2024 D&A

인공신경망과 오차역전파법

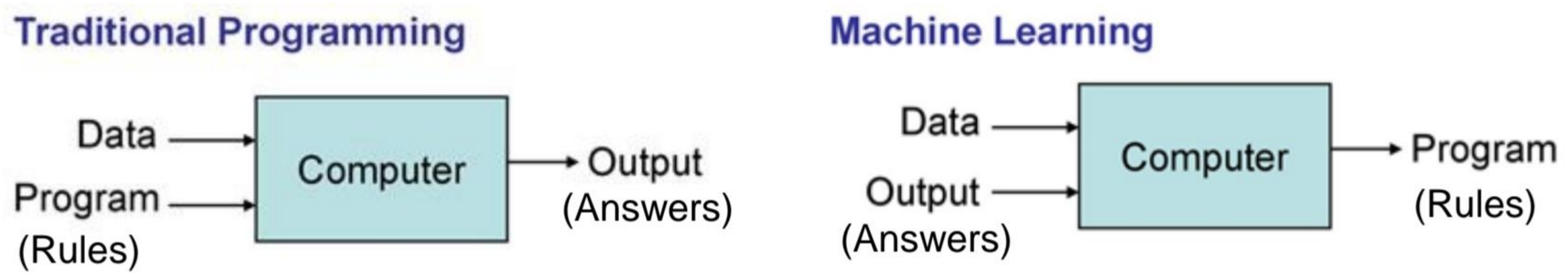
Deep Session 2차시

CONTENTS.

머신러닝

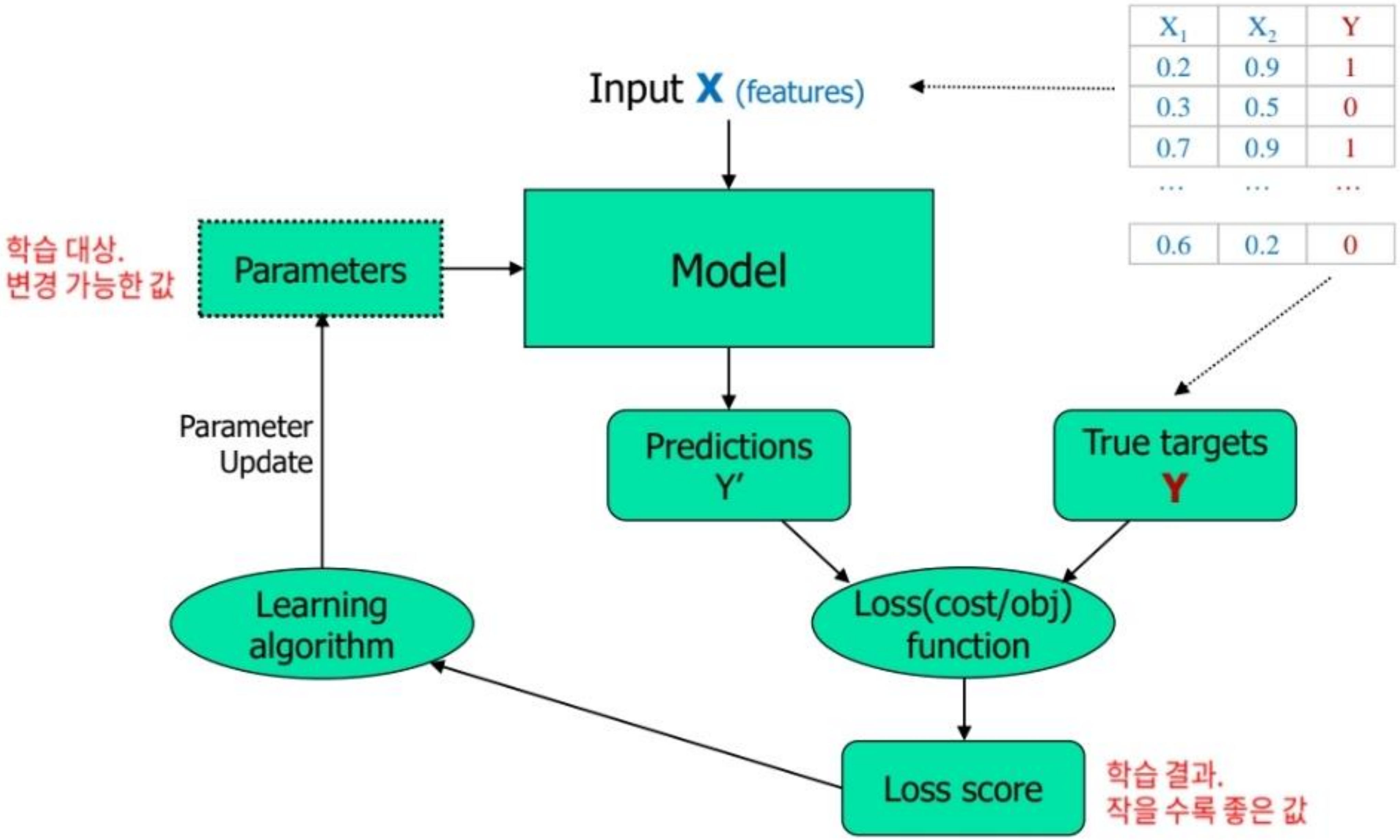
머신러닝

- 기계가 일일이 코드로 명시하지 않은 동작을 데이터로부터 학습하여 실행할 수 있도록 하는 알고리즘을 개발하는 연구분야
  Arthur Samuel(1959)



머신러닝의 동작과정

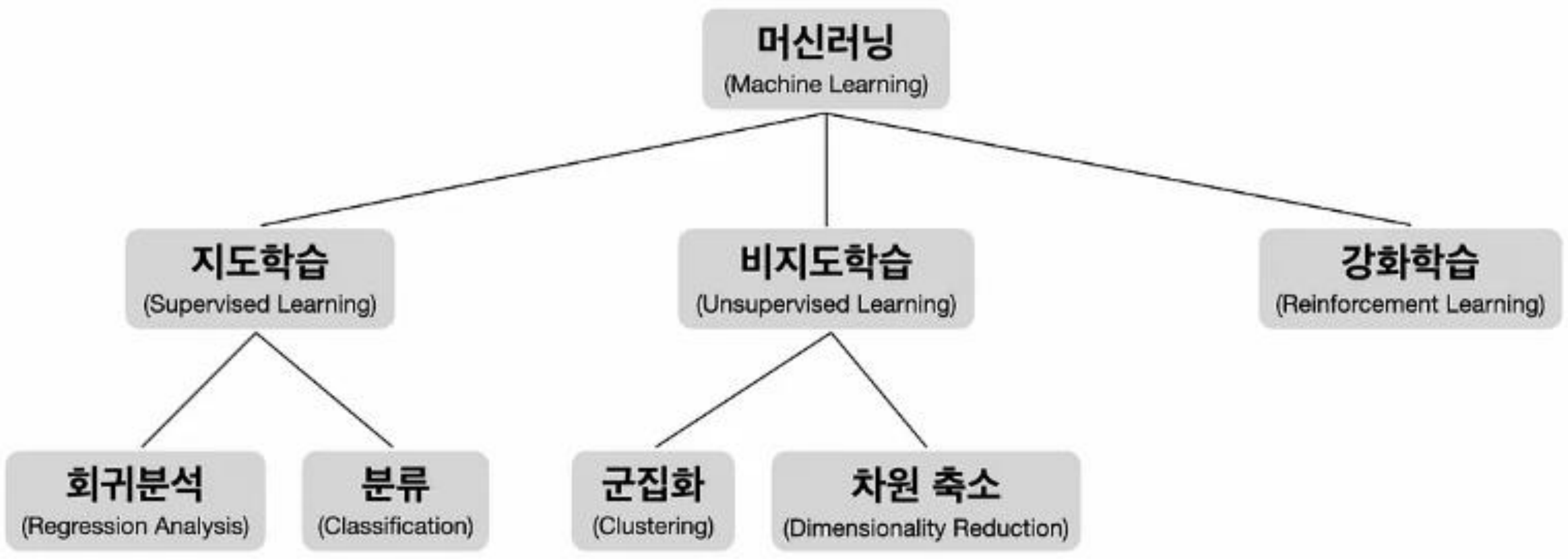
동작과정

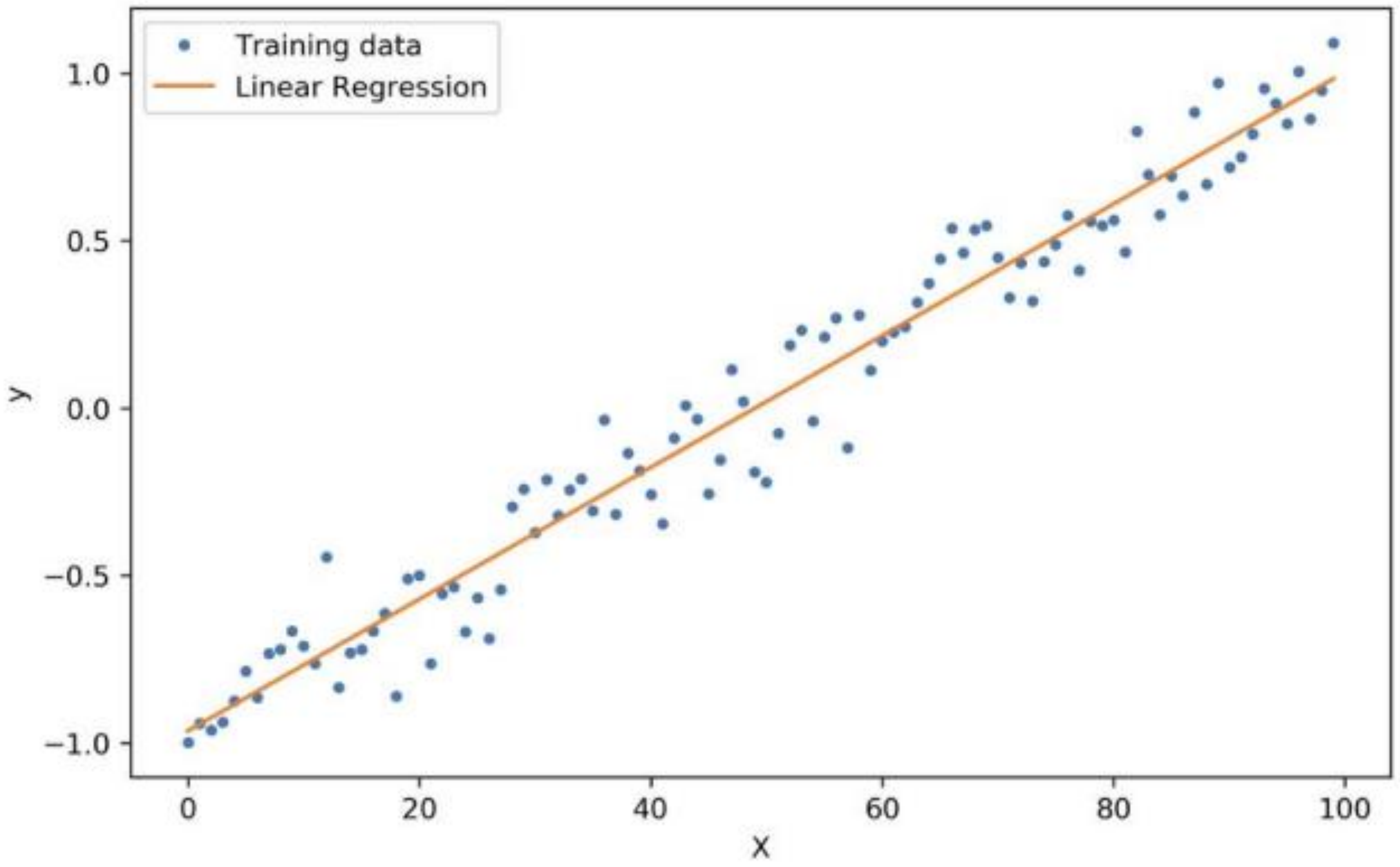


"Find the parameters that minimize the loss"

선형회귀

선형회귀


머신러닝
(Machine Learning)
지도학습
(Supervised Learning)
비지도학습
(Unsupervised Learning)
강화학습
(Reinforcement Learning)
회귀분석
(Regression Analysis)
분류
(Classification)
군집화
(Clustering)
차원 축소
(Dimensionality Reduction)


Training data
Linear Regression
1.0
0.5
0.0
−0.5
−1.0
y
0
20
40
60
80
100
X

선형회귀

선형회귀

지도학습 중 회귀문제를 예시로 살펴보자.

a, b 라는 파라미터를 변경하면서 최적의 함수를 찾는다.

