

粒子群优化算法训练神经网络辨识混沌系统

吴正龙,王 奇,刘开刚

WU Zheng-long, WANG Qi, LIU Kai-gang

解放军炮兵学院 一系,合肥 230031

Artillery Academy of PLA, Hefei 230031, China

E-mail: zhenglongw@yahoo.com.cn

WU Zheng-long, WANG Qi, LIU Kai-gang. Training neural networks with PSO for identification of chaotic systems. *Computer Engineering and Applications*, 2008, 44(14): 76-79.

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 Particle Swarm Optimization (PSO) is used to train BP in the paper. The efficiency of BP trained with PSO is compared with those of BP and BP trained with GA based on the identification of chaotic system. Comparison based on the searching precision and convergence speed of each method shows that BP trained with PSO is dominant and effective to identify chaotic system.

Key words: neural networks; chaotic system; particle swarm optimization; system identification

摘 要: 提出利用粒子群优化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,利用粒子群优化算法训练神经网络进行混沌系统辨识,在不明显增加执行时间的基础上,寻求最优解的质量有显著提高,并且原理简单,容易实现,可有效用于混沌系统的辨识。

关键词: 神经网络;混沌系统;粒子群优化算法;系统辨识

DOI: 10.3778/j.issn.1002-8331.2008.14.021 文章编号: 1002-8331(2008)14-0076-04 文献标识码: A 中图分类号: TP183

1 引言

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 传递函数的三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]。

粒子群优化算法(Particle Swarm Optimization, PSO)是一种基于群体智能理论(Swarm Intelligence)的优化算法,由 Eberhart 和 Kennedy 于 1995 年提出^[11],源于对鸟群捕食的行为研究,基本思想是通过群体中个体之间的协作和信息共享来

进行优化搜索。PSO 的优势在于实现简单,而且没有许多参数的调节。目前已被广泛应用于搜索和预测^[12,13]等复杂问题中。

本文采用 PSO 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,PSO 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,PSO 也具有操作简单的优点。

2 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。

3 混沌系统辨识

3.1 神经网络辨识

首先采用 BP 网络对图 2 给出的混沌数据序列进行辨识。

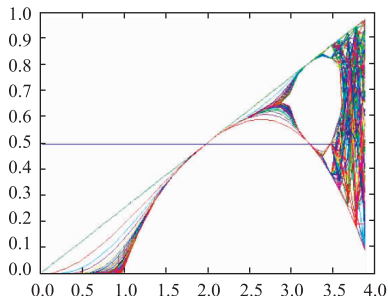
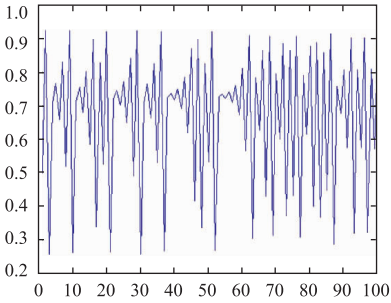


图1 Logistic映射的分岔过程

图2 $x=0.5; \lambda=3.7$ 时的混沌数据序列

采用三层结构的BP网络,分别为输入层、隐层和输出层。输入层和输出层都为—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:

$$u_i = \frac{1}{1 + e^{-a_i}} \quad (2)$$

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:

$$y = \sum_{j=1}^5 w_j u_k - b \quad (3)$$

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 :

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

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,这里不再赘述。将使用BP神经网络辨识混沌数据序列的算法记为BP。

3.2 遗传神经网络辨识

根据3.1节中的设计,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:

$$E = \frac{1}{m} \sum_{j=1}^m (y_j - t_j)^2 \quad (5)$$

其中 m 为训练样本数, t_j 为训练信号的目标输出值, y_j 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:

$$P_i = \frac{f_i}{\sum_{i=1}^n f_i} \quad (6)$$

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。

4 粒子群优化神经网络辨识混沌系统

4.1 粒子群优化算法

对于 n 维问题,定义 n 维空间的粒子群,其中粒子 i 的位置表示为矢量 $X_i = (x_1, x_2, \dots, x_n)$,飞行速度表示为矢量 $V_i = (v_1, v_2, \dots, v_n)$ 。每个粒子都有一个由目标函数决定的适应值。PSO初始化为—群随机粒子(随机解),然后通过迭代找到最优解。每个粒子状态的更新通过在每一次迭代中对两个“极值” $pbest$ 和 $gbest$ 的跟踪来进行,其中 $pbest$ 是该粒子到目前时刻为止发现的最好位置($pbest$),这可以看作是粒子自身的经验; $gbest$ 是整个粒子群到目前时刻为止发现的最好位置($gbest$ 是 $pbest$ 中的最好值),这可以看作是粒子同伴的经验。粒子就是通过自身的经验和同伴的经验来决定下一步的运动。粒子 i 状态的更新通过下式来进行^[14]:

$$V_{it} = \omega_t * V_{it} + c_1 * rand() * (pbest_i - X_{it}) + c_2 * rand() * (gbest_t - X_{it}) \quad (7)$$

$$X_{it} = X_{it} + V_{it}$$

上式中 $i=1, 2, \dots, n$, n 是群体中粒子总数; V_{it} 是粒子 i 在 t 时刻的速度; $rand()$ 产生介于(0, 1)之间的随机数; X_{it} 是粒子 i 在 t 时刻的位置; c_1 和 c_2 是学习因子,通常取 $c_1 = c_2 = 2$;在每一维,粒子都有一个最大限制速度 V_{max} ($V_{max} > 0$),如果某一维的速度超过设定的 V_{max} ,那么这一维的速度就被限定为 V_{max} ; ω_t 是 t 时刻粒子群的惯性因子,为一非负值,且有:

$$\omega_t = \frac{(\omega_{ini} - \omega_{end}) * (G_k - g)}{G_k} + \omega_{end} \quad (8)$$

其中 G_k 为最大进化代数, ω_{ini} 为初始惯性权值, ω_{end} 为迭代至最大代数时惯性权值。一般取 $\omega_{ini} = 0.9$, $\omega_{end} = 0.4$ 。

PSO算法的一般流程:

步骤1 初始化群体规模为 m 的粒子群,随机给定每个粒子的位置和速度;

步骤2 计算每个粒子的适应度;

步骤3 对每个粒子,如果其适应值好于其经过的最好位置 $pbest$,则用适应值取代 $pbest$;

步骤4 将所有粒子中最好的 $pbest$ 作为当前的最好位置 $gbest$;

步骤5 根据式(7)更新粒子状态,包括速度和位置;

步骤6 未达到结束条件,转步骤2;否则结束。

迭代终止条件根据具体问题一般选为最大迭代次数 G_k 或/和粒子群迄今为止搜索到的最优位置满足预定最小适应阈值。

4.2 粒子群优化神经网络

根据 4.1 节讨论,结合 2.1 节中设定的神经网络结构,定义粒子状态由两部分描述,分别是神经元连接权值的速度和位置,神经元阈值的速度和位置。为与粒子群算法中的惯性因子 ω 记号相区别,用 ξ 表示神经元的连接权值。

神经元连接权值和阈值速度迭代按下式进行:

$$V_{u\xi} = \omega_i * V_{u\xi} + c_1 * rand() * (pbest_i - \xi_u) + c_2 * rand() * (gbest_i - \xi_u) \tag{9}$$

$$V_{u\theta} = \omega_i * V_{u\theta} + c_1 * rand() * (pbest_i - \theta_u) + c_2 * rand() * (gbest_i - \theta_u)$$

神经元连接权值和阈值变量位置的迭代按下式进行:

$$\xi_u = \xi_u + V_{u\xi} \tag{10}$$

$$\theta_u = \theta_u + V_{u\theta}$$

粒子的适应度函数仍然如 2.2 节中式(5)定义。将利用粒子群算法训练 BP 神经网络再进行混沌数据序列辨识的算法记为 PSO-BP。

5 实验

对第 2 章给出的当 $x = 0.5; \lambda = 3.7$ 时由 Logistic 映射所产生的混沌数据序列,分别应用 BP、GA-BP 和 PSO-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;PSO-BP 中参数设置为:粒子群规模为 40,初始惯性权重为 0.9 并随迭代次数递减为 0.4, $c_1 = c_2 = 2$,神经网络连接权值为 $[-1, 1]$ 区间变量,神经元阈值都初始化为 1。实验环境为 P4 3.0 G, 1 G 内存。图 3、图 4 和图 5 分别给出对大小为 20、40、50 的数据序列的辨识对比结果(取各算法独立运行 10 次最好结果),其中长划线-点为 BP 结果,长划线为 GA-BP 结果,实线为 PSO-BP 结果。表 1 给出各算法的误差比较,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)。

