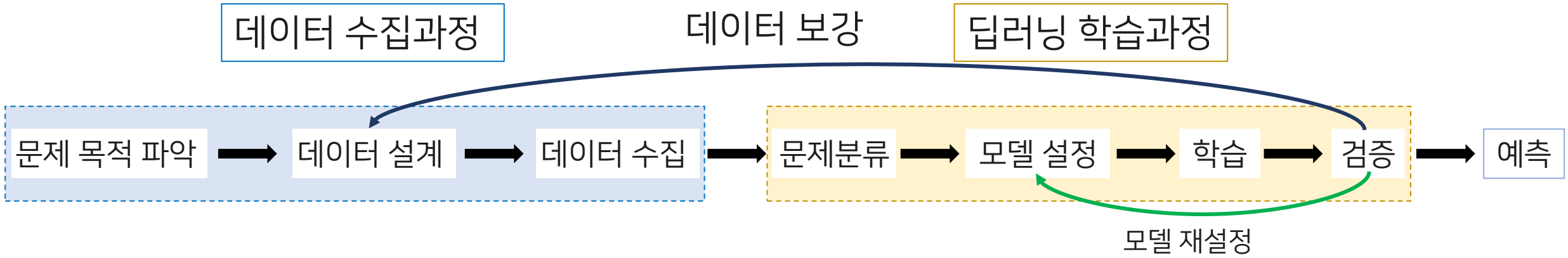
Contents

2장 딥러닝 흐름잡기

머신러닝 과정


데이터 수집과정
데이터 보강
딥러닝 학습과정
문제 목적 파악
데이터 설계
데이터 수집
문제분류
모델 설정
학습
검증
예측
모델 재설정

Contents

2장 딥러닝 흐름잡기

이미지 데이터셋 : 이미지데이터+레이블

분류 레이블이 주어진 이미지 데이터셋

3채널

Data — 1280x800x3 정수 데이터(0~255)

Label — 0: dog, 1: cat, 2: truck, 3: plane, ···

➡ 1 ➡ Cat

이미지 분류 [Image Classification]

Computer Vision의 핵심 작업

CIFAR10 데이터셋

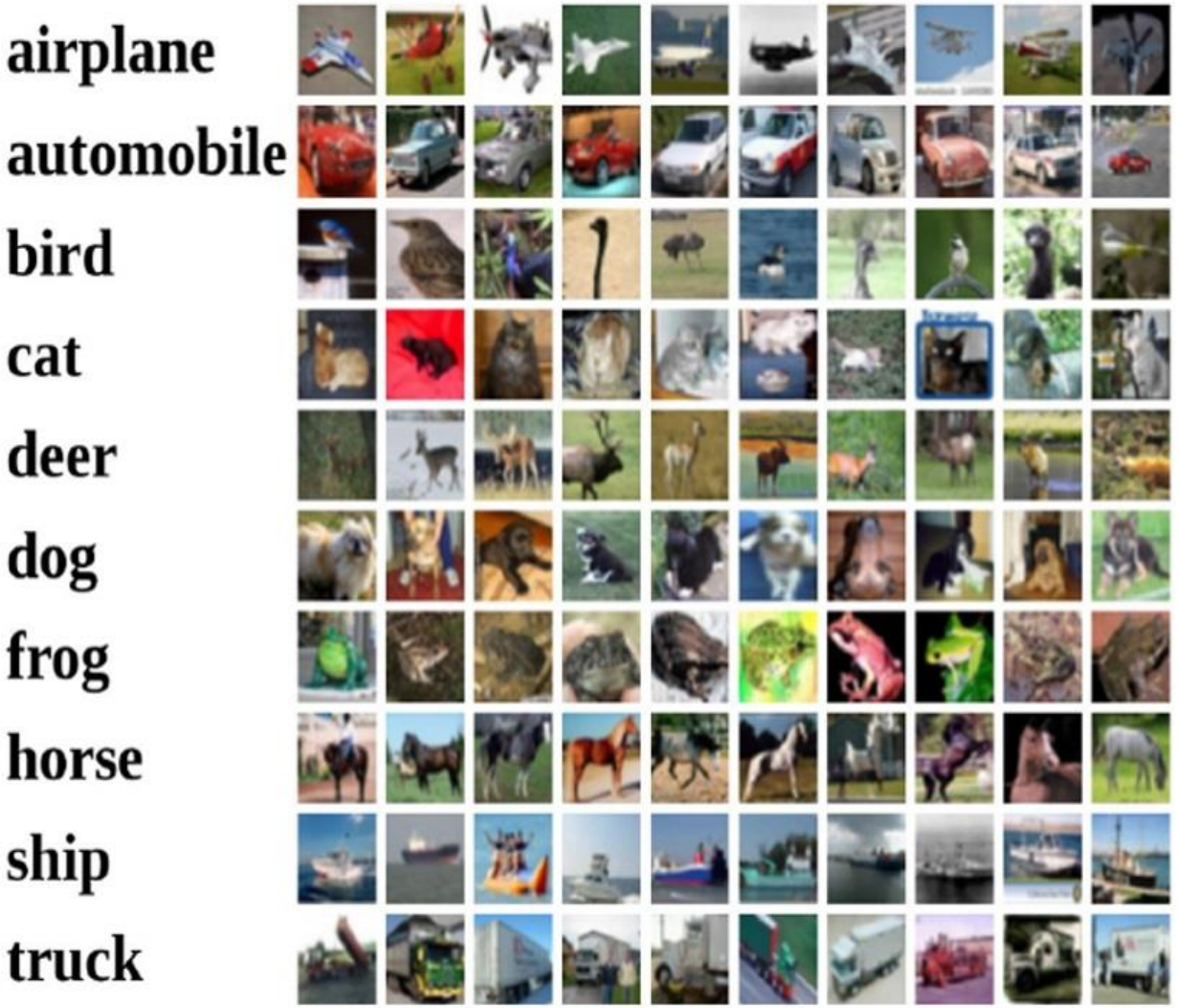
Class

10

데이터 개수

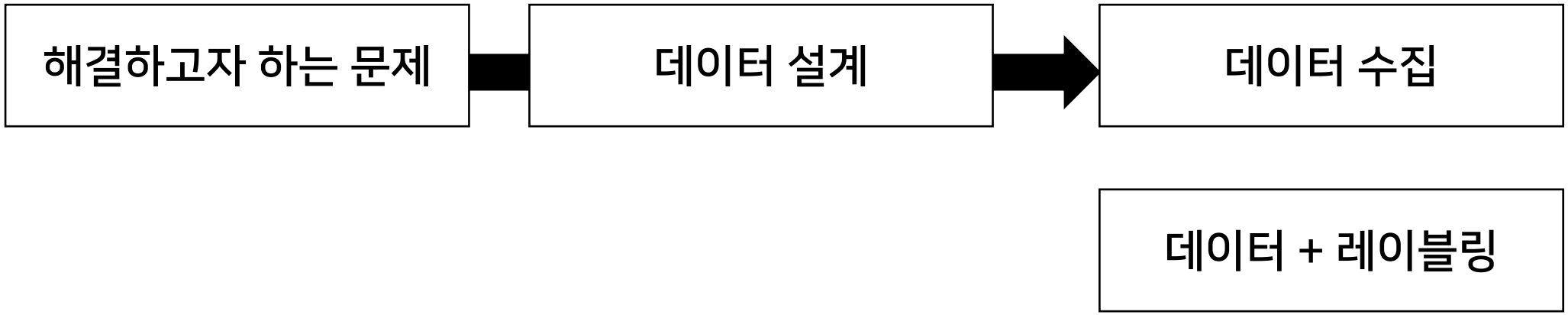
훈련용 : 50000

검증용 : 10000



Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", Technical Report, 2009.

데이터 설계 및 수집

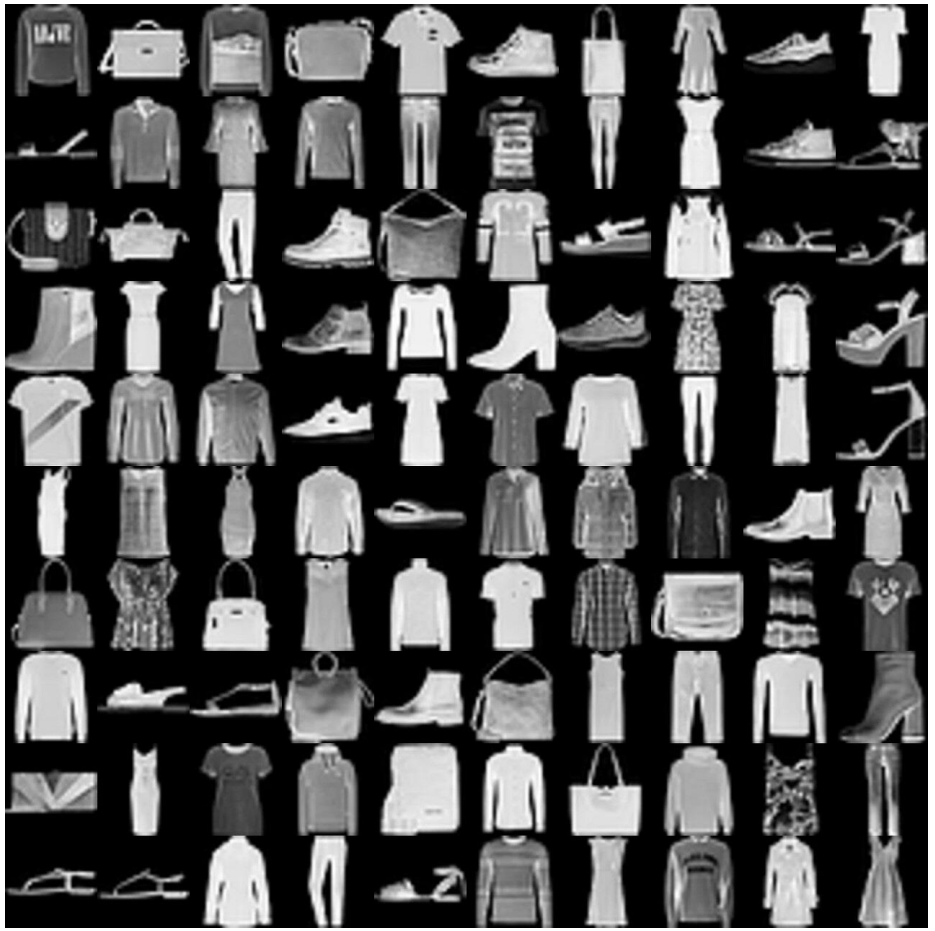


데이터 종류

	x1	x2	x3	x4	y
0	5.1	3.5	1.4	0.2	0
1	4.9	3.0	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	4.6	3.1	1.5	0.2	0
4	5.0	3.6	1.4	0.2	0
...	...	...	...	...	...
145	6.7	3.0	5.2	2.3	2
146	6.3	2.5	5.0	1.9	2
147	6.5	3.0	5.2	2.0	2
148	6.2	3.4	5.4	2.3	2
149	5.9	3.0	5.1	1.8	2

