오차역전파

계산 그래프, 연쇄법칙, 역전파

비타민 13기 4조 | 이예령, 엄성원, 임형수, 차용진

발표 순서

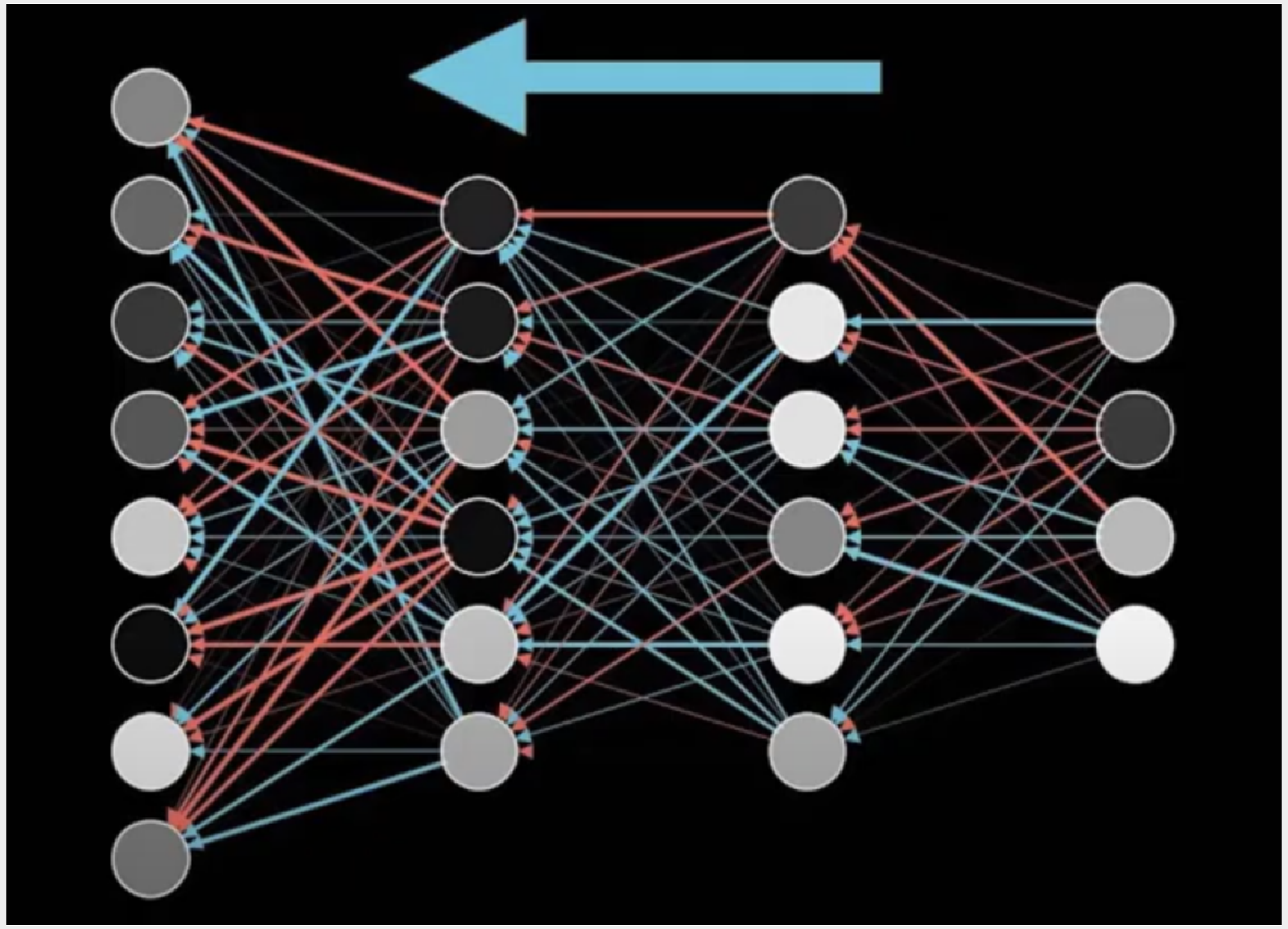
역전파 간단 소개

계산 그래프

연쇄법칙 (Chain Rule)

역전파

역전파 간단 소개



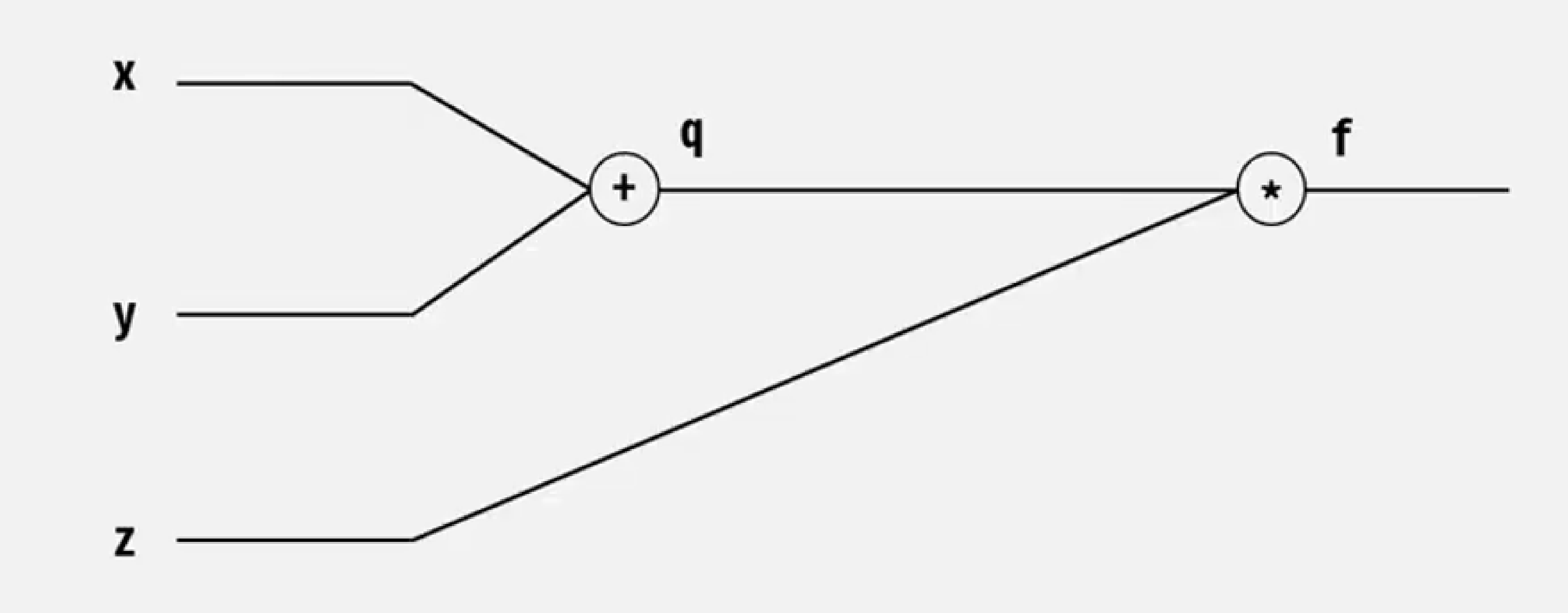
역전파 알고리즘의 작동 방식

손실함수 오차값 (실제값 – 예측값) 계산 후, 거꾸로 뒤로 전파해 가며 각 노드가 가지고 있는 가중치 매개변수 값, 즉 weight 값을 갱신해나가는 과정

계산 그래프

$$f(x, y, z) = (x + y)z$$

$$q = x + y$$
$$f = qz$$



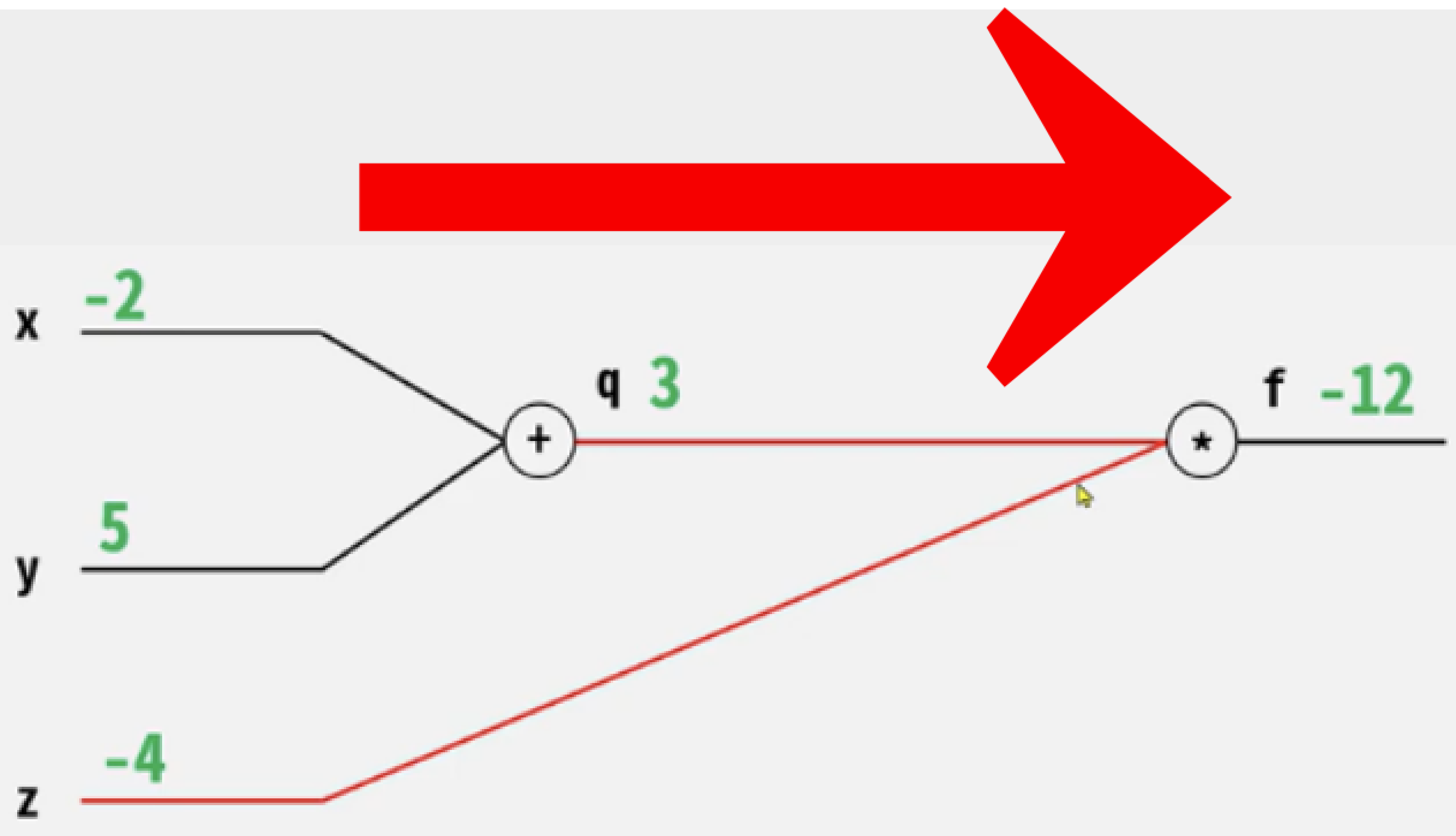
계산 그래프

$$f(x, y, z) = (x + y)z$$

$$q = x + y$$

$$f = qz$$

x=-2, y=5, z=-4



Forward Pass (순전파)

