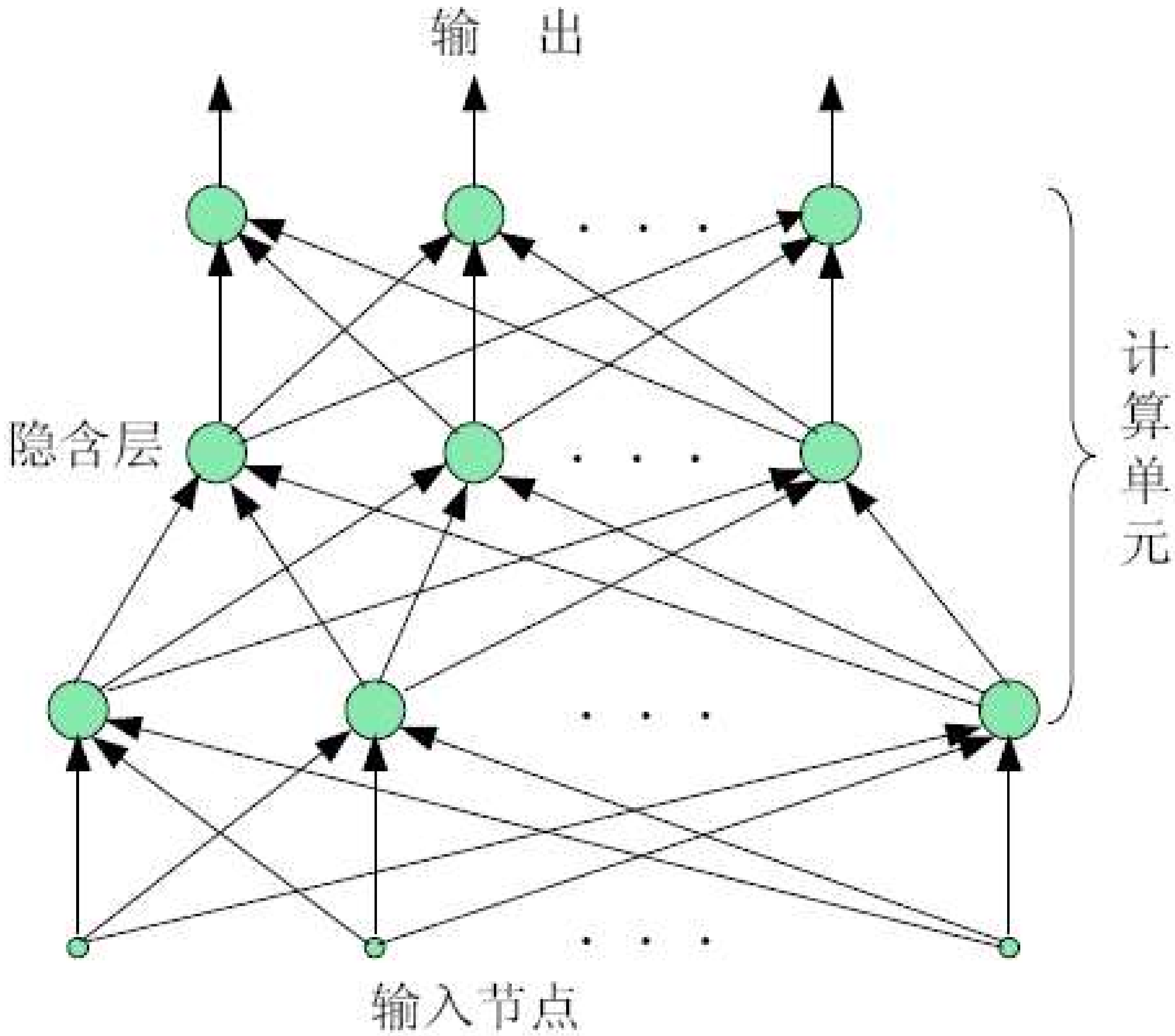
第六章 非线性规划

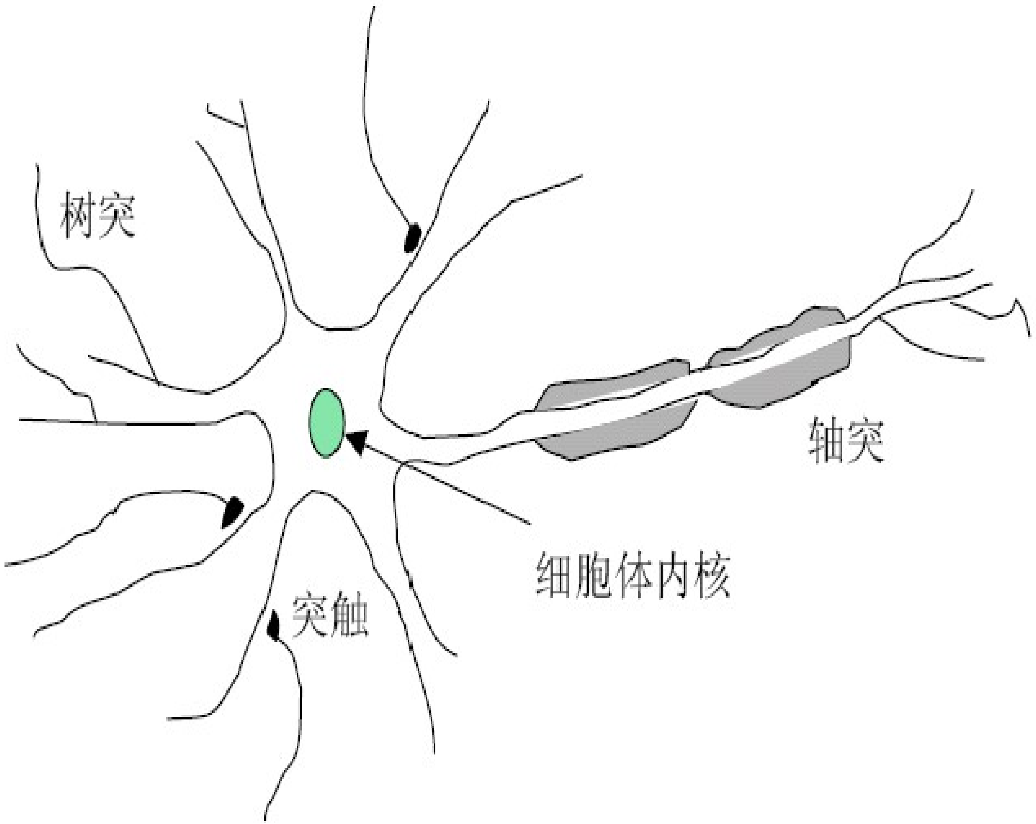
- 非线性规划与智能计算
  - **BP算法**
