最优化方法

深度学习及其优化算法

张伯雷

南京邮电大学 计算机学院

bolei.zhang@njupt.edu.cn

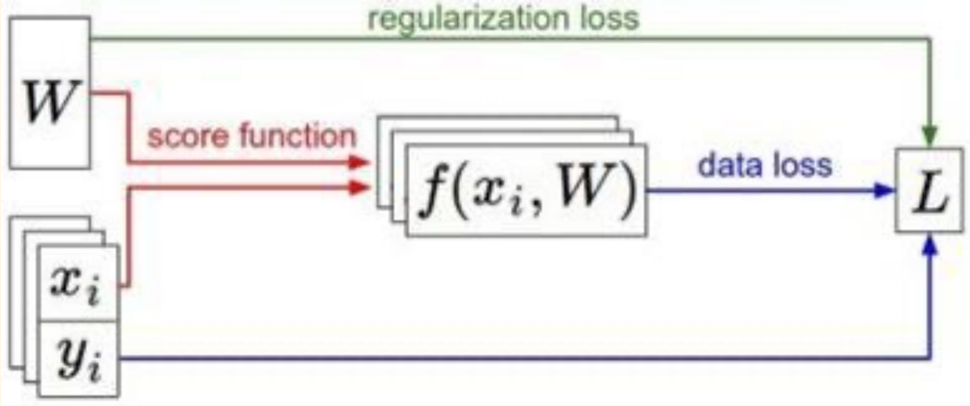
http://bolei-zhang.github.io/course/opt.html

上一节课

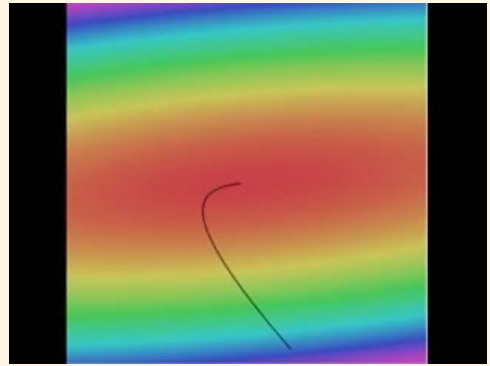
- 数据集 (X, y)
- 模型 $f(x; W)$
- 损失函数

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$ Softmax

$$L = \frac{1}{N}\sum_{i=1}^{N} L_i + R(W)$$ Full loss



梯度下降法



```
# Vanilla Gradient Descent

while True:
  weights_grad = evaluate_gradient(loss_fun, data, weights)
  weights += - step_size * weights_grad # perform parameter update
```

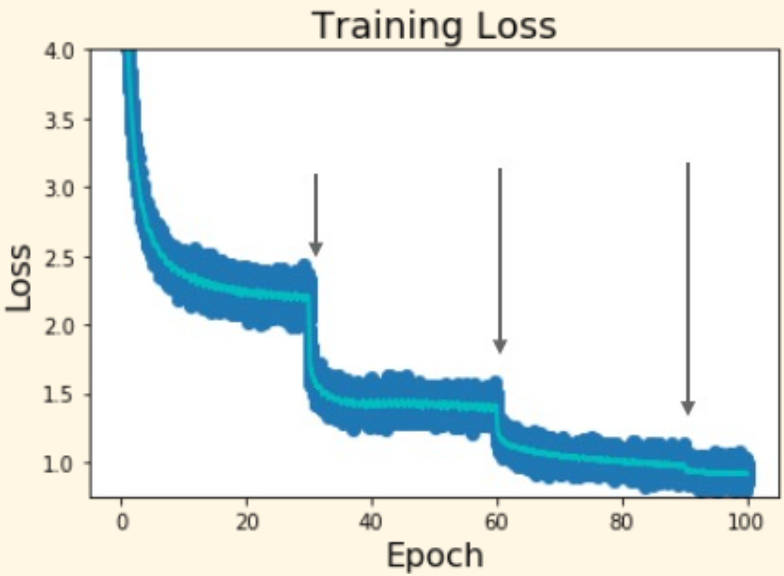
Stochastic Gradient Descent (SGD)

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

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

```
# Vanilla Minibatch Gradient Descent

while True:
  data_batch = sample_training_data(data, 256) # sample 256 examples
  weights_grad = evaluate_gradient(loss_fun, data_batch, weights)
  weights += - step_size * weights_grad # perform parameter update
```

学习率（步长）



步骤衰减：在几个固定节点降低学习率。

例如，对于 ResNets，在第 30、60 和 90 个 epoch 后将学习率乘以 0.1。

DALL-E 2

"Teddy bears working on new AI research on the moon in the 1980s."

"Rabbits attending a college seminar on human anatomy."

"A wise cat meditating in the Himalayas searching for enlightenment."

Image source: Sam Altman, https://openai.com/dall-e-2/, https://twitter.com/sama/status/1511724264629678084

DALL-E 3

GPT-4

User What is unusual about this image?

Source: Barnorama

GPT-4 The unusual thing about this image is that a man is ironing clothes on an ironing board attached to the roof of a moving taxi.

User Can you explain this meme?

GPT-4 This meme is a joke that combines two unrelated things: pictures of the earth from space and chicken nuggets.
The text of the meme suggests that the image below is a beautiful picture of the earth from space. However, the image is actually of chicken nuggets arranged to vaguely resemble a map of the world.
The humor in this meme comes from the unexpected juxtaposition of the text and the image. The text sets up an expectation of a majestic image of the earth, but the image is actually something mundane and silly.

SAM

Sora

A Shiba Inu dog wearing a beret and black turtleneck.

put the video in space with a rainbow road

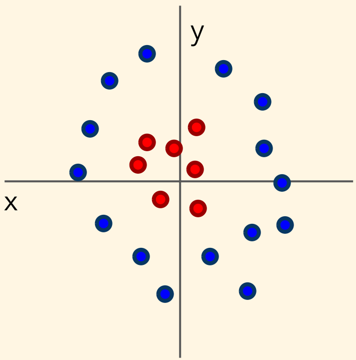
change the video setting to be different than a mountain? perhaps joshua tree

