

第三部分 神经网络控制

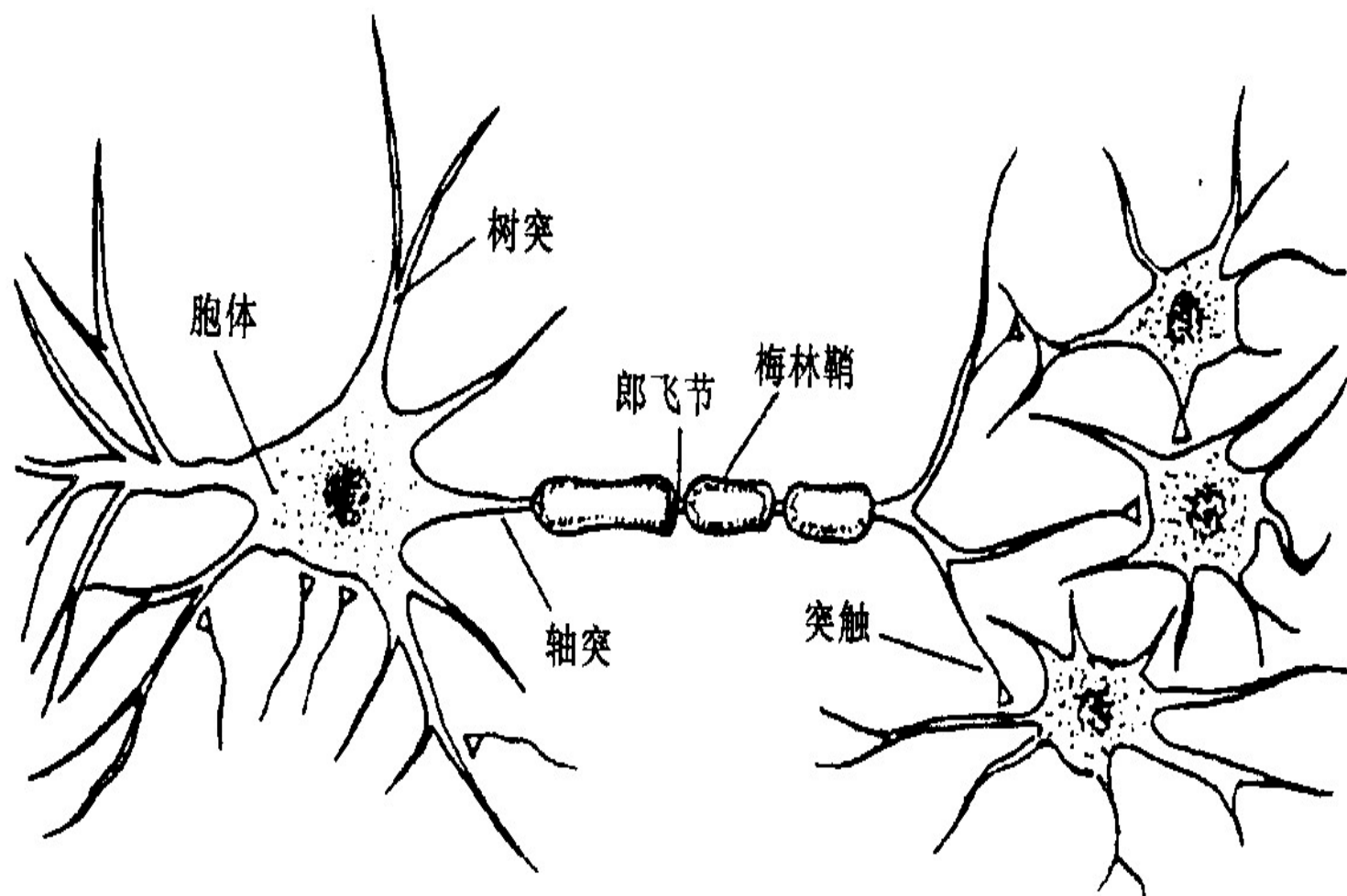
- 一、人工神经网络基础知识
- 二、神经网络控制的结构
- 三、神经网络控制器的设计

一、人工神经网络基础知识

(一) 概述

1、生物神经元模型

人脑大约包含 10^{12} 个神经元，分成约1000种类型，每个神经元大约与 $10^2 \sim 10^4$ 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。每个神经元虽然都十分简单，但是如此大量的神经元之间、如此复杂的连接却可以演化出丰富多彩的行为方式。同时，如此大量的神经元与外部感受器之间的多种多样的连接方式也蕴含了变化莫测的反应方式。



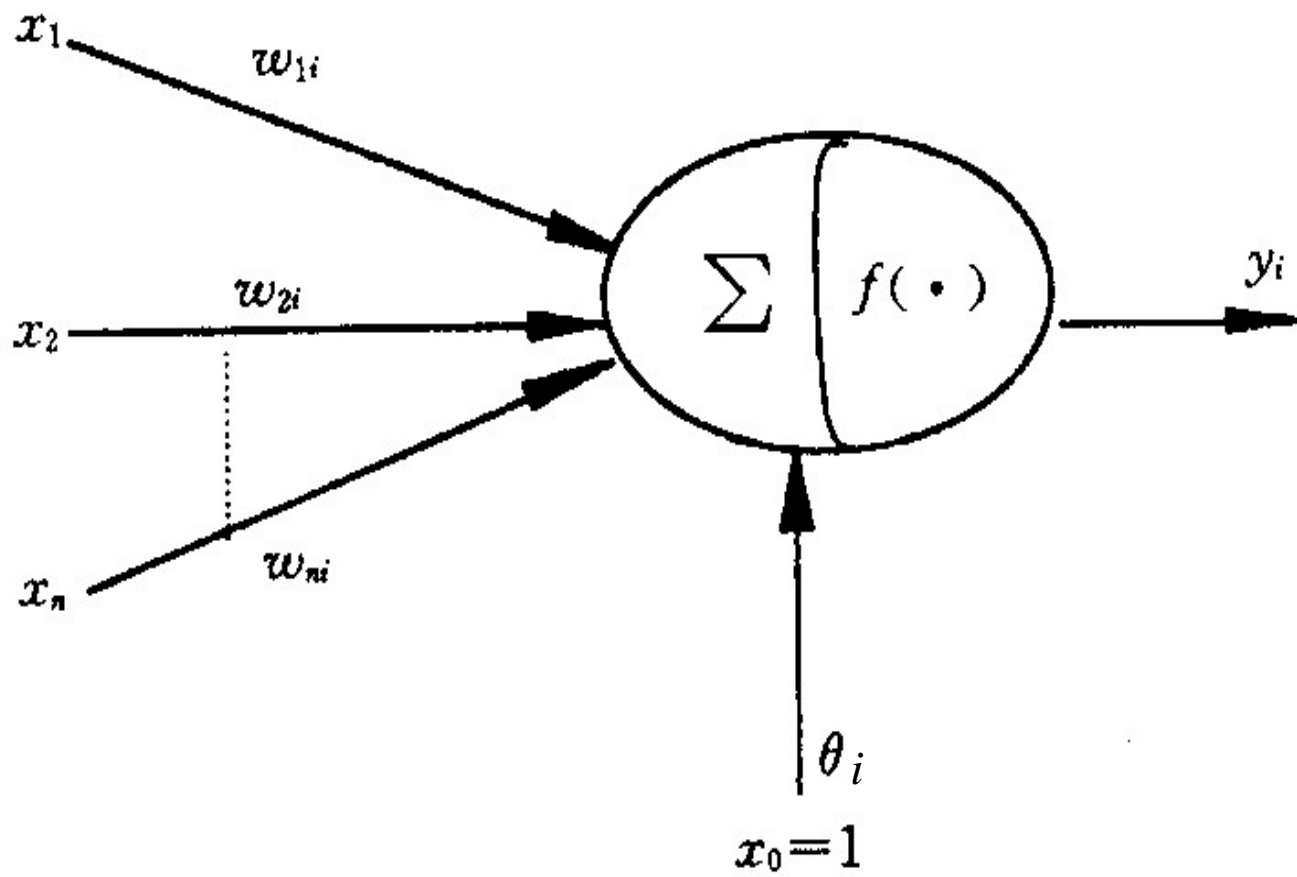
从生物控制论的观点来看，神经元作为控制和信息处理的基本单元，具有下列一些重要的功能与特性：

- 时空整合功能
- 兴奋与抑制状态
- 脉冲与电位转换
- 神经纤维传导
- 突触延时和不应期
- 学习、遗忘和疲劳

随着脑科学和生物控制论研究的进展，人们对神经元的结构和功能有了进一步的了解，神经元并不是一个简单的双稳态逻辑元件，而是超级的微型生物信息处理机或控制机单元。

2、人工神经元模型

- 人工神经元是对生物神经元的一种简化模拟。它是人工神经网络的基本处理单元。如图所示为一种简化的人工神经元结构。它是一个多输入、单输出的非线性元件。



- 其输入、输出关系可描述为

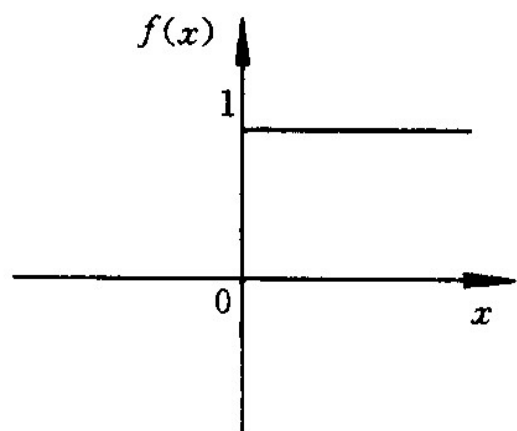
$$I_i = \sum_{j=1}^n w_{ji} x_j - \theta_i$$

$$y_i = f(I_i)$$

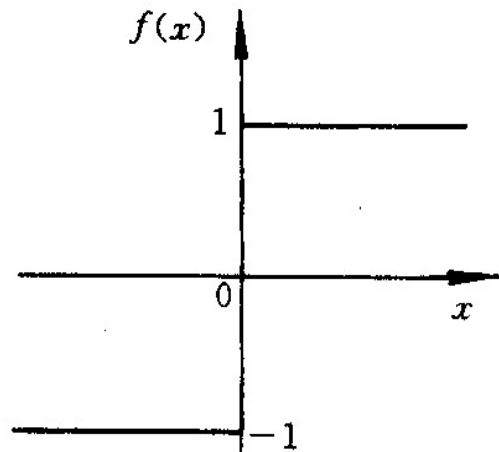
- 其中， $x_j (j=1,2,\cdots,n)$ 是从其他神经元传来的输入信号； w_{ji} 表示从神经元 j 到神经元 i 的连接权值； θ_i 为阈值； $f(\bullet)$ 称为激发函数或作用函数。

输出激发函数 $f(\bullet)$ 又称为变换函数，它决定神经元（节点）的输出。该输出为1或0，取决于其输入之和大于或小于内部阈值 θ_i 。函数 $f(\bullet)$ 一般具有非线性特性。下图表示了几种常见的激发函数。

