深度学习平台与应用

第四讲：神经网络与反向传播

范琦

fanqi@nju.edu.cn

2024年9月25日

大　　纲

- **神经网络**
- **反向传播**

神经网络

- 最简单的线性分类器

$$f = Wx$$

$$x \in \mathbb{R}^D, W \in \mathbb{R}^{C \times D}$$

神经网络

- 最简单的线性分类器

$$f = Wx$$

$$x \in \mathbb{R}^D, W \in \mathbb{R}^{C \times D}$$

- 2 层神经网络

非线性

$$f = W_2 \max(0, W_1 x)$$

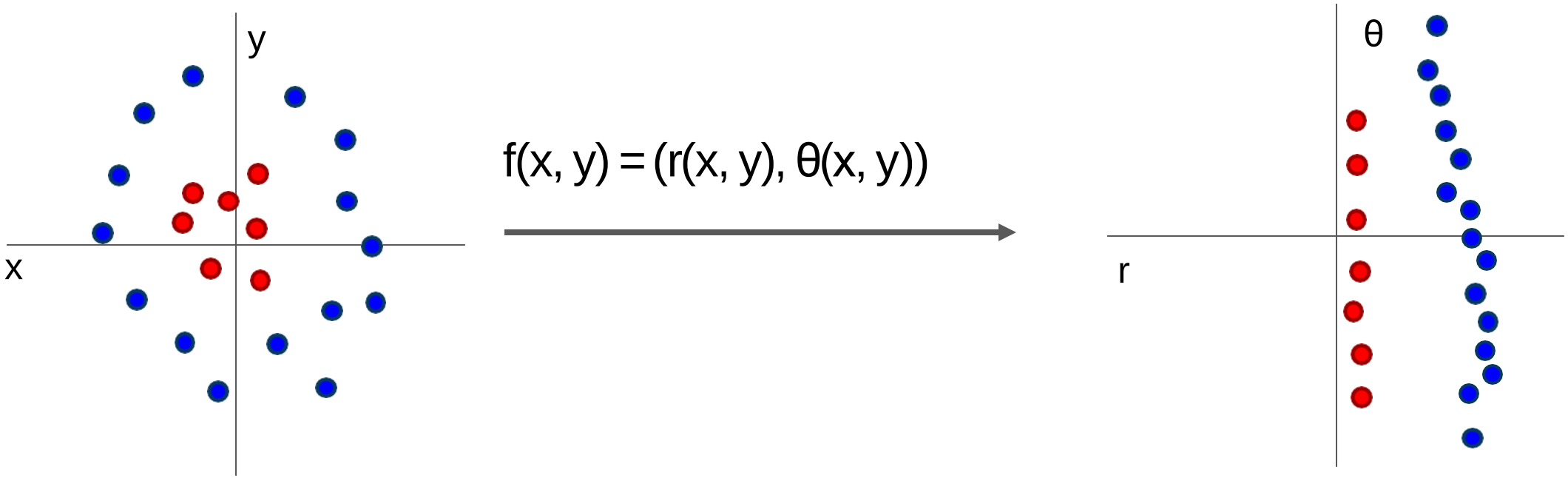
$$x \in \mathbb{R}^D, W_1 \in \mathbb{R}^{H \times D}, W_2 \in \mathbb{R}^{C \times H}$$

神经网络

南京大学 NANJING UNIVERSITY

- 为什么我们需要非线性?

问题：线性变换可以吗?



无法使用线性分类器分开蓝色和红色点数据

应用特征变换后，可以使用线性分类器分开不同颜色的点

神经网络

- 最简单的线性分类器

$$f = Wx$$

$$x \in \mathbb{R}^D, W \in \mathbb{R}^{C \times D}$$

- 2 层神经网络 → 全连接网络（FC）/多层感知机（MLP）

$$f = W_2 \max(0, W_1 x)$$

$$x \in \mathbb{R}^D, W_1 \in \mathbb{R}^{H \times D}, W_2 \in \mathbb{R}^{C \times H}$$

神经网络

- 最简单的线性分类器

$$f = Wx$$

$$x \in \mathbb{R}^D, W \in \mathbb{R}^{C \times D}$$

- 2 层神经网络 → 全连接网络（FC）/多层感知机（MLP）

$$f = W_2 \max(0, W_1 x)$$

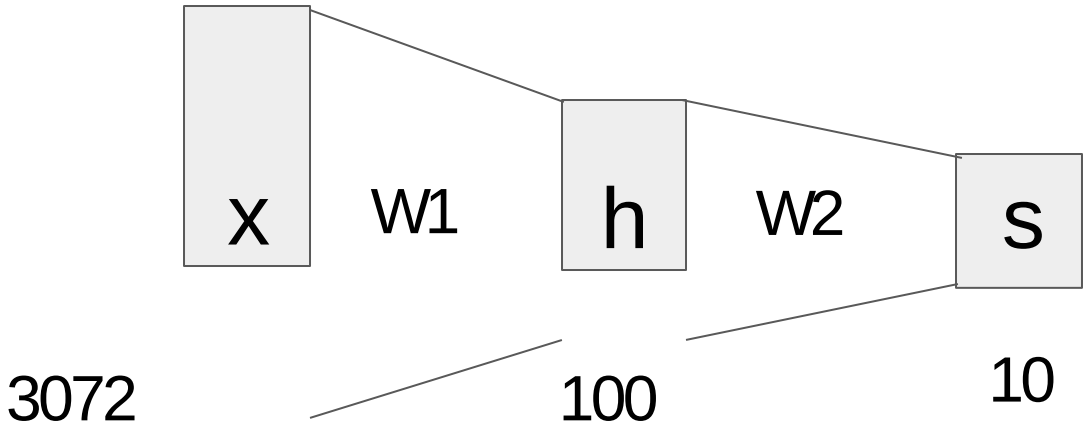
- 3 层神经网络

$$f = W_3 \max(0, W_2 \max(0, W_1 x))$$

$$x \in \mathbb{R}^D, W_1 \in \mathbb{R}^{H_1 \times D}, W_2 \in \mathbb{R}^{H_2 \times H_1}, W_3 \in \mathbb{R}^{C \times H_2}$$

神经网络

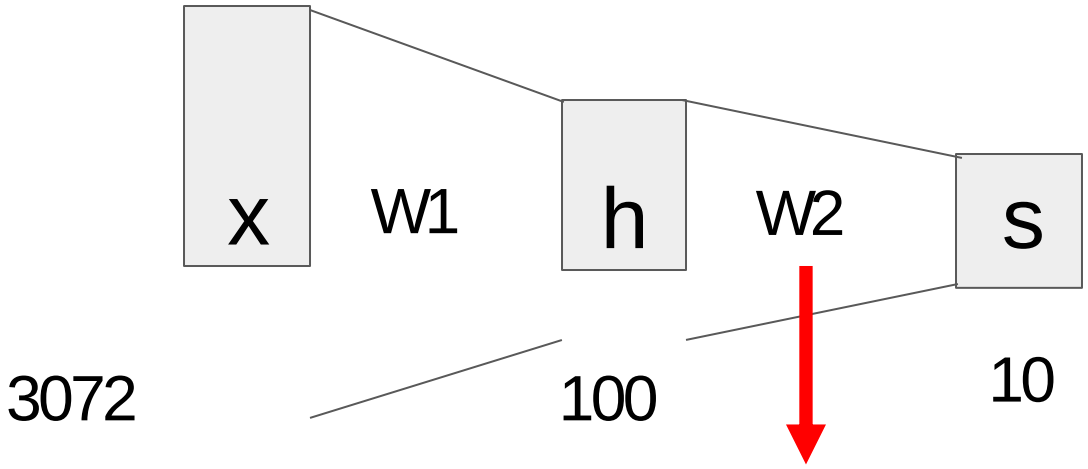
- **神经网络的分层计算** $f = W_2 \max(0, W_1 x)$



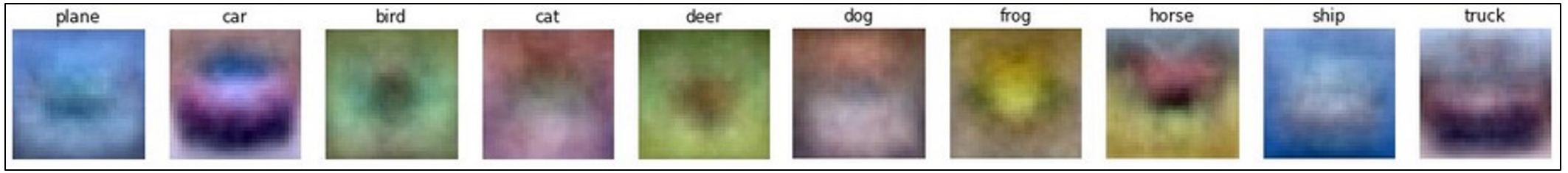
$$x \in \mathbb{R}^D, W_1 \in \mathbb{R}^{H \times D}, W_2 \in \mathbb{R}^{C \times H}$$

神经网络

- **神经网络的分层计算** $f = W_2 \max(0, W_1 x)$



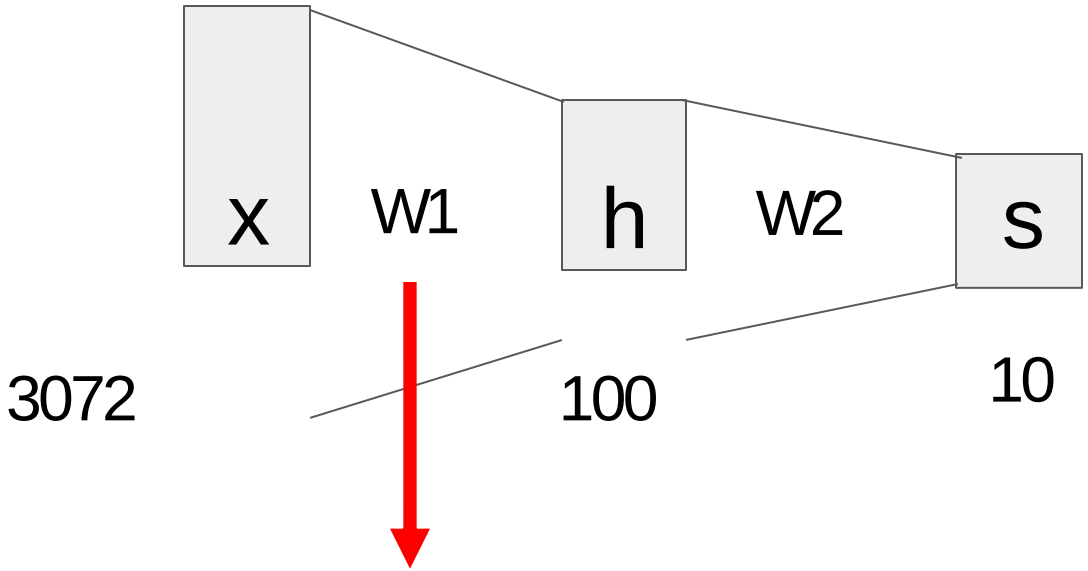
W2: 10个类别的视觉模板



神经网络

- 神经网络的分层计算

$$f = W_2 \max(0, W_1 x)$$



W1 是什么?

100 个类别共享的视觉模板

神经网络

- **神经网络的非线性**

$$f = W_2 \max(0, W_1 x)$$

神经网络

- **神经网络的非线性**

$$f = W_2 \max(0, W_1 x)$$

- **激活函数：**

$$\max(0, z)$$

神经网络

- 神经网络的非线性

$$f = W_2 \max(0, W_1 x)$$

- 激活函数：

$$\max(0, z)$$

- 如果我们去掉激活函数？

$$f = W_2 W_1 x$$

神经网络

- **神经网络的非线性** $f = W_2 \max(0, W_1 x)$
- **激活函数：**

$$\max(0, z)$$

- **如果我们去掉激活函数?**

$$f = W_2 W_1 x \qquad W_3 = W_2 W_1 \in \mathbb{R}^{C \times H}, f = W_3 x$$

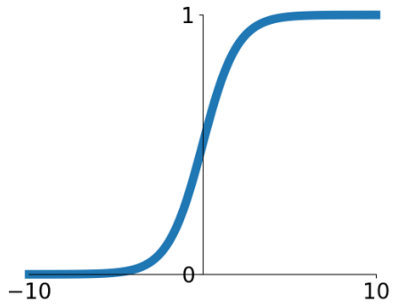
- **我们得到的还是线性分类器!**

神经网络

激活函数

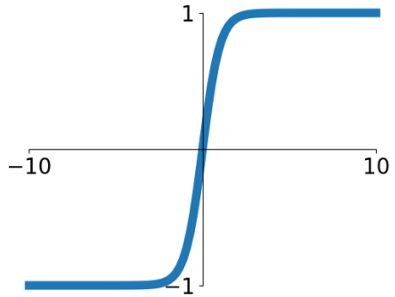
Sigmoid

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



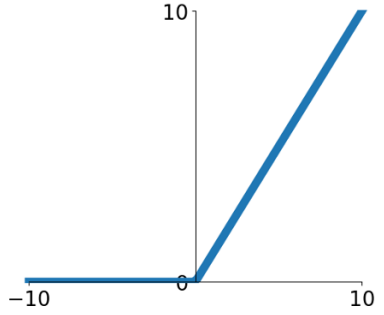
tanh

$$\tanh(x)$$



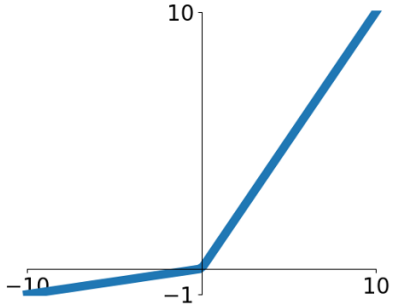
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

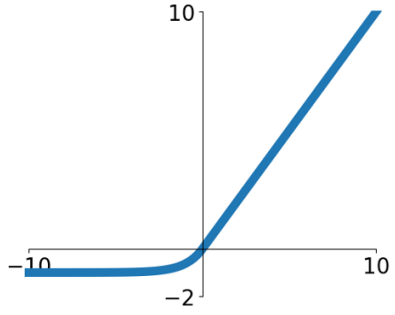


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

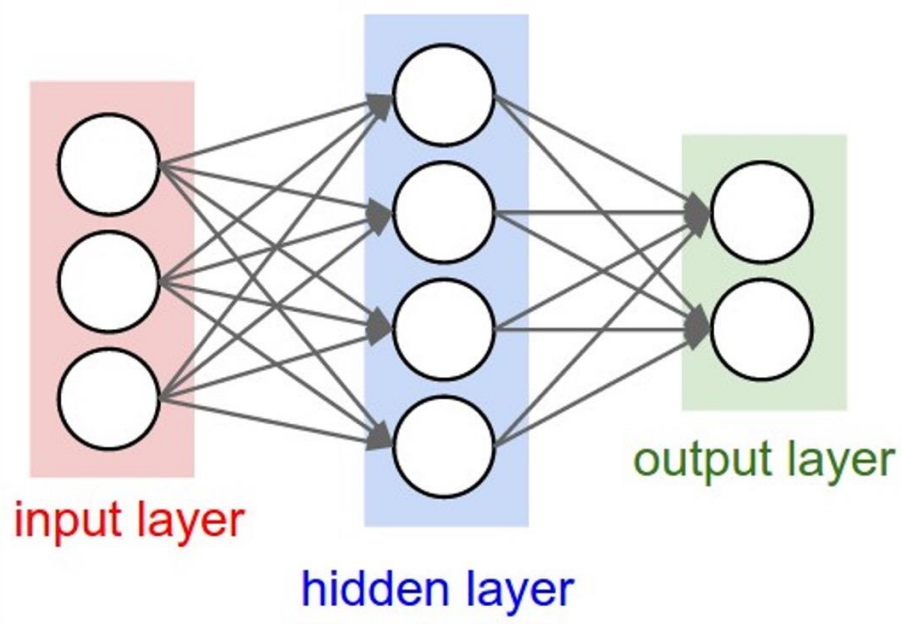
$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$

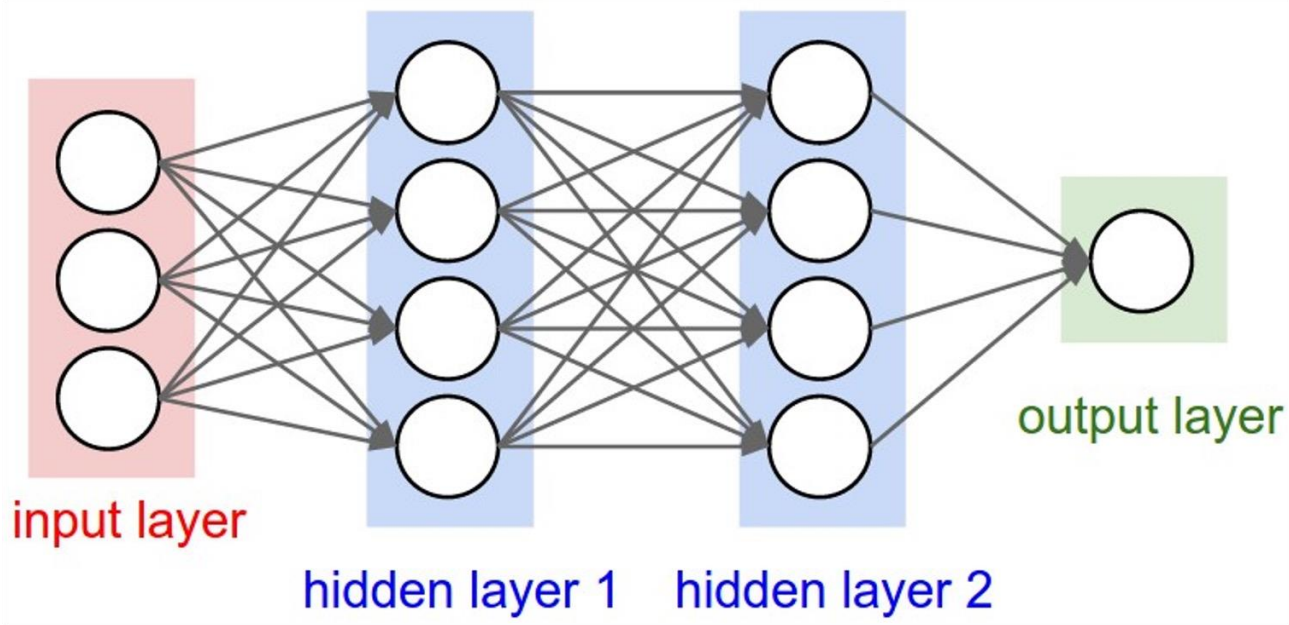


神经网络

南京大学 NANJING UNIVERSITY

神经网络架构



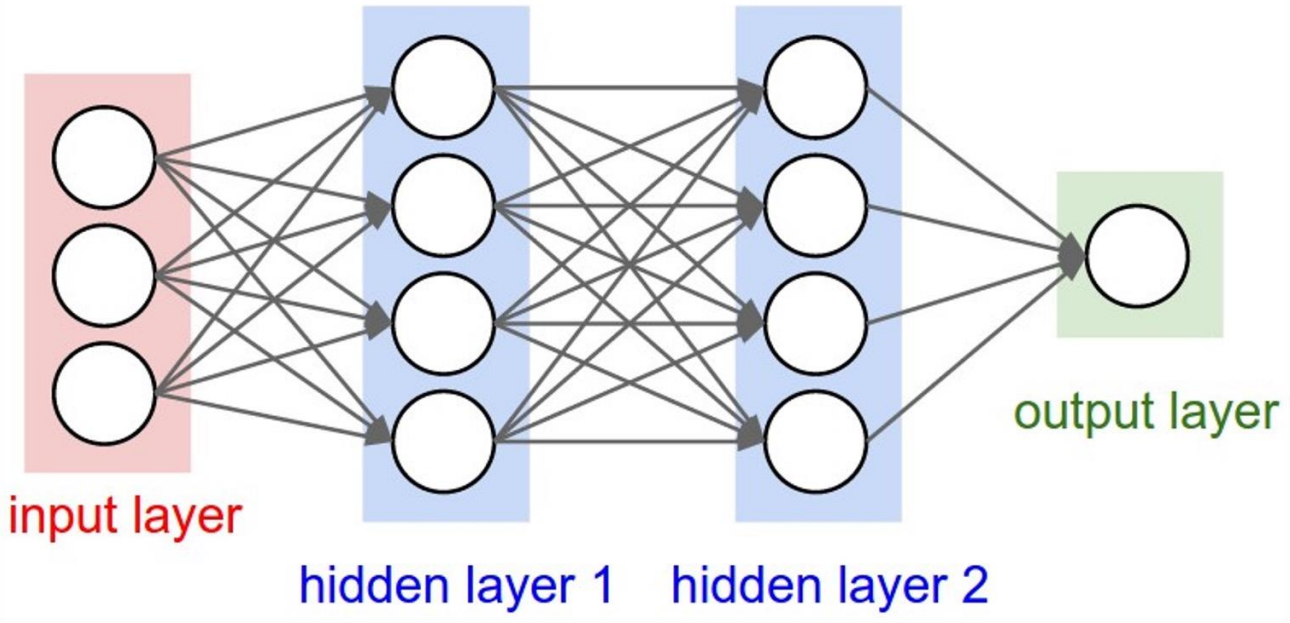


2 层神经网络
1 隐藏层神经网络

3 层神经网络
2 隐藏层神经网络

神经网络

神经网络前向传播




```
# forward-pass of a 3-layer neural network:
f = lambda x: 1.0/(1.0 + np.exp(-x)) # activation function (use sigmoid)
x = np.random.randn(3, 1) # random input vector of three numbers (3x1)
h1 = f(np.dot(W1, x) + b1) # calculate first hidden layer activations (4x1)
h2 = f(np.dot(W2, h1) + b2) # calculate second hidden layer activations (4x1)
out = np.dot(W3, h2) + b3 # output neuron (1x1)
```

神经网络

南京大学 NANJING UNIVERSITY

- 训练 2 层神经网络

```python
import numpy as np
from numpy.random import randn

N, D_in, H, D_out = 64, 1000, 100, 10
x, y = randn(N, D_in), randn(N, D_out)
w1, w2 = randn(D_in, H), randn(H, D_out)

for t in range(2000):
  h = 1 / (1 + np.exp(-x.dot(w1)))
  y_pred = h.dot(w2)
  loss = np.square(y_pred - y).sum()
  print(t, loss)

  grad_y_pred = 2.0 * (y_pred - y)
  grad_w2 = h.T.dot(grad_y_pred)
  grad_h = grad_y_pred.dot(w2.T)
  grad_w1 = x.T.dot(grad_h * h * (1 - h))

  w1 -= 1e-4 * grad_w1
  w2 -= 1e-4 * grad_w2
```

神经网络

南京大学 NANJING UNIVERSITY

- 训练 2 层神经网络

定义神经网络

```
import numpy as np
from numpy.random import randn

N, D_in, H, D_out = 64, 1000, 100, 10
x, y = randn(N, D_in), randn(N, D_out)
w1, w2 = randn(D_in, H), randn(H, D_out)

for t in range(2000):
  h = 1 / (1 + np.exp(-x.dot(w1)))
  y_pred = h.dot(w2)
  loss = np.square(y_pred - y).sum()
  print(t, loss)

  grad_y_pred = 2.0 * (y_pred - y)
  grad_w2 = h.T.dot(grad_y_pred)
  grad_h = grad_y_pred.dot(w2.T)
  grad_w1 = x.T.dot(grad_h * h * (1 - h))

  w1 -= 1e-4 * grad_w1
  w2 -= 1e-4 * grad_w2
```

神经网络

南京大学 NANJING UNIVERSITY

- 训练 2 层神经网络

前向传播

```
1  import numpy as np
2  from numpy.random import randn
3
4  N, D_in, H, D_out = 64, 1000, 100, 10
5  x, y = randn(N, D_in), randn(N, D_out)
6  w1, w2 = randn(D_in, H), randn(H, D_out)
7
8  for t in range(2000):
9    h = 1 / (1 + np.exp(-x.dot(w1)))
10   y_pred = h.dot(w2)
11   loss = np.square(y_pred - y).sum()
12   print(t, loss)
13
14   grad_y_pred = 2.0 * (y_pred - y)
15   grad_w2 = h.T.dot(grad_y_pred)
16   grad_h = grad_y_pred.dot(w2.T)
17   grad_w1 = x.T.dot(grad_h * h * (1 - h))
18
19   w1 -= 1e-4 * grad_w1
20   w2 -= 1e-4 * grad_w2
```

神经网络

南京大学 NANJING UNIVERSITY

- 训练 2 层神经网络

```
1  import numpy as np
2  from numpy.random import randn
3
4  N, D_in, H, D_out = 64, 1000, 100, 10
5  x, y = randn(N, D_in), randn(N, D_out)
6  w1, w2 = randn(D_in, H), randn(H, D_out)
7
8  for t in range(2000):
9    h = 1 / (1 + np.exp(-x.dot(w1)))
10   y_pred = h.dot(w2)
11   loss = np.square(y_pred - y).sum()
12   print(t, loss)
13
14   grad_y_pred = 2.0 * (y_pred - y)
15   grad_w2 = h.T.dot(grad_y_pred)
16   grad_h = grad_y_pred.dot(w2.T)
17   grad_w1 = x.T.dot(grad_h * h * (1 - h))
18
19   w1 -= 1e-4 * grad_w1
20   w2 -= 1e-4 * grad_w2
```

计算梯度

神经网络

南京大学 NANJING UNIVERSITY

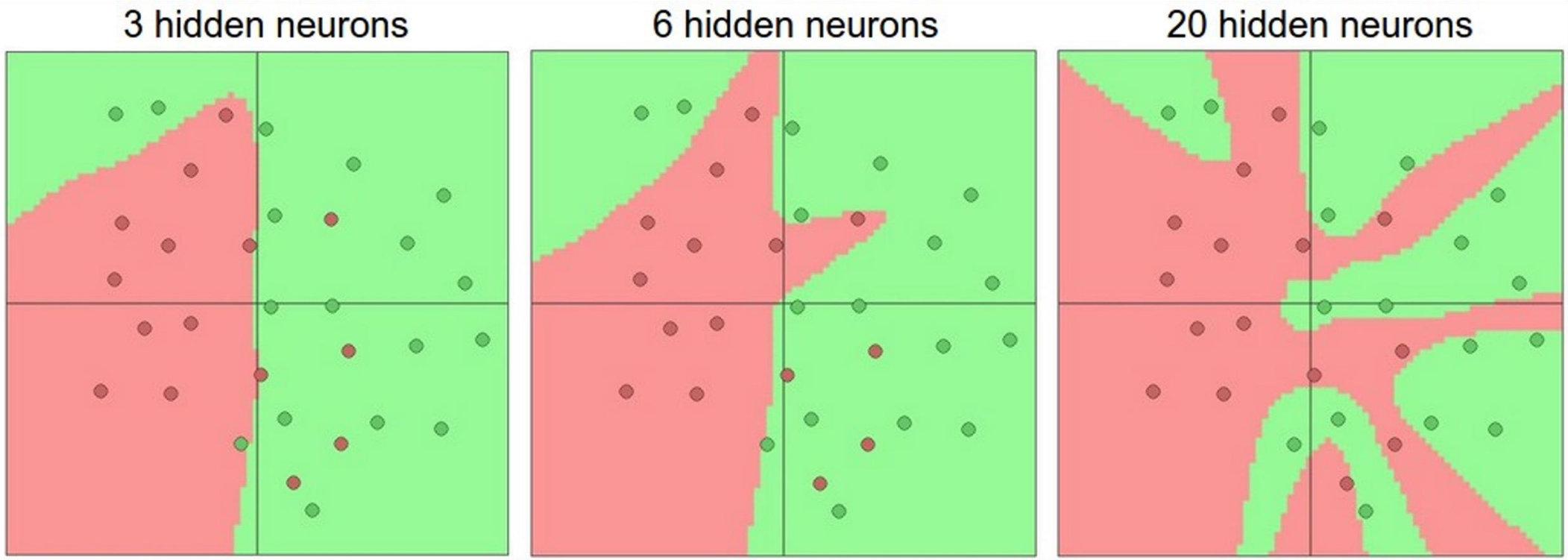
- 训练 2 层神经网络

```
1  import numpy as np
2  from numpy.random import randn
3
4  N, D_in, H, D_out = 64, 1000, 100, 10
5  x, y = randn(N, D_in), randn(N, D_out)
6  w1, w2 = randn(D_in, H), randn(H, D_out)
7
8  for t in range(2000):
9    h = 1 / (1 + np.exp(-x.dot(w1)))
10   y_pred = h.dot(w2)
11   loss = np.square(y_pred - y).sum()
12   print(t, loss)
13
14   grad_y_pred = 2.0 * (y_pred - y)
15   grad_w2 = h.T.dot(grad_y_pred)
16   grad_h = grad_y_pred.dot(w2.T)
17   grad_w1 = x.T.dot(grad_h * h * (1 - h))
18
19   w1 -= 1e-4 * grad_w1
20   w2 -= 1e-4 * grad_w2
```

