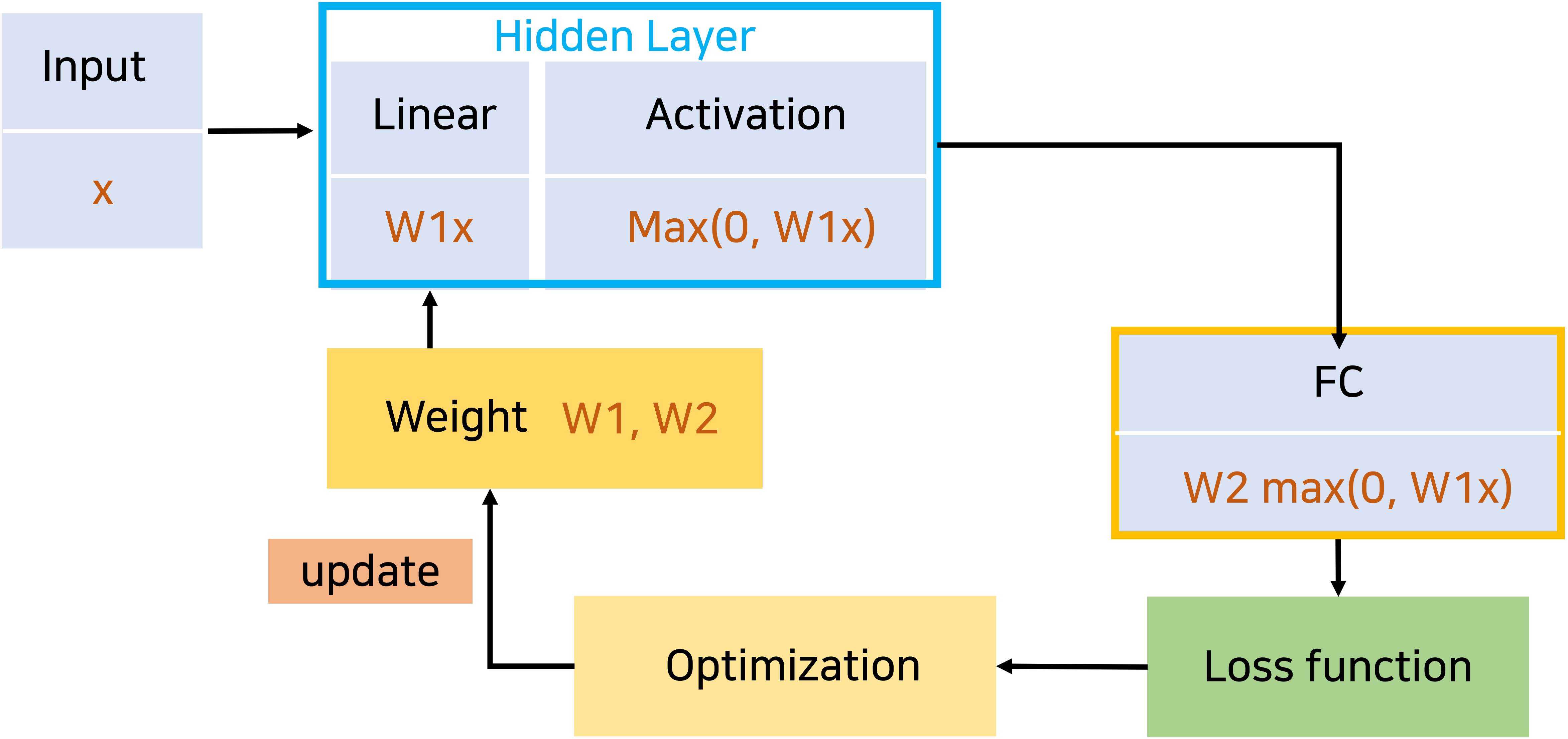
Contents

5장 신경망 모델

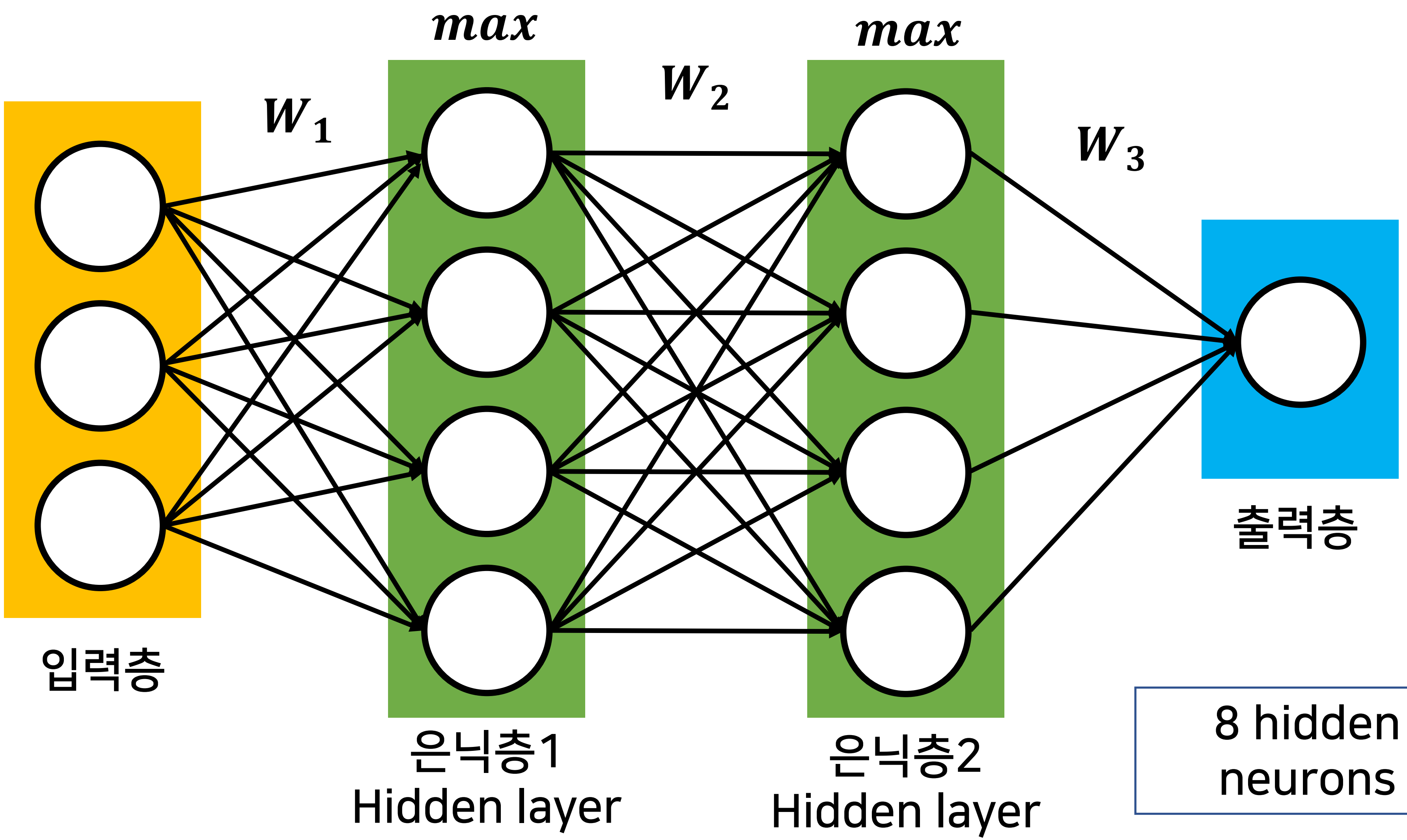
신경망 모델



Input
x
Hidden Layer
Linear
Activation
W1x
Max(0, W1x)
Weight W1, W2
update
Optimization
FC
W2 max(0, W1x)
Loss function

신경망

$$f = W_3 \max(W_2 \max(W_1 x, 0), 0)$$



8 hidden neurons

Contents

5장 신경망 모델

층 결합

활성화 함수
(activation function)

$\max(x, 0)$

신경망
2-layer

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

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

$f = W_2 W_1 x = Wx$

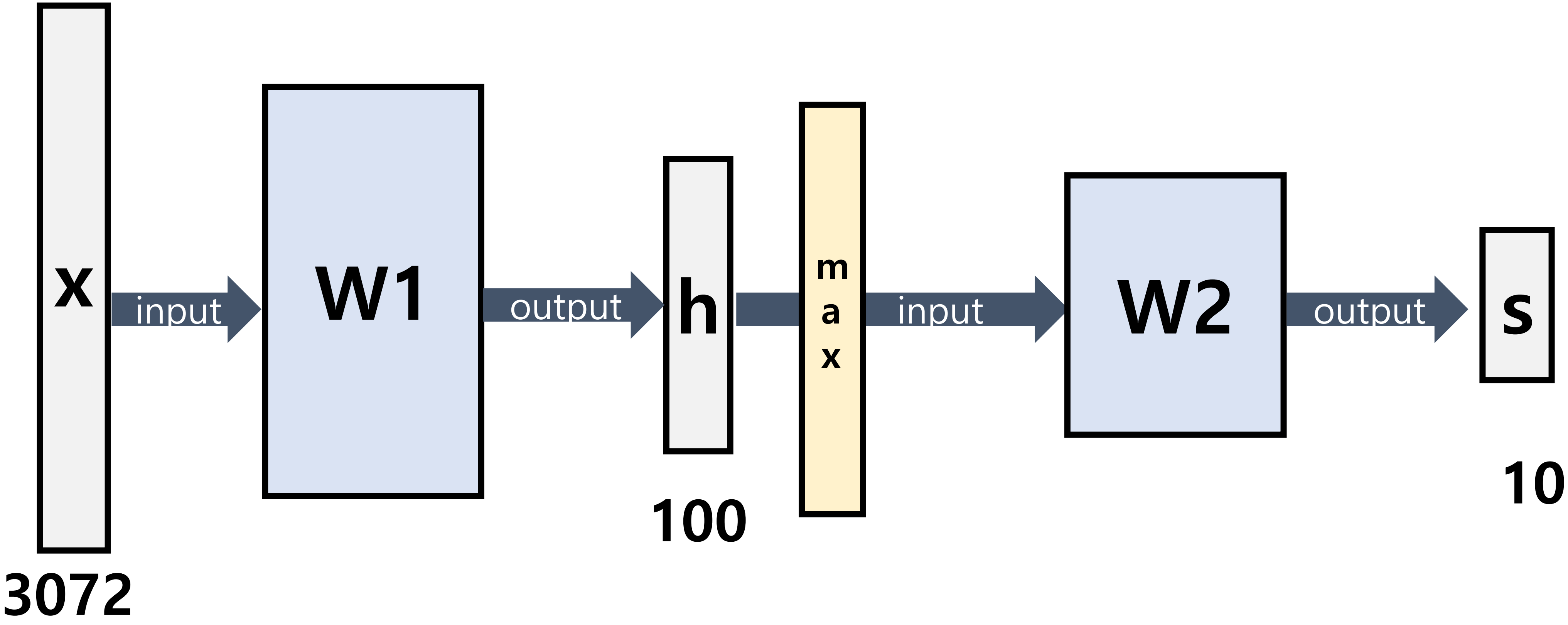
활성화 함수가 없다면 ➡ 선형 분류기

신경망 구조

신경망
2-layer

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

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



Contents

5장 신경망 모델

활성화 함수

Hidden layer

선형 처리 → 활성화(activation) → 선형 / FC

역 할

출력범위 조절

→ 학습 효율 증대

단순 반복으로

→ 복잡한 문제 해결

층을 더 깊게

Backpropagation 기반 학습

비선형적

↓

미분가능

↓

학습 가능 (최적화)

종 류

sigmoid

tanh

ReLU

활성화

-1	3	2
0	2	-3
-1	-4	1

max(0,x)

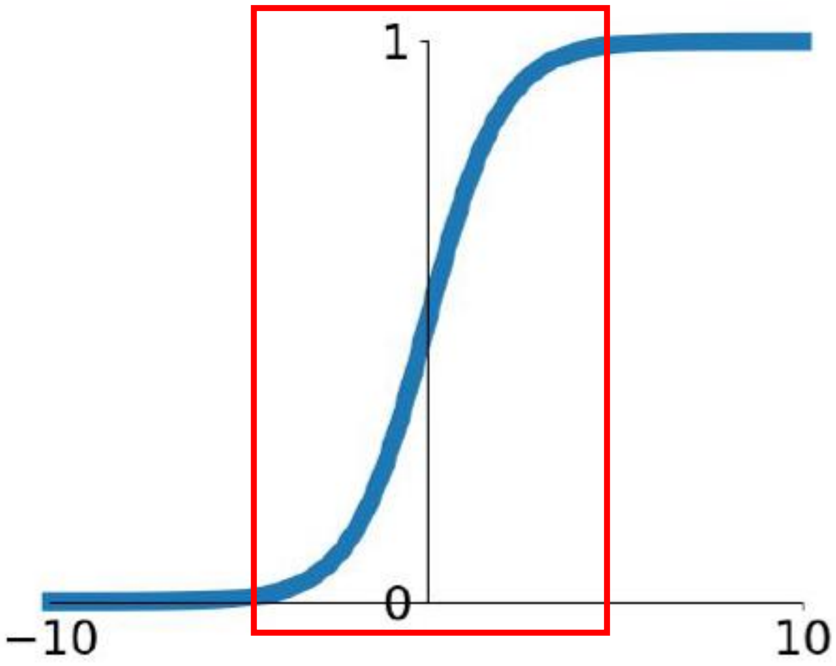
0	3	2
0	2	0
0	0	4

-1	3	2
0	2	-3
-1	-4	1

tanh(x)

