

Convolutional Neural Network

2024 Summer DL Session 5주차

2024.08.01 Session Leader, Yunju Jeong, Seongmin Yun



KOREA
UNIVERSITY

목차

KUBIG 2024 Summer DL Session

- 01** 들어가기 전에,,
- 02** MLP vs CNN
- 03** Convolution Layer
- 04** Padding & Stride
- 05** Pooling Layer
- 06** 다음주차 예고

01. 들어가기 전에,,

학습 목표

4주차 과제의 의도

전체 학습 과정에 대해 이해하고 정규화와 초기화 방법을 어떤 순서에 적용해야 하는지 파악하고자 함

5주차 학습 목표

MLP와 CNN의 차이를 알고, 이미지 처리에 강점을 가지는 CNN이 무엇인지 알아보자.

02

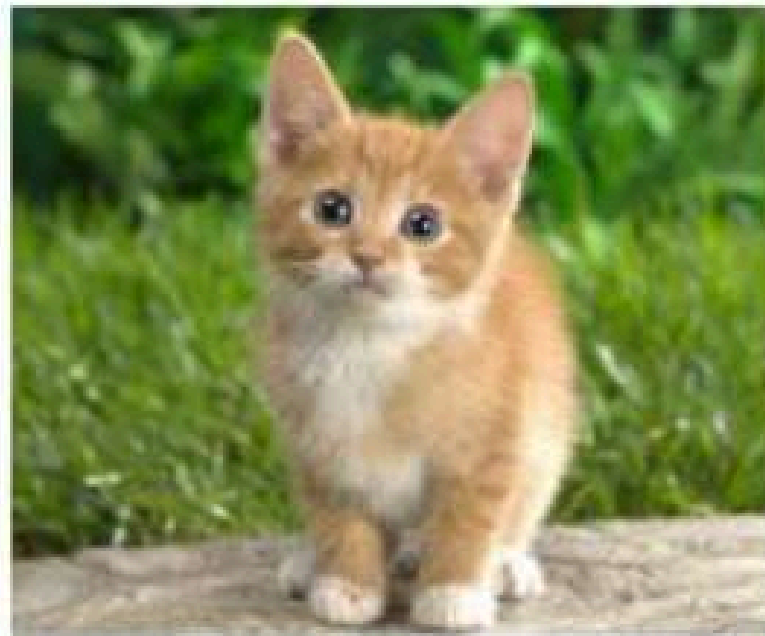
MLP vs CNN



02. MLP vs CNN

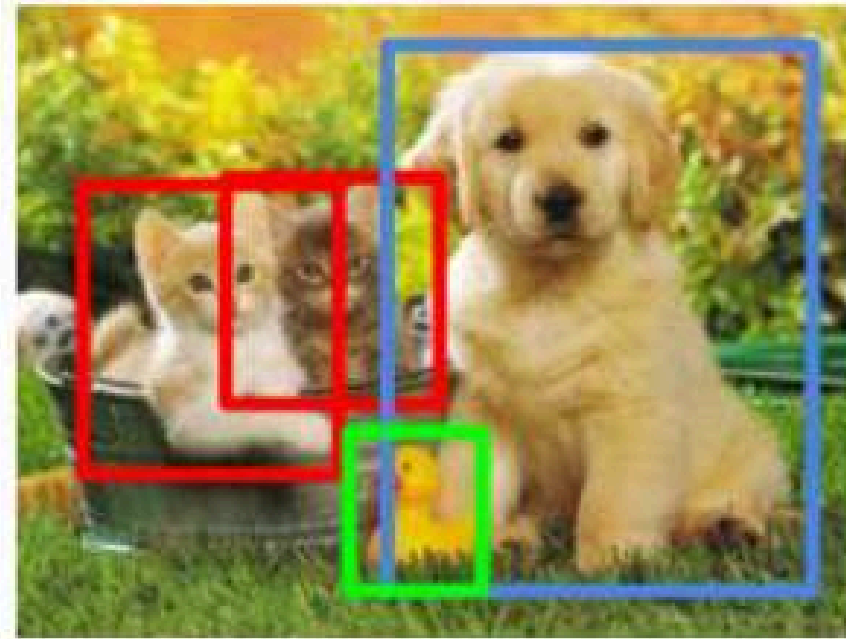
Computer Vision Tasks

Classification



CAT

Object Detection



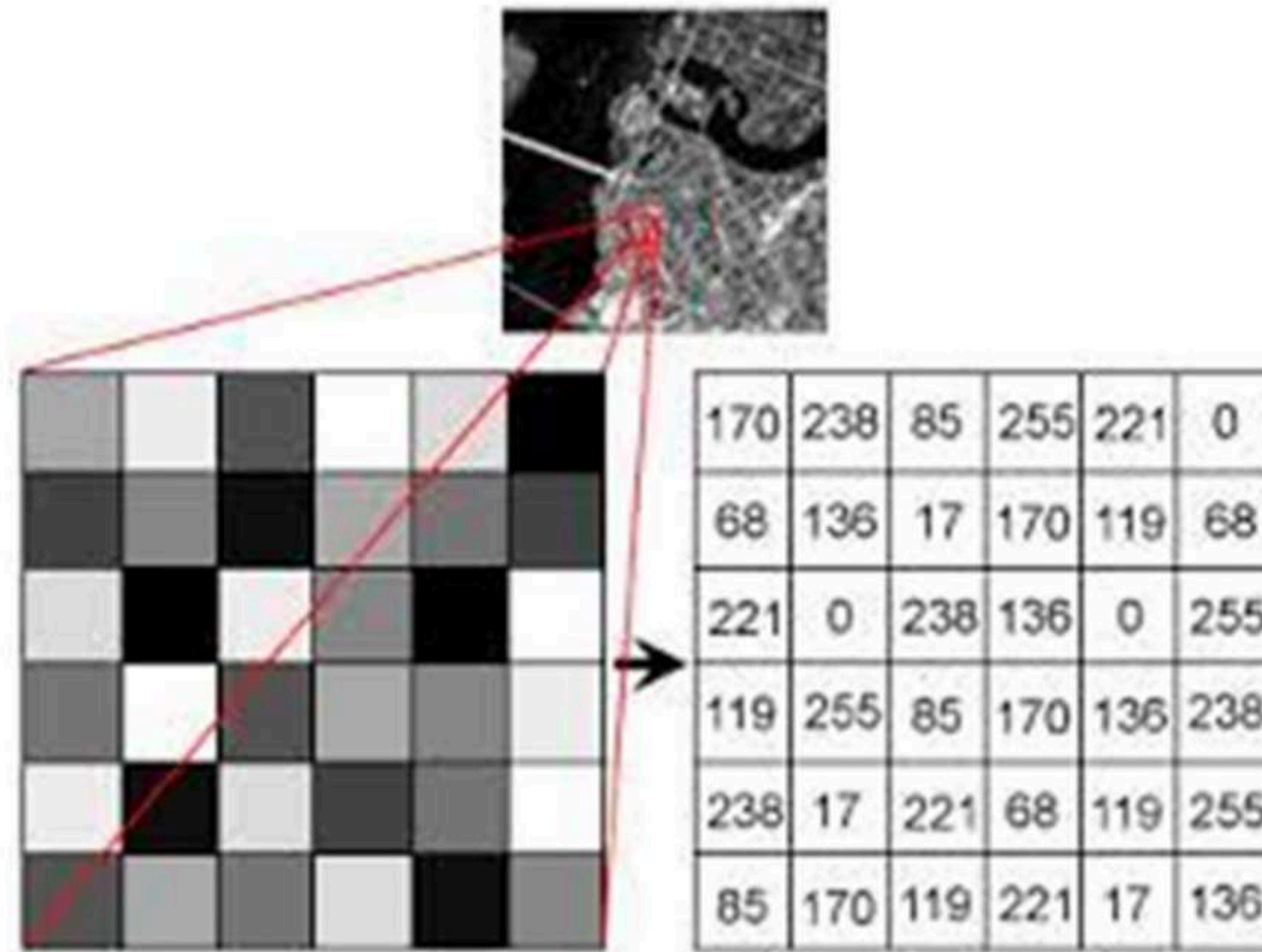
CAT, DOG, DUCK

Instance Segmentation



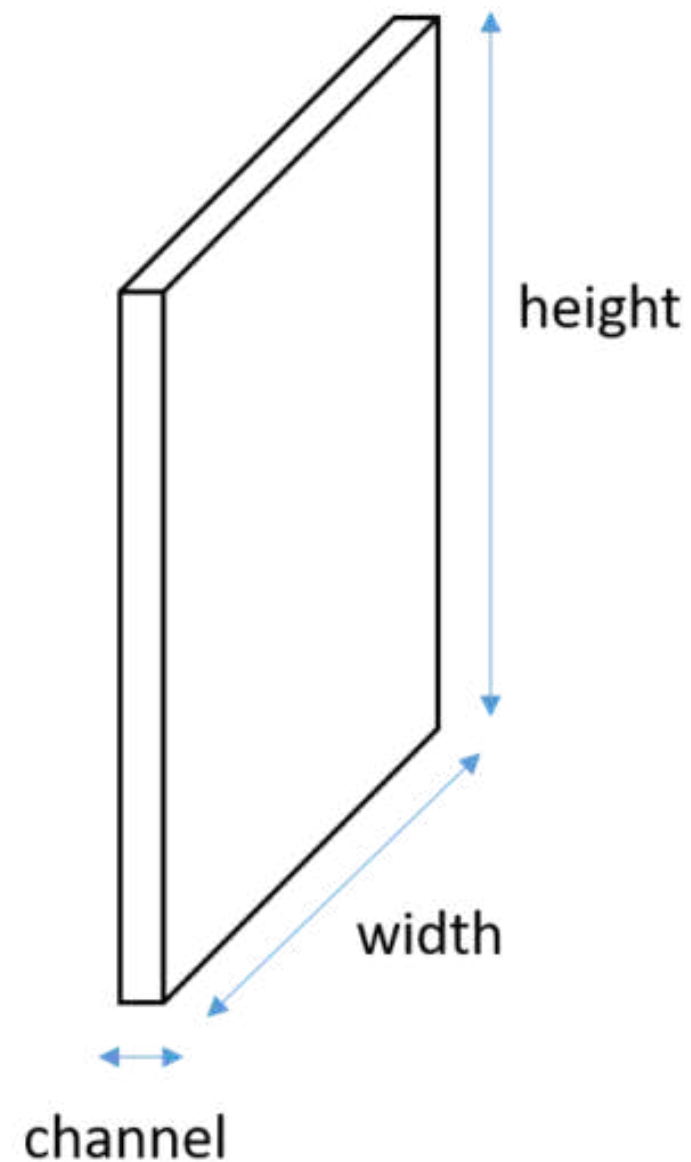
CAT, DOG, DUCK

Image Data



- $\#(\text{pixel}) = H * W$
- 픽셀 수 \uparrow , 화질(resolution) \uparrow
- 픽셀 값 = 밝기 값
- 0~255 사이, 0: 검정 255: 하양

Image Data



이미지 텐서의 크기

$H * W * C$

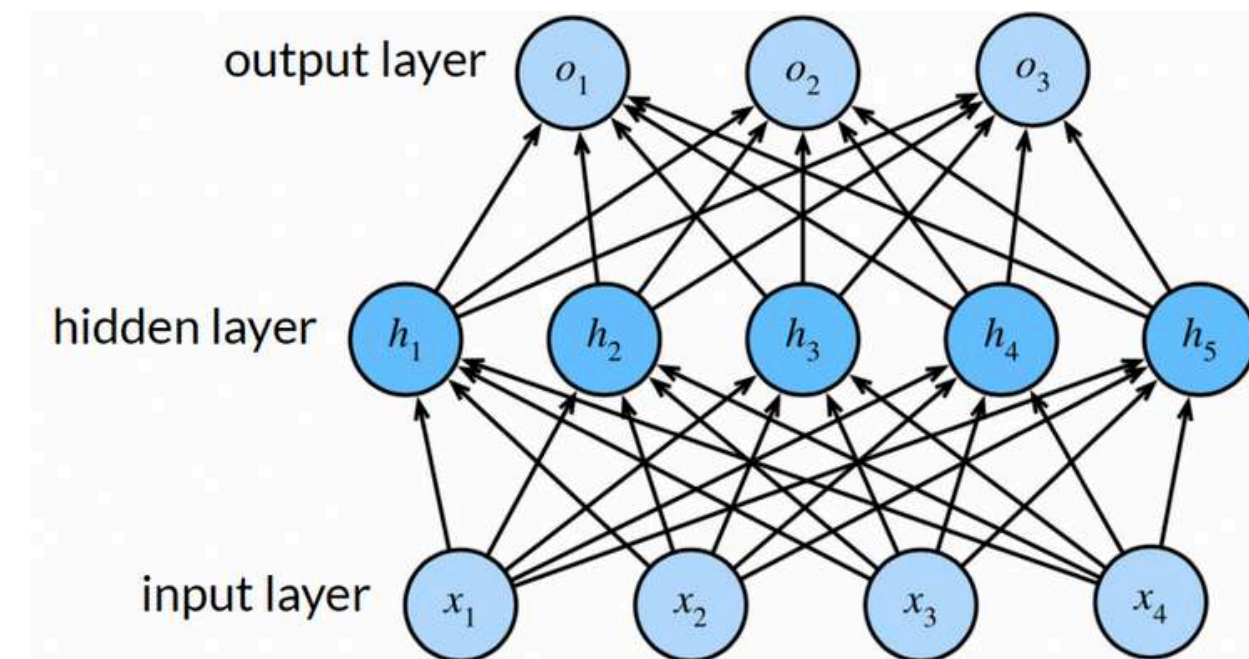
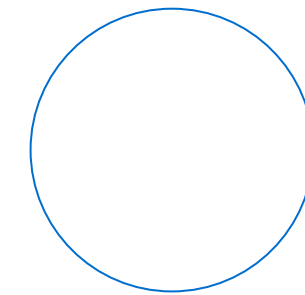
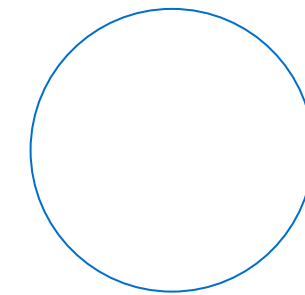
#(Channel) = 3 : RGB 채널 이미지

#(Channel) = 1 : 흑백 이미지

02. MLP vs CNN



$$h_i = f \left(\sum_{j=1}^m (w_{ij} \cdot x_j) + b_i \right)$$



Principles for Visual Domain

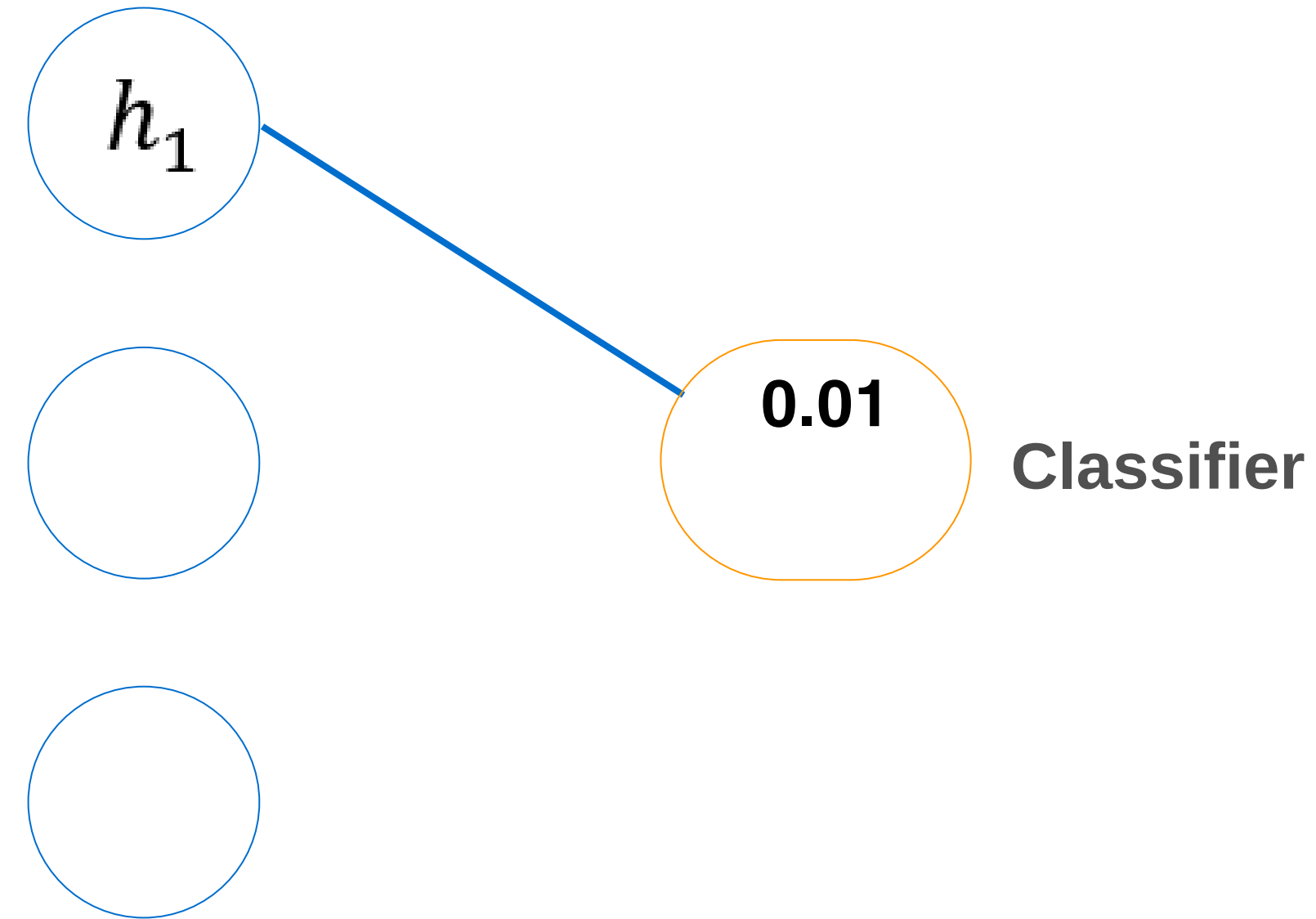
1. Locality principle

- 먼 지역의 내용은 무관하다.
- 이미지의 특정 부분을 인식할 때, 그 부분과 멀리 떨어진 다른 지역의 정보는 중요하지 않다고 간주된다. -> 이미지를 스캔하면서 각 이미지 조각(패치)에 점수를 매긴다 !

