

混沌灰狼优化算法训练多层感知器

晏福 徐建中* 李奉书

(哈尔滨工程大学经济管理学院 哈尔滨 150001)

摘要: 灰狼优化算法(GWO)是一种新的基于灰狼捕食行为的元启发式算法, 被证明是一种具有高水平的探索和开发能力的算法。但是存在开发和探索不平衡的问题, 以至于其优化性能并不理想。该文将混沌理论引入GWO中, 用于平衡GWO的探索和开发, 提出一种改进的混沌灰狼优化算法(CGWO), 并应用于多层感知器(MLPs)的训练。首先, 基于Cubic混沌理论对GWO的位置更新公式进行改进, 以增加个体的多样性, 增大跳出局部最优的概率和对解空间进行深入的搜索; 其次, 设计一种非线性收敛因子, 用于协调和平衡CGWO算法在不同迭代进化时期的探索和开发能力; 最后, 将CGWO算法作为MLPs的训练器, 用于对3个复杂分类问题进行分类实验。结果表明: CGWO在分类准确率, 避免陷入局部最优, 全局收敛速度和鲁棒性方面相较于其他对比算法均具有较好的性能。

关键词: 灰狼优化算法; 混沌理论; 非线性收敛因子; 多层感知器; 分类问题

中图分类号: TP301.6

文献标识码: A

文章编号: 1009-5896(2019)04-0872-08

DOI: 10.11999/JEIT180519

Training Multi-layer Perceptrons Using Chaos Grey Wolf Optimizer

YAN Fu XU Jianzhong LI Fengshu

(School of Economics and Management, Harbin Engineering University, Harbin 150001, China)

Abstract: The Grey Wolf Optimizer (GWO) algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves in nature, and it is an algorithm with high level of exploration and exploitation capability. This algorithm has good performance in searching for the global optimum, but it suffers from unbalance between exploitation and exploration. An improved Chaos Grey Wolf Optimizer called CGWO is proposed, for solving complex classification problem. In the proposed algorithm, Cubic chaos theory is used to modify the position equation of GWO, which strengthens the diversity of individuals in the iterative search process. A novel nonlinear convergence factor is designed to replace the linear convergence factor of GWO, so that it can coordinate the balance of exploration and exploitation in the CGWO algorithm. The CGWO algorithm is used as the trainer of the Multi-Layer Perceptrons (MLPs), and 3 complex classification problems are classified. The statistical results prove the CGWO algorithm is able to provide very competitive results in terms of avoiding local minima, solution precision, converging speed and robustness.

Key words: Grey Wolf Optimizer (GWO); Chaos theory; Nonlinear convergence factor; Multi-Layer Perceptrons (MLPs); Classification problem

1 引言

神经网络(Neural Networks, NNs)是智能计算

领域最伟大的发明之一, 它模仿了人类大脑的神经元并主要用于解决分类问题和进行数据预测。1943年, McCulloch等人^[1]首次提出神经网络的概念。从此, 神经网络便得以迅速发展, 因而不同类型的神经网络相继被提出, 诸如: 前馈神经网络(FNNs)^[2]、Kohonen自组织网络^[3]、径向基函数(RBF)神经网络^[4]、递归神经网络^[5]和强化神经网络^[6]。近年来, 前馈神经网络, 尤其是双层前馈神经网络在实际应用中十分广泛, 双层前馈神经网络即只有一个隐含层的前馈神经网络, 又被称为多层感知器(Multi-Layer Perceptrons, MLPs)^[7]。多层感知器适用于对非线性可分模式的分类^[8]和函数逼

收稿日期: 2018-05-28; 改回日期: 2018-12-03; 网络出版: 2018-12-14

*通信作者: 徐建中 xujianzhongxjz@163.com

基金项目: 国家社会科学基金(16BJY078), 黑龙江省经济社会发展重点研究课题(KY10900170004), 黑龙江省哲学社会科学研究规划(17JYH49)

Foundation Items: The National Social Science Foundation of China (16BJY078), The Key Program of Economic and Social of Heilongjiang Province (KY10900170004), The Philosophy and Social Science Research Planning Program of Heilongjiang Province (17JYH49)

近^[9,10],并且文献[11]已经证明多层感知器能够逼近任意连续和离散函数。从经验中获得学习的能力对神经网络来说至关重要,对前馈神经网络也是如此。在前馈神经网络中,用于学习的方法被称为训练器。训练器是用来寻找最佳的连接权重和偏差以保证输出误差最小。反向传播学习算法和基于梯度的方法是两类经典的训练器^[12]。对于前馈神经网络而言,研究人员往往倾向于使用标准或改进的反向传播学习算法作为学习训练器,而基于梯度的方法由于其较慢的收敛速度和容易陷入局部最优的缺陷而受到应用制约。

