

对含噪声数据的一种鲁棒学习算法^{*, 1)}

李杰星 章云 符曦

(广东工业大学自动化研究所, 广州 510090)

A Robust Learning Algorithm for Noise Data

Li Jie-xing Zang Yun Fu Xi

(Automation Research Institute, Guangdong University of Technology, Guangzhou 510090)

Abstract

Allowing for the limitations of LS energy function used in BP algorithm, this paper proposes a robust learning algorithm based on the study of how clustering puts down random noise's effects and the consideration of intensified training for high-quality examples. Some simulation results demonstrate that the robust algorithm is clearly superior to BP algorithm in anti-disturbance and astringency.

Keywords: noise, neural network, energy function, clustering, robustness, astringency

§1. 引言

对一个难于建立数学模型的复杂控制对象,以神经网络和模糊控制为代表的智能控制方法比传统控制方法具有很大的优越性.然而,在一个实际控制系统中,输入神经网络的外界测量和传输数据不可避免地受到或大或小的噪声干扰(如测量误差,元件的热噪声、电噪声,电磁干扰,甚至人为的过失误差等),给神经网络的学习带来困难.例如传统的BP算法,当输入-输出数据样本受到噪声干扰时,会导致网络输出曲线(面)朝着受到噪声污染的样本曲线(面)靠近,而得不到真实的曲线(面);同时,受到噪声污染的样本曲线(面)在某些局部常常变得更为凹凸不平,加大了学习难度,以致学习的收敛速度慢且易陷入局部极小.

为解决这些问题,文献[1]提出了“相对学习强度”、“误差中心曲面”等概念,由此构成新的鲁棒能量函数.它在理论上给出了算法鲁棒性的一个可行方向,但其实现方法却有所不足.

* 1998年12月11日收到.

1) 广东省重点学科项目(970003)及广东省自然科学基金项目(960101).

对于幅值小, 均值为零的随机噪声, 选择恰当的标准和方法对样本进行聚类分析, 形成聚类, 能较满意地减轻噪声的影响; 而对于幅值较大、均值不为零的噪声 (如过失误差), 本文引入鲁棒因子, 其数值能够根据所选的样本性质 (如与附近样本的偏离程度) 而变化, 把它加到能量函数里面以控制学习进程, 达到减少噪声干扰的目的. 基于上述两条思路, 本文提出了一种鲁棒学习算法, 并进行了系统仿真和性能比较.

§2. LS 能量函数的局限性

设网络学习样本集为

$$\mathfrak{R} = \{(x_p, y_p) \mid y_p = f(x_p) + v_p, x_p \in R^n, y_p \in R^m, p = 1, \dots, P\}, \quad (1)$$

其中 f 为系统真实非线性函数, v_p 为噪声.

设网络输出 \hat{y}_p , 真实输出 \tilde{y}_p , 拟合偏差 e_p , 真实偏差 \tilde{e}_p 分别为

$$\hat{y}_p = N[x_p, w], \quad \tilde{y}_p = f(x_p), \quad e_p = y_p - \hat{y}_p, \quad \tilde{e}_p = \tilde{y}_p - \hat{y}_p, \quad (2)$$

式中 N 为网络的非线性函数, w 为权值. LS 能量函数为

$$E_{LS} = \sum_{p=1}^P \phi(e_p) = \sum_{p=1}^P e_p^2. \quad (3)$$

这样, 如果按传统的 BP 算法进行学习, 便会产生如下问题:

1. 由式 (3) 知道, LS 能量函数指导整个学习过程的最终目标是使所有样本的拟合误差 e_p 机会均衡地趋向零, 即网络输出逼近于每个受噪声污染的实际输出, 而不是真实输出. 这正是 BP 算法在实际中出现的迭代次数越多, 目标函数值越小, 反而外推精度 (容错能力) 越差的重要原因.

2. 由于噪声等因素的影响, 在绝大多数情况下, 增加了待学习曲线 (面) 的不平坦性, 加大了学习难度, 使收敛慢, 易于陷入局部极小.

因此, 就原理而言, 最小方差型能量函数是不具有抗噪声能力的.

§3. 鲁棒因子的提出

为了使多层前馈网络具有抵抗噪声干扰的鲁棒性, 我们必须舍弃长期使用的最小方差能量函数, 构造新的鲁棒能量函数, 形成鲁棒算法. 构造鲁棒能量函数的基本思想是: 不再要求网络逼近每一个样本点, 转而要求网络有足够的“力量”逼近于或优先逼近于系统真实输出的估计的置信区间中心.

设鲁棒能量函数的结构为^[1]

$$E_R = \sum_{p=1}^P \phi(e_p, c_p), \quad (4)$$

其中 c_p 为在 x_p 附近对真实偏差 \tilde{e}_p 作统计估计所得的置信区间的中心值. $\phi(e_p, c_p)$ 要满足两个要求

$$\begin{cases} \frac{\partial \phi(e_p, c_p)}{\partial e_p} = e_p \cdot \psi(e_p, c_p), \\ \phi(0, c_p) = 0, \end{cases} \quad (5)$$

其中 $\psi(e_p, c_p) \geq 0$ 是关于 e_p 的非负可积函数, 且 $\max \psi(e_p, c_p) = \psi(c_p, c_p)$. 根据不同的情况, 可以通过不同的 $\psi(e_p, c_p)$ 来构造不同的 E_R .

由式 (5) 可知, $\psi(e_p, c_p) = \frac{\partial \phi(e_p, c_p)}{\partial e_p} / e_p$, 本文称之为样本 p 的鲁棒因子, 其意义可理解为样本 p 的相对学习强度. 定义强化学习域为

$$S(\delta) = \{e \mid \psi(e, c) > \delta\}, \quad (6)$$

式中, δ 为为强化学习参数.

在学习过程中, 鲁棒算法相对 BP 算法最大的改进在于

$$\text{BP 算法:} \quad \frac{\partial \phi(e_p)}{\partial e_p} = 2 \cdot e_p \quad (7)$$

$$\text{鲁棒算法:} \quad \frac{\partial \phi(e_p, c_p)}{\partial e_p} = e_p \cdot \psi(e_p, c_p) \quad (8)$$

即 BP 算法对所有样本的相对学习强度均为 2, 就是说无论样本受到噪声污染与否都一视同仁; 而鲁棒算法可以利用合适的 $\psi(e_p, c_p)$ 来控制学习过程, 即对某些被认为品质好 (受到噪声污染轻) 的样本, 通过加大其 $\psi(e_p, c_p)$, 强化它们的学习; 反之, 则弱化其学习. 鲁棒因子考虑了真实误差, 有可能通过合理设置它来使网络逼近系统的真实输出.

$\psi(e_p, c_p)$ 的选择多种多样, 下面以双指数核函数为例进行分析, 即

$$\psi(e_p, c_p) = 2 \cdot \exp(-\lambda_p |e_p - c_p|), \quad (9)$$

则有

$$\phi(e_p, c_p) = \frac{2}{\lambda_p^2} [e^{-\lambda_p |c_p|} - (1 + \bar{\lambda}_p e_p) e_p^{-\lambda_p |c_p - c_p|} + \lambda_p |c_p| + \bar{\lambda}_p c_p], \quad (10)$$

其中 λ_p 为大于零的坡度参数, $\bar{\lambda}_p = \text{sgn}(e_p - c_p) \cdot \lambda_p$. 从式 (9) 知, 当样本 p 受到较严重的噪声干扰时, $|e_p - c_p|$ 较大, 鲁棒因子较小, 学习率也小, 学习被弱化. 也就是说, 那些拟合偏差 e_p 接近真实偏差 \tilde{e}_p 的样本 (被认为是品质好的), 相对而言, 它们的学习被强化了.

§4. 鲁棒学习算法

使用鲁棒因子, 首先要确定 c_p 和 λ_p , 在 c_p 的确定过程中, 文献 [1] 做法的实质是:

1. 随意地将样本集的输入空间分成 k 个球域.
2. 在各个学习阶段, 分别统计球域内样本拟和偏差的均值 u_k 和方差.

3. 将样本偏差看作正态白噪声, 以 u_k 作为球域内样本真实偏差的估计值, 利用 t -分布和方差求上述估计的置信区间, 得到置信区间界 cl_k .

4. 由于一个样本可能属于几个不同的球域, 因此对某一个样本, 其 c_p 为所属球域 u_k 的均值, 令 cl_p 为所属球域 cl_k 的均值, 则 $\lambda_p = \frac{\ln(2/\delta)}{cl_p}$.

经研究, 发现上述做法有不足之处: 1) 球域的划分无规可循, 全凭经验, 有一定的盲目性; 2) 用 t -分布作区间估计仅适用于处理样本为正态分布量的情况^[2]. 一般地, 并不能保证噪声是正态(高斯)白噪声, 特别是噪声中包含过失偏差时, 更不可能为正态白噪声; 3) c_p, cl_p 分别为所属球域 u_k, cl_k 的均值(见步骤 4), 但由于球域的划分缺乏理论指导, 一个样本属于那几个球域没有规范的做法, 其结果带有随意性.

考虑到文献 [1] 计算方法在样本划分上的不足, 以及样本聚类对噪声一定的抵御能力, 本文引入聚类分析, 它能自适应地确定聚类的数目、中心和样本的归属. 同时, 在区间估计中使用适应性更广的标准正态分布(它对处理样本没有特别要求)^[2]. 具体步骤如下:

1. 对样本进行聚类分析, 形成聚类.

①算出所有样本相互之间的距离 d_{ij} ($i, j = 1 \sim n, n$ 为样本总数)

$$d_{ij} = \sqrt{\|x^i - x^j\|^2 + \|y^i - y^j\|^2}. \quad (11)$$

找出与样本 i 距离最相近的 t 个样本, 设这 t 个样本与样本 i 的距离是 $(d_i^1 - d_i^t)$, 求出这些距离的平均值

$$\bar{d}_i = \frac{1}{t} \sum_{p=1}^t d_i^p. \quad (12)$$

\bar{d}_i 称为样本 i 的样本区域密度, 它间接地表示样本 i 附近区域的样本密度; 算出所有样本之间距离的平均值

$$\bar{d} = \frac{2}{n(n-1)} \sum_{p=1}^{n-1} \sum_{q=p+1}^n d_{pq}. \quad (13)$$

②从第一个样本 (x^1, y^1) 开始, 令第一个样本为第一个聚类 (\bar{x}^1, \bar{y}^1) . 设

$$A^1 = \bar{x}^1, \quad B^1 = \bar{y}^1, \quad C^1 = 1 \quad (14)$$

设一个聚类半径初始值 r .

③当考虑到第 k 个样本 (x^k, y^k) 时, 假设已有 m 个聚类, 求出样本 k 与各个聚类之间的距离

$$\sqrt{\|x^k - \bar{x}^l\|^2 + \|y^k - \bar{y}^l\|^2}, \quad l = 1 \sim m. \quad (15)$$

设其中的最小值为 $\sqrt{\|x^k - \bar{x}^p\|^2 + \|y^k - \bar{y}^p\|^2}$, 则聚类 p 是样本 k 的最近邻聚类.

④令

$$r_k^* = \frac{r \cdot \bar{d}_k}{\bar{d}}, \quad (16)$$

如果

$$\sqrt{\|x^k - \bar{x}^p\|^2 + \|y^k - \bar{y}^p\|^2} > r_k^* \quad (17)$$

成立, 则样本 (x^k, y^k) 作为一个新的聚类, 有

$$A^{m+1} = \bar{x}^{m+1} = x^k, \quad B^{m+1} = \bar{y}^{m+1} = y^k, \quad C^{m+1} = 1. \quad (18)$$

否则, 把样本 k 归入聚类 p , 且有

$$A^p = A^p + x^k, \quad B^p = B^p + y^k, \quad C^p = C^p + 1. \quad (19)$$

⑤ 确定完样本对聚类的归属后, 令

$$\bar{x}^l = \frac{A^l}{C^l}, \quad \bar{y}^l = \frac{B^l}{C^l}, \quad (20)$$

式中 $l = 1 \sim m$, m 为聚类总数. 这些聚类由于对所属样本在输入、输出空间作了均化处理 (见 (18) 式), 因此能在一定的程度上消除噪声的干扰, 特别是幅值小, 均值为零的随机噪声.

2. 将 $\{(\bar{x}^l, \bar{y}^l) \mid l = 1 \sim m\}$ 作为附加样本与学习样本一起让神经网络学习, 分别得到 e_l 和 e_p .

3. 令

$$\begin{cases} u_l = \frac{1}{n_l} \sum_{i=1}^{n_l} e_i, \\ s_l^2 = \frac{1}{n_l - 1} \sum_{i=1}^{n_l} (e_i - u_l)^2, \end{cases} \quad (21)$$

其中 n_l 为聚类 l 所属样本的个数, i 为聚类 l 所属样本的序号.

4. 如果样本 p 属于聚类 l , 则 $c_p = e_l$, 此估计的 $(1 - \alpha)$ 置信区间界为 [2]

$$N_{\alpha/2} \cdot s_l / \sqrt{n_l}, \quad (22)$$

式中 $N_{\alpha/2}$ 是标准正态分布关于 $\alpha/2$ 的上侧分位数. 这个估计方法被估量的分布可以是任意的, 但样本数要求要大.

5. 令

$$L_l = N_{\alpha/2} \cdot s_l / \sqrt{n_l}, \quad (23)$$

让 c_p 的 $(1 - \alpha)$ 置信区间置于强化学习域内. 由于在式 (9) 的鲁棒因子下, δ 强化学习域为

$$S(\delta) = \left\{ e \mid |e - c| < \frac{1}{\lambda} \ln \left(\frac{2}{\delta} \right) \right\}, \quad (24)$$

所以如果样本 p 属于聚类 l , 则有

$$\lambda_p = \ln \left(\frac{2}{\delta} \right) / L_l. \quad (25)$$

求出鲁棒因子 $\psi(e_p, c_p)$ 后, 按通常的误差反响传播算法修正权值, 直到 E_R 达到最优.

当样本受到噪声污染越大, 其鲁棒因子越小、学习强度越低, 这个措施对大幅值噪声效果明显, 再结合聚类对小幅值噪声的作用, 就构成了一套完整的噪声处理体系.

§5. 仿真研究

仿真用的神经网络分成三层, 第一层为直通输入层, 第二层是隐层, 其神经元的非线性函数为 MATLAB 5.1 版提供的 Tansig 函数, 第三层是输出层, 非线性函数为 Purelin 函数. 所有样本学习一遍为迭代一次. 本文的各种仿真算法都采用了 MATLAB 5.1 版的 Levenberg-Marquardt 技术, 仿真用的计算机是 $6 \times 86 - 200$.

鲁棒因子取为两种, 第一种为式 (9) (由此构成鲁棒算法 1), 第二种 (鲁棒算法 2) 取为更简单的

$$\psi(e_p, c_p) = \frac{1}{1 + k|e_p - c_p|}, \quad (26)$$

式中 k 是一个常参数.

1. 三维曲面仿真

待仿真函数为

$$z = 10 \cdot x_p \cdot y_p \cdot e^{(-x_p^2 - y_p^2)} + v_p. \quad (27)$$

x_p, y_p 在 $[-2, 2]$ 内以随机取样, 样本数为 400, 噪声 v_p 为 $(-0.5, 0.5)$ 内均匀分布随机噪声. 为了检验网络是否真的学到真实函数, 即在学习样本外的检验样本输出是否与真实函数输出一致, 因此在检验时 x, y 在 $[-2, 2]$ 内以步长 0.1 取样, 检验样本是学习样本的 8 倍, 仿真结果如下:

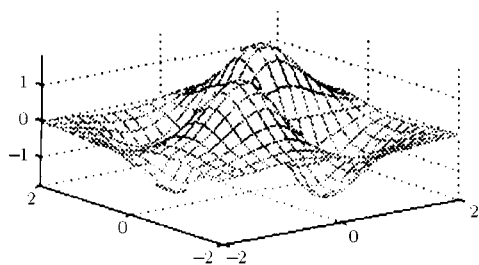


图 1 真实函数曲面

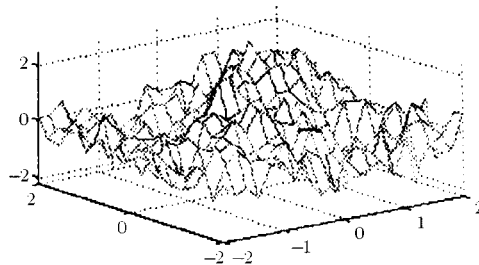


图 2 被噪声污染后的样本曲面

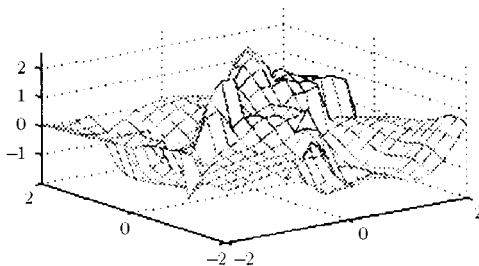


图 3 BP 算法的检验曲面

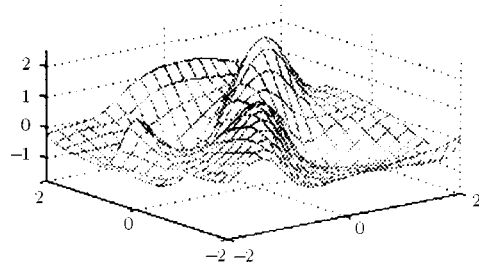


图 4 文献 [1] 鲁棒算法的检验曲面

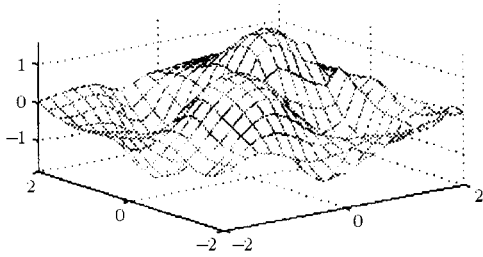


图 5 本文鲁棒算法 1 的检验曲面

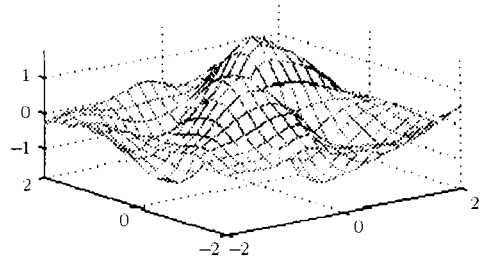


图 6 本文鲁棒算法 2 的检验曲面

表 1 三维曲面仿真结果

算法	迭代次数	CPU 用时 (秒)	拟合偏差	真实偏差	迭代结束原因
BP 算法	500	234.27	24.7275	27.6468	迭代次数达到预定值
文献 [1] 算法	24	4.92	14.9102	10.8253	误差导数达到预定最小值
鲁棒算法 1	30	8.08	9.0812	6.9382	误差导数达到预定最小值
鲁棒算法 2	59	11.30	8.4831	5.1243	误差导数达到预定最小值

从上述图表可以看出,鲁棒算法并非逼近受噪声污染的样本点,而是逼近系统真实输出的估计的置信区间中心,在绝大多数情况下,后者组成的曲线(面)要比前者的平坦,因此,学习效率高(误差导数下降快)、用时少,虽然拟合偏差大,但真实偏差小,检验曲面更接近真实函数.显然,本文的鲁棒算法 1,2 效果优于文献 [1] 的鲁棒算法,而 BP 算法则学不到真实函数.

2. 混合噪声下的仿真研究

在通常情况下,除了环境、设备带来的随机噪声之外,有时还有人为的过失偏差,它的特点是出现机率比较小,数值比较大,一般均值不为零.对数据而言,它也是一种噪声.

为了便于用几何图表表示,选取的仿真对象为

$$y_p = f(x_p) + v_p = e^{x_p} + \sin(2\pi x_p) + v_p, \tag{28}$$

式中 x_p 在 $[-1, 1]$ 之间以 0.05 步长取样,检验时,以 0.01 步长取样,检验样本是学习样本的 25 倍. v_p 为 $(-0.5, 0.5)$ 内均匀分布随机噪声加上绝对值在 $[1, 1.5]$ 的过失偏差. BP 算法和鲁棒算法的结果对比如下.图中,‘+’表示受噪声污染的样本点,实线表示待

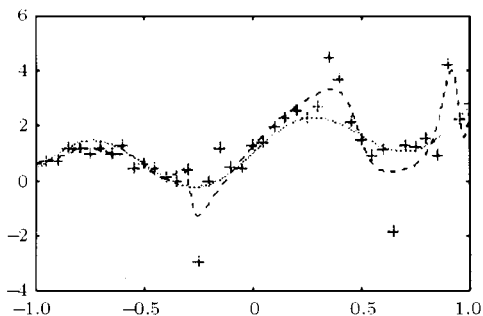


图 7 BP 算法迭代 100 次的结果

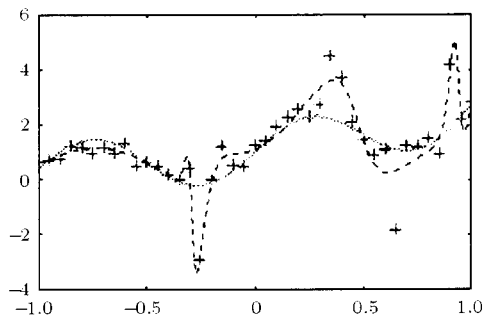


图 8 BP 算法迭代 200 次的结果

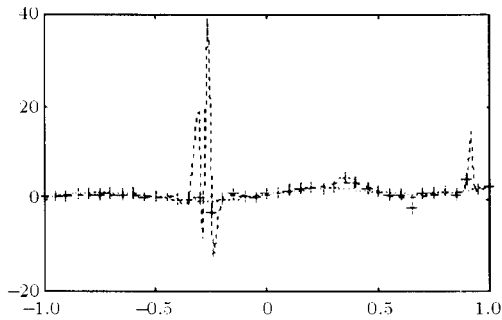


图 9 BP 算法迭代 500 次的结果

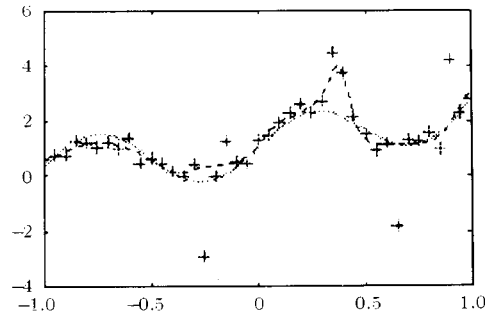


图 10 文献 [1] 算法迭代 22 次的结果

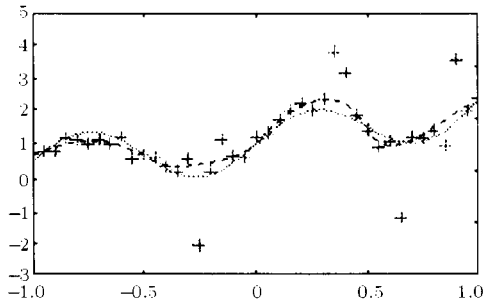


图 11 鲁棒算法 1 迭代 13 次的结果

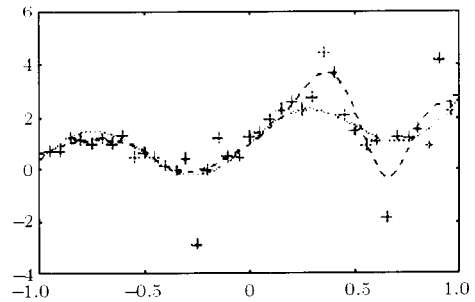


图 12 鲁棒算法 2 迭代 10 次的结果

表 2 仿真结果比较

算法	迭代次数	CPU 用时 (秒)	拟合偏差	真实偏差	迭代结束原因
BP 算法	100	3.61	17.0514	14.5096	迭代次数达到预定值
	200	7.39	10.1738	22.3250	迭代次数达到预定值
	500	18.47	9.2073	23.8357	迭代次数达到预定值
文献 [1] 算法	22	0.90	27.1612	7.9482	误差导数达到预定最小值
鲁棒算法 1	13	0.62	27.8475	2.5801	误差导数达到预定最小值
鲁棒算法 2	10	0.87	22.0921	7.0683	误差导数达到预定最小值

学习的真实函数曲线, 虚线表示神经网络的拟合曲线.

从上面的图表可以看出, 对 BP 算法, 虽然拟合误差最小, 但实际上只能单纯地逼近样本点, 学不到真正的函数, 而且, 随着学习次数的增加, 外推能力反而下降, 即存在“过学习”问题. 鲁棒算法能较快地逼近真实函数曲线. 由于鲁棒算法 1 引入了聚类分析和鲁棒因子, 其效果比文献 [1] 算法要好, 对噪声有较好的抵抗能力, 而鲁棒算法 2 因为其鲁棒因子较简单, 在拟合上效果比不上鲁棒算法 1, 而与文献 [1] 算法相当, 但从中可以看出鲁棒因子的选择范围是较宽的, 它表明鲁棒算法有广泛的适应性.

§6. 结论

本文针对噪声的特点, 提出了两条措施: 1) 利用聚类对噪声一定的抵御能力来抵御幅值小, 均值为零的随机噪声; 2) 引入鲁棒因子, 通过对某些品质好 (受到噪声污染小)

的样本的强化学习, 控制学习进程以抑制大幅值噪声的对学习影响. 仿真结果表明, 本章提出的方法是有效的和实用的.

参 考 文 献

- [1] 李银国, 曹长修, 神经网络鲁棒能量函数的构成原理, 模式识别与人工智能, 9: 3 (1996).
- [2] 汪荣鑫, 随机过程, 西安交通大学出版社, 1986. 12.
- [3] David S. Chen, Ramesh C. Jain, A Robust Back Propagation Learning Algorithm for Function Approximation, IEEE Trans. on Neural Networks, 5: 3 (1994).