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

人工智能

第6讲：深度学习I

梯度下降与前馈神经网络

张晶

2025年春季

- 参考资料： 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

北京航空航天大學 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

北京航空航天大學 软件学院

提纲

一、线性回归与梯度下降

二、前馈神经网络

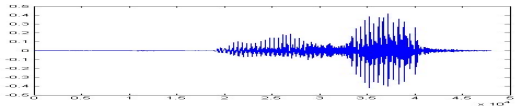
三、卷积神经网络

四、序列数据模型

五、深度学习应用

机器学习：拟合函数$f(x) = y$

- 语音识别

$f($  $)=$ “你好”

- 图像分类

$f($  $)=$ “猫”

- 围棋游戏

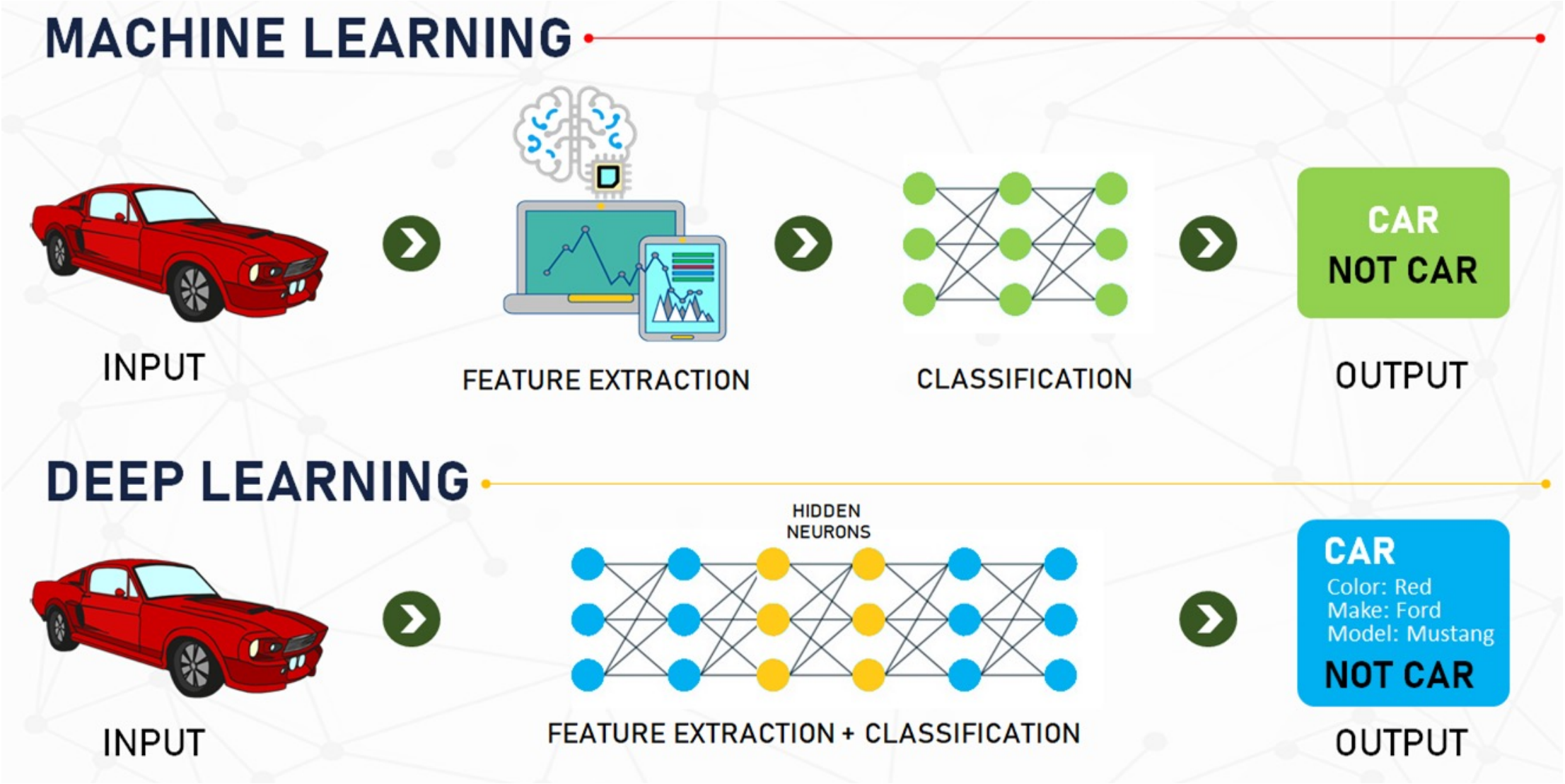
$f($  $)=$ “5–5”

（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

北京航空航天大學 软件学院

机器学习 v. s. 深度学习



回顾：线性回归

- 任务：网络视频观看量预测
- 训练数据$\mathcal{D}$：2017/01/01 – 2020/12/31

$$\mathcal{D} = \{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \cdots, (x^N, \hat{y}^N)\}$$

x^1：2017/01/01 观看量

$\hat{y}^1$：2017/01/02 观看量

x^2：2017/01/02 观看量

$\hat{y}^2$：2017/01/03 观看量

x^N：2020/12/30 观看量

$\hat{y}^N$：2020/12/31 观看量

北京航空航天大學 软件学院

回顾：线性回归

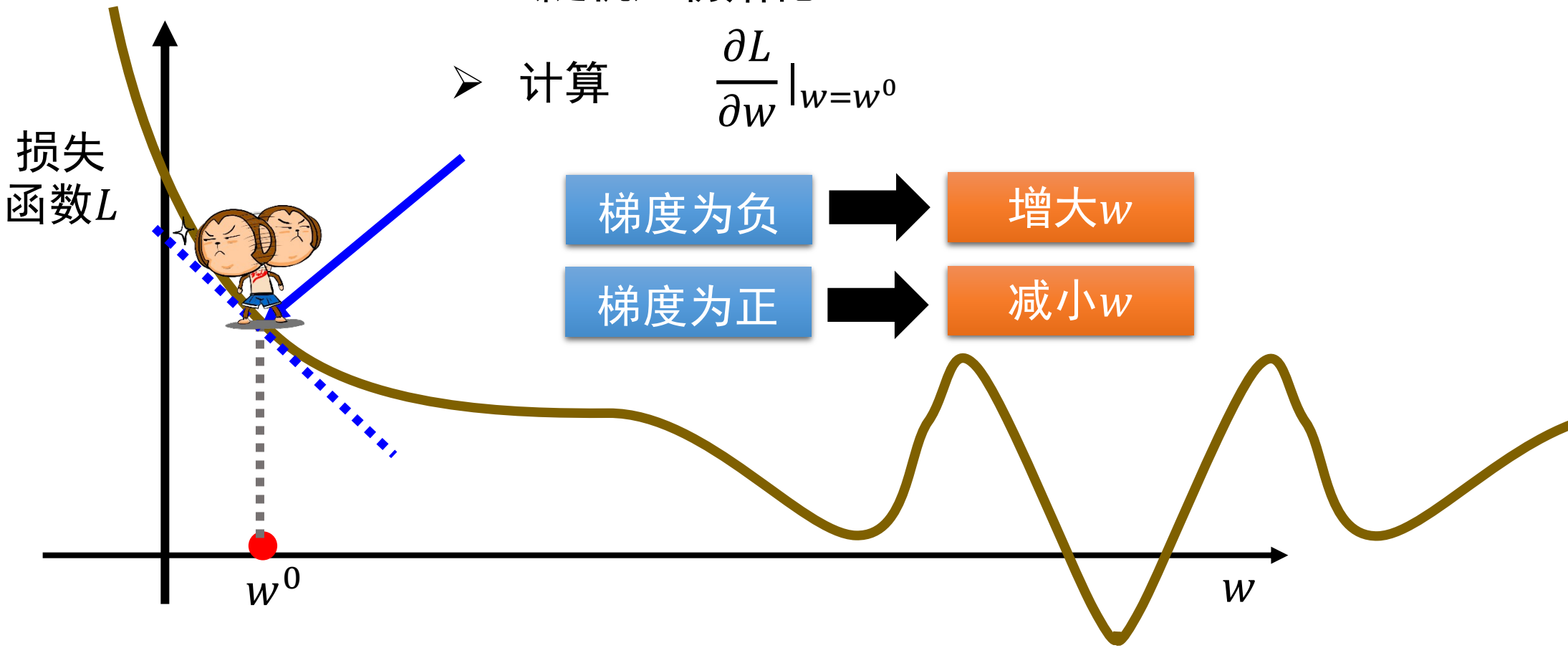
- 训练数据：$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$
- 学习模型：$f(x_i) = wx_i + b\ (1 \leq i \leq n)$
- 损失函数：$L(w, b) = \sum_{i=1}^{n} loss(x_i, y_i) = \sum_{i=1}^{n} (y_i - (wx_i + b))^2$
- 优化方法：最小二乘法或梯度下降法

Source of image: http://chico386.pixnet.net/album/photo/171572850

优化方法：梯度下降

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

- （随机）初始化w^0
- 计算 $\frac{\partial L}{\partial w}|_{w=w^0}$



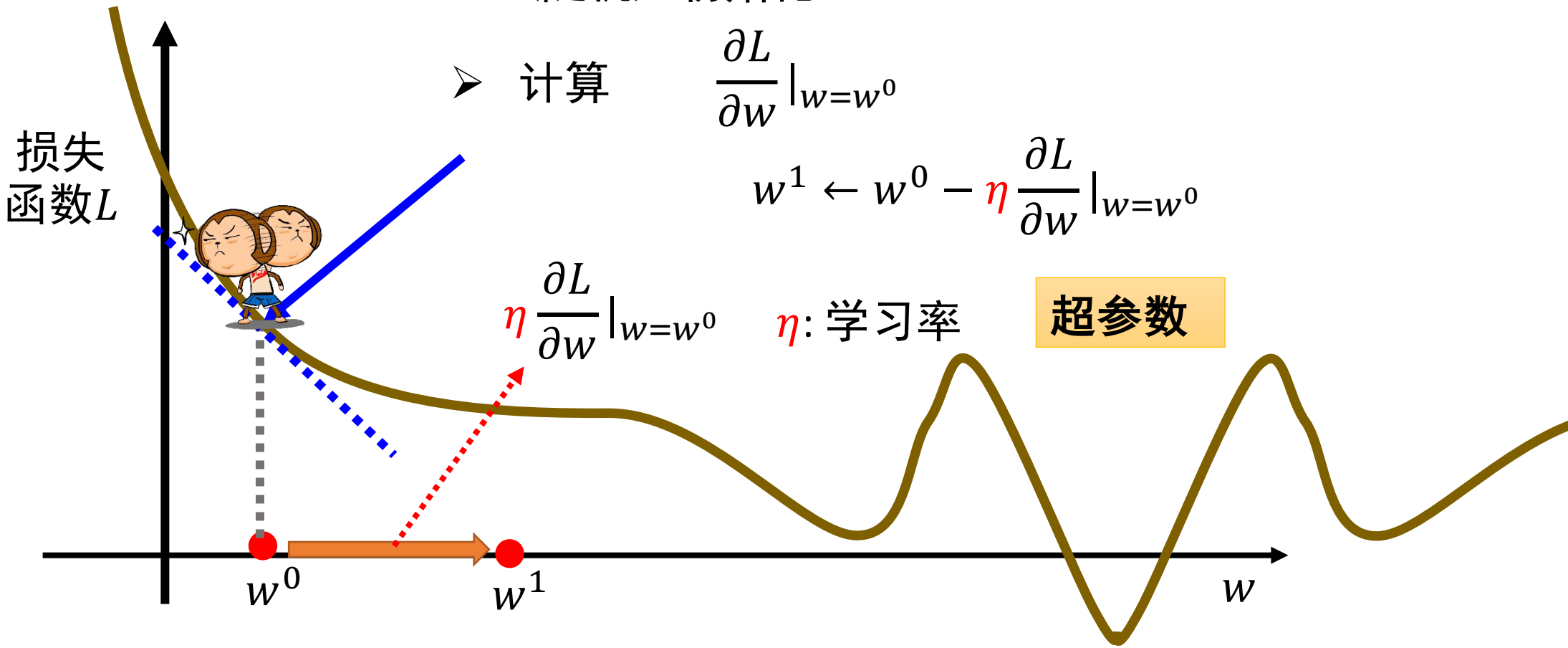
Source of image: http://chico386.pixnet.net/album/photo/171572850