02. MLP vs CNN



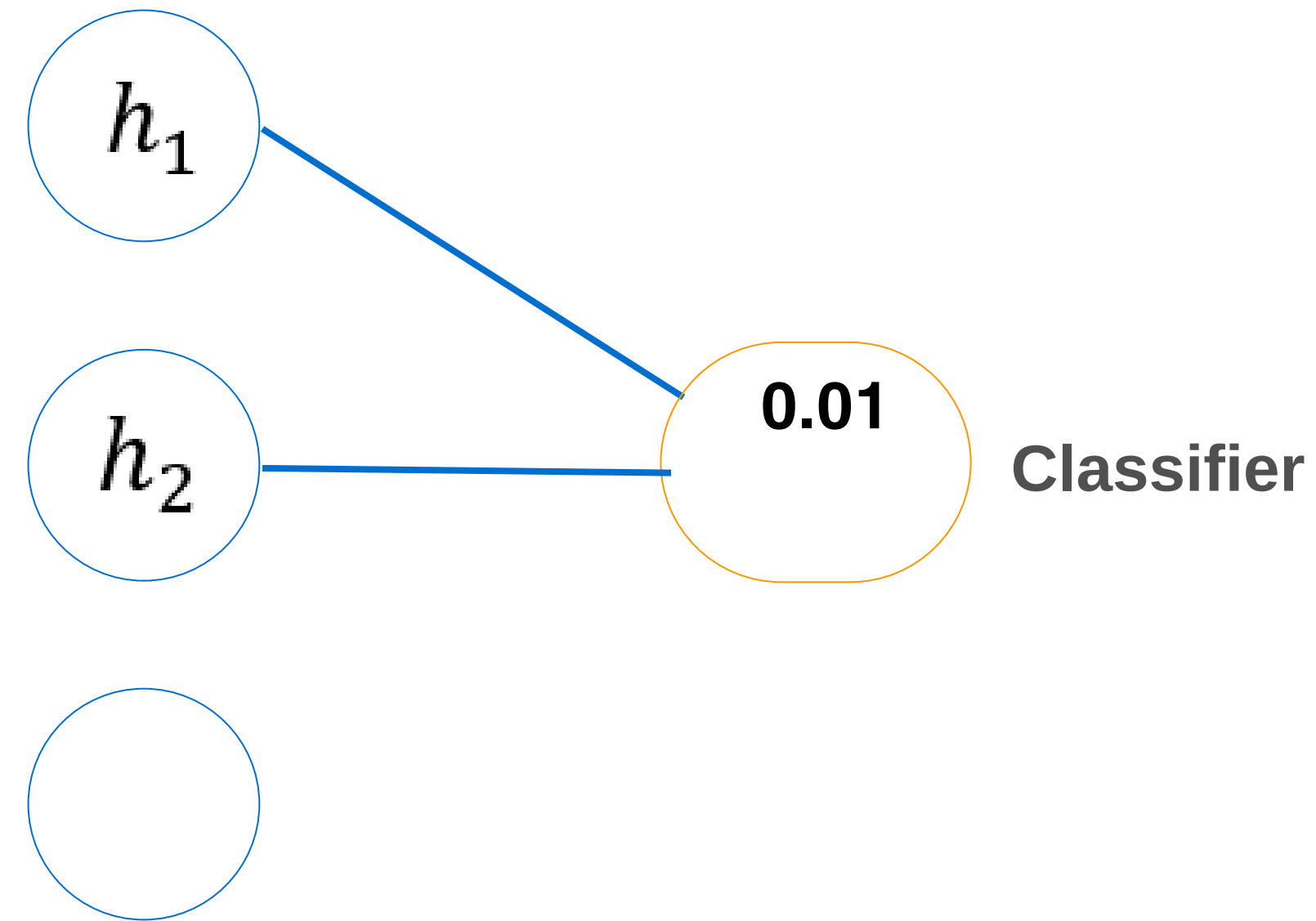
Locality principle



02. MLP vs CNN



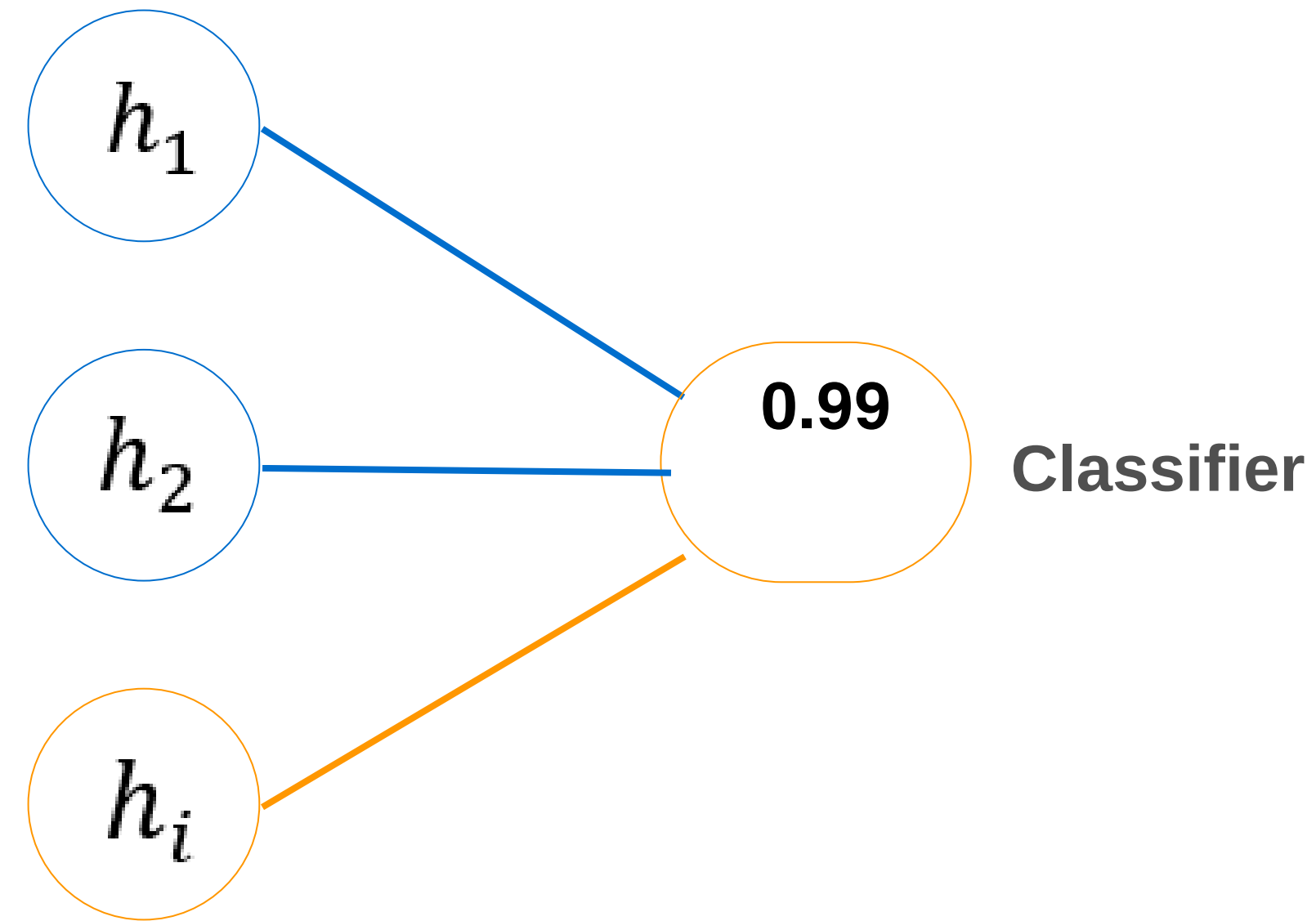
Locality principle



02. MLP vs CNN



Locality principle



Principles for Visual Domain

1. Locality principle

- 먼 지역의 내용은 무관하다.
- 이미지의 특정 부분을 인식할 때, 그 부분과 멀리 떨어진 다른 지역의 정보는 중요하지 않다고 간주된다. -> 이미지를 스캔하면서 각 이미지 조각(패치)에 점수를 매긴다 !

2. Spatial Invariance

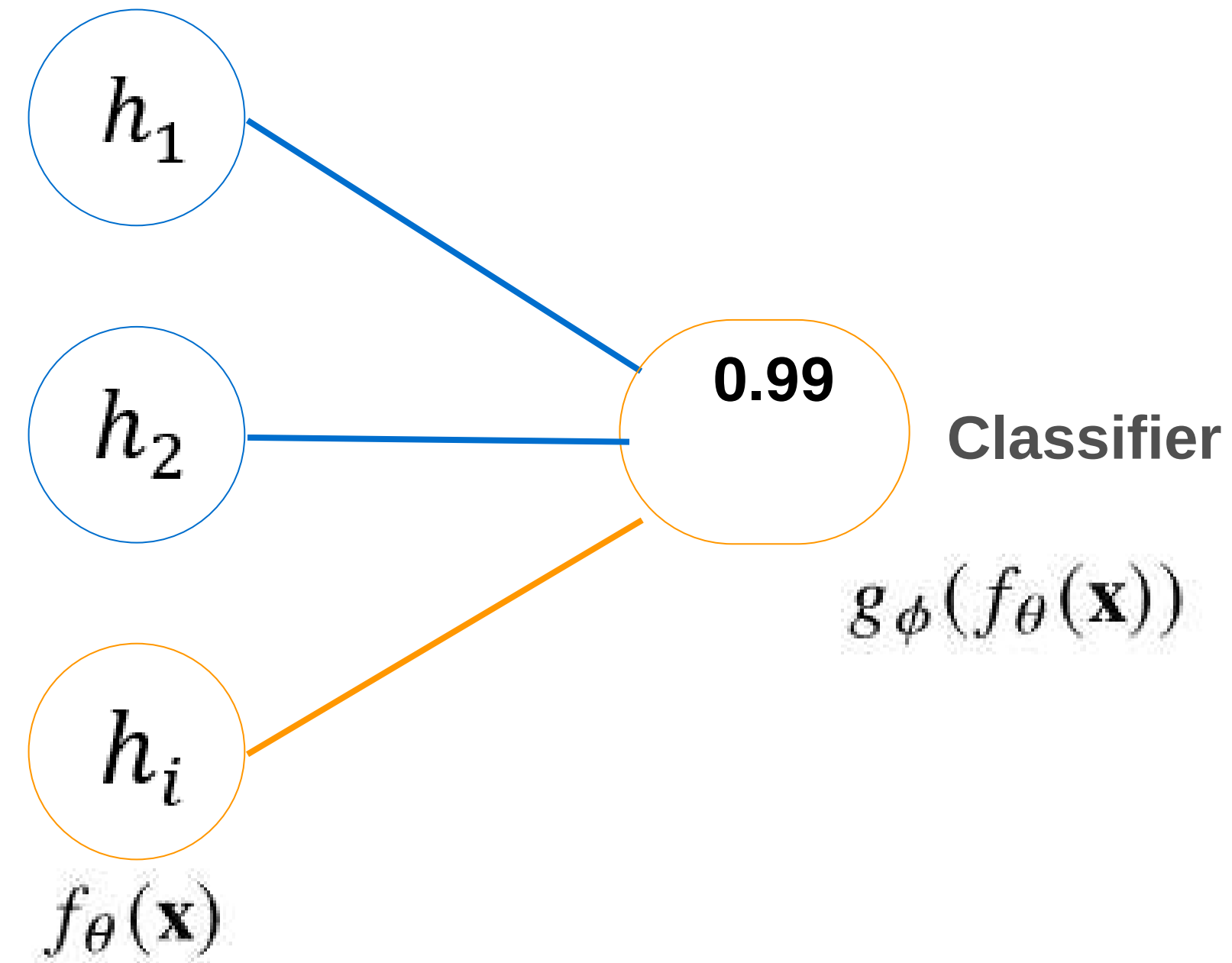
- 객체의 모습은 그 위치에 의존하지 않는다.
- 같은 객체가 이미지의 어느 위치에 있든지 간에 시스템이 일관되게 그 객체를 인식할 수 있어야 한다.

| | |
|--|---|
| translation invariance | translation equivariance |
| $g_{\phi}(\mathcal{T}[\mathbf{x}]) = g_{\phi}(\mathbf{x})$ | $f_{\theta}(\mathcal{T}[\mathbf{x}]) = \mathcal{T}[f_{\theta}(\mathbf{x})]$ |
| \mathcal{T} translation operation | f_{θ}, g_{ϕ} models |

