

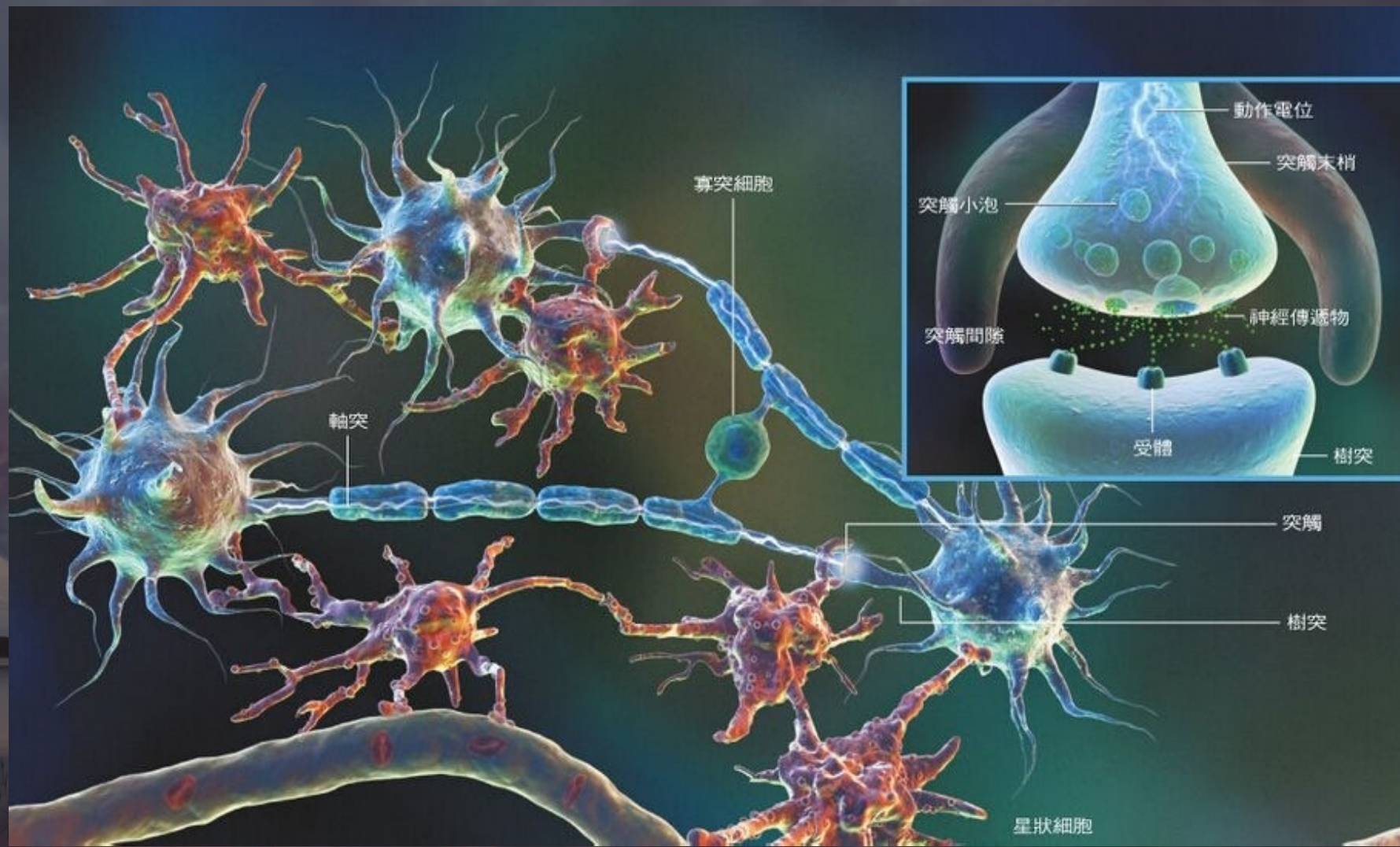
目录

- 1 什么是神经网络
- 2 BP(误差逆传播)神经网络
- 3 遗传算法原理
- 4 遗传—BP神经网络相结合
- 5 仿真

1 什么是神经网络

神经元：生物学上的神经元

人工神经元是对生物神经元的简化和模拟。生物神经元由细胞体、树突和轴突这三部分组成，如图所示。树突是细胞的输入端，通过细胞间连接的节点“突触”接触四周细胞传出的神经冲动，轴突相当于细胞的输出端，其端部的众多神经末梢为信号的输出端子，用于传出神经冲动。生物神经元具有兴奋和抑制两种工作状态。当传入的神经冲动，使细胞膜电位升高到阈值(约为40mV)时，细胞进入兴奋状态，产生神经冲动，由轴突输出。相反，若传入的神经冲动，使细胞膜电位降低到低于阈时，细胞进入抑制状态，没有神经冲动输出。

