

# 第 8 章 人工神经网络及其应用

---

**教材：**

王万良 《人工智能导论》（第5版）

高等教育出版社，2020

2023/10/16

# 第8章 人工神经网络及其应用

- 人工神经网络是对人脑或生物神经网络若干基本特性的抽象和模拟。为机器学习等许多问题的研究提供了一条新的思路，目前已经在模式识别、机器视觉、联想记忆、自动控制、信号处理、软测量、决策分析、智能计算、组合优化问题求解、数据挖掘等方面获得成功应用。
- 本章着重介绍最基本、最典型、应用最广泛的BP神经网络和Hopfield神经网络及其在模式识别、联想记忆、软测量、智能计算、组合优化问题求解等方面的应用。

# 第8章 人工神经网络及其应用

## ● 神经网络 (neural networks, NN)

- **生物神经网络** (natural neural network, NNN): 由中枢神经系统（脑和脊髓）及周围神经系统（感觉神经、运动神经等）所构成的错综复杂的神经网络，其中最重要的是**脑神经系统**。
- **人工神经网络** (artificial neural networks, ANN): 模拟**人脑神经系统的**结构和功能，运用大量简单处理单元经广泛连接而组成的人工网络系统。

神经网络方法：**隐式**的  
知识表示方法

# 第8章 人工神经网络及其应用

- 8.1 神经元与神经网络
- 8.2 BP神经网络及其学习算法
- 8.3 BP神经网络在模式识别中的应用
- 8.4 Hopfield神经网络及其改进
- 8.5 Hopfield神经网络的应用
- 8.6 卷积神经网络与深度学习
- 8.7 生成对抗网络及其应用

# 第8章 人工神经网络及其应用

## ✓ 8.1 神经元与神经网络

### □ 8.2 BP神经网络及其学习算法

### □ 8.3 BP神经网络在模式识别中的应用

### □ 8.4 Hopfield神经网络及其改进

### □ 8.5 Hopfield神经网络的应用

### □ 8.6 卷积神经网络与深度学习

### □ 8.7 生成对抗网络及其

# 8.1 神经元与神经网络

- 8.1.1 生物神经元的结构
- 8.1.2 神经元数学模型
- 8.1.3 神经网络结构与工作方式
- 8.1.4 神经网络的学习

# 8.1.1 生物神经元的结构

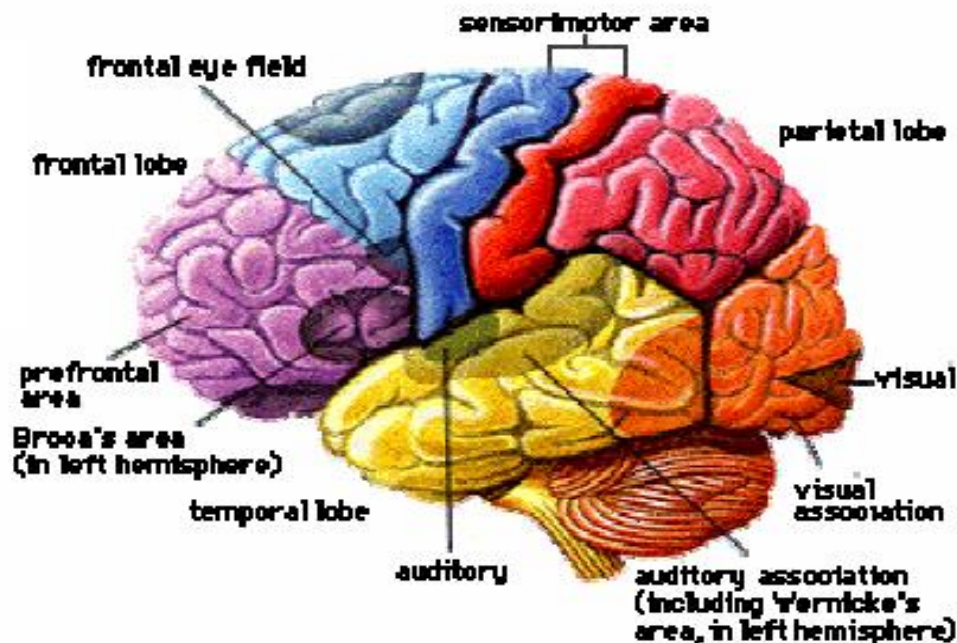
## ■ 人脑构造:

皮层 (cortex)

中脑 (midbrain)

脑干 (brainstem)

小脑 (cerebellum)



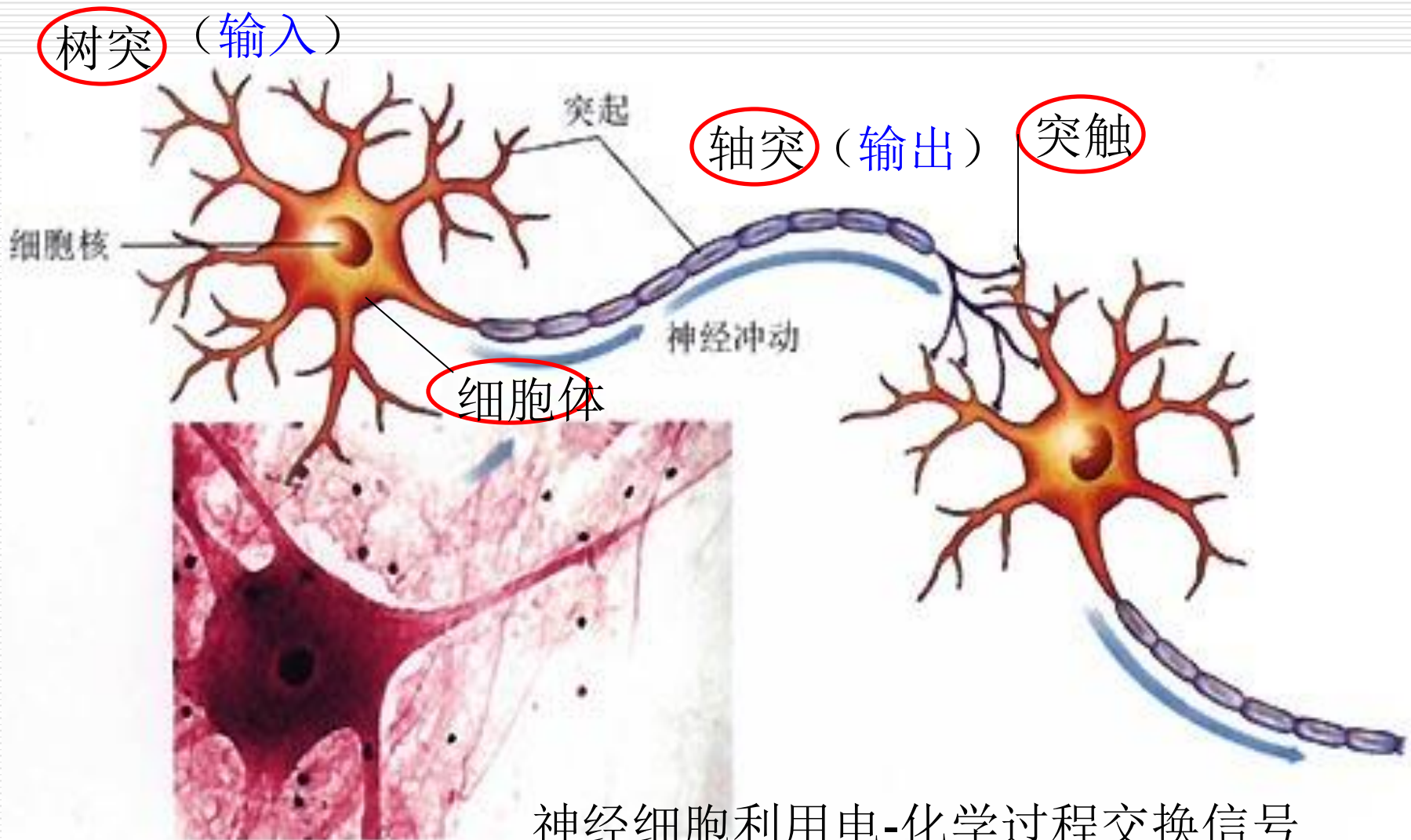
■ 人脑由一千多亿（ $10^{11}$ 亿— $10^{14}$  亿）个神经细胞（神经元）交织在一起的网状结构组成，其中大脑皮层约140亿个神经元，小脑皮层约1000亿个神经元。

