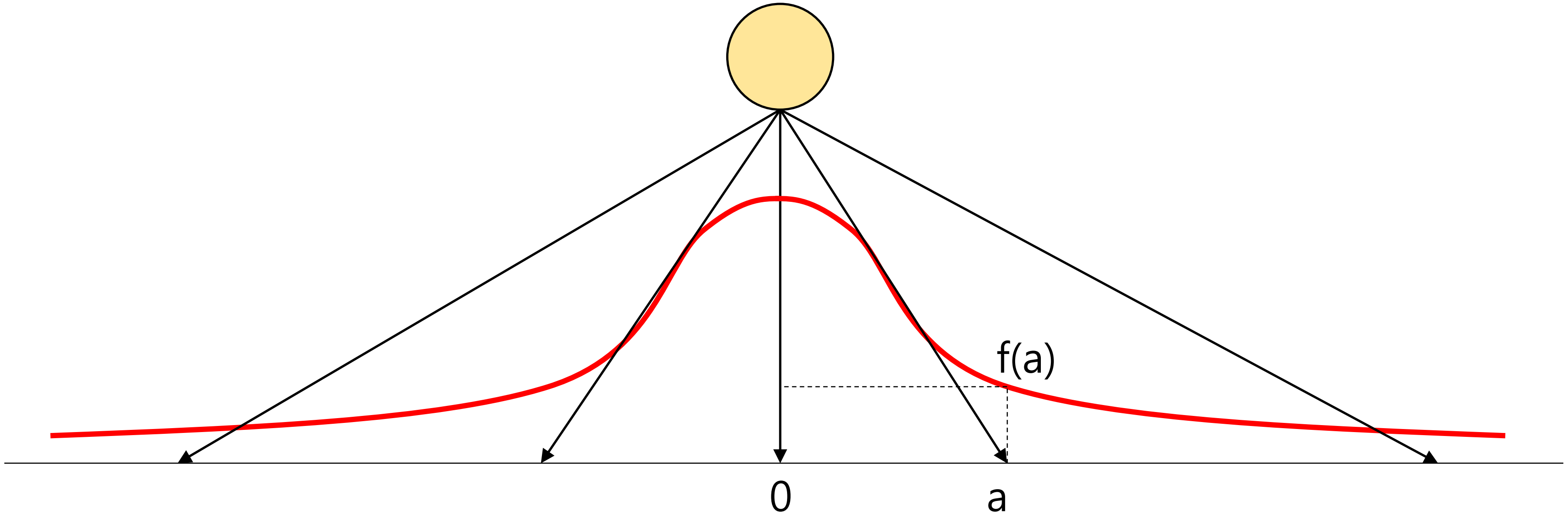

*
× n
n

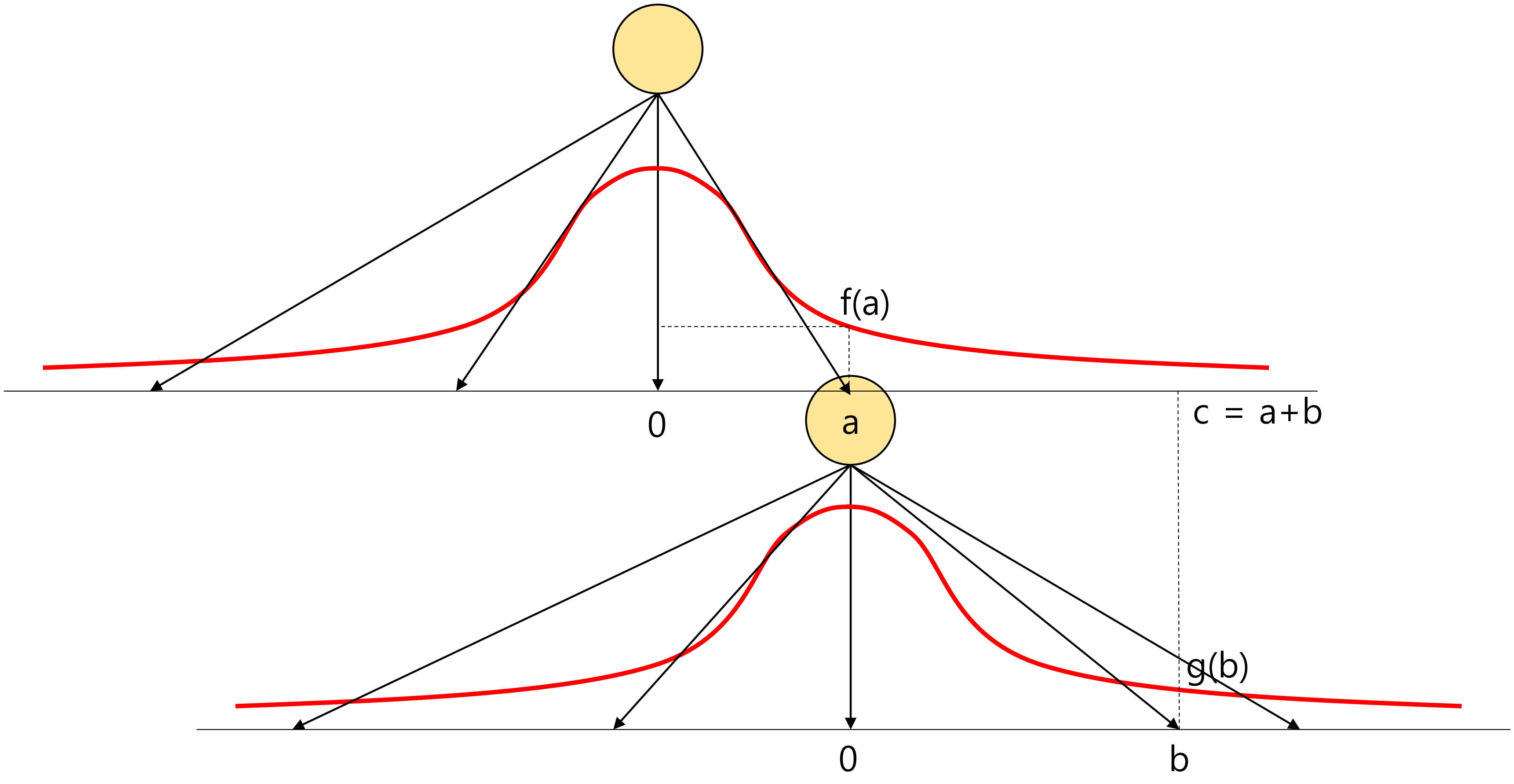
Sampling (Pooling)