MSE 등 loss function을 줄이는 방향으로 파라미터 업데이트

어떤 파라미터가 좋은 파라미터?

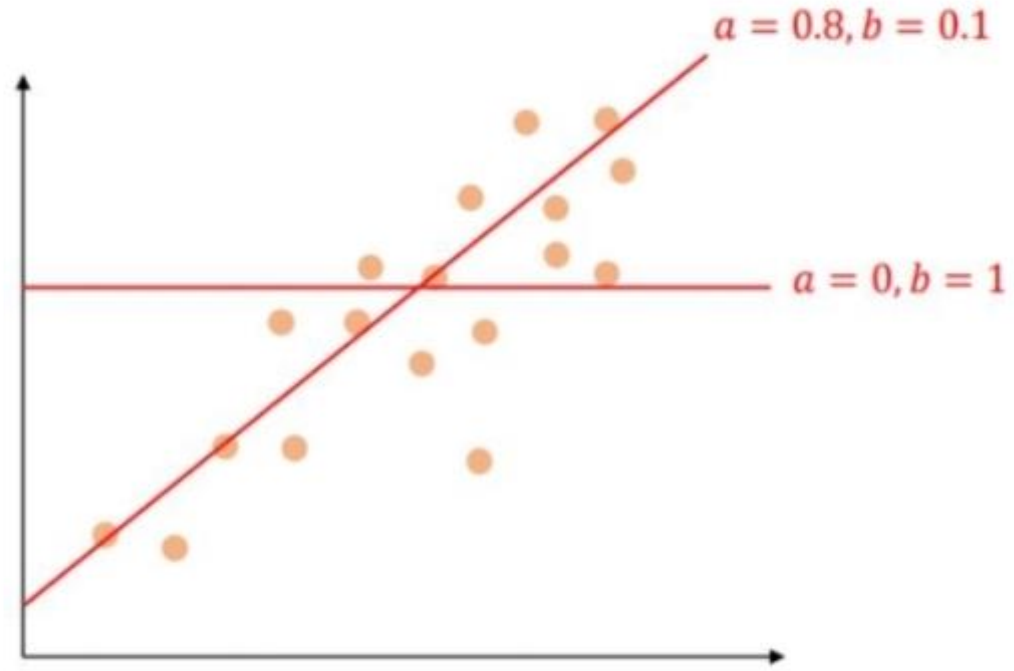
파라미터 a, b의 조합을 봤을 때 loss function이 최소가 되는 파라미터의 조합이 좋은 파라미터다~

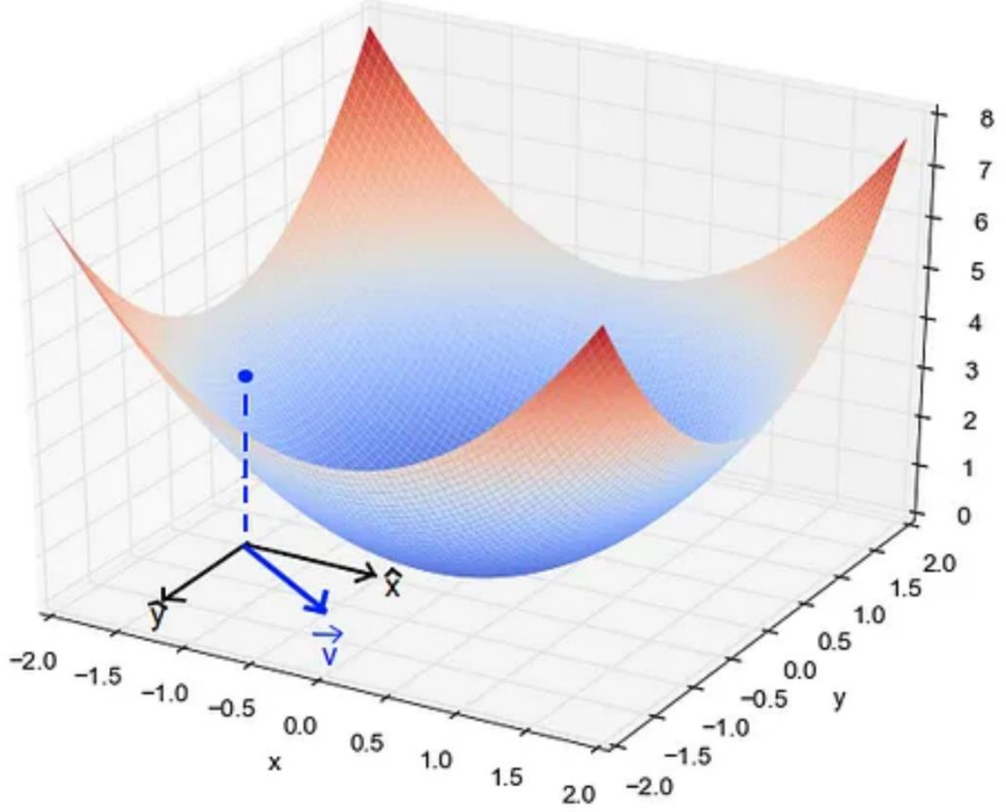
학습 모델

$$y = \underline{a}x + \underline{b}$$

출력 입력

학습 매개변수





경사하강법이란

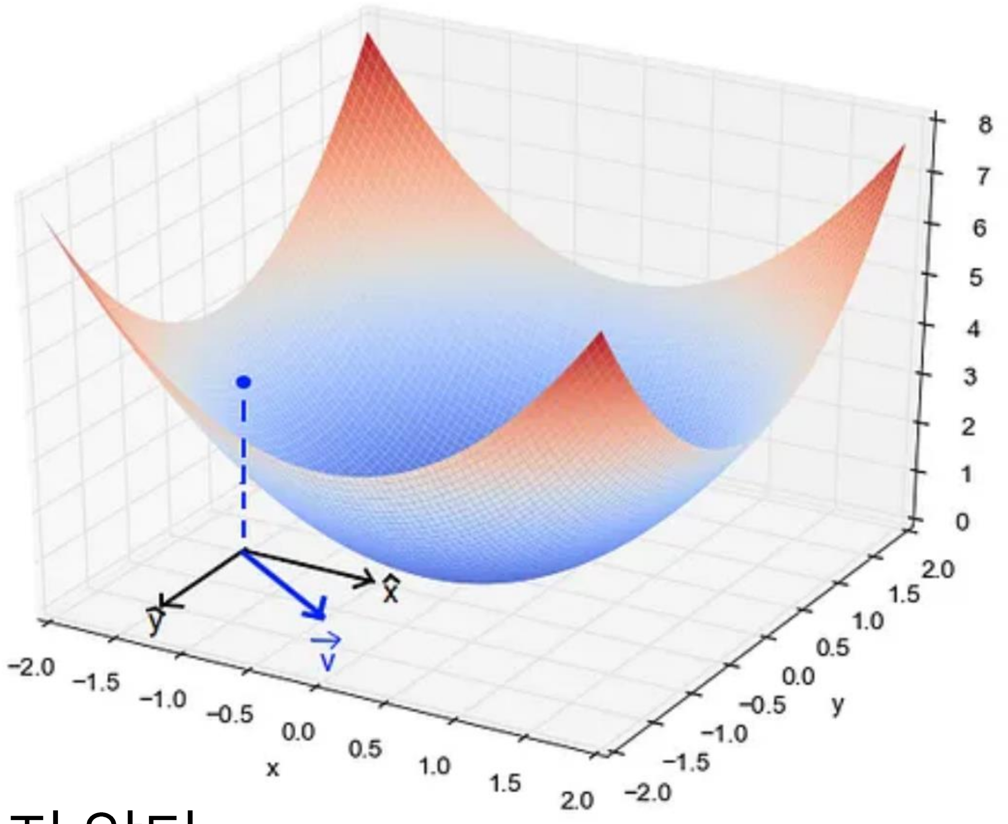
경사하강법의 필요성

모든 a,b의 조합을 고려해서 loss function의 최소를 만드는 최적의 조합을 찾는 것은 계산비용이 많이 든다.

무작위의 어떤 초기값을 기준으로 가장 가파른 방향으로 한 걸음씩 가면 어떨까?

그것이 gradient descent!

$$W := W - \alpha \frac{\partial}{\partial W} cost(W)$$



퍼셉트론에서의 경사하강

다음에서 살펴보겠지만 퍼셉트론, 나아가 딥러닝에서 사용되는 파라미터수는 엄청나다.

따라서 모든 지역에서의 loss function의 최솟값을 찾는것이 어려운 것은 물론 시각화 하기도 쉽지 않다.

다음 슬라이드에 등장할 퍼셉트론을 보면서 어떻게 파라미터를 업데이트 하면 좋을지 생각해보면서 들어보자!

퍼셉트론

생물학적 뉴런의 네트워크

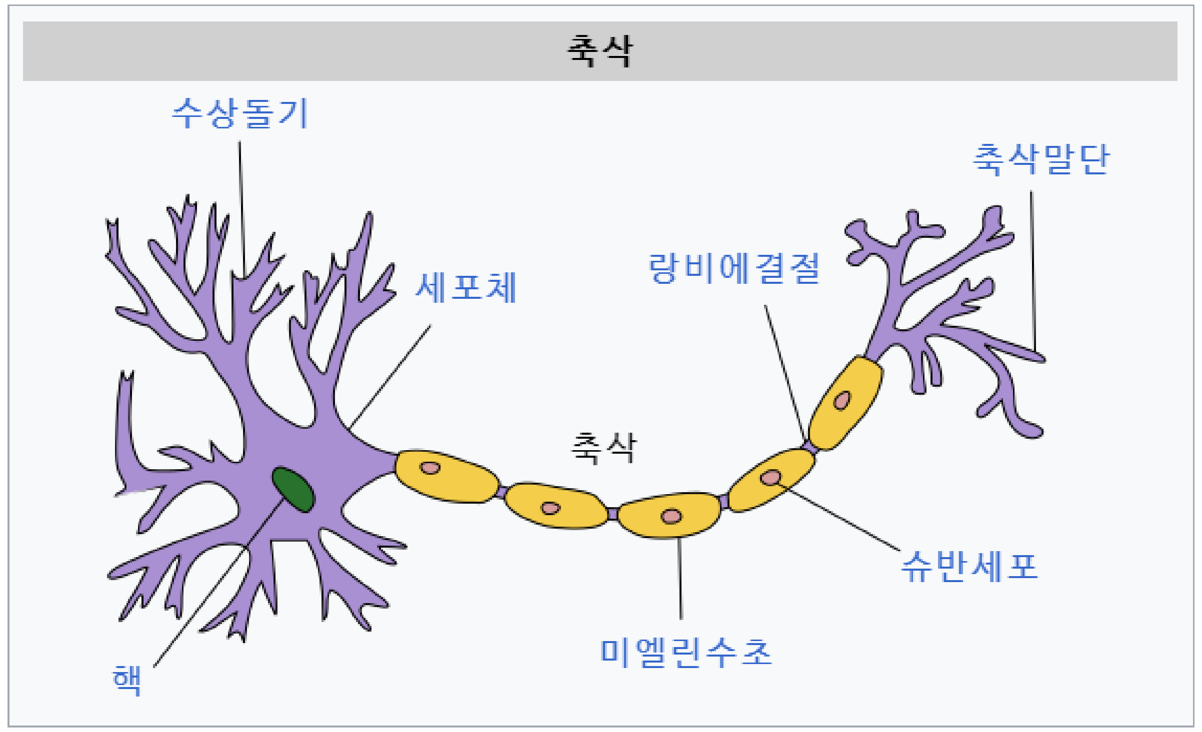
인공신경망은 신경세포에서 영감을 받았다.

