

c:\>git clone <https://github.com/yungbyun/mllecture.git>

AI and Deep Learning

Linear Regression & Back-propagation (2)

Jeju National University

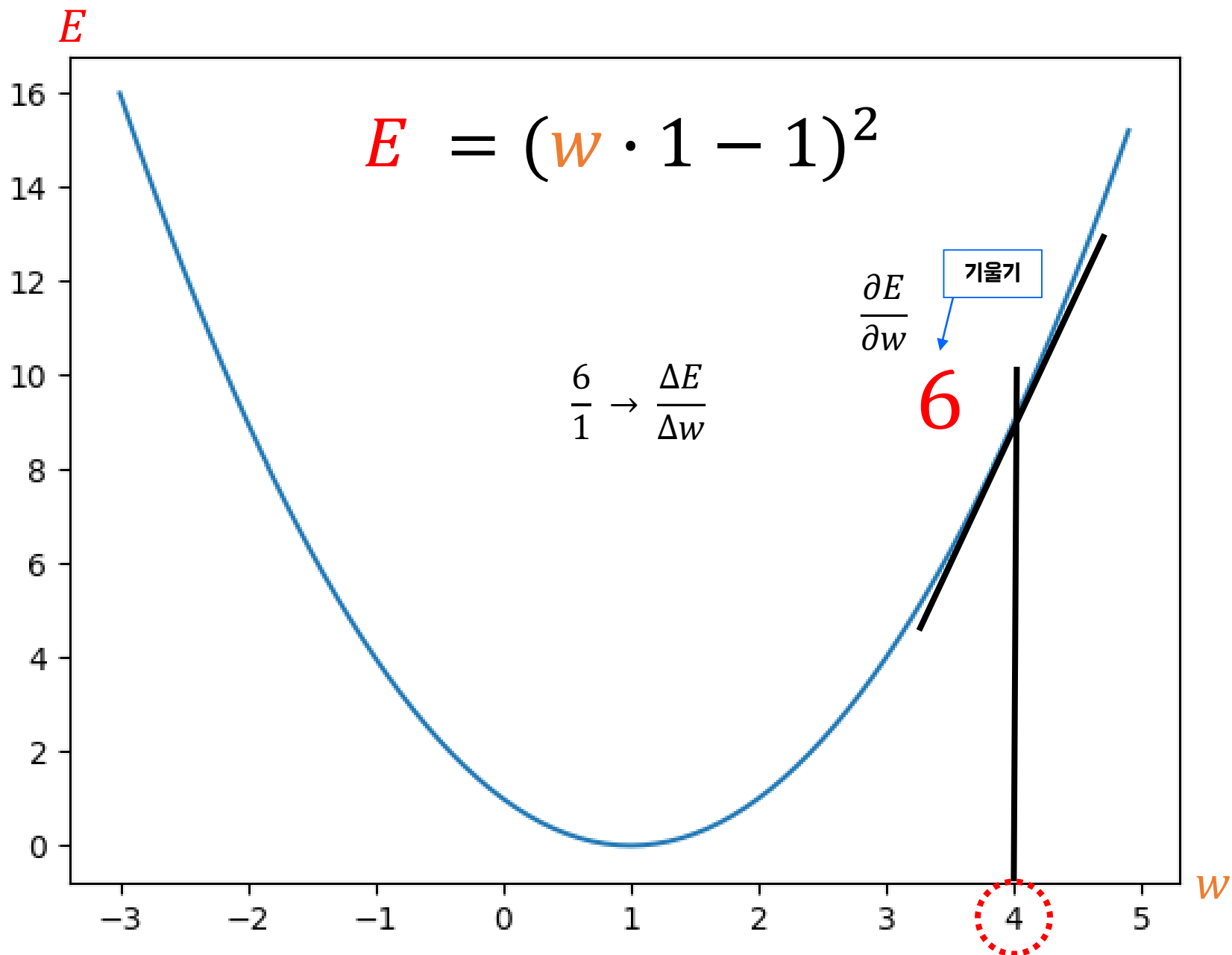
Yung-Cheol Byun

어떻게 자동으로

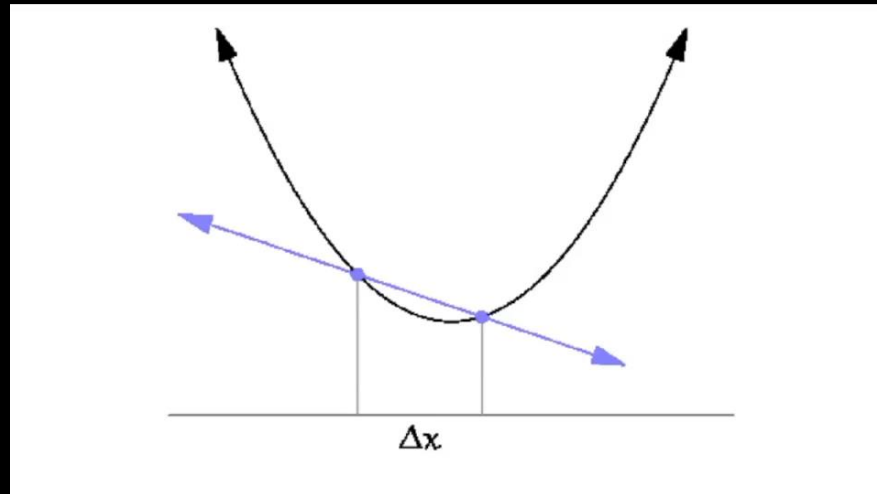
- 오류 E 를 최소화하는 w 값을 찾을까?

미치는 영향, 기울기에 대한 생각

- 기울기가 6
- w 가 1만큼 증가하면 오류 E 는 6만큼 증가함을 의미 (미치는 영향이 큼)
- 오류 최소 지점에서 멀리 떨어져 있음.
- 따라서 오류를 줄이고 싶으면? w 를 감소시켜야 함.
- 많이 감소시켜도 됨.

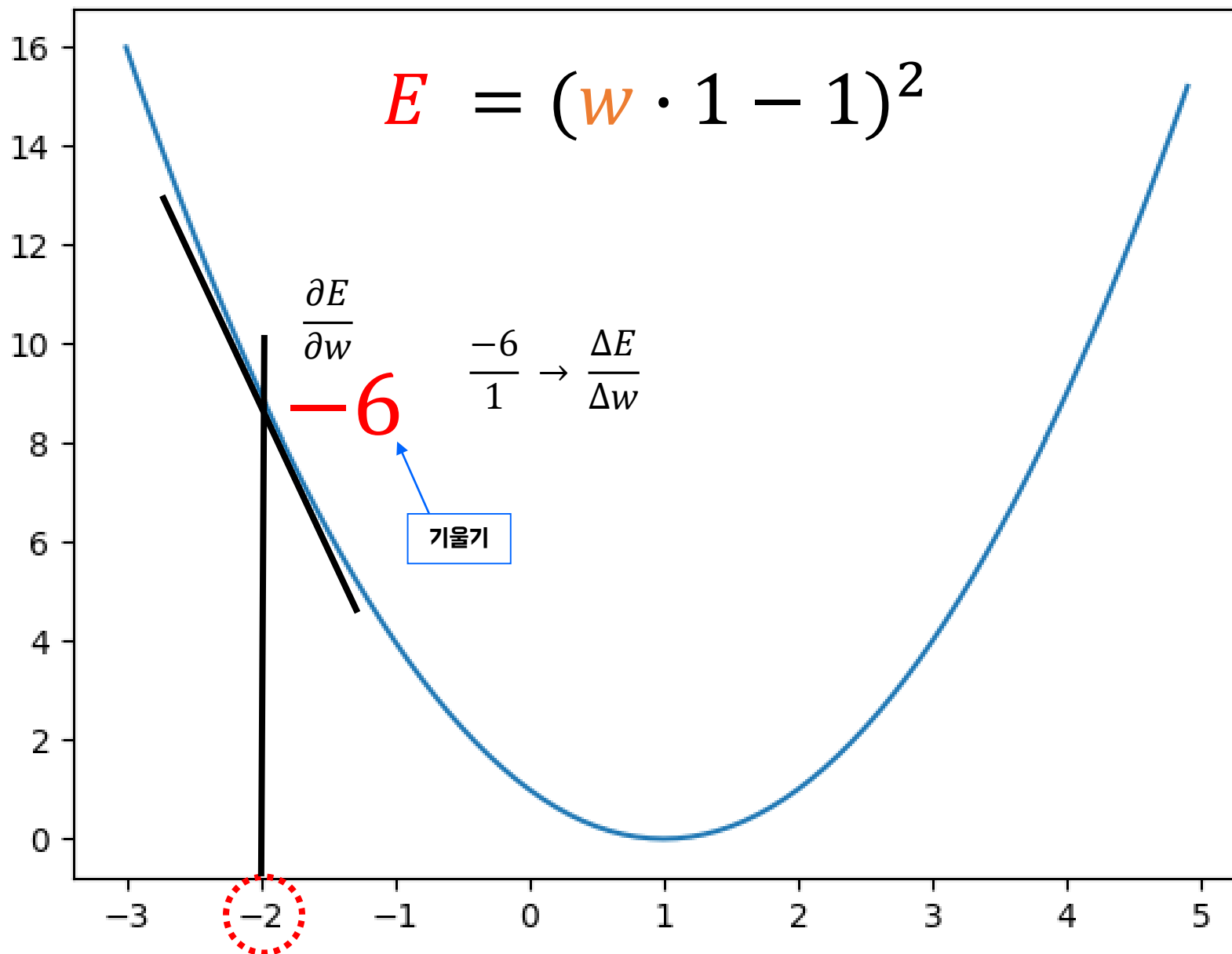


$$\lim_{\Delta x \rightarrow 0} \frac{\Delta E}{\Delta w} \rightarrow \frac{\partial E}{\partial w}$$