优化方法：梯度下降

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

- （随机）初始化w^0
- 计算 $\frac{\partial L}{\partial w}|_{w=w^0}$

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



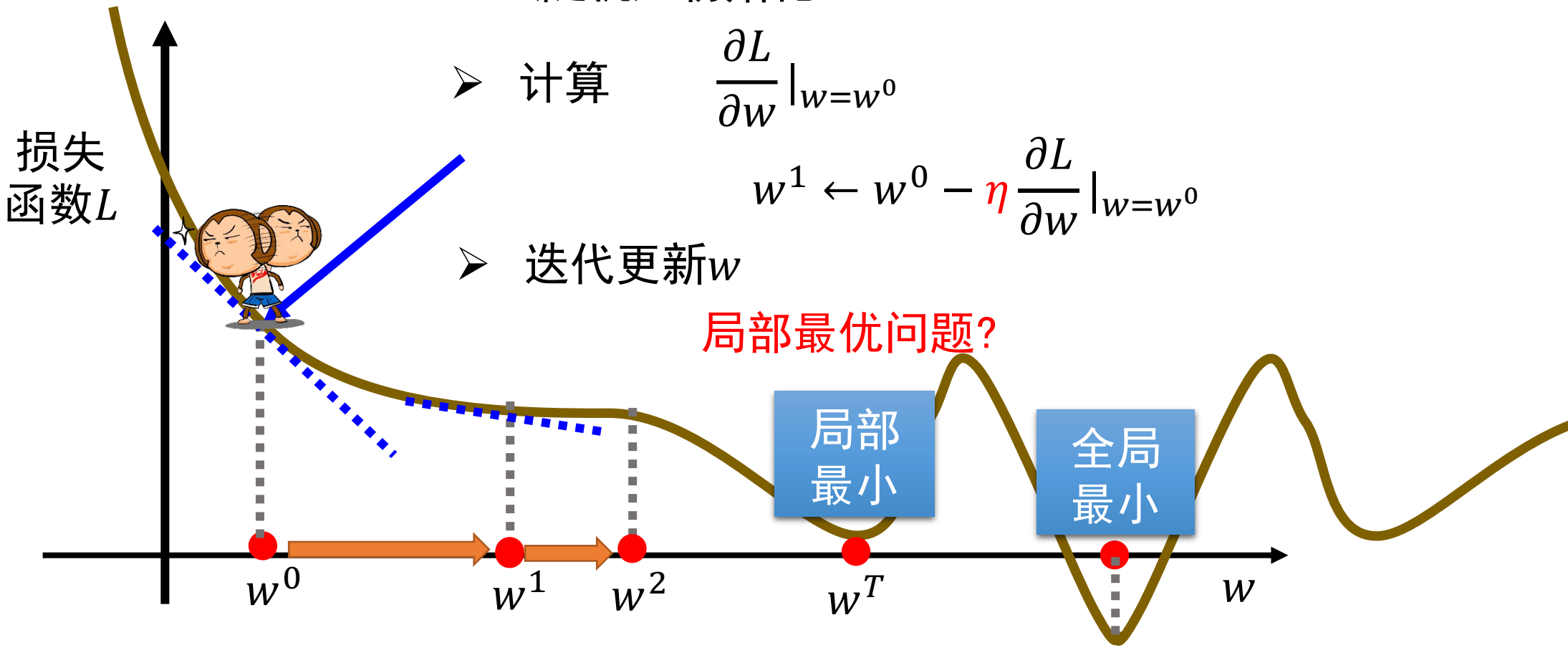
Source of image: http://chico386.pixnet.net/album/photo/171572850

优化方法：梯度下降

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

- （随机）初始化w^0
- 计算 $\frac{\partial L}{\partial w}|_{w=w^0}$
- 迭代更新w

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



优化方法：梯度下降

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

- （随机）初始化w^0, b^0
- 计算

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

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

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

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

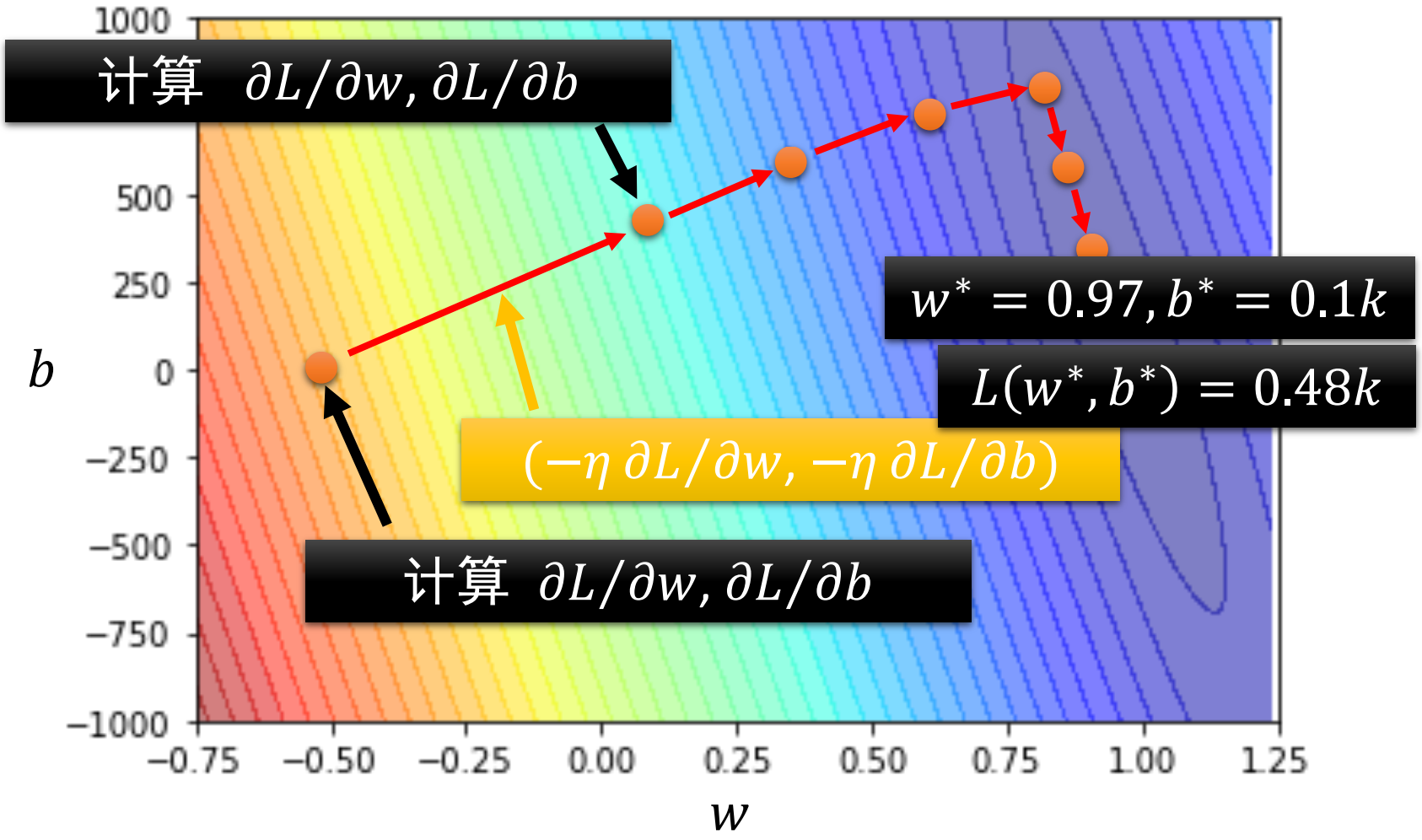
深度学习框架下一行代码足以实现！

- 迭代更新 w 和 b

优化方法：梯度下降

模型 $y = b + wx_1$

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

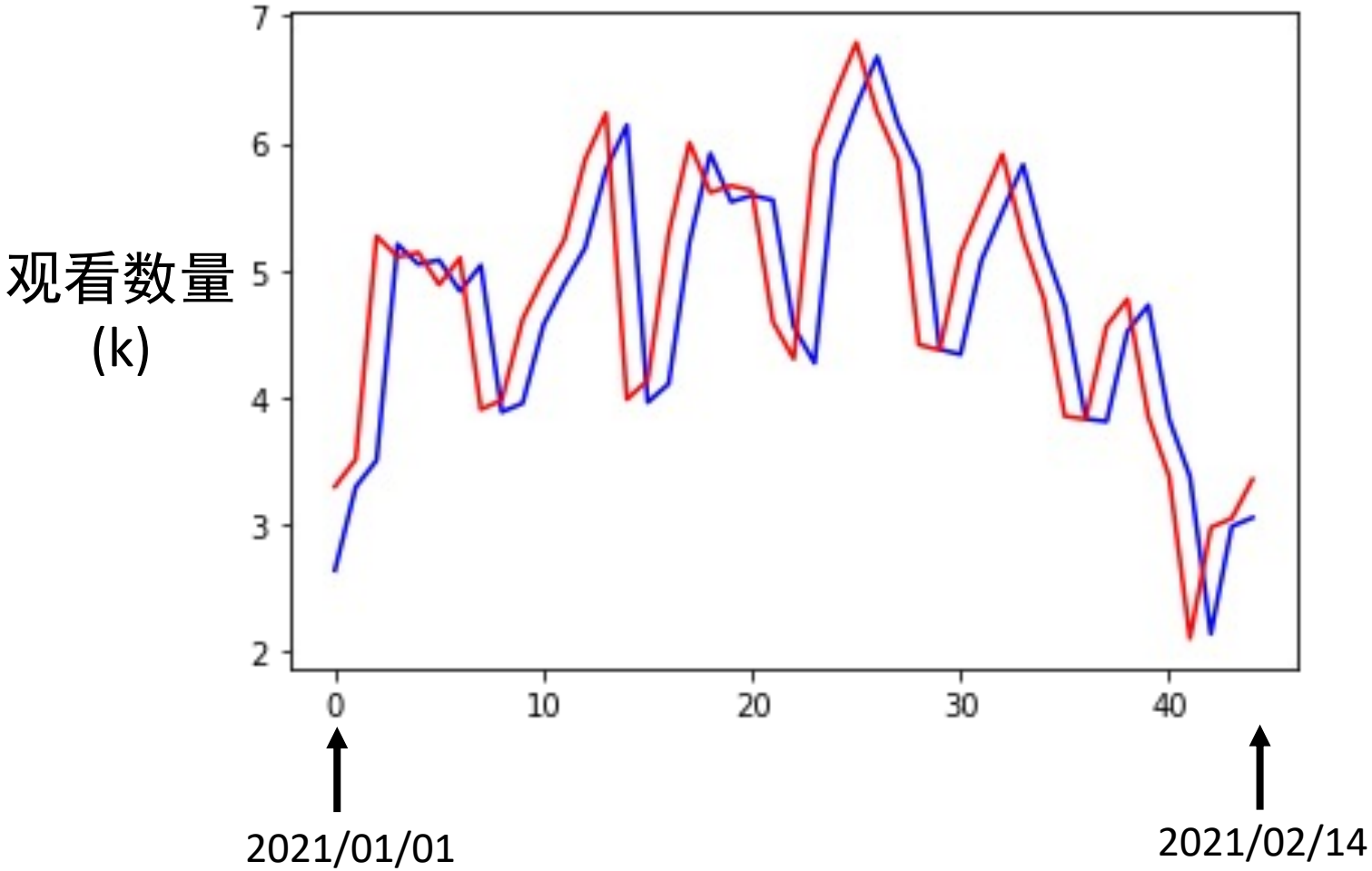


线性回归：一元线性回归

- 训练数据：$\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$
- 学习模型：$f(x_i) = wx_i + b \ (1 \leq i \leq n)$
- 损失函数：$L(w, b) = \sum_{i=1}^{n} loss(x_i, y_i) = \sum_{i=1}^{n} (y_i - (wx_i + b))^2$
- 优化方法：梯度下降法得到$w^* = 0.97, b^* = 0.1k$
- 结果预测：$f(x) = 0.97x + 100$

$$y = 0.1k + 0.97x_1$$

Red: 真实观看数量
blue: 预测观看数量



线性回归：多元线性回归

- 训练数据：$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\}$
  - 其中，$\boldsymbol{x_i} = [x_{i,1}, x_{i,2}, \ldots, x_{i,D}] \in \mathbb{R}^D$为代表多元特征的第$i$个数据向量
  - 可以用前N天的观看量作为多元特征。
- 学习模型：$f(\boldsymbol{x_i}) = \sum_{j=1}^{D} w_j\, x_{i,j} + w_0 = \boldsymbol{w^T x_i} + w_0, (1 \le i \le n)$
- 损失函数：$L(\boldsymbol{w}) = \sum_{i=1}^{n} loss(\boldsymbol{x}_i, y_i) = \sum_{i=1}^{n} (y_i - (\boldsymbol{w^T x_i} + w_0))^2$
- 优化方法：最小二乘法或梯度下降法

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

$$y = b + wx_1$$

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 $L' = 0.49k$

填空题 3分

设置

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

1. 如果用前7天的数据作为输入特征，那么特征维度是 [填空1]

