

文章编号: 1001-0920(2007)05-0562-04

混沌时间序列的混合粒子群优化预测

刘伟^{1,2}, 王科俊¹, 邵克勇²

(1. 哈尔滨工程大学 自动化学院, 哈尔滨 150001; 2. 大庆石油学院 电气工程系, 黑龙江 大庆 163318)

摘要: 提出一种混合粒子群优化算法,即在改进粒子群优化算法全局搜索模型参数的基础上,利用梯度下降法进一步确定径向基神经网络模型参数,以提高网络的收敛精度和网络性能.采用基于 RBFNN 的混合粒子群优化算法进行离散 Henon 和连续 Mackey-Glass 混沌时间序列预测仿真,结果表明该算法能快速精确地预测混沌时间序列,是研究复杂非线性动力系统辨识和控制的一种有效方法.

关键词: 混沌时间序列; 粒子群算法; 径向基神经网络; 梯度下降法

中图分类号: TP18 **文献标识码:** A

Predicting chaotic time series using hybrid particle swarm optimization algorithm

LIU Wei^{1,2}, WANG Ke-jun¹, SHAO Ke-yong²

(1. College of Automation, Harbin Engineering University, Harbin 150001, China; 2. Department of Electrical Engineering, Daqing Petroleum Institute, Daqing 163318, China. Correspondent: LIU Wei, E-mail: oliuwei@163.com)

Abstract: A hybrid particle swarm optimization (HPSO) is proposed, in which the gradient descent is combined with modified particle swarm optimization (MPSO). The MPSO is determined by linearly decreasing inertia weight and constriction factor weight to speed up global search. Crossover and mutation operation is embedded to avoid the common defect of premature convergence. By using the proposed HPSO algorithm based on RBFNN, simulation for the chaotic time series prediction of discrete Henon and continuous Mackey-Glass chaotic time series is made to test the validity. Simulation results show that the HPSO can accurately predict chaotic time series and an effective approach is provided to study the properties of complex nonlinear dynamic system.

Key words: Chaotic time series; Particle swarm optimization; Radial basis function neural networks; Gradient descent

1 引言

混沌是非线性动力学所特有的一种运动形式,它广泛存在于自然界.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,并在信号处理、自动控制等领域得到了广泛的应用^[1].混沌时间序列是指对混沌系统进行观测采样而得到的一个单变量时间序列.混沌时间序列的预测可看作动力学系统研究的反问题^[2].国内外学者提出了多种混沌时间序列预测方法,如少参数二阶 Volterra 滤波法^[3]、模糊神经网络模型^[4]、支持向量机模型^[5]等.

PSO 算法是一种新兴的随机全局优化技术,是一类有着潜在竞争力的神经网络学习算法.文献[6]利用 PSO 算法训练积单元神经网络,取得了良好的效果.但将 PSO 算法用于训练 RBFNN 的研究尚不多见.

2 RBFNN 模型

RBFNN 是在鉴借生物局部调节和交叠接受区域知识的基础上,提出的一种采用局部接受域来执行函数映射的人工神经网络.RBFNN 是由输入层、隐含层和输出层构成的三层前向网络^[7].隐含层神经元采用径向基函数作为激励函数,它具有形式简单、径向对称、光滑性好和任意阶导数存在的优点.

收稿日期: 2006-06-08; 修回日期: 2006-08-23.

基金项目: 黑龙江省教育厅科学技术研究项目(10551018).

作者简介: 刘伟(1971—),男,黑龙江宾县人,副教授,硕士,从事智能控制理论与应用等研究;王科俊(1962—),男,吉林省吉林市人,教授,博士,从事模式识别、智能系统等研究.

理论上已证明,只要网络隐层的节点数足够多, RBFNN 所执行的从 m 维输入空间到 n 维输出空间的非线性映射,可以任意精度逼近任何单值连续函数^[8].

