경사하강법과 오차역전파

21.07.27

장혜선

목차

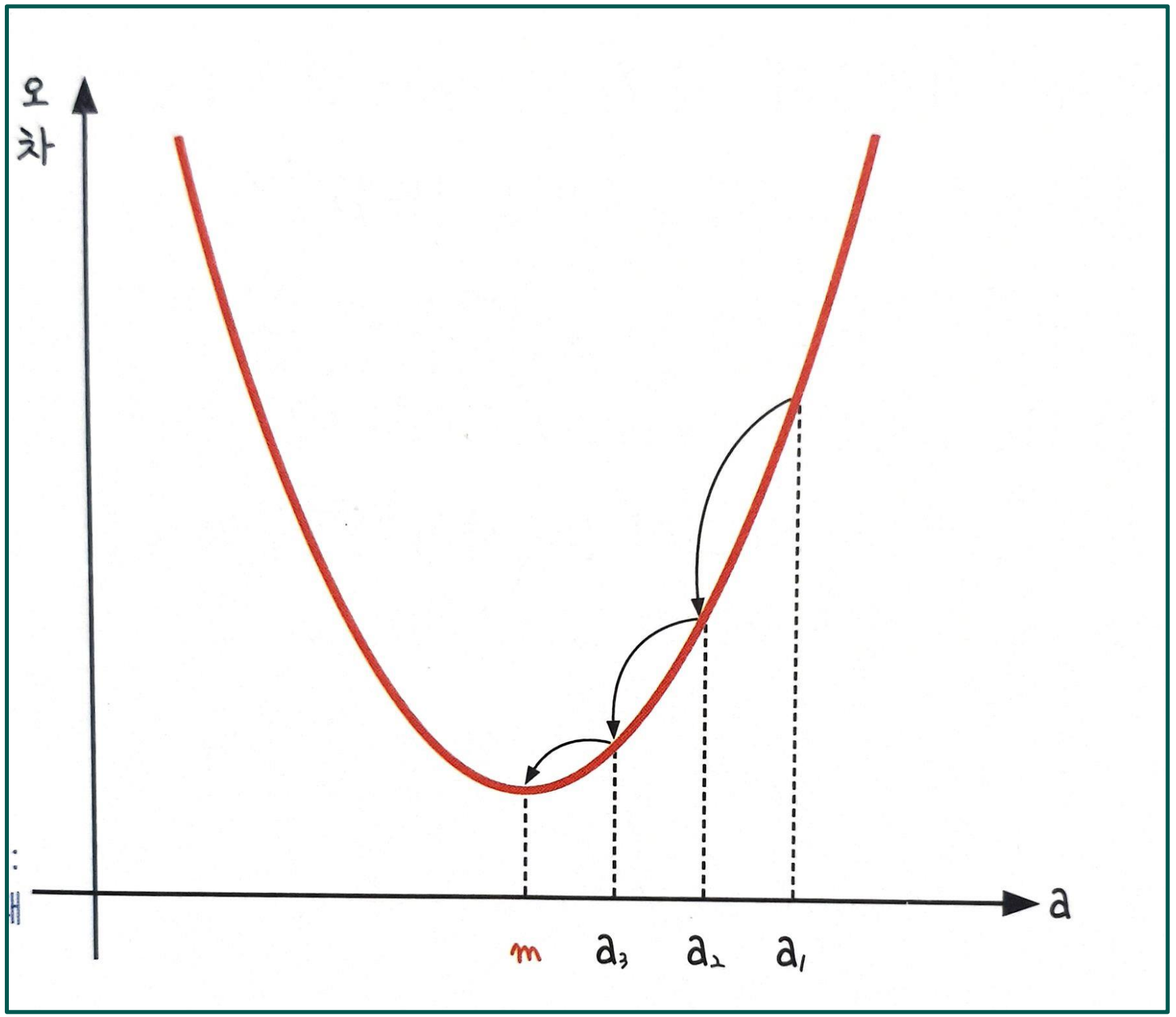
1-1.

경사하강법

오차의 변화에 따라 이차 함수 그래프를 만들고 적절한 학습률을 설정해 미분 값이 0인 지점을 구하는 것

오차가 가장 작을 때 => m 위치에 있을 때

위치 이동 : $a_1 \rightarrow a_2 \rightarrow a_3$



1-1.

