

秃鹰搜索算法的优化及图像分类应用^①



刘世杰^{1,2}, 刘美², 孟亚男¹, 杨涛^{1,2}

¹(吉林化工学院 信息与控制工程学院, 吉林 132022)

²(广东石油化工学院 自动化学院, 茂名 525000)

通信作者: 刘美, E-mail: 879643579@qq.com

摘要: 针对秃鹰搜索算法 (BES) 存在求解的稳定性差且准确性低, 鲁棒性差等缺点, 提出了一种基于秃鹰搜索算法的新型算法 (NBES). 首先, 在 BES 算法的选择搜索空间阶段融合正余弦优化机制算法, 构建融合后的位置更新公式. 其次, 在 BES 算法的搜索空间猎物阶段加入惯性权重自适应位置更新策略. 最后, 在 BES 算法俯冲阶段融合萤火虫优化机制算法, 重新定义位置更新公式. 通过 11 个标准测试函数验证 NBES 算法性能, 实验表明, NBES 算法寻优准确性、收敛速度、鲁棒性都优于 BES 算法. 为了验证新算法的实际应用价值, 利用 NBES 算法优化卷积神经网络 (CNN) 中的超参数学习率, 并将优化后的图像分类模型用于医学影像病理性分类预测, 实验表明, 经过优化的 CNN 模型分类精度提高 9%.

关键词: 秃鹰搜索算法; 正余弦优化机制; 惯性权重自适应策略; 萤火虫优化机制; 测试函数; 图像分类

引用格式: 刘世杰, 刘美, 孟亚男, 杨涛. 秃鹰搜索算法的优化及图像分类应用. 计算机系统应用, 2023, 32(11): 182-192. <http://www.c-s-a.org.cn/1003-3254/9300.html>

Optimization of Bald Eagle Search Algorithm and Its Application in Image Classification

LIU Shi-Jie^{1,2}, LIU Mei², MENG Ya-Nan¹, YANG Tao^{1,2}

¹(School of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin 132022, China)

²(School of Automation, Guangdong University of Petrochemical Technology, Maoming 525000, China)

Abstract: A new algorithm based on the bald eagle search algorithm (NBES) is proposed to address the drawbacks of poor stability and low accuracy of the solution and poor robustness of the bald eagle search (BES) algorithm. First, the sine cosine optimization mechanism algorithm is fused in the search space selection stage of the BES algorithm, and the fused position update formula is constructed. Secondly, the inertial weight adaptive position update strategy is added in the search space prey phase of the BES algorithm. Finally, the position update formula is redefined by fusing the firefly optimization mechanism algorithm in the swoop phase of the BES algorithm. The performance of the NBES algorithm is verified by 11 standard test functions, and the experiments show that the NBES algorithm outperforms the BES algorithm in terms of search accuracy, convergence speed, and robustness. To verify the practical application value of the new algorithm, the hyperparameter learning rate in the convolutional neural network (CNN) is optimized by using the NBES algorithm, and the optimized image classification model is used in medical image pathology classification prediction, and the experiments show that the classification accuracy of the optimized CNN model is improved by 9%.

Key words: bald eagle search (BES) algorithm; sine cosine optimization mechanism; inertia weighting adaptive strategy; firefly optimization mechanism; test functions; image classification

① 基金项目: 国家自然科学基金 (62073091); 广东省普通高校重点领域 (新一代信息技术) 专项 (2020ZDZX3042)

收稿时间: 2023-04-24; 修改时间: 2023-05-23, 2023-06-14; 采用时间: 2023-06-26; csa 在线出版时间: 2023-09-15

CNKI 网络首发时间: 2023-09-19

1 引言

近年来,深度学习图像分类领域随着卷积神经网络(CNN)和Transformer等模型的逐步发展,已经取得了显著的成果,但图像分类算法仍然面临着大量超参数优化问题.超参数的设置会直接影响模型性能^[1-3],因此超参数优化显得至关重要.超参数优化包括经验调参、网格搜索、贝叶斯优化、群智能算法优化等方法.He等人^[4]通过经验调参法提高图像分类模型精度.虽然经验调参法可以提高模型性能,但需要花费巨大的时间成本,且很难找到最优的模型.文献[5,6]通过网格搜索优化超参数,提高网络模型性能.网格搜索方法简单易行,能够找到给定取值范围内的最优解,但是计算效率低下,并且需要人为确定超参数的取值范围.文献[7,8]通过贝叶斯优化超参数,获得高性能的分类模型,但是贝叶斯优化超参数计算复杂度高.文献[9-13]中通过群智能算法优化超参数,实验表明,该方法可以有效地提高分类模型的精度.群智能算法优化不仅能够处理大规模的复杂问题,不需要预先定义参数取值范围,并且具有全局优化能力.综上所述,群智能优化超参数存在优越性,故文中选取秃鹰搜索(bald eagle search, BES)算法^[4]优化CNN网络结构中的超参数,提高网络分类模型精度.秃鹰搜索算法为2020年提出的一种新型的元启发算法,虽然算法具有结构简单、收敛速度快、全局搜索能力较强等优点,在解决各类数值优化问题方面非常有效,但缺点是易陷入局部最优解,且跳出局部最优解的能力较弱,即算法在全局最优探索和局部利用之间存在不协调.针对原算法存在的缺点,学者通过Tent混沌映射^[15-17]、Sinusoidal映射^[18]、levy飞行策略^[19]等方法代替随机初始化种群,扩大种群前期的多样性;通过自适应t分布^[16]、反向学习机制^[17,20]、自适应惯性权重^[15,20,21]等策略来提高算法全局搜索能力,同时平衡算法的收敛速度与局部寻优能力.

为了解决BES存在的问题,文中引入了正余弦搜索机制、惯性权重自适应策略、萤火虫搜索机制对BES位置更新公式进行修改,提出一种基于BES的新型秃鹰搜索算法.实验表明,改进后的新型秃鹰搜索算法可以有效地改善全局搜索与局部搜索的不协调,提高算法寻优准确率和收敛速度.最后,将改进后的算法应用于优化CNN的超参数学习率,有效地提高分类模型的性能.

2 背景知识

秃鹰搜索算法整体思路是对秃鹰捕获猎物的行为进行模拟,整个算法可分为选择空间、搜索空间、俯冲这3个阶段.

1) 选择搜索空间阶段.该行为数学模型为:

$$X_{i,new} = X_{best} + \alpha \times r \times (X_{mean} - X_i) \quad (1)$$

其中,秃鹰当前位置为 $X_{i,new}$,当前最佳秃鹰位置为 X_{best} , α 表示控制位置变化参数,其变化范围为(1.5, 2); r 为(0, 1)之间的随机数; X_i 表示第 i 只秃鹰的位置; X_{mean} 为所有秃鹰的平均位置.

