



# 神经网络

---

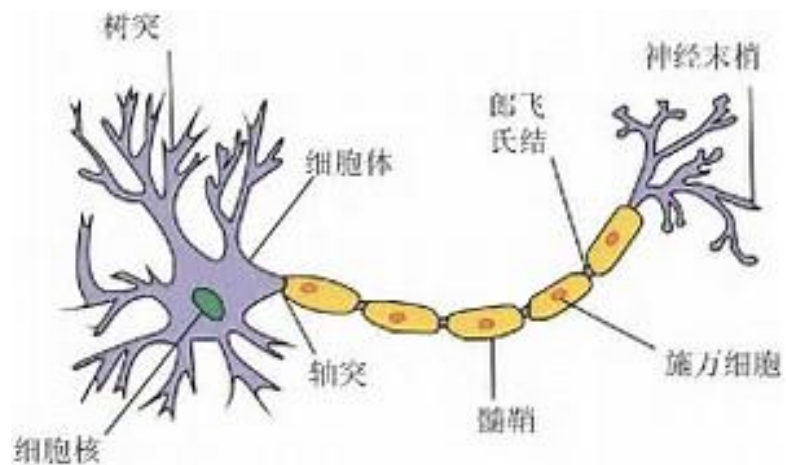
# 感知机

Neural Networks

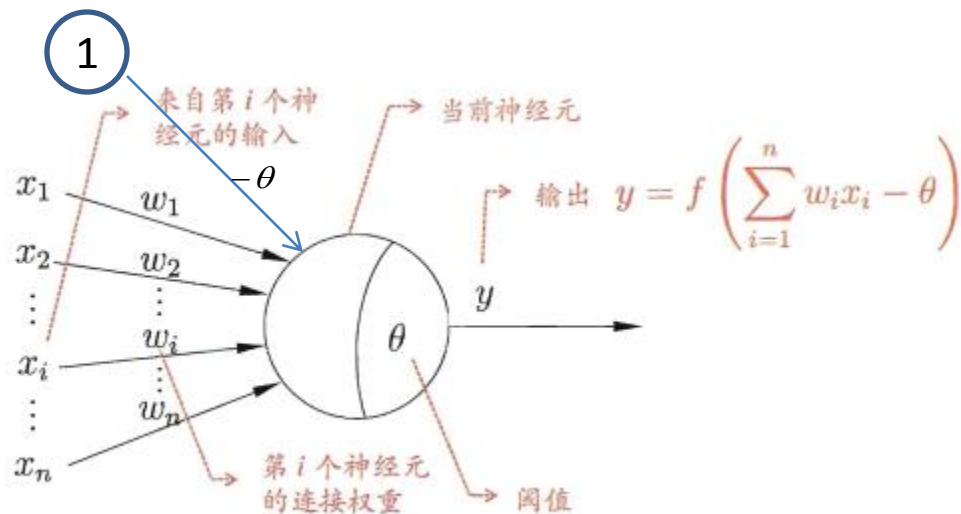
# 目录

- 神经元模型
- 感知机
- 神经网络

# 神经元模型

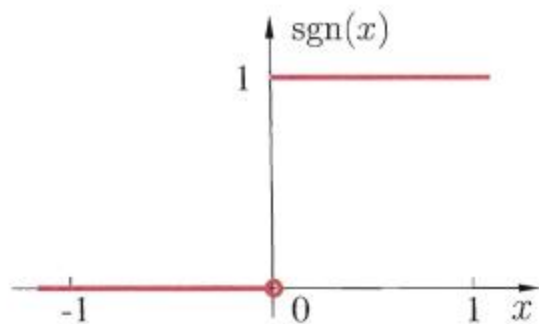


神经元



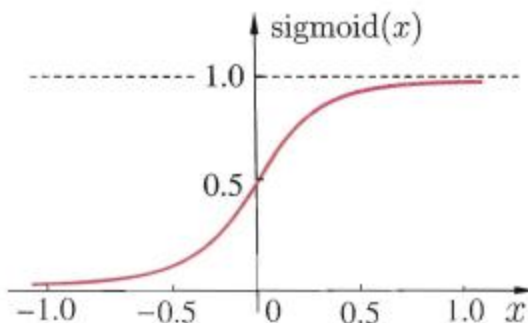
M-P神经元模型

# 激活函数



$$\text{sgn}(x) = \begin{cases} 1, & x \geq 0; \\ 0, & x < 0. \end{cases}$$

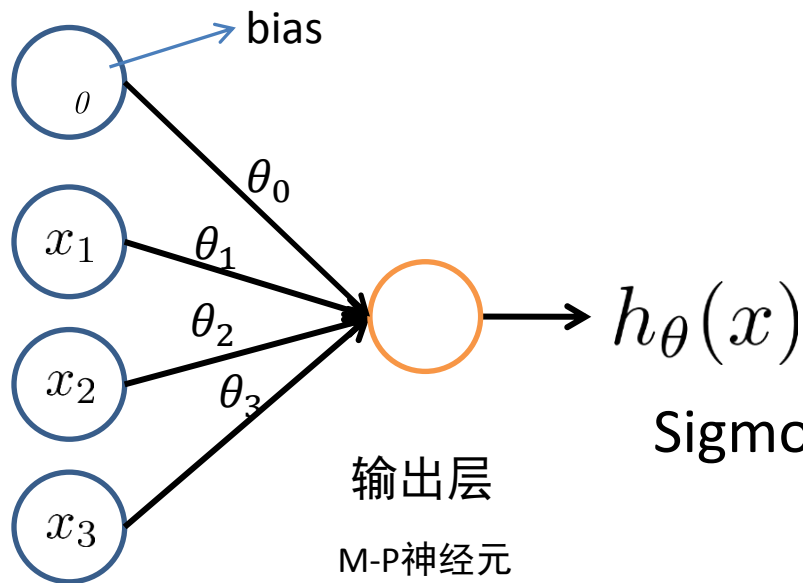
(a) 阶跃函数



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

(b) Sigmoid 函数

# 感知机



输入层