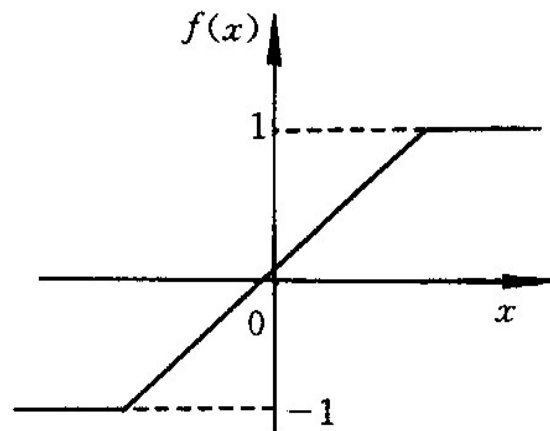
- ① 阈值型函数（见图（a），（b））
- ② 饱和型函数（见图（c））
- ③ 双曲函数（见图（d））
- ④ S型函数（见图（e））
- ⑤ 高斯函数（见图（f））



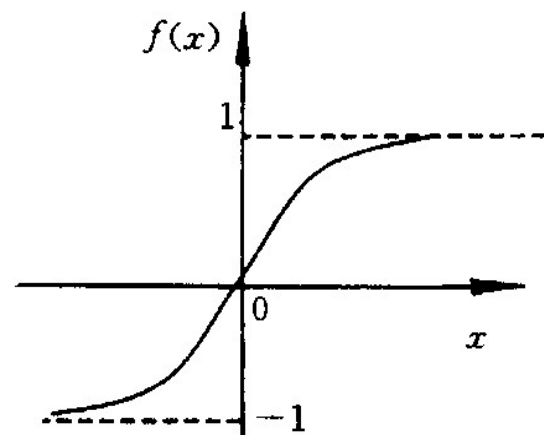
(a)



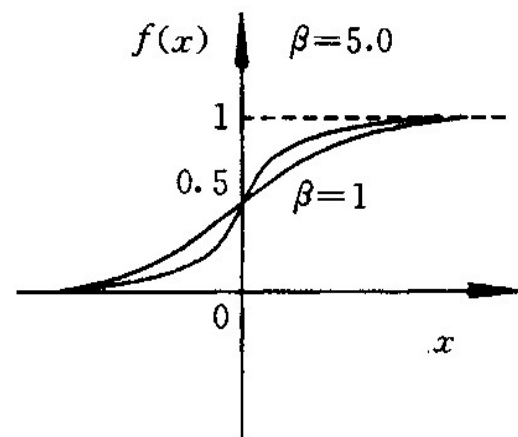
(b)



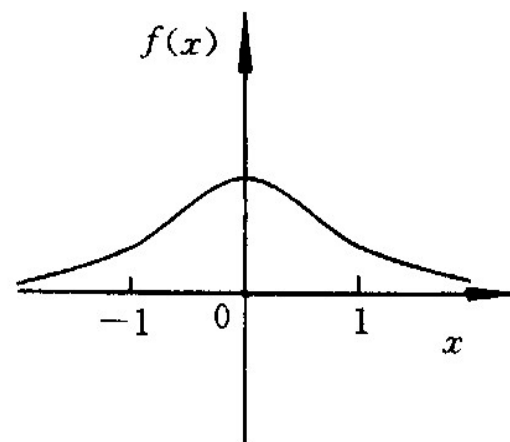
(c)



(d)



(e)



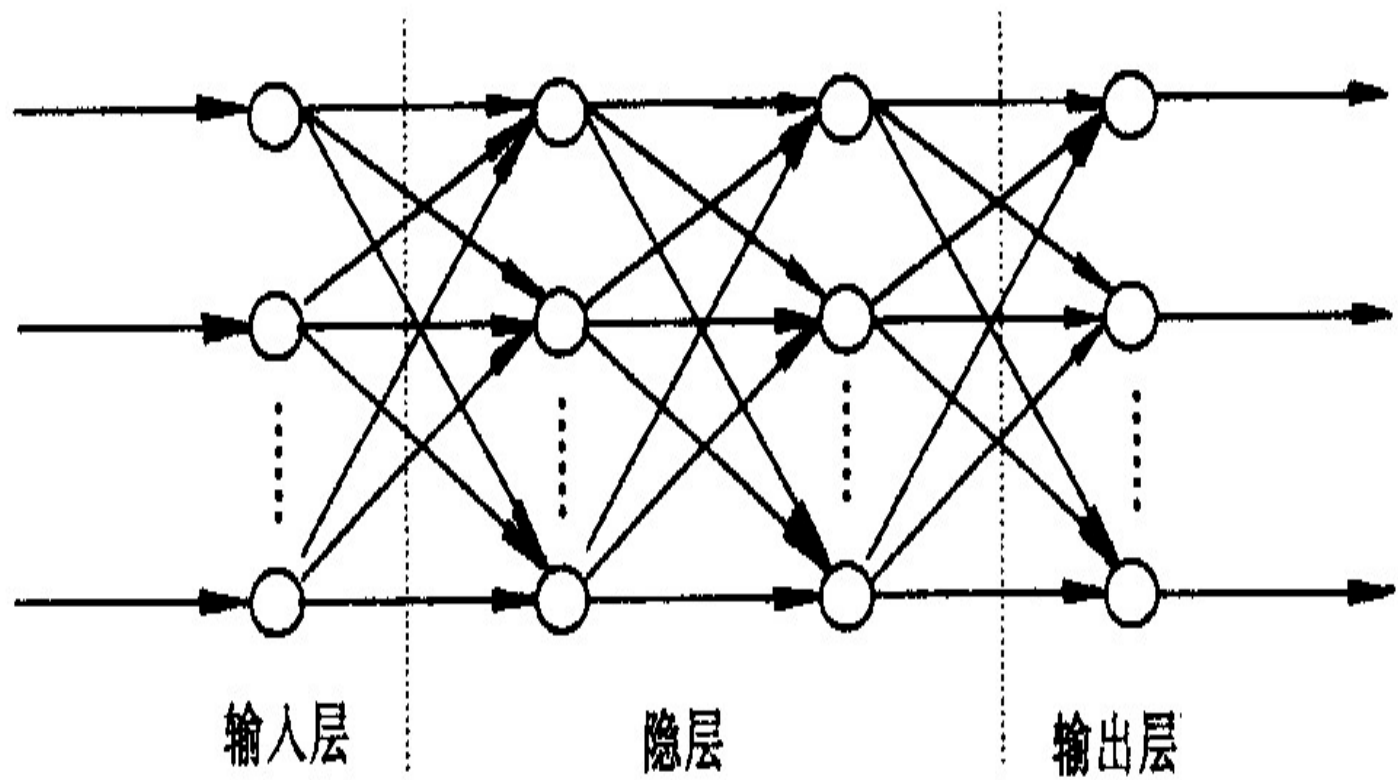
(f)

3、人工神经网络模型

- 人工神经网络是以工程技术手段来模拟人脑神经元网络的结构与特征的系统。利用人工神经元可以构成各种不同拓扑结构的神经网络，它是生物神经网络的一种模拟和近似。就神经网络的主要连接型式而言，已有许多种不同的神经网络模型，其中前馈型网络和反馈型网络是两种典型的结构模型。

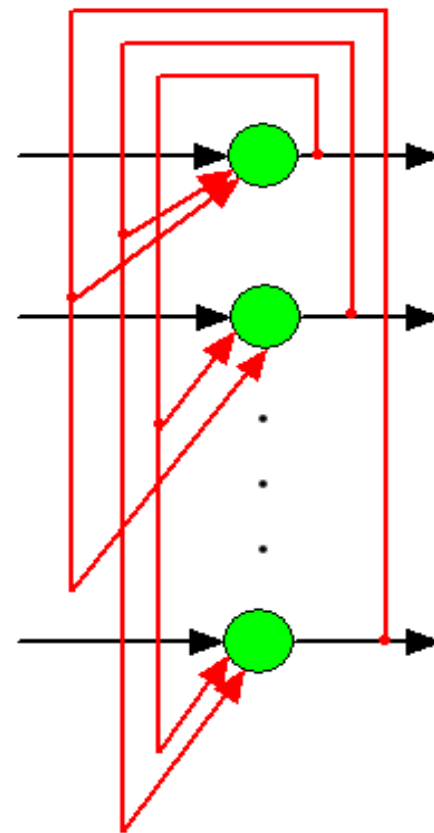
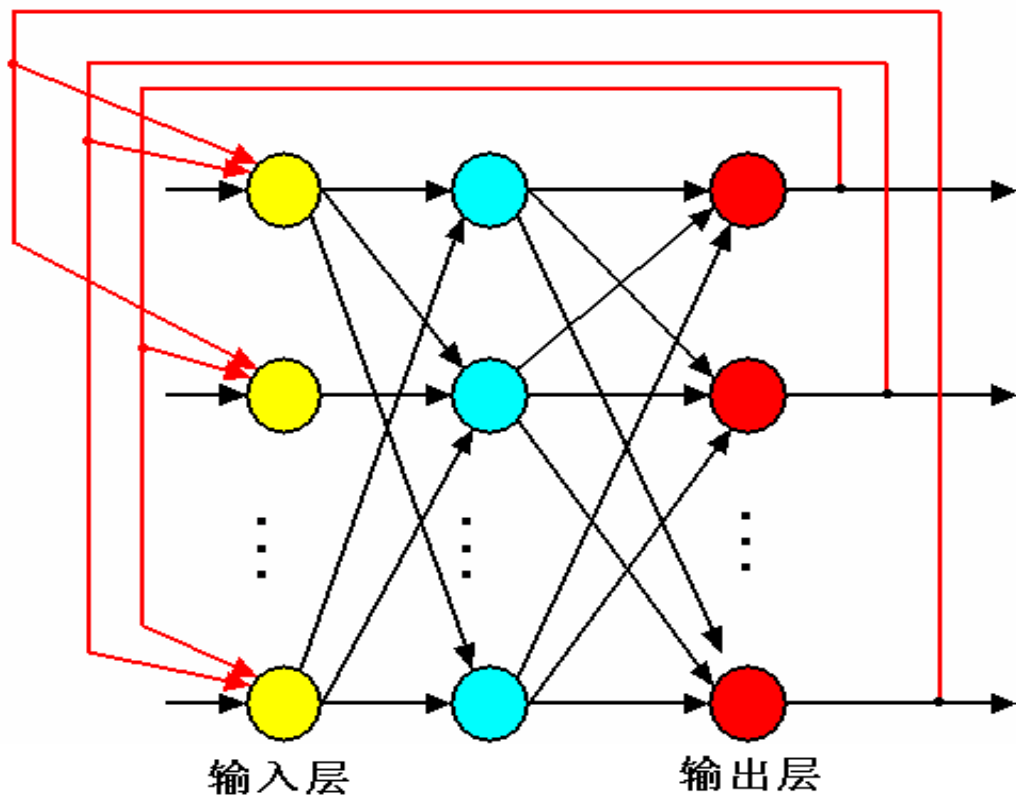
(1) 前馈型神经网络

