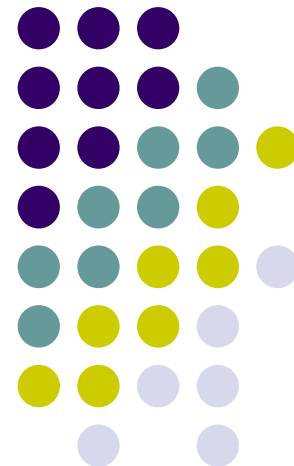


感知机模型与多层感知机

哈尔滨工业大学计算学部 刘远超



感知机模型与多层感知机



本章内容简介

- 感知机模型简介
- 感知机模拟逻辑电路的能力
- 感知机模型的学习策略
- 多层感知机

感知机模型与多层感知机



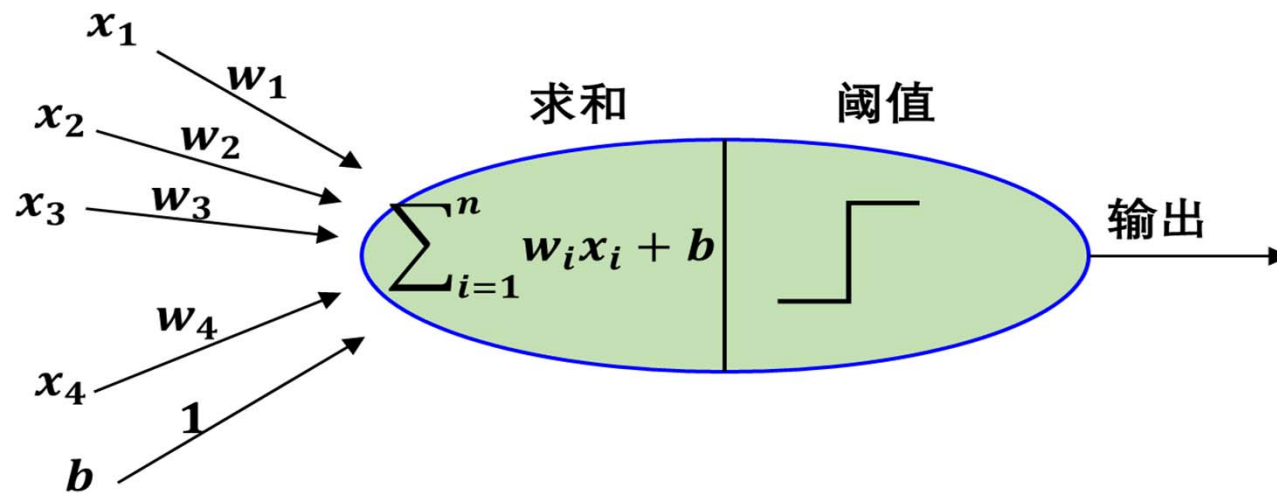
本章内容简介

- 感知机模型简介
- 感知机模拟逻辑电路的能力
- 感知机模型的学习策略
- 多层感知机

感知机模型简介

4

原理及数学描述



输出 y 的数学描述为:

$$y = g(\sum_{i=1}^n w_i x_i + b) = g(W^T X),$$

$$\text{其中, } W = [w_1, w_2, \dots, w_n, b]^T, X = [x_1, x_2, \dots, x_n, 1]^T, g(z) = \begin{cases} +1, & z \geq 0 \\ -1, & z < 0 \end{cases}.$$

感知机模型与多层感知机

5

本章内容简介

- 感知机模型简介
- 感知机模拟逻辑电路的能力
- 感知机模型的学习策略
- 多层感知机

感知机模型与多层感知机

6

本章内容简介

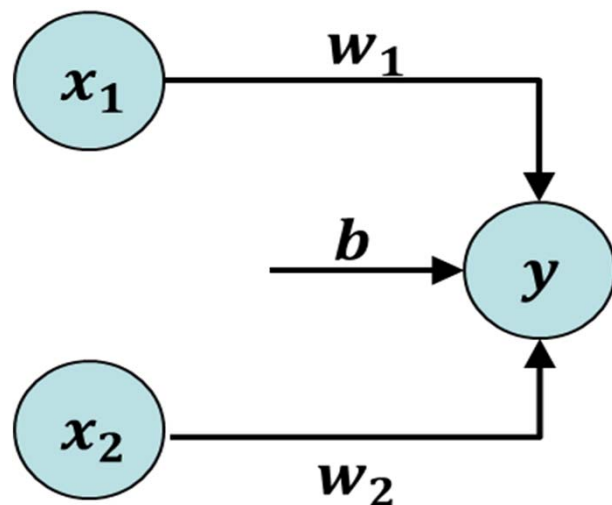
- 感知机模型简介
- **感知机模拟逻辑电路的能力**
- 感知机模型的学习策略
- 多层感知机

感知机的表达能力：模拟逻辑电路

7

如何使用感知机表示逻辑门电路？

可以使用图所示的感知机结构，并设定感知机的阶跃函数为 $g(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$ 。则需要确定参数 w_1 、 w_2 、 b 的具体取值，以使感知机的输入输出能满足逻辑门的真值表，如下面各小节所示。



