



北京航空航天大学
COLLEGE OF SOFTWARE 软件学院
BEIHANG UNIVERSITY

人工智能

第10讲：机器学习-深度学习 I

张晶

2023年春季

- 参考教材： 吴飞，《人工智能导论：模型与算法》，高等教育出版社
- 在线课程：<https://www.icourse163.org/course/ZJU-1003377027?from=searchPage>
- 本部分参考：李宏毅，《机器学习》课程，台湾大学



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

提纲

一、线性回归与梯度下降

二、前馈神经网络

三、卷积神经网络

四、序列数据模型

五、深度学习应用



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

提纲

一、线性回归与梯度下降

二、前馈神经网络

三、卷积神经网络

四、序列数据模型

五、深度学习应用

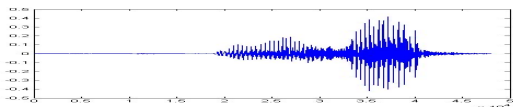


北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

机器（深度）学习：从数据中学习知识

- 语音识别

$$f(\text{语音数据}) = \text{“你好”}$$



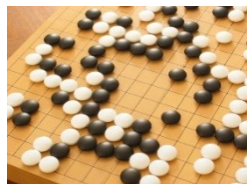
- 图像分类

$$f(\text{图像数据}) = \text{“猫”}$$



- 围棋游戏

$$f(\text{棋局数据}) = \text{“5-5”}$$



(下一步落子位置)

- 从原始数据中提取特征
- 学习映射函数 f
- 通过映射函数 f 将原始数据映射到语义任务空间，即寻找数据和任务目标之间的关系

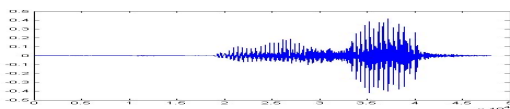


北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

机器（深度）学习 \approx 函数拟合

- 语音识别

$$f(\text{语音波形图}) = \text{“你好”}$$



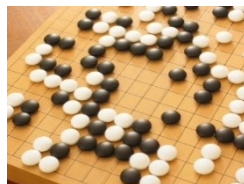
- 图像分类

$$f(\text{猫咪照片}) = \text{“猫”}$$



- 围棋游戏

$$f(\text{围棋棋盘}) = \text{“5-5”}$$



(下一步落子位置)

- 从原始数据中提取特征
- 学习映射函数 f
- 通过映射函数 f 将原始数据映射到语义任务空间，即寻找数据和任务目标之间的关系



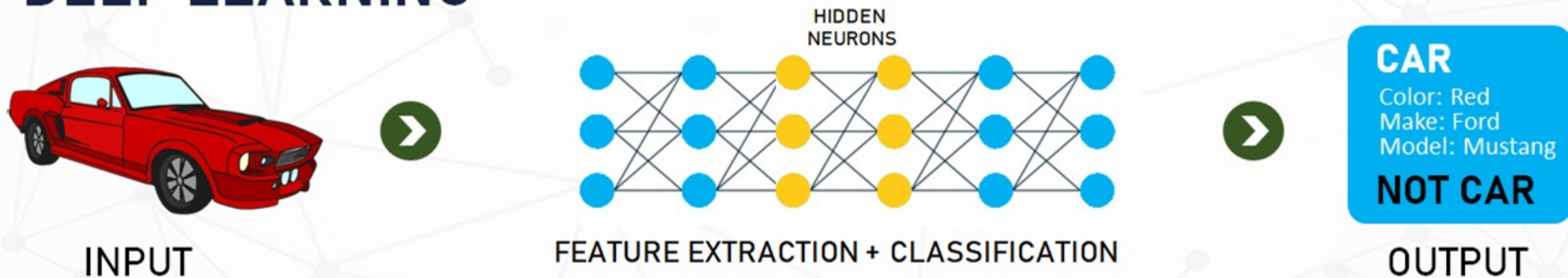
北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

机器学习 v. s. 深度学习

MACHINE LEARNING



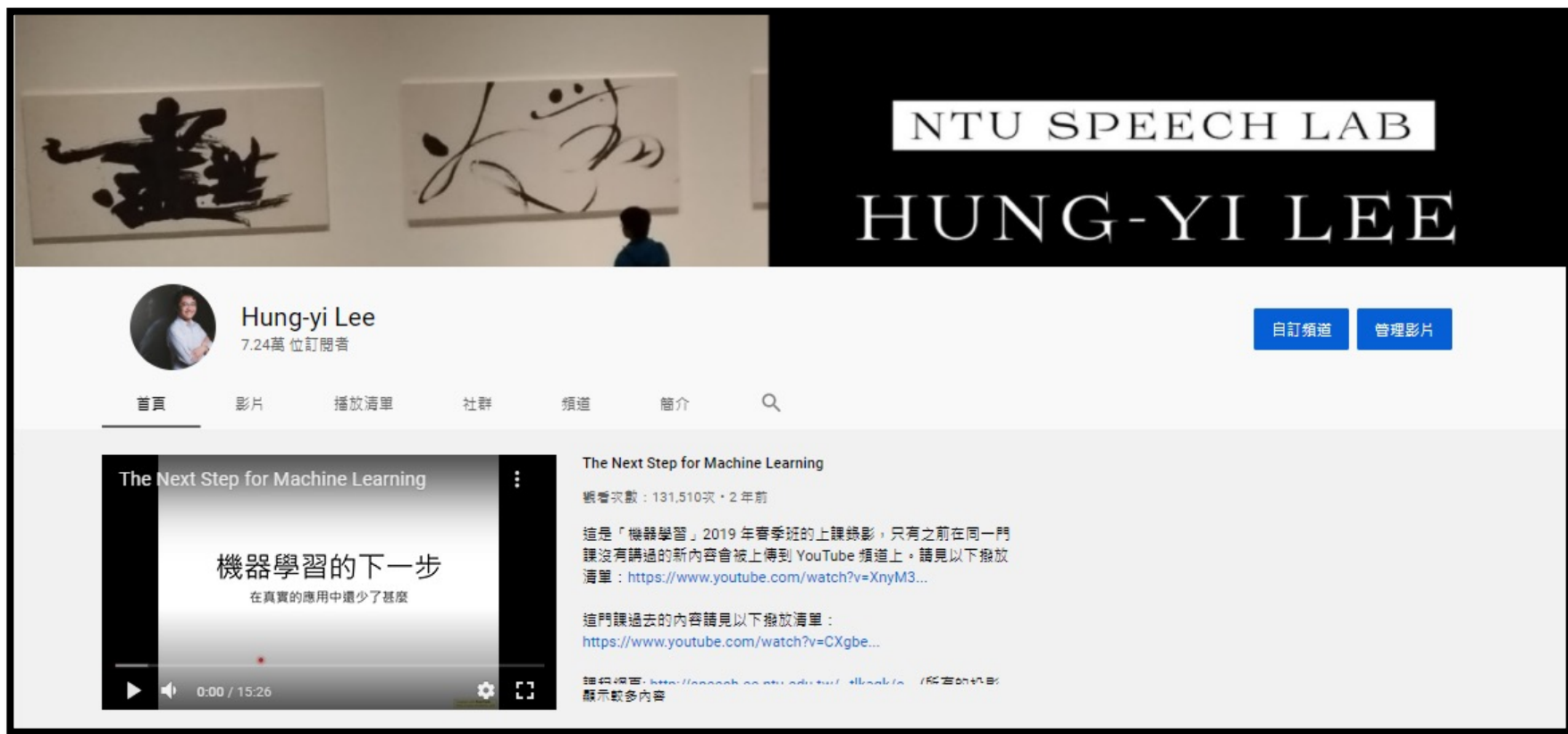
DEEP LEARNING





北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

YouTube 视频观看量预测




<https://www.youtube.com/c/HungyiLeeNTU>



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

函数拟合

$y = f($
2/26日
观看量

 Hung-yi Lee						
篩選器						
影片	流量來源	地理位置	觀眾年齡	觀眾性別	日期	訂閱狀態
日期 ↓				+	喜歡的人數	訂閱人數
2021年1月26日				54 4.9%	69 5.5%	6,788 5.2%
2021年1月27日				60 5.4%	71 5.6%	6,242 4.7%
2021年1月28日				36 3.2%	63 5.0%	5,868 4.5%
2021年1月29日				27 2.4%	40 3.2%	4,413 3.4%
2021年1月30日				40 3.6%	40 3.2%	4,372 3.3%
2021年1月31日				47 4.2%	51 4.0%	5,135 3.9%
2021年2月1日				61 5.5%	29 2.3%	5,527 4.2%
2021年2月2日				49 4.4%	43 3.4%	5,911 4.5%
2021年2月3日				26 2.3%	44 3.5%	5,248 4.0%
2021年2月4日				43 3.9%	33 2.6%	4,771 3.6%
2021年2月5日				45 4.0%	49 3.9%	3,850 2.9%
2021年2月6日				29 2.6%	42 3.3%	3,828 2.9%
2021年2月7日				26 2.3%	46 3.6%	4,559 3.5%
2021年2月8日				38 3.4%	26 2.1%	4,772 3.6%
2021年2月9日				29 2.6%	25 2.0%	3,847 2.9%
2021年2月10日				31 2.8%	35 2.8%	3,382 2.6%



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

1. 定义带有未知参数的函数

$$y = f($$



模型

$$y = b + wx_1$$

基于领域知识定义模型

y : 2/26日观看量

输出结果

x_1 : 2/25日观看量

输入特征

w 和 b 为未知参数(需要从数据中学习)

weight bias

日期	新增影片数	点赞人数	评论人数	播放次数	观看次数	观看时长(小时)	平均观看时长
统计	199	17,022	26,011	27,602,732	2,066,634	268,778.0	7:48
2020年1月1日	-	16 0.1%	52 0.2%	57,093	3,977 0.2%	565.6 0.2%	8:32
2020年1月2日	-	33 0.2%	58 0.2%	56,204	4,214 0.2%	589.8 0.2%	8:23
2020年1月3日	-	24 0.1%	89 0.3%	53,321	3,288 0.2%	457.4 0.2%	8:20
2020年1月4日	1 0.5%	27 0.2%	66 0.3%	53,599	3,559 0.2%	483.5 0.2%	8:09
2020年1月5日	-	35 0.2%	85 0.3%	63,001	4,677 0.2%	596.4 0.2%	7:39
2020年1月6日	-	31 0.2%	69 0.3%	60,175	4,682 0.2%	642.0 0.2%	8:13
2020年1月7日	-	40 0.2%	70 0.3%	63,638	4,895 0.2%	618.4 0.2%	7:54
2020年1月8日	-	39 0.2%	59 0.2%	59,900	4,785 0.2%	646.7 0.2%	8:06
2020年1月9日	-	28 0.2%	64 0.3%	54,988	4,911 0.2%	670.9 0.3%	8:11
2020年1月10日	-	17 0.1%	51 0.2%	40,631	3,069 0.2%	372.0 0.1%	7:16
2020年1月11日	-	12 0.1%	54 0.2%	36,168	2,898 0.1%	369.5 0.1%	7:38
2020年1月12日	-	40 0.2%	169 0.7%	53,964	4,477 0.2%	572.9 0.2%	7:40
2020年1月13日	-	29 0.2%	75 0.3%	61,043	5,017 0.2%	661.4 0.3%	7:54
2020年1月14日	-	32 0.2%	83 0.3%	64,968	5,186 0.3%	618.3 0.2%	7:09

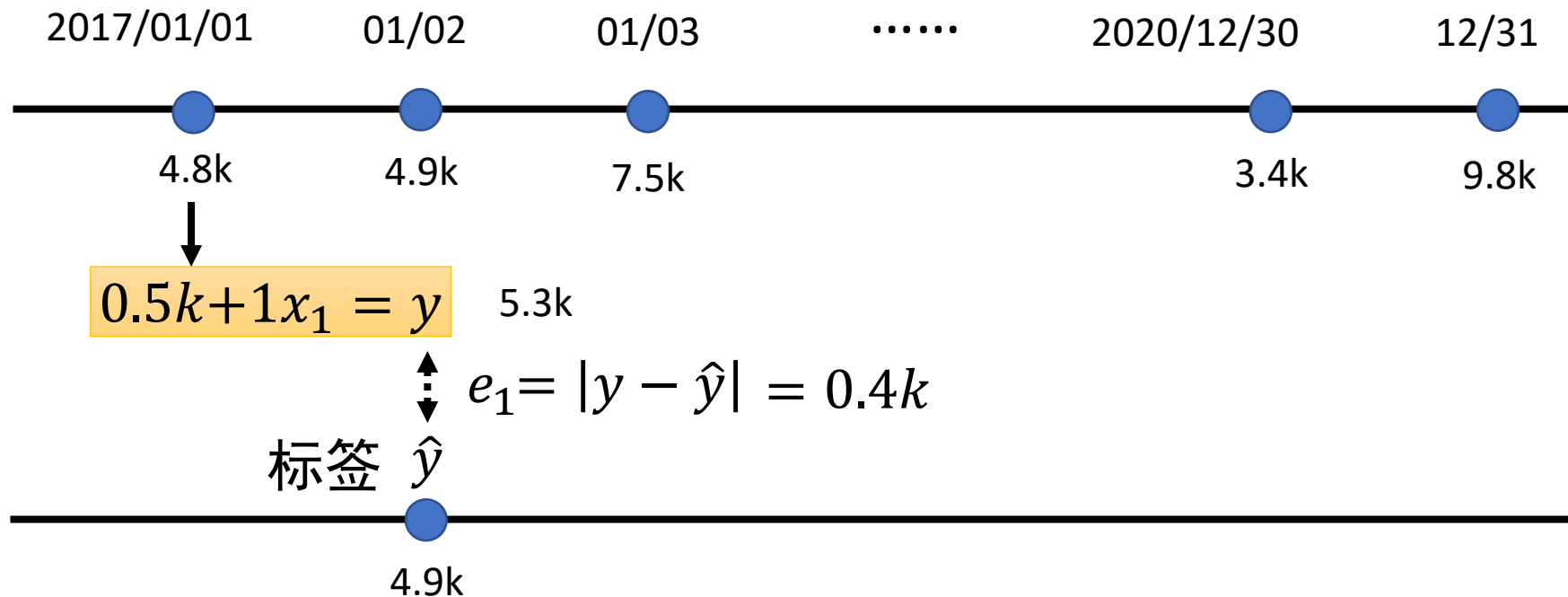


2. 从训练数据中定义损失函数

- 关于未知参数的损失函数 $L(b, w)$
- 损失：模型预测值与真实值间的误差

$$L(0.5k, 1) \quad y = b + wx_1 \longrightarrow y = 0.5k + 1x_1 \quad \text{拟合程度?}$$

训练数据：2017/01/01 – 2020/12/31



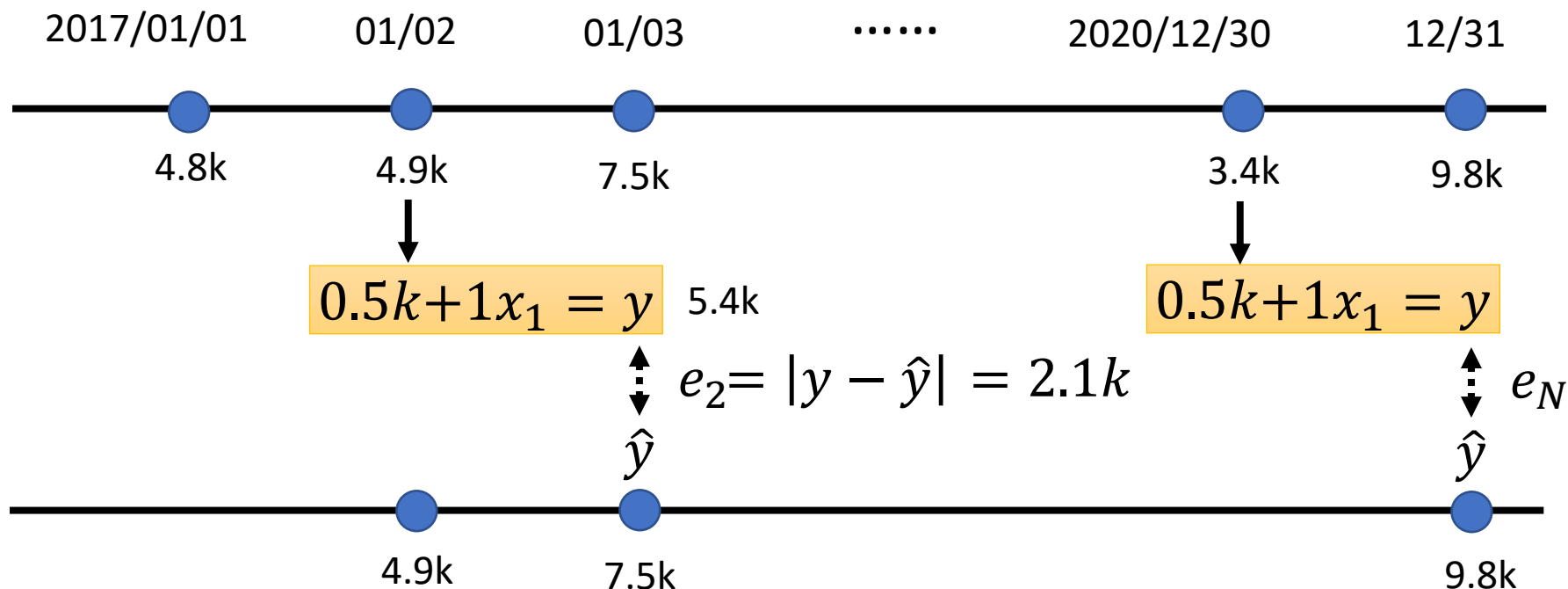


2. 从训练数据中定义损失函数

- 关于未知参数的损失函数 $L(b, w)$
- 损失：模型预测值与真实值间的误差

$$L(0.5k, 1) \quad y = b + wx_1 \longrightarrow y = 0.5k + 1x_1 \quad \text{拟合程度?}$$

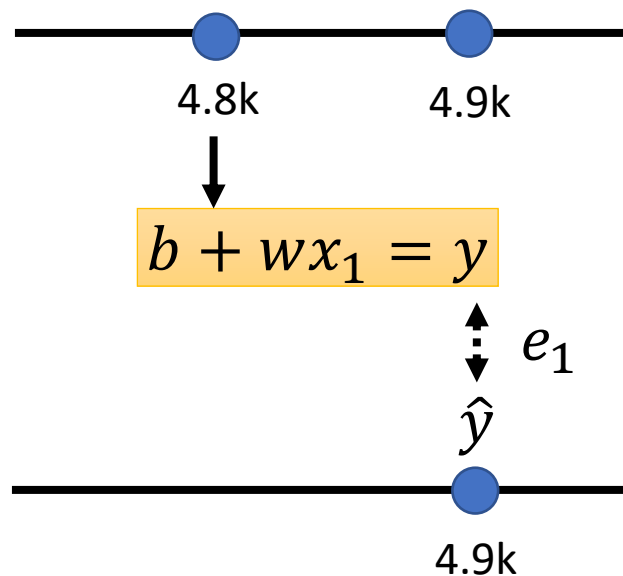
Data from 2017/01/01 – 2020/12/31





2. 从训练数据中定义损失函数

- 关于未知参数的损失函数 $L(b, w)$
- 损失：模型预测值与真实值间的误差



Loss:
$$L = \frac{1}{N} \sum_n e_n$$

$e = |y - \hat{y}|$ L 为平均绝对损失, mean absolute error (MAE)

$e = (y - \hat{y})^2$ L 为均方误差损失, mean square error (MSE)

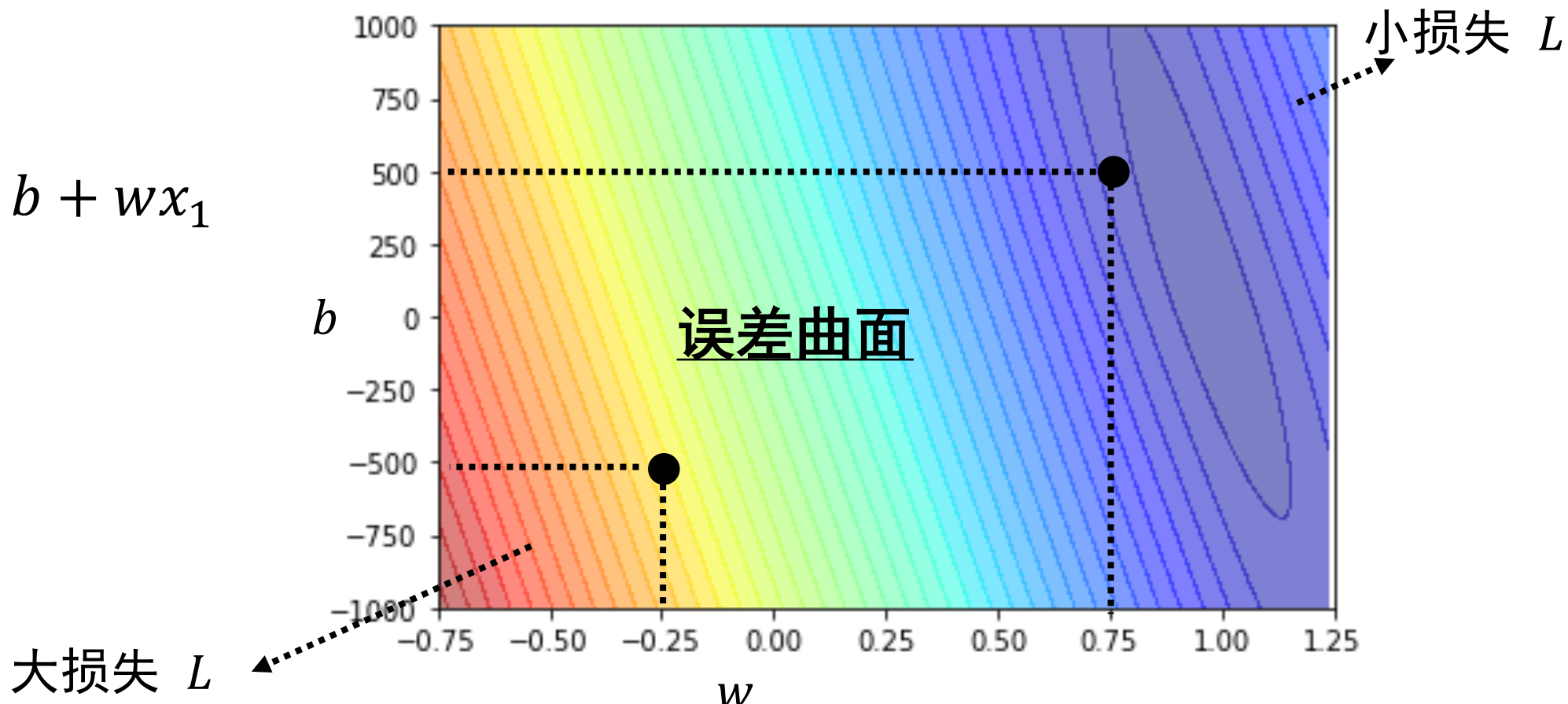
如 y 和 \hat{y} 均为概率分布 \longrightarrow 交叉熵损失, Cross-entropy



