模式识别的理论与方法

Pattern Recognition

裴继红

Chapter 6 (Part 2):

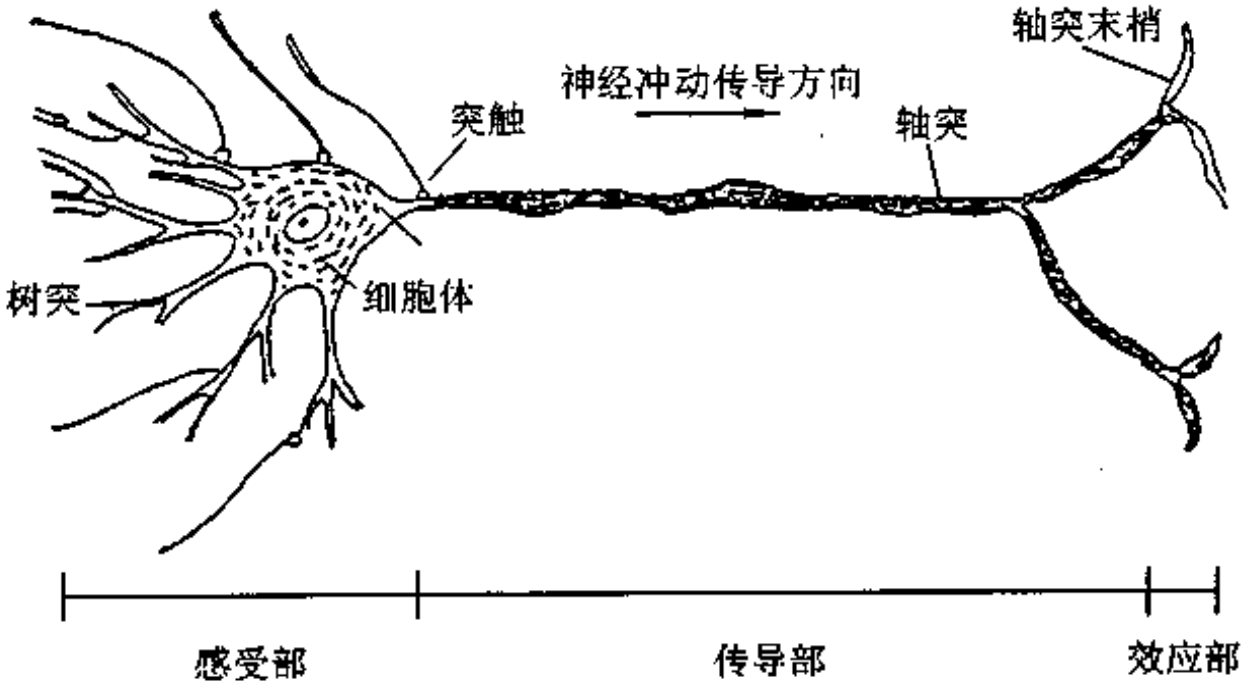
BP Neural Network

本讲内容

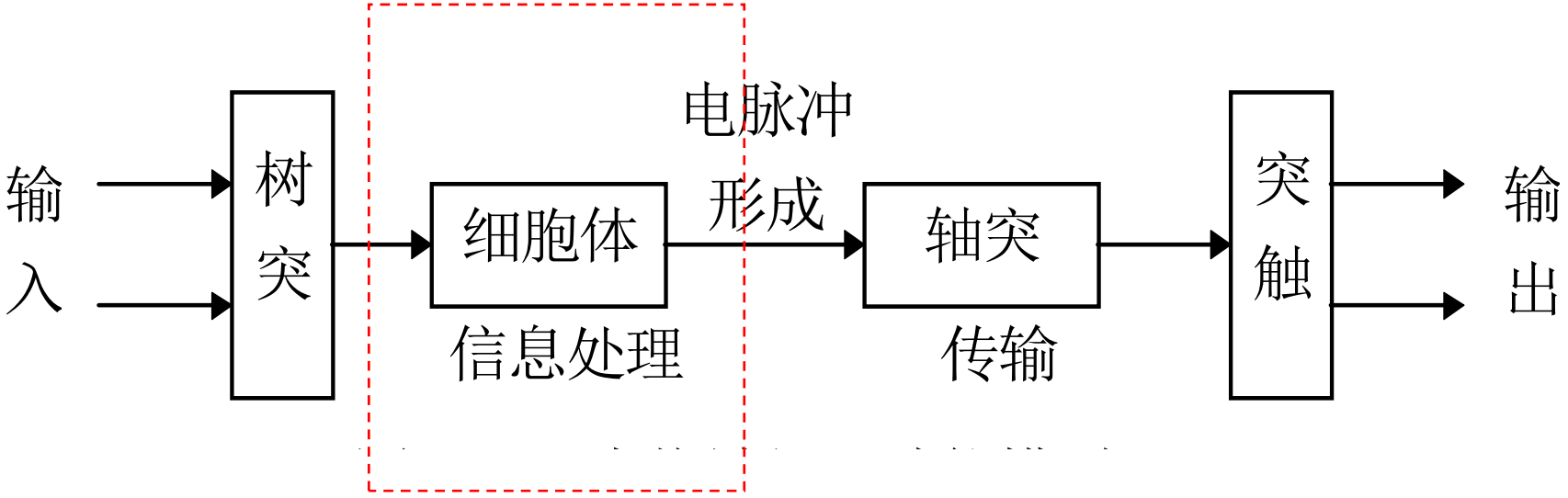
- 人工神经元的信号模型
- 单层神经网络——感知器
- 多层神经网络——BP神经网络
- BP网络设计中的一些说明
- 其他多层网络及学习算法