■ 神经元约有1000种类型，每个神经元大约与 $10^3$ — $10^4$ 个其他神经元相连接，形成极为错综复杂而又灵活多变的神经网络。



# 8.1.1 生物神经元的结构

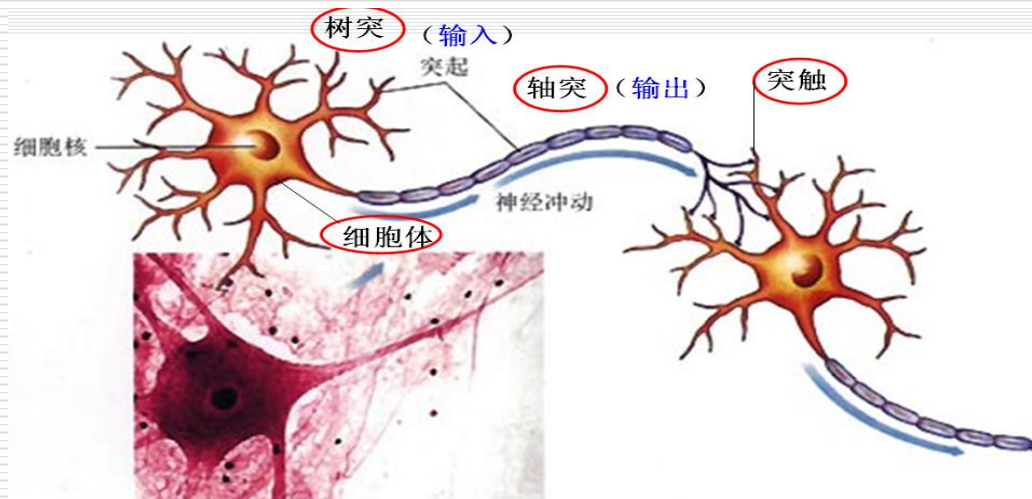
## □ 1. 生物神经元结构





# 8.1.1 生物神经元的结构

- 工作状态：
  - 兴奋状态：细胞膜电位  $>$  动作电位的阈值  $\rightarrow$  神经冲动
  - 抑制状态：细胞膜电位  $<$  动作电位的阈值
- 学习与遗忘：由于神经元结构的可塑性，突触的传递作用可增强和减弱。



