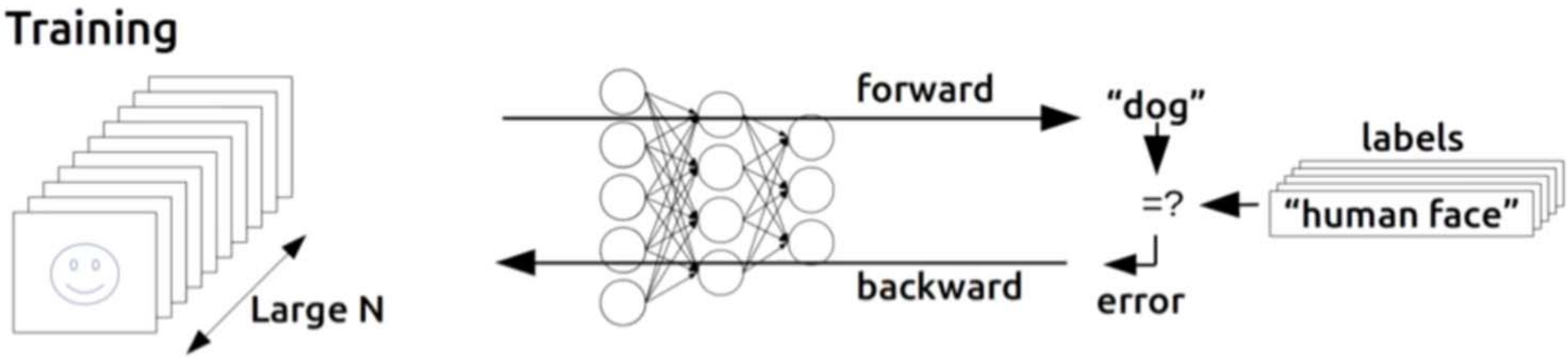
Backpropagation

오차 역전파의 동작원리

김 대 환
2022.

오차 역전파(Backpropagation)

Backpropagation
(1974, 1982 by Paul Werbos, 1986 by Hinton)



오차 역전파(Backpropagation)

- **오차 역전파 (Backpropagation)**
  - 다층 퍼셉트론 학습에 사용되는 통계적 기법
  - 다층 퍼셉트론의 형태는 입력층 - 은닉층 - 은닉층 - ... - 출력층으로 구성되며, 각 층은 서로 교차되는 가중치 값으로 연결
  - 동일 입력층에 대해 원하는 값이 출력되도록 개개의 가중치를 조정하는 방법
- **개요**
  - 신경망을 통해 도달한 결과 값은 오차 (Error)를 포함하며, 최종 오차는 여러 계산 과정을 거치며 합산된 값
  - 각 과정마다 발생한 오차를 관찰해서 계산과정을 수정
  - 속도는 느리지만 안정적인 결과를 얻을 수 있는 장점이 있어 기계 학습에 널리 사용
- **오차 역전파에 의한 가중치 계산**
  - 수식이나 그래프를 통해 계산 과정을 이해할 수 있다

오차 역전파

연쇄 법칙 (Chain Rule)

- 합성함수의 미분에 대한 법칙
  - ✓ 합성함수를 구성하는 각 함수의 미분을 곱으로 표현
  - ✓ 다음처럼 **z**를 **x**로 미분하는 것은 **z**를 **t**로 미분한 것과 **t**를 **x**로 미분한 것의 곱을 나타낼 수 있다