- **前馈型神经网络**，又称**前向网络**（Feedforward NN）。
如图所示，神经元分层排列，有**输入层**、**隐层**（亦称中间层，可有若干层）和**输出层**，**每一层的神经元只接受前一层神经元的输入**。
- **前馈网络是一种强有力的学习系统**，其结构简单而易于编程；从系统的观点看，前馈网络是一静态非线性映射，通过简单非线性处理单元的复合映射，可获得复杂的非线性处理能力，但缺乏丰富的动力学行为。**大部分前馈网络都是学习网络**，它们的分类能力和模式识别能力一般都强于反馈网络，**典型的前馈网络有感知器网络、BP 网络等**。



(2) 反馈型神经网络

- **反馈型神经网络**（Feedback NN）的结构如图所示。是指允许采用反馈连接方式所形成的神经网络，一个神经元的输出可以被反馈到同层或前层神经元重新作为输入。
- 反馈神经网络是一种反馈动力学系统，它需要工作一段时间才能达到稳定。**Hopfield**神经网络是反馈网络中最简单且应用广泛的模型，它具有联想记忆功能。



(Hopfield神经网络)

4、神经网络的学习方法

学习方法是体现人工神经网络智能特性的主要标志，离开了学习算法，人工神经网络就失去了自适应、自组织和自学习的能力。

目前神经网络的学习方法有多种，按有无导师来分类，可分为：

- 有教师学习（Supervised Learning）
- 无教师学习（Unsupervised Learning）
- 再励学习（Reinforcement Learning）

有教师的学习方式：网络的输出和期望的输出（即教师信号）进行比较，然后根据两者之间的差异调整网络的权值，最终使差异变小。

无教师的学习方式：输入模式进入网络后，网络按照一预先设定的规则（如竞争规则）自动调整权值，使网络最终具有模式分类等功能。

再励学习是介于上述两者之间的一种学习方式。再励学习没有教师信号，只能获得行为结果的好坏评价信息，利用评价信息来改善行为。这是模仿生物通过“**行为—评价—改进**”的方式适应环境。

神经网络中几种基本学习方法

(1) Hebb学习规则

- 两个神经元同时处于激发状态时，它们之间的连接强度将得到加强，其数学描述被称为Hebb学习规则，即

$$w_{ij}(k+1) = w_{ij}(k) + I_i I_j$$

其中， $w_{ij}(k)$ 为连接从神经元 i 到神经元 j 的当前权值， I_i ， I_j 为神经元的激活水平。

- Hebb学习规则是一种无教师的学习方法，它只根据神经元连接间的激活水平改变权值，因此这种方法又称为相关学习或并联学习。

(2) Delta (δ) 学习规则

- δ 规则实现误差准则函数 E 中的梯度下降，使误差函数达到最小值。其基本思想是**沿着 E 的负梯度方向不断修正权值，直到 E 达到最小**。
- 简单的 δ 学习规则只适用于线性可分函数，无法用于多层网络。**BP**网络的学习算法称为**BP**算法，是在 δ 规则基础上发展起来的，可在多网络上有效地学习。

(3) 概率式学习

- 从统计力学、分子热力学和概率论中关于系统稳态能量的标准出发，进行神经网络学习，使网络的目标函数达到最大（或最小）。（神经网络处于某一状态的概率主要取决于在此状态下的能量，能量越低，概率越大）

(4) 竞争式学习

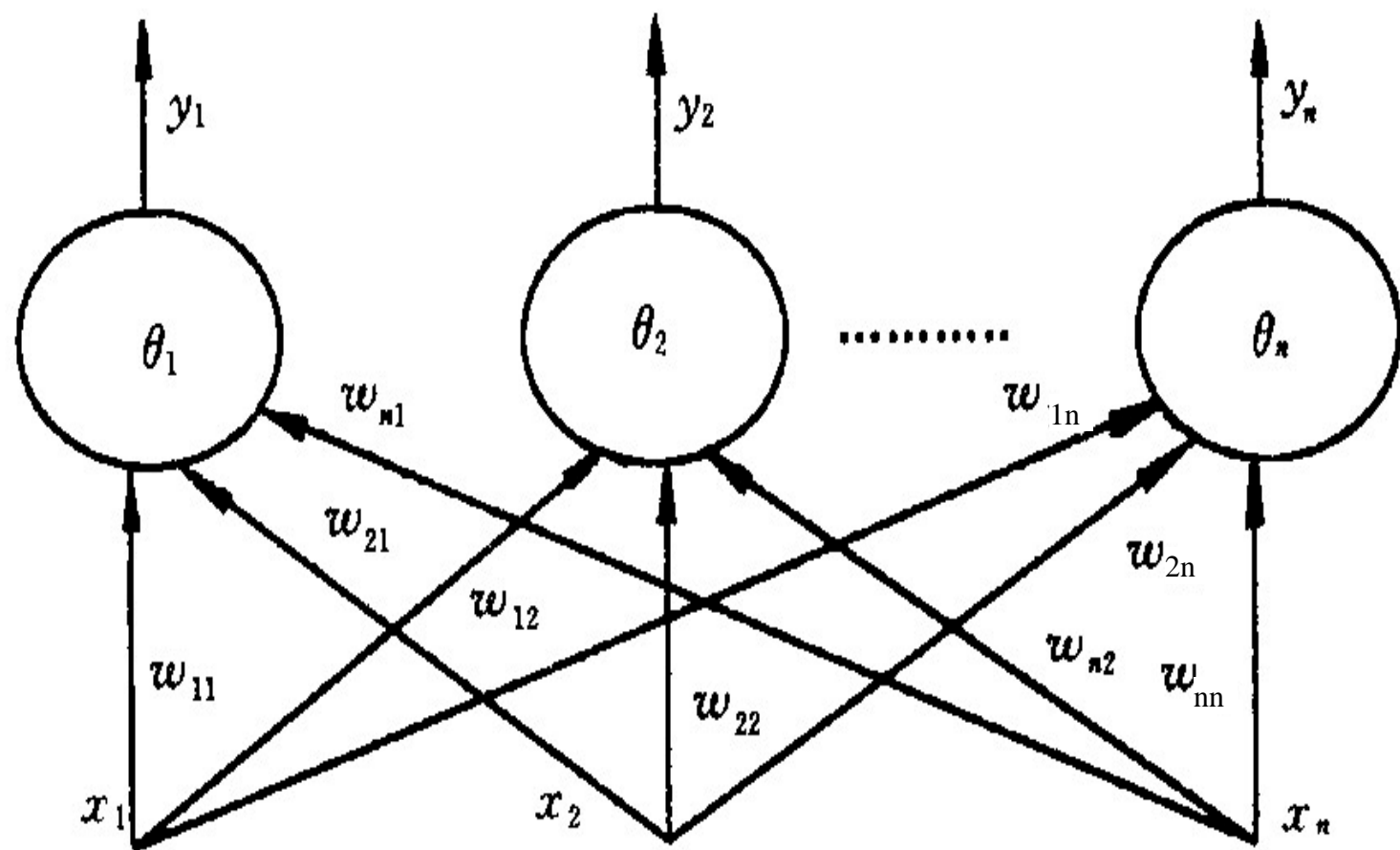
- 是指网络中某一组神经元互相竞争对外界刺激模式响应的权力，竞争中获胜的神经元其连接权值会向着对这一刺激模式竞争更有利的方向发展，抑制了竞争失败神经元对刺激模式的响应。竞争式学习属于无教师学习方式。

- 从上述几种学习规则可以看出，所谓神经网络的学习，主要是指通过一定的学习算法实现权值的调整，使其达到具有记忆、识别、分类、信息处理和问题优化求解等功能。

（二）前向神经网络

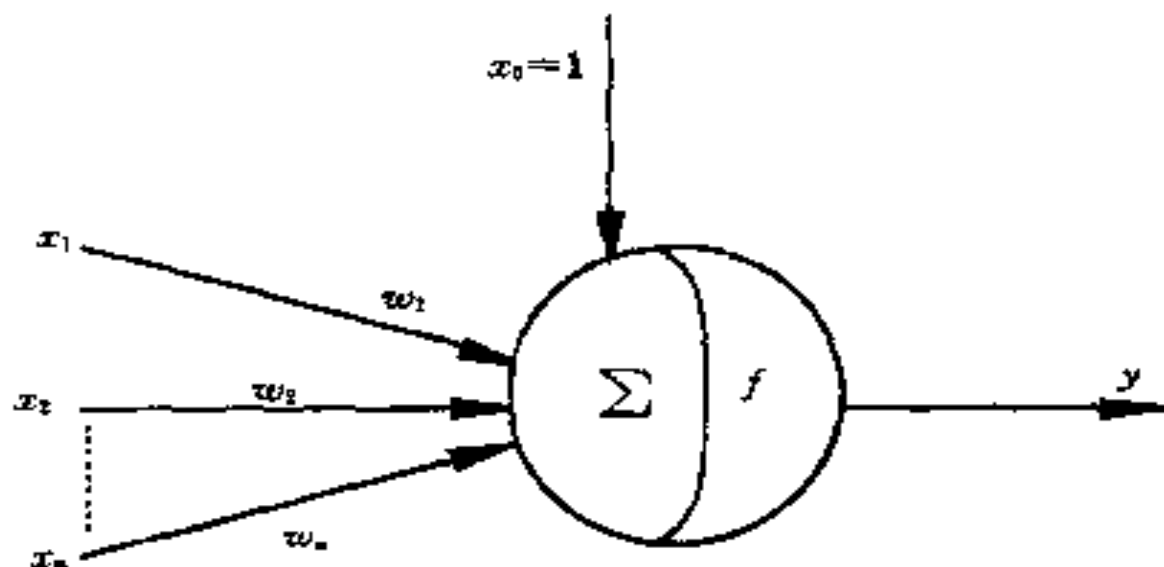
1、感知器网络

- 感知器（**perceptron**）是一个具有单层神经元的神经网络，并由线性阈值元件组成，是最简单的前向网络。它主要用于模式分类，单层的感知器网络结构如下图所示。



图中, $X = [x_1, x_2, \dots, x_n]^T$, 是输入特征向量; y_i ($i=1, 2, \dots, n$) 为输出量, 是按照不同特征分类的结果; w_{ji} 是 x_j 到 y_i 的连接权值, 此权值是可调整的, 因而有学习功能。

取其中的一个神经元来讨论。



为方便起见，令输入量 $x_0=1$ ，将阈值 θ 并入权中（因为 θ 值也需要学习）， $-\theta=w_0$ ，感知器的输入输出关系可表示为

$$y = f\left(\sum_{i=0}^n w_i x_i\right)$$

当其输入的加权和大于或等于阈值时，输出为 1，否则为 -1（或为 0）。

感知器的一种学习算法:

1) 随机地给定一组连接权 $w_i(0)$ (较小的非零值), 这里 $w_i(k)$ 为 k 时刻第 i 个输入上的权 ($1 \leq i \leq n$), $w_0(k)$ 为 k 时刻的阈值。

2) 输入一组样本 $X = (x_0, x_1, \dots, x_n)$ 和期望的输出 d (亦称之为教师信号)。如果 $X \in A$ 类 (某一类), 则 $y_d = 1$; 如果 $X \in B$ 类 (另一类), 则 $y_d = -1$ 。