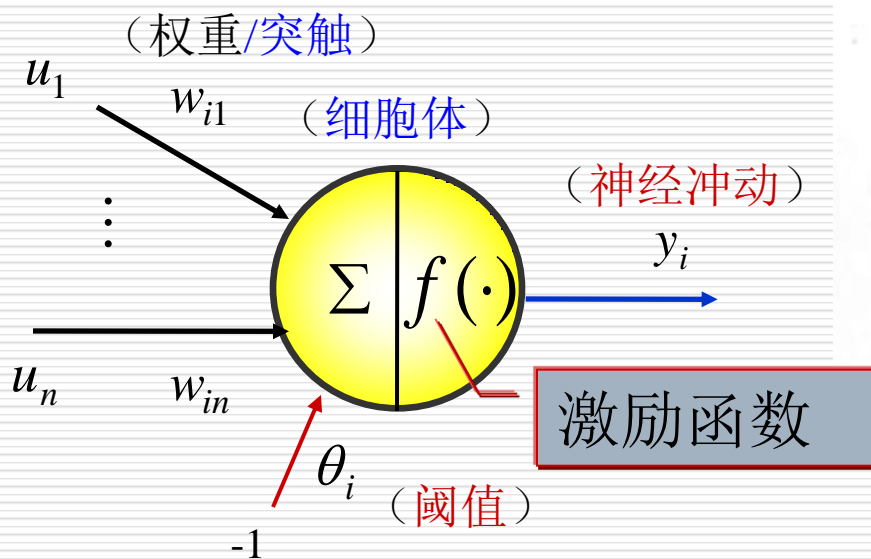
# 8.1 神经元与神经网络

- 8.1.1 生物神经元的结构
- 8.1.2 神经元数学模型
- 8.1.3 神经网络的结构与工作方式
- 8.1.4 神经网络的学习

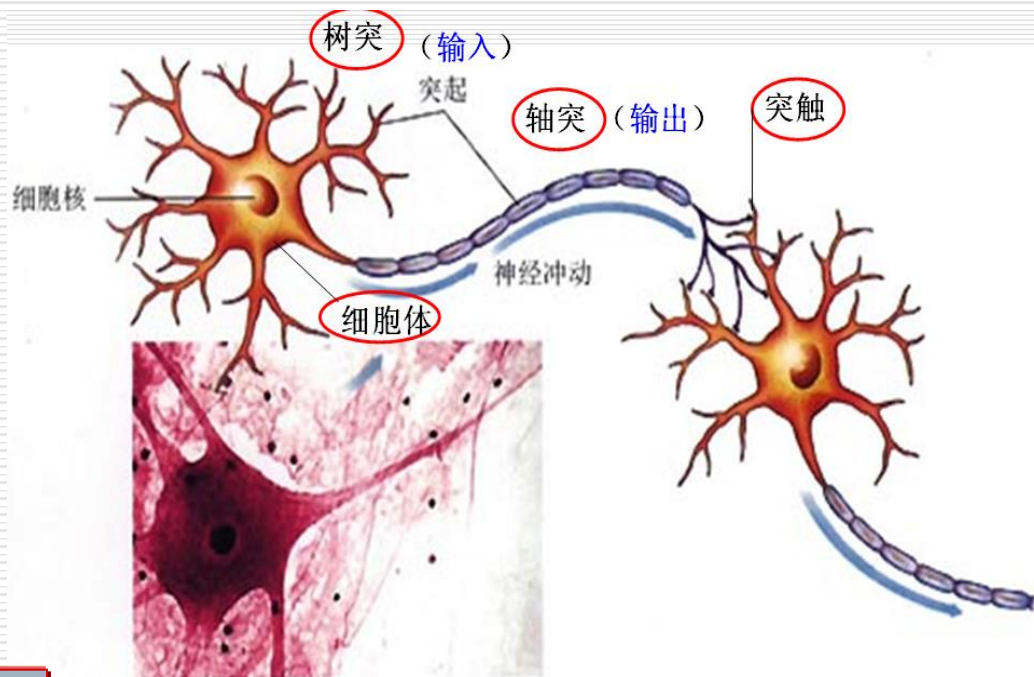
# 8.1.2 神经元数学模型

## 2. 人工神经元模型

1943年，麦克洛奇和皮兹提出M—P模型。



人工神经元  $i$  的模型图

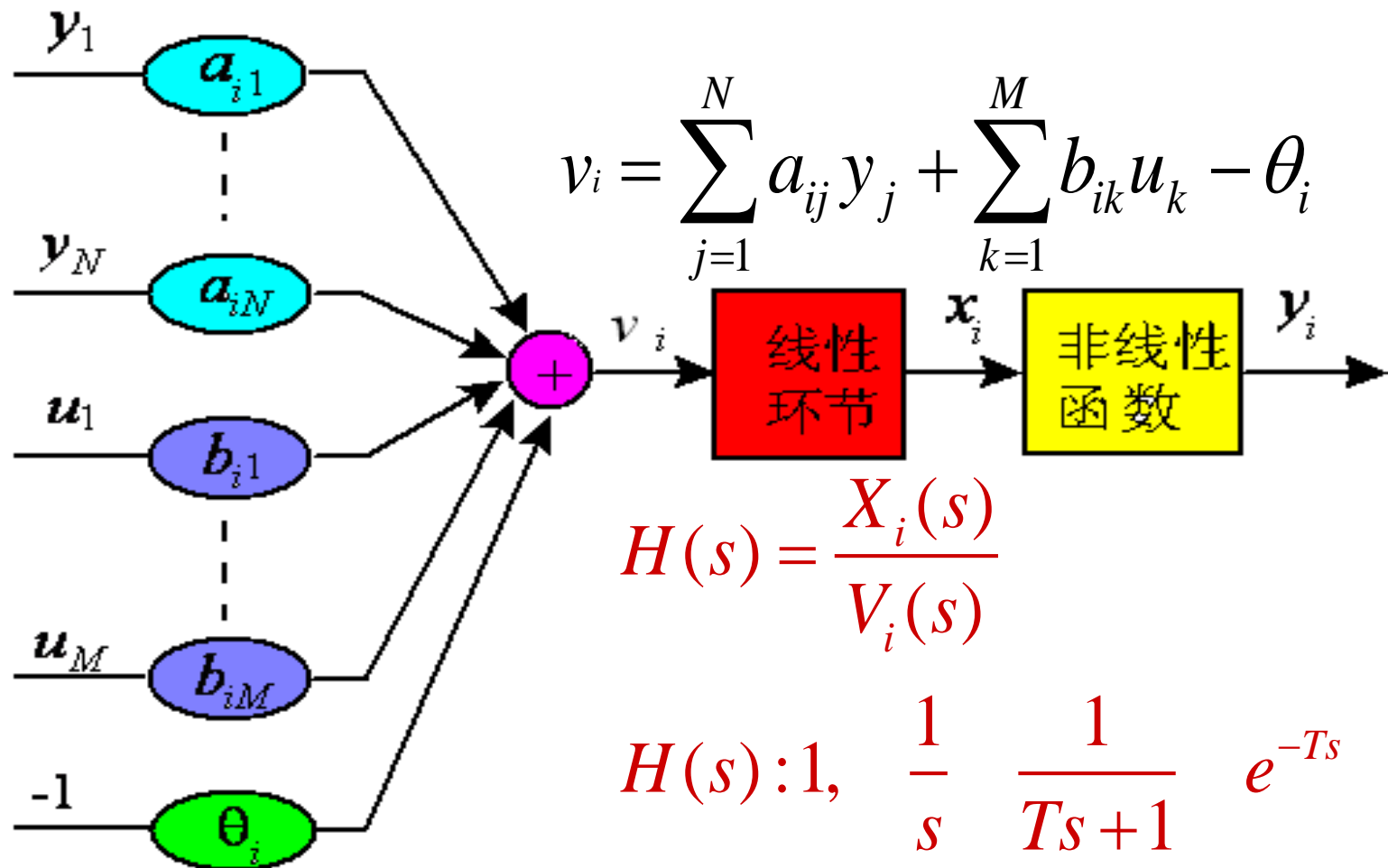


$$y_i = f\left(\sum_{j=1}^n w_{ij}u_j - \theta_i\right) = f\left(\sum_{j=0}^n w_{ij}u_j\right)$$

$(u_0 = -1, w_{i0} = \theta_i)$

## 8.1.2 神经元数学模型

一般模型：



## 8.1.2 神经元数学模型

$y_i(t)$  : 第  $i$  个神经元的输出。

$\theta_i$  : 第  $i$  个神经元的阈值。

$u_k(t) (k=1, 2, \dots, M)$  : 外部输入。

$a_{ij}, b_{ik}$  : 权值。

■ 加权求和:

$$v_i(t) = \sum_{j=1}^N a_{ij} y_j(t) + \sum_{k=1}^M b_{ik} u_k(t) - \theta_i$$

其矩阵形式:  $V(t) = AY(t) + BU(t) - \theta$

$$A = \{a_{ij}\}_{N \times N}$$

$$B = \{b_{ik}\}_{N \times M}$$

$$V = [v_1 \quad \cdots \quad v_N]^T$$

$$U = [u_1 \quad \cdots \quad u_M]^T$$

$$\theta = [\theta_1 \quad \cdots \quad \theta_N]^T$$

$$Y = [y_1 \quad \cdots \quad y_N]^T$$

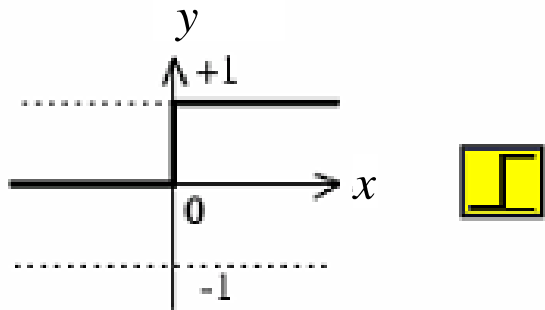
## 8.1.2 神经元数学模型

- 线性环节的传递函数:  $X_i(s) = H(s)V_i(s)$

$H(s)$ :  $1$ ;  $\frac{1}{s}$ ;  $\frac{1}{Ts+1}$  ;  $e^{-Ts}$  及其组合等。

## 8.1.2 神经元数学模型

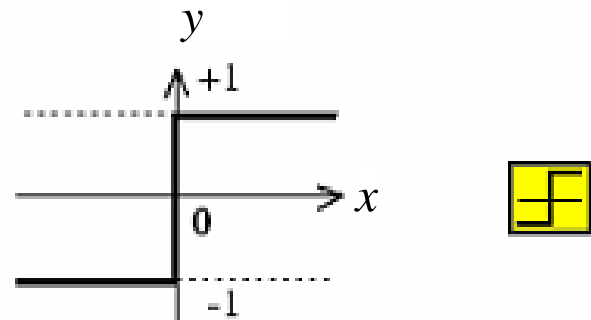
### ■ 非线性激励函数（传输函数、输出变换函数）



$$y = \text{hardlim}(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$

**Hard-Limit Transfer Function**

（硬极限函数或阶跃函数）



$$y = \text{hardlims}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

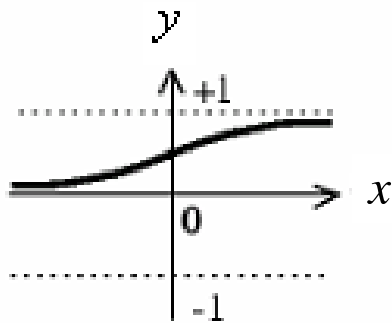
**Symmetric Hard-Limit Trans. Funct.**

（对称硬极限函数）



## 8.1.2 神经元数学模型

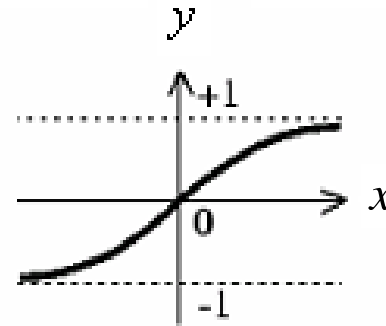
### ■ 非线性激励函数（传输函数、输出变换函数）



$$y = \text{logsig}(x) = \frac{1}{1 + e^{-\alpha x}}$$
$$\alpha = 1$$

Log-Sigmoid Transfer Function

（对数-S形函数或S型函数）



$$y = \text{tansig}(x) = \frac{e^{\alpha x} - e^{-\alpha x}}{e^{\alpha x} + e^{-\alpha x}}$$
$$\alpha = 1$$

Tan-Sigmoid Transfer Function

（双曲正切S形函数）

## 8.1.2 神经元数学模型

### ■ 工作过程:

- 从各输入端接收输入信号  $u_j$  ( $j = 1, 2, \dots, n$ )
- 根据连接权值求出所有输入的加权和

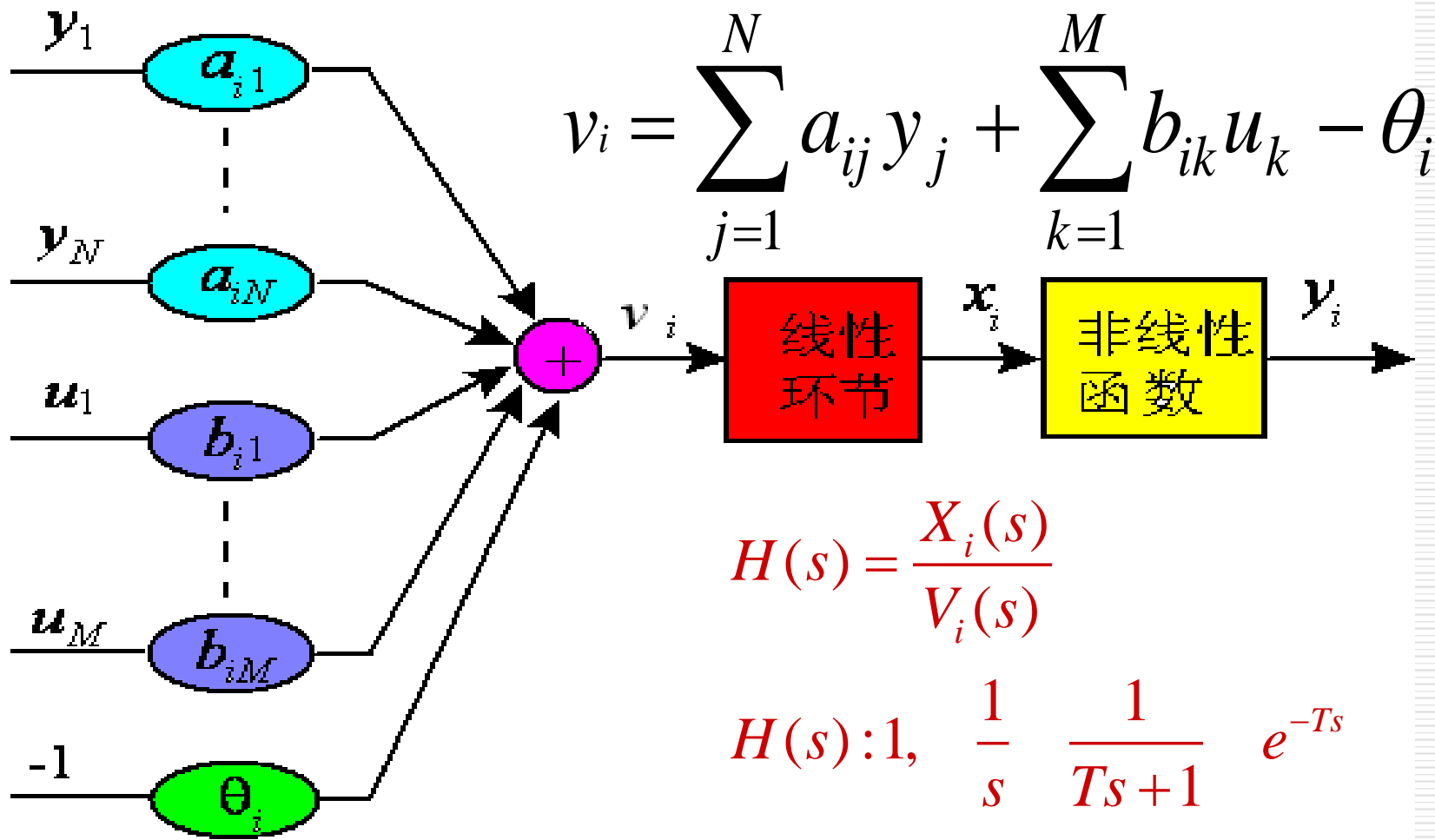
$$x_i = \sum_{j=1}^n w_{ij} u_j - \theta_i = \sum_{j=1}^n w_{ij} u_j + b_i = \sum_{j=0}^n w_{ij} u_j$$