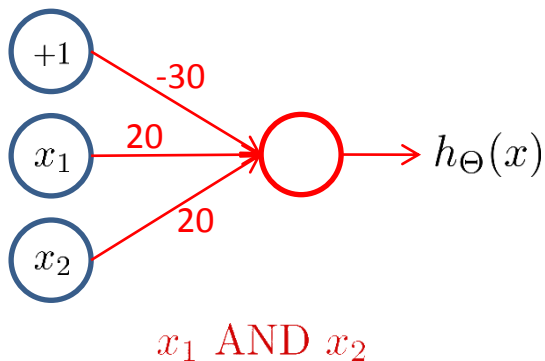
接收外界输入信号

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Sigmoid 激活函数

Output: 
$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

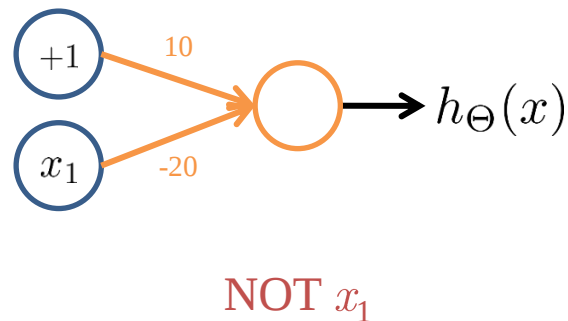
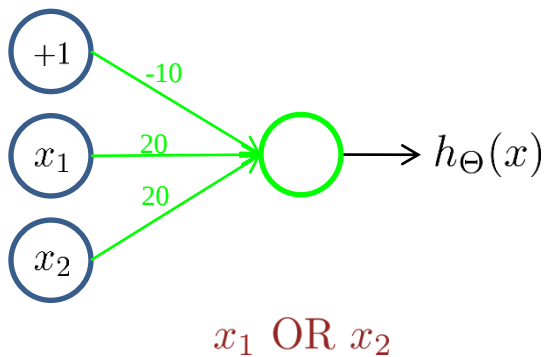
# 感知机实现 “与” “或” “非”



| $x_1$ | $x_2$ | $h_{\Theta}(x)$ |
|-------|-------|-----------------|
| 0     | 0     | $(-30)=0$       |
| 0     | 1     | $(-10)=0$       |
| 1     | 0     | $g(-10)=0$      |
| 1     | 1     | $(10)=1$        |

