A Novel Two-Level Clustering-Based Differential Evolution Algorithm for Training Neural Networks

Seyed Jalaleddin Mousavirad[1(✉)], Diego Oliva[2], Gerald Schaefer[3], Mahshid Helali Moghadam[4], and Mohammed El-Abd[5]

[1] Mid Sweden University, Sundsvall, Sweden
jalalmousavirad@gmail.com

[2] Depto. de Ingeniería Electro -Fotónica, Universidad de Guadalajara, CUCEI, Guadalajara, Mexico

[3] Department of Computer Science, Loughborough University, Loughborough, UK

[4] Mälardalens University, Västerås, Sweden

[5] College of Engineering and Applied Sciences, American University of Kuwait, Salmiya, Kuwait

Abstract. Determining appropriate weights and biases for feed-forward neural networks is a critical task. Despite the prevalence of gradient-based methods for training, these approaches suffer from sensitivity to initial values and susceptibility to local optima. To address these challenges, we introduce a novel two-level clustering-based differential evolution approach, C2L-DE, to identify the initial seed for a gradient-based algorithm. In the initial phase, clustering is employed to detect some regions in the search space. Population updates are then executed based on the information available within each region. A new central point is proposed in the subsequent phase, leveraging cluster centres for incorporation into the population. Our C2L-DE algorithm is compared against several recent DE-based neural network training algorithms, and is shown to yield favourable performance.

Keywords: Differential evolution · clustering · neural network training · regularisation

1 Introduction

Feed-forward neural networks (FFNNs) are a widely adopted artificial neural network (ANN) architecture employed in diverse classification and regression problems [11,27]. Comprising basic components known as neurons and connections linking them, FFNNs allow the flow of information from the input layer through hidden layers, ultimately reaching the output layer. Each connection is characterised by a weight that signifies its strength. The training process in FFNNs aims to determine optimal weights that minimise the error between actual and predicted outputs. Although gradient-based approaches such as the

S. Smith et al. (Eds.): EvoApplications 2024, LNCS 14634, pp. 259–272, 2024.
https://doi.org/10.1007/978-3-031-56852-7_17

back-propagation (BP) algorithm are prevalent, they tend towards local optima and thus provide sub-optimal results [30].

Population-centric metaheuristic (PCM) algorithms, such as differential evolution (DE) [40] and particle swarm optimisation (PSO) [38], provide a useful alternative to address the challenges encountered by traditional algorithms. Evolutionary algorithms (EAs) are group of PCMs that has been widely applied in the training of FFNNs. [37] compares (BP) with a genetic algorithm (GA) for FFNN training, concluding that the latter excels in terms of effectiveness. [13] uses a modified GA for rapidly training FFNNs, demonstrating superior efficiency compared to conventional GA-based training algorithms. [7] proposes a hybrid approach combining GA and BP for determining the weights in FFNNs, outperforming both GA and BP individually.

Swarm intelligence algorithms form another group of PCMs. [34] combines PSO with the Levenberg-Marquardt (LM) algorithm to achieve faster convergence. [39] introduces an opposition PSO-based training method and evaluates it on various clinical datasets. [31] proposes a comprehensive learning strategy integrated with PSO and LM as a local search algorithm for neural network training. Various other PCM algorithms have been applied for FFNN training, including the imperialist competitive algorithm (ICA) [8,22], the firefly algorithm (FA) [17], the grey wolf optimiser (GWO) [2,19], and Lévy flight distribution [3,35], among others.

Differential evolution is a well-established PCM renowned for its outstanding performance in addressing complex optimisation problems [5,10,20,21]. It comprises three primary operators: mutation, crossover, and selection. Mutation facilitates the exchange of information among different individuals, crossover integrates a mutant vector with a target vector, and selection chooses superior individuals from old and new individuals into a new population.

DE has also been widely employed for FFNN training. [12] introduces a DE-based training algorithm, showcasing its ability to outperform gradient-based methods. [28] incorporates opposition-based learning into DE, demonstrating good performance across various classification problems. [32] employs an improved DE algorithm that incorporates opposition-based learning and a region-based strategy, while [36] proposes a centroid-based differential evolution algorithm with composite trial vector generation strategies and control parameters to optimise the weights and biases in FFNNs. In [24], a clustering-based DE approach for neural network training is employed.

