

Review

Optimizing Artificial Neural Network Learning Using Improved Reinforcement Learning in Artificial Bee Colony Algorithm

Taninnuch Lamjiak*; Booncharoen Sirinaovakul; Siriwan Kornthongnimit; Jumpol Polvichai; Aysha Sohail

Department of Computer Engineering, King Mongkut's University of Technology, Thonburi, Bangkok, Thailand

*Corresponding author

Taninnuch Lamjiak

Department of Computer Engineering, King Mongkut's University of Technology, Thonburi, Bangkok, Thailand

Article information

Received: January 6th, 2024; Revised: April 2nd, 2024; Accepted: April 30th, 2024; Published: May 7th, 2024

Cite this article

Lamjiak T, Sirinaovakul B, Kornthongnimit S, Polvichai J, Sohail A. Optimizing artificial neural network learning using improved reinforcement learning in artificial bee colony algorithm. 2024; 2(1). doi: <https://doi.org/10.70705/ppp.ltcs.2024.v02.i01.pp32-38>

ABSTRACT

Machine learning methods like artificial neural networks (ANNs) have many uses and applications. The loss function in artificial neural networks (ANNs) is usually minimized using heuristic search optimization techniques. Nevertheless, the network's performance may be severely hindered if these strategies cause it to get trapped in local optima. enhanced R-ABC algorithm, which stands for "improved reinforcement learning in the artificial bee colony," is introduced as a new optimization strategy to ANNs in this paper to address this difficulty. To increase the effectiveness of weight modification in ANNs and overcome the constraints of heuristic search, the suggested technique is put forth. By including the properties of nearby food sources, this novel method improves the discovery phase of the conventional R-ABC and increases the search capabilities for finding the best solution. The enhanced R-ABC was tested against several artificial neural networks (ANNs) that use backpropagation with stochastic gradient descent (SGD) and Adam optimizers, as well as other SI approaches including particle swarm optimization (PSO) and conventional R-ABC. Across all benchmark datasets, the findings demonstrated that solutions were consistently improved by both PSO and R-ABC. Unlike SGD and Adam, all SI methods routinely obtained F1-scores higher than 0.94 on the iris dataset. Overall, the SI method performed better than the other optimization techniques for the other datasets. Applying the enhanced R-ABC to ANNs yields better results than heuristic search optimization, particularly as the size of the network increases, according to the findings. While SGD and Adam were able to achieve quicker execution times using TensorFlow, the research indicates that PSO and enhanced R-ABC may enhance model correctness and efficiency. The optimization process and ANNs' capacity to find optimum solutions are both improved by modern SI methodologies. In complicated and high-dimensional datasets, the performance and efficiency of artificial neural network (ANN) training may be greatly enhanced with the use of improved R-ABC and PSO algorithms.

INTRODUCTION

In order to handle and analyze massive amounts of complicated data, machine learning methods are widely used in management and problem-solving. Numerous fields may benefit from these methods, including healthcare, finance, and image processing [1, 3]. A machine learning paradigm that takes its cues from the brain's pattern recognition mechanism is known as an artificial neural network (ANN). This method has been used because of its renowned high-performance characteristics.

providing usefulness in supervised, unsupervised, and predictive learning, as shown in several research [4]. Starting with an input layer and ending with an output layer are the numerous components that make up an ANN. Hidden layers are those that are located between

the input and output layers in an ANN framework; these layers might have an unlimited number of them. A number of further layers could be concealed inside these layers. An ANN with more than one layer is called a "multilayer perceptron" (MLP) [5]. The weights of the perceptrons in each layer dictate the connections between them.

Because it is the basis for the program's learning, weight is very important in an ANN. In artificial neural networks (ANNs), feedforward and backpropagation methods are the main means of learning. The ANN's feedforward method uses the input data to find the answer.

The procedure starts at the input layer and goes all the way to the output layer. The value that is acquired when it is decided gives the outcome. An incorrect model with more errors was produced at first

by using a feedforward method with randomly assigned weights. So, to improve the model's accuracy, the ANN used the backpropagation strategy to adjust the weights. Designed by Rumelhart et al. [6], this algorithm modifies the ANN's weights according to the cost or error detected in the feedforward phase [7].