神经网络

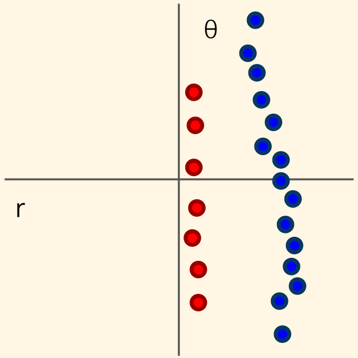
- 线性模型 $f(x;W) = Wx$
- 两层的神经网络

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

引入非线性层



$$f(x,y) = (r(x,y),\theta(x,y))$$



神经网络

- 线性模型 $f(x;W) = Wx$
- 两层的神经网络

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

- 三层

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

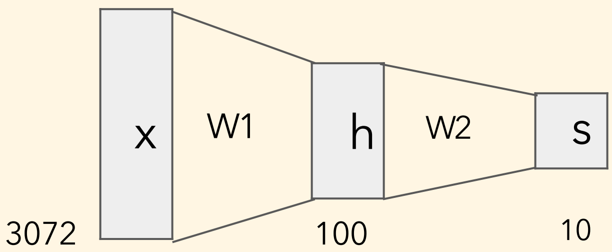
全连接网络（fully connected network）

多层感知机（multi layer perceptrons）

神经网络

- 线性模型 $f(x;W) = Wx$
- 两层的神经网络

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



神经网络

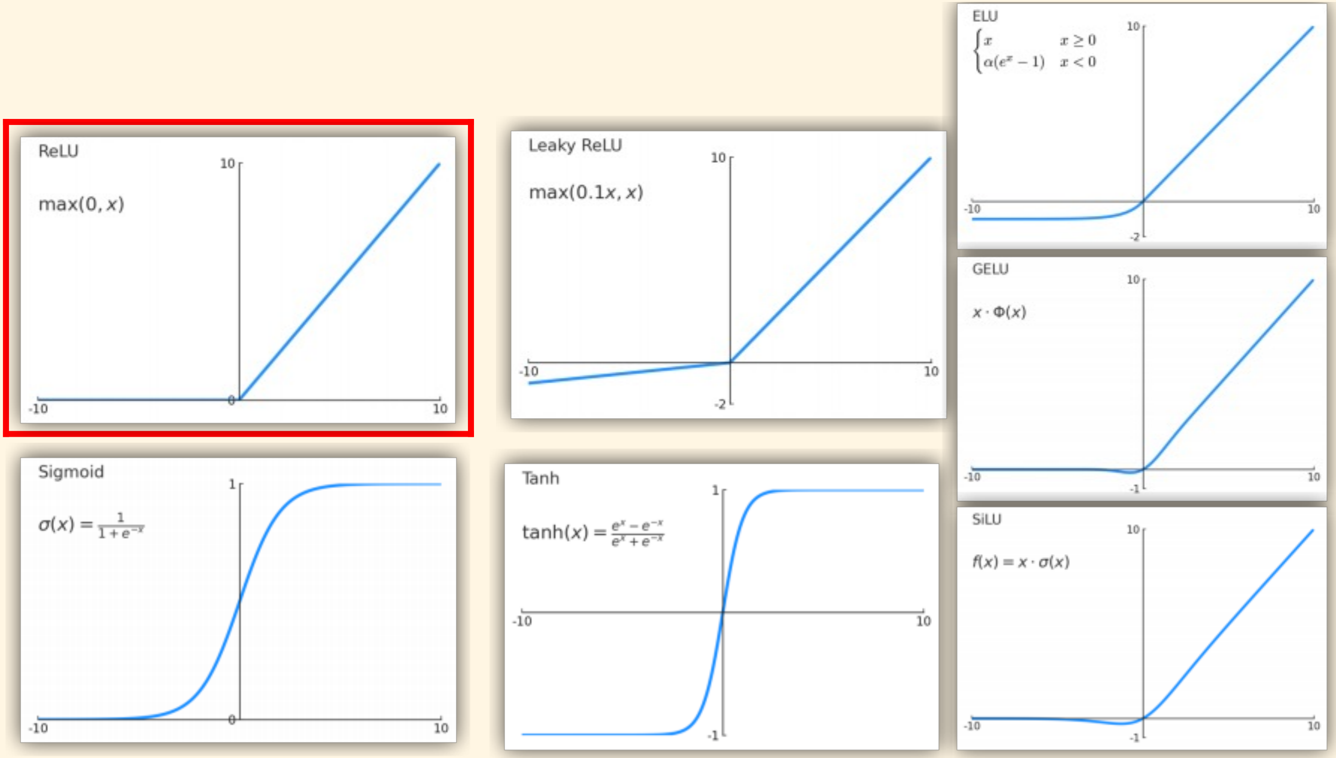
- 线性模型 $f(x;W) = Wx$
- 两层的神经网络

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

- 激活层的作用： $\max(0, W_1 x)$
    - 如果没有激活层

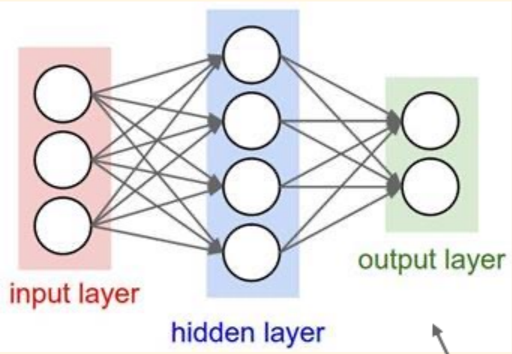
$$f = W_2 W_1 x \qquad W_3 = W_2 W_1 \in \mathbb{R}^{C \times H}, f = W_3 x$$

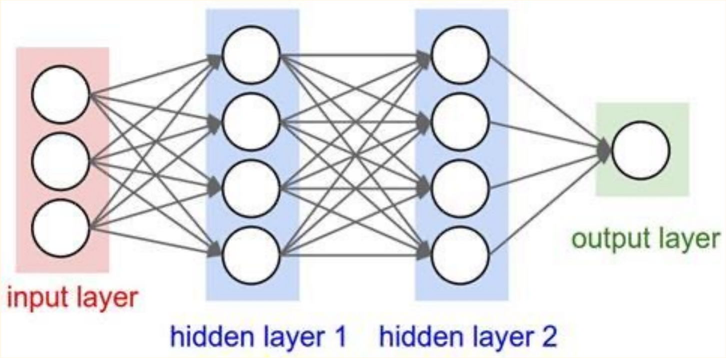
激活函数



ReLU
max(0, x)
Leaky ReLU
max(0.1x, x)
ELU
$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$
GELU
$x \cdot \Phi(x)$
Sigmoid
$\sigma(x) = \frac{1}{1+e^{-x}}$
Tanh
$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
SiLU
$f(x) = x \cdot \sigma(x)$

神经网络结构



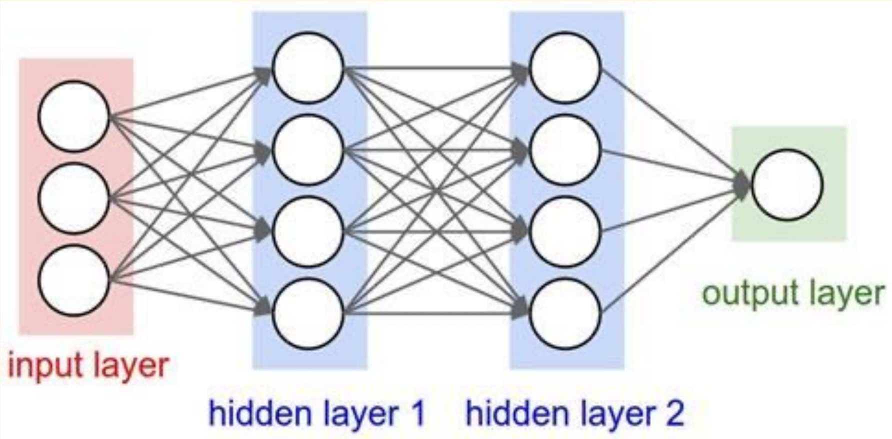


“2-layer Neural Net”, or
“1-hidden-layer Neural Net”

全连接层

“3-layer Neural Net”, or
“2-hidden-layer Neural Net”