In a recent enhancement to DE, [29] introduces a methodology involving centre-based sampling at the population level of DE, with the centre of the entire population incorporated as a new individual. Integrating the centre point is shown to effectively guide the population towards improved individuals. On the other hand, [4] indicates that cluster centres in a population are viable candidates in the search space to move towards. Building upon these two concepts, in this paper, we propose a novel two-level clustering-based differential evolution algorithm, C2L-DE, for training FFNNs. At the first level, the clustering algorithm works like a multi-parent crossover to update the population. In contrast, at the second level, the central point of population clustering is injected as a new individual into the population.

The main characteristics of C2L-DE are:

- a clustering strategy is employed at the first level to update the population;
- clustering is used to introduce a new individual into the population at the second level;
- a regularisation term is incorporated into the objective function to enhance generalisation;
- the weights and biases determined by C2L-DE are fed into the Levenberg-Marquardt algorithm as the initial seed.

The remainder of the paper is organised as follows: Section 2 gives an overview of some essential concepts. Section 3 presents our proposed approach, detailing the fundamental components of C2L-DE and explaining its overall structure. In Sect. 4, the performance of C2L-DE is assessed across various benchmark problems, while Sect. 5 concludes the paper.

2 Background

2.1 Differential Evolution

Differential evolution (DE) [40] is a straightforward yet highly effective PCM algorithm widely recognised for excellent performance in addressing complex optimisation problems [10,41]. DE begins with N_P individuals randomly generated from a uniform distribution. To update the population, three primary operators are employed: mutation, crossover, and selection.

The mutation operator produces a mutant vector, $v_i = (v_{i,1}, v_{i,2}, ..., v_{i,D})$, defined as

$$v_i = x_{r1} + F(x_{r2} - x_{r3}), \tag{1}$$

where x_{r1}, x_{r2}, and x_{r3} are three distinct randomly chosen individuals from the current population, and F represents a scale factor.

Crossover is responsible for incorporating the mutant vector into the target vector. For binomial crossover, this is performed as

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } \mathrm{rand}(0,1) \leq CR \text{ or } j == j_{rand} \\ x_{i,j} & \text{otherwise} \end{cases}, \tag{2}$$

where CR denotes the crossover rate, j_{rand} is a random number ranging from 1 to N_P, and $i = 1, ..., NP$, $j = 1, ..., D$.

Selection identifies the superior individual from the trial and target vectors, ensuring the progression of more promising solutions in the population.

The iterative process enhances the algorithm's ability to effectively explore and exploit the search space.

2.2 Pattern Clustering

The fundamental aim of clustering is to arrange a collection of patterns so that the members within each group share more similarities than those in different groups. Mathematically, clustering involves defining a set P consisting of N d-dimensional patterns, denoted as $P = \{p_1, p_2, \cdots, p_N\}$. The k-means algorithm [16] is the most widely adopted clustering algorithm and proceeds in the following steps:

1. Randomly initialise the cluster centres;
2. In the allocation step, assign each pattern to its nearest cluster centre (e.g., using Euclidean distance);
3. In the update step, recalculate the position of each cluster centre as the centroid of its assigned patterns;
4. Repeat steps 2 and 3 until convergence or a predefined stopping criterion is met.

2.3 Feed-Forward Neural Networks

FFNNs, a widely employed class of ANNs, are trained in a supervised manner to handle pattern recognition problems [1,33]. The typical architecture of an FFNN consists of three types of layers: an input layer, one or more hidden layers, and an output layer. Each node in these layers incorporates an activation function that defines how the weighted sum of inputs transforms into the output. The connections between layers are assigned weights, indicating the strength between the respective nodes. Weights are critical for FFNN performance, making determining suitable weight values one of the most vital and challenging aspects of FFNNs. Among various approaches, gradient descent-based methods form the most widely adopted technique for this training process.

