

基于混沌 PSO 算法的选择性神经网络集成方法

田雨波,李正强,朱人杰

(江苏科技大学 电子信息学院,江苏 镇江 212003)

(yuboe@sohu.com; zqli08@163.com)

摘要:提出基于十进制粒子群优化算法(DePSO)和二进制 PSO 算法(BiPSO)的选择性神经网络集成(NNE)方法,通过 PSO 算法合理选择组成神经网络集成的各个神经网络,使个体间保持较大的差异度,减小“多维共线性”和样本噪声的影响。为有效保证 PSO 算法的粒子多样性,在迭代过程中加入混沌变异。试验表明,混沌 PSO 算法是组合优化权值的有效方法,同已有方法比较可以有效提高神经网络集成的泛化能力。

关键词:神经网络集成;粒子群优化;混沌

中图分类号: TP311.11 **文献标志码:** A

Selective neural network ensemble methods based on chaos PSO

TIAN Yu-bo, LI Zheng-qiang, ZHU Ren-jie

(School of Electronics and Information, Jiangsu University of Science and Technology, Zhenjiang Jiangsu 212003, China)

Abstract: Selective Neural Network Ensemble (NNE) methods based on Decimal Particle Swarm Optimization (DePSO) and Binary Particle Swarm Optimization (BiPSO) were proposed in this paper. The basic idea of the methods was to optimally select Neural Networks (NNs) to construct NNE with the aid of PSO. This may maintain the diversity of NNs and decrease the effect of collinearity and noise of sample. Meanwhile, chaos mutation was adopted in order to increase the diversity of particles of PSO. The experimental results show that the chaos PSO algorithm is an effective ensemble method, and it may improve the generalization ability of NNE in comparison with the available ones.

Key words: Neural Network Ensemble (NNE); Particle Swarm Optimization (PSO); chaos

0 引言

文献[1]开创性地提出了神经网络集成(Neural Network Ensemble, NNE)方法,他们证明可以简单地通过训练多个神经网络并将其结果通过相对多数投票或绝对多数投票法进行合成,显著提高神经计算系统的泛化能力。由于该方法易于使用且效果明显,为解决神经网络研究中涉及的网络结构选择、过拟合等问题提供了新的途径,它被视为一种非常有效的神经计算方法,已经在工程实践问题中得到成功运用^[2]。

目前,关于神经网络集成构成方法的研究主要集中在两个方面:一是如何生成(选定)集成误差中的个体网络,二是个体网络的输出如何合成为网络集成的输出。以回归问题为例,一些加权平均方法求取组合权值时存在矩阵求逆,容易受个体网络之间的“多维共线性”以及数据中噪声的影响,会降低神经网络集成的泛化能力^{[3]11, [4]}。为解决“多维共线性”问题,可以采用避免矩阵求逆、限制组合权值的方法^{[3]12}、选择性集成方法^[5]以及提取主成分^{[3]13}等方法。为减小噪声的影响,可以采用限制组合权值的方法以及调整优化组合权值所用的目标函数^[6]等方法。

本文主要从避免矩阵求逆、限制组合权值的角度出发,分别采用十进制粒子群优化算法(Decimal Particle Swarm Optimization, DePSO)和二进制 PSO 算法(Binary Particle Swarm Optimization, BiPSO)形成选择性神经网络集成方法,通过 PSO 算法合理选择组成神经网络集成的各个神经网络,使个体间保持较大的差异度,减小“多维共线性”和样本噪声

的影响。同时为有效保证 PSO 算法的粒子多样性,在迭代过程中加入混沌变异。采用本文的方法可以提高神经网络集成的泛化能力。

1 PSO 算法的基本原理

粒子群优化算法(Particle Swarm Optimization, PSO)是一种基于群体智能的进化计算技术,其思想来源于人工生命和进化计算理论,最早是由文献[7]受鸟群觅食行为的启发提出的。PSO 算法采用实数求解,并且需要调整的参数较少,易于实现,是一种通用的全局搜索算法。因此,算法一提出就得到众多学者的重视,并且已经在神经网络训练、函数优化和模糊系统控制等领域取得了大量的研究成果^[8]。PSO 的优势在于简单容易实现,同时又有深刻的智能背景,既适合科学研究,又特别适合工程应用。