2. 如果用前7天的数据作为输入特征，那么该线性模型参数量是 [填空2]

3. 直觉上， 你认为前7天（前1天，前2天，...，前7天）的数据中，哪一天的数据对当天观看量的预测最重要？ [填空3]

作答

多元线性模型（$y = b + \boldsymbol{w}^T \boldsymbol{x}$）

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

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 $L' = 0.46k$

北京航空航天大學 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 软件学院

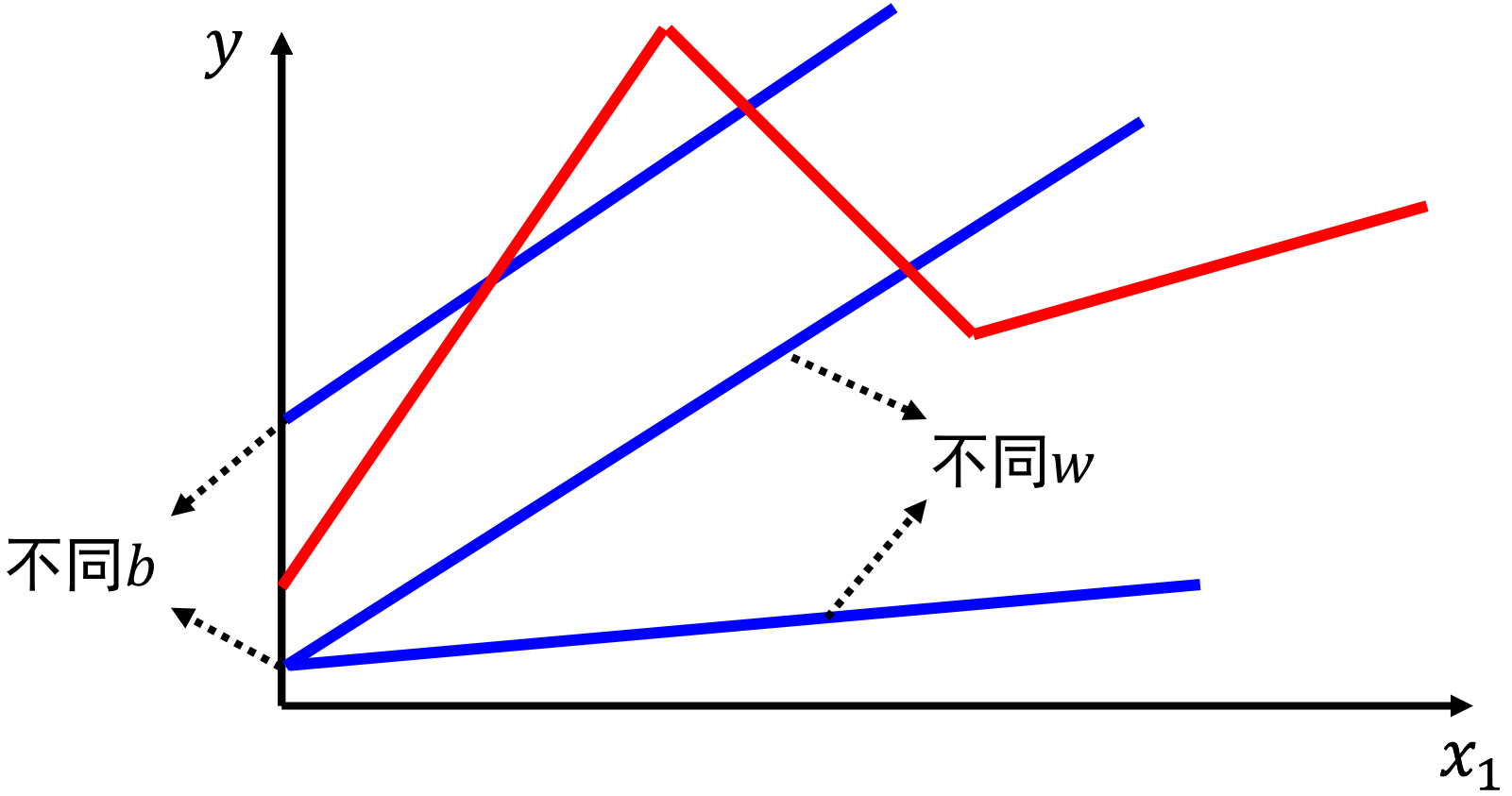
为何着重讨论线性模型?（ $y = b + wx_1$）

- 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

回归分析

- 训练数据：$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\}$
- 学习模型：$f(\boldsymbol{x_i}) = \boldsymbol{w^T x_i} + w_0, (1 \leq i \leq n)$
  - 扩展线性模型为非线性！
- 损失函数：$L(\boldsymbol{w}) = \sum_{i=1}^{n} loss(\boldsymbol{x}_i, y_i) = \sum_{i=1}^{n} (y_i - (\boldsymbol{w^T x_i} + w_0))^2$
- 优化方法：最小二乘法或梯度下降法

