第 5 章 神经网络建模方法

一、复习题

- 1、神经元计算模型（即 MP 模型）（结构、传递函数、数学模型等）
- 2、感知机模型（结构、传递函数、学习算法等）
- 3、BP 网络（结构、输入输出变换关系、传递函数、学习算法、工作过程等）
- 基于单一训练样本的学习算法
- 基于一批训练样本的学习算法

二、人工神经网络简介

1 人工神经网络简介

- 1943 年，美国心理学家麦克洛奇（W. Mcculloch）和数学家皮兹（W . Pitts）提出了用神经元网络对信息进行处理的数学模型——神经元计算模型（即 MP 模型）。
- 1949 年, 赫布（D.O.Hebb）提出了神经元之间连接强度变化的学习规则——Hebb 规则，从而开创了神经元网络研究的新局面。
- 1957 年, 美国心理学家罗森布拉特（F.Rosenblate）提出的感知机（Perceptron ）模型是第一个真正的人工神经网络，具有自学习能力，第一次把神经网络的研究从纯理论的探讨付诸工程实践，掀起了人工神经元网络的第一次研究热潮。
- 罗森布拉特（F.Rosenblate）提出感知器模型是一种学习和自组织的心理学模型，给出了两层感知器的收敛定理。
- 感知器模型基本上符合神经生理学的知识，模型的学习环境是有噪声的，网络构造中存在随机连接，这符合动物学习的自然环境。
- 后来的一大类神经网络模型都是感知器模型的变形。利用感知机所提出的数学概念直至今天仍在模式识别中起着很大的作用。
- 1959 年, 威德罗(B. Widrow)等提出自适应线性元件网络（Adaptive Linear element, Adaline）——两层前馈感知机型网络，通过训练后可用于抵消通信中的回波和噪声。
- 1960 年，威德罗(B. Widrow)和 M. Hoff 提出 LMS （Least Mean Square 最小方差）算法的学习规则。
- 1969 年, 人工智能创始者之一明斯基（M. Minsky）出版了《感知器（Perceptron）》一书，指出感知器仅适合于线性样本的情况，对非线性样本如 XOR 异或问题，它解决不了，但也指出若增加隐结点，能够解决非线性样本问题，但他们没有提出增加隐结点的多层网络学习算法。

由于 Misky 在学术界的影响，使很多研究者转向以符号处理为基础的人工智能研究，从而使人工神经元网络的研究处于低潮。

在此期间，仍有为数不多的学者致力于神经网络的研究：

- 1969 年，Grossberg 等提出了自适应共振理论模型。
- 1972 年, Kohenen 提出自组织特征映射的理论模型, 称神经网络为联想存贮器，能完成记忆功能。
- 1976 年，Grossberg 在自组织神经网络方面的研究十分活跃。

————所有这些理论为神经网络的进一步发展奠定了理论基础。

1982 年，美国加州工学院物理学家 Hopfield 提出了仿人脑的神经网络模型——离散的神经网络模型（Hopfield 模型），用于联想记忆及优化计算，标志着神经网络的研究又进入了一个新高潮。

- 1984 年，Hopfield 又提出连续神经网络模型，开拓了计算机应用神经网络的新途径。
- Hopfield 引入了 Lyapunov 函数——计算能量函数，给出了网络稳定性判断依据，证明一个互连单元的神经网络系统将达到能量损耗最小的原理。

Hopfield 模型成功地解决了复杂的 NP 问题，即“推销员旅行路径”问题(计算量随城市的个数的增加而成指数增长)。

Hopfield 连续型模型能够通过集成电路来实现，这为神经元计算机奠定了基础。Hopfield 的研究掀起了人工神经元网络的第二次研究热潮。

在这次热潮中另一个有代表性的工作——1985 年鲁姆尔哈特（Rumelhart） 和麦克劳（McCelland） 提出的多层网络的误差传播学习算法——Back-Propagation 反向传播学习算法（BP 模型）。该模型引入多层隐结点，解决了非线性样本的问题。同一时期不少研究者也提出了很多成功的神经网络模型。这些成果大大促进了人工神经元网络的发展。

- BP 算法目前已成为迄今为止应用最普遍的神经网络学习算法。
- 1985 年，Terrence Sejnowski、Hinton 和 Ackley 对 Hopfield 模型引入随机机制，用统计物理学的概念和方法研究神经网络，提出 Boltzmann（玻尔兹曼）机，首次采用多层网络的学习算法，并用模拟退火过程来模拟外界环境。拓扑结构与一般二层前馈网络相似，运行过程与 BP 算法的感知器相似，学习算法不同。
- 1987 年 6 月在美国召开了第一次神经网络国际会议(ICNN)，宣告了神经网络计算机学科的诞生。
- 日本于 1985 年开始大规模研制神经元计算机，并把 1988 年定为神经元计算机元年。
- 美国和欧洲各国也都大力投资进行人工神经网络研究，从而形成了第三次神经元网络研究热潮。
- 我国也在大力开展神经元网络研究，每两年召开一次全国性学术会议，已有大量学术论文和专著出版。

目前，人工神经网络仍为人工智能领域的研究热点。

历史总结：

- 20 世纪 40 年代

兴奋与抑制型神经元模型（Mcculloch，Pitts：MP 模型）

神经元连接强度的修改规则（Hebb：Hebb 规则）

- 20 世纪 50 年代、60 年代

感知机（Rosenblatt）和自适应性元件（Widrow）

- 20 世纪 70 年代

Perceptron 一书出版（Minsky 和 Papert）研究处于低潮。

- 20 世纪 80 年代后

Rumelhart，Mcclelland 以及 Hopfield 等取得突破性进展

- 20 世纪 90 年代开始

功能柱和神经场模型的提出

1991 年-现在的问题：

1）应用面还不够宽。2）结果不够精确。3）存在可信度的问题。

研究：

1）开发现有模型的应用，并在应用中根据实际运行情况对模型、算法加以改造，以提高网络的训练速度和运行的准确度。

2）充分发挥 AI 与 ANN 各自的优势是一个有效方法。

3）希望在理论上寻找新的突破，建立新的专用/通用模型和算法。

4）进一步对生物神经系统进行研究，不断地丰富对人脑的认识。

通过人工神经网络研究历史，可以知道：

1）遵从神经元工作原理，采用定量方法研究人工神经网络模型。如神经元模型。

2）人工神经网络拓扑结构：两层前馈网络结构，含隐含层的多层网络结构。感知机，BP 网络等。

3）人工神经网络学习算法：Hebb 学习规则，LMS（Least Mean Square 最小方差）算法，BP 算法（反向传播学习算法等）。

4）人工神经网络网络稳定性：Hopfield 模型。

5）人工神经网络应用：解决非线性问题，用于优化计算，增强联想记忆和存储能力等。

1.1 人工神经元模型（书 P175）

神经元的数学模型：

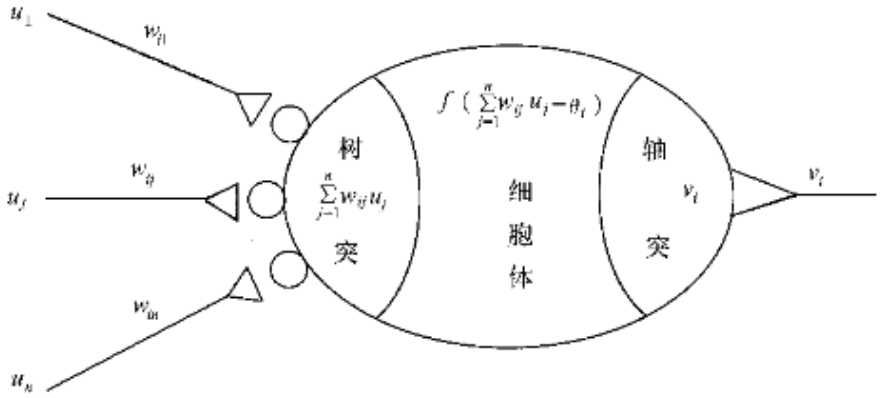


图 8.2　人工神经元的数学模型

其中：

u_j——输入量，每一个处理单元都有许多输入量，对每一个输入量都相应有一个相关联的权重；

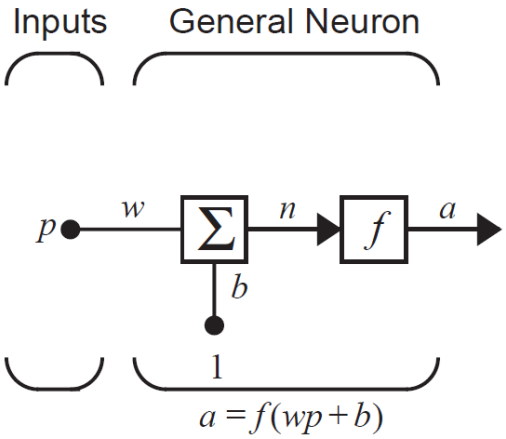
w_{ij}——权重，外面神经元与该神经元的连接强度，是变量。初始权重可以由一定算法确定，也可以随意确定。权重的动态调整是学习中最基本的过程，随学习规则变化，目的是调节权重以减少输出误差；