3) 计算感知器实际输出:

$$y(k) = f\left(\sum_{i=0}^n w_i(k)x_i\right) = \begin{cases} 1, & \text{当 } \sum_{i=0}^n w_i(k)x_i \geq 0 \\ -1, & \text{当 } \sum_{i=0}^n w_i(k)x_i < 0 \end{cases}$$

$$(x_0 = 1, \quad w_0(0) = -\theta)$$

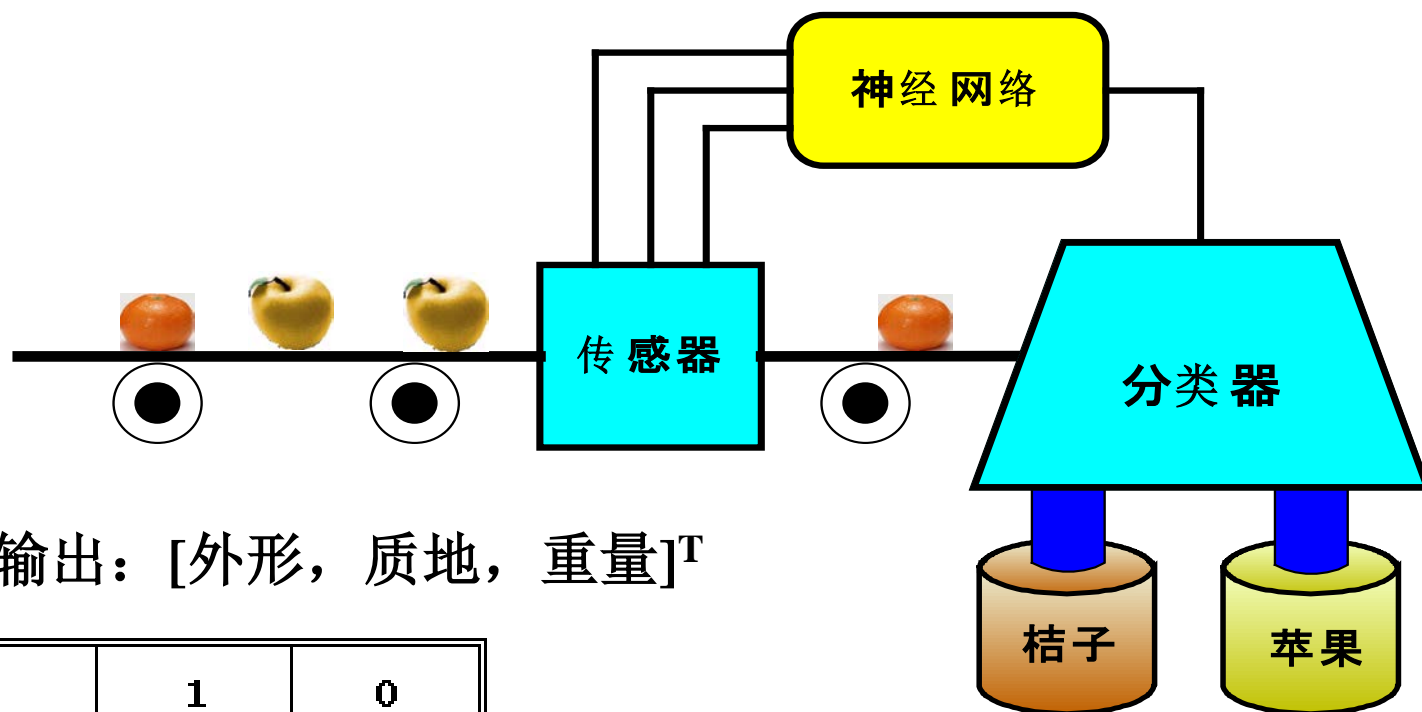
4) 按下式修正权值:

$$w_i(k+1) = w_i(k) + \eta[d(k) - y(k)]x_i$$

$$i = 0, 1, 2, \dots, n$$

其中, $w_i(k)$ 为当前权值; $d(k)$ 为导师信号; $y(k)$ 为感知器的输出值; η 为学习速率 ($0 < \eta < 1$), η 选取太小学习太慢, η 太大会影响 $w_i(k)$ 的稳定, 即引起震荡。

5) 选取另外一组样本, 重复上述 2) ~ 4) 的过程, 直到权值对一切样本均稳定不变为止, 学习过程结束。



传感器输出: [外形, 质地, 重量]^T

	1	0
外形	圆形	椭圆形
质地	光滑	粗糙
重量	< 1 磅	> 1 磅

标准桔子: $X^{(1)} = [1 \quad 0 \quad 1]^T$

标准苹果: $X^{(2)} = [0 \quad 1 \quad 0]^T$

单层感知器

- 用于两类模式分类时

相当于在高维样本空间中，用一个超平面将两类样本分开。

- 已证明

若输入的两类模式是线性可分集合（指存在一个超平面能将其分开），则算法一定收敛。

- 局限性

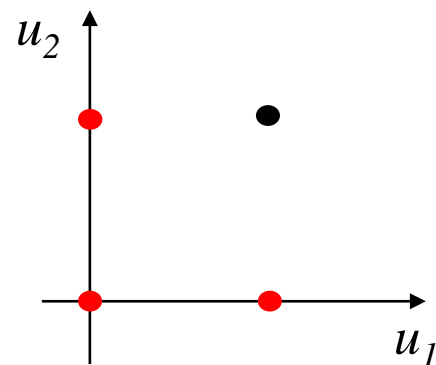
若输入模式为线性不可分集合，网络的学习算法不收敛，不能进行正确分类。

例1:

线性可分集合(1)

二维平面上的两类模式，见表。

u_1	u_2	y
0	0	0
1	0	0
0	1	0
1	1	1



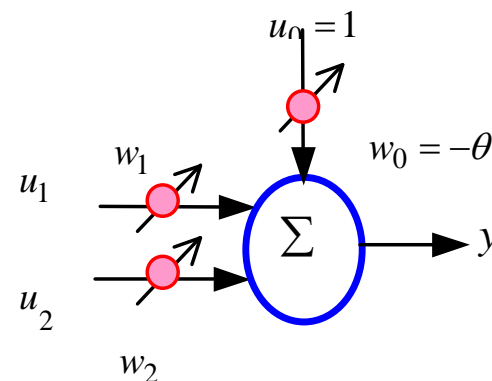
使用单个感知器分类线性可分集合(1):

用图所示二输入/单输出单层感知器，输入输出描述：

$$y = f(w_1 u_1 + w_2 u_2 - \theta) = f(\bullet) = \begin{cases} 1 & , \bullet \geq 0 \\ 0 & , \bullet < 0 \end{cases}$$

即

$$y = \begin{cases} 1 & , w_1 u_1 + w_2 u_2 \geq \theta \\ 0 & , w_1 u_1 + w_2 u_2 < \theta \end{cases}$$



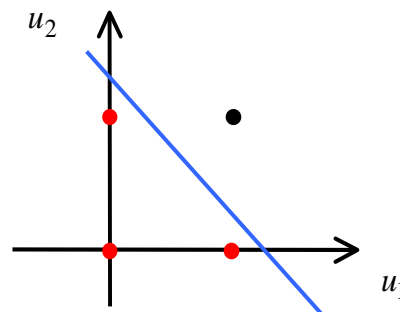
可见：输入输出为线性可分集合，因此一定可找到一条直线，将输入模式分为两类，此直线方程：

$$w_1 u_1 + w_2 u_2 - \theta = 0$$

则

$$u_2 = \frac{\theta}{w_2} - \frac{w_1}{w_2} u_1$$

见图，此直线与权值及阈值有关。

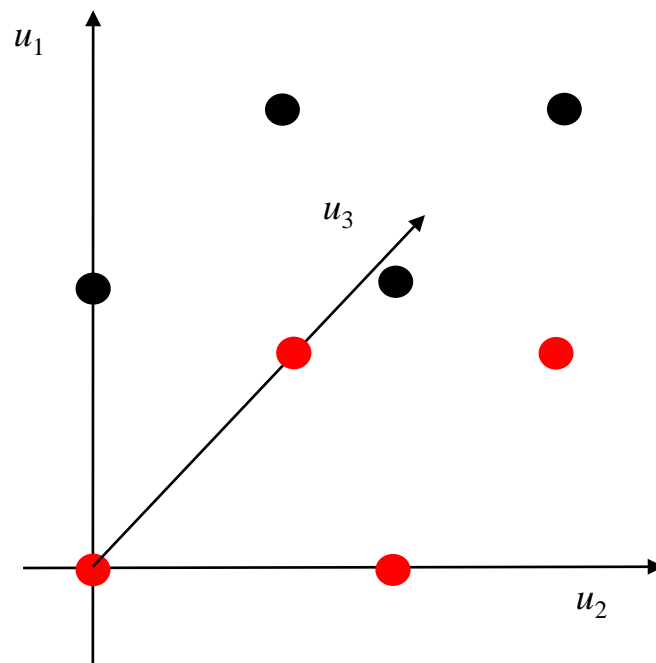


例2:

线性可分集合(2)

三维空间上的两类模式，见表

u_1	u_2	u_3	y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1



使用单个感知器分类线性可分集合(2):

用图所示三输入/单输出的单层感知器，输入输出：

$$y = f(w_1 u_1 + w_2 u_2 + w_3 u_3 - \theta) = f(\bullet) = \begin{cases} 1 & , \bullet \geq 0 \\ 0 & , \bullet < 0 \end{cases}$$

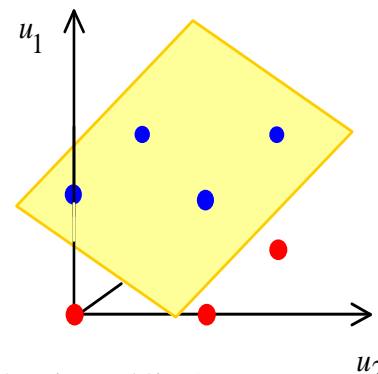
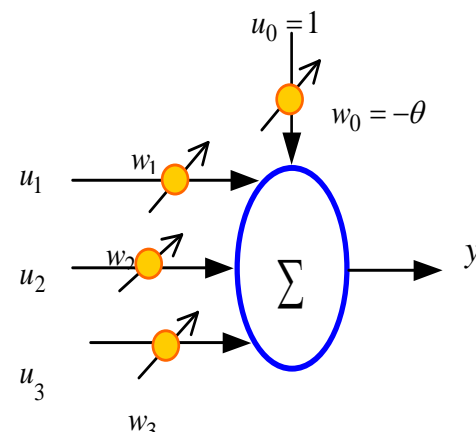
即
$$y = \begin{cases} 1 & , w_1 u_1 + w_2 u_2 + w_3 u_3 \geq \theta \\ 0 & , w_1 u_1 + w_2 u_2 + w_3 u_3 < \theta \end{cases}$$

可见，输入输出为线性可分集合，一定可找到一个平面，将输入模式分为两类，平面方程：

$$w_1 u_1 + w_2 u_2 + w_3 u_3 - \theta = 0$$

则
$$u_3 = \frac{\theta}{w_3} - \frac{w_1}{w_3} u_1 - \frac{w_2}{w_3} u_2$$

此平面与权值及阈值有关，见图。



可引伸到 $n > 3$ 维空间上的线性可分集合，一定可找到一超平面，将输入模式分为两类。由 n 输入/单输出的单层感知器实现。

例3:

