

# 一种预测混沌时间序列的模糊神经网络方法

胡玉霞 高金峰

(郑州大学电气工程学院, 郑州 450002)

(2004 年 11 月 18 日收到, 2005 年 5 月 14 日收到修改稿)

给出了一种预测混沌时间序列的模糊神经网络及其学习方法, 给出的方法能直接从数据中提取模糊规则, 经过优化得到最佳模糊规则库, 并利用神经网络的自学习功能修改隶属函数的参数和网络的权值, 减少了规则的匹配过程, 加快了推理速度, 增强了网络的自适应能力. 使用该神经网络及其学习方法对 Lorenz 混沌时间序列进行了预测仿真研究, 试验结果表明给出的预测工具和方法是有效的.

关键词: 模糊神经网络, 模糊规则提取, 混沌时间序列预测

PACC: 0545

## 1. 引言

在混沌动力学系统中, 通过时间序列来研究整个系统的动力学行为并对混沌序列进行预测已成为一个重要的研究方向. 目前, 国内外学者提出了多种混沌时间序列预测方法, 如最大李雅普诺夫指数法<sup>[1,2]</sup>、少参数二阶 Volterra 滤波法、自适应高阶非线性滤波法、神经网络法等<sup>[3-5]</sup>. 已有方法在信号处理、控制理论、经济管理等领域<sup>[6]</sup>获得了广泛应用.

模糊神经网络将模糊系统的逻辑推理能力和神经网络的自适应能力相结合, 既有模糊推理的功能, 又给网络权值赋予了明确的物理意义. 用神经网络技术进行模糊信息处理, 使得模糊隶属度函数的自动生成和模糊规则的自动提取成为可能, 近年来模糊神经技术逐渐在预测混沌时间序列中得到应用<sup>[7-12]</sup>.

本文利用前向模糊神经网络直接从数据中提取模糊规则, 经过优化后得到最佳模糊规则库, 并建立起模糊神经网络模型. 利用神经网络强大的反向传播算法修改隶属函数的参数和神经网络的权值, 弥补了专家经验的不足, 减少了规则匹配时间, 加快了推理速度, 提高了预测混沌时间序列的速度和精度. 预测 Lorenz 系统混沌时间序列的仿真实验结果表明, 本文给出的模糊神经网络方法快速且有效.

## 2. 模糊神经网络结构模型

本文采用的模糊神经网络是一个多输入单输出的多层前向系统, 由五层组成, 如图 1 所示. 设各层的输入用  $I$ 、输出用  $O$  表示, 输入输出的下标表示层, 上标表示层中神经元, 则第 3 层中第 2 个神经元的输入可表示为  $I_3^2$ , 输出可表示为  $O_3^2$ .

各层神经元之间的输入输出关系如下.

第一层为输入层, 层中的节点代表输入变量, 输入节点的个数由被预测系统的维数决定. 该层节点的输入输出关系为

$$I_1^i = x_i, \quad O_1^i = I_1^i, \quad i = 1, 2, \dots, n, \quad (1)$$

其中  $n$  为输入变量的个数.

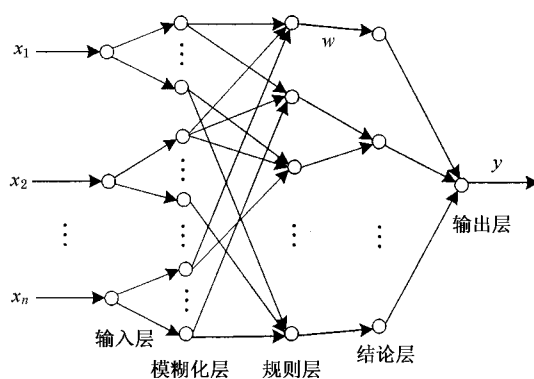


图 1 模糊神经网络结构示意图

第二层为模糊化层,在这一层中把语言变量模糊化,输出为每个语言变量不同分区的隶属度值,本文采用高斯隶属度函数,该层的输入输出关系为

$$I_2^j = O_1^i, O_2^j = \exp\left(-\frac{(I_2^j - c_{ij})^2}{\sigma_{ij}^2}\right), \quad (2)$$