2) 搜索空间猎物阶段.相关公式如下:

$$\theta(i) = a \times \pi \times rand \quad (2)$$

$$r(i) = \theta(i) + R \times rand \quad (3)$$

$$xr(i) = r(i) \times \sin[\theta(i)] \quad (4)$$

$$yr(i) = r(i) \times \cos[\theta(i)] \quad (5)$$

$$x(i) = xr(i) / \max(|xr|) \quad (6)$$

$$y(i) = yr(i) / \max(|yr|) \quad (7)$$

$$X_{i,new} = X_i + x(i) \times (X_i - X_{mean}) + y(i) \times (X_i - X_{i+1}) \quad (8)$$

其中, $x(i)$ 与 $y(i)$ 表示极坐标下的秃鹰位置,取值范围均为(-1, 1); $rand$ 表示为(0, 1)之间的随机数; $\theta(i)$ 表示为螺旋方程的极角; $r(i)$ 表示为螺旋方程的极径; a 为控制螺旋轨迹中极角变化的大小,其变化范围为(5, 10); R 表示控制螺旋轨迹中极径的大小,其变化范围为(0.5, 2); X_{mean} 表示所有秃鹰位置的平均值.

3) 俯冲阶段.极坐标方程描述运动数学模型,具体如下:

$$\theta(i) = a \times \pi \times rand \quad (9)$$

$$r(i) = \theta(i) \quad (10)$$

$$x_1 r(i) = r(i) \times \sinh[\theta(i)] \quad (11)$$

$$y_1 r(i) = r(i) \times \cosh[\theta(i)] \quad (12)$$

秃鹰当前位置更新的数学模型如下:

$$x_2(i) = \frac{x_1 r(i)}{\max(|x_1 r|)} \times (X_i - c_1 \times X_{mean}) \quad (13)$$

$$y_2(i) = \frac{y_1 r(i)}{\max(|y_1 r|)} \times (X_i - c_2 \times X_{best}) \quad (14)$$

$$X_{i,new} = rand \times X_{best} + x_2 + y_2 \quad (15)$$

其中, c_1 与 c_2 分别表示秃鹰向最优点与中心点移动的强

度,且都属于 [1, 2] 之间的随机数值。

3 改进秃鹰搜索算法

3.1 正余弦优化机制

正余弦算法 (sine cosine algorithm, SCA)^[22] 作为一种新型的随机搜索算法,具有随机搜索算法的共性,其优化过程分为探索与利用两个阶段,具有较强地搜索空间、逃避局部最优的能力。其数学模型具体如下:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \times \sin(r_2) \times |r_3 \times P_i^t - X_i^t|, & r_4 < 0.5 \\ X_i^t + r_1 \times \cos(r_2) \times |r_3 \times P_i^t - X_i^t|, & r_4 \geq 0.5 \end{cases} \quad (16)$$

$$r_1 = h - h \times t / T \quad (17)$$

其中, h 为常数, t 表示当前迭代次数, T 表示最大迭代次数; r_2 为 $[0, 2\pi]$ 范围内的随机数; r_3 为 $[0, 2]$ 范围内的随机数; r_4 为 $[0, 1]$ 范围内的随机数; X_i^t 为第 t 次迭代时第 i 维位置; P_i^t 为目的地在 i 维的位置; X_i^{t+1} 为第 $t+1$ 次迭代时第 i 维位置。

秃鹰搜索算法通过一组随机值来求解全局最优解,寻优过程中起始阶段的最优值和位置状态会直接影响下一阶段寻优结果,如果起始状态以较少的随机值或局部最优范围内的随机值来启动算法,则较易陷入局部最优。因此利用正余弦优化机制具有高探索性和局部最优避免性来修正秃鹰搜索算法选择搜索空间阶段的位置更新公式,具体数学模型如下:

$$X_{i,\text{new}} = \begin{cases} (X_{\text{mean}} - X_i) \times r_5 + r_1 \sin(r_2) \times |r_3 X_{\text{best}} - X_i|, & r_4 < 0.5 \\ (X_{\text{mean}} - X_i) \times r_5 + r_1 \cos(r_2) \times |r_3 X_{\text{best}} - X_i|, & r_4 \geq 0.5 \end{cases} \quad (18)$$

其中, $X_{i,\text{new}}$ 表示第 i 只秃鹰的位置; X_{mean} 表示所有秃鹰位置的平均值; X_i 为第 i 只秃鹰的当前位置; X_{best} 表示当前秃鹰的最优位置; r_1 与式 (17) 中符号含义相同; r_2 、 r_3 、 r_4 与式 (16) 中符号含义相同; r_5 属于 $[0, 1]$ 范围内的值。

3.2 惯性权重自适应策略

群智能搜索算法通过适应度值来评估解的优劣性,然而随着迭代次数的增加,陷入局部最优的概率增大。因此需要通过适应度值,不断调整位置更新公式的权重参数值,平衡局部开发与全局搜索之间存在的冲突,防止陷入局部最优,加快求得全局最优解。

为了加强 BES 算法全局探索与后期局部利用能力,修改惯性权重自适应公式。改进后数学公式如下:

$$w_i^t = \begin{cases} w_{\min} + (w_{\max} - w_{\min}) \frac{fit(X_i^t) - fit_{\min}(X^t)}{fit_{\text{mean}}(X^t) - fit_{\min}(X^t)}, & fit(X_i^t) \leq fit_{\text{mean}}(X^t) \\ w_{\max} + r_6 e^{-t/T}, & fit(X_i^t) > fit_{\text{mean}}(X^t) \end{cases} \quad (19)$$

其中, w_{\max} 与 w_{\min} 分别表示预设的权重最大值与最小值,取 $w_{\max}=0.9$, $w_{\min}=0.4$; w_i^t 为第 t 次迭代个体 i 的权重值; $fit(X_i^t)$ 表示第 t 次迭代时个体 i 的适应度值; $fit_{\text{mean}}(X^t)$ 表示第 t 次迭代所对应的平均适应度值; $fit_{\min}(X^t)$ 表示第 t 次迭代所对应的最小适应度值; r_6 属于 $[0, 1]$ 范围内的常数; T 表示最大迭代次数。

利用式 (19) 改进 BES 算法搜索空间阶段的权重参数,调整个体和群体之间的联系强度。具体数学公式如下:

$$X_{i,\text{new}} = X_i + x(i) \times (X_i - X_{\text{mean}}) + w_i^t \times y(i) \times (X_i - X_{i+1}) \quad (20)$$

其中, $X_{i,\text{new}}$ 、 X_{mean} 、 X_i 与式 (18) 中符号含义相同, X_{i+1} 为第 i 只秃鹰下一阶段的位置; $x(i)$ 与式 (6) 中符号含义相同; $y(i)$ 与式 (7) 中符号含义相同; w_i^t 与式 (19) 中符号含义相同。