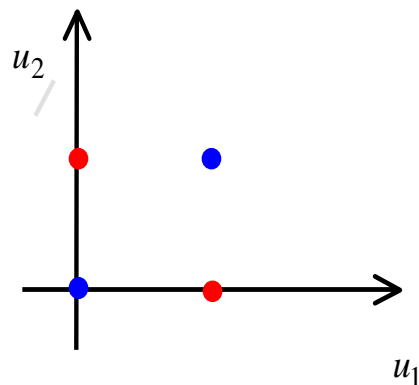
线性不可分集合

二维平面上的两类模式——**异或（XOR）问题**，见表。

二维平面中不存在一条直线，将输入模式分为两类，此输入模式称**线性不可分集合**。

单层感知器不能解决异或问题。

u_1	u_2	y
0	0	0
0	1	1
1	0	1
1	1	0



多层感知器

在输入和输出层间加一或多层隐单元，构成多层感知器（多层前馈神经网络）。

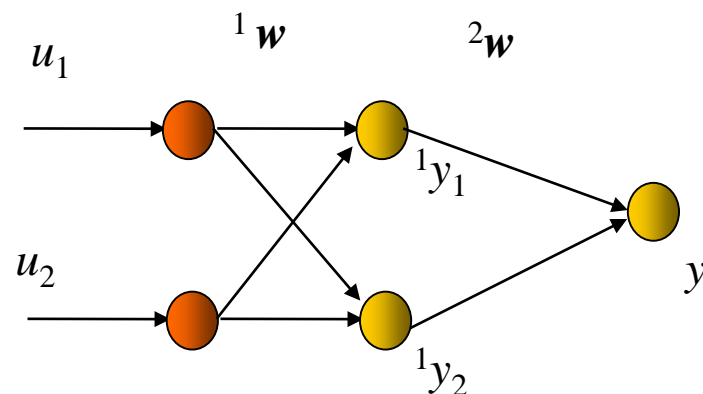
加一层隐节点（单元）为三层网络，可解决异或（XOR）问题，见图。
由输入 $\mathbf{u} = (u_1, u_2)$ 得到两个隐节点、一个输出层节点的输出：

$${}^1y_1 = f[{}^1w_{11}u_1 + {}^1w_{12}u_2 - {}^1\theta_1]$$

$${}^1y_2 = f[{}^1w_{21}u_1 + {}^1w_{22}u_2 - {}^1\theta_2]$$

$$y = f[{}^2w_1z_1 + {}^2w_2z_2 - \theta]$$

$$f[\bullet] = \begin{cases} 1, & \bullet \geq 0 \\ 0, & \bullet < 0 \end{cases}$$



因此得到

$${}^1y_1 = \begin{cases} 1, & {}^1w_{11}u_1 + {}^1w_{12}u_2 \geq \theta_1 \\ 0, & {}^1w_{11}u_1 + {}^1w_{12}u_2 < \theta_1 \end{cases}$$

$${}^1y_2 = \begin{cases} 1, & {}^1w_{21}u_1 + {}^1w_{22}u_2 \geq \theta_2 \\ 0, & {}^1w_{21}u_1 + {}^1w_{22}u_2 < \theta_2 \end{cases}$$

$$y = \begin{cases} 1, & {}^2w_1{}^1y_1 + {}^2w_2{}^1y_2 \geq \theta \\ 0, & {}^2w_1{}^1y_1 + {}^2w_2{}^1y_2 < \theta \end{cases}$$

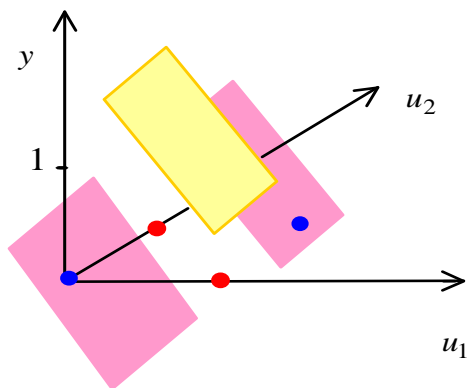
三层感知器解决异或问题

设网络有如下一组权值和阈值，可得各节点的输出：

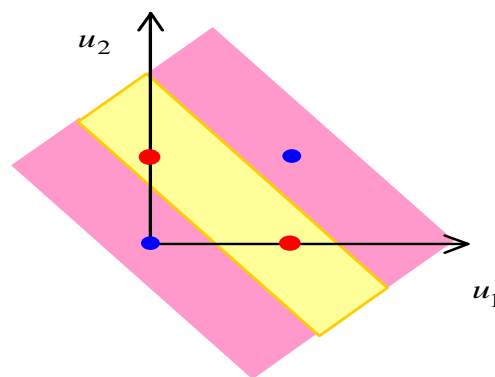
$$^1y_1 = f[1 \cdot u_1 + 1 \cdot u_2 - 0.5]$$

$$^1y_2 = f[(-1) \cdot u_1 + (-1) \cdot u_2 - (-1.5)]$$

$$y = f[1 \cdot ^1y_1 + 1 \cdot ^1y_2 - 1.2]$$



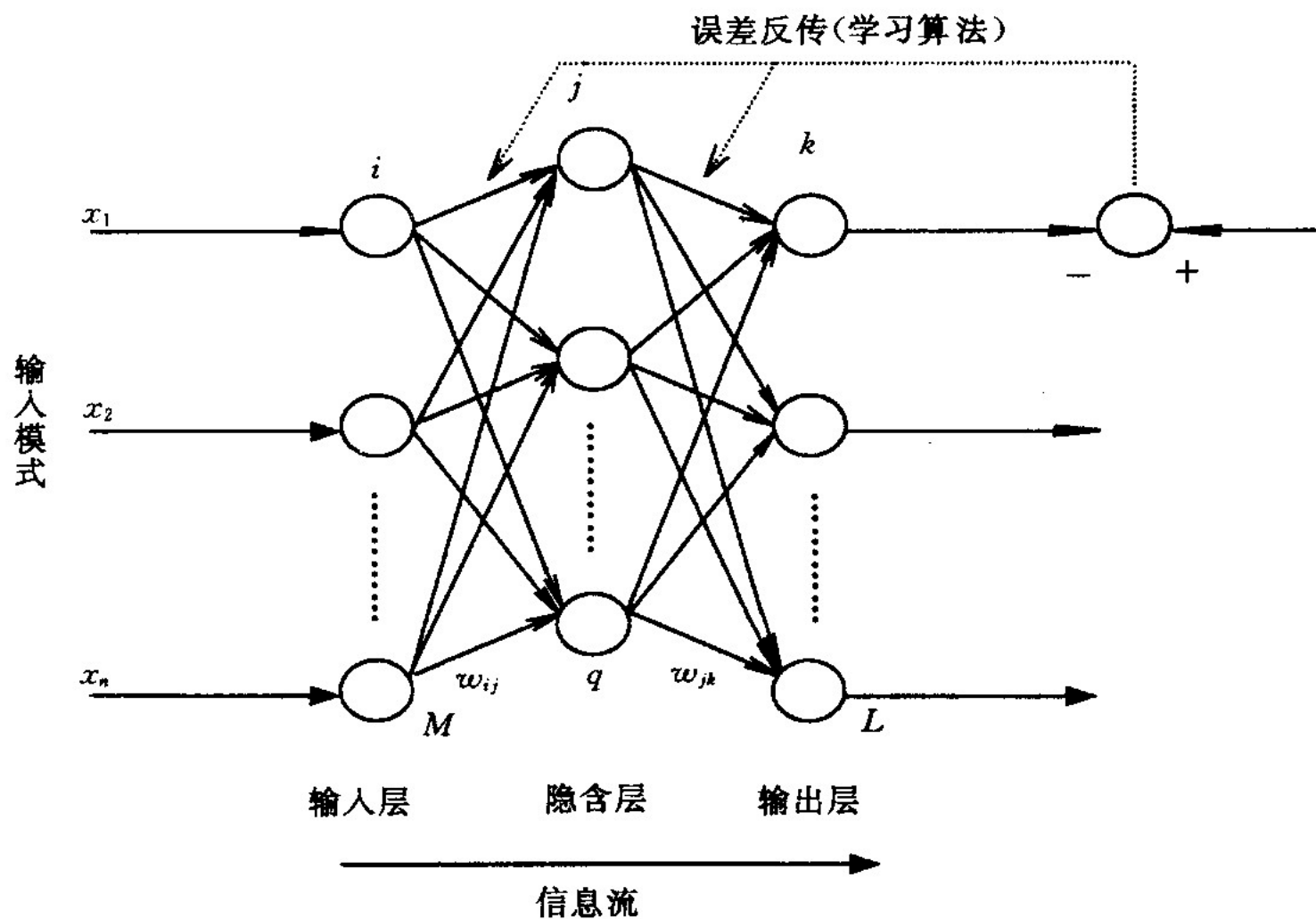
y 与 u_1 、 u_2



模式空间分两类

2、BP网络

- 误差反向传播神经网络，简称BP网络（Back Propagation），是一种单向传播的多层前向网络。在模式识别、图像处理、系统辨识、函数拟合、优化计算、最优预测和自适应控制等领域有着较为广泛的应用。如图是BP网络的示意图。