계산 그래프

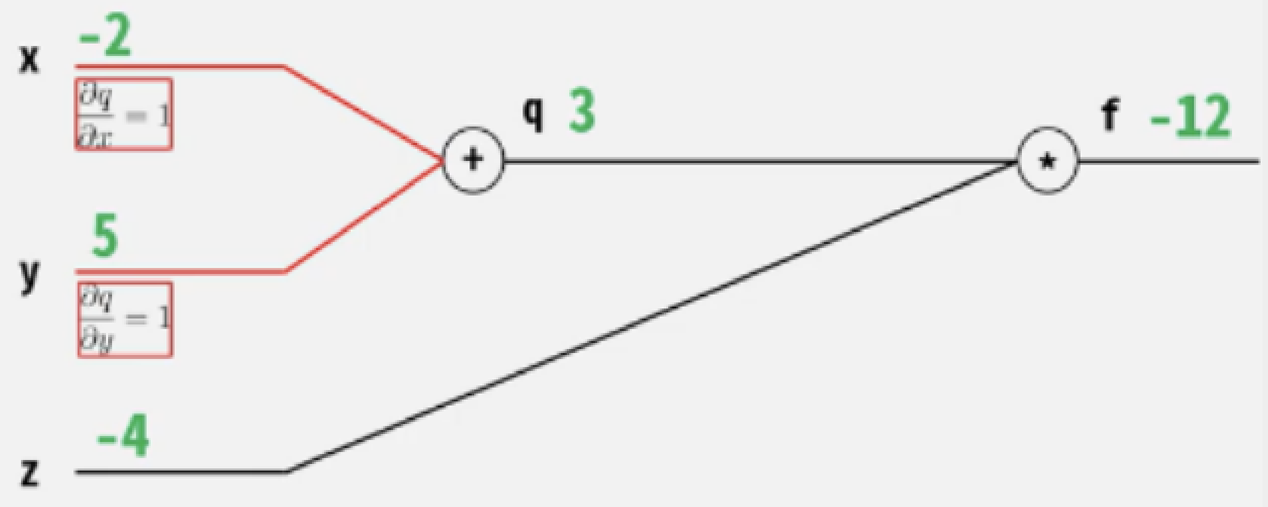
$$\frac{\partial q}{\partial x} = ? \qquad \frac{\partial q}{\partial y} = ?$$

$$f(x, y, z) = (x + y)z$$

x=-2, y=5, z=-4

$$q = x + y$$

$$\frac{\partial q}{\partial x} = 1 \qquad \frac{\partial q}{\partial y} = 1$$



Q1. df/dy (f에 대한 y의 미분값) ???

$$\boxed{\frac{\partial f}{\partial y} = ?}$$

60

연쇄 법칙 (Chain Rule)

$$\boxed{f = qz}$$

$$\frac{\partial f}{\partial q} = ?$$

$$\frac{\partial f}{\partial z} = ?$$

$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}}$$

연쇄 법칙 (Chain Rule)

$$\boxed{f = qz}$$

$$\frac{\partial f}{\partial q} = ?$$

$$\frac{\partial f}{\partial z} = ?$$

$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\cancel{\partial q}} \frac{\cancel{\partial q}}{\partial x}}$$

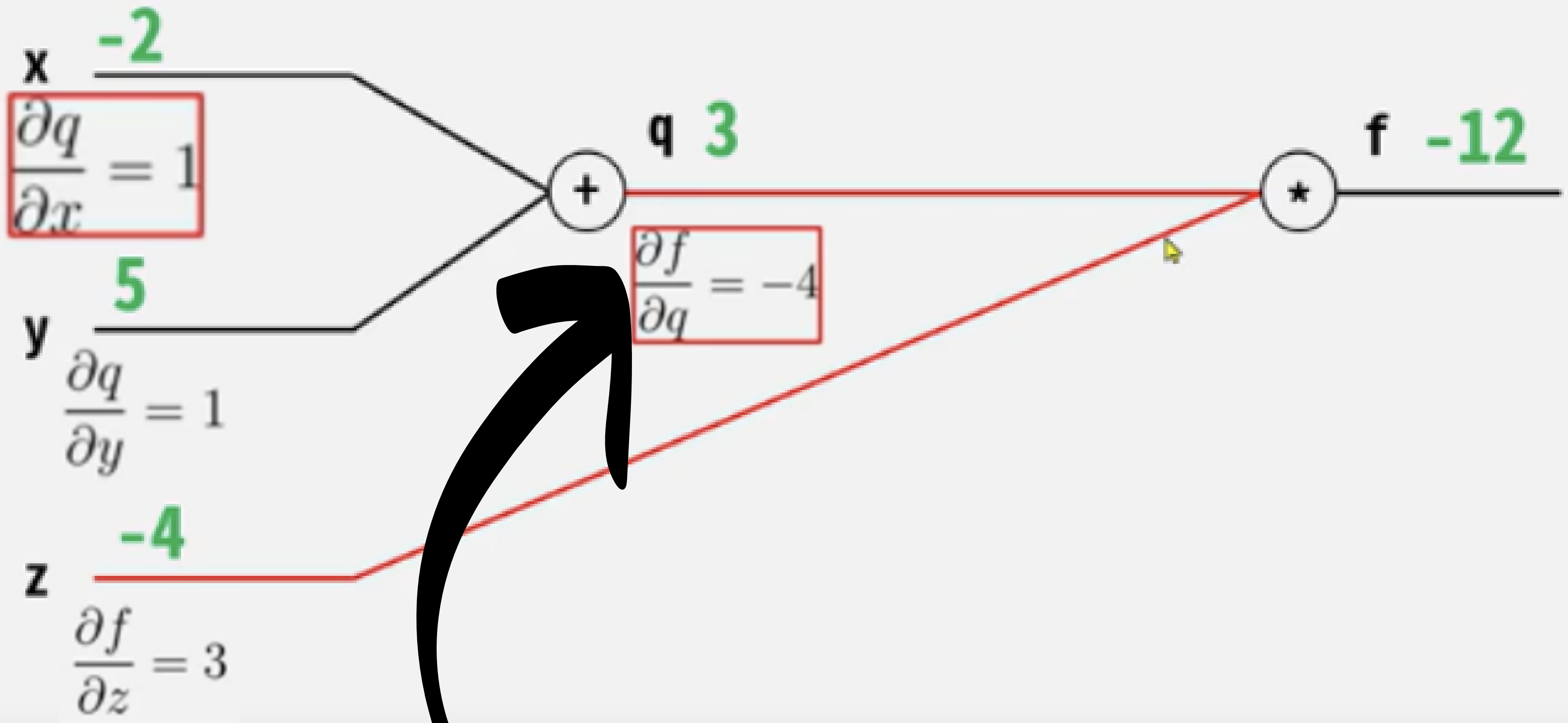
연쇄 법칙 (Chain Rule)