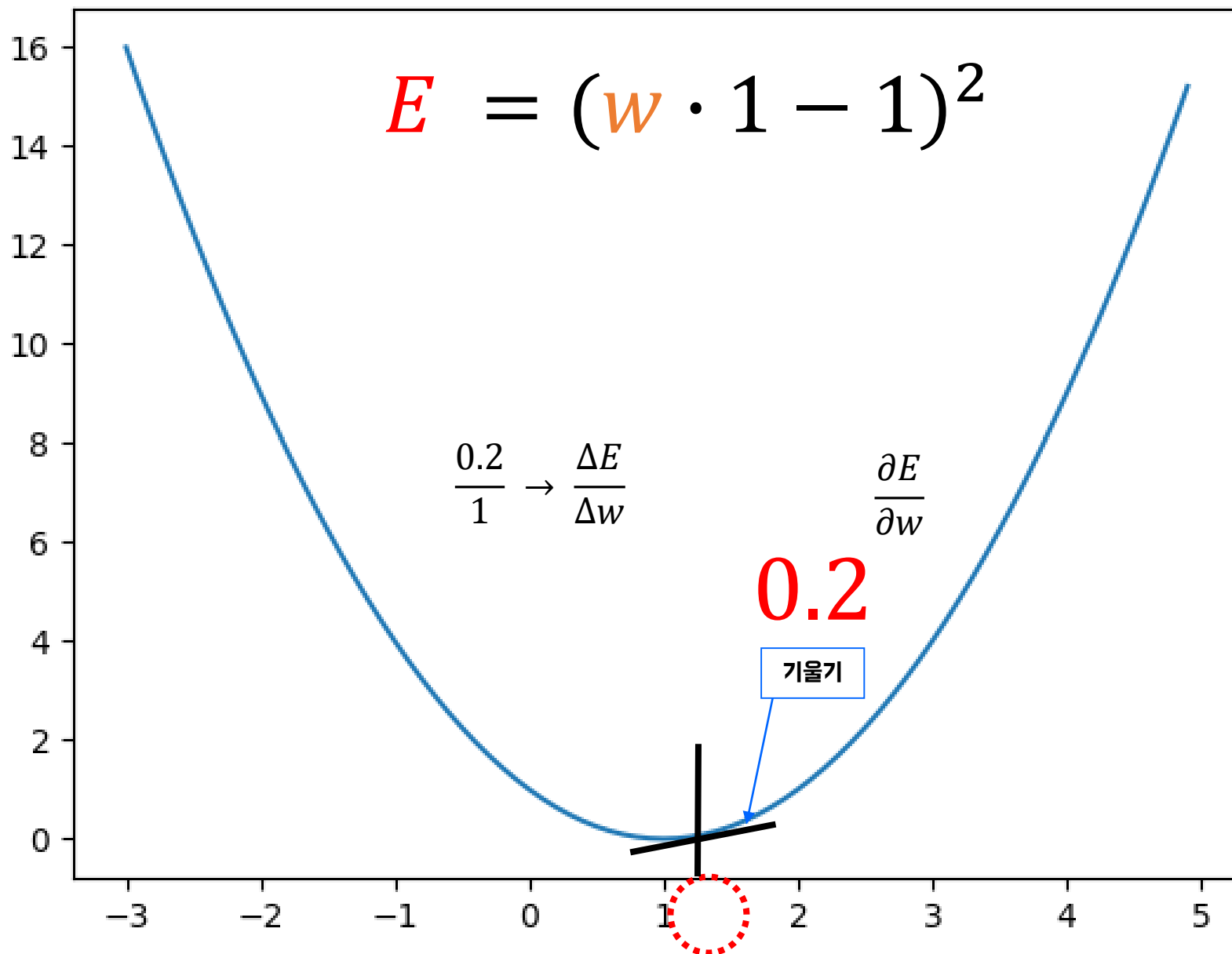
미치는 영향, 기울기에 대한 생각

- 기울기가 -6
- w 가 1만큼 증가하면 오류 E 는 6만큼 감소함을 의미 (미치는 영향이 큼)
- 오류 최소 지점에서 멀리 떨어져 있음.
- 따라서 오류를 줄이고 싶으면? w 를 증가시켜야 함.
- 이때 많이 증가시켜도 됨.



미치는 영향, 기울기에 대한 생각

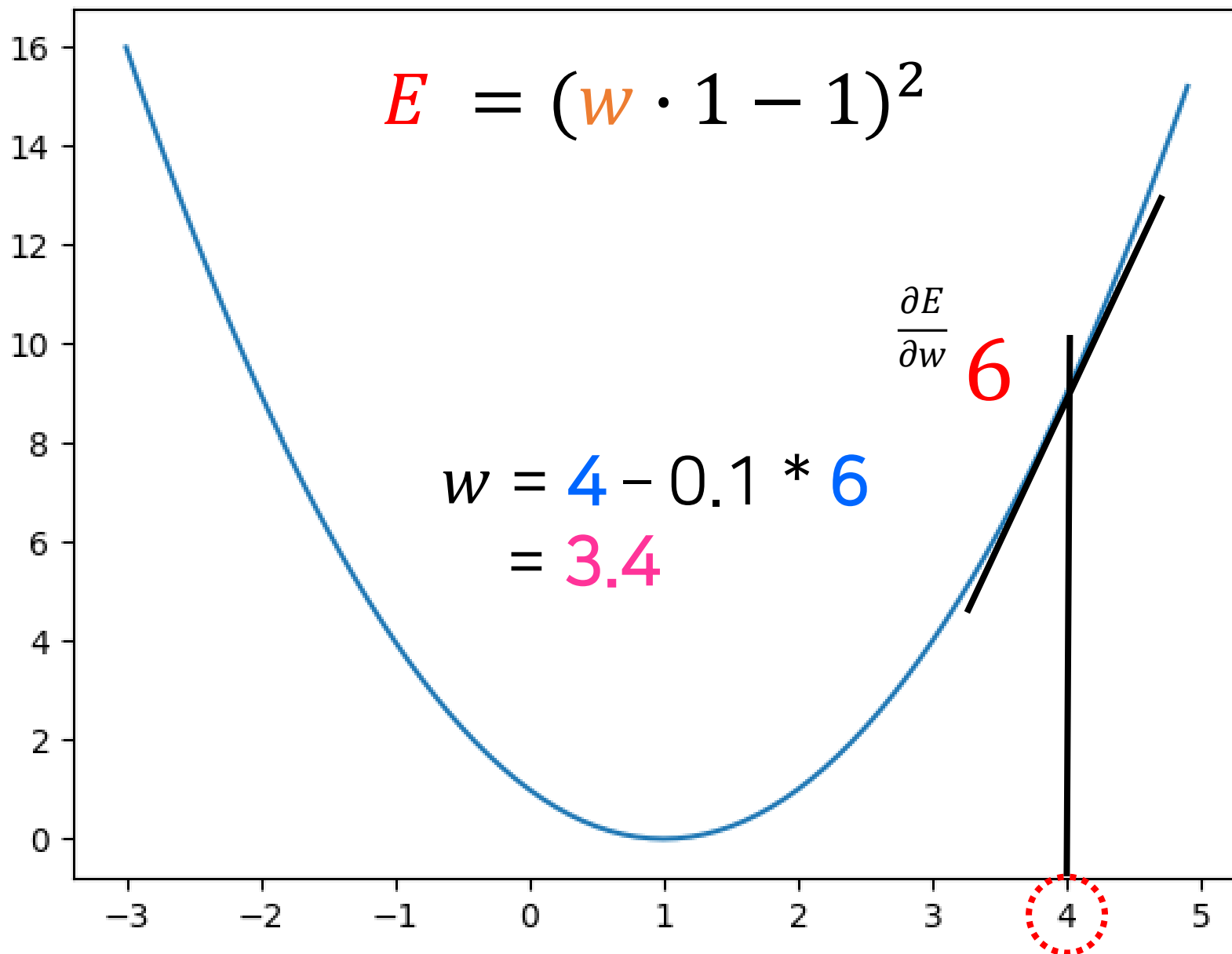
- 기울기가 0.2
- w 가 1만큼 증가해도 오류 E 는 기껏해야 0.2만큼 증가함을 의미 (미치는 영향은 별로 없음)
- 오류 최소지점 가까이 왔음.
- 따라서 오류를 감소시키고자 한다면 w 를 감소시킴.
- 이때 거의 많이 감소시키면 오류가 커질 수 있으므로 조금만 감소시킴.