  - 遗传算法

神经网络

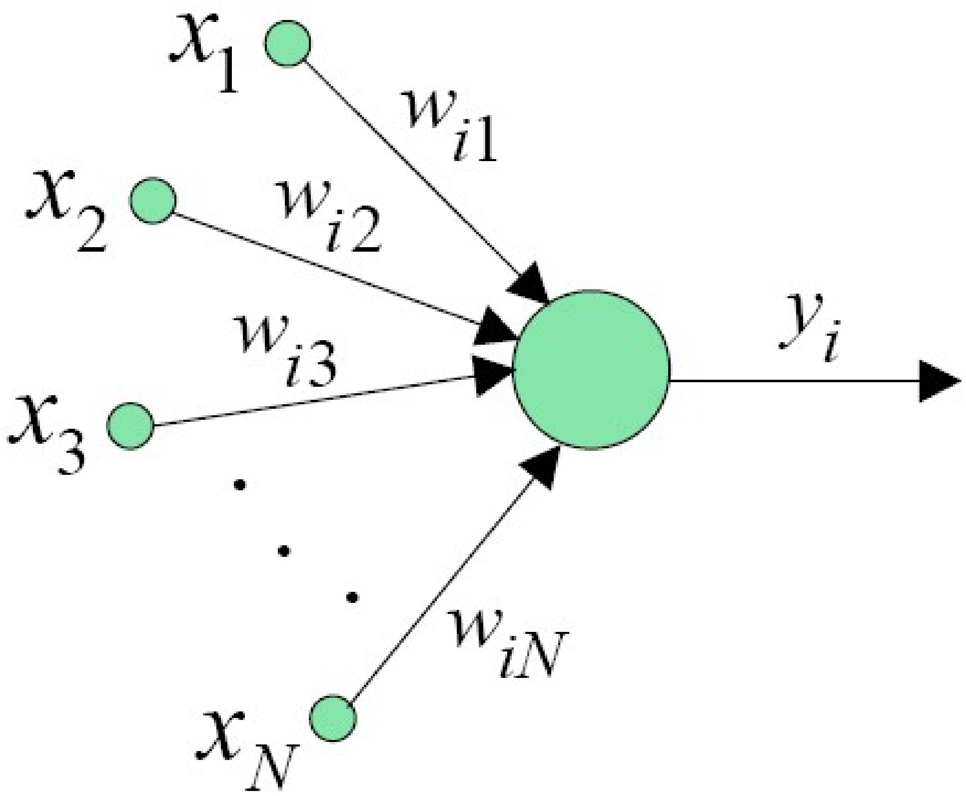
- 前馈全连接网络
- 参数
  - 层数
  - 每层神经元的个数
  - 神经元的激励函数