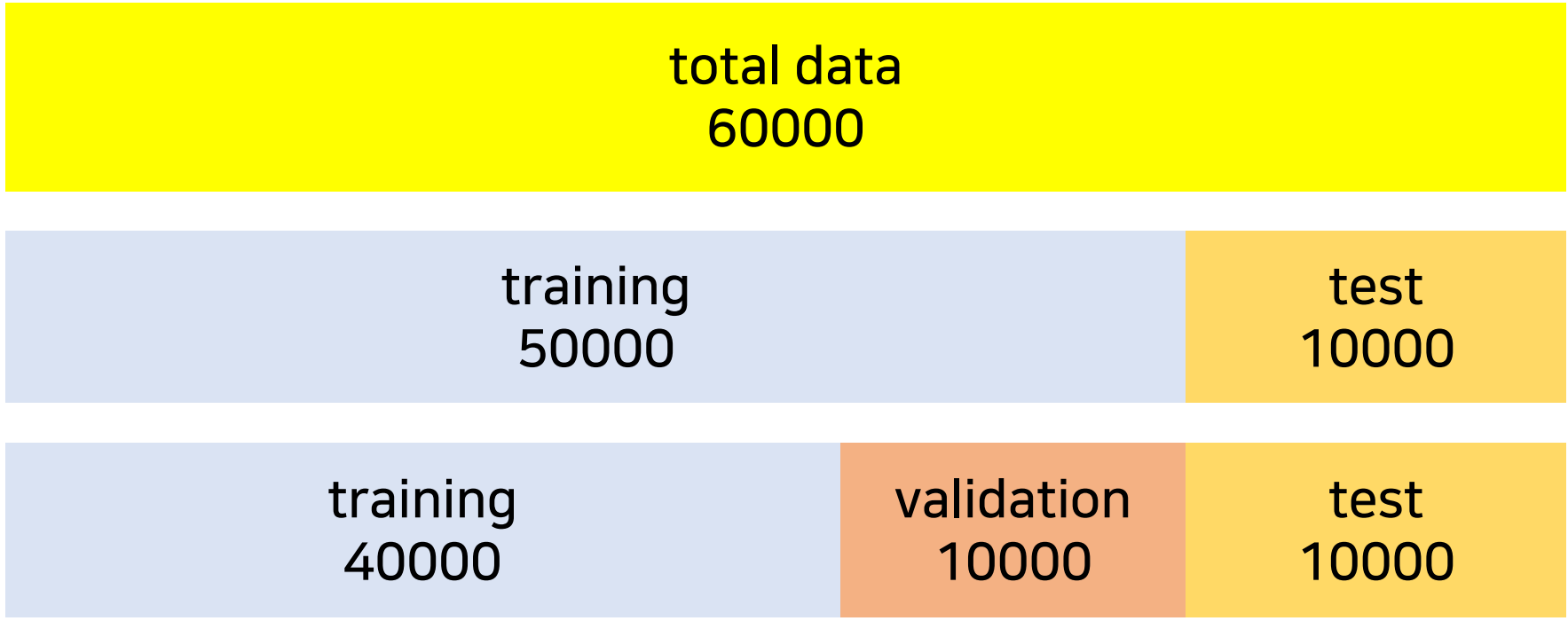
iris dataset

Mnist dataset



Fashion Mnist dataset

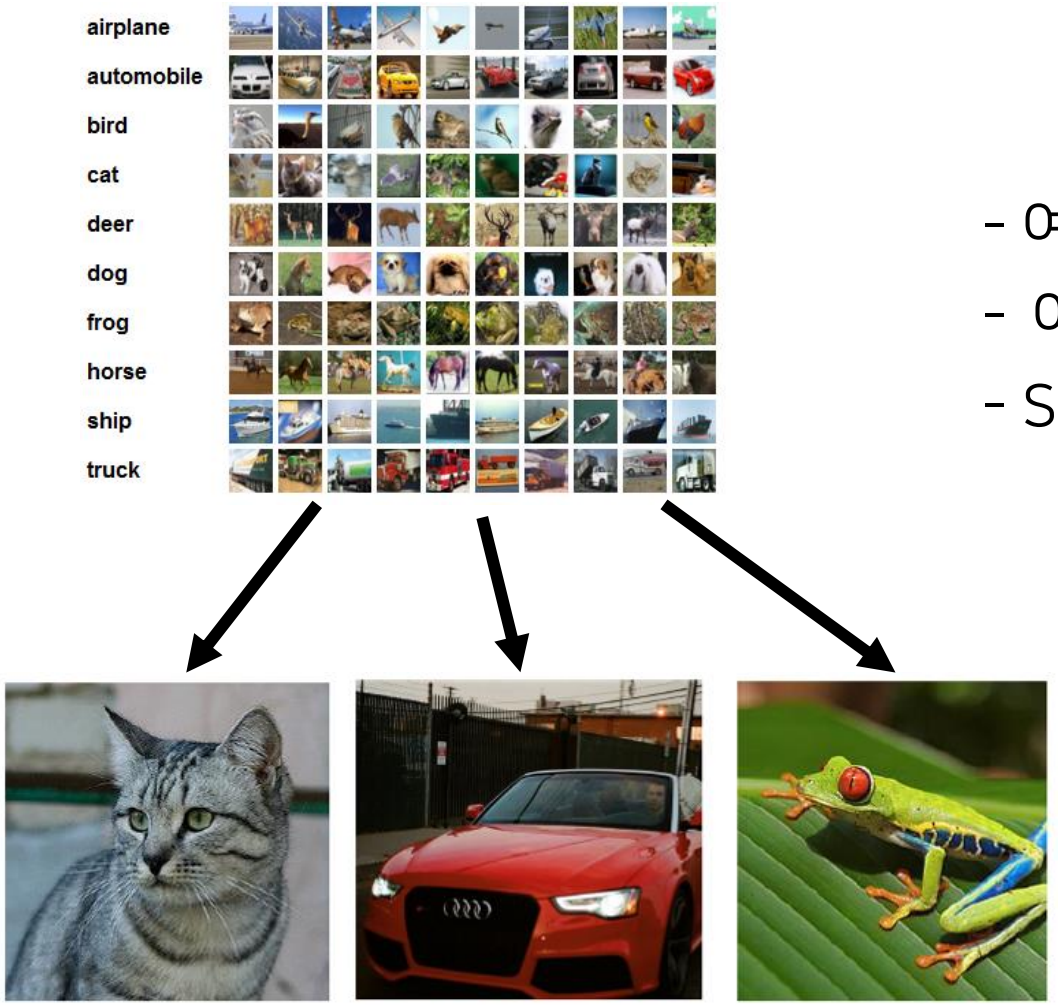
CIFAR10


total data
60000
training
50000
test
10000
training
40000
validation
10000
test
10000

Contents

2장 딥러닝 흐름잡기

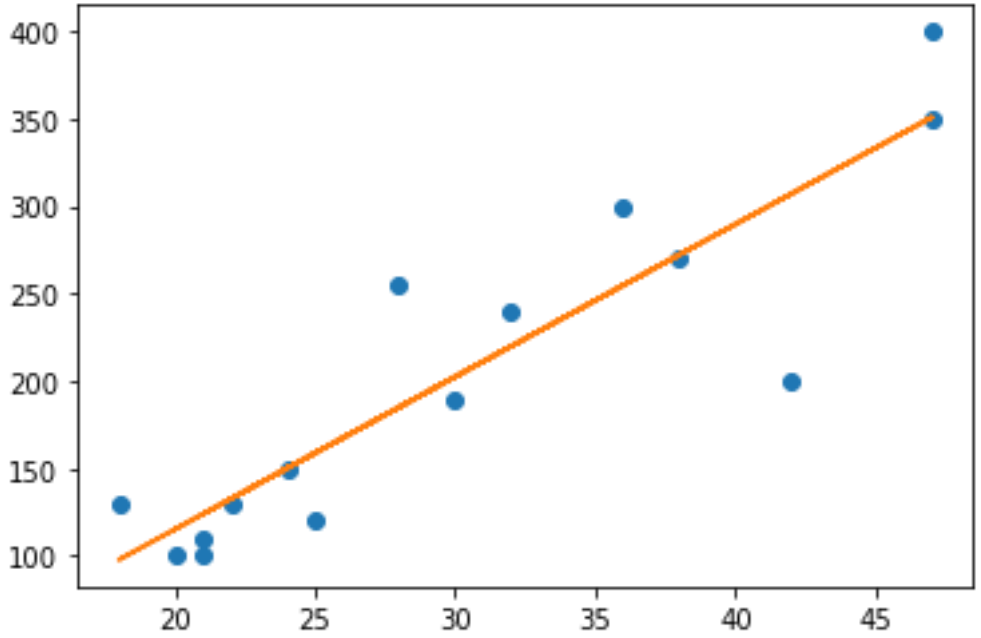
분류 Classification



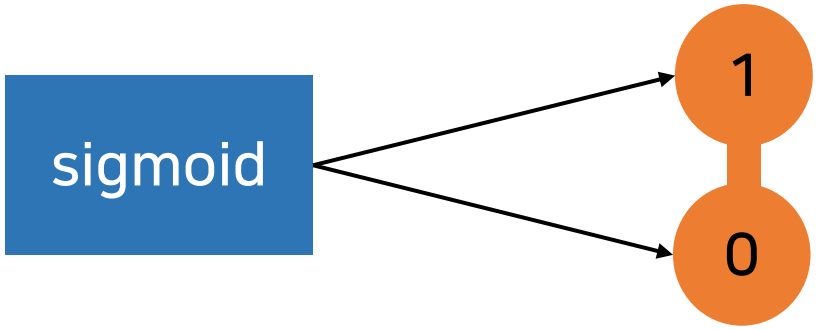
- 여러 개의 결과값 중에 하나를 선택
- 이미지 분류 등
- Softmax classifier

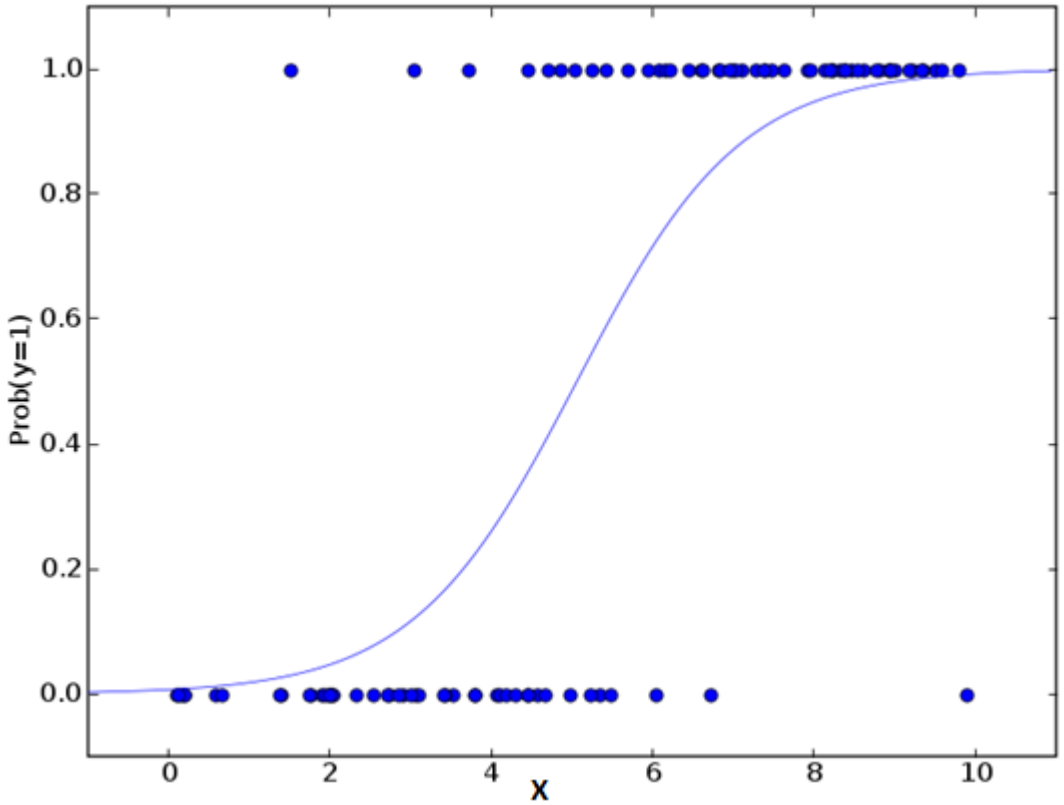
회귀 Regression

- 결과값 하나를 추론하는 문제
- 실수 값 추정
- 종류 : 선형, 다항식 등
- 규제 회귀 : Ridge, Lasso, Elastic Net



*Logistic Regression



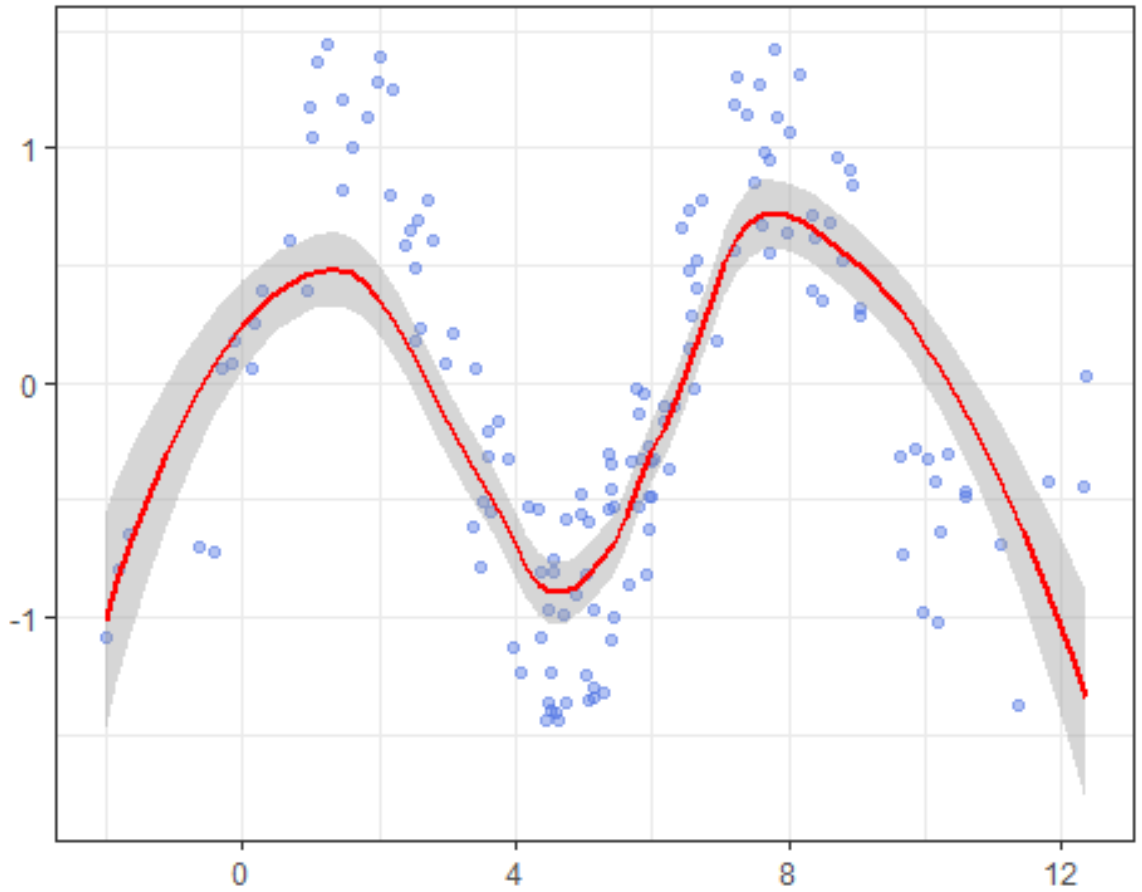


- True/False 확률을 찾는데 사용
- Binary Classification 에 사용
- 종속 변수(Y)와 독립 변수(x) 간의 선형 관계를 요구하지 않음