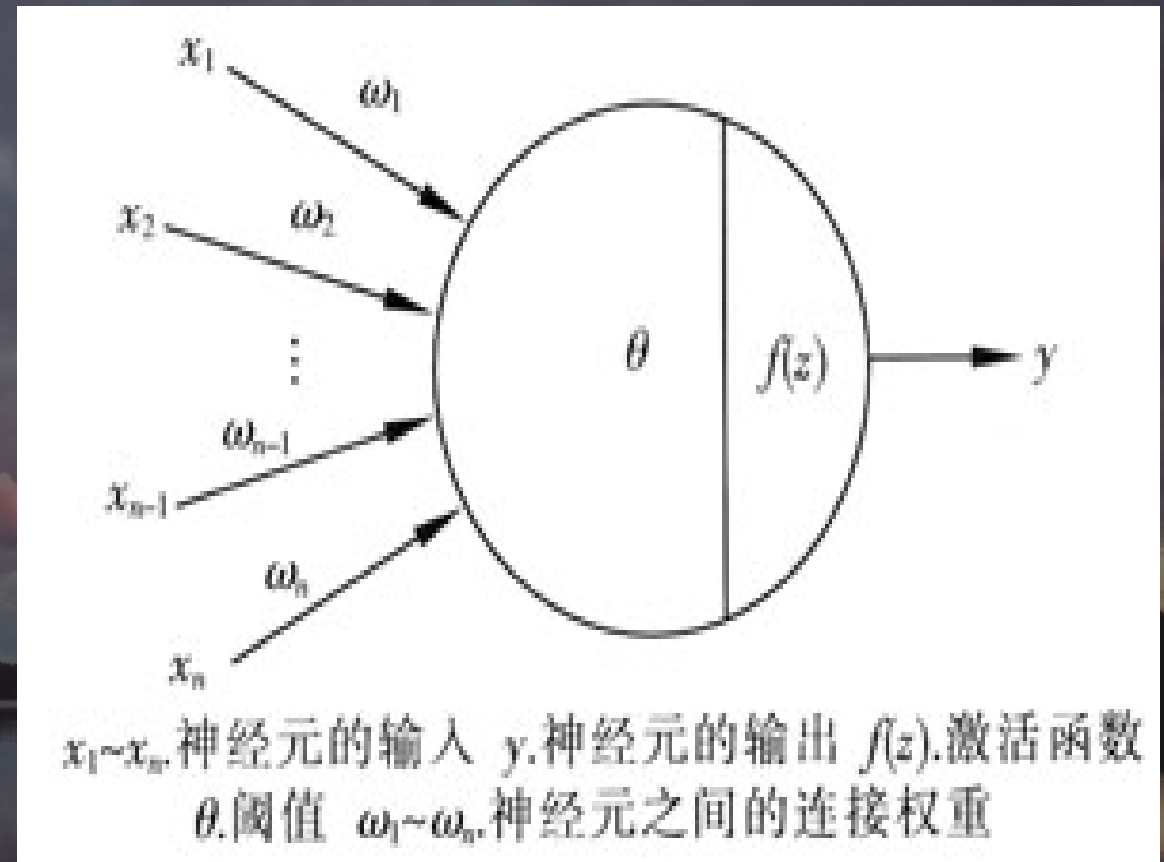


1 什么是神经网络

神经元模型：为了模拟生物神经元，一个简化的人工神经元如图所示。

它有三个基本要素：

1. 一组连接权(对应于生物神经元的突触)，连接强度由各连接上的权值表示，权值为正表示激励，为负值表示抑制。
2. 一个求和单元，用于求取各输入信息的加权和(线性组合)。
3. 一个非线性激励函数，起非线性映射作用并限制神经元输出幅度在一定的范围之内。此外还有一个阈值。阈值也被看作是一个输入分量，也就是阈值也是一个权值。在网络的设计中，偏差起着重要的作用，它使得激活函数的图形可以左右移动而增加解决问题的可能性。