-0.762	0.995	0.964
0	0.964	-0.995
-0.762	-0.999	0.762

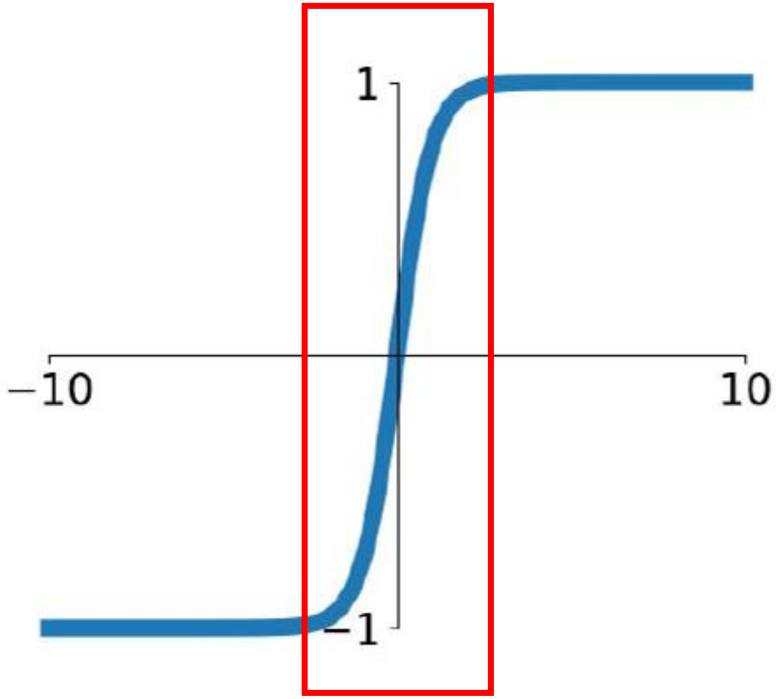
sigmoid



$$\sigma(x) = \frac{1}{1+e^{-x}}$$

input		output
$(-\infty, \infty)$	$\longrightarrow$	$(0, 1)$

tanh



$$\boldsymbol{tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}}$$

input → output

$(-\infty, \infty) \longrightarrow (\mathbf{-1, 1})$

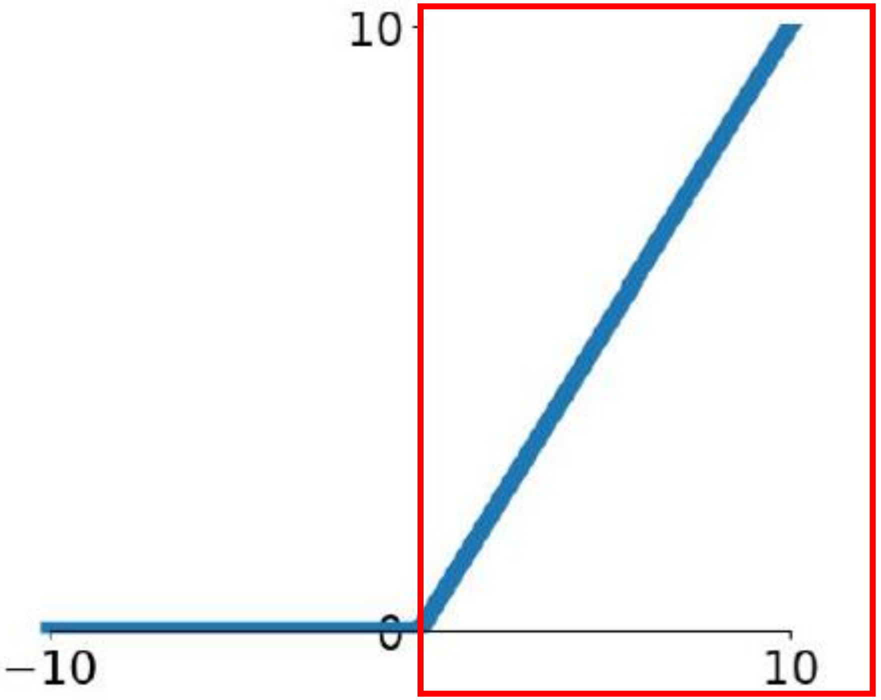
미분 >0

$(-a, a) \longrightarrow$ positive

outside $\longrightarrow$ 0

ReLU

$$\mathrm{ReLU}(x) = \max(x, 0) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$



input

output

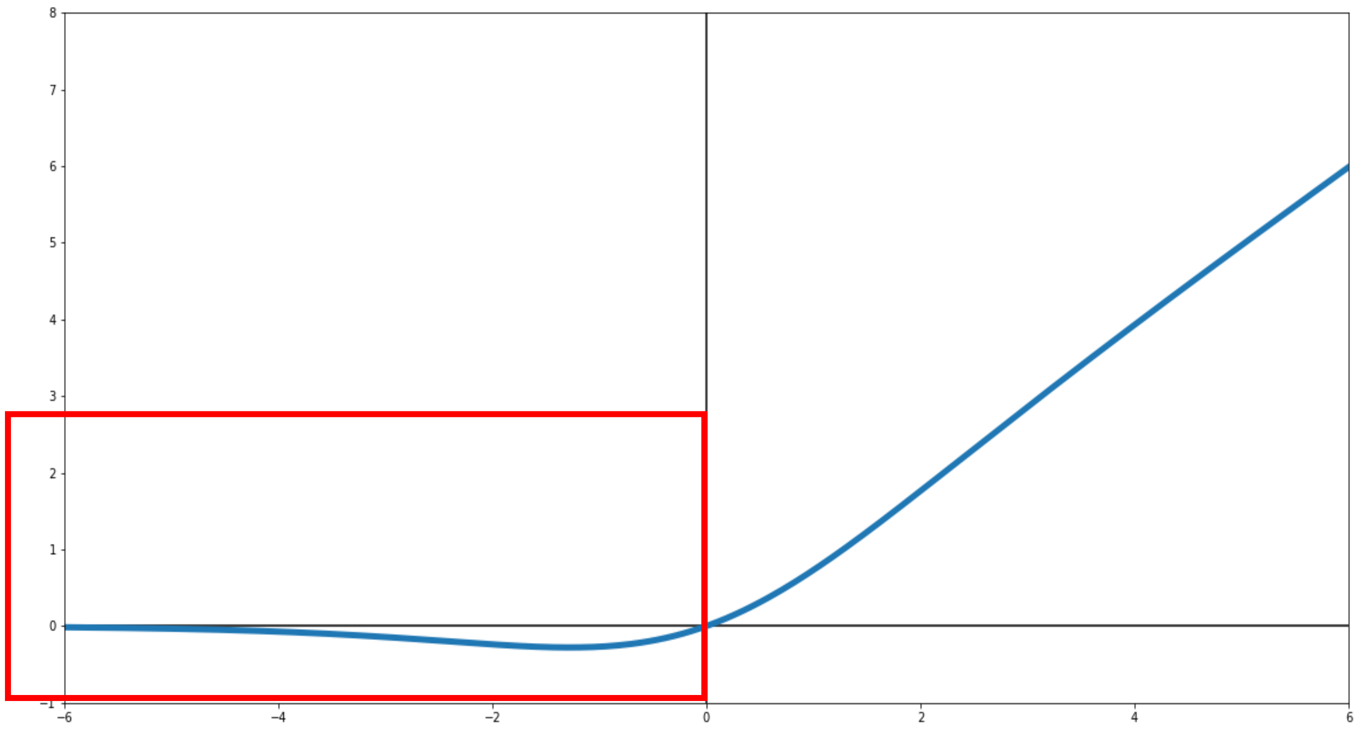
$(-\infty, \infty) \longrightarrow (\mathbf{0}, \infty)$

미분 ≥ 0

$x > 0 \longrightarrow 1$

$x < 0 \longrightarrow 0$

swish



$$swish(x) = x \cdot \sigma(x)$$

input	output
$(-\infty, \infty)$ ⟶	$(\mathbf{0}, \infty)$

$x > 0$ ⟶ $(\mathbf{0}, \infty)$ unbounded

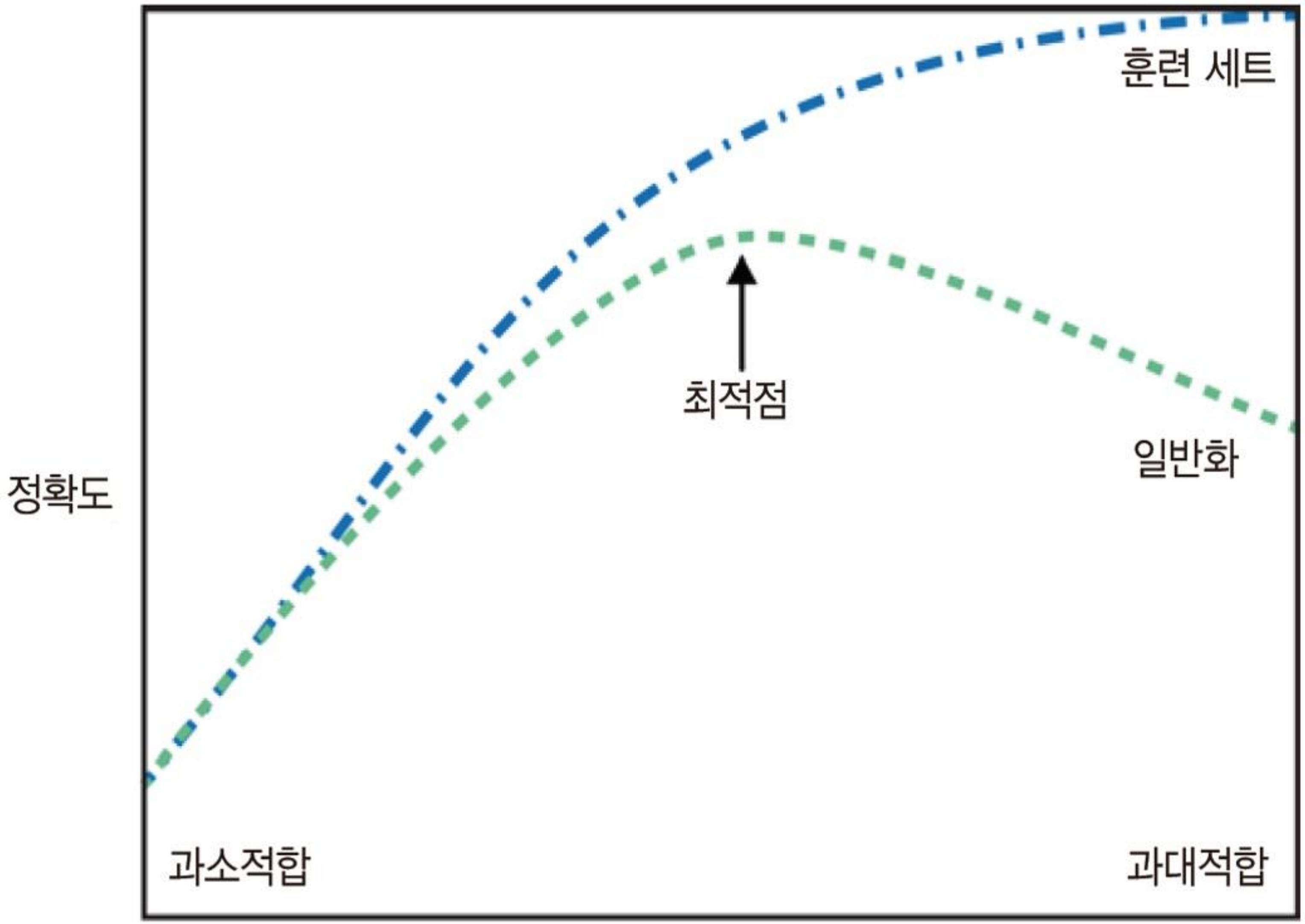
$x < 0$ ⟶ $(\boldsymbol{a}, \mathbf{0})$ bounded

