오차함수

오차함수 (Error Function)

- 오차는 학습 목표 값과 실제 값 간의 차이
- ***평균 제곱 오차***와 ***교차 엔트로피 오차*** 사용

실제 결과 값	목표 값	오차 (목표 값 – 실제 값)	오차 |목표 값 – 실제 값|	오차 (목표 값 – 실제 값)^2
0.4	0.5	0.1	0.1	0.01
0.8	0.7	-0.1	0.1	0.01
1.0	1.0	0	0	0
합		**0**	**0.2**	**0.02**

제곱오차

오차함수 (Error Function)

- ***평균 제곱 오차***
  - Mean Squared Error (MSE)

$$E = \frac{1}{N}\sum_{k}(y_k - t_k)^2$$

y_k → 신경망의 출력

t_k → 정답 레이블

k → 데이터의 차원 수

softmax

```
>>> y = [0.1, 0.05, 0.6, 0.0, 0.05, 0.1, 0.0, 0.1, 0.0, 0.0]
>>> t = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
```

one-hot encoding

```
def mean_squared_error(n, y, t):
    return (1/n) * np.sum( ( y – t ) ** 2)
```

오차함수 (Error Function)

- ***평균 제곱 오차***

```
>>> t = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]


# ex1)
>>> y = [0.1, 0.05, 0.6, 0.0, 0.05, 0.1, 0.0, 0.1, 0.0, 0.0]
>>> mean_suared_error(10, np.array(y), np.array(t))
0.019500000000000007
>>
# ex2)
>>> y = [0.1, 0.05, 0.1, 0.0, 0.05, 0.1, 0.0, 0.6, 0.0, 0.0]
>>> mean_suared_error(10, np.array(y), np.array(t))
0.11950000000000001
```

ex1)이 오차 함수 쪽 출력이 작음 (오차 적음)

→ 정답에 가까움

오차함수 (Error Function)

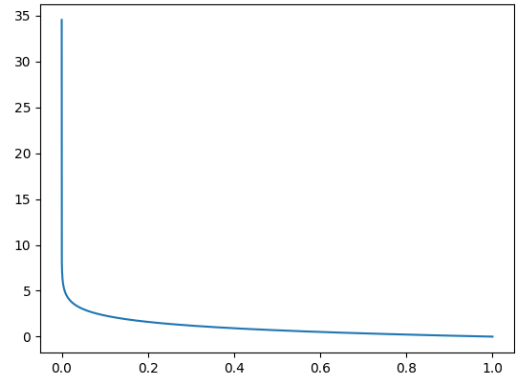
- ***교차 엔트로피 오차***
  - Cross Entropy Error (CEE)
  - One-hot encoding만 적용

$$E = -\sum_{k} t_k(\log y_k)$$

log → 밑이 e인 자연로그($\log_e$)

y_k → 신경망의 출력

t_k → 정답 레이블



자연로그 $y = -\log_x$ 의 그래프

 # ex1) 신경 출력망이 0.6 → 교차 엔트로피 오차 –log0.6 = 0.51

 # ex2) 신경 출력망이 0.1 → 교차 엔트로피 오차 –log0.1 = 2.30

```
def cross_entropy_error(y, t):
    delta = 1e-7
    return –np.sum ( t * np.log ( y + delta ) )
```

오차함수 (Error Function)

- ***교차 엔트로피 오차***

```
>>> t = [0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
# ex1)
>>> y = [0.1, 0.05, 0.6, 0.0, 0.05, 0.1, 0.0, 0.1, 0.0, 0.0]
>>> cross_entropy_error(np.array(y), np.array(t))
0.51082545709933802
>>
# ex2)
>>> y = [0.1, 0.05, 0.1, 0.0, 0.05, 0.1, 0.0, 0.6, 0.0, 0.0]
>>> cross_entropy_error(np.array(y), np.array(t))
2.3025840929945458
```

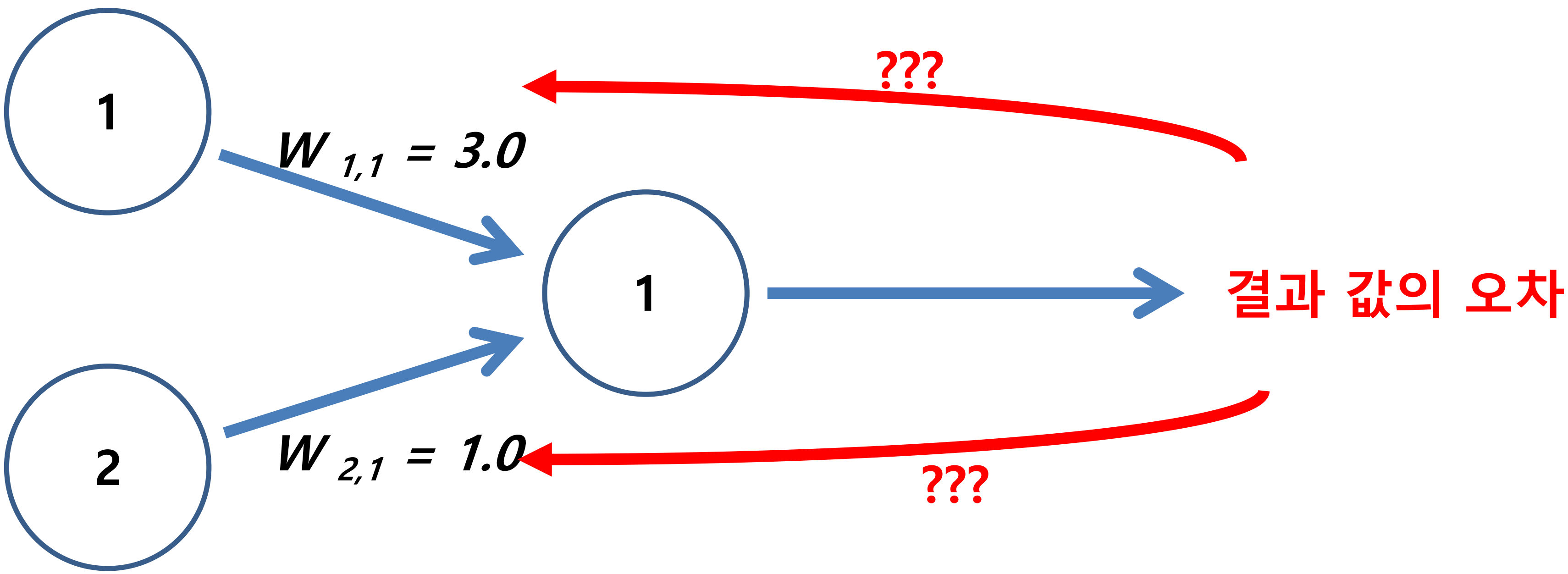
ex1) 신경 출력망이 0.6이면 오차 ***0.51***

→ 정답에 가까움

ex2) 신경 출력망이 0.1이면 오차 ***2.3***

가중치 학습

- 여러 개의 노드가 결과 값과 오차에 영향을 주는 경우에는 가중치를 어떻게 업데이트 해야 할까?



오차를 어떻게 배분해야 할까?