高斯神经网络可用数学公式表示为

$$\phi_i(x) = \exp(-\frac{(x - c_i)^2}{\sigma_i^2}), \quad (1)$$

$$y = W^T \Phi(x), \quad i = 1, 2, \dots, p. \quad (2)$$

其中: $x \in R^n$ 为网络的输入向量; $y \in R^m$ 为网络的输出向量; $\Phi(x) = [\phi_1(x), \phi_2(x), \dots, \phi_p(x)]$ 为隐层的输出向量, p 为隐层单元的个数; c_i, σ_i 和 w_i 分别为第 i 个隐层单元的中心、宽度和权值向量.

3 粒子群优化(PSO) 算法

PSO 是一种基于群智能的演化计算技术^[9], 是演化计算领域的一个新的分支,它源于对鸟群捕食行为的研究.

3.1 PSO 算法数学描述

假设在一个 d 维的目标搜索空间,有 q 个代表潜在问题解的粒子组成一个种群 $S = (X_1, X_2, \dots, X_q)$, 其中 $X_i = (x_{i1}, x_{i2}, \dots, x_{id}) (i = 1, 2, \dots, q)$ 表示第 i 个粒子在 d 维解空间的一个矢量点. 将 X_i 代入一个与求解问题相关的目标函数,可计算出相应的适应度. 用 $P_i = (p_{i1}, p_{i2}, \dots, p_{id}) (i = 1, 2, \dots, q)$ 记录第 i 个粒子自身搜索到的最好点(即 p_{best}). 在这个种群中,至少有一个粒子是最好的,将其编号记为 g . 则 $P_g = (p_{g1}, p_{g2}, \dots, p_{gd}) (g \in \{1, 2, \dots, q\})$ 就是种群搜索到的最好值(即 g_{best}). 每个粒子还有一个速度变量,可用 $V_i = (v_{i1}, v_{i2}, \dots, v_{id}) (i = 1, 2, \dots, q)$ 表示^[10].

标准 PSO 算法操作的公式如下:

$$V_i^{k+1} = V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k), \quad (3)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1}. \quad (4)$$

其中: k 为迭代代数; 学习因子 c_1 和 c_2 是两个正常数,一般取值为 2; r_1 和 r_2 是均匀分布于 $[0, 1]$ 之间的两个随机数. 为将 V_i^k 和 X_i^k 的值控制在合理的区域内,需要指定 V_{max} 和 X_{max} 予以限制.

对于各类问题的解决,局部搜索和全局搜索都起着重要作用. 在某一类优化问题的具体解决中,应权衡局部搜索和全局搜索的贡献. Shi 等在式(1)中引入一个惯性权重因子 w , 以权衡全局搜索和局部搜索的能力^[11].

3.2 改进的粒子群优化算法

本文提出改进的粒子群优化算法(MPSO), 即同时采用线性递减惯性权重和收缩因子^[12], 并结合遗传算法进行交叉变异. 于是式(3)和(4)变为

$$V_i^{k+1} = [wV_i^k + c_1 r_1 (P_i^k - X_i^k) +$$

$$c_2 r_2 (P_g^k - X_i^k)] J, \quad (5)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1}. \quad (6)$$

对于惯性权重 w , 采用线性递减权策略, 即

$$w(t) = \frac{(W_{ini} - W_{end})(T_{max} - t)}{T_{max}} + W_{end}. \quad (7)$$

其中: t 为当前进化代数, T_{max} 为最大进化代数, w_{ini} 为初始惯性权值, w_{end} 为进化到最大代数时的惯性权值. 对于收缩因子 ω , 取 $\omega = 0.73$.

将父代最大适应度邻域 k 个粒子对应位置和速度与最小适应度邻域 k 个粒子对应位置和速度交叉变异, 得到子代新的位置和速度

$$child_1(X_i) = \omega parent_1(X_i) + (1 - \omega) parent_2(X_i), \quad (8)$$

$$child_2(X_i) = \omega parent_2(X_i) + (1 - \omega) parent_1(X_i), \quad (9)$$

$$child_1(V_i) = \frac{\omega parent_1(V_i) + parent_2(V_i)}{\omega parent_1(V_i) + parent_2(V_i)} / parent_1(V_i), \quad (10)$$

$$child_2(V_i) = \frac{\omega parent_1(V_i) + parent_2(V_i)}{\omega parent_1(V_i) + parent_2(V_i)} / parent_2(V_i). \quad (11)$$

其中 ω 是均匀分布在 $[0, 1]$ 内的随机数.