$\boldsymbol{a} \simeq -\mathbf{1.28}$

Contents

5장 신경망 모델

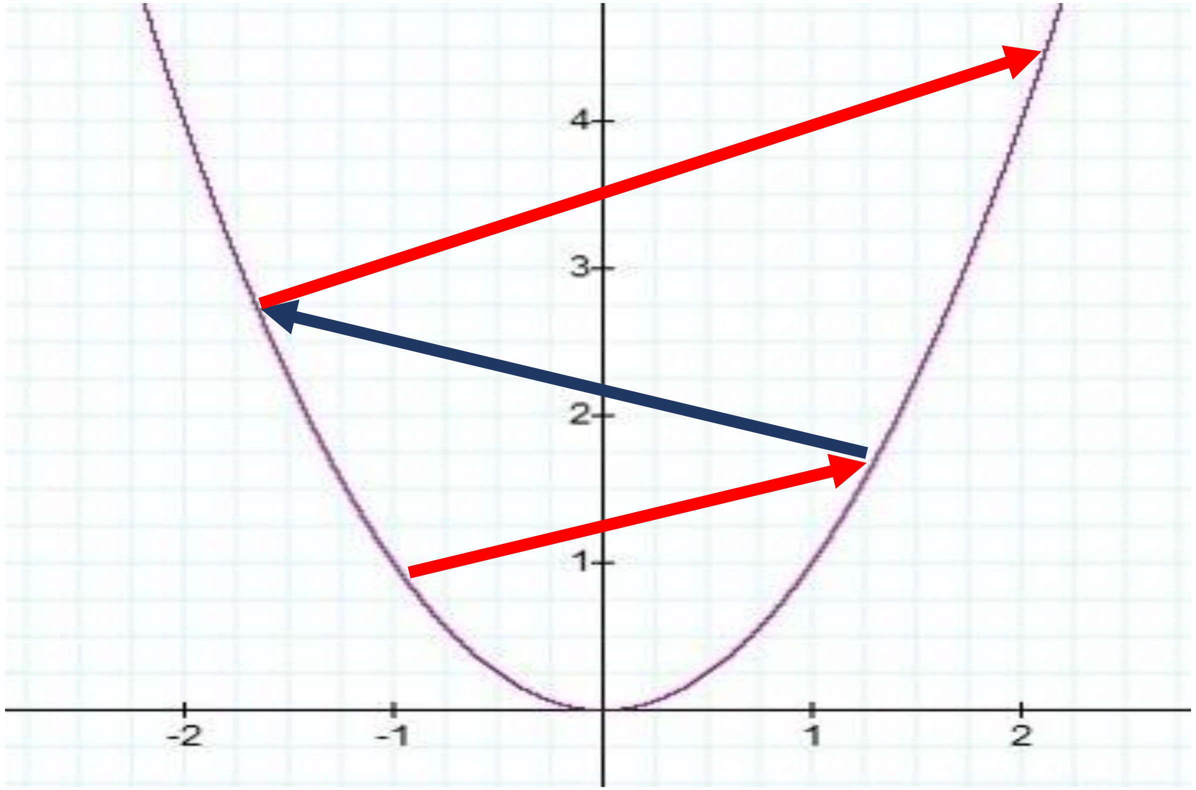
과적합



학습률 Learning Rate

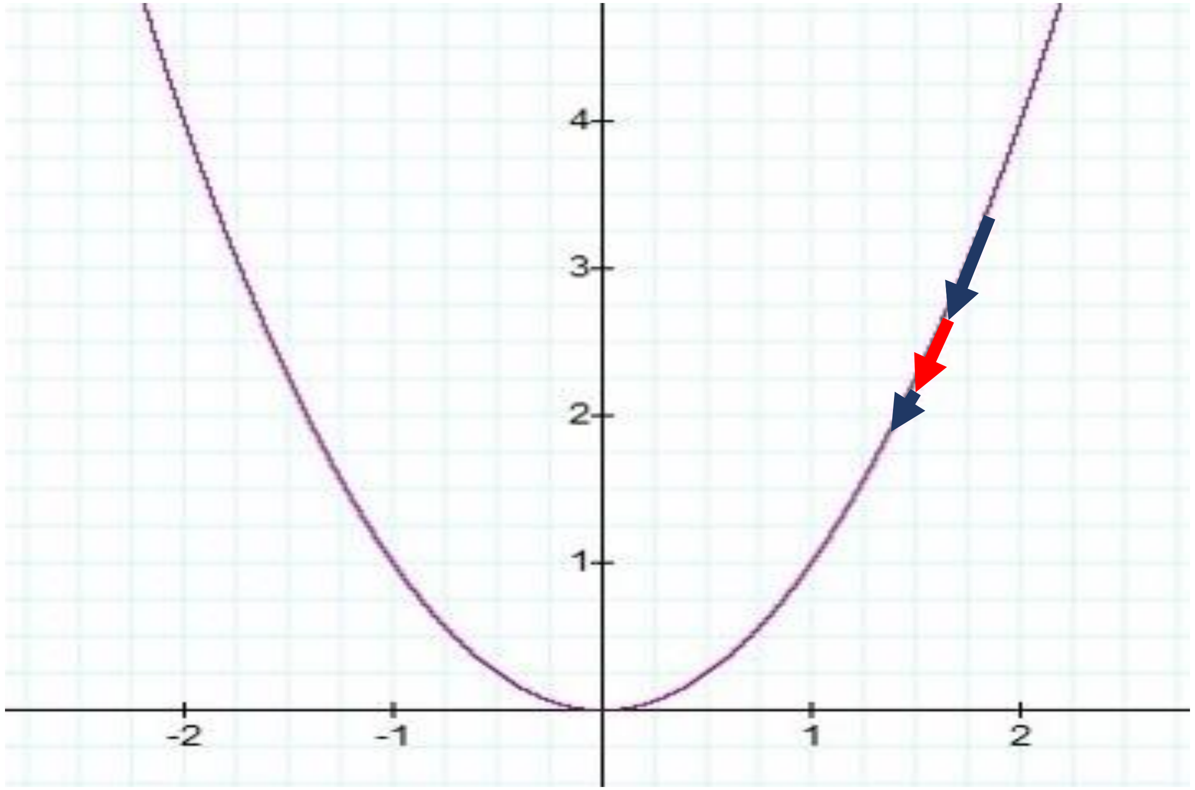
$$W \leftarrow W - \textcolor{red}{\boldsymbol{h}} \frac{dL}{dW}$$