人工神经元的信号模型

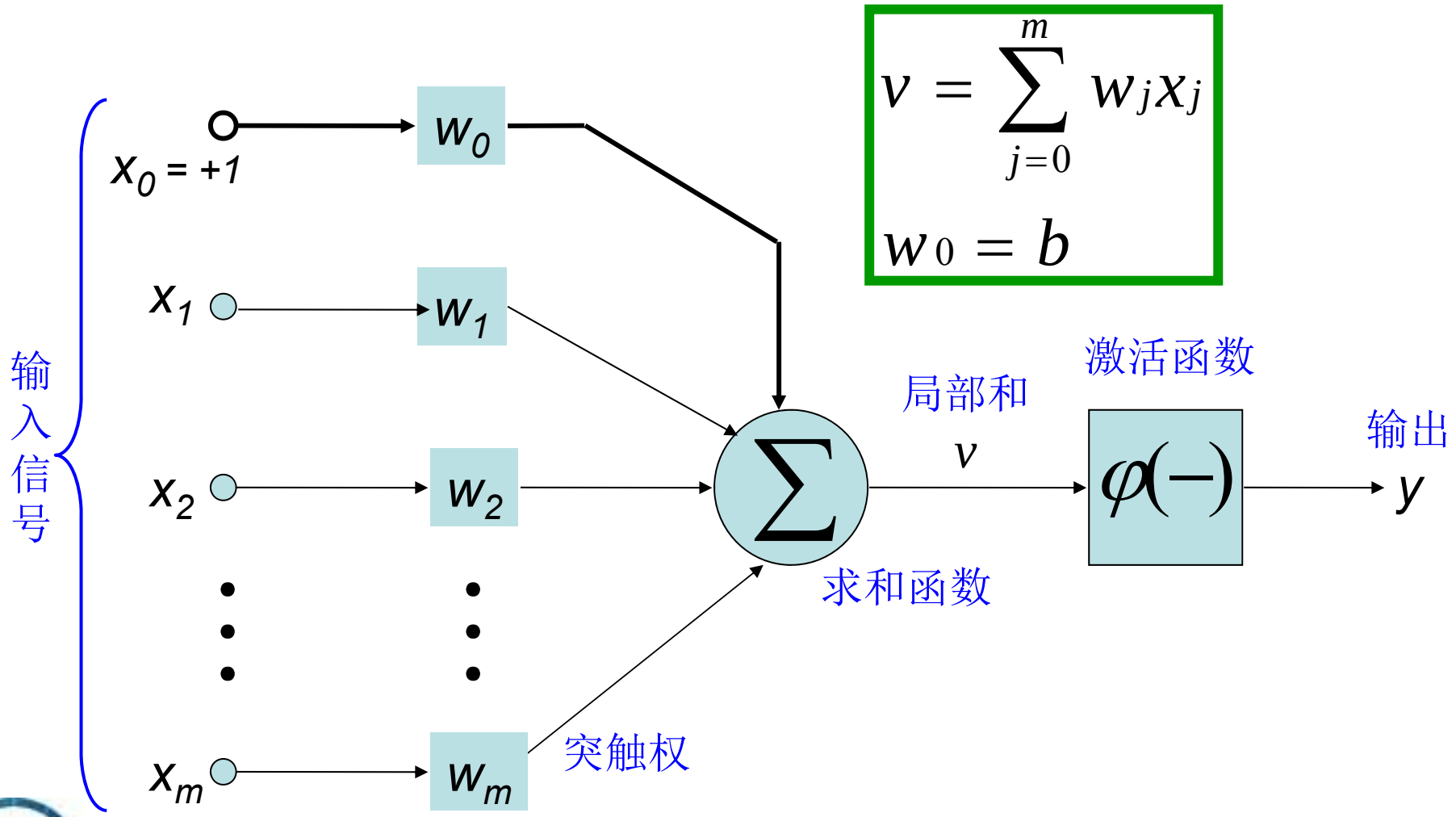
生物神经元



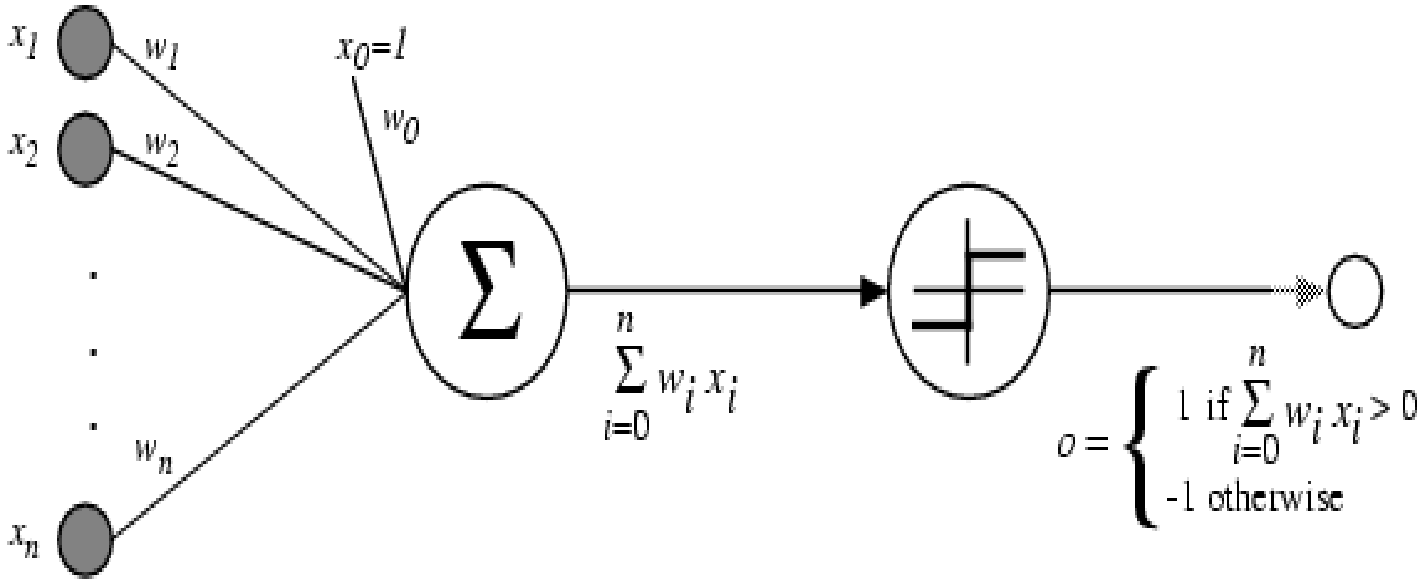
神经元的系统模型



神经元的信号模型

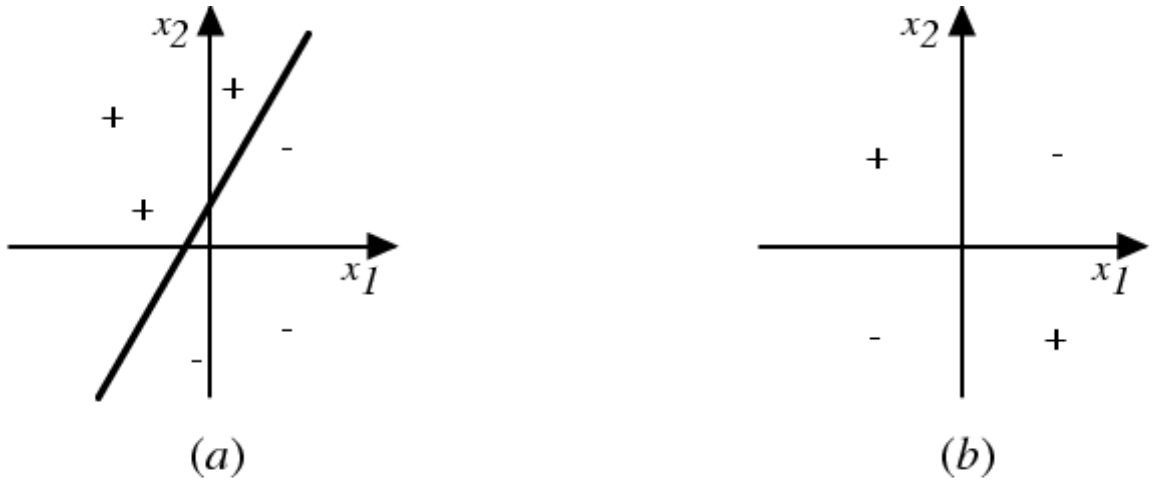

$v = \sum_{j=0}^{m} w_j x_j$
$w_0 = b$
$x_0 = +1$
w_0
x_1
w_1
x_2
w_2
x_m
w_m
输入信号
局部和
v
激活函数
$\varphi(-)$
