

基于改进的 PSO 算法的 神经网络相关性剪枝优化*

涂娟娟^{1,2}, 詹永照¹, 韩飞¹

(1. 江苏大学 计算机科学与通信工程学院, 江苏 镇江 212013; 2. 江苏科技大学 计算机科学与工程学院, 江苏 镇江 212003)

摘要: 针对传统的神经网络训练算法收敛速度慢、易陷入局部最优的问题, 提出了一种基于改进的分期变异微粒群优化算法(SMP SO)的神经网络相关性剪枝优化方法。SMP SO 在初期使适应度过低的微粒发生变异, 在后期使停滞代数过高的个体极值和全局极值发生变异, 后将 SMP SO 用于优化神经网络相关性剪枝算法。实验结果表明, 该方法与采用 BP 算法及标准 PSO 算法进行相关性剪枝相比, 在训练收敛速度、剪枝效率及分类正确率三方面都有较大提高。

关键词: 神经网络; 剪枝; 微粒群优化算法

中图分类号: TP183

文献标志码: A

文章编号: 1001-3695(2010)09-3253-03

doi:10.3969/j.issn.1001-3695.2010.09.013

Neural network correlation pruning optimization based on improved PSO algorithm

TU Juan-juan^{1,2}, ZHAN Yong-zhao¹, HAN Fei¹

(1. School of Computer Science & Telecommunication Engineering, Jiangsu University, Zhenjiang Jiangsu 212013, China; 2. School of Computer Science & Engineering, Jiangsu University of Science & Technology, Zhenjiang Jiangsu 212003, China)

Abstract: The traditional neural network training algorithm converges slowly and is easy to fall into local optimum. In response to these shortcomings, this paper proposed a neural network correlation pruning method optimized with improved staging mutation particle swarm optimization algorithm (SMP SO). SMP SO mutate particles that had too low fitness at early stage and mutate individual extreme and global extreme that stagnate in excessive iteration latterly. Then used SMP SO to optimize neural network correlation pruning algorithm. The experiment results show that neural network correlation pruning method optimized by SMP SO is more efficient than that optimized by BP and standard PSO. It has greater improvement in the convergence velocity of training, the efficiency of pruning and the accuracy of classification.

Key words: neural network; pruning; particle swarm optimization algorithm

0 引言

神经网络结构设计的关键是确定合适的隐含层节点的数目, 方法主要有构造法、剪枝法、自适应法等^[1]。相关性剪枝算法(correlation pruning algorithm, CPA)是一种较简单有效的剪枝法, 根据节点间的相关性情况进行剪枝^[2,3]。首先采用 BP 算法训练神经网络参数; 然后计算隐节点输出之间的相关性, 合并具有较大相关性的隐节点。CPA 与其他剪枝法如权衰减法、灵敏度计算法等相比, 原理较简单, 计算量较小。但其采用 BP 算法训练神经网络参数, 只是利用误差函数一阶梯度的信息确定下一步训练的方向, 因此收敛速度比较慢, 越是接近极小值点或局部极小值点时, 收敛速度越慢, 并且在训练过程中还会时常陷入局部最小点, 从而影响隐节点输出之间的相关性的计算, 导致剪枝效率降低。BP 算法的上述缺陷尤其是局部优化特性还会使其训练的神经网络的输出具有不一致性和不

可预测性, 导致分类的可靠性降低^[4]。

遗传算法(GA)采用并行搜索策略, 具有较强的全局优化特性, 有研究者将它用来训练神经网络参数, 实验证明可以提高分类正确率^[5]。但是 GA 复杂的遗传操作如选择、复制、交叉、变异, 使神经网络的训练时间随问题的规模及复杂程度呈指数级增长。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。

采用同样基于全局搜索策略的标准微粒群优化(PSO)算法来训练神经网络参数, 可以避免 GA 的上述缺陷, 原因是其速度一位移模型操作简单。标准 PSO 算法简单高效, 却也存在如群体多样性易丢失、不能保证以概率 1 收敛于全局最优解等问题^[6]。国内外研究者提出了很多改进的 PSO 算法, 按照具体操作方法主要可分为三类: 改变部分或全部微粒位置、改变个体极值、改变最优极值^[7]。这些算法能分别克服标准 PSO 算法的某些不足, 但却未全面考虑到微粒群在迭代过程的不同

收稿日期: 2010-02-24; 修回日期: 2010-05-10 基金项目: 国家自然科学基金资助项目(60702056)

作者简介: 涂娟娟(1981-), 女, 江西南昌人, 讲师, 博士研究生, 主要研究方向为计算智能(ecsitu@126.com); 詹永照(1962-), 男, 教授, 博导, 主要研究方向为模式识别、图像处理、计算智能; 韩飞(1976-), 男, 副教授, 硕导, 主要研究方向为神经网络、计算智能。