$$f(x, y, z) = (x + y)z$$

$$q = x + y$$

$$f = qz$$

x=-2, y=5, z=-4



$$\boxed{\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}}$$

Q1. df/dy (f에 대한 y의 미분값) 정답

$$\frac{\partial f}{\partial y} = ?$$

$$f(x, y, z) = (x + y)z$$

x=-2, y=5, z=-4

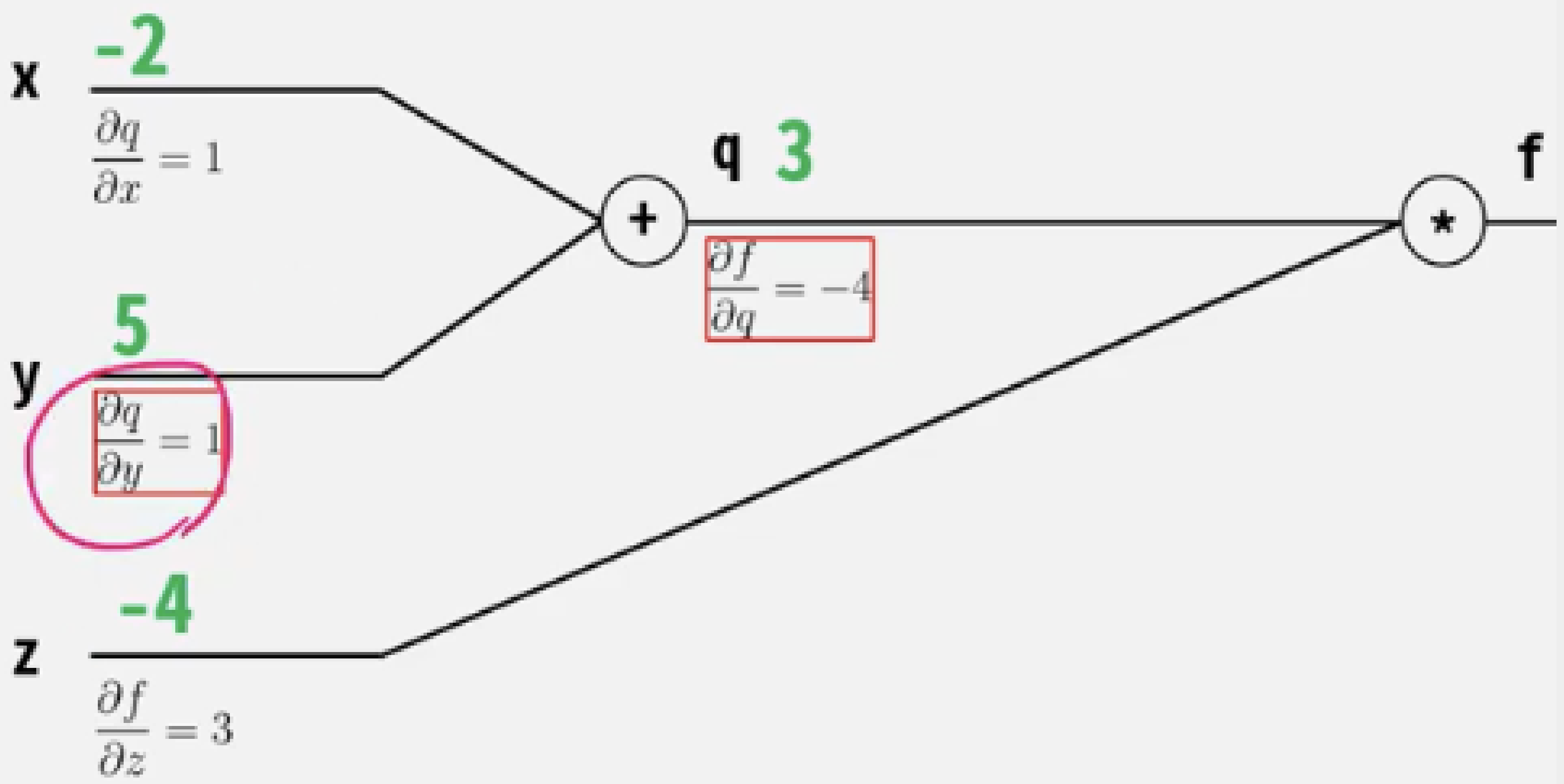
$$q = x + y$$

$$f = qz$$

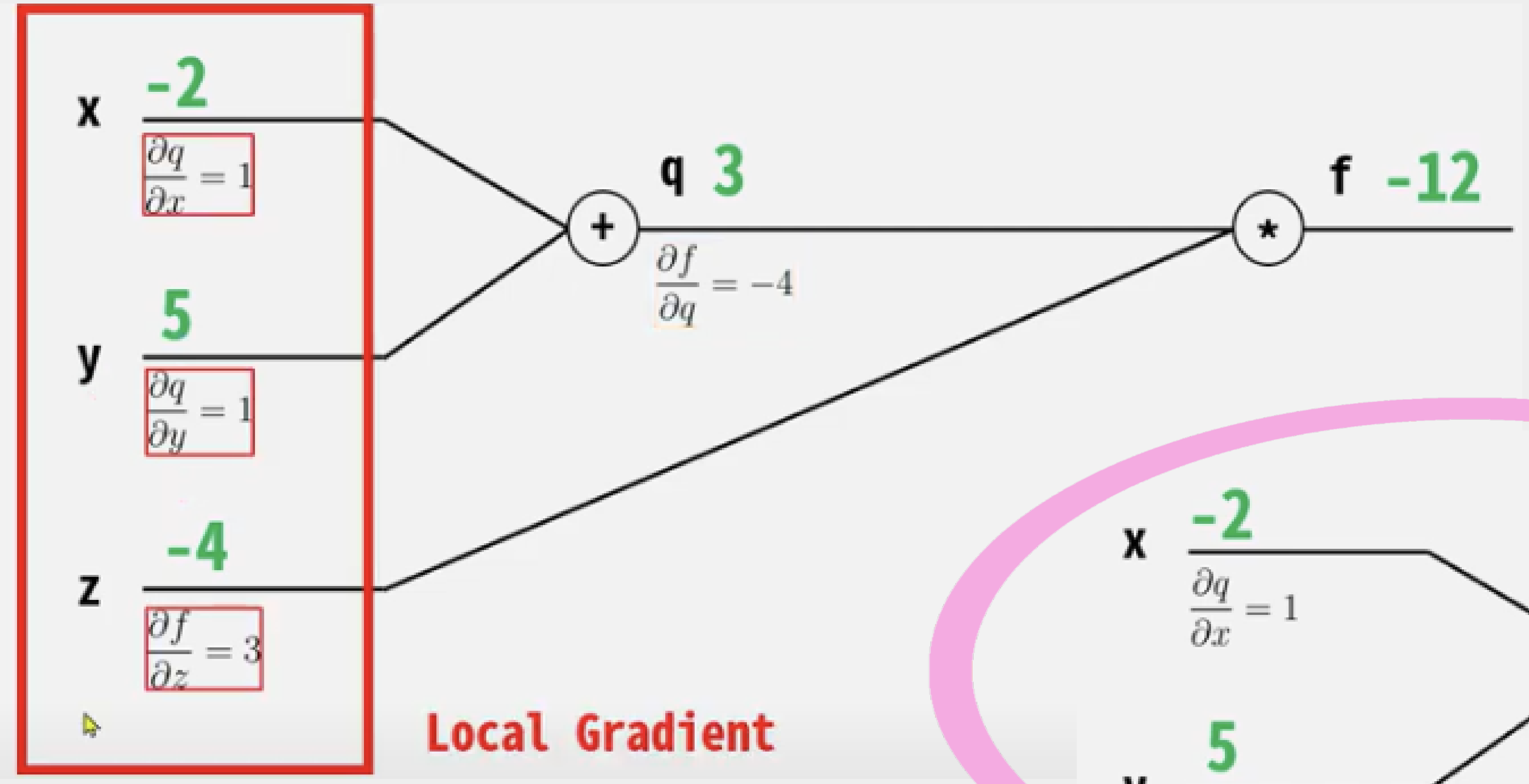
Chain Rule

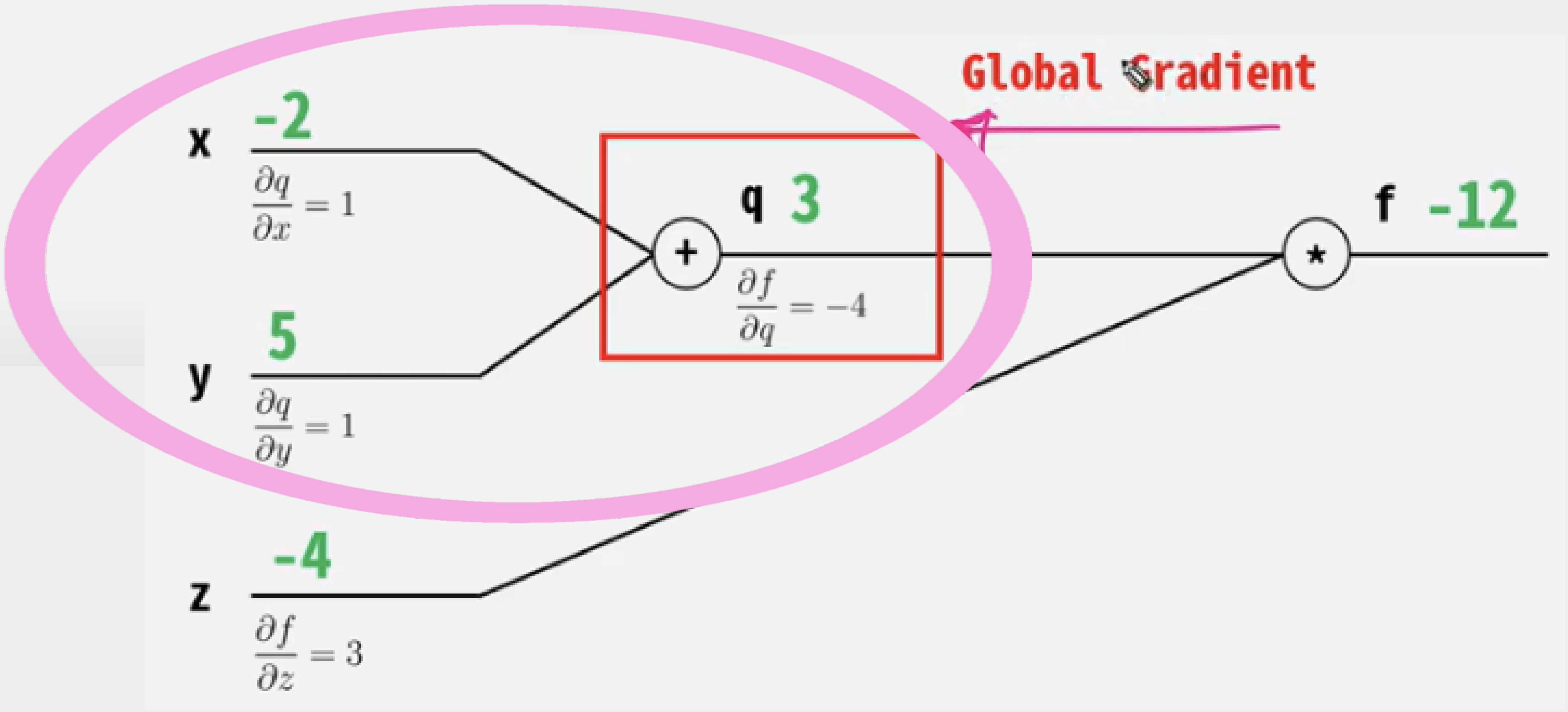
$$\frac{\partial f}{\partial x} = -4$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial y} =$$



연쇄 법칙 (Chain Rule)





연쇄 법칙 (Chain Rule)