미분

순간변화율 : 변수 값이 매우 작게 변하는 동안의 변화 정도

$$\frac{d}{dx}f(x) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

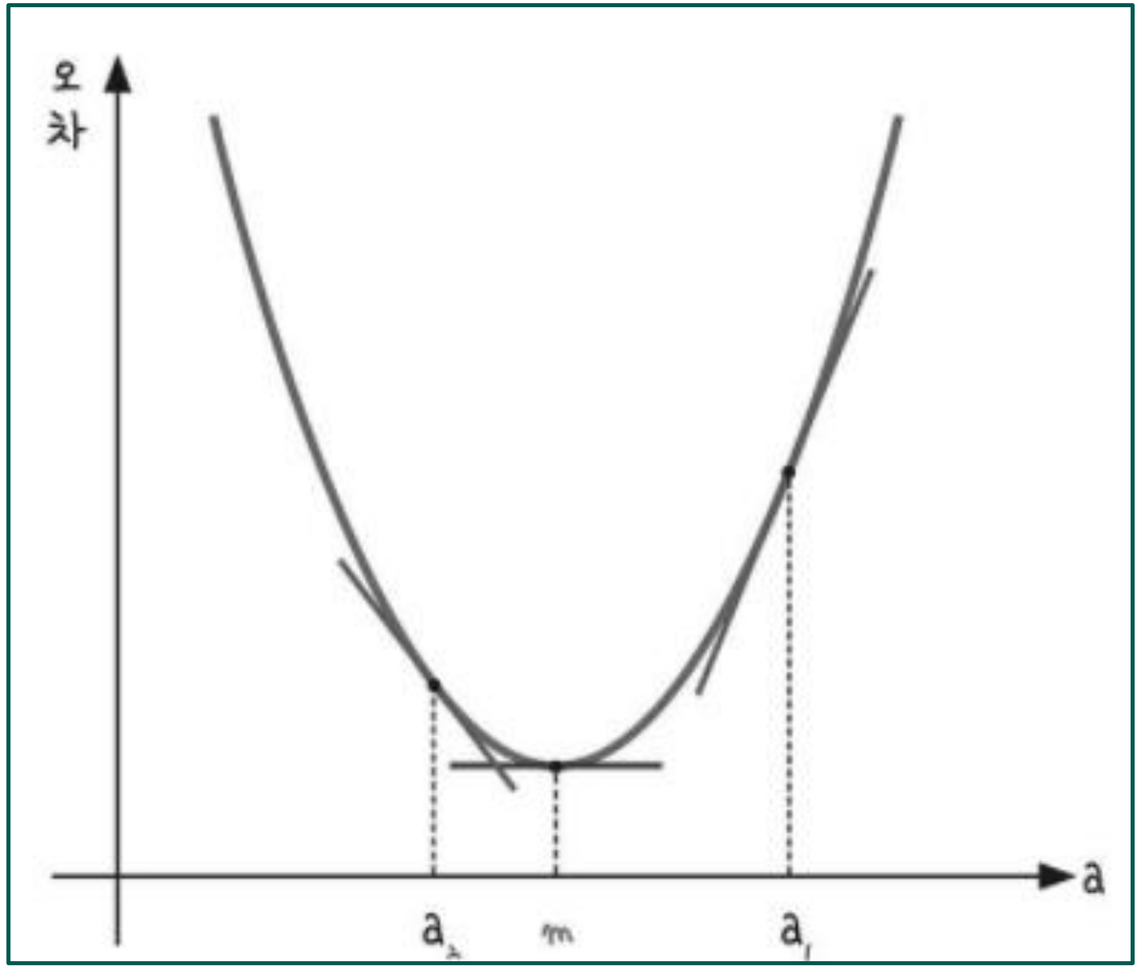
ex)

$f(x) = x^a$

$f'(x) = ax^{a-1}$

$f(x) = x^2$

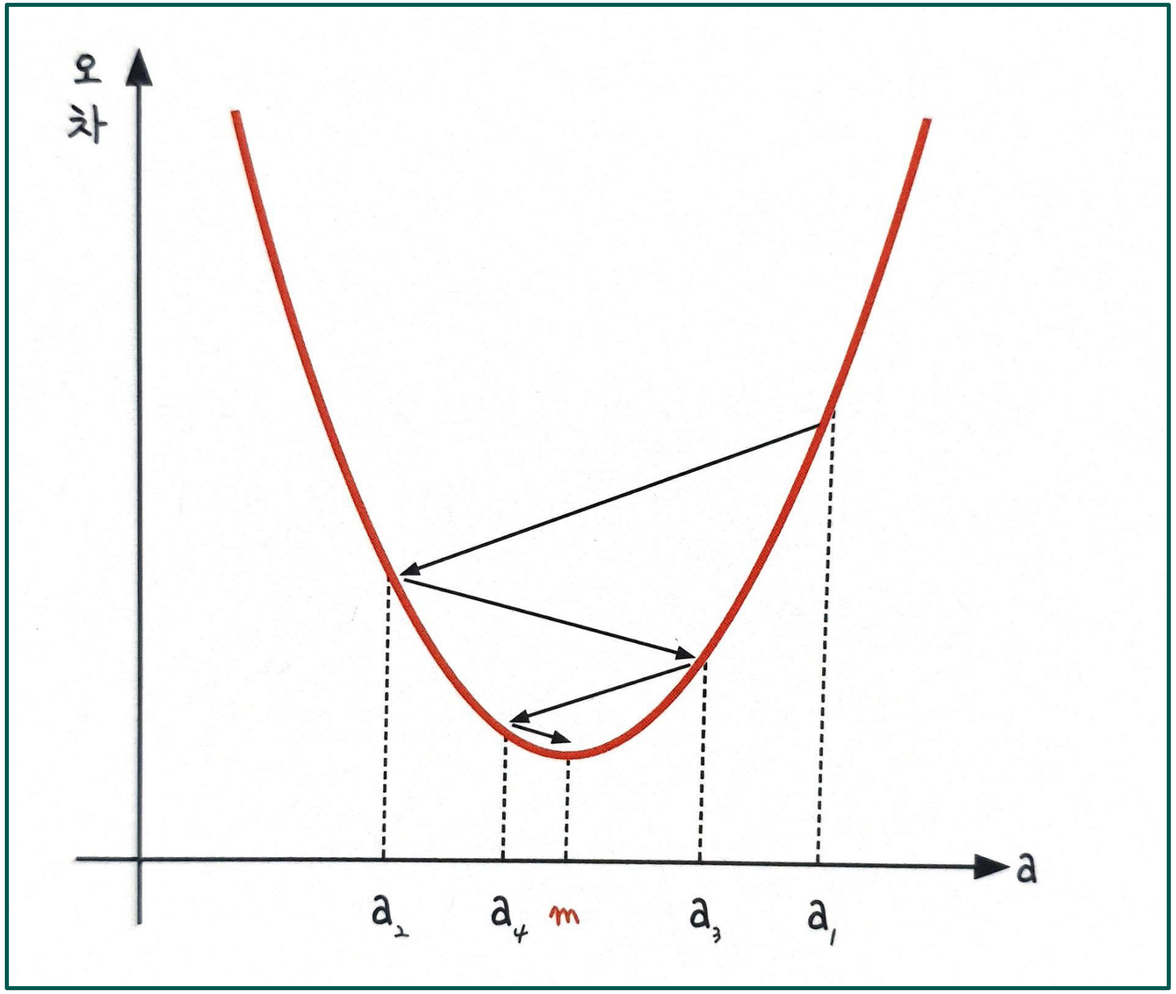
$f'(x) = 2x$



1-1.