3 Proposed C2L-DE Algorithm

Our proposed C2L-DE algorithm leverages clustering at two distinct levels. At the first level, specific individuals are substituted with cluster centres, while at the second level, the central point of a cluster centre is introduced as a new individual into the population. Additionally, our proposed algorithm incorporates a regularisation-based objective function to enhance the generalisation capabilities of the algorithm.

3.1 First-Level Clustering

At the first level, C2L-DE employs a clustering algorithm to construct areas in search space using the k-means algorithm. Determining the number of clusters is accomplished by selecting a random number within the range of 2 to $\sqrt{N_P}$. The resulting cluster centres are analogous to a multi-parent crossover, representing the cumulative solutions within a cluster.

C2L-DE's population update strategy involves adopting a generic population-based algorithm (GPBA) [6]. This approach aligns with a GPBA methodology and encompasses the folloingw steps:

- **Diversity selection:** individuals are randomly chosen from the current population, mirroring the initialisation of points in the k-means algorithm;
- **Clustered generation:** k-means is applied to generate m individuals (set A). Each cluster centre determined through this process corresponds to a new individual;
- **Individual substitution:** from the current population, m individuals (set B) are (randomly) selected for substitution;
- **Elite update:** the best m individuals from the combined set $A \cup B$ are selected as $\bar{B}$, and the new population is formed as $(P - B) \cup \bar{B}$.

This population update procedure integrates elements from clustering algorithms and population-based strategies, ensuring an effective and dynamic approach in C2L-DE.

It is important to note that C2L-DE does not employ the clustering algorithm in each iteration. Instead, following [5,32], clustering is applied periodically based on a clustering period.

3.2 Second-Level Clustering

DE-centre-p [29] is a centre-based DE algorithm where an individual, determined by the central point defined as the centre of the N best individuals, is introduced as a new member of the population. The population is then divided into two parts, one a set of individuals that undergo positional updates through standard mutation and crossover operations, and one that is an individual exclusively devoted to preserving the centre of the N best individuals. On the other hand, [4] suggests that cluster centres within a population represent promising candidates in search space, in particular for directional movement. Consequently, at the second level of our proposed clustering scheme, we introduce a novel approach for incorporating a new individual into the population based on cluster centres.

Following the initial clustering phase, the N most promising areas are identified using a one-step k-means algorithm. The value of N is not fixed and is randomly chosen between 2 and $\sqrt{N_P}$. Cluster centres serve as representatives for each cluster. Subsequently, the central point of these cluster centres is selected as a new individual, obtained as

$$\overrightarrow{x_{centre}} = \frac{\overrightarrow{x_{c1}} + ...\overrightarrow{x_{ci}} + ... + \overrightarrow{x_{cN}}}{N}, \tag{3}$$

where x_{ci} is the i-th cluster centre. While DE-centre-p injects this individual into the population with a fixed location, in C2L-DE we dynamically select this location based on the objective function. In other words, this new individual replaces the worst individual and endeavours to substitute the least favourable solution with the central point derived from several promising candidates within the population. Figure 1 illustrates the process creating a new individual based on the central point of cluster centres.

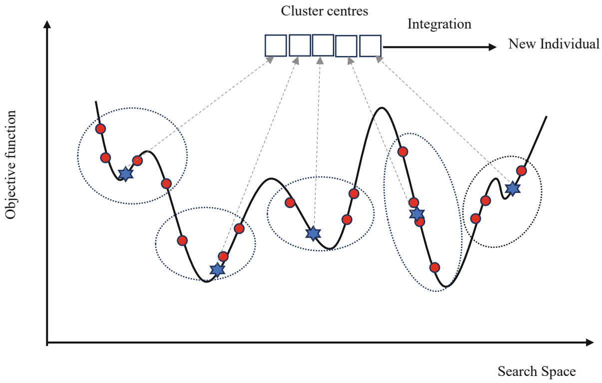

Fig. 1. Clustering at the second level. Circle-shaped points show individuals in the population, while star-shapesd points indicate cluster centres.

3.3 Encoding Strategy

Our approach uses a real-valued encoding scheme to represent individuals. Each solution is described by a vector comprising connection weights and bias values. The encoding length directly correlates with the problem's complexity, reflecting the total number of connection weights and biases that require optimisation.