$$(w_{i0} = -1, u_0 = \theta_i \text{ 或 } w_{i0} = 1, u_0 = b_i)$$

- 用非线性激励函数进行转换，得到输出

$$y_i = f(x_i) = f\left(\sum_{j=0}^n w_{ij} u_j\right)$$

## 8.1.2 神经元数学模型



# 8.1 神经元与神经网络

- 8.1.1 生物神经元的结构
- 8.1.2 神经元的数学模型
- 8.1.3 神经网络的结构与工作方式
- 8.1.4 神经网络的学习

## 8.1.3 神经网络的结构与工作方式

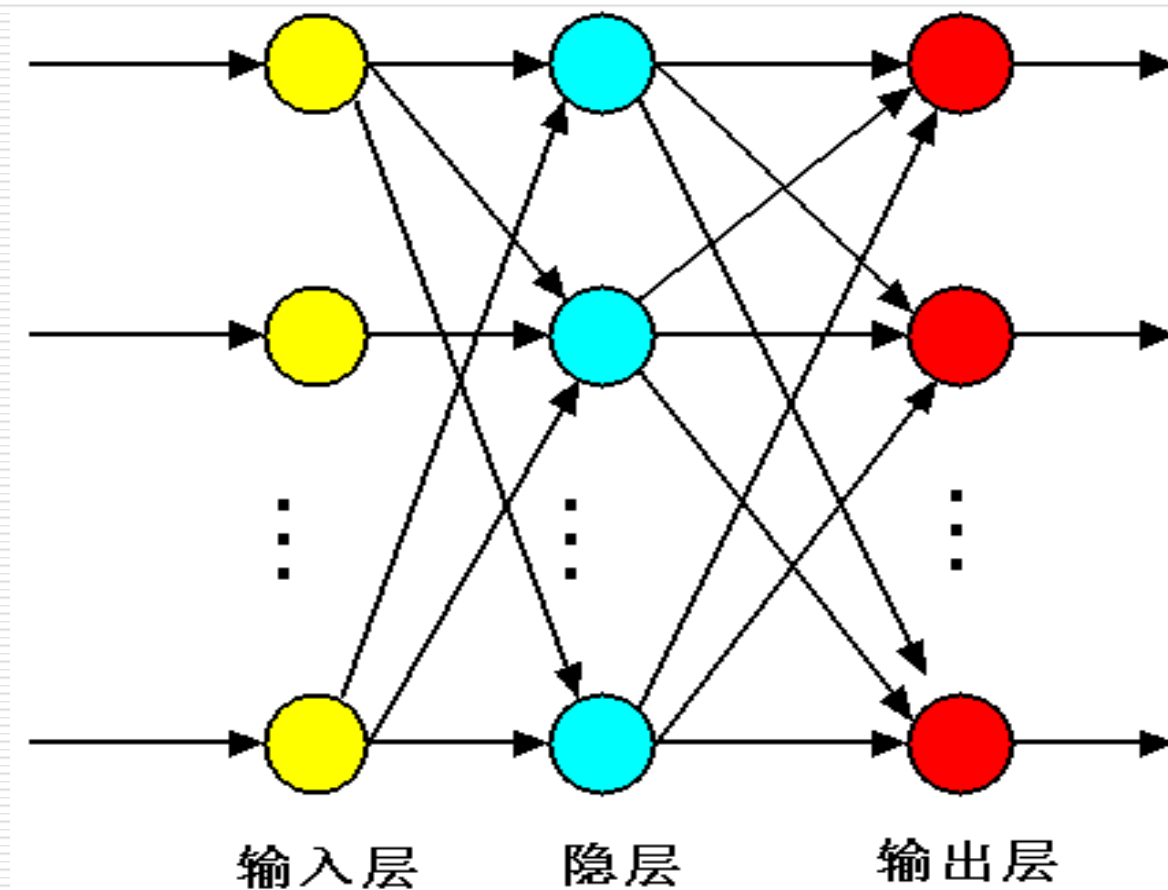
□ 决定人工神经网络性能的三大要素：

- 神经元的特性。
- 神经元之间相互连接的形式——拓扑结构。
- 为适应环境而改善性能的学习规则。

## 8.1.3 神经网络的结构与工作方式

### □ 1. 神经网络的结构

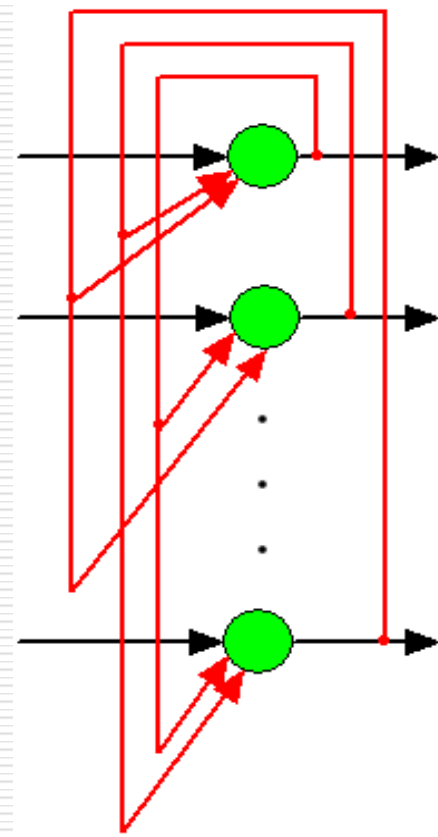
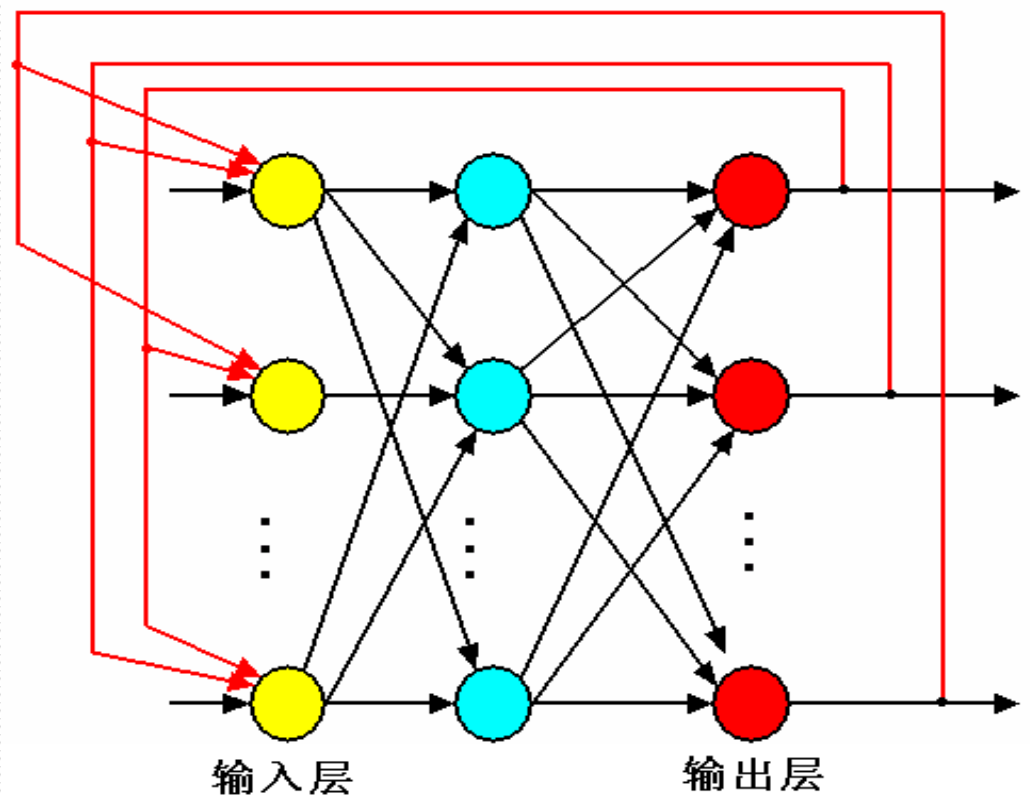
#### (1) 前馈型（前向型）