Too big



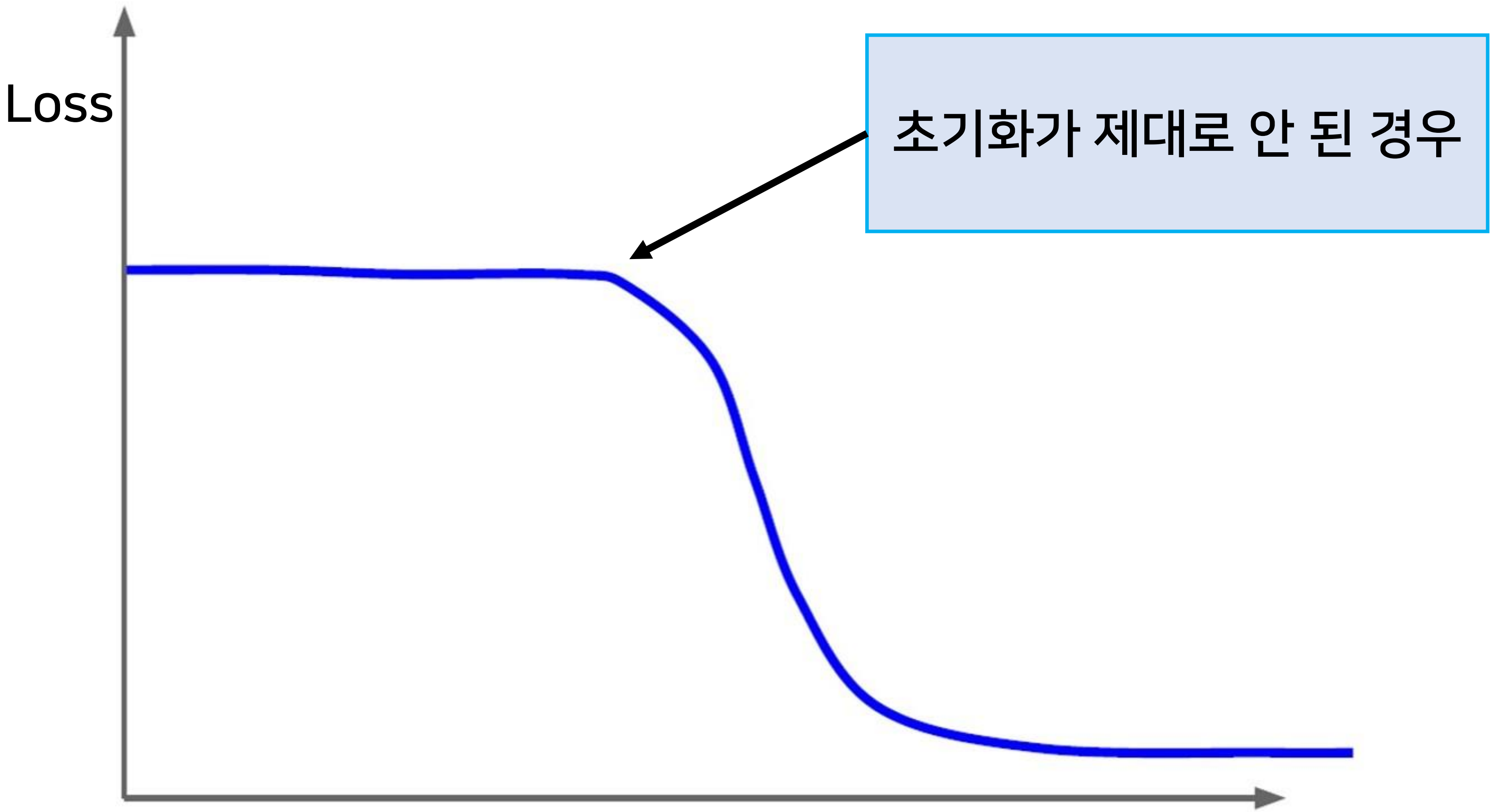
발산 : 학습 불가능

Too small

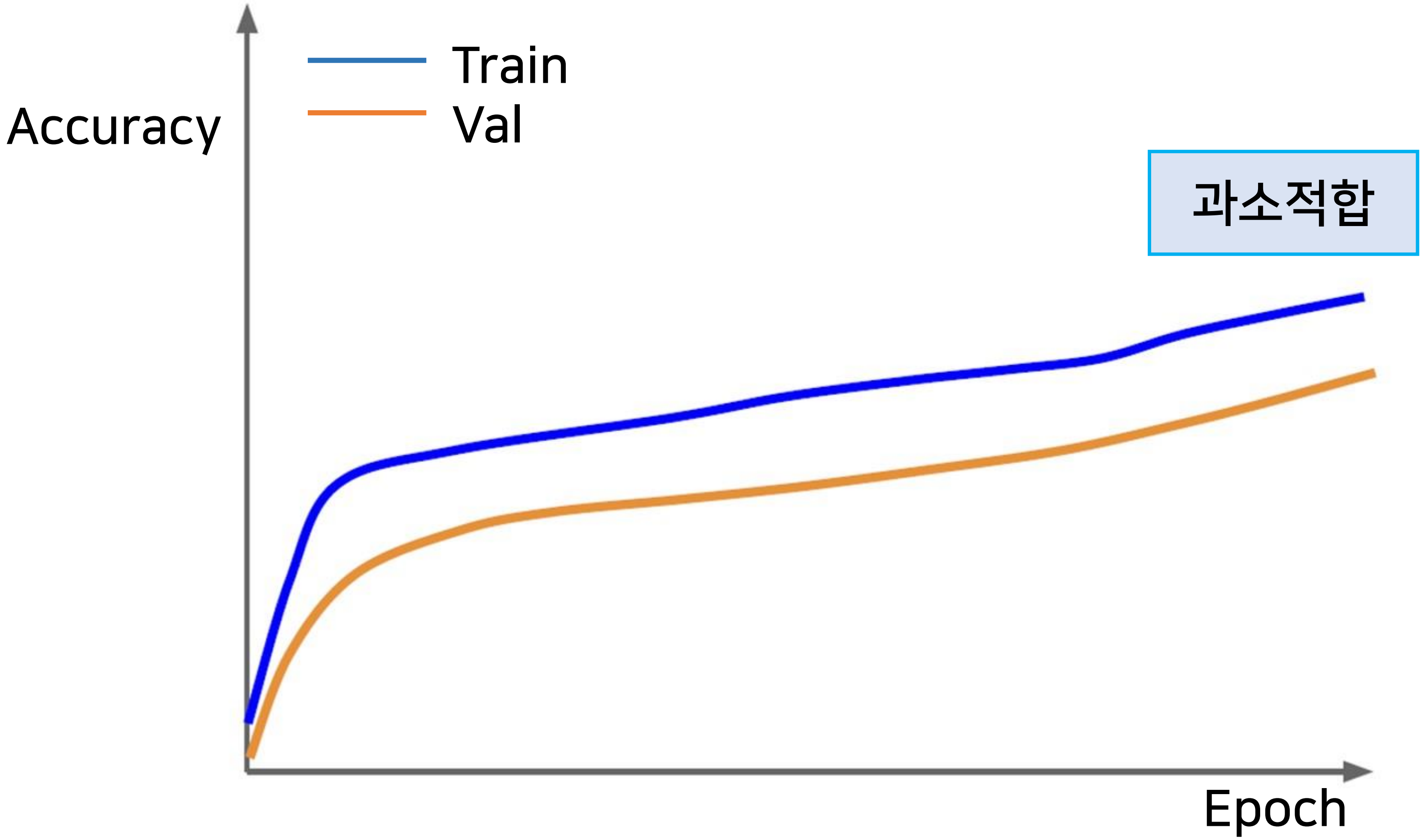


정체 : 학습 불가능

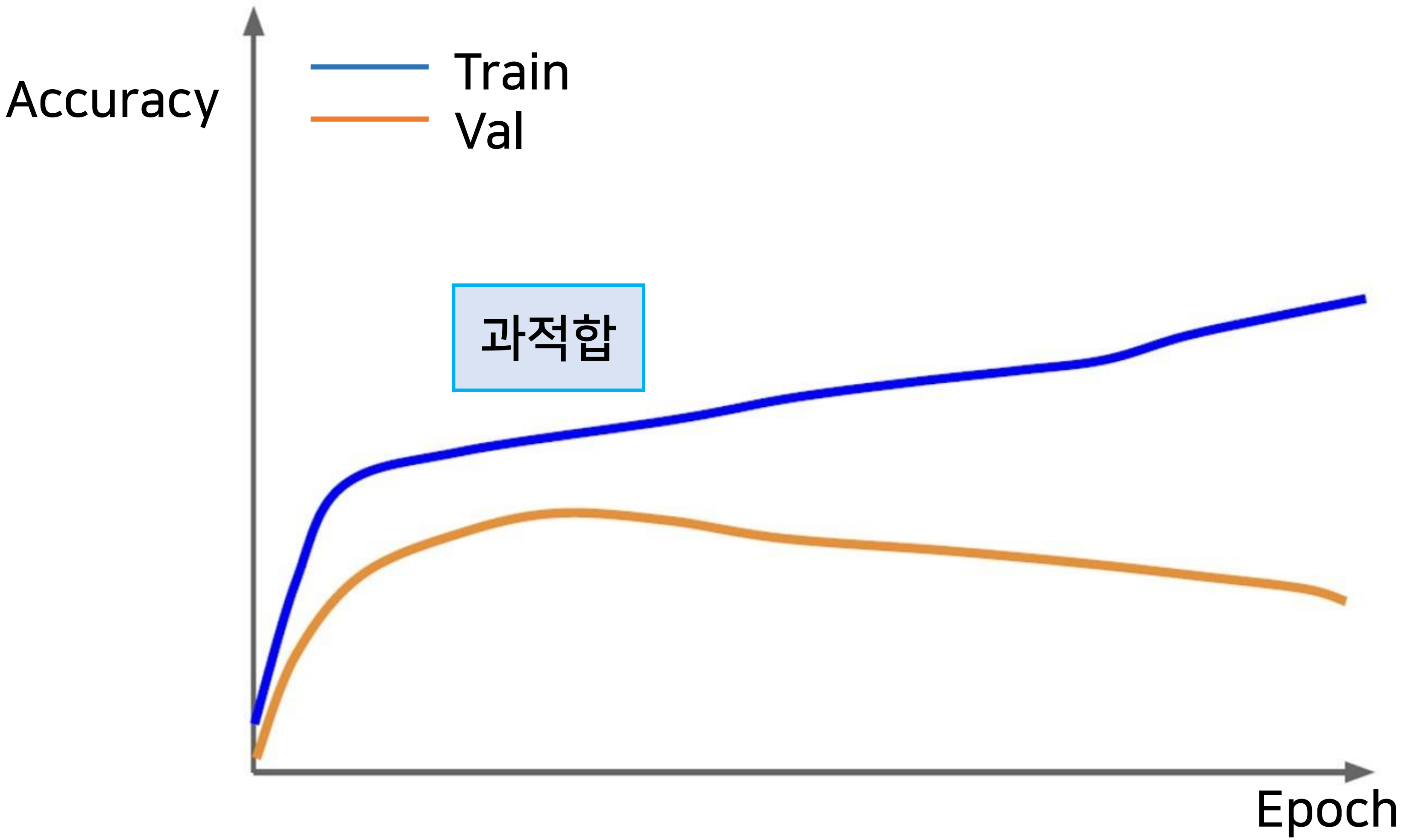
학습 분석: 초기화



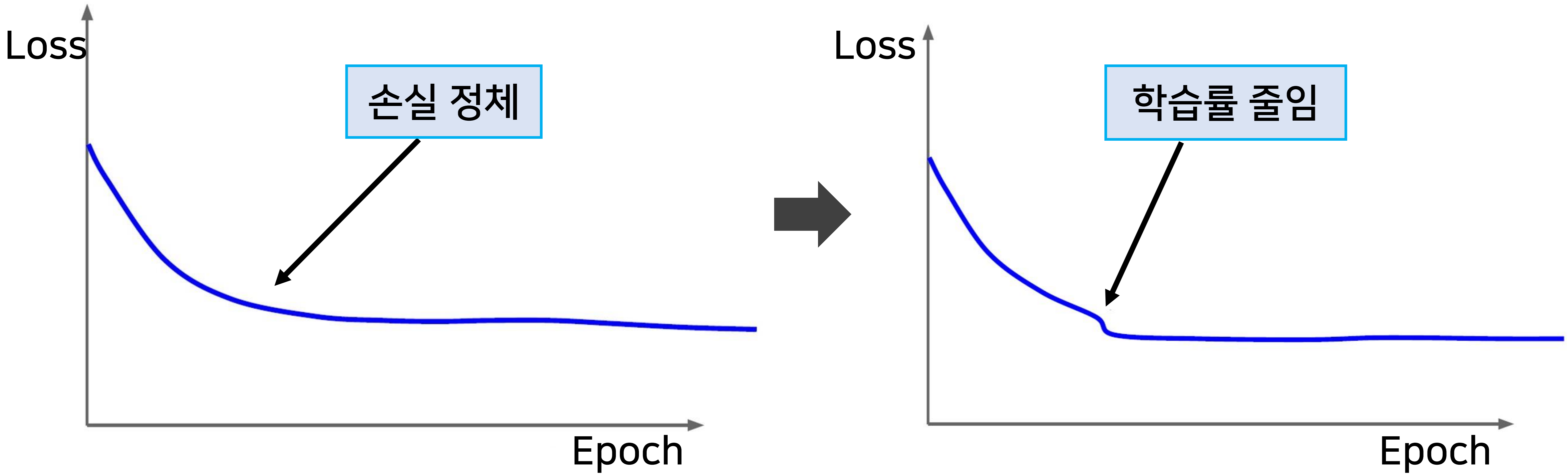
학습 분석: 과소적합



학습 분석: 과적합



학습 분석: 학습 정체



Contents

5장 신경망 모델

오류역전파 기본 공식

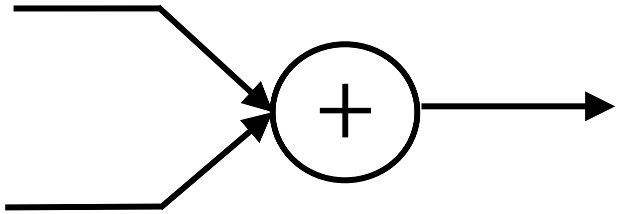
$$\left[\frac{\partial f}{\partial x}\right] = \left[\frac{\partial f}{\partial q}\right] \cdot \left[\frac{\partial q}{\partial x}\right]$$

$$\boxed{\text{gradient}} = \boxed{\text{Upstream gradient}} \times \boxed{\text{local gradient}}$$