Forward Pass

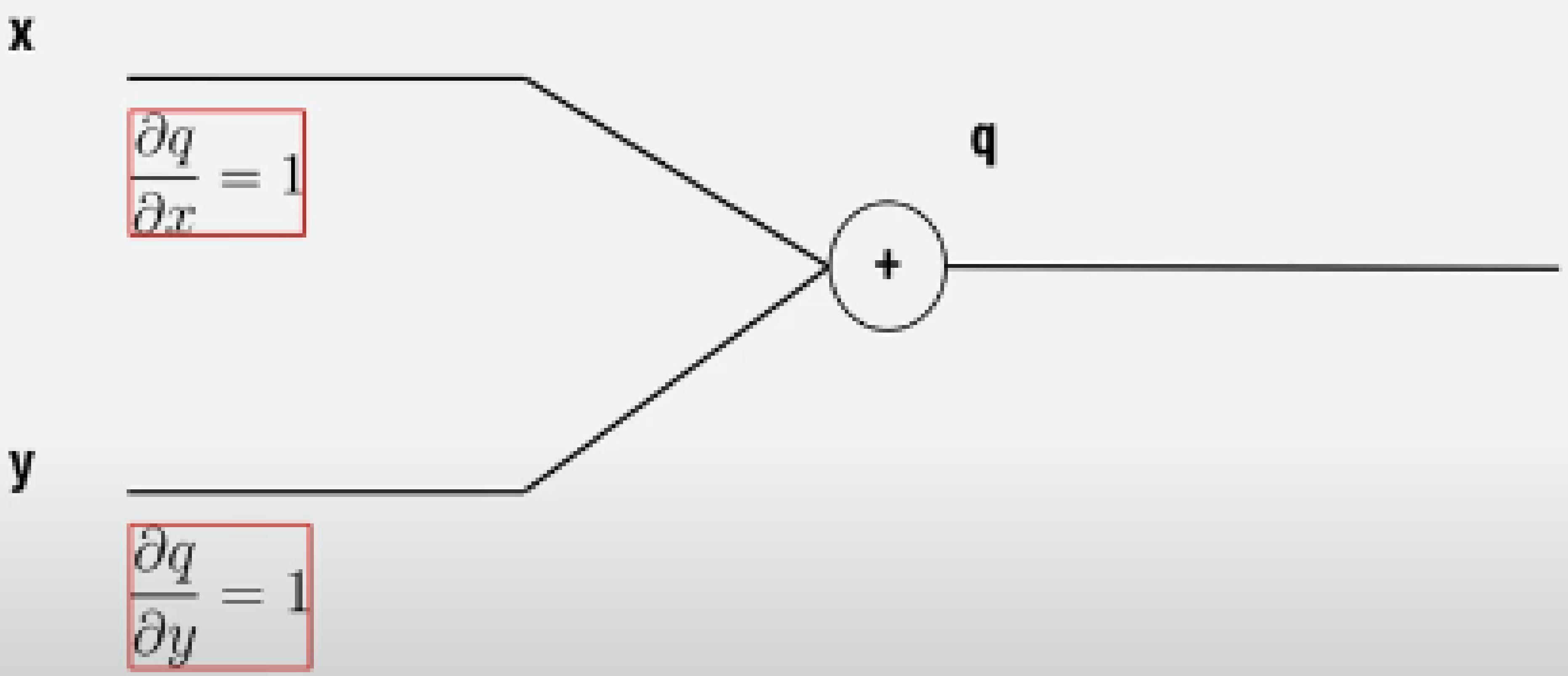
$$q = x + y$$

$$\frac{\partial q}{\partial x} = 1 \qquad \frac{\partial q}{\partial y} = 1$$

$$f = qz$$

$$\frac{\partial f}{\partial q} = z \qquad \frac{\partial f}{\partial z} = q$$

Forward Pass 시 우리는 Local Gradient를 미리 구하여 저장할 수 있다!



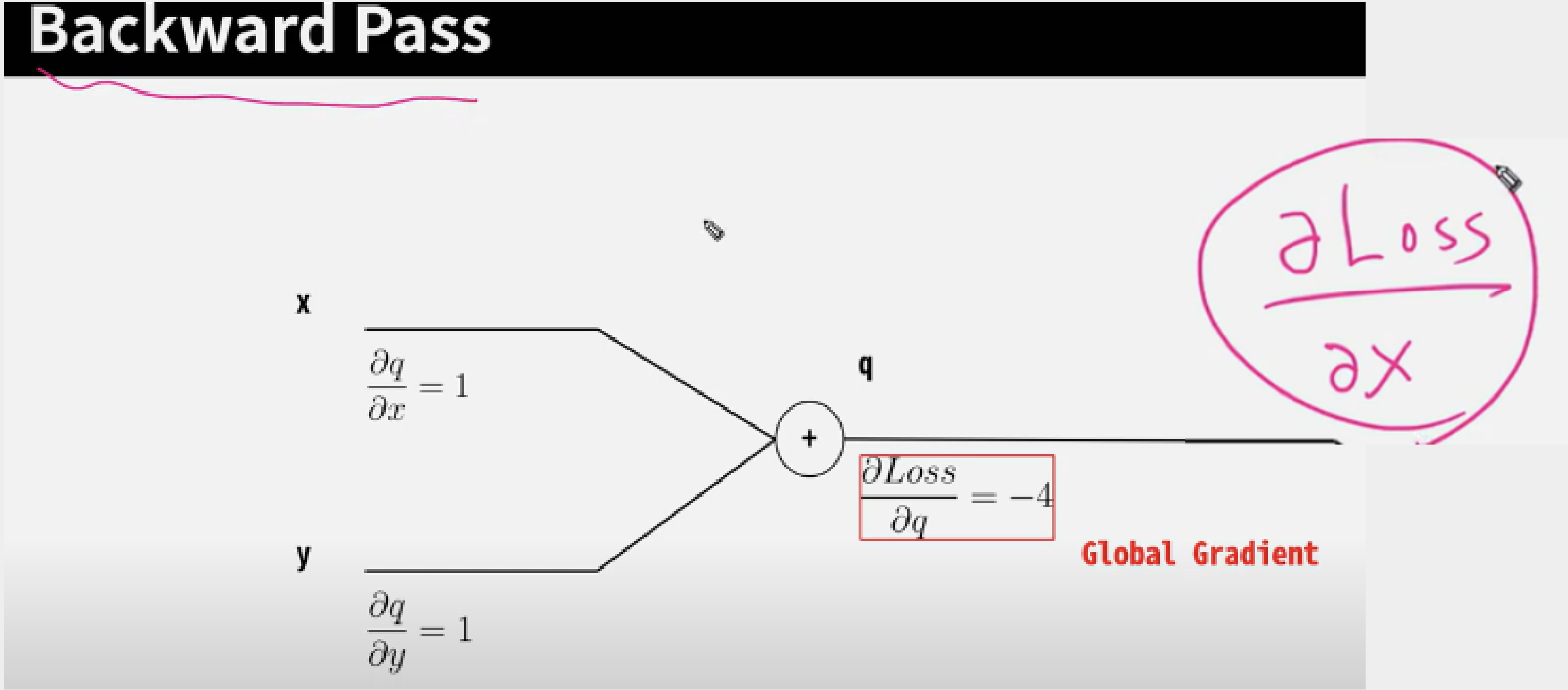
$$q = x + y$$

$$\frac{\partial q}{\partial x} = 1 \qquad \frac{\partial q}{\partial y} = 1$$

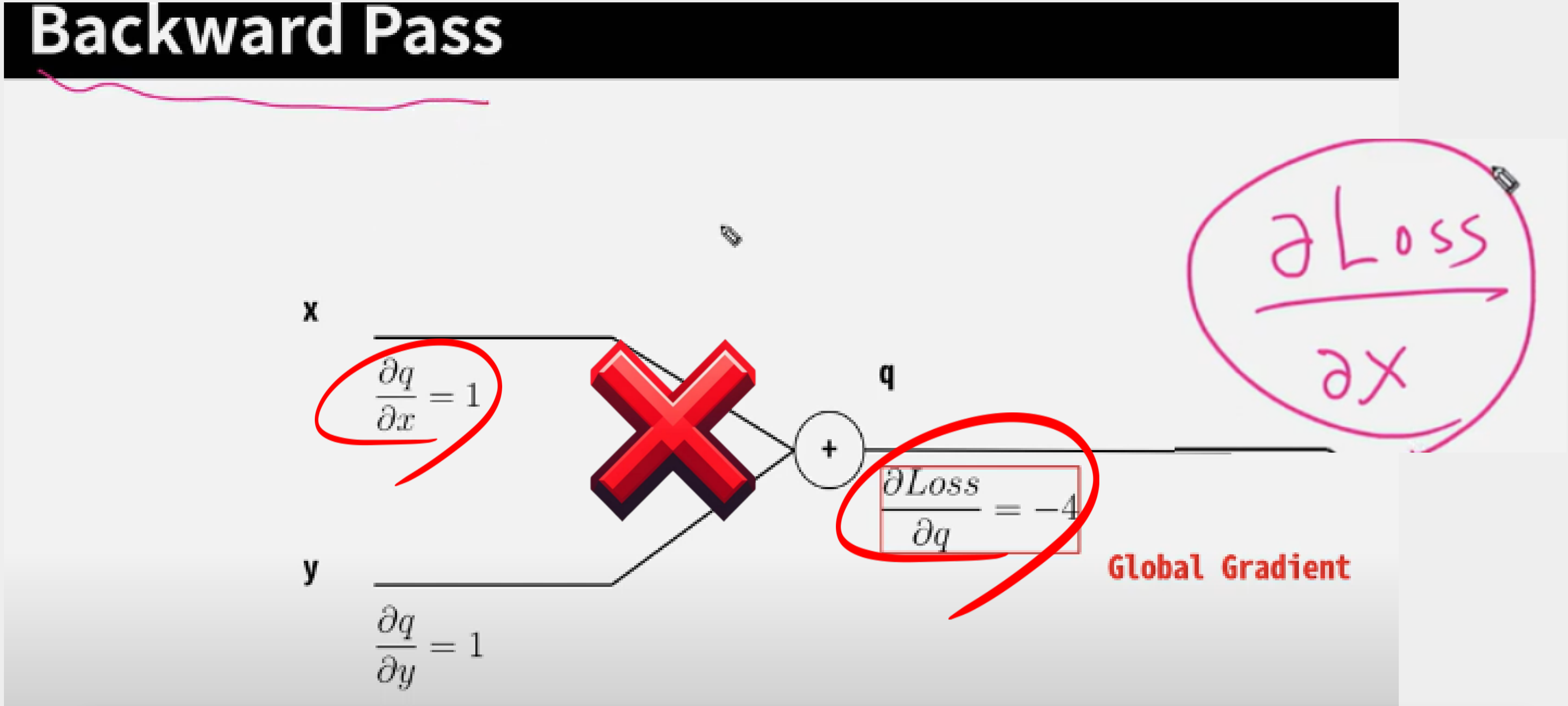
$$f = qz$$

$$\frac{\partial f}{\partial q} = z \qquad \frac{\partial f}{\partial z} = q$$

연쇄 법칙 (Chain Rule)

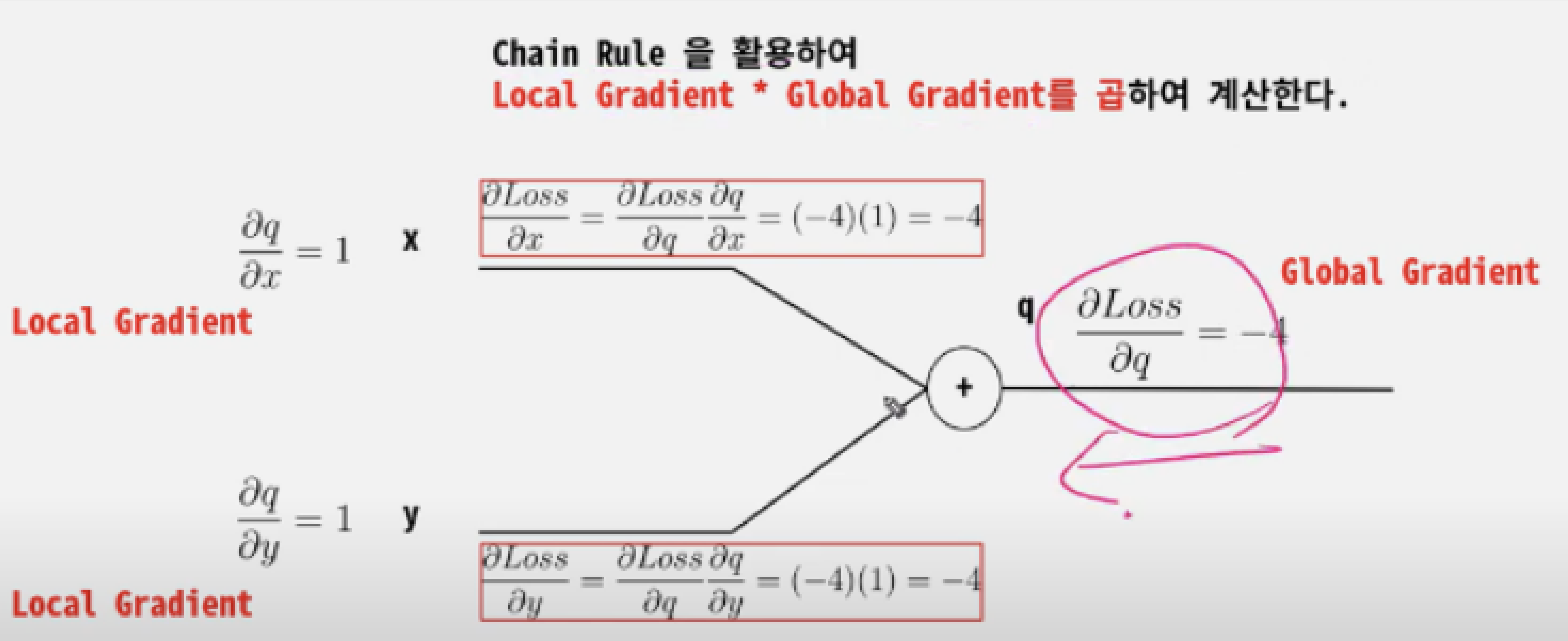


연쇄 법칙 (Chain Rule)