과정

1. a_1 에서 미분을 구한다.
2. 구해진 기울기의 반대 방향으로 얼마간 이동시킨 a_2에서 미분을 구한다.
3. a_3 에서 미분을 구한다.
4. 값이 0이 아니면 과정 반복

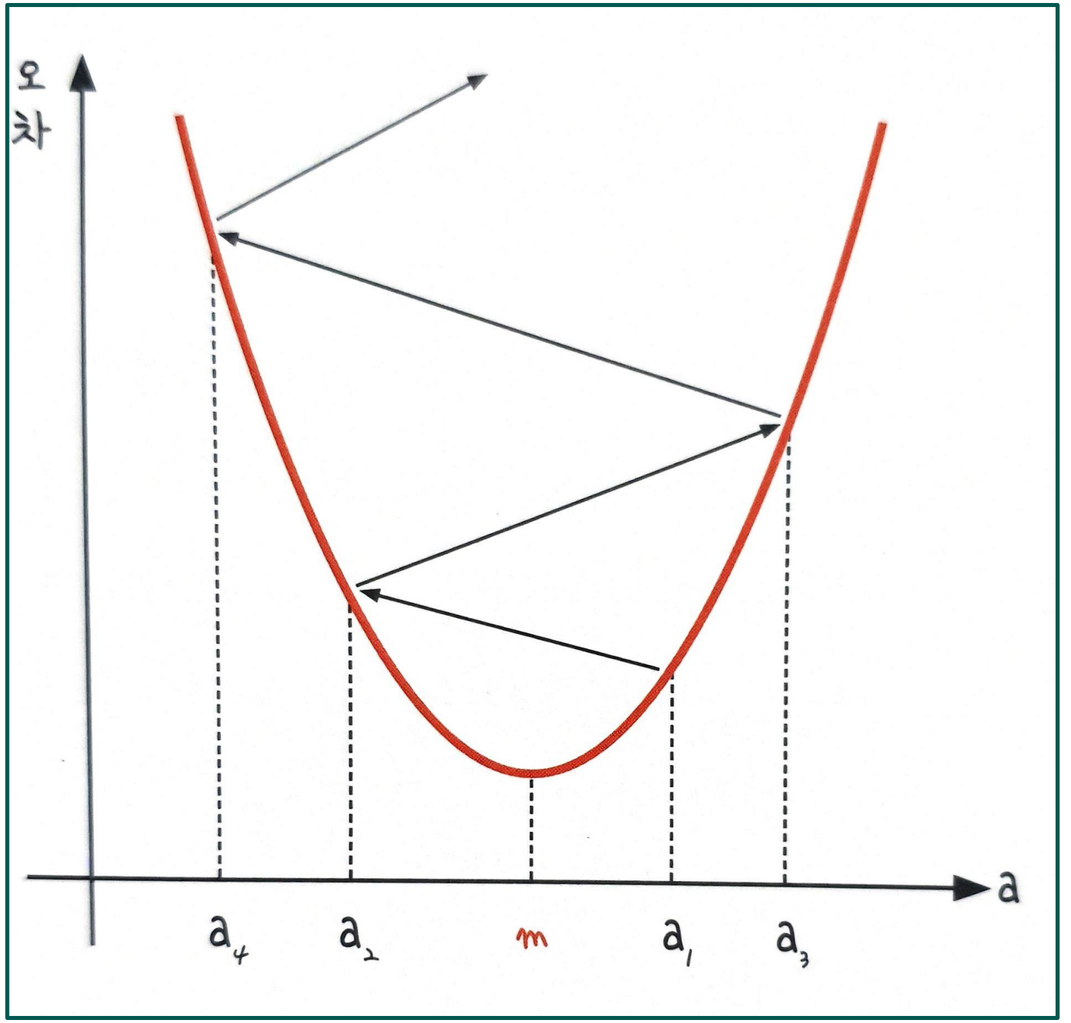


1-1.

학습률

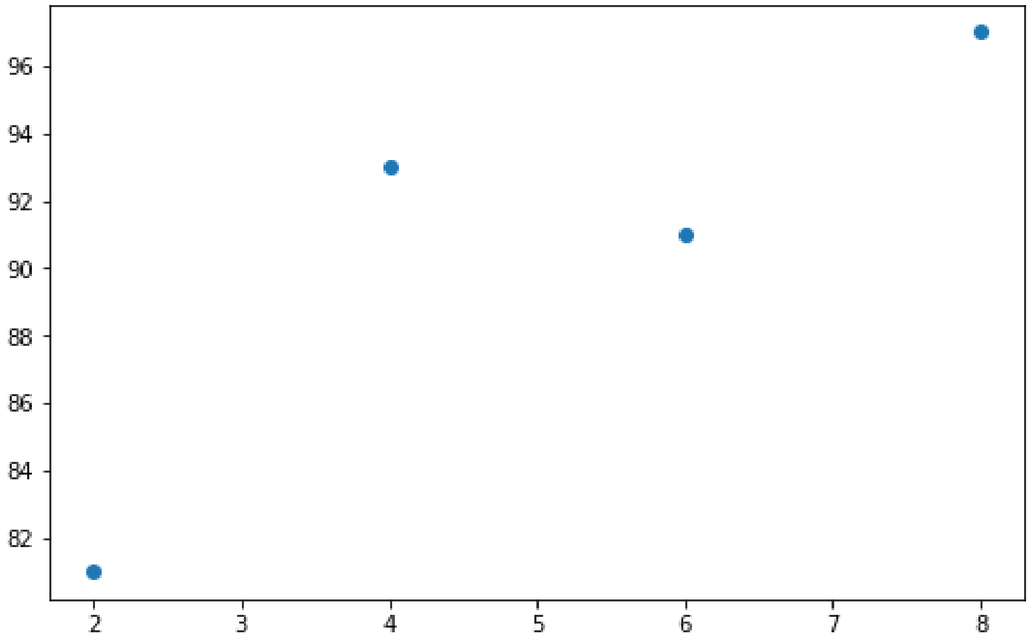
적절한 거리를 찾기 못해 너무 멀리 이동하면
a 값이 위로 솟음

이동 거리 정해주는 것 => **학습률**



1-2.