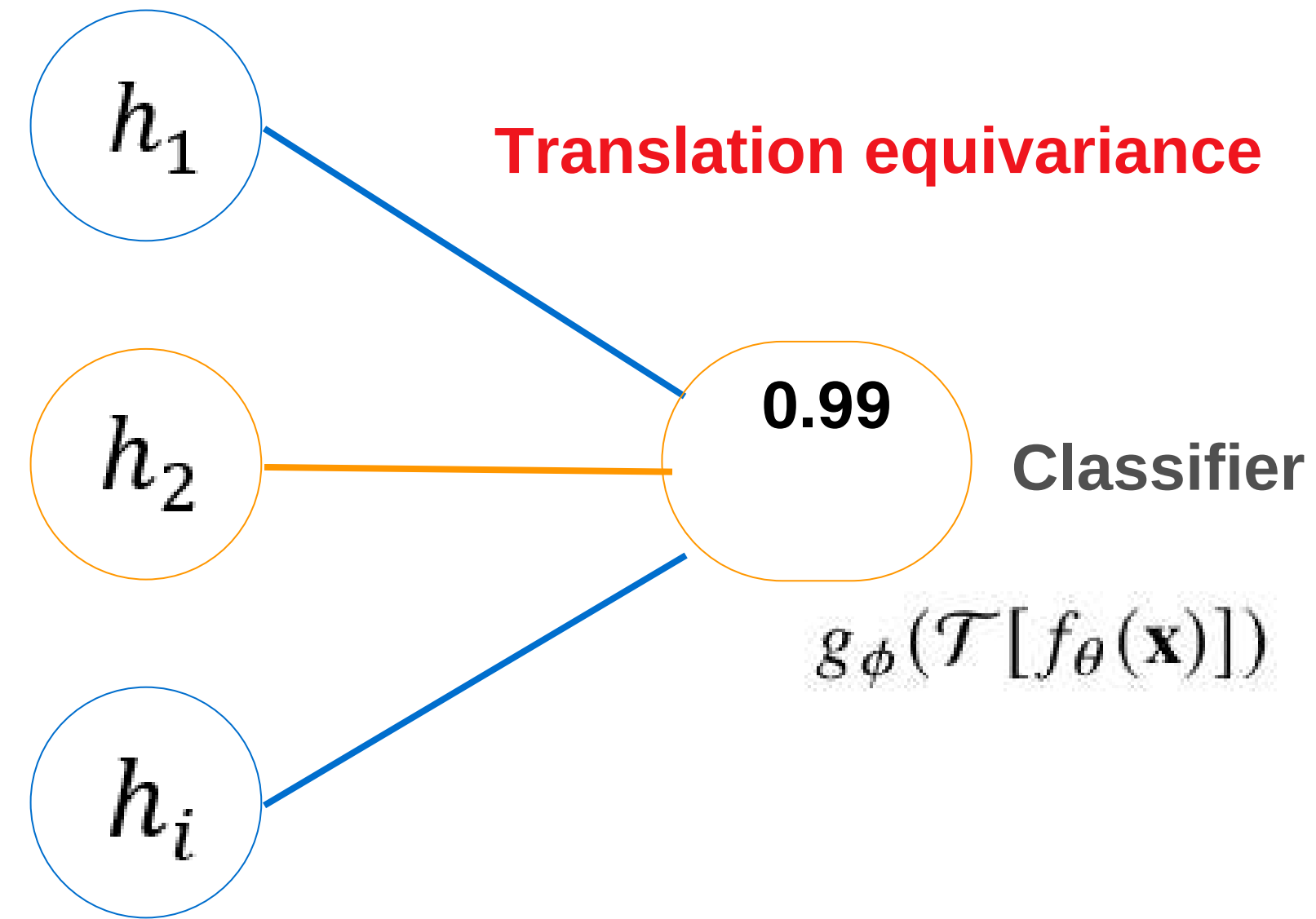
02. MLP vs CNN



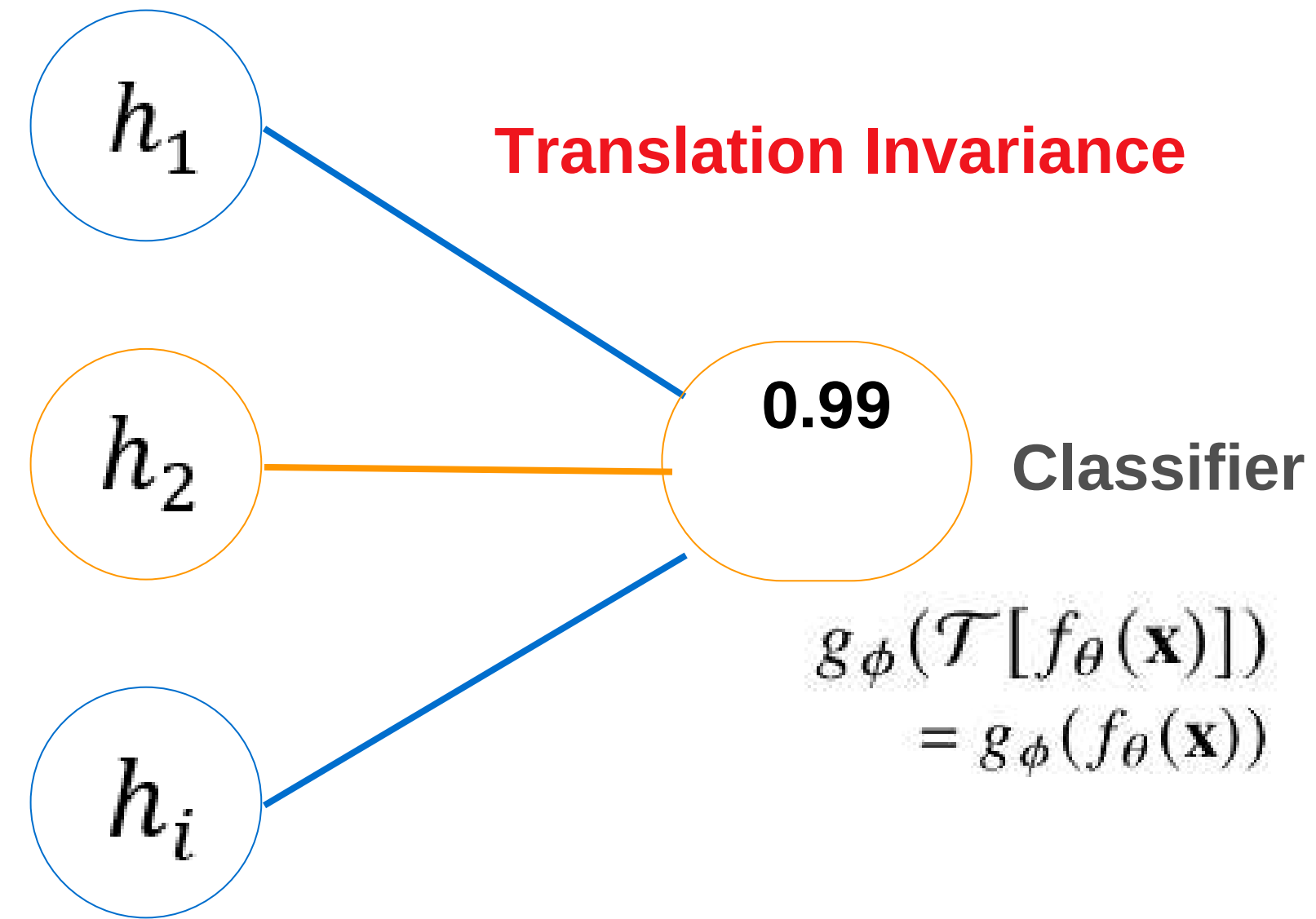
Locality principle



02. MLP vs CNN



02. MLP vs CNN



02. MLP vs CNN

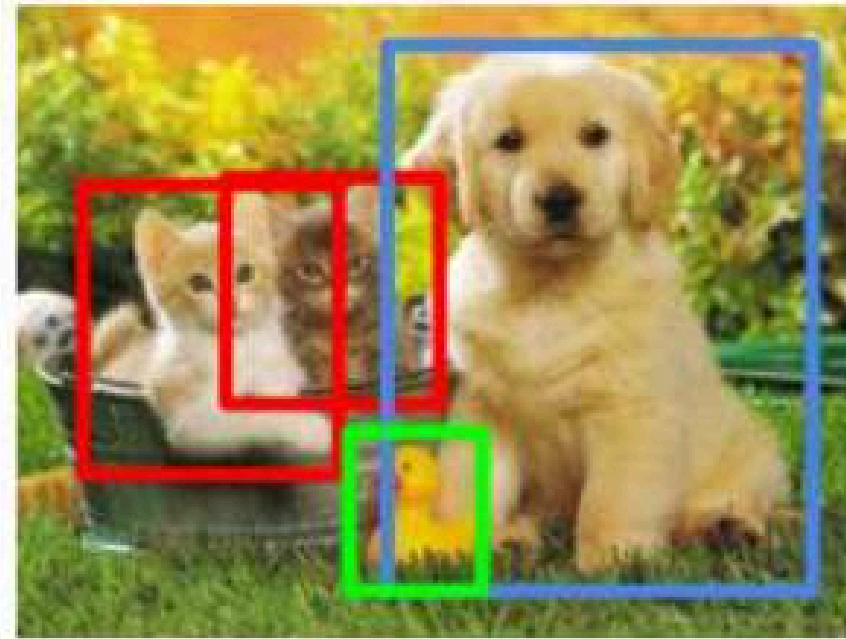
Classification



CAT

requires invariance

Object Detection



CAT, DOG, DUCK

requires equivariance

Instance Segmentation



CAT, DOG, DUCK

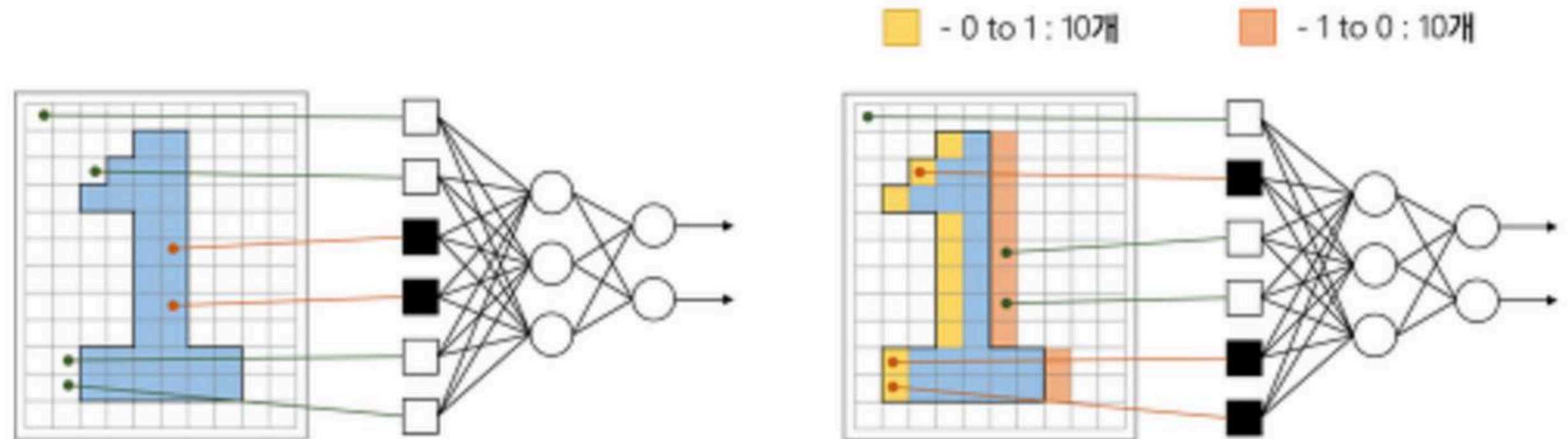
requires equivariance

02. MLP vs CNN

이미지에서 MLP 의 한계

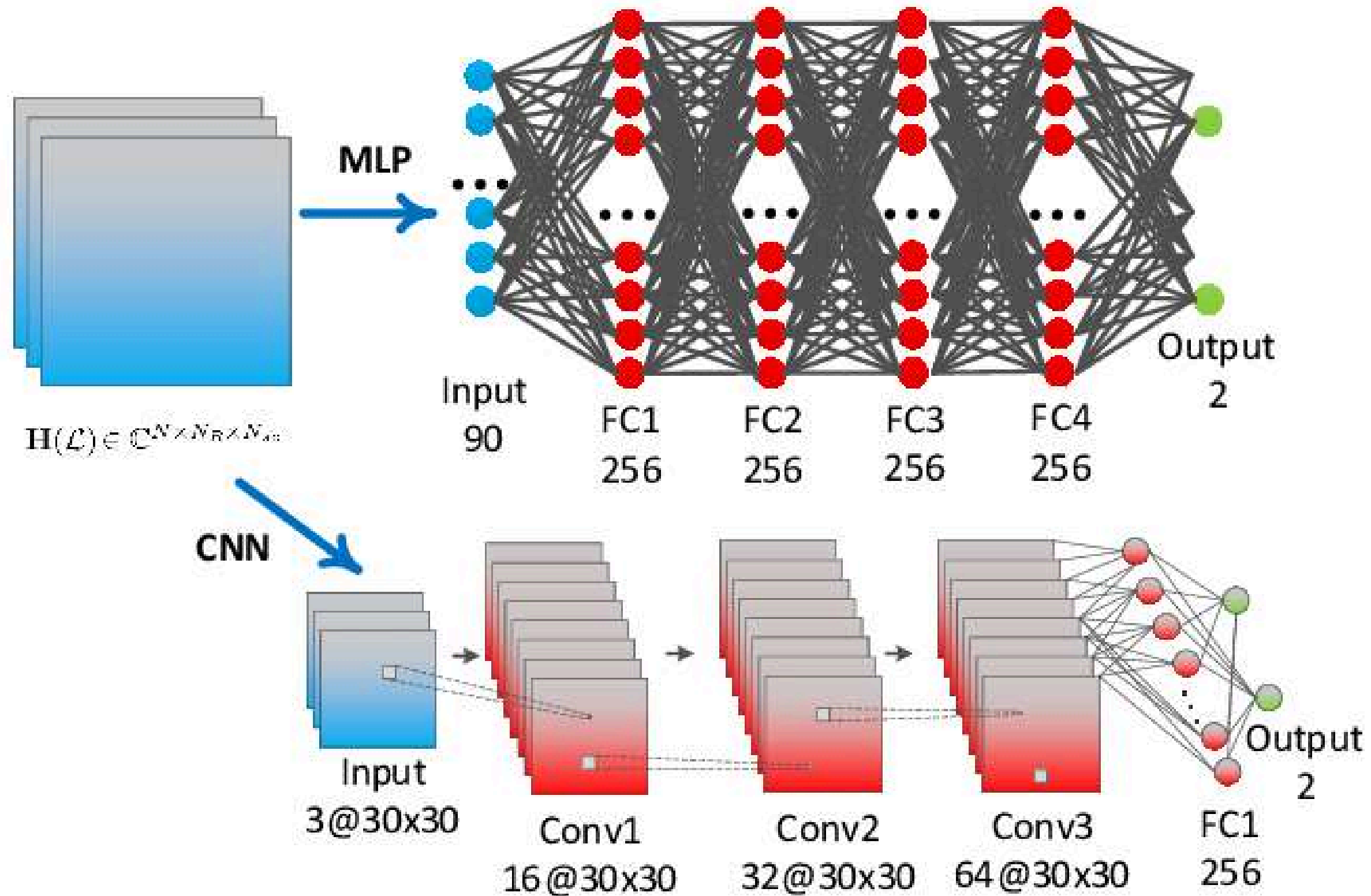
MLP : 이미지 행렬을 1차원 배열로 만들고 신경망에 입력으로 넣어 가중치를 계산

CNN : 사람이 보는 것 처럼 이미지 행렬 2차원 배열에서 가로, 세로 축 전부 합성곱 연산과 풀링 연산으로 특징점을 추출



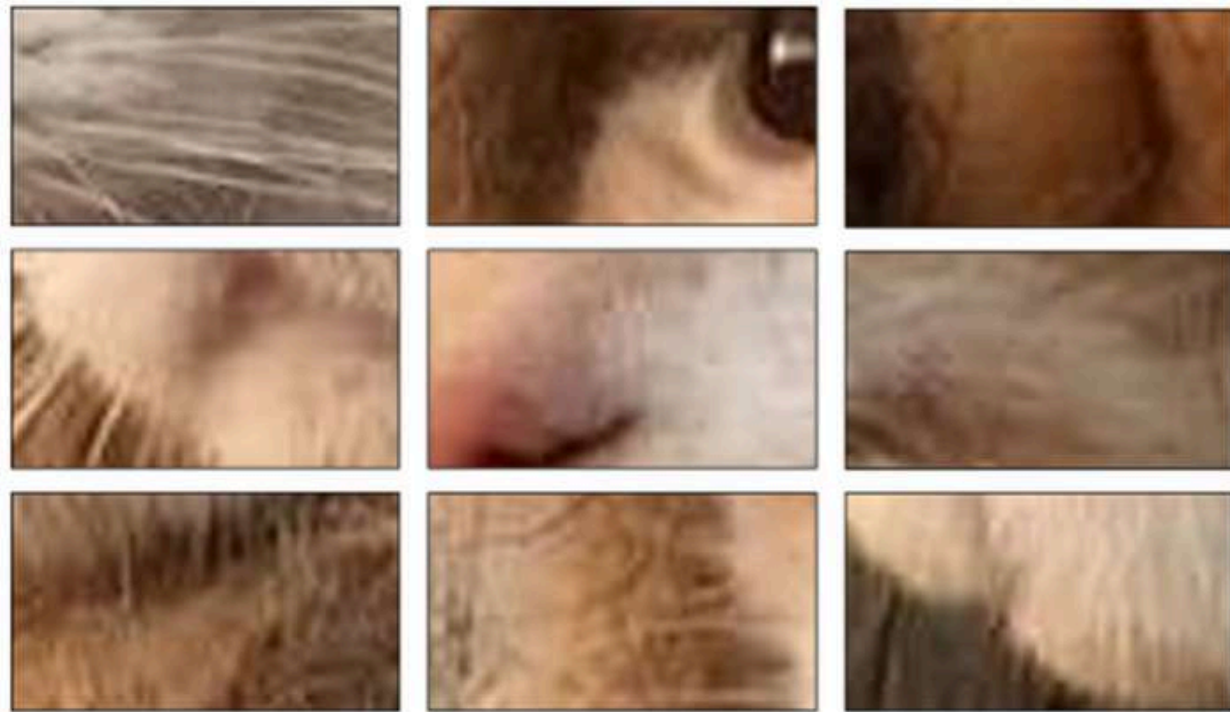
한 칸씩만 움직였는데
변화하는 인풋값이 20개

02. MLP vs CNN



02. MLP vs CNN

CNN이 이미지를 인식하는 단계



1단계 : 가로, 동그라미, 세모, 부드러움



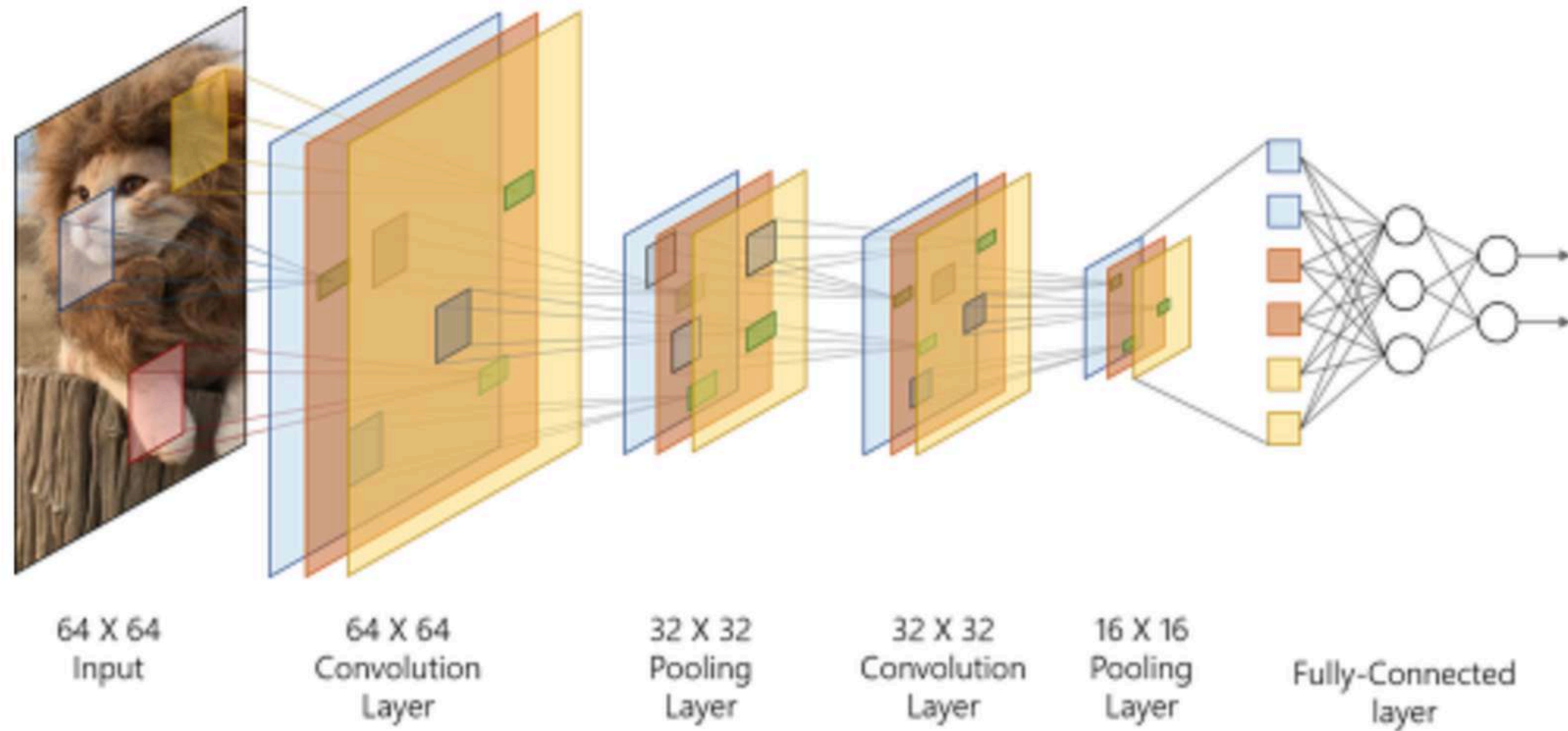
2단계 : 눈, 코, 귀, 발



3단계 : 고양이!