연쇄 법칙(Chain Rule)을 활용하여
Local Gradient * Global Gradient를 곱하여 계산

연쇄 법칙 (Chain Rule)



Q2

손실함수와 기울기 값을 통해 가중치, 즉 웨이트 값을
업데이트하는 과정을 모델이 ( )한다고 말한다.

30

Q2. 손실함수와 기울기 값을 통해 웨이트 값을 업데이트하는 과정

$$q = x + y$$

$$\frac{\partial q}{\partial x} = 1 \qquad \frac{\partial q}{\partial y} = 1$$

$$f = qz$$

$$\frac{\partial f}{\partial q} = z \qquad \frac{\partial f}{\partial z} = q$$

결론

내용 요약

아무리 깊고 복잡한 층으로 구성되어 있다 하더라도
Chain Rule을 활용하여 미분 값을 얻어낼 수 있다.

Forward Pass 시 Local Gradient를 미리 계산하여 저장해둔다.

저장해둔 Local Gradient와 Global Gradient를 Backward Pass시
곱하여 최종 미분 값을 얻는다.

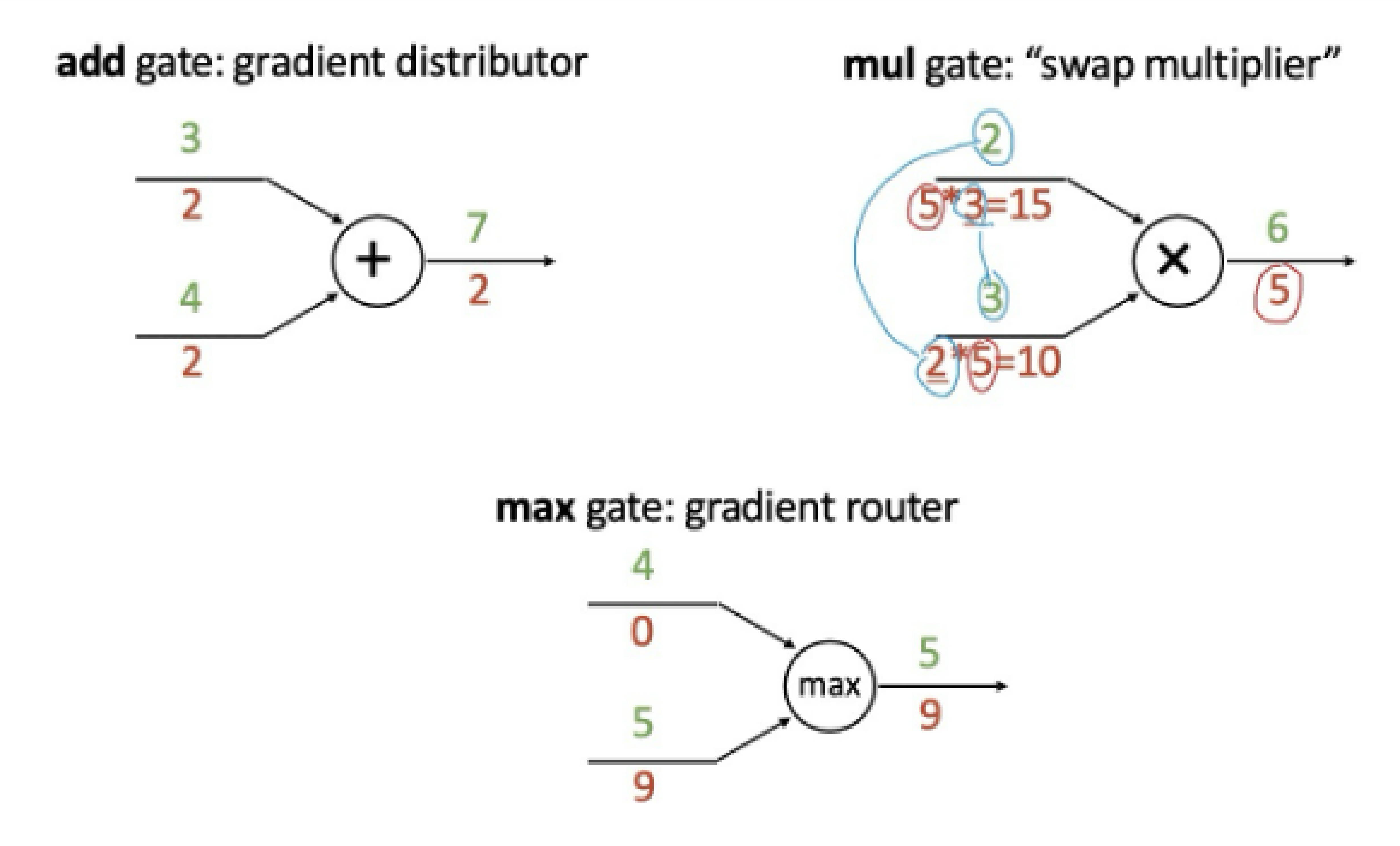
**Q3. Loss에 대한 최종 미분값을 구하기 위해서는
두가지 값을 곱하면 된다. 그 두가지 값에 해당하는 것?
A3. ( ) * ( )**