오류역전파 과정

$$f(x, y, z) = (x + y)z$$

Computational Graph 작성 → 단계별 함수와 도함수 계산 → 함수값 계산 Forward → 미분값 계산 Backward



$$g(x) = x + a$$

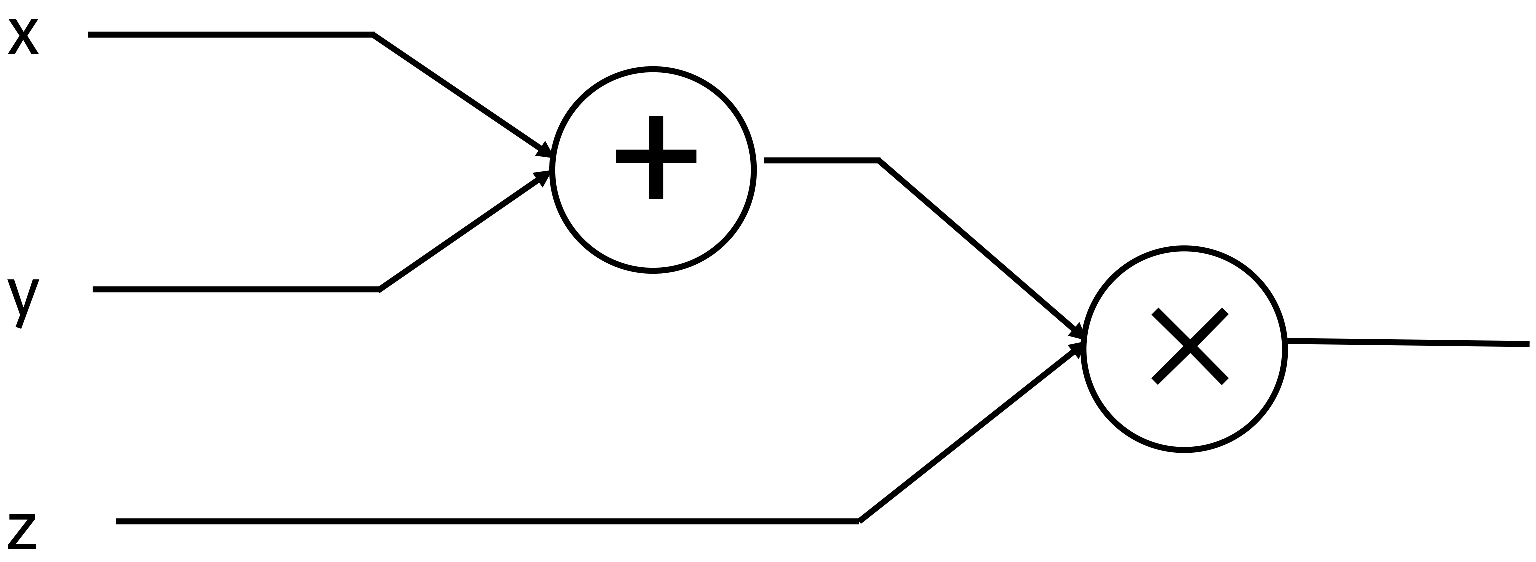
$$g'(x) = 1$$

$$f(1,2,1) = 3$$

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$$

오류역전파 예제 : 1. Computational Graph

$$f(x, y, z) = (x + y)z$$



오류역전파 예제 : 2 단계별 함수와 도함수 계산

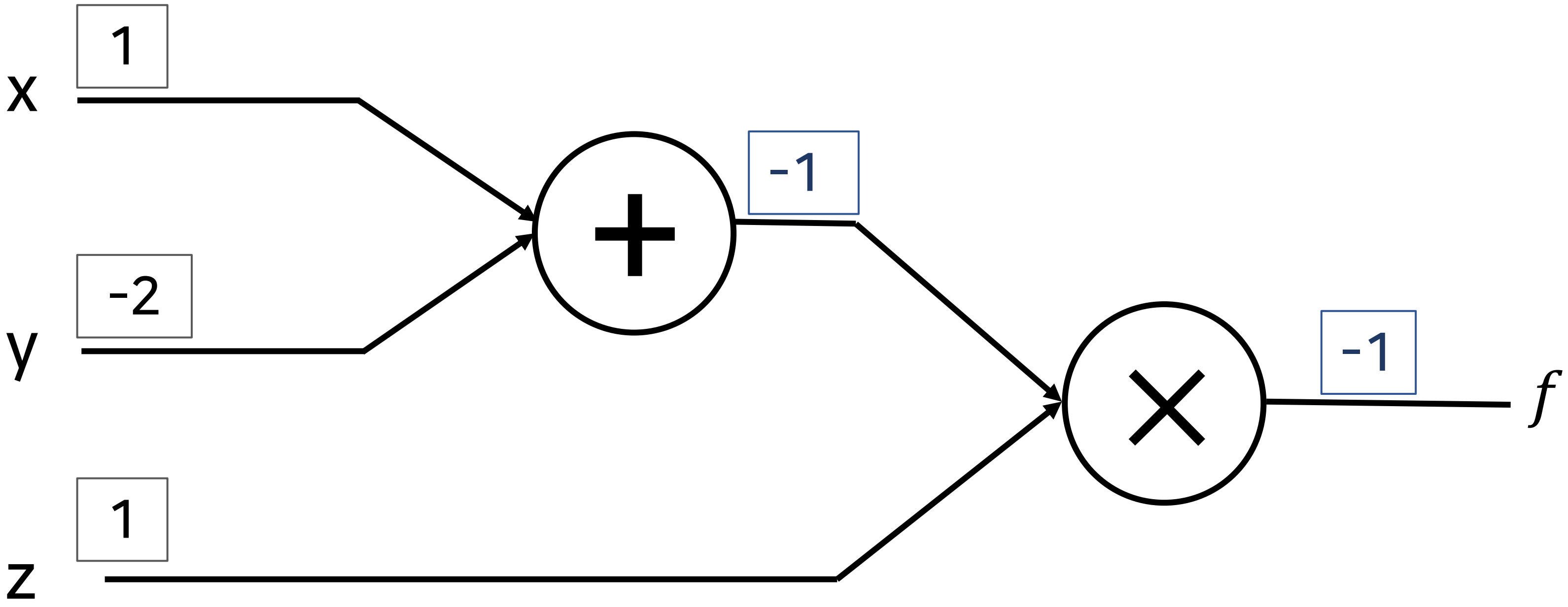
$$f(x, y, z) = (x + y)z$$

함수1: $\oplus$ $g(x) = x + a$ ➡ $g'(x) = 1$

함수2: $\otimes$ $h(x) = ax$ ➡ $h'(x) = a$

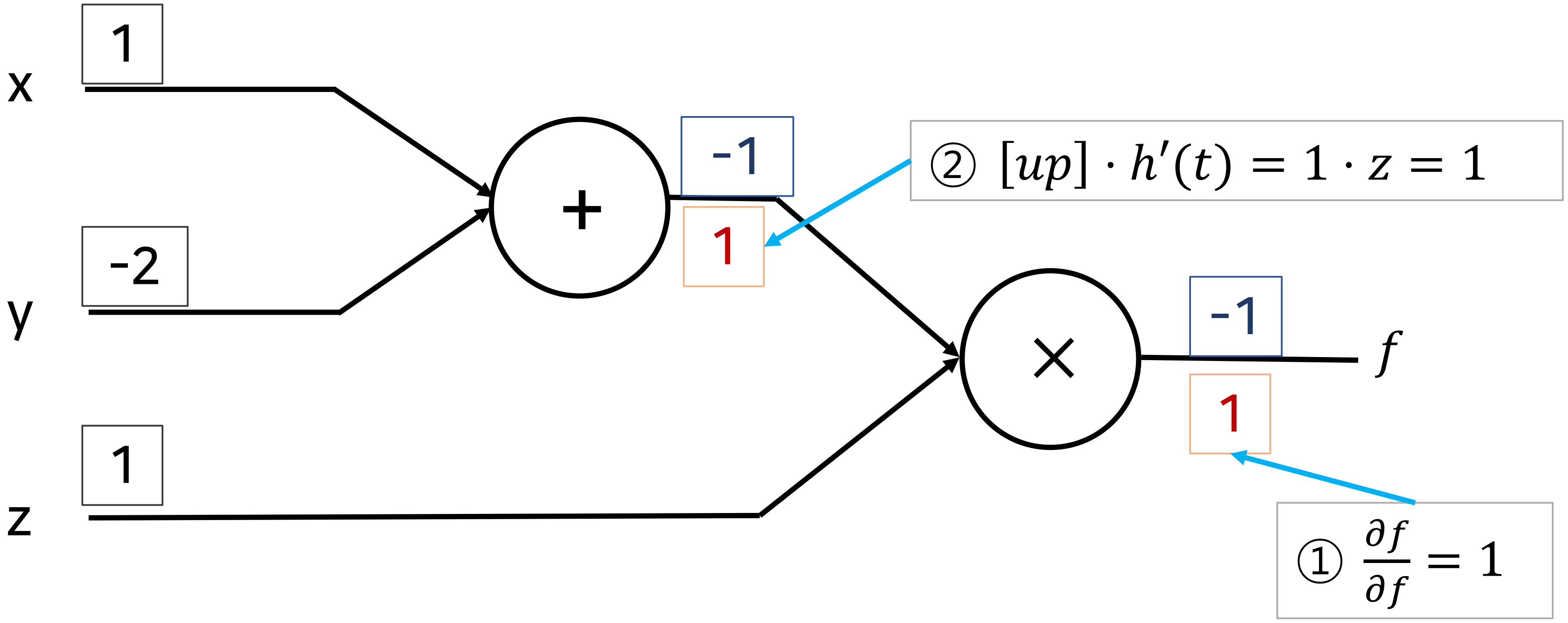
오류역전파 예제 : 3. 함수값 계산

$$f(x, y, z) = (x + y)z$$



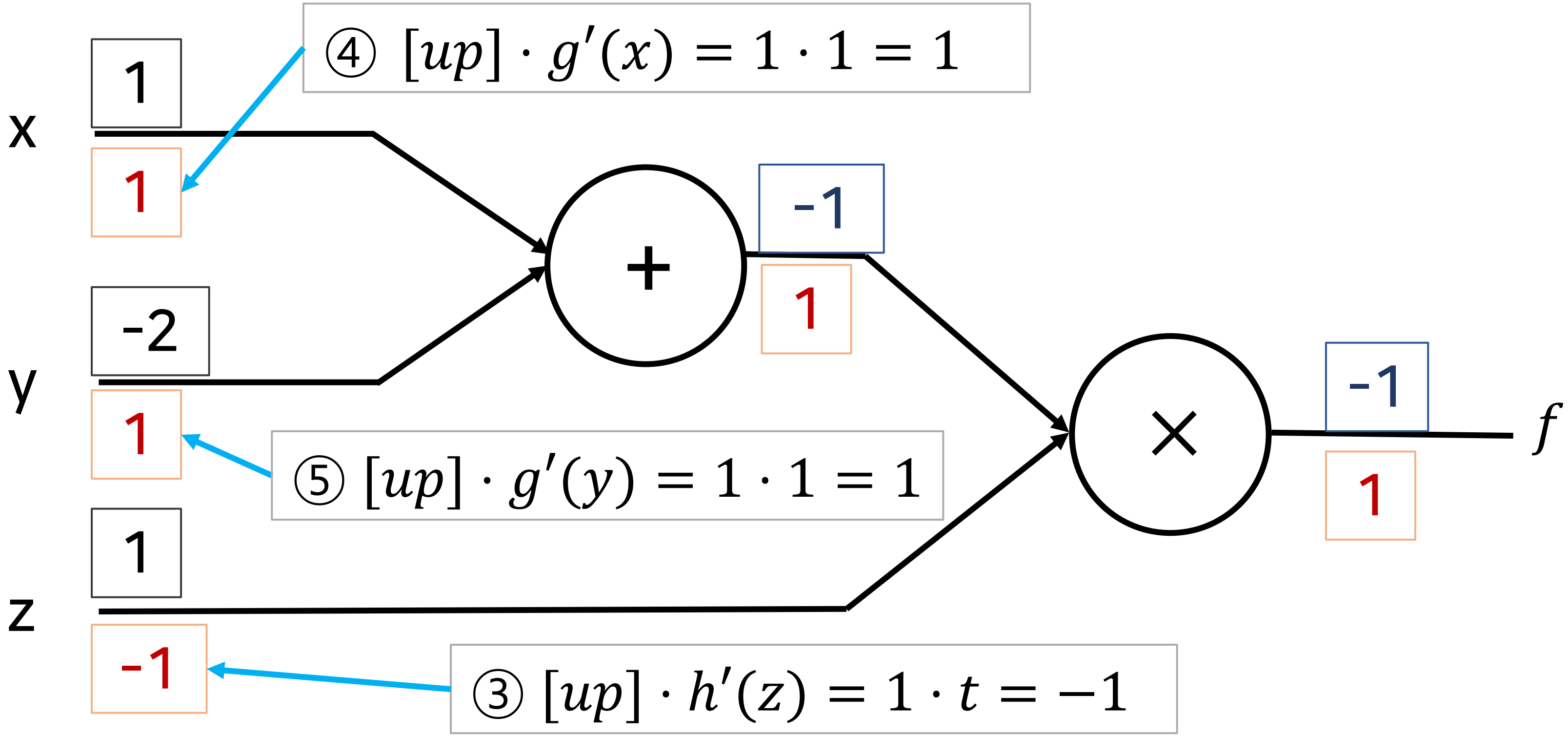
오류역전파 예제 : 4.미분값 계산1

$$f(x, y, z) = (x + y)z$$



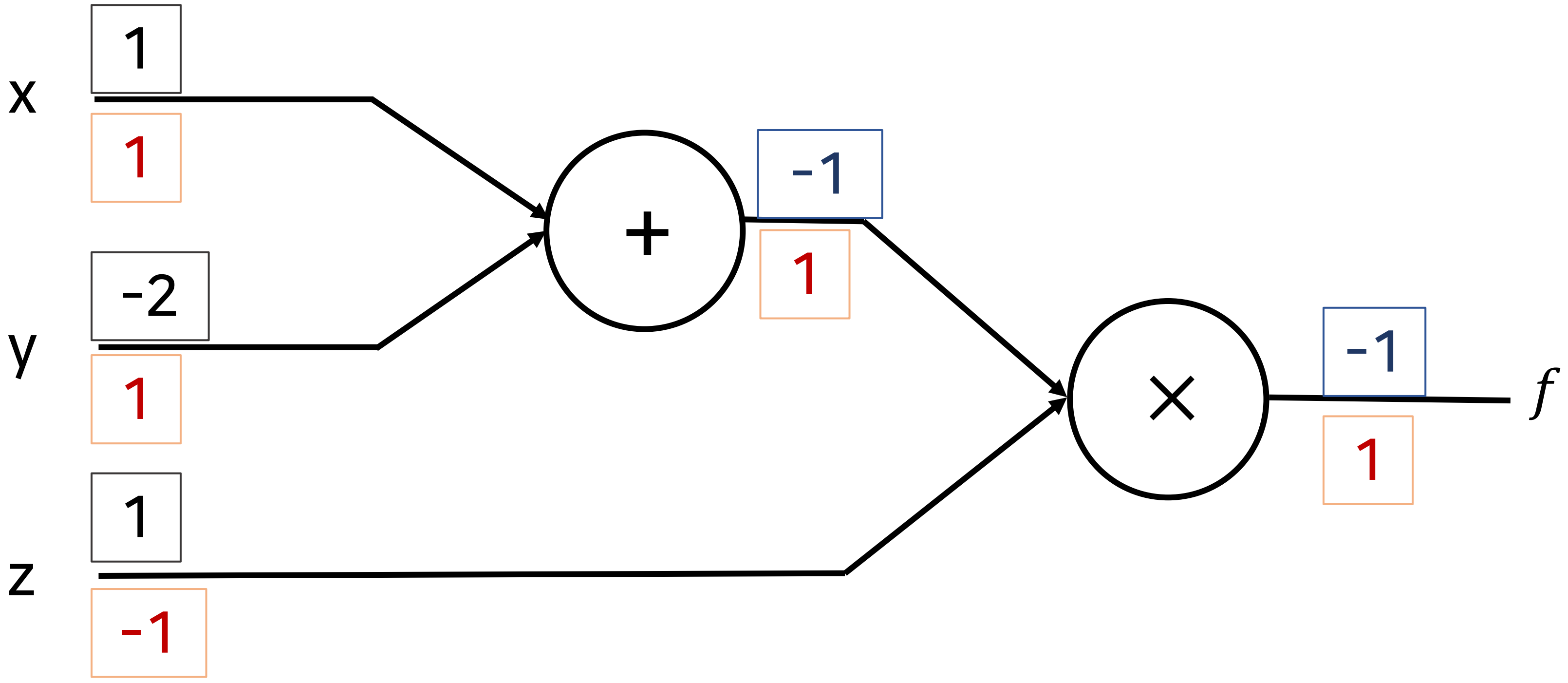
오류역전파 예제 : 4.미분값 계산2

$$f(x, y, z) = (x + y)z$$



오류역전파 예제 : 4.미분값 계산3

$$f(x,y,z) = (x+y)z$$



$$\Rightarrow \quad Df(1,-2,1) = [1,1,-1]$$

오류역전파 문제

$$f(\mathbf{w}, \mathbf{x}) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$

$$\mathbf{x} \xrightarrow{\text{선형분류기}} \mathbf{W}\mathbf{x} + \mathbf{b} \xrightarrow[\text{sigmoid}]{\text{활성화}} \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

$\mathbf{x} = [x_0, x_1]$

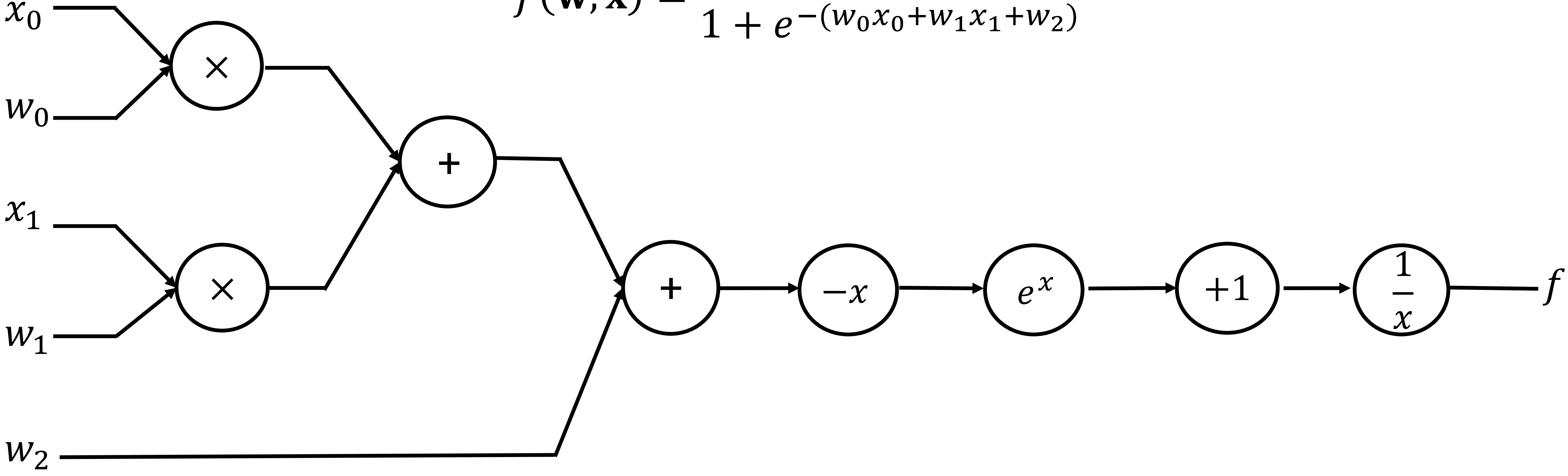
$\mathbf{W} = [w_0, w_1]$

$\mathbf{b} = [w_2]$

$\Rightarrow Df$?