输出
y
求和函数
突触权

单层人工神经网络：感知器模型



$$o(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + \cdots + w_n x_n > 0 \\ -1 & \text{otherwise.} \end{cases}$$

感知器的表达能力



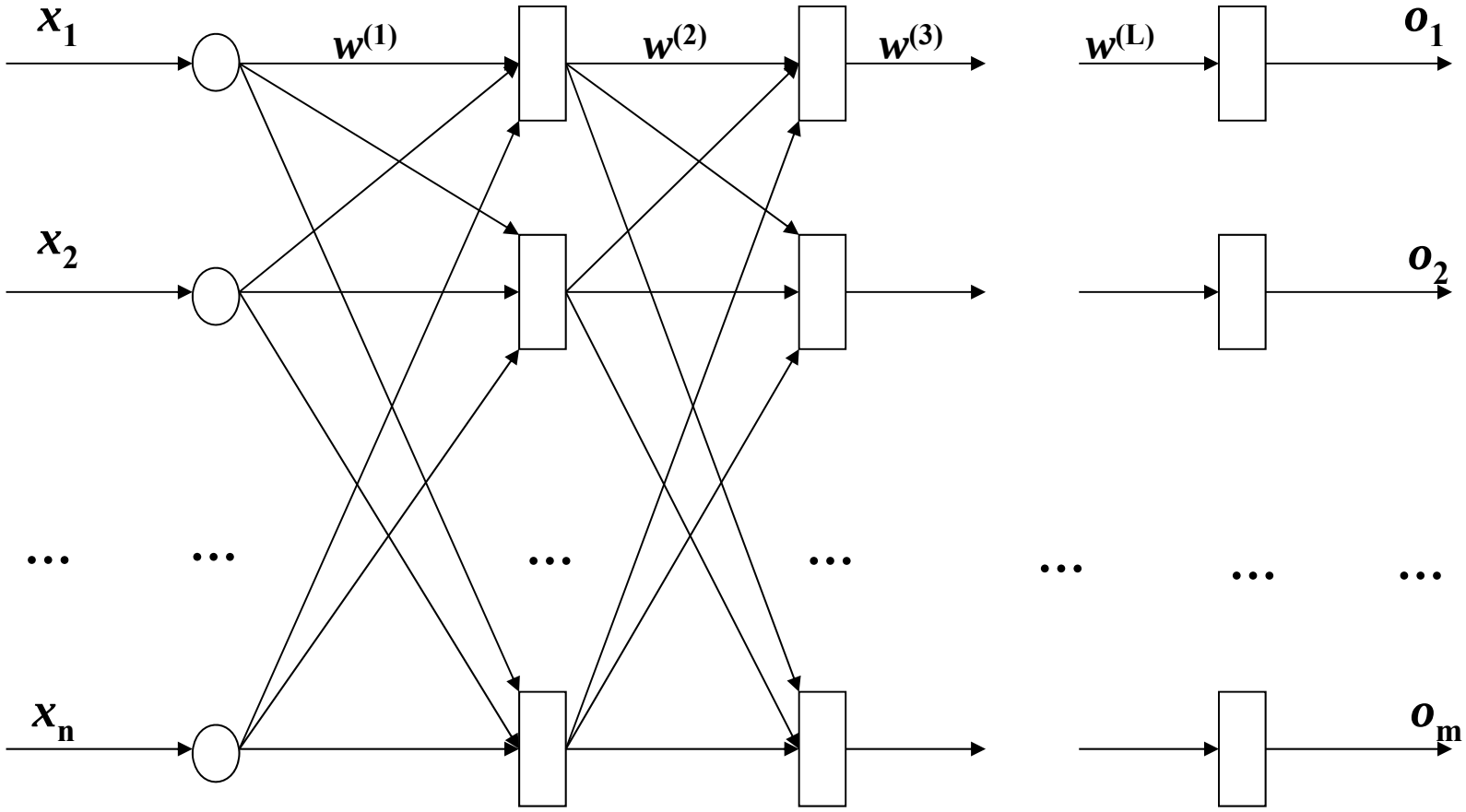
- 感知器可以表示所有的原子布尔函数，与，或，非，与非，或非。
- 感知器可以直接处理线性可分问题
- 无法直接处理线性不可分的问题