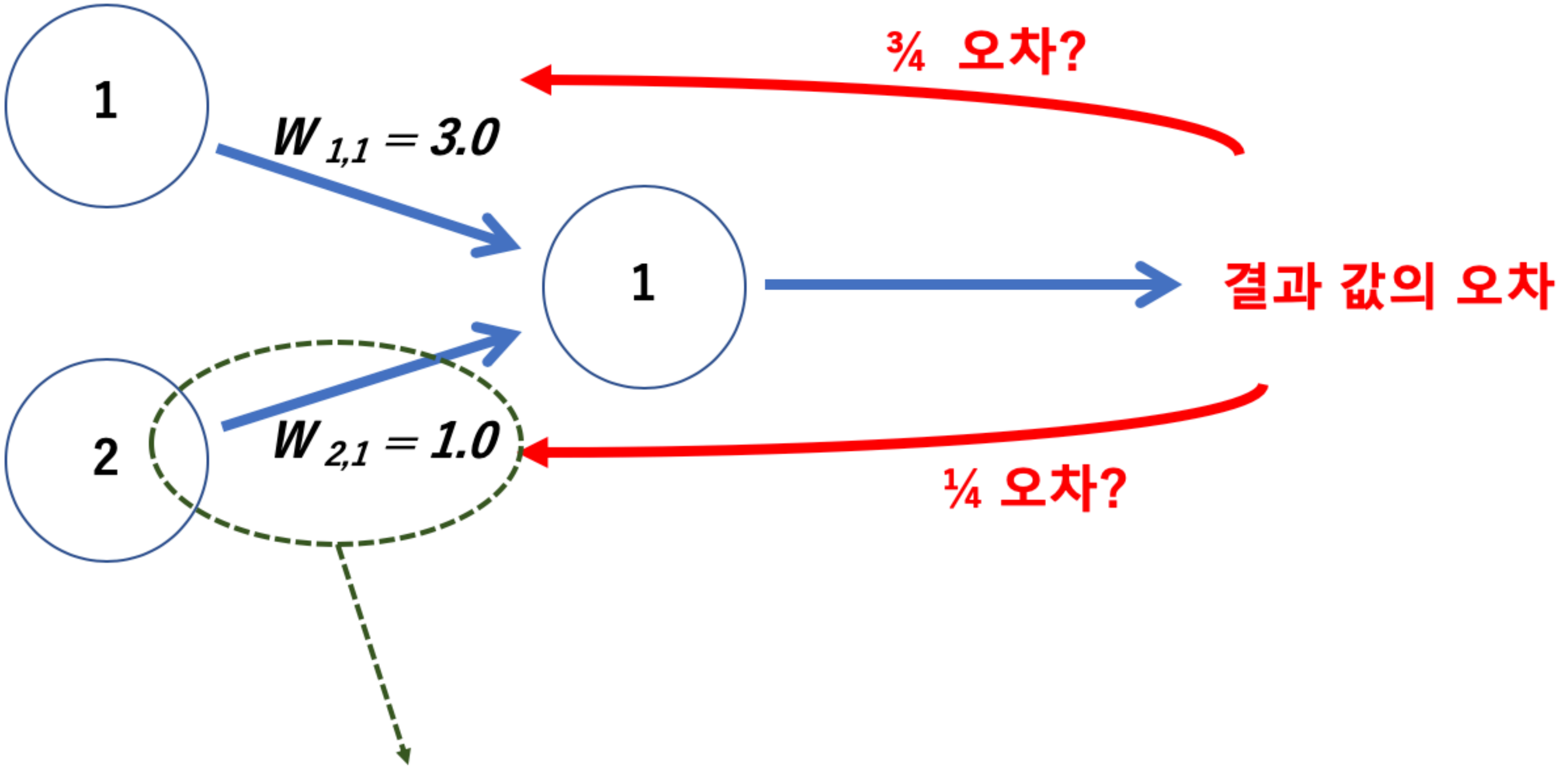
가중치 초기화

- 가중치 초기화는 대부분 균등분포 $U(-0.5, 0.5)$의 난수를 사용. 그러나 [Duda 등 2001]은 균등분포 $U(-1/\sqrt{m_l}, 1/\sqrt{m_l})$ 의 난수를 사용하여 l번째 층과 $l+1$번째 층 사이의 연결강도인 가중치를 초기화하는 것을 추천.
  여기서 m_l은 l번째 층의 노드 개수이고, 0번째 층은 입력층을 나타냄. 이 결과는 각 노드의 입력이 정규분포를 따른다는 가정에 기반을 두고 시그모이드 함수를 사용할 때 최적의 학습을 유도하기 위한 것.

표 2-1 활성함수를 고려한 초기화 기법

활성함수	균등분포 $U(-r, r)$	정규분포 $N(0, \sigma^2)$
시그모이드	$r = \sqrt{\frac{6}{m_l + m_{l+1}}}$	$\sigma = \sqrt{\frac{2}{m_l + m_{l+1}}}$
쌍곡탄젠트	$r = 4\sqrt{\frac{6}{m_l + m_{l+1}}}$	$\sigma = 4\sqrt{\frac{2}{m_l + m_{l+1}}}$
ReLU	$r = \sqrt{2}\sqrt{\frac{6}{m_l + m_{l+1}}}$	$\sigma = \sqrt{2}\sqrt{\frac{2}{m_l + m_{l+1}}}$