梯度下降，更新参数

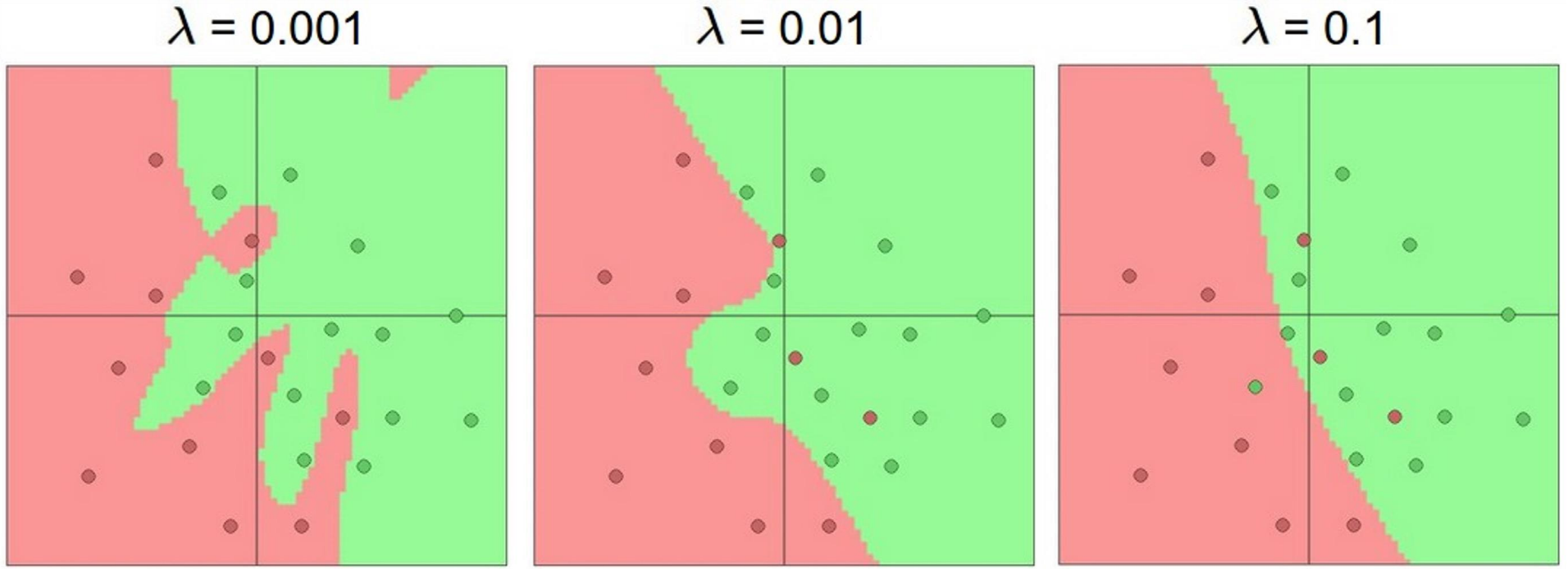
神经网络

- 神经网络的大小



更多的神经元
更大的模型容量

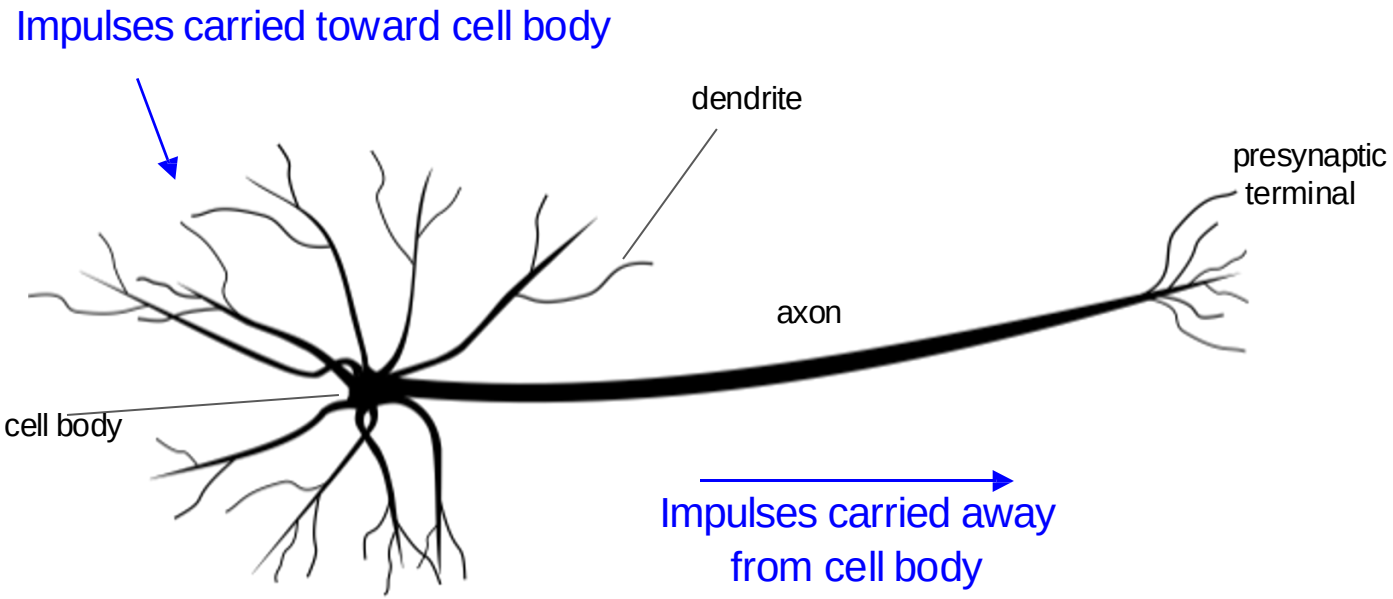
- 更大的神经网络使用更强的正则化进行约束



$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

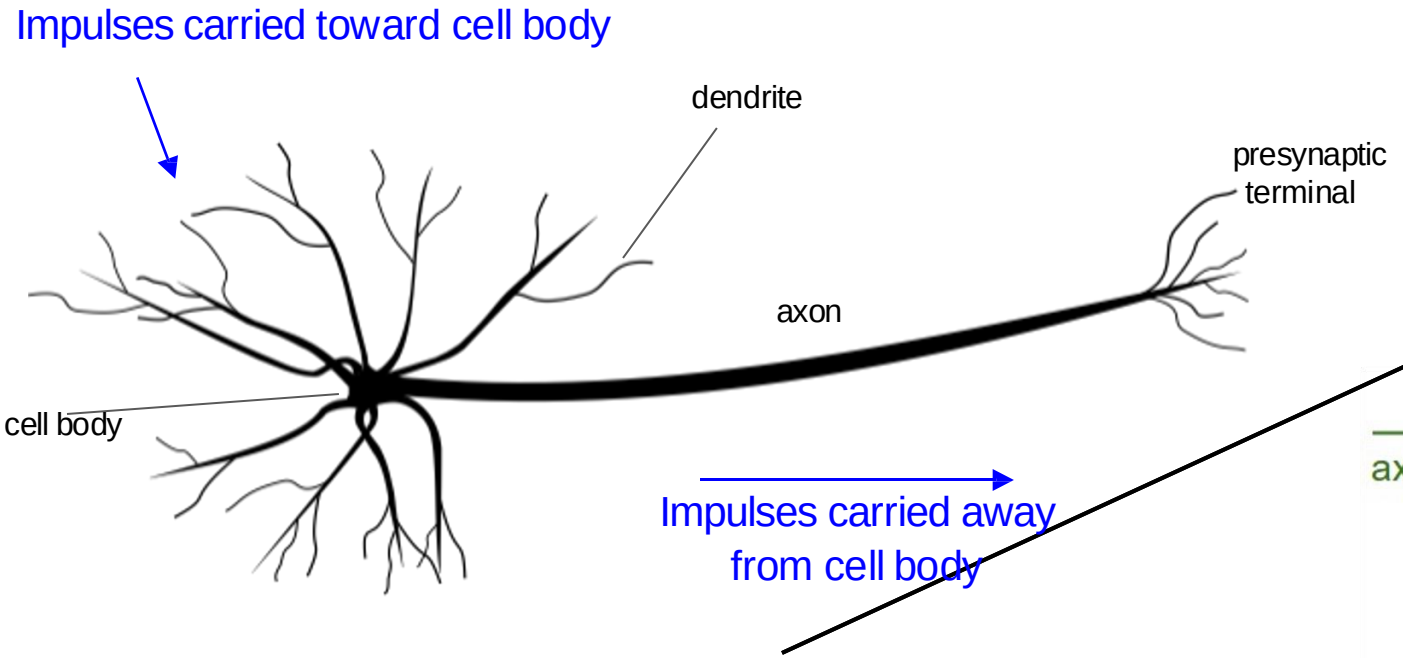
神经网络

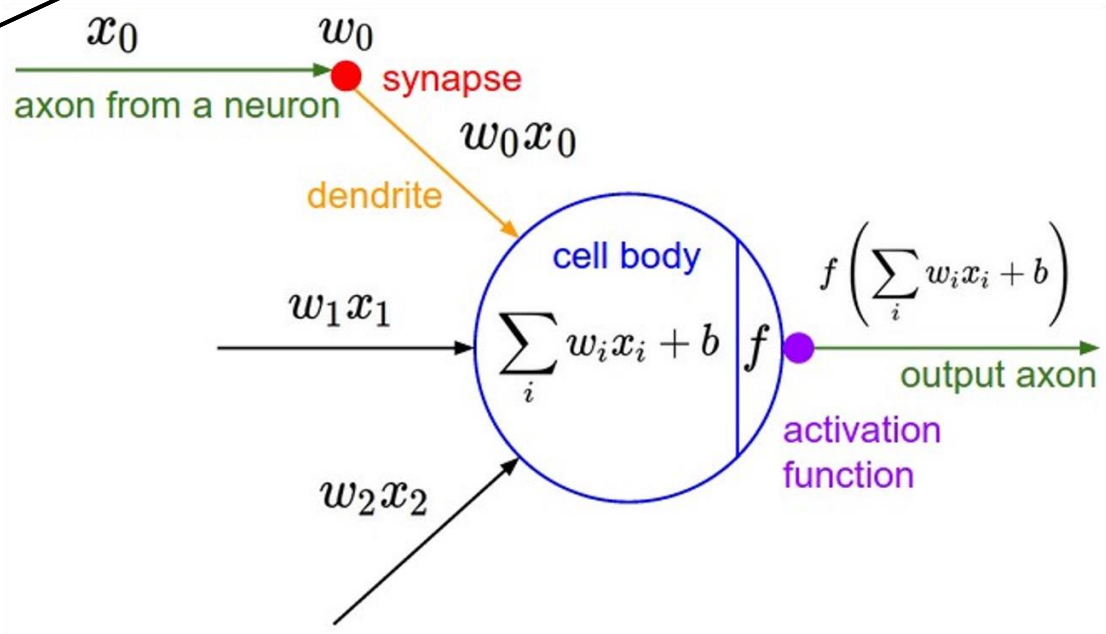
神经网络的神经元



神经网络

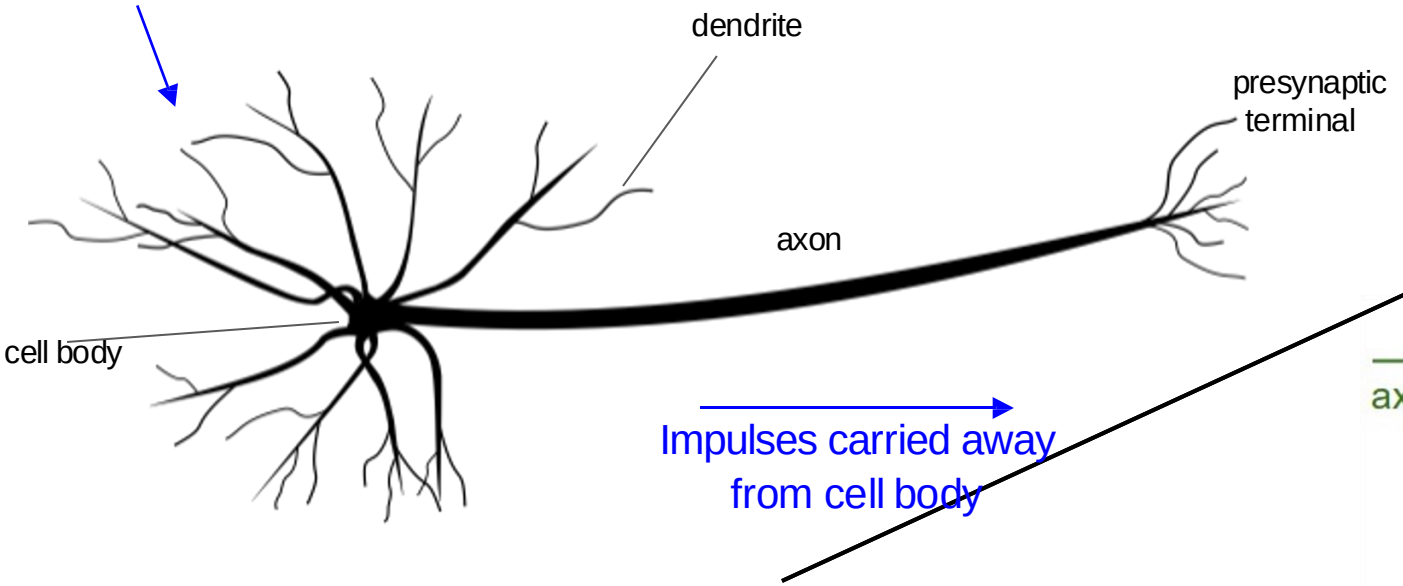
神经网络的神经元

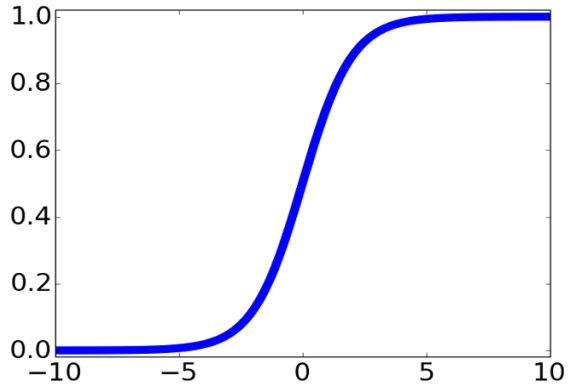




神经网络

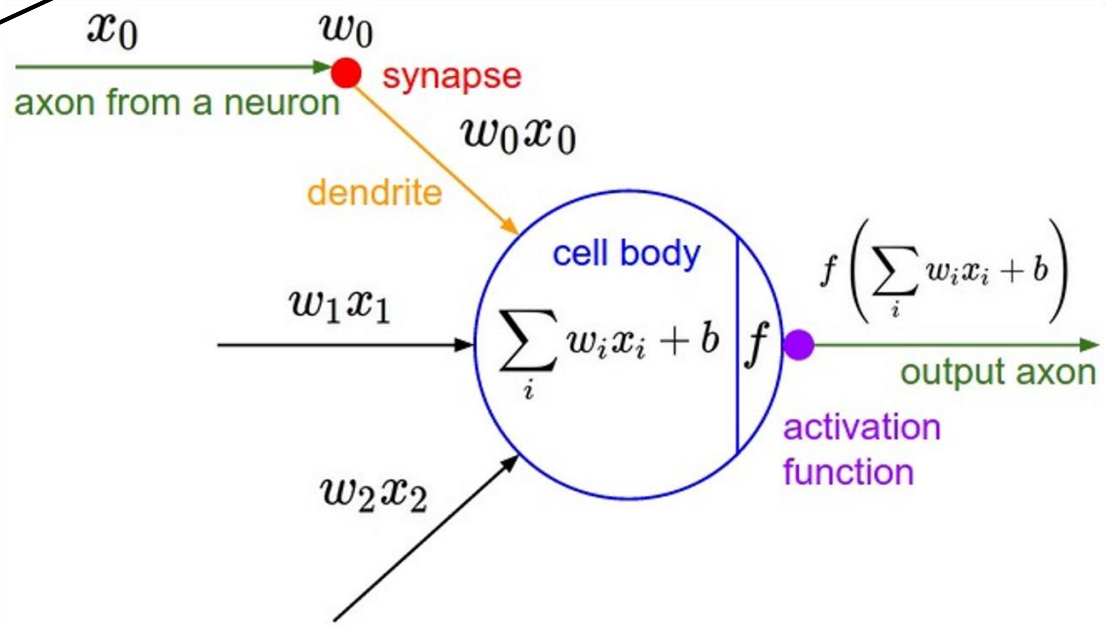
神经网络的神经元





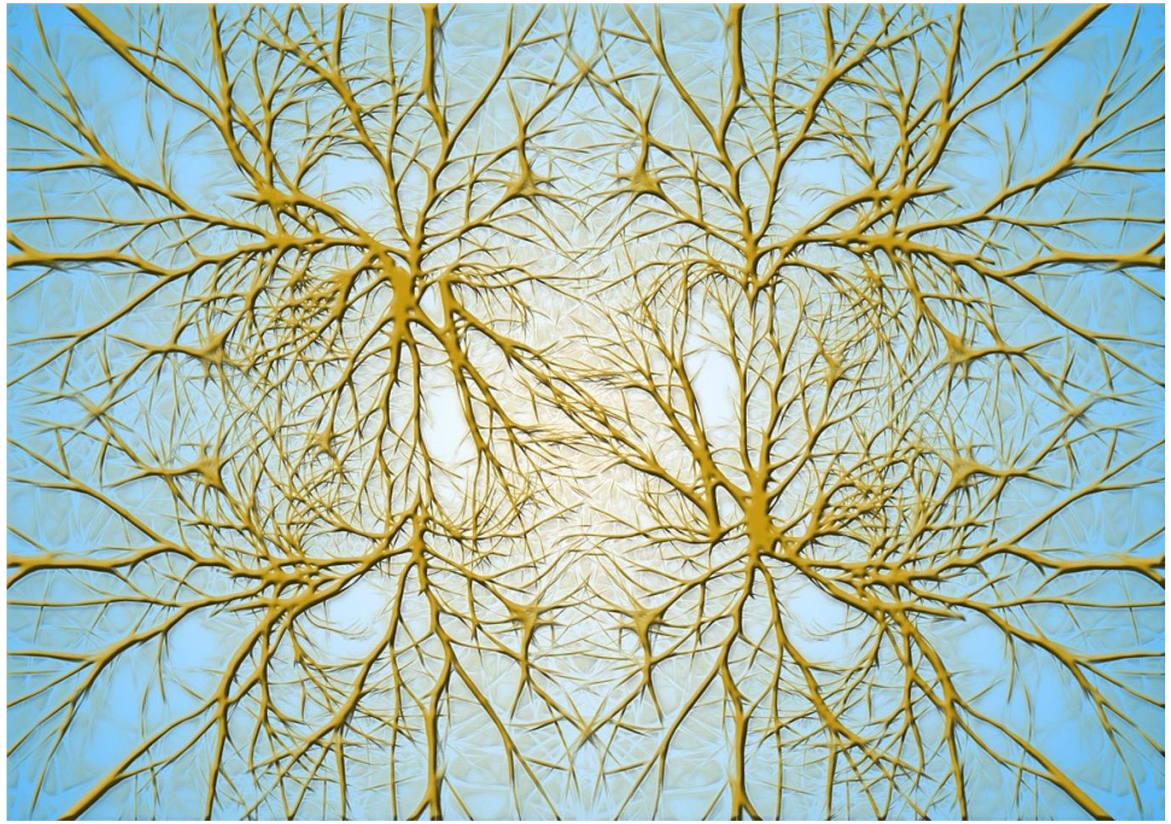
sigmoid 激活函数

$$\frac{1}{1+e^{-x}}$$

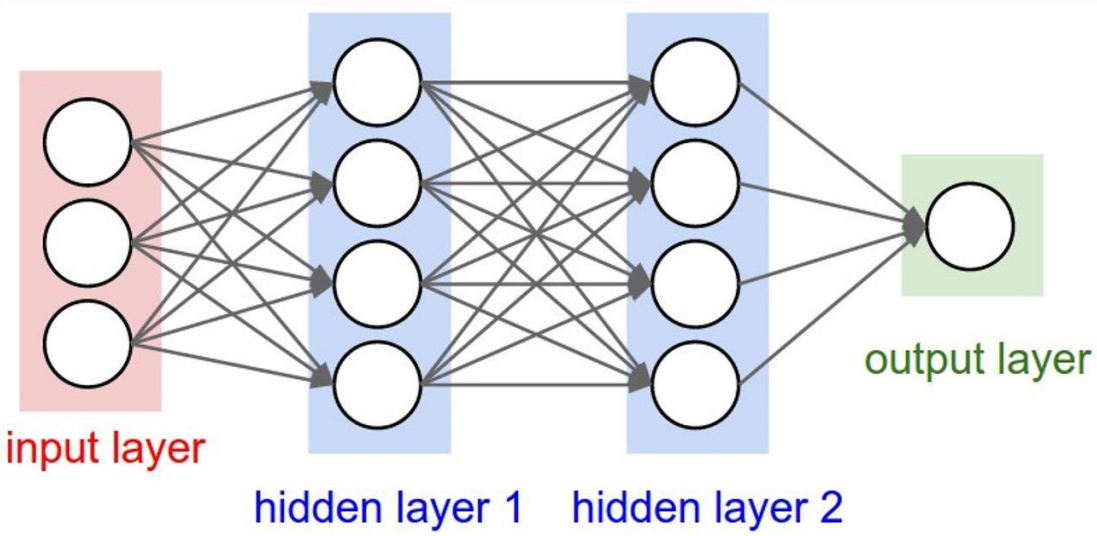


神经网络

神经网络的神经元



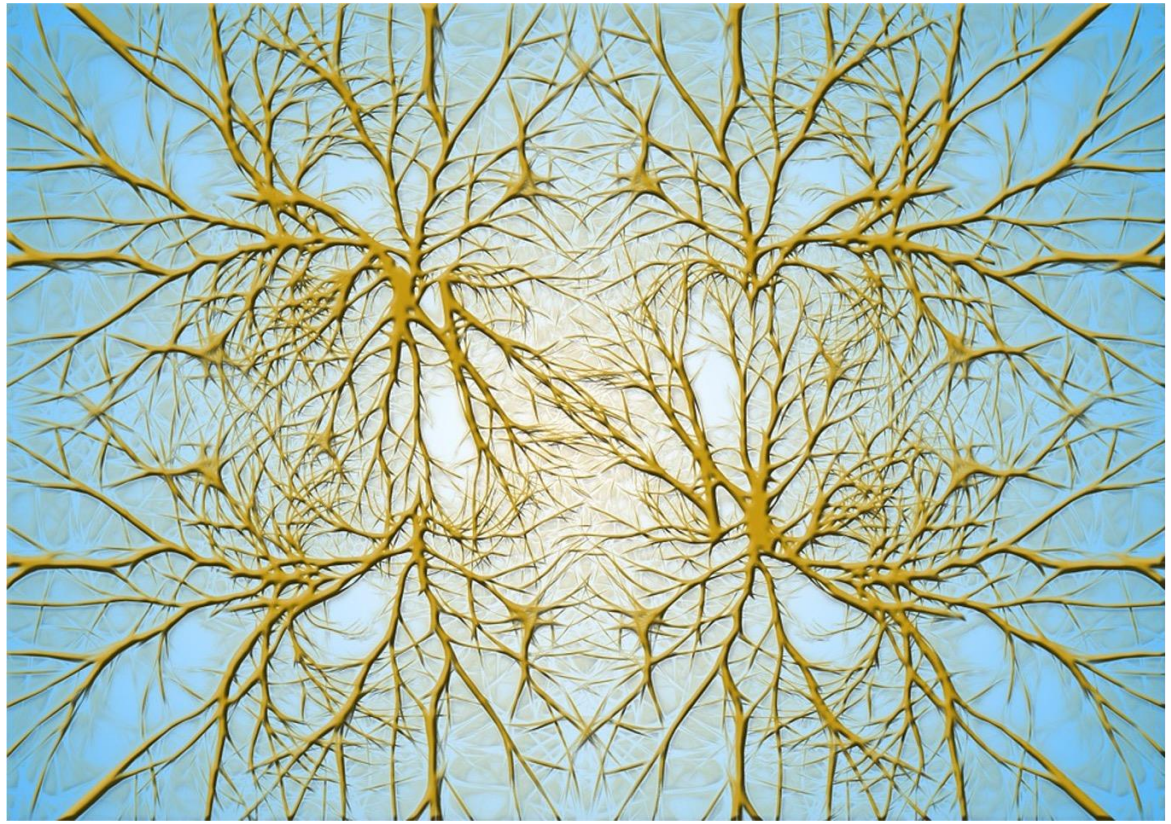
生物神经元：复杂的连接模式



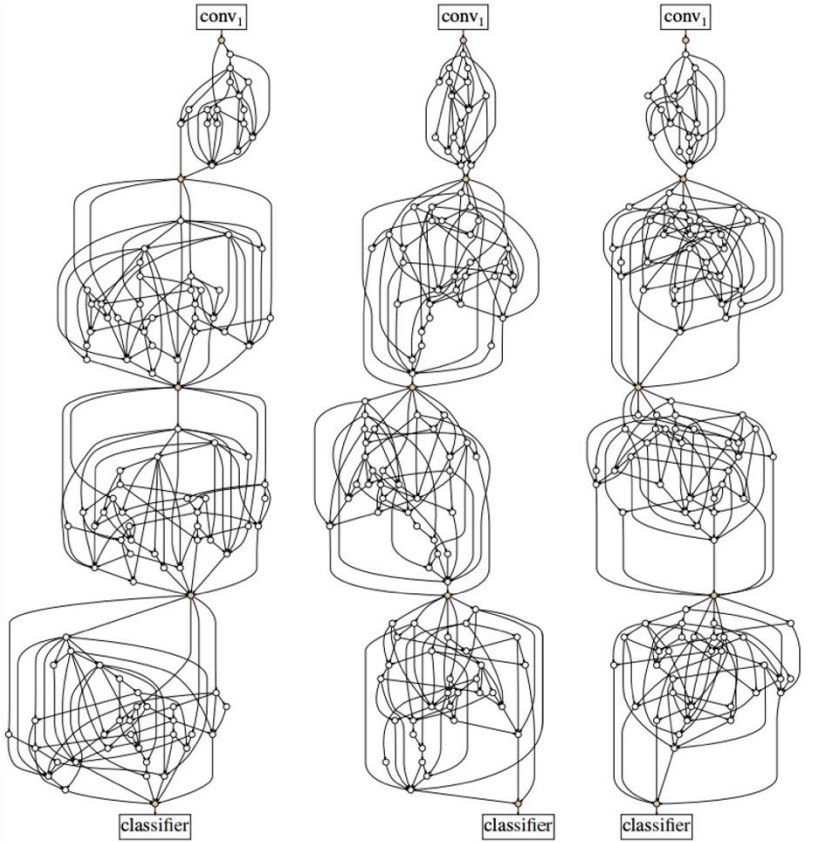
神经网络中的神经元：
组织成规则层以提高计算效率

神经网络

神经网络的神经元



生物神经元：复杂的连接模式



但具有随机连接的神经网络
也可以工作

神经网络

南京大学 NANJING UNIVERSITY

神经网络的权重参数更新

$$s = f(x; W_1, W_2) = W_2 \max(0, W_1 x)$$ 非线性预测函数

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$ SVM Loss on predictions

$$R(W) = \sum_k W_k^2$$ 正则化

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + \lambda R(W_1) + \lambda R(W_2)$$ 总损失: data loss + regularization

神经网络

- **神经网络的权重参数更新**

$$s = f(x; W_1, W_2) = W_2 \max(0, W_1 x)$$ 非线性预测函数

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$ SVM Loss on predictions

$$R(W) = \sum_k W_k^2$$ 正则化

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + \lambda R(W_1) + \lambda R(W_2)$$ 总损失: data loss + regularization

- **如果我们能够计算 $\frac{\partial L}{\partial W_1}, \frac{\partial L}{\partial W_2}$ ，我们就可以学习参数 W**

神经网络

南京大学 NANJING UNIVERSITY

- **想法一：推导** $\nabla_W L$

$$s = f(x; W) = Wx$$

$$L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

$$= \sum_{j \neq y_i} \max(0, W_{j,:} \cdot x + W_{y_i,:} \cdot x + 1)$$

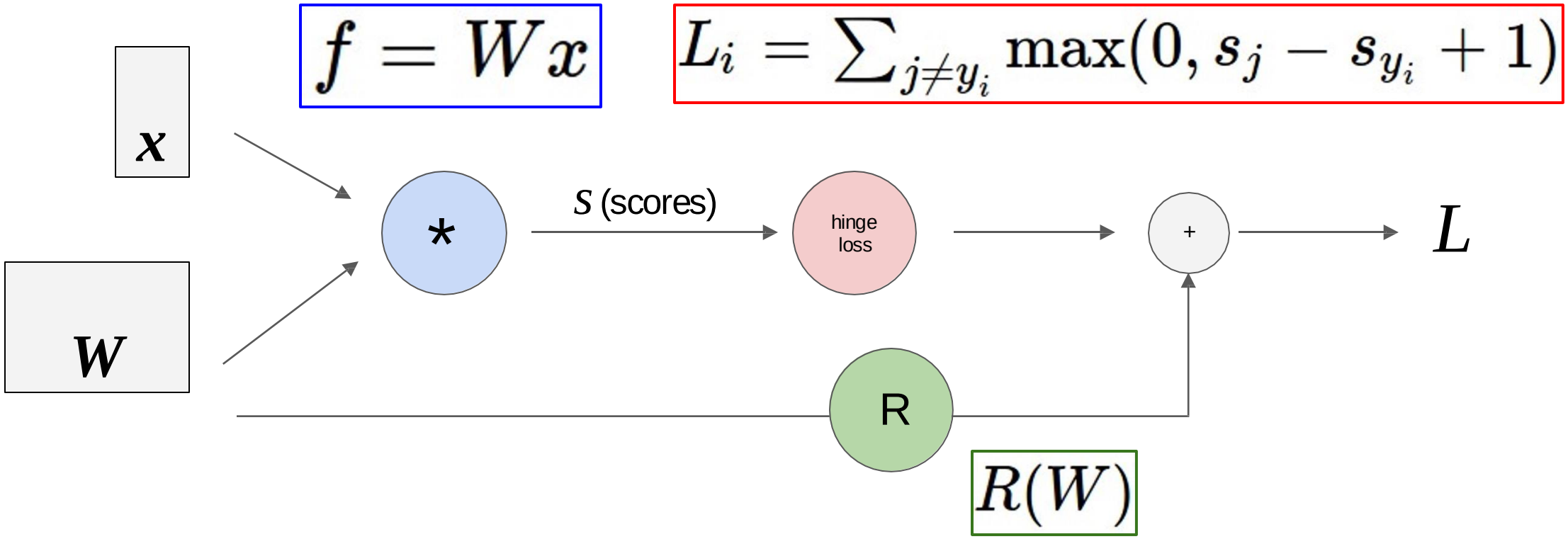
$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + \lambda \sum_k W_k^2$$

$$= \frac{1}{N} \sum_{i=1}^{N} \sum_{j \neq y_i} \max(0, W_{j,:} \cdot x + W_{y_i,:} \cdot x + 1) + \lambda \sum_k W_k^2$$

$$\nabla_W L = \nabla_W \left(\frac{1}{N} \sum_{i=1}^{N} \sum_{j \neq y_i} \max(0, W_{j,:} \cdot x + W_{y_i,:} \cdot x + 1) + \lambda \sum_k W_k^2 \right)$$

- **问题一：推导非常繁琐**
- **问题二：需要为每个模型和损失函数进行推导**
- **问题三：对于复杂的模型不可行！**

- 更好的想法：计算图+反向传播



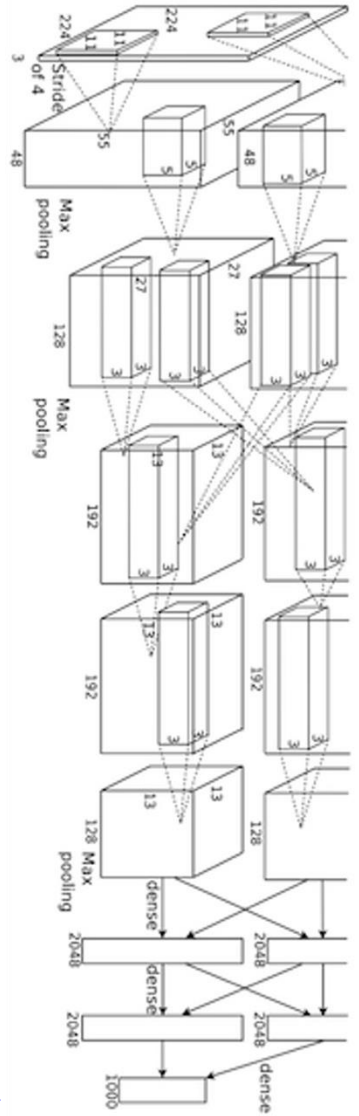
神经网络

- 例子：AlexNet

input image

weights

loss

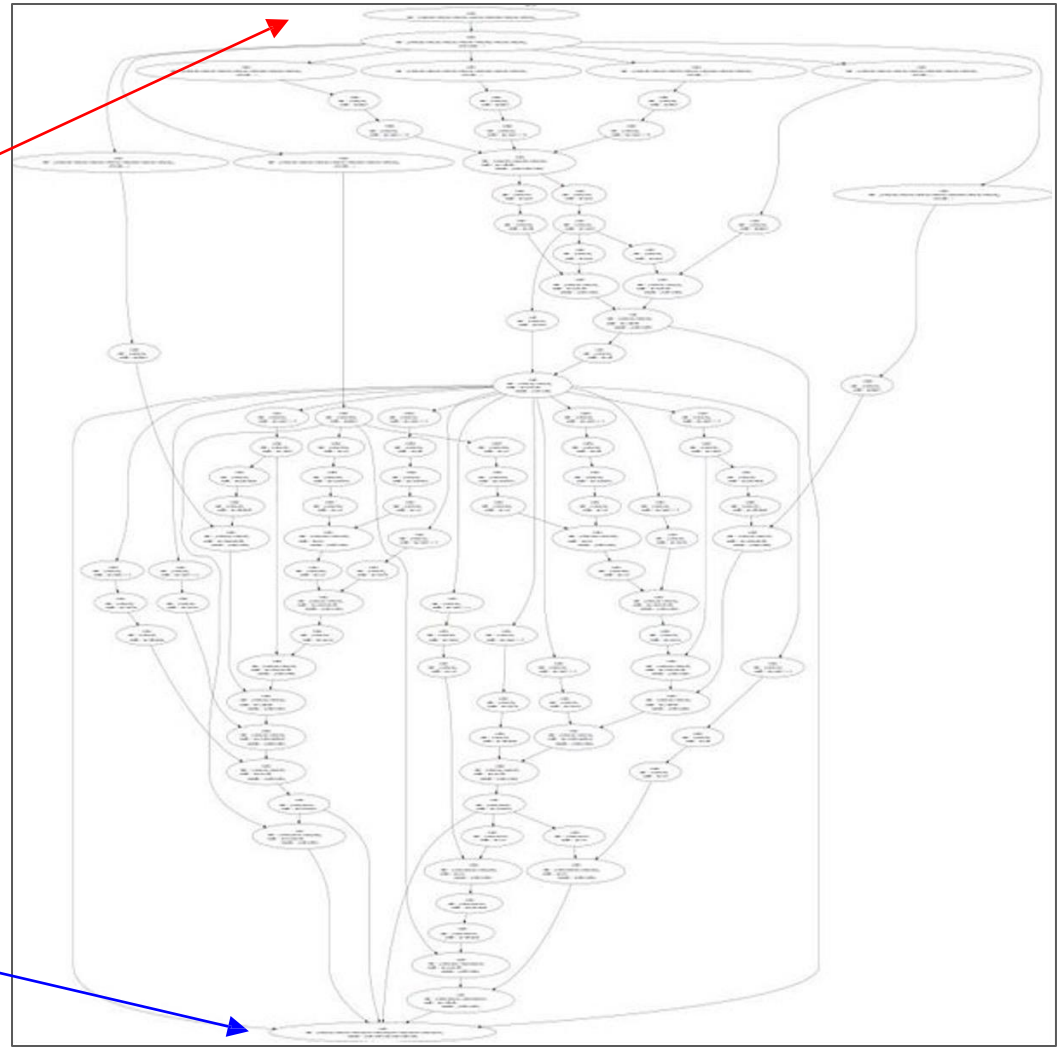


神经网络

- 非常复杂的神经网络

input image

loss



大　　纲

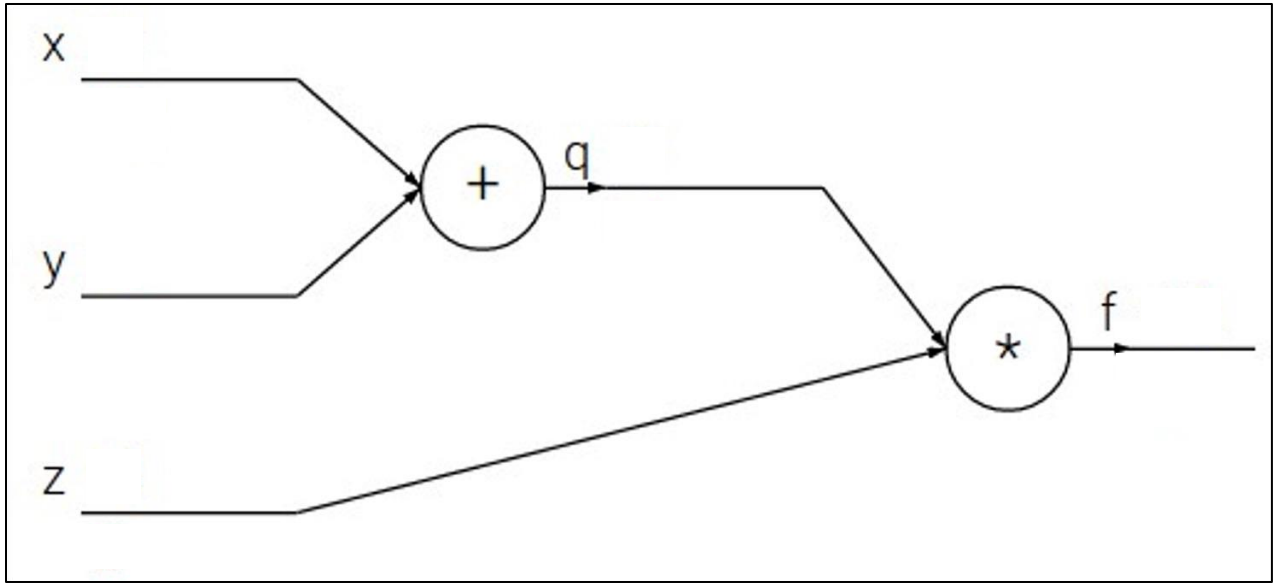
- 神经网络
- 反向传播

反向传播

南京大学 NANJING UNIVERSITY

- 反向传播例子

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

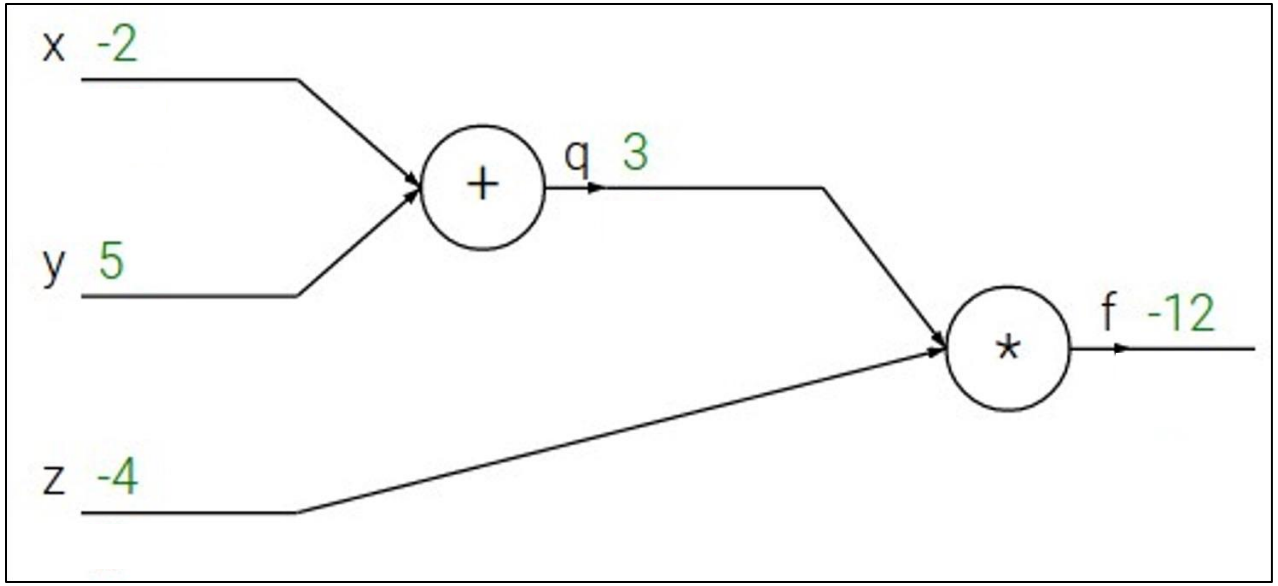


反向传播

- **反向传播例子**

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

e.g. x = -2, y = 5, z = -4



反向传播

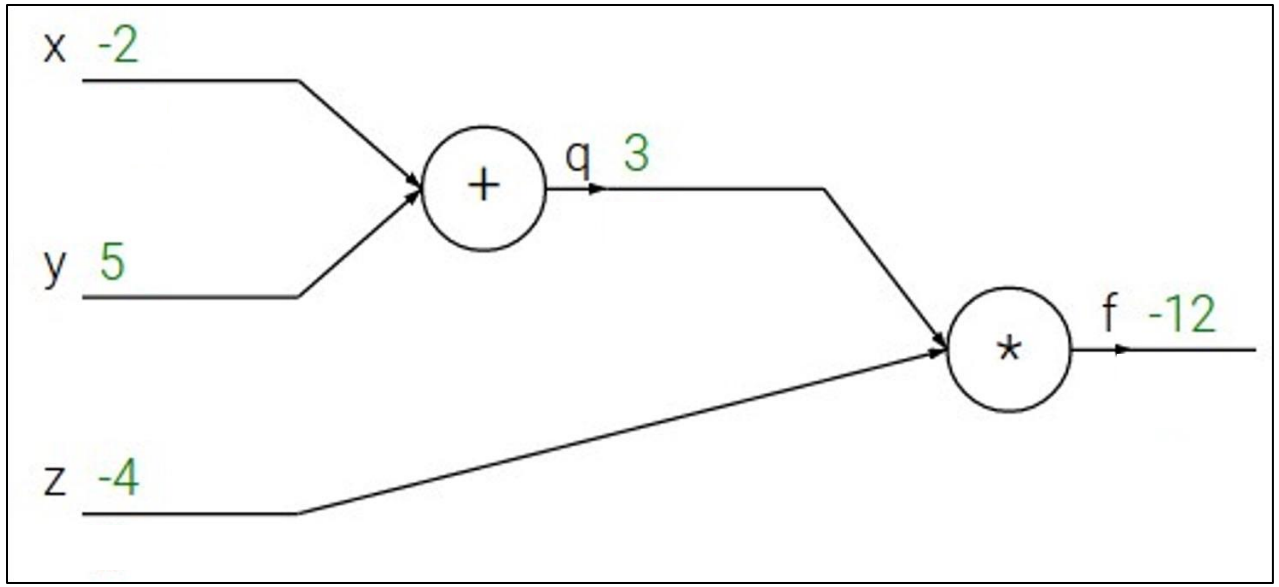
南京大学 NANJING UNIVERSITY

- **反向传播例子**

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

e.g. x = -2, y = 5, z = -4

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



反向传播

南京大学 NANJING UNIVERSITY

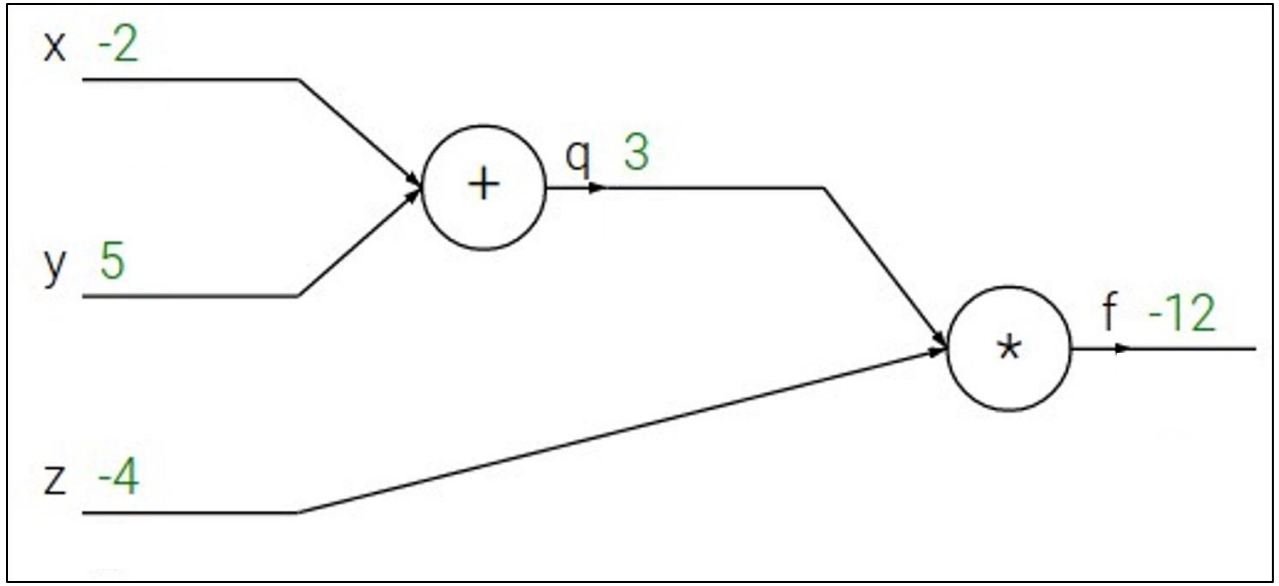
反向传播例子

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

e.g. x = -2, y = 5, z = -4

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

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



反向传播

南京大学 NANJING UNIVERSITY

- **反向传播例子**

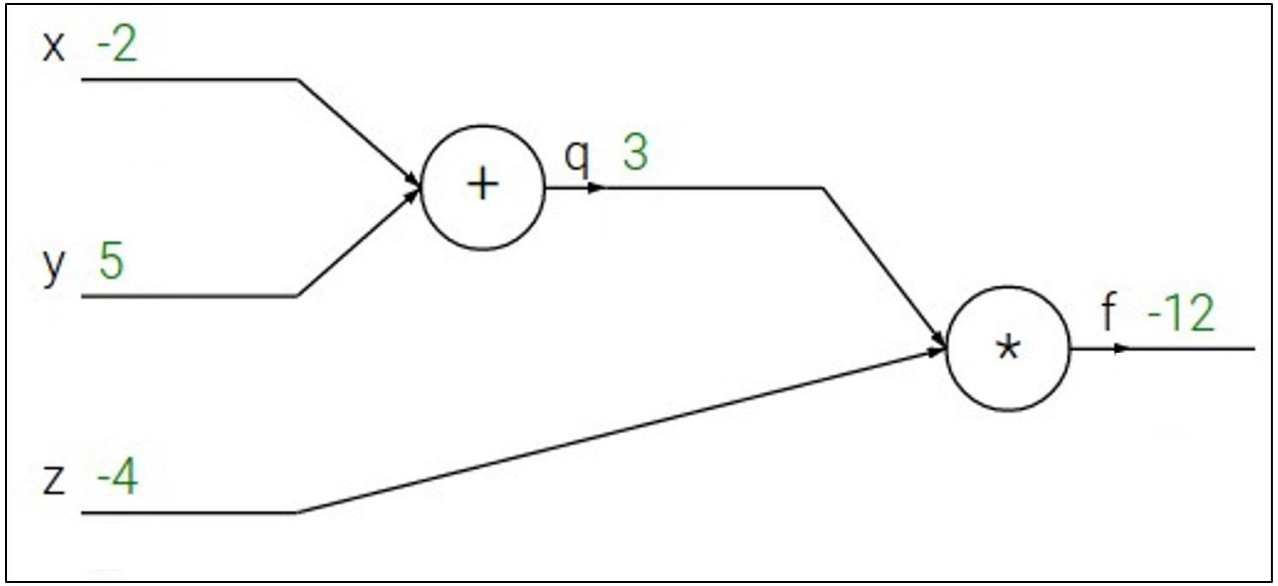
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