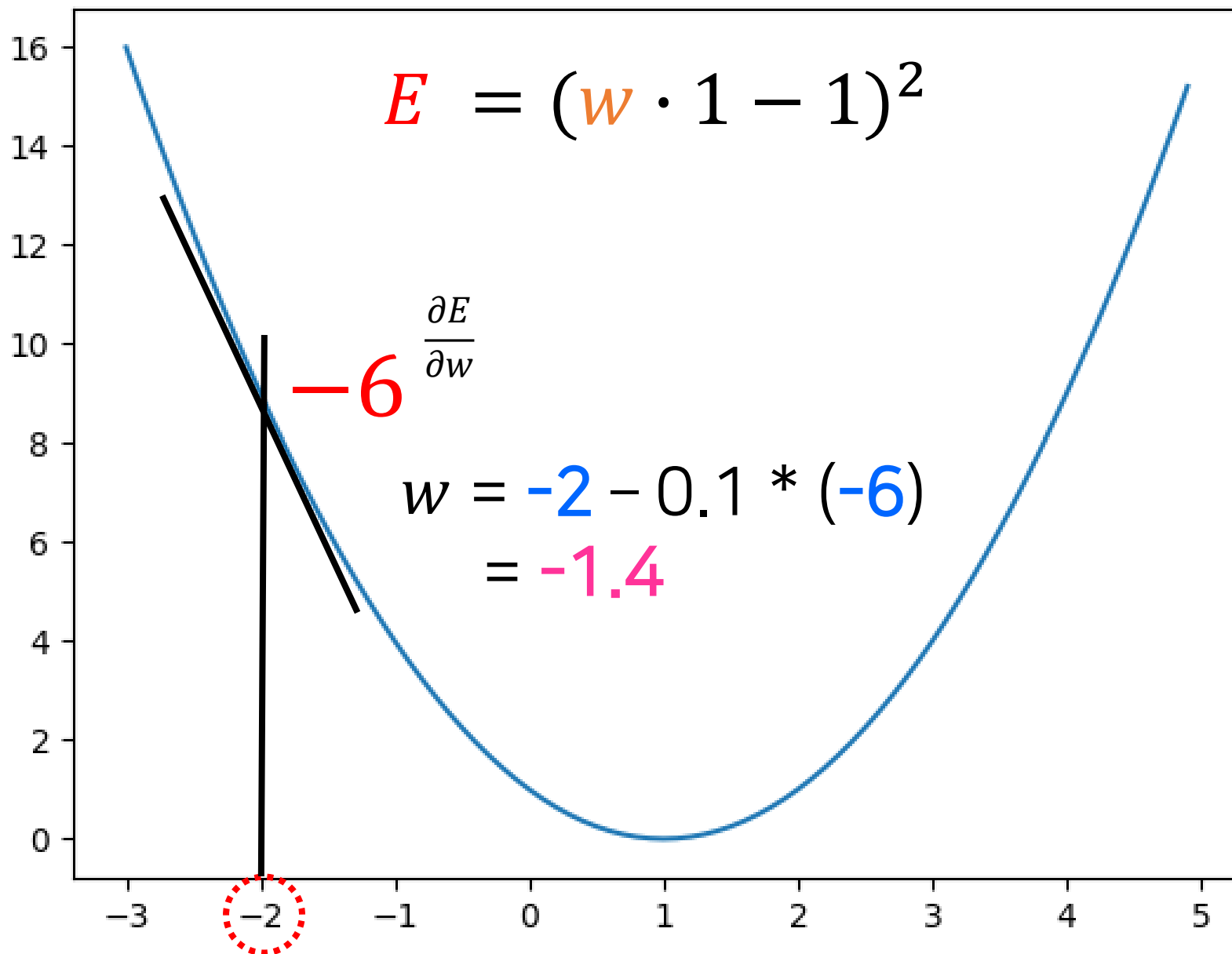


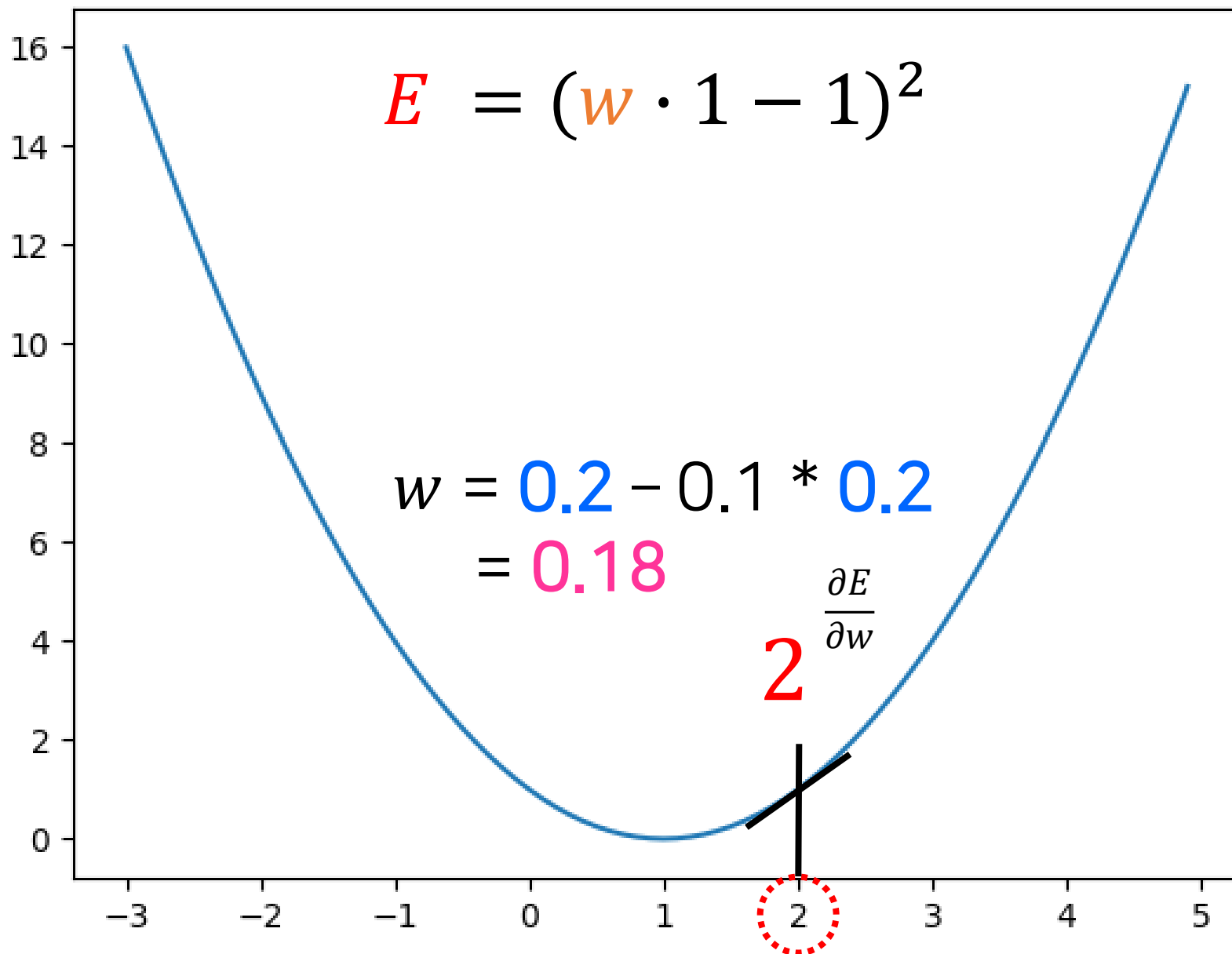
아래와 같이 하면 어떨까?

$$W = W - \alpha * \text{기울기}$$

α = 얼마나 반영할 지를 의미하는 상수(가령, 0.1)







학습 방법(w 업데이트)

- 난수로 w 값 초기화 (ex, -3)
- w 에서 기울기 구함
- 기울기로 w 를 업데이트

$$W = W - \alpha * (\text{기울기})$$

α : 반영 비율 (learning rate)

TensorFlow

- Tensor : '잡아당기다'라는 뜻인 라틴어 'tensus'
- 무언가를 잡아당기면 그 주위에 굉장히 복잡한 변형이 일어나는데, 이를 기술하는 수학적 언어가 Tensor
- Tensor는 계산그래프에서 전달되는 데이터(벡터, 행렬 등)를 포함하며 변수는 Tensor를 저장하는 버퍼
- 따라서 TensorFlow는 텐서(데이터)의 흐름(flow)하며, 우리가 설치하는 Google의 TensorFlow는 머신러닝 프레임워크를 말함.

TensorFlow

- 텐서플로우 프레임워크에서는 가중치 w 파라미터를 자동으로 업데이트하여 찾아 줌 (튜닝).
- 우리가 업데이트하는 것이 아님.
- 이를 위해 w 변수를 텐서플로우 프레임워크 내에서 관리할 수 있도록 정의함.
- 또한 hypothesis와 cost function(E)도 정의해야 함.

TF를 이용한 선형 회귀 학습

③

`w = tf.Variable(tf.random_normal([1]))`

x

x

w

①

`x = [1]`

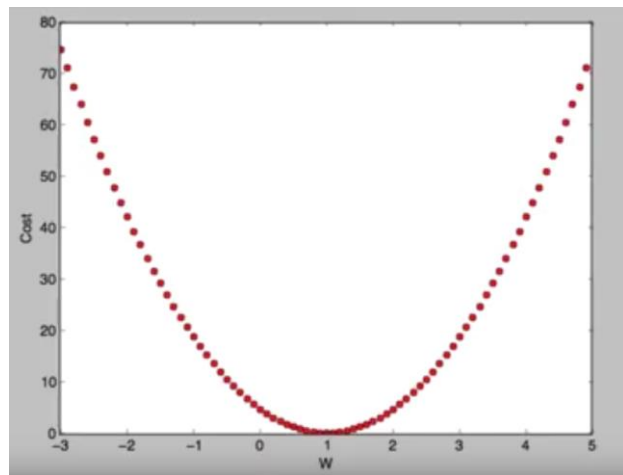
`hypo = w * x`

h

④

`y = [1]`

②



`cost_function = (hypo - y) ** 2`

⑤

$$E = (\text{hypo} - y)^2$$

myml.git 다운로드

- 1) 명령 프롬프트 실행
- 2) 원하는 폴더로 이동
- 3) `git clone https://github.com/yungbyun/myml.git`
- 4) PyCharm으로 오픈 (File | Open...)

01.py

Finding w in
linear regression

```
import tensorflow as tf
```

```
#----- training data
```

```
x_data = [1]
```

```
y_data = [1]
```

```
#----- a neuron
```

```
w = tf.Variable(tf.random_normal([1]))
```

```
hypo = w * x_data
```

```
#----- learning
```

```
cost = (hypo - y_data) ** 2
```

```
train = tf.train.GradientDescentOptimizer(learning_rate=0.01).minimize(cost)
```

```
sess = tf.Session()
```

```
sess.run(tf.global_variables_initializer())
```

```
for i in range(1001):
```

```
    sess.run(train) #w가 한번 업데이트 됨. 1001번 루프이므로 1001번 업데이트 됨.
```

```
    if i % 100 == 0:
```


```
        print('w:', sess.run(w), 'cost:', sess.run(cost))
```

```
#----- testing(prediction)
```

```
x_data = [2]
```

```
print(sess.run(x_data * w))
```

cost(error)를
최소화하도록
w를 업데이트하는
오퍼레이션 train 선언



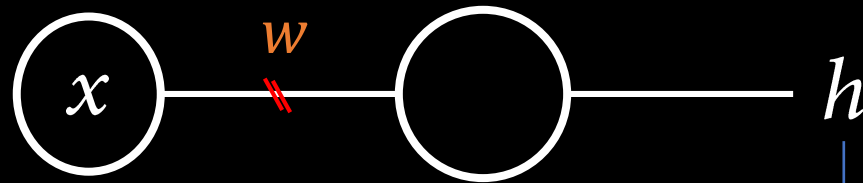
02.py

Drawing
cost function

오류 함수 생각하기

$$E = (wx - y)^2$$

- 어느 부분이 뉴런인가?
- 뉴런의 모습 상상하기
- 입력 데이터는?
- 출력 데이터는?
- 시냅스는?
- 가설(hypothesis)은?
- 뉴런의 출력
- 오류 함수의 의미는?
- 뉴런 입력이 여러 개일 경우