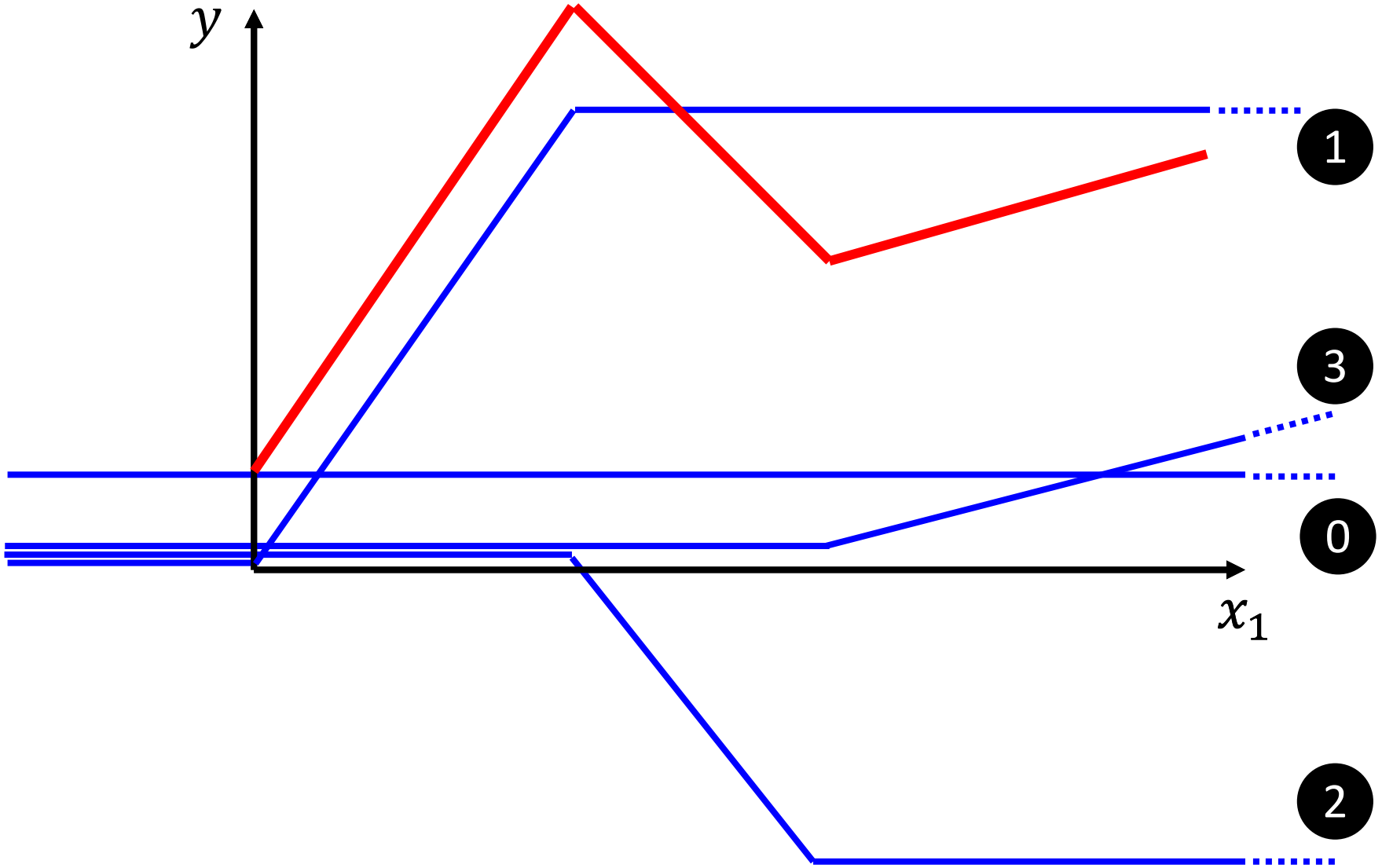
线性模型的扩展



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。

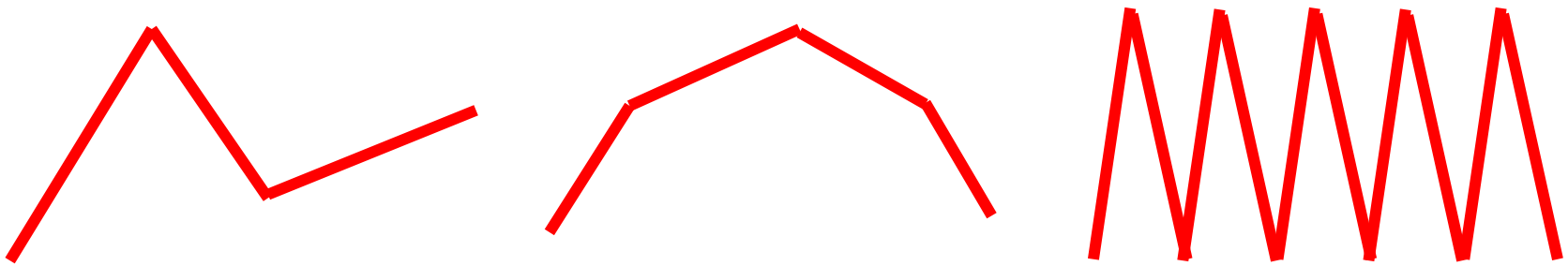
线性模型的扩展

red curve = constant + sum of a set of



分段线性函数

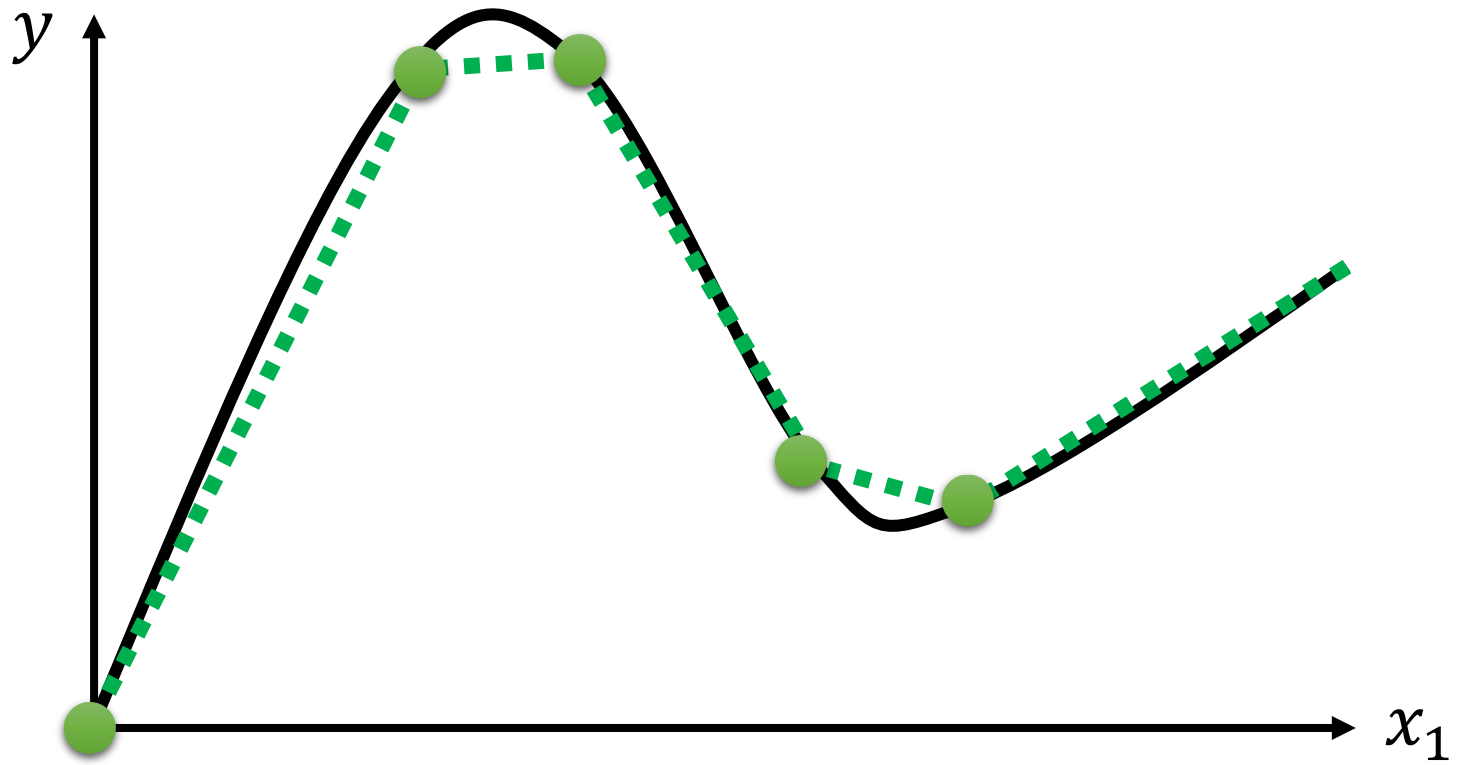
= constant + sum of a set of 



More pieces require more 

分段线性模型

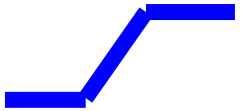
用分段线性模型拟合连续曲线



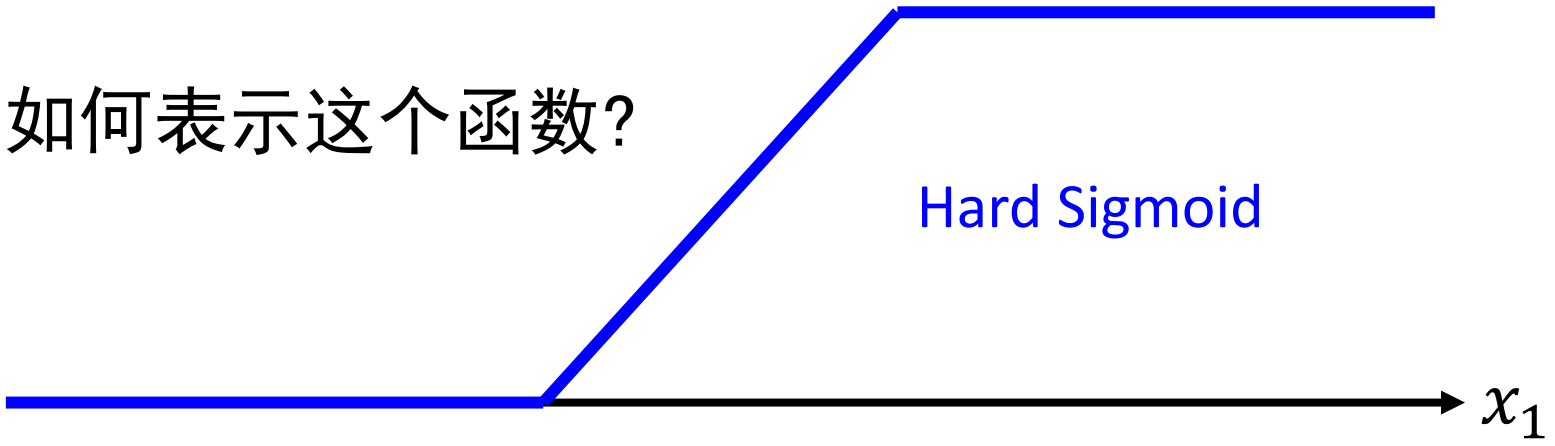
为了更好的近似连续曲线，需要更多分段线性模型

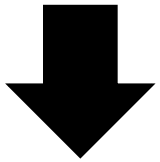
北京航空航天大學 软件学院

red curve = constant + sum of a set of



如何表示这个函数?