*다항식 회귀 Polynomial regression

$$f(x) = ax^4 + bx^3 + cx^2 + dx + e$$

- 더 낮은 차수의 오차를 얻기 위해 다항식 사용
- 과도한 피팅 초래 가능



Contents

2장 딥러닝 흐름잡기

가중치

$$z = ax + by + c \qquad \mathbf{Z} = \boldsymbol{W}\mathbf{X} + \boldsymbol{b}$$

x, y	입력 변수	$\mathbf{X}$	vector
z	출력 변수	$\mathbf{Z}$	vector
a, b	Weight 가중치	$\boldsymbol{W}$	matrix
c	Bias 편향	$\boldsymbol{b}$	vector

머신러닝 모델

결정 트리

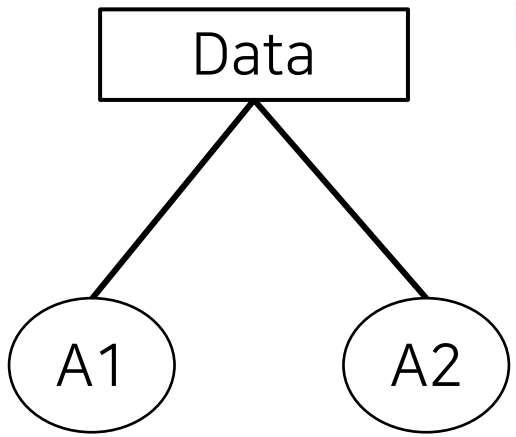
Ensemble

RandomForest

Boosting

결정 트리 Decision Tree

- 의사 결정 규칙과 그 결과들을 트리 구조로 도식화한 의사 결정 지원 도구
- flowchart와 같은 구조
- 입력 데이터 포인트를 분류
- 주어진 입력에 대한 출력 값을 예측

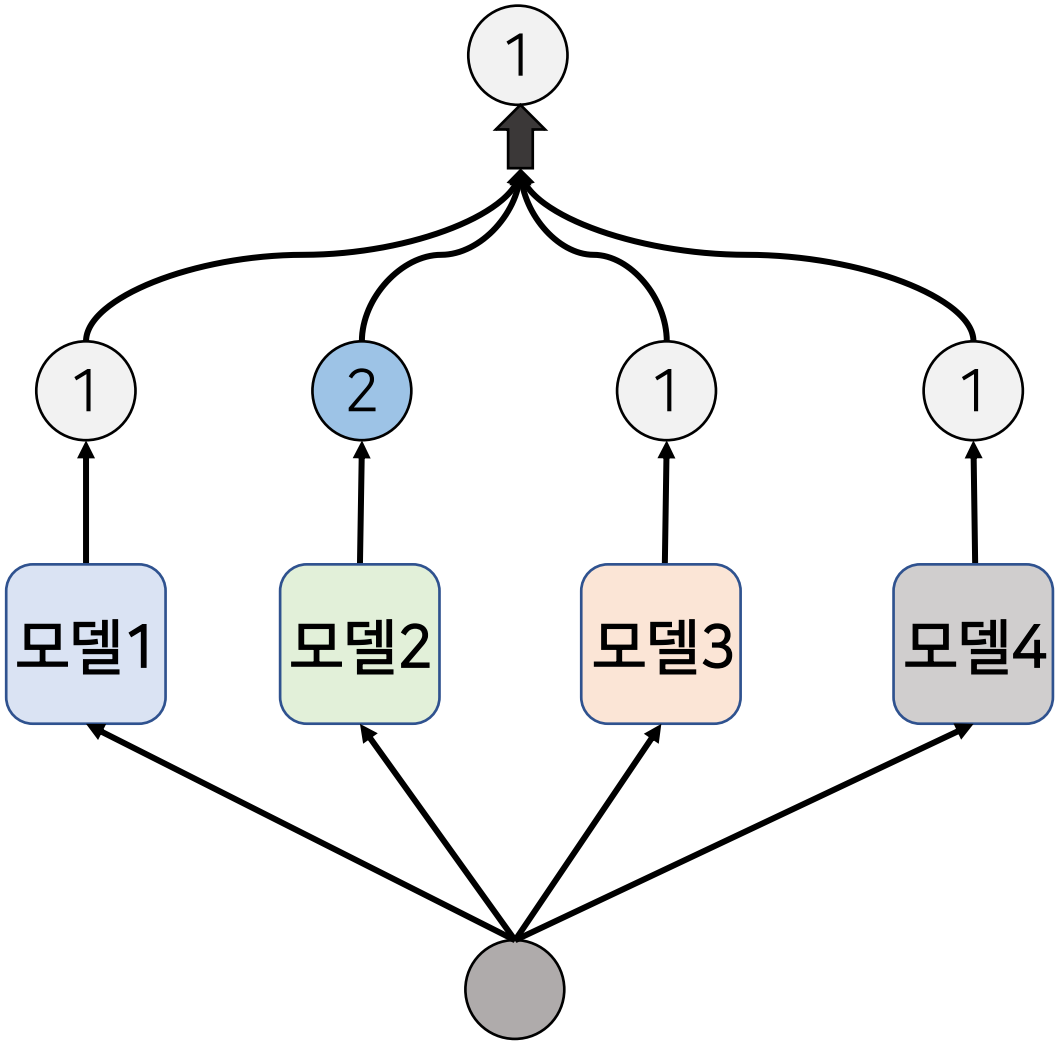


Root node

Leaf node

Terminal node

Ensemble



예측 통합

예측

여러개의
분류기 생성

여러개의 분류기 생성

예측 결합보다 정확한 최종예측을 도출

대부분의 정형데이터 분류시 뛰어난 성능

기본유형 :

보팅, 배깅, 부스팅 의 세가지

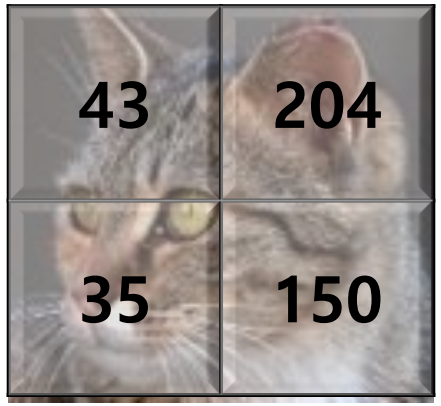
+ **스태킹**을 포함한 다양한 방법

예 : 랜덤포레스트, 그래디언트 부스팅

이미지 분류 모델

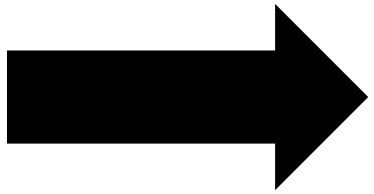
Image to vector

입력 이미지



2x2 pixel

Label: 1



43
204
35
150

4-element Vector

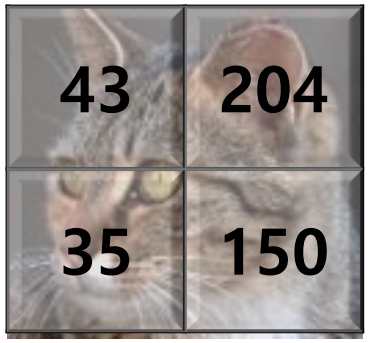
선형 분류기 : score function

$$f(x, W) = Wx + b = y$$

score function

f

Input image



x → Wx+b

score

36.8

z: Label

Class: cat

W: weight

b: bias

2-layer 신경망

선형 분류기
Score 함수

$$f = Wx$$

$$x \in \mathbb{R}^D, W \in \mathbb{R}^{C \times D}$$

신경망
2-layer

$$f = W_2 \max(W_1 x, 0)$$

$$x \in \mathbb{R}^D, W_1 \in \mathbb{R}^{H \times D},$$
$$W_2 \in \mathbb{R}^{C \times H}$$

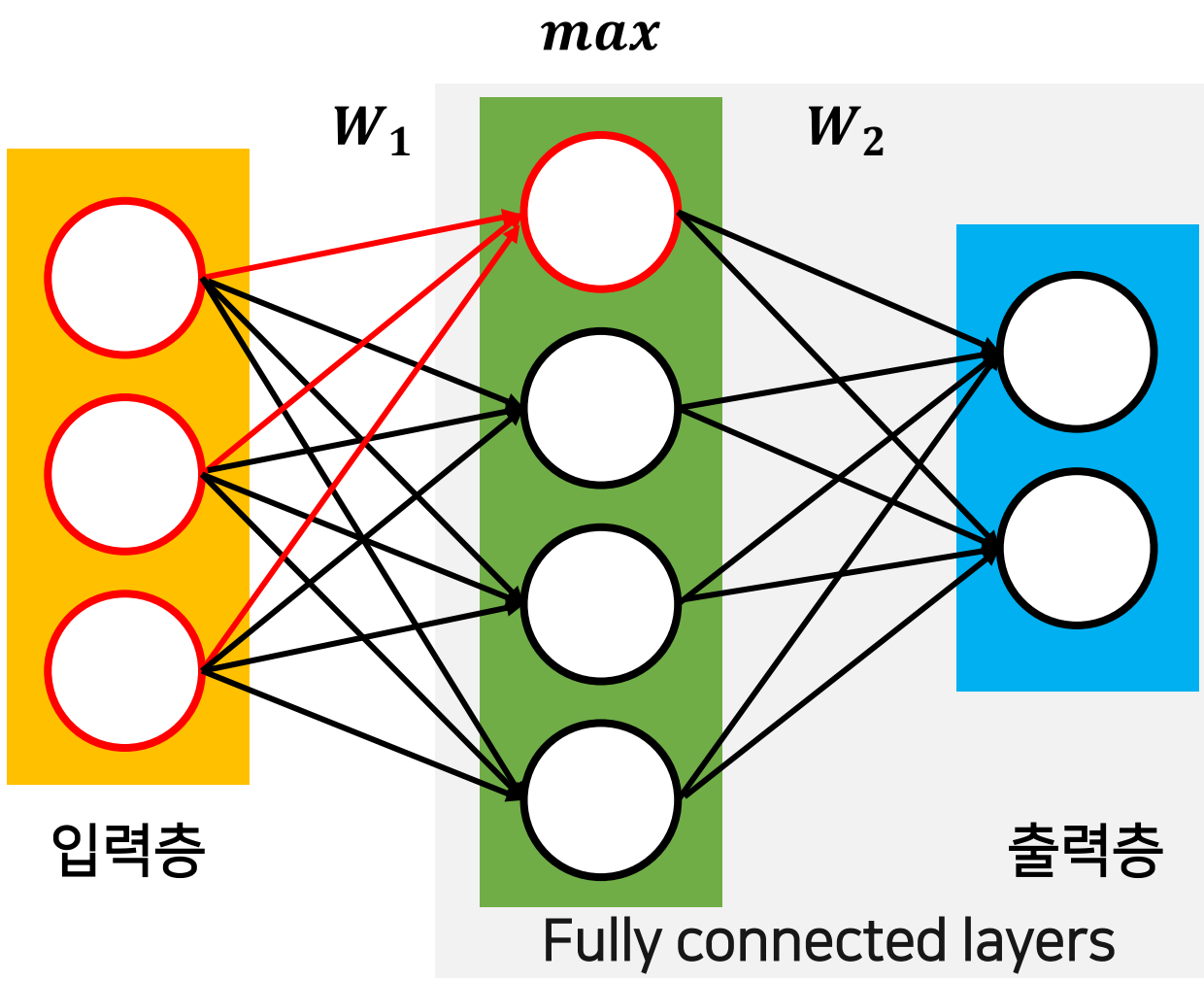
Fully-connected networks

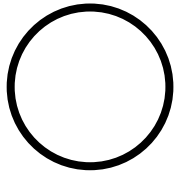
Multi-layer
perceptrons(MLP)