θ_i——阈值；

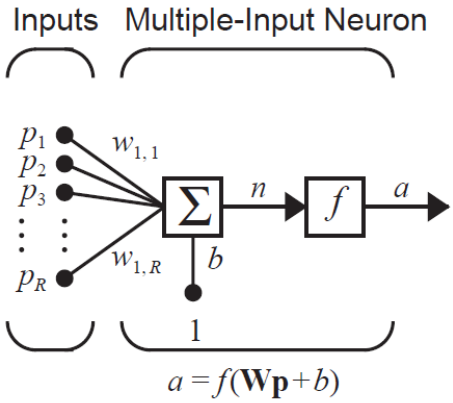
v_i——输出；

$f(x)$——该神经元的传递函数，就是将输入激励转换为输出响应的数学表达式

单输入神经元模型



多个输入神经元



权值下标：权值矩阵元素下标的第一个下标表示权值相应连接所指定的目标神经元编号，第二个下标表示权值相应连接的源神经元编号。

MP 模型:

$f(x)$——该神经元的传递函数，就是将输入激励转换为输出响应的数学表达式，有如下常用的形式：

(1) 阶梯函数(图 8.3)

$$f(x) = \begin{cases} 1, & x \geqslant 0 \\ 0, & x < 0 \end{cases}$$

常称此种神经元为 M－P 模型，即每个神经元的状态 v_i 满足 M－P 方程：

$$v_i = f\left(\sum_j w_{ij}u_j - \theta_i\right) = \begin{cases} 0, & \text{未激发}, \\ 1, & \text{激发}, \end{cases} \quad i = 1,2,\cdots,n$$

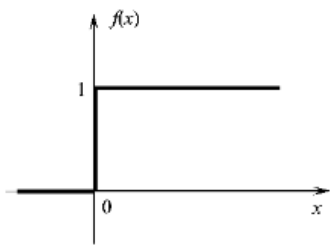


图 8.3　阶梯函数示意图

- 1943 年，美国心理学家麦克洛奇（W. Mcculloch）和数学家皮兹（W . Pitts）提出了用神经元网络对信息进行处理的数学模型——*神经元计算模型（即*MP *模型)*。

- f(x)是作用函数(Activation Function)——激发函数。MP 神经元模型中的作用函数为单位阶跃函数。当神经元 i 的输入信号加权和超过阈值时，输出为“1”，即“兴奋”状态；反之输出为“0”，是“抑制”状态。

- **激发函数的基本作用：**
  - 控制输入对输出的激活作用；
  - 对输入、输出进行函数转换；
  - 将可能无限域的输入变换成指定的有限范围内的输出。
- MP 神经元模型是人工神经元模型的基础，也是神经网络理论的基础。
- 在神经元模型中，作用函数除了单位阶跃函数之外，还有其它形式。不同的作用函数，可构成不同的神经元模型。

(2) 分段线性(图 8.4)

$$f(x)=\begin{cases}1, & x\geqslant 1\\ x, & -1<x<1\\ -1, & x\leqslant -1\end{cases}$$

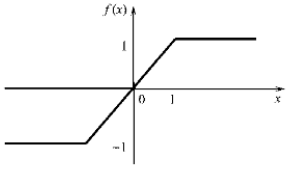


图 8.4 分段线性示意图

(3) S 型函数(图 8.5)

具有平滑和渐进性，并保持单调性，最常用的函数形式为

$$f(x)=\frac{1}{1+e^{-\alpha x}}$$

参数 α 可控制其斜率。另一种常用的是双曲正切函数：

$$f(x)=\tanh(x/2)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$

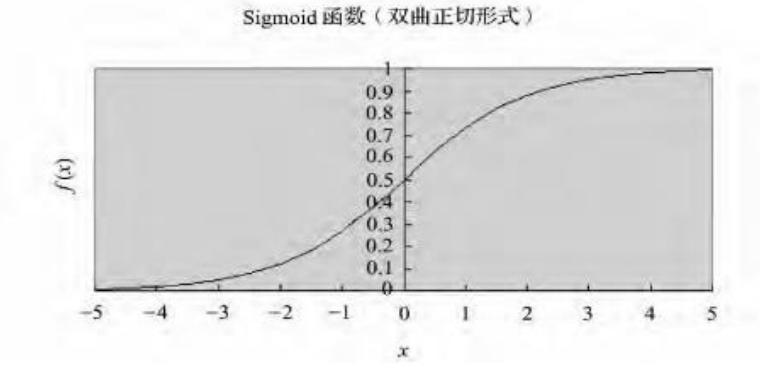


图 8.5 S 型函数

1.2 人工神经网络的分类

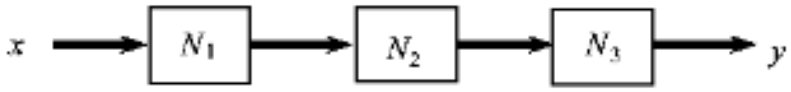
目前，神经网络模型的种类比较多，已有近 40 余种神经网络模型，其中典型的有 BP 网络、Hopfield 网络、CMAC 小脑模型、ART 自适应共振理论和 Boltzmann 机网络。

分类依据	分　类
网络的性能	连续型与离散型网络
	确定型与随机型网络
网络的结构	反馈网络
	前馈网络
	脆状网络
学习方式	有教师学习
	无教师学习

神经网络强大的计算功能——是通过神经元的互连而达到的。根据神经元的拓扑结构形式不同，神经网络可分成以下两大类：

1）前馈(多层)网络

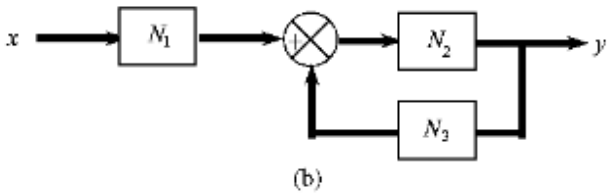
- 每个神经元从外部环境或别的神经元接收输入，但没有反馈。
- 对于一定的输入形式计算一个输出。一旦被训练(具有了固定的连接权值)，网络相应于给定输入形式的输出将是相同的，不管网络先前的激活性如何，*没有展示任何动力学特性*，*不存在稳定性问题*。
- 对于前馈网络，动态特性常被简化为单个瞬时非线性映射，前馈网络由 一个静态非线性映射表示。从作用效果来看，前馈网络主要是函数映射，可用于模式识别与函数逼近。



感知器(Perceptron)、BP 神经网络和径向基函数(RBF-Redial Basis Function)神经网络都属于前馈(多层)网络。

2）反馈(递归)网络

- 按对能量函数的所有极小点的利用情况，可将反馈网络分为两类：
  一类是能量函数的所有极小点都起作用——各种联想存储器
  另一类只利用全局极小点——求解优化问题。
- 反馈神经网络具有动态特性，有包含动态构造模块的处理单元(积分器或单延迟)，以反馈方式操作。
- 网络动态特性由非线性常微分方程组或差分方程组描述——反馈神经网络由非线性动力系统表示。
- 反馈式网络用于优化计算和联想记忆。



Hopfield 网络、Boltzmann 机网络属于这一类。

- 一般来讲，人工神经网络并不只是被它的结构所表征，还有所使用的神经元类型、学习过程和操作方式(原则)。根据操作方式，人工神经网络可以作为确定系统或随机系统来操作。
  - 在确定型人工神经网络中，所有参数和信号都是确定型的；
  - 在随机型人工神经网络中，信号和参数(连接权值)被一些随机量随机地改变(在时间上有相同的概率)。
- 至今所提出的真实生物神经网络的人工模型只是真实生物结构的粗略近似。
- 人工神经网络结构问题仍是一个困难的尚未解决的问题，是许多研究者正在努力探索的领域。这一问题的难点在于:
  - 是否有必要尽可能精确地模型化生物构造；
  - 是否能通过使用不完全对应于真实生物神经系统的模型就可以取得期望的性质和效果。

1.3 人工神经网络的工作过程

1.3.1 人工神经网络的工作过程主要分为两个阶段:

1）学习期

前提：各计算单元传递函数不变，

结果：其输出由两个因素决定:

- 输入数据；
- 与此输入单元连接的各输入量的权重。
- 因此，若处理单元要学会正确反映所给数据的模式，唯一用以改善处理单元性能的元素就是连接的权重，各连线上的权值通过学习来修改。

2）工作期

前提：连接权固定；

结果：计算神经元输出。

编制神经网络程序，主要是确定:

传递函数——决定阈值的方程;

训练计划——设置初始权重的规则及修改权重的方程;

网络结构——处理单元数，层数及相互连接状况。

1.3.2 人工神经网络的学习方式主要有三种:

1）有指导的学习(监督学习)

前提：要有输入数据及一定条件下的输出数据；

结果：网络根据输入输出数据来调节本身的权重。

学习过程的目的——减小网络应有输出与实际输出之间的误差。

学习过程终止条件：当误差达到了允许范围，权重不再改动，网络的输出符合于实际的输出。

特点：

- 不能保证得到全局最优解。
- 要求大量训练样本，收敛速度慢。
- 对样本的表示次序变化比较敏感。

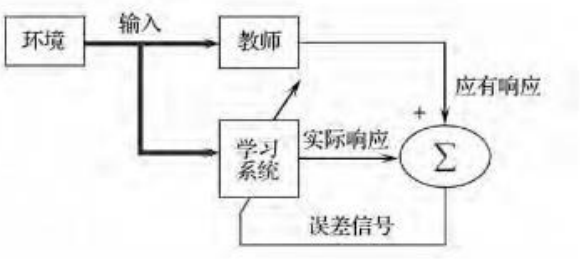


图 8.7　监督学习框图

有指导的学习的训练算法的主要步骤包括：

- 从样本集合中取一个样本（Ai，Bi）；
- 计算出网络的实际输出 O；
- 求 D=Bi-O；
- 根据 D 调整权矩阵 W；
- 对每个样本重复上述过程，直到对整个样本集来说，误差不超过规定范围。