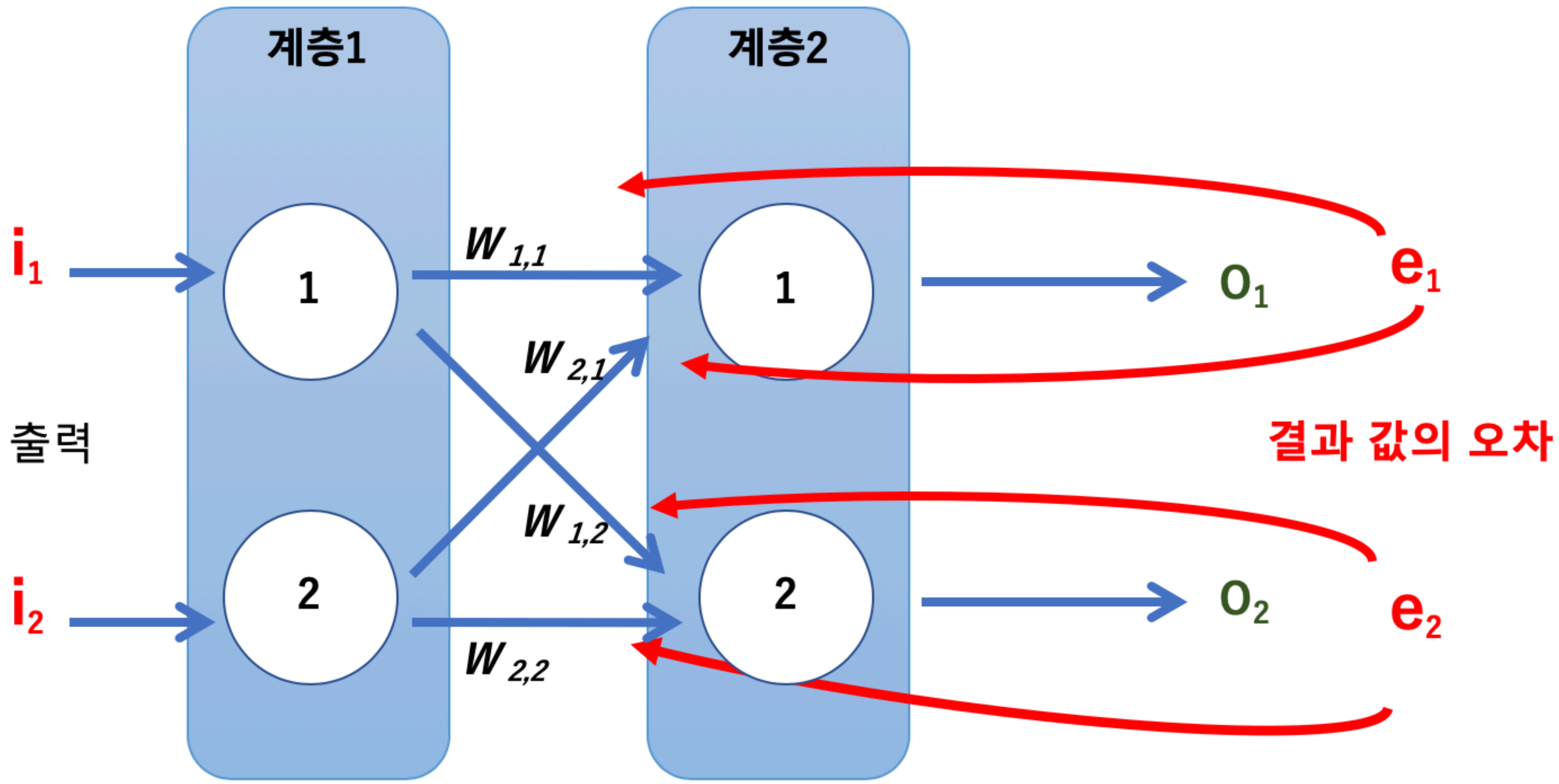
가중치 학습



- 가중치는
  - 하나의 계층에서 다음 계층으로 전파하는 데 사용
  - 오차를 하나의 계층에서 직전 계층으로, 역으로 전파하는 데 사용

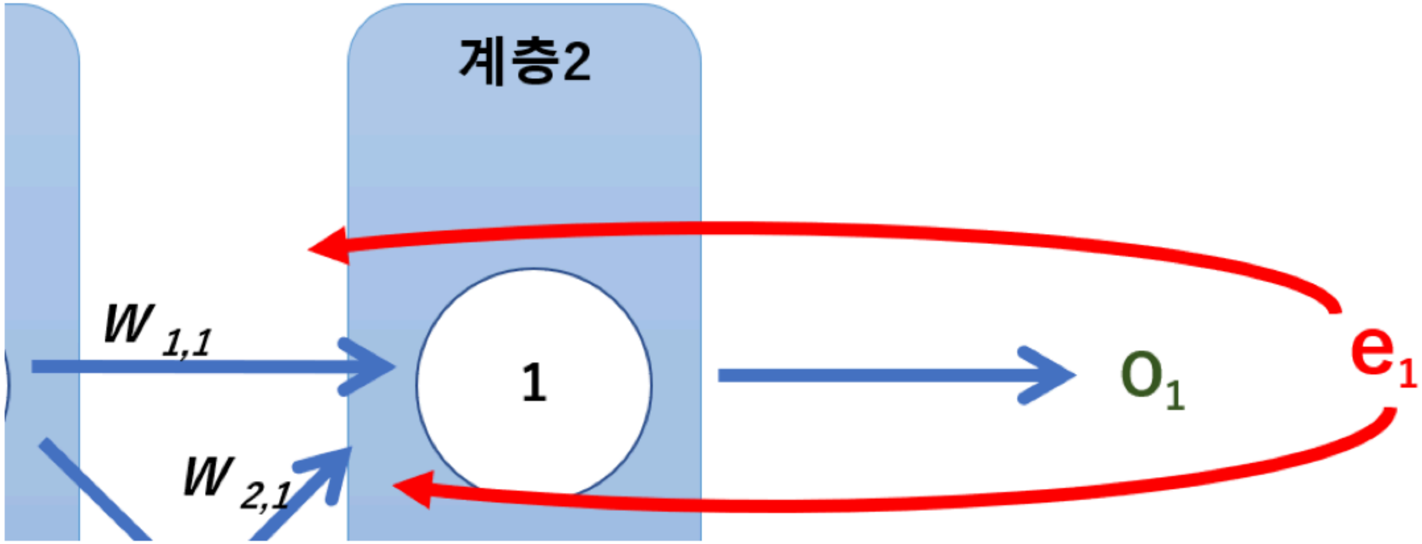
역전파(backpropagation)