축삭말단은 다른 뉴런의 세포체나 수상돌기와 시냅스를 통해 연결되어 있다.

뉴런은 충분한 양의 신경전달물질을 받았을 때, 자체적인 신호를 발생시킨다.

(뉴런의 신호 발생을 막는 신경전달물질도 존재)

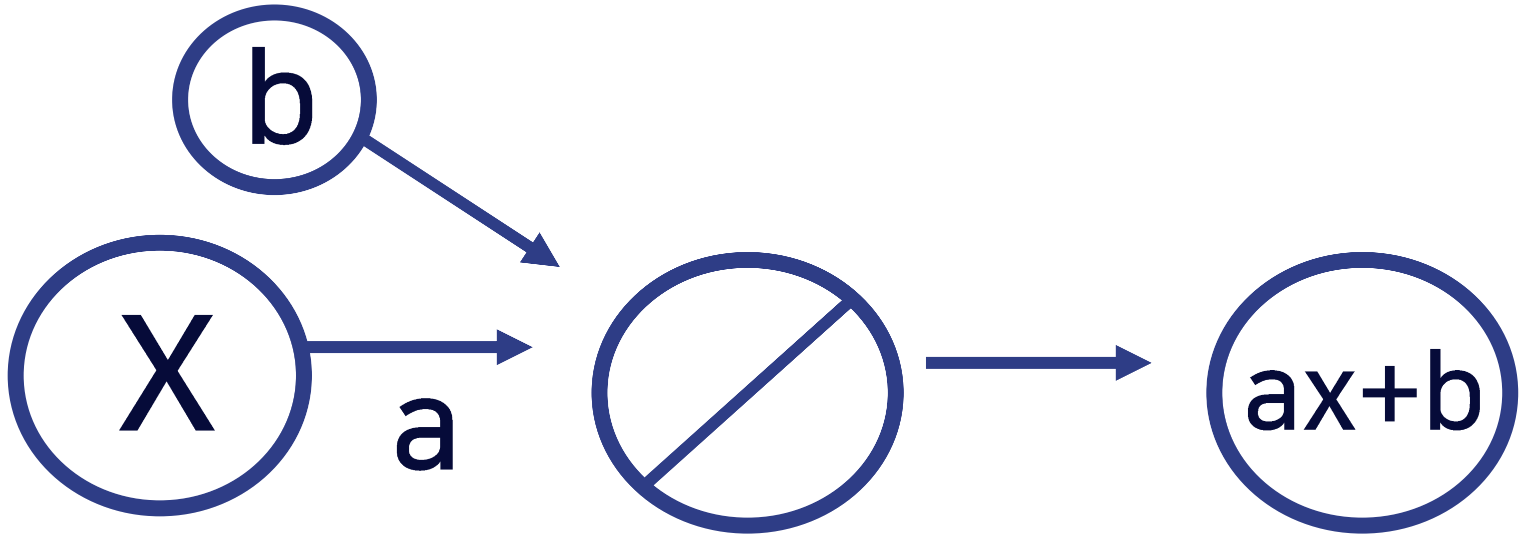
일반적인 뉴런의 구조



퍼셉트론으로 선형회귀를 표현

Input x에 weight a를 곱하고, Bias b를 더한 다음

노드의 활성화 정도에 따라 다음 노드로 값이 전달된다.

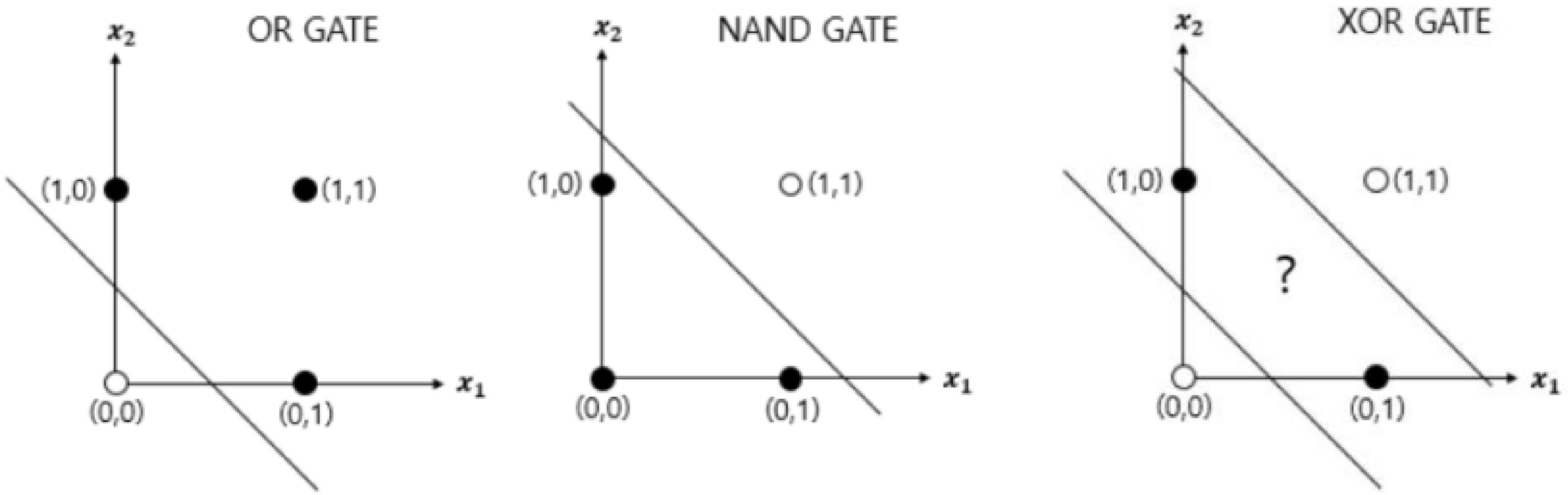


퍼셉트론

퍼셉트론의 필요성

XOR와 같은 비선형성을 표현할 수 있다.

단순한 선형 함수로 표현하지 못했던 복잡한 함수에 대한 학습을 할 수 있다.



x1	x2	s1	s2	y
0	0	1	0	0
1	0	1	1	1
0	1	1	1	1
1	1	0	1	0

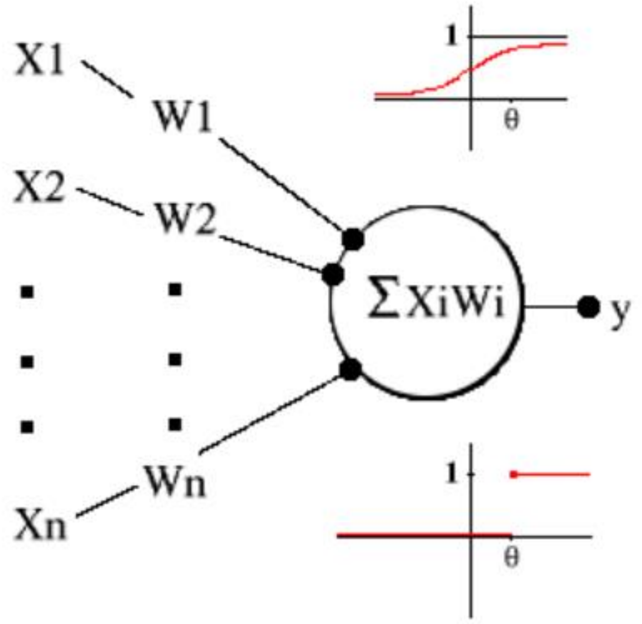
활성화 함수

TLU

Threshold logic unit

입력값의 weighted sum(가중합)을 계산한 다음,

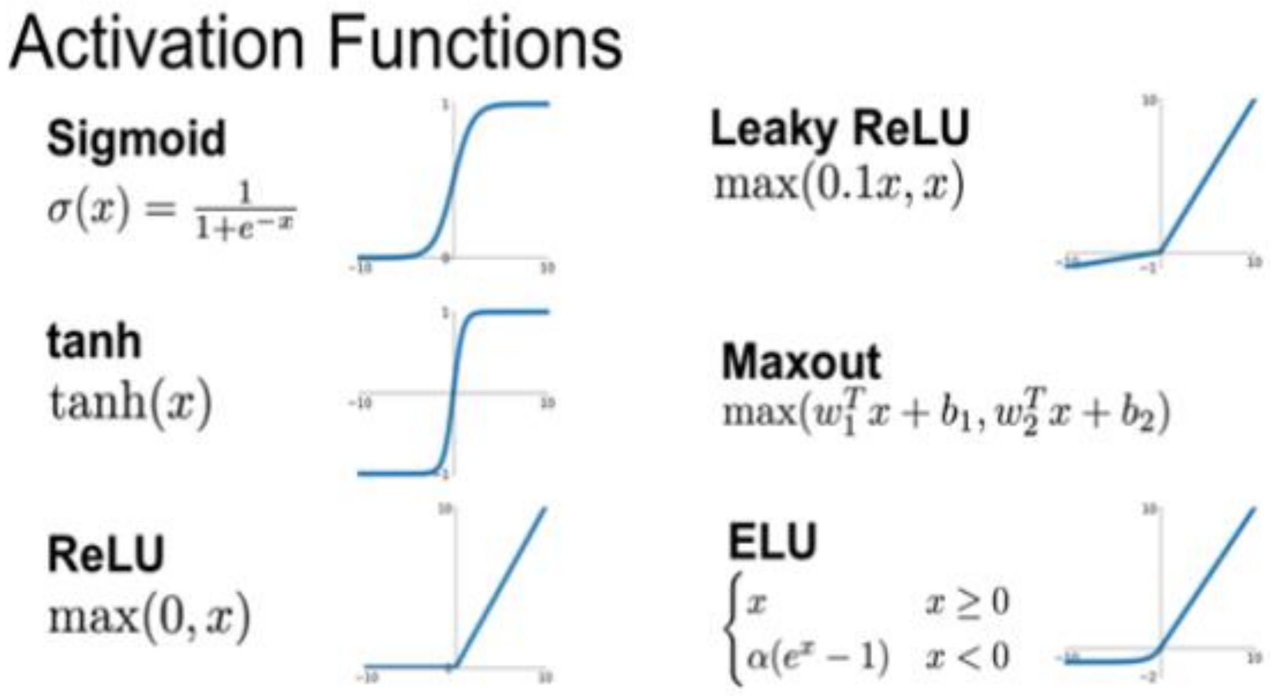
계단 함수를 적용하여 결과를 출력한다.



활성화 함수의 필요성

다음 뉴런으로 정보를 얼마나 흘려보낼지를 결정한다.

활성화 정도나 방법은 활성화 함수에 따라 다르다.