2-layer 신경망

$f = W_2 \max(W_1 x, 0)$

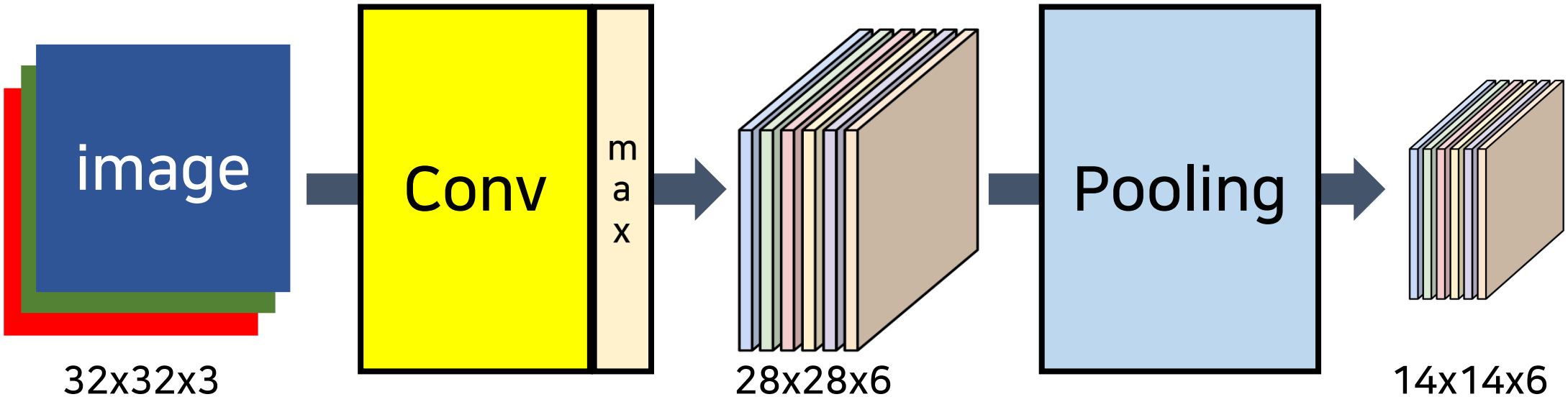
1-은닉층 신경망





뉴런(neuron)

CNN 모델



자연어처리 모델

Word Embedding

One Hot

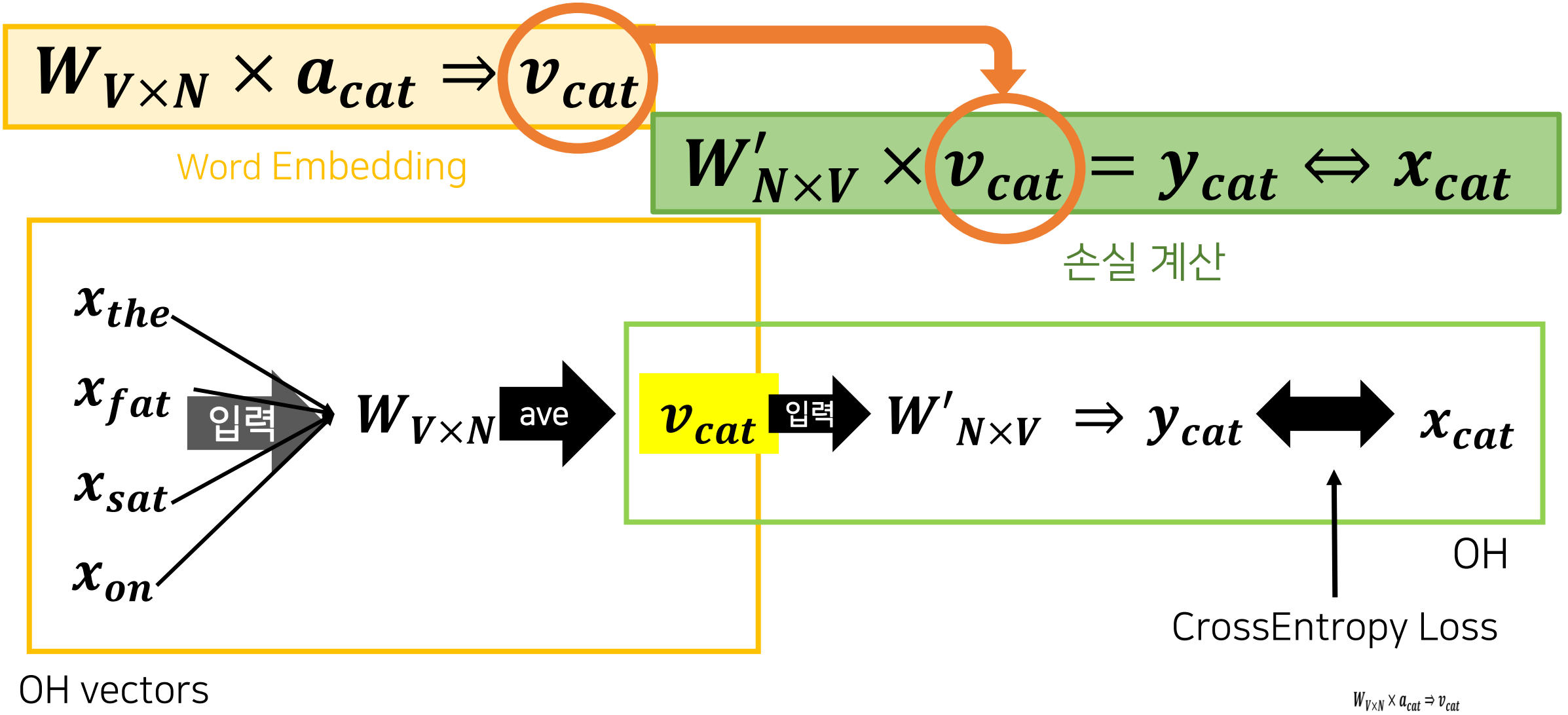
CBOW

Model

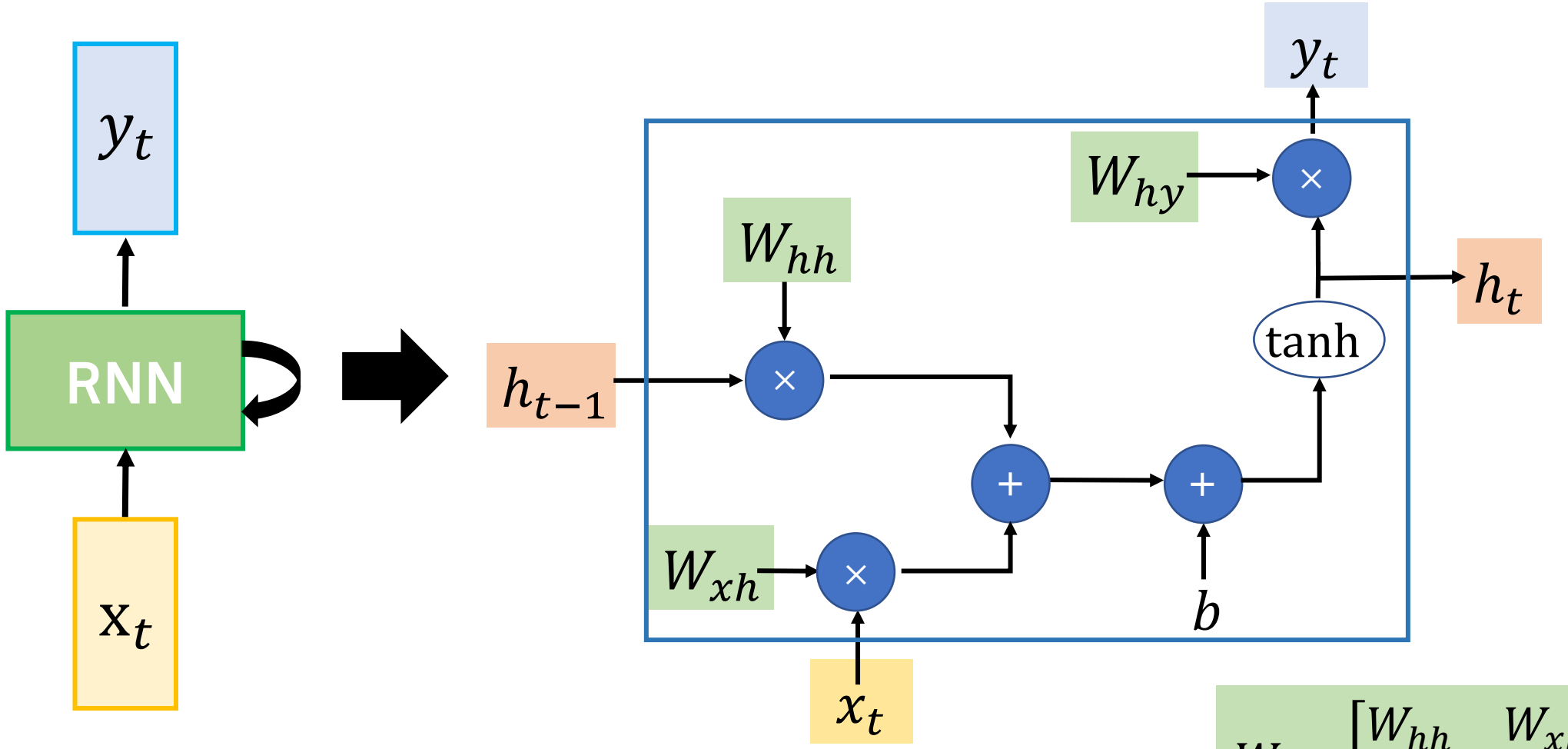
RNN

LSTM

Word Embedding : CBOW



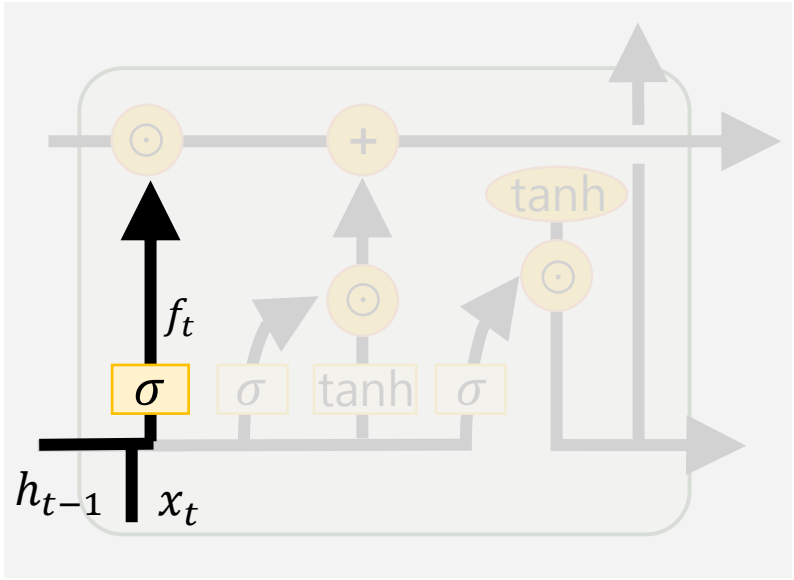
Simple RNN : 단위 모듈 계산



$$W = \begin{bmatrix} W_{hh} & W_{xh} \\ W_{hy} & 0 \end{bmatrix}$$

LSTM 진행과정

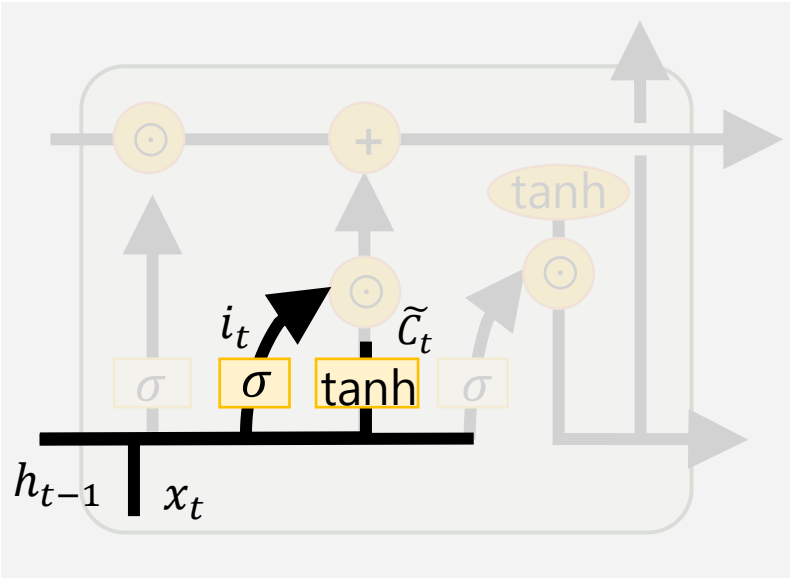
Drop 정보
선택 과정



Forget Gate Layer

Drop할 정보를 sigmoid 출력
(0~1사이)