2. 从训练数据中定义损失函数

- 关于未知参数的损失函数 $L(b, w)$
- 损失：模型预测值与真实值间的误差

模型 $y = b + wx_1$





3. 优化

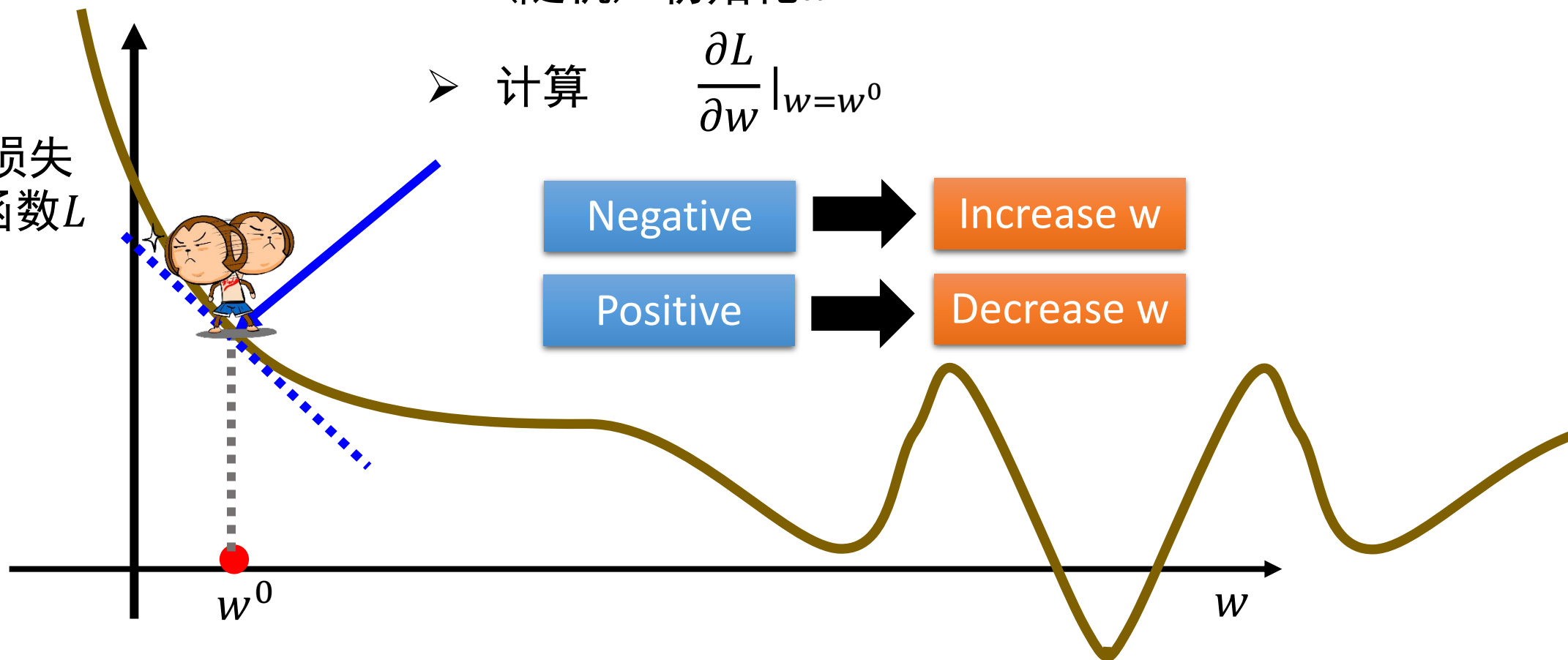
$$w^* = \arg \min_w L$$

梯度下降

➤ (随机) 初始化 w^0

➤ 计算 $\frac{\partial L}{\partial w} \big|_{w=w^0}$

损失
函数 L



Negative

Increase w

Positive

Decrease w



3. 优化

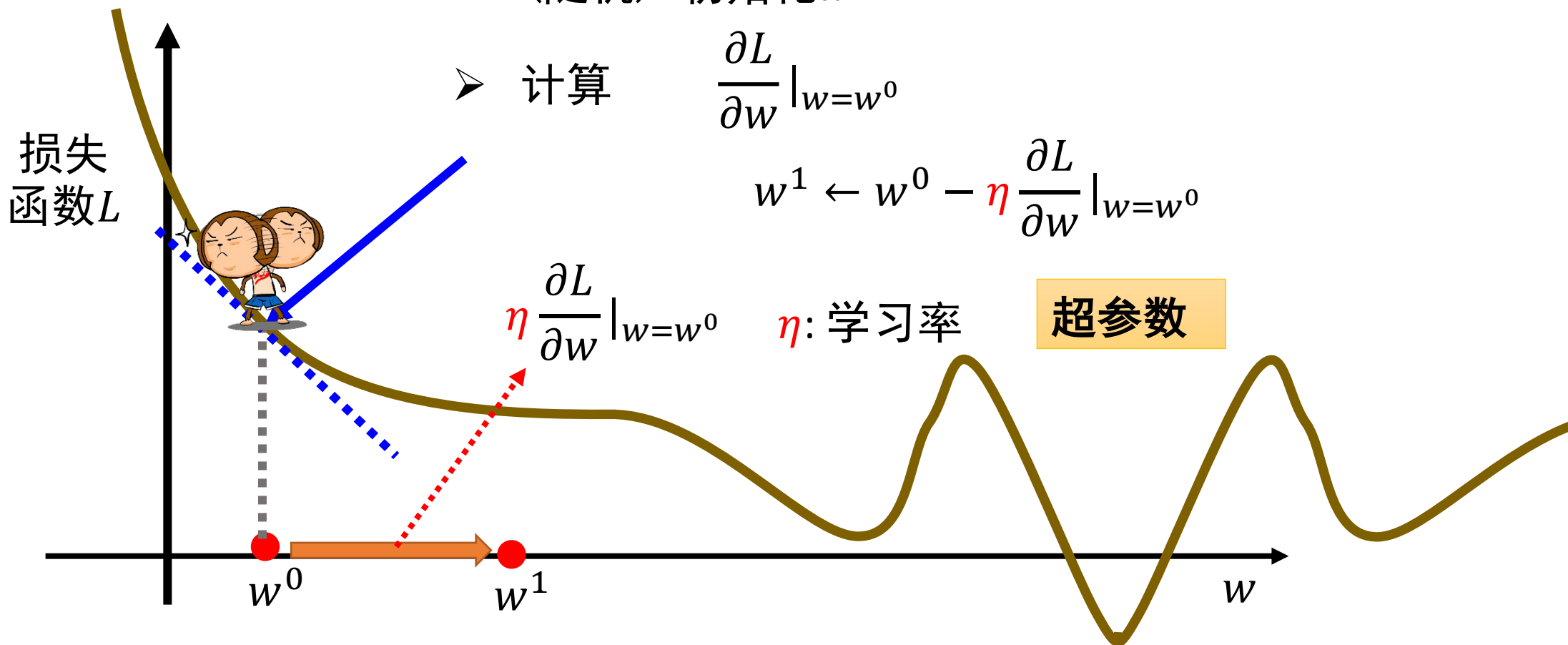
$$w^* = \arg \min_w L$$

梯度下降

➤ (随机) 初始化 w^0

➤ 计算 $\frac{\partial L}{\partial w} \big|_{w=w^0}$

$$w^1 \leftarrow w^0 - \eta \frac{\partial L}{\partial w} \big|_{w=w^0}$$





3. 优化

$$w^* = \arg \min_w L$$

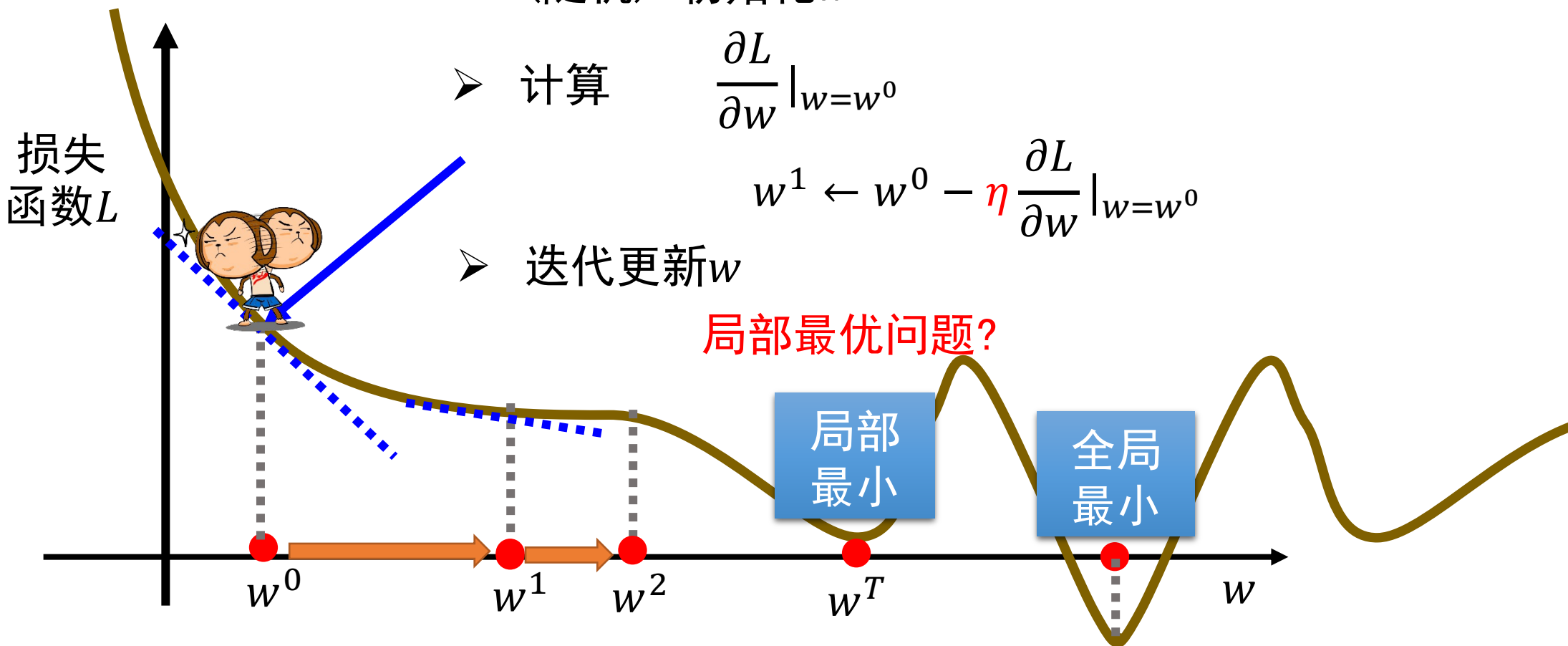
梯度下降

➤ (随机) 初始化 w^0

➤ 计算 $\frac{\partial L}{\partial w} \big|_{w=w^0}$

$$w^1 \leftarrow w^0 - \eta \frac{\partial L}{\partial w} \big|_{w=w^0}$$

➤ 迭代更新 w





3. 优化

$$w^*, b^* = \arg \min_{w, b} L$$

- (随机) 初始化 w^0, b^0
- 计算

$$\begin{aligned} \frac{\partial L}{\partial w} \Big|_{w=w^0, b=b^0} \\ \frac{\partial L}{\partial b} \Big|_{w=w^0, b=b^0} \end{aligned}$$



深度学习框架下一行代码足以实现！

- 迭代更新 w 和 b

$$w^1 \leftarrow w^0 - \eta \frac{\partial L}{\partial w} \Big|_{w=w^0, b=b^0}$$

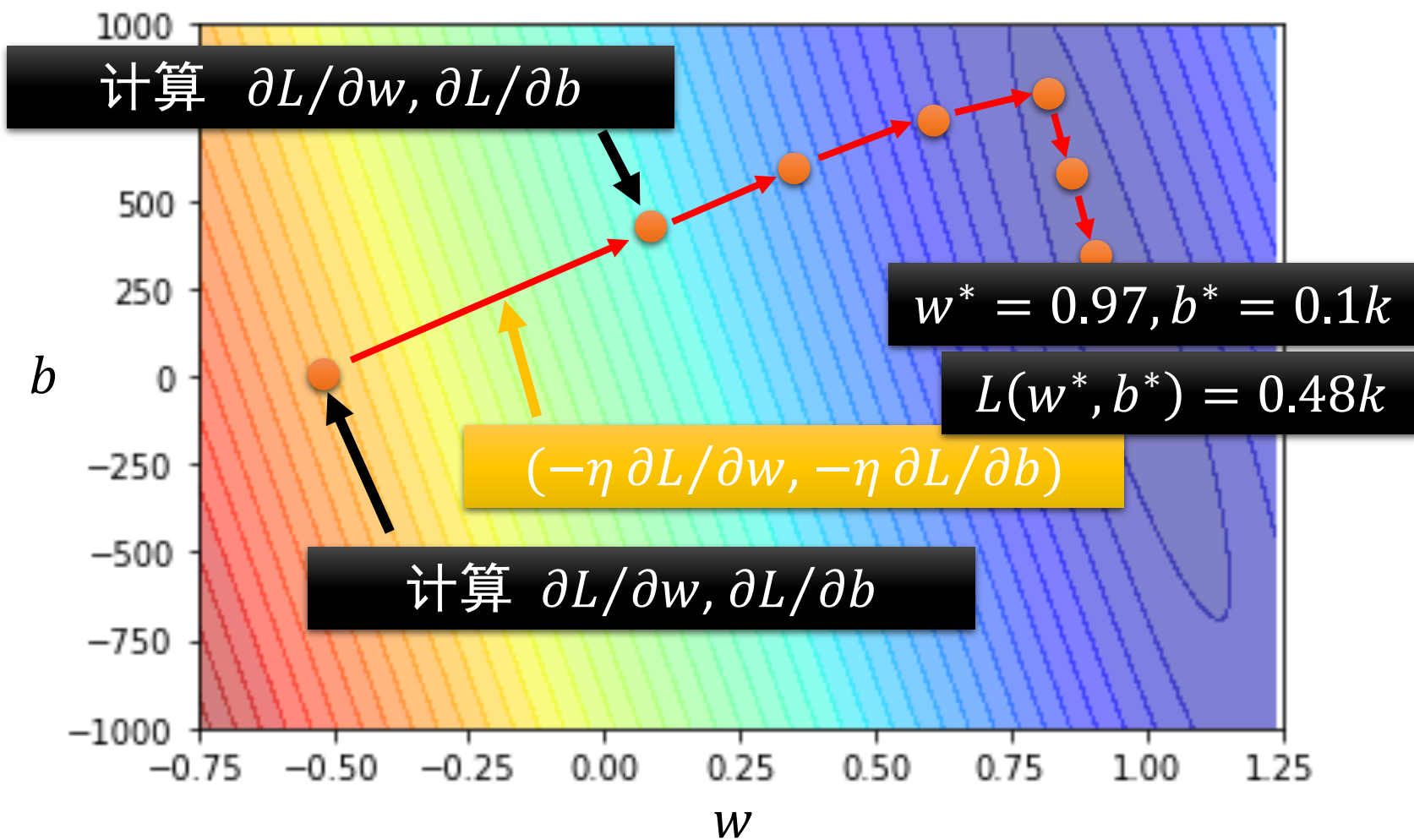
$$b^1 \leftarrow b^0 - \eta \frac{\partial L}{\partial b} \Big|_{w=w^0, b=b^0}$$



3. 优化

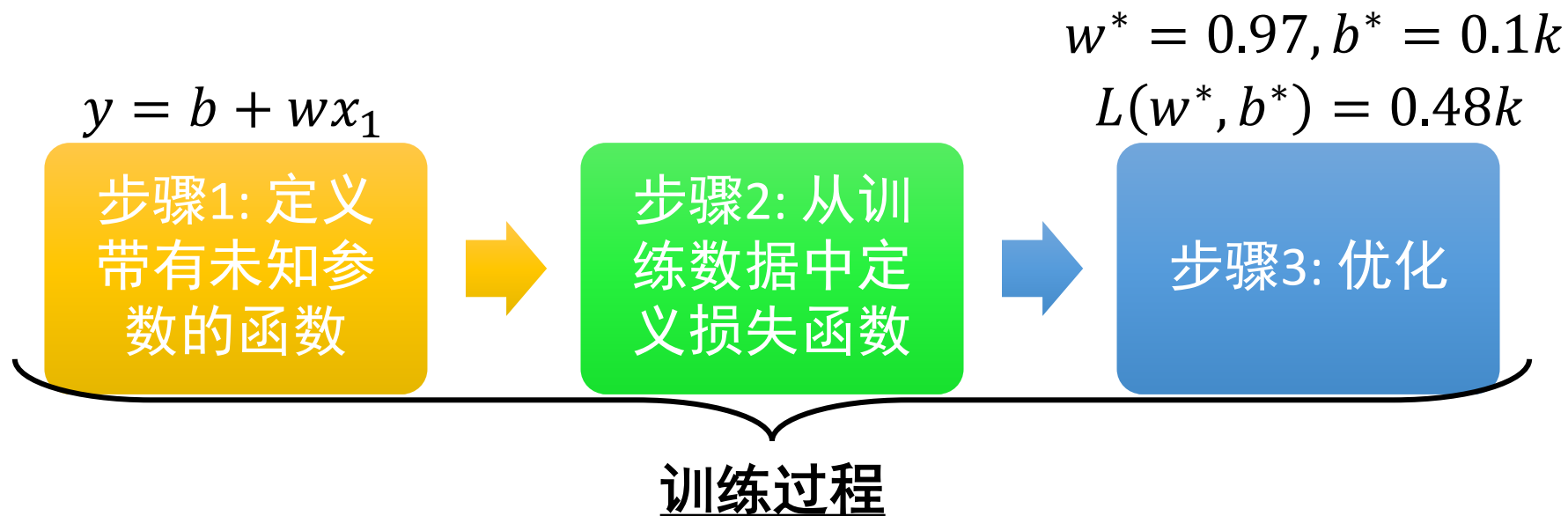
模型 $y = b + wx_1$

$$w^*, b^* = \arg \min_{w, b} L$$





深度学习主要步骤



$y = 0.1k + 0.97x_1$ 可以在2017 – 2020的数据上（训练数据）得到最小的损失 $L = 0.48k$

2021年数据上表现如何 (训练过程中未见过的数据)? $L' = 0.58k$



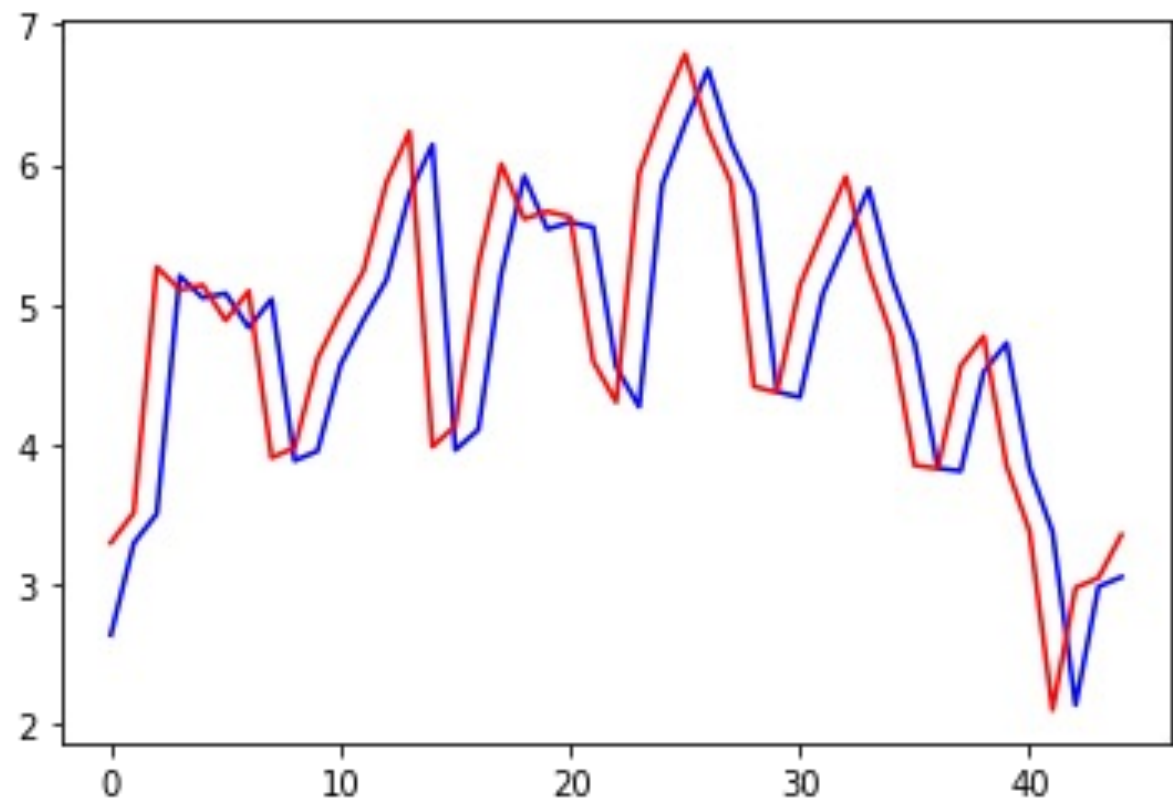
北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

$$y = 0.1k + 0.97x_1$$

Red: 真实观看数量

blue: 预测观看数量

观看数量
(k)



