

文章编号: 1001-0920(2013)09-1393-06

一种基于混沌遗传与粒子群混合优化的 过程神经网络训练算法

许少华^{1,2}, 何新贵²

(1. 东北石油大学 计算机与信息技术学院, 黑龙江 大庆 163318; 2. 北京大学 信息科学技术学院, 北京 100871)

摘要: 针对时变输入/输出过程神经网络的训练问题, 提出一种基于混沌遗传与带有动态惯性因子的粒子群优化相结合的学习方法. 综合利用粒子群算法的经验记忆、信息共享和混沌遗传算法的混沌轨道遍历搜索性质, 基于PNN训练目标函数, 构建两种算法相混合的进化寻优机制, 通过适应度评估和优化效率分析自适应调节混沌遗传与粒子群算法的切换, 实现网络参数在可行解空间的全局优化求解. 实验结果表明, 该算法较大提高了PNN的训练效率.

关键词: 过程神经网络; 训练算法; 混沌遗传算法; 粒子群算法; 混合优化策略

中图分类号: TP183

文献标志码: A

A training algorithm of process neural networks based on CGA combined with PSO

XU Shao-hua^{1,2}, HE Xin-gui²

(1. School of Computer and Information Technology, Northeast Petroleum University, Daqing 163318, China; 2. School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China. Correspondent: XU Shao-hua, E-mail: xush62@163.com)

Abstract: Aiming at the training problem of time-varying input-output process neural networks(PNN), a learning algorithm based on chaos genetic algorithm(CGA) combined with particle swarm optimization(PSO) whose inertial factor is dynamic is proposed in the paper. With the application of the experience memory and sharing information of PSO algorithm, and chaos track traverse searching of CGA, the hybrid evolutionary optimization mechanism of CGA and PSO algorithm is built based on the PNN's training objective function. The adaptive switching of two algorithms is implemented through estimating the fitness and optimization efficiency, and the global optimal solution is obtained in feasible solution space. Experimental results show that the algorithm considerably improves the training efficiency of PNN.

Key words: process neural networks; training algorithm; chaos genetic algorithm; particle swarm optimization; hybrid optimization strategy

0 引言

训练算法是影响人工神经网络系统建模和模型泛化能力的重要因素之一. 对于神经网络的训练问题, 已有许多成熟的学习方法, 其中应用最为广泛和有效的是基于梯度下降的误差反传播学习算法及其各种改进形式, 以及近几年提出的与进化优化策略相结合的训练方法^[1-5]. 过程神经网络(PNN)是一种新型的动态神经网络模型^[6-7], 具有对多元动态信息的时空维聚合运算机制和直接对非线性系统输入/输出关系的建模能力, 对于解决与时间过程有关的众

多问题在机制上有着良好的适应性. 由于PNN的输入/输出信号和连接权等都可以是时间函数, 过程神经元的时延效应累积算子是一种连续时间的泛函映像, 使得PNN的映射机制和学习过程与传统神经网络有着很大的不同. 目前, PNN训练的主要算法是基于函数基展开结合最小均方算法(LMS)^[8]和基于数值积分^[9]的训练方法, 但由于网络训练目标函数中所包含的待确定参数众多且参数之间的约束具有较大自由度, 影响了算法的学习效率和稳定性. 近几年, 基于进化计算的优化技术被广泛应用于求解各种工程领域问题中的最优解, 在复杂函数计算、过程工艺

收稿日期: 2012-04-11; 修回日期: 2012-07-30.

基金项目: 国家自然科学基金项目(61170132); 中国石油科技创新基金项目(2010D-5006-0302).

作者简介: 许少华(1962-), 男, 教授, 博士生导师, 从事神经网络、优化算法等研究; 何新贵(1938-), 男, 教授, 博士生导师, 中国工程院院士, 从事模糊逻辑、神经网络、进化计算、数据库理论等研究.