多层人工神经网络：BP网络

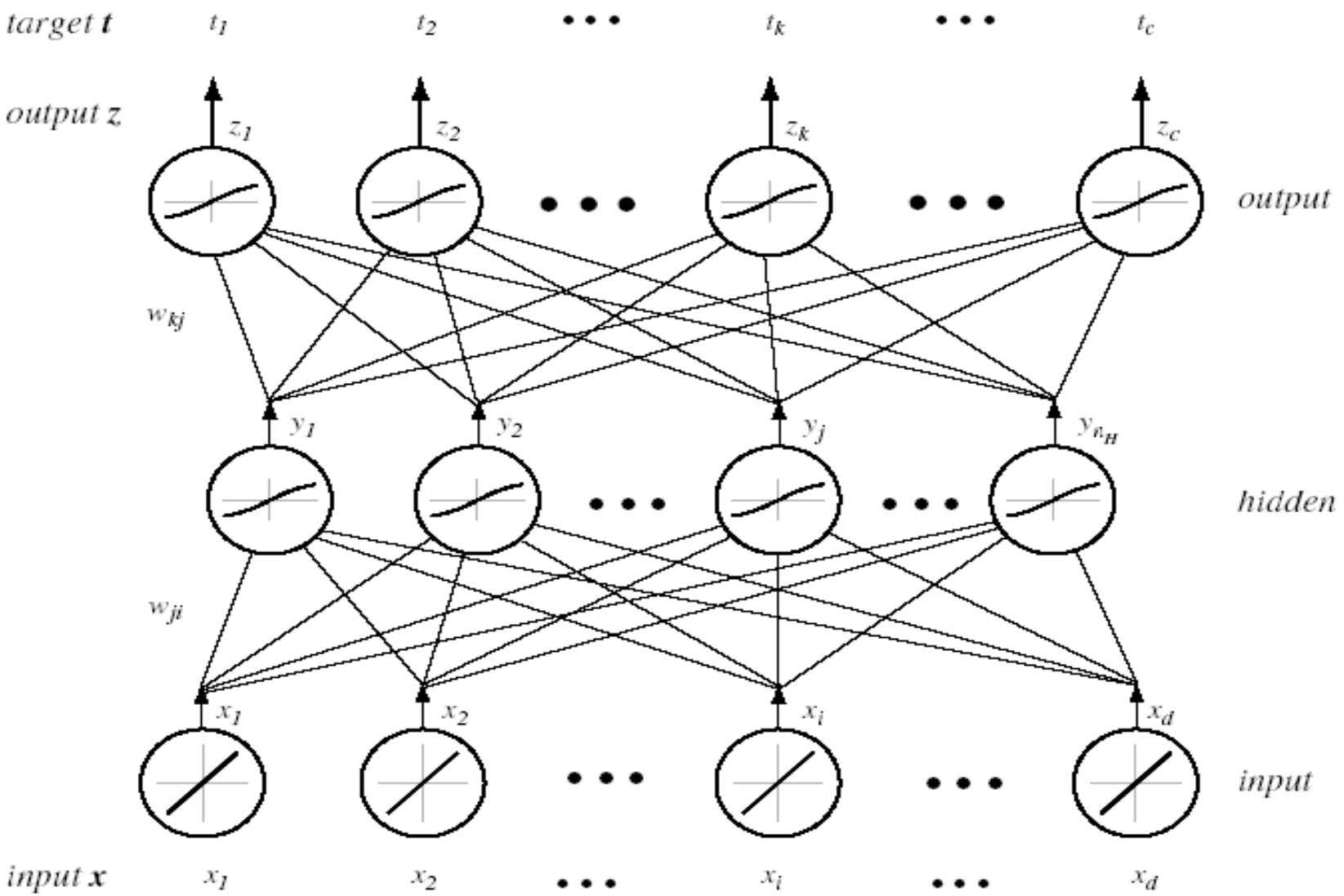
多层神经网络的特点

- 非线性映射能力：神经网络能以任意精度逼近任何非线性连续函数。在建模过程中的许多问题正是具有高度的非线性。
- 并行分布处理方式：在神经网络中信息是分布储存和并行处理的，这使它具有很强的容错性和很快的处理速度。
- 自学习和自适应能力：神经网络在训练时，能从输入、输出的数据中提取出规律性的知识，记忆于网络的权值中，并具有泛化能力，即将这组权值应用于一般情形的能力。神经网络的学习也可以在线进行。
- 多变量系统：神经网络的输入和输出变量的数目是任意的，对单变量系统与多变量系统提供了一种通用的描述方式，不必考虑各子系统间的解耦问题。