2021/01/01

2021/02/14



$$y = b + wx_1$$

2017 - 2020

$$L = 0.48k$$

2021

$$L' = 0.58k$$

$$y = b + \sum_{j=1}^7 w_j x_j$$

2017 - 2020

$$L = 0.38k$$

2021

$$L' = 0.49k$$

b	w_1^*	w_2^*	w_3^*	w_4^*	w_5^*	w_6^*	w_7^*
0.05k	0.79	-0.31	0.12	-0.01	-0.10	0.30	0.18

$$y = b + \sum_{j=1}^{28} w_j x_j$$

2017 - 2020

$$L = 0.33k$$

2021

$$L' = 0.46k$$

$$y = b + \sum_{j=1}^{56} w_j x_j$$

2017 - 2020

$$L = 0.32k$$

2021

$$L' = 0.46k$$

线性模型



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

提纲

一、线性回归与梯度下降

二、前馈神经网络

三、卷积神经网络

四、序列数据模型

五、深度学习应用



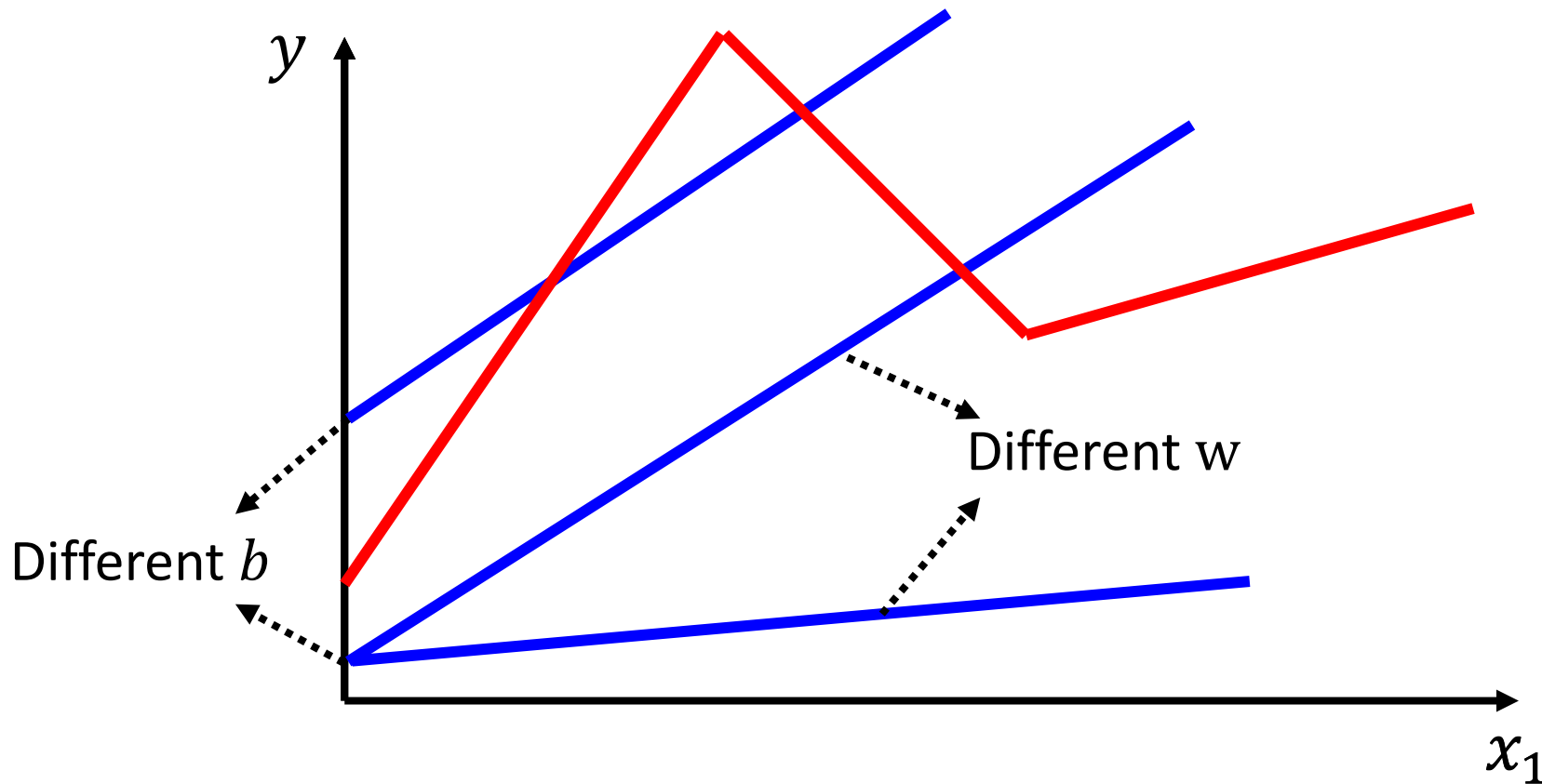
为何着重讨论线性模型? ($y = b + wx_1$)

- 回顾传统机器学习模型，很多基于线性模型
 - Mean Squared Error (generally for regression)
 - Perceptron
 - Fisher's linear discriminant
 - Support Vector Machines (SVM)
 - Logistic Regression
 - ...
- 前馈神经网络也建立在线性模型基础之上



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

线性模型的扩展

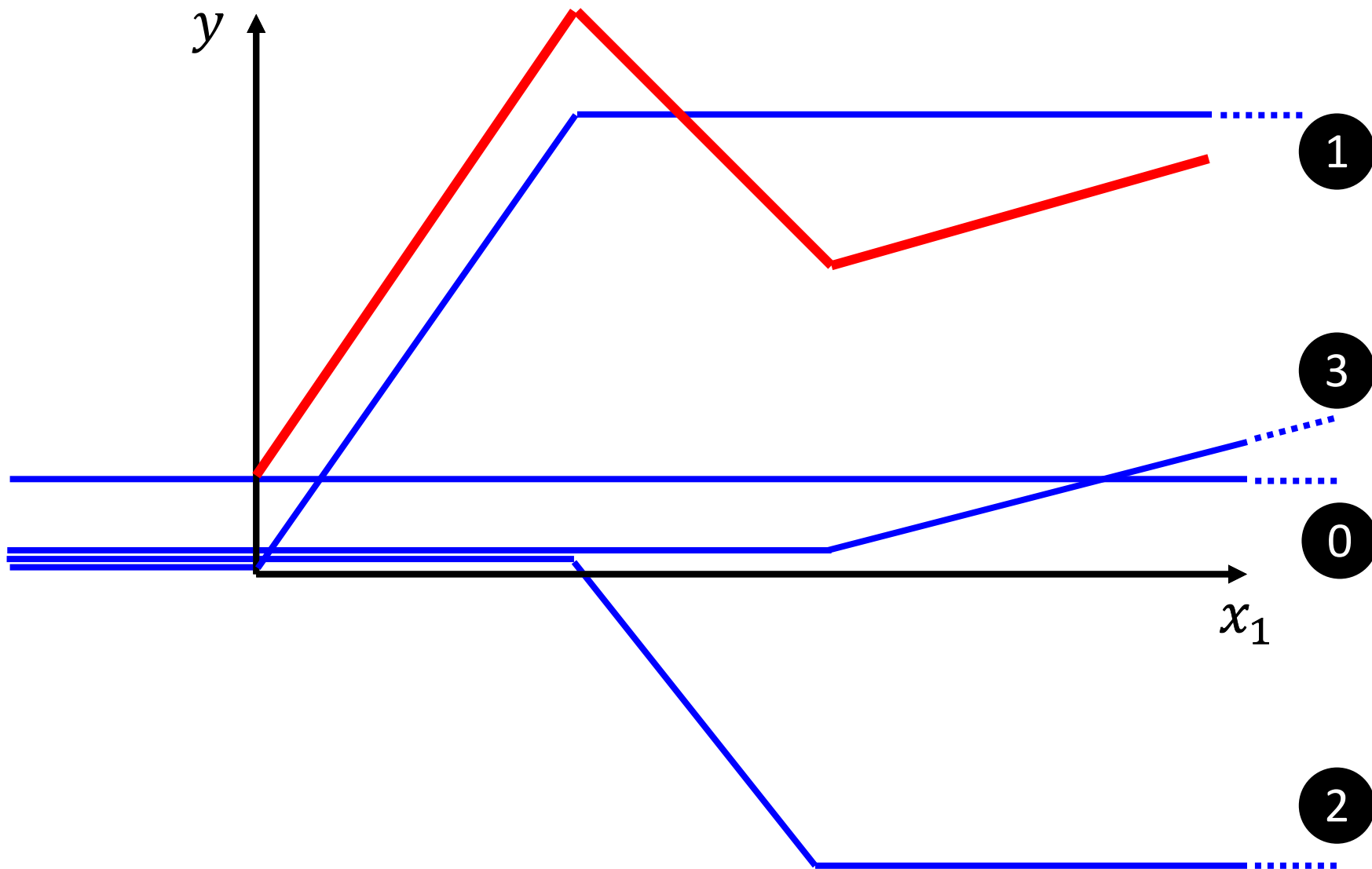


线性模型过于简单(欠拟合)，为了拟合复杂数据需要更加复杂的模型。



线性模型的扩展

red curve = constant + sum of a set of





北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

分段线性函数

= constant + sum of a set of



More pieces require more

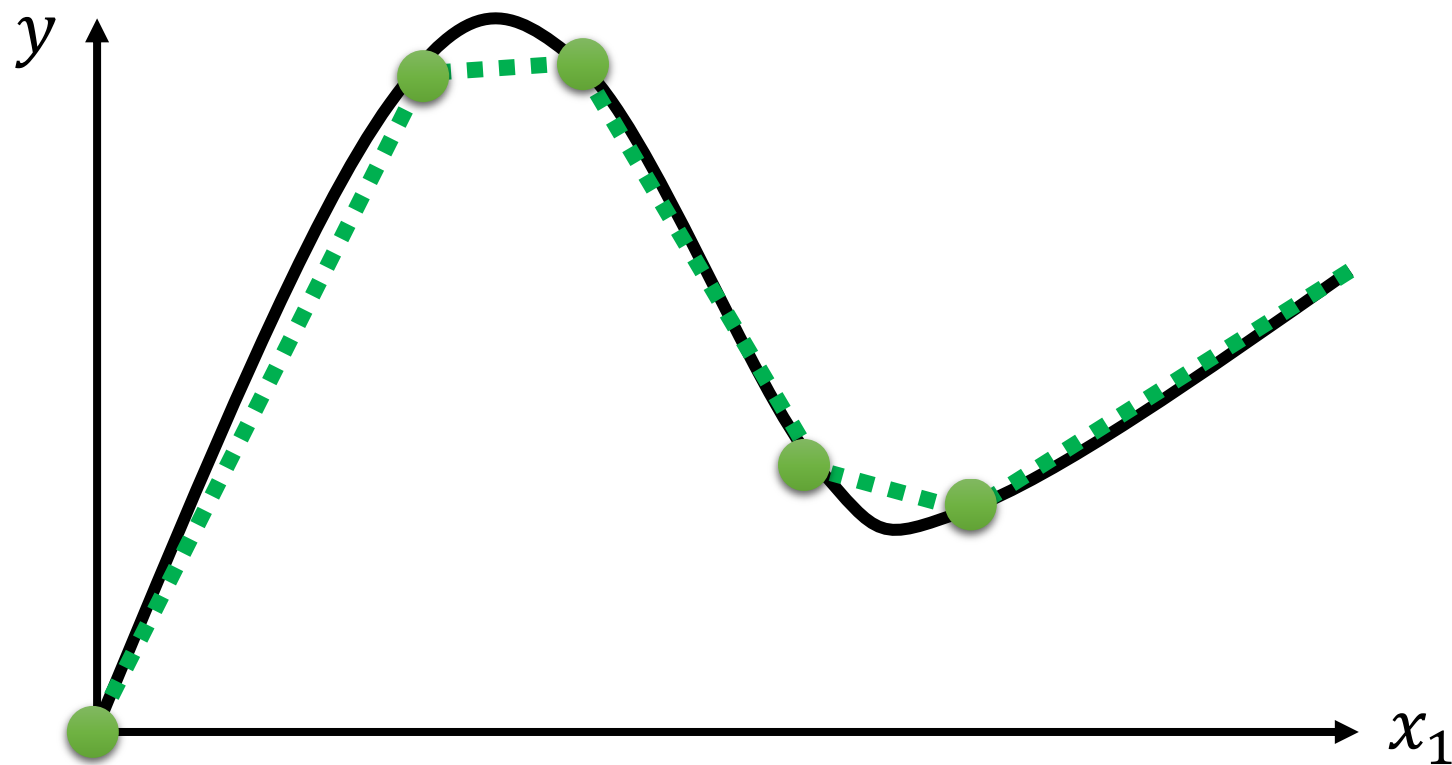




北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

分段线性模型

用分段线性模型拟合连续曲线



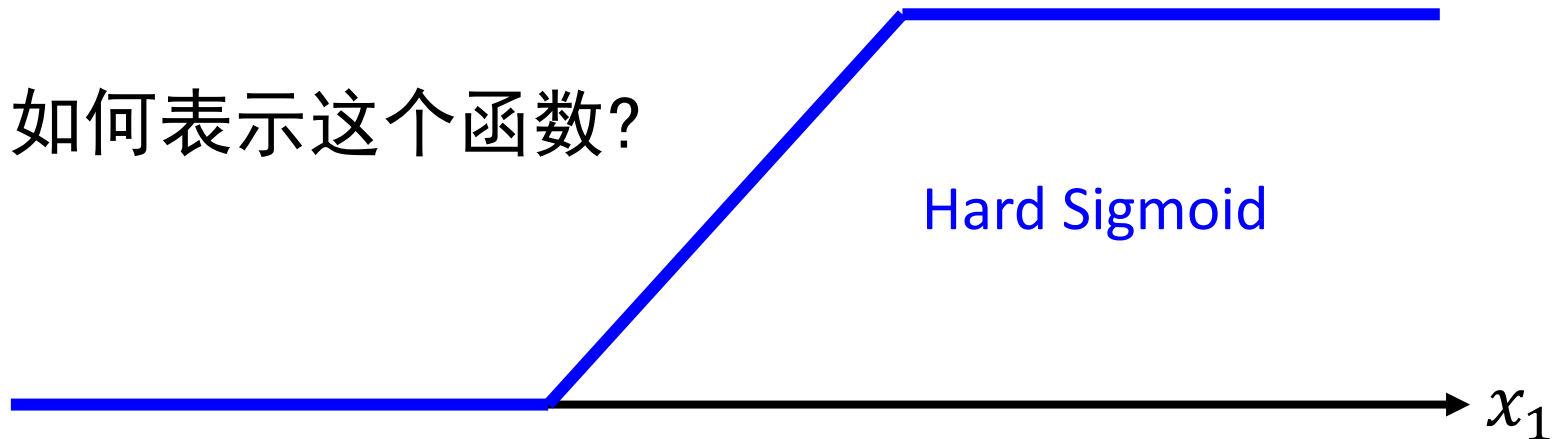
为了更好的近似连续曲线，需要更多分段线性模型



red curve = constant + sum of a set of



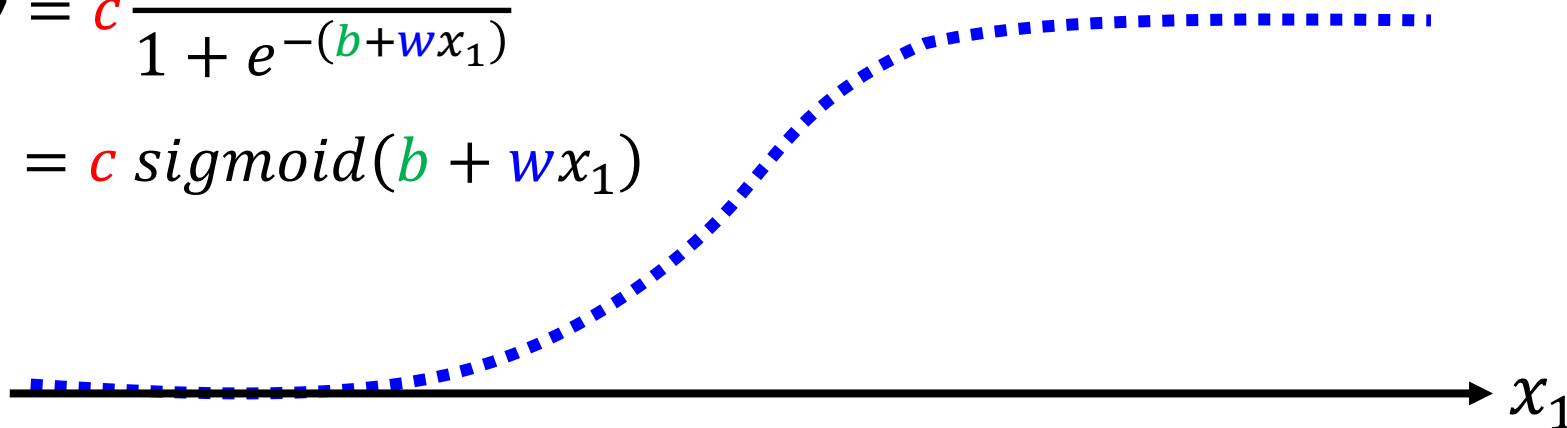
如何表示这个函数?



Hard Sigmoid

Sigmoid 函数

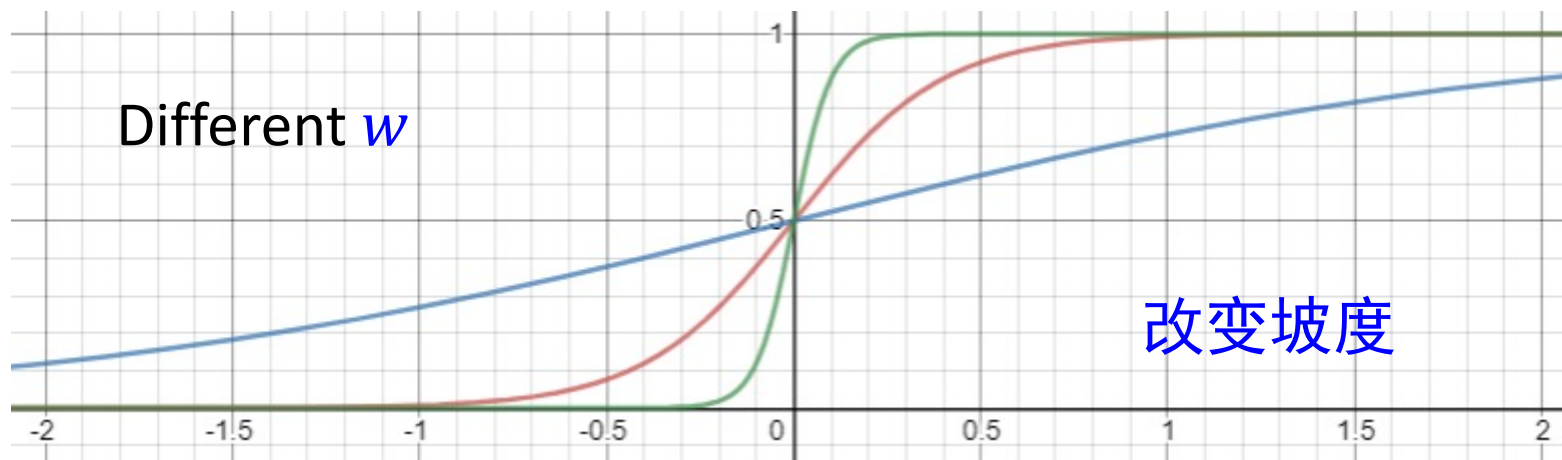
$$y = c \frac{1}{1 + e^{-(b + wx_1)}}$$
$$= c \operatorname{sigmoid}(b + wx_1)$$