优化、系统辨识及控制等领域得到了成功应用. 混沌遗传算法 (CGA) 是一种融合遗传算法进化机制和混沌搜索策略各自优势的一种智能算法, 具有群体搜索、轨道遍历和全局优化等性质^[10-11], 例如在多目标约束下的系统寻优、高维空间复杂极值问题求解等方面, 其在算法原理和优化策略上表现出了优势, 但存在全局信息共享程度低等缺陷; 粒子群优化 (PSO) 是对自然界中鸟类捕食现象的模拟而建立起来的一种基于群体智能的演化计算技术, 具有记忆粒子最佳位置和粒子间信息共享的机制以及较强的鲁棒性, 但收敛速度较慢^[12-14]. CGA 和 PSO 具有不同的搜索和进化机制, 且两种算法在某些寻优策略下可相互取长补短, 因此, 如果能够寻找一种 CGA 与 PSO 相结合的优化策略和混合算法, 则可提高对复杂优化问题求解的能力和适应性.

本文针对输入/输出均为时变函数 PNN 的学习问题, 提出了一种基于混沌遗传与带有动态惯性因子的粒子群优化^[15]相结合的训练算法. 首先采用基于函数正交基展开的方法, 将针对网络连接权函数和阈值函数的学习问题转化为对相应函数基展开系数的训练问题, 并把过程神经元的时间聚合算子转换为数值积分运算, 根据最小均方误差准则构建网络训练目标函数, 然后针对目标函数进行全局优化求解. 通过建立可行解适应度评估和算法优化效率分析准则, 动态监测算法运行状态, 自适应调度 CGA 与 PSO 的交替迭代, 发挥 PSO 和 CGA 各自不同的算法特性, 改善和提高 PNN 在训练中的搜索性质以及对目标函数的全局优化效率. 本文给出了 PNN 基于 CGA-PSO 的混合优化训练步骤, 并将其应用于非线性时变系统建模和油田开发并组注采过程模拟研究, 取得了较好的应用效果.

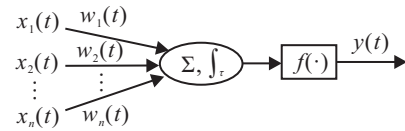
1 过程神经网络模型

1.1 时变输入/输出过程神经元

考虑实际非线性动态系统信息变换的一般情况, 过程神经元设计为一种输入/输出均为连续时间函数的模型, 结构如图 1 所示. 其中: $x_1(t), x_2(t), \dots, x_n(t)$ 为过程神经元的时变输入信号; $w_1(t), w_2(t), \dots, w_n(t)$ 为相应的连接权函数; Σ 为过程神经元空间聚合算子, \int_{τ} 为过程神经元时间效应累积算子, $f(\cdot)$ 为激励函数, $y(t)$ 为输出函数. 若 \int_{τ} 取为含时间变参积分, Σ 为多输入信号加权求和, 则过程神经元输入与输出之间的映射关系为

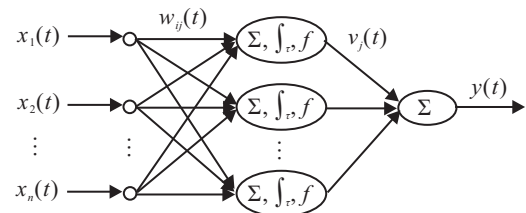
$$y(t) = f\left(\int_0^t \sum_{i=1}^n w_i(\tau)x_i(\tau)d\tau - \theta(t)\right), \quad (1)$$

$\theta(t)$ 为过程神经元的激励阈值函数.



1.2 时变输入/输出过程神经网络

将采用若干由式 (1) 定义的过程神经元和其他类型的神经元, 按照一定的拓扑结构和信息传递关系构成输入输出均为时变函数的过程神经网络. 考虑含有一个过程神经元隐层、单输出的前馈网络模型, 拓扑结构如图 2 所示. 其中: $w_{ij}(t) (i = 1, 2, \dots, n, j = 1, 2, \dots, m)$ 为输入层节点与隐层节点的连接权函数, $v_j(t) (j = 1, 2, \dots, m)$ 为隐层节点到输出节点的连接权函数, $y(t)$ 为网络输出.



由图 2 可以看出, 过程神经网络输入和输出之间的映射关系为

$$y(t) = \sum_{j=1}^m v_j(t) f\left(\int_0^t \left(\sum_{i=1}^n w_{ij}(\tau)x_i(\tau)\right) d\tau - \theta_j(t)\right), \quad t \in [0, T]. \quad (2)$$

其中: $[0, T]$ 为时变信号输入过程区间, f 为激励函数, $\theta_j(t)$ 为阈值函数.