$$z = t^2$$

$$t = x + y$$

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

- 위 식에 연쇄법칙을 적용하여 아래와 같이 각각의 국소적**(local)** 미분 을 구한 뒤 **(편미분)**, 곱한다

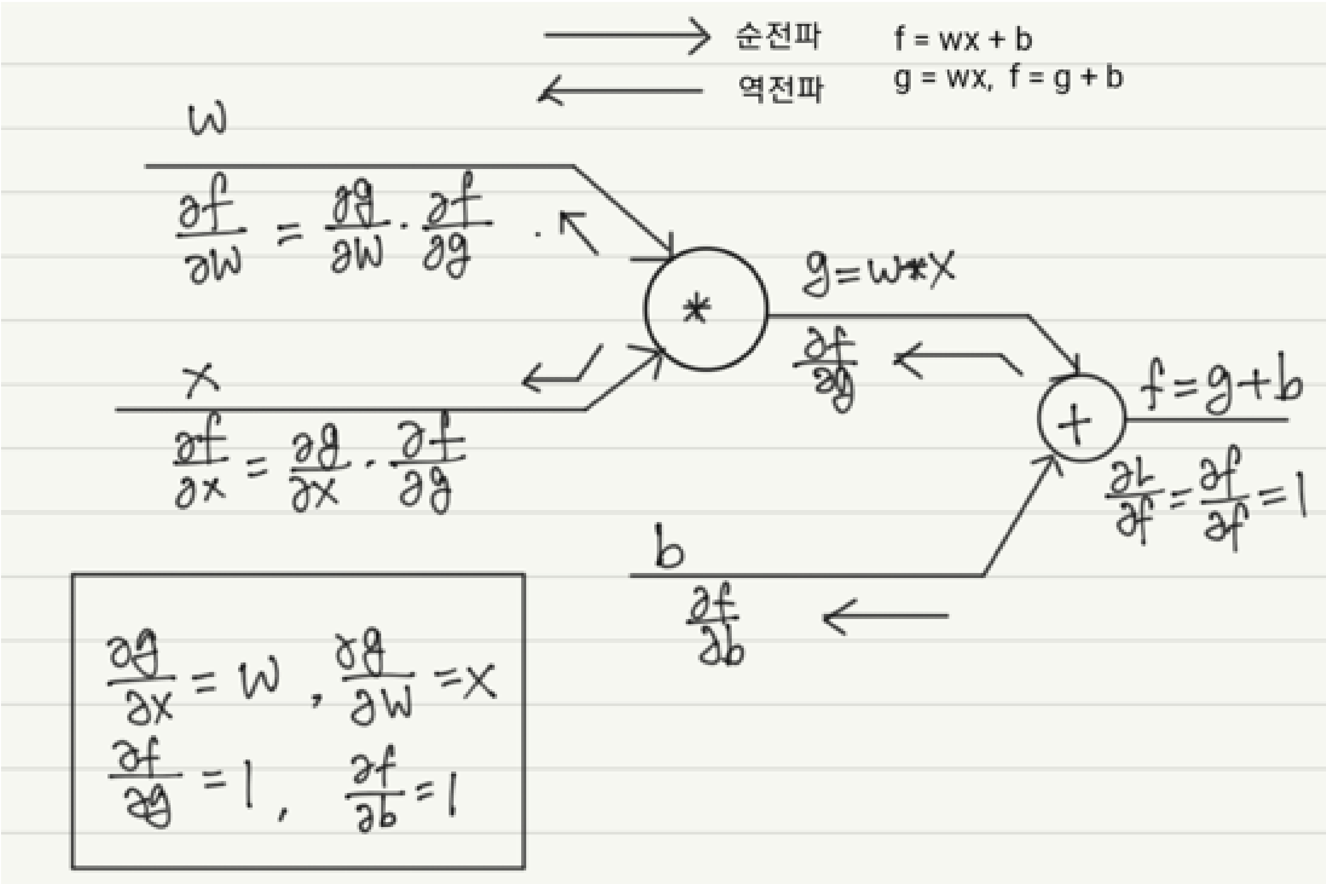
$$\frac{\partial z}{\partial t} = 2t$$

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

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