设定预测问题为根据混沌数据序列中前 10 个数据预测下一个数据。表 3 给出了训练样本为 50 而预测长度分别为 10、20 和 30(即根据 50 个混沌数据学习后对随后的 10、20 和 30 个数据进行预测)时各算法的预测精度(用各数据预测误差均值表示,取各算法独立运行 10 次平均结果);表 4 给出了训练样本为 80 时各算法的预测精度;表 5 给出了训练样本为 100 时各算法预测精度。

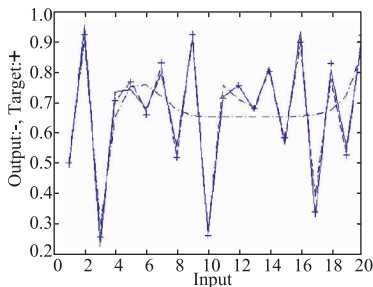


图 3 数据序列大小为 20 时各算法辨识对比结果

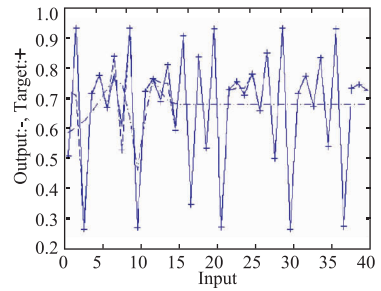


图 4 数据序列大小为 40 时各算法辨识对比结果

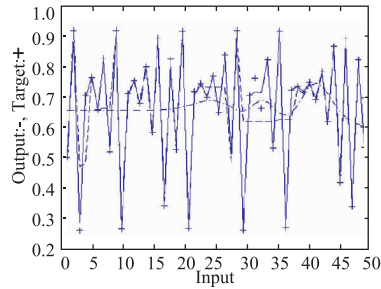


图 5 数据序列大小为 50 时各算法辨识对比结果

表 1 各算法的误差比较

算法	数据序列		
	20	40	50
BP	0.737 725 00	1.556 460 00	1.920 390 00
GA-BP	0.009 664 38	0.109 332 00	0.172 740 00
PSO-BP	0.000 999 35	0.009 999 56	0.009 969 63

表 2 各算法 CPU 时间比较

算法	数据序列		
	20	40	50
BP	8.84	9.42	9.52
GA-BP	6.89	6.22	8.84
PSO-BP	0.87	7.66	9.23

表 3 训练样本为 50 时各算法的预测精度比较

算法	数据序列		
	10	20	30
BP	0.096 5	0.471 0	0.569 1
GA-BP	0.037 5	0.038 7	0.048 8
PSO-BP	0.032 1	0.034 1	0.046 9

表 4 训练样本为 80 时各算法的预测精度比较

算法	数据序列		
	10	20	30
BP	0.079 8	0.235 2	0.361 2
GA-BP	0.021 4	0.021 5	0.022 0
PSO-BP	0.012 7	0.017 4	0.018 9

表 5 训练样本为 100 时各算法的预测精度比较

算法	数据序列		
	10	20	30
BP	0.061 1	0.214 2	0.275 3
GA-BP	0.017 3	0.018 3	0.021 4
PSO-BP	0.015 9	0.016 6	0.019 1

从实验中可以看出,当混沌数据序列较小时,GA-BP 和 PSO-BP 性能差不多,但 PSO-BP 运行时间显著小于 GA-BP,而两者性能都显著优于 BP;当混沌数据序列较大时,PSO-BP 性能显著优于 GA-BP,而后者又显著优于 BP,运行时间则三者相差不大。GA-BP 和 PSO-BP 都适宜用于混沌数据序列的辨识,后者性能在序列较小时与前者相差不大,而在序列较大时显著优于前者;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。综合来看,PSO 更适宜训练神经网络用于混沌系统的辨识。

6 结论

混沌系统辨识一直相当困难。本文分别采用遗传算法和粒子群优化算法训练 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 PSO-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,PSO-BP 寻求最优解的质量有显著提高,并且原理简单,容易实现,可有效的用于混沌系统的辨识。下一步将应用 PSO-BP 对其他混沌系统进行辨识。

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 [M]. [S. l.]: A Wiley-Interscience Publication, 1989.
- [4] Hornik K, Stinchcombe M, White H. Multilayer feed-forward networks are universal approximators [J]. Neural Networks, 1989, 2(5): 359-366.