3.4 Objective Function

We use an objective function for FFNN training that incorporates a regularisation term and is calculated as

$$f = \frac{100}{P}\sum_{p=1}^{P}\xi(x_p) + \frac{\lambda}{2m}\sum ||W||^2, \tag{4}$$

with

$$\xi(\overrightarrow{p}) = \begin{cases} 1 & \text{if } \overrightarrow{o_p} \neq \overrightarrow{d_p} \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

where d_p and o_p are the actual and predicted outputs, respectively, and m is the total number of samples. The regularisation parameter, λ, serves as a hyperparameter, penalising large values of weights and biases. If λ is excessively large, numerous weights will approach zero, simplifying the FFNN and making it prone to underfitting. Conversely, if λ is too small, the regularisation term's influence diminishes. An optimal choice of λ is crucial as it helps control the weights, preventing overfitting while maintaining the model's performance.

3.5 Levenberg-Marquardt Algorithm

We use the weights obtained by C2L-DE as an initial seed to the Levenberg-Marquardt (LM) algorithm [15,18]. LM aims to optimise the objective function by adjusting the network weights using an update rule defined as

$$w_{t+1} = w_t - (J_t^T J_t + \mu I)^{-1} J_t^k E_t, \tag{6}$$

with

$$E_t = \sum_{i=1}^{N} (d_i - y_i)^2, \tag{7}$$

where J is the Jacobian matrix of the error vector E_t, J^T is its transpose, I is the identity matrix with dimensions matching the Hessian $J^p J$, N is the number of training samples, and μ is a damping factor adjusted during the optimisation process. $J^k E$ indicates the gradient of the error function E.

It is worth noting that the LM algorithm converges faster compared to other algorithms, such as BP or back-propagation with momentum [9,14].

3.6 C2L-DE Algorithm

Algorithm 1 presents our proposed C2L-DE algorithm in pseudo-code form. C2L-DE first creates an initial population and evaluates the objective function of each individual. The algorithm then iteratively performs mutation, crossover, and selection operations. Periodically, it undergoes the two levels of clustering. At the first level, based on the clustering period (C_P), a combination of k-means clustering and random selection is employed to update the population, while at the second level, k-means is employed to identify cluster centres and create a new individual as the average of these centres. This new individual then replaces the worst individual in the population. The algorithm iterates until the maximum number of function evaluations is reached. It is worth noting that we utilise a one-step k-means algorithm due to its O(1) complexity, ensuring no change in the overall complexity.

Upon completion, the best individual, $\overrightarrow{x^*}$, is identified. If the maximum number of function evaluations is surpassed, the algorithm proceeds to the secondary phase. It initialises ω as the best individual and resets the iteration count. It then iteratively computes the Jacobian, the approximated Hessian, and the error, updating the weights using the Levenberg-Marquardt algorithm. This process continues until a specified maximum number of iterations is reached.

4 Experimental Results

To evaluate the effectiveness of the proposed C2L-DE algorithm, we conduct a set of experiments on diverse datasets from the UCI machine learning repository[1], namely:

- *Iris*: a well-known classification dataset with 150 samples, 4 features, and 3 classes;

[1] https://archive.ics.uci.edu/ml/index.php.

