感知机
（Perceptron）

程煦

xcheng8@njust.edu.cn

计算机科学与工程学院

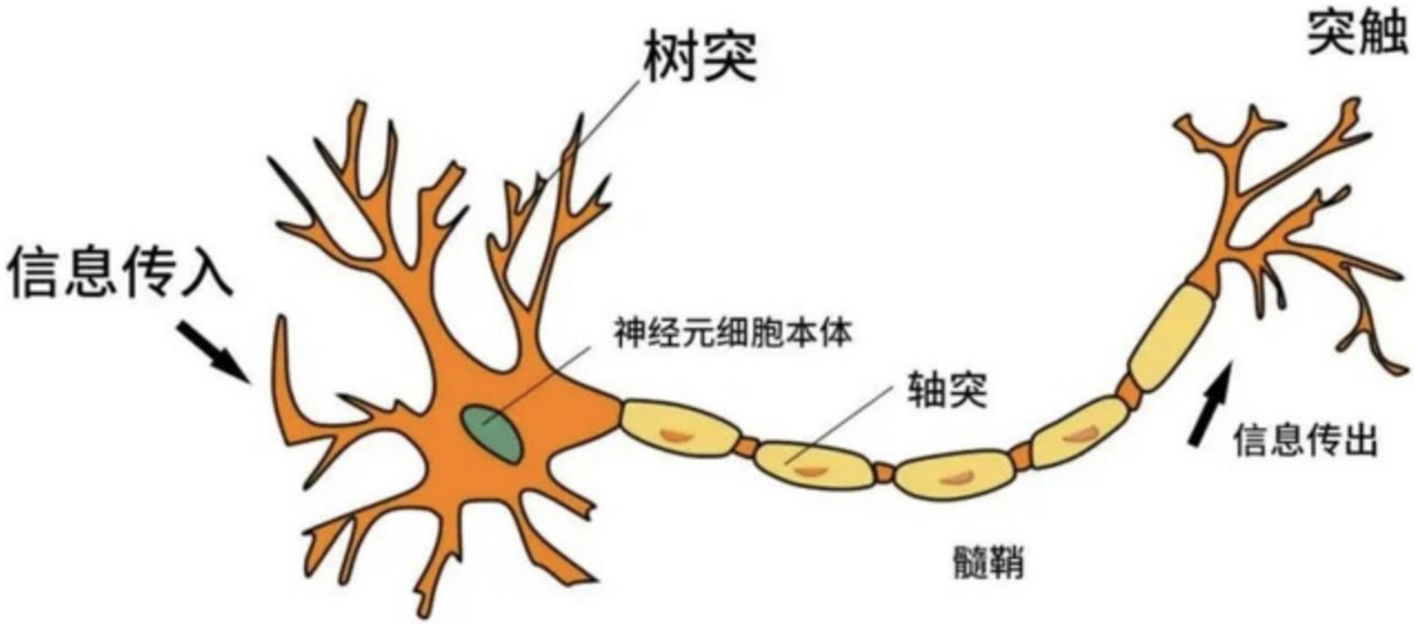
Outline

- 感知机模型
- 感知机学习策略
- 感知机学习算法
- 多层感知机与神经网络

感知机：提出背景

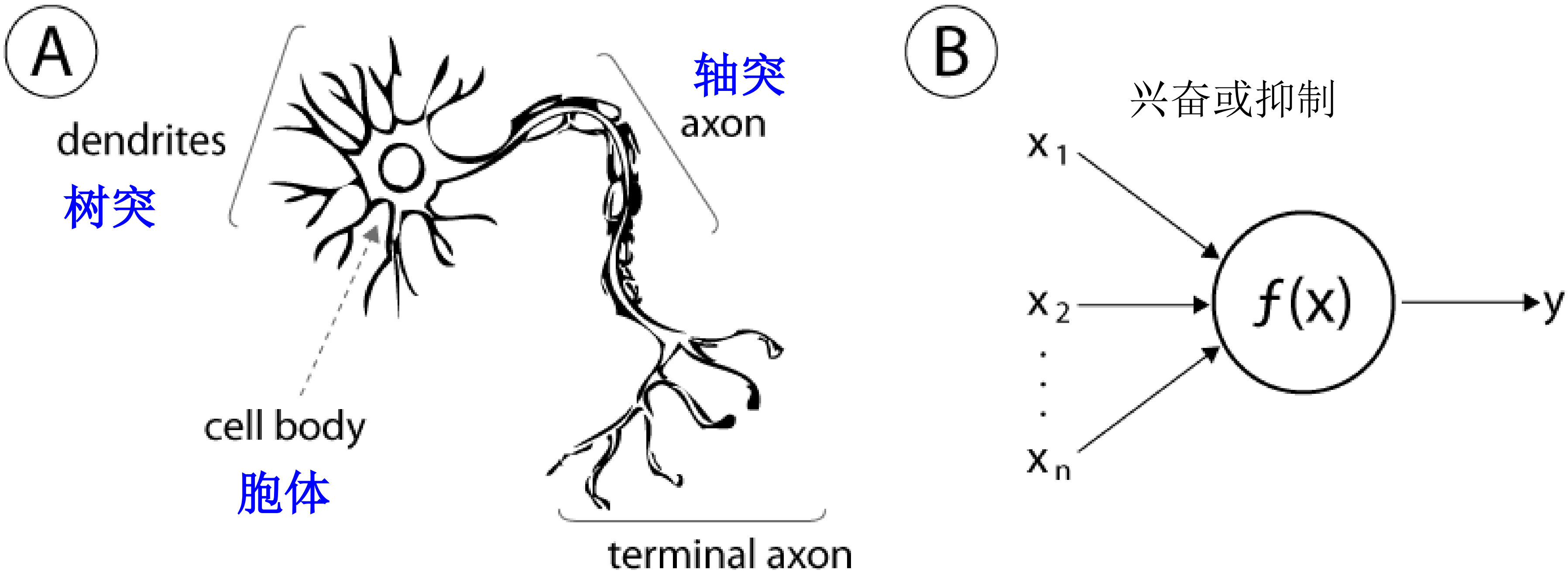
- 1958年由Frank Rosenblatt在康奈尔航空实验室提出，这是最早的人工神经网络模型之一，被认为是**现代神经网络的起点**，为后来的**深度学习**奠定了基础。
- **早期模式识别需求：**当时的计算机主要用于数值计算，而科学家希望机器能像人类一样识别**图片、字符、声音**等模式。
- 神经科学的启发，感知机模型基于**神经元的工作原理**，模拟大脑神经元的输入、加权求和、激活函数等计算方式
- 感知机是一种线性分类器。

