

卷积神经网络

Convolutional neural network (CNN)

本节课内容

1. 图像分类

1.1 线性分类器

1.2 损失函数

1.3 梯度下降

1.4 反向传播

2. 卷积神经网络

2.1 卷积层

2.2 池化层

2.3 激活函数

2.4 全连接层

3. 经典CNN (AlexNet)

1. 图像分类

图像分类——计算机视觉的核心问题



假设给定一组离散标签: dog, car, cow,
automobile ...

→ **cat**

图片的语义鸿沟

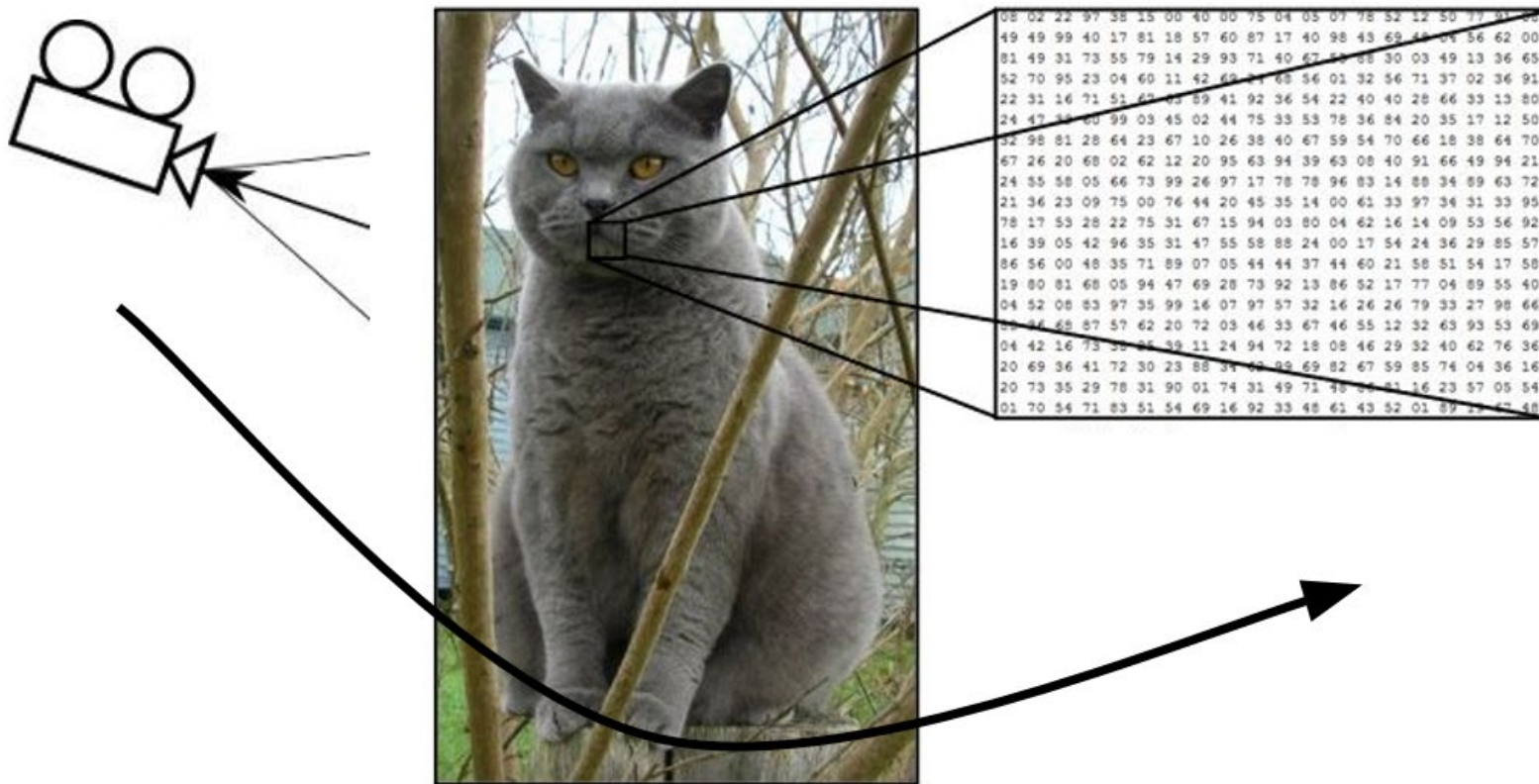
- 一个RGB图片可以被表示成3D的数字阵列
(3维张量tensor)
- 取值范围为[0,255]
- E.g. 200 x 100 x 3 (3表示三个通道)



08	02	22	97	38	15	00	40	00	75	04	05	07	78	52	12	50	77	87	28
49	49	99	40	17	81	18	57	60	87	17	40	98	43	69	48	84	56	62	00
81	49	31	73	55	79	14	29	93	71	40	67	53	88	30	03	49	13	36	65
52	70	95	23	04	60	11	42	69	21	68	56	01	32	56	71	37	02	36	91
22	31	16	71	51	67	83	59	41	92	36	54	22	40	40	28	66	33	13	80
24	47	38	60	99	03	45	02	44	75	33	53	78	36	84	20	35	17	12	50
32	98	81	28	64	23	67	10	26	38	40	67	59	54	70	66	18	38	64	70
67	26	20	68	02	62	12	20	95	63	94	39	63	08	40	91	66	49	94	21
24	55	58	05	66	73	99	26	97	17	78	78	96	83	14	88	34	89	63	72
21	36	23	09	75	00	76	44	20	45	35	14	00	61	33	97	34	31	33	95
78	17	53	28	22	75	31	67	15	94	03	80	04	62	16	14	09	53	56	92
16	39	05	42	96	35	31	47	55	58	88	24	00	17	54	24	36	29	85	57
86	56	00	48	35	71	89	07	05	44	44	37	44	60	21	58	51	54	17	58
19	80	81	68	05	94	47	69	28	73	92	13	86	52	17	77	04	89	55	40
04	52	08	83	97	35	99	16	07	97	57	32	16	26	26	79	33	27	98	66
69	86	68	87	57	62	20	72	03	46	33	67	46	55	12	32	63	93	53	69
04	42	16	73	55	85	39	11	24	94	72	18	08	46	29	32	40	62	76	36
20	69	36	41	72	30	23	88	34	62	89	69	82	67	59	85	74	04	36	16
20	73	35	29	78	31	90	01	74	31	49	71	48	88	61	16	23	57	05	54
01	70	34	71	83	51	54	69	16	92	33	48	61	43	52	01	89	19	67	48

What the computer sees

挑战——视角变化 (Viewpoint Variation)



挑战——光照条件 (Illumination)



挑战——形变 (Deformation)



猫的行为艺术

挑战——遮挡 (Occlusion)



You cannot see me!