神经元的数学模型



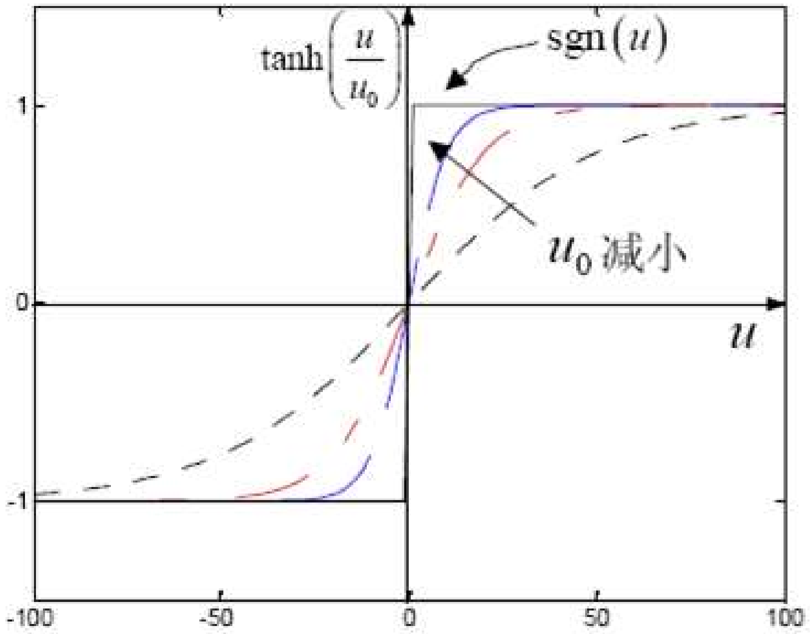
神经元结构



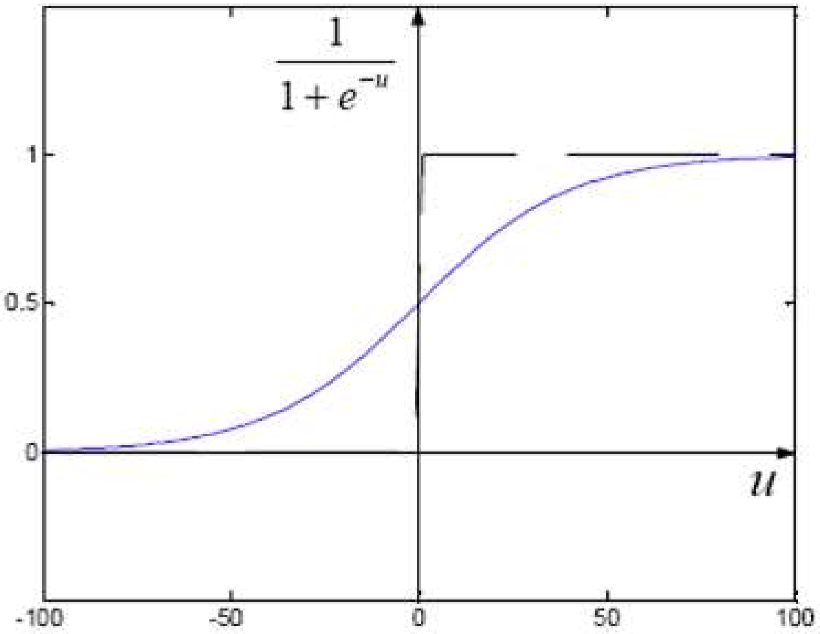
$$y_i = \Gamma(\sum_{j=1}^{N} w_{ij} x_j - \theta_i)$$

McCulloch-Pitts 模型

激励函数(Activation Function)



(a)



(b)