1.3 网络训练目标函数

设时变输入/输出过程神经网络的输入空间 $U \in (C[0, T])^n (b_1(t), b_2(t), \dots, b_L(t))$ 为 $C[0, T]$ 中一组满足 U 中输入函数拟合精度要求的标准正交基函数, $(x_1(t), x_2(t), \dots, x_n(t))$ 为 U 中的任意函数, 则 $x_i(t)$ 的基函数展开形式可表示为

$$x_i(t) = \sum_{l=1}^L a_{il} b_l(t), \quad i = 1, 2, \dots, n, \quad (3)$$

其中 L 为满足基函数展开拟合精度要求的基函数项数.

将式 (2) 中权函数 $w_{ij}(t)$, $v_j(t)$ 和阈值函数 $\theta_j(t)$ 也采用 $b_1(t), b_2(t), \dots, b_L(t)$ 的展开形式表示, 即

$$w_{ij}(t) = \sum_{l=1}^L w_{ij}^{(l)} b_l(t),$$

$$v_j(t) = \sum_{l=1}^L v_j^{(l)} b_l(t),$$

$$\theta_j(t) = \sum_{l=1}^L \theta_j^{(l)} b_l(t),$$

$$i = 1, 2, \dots, n, j = 1, 2, \dots, m, \quad (4)$$

其中 $w_{ij}^{(l)}, v_j^{(l)}, \theta_j^{(l)}$ 分别为 $w_{ij}(t), v_j(t), \theta_j(t)$ 相对于 $b_l(t)$ 的基展开系数。

将式(4)代入(2), 整理得

$$y(t) = \sum_{j=1}^m \left(\sum_{l=1}^L v_j^{(l)} b_l(t) \right) \cdot$$

$$f \left(\sum_{i=1}^n \sum_{l=1}^L \sum_{s=1}^L a_{il} w_{ij}^{(l)} \int_0^t b_l(\tau) b_s(\tau) d\tau - \right.$$

$$\left. \sum_{l=1}^L \theta_j^{(l)} b_l(t) \right). \quad (5)$$

给定 K 个学习样本 $(x_{k1}(t), x_{k2}(t), \dots, x_{kn}(t), d_k(t))$, $k = 1, 2, \dots, K$, 其中 $d_k(t)$ 为对应输入 $x_{k1}(t), x_{k2}(t), \dots, x_{kn}(t)$ 网络的期望输出函数. 设对应第 k 个样本输入网络的实际输出为 $y_k(t)$, 根据最小均方误差准则, 网络训练误差函数定义为

$$E = \sum_{k=1}^K \|y_k(t) - d_k(t)\| =$$

$$\sum_{k=1}^K \left(\int_0^T \left(\sum_{j=1}^m \left(\sum_{l=1}^L v_j^{(l)} b_l(t) \right) \cdot \right.$$

$$f \left(\sum_{i=1}^n \sum_{l=1}^L \sum_{s=1}^L a_{il}^{(k)} w_{ij}^{(l)} \int_0^t b_l(\tau) b_s(\tau) d\tau - \right.$$

$$\left. \sum_{l=1}^L \theta_j^{(l)} b_l(t) \right) - d_k(t) \Big)^2 dt \Big)^{\frac{1}{2}}, \quad (6)$$

其中 $a_{il}^{(k)}$ 为 $x_{ki}(t)$ 在基函数展开式中对应于 $b_l(t)$ 的系数.

将过程区间 $[0, T]$ 进行适当的 P 等分, 记分割点为 t_0, t_1, \dots, t_P . 将式(3)和(4)另写为

$$x_i(t_p) = \sum_{l=1}^L a_{il} b_l(t_p), w_{ij}(t_p) = \sum_{l=1}^L w_{ij}^{(l)} b_l(t_p),$$

$$v_j(t_p) = \sum_{l=1}^L v_j^{(l)} b_l(t_p), \theta_j(t_p) = \sum_{l=1}^L \theta_j^{(l)} b_l(t_p). \quad (7)$$