2）无指导的学习(非监督学习)

- 前提：只提供输入数据，无相应输出数据。
- 结果：网络检查输入数据的规律或趋向，根据网络本身的功能来调整权重。
- 学习过程——系统提供动态输入信号，使各个单元以某种方式竞争，获胜者的神经元和其邻域得到增强，其他神经元进一步被抑制，从而将信号空间划分为多个有用的区域。

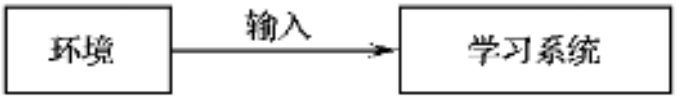


图 8.8　非监督学习系统

3）强化学习(再励学习)——介于上述两种学习之间。

- 前提：外部环境对系统输出结果只给出评价信息(奖或惩)，不提供正确答案。
- 结果：学习系统通过强化那些受奖的动作来改善自身的性能。

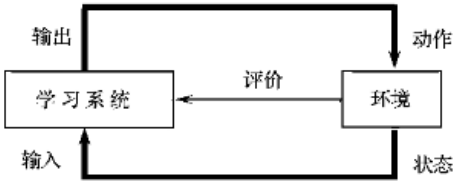


图 8.9　强化学习系统

1.3.3 人工神经网络的学习规则

- 在人工神经网络中，*学习规则是修正权值的一个算法，以获得合适的映射函数或其他系统性能*。
- **学习规则分为如下几类:**
- **相关规则：仅仅根据连接间的激活水平改变权值;**

  Donall Hebb 根据生理学中的条件反射机理，于 1949 年提出的神经元连接强度变化的规则:

- 如果两个神经元同时兴奋(即同时被激活)，则它们之间的突触连接加强：

$$\Delta w_{ij} = \alpha o_i o_j$$

- α为学习速率，o_i、o_j为神经元 i 和 j 的输出。

Hebb 学习规则是人工神经网络学习的基本规则，几乎所有神经网络的学习规则都可以看作 Hebb 学习规则的变形。

- **纠错规则：依赖关于输出节点的外部反馈改变权值;**

纠错规则的最终目的——基于误差(实际输出与期望输出之差)的目标函数达到最小，以使网络中每一输出单元的实际输出在某种统计意义上逼近期望输出。

- 在具体应用中，可以转化为最小均方差规则，采用最优梯度下降法。通过在局部最大改善的方向上一小步、一小步地进行修正，力图达到表示函数功能问题的全局解。
- **感知器学习使用纠错规则:**

①如果一节点的输出正确，一切不变;

②如输出本应为 0 而为 1，则相应地减小权值;

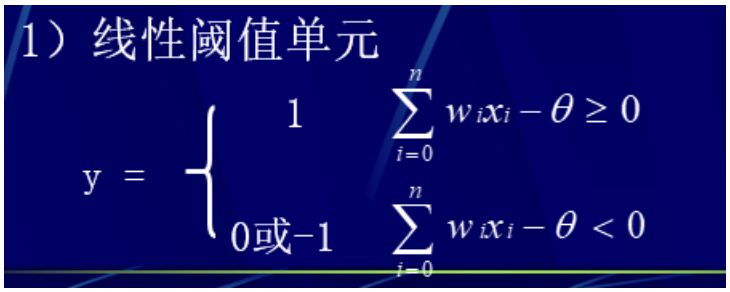
③如果应为 1 而输出为 0，则权值增加一增量。

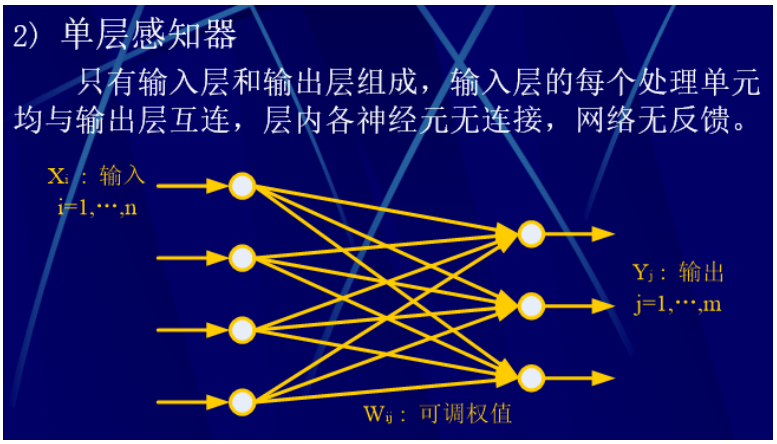
1957 年，美国心理学家罗森布拉特(F.Rosenblate)提出的感知机(Perceptron)模型是第一个真正的人工神经网络，具有自学习能力，第一次把神经网络的研究从纯理论的探讨付诸工程实践，*掀起了人工神经元网络的第一次研究热潮*。

感知器

罗森勃拉特(Rosenblatt)于 1957 年提出,把神经网络的研究从纯理论探讨引向了工程实践。

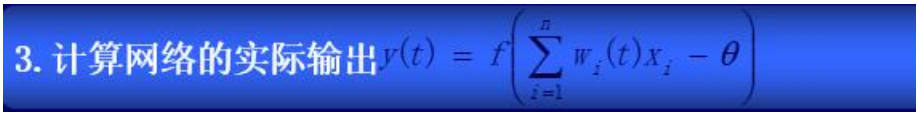
感知器是只有单层计算单元的前向神经网络,由线性阈值单元组成

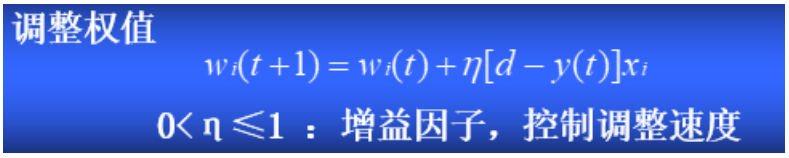




学习算法:假设只有一个输出 y(t)

1. 给 $w_i(0)(i=1,2\cdots\cdots n)$及θ赋一个较小的非 0 随机数作为初值

2. **输入一个样例 X = [x1,x2……xn]和期望的输出 d**
3. 

4. 

5. 转第二步，直到 w_i 对一切样例稳定为止

3) 多层感知器(MLP)

在单层感知器的输入和输出层之间增加一个或多个隐层。可产生复杂的决策界面和任意的布尔函数。

前向多层神经网络也即多层感知器(MLP)，也叫 BP 网。BP(Back-Propagation)算法，是用于前向多层神经网络的反传学习算法，是目前应用最广泛且重要的训练前向神经网络的学习算法。

纠错规则分类：

对于 δ 学习规则，可分为一般 δ 规则和广义 δ 规则。

(1) δ 学习规则　它优于感知器学习，因为这里 Δw 不是一固定量而是与误差成正比的，即

$$\Delta w_{ij} = \eta \delta_i v_j$$

这里 η 是全局控制系数，而 $\delta_i = t_i - v_i$，即期望值与实际值之差。

δ 学习规则和感知器学习规则一样只适用于线性可分函数，无法用于多层网络。

(2) 广义 δ 学习规则　它可在多层网络上有效地学习，其关键是对隐节点的偏差 δ 如何定义和计算。对 BP 算法　　，当 i 为隐节点时，定义

$$\delta_i = f'(\mathrm{net}_i) * \sum_k \delta_k w_{ki}$$

这里 w_{ki} 是节点 i 到上一层节点 k 的权值，$f'(\,)$ 为激励函数 $f(\,)$ 的一阶导数；将某一隐节点馈入上一层节点的误差的比例总和(加权和)作为该隐节点的误差，通过可观察的输出节点的误差，下一层隐节点的误差就能递归得到。广义 δ 规则可学习非线性可分函数。

1985 年鲁姆尔哈特（Rumelhart） 和麦克劳（McCelland） 提出的多层网络的误差传播学习算法——Back-Propagation 反向传播学习算法（BP 模型）。该模型引入多层隐结点，解决了非线性样本的问题。

- BP 算法目前已成为迄今为止应用最普遍的神经网络学习算法。

(3) Holtzman 机学习规则　它是基于模拟退火的统计方法来替代广义 δ 规则，适用于多层网络。它提供了学习隐节点的一个有效方法，能学习复杂的非线性可分函数。主要缺点是学习速度太慢，因为在模拟退火过程中要求当系统进入平衡时，“冷却”必须慢慢地进行，否则易陷入局部极小。它基本上是梯度下降法，所以要求提供大量的例子。

- **由上述可知，纠错规则基于梯度下降法，缺点：**

1.不能保证得到全局最优解;

2.同时要求大量训练样本，因而收敛速度慢;