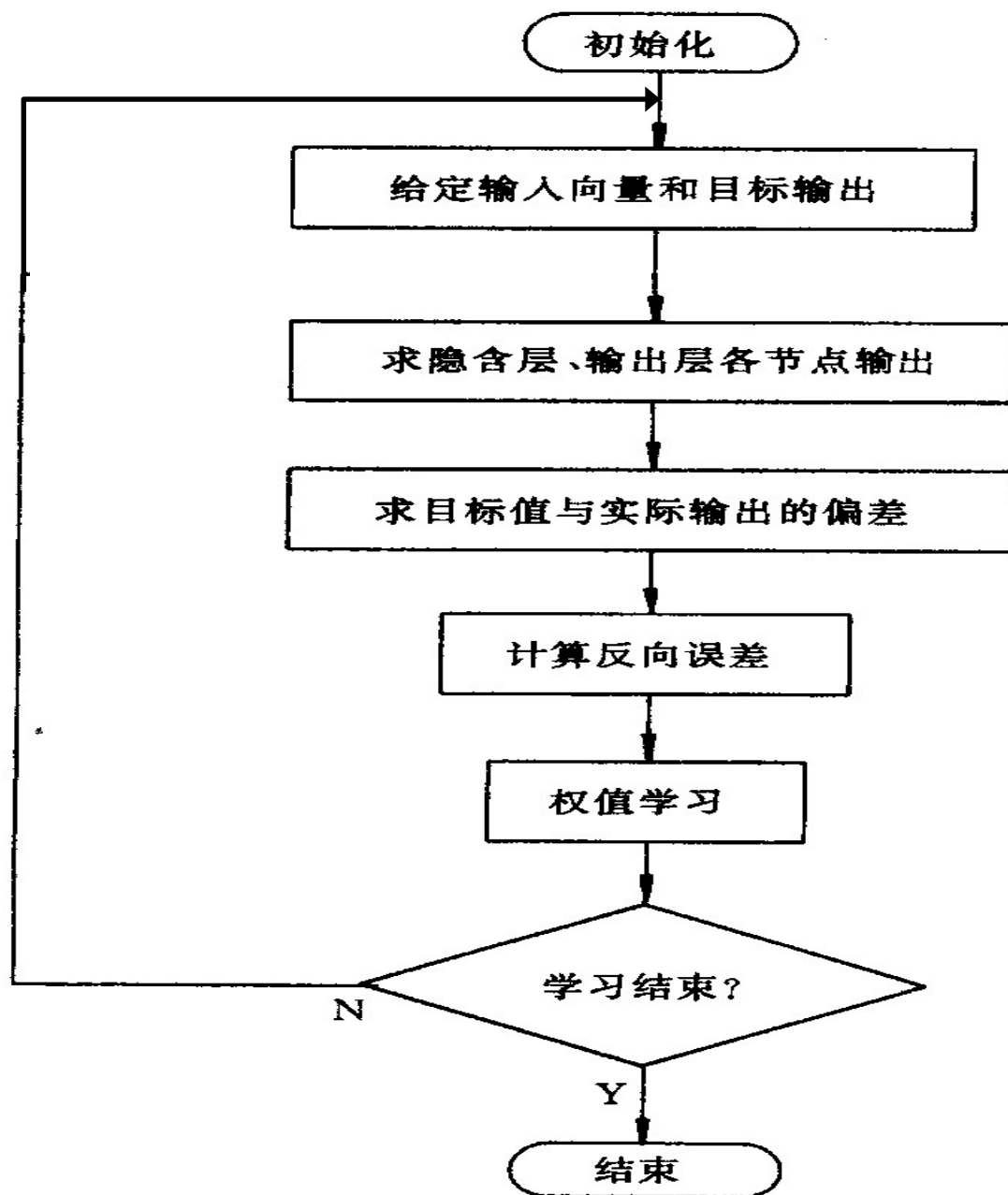


有 M 个输入节点，输入层节点的输出等于其输入。输出层有 L 个输出节点，网络的隐含层有 q 个节点， w_{ij} 是输入层和隐含层节点之间的连接权值。 w_{jk} 是隐含层和输出层节点之间的连接权值，隐含层和输出层节点的输入是前一层节点的输出的加权和，每个节点的激励程度由它的激励函数来决定。

- **误差反向传播的BP算法简称BP算法**，其基本思想是最小二乘算法。它采用梯度搜索技术，以期使网络的实际输出值与期望输出值的误差均方值为最小。
- BP算法的学习过程**由正向传播和反向传播组成**。在正向传播过程中，输入信息从输入层经隐含层逐层处理，并传向输出层，每层神经元（节点）的状态只影响下一层神经元的状态。**如果在输出层不能得到期望的输出，则转入反向传播，将误差信号沿原来的连接通路返回，通过修改各层神经元的权值，使误差信号最小。**

BP学习算法的计算步骤:

- 1) 初始化。置所有权值为较小的随机数;
- 2) 提供训练集。给定输入向量和期望的目标输出向量;
- 3) 计算实际输出。计算隐含层、输出层各神经元输出;
- 4) 计算目标值与实际输出的偏差 E_p ;
- 5) 计算 $\Delta_p w_{jk}$
- 6) 计算 $\Delta_p w_{ij}$
- 7) 返回“2)” 重复计算, 直到误差满足要求为止。



在使用BP算法时，应注意的几个问题：

- 1) 学习开始时，各隐含层连接权系数的初值应以设置较小的随机数较为适宜。
- 2) 采用S型激发函数时，由于输出层各神经元的输出只能趋于1或0，不能达到1或0。在设置各训练样本时，期望的输出分量 d_{pk} 不能设置为1或0，以设置为0.9或0.1较为适宜。
- 3) 学习速率 η 的选择，在学习开始阶段， η 选较大的值可以加快学习速度。学习接近优化区时， η 值必须相当小，否则权系数将产生振荡而不收敛。

多层前向BP网络的优点：

- 1) 网络实质上实现了一个从输入到输出的映射功能，数学理论已证明它具有实现任何复杂非线性映射的功能。这使得它特别适合于求解内部机制复杂的问题；
- 2) 网络能通过学习带正确答案的实例集自动提取“合理的”求解规则，即具有自学习能力；

3、神经网络的训练

在实际应用中尚未找到较好的网络构造方法，确定网络的结构和权值参数，来描述给定的映射或逼近一个未知的映射，只能通过学习来得到满足要求的网络模型。

神经网络训练的具体步骤:

(1) 获取训练样本集

获取训练样本集是训练神经网络的第一步，也是十分重要和关键的一步。它包括训练数据的收集、分析、选择和预处理等。

(2) 选择网络类型与结构

神经网络的类型很多，需要根据任务的性质和要求来选择合适的网络类型。

(3) 训练与测试

利用获取的训练样本对网络进行反复训练，直至得到合适的映射结果。

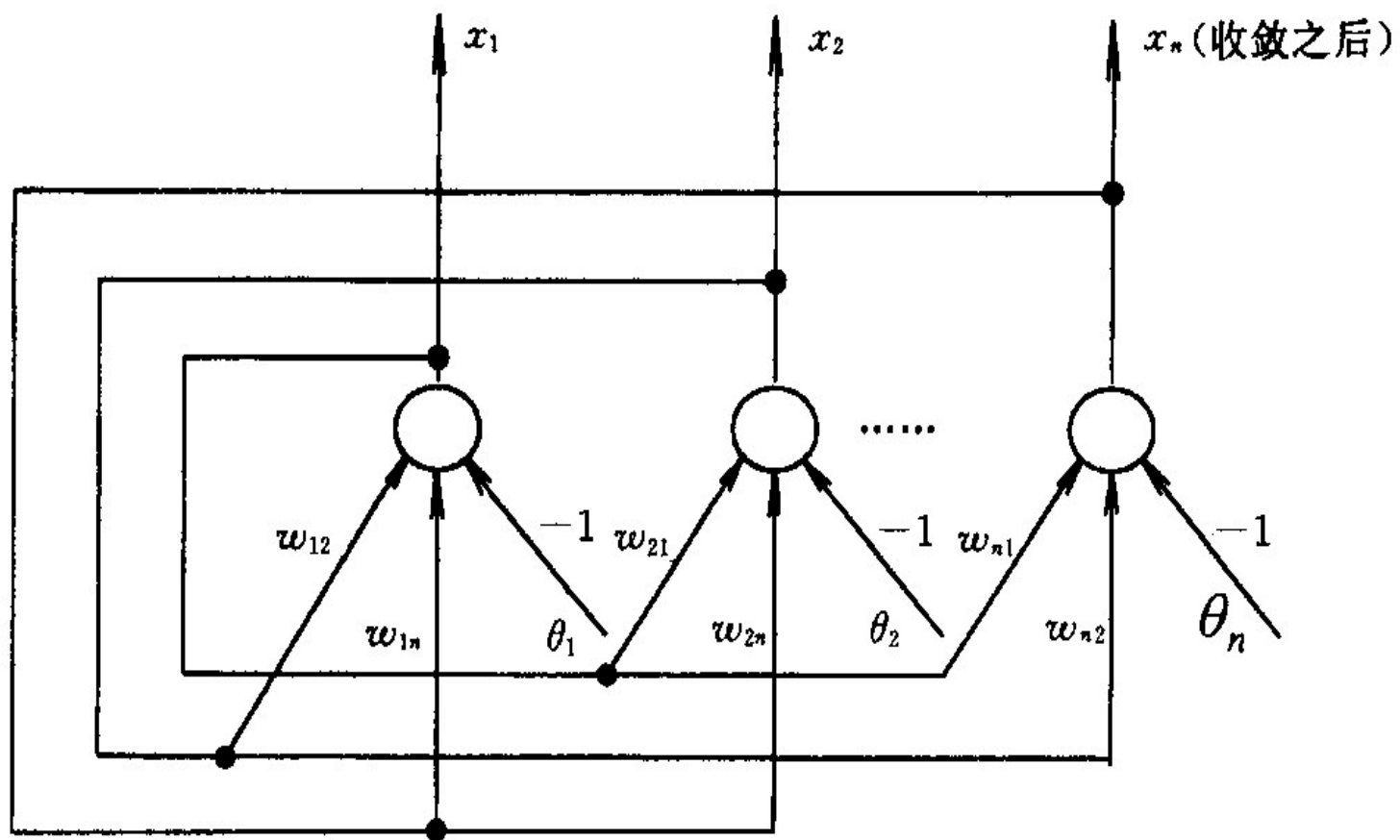
(三) 反馈神经网络

- **反馈网络**，又称**自联想记忆网络**，其目的是为了设计一个网络，储存一组平衡点，使得当给网络一组初始值时，网络通过自行运行而最终收敛到这个设计的平衡点上。
- 反馈网络能够表现出非线性动力学系统的动态特性。它所具有的主要特性为以下两点：
- **第一**、网络系统具有若干个稳定状态。当网络从某一初始状态开始运动，网络系统总可以收敛到某一个稳定的平衡状态；
- **第二**、系统稳定的平衡状态可以通过设计网络的权值而被存储到网络中。

1、离散Hopfield网络

(1) 网络的结构和工作方式

- 离散Hopfield网络是一个单层网络，有 n 个神经元节点，每个神经元的输出均接到其它神经元的输入。
- 各节点没有自反馈，每个节点都附有一个阈值。每个节点都可处于一种可能的状态（1或-1），即当该神经元所受的刺激超过其阈值时，神经元就处于一种状态（比如1），否则神经元就始终处于另一状态（比如-1）。



- **整个网络有两种工作方式**：即异步方式和同步方式。

1) 异步方式

每次只有一个神经元节点进行状态的调整计算，其他节点的状态均保持 不变。其调整次序可以随机选定，也可按规定次序进行。

2) 同步方式

所有神经元节点同时调整状态。

若稳态视为一个记忆样本，那么初态朝稳态的收敛过程就是寻找记忆样本的过程。初态可以认为是给定样本的部分信息，网络改变的过程可以认为是从部分信息找到全部信息，从而实现联想记忆的功能。

若将稳态与某种优化计算的目标函数相对应，并作为目标函数的极小点，那么初态朝稳态的收敛过程便是优化计算过程，该优化计算是在网络演变过程中自动完成的。

2、连续Hopfield网络

- 连续Hopfield 网络也是单层的反馈网络。其实质上是一个连续的非线性动力学系统，它可以用一组非线性微分方程来描述。当给定初始状态，通过求解非线性微分方程组即可求得网络状态的运行轨迹。若系统是稳定的，则它最终可收敛到一个稳定状态。