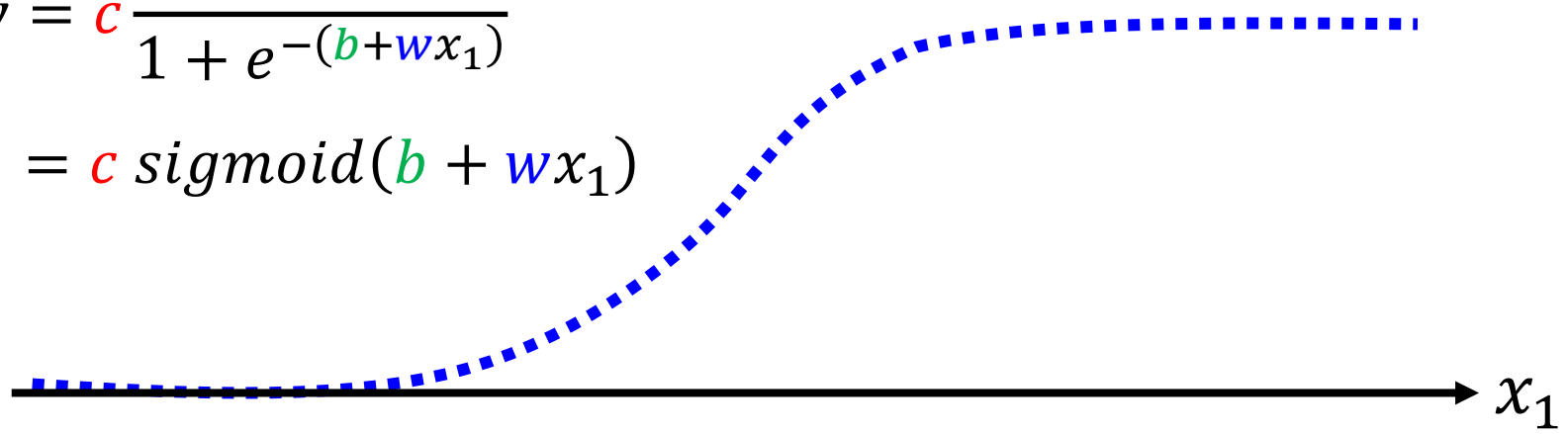




***Sigmoid* 函数**

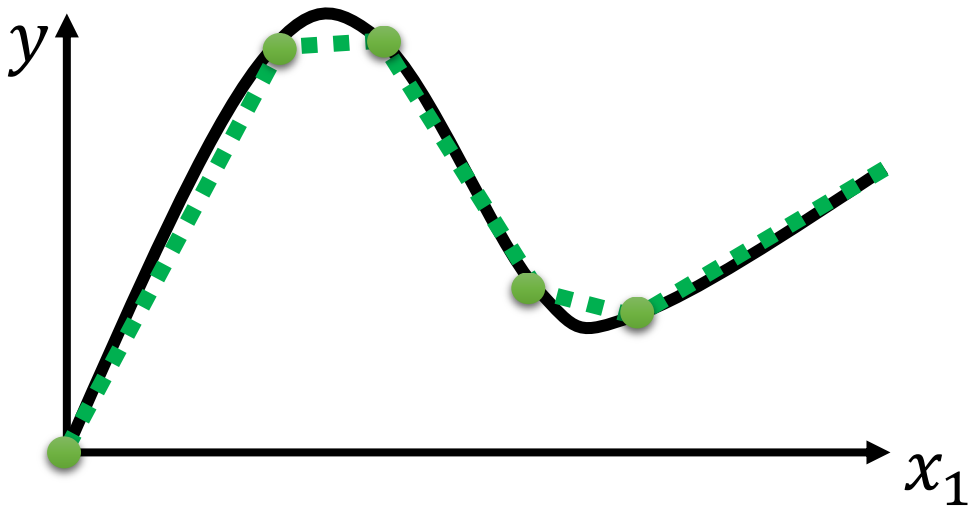
$$y = c\frac{1}{1+e^{-(b+wx_1)}}$$

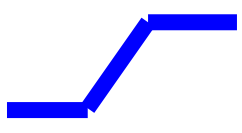
$$= c\ sigmoid(b + wx_1)$$



填空题 2分

设置

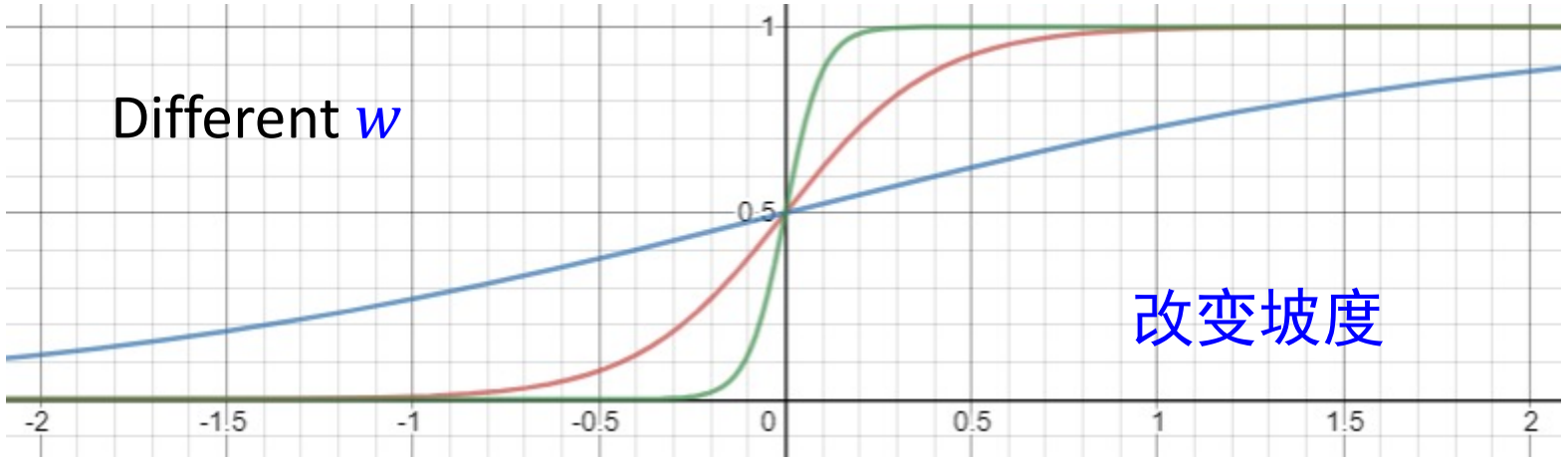


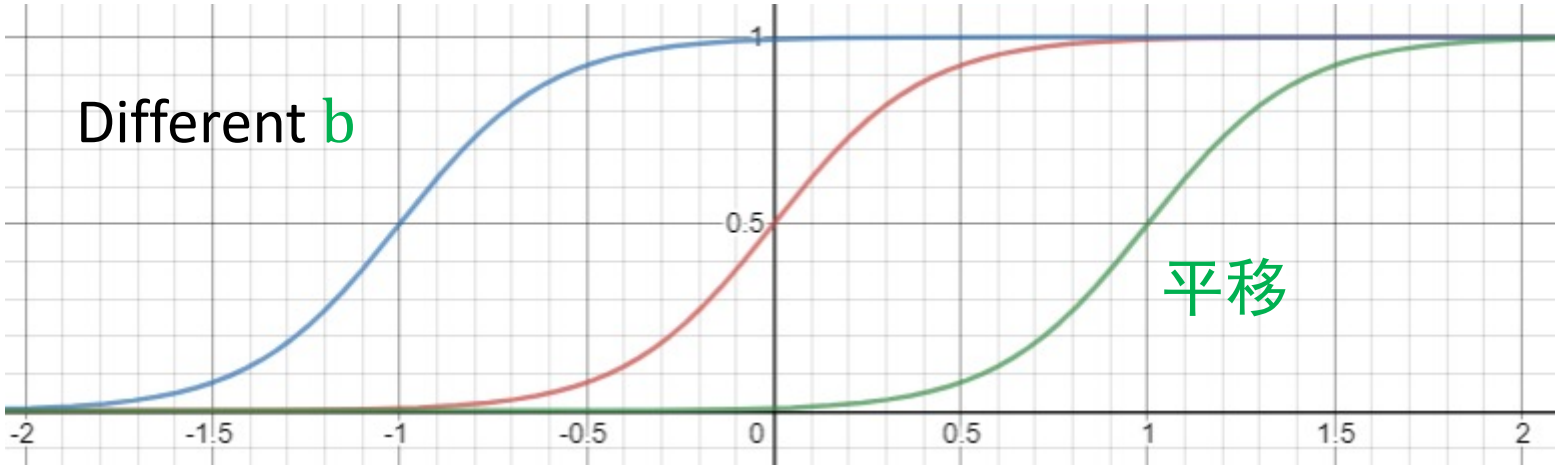
分段线性函数 = 常量 + $\sum$ 

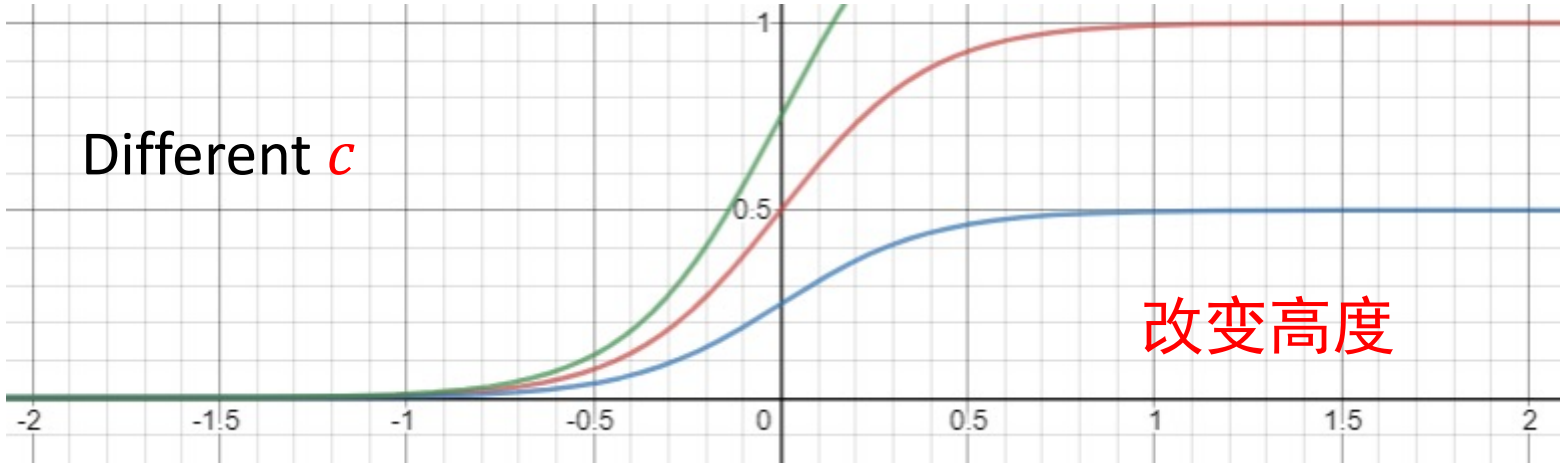
如果激活函数是如下简单线性模型（而非分段线性），是否还能近似复杂非线性函数 [填空1]？为什么 [填空2]？

常量 + $\sum$ 

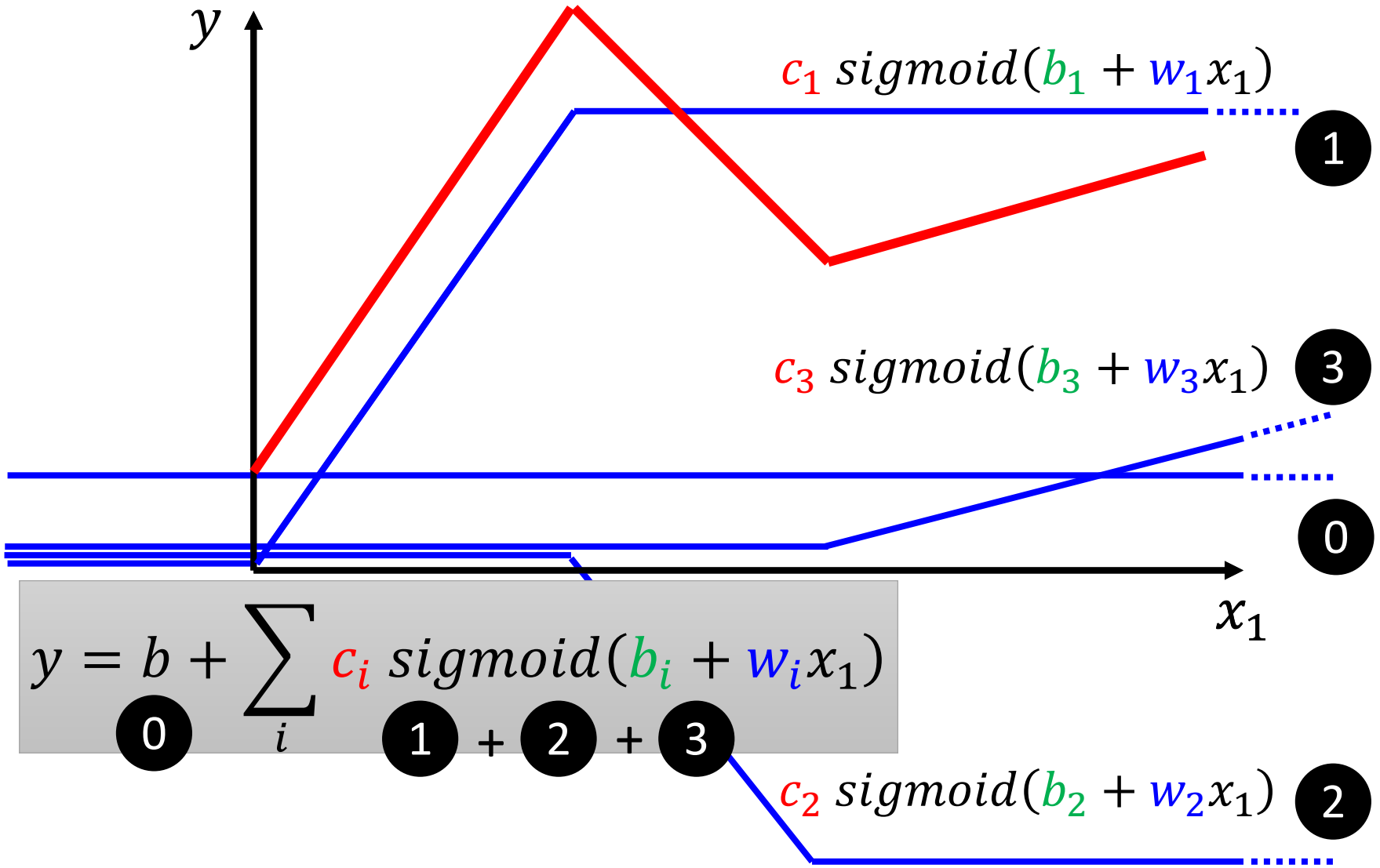
作答


Different w
改变坡度
1
0.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Different b
平移
1
0.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


Different c
改变高度
1
0.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

red curve = sum of a set of  + constant



$$y = \underset{0}{b} + \sum_i \underset{1}{c_i}\ sigmoid(\underset{2}{b_i} + \underset{3}{w_i} x_1)$$

北京航空航天大學 软件学院

新模型：更多模型组合

$$y = b + wx_1$$

$$y = b + \sum_i c_i \, sigmoid(b_i + w_i x_1)$$

$$y = b + \sum_j w_j x_j$$

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

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

j: 1,2,3　特征个数

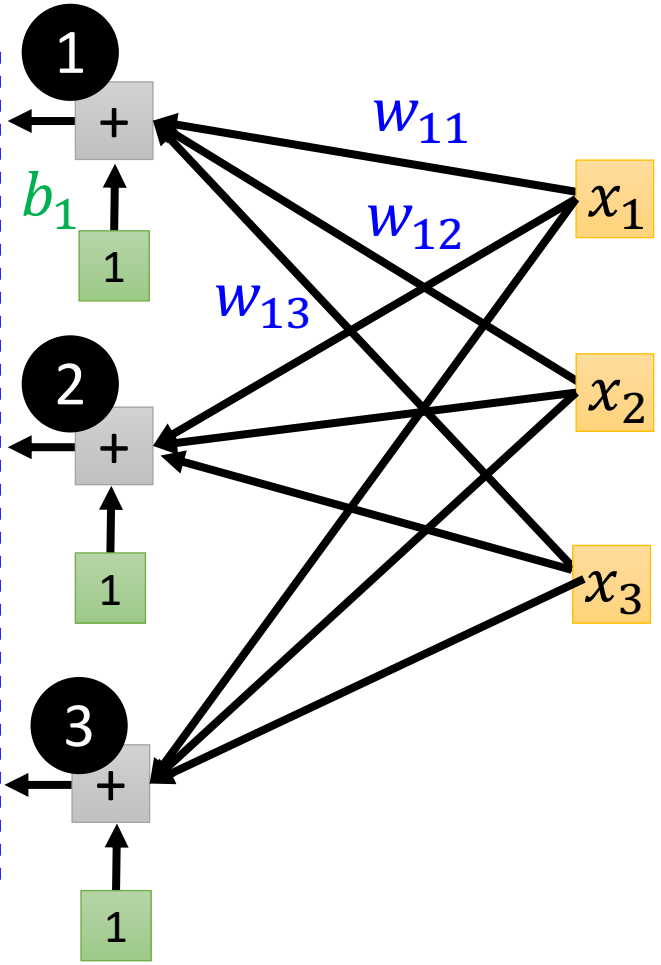
i: 1,2,3　sigmoid个数

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

w_{ij}: weight for x_j for i-th sigmoid

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

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



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

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

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

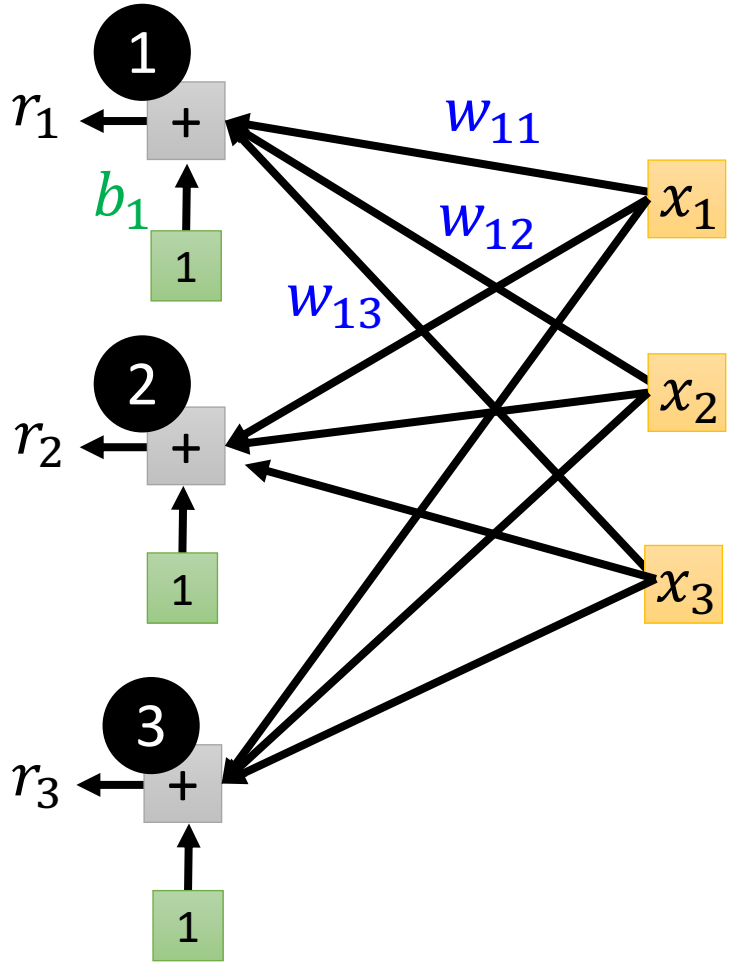
$$\boldsymbol{r} = \boldsymbol{b} + W \boldsymbol{x}$$