구현




```
data = [[2, 81], [4, 93], [6, 91], [8, 97]]
x = [i[0] for i in data]
y = [i[1] for i in data]


plt.figure(figsize=(8,5))
plt.scatter(x, y)
plt.show()


x_data = np.array(x)
y_data = np.array(y)


a = 0
b = 0


lr = 0.03


epochs = 2001
```

1-2.
구현

```
epoch=0, 기울기=27.8400, 절편=5.4300 오차=90.5000
epoch=100, 기울기=7.0739, 절편=50.5117 오차=4.6644
epoch=200, 기울기=4.0960, 절편=68.2822 오차=1.7548
epoch=300, 기울기=2.9757, 절편=74.9678 오차=0.6602
epoch=400, 기울기=2.5542, 절편=77.4830 오차=0.2484
epoch=500, 기울기=2.3956, 절편=78.4293 오차=0.0934
epoch=600, 기울기=2.3360, 절편=78.7853 오차=0.0352
epoch=700, 기울기=2.3135, 절편=78.9192 오차=0.0132
epoch=800, 기울기=2.3051, 절편=78.9696 오차=0.0050
epoch=900, 기울기=2.3019, 절편=78.9886 오차=0.0019
epoch=1000, 기울기=2.3007, 절편=78.9957 오차=0.0007
epoch=1100, 기울기=2.3003, 절편=78.9984 오차=0.0003
epoch=1200, 기울기=2.3001, 절편=78.9994 오차=0.0001
epoch=1300, 기울기=2.3000, 절편=78.9998 오차=0.0000
```

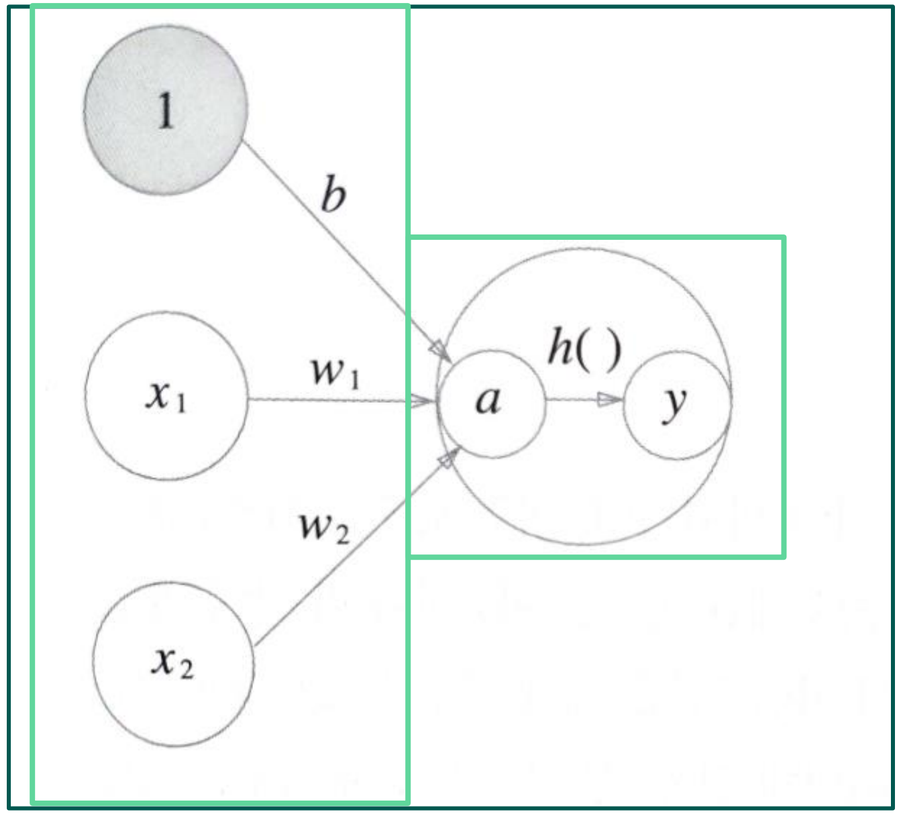
```python
for i in range(epochs):
    y_hat = a * x_data + b  #y를 구하는 식
    error = y_data - y_hat  #오차를 구하는 식
    a_diff = -(2/len(x_data)) * sum(x_data * (error)) # 오차함수를 a로 미분한 값
    b_diff = -(2/len(x_data)) * sum(error)  # 오차함수를 b로 미분한 값
    a = a - lr * a_diff  # 학습률을 곱해 기존의 a값을 업데이트
    b = b - lr * b_diff  # 학습률을 곱해 기존의 b값을 업데이트
    if i % 100 == 0:    # 100번 반복될 때마다 현재의 a값, b값을 출력
        print("epoch=%.f, 기울기=%.04f, 절편=%.04f 오차=%.04f" % (i, a, b,error.mean()))


y_pred = a * x_data + b
plt.scatter(x, y)
plt.plot([min(x_data), max(x_data)], [min(y_pred), max(y_pred)])
plt.show()
```

2-1.

퍼셉트론

$$a = b + w_1 x_1 + w_2 x_2$$

$$y = h(a)$$



perceptron

2-1.

되짚기

예측 => 정확한 a와 b의 값을 알아야함 (최소제곱법)

여러 개의 입력 처리 어려움 ↓

임의의 선을 그리고 오차를 구함 (평균 제곱근 오차) ↓

오차가 최소인 지점으로 계속해서 이동(경사하강법)