오차역전파

계산 그래프, 연쇄법칙, 역전파

비타민 13기 4조 | 이예령, 엄성원, 임형수, 차용진

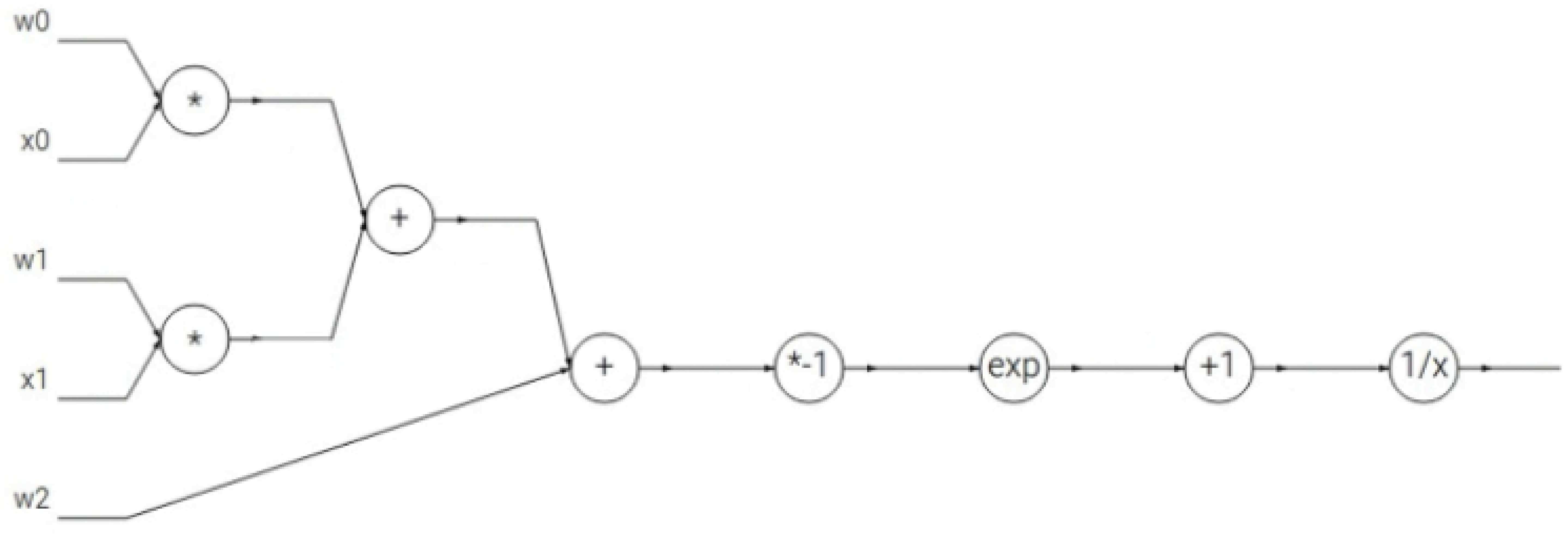
역전파 시 연산의 특성



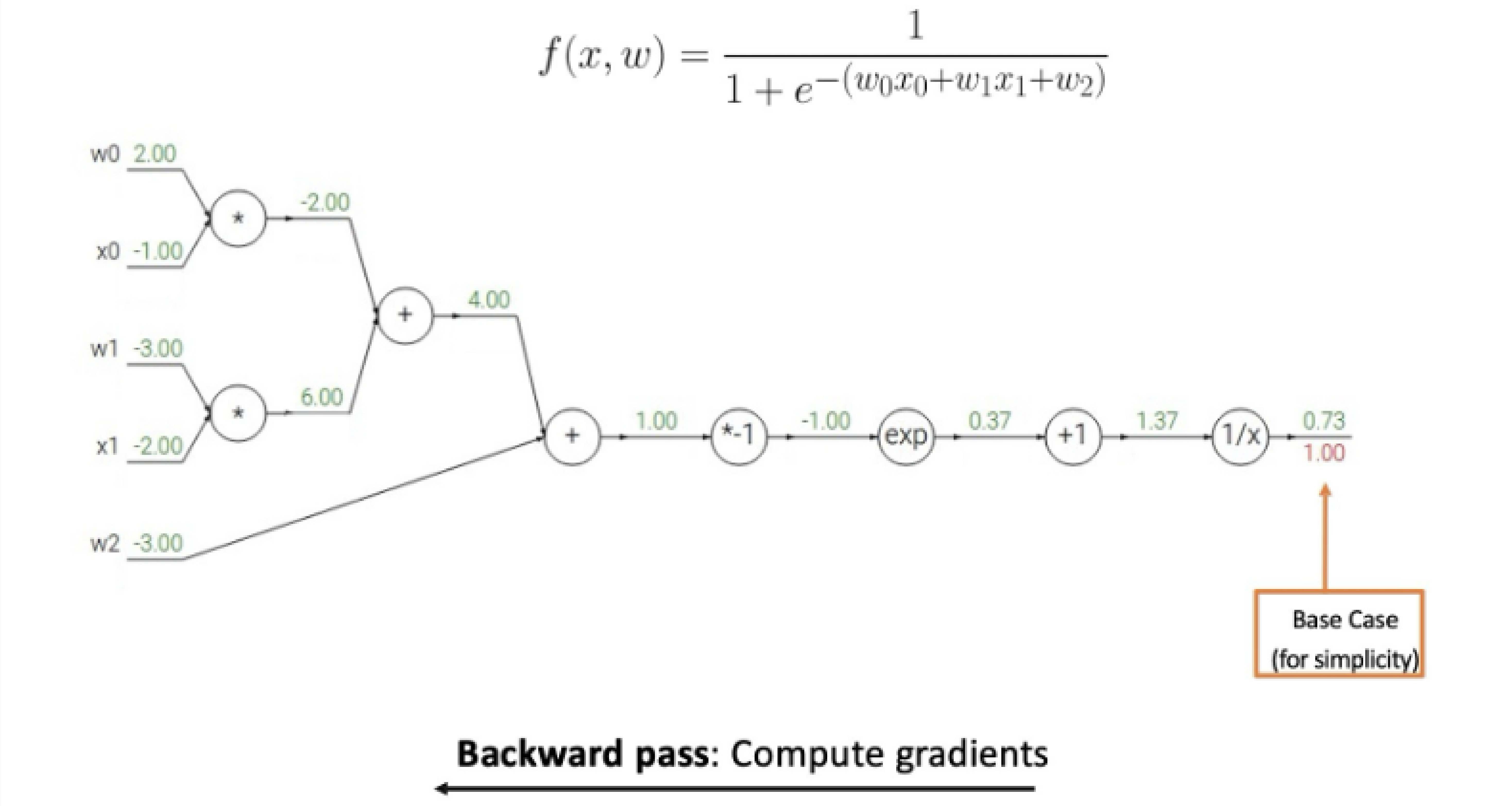
add gate: gradient distributor
3
2
4
2
7
2
mul gate: "swap multiplier"
2
5*3=15
3
2*5=10
6
5
max gate: gradient router
4
0
5
9
max
5
9

역전파

$$f(x, w) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$

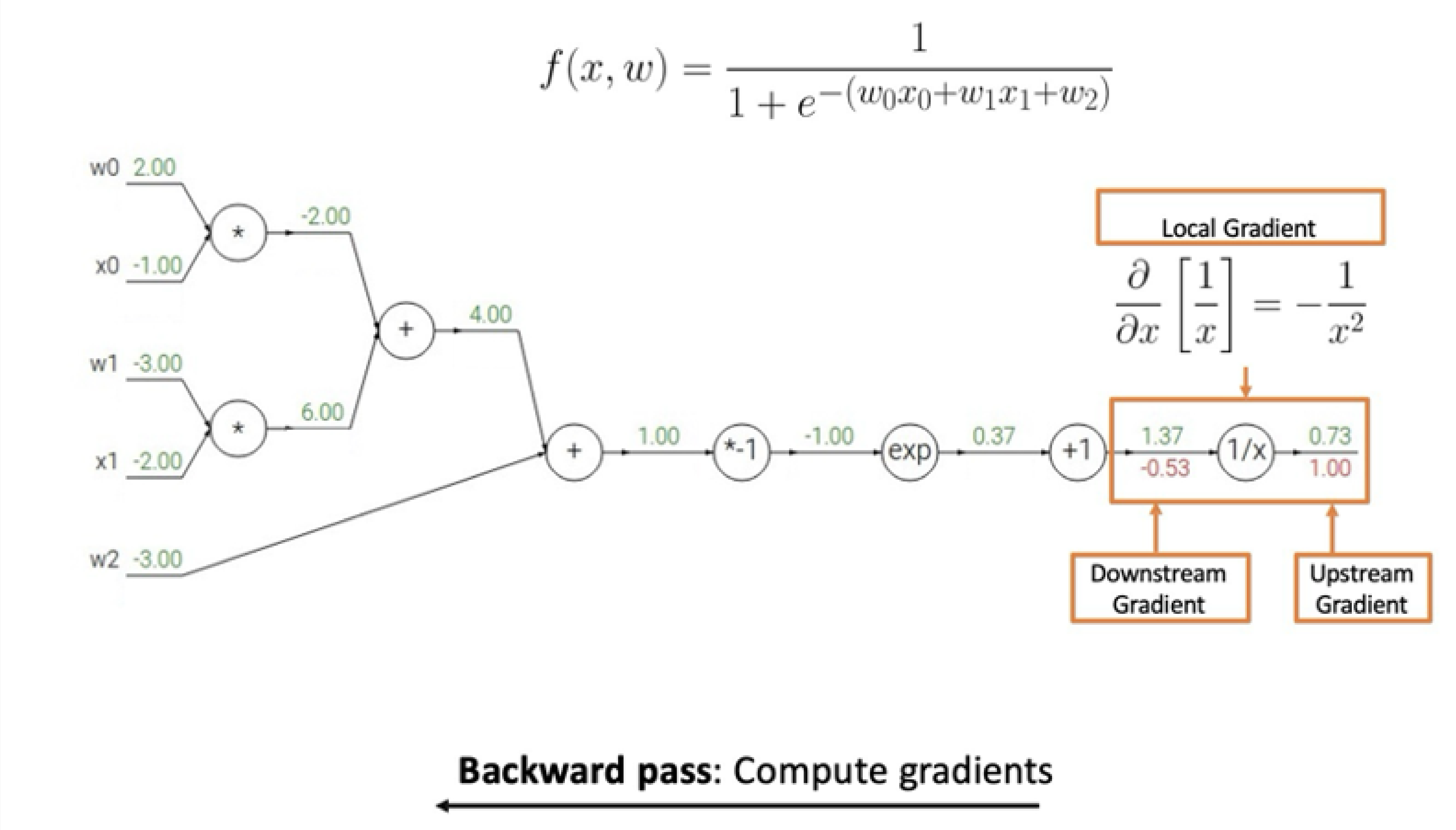


역전파



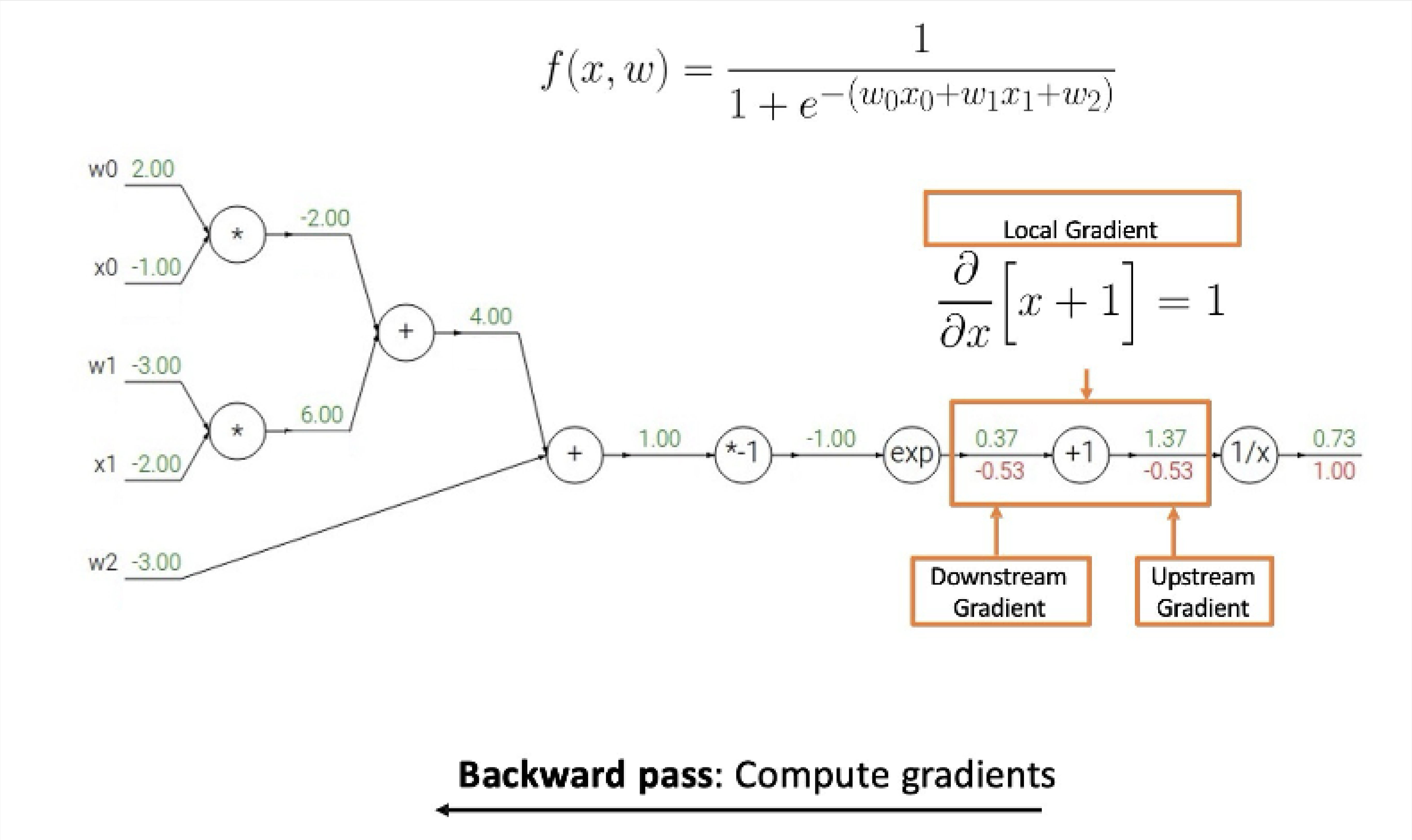
$$f(x,w) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$
w0 2.00
x0 -1.00
w1 -3.00
x1 -2.00
w2 -3.00
-2.00
6.00
4.00
1.00
*-1
-1.00
exp
0.37
+1
1.37
1/x
0.73
1.00
Base Case
(for simplicity)
Backward pass: Compute gradients