3.纠错规则对样本的表示次序变化比较敏感，这就像教师必须认真备课，精心组织才能有效地学习。

- **无教师学习规则：学习表现为自适应于输入空间的检测规则。**
  ——调整参数以反映观察事件的分布
  1 不寻找一个特殊映射函数的表示。
  2 将事件空间分类成输入活动区域，有选择地对这些区域作出响应。

- **总结:**
- Hebb 学习规则的相关假设是许多规则的基础，尤其是相关规则;
- HNN 和自组织特征展示了有效的模式识别能力;
- 纠错规则使用梯度下降法，因而有局部极小问题;
- 无教师学习规则提供了新的选择，利用自适应学习方法，*使节点有选择地接收输入空间上的不同特性，抛弃了普通 ANN 学习映射函数的学习概念*，提供了基于检测特性空间的活动规律的性能描写。

1.4 人工神经网络的几何意义

线性样本——对 n 维空间中的一个两类样本,若能找到一个超平面将二者分开。否则为非线性样本。

- 神经元代表的超平面将空间分割成若干区，使每个区中只含同类样本的结点。
- 一个神经元将其他神经元对它的信息总输入作用(通过作用函数)以后的输出，相当于该神经元所代表的超平面将 n 维空间（n 个输入神经元构成的空间）超平面上部结点 P 转换成数据 1 类，超平面及其下部结点转换成数据 0 类。
  由此可以得出结论：神经元起一个分类作用。
- 实际系统大都是多输入多输出的非线性系统，很难用机理分析或系统辨识的方法获得足够精确的数学模型。
- 人工神经网络的输入和输出变量的数目是任意的，并且具有逼近任意非线性函数的能力：为多输入多输出的非线性系统提供了一种通用的建模方法。
- 人工神经网络系统的模型是非算式的，人工神经网络本身就是辨识模型，可调参数反映在网络内部的连接权上，不需要建立以实际系统数学模型为基础的辨识格式，可以省去系统结构辨识这一步骤。
- 神经网络建模法的任务：利用已有的输入输出数据来训练一个由神经网络构成的模型，能足够精确地近似给定的非线性系统。
- 目前，神经网络建模方法发展得很快，出现了很多模型和算法，应用也越来越广泛，是系统建模技术的一个重要发展方向。

三、BP 网络

1. BP 网络结构

- 从结构上来讲，BP 网络为分层型网络，不仅有输入层节点、输出层节点，而且有隐层节点(可以是一层或多层)。
- **节点的作用：**
  1）隐节点的作用 ——将原非线性样本变换成线性样本。
  2）输出结点的作用——将线性样本变换成两类（1 类或 0 类）。

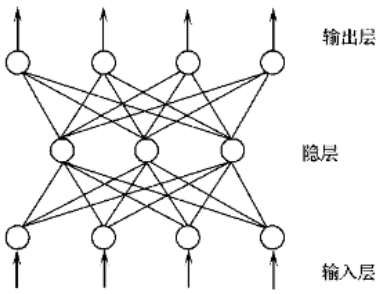
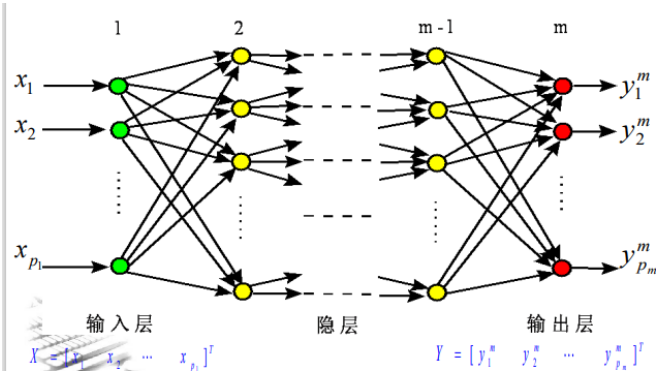


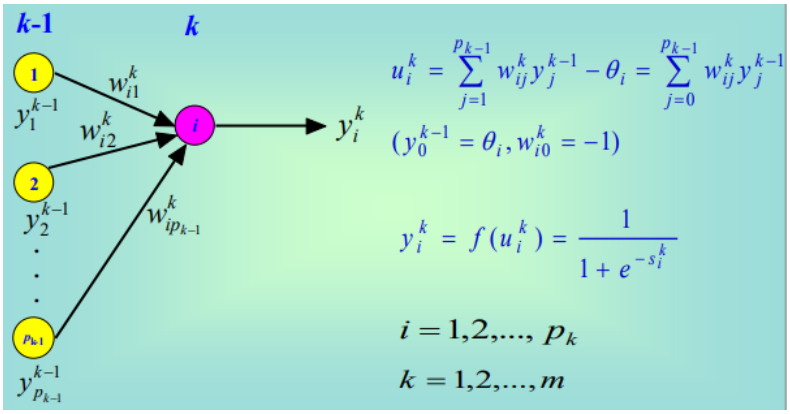
图 8.12　BP 反向传播模型的网络结构

- **BP 网络与感知机的主要区别**——每一层的权值都可以通过学习来调整。
- 对于一个 BP 网络，中间层可以有多个，具有一个中间层的 BP 网络为基本的 BP 网络模型。
- BP 网络是一种前馈型网络。

（1）BP 网络结构



（2）输入输出变换关系



（3）工作过程

- **第一阶段或网络训练阶段：**
  - *N* 组输入输出样本：$x_i=[x_{i1}, x_{i2},\ldots, x_{ip1}]^T$

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

    $i=1, 2,\ldots, N$
  - 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。

2. BP 学习算法

- BP 网络的学习算法——学习算法的推广和发展，是一种有教师的学习。这个算法的学习过程由正向传播和反向传播组成。
  - 正向传播：输入信息从输入层经隐单元层逐层处理，传向输出层，每一层神经元的状态只影响下一层神经元的状态。
  - 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，使得误差信号最小。
- BP 网络采用最小二乘学习算法和梯度搜索技术，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 网络学习过程——一种误差边向后传播边修正权系数的过程。
- **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：输入信息正向传播、误差信号反向传播。 、
  - 层与层的连接是单向的，信息的传播是双向的。
- **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。
  - 算法不一定收敛。
- 下图为多层前向网络的一部分，其中有两种信号在流动

  1）工作信号(用实线表示)：施加输入信号后向前传播直到 在输出端产生实际输出的信号，是输入和权的函数。

  2）误差信号(用虚线表示)：网络实际输出与应有输出间的 差值，由输出端开始逐层向后传播。

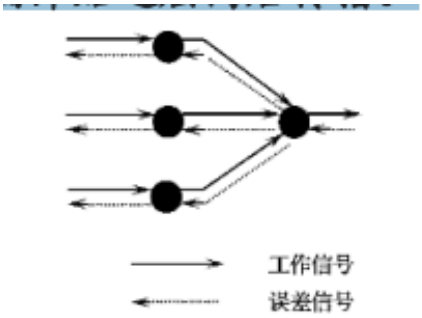


图 8.13 前向工作信号与反向误差信号

- 节点的特性要求是可微的，通常选 S 型

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

Sigmoid 函数的特点:

- 连续可微
- 单调
- 取值在[0,1]

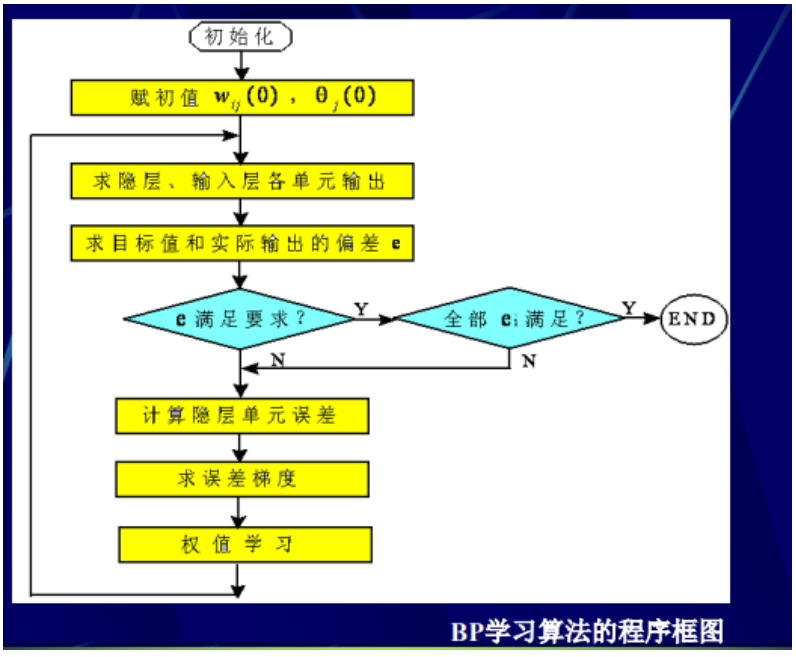
B-P 算法

- 学习的目的：对网络的连接权值进行调整， 使得对任一输入都能得到所期望的输出。
- 学习的方法：用一组训练样本对网络进行训练，其中每一个样例都包括输入及期望的输出两部分。
- 应用：对待识别的样本进行模式识别