BP神经网络的拓扑结构

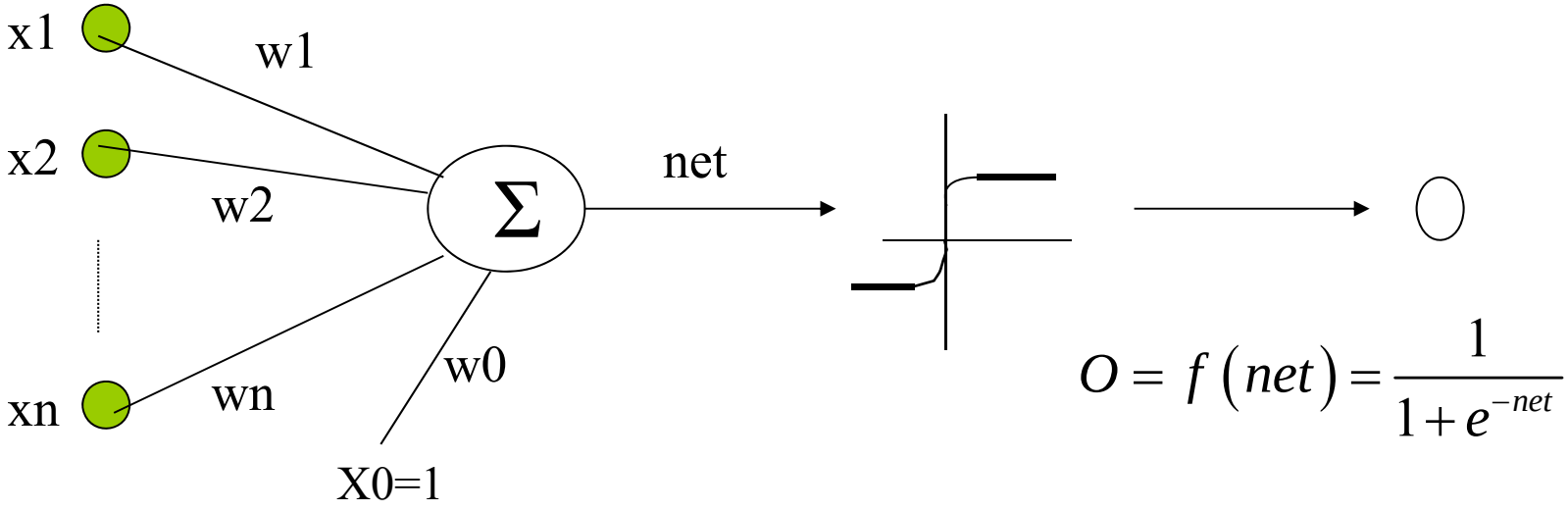


3-层 BP神经网络


target t
output z
input x
output
hidden
input
t_1 t_2 t_k t_c
z_1 z_2 z_k z_c
w_{kj}
y_1 y_2 y_j y_{n_H}
w_{ji}
x_1 x_2 x_i x_d

BP网络中的神经元模型

- Sigmoid 函数是PB网络中常用的神经元非线性函数



Sigmoid神经元函数的特点

Sigmoid神经元函数及其导数

$$f\left(x_1, x_2, \cdots, x_d\right) = f\left(\mathrm{w}^t \mathrm{x}\right) = \frac{1}{1+e^{-\mathrm{w}^t \mathrm{x}}}$$

$$\frac{df(y)}{dy} = f(y)\left(1 - f(y)\right)$$

BP网络的信号前向输出

神经元函数: $$f(net)=\frac{1}{1+e^{-net}}$$

隐层节点的输出： $$y_j=f\left(\sum_{i=1}^{d}w_{ji}x_i+w_{j0}\right)\qquad j=1,2,\cdots,n_H$$

输出层节点的输出：

$$z_k=f\left(\sum_{j=1}^{n_H}w_{kj}y_j+w_{k0}\right)=f\left(\sum_{j=1}^{n_H}w_{kj}\,f\left(\sum_{i=1}^{d}w_{ji}x_i+w_{j0}\right)+w_{k0}\right)$$

$$k=1,2,\cdots,c$$

BP神经网络的训练方法：学习

- 已经标记的样本集合. 集合中的每一个样本已设定一个期望输出值。
- “模式顺传播” 。对每一个样本, 计算其经过网络后的实际输出与期望输出的误差。
- “误差逆传播”。根据每一个样本的实际输出误差, 按照误差平方最小规则，由输出层往中间层逐层修正联结权值.
- “模式顺传播”和“误差逆传播”过程的交替反复进行，直到收敛.

求解优化问题的梯度下降技术

算法基本步骤：

1. 定义一个准则函数（能量函数、误差函数） $\boldsymbol{J}(\boldsymbol{w})$
2. 令 $k=1$, 选取任意的初始化解矢量 $\boldsymbol{w}$ 值，记为 $\boldsymbol{w}(k)$.
3. 计算 $\boldsymbol{J}(\boldsymbol{w}(k))$ 的梯度：$\nabla \boldsymbol{J}(\boldsymbol{w}(k))$
4. 沿负梯度方向计算新的解 $\boldsymbol{w}(k+1)$ ，

   (梯度的负方向是梯度最陡下降的方向).

一般地: $$w(k+1)=w(k)-\eta(k)\nabla J\left(w(k)\right)$$

这里，η 称为学习速率。

BP 学习算法的推导

学习目的: 对多层神经网络内部的互连接权值进行改变

基本方法: 使用梯度下降技术在权空间中寻找使误差函数达到全局最小的点所对应的权。

梯度下降技术可以找到一个极小点, 但不能保证一定可以找到全局最小点

定义BP网络的误差度量函数:

$$J(w) \equiv \frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2 = \frac{1}{2}\|t - z\|^2$$

BP网络——信号的前向传输

节点函数:

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

$$y_j = f\left(\sum_{i=1}^{d} w_{ji}x_i + w_{j0}\right) \rightarrow \frac{1}{1+e^{-\left(\sum_{i=1}^{d} w_{ji}x_i + w_{j0}\right)}} \qquad j=1,2,\cdots,n_H$$

$$z_k = f\left(\sum_{j=1}^{n_H} w_{kj}y_j + w_{k0}\right) = f\left(\sum_{j=1}^{n_H} w_{kj} f\left(\sum_{i=1}^{d} w_{ji}x_i + w_{j0}\right) + w_{k0}\right)$$

$$k=1,2,\cdots,c$$

BP网络——误差的反向传播

$$J(w) \equiv \frac{1}{2}\sum_{k=1}^{c}(t_k - z_k)^2 = \frac{1}{2}\|t - z\|^2$$

$$w(m+1) = w(m) + \Delta w(m) \qquad d \times n_H + n_H \times c$$

$$\Delta w = -\eta \frac{\partial J}{\partial w} \qquad \Delta w_{pq} = -\eta \frac{\partial J}{\partial w_{pq}}$$