feed-forward

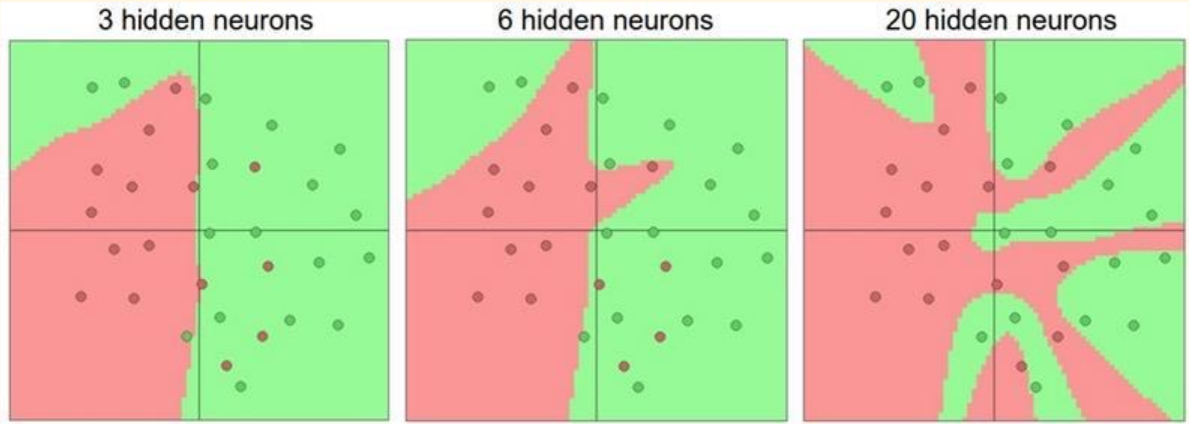



```
# forward-pass of a 3-layer neural network:
f = lambda x: 1.0/(1.0 + np.exp(-x)) # activation function (use sigmoid)
x = np.random.randn(3, 1) # random input vector of three numbers (3x1)
h1 = f(np.dot(W1, x) + b1) # calculate first hidden layer activations (4x1)
h2 = f(np.dot(W2, h1) + b2) # calculate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

训练一个两层的神经网络

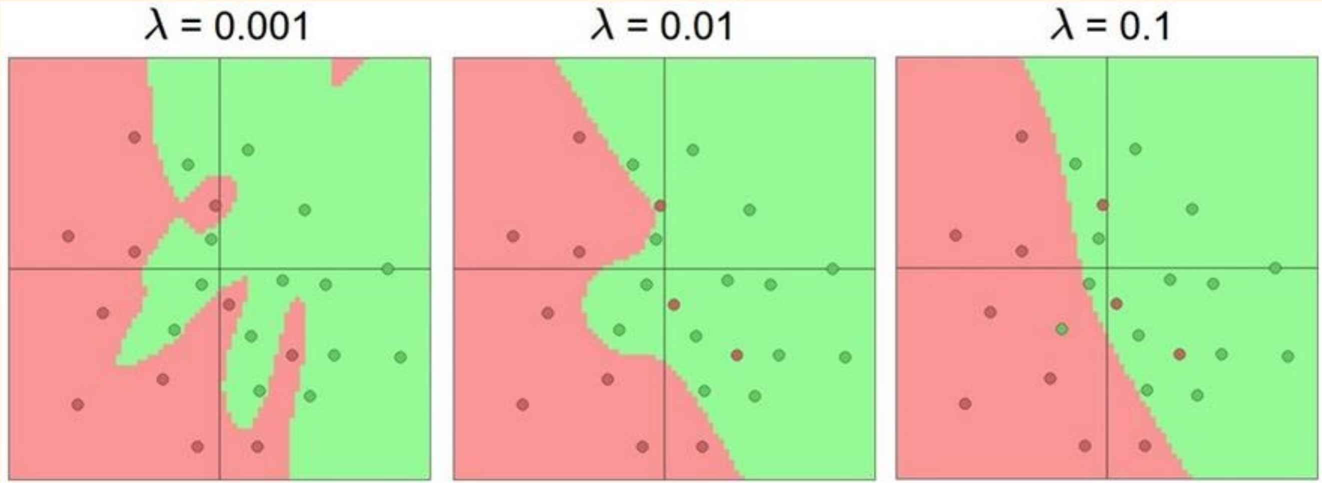
```
import numpy as np
from numpy.random import randn