感知机基于神经元的工作原理



- **神经元**（Neuron）是大脑的基本计算单元，由树突（Dendrites）、胞体（Soma）、轴突（Axon）组成
- **树突**接收其他神经元的信号，并将其传输到**胞体**。
- 当输入信号的总和超过某个**阈值**时，神经元会被激活，并通过**轴突**向下一个神经元传递信号。

感知机基于神经元的工作原理



感知机是**一个简单的线性分类器**，核心思想是：

1.输入信号（对应神经元的树突接收信号）。

2.加权求和（对应神经元对输入信号进行加权累加）。

3.激活函数（对应神经元超过阈值时被激活）。

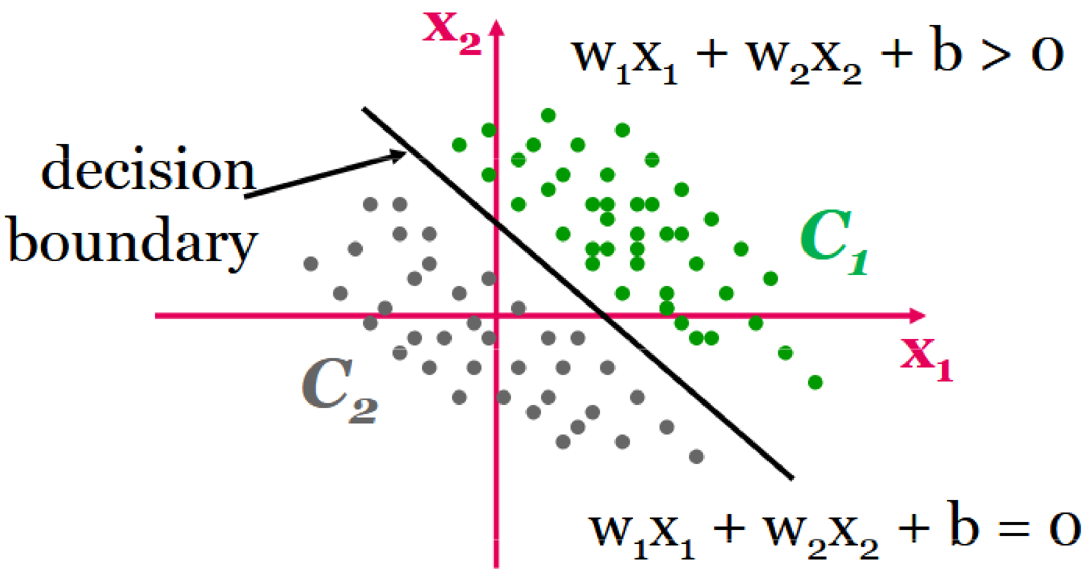
4.输出结果（对应神经元向下游神经元传递信号）。

感知机模型

- 是一种**线性分类模型**，用于解决**线性可分**的二分类问题

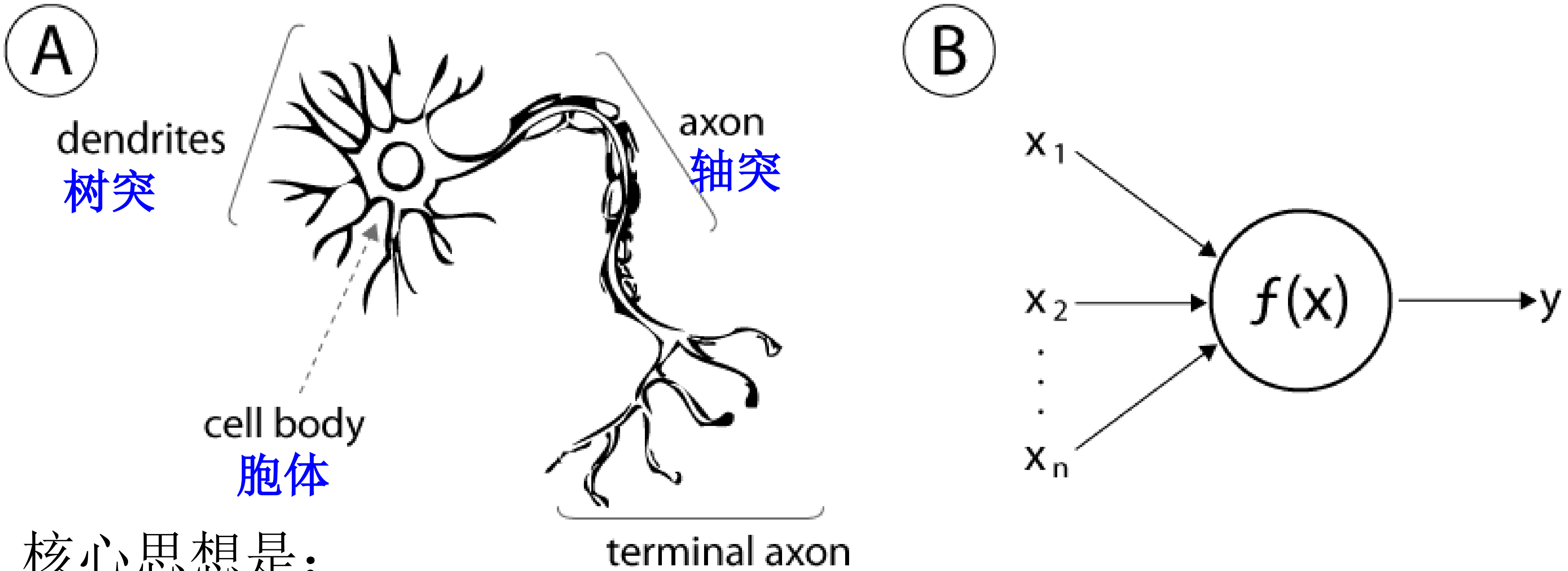
■ 在二维就是找到一条直线，在三维或者更高维就是找到一个超平面，将两个类别的数据分开。

