文章编号:1672 -058X(2009)02 -0156 -06

蚁群算法训练神经网络辨识混沌系统

成 伟
(重庆大学 自动化学院, 重庆 400044)

摘 要:提出了一种利用蚁群算法训练神经网络的算法,进行混沌系统辨识,并与神经网络、遗传神经网络对同一混沌系统辨识的结果进行比较;实验表明:利用蚁群算法训练神经网络进行混沌系统的辨识,能克服 BP 求解精度低、搜索速度慢、易于陷入局部极小的缺点;与遗传神经网络相比,虽然执行时间有所增加,但求解精度显著提高,可有效用于混沌系统辨识。

关键词:神经网络;混沌系统;蚁群算法;系统辨识
中图分类号:TP 183 **文献标识码**:A

看似复杂且随机变化但却遵循某种特点物理定律的系统,称为混沌系统。1963 年气象学家 Lorenz 提出"确定系统中存在非周期性运动",首次描述了混沌现象[1]。Li 和 Yorke[2]正式将这种现象命名为混沌(Chaos)。由于具有"初始值敏感依赖"特性[3],混沌系统辨识一直非常困难。Hornik[4]证明采用 Sigmoid 传递函数的 3 层前馈神经网络能够以任意精度逼近复杂的非线性关系,因此 BP 适合辨识非线性的混沌系统[5,6]。但是研究证明[7],基于梯度下降的 BP 算法依赖于初始权值的选择,收敛速度缓慢且容易陷入局部最优。

由于具有并行搜索策略及全局优化特性,遗传算法广泛用来训练神经网络。实验表明[8],与 BP 比较,采用遗传算法训练的神经网络同时提高了分类的正确率和收敛速度。但复杂的遗传操作使训练时间随问题规模及复杂程度呈指数级增长[9]。而且由于缺乏有效的局部区域搜索机制,算法在接近最优解时收敛缓慢甚至出现收敛停滞现象[10]。蚁群优化算法(Ant Colony Optimization,简称 ACO)是 20 世纪 90 年代初由意大利学者 M. Dorigo、V. Maniezzo、A. Colorni 等人提出的仿生优化算法,目前用于解决包括任务规划[11]、资源优化[12]、电磁设备设计[13]在内的诸多问题,都取得了较好应用结果。采用 ACO 训练 BP 神经网络,用于混沌系统的辨识。与 BP、采用遗传算法训练的 BP 等对比实验表明,ACO 是有效的神经网络训练算法,能够克服 BP 收敛速度缓慢且容易陷入局部最优等缺点;与遗传算法相比,ACO 虽执行时间略有增加,但搜索精度显著提高。

1 Logistic 映射混沌系统

1976 年数学生态学家 R. May 在《自然》杂志上发表的生态学中的没有世代交叠的虫口模型(现称 Logistic 映射),可用非线性差分方程来描述:

$$x_{n+1} = \lambda x_n(1 - x_n); \lambda \in [0,4], x \in [0,1] \quad (1)$$

对 Logistic 映射的研究发现,Logistic 映射是经过倍周期分岔达到混沌的,如图 1 所示。图 2 给出 $x = 0.5$; $\lambda = 3.7$时的混沌数据序列。

收稿日期:2009 -02 -12;**修回日期**:2009 -03 -15。
作者简介:成伟(1975 -),男,重庆市北碚区人,硕士研究生,从事控制理论研究。

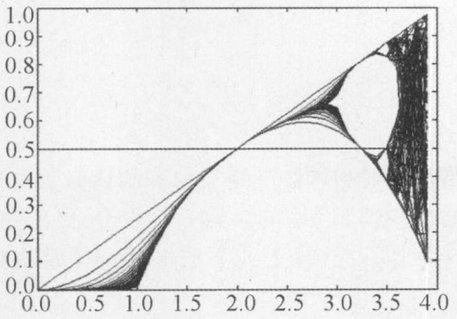

图 1　Logistic 映射的分岔过程

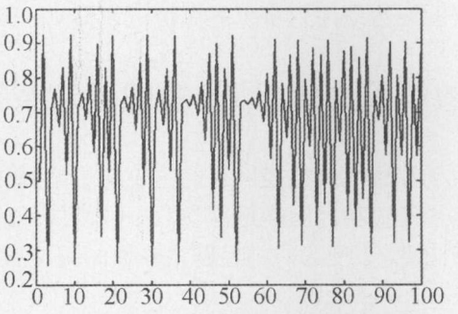

图 2　$x=0.5;\lambda=3.7$ 时的混沌数据序列

2　混沌系统辨识

2.1　神经网络辨识

如引言中所提及,首先采用BP网络对图2给出的混沌数据序列进行辨识。采用3层结构的BP网络,分别为输入层、隐层和输出层。输入层和输出层都为一个神经元,隐层5个神经元。隐层采用对数S型传递函数,输出层采用线性传递函数。网络训练采用梯度下降法。则隐层输出 u_i 由式(2)计算:

$$u_i = \frac{1}{1+e^{-a_i}} \tag{2}$$

其中:$a_i = w_i x - b_i$,w_i 为隐层第 i 个神经元与输入神经元连接权值,b_i 为隐层第 i 个神经元阈值。网络输出 y 由下式计算:

$$y = \sum_{1}^{5} w_j u_k - b \tag{3}$$

其中:w_j 为隐层第 j 个神经元与输出神经元连接权值,b 为输出神经元阈值。误差函数 E:

$$E = \frac{1}{2}\sum_{j=1}^{m}(t_j - y_j)^2 \tag{4}$$

其中:t_j 为目标值,m 为训练样本数。

为提高BP的学习速度和可靠性,采用动量法和学习率自适应调整两种策略。网络的训练可参考相关文献,在此不再赘述。将使用BP神经网络辨识混沌数据序列的算法记为BP。

2.2　遗传神经网络辨识

根据2.1中的设计,神经网络中共有10个连接权值和6个阈值。按照隐层、输出层的顺序,采用二进制对网络权值和阈值向量依次编码。

根据群体方差来定义适应度函数:

$$F = \frac{1}{m}\sum_{j=1}^{m}(y_i - t_i)^2 \tag{5}$$

其中:m 为训练样本数,t_i 为训练信号的目标输出值,y_i 为对应输入值的计算输出值。

为了产生更好的初始种群,采用如下策略:先随机产生一定数目的个体,然后从中挑出适应度值最大的个体加入到初始种群中。这种过程不断迭代,直到初始种群中个体数达到了预先确定的规模。

采用赌轮选择机制执行遗传算法的选择功能。个体每次被选中的概率与其在群体环境中的相对适应度成正比。设群体大小为 n,个体 i 的适应度值为 f_i,则 i 被选择概率 P_i 为:

$$P_i = f_i / \sum_{i=1}^{n} f_i \tag{6}$$

采用单点交叉策略,变异采用简单的位点变异方式。在新一代群体构成中,采用如下策略:如果子代中适应度最大值小于父代中适应度最大值,则用父代中适应度最大值对应个体替换子代中适应度最大值对应个体,从而保证算法终止时得到的最后结果一定是历代出现过的最高适应度值的个体。将利用GA训练BP神经网络

再进行混沌数据序列辨识的算法记为 GA - BP。

3 蚁群算法优化 BP 辨识混沌系统

蚁群算法优化神经网络的主要思想是:蚂蚁到食物源的路径映射为神经网络权值和阈值参数的集合,则最优路径对应神经网络最优参数。通过对蚂蚁选择的路径(也即是神经网络的参数)的评价,更新路径上的外激素浓度(表现为对神经网络参数的调整)。当所有蚂蚁选择到同一路径上或算法循环到预定次数,则算法结束。

假定神经网络共有 m 个参数(包括权值和阈值),对每个参数随机初始化 n 个值,则网络参数可构成图3所示网络。

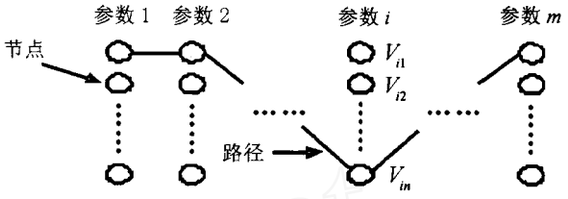

图3 神经网络参数节点与路径

图3中参数节点表示对参数的某个取值,路径表示从一个节点开始,蚂蚁选择的下一个节点。注意此处参数节点的值不仅代表神经网络某个参数的取值,也代表节点的外激素浓度。蚂蚁依次从每个参数的 n 个参数节点中选择一个节点,直至选择完毕所有参数,即构造了一条完整路径,也即确定了所有神经网络参数。用此时对应的神经网络性能来衡量该条路径的质量。蚂蚁经过节点后将在节点上留下外激素。一只蚂蚁路径构造完后,根据路径的质量更新所有节点的外激素浓度,对下一只蚂蚁构造路径施加影响(模拟蚂蚁间的信息交流)。