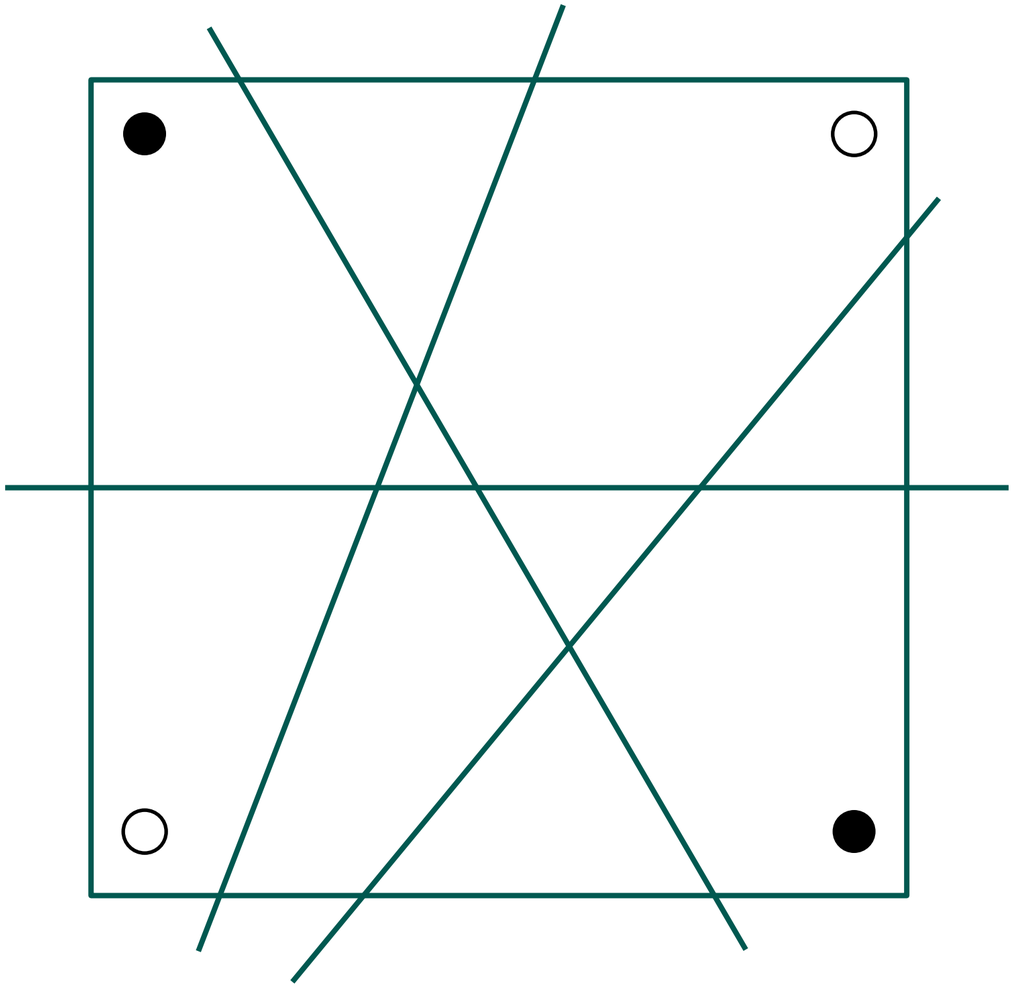
2-2.

배경

Q 선 하나로 한쪽면에는 검정, 다른 면에는 흰점만 있게 나눌 수 있을까?

A 없다.

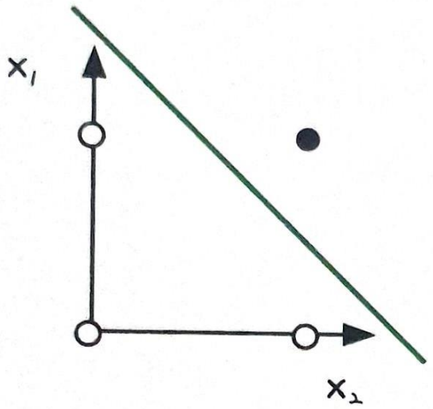
XOR 문제



2-2.

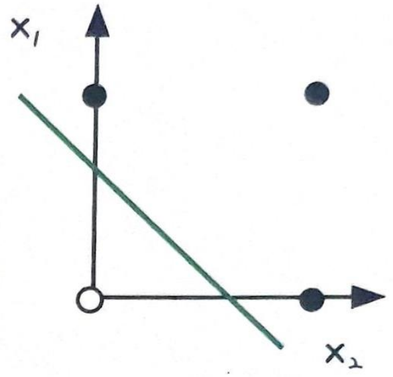
배경 XOR문제

검은점(1), 흰 점(0)



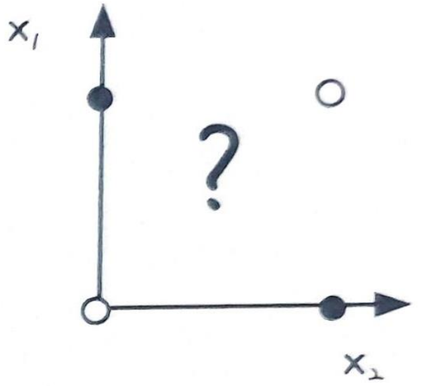
AND

X1	X2	Output
0	0	0
0	1	0
1	0	0
1	1	1



OR

X1	X2	Output
0	0	0
0	1	1
1	0	1
1	1	1

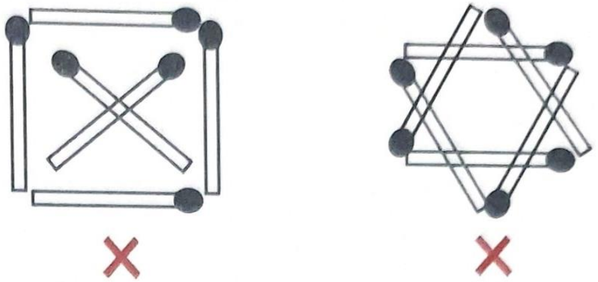


XOR

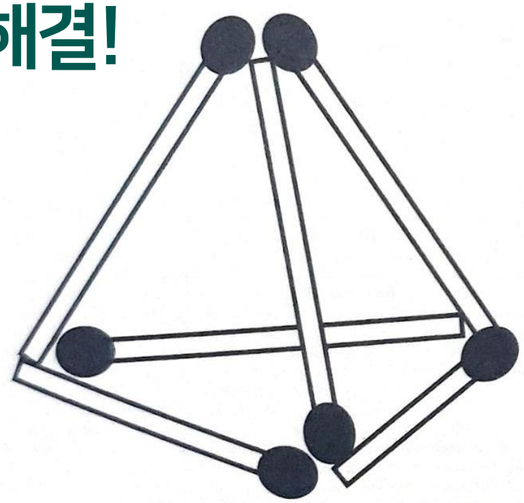
X1	X2	Output
0	0	0
0	1	1
1	0	1
1	1	0

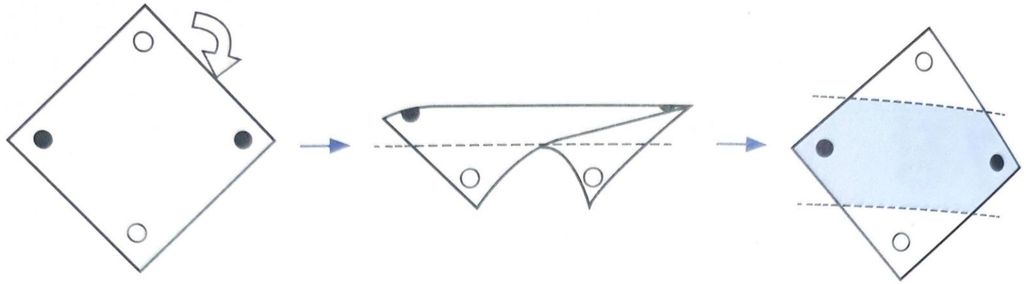
2-2.