■ 找不到这条直线或超平面就是类别线性不可分，感知机不适合该数据分类。使用感知机最大前提就是数据线性可分。



感知机模型

- 是一种**线性分类模型**，用于解决**线性可分**的二分类问题



核心思想是：

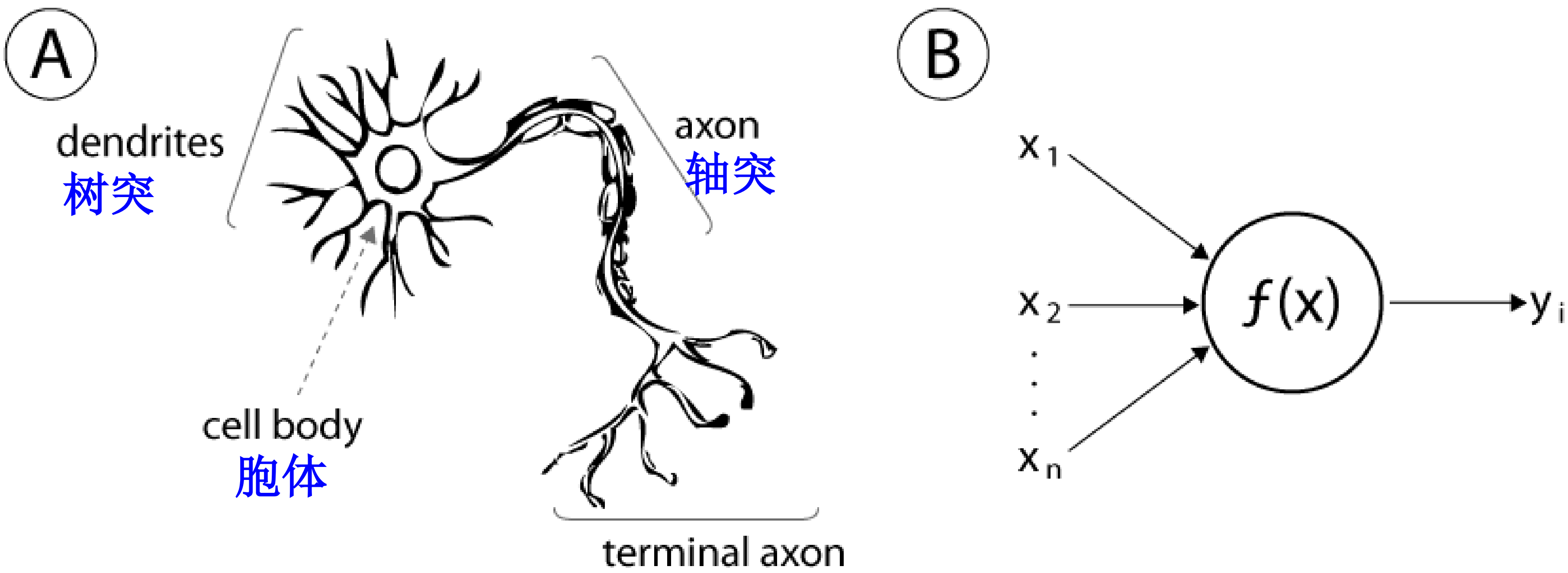
1.输入信号（对应神经元的树突接收信号）。

2.加权求和（对应神经元对输入信号进行加权累加）。

- **输入特征**：给定输入数据 $\boldsymbol{x}$ 以及权重 $\boldsymbol{w}$ ，计算加权求和$\boldsymbol{w}^T\boldsymbol{x}+b$, $\boldsymbol{x}^T=[x_1,x_2,\cdots,x_{\mathrm{d}}]$, $\boldsymbol{w}^T=[w_1,w_2,w_3,\cdots,w_d]$