Algorithm 1: C2L-DE algorithm

Initialisation:;
Initialise N_{pop}, $NFE_{\max}$, $iter_{\max}$, J_r, C_P, λ;
$NFE = 0$, $iter = 1$;
while $NFE \leq NFE_{max}$ **do**
  Generate initial population Pop using uniformly distributed random numbers;
  Calculate objective function of each individual in Pop using Eq. (4);
  $NFE = N_{\text{pop}}$;
  foreach *individual* **do**
    Perform mutation operation;
    Perform crossover operator;
    Calculate objective function using Eq. (4);
    Perform selection operation;
  end
  $NFE = NFE + N_{\text{pop}}$;
  `// First-level Clustering`
  if $rem(iter, C_P) == 0$ **then**
    Randomly generate k as a random number between 2 and $\sqrt{N_P}$;
    Conduct a single step of k-means clustering and designate the cluster centres as set A;
    Randomly pick k individuals from the current population and designate them as set B;
    From the union of sets A and B, select the best k individuals and denote them as $\bar{B}$;
    Choose the new population as $(\text{Pop} - B) \cup \bar{B}$;
  end
  `// Second-level Clustering`
  Randomly generate k as a random number between 2 and $\sqrt{N_P}$;
  Conduct a single step of k-means clustering;
  Select N cluster centre solutions as $\overrightarrow{x_{c1}}$, $\overrightarrow{x_{c2}}$,...,$\overrightarrow{x_{cN}}$;
  $x_{new} = \frac{\overrightarrow{x_{c1}}+\overrightarrow{x_{c2}}+\ldots+\overrightarrow{x_{cN}}}{N}$;
  $x_{worst} \leftarrow x_{new}$;
end
$\overrightarrow{x^*} \leftarrow$ the best individual in *pop*
$iter = iter + 1$;
if $NFE > NFE_{max}$ **then**
  Initialise ω as $\overrightarrow{x^*}$ (i.e. the best individual in the current population);
  Set the current iteration *iter* to 0;
  while $iter < iter_{max}$ **do**
    Compute the Jacobian J, the approximated Hessian $J^T J$, and the error E_t;
    Update weights using Eq. (7);
    Recalculate E_t;
    if $iter < iter_{max}$ **then**
      Increment *iter* by 1;
    end
  end
end

– *Breast Cancer*: comprising 699 samples, 9 features, and 2 classes;
– *Liver*: a binary clinical dataset from BUPA Medical Research Ltd., with 345 instances and 7 features;
– *Pima*: a challenging clinical classification dataset featuring 768 samples, 2 classes, and 8 features;
– *Seed*: an agricultural dataset with seven geometrical features of wheat kernels, containing 210 samples divided into 3 categories.
– *Vertebral*: A clinical dataset incorporating biomechanical features, categorized into 3 classes with 310 samples.

Here we do not focus on determining the optimal FFNN architecture, but adopt the approach from [23,25], setting the number of neurons in the single hidden layer to $2 \times N + 1$, where N is the number of inputs. For evaluation, we employ 10-fold cross-validation.

C2L-DE is benchmarked against a number of state-of-the-art and recently proposed DE-based trainers, including standard DE, QODE, RDE-OP, Reg-IDE, and Cen-CODE. The number of function evaluations for all PCMs is fixed at 25,000 [26]. The population size for all PCMs is set to 50. For C2L-DE, the crossover probability, scaling factor, and jumping rate are set to 0.9, 0.5, and 0.3, respectively, and the clustering period and regularisation parameter are also chosen as 10 and 0.1, respectively. For the remaining algorithms, we use the default parameters as per the cited publications.

The obtained results on the Iris dataset, presented in Table 1, reveal valuable insights into the performance of different DE algorithms. Our proposed C2L-DE algorithm stands out prominently, achieving the joint highest mean fitness value of 99.33 (along with Reg-IDE), showcasing the effectiveness of C2L-DE in converging towards optimal solutions. In addition, the low standard deviation of 2.10 indicates the robustness of C2L-DE across multiple runs. In contrast, standard DE and other comparative algorithms such as QODE and Cen-CODE exhibit lower mean fitness values and higher standard deviations.

Table 1. Experimental results on Iris dataset.

	mean	std.dev	rank
DE	92.00	5.26	6
QODE	95.33	6.32	5
RDE-OP	96.67	6.48	4
Reg-IDE	99.33	2.11	1.5
Cen-CODE	98.00	3.22	3
C2L-DE	99.33	2.10	1.5

The results on the Breast Cancer dataset are given in Table 2. From there, we can see that all algorithms except DE provide a similar mean accuracy. C2L-DE

Table 2. Experimental results on Breast Cancer dataset.

	mean	std.dev	rank
DE	97.36	2.06	6
QODE	98.10	0.99	5
RDE-OP	98.82	1.67	1
Reg-IDE	98.39	2.24	3
Cen-CODE	98.38	1.61	4
C2L-DE	98.53	1.64	2

is second ranked with a mean fitness value of 98.53, demonstrating its competitive performance.