挑战——背景干扰（Background clutter）



挑战——类间细分 (Intraclass variation)



上一节课回顾

- 数字图像处理
 - 二值图、灰度图、彩色图
 - 数字图像采样 (Image Sampling)
 - 数字图像操作 (Image Augmentation)
 - 数字图像滤波 (Image Filter)
- 图像识别的挑战
- 参考文献来源
 - 期刊: PAMI, IJCV, TIP, CVIU
 - 会议: CVPR, ICCV, ECCV, ICML, ICLR, AAAI, NeurIPS

Anaconda

- <https://www.anaconda.com/distribution/>
- Anaconda中包含Python、科学包及依赖库
- 学习使用Jupyter（是一个交互式笔记本，Kaggle常用工具）

1.1 线性图像分类器

airplane



automobile



bird



cat



deer



dog



frog



horse



ship



truck



Example dataset: **CIFAR-10**

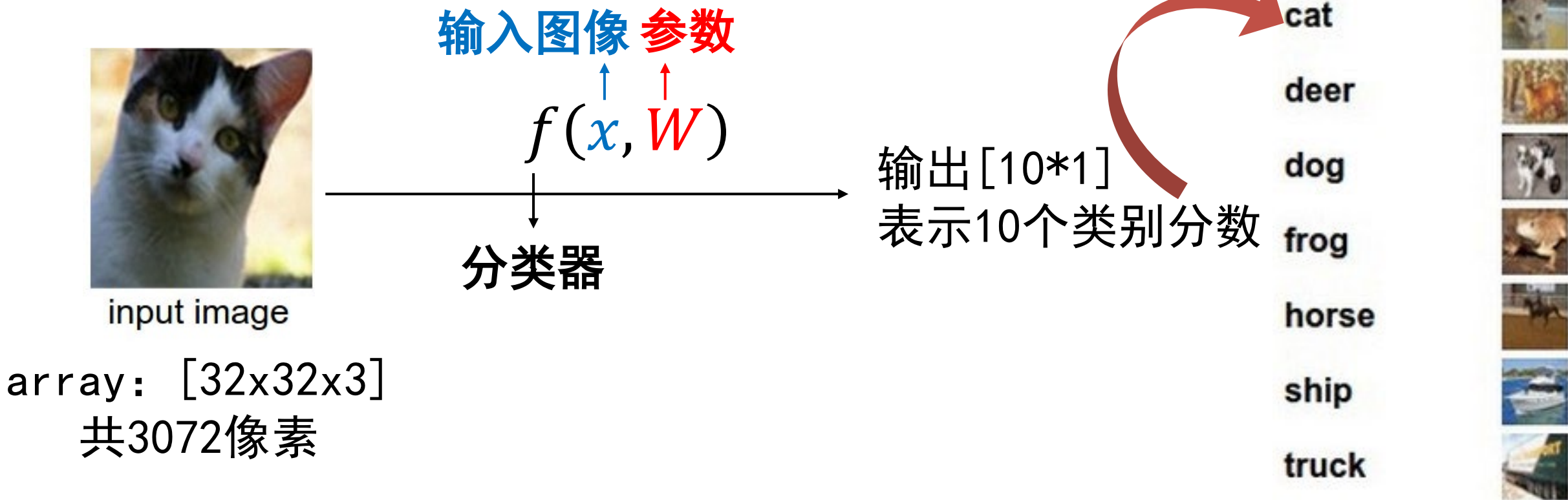
10 labels

50,000 training images

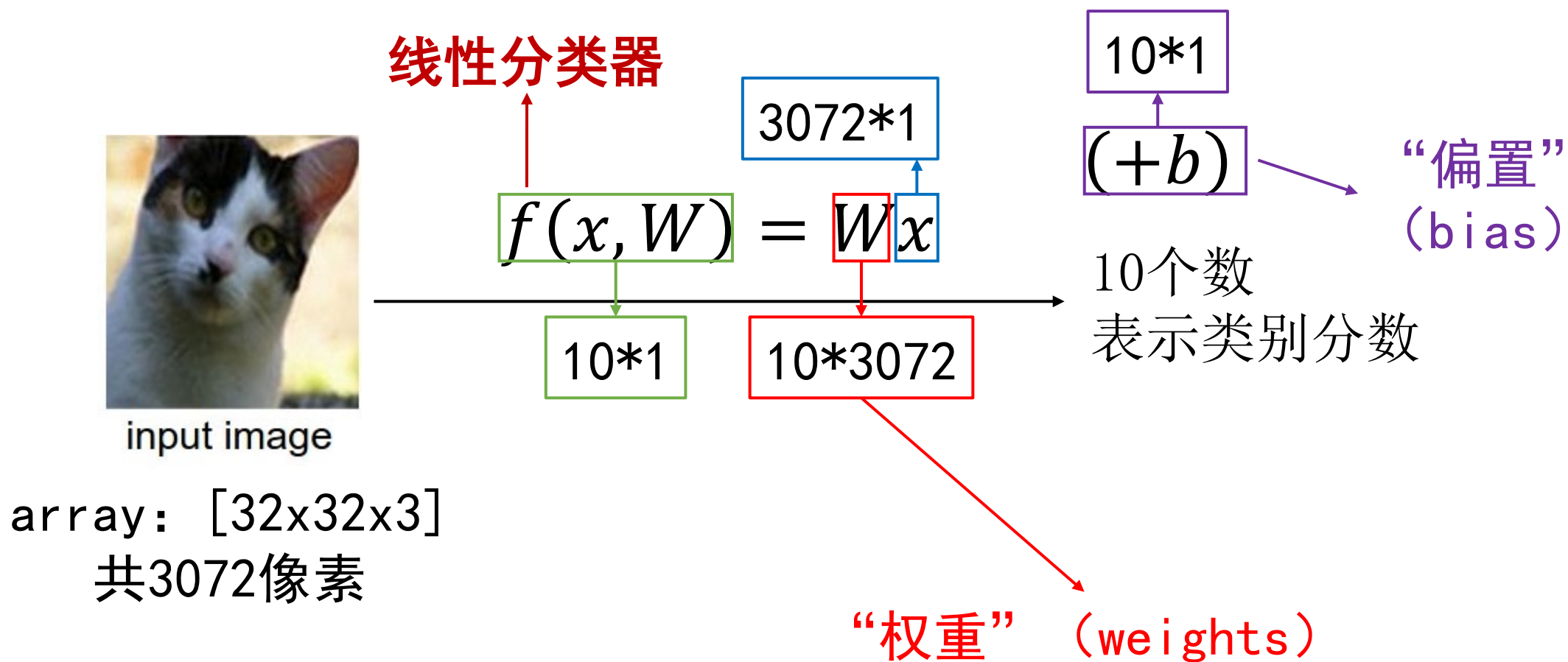
10,000 test images

each image is **32x32x3**.