感知机模型

- 是一种**线性分类模型**，用于解决**线性可分**的二分类问题



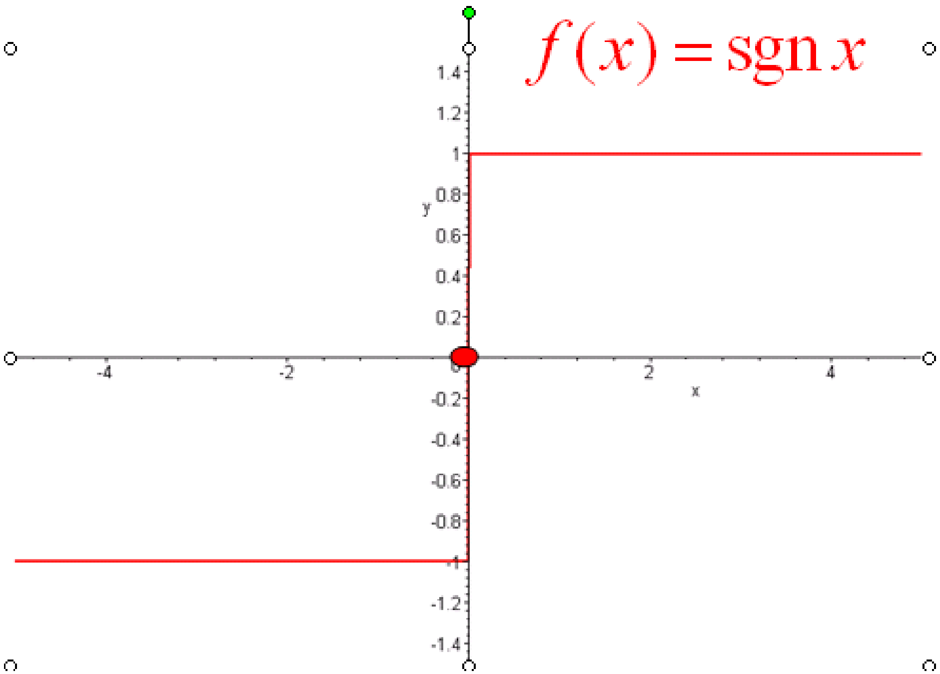
3. 激活函数（对应神经元超过阈值时被激活）。

4. 输出结果（对应神经元向下游神经元传递信号）。

- **激活函数：**使用阶跃函数sign() 将加权求和结果转换为 -1 或 1，实现二分类$\hat{y} = \text{sign}\left(\boldsymbol{w}^T\boldsymbol{x} + b\right)$。

感知机模型

- **激活函数：** 使用阶跃函数sign() 将加权求和结果转换为 -1 或 1， 实现二分类$\hat{y} = \text{sign}\,(\boldsymbol{w}^T\boldsymbol{x} + b)$。



即 x>0,sgnx=1 → 分为正类

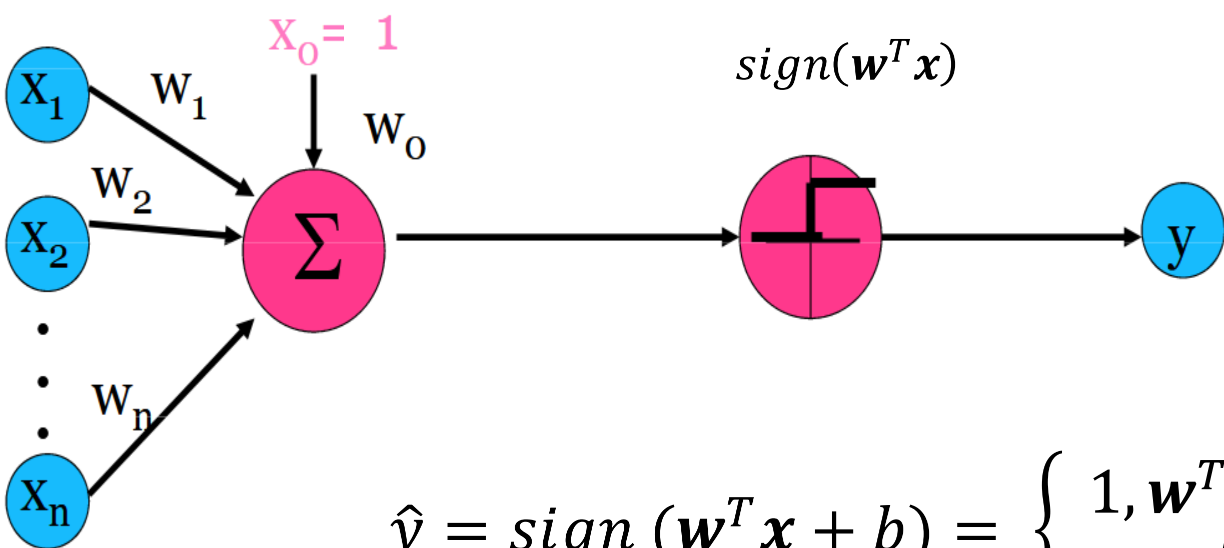
x=0,sgnx= 0 → $\boldsymbol{w}^T\boldsymbol{x} + b = 0$ 是决策边界

x<0,sgnx=-1 → 分为负类

感知机模型

- 是一种**线性分类模型**，用于解决**线性可分**的二分类问题

1. **输入特征**：给定输入数据 $\boldsymbol{x}$ 以及权重 $\boldsymbol{w}$ ，计算加权求和 $\boldsymbol{w}^T\boldsymbol{x}+b, \boldsymbol{x}^T=[x_1,x_2,\cdots,x_\mathrm{d}], \boldsymbol{w}^T=[w_1,w_2,w_3,\cdots,w_d]$
2. **激活函数**：使用阶跃函数sign() 将加权求和结果转换为 -1 或 1，实现二分类$\hat{y}=\mathrm{sign}\,(\boldsymbol{w}^T\boldsymbol{x}+b)$。



$$\hat{y}=sign\,(\boldsymbol{w}^T\boldsymbol{x}+b)=\begin{cases}1, \boldsymbol{w}^T\boldsymbol{x}+b>0\\-1, \boldsymbol{w}^T\boldsymbol{x}+b<0\end{cases}$$