An important part of the back-end process is the optimizer dissemination process. When adjusting the parameters, backpropagation is used to get the cost function's gradient. One method for optimizing backpropagation is gradient descent (GD). In order to get answers, this method is simple. It is possible, nonetheless, for GD to become stuck in local minima [8]. In addition, updating the weights in each iteration of GD takes a considerable amount of time. The optimization of ANN-based methods has been the subject of several research efforts. Robbins and Monro were the first to suggest stochastic generalized GD (SGD) [9]. Compared to the GD approach, this technique changes weights more often. In contrast to GD, which requires whole datasets to update weights, this method updates them per every row of data. The likelihood of reaching the global optimum is enhanced by this approach. When the goal is convex or pseudoconvex, SGD works well. Despite its local nature, SGD is often used to minimize highly convex functions; nonetheless, it does not ensure the discovery of a global minimum [10]. In addition, in order to update the weights and get the best solution when using the backpropagation method, the procedure must be inverted. Hence, this method requires a significant amount of time to fine-tune the weights via several cycles of computation. Metaheuristic optimization using ANN has been used in several research to address this challenge. One optimization strategy that works well with ANNs is the genetic algorithm (GA) method. A variable-structure neural network with a controller for the network switch and node-to-node relationships was suggested by the authors in [11]. It models various input patterns using features variable structures and uses GA for training. The experimental findings demonstrate that the suggested network outperforms the conventional method on this particular challenge. Each pattern in the XOR problem and handwritten pattern recognition tasks is shown in tables along with its fitness value, network output, and switch state. By including a diffusing operator into both global and local searches, the authors of [12] enhanced the GA used to train ANNs. This approach improved ANN training efficiency, as shown by the standard XOR problem. A hybrid strategy combining the Taguchi method with a GA was suggested by the authors in [13] to optimize both the structure and the variables that make up a feedforward neural network. In order to choose better genes and improve overall performance, the Taguchi approach was implemented into the GA's crossover and mutation procedures. The experimental results show that this hybrid method is effective for tuning the parameters and network structure, especially for global numerical optimization problems and applications as sunspot number forecasting, associative memory tuning, and the exclusive-OR (XOR) problem. A discrete Hopfield neural network (DHNN) is an early kind of artificial neural network (ANN) defined by linked neurons without hidden layers. In combination with GA, the authors of [14] used DHNN as an optimization method. A number of optimization techniques, including the election algorithm and exhaustive search, were used to measure GA's performance. According to these results, the GA did not provide the best possible results. It is clear from the aforementioned research that the XOR issue, a very simple

problem, has been solved using GA in conjunction with an ANN. There was only a single hidden layer in the ANN structure in these cases. Even with an increase to the ANN structure, GA still has trouble breaking out of local optimums and takes a very long time to achieve global optimums [15].

One optimization approach is swarm intelligence (SI), which is really good at discovering the best solutions. Taking cues from the actions of social animals like fish, bees, and foraging ants, SI is a metaheuristic optimization technique. This approach is distinct from hill-climbing search, which has the potential to get stuck in a local optimum very fast. Additionally, unlike GA, SI does not need a complicated set of evolutionary operators. A problem-solving method known as SI increases the likelihood of obtaining a global maximum [16] by having each agent traverse the issue space in pursuit of the optimal solution. A number of research have tackled practical problems by using a SI method. As stated by the writers in [17] presented Antares, which incorporates P2P technology and draws influence from ant behavior.