二、神经网络控制的结构

1、神经网络控制的基本思想

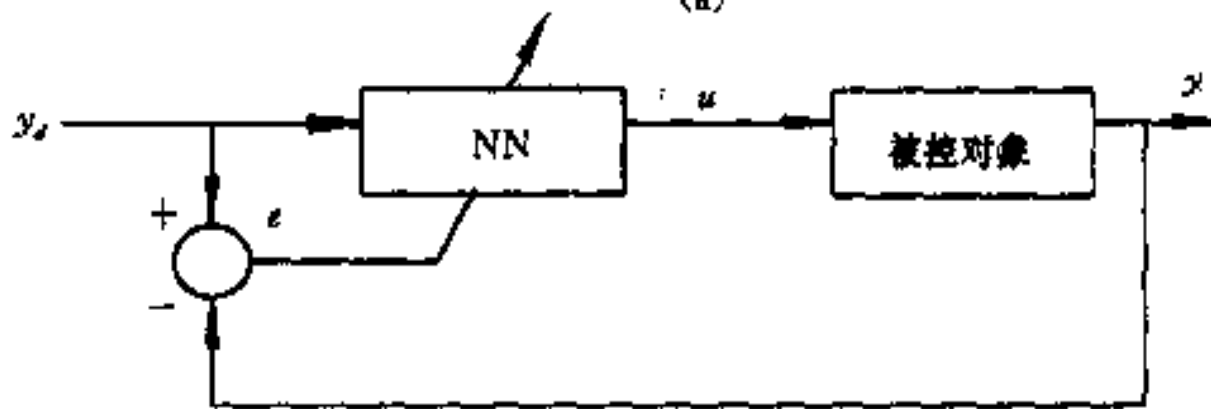
神经网络控制或神经控制是指在控制系统中，应用神经网络技术，对难以精确建模的复杂非线性对象进行神经网络模型辨识，或作为控制器，或进行优化计算，或进行推理，或进行故障诊断，或同时兼有上述多种功能。这样的系统称为基于神经网络的控制系统，称这种控制方式为神经网络控制。

- 尽管神经网络控制技术有许多潜在的优势，但单纯使用神经网络的控制方法的研究仍有待进一步发展。通常将人工神经网络技术与传统的控制理论或智能技术综合使用。**神经网络在控制中的作用有以下几种：**
 - (1)** 在传统的控制系统中用以动态系统建模，充当对象模型；
 - (2)** 在反馈控制系统中直接充当控制器的作用；
 - (3)** 在传统控制系统中起优化计算作用；
 - (4)** 与其他智能控制方法如模糊逻辑、遗传算法、专家控制等相融合。

下图给出了一般反馈控制的原理图，其中图(b)采用神经网络NN代替图(a)中的控制器。网络NN可以是前向多层网络，采用BP法。



(a)

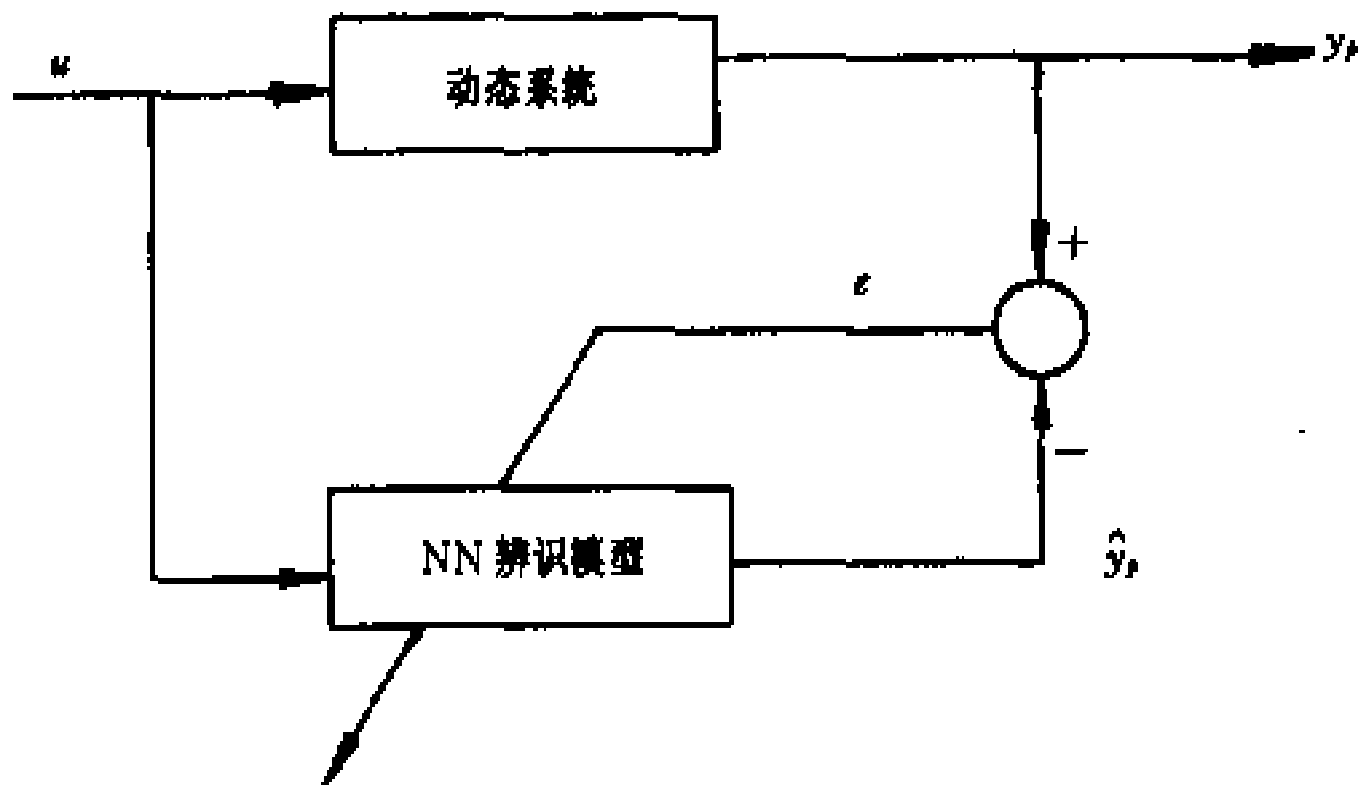


(b)

当用训练误差信号来调整网络中的连接权值而使 $e \rightarrow 0$ 时，就是网络的学习算法实现被控对象求逆模型的过程。可以说，被控对象或过程的模型求逆过程就是神经网络实现直接控制的基本思想。

2、正向建模

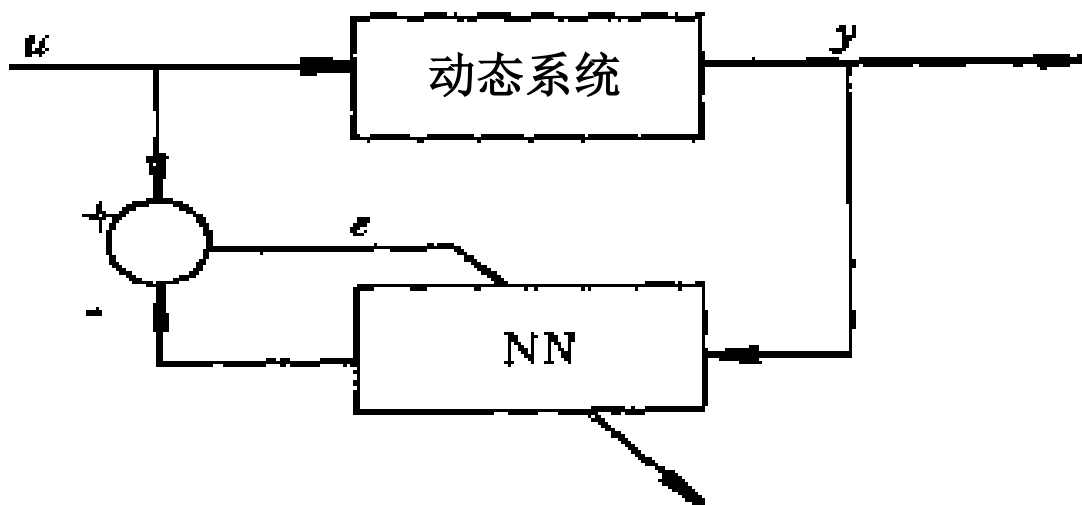
所谓**正向建模**，是指利用多层前馈神经网络，通过训练或学习，使其能够**表达系统正向动力学特性**的模型。下图给出了系统正向建模的一般结构示意图。其中**神经网络辨识模型与待辨识的动态系统并联，两者的输出误差，被用作网络的训练信号**。显然这是一个典型的有监督学习问题，实际系统作为教师，向神经网络提供学习算法所需的期望输出。