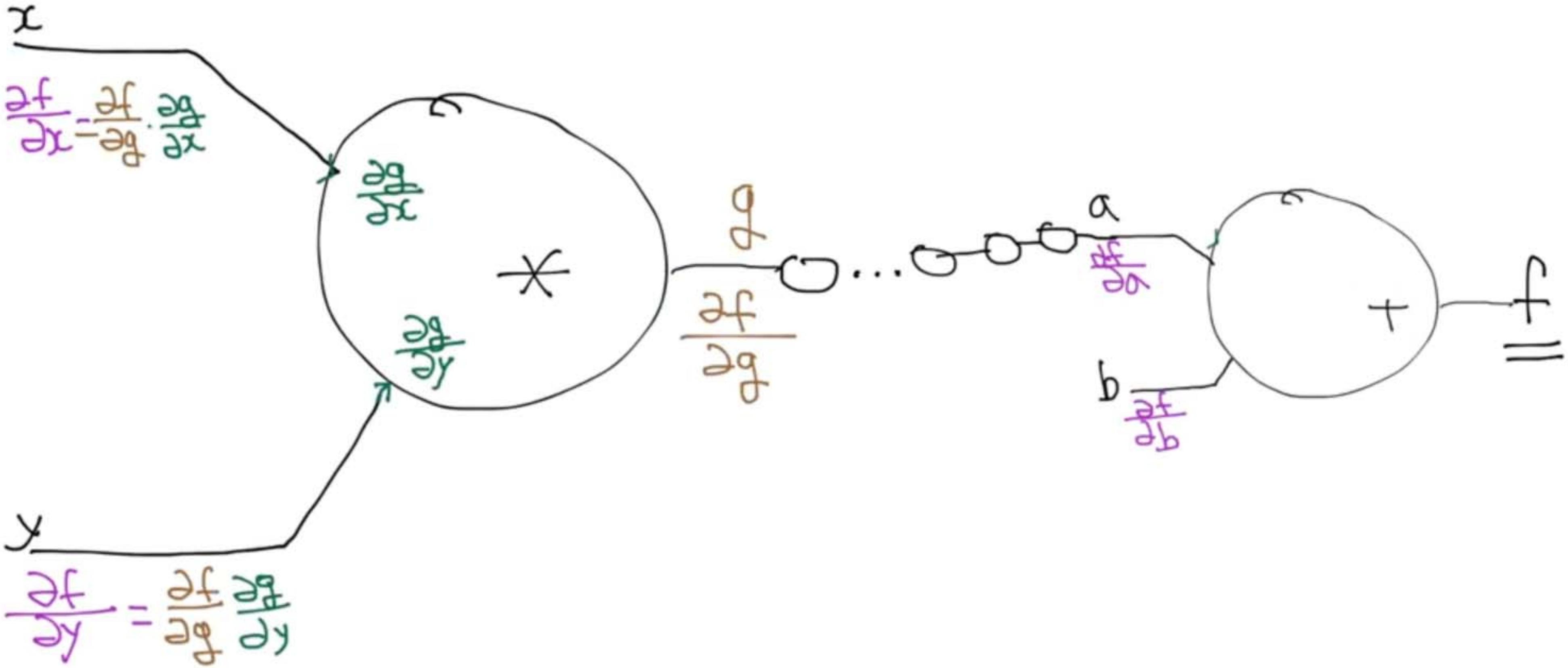
오차 역전파

- **연쇄 법칙에 따른 역전파 계산**



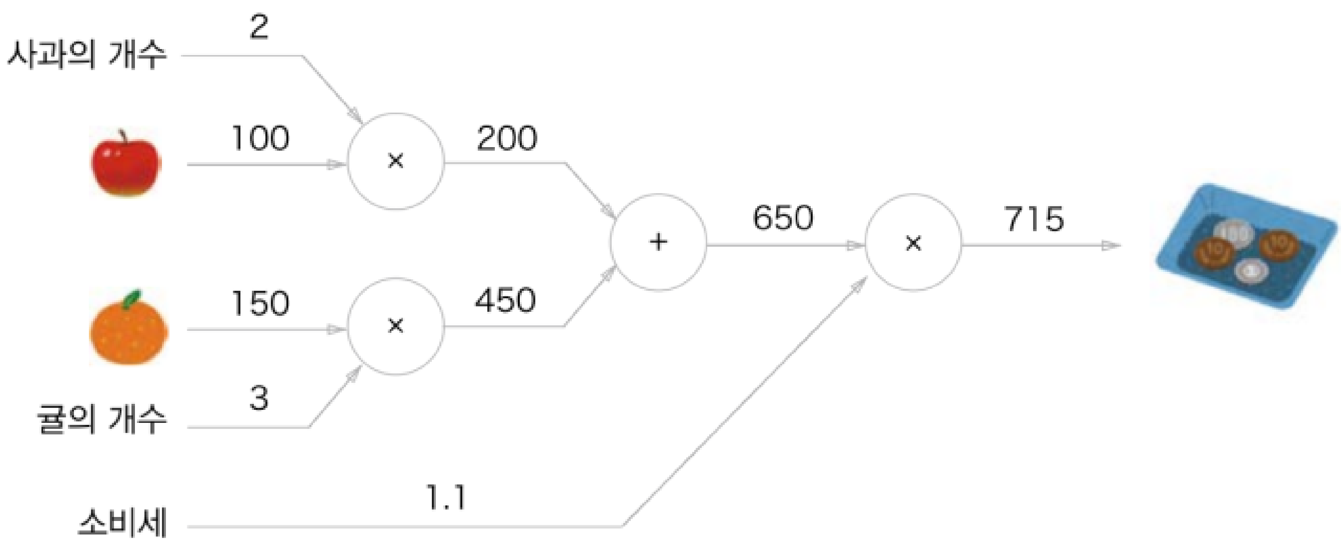
오차 역전파

- **연쇄 법칙에 따른 역전파 계산**



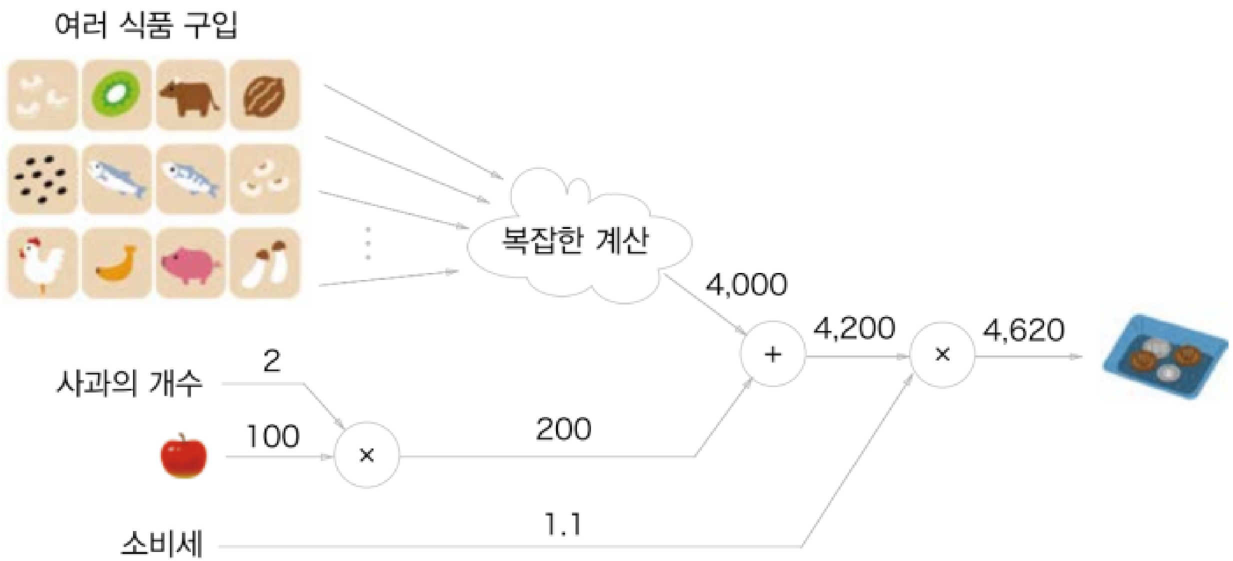
계산 그래프

- **계산 과정을 그래프로 표현**
  - 복수의 노드**(node)**와 에지**(edge)**로 표현
- **그래프에 의한 문제 풀이**