역전파



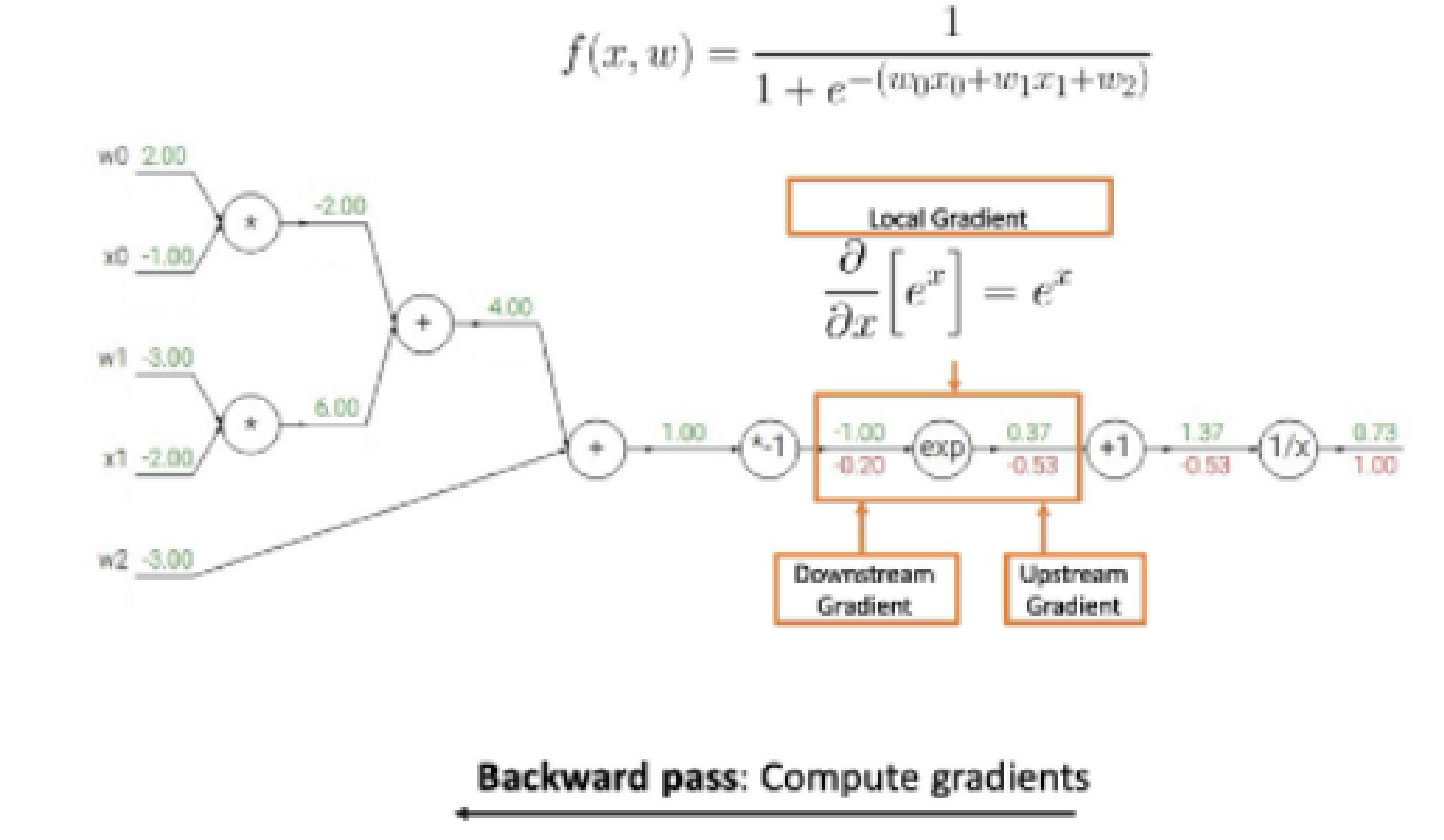
$$f(x,w) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$
w0 2.00
x0 -1.00
w1 -3.00
x1 -2.00
w2 -3.00
-2.00
6.00
4.00
1.00
*-1
-1.00
exp
0.37
+1
1.37
-0.53
1/x
0.73
1.00
Local Gradient
$$\frac{\partial}{\partial x}\left[\frac{1}{x}\right] = -\frac{1}{x^2}$$
Downstream Gradient
Upstream Gradient
Backward pass: Compute gradients

역전파



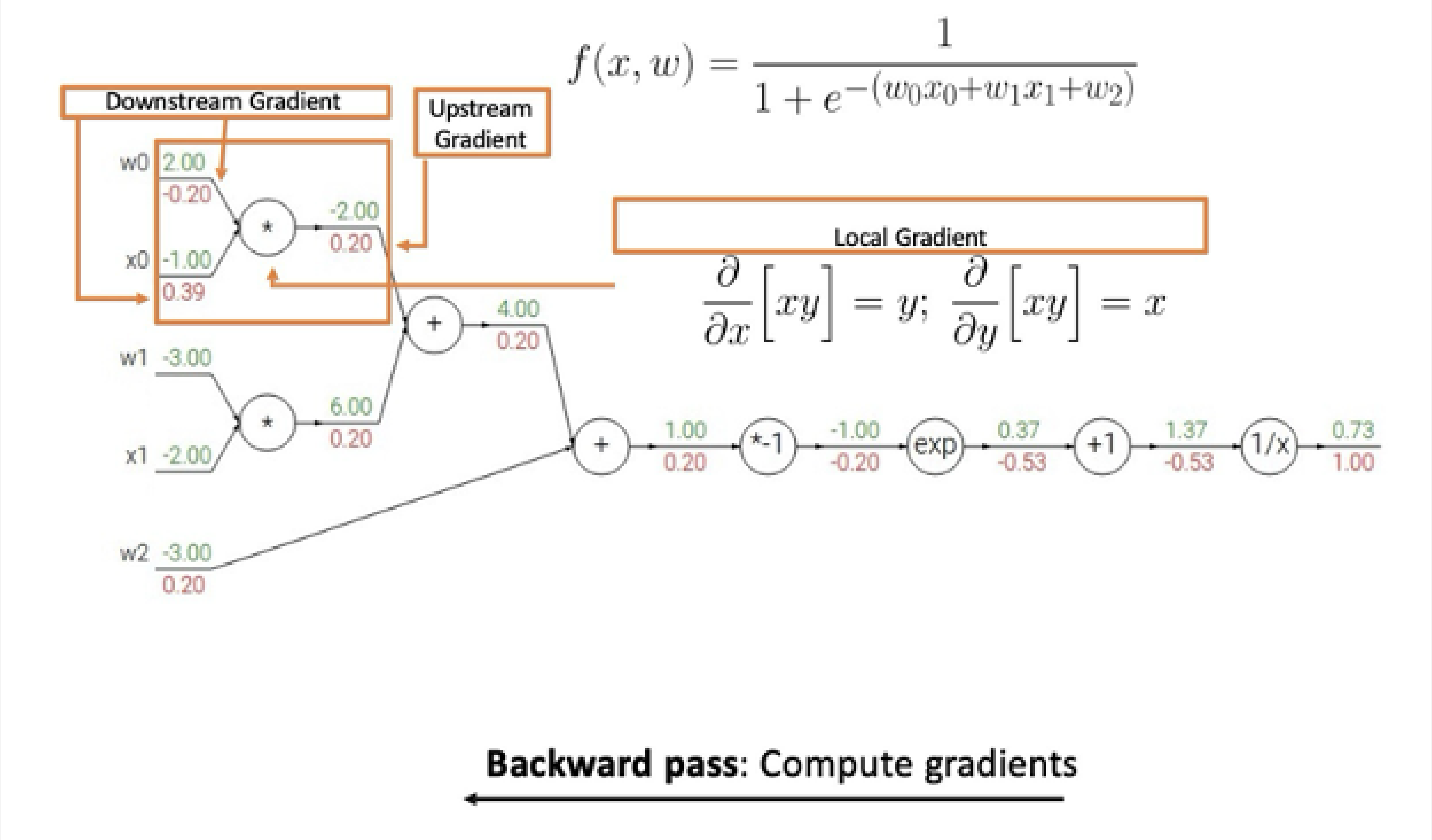
$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + w_2)}}$$
w0 2.00
x0 -1.00
w1 -3.00
x1 -2.00
w2 -3.00
-2.00
6.00
4.00
1.00
*-1
-1.00
exp
0.37
-0.53
+1
1.37
-0.53
1/x
0.73
1.00
Local Gradient
$$\frac{\partial}{\partial x}\left[x + 1\right] = 1$$
Downstream Gradient
Upstream Gradient
Backward pass: Compute gradients

역전파



$$f(x,w) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$
w0 2.00
x0 -1.00
w1 -3.00
x1 -2.00
w2 -3.00
-2.00
6.00
4.00
1.00
*-1
-1.00
-0.20
exp
0.37
-0.53
+1
1.37
-0.53
1/x
0.73
1.00
Local Gradient
$$\frac{\partial}{\partial x}\left[e^x\right] = e^x$$
Downstream Gradient
Upstream Gradient
Backward pass: Compute gradients

역전파



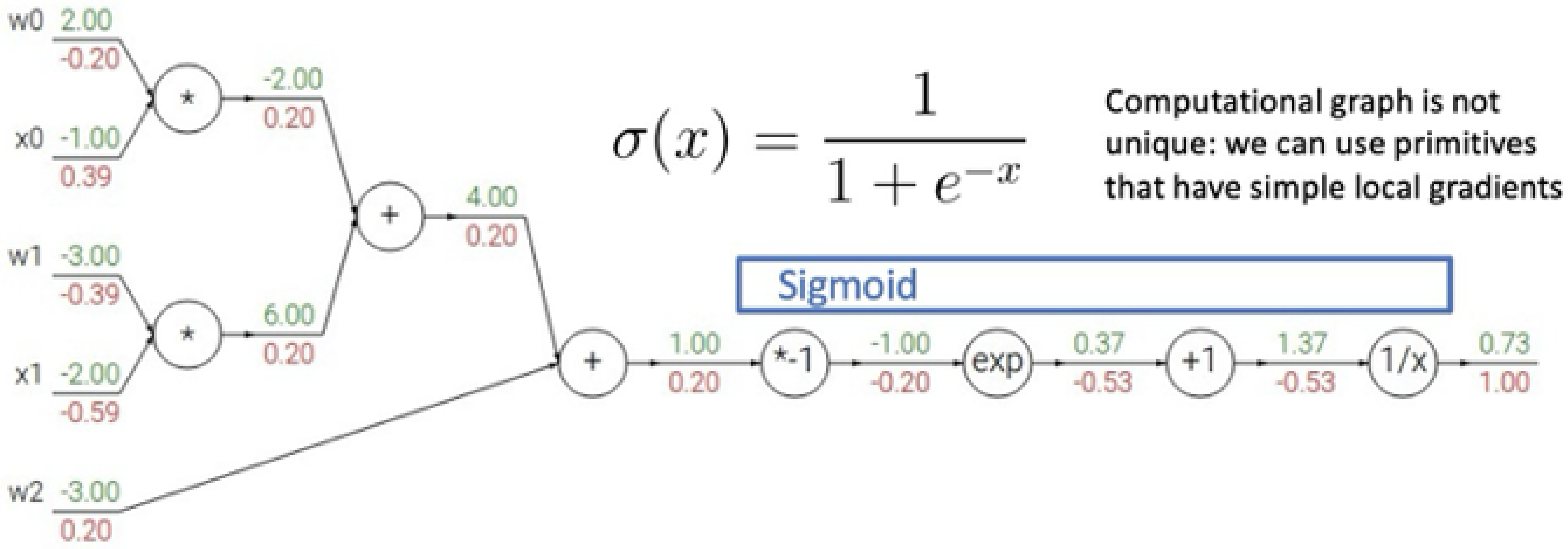
$$f(x, w) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$
Downstream Gradient
Upstream Gradient
Local Gradient
$$\frac{\partial}{\partial x}[xy] = y; \ \frac{\partial}{\partial y}[xy] = x$$
w0 2.00 -0.20
x0 -1.00 0.39
w1 -3.00
x1 -2.00
w2 -3.00 0.20
-2.00 0.20
6.00 0.20
4.00 0.20
1.00 0.20
*-1
-1.00 -0.20
exp
0.37 -0.53
+1
1.37 -0.53
1/x
0.73 1.00
Backward pass: Compute gradients