正向建模的一般结构

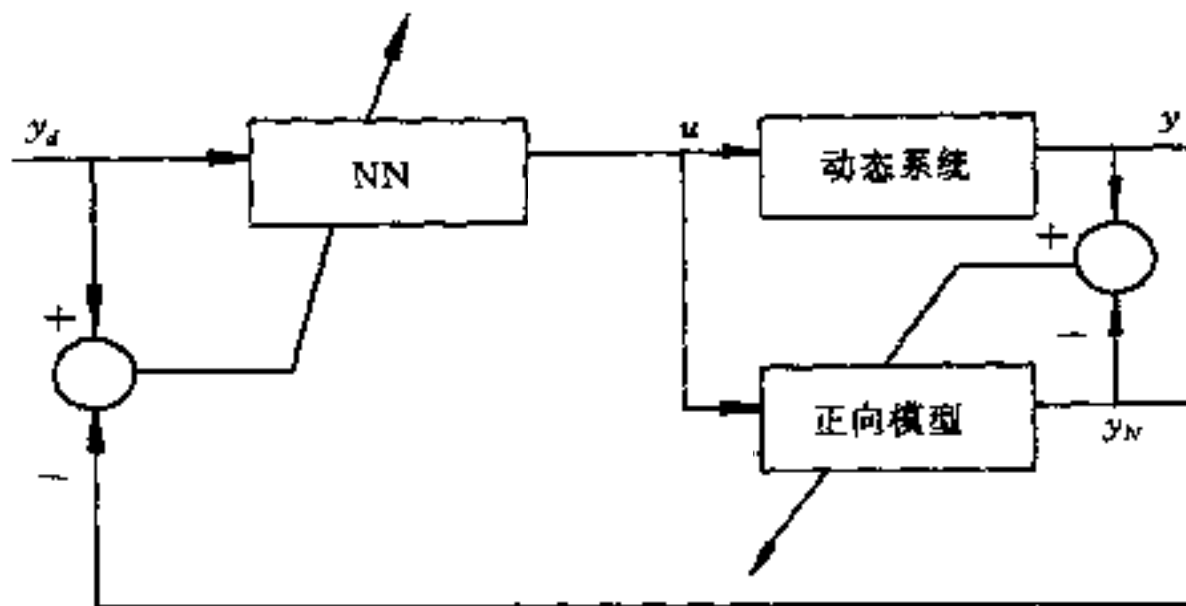
3、逆模型

下图是一种直接逆建模结构。待辨识系统的输出作为网络的输入，网络输出与系统输入比较，相应的输入误差用来进行训练，因而网络将通过学习建立系统的逆模型。



直接逆建模

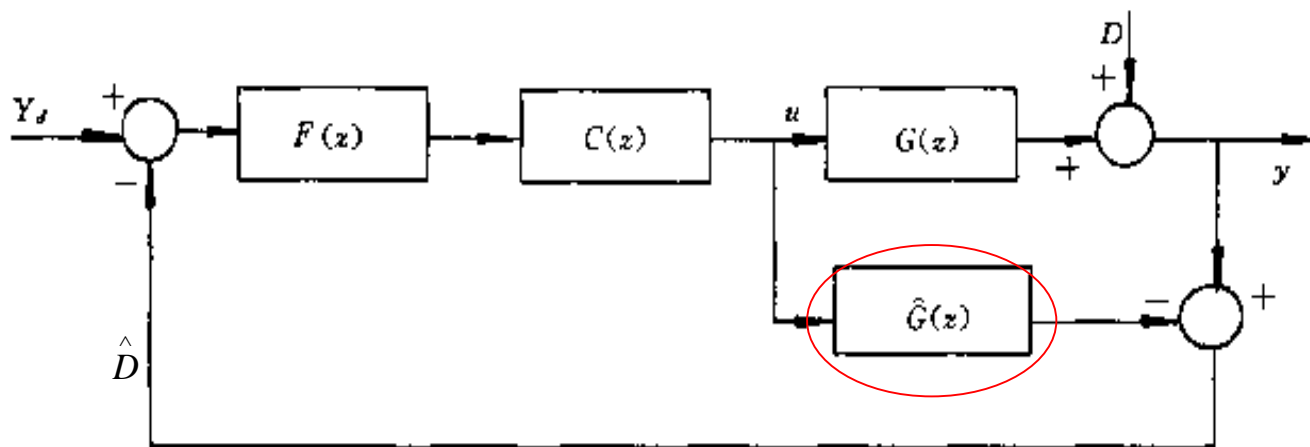
由于上述学习过程不是目标导向的，在实际工作中系统输入也不可能预先定义，因此采用下图所示的**正—逆建模结构**。这时待辨识的网络**NN**位于系统前面，并与之串联。网络的输入为系统的期望输出 $y_d(k)$ ，训练误差可以是期望输出与系统实际输出 $y(k)$ 之差或者为期望输出与已建模的正向模型输出 y_N 之差。



正-逆建模

4、神经网络内模控制

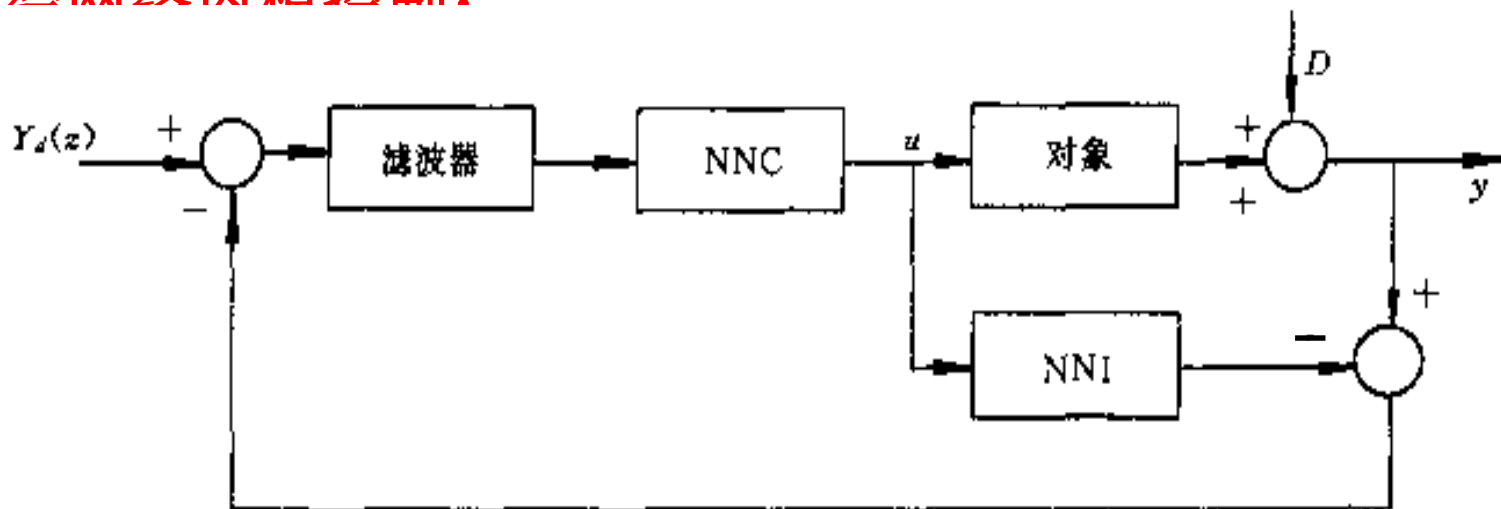
内模控制是一种采用系统对象的内部模型和反馈修正的预测控制，有较强的鲁棒性，在线调整方便，已被发展为非线性控制的一种重要方法。



带有滤波器的内模控制系统

反馈量为扰动估计量。在实际应用中，考虑到模型与对象失配时的影响，通常在控制器前附加一个滤波器 $F(z)$ ，可提高系统的鲁棒性。

神经网络内模控制.



神经网络内模控制系统

分别用两个神经网络NNC和NNI取代了 $C(z)$ 和 $\hat{G}(z)$ ，**NNC**称为神经网络控制器，**NNI**称为神经网络状态估计器。图中的神经网络状态估计器**NNI**用于充分逼近被控对象的动态模型，相当于正向模型。神经网络控制器**NNC**学习被控对象的逆动态特性。

NNI作为被控对象的近似模型与实际对象并行设置，它们的差值用于反馈，同期望的给定值之差经线性滤波器处理后送给**NNC**，经过多次训练，它将间接地学习对象的逆动态特性。从而系统误差将趋于零。

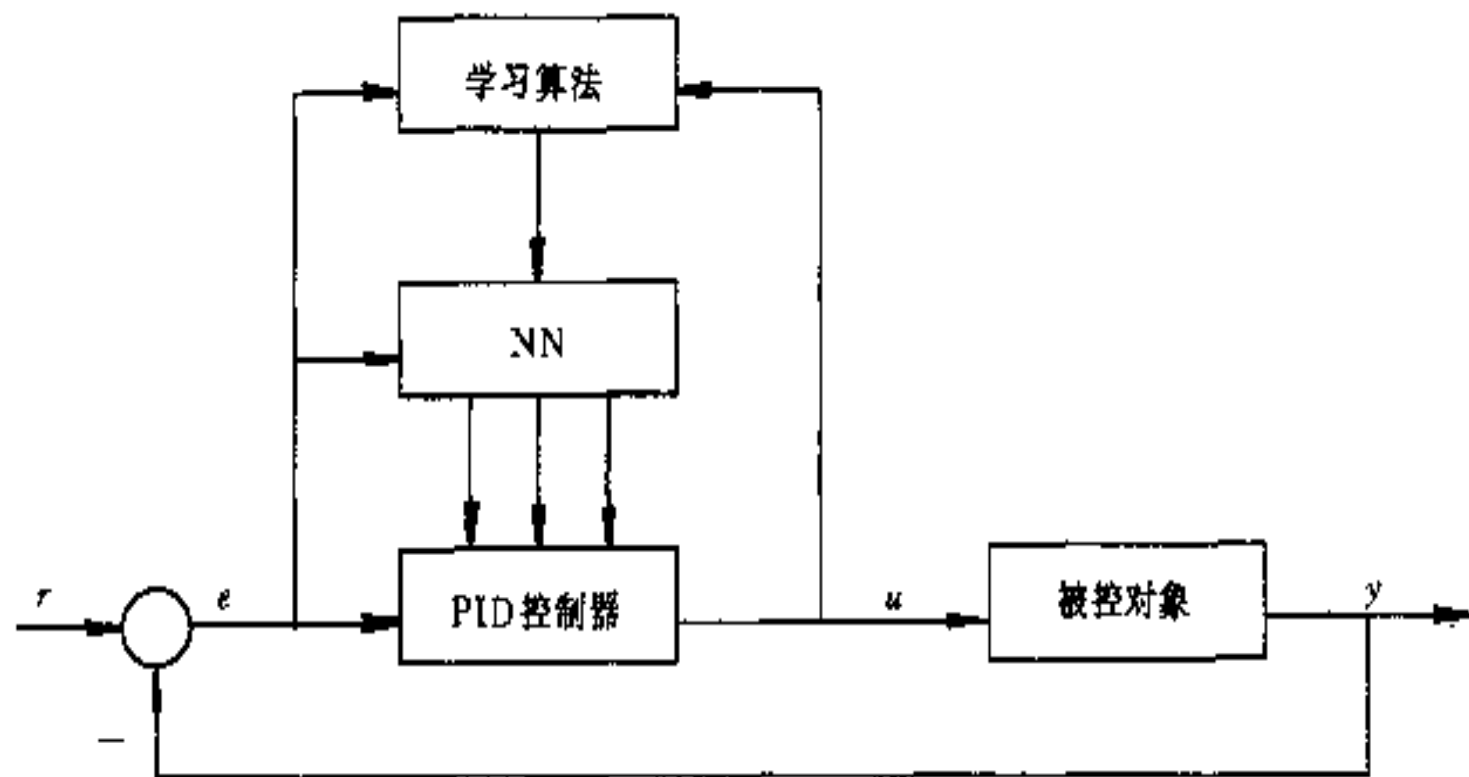
5、神经网络PID控制

PID控制要取得好的控制效果，就必须对比例、积分和微分三种控制作用进行调整以形成相互配合又相互制约的关系，这种关系不是简单的“线性组合”，可从变化无穷的非线性组合中找出最佳的关系。神经网络所具有的任意非线性表示能力，可以通过对系统性能的学习来实现具有最佳组合的PID控制。

BP神经网络具有逼近任意非线性函数的能力，而且结构和学习算法简单明确。通过神经网络自身的学习，可以找到某一最优控制律下的 P ， I ， D 参数。

基于BP神经网络的PID控制系统结构如下图所示，控制器由两部分组成：

- ① 经典的PID控制器：直接对被控对象进行闭环控制，并且 K_P, K_I, K_D 三个参数为在线整定；
- ② 神经网络NN：根据系统的运行状态，调节PID控制器的参数，以期达到某种性能指标的最优化。即使输出层神经元的输出状态对应于PID控制器的三个可调参数 K_P, K_I, K_D ，通过神经网络的自学习调整权系数，从而使其稳定状态对应于某种最优控制律下的PID控制器参数。



基于 BP 神经网络的 PID 控制系统结构

三、神经网络控制器的设计

- 对神经网络控制系统尚缺乏规范的设计方法。在实际应用中，根据受控对象及其控制要求，人们通常运用神经网络的基本原理和各种控制结构，设计出行之有效的神经控制系统。
- **神经网络控制器的设计一般包括以下内容：**
 - 建立受控对象的数学计算模型或知识表示模型；
 - 选择神经网络及其算法，进行初步辨识与训练；
 - 设计神经控制器，包括控制器结构、功能与推理；
 - 控制系统仿真实验，并通过实验结果改进设计。