a 什么是神经网络

神经元模型：数学表达

图给出了第j个基本BP神经元（节点），它只模仿了生物神经元所具有的三个最基本也是最重要的功能：加权、求和与转移。其中 $x_1, x_2 \dots x_i \dots x_n$ 分别代表来自神经元1, 2...i...n的输入； $w_{j1}, w_{j2} \dots w_{ji} \dots w_{jn}$ 则分别表示神经元1, 2...i...n与第j个神经元的连接强度，即权值； b_j 为阈值； $f(\cdot)$ 为传递函数； y_j 为第j个神经元的输出。第j个神经元的净输入值为：

$$s_j = \sum_{i=1}^n w_{ji} * x_i + b_j$$

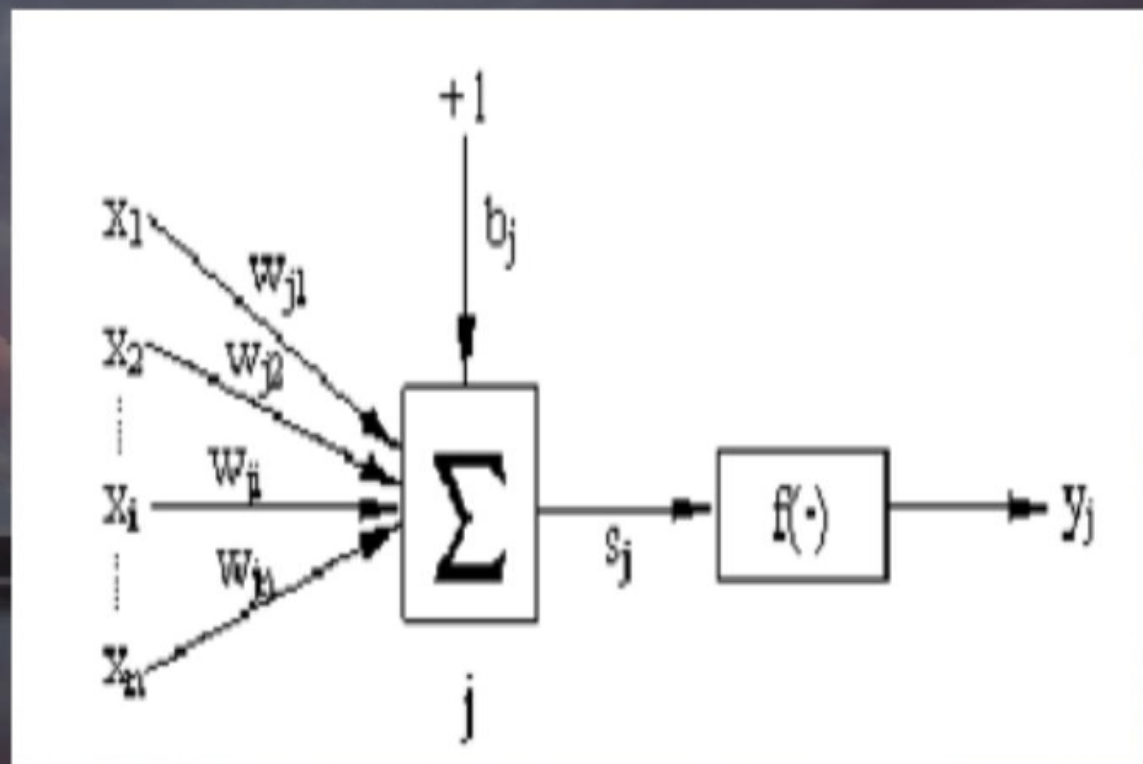
$$\text{设 } w_j = [w_{j0} \ w_{j1} \ w_{j2} \ \dots \ w_{ji} \ \dots \ w_{jn}] \quad x = [x_0 \ x_1 \ x_2 \ \dots \ x_i \ \dots \ x_n]^T$$
$$x_0 = +1 \quad w_j = b_j$$