$$(20x_1 + 20x_2 - 30)$$

# 感知机实现 “与” “或” “非”



# 感知机的学习算法

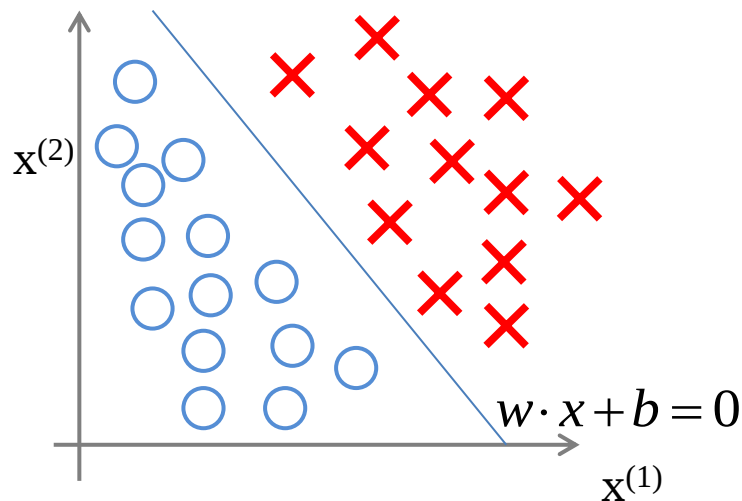
给定一个训练数据集  $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$

其中,  $x_i \in R^n, y_i \in \{+1, -1\}, i = 1, 2, \dots, n$

学习目标: 求得将训练集正例和负例完全分开的超平面 $S$

$$w \cdot x + b = 0$$

即确定参数  $w, b$



# 损失函数

误分类点到超平面S的总距离

任意一点 $x_0$ 到超平面S的距离： $\frac{1}{\|w\|} |w \cdot x_0 + b|$

对误分类点 $(x_0, y_0)$ 来说， $-y_i(w \cdot x_i + b) > 0$

所以误分类点 $x_i$ 到超平面S的距离为  $-\frac{1}{\|w\|} y_i(w \cdot x_i + b)$

假设超平面S的误分类点集合为M，那么所有误分类点到超平面S的总距离为

$$-\frac{1}{\|w\|} \sum_{x_i \in M} y_i(w \cdot x_i + b)$$

不考虑  $\frac{1}{\|w\|}$ ，就得到感知机学习的损失函数。

# 随机梯度下降法

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

首先任取一个超平面  $w_0, b_0$ ，然后用随机梯度下降法不断极小化目标函数，每次随机选取一个误分类点使其梯度下降。

损失函数  $L(w,b)$  的梯度

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

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

随机选取一个误分类点  $(x_i, y_i)$ ，对  $w, b$  进行更新：

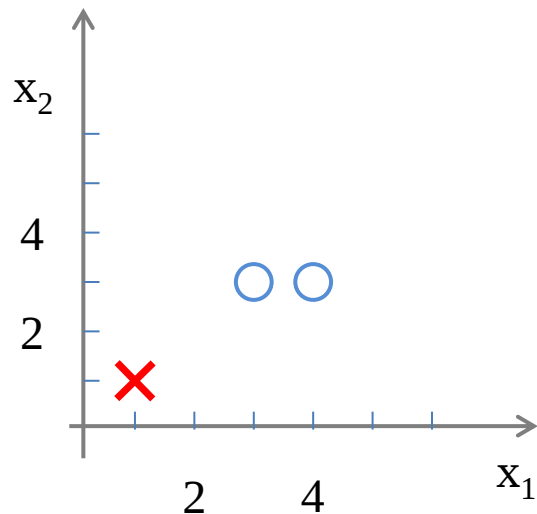
$$w \leftarrow w + \eta y_i x_i$$

$$b \leftarrow b + \eta y_i$$

$\eta$  为学习率。

# 示例

构建最优化问题:  $\min_{w,b} L(w,b) = - \sum_{x_i \in M} y_i (w \cdot x_i + b)$



+ :  $x_1 = (3, 3)^T$ ,  $x_2 = (4, 3)^T$

- :  $x_3 = (1, 1)^T$

1. 取初值  $w_0 = 0, b_0 = 0; \eta = 1$
2. 对  $x_1 = (3, 3)^T, y_1(w_0 \cdot x_1 + b_0) = 0$ , 未能被正确分类, 更新  $w, b$