3.3 萤火虫优化机制

萤火虫算法 (firefly algorithm, FA)^[23] 作为新型搜索算法,性能处于粒子群优化算法与随机搜索算法之间。数学模型具体如下:

$$X_i^{t+1} = X_i + \beta_0 e^{-\gamma r_{i,j}^2} (X_i - X_j) + \sigma(\text{rand} - 0.5) \quad (21)$$

$$\sigma = \sigma_0 \times \theta^t \quad (22)$$

其中, γ 为常数且范围为 $[0.01, 100]$ 内的任意值; 式 (22) 一般取 $\sigma = 0.97^t$, t 表示当前的迭代次数; $r_{i,j}$ 表示当前萤火虫与全局最亮萤火虫之间的距离; X_i^{t+1} 表示萤火虫更新后位置; X_i 、 X_j 分别表示萤火虫 i 、 j 的位置。

由于 BES 算法存在全局探索与局部利用不平衡性问题,而萤火虫算法具有非常有效的求解方式,可以有效地找到全局最优解^[23]。因此利用式 (19) 和式 (21) 优化 BES 俯冲阶段的位置更新公式,具体数学模型如下:

$$X_{i,\text{new}} = X_i + x_2(i) \times e^{-w_i^t \gamma y_2^2(i)} \times (X_i - X_{\text{best}}) + \alpha \times (\text{rand} - 0.5) \quad (23)$$

其中, $X_{i,\text{new}}$ 、 X_{best} 、 X_i 与式 (18) 中符号含义相同; $x_2(i)$ 见式 (13); $y_2(i)$ 见式 (14); α 表示 $[0, 1]$ 范围内的任意值。

为了简化计算,提高寻优速度,式(23)根据泰勒展开式可近似于:

$$X_{i,new} = X_i + x_2(i) \times \frac{(X_i - X_{best})}{1 + w_i^t \gamma y_2^2(i)} + \alpha \times (rand - 0.5) \quad (24)$$

将经过以上改进后的秃鹰算法命名为新型秃鹰搜索(new bald eagle search, NBES).

3.4 NBES 算法步骤

新算法流程图如图1所示,具体步骤为 Step1–Step6. Step1: 种群初始化、算法参数设置. 种群数量设为 N ; 维度设为 D ; 探索空间上下边界分别设为 ub 与 lb , 最大寻优迭代次数设为 T . Step2: 根据目标函数计算个体适应度值, 依据适应度值判断个体的优劣. 更新秃鹰当前位置 $X_{i,new}$, 并找出全局最优适应度值所对应的最优位置值 X_{best} . Step3: 算法搜索空间选择阶段. 将当前的最优位置值 X_{best} 代入式(18)完成对当前秃鹰位置的更新. Step4: 算法探索空间猎物阶段. 利用式(20)完成当前阶段秃鹰位置更新, 同时保留当前阶段的位置值和全局最优值, 分别记为 $X_{i,new}$, X_{best} . Step5: 算法俯冲捕食阶段. 利用式(24)完成秃鹰当前位置的更新. Step6: 判断是否达到结束条件, 当满足输出条件时, 则输出当前的全局最优值, 否则重复 Step2–Step5 的步骤.

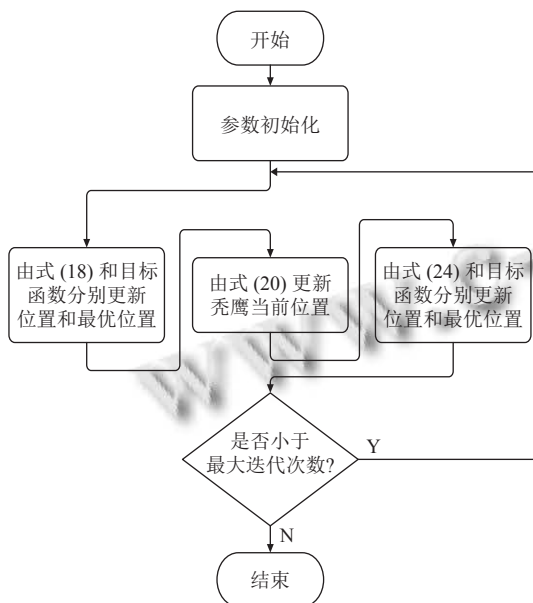


图1 NBES 算法流程图

4 算法实际应用

对图像分类模型进行研究时, CNN 模型则扮演了

一个非常重要的角色. 构建 CNN 模型时, 超参数的设置会直接影响模型性能. 由于 CNN 模型具有复杂性高、超参数多等现象, 人们往往需要通过试错策略和经验来选择最佳的超参数. 这种方法既耗时又费力, 并且不一定能够获得最优结果. 因此, 研究人员开始尝试在 CNN 模型中应用群智能算法对超参数进行自动化优化, 从而提高模型性能. 群智能算法通过对超参数进行优化搜索, 可以在短时间内找到最优的超参数值, 从而提高 CNN 模型的性能和训练效率. 具体应用场景包括医疗图像和人脸识别等领域. 因此, 群智能算法的应用在 CNN 模型性能优化中发挥着重要的作用, 它可以帮助我们更快地找到最优模型的超参数, 进一步提高图像分类的准确性和鲁棒性.

文中将通过多种群智能算法优化超参数学习率, 提高图像分类的识别精度. 同时对比不同群智能算法改进后的分类精度, 证明新算法 NBES 的实际应用价值.

4.1 模型构建

首先需要选取优质的医疗影像数据集, 文中使用的数据集为 Chest X-ray Database^[24,25]. 其次对选取的医疗影像数据集进行预处理, 处理方式主要包括数据集划分、数据清洗、数据增强. 最后根据医疗影像数据集搭建适合的卷积神经网络结构. 为了方便后期将模型进行 FPGA (field programmable gate array) 硬件部署, 则搭建小型的网络模型训练医疗影像数据集. 其模型结构如图2所示.

4.2 模型优化

模型优化方法包括数据优化、结构优化、损失函数优化、超参数优化等^[4]. 为了降低通过经验优化模型的难度与时间成本, 使用群智能算法优化学习率, 使学习率随模型的损失函数值变化而自动更新, 从而得到最优模型. 其中涉及的相关数学公式如下所示:

$$MSE(y_{-}, y) = \frac{1}{n} \sum_{i=1}^n (y - y_{-})^2 \quad (25)$$

$$W_{t+1} = W_t - I_{r,i} \times \frac{\partial MSE}{\partial W_t} \quad (26)$$

$$I_{r,i} = \frac{1}{n} \sum_{i=1}^n X_{i,best} \quad (27)$$

$$b = \sum_{i=1}^D i \times X_i^8 + rand \quad (28)$$

$$d = \frac{1}{1 + e^{-MSE}} \quad (29)$$

$$fitness5 = 0.5 + \frac{(\sin(b^2 + d^2))^{0.5}}{(1 + 0.001 \times (b^2 + d^2))^2} \quad (30)$$

$$R = \frac{TP}{TP + FN} \quad (31)$$

$$P = \frac{TP}{TP + FP} \quad (32)$$

其中, 均方误差 $MSE(y_{\hat{y}}, y)$ 表示模型损失值, D 表示群智能算法维度, W_t 和 W_{t+1} 分别表示 t 时刻与 $t+1$ 时刻的权重值, $I_{r,i}$ 与 $X_{i,best}$ 表示为第 i 时刻的学习率与最优值, $fitness5$ 表示适应度函数, TP 、 FN 、 FP 含义如表 1 所示, R 、 P 分别表示查全率与查准率.