理论上节点的选择可以是完全随机的,但这可能需要漫长的计算时间作为代价。通常可以设计一个与问题相关的启发式函数,来引导蚁群的搜索。在这里,启发式函数被用来与外激素浓度一起,决定参数节点的选择。对于表示神经网络第 i 个参数的第 j 个取值的参数节点,定义启发式函数值为与该参数对应的神经元节点的输出,即:

$$\eta_{ij} = o_i \tag{7}$$

这样,采用赌轮选择机制,对于还没有出现在路径中的参数节点 V_{ij} 被选择的概率 $P_{ij}(t)$ 由式(8)计算:

$$P_{ij}(t) = \frac{\tau_{ij}(t)\eta_{ij}(t)}{\sum_{h}\sum_{j=1}^{n}\tau_{hj}(t)\eta_{hj}(t)} \tag{8}$$

其中 h 表示在 t 时刻尚未出现在路径中的参数序号集合。注意蚂蚁并非按照1到 m 的顺序选择参数节点,比如在选择了第一个参数的某个参数节点之后,蚂蚁可以选择2到 m 参数中的任一参数的某个参数节点。选择出的参数节点将被加入到路径中去。

路径构造完成之后,按照式(9)更新所有出现在路径中的参数节点 V_{ij} 的外激素浓度:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) \times (Q/E) \tag{9}$$

其中 Q 为常数,用于控制外激素调整速度;E 是神经网络使用路径对应参数的输出误差。可以看出,E 越小,则路径中包含的参数节点外激素增加就越多。由于对于所有参数节点,其外激素挥发率都是一样的,并且所有节点的外激素浓度还要经过下面的归一化过程,所以这里可以简单的不考虑外激素挥发。

再将所有参数节点的外激素浓度进行归一化处理:

$$\tau_{ij}(t+1) = \frac{\tau_{ij}(t)}{\sum_{i=1}^{m}\sum_{j=1}^{n_i}\tau_{ij}(t)} \tag{10}$$

这样,包含在路径中的参数节点的外激素浓度增加了,而没有包含在路径中的路径节点的外激素浓度减小了。这也体现了外激素挥发的过程。

已有研究表明[14],蚁群算法需要较长的搜索时间。主要原因是各个蚂蚁的运动是随机的,在进化的初始阶段,各个路径上的外激素浓度相差不明显,只有经过较长一段时间后,才能使得较好路径上的外激素浓度明显高于其他路径上的外激素浓度,而这个过程一般需要较长的时间。因此在进化过程中引入变异算子,对构造完成的路径,进行单点变异,即随机选择一个变异位置(也即某个参数节点,用这个参数节点对应参数

的另一个参数节点取代原有参数节点从而构成新的路径,如果新路径对应的神经网络性能大于原有路径对应的神经网络性能,则执行变异,否则取消变异。重复这个过程 p 次($p < m/2$)。这其实是一种简单局部搜索策略,操作简单且只需很短时间,但却能够扩大搜索范围,提高解的质量,从而加快算法收敛速度。由以上讨论,给出用蚁群算法优化神经网络的算法,记为 ACO - BP。

```
Definition:
    N: number of ants; Pm: mutation rate;
    M: maximal number of iteration;
    Aroute: a route created by an ant;
    NSroutes: number of same routes;
    MNSroutes: maximal number of same routes;
    BMNSroutes: boolean variable;
Algorithm:
While the number of iteration < M do
Begin
    int i = 0; NSroutes = 0; BMNSroutes = false;
    set RouteList to empty;
    Repeat
      i = i + 1;
      Aroute = ConstructRule(i);//蚂蚁构造路径;
      if mutation //controlled by mutation rate;
        Mutation(Aroute);
      if Aroute in RouteList
        NSroutes + + ;
        if NSroutes > = MNSroutes
          BMNSroute = true, exit;
        else
        Add Aroute to RouteList;
      end;
        UpdatePheromone(Aroute);//更新外激素
    Until i > N;
    if BMNSroutes exit;
end;
return the best route in RouteList。
```