感知机损失函数

- 感知机的目标是找到一个线性超平面将数据分类，即

$$\boldsymbol{w}^T\boldsymbol{x} + \mathrm{b} = 0$$

- 因为：感知机对一个输入样本$\boldsymbol{x}_i$进行分类的决策规则是

$$\hat{y}_i = \mathrm{sign}(\boldsymbol{w}^T\boldsymbol{x} + b)$$

  - $\hat{y}_i = 1$ 表示被分类到正类，
  - $\hat{y}_i = -1$ 表示被分类到负类。
  - $\boldsymbol{w}^T\boldsymbol{x} + b = 0$: 模型无法区分，数据点正好位于分类边界上

- 决策边界定义：**使得模型预测发生变化的分界面**，即使得 $\hat{y}_i$ 由 +1 变为 –1 的点的集合。在感知机中，这些点满足$\boldsymbol{w}^T\boldsymbol{x} + b = 0$

该方程描述了一个超平面：

- **在二维空间**，它是一直线：$w_1x_1 + w_2x_2 + b = 0$
- **在三维空间**，它是一个平面
- **在更高维空间**，它是一个超平面

这个超平面将数据分为两类，不同类别的数据点落在超平面的两侧。因此，它是感知机的**决策边界**。

感知机损失函数

- 为了正确分类，理想情况下应该满足

$$y_i(\boldsymbol{w}^T\boldsymbol{x}+b)>0$$

$y_i \in \{-1,1\}$ 是样本真实类别标签。

- 损失函数：只对错误分类的样本$y_i(\boldsymbol{w}^T\boldsymbol{x}+b) \le 0$进行优化
  - 给定训练集：$\{(\boldsymbol{x}_i,y_i)\}$，$i \in \{1,2,\cdots N\}$

$$\min_{\boldsymbol{w}} \quad L(\boldsymbol{w})=-\sum_{\boldsymbol{x}_i\in\mathcal{M}} y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right)$$

➢ $\mathcal{M}$为分类错误样本的集合：$\mathcal{M}=\{\boldsymbol{x}_i|y_i(\boldsymbol{w}^T\boldsymbol{x}+b)\le 0\}$

感知机损失函数

- 损失函数：只对错误分类的样本$y_i(\boldsymbol{w}^T\boldsymbol{x}+b)\leq 0$进行优化

$$\min_{\boldsymbol{w},b} \qquad L(\boldsymbol{w}) = -\sum_{\boldsymbol{x}_i\in\mathcal{M}} y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right)$$

  - **损失函数的motivation:** 如果样本 $\boldsymbol{x}_i$ 被误分类，意味着 $y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)\leq 0$。直觉上，我们希望增大 $y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)$，使其变为正数，即调整$\boldsymbol{w}$使得误分类样本向正确的决策边界靠近。

  - **损失函数的物理意义:** 如果$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)$是负的，那么损失$L(\boldsymbol{w})$会较大。目标是让损失变小，即让所有$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)$都变为正值。

感知机损失函数

- 损失函数：只对错误分类的样本$y_i(\boldsymbol{w}^T\boldsymbol{x}+b)\le 0$进行优化

$$\min_{\boldsymbol{w},b} \quad L(\boldsymbol{w}) = -\sum_{\boldsymbol{x}_i\in\mathcal{M}} y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right)$$

$$\Rightarrow \quad \min_{\boldsymbol{w},b} \quad L(\boldsymbol{w}) = \sum_{i=1}^{N} \max(0, -y_i\left(\boldsymbol{w}^T\boldsymbol{x}_i+b\right))$$

- 若样本被正确分类（$y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)>0$），损失为0
- 若样本被错误分类，损失为$-y_i(\boldsymbol{w}^T\boldsymbol{x}_i+b)$

梯度下降更新感知机参数

- 目标： $\min_{\boldsymbol{w},b} L(\boldsymbol{w}) = \sum_{i=1}^{N} \max(0, -y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b))$
- 梯度下降
  - 若样本被错误分类，损失为 $-y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b)$

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i \boldsymbol{x}_i$$

$$\frac{\partial L(\boldsymbol{w})}{\partial b} = -\sum_{\boldsymbol{x}_i \in \mathcal{M}} y_i$$

  - 若样本被正确分类，损失为0

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}} = 0 \qquad \frac{\partial L(\boldsymbol{w})}{\partial b} = 0$$

梯度下降更新感知机参数

- 目标： $\min_{\boldsymbol{w},b} L(\boldsymbol{w}) = \sum_{i=1}^{N} \max(0, \ -y_i\,(\boldsymbol{w}^T\boldsymbol{x}_i + b))$

- 梯度下降

（1）选取初值 w_0, b_0

（2）在训练集中选取数据 (x_i, y_i)