02. MLP vs CNN

CNN이 이미지를 인식하는 단계



03

Convolution Layer

03. Convolution Layer

What is convolution?

what is convolution? cross correlation?

합성곱: $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t - \tau) d\tau$

교차상관: $(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t + \tau) d\tau$

| | | | | |
|-----------------|-----------------|-----------------|---|---|
| 1 _{x1} | 1 _{x0} | 1 _{x1} | 0 | 0 |
| 0 _{x0} | 1 _{x1} | 1 _{x0} | 1 | 0 |
| 0 _{x1} | 0 _{x0} | 1 _{x1} | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Image

| | | |
|---|--|--|
| 4 | | |
| | | |
| | | |

Convolved
Feature

03. Convolution Layer

What is convolution?

1D Conv $[\mathbf{f} * \mathbf{x}](i) = \sum_p \mathbf{f}(p) \mathbf{x}(i + p)$

2D Conv $[\mathbf{f} * \mathbf{x}](i, j) = \sum_{p, q} \mathbf{f}(p, q) \mathbf{x}(i + p, j + q)$

3D Conv $[\mathbf{f} * \mathbf{x}](i, j, k) = \sum_{p, q, r} \mathbf{f}(p, q, r) \mathbf{x}(i + p, j + q, k + r)$

| | | | | |
|-----------------|-----------------|-----------------|---|---|
| 1 _{x1} | 1 _{x0} | 1 _{x1} | 0 | 0 |
| 0 _{x0} | 1 _{x1} | 1 _{x0} | 1 | 0 |
| 0 _{x1} | 0 _{x0} | 1 _{x1} | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

Image

| | | |
|---|--|--|
| 4 | | |
| | | |
| | | |

Convolved
Feature

02. MLP vs CNN



$$\begin{aligned} h_1 &= \sum_{|p| \leq \Delta} \mathbf{f}(p) \mathbf{x}(1 + p) \in \text{1st patch} \\ h_2 &= \sum_{|p| \leq \Delta} \mathbf{f}(p) \mathbf{x}(2 + p) \in \text{2nd patch} \\ &\vdots \\ h_i &= \sum_{|p| \leq \Delta} \mathbf{f}(p) \mathbf{x}(i + p) \in i\text{-th patch} \end{aligned}$$

02. MLP vs CNN

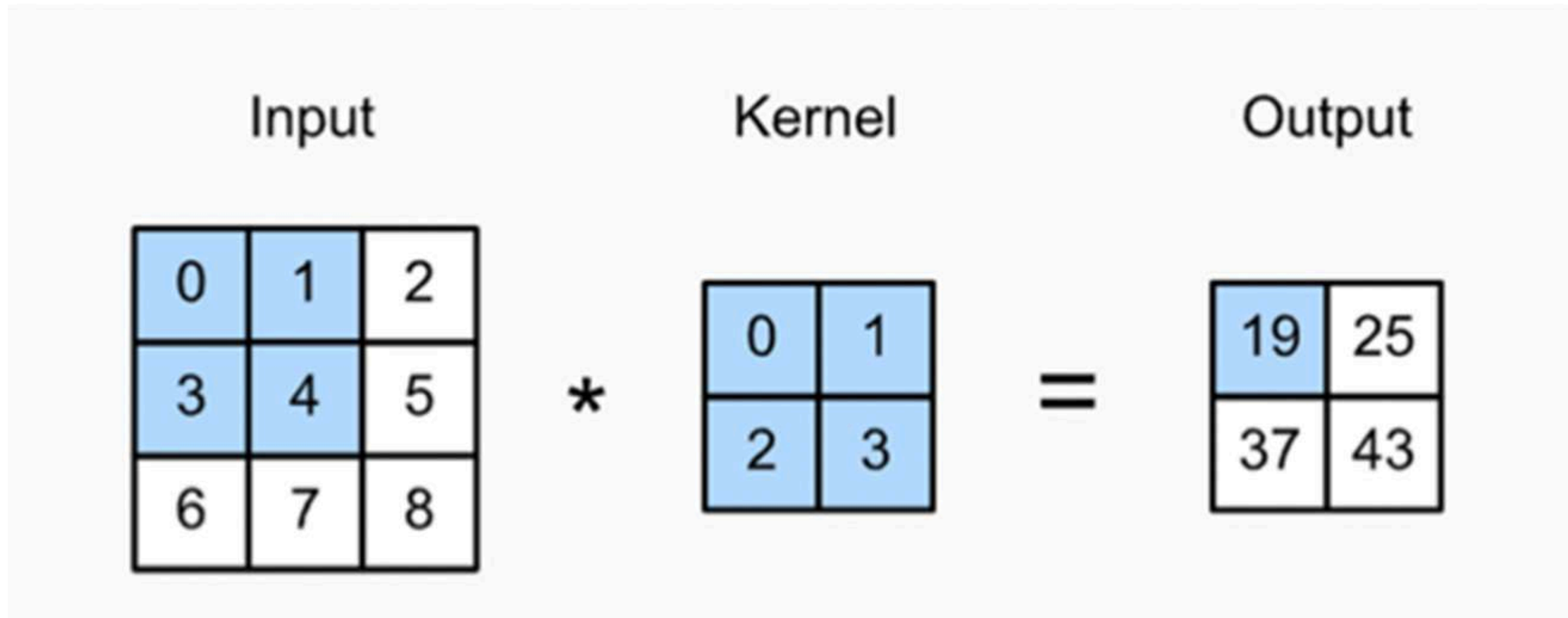


$$\begin{aligned} h_1 &= \sum_{|p| \leq \Delta} \mathbf{f}(p) \mathbf{x}(1 + p) \in \text{1st patch} \\ h_2 &= \sum_{|p| \leq \Delta} \mathbf{f}(p) \mathbf{x}(2 + p) \in \text{2nd patch} \\ &\vdots \\ h_i &= \sum_{|p| \leq \Delta} \mathbf{f}(p) \mathbf{x}(i + p) \in i\text{-th patch} \end{aligned}$$

filters

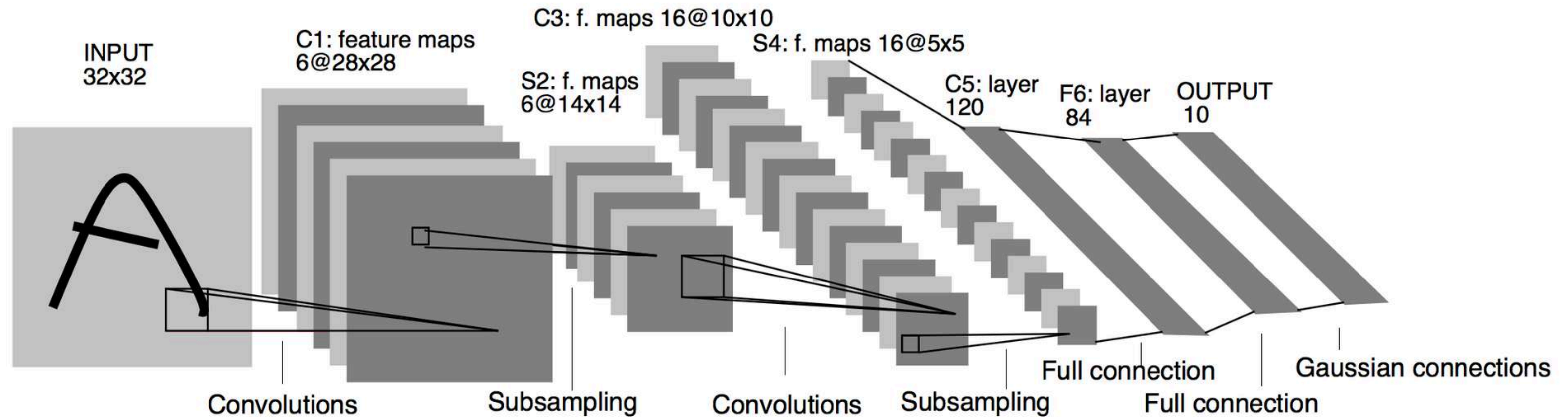
03. Convolution Layer

CNN 에서의 Convolution

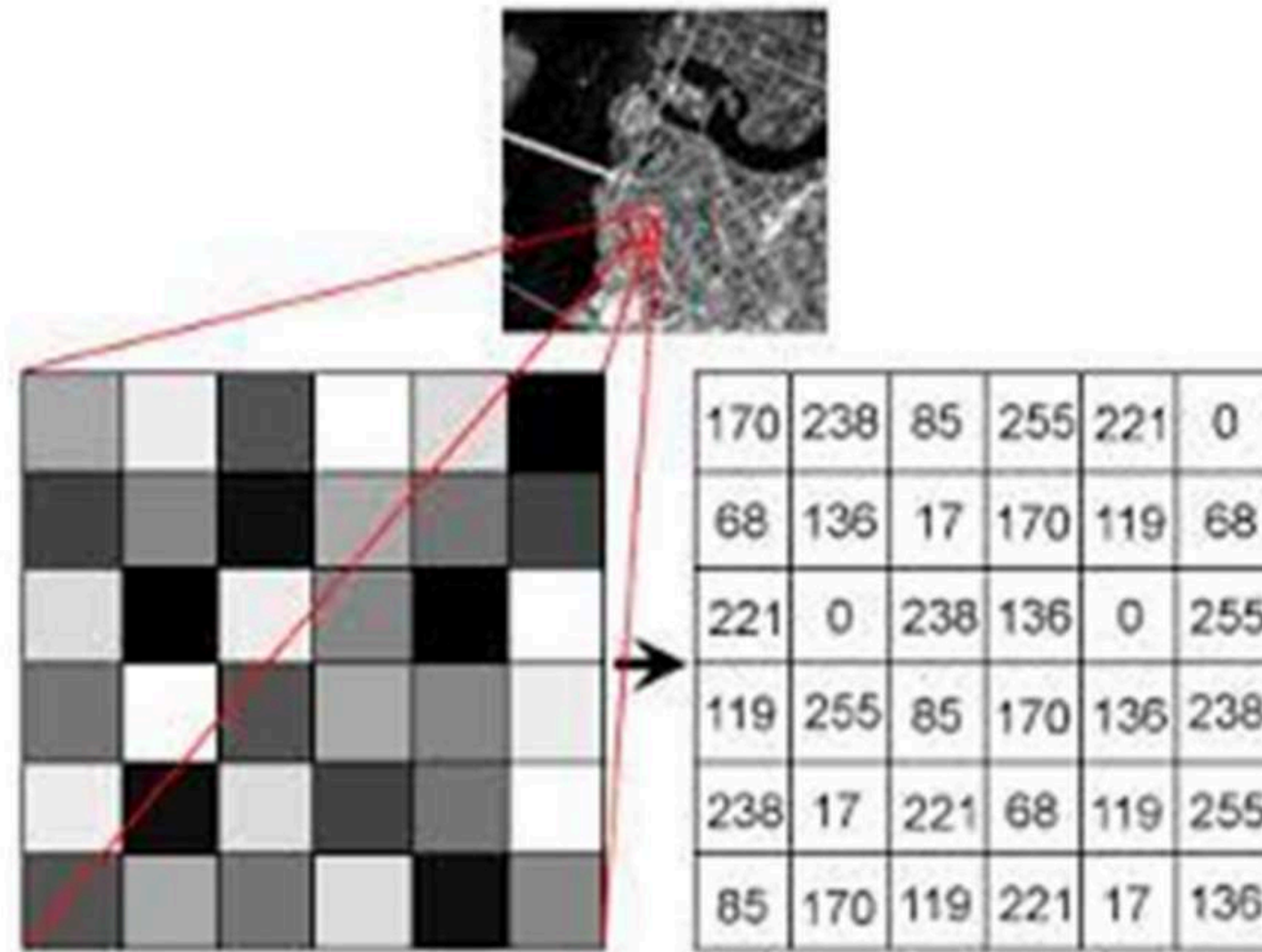


03. Convolution Layer

CNN 에서의 Convolution



03. Image Data



- $\#(\text{pixel}) = H * W$
- 픽셀 수 \uparrow , 화질(resolution) \uparrow
- 픽셀 값 = 밝기 값
- 0~255 사이, 0: 검정 255: 하양

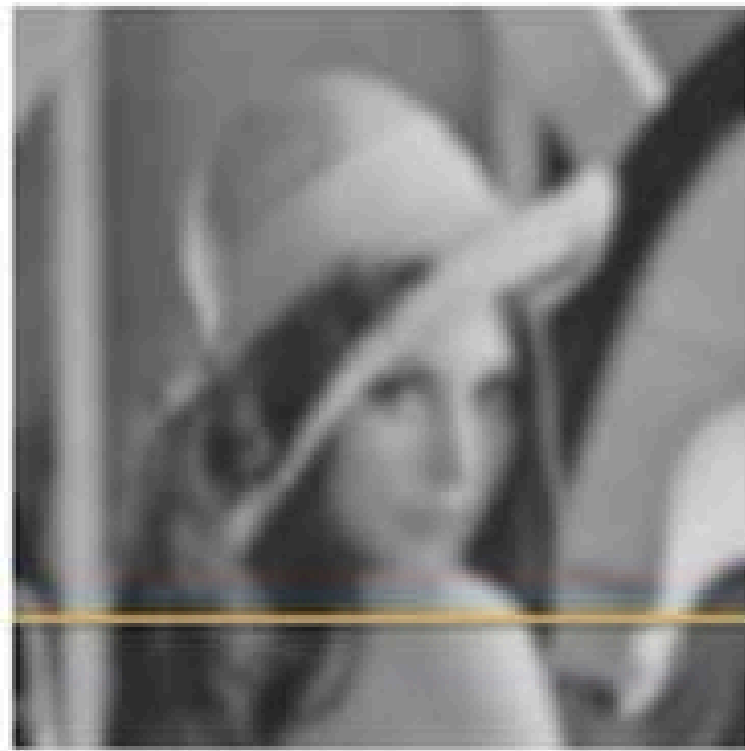
02. Image Data

- **Low frequency** : 이미지의 넓은 영역에 걸쳐 천천히 변화하는 부분
주로 이미지의 일반적인 형태나 배경, 대체적인 색상 변화
- **High Frequency**: 이미지에서 빠르게 변화하는 부분
즉 세밀한 텍스처나 에지(객체의 경계), 노이즈 등



original image

=



low frequency region

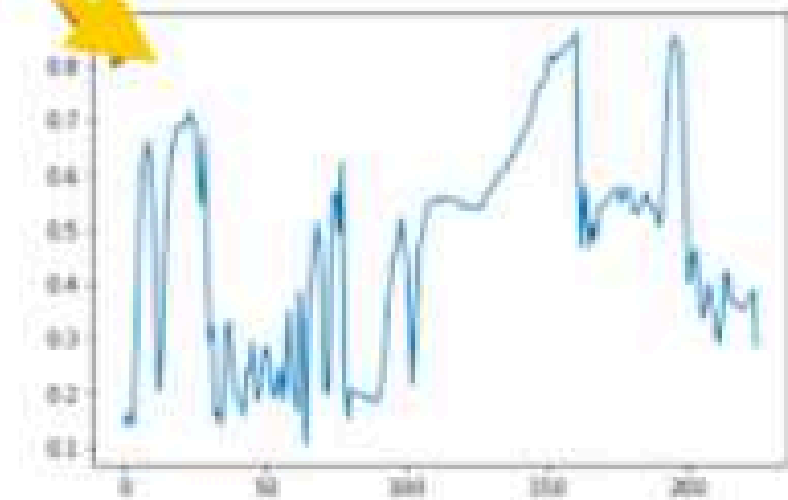
+



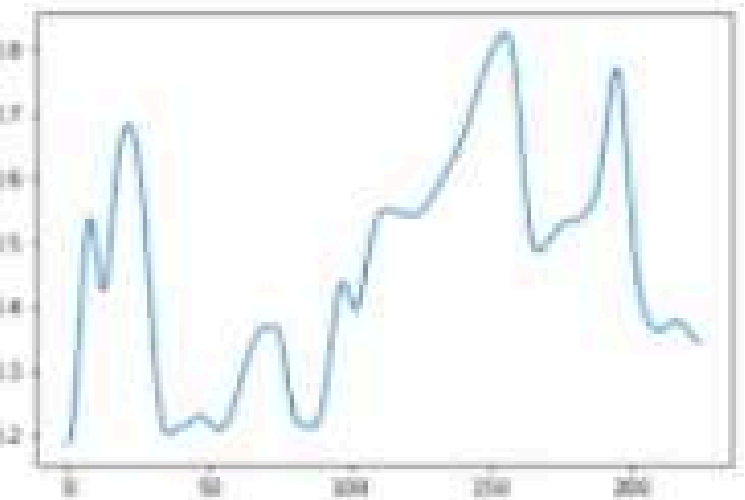
high frequency region

$$\mathcal{F}[\mathbf{f} * \mathbf{x}] = \mathcal{F}[\mathbf{f}] \mathcal{F}[\mathbf{x}]$$