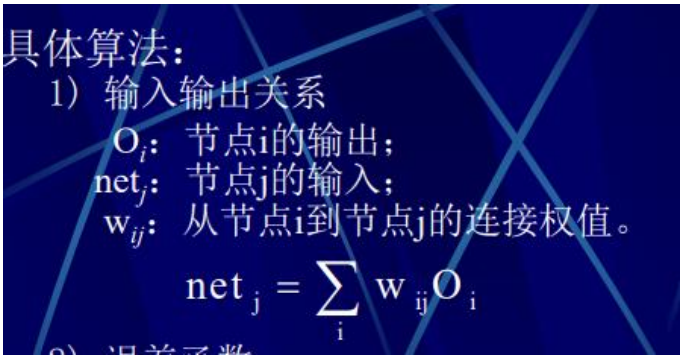
BP 算法的具体步骤：

- 从训练样例集中取一样例，把输入信息输入到网络中。
- 由网络分别计算各层节点的输出。
- 计算网络的实际输出与期望输出的误差。
- 从输出层反向计算到第一个隐层，按一定原则向减小误差方向调整网络的各个连接权值 - 反向传播。
- 对训练样例集中的每一个样例重复以上步骤，直到对整个训练样例集的误差达到要求时为止。

  在训练时，需要反向传播，而一旦训练结束，求解实际问题时，则只需正向传播



BP学习算法的程序框图



2） 误差函数

$$e = \frac{1}{2}\sum_{k}(\hat{y}_k - y_k)^2$$

$\hat{y}, y$ 分别表示输出层上节点k的期望输出与实际输出

3） 连接权值的修正

$$w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}$$

$w_{jk}(t+1)$和$w_{jk}(t)$分别表示t+1和t时刻上从节点j到节点k的连接权值， Δw_{jk}为修正量。

为了使连接权值沿着e的梯度变化方向得以改善，网络逐渐收敛，取

$$\Delta w_{jk} = -\eta \frac{\partial e}{\partial w_{jk}}$$

其中，η为增益因子

$$\frac{\partial e}{\partial w_{jk}} = \frac{\partial e}{\partial net_k} \bullet \frac{\partial net_k}{\partial w_{jk}}$$

$$\because net_k = \sum_j w_{jk} O_j \rightarrow \frac{\partial net_k}{\partial w_{jk}} = \frac{\partial}{\partial w_{jk}} \sum_j w_{jk} O_j = O_j$$

令 $\delta_k = \frac{\partial e}{\partial net_k}$

$$\therefore \Delta w_{jk} = -\eta \frac{\partial e}{\partial w_{jk}} = -\eta \cdot \delta_k \cdot O_j$$

分两种情况计算δ_k

节点k是输出层上的节点，此时$O_k = y_k$

$$\delta_k = \frac{\partial e}{\partial net_k} = \frac{\partial e}{\partial y_k} \cdot \frac{\partial y_k}{\partial net_k}$$

$$\because e = \frac{1}{2}\sum(\hat{y}_k - y_k)^2 \rightarrow \frac{\partial e}{\partial y_k} = -(\hat{y} - y)$$

$$又 \frac{\partial y_k}{\partial net_k} = f'(net_k)$$

$$\therefore \delta_k = -(\hat{y}_k - y_k) \cdot f'(net_k)$$

节点k不是输出层上的节点

$$\delta_k = \frac{\partial e}{\partial net_k} = \frac{\partial e}{\partial O_k} \cdot \frac{\partial O_k}{\partial net_k}$$

$$又 \frac{\partial e}{\partial O_k} = \sum_m \delta_m w_{km}$$

$$\frac{\partial O_k}{\partial net_k} = f'(net_k)$$

$$\therefore \delta_k = f'(net_k) \cdot \sum_m \delta_m w_{km}$$

故先计算 最高层（输出 层）上各节点 δ 值，再反传到较低层上， 计算各隐层节点的 δ 值

$$设 f(x) = \frac{1}{1+e^{-x}}$$

$$则 f'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{(1+e^{-x})-1}{(1+e^{-x})^2}$$

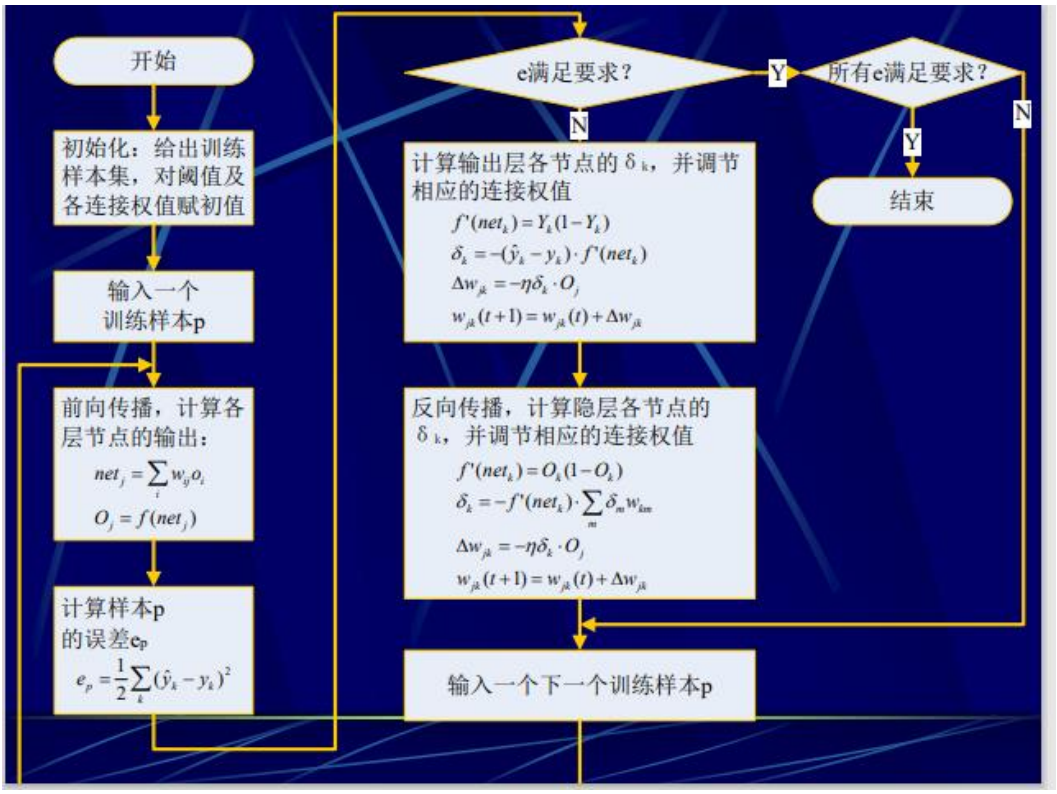
$$= \frac{1}{(1+e^{-x})} - \frac{1}{(1+e^{-x})^2} = \frac{1}{(1+e^{-x})}\left(1 - \frac{1}{(1+e^{-x})}\right)$$

$$= f(x)[1 - f(x)]$$

$$\therefore 对于输出层节点，f'(net_k) = f(net_k)[1 - f(net_k)]$$

$$= y_k(1 - y_k)$$

$$对于非输出层节点，f'(net_k) = O_k(1 - O_k)$$



3 BP 算法的不足及其改进

- BP 算法成功解决了感知机无能为力的非线性可分模式的分类问题，被广泛用于模式匹配、分类、识别和自动控制等应用领域。
- BP 网络在本质上是一种输入到输出的映射，能够学习大量的输入与输出之间的映射关系，不需要任何输入和输出间的精确的数学表达式，只要用已知的模式对 BP 网络加以训练，网络就具有输入输出对之间的映射能力。
- BP 算法的关键——在于中间层的学习规则，而中间层就 相当于对输入信息的一个特征抽出器。
- **BP 模型存在有如下问题:**
  1) 是一个非线性优化问题，存在局部极小问题;
  2) 学习算法的收敛速度很慢(通常要几千步迭代或更多)，通常只能用于离线的模式识别问题;
  3) 网络运行是单向传播，没有反馈，不是一个非线性动力系统，只是一个非线性映射;
  4) 中间层个数和中间层的神经元个数的选取尚无理论上的指导，而是根据经验选取的;
  5) 对新加入的样本要影响到已经学完的样本，刻画每个输入样本特征的数目也要求必须相同。
- **提高 BP 算法收敛性的措施：**

1） 加快迭代收敛的公式:

为加快权值的修正，在迭代公式中，增加修正项，即

$$w_{ij}(k+1) = w_{ij}(k) + \eta\delta_i x_j + \alpha(w_{ij}(k) - w_{ij}(k-1))$$

其中，α 称为松弛因子。

如果误差函数值下降，则 α 取大于 1（如 α 取 1.7）；如果误差函数值不变或上升，则取 $0<\alpha<1$（如 α 取 0.7）。

2） 作用函数的修改

新的作用函数为

$$f(x) = \frac{1}{1+e^{-\lambda(x-\theta)}}$$

式中，θ 是阈值，$\theta>0$ 时使 S 曲线沿水平右移。$\lambda<1$ 时，使 S 曲线变得平缓，如图 8.17。

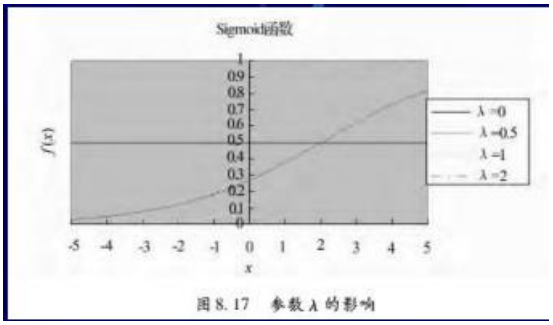


图 8.17 参数 λ 的影响

3） 学习系数的自适应调整

学习系数 η 由样本平均误差 E 的大小来调整，计算公式为

$$\eta^{(n+1)} = \eta^{(n)} \cdot \frac{E^{(n-1)}}{E^{(n)}}$$

其中，n 为迭代次数。当权值使 E 远离稳定点（偏大）时，学习系数 η 取较大值，而当接近稳定点时，η 取较小值。

学习系数影响系统学习过程的稳定性：

- 大的学习系数——可能使网络权值每一次的修正过大，甚至会导致权值在修正过程中超出某个误差的极小值呈不规则跳跃而不收敛;
- 过小的学习系数——导致学习时间过长。不过能保证收敛于某个极小值。