$$E = (h - y)^2$$

$$E = (w \cdot x - y)^2$$

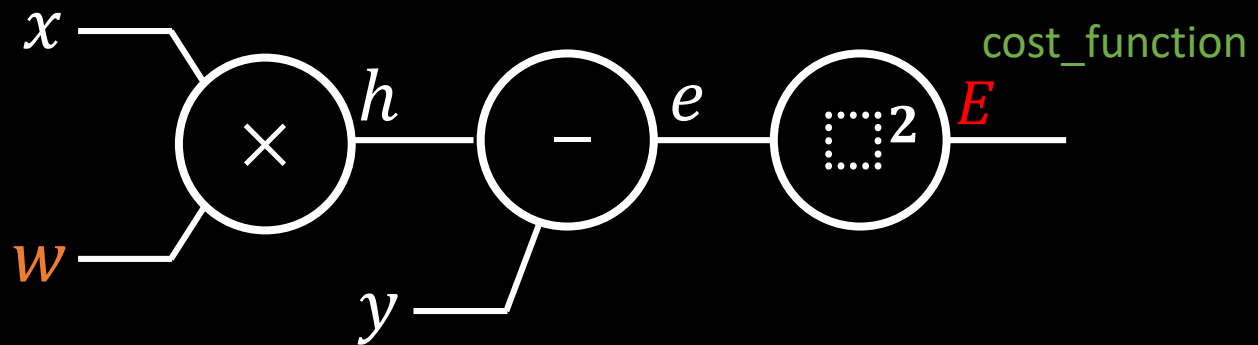
$$E = (w \cdot 1 - 1)^2$$

오류 계산 그래프

$$E = (wx - y)^2$$

hypo = w * x

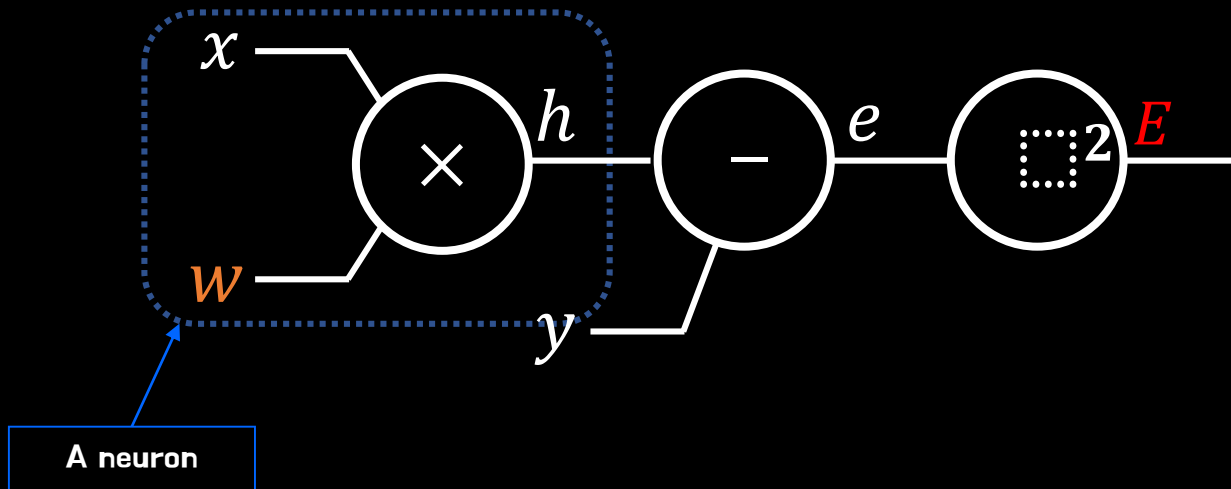
cost_function(E) = (hypo - y) ** 2



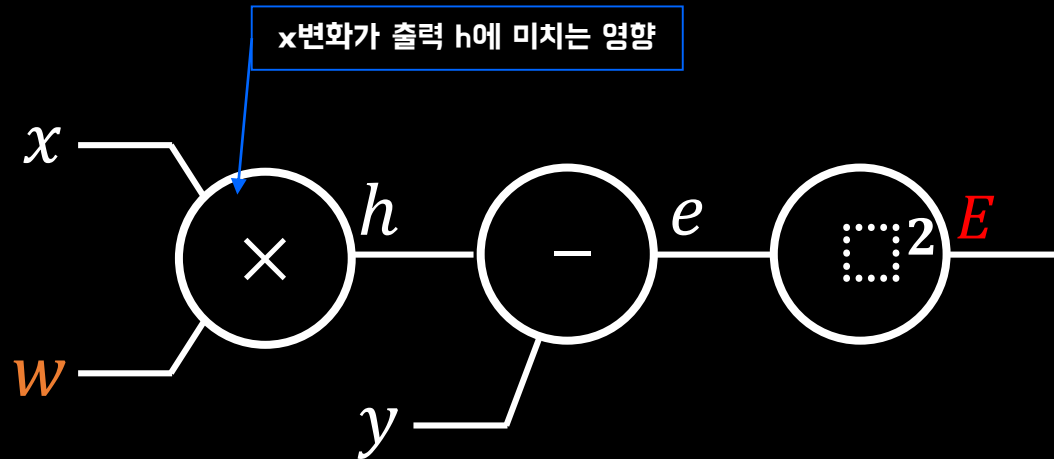
텐서란 무엇이고, 텐서 플로우란 무엇인가?
텐서플로우 프레임워크가 파라미터 튜닝

오류 계산 그래프와 미치는 영향

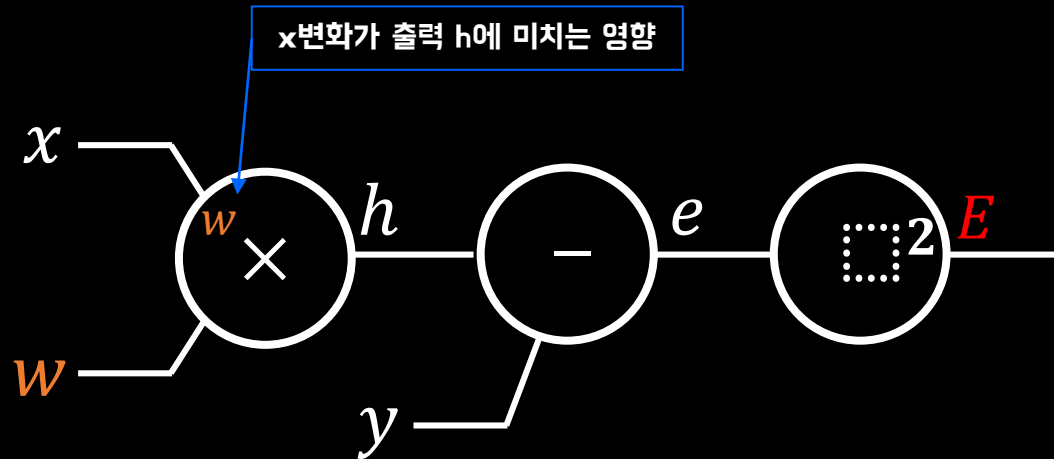
계산 그래프에서 w 가 E에 미치는 영향을 쉽게 알 수 있다.
그러면 오류를 줄일 수 있도록 w 를 조절할 수 있다.