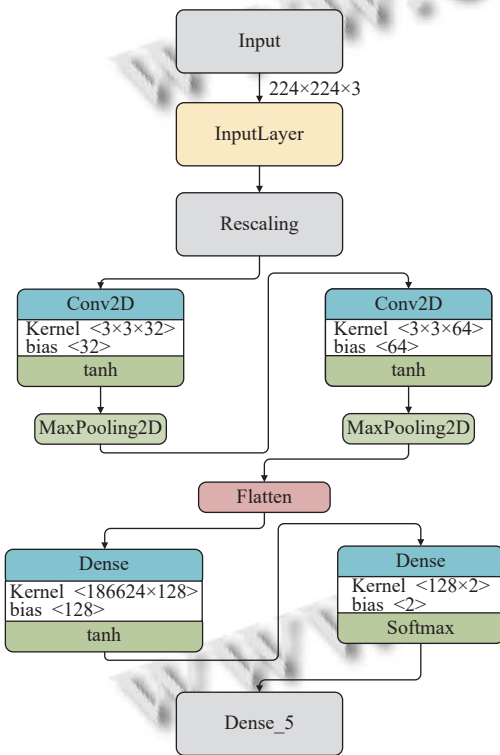


图 2 网络模型结构图

表 1 混淆矩阵

	预测为肺炎	预测正常
实际为肺炎	TP	FN
实际为正常	FP	TN

模型优化具体流程图如图 3 所示. 首先根据医疗图像数据集构建适合的深度神经网络, 模型每次 epoch

训练过程中, 会根据式 (25) 和式 (26) 输出当前的 loss 值和权重值, 将每次输出的 loss 值代入式 (30) 中求出适应度值, 以适应度值为指标不断更新模型的学习率, 其次判断当前模型训练次数是否小于设定的模型训练次数, 判断成立则将学习率输入卷积神经网络中继续训练, 反之结束模型训练并保存当前最优模型.

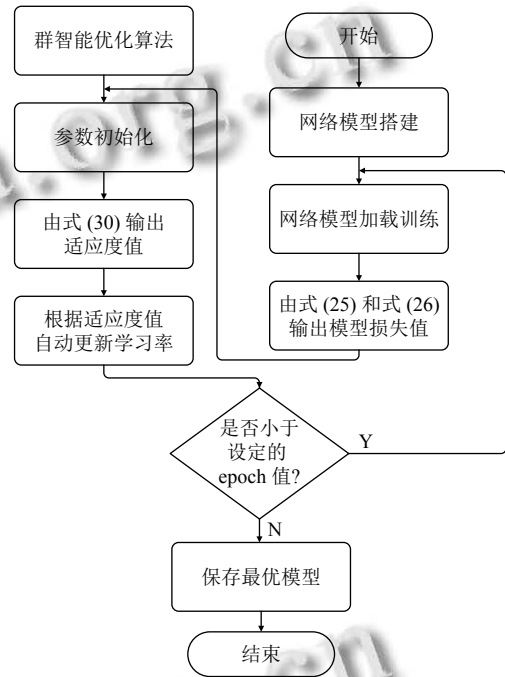


图 3 模型优化流程图

5 实验结果与分析

通过多种对比实验验证 NBES 算法性能. 首先对比 BES 算法在不同改进方案的效果, 保证改进方案的有效性, 将效果最好的算法记为 NBES; 其次将 NBES 算法与 BES、GSCBES^[21]、人工蜂群算法 (ABC)、粒子群算法 (PSO) 进行实验结果对比, 展现 NBES 算法性能. 最后利用 BES、NBES、GSCBES 优化超参数学习率, 通过对比实验效果, 展现 NBES 的应用性能.

5.1 改进方案有效性分析

为了体现出不同改进方案对算法性能的影响, 使用表 2 中 11 个标准测试函数^[26] 作为算法的目标函数. 其中 f_1-f_7 属于单峰函数, f_8-f_{11} 属于多峰函数. 单峰函数验证算法的收敛速率, 多峰函数验证算法的全局寻

优和逃脱局部最优的能力. 为了实验对比的公平性, 将 BES、NBES、GSCBES、ABC、PSO 算法中共有参数

设为相同值. 将其中最大迭代次数设为 500、维度设为 30、种群数量设为 30.

表 2 标准测试函数

Function	Type	Dimension	Range	f_{min}
$f_1(x) = \sum_{i=1}^n x_i^2$	单峰	30/100	[-100, 100]	0
$f_2(x) = \sum_{i=1}^n x_i - \prod_{i=1}^n x_i $	单峰	30/100	[-10, 10]	0
$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2$	单峰	30/100	[-100, 100]	0
$f_4(x) = \max_i(x_i), 1 \leq i \leq n$	单峰	30/100	[-100, 100]	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	单峰	30/100	[-10, 10]	0
$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	单峰	30/100	[-100, 100]	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + random[0, 1]$	单峰	30/100	[-1.28, 1.28]	0
$f_8(x) = \sum_{i=1}^n 418.9829i - x_i \sin(\sqrt{ x_i })$	多峰	30/100	[-500, 500]	0
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	多峰	30/100	[-5.12, 5.12]	0
$f_{10}(x) = -20e^{-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}} - e^{\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)} + 20 + e$	多峰	30/100	[-32, 32]	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	多峰	30/100	[-600, 600]	0

记 BES 算法加上正余弦优化机制为 BES1; BES 算法加上惯性权重自适应位置更新策略为 BES2; BES 算法加上正余弦优化机制和惯性权重自适应位置更新策略为 BES3; BES 算法加上惯性权重自适应位置更新策略和萤火虫优化机制为 BES4; BES 算法加上正余弦优化机制、惯性权重自适应策略、萤火

虫优化机制为 BESz (NBES). 为了体现每次改进方案的有效性, 对比 BES1、BES2、BES3、BES4、BESz (NBES) 算法的实验效果. 同理为了实验的公平性与专业性, 对不同算法的共有参数设置为相同值, 且进行 50 次重复性实验取均值. 具体的实验效果如表 3 所示.