（3）如果 $y_i(w \cdot x_i + b) \leqslant 0$

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha y_i \boldsymbol{x}_i$$

$$b \leftarrow b + \alpha y_i$$

（4）转至（2），直至训练集中没有误分类点

感知机模型的训练原理

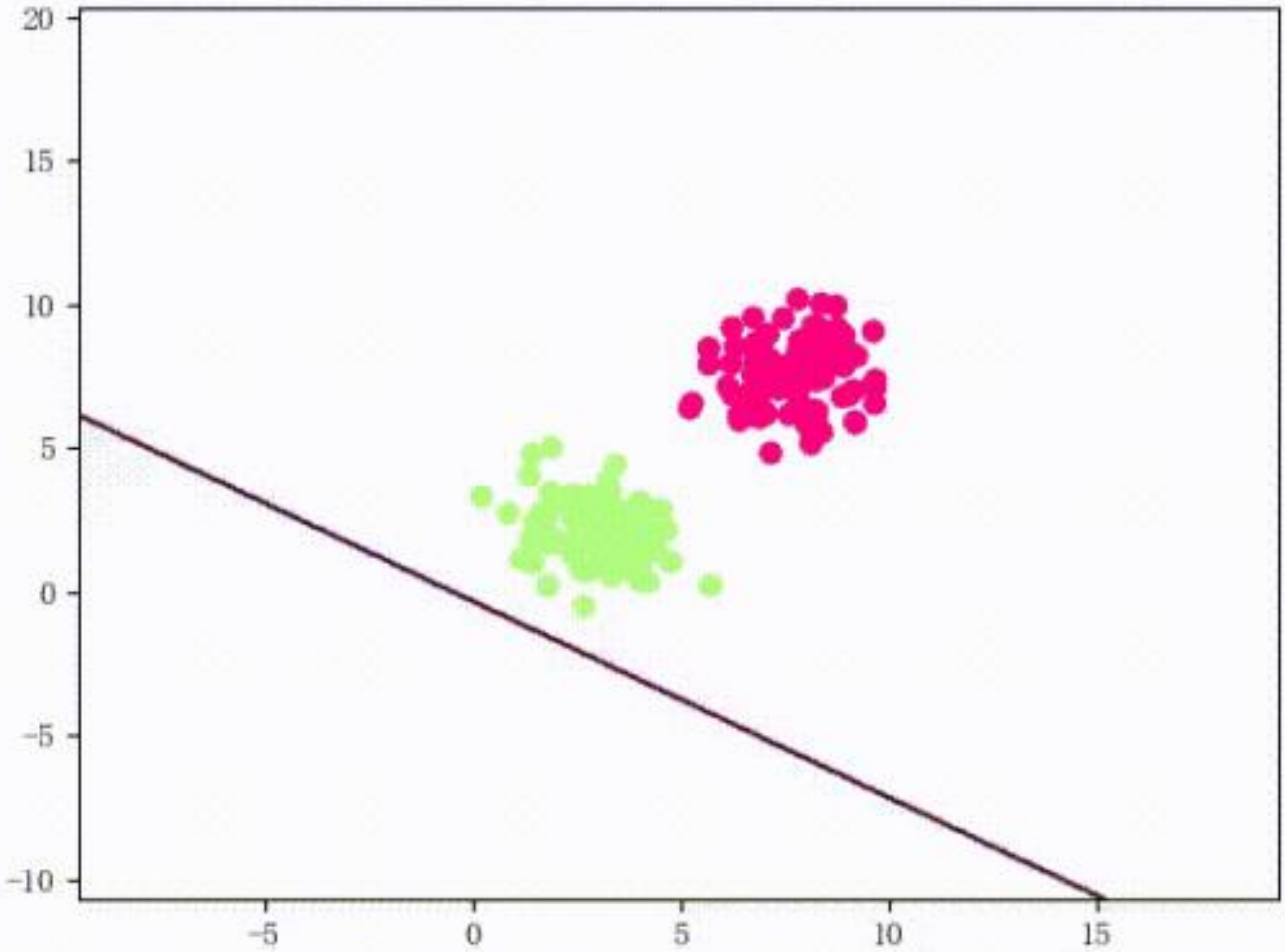
- 基本原理就是逐点修正

（1）在超平面上随意取一个分类面，统计分类错误的点；

（2）随机对某个错误点进行修正，即变换直线的位置（w和b），使该错误点得以修正；

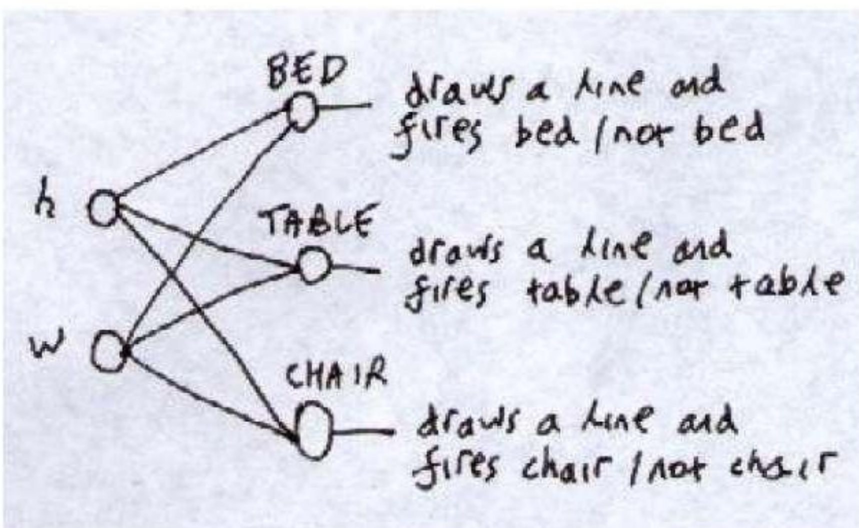
（3）再随机选择一个错误点进行纠正，分类面不断变化，直到所有的点都完全分类正确了，就得到了最佳的分类面。