가중치 학습



$e_1 = t_1 - o_1$

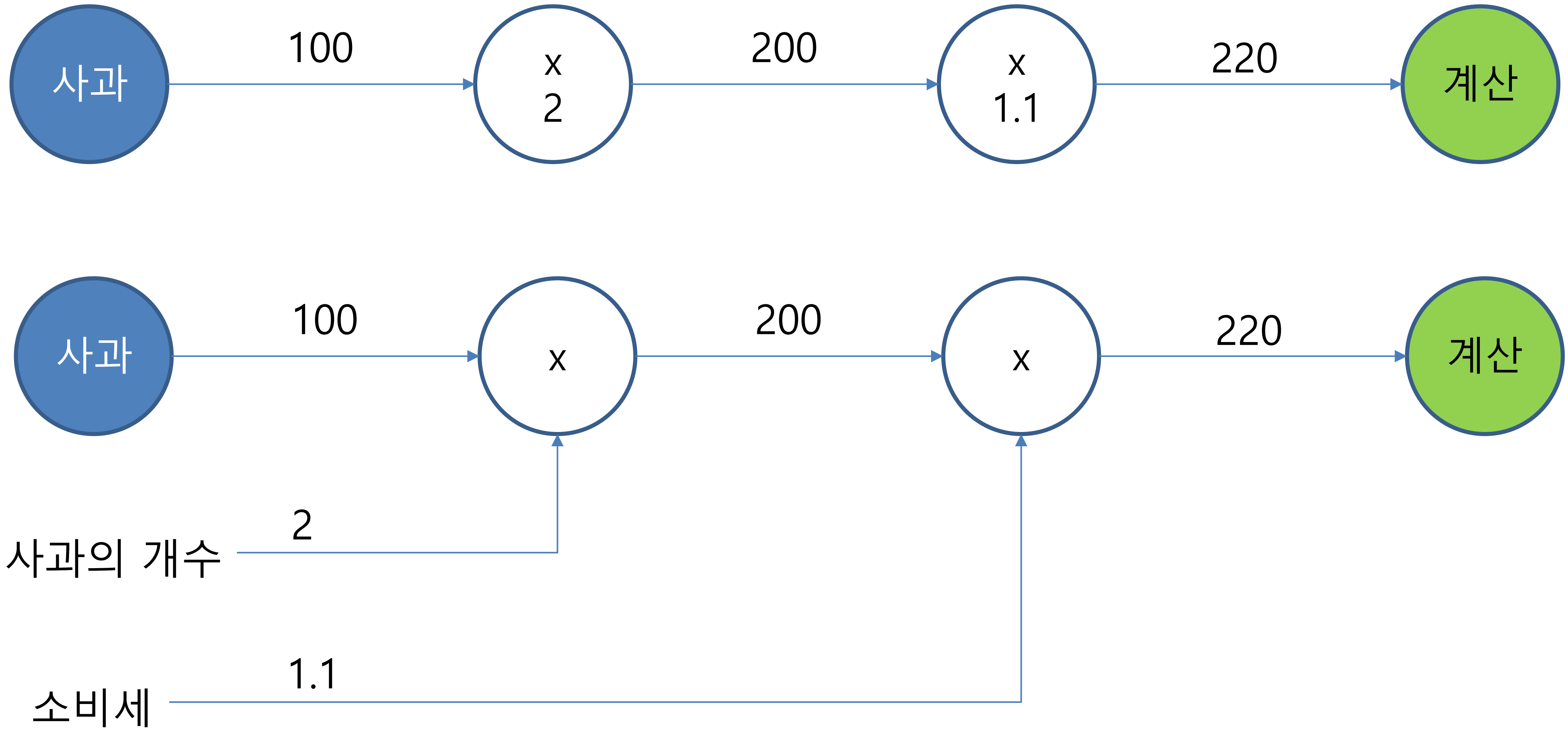
가중치 학습



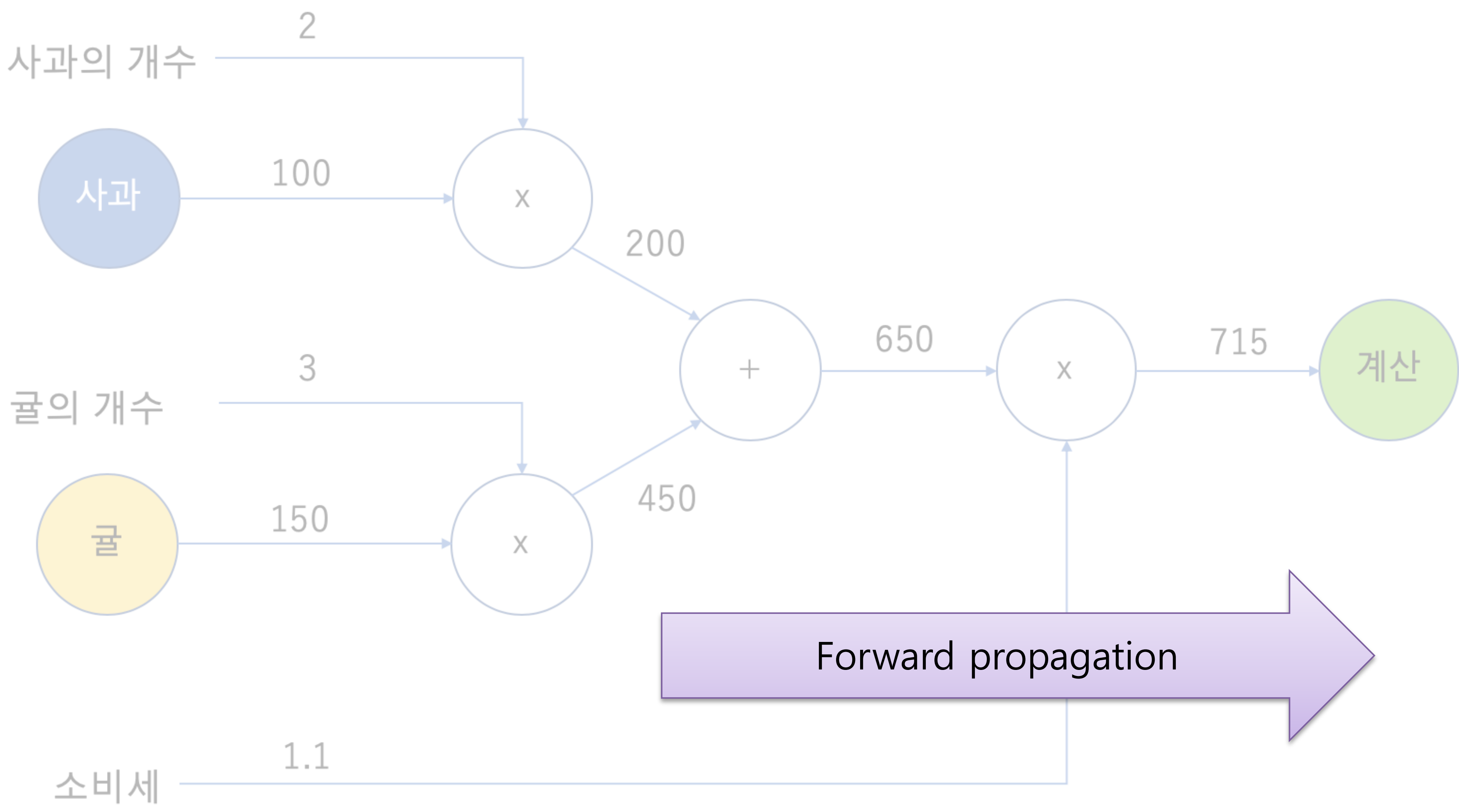
$$\frac{w_{11}}{w_{11} + w_{21}}$$

$$\frac{w_{21}}{w_{21} + w_{22}}$$

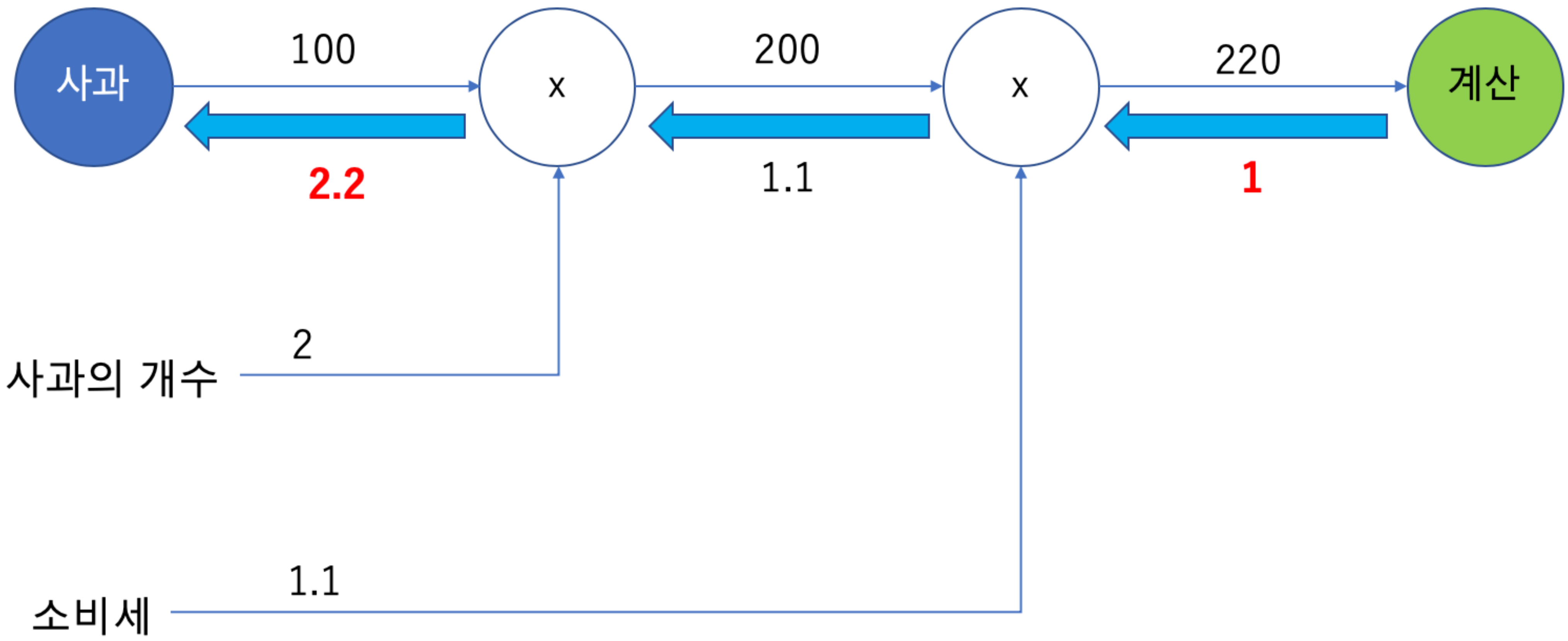
오차역전법(backward propagation)



오차역전법(backward propagation)

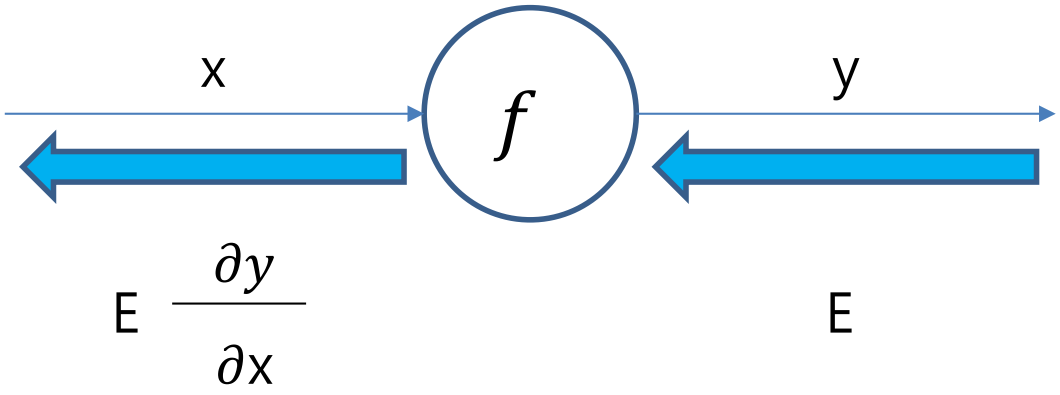


오차역전법(backward propagation)



오차역전법(backward propagation)

- 연쇄법칙 (Chain Rule)



- 연쇄법칙 → 합성함수

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

    $z = t^2$

    $t = x + y$

- *합성함수의 미분은 합성함수를 구성하는 각 함수의 미분의 곱*

오차역전법(backward propagation)