式中  $c_{ij}$  和  $\sigma_{ij}$  分别表示第  $i$  个输入的第  $j$  个隶属函数的中心和宽度,为可调参数。

第三层为模糊规则层,层中的每个神经元对应于一条模糊规则,每个神经元的输入为该规则的前件部分(IF 部分),输出为该规则前件部分隶属度的乘积,即

$$O_3^k = \prod_{s \in I_k} O_2^s, \quad (3)$$

其中  $I_k$  表示与第  $k$  条规则相关的输入变量的下标集。

第四层为结论层(模糊或层),本层对所有激活规则的后件部分(THEN 部分)在不同的模糊分区取或运算,得到推理的结果,该层与第三层节点之间权值为模糊规则的强度  $w_{lk}$ ,为可调参数,则该层节点的输入输出特性为

$$O_4^l = \max_{k \in I_l} (O_3^k w_{lk}), \quad (4)$$

其中  $I_l$  表示与第四层中第  $l$  个神经元相连接的第三层的神经元的下标集。

第五层为输出层,本层神经元及与之相连的权值一起实现解模糊的功能,把模糊集映射成一个确定的值,此处采用面积中心法实现解模糊,即

$$O_5 = \frac{\sum_{l=1}^m O_4^l \sigma_l c_l}{\sum_{l=1}^m O_4^l \sigma_l}, \quad (5)$$

其中  $c_l$  和  $\sigma_l$  分别为输出  $y$  的第  $l$  个模糊分区的中心和宽度。

### 3. 学习算法

在该模糊神经网络中,学习过程分为两部分:第一部分为结构学习阶段,该阶段对输入的数据进行处理,提取出最佳模糊规则,并确定整个系统的结构;第二部分为参数学习阶段,即利用反向传播算法对网络的权值和隶属函数的参数进行调整。

#### 3.1. 结构学习

在模糊系统中,模糊规则的提取是至关重要的

一部分,它决定了模糊神经网络的规模和计算的时间。本文直接通过样本数据来提取模糊规则,并通过计算每条模糊规则出现的次数来确定其重要程度,克服了无专家经验带来的不足。具体实现步骤如下。

第一步:输入输出空间的模糊分割。将输入输出空间分割成若干个子空间,确定每个空间的模糊标记,把数据信息变成语言信息。本系统采用高斯隶属度函数,按隶属度函数中心等分输入输出空间的原则,得出隶属函数的初始参数。

第二步:从样本中提取最佳模糊规则。当模糊逻辑系统输入变量的维数较高且每个变量模糊分区较多的情况下,模糊规则数量呈指数规律增长,将变得非常庞大。尽管描述一个系统的规则很多,但它们所起作用是不同的,有些模糊规则根本就没有出现或出现的概率很小,则这些规则是冗余的,可从规则库中剔除,以简化网络和加快运算速度。本文采用如下算法予以实现。

首先,模糊规则提取及每条规则出现的次数统计。计算第一组数据不同分区的隶属度,找出每个变量对应的最大隶属度所在的分区,组成第一条模糊规则,然后以同样的方法计算第二组数据,若提取出的模糊规则与第一条相同,则第一条规则出现的次数加 1,否则将其作为第二条规则。依次计算所有的样本数据,若样本数据足够多,则可得到系统所需的所有模糊规则及每条规则所出现的次数。

其次,规则合并与优化。给每条规则赋予一个置信度,其大小为规则中各变量隶属度的乘积。在相同规则中,取置信度最大值作为该条规则的置信度。将前件相同而后件不同的规则合并成一条规则,出现次数最多的那条作为合并后的规则,其置信度作为该规则的置信度,且出现次数为所有前件相同规则出现的次数和。

最后,最佳规则库确定。一条规则出现的次数越多表明其越重要,这里用每条规则出现的概率表示其重要度,即

$$p_i = \frac{n_i}{N} \times 100\%, \quad (6)$$

其中  $N$  表示样本总数,  $n_i$  表示第  $i$  条规则出现的次数。根据经验和实际需要,事先定义一个小常数  $\xi$ 。若  $p_i$  小于  $\xi$  则剔除该规则,即可得最佳模糊规则库。

#### 3.2. 参数学习

一旦确定了模糊规则库,则整个系统的结构也

就建立起来了. 利用该系统即可得到从输入到输出的映射, 其精度取决于隶属函数及规则库的准确度. 当系统的推理精度不够高时, 需要对隶属函数的参数和神经网络的权值调整优化. 这里采用梯度下降法使目标误差函数达到最小, 其中目标误差函数为

$$E = \frac{1}{2}(y_d - y)^2, \quad (7)$$

式中  $y_d$  和  $y$  分别表示系统的目标输出  $O_d$  和实际输出  $O_5$ .