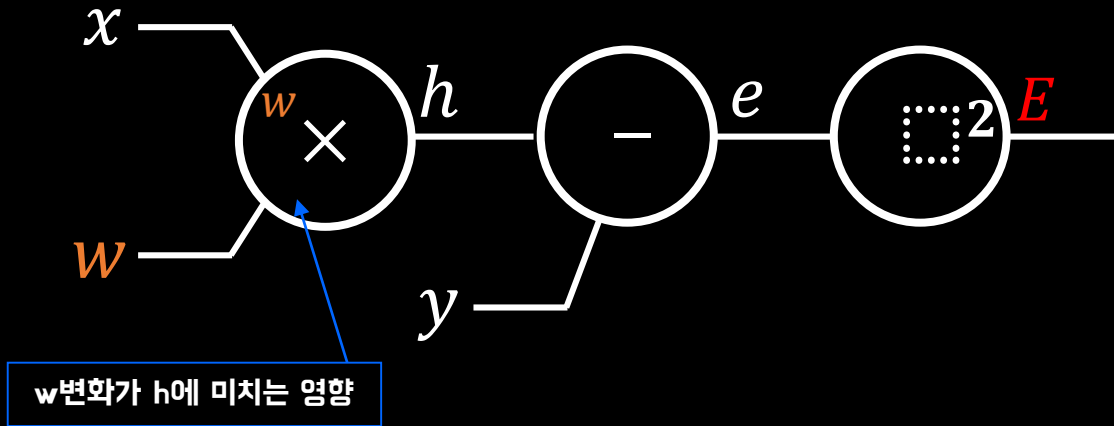
오류 계산 그래프와 미치는 영향



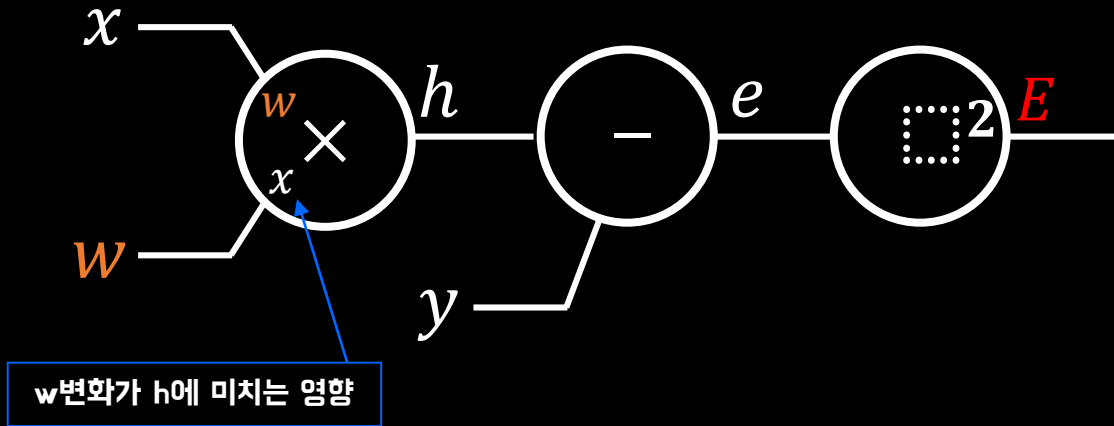
오류 계산 그래프와 미치는 영향



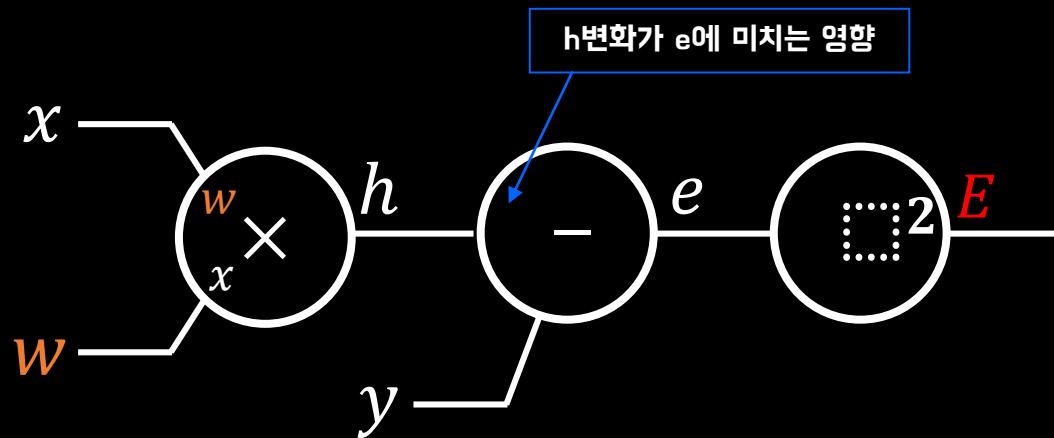
오류 계산 그래프와 미치는 영향



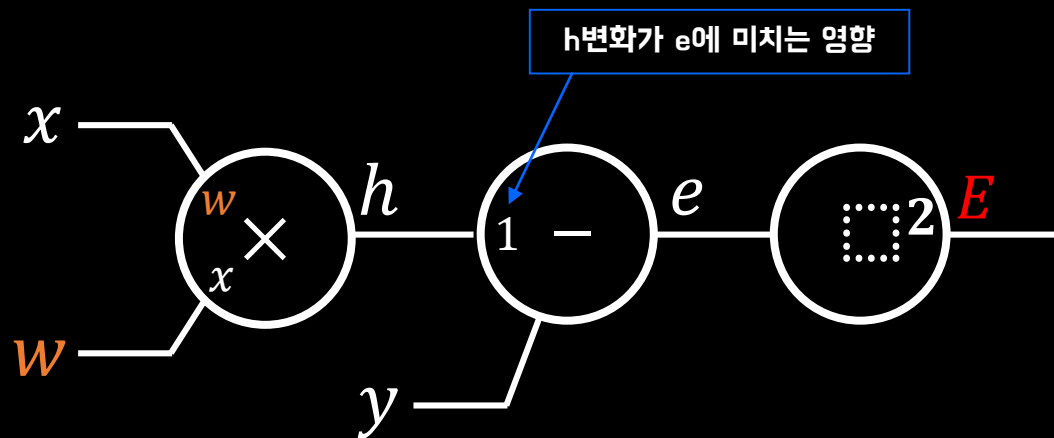
오류 계산 그래프와 미치는 영향



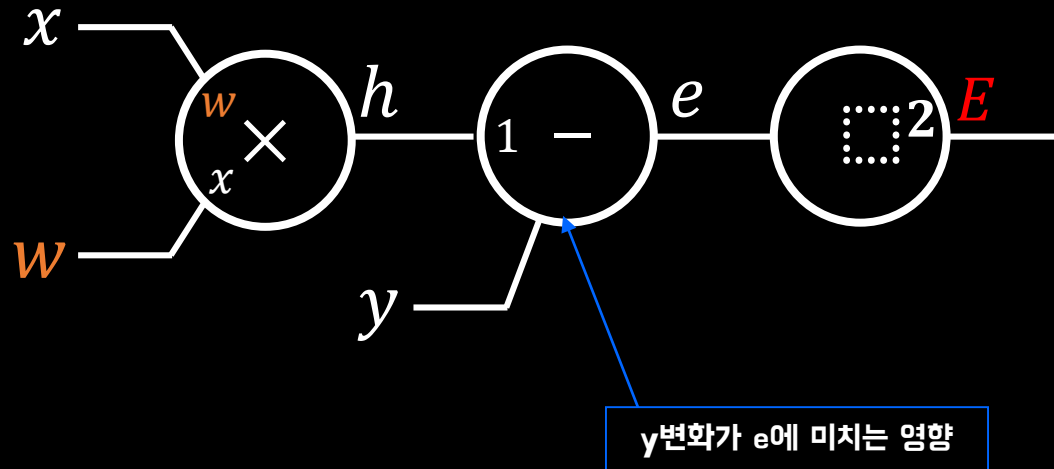
오류 계산 그래프와 미치는 영향



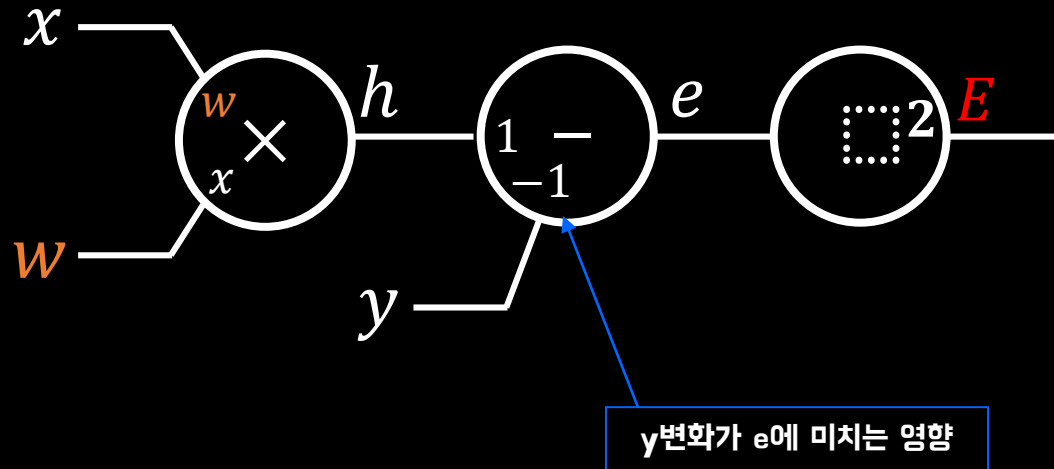
오류 계산 그래프와 미치는 영향



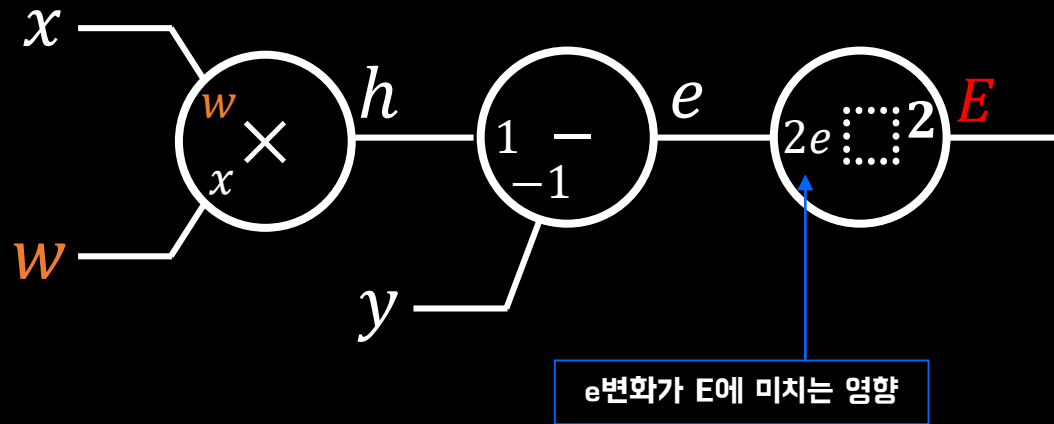
오류 계산 그래프와 미치는 영향



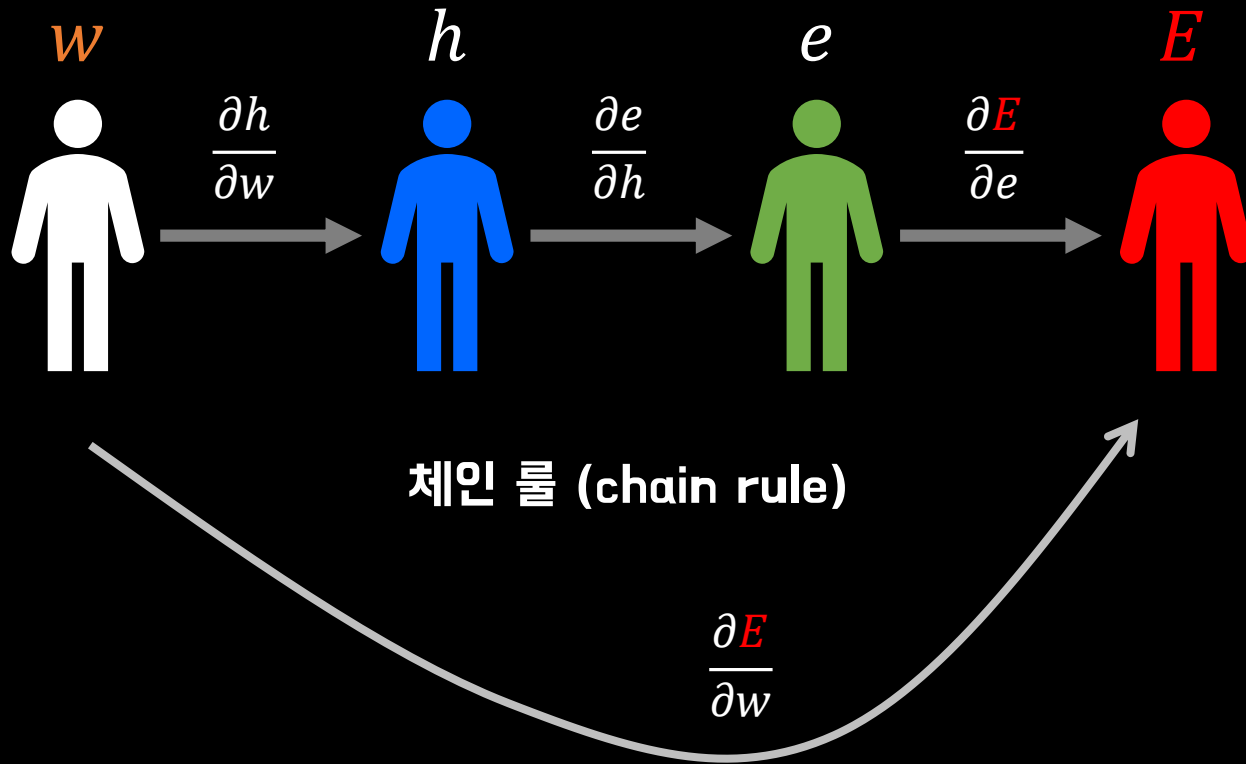
오류 계산 그래프와 미치는 영향



오류 계산 그래프와 미치는 영향



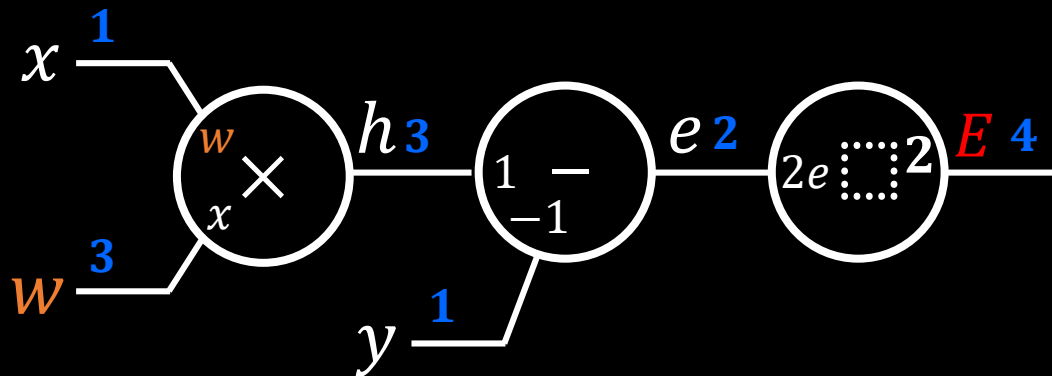
사람 사이의 영향력



w 변화가
 E 에 미치는 영향력은?