- 합성함수의 미분

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

$$\frac{\partial z}{\partial t} = 2t$$

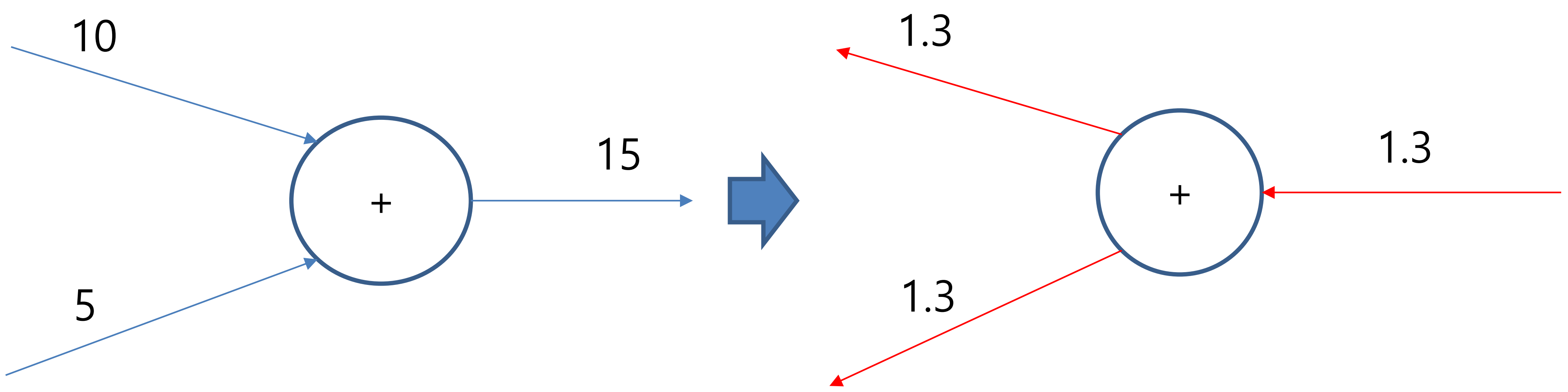
$$\frac{\partial t}{\partial x} = 1$$

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

오차역전법(backward propagation)

- 덧셈 노드의 역전파 예

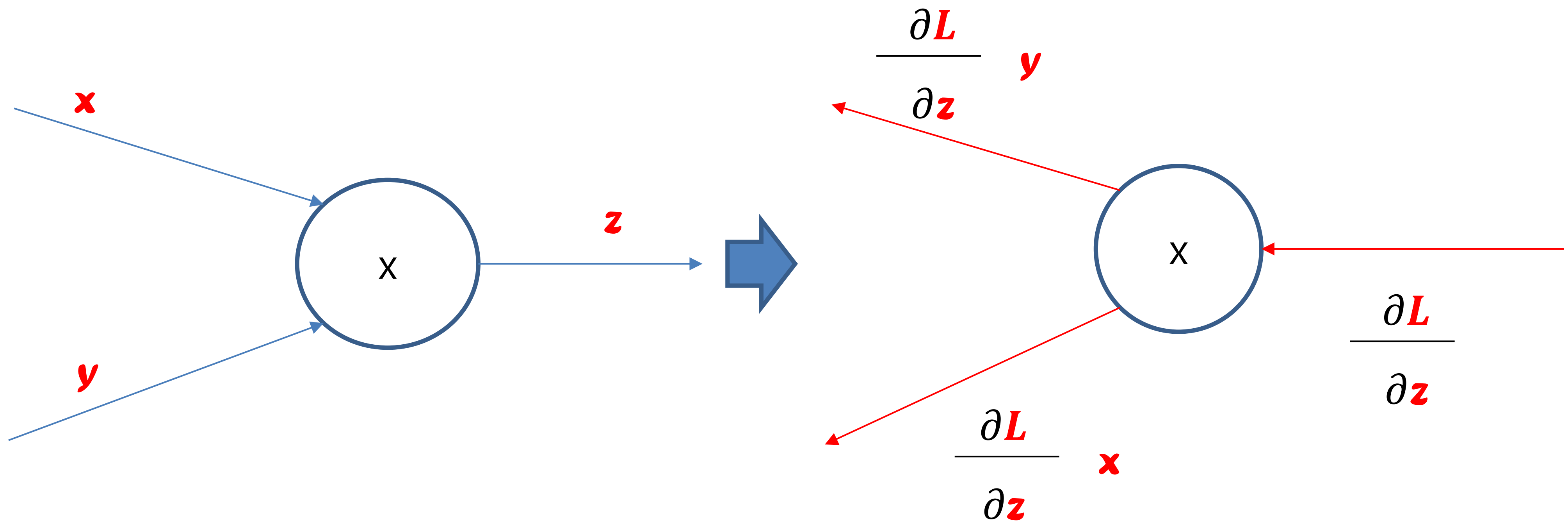
$z = x + y$의 미분은? $\frac{\partial z}{\partial x} = 1$ $\frac{\partial z}{\partial y} = 1$



오차역전법(backward propagation)

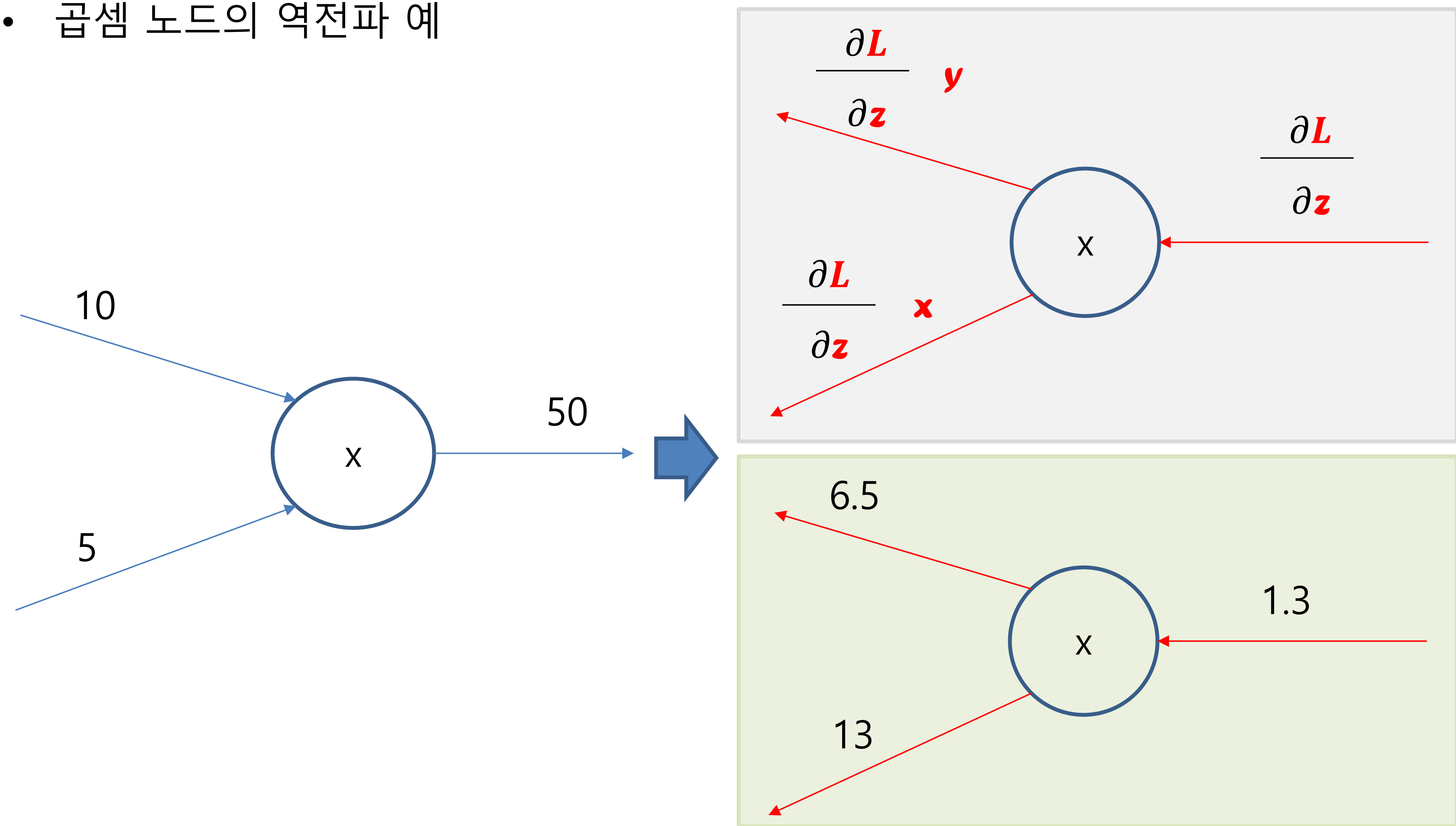
- 곱셈 노드의 역전파 예

$z = xy$ 의 미분은? $\qquad \dfrac{\partial z}{\partial x} = y \qquad \dfrac{\partial z}{\partial y} = x$