4　实　验

对 1 节给出的当 $x = 0.5$;$\lambda = 3.7$ 时由 Logistic 映射所产生的混沌数据序列,分别应用 BP、GA - BP 和 ACO - BP 进行辨识。其中 BP 中参数设置为:最大学习次数为 5 000,最小误差为 0.01,学习速率增加比率为 1.05,减少比率为 0.7,动量常数为 0.9,最大误差比率为 1.04;GA - BP 中参数设定为:染色体长度为 1 000,种群规模为 80,最大进化代数为 500,交叉率为 0.9,变异率为 0.01,遗传代沟为 0.9;ACO - BP 中参数设置为:蚂蚁数为 20,变异率为 0.01,所有连接权值都初始为[-1,1]区间的某个随机量,阈值初始值为 1,为每个参数初始化 10 个值,最大相同路径数为 16,Q 为 1,最大循环次数为 500。实验环境为 P4 3.0G,1G 内存。图 4、图 5 和图 6 分别给出对大小为 20、40、50 的数据序列的辨识对比结果,其中长划线 - 点为 BP 结果,长划线为 GA - BP 结果,实线为 ACO - BP 结果。

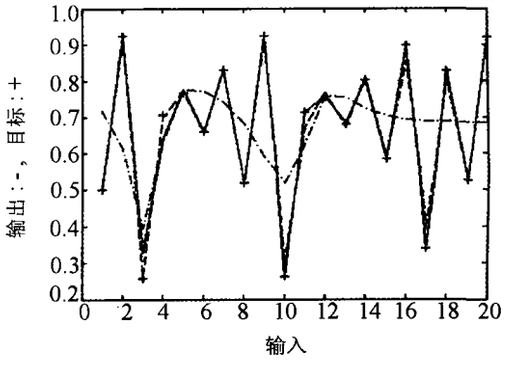

图 4　数据序列大小为 20 时各算法辨识对比结果

表1给出各算法的误差比较(取各算法独立运行10次最好结果),表2给出各算法CPU时间比较(取各算法独立运行10次平均结果)。

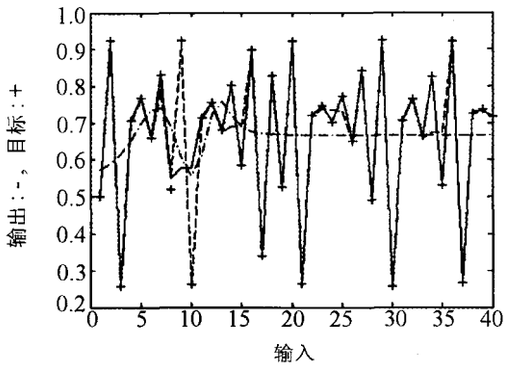

图5 数据序列大小为40时各算法辨识对比结果

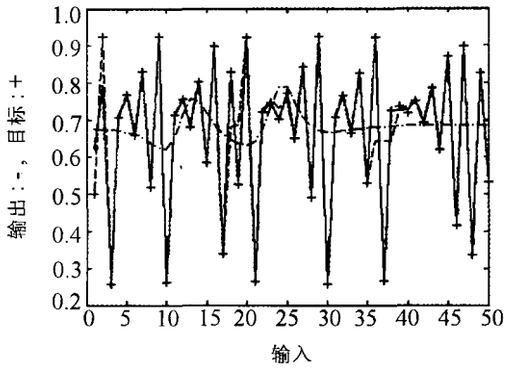

图6 数据序列大小为50时各算法辨识对比结果

表1 各算法的误差比较

算法	数据序列		
	20	40	50
BP	0.737 725	1.556 46	1.920 39
GA-BP	0.009 664 38	0.109 332	0.172 74
ACO-BP	0.000 999 35	0.009 999 56	0.009 999 63

表2 各算法CPU时间比较 s

算法	数据序列		
	20	40	50
BP	8.84	9.42	9.52
GA-BP	6.89	6.22	8.84
ACO-BP	8.87	10.66	11.23