在多层感知器的学习训练过程中,其主要目的就是找到最佳的连接权重和偏差使得输出误差最小。然而,大多数情况下多层感知器收敛到的是局部最优解而不是全局最优解。也就是说,用于训练多层感知器的学习算法容易陷入局部最优而很难获得全局最优,所以很难找到多层感知器的最佳连接权重和偏差。群智能算法(SI)是一类随机优化算法,它具有较好的全局最优收敛性能。因此,越来越多的研究将群智能算法作为MLPs的学习算法,比如:遗传算法(GAs)^[13,14]、差分进化算法(DE)^[15,16]、蚁群优化(ACO)^[17,18]、灰狼优化算法(GWO)^[7]和粒子群优化算法(PSO)^[19]。这些基于群智能的随机优化算法在很大程度上降低了陷入局部最优的概率,但它们在收敛速度和计算准确度方面仍然不尽人意^[20]。为了克服这些缺陷,在群智能算法中引入混沌理论能极大改善其优化性能,这主要归因于混沌理论具有较高的混合能力和遍历特性,它能产生具有更高的多样性和灵活性的潜在解^[21]。因此,在群智能算法中引入混沌理论能有效改善算法的优化性能。GWO是一种寻优性能较好的群智能算法,在GWO中,位置更新是由前3个最佳适应度狼(α wolf, β wolf和 δ wolf)所决定,这一位置更新原理有利于增强粒子的开发能力但其探索能力则较差,因此导致了GWO算法的以上两方面的不足^[22]。为提高GWO算法的性能,一个积极的研究突破点就是对其位置更新进行改进。因此,本文尝试将混沌理论引入到GWO中,对其位置更新方程进行改进,并将改进后的GWO(CGWO)作为MLPs的学习算法对其进行训练,数值实验结果表明CGWO算法是一种有效的多层感知器训练算法。

2 灰狼优化算法

灰狼是属于犬科家族的群居动物,它是一种处于动物食物链顶端的顶级掠食者。在这个群落中,阶级等级鲜明、群体组织严明且分工明确。其中,起首要领导作用的狼被称为阿尔法狼(α wolf),处

于第2领导梯队的狼,即协助阿尔法狼决策和领导整个群落的狼被称为贝塔狼(β wolf),而协助阿尔法狼和贝塔狼的第3领导梯队的狼被称为西格玛狼(δ wolf),其余的则被称为欧米伽狼(ω wolf)且受前面3个领导者的领导和指挥^[23]。灰狼有一种独特的捕食行为,即包围攻击捕食方法。基于这种捕食方法,Mirjalili等人^[24]提出一种新的群智能优化算法,即灰狼优化算法(GWO)。在GWO算法中,假设灰狼的种群规模为 N ,搜索空间为 d 维,则第 i 只灰狼在空间中的位置为 $\mathbf{X}_i = (x_1, x_2, \dots, x_d)$,而猎物的空间位置即为全局最优解。

根据文献^[24],灰狼在捕食过程中是受阿尔法狼(α wolf)、贝塔狼(β wolf)和西格玛狼(δ wolf)的指挥和控制,并通过包围攻击方式围捕猎物,因此,灰狼包围猎物的位置更新如式(2)^[24]:

$$\mathbf{D} = |\mathbf{C} \cdot \mathbf{X}_p(t) - \mathbf{X}(t)| \quad (1)$$

$$\mathbf{X}(t+1) = \mathbf{X}_p(t) - \mathbf{A} \cdot \mathbf{D} \quad (2)$$

其中, t 为当前迭代次数, $\mathbf{X}_p = (x_1, x_2, \dots, x_d)$ 为猎物位置, $\mathbf{A} \cdot \mathbf{D}$ 为包围步长,向量 \mathbf{A} 和 \mathbf{C} 定义为

$$\mathbf{A} = 2(\mathbf{r}_1 - \mathbf{E}) \cdot \mathbf{a} \quad (3)$$

$$\mathbf{C} = 2\mathbf{r}_2 \cdot \mathbf{a} \quad (4)$$

其中, \mathbf{r}_1 和 \mathbf{r}_2 为区间 $[0, 1]$ 之间的1行 d 列随机向量, \mathbf{E} 是每一个元素都是1的1行 d 列向量, \mathbf{a} 为收敛因子向量,随着迭代次数增加从2线性递减到0,即

$$\mathbf{a} = 2(1 - t/t_{\max}) \cdot \mathbf{E}^T \quad (5)$$

由式(1)–式(5)可知,其他灰狼个体在捕食过程中由阿尔法狼(α wolf)、贝塔狼(β wolf)和西格玛狼(δ wolf)引导指挥的捕食位置更新如式(6)–式(8)^[24]:

$$\begin{aligned} \mathbf{D}_\alpha &= |\mathbf{C}_1 \cdot \mathbf{X}_\alpha - \mathbf{X}|, \mathbf{D}_\beta = |\mathbf{C}_2 \cdot \mathbf{X}_\beta - \mathbf{X}|, \\ \mathbf{D}_\delta &= |\mathbf{C}_3 \cdot \mathbf{X}_\delta - \mathbf{X}| \end{aligned} \quad (6)$$

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{X}_\alpha - \mathbf{A}_1 \cdot \mathbf{D}_\alpha, \mathbf{X}_2 = \mathbf{X}_\beta - \mathbf{A}_2 \cdot \mathbf{D}_\beta, \\ \mathbf{X}_3 &= \mathbf{X}_\delta - \mathbf{A}_3 \cdot \mathbf{D}_\delta \end{aligned} \quad (7)$$

$$\mathbf{X}(t+1) = \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3} \quad (8)$$

3 混沌灰狼优化(CGWO)算法

3.1 基于混沌理论的位置更新

GWO算法自2014年首次提出以来,便被广泛用于对实际工程和管理问题进行优化应用,并且许多改进版本的GWO也被相继提出。然而,正如前文所描述那样,GWO算法的位置更新是由前3个适应度最好的阿尔法狼(α wolf)、贝塔狼(β wolf)和西格玛狼(δ wolf)所决定,这样的位置更新原理一方面极大地利用了前3个适应度最好的狼已有的信息