Table 3 shows the results on the Liver dataset. C2L-DE is top ranked with a mean fitness value of 77.64, highlighting its superior performance. QODE and Reg-IDE also exhibit competitive mean fitness values of 76.82 and 76.26, respectively, resulting in the second and third ranks.

Table 3. Experimental results on Liver dataset.

	mean	std.dev	rank
DE	67.81	8.21	6
QODE	76.82	9.46	2
RDE-OP	75.63	6.45	4
Reg-IDE	76.26	4.03	3
Cen-CODE	75.10	6.66	5
C2L-DE	77.64	5.83	1

Table 4 presents the results on Pima dataset. C2L-DE is again top ranked here, with a mean fitness of 81.50. Reg-IDE is second ranked with a mean fitness value of 80.60, followed by RDE-OP (80.21) and QODE (79.55).

The results on the Seed dataset, given in Table 5, also show our C2L-DE algorithms as the top-performing approach, achieving a mean accuracy of 93.80. Cen-CODE follows with a mean accuracy of 82.38, while algorithms like DE, QODE, and RDE-OP perform less effectively.

The experimental results on the Vertebral dataset, reported in Table 6, reveal QODE as the top-performing algorithm. However, QODE generally does not achieve satisfactory results on the other datasets. C2L-DE follows closely with a mean accuracy of 87.74, while DE and Cen-CODE exhibit lower performance.

The obtained results across multiple datasets clearly demonstrate the superior performance of our proposed C2L-DE algorithm compared to the other methods, while also proving it to be a robust method.

Table 4. Experimental results on Pima dataset.

	mean	std.dev	rank
DE	76.94	4.97	6
QODE	79.55	4.94	4
RDE-OP	80.21	5.73	3
Reg-IDE	80.60	4.15	2
Cen-CODE	77.99	4.12	5
C2L-DE	81.50	5.34	1

Table 5. Experimental results on Seed dataset.

	mean	std.dev	rank
DE	70.00	11.01	4
QODE	67.62	3.01	5
RDE-OP	67.62	4.92	5
Reg-IDE	80.60	4.15	3
Cen-CODE	82.38	8.1	2
C2L-DE	93.80	5.96	1

Table 6. Experimental results on Vertebral dataset.

	mean	std.dev	rank
DE	85.16	5.31	5.5
QODE	88.39	8.76	1
RDE-OP	86.77	4.42	3.5
Reg-IDE	86.77	5.37	3.5
Cen-CODE	85.16	6.48	5.5
C2L-DE	87.74	6.23	2

5 Conclusions

In this paper, we have presented the C2L-DE algorithm as a novel effective solution for the complex task of determining optimal weights and biases in feed-forward neural networks. Traditional gradient-based methods, while widely employed, encounter challenges such as sensitivity to the initial values and susceptibility to local optima. Our two-level clustering-based differential evolution approach addresses these issues by introducing a dynamic and informed population update strategy. In the initial phase, clustering identifies diverse regions within the search space, guiding population updates based on localised information. Subsequently, a central point derived from cluster centres, is introduced as a new individual into the population. A comparative analysis against several recent DE training algorithms confirms the promising performance of C2L-DE.

In future work, we intend to extend the application of our algorithm to other ANN-related tasks, such as neural architecture search. Additionally, C2L-DE holds potential for hyperparameter optimisation, showcasing its versatility and adaptability in various aspects of neural network optimisation.