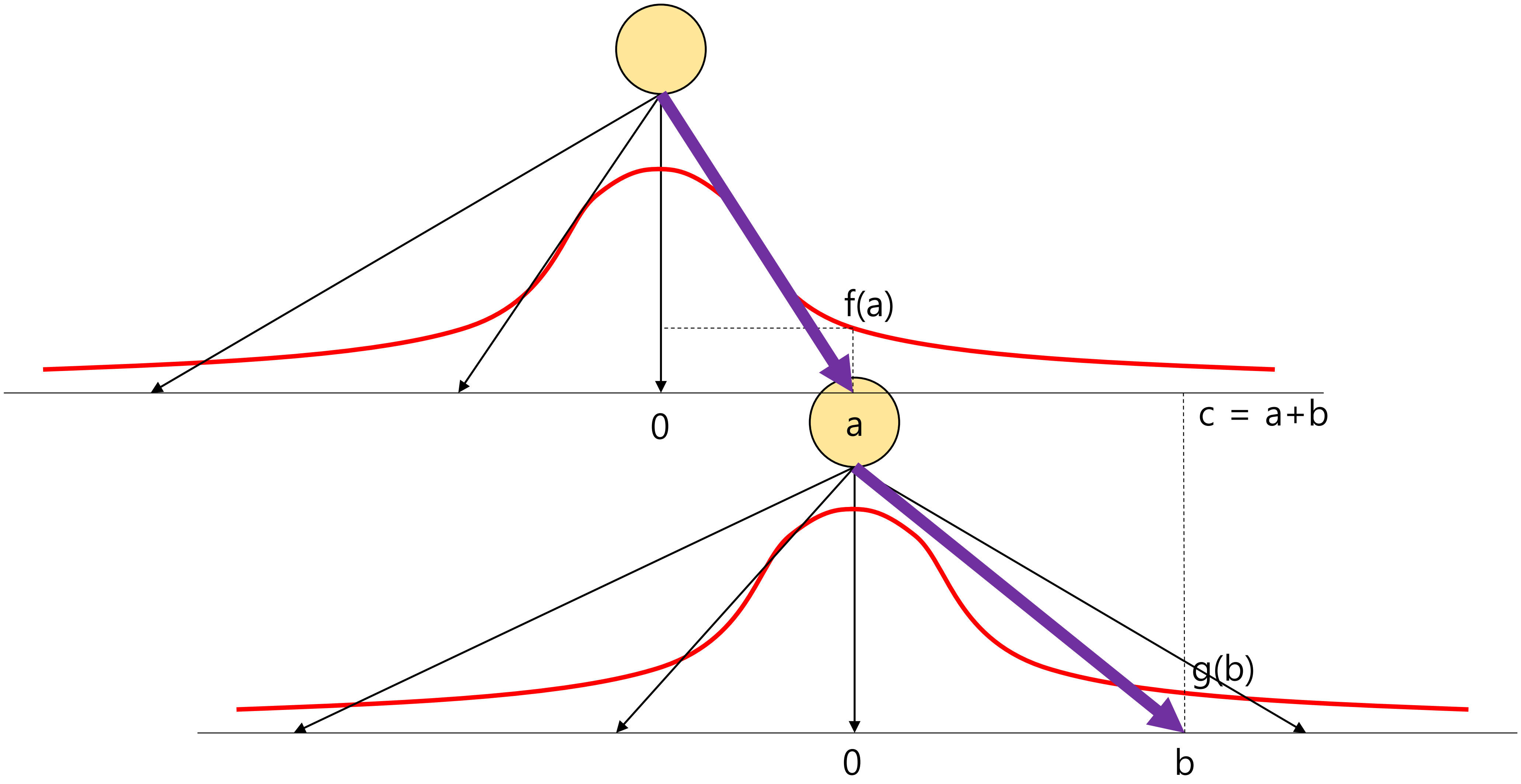
- 이동 불변성.
- 이미지의 크기를 줄이기 때문에 학습할 노드의 수가 줄어들어 **학습속도를 높**이는 효과.
- 하지만 **정보 손실**이 일어남.