对解空间的某些邻域进行局部搜索, 增强了GWO的开发性能, 但另一方面仅依据前3个适应度最好的狼的位置来更新整个种群的位置, 使得参与位置更新的粒子变得稀少, 从而引起群体位置更新所需要的信息不足, 导致GWO在探测更广泛的搜索区域时所需的探测信息不足, 降低了GWO的全局探测能力, 进而使得GWO的开发性能和探索性能不能保持平衡, 这就导致了GWO算法容易早熟收敛和易陷入局部最优。而混沌理论因为具有强大的遍历性、敏感性和非重复性而被广泛引入群智能算法中用以增强粒子多样性以改善其优化性能^[21], 与随机搜索相比, 它可以以更高的概率和速度对搜索空间进行彻底搜索。基于以上分析, 本文将混沌理论引入GWO的位置更新式(8), 得到改进后的位置更新如式(9):

$$\mathbf{X}(t+1) = \frac{1}{3} \sum_{i=1}^3 (\mathbf{X}_i - \mathbf{R}_n \cdot (\mathbf{R}'_n \cdot \mathbf{X}_i - \mathbf{X}_i)) \quad (9)$$

其中, 式(9)的构建是受灰狼包围捕食原理的启发, \mathbf{R}_n 和 \mathbf{R}'_n 为混沌映射向量, 定义为

$$\mathbf{R}_n = (2r_n - 1) \cdot \mathbf{E} \cdot \mathbf{a}' \quad (10)$$

$$\mathbf{R}'_n = 2 - \mathbf{R}_n \quad (11)$$

式(10)中, \mathbf{a}' 为非线性递减收敛因子, 将在下一节讨论, r_n 为基于Cubic混沌映射^[25]的优化算子

$$\begin{aligned} r_n &= 4r_{n-1}^3 - 3r_{n-1}, \quad -1 \leq r_{n-1} \leq 1, \\ r_0 &= 0.0068 \end{aligned} \quad (12)$$

3.2 非线性递减收敛因子

对于群智能优化算法而言, 探索和开发能力对于其优化性能至关重要。而对于GWO来说, 其探索和开发能力在很大程度上也依赖于收敛因子 \mathbf{a} 的取值, 如何选取合适的收敛因子 \mathbf{a} 来协调GWO的探索和开发能力是一个值得深究的研究问题。探测能力意味着群体需要探测更广泛的搜索区域, 避免算法陷入局部最优; 开发能力主要是利用群体已有的信息对解空间的某些邻域进行局部搜索, 对算法的收敛速度具有决定性影响^[22]。根据文献^[22,26], 一个大的收敛因子 \mathbf{a} 有利于全局探索, 相反, 一个小的收敛因子 \mathbf{a} 则有利于局部开发。然而, 基本GWO中的收敛因子 \mathbf{a} 是随着迭代次数从2线性递减到0的, 并不能完全体现出GWO的探索和开发过程。基于以上分析和启发, 本文设计了一个非线性递减收敛因子 \mathbf{a}' , 用于平衡CGWO的探索和开发, 其表达式见式(13)

$$\mathbf{a}' = 2 \left(1 - \left(\frac{t}{t_{\max}} \right)^\mu \right) \cdot \mathbf{E}^T \quad (13)$$

其中, t 为当前迭代次数, t_{\max} 为最大迭代次数, $\mu > 0$ 为非线性控制参数, 本文取 $\mu = 0.68$ 。同时, 用式(13)代替式(5)。

3.3 CGWO的算法实现

综上所述, 本文提出的CGWO算法实现过程如下:

步骤 1 算法参数初始化: 灰狼种群规模 N ; 最大的迭代次数 t_{\max} ; 问题的空间维数 d ; 非线性控制参数 μ ;

步骤 2 种群初始化: 随机生成 N 个灰狼个体 $\{\mathbf{X}_i, i = 1, 2, \dots, N\}$; 计算这 N 个灰狼个体的适应度值 $\{f(\mathbf{X}_i), i = 1, 2, \dots, N\}$, 并找出 \mathbf{X}_α , \mathbf{X}_β 和 \mathbf{X}_δ ;

步骤 3 设置 $t = 1$ (当前迭代次数);

步骤 4 如果 $t < t_{\max}$ (或当迭代停止条件未满足时);

步骤 5 利用式(9)更新当前迭代食物源位置;

步骤 6 利用式(13)更新收敛因子 \mathbf{a}' , 利用式(10)–式(12)更新混沌映射 \mathbf{R}_n 和 \mathbf{R}'_n ;

步骤 7 计算当前迭代的每一个个体的适应度值 $\{f(\mathbf{X}_i(t)), i = 1, 2, \dots, N\}$;

步骤 8 更新 \mathbf{X}_α , \mathbf{X}_β 和 \mathbf{X}_δ ;

步骤 9 $t = t + 1$ 。

4 基于CGWO算法的多层感知器

对于MLPs而言, 由于数据样本空间大多是高维、多模态、被噪声污染和数据缺失的情形, MLPs的权值和偏差训练被认为是一个极具挑战性的优化问题。CGWO算法是一种启发式算法, 而采用启发式算法作为MLPs的训练器进行训练最重要步骤就是选择合适的编码策略。选择编码策略是为了把MLPs的权重和偏差表示为进化算法的迭代计算粒子变量。根据文献^[27], 编码策略被划分为3种不同的类型, 即: 向量编码、矩阵编码和二进制编码。根据文献^[20], 矩阵编码非常适合用来对神经网络进行训练, 因为编码策略使得神经网络解码变得比较容易。