배경 XOR문제



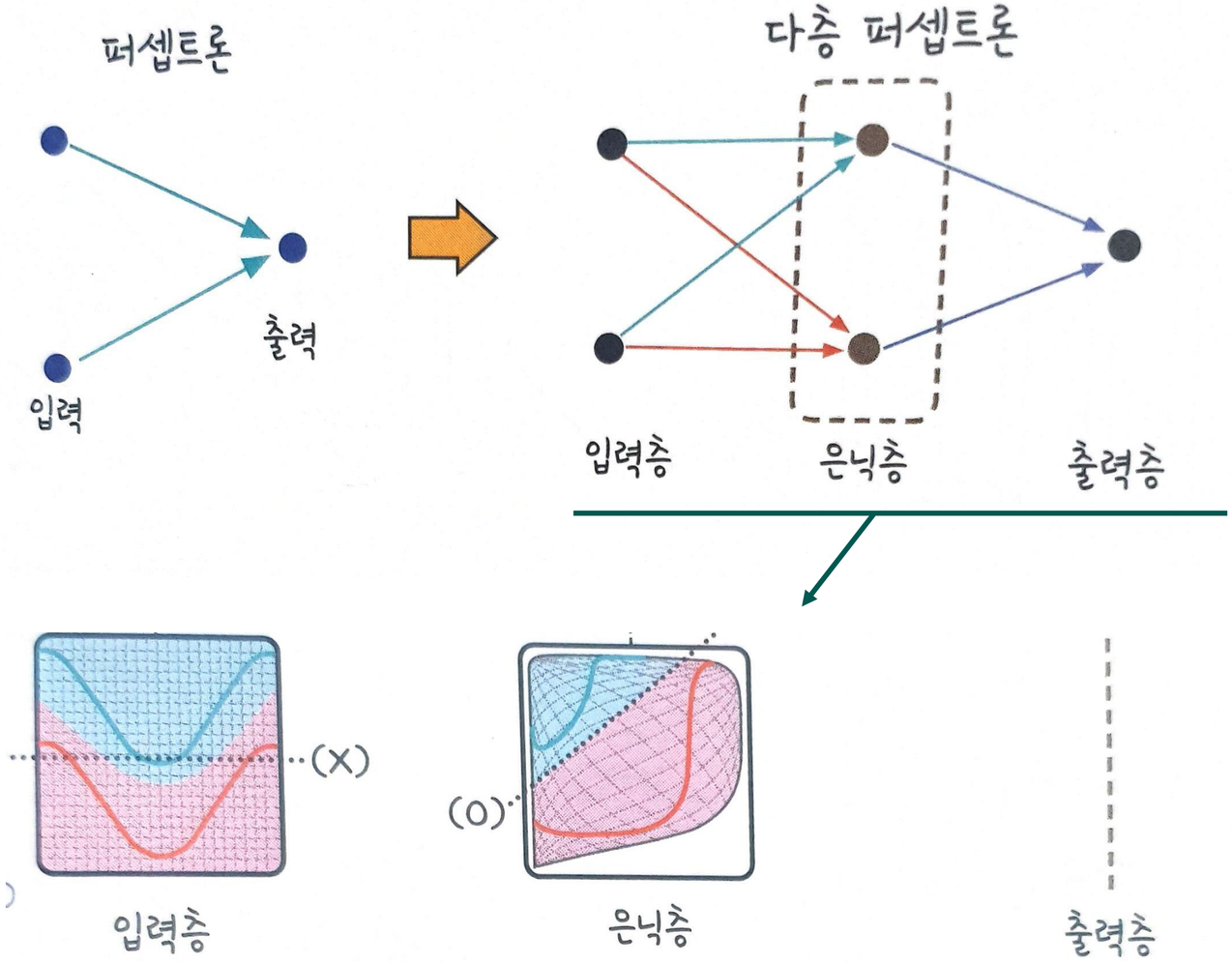
해결!





2-2.

배경 XOR문제



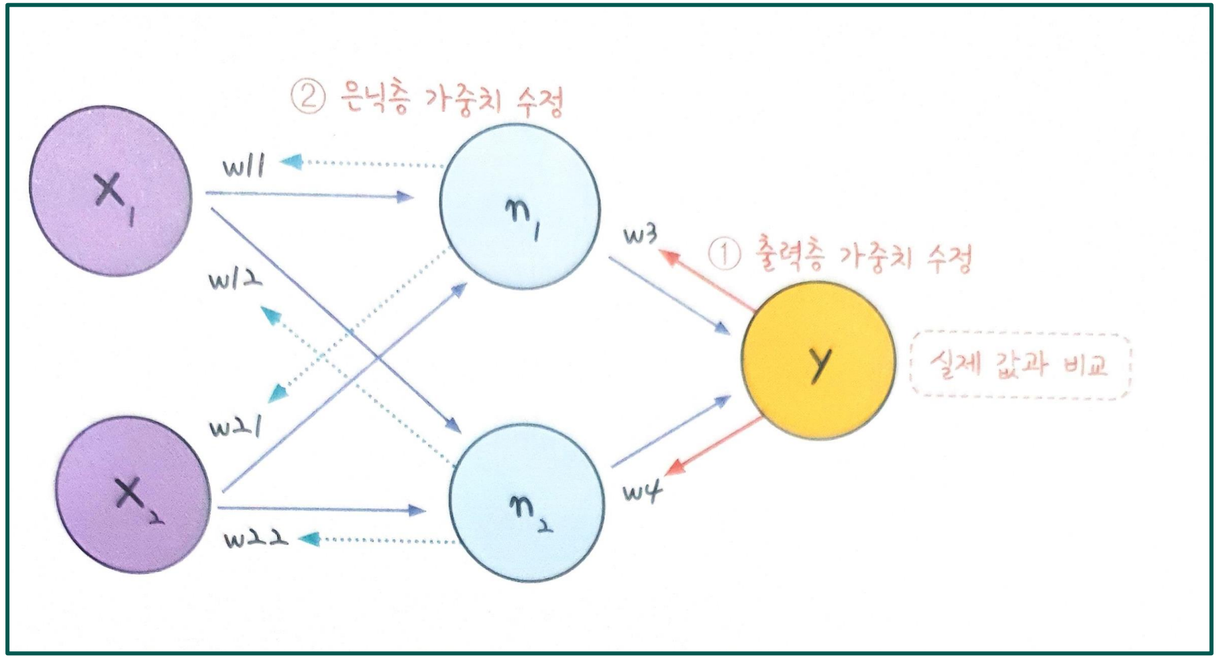
2-3.