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
```

神经网络的层数与神经元的个数



more neurons = more capacity

使用正则化项，而不是神经网络大小



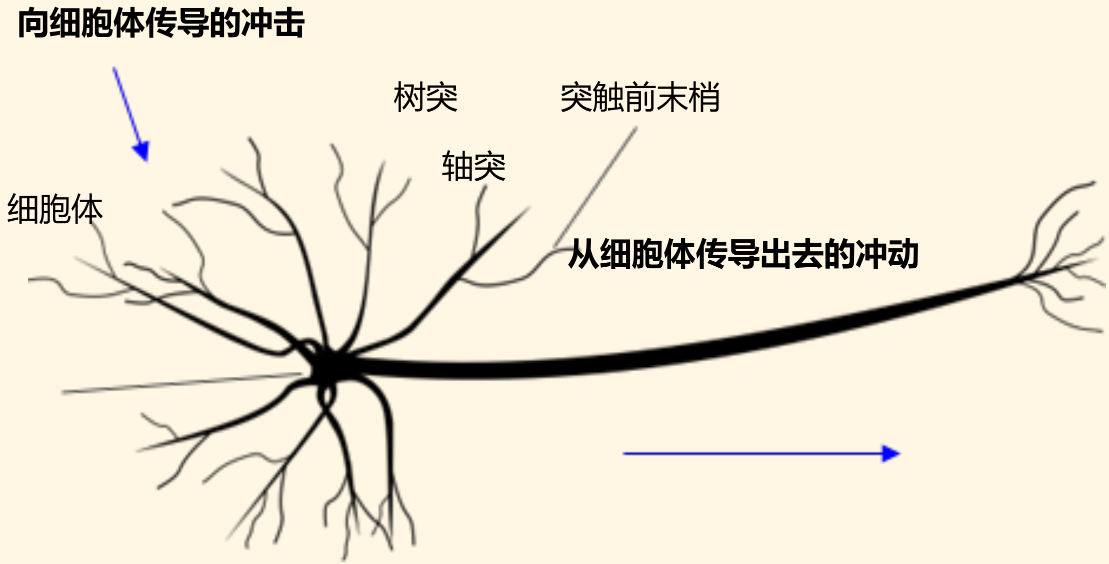
(Web demo with ConvNetJS: http://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html)

TensorFlow Play Ground: https://playground.tensorflow.org/

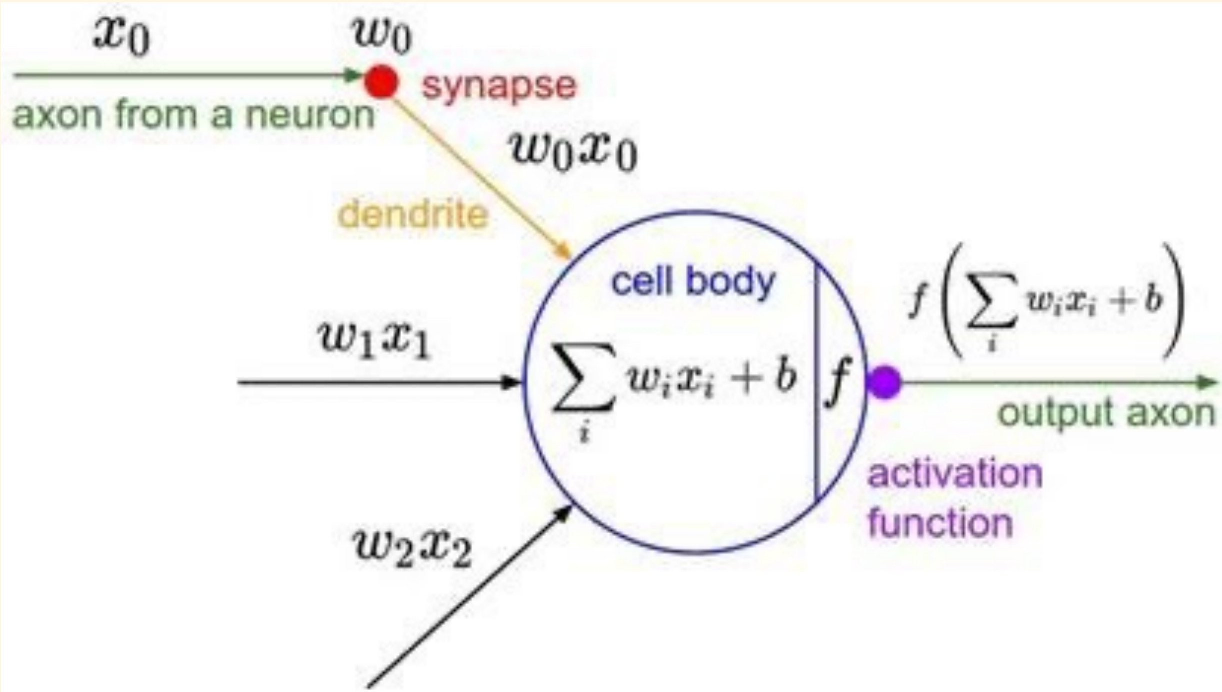
$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(f(x_i, W), y_i) + \lambda R(W)$$

神经元

神经元



神经网络神经元



x_0
w_0
synapse
axon from a neuron
$w_0 x_0$
dendrite
cell body
$f\left(\sum_i w_i x_i + b\right)$
$w_1 x_1$
$\sum_i w_i x_i + b$
f
output axon
activation function
$w_2 x_2$

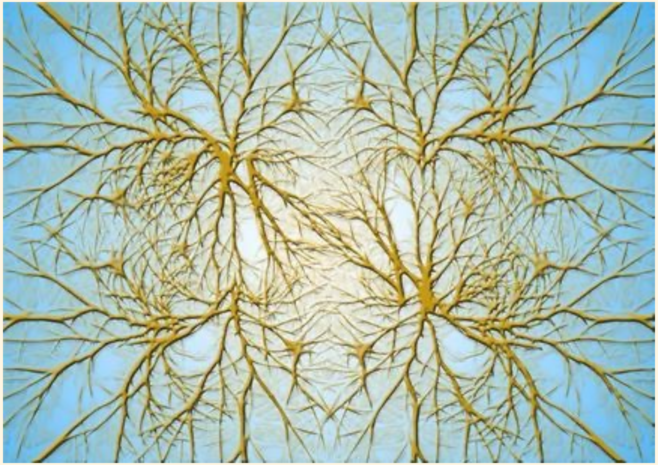
对比

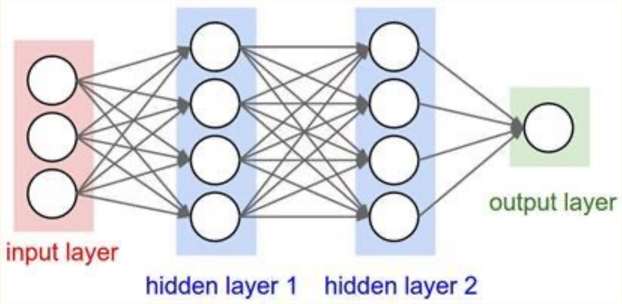
生物神经元：

复杂连接模型

神经网络神经元：

比较规则的连接模式

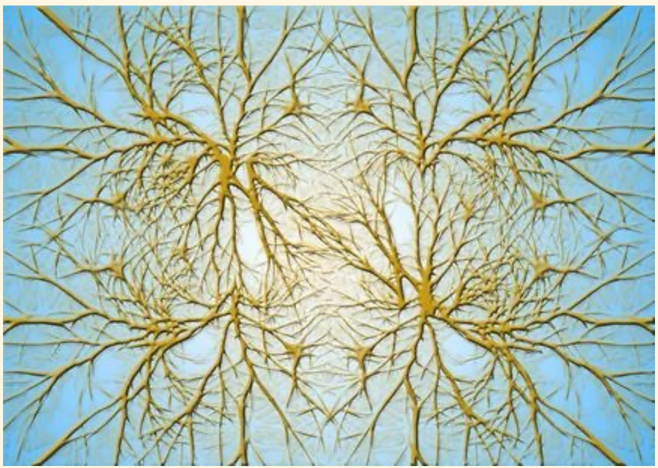




对比

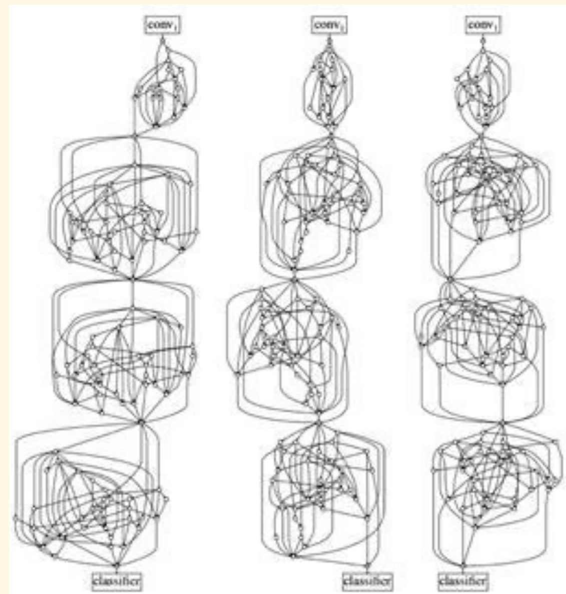
生物神经元：

复杂连接模型



通过激活层，人工神经网络

也会非常复杂



区别

生物神经元的特点:

- 存在多种不同类型
- 树突能够执行复杂的非线性计算
- 突触并非单一权重，而是复杂的非线性动力学系统

损失函数

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

非线性损失函数，Hinge Loss

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

$$R(W) = \sum_k W_k^2$$

正则化项

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + \lambda R(W_1) + \lambda R(W_2)$$

总的损失：Hinge Loss+正则化

如何计算梯度

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

非线性损失函数，Hinge Loss

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

$$R(W) = \sum_k W_k^2$$

正则化项

$$L = \frac{1}{N} \sum_{i=1}^{N} L_i + \lambda R(W_1) + \lambda R(W_2)$$

总的损失：Hinge Loss+正则化

核心是计算损失函数关于权重的梯度$\frac{\partial L}{\partial W_1}$, $\frac{\partial L}{\partial W_2}$

如果直接推导梯度

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

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

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