  - 소비세가 10%인 100원하는 사과 2개와 150원하는 귤 3개를 구매하는 경우
  - 계산은 왼쪽에서 오른쪽으로 이루어 짐 (순전파 – forward propagation)



계산 그래프

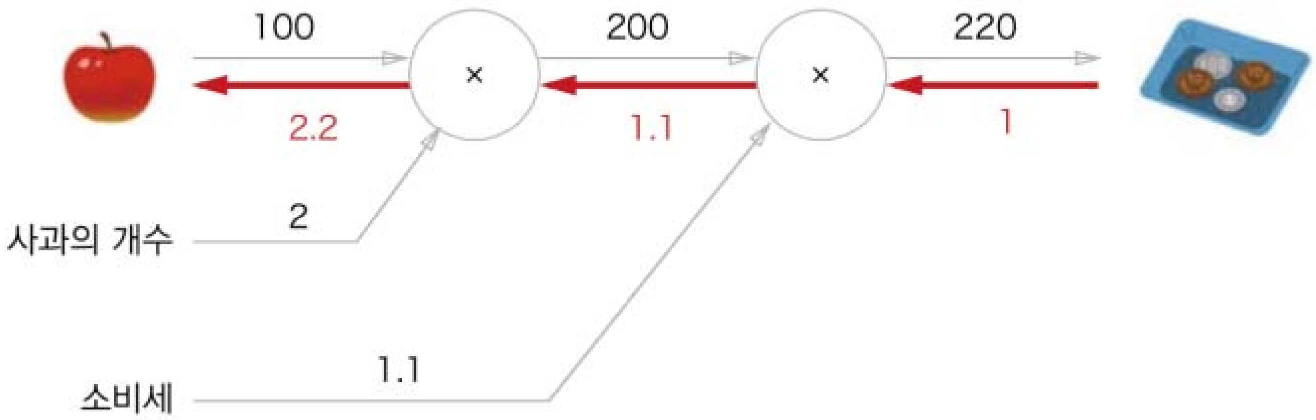
- **국소적 (local) 계산**
  - 국소적 계산을 전파함으로써 최종 결과를 쉽게 얻을 수 있다
    - ✓ 자신과 직접 관계된 작은 범위에 대해서만 계산하고 그 결과를 전파
  - 아래 그림의 사과는 다른 물품 값(4,000)이 어떻게 계산되었는지 상관없이 덧셈을 진행한다
    - ✓ 연산이 매우 단순해 진다