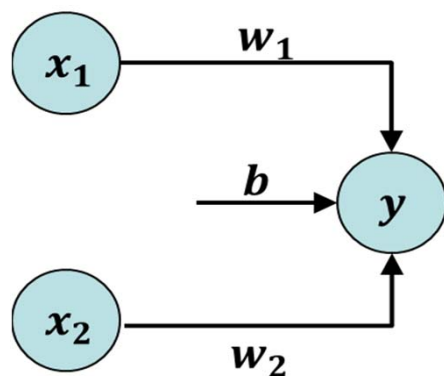
感知机的表达能力：模拟逻辑电路

8

感知机可以模拟与门电路

与门的真值表

x_1	x_2	y
0	0	0
1	0	0
0	1	0
1	1	1



满足与门真值表的感知机参数取值组合有无数个：

- $(w_1, w_2, b) = (0.5, 0.5, -0.7)$
- $(w_1, w_2, b) = (0.5, 0.5, -0.8)$
- $(w_1, w_2, b) = (0.5, 0.5, -0.9)$

设定感知机的阶跃函数为 $g(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$ 。

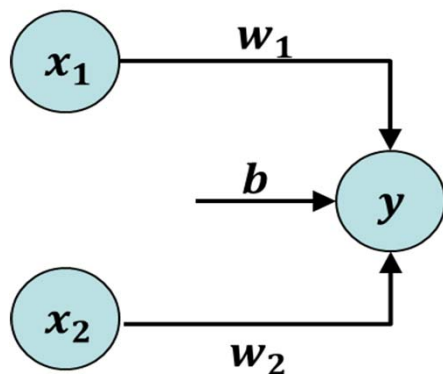
感知机的表达能力：模拟逻辑电路

9

感知机可以模拟与非门电路

与非门的真值表

x_1	x_2	y
0	0	1
1	0	1
0	1	1
1	1	0



满足与非门真值表条件的感知机参数取值组合也有无数个。例如：

- $(w_1, w_2, z_\tau) = (-0.5, -0.5, 0.7)$
- $(w_1, w_2, z_\tau) = (-0.5, -0.5, 0.8)$
- $(w_1, w_2, z_\tau) = (-0.5, -0.5, 0.9)$

设定感知机的阶跃函数为 $g(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$ 。

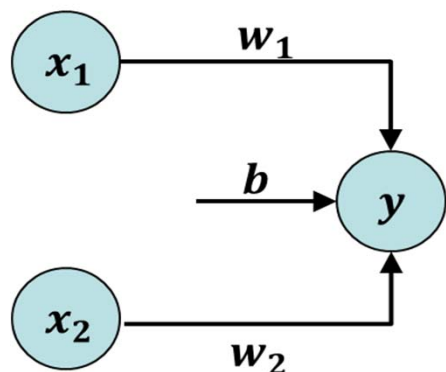
感知机的表达能力：模拟逻辑电路

10

感知机可以模拟或门电路

或门的真值表

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	1



满足或门真值表条件的感知参数取值组合有无数个。例如：

- $(w_1, w_2, z_\tau) = (1, 1, -0.7)$
- $(w_1, w_2, z_\tau) = (1, 1, -0.8)$
- $(w_1, w_2, z_\tau) = (1, 1, -0.9)$

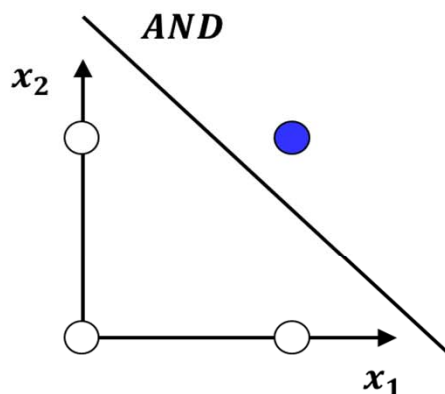
设定感知机的阶跃函数为 $g(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$ 。

感知机的表达能力：模拟逻辑电路

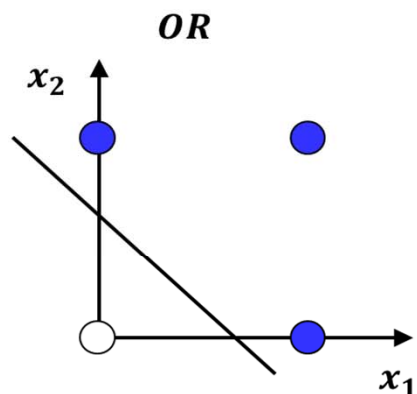
11

感知机无法实现异或门

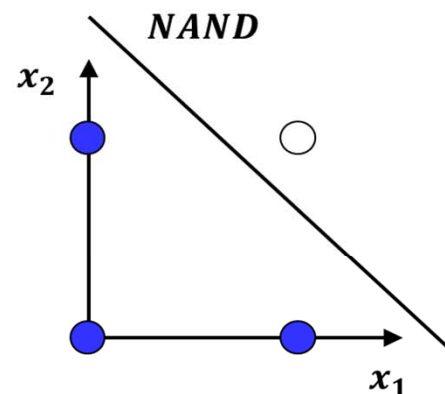
x_1	x_2	y
0	0	0
1	0	0
0	1	0
1	1	1



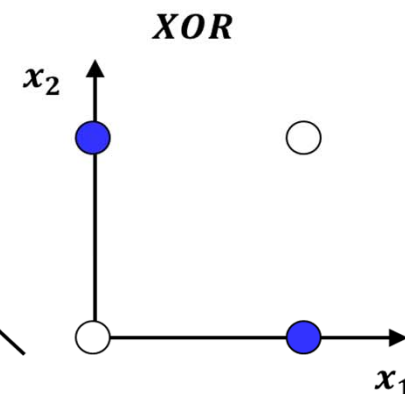
x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	1