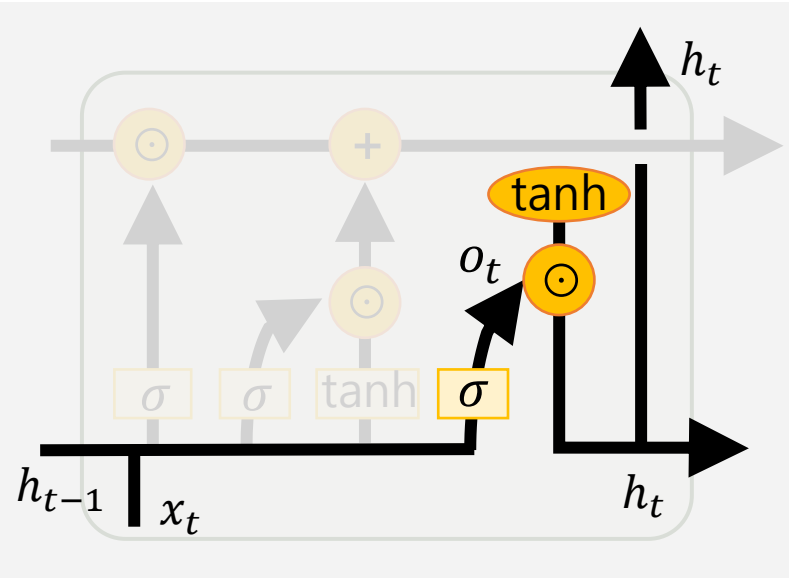
저장 정보
선택 과정



Input Gate Layer

업데이트 Data 결정
Cell State에 저장

출력 정보
선택 과정



Output Gate Layer

출력 Data 결정
다음 노드로 전파

Contents

2장 딥러닝 흐름잡기

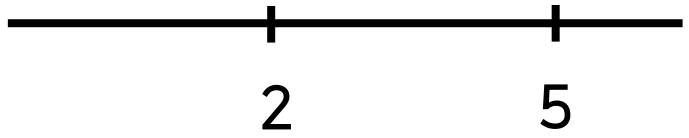
손실

에러
참값과 근사값의 차이

손실
모델링에서 얻은 값과 참값의 차이

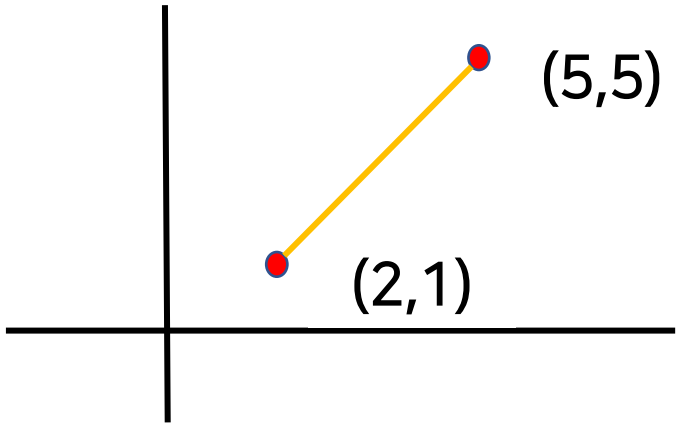
거리 개념

1차원



$$|5-2| = 3$$

2차원



$$\sqrt{(2-5)^2+(1-5)^2} = 5$$

거리 정의

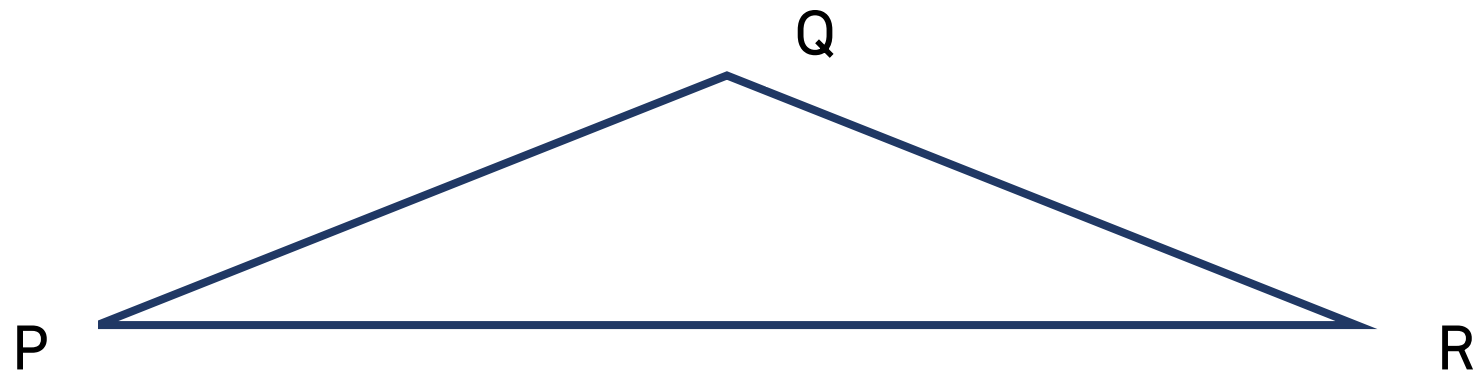
V : set $\quad P, Q, R \in V$

Function $d : V \times V \rightarrow R^{+}$ **Distance**

1. Non-negativity $\quad d(P,Q) \geq 0$ and $(P,Q) = 0 \Leftrightarrow \ P = Q$

2. Symmetry $\quad d(P,Q) = d(Q,P)$

3. Triangular Inequality $\quad d(P,R) \leq d(P,Q) + d(Q,R)$



거리 종류

$$P = (P_1, P_2, \cdots, P_n) Q = (Q_1, Q_2, \cdots, Q_n)$$

L_1 norm distance

$$d_1(P,Q) = \sum_{k=1}^{n} |P_i - Q_i|$$

L_2 norm distance

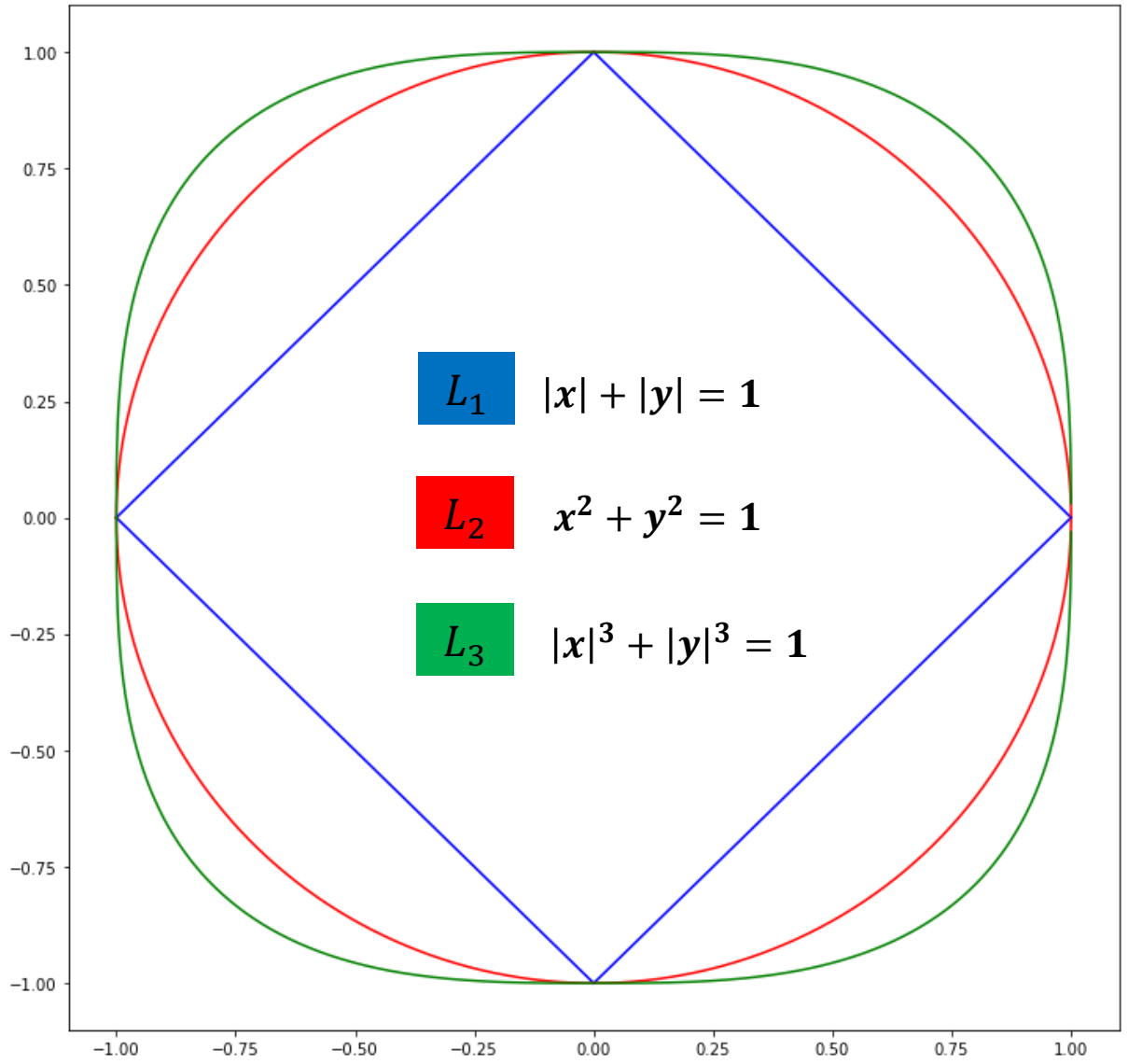
$$d_2(P,Q) = \left(\sum_{k=1}^{n} (P_i - Q_i)^2 \right)^{\frac{1}{2}}$$

Max norm distance

$$d_\infty(P,Q) = \max_i |P_i - Q_i|$$

L_p $(p \geq 1)$ Minkowski

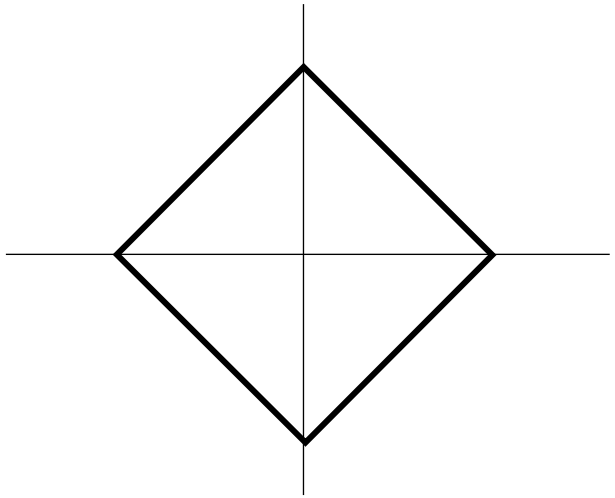
$$d_p(P,Q) = \left(\sum_{k=1}^{n} |P_i - Q_i|^p \right)^{\frac{1}{p}}$$