4 基于 RBFNN 网络的混合粒子群优化算法

RBFNN 网络的优化学习过程分为两个阶段: 第 1 阶段, 根据所有的输入样本, 利用 MPSO 全局搜索隐层各节点的高斯函数的中心值 c_i , 宽度 σ_i 和隐层与输出层的连接权值 w_{ij} ; 第 2 阶段, 在 MPSO 全局搜索模型参数的基础上, 利用梯度下降法进一步确定网路模型参数, 以提高网络的精度.

具体操作方法为: 首先确定神经网络的结构: 输入层为 n 个输入, 隐层神经元数为 p , 输出层为 m 个神经元. 定义粒子群的位置向量 X_i 包括 RBFNN 的隐层单元的中心 $C \in R^{p \times n}$, 宽度 $\sigma \in R^p$ 和隐层与输出层的连接权值 $W \in R^{m \times (p+1)}$. 分别按行展开, 有

$$X_i = (c_{i1}, c_{i2}, \dots, c_{ip}, \sigma_{i1}, \sigma_{i2}, \dots, \sigma_{ip}, w_{i10}, w_{i11}, \dots, w_{imp}). \quad (12)$$

然后用改进 PSO 算法搜索最优位置, 采用均方误差使如下性能指标达到最小:

$$J = \frac{1}{N} \sum_{j=1}^m \sum_{i=1}^N (y_{ji} - \hat{y}_{ji})^2. \quad (13)$$

其中: N 为训练机的样本数, y_{ji} 为第 i 个样本第 j 个网络输出节点的理想输出值, \hat{y}_{ji} 为第 i 个样本第 j 个网络输出节点的实际输出值.

粒子群优化算法步骤如下:

步骤 1: 输入系统网络结构数据参数, 确定变量速度 V 和位置 X 的上限和下限.

步骤 2: 初始化粒子种群.

步骤 3: 根据采样输入计算网络系统模型输出和误差.

步骤 4: 评价每个粒子的适应度, 计算当前进化代数时每个粒子的个体最优适应度 $p_{f\text{best}}$ 和种群全局最优适应度 $g_{f\text{best}}$.

步骤 5: 根据 $p_{f\text{best}}$ 和 $g_{f\text{best}}$, 计算每个粒子的个体最优位置 p_{best} 和种群全局最优位置 g_{best} .

步骤 6: 根据式(5)~(7)进行粒子速度更新、位置更新和惯性权重更新.

步骤 7: 如果速度 $V > V_{\text{max}}$, 则 $V = V_{\text{max}}$; 如果速度 $V < -V_{\text{max}}$, 则 $V = -V_{\text{max}}$.

步骤 8: 如果位置 $X > X_{\text{max}}$, 则 $X = X_{\text{max}}$; 如果位置 $X < -X_{\text{max}}$, 则 $X = -X_{\text{max}}$.

步骤 9: 根据式(8)~(11), 对父代最大适应度邻域 k 个粒子对应位置和速度与最小适应度邻域 k 个粒子对应位置和速度进行交叉变异操作, 得到子代.

步骤 10: 如果满足收敛条件 MSE, 则结束迭代进化, 输出最优粒子位置; 否则, 转到步骤 3, 重复迭代进化.

步骤 11: 根据得到的最优粒子位置的神经网络连接权值预测混沌序列.

步骤 12: 绘制预测混沌序列与实际混沌序列仿真对比曲线和误差曲线.