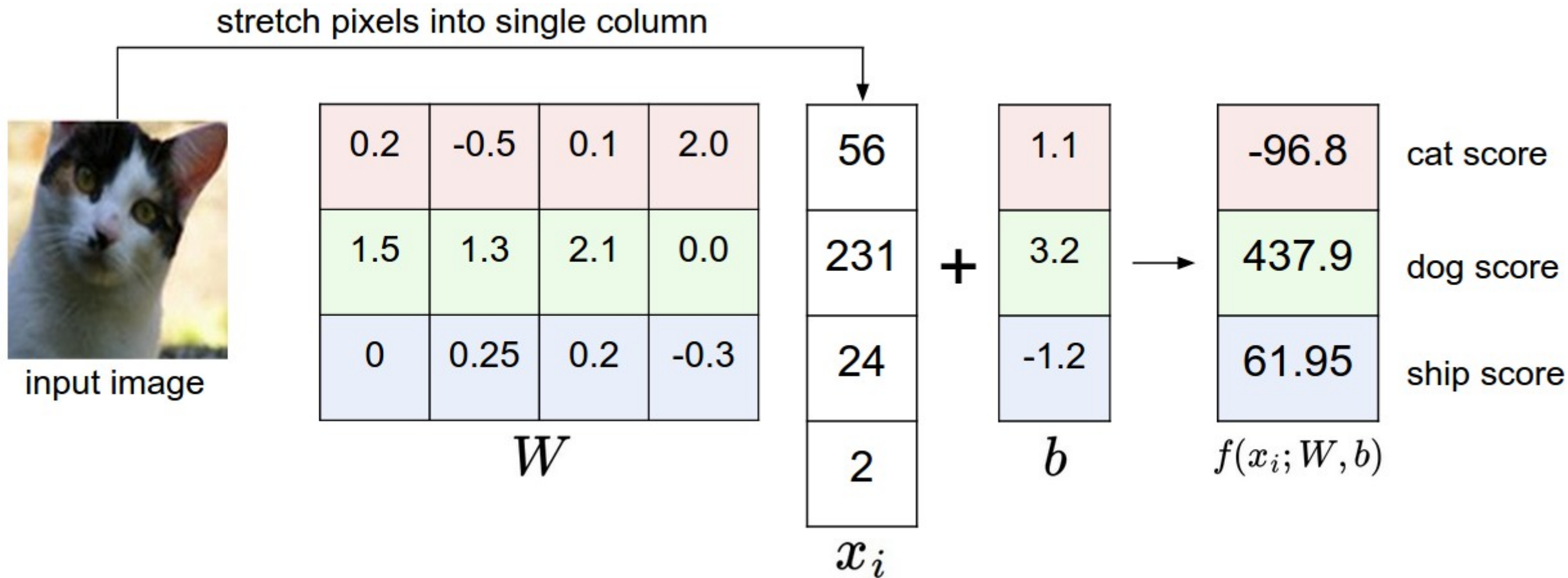
- 图像分类器



- 线性图像分类器

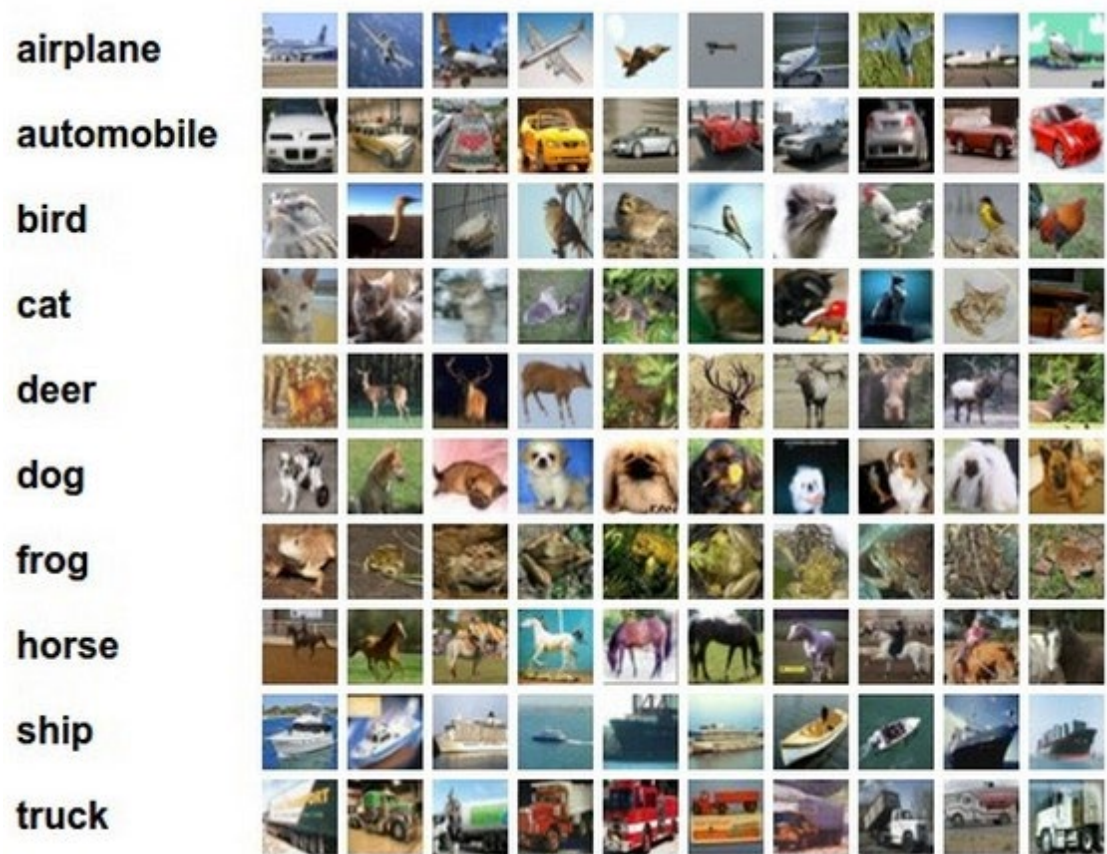


- 线性分类器



- 示例：输入图像4像素，有3类（**cat**/**dog**/**ship**）

• 线性分类器

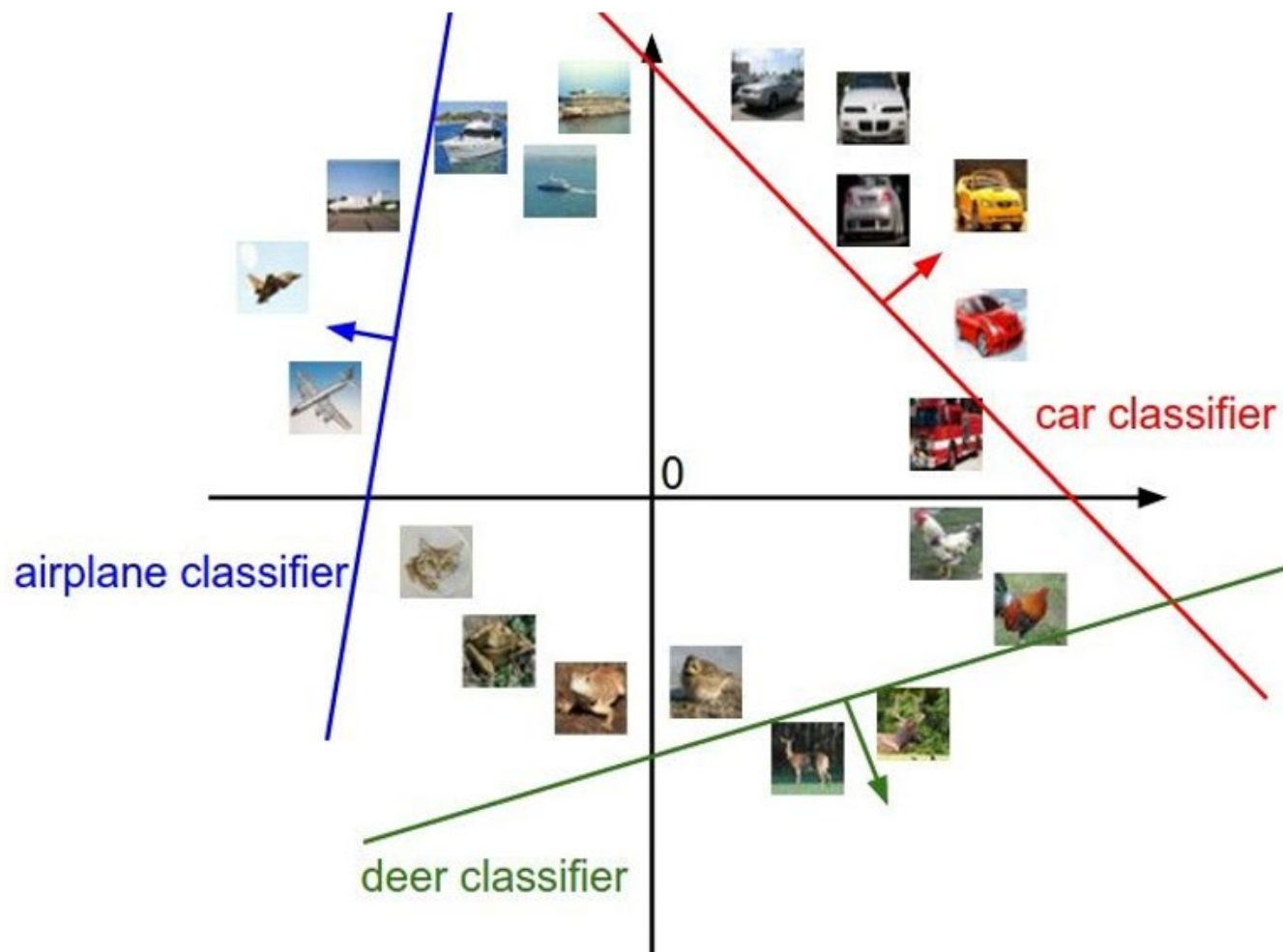


线性分类器

$$f(x_i; W, b) = Wx_i + b$$

线性分类器的权重示例
Trained on CIFAR-10

- 线性分类器



$$f(x_i, W, b) = Wx_i + b$$



[32x32x3]

- 线性分类器

score function:

$$f(x_i; W, b) = Wx_i + b$$



随机设定权重W
3张示例图的类别得分

airplane	-3.45	-0.51	3.42
automobile	-8.87	6.04	4.64
bird	0.09	5.31	2.65
cat	2.9	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	-4.34
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

输入为 *automobile* 输出 *automobile* 得分最高

输入为 *cat* 输出 *dog* 得分最高

输入为 *frog* 输出 *truck* 得分最高



分类结果:

✗



✓



✗



airplane	-3.45	-0.51	3.42
automobile	-8.87	6.04	4.64
bird	0.09	5.31	2.65
cat	2.9	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	-4.34
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

给定随机权重 W ，分类结果差

• 解决方案:

定义**损失函数**，通过优化权重 W 来最小化损失函数，以找到最好的分类器，达到最好的分类结果。

1.2 损失函数 (loss function)

定义 loss function: L_i \longrightarrow 衡量分类结果与真实值的不一致程度



1	cat	3.2	1.3	2.2
2	car	5.1	4.9	2.5
3	frog	-1.7	2.0	-3.1

Multiclass loss:

给定一个训练样本 (x_i, y_i)

x_i 为输入图像

y_i 为输出 label $\in [1, 2, 3]$

记 $s_i = f(x_i, W) = Wx_i$

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

示例：训练样本3张，有3类 (cat 1/car 2/frog 3)

损失函数计算