References

1. Abrishami, N., Sepaskhah, A.R., Shahrokhnia, M.H.: Estimating wheat and maize daily evapotranspiration using artificial neural network. Theor. Appl. Climatol. **135**(3), 945–958 (2018). https://doi.org/10.1007/s00704-018-2418-4
2. Amirsadri, S., Mousavirad, S.J., Ebrahimpour-Komleh, H.: A levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training. Neural Comput. Appl. **30**(12), 3707–3720 (2017). https://doi.org/10.1007/s00521-017-2952-5
3. Bojnordi, E., Mousavirad, S.J., Pedram, M., et al.: Improving the generalisation ability of neural networks using a Lévy flight distribution algorithm for classification problems. New Gener. Comput. **41**(2), 225–242 (2023). https://doi.org/10.1007/s00354-023-00214-5
4. Bojnordi,E., Mousavirad,S.J., Schaefer, G., Korovin, I.: MCS-HMS: a multi-cluster selection strategy for the human mental search algorithm. In: IEEE Symposium Series on Computational Intelligence, pp. 1–6, 2021
5. Cai, Z., Gong, W., Ling, C.X., Zhang, H.: A clustering-based differential evolution for global optimization. Appl. Soft Comput. **11**(1), 1363–1379 (2011)
6. Deb, K.: A population-based algorithm-generator for real-parameter optimization. Soft. Comput. **9**(4), 236–253 (2005)
7. Ding, S., Chunyang, S., Junzhao, Yu.: An optimizing BP neural network algorithm based on genetic algorithm. Artif. Intell. Rev. **36**(2), 153–162 (2011)
8. Duan, H., Huang, L.: Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning. Neurocomputing **125**, 166–171 (2014)
9. El-Bakry, M.Y., El-Dahshan, E.-S.A., El-Hamied, E.F.A.: Charged particle pseudorapidity distributions for Pb-Pb and Au-Au collisions using neural network model. Ukrainian J. Phys. **58**(8), 709–709 (2013)
10. Fister, I., Fister, D., Deb, S., Mlakar, U., Brest, J.: Post hoc analysis of sport performance with differential evolution. Neural Comput. Appl. **32**, 1–10 (2018)
11. Hosaka, T.: Bankruptcy prediction using imaged financial ratios and convolutional neural networks. Expert Syst. Appl. **117**, 287–299 (2019)
12. Ilonen, J., Kamarainen, J.-K., Lampinen, J.: Differential evolution training algorithm for feed-forward neural networks. Neural Process. Lett. **17**(1), 93–105 (2003)
13. Kim, D., Kim, H., Chung, D.: A modified genetic algorithm for fast training neural networks. In: International Symposium on Neural Networks, pp. 660–665 (2005)
14. Lera, G., Pinzolas, M.: Neighborhood based Levenberg-Marquardt algorithm for neural network training. IEEE Trans. Neural Networks **13**(5), 1200–1203 (2002)
15. Levenberg, K.: A method for the solution of certain non-linear problems in least squares. Q. Appl. Math. **2**(2), 164–168 (1944)
16. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)