1. Computational Graph

$$f(\mathbf{w},\mathbf{x}) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



2. 단계별 함수와 도함수

$$f(\mathbf{w}, \mathbf{x}) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$

함수1: $h(x) = ax$ ➡ $h'(x) = a$

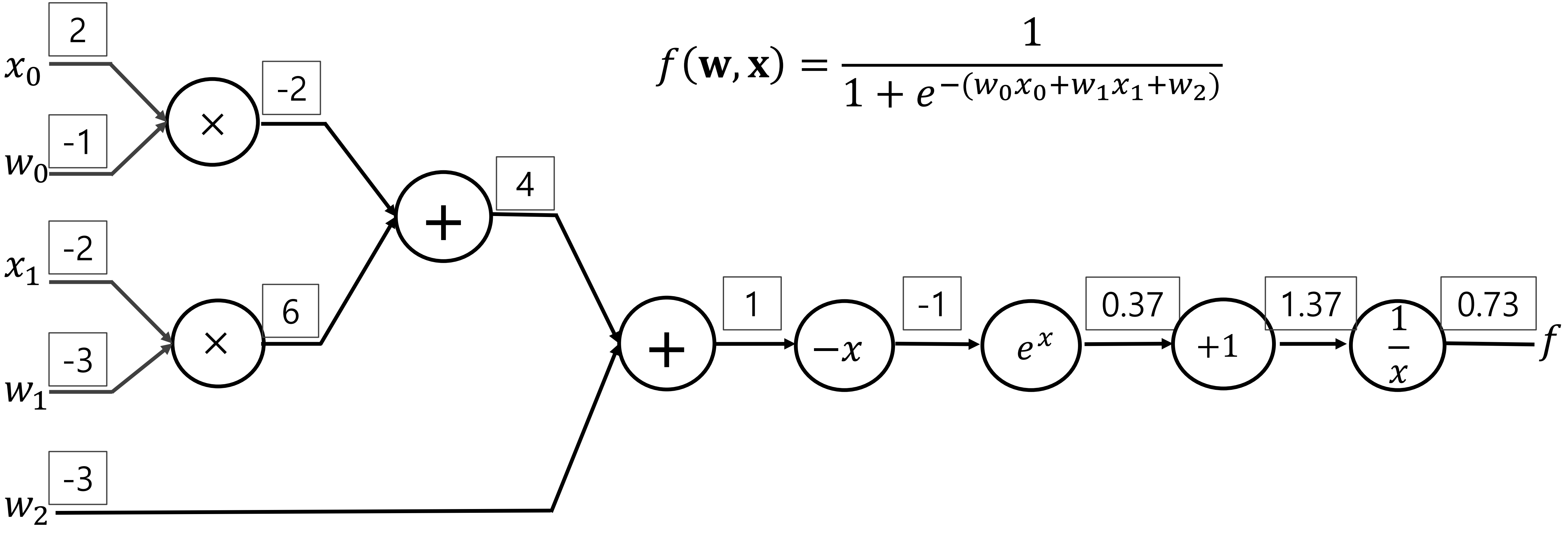
함수2: $g(x) = x + a$ ➡ $g'(x) = 1$

함수3: $k(x) = e^x$ ➡ $k'(x) = e^x$

함수4: $m(x) = \frac{1}{x}$ ➡ $m'(x) = \frac{-1}{x^2}$

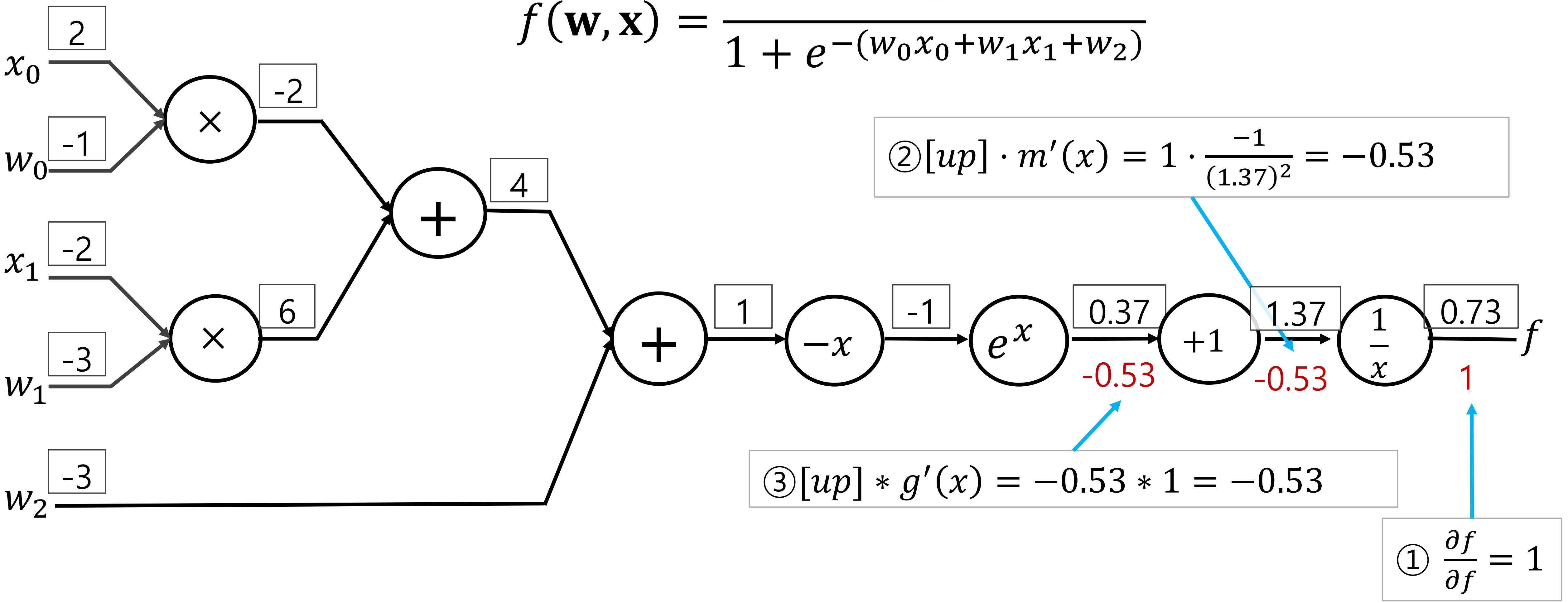
3. 함수값 계산

$$f(\mathbf{w}, \mathbf{x}) = \frac{1}{1 + e^{-(w_0x_0+w_1x_1+w_2)}}$$



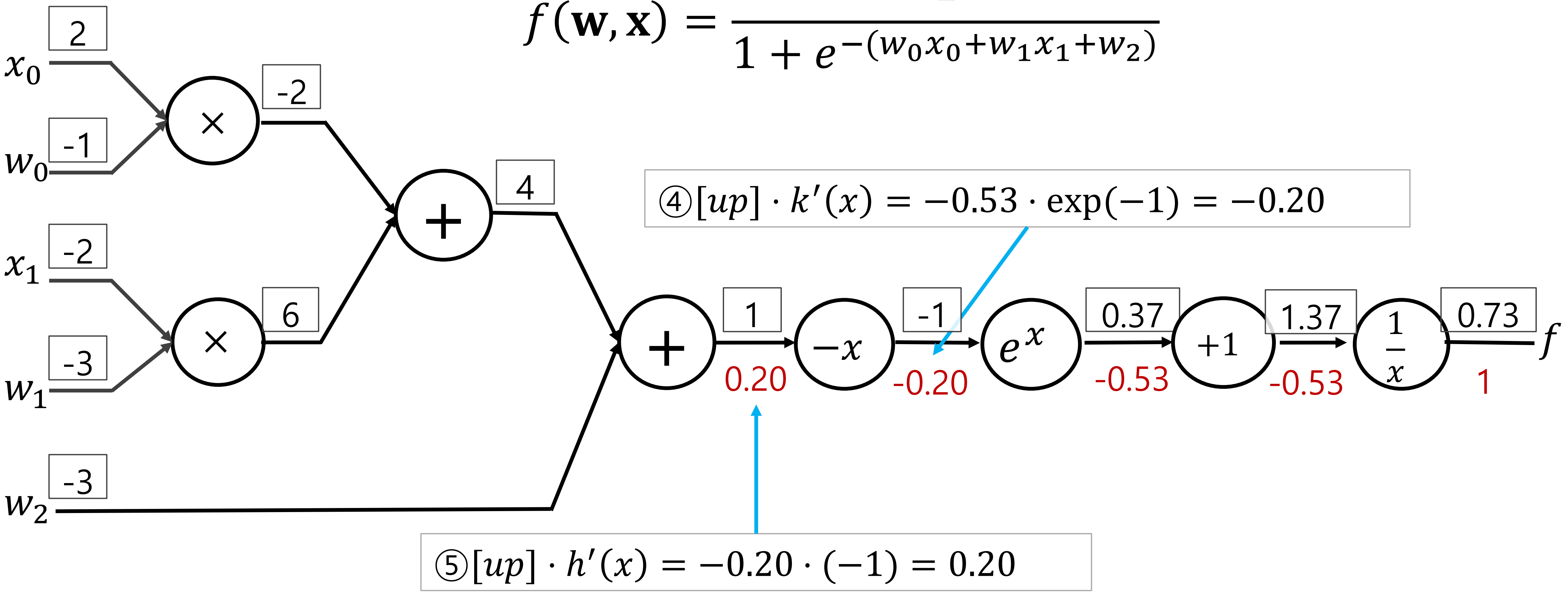
4. 미분값 계산 1

$$f(\mathbf{w},\mathbf{x}) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



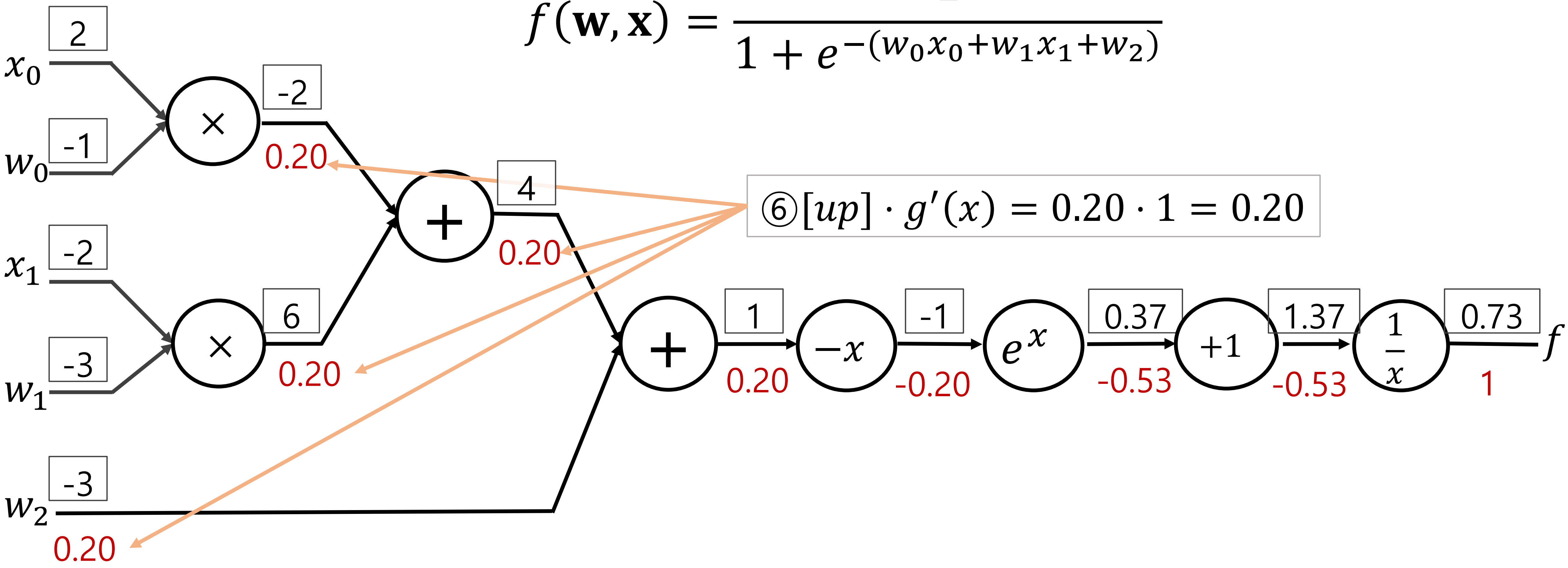
4. 미분값 계산 2

$$f(\mathbf{w}, \mathbf{x}) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$