定义 loss function: L_i \longrightarrow 衡量分类结果与真实值的不一致程度



1	cat	3.2	1.3	2.2
2	car	5.1	4.9	2.5
3	frog	-1.7	2.0	-3.1

loss: **2.9**

Multiclass loss:

$$\begin{aligned} L_1 &= \sum_{j \neq y_1} \max(0, s_j - s_{y_1} + 1) \\ &= \max(0, 5.1 - 3.2 + 1) \\ &\quad + \max(0, -1.7 - 3.2 + 1) \\ &= \max(0, 2.9) + \max(0, -3.9) \\ &= 2.9 + 0 \\ &= 2.9 \end{aligned}$$

损失函数计算

定义 loss function: L_i

衡量分类结果与真实值的不一致程度



1	cat	3.2	1.3	2.2
2	car	5.1	4.9	2.5
3	frog	-1.7	2.0	-3.1

loss: 2.9

0

Multiclass loss:

$$\begin{aligned} L_2 &= \sum_{j \neq y_2} \max(0, s_j - s_{y_2} + 1) \\ &= \max(0, 1.3 - 4.9 + 1) \\ &\quad + \max(0, 2.0 - 4.9 + 1) \\ &= \max(0, -2.6) + \max(0, -1.9) \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

损失函数计算

定义 loss function: L_i \longrightarrow 衡量分类结果与真实值的不一致程度



1	cat	3.2	1.3	2.2
2	car	5.1	4.9	2.5
3	frog	-1.7	2.0	-3.1
loss:		2.9	0	12.9

Multiclass loss:

$$\begin{aligned} L_3 &= \sum_{j \neq y_3} \max(0, s_j - s_{y_3} + 1) \\ &= \max(0, 2.2 - (-3.1) + 1) \\ &\quad + \max(0, 2.5 - (-3.1) + 1) \\ &= \max(0, 6.3) + \max(0, 6.6) \\ &= 6.3 + 6.6 \\ &= 12.9 \end{aligned}$$

损失函数计算

定义 loss function: L_i \longrightarrow 衡量分类结果与真实值的不一致程度



1	cat	3.2	1.3	2.2
---	-----	------------	-----	-----

2	car	5.1	4.9	2.5
---	-----	-----	------------	-----

3	frog	-1.7	2.0	-3.1
---	------	------	-----	-------------

loss:	2.9	0	12.9
-------	-----	---	------

Multiclass loss:

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

总的训练损失:

$$L = \frac{1}{N} \sum_{i=1}^N L_i$$

$$= (2.9 + 0 + 12.9) / 3$$
$$= 5.3$$

分类结果:

×



✓



×



airplane	-3.45	-0.51	3.42
automobile	-8.87	6.04	4.64
bird	0.09	5.31	2.65
cat	2.9	-4.22	5.1
deer	4.48	-4.19	2.64
dog	8.02	3.58	5.55
frog	3.78	4.49	-4.34
horse	1.06	-4.37	-1.5
ship	-0.36	-2.09	-4.79
truck	-0.72	-2.93	6.14

给定随机权重 W ，分类结果差

• 解决方案:

定义**损失函数**，通过**优化权重 W** 来最小化损失函数，以找到最好的分类器，达到最好的分类结果。

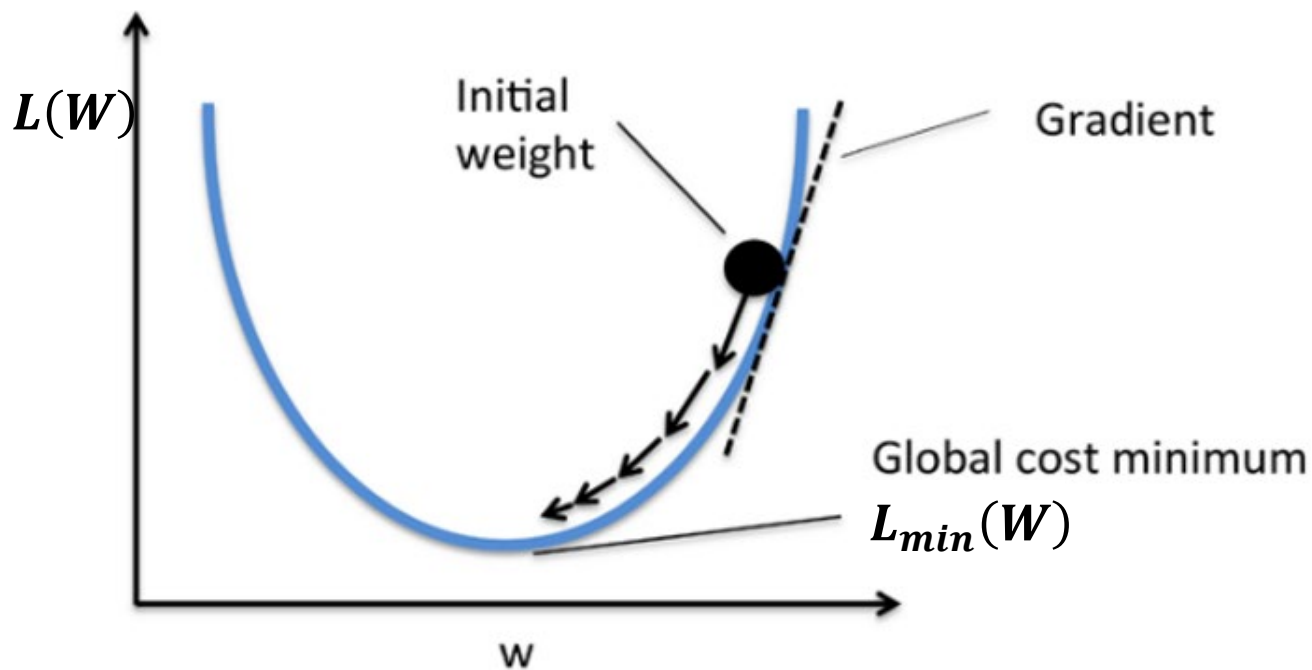
梯度下降

1.3 梯度下降

• 问题

如何优化调整参数 W, b ，以达到损失函数最小化的目的？

解决方案——**梯度下降**用于更新权重 W ，以及偏置 b 。



1. 初始化参数，计算损失 L ；
2. 梯度下降法每次迭代：

??

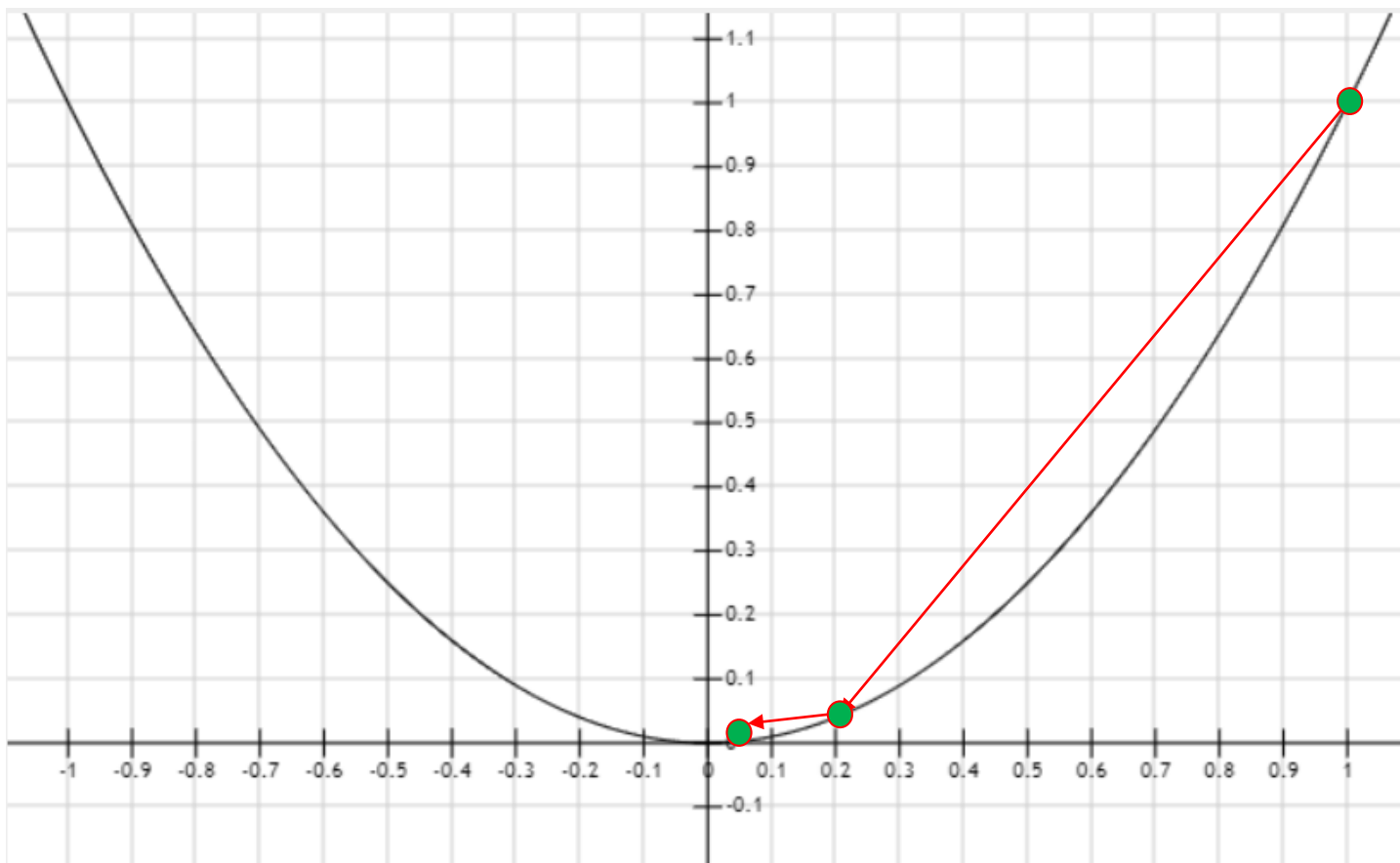
计算梯度 $\frac{\partial L(W,b)}{\partial W}, \frac{\partial L(W,b)}{\partial b}$;

更新参数：

$$W := W - \alpha \frac{\partial L(W,b)}{\partial W} \quad b := b - \alpha \frac{\partial L(W,b)}{\partial b}$$

3. 重复步骤2，直至 $L(\min)$.

梯度下降法的细节说明



$$L(w) = w^2, w_{(0)} = 1, \alpha = 0.4$$

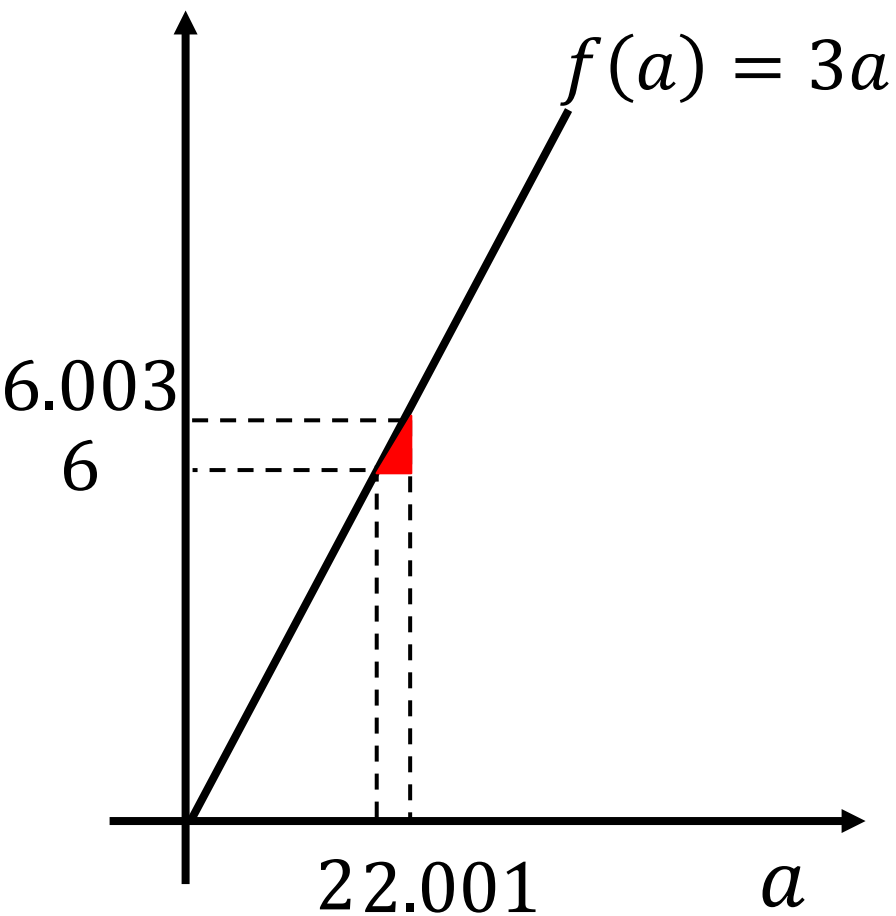
$$w := w - \alpha * \frac{dL(w)}{dw}$$

$$\begin{aligned} w_{(1)} &= w_{(0)} - \alpha * \frac{dL(w)}{dw} \\ &= 1 - 0.4 * 2 * 1 = 0.2 \end{aligned}$$

$$\begin{aligned} w_{(2)} &= w^1 - \alpha * \frac{dL(w)}{dw} \\ &= 0.2 - 0.4 * 2 * 0.2 = 0.04 \end{aligned}$$

导数(1)

$$\lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$



$$a = 2 \quad f(a) = 6$$

$$a = 2.001 \quad f(a) = 6.003$$

因此在 $a=2$ 处斜率为 3

$$a = 5 \quad f(a) = 15$$

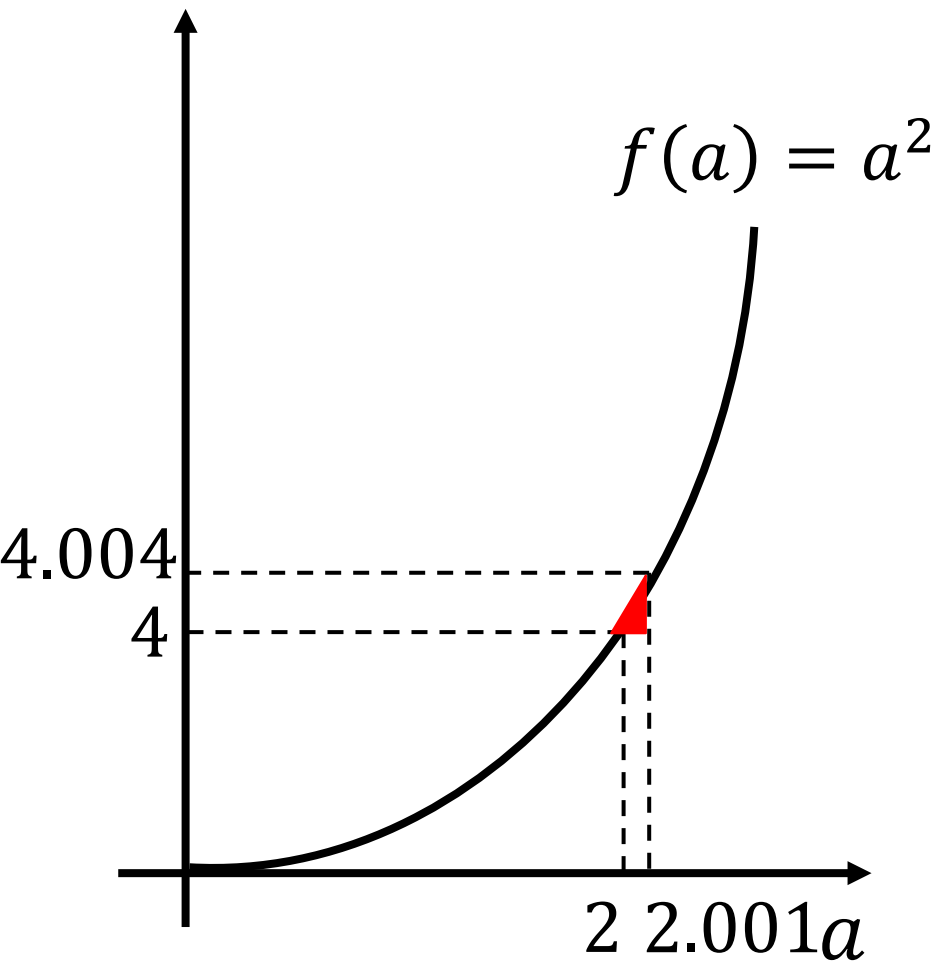
$$a = 5.001 \quad f(a) = 15.003$$

因此在 $a=5$ 处斜率为 3

$$\frac{df(a)}{da} = 3 = \frac{d}{da} f(a)$$

导数(2)