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

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

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

极其繁琐 —— 涉及大量矩阵微积分运算，需要耗费大量纸张。

想更改损失函数该怎么办？例如用 softmax 替代合页损失函数？必须从头重新推导所有内容！

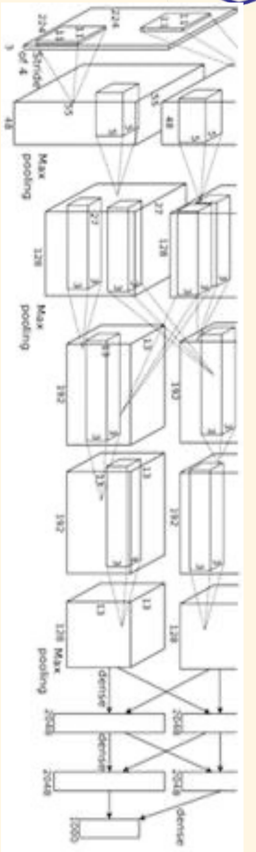
对于极为复杂的模型而言完全不可行！

卷积神经网络（AlexNet）

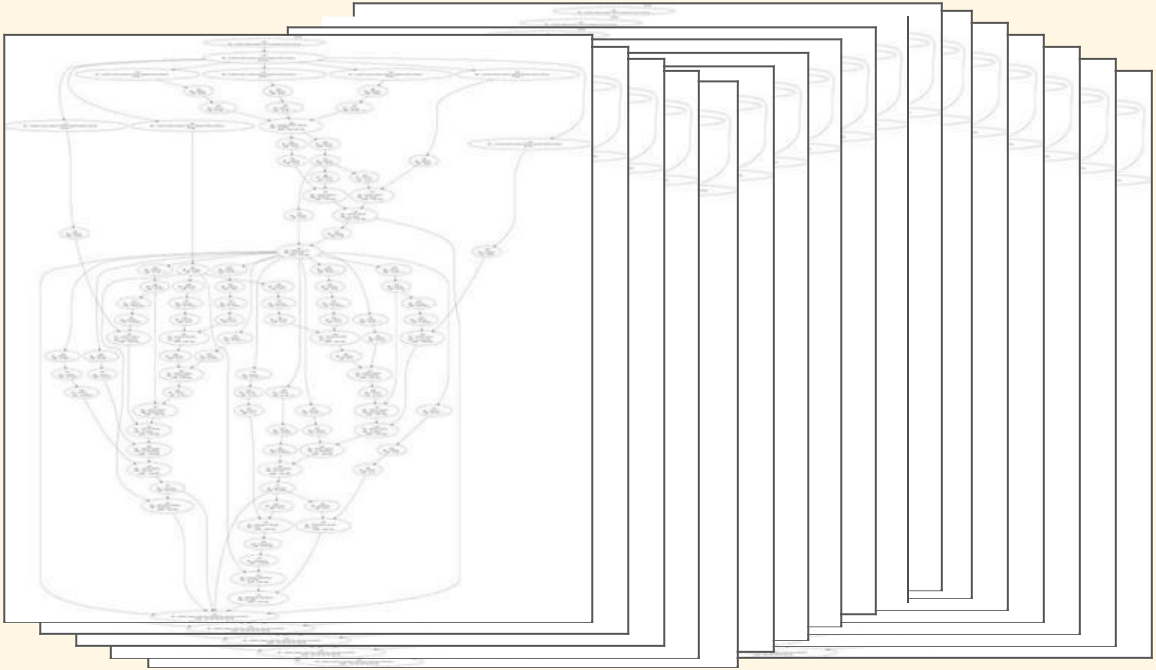
input image

weights

loss

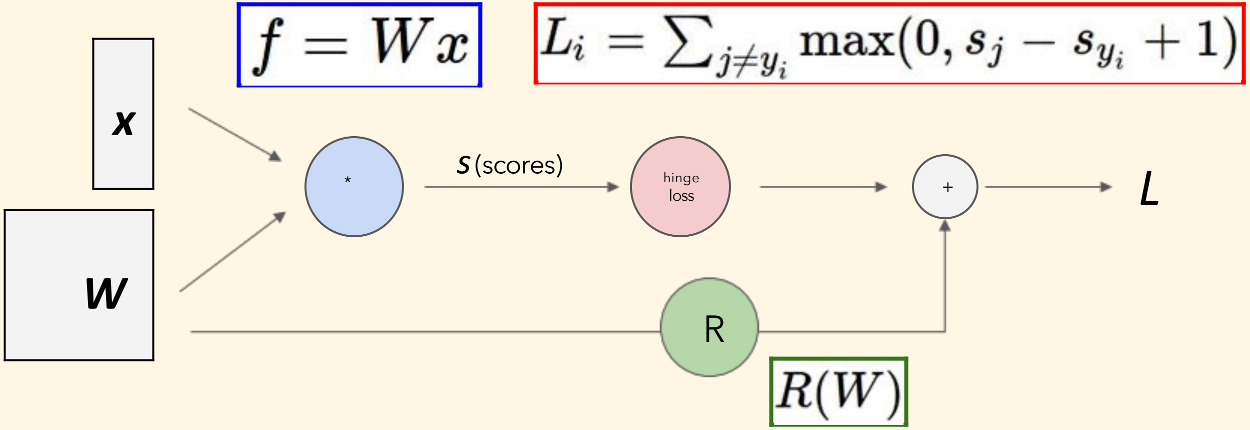


Neural Turing Machine



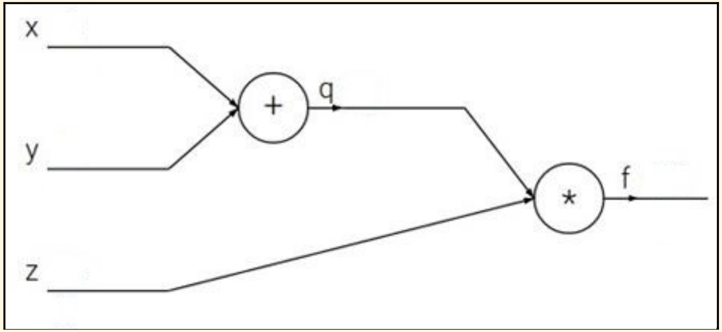
Backpropagation

更好的方法: 计算图+ Backpropagation



Backpropagation

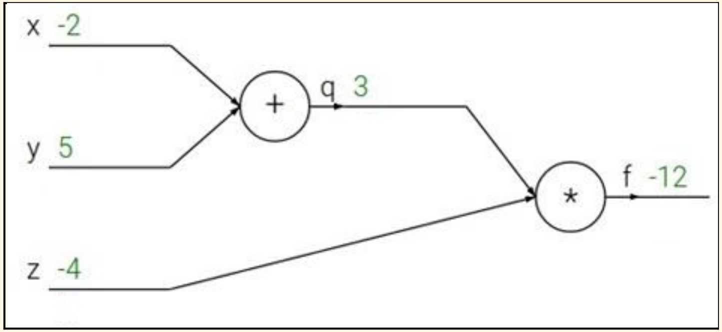
$$f(x, y, z) = (x + y)z$$



Backpropagation

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

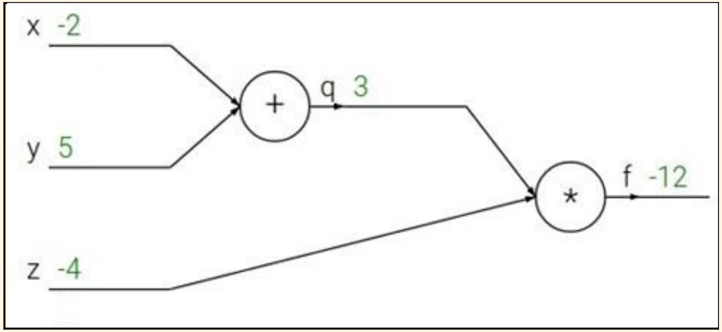
e.g. x = -2, y = 5, z = -4



Backpropagation

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

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



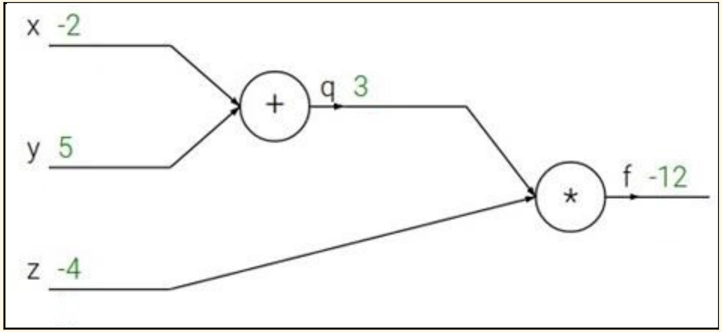
$$\boxed{q = x + y}$$

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

Backpropagation

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

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



$$\boxed{q = x + y}$$

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

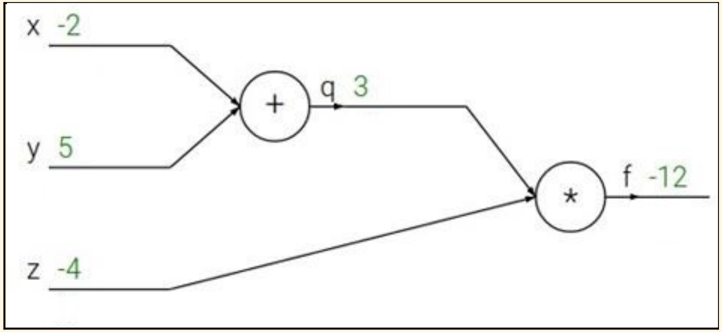
$$\boxed{f = qz}$$

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

Backpropagation

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

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



$$\boxed{q = x + y}$$

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

$$\boxed{f = qz}$$

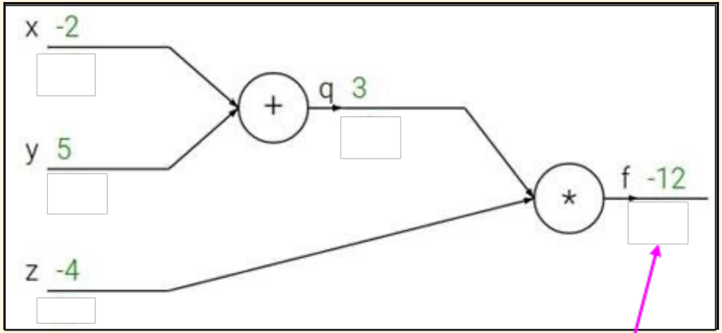
$$\frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

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

Backpropagation

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

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



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

$$q = x + y$$

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

$$f = qz$$

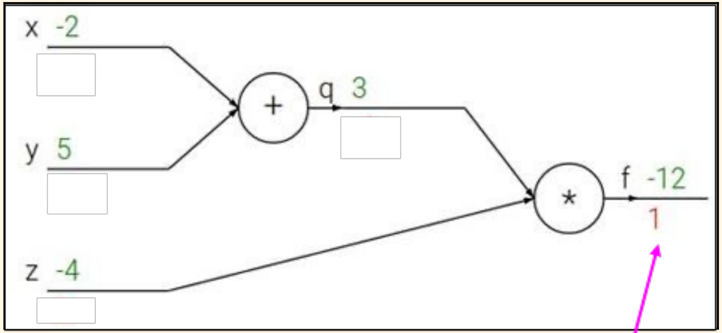
$$\frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

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

Backpropagation

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

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



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

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

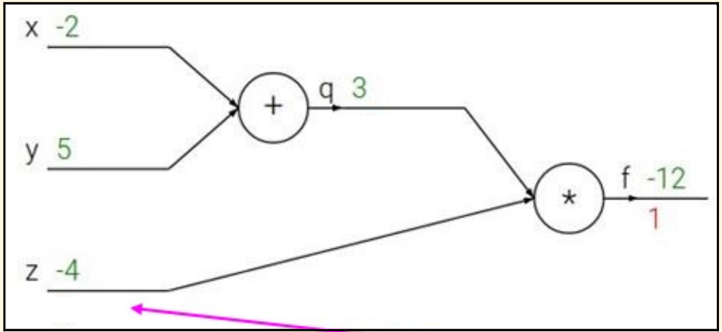
$$\boxed{f = qz} \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

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

Backpropagation

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

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



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

$$q = x + y$$

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

$$f = qz$$

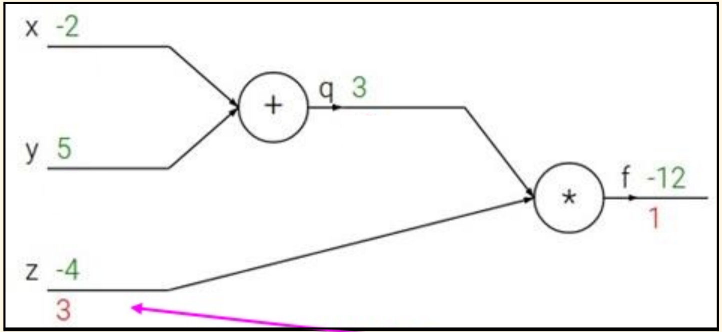
$$\frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

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

Backpropagation

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

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



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

$$\boxed{q = x + y}$$

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

$$\boxed{f = qz}$$

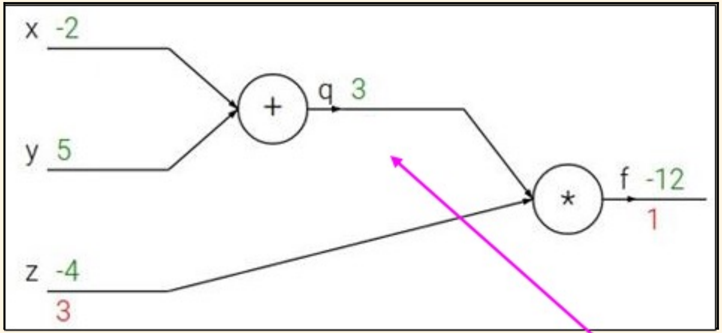
$$\frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

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

Backpropagation

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

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



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

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

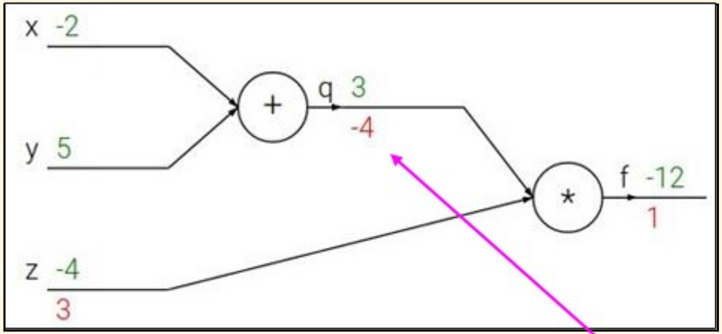
$$\boxed{f = qz}$$ $$\frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

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

Backpropagation

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

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



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

$$\boxed{q = x + y}$$

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

$$\boxed{f = qz}$$

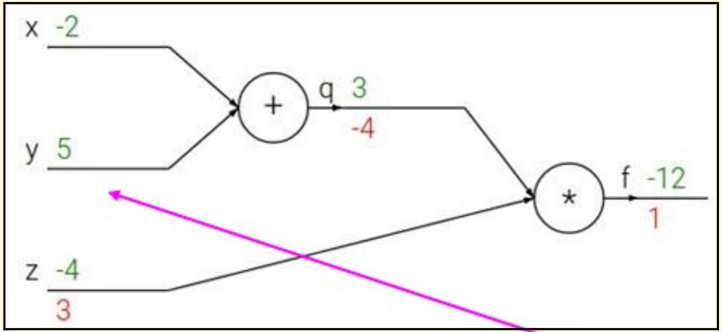
$$\frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

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

Backpropagation

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

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



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

$$\boxed{q = x + y}$$

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

$$\boxed{f = qz}$$

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

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

链式法则

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

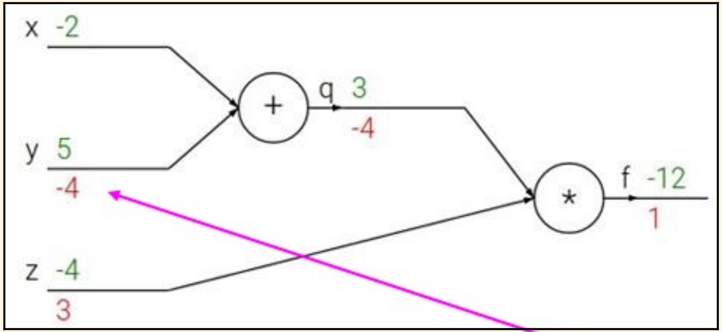
上游梯度

局部梯度

Backpropagation

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

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



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

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

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

链式法则

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

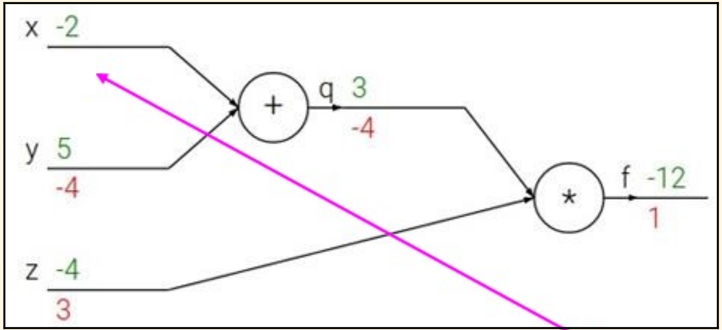
上游梯度　　局部梯度

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

Backpropagation

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