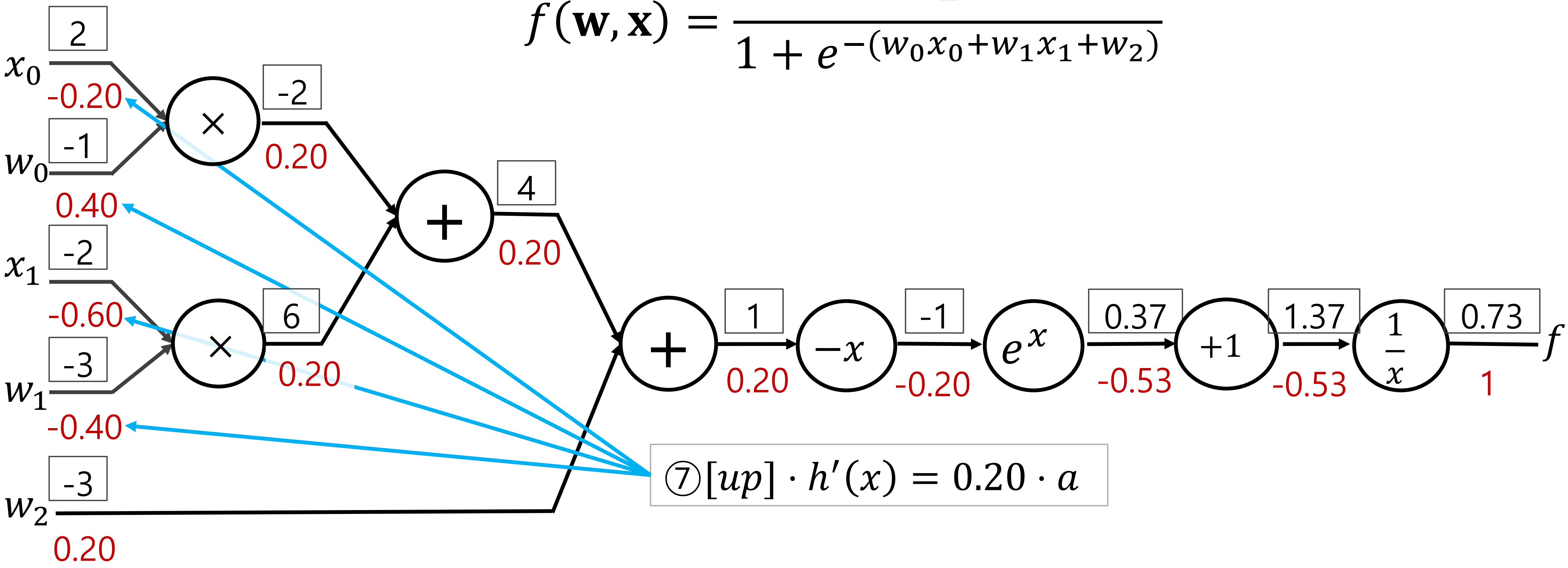
4. 미분값 계산 3

$$f(\mathbf{w},\mathbf{x}) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



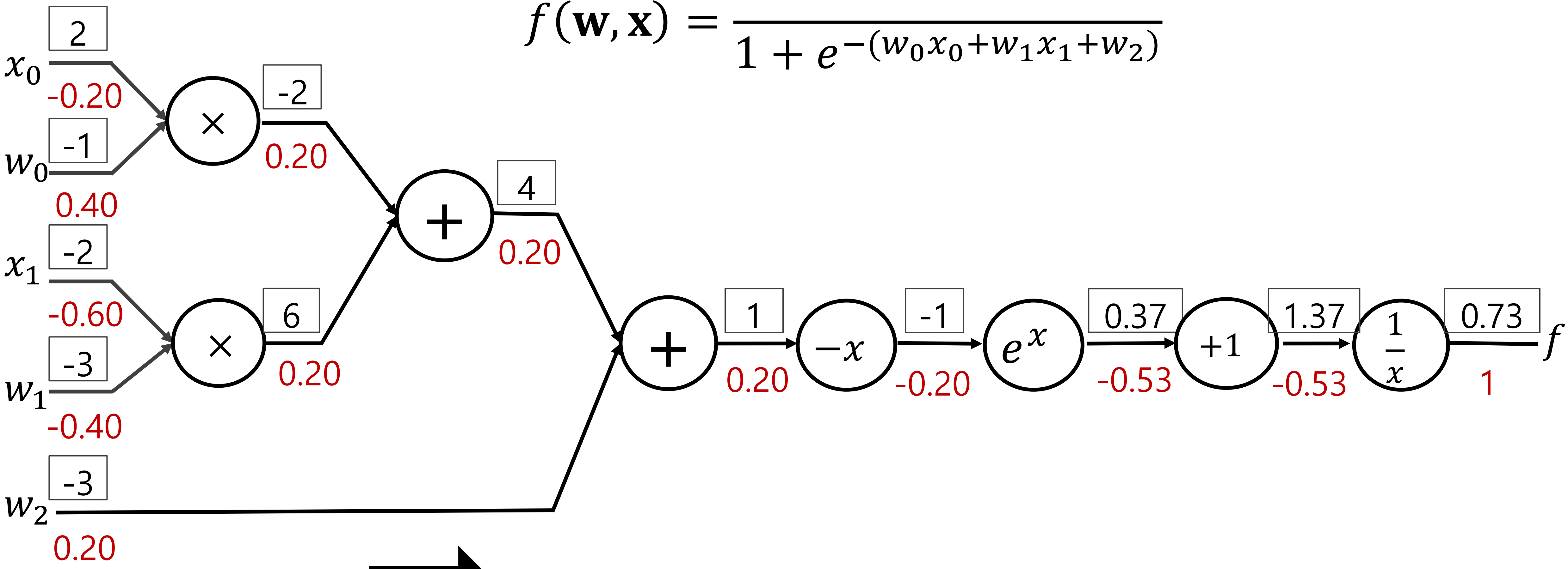
4. 미분값 계산 4

$$f(\mathbf{w},\mathbf{x}) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



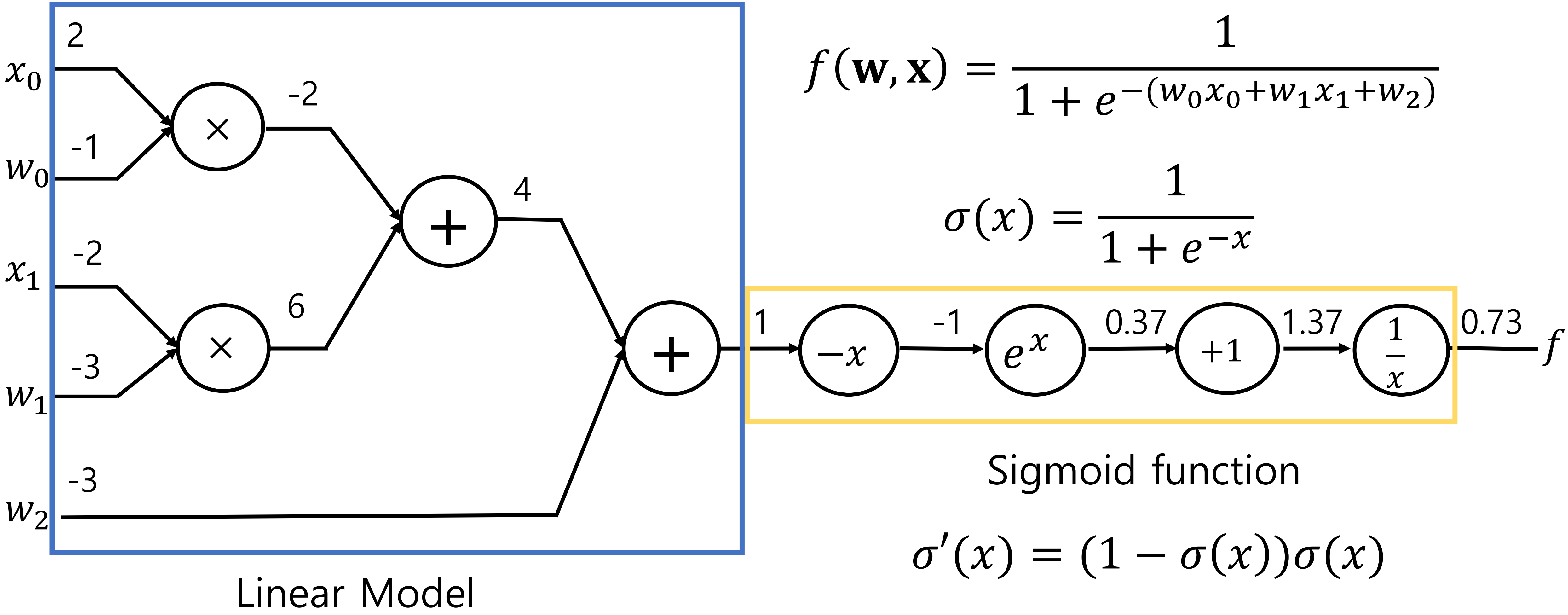
4. 미분값 계산 4

$$f(\mathbf{w},\mathbf{x}) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$Df(-1,-3,-3) = [0.4, -0.4, 0.2]$$

선형모델과 sigmoid 활성화



$$f(\mathbf{w}, \mathbf{x}) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

$$\sigma'(x) = (1 - \sigma(x))\sigma(x)$$

Contents

5장 신경망 모델

규제 강화 Regularization

훈련 오류가 아닌, **일반화 오류**를 줄이려는 의도를 가지고 **학습 알고리즘**을 수정하는 모든 방법

과적합 해소

손실 추가

$+R(W)$

배치 정규화

데이터 증강

모델 단순화

Dropout

DropConnect

Fractional MaxPooling

Stochastic Depth

Cutout/Mixup

규제 강화 종류

1. 손실 추가

2. 데이터 증강

3. Dropout

모델 단순화

규제 강화 : 1. 손실 추가

$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \boxed{R(W)}$$

Lasso(L_1) : $R(W) = \lambda \sum_i \sum_j |W_{i,j}|$

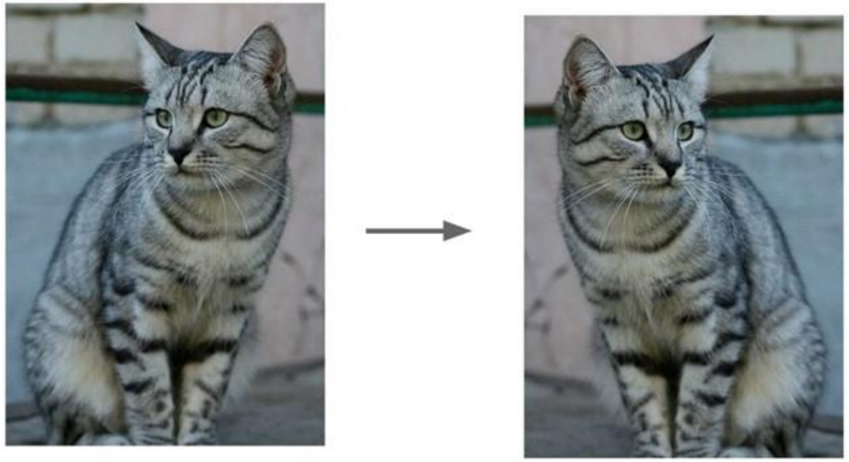
Ridge(L_2) : $R(W) = \lambda \sum_i \sum_j W_{i,j}^2$

Elastic Net ($L_1 + L_2$): $R(W) = \sum_i \sum_j \lambda_1 \left|W_{i,j}\right| + \lambda_2 W_{i,j}^2$

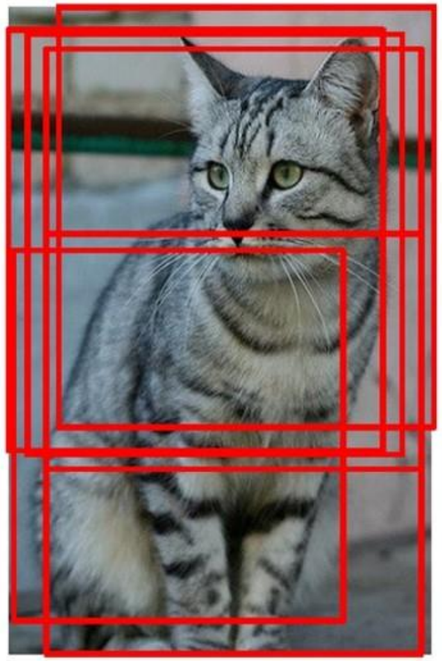
$\lambda, \lambda_1, \lambda_2$:규제 강도
Hyper-parameter