개념

오차역전파

다층 퍼셉트론에서의 최적화 과정

1. 임의의 초기 가중치(W)를 준 뒤 결과를 계산
2. 계산 결과와 우리가 원하는 값 사이의 오차 구함
3. 경사하강법을 이용해 앞 가중치를 오차가 작아지는 방향으로 업데이트
4. 과정 반복



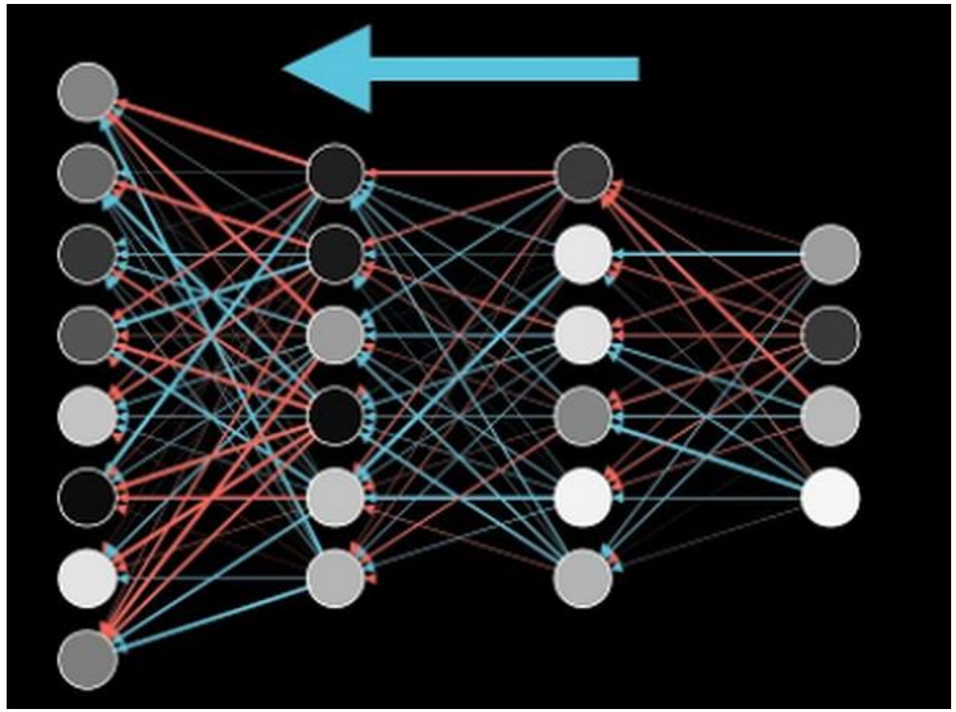
$$W(t+1) = Wt - \frac{\partial 오차}{\partial W}$$

2-3.

개념

Q 왜 오차역전파를 사용할까?

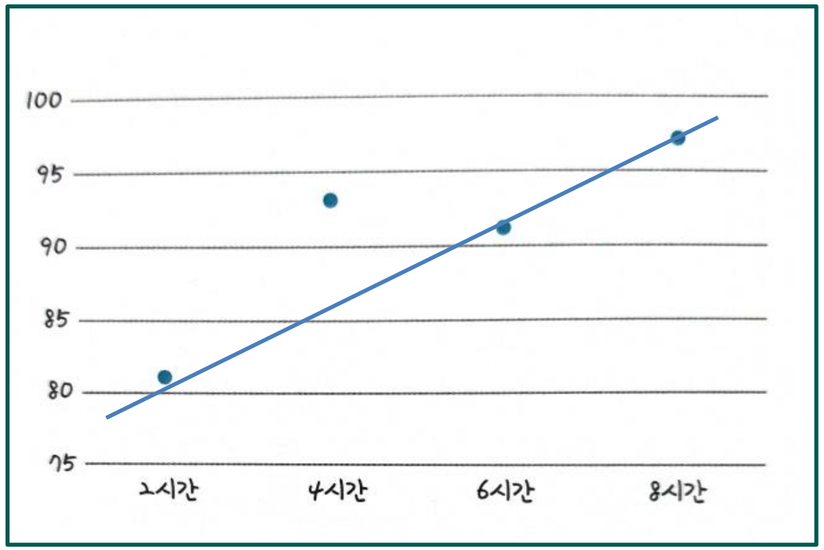
A 효율적!!



plus.