表 3 改进方案效果对比

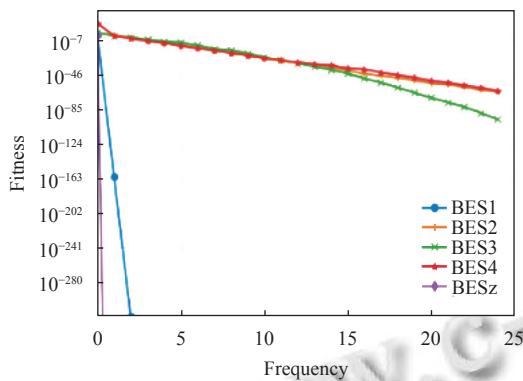
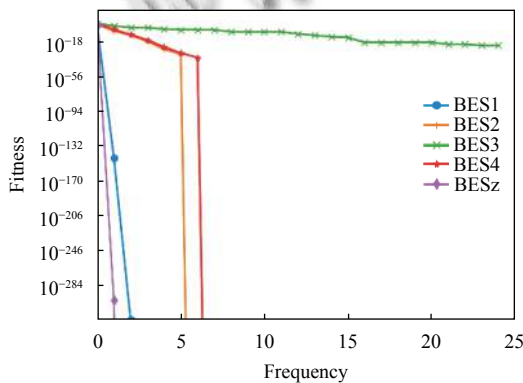
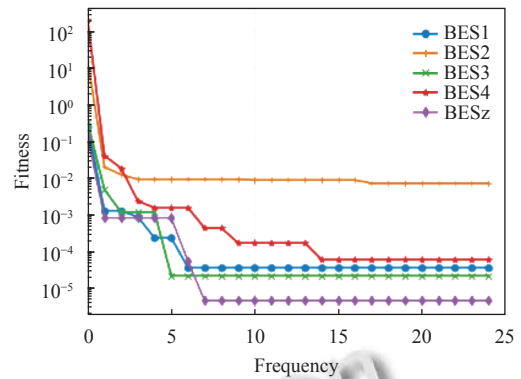
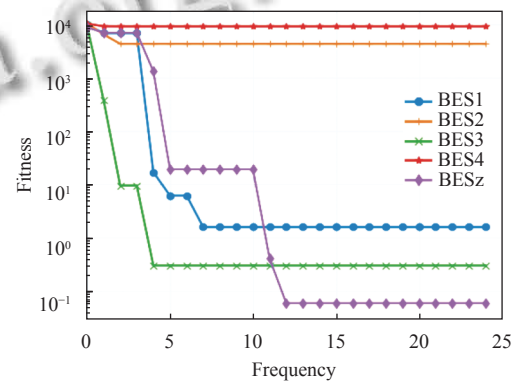
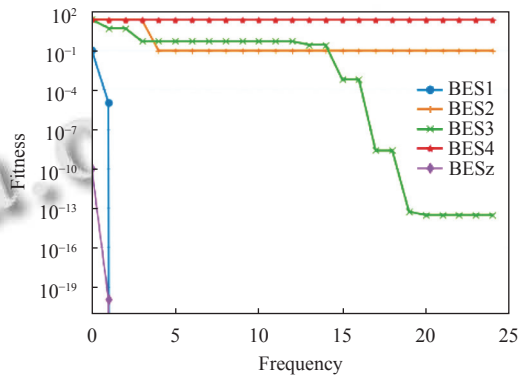
Function	Algorithm	Best	Mean	Standard	Time (s)
f_1	BES	1.41E-141	6.48E-119	3.49E-118	2.212
	BES1	0.00E+00	0.00E+00	0.00E+00	2.426
	BES2	5.63E-143	1.77E-116	1.21E-115	2.245
	BES3	2.88E-212	2.21E-187	0.00E+00	2.474
	BES4	9.12E-144	8.66E-127	4.89E-126	2.728
	BESz (NBES)	0.00E+00	0.00E+00	0.00E+00	2.369
f_2	BES	5.16E-71	6.30E-60	3.36E-59	2.564
	BES1	0.00E+00	0.00E+00	0.00E+00	2.692
	BES2	7.49E-73	5.85E-59	4.09E-58	2.318
	BES3	0.00E+00	0.00E+00	0.00E+00	2.522
	BES4	2.27E-72	3.84E-60	1.83E-59	2.621
	BESz (NBES)	0.00E+00	0.00E+00	0.00E+00	2.480

表3 (续) 改进方案效果对比

Function	Algorithm	Best	Mean	Standard	Time (s)
f_3	BES	0.00E+00	0.00E+00	0.00E+00	7.043
	BES1	0.00E+00	0.00E+00	0.00E+00	7.296
	BES2	0.00E+00	0.00E+00	0.00E+00	7.340
	BES3	0.00E+00	0.00E+00	0.00E+00	8.331
	BES4	0.00E+00	0.00E+00	0.00E+00	7.562
	BESz (NBES)	0.00E+00	0.00E+00	0.00E+00	7.216
f_4	BES	1.61E-74	1.34E-01	3.71E-01	2.299
	BES1	0.00E+00	0.00E+00	0.00E+00	2.567
	BES2	2.29E-75	1.75E-01	4.17E-01	2.196
	BES3	0.00E+00	0.00E+00	0.00E+00	2.529
	BES4	8.48E-75	1.61E-01	3.45E-01	2.523
	BESz (NBES)	0.00E+00	0.00E+00	0.00E+00	2.377
f_5	BES	4.66E-143	4.35E+00	2.34E+01	2.227
	BES1	0.00E+00	2.00E-01	7.66E-02	2.522
	BES2	7.64E-151	8.51E-03	5.10E-02	2.405
	BES3	0.00E+00	0.00E+00	0.00E+00	2.615
	BES4	3.09E-145	6.91E-03	3.34E-02	2.738
	BESz (NBES)	0.00E+00	0.00E+00	0.00E+00	2.518
f_6	BES	7.48E-139	4.56E-14	2.46E-13	2.181
	BES1	0.00E+00	0.00E+00	0.00E+00	2.338
	BES2	2.02E-146	1.10E-108	7.22E-108	2.269
	BES3	0.00E+00	0.00E+00	0.00E+00	2.647
	BES4	2.21E-143	4.82E-120	2.51E-119	2.578
	BESz (NBES)	0.00E+00	0.00E+00	0.00E+00	2.408
f_7	BES	4.74E-05	0.70E-03	0.20E-02	6.080
	BES1	1.06E-06	8.06E-05	7.42E-05	6.652
	BES2	2.06E-05	4.33E-04	4.78E-04	6.219
	BES3	1.79E-06	7.52E-05	6.81E-05	6.734
	BES4	2.01E-05	9.03E-04	1.99E-03	6.471
	BESz (NBES)	0.00E+00	5.47E-05	4.82E-05	6.148
f_8	BES	5.79E+03	8.35E+03	1.23E+03	5.162
	BES1	3.82E-04	9.11E-01	1.90E+01	5.228
	BES2	3.13E+03	8.42E+03	1.66E+03	5.092
	BES3	2.07E-03	1.64E+00	3.72E+00	5.282
	BES4	3.51E+03	8.26E+03	1.57E+03	5.355
	BESz (NBES)	5.44E-04	3.38E-01	9.97E-01	5.420
f_9	BES	0.00E+00	5.90E+00	3.12E+01	2.122
	BES1	0.00E+00	0.00E+00	0.00E+00	2.412
	BES2	0.00E+00	1.16E+00	4.58E+01	2.302
	BES3	0.00E+00	0.00E+00	0.00E+00	2.554
	BES4	0.00E+00	0.70E+00	4.36E+00	2.590
	BESz (NBES)	0.00E+00	0.00E+00	0.00E+00	2.588
f_{10}	BES	2.20E-19	2.56E+01	1.28E+01	9.786
	BES1	0.00E+00	0.00E+00	0.00E+00	10.20
	BES2	4.44E-16	6.30E-01	1.48E+00	10.28
	BES3	0.00E+00	0.00E+00	0.00E+00	10.78
	BES4	4.44E-16	3.46E-01	1.23E+00	10.81
	BESz (NBES)	0.00E+00	0.00E+00	0.00E+00	10.54
f_{11}	BES	0.00E+00	1.78E-01	5.62E-01	8.389
	BES1	0.00E+00	0.00E+00	0.00E+00	8.515
	BES2	0.00E+00	6.88E-02	3.28E-01	8.100
	BES3	0.00E+00	0.00E+00	0.00E+00	8.901
	BES4	0.00E+00	4.38E-02	3.06E-01	8.460
	BESz (NBES)	0.00E+00	0.00E+00	0.00E+00	8.050