于是节点j的净输入可表示为：

$$\text{则 } S_j = w_j * x + b_j$$

净输入通过传递（Transfer Function） $f(\cdot)$ 后，便得到第j个神经元的输出：

$$y_j = f(s_j) = f(w_j * x + b_j)$$



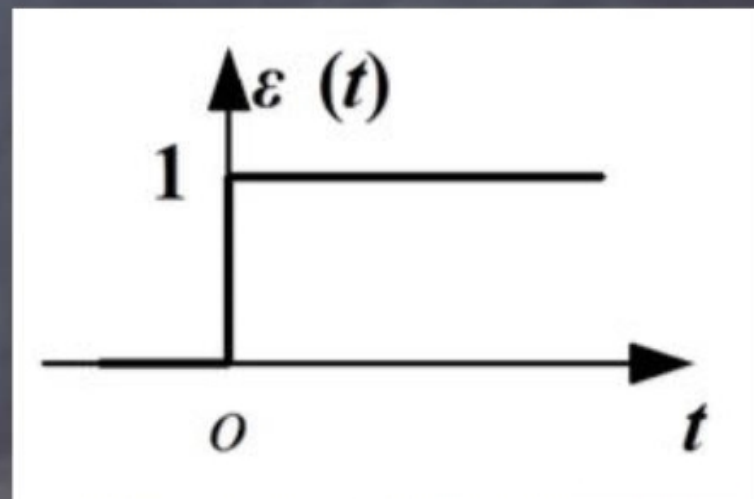
a 什么是神经网络

神经元模型：传递函数/激励函数

阶跃函数

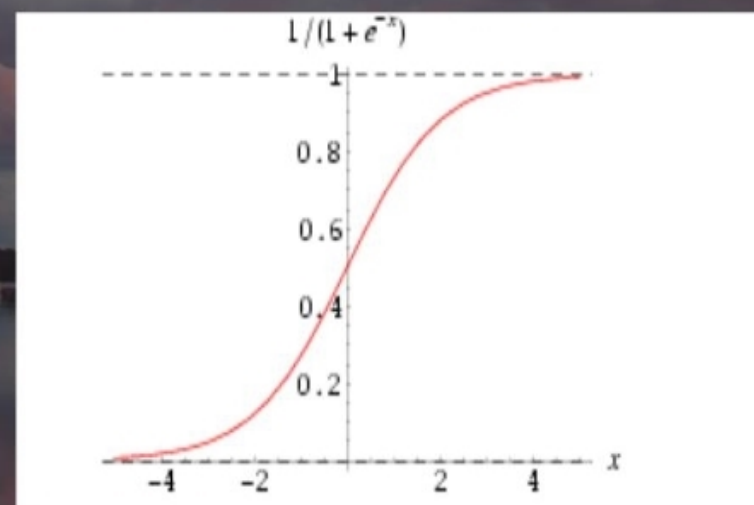
$$\text{sgn}(y) = 1 \quad x \geq 0$$

$$\text{sgn}(y) = 0 \quad x \leq 0$$



Sigmoid (x) 函数

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



BP算法由数据流的前向计算（正向传播）和误差信号的反向传播两个过程构成。正向传播时，传播方向为输入层→隐层→输出层，每层神经元的状态只影响下一层神经元。若在输出层得不到期望的输出，则转向误差信号的反向传播流程。通过这两个过程的交替进行，在权向量空间执行误差函数梯度下降策略，动态迭代搜索一组权向量，使网络误差函数达到最小值，从而完成信息提取和记忆过程。