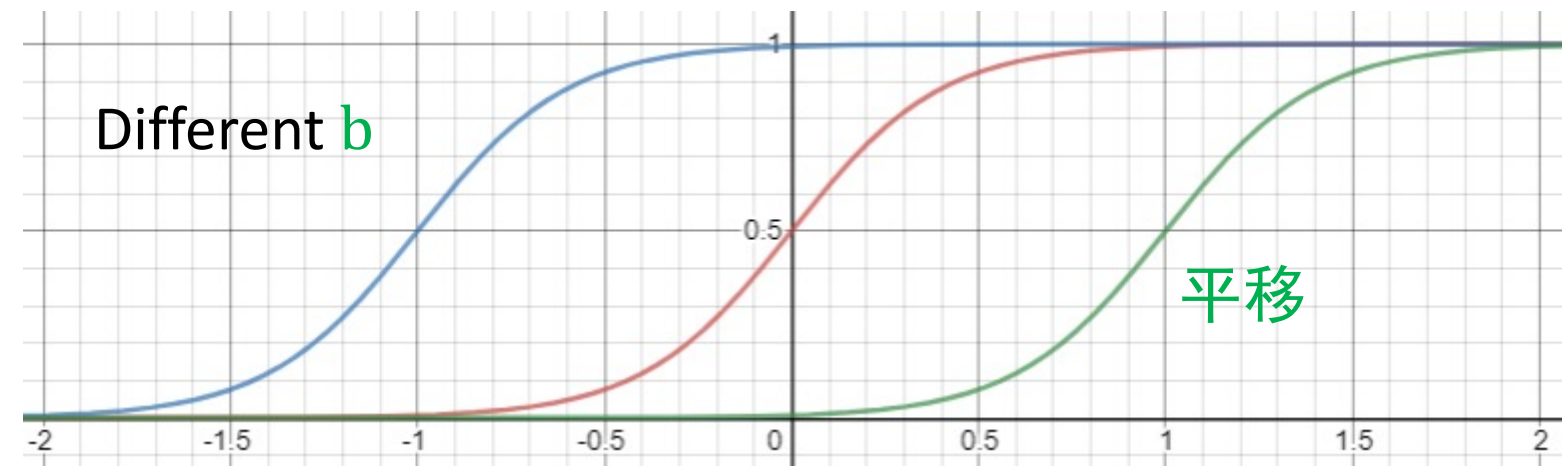


北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

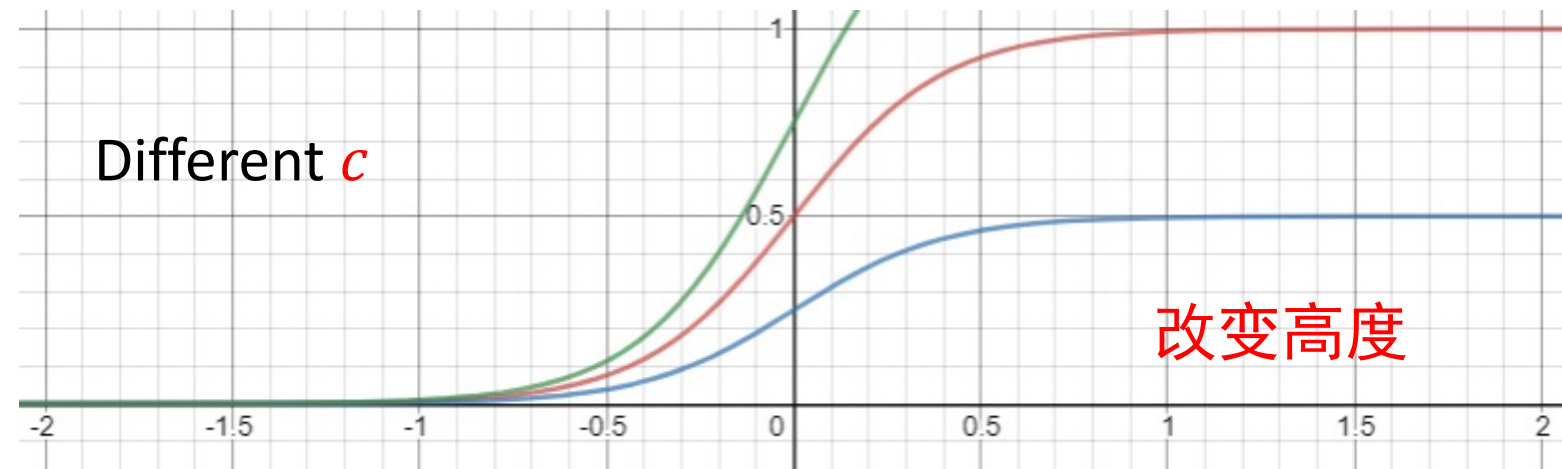
Different w



Different b

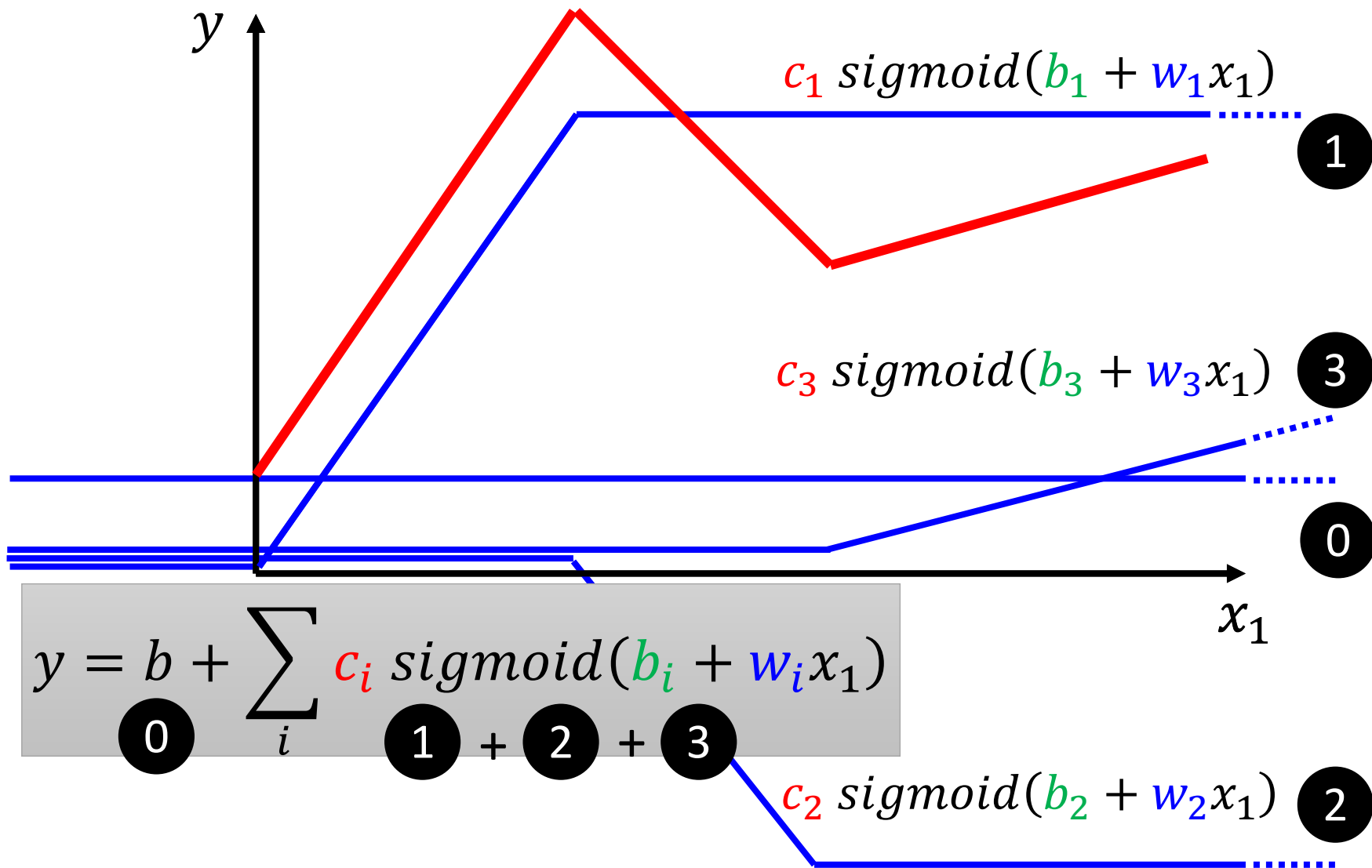


Different c





red curve = sum of a set of  + constant





新模型：更多模型组合

$$y = b + \underline{wx_1}$$



$$y = b + \sum_i c_i \operatorname{sigmoid}(\underline{b_i + w_i x_1})$$

$$y = b + \underline{\sum_j w_j x_j}$$

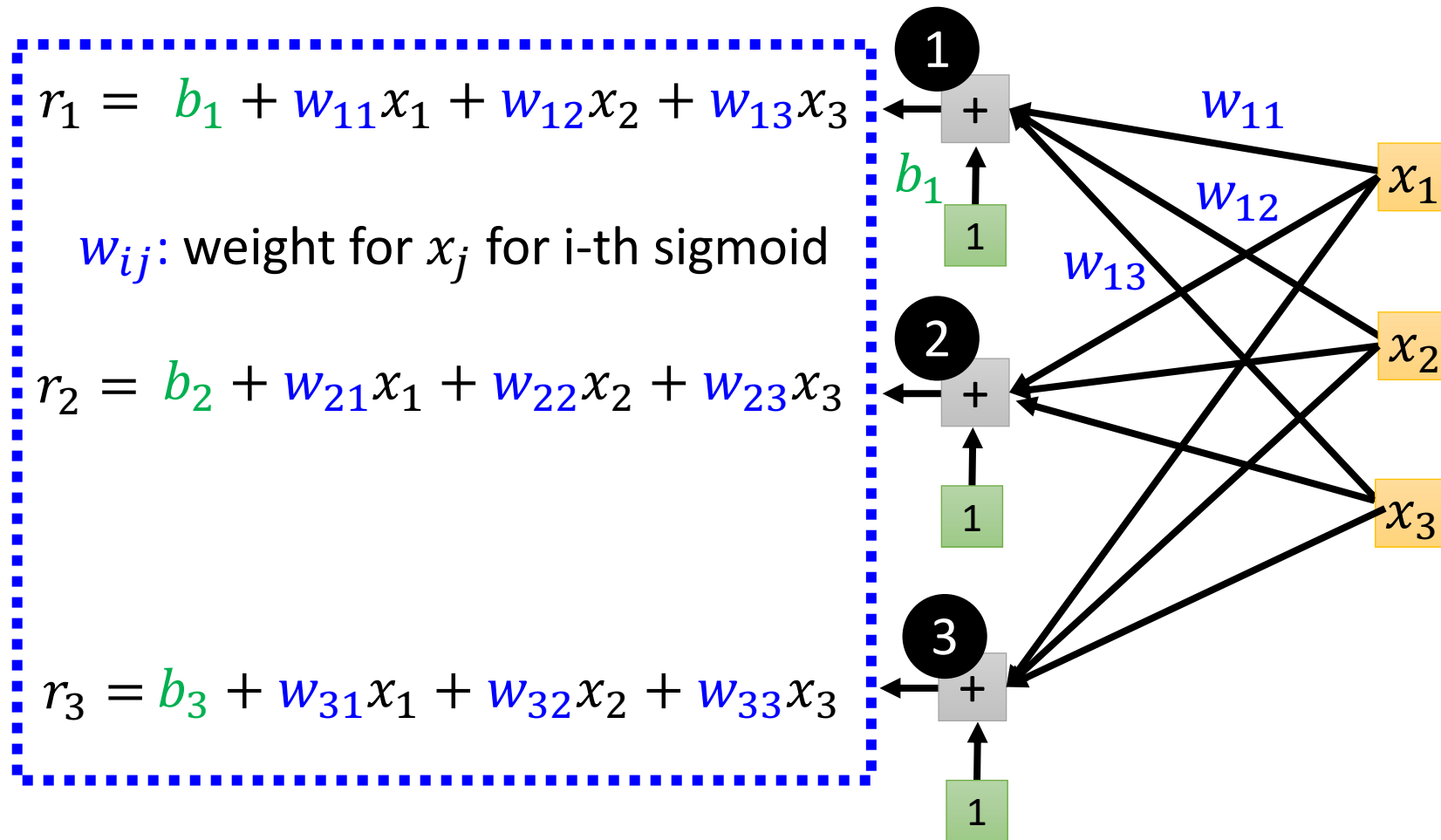


$$y = b + \sum_i c_i \operatorname{sigmoid}\left(\underline{b_i + \sum_j w_{ij} x_j}\right)$$



$$y = b + \sum_i c_i \operatorname{sigmoid} \left(b_i + \sum_j w_{ij} x_j \right)$$

$j: 1, 2, 3$ 特征个数
 $i: 1, 2, 3$ sigmoid个数





$$y = b + \sum_i c_i \operatorname{sigmoid} \left(b_i + \sum_j w_{ij} x_j \right) \quad \begin{array}{l} i: 1, 2, 3 \\ j: 1, 2, 3 \end{array}$$

$$r_1 = b_1 + w_{11}x_1 + w_{12}x_2 + w_{13}x_3$$

$$r_2 = b_2 + w_{21}x_1 + w_{22}x_2 + w_{23}x_3$$

$$r_3 = b_3 + w_{31}x_1 + w_{32}x_2 + w_{33}x_3$$

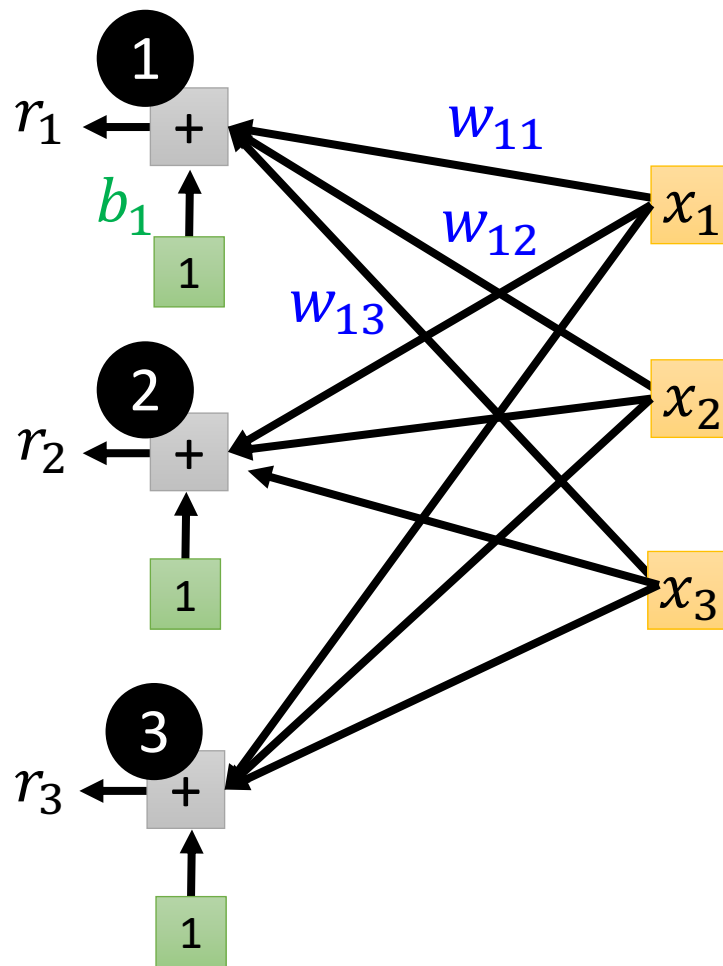
$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} + \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\mathbf{r} = \mathbf{b} + \mathbf{W} \mathbf{x}$$



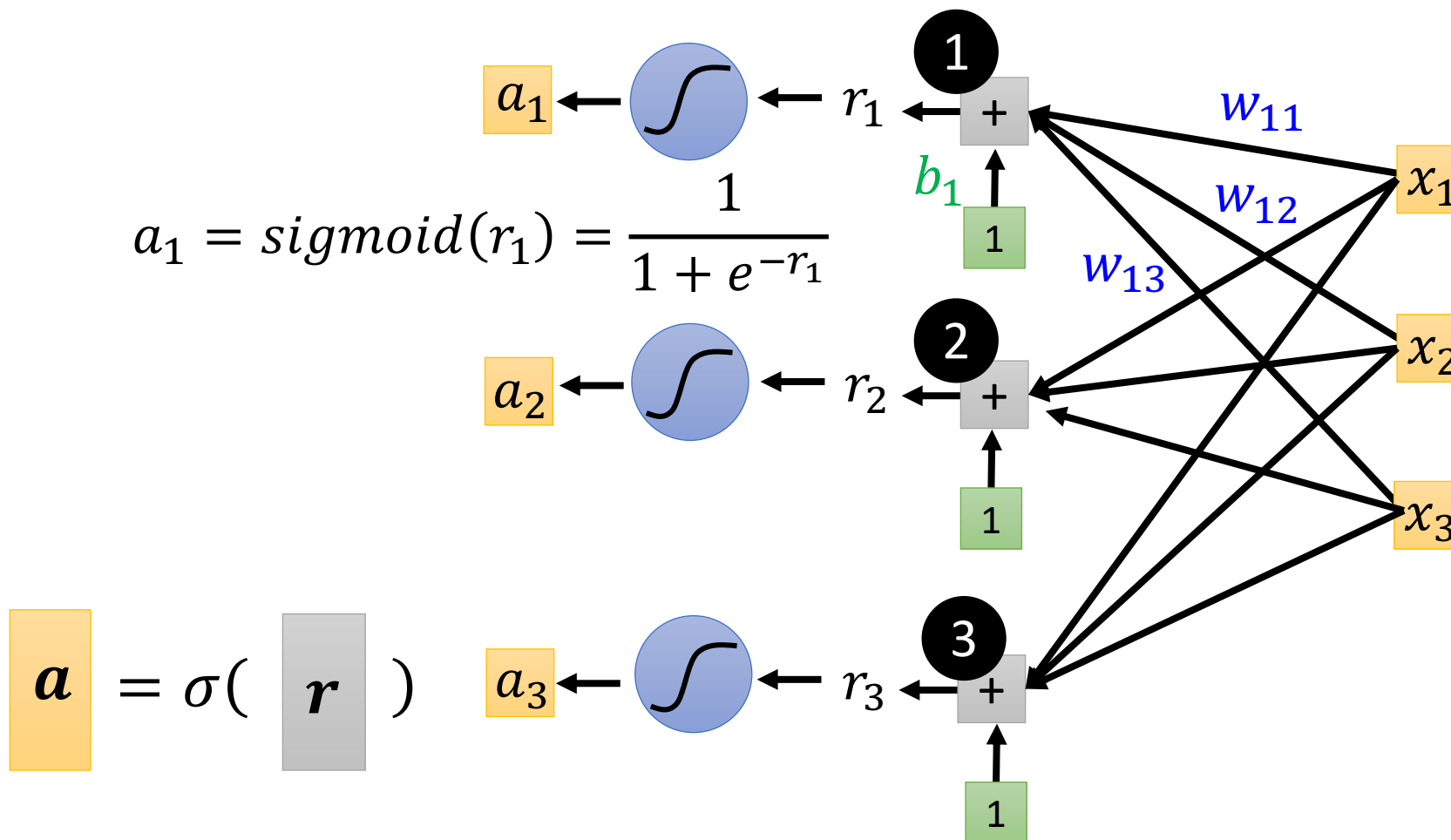
$$y = b + \sum_i c_i \operatorname{sigmoid} \left(b_i + \sum_j w_{ij} x_j \right) \quad \begin{array}{l} i: 1, 2, 3 \\ j: 1, 2, 3 \end{array}$$

$$\mathbf{r} = \mathbf{b} + \mathbf{W} \mathbf{x}$$



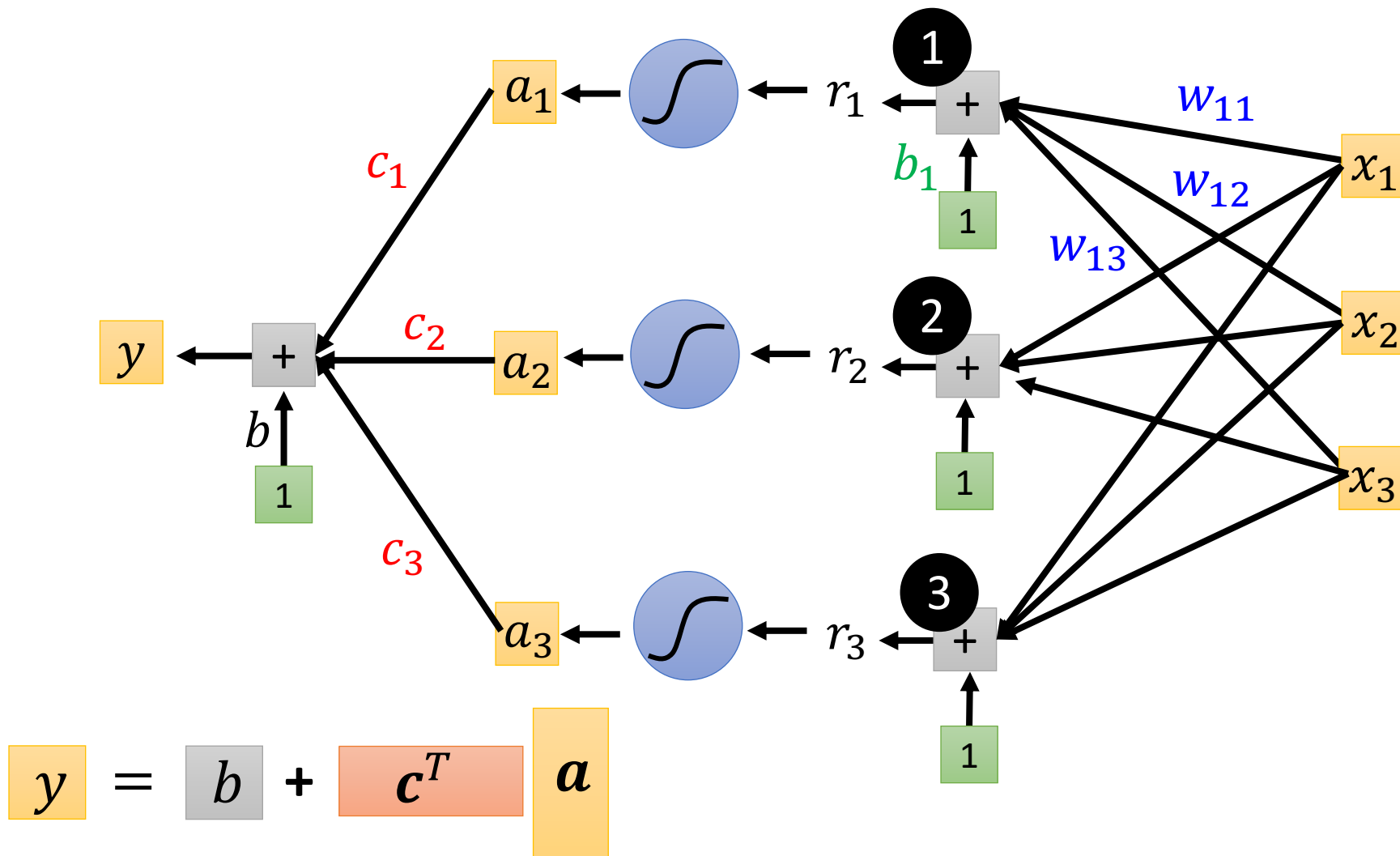


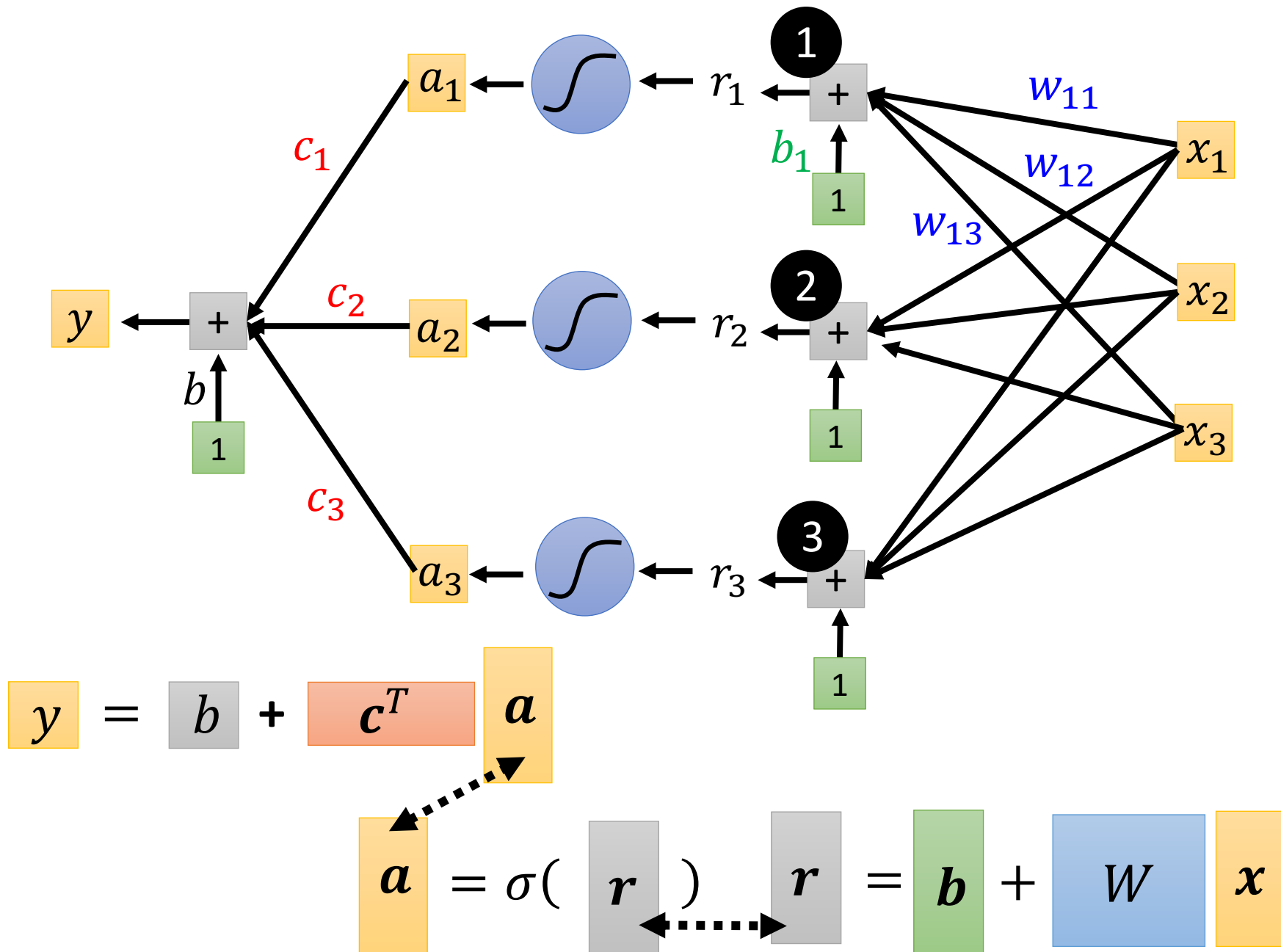
$$y = b + \sum_i c_i \text{sigmoid} \left(b_i + \sum_j w_{ij} x_j \right) \quad \begin{matrix} i: 1,2,3 \\ j: 1,2,3 \end{matrix}$$