MLP

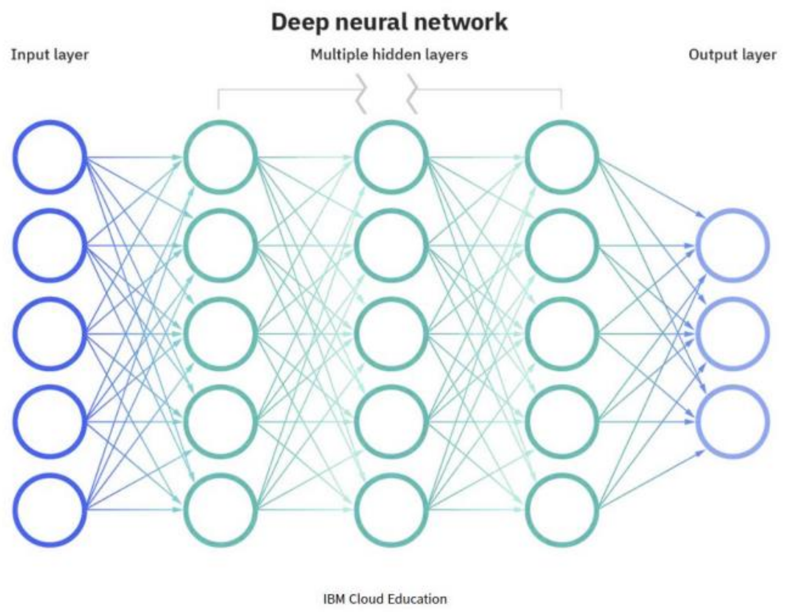
MLP의 구성

Input layer : 입력층

Output layer : 출력층

Hidden layer : 은닉층

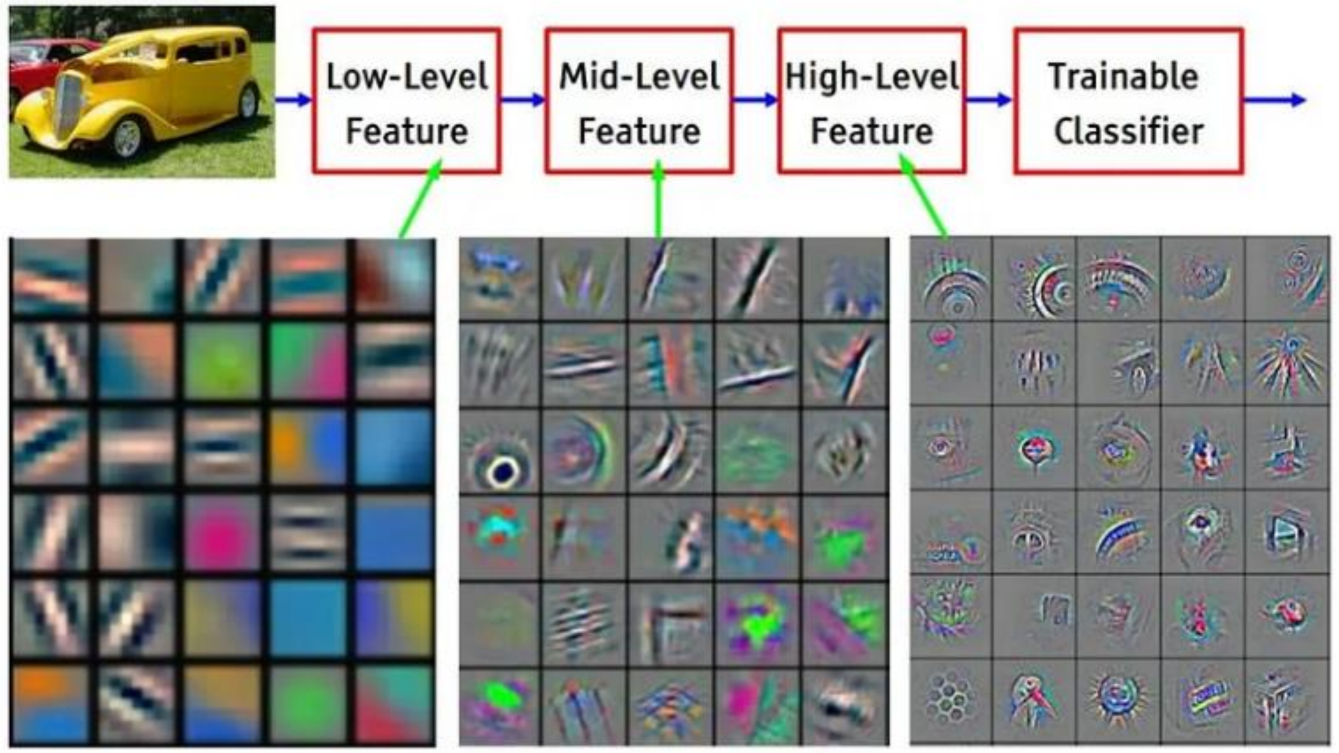
Node : 노드



참고사항

CNN에서 각 필터가 이미지에서 포착하는 특징이 layer별로 다름.

MLP의 가중치가 대칭이면 안된다.



순전파와 역전파

순전파

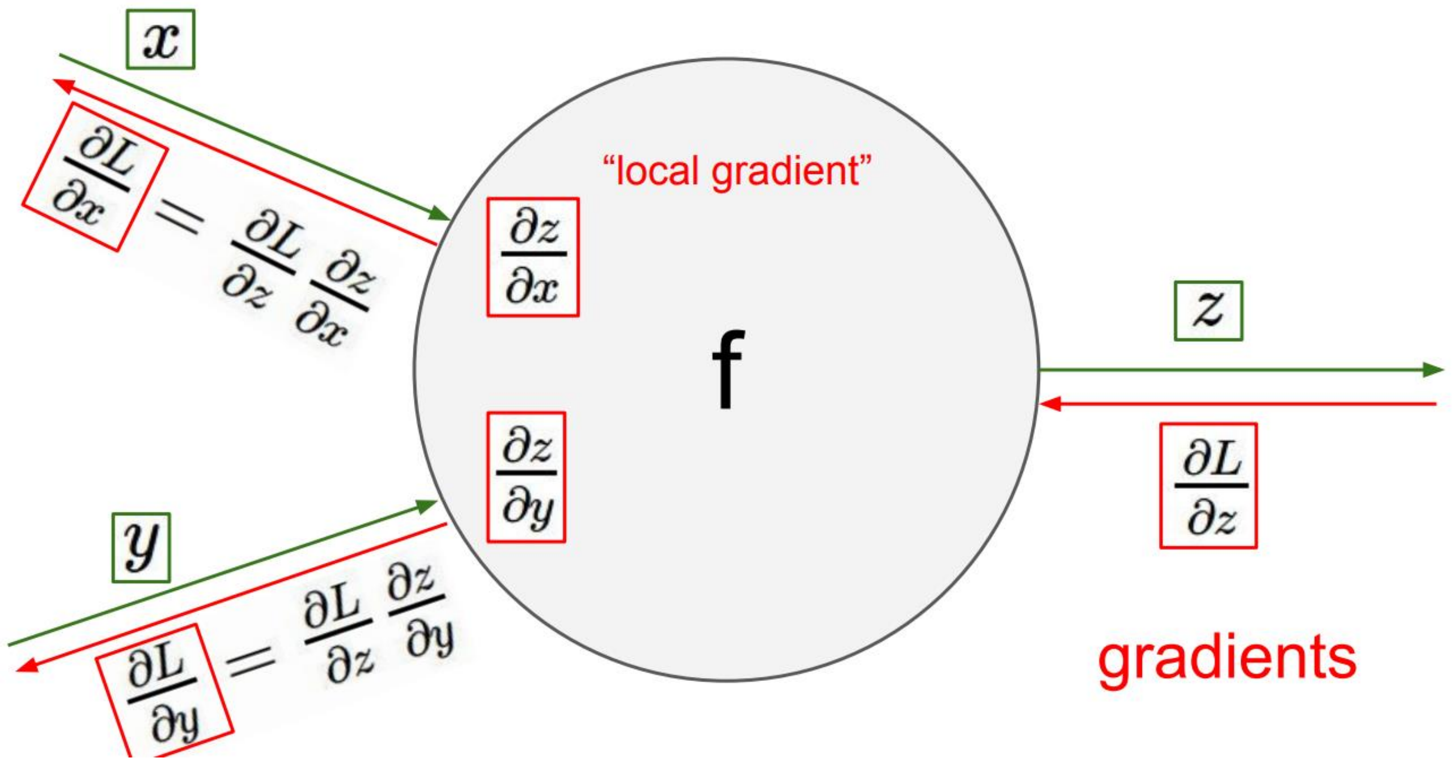
MLP의 매개변수들을 활용하여 결과값을 계산하는 방법

- 이전 layer에서 넘어온 값에 가중치(w)와 편향(b)를 적용해 다음 layer로 넘기는 방식
- Output layer에서 순전파의 결과인 예측값과 실제값의 차이(loss)를 계산한다.
- 순방향으로 값을 전파한다~

역전파

MLP의 매개변수들을 업데이트하는 과정

- Loss를 활용하여 각 layer의 weight와 bias를 최적화한다.
- Loss function은 최소화하는 방향으로 weight와 bais를 수정한다.
- 역방향으로 파라미터를 업데이트해 나간다 ~



계산그래프

계산그래프 1

원우형이 슈퍼에서 1개에 100원인 사과를 2개 구매했다….

-> 이때! 지불 금액을 구하세요! 단, 소비세가 10% 부과됩니다~

계산그래프 2

이번엔 원우형이 슈퍼에서 사과를 2개 귤을 3개 구매했다….

사과는 1개에 100원, 귤은 1개에 150원

-> 이때, 지불 금액을 구하세요! (단, 소비세가 10% 부과됩니다~)

근데, 왜 계산그래프?

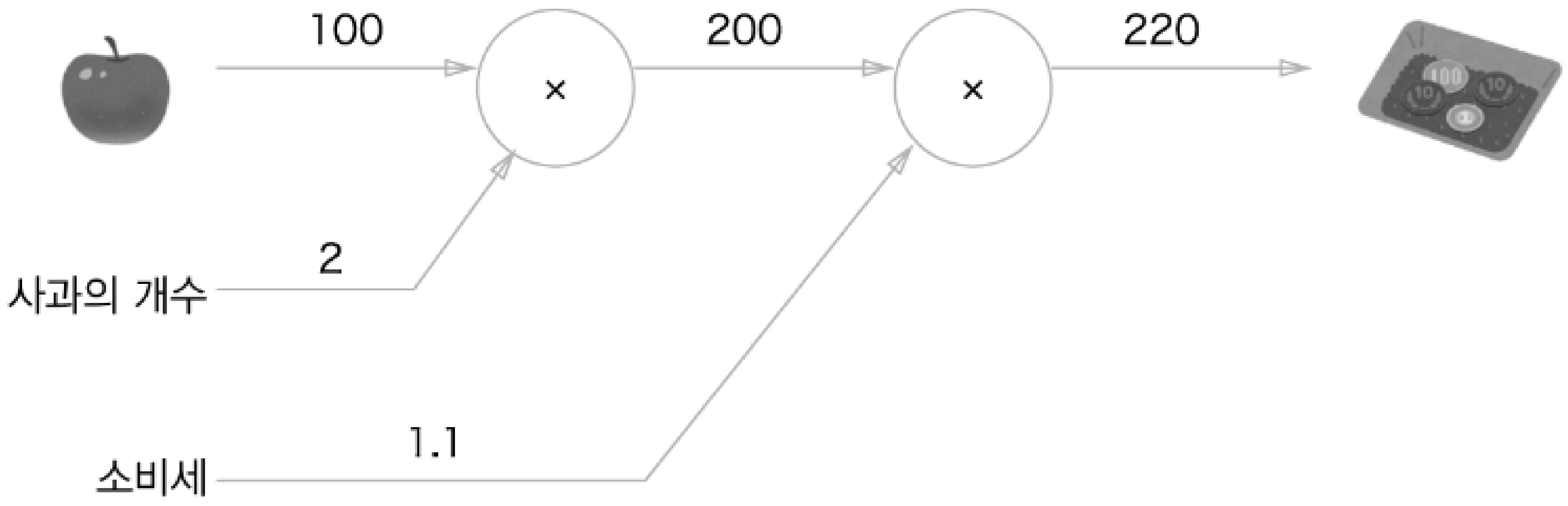
- 일부분의 단순한 계산으로 복잡한 전체 계산을 할 수 있기 때문.
- 복잡한 문제를 단순한 문제로 분해해서 풀어내는 것~

계산그래프

계산그래프 1

원우형이 슈퍼에서 1개에 100원인 사과를 2개 구매했다….