输入节点→隐节点: $p = 1, 2, \cdots, n_H; \quad q = 1, 2, \cdots, d$

隐节点→ 输出节点: $p = 1, 2, \cdots, c; \quad q = 1, 2, \cdots, n_H$

BP网络——误差的反向传播-1：

计算每一个输出节点 *k* 的误差

$$z_k = f(net_k) = \frac{1}{1+e^{-net_k}} \qquad net_k = \sum_{j=1}^{n_H} w_{kj} y_j + w_{k0}$$

$$\frac{\partial J}{\partial w_{kj}} = \frac{\partial J}{\partial z_k} \cdot \frac{\partial z_k}{\partial net_k} \cdot \frac{\partial net_k}{\partial w_{kj}}$$

$$= -(t_k - z_k) \cdot f'(net_k) \cdot y_j$$

$$= -(t_k - z_k) \cdot f(net_k)(1 - f(net_k)) \cdot y_j$$

$$\Delta w_{kj} = -\eta \frac{\partial J}{\partial w_{kj}} = \eta (t_k - z_k) \cdot f(net_k)(1 - f(net_k)) \cdot y_j$$

BP网络——误差的反向传播-2：

计算每一个隐节点 *h* 的误差

$$\frac{\partial J}{\partial w_{ji}}=\frac{\partial J}{\partial y_j}\cdot\frac{\partial y_j}{\partial net_j}\cdot\frac{\partial net_j}{\partial w_{ji}}=\sum_{k=1}^{c}\left[\frac{\partial J}{\partial z_k}\cdot\frac{\partial z_k}{\partial net_k}\cdot\frac{\partial net_k}{\partial y_j}\right]\cdot\frac{\partial y_j}{\partial net_j}\cdot\frac{\partial net_j}{\partial w_{ji}}$$

$$J(w)\equiv\frac{1}{2}\sum_{k=1}^{c}(t_k-z_k)^2 \qquad z_k=f(net_k)$$

$$net_k=\sum_{j=1}^{n_H}w_{kj}y_j+w_{k0} \qquad y_j=f(net_j)$$

$$net_j=\sum_{i=1}^{d}w_{ji}x_i+w_{j0}$$

BP网络——误差的反向传播-2：

计算每一个隐节点 *h* 的误差

$$\frac{\partial J}{\partial w_{ji}} = -\sum_{k=1}^{c}\left[\left(t_k - z_k\right)\cdot f'\left(net_k\right)\cdot w_{kj}\right]\cdot f'\left(net_j\right)\cdot x_i$$

$$f'\left(net_k\right) = f\left(net_k\right)\left(1 - f\left(net_k\right)\right)$$

$$f'\left(net_j\right) = f\left(net_j\right)\left(1 - f\left(net_j\right)\right)$$

$$\Delta w_{ji} = -\eta\frac{\partial J}{\partial w_{ji}} = \eta\sum_{k=1}^{c}\left[\left(t_k - z_k\right)\cdot f'\left(net_k\right)\cdot w_{kj}\right]\cdot f'\left(net_j\right)\cdot x_i$$

BP网络——误差的反向传播-3：

更新网络的每一个权值

$$w_{kj}(m+1)=w_{kj}(m)+\Delta w_{kj}(m)$$

$$k=1,2,\cdots,c;\quad j=1,2,\cdots,n_H$$

$$w_{ji}(m+1)=w_{ji}(m)+\Delta w_{ji}(m)$$

$$j=1,2,\cdots,n_H;\quad i=1,2,\cdots,d$$

基本的BP学习算法

- **样本集：** $S=\{(X_1, t_1), (X_2, t_2), \ldots, (X_s, t_s)\}$
- **基本算法：**
  - 依次使用样本集中的样本 (X_k, t_k) 计算出实际输出 Z_k，计算与标准输出的误差 E_1；
  - 对 $W^{(1)}$，$W^{(2)}$，…，$W^{(L)}$ 各做一次调整；
  - 重复这个循环，直到 $\sum E_p<\varepsilon$。
- **学习算法的直观解释：误差反传**
  - 用输出层的误差调整输出层权矩阵，并用此误差估计输出层的直接隐含层的误差
  - 再用隐含层误差估计更前一层的误差，如此获得所有其它各层的误差估计
  - 用各层的这些估计误差分别对各层权矩阵进行修改

BP网络设计中的一些说明

BP算法的特点

- 特点：
  - 网络权重的调节用的是梯度下降算法
  - 容易推广到任意有向网络
  - 训练的时候迭代的次数可能很多，慢
  - 训练后，网络运行非常快
- 问题
  - 收敛性和局部极小值
  - 表征能力
  - 过拟合的问题

BP神经网络的设计

- 理论上早已经证明：具有1个隐层（采用Sigmoid转换函数）的BP网络可实现对任意函数的任意逼近。
  - 但遗憾的是，迄今为止还没有构造性结论，即在给定有限个（训练）样本的情况下，如何设计一个合理的BP网络模型，并通过所给的有限个样本的学习（训练）来满意地逼近样本所蕴含的规律（函数关系，不仅仅是使训练样本的误差达到很小）。
- 目前在很大程度上还需要依靠经验知识和设计者的经验，在国外被称为“艺术创造的过程”。
- 注意：目前深度神经网络采用的是多个隐含层的网络