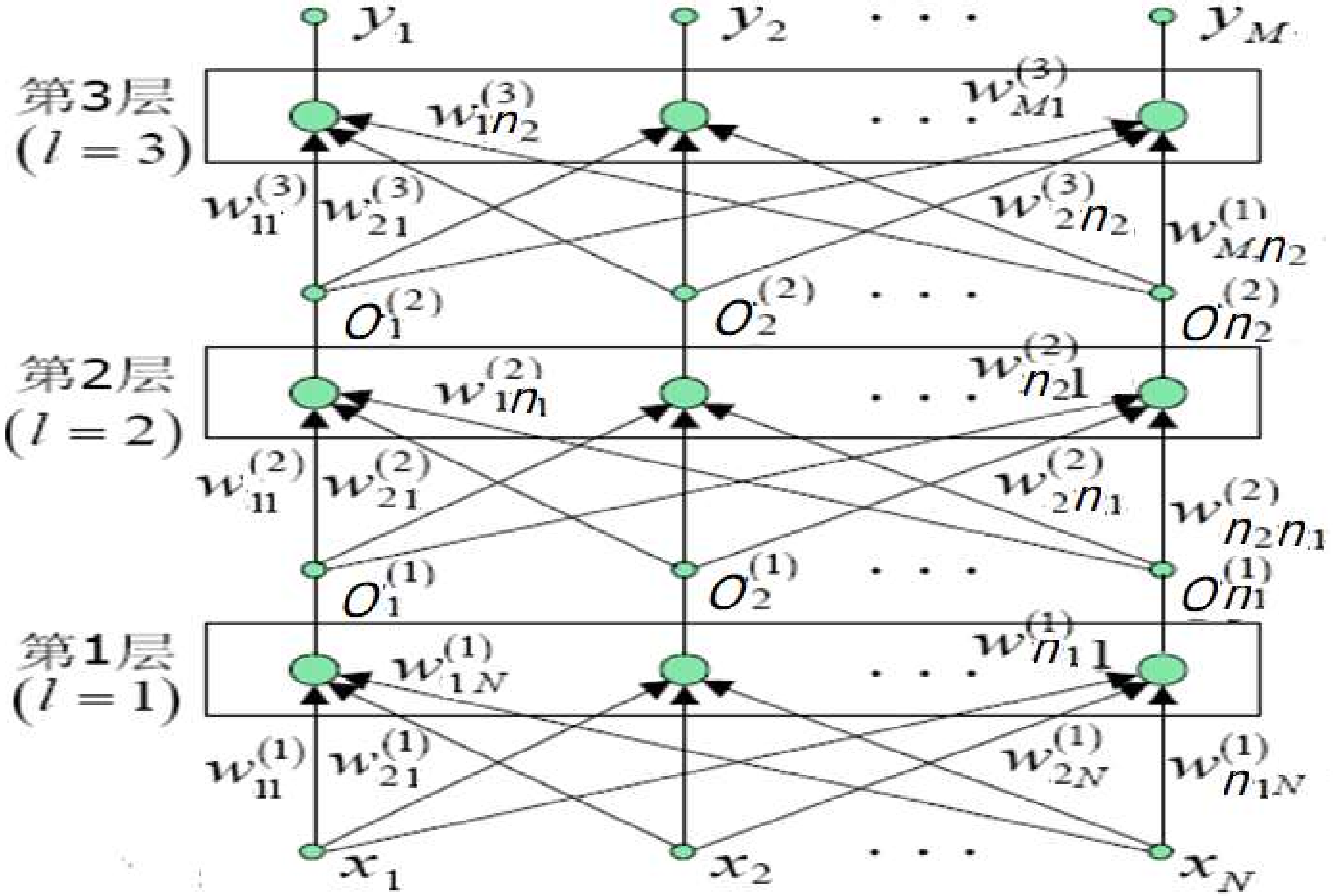
双曲正切函数

$$y(u) = \tanh(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}$$

Sigmoid函数

$$y(u) = \frac{1}{1+e^u}$$

具体结构



y_1
y_2
y_M
第3层
$(l=3)$
第2层
$(l=2)$
第1层
$(l=1)$
$w_{1n_2}^{(3)}$
$w_{M1}^{(3)}$
$w_{11}^{(3)}$
$w_{21}^{(3)}$
$w_{2n_2}^{(3)}$
$w_{Mn_2}^{(1)}$
$O_1^{(2)}$
$O_2^{(2)}$
$O_{n_2}^{(2)}$
$w_{1n_1}^{(2)}$
$w_{n_21}^{(2)}$
$w_{11}^{(2)}$
$w_{21}^{(2)}$
$w_{2n_1}^{(2)}$
$w_{n_2n_1}^{(2)}$
$O_1^{(1)}$
$O_2^{(1)}$
$O_{n_1}^{(1)}$
$w_{1N}^{(1)}$
$w_{n_11}^{(1)}$
$w_{11}^{(1)}$
$w_{21}^{(1)}$
$w_{2N}^{(1)}$
$w_{n_1N}^{(1)}$
x_1
x_2
x_N

网络模型

- 输入层

$$o_j^{(1)} = \Gamma_1(u_j^{(1)}) = \Gamma_1(\sum_{k=1}^{N} w_{jk}^{(1)} x_k) \qquad j = 1, 2 \ldots n_1$$

- 第l+1个隐含层

$$u_j^{(l+1)} = \sum_{k=1}^{n_l} w_{jk}^{(l+1)} o_k^{(l)}$$

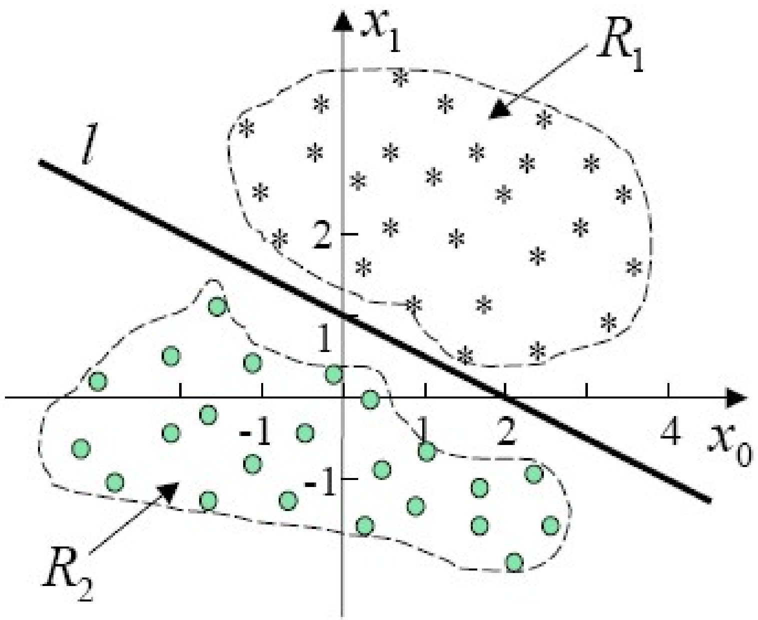
$$o_j^{(l+1)} = \Gamma_{l+1}(u_j^{(l+1)}) \qquad j = 1, 2 \ldots n_{l+1} \qquad l = 1, 2 \ldots L-1$$