앞으로 전파

$(x, y) = (1, 1)$ 이고 w 는 3일 경우 **에러** 값은?



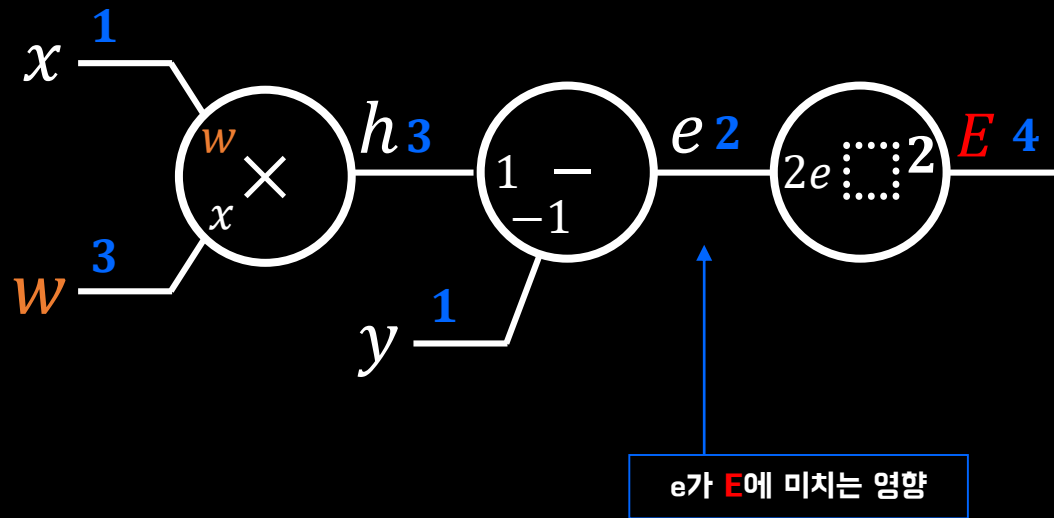
에러(E)가 크다.

에러(E)가 줄어들도록 W 를 조절하자. 어떻게???

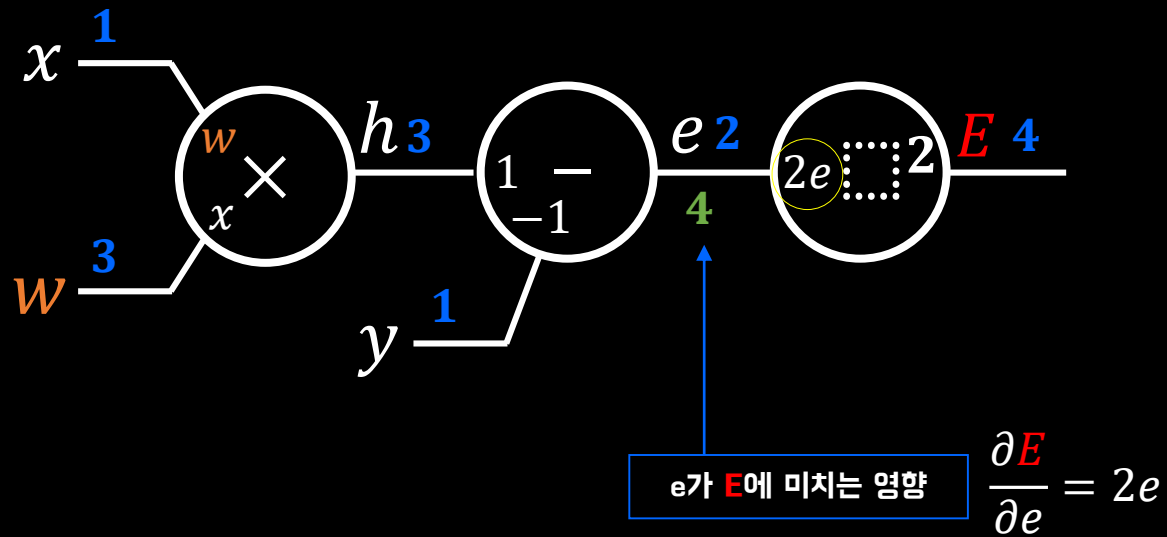
W 변화가 **E**에 미치는 영향(기울기)을 구한 후 $W = W - \alpha * (\text{기울기})$

미치는 영향,
기울기를 구하자.

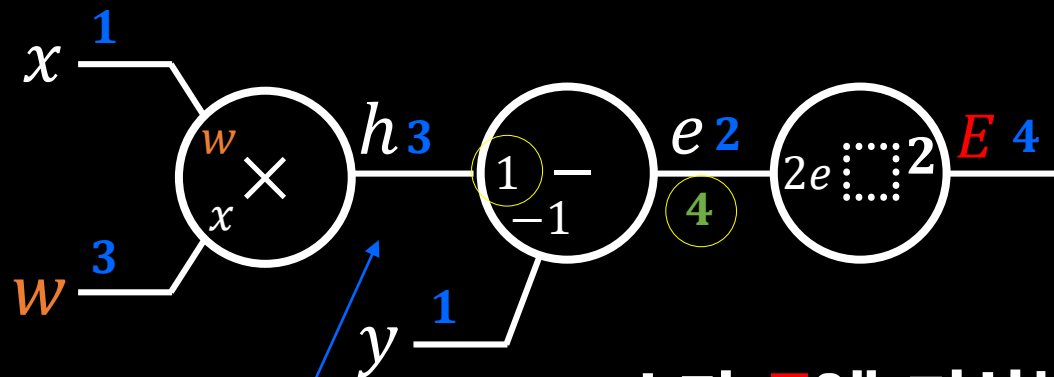
뒤로(역) 전파



뒤로(역) 전파



뒤로(역) 전파

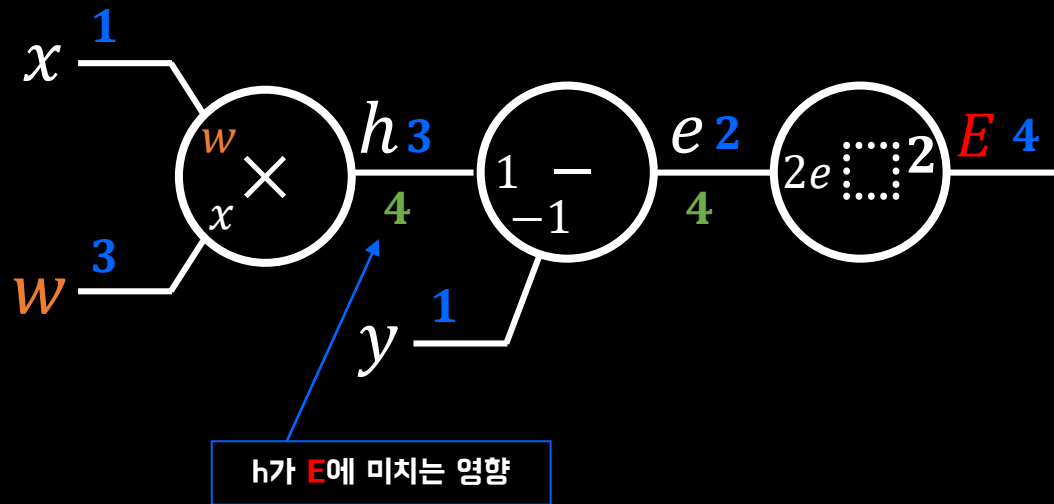


h가 E에 미치는 영향

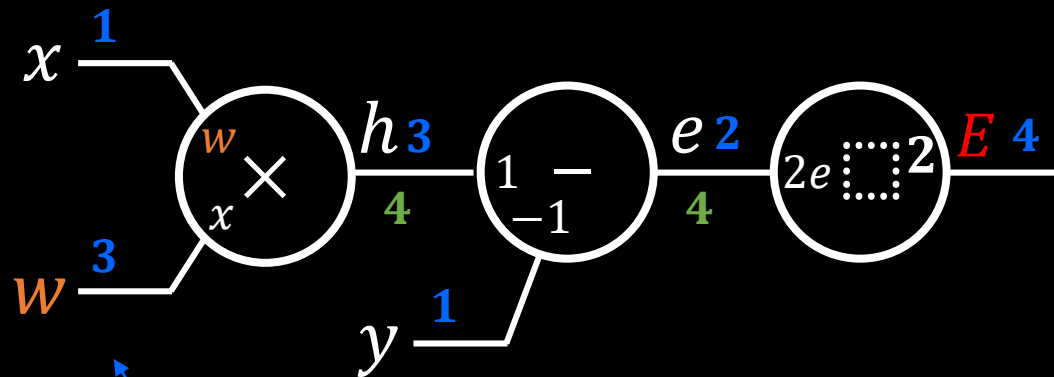
h가 E에 미치는 영향

$$\frac{\partial E}{\partial h} = \frac{\partial e}{\partial h} \cdot \frac{\partial E}{\partial e} = 1 \cdot 4$$

