人工神经元网络
(Artificial Neural Network)

张文生

中国科学院自动化所

- 什么叫人工神经元网络
  - 采用物理可实现的系统来模仿人脑神经细胞的结构和功能的系统。
- 为什么要研究ANN
  - 人工智能主要是研究人的智能
  - 人脑是人的智能的载体
    - 识别图像, 声音, 语言, 解决问题
  - 模拟人脑
    - 神经元网络
  - 研究层次
    - 认知心理学
    - 神经生理学

解决的问题

- 带决策属性的样本集学习
  - 条件属性值是数字，而非符号；
  - 决策属性值是类别(字符识别)或数字(自动驾驶);

属性1	属性2	属性3	决策属性
1	25	1.5	1
2	88	2.3	2
1	63	4.7	2

- 目的是找到知识, 对样本集正确分类或尽量接近决策属性值; 继而泛化;

知识表示与学习

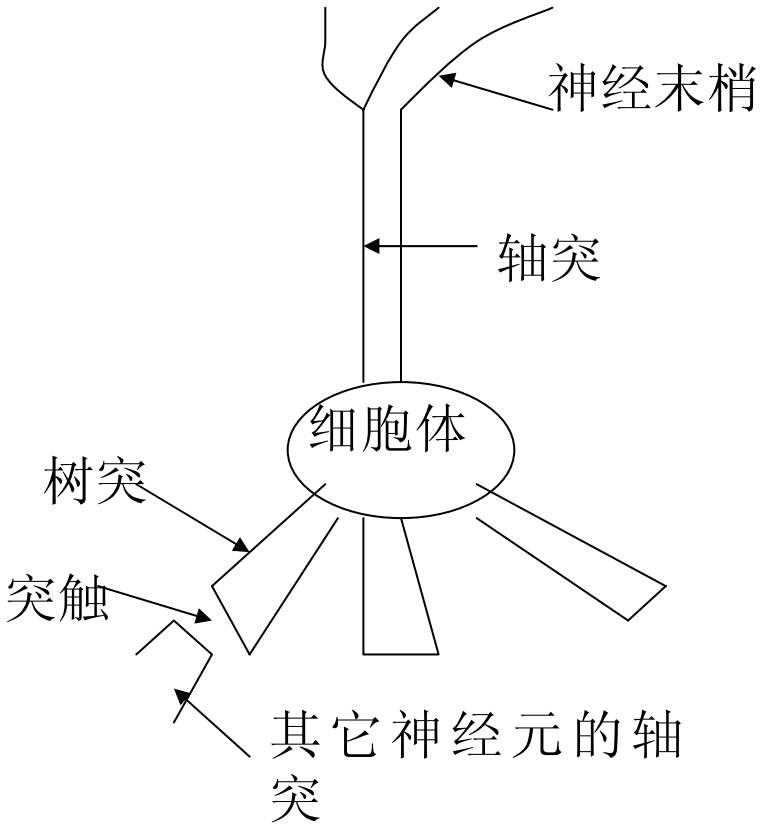
- 知识的基函数是网络形式;
- 学习的任务是找到正确的参数;
  - 权值、网络结构
- 参数的初始化是随机的，对样本集而言，不一定分类正确;
- 学习过程就是调整参数的过程;

- 神经元模型
- M-P模型与感知机
- Hebb学习与 δ 学习
- XOR问题与函数型联接神经网络
- 多层感知机与BP算法

早期研究

- 1890年，美国生物学家W.James出版了《Physiology》（生理学）一书。首次阐明了有关人脑结构及其功能，以及相关学习、联想、记忆的基本规律。指出：人脑中当两个基本处理单元同时活动，或两个单元靠得比较近时，一个单元的兴奋会传到另一个单元。而且一个单元的活动程度与他周围的活动数目和活动密度成正比。

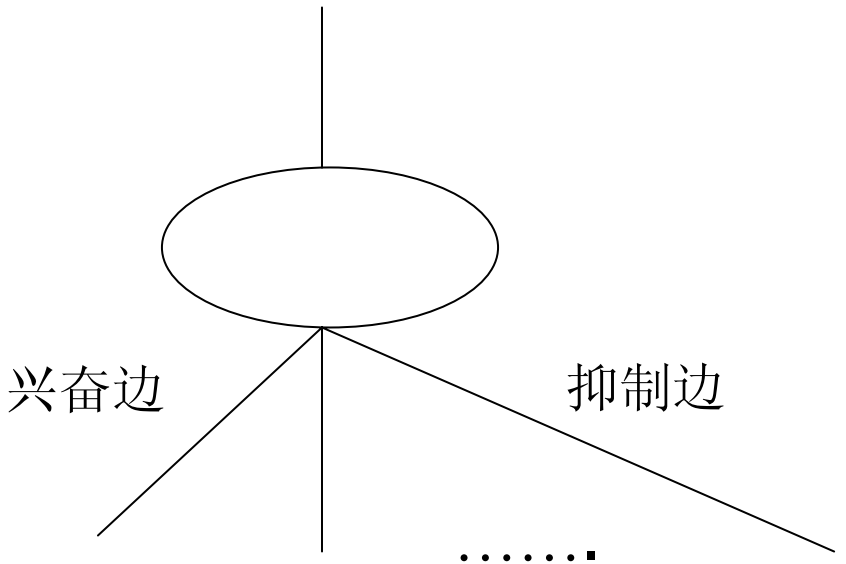
神经元模型



- 神经元模型
- M-P模型与感知机
- Hebb学习与 δ 学习
- XOR问题与函数型联接神经网络
- 多层感知机与BP算法

M-P模型

- 1943年McCulloch和Pitts发表文章，提出M-P模型。描述了一个简单的人工神经元模型的活动是服从二值（兴奋和抑制）变化的。总结了神经元的基本生理特性，提出了神经元的数学描述和网络的结构方法。——标志神经计算时代的开始


兴奋边
抑制边
……

- 神经元的输入有两种类型:兴奋边/抑制边;
- 神经元的输出有两种状态, 兴奋/抑制;
- 如果一条抑制边处于激活状态, 则神经元处于抑制状态;
- 如果没有抑制边处于激活状态, 则当兴奋边的数目超过一个阈值时, 神经元处于兴奋状态, 否则处于抑制状态;

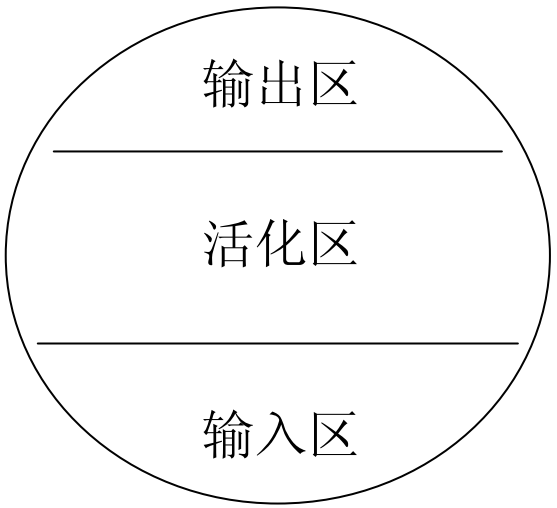
特点

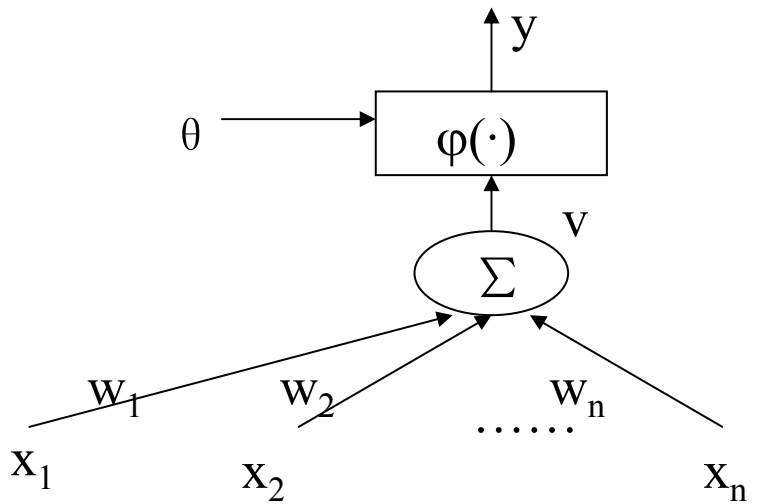
- 边的权值都为固定值1, 无法调整;
- 阈值;
- 输出是0/1;
- 一票否决;

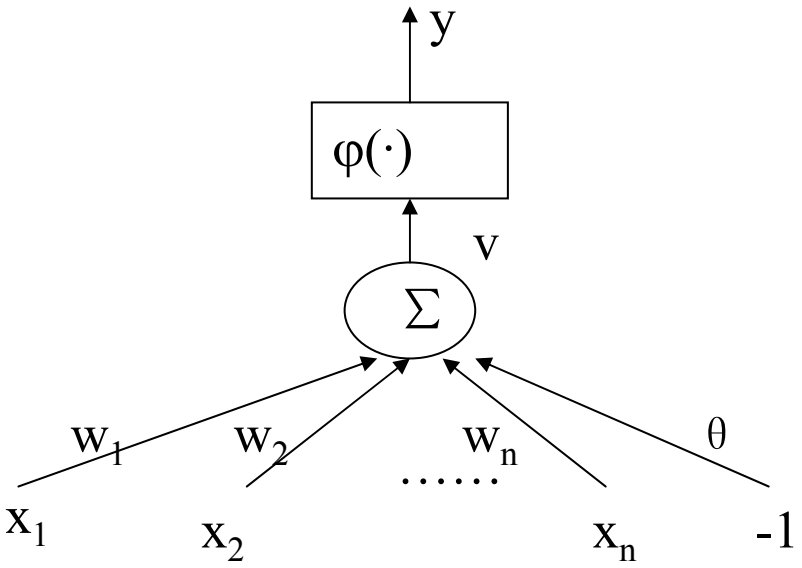
改进

- 统一兴奋边和抑制边
- 边的权值都为固定值1 -> 权值可调整
- 一票否决 -> 相对抑制

改进的神经元模型



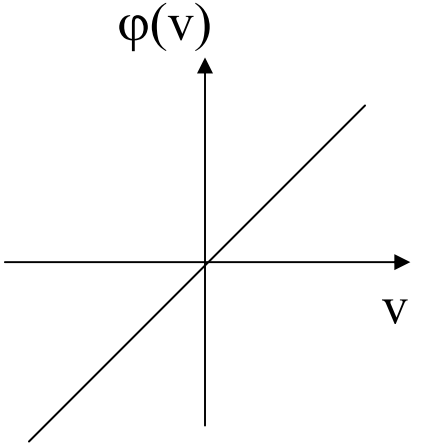




$$v = \sum_{i=1}^{n} w_i * x_i - \theta$$

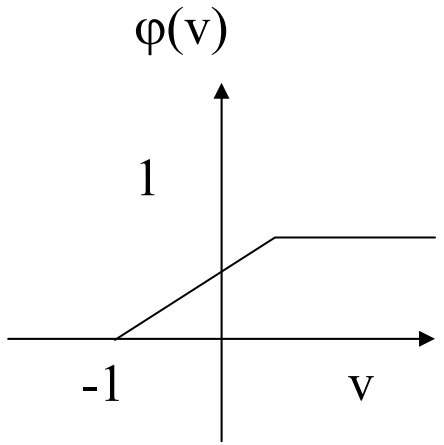
激活函数(活化函数/作用函数)

- 线性活化函数
  - φ(v) = v



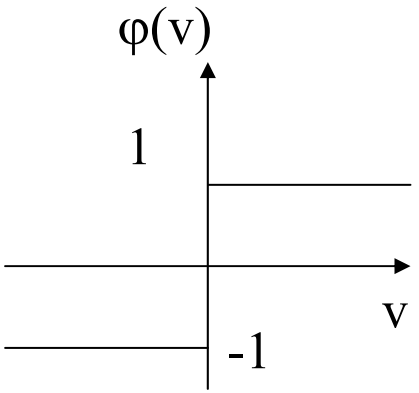
- 分段线性函数

$$\varphi(v)=\begin{cases}1 & v\geq 1\\ \dfrac{1+v}{2} & -1<v<1\\ 0 & v\leq -1\end{cases}$$



- 阈值函数：

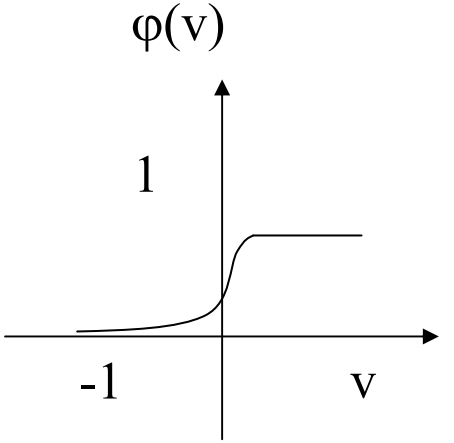
$$\varphi(v)=\begin{cases}1 & v\geq 0\\ -1 & v<0\end{cases}$$



$$v=\sum_{i=1}^{n} w_i * x_i-\theta$$

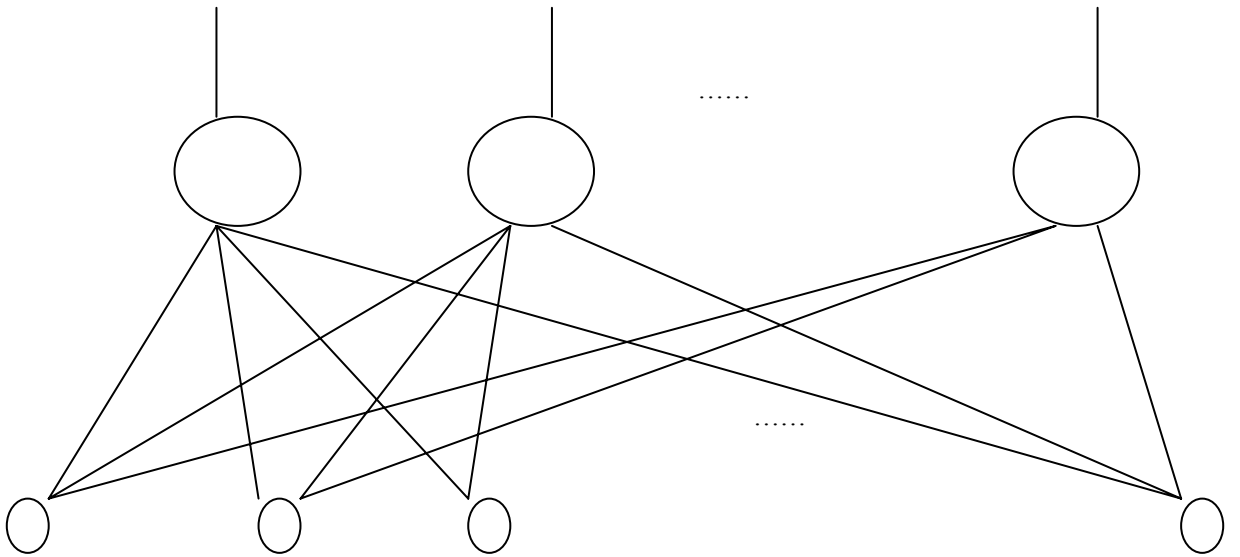
- sigmoid函数

$$\varphi(v) = \frac{1}{1 + e^{-\alpha v}}$$



感知机

1957年Frank Rosenblatt定义了一个神经网络结构，称为感知机(感知器, Perceptron) 。



- **意义**：第一次把神经网络研究从纯理论的探讨推向工程实现，在IBM704计算机上进行了模拟，证明了该模型有能力通过调整权的学习达到正确分类的结果。掀起了神经网络研究高潮。

学习

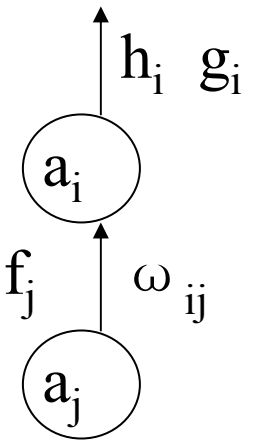
- 一个细胞受到触发并输出信号后，它的物化性质又恢复到原先的状态，然而细胞与细胞之间的联接状态却会由于多次传递信号而起趋于加强.
- 知识似乎并不存贮在神经细胞之内，而是存在于各细胞间的联接.
- 对联接强度的修改变化则是学习过程的表现.

- 神经元模型
- M-P模型与感知机
- Hebb学习与 δ 学习
- XOR问题与函数型联接神经网络
- 多层感知机与BP算法

Hebb学习

- 基本思想是心理学家Hebb于1949年提出
- Hebb认为，神经元突触的强度是可变的。学习就是在突触上发生的
- 突触强度: 权值

- Hebb认为，如果两个神经元a_i，a_j相连，且a_j的输出连接到a_i的输入，则如果a_i和a_j的激活状态同步（同为激活或抑制），则a_j 的状态与a_i有密切的关系，应该增强a_i和a_j之间的连接。反之，则应减弱；



f_j 是a_j 的真实输出；

h_i 是a_i 的真实输出；

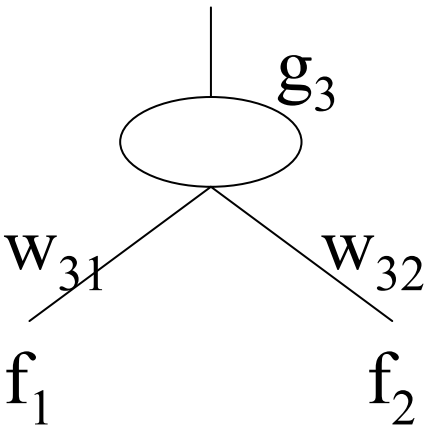
g_i 是a_i 的理想输出；

- 调整公式： $\Delta \omega_{ij} = \alpha * g_i * f_j$

$\alpha > 0$

例1

样本	f_1	f_2	g_3
S_1	1	1	1
S_2	-1	1	-1

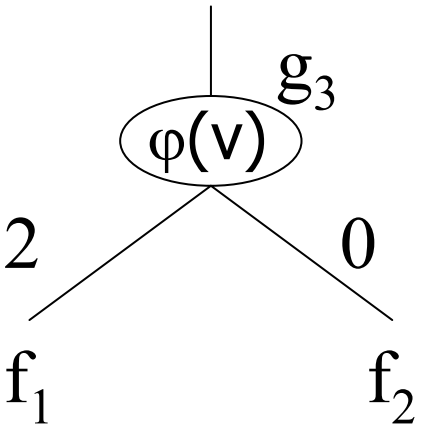


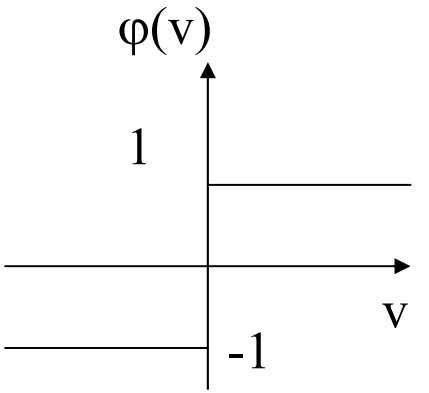
- 激活函数使用阈值函数, θ=0。
- ω 初始化为（0，0），α =1。
- $\Delta \omega_{31} = \sum g_3 * f_1$

  $= 1*1+(-1)*(-1) = 2$

  (∑对第一个和第二个样本求和)
- $\Delta \omega_{32} = \sum g_3 * f_2 = 1*1+(-1)*1 = 0$

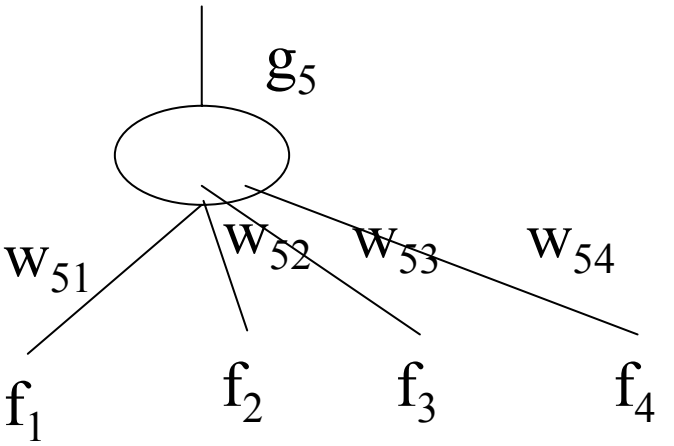
样本	f_1	f_2	g_3
S_1	1	1	1
S_2	-1	1	-1





例2

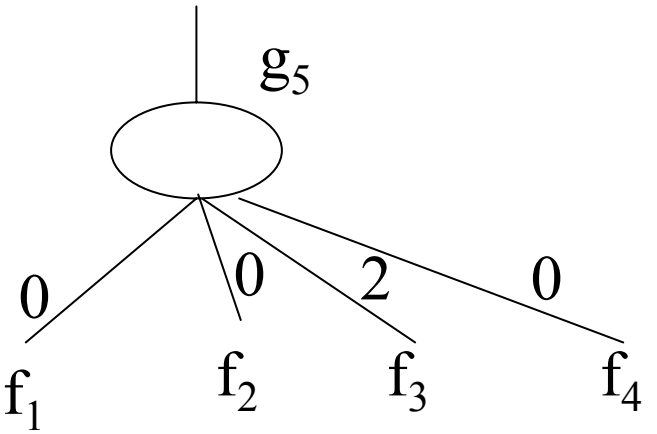
样本	f_1	f_2	f_3	f_4	g_5
S_1	1	-1	1	-1	1
S_2	1	1	1	1	1
S_3	1	1	1	-1	-1
S_4	1	-1	-1	1	-1



- 激活函数使用阈值函数, θ=0。
- ω 初始化为（0，0），α =1。

- $\Delta\omega 51 = \sum g5 * f1 = 1*1 + 1*1 + -1*1 + -1*1 = 0$
- $\Delta\omega 52 = \sum g5 * f2 = 1*-1 + 1*1 + -1*1 + -1*-1 = 0$
- $\Delta\omega 53 = \sum g5 * f3 = 1*1 + 1*1 + -1*1 + -1*-1 = 2$
- $\Delta\omega 54 = \sum g5 * f4 = 1*-1 + 1*1 + -1*-1 + -1*1 = 0$

样本	f_1	f_2	f_3	f_4	g_5
S_1	1	-1	1	-1	1
S_2	1	1	1	1	1
S_3	1	1	1	-1	-1
S_4	1	-1	-1	1	-1



例1与例2的区别

- 例1：两个样本的对应输入的乘积之和为0。即，1*-1 + 1*1 = 0

样本	f_1	f_2	g_3
S_1	1	1	1
S_2	-1	1	-1

- 而例2不满足：S_1和S_3的对应输入的乘积之和不为0。1*1 + -1*1 + 1*1 + -1*-1 = 2

样本	f_1	f_2	f_3	f_4	g_5
S_1	1	-1	1	-1	1
S_2	1	1	1	1	1
S_3	1	1	1	-1	-1
S_4	1	-1	-1	1	-1

- 样本可以看作空间中的矢量，两个样本的对应输入的乘积就是两个矢量的内积。如果两个矢量的内积为0，称为正交。

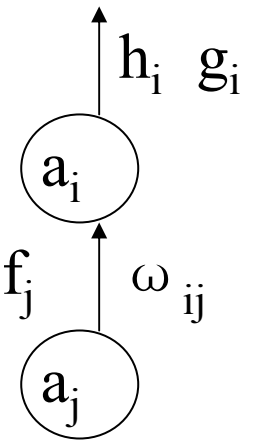
$$\sum_{p=1}^{n} fps \bullet fpt = 0$$

s,t是样本，n是输入量的数目。

- 猜测：对Hebb学习规则，如果样本集中任意两个样本正交，则学习出的权值可以保证对样本集中所有样本正确分类；否则，不能保证.

- 学习后的权值：

$$\omega_{ij} = \alpha \sum_{k=1}^{m} f_{jk} \bullet g_{ik}$$



m是样本数

$$\Delta \omega_{ij} = \alpha * g_i * f_j$$

- 激活函数使用阈值函数, $\theta=0$
- 对样本集中任一个样本S，f_{ps}是它的第p个输入，考虑神经网络的第i个输出节点的输出h_{is}：

$$h_{is} = \varphi\left(\sum_{p=1}^{n} \omega_{ip} f_{ps}\right)$$

n是输入节点的数目

$$h_{is} = \varphi\left(\sum_{p=1}^{n} \omega_{ip} f_{ps}\right)$$

$$= \varphi\left(\alpha \sum_{p=1}^{n} \sum_{k=1}^{m} f_{pk} \bullet g_{ik} f_{ps}\right)$$

$$= \varphi\left(\alpha \sum_{k=1}^{m} g_{ik} \sum_{p=1}^{n} f_{pk} f_{ps}\right)$$

$$= \varphi(\beta \bullet g_{is}) \qquad \beta > 0$$

δ 学习规则

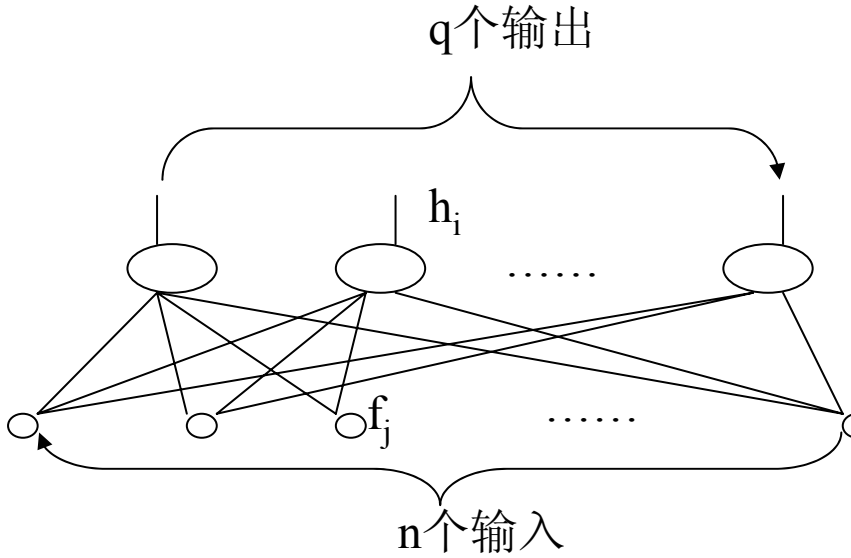
- 1960年，Widrow，Hoff提出广义Hebb学习方法，称为 δ 学习规则.

$$E = \sum_{p=1}^{m} E_p$$

$$E_p = \frac{1}{2}\sum_{i=1}^{q}(g_i - h_i)^2$$

m个样本，q个输出

$$\frac{\partial E_p}{\partial \omega_{ij}} = \frac{\partial E_p}{\partial h_i} \cdot \frac{\partial h_i}{\partial \omega_{ij}}$$



$$\frac{\partial E_p}{\partial h_i} = \frac{\partial(\frac{1}{2}\sum_{i=1}^{q}(g_i - h_i)^2)}{\partial h_i} = -(g_i - h_i)$$

$$h_i = \sum_{j=1}^{n} \omega_{ij} \cdot f_j - \theta \qquad \frac{\partial h_i}{\partial \omega_{ij}} = f_j$$

$$\frac{\partial E_p}{\partial \omega_{ij}} = -(g_i - h_i) \cdot f_j$$

$$\Delta \omega_{ij} = \beta \cdot (g_i - h_i) \cdot f_j \qquad \beta > 0$$

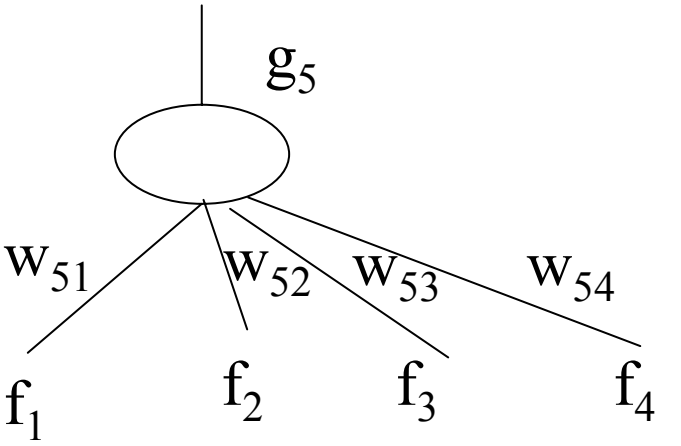
$$\delta_i = (g_i - h_i)$$

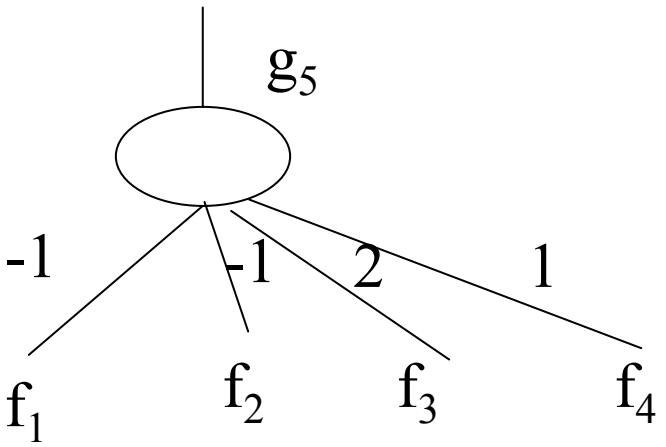
$$\Delta\omega_{ij} = \beta \cdot \delta_i \cdot f_j$$

Hebb: $\Delta\ \omega_{ij} = \alpha * g_i * f_j$

例子

样本	f_1	f_2	f_3	f_4	g_5
S_1	1	-1	1	-1	1
S_2	1	1	1	1	1
S_3	1	1	1	-1	-1
S_4	1	-1	-1	1	-1





- β=0.25, w51=…w54=0

- 虽然δ学习规则是以线型函数为激活函数的，但可以推广到阈值函数。

- Minsky给出一个基于δ学习规则的感知机算法。
- 样本集包含n个样本，前k个属于第一类，后n-k个属于第二类。
- 激活函数使用阈值函数, θ=0。
- 调整权值, 使:
  - 前k个样本满足 ω*s≥0;
  - 后n-k个样本满足ω*s<0;

1 初始化向量ω。

2对n个样本中的任一个样本s_i，如果满足：若$1 \leqslant i \leqslant k$，则$\omega * s_i \geqslant 0$，若$k < i \leqslant n$，则$\omega * s_i < 0$，则ω已将所有样本正确分类，算法结束；

3 随机生成[1，n]之间的整数i=random(1, n)，选择第i个样本s_i；

3.1 若$1 \leqslant i \leqslant k$，且$\omega * s_i \geqslant 0$，

或$k < i \leqslant n$，且$\omega * s_i < 0$，

则删除s_i；转3；

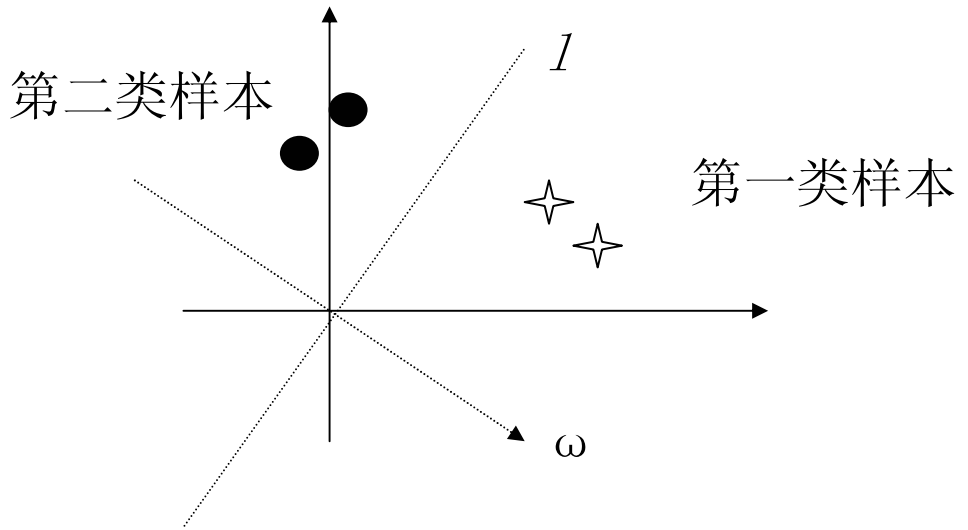
3.2 若$1 \leqslant i \leqslant k$（必有$\omega * s_i < 0$），$\omega \leftarrow \omega + s_i$；

若$k < i \leqslant n$（必有$\omega * s_i \geqslant 0$），$\omega \leftarrow \omega - s_i$；

转2；

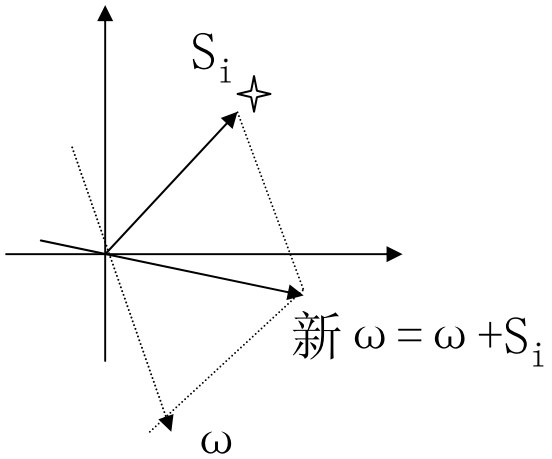
几何解释

- 空间中两个矢量的乘积：
  $\omega * s_i = |\omega| * |s_i| * \cos\alpha$ 。
  其中 α 是两个矢量的夹角。
- 若 $\omega * s_i \geq 0$,则意味着 ω 与s_i的夹角 α 满足：$-90 \leqslant \alpha \leqslant 90$。
- 而 $\omega * s_i < 0$则意味着 ω 与s_i的夹角 α 满足：$90 < \alpha < 270$。

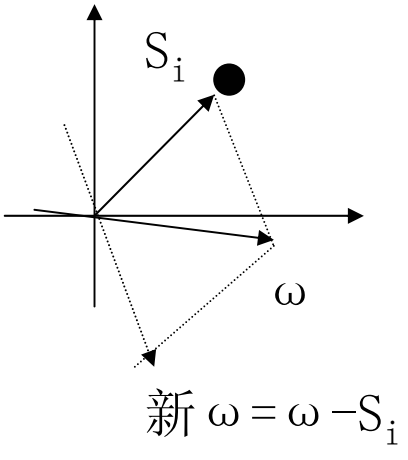

第二类样本
l
第一类样本
ω

调整权值的几何解释

第一类样本



第二类样本



例子

样本	X1	X2	类别
S_1	1	0	1
S_2	1	1	2

1. 初始化 $\omega=(\omega 1, \omega 2)=(0,0)$。
2. 对S_1，$\omega * s_1 = (0,0) * (1,0) = 0 \geqslant 0$,
   对S_2，$\omega * s_2 = (0,0) * (1,1) = 0 \geqslant 0$,
   由于不满足结束条件，进入第3步。
3. 随机产生1，对S_1，$\omega * s_1 \geqslant 0$, 删除S_1。转3。
   对S_2，$\omega = \omega - S_2$=（0，0）-（1，1）=（-1，-1）。转2。
4. 对S_1，$\omega * s_1 = (-1,-1) * (1,0) = -1 < 0$,
   由于不满足结束条件，进入第3步。

5. 随机产生2，对S_2，$\omega * s_2=(-1,-1)*(1,1)= -2 < 0$，删除$S_2$。转3。

对S_1，$\omega * s_1=(-1,-1)*(1,0)=-1<0$，

$\omega = \omega + S_1=$（-1，-1）+（1，0）=（0，-1）。转2。

6. 对S_1，$\omega * s_1 = (0,-1) * (1,0) =0 \geqslant 0$，

对S_2，$\omega * s_2 = (0,-1) * (1,1) =-1<0$，

由于满足结束条件，结束。

总结

- 算法的目的是找到ω，然后用ω*X=0作为分界面将样本集分开。如果存在这样的ω，则称样本集线性可分；否则，称为线性不可分.
  - 如果样本是二维的,则ω称为分界线
  - 如果样本是三维的,则ω称为分界面
  - 如果样本超过三维,则ω称为分类超平面

- 如果样本集线性可分，则以上算法一定在有限步内终止，并找到 ω 。
- 存在的 ω 可能不止一个.
- 多次调整、不断迭代
- 泛化

成功应用

- 感知机被应用于如气候预测、心电图分析、人造机器视觉等许多不同的领域．这些早期的成功曾使人们一度产生了对人工神经网络的乐观看法，似乎只要建造一个足够大、足够复杂的网，就可以仿制出人类脑神经系统的功能．

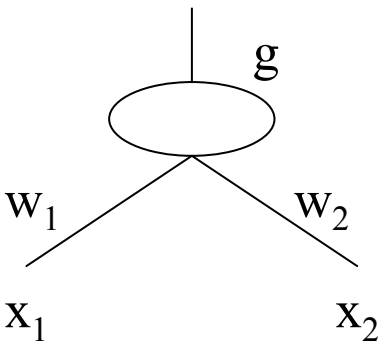
- 神经元模型
- M-P模型与感知机
- Hebb学习与 δ 学习
- XOR问题与函数型联接神经网络
- 多层感知机与BP算法

批判

- Marvin Minsky and Seymour Papert.
  Perceptrons; *an introduction to computational geometry*.
  MIT Press, Cambridge, MA, 1969.

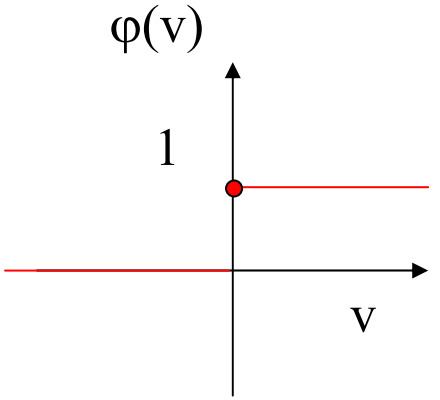
异或问题（XOR问题）

样本	x_1	x_2	g
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0



- 用单层感知机解决不了XOR问题
- 说明:
  - 激活函数使用阈值函数

$$v = \sum_{i=1}^{n} w_i * x_i - \theta$$



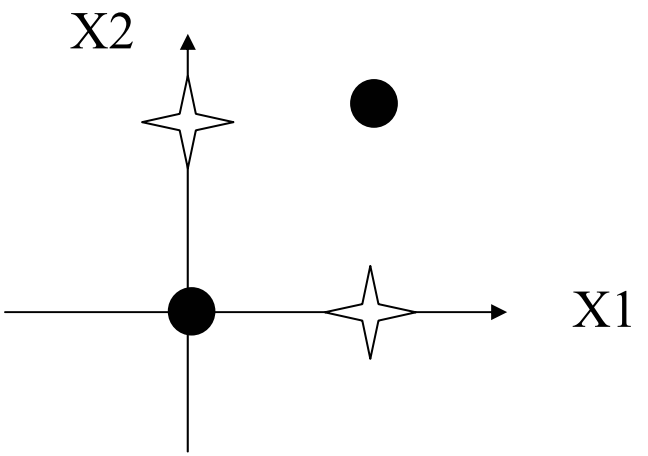
样本	x_1	x_2	g
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0

$$v = \sum_{i=1}^{n} w_i * x_i - \theta$$

- 若存在w和θ使感知机对XOR问题正确分类，
- 则将样本1代入，$-\theta < 0$ （1）
- 将样本2代入，$w_2 \geqslant \theta$ （2）
- 将样本3代入，$w_1 \geqslant \theta$ （3）
- 将样本4代入，$w_1 + w_2 - \theta < 0$，即，$w_1 + w_2 < \theta$ （4）
- 由(2),(3)，$w_1 + w_2 \geqslant 2\theta$，与（1）（4）矛盾。
- 所以不存在这样的w和θ。

线性不可分

样本	x_1	x_2	g
1	0	0	0
2	0	1	1
3	1	0	1
4	1	1	0



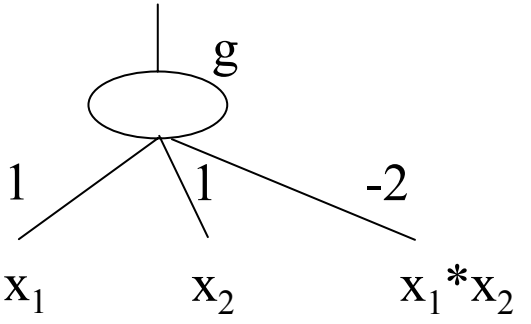
- 对这种非线性问题，两种解法：
  - 函数型联接神经网络
  - 多层感知机

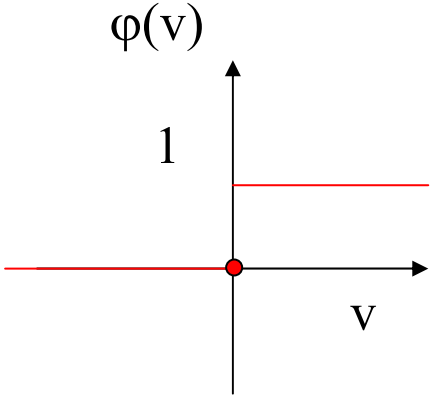
- 函数型联接神经网络的思想:
  - 扩维

样本	x_1	x_2	$x_{1*}\ x_2$	g
1	0	0	0	0
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0

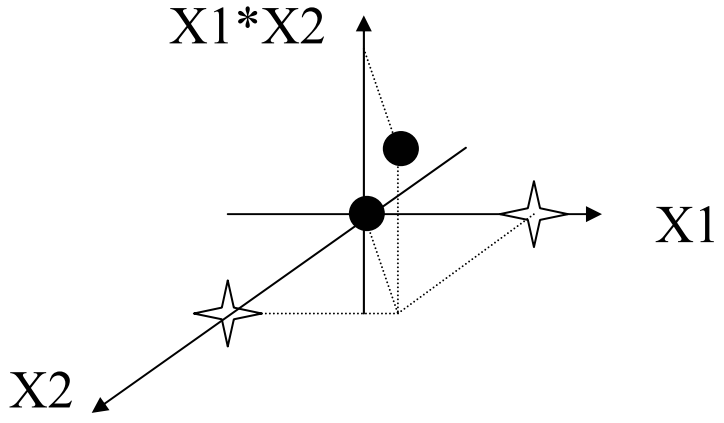
- 用感知机学习算法学习出以下结构：

样本	x_1	x_2	$x_{1*}\ x_2$	g
1	0	0	0	0
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0





从几何角度看



样本	x_1	x_2	$x_{1*}\ x_2$	g
1	0	0	0	0
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0

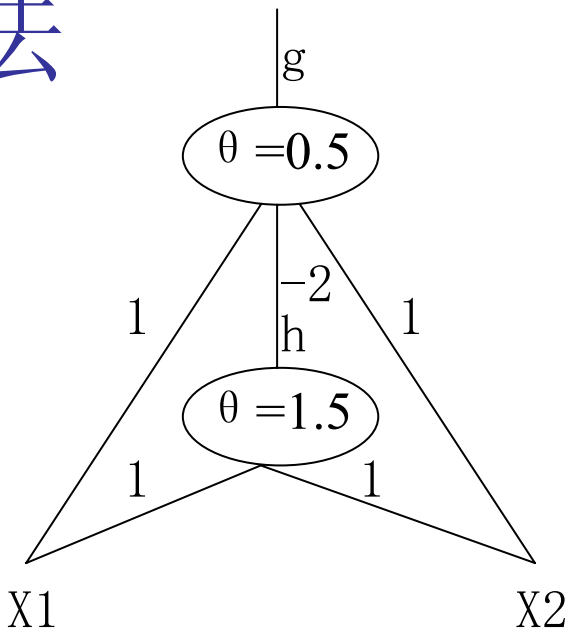
分界面为x+y-2z-1=0，即，z=(x+y-1)/2

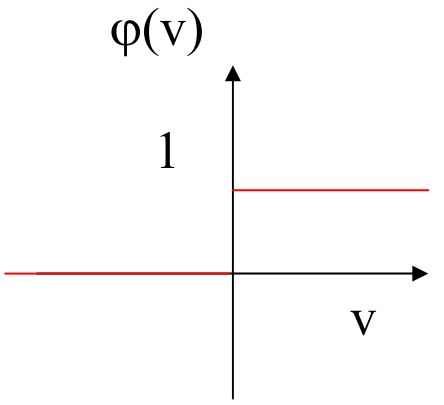
- 缺点:
  - 如何升维，没有一般的方法
  - SVM的核函数

- 神经元模型
- M-P模型与感知机
- Hebb学习与 δ 学习
- XOR问题与函数型联接神经网络
- 多层感知机与BP算法

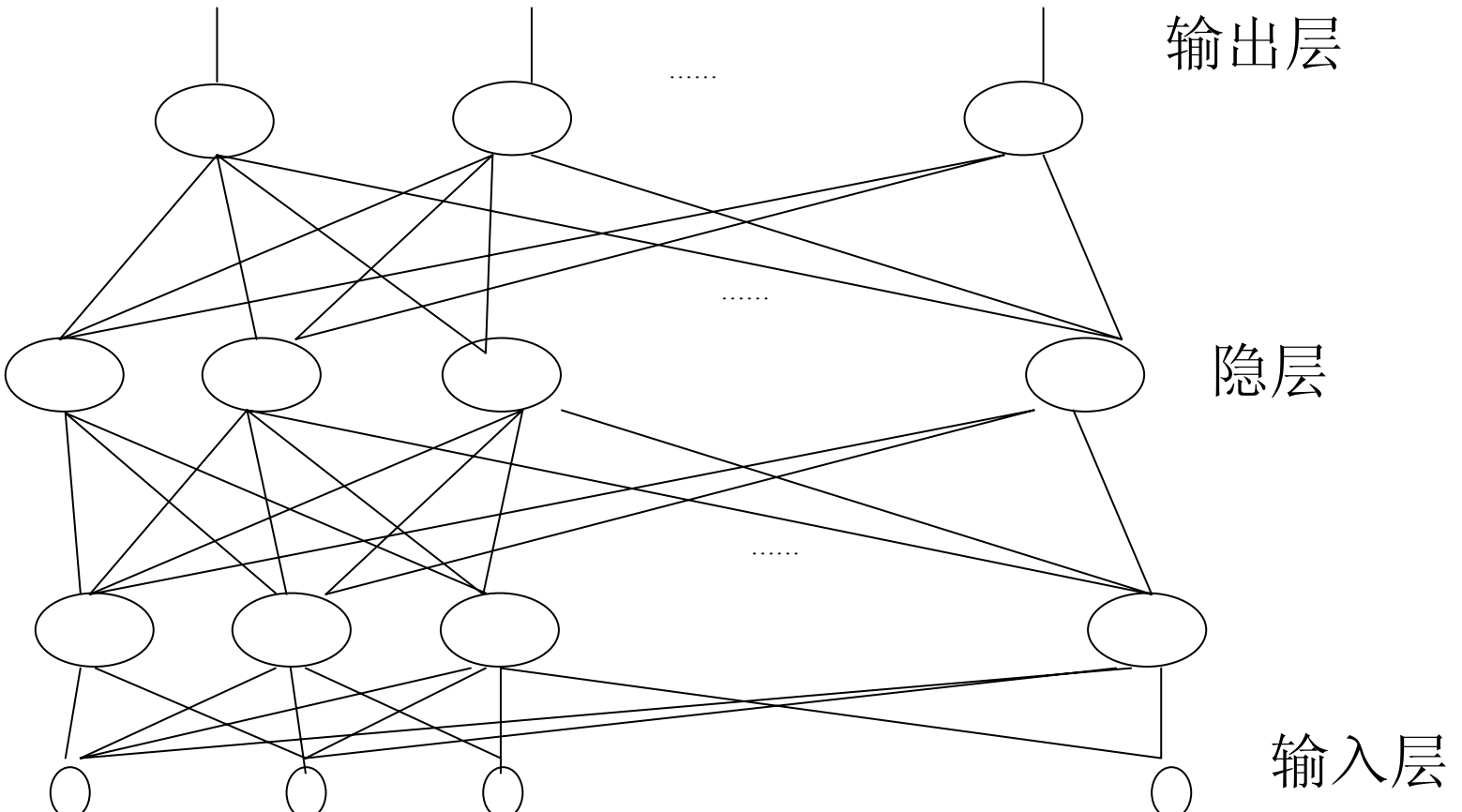
Minsky给出的方法

样本	x_1	x_2	h	g
1	0	0	0	0
2	0	1	0	1
3	1	0	0	1
4	1	1	1	0





- 这种方法的本质是多层感知机.



- 多层感知机其实在单层感知机提出后不久就提出了，但一直没有通用的算法;
- 在Minsky提出对感知机的批评后，其研究陷入低潮.
- 七十年代,据说全球只有几十个人在研究，但还是成果的。如：日本Fukusima的Neocognitron； 芬兰Kohonen的自组织神经网络；Stephen Crossberg的共振自适应理论ART网络等 。

- 1982年John J. Hopfield（物理学家）提出了全联接网络，离散的神经网络模型。
  - 全新的具有完整理论基础的神经网络模型。
  - 基本思想是对于一个给定的神经网络，设计一个能量函数，这个能量函数是正比于每一个神经元的活动值和神经元之间的联接权。而活动值的改变算法是向能量函数减少的方向进行，一直达到一个极小值为止。
  - 证明了网络可达到稳定的离散和连续两种情况。
  - 3 年后AT&T等做出了半导体芯片。
  - 神经网络复兴时期开始。

- 1986年，Rumelhart提出多层感知机的通用算法: BP算法（Back Propagation）
- 1987年在美国召开了第一届世界神经网络大会1000人参加。

BP算法

- BP算法称为推广的δ学习规则

$$\delta_i = (g_i - h_i)$$

$$\Delta\omega_{ij} = \beta \cdot \delta_i \cdot f_j$$

从上向下递推的思想

$$E = \sum_{p=1}^{m} E_p$$

$$E_p = \frac{1}{2} \sum_{i=1}^{q} (g_i - h_i)^2$$

m个样本，q个输出

$$\Delta\omega_{ji} = \beta \cdot \delta_j \cdot h_i^{n-1}$$

$$\delta_j = \begin{cases} (g_j - h_j) \cdot h_j \cdot (1 - h_j) & \text{若j是输出节点} \\ (\sum_k \delta_k \cdot \omega_{kj}) \cdot h_j \cdot (1 - h_j) & \text{若j是隐层节点} \end{cases}$$

BP算法

1. 权值初始化。
2. 直到对任一个样本的所有输出节点k，

   $|g_k-h_k|<\varepsilon$；

   1) 选择一个样本，

   2) 从输入节点向前推，推出输出值；

   3) 如果对所有输出节点k, $|g_k-h_k|<\varepsilon$,则转2；

   4) 计算输出层连接权的 δ 值和 $\Delta\omega$，并反向传播，求出各隐层连接的 δ 值和 $\Delta\omega$ 。

能力

- 定理：

  对只有一个隐层的网络，在隐层中使用sigmoid函数，则上述过程可以以任意精度逼近任何平方可积函数。

BP算法的缺点

- 采用梯度法，可能陷入局部极小；
- 迭代算法，收敛速度慢；
- 隐层节点数目的确定没有理论指导，只能根据经验选择；