- 输出层

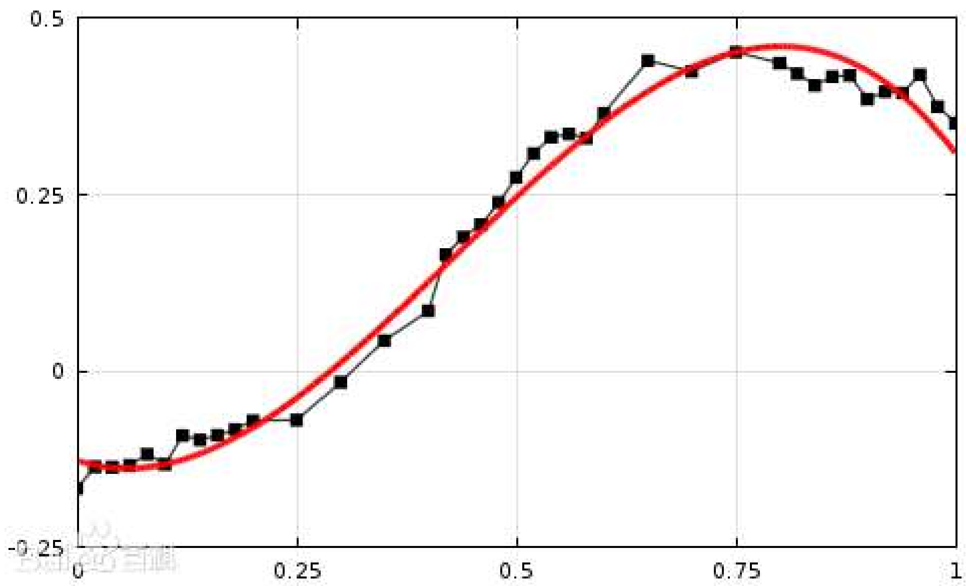
$$y_j = \Gamma_L(u_j^{(L)}) = \Gamma_L(\sum_{i=1}^{n_{L-1}} w_{ji}^{(L)} o_i^{(L-1)}) \qquad j = 1, 2 \ldots M$$