계산 그래프

역전파에 의한 미분 값의 전달

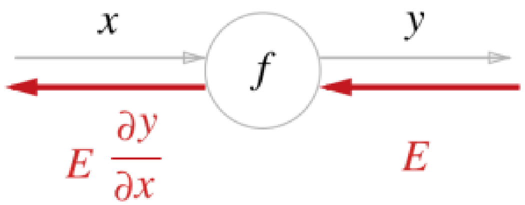
- 아래 그림에서 사과 가격이 최종 금액에 미치는 영향은 지불 금액을 사과 가격으로 미분
  - ✓ 미분 값은 역전파를 통해 계산



계산 그래프

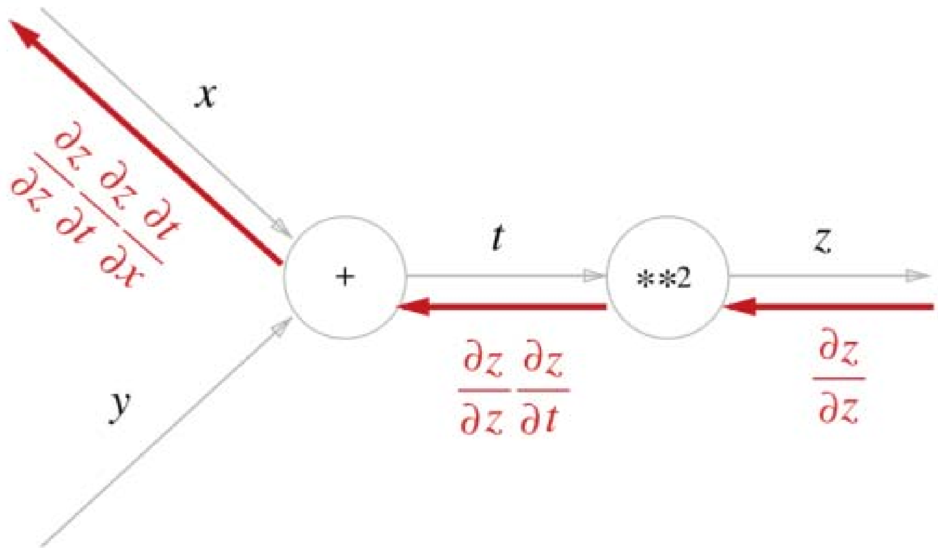
계산 그래프의 역전파

- 순방향과 반대 방향으로 국소적 미분 값을 곱한다.
- 아래 그림은 y = f(x) 계산의 역전파를 표현하고 있다
  - ✓ $f(x) = x^2$ 이라면 x에 대한 미분 값은 2x 가 되며, 이를 노드 왼쪽으로 전달한다