규제 강화 : 2. 데이터 증강

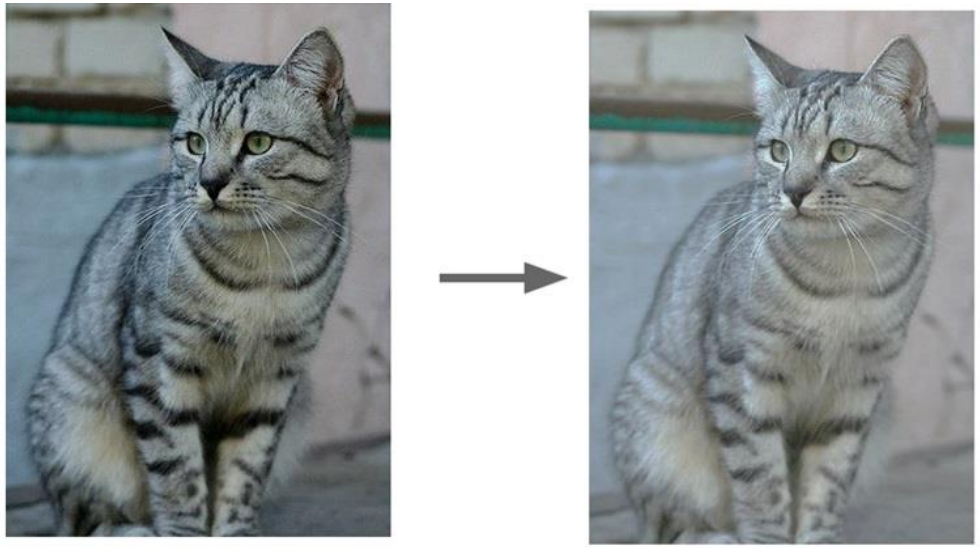
이미지 반전



자르기 & 크기 조절



색상 변조

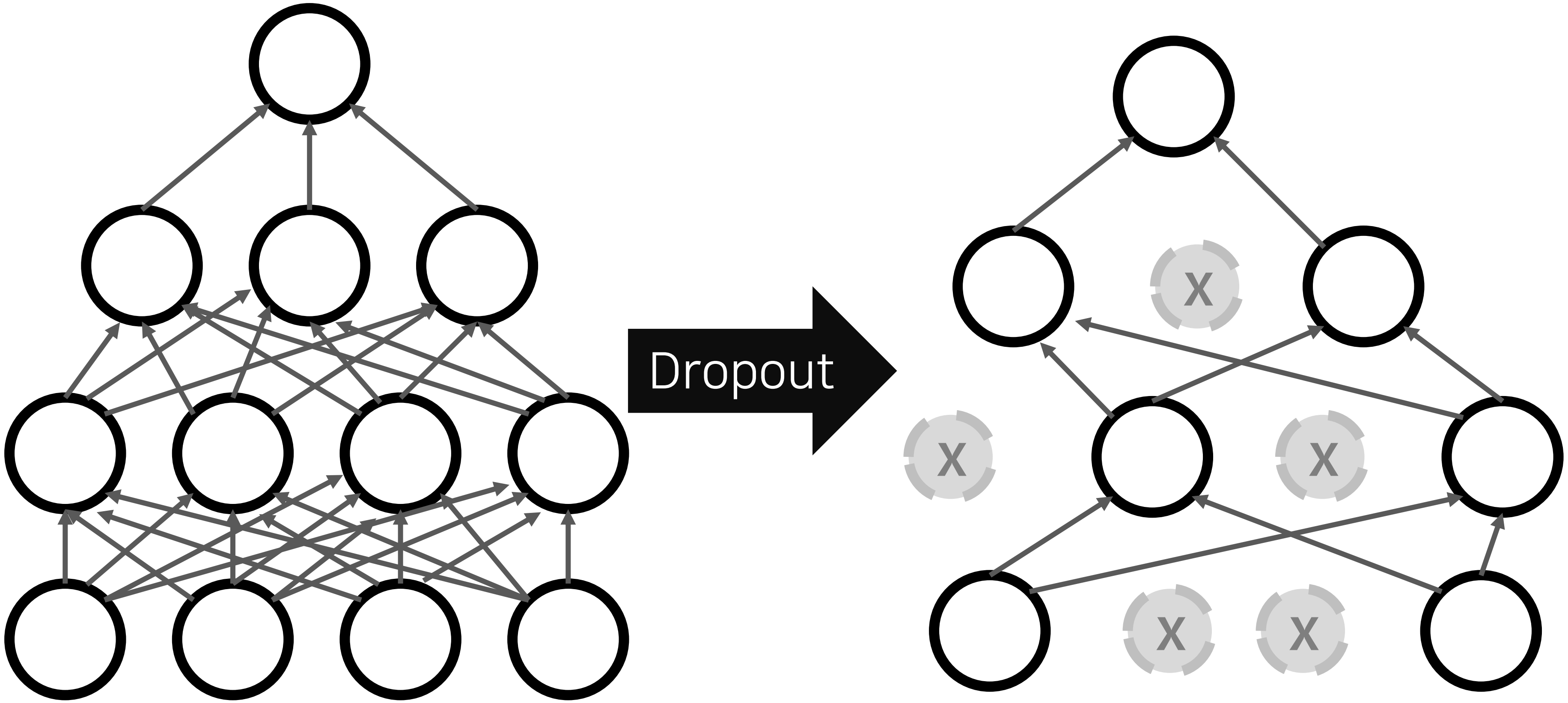


색조 밝기 임의로 변조

Translation
Rotation
Stretching

Shearing
Lens
Distortion

규제 강화 : 3. Dropout (모델 간소화)



신경망 모델

Drop 확률 0.5

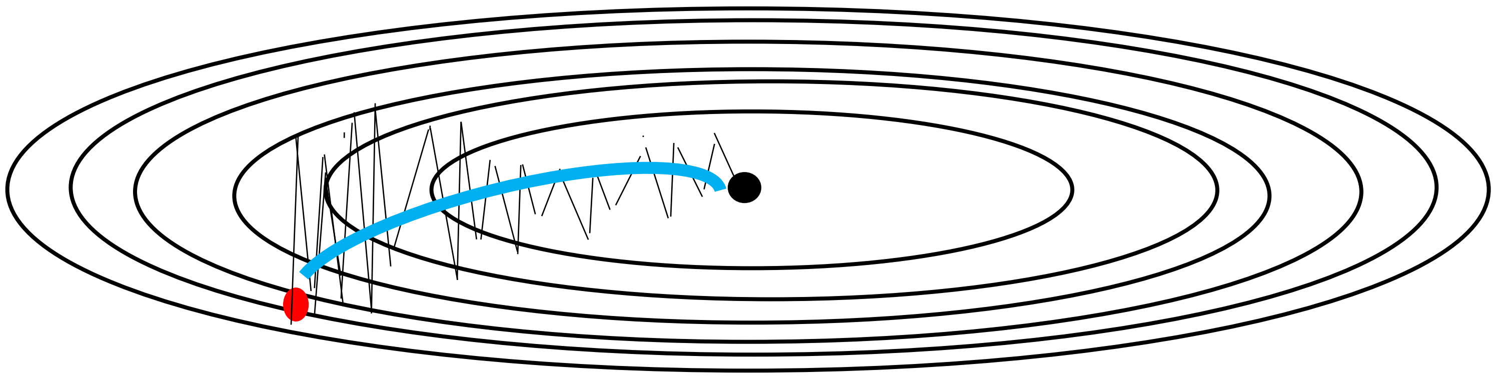
Srivastava et al, "Dropout: A simple way to prevent neural networks from overfitting", JMLR 2014

Contents

5장 신경망 모델

SGD 한계

Poor Conditioning

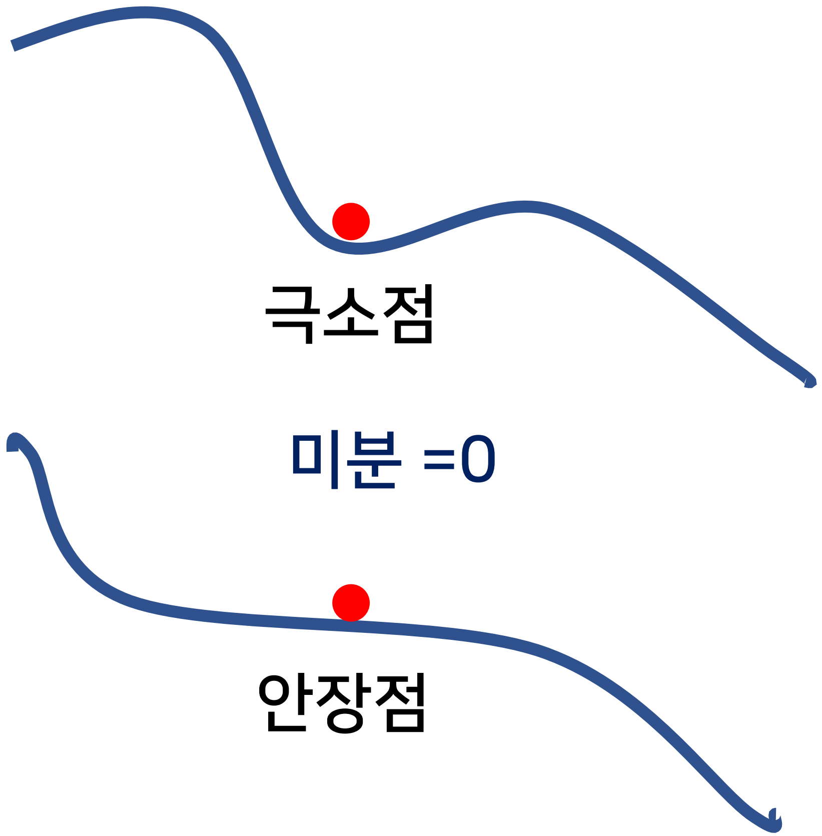


SGD 한계

→ ??

경사하강 진행 불가능

고차원: 안장점이 매우 흔한 현상



SGD 계산 방법

GD

모든 데이터를
한꺼번에 계산

SGD

Minibatch 사용
시간 성능 개선

스텝 방향 추가

Momentum

스텝 계산해서 움직인 후
관성 방향 추가

NAG Nesterov Accelerated Gradient

관성 방향으로 먼저
움직인 후 스텝 계산

스텝사이즈 변경

Adagrad

처음은 빠르게
나중은 세밀히 탐색

RMSProp

보폭을 줄일 때
이전 맥락 상황 반영

\+

Adam

RMSProp + Momentum
방향도 스텝사이즈도 적절하게 !

Contents

5장 신경망 모델

실습예제

신경망 Mnist

신경망 coding / data loading

numpy 호출

```
import numpy as np
import pandas as pd
```

data load

```
from tensorflow.keras.datasets.mnist import load_data
```

```
(train_x, train_y), (test_x, test_y) = load_data()
```

```
train_x.shape, train_y.shape
```

신경망 coding / data 확인

data 확인

```
from PIL import Image
img=train_x[0]

import matplotlib.pyplot as plt
img1=Image.fromarray(img, mode='L')
plt.imshow(img1)

train_y[0]    #첫번째 데이터 label 확인
```

신경망 coding / data 전처리

data 전처리

벡터화

```
train_x1=train_x.reshape(60000,-1)

test_x1=test_x.reshape(10000,-1)
```

크기 조절

```
train_x2=train_x1/255

test_x2=test_x1/255
```

신경망 coding / 모델 설정

모델 설정

```
from keras.models import Sequential
from keras.layers import Dense
```

```
md=Sequential()  # 모델명을 md로 정의
md.add(Dense(128,activation='relu', input_shape=(784,))) #28*28
md.add(Dense(64,activation='relu'))
md.add(Dense(10,activation='softmax’))
      #분류문제이므로 마지막 층은 softmax로 활성화
md.summary()
```

신경망 coding / 모델 그리기

Model 그리기

```
from tensorflow.keras.utils import plot_model

plot_model(md) #, to_file='./model.png') #파일로 저장
```

신경망 coding / 학습 설정

compile/fit

```
md.compile(loss='sparse_categorical_crossentropy',  optimizer='sgd’,
           metrics='acc')
hist=md.fit(train_x, train_y, epochs=30, batch_size=64,
           validation_split=0.2)
```

학습 분석 그래프

```
acc=hist.history['acc']
val_acc=hist.history['val_acc']
epoch=np.arange(1,len(acc)+1)
plt.figure(figsize=(10,8))
plt.xlim(250,len(acc)+1)
plt.plot(epoch,acc, 'b',label='acc')
plt.plot(epoch, val_acc, 'g', label='val_acc')
plt.legend()
```

선형분류 coding 평가

평가

```
md.evaluate(test_x2, test_y)
```

#0.9757999