PSO 算法模拟鸟群的捕食行为。设想这样一个场景:一群鸟在随机搜索食物。在这个区域里只有一块食物,所有的鸟都不知道食物在那里,但是他们知道当前的位置离食物还有多远。那么找到食物的最优算法是什么呢?最简单有效的就是搜寻目前离食物最近的鸟的周围区域。PSO 从这种模型中得到启示并用于解决优化问题。PSO 中每个优化问题的解都是搜索空间中的一只鸟,称之为“粒子”。所有粒子都有一个由被优化函数决定的适应值(候选解)和一个决定它们飞翔方向与距离的速度。在优化过程中,每个粒子记忆、追随当前的最优粒子,在解空间中进行搜索。PSO 算法初始化为一群随机粒子(随机候选解),然后通过迭代找到最优解。在每

收稿日期:2008-05-09;修回日期:2008-07-29。 **基金项目:**江苏省高校自然科学基金基础研究项目(07KJB510032)。

作者简介:田雨波(1971-),男,辽宁昌图人,副教授,博士,主要研究方向:计算智能应用于电子学与电磁学问题;李正强(1982-),男,湖北鄂州人,硕士研究生,主要研究方向:计算智能应用于信号与信息处理;朱人杰(1982-),男,山东滨州人,硕士研究生,主要研究方向:计算智能应用于信号与信息处理。

一次迭代过程中,粒子通过追逐两个极值来更新自己的位置:一个是粒子自身所找到的当前最优解,这个解称为个体极值 $pbest$;另一个是整个群体当前找到的最优解,这个解称为全局极值 $gbest$ 。

粒子在找到上述两个极值后,就根据下面两个公式来更新自己的速度与位置^[7-8]:

$$v_{i,d}^{k+1} = \omega \cdot v_{i,d}^k + c_1 \cdot rand() \cdot (pbest_{i,d}^k - x_{i,d}^k) + c_2 \cdot rand() \cdot (gbest_d^k - x_{i,d}^k) \quad (1)$$

$$x_{i,d}^{k+1} = x_{i,d}^k + v_{i,d}^{k+1} \quad (2)$$

其中 ω 为惯性权重, c_1 和 c_2 被称为学习因子, $rand()$ 为介于 (0, 1) 之间的随机数,这几个参数决定了 PSO 的开发和探测能力。 $v_{i,d}^k$ 和 $x_{i,d}^k$ 分别为粒子 i 在第 k 次迭代中第 d 维的速度和位置,两者均被限制在一定的范围内, $pbest_{i,d}^k$ 是粒子 i 在第 d 维的个体极值的位置, $gbest_d^k$ 是群体在第 d 维的全局极值的位置。

2 基于 PSO 算法的神经网络集成方法

假定已经分别训练出 n 个神经网络 f_1, f_2, \dots, f_n , 利用这 n 个神经网络组成的集成对 $f: R^m \rightarrow R^n$ 进行近似。为讨论简单起见,这里假设各个网络均只有一个输出变量,即所需近似的函数为 $f: R^m \rightarrow R$, 但本文的结论可以很容易推广到多个输出分量的情况。这种集成的过程可采用上述 PSO 优化算法实现。令 PSO 算法中的每一个粒子代表 $\{f_1, f_2, \dots, f_n\}$ 的一种集成,且粒子长度(粒子空间的维数)等于神经网络的数量 n 。可以分别采用 DePSO 算法和 BiPSO 算法来实现上述选择。