- 一般倾向选取较小的学习系数保证学习讨程的收敛性(稳定性)，通常在 0.01 一 0.8 之间

4 BP 神经网络的训练

4.1、产生数据样本集

包括原始数据的收集、数据分析、变量选择以及数据的预处理

✓ 首先要在大量的原始测量数据中确定出最主要的输入模式。

✓ 在确定了最重要的输入量后，需进行尺度变换和预处理。 尺度变换常常将它们变换到[-1,1]或[0,1]的范围。

在进行尺度变换前必须先检查是否存在异常点（或称野点），这些点必须删除。

通过对数据的预处理分析还可以检验其是否存在周期性、固定 变换趋势或其它关系。

- 对数据的预处理就是要使得经变换后的数据对于神经网络更 容易学习和训练

对于一个问题应该选择多少数据，这也是一个很关键的问题。

系统的输入输出关系就包含在数据样本中。一般来说，取的数据越多，学习和训练的结果便越能正确反映输入输出关系。

- 但选太多的数据将增加收集、分析数据以及网络训练付出的代价
- 选太少的数据则可能得不到正确的结果。

事实上数据的多少取决于许多因素，如 1）网络的大小 2）网络测试的需要以及 3）输入输出的分布等。

1） 网络的大小最关键。

通常较大的网络需要较多的训练数据。一个经验规则是：训练模式应是连接权总数的 5 至 10 倍

2） 网络测试的需要

在神经网络训练完成后，需要有另外的测试数据来对网络加以检验，测试数据应是独立的数据集合。

最简单的方法是：将收集到的可用数据随机地分成两部分，比如 其中三分之二用于网络的训练。另外三分之一用于将来的测试。 随机选取的目的是为了尽量减小这两部分数据的相关性。

3） 输入输出的分布

影响数据大小的另一个因素是输入模式和输出结果的分布，对数据预先加以分类可以减小所需的数据量。相反，数据稀薄不匀甚至覆盖则势必要增加数据量。

4.2 确定网络的类型和结构

神经网络的类型很多，需根据问题的性质和任务的要求来合适地选择网络类型。

- 一般从已有网络类型中选用一种比较简单而又能满足要求的网络，新设计一个网络类型来满足问题的要求往往比较困难。
  1） 若主要用于模式分类，尤其是线性可分的情况，则可采用较为简单的感知器网络
  2） 若主要用于函数估计，则可应用 BP 网络。
- 在网络的类型确定后，就要选择网络的结构和参数。 以 BP 网络为例，需选择网络的层数、每层的节点数、初始权值、 阈值、学习算法、数值修改频度、结点变换函数及参数、学习率等参数

有些项的选择有一些指导原则，但更多的是靠经验和试凑。

1） 网络层数的选取：

理论上早已证明：具有偏差和至少一个 S 型隐含层加上一个线性输出层的网络，能够逼近任何有理函数。

- 增加层数主要可以更进一步降低误差，提高精度，但同时也使网络复杂化，从而增加了网络权值的训练时间。

而误差精度的提高实际上也可以通过增加隐含层中的神经元数目来获得，其训练效果也比增加层数更容易观察和调整，所以，一般情况下，应优先考虑增加隐含层中的神经元数

2） 每层节点数的选取：

对于具体问题若确定了输入和输出变量后，网络输入层和输出层 的节点个数也便随之确定了。

- 隐层节点数对网络的泛化能力有很大的影响。
  - 节点数太多，倾向于记住所有的训练数据，包括噪声的影响，反而降低了泛化能力；
  - 节点数太少，不能拟和样本数据，没有较好的泛化能力。
  - 原则：选择尽量少的节点数以实现尽量好的泛化能力。

具体选择可采用如下方法：先设较少的节点，对网络进行训练， 并测试网络的逼近误差，然后逐渐增加节点数，直到测试的误差 不再有明显的减少为止。

3） 初始权值的选取：

由于系统是非线性的，初始值对于学习是否达到局部最小、是否能够收敛以及训练时间的长短关系很大。

如果初始值太大，使得加权后的输入落到激活函数的饱和区（两端接近 1 或－1 的部分），从而导致其导数非常小，而在计算权值的修正公式中，修正量正比与其导数，从而使调节过程几乎停顿下来。

一般总是希望经过初始加权后的每个神经元的输出值都接近于零，这样可以保证每个神经元的权值都能够在他们的 S 型激活函数变化最大之处进行调节，所以，一般取初始权值在(-1,1)之间的随机数。

4） 学习速率的选取：

学习速率决定每一次循环训练中所产生的权值变化量。

- 大的学习速率可能导致系统的不稳定
- 小的学习速率会导致训练较长，收敛速度很慢。不过能保证网络的误差值不跳出表面的低谷而最终趋于最小误差值。

- 小的学习速率会导致训练较长，收敛速度很慢。不过能保证网络的误差值不跳出表面的低谷而最终趋于最小误差值。
  - 作法：和初始权值的选取过程一样，在一个神经网络的设计中，网络要经过几个不同的学习速率的训练。通过观察每一次训练后的误差平方和的下降速率来判断选定的学习速率是否合适。 如果下降很快，说明学习速率合适。若出现振荡，则说明学习速率过大。

    对于较复杂的网络，为了减小寻找学习速率的训练次数以及训练时间，比较合适的方法是采用自适应学习速率。

4.3 训练和测试

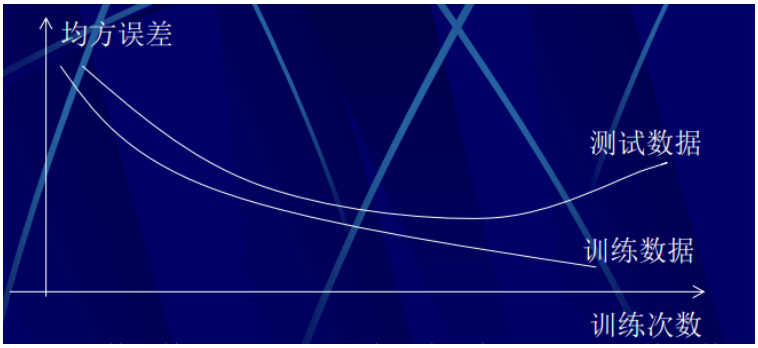
一次训练(或一次学习)——对所有样本数据正向运行一次并反向修改连接权一次。通常训练一个网络需要成百上千次。并非训练的次数越多，越能得到正确的输入输出的映射关系。

- 训练网络的目的——找出蕴含在样本数据中的输入和输出之间的本质联系，从而对于未经训练的输入也能给出合适的输出，即局部泛化能力。
- 网络的性能——主要是用它的泛化能力来衡量，不是用对训练 数据的拟和程度来衡量，

而是用一组独立的数据来加以测试和检验。

由于所收集的数据都是包含噪声的，训练的次数过多，网络将包含噪声的数据都记录了下来，在极端情况下，训练后的网络可以实现相当于查表的功能。但是对于新的输入数据却不能给出合适的输出，即并不具备很好的泛化能力。