Antares created a self-structured grid information system to improve the organization and distribution of resource descriptions in dynamic environments. Antares' ability to adapt to the dynamic grid environment is shown by its experimental findings, which show that it is successful at spatially sorting and reproducing descriptors. The author of [18] used SI, which takes its cues from the way social insects behave in groups, to spot unusual data. As they move about in a simulated world, this algorithm forms flocks of mobile agents based on their common traits. Subsequently, we used these flocks to identify the dataset points that stood out from the others. The results demonstrate the practicality and efficacy of the proposed approach in detecting outliers in distributed systems. The author used a flocking approach to detect anomalies in remote data streams in [19]. The flocking algorithm is a paradigm that draws inspiration from nature; it functions similarly to how a school of fish or flock of birds moves. Testing validates the proposed algorithm's viability as a technique for anomaly identification in distributed data streams; it also demonstrates its efficacy for clustering and anomaly detection in centralized and distributed systems. A flocking method was used to identify outliers in dispersed data streams. This model takes its cues from nature and attempts to replicate the coordinated actions of groups of animals, such as fish or birds. The suggested algorithm is an effective method for detecting anomalies in remote data streams, according to the experimental findings. It also helps with anomaly detection in centralized and remote systems and is an effective clustering tool. The aforementioned research clearly shows that SI approaches can handle a wide range of practical issues. In terms of a well-known method for implementing SI techniques on the ANN, PSO is noteworthy. The use of PSO to improve ANN performance has been explored extensively. In [15], the authors covered the topic of CQGAPSO, a cooperative quantum genetic algorithm-particle swarm method used to simultaneously adjust the parameters and topology of a feedforward neural network. By combining quantum GA with PSO, this technique improves prediction accuracy and resilience while reducing the negative effects of duplicated network architectures. With reduced mean best fitness and standard errors, CQGAPSO performed better than GD and GA. Another framework for SI methods, created by Karaboga, is the artificial bee colony (ABC) algorithm [20]. The way honeybees forage served as an

inspiration for ABC. Bees in the ABC structure may be either employed or jobless; the latter group can be further divided into scout bees and bystanders. Bees that are not working provide food for the workers. When faced with difficult optimization issues, researchers turned to a metaheuristic optimization strategy. This approach is helpful because it employs a random procedure that has a higher probability of obtaining the global optimum, and it can be used to any optimization issue. ANN performance has also been enhanced with the use of ABC. An improved method for training ANNs was described in [21] by combining the ABC and Lev-enberg-Marquardt algorithms. The results demonstrated that the hybrid ABC-LM algorithm achieved the most favorable outcomes. To solve a simple XOR issue, the suggested approach was used. In the context of weighted random satisfiability, the authors of [22, 23] used the DHNN with other metaheuristic methods, such as the ABC. In order to improve the solution for global minima and tackle overfitting concerns, this was done before training the DHNN. The purpose of these experiments was to find ways to make ABC work better. The Gbest-guided ABC (GABC) was suggested by the authors of [24] as an alternative ABC that draws inspiration from PSO, another SI approach. A best-so-far ABC (BSF-ABC) was suggested by the authors of [25]. It refreshes the best food sources across all dimensions and restricts the search region for scout bees. The suggested ABC technique shortened the time needed to find the best answer. Adaptive and hybrid ABC (aABC) was named after the authors' innovative ABC method in [26]. Using an arithmetic crossover operation, the aABC algorithm updates the food source with an adaptive amount of perturbation and then develops a new food source from the best food source. A more accurate estimate of transportation energy demand may be obtained by using an improved ABC that incorporates an adaptive search equation, as suggested by the authors in [27, 28].

The findings showed that a more precise solution to the issue might be provided by an improved ABC using an adaptive search equation. As the problem's dimensions grow, the performance of the aforementioned ABC techniques degrades. To address issues with many dimensions, a novel ABC method was introduced in 2018; it is known as reinforcement learning for solution updating in the R-ABC [29]. This method updates all dimensions' values at the same time. An incentive is offered to the worker bee if it identifies a more effective solution. But if the worker bee is unable to find a better answer for a while, that period is punishment. The R-ABC method was shown to be the most effective approach in terms of convergence speed and solution quality for the majority of benchmark functions. In particular, the R-ABC algorithm outperformed its competitors on six of eight 100-dimensional benchmark functions and on all eight 500-, 700-, and 900-dimensional benchmark functions. As the results of the experiments show, the effectiveness of the ABC method in solving high-dimensional optimization problems may be improved by including reinforcement learning. Using a reinforcement learning strategy helps improve problem-solving, according to many research. The authors tackled energy consumption difficulties by using Q-learning, a reinforcement learning technique, as described in [30, 31]. According to the outcomes, the suggested algorithm performs better than competing algorithms. By combining deep reinforcement learning with coevolutionary differential evolution, the authors of [32] were able to improve overall performance while