-> 이때! 지불 금액을 구하세요! 단, 소비세가 10% 부과됩니다~



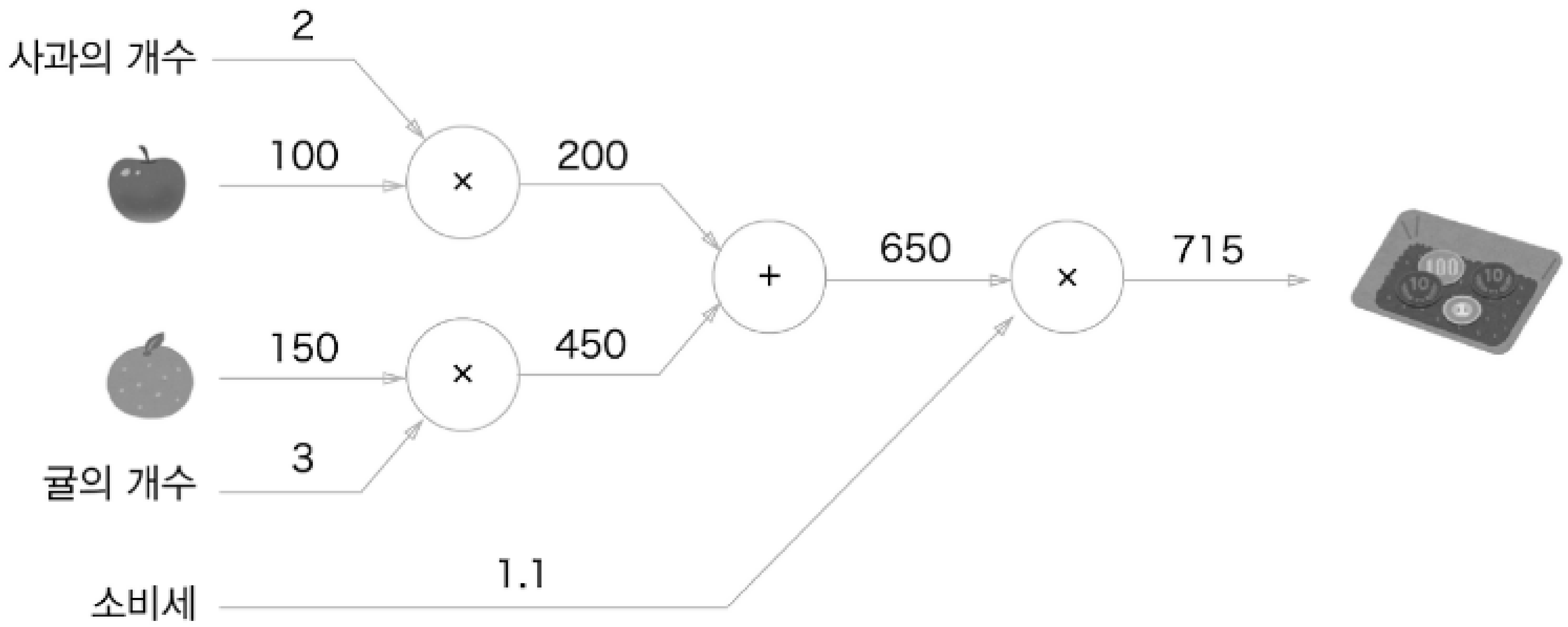
계산그래프

계산그래프 2

이번엔 원우형이 슈퍼에서 사과를 2개 귤을 3개 구매했다….

사과는 1개에 100원, 귤은 1개에 150원

-> 이때, 지불 금액을 구하세요! (단, 소비세가 10% 부과됩니다~)