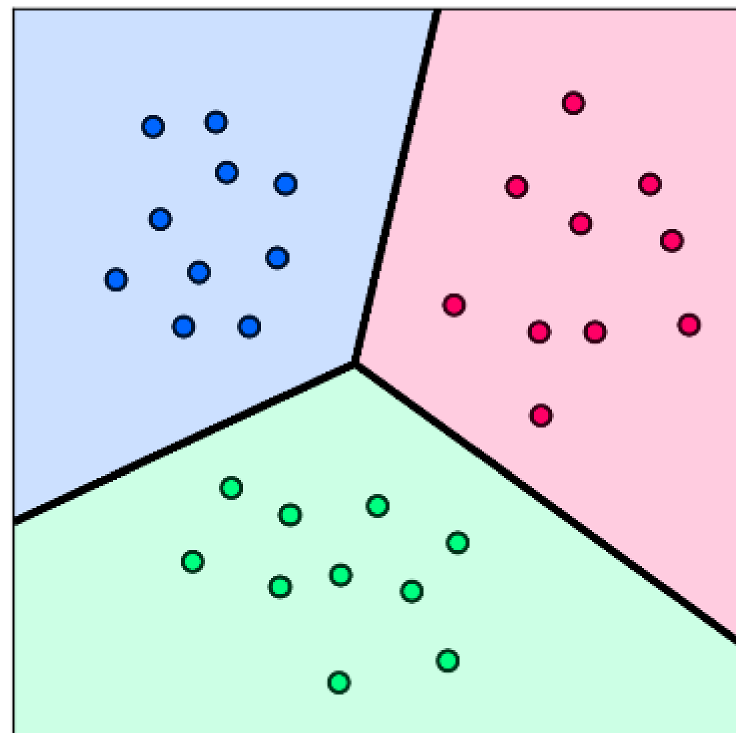
感知机原理



多分类感知机

- 多类感知机是对感知机的扩展，用于解决多类分类问题。
- 广泛应用在图像分类中、自然语言处理（例如，在情感分析中可以应用多分类感知机将输入的文本映射到“正面”，“负面”和“中立”）等问题上。





多分类感知机模型

- **核心思想**：通过一对多方法（one-vs-rest）应用感知器模型

在一对多方法中，针对每个类别 j 训练一个二分类器，该分类器的任务是区分该类别和所有其他类别。也就是说，分类器的输出是：

- 对于类别 j：$y_j = 1$
- 对于其他类别 $k \neq j$：$y_k = -1$

假设有 C 个类别，则将训练 C 个感知机，每个感知机的目标是区分类别 j 和其他类别。最终的预测是通过选取预测概率最大的类别来确定。

- **多分类感知机模型**：

  - 对于每个类别j, 训练一个感知机去区分是类别j vs 非类别j

$$f_j = \mathrm{sign}\left(\boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j\right)$$

多分类感知机模型

- 对于每个类别j, 训练一个感知机区分是类别j vs 非类别j

$$f_j = \text{sign}\left(\boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j\right)$$

- 最终输出C^*：将样本分类到得分最大的类别上

$$C^* = \arg\max_{j=1,\dots,C} \boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j$$

- 损失函数：对于每个样本$\boldsymbol{x}_i$，感知机的目标是尽量让正确类别y_i的得分最大，同时让其他类别的得分较小。

$$\min \quad L(\boldsymbol{w}) = \sum_{i=1}^{N} \sum_{\mathrm{j} \neq y_i}^{N} \max\left(0, \left(\boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j\right) - \left(\boldsymbol{w}_{y_i}{}^T \boldsymbol{x}_i + b_{y_i}\right)\right)$$

多分类感知机模型

- 损失函数：对于每个样本$\boldsymbol{x}_i$，感知机的目标是尽量让正确类别y_i的得分最大，同时让其他类别的得分较小。

$$\min \quad L(\boldsymbol{w}) = \sum_{i=1}^{N} \sum_{j \neq y_i}^{N} \max(0, \left(\boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j\right) - \left(\boldsymbol{w}_{y_i}{}^T \boldsymbol{x}_i + b_{y_i}\right))$$

✓ max(0,·)：如果$\boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j$大于正确类别$\boldsymbol{w}_{y_i}{}^T \boldsymbol{x}_i + b_{y_i}$，则会增加损失值，反之不增加损失。

✓ **最大化正确类别的得分：** 确保每个样本的正确类别得分大于其他类别的得分。

✓ **最小化误分类的得分差距：** 通过惩罚那些得分差距较小的错误分类，鼓励模型进行更准确的分类。

梯度下降更新多分类感知机模型

- 优化目标

$$\min L(\boldsymbol{w}) = \sum_{i=1}^{N} \sum_{j \neq y_i}^{N} \max(0, \left(\boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j\right) - \left(\boldsymbol{w}_{y_i}{}^T \boldsymbol{x}_i + b_{y_i}\right))$$

- 计算梯度，对于$\forall \mathrm{j} \neq y_i$

$$\frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}_j} = \begin{cases} \boldsymbol{x}_i, & \text{if} \left(\boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j\right) - \left(\boldsymbol{w}_{y_i}{}^T \boldsymbol{x}_i + b_{y_i}\right) > 0 \\ 0, otherwise & \end{cases}$$