### 3.2.1. 网络权值的调整

权值修正公式为

$$w_{ik}(t+1) = w_{ik}(t) - \eta \left( \frac{\partial E}{\partial w_{ik}} \right), \quad (8)$$

其中  $\eta$  为学习率, 把(4)式和(5)式代入(8)式, 连续求偏导可得

$$w_{ik}(t+1) = w_{ij}(t) - \eta \frac{\partial E}{\partial O_5} \times \frac{\partial O_5}{\partial O_4^l} \times \frac{\partial O_4^l}{\partial w_{ik}}. \quad (9)$$

定义各层的误差信号为

$$\delta = \frac{\partial E}{\partial O_5} = y - y_d, \quad (10)$$

$$\delta_5 = \frac{\partial O_5}{\partial O_4^l} = \frac{c_l \sigma_l \sum_{l=1}^m O_4^l \sigma_l - \sigma_l \sum_{i=1}^m O_4^i c_i \sigma_i}{\left( \sum_{j=1}^m O_4^j \sigma_j \right)^2}, \quad (11)$$

$$\delta_4 = \frac{\partial O_4^l}{\partial O_3^k} = w_{ik}, \quad (12)$$

$$\delta_3 = \frac{\partial O_3^k}{\partial O_2^j} = \prod_{lk} O_2^{lk}, \quad (13)$$

其中  $\prod_{lk} O_2^{lk}$  表示第  $k$  条规则前件部分各隶属度去掉  $O_2^j$  后的乘积, 则(9)式可写为

$$w_{ik}(t+1) = w_{ij}(t) - \eta \delta \delta_5 \times \frac{\partial O_4^l}{\partial w_{ik}}. \quad (14)$$

### 3.2.2. 输出变量隶属函数参数的调整

采用梯度下降法对输出变量  $y$  的隶属函数的中心  $c_l$  和宽度  $\sigma_l$  进行调整, 可得

$$\begin{aligned} c_l(t+1) &= c_l(t) - \eta \frac{\partial E}{\partial c_l} \\ &= c_l(t) - \eta \delta \frac{O_4^l \sigma_l}{\sum_{l=1}^m O_4^l \sigma_l}, \end{aligned} \quad (15)$$

$$\sigma_l(t+1) = \sigma_l(t) - \eta \frac{\partial E}{\partial \sigma_l}$$

$$= \sigma_l(t) - \eta \delta \frac{\left( \sum_{l=1}^m O_4^l \sigma_l \right) O_4^l c_l - \left( \sum_{l=1}^m O_4^l c_l \sigma_l \right) O_4^l}{\left( \sum_{l=1}^m O_4^l \sigma_l \right)^2}. \quad (16)$$

### 3.2.3. 输入变量隶属函数参数的调整

采用梯度下降法对输入变量隶属函数参数进行调整, 令  $c_{ij}$  和  $\sigma_{ij}$  分别表示第  $i$  个输入变量的第  $j$  个模糊分区的中心和宽度, 由于

$$\frac{\partial E}{\partial c_{ij}} = \frac{\partial E}{\partial O_5} \times \frac{\partial O_5}{\partial O_4^l} \times \frac{\partial O_4^l}{\partial O_3^k} \times \frac{\partial O_3^k}{\partial O_2^j} \times \frac{\partial O_2^j}{\partial c_{ij}}, \quad (17)$$

$$\frac{\partial E}{\partial \sigma_{ij}} = \frac{\partial E}{\partial O_5} \times \frac{\partial O_5}{\partial O_4^l} \times \frac{\partial O_4^l}{\partial O_3^k} \times \frac{\partial O_3^k}{\partial O_2^j} \times \frac{\partial O_2^j}{\partial \sigma_{ij}}. \quad (18)$$

由(2)式求偏导可得

$$\frac{\partial O_2^j}{\partial c_{ij}} = - \frac{\chi(x_i - c_{ij})}{\sigma_{ij}^2} O_2^j, \quad (19)$$

$$\frac{\partial O_2^j}{\partial \sigma_{ij}} = - \frac{\chi(x_i - c_{ij})}{\sigma_{ij}^2} O_2^j. \quad (20)$$