将式(7)带入(6), 得到时变输入/输出PNN的训练目标函数

$$E =$$

$$\sum_{k=1}^K \left(\sum_{p=0}^P \left(\sum_{j=1}^m \left(\sum_{l=1}^L v_j^{(l)} b_l(t_p) \right) f \left(\sum_{i=1}^n \sum_{l=1}^L \sum_{s=1}^L a_{il} w_{ij}^{(l)} \cdot \right.

$$\left. \left. \int_{t_{p-1}}^{t_p} b_l(\tau) b_s(\tau) d\tau - \sum_{l=1}^L \theta_j^{(l)} b_l(t_p) \right) - d_k(t_p) \right)^2 \Delta t \Big)^{\frac{1}{2}}, \quad (8)$$$$

其中 $\Delta t = T/P$.

2 混沌遗传及粒子群算法

2.1 混沌搜索策略与混沌遗传算法

混沌是自然界中普遍存在的一种非线性现象, 具有内在随机、轨道遍历和隐含规律等性质. 混沌搜索是将混沌状态引入优化变量中, 通过施加混沌扰动, 使其在一定范围内按系统内在规律不重复地遍历所有状态^[16]. 考虑一种基于虫口模型的混沌序列生成方法, 以 Logistic 映射为发生器, 按下式产生混沌序列:

$$\delta^{j+1} = u\delta^j(1 - \delta^j), \quad (9)$$

其中 u 是混沌吸引子. 当 $u = 4$ 时, 系统进入混沌状态, 产生混沌变量 $\delta^j (j = 1, 2, \dots)$, 其值在 $[0, 1]$ 区间内变化.

混沌遗传算法^[17]是基于混沌变量运动的轨道遍历性和遗传算法的进化机制, 将混沌运动性质结合优化变量的搜索中, 对得到的混沌变量进行编码, 表示成染色体, 并将它们置于问题的环境中, 根据适者生存的原则, 进行选择、混沌交叉、混沌变异等遗传操作, 通过遗传迭代的不断进化, 最后收敛到可行解空间中一个最适合环境的个体上, 求得问题的全局最优解.

2.2 带动态惯性因子的 PSO 算法

考虑过程神经网络训练中对算法的全局和局部搜索能力, 以及对搜索范围控制的要求, 引入一种带动态惯性因子的 PSO 算法^[15]. 算法描述如下: 设在 n 维空间中, 有 m 个粒子组成一个种群, 其中第 i 个粒子的位置 X_i 、速度 V_i 、自身最优位置 P_i 及全局最优位置 P_g 分别为 $X_i = (x_{i1}, x_{i2}, \dots, x_{in}), V_i = (v_{i1}, v_{i2}, \dots, v_{in}), P_i = (p_{i1}, p_{i2}, \dots, p_{in})$ 和 $P_g = (p_{g1}, p_{g2}, \dots, p_{gn})$. 粒子状态更新策略为

$$V_i(s+1) = w(s)V_i(s) + c_1 r_1 (P_i - X_i(s)) +$$

$$c_2 r_2 (P_g - X_i(s)), \quad (10)$$

$$X_i(s+1) = X_i(s) + V_i(s+1), \quad i = 1, 2, \dots, m. \quad (11)$$

其中: $w(s)$ 是动态惯性因子, 采用线性递减权重策略进行设置, 即 $w(s) = w_{\max} - \frac{(w_{\max} - w_{\min})}{T_{\max}} \cdot s$, s 为迭代次数, T_{\max} 为最大迭代次数, w_{\max} 和 w_{\min} 分别为最大惯性权重和最小惯性权重; c_1, c_2 为常数; r_1, r_2 为 $[0, 1]$ 之间的随机数. 对种群中每个粒子应用式(10)和(11)循环迭代, 实现整个种群向全局最优解的逐步逼近.

3 基于 CGA 与 PSO 混合算法的 PNN 训练

过程神经网络的 CGA-PSO 混合优化训练策略是融合 PSO 和 CGA 两种算法的优势, 通过设置适应度评估及算法效率分析准则, 实现 PSO 与 CGA 两种

算法在控制准则下的自适应交替迭代, 达到性质互补、全局优化的目标.