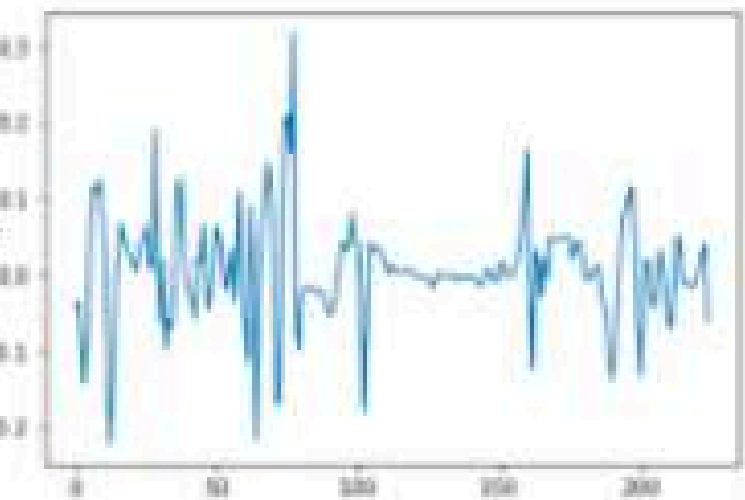
convolution filter image



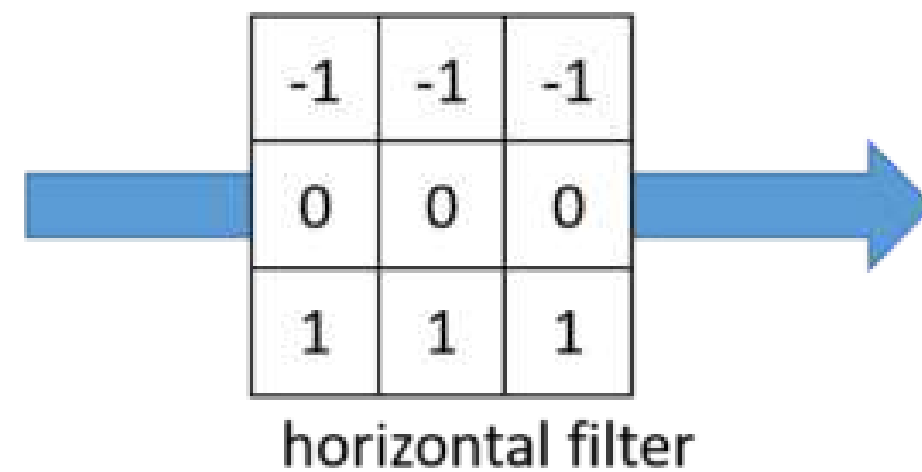
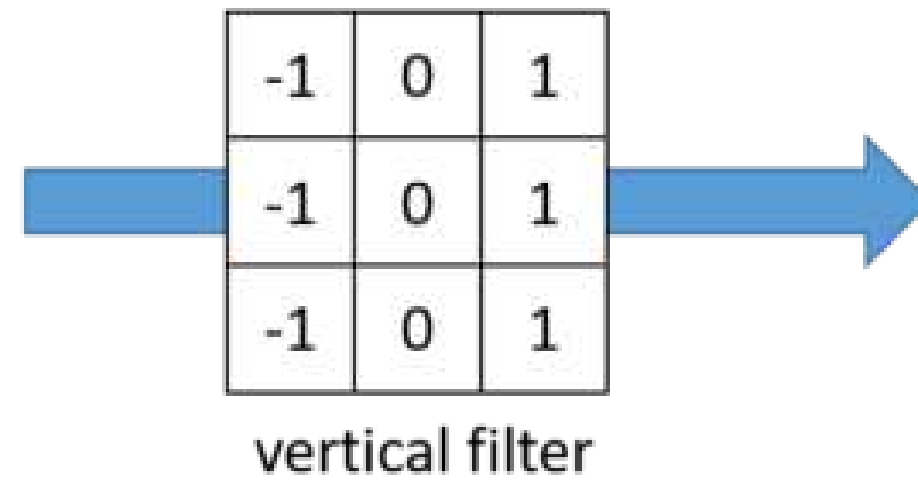
=



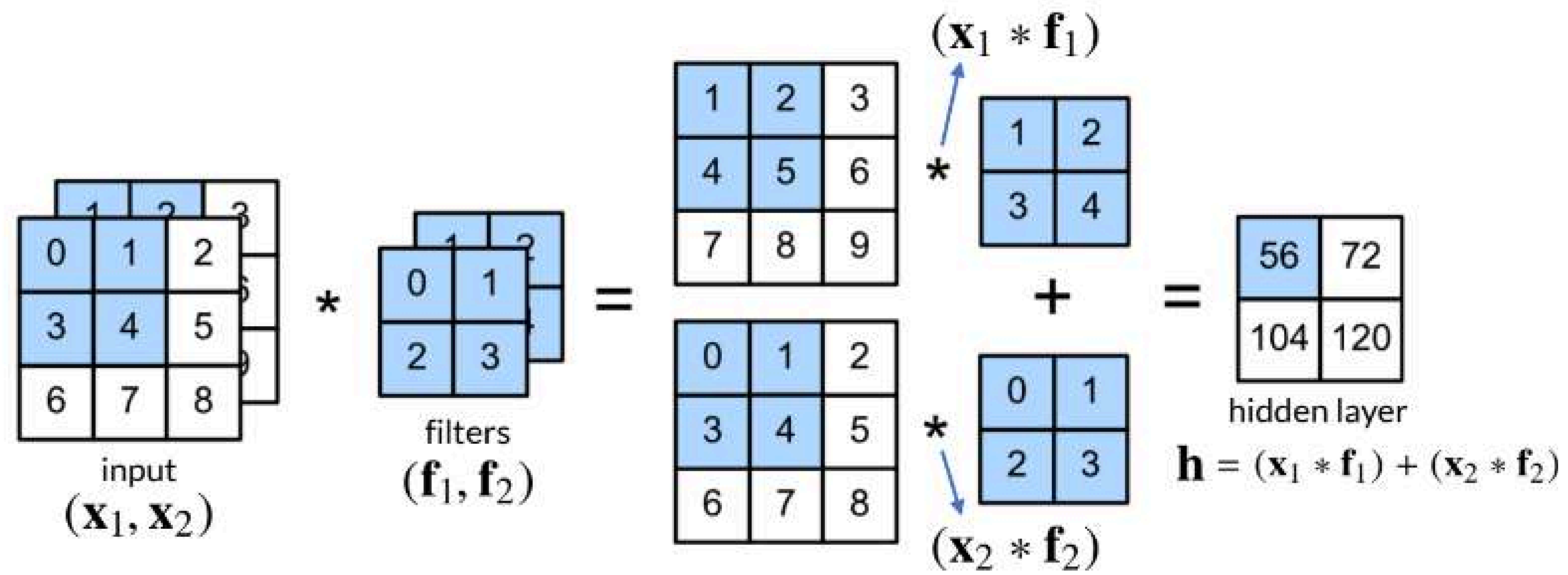
+



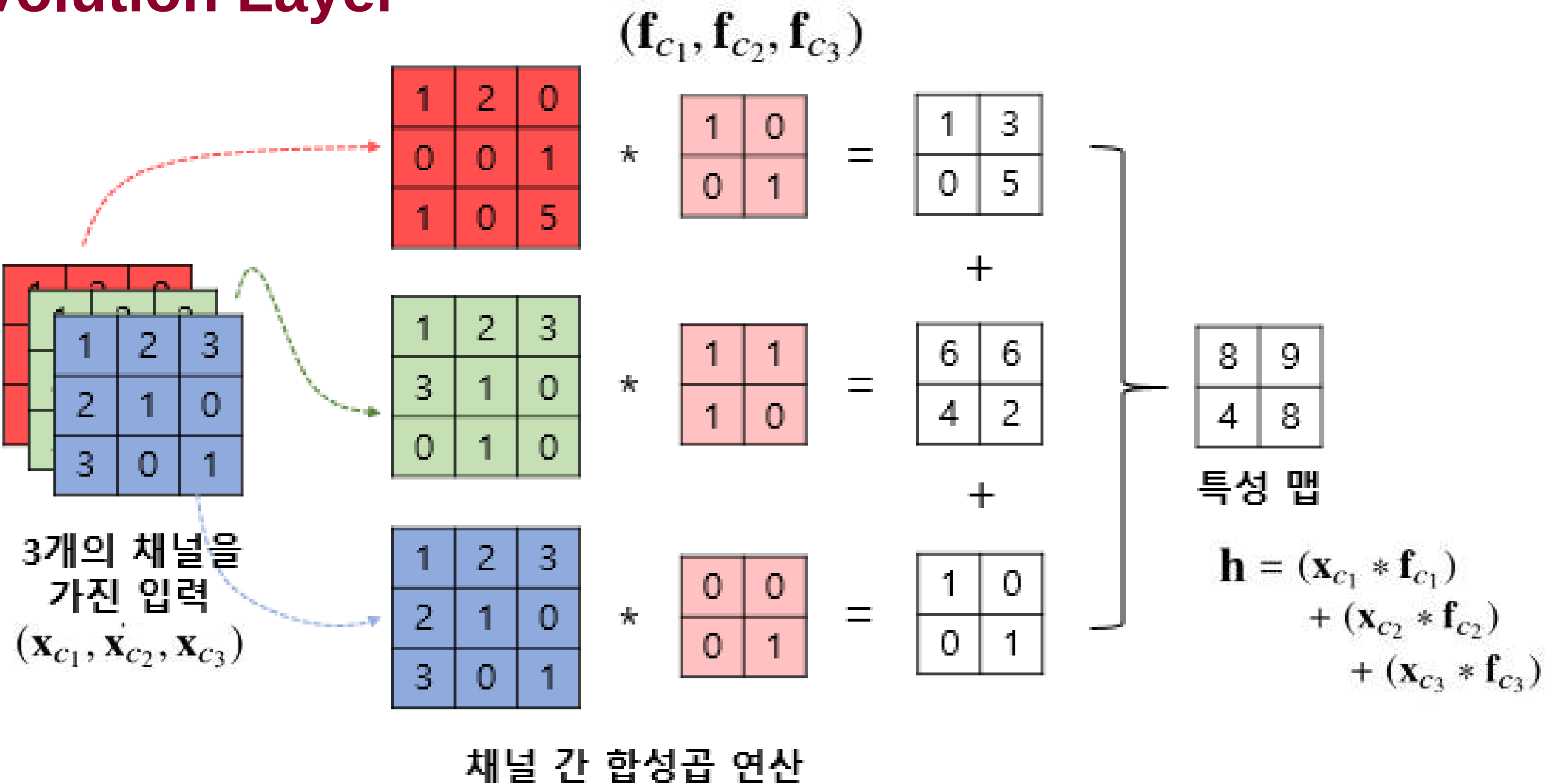
02. Filter



02. Filter



03. Convolution Layer

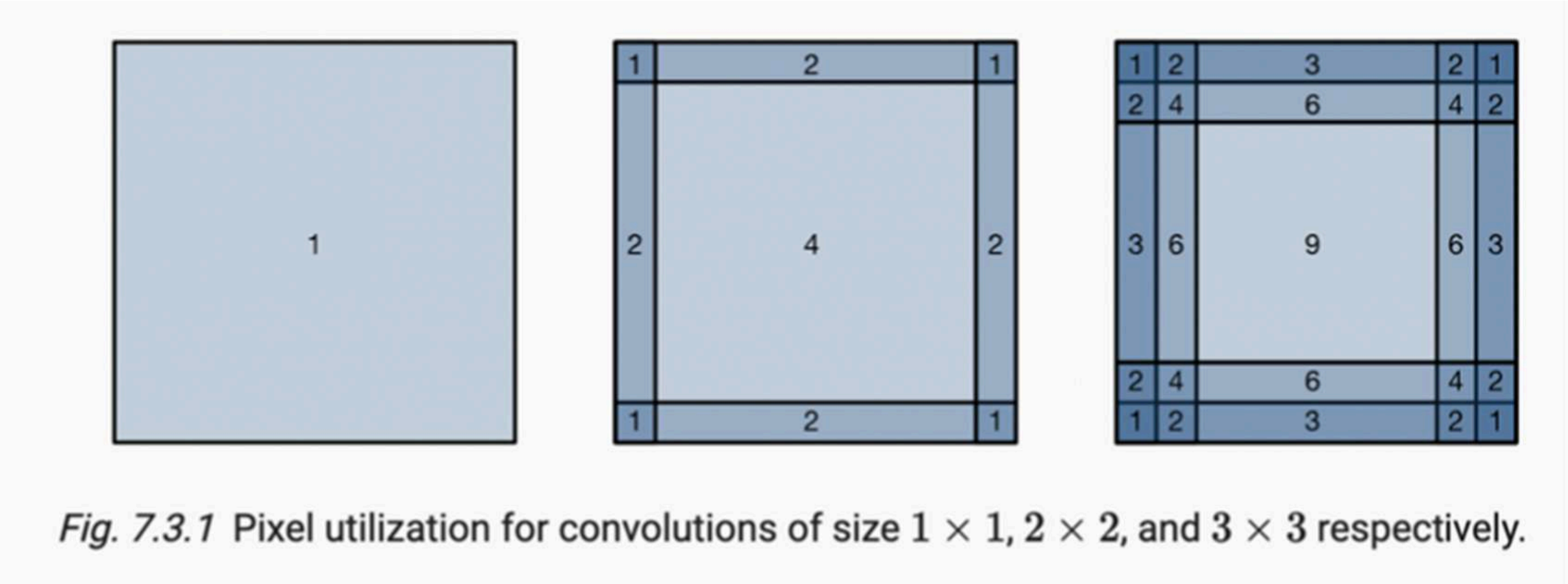


04

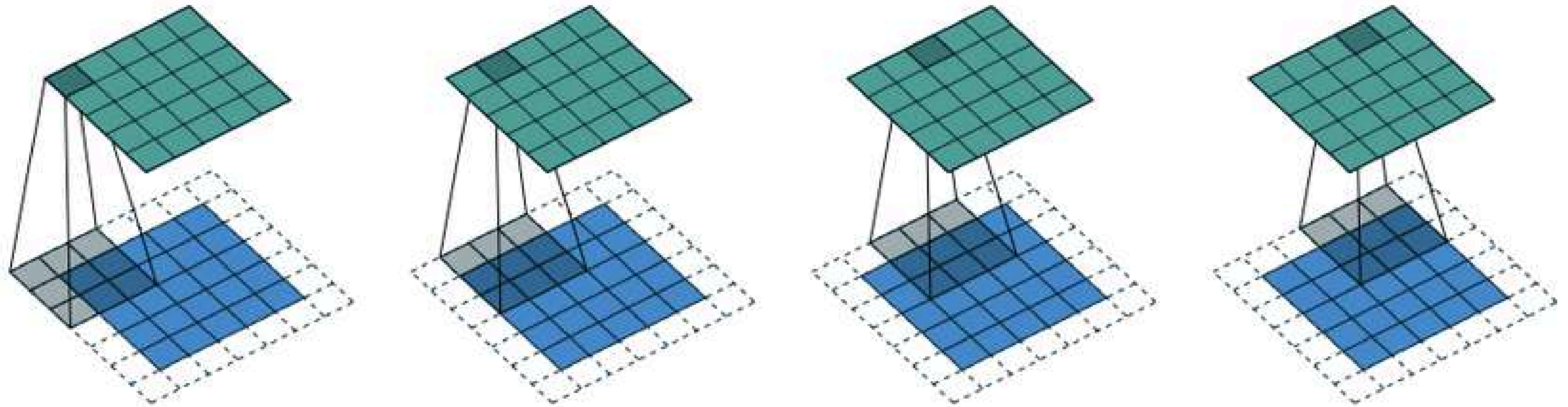
Padding & Stride

03. Padding & Stride

Padding?

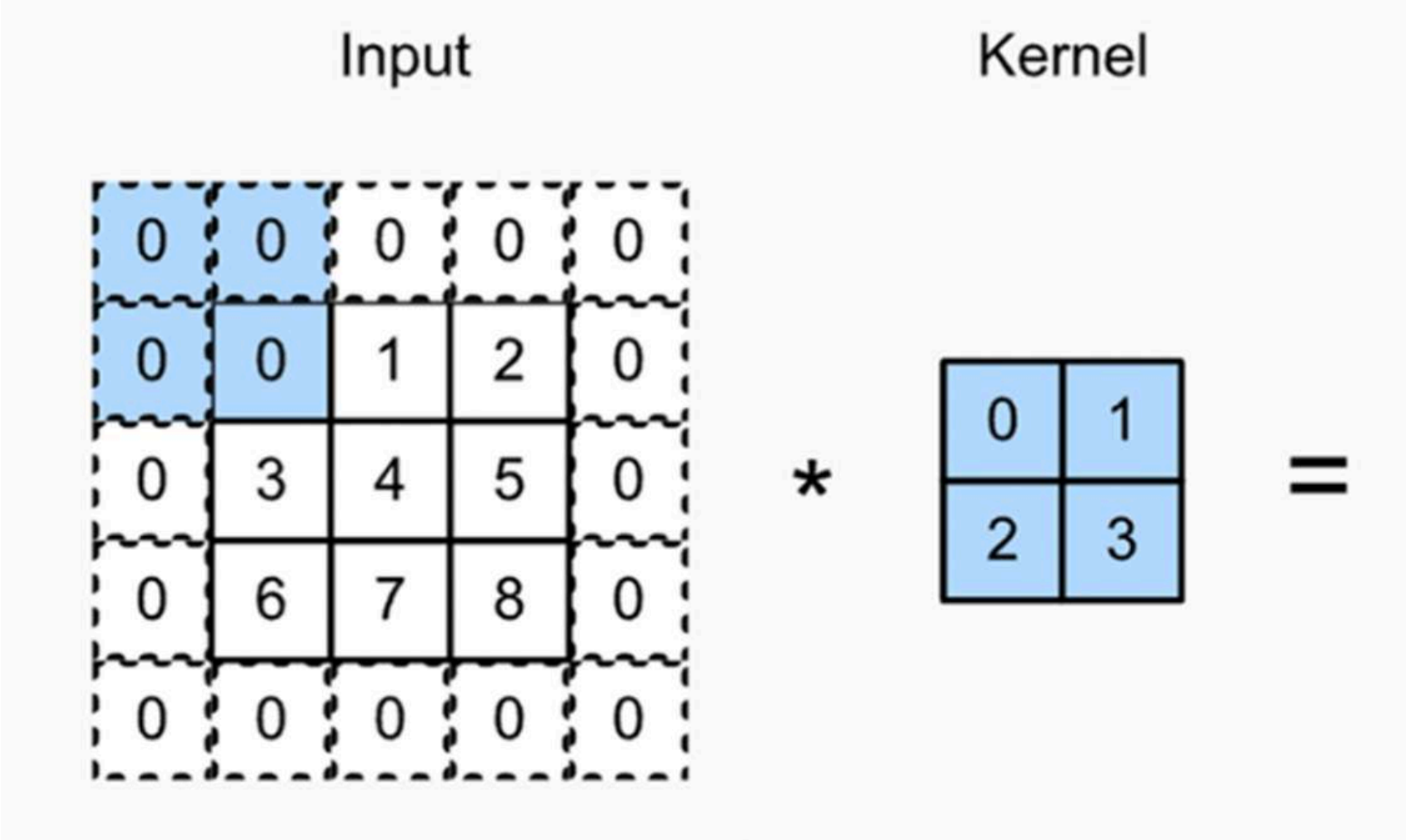


03. Padding & Stride



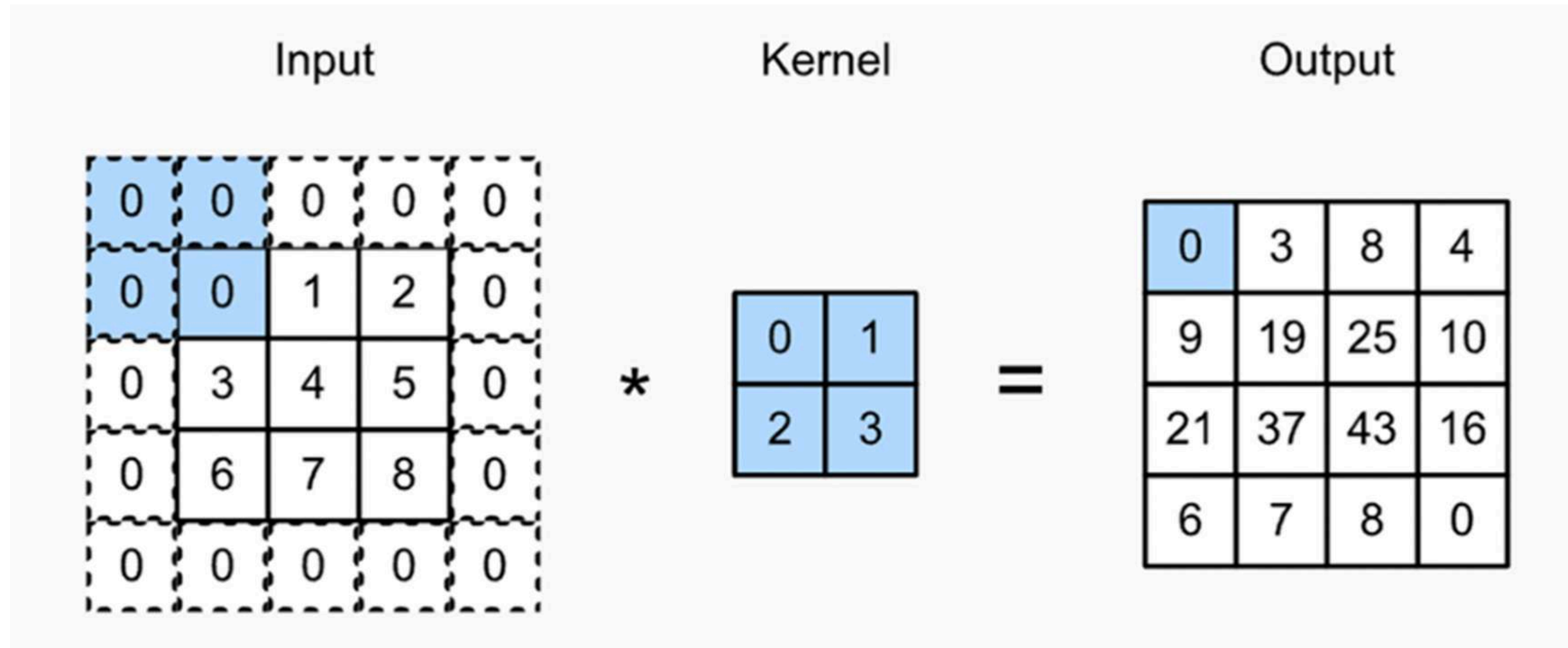
03. Padding & Stride

실습 1



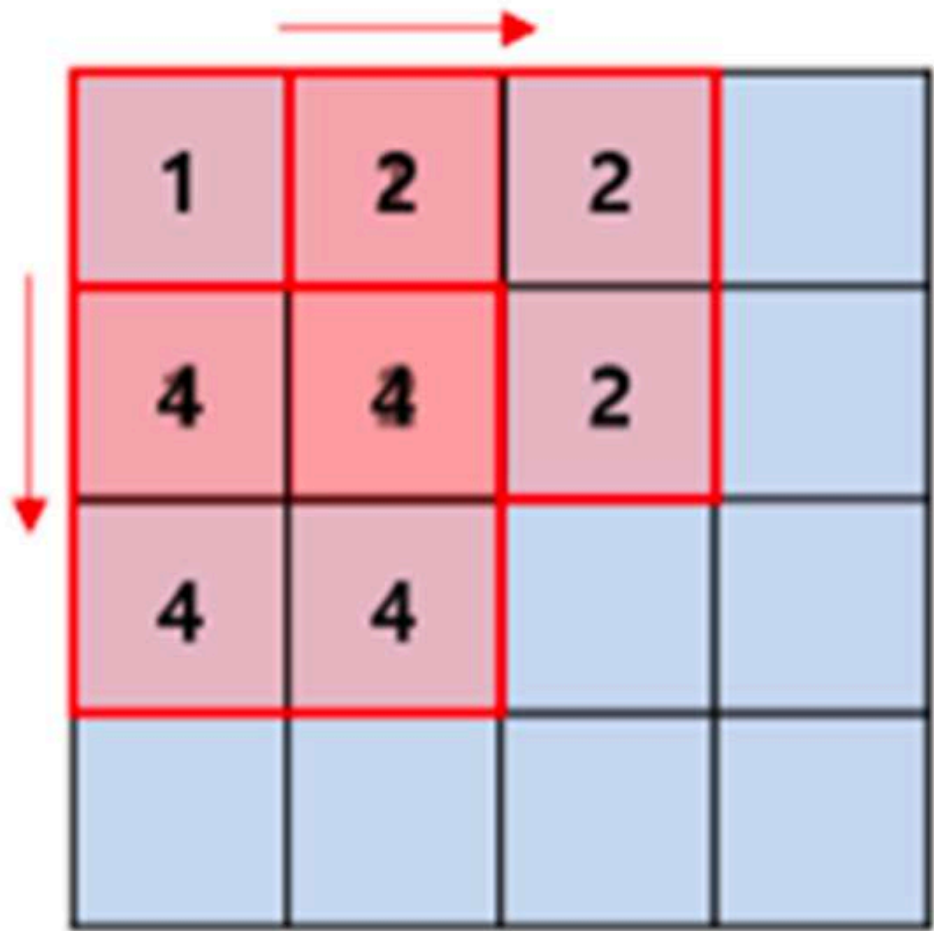
03. Padding & Stride

실습 1

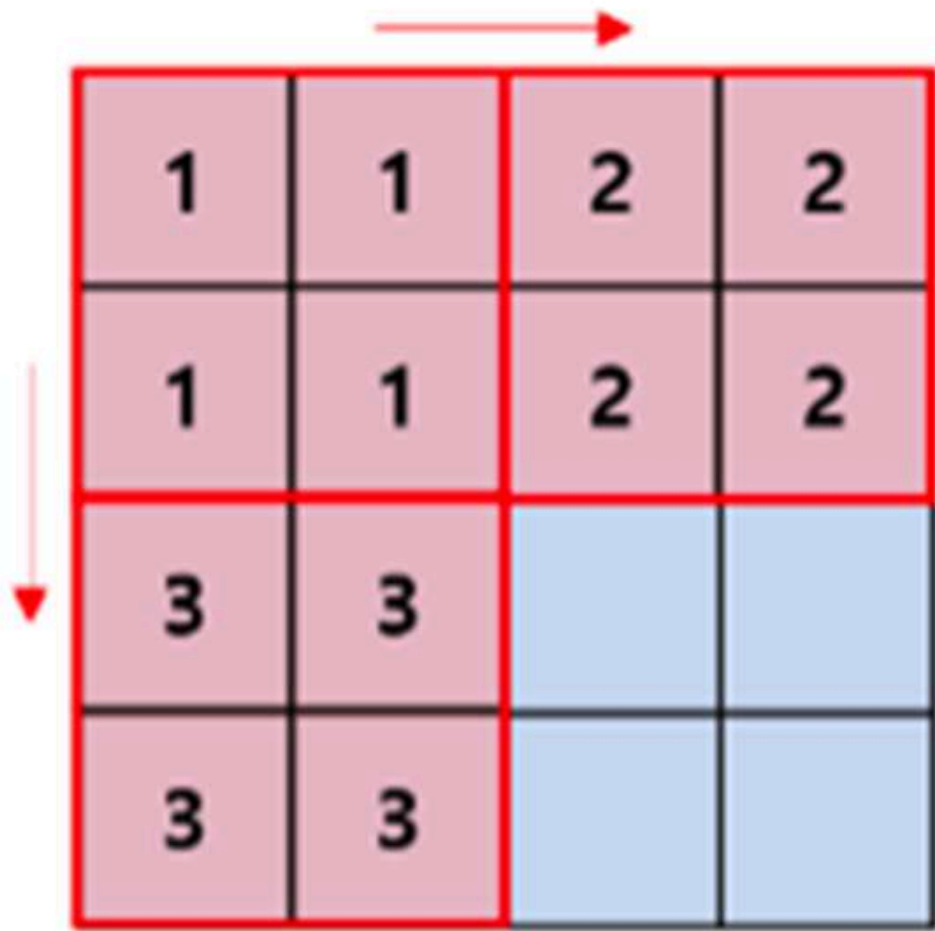


03. Padding & Stride

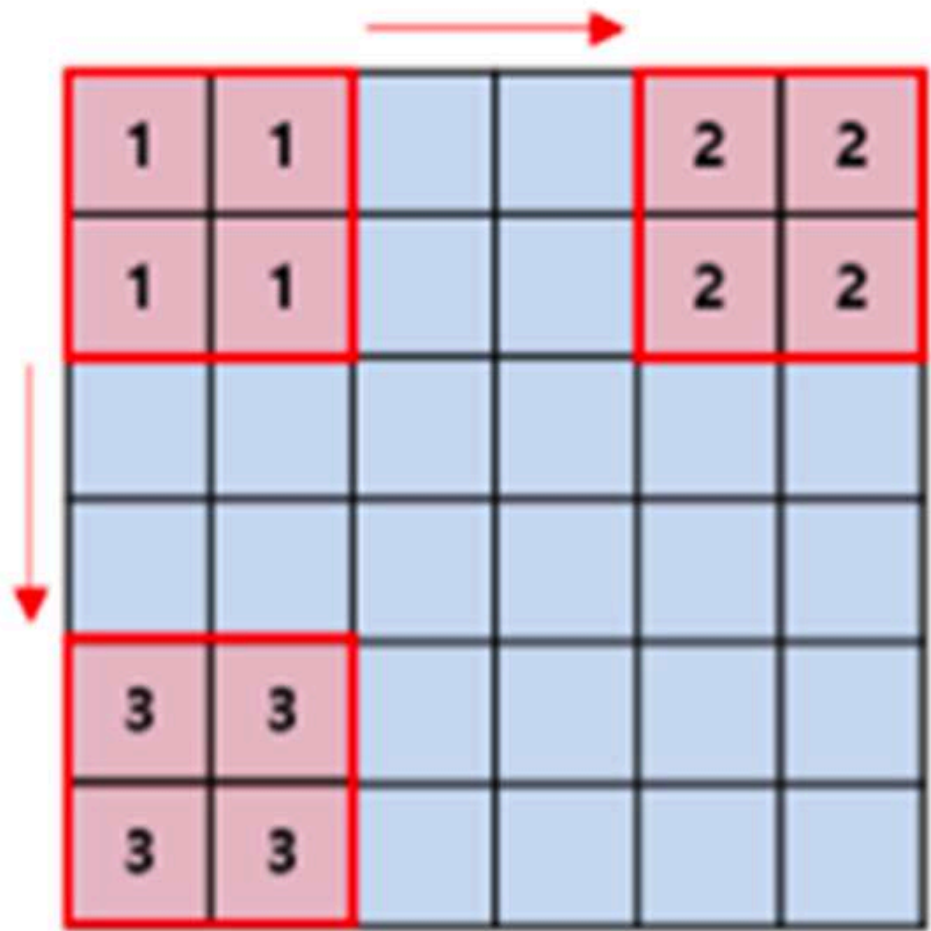
Stride?



[Stride: 1]



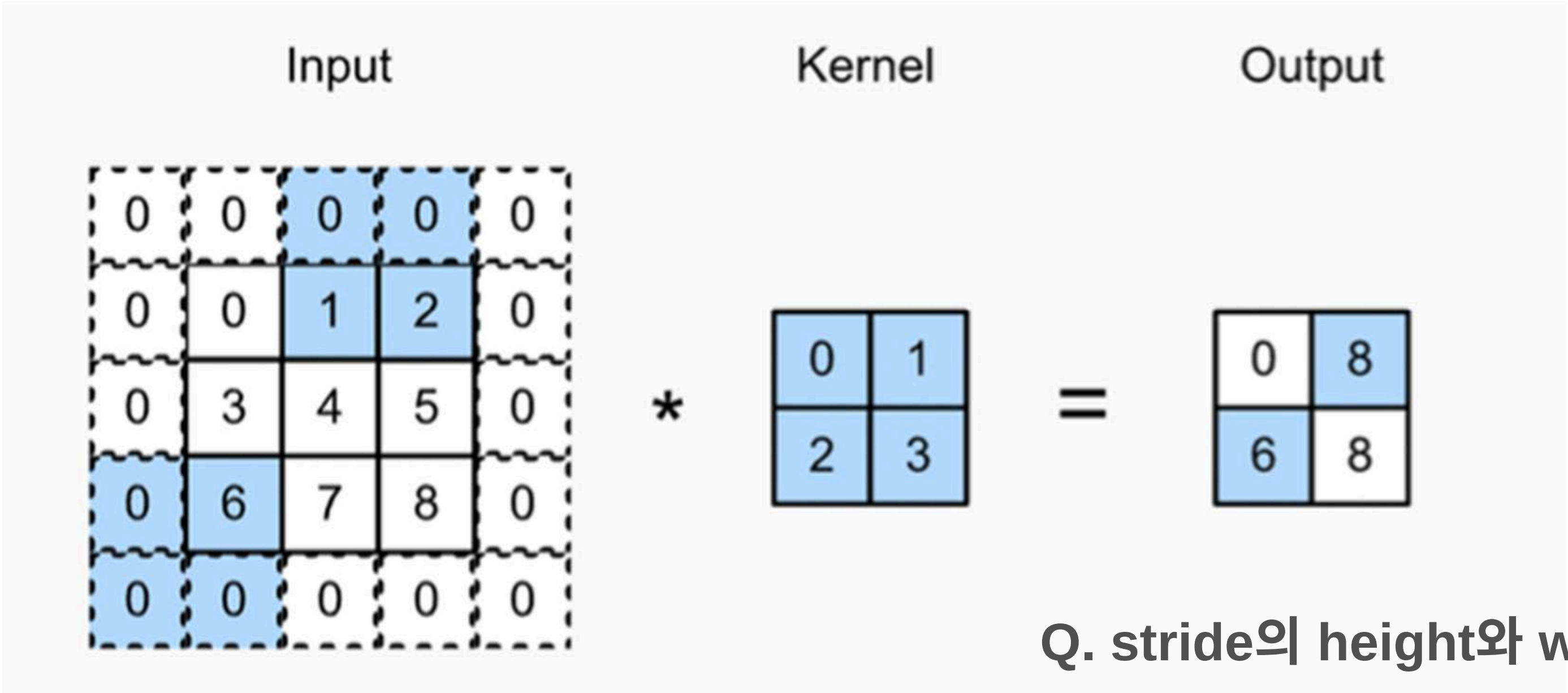
[Stride: 2]



[Stride: 3]

03. Padding & Stride

실습 2



Q. stride의 height와 width는?

03. Padding & Stride

$$O_h = \left\lfloor \frac{I_h - K_h + 2P}{S_h} + 1 \right\rfloor$$
$$O_w = \left\lfloor \frac{I_w - K_w + 2P}{S_w} + 1 \right\rfloor$$

O_h : 출력 이미지의 높이(Height)

O_w : 출력 이미지의 너비(Width)

I_h : 입력 이미지의 높이

I_w : 입력 이미지의 너비

K_h : 컨볼루션 커널의 높이

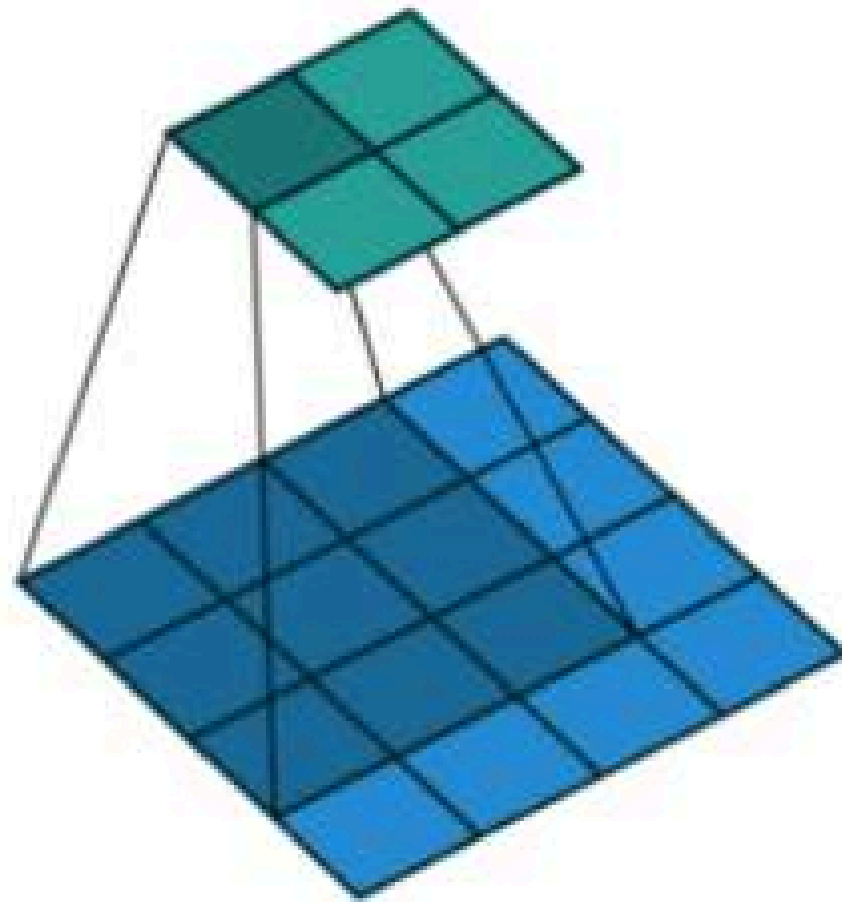
K_w : 컨볼루션 커널의 너비

P : 패딩(padding)의 양

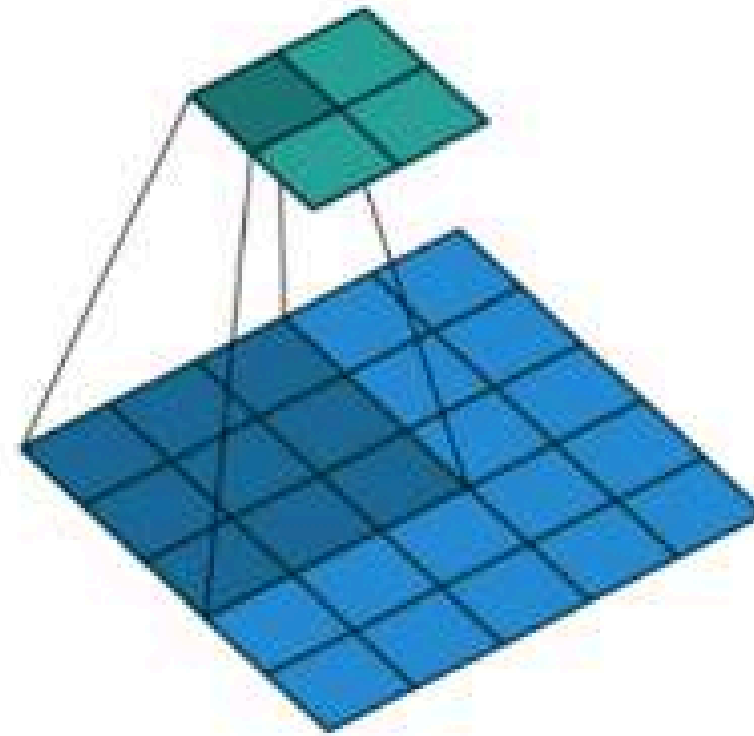
S_h : 스트라이드(stride)의 높이

S_w : 스트라이드(stride)의 너비

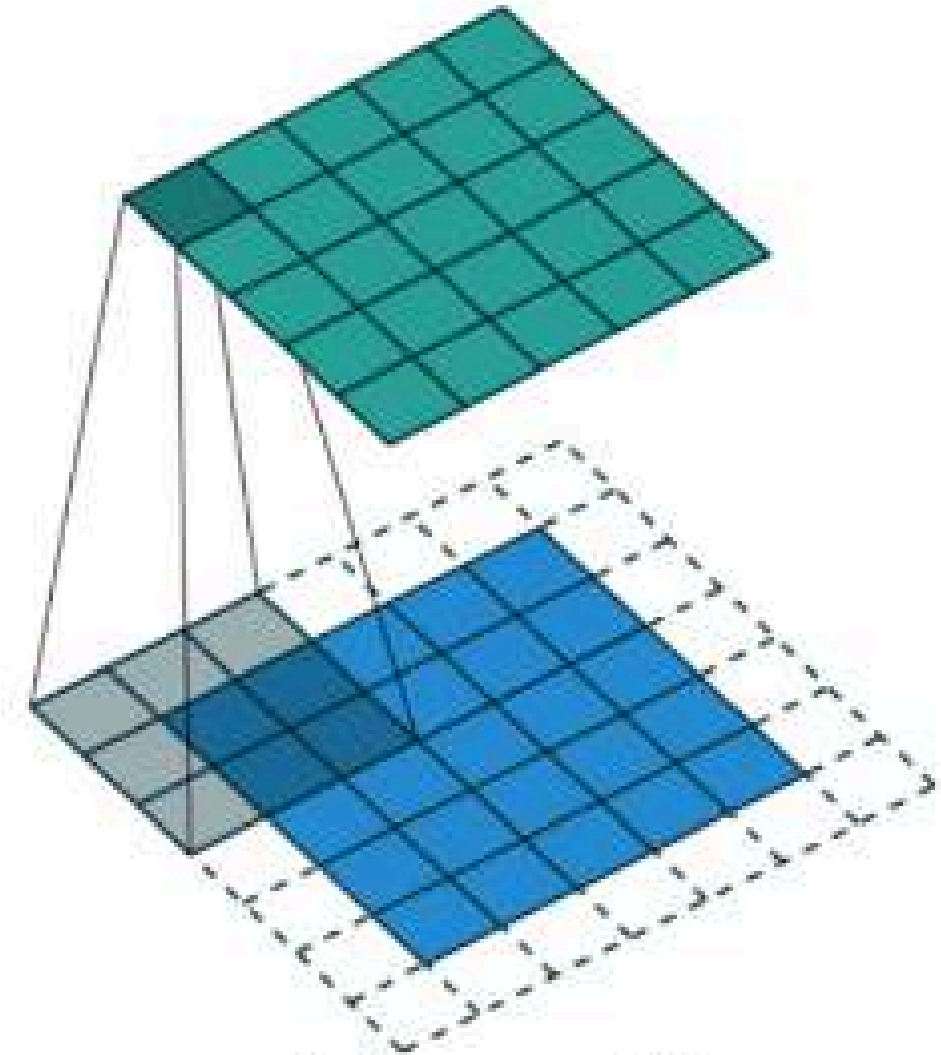
03. Padding & Stride



Vanilla convolution



convolution + stride



convolution + padding

05

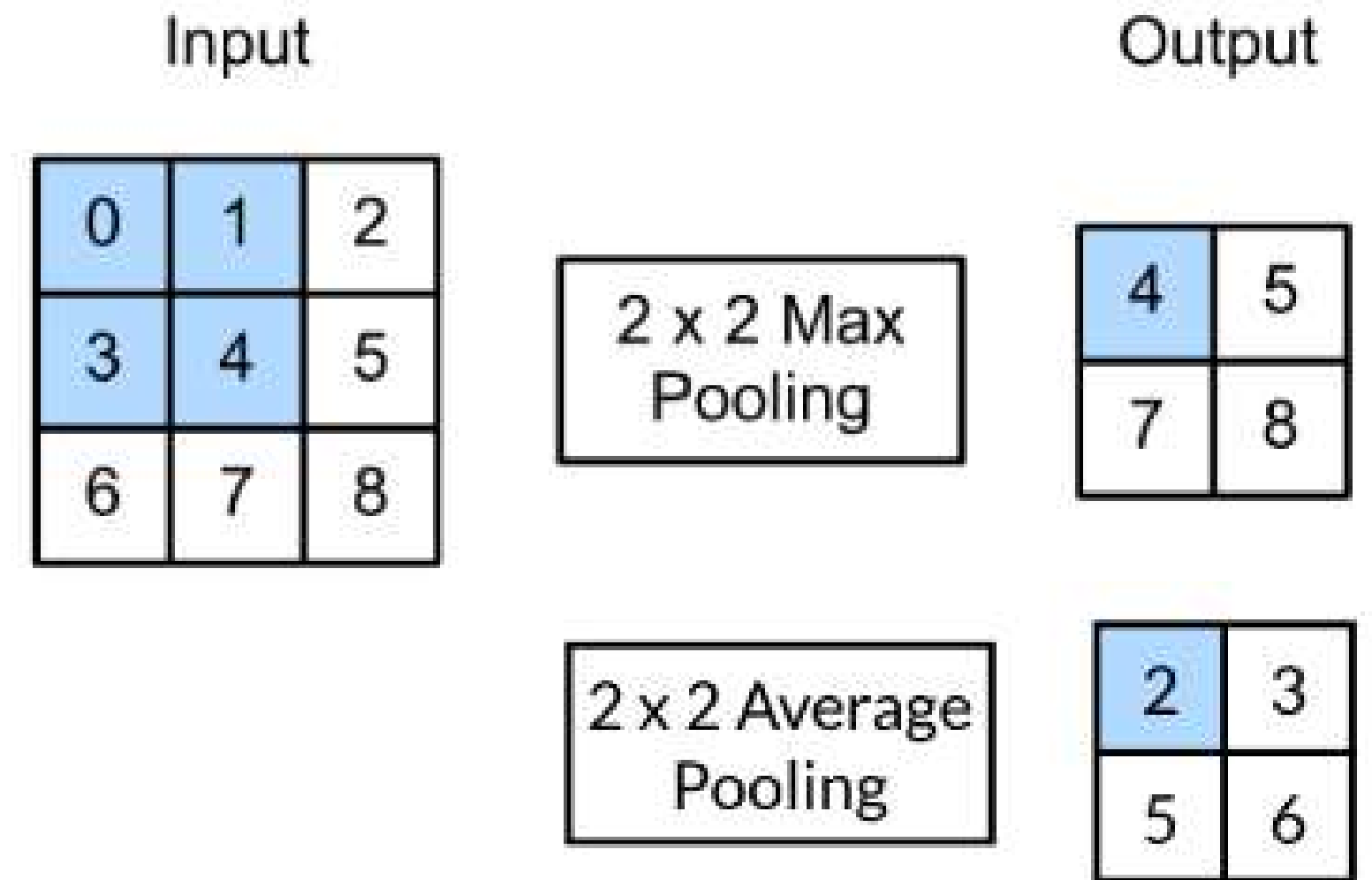
Pooling Layer

05. Pooling Layer

Pooling ?

점진적으로 spatial resolution 을 줄여준다.

- aggregate information
- downsampling
- parameter-free operator
- locally translation invariance