## 8.1.3 神经网络的结构与工作方式

### 1. 神经网络的结构

#### (2) 反馈型



( Hopfield神经网络 )



## 8.1.3 神经网络的结构与工作方式

### □ 2. 神经网络的工作方式

- **同步**（并行）方式：任一时刻神经网络中所有神经元同时调整状态。
- **异步**（串行）方式：任一时刻只有一个神经元调整状态，而其它神经元的状态保持不变。

# 8.1 神经元与神经网络

- 8.1.1 生物神经元的结构
- 8.1.2 神经元的数学模型
- 8.1.3 神经网络的结构与工作方式
- 8.1.4 神经网络的学习

- ❑ [https://www.bilibili.com/video/BV1Hr4y1q7Y1/?spm\\_id\\_from=333.788.recommend\\_more\\_video.-1&vd\\_source=e81dcbaa3b668b4eedfa281d2c2b69c3](https://www.bilibili.com/video/BV1Hr4y1q7Y1/?spm_id_from=333.788.recommend_more_video.-1&vd_source=e81dcbaa3b668b4eedfa281d2c2b69c3) 7:51
- ❑ [https://www.bilibili.com/video/BV1bx411M7Zx/?spm\\_id\\_from=333.788.recommend\\_more\\_video.0&vd\\_source=e81dcbaa3b668b4eedfa281d2c2b69c3](https://www.bilibili.com/video/BV1bx411M7Zx/?spm_id_from=333.788.recommend_more_video.0&vd_source=e81dcbaa3b668b4eedfa281d2c2b69c3) 16:05

# 8.1 神经元与神经网络

## □ 8.1.4 神经网络的学习

神经网络方法是一种知识表示方法和推理方法。  
神经网络知识表示是一种隐式的表示方法。

1944年赫布（Hebb）提出了改变神经元连接强度的Hebb学习规则。

**Hebb学习规则：**当某一突触两端的神经元同时处于兴奋状态，那么该连接的权值应该增强。

# 第8章 人工神经网络及其应用

- 8.1 神经元与神经网络
- ✓ 8.2 BP神经网络及其学习算法
- 8.3 BP神经网络在模式识别中的应用
- 8.4 Hopfield神经网络及其改进
- 8.5 Hopfield神经网络的应用
- 8.6 卷积神经网络与深度学习
- 8.7 生成对抗网络及其应用