阶段的特点。鉴于此,提出了一种改进的分期变异微粒群优化算法(staging mutation PSO, SMPSO),在初期使适应度过低的微粒发生变异,在后期使停滞代数过高的个体极值和全局极值发生变异,始终将微粒群的多样性控制在合理范围内。将 SMPSO 算法用于 CPA 的网络参数训练中,通过实验证明在训练收敛速度、剪枝效率及分类正确率等方面都有较大提高,是一种更加有效的方法。

1 改进的分期变异 PSO 算法

标准 PSO 算法在迭代过程中计算每个微粒 i 的适应度,通过跟踪个体极值 P_i 和全局极值 P_g 来更新自己^[8]。微粒 i 根据如下的公式来更新自己的速度和新的位置:

$$V_i(t+1) = w \times V_i(t) + c_1 \times r_1 \times [P_i(t) - X_i(t)] + c_2 \times r_2 \times [P_g(t) - X_i(t)] \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

其中: $V_i(t)$ 是微粒 i 在 t 时刻的速度; w 是惯性权重; $X_i(t)$ 是粒子在 t 时刻的位置; r_1, r_2 是(0,1)的随机数; c_1, c_2 是学习因子,通常 $c_1 = c_2 = 2$ 。

群体的多样性是影响 PSO 算法全局收敛性能的一个重要因素。群体的高多样性表明群体能够在较大的空间内进行搜索,这有助于跳出局部最优和减少早熟现象的产生。但是,若一直保持较高的多样性会减慢算法的收敛速度,甚至导致不收敛。群体的低多样性表明,群体在一个较小的空间内进行搜索,可以提高解的精度,但是算法易陷入局部最优,发生早熟现象^[9]。

标准 PSO 算法的多样性在进化过程中不能很好地保持。因为在初期,微粒群中的个体相对分散,部分微粒可能距离其他微粒很远,多样性偏高导致初始迭代进程缓慢。随着迭代的进行,微粒都朝着 P_i 和 P_g 的方向聚集,在后期会变得较为集中,甚至出现多次迭代后极值未发生变化的情况。群体多样性下降到较低程度,不利于搜索到全局最优解。根据这些特点,提出了一种分期变异的 PSO 算法(SMPSO),初期使部分适应度过低的微粒发生变异,后期采用进化停滞步数作为触发条件,使个体极值 P_i 和全局极值 P_g 产生变异。经过以上分时期的有针对性的变异操作,能够始终将微粒群的多样性控制在合理的范围之内,从而有效提高全局收敛能力。

算法主要步骤如下:

a) 随机产生微粒群,初始化参数。

b) 计算每个微粒的适应度值,令适应度值最大的个体编号为 m 。

c) 若某微粒 j 的适应度值低于预设值,令 $X_j = (X_j + X_m) / 2$,并重新计算其适应度值。

d) 将每个微粒的适应度值与个体极值及全局极值作比较,确定新的个体极值及全局极值。

e) 判断微粒 i 的个体极值进化停滞步数 T_i 是否大于预设值,若是,按式(3)更新速度;否则按式(1)更新速度。

$$V_i(t+1) = w \times V_i(t) + c_1 \times r_1 \times [r_3 \times P_i(t) - X_i(t)] + c_2 \times r_2 \times [P_g(t) - X_i(t)] \quad (3)$$

其中: r_3 是(0,1)的随机数。

f) 判断全局极值进化停滞步数 T_g 是否大于预设值,若是,按式(4)更新速度;否则按式(1)更新速度。

$$V_i(t+1) = w \times V_i(t) + c_1 \times r_1 \times [P_i(t) - X_i(t)] + c_2 \times r_2 \times [r_4 \times P_g(t) - X_i(t)] \quad (4)$$

其中: r_4 是(0,1)的随机数。

g) 按式(2)更新位置。

h) 判断是否达到结束条件,若是,停止进化;否则,返回 b)。

2 基于 SMPSO 算法的神经网络相关性剪枝优化

神经网络 CPA 首先采用 BP 算法训练网络参数;然后计算隐节点输出之间的相关性,合并相关性较大的隐节点。BP 算法的缺陷会导致训练缓慢,计算出的参数不一定是最优值,进而影响剪枝的结果。将改进的 SMPSO 算法与 CPA 相结合,可以在一定程度上弥补上述缺陷。

基于 SMPSO 算法的神经网络 CPA(SMPSO-CPA)的思想是,首先采用 SMPSO 算法训练神经网络参数,然后计算隐节点输出之间的相关性。当两个隐节点的输出对所有样本具有较大的相关性时,将它们合并成一个节点;当某隐节点的输出对所有样本的标准差均较小时,将该输出看成常数,并入下一层的偏置节点。