在采用 DePSO 算法实现选择性神经网络集成过程中,神经网络集成对输入 x 的实际输出为:

$$\bar{f}(x) = \sum_{i=1}^n \bar{\omega}_i f_i(x) \quad (3)$$

其中:

$$\bar{\omega}_i = \frac{\omega_i}{\sum_{i=1}^n \omega_i}, 0 < \omega_i < 1 \quad (4)$$

式(3)中的 $\bar{\omega}$ 反映了每个网络在集成中的重要性,对应于 PSO 算法中的位置矢量,采用上章介绍的 DePSO 进行进化。

在采用 BiPSO 算法实现选择性神经网络集成过程中,每个粒子在每一维的取值为离散的 0 或 1。若在某一维取值为 1,表示对应的网络个体参与集成;若为 0,则不参与。由此选择网络个体构建神经网络集成的问题可以转化为在 n 维 0-1 空间选择最优粒子的 PSO 优化问题。上章介绍的 PSO 优化算法只能用于连续空间,为解决离散空间的优化问题,文献[9]提出了离散二进制 PSO 算法。离散 PSO 算法中,粒子的位置每一维只有 0 或 1 两种状态,速度更新的方法可以与连续 PSO 相同,位置的更新则取决于由粒子速度决定的状态转移概率,速度大于一定的数值,粒子取 1 的可能性越大,反之越小,迭代算法如下式所示:

$$x_{i,d}^{k+1} = \begin{cases} 1, & \rho_{i,d}^{k+1} \leq sig(v_{i,d}^{k+1}) \\ 0, & \rho_{i,d}^{k+1} > sig(v_{i,d}^{k+1}) \end{cases} \quad (5)$$

其中 $\rho_{i,d}^{k+1} \in [0, 1]$ 是一随机数, $sig(\cdot)$ 为 Sigmoid 函数,是速度转换函数。采用 BiPSO 算法实现选择性神经网络集成,神经网络集成对输入 x 的实际输出为:

$$\bar{f}(x) = \frac{\sum_{i=1}^n \omega_i f_i(x)}{Number} \quad (6)$$

同样,其中的 ω 即对应于式(5)中的位置矢量, $Number$ 为 ω 中取值为 1 的个数。

为保证粒子在进化后期的多样性,对速度矢量采用混沌原理进行变异。混沌是现代科学的重要概念,是非线性科学的一个非常重要的内容^[10]。它虽看似混沌,却有着精致的内在结构,对初始条件依赖敏感,具有随机性、遍历性、规律性等特点。相对于一般的随机搜索方法,混沌搜索在小空间具有较强的局部搜索能力,细致搜索的有效性较强。本文取一个典型的混沌系统——Logistic 映射作为混沌信号发生器,迭代公式如下:

$$c^{k+1} = \mu c^k (1 - c^k) \quad (7)$$

其中: μ 为控制参量,当 $\mu = 4, 0 \leq c^0 \leq 1$ 时, Logistic 完全处于混沌状态。混沌信号产生后利用载波的方式将混沌引入到优化变量使其呈现混沌状态,同时把混沌运动的遍历范围放大到优化变量的取值范围,然后直接利用混沌变量搜索。混沌信号产生后,式(5)中的 $v_{i,d}^{k+1}$ 按照下式变异:

$$v_{i,d}^{k+1} = v_{i,d}^{k+1} + 2\alpha c^{k+1} - \alpha \quad (8)$$

其中 α 控制混沌变异的幅度。

3 数值试验

实验数据选用 2 个具有代表性的、被广泛研究的回归分析型问题 Friedman #1 函数和 Friedman #3 函数^[3-5,11], 函数表达式如表 1 所示。本文比较了 7 种神经网络集成方法:简单平均方法(Basic Ensemble Method, BEM)^[3], 广义集成方法(Generalized Ensemble Method, GEM)^[3-4], 线性回归(Linear Regression, LR)方法^[3-4], 主成分回归(Principal Components Regression, PCR)方法^[3], 基于遗传算法的选择性集成(Genetic Algorithm Selective ENsemble, GASEN)方法^[5], 以及基于混沌 DePSO 算法和基于混沌 BiPSO 算法。