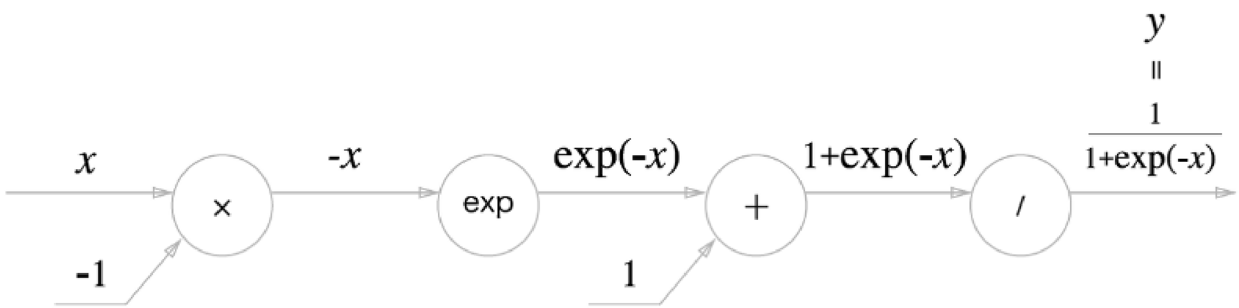
계산 그래프

- **연쇄 법칙과 계산 그래프**
  - $z = t^2$, $t = x + y$ 를 계산 그래프로 표현

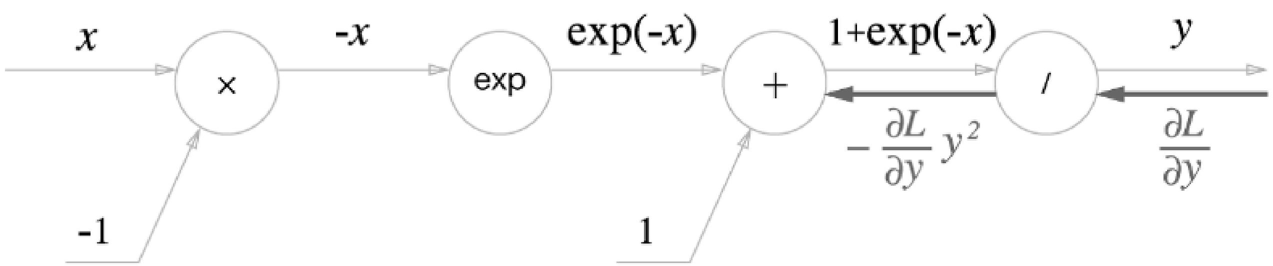


계산 그래프

- **계산 그래프를 이용한 시그모이드 미분**
  - 그래프 표현

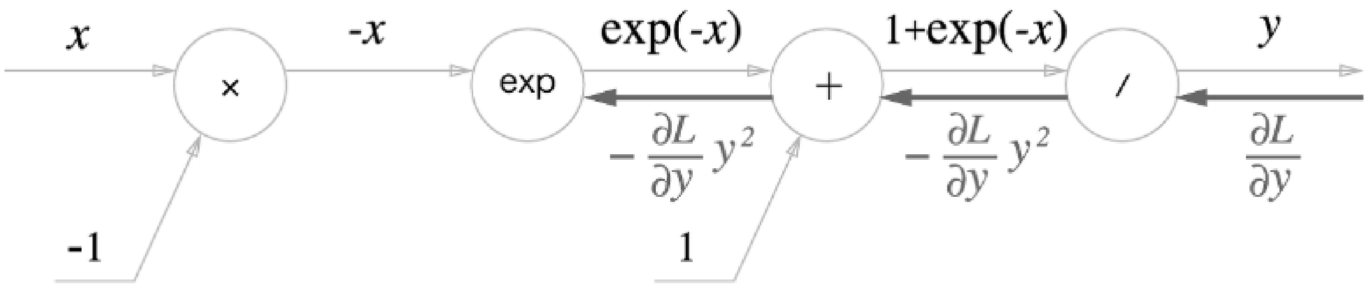


  - 1 단계
    - ✓ '/' 노드 즉, y = 1/x 미분은 $\frac{\partial y}{\partial x} = -\frac{1}{x^2} = -y^2$ 이 된다