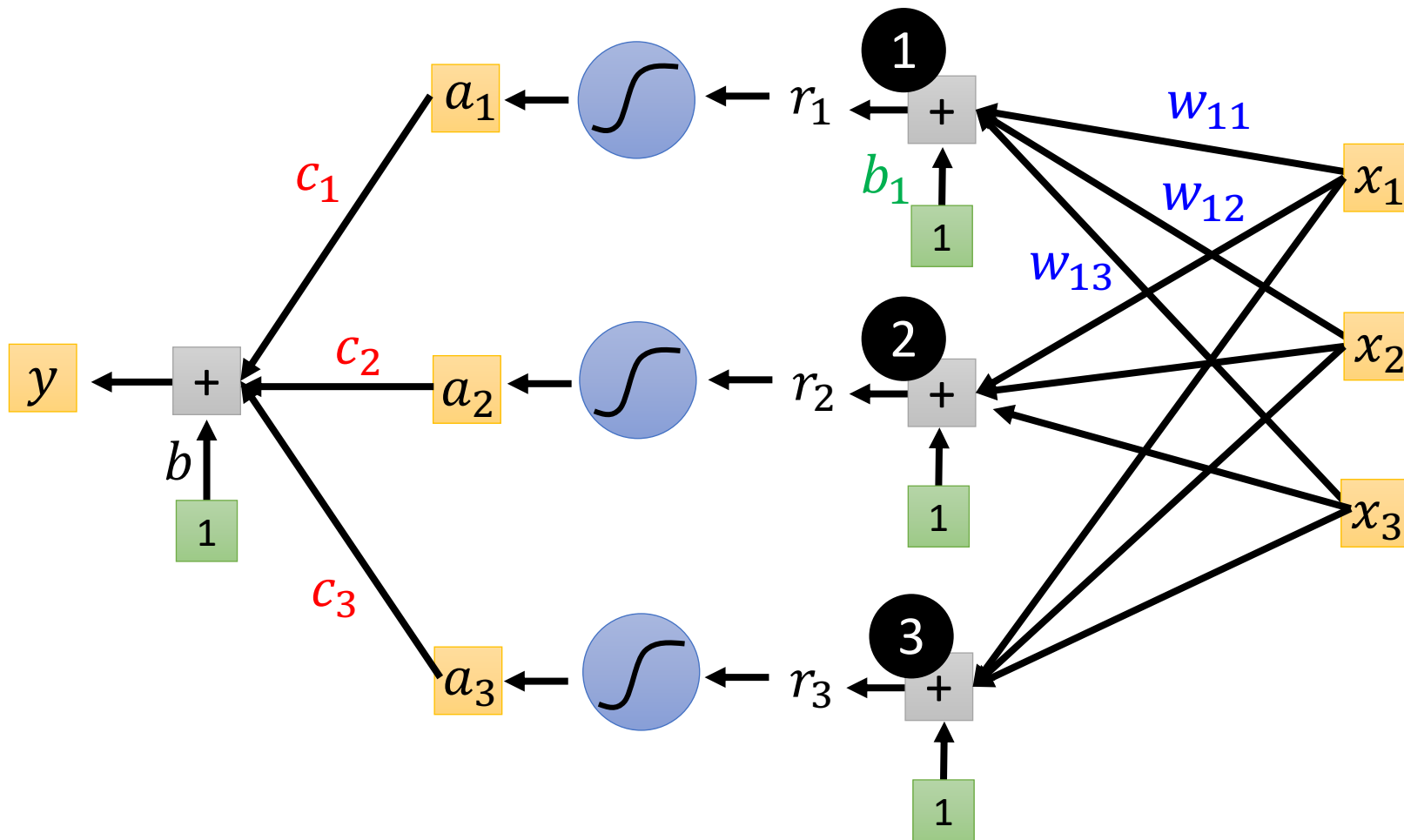




$$y = b + \sum_i c_i \operatorname{sigmoid} \left(b_i + \sum_j w_{ij} x_j \right) \quad \begin{array}{l} i: 1, 2, 3 \\ j: 1, 2, 3 \end{array}$$







$$y = b + c^T \sigma(b + Wx)$$



带有未知参数的函数

$$y = b + c^T \sigma(b + Wx)$$

x 特征

未知参数

W

b

c^T

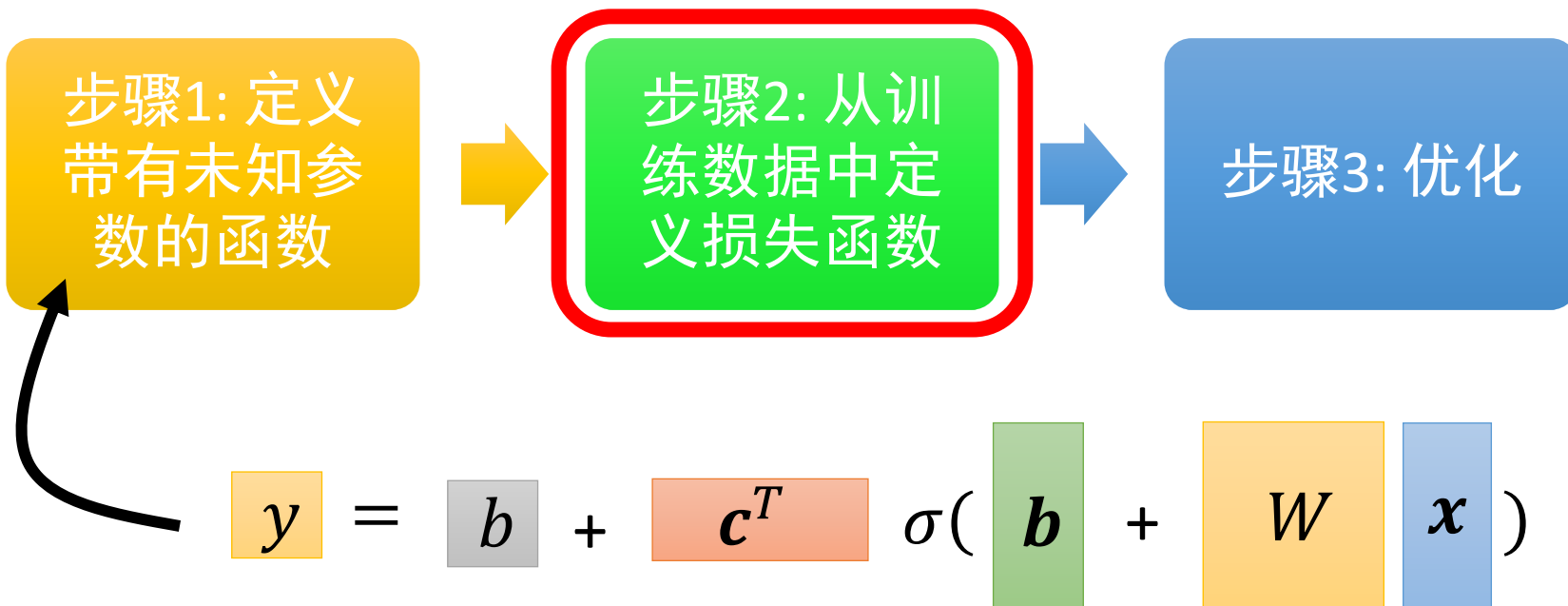
b

Rows of W

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \end{bmatrix}$$



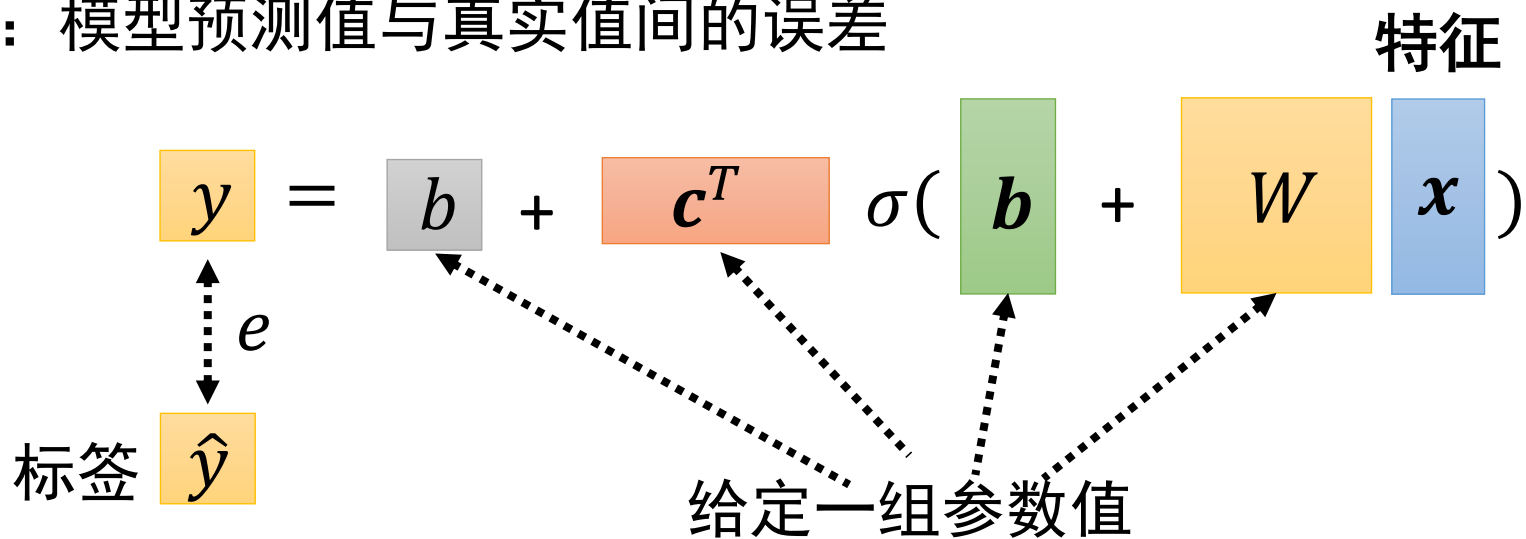
机器学习主要步骤





损失

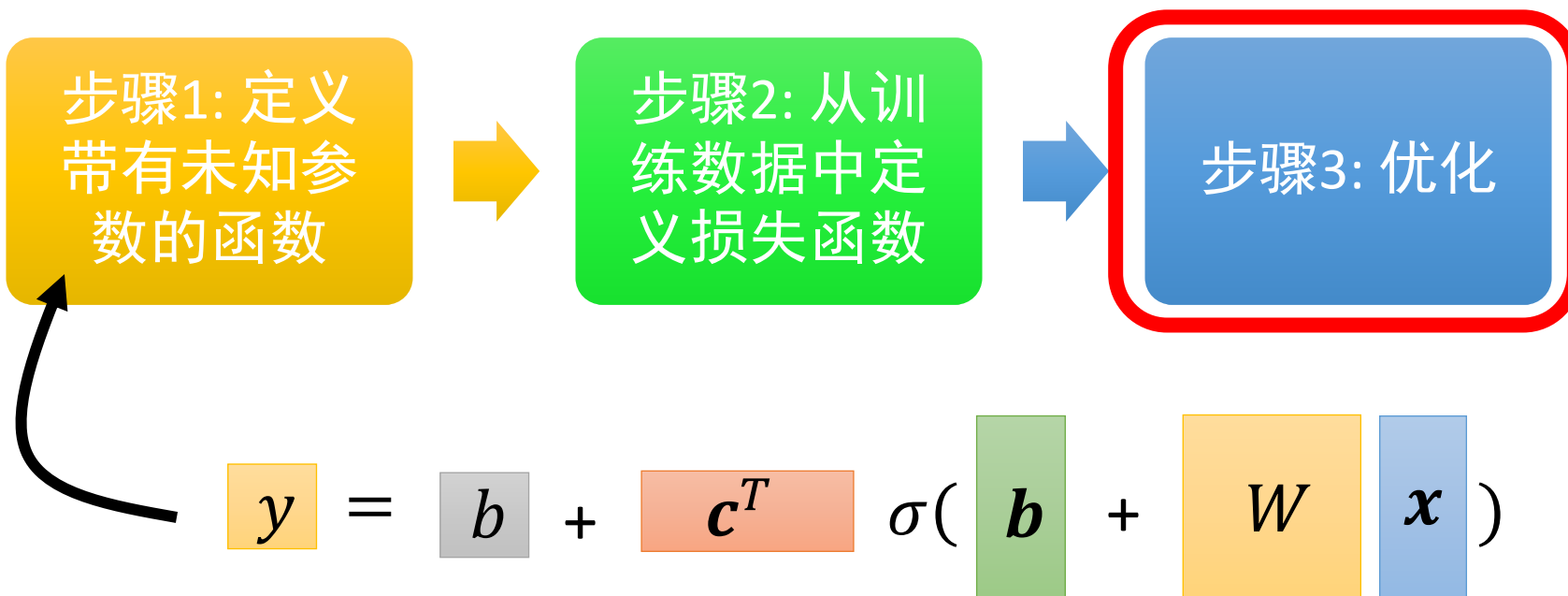
- 关于未知参数的损失函数 $L(\theta)$
- 损失：模型预测值与真实值间的误差



损失:
$$L = \frac{1}{N} \sum_n e_n$$



机器学习主要步骤





新模型的优化问题

$$\theta^* = \arg \min_{\theta} L$$

$$\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \vdots \end{bmatrix}$$

➤ (随机) 初始化参数 θ^0

$$\text{梯度 } g = \begin{bmatrix} \frac{\partial L}{\partial \theta_1} |_{\theta=\theta^0} \\ \frac{\partial L}{\partial \theta_2} |_{\theta=\theta^0} \\ \vdots \end{bmatrix}$$

$$g = \nabla L(\theta^0)$$

$$\begin{bmatrix} \theta_1^1 \\ \theta_2^1 \\ \vdots \end{bmatrix} \leftarrow \begin{bmatrix} \theta_1^0 \\ \theta_2^0 \\ \vdots \end{bmatrix} - \begin{bmatrix} \eta \frac{\partial L}{\partial \theta_1} |_{\theta=\theta^0} \\ \eta \frac{\partial L}{\partial \theta_2} |_{\theta=\theta^0} \\ \vdots \end{bmatrix}$$

$$\theta^1 \leftarrow \theta^0 - \eta g$$



新模型的优化问题

$$\theta^* = \arg \min_{\theta} L$$

➤ (随机) 初始化参数 θ^0

➤ 计算梯度 $g = \nabla L(\theta^0)$

$$\theta^1 \leftarrow \theta^0 - \eta g$$

➤ 计算梯度 $g = \nabla L(\theta^1)$

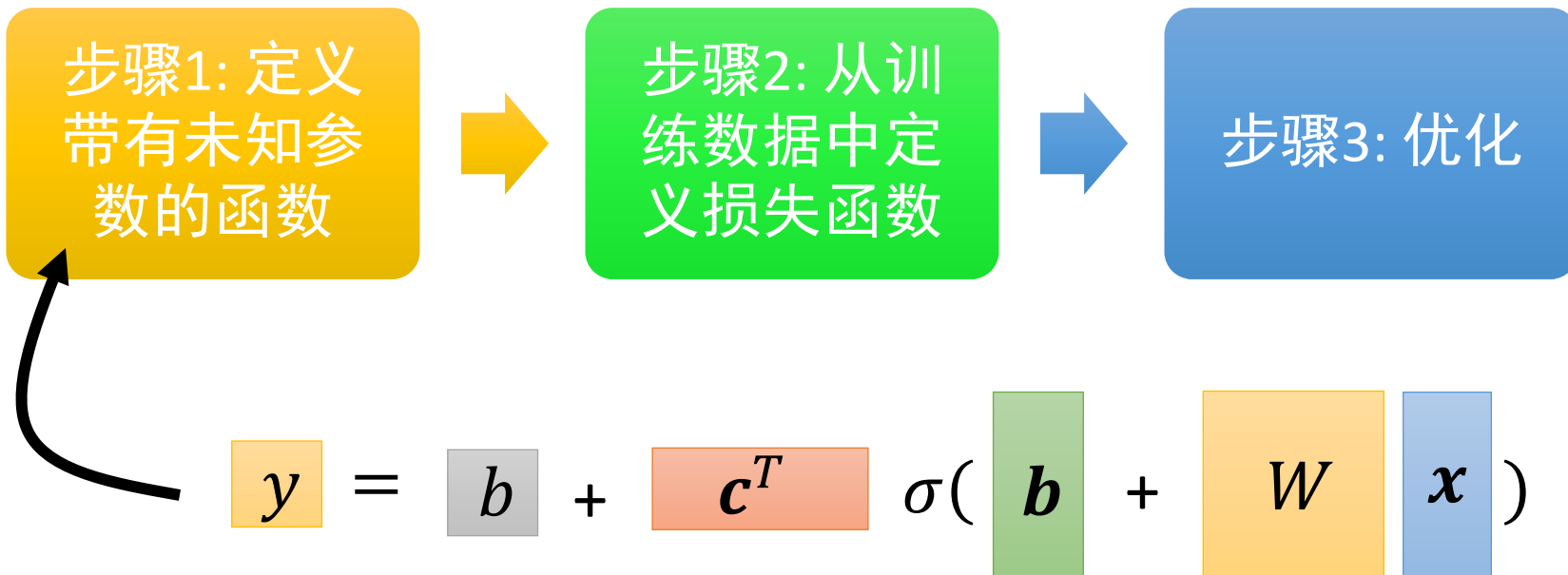
$$\theta^2 \leftarrow \theta^1 - \eta g$$

➤ 计算梯度 $g = \nabla L(\theta^2)$

$$\theta^3 \leftarrow \theta^2 - \eta g$$



机器学习主要步骤



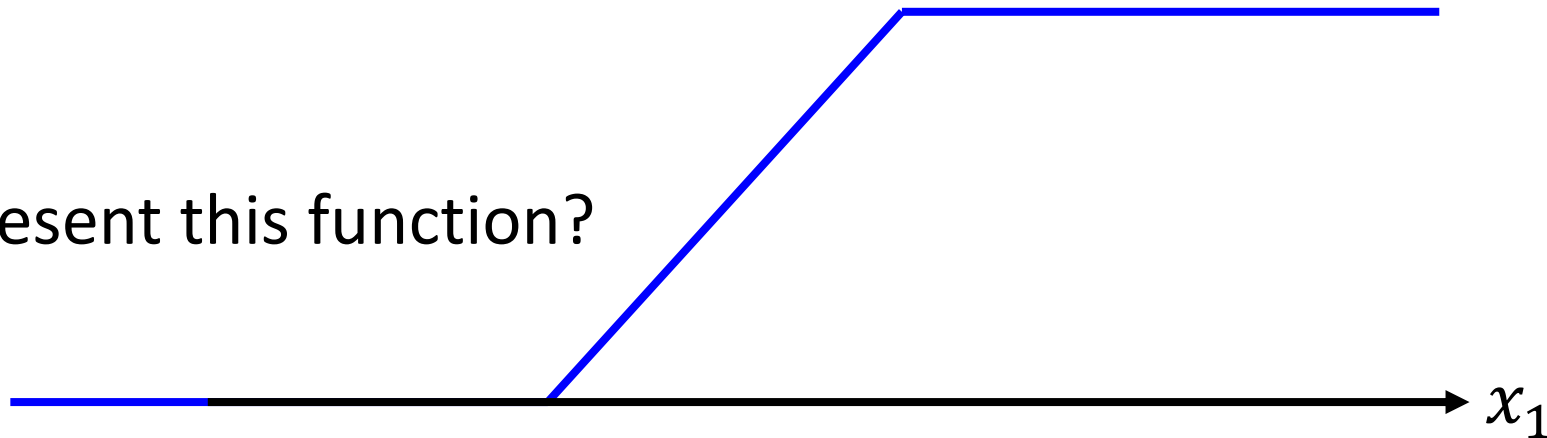
还可以对模型有更多的变形。比如可以修改激活函数



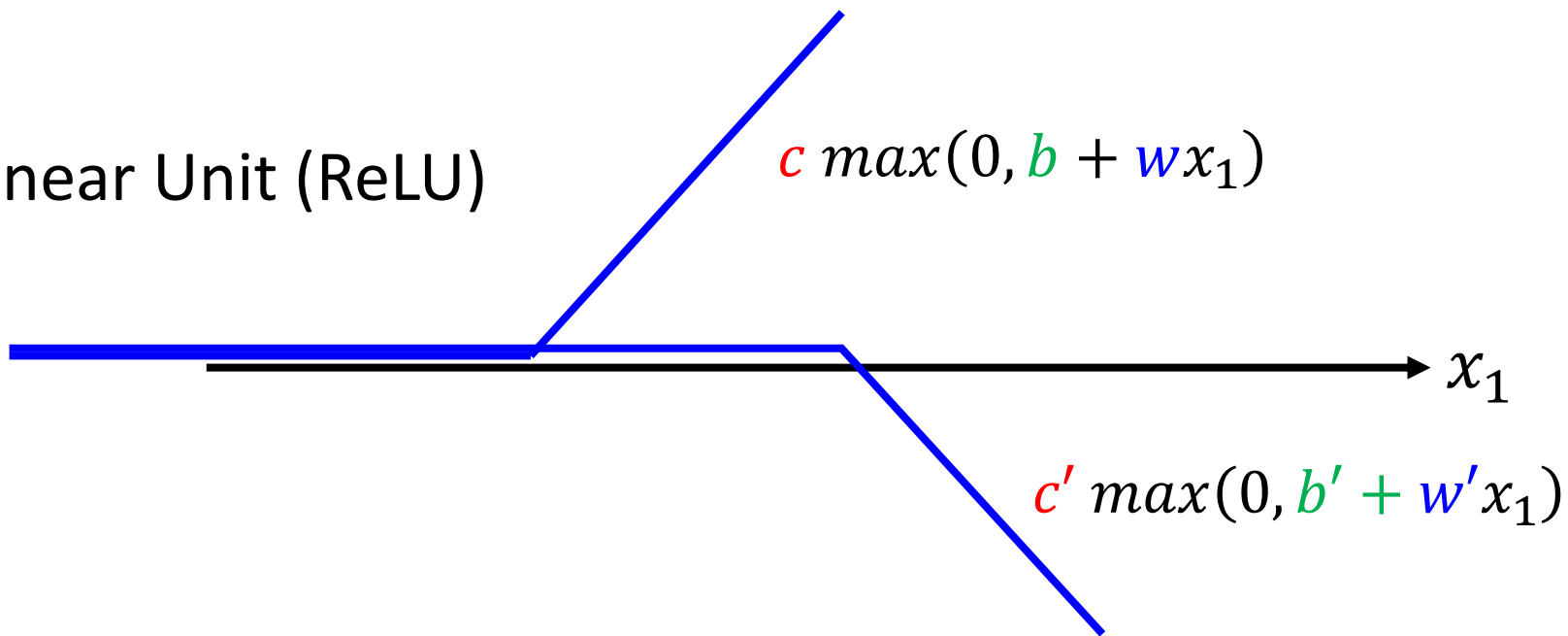
北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

Sigmoid \rightarrow ReLU

How to represent this function?