实际操作时应该训练和测试交替进行，即每训练一次，同时用测 试数据测试一遍，画出均方误差随训练次数的变换曲线

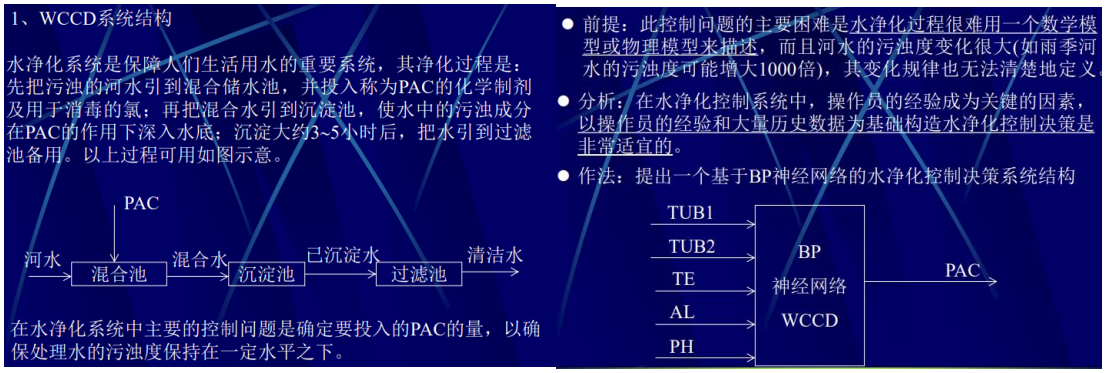


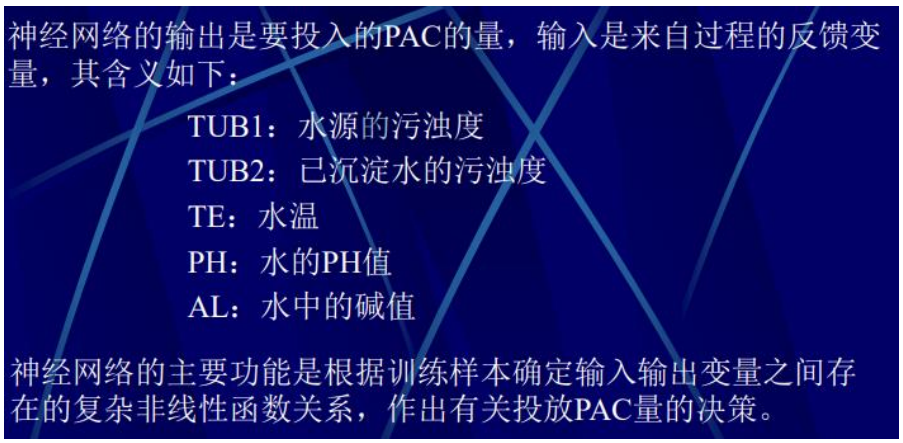
在用测试数据检验时，均方误差开始逐渐减小，当训练次数再增加时，测试检验误差反而增加，误差曲线上极小点所对应的即为恰当的训练次数，若再训练即为“过渡训练”了

5 BP 网络应用举例

BP 网络的本质功能——通过对简单非线性函数(S 型函数)的数次复合来实现输入到输出的高度非线性映射，隐含表达现实物理世界存在的及其复杂的非线性函数关系。

工程中存在的许多实际问题，如模式识别、特征压缩、图形处理、 预测预报、控制、决策、函数拟和等都可归结为求解输入到输出的高度非线性映射，因而可用 BP 网络有效地求解。下面以水净化控制决策系统(WCCD)为例说明 BP 网络的应用。





2、网络结构及数据规格化

该网络为三层BP网络，具有5个输入点(对应5个输入变量)，1个输出节点(对应系统的1个输出变量)。

从现场收集到的操作员进行PAC投入量决策的部分历史数据如表。这些数据可用作网络的训练数据。

由于选择的输出函数为S型函数，要求其输入范围在[0，1]之间的实数，而表的数据范围大大超出[0，1]区间，所以必须把它们规格化为[0，1]之间的实数，再送到数据网络处理。

$$Y = \frac{X - MINNUM}{MAXNUM - MINNUM}$$

X，Y分别是变换前后的数据，因为该表每一列的数据范围不同，所以应分别进行变换。

TUB1	TUB2	TE	AL	PH	PAC
10.0	1.0	18.8	53.0	7.1	1300
22.0	2.0	19.4	46.0	7.3	1400
50.0	1.0	19.5	40.0	7.1	1400
9.0	4.0	23.3	48.0	7.3	900
14.0	4.0	23.6	53.0	7.2	900
20.0	1.0	16.6	40.0	7.0	1100
12.0	3.0	18.8	55.0	7.2	900
8.0	1.5	18.0	50.0	7.2	1000
35.0	1.5	17.7	42.0	7.0	1200
16.0	3.0	19.3	42.0	7.1	1100

3、仿真结果

在仿真试验中，对该网络进行训练，达到了比较理想的学习精度(训练误差小于0.095)。

经对比，训练后的系统作出的关于PAC投放量的决策输出与操作员的决策几乎完全相同。

换言之，该决策系统准确地模拟了经验丰富的人类操作员的控制决策功能，且其性能明显高于基于统计模型的系统，很好地解决了水净化过程很难用数学模型或物理模型来精确描述，使用常规控制系统性能不佳的难题。

以上介绍的多层前向反馈网是无反馈的网络，且是静态网络。所以它不存在稳定性问题，通常只能用于解决与时间无关的模式识别、函数拟合等问题。

对与时间有关的问题如飞行器控制、语音处理等则需要引入动态系统。其中，一种是单元信息的处理引入时间参量；一种是网络结构中引入反馈。从而形成动态网络。

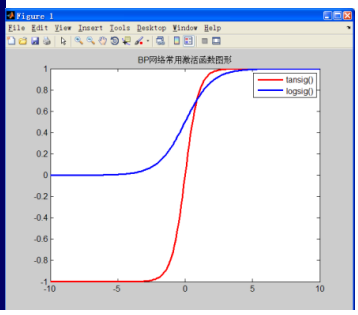
	前馈网（动态网）	反馈网（静态网）
信号方向	只有向前	既向前，也向后
训练阶段	动态，反复迭代搜索确定权矩阵	静态，一次性计算权矩阵
应用阶段	静态，一次性输出结果	动态，反复迭代后稳定在某个输出结果上

BP神经网络学习算法的MATLAB实现

- MATLAB中BP神经网络的重要函数和基本功能

函 数 名	功　能
newff()	生成一个前馈BP网络
tansig()	双曲正切S型(Tan-Sigmoid)传输函数
logsig()	对数S型(Log-Sigmoid)传输函数
traingd()	梯度下降BP训练函数

- MATLAB中BP神经网络的重要函数和基本功能
  - newff()
    - 功能 建立一个前向BP网络
    - 格式 net = newff(PR，[S1 S2...SN1]，{TF1 TF2...TFN1}，BTF，BLF，PF)
    - 说明 net为创建的新BP神经网络；PR为网络输入取向量取值范围的矩阵；[S1 S2...SNl]表示网络隐含层和输出层神经元的个数；{TFl TF2...TFN1}表示网络隐含层和输出层的传输函数，默认为‘tansig’；BTF表示网络的训练函数，默认为‘trainlm’；BLF表示网络的权值学习函数，默认为‘learngdm’；PF表示性能数，默认为‘mse’。

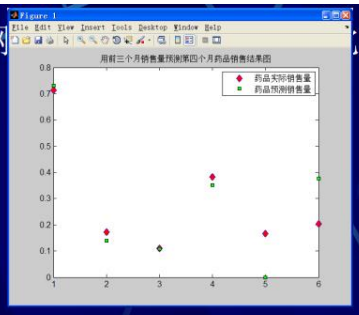


例2，下表为某药品的销售情况，现构建一个如下的三层BP神经网络对药品的销售进行预测：输入层有三个结点，隐含层结点数为5，隐含层的激活函数为tansig（双曲正切 S 型）；输出层结点数为1个，输出层的激活函数为logsig（对数 S 型），并利用此网络对药品的销售量进行预测，预测方法采用滚动预测方式，即用前三个月的销售量来预测第四个月的销售量，如用1、2、3月的销售量为输入预测第4个月的销售量，用2、3、4月的销售量为输入预测第5个月的销售量.如此反复直至满足预测精度要求为止。

月份	1	2	3	4	5	6
销量	2056	2395	2600	2298	1634	1600
月份	7	8	9	10	11	12
销量	1873	1478	1900	1500	2046	1556

```
%以每三个月的销售量经归一化处理后作为输入
P=[0.5152  0.8173  1.0000 ;
0.8173  1.0000  0.7308;
1.0000  0.7308  0.1390;
0.7308  0.1390  0.1087;
0.1390  0.1087  0.3520;
0.1087  0.3520  0.0000;]';
%以第四个月的销售量归一化处理后作为目标向量
T=[0.7308 0.1390 0.1087 0.3520 0.0000 0.3761];
%创建一个BP神经网络，每一个输入向量的取值范围为[0 ,1]，隐含层有5个神经%元，输出层有一个神经元，隐含层的激活函数为tansig，输出层的激活函数为%logsig，训练函数为梯度下降函数，即前面所描述的标准学习算法
net=newff([0 1;0 1;0 1],[5,1],{'tansig','logsig'},'traingd');
net.trainParam.epochs=15000;
net.trainParam.goal=0.01;
%设置学习速率为0.1
LP.lr=0.1;
net=train(net,P,T);
```

- BP网[illegible]图



- 由对比图可以看出预测效果与实际存在一定误差，此误差可以通过增加运行步数和提高预设误差精度业进一步缩小

三、补充：

也可以基于E来完成在权值空间的梯度搜索。

$$E = \frac{1}{2P}\sum_{p} e_p = \frac{1}{2P}\sum_{p}\sum_{k}(\hat{y}_{pk} - y_{pk})^2$$