在本文中, 选择矩阵编码策略来把MLPs的权重和偏差表示为每一次迭代训练过程中训练器的粒子变量, 因此, 本文MLPs的训练器的变量定义为^[7]

$$\begin{aligned} \mathbf{V} &= \{\mathbf{W}, \theta\} \\ &= \{w_{1,1}, w_{1,2}, \dots, w_{n,n}, \theta_1, \theta_2, \dots, \theta_h\} \end{aligned} \quad (14)$$

其中, n 为输入节点数, $w_{i,j}$ 表示第 i 个节点与第 j 个节点的连接权重, θ_j 表示第 j 个隐藏节点的偏差。

在定义完MLPs训练器的变量之后, 还需对MLPs的训练器即CGWO算法定义适应度函数。由前文分析可知, 训练MLPs的目的就是通过向MLPs

输入训练样本和测试样本, 利用CGWO算法训练求得合适的连接权重和偏差, 使得MLPs对目标问题获得高的分类准确率、逼近度和预测精度。因此, 把MLPs的实际输出值与期望输出值的均方差(MSE)作为衡量MLPs性能好坏的指标, MSE越小, 则MLPs性能越好, 反之则越坏。

定义 1 均方差(Mean Square Error, MSE)^[7,12]: 设 o_i^k 表示MLPs中第 k 个训练样本作为输入单元时第 i 个输出单元的实际输出值, d_i^k 表示MLPs中第 k 个训练样本被使用时第 i 个输出单元的期望输出值。则MSE的表达式为

$$\text{MSE} = \sum_{i=1}^m (o_i^k - d_i^k)^2 \quad (15)$$

显然, 定义1针对的是只有1个训练样本的情况, 然而, 实际研究对象往往却不只1个训练样本。假设研究对象有 s 个训练样本, 此时, 用以衡量MPLs性能指标的MSE则变为这 s 个训练样本的平均MSE, 其表达式为

$$\overline{\text{MSE}} = \frac{1}{s} \sum_{k=1}^s \sum_{i=1}^m (o_i^k - d_i^k)^2 \quad (16)$$

其中, s 为训练样本数, m 为输出个数。

综上, MPLs的训练器(CGWO算法)的适应度函数如定义2。

定义 2 训练器适应度函数: 设向量 \mathbf{V} 为MLPs训练器的变量, $\overline{\text{MSE}}$ 为MLPs的平均MSE, 则MLPs的训练器的适应度函数表达式如式(17)所示^[7]:

$$\text{MinF}(\mathbf{V}) = \overline{\text{MSE}} \quad (17)$$

定义 3 训练器分类准确率(Classification Accuracies, CA): 分类准确率又称“精度”或“正确率”, 它是评价分类算法性能的重要指标。设待分类集 S 有 n 个属性, S 集的第 i 个属性个数记为 S_i ($1 \leq i \leq n$), T_i 表示分类器能对属性 i 进行正确分类的个数, 则定义分类器对待分类集 S 的分类准确率为

$$\text{CA} = \left(\sum_{i=1}^n T_i / \sum_{i=1}^n S_i \right) \times 100 \quad (18)$$

图1为基于CGWO算法的多层感知器的原理图。由图1可知, CGWO算法通过接收所有训练样本和所有期望输出样本的平均MSE作为目标函数进行迭代训练, 通过不断迭代进化调整连接权重和偏差, 为MLPs提供迭代训练后的最佳连接权重和偏差。

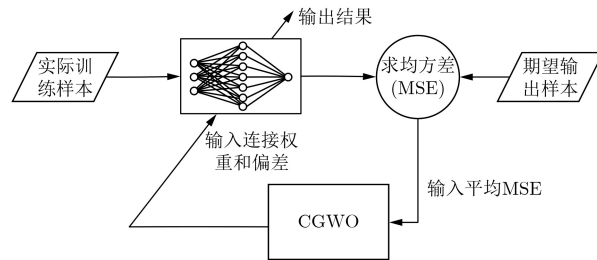


图1 基于CGWO的多层感知器

5 结果和讨论

5.1 测试问题及实验设置

为了测试本文所提出的CGWO算法对MLPs的训练性能, 选取了问题1: 3位奇偶校验问题, 问题2: 气球分类问题和问题3: 虹膜分类问题作为本文的3个研究对象^[28]。为了对CGWO算法的训练性能进行客观对比, 本文选取了GWO^[7], PSO, GSA (Gravitational Search Algorithm)和PSOGSA^[12]算法作为MLPs的训练器, 分别对3个研究对象进行实验仿真。本文对以上选取的算法的参数设置如下:

对于PSO-MLP, c_1 和 c_2 的值设置为2, r_1 和 r_2 为0到1之间的随机数, w 为从0.9线性递减到0.4, 初始粒子速度在0到1之间的随机生成。

对于GSA-MLP, a 设置为20, 引力常数 G_0 设为1, 初始粒子速度由0到1之间的随机生成, 对于每一个粒子的初始加速度和质量均设置为0。

对于PSOGSA-MLP, c_1 和 c_2 的值设置为1, w 为从0.9线性递减到0.4, 初始粒子速度在0到1之间随机生成。

对于CGWO-MLP, μ 设置为0.55, r_0 设置为0.0068。

GWO-MLP, PSO-MLP, GSA-MLP, PSOGSA-MLP和CGWO-MLP对于问题1、问题2和问题3的种群规模分别设置为50, 200, 200。最大迭代次数为250。对于多层感知器的网络结构设置, 隐含层节点数的选取规则为 $2N+1$, N 为数据集的特征个数^[7]。