17. Mandal, S., Saha, G., Pal, R.K.: Neural network training using firefly algorithm. Glob. J. Adv. Eng. Sci. **1**(1), 7–11 (2015)
18. Marquardt, D.W.: An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. **11**(2), 431–441 (1963)
19. Mirjalili, S.: How effective is the Grey Wolf optimizer in training multi-layer perceptrons. Appl. Intell. **43**(1), 150–161 (2015). https://doi.org/10.1007/s10489-014-0645-7
20. Moravvej, S.V., Mousavirad, S.J., Oliva, D., Schaefer, G., Sobhaninia, Z.: An improved DE algorithm to optimise the learning process of a BERT-based plagiarism detection model. In: IEEE Congress on Evolutionary Computation, pp. 1–7 (2022)
21. Mousavirad, S.J., Bidgoli, A.A., Rahnamayan, S.: Tackling deceptive optimization problems using opposition-based DE with center-based Latin hypercube initialization. In: 14th International Conference on Computer Science and Education (2019)
22. Mousavirad, S.J., Bidgoli, A.A., Ebrahimpour-Komleh, H., G.S.: A memetic imperialist competitive algorithm with chaotic maps for multi-layer neural network training. Int. J. Bio-Inspired Comput. **14**(4), 227–236 (2019)
23. Mousavirad, S.J., Bidgoli, A.A., Ebrahimpour-Komleh, H., Schaefer, G., Korovin, I.: An effective hybrid approach for optimising the learning process of multi-layer neural networks. In: International Symposium on Neural Networks, pp. 309–317 (2019)
24. Mousavirad, S.J., Gandomi, A.H., Homayoun, H.: A clustering-based differential evolution boosted by a regularisation-based objective function and a local refinement for neural network training. In: IEEE Congress on Evolutionary Computation, pp. 1–8 (2022)
25. Mousavirad, S.J., Jalali, S.M.J., Sajad, A., Abbas, K., Schaefer, G., Nahavandi, S.: Neural network training using a biogeography-based learning strategy. In: International Conference on Neural Information Processing (2020)
26. Mousavirad, S.J., Oliva, D., Hinojosa, S., Schaefer, G.: Differential evolution-based neural network training incorporating a centroid-based strategy and dynamic opposition-based learning. In: IEEE Congress on Evolutionary Computation, pp. 1233–1240 (2021)
27. Mousavirad, S.J., Rahmani, R., Dolatabadi, N.: A transfer learning based artificial neural network in geometrical design of textured surfaces for tribological applications. Surf. Topogr. Metrol. Prop. **11**(2), 025001 (2023)
28. Mousavirad, S.J., Rahnamayan, S.: Evolving feedforward neural networks using a quasi-opposition-based differential evolution for data classification. In: IEEE Symposium Series on Computational Intelligence (2020)
29. Mousavirad, S.J., Rahnamayan, S.: A novel center-based differential evolution algorithm. In: Congress on Evolutionary Computation (2020)
30. Mousavirad, S.J., Schaefer, G., Jalali, S.M.J., Korovin, I.: A benchmark of recent population-based metaheuristic algorithms for multi-layer neural network training. In: Genetic and Evolutionary Computation Conference Companion, pp. 1402–1408 (2020)
31. Mousavirad, S.J., Schaefer, G., Korovin, I.: An effective approach for neural network training based on comprehensive learning. In: International Conference on Pattern Recognition (2020)
32. Mousavirad, S.J., Schaefer, G., Korovin, I., Oliva, D.: RDE-OP: a region-based differential evolution algorithm incorporation opposition-based learning for optimising the learning process of multi-layer neural networks. In: Castillo, P.A., Jiménez

Laredo, J.L. (eds.) EvoApplications 2021. LNCS, vol. 12694, pp. 407–420. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72699-7_26
33. Munkhdalai, L., Lee, J.Y., Ryu, K.H.: A hybrid credit scoring model using neural networks and logistic regression. In: Pan, J.-S., Li, J., Tsai, P.-W., Jain, L.C. (eds.) Advances in Intelligent Information Hiding and Multimedia Signal Processing. SIST, vol. 156, pp. 251–258. Springer, Singapore (2020). https://doi.org/10.1007/978-981-13-9714-1_27
34. Nawi, N.M., khan, A., Rehman, M.Z., Aziz, M.A., Herawan, T., Abawajy, J.H.: An accelerated particle swarm optimization based Levenberg Marquardt back propagation algorithm. In: Loo, C.K., Yap, K.S., Wong, K.W., Teoh, A., Huang, K. (eds.) ICONIP 2014. LNCS, vol. 8835, pp. 245–253. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12640-1_30
35. Pedram, M., Mousavirad, S.J., Schaefer, G.: Training neural networks with Lévy flight distribution algorithm. In: 7th International Conference on Harmony Search, Soft Computing and Applications, pp. 93–103 (2022)
36. Rahmani, S., Mousavirad, S.J., El-Abd, M., Schaefer, G., Oliva, D.: Centroid-based differential evolution with composite trial vector generation strategies for neural network training. In: Correia, J., Smith, S., Qaddoura, R. (eds.) International Conference on the Applications of Evolutionary Computation, vol. 13989, pp. 608–622. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-30229-9_39
37. Sexton, R.S., Gupta, J.N.D.: Comparative evaluation of genetic algorithm and backpropagation for training neural networks. Inform. Sci. **129**(1–4), 45–59 (2000)
38. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: IEEE International Conference on Evolutionary Computation, pp. 69–73 (1998)
39. Si, T., Dutta, R.: Partial opposition-based particle swarm optimizer in artificial neural network training for medical data classification. Int. J. Inform. Technol. Decis. Making **18**(5), 1717–1750 (2019)
40. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)
41. Wang, X., et al.: Massive expansion and differential evolution of small heat shock proteins with wheat (triticum aestivum l.) polyploidization. Sci. Rep. **7**(1), 1–12 (2017)