## 8.2 BP神经网络及其学习算法

- 8.2.1 BP神经网络 的结构
- 8.2.2 BP学习算法
- 8.2.3 BP算法的实现
- 8.2.3 BP算法的应用

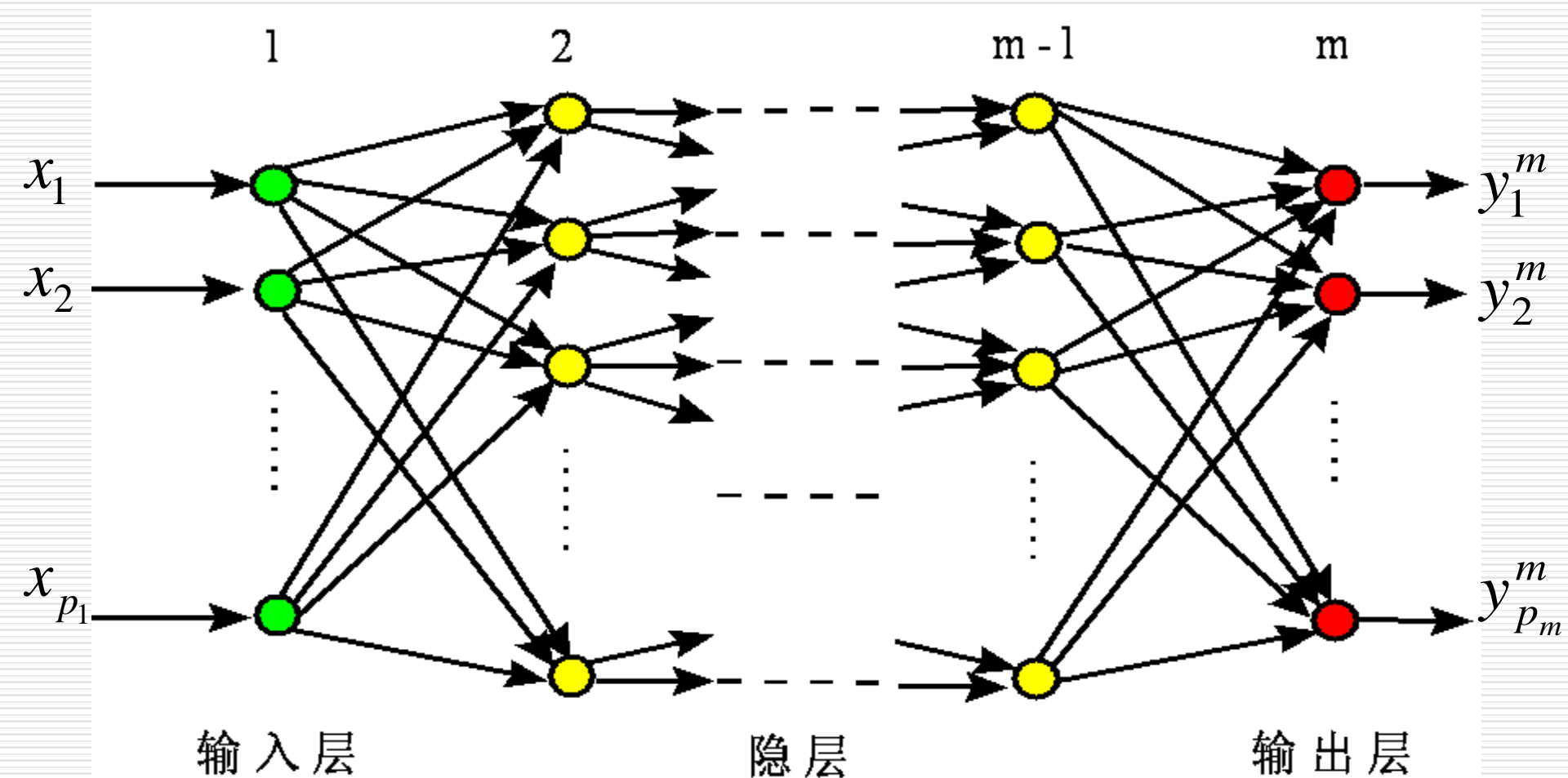
## 8.2 BP神经网络及其学习算法

- 8.2.1 BP神经网络的结构
- 8.2.2 BP学习算法
- 8.2.3 BP算法的实现



# 8.2.1 BP神经网络的结构

## 1. BP 网络结构

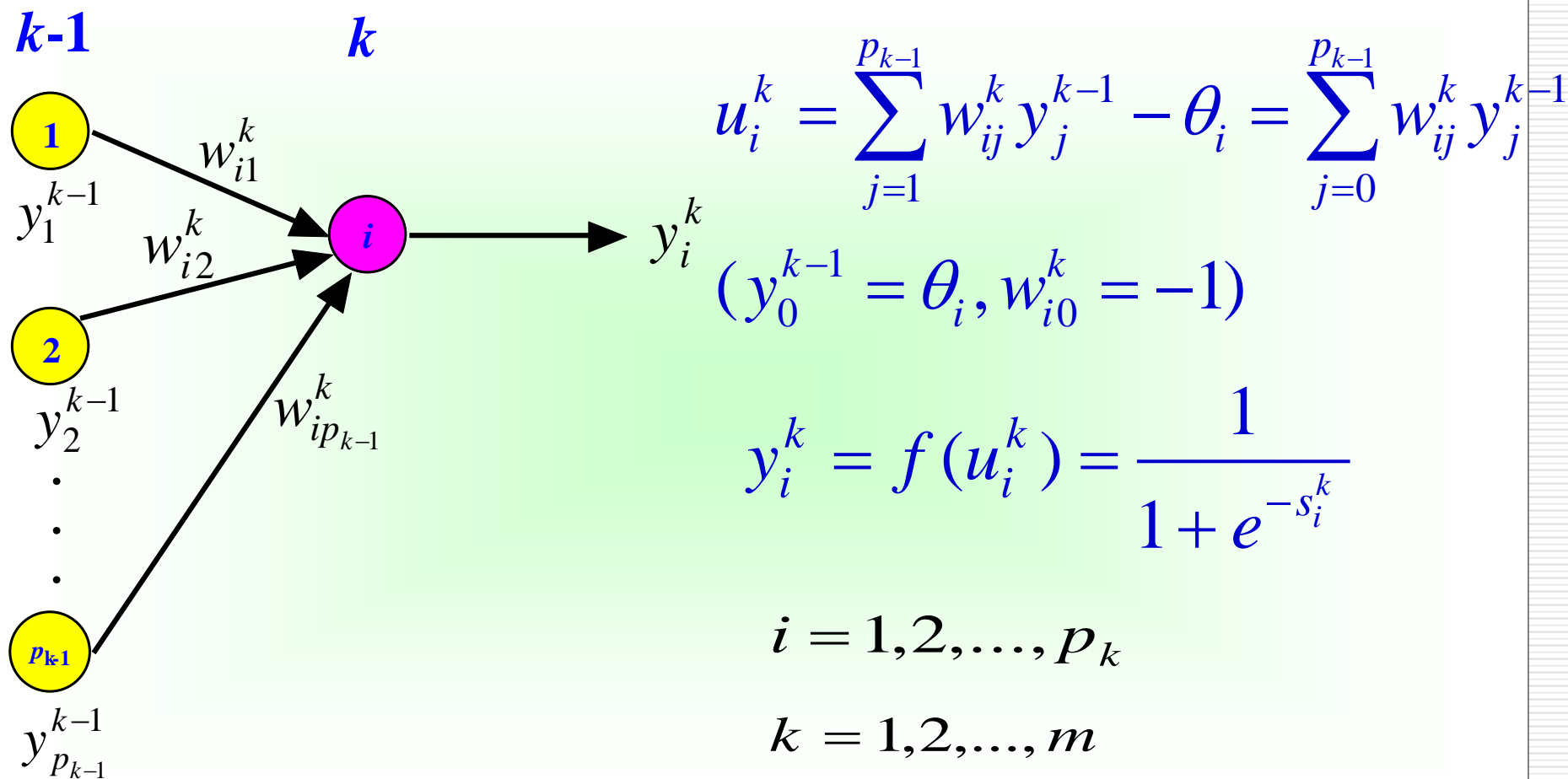


$$X = [x_1 \quad x_2 \quad \cdots \quad x_{p_1}]^T$$

$$Y = [y_1^m \quad y_2^m \quad \cdots \quad y_{p_m}^m]^T$$

## 8.2.1 BP神经网络的结构

### 2. 输入输出变换关系



## 8.2.1 BP神经网络的结构

### 3. 工作过程

- 第一阶段或网络训练阶段：

- $N$  组输入输出样本：  $x_i=[x_{i1}, x_{i2}, \dots, x_{ip1}]^T$

$$d_i=[d_{i1}, d_{i2}, \dots, d_{ipm}]^T$$

$$i=1, 2, \dots, N$$

- 对网络的连接权进行学习和调整，以使该网络实现给定样本的输入输出映射关系。

- 第二阶段或称工作阶段：把实验数据或实际数据输入到网络，网络在误差范围内预测计算出结果。

## 8.2 BP神经网络及其学习算法

- 8.2.1 BP神经网络的结构
- 8.2.2 BP学习算法
- 8.2.3 BP算法的实现
- 8.2.3 BP算法的应用

## 8.2.2 BP学习算法

### ■ 两个问题:

- (1) 是否存在一个BP神经网络能够逼近给定的样本或者函数。

**BP定理:** 给定任意  $\varepsilon > 0$  , 对于任意的连续函数, 存在一个三层前向神经网络, 它可以在任意  $\varepsilon$  平方误差精度内逼近连续函数。

- (2) 如何调整BP神经网络的连接权, 使网络的输入与输出与给定的样本相同。
- ◆ 1986年, 鲁梅尔哈特 (D. Rumelhart) 等提出BP学习算法。

## 8.2.2 BP学习算法

### ■ 1. 基本思想

- 目标函数: 
$$J = \frac{1}{2} \sum_{j=1}^{p_m} (y_j^m - d_j)^2$$

- 约束条件: 
$$u_i^k = \sum_j w_{ij}^{k-1} y_j^{k-1} \quad i = 1, 2, \dots, p_k$$

$$y_i^k = f_k(u_i^k) \quad k = 1, 2, \dots, m$$

- 连接权值的修正量:

$$\Delta w_{ij}^{k-1} = -\varepsilon \frac{\partial J}{\partial w_{ij}^{k-1}} \quad j = 1, 2, \dots, p_{k-1}$$

## 8.2.2 BP学习算法

先求 
$$\frac{\partial J}{\partial w_{ij}^{k-1}} = \frac{\partial J}{\partial u_i^k} \frac{\partial u_i^k}{\partial w_{ij}^{k-1}} = \frac{\partial J}{\partial u_i^k} \frac{\partial}{\partial w_{ij}^{k-1}} \left( \sum_j w_{ij}^{k-1} y_j^{k-1} \right) = \frac{\partial J}{\partial u_i^k} y_j^{k-1}$$

记 
$$d_i^k = \frac{\partial J}{\partial u_i^k} = \frac{\partial J}{\partial y_i^k} \frac{\partial y_i^k}{\partial u_i^k} = \frac{\partial J}{\partial y_i^k} f'_k(u_i^k)$$

(1) 对输出层的神经元 
$$\frac{\partial J}{\partial y_i^k} = \frac{\partial J}{\partial y_i^m} = y_i^m - y_{si}$$

$$d_i^m = (y_i^m - y_{si}) f'_m(u_i^m)$$

(2) 对隐单元层，则有 
$$\frac{\partial J}{\partial y_i^k} = \sum_l \frac{\partial J}{\partial u_l^{k+1}} \frac{\partial u_l^{k+1}}{\partial y_i^k} = \sum_l d_l^{k+1} w_{li}^k$$

$$d_i^k = f'_k(u_i^k) \sum_l d_l^{k+1} w_{li}^k$$

$$\Delta w_{ij}^{k-1} = -\varepsilon d_i^k y_j^{k-1}$$



## 8.2.2 BP学习算法

### ■ 2. BP学习算法

$$\Delta w_{ij}^{k-1} = -\varepsilon d_i^k y_j^{k-1}$$

$$d_i^m = (y_i^m - y_{si}) f'_m(u_i^m)$$

——输出层连接权调整公式

$$d_i^k = f'_k(u_i^k) \sum_l d_l^{k+1} w_{li}^k$$

——隐层连接权调整公式

## 8.2.2 BP学习算法

### ■ 2. 学习算法

$$\text{当 } y_i^k = \frac{1}{1 + e^{-u_i^k}} \text{ 时}$$

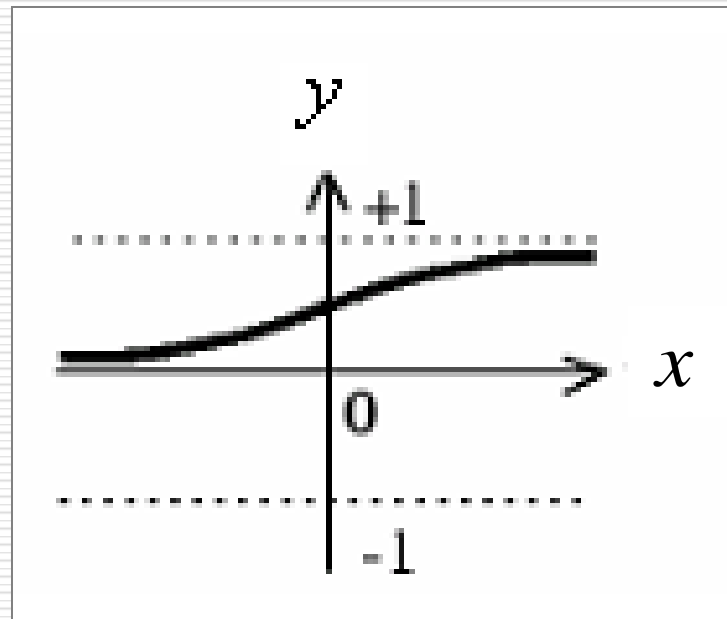
$$\Delta w_{ij}^{k-1} = -\varepsilon d_i^k y_j^{k-1}$$