典型应用

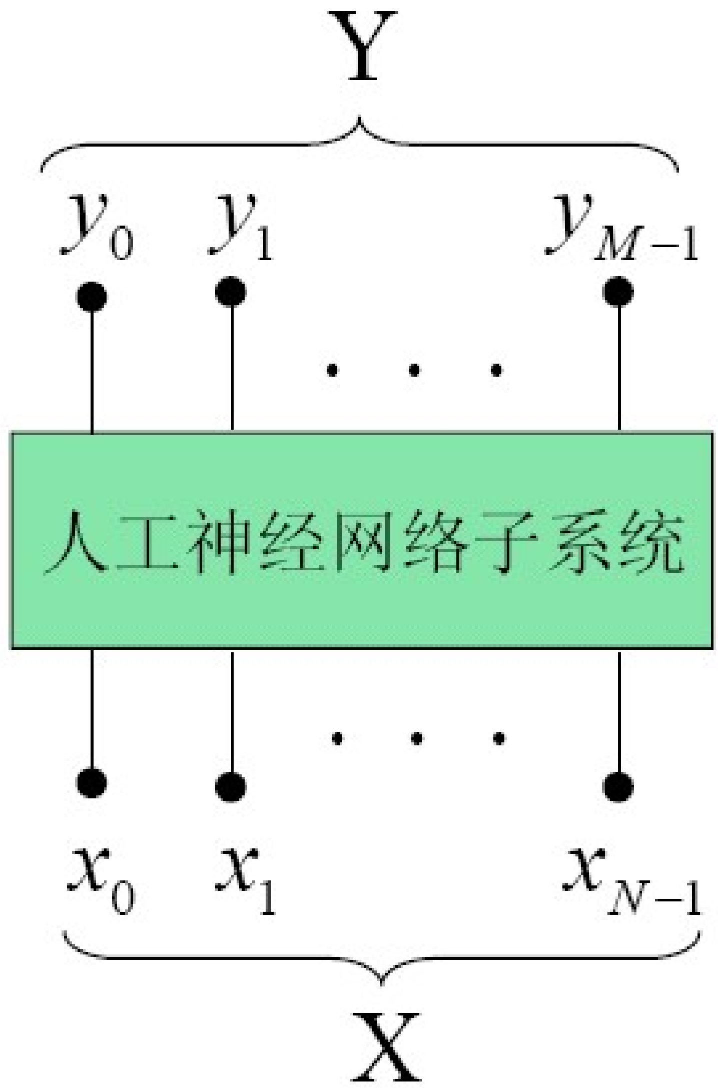
- 分类
- 拟合



二维空间中线性可分类的示例



有导师学习



- 样本集（**P**个样本）

$$(\boldsymbol{X}_p, \boldsymbol{T}_p) \qquad p = 1 : P$$

T$_p$为第***p***个样本的教师信息

- 学习目标

$$\min \sum_{p=1}^{P} E_p$$

$$E_p = \frac{1}{2}\left\| \boldsymbol{T}_p - \boldsymbol{Y}_p \right\|_2^2 = \frac{1}{2}\sum_{j=1}^{M}(t_{pj} - y_{pj})^2$$

优化模型

$$\min \frac{1}{2}\sum_{p=1}^{P}\sum_{j=1}^{M}(t_{pj}-y_{pj})^2$$

s.t.

$$o_{pj}^{(1)}=\Gamma_1\left(\sum_{k=1}^{N} w_{jk}^{(1)} x_{pk}\right) \qquad j=1,2\ldots n_1$$

$$o_{pj}^{(l+1)}=\Gamma_{l+1}\left(\sum_{k=1}^{n_l} w_{jk}^{(l+1)} o_{pk}^{(l)}\right) \qquad j=1,2\ldots n_{l+1} \qquad l=1,2\ldots L-1$$

$$y_{pj}=\Gamma_L\left(\sum_{i=1}^{n_{L-1}} w_{ji}^{(L)} o_{pi}^{(L-1)}\right) \qquad j=1,2\ldots M$$

$$p=1,2\ldots P$$

问题：谁是规划变量？

无约束优化问题

$$\min_{\boldsymbol{W}} \sum_{p=1}^{P} E_p = \frac{1}{2} \sum_{p=1}^{P} \left\| \boldsymbol{T}_p - NN(\boldsymbol{W}, \boldsymbol{X}_p) \right\|_2^2$$

$$\boldsymbol{W} = \{w_{ji}^{(l)}\} \qquad j = 1, 2 \ldots n_l \qquad i = 1, 2 \ldots n_{l-1} \quad l = 1, 2 \ldots L$$

- 梯度下降法

$$w_{ji}^{(l)}(k+1) = w_{ji}^{(l)}(k) - \eta \sum_{p=1}^{P} \frac{\partial E_p}{\partial w_{ji}^{(l)}}$$

梯度计算

$$\frac{\partial E_p}{\partial w_{ji}^{(l)}} = \frac{\partial E_p}{\partial u_{pj}^{(l)}} \frac{\partial u_{pj}^{(l)}}{\partial w_{ji}^{(l)}} \qquad l = 1, 2, \ldots, L$$

$$\frac{\partial u_{pj}^{(l)}}{\partial w_{ji}^{(l)}} = o_{pi}^{(l-1)} \qquad u_j^{(l+1)} = \sum_{i=1}^{n_l} w_{ji}^{(l+1)} o_i^{(l)}$$

$$\delta_{pj}^{(l)} \triangleq \frac{\partial E_p}{\partial u_{pj}^{(l)}}$$ 广义误差

$$u_j^{(l+1)} = \sum_{i=1}^{n_l} w_{ji}^{(l+1)} o_i^{(l)}$$

反向传播(Back Propagation)

- 输出层

$$\delta_{pj}^{(L)} = \frac{\partial E_p}{\partial u_{pj}^{(L)}} = \frac{\partial E_p}{\partial y_{pj}} \frac{\partial y_{pj}}{\partial u_{pj}^{(L)}} = -(t_{pj} - y_{pj})\Gamma_L'(u_{pj}^{(L)})$$

- 隐含层

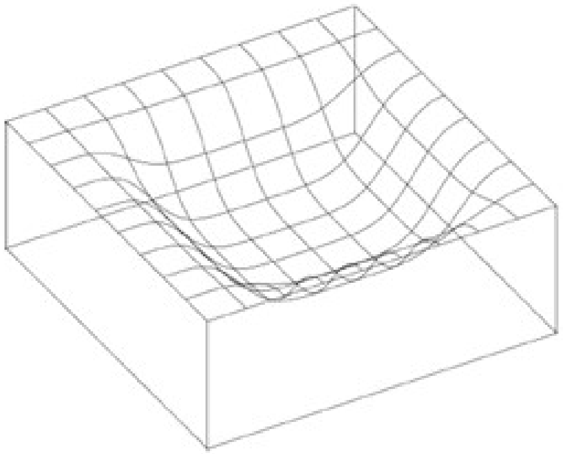
$$\delta_{pj}^{(l)} = \frac{\partial E_p}{\partial u_{pj}^{(l)}} = \frac{\partial E_p}{\partial o_{pj}^{(l)}} \frac{\partial o_{pj}^{(l)}}{\partial u_{pj}^{(l)}} = \left(\sum_{k=1}^{n_{l+1}} \frac{\partial E_p}{\partial u_{pk}^{(l+1)}} \frac{\partial u_{pk}^{(l+1)}}{\partial o_{pj}^{(l)}} \right) \Gamma_l'(u_{pj}^{(l)})$$

$$= \sum_{k=1}^{n_{l+1}} (\delta_{pk}^{(l+1)} \cdot w_{kj}^{(l+1)}) \Gamma_l'(u_{pj}^{(l)})$$

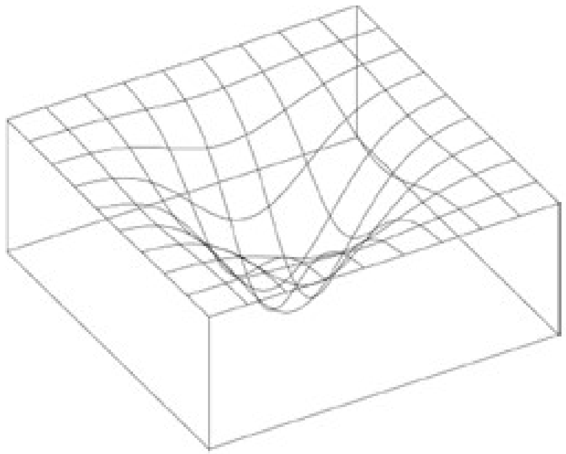
$$l = 1, 2, \ldots, L-1$$

BP学习的挑战

- 维度高，需研究提高训练效率。
- 目标函数曲面复杂，存在多种类型的驻点。
- 不是纯粹优化问题，可能过拟合，影响泛化。



宽而浅的局部极小值



尖而深的全局极小值

问题：哪种极小值比较好？

训练策略

- 批量梯度下降法(**Batch Gradient Decent, BGD)**:
  所有**P**个样本同时训练。
  精度高，但容易过拟合；速度慢，不适合大批量训练。
- 随机梯度下降法（**Stochastic Gradient Decent, SGD**）：
  每次只使用**1**个随机选出的样本，也称为增量梯度下降法。
  目标函数严格凸时，多轮训练可保证期望意义下的收敛性。
- 小批量梯度下降法（**mini-Batch Gradient Decent，MBGD**:
  将**P**个样本分为多组小批量学习。
  性能折中，是大规模网络学习的主流算法。

$$\Delta_p w_{ji}^{(l)} = -\eta \frac{\partial E_p}{\partial w_{ji}^{(l)}} = \eta \delta_{pj}^{(l)} o_{pi}^{(l-1)}$$

梯度校正

- 梯度振荡：
  - 对策：增加惯量矩（**Momentum**)：减小振荡，加速学习

$$\Delta w_{ji}^{(l)}(k) = \eta \delta_{pj}^{(l)} o_{pi}^{(l-1)} + \alpha \Delta w_{ji}^{(l)}(k-1)$$

- 梯度消失和梯度爆炸：随着层数增加，反向传播出现梯度过小或过大的现象，导致学习缓慢或网络不稳定
  - 对策：改变激励函数（**ReLU**）、权值正则化（损失函数中增加权值大小的惩罚项）、梯度截断（设置梯度阈值）、数据归一化…

$$\delta_{pj}^{(l)} = -\frac{\partial E_p}{\partial u_{pj}^{(l)}} = \sum_k (\delta_{pk}^{(l+1)} \cdot w_{kj}^{(l+1)}) \cdot \Gamma_l'(u_{pj}^{(l)}) \qquad o_j^{(l-1)} = \Gamma_{l-1}(u_j^{(l-1)})$$

自适应学习率

- **AdaGrad（Adaptive Gradient）**

  目的：自适应调整学习率，适合凸函数下的寻优

$$\Delta \boldsymbol{w}(k) = -\frac{\eta}{\sqrt{\boldsymbol{s}_k + \varepsilon}} \boldsymbol{g}_k \qquad \boldsymbol{g}_k = \frac{1}{P}\sum_{p=1}^{P}\frac{\partial E_p}{\partial \boldsymbol{w}} \qquad \boldsymbol{s}_k = \boldsymbol{s}_{k-1} + \boldsymbol{g}_k \odot \boldsymbol{g}_k$$

$$\varepsilon = 10^{-7} \qquad n_0 = 0$$

累积平方梯度

- **RMSProp (Root Mean Square propagation)**

  目的：防止**Adagrad**算法训练后期步长过小，适合非凸情形

$$\boldsymbol{s}_k = \rho \boldsymbol{s}_{k-1} + (1-\rho)\boldsymbol{g}_k \odot \boldsymbol{g}_k \qquad \rho = 0.9$$