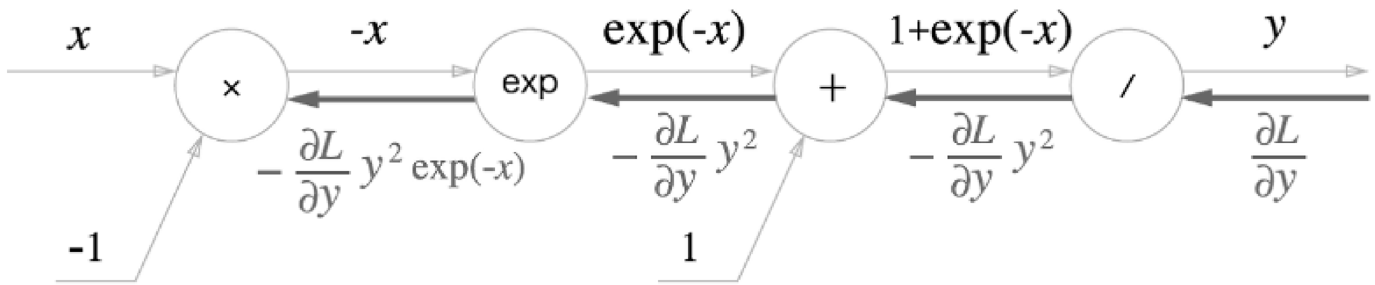


계산 그래프

- **계산 그래프를 이용한 시그모이드 미분**
  - 2 단계
    - ✓ '+' 노드는 상류의 값을 여과 없이 하류로 내보낸다

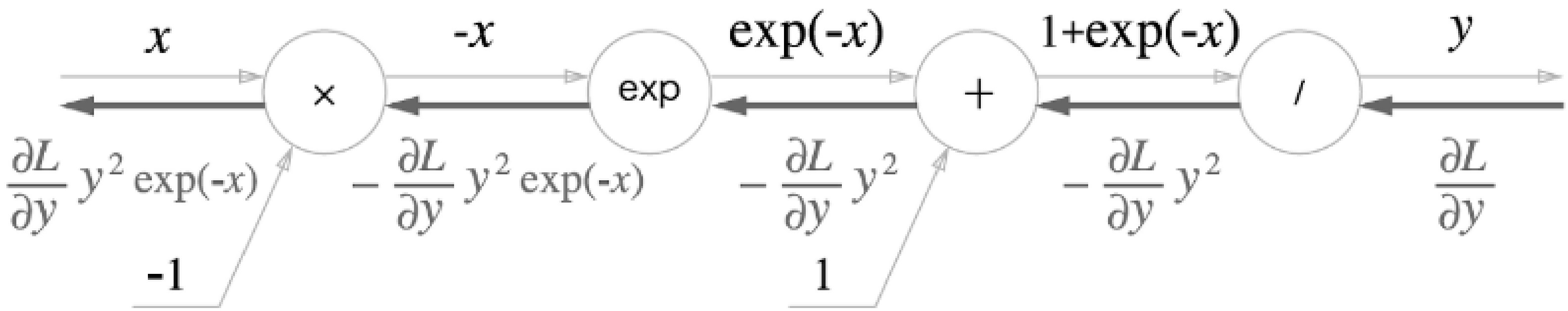


  - 3단계
    - ✓ 'exp' 노드의 미분

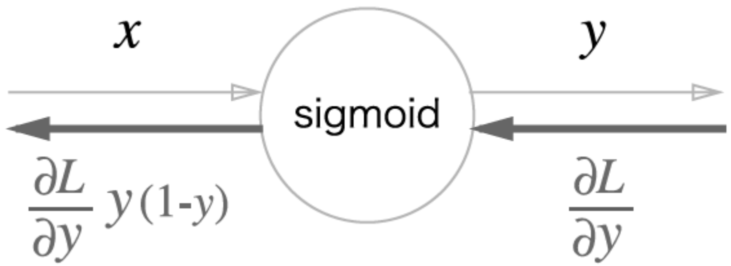


계산 그래프

- **계산 그래프를 이용한 시그모이드 미분**
  - 4 단계
    - ✓ 'x' 노드는 순전파 때의 값을 '서로 바꿔' 곱한다. 이 예에서는 -1을 곱하면 된다



$$
\begin{aligned}
\frac{\partial L}{\partial y} y^2 \exp(-x) &= \frac{\partial L}{\partial y} \frac{1}{(1+\exp(-x))^2} \exp(-x) \\
&= \frac{\partial L}{\partial y} \frac{1}{1+\exp(-x)} \frac{\exp(-x)}{1+\exp(-x)} \\
&= \frac{\partial L}{\partial y} y(1-y)
\end{aligned}
$$



계산 그래프

- **시그모이드 계층에 대한 파이썬 구현**

```python
class Sigmoid:
    def __init__(self):
        self.out = None

    def forward(self, x):
        out = sigmoid(x)
        self.out = out
        return out

    def backward(self, dout):
        dx = dout * (1.0 - self.out) * self.out

        return dx
```

934v00