反向传播

南京大学 NANJING UNIVERSITY

- **反向传播例子**

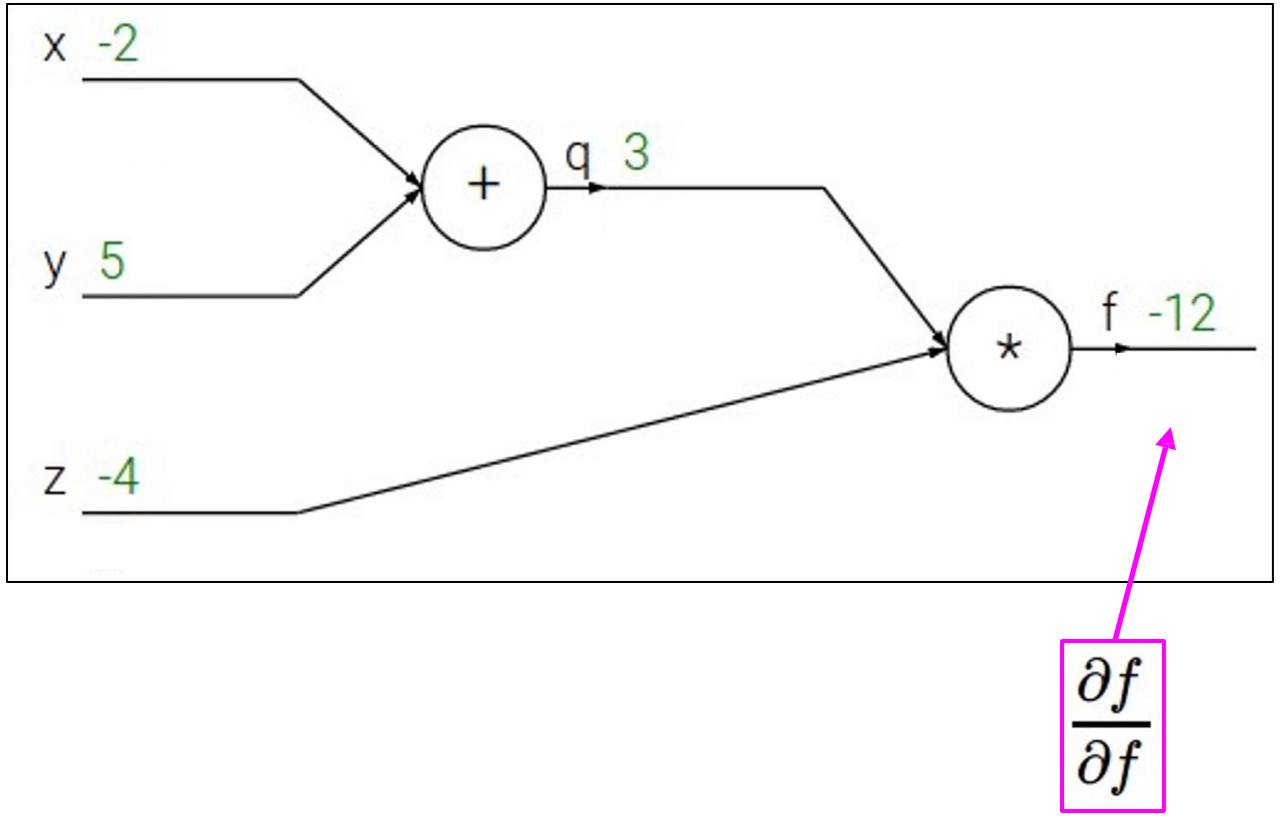
$$f(x,y,z)=(x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



反向传播

南京大学 NANJING UNIVERSITY

- **反向传播例子**

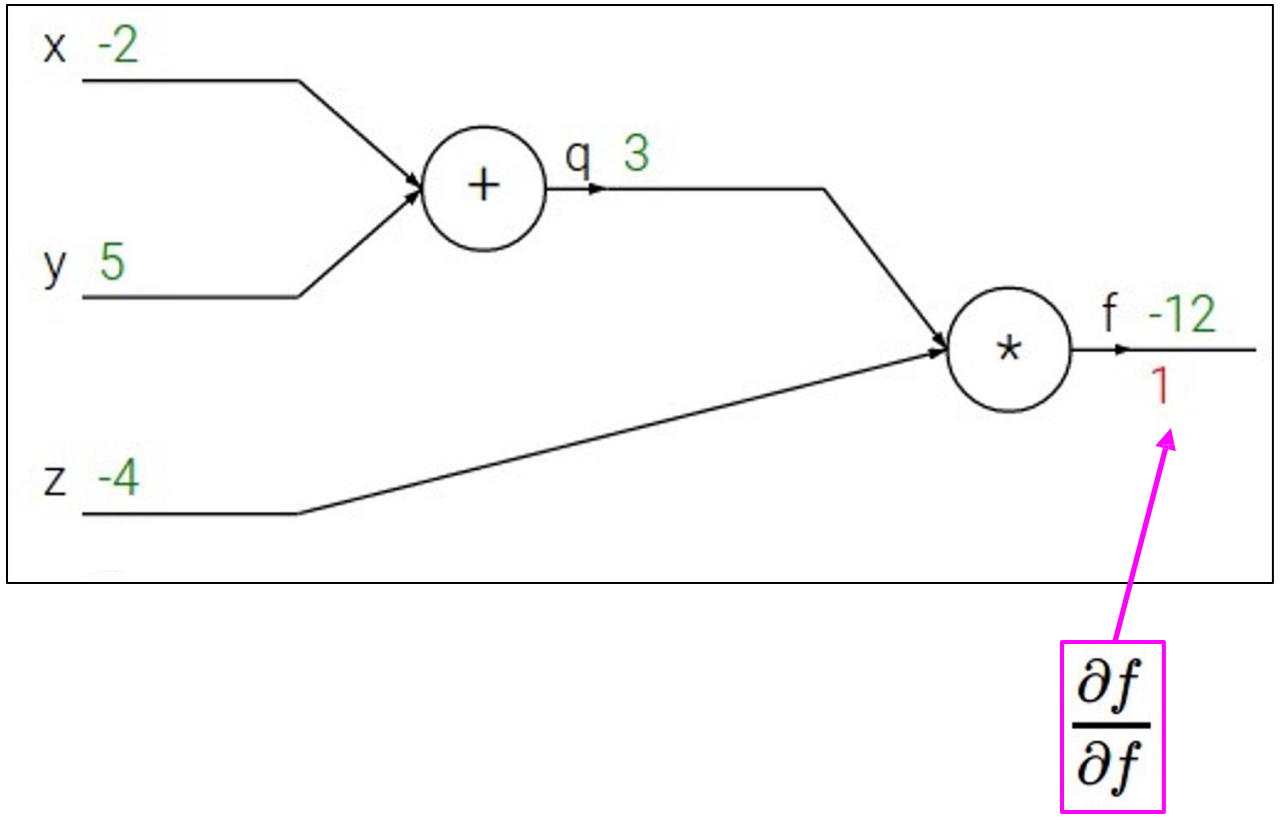
$$f(x, y, z) = (x + y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



反向传播

南京大学 NANJING UNIVERSITY

- **反向传播例子**

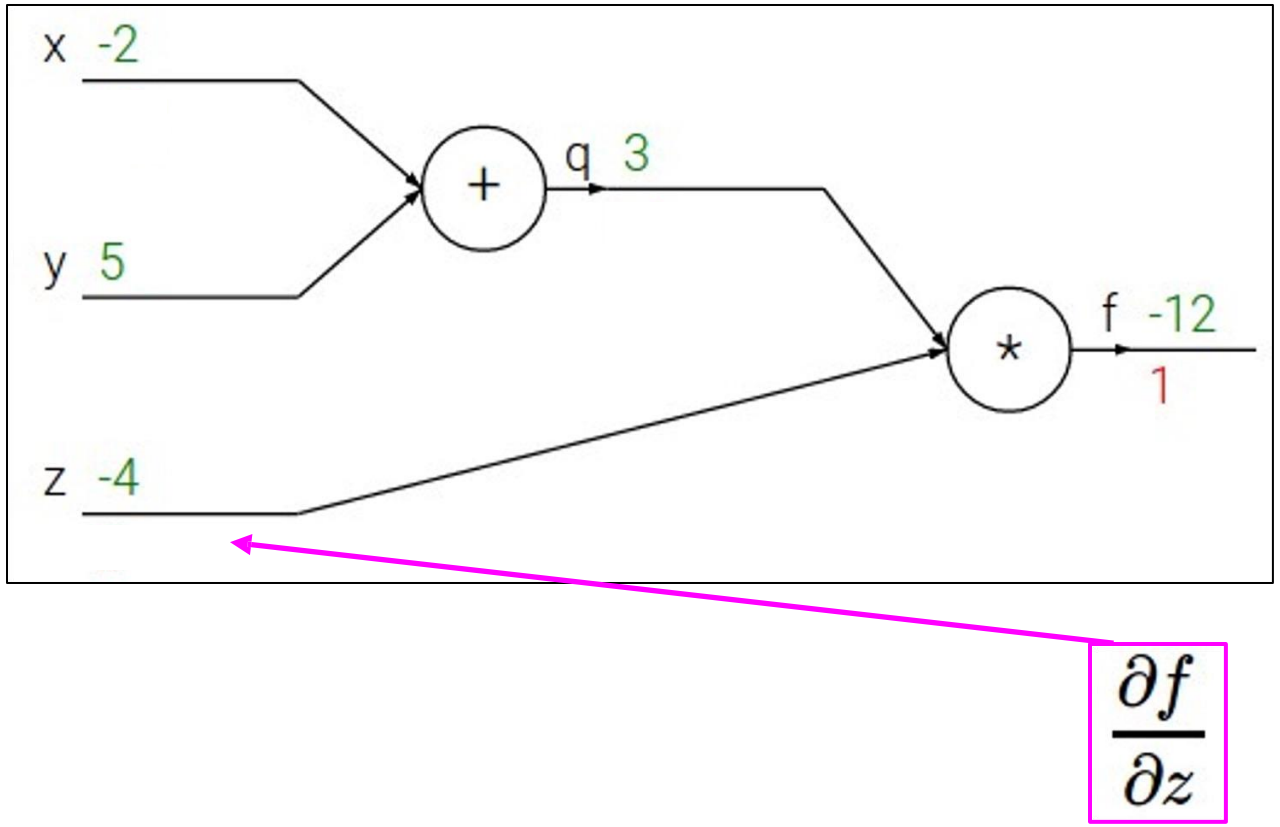
$$f(x,y,z)=(x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



反向传播

南京大学 NANJING UNIVERSITY

- **反向传播例子**

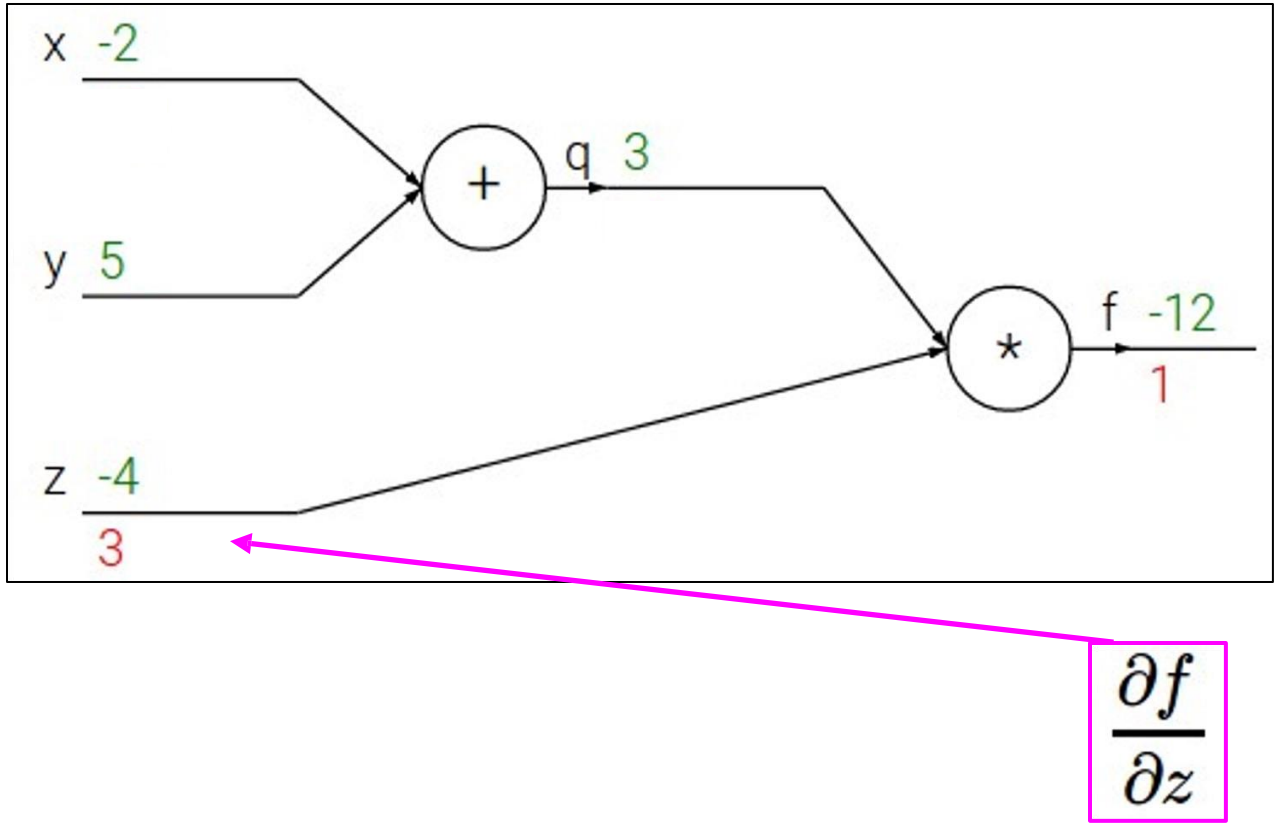
$$f(x,y,z)=(x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



反向传播

南京大学 NANJING UNIVERSITY

反向传播例子

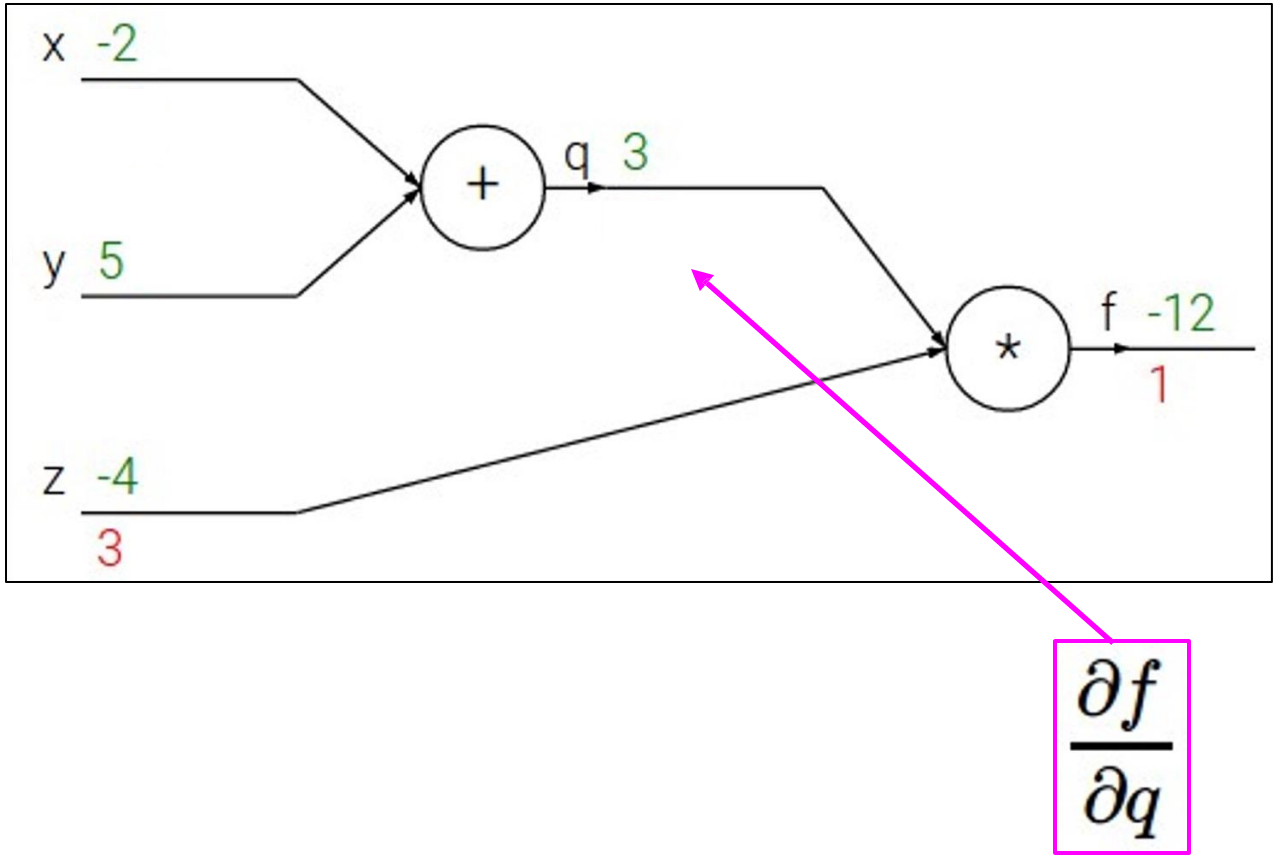
$$f(x,y,z)=(x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



反向传播

南京大学 NANJING UNIVERSITY

- **反向传播例子**

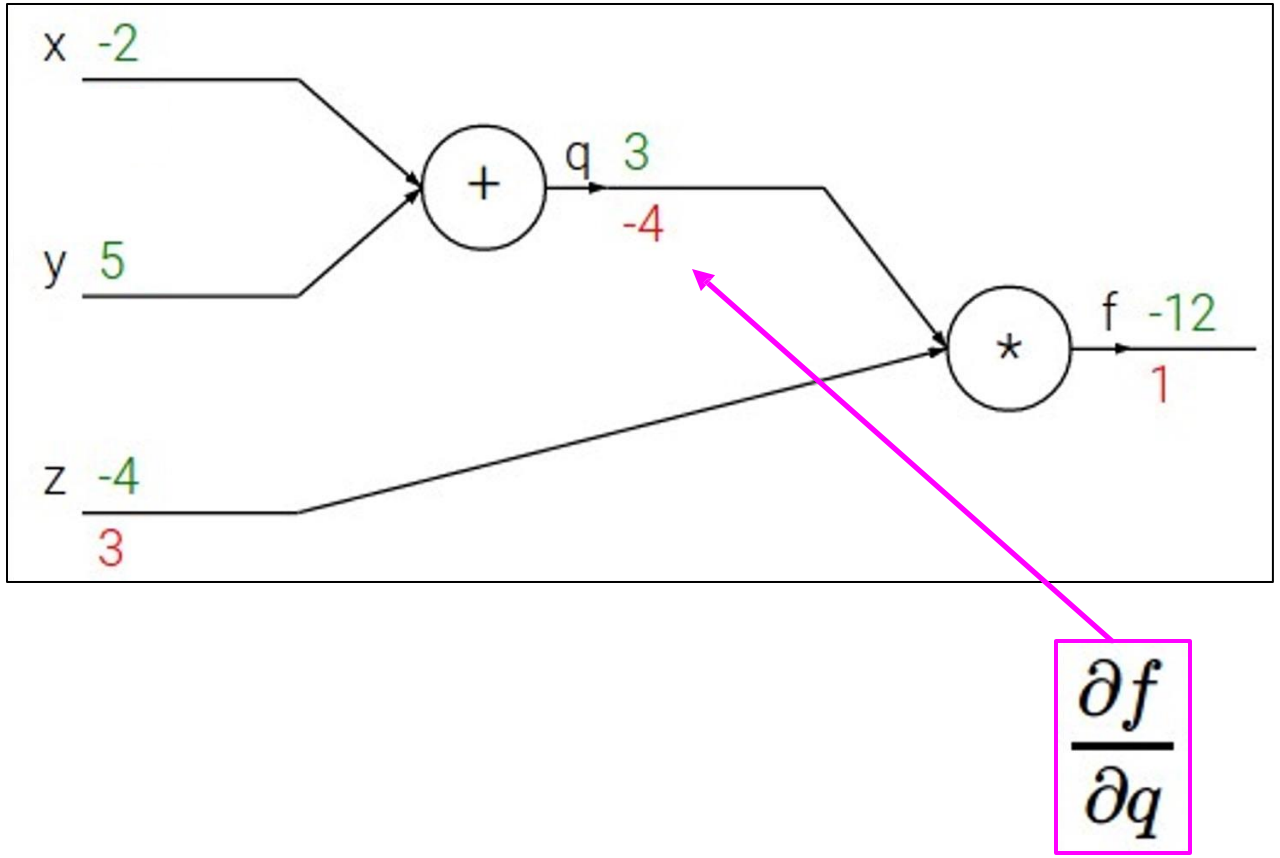
$$f(x,y,z)=(x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



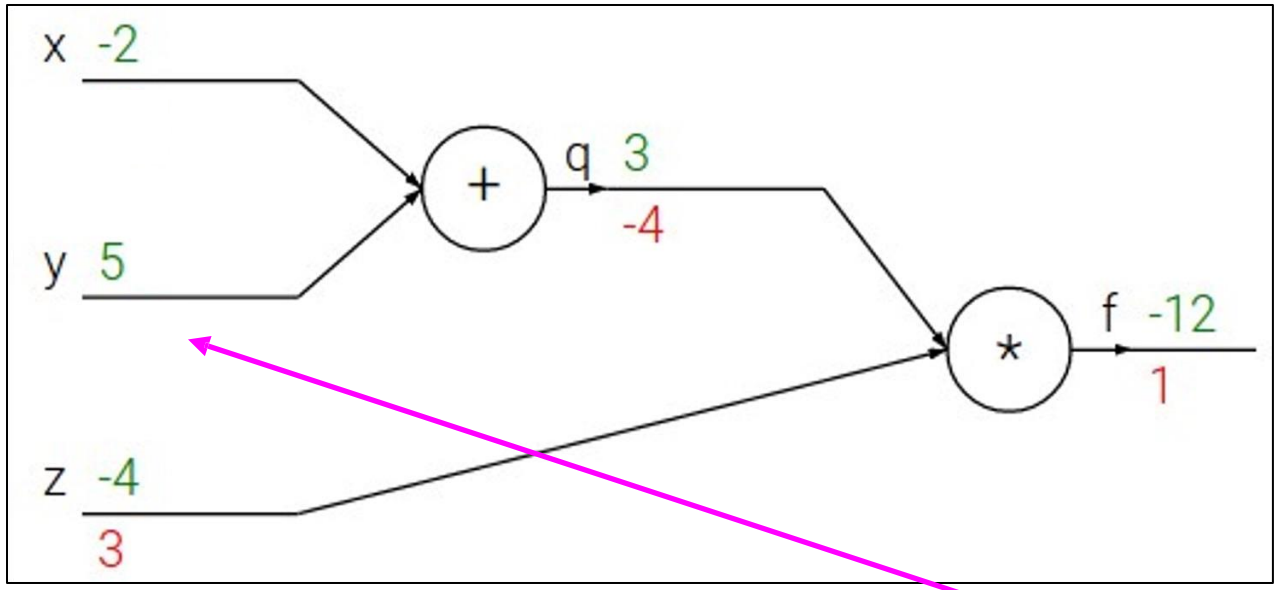
反向传播

南京大学 NANJING UNIVERSITY

- **反向传播例子**

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

e.g. x = -2, y = 5, z = -4



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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

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

Chain rule:

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

上游梯度　局部梯度

反向传播

- **反向传播例子**

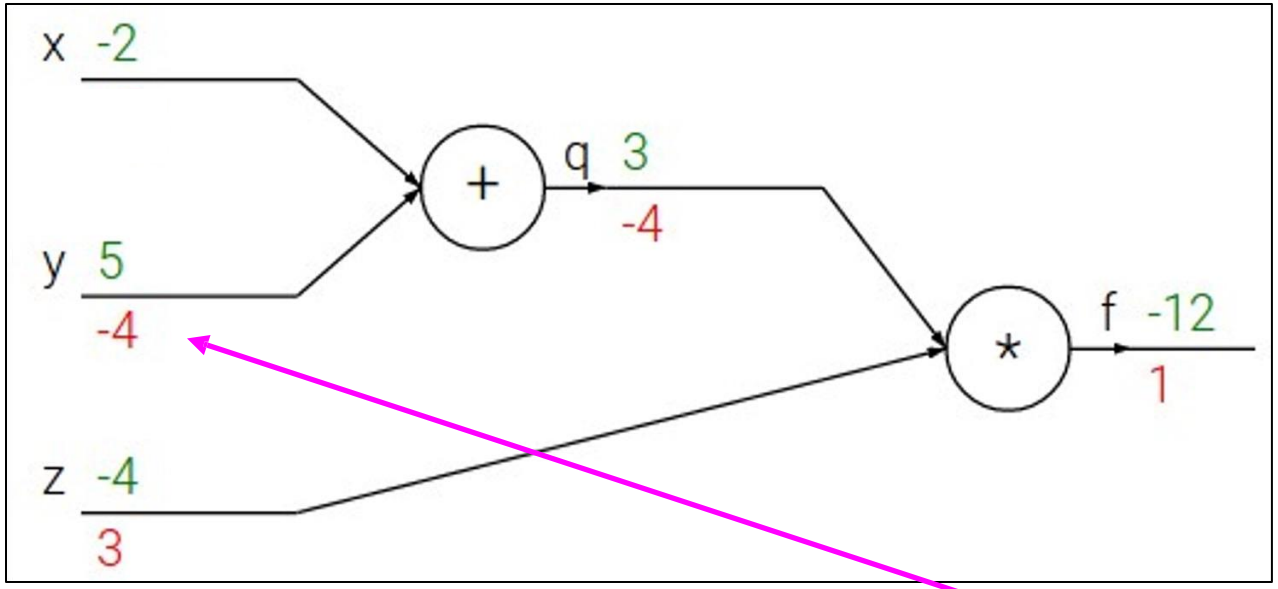
$$f(x,y,z)=(x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



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

Chain rule:

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

上游梯度　局部梯度

反向传播

南京大学 NANJING UNIVERSITY

- **反向传播例子**

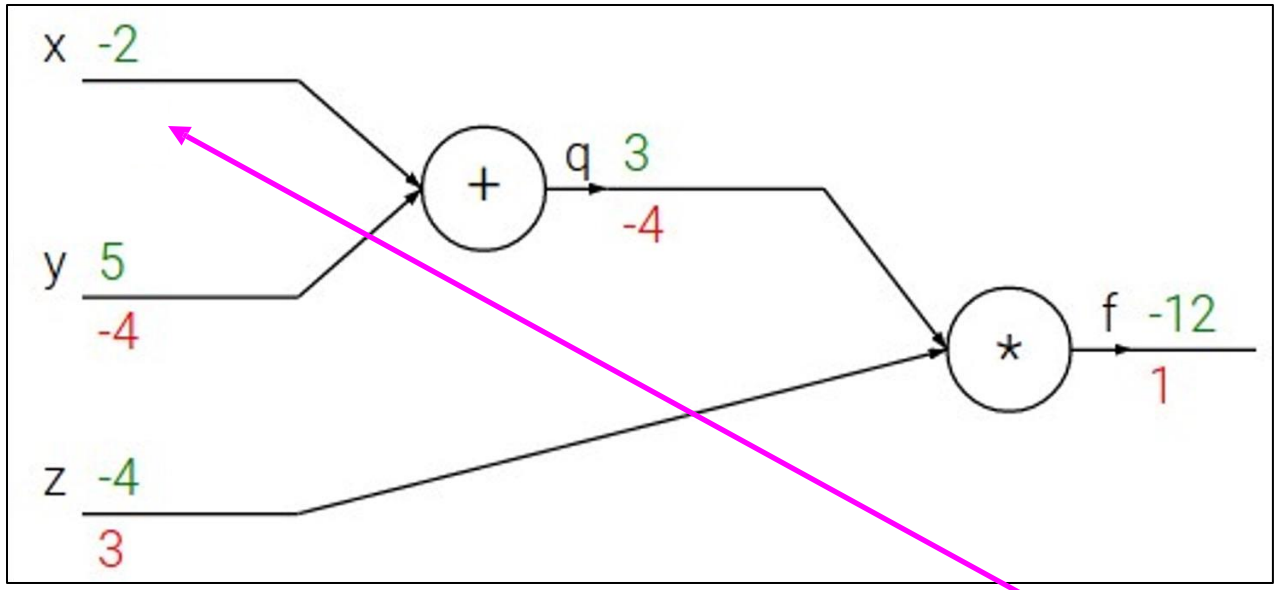
$$f(x,y,z)=(x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



$\frac{\partial f}{\partial x}$

Chain rule:

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

上游梯度　局部梯度

反向传播

南京大学 NANJING UNIVERSITY

- **反向传播例子**

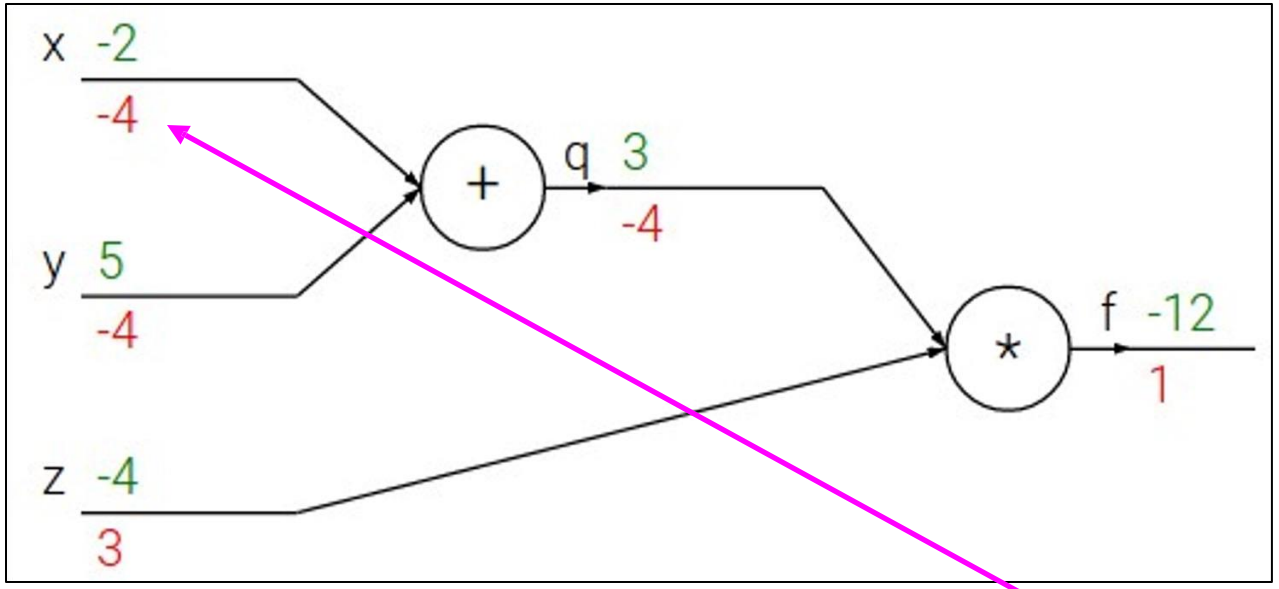
$$f(x,y,z)=(x+y)z$$

e.g. x = -2, y = 5, z = -4

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

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

Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$



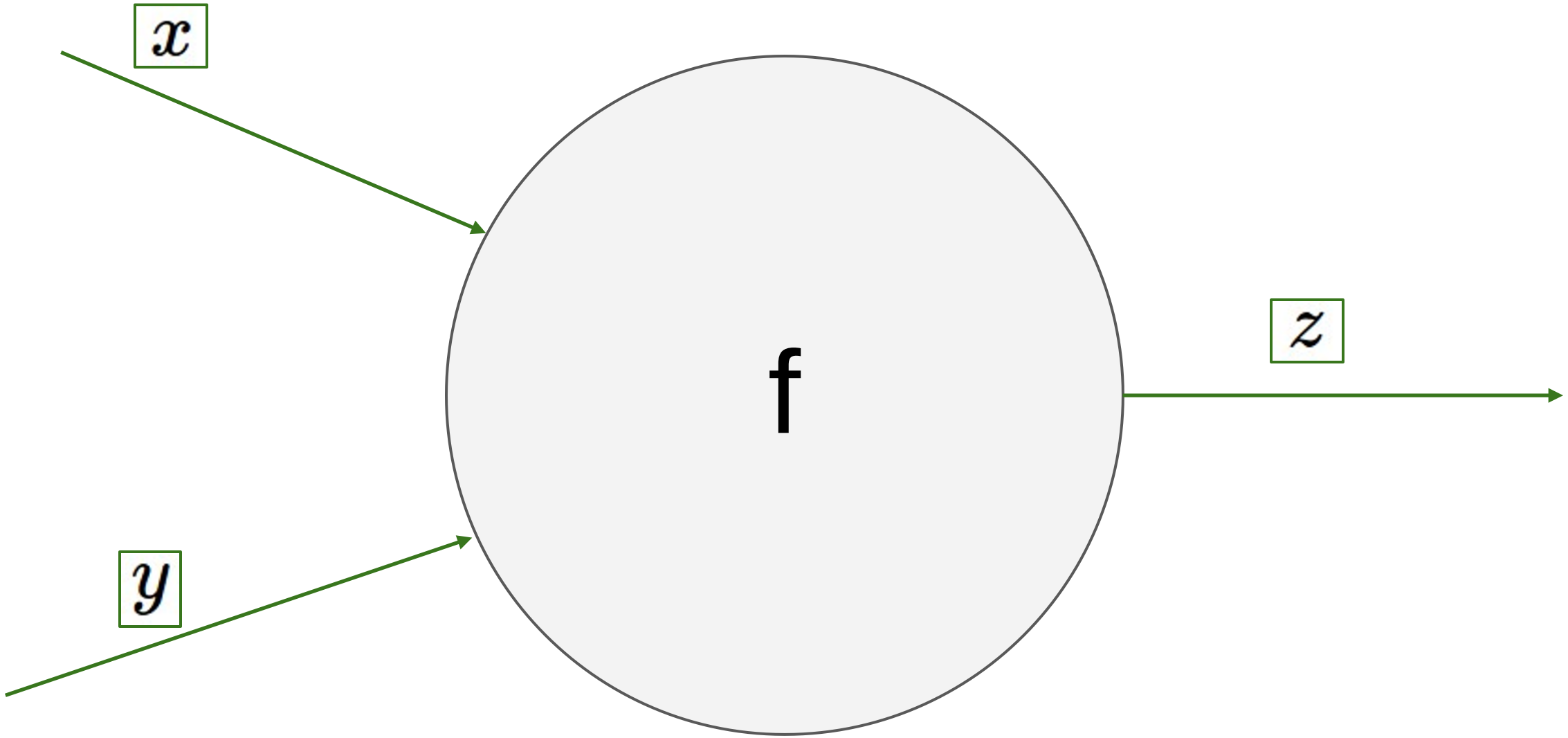
Chain rule:

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

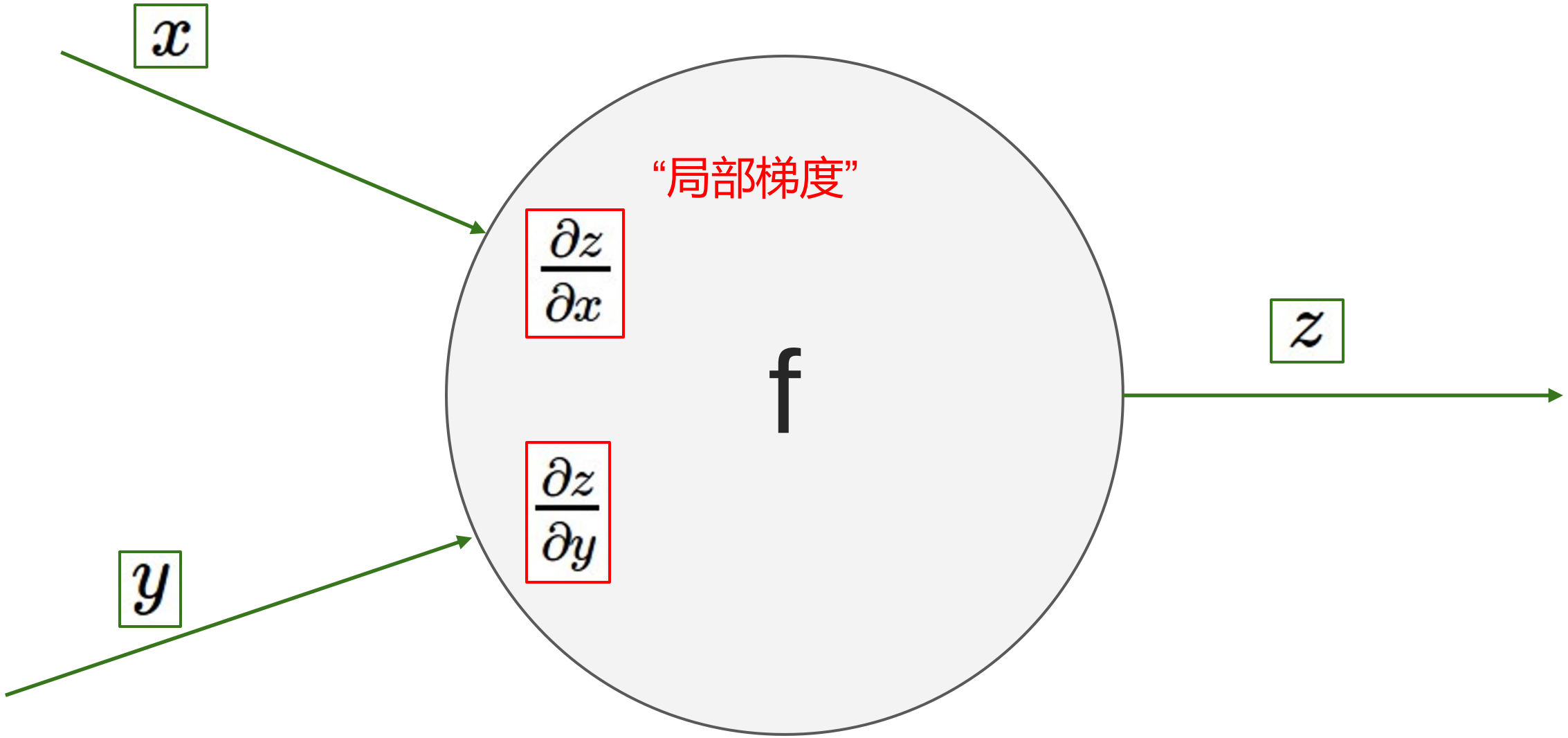
上游梯度　局部梯度

$\frac{\partial f}{\partial x}$

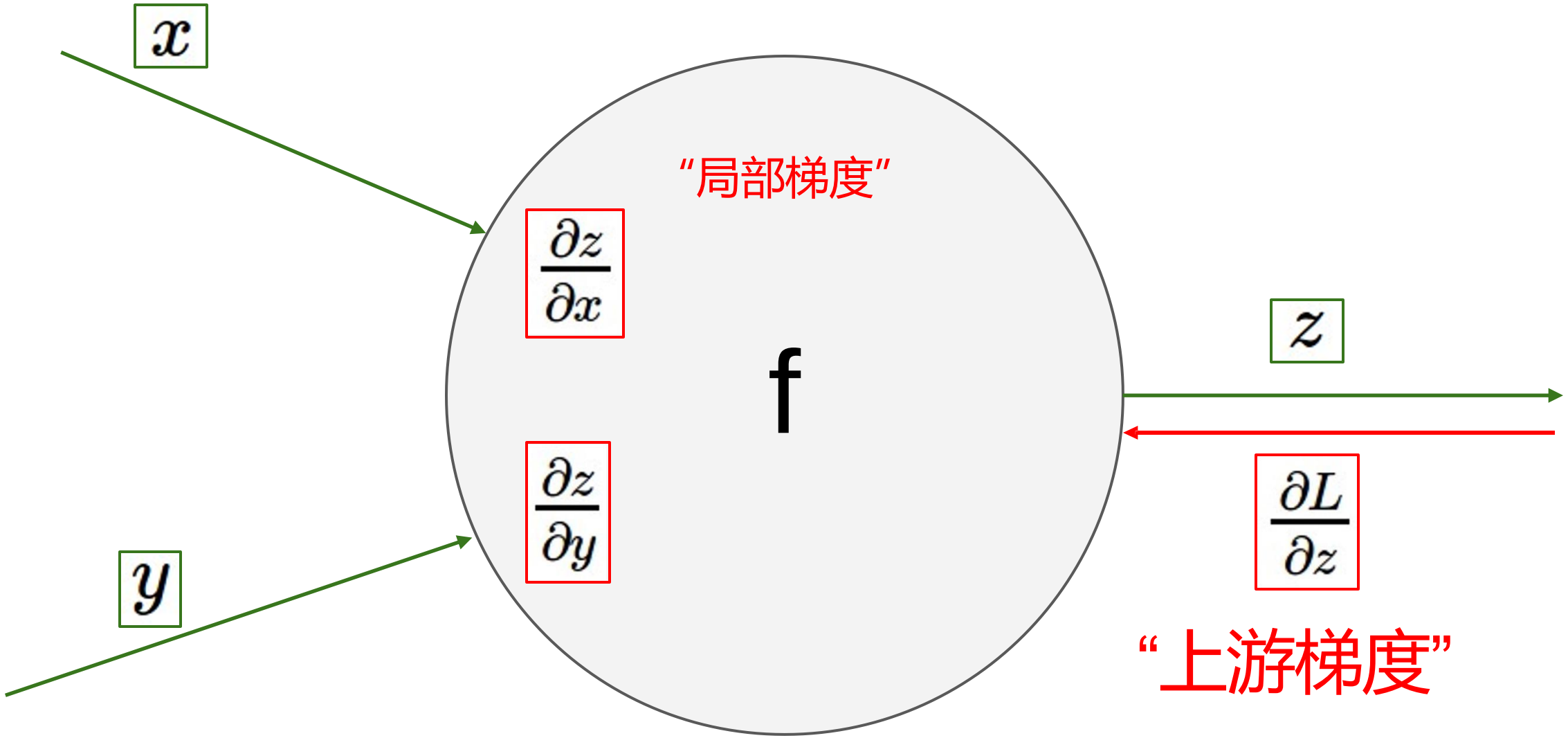
反向传播



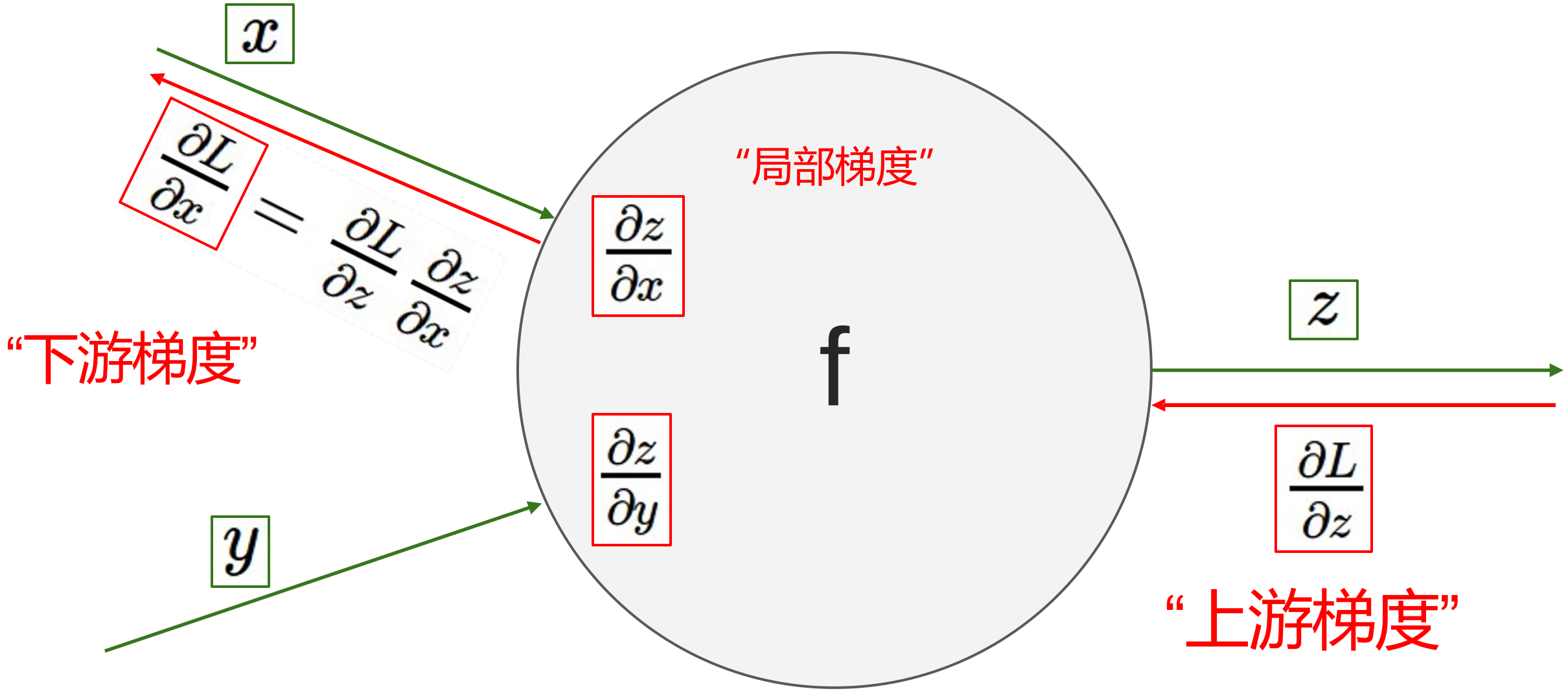
反向传播



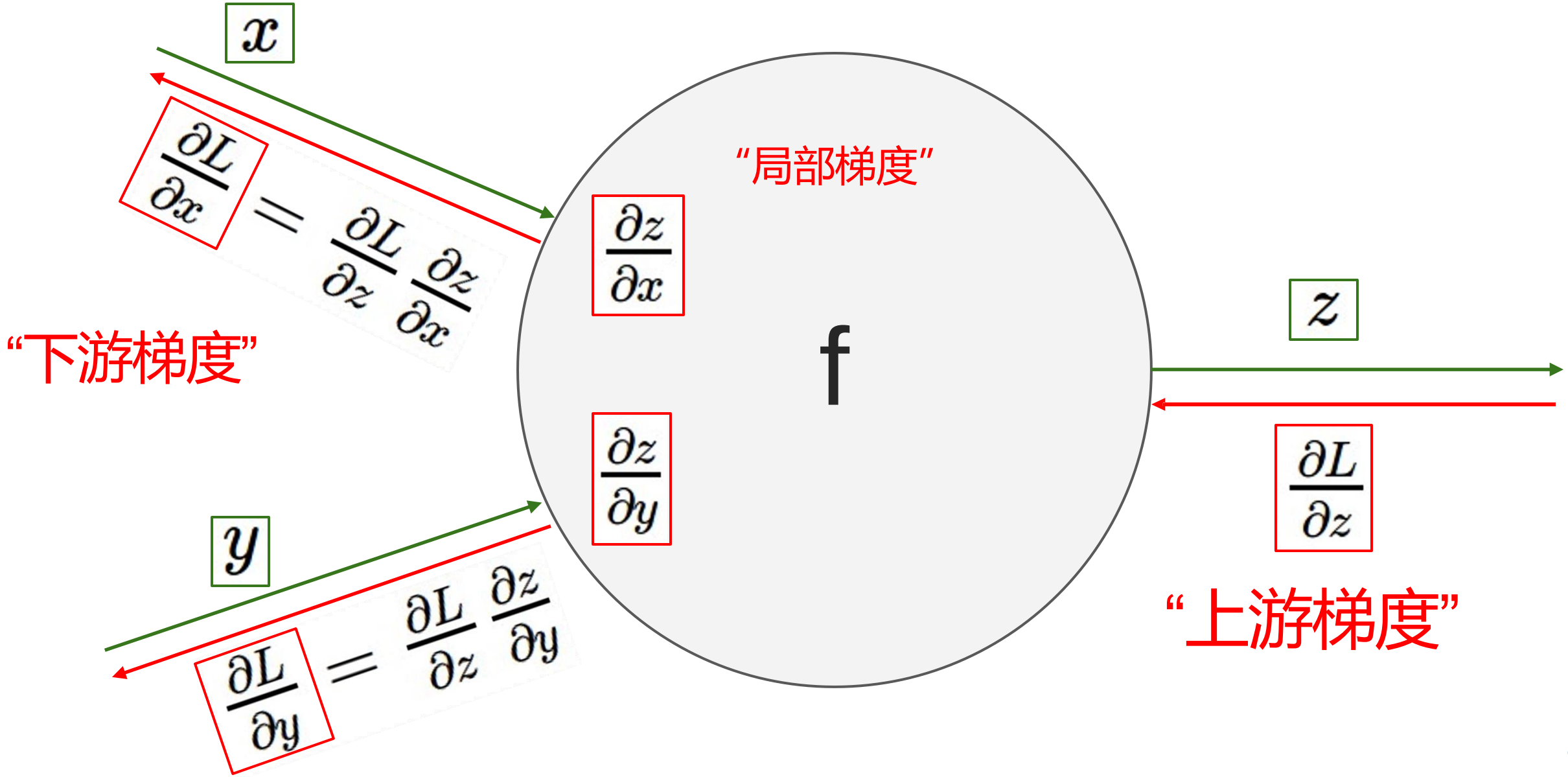
反向传播


x
y
“局部梯度”
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
f
z
$\frac{\partial L}{\partial z}$
“上游梯度”

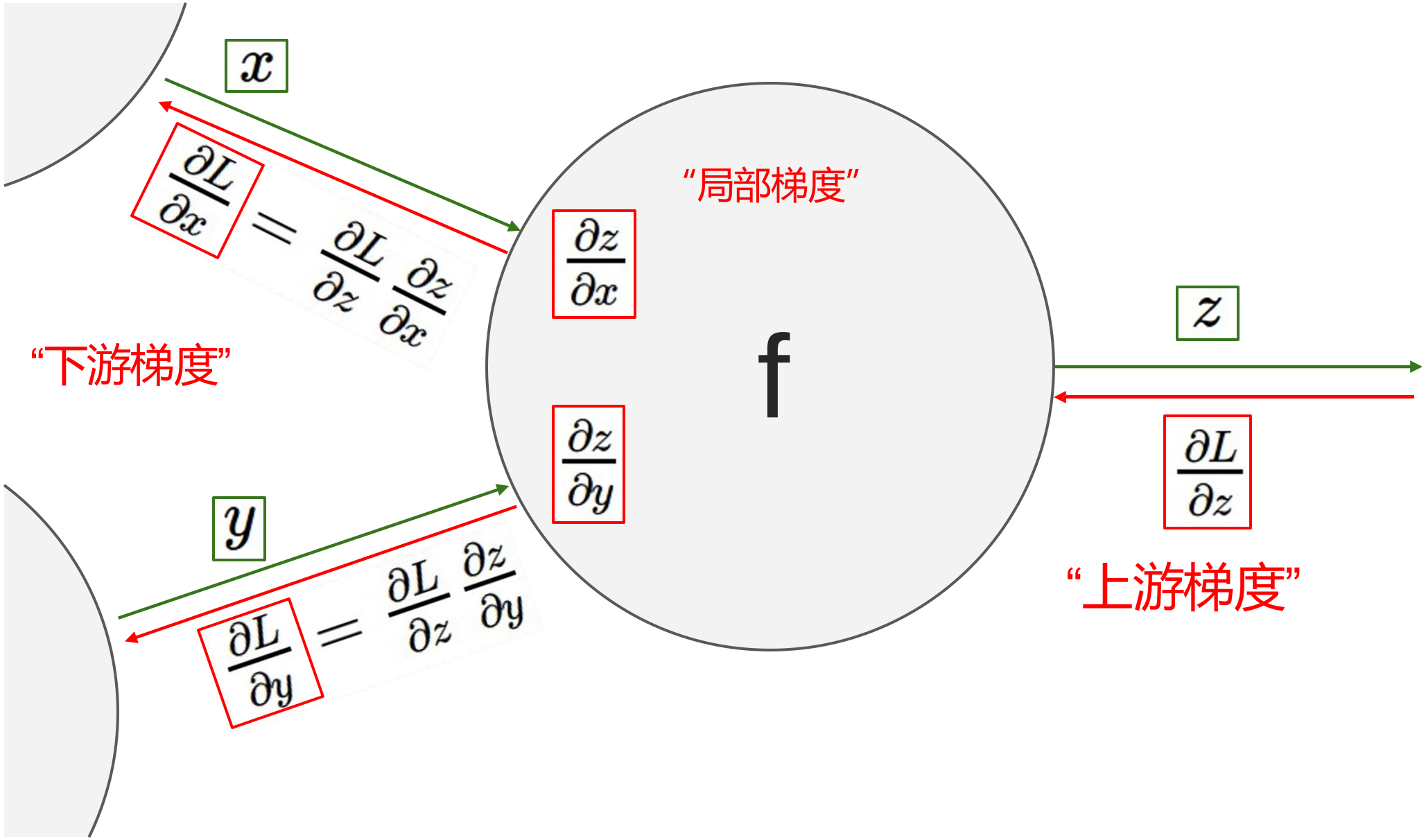
反向传播


x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
“下游梯度”
“局部梯度”
$\frac{\partial z}{\partial x}$
f
$\frac{\partial z}{\partial y}$
y
z
$\frac{\partial L}{\partial z}$
“上游梯度”

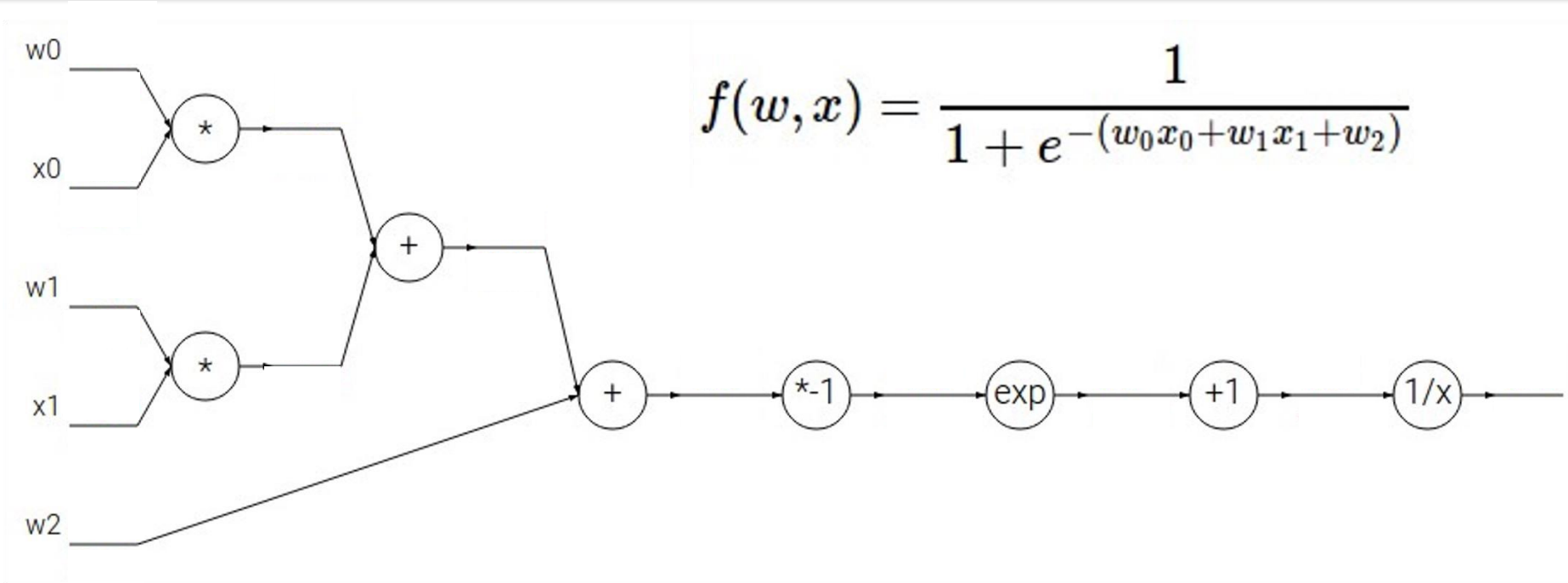
反向传播


x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
“下游梯度”
“局部梯度”
$\frac{\partial z}{\partial x}$
f
$\frac{\partial z}{\partial y}$
z
$\frac{\partial L}{\partial z}$
“上游梯度”
y
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$

反向传播

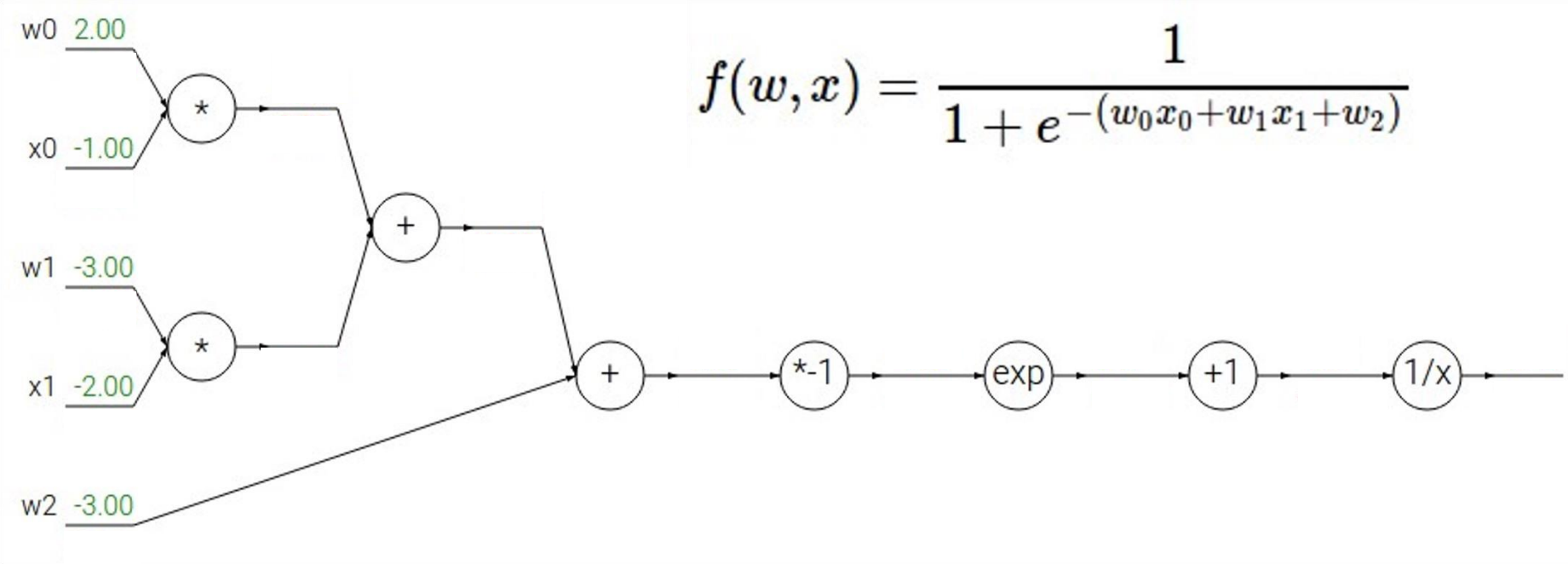

x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
“下游梯度”
y
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
“局部梯度”
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
f
z
$\frac{\partial L}{\partial z}$
“上游梯度”

反向传播



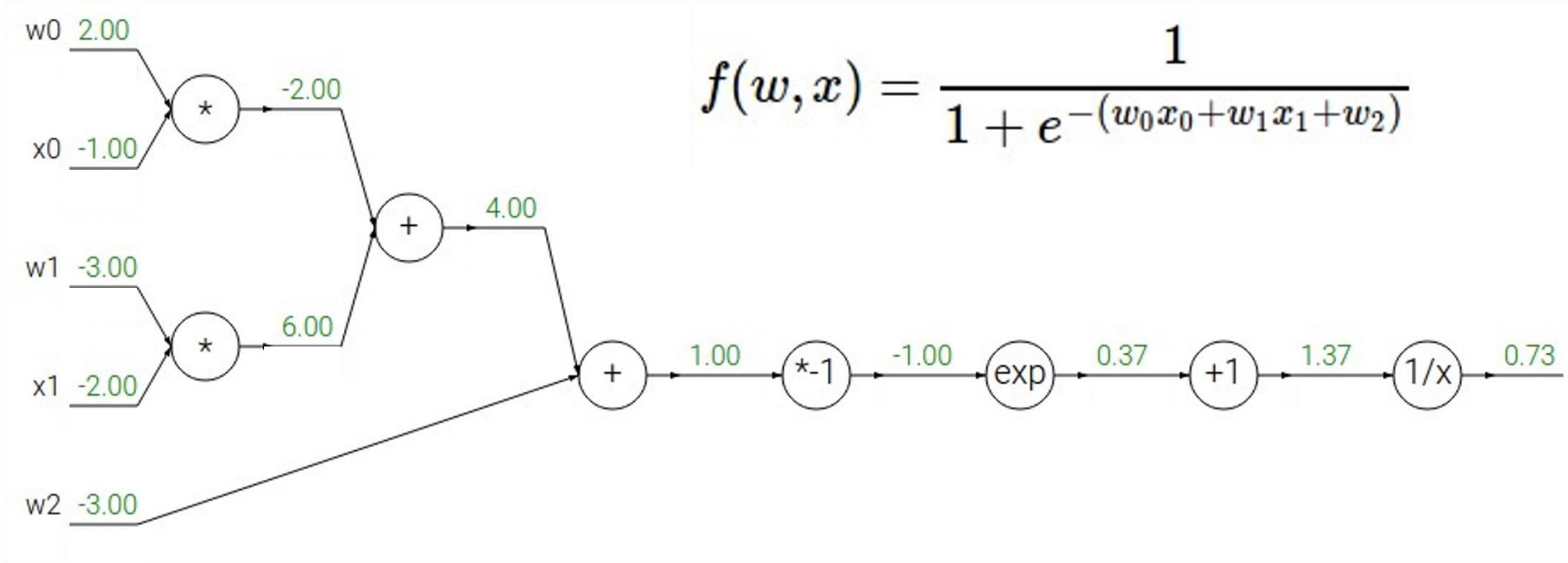
$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

反向传播



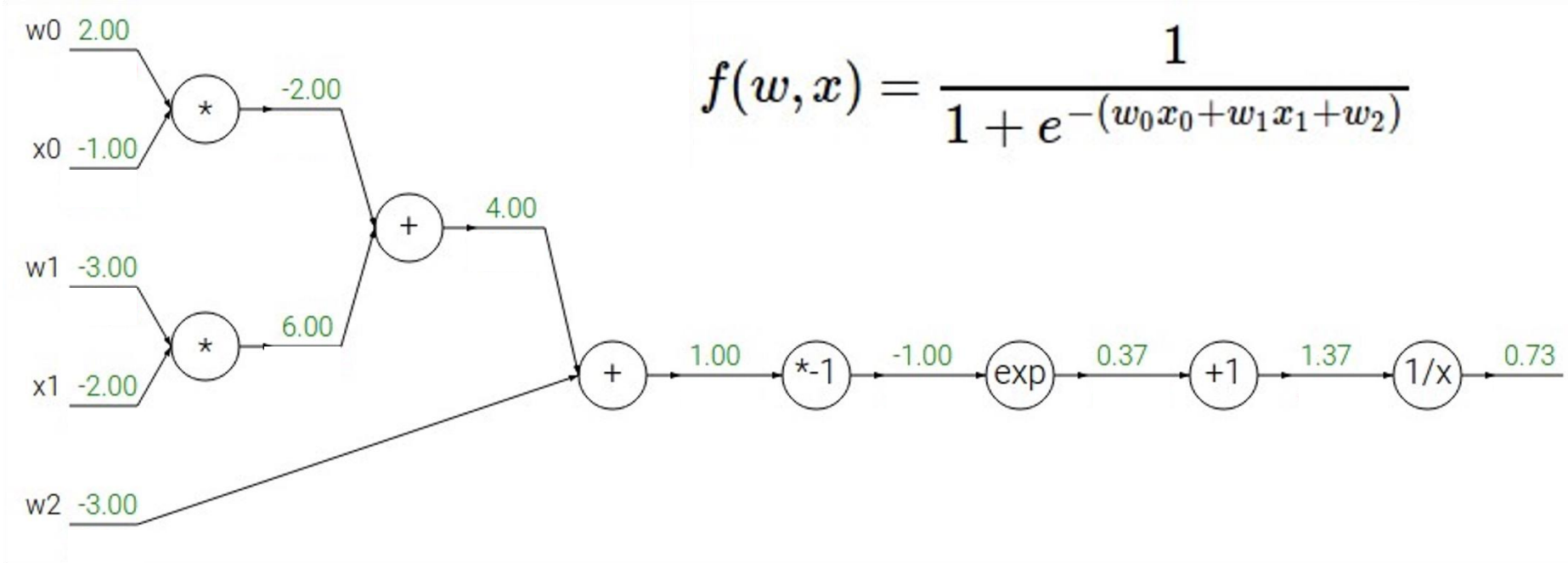
$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

反向传播



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

反向传播



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

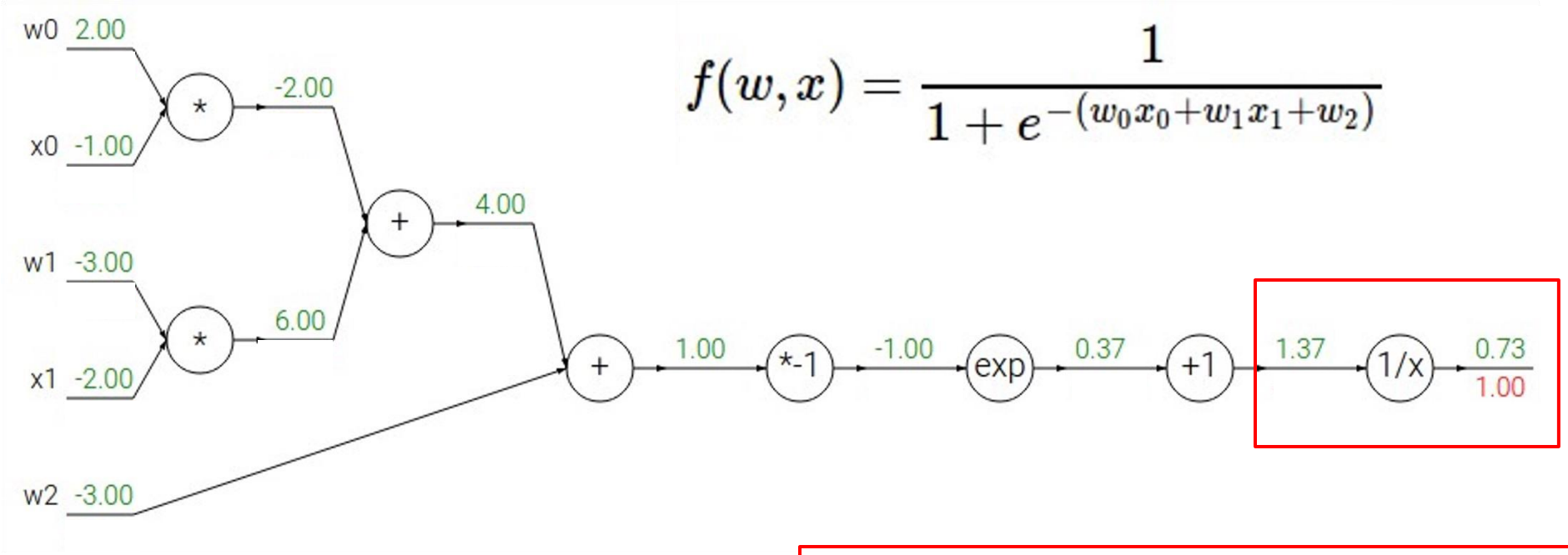
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



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

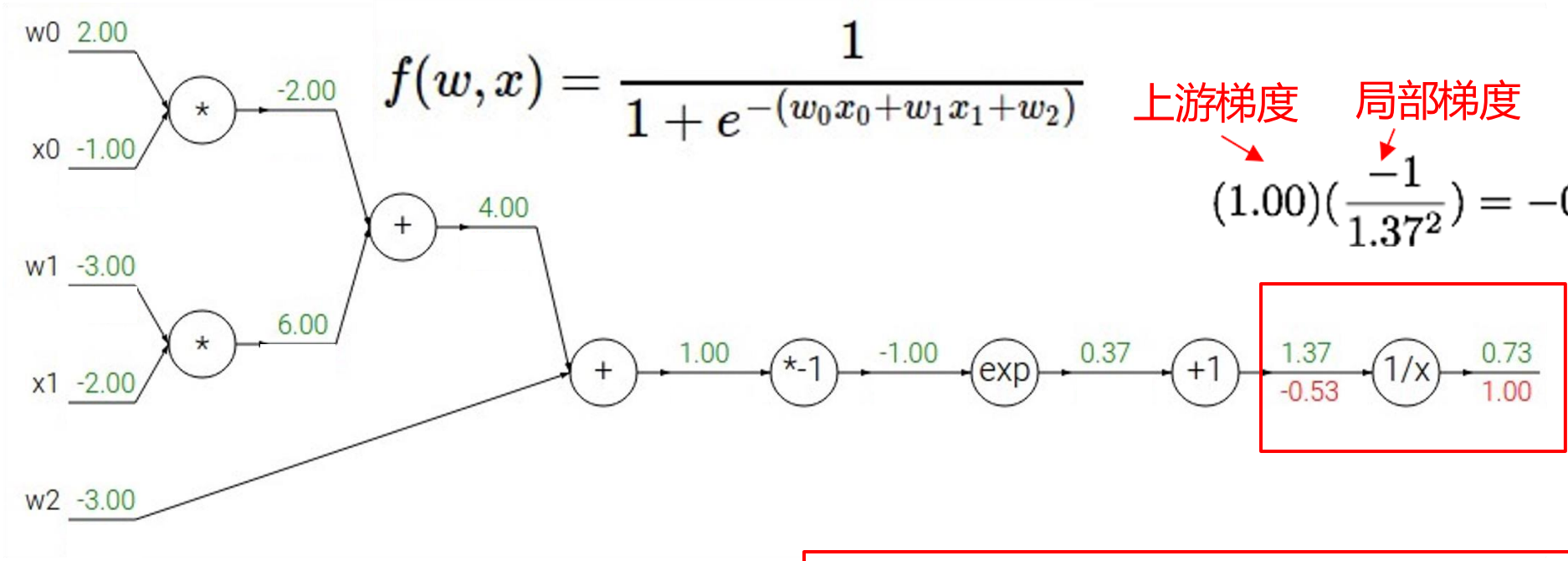
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