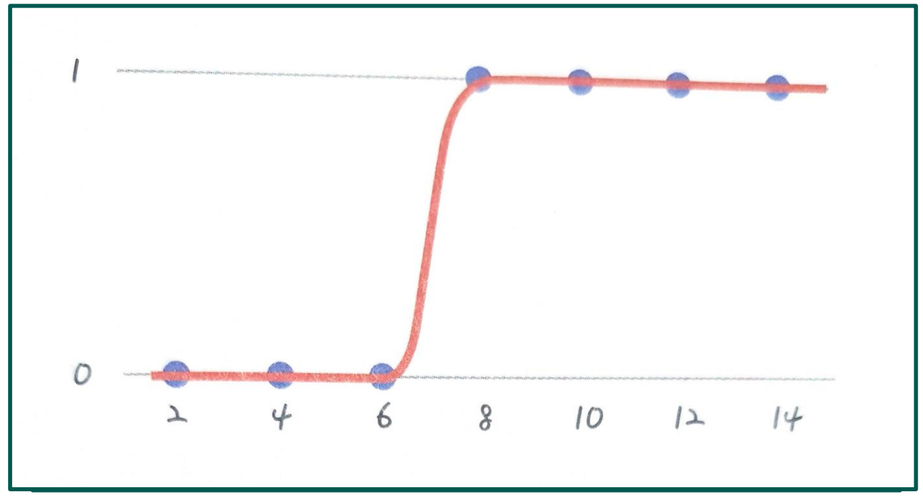
Q. 상황에 맞는 함수는?

A 선형회귀



ex) 공부 시간에 따라 성적이 오른다.

B 시그모이드



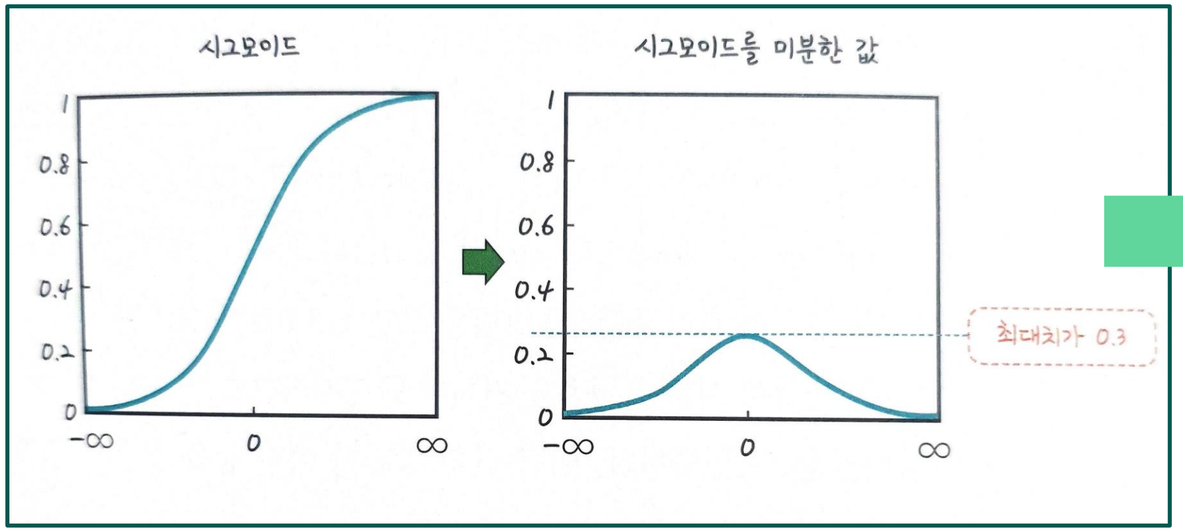
ex) 공부 시간에 따라 합격(1)/불합격(0)

plus.

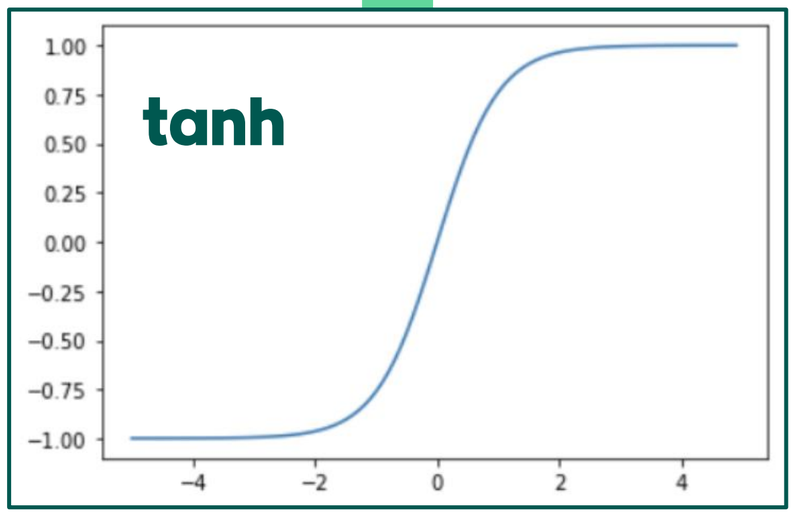
Q. 상황에 맞는 함수는?

문제 발생

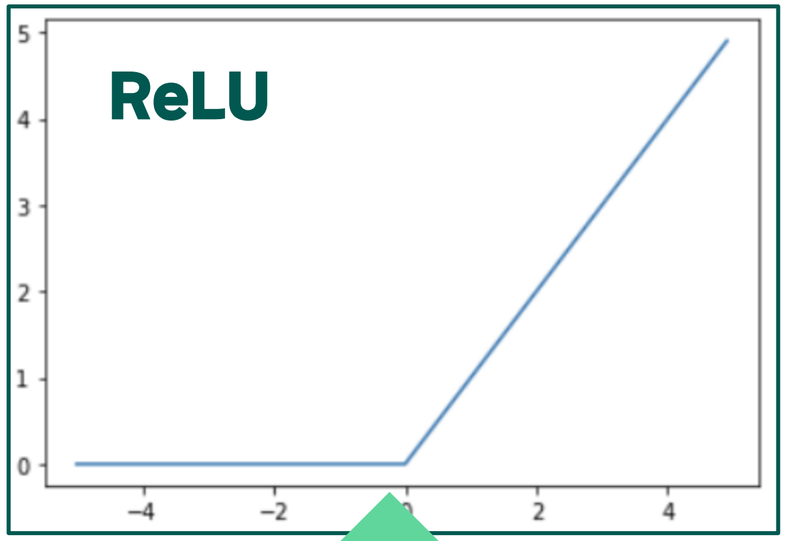
층이 늘어나면서 기울기 소실 문제!



C



D



THANK YOU

감사합니다

21.07.27

장혜선