$$w_1 = w_0 + y_1 x_1 = (3, 3)^T \quad b_1 = b_0 + y_1 = 1$$

得到线性模型  $w_1 \cdot x + b_1 = 3x^{(1)} + 3x^{(2)} + 1$

3. 对  $x_1, x_2$ ,  $y_i(w_1 \cdot x_i + b_1) > 0$ , 被正确分类;  
对  $x_3 = (1, 1)^T$ ,  $y_3(w_1 \cdot x_3 + b_1) < 0$ , 被误分类。

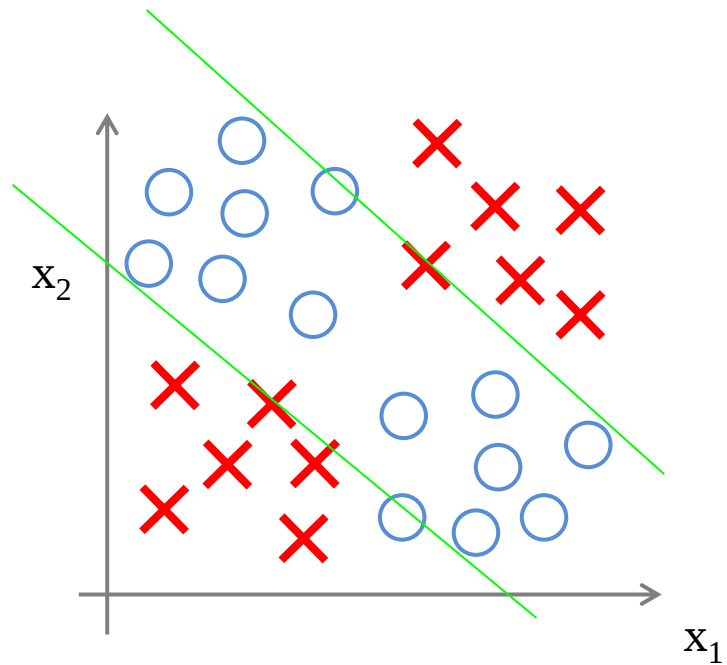
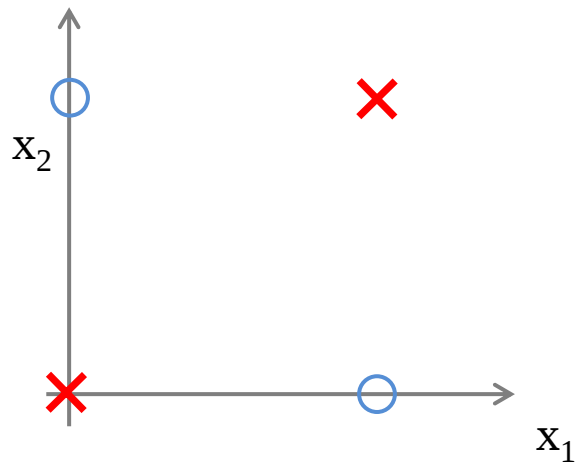
$$w_2 = w_1 + y_3 x_3 = (2, 2)^T, b_2 = b_1 + y_3 = 0$$

得到线性模型  $w_2 \cdot x + b_2 = 2x^{(1)} + 2x^{(2)}$

如此继续下去, 直到  $w_7 = (1, 1)^T, b_7 = -3$   $w_7 \cdot x + b_7 = x^{(1)} + x^{(2)} - 3$