3.1 混合优化算法的切换准则

综合 PNN 训练目标函数值的变化和算法运行效率两方面, 提出如下 CGA 和 PSO 两种算法的自适应切换准则: 设 θ 为切换阈值, ΔG 为代数间隔, ΔE 为经 ΔG 次迭代后目标函数值的变化量. 若 $|\Delta E|/\Delta G < \theta$, 则实施算法切换. 根据上述准则, 算法自运行开始每隔 ΔG 代判断一次并决定是否切换, 直至满足优化精度或事先设定的最大迭代步数为止.

3.2 混合优化算法的实施步骤

算法首先按照一定的种群规模进行 PSO 或 CGA 进化迭代, 同时进行适应度及算法效率评估检查. 当单一算法满足终止条件时, 在当前进化情况下, 按比例选择适应度靠前的粒子作为种子进入下一轮寻优, 同时随机选择其他粒子补充种群, 达到设定的规模, 而后调用另一算法继续进行迭代. 混合算法具体实施步骤描述如下:

Step 1: 确定种群规模 N , 随机生成初始种群 G^0 , 采用十进制数对染色体进行编码, 每条染色体上基因数为待优化的变量个数; 设置迭代间隔 ΔG , 算法切换阈值 θ , 最大迭代代数, 误差精度 ε .

Step 2: 构造适应度函数. 由于 PNN 训练为目标函数极小值优化问题, 适应度函数取为 $\text{fit} = e^{-E}$.

Step 3: 执行带动态惯性因子的 PSO 算法:

Step 3.1: PSO 初始化, $w(s)$ 计算式中的 w_{\min} 取 0.4, w_{\max} 取 0.9;

Step 3.2: 按照式 (10) 和 (11) 进行 ΔG 次循环迭代, 计算目标函数值变化量 ΔE ;

Step 3.3: 若 $|\Delta E|/\Delta G < \theta$, 则转 Step 3.4, 否则转 Step 3.2;

Step 3.4: 若满足终止条件, 则保存最优解停机; 否则, 选择适应度前 50% 的粒子, 并在可行解空间随机选择其他 50% 粒子补充到种群, 达到设定的种群规模 N , 生成新的种群 G^1 , 执行 Step 4 切换到 CGA.

Step 4: 执行 CGA 搜索:

Step 4.1: 按以下步骤进行 ΔG 次循环迭代, 计算目标函数值变化量 ΔE .

选择操作: 在 G^1 中采用转轮规则选择染色体, 每个染色体的选择概率正比于其适应度值.

混沌交叉: 两条染色体按如下方式组合: $W'_1 = \bar{\lambda}W_1 + (\bar{1} - \bar{\lambda})W_2$, $W'_2 = \bar{\lambda}W_2 + (\bar{1} - \bar{\lambda})W_1$, 其中 $\lambda_i \in (-1, 1)$ 为混沌变量. 定义交叉幅值 λ_k , 按下式确定 λ_i :

$$\lambda_i = \lambda_k \delta_i^{j+1}. \quad (12)$$

为使遍历呈现双向性, 混沌变量 δ_i^{j+1} 按下式确定:

$$\delta_i^{j+1} = 8\delta_i^j(1 - \delta_i^j) - 1. \quad (13)$$

变异操作: 设变异基因为 w_i , 变异后为 $w'_i = w_i + \beta(w_i^U - w_i)$ 或 $w'_i = w_i + \beta(w_i - w_i^L)$. 其中: w_i^U 和 w_i^L 为 w_i 上下界, $\beta \in (-1, 1)$ 为混沌变量. 在变异操作中, 首先定义一个变异幅值 $\tilde{\lambda}_k$, 然后引入混沌变量. 对于需要变异的母体, 适应度越低, 所迭加的混沌扰动应越大. 对于选出的 m 个待变异个体, 按适应度升序排序. 对于第 k 个个体, 变异幅值 $\tilde{\lambda}_k$ 可按下式选取:

$$\tilde{\lambda}_k = \tilde{\lambda}_0 \exp((m - k)/m). \quad (14)$$

其中: $\tilde{\lambda}_0$ 为控制混沌扰动大小的参数, 此时, β 转角的遍历范围为 $[-\tilde{\lambda}_k, \tilde{\lambda}_k]$.

Step 4.2: 若 $|\Delta F|/\Delta G < \theta$, 则转 Step 4.3, 否则转 Step 4.1.