确定合理的网络模型

合理网络模型必须具有

1. 合理隐层节点数
2. 训练时没有发生“过拟合”现象
3. 求得全局极小点
4. 同时综合考虑网络结构复杂程度和误差大小。

- 设计合理BP网络模型的过程是一个不断调整参数的过程，也是一个不断对比结果的过程，比较复杂且有时还带有经验性。

BP网络神经元函数需要具备的特性

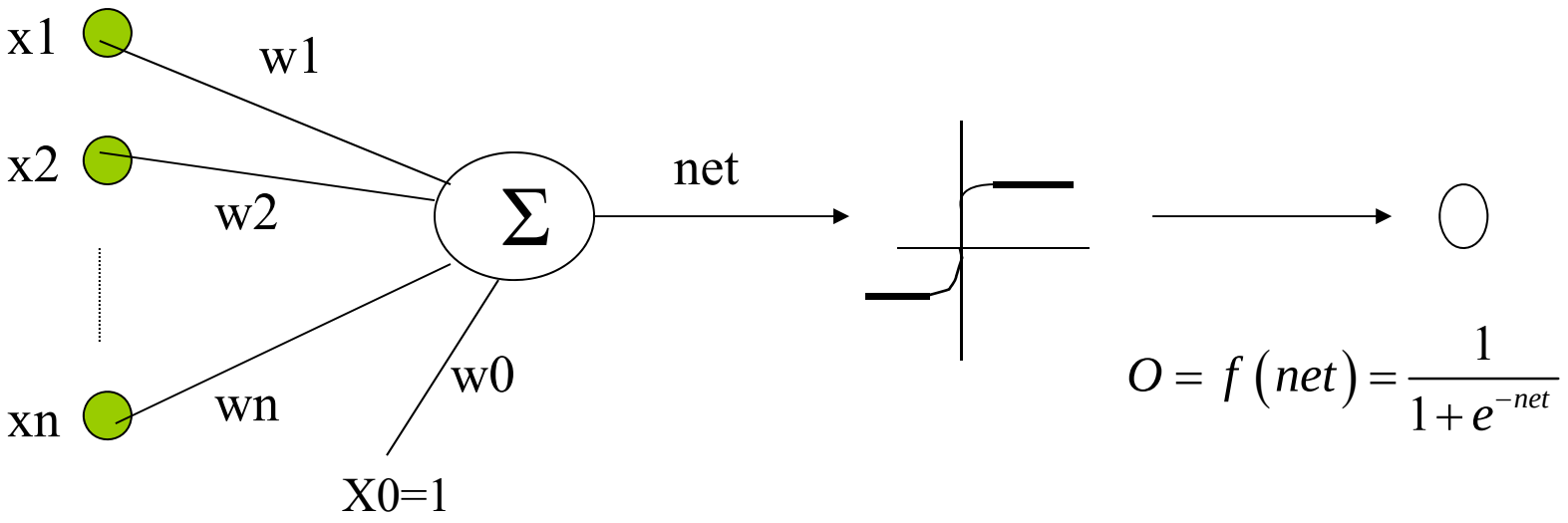
① 必须是**非线性的** (可以增强网络的运算能力)

② 必须具有**饱和特性** (权值、激活量是有界的)

③ 是**连续的**和**光滑的**. f(.) 及其一阶导数f'(.)存在.

④ 是**单调的** (是需要的，但不是必须的).

基本 Sigmoid 神经元函数



双曲正切形式的Sigmoid函数

$$f(net) = a\tanh(b \cdot net) = a \cdot \frac{1-e^{-2b \cdot net}}{1+e^{-2b \cdot net}} = \frac{2a}{1+e^{-2b \cdot net}} - a$$

$$f'(net) = ab - \frac{b}{a}(f(net))^2$$

- 其中的经验参数为 $a = 1.716$，$b = 2/3$

训练样本数据，预处理

- 收集、整理和分组
  - 使用BP神经网络的首要前提条件是有足够多、典型性好、精度高的样本。
  - 为在训练（学习）过程不发生“过拟合”，须将样本集合随机分成：训练样本、检验样本（10%以上）和测试样本（10%以上）3部分。
  - 数据分组时还应尽可能考虑样本模式间的平衡。

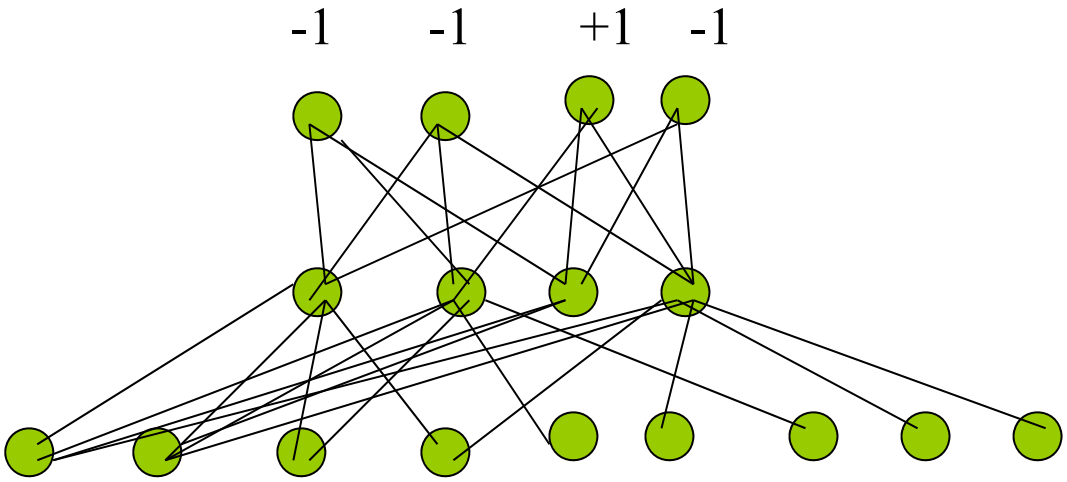
- 样本的预处理

若样本模式矢量的某一个特征分量的值远远大于其它的特征分量，则权的调整将主要受该特征分量变化的影响。而这是我们不希望的。解决的办法:

  - 在训练以前对所有的特征进行标准化
  - 使每个特征的均值为0
  - 方差为一个恒定值（如1.0）

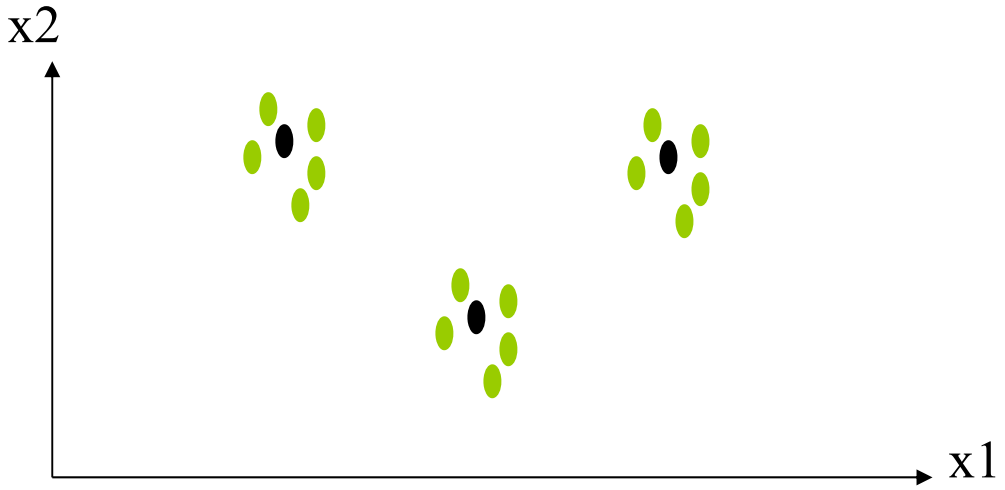
网络输出的目标量

- 一般使用 +1 表示目标类别，使用 –1表示非目标类别
  - 例如：对于具有 4 类的分类问题，若目标类别是第 3 类，则输出模式可以是 (-1, -1, +1, -1).