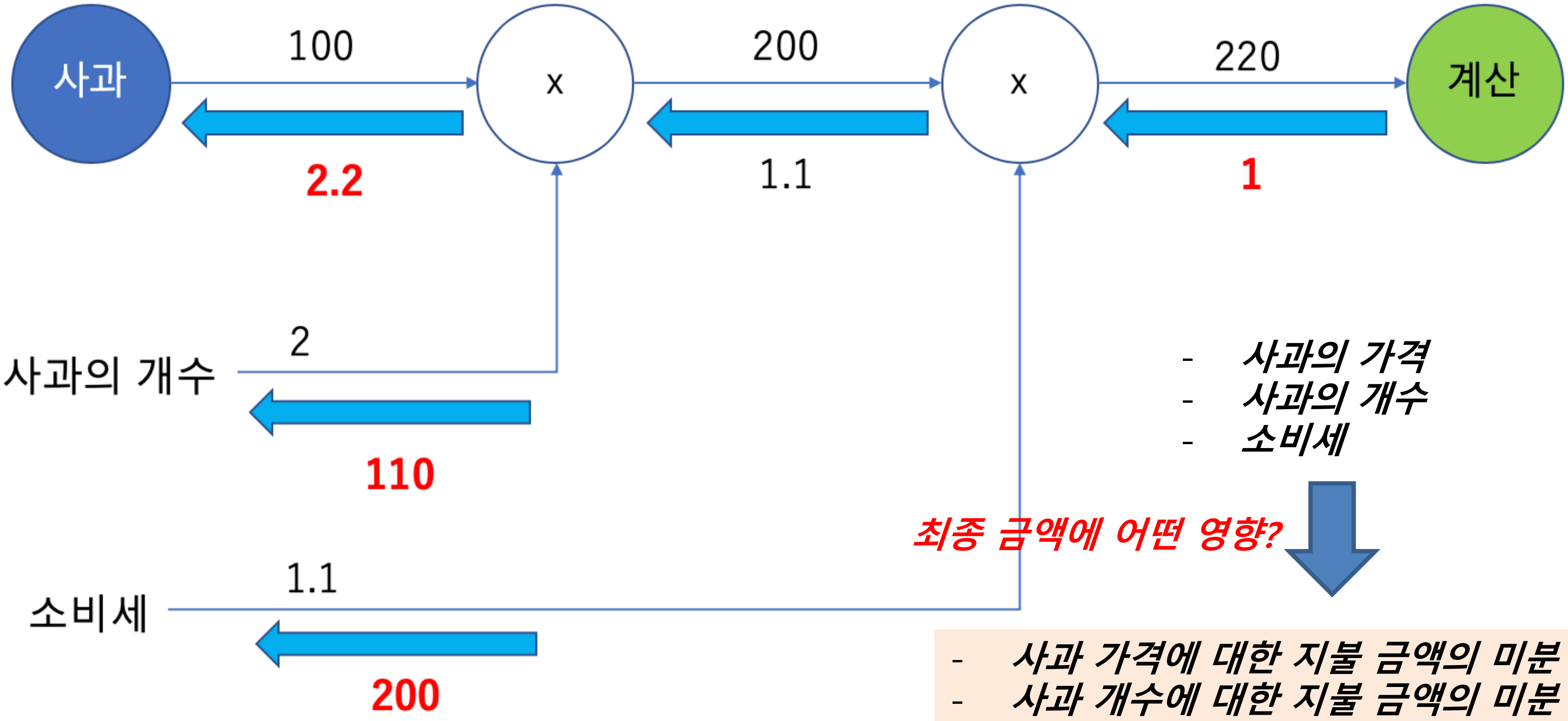


오차역전법(backward propagation)

- 곱셈 노드의 역전파 예



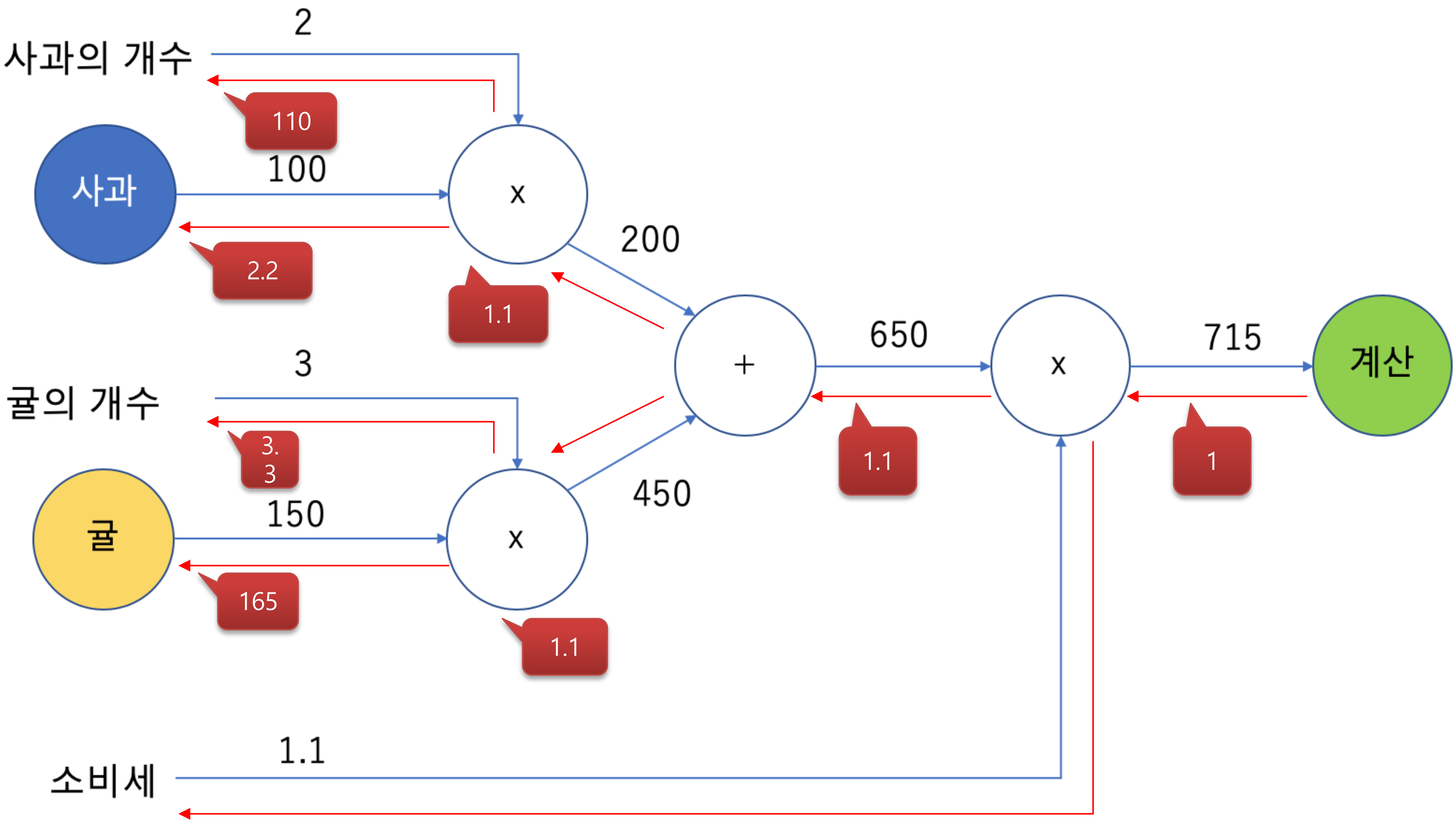
오차역전법(backward propagation)