Appendix.

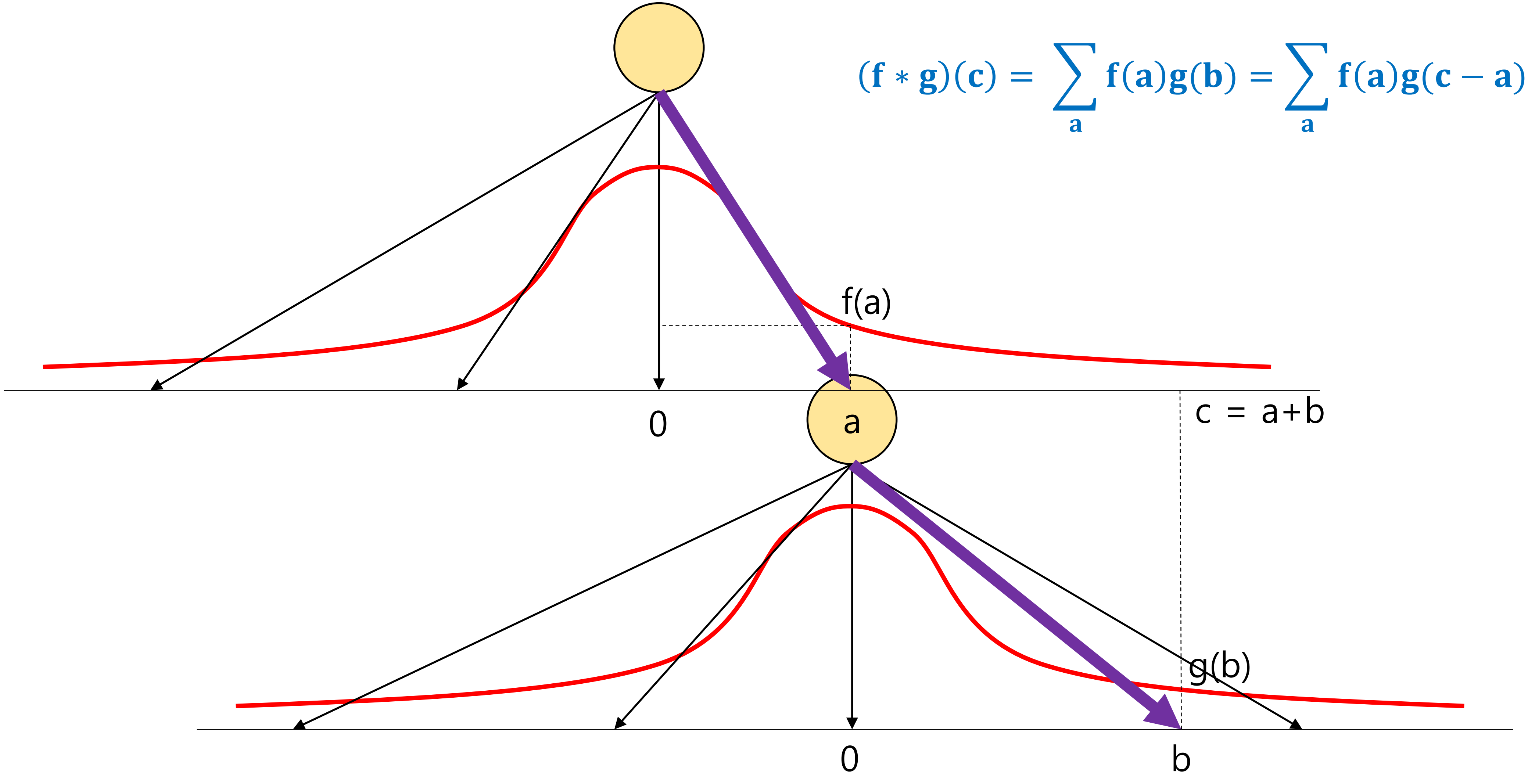

f(a)
0
a


f(a)
0
a
c = a+b
g(b)
0
b



$f(a)g(b)$
$= f(a)g(c-a)$

$$(\mathbf{f} * \mathbf{g})(\mathbf{c}) = \sum_{\mathbf{a}} \mathbf{f}(\mathbf{a})\mathbf{g}(\mathbf{b}) = \sum_{\mathbf{a}} \mathbf{f}(\mathbf{a})\mathbf{g}(\mathbf{c}-\mathbf{a})$$



f(a)g(b)
= f(a)g(c-a)