L1 L2

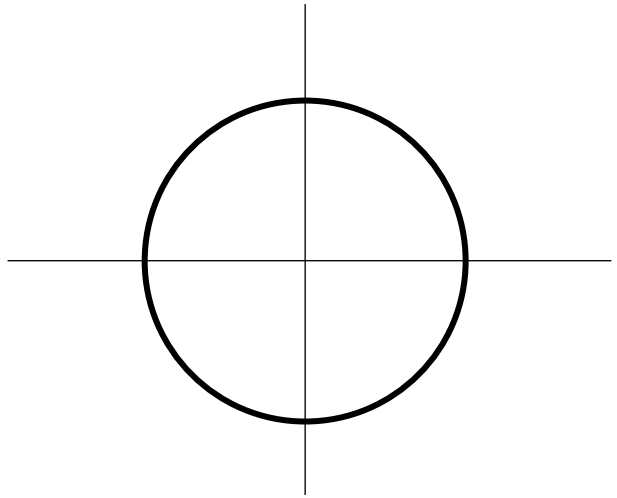
L1 distance
(Manhattan)

$$d_1(P,Q) = \sum_{k=1}^{n} |P_i - Q_i|$$



L2 distance
(Euclidean)

$$d_2(P,Q) = \sqrt{\sum_{i=1}^{n} (P_i - Q_i)^2}$$



손실 함수 종류

손실은 거리(차이)로 측정

회귀용	분류용
MAE MSE	Cross Entropy

손실 함수 종류 : 회귀

MAE

$$\frac{1}{m}\sum_{i}|H(x_i)-y_i|$$

H(x)	3	4	예측값
y	2	1	참값

$$|3-2|+|4-1|=4$$

$1/m$ → $4/2=2$

MSE

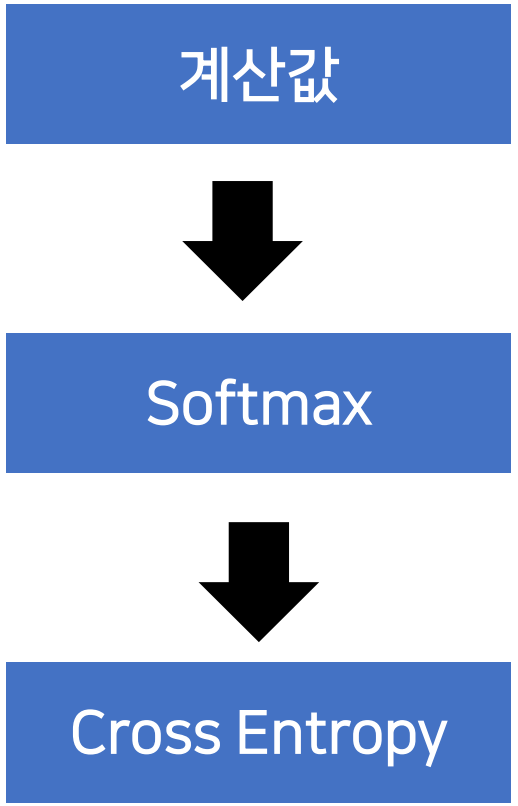
$$\frac{1}{m}\sum_{i}(H(x_i)-y_i)^2$$

$$|3-2|^2+|4-1|^2=10$$

$1/m$ → $10/2=5$

손실 함수 종류 : 분류

Cross Entropy Loss
확률 분포를 활용 확률이 100%면 손실 0



$$f_j(z) = \frac{e^{z_j}}{\sum_k e^{z_k}}$$

$$L_i = -\log f_j(z)$$

Contents

2장 딥러닝 흐름잡기

함수 최소값 구하기

함수 최소값

1. 함수 구하기 : $y = f(x)$

2. 미분하기 : $f'(x)$

3. Critical Point 찾기 : $f'(x) = 0 \rightarrow x = a$

4. Minimum 구하기 : $f(a)$

참값 계산 가능
해석적으로 참값을 직접 계산

함수 최소값 예제

1. 함수 구하기

$$y = x^2 + 4x + 2$$

2. 미분하기

$$y' = 2x + 4$$

3. 임계점 찾기

$$2x + 4 = 0 \rightarrow x = -2$$

4. 최소값 구하기

$$y(-2) = -2$$

손실 최소화 하기

손실 최소화

1. Loss function 구하기 : $y = L(W)$

2. 미분하기 : $\frac{dL}{dW}$

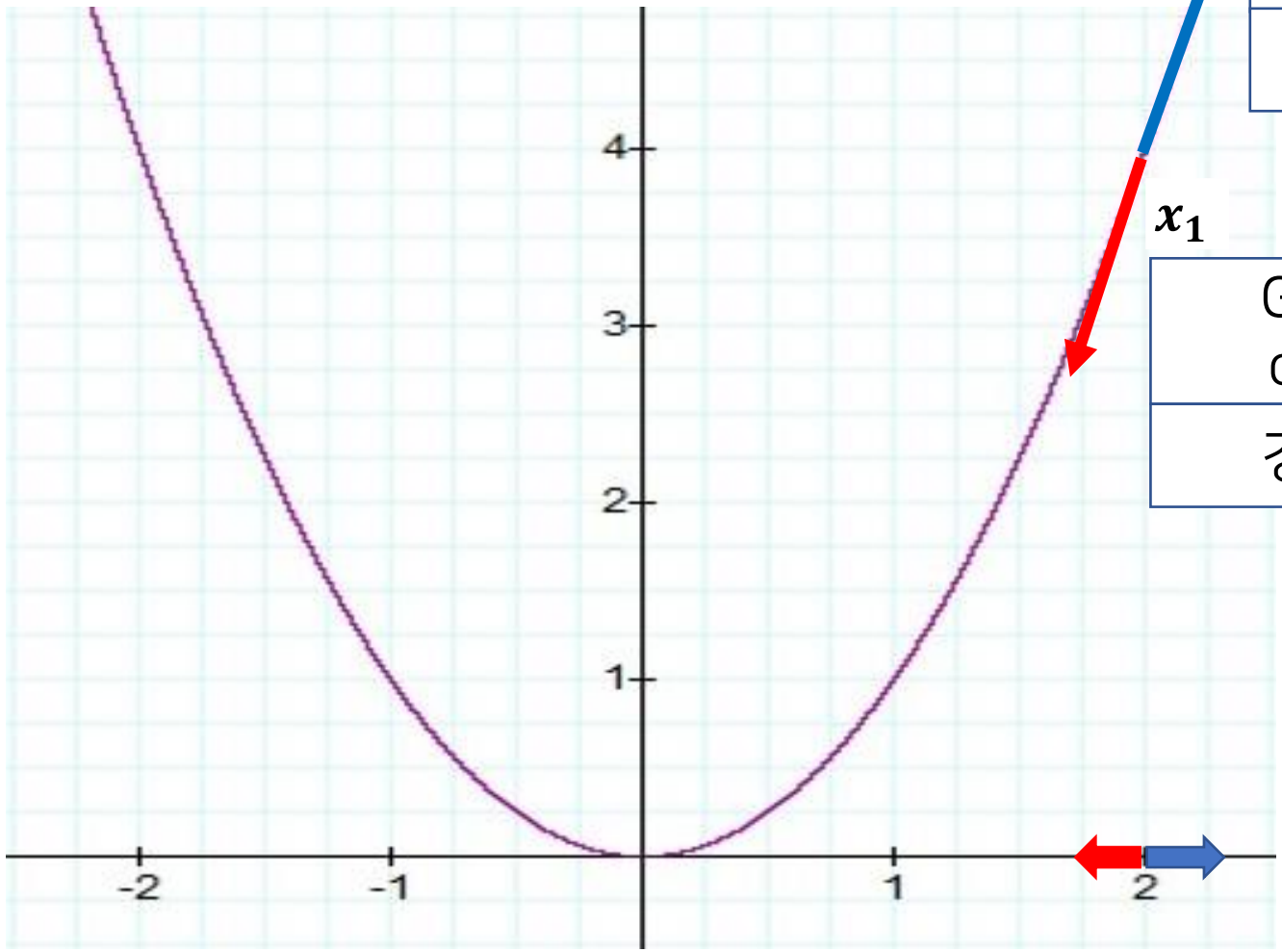
3. 경사 하강 방향으로 업데이트

$$W = W - h \frac{dL}{dW}$$

참값 계산이 어려움
근사적으로 참값에 접근

경사하강법

$y = x^2$



Gradient ascent
경사 상승

Gradient descent
경사 하강

경사하강법

$$y = x^2$$

$$x_{n+1} = x_n - h * f'(x_n)$$

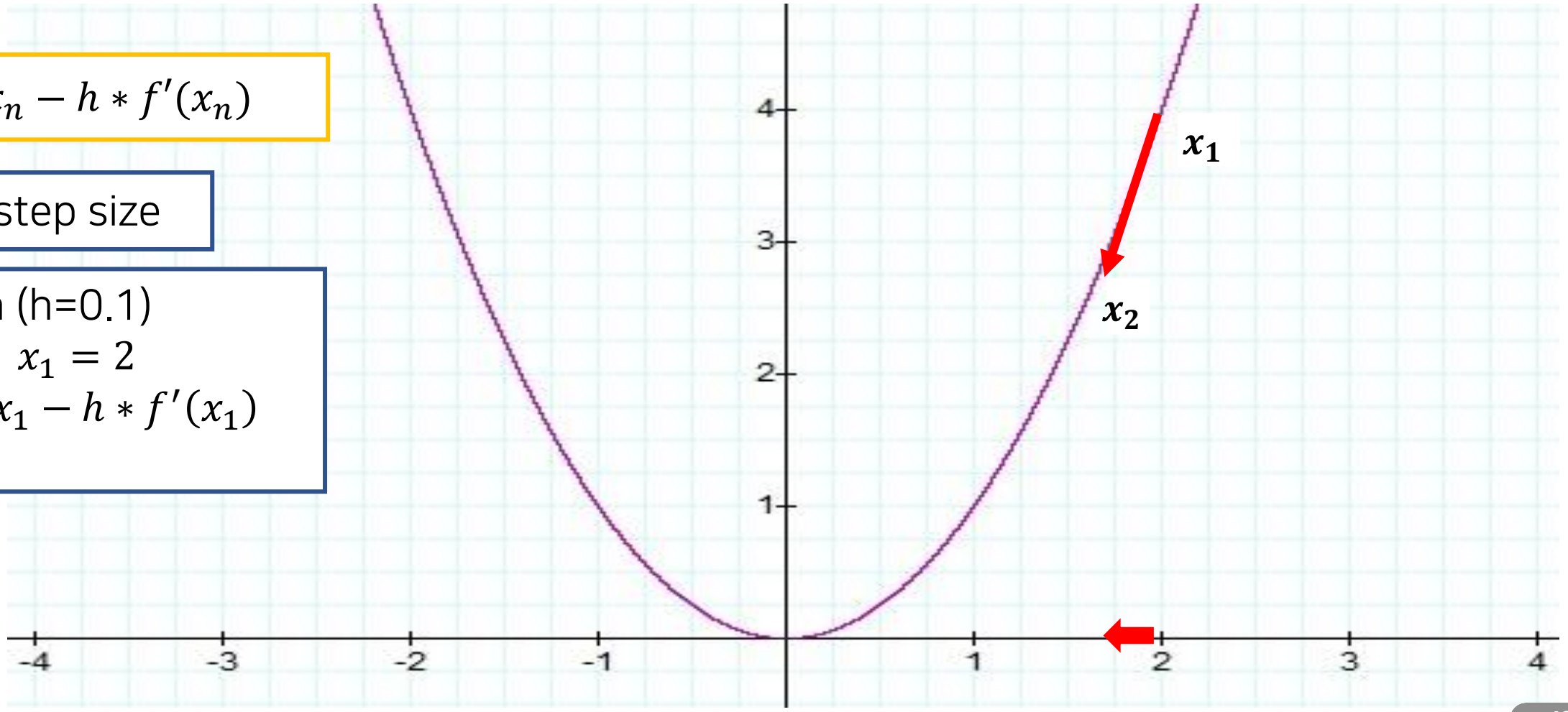
h: step size

Iteration (h=0.1)