BP神经网络原理：正向传播

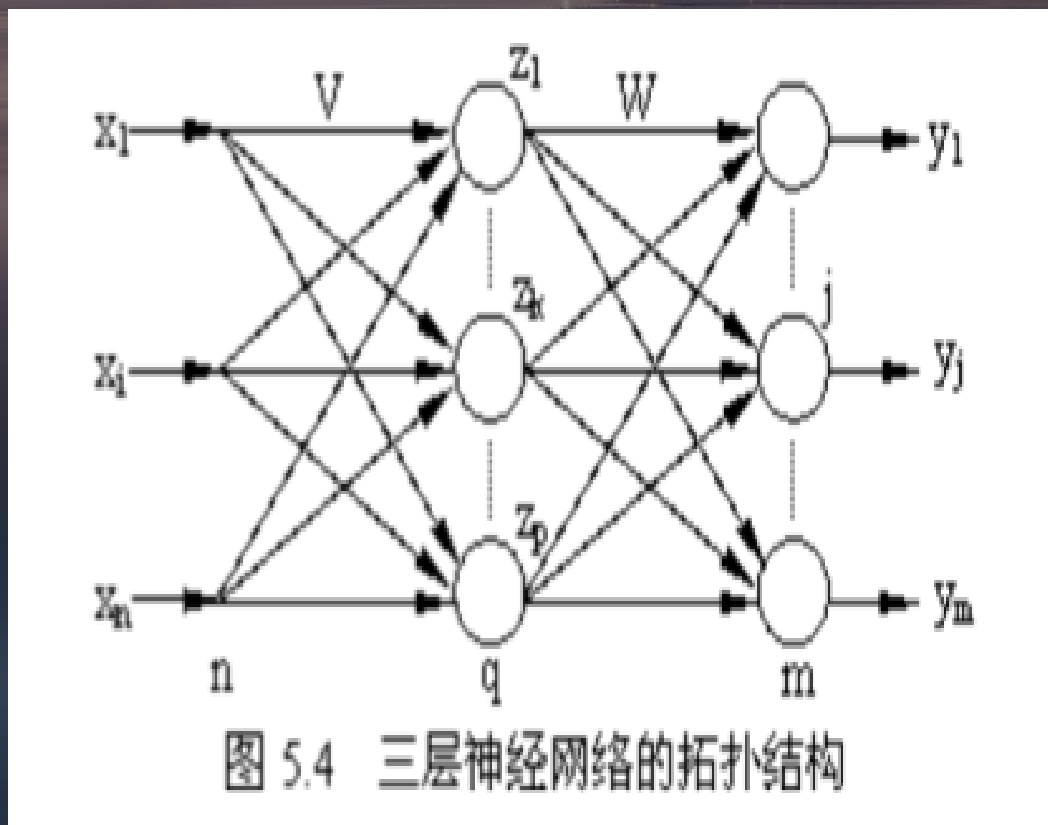
设 BP网络的输入层有 n 个节点，隐层有 q 个节点，输出层有 m 个节点，输入层与隐层之间的权值为 v ，隐层与输出层之间的权值为 w ，如图5.4所示。隐层的传递函数为 $f_1(\cdot)$ ，输出层的传递函数为 $f_2(\cdot)$ ，则隐层节点的输出为（将阈值写入求和项中）

$$z_k = f_1 \left(\sum_{i=0}^n x_i * v_{ki} \right) \quad k = 1, 2 \dots q$$

输出层节点的输出为

$$y_j = f_2 \left(\sum_{k=0}^q z_k * w_{jk} \right) \quad j = 1, 2 \dots m$$

至此B-P网络就完成了 n 维空间向量对 m 维空间的近似映射。



a) 定义误差函数

输入个 p 学习样本，用 $x^1, x^2 \dots x^h \dots x^p$ 来表示。第 h 个样本输入到网络后得到输出 y_j^h ($j=1,2\dots m$)。采用平方型误差函数，于是得到第 h 个样本的误差 E_h ：

$$E_h = \frac{1}{2} \sum_{j=1}^m (t_j^h - y_j^h)^2$$

式中 t_j^h 为期望输出。对于 p 个样本，全局误差为：

$$E = \frac{1}{2} \sum_{h=1}^p \sum_{j=1}^m (t_j^h - y_j^h)^2 = \sum_{h=1}^p E_h$$

BP神经网络原理：反向传播

b) 输出层权值的变化

采用累计误差BP算法调整 w_{jk} ，使全局误差 E 变小，即

$$\Delta W_{JK} = -\eta \frac{\partial E}{\partial W_{JK}} = \sum_{h=1}^p \eta \frac{\partial E_h}{\partial W_{JK}} \quad (\eta \text{ 学习效率})$$

于是最终输出层各神经元的权值调整公式为：

$$\Delta W_{JK} = \sum_{h=1}^p \sum_{j=1}^m \eta (t_j^h - y_j^h) f_2'(S_J) w_{JK} f_1'(S_K) x_i$$