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

i: 1,2,3

j: 1,2,3

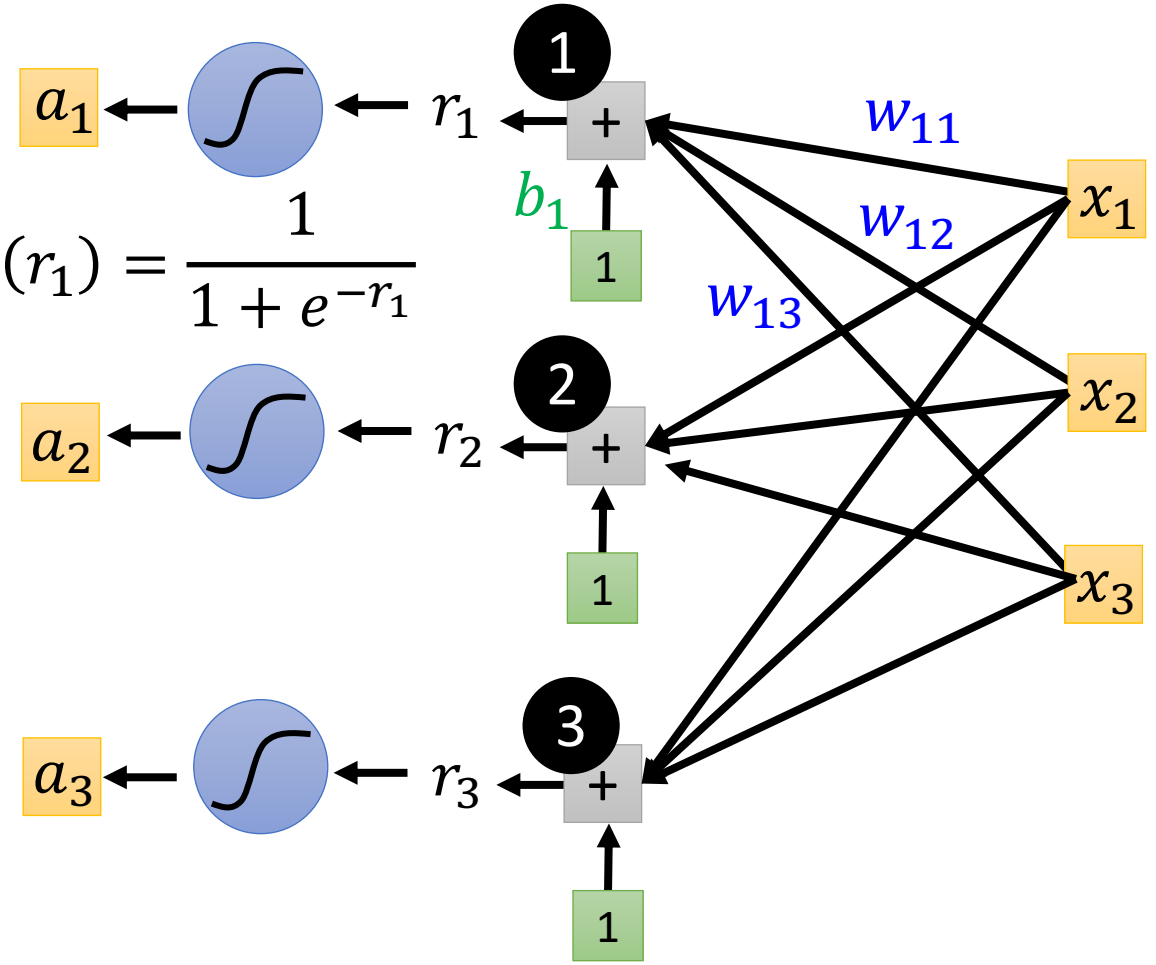
$$\boldsymbol{r} = \boldsymbol{b} + W \boldsymbol{x}$$



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

i: 1,2,3

j: 1,2,3



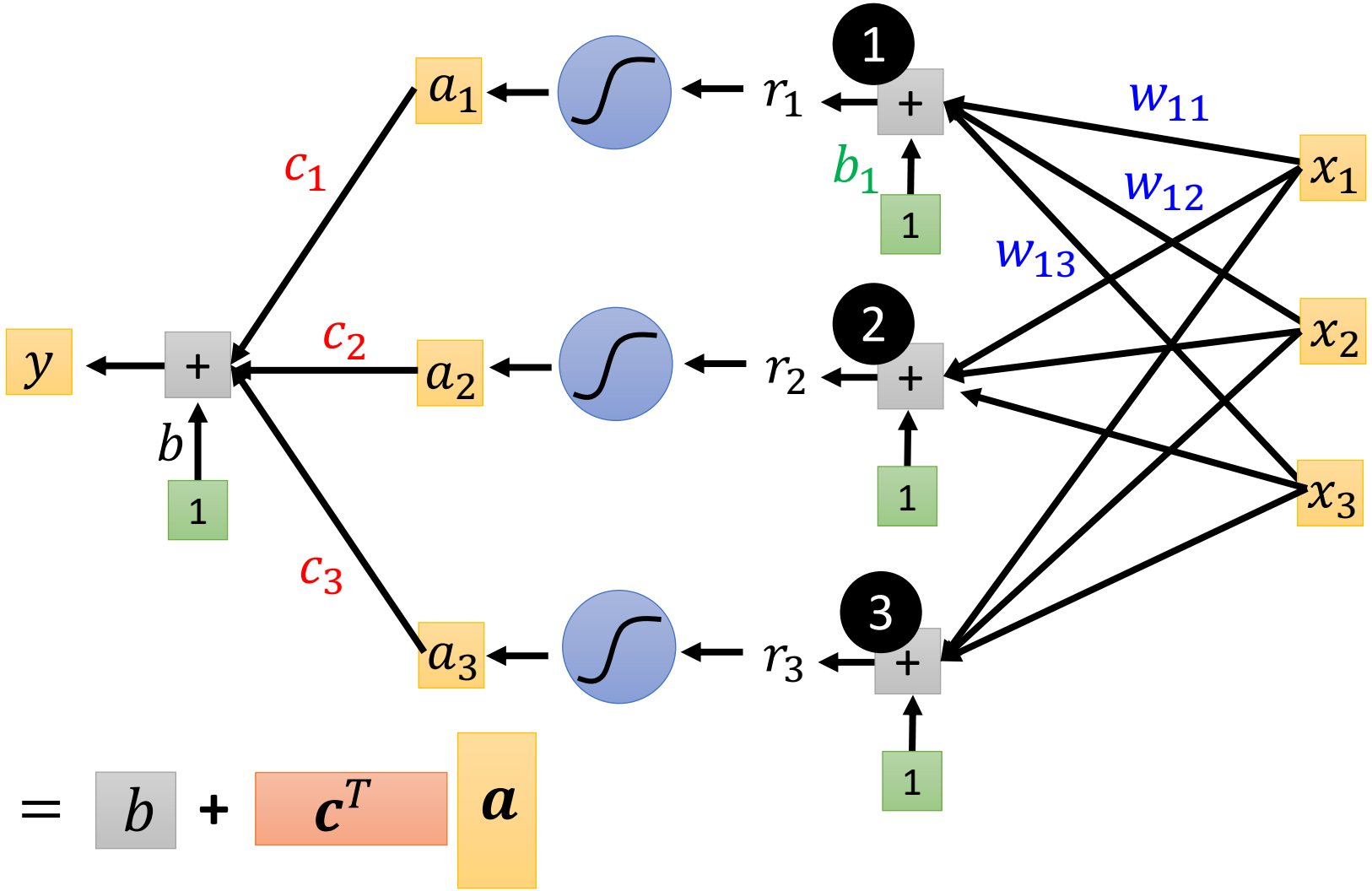
$$a_1 = sigmoid(r_1) = \frac{1}{1 + e^{-r_1}}$$

$$\boldsymbol{a} = \sigma(\ \boldsymbol{r}\ )$$

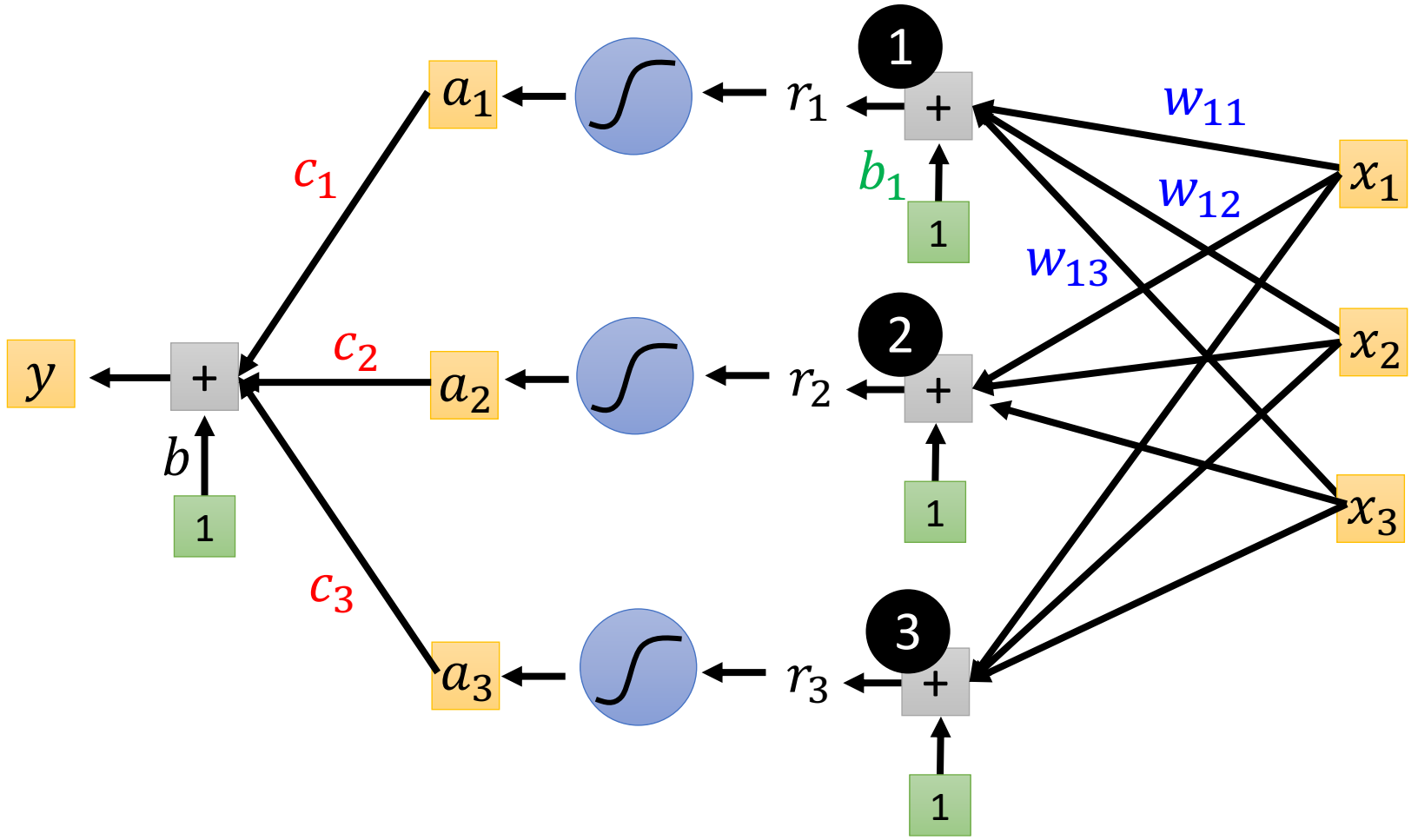
$$y = b + \sum_i c_i \, sigmoid\left(b_i + \sum_j w_{ij} x_j\right)$$