5.2 3位奇偶校验(XOR)问题

N 位奇偶校验问题是一个著名的非线性标准测试问题。在该问题中, 目的是要在输入向量中识别“1”的数量, 并返回一个异或结果(XOR), 即: 如果输入向量包含奇数个“1”, 则输出为“1”; 如果输入向量包含偶数个“1”, 则输出为“0”。表1显示了这一问题的输入和期望输出。本文把3位奇偶校验问题的MLPs结构设置为3~7~1, 即3个输入, 7个隐藏层节点和1个输出, 并分别对5种算法进行10次的独立实验, 选取 $\overline{\text{MSE}}$ 的统计参数: 平均值、中值、标准差和最好值作为对比参数。其实验结果如表2和图2所示。

表 1 3位奇偶校验问题(3 bit XOR)

输入	输出
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	1
1 0 1	0
1 1 0	0
1 1 1	1

表 2 5种算法对3 bit XOR问题10次独立运行结果的比较

算法	平均值	中值	标准差	最好值
PSO-MLP	1.48e-04	1.65e-05	2.40e-04	7.67e-09
GSA-MLP	2.35e-01	2.38e-01	1.17e-02	2.10e-01
PSOGSA-MLP	1.27e-02	9.29e-06	2.57e-02	1.64e-09
GWO-MLP	7.00e-03	6.07e-03	1.89e-02	2.90e-05
CGWO-MLP	6.01e-06	1.21e-08	1.33e-05	2.69e-09

由表2可知, CGWO算法在10次独立运算中关于3个MSE统计参数: 平均值、中值和标准差获得

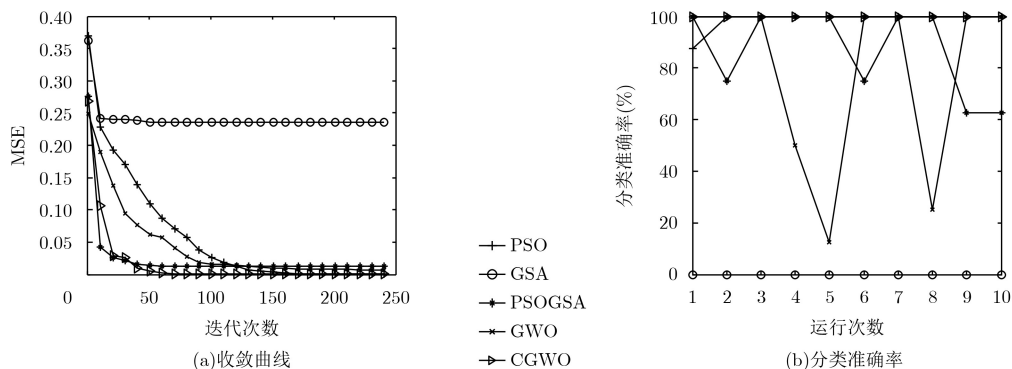


图 2 5种算法对3 bit XOR问题10次独立运行的收敛曲线和分类准确率

于MSE的4个统计参数均得到了理论最优值。其次, PSOGSA算法关于4个统计参数计算获得了优于GWO, CGWO和GSA的计算结果。CGWO算法和GWO算法的计算结果十分接近, 但GWO要稍优于CGWO, 而GSA算法要明显劣于其余4种算法。

表 3 5种算法对气球分类问题10次独立运行结果的比较

算法	平均值	中值	标准差	最好值
PSO-MLP	0	0	0	0
GSA-MLP	5.90e-03	4.10e-03	6.00e-03	4.69e-04
PSOGSA-MLP	9.85e-32	9.73e-40	2.94e-31	3.81e-61
GWO-MLP	1.12e-18	3.56e-20	2.38e-18	2.49e-26
CGWO-MLP	2.35e-15	2.07e-18	5.03e-15	1.00e-18

了最优值, 而在最好值处, PSOGSA算法的计算结果最佳, CGWO算法的计算结果与PSOGSA算法相近, 位居第2。以上统计结果表明, CGWO算法对3位奇偶校验问题的分类性能最好, 与GWO算法相比, 其寻优性能得到了很大的提高, 达到了性能改善的目的。

由图2可以看出, 图2(a)中, GSA算法的收敛性能最差, CGWO算法的收敛速度和收敛精度在5种算法中是最好的, PSO算法的收敛精度要高于GWO和PSOGSA算法, 但其收敛速度却较差; 图2(b)中, CGWO算法的平稳性最好且分类准确率最高, GWO和PSOGSA算法表现出了明显的波动性, PSO算法的平稳性和分类准确率仅次于CGWO。由此可以得出CGWO算法的收敛速度、分类准确度和鲁棒性相较于其他对比算法较好。

5.3 气球分类问题(Balloon classification)

气球数据集包含了4个特征, 18个训练/测试样本以及2个类别。该问题的维数为55维, MLPs对该数据集的结构设置为4~9~1, 即4个输入, 9个隐藏节点和1个输出。其实验结果见表3和图3。