5 预测实验

5.1 Henon 离散混沌时间序列预测

Henon 映射是著名的简单动力学系统之一, 其动力学方程为

$$\begin{cases} x(k+1) = 1 + y(k) - ax^2(k), \\ y(k+1) = by(k). \end{cases} \quad (14)$$

当 $a = 1.4$, $b = 0.3$ 时, 系统成为混沌态. 利用式(14)的迭代计算, 可得到时间序列. 抽取 500 对输入输出数据, 前 400 个数据对用于训练集, 后 100 个数据对验证模型的准确性. 用基于 RBFNN 混合粒子群优化算法进行建模, RBFNN 的学习过程分为两个阶段: 第 1 阶段, 根据所有的输入样本利用 MPSO 全局搜索, 进化 500 次后其均方误差达到 0.035 835; 第 2 阶段, 在 MPSO 模型参数的基础上, 利用梯度下降法进一步确定网路模型参数, 以提高网络的精度, 迭代 10 000 次, 其均方误差达到 0.000 302.

该方法对 Henon 时间序列预测与实际输出的

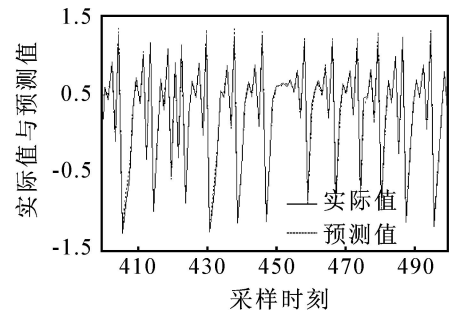


图1 Henon 时间序列预测结果

比较如图 1 所示. 可见, 对于离散混沌模型, 预测值与系统的实际值吻合得很好.

5.2 Mackey-Glass 连续混沌时间序列预测

Mackey-Glass 混沌时间序列可由时滞微分方程得到

$$\frac{dx(t)}{dt} = \frac{x(t-\tau)}{1+x(t-\tau)} - x(t). \quad (15)$$

令 $\tau = 0.2$, $\alpha = 0.1$, $\beta = 10$, 是唯一可调的时滞参数. 文献[13]对 Mackey-Glass 方程的行为特性作了深入研究. 当 $\tau > 17$ 时, 方程(15)产生混沌, 并具有分形维数近似为 2.1 的奇异吸引子. τ 值越大, 混沌程度越高. 选取 $\tau = 17$, 为了获取每一积分时间点的的时间序列值, 用四阶龙格-库塔法寻找方程的数值解, 时间步长为 0.1. 选取 1 000 对学习样本数据, 即 $[x(t-18), x(t-12), x(t-6), x(t), x(t+6)]$, $t = 19, 20, \dots, 1 018$, $\tau = 6$.

式中前 4 个数据为输入, 最后一个为输出, 记为 $x(t+\tau)$. 前 500 对数据作为训练数据, 其余 500 对数据作为测试数据, 以验证辨识模型的有效性. 根据所有的输入样本利用 MPSO 全局搜索, 进化 500 次后, 其均方误差达到 0.000 125. 在 MPSO 模型参数基础上, 利用梯度下降法进一步优化网络模型参数, 迭代 10 000 次, 其均方误差达到 0.000 041.

该方法对 Mackey-Glass 时间序列的预测值与实际输出的比较如图 2 所示. 可见, 对于连续混沌模型, 预测值与系统的实际值吻合得更好.

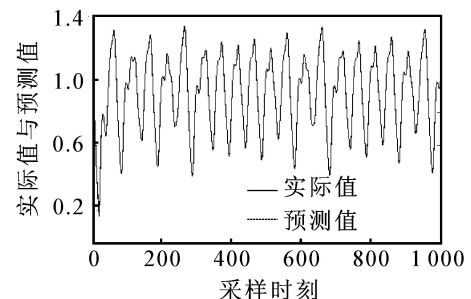


图2 Mackey-Glass 混沌时间序列预测结果

6 结论

本文提出一种预测混沌时间序列的混合粒子群优化 RBFNN 方法. 首先利用 MPSO 算法的全局搜

索能力,确定 RBFNN 网络模型权值初始值;然后在 MPSO 全局搜索模型参数的基础上,利用梯度下降法进一步确定模型参数,以提高模型的精度.该方法能克服普通梯度下降法中网络模型权值初始化不好时,易对网络收敛性能产生影响的不足.该方法用于离散和连续混沌时间序列预测的仿真结果表明,其性能优于常规的高斯径向基函数神经网络和一般粒子群优化算法.