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

上游梯度　局部梯度

$$(1.00)(\frac{-1}{1.37^2}) = -0.53$$

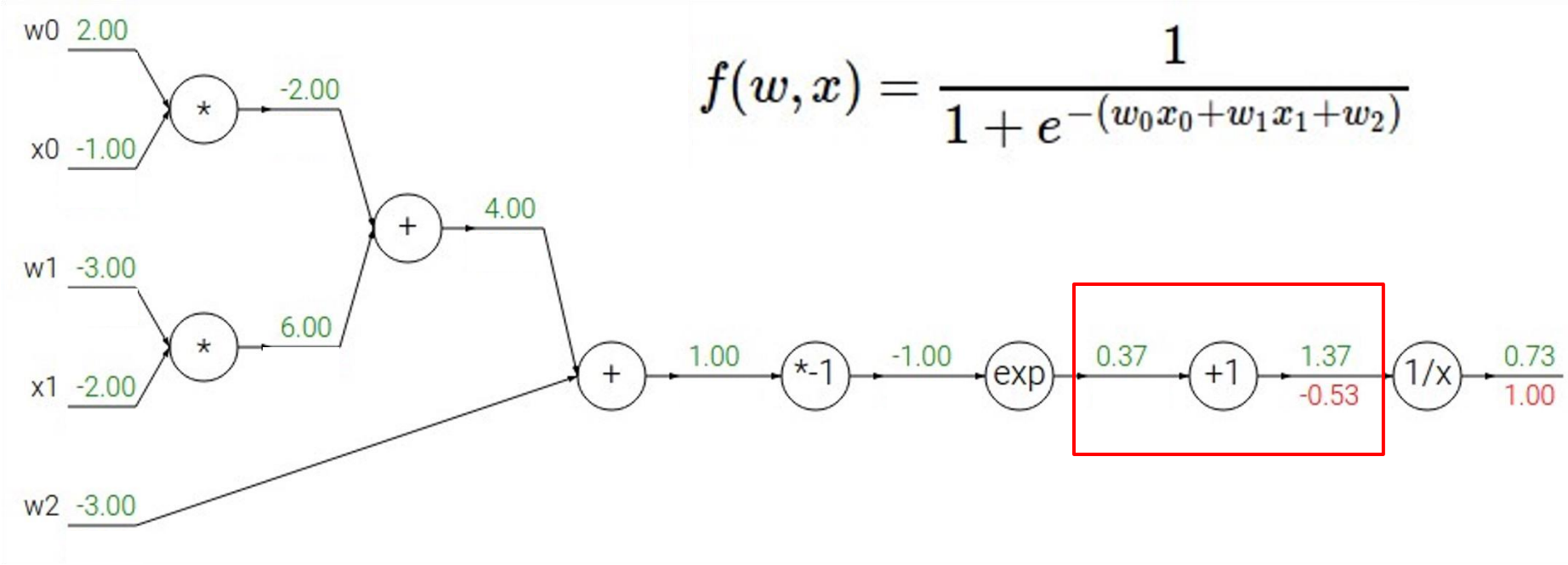
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



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

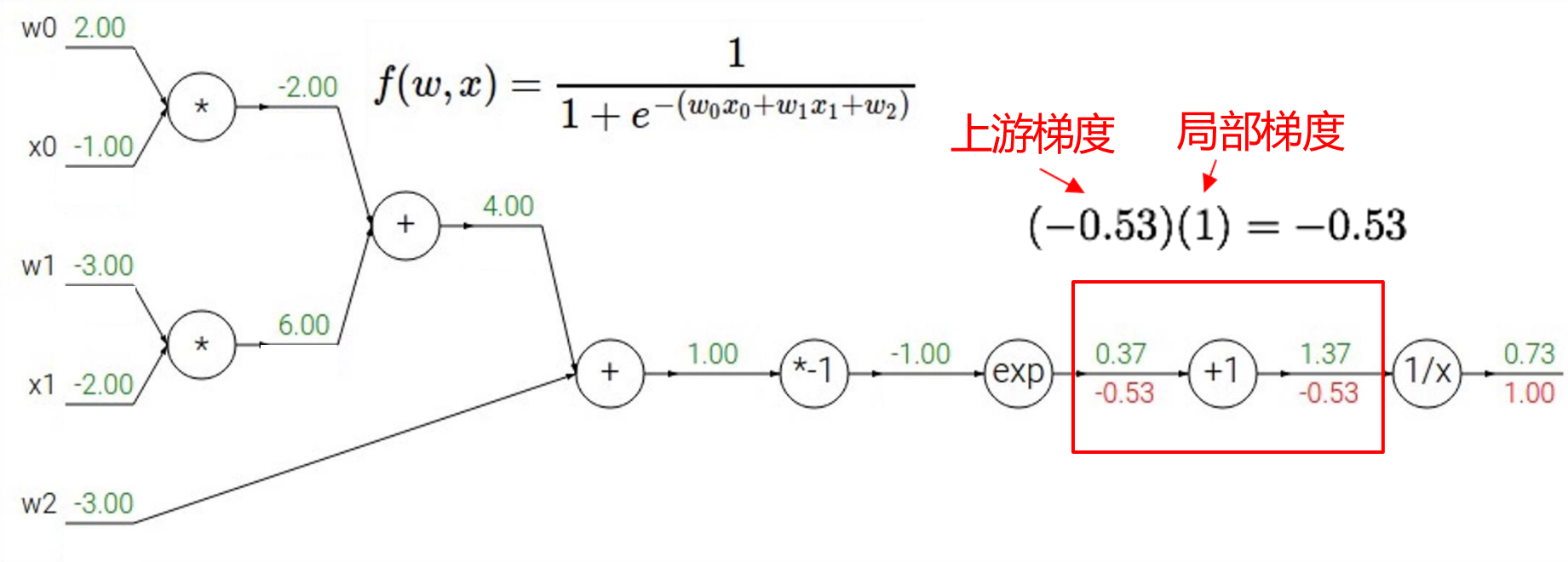
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



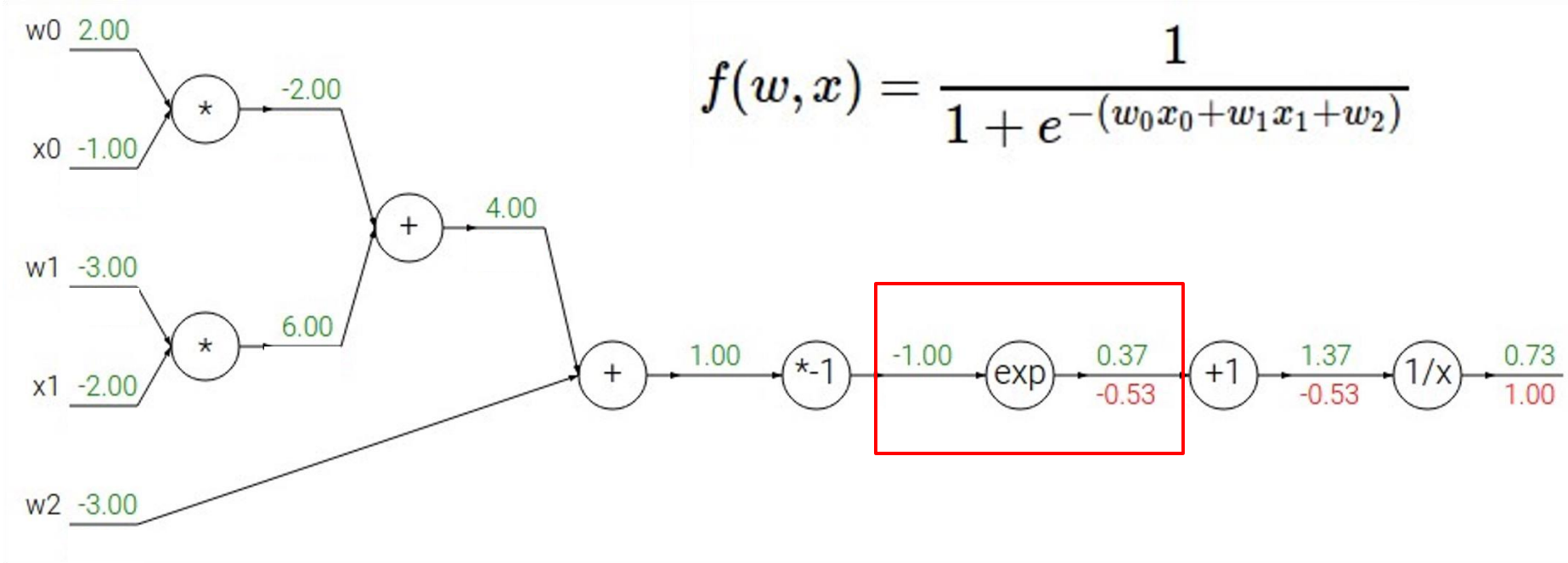
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



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

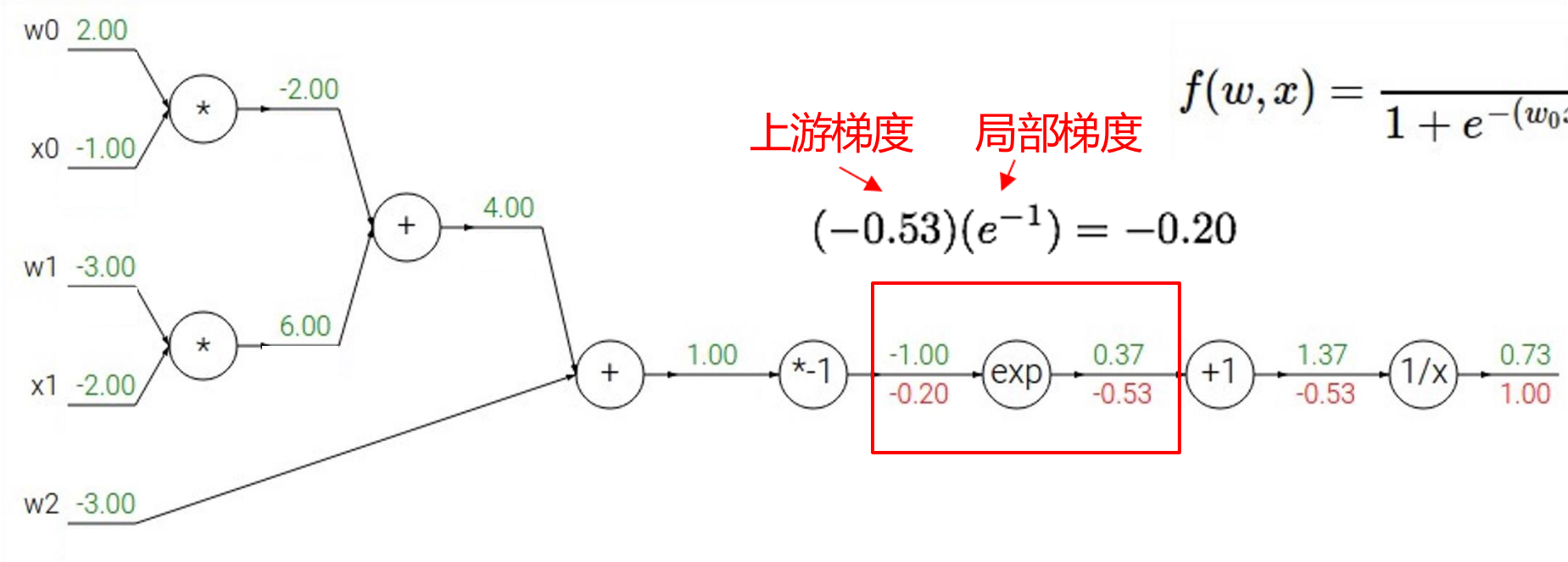
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



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

$$(-0.53)(e^{-1}) = -0.20$$

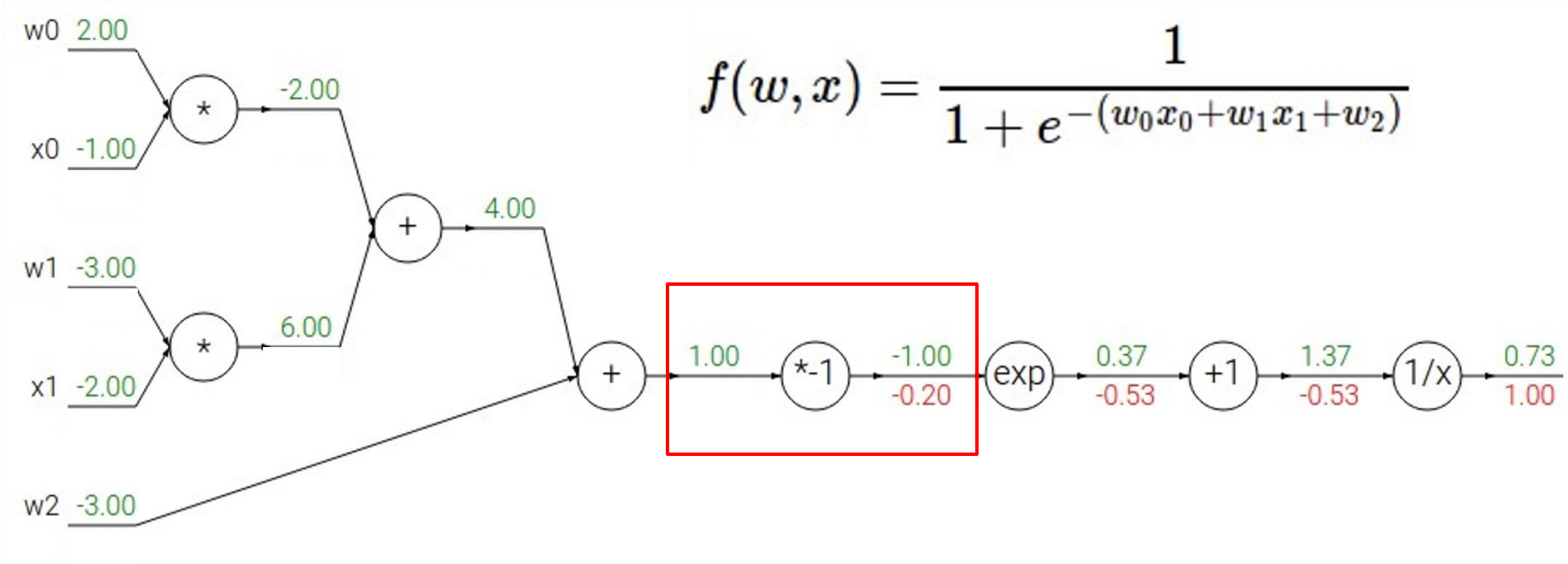
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



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

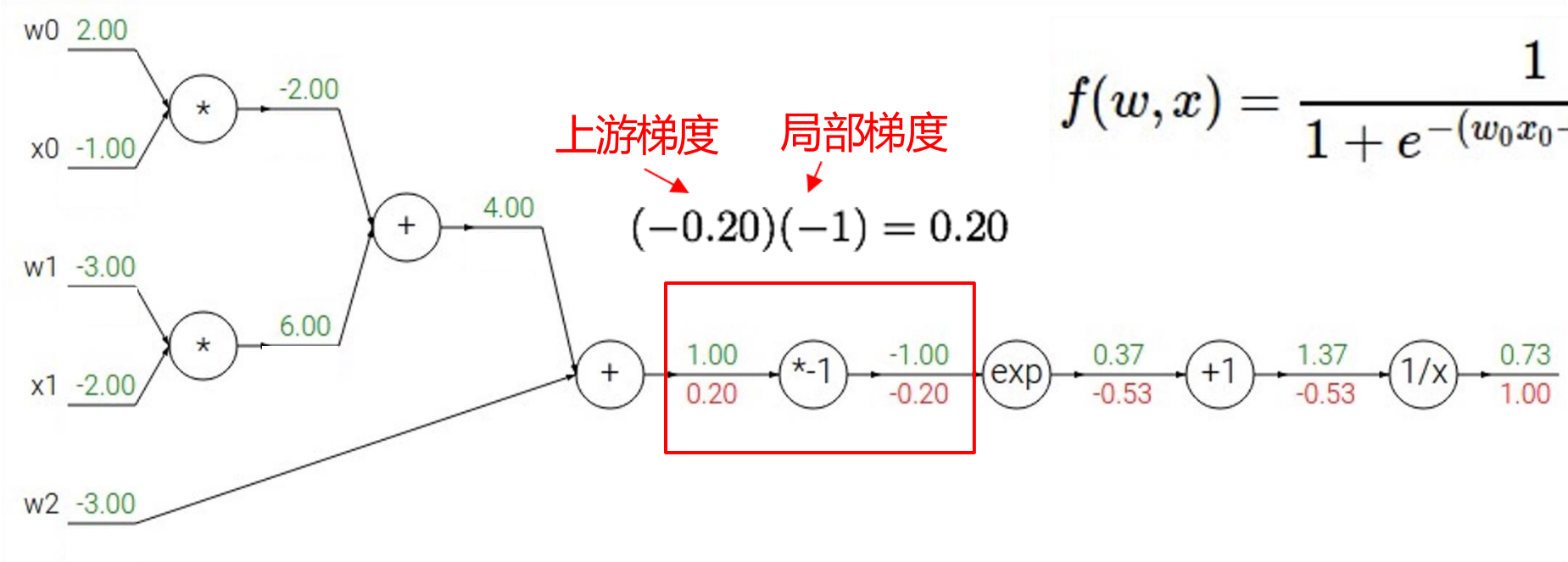
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



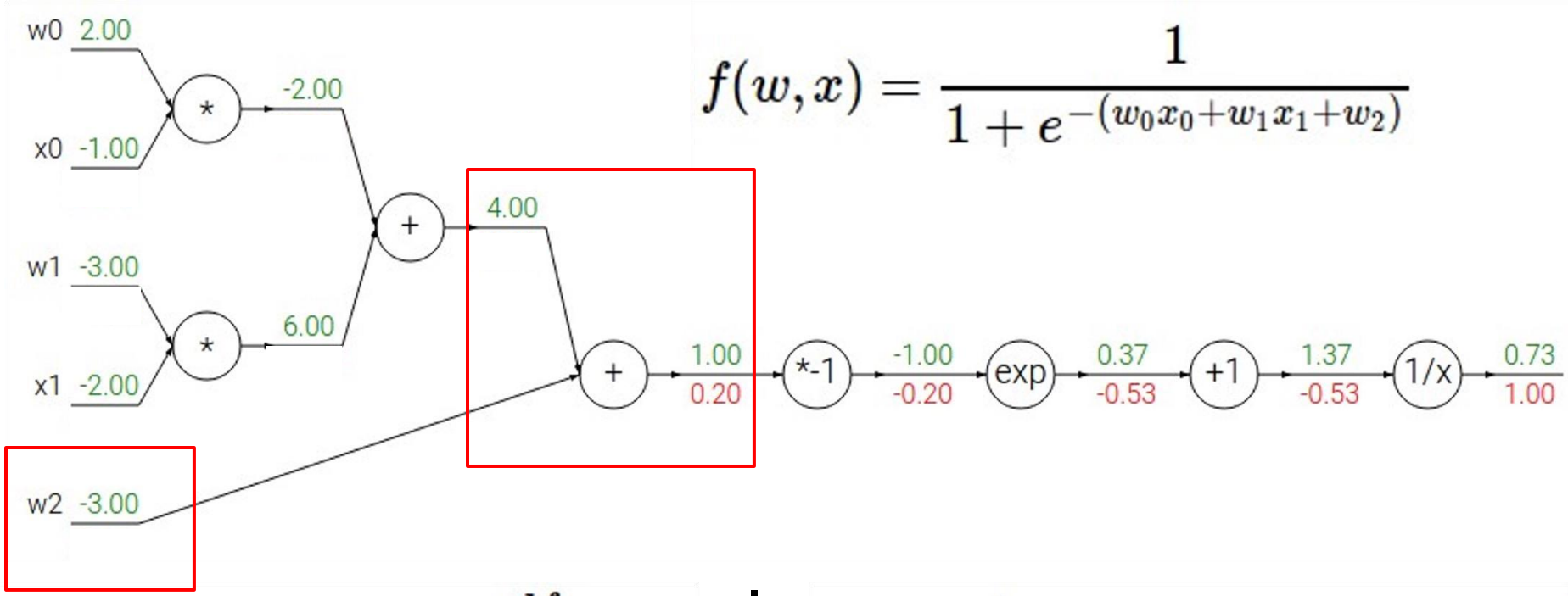
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



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

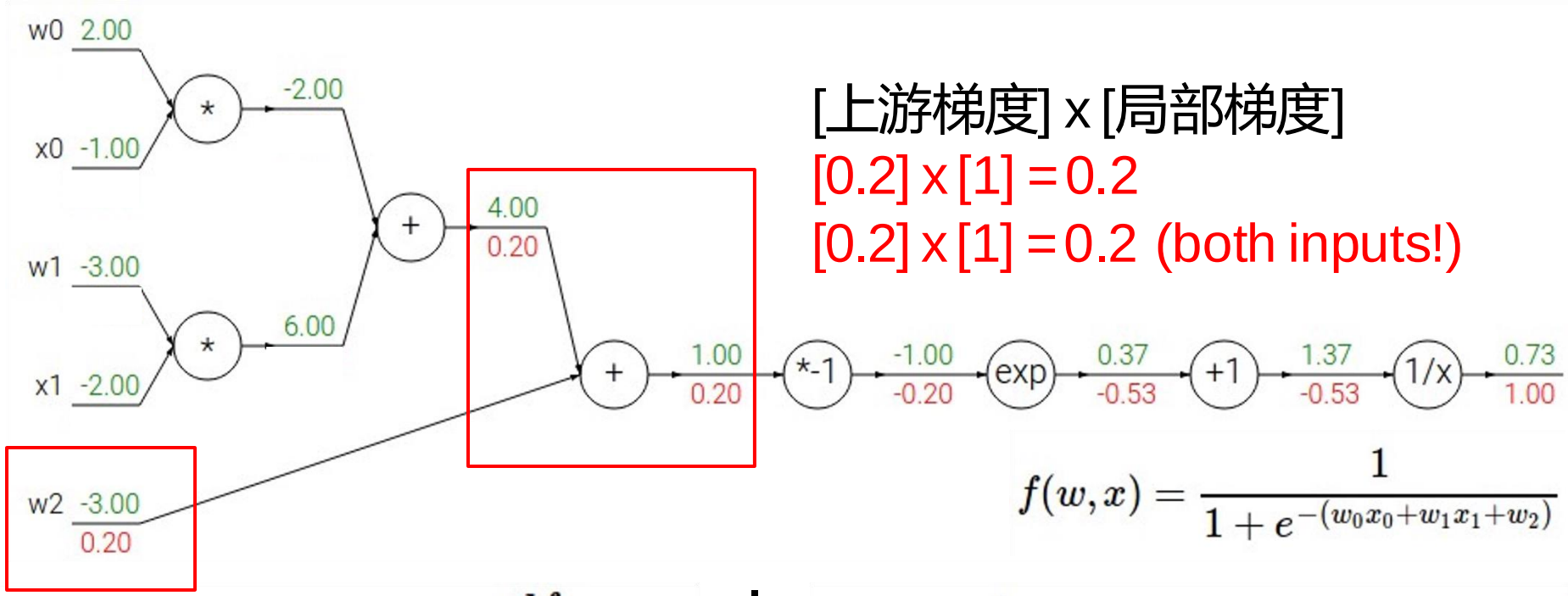
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



[上游梯度] x [局部梯度]

[0.2] x [1] = 0.2

[0.2] x [1] = 0.2 (both inputs!)

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

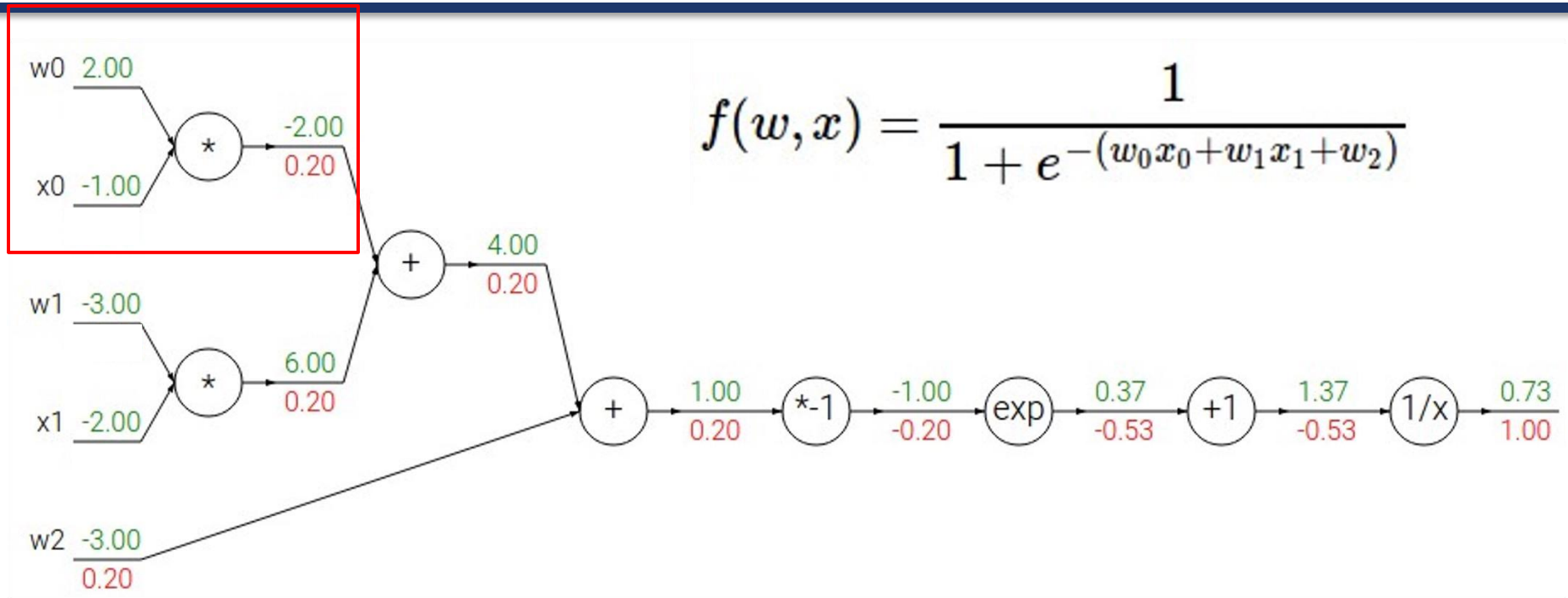
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



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

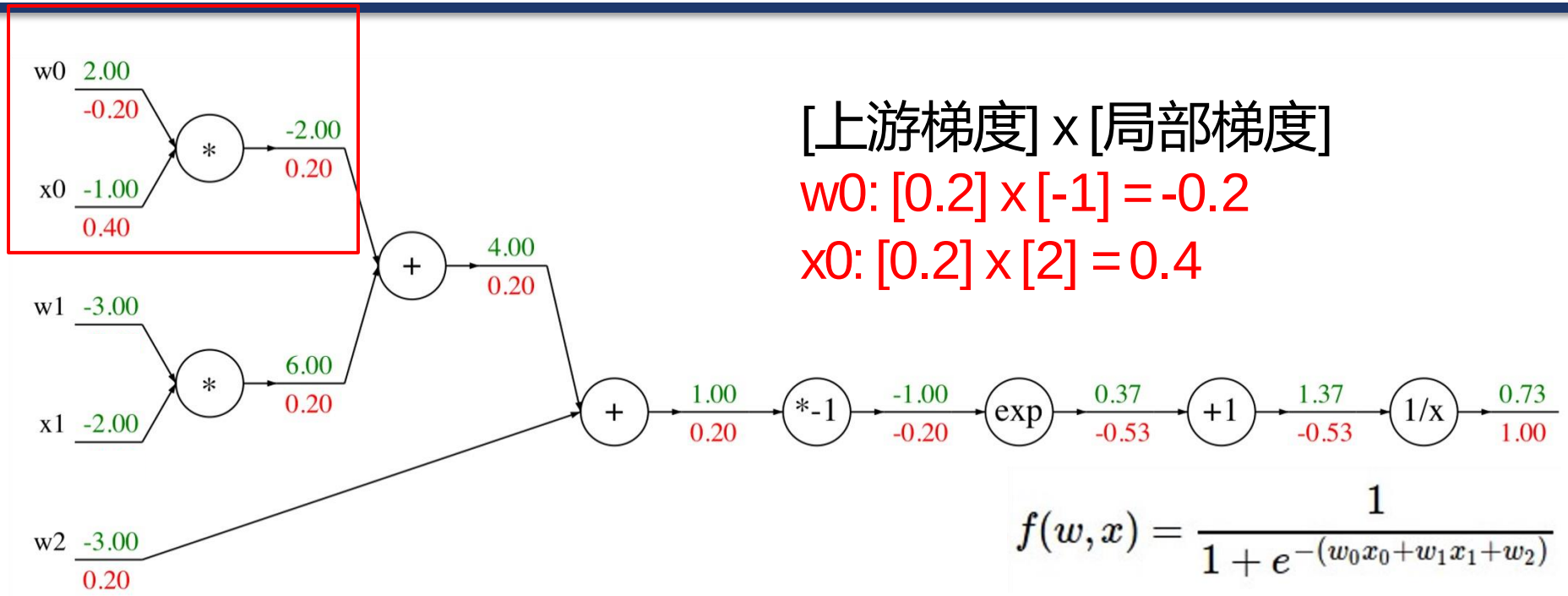
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播



[上游梯度] x [局部梯度]

w0: [0.2] x [-1] = -0.2

x0: [0.2] x [2] = 0.4

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

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

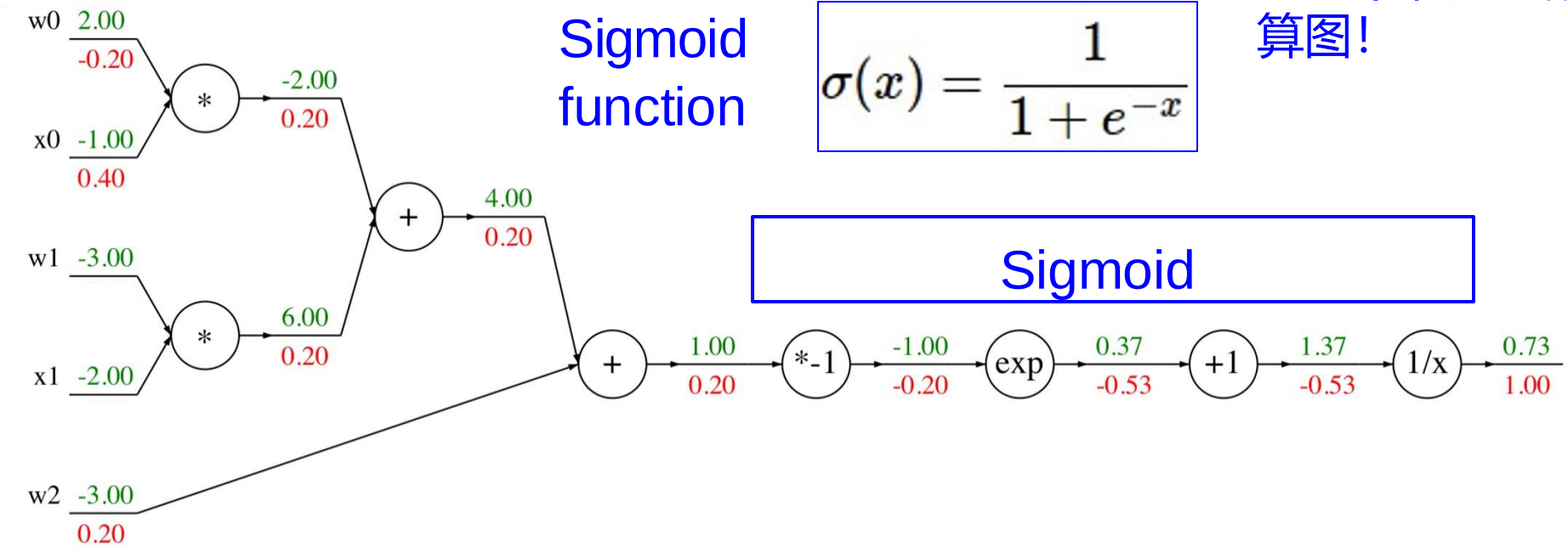
$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

反向传播

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



Sigmoid function

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

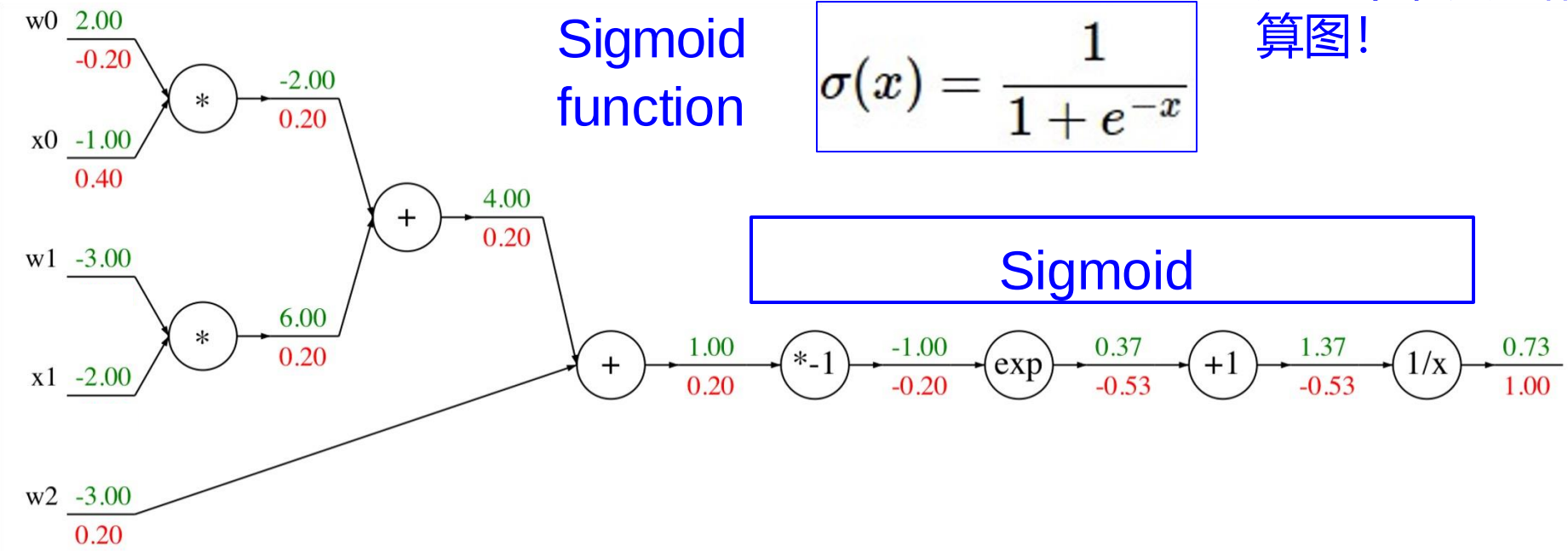
Sigmoid

计算图表示可能不是唯一的。选择一个可以较易表达每个节点局部梯度的计算图!

反向传播

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

计算图表示可能不是唯一的。选择一个可以较易表达每个节点局部梯度的计算图!



Sigmoid function

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

Sigmoid

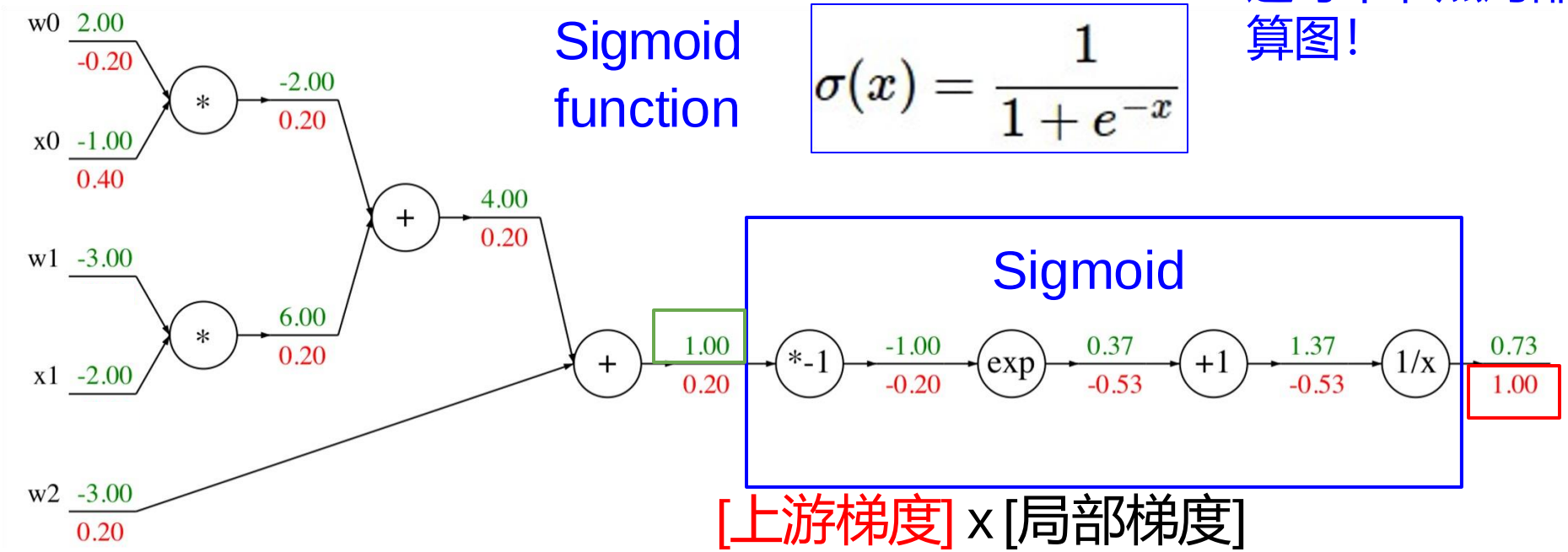
Sigmoid local gradient:

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$

反向传播

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

计算图表示可能不是唯一的。选择一个可以较易表达每个节点局部梯度的计算图!