从实验中可以看出,当混沌数据序列较小时,GA-BP和ACO-BP的求解质量和运行时间差不多,两者的求解质量都显著优于BP;当混沌数据序列较大时,ACO-BP性能显著优于GA-BP,而后者又显著优于BP,运行时间则ACO-BP较长,但考虑到求解精度,这是值得的。因此ACO-BP更适宜用于混沌数据序列的辨识;而简单结构的BP则不论数据序列大小都不适宜用于混沌数据序列的辨识,此时可能需要增加采用更加复杂的网络结构或增加隐层节点数来提高BP的辨识性能,但这将使得CPU时间大大增加。

5 结 论

混沌系统辨识一直相当困难。分别采用遗传算法和蚁群算法训练BP神经网络,并用于辨识Logistic映射产生的混沌数据序列。分别应用BP、GA-BP和ACO-BP进行辨识的实验结果表明:与BP和GA-BP相比,在不明显增加执行时间的基础上,ACO-BP寻求最优解的质量有显著提高,可有效用于混沌系统的辨识。下一步研究包括在ACO-BP中引入更好的启发式函数和局部搜索策略以缩短其搜索时间。

参考文献:

[1] LORENZ E N. Deterministic non-periodic flow [J]. Journal of Atmospheric Sciences, 1963, 20:130-141

[2] LI T Y, YORKE J A. Period three implies chaos [J]. Amer Math Monthly, 1975, 82(10):985-992

[3] RESBAND S N. Chaotic Dynamics of Nonlinear Systems [J]. A Wiley-Interscience Publication, 1989(9):118-120

[4] HORNIK K, STINCHCOMBE M, WHITE H. Multilayer Feed-forward networks are universalapproximators [J]. Neural Networks, 1989(5):359-366

[5] POZNYAK A S, YU W, SANCHEZ E N. Identification and control of unknown chaotic systems dynamic neural networks [J].

Fundamental Theory Applications, 1999, 46(12): 1491 - 1495

[6] GUERRA FA, DOS S, COELHO L. Radial Basis Neural Network Learning Based on Particle Swarm Optimization to Multistep Prediction of Chaotic Lorenz's System [J]. IEEE Proceedings of the Fifth International Conference on Hybrid Intelligent Systems, 2005(4):521 - 523

[7] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back propagating errors [J]. Nature, 1986, 323(11):533 - 536

[8] SEXTON R S, DORSEY R E. Reliable classification using neural networks: a genetic algorithm and backpropagation comparison [J]. Decision Support Systems, 2000, 30(1):11 - 22

[9] YANG J M, KAO C Y. A robust evolutionary algorithm for training neural networks [J]. Neural Computing and Application, 2001, 10(3):214 - 230

[10] FRANCBINI M. Use of a genetic algorithm combined with a local search method for the automatic calibration of conceptual rainfall - runoff models [J]. Hydrological Science Journal, 1996, 41(1):21 - 39

[11] ALBA E, LEGUIZAMON G, ORDONEZ G. Analyzing the behavior of parallel ant colony systems for large instances of the task scheduling problem [J]. 19th IEEE International Proceedings of Parallel and Distributed Processing Symposium, 2005(3): 14 - 18

[12] ZECCHIN AC, SIMPSON A R, MAIER H R, et al. Parametric study for an ant algorithm applied to water distribution system optimization [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(2):175 - 191

[13] HO S L, SHIYOU Y, WONG H C, et al. An improved ant colony optimization algorithm and its application to electromagnetic devices designs [J]. IEEE Transactions on Magnetics, 2005, 41(5):1764 - 1767

[14] PARPINELLI R S, LOPES H S, FREITAS A A. Data mining with an ant colony optimization algorithm [J]. IEEE Transactions on Evolutionary Computation, 2002,6(4):321 - 332

Training BP neural networks with ACO for identification of chaotic systems

CHENG Wei

(School of Automating, Chongqing University, Chongqing 400044, China)

Abstract: BP is the most commonly used artificial neural networks, but it suffers from extensive computations, relatively slow convergence speed and other possible weaknesses for complex problems. Genetic Algorithm (GA) has been successfully used to train neural networks, but often with the result of exponential computational complexities and hard implementation. Hence Ant Colony Optimization (ACO) is used to train BP in the paper. The efficiency of BP trained with ACO is compared with those of BP and BP trained with GA based on the identification of the same chaotic system. Comparison based on the searching precision and convergence speed of each method show that BP trained with ACO is dominant and effective to identify chaotic system.

Key words: neural networks; chaotic system; ant colony optimization; system identification

责任编辑:田　静