$$\lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$



$$\frac{df(a)}{da} = \frac{d}{da} a^2 = 2a$$

$$a = 2 \quad f(a) = 4$$

$$a = 2.001 \quad f(a) = 4.004001$$

当 $a = 2$ 时, 斜率为 4, $\frac{df(a)}{da} = 4$

$$a = 5 \quad f(a) = 25$$

$$a = 5.001 \quad f(a) = 25.010$$

(25.010001)

当 $a = 5$ 时, 斜率为 10, $\frac{df(a)}{da} = 10$

1.4 反向传播 (Backpropagation)

- Backpropagation是the backward propagation errors的缩写。
- 反向传播一般用于更新深度神经网络中的参数，具体是通过对损失函数求导来实现的。
- 反向传播包含两个环节：传播和参数更新。
- 利用计算图来解释反向传播算法。

1.4 反向传播

- 计算图导数计算

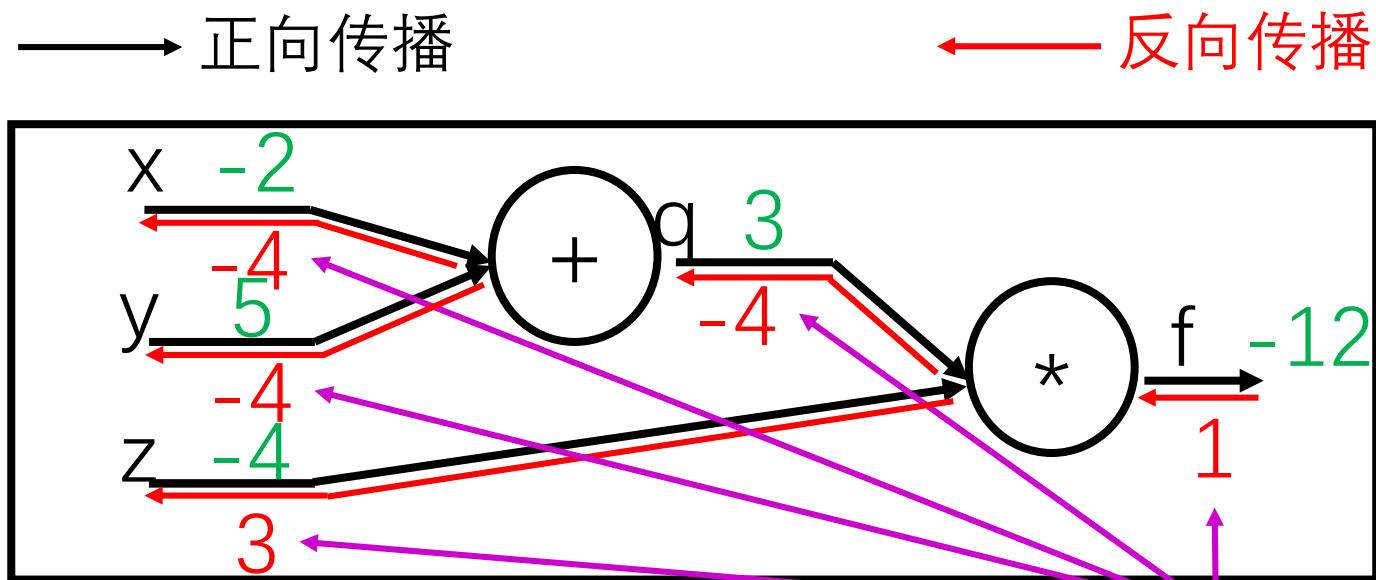
$$f(x, y, z) = (x + y)z$$

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

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

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

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

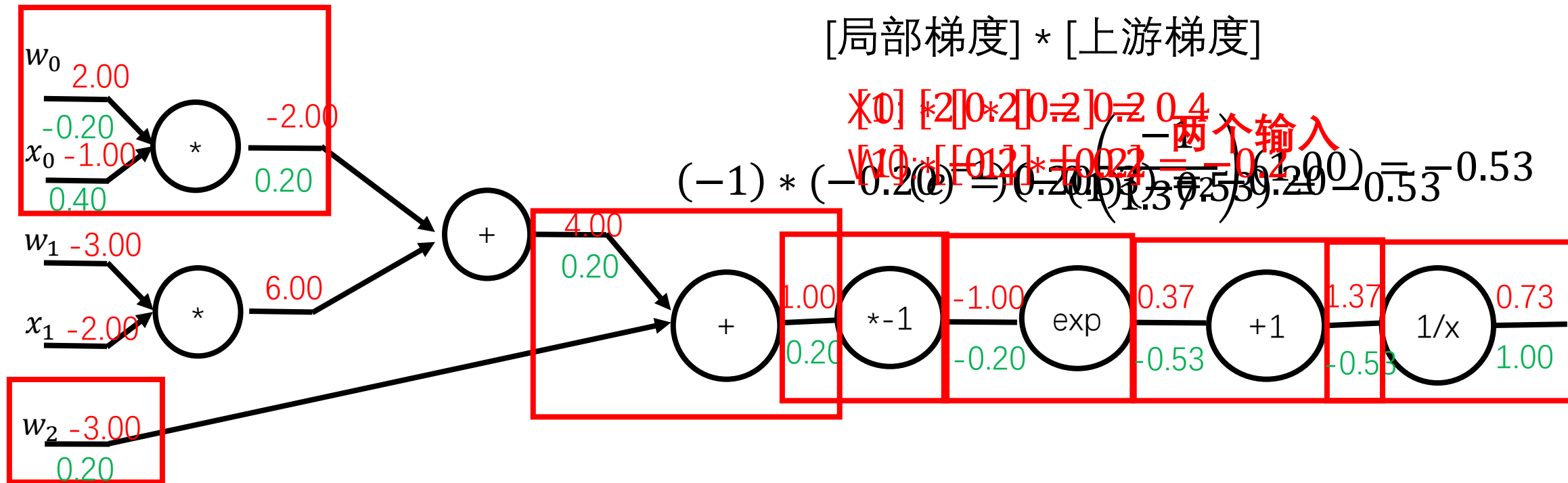


链式法则:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q} \frac{\partial q}{\partial x} = -4 \cdot 1 = -4$$