其中P为训练样本的个数

此时，按反向传播计算样本p在各层的连接权值变化量$\Delta_p w_{jk}$，但并不对各层神经元的连接权值进行修改，而是不断重复这一过程，直至完成对训练样本集中所有样本的计算，并产生这一轮训练的各层连接权值的改变量Δw_{jk}

$$\Delta w_{jk} = \sum_{p} \Delta_p w_{jk}$$

此时才对网络中各神经元的连接权值进行调整，若正向传播后重新计算的E仍不满足要求，则开始下一轮权值修正。

感知器模型中神经元的变换函数采用的是符号函数，因此输出的是二值量。它主要用于模式分类。

多层前馈网具有相同的结构，只是神经元的变换函数采用S型函数，因此输出量是0到1之间的连续量，它可实现从输入到输出的任意的非线性映射。

由于连接权的调整采用的是反向传播(Back Propagation)的学习算法，因此该网络也称为BP网络。

在多层前馈网络中，第0层为输入层，第Q层为输出层，有n_Q个输出，中间层为隐层。设第q层(q=0,2,…，Q)的神经元个数为n_q，输入到第q层的第i个神经元的连接权系统为：

$$w_{ij}^{(q)}\ (i = 1,2,\cdots n_q\ ;\ j = 1,2,\cdots n_{q-1})$$

网络的输入输出变化关系为：

$$s_i^{(q)} = \sum_{j=0}^{n_{q-1}} w_{ij}^{(q)} x_j^{(q-1)} \qquad (x_0^{(q-1)} = \theta_i^{(q)}, w_{i0}^{(q)} = -1)$$

$$x_i^{(q)} = f(s_i^{(q)}) = \frac{1}{1+\exp(-\mu s_i^{(q)})}$$

$$i = 1,2,\cdots,n_q, j = 1,2,\cdots n_{q-1}, q = 1,2,\cdots Q$$

1、 标准 BP 算法

设给定P组输入输出样本：

$$\mathbf{x}_p^{(0)} = \begin{bmatrix} x_{p1}^{(0)} & x_{p2}^{(0)} & \cdots & x_{p,n_0}^{(0)} \end{bmatrix}^T$$

$$\mathbf{d}_p = \begin{bmatrix} d_{p1} & d_{p1} & \cdots & d_{p,n_Q} \end{bmatrix}^T$$

$$(p = 1,2,\cdots P)$$

利用该样本集首先对BP网络进行训练，也即对网络的连接权系数进行学习和调整，以使该网络实现给定的输入输出映射关系。

经过训练的BP网络，对于不是样本集中的输入也能给出合适的输出。该性质称为泛化(generalization)功能。

从函数拟和的角度，它说明BP网络具有插值功能。

设取拟和误差的代价函数为：

$$E = \frac{1}{2}\sum_{p=1}^{P}\sum_{i=1}^{n_Q}(d_{pi} - x_{pi}^{(Q)})^2 = \sum_{p=1}^{P} E_p$$

$$E_p = \frac{1}{2}\sum_{i=1}^{n_Q}(d_{pi} - x_{pi}^{(Q)})^2$$

如何调整连接权系数以使代价函数E最小。

优化计算的方法很多，比较典型的是采用一阶梯度法，即最速下降法。

一阶梯度法寻优的关键是计算优化目标函数(即本问题中的误差代价函数)E对寻优参数的一阶导数。

$$\frac{\partial E}{\partial w_{ij}^{(q)}},(q = Q, Q-1, \cdots 1) \qquad \frac{\partial E}{\partial w_{ij}^{(q)}} = \sum_{p=1}^{P}\frac{\partial E_p}{\partial w_{ij}^{(q)}}$$

对于第Q层有：

$$\frac{\partial E_p}{\partial w_{ij}^{(Q)}} = \frac{\partial E_p}{\partial x_{pi}^{(Q)}}\frac{\partial x_{pi}^{(Q)}}{\partial s_{pi}^{(Q)}}\frac{\partial s_{pi}^{(Q)}}{\partial w_{ij}^{(Q)}} = -(d_{pi} - x_{pi}^{(Q)}) f'(s_{pi}^{(Q)}) x_{pj}^{(Q-1)}$$

$$= -\delta_{pi}^{(Q)} x_{pj}^{(Q-1)} \qquad \delta_{pi}^{(Q)} = -\frac{\partial E_p}{\partial s_{pi}^{(Q)}} = (d_{pi} - x_{pi}^{(Q)}) f'(s_{pi}^{(Q)})$$

$x_{pi}^{(Q)}$ $s_{pi}^{(Q)}$ $x_{pj}^{(Q-1)}$ 表示用第p组输入样本所算得的结果。

对于第Q－1层有：

$$\frac{\partial E_p}{\partial w_{ij}^{(Q-1)}} = \frac{\partial E_p}{\partial x_{pi}^{(Q-1)}} \frac{\partial x_{pi}^{(Q-1)}}{\partial w_{ij}^{(Q-1)}}$$

$$= (\sum_{k=1}^{n_Q} \frac{\partial E_p}{\partial s_{pk}^{(Q)}} \frac{\partial s_{pk}^{(Q)}}{\partial x_{pi}^{(Q-1)}}) \frac{\partial x_{pi}^{(Q-1)}}{\partial s_{pi}^{(Q-1)}} \frac{\partial s_{pi}^{(Q-1)}}{\partial w_{ij}^{(Q-1)}}$$

$$= (\sum_{k=1}^{n_Q} -\delta_{pk}^{(Q)} w_{ki}^{(Q)}) f'(s_{pi}^{(Q-1)}) x_{pj}^{(Q-2)} = -\delta_{pi}^{(Q-1)} x_{pj}^{(Q-2)}$$

$$\delta_{pi}^{(Q-1)} = -\frac{\partial E_p}{\partial s_{pi}^{(Q-1)}} = (\sum_{k=1}^{n_Q} \delta_{pk}^{(Q)} w_{ki}^{(Q)}) f'(s_{pi}^{(Q-1)})$$

显然，它是反向递推计算的公式

即首先计算出 $\delta_{pk}^{(Q)}$ 然后再由上式递推计算出 $\delta_{pi}^{(Q-1)}$

依次类推，可继续反向递推计算出

$\delta_{pi}^{(q)}$ 和 $\frac{\partial E_p}{\partial w_{ij}^{(q)}}$ (q=Q－2，…，1)

$\delta_{pi}^{(q)}$ 的表达式中包含了导数项 $f'(s_{pi}^{(q)})$

假定$f(\cdot)$为S形函数 $x_{pi}^{(q)} = f(s_{pi}^{(q)}) = \frac{1}{1+\exp(-\mu s_{pi}^{(q)})}$

$$f'(s_{pi}^{(q)}) = \frac{\mu \exp(-\mu s_{pi}^{(q)})}{(1+\exp(-\mu s_{pi}^{(q)}))^2} = \mu f(s_{pi}^{(q)})[1 - f(s_{pi}^{(q)})]$$

$$= \mu x_{pi}^{(q)}(1 - x_{pi}^{(q)})$$

最后可归纳出BP网络的学习算法如下：

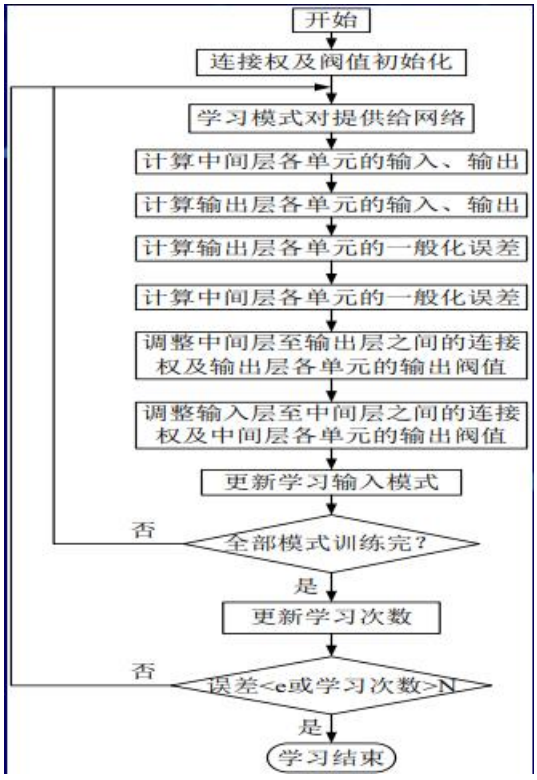
$$w_{ij}^{(q)}(k+1) = w_{ij}^{(q)}(k) + \alpha D_{ij}^{(q)}(k), \alpha > 0$$

$$D_{ij}^{(q)} = \sum_{p=1}^{P} \delta_{pi}^{(q)} x_{pj}^{(q-1)}$$

$$\delta_{pi}^{(q)} = (\sum_{k=1}^{n_{q+1}} \delta_{pk}^{(q+1)} w_{ki}^{(q+1)}) \mu x_{pi}^{(q)} (1 - x_{pi}^{(q)})$$

$$\delta_{pi}^{(Q)} = (d_{pi} - x_{pi}^{(Q)}) \mu x_{pi}^{(Q)} (1 - x_{pi}^{(Q)})$$

$$q = Q, Q-1, \cdots, 1, i = 1,2, \cdots, n_q, j = 1,2, \cdots n_{q-1}$$



由于该算法式反向递推(Back Propagation)计算出的，因而通常称该多层前馈网络为 BP 网络。

该网络实质上是对任意非线性映射关系的一种逼近，由于采用的是全局逼近的方法，因而 BP 网络具有良好的泛化能力。

真正的梯度下降是沿着梯度确定的方向以无穷小步长进行的。很明显，这是不切实际的，因此定义学习速率α，确定了沿梯度 方向的一个有限步长。

这里α是常数，它相当于确定步长的增益。

其中心思想就是选择足够大的α，使得网络迅速收敛，而不会因 调整过渡而振荡