solving limited optimization problems. With the goal of striking a balance between exploration and exploitation while accommodating for irregular boundaries, the authors of [33] presented a framework to handle multi-objective optimization problems. Improved exploration-exploitation equilibrium is the goal of this approach, which makes use of a multistrategy and multi-crossover differential evolution optimizer. To manage the reference points' constant repositioning, it also makes use of a reinforcement learning-based adaptive reference point activation method. Also, to improve the algorithm's performance on irregular frontier situations, a reference point adaptation approach is proposed. The results show that compared to existing algorithms, the suggested approach performs much better on test issues. The aforementioned research suggests that optimization algorithms may benefit from reinforcement learning's capacity to improve their efficiency and speed of execution. This strategy has been effectively used to address optimization issues with several dimensions, with an emphasis on R-ABC. To solve the traveling salesman issue, the authors of [34] used R-ABC, which outperformed ABC in terms of average and best total route lengths. In addition, R-ABC showed quicker outcomes compared to ABC. Further, an integer linear programming subroutine was used in conjunction with R-ABC in another application to optimize trim loss in the paper sector. The results showed that R-ABC required the least amount of processing time while drastically reducing trim loss [35]. But still, BC exhibits limitations in its capacity to effectively search for an optimal solution. Consequently, this study aims to enhance the discovery equation within the onlooker bee phase by incorporating neighborhood information to update food sources. This method facilitates the improvement of the onlooker bees, enabling them to better approach the optimal solution.

This study aims to address the weight adjustment challenges in a heuristic search using the backpropagation approach of an ANN. To overcome these challenges, this study applied SI techniques, which are known for their effectiveness in achieving optimal solutions. Previous research studies demonstrated the high performance of PSO when implemented in conjunction with an ANN. Although PSO was initially implemented with a straightforward ANN structure featuring a few dimensions, real-world problems often involve more extensive ANN structures, leading to an increase in the number of dimensions. To address this issue, this study introduces R-ABC, an SI approach designed to handle high-dimensional optimization problems. The R-ABC was used instead of the traditional backpropagation process. Furthermore, an R-ABC algorithm customized for ANNs was developed to optimize many parameters that need to be adjusted. This study improves the onlooker bee phase by adjusting positions across all dimensions instead of just one randomly selected dimension, mirroring the backpropagation method in a heuristic search. In addition, the position-update formula incorporates the positions of neighboring food sources. These improvements enhance the exploitative behavior of R-ABC. This study also integrated PSO to assess its performance alongside R-ABC. The key contributions of this study are summarized as follows:

- (i) Substituting the backpropagation methodology in the ANN with the application of efficient SI techniques, particularly PSO and R-ABC, improves the effectiveness of weight adjustment
- (ii) Recognizing the real-world complexity involving extensive ANN structures with increased dimensions

(iii) Introducing and applying R-ABC, an SI approach tailored for high-dimensional optimization problems, as an alternative to traditional backpropagation

(iv) Developing an algorithm for R-ABC to adjust multiple ANN parameters, enhancing its exploit-

Methodology

2.1. Artificial Neural Network (ANN). ANNs are a class of machine learning strategies that simulate an organism's pattern recognition system in the brain. An ANN is composed of many basic processors, called neurons, which are linked together by weights. Each layer of the ANN stores the neurons. An ANN consists of two or more layers, with additional layers causing the structure and weight of the network topologies to rapidly increase. The structure of an ANN is shown in Figure 1. Figure 1(a) shows three layers: input, output, and hidden layer 1. The number of weights in Figure 1(a) is 30. The weight increases to 50 when hidden layer 2 is added, as shown in Figure 1(b). From Figure 1, adding only one hidden layer rapidly increases the number of weights because the nodes of each layer are connected to each other. As a result, it takes longer to calculate.

ANN learning involves three primary steps: firstly, computing the outputs through a feedforward process across the network; secondly, evaluating the error; and finally, adjusting the weight and bias parameters based on the heuristic search directions derived from the local gradient of the network. The steps of the ANN algorithm can be outlined in Algorithm 1.