Step 4.3: 若满足终止条件, 则保存最优解停机; 否则, 选择适应度前 50% 的粒子, 并在可行解空间随机选择其他 50% 粒子补充到种群, 达到设定的种群规模 N , 生成新的种群 G^2 , 执行 Step 3 切换到 PSO.

4 实验结果分析

采用 Matlab 7.1 编程进行算法软件实现, 并将其应用于非线性时变系统建模和油田开发井组注采过程模拟的研究.

4.1 在非线性动态系统建模中的应用

考虑具有如下 26 个时变输入/输出函数样本对的非线性系统过程神经网络建模问题. 样本集为: $\{t \sin((0.5 + k\tau) \cdot t\pi) + 2.0, t \cos((0.5 + k\tau) \cdot t\pi) + 2.0\}$, $k = 0, 1, \dots, 15, t \in [-2, 2], \tau \in \{0.1, 0.15\}$.

用于系统建模的时变输入/输出 PNN 结构选择为 1-5-1, 网络输入为 $t \sin((0.5 + k\tau) \cdot t\pi) + 2.0$, 对应的输出为 $t \cos((0.5 + k\tau) \times t\pi) + 2.0$. 将网络输入函数和连接权函数、阈值函数表示为项数为 16 的连续 Walsh 正交函数基的展开形式, 并按式 (3)~(9), 确定网络的训练误差函数. 分别采用 CGA-PSO 混合优化算法 ($N = 20, \Delta G = 50, \theta = 0.05$), 基于正交函数基展开训练算法 (OFBE)^[8] 和基于数值积分的学习算法 (NI)^[9] (时间粒度取为 0.1) 对网络进行训练, 误差精度设定为 0.01, 每种算法运行 10 次, 3 种算法训练结果对比如表 1 所示, 学习过程如图 3 所示. 其中: CGA-PSO 混合算法的迭代次数为 CGA 与 PSO 各自算法总的迭代次数之和, CGA 与 PSO 的平均切换次数为 3.2.

表 1 中预测误差计算公式为

$$\delta = \frac{1}{26} \sum_{k=1}^{26} \int_{-2}^2 (y_k(t) - d_k(t))^2 dt)^{\frac{1}{2}}.$$

表1 过程神经网络训练结果对比

指标	CGA-PSO	OFBE	NI
平均迭代次数	272	516	383
平均时间/s	3.587	9.719	15.0865
平均预测误差	0.011	0.026	0.017

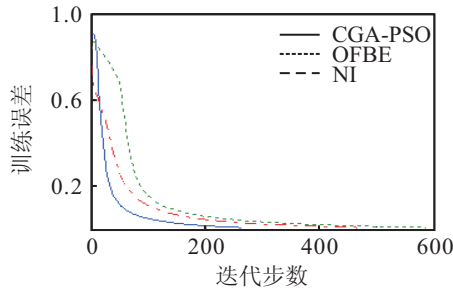


图3 CGA-PSO算法与其他算法收敛曲线比较

4.2 在油田开发井组注采过程模拟中的应用

石油开采是一个复杂的非牛顿流体运动过程. 科技人员在进行油藏开发过程模拟及预测研究时, 一般需要建立和求解复杂的油层物理及渗流方程, 各方程中所涉及的油藏参数众多, 在实际中很难确定和求准. 本文利用时变输入/输出PNN来建立开发井组注采关系的动态模型.

选取某油田开发区块一个注采井组近3年的实际生产数据, 生产变量主要包括水井注水量 $Q_w(t)$, 注水压力 $P(t)$ 和油井产油量 $Q_o(t)$, 采样间隔为1天, 得到一个包含1010组实际生产数据的时间序列 $\{x_1(i), x_2(i); y(i)\}_{i=1}^{1010}$. 部分采样数据见表2.