Rectified Linear Unit (ReLU)





Sigmoid \rightarrow ReLU

$$y = b + \sum_i c_i \text{sigmoid} \left(b_i + \sum_j w_{ij} x_j \right)$$

激活函数 Activation function

$$y = b + \sum_{2i} c_i \max \left(0, b_i + \sum_j w_{ij} x_j \right)$$

Which one is better?



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

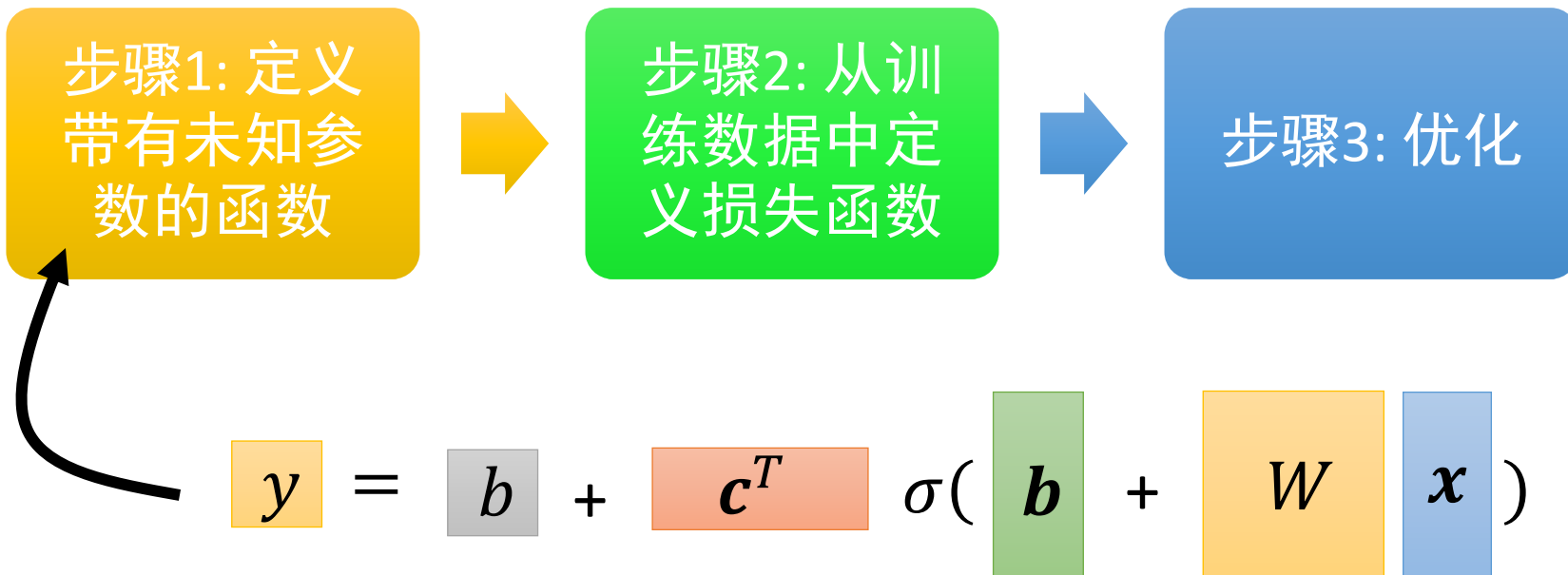
实验结果

$$y = b + \sum_{\textcolor{red}{i}} \textcolor{red}{c}_i \max \left(0, \textcolor{green}{b}_i + \sum_j \textcolor{blue}{w}_{ij} x_j \right)$$

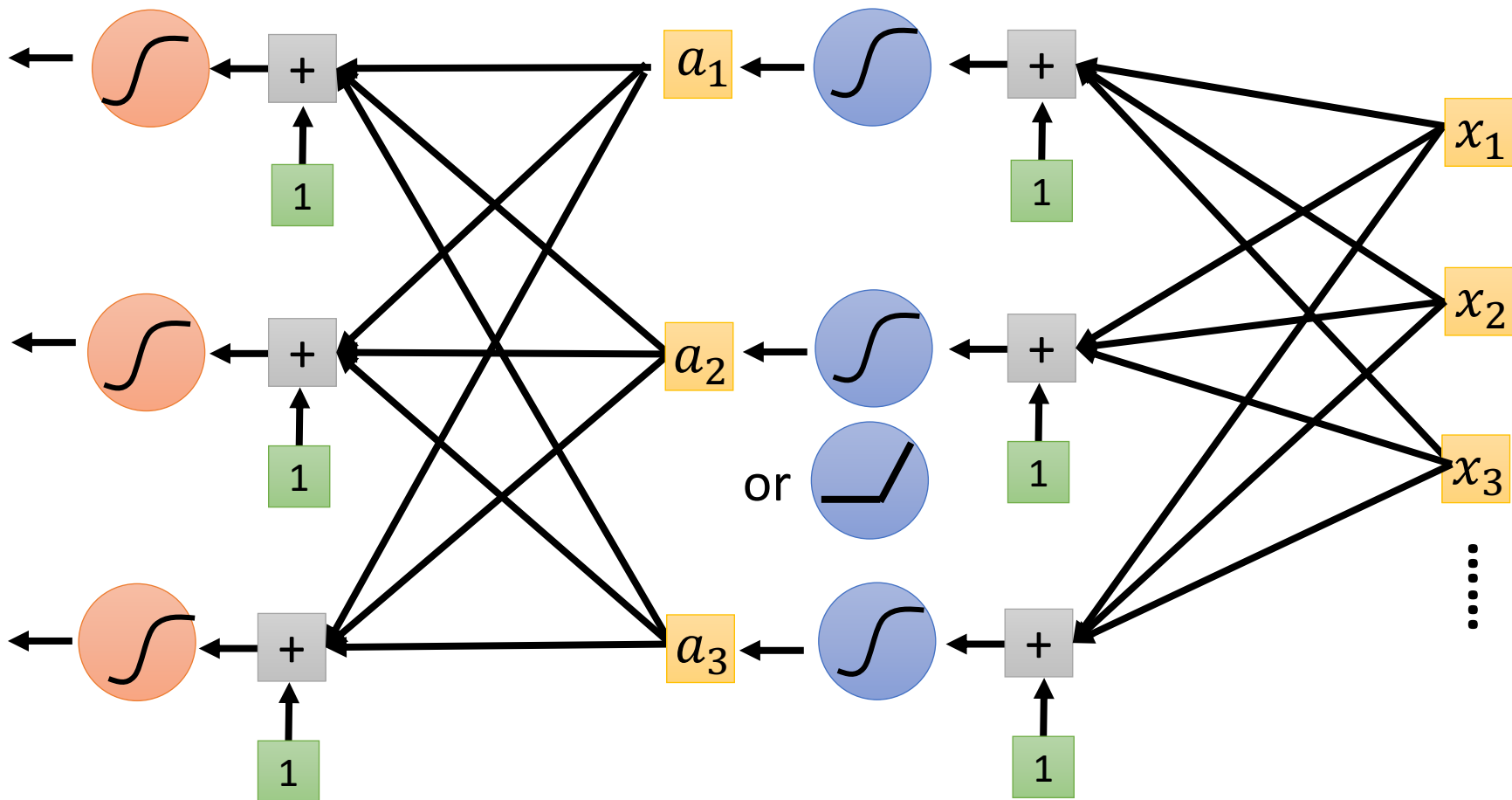
	linear
2017 – 2020	0.32k
2021	0.46k



机器学习主要步骤



还可以对模型有更多的变形。比如可以增加模型层数



$$\mathbf{a}' = \sigma(\mathbf{b}' + \mathbf{W}' \mathbf{a}) \quad \mathbf{a} = \sigma(\mathbf{b} + \mathbf{W} \mathbf{x})$$



实验结果

- 多个隐含层模型的损失
 - 每层100个 ReLU 神经元
 - 输入特征是过去56天的观看量

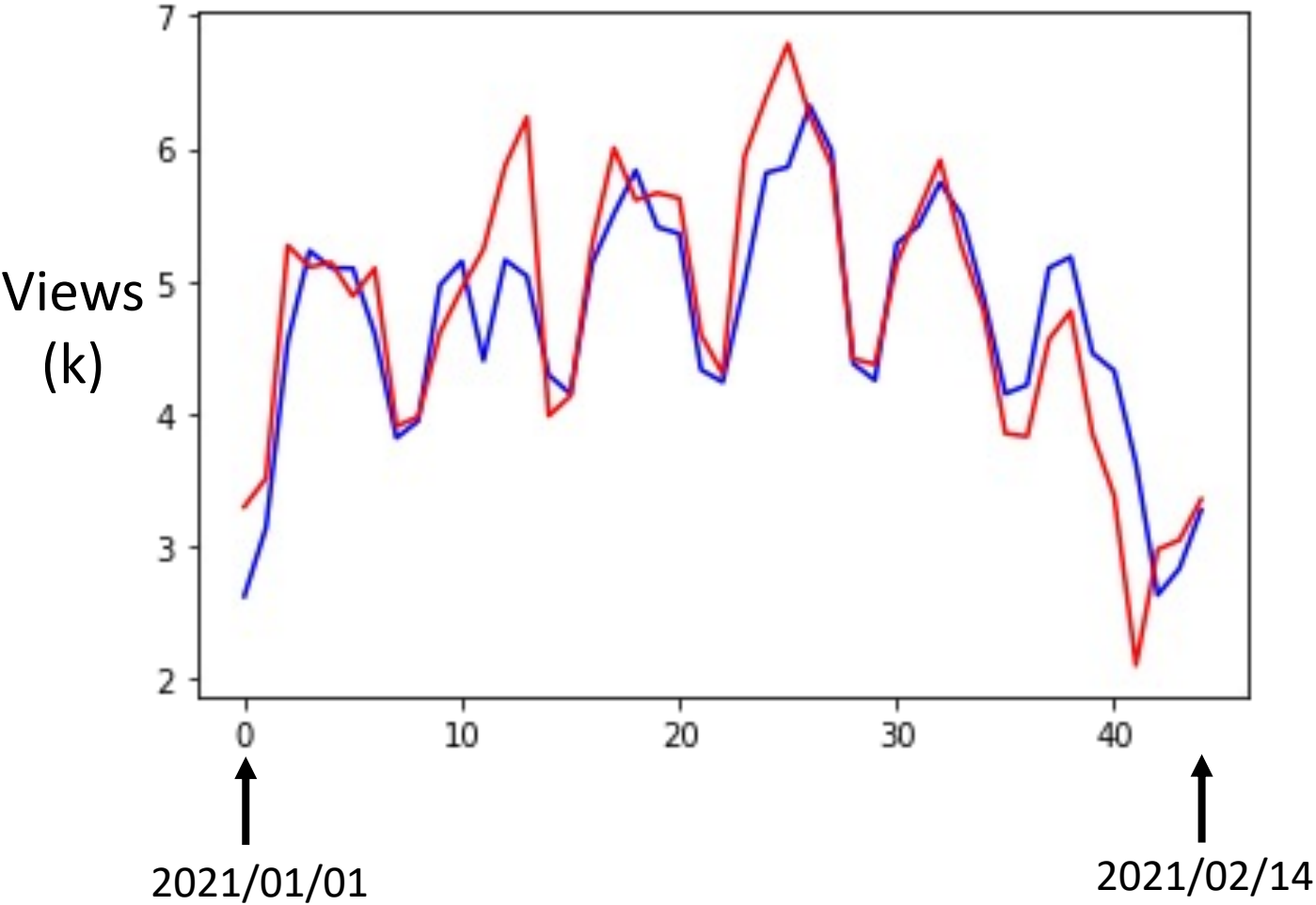
	1 layer
2017 – 2020	0.28k
2021	0.43k



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

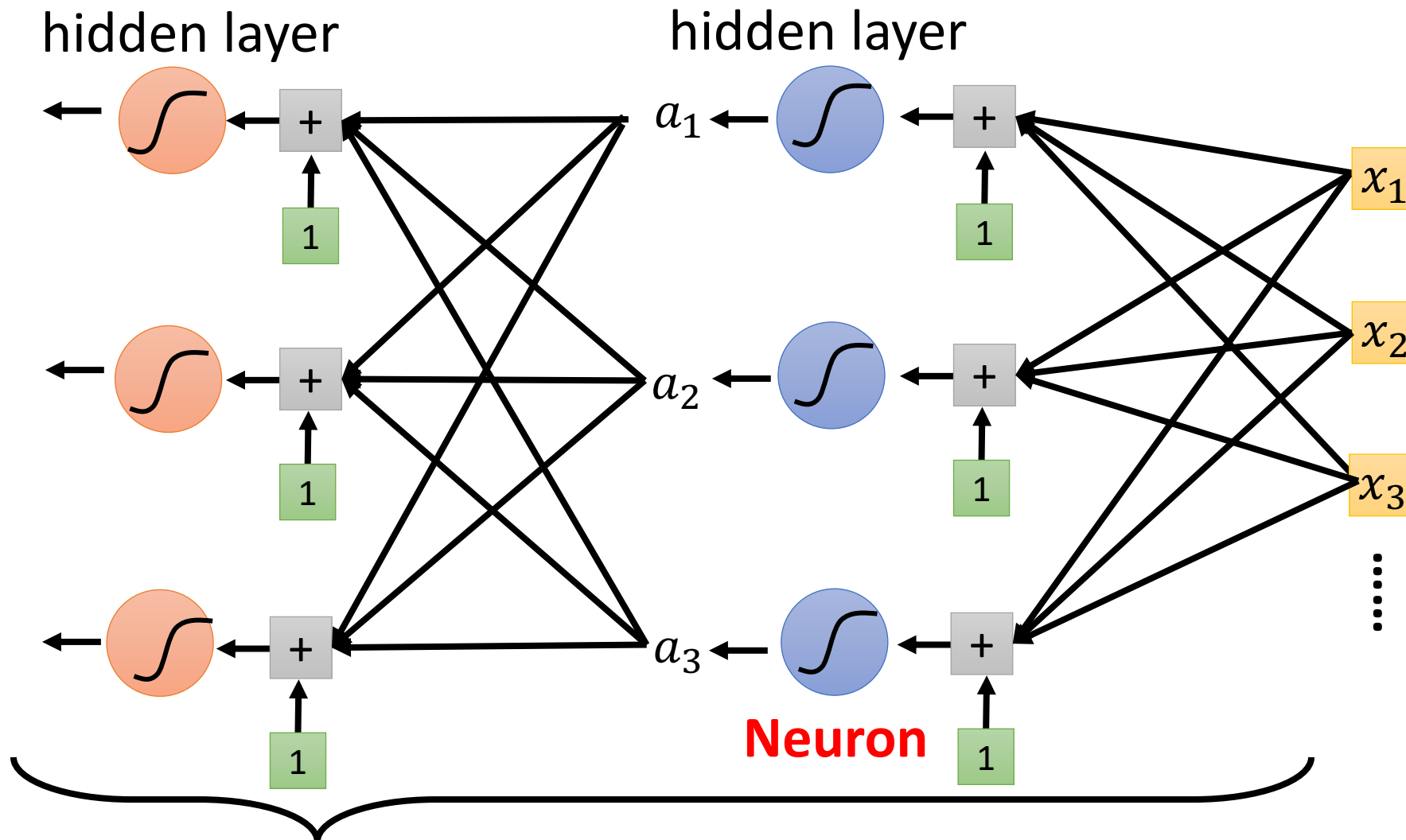
3 layers

Red: real no. of views
blue: estimated no. of views





北京航空航天大学
COLLEGE OF SOFTWARE BEIHANG UNIVERSITY 软件学院



神经网络 Neural Network

Many layers means **Deep** ➡ 深度学习 Deep Learning



前馈神经网络

- 输入层、输出层和至少一层的隐藏层构成。网络中各个隐藏层中神经元可接收相邻前序隐藏层中所有神经元传递而来的信息，经过加工处理后将信息输出给相邻后续隐藏层中所有神经元。
- 各个神经元接受前一级的输入，并输出到下一级，模型中没有反馈
- 层与层之间通过“全连接”进行链接，即两个相邻层之间的神经元完全成对连接，但层内的神经元不相互连接。
- 也被称为全连接网络，或多层感知机。



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

提纲

一、线性回归与梯度下降

二、前馈神经网络

三、卷积神经网络

四、序列数据模型

五、深度学习应用



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

Convolutional Neural Networks

卷积神经网络

Network architecture designed for image



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

图像分类



100 x 100



$$\begin{bmatrix} \vdots \\ 0.2 \\ 0.7 \\ 0.1 \\ \vdots \end{bmatrix}$$

y'

$$\begin{matrix} \text{dog} & \begin{bmatrix} \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix} \\ \text{cat} & \\ \text{tree} & \end{matrix}$$

\hat{y}

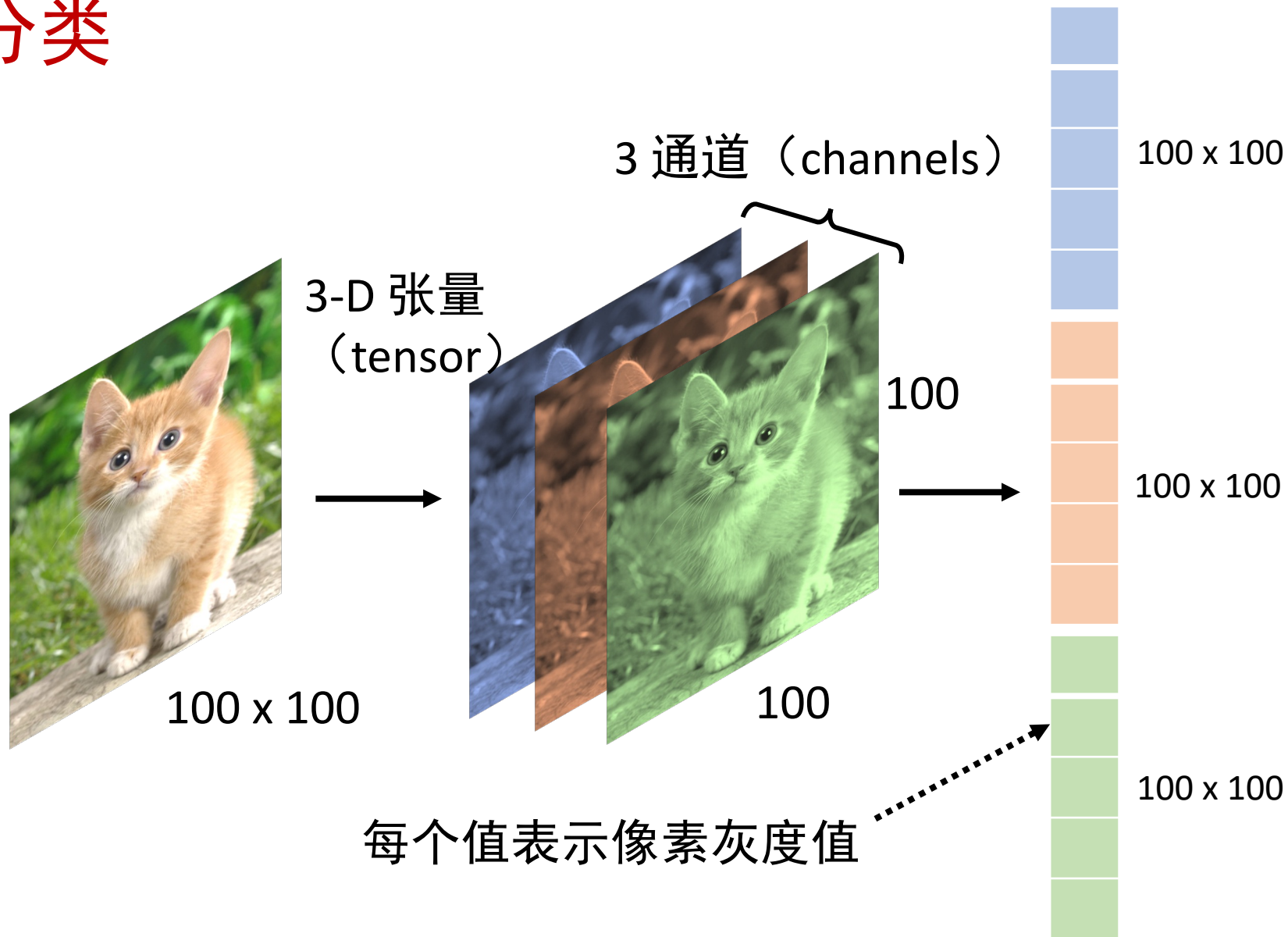
交叉熵损失

(待分类图像具有相同尺寸)



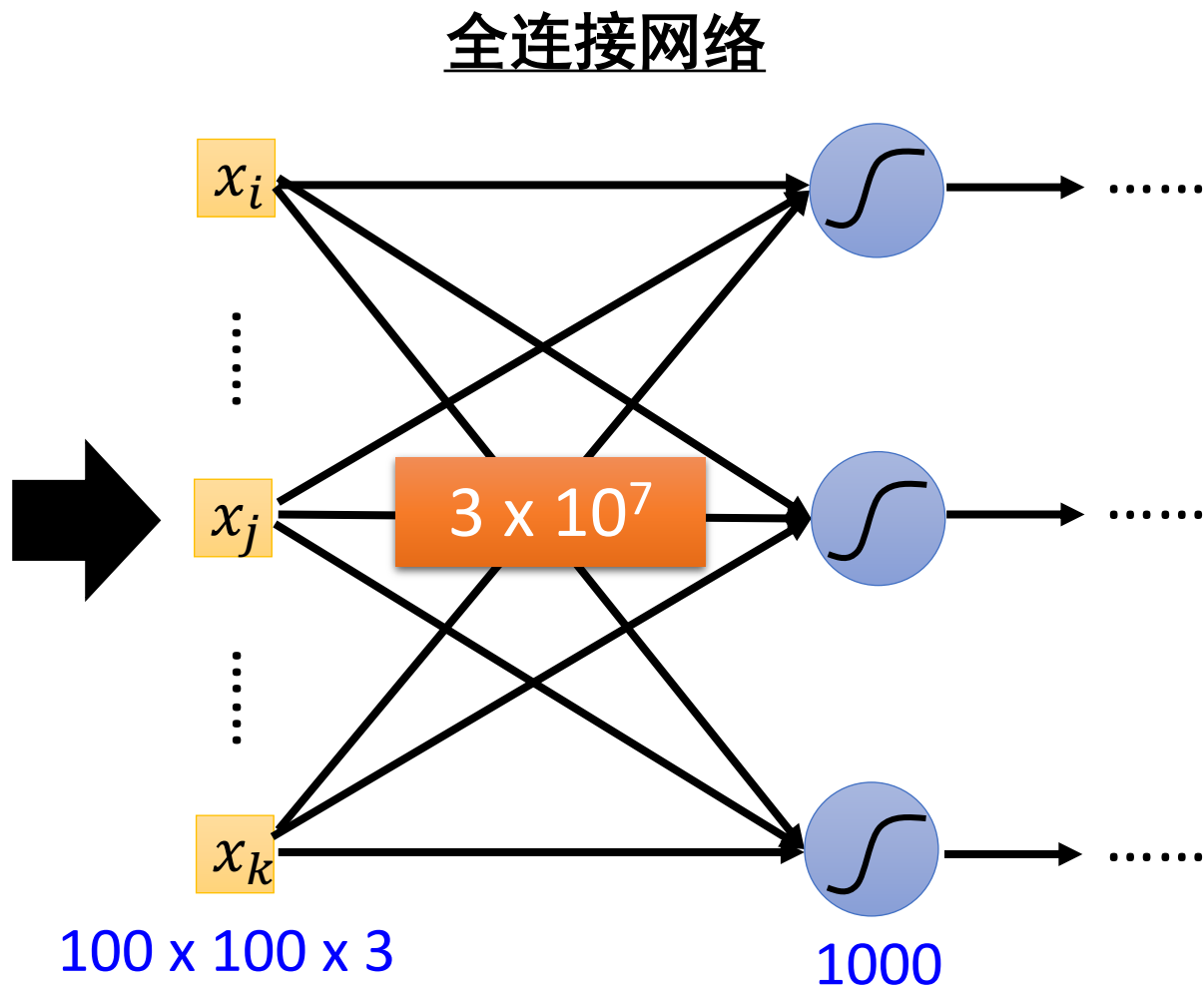
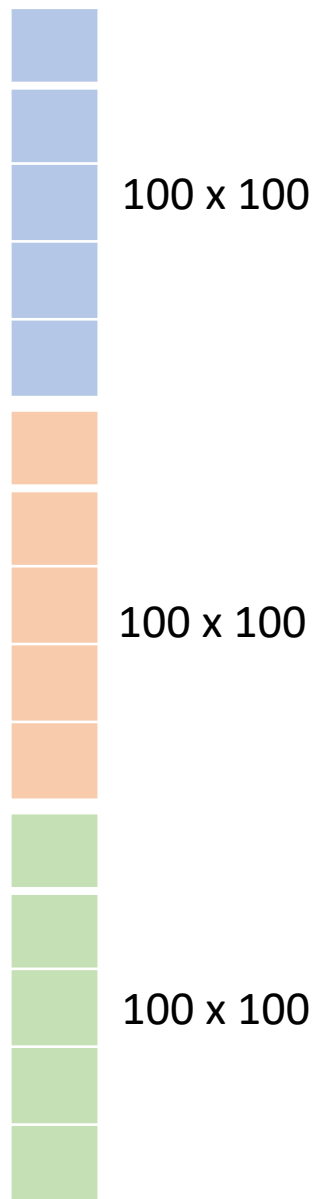
北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