2.2. Particle Swarm Optimization (PSO) Algorithm. PSO is one of the SI approaches known for its high performance and short problem-solving time. Many areas of work have applied PSO to solve various problems [37]. PSO was implemented based on the concept introduced in [38].

Experimental Setting

Data Benchmarks. The iris, balancing scale, heart, wine, liver problems, and seeds datasets were used as multivariate benchmarks. It was usual practice for the UC Irvine Machine Learning Repository to provide these benchmark datasets. These datasets have been used to evaluate the efficacy of several research' methodologies. Ronald Fisher, a British statistician, eugenicist, and biologist, brought the iris dataset to light [39]. There are three different kinds of iris blooms in this collection, and the characteristics that distinguished them were the following: sepal length, petal width, sepal width (all in centimeters), and petal length. The second piece of data is the equilibrium scale. The purpose of creating this dataset was to simulate the psychological trials detailed by R. Siegler [40]. It takes into account four attributes—left weight, left distance, right weight, and right distance—to determine whether the balance scale was sloping to the right, left, or balanced. It comprises three classes for this purpose. There are five tiers to each attribute that specifies a class. This study also makes use of the heart dataset as a benchmark. There is data on cardiac disease diagnoses, split down the middle: 0 for individuals with a risk of heart disease of less than 50% and 1 for patients with a risk of heart disease of 50% or more. While there are 76 characteristics in the dataset, the majority of users only make use of 13 of them [41]. The wine dataset is the fourth one. Chemical

examination of three distinct grape varieties produced in the same Italian area yielded this dataset. The amount of components in each of the three wine varieties is reflected in the thirteen attributes [42]. There are six characteristics in the liver disorder dataset. The first five variables are blood tests that may be sensitive indications of liver disorders associated with heavy alcohol use. The drink characteristic is the last one, and the dataset contains records for individual males [43]. Three separate wheat varieties—Kama, Rosa, and Canadian—had their interior kernel structures described by seven attributes in the seed dataset, the last dataset used in this research. We used a soft X-ray method with 70 randomly chosen pieces for each variety to identify the underlying kernel structure [44].

into two algorithms: PSO and R-ABC. The parameters defined in the PSO are the upper and lower bounds of the velocity and position of each particle, swarm size, dimension of each particle, C1 and C2, which are parameters for controlling the direction of each particle and swarm, respectively. The R-ABC variables that must be defined include the upper and lower bounds of the food source, the colony size, the dimensions of the food source, and the limitation that the employed bee cannot find a better solution. Tables 3 and 4 show the details of PSO and R-ABC variables defined for each benchmark dataset.

From Tables 3 and 4, most parameters for each dataset exhibited similarities between the PSO and R-ABC algorithms. The upper and lower bound values for PSO and R-ABC were determined based on the minimum and maximum values of the final adjusted weights obtained from the general backpropagation in the ANN, utilizing SGD and Adam as optimizers. The dimensions of the particles in the PSO and the dimensions of the food source in the R-ABC were similar for each dataset. These dimensions were established based on the structure of the ANN algorithm, considering the distinct number of input and output nodes in each dataset. While all dimensions of each particle in the PSO algorithm are updated in each iteration, only one dimension of the food source in the R-ABC algorithm is updated in the employed bee phase. The colony size in R-ABC significantly influenced its performance, leading to different definitions of the colony size for each dataset. To determine the optimal colony size for R-ABC, we initially set the colony size for each dataset (except for the balance scale with three hidden layers) to 100. Subsequently, we assessed performance using a 1-fold and 1-time configuration for each dataset. If the performance of both R-ABC and improved R-ABC did not surpass that of backpropagation in the ANN method, we incremented the colony size by 100 for each subsequent test. For the balance scale with three hidden layers, we initiated the colony size at 1,000 and increased it by 100 in each iteration, similar to smaller ANN structures. The colony size of the iris dataset was smaller than those of the other datasets, reflecting its lower complexity. This choice minimizes the need for an extensive colony size, consequently reducing running time. In contrast, the balance scale with 3 hidden layers, being highly dimensional, starts with a colony size of 1,400 to enhance the likelihood of discovering food sources in the R-ABC algorithm.

Results and Discussion

Compared to backpropagation and heuