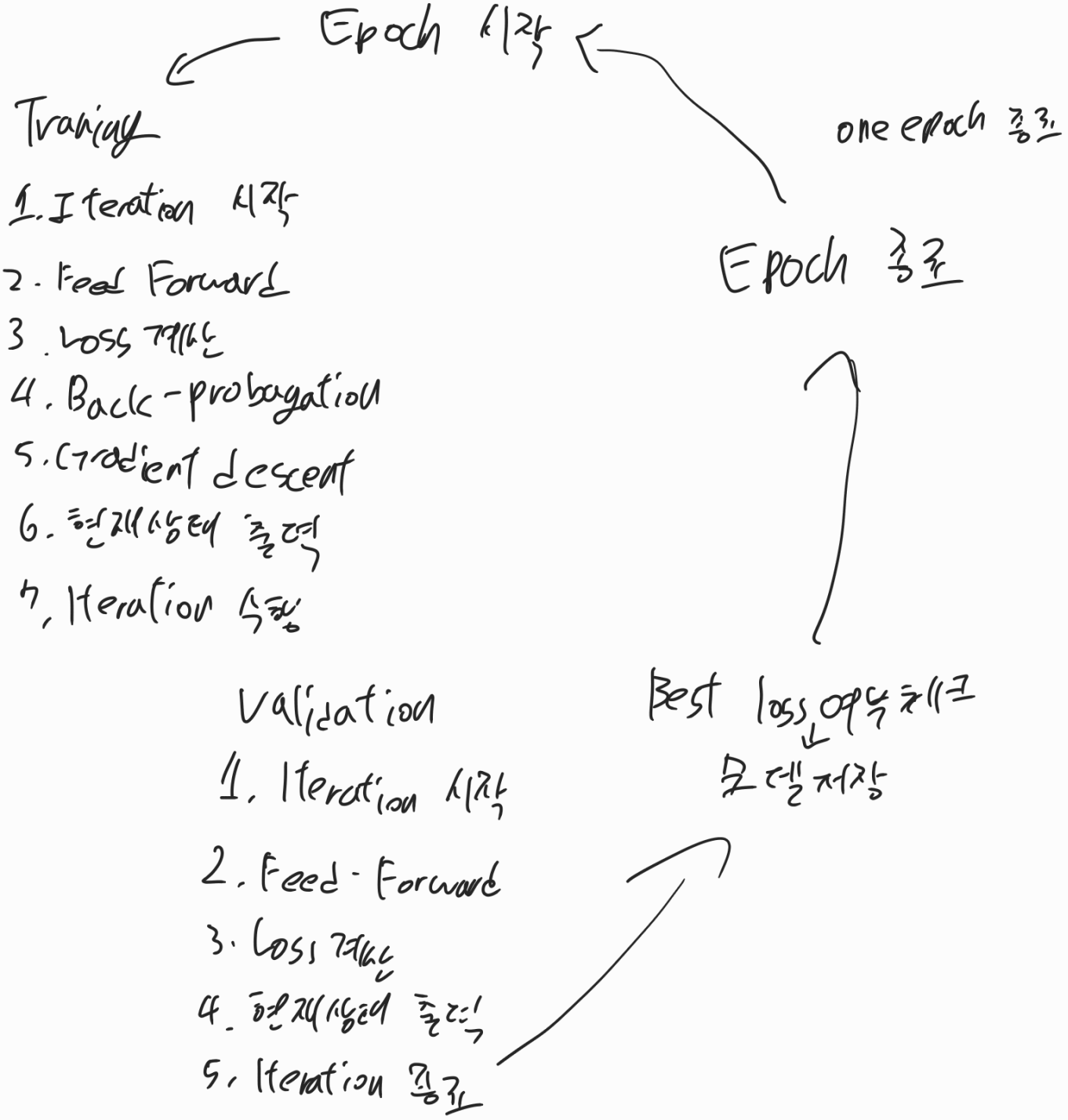
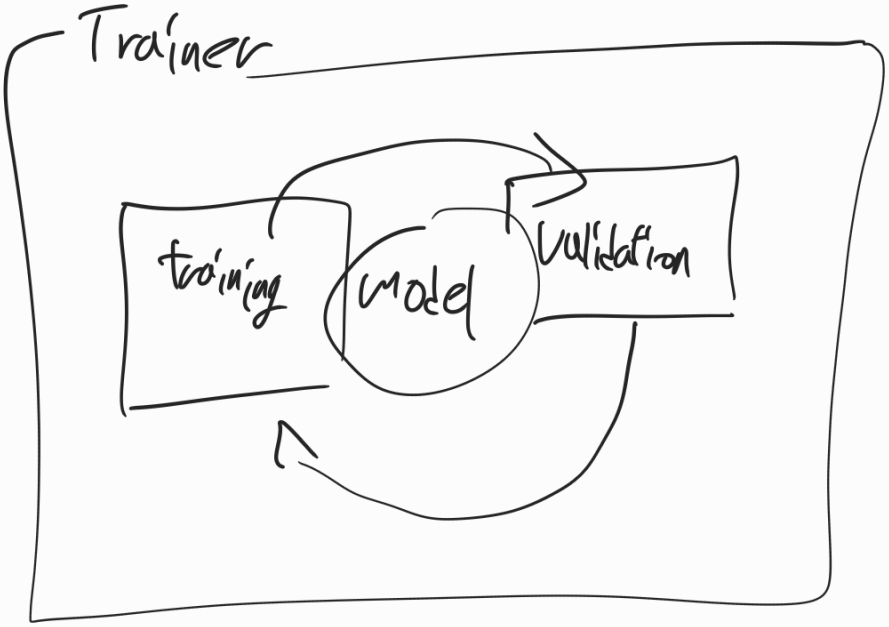
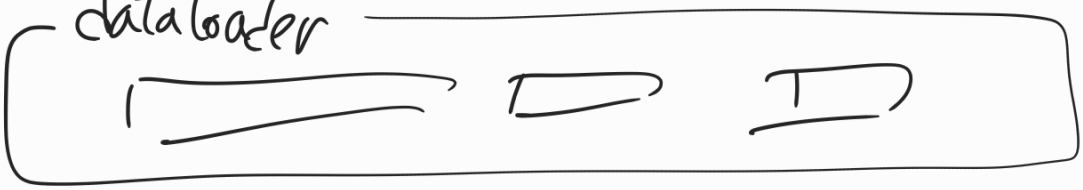
Boilerplate

- 템플릿 같은 느낌 ...?
- 학습 Scheme이 비슷한 경우 모델과 dataset을 제외한 코드를 동일







위의 것들을 Pytorch Lightning을 통해 제공 +알파

pros
- pytorch 공식 라이브러리
- Call-back 함수를 위한 event가 잘 정의되어 있다.
- 손쉽게 학습 통제 logging

cons
- 진입장벽 존재
- 기본 설계에 이슈들

tensorflow model.fit 느낌인가?

dezero.engine.Engine

↙ for문을 수행

train

1. Iteration 시작
2. Feed Forward
3. Loss 계산
4. Back-propagation
5. Gradient descent
6. 현재상태 출력
7. Iteration 수행

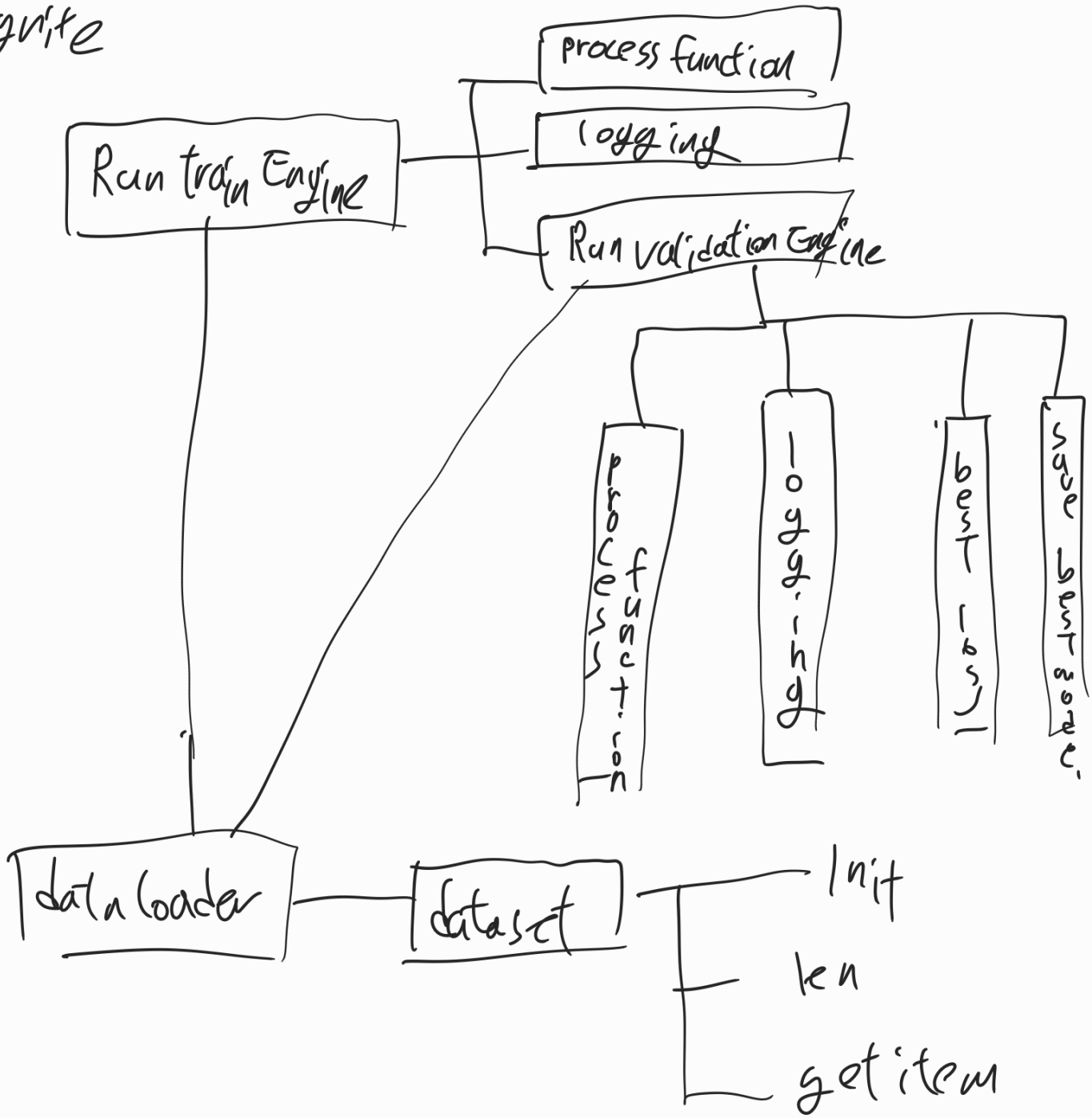
process function
사용자 코딩

process function return 값을 활용
(Logging)

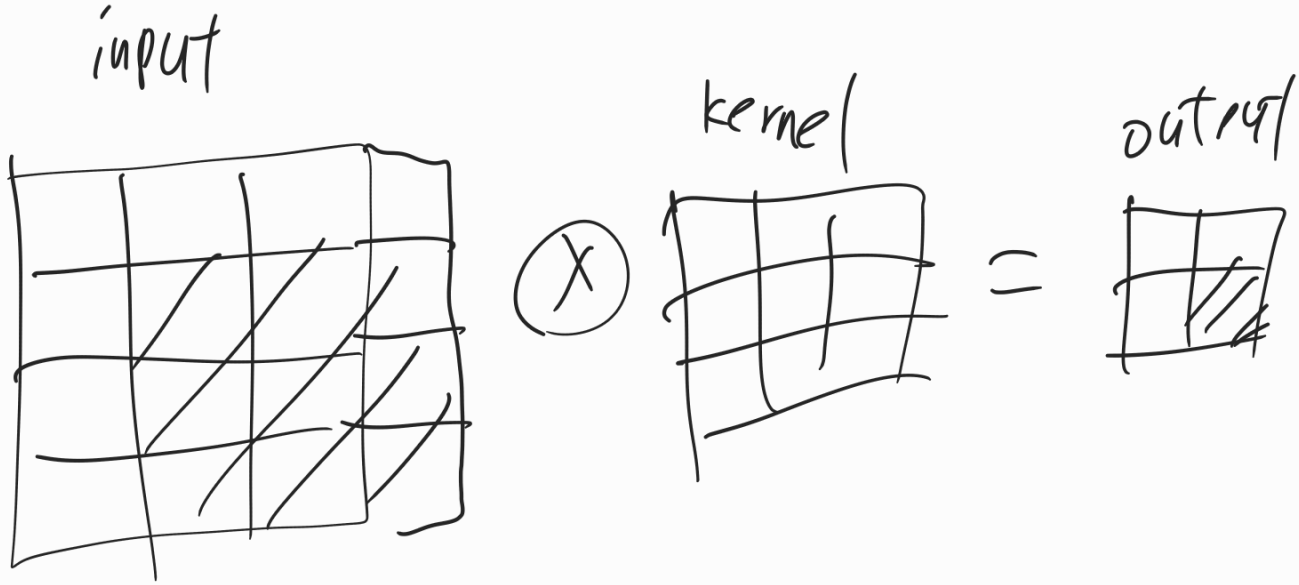
Add event handler

- decorator
- Add event handler

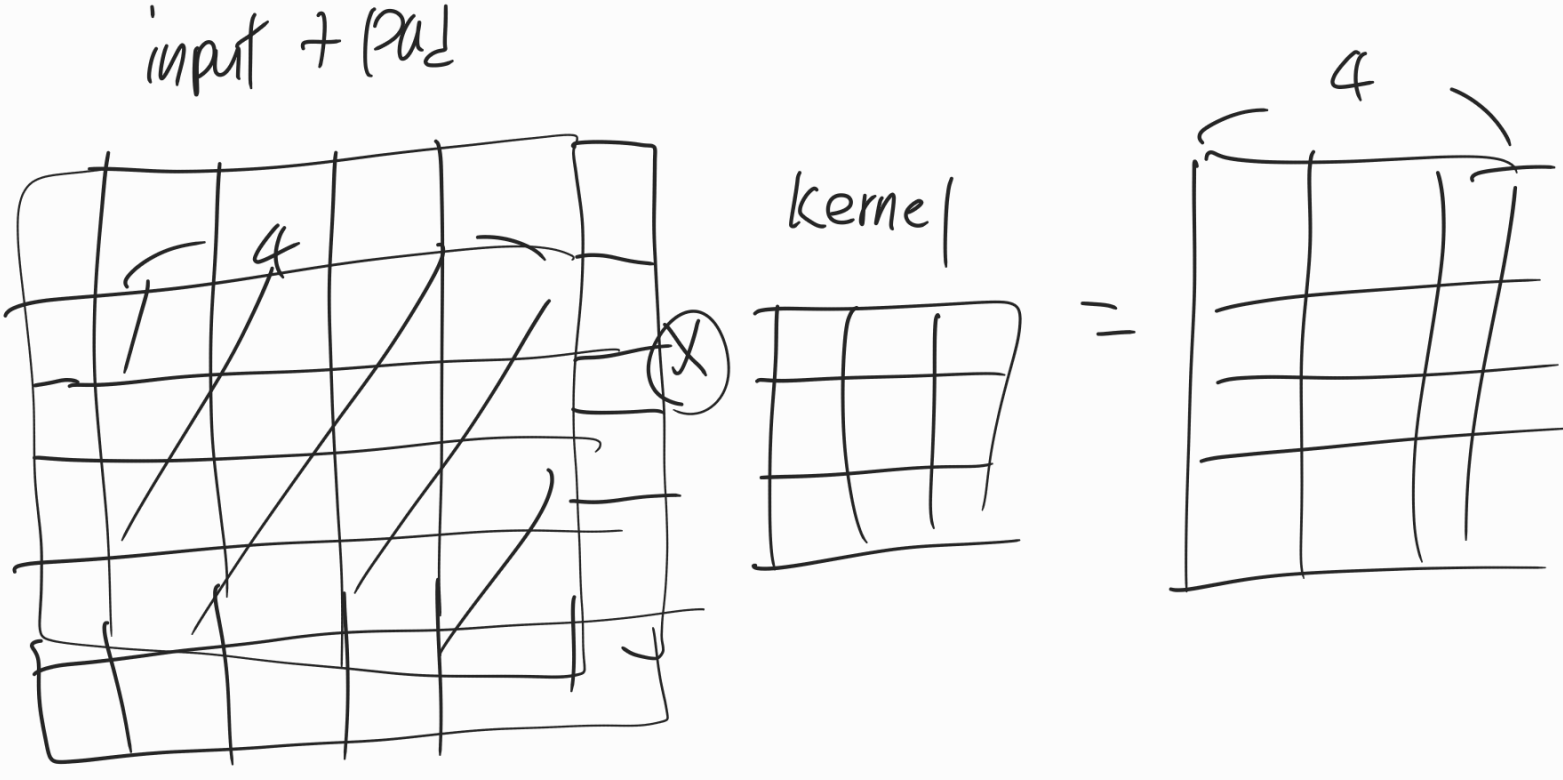
Ignite

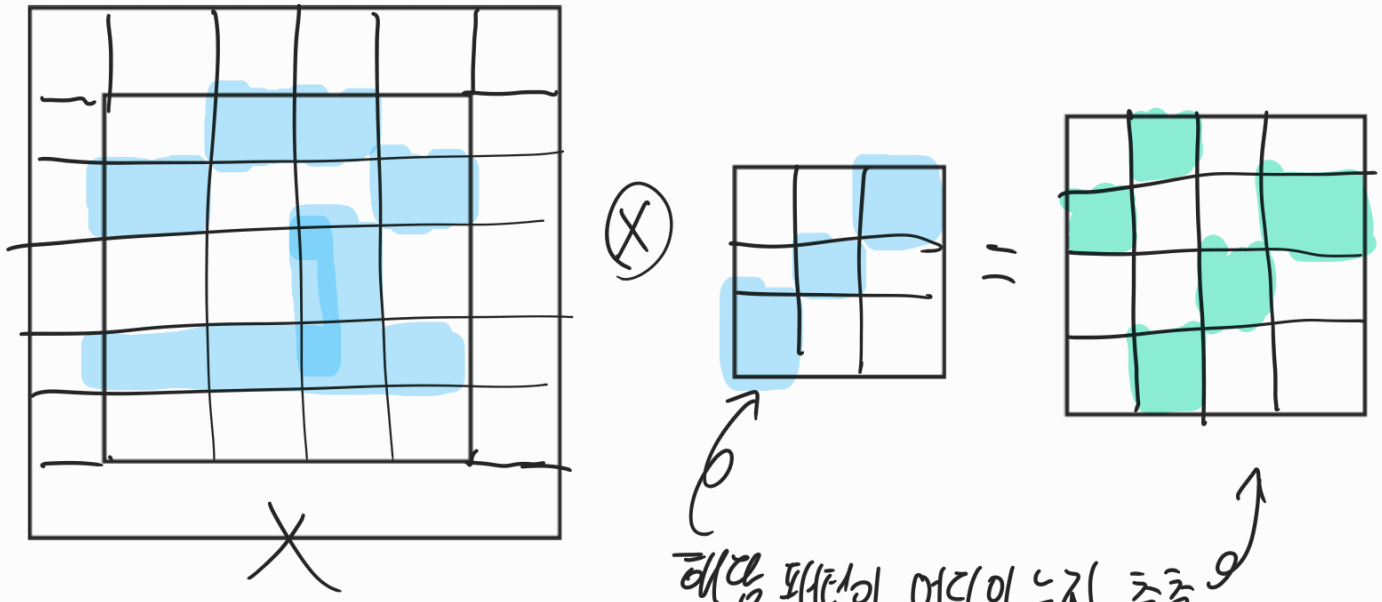

process function
logging
Run train Engine
Run validation Engine
process function
logging
best loss
save best model
data loader
dataset
init
len
getitem

CNN - Convolution Neural Network

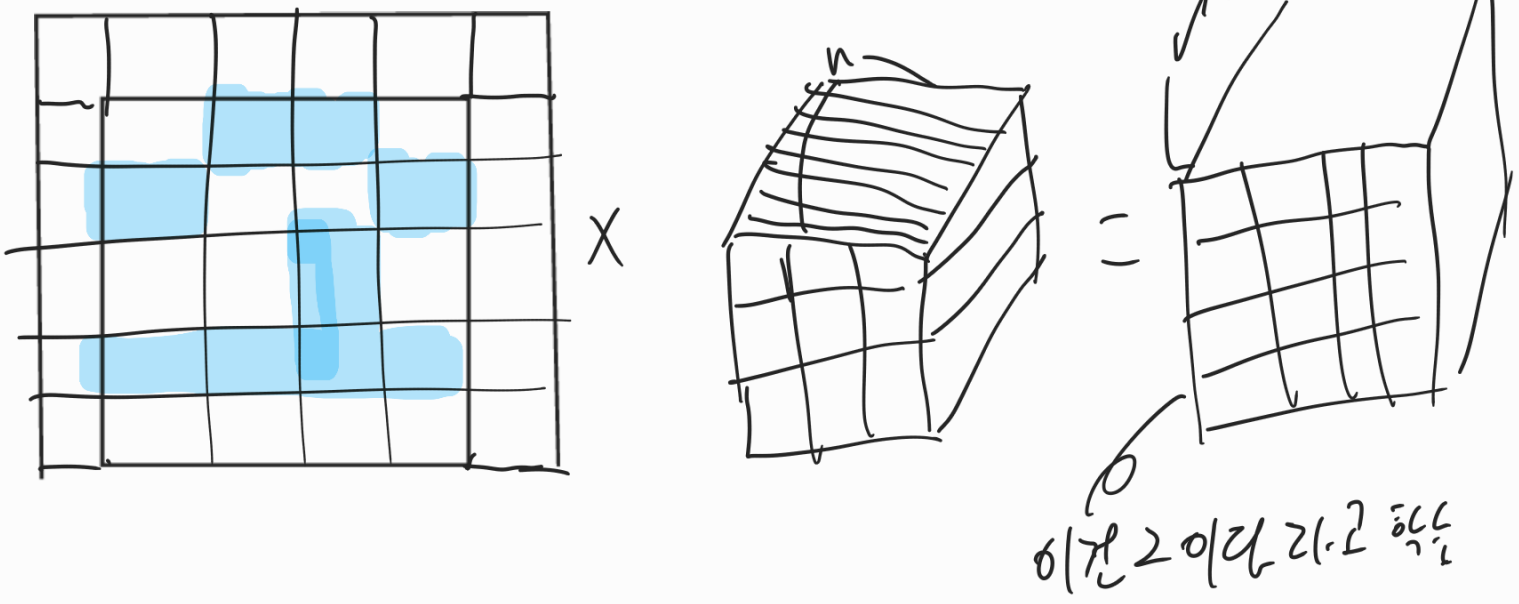


Padding





여러겹이 여러개



Input tensor shape [batchsize, channel, height, width]

greyscale channel = 1

RGB channel = 3

입출력 shape 구하기

b = batch_size

(xheight, xwidth) = input size

Cin = input channels

Cout = output channels

(kheight, kwidth) = kernel size

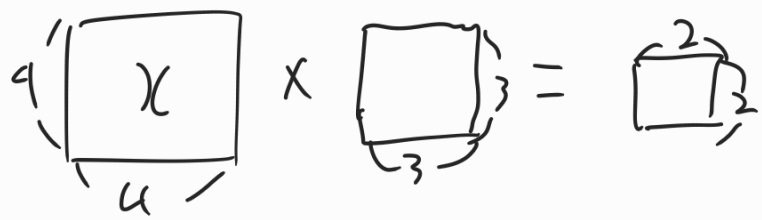
(yheight, ywidth) = output size

$$y = conv2d(x)$$

|x| = (b, Cin, xheight, xwidth)

|y| = (yheight, ywidth)

= (b, Cin, xheight - kheight + 1, xwidth - kwidth + 1)



tip 3x3 kernel 에다가 padding - input 을 입력의 크기로

CNN 특징

- feature의 위치에 구애받지 않는다.
- FC layer에 비해 적은 weight를 가진다.
- 병렬계산 구성이 쉬우므로 GPU에서 연산이 매우 빠르다.
- 입출력계산의 개수라서 네트워크 구성이 쉽지 않다.