$$x_1 = 2$$

$$x_2 = x_1 - h * f'(x_1)$$

$$= 1.6$$



경사하강법

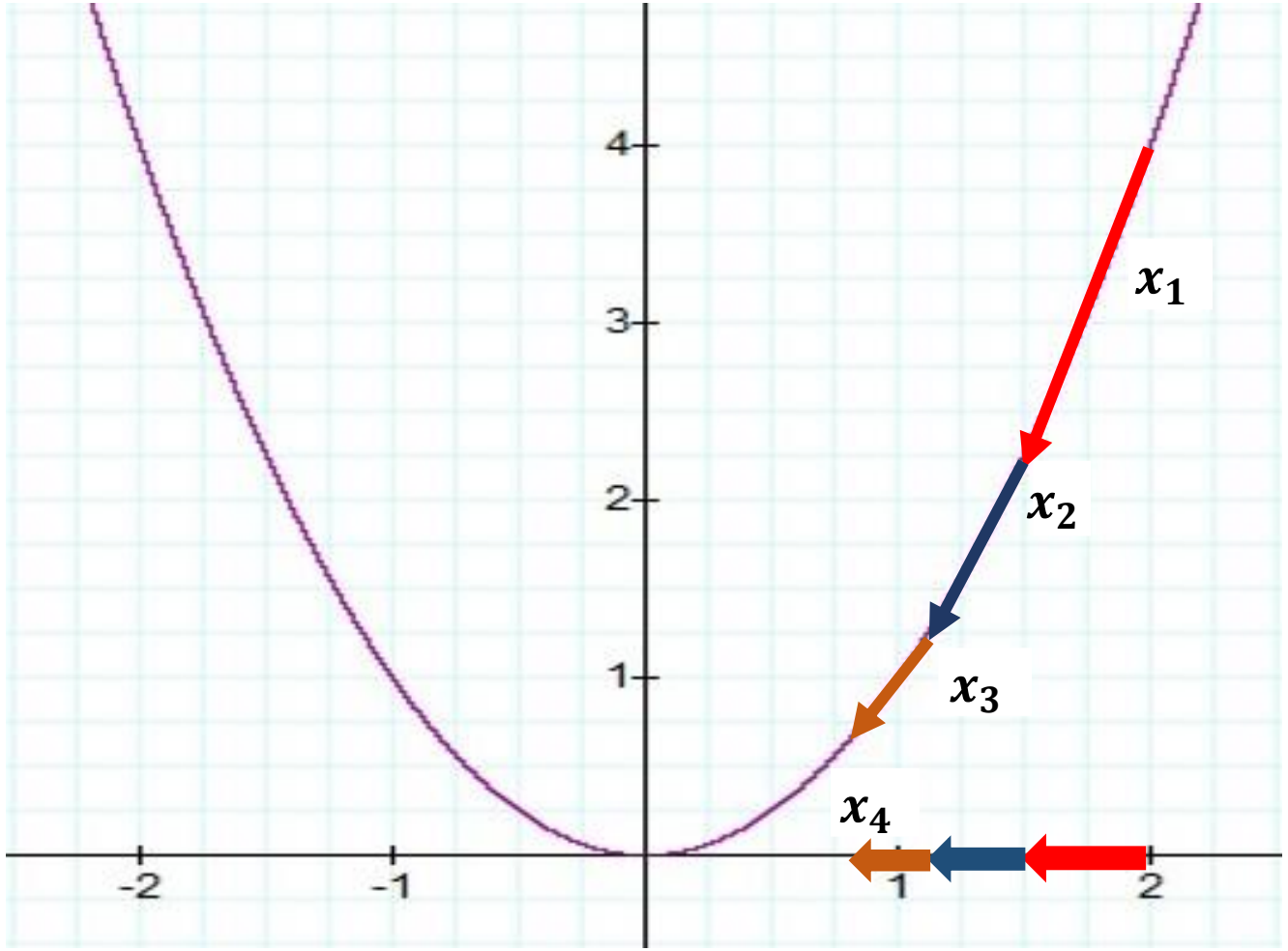
$$y = x^2$$

$$x_{n+1} = x_n - h * f'(x_n)$$

h: step size

Iteration

$$x_1 = 2$$
$$x_2 = x_1 - h * f'(x_1)$$
$$x_3 = x_2 - h * f'(x_2)$$
……



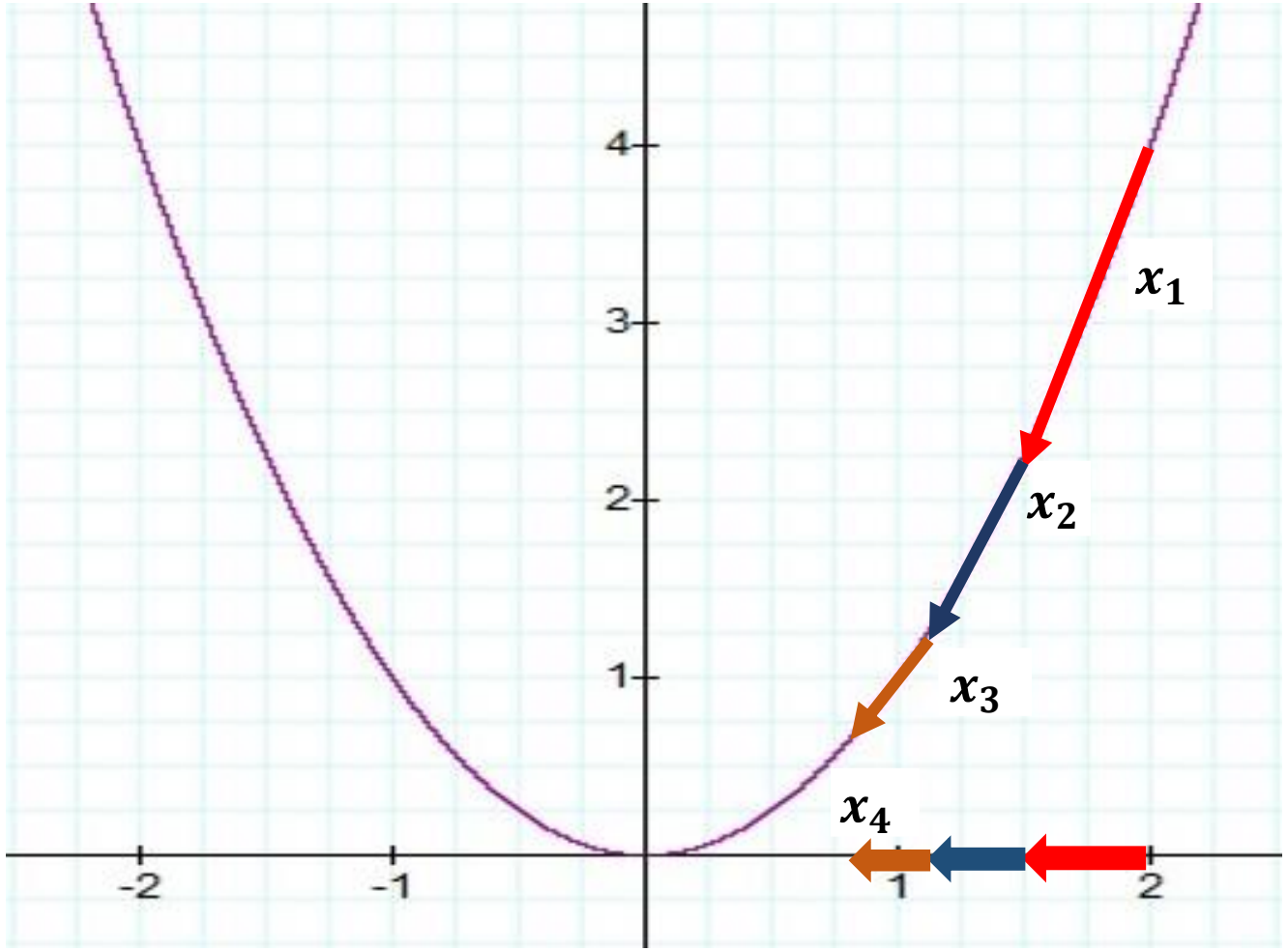
경사하강법

$$y = x^2$$

$$x_{n+1} = x_n - h * f'(x_n)$$

	h=0.1	h=0.2
X1	2	2
X2	1.6	1.2
X3	1.28	…
…	…	…
X21	…	7.31e-5
X50	3.57e-5	

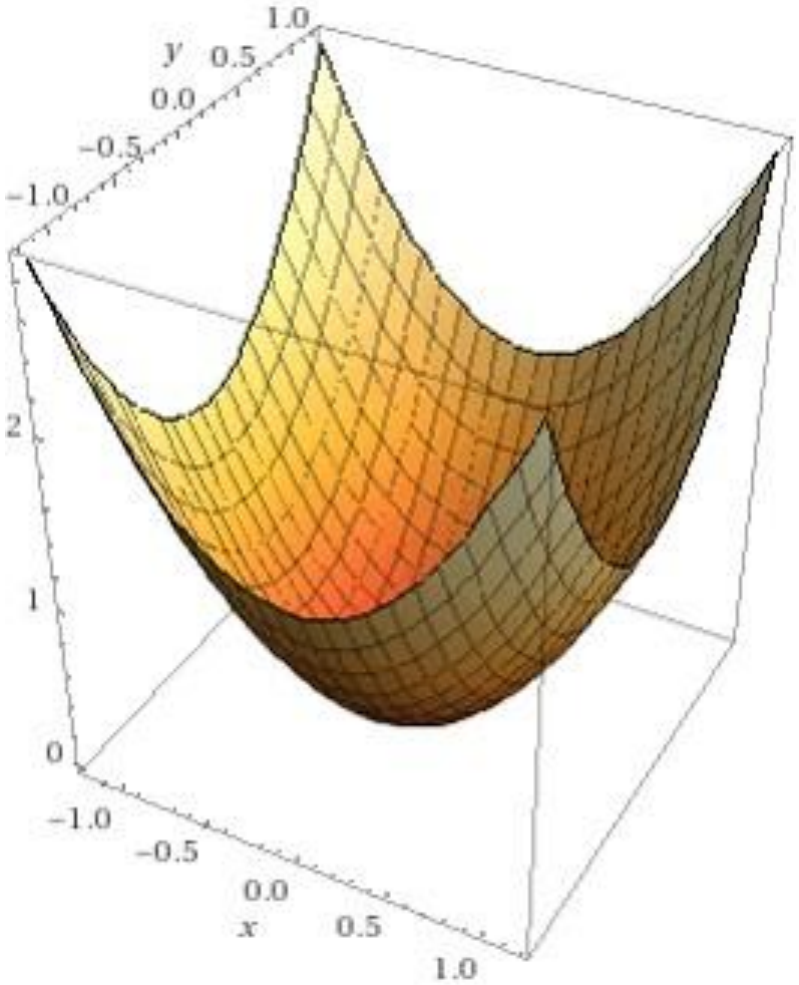
h: learning rate



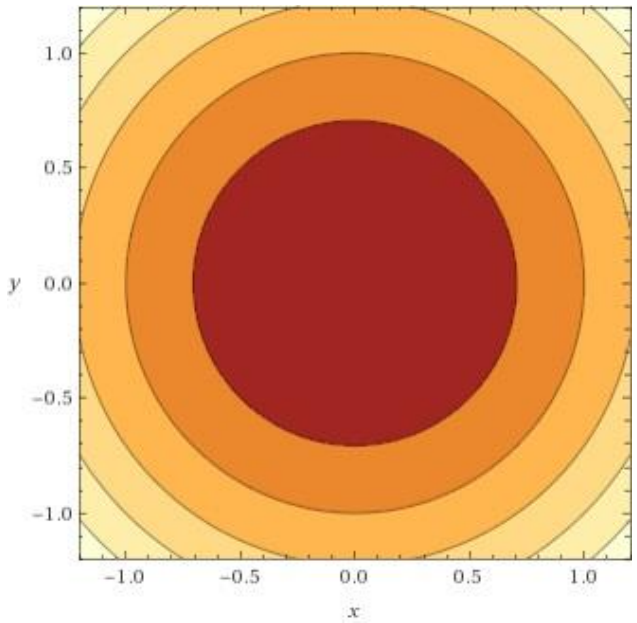
경사하강 예제

$$f(x, y) = x^2 + y^2$$

$$Df = \left(f_x, f_y\right) = (2x, 2y)$$



wolframalpha



경사하강 예제

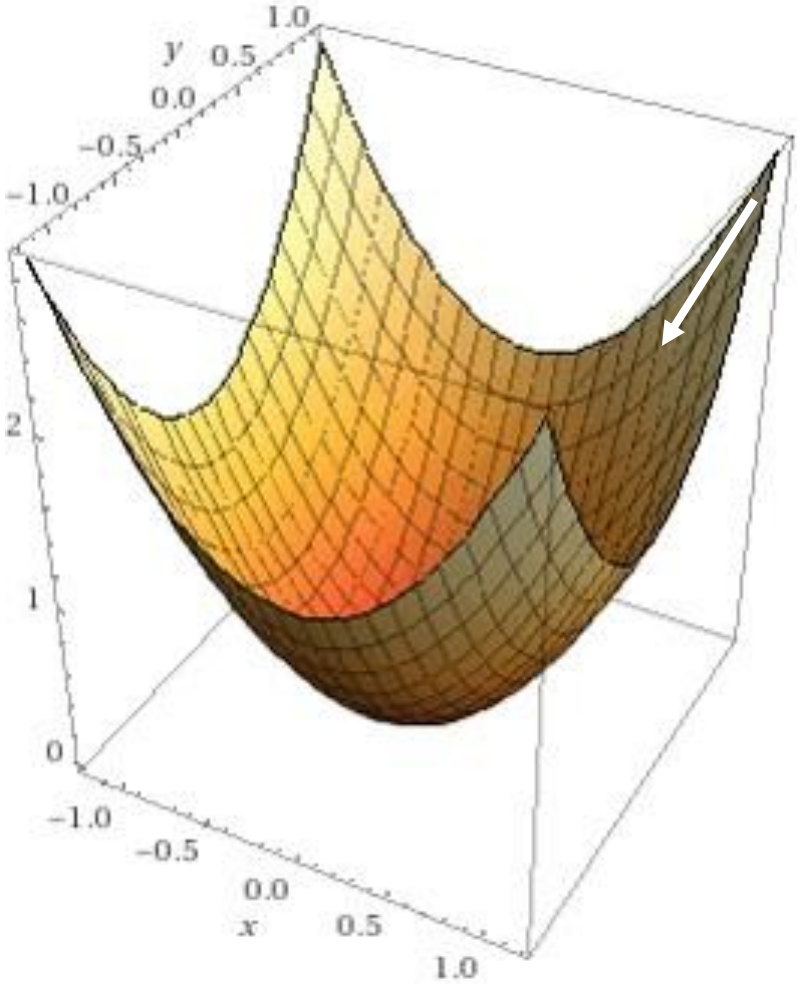
함수

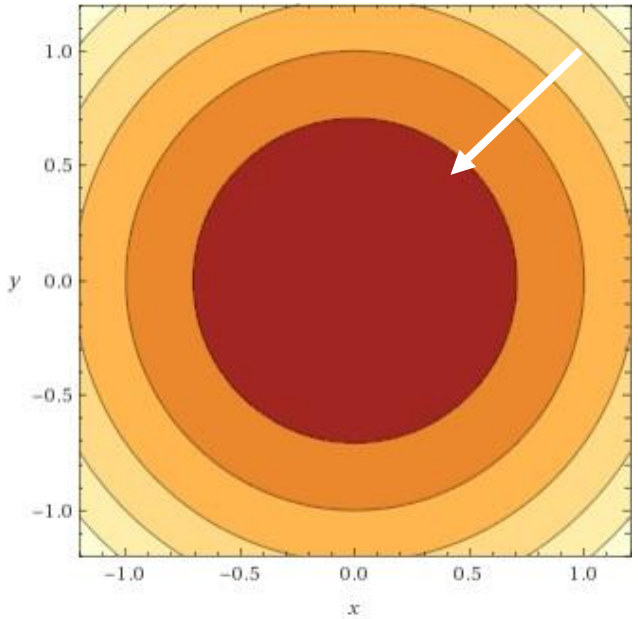
$$f(x,y) = x^2 + y^2$$

미분함수

$$Df = \left(f_x, f_y\right) = (2x, 2y)$$

Gradient
변화율이 가장 큰 방향

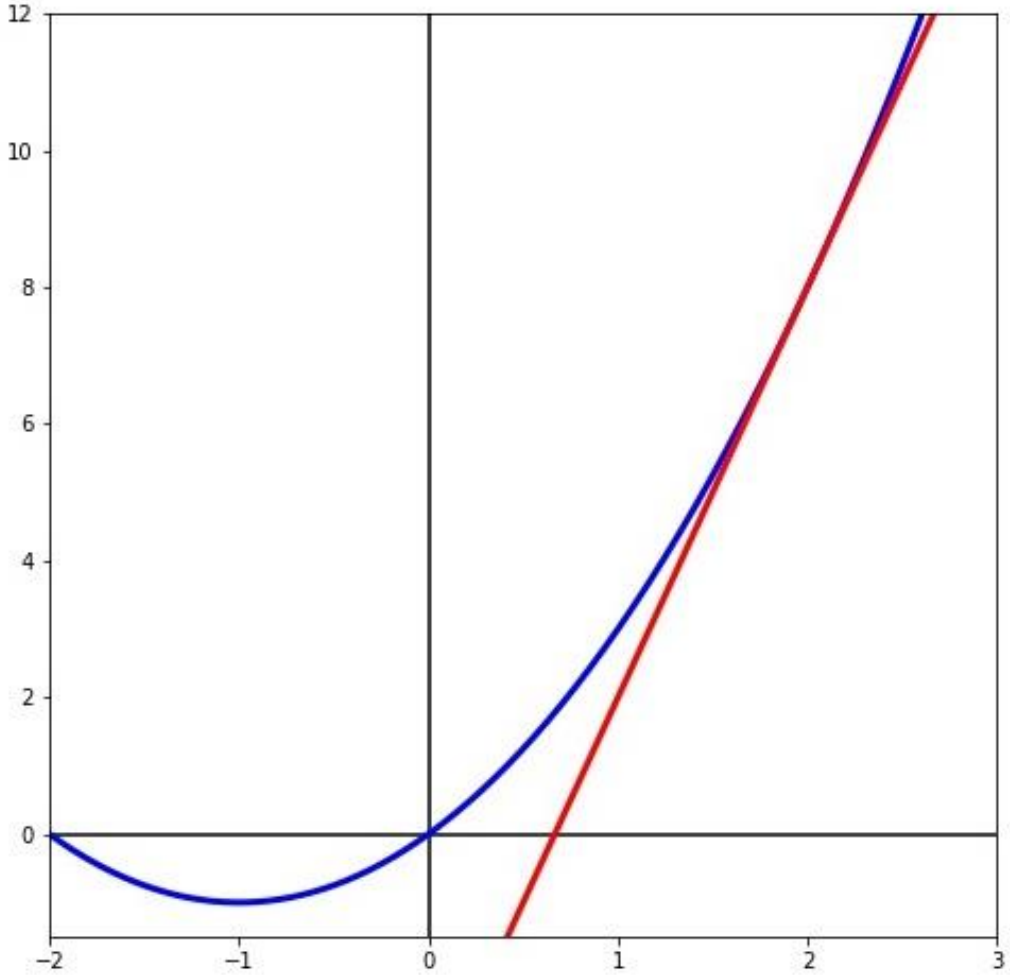




Gradient Descent at (1,1)
(-2,-2)

by wolframalpha

*해석 미분

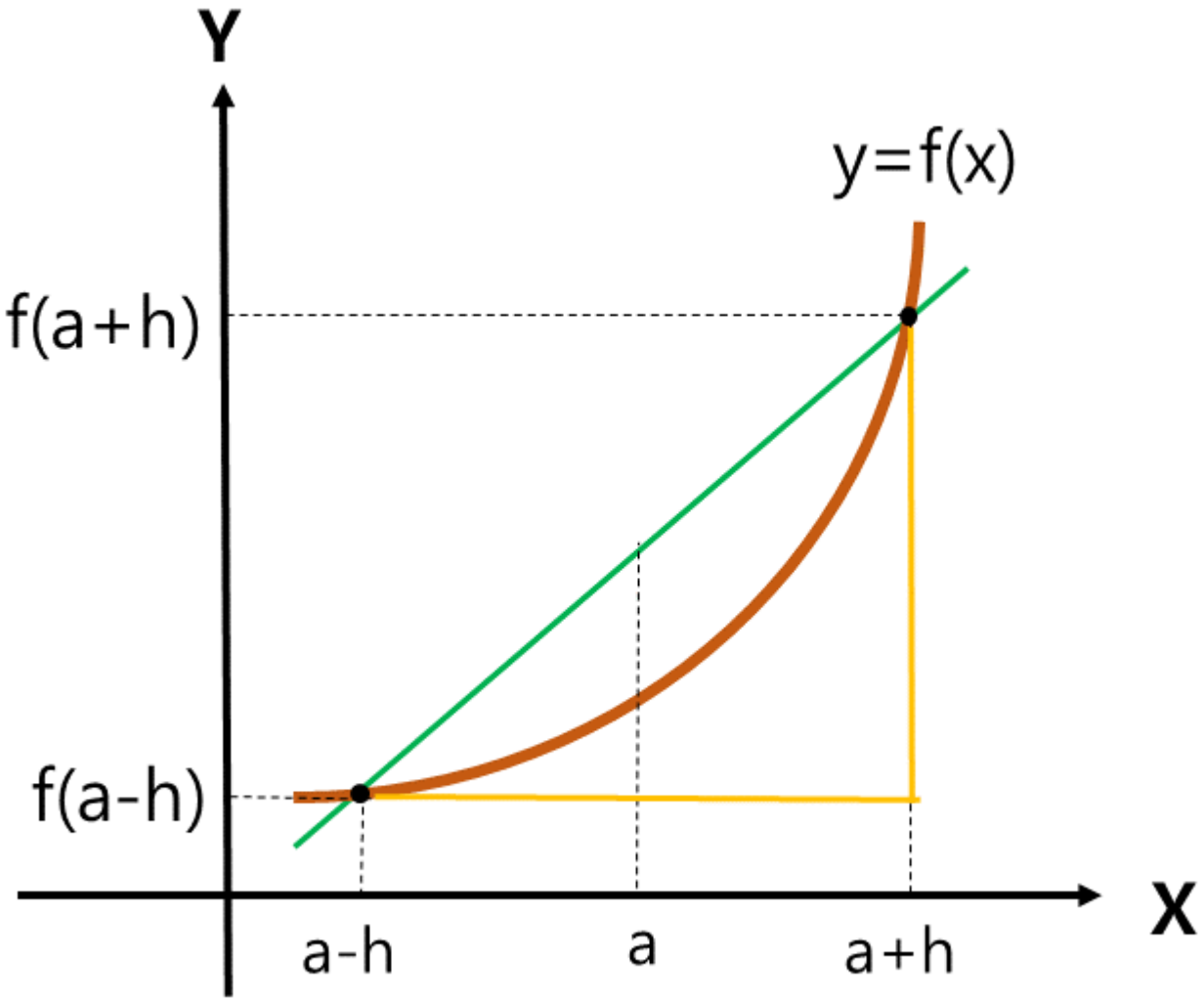


$$f'(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$f(x) = x^2 + 2x$$

$$y = 6x - 4$$

*수치 미분



$$\frac{\Delta y}{\Delta x} = \frac{f(b) - f(a)}{b - a}$$ 평균 변화율