则输入变量的隶属函数的中心和宽度调整为

$$c_{ij}(t+1) = c_{ij}(t) - \eta \delta \delta_5 \delta_4 \delta_3 \frac{\chi(x_i - c_{ij})}{\sigma_{ij}^2} O_2^j, \quad (21)$$

$$\sigma_{ij}(t+1) = \sigma_{ij}(t) - \eta \delta \delta_5 \delta_4 \delta_3 \frac{\chi(x_i - c_{ij})}{\sigma_{ij}^2} O_2^j. \quad (22)$$

输入变量的中心和宽度在学习的过程中按(21)和(22)式进行修正, 直至学习结束. 当预测精度达到要求或学习时间结束时, 学习过程结束. 这里采用系统输出的均方根误差作为评判标准, 即

$$\text{RMSE} = \left( \frac{1}{N-1} \sum_{i=1}^N (y_{id} - y_i)^2 \right)^{1/2}, \quad (23)$$

式中  $N$  为训练数据样本个数,  $y_{id}$  和  $y_i$  分别为第  $i$  个时刻期望输出和实际输出, 当 RMSE 小于预定精度时, 学习结束, 否则继续学习.

## 4. 预测实验

为验证上述模糊神经网络及其学习算法的性能, 对 Lorenz 系统的混沌时间序列进行预测仿真实验. Lorenz 系统方程为

$$\begin{cases} \dot{x} = a(y - x), \\ \dot{y} = -xz + rz - cy, \\ \dot{z} = xy - bz. \end{cases} \quad (24)$$

取  $a = 10$ ,  $b = 8/3$ ,  $c = 28$ , 初值  $x(0) = 0$ ,  $y(0) = 1$ ,  $z(0) = 0$ . 利用定步长 ( $\Delta t = 0.005$ ) 四阶龙格库塔法

获取变量  $x$  的序列, 并进行相空间重构. 取嵌入维数  $m = 7$ , 延迟时间  $\tau = 10 \times \Delta t = 0.05$ , 由  $t$  时刻及以前的  $m$  个值预测未来  $t + \tau$  时刻的值的映射为

$$x(t + \tau) = f(x(t), x(t - \tau), \dots, x(t - (m - 1)\tau)). \quad (25)$$

用模糊神经网络逼近映射函数  $f$ , 从 9000 组数据中提取模糊规则, 取  $\xi = 0.005$ , 经过结构学习, 得到 48 条模糊规则, 从而建立了一个 5 层 112 个节点 160 个参数的神经网络.

用本文的学习方法把规则由  $7^8 = 5764801$  个减

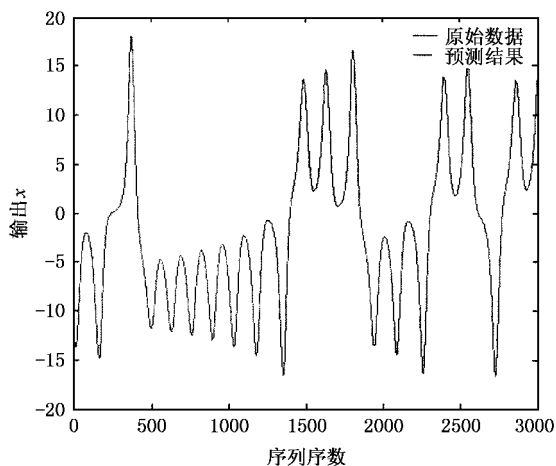


图 2 预测结果与实际值

少到了 48 个, 大大减小了系统的规模和参数的数量, 减小了规则匹配时间, 加快了推理速度.

取 3000 组数据参数训练, 3000 组数据校验网络, 取学习率  $\eta = 0.01$ ,  $RMSE = 0.005$ , 然后用所得系统进行仿真. 仿真的结果为  $RMSE = 0.0168$ , 最大相对误差为 0.0289, 预测结果和真实值之间的比较如图 2 所示, 相对误差结果如图 3 所示. 由此可见, 所得结果具有较高精度. 对蔡氏电路、Rössler 系统和 Chen 系统的时间序列的预测结果也表明该方法具有较高的精度.