Sigmoid function

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

[上游梯度] x [局部梯度]

[1.00] x [(1 - 1/(1+e^{-1})) (1/(1+e^{-1}))] = 0.2

Sigmoid local gradient:

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$

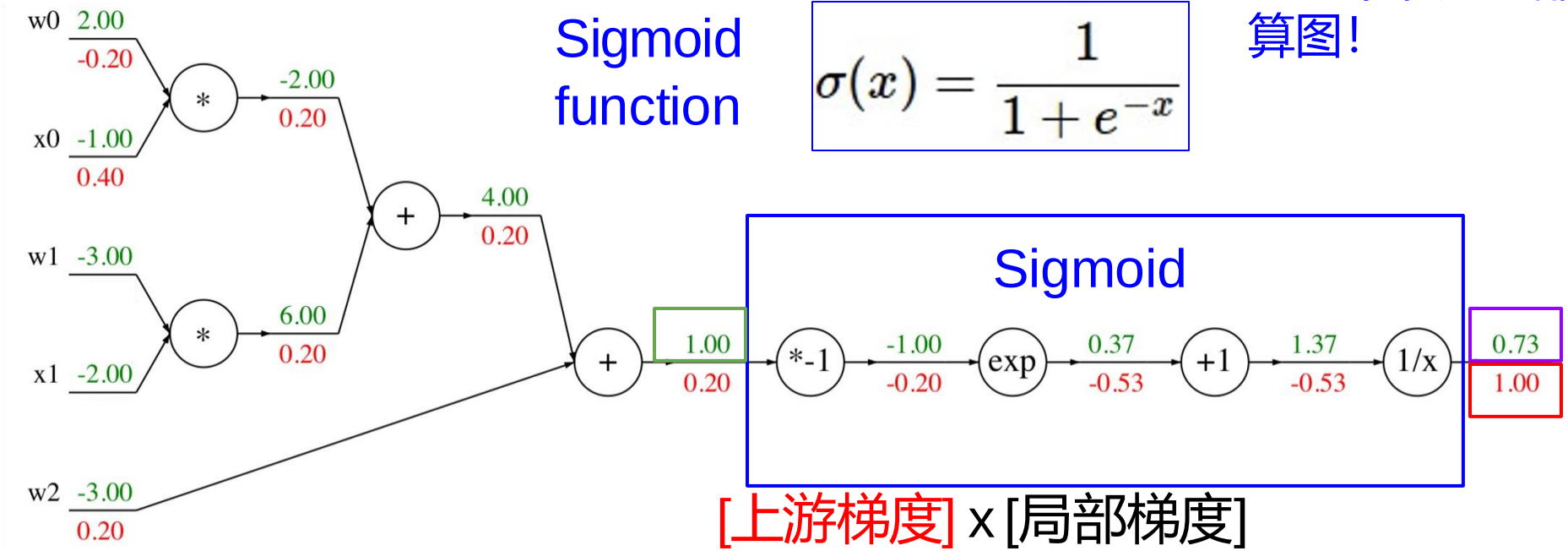
反向传播

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

计算图表示可能不是唯一的。选择一个可以较易表达每个节点局部梯度的计算图!

Sigmoid function

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



[上游梯度] x [局部梯度]

[1.00] x [(1 - 0.73) (0.73)] = 0.2

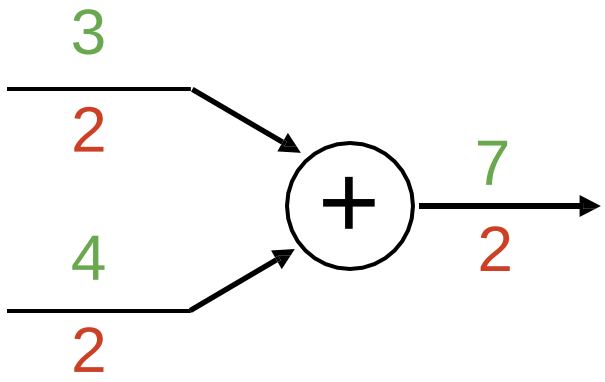
Sigmoid local gradient:

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$

反向传播

- **梯度流中的基本计算模式**

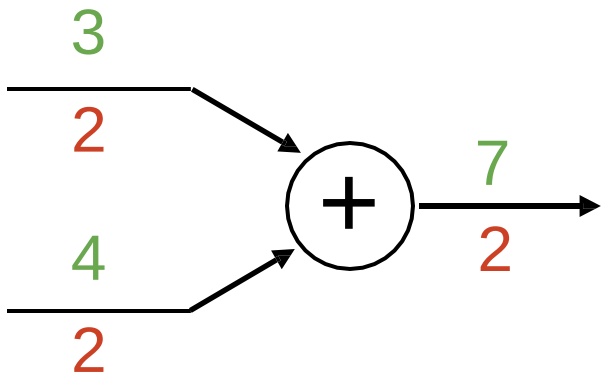
  add gate: gradient distributor



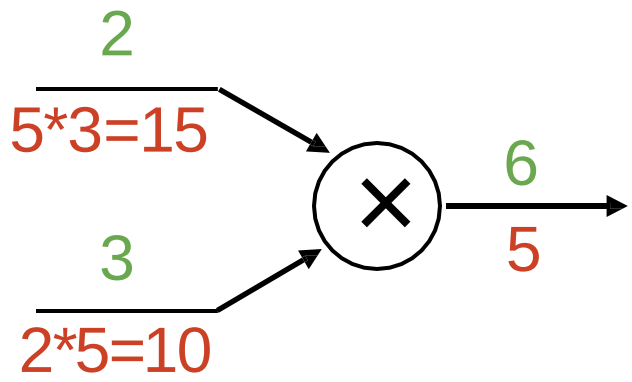
反向传播

- **梯度流中的基本计算模式**

add gate: gradient distributor



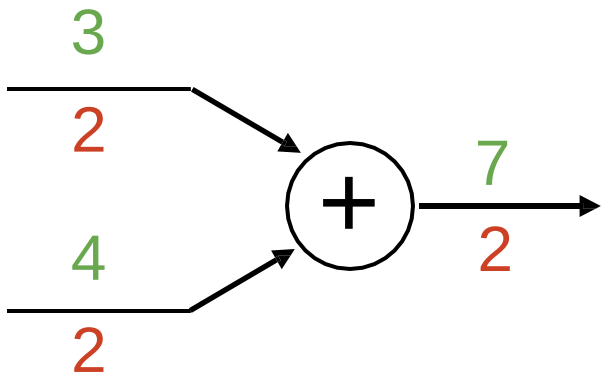
mul gate: “swap multiplier”



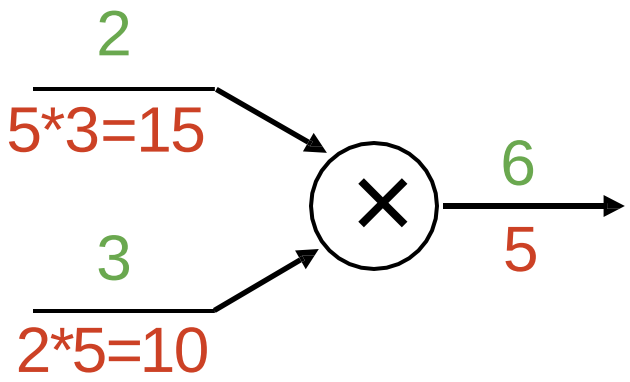
反向传播

- **梯度流中的基本计算模式**

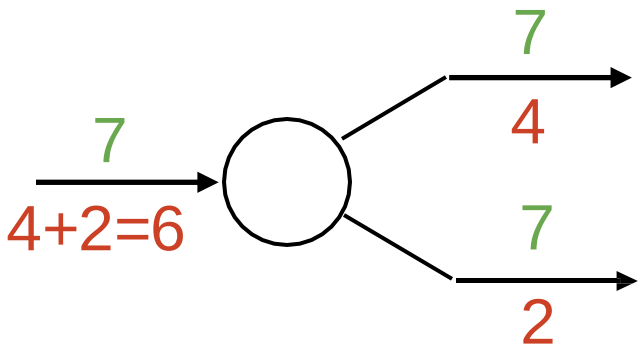
add gate: gradient distributor



mul gate: “swap multiplier”



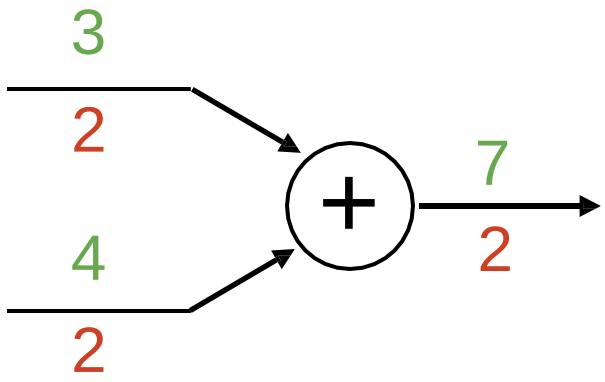
copy gate: gradient adder



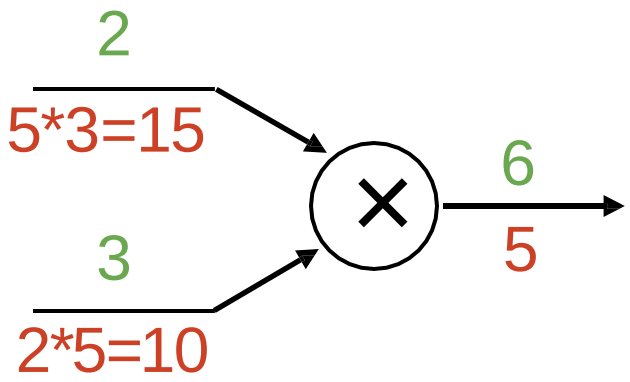
反向传播

- **梯度流中的基本计算模式**

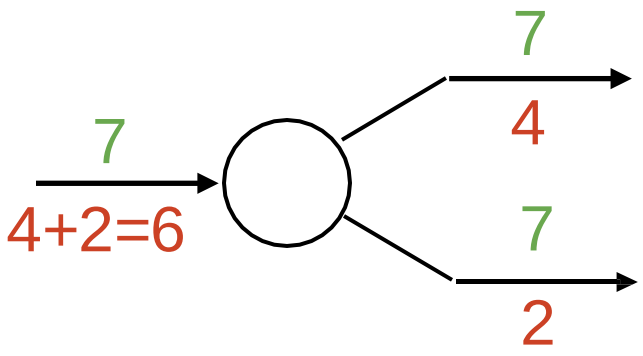
add gate: gradient distributor



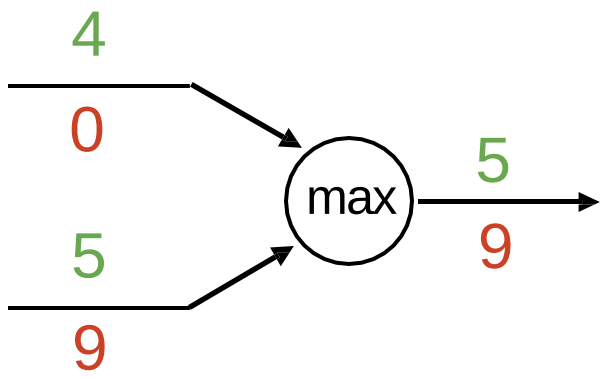
mul gate: “swap multiplier”



copy gate: gradient adder

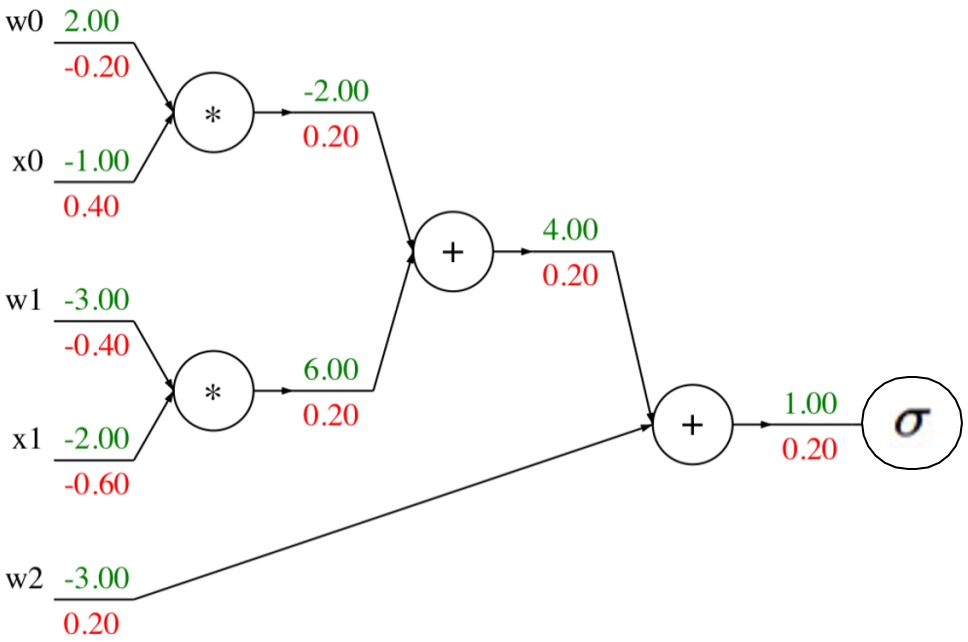


max gate: gradient router



反向传播

反向传播的简单代码实现



Forward pass:
Compute output

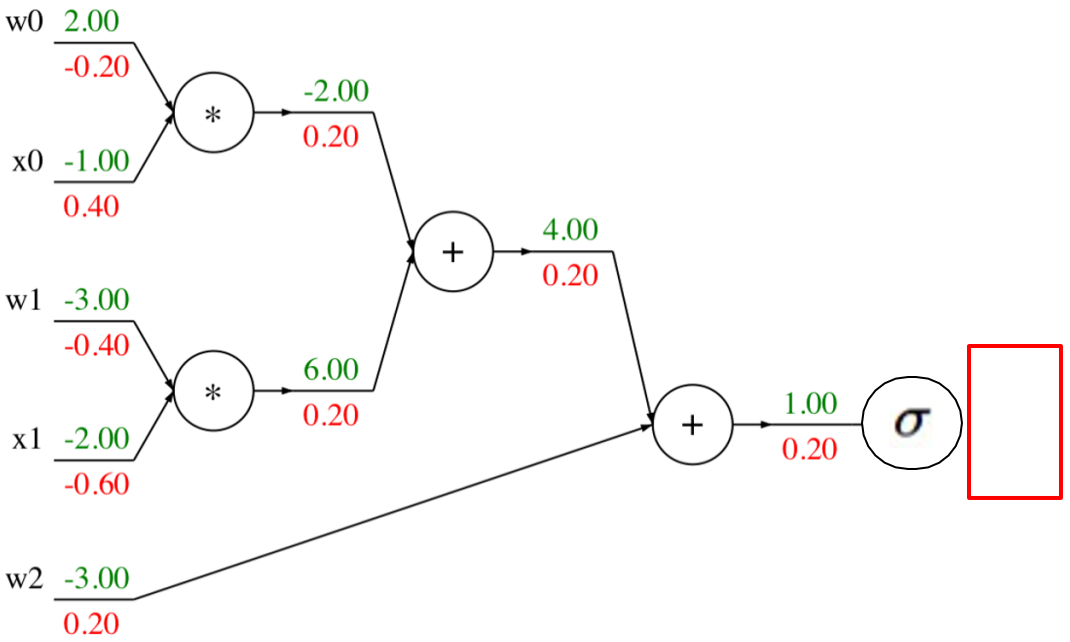
```
def f(w0, x0, w1, x1, w2):
  s0 = w0 * x0
  s1 = w1 * x1
  s2 = s0 + s1
  s3 = s2 + w2
  L = sigmoid(s3)
```

Backward pass:
Compute grads

```
grad_L = 1.0
grad_s3 = grad_L * (1 - L) * L
grad_w2 = grad_s3
grad_s2 = grad_s3
grad_s0 = grad_s2
grad_s1 = grad_s2
grad_w1 = grad_s1 * x1
grad_x1 = grad_s1 * w1
grad_w0 = grad_s0 * x0
grad_x0 = grad_s0 * w0
```

反向传播

反向传播的简单代码实现



Forward pass:
Compute output

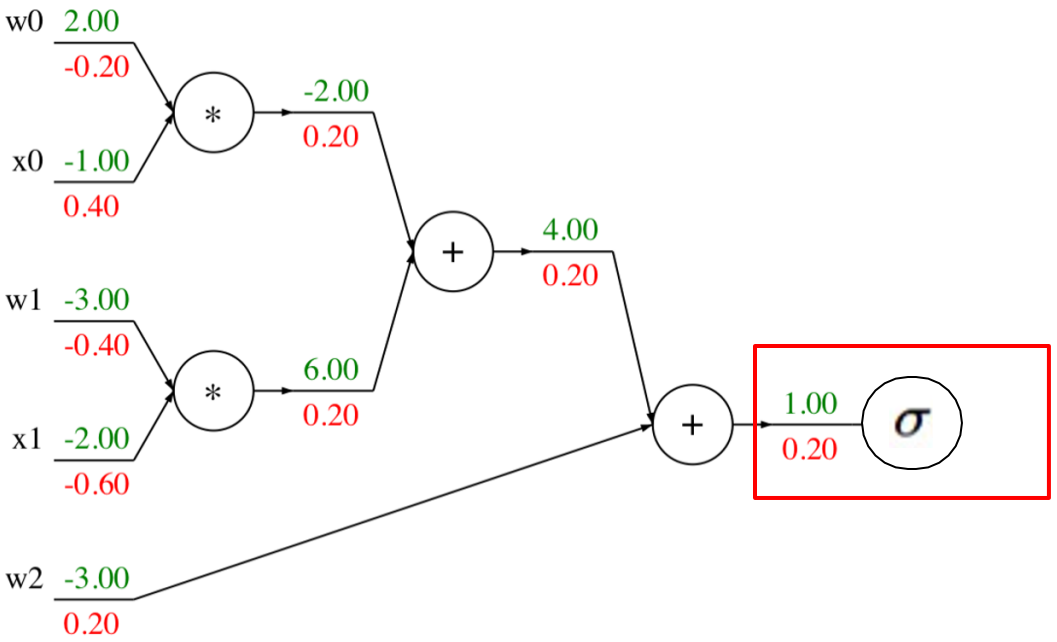
```
def f(w0, x0, w1, x1, w2):
  s0 = w0 * x0
  s1 = w1 * x1
  s2 = s0 + s1
  s3 = s2 + w2
  L = sigmoid(s3)
```

Base case

```
  grad_L = 1.0
  grad_s3 = grad_L * (1 - L) * L
  grad_w2 = grad_s3
  grad_s2 = grad_s3
  grad_s0 = grad_s2
  grad_s1 = grad_s2
  grad_w1 = grad_s1 * x1
  grad_x1 = grad_s1 * w1
  grad_w0 = grad_s0 * x0
  grad_x0 = grad_s0 * w0
```

反向传播

反向传播的简单代码实现



Forward pass:
Compute output

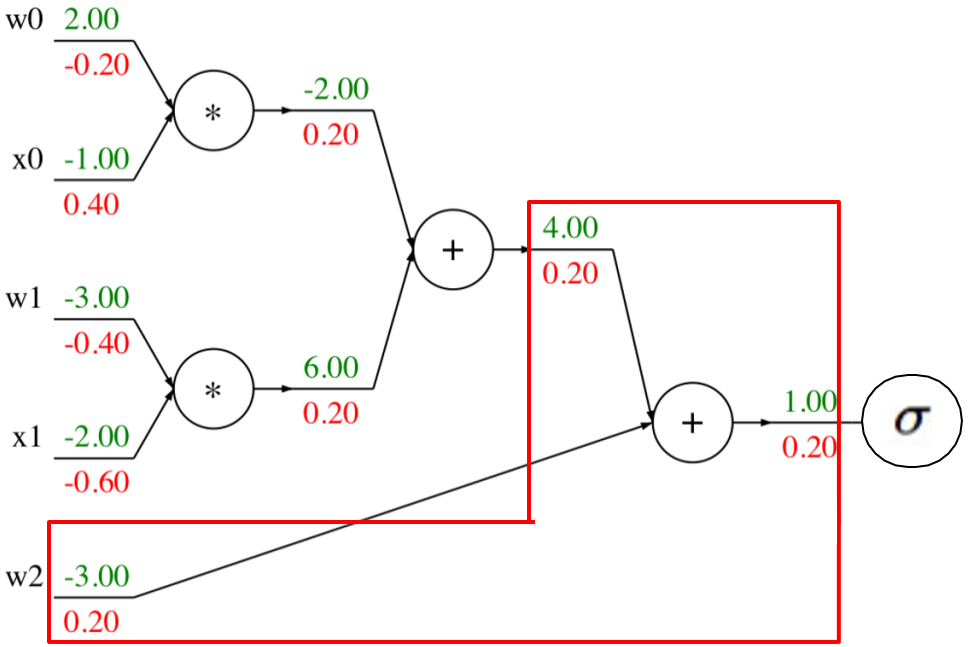
```
def f(w0, x0, w1, x1, w2):
  s0 = w0 * x0
  s1 = w1 * x1
  s2 = s0 + s1
  s3 = s2 + w2
  L = sigmoid(s3)
```

Sigmoid

```
  grad_L = 1.0
  grad_s3 = grad_L * (1 - L) * L
  grad_w2 = grad_s3
  grad_s2 = grad_s3
  grad_s0 = grad_s2
  grad_s1 = grad_s2
  grad_w1 = grad_s1 * x1
  grad_x1 = grad_s1 * w1
  grad_w0 = grad_s0 * x0
  grad_x0 = grad_s0 * w0
```

反向传播

反向传播的简单代码实现



Forward pass:
Compute output

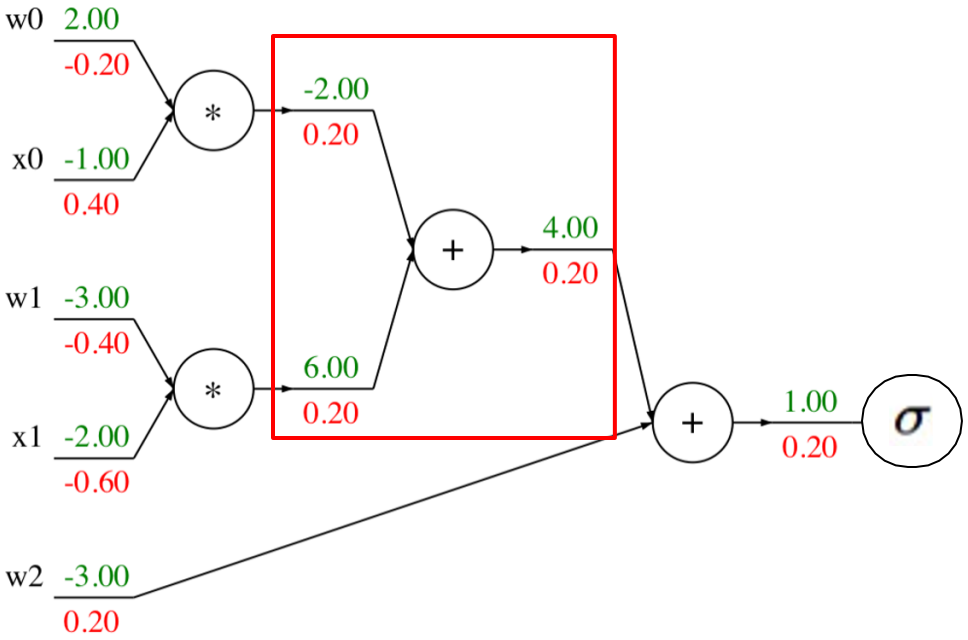
```
def f(w0, x0, w1, x1, w2):
  s0 = w0 * x0
  s1 = w1 * x1
  s2 = s0 + s1
  s3 = s2 + w2
  L = sigmoid(s3)
```

Add gate

```
  grad_L = 1.0
  grad_s3 = grad_L * (1 - L) * L
  grad_w2 = grad_s3
  grad_s2 = grad_s3
  grad_s0 = grad_s2
  grad_s1 = grad_s2
  grad_w1 = grad_s1 * x1
  grad_x1 = grad_s1 * w1
  grad_w0 = grad_s0 * x0
  grad_x0 = grad_s0 * w0
```

反向传播

反向传播的简单代码实现



Forward pass:
Compute output

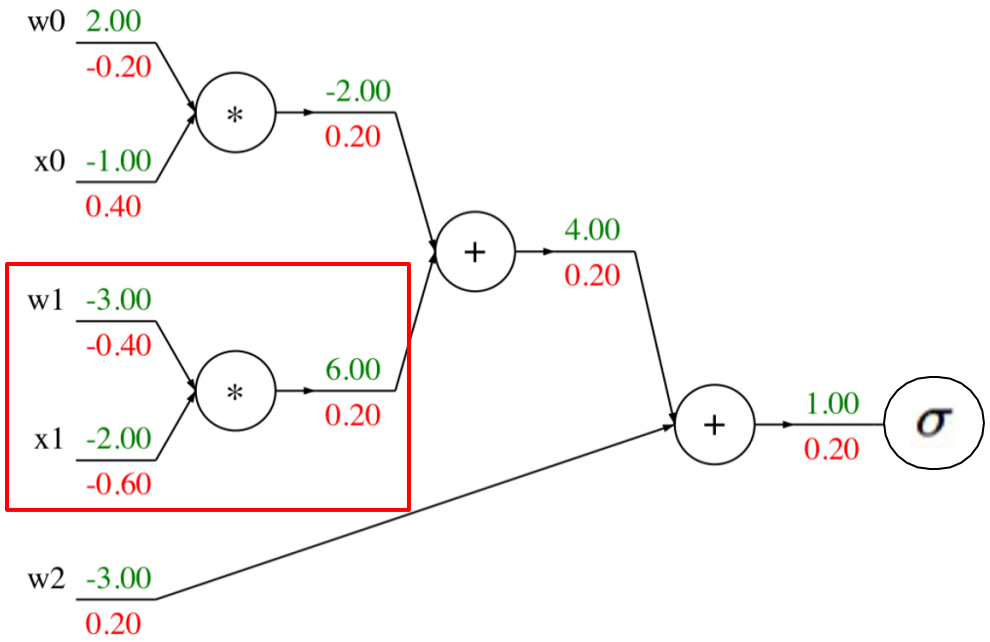
```
def f(w0, x0, w1, x1, w2):
  s0 = w0 * x0
  s1 = w1 * x1
  s2 = s0 + s1
  s3 = s2 + w2
  L = sigmoid(s3)

  grad_L = 1.0
  grad_s3 = grad_L * (1 - L) * L
  grad_w2 = grad_s3
  grad_s2 = grad_s3
  grad_s0 = grad_s2
  grad_s1 = grad_s2
  grad_w1 = grad_s1 * x1
  grad_x1 = grad_s1 * w1
  grad_w0 = grad_s0 * x0
  grad_x0 = grad_s0 * w0
```

Add gate

反向传播

反向传播的简单代码实现



Forward pass:
Compute output

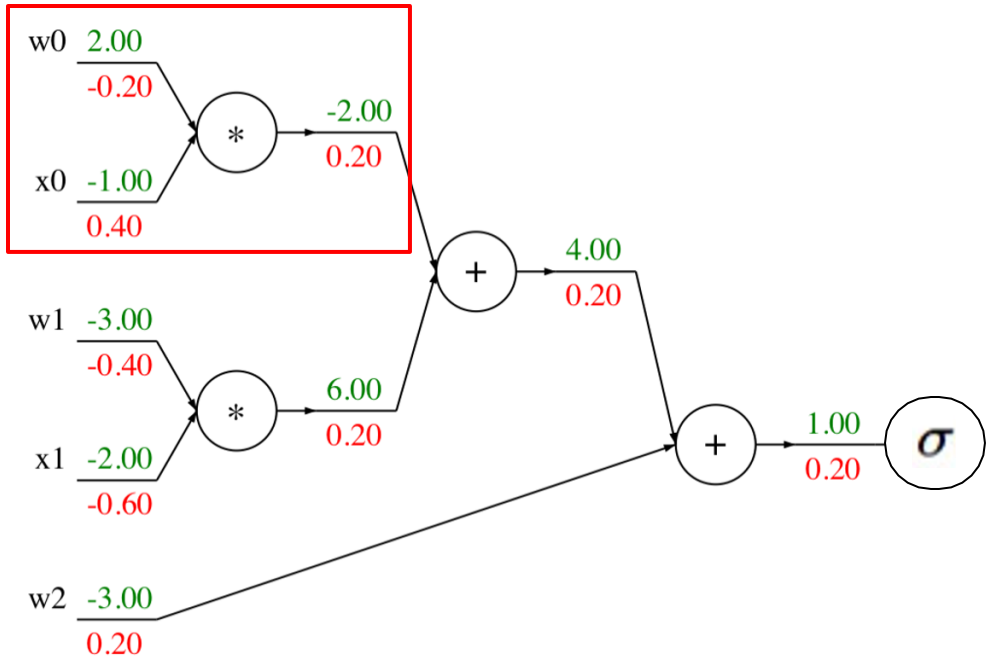
```
def f(w0, x0, w1, x1, w2):
  s0 = w0 * x0
  s1 = w1 * x1
  s2 = s0 + s1
  s3 = s2 + w2
  L = sigmoid(s3)
```

```
  grad_L = 1.0
  grad_s3 = grad_L * (1 - L) * L
  grad_w2 = grad_s3
  grad_s2 = grad_s3
  grad_s0 = grad_s2
  grad_s1 = grad_s2
  grad_w1 = grad_s1 * x1
  grad_x1 = grad_s1 * w1
  grad_w0 = grad_s0 * x0
  grad_x0 = grad_s0 * w0
```

Multiply gate

反向传播

反向传播的简单代码实现



Forward pass:
Compute output

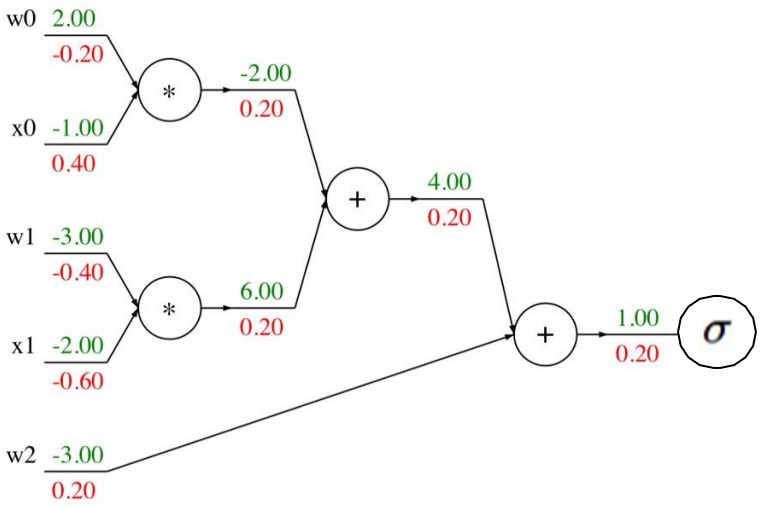
```
def f(w0, x0, w1, x1, w2):
  s0 = w0 * x0
  s1 = w1 * x1
  s2 = s0 + s1
  s3 = s2 + w2
  L = sigmoid(s3)
```

```
  grad_L = 1.0
  grad_s3 = grad_L * (1 - L) * L
  grad_w2 = grad_s3
  grad_s2 = grad_s3
  grad_s0 = grad_s2
  grad_s1 = grad_s2
  grad_w1 = grad_s1 * x1
  grad_x1 = grad_s1 * w1
  grad_w0 = grad_s0 * x0
  grad_x0 = grad_s0 * w0
```

Multiply gate

反向传播

反向传播的模块化实现

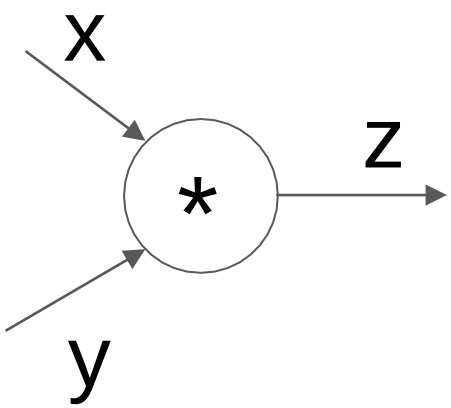



```
class ComputationalGraph(object):
    #...
    def forward(inputs):
        # 1. [pass inputs to input gates...]
        # 2. forward the computational graph:
        for gate in self.graph.nodes_topologically_sorted():
            gate.forward()
        return loss # the final gate in the graph outputs the loss
    def backward():
        for gate in reversed(self.graph.nodes_topologically_sorted()):
            gate.backward() # little piece of backprop (chain rule applied)
        return inputs_gradients
```

反向传播

南京大学 NANJING UNIVERSITY

反向传播的模块化实现：前向/反向传播



(x,y,z 都是标量)

```
class Multiply(torch.autograd.Function):
  @staticmethod
  def forward(ctx, x, y):
    ctx.save_for_backward(x, y)
    z = x * y
    return z
  @staticmethod
  def backward(ctx, grad_z):
    x, y = ctx.saved_tensors
    grad_x = y * grad_z    # dz/dx * dL/dz
    grad_y = x * grad_z    # dz/dy * dL/dz
    return grad_x, grad_y
```

需要缓存一些值以供反向传播使用

上游梯度

上游梯度和局部梯度相乘

反向传播

- 截至目前，我们讨论的都是面向标量的反向传播
- 面向向量的反向传播是怎样的?

反向传播

- **一元函数的导数** $f : \mathbb{R} \to \mathbb{R}$

Scalar to Scalar

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

常规导数：

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

如果x变化很小，y会变化多少?

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

反向传播

- **多元函数的导数** $f: \mathbb{R}^n \to \mathbb{R}$

Vector to Scalar

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

导数是梯度：

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

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

$$\nabla_{\boldsymbol{x}} \stackrel{\text{def}}{=} \left[\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \cdots, \frac{\partial}{\partial x_n}\right]^T = \frac{\partial}{\partial \boldsymbol{x}}$$

$$f(\vec{x}) \approx f(\vec{x}_0) + \nabla f(\vec{x}_0) \cdot (\vec{x} - \vec{x}_0)$$

对于x的每个元素，如果它变化很小，那么y会变化多少?