分别用表 1 中的两个方程产生 2200 个数据,其中 200 个用作训练数据,其余 2000 个用作测试数据。采用 Matlab 中的神经网络工具箱,用基于 SCG 的变梯度 BP 算法训练 20 个网络,通过基于混沌理论随机产生权值,确保形成具有一定差异性的个体网络,然后根据训练集上的均方误差(Mean Square Error, MSE)来优化输出组合权值。考查神经网络集成的性能时,用集成输出与不带误差项的真值比较。在训练集上的均方误差用 MSE_{train} 表示,测试集上的均方误差 MSE_{test} 表示,它反映了集成的泛化能力。表 2 给出已有的计算结果,表 3 给出本文基于混沌 DePSO 算法和基于混沌 BiPSO 算法的实验结果,其中 PSO 的粒子个数为 30,学习因子按照文献[12]选取,即 $c_1 = 2.8, c_2 = 1.3$,惯性权重 ω 按照从 1 线性变化到 0.4^[13],每次进化 1000 代。为消除随机性造成的影响,本文对每个试验函数的每种集成方法分别进行 20 次重复试验。表 3 中的括号中的数据代表基于 BiPSO 算法时参与集成的神经网络的个数。表 4 给出基于 BiPSO 算法时在 20 次重复计算中参与集成的神经网络个数的统计。

由实验结果可知:1)对于 Friedman #1 和 Friedman #3 函数,同时存在“多维共线性”和噪声问题。GEM 和 LR 方法对权值限制低,又存在矩阵求逆,对噪声比较敏感,它们所建立集成的性能明显劣于其他方法。BEM 对权值限制较高,对噪声不敏感。PCR 通过提取主成分能消除一定的噪声。GASEN 的输出采用简单平均,也能抑制噪声。BiPSO 算法和 DePSO 算法对权值的限制较高,抑制噪声的效果较好。因为这 5 种方法都限制权值,故它们建立的神经网络集成性能较好。2)BiPSO 算法和 DePSO 算法受“多维共线性”和噪声的影响最小,所建立集成的性能明显优于 GEM 和 LR 方法,也优于 PCR 方法。与 GASEN 及 BEM 方法相比,对于 Friedman #1 函数 BiPSO 算法和 DePSO 算法明显好于 GASEN 和 BEM

方法,对于 Friedman # 3 函数 BiPSO 算法和 DePSO 算法同 GASEN 和 BEM 方法所得结果基本相当。如果考虑增大 PSO 算法的迭代次数,有望取得更好的计算结果。另外,在相同迭代次数的情况下,PSO 算法由于其简洁性,计算时间会比

GASEN 的计算时间短。3)从表 3 可以看出,基于 BiPSO 算法的计算结果要稍好于基于 DePSO 算法得到的结果。4)从表 4 可以看出,对于 BiPSO 算法参与集成的神经网络的个数一般为 8~11 左右。

表 1 试验用数据集

数据集	函数表达式	变量范围
Friedman # 1	$y = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon$	$x_i \in U[0, 1], \varepsilon \sim N(0, 1)$
Friedman # 3	$y = \tan^{-1}\left(\frac{x_2 x_3 - \frac{1}{x_2 x_4}}{x_1}\right) + \varepsilon$	$x_1 \in U[0, 100], x_2 \in U[40\pi, 560\pi]$ $x_3 \in U[0, 1], x_4 \in U[0, 11]$ $\varepsilon \sim N(0, 0.01)$

表 2 已有集成方法的试验结果

集成方法	Friedman # 1		Friedman # 3 ($\times 10^{-2}$)	
	MSE_{train}	MSE_{test}	MSE_{train}	MSE_{test}
BEM	0.207	0.341	0.258	0.820
GEM	0.270	0.467	0.368	1.624
LR	0.270	0.469	0.370	1.638
PCR	0.184	0.280	0.288	0.960
GASEN	0.183	0.281	0.252	0.806