x_1	x_2	y
0	0	1
1	0	1
0	1	1
1	1	0



x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0



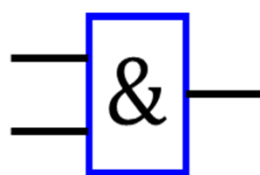
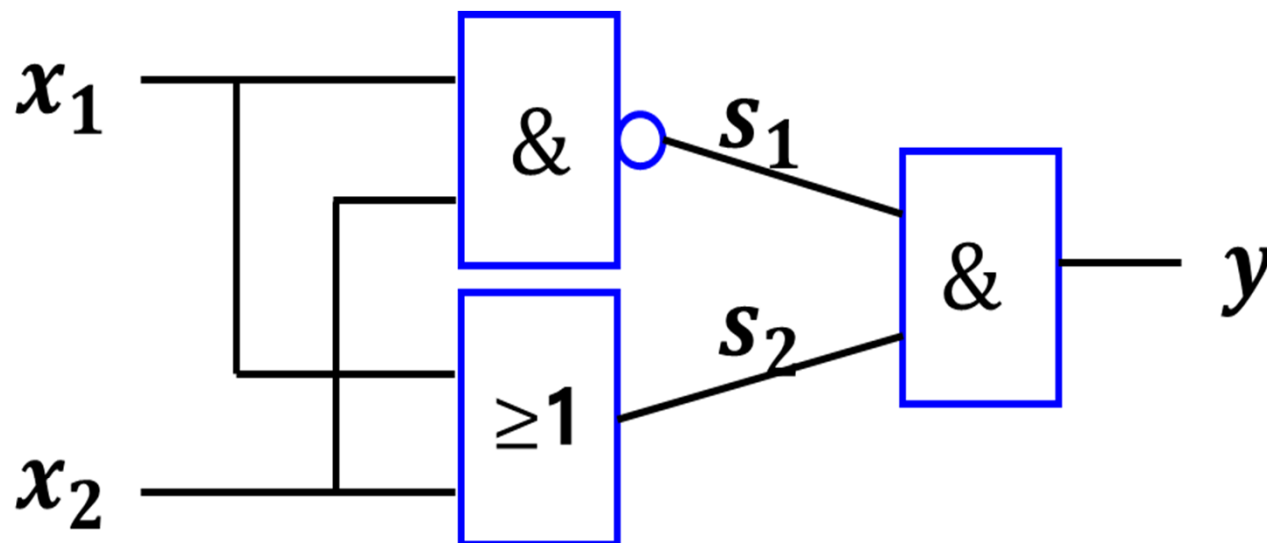
输出：● : 1 ○ : 0

与门、或门、与非门、以及异或门所对应的真值表中的四个点的位置
(蓝色点表示1，空心点表示0)

