

73-74,72 基于线性规划的模糊神经网络

Fuzzy Neural Networks Based on the Linear Programming

江学军¹ 陈 洁²(山东经济学院计划统计系¹ 计算机管理信息系² 济南250014)江志超[✓]

(山东师范大学计算机科学系 济南250014)

F201 TP183

摘 要 本文主要研究网络权值和阈值为模糊数的前馈神经网络,给出了一种利用线性规划确定这种网络权值和阈值的算法,并结合实际算例验证了该算法的优越性。

关键词 模糊神经网络,线性规划,经济预测

一 前言

我们研究的是权值和阈值为模糊数的前馈神经网络,网络模型如图1。假设三层网络有 M 个输入单元, N 个输出单元,隐含层单元个数为 K 个,神经元激活函数为线性函数,训练样本个数为 P 个 (x_p, y_p) , 其中 $X_p = (x_{p1}, x_{p2}, \dots, x_{pM})^T$ 为输入向量, $Y_p = (y_{p1}, y_{p2}, \dots, y_{pN})^T$ 为输出向量, $(p=1, 2, \dots, P)$ 。隐含层节点与输入层节点之间的连接权为 $w_{ij}, i=1, 2, \dots, K, j=1, 2, \dots, M$; 输出层节点与隐含层节点之间的连接权为 $v_{ij}, i=1, 2, \dots, N, j=1, 2, \dots, K$; 隐含层节点阈值为 $u_i, i=1, 2, \dots, K$ 。

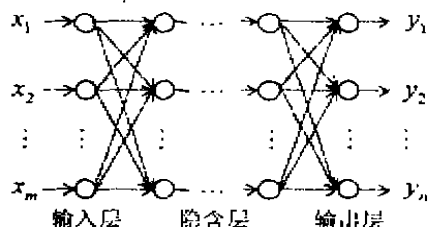


图1 前馈神经网络模型

关于这类网络的研究已有不少^[1,2], Buckley 和 Hayashi^[3]更是对各种形式的模糊神经网络进行了对比和分析。但是,所有这些模糊神经网络都是通过类似BP算法的反向算法修改权值和阈值,操作起来过于复杂,使得模糊神经网络的应用受到限制,权值和阈值的确定只能使用正规模糊数,而本文给出的算法将权值和阈值的确定转化为一个线性规划问题,只要给出模糊集的隶属函数,就可以利用单纯型算法方便地求解。

二 算法

假设,网络实际输出向量为 $\hat{Y}_p = (\hat{y}_{p1}, \hat{y}_{p2}, \dots, \hat{y}_{pN})^T, p=1, 2, \dots, P$; 输出误差为 E :

$$E = \sum_{p=1}^P E_p \quad E_p = \frac{1}{2} \sum_{i=1}^N (y_{pi} - \hat{y}_{pi})^2$$

则网络参数 w_{ij}, v_{ij}, u_i 由使网络误差 E 达到最小求得。

激活函数为(如图2):

$$y(x) = \begin{cases} 1, & x \geq a, \\ bx, & -a < x < a \\ -1, & x < -a \end{cases}$$

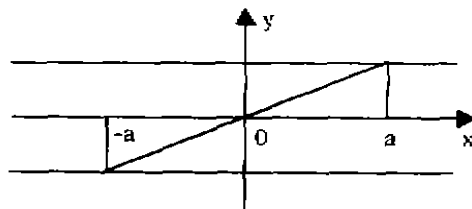


图2 阶跃函数

如果网络参数是对称模糊数 $\tilde{w}_{ij}, \tilde{v}_{ij}, \tilde{u}_i$, 则网络输出也变为一个对称模糊数 $\hat{y}_{pi}, i=1, 2, \dots, N$ 。

对于输入样本 $X_p = (x_{p1}, x_{p2}, \dots, x_{pM})$, 网络的隐含层节点的输入为:

$$\tilde{NodeIn}(i) = \sum_{j=1}^M \tilde{w}_{ij} x_{pj} - \tilde{u}_i = (C_i(x_p), d_i(x_p))_L$$

其中 $c_i(x_p) = c_{i1}x_{p1} + c_{i2}x_{p2} + \dots + c_{iM}x_{pM} - cu_i$,

$$d_i(x_p) = d_{i1}|x_{p1}| + d_{i2}|x_{p2}| + \dots + d_{iM}|x_{pM}| + du_i$$

网络的隐含层节点的输出为:

$$\tilde{NodeOut}(i) = \begin{cases} (1, 0.5)_L \\ (ac_i(x_p), |a|d_i(x_p))_L \\ (-1, 0.5)_L \end{cases}$$

网络输出节点的输出为:

$$\hat{y}_p = \sum_{j=1}^n \tilde{v}_{ij} \tilde{NodeOut}(j) = (cy_i(x_p), dy_i(x_p))_L$$

$$\text{其中 } cy_i(x_p) = cN_1 \times cv_{i1} + cN_2 \times cv_{i2} + \dots + cN_k \times cv_{ik}$$

$$dy_i(x_p) = |cN_1| \times cv_{i1} + |cN_2| \times cv_{i2} + \dots + |cN_k| \times cv_{ik}$$

则网络参数 $\tilde{w}_{ij}, \tilde{v}_{ij}, \tilde{a}_i$ 由下列条件确定:

(1) y_p 包含在隶属度 h_i 以上的子集 $[y_i(x_p)]_{h_i}$ 里, 其中:

$$[y_i(x_p)]_{h_i} = \{y | \mu_{y_i(x_p)}(y) \geq h_i\}$$

(2) 幅宽 $dy_i(x_p)$ 之和最小。

则网络参数的确定, 即可以转化成下述线性规划问题:

$$\min \sum_{i=1}^n \sum_{j=1}^n dy_i(x_p)$$

$$\text{s. t. } y_p \leq c(x_p) + |L^{-1}(h)|w(x_p), i=1, 2, \dots, N$$

$$y_p \geq c(x_p) - |L^{-1}(h)|w(x_p), i=1, 2, \dots, N$$

$$w_i \geq 0, i=0, 1, \dots, n$$

对于这样一个线性规划问题, 只要使用现有的线性规划算法(如: 单纯型方法等)就可以解决, 而不需要 BP 算法那样对样本数进行反复迭代, 从而大大节省了计算量和计算时间。

表1 某开发区主要经济指标及数据(单位: 亿元、万人)

指标 年份	第一产业 增加值	第二产业 增加值	第三产业 增加值	第二产业 投资	第三产业 投资	社会商品 零售额	开发区年 末总人口
86	0.03	0.52	0.235	—	—	—	—
87	0.07	0.73	0.20	—	—	—	—
88	0.06	2.52	0.59	1.61	0.74	0.11	1.19
89	0.11	3.86	1.78	2.51	1.89	0.71	1.95
90	0.12	3.53	1.98	6.93	1.84	1.44	2.05
91	0.75	5.90	3.96	11.41	2.19	1.75	3.60
92	0.63	15.30	6.54	17.35	12.06	2.69	4.01
93	1.10	25.58	16.02	51.40	27.44	3.62	8.37
94	1.50	36.70	18.00	47.43	32.95	4.51	9.04
95	1.82	42.00	26.38	27.03	18.52	5.53	9.44
96	1.60	54.00	32.60	33.93	11.58	6.30	9.74
97	1.80	69.00	34.80	39.00	13.31	7.60	10.93
98	2.50	73.80	44.50	47.69	15.42	7.97	12.04

三 实例

我们利用模糊神经网络模型对某经济技术开发区

表2 模糊神经网络模型的经济预测结果

年份	实际值	下限	中心	上限
86	0.52	0.28	0.51	0.74
87	0.73	0.51	0.79	1.06
88	2.52	2.24	3.19	4.13
89	3.86	3.28	3.96	4.64
90	3.53	3.12	3.83	4.53
91	5.90	5.02	5.98	6.93
92	15.30	13.92	15.88	17.84
93	25.58	22.62	25.95	29.27
94	36.70	31.86	35.89	39.92
95	42.00	38.43	42.40	48.37
96	54.00	47.09	55.64	62.79
97	69.00	60.40	68.27	70.31
98		64.58	75.37	86.15
99		83.52	93.69	103.86
2000		100.83	112.00	123.17

的经济系统进行预测, 以预测第二产业增加值为例, 各项指标数据如表1。我们使用经济模型如下:

$$Y_{2,t} = f(Y_{2,t-1}, Y_{2,t-2}, Y_{2,t-3}, Y_{1,t-1}, Y_{3,t-1}, RK_t, TZ_{2,t})$$

其中 $Y_{1,t}$ 表示第 t 年的第一产业增加值, $Y_{2,t}$ 表示第 t 年的第二产业增加值, $Y_{3,t}$ 表示第 t 年的第三产业增加值, RK_t 表示第 t 年的年末总人口, $TZ_{2,t}$ 表示第 t 年的第二产业固定资产投资, $TZ_{3,t}$ 表示第 t 年的第三产业固定资产投资, LS_t 表示第 t 年的社会商品零售额。

在模糊神经网络模型中, 取 $L(x) = \max(0, 1 - |x|)$, $h = 0.8$, 使用本文算法预测结果如表2。

对于同样的预测问题, 我们使用相同的网络结构和参数, 用原有的 BP 算法也进行了预测, 预测结果略。

表2的中心值与表1的实际值十分接近, 这说明了这种方法的准确性, 而使用原有的 BP 算法进行预测所得到的结果与实际值相比, 偏差较大。在计算时间

(下转第72页)

第二个 HFLANN 的输出为:

$$y_i = f_2(y_i, x_i) = \sum_{j=1}^n \sum_{r=1}^m w_{ijr} \phi_r(v_i) \theta_j(x_i) \quad (7)$$

依次进行, 这样就可得到整个 HHFLANN 的输出

$$f(x_1, x_2, x_3) = f_2(f_1(x_1, x_2), x_3) \quad (8)$$

若 $g(x_1, x_2, x_3)$ 是目标函数, 给定一组训练样本通过学习, 我们可确定权值 w_{ij}, w_{ij} , 使得 $f(x_1, x_2, x_3)$ 逼近于目标函数 g 。

4 HHFLANN 网络整体学习算法

先考察第一层 HFLANN 网络, 随机地给定一组训练数据, $\{(x_1^{(k)}, x_2^{(k)}, y_1^{(k)}) | k=1, 2, \dots, p\}$ 代入 (6) 式, 得到以 $w_{ij} (i=1, 2, \dots, n, j=1, 2, \dots, m)$ 为未知变元的方程组:

$$\sum_{j=1}^n \sum_{r=1}^m w_{ijr} \phi_r(x_1^{(k)}) \theta_j(x_2^{(k)}) = y_1^{(k)} \quad (9)$$

令: $a_{ikr} = \phi_r(x_1^{(k)}) \theta_j(x_2^{(k)})$, 则 (9) 式转化为:

$$\sum_{j=1}^n \sum_{r=1}^m w_{ijr} a_{ikr} = y_1^{(k)} \quad k=1, 2, \dots, p \quad (10)$$

这样, 通过对权值 w_{ij} 学习^[2], 得到满意的解 y_1 。

我们继续考察第二层 HFLANN 网络, 在第一层学习的基础上, 添加新的学习样本:

$$\{(y_1^{(k)}, x_3^{(k)}, y_2^{(k)}) | k=1, 2, \dots, p\}$$

同理, 我们可得到以 w_{ij} 为未知变元方程组:

$$\sum_{j=1}^n \sum_{r=1}^m w_{ijr} (y_1^{(k)}) \theta_j(x_3^{(k)}) = y_2^{(k)} \quad (11)$$

同样的方法将 (11) 转化为:

$$\sum_{j=1}^n \sum_{r=1}^m w_{ijr} b_{ikr} = y_2^{(k)} \quad k=1, 2, \dots, p \quad (12)$$

对权值 w_{ij} 学习, 得到满意的 y_2 作为下一层输入, 依次类推, 最终完成对 HHFLANN 整体系统的学习。

实际上, 我们进一步简化 (10) 式, 若令: $u = (i-1)m + j$, 则 (10) 式可写成:

$$\sum_{k=1}^n a_{uk} w_k = y_1^{(k)} \quad k=1, 2, \dots, p \quad (13)$$

置 $v = mn$, $A = (a_{uk})$, $Y = (y_1^{(1)}, y_1^{(2)}, \dots, y_1^{(p)})^T$, $W = (w_1, w_2, \dots, w_v)$, 则 (13) 式可转化成:

$$AW = Y \quad (14)$$

通过求解矩阵方程 (14), 可方便地完成对 HHFLANN 网络的学习。

结语 文中首先设计出一类双曲函数型网络, 提出一种层次函数网络模型, 讨论了 HHFLANN 网络的整体逼近特性, 给出了 HHFLANN 网络的整体学习算法。该算法不存在局部极小, 更适用于具有层次知识结构的应用领域, 因而, 具有较高的理论价值和广泛的应用背景。

参考文献

- 1 Pao Y H. Adaptive pattern recognition and neural networks. New-Wesley publish company Inc. 1989
- 2 Pao Y H, et al. Neural-net computing and intelligent control of system. Int. J. Control, 1992, 56(2): 263~289
- 3 Patra J C, et al. A functional link artificial neural networks for adaptive channel equalization. signal Processing, 1995 (43): 181~195
- 4 於东军, 王士同, 等. 层次径向神经网络的全局逼近理论. 计算机研究与发展, 1999, 36(11): 1329~1334
- 5 史忠植. 神经计算. 北京: 电子工业出版社, 1993

(上接第74页)

上, 基于线性规划的网络算法是 BP 算法的三分之一。这说明, 原有的 BP 算法计算复杂, 并且易于产生局部极小, 使训练得到的网络不能正确反映预测系统的实际情况。而本文提出的网络算法有效地克服了这些问题。

参考文献

- 1 Buckley J J, Hayashi Y. Numerical relationships between

neural networks, continuous functions. Fuzzy Sets and Systems, 1993, 60: 1~8

- 2 Ishibuchi H, et al. Fuzzy neural networks with fuzzy weights and fuzzy biases. In: Proc. IEEE Int. Conf. Neural Networks, San Francisco, 1993, 3: 1650~1655
- 3 Buckley J J, Hayashi Y. Fuzzy neural networks. A survey. Fuzzy Sets and Systems, 1994, 66: 1~13