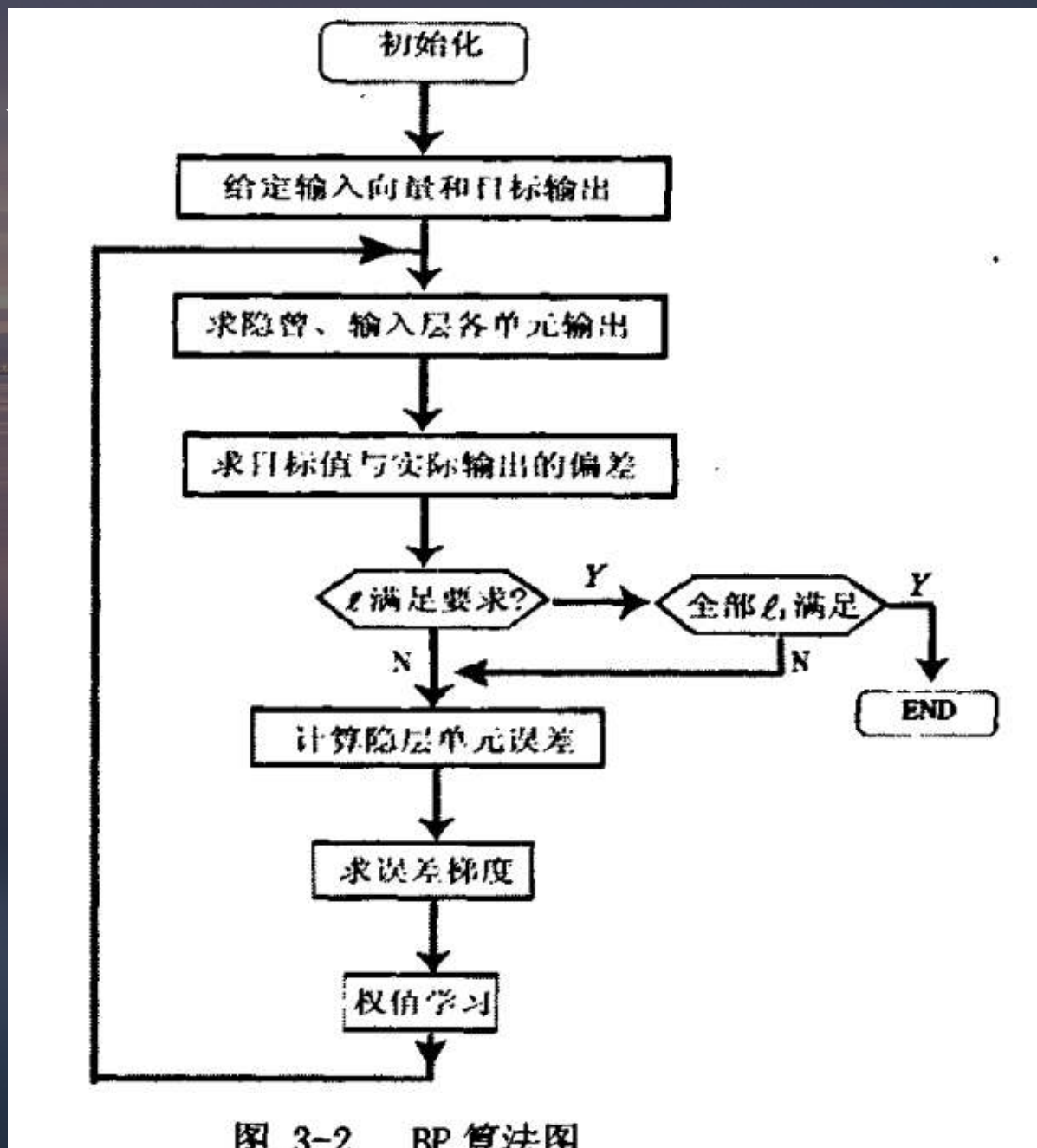


图 3-2 BP 算法图

BP神经网络存在的问题

BP算法理论具有依据可靠、推导过程严谨、精度较高、通用性较好等优点，但标准BP算法存在以下缺点：

1. 收敛速度缓慢；
2. 容易陷入局部极小值；
3. 难以确定隐层数和隐层节点个数。

在实际应用中，BP算法很难胜任，因此出现了很多改进算法。

自适应调整学习速率：调整的基本指导思想是：在学习收敛的情况下，增大 η ，以缩短学习时间；当 η 偏大致使不能收敛时，要及时减小 η ，直到收敛为止。

利用动量法改进BP算法：动量法权值调整算法的具体做法是：将上一次权值调整量的一部分迭加到按本次误差计算所得的权值调整量上，作为本次的实际权值调整量。

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/208135036104006124>