由实验结果可知,不同改进方案下的算法在标准测试函数上都展现出不同层次的优越性.例如其中 BES1、BES3、BESz (NBES) 对测试函数寻优的稳定性和精准性都优于原算法,证明这3种改进方案不仅提高了原算法的全局寻优能力,也提高了算法逃脱局部最优的能力. BES2 在 f_1 、 f_3 – f_7 函数上稳定性和准确性优于原算法,在 f_2 、 f_4 上准确性优于原算法,在 f_9 – f_{10} 上稳定性优于原算法. BES4 在 f_1 – f_2 、 f_5 – f_7 、 f_{11} 函数上稳定性和准确性优于原算法, f_4 、 f_8 – f_9 函数上准确性优于原算法, f_{10} 函数上稳定性优于原算法. 相比于原算法, BES2 与 BES4 实验结果证明这两种改进方案针对不同目标函数,展现出不同层次的优越性.

为了突显出不同算法对于标准测试函数的寻优性能,选取5个测试函数进行图像展示,其中包括3个单峰函数和2个多峰函数.由图4–图8可知, BESz (NBES) 算法对于测试函数的寻优准确率和寻得最优解的速度都远优于 BES1、BES2、BES3、BES4. 为了使得线图标记更加清晰,能够快速区分出不同算法所对应的线条,对迭代次数进行20倍等比例缩小,实际迭代次数 (frequency) 为500次. 综上所述 NBES 算法的整体寻优性能最为优越.

图4 f_2 效果对比图图5 f_3 效果对比图图6 f_7 效果对比图图7 f_8 效果对比图图8 f_{10} 效果对比图

5.2 与不同群智能算法对比

在维度为30与100的条件下,对比NBES、PSO、ABC、GSCBES、BES算法的实验效果,验证新算法的性能.具体的实验效果如表4所示.

首先从相同算法但不同维度的角度分析实验结果.随着维度的不断增大,群智能优化算法的整体性能会呈现下降趋势,这是由于随着维度的增大求解空间随着变大,导致其求解难度也加大.其中 PSO、ABC、

GSCBES、BES 算法随着维度增大, 其对测试函数求解的稳定性、准确性、收敛速率都变差; 而 NBES 算法在 f_1 - f_6 函数和 f_9 - f_{11} 函数上, 除了寻优的单次运行时间变长之外, 其稳定性和准确性并未随着维度的增大而变差, 在 f_7 - f_8 函数上其寻优性能略微下降. 其次从不同算法但相同维度的角度分析实验结果. 对 f_1 - f_{11} 函数

整体寻优性能最好的是 NBES. 综上所述, NBES 对于标准测试函数的寻优求解性能整体优于 PSO、ABC、GSCBES、BES 算法.

5.3 应用效果对比

通过实验结果展现群智能算法在图像分类领域的应用价值, 设计原理如图 3 所示.