反向传播

- **向量函数的导数** $f : \mathbb{R}^n \to \mathbb{R}^m$

Vector to Vector

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

导数是雅可比矩阵

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

$$\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_j}$$

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

意义：如果x的每个元素产生一个小的变化，y的每个元素会产生的变化的大小

$$f(\vec{x}) \approx f(\vec{x}_k) + J(\vec{x}_k)(\vec{x} - \vec{x}_k)$$

反向传播

南京大学 NANJING UNIVERSITY

向量的导数

Scalar to Scalar

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

常规导数：

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

意义：如果x产生一个小的变化，y会产生的变化的大小

Vector to Scalar

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

导数是梯度：

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

意义：如果x的每个元素产生一个小的变化，y会产生的变化的大小

Vector to Vector

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

导数是雅可比矩阵

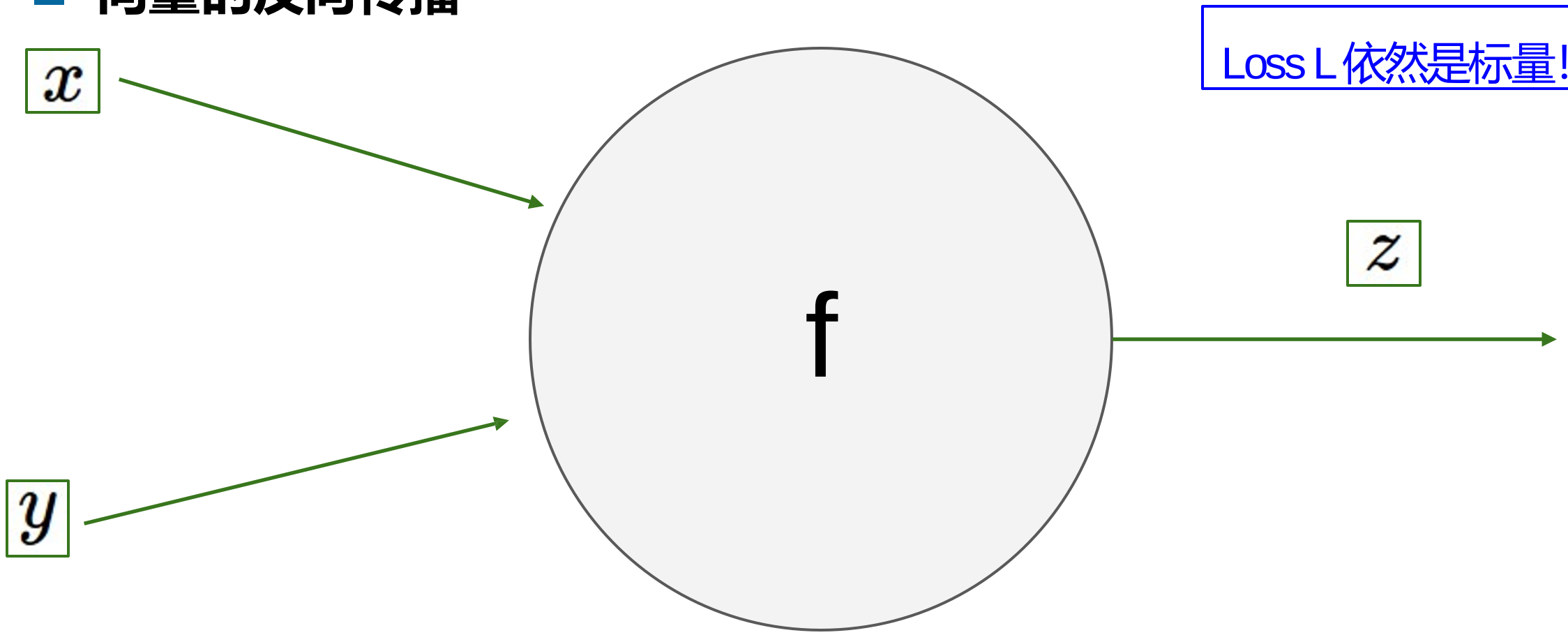
$$\frac{\partial y}{\partial x} \in \mathbb{R}^{N \times M} \quad \left(\frac{\partial y}{\partial x}\right)_{n,m} = \frac{\partial y_m}{\partial x_n}$$

意义：如果x的每个元素产生一个小的变化，y的每个元素会产生的变化的大小

反向传播

南京大学 NANJING UNIVERSITY

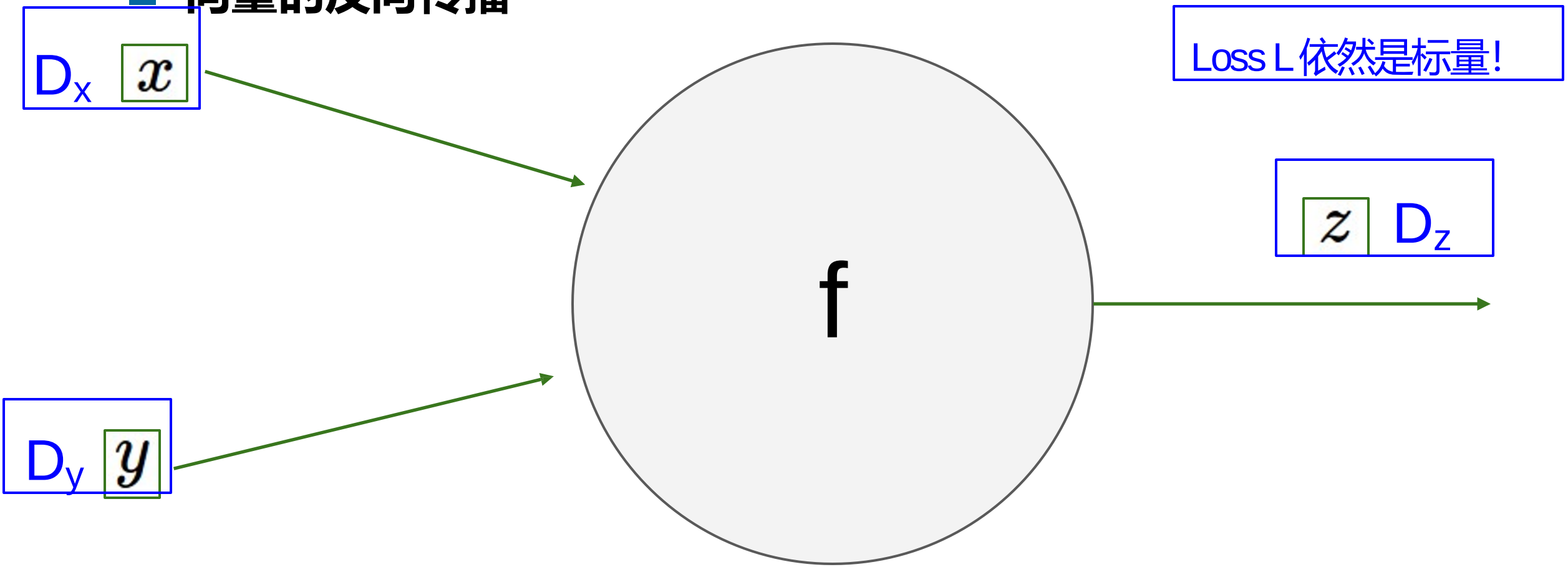
向量的反向传播



Loss L 依然是标量!