另外一个例子:

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



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

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

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

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

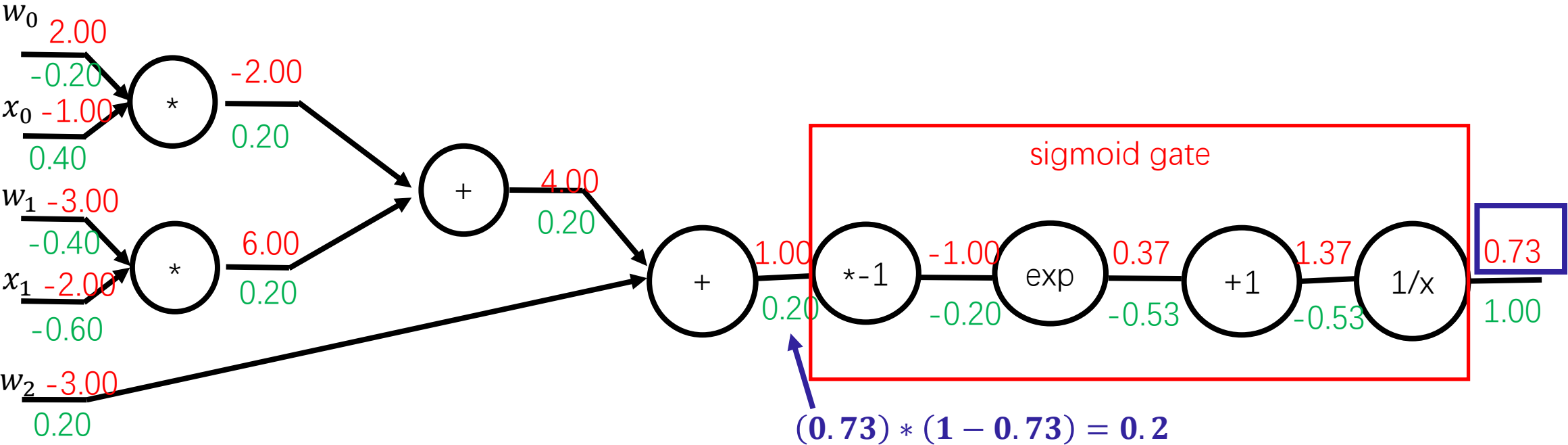
另外一个例子: $f(w, x) = \frac{1}{1 + e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$

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

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

sigmoid function

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



反向传播

- Multiclass loss:

$$L(W, b) = \frac{1}{N} \sum_{i=1}^N L_i$$

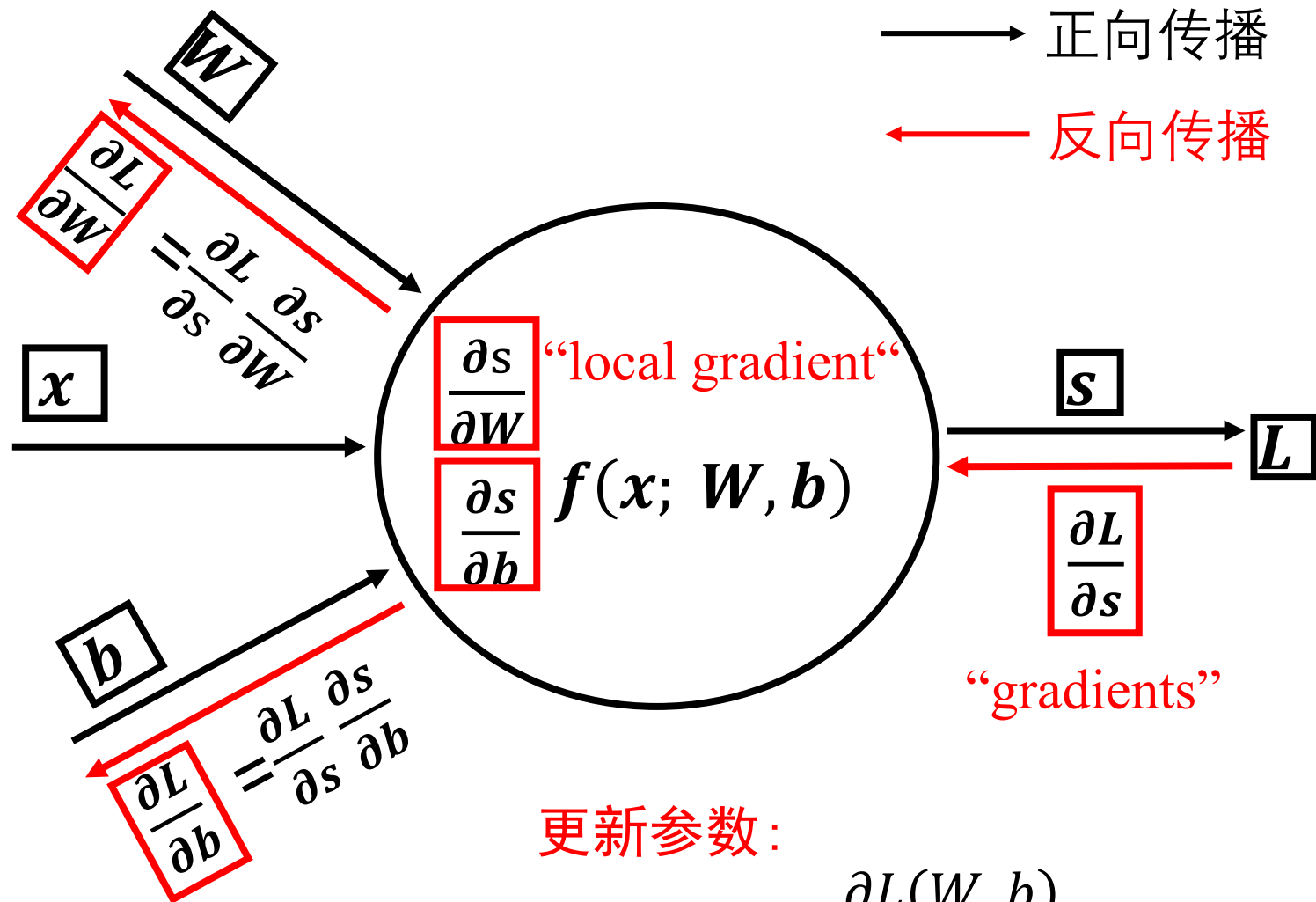
其中

Score function:

$$s = f(x_i; W, b) = Wx_i + b$$

损失函数:

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



更新参数:

$$W := W - \alpha \frac{\partial L(W, b)}{\partial W}$$
$$b := b - \alpha \frac{\partial L(W, b)}{\partial b}$$