- [5] Poznyak A S, Yu W, Sanchez E N. Identification and control of unknown chaotic systems dynamic neural networks [J]. IEEE Transactions on Circuits and Systems I: Fundamental Theory Applications, 1999, 46(12): 1491-1495.
- [6] Guerra F A, dos S Coelho L. Radial basis neural network learning based on particle swarm optimization to multistep prediction of chaotic Lorenz's system [C]// IEEE Proceedings of the Fifth International Conference on Hybrid Intelligent Systems (HIS'05), 2005: 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] Francini 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] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proceedings of the IEEE International Conference on Neural Networks, 1995: 1942-1948.
- [12] Nenortait J, Simutis R. Adapting particle swarm optimization to stock markets [C]// Proceedings of the 2005 5th International Conference on Intelligent Systems Design and Applications (ISDA'05), 2005: 520-525.
- [13] Franken N, Engelbrecht A P. Particle swarm optimization approaches to coevolve strategies for the iterated prisoner's dilemma [J]. IEEE Transactions on Evolutionary Computation, 2005, 9(6): 562-579.
- [14] Shi Y H, Eberhart R C. Empirical study of particle swarm optimization [C]// Proceedings of IEEE International Congress on Evolutionary Computation, 1999: 6-9.

(上接 66 页)

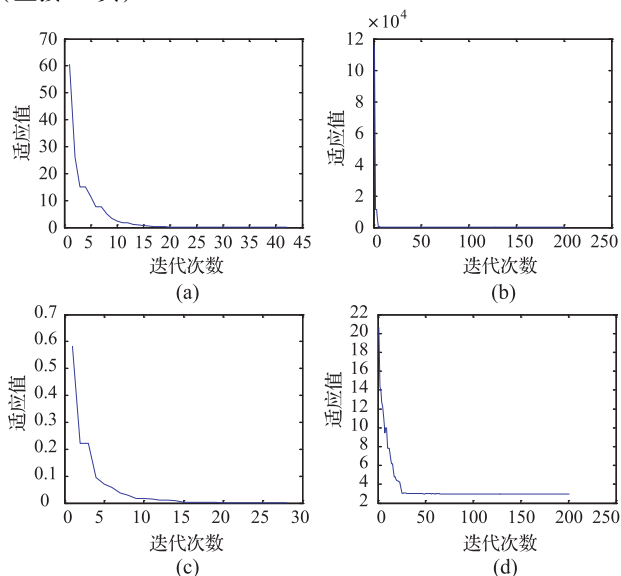


图3 收缩因子 χ +遗传操作的 PSO 算法适应值变化图

优。这为求解其他问题提出了有益的启示。

参考文献:

- [1] Shi Y H, Eberhart R C. Parameter selection in particle swarm optimization [C]// Evolutionary Programming VII; Proc EP98. New York: Springer-Verlag, 1998: 591-600.
- [2] Eberhart R C, Shi Y H. Comparing inertia weights and constriction factors in particle swarm optimization [C]// Proceedings of the IEEE Congress on Evolutionary Computation, San Diego, CA, 2000: 84-88.
- [3] Shi Y H, Eberhart R C. Empirical study of particle swarm optimization [C]// Proceedings of the 1999 Congress on Evolutionary Computation, Washington D C. Piscataway, NJ: IEEE Service Center, 1999: 1945-1950.
- [4] 高鹰. 具有遗传特性的粒子群优化算法及在非线形盲分离中的应用 [J]. 广州大学学报: 自然科学版, 2006, 5(5): 49-53.
- [5] Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm Optimization [C]// Proceedings of the 1999 Congress on Evolutionary Computation, Piscataway, Washington D C. NJ: IEEE Service Center, 1999: 1951-1957.