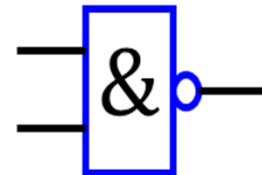
感知机的表达能力：模拟逻辑电路

12

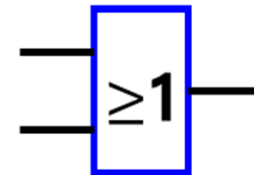
可以利用多层感知机实现异或门



AND



NAND

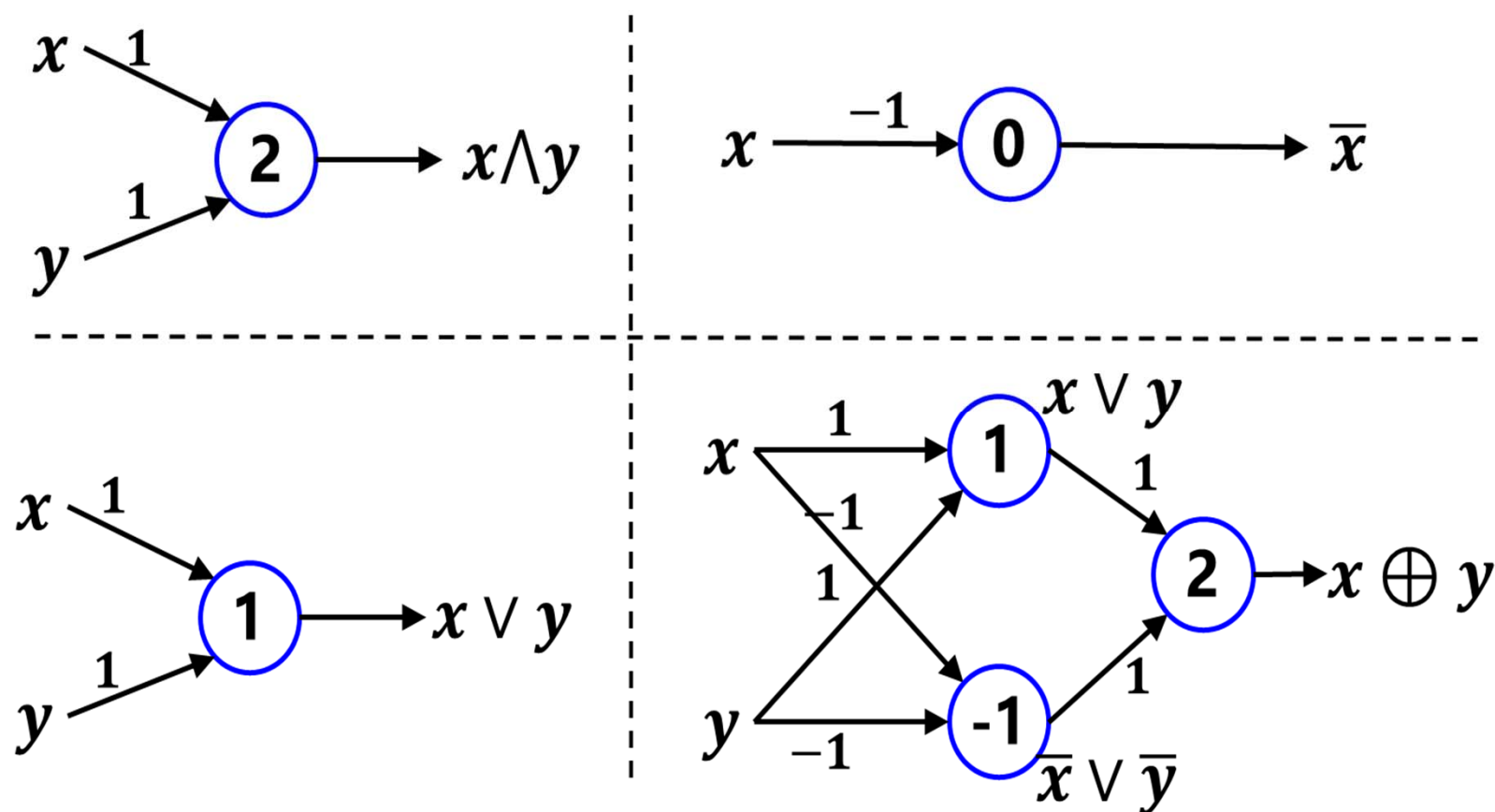


OR

感知机的表达能力：模拟逻辑电路

13

理论上，利用多层感知机构成的神经网络可模拟任意逻辑函数



感知机模型与多层感知机

14

本章内容简介

- 感知机模型简介
- 感知机模拟逻辑电路的能力
- 感知机模型的学习策略
- 多层感知机

感知机模型与多层感知机

15

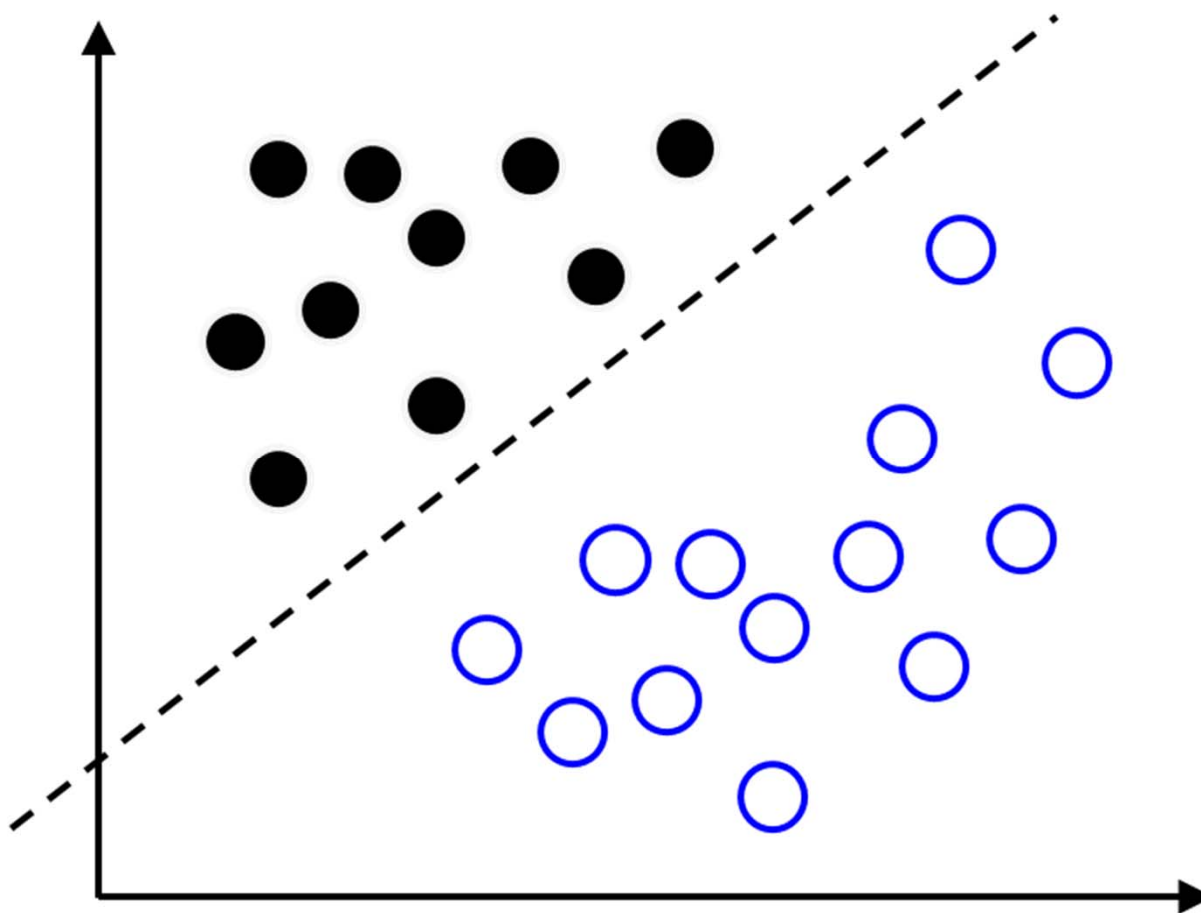
本章内容简介

- 感知机模型简介
- 感知机模拟逻辑电路的能力
- **感知机模型的学习策略**
- 多层感知机

感知机模型的学习策略

16

感知机学习模型要解决的问题是找到二分类问题的超平面



感知机的表达能力：模拟逻辑电路

17

感知机学习模型的目标函数

- **感知机模型的目标函数是所有误分类点到超平面的总距离。**可以证明，误分类点 x_i 到用参数 (w, b) 描述的直线的距离为： $\frac{1}{\|w\|} |w * x_i + b|$ ，其中 $\|w\|$ 是L2范数。
- **因此学习的目标**是找到能使训练集中误分类点 x_i 到超平面（本例中为直线）总距离最小的模型参数 w 和 b 的取值。
- 对于误分类点 (x_i, y_i) 来说： $-y_i(w * x_i + b) > 0$ 总是成立（使用罗森布拉特在其最初的感知机中采用的激活函数，因此 y_i 的取值为+1或者-1），则误分类点 x_i 到超平面的距离可以写为 $-\frac{1}{\|w\|} y_i(w * x_i + b)$ ，从而所有误分类点到超平面的总距离为 $-\frac{1}{\|w\|} \sum_{x_i \in M} y_i(w * x_i + b)$ 。不考虑 $\frac{1}{\|w\|}$ ，得到目标函数为

$$L(w, b) = - \sum_{x_i \in M} y_i(w * x_i + b),$$

其中 M 为误分类点的集合。

感知机的表达能力：模拟逻辑电路

18

基于随机梯度下降的感知机模型方法

- 感知机的上述目标函数是一个凸函数，可以用梯度下降法来求解。即基于梯度下降法找到使目标函数 $L(\theta) = L(w, b) = -\sum_{x_i \in M} y_i(w * x_i + b)$ 变小的模型参数 θ (即 w, b)，因此模型参数 θ 的迭代更新公式为 $\theta := \theta - \eta \nabla_{\theta} L(\theta)$ (η 为学习速率)。

- 由于 $\frac{\partial L(\theta)}{\partial w} = -\sum_{x_i \in M} y_i x_i$ ， $\frac{\partial L(\theta)}{\partial b} = -\sum_{x_i \in M} y_i$ ，因此参数的更新公式分别为

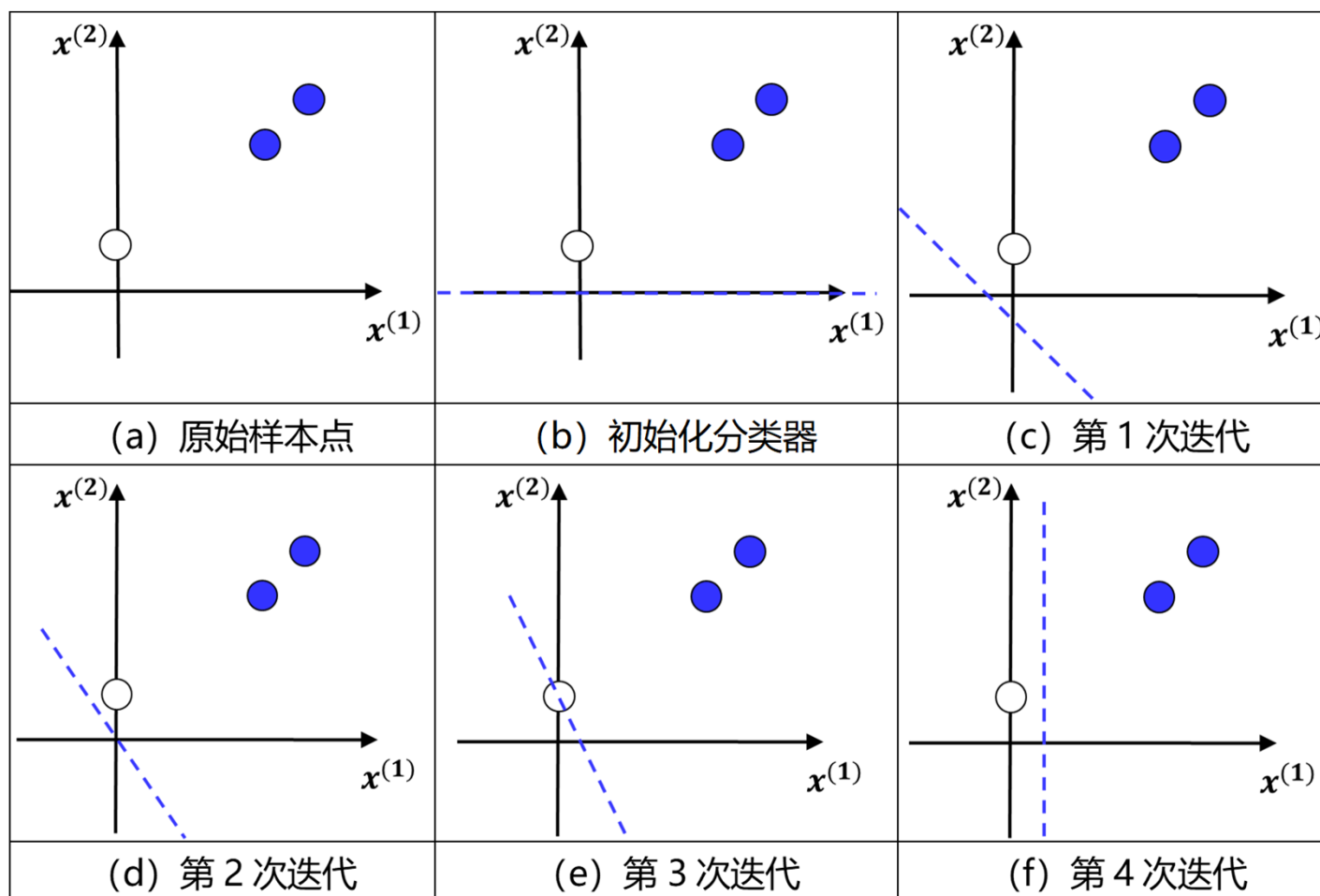
$$w := w + \eta y_i x_i, \quad b := b + \eta y_i$$

- 由于只有误分类样本参与目标函数的优化，所以只能采用随机梯度下降或小批量梯度下降。
- 标准感知机模型使用的是随机梯度下降法，即每次迭代只用一个误分类样本点来更新梯度。

感知机的表达能力：模拟逻辑电路

19

一个二维平面上的感知机迭代实例



感知机模型与多层感知机

20

本章内容简介

- 感知机模型简介
- 感知机模拟逻辑电路的能力
- 感知机模型的学习策略
- 多层感知机

感知机模型与多层感知机

21

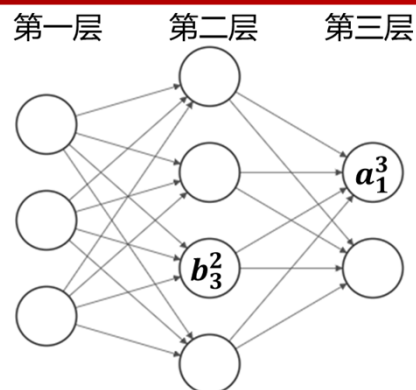
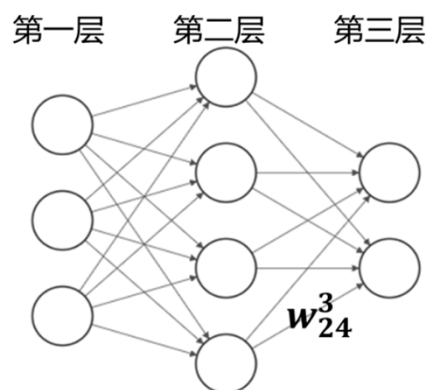
本章内容简介

- 感知机模型简介
- 感知机模拟逻辑电路的能力
- 感知机模型的学习策略
- **多层感知机**

多层感知机与反向传播算法

22

多层感知机的正向传播



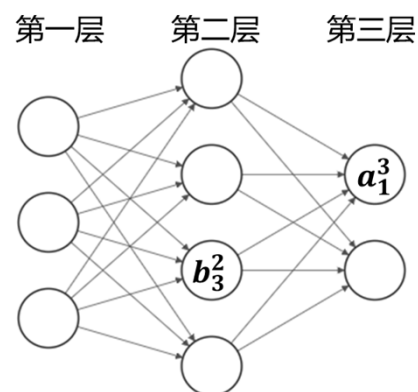
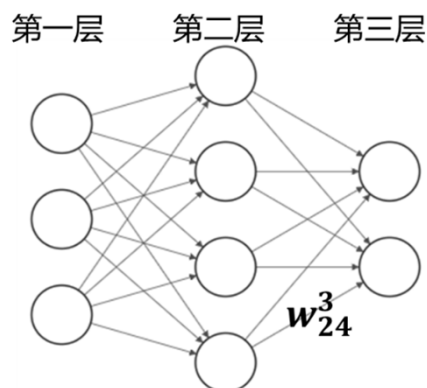
正向传播：是指从输入层输入样本，逐步经各隐藏层计算，最后在输出层经过运算得到样本的类别预测结果的过程。每层计算中有三个重要的概念：

- **权重(weight)**：设 w_{jk}^l 为第 l 层的第 j 个神经元与其前一层即第 $l-1$ 层的第 k 个神经元之间连接的权重。
- **偏置(bias)**：用 b_j^l 来表示第 l 层的第 j 个神经元上的偏置。
- **激活(activation)**：用 a_j^l 来表示第 l 层的第 j 个神经元上的激活值。由于多层感知机的多层结构，如第 l 层不是输出层，则该层激活值将作为下层的输入。

多层感知机与反向传播算法

23

多层感知机的正向传播



- 第 l 层的第 j 个神经元上的激活值 a_j^l 与其前一层，即第 $l-1$ 层的所有神经元 k 的激活值具有如下关系：

$$a_j^l = \sigma(\sum_k w_{jk}^l a_k^{l-1} + b_j^l), \quad \text{其中, } \sigma \text{ 为sigmoid函数。}$$

- 考虑到第 l 层所有神经元的整体，则可写成矩阵形式：

$$a^l = \sigma(w^l a^{l-1} + b^l)$$

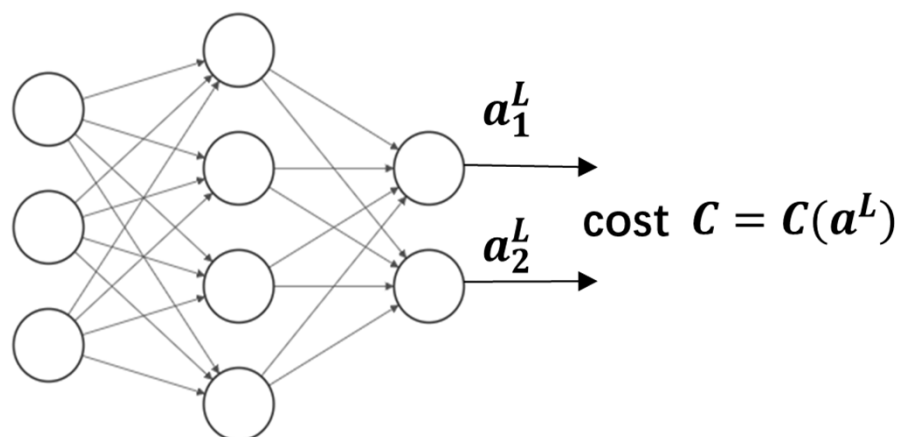
- 将括号内表达式 $w^l a^{l-1} + b^l$ 用**中间变量** z^l 表示，则上式变为

$$a^l = \sigma(z^l), \quad \text{其中, } z^l \text{ 为第 } l \text{ 层的神经元的加权输入。}$$

多层感知机与反向传播算法

24

多层感知机的反向传播算法



- 以上图为例，多层感知机通常以误差平方和作为目标函数，即

$$C = \frac{1}{2n} \sum_x \|y(x) - a^L(x)\|^2, \quad (7-7)$$

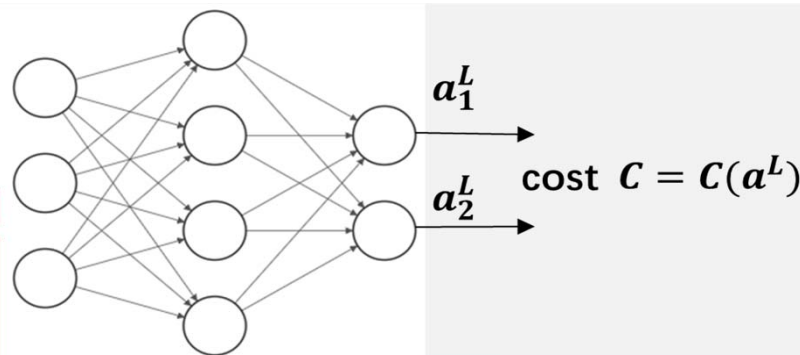
其中，

- $y(x)$ 是输入 x 的真实值输出， n 是训练样本个数， L 是神经网络的层数。
- $a^L(x)$ 是对输入 x 经过网络最后一层处理得到的激活输出。

多层感知机与反向传播算法

25

多层感知机的反向传播算法



- **反向传播过程**是指，按照从输出层到输入层的方向（即反方向），逐层求出上述目标函数对各神经元权值参数的偏导，形成目标函数对这些参数的梯度，作为修正和更新权值的依据。然后正向传播对参数取值进行更新，并利用更新后的参数取值再次进行预测并计算目标函数。上述“正向传播+反向传播”的过程将多次重复，直到网络收敛或者完成了预定的迭代次数。
- 多层感知机的反向传播过程需要计算目标函数对每层中各个权重和偏置参数的偏导 $\frac{\partial C}{\partial w_{jk}^l}$, $\frac{\partial C}{\partial b_j^l}$ 。
- 为便于计算，可以定义一个中间变量用于表示第 l 层所有神经元的误差 δ^l ，同时令 δ_j^l 表示第 l 层的第 j 个神经元的误差，即

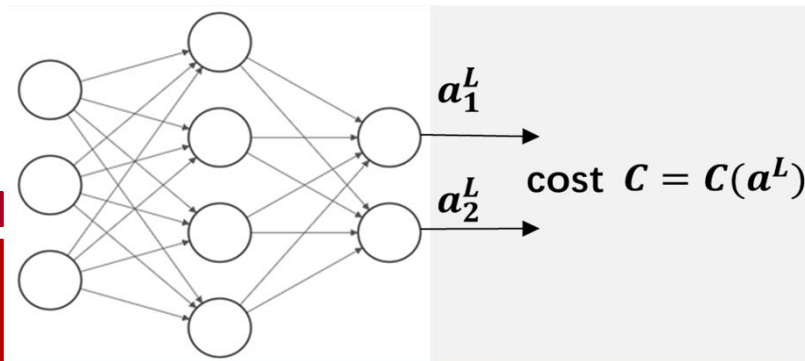
$$\delta_j^l = \frac{\partial C}{\partial z_j^l}$$

，其中 z^l 为第 l 层神经元的加权输入。

多层感知机与反向传播算法

26

多层感知机的反向传播算法



则对于多层感知机的反向传播，按照从输出层到输入的反向计算顺序，可以证明（证明过程略）有如下4个重要公式成立：

1) 输出层误差（第 L 层的第 j 个神经元的误差）计算公式，即 BP1：

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L)。$$

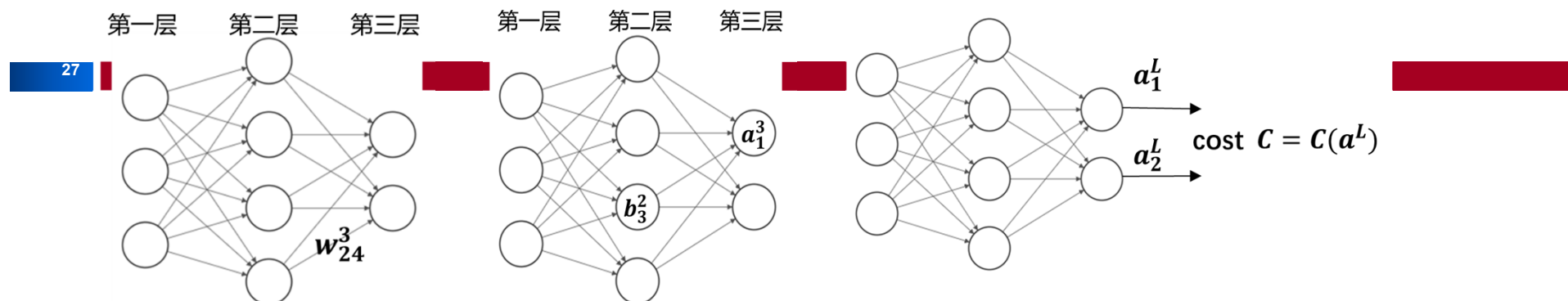
2) 利用误差 δ^{l+1} 计算第 l 层误差 δ^l ，即 BP2：

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l)。$$

3) 目标函数对任意权重参数 w_{jk}^l 的偏导，即 BP3： $\frac{\partial C}{\partial w_{jk}^l} = \delta_j^l a_k^{l-1}。$

4) 损失函数关于任意偏置的偏导数 BP4： $\frac{\partial C}{\partial b_j^l} = \delta_j^l。$

多层感知机的反向传播算法



假设使用的是由 m 个训练样本构成的小批量梯度下降优化方法，则多层感知机的反向传播算法[Michael, 2015]可以描述如下：

1. 输入训练样本集合；
2. 对每个训练样本 x ，完成如下步骤：

a) **正向传播**。对于中间层 $l = 2, 3, \dots, L$ ，逐层计算加权输入 $z^{x,l}$ 和激活 $a^{x,l}$ ：

$$\text{加权输入: } z^{x,l} = w^l a^{x,l-1} + b^l \quad (7-17)$$

$$\text{激活: } a^{x,l} = \sigma(z^{x,l}) \quad (7-18)$$

b) **根据BP1公式计算输出层的误差 $\delta^{x,L}$** ：

$$\delta^{x,L} = \nabla_a C_x \odot \sigma'(z^{x,L}) \quad (7-19)$$

c) **根据BP2公式反向逐层传播误差**。即对于 $l = L - 1, L - 2, \dots, 2$,

$$\delta^{x,l} = ((w^{l+1})^T \delta^{x,l+1}) \odot \sigma'(z^{x,l}) \quad (7-20)$$

多层感知机与反向传播算法

28

多层感知机的反向传播算法

3. 基于上述结果，反向逐层更新权重和偏置参数：即对于 $l = L, L - 1, \dots, 2$ ，：

$$\mathbf{w}^l \rightarrow \mathbf{w}^l - \frac{\eta}{m} \sum_x \delta^{x,l} (\mathbf{a}^{x,l-1})^T \quad (7-21)$$

$$\mathbf{b}^l \rightarrow \mathbf{b}^l - \frac{\eta}{m} \sum_x \delta^{x,l} \quad (7-22)$$

- 以上算法给出了一次正向传播和反向传播的迭代过程，实践中上述过程需要反复计算，直到参数取值收敛或者完成预先定义的迭代次数为止。

感知机模型与多层感知机

29

本章小结

- 感知机模型简介
- 感知机模拟逻辑电路的能力
- 感知机模型的学习策略
- 多层感知机