$$d_i^m = y_i^m (1 - y_i^m) (y_i^m - y_i)$$

——输出层连接权调整公式

$$d_i^k = y_i^k (1 - y_i^k) \sum_{l=1}^{p_{k+1}} w_{li}^{k+1} d_l^{k+1}$$

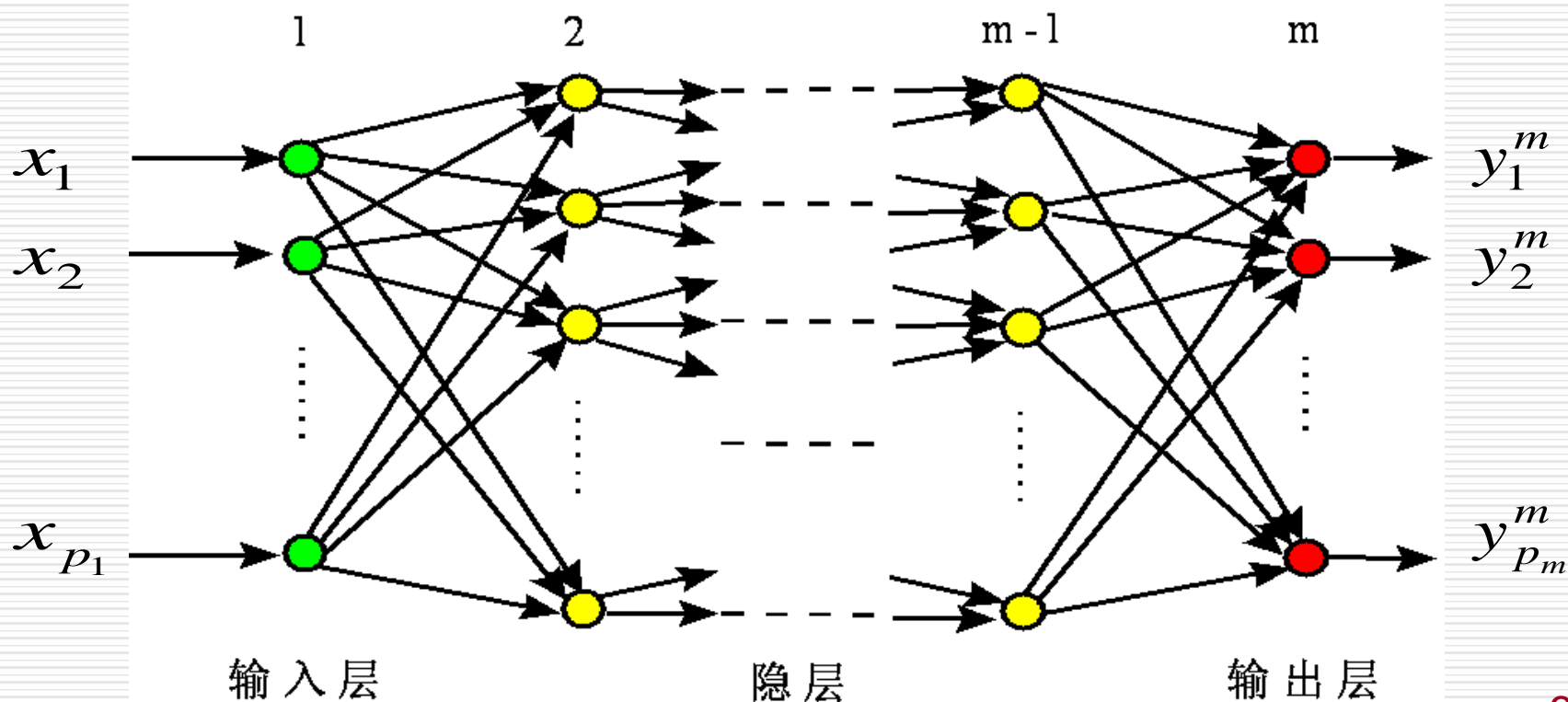
——隐层连接权调整公式



## 8.2.2 BP学习算法

### ■ 2. 学习算法

- 正向传播：输入信息由输入层传至隐层，最终在输出层输出。
- 反向传播：修改各层神经元的权值，使误差信号最小。



## 8.2 BP神经网络及其学习算法

□ 8.2.1 BP神经网络的结构

□ 8.2.2 BP学习算法

□ 8.2.3 BP算法的实现

## 8.2.3 BP算法的实现

### ■ 1. BP算法的设计

- (1) **隐层数及隐层神经元数的确定**：目前尚无理论指导。
- (2) **初始权值的设置**：一般以一个均值为0的随机分布设置网络的初始权值。
- (3) **训练数据预处理**：线性的特征比例变换，将所有的特征变换到 $[0, 1]$ 或者 $[-1, 1]$ 区间内，使得在每个训练集上，每个特征的均值为0，并且具有相同的方差。
- (4) **后处理过程**：当应用神经网络进行分类操作时，通常将输出值编码成所谓的名义变量，具体的值对应类别标号。

## 8.2.3 BP算法的实现

### ■ 2. BP算法的计算机实现流程

(1) 初始化：对所有连接权和阈值赋以随机任意小值；

$$w_{ij}^k(t), \quad \theta_i^k(t), (k = 1, \dots, m; i = 1, \dots, p_k; j = 1, \dots, p_{k-1}; t = 0)$$

(2) 从  $N$  组输入输出样本中取一组样本：  $x=[x_1, x_2, \dots, x_{p_1}]^T$ ,  $d=[d_1, d_2, \dots, d_{p_m}]^T$ , 把输入信息  $x=[x_1, x_2, \dots, x_{p_1}]^T$  输入到BP网络中

(3) 正向传播：计算各层节点的输出：

$$y_i^k \quad (i = 1, \dots, p_k; k = 1, \dots, m)$$

(4) 计算网络的实际输出与期望输出的误差：

$$e_i = y_i - y_i^m \quad (i = 1, \dots, p_m)$$

## 8.2.3 BP算法的实现

### ■ 2. BP算法的计算机实现流程

(5) 反向传播：从输出层方向计算到第一个隐层，按连接权值修正公式向减小误差方向调整网络的各个连接权值。

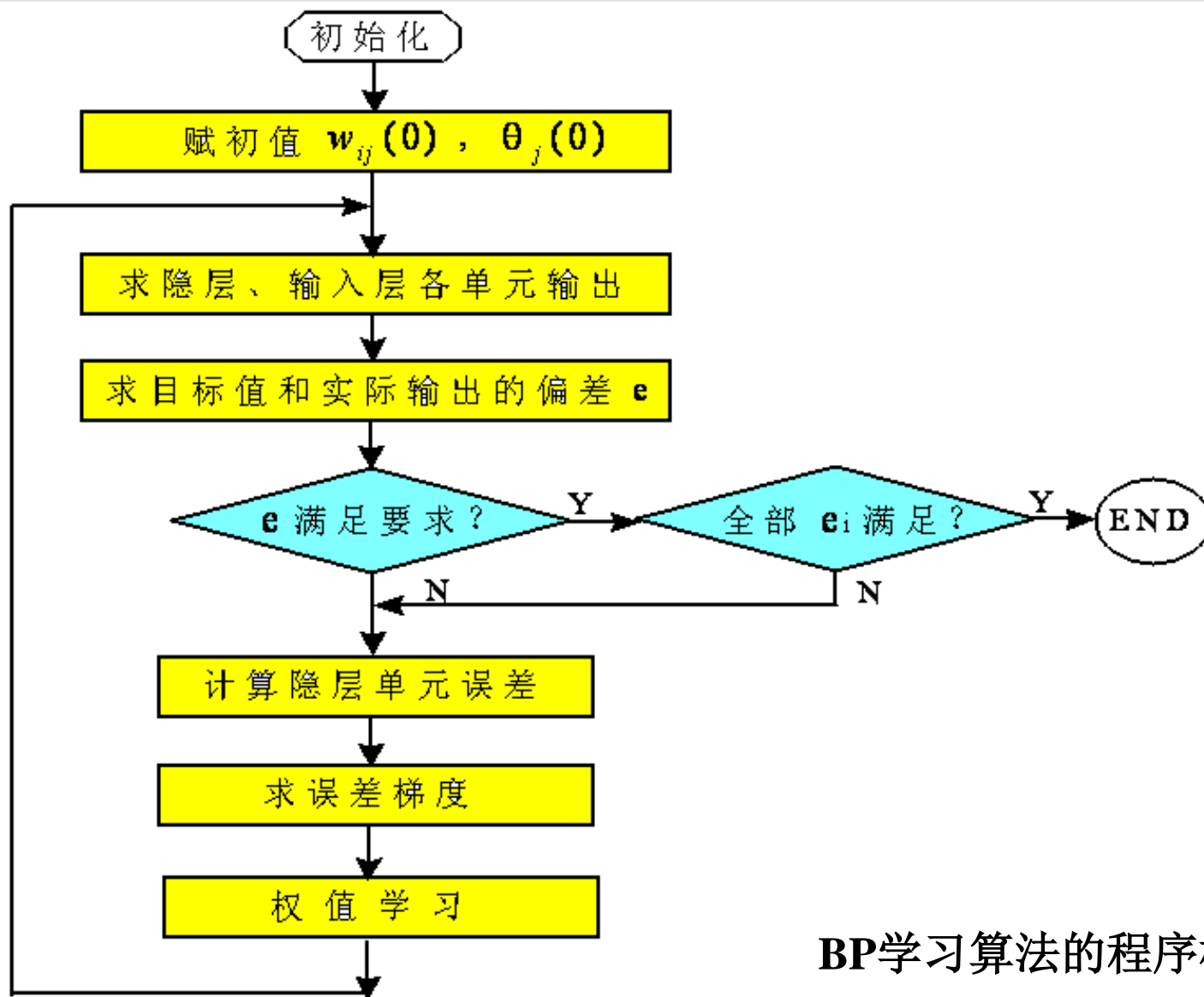
$$\Delta w_{ij} = -\alpha d_i^k y_j^{k-1}$$