SMPSO 算法训练神经网络参数的方法是,将网络参数(权值和阈值)编码成实数码串表示的个体,一个微粒对应一组参数,微粒的每一维对应该组中的每一个权值或阈值。对每一个微粒对应的神经网络,输入训练样本进行训练,产生的均方误差作为适应度函数。公式如下:

$$f(x) = \frac{1}{1 + \frac{1}{2n} \sum_{p=1}^n (y_p - t_p)} \quad (5)$$

以三层前向神经网络为例,SMPSO-CPA 的具体步骤如下:

a) 初始化。确定输入层、输出层、隐含层初始节点数,初始参数(权值和阈值),学习系数,动量系数,合成时刻误差 e_1 ,目标误差 e_2 ,相关系数阈值 h_1 ,标准差阈值 h_2 ,令当前学习次数 $k = 1$ 。

b) 使用 SMPSO 算法来训练网络参数,计算当前训练误差值 E 和对所有样本的隐节点输出序列。

c) 如果 $E > e_1, k = k + 1$,转到步骤 b);如果 $e_2 < E < e_1$,转到步骤 d);如果 $E < e_2$,终止学习。

d) 计算隐节点 i 的输出序列的标准差:

$$S_i^2 = \sum_{p=1}^n (V_{ip} - \bar{V}_i)^2 \quad (6)$$

其中: V_{ip} 是节点 i 对样本 p 的输出值; \bar{V}_i 是 V_{ip} 的平均值。

计算隐节点 i 和 j 的相关系数:

$$C_{ij} = \frac{\sum_{p=1}^n V_{ip} V_{jp} - n \bar{V}_i \bar{V}_j}{S_i S_j} \quad (7)$$

e) 如果 $C_{ij} > h_1$,且 $S_i^2 > h_2, S_j^2 > h_2$,删除隐节点 j ,同时对下一层的任意节点 k ,令

$$W_{ki} = W_{ki} + a \times W_{ki} \quad (8)$$

$$W_{kb} = W_{kb} + b \times W_{kj} \quad (9)$$

其中:

$$a = \frac{\frac{1}{n} \sum_{p=1}^n V_{ip} V_{jp} - \bar{V}_i \bar{V}_j}{\frac{1}{n} \sum_{p=1}^n V_{ip}^2 - \bar{V}_i^2}$$

$$b = \bar{V}_j - a \times \bar{V}_i$$

W_{ki} 是隐节点 i 与输出层节点 k 之间的连接权值, W_{kb} 是节点 k 的阈值。

f) 如果 $S_i^2 < h_2$, 将隐节点 i 并入输出层节点。删除隐节点 i , 并按式(10)修改输出层节点 k 的阈值:

$$W_{kb} = W_{kb} + \bar{V}_i \times W_{ki} \quad (10)$$

g) $k = k + 1$, 转向 b)。

采用改进的 SMPSO 算法取代 BP 算法用于 CPA 中神经网络参数的训练, 能加快神经网络参数训练收敛速度, 进而提高剪枝效率及仿真精确度。

3 实验结果分析

3.1 两类数据集上的实验

使用 Iris 及 Lung Cancer 数据集来进行分类测试实验。针对 Iris 设计的神经网络输入层节点数为 4, 输出层节点数为 3, 据经验初始隐节点数定为 8, 网络结构用 4-8-3 来表示, 最大迭代次数 2 000。针对 Lung Cancer 设计的初始神经网络结构为 9-12-1, 最大迭代次数 4 000。其他参数设置如下: 微粒群个数 $n = 40$, 惯性权重 $w = 0.9$, $c_1 = c_2 = 2$, 连接权值为 $[-1, 1]$ 变量, 合成时刻误差 $e_1 = 5 \times 10^{-4}$, 目标误差 $e_2 = 10^{-5}$ 。

针对两类数据集的神经网络 CPA、基于标准 PSO 算法的神经网络 CPA (PSO-CPA) 及 SMPSO-CPA 的训练收敛曲线如图 1 和 2 所示, 其他性能指标的剪枝后隐节点数和分类正确率如表 1 所示。