具体来说，如果 $j \neq y_i$ 且 $\mathbf{w}_j^T \mathbf{x}_i + b_j - (\mathbf{w}_{y_i}^T \mathbf{x}_i + b_{y_i}) > 0$，那么就更新权重 $\mathbf{w}_j$ 使得 $\mathbf{w}_j$ 更接近样本 $\mathbf{x}_i$ 以减少损失。

梯度下降更新多分类感知机模型

- 优化目标

$$\min L(\boldsymbol{w}) = \sum_{i=1}^{N} \sum_{\mathrm{j} \neq y_i}^{N} \max(0, \left(\boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j\right) - \left(\boldsymbol{w}_{y_i}{}^T \boldsymbol{x}_i + b_{y_i}\right))$$

- 计算梯度，对于$\forall \mathrm{j} \neq y_i$

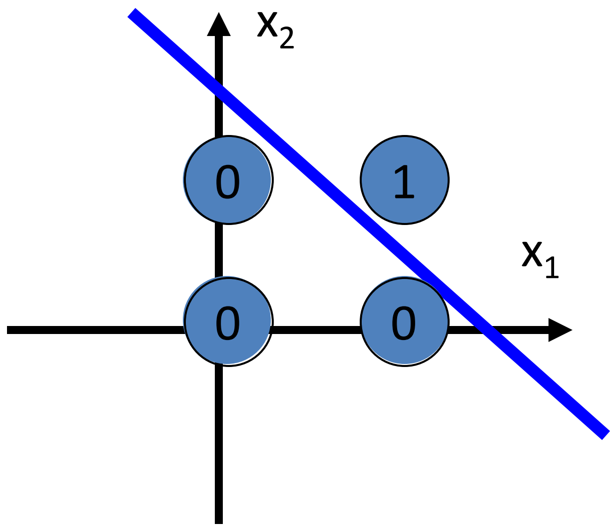
$$\frac{\partial L(\boldsymbol{w})}{\partial b_j} = \begin{cases} 1, & \mathrm{if}\left(\boldsymbol{w}_j{}^T \boldsymbol{x}_i + b_j\right) - \left(\boldsymbol{w}_{y_i}{}^T \boldsymbol{x}_i + b_{y_i}\right) > 0 \\ & 0, otherwise \end{cases}$$

- 梯度下降，对于$\forall \mathrm{j} \neq y_i$

$$\boldsymbol{w}_j \leftarrow \boldsymbol{w}_j - \alpha \frac{\partial L(\boldsymbol{w})}{\partial \boldsymbol{w}_j} \qquad b_j \leftarrow b_j - \alpha \frac{\partial L(\boldsymbol{w})}{\partial b_j}$$

感知机的缺点：仅适用于线性可分问题

- 线性可分：逻辑与关系→只有当所有输入都为真时，输出才为真

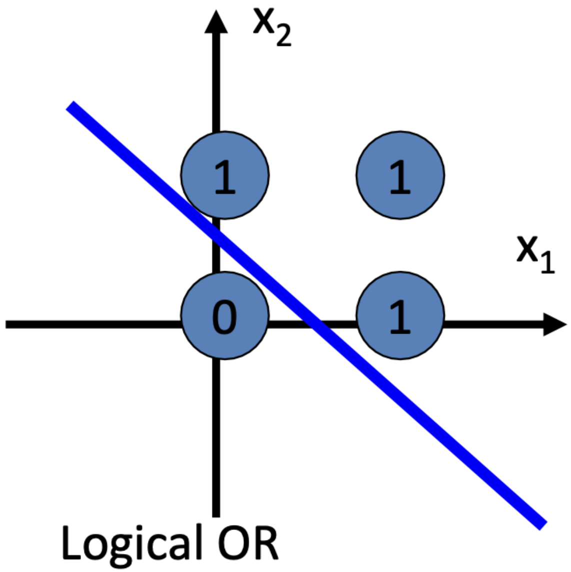


Logical AND

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

感知机的缺点：仅适用于线性可分问题

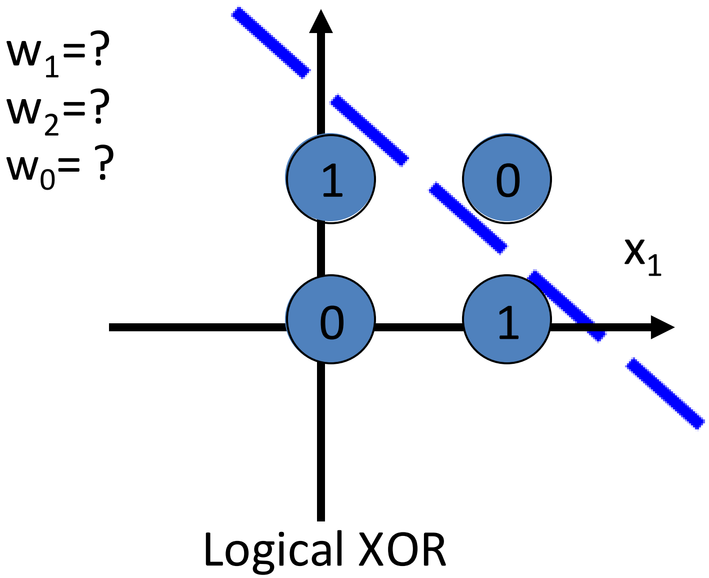
- 线性可分：逻辑或关系→只要其中一个输入为真，输出就为真



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

感知机的缺点：仅适用于线性可分问题

- 非线性可分：逻辑异或关系→我们无法通过一条直线将 (0,1) 和 (1,0) 分开，同时又保持 (0,0) 和 (1,1) 的分类正确



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


Thank You!