图像分类





北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院



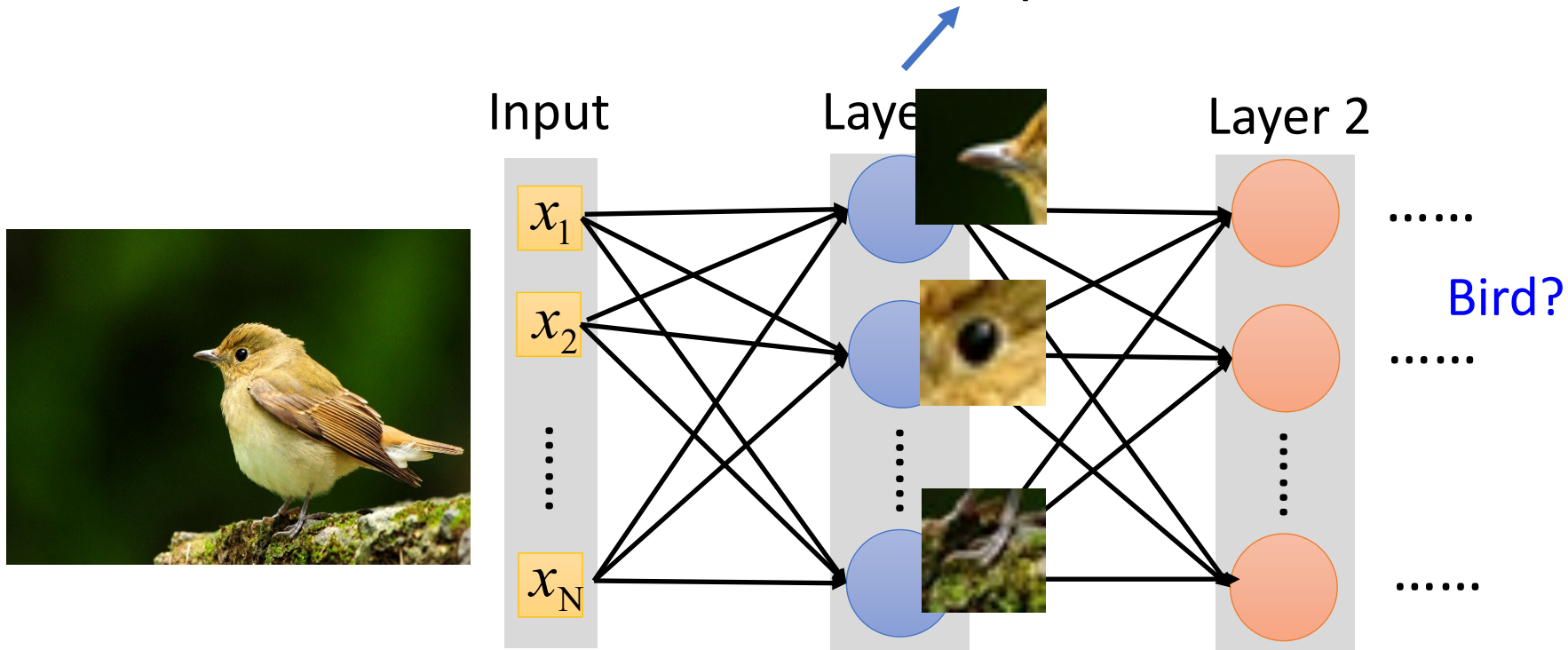
是否需要全连接网络来处理图像？



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

观察 1

根据特定的模式（pattern）进行识别



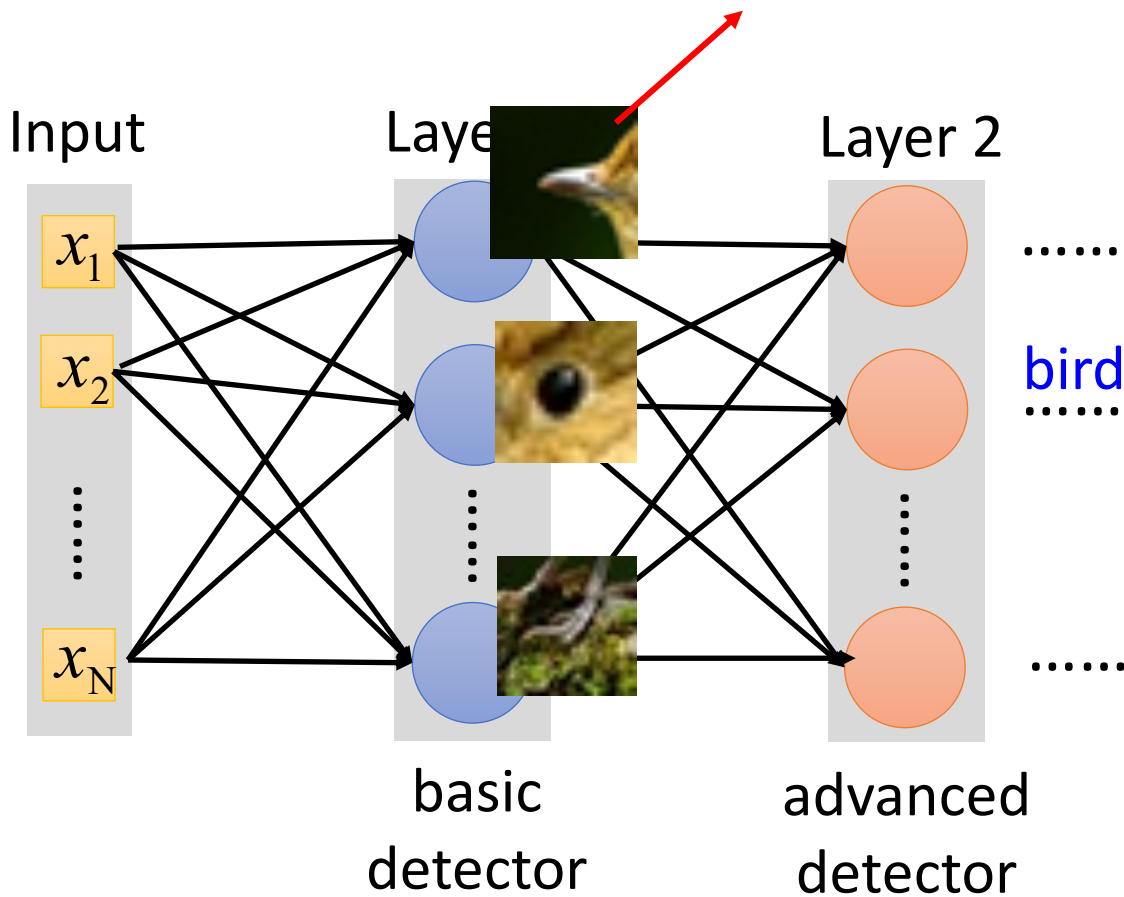
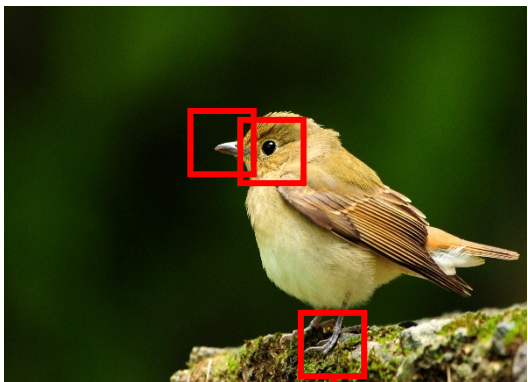
人也许用的是同样的方式实现鸟的识别... 😊



观察 1

每个神经元无需覆盖整个图像.

是否有必要看全图?



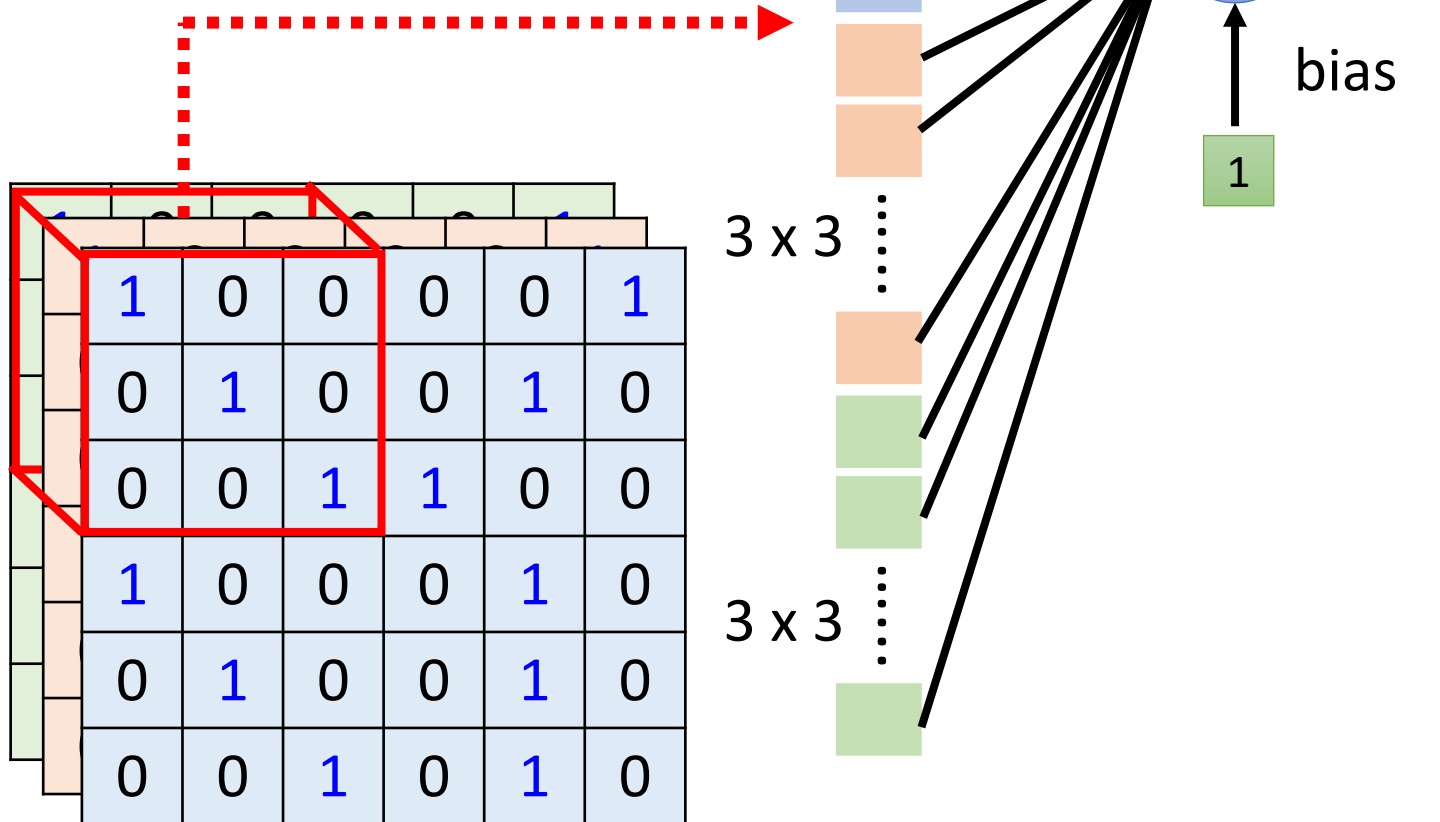
很多patterns只占据图像的一个小区域.



北京航空航天大学
COLLEGE OF SOFTWARE
BEIHANG UNIVERSITY 软件学院

简化 1

Receptive field
感受野

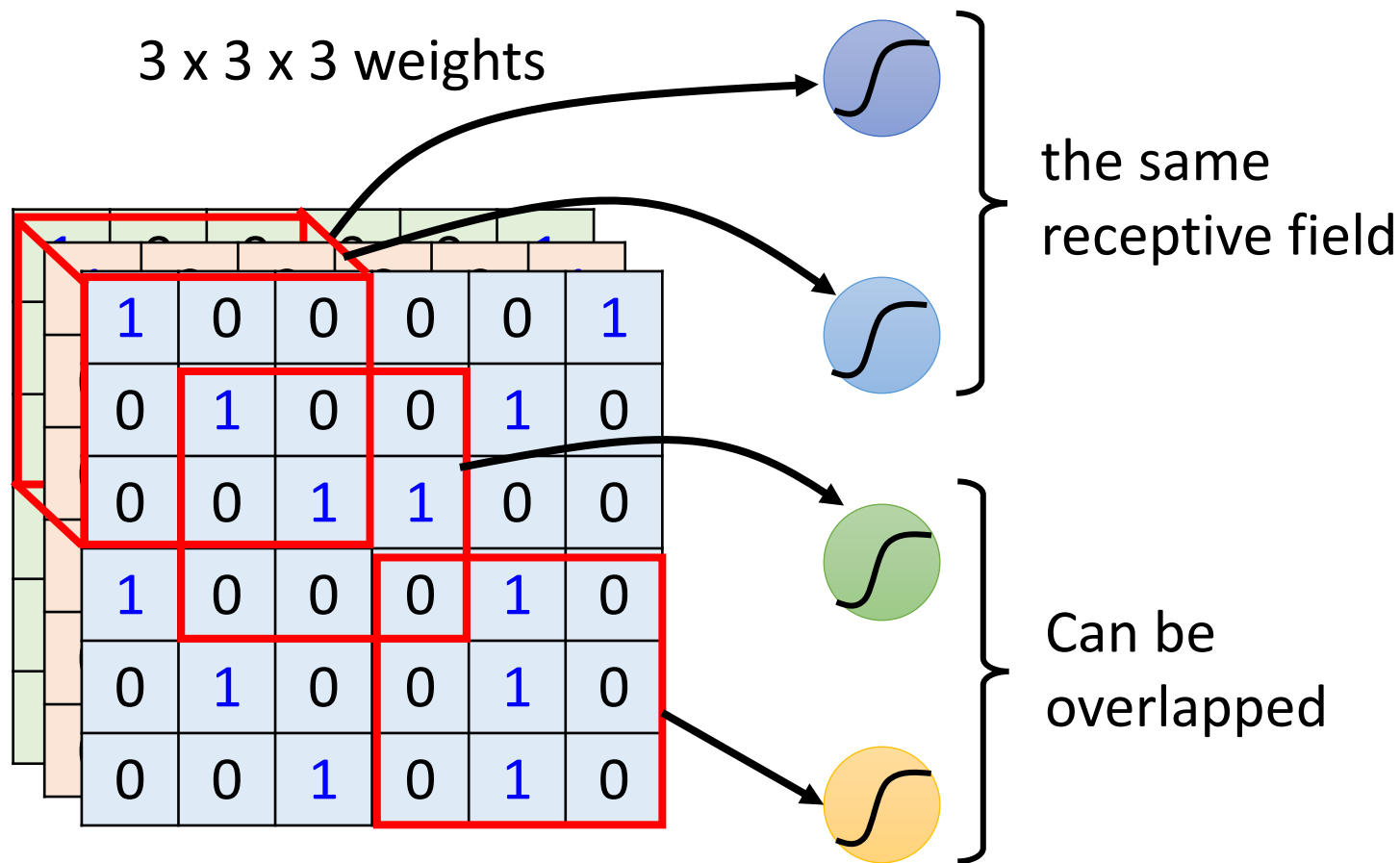




简化 1

- 不同神经元覆盖不同感受野大小?
- 只覆盖部分通道?
- 非正方形感受野?

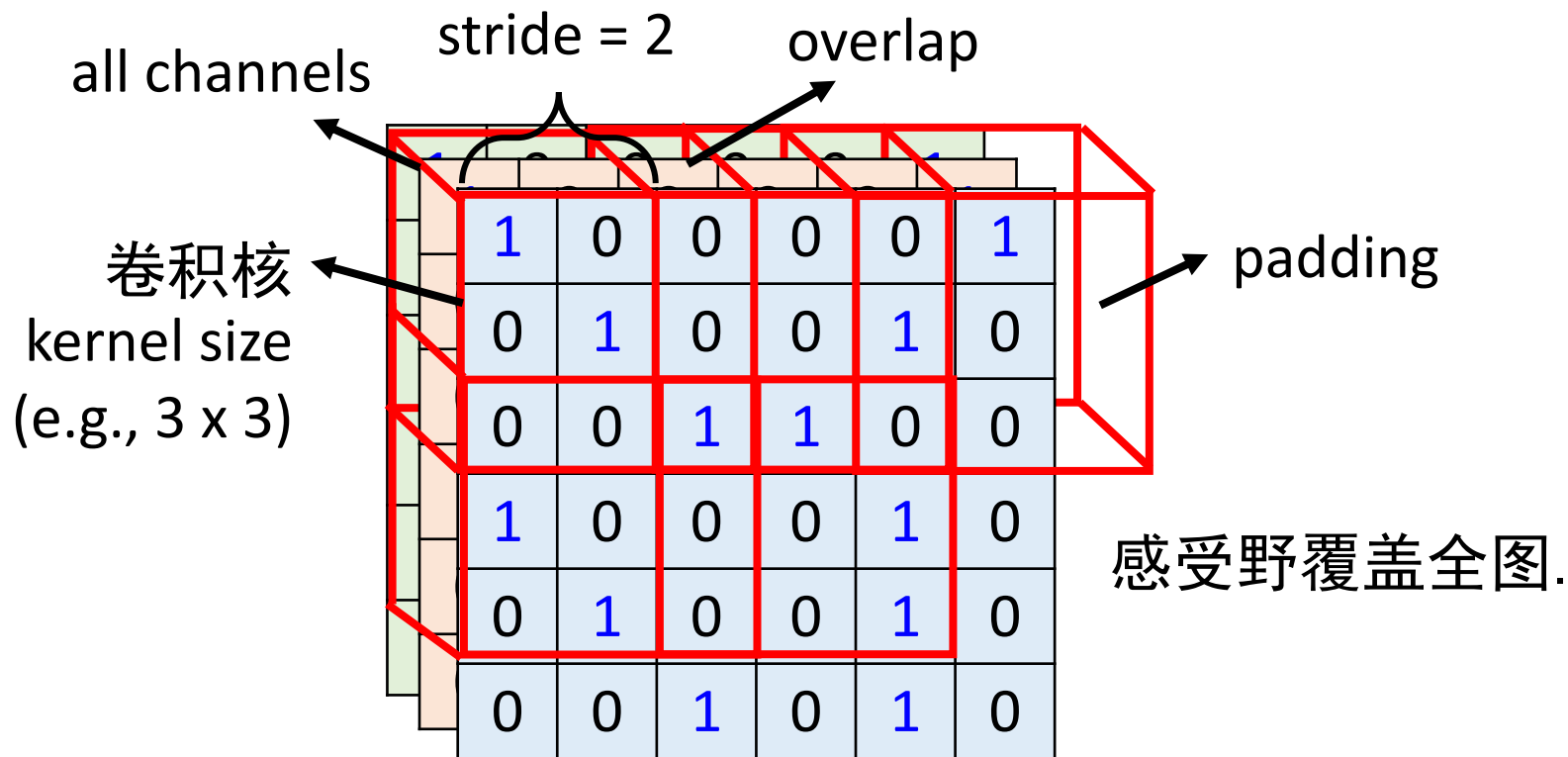
Receptive field
感受野





简化 1 - 典型设置

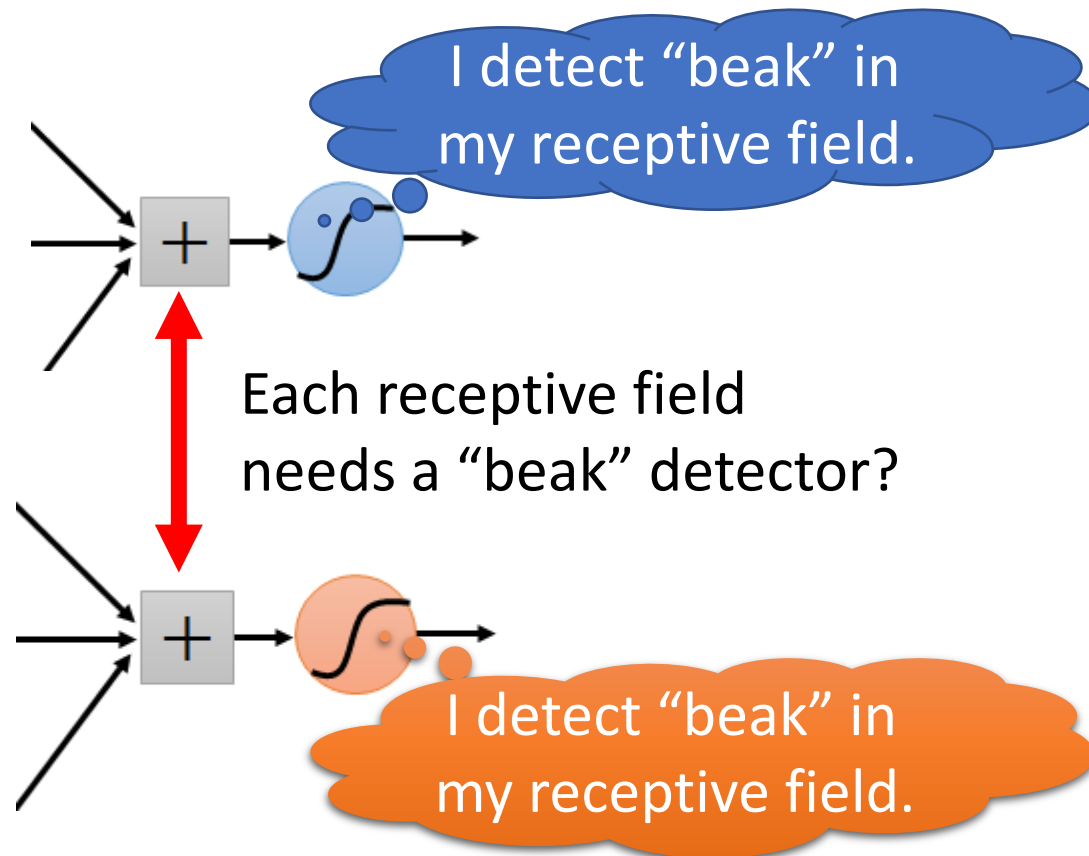
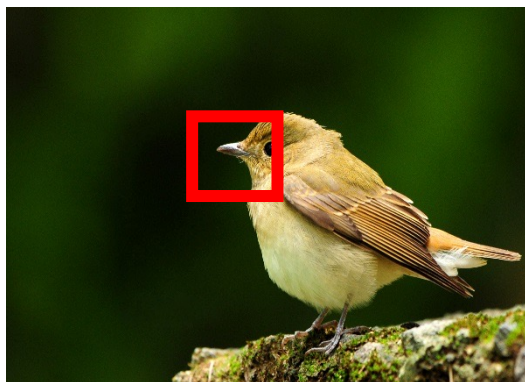
每个感受野覆盖一组神经元 (e.g., 64 neurons).