表2 井组注采开发生产采样数据(部分)

$Q_w/(m^3/d)$	P/MPa	$Q_o/(m^3/d)$	$Q_w/(m^3/d)$	P/MPa	$Q_o/(m^3/d)$
...	60	6.69	14
39	10.2	21.6	73	6.88	20.2
74	9.7	18.4	67	6.82	22.5
36	12.9	18.1	70	6.87	21.6
39	12.8	17.4	62	6.81	21.6
39	12.8	17.1	64	6.74	18.1
35	12.7	19.5	64	6.81	17.1
41	7.27	21.9	69	7.12	16.4
54	7.13	20.6	62	7.37	16.3
55	7.05	15.2	17	7.18	15.1
60	7	13.1

由于油田开采过程的不可重复性和注水驱替开发的延时效应, 根据油田开发理论和现场实际经验, 采用如下方式构建注采关系数据集 $\{x_1(i + 10 \cdot k \cdot \Delta T), x_2(i + 10 \cdot k \cdot \Delta T); y(i + 10 \cdot (k + 1) \cdot \Delta T)\}$, 其中: $i = 1, 2, \dots, 10, k = 0, 1, \dots, 9; \Delta T$ 为时间单元, 实际资料处理时 ΔT 取为1. 以10个采样点间隔对时间序列进行分割, 在每个分段上采用基函数个数为4的Legendre正交函数基分别对 $Q_w(t), P(t)$ 的离散数据进行拟合作为系统输入, 把后续邻接区间中10个时间点上的 $Q_o(t)$ 值进行相同基函数拟合作

为对应的系统输出. 为统一输入过程区间, 将各子区间上的Legendre拟合函数扩展定义为整个采样区间上的分段函数, 共得到100组输入/输出函数样本对, 以此作为PNN的训练集. 经实验对比分析, 网络拓扑结构确定为2-8-1. 分别用CGA-PSO混合算法 ($N = 10, \Delta G = 50, \theta = 0.01$)、正交函数基展开算法和数值积分算法对网络进行训练, 总的迭代次数均设定为1000, 每种算法运行10次, 训练结果对比如表3所示. 在CGA-PSO算法中, CGA和PSO的平均切换次数为23.6.

表3 3种算法过程神经网络训练结果对比

指标	CGA-PSO	OFBE	NI
平均训练误差	0.00172	0.0319	0.0203
平均时间/s	6.5870	17.3195	27.0865
平均预测误差	0.0275	0.0424	0.0332

上述两例实验的结果表明, 本文建立的CGA-PSO混合优化算法相比于PNN的正交函数基展开训练算法和数值积分训练算法, 在学习效率和拟合度方面有较大提高. 这是由于在整个优化过程中, CGA-PSO实现了两种进化算法的自适应切换, 当一种算法略显停滞时立即切换到另一算法, 因而有效实现了两种算法的优势互补, 达到了比较理想的结果.

5 结论

本文提出了一种基于混沌遗传与粒子群优化相结合的过程神经网络训练算法. 算法将PNN训练的泛函寻优转换为对多元函数求极值的优化问题, 综合利用PSO的经验记忆、信息共享性质以及CGA具有的混沌轨道遍历搜索和进化机制, 实现网络参数在可行解空间的全局优化求解. 实验结果表明, 该算法具有明显优于单一算法的收敛速度和逼近能力, 这对于其他机器模型的学习和复杂函数优化等问题的解决也具有一定的参考和借鉴价值.

参考文献(References)

- [1] 高海兵, 高亮, 周驰, 等. 基于粒子群优化的神经网络训练算法研究[J]. 电子学报, 2004, 32(9): 1572-1574. (Gao H B, Gao L, Zhou C, et al. Particle swarm optimization based algorithm for neural network learning[J]. Acta Electronica Sinica, 2004, 32(9): 1572-1574.)
- [2] 孔维健, 丁进良, 柴天佑. 高维多目标进化算法研究综述[J]. 控制与决策, 2010, 25(3): 321-325. (Kong W J, Ding J L, Chai T Y. Survey on large-dimensional multi-objective evolutionary algorithms[J]. Control and Decision, 2010, 25(3): 321-325.)
- [3] Estatico C. A two-steps inexact Newton method for atmospheric remote sensing[C]. IEEE Int Workshop on