梯度二阶矩的指数加权平均估计

- **Adam: Momentum+RMSProp +偏差修正**

  目的：修正指数加权平均开始阶段值偏小的情况 $\boldsymbol{s}_k' = \dfrac{\boldsymbol{s}_k}{1-\rho^k}$

第六章 非线性规划

- 非线性规划与智能计算
  - BP算法
  - 遗传算法

智能优化算法

- 美国的**J. Holland**教授于**1975**年提出

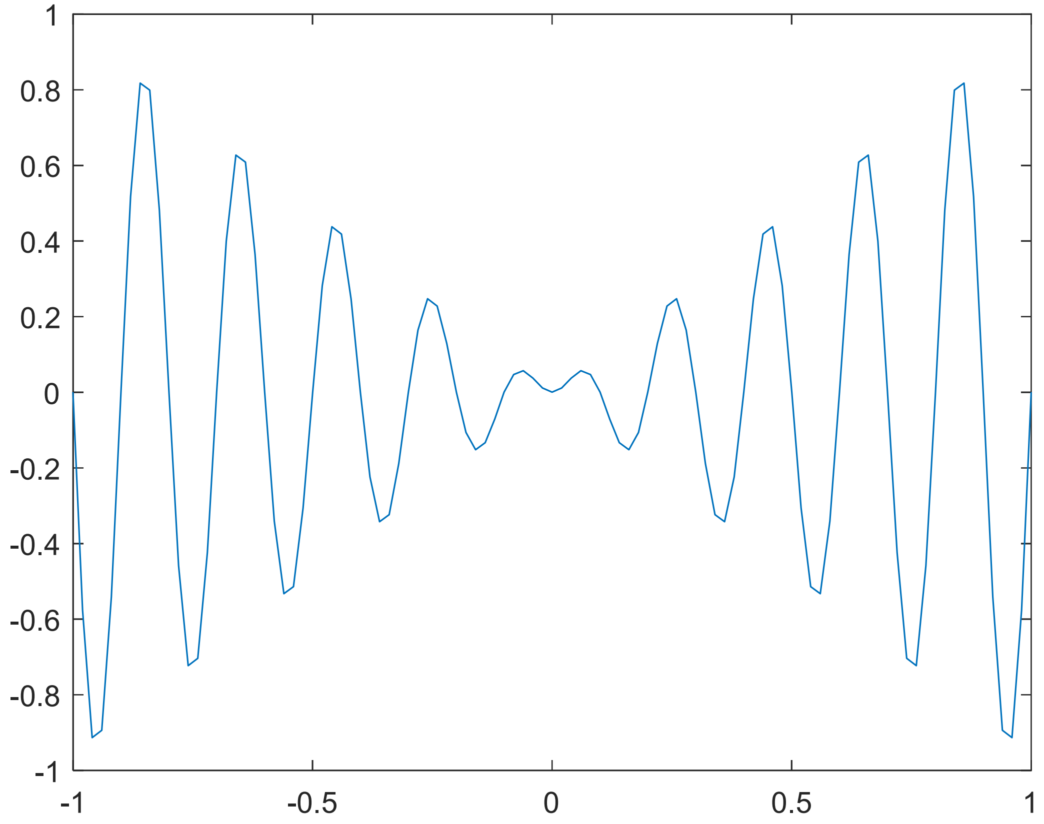
特点：

1、种群搜索（多点并行搜索）

2、随机化搜索（模拟染色体运作）

目标函数

$$\min f(x) = x \cdot \sin(10\pi \cdot x)$$



基本概念

■ 染色体$\boldsymbol{x}$：二进制编码的随机搜索样本点。

$$x = 0.637 \Leftrightarrow 1001111101$$

基因

■ 种群：所有个体（染色体）的集合$\{\boldsymbol{x_i}\}_{i=1,\ldots,N}$

■ 适应度函数：个体的目标函数值$\boldsymbol{f(x_i)}$。

种群迭代

- 复制：适者生存

$$P_i = f(x_i) / \sum_{i=1}^{N} f(x_i)$$ 复制概率

- 交叉：

交叉点

$$\begin{array}{ccc} 00000|01110 & \rightarrow & 00000|00000 \\ 11100|00000 & & 11100|01110 \end{array}$$

- 变异

变异点

$$0000\underline{0}01110 \rightarrow 0000\underline{1}01110$$

Exploration and Exploitation

- 引入扰动的概率选择
  - $P_{crossover} = 0.4\text{--}0.9$，
  - $P_{mutation} = 0.001\text{-}0.01$

- 改进算法
  - 保护已寻得的最好解

收敛性分析

- 理论上可收敛到全局最优解。
  - 模式定理
  - **Markov**链分析

◆实际当适应度函数值的变化很小或达到最大种群迭代次数时终止。

问题：终止时如何判断解的质量？

类似的算法

- 蚁群算法（**Ant Colony Optimization，ACO**）
- 粒子群算法（**Particle Swarm Optimization，PSO**）
- 免疫算法（**Immune Algorithm，IA**）
- 。。。

■ 本质：启发式并行随机搜索算法。