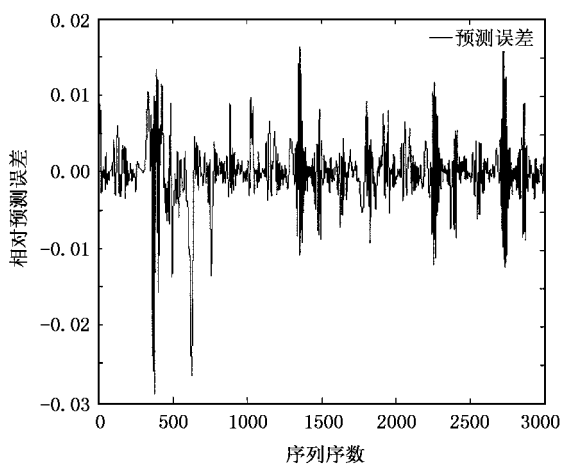


图 3 相对预测误差曲线

## 5. 结 论

本文所提出的模糊神经网络能自动提取系统的

最佳模糊规则, 克服了建立规则时需要专家指导的不足, 结合了神经网络强大的计算能力和自适应能力, 加快了学习速度, 提高了网络精度, 为混沌时间序列的预测提供了一条有效的途径.

- [ 1 ] Zhang J et al 2004 *Computers Electrical Engineering* **30** 1
- [ 2 ] Bolli E M 2000 *International Journal of Bifurcation and Chaos* **10** 1407
- [ 3 ] Zhang J S and Xiao X C 2000 *Acta Phys. Sin.* **49** 403 (in Chinese) [ 张家树、肖先赐 2000 物理学报 **49** 403 ]
- [ 4 ] Zhang J S and Xiao X C 2000 *Acta Phys. Sin.* **49** 2333 (in Chinese) [ 张家树、肖先赐 2000 物理学报 **49** 2333 ]
- [ 5 ] Zhang J S and Xiao X C 2001 *Acta Phys. Sin.* **50** 1248 (in Chinese) [ 张家树、肖先赐 2001 物理学报 **50** 1248 ]
- [ 6 ] Tao X H and Huang M X 2000 *Proceedings of the EPSA* **12**(5)37 (in Chinese) [ 陶小虎、黄民翔 2000 电力系统及其自动化 **12**(5)37 ]
- [ 7 ] Liang R H and Chang C C 2002 *Electrical Power and Energy Systems*

**24** 103

- [ 8 ] Tan W and Wang Y N 2003 *Acta Phys. Sin.* **52** 795 (in Chinese) [ 谭文、王耀南 2003 物理学报 **52** 795 ]
- [ 9 ] Magure L P et al 1998 *Information Sciences* **112** 125
- [ 10 ] Marian B et al Sept 2000 *Forth international conference on knowledge-based Intelligent Engineering Systems & Allied Technologies* 450 - 453
- [ 11 ] Zhang J M and Wang SH Q Aug 2002 *Proceedings of the second International Symposium on Instrumentation Science and Technology* 820 - 824
- [ 12 ] Ajoy Kumar Palit and Popovic D Jul 1999 *International Joint Conference on Neural Networks (IJCNN '99)* 1538 - 1543

# A neuro-fuzzy method for predicting the chaotic time series

Hu Yu-Xia Gao Jin-Feng

( *School of Electrical Engineering , Zhengzhou University , Zhengzhou 450002 , China* )

( Received 18 November 2004 ; revised manuscript received 14 May 2005 )

## Abstract

A neuro-fuzzy approach based on a novel hybrid learning method is presented , which can generate the best fuzzy rule set automatically from the desired input-output data pairs only and can give the initial neuro-fuzzy system and the initial parameters of fuzzy membership functions. Then the parameters of fuzzy membership functions and the weights can be easily tuned by employing neural network 's self-learning techniques. This approach reduces the rule matching time and accelerates the speed of the fuzzy logic referencing and improves the adaptability of the neuro-fuzzy system. Using the proposed neuro-fuzzy system and the learning algorithms we simulated the prediction of the Lorenz chaotic time series , the results demonstrate the effectiveness of the chaotic time series prediction approach.

**Keywords** : neuro-fuzzy network , fuzzy rules extraction , chaotic time series prediction

**PACC** : 0545