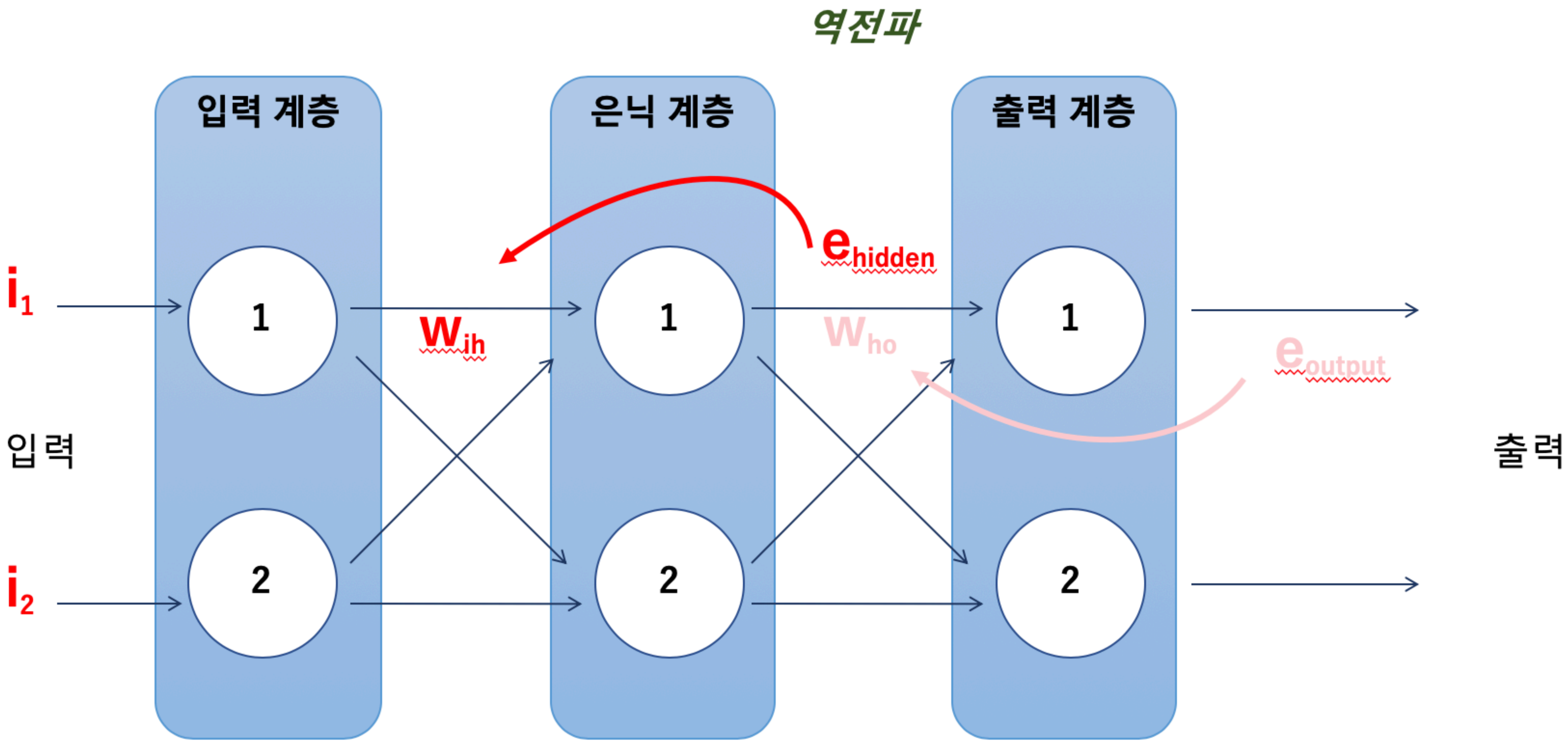
- *사과의 가격*
- *사과의 개수*
- *소비세*

- *사과 가격에 대한 지불 금액의 미분*
- *사과 개수에 대한 지불 금액의 미분*
- *소비세에 대한 지불 금액의 미분*

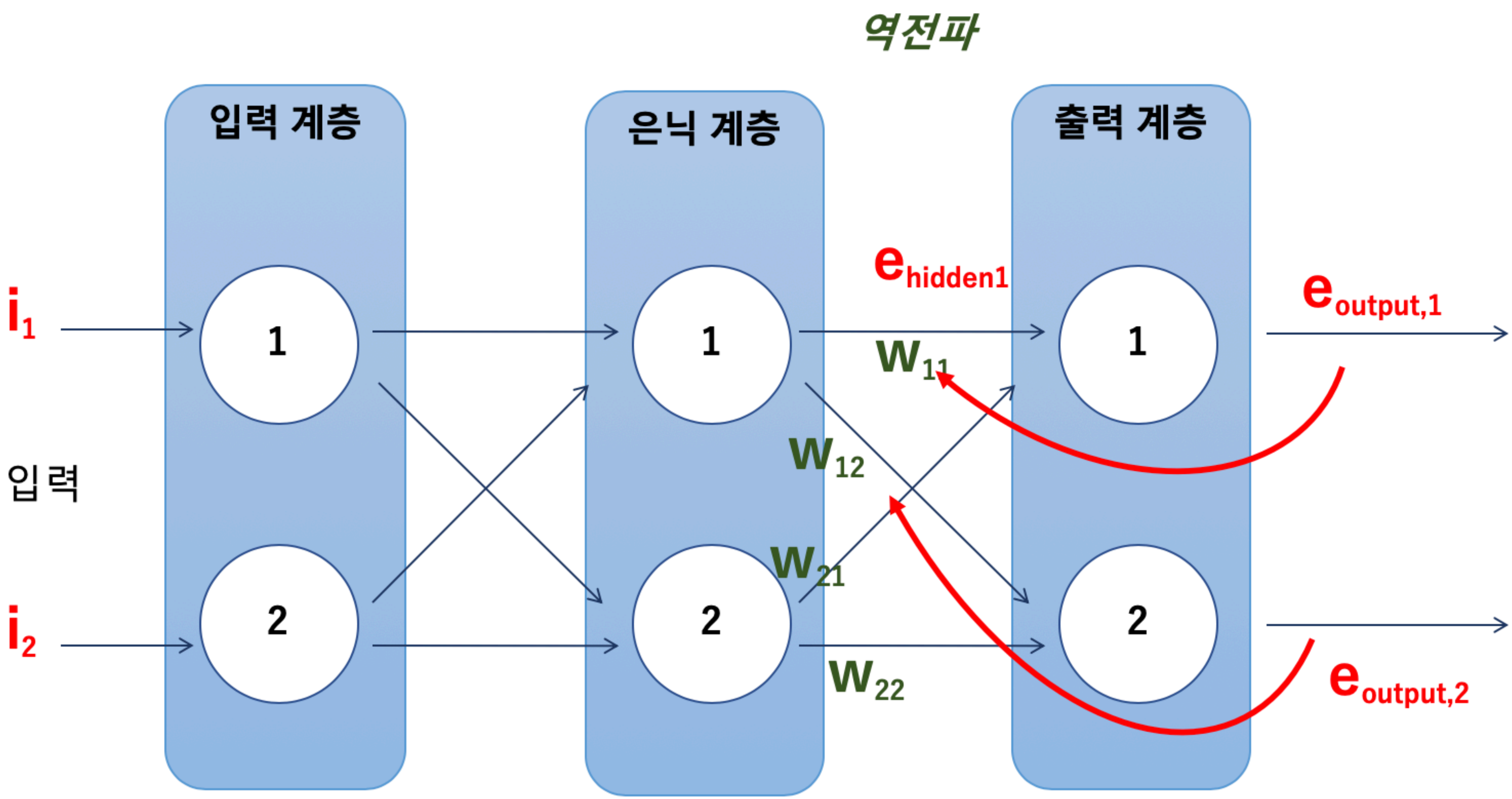
오차역전법(backward propagation)



오차역전법(backward propagation)



오차역전법(backward propagation)