i: 1,2,3

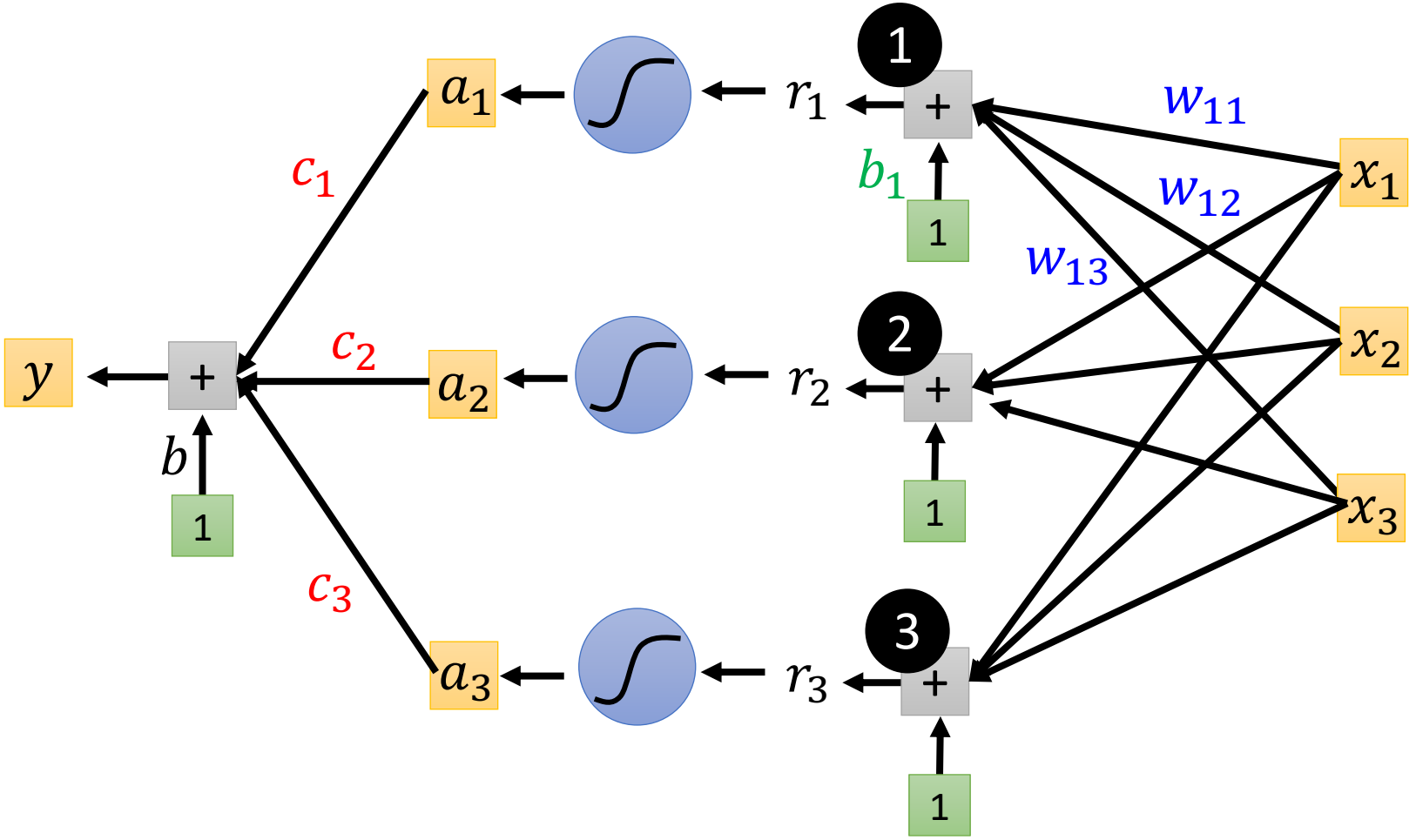
j: 1,2,3



$$y = b + \boldsymbol{c}^T \boldsymbol{a}$$



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



$$y = b + \boldsymbol{c}^T \sigma(\boldsymbol{b} + W\boldsymbol{x})$$

带有未知参数的函数

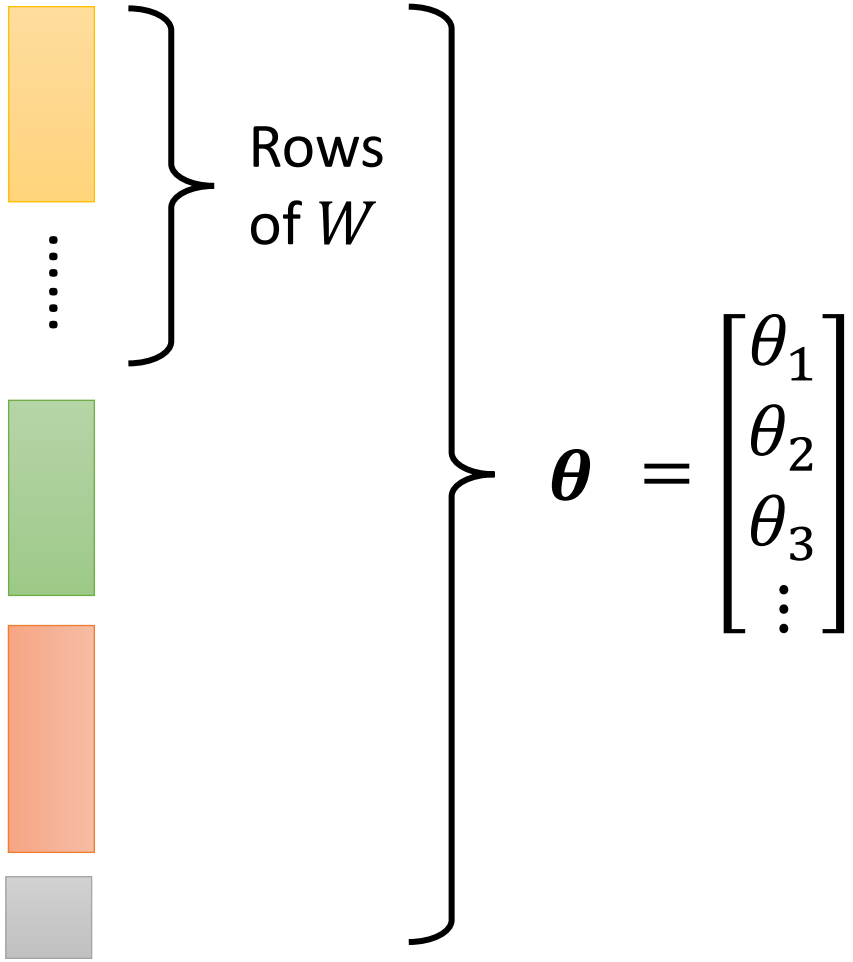
$$y = b + c^T \sigma(b + W x)$$

x 特征

未知参数

W b

c^T b



北京航空航天大學 软件学院

回归分析：非线性回归

- 训练数据：$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\}$
- 学习模型：$f(\boldsymbol{x_i}) = b + \boldsymbol{c}^T \sigma(\boldsymbol{b} + W\boldsymbol{x_i}), (1 \leq i \leq n)$
- 损失函数：$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} loss(\boldsymbol{x}_i, y_i) = \sum_{i=1}^{n} (y_i - f(\boldsymbol{x_i}))^2$
- 优化方法：梯度下降法

优化方法：梯度下降

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

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

- （随机）初始化参数 $\boldsymbol{\theta}^0$

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

梯度

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

$$\boldsymbol{g} = \nabla L(\boldsymbol{\theta}^0)$$

$$\boldsymbol{\theta}^1 \leftarrow \boldsymbol{\theta}^0 - \eta \boldsymbol{g}$$

优化方法：梯度下降

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

- （随机）初始化参数 $\boldsymbol{\theta}^0$
- 计算梯度 $\boldsymbol{g} = \nabla L(\boldsymbol{\theta}^0)$

$$\boldsymbol{\theta}^1 \leftarrow \boldsymbol{\theta}^0 - \eta \boldsymbol{g}$$

- 计算梯度 $\boldsymbol{g} = \nabla L(\boldsymbol{\theta}^1)$

$$\boldsymbol{\theta}^2 \leftarrow \boldsymbol{\theta}^1 - \eta \boldsymbol{g}$$

- 计算梯度 $\boldsymbol{g} = \nabla L(\boldsymbol{\theta}^2)$

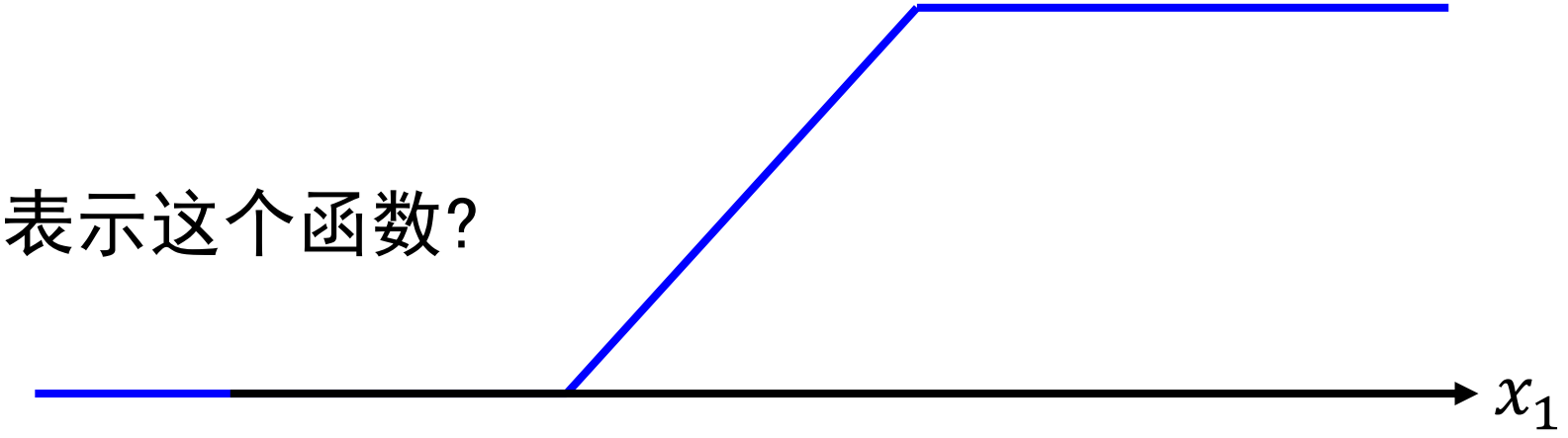
$$\boldsymbol{\theta}^3 \leftarrow \boldsymbol{\theta}^2 - \eta \boldsymbol{g}$$

回归分析

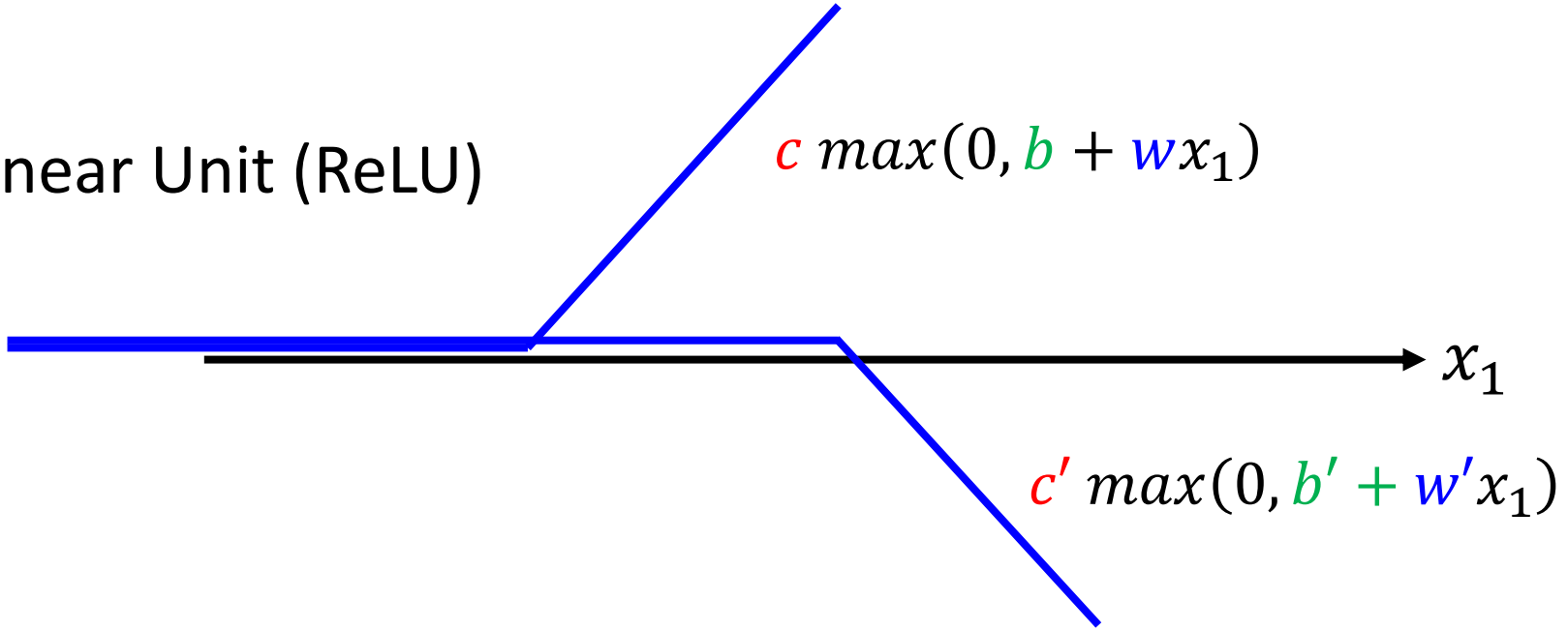
- 训练数据：$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\}$
- 学习模型：$f(\boldsymbol{x_i}) = b + \boldsymbol{c}^T \sigma(\boldsymbol{b} + W\boldsymbol{x_i}), (1 \leq i \leq n)$
  - 通过改变激活函数，改进学习模型！
- 损失函数：$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} loss(\boldsymbol{x}_i, y_i) = \sum_{i=1}^{n} (y_i - f(\boldsymbol{x_i}))^2$
- 优化方法：梯度下降法

Sigmoid → ReLU

如何表示这个函数?



Rectified Linear Unit (ReLU)



Sigmoid → ReLU

$$y = b + \sum_{i} c_i \, sigmoid\left(b_i + \sum_{j} w_{ij} x_j\right)$$

激活函数Activation function

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

如何选择激活函数?