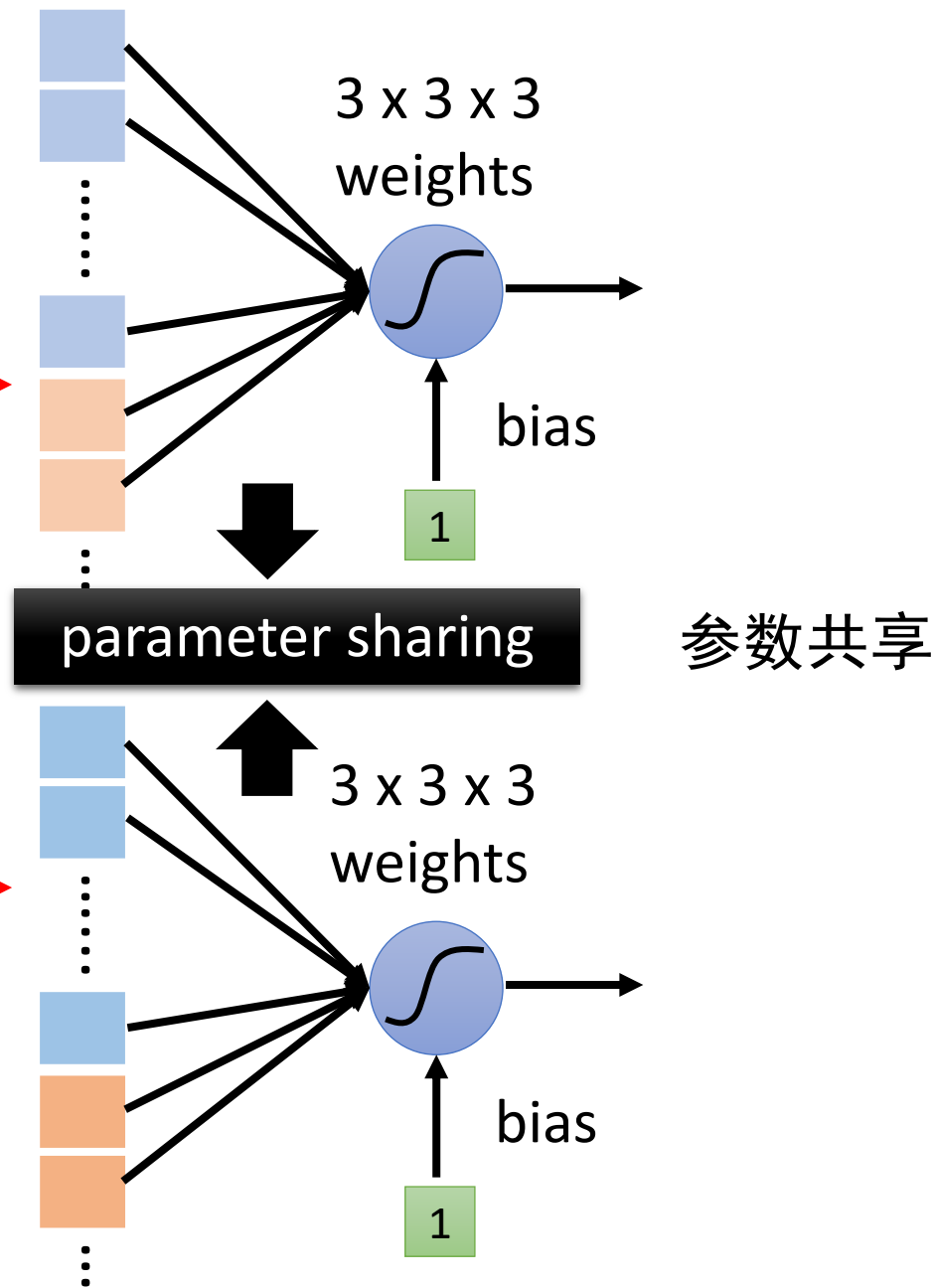
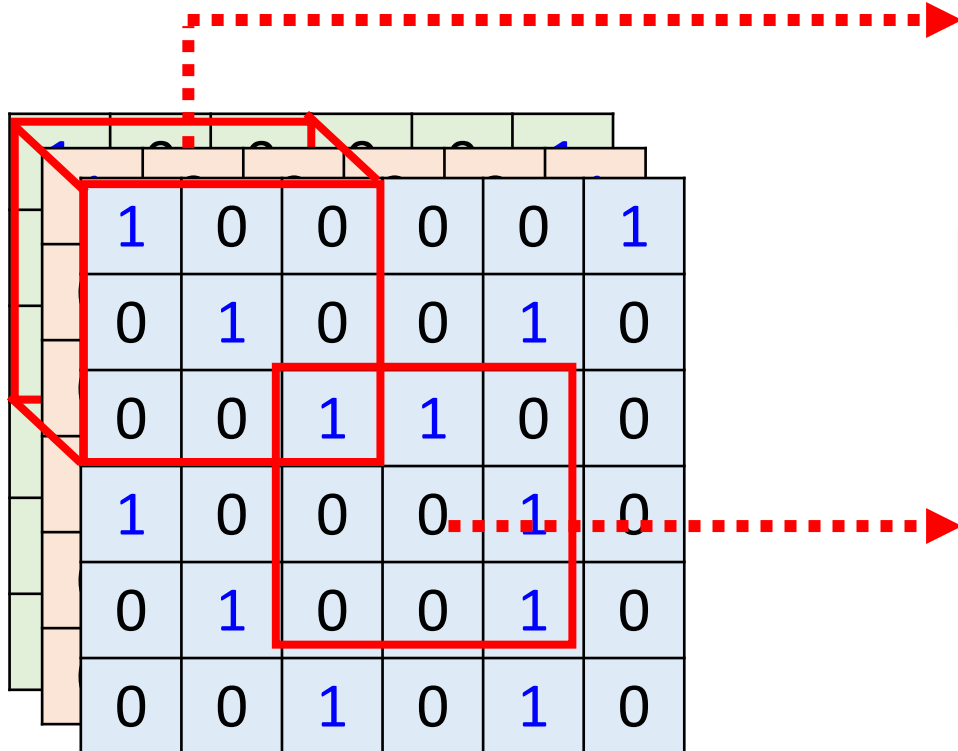
观察 2

- 同一pattern可能出现在不同图像的不同区域



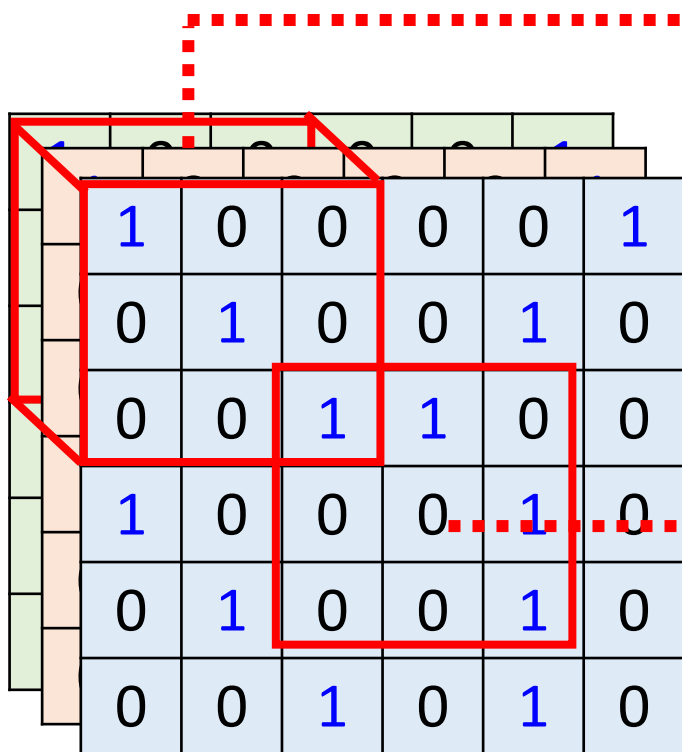


简化 2

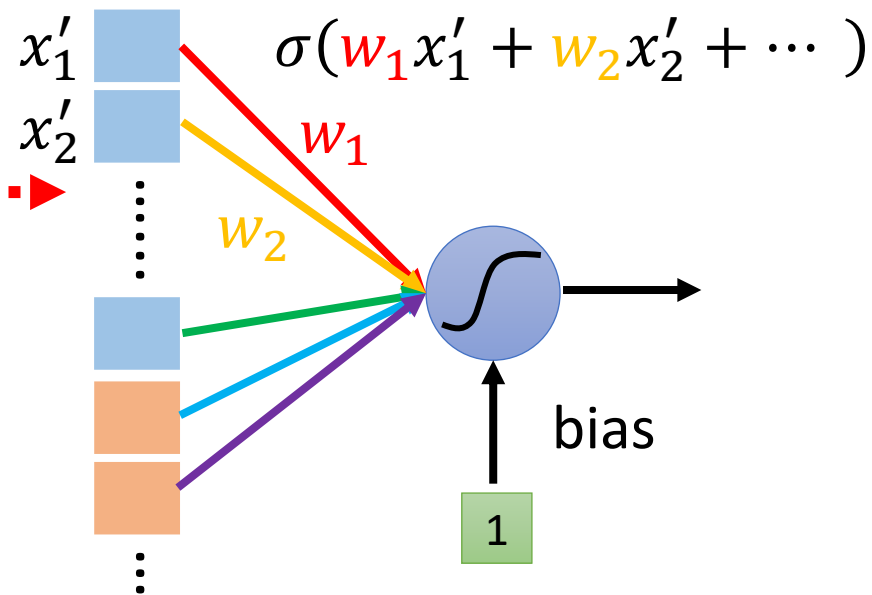
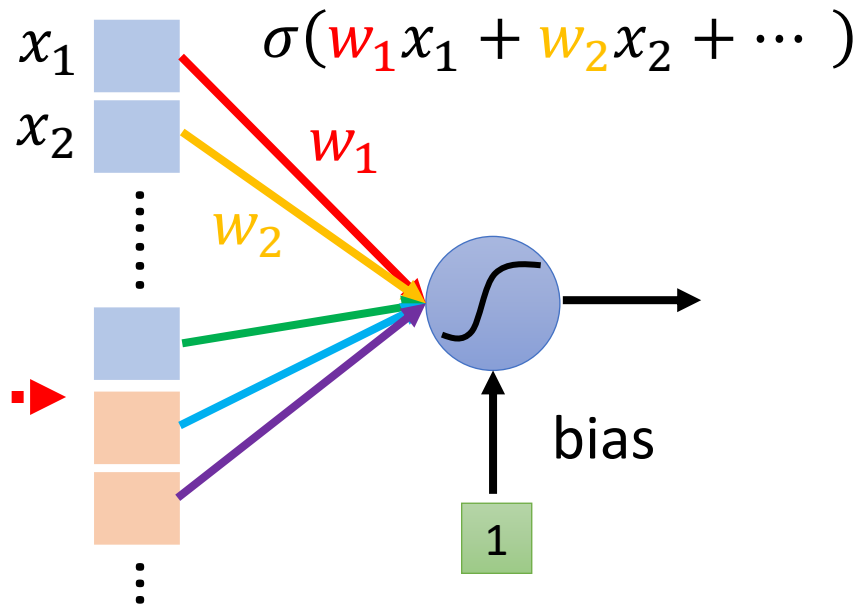




简化 2



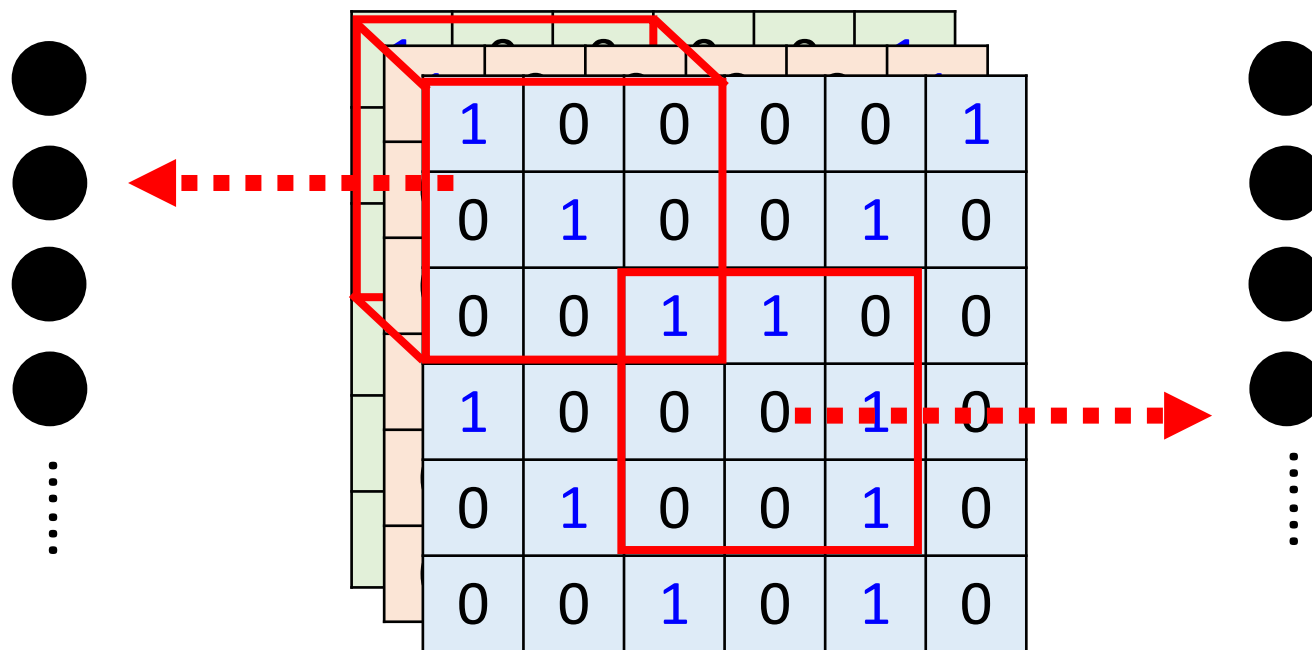
同一感受野的不同神经元不共享参数





简化 2 - 典型设置

每个感受野有多个神经元 (e.g., 64 个神经元).

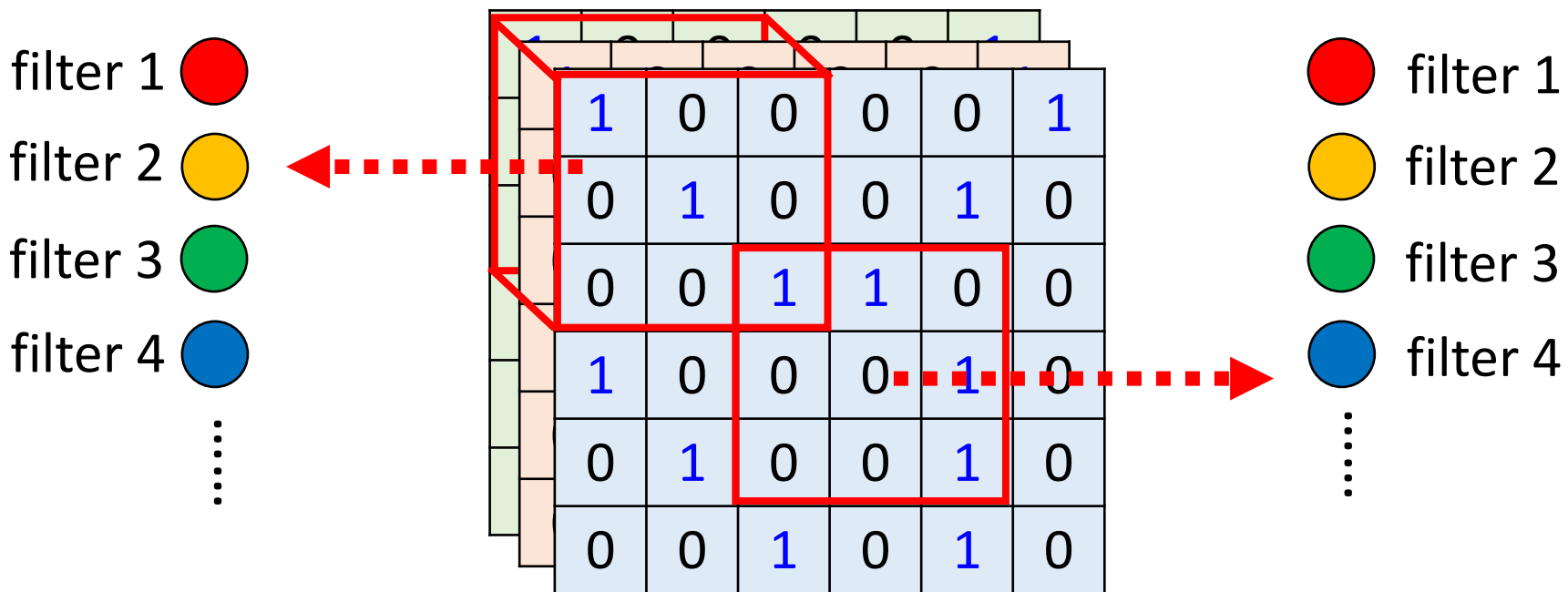




简化 2 - 典型设置

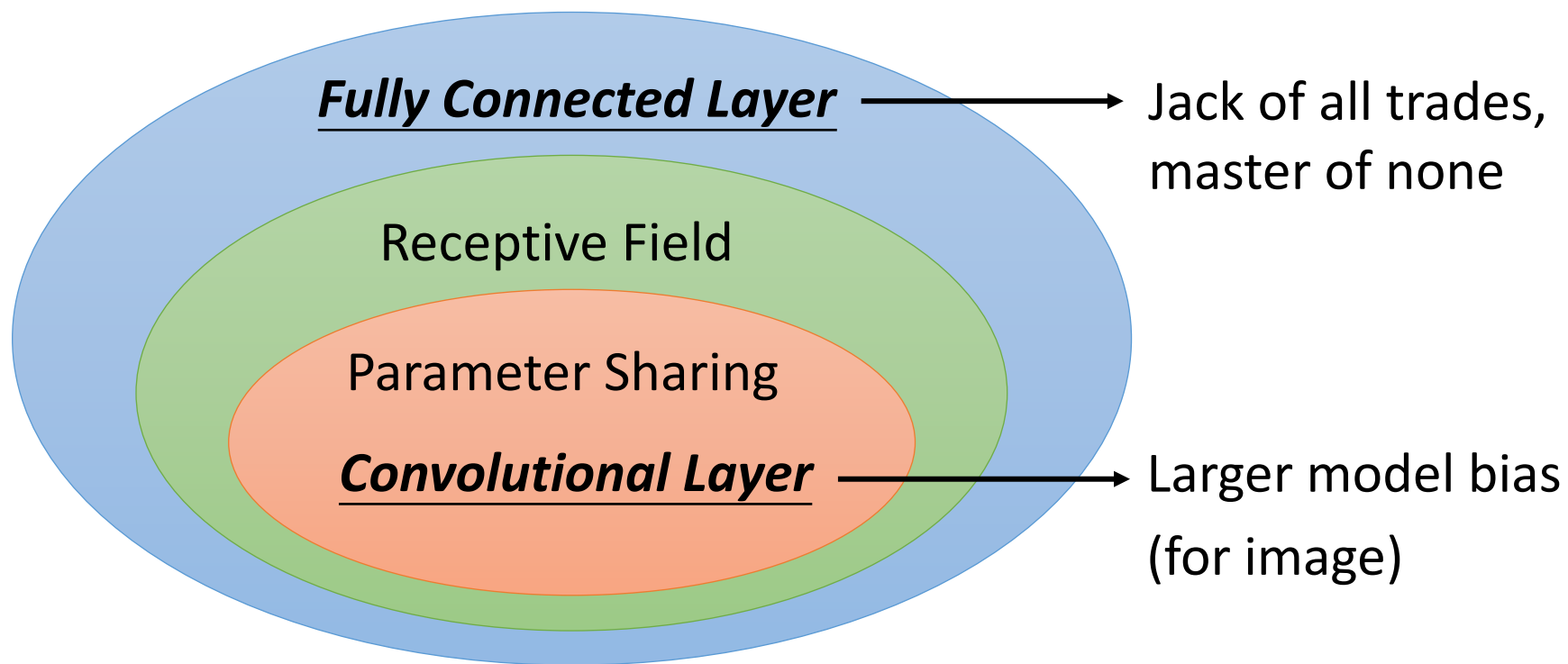
每个感受野有多个神经元 (e.g., 64 个神经元).

不同感受野的神经元共享参数 (filter, 滤波器)





卷积层的优势



- 通常pattern会远小于整图大小
- 同样的pattern会出现在图上不同区域