- Imaging Systems and Techniques, 2004: 66-70.
- [4] Shing-Tai Pan, Shih-Chuan Chen, Shih-Hung Chiu. A new learning algorithm of neural network for identification of chaotic systems[C]. IEEE Int Conf on Systems, Man and Cybernetics. Washington DC, 2003: 1316-1321.
- [5] Xu H K. Iterative algorithms for nonlinear operators[J]. J of the London Mathematical Society, 2002, 66(4): 240-256.
- [6] 何新贵, 梁久祯. 过程神经网络的若干理论问题[J]. 中国工程科学, 2000, 2(12): 40-44.
(He X G, Liang J Z. Some theoretical issues on process neural networks[J]. Engineering Science, 2000, 2(12): 40-44.)
- [7] 许少华, 何新贵. 关于连续过程神经网络的一些理论问题[J]. 电子学报, 2006, 34(10): 1838-1841.
(Xu S H, He X G. Some theoretical issues on continuous process neural networks[J]. Acta Electronica Sinica, 2006, 34(10): 1838-1841.)
- [8] 许少华, 何新贵. 基于函数正交基展开的过程神经网络学习算法[J]. 计算机学报, 2004, 27(5): 645-649.
(Xu S H, He X G. Learning algorithms of process neural networks based on orthogonal function basis expansion[J]. Chinese J of Computers, 2004, 27(5): 645-649.)
- [9] 许少华, 王颖. 一种基于数值积分的过程神经网络训练算法[J]. 计算机科学, 2010, 38(11): 203-20.
(Xu S H, Wang Y. Training algorithm of process neural networks based on numerical integration[J]. Computer Science, 2010, 38(11): 203-20.)
- [10] 陈炳瑞, 杨成祥. 自适应混沌遗传混合算法及其参数敏感性分析[J]. 东北大学学报, 2006, 27(6): 689-692.
(Chen B R, Yang C X. Self-adapting chaos-genetic hybrid algorithm and sensitivity analysis of its parameters[J]. J of Northeastern University, 2006, 27(6): 689-692.)
- [11] 李兵, 蒋慰孙. 混沌优化方法及应用[J]. 控制理论与应用, 1997, 14(4): 613-615.
(Li B, Jiang W S. Chaos optimization method and its application[J]. Control Theory & Applications, 1997, 14(4): 613-615.)
- [12] Kennedy J, Eberhart R. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Piscataway: IEEECS, 1995: 1942-1948.
- [13] Trelea I C. The particle swarm optimization algorithms: Aonvergence analysis and parameter selection[J]. Information Proecessing Letters, 2003, 8(5): 317-325.
- [14] 申元霞, 王国胤, 曾传华. PSO 模型种群多样性与学习参数的关系研究[J]. 电子学报, 2011, 39(6): 1238-1244.
(Shen Y X, Wang G Y, Zeng C H. Study on the relationship between population diversity and learning parameters in particle swarm optimization[J]. Acta Electronica Sinica, 2011, 39(6): 1238-1244.)
- [15] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization[C]. Proc of the 2000 Congress on Evolutionary Computation. Piscataway: IEEE, 2000: 84-88.
- [16] 罗均, 李研. 具有混沌搜索策略的蜂群优化算法[J]. 控制与决策, 2010, 25(12): 1913-1916.
(Luo J, Li Y. Artificial bee colony algorithm with chaotic-search strategy[J]. Control and Decision, 2010, 25(12): 1913-1916.)
- [17] 伍铁斌, 成运, 周桃云. 基于混沌遗传算法的PID 参数优化[J]. 计算机仿真, 2009, 26(5): 202-205.
(Wu T B, Cheng Y, Zhou T Y. Optimization control of PID based on chaos genetic algorithm[J]. Computer Simulation, 2009, 26(5): 202-205.)

~~~~~

(上接第1392页)

- [11] 李强. 供应链协调下的随机比较[D]. 北京: 北京工业大学经济管理学院, 2010.  
(Li Q. Stochastic comparison of supply chain under coordination[D]. Beijing: School of Economics and Management, Beijing University of Technology, 2010.)
- [12] Karlin S, Novikoff A. Generalized convex inequalities[J]. Pacific J of Mathematics, 1963, 13(4): 1251-1279.
- [13] Whitt W. Uniform conditional variability ordering of probability distributions[J]. J of Applied Probability, 1985, 22(3): 619-633.
- [14] Stoyan D. Comparison methods for queues and other stochastic models[M]. New York: Wily, 1983.
- [15] Shaked M, Shanthikumar J G. Stochastic orders and their applications[M]. New York: Academic Press, 1994.