小样本情况下，样本集的扩展

- 若训练样本集合较小，则可以通过在已有样本上加入噪声生成新的训练样本，并将这些样本看成是由相同分布产生的正常样本使用。
  - 例如：向真实的样本中加入 *d*-维高斯噪声.



BP神经网络拓扑结构：隐层数

- 一般认为，增加隐层数可以降低网络误差，提高精度，但也使网络复杂化，从而增加了网络的训练时间，出现“过拟合”的倾向。

- Hornik等证明：若输入层和输出层采用线性转换函数，隐层采用Sigmoid转换函数，则含一个隐层的BP网络能够以任意精度逼近任何有理函数。
  - 在设计BP网络时**应优先考虑 3 层BP网络（即有 1 个隐层）。**
  - 一般地，靠增加隐层节点数来获得较低的误差，其训练效果要比增加隐层数更容易实现。

隐层节点数

1. 隐单元数决定了神经网络的表示能力，决定了决策边界的复杂性。
   - 目前理论上还没有一种科学、普遍的确定隐层节点数的方法
2. 确定隐层节点数的最基本原则是：在满足精度要求的前提下取尽可能紧凑的结构，即取尽可能少的隐层节点数。
   - 隐层节点数不仅与输入/输出层的节点数有关，与需解决的问题的复杂程度有关，与样本数据的特性等因素有关。
   - 若样本容易区分，则需要的节点可以很少；反之，对于复杂问题则需要的内部节点多

3. 确定隐层节点数须满足下列条件：
   - 隐层节点数须小于 n-1（其中 n 为训练样本数），否则，网络模型极易过拟合，网络模型失去泛化能力。
   - 训练样本数必须多于网络模型的连接权数，一般为 2~10 倍
   - 一个简便的经验化规则是：选取隐单元的个数满足使网络中的权个数约为 $n / 10$, 这里 n 是训练样本数

隐层节点数

- 若隐层节点数太少，网络可能根本不能训练或网络性能很差；

- 若隐层节点数太多，虽然可使网络的系统误差减小，但一方面使网络训练时间延长，另一方面，训练容易陷入局部极小点而得不到最优点，这是训练时出现“过拟合”的内在原因。

- 合理隐层节点数应在综合考虑网络结构复杂程度和误差大小的情况下，用节点删除法和扩张法确定。

权值初始化

- BP学习算法决定了误差函数一般存在（很）多个局部极小点，不同的网络初始权值直接决定了BP算法收敛于哪个局部极小点或是全局极小点。

- 要求计算程序必须能够自由改变网络初始连权值。考虑到Sigmoid函数的特性，一般选择初始权值分布在-0.5~0.5之间的均匀分布的随机数比较有效。

- 一些标准通用软件，如Statsoft公司出品的Statistica Neural Networks软件和Matlab 软件，可以设定初始权值

学习率

- **不同的学习率对网络性能有显著的影响，影响系统学习过程的稳定性。**
  - 大的学习率可能使网络权值每一次的修正量过大，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；
  - 过小的学习率导致学习时间过长，不过能保证收敛于某个极小值
  - 一般倾向选取较小的学习率以保证学习过程的收敛性（稳定性），通常学习率在0.01~0.8之间。
- **最优学习率:**
  - 经过一步学习后，引起的局部误差最小的学习率称为**最优学习率**
  - 设置学习率的主要原则：对每一个权分别设置最优的学习率

冲量项

- 在误差平面中也许会存在一些平坦区域，在这些区域中误差函数的导数非常小。为了避免网络训练陷于较浅的局部极小点可增加冲量项
  - 理论上其值大小应与权值修正量的大小有关，但实际应用中一般取常量。通常在0~1之间，而且一般比学习率要大。
- 需要指出的是:
  - 当权值太多时，就会有平坦区域。当平坦区域存在时，冲量会使学习加快。我们可以通过修改学习规则，使其包含以前权值更新量的α 倍:

$$w(m+1)=w(m)+(1-\alpha)\cdot\Delta w_{pq}(m)+\alpha\cdot\Delta w_{pq}(m-1)$$

  - 若 α = 0, 规则是标准的BP算法。若α = 1 反向传播被忽略，权值矢量以恒定速率运动。

权衰减

- 为避免过拟合，可以采用在逼近最优解时对权值集合进行衰减方法
- 方法之一是：在每一步更新后，对权值进行修正：

$$w^{new} = w^{old}\left(1-\varepsilon\right)$$

这里 0 < ε < 1.

- 权衰减的结果是，在准则函数变小的同时，那些不需要的权值被减小了

BP例子程序介绍

MATLAB神经网络工具箱

1. Matlab 6.0 以上的软件，对于BP神经元网络的训练使用了Neural Networks Toolbox for Matlab。

2. 美国的Mathwork公司推出的MATLAB软件包既是一种非常实用有效的科研编程软件环境，又是一种进行科学和工程计算的交互式程序。

3. MATLAB本身带有神经网络工具箱，可以大大方便权值训练，减少训练程序工作量，有效的提高工作效率.

其它的学习方法和网络

- 其它的学习方法
  - Hebb学习
  - 概率式学习
  - 竞争学习
- 其它的网络及学习算法
  - 对传网：CPN
  - 统计网络
    - 基本训练方法
    - 模拟退火算法
    - Cauchy训练
    - Boltzmann机训练
  - 循环网络：Hopfield网络
  - 自适应共振理论：ART

神经网络的其它形式

- 卷积网络
- 递归网络
- 级联相关