表 4 不同的维度和算法效果对比

Function	Algorithm	Dim=30			Dim=100		
		Mean	Standard	Time (s)	Mean	Standard	Time (s)
f_1	PSO	1.41E-01	3.93E-02	0.122	3.45E+01	1.20E+01	0.160
	ABC	5.42E-14	3.93E-14	4.734	1.66E+02	3.74E+01	10.24
	BES	6.48E-119	3.49E-118	2.212	2.10E+00	1.12E+01	4.855
	GSCBES	3.62E-94	2.45E-93	2.561	7.32E-76	4.86E-75	4.814
	NBES	0.00E+00	0.00E+00	2.369	0.00E+00	0.00E+00	5.116
f_2	PSO	2.43E+00	1.0E+00	0.190	3.28E+01	5.66E+00	0.199
	ABC	6.85E+01	1.78E+01	5.822	4.64E+06	5.67E+05	5.592
	BES	6.30E-60	3.36E-59	2.564	7.84E+01	5.36E+01	4.629
	GSCBES	4.55E-54	3.18E-53	2.555	1.99E-49	1.39E-48	5.143
	NBES	0.00E+00	0.00E+00	2.480	0.00E+00	0.00E+00	5.202
f_3	PSO	0.00E+00	0.00E+00	5.477	0.00E+00	0.00E+00	18.09
	ABC	4.53E-04	7.43E-04	19.20	3.96E-03	6.43E-03	59.02
	BES	0.00E+00	0.00E+00	7.043	0.00E+00	0.00E+00	26.06
	GSCBES	0.00E+00	0.00E+00	7.490	0.00E+00	0.00E+00	21.79
	NBES	0.00E+00	0.00E+00	7.216	0.00E+00	0.00E+00	22.44
f_4	PSO	2.55E+00	1.31E+00	0.123	9.93E+00	1.20E+00	0.137
	ABC	5.99E+01	4.37E+00	3.863	9.44E+01	1.13E+00	5.733
	BES	1.34E-01	3.71E-01	2.299	1.44E-01	3.15E-01	4.862
	GSCBES	2.53E-01	5.89E-01	2.480	2.54E-01	6.15E-01	5.369
	NBES	0.00E+00	0.00E+00	2.377	0.00E+00	0.00E+00	5.110
f_5	PSO	2.73E+01	5.74E+01	0.231	8.90E+02	2.41E+02	0.239
	ABC	3.42E+04	1.25E+04	3.686	1.30E+06	8.87E+05	6.809
	BES	4.35E+00	2.34E+01	2.227	6.24E+04	2.77E+05	4.926
	GSCBES	3.00E+01	9.41E+00	2.544	1.04E+03	2.64E+01	4.903
	NBES	0.00E+00	0.00E+00	2.518	0.00E+00	0.00E+00	5.195
f_6	PSO	1.44E-01	3.28E-02	0.140	3.75E+01	1.56E+01	0.170
	ABC	1.71E+02	3.57E+01	2.614	2.46E+05	1.21E+04	5.756
	BES	4.56E-14	2.46E-13	2.181	3.49E+00	1.88E+01	4.208
	GSCBES	1.85E+00	1.03E+01	2.432	1.26E+01	5.04E+01	4.772
	NBES	0.00E+00	0.00E+00	2.408	0.00E+00	0.00E+00	5.300
f_7	PSO	2.41E-01	1.11E-01	3.994	9.46E+01	3.78E+01	17.38
	ABC	3.99E+01	1.20E+01	15.32	3.48E+03	2.12E+02	52.26
	BES	0.70E-03	0.20E-02	6.080	0.78E-03	0.20E-02	20.77
	GSCBES	8.28E-04	1.66E-03	6.505	1.44E-03	3.64E-03	22.10
	NBES	5.47E-05	4.82E-05	6.148	7.70E-05	7.03E-05	19.30
f_8	PSO	9.49E+03	3.97E+02	3.175	3.61E+05	8.06E+03	11.71
	ABC	7.99E+79	2.91E+80	14.05	2.09E+80	7.71E+80	47.56
	BES	8.35E+03	1.23E+03	5.162	3.15E+04	5.24E+03	15.53
	GSCBES	8.50E+03	1.14E+03	5.595	3.10E+05	6.23E+04	16.25
	NBES	3.38E-01	9.97E-01	5.420	3.20E+00	1.11E+01	15.86
f_9	PSO	4.40E+01	1.26E+01	0.189	3.50E+02	3.58E+01	0.227
	ABC	2.82E+02	1.41E+01	2.639	1.56E+03	3.65E+01	6.386
	BES	5.90E+00	3.12E+01	2.122	3.27E+01	1.51E+02	4.244
	GSCBES	1.08E+01	5.00E+01	2.492	1.26E+01	5.03E+01	4.906
	NBES	0.00E+00	0.00E+00	2.588	0.00E+00	0.00E+00	4.937

表4(续) 不同的维度和算法效果对比

Function	Algorithm	Dim=30			Dim=100		
		Mean	Standard	Time (s)	Mean	Standard	Time (s)
f_{10}	PSO	2.03E+01	9.36E-02	7.437	2.01E+01	2.92E-02	24.22
	ABC	1.99E+01	5.56E-03	25.63	2.00E+01	2.82E-02	75.42
	BES	2.56E+01	1.28E+01	9.786	2.74E+01	1.65E+01	30.01
	GSCBES	1.77E+01	6.08E+00	9.600	1.61E+01	7.71E+00	31.08
	NBES	0.00E+00	0.00E+00	10.54	0.00E+00	0.00E+00	30.06
f_{11}	PSO	1.67E+02	1.78E+01	19.13	7.81E+03	3.09E+01	68.57
	ABC	6.02E+01	1.28E+01	20.81	2.25E+03	1.14E+02	60.11
	BES	1.78E-01	5.62E-01	8.389	1.79E-01	5.74E-01	24.20
	GSCBES	1.75E-01	5.95E-01	7.934	3.82E-01	1.47E+00	24.32
	NBES	0.00E+00	0.00E+00	8.050	0.00E+00	0.00E+00	24.47

当未使用群智能算法优化学习率时,模型经过100轮次训练后,得到分类精度稳定在0.788。经过BES、GSCBES、NBES优化后的CNN,将其分别记为BES_CNN、GSCBES_CNN、NBES_CNN。表5中的“Best accuracy”指进行100轮次模型训练的过程中,测试集中出现的最高分类精度。“Final accuracy”则指代100轮次模型训练后,在测试集上达到的最终稳定的分类精度。“Test loss”表示测试集的损失值。“Avg accuracy”

为3次重复实验,测试集精度的平均值。“R”表示查全率。“P”表示查准率。由实验可知,当以不同的算法优化学习率或以不同的目标函数作为寻优指标时,可以得到不同的分类模型精度。由表5的实验结果可知,当以fitness5函数为目标函数,NBES_CNN为模型时,其分类模型的精度、查准率、查全率最高,代表这种情况下,获取的模型性能最好。综上所述,新算法NBES在优化医疗图像分类模型上展现出较好的实际应用价值。

表5 不同的算法优化效果对比

Function	Model	Best accuracy	Final accuracy	Test loss	Avg accuracy	R	P
$fitness1 = \frac{b+d}{2}$	BES_CNN	0.856	0.811	2.45	0.826	0.851	0.791
	GSCBES_CNN	0.856	0.826	1.67	0.836	0.851	0.814
	NBES_CNN	0.864	0.833	1.67	0.854	0.851	0.826
$fitness2 = (b^2 + d^2)^{0.5}$	BES_CNN	0.856	0.848	2.45	0.851	0.836	0.862
	GSCBES_CNN	0.864	0.803	1.68	0.823	0.851	0.781
	NBES_CNN	0.864	0.856	1.68	0.859	0.851	0.846
$fitness3 = 10(d^2 + 4d)^2 + 10(1 - 4d)^2$	BES_CNN	0.864	0.833	2.45	0.836	0.821	0.846
	GSCBES_CNN	0.856	0.833	1.68	0.841	0.851	0.826
	NBES_CNN	0.864	0.833	1.68	0.843	0.851	0.826
$fitness4 = 4b^2 - 2.1b^4 + 0.3b^6 + bd - 4b^2 + 4b^4$	BES_CNN	0.841	0.833	2.45	0.836	0.836	0.836
	GSCBES_CNN	0.864	0.788	1.68	0.828	0.866	0.753
	NBES_CNN	0.864	0.841	1.67	0.849	0.851	0.851
$fitness5 = 0.5 + \frac{(\sin(b^2 + d^2))^{0.5}}{(1 + 0.001 \times (b^2 + d^2))^2}$	BES_CNN	0.856	0.841	2.46	0.846	0.851	0.838
	GSCBES_CNN	0.864	0.841	1.68	0.849	0.851	0.838
	NBES_CNN	0.871	0.871	1.68	0.871	0.881	0.868

6 结语

本文基于秃鹰搜索算法提出一种新型算法,该算法在原算法的基础上融合正余弦优化机制、惯性权重自适应位置更新策略、萤火虫优化机制,使得算法前期全局探索范围增大,中后期平衡全局探索与局部利用,使算法不仅拥有极强地全局探索能力和局部利用能力,且算法稳定性、准确性、收敛性都极强。为了体现新算法的实用性,利用新算法优化卷积神经网络的超参数学习率,达到自动寻找最佳的分类模型,并将模型用于医疗影像病理性分类预测中,实验表明,新算法

NBES的模型优化效果优于其他群智能算法。

NBES不仅可以用于优化分类模型的超参数,也可以优化分割模型、识别模型、检测模型中的各种超参数。例如优化超参数正则化参数、小批量数据大小、模型训练次数、神经网络层数和神经元个数等。

参考文献

- Feurer M, Hutter F. Hyperparameter optimization. Automated Machine Learning: Methods, Systems, Challenges. Cham: Springer, 2019. 3-33.