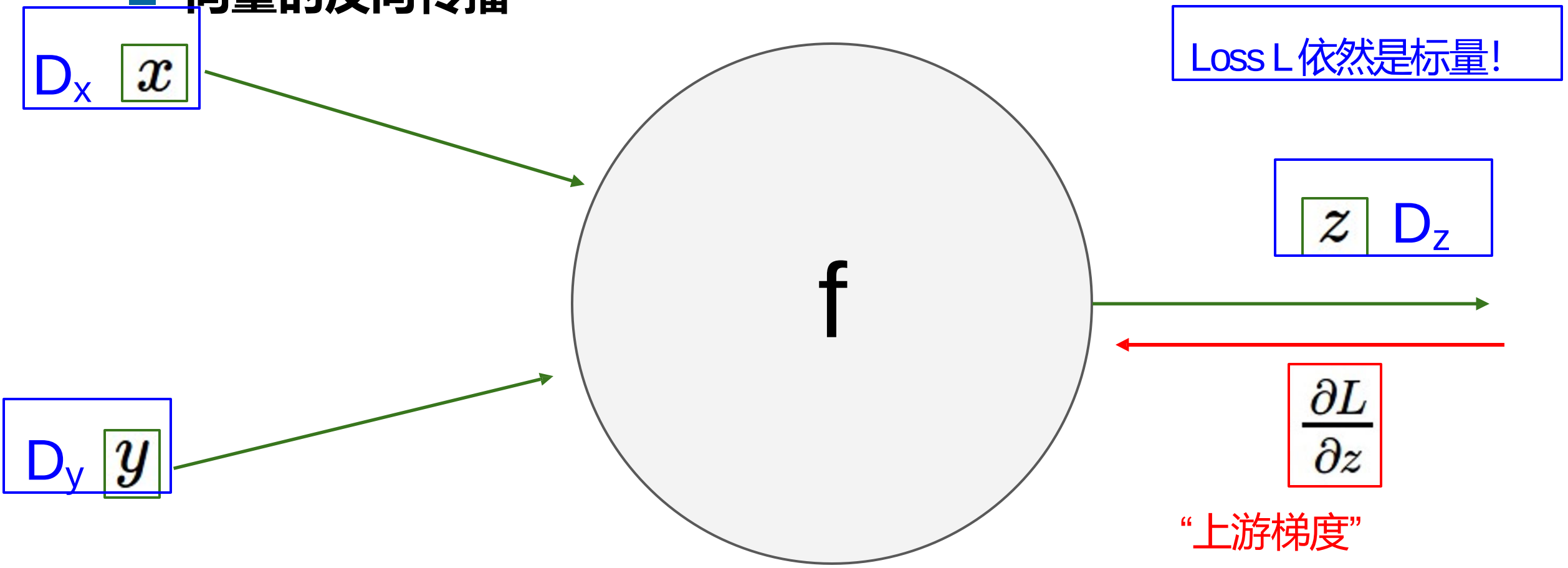
反向传播

- 向量的反向传播



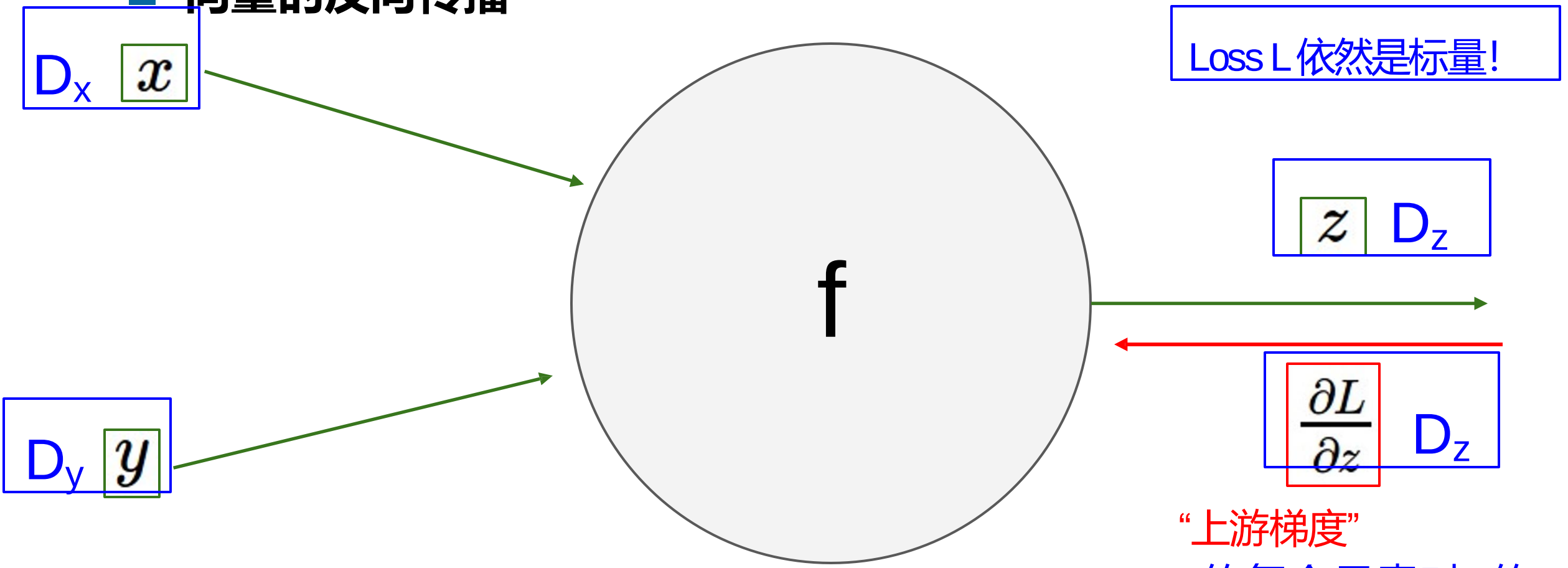
反向传播

- 向量的反向传播



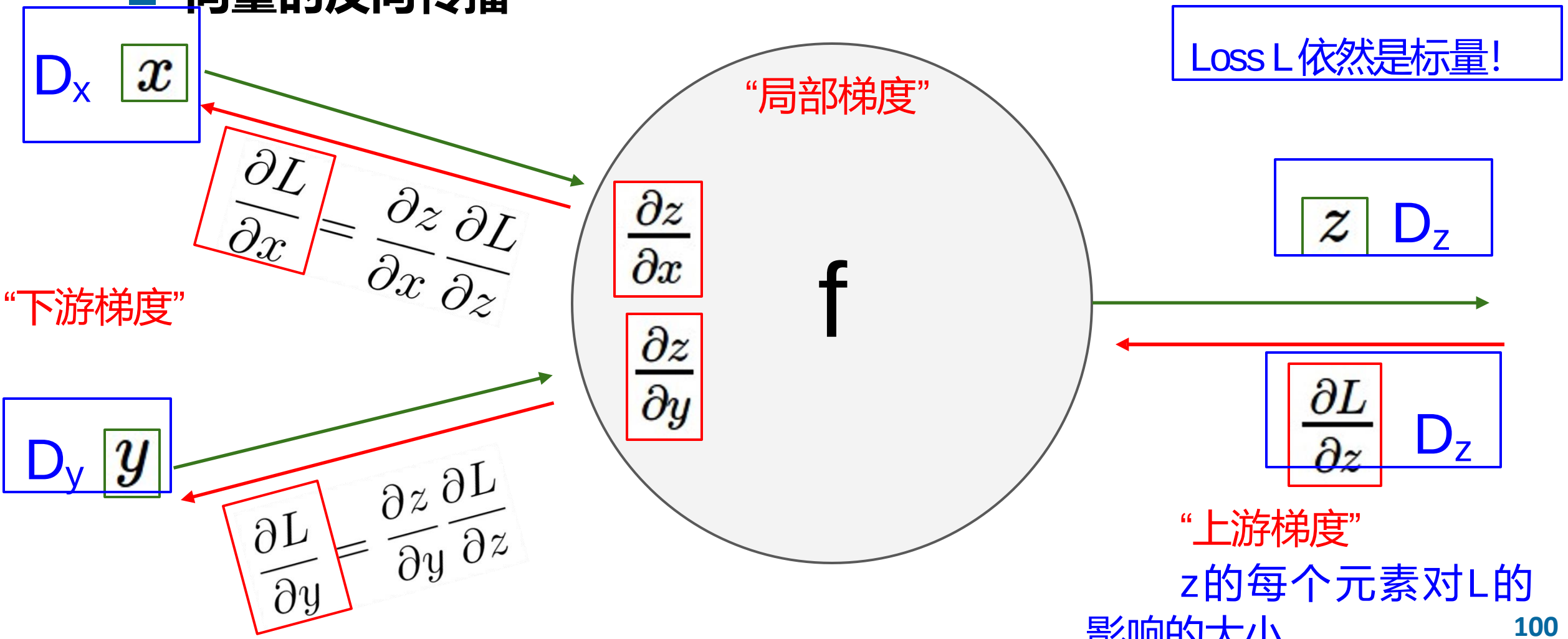
反向传播

- 向量的反向传播



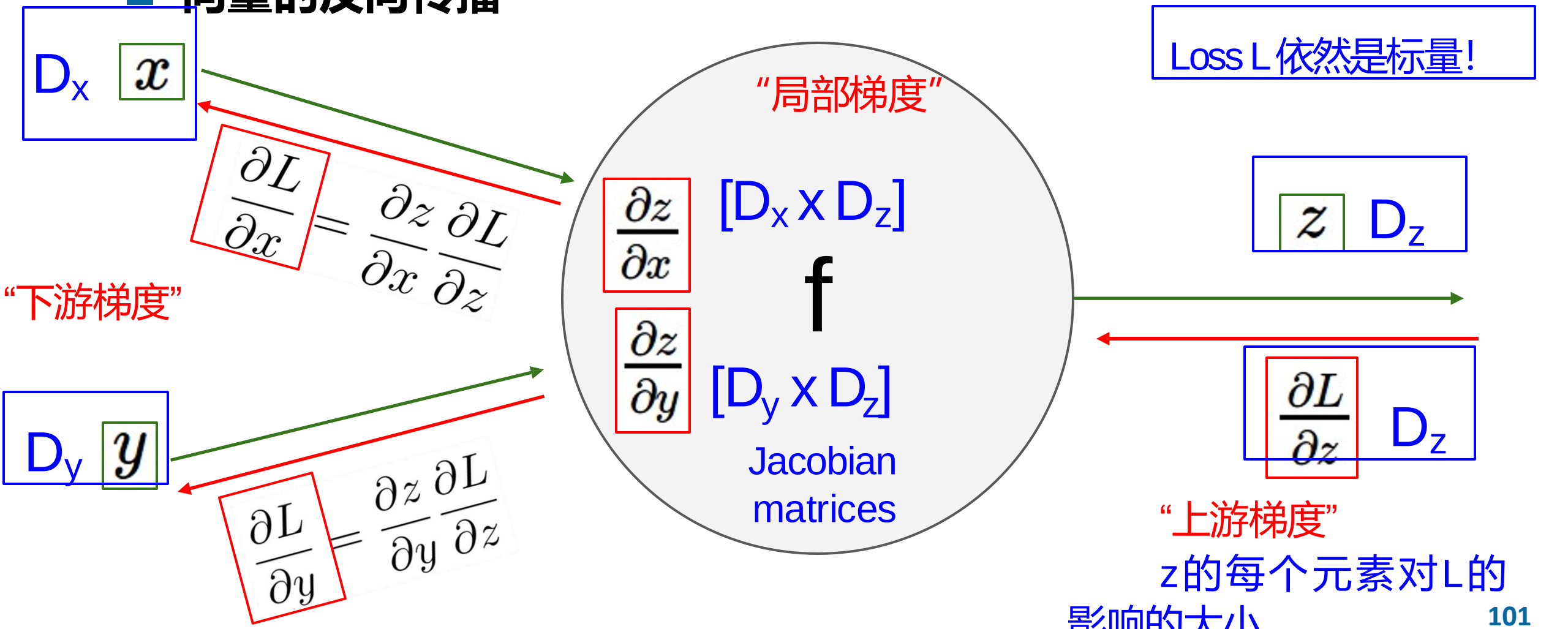
反向传播

向量的反向传播



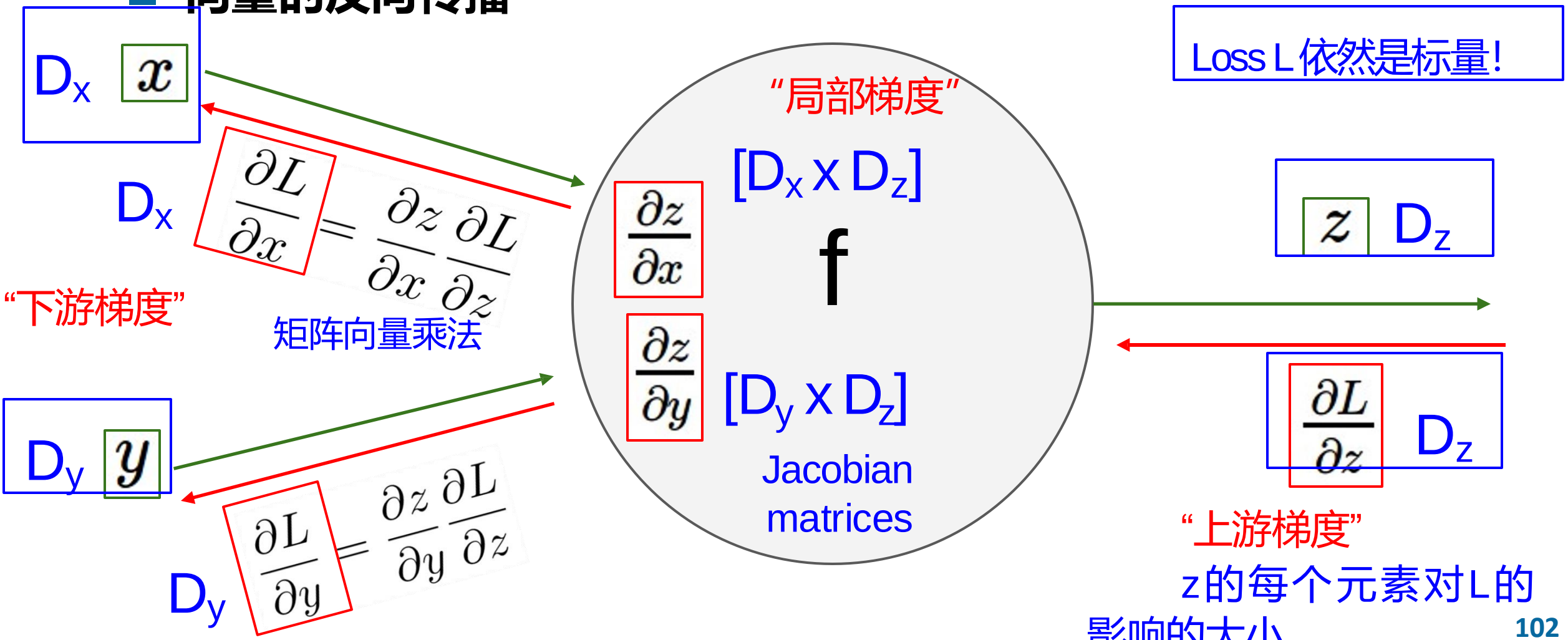
反向传播

向量的反向传播



反向传播

向量的反向传播

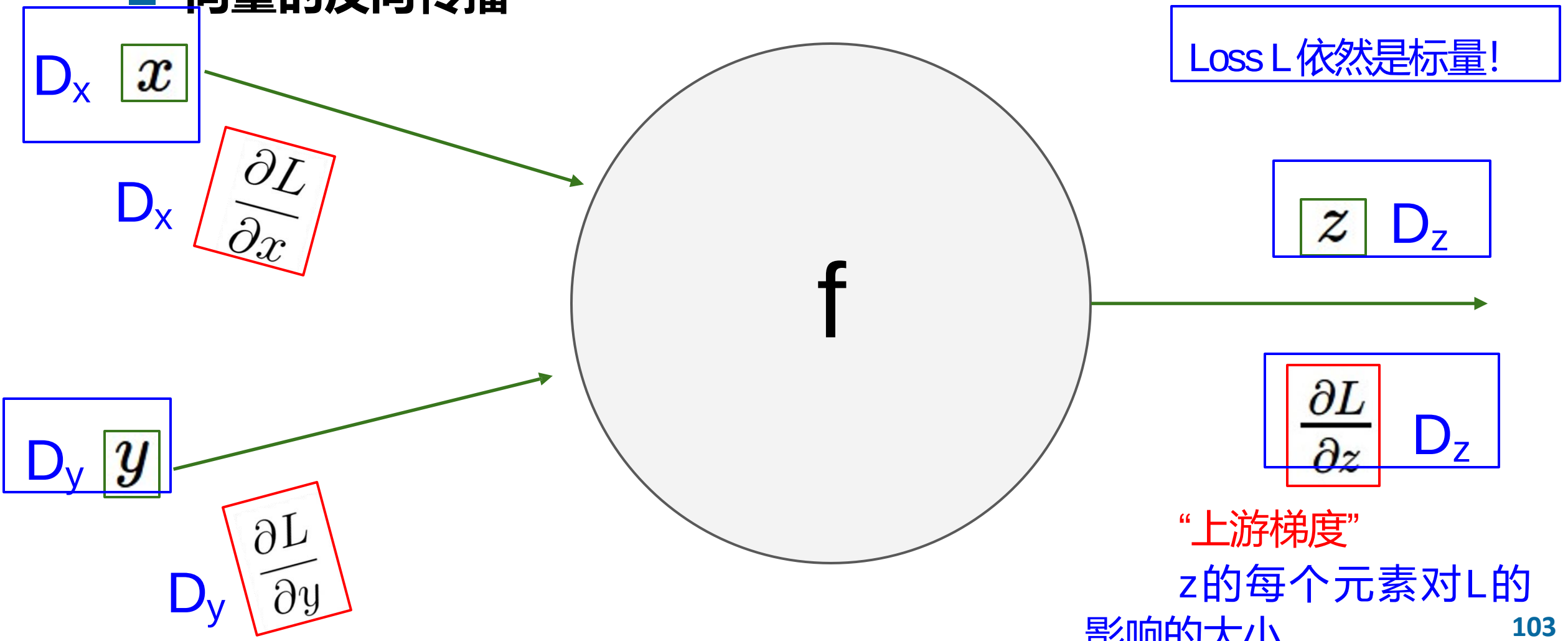


反向传播

南京大学 NANJING UNIVERSITY

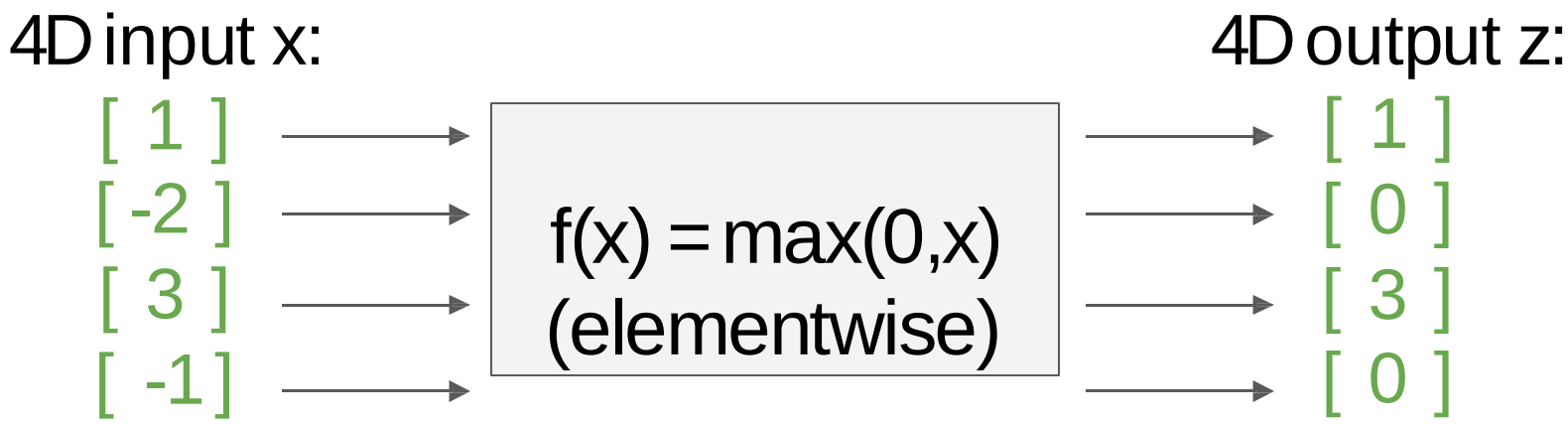
- 向量的反向传播

变量的梯度与原始变量具有相同的维度



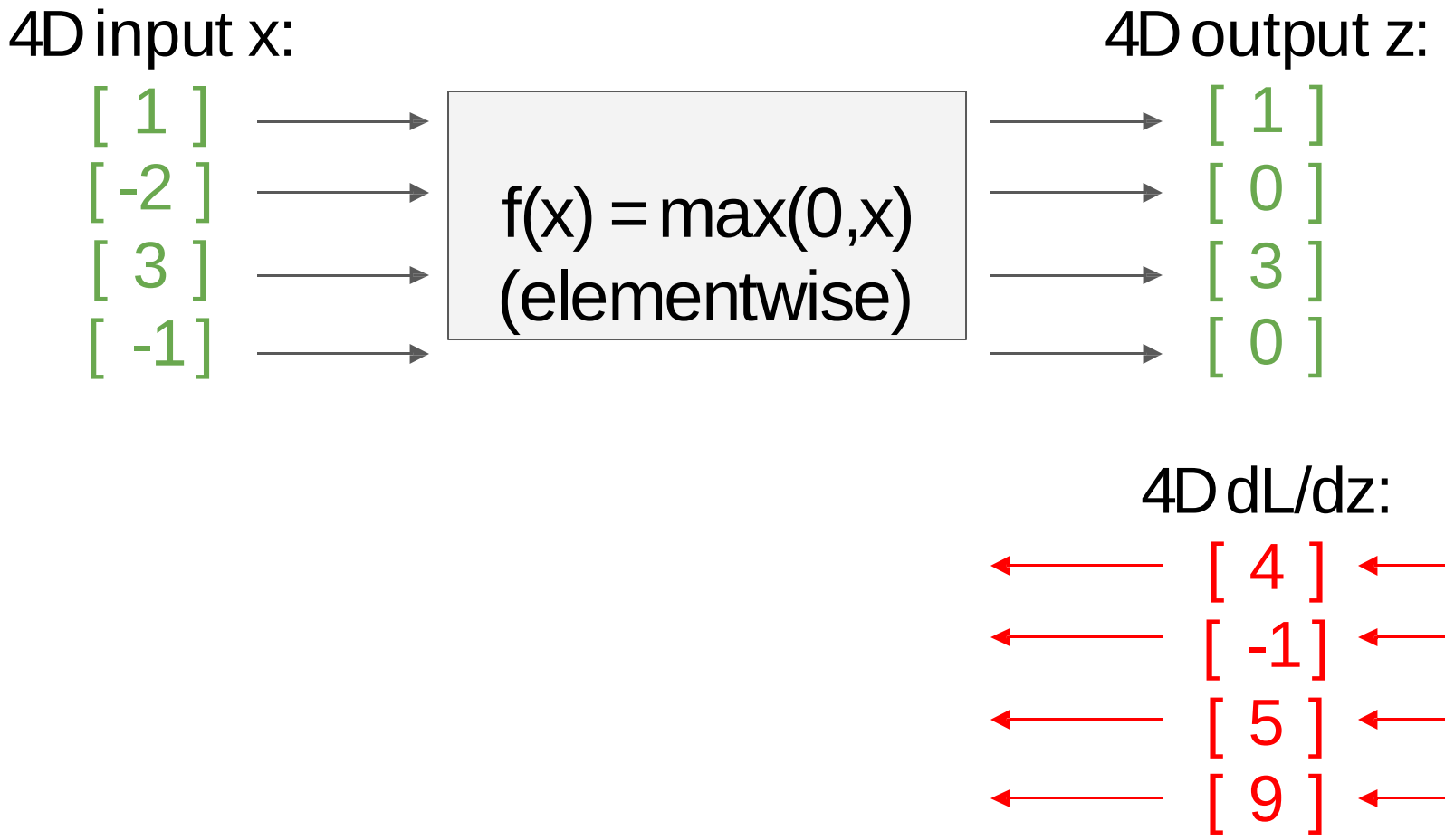
反向传播

向量的反向传播



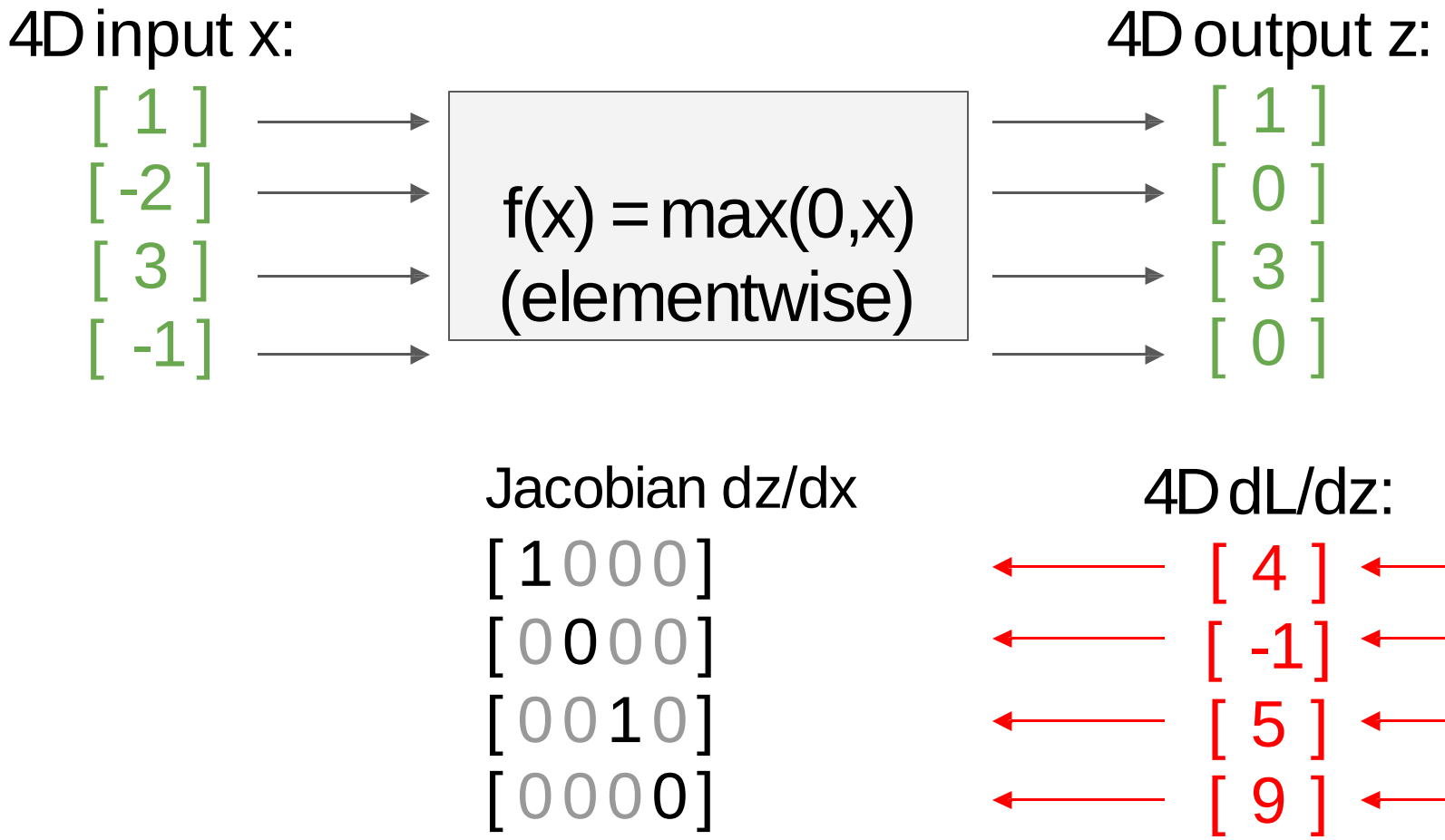
反向传播

- **向量的反向传播**



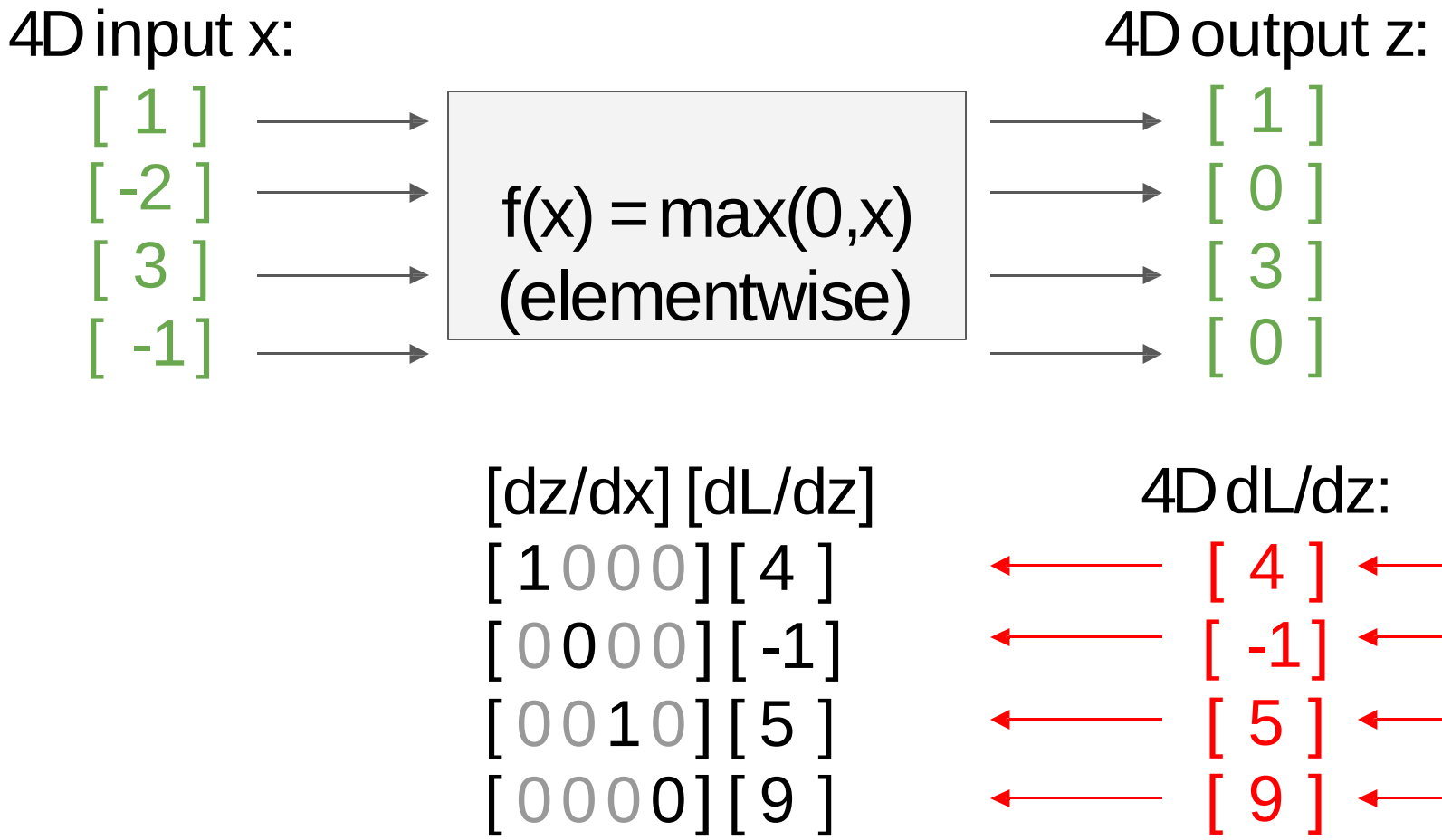
反向传播

向量的反向传播



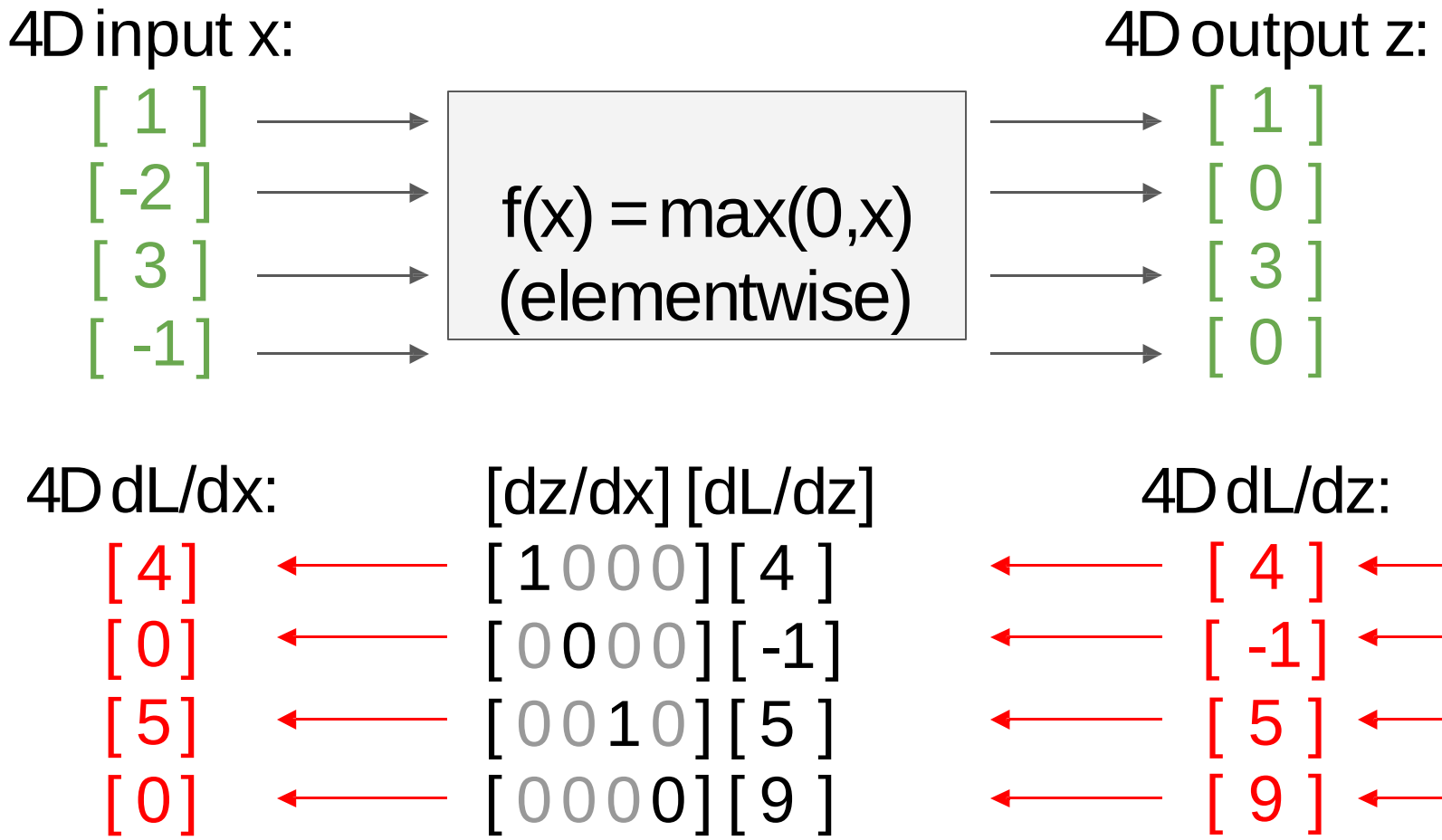
反向传播

向量的反向传播



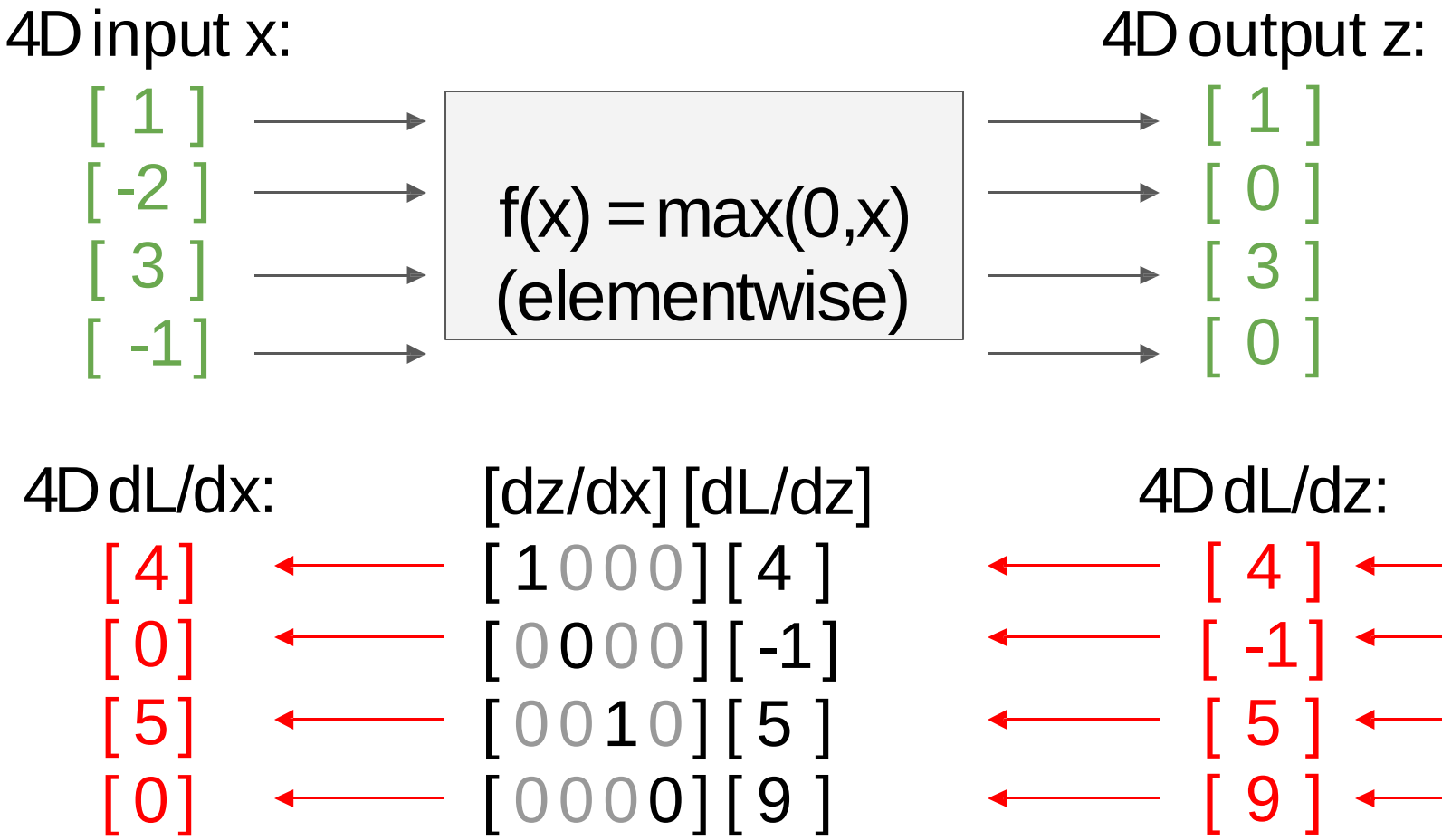
反向传播

向量的反向传播



反向传播

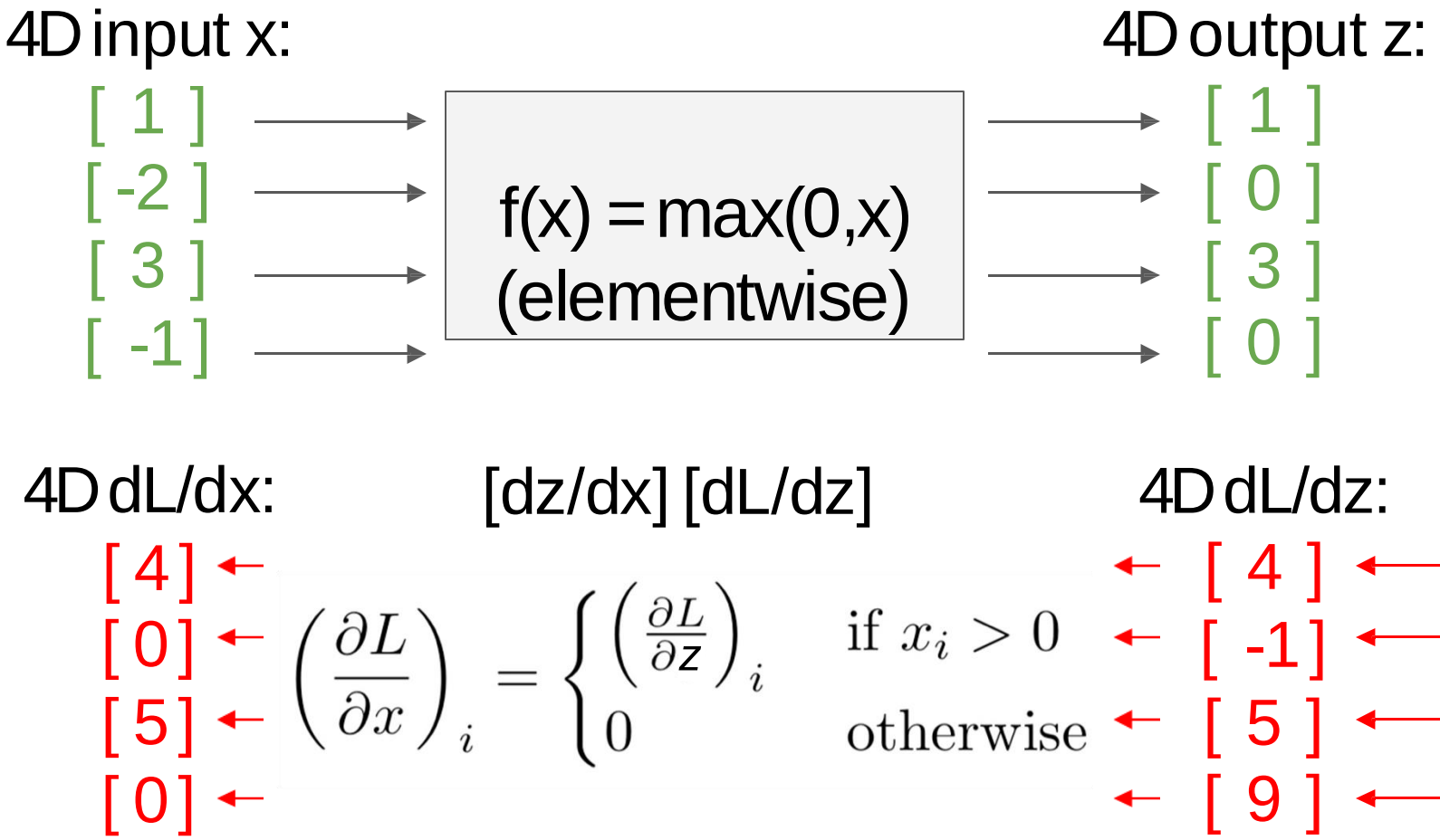
向量的反向传播



- 雅可比矩阵是稀疏的!
- 非对角线元素总是为零!
- 不要显式计算雅可比矩阵，而是使用隐式乘法

反向传播

向量的反向传播



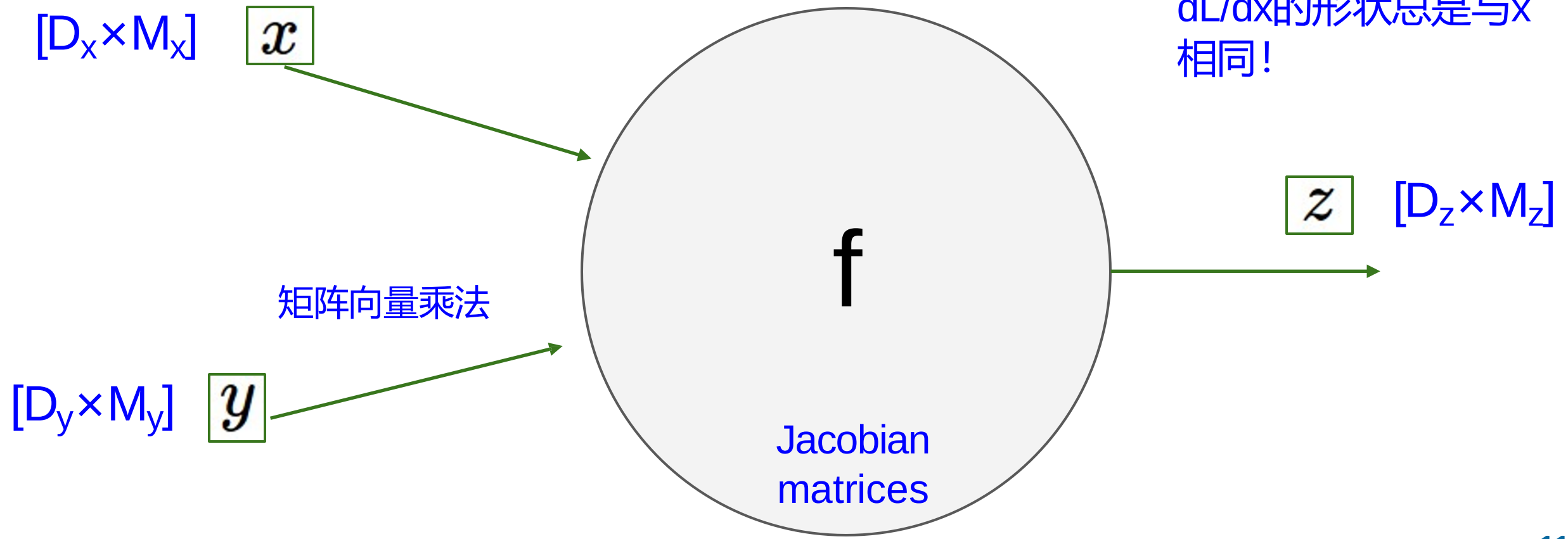
- 雅可比矩阵是稀疏的!
- 非对角线元素总是为零!
- 不要显式计算雅可比矩阵，而是使用隐式乘法

反向传播

- **向量的反向传播**

Loss L 依然是标量!

dL/dx的形状总是与x相同!



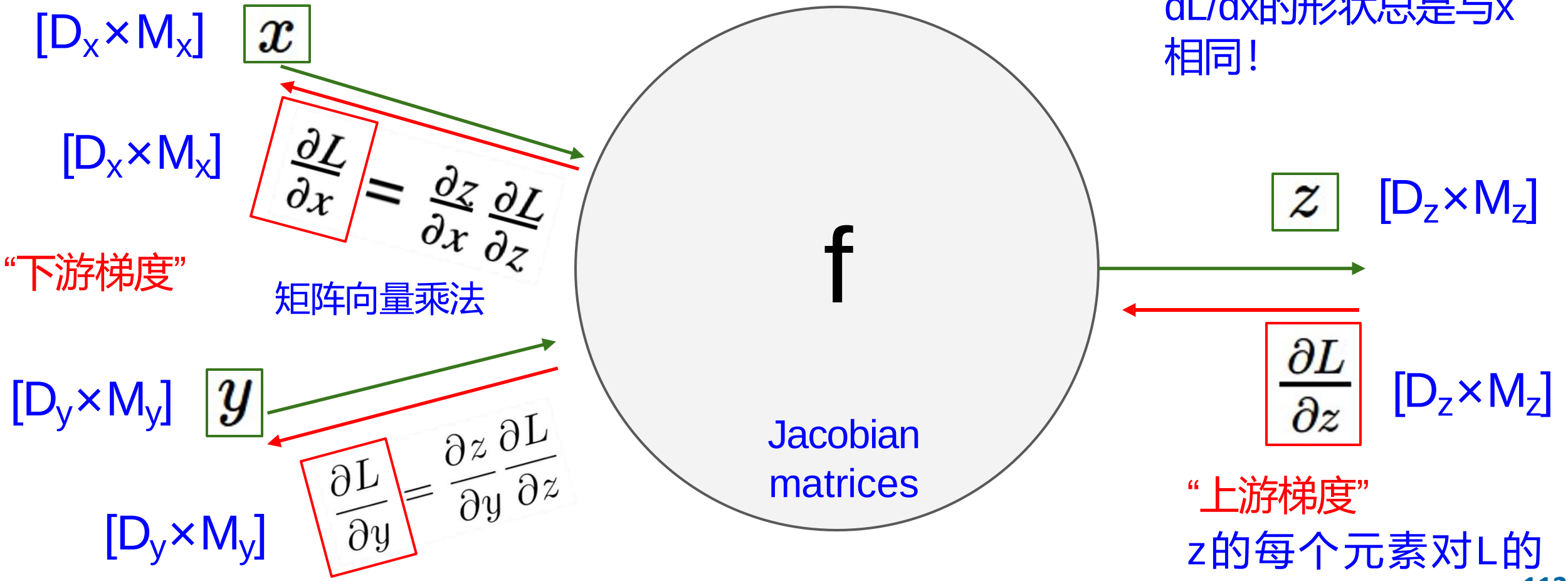
反向传播

南京大学 NANJING UNIVERSITY

向量的反向传播

Loss L 依然是标量!

dL/dx的形状总是与x相同!

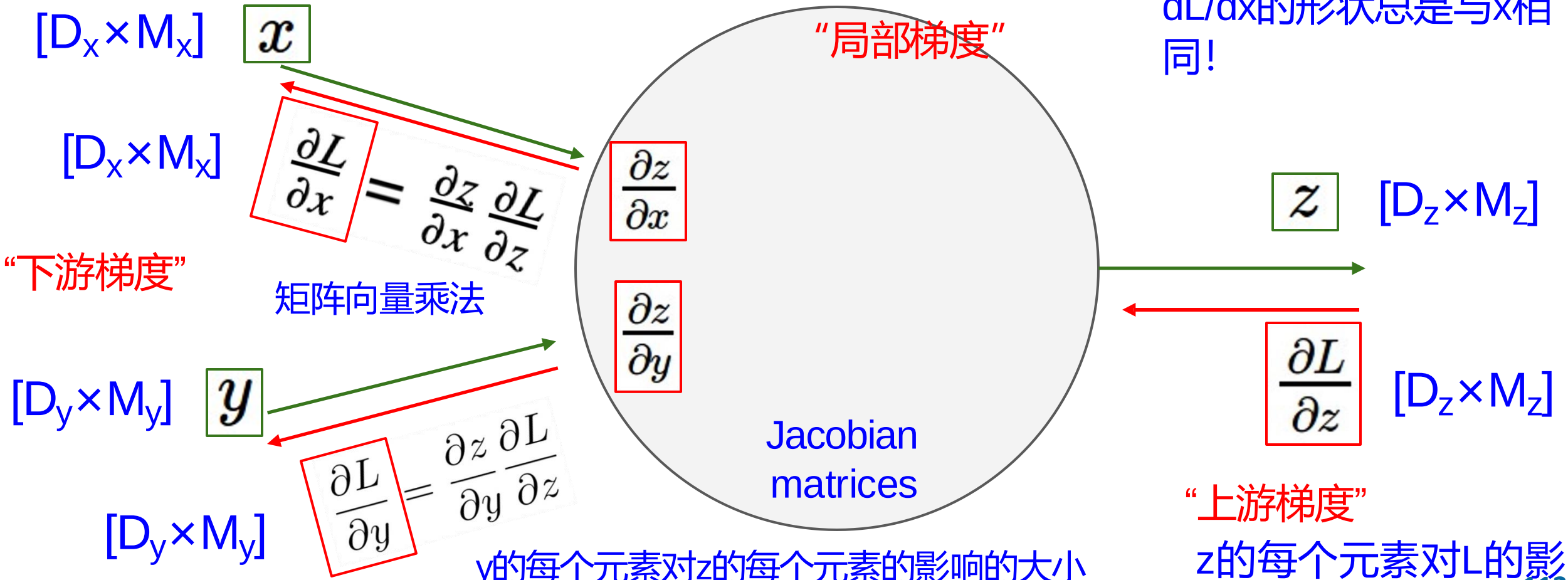


反向传播

向量的反向传播

Loss L 依然是标量!

dL/dx的形状总是与x相同!

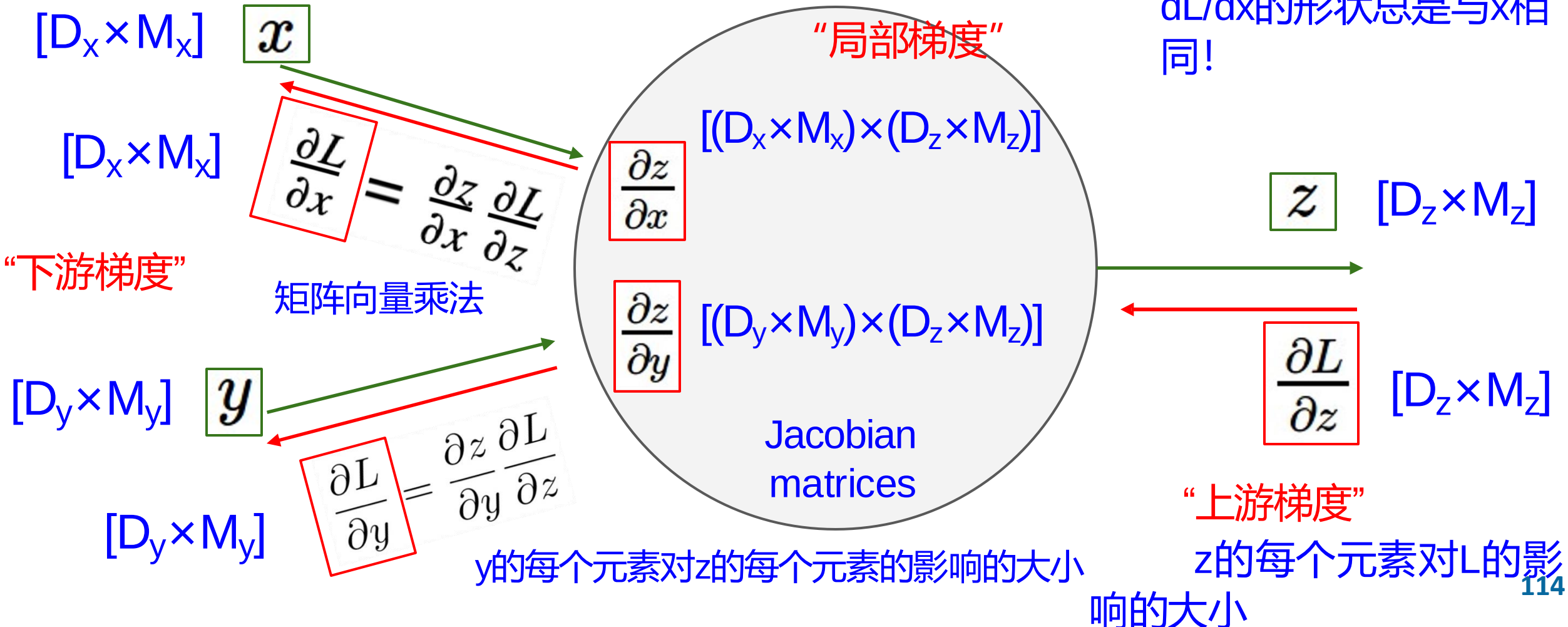


反向传播

向量的反向传播

Loss L 依然是标量!

dL/dx的形状总是与x相同!

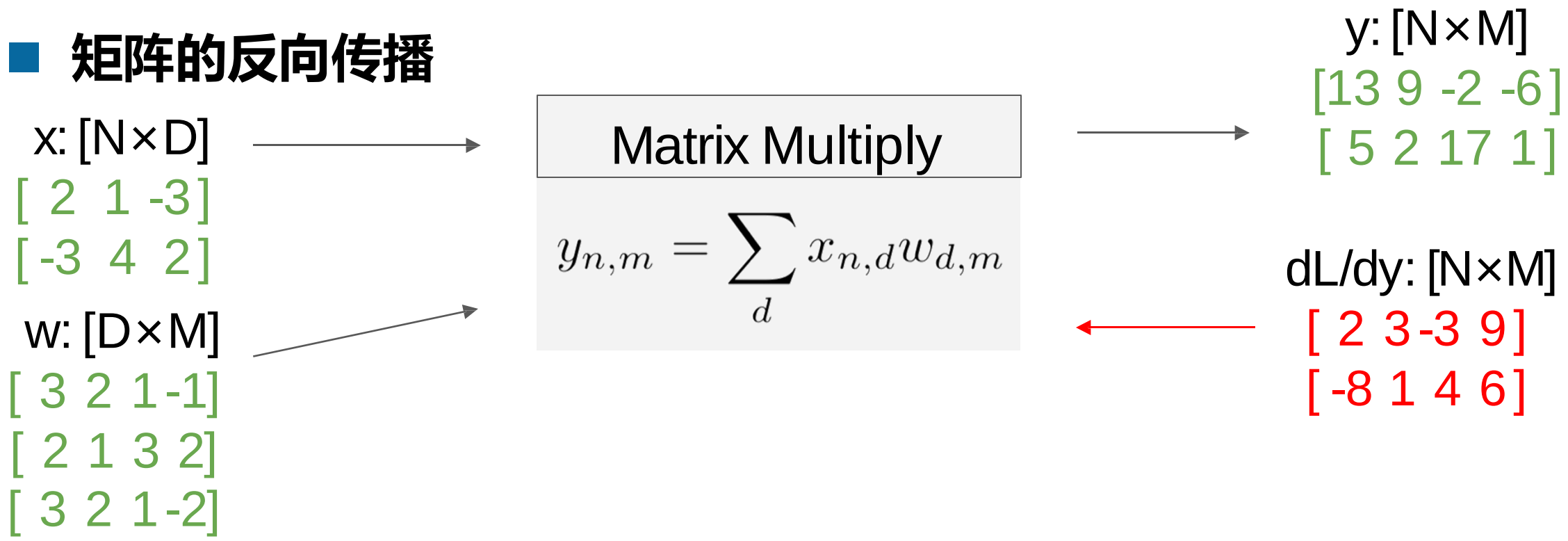


y的每个元素对z的每个元素的影响的大小

z的每个元素对L的影响的大小

反向传播

矩阵的反向传播



反向传播

矩阵的反向传播

x: [N×D]
[2 1 -3]
[-3 4 2]

w: [D×M]
[3 2 1 -1]
[2 1 3 2]
[3 2 1 -2]

Matrix Multiply

$$y_{n,m} = \sum_d x_{n,d} w_{d,m}$$

y: [N×M]
[13 9 -2 -6]
[5 2 17 1]

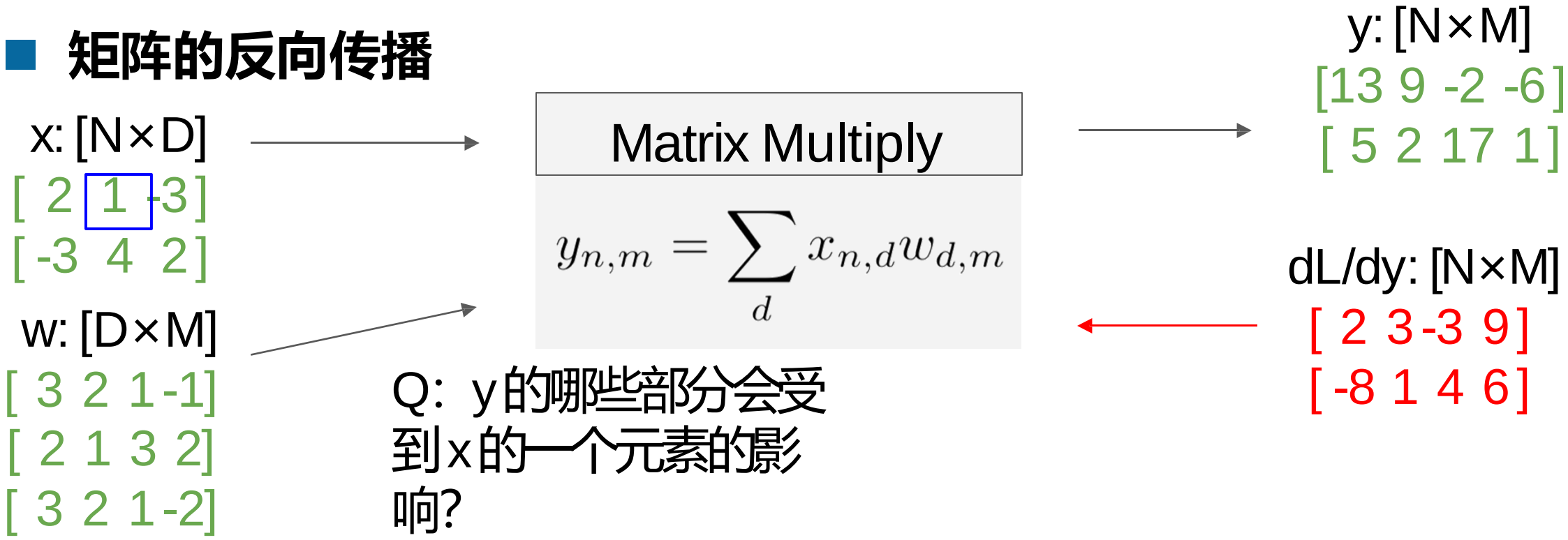
dL/dy: [N×M]
[2 3 -3 9]
[-8 1 4 6]

Jacobians:
dy/dx: [(N×D)×(N×M)]
dy/dw: [(D×M)×(N×M)]

当 N=64, D=M=4096 时，每个雅可比矩阵需要占用大概256GB内存进行计算

反向传播

矩阵的反向传播



反向传播

矩阵的反向传播

x: [N×D]

[2 1 -3]

[-3 4 2]

w: [D×M]

[3 2 1 -1]

[2 1 3 2]

[3 2 1 -2]

Matrix Multiply

$$y_{n,m} = \sum_d x_{n,d} w_{d,m}$$

y: [N×M]

[13 9 -2 -6]

[5 2 17 1]

dL/dy: [N×M]

[2 3 -3 9]

[-8 1 4 6]

Q：y的哪些部分会受到x的一个元素的影响?

A: $x_{n,d}$ 会影响 y 的整个行 $y_{n,\cdot}$

$$\frac{\partial L}{\partial x_{n,d}} = \sum_m \frac{\partial L}{\partial y_{n,m}} \frac{\partial y_{n,m}}{\partial x_{n,d}}$$

反向传播

矩阵的反向传播

x: [N×D]

[2 1 -3]

[-3 4 2]

w: [D×M]

[3 2 1 -1]

[2 1 3 2]

[3 2 1 -2]

Matrix Multiply

$$y_{n,m} = \sum_d x_{n,d} w_{d,m}$$

y: [N×M]

[13 9 -2 -6]

[5 2 17 1]

dL/dy: [N×M]

[2 3 -3 9]

[-8 1 4 6]

Q: y的哪些部分会受到x的一个元素的影响?

A: $x_{n,d}$ 会影响 y 的整个行 $y_{n,\cdot}$

Q: $x_{n,d}$ 会对 $y_{n,m}$ 产生多大影响?

$$\frac{\partial L}{\partial x_{n,d}} = \sum_m \frac{\partial L}{\partial y_{n,m}} \frac{\partial y_{n,m}}{\partial x_{n,d}}$$

反向传播

矩阵的反向传播

x: [N×D]

[2 1 -3]

[-3 4 2]

w: [D×M]

[3 2 1 -1]

[2 1 3 2]

[3 2 1 -2]

Matrix Multiply

$$y_{n,m} = \sum_d x_{n,d} w_{d,m}$$

y: [N×M]

[13 9 -2 -6]

[5 2 17 1]

dL/dy: [N×M]

[2 3 -3 9]

[-8 1 4 6]

Q: y的哪些部分会受到x的一个元素的影响?

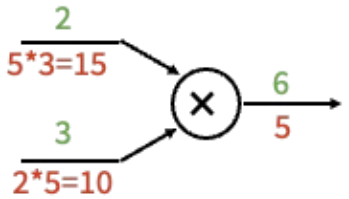
A: $x_{n,d}$ 会影响 y 的整个行 $y_{n,\cdot}$

Q: $x_{n,d}$ 会对 $y_{n,m}$ 产生多大影响?

A: $w_{d,m}$

$$\frac{\partial L}{\partial x_{n,d}} = \sum_m \frac{\partial L}{\partial y_{n,m}} \frac{\partial y_{n,m}}{\partial x_{n,d}} = \sum_m \frac{\partial L}{\partial y_{n,m}} w_{d,m}$$

mul gate: "swap multiplier"



反向传播

矩阵的反向传播

x: [N×D]

[2 1 -3]

[-3 4 2]

w: [D×M]

[3 2 1 -1]

[2 1 3 2]

[3 2 1 -2]

Matrix Multiply

$$y_{n,m} = \sum_d x_{n,d} w_{d,m}$$

y: [N×M]

[13 9 -2 -6]

[5 2 17 1]

dL/dy: [N×M]

[2 3 -3 9]

[-8 1 4 6]

Q: y的哪些部分会受到x的一个元素的影响?

A: $x_{n,d}$ 会影响 y 的整个行 $y_{n,.}$

Q: $x_{n,d}$ 会对 $y_{n,m}$ 产生多大影响?

A: $w_{d,m}$

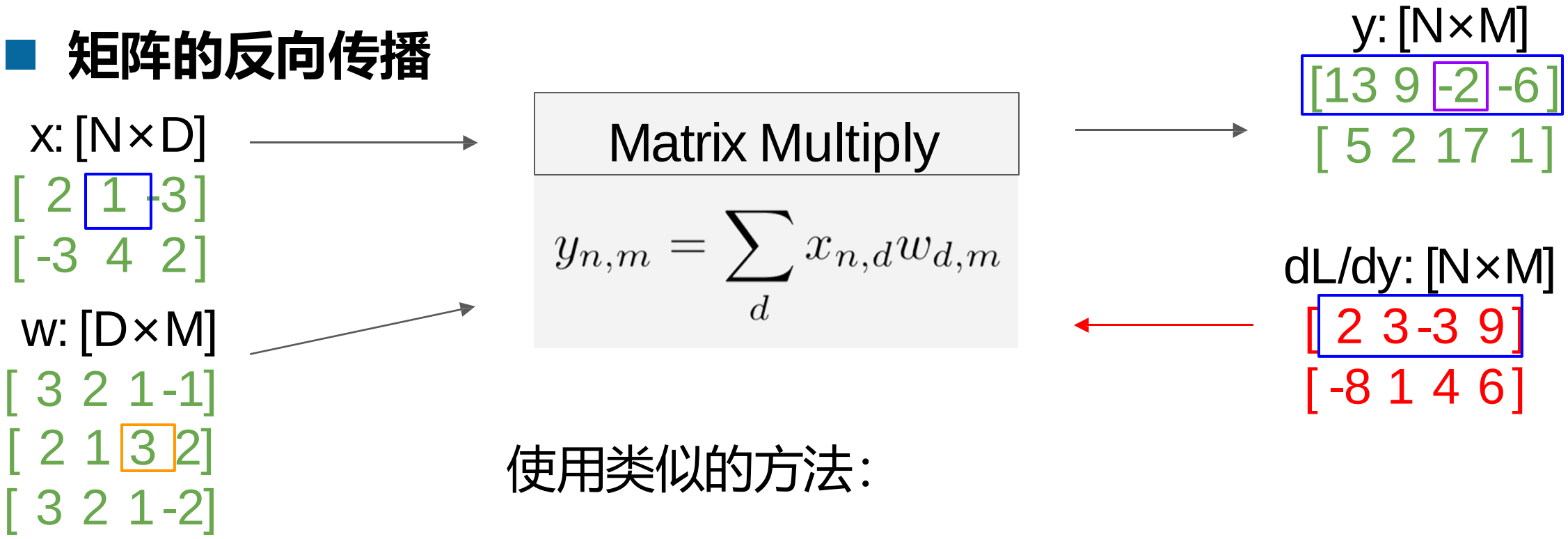
[N×D] [N×M] [M×D]

$$\frac{\partial L}{\partial x} = \left(\frac{\partial L}{\partial y}\right) w^T$$

$$\frac{\partial L}{\partial x_{n,d}} = \sum_m \frac{\partial L}{\partial y_{n,m}} \frac{\partial y_{n,m}}{\partial x_{n,d}} = \sum_m \frac{\partial L}{\partial y_{n,m}} w_{d,m}$$

反向传播

矩阵的反向传播



使用类似的方法：

[N×D] [N×M] [M×D]

$$\frac{\partial L}{\partial x} = \left(\frac{\partial L}{\partial y}\right) w^T$$

[D×M] [D×N] [N×M]

$$\frac{\partial L}{\partial w} = x^T \left(\frac{\partial L}{\partial y}\right)$$

这些公式很容易记住！
使矩阵形状匹配！