뒤로(역) 전파

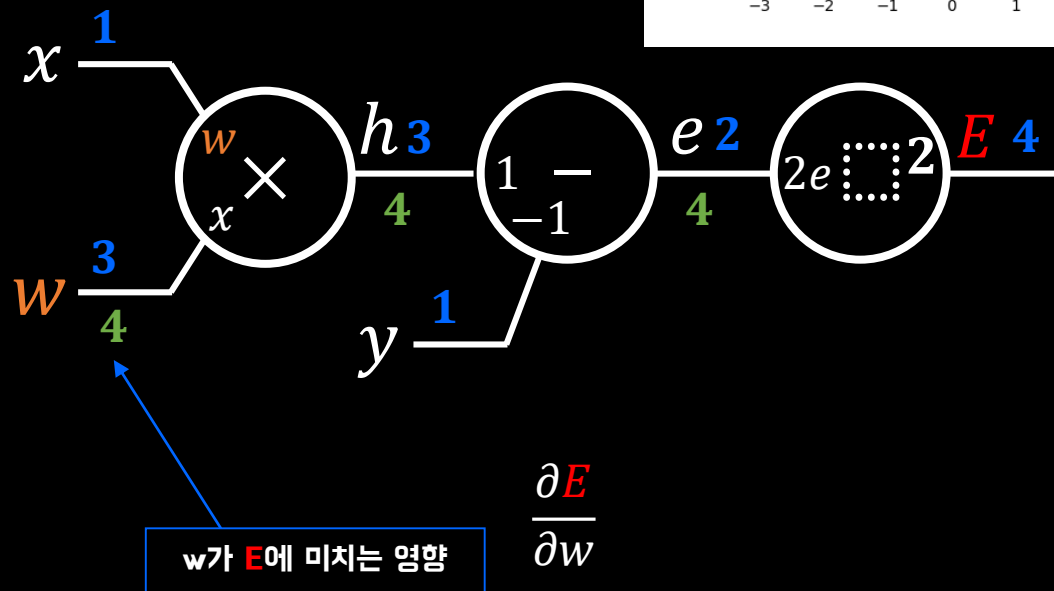
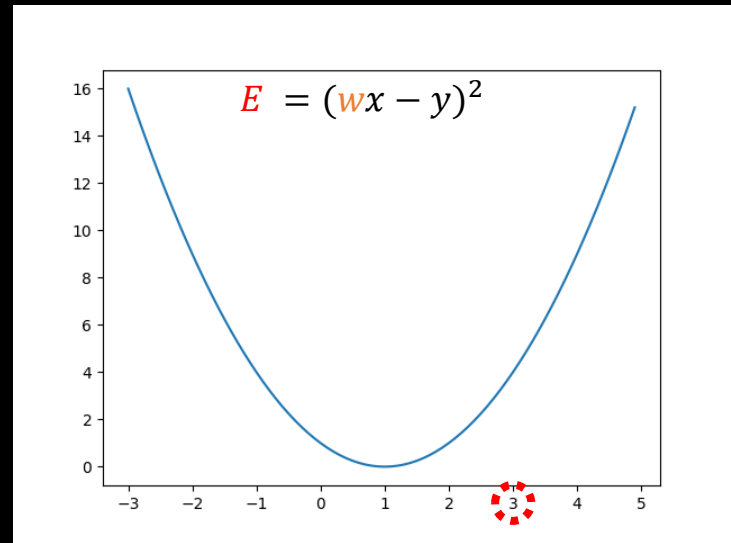


뒤로(역) 전파



w가 E에 미치는 영향

$$\frac{\partial E}{\partial w} = \frac{\partial h}{\partial h} \cdot \frac{\partial e}{\partial h} \cdot \frac{\partial E}{\partial e} = x \cdot 1 \cdot 4$$



따라서 역 전파 (back-propagation)

체인 룰(chain rule)을 적용하여 w 가
오류 E 에 얼마나 영향을 미치는지(기
울기)를 알아내는 과정

$$\frac{\partial E}{\partial w}$$

$$W = 3 - 0.1 * 4^{\frac{\partial E}{\partial w}}$$
$$W = 2.6$$



Tuned parameter
after 1 step learning.

미치는 영향을 구하는 방법

현재 w 값에서 w 변화가 오류 E 에 미치는 영향 구하기

[방법1] w 가 아주 조금 변할 때 E 는 얼마나 변하는지 계산 (계산기 이용)

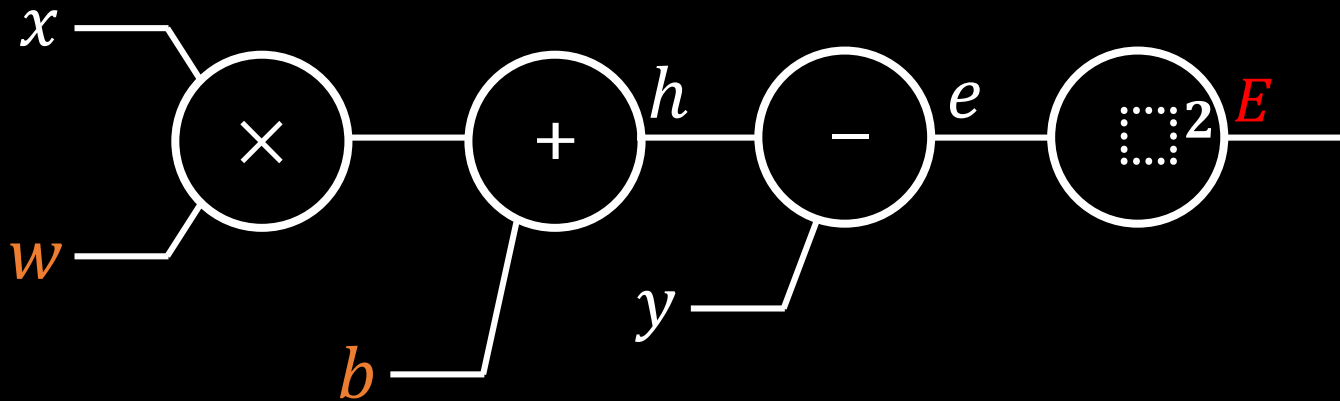
[방법2] 계산 그래프에서 역전파와 체인 룰을 이용한 방법 (텐서플로우)

[방법3] 고등학교 때 배운 방법(?)