表 3 本文所用集成方法的试验结果

集成方法	函数 Friedman # 1						函数 Friedman # 3					
	$MSE_{train} (\times 10^{-2})$			$MSE_{test} (\times 10^{-2})$			$MSE_{train} (\times 10^{-2})$			$MSE_{test} (\times 10^{-2})$		
	最好值	最差值	平均值	最好值	最差值	平均值	最好值	最差值	平均值	最好值	最差值	平均值
BiPSO	0.5195(9)	1.1867(7)	0.7699	1.1189(8)	4.6976(13)	2.1402	0.0815(11)	0.9503(7)	0.2584	0.1889(11)	1.3688(10)	0.8287
DePSO	0.5516	1.0008	0.7334	1.3102	4.1716	2.3651	0.0884	0.9799	0.2617	0.1895	1.5768	0.8346

表 4 BiPSO 集成方法参与集成的神经网络个数统计(共 20 次计算)

函数	参与集成网络数	出现次数统计
Friedman # 1	5	1
	7	1
	8	6
	9	5
	10	4
	11	2
Friedman # 3	13	1
	5	1
	6	2
	7	2
	8	3
	9	4
	10	4
	11	3
	12	1

4 结语

基于 DePSO 和 BiPSO 形成选择性神经网络集成方法,通过 PSO 算法合理选择组成神经网络集成的各个神经网络,使个体间保持较大的差异度。在网络集成过程中,通过对组合权值的适当限制可以提高神经网络集成的性能,从而避免了矩阵求逆,减小“多维共线性”和对噪声的过拟合问题。为有效保证 PSO 算法的粒子多样性,在迭代过程中充分利用混沌原理的随机性、遍历性、规律性等特点而进行变异。通过试验结果可以看出,采用本文的方法可以提高神经网络集成的泛化能力。

参考文献:

[1] HANSEN L K, SALAMON P. Neural network ensembles[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990,

12(10): 993-1001.

- [2] 周志华,陈世福. 神经网络集成[J]. 计算机学报, 2002, 25(1): 1-7.
- [3] MERZ C J, PAZZANI M J. A principal components approach to combining regression estimates[J]. Machine Learning, 1999, 36(1/2): 9-32.
- [4] HASHEM S. Treating harmful collinearity in neural network ensembles[C]// Combining Artificial Neural Nets: Ensemble and Modular Multi2net Systems. Berlin: Springer-Verlag, 1999: 101-123.
- [5] ZHOU ZHI-HUA, WU JIAN-XIN, TANG WEI. Ensembling neural networks: Many could be better than all[J]. Artificial Intelligence, 2002, 137(1/2): 239-263.
- [6] DIETTERICH T G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization[J]. Machine Learning, 2000, 40: 139-157.
- [7] KENNEDY J, EBERHART R. Particle swarm optimization[C]// Proceedings of the 1995 IEEE International Conference on Neural Networks. Piscataway, NJ: IEEE Computer Society, 1995: 1942-1948.
- [8] 曾建潮,介婧,崔志华. 微粒群算法[M]. 北京: 科学出版社, 2004.
- [9] KENNEDY J, EBERHART R. A discrete binary version of the particle swarm optimization[C]// Proceedings of the 1997 IEEE International Conference on Computational Cybernetics and Simulation. Piscataway, NJ: IEEE Computer Society, 1997: 4104-4108.
- [10] 黄润生,黄浩. 混沌及其应用[M]. 武汉: 武汉大学出版社, 2005.
- [11] BREIMAN L. Bagging predictors[J]. Machine Learning, 1996, 24(2): 123-140.
- [12] 张丽平. 微粒群算法的理论与实践[D]. 杭州: 浙江大学, 2005.
- [13] SHI Y, EBERHART R. Empirical study of particle swarm optimization[C]// Proceedings of the 1999 Congress on Evolutionary Computation. Washington, DC: IEEE Computer Society, 1999: 1945-1950.