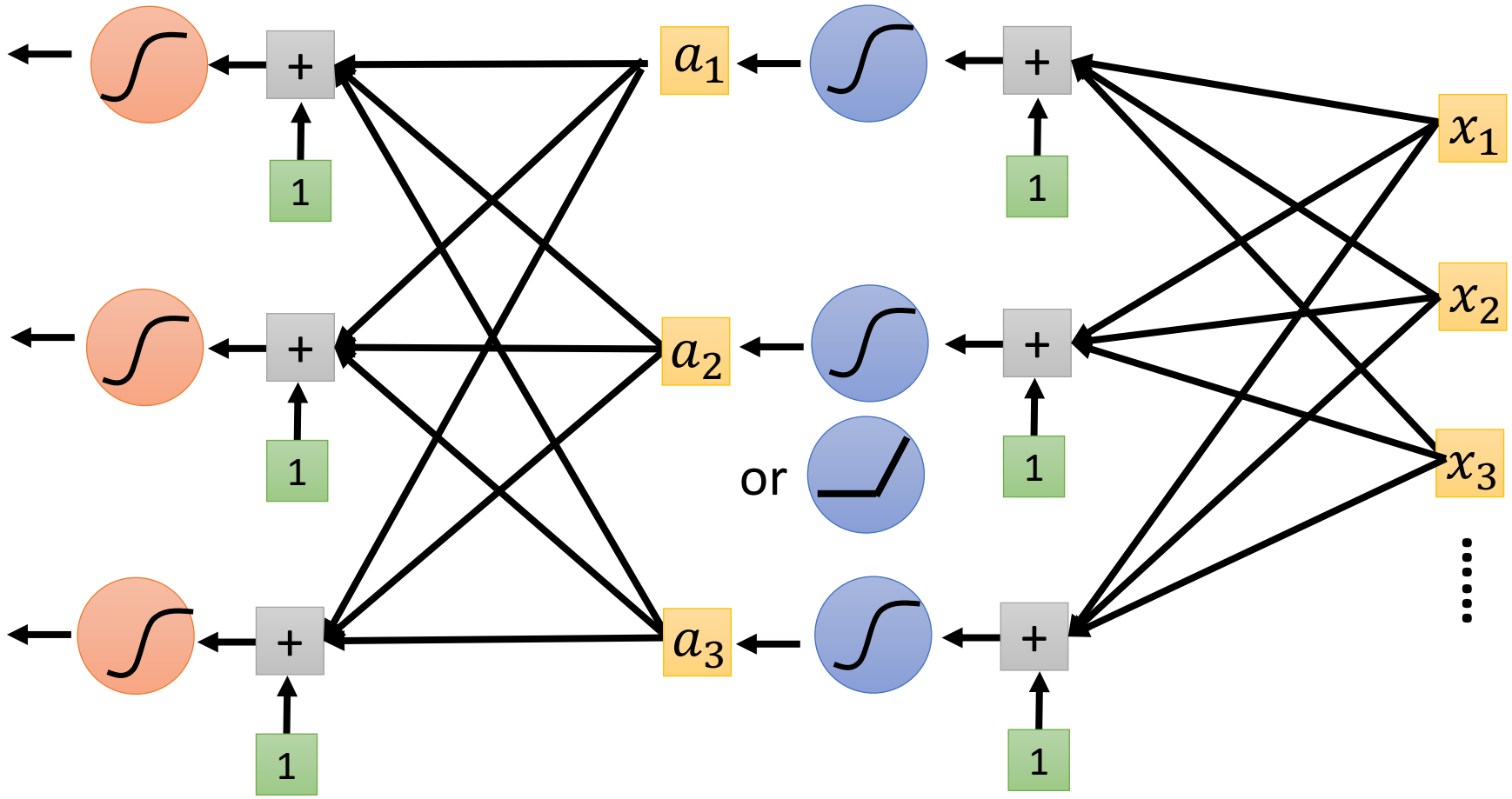
实验结果

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

	linear
2017 – 2020	0.32k
2021	0.46k

回归分析

- 训练数据：$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\}$
- 学习模型：$f(\boldsymbol{x_i}) = b + \boldsymbol{c}^T \sigma(\boldsymbol{b} + W\boldsymbol{x_i}), (1 \leq i \leq n)$
  - 通过增加模型层数，继续改进学习模型！
- 损失函数：$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} loss(\boldsymbol{x}_i, y_i) = \sum_{i=1}^{n} (y_i - f(\boldsymbol{x_i}))^2$
- 优化方法：梯度下降法



$$\boldsymbol{a}' = \sigma(\boldsymbol{b}' + W'\boldsymbol{a}) \qquad \boldsymbol{a} = \sigma(\boldsymbol{b} + W\boldsymbol{x})$$

北京航空航天大學 软件学院

回归分析

- 训练数据：$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_n, y_n)\}$
- 学习模型：$f(\boldsymbol{x_i}) = b + \boldsymbol{c}^T \sigma(\boldsymbol{b}' + W'(\sigma(\boldsymbol{b} + W\boldsymbol{x_i}))), (1 \leq i \leq n)$
- 损失函数：$L(\boldsymbol{\theta}) = \sum_{i=1}^{n} loss(\boldsymbol{x}_i, y_i) = \sum_{i=1}^{n} (y_i - f(\boldsymbol{x_i}))^2$
- 优化方法：梯度下降法

北京航空航天大学 软件学院

实验结果

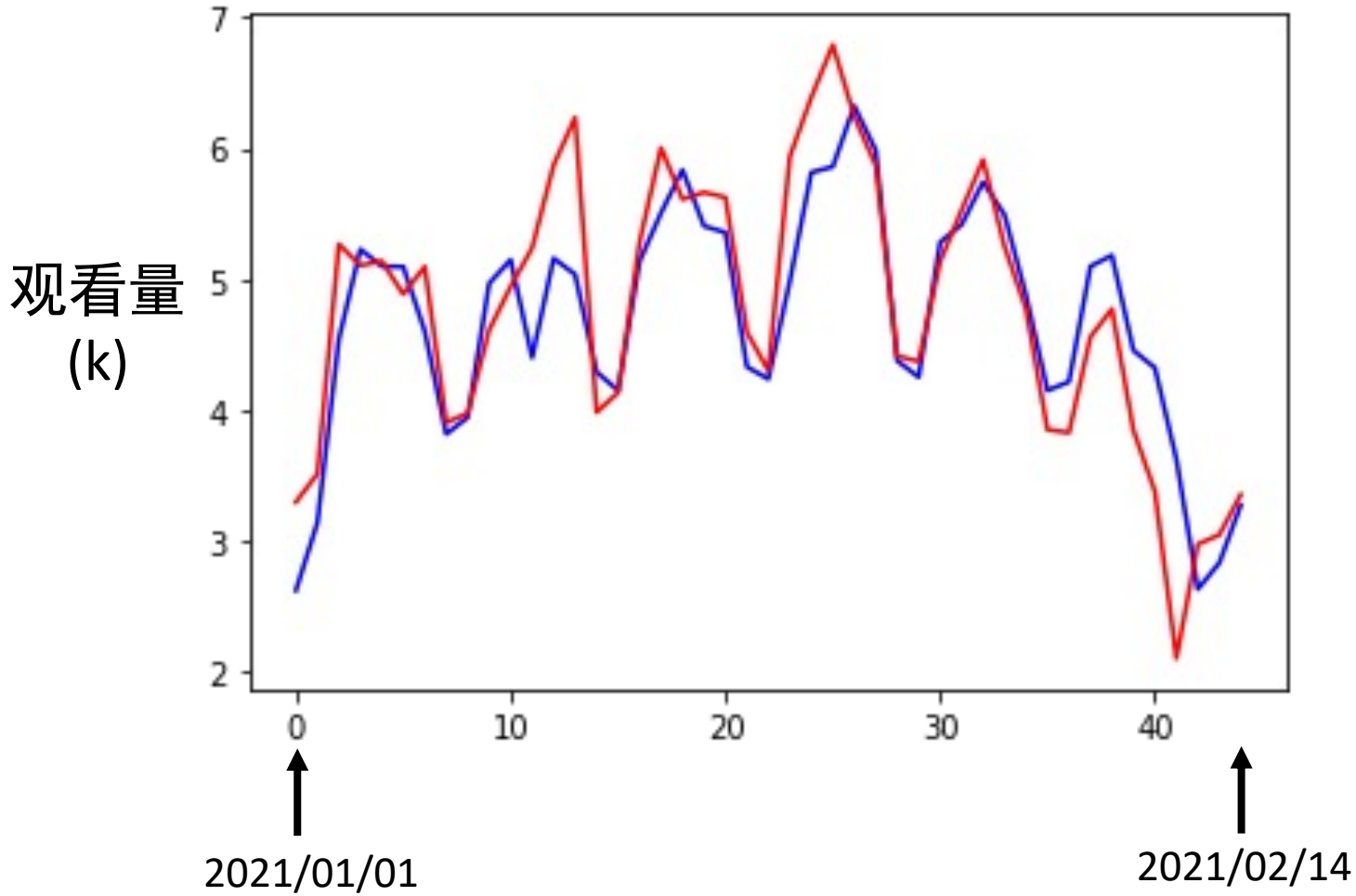
- 多个隐含层模型的损失
  - 每层100个 ReLU 神经元
  - 输入特征是过去56天的观看量

	1 layer
2017 – 2020	0.28k
2021	0.43k

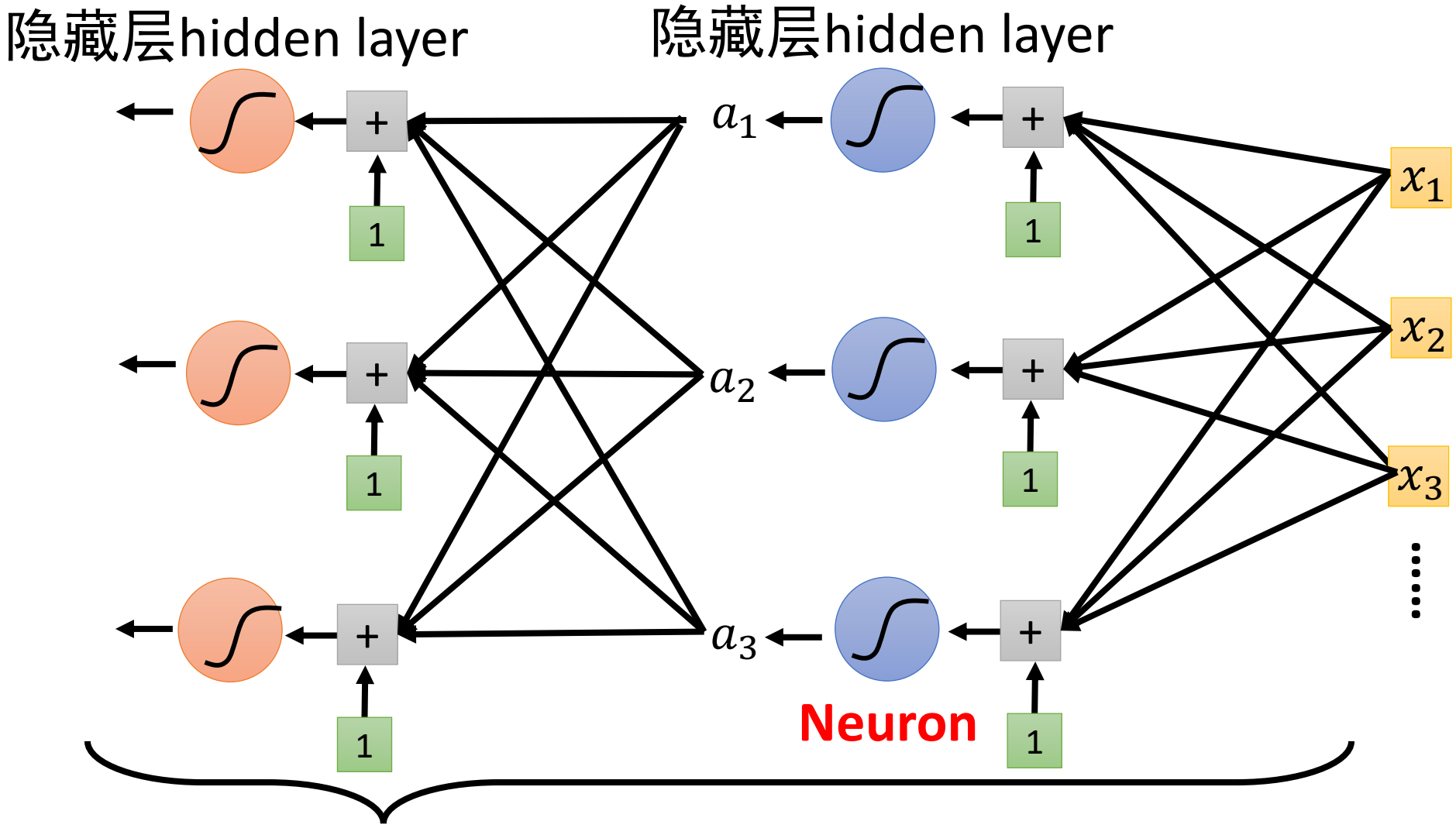
北京航空航天大學 软件学院

3 layers

Red：真实观看量
blue：模型预测观看量



北京航空航天大学 软件学院



神经网络 Neural Network

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

北京航空航天大學 软件学院

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。