$$f'(a) \cong \frac{f(a+h) - f(a-h)}{2h}$$

Contents

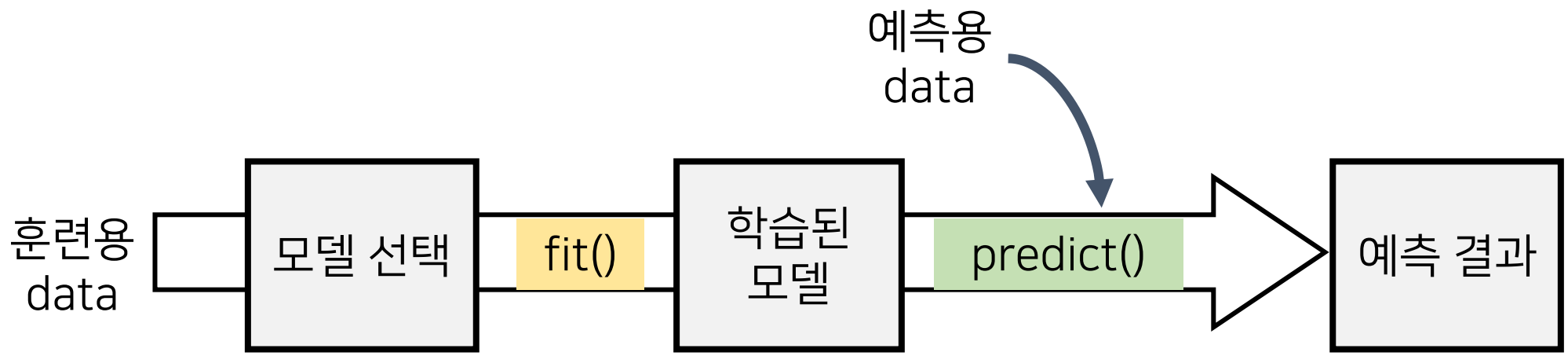
2장 딥러닝 흐름잡기

예측 및 평가

- 학습시킨 모델로 결과값을 예측

model.predict()
결과값 예측

model.evaluate()
정확도 평가



Contents

2장 딥러닝 흐름잡기

실습예제 선형회귀

주요 명령어 1/3

pandas

```
read_csv()    #csv 파일을 읽어 들일 때 사용하는 함수
describe()    #기본적인 통계 자료를 출력
isnull()    #결측치가 있는지를 출력, 전체 데이터를 개별적으로 일일이 출력
isnull().any()    #column별로 결측치 유무 확인
dropna()    #결측치가 들어 있는 데이터 raw를 삭제
fillna()    #함수는 결측치를 다른 값으로 채울 때 사용
```

주요 명령어 2/3

numpy

```
np.array() #기존 데이터를 numpy 형식으로 변환할 때 사용
```

pyplot

```
plt.figure(figsize=(10,8)) #그림의 크기를 설정할 때 사용
plt.plot(x1, y1, 'o')
        #x축이 x1이고 y축이 y1인 데이터를 점(o) 형태로 그리기
```

주요 명령어 3/3

sklearn

```
from sklearn.linear_model import LinearRegression
#sklearn 에 내장되어 있는 linear_model 중 LinearRegression 함수 불러오기
```

LinearRegression()

호출

```
lr=LinearRegression() #불러온 LinearRegression 함수를 lr 변수로 호출
```

학습

```
lr.fit(x2,y1) #x 데이터 x2, y데이터 y1으로 선형회귀를 학습
```

계수

```
a=lr.coef_ # 선형회귀 결과로 구해진 선형 계수를 변수 a에 저장
b=lr.intercept_   # 선형회귀 결과로 구해진 편차를 변수 b에 저장
```

예측

```
lr.predict([[105]])   # 새로운 값을 학습된 선형회귀 모델로 예측하기
```