参考文献(References)

- [1] 刘秉正. 非线性动力学与混沌基础[M]. 长春:东北师范大学出版社,1994.
(Liu B Z. Nonlinear dynamics and chaos basis [M]. Changchun: Northeast Normal University Press, 1994.)
- [2] 李军,刘君华. 一种新型广义 RBF 神经网络在混沌时间序列预测中的研究[J]. 物理学报, 2005, 54(10): 4569-4577.
(Li J, Liu J H. On the prediction of chaotic time series using a new generalized radical basis funtion neural networks[J]. Acta Physica Sinica, 2005, 54(10): 4569-4577.)
- [3] 张家树,肖先赐. 用于混沌时间序列自适应预测的一种少参数二阶 Volterra 滤波器[J]. 物理学报, 2001, 50(7): 1248-1254.
(Zhang J S, Xiao X C. A reduced parameter second-order Volterra filter with application to nonlinear adaptive prediction of chaotic time series [J]. Acta Physica Sinica, 2001, 50(7): 1248-1254.)
- [4] Maguire L P, Roche B, McGinnity T M, et al. Predicting a chaotic time series using a fuzzy neural network[J]. Information Sciences, 1998, 112(1): 125-136.
- [5] 崔万照,朱长纯,保文星,等. 混沌时间序列的支持向量机预测[J]. 物理学报, 2005, 53(10): 3303-3340.
(Cui W Z, Zhu C C, Bao W X, et al. Prediction of the chaotic time series using support vector machines for fuzzy rule-based modeling [J]. Acta Physica Sinica, 2005, 53(10): 3303-3340.)
- [6] Van den Bergh F, Engelbrecht A P. Training product unit network using cooperative particle swarm optimizers [C]. Proc of the 3th Genetic and Evolutionary Computation Conf. Washington, 2001: 126-132.
- [7] Chng E S, Chen S, Mulgrew B. Gradient radial basis function networks for nonlinear and nonstationary time series prediction[J]. IEEE Trans on Neural Networks, 1996, 7(1): 190-194.
- [8] Park J Y, Irwin W Sandberg. Approximation and radial-basis-function networks march [J]. Neural Computation, 1993, 5(2): 305-316.
- [9] Kennedy J, Eberhart R C. Particle swarm optimization [C]. Proc of IEEE Int Conf on Neural Network. Piscataway: IEEE Press, 1995: 1942-1948.
- [10] 谢晓锋,张文俊,杨之廉. 微粒群算法综述[J]. 控制与决策, 2003, 18(2): 129-134.
(Xie X F, Zhang W J, Yang Z L. Overview of particle swarm optimization[J]. Control and Decision, 2003, 18(2): 129-134.)
- [11] Shi Y, Eberhart R C. A modified particle swarm optimizer [C]. 1998 IEEE Int Conf on Evolutionary Computation. Anchorage, 1998: 69-73.
- [12] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization [C]. Proc of 2000 Congress Evolutionary Computation. Piscataway, 2000: 84-88.
- [13] Mackey M C, Glass L. Oscillation and chaos in physiological control systems[J]. Science, 1977, (2): 287-289.

(上接第 561 页)

- [7] Olfati Saber R, Murray R M. Consensus problems in networks of agents with switching topology and time-delays[J]. IEEE Trans on Automatic Control, 2004, 49(9): 1520-1533.
- [8] Tanner H G, Jadbabaie A, Pappas GJ. Stable flocking of mobile agents: Part I — Fixed topology[C]. Proc of IEEE Conf on Decision and Control. Maui, 2003: 2010-2015.
- [9] Tanner H G, Jadbabaie A, Pappas GJ. Stable flocking of mobile agents: Part II — Dynamic topology[C]. Proc of IEEE Conf on Decision and Control. Maui, 2003: 2016-2021.
- [10] Yu H, Wang Y J. Stable flocking motion of mobile agents following a leader in fixed and switching networks [J]. Int J of Automation and Computing, 2006, 3(1): 8-16.
- [11] Godsil C, Royle G. Algebraic graph theory [M]. New York: Springer-Verlag, 2001.