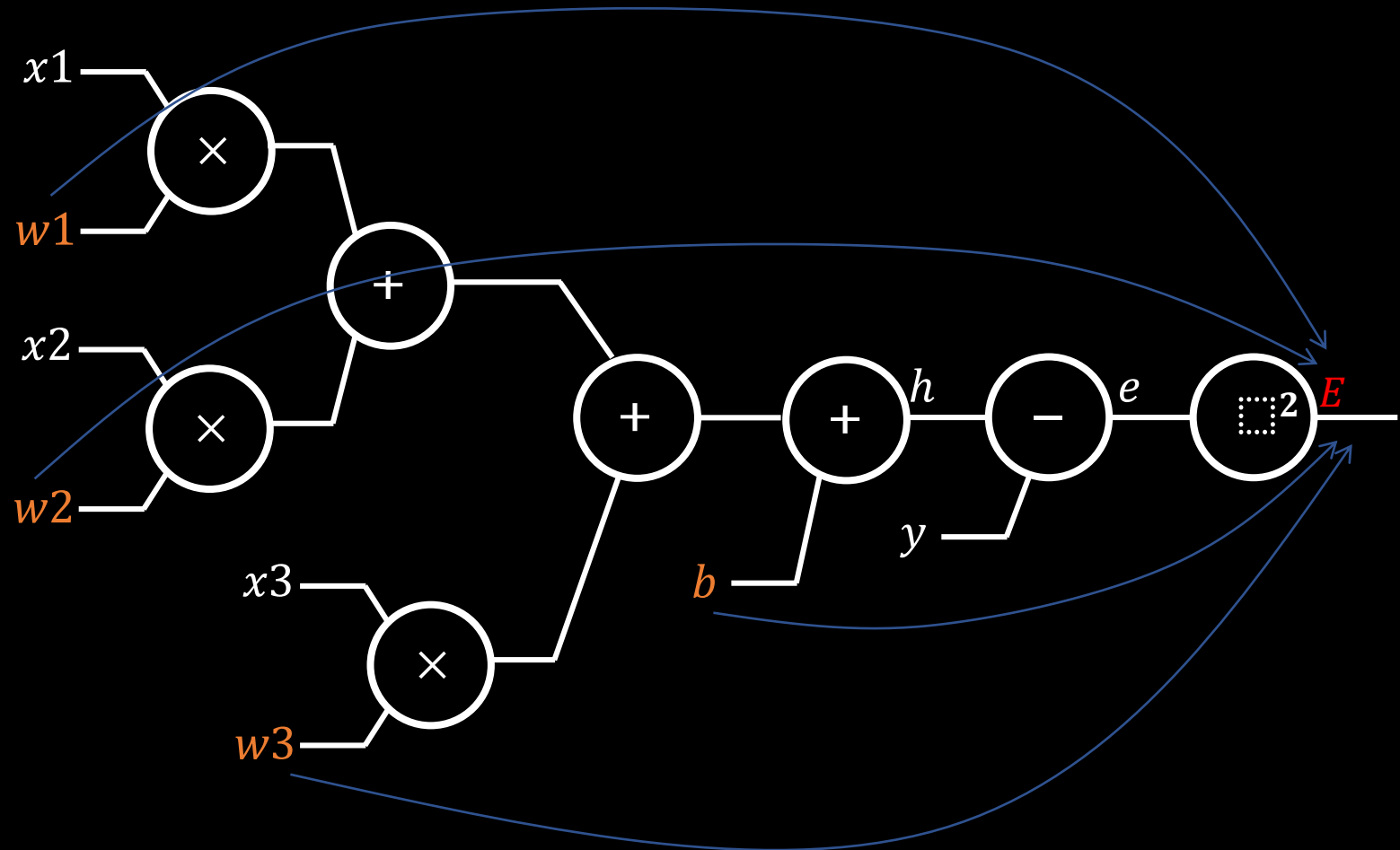
계산 그래프 확장

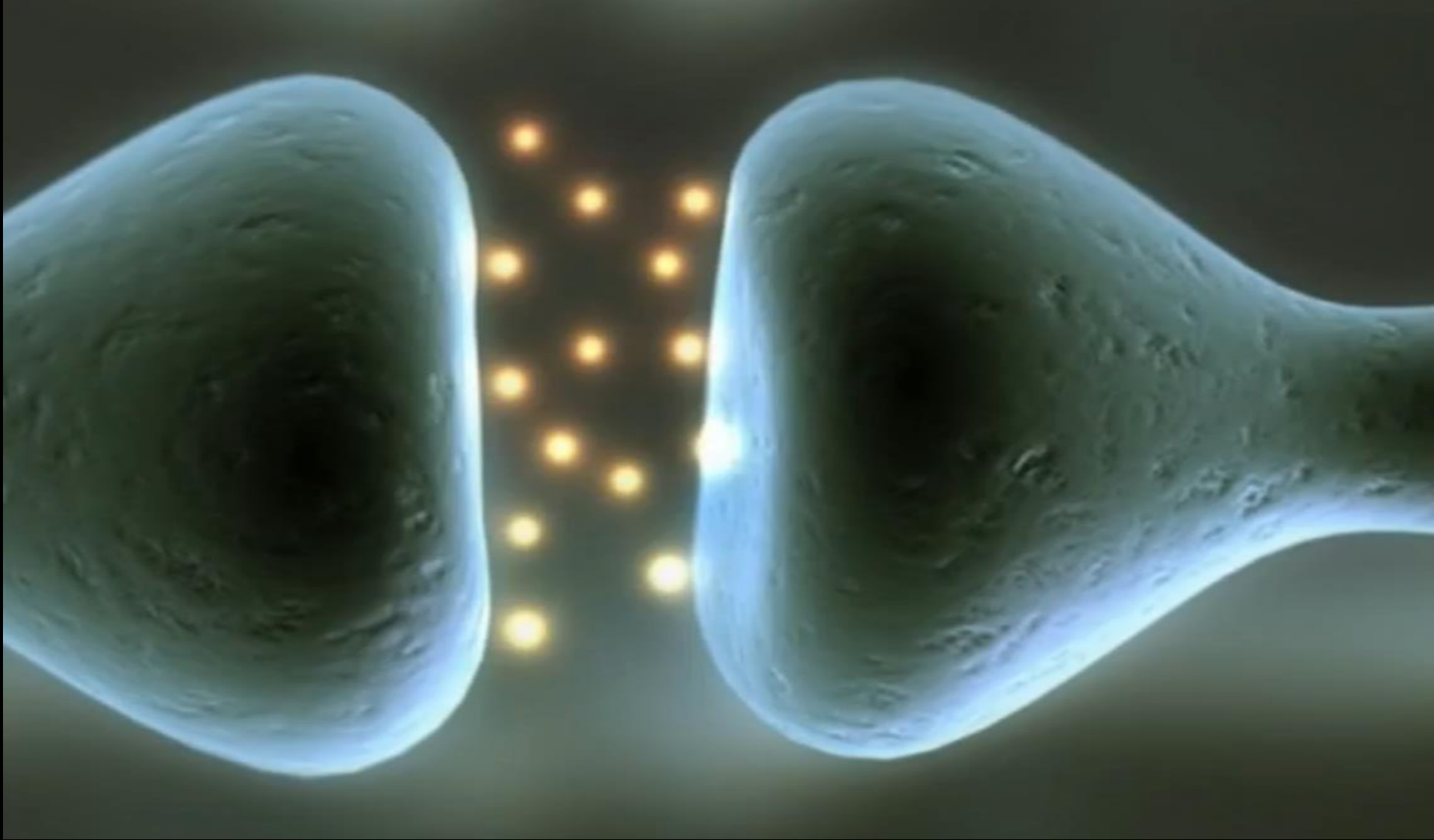
- bias가 있을 경우 (+ 게이트)
- 뉴런 입력이 3개일 때 (+ 게이트)
- 뉴런이 2개일 때
- 튜닝할 파라미터는 모두 몇 개?

$$E = ((wx + b) - y)^2$$



$$E = ((w1x1 + w2x2 + w3x3 + b) - y)^2$$





학습, 더 새롭고 좋은 연결을 만드는 것

Meaning of cost(error)

- 기울기가 큼 → **bad!** → big penalty(아주 힘들다) → big update(w)
- 기울기가 작음 → **not bad!** → small penalty(많이 힘들지 않다) → small update(w)
- 기울기 0 → **great!** → no penalty → no update(w) → learning ended!

우리 마음 속의 cost(error, loss, stress) function

‘좋다’, ‘나쁘다’를 느끼게 하는 기저

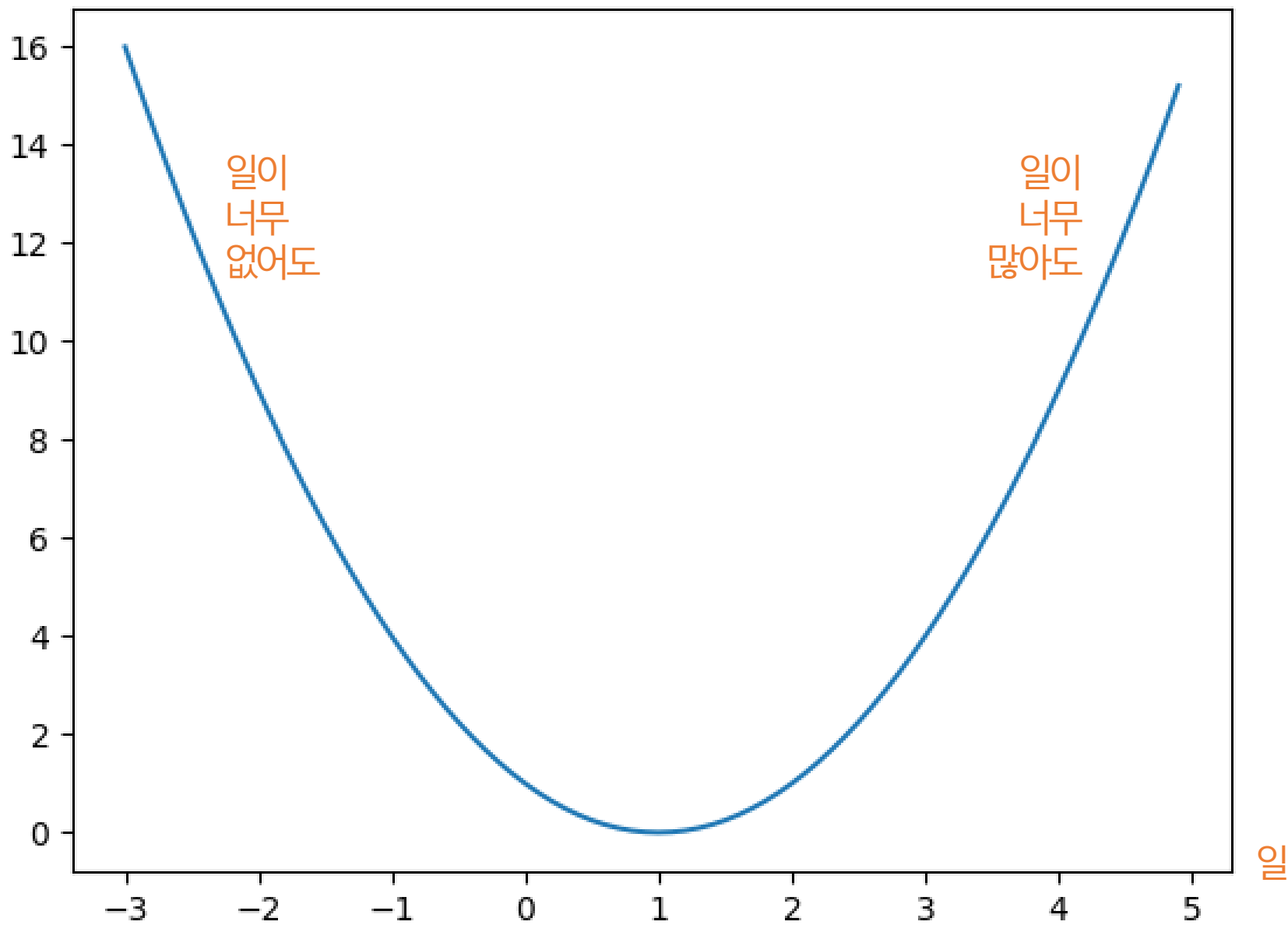
- **좋다**

- 열심히 공부해서 알게 되니 기분이 좋다.
- 낚시가 너무 재미있다.
- 물건 훔치니 기분이 짜릿하다.

- **나쁘다**

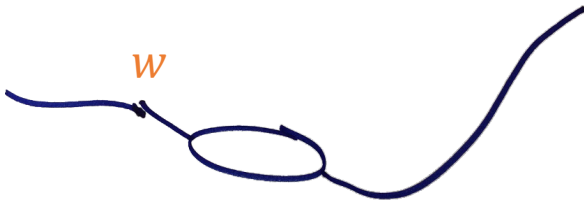
- 많이 틀리니 기분이 나쁘다.
- 과식하니 속이 쓰리다.
- 지령이를 밟았다. 기분이 별로다.

스트레스

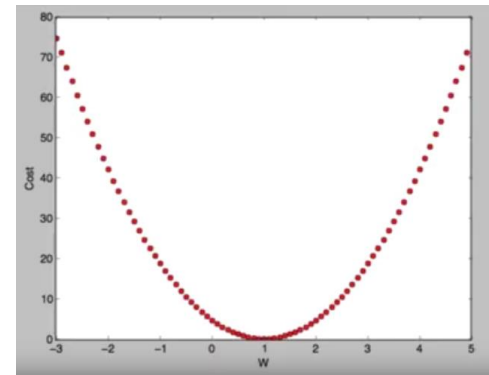


Cost(error) graph

$$E = (w \cdot 1 - 1)^2$$



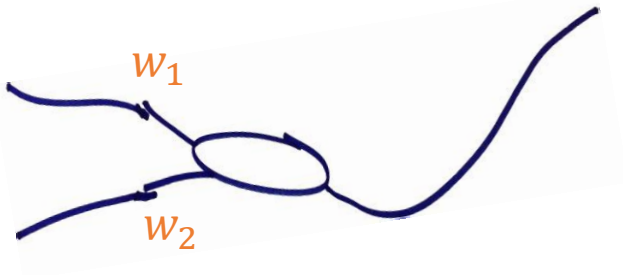
E



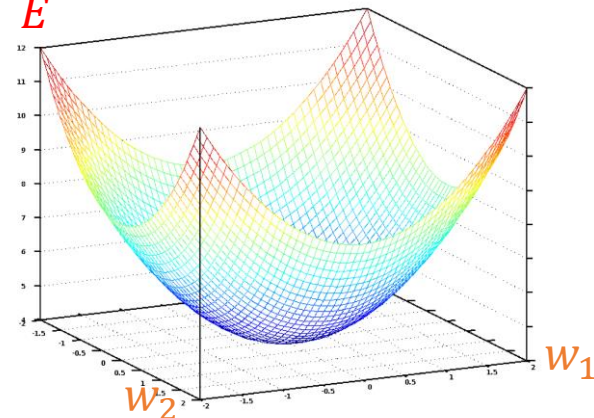
w

convex function

$$E = (W \cdot 1 - 1)^2$$



E



convex function