由表3可得, 针对气球分类问题, PSO算法关

由图3可知, 在图3(a)中, GSA的收敛速率和收敛精度最差, CGWO的收敛速度最快, GWO和PSOGSA具有相近的收敛速度, PSO的收敛速度虽然不是最快的, 但是其收敛精度却是最高的, 其次为PSOGSA。CGWO的收敛精度与GWO相近, 但低于PSO和PSOGSA; 由图3(b), GSA在10次独立运行过程中, 分类准确度表现出剧烈的波动, 而PSO, PSOGSA, GWO和CGWO都达到了100%的分类准确度。

5.4 虹膜分类问题(Iris classification)

虹膜分类问题已被广泛应用于FNN领域。虹膜数据集有150个训练样本, 分为3个类别: Setosa, Versicolor和Virginica, 共4个基本特征: 萼片长度、萼片宽度、花瓣长度和花瓣宽度。本文使用MLPs

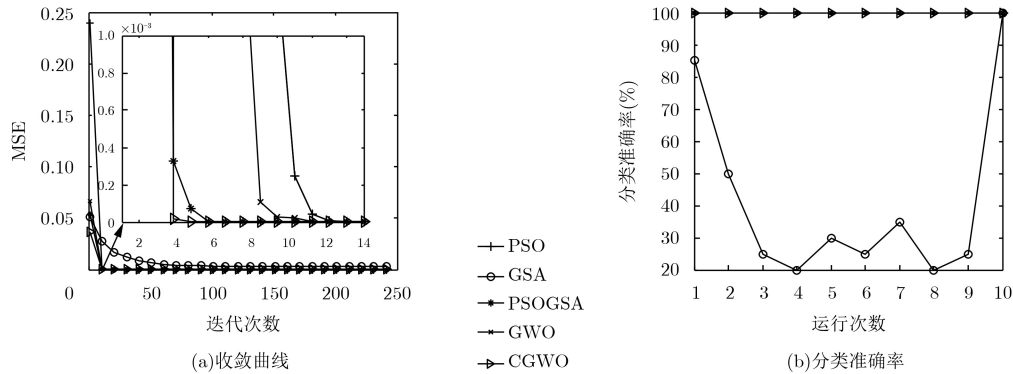


图3 5种算法对气球分类问题10次独立运行的收敛曲线和分类准确率

的结构为4~9~3来对该问题进行分类。其统计结果见表4，平均收敛趋势和分类准确率见图4所示。

由表4可知，对于虹膜分类问题的5种算法的10次独立运行计算的统计结果，CGWO的平均值和中值要优于其余4种算法且标准差接近于GWO，排名第2。PSO的最好值在所有对比算法中是最优的。CGWO计算所得的MSE的平均值和中值要优于其他算法，这表明CGWO算法的寻优性能在这5种算法中是最好的。同时，CGWO与GWO具有相近的标准差也表明CGWO算法继承了GWO算法的鲁棒性且要比PSO, GSA和PSOGSA的鲁棒性要好。

由图4可知，图4(a)展示了5种算法对虹膜分类问题的分类收敛曲线，其中，GSA算法的收敛速度和收敛精度最差，PSOGSA算法与CGWO算法具有相近的收敛速率且优于GWO和PSO算法。CGWO

算法在收敛精度方面是最高的，其次为PSO和GWO算法；从图4(b)中可以看出，PSO算法和PSOGSA算法在10次独立运行的分类准确度方面表现出明显波动，CGWO算法的平稳性最好且平均分类准确度最高，这充分验证了CGWO算法的鲁棒性和分类可靠性。

6 结束语

灰狼优化算法(GWO)是一种新提出的元启发式算法，它主要模拟了灰狼狩猎的3个主要步骤：寻找猎物，包围猎物和攻击猎物的领导协作过程。通过分析灰狼算法在该捕食过程中位置更新的不足，本文提出了一种基于混沌理论的改进灰狼优化(CGWO)算法用于训练多层感知器。选取3个分类为题：3位奇偶校验问题、气球分类问题和虹膜分类问题进行实验，用于评估改进灰狼优化算法的优化性能和鲁棒性。实验结果表明，与PSO, GSA, PSO-GSA和GWO算法相比，本文提出的CGWO算法在训练多层感知器时，在分类准确率、收敛精度和鲁棒性方面具有更好的性能。通过与PSO, GSA和PSOGSA的对比试验结果分析可知，PSO算法在迭代开发方面具有较好的效果，下一步有趣的研究突破点即将PSO引入CGWO中，应用于复杂函数优化，工程优化和求解多目标问题中。

表4 5种算法对虹膜分类问题10次独立运行结果的比较

算法	平均值	中值	标准差	最好值
PSO-MLP	2.70e-02	2.47e-02	1.76e-02	6.20e-03
GSA-MLP	1.83e-01	1.89e-01	2.30e-02	1.48e-01
PSOGSA-MLP	4.91e-02	1.84e-02	1.01e-01	1.16e-02
GWO-MLP	2.27e-02	2.19e-02	2.70e-03	1.71e-02
CGWO-MLP	1.90e-02	1.82e-02	4.10e-03	1.39e-02