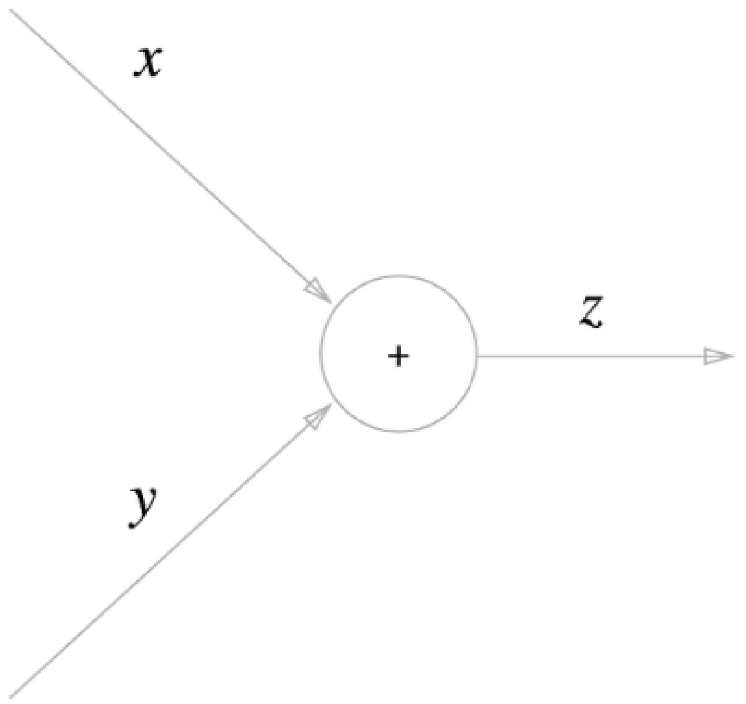
순전파와 역전파

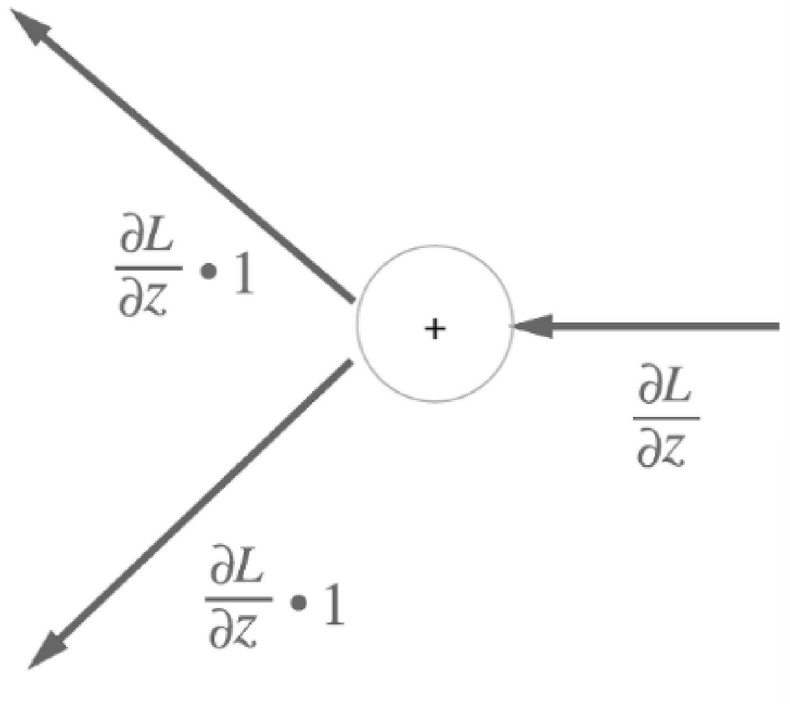
덧셈노드 역전파

z = x + y

$$\frac{\partial z}{\partial x} = 1$$

$$\frac{\partial z}{\partial y} = 1$$



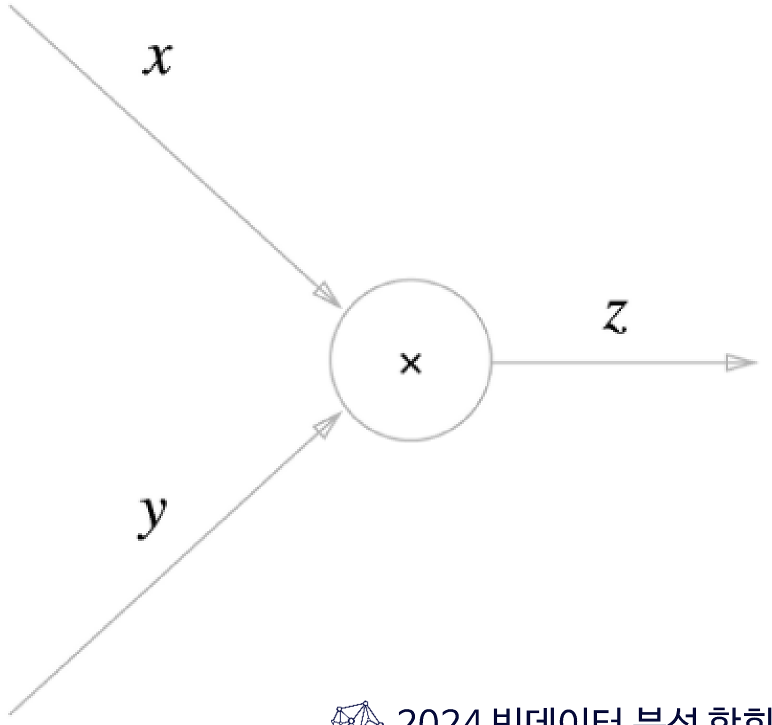


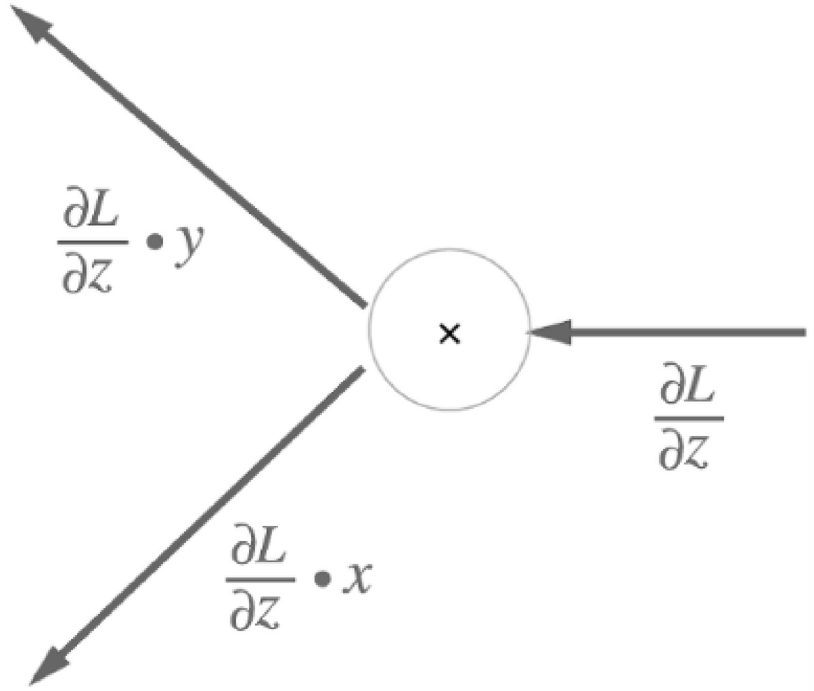
곱셈노드 역전파

z = xy

$$\frac{\partial z}{\partial x} = y$$

$$\frac{\partial z}{\partial y} = x$$



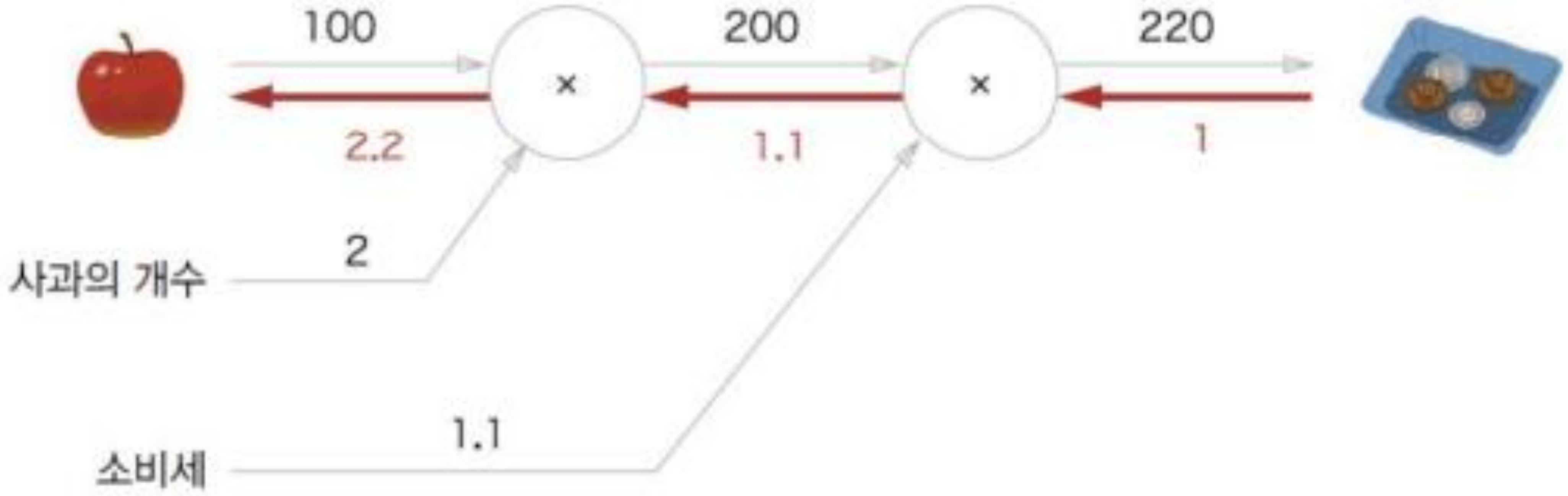


역전파

간단한 역전파를 해보자

-> : 순전파
<- : 역전파

사과의 가격 : 100원
사과의 개수 : 2
소비세 : 10%



참고사항

Chain rule (연쇄법칙)

- 합성 함수의 미분은 합성 함수를 구성하는 각 함수의 미분의 곱으로 나타낼 수 있다.
- 역전파 시, 국소적 미분을 전달하는 원리는 연쇄법칙을 따르므로 숙지할 것.

- $z = (x + y)^2$
  - $z = t^2$
  - $t = x + y$

$\rightarrow$

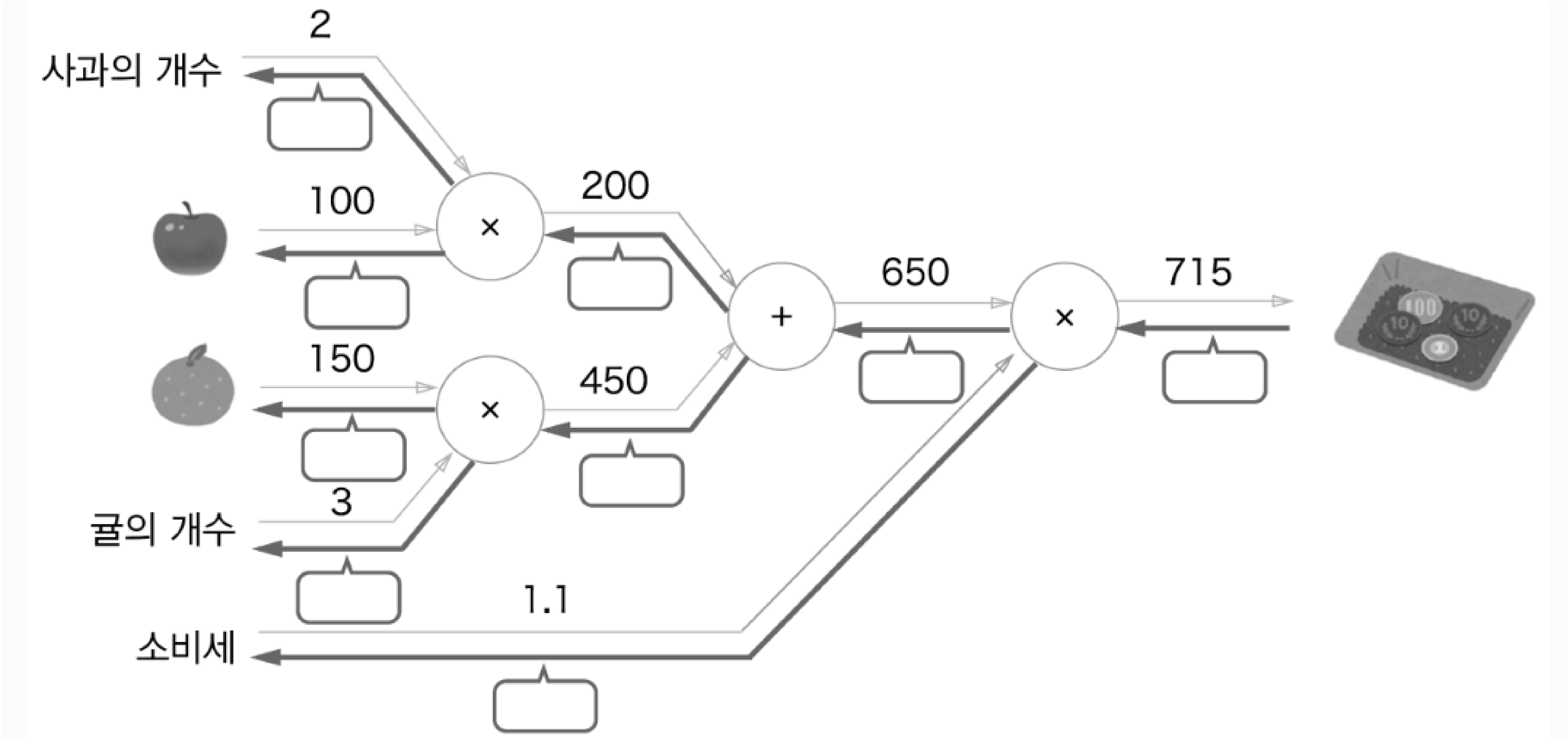
$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x}$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\cancel{\partial t}}\frac{\cancel{\partial t}}{\partial x}$$

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial t}\frac{\partial t}{\partial x} = 2t \times 1 = 2(x + y)$$