- 2 Kohavi R, John GH. Automatic parameter selection by minimizing estimated error. Proceedings of the 1995 Conference on Machine Learning. Amsterdam: Elsevier, 1995. 304–312.
- 3 Yang L, Shami A. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 2020, 415: 295–316. [doi: [10.1016/j.neucom.2020.07.061](https://doi.org/10.1016/j.neucom.2020.07.061)]
- 4 He T, Zhang Z, Zhang H, *et al.* Bag of tricks for image classification with convolutional neural networks. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 558–567.
- 5 Liu RM, Liu EQ, Yang J, *et al.* Optimizing the hyperparameters for SVM by combining evolution strategies with a grid search. Proceedings of the 2006 International Conference on Intelligent Computing on Intelligent Control and Automation. Kunming: Springer, 2006. 712–721.
- 6 Prabu S, Thiyaneswaran B, Sujatha M, *et al.* Grid search for predicting coronary heart disease by tuning hyperparameters. Computer Systems Science and Engineering, 2022, 43(2): 737–749. [doi: [10.32604/csse.2022.022739](https://doi.org/10.32604/csse.2022.022739)]
- 7 Klein A, Falkner S, Bartels S, *et al.* Fast Bayesian optimization of machine learning hyperparameters on large datasets. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Fort Lauderdale: PMLR, 2017. 528–536.
- 8 Wu J, Chen XY, Zhang H, *et al.* Hyperparameter optimization for machine learning models based on Bayesian optimization. Journal of Electronic Science and Technology, 2019, 17(1): 26–40.
- 9 Aszemi NM, Dominic PDD. Hyperparameter optimization in convolutional neural network using genetic algorithms. International Journal of Advanced Computer Science and Applications, 2019, 10(6): 269–278.
- 10 Ayumi V, Rere LMR, Fanany MI, *et al.* Optimization of convolutional neural network using microcanonical annealing algorithm. Proceedings of the 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS). Malang: IEEE, 2016. 506–511.
- 11 Serizawa T, Fujita H. Optimization of convolutional neural network using the linearly decreasing weight particle swarm optimization. Proceedings of the 36th Annual Conference of the Japanese Society for Artificial Intelligence. Kyoto: JSAI, 2022. 1–5.
- 12 Singh P, Chaudhury S, Panigrahi BK. Hybrid MPSO-CNN: Multi-level particle swarm optimized hyperparameters of convolutional neural network. Swarm and Evolutionary Computation, 2021, 63: 100863. [doi: [10.1016/j.swevo.2021.100863](https://doi.org/10.1016/j.swevo.2021.100863)]
- 13 Yamasaki T, Honma T, Aizawa K. Efficient optimization of convolutional neural networks using particle swarm optimization. Proceedings of the 3rd IEEE International Conference on Multimedia Big Data (BigMM). Laguna Hills: IEEE, 2017. 70–73.
- 14 Alsattar HA, Zaidan AA, Zaidan BB. Novel meta-heuristic bald eagle search optimisation algorithm. Artificial Intelligence Review, 2020, 53(3): 2237–2264. [doi: [10.1007/s10462-019-09732-5](https://doi.org/10.1007/s10462-019-09732-5)]
- 15 丁容, 高建瓴, 张倩. 融合自适应惯性权重和柯西变异的秃鹰搜索算法. 小型微型计算机系统, 2023, 44(5): 910–915.
- 16 耿棒棒, 王勇. 基于改进秃鹰搜索算法的 Kubernetes 资源调度应用. 计算机系统应用, 2023, 32(4): 187–196. [doi: [10.15888/j.cnki.csa.009034](https://doi.org/10.15888/j.cnki.csa.009034)]
- 17 莫仕勋, 杨皓, 蒋坤坪, 等. 基于改进秃鹰搜索算法的变压器 J-A 模型参数辨识. 电工电能新技术, 2022, 41(4): 67–74.
- 18 杨国华, 冯骥, 柳萱, 等. 基于改进秃鹰搜索算法的含分布式电源配电网分区故障定位. 电力系统保护与控制, 2022, 50(18): 1–9.
- 19 黄维, 吕丽平. 基于改进秃鹰搜索算法的工业机器人几何参数标定方法. 机械设计与研究, 2022, 38(6): 21–24, 33.
- 20 陈将宏, 胡炆, 饶佳黎, 等. 基于自适应反向学习秃鹰搜索算法的最优潮流计算. 电工材料, 2023, (1): 85–93.
- 21 赵沛雯, 张达敏, 张琳娜, 等. 融合黄金正弦算法和纵横交叉策略的秃鹰搜索算法. 计算机应用, 2023, 43(1): 192–201.
- 22 Mirjalili S. SCA: A sine cosine algorithm for solving optimization problems. Knowledge-based Systems, 2016, 96: 120–133. [doi: [10.1016/j.knosys.2015.12.022](https://doi.org/10.1016/j.knosys.2015.12.022)]
- 23 Yang XS, He XS. Firefly algorithm: Recent advances and applications. International Journal of Swarm Intelligence, 2013, 1(1): 36–50. [doi: [10.1504/IJSI.2013.055801](https://doi.org/10.1504/IJSI.2013.055801)]
- 24 Jaeger S, Karargyris A, Candemir S, *et al.* Automatic tuberculosis screening using chest radiographs. IEEE Transactions on Medical Imaging, 2014, 33(2): 233–245. [doi: [10.1109/TMI.2013.2284099](https://doi.org/10.1109/TMI.2013.2284099)]
- 25 Candemir S, Jaeger S, Palaniappan K, *et al.* Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. IEEE Transactions on Medical Imaging, 2014, 33(2): 577–590. [doi: [10.1109/TMI.2013.2290491](https://doi.org/10.1109/TMI.2013.2290491)]
- 26 Yao X, Liu Y, Lin GM. Evolutionary programming made faster. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82–102. [doi: [10.1109/4235.771163](https://doi.org/10.1109/4235.771163)]

(校对责编: 牛欣悦)