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



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

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

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

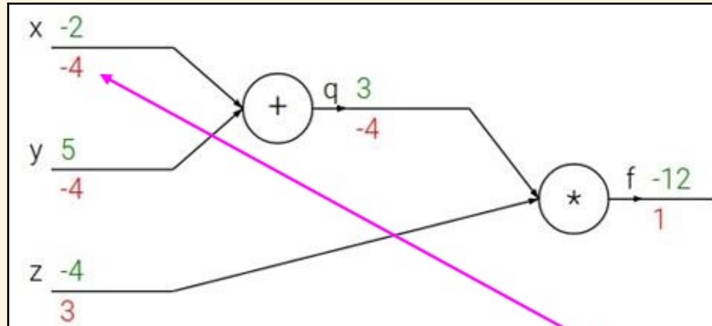
$$\boxed{\frac{\partial f}{\partial y} = \underbrace{\frac{\partial f}{\partial q}}_{\text{上游梯度}} \underbrace{\frac{\partial q}{\partial y}}_{\text{局部梯度}}}$$

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

Backpropagation

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

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



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

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

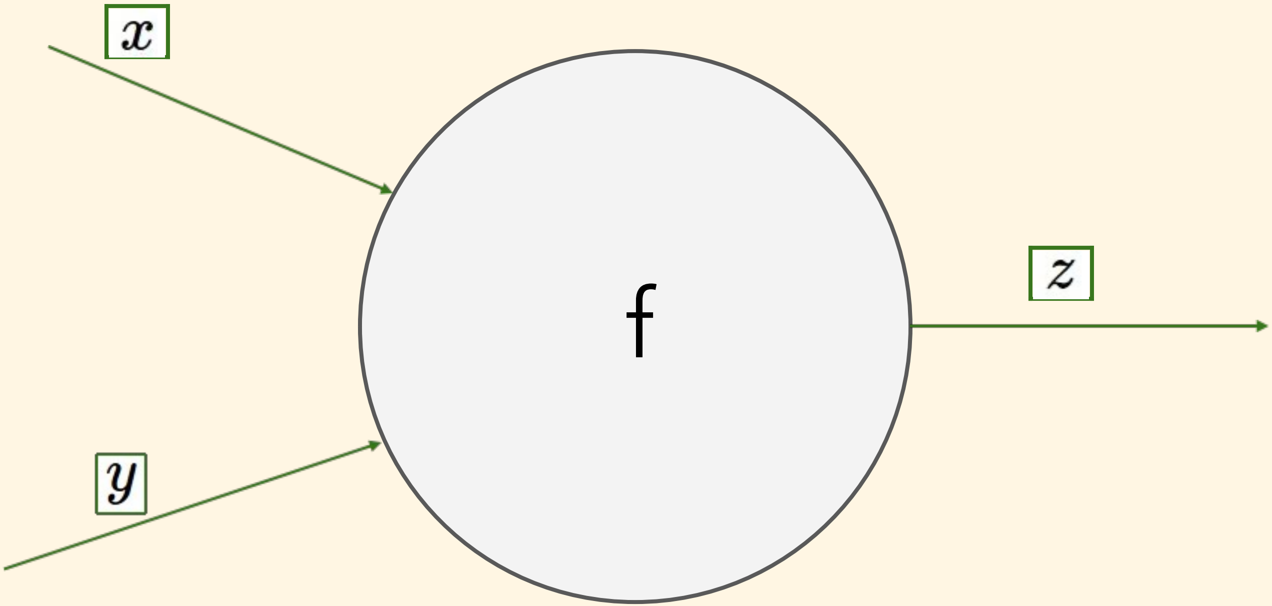
$$\boxed{f = qz} \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$$

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

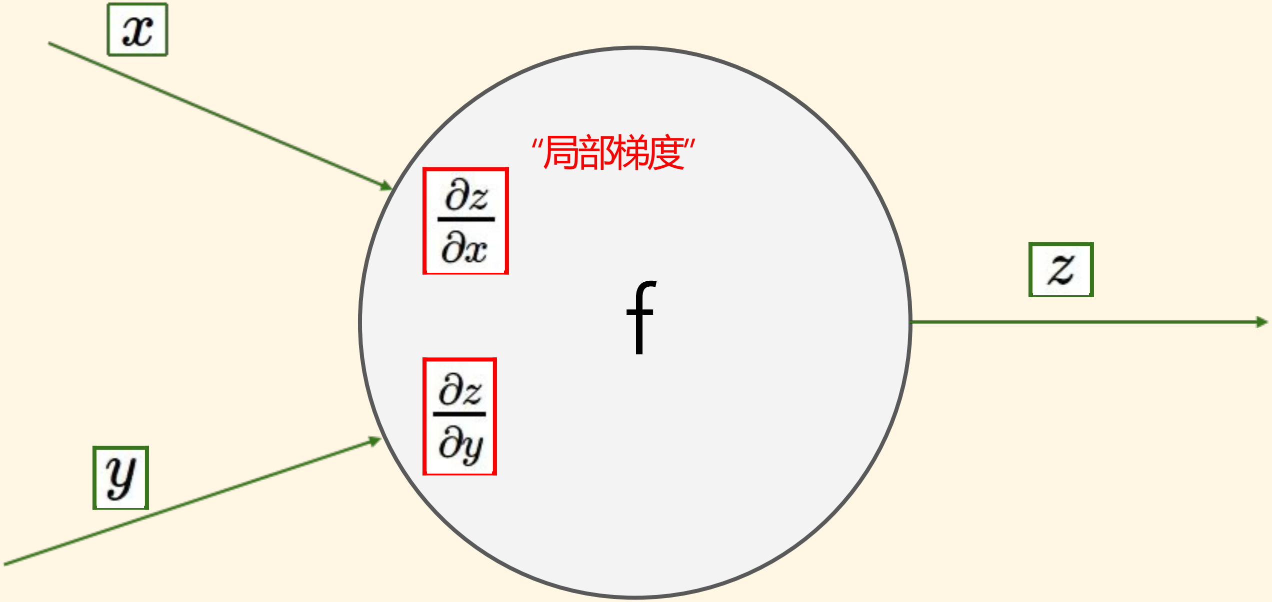
上游梯度 局部梯度

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

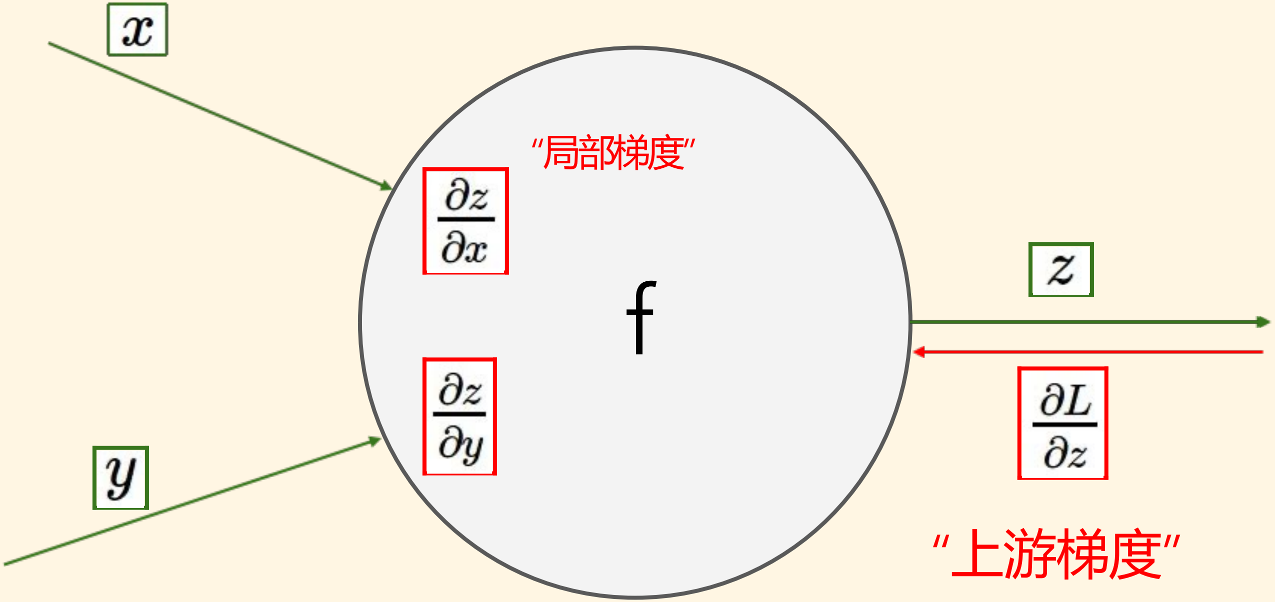
神经元



神经元

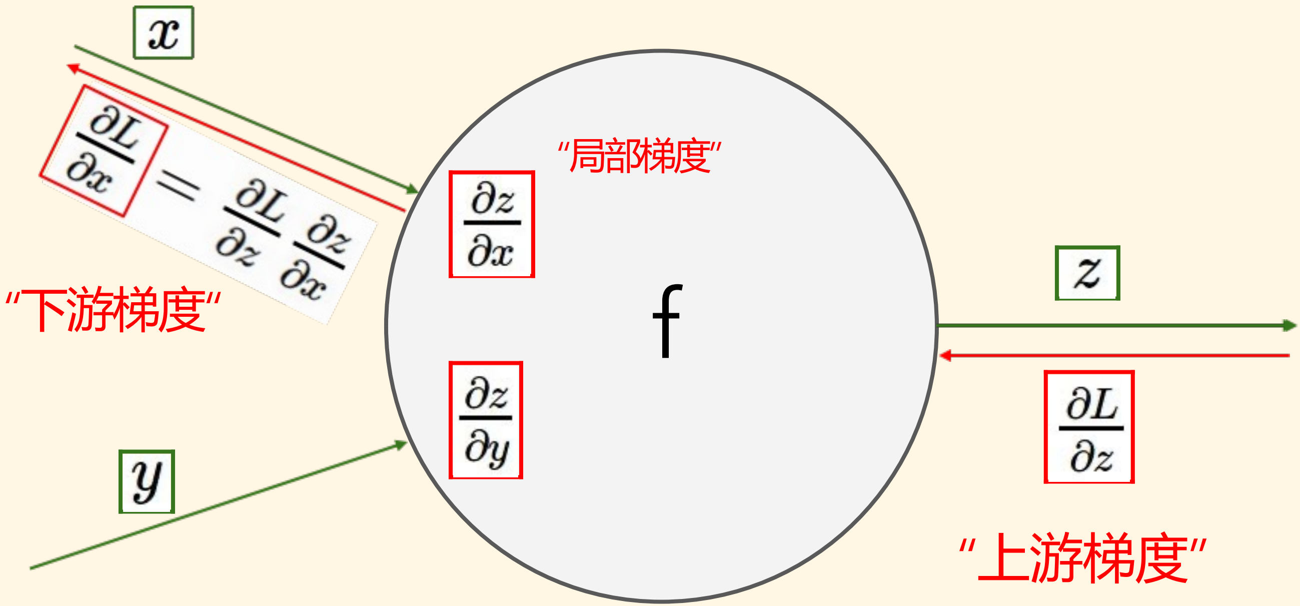


神经元



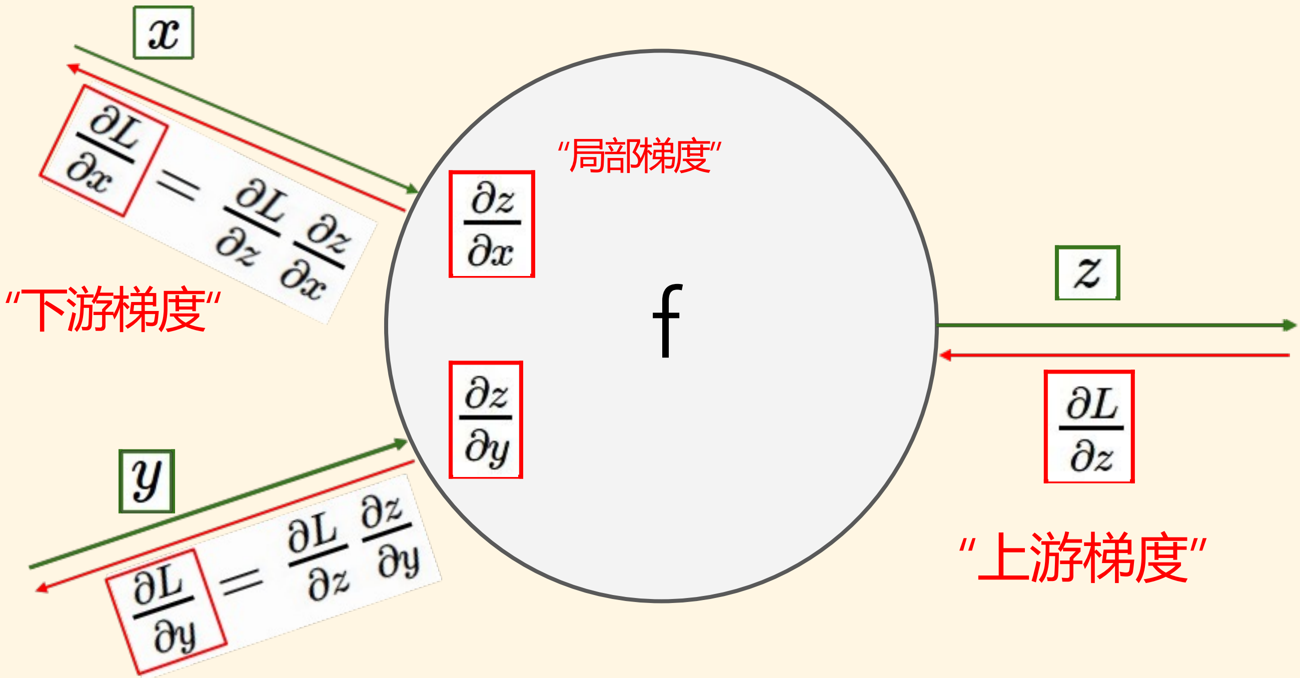
x
"局部梯度"
$\frac{\partial z}{\partial x}$
f
z
$\frac{\partial z}{\partial y}$
y
$\frac{\partial L}{\partial z}$
"上游梯度"

神经元



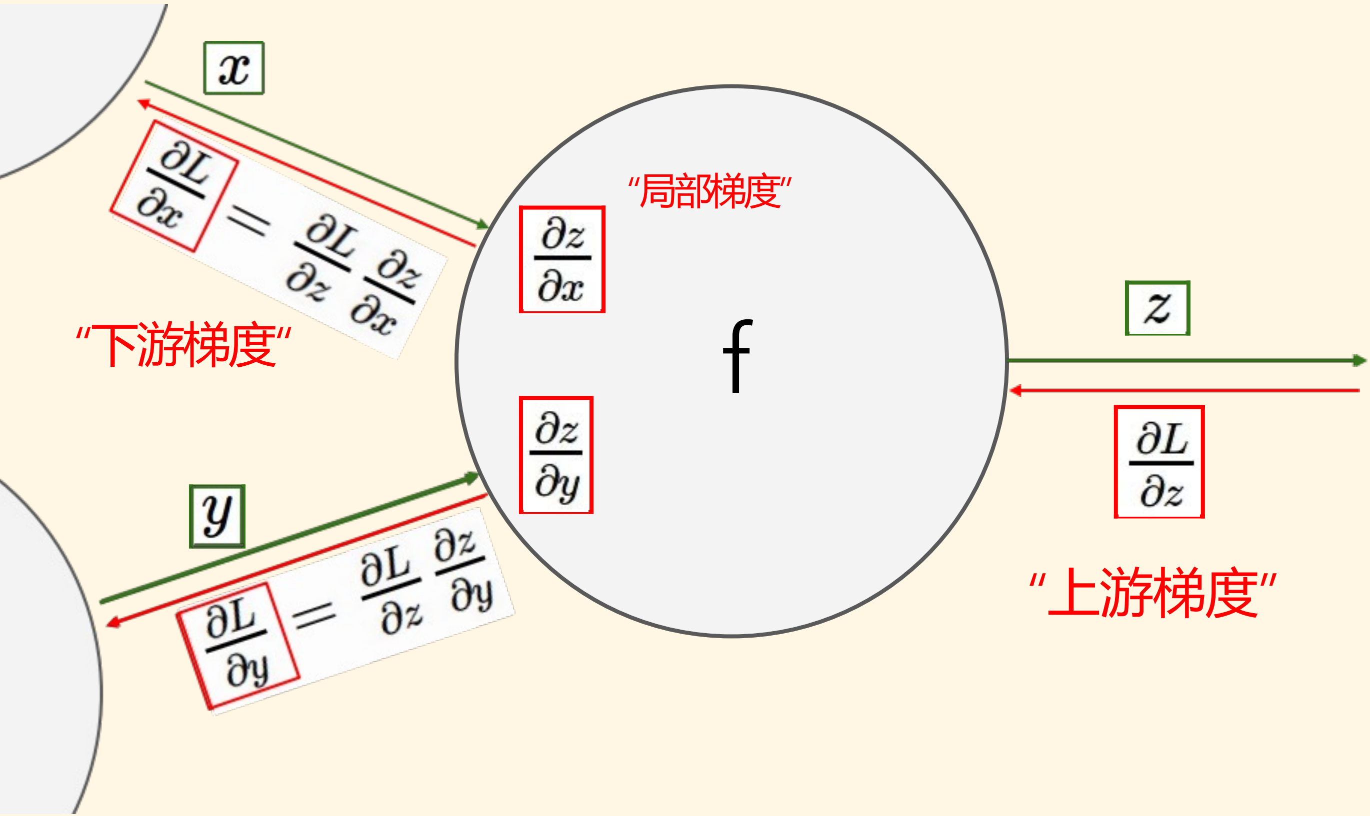
x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
"下游梯度"
"局部梯度"
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
f
z
$\frac{\partial L}{\partial z}$
"上游梯度"
y

神经元



x
$\boxed{\frac{\partial L}{\partial x}} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
"下游梯度"
y
$\boxed{\frac{\partial L}{\partial y}} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
"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"

神经元



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

谢谢！