$$d_i^m = y_i^m (1 - y_i^m) (y_i^m - y_i) \quad \text{——输出层连接权调整公式}$$

$$d_i^k = y_i^k (1 - y_i^k) \sum_{l=1}^{p_{k+1}} w_{li}^{k+1} d_l^{k+1} \quad \text{——隐层连接权调整公式}$$

(6) 让 $t+1 \rightarrow t$ ，取出另一组样本重复 (2) — (5)，直到  $N$  组输入输出样本的误差达到要求时为止。

## 8.2.3 BP算法的实现



BP学习算法的程序框图



# BP算法的特点分析

## □ 1. 特点

- **BP网络**：多层前向网络（输入层、隐层、输出层）。
- **连接权值**：通过**Delta**学习算法进行修正。
- **神经元传输函数**：S形函数。
- **学习算法**：正向传播、反向传播。
- **层与层的连接**是单向的，信息的传播是双向的。

# BP算法的特点分析

## □ 2. BP网络的主要优缺点

### ■ 优点

- 很好的逼近特性。
- 具有较强的泛化能力。
- 具有较好的容错性。

### ■ 缺点

- 收敛速度慢。
- 局部极值。
- 难以确定隐层和隐层结点的数目。

# 第8章 人工神经网络及其应用

- 8.1 神经元与神经网络
- 8.2 BP神经网络及其学习算法
- 8.3 BP神经网络在模式识别中的应用
- 8.4 Hopfield神经网络及其改进
- 8.5 Hopfield神经网络的应用
- 8.6 卷积神经网络与深度学习
- 8.7 生成对抗网络及其应用

## 8.3 BP神经网络在模式识别中的应用

模式识别研究用计算机模拟生物、人的感知，对模式信息，如图像、文字、语音等，进行识别和分类。

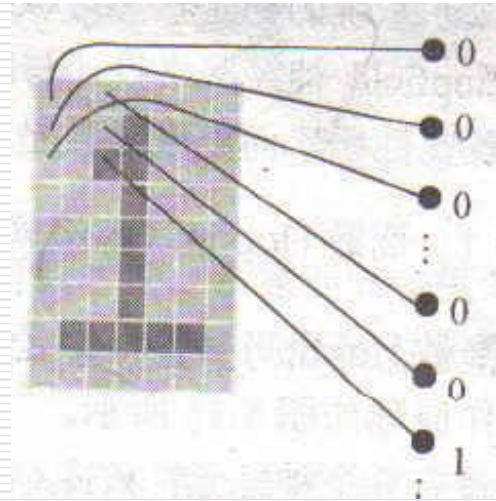
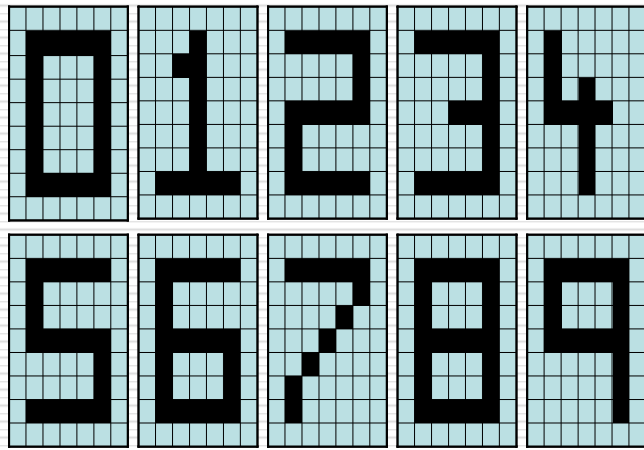
传统人工智能的研究部分地显示了人脑的归纳、推理等智能。但是，对于人类底层的智能，如视觉、听觉、触觉等方面，现代计算机系统的信息处理能力还不如一个幼儿园的孩子。

神经网络模型模拟了人脑神经系统的特点：处理单元的广泛连接；并行分布式信息储存、处理；自适应学习能力等。

神经网络模式识别方法具有较强的容错能力、自适应学习能力、并行信息处理能力。

## 8.3 BP神经网络在模式识别中的应用

例8.1 设计一个三层BP网络对数字0至9进行分类。



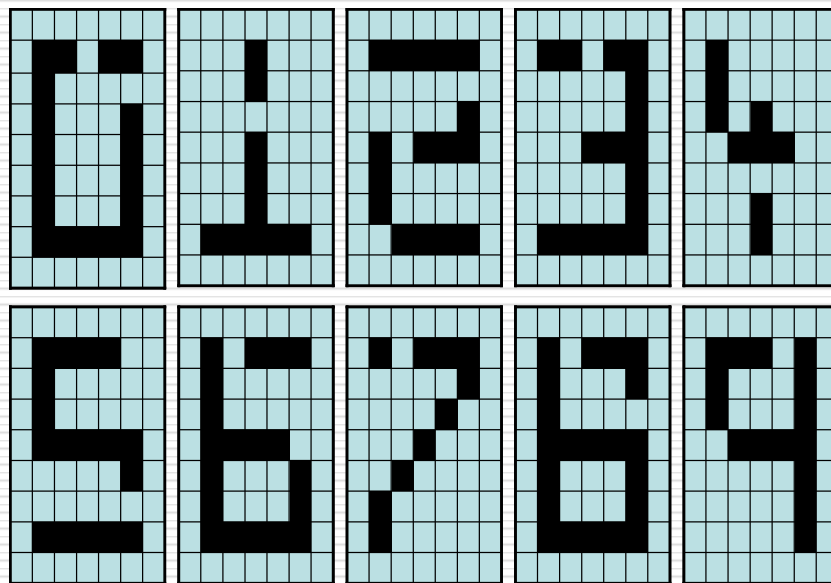
每个数字用 $9 \times 7$ 的网格表示，灰色像素代表0，黑色像素代表1。将每个网格表示为0, 1的长位串。位映射由左上角开始向下直到网格的整个一列，然后重复其他列。

选择BP网络结构为63-6-9。97个输入结点，对应上述网格的映射。9个输出结点对应10种分类。

使用的学习步长为0.3。训练600个周期，如果输出结点的值大于0.9，则取为ON，如果输出结点的值小于0.1，则取为OFF。

## 8.3 BP神经网络在模式识别中的应用

当训练成功后，对如图所示测试数据进行测试。测试数据都有一个或者多个位丢失。



测试结果表明：除了8以外，所有被测的数字都能够被正确地识别。

对于数字8，神经网络的第6个结点的输出值为0.53，第8个结点的输出值为0.41，表明第8个样本是模糊的，可能是数字6，也可能是数字8，但也不完全确信是两者之一。

BP算法

[https://www.bilibili.com/video/BV16x411V7Qg/?spm\\_id\\_from=333.880.my\\_history.page.click&vd\\_source=e81dcbaa3b668b4eedfa281d2c2b69c3](https://www.bilibili.com/video/BV16x411V7Qg/?spm_id_from=333.880.my_history.page.click&vd_source=e81dcbaa3b668b4eedfa281d2c2b69c3)



**THE END**