对所有数据点  $y_i(w_7 \cdot x + b_7) > 0$ , 没有误分类点。

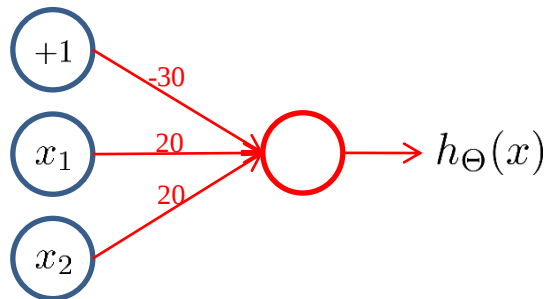
# 线性不可分问题



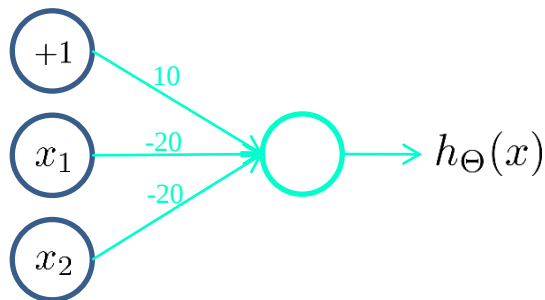
# 线性不可分问题

$x_1$  XNOR  $x_2$

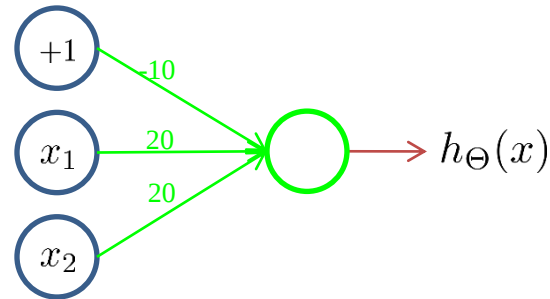
$(x_1 \& x_2) | (\neg x_1 \& \neg x_2)$



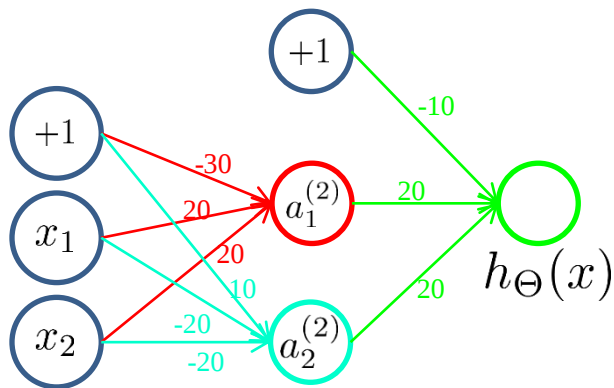
$x_1$  AND  $x_2$



$(\text{NOT } x_1) \text{ AND } (\text{NOT } x_2)$

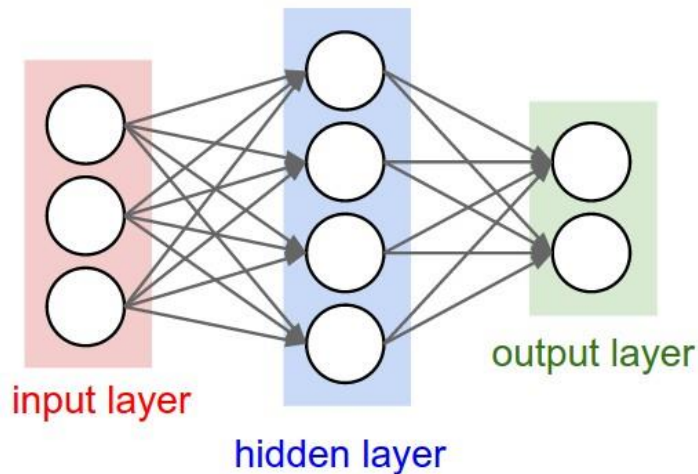


$x_1$  OR  $x_2$



| $x_1$ | $x_2$ | $a_1^{(2)}$ | $a_2^{(2)}$ | $h_{\Theta}(x)$ |
|-------|-------|-------------|-------------|-----------------|
| 0     | 0     | 0           | 1           | 1               |
| 0     | 1     | 0           | 0           | 0               |
| 1     | 0     | 0           | 0           | 0               |
| 1     | 1     | 1           | 0           | 1               |

# 神经网络



## 单隐层神经网络

- 神经元按照层来布局。最左边的层叫做**输入层**，负责接收输入数据；最右边的层叫**输出层**，可以从这层获取神经网络的输出数据。输入层和输出层之间的层叫做**隐层**，因为它们对于外部来说是不可见的。
- 同一层的神经元之间没有连接。
- 每个连接都有一个**权值**。