NLP 및 음성 시계열 분야에도 사용가능

Dimension Reduction

고차원 공간의 sparse한 데이터를 저차원 공간에 mapping

왜냐하면 보통하는 것

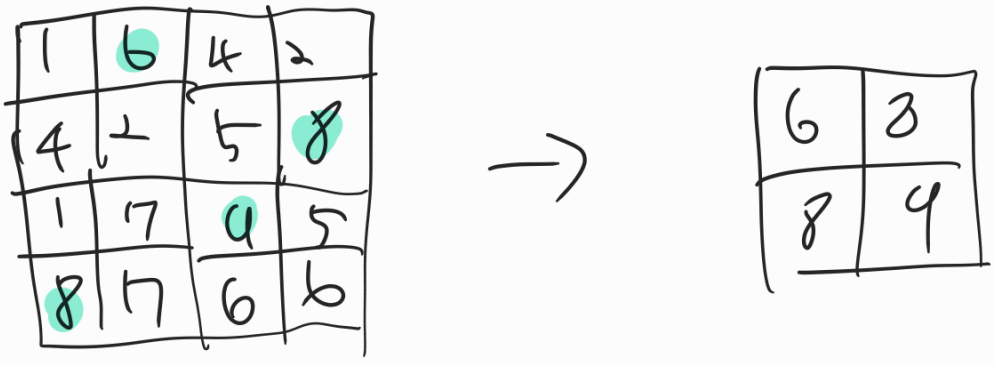
하지만 CNN을 활용한

3X3 + 1padding conv layer 입출력 tensor의 크기가 같음

이러한 Dimension Reduction 기술

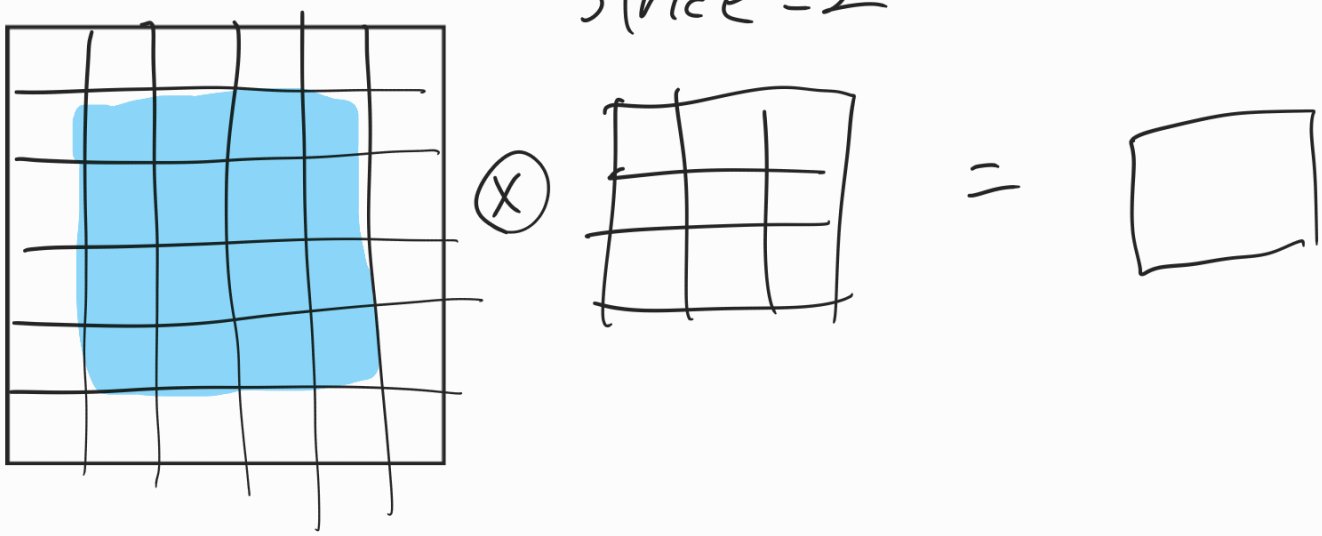
- pooling layer
- stride

Max pooling 3X3 + (pad) + 2X2 max pool



Stride

· 이동하는 거리



Stride가 같은 conv layer 내에서 간단히 동작 원리를 그림에 좋게 예제

(Pheight, Pwidth) = Padd size

(Sheight, Swidth) = Stride size

$$= (b, C_{in}, \frac{x_{height} + 2 \times P_{height} - (k_{height}-1) - 1}{S_{height}}, \frac{x_{width} + 2 \times P_{width} - (k_{width}-1) - 1}{S_{width}}$$