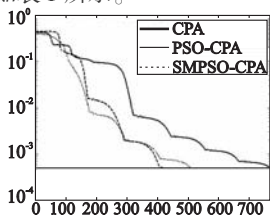


图1 Iris训练收敛曲线图

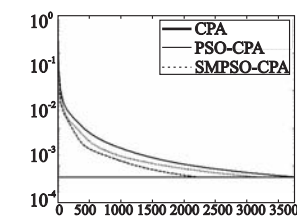


图2 Lung Cancer训练收敛曲线图

表1 三种算法性能指标

算法	指标	Iris	Lung Cancer
CPA	剪枝后隐节点数	6	9
	分类正确率/%	93.73	94.15
PSO-CPA	剪枝后隐节点数	4	7
	分类正确率/%	94.32	94.76
SMPSO-CPA	剪枝后隐节点数	4	6
	分类正确率/%	95.21	96.11

根据以上的图表可以得出如下结论:

a) 训练收敛速度。由图 1 和 2 可知, 采用 BP 算法训练的 CPA 收敛最慢, SMPSO-CPA 由于要进行变异操作, 训练时间比 PSO-CPA 略长。说明采用 PSO 算法优化神经网络参数在一定程度上克服了 BP 算法训练收敛速度慢的缺陷。

b) 剪枝效率和分类正确率。由表 1 可知, 在两类数据集上的实验都证明了 PSO-CPA 比采用 BP 算法训练网络参数的 CPA 剪枝效果更好, 分类正确率也更高。而 SMPSO-CPA 尽管在 Iris 数据集上的剪枝效果与 PSO-CPA 基本一样, 但分类正确率还是有提高。另外, 在 Lung Cancer 数据集上的剪枝效果和分类正确率均比 PSO-CPA 要好。

以上实验结果证明了将 SMPSO 算法用于神经网络 CPA 是一种有效的方法。

3.2 与其他神经网络结构优化方法的比较

剪枝法被公认为是优化网络结构、提高网络泛化能力的有

效办法。不同的剪枝算法都有各自的优缺点, SMPSO-CPA 与其他剪枝法如灵敏度法、权衰减法相比, 具有计算量小、训练结果稳定等优点^[10]。

与其他两类神经网络结构设计构造法和自适应法相比, SMPSO-CPA 在小规模神经网络训练方面具有一定优势, 因为构造法随着隐单元数的增加, 易出现过拟合的现象, 网络的泛化性能会变差^[11], 而自适应法的计算过程相对网络规模而言过于复杂^[1]。SMPSO-CPA 不太适合大规模网络的训练, 因为每次训练完网络参数以后只进行一次剪枝, 计算时间过长, 并且难以根据经验确定初始隐节点数^[12]。

4 结束语

本文提出了一种改进的分期变异 PSO 算法, 用于神经网络相关性剪枝优化, 其克服了 BP 算法及标准 PSO 算法在训练神经网络时的一些不足, 如收敛速度慢、易陷入局部最优、群体多样性易丢失等。实验证明, 该方法在一定程度上提高了神经网络剪枝效率和分类正确率。此外, 与其他神经网络结构优化方法相比, 该方法对小规模神经网络更加有效。

参考文献:

- [1] ISLAM M M, SATTAR M A, AMIN M F, *et al.* A new adaptive merging and growing algorithm for designing artificial neural networks [J]. *IEEE Trans on System, Man, and Cybernetics-Part B: Cybernetics*, 2009, 39(3):705-722.
- [2] CASTELLANO G, FANELLI A M, PELILLO M. An interactive pruning algorithm for feedforward neural networks [J]. *IEEE Trans on Neural Networks*, 1997, 8(3):519-531.
- [3] EENGELBRECHT A P. A new pruning heuristic based on variance analysis of sensitivity information [J]. *IEEE Trans on Neural Network*, 2001, 12(6):1386-1399.
- [4] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors [J]. *Nature*, 1986, 323(11):533-536.
- [5] 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.
- [6] 曾建潮, 崔志华. 一种保证全局收敛的 PSO 算法 [J]. *计算机研究与发展*, 2004, 14(8):1333-1338.
- [7] 高海兵, 周驰, 高亮. 广义粒子群优化模型 [J]. *计算机学报*, 2005, 28(12):1980-1987.
- [8] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// *Proc of IEEE International Conference on Neural Networks*. 1995: 1942-1948.
- [9] 赫然, 王永吉, 王青, 等. 一种改进的自适应逃逸微粒群算法及实验分析 [J]. *软件学报*, 2005, 16(12):2036-2044.
- [10] LAURET P, FOCK E, MARA T A. A node pruning algorithm based on a Fourier amplitude sensitivity test method [J]. *IEEE Trans on Neural Network*, 2006, 17(2):273-293.
- [11] GEORGE B, MICHAEL G. Feed-forward neural networks [J]. *IEEE Potentials*, 1997, 13(4):27-31.
- [12] KWOK T Y, YEUNG D Y. Constructive algorithms for structure learning in feedforward neural networks for regression problems [J]. *IEEE Trans on Neural Network*, 1997, 8(3):630-645.