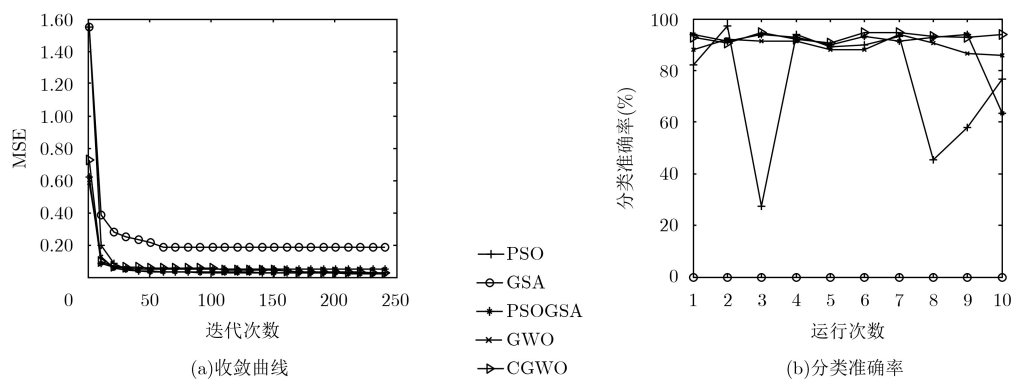


图4 5种算法对虹膜分类问题10次独立运行的收敛曲线和分类准确率

参考文献

- [1] MCCULLOCH W S and PITTS W. A logical calculus of the ideas immanent in nervous activity[J]. *The Bulletin of Mathematical Biophysics*, 1943, 5(4): 115–133. doi: [10.1007/BF02478259](https://doi.org/10.1007/BF02478259).
- [2] BEBIS G and GEORGIPOULOS M. Feed-forward neural networks[J]. *IEEE Potentials*, 1994, 13(4): 27–31. doi: [10.1109/45.329294](https://doi.org/10.1109/45.329294).
- [3] KOHONEN T. The self-organizing map[J]. *Proceedings of the IEEE*, 1990, 78(9): 1464–1480. doi: [10.1109/5.58325](https://doi.org/10.1109/5.58325).
- [4] JAFRASTEH B and FATHIANPOUR N. A hybrid simultaneous perturbation artificial bee colony and back-propagation algorithm for training a local linear radial basis neural network on ore grade estimation[J]. *Neurocomputing*, 2017, 235: 217–227. doi: [10.1016/j.neucom.2017.01.016](https://doi.org/10.1016/j.neucom.2017.01.016).
- [5] CHICEA D. Using neural networks for dynamic light scattering time series processing[J]. *Measurement Science and Technology*, 2017, 28(5): 055206. doi: [10.1088/1361-6501/aa61b4](https://doi.org/10.1088/1361-6501/aa61b4).
- [6] MORRO A, CANALS V, OLIVER A, et al. A stochastic spiking neural network for virtual screening[J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, 29(4): 1371–1375. doi: [10.1109/TNNLS.2017.2657601](https://doi.org/10.1109/TNNLS.2017.2657601).
- [7] MIRJALILI S. How effective is the Grey Wolf Optimizer in training multi-layer perceptrons[J]. *Applied Intelligence*, 2015, 43(1): 150–161. doi: [10.1007/s10489-014-0645-7](https://doi.org/10.1007/s10489-014-0645-7).
- [8] ISA N A M and MAMAT W M F W. Clustered-hybrid multilayer perceptron network for pattern recognition application[J]. *Applied Soft Computing*, 2011, 11(1): 1457–1466. doi: [10.1016/j.asoc.2010.04.017](https://doi.org/10.1016/j.asoc.2010.04.017).
- [9] 韦岗, 李华, 徐秉铮. 关于前馈多层神经网络多维函数逼近能力的一个定理[J]. *电子科学学刊*, 1997, 20(4): 433–438. WEI Gang, LI Hua, and XU Bingzheng. A novel theorem on the multi-dimensional function approximation ability of feed forward multi-layer neural networks[J]. *Journal of Electronics*, 1997, 20(4): 433–438.
- [10] LIBANO F, RECH P, TAMBARA L, et al. On the reliability of linear regression and pattern recognition feedforward artificial neural networks in FPGAs[J]. *IEEE Transactions on Nuclear Science*, 2018, 65(1): 288–295. doi: [10.1109/TNS.2017.2784367](https://doi.org/10.1109/TNS.2017.2784367).
- [11] LI Feng, ZURADA J M, LIU Yan, et al. Input layer regularization of multilayer feedforward neural networks[J]. *IEEE Access*, 2017, 5: 10979–10985. doi: [10.1109/ACCESS.2017.2713389](https://doi.org/10.1109/ACCESS.2017.2713389).
- [12] MIRJALILI S, MOHD H S, and MORADIAN S H. Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm[J]. *Applied Mathematics and Computation*, 2012, 218(22): 11125–11137. doi: [10.1016/j.amc.2012.04.069](https://doi.org/10.1016/j.amc.2012.04.069).
- [13] 张长利, 房俊龙, 潘伟. 用遗传算法训练的多层前馈神经网络对番茄成熟度进行自动检测的研究[J]. *农业工程学报*, 2001, 17(3): 153–156. doi: [10.3321/j.issn:1002-6819.2001.03.037](https://doi.org/10.3321/j.issn:1002-6819.2001.03.037). ZHANG Changli, FANG Junlong, and PANG Wei. Automated identification of tomato maturation using multilayer feedforward neural network with genetic algorithms (GA)[J]. *Transactions of the Chinese Society of Agricultural Engineering*, 2001, 17(3): 153–156. doi: [10.3321/j.issn:1002-6819.2001.03.037](https://doi.org/10.3321/j.issn:1002-6819.2001.03.037).
- [14] ZHANG Ridong and TAO Jili. A nonlinear fuzzy neural network modeling approach using an improved genetic algorithm[J]. *IEEE Transactions on Industrial Electronics*, 2018, 65(7): 5882–5892. doi: [10.1109/TIE.2017.2777415](https://doi.org/10.1109/TIE.2017.2777415).
- [15] 海宇娇, 刘青昆. 基于差分进化的ELM加权集成分类[J]. *计算机工程与应用*, 2017, 53(8): 57–60. doi: [10.3778/j.issn.1002-8331.1510-0051](https://doi.org/10.3778/j.issn.1002-8331.1510-0051). HAI Yujiao and LIU Qingkun. ELM weighting ensemble classification based on differential evolution[J]. *Computer Engineering and Applications*, 2017, 53(8): 57–60. doi: [10.3778/j.issn.1002-8331.1510-0051](https://doi.org/10.3778/j.issn.1002-8331.1510-0051).
- [16] LEEMAA N, NEHEMIAHA H K, and KANNANB A. Neural network classifier optimization using differential evolution with global information and back propagation algorithm for clinical datasets[J]. *Applied Soft Computing*, 2016, 49: 834–844. doi: [10.1016/j.asoc.2016.08.001](https://doi.org/10.1016/j.asoc.2016.08.001).
- [17] MICHALIS M and YANG Shengxiang. Training neural networks with ant colony optimization algorithms for pattern classification[J]. *Soft Computing*, 2015, 19(6): 1511–1522. doi: [10.1007/s00500-014-1334-5](https://doi.org/10.1007/s00500-014-1334-5).
- [18] KRZYSZTOF S and CHRISTIAN B. An ant colony optimization algorithm for continuous optimization: application to feed-forward neural network training[J]. *Neural Computing and Applications*, 2007, 16(3): 235–247. doi: [10.1007/s00521-007-0084-z](https://doi.org/10.1007/s00521-007-0084-z).
- [19] ROBERT C G, WANG Lingfeng, and MANSOOR A. Training neural networks using central force optimization and particle swarm optimization: Insights and comparisons[J]. *Expert Systems with Applications*, 2012, 39(1): 555–563. doi: [10.1016/j.eswa.2011.07.046](https://doi.org/10.1016/j.eswa.2011.07.046).
- [20] ZHANG Jingru, ZHANG Jun, LOCK T M, et al. A hybrid particle swarm optimization–back-propagation algorithm for feedforward neural network training[J]. *Applied Mathematics and Computation*, 2007, 185(2): 1026–1037. doi: [10.1016/j.amc.2006.07.025](https://doi.org/10.1016/j.amc.2006.07.025).
- [21] CHEN Ke, ZHOU Fengyu, and LIU Aling. Chaotic dynamic weight particle swarm optimization for numerical function optimization[J]. *Knowledge-Based Systems*, 2018, 139: 23–40. doi: [10.1016/j.knsys.2017.10.011](https://doi.org/10.1016/j.knsys.2017.10.011).
- [22] LONG Wen, JIAO Jianjun, LIANG Ximing, et al. An

- exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization[J]. *Engineering Applications of Artificial Intelligence*, 2018, 68: 63–80. doi: [10.1016/j.engappai.2017.10.024](https://doi.org/10.1016/j.engappai.2017.10.024).
- [23] MURO C, ESCOBEDO R, SPECTOR L, *et al.* Wolf-pack hunting strategies emerge from simple rules in computational simulations[J]. *Behavioral Processes*, 2011, 88(3): 192–197. doi: [10.1016/j.beproc.2011.09.006](https://doi.org/10.1016/j.beproc.2011.09.006).
- [24] MIRJALILI S, MIRJALILI S M, and LEWIS A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46–61. doi: [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007).
- [25] UDWADIA F E and GUTTALU R S. Chaotic dynamics of a piecewise cubic map[J]. *Physical Review A*, 1989, 40(7): 4032–4044. doi: [10.1103/PhysRevA.40.4032](https://doi.org/10.1103/PhysRevA.40.4032).
- [26] 龙文, 蔡绍洪, 焦建军, 等. 求解大规模优化问题的改进鲸鱼优化算法[J]. *系统工程理论与实践*, 2017, 37(11): 2983–2994. doi: [10.12011/1000-6788\(2017\)11-2983-12](https://doi.org/10.12011/1000-6788(2017)11-2983-12).
- LONG Wen, CAI Shaohong, JIAO Jianjun, *et al.* Improved whale optimization algorithm for large scale optimization problems[J]. *Systems Engineering-Theory & Practice*, 2017, 37(11): 2983–2994. doi: [10.12011/1000-6788\(2017\)11-2983-12](https://doi.org/10.12011/1000-6788(2017)11-2983-12).
- [27] ALI M H, MOHAMMED B A D A, ISMAIL A, *et al.* A new intrusion detection system based on fast learning network and particle swarm optimization[J]. *IEEE Access*, 2018, 6: 20255–20261. doi: [10.1109/ACCESS.2018.2820092](https://doi.org/10.1109/ACCESS.2018.2820092).
- [28] BLAKE C I and MERZ C J. UCI Repository of machine learning databases[OL]. <http://archive.ics.uci.edu/ml/datasets.html>, 1998.
- 晏福: 男, 1989年生, 博士生, 研究方向为智能优化算法、神经网络和数据挖掘。
- 徐建中: 男, 1959年生, 教授, 博士生导师, 研究方向为管理科学前沿研究。
- 李奉书: 男, 1989年生, 博士生, 研究方向为管理科学前沿研究。