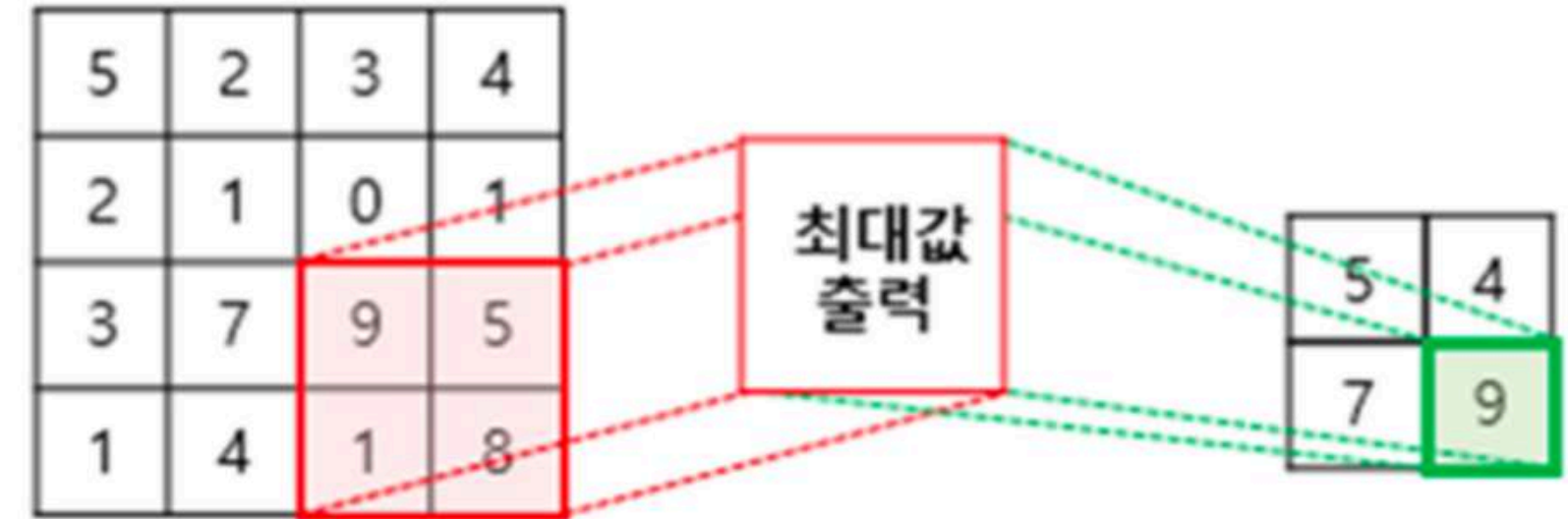
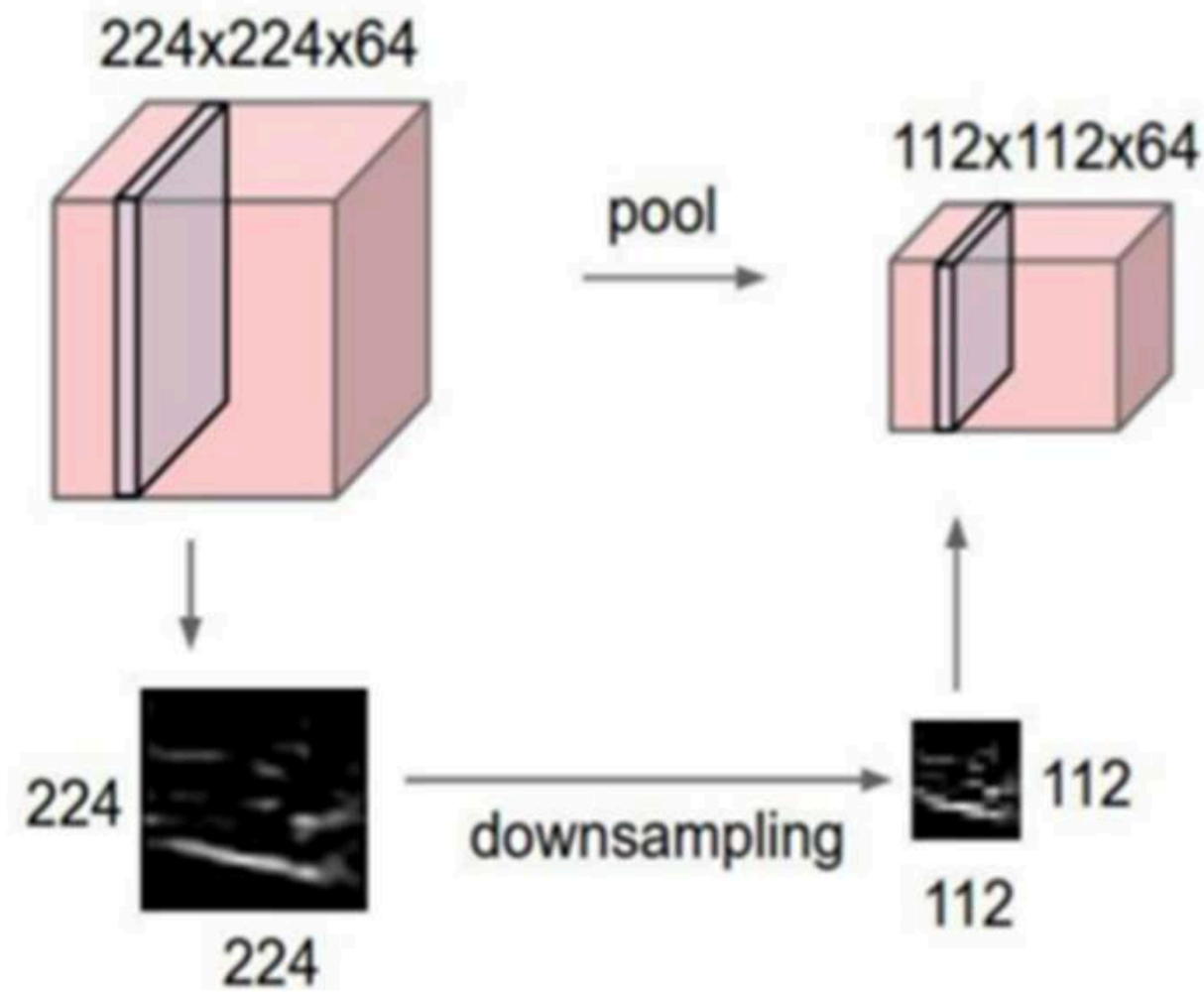


05. Pooling Layer

Pooling ?

점진적으로 spatial resolution 을 줄여준다.

- aggregate information
- downsampling
- parameter-free operator
- locally translation invariance



Pooling을 통해서 차원을 축소

Max Pooling, Mean Pooling

Conv 계산처럼 Stride가 존재

05. Pooling Layer

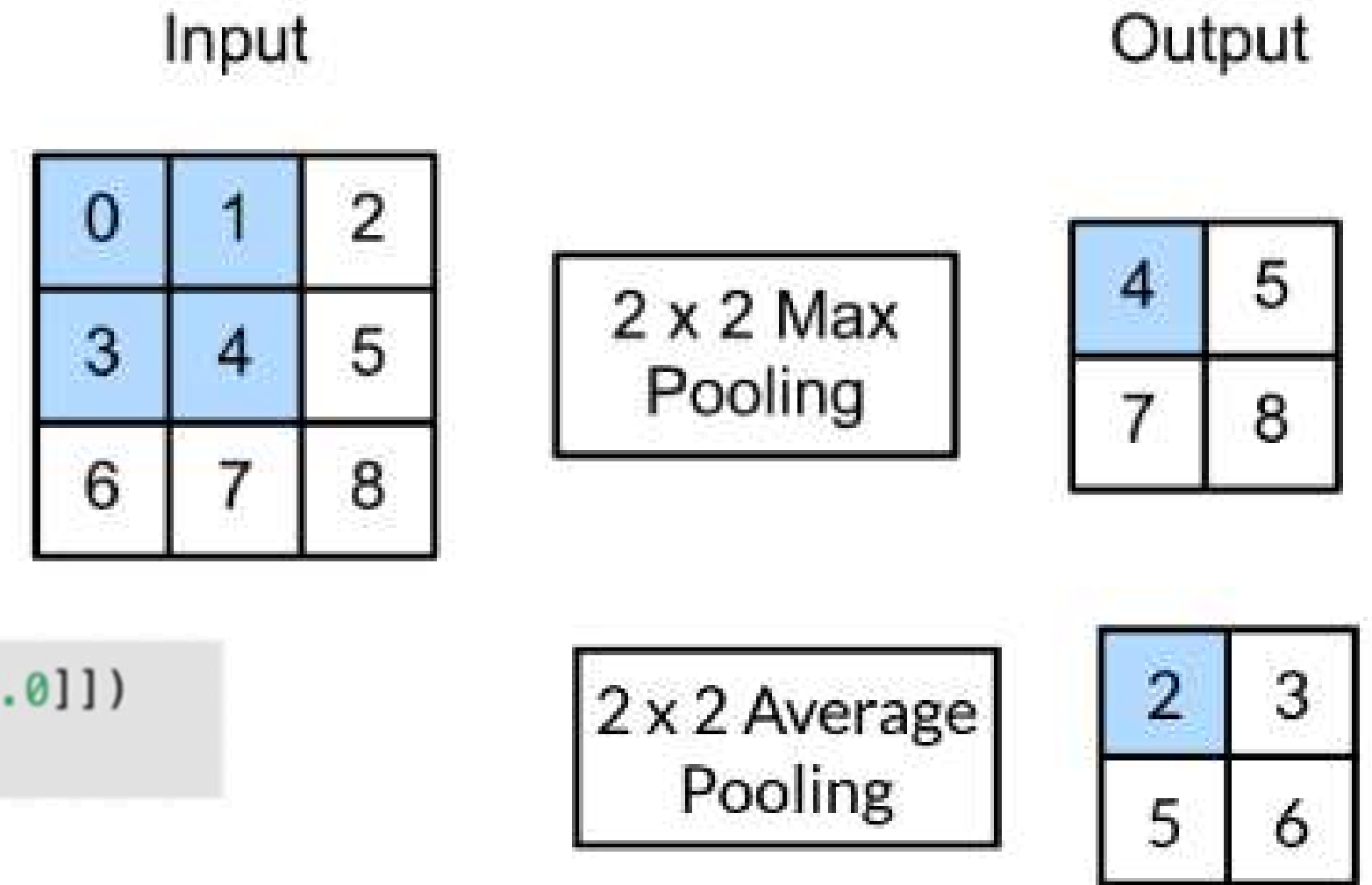
Pooling의 종류

- Maximum pooling

```
X = torch.tensor([[0.0, 1.0, 2.0], [3.0, 4.0, 5.0], [6.0, 7.0, 8.0]])  
pool2d(X, (2, 2))
```

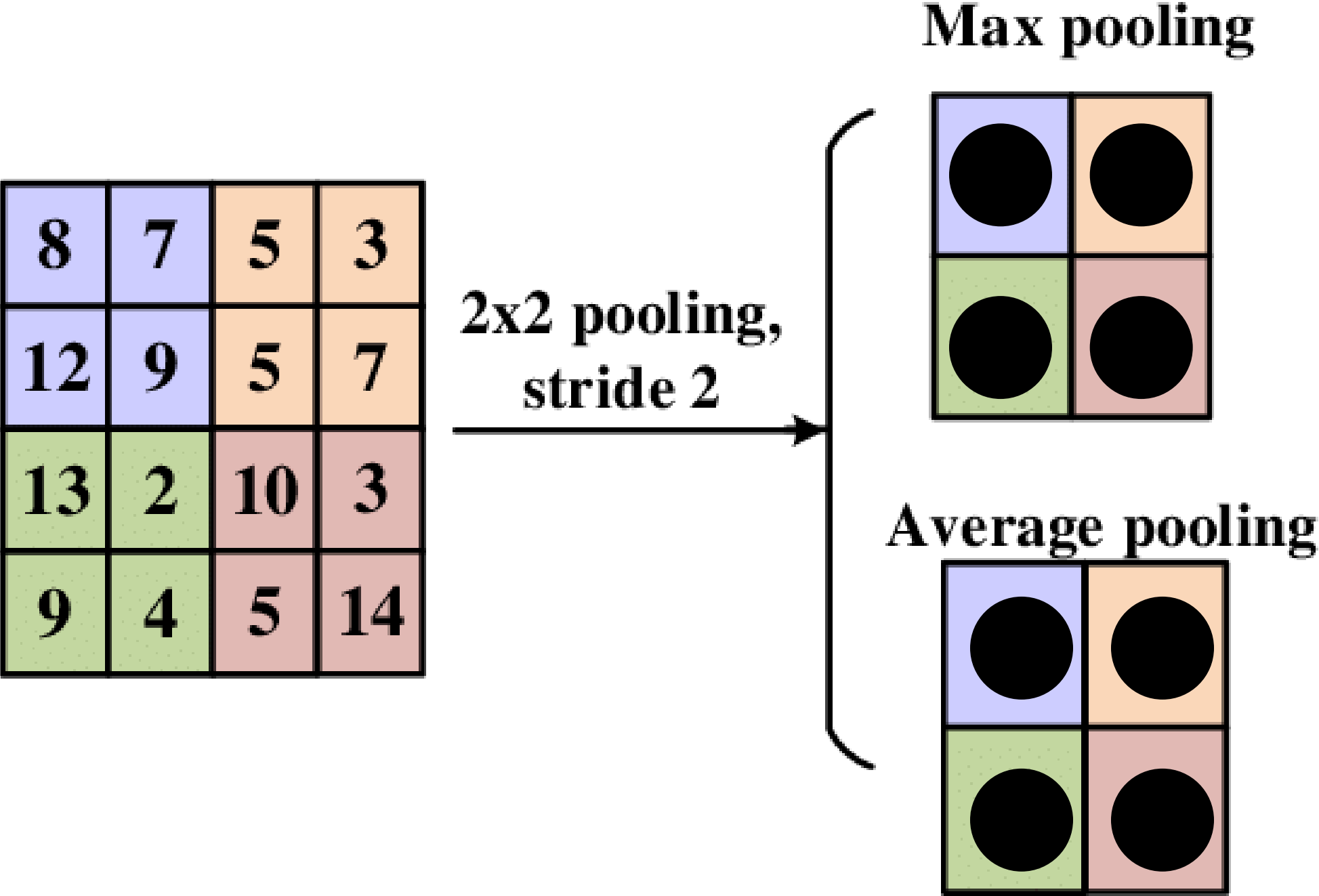
- Average pooling

```
pool2d(X, (2, 2), 'avg')
```



05. Pooling Layer

실습 3



Code Exploration



```
import torch
import torch.nn as nn
import torch.nn.functional as F

class SimpleCNN(nn.Module):
    def __init__(self):
        super(SimpleCNN, self).__init__()

        self.conv1 = nn.Conv2d(in_channels=3, out_channels=6, kernel_size=5, stride=2, padding=2)
        self.pool = nn.MaxPool2d(kernel_size=2, stride=2, padding=0)

    def forward(self, x):
        x = self.conv1(x)
        x = self.pool(x)
        return x
```


MLP vs CNN

파라미터 개수 비교

시나리오

- 입력 이미지의 크기: 32x32 픽셀
- 입력 이미지의 채널 수: 3 (RGB)
- MLP의 hidden layer: 하나, 128 뉴런
- CNN의 convolution layer: 32개의 필터, 각 필터의 크기 3x3

MLP의 파라미터 계산

MLP vs CNN

파라미터 개수 비교

시나리오

- 입력 이미지의 크기: 32x32 픽셀
- 입력 이미지의 채널 수: 3 (RGB)
- MLP의 hidden layer: 하나, 128 뉴런
- CNN의 convolution layer: 32개의 필터, 각 필터의 크기 3x3

CNN의 파라미터 계산

02. MLP vs CNN

Summary

MLP

- 전통적인 신경망 구조로, 입력층, 하나 이상의 은닉층, 그리고 출력층으로 구성된다.
- 각 층의 모든 뉴런은 이전 층의 모든 뉴런과 연결되어 있다.
- 이미지와 같은 고차원 데이터에서는 매개변수의 수가 매우 많아질 수 있어 비효율적일 수 있다.

CNN

- 이미지 처리에 최적화된 신경망 구조로, 합성곱층과 풀링층을 통해 특징을 추출한다.
- 가중치 공유를 통해 훨씬 적은 수의 매개변수를 사용한다.
- 이미지의 공간적 계층 구조를 인식할 수 있어 이미지 인식과 분류에 매우 효과적이다.

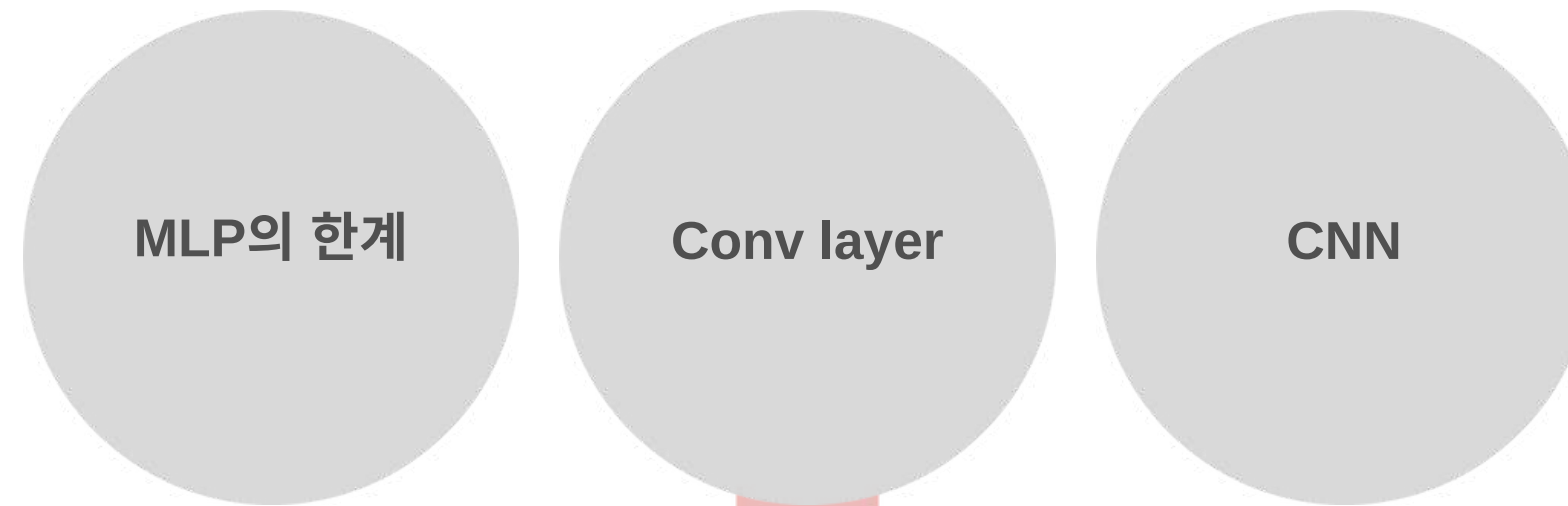
06

다음 주차 예고

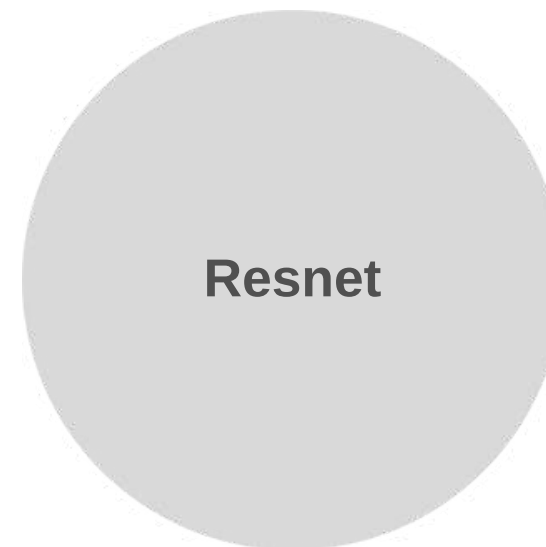


06. 다음주차 예고

5주차 세션 Key Word



6주차 세션의 Key Word



06. 다음주차 예고

5주차 과제

공부 과제



코드 과제



Thank You

2024 Summer DL Session 5주차 끝!



KOREA
UNIVERSITY