역전파

$$f(x,w) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}} = \boxed{\sigma(w_0x_0+w_1x_1+w_2)}$$

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

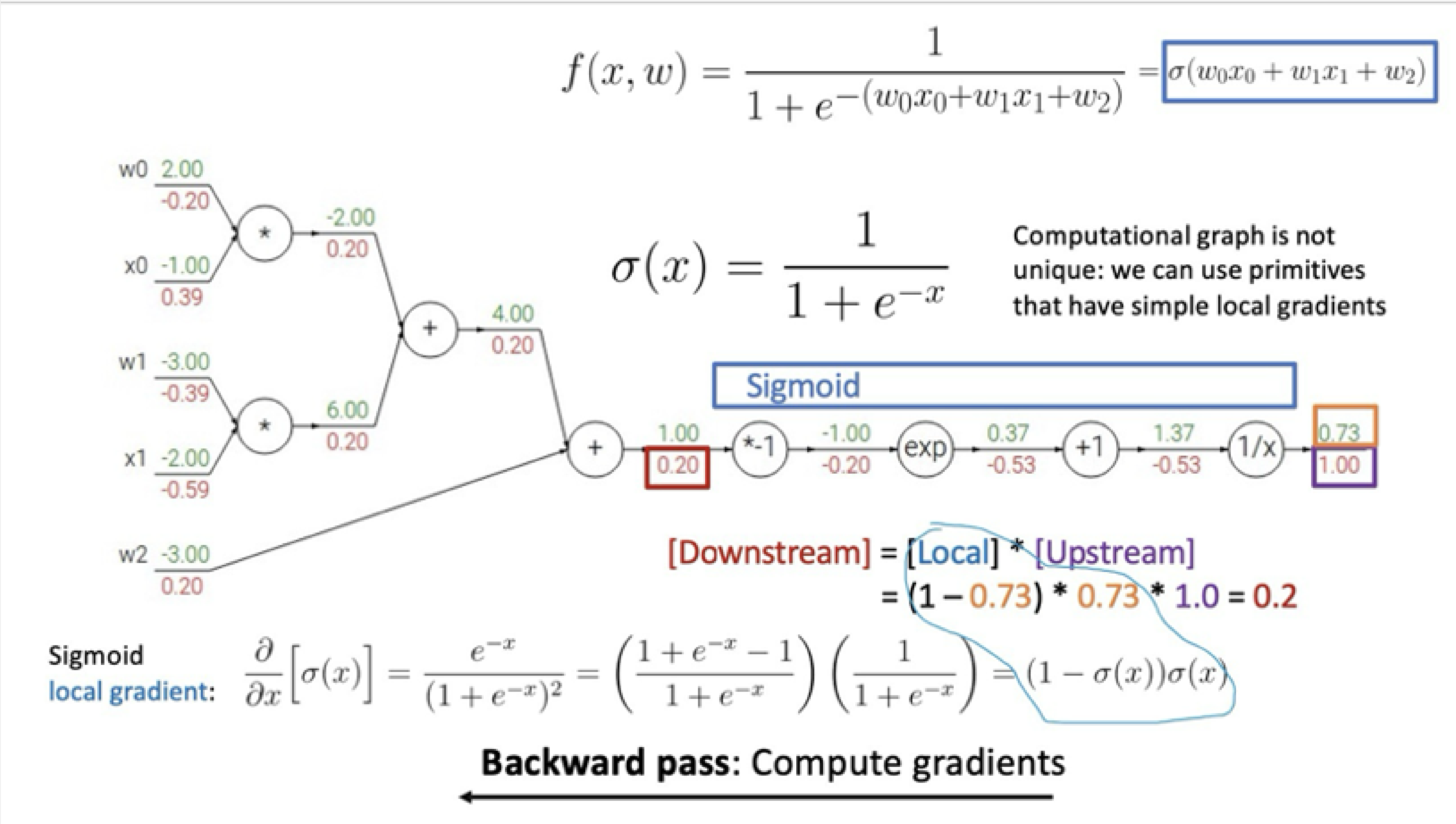
Computational graph is not unique: we can use primitives that have simple local gradients



Sigmoid local gradient: $\frac{\partial}{\partial x}\left[\sigma(x)\right] = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\sigma(x)$

Backward pass: Compute gradients

역전파



$f(x,w) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}} = \sigma(w_0x_0+w_1x_1+w_2)$
w0 2.00 -0.20
x0 -1.00 0.39
w1 -3.00 -0.39
x1 -2.00 -0.59
w2 -3.00 0.20
-2.00 0.20
6.00 0.20
4.00 0.20
1.00 0.20
-1.00 -0.20
0.37 -0.53
1.37 -0.53
0.73 1.00
*
+
*-1
exp
+1
1/x
$\sigma(x) = \frac{1}{1+e^{-x}}$
Computational graph is not unique: we can use primitives that have simple local gradients
Sigmoid
[Downstream] = [Local] * [Upstream]
= (1 – 0.73) * 0.73 * 1.0 = 0.2
Sigmoid local gradient: $\frac{\partial}{\partial x}\left[\sigma(x)\right] = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\sigma(x)$
Backward pass: Compute gradients

역전파

$$x \in \mathbb{R}, y \in \mathbb{R}$$

Regular derivative:

$$\frac{\partial y}{\partial x} \in \mathbb{R}$$

If x changes by a small amount, how much will y change?

$$x \in \mathbb{R}^N, y \in \mathbb{R}$$

Derivative is **Gradient**:

$$\frac{\partial y}{\partial x} \in \mathbb{R}^N \quad \left(\frac{\partial y}{\partial x}\right)_n = \frac{\partial y}{\partial x_n}$$

For each element of x, if it changes by a small amount then how much will y change?

$$x \in \mathbb{R}^{\boxed{N}}, y \in \mathbb{R}^{\boxed{M}}$$

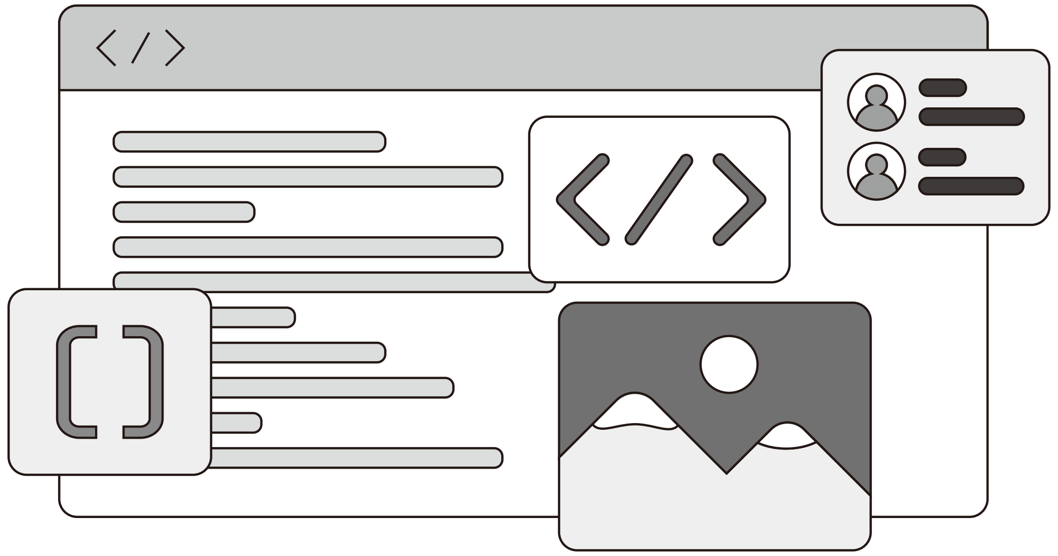
Derivative is **Jacobian**:

$$\frac{\partial y}{\partial x} \in \mathbb{R}^{N \times M} \quad \left(\frac{\partial y}{\partial x}\right)_{n,m} = \frac{\partial y_m}{\partial x_n}$$

For each element of x, if it changes by a small amount then how much will each element of y change?

역전파

```
# 모델 학습
num_epochs = 5
for epoch in range(num_epochs):
    rnn_model.train()
    running_loss = 0.0
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad() # 기울기 초기화
        outputs = rnn_model(images) # 순전파
        loss = criterion(outputs, labels) # 손실 계산
        loss.backward() # 역전파
        optimizer.step() # 최적화
        running_loss += loss.item()
    print(f"Epoch {epoch+1}, Loss: {running_loss/len(train_loader)}")
```



역전파

Q4. 코드 중 역전파를 구현하는 함수는?

```
# 모델 학습
num_epochs = 5
for epoch in range(num_epochs):
    rnn_model.train()
    running_loss = 0.0
    for images, labels in train_loader:
        images, labels = images.to(device), labels.to(device)
        optimizer.zero_grad() # 기울기 초기화
        outputs = rnn_model(images) # 순전파
        loss = criterion(outputs, labels) # 손실 계산
        loss.backward() # 역전파
        optimizer.step() # 최적화
        running_loss += loss.item()
    print(f"Epoch {epoch+1}, Loss: {running_loss/len(train_loader)}")
```

30

출처

https://www.youtube.com/watch?v=1Q_etC_GHHk&t=269s

https://www.youtube.com/watch?v=qGuNJbGWVvI

ChatGPT

발표 경청해 주셔서 감사합니다.