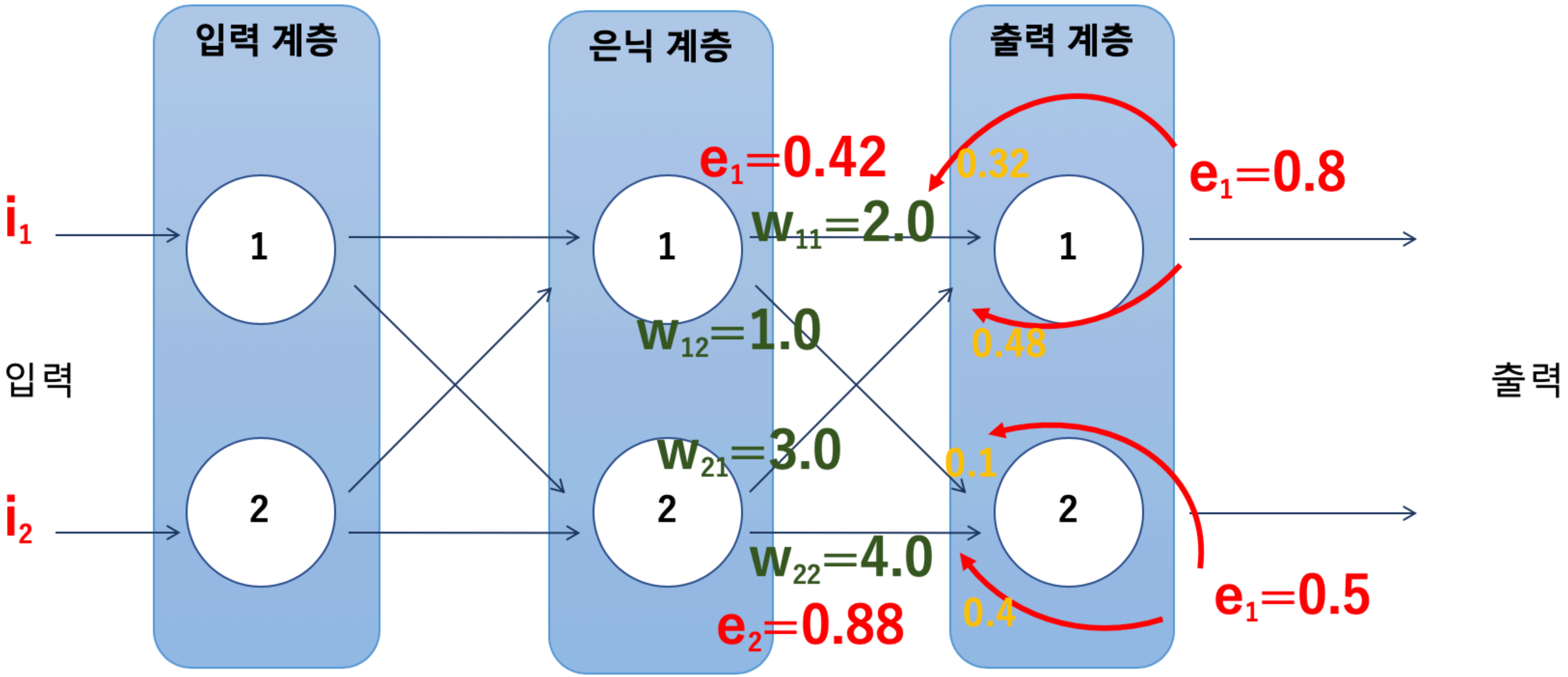
오차역전법(backward propagation)

- 은닉 계층의 첫번째 노드의 오차
  - 이 은닉 계층에서 다음 계층으로 연결되는 모든 연결 노드에 있는 나뉜 **오차들의 합**

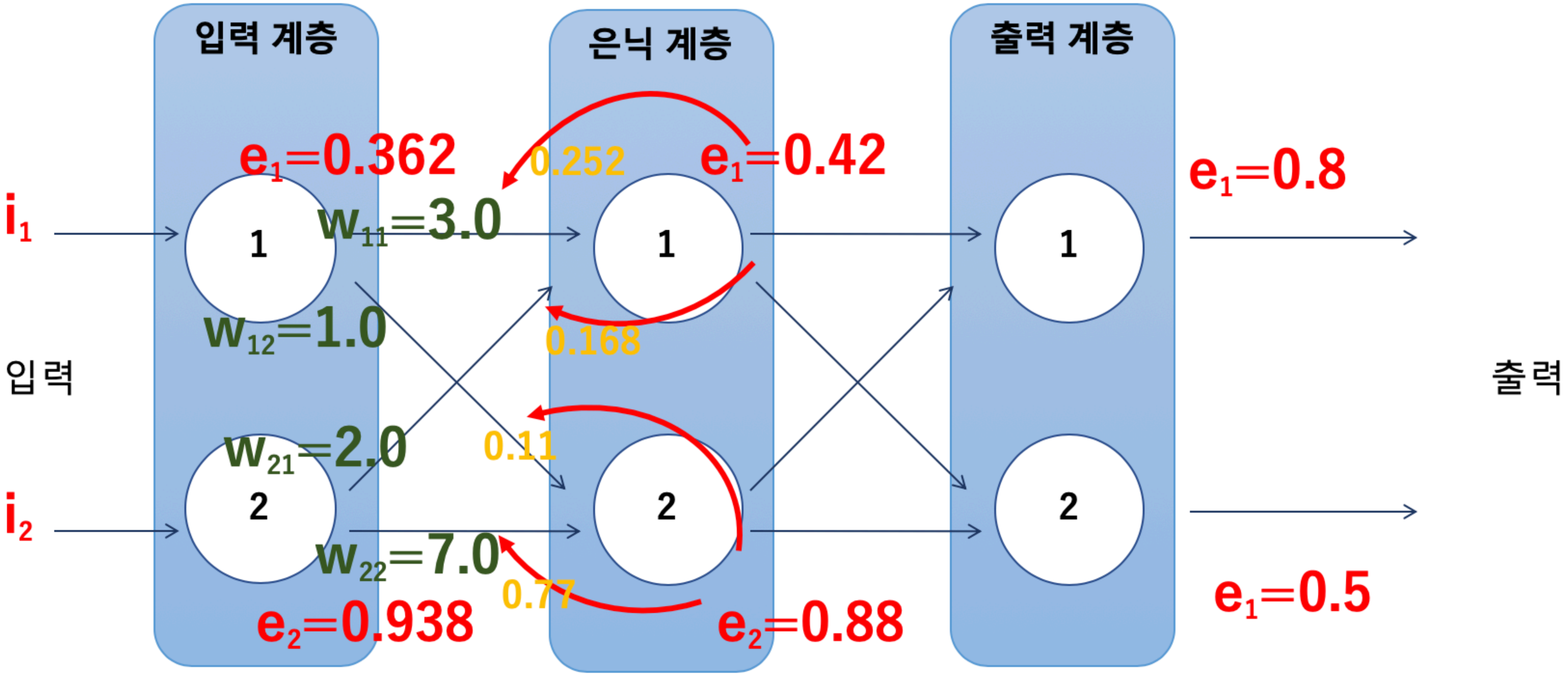
$e_{hidden,1}$ = 연결 노드 w_{11}, w_{21} 로 나뉘어 전달되는 오차의 합

$$= e_{output,1} * \frac{w_{11}}{w_{11} + w_{21}} + e_{output,2} * \frac{w_{12}}{w_{12} + w_{22}}$$

오차역전법(backward propagation)



입력 계층의 오차


입력 계층
은닉 계층
출력 계층
i_1
i_2
입력
출력
1
2
1
2
1
2
$e_1=0.362$
$e_2=0.938$
$w_{11}=3.0$
$w_{12}=1.0$
$w_{21}=2.0$
$w_{22}=7.0$
0.252
0.168
0.11
0.77
$e_1=0.42$
$e_2=0.88$
$e_1=0.8$
$e_1=0.5$

행렬곱을 이용한 오차의 역전파

$$error_{output} = \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$$

$$e_{hidden} = \begin{pmatrix} \frac{w_{11}}{w_{11}+w_{21}} & \frac{w_{12}}{w_{12}+w_{22}} \\ \frac{w_{21}}{w_{21}+w_{11}} & \frac{w_{22}}{w_{22}+w_{12}} \end{pmatrix} \cdot \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$$

$$error_{hidden} = w^T_{hidden\_output} \cdot error_{output}$$

$$\boxed{e_{hidden} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \cdot \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}}$$