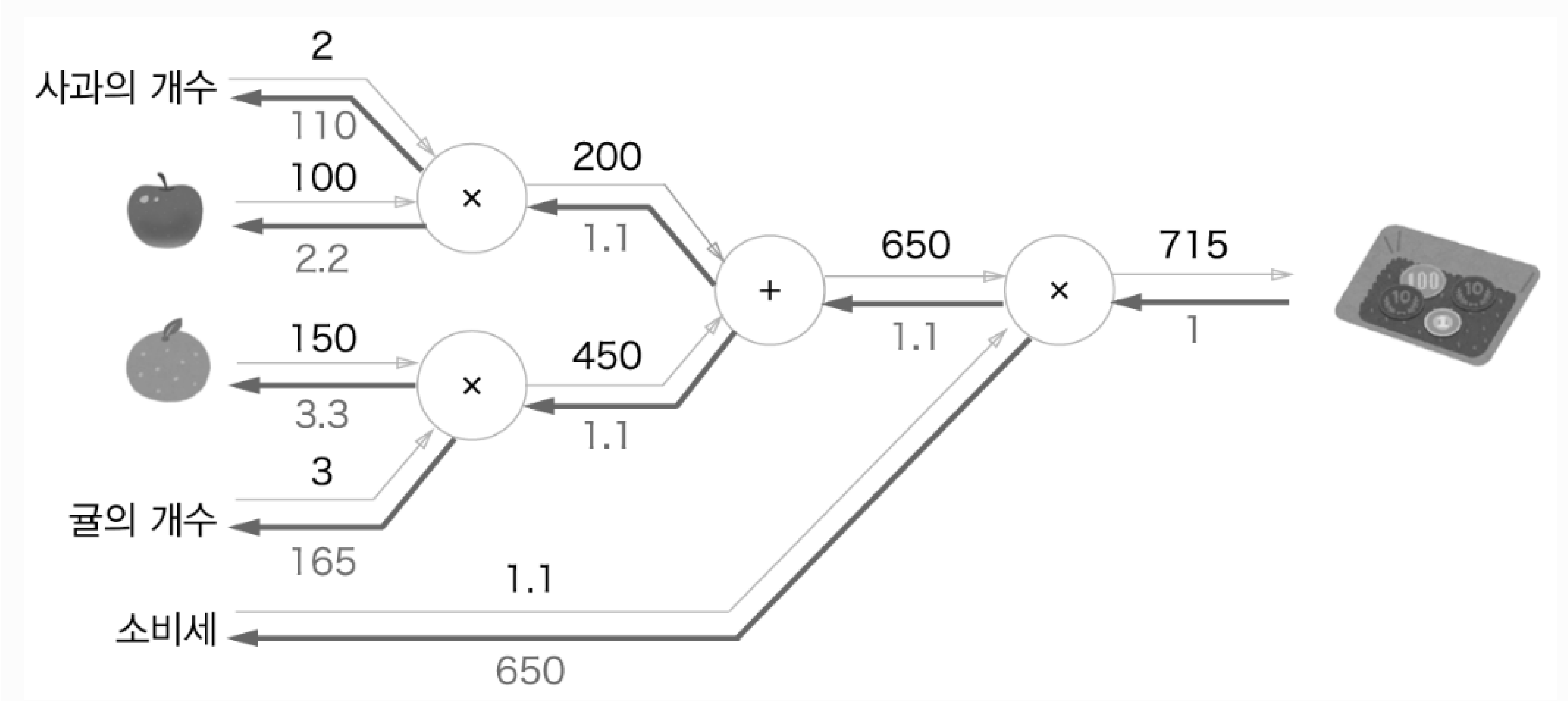
역전파 연습문제

계산그래프2를 역전파 해보자



역전파 연습문제

계산그래프2를 역전파 해보자



MLP 오차역전파 예제

MLP 오차역전파

파란숫자 : 입력값

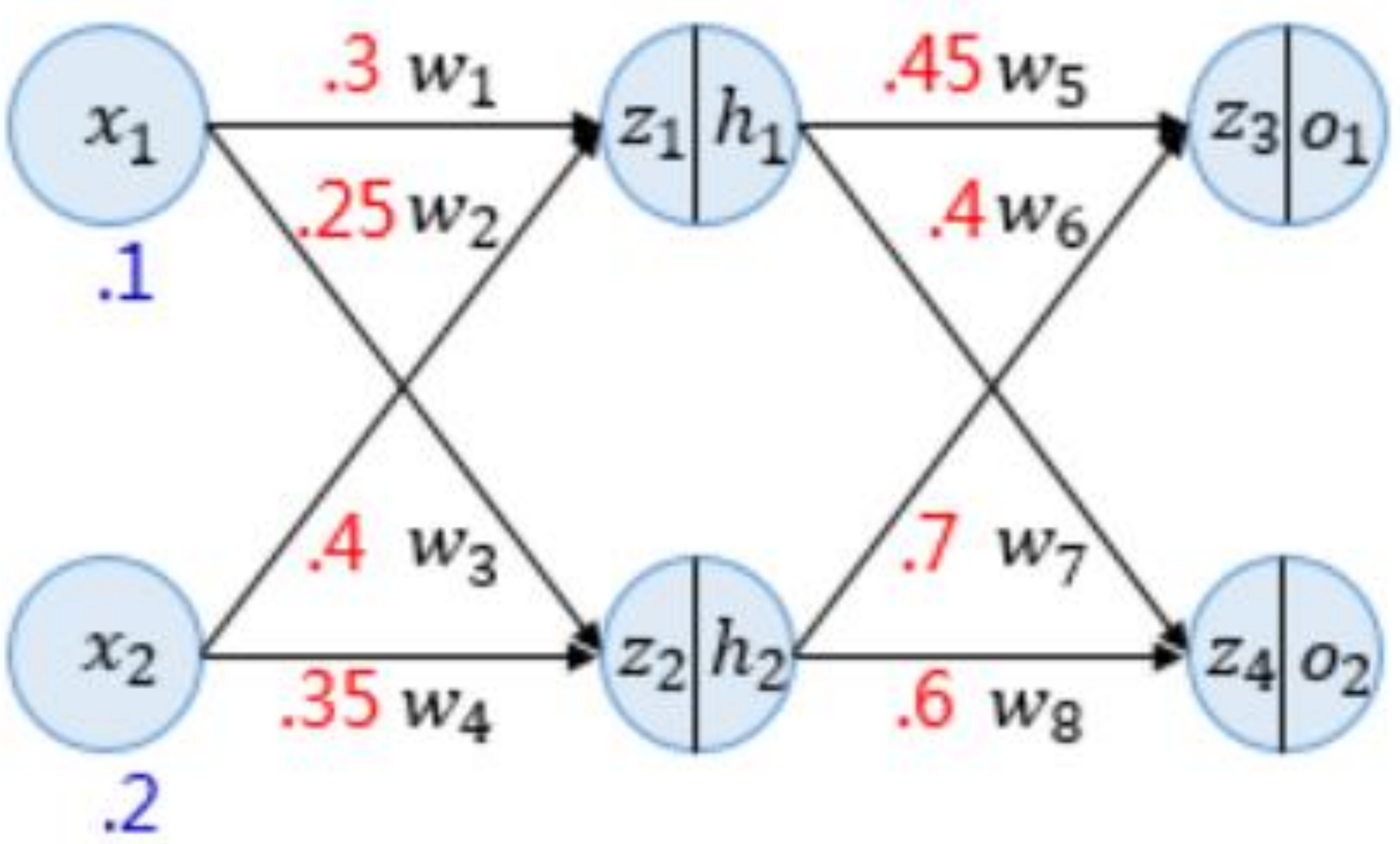
빨간숫자 : weight

h : sigmoid(z)

o : sigmoid(z)

···> sigmoid 미분

Loss function : MSE



MLP 오차역전파 예제

MLP 오차역전파(순전파)

$z_1 = w_1x_1 + w_2x_2 = 0.3\times0.1 + 0.25\times0.2 = 0.08$

$z_2 = w_3x_1 + w_4x_2 = 0.4\times0.1 + 0.35\times0.2 = 0.11$

$h_1 = sigmoid(z_1) = 0.51998934$

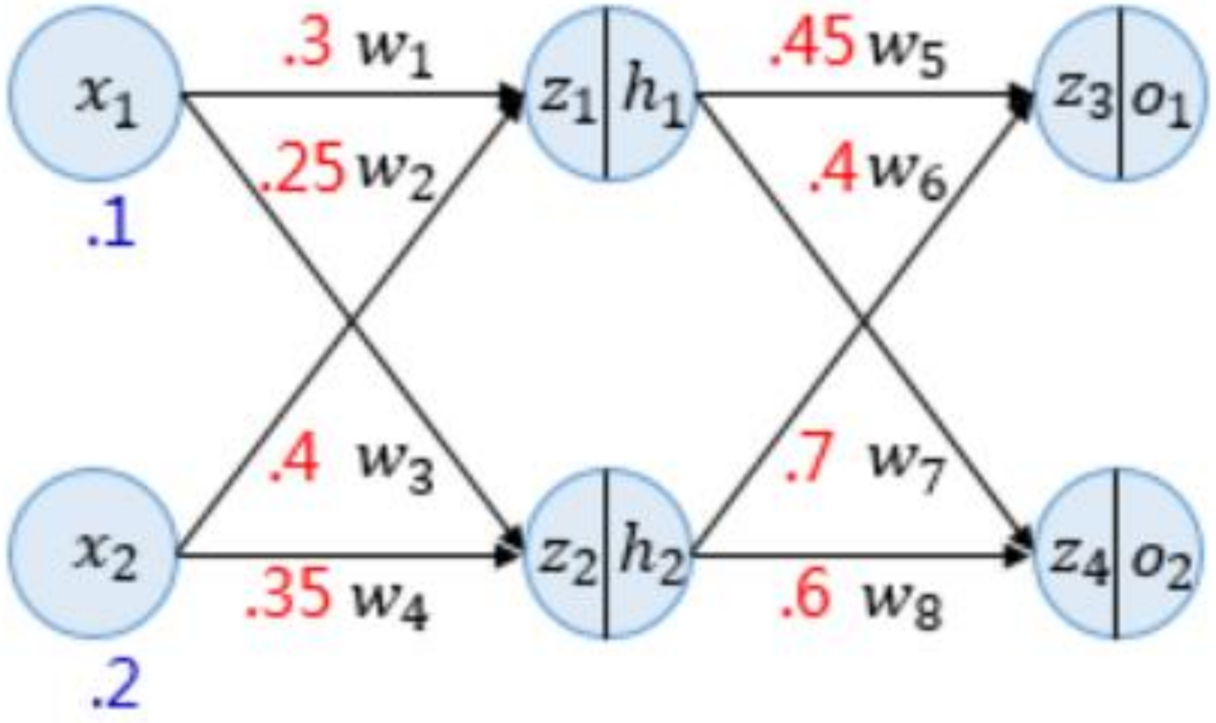
$h_2 = sigmoid(z_2) = 0.52747230$

$z_3 = w_5h_1 + w_6h_2 = 0.45\times h_1 + 0.4\times h_2 = 0.44498412$

$z_4 = w_7h_1 + w_8h_2 = 0.7\times h_1 + 0.6\times h_2 = 0.68047592$

$o_1 = sigmoid(z_3) = 0.60944600$

$o_2 = sigmoid(z_4) = 0.66384491$



실제값 : 0.4

실제값 : 0.6

0.4 – 0.60944600

$$E_{o1} = \frac{1}{2}(target_{o1} - output_{o1})^2 = 0.02193381$$

0.6 – 0.66384491

$$E_{o2} = \frac{1}{2}(target_{o2} - output_{o2})^2 = 0.00203809$$

$$E_{total} = E_{o1} + E_{o2} = 0.02397190$$

오차역전파법

MLP 오차역전파 예제

MLP 오차역전파(역전파1단계)

$\frac{\partial E_{total}}{\partial w_5} = \boxed{\frac{\partial E_{total}}{\partial o_1}} \times \boxed{\frac{\partial o_1}{\partial z_3}} \times \boxed{\frac{\partial z_3}{\partial w_5}}$ ···> 그레디언트 구하기 for 경사 하강법

$E_{total} = \frac{1}{2}(target_{o1} - output_{o1})^2 + \frac{1}{2}(target_{o2} - output_{o2})^2$···> 오차제곱 x 1/2

$\frac{\partial E_{total}}{\partial o_1} = 2 \times \frac{1}{2}(target_{o1} - output_{o1})^{2-1} \times (-1) + 0$

$\frac{\partial E_{total}}{\partial o_1} = -(target_{o1} - output_{o1}) = -(0.4 - 0.60944600) = 0.20944600$

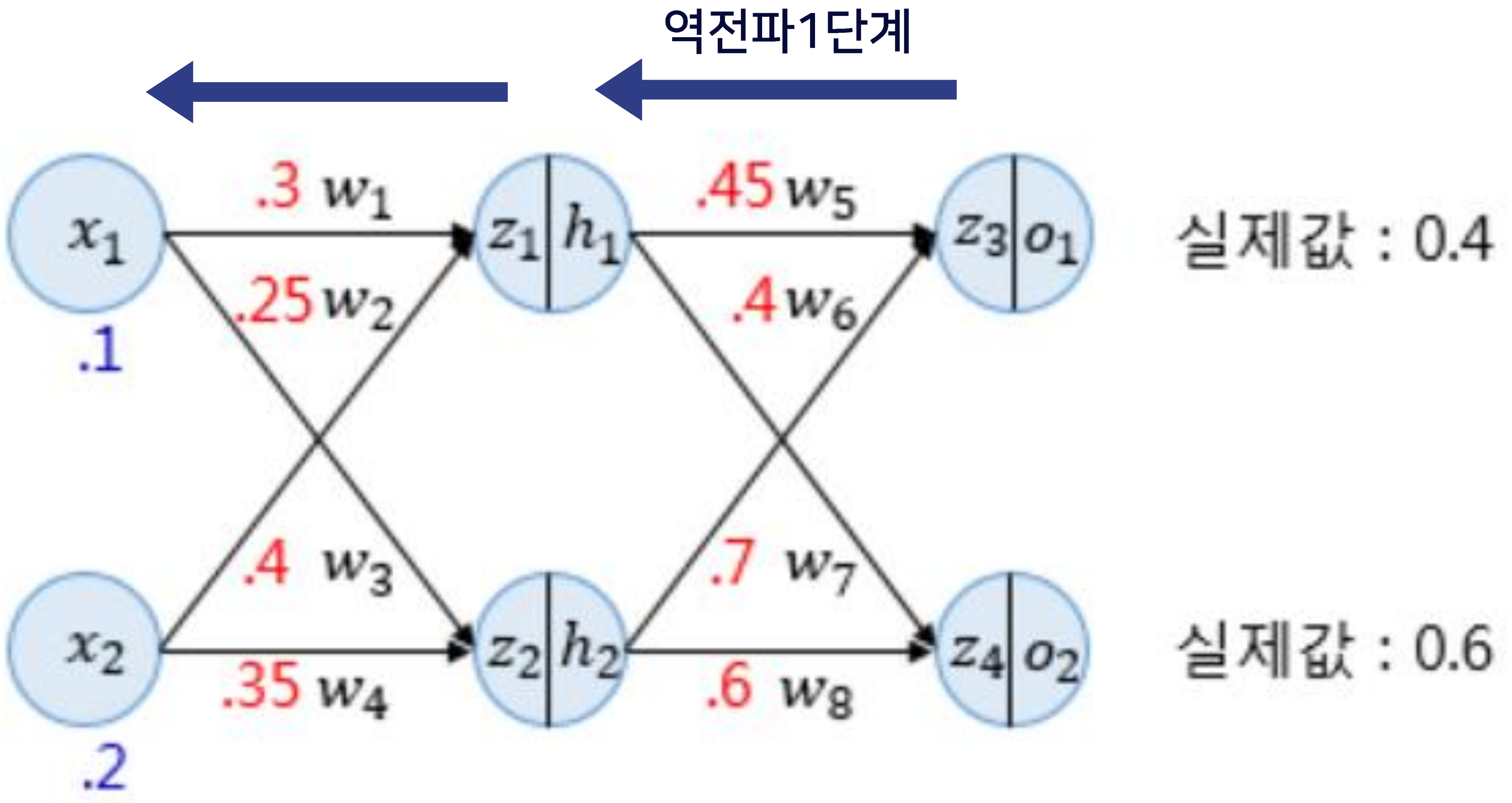
···> 미분 후 값 대입

$\frac{\partial o_1}{\partial z_3} = o_1 \times (1 - o_1) = 0.60944600(1 - 0.60944600) = 0.23802157$ ···> sigmoid 미분

$\frac{\partial z_3}{\partial w_5} = h_1 = 0.51998934$ ···> $z_3 = w_5 h_1 + w_6 h_2$ 을 보면 자명하다.

$\frac{\partial E_{total}}{\partial w_5} = 0.20944600 \times 0.23802157 \times 0.51998934 = 0.02592286$···> 모두 곱하기

$w_5^+ = w_5 - \alpha \boxed{\frac{\partial E_{total}}{\partial w_5}} = 0.45 - 0.5 \times 0.02592286 = 0.43703857$···> 경사하강법으로 weight 갱신 !
(learning_rate = 0.5)



$\frac{\partial E_{total}}{\partial w_6} = \frac{\partial E_{total}}{\partial o_1} \times \frac{\partial o_1}{\partial z_3} \times \frac{\partial z_3}{\partial w_6} \rightarrow w_6^+ = 0.38685205$

$\frac{\partial E_{total}}{\partial w_7} = \frac{\partial E_{total}}{\partial o_2} \times \frac{\partial o_2}{\partial z_4} \times \frac{\partial z_4}{\partial w_7} \rightarrow w_7^+ = 0.69629578$

$\frac{\partial E_{total}}{\partial w_8} = \frac{\partial E_{total}}{\partial o_2} \times \frac{\partial o_2}{\partial z_4} \times \frac{\partial z_4}{\partial w_8} \rightarrow w_8^+ = 0.59624247$

MLP 오차역전파 예제

MLP 오차역전파(역전파2단계)

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial h_1} \times \frac{\partial h_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_1}$$ ···> 그레디언트 구하기 for 경사 하강법

$$\frac{\partial E_{total}}{\partial h_1} = \frac{\partial E_{o1}}{\partial h_1} + \frac{\partial E_{o2}}{\partial h_1}$$ ···> total error에 대한 h_1의 미분을 두개로 분할

$$\frac{\partial E_{o1}}{\partial h_1} = \frac{\partial E_{o1}}{\partial z_3} \times \frac{\partial z_3}{\partial h_1} = \frac{\partial E_{o1}}{\partial o_1} \times \frac{\partial o_1}{\partial z_3} \times \frac{\partial z_3}{\partial h_1}$$ ···> chain rule

$$= -(target_{o1} - output_{o1}) \times o_1 \times (1 - o_1) \times w_5$$ ···> sigmoid 미분

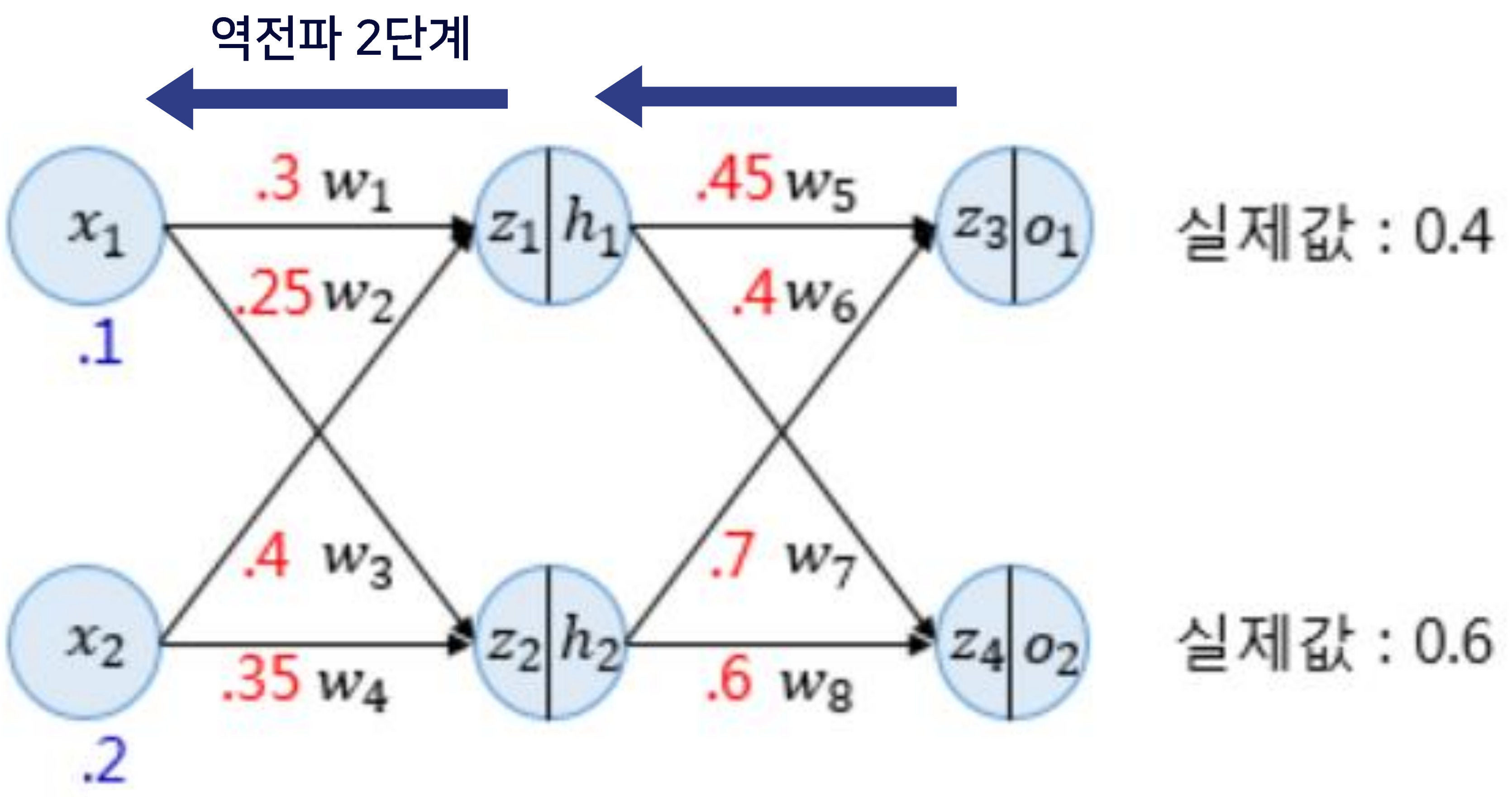
$$= 0.20944600 \times 0.23802157 \times 0.45 = 0.02243370$$ ···> 값 대입

$$\frac{\partial E_{o2}}{\partial h_1} = \frac{\partial E_{o2}}{\partial z_4} \times \frac{\partial z_4}{\partial h_1} = \frac{\partial E_{o2}}{\partial o_2} \times \frac{\partial o_2}{\partial z_4} \times \frac{\partial z_4}{\partial h_1} = 0.00997311$$ ···> h_1과 같은 과정으로 계산

$$\frac{\partial E_{total}}{\partial h_1} = 0.02243370 + 0.00997311 = 0.03240681$$ ···> 분할했던 값을 더한다.

$$\frac{\partial h_1}{\partial z_1} = h_1 \times (1 - h_1) = 0.51998934(1 - 0.51998934) = 0.24960043$$ ···> sigmoid 미분

$$\frac{\partial z_1}{\partial w_1} = x_1 = 0.1$$ ···> $z_1 = w_1 x_1 + w_2 x_2$ 을 보면 자명하다.



$$\frac{\partial E_{total}}{\partial w_1} = 0.03240681 \times 0.24960043 \times 0.1 = 0.00080888$$

$$\frac{\partial E_{total}}{\partial w_2} = \frac{\partial E_{total}}{\partial h_1} \times \frac{\partial h_1}{\partial z_1} \times \frac{\partial z_1}{\partial w_2} \rightarrow w_2^+ = 0.24919112$$

$$\frac{\partial E_{total}}{\partial w_3} = \frac{\partial E_{total}}{\partial h_2} \times \frac{\partial h_2}{\partial z_2} \times \frac{\partial z_2}{\partial w_3} \rightarrow w_3^+ = 0.39964496$$ ···> w_1과 같은 과정으로 계산

$$\frac{\partial E_{total}}{\partial w_4} = \frac{\partial E_{total}}{\partial h_2} \times \frac{\partial h_2}{\partial z_2} \times \frac{\partial z_2}{\partial w_4} \rightarrow w_4^+ = 0.34928991$$

MLP 오차역전파 예제

MLP 오차역전파(순전파)

···> 업데이트된 가중치로 순전파 !

$$z_1 = w_1x_1 + w_2x_2 = 0.29959556 \times 0.1 + 0.24919112 \times 0.2 = 0.07979778$$
$$z_2 = w_3x_1 + w_4x_2 = 0.39964496 \times 0.1 + 0.34928991 \times 0.2 = 0.10982248$$

$$h_1 = sigmoid(z_1) = 0.51993887$$
$$h_2 = sigmoid(z_2) = 0.52742806$$

$$z_3 = w_5h_1 + w_6h_2 = 0.43703857 \times h_1 + 0.38685205 \times h_2 = 0.43126996$$
$$z_4 = w_7h_1 + w_8h_2 = 0.69629578 \times h_1 + 0.59624247 \times h_2 = 0.67650625$$

$$o_1 = sigmoid(z_3) = 0.60617688$$
$$o_2 = sigmoid(z_4) = 0.66295848$$

0.4 - 0.60617688

$$E_{o1} = \frac{1}{2}(target_{o1} - output_{o1})^2 = 0.02125445$$

0.6 - 0.66295848

$$E_{o2} = \frac{1}{2}(target_{o2} - output_{o2})^2 = 0.00198189$$

$$\boxed{E_{total} = E_{o1} + E_{o2} = 0.02323634}$$

0.4 - 0.60944600

$$E_{o1} = \frac{1}{2}(target_{o1} - output_{o1})^2 = 0.02193381$$

0.6 - 0.66384491

$$E_{o2} = \frac{1}{2}(target_{o2} - output_{o2})^2 = 0.00203809$$

$$\boxed{E_{total} = E_{o1} + E_{o2} = 0.02397190}$$

Learning rate = 0.5, epoch 1로 가중치를 업데이트 했더니 무려 오차가 0.00073556 줄었다!!!

코드실습

코드실습

2024_Deep_2주차_실습.ipynb를 열어주세요!

REVIEW

과제

역전파 코드 주석달기

역전파 코드를 직접 돌려보고, 각 코드별 주석을 달아본다.

파라미터에 대한 이해를 한다.

왜 이렇게 돌아가는지를 이해한다!

튜닝을 통해 성능변화 확인하기

간단한 MLP모델에서 간단하게 튜닝 해보기

많이 업데이트되는 가중치와 작게 업데이트되는 가중치의 차이를 이해한다.

각 파라미터가 어떤 영향을 미치는지 확인한다.

REFERENCE

REFERENCE

https://developer.ibm.com/articles/l-neural/

https://medium.com/@chriskevin_80184/feature-maps-ee8e11a71f9e

밑바닥부터 시작하는 딥러닝1 – 사이토 고키

https://wikidocs.net/37406

국민대학교 ai빅데이터융합경영학과 조윤호 교수님 – 머신러닝

https://medium.com/swlh/the-math-of-machine-learning-i-gradient-descent-with-univariate-linear-regression-2afbfb556131

2024 D&A

THANK YOU

2024. 03. 21

Deep Session 2차시