

62-64/61 正交神经网络的动态建模方法研究*)

Research of a Dynamic Modeling Method Based on Orthogonal Neural Network

肖少拥 TP18

(浙江大学 CAD&CG 国家重点实验室 浙江大学计算中心 杭州310027)

Abstract This paper presents a dynamic modeling method based on orthogonal neural network, it fully uses the characteristics of the nonlinear processing ability of neural networks and the efficient disposal of the large scaling sparse problems that Givens transform can process. It can not only train the network quickly, but also can optimize the structure of the networks. Simulating experiments show that the new modeling method is a simple universal modeling method for the nonlinear systems.

Keywords Orthogonal neural network, Dynamic modeling, Givens transform

1 引言

在许多动态系统中,系统的输入输出数据容易得到,而其模型难以确定,使得系统辨识和控制变得非常困难。多层前向神经网络可以通过输入输出数据点对逼近任意连续函数^[1],为这个问题的解决提供了一种方法^[2],但传统前向神经网络的 BP 学习算法存在以下问题^[3,4]:

(1)网络结构的确定(隐层神经元个数的选取)无理论上的指导,而是凭经验;

(2)从数学上看, BP 算法是一个非线性优化算法,不可避免存在局部极小问题;

(3)网络初始权值和参数(如学习率)的选取对网络的训练有较大影响,如何选取缺乏理论指导;

(4)学习算法的收敛速度很慢,通常要迭代几千次甚至更多;

(5)对新加样本的学习要影响到已学完的样本,学习效率低。

针对以上问题,文[3]提出一种能克服陷入局部极小的 BP 算法,文[4]则提出无隐层的能快速学习输入输出数据对的正交神经网络模型,它基于 Legendre 正交多项式基进行训练。由于三层前向神经网络结构具有很好的逼近能力^[1],文[5]提出三层正交投影神经网络模型,其学习采用了 Gram-Schmidt 正交化过程,而不对输入层到隐层的权值进行学习。由于正交投影神经网络学习算法存在某些局限性,文[6]采用 Housholder 正交变换加以了改进。

不同于文[4]的无隐层的正交神经网络,本文提出一种正交神经网络三层结构模型,采用 Givens 变换^[7]求解隐层到输出层的权值和阈值,使用 BP 算法微调输入层到隐层的权值,通过 Givens 算法和 BP 算法的循环训练完成网络的学习,同时也完成了网络结构的学习,即确定了网络的隐层节点数,而且对新增样本的学习,可利用以前样本的学习成果,这无疑提高了学习效率。因为网络采用 Givens 正交变换来求解,还具有学习速度快和获得全局最优解的特点,而且不需给出输出层到隐层的初始权值和阈值,使网络训练更加简单,故基于该正交神经网络模型的建模方法是一种动态、高效的及简单可行的建模方法。

2 正交神经网络建模方法

正交神经网络结构如图1所示。设 (x^p, y^p) , $p=1, 2, \dots, P$ 为 P 个建模数据样本。

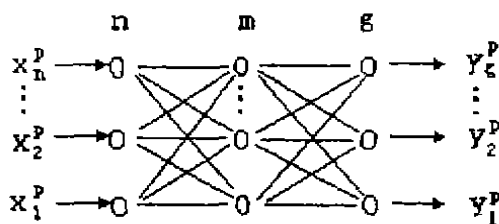


图1 正交神经网络结构

设神经网络的输入层含 n 个神经元,用 $x = (x_1,$

*)本文研究得到浙江大学 CAD&CG 国家重点实验室的支持,肖少拥 博士生,研究兴趣为数值分析、人工智能、神经网络。

$x_1, \dots, x_n$ 表示输入矢量, x^p 表示第 p 个输入样本; 隐层含 m 个神经元, 其输入矢量由 $u_1 = (u_{11}, u_{12}, \dots, u_{1m})$ 表示, 输出矢量由 $h = (h_1, h_2, \dots, h_m)$ 表示; 输出层含 g 个神经元, 其输入矢量由 $u_2 = (u_{21}, u_{22}, \dots, u_{2g})$ 表示, 输出矢量由 $v = (v_1, v_2, \dots, v_g)$ 表示, w_{ij}^1 表示输入层第 i 个神经元与隐层第 j 个神经元的连接权, 其阈值为零; w_{jk}^2 表示隐层第 j 个神经元与输出层第 k 个神经元的连接权, 其阈值为 θ_k , 网络中的非线性函数取 Sigmoid 非线性单调上升函数, 由 $f(x) = 1/(1+e^{-x})$, $f(x) \in (0, 1)$ 表示, 对输入样本 x^p , 网络各层的输入输出关系为:

$$u_{ij}^1 = \sum_{i=1}^n w_{ij}^1 x_i^p, h_j^p = f(u_{ij}^1); j = 1, 2, \dots, m \quad (1)$$

$$u_{jk}^2 = \sum_{j=1}^m w_{jk}^2 h_j^p - \theta_k, v_k^p = f(u_{jk}^2); k = 1, 2, \dots, g \quad (2)$$

对输入矢量 x^p , 网络的实际输出为 v^p , 希望输出为 y^p , 网络的能量函数定义为:

$$E = \frac{1}{2} \sum_{p=1}^P \sum_{k=1}^g (v_k^p - y_k^p)^2 \quad (3)$$

正交神经网络的动态建模方法就是根据学习样本, 一方面确定网络的结构, 即隐层节点数, 另一方面获得网络的权值和阈值。为方便讨论, 不妨设 g 为 1, 且令 $v^p = y^p$, $p = 1, 2, \dots, P$, 显然隐层到输出层的权值和阈值的学习变成典型的插值或拟合问题。这样得到如下线性方程组

$$\sum_{j=1}^m w_{jk}^2 h_j^p - \theta = f^{-1}(y^p); p = 1, 2, \dots, P \quad (4)$$

其中, $f^{-1}(x)$ 为 $f(x)$ 函数的逆映射。

令 $b_p = f^{-1}(y^p)$, $q = m+1$, 则(4)式表示成矩阵形式为:

$$Aw = b \in R^P \quad (5)$$

其中:

$$A = \begin{bmatrix} h_1^1 & h_2^1 & \dots & h_m^1 & -1 \\ h_1^2 & h_2^2 & \dots & h_m^2 & -1 \\ \dots & \dots & \dots & \dots & \dots \\ h_1^P & h_2^P & \dots & h_m^P & -1 \end{bmatrix} \in R^{P \times m}$$

$$W = \begin{bmatrix} w_1^2 \\ w_2^2 \\ \dots \\ w_m^2 \end{bmatrix} \in R^m, w_q^2 = \theta$$

不妨设矩阵 A 的秩为 $r \leq \min(P, m+1)$, 对 A 实施 Givens 变换^[7]得 A 的 QR 分解为:

$$A = QR \quad (6)$$

其中: $Q \in R^{P \times P}$ 的正交阵, $R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix} \in R^{P \times m}$, $R_{11} \in R^{r \times r}$, $R_{12} \in R^{(P-r) \times r}$, $R_{21} \in R^{r \times (P-r)}$, $R_{22} \in R^{(P-r) \times (P-r)}$, 且 $R_{21} = 0, R_{22} = 0$ 。

将 A 的 QR 分解代入线性方程组(5)且两端乘以 Q^{-1} 得

$$Rw = Q^{-1}b \quad (7)$$

令 $w = \begin{pmatrix} X \\ Z \end{pmatrix}$, $Q^{-1}b = \begin{pmatrix} C \\ D \end{pmatrix}$, 其中 $X, C \in R^r$, $Z, D \in R^{P-r}$, 且令

$$w^* = \begin{pmatrix} R_{11}^{-1}C \\ 0 \end{pmatrix} \quad (8)$$

由文[6]的讨论可知, w^* 就是在 w^1 给定的情况下隐层到输出层的最小二乘意义下的最优权值和阈值, 对于 g 大于 1, 就是求解 g 组(5)式的线性方程组。此时若网络的能量值未达到规定精度, 再根据误差对权值 w^1 进行调整, 保证网络的能量值下降, 故有:

$$\Delta w_{ij}^1 = w_{ij}^1(t+1) - w_{ij}^1(t) = -\eta \frac{\partial E}{\partial w_{ij}^1}, \eta > 0 \quad (9)$$

$$\text{由于 } \frac{\partial E}{\partial w_{ij}^1} = \sum_{p=1}^P \sum_{k=1}^g (v_k^p - y_k^p) \frac{\partial v_k^p}{\partial w_{ij}^1}$$

$$\frac{\partial v_k^p}{\partial w_{ij}^1} = \frac{\partial v_k^p}{\partial u_{jk}^2} \cdot \frac{\partial u_{jk}^2}{\partial w_{ij}^1} = v_k^p(1-v_k^p) \cdot \frac{\partial u_{jk}^2}{\partial w_{ij}^1}$$

$$\frac{\partial u_{jk}^2}{\partial w_{ij}^1} = \partial \left(\sum_{i=1}^n w_{ij}^1 h_i^p - \theta_k \right) / \partial w_{ij}^1 = w_{ij}^1 \cdot \frac{\partial h_i^p}{\partial w_{ij}^1} = w_{ij}^1 \cdot$$

$$(1-h_i^p) \cdot h_i^p \frac{\partial u_{ij}^1}{\partial w_{ij}^1}$$

$$\frac{\partial u_{ij}^1}{\partial w_{ij}^1} = \partial \left(\sum_{i=1}^n w_{ij}^1 x_i^p \right) / \partial w_{ij}^1 = x_i^p, \text{ 故有}$$

$$\frac{\partial E}{\partial w_{ij}^1} = \sum_{p=1}^P \sum_{k=1}^g (v_k^p - y_k^p) \cdot v_k^p(1-v_k^p) \cdot h_i^p(1-h_i^p) \cdot w_{ij}^1 \cdot x_i^p \quad (10)$$

由公式(9)、(10)对权值 w^1 进行调整, 这实质是 BP 算法, 只是公式(10)使用的隐层到输出层的权值是通过 Givens 算法得到, 而不是通过 BP 算法学习。

3 正交神经网络学习算法

由上面的讨论可知, 非线性系统的正交神经网络建模方法就是根据给定的学习样本, 通过学习得到 w^1, w^2, θ , 权值和阈值的学习算法如下。

算法1 正交神经网络学习算法(设 $g=1$)

第1步 给定初始权值 w^1 及 η 学习率。

第2步 使用正交 Givens 变换求解得到 w^2 和 θ 。

第3步 权值 w^1 的学习:

(1) 计算网络的能量值 E ;

(2) 若网络训练达到精度要求或迭代次数超限, 转第4步, 否则继续;

(3) 按公式(9)、(10)调整 w^1 的各分量值, 转第2步。

第4步 输出网络的所有权值和阈值及能量值 E , 结束。

当输出神经元有 g 个, 得到 g 组独立的线性方程

组如(5)式所示。对 $k=1,2,\dots,g$ 重复实施算法1, 可得到 w_{ij}^k ($i=1,2,\dots,m; j=1,2,\dots,g$) 和 θ_j^k ($j=1,2,\dots,g$)。网络训练完成后, 由(8)式可知, 训练前的神经网络结构中存在对网络不起作用的隐单元(其与输出层神经元的权值为0), 在设计网络时, 将其忽略, 故该网络模型在完成权值和阈值的学习过程中, 同时完成了网络结构的优化设计。

由于网络隐层节点数在训练完成时可确定, 故训练开始时可设计为样本数。增加新样本时, 先不考虑对 w 的学习, 且令 g 为1。由上面的讨论可知, 得到的线性方程组仅是在(5)式的基础上增加一个线性方程, 由于 Givens 变换具有有效处理稀疏矩阵的特点, 故学习算法可以吸收前面样本学习的成果, 仅需对矩阵最后一行进行处理, 这样提高了网络的学习效率。

4 实验结果

实验一 对 XOR 问题求解, 与文[5]的结果进行比较。

设 Sigmoid 函数为:

$$F(x) = \frac{2(e^x - 1)}{e^x + 1}, F(x) \in (-2, 2) \quad (11)$$

学习样本见表1, 隐层节点数取为3, 初始的网络权值、阈值和 W_0 的取值见表2, 取 ε 值为0.0001。本例设计输入层到隐层的初始权值(其阈值为0), 使 V_1, V_2, V_3, V_4 是线性相关的, 输出神经元的输入端得到的四个矢量见表3, 显然 $V_2=V_3$, 采用文[5]中的算法无法求解。

采用本文的算法1, 不需给出 W_1^0 和 W_0 , 经一次 Givens 算法, 不用 BP 算法调整 W^1 即求得的隐层到输出层的三个权值和阈值见表2, 网络输出神经元的实际输出值见表1。

表1

	输入1	输入2	希望输出值	实际输出值
样本1	0.0	0.0	0.0	0.000000
样本2	0.0	1.0	1.0	0.999997
样本3	1.0	0.0	1.0	0.999997
样本4	1.0	1.0	0.0	-0.000007

表2

W_1^1	1.0	2.0	2.0	θ_1	θ^*
W_2^1	1.0	2.0	2.0	2.0	0.0
W_3^1	10.3	12.5	11.7	—	—
W_0	8.0	7.0	9.0	6.0	3.0
W^{2*}	-3.926	6.144	-3.034	—	—

表3

V_1	0.000000	0.000000	0.000000	2.000000	0.000000
V_2	9.519614	19.039854	17.821304	2.000000	1.098612
V_3	9.519614	19.039854	17.821304	2.000000	1.098612
V_4	15.68884	24.100689	22.558245	2.000000	0.000000

实验二 增加新样本的学习。在实验一的基础上, 增加样本(0.5, 0.5), 由文[3]可知, 网络存在极小点。隐层节点数取为4, 实验数据及结果见表4、表5、表6。输入层到隐层的权值不用学习, 学习是在前四个样本学习的基础上继续, 学习效率高。

表4

	输入1	输入2	希望输出值	实际输出值(前)	实际输出值(后)
样本1	0.0	0.0	0.0	0.000000	0.000000
样本2	0.0	1.0	1.0	0.999995	0.999971
样本3	1.0	0.0	1.0	0.999996	1.000069
样本4	1.0	1.0	0.0	-0.000020	-0.000039
样本5	0.5	0.5	0.0	0.845128	-0.000026

表5

W_1^1	1.0	2.0	2.0	1.0	θ_0	—
W_2^1	2.0	2.0	1.0	1.0	2.0	—

表6

前四个样本训练	W^{2*}	-2.114214	4.608951	-2.114213	-0.808583
	θ^*	0.0	—	—	—
增加第五个样本	W^{2*}	-11.79177	11.47233	-11.79155	13.50663
	θ^*	0.0	—	—	—

表7

x	1	9	18	27	36	45
y(希望)	0.017452	0.156434	0.309017	0.453990	0.587785	0.707106
y(实际)	0.017452	0.156430	0.309011	0.453915	0.587868	0.707047

表8

W^1	0.19	0.46	0.54	0.64	0.94	0.92	—
W^{1*}	0.0089	0.5460	0.2678	0.7322	0.9402	0.92	θ^*
W^2	27.046	-146.91	12.355	395.45	-276.21	0.0	0.0

实验三 网络结构的优化设计。输入输出样本数据见表7, 网络输入输出层神经元个数为1, 隐层节点数取为6, η 为0.01, ε 为 10^{-4} , 初始的网络权值 W^1 见表8, 网络训练完成后的权值和阈值见表8。网络经过3次

(下转第61页)

如何根据不同的多个分类器所给出的识别结果以及结果的置信度,选择最优的结果从而达到提高识别率之目的。在一些对识别速度敏感的应用场合,分类器的增加虽然能够有效地提高识别率,但却会较大地影响识别速度,造成性能的下降。文[11]中提出的方法就是针对这一问题提出的解决方案。此方案能够根据对输入样本影响分类器性能的某些特征——性能预测特征,动态地选择最优的分类器集合,使这样选择的分类器进行组合时能够在统计上使误差与识别消耗时间所造成的代价最小。

结束语 本文从成员分类器的生成方法和组合方法两个角度讨论了多分类器组合的问题,评述了近年来的研究发展,从理论上讨论了多分类器组合的四种原则,同时对它们作了一定的比较。我们认为今后多分类器组合研究的主要重点应当集中于从理论上提出更为完善的分析,如怎样更好地衡量各成员分类器之间相关程度,这是多分类器组合中的一个关键问题。目前对多分类器组合的研究通常是分别讨论成员分类器的生成和组合问题,如何将这两个问题有机地结合起来也应是今后研究的主要领域之一。

参考文献

- 1 Ali K M, Pazzani M J. Error Reduction through Learning Multiple Descriptions, *Machine Learning*, 24, 173~302
- 2 Cherkauer T G. Human Expert-Level Performance on A Scientific Image Analysis Task by A System Using Combined Artificial Neural Networks. In: Chan P, ed. Working Notes of the AAAI Workshop on Integrating Multiple Learned Models. AAAI Press, Menlo Park, CA
- 3 Tax D M J, et al. Comparison between Product and Mean Combination Rules
- 4 Dietterich T G, Bakiri G. Solving Multiclass Learning Problems via Error-Correcting Output Codes. *J. Artificial Intelligence Res.*, 1995, 2: 263~286
- 5 Dietterich T G, Kong E B. Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithms. [Technical Report]. Dept. Computer Science, Oregon State Univ., Corvallis, Oregon
- 6 Ho T K, et al. Decision Combination in Multiple Classifier Systems. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1994, 16(1): 66~75
- 7 Ho T K. The Random Subspace Method for Constructing Decision Forests. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1998, 20(8)
- 8 Ji C, Ma S. Combinations of Weak Classifiers. *IEEE Transactions on Neural Networks*, 1997, 8(1): 867~888
- 9 Kutler J, et al. On Combining Classifiers. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1998, 20(3)
- 10 Kong E B, Dietterich T G. Error-Correcting Output Coding Corrects Bias and Variance. In: Prieditis A, Russell S, eds. Twelfth International Conference on Machine Learning. Morgan Kaufmann, San Francisco, 1993
- 11 Perrone M P, Cooper L N. When Networks Disagree Ensemble Methods for Hybrid Neural Networks. In: Mammon R J, ed. *Neural Networks for Speech and Image Processing*. Chapman and Hall, London
- 12 征荆, 丁晓青, 吴佑青. 基于最小代价的多分类器动态集成. *计算机学报*, 1999, 22(2)

(上接第64页)

Givens 算法和2次 BP 算法的训练,最后确定隐层神经元个数为5,网络的实际输出见表7。

结束语 实验表明本文提出的正交神经网络建模方法是一种网络结构可优化设计的、高效的、具有自适应处理的、简单可行的动态神经网络建模方法。

参考文献

- 1 Hornik K, Stinchcombe M. Multilayer Feedforward Networks Are Universal Approximation. In: White H, ed. *Artificial Neural Networks: Approximation and Learning Theory*. Blackwell Publishers, 1992. 13~28
- 2 Narendra K S, Mukhopadhyay S. Adaptive Control of Nonlinear Multivariable Systems Using Neural Networks. *Neural Networks*, 1994, 7(5): 737~752
- 3 孙德保, 高超. 一种实用的克服极小的 BP 算法研究. *信息与控制*, 1995(5): 283~287
- 4 Yang S S, Tseng C S. An Orthogonal Neural Network for Function Approximation. *IEEE Trans. on System, Man, and Cybernetics, Part B*, 1996(5): 779~785
- 5 肖丹, 施渝萍. 正交投影神经网络. *计算机学报*, 1996(9): 673~678
- 6 肖少拥, 冯树椿, 胡上序. 基于正交变换的神经网络学习算法. *清华大学学报(自然科学版)*, 1998, 38(S2): 148~149
- 7 Golub G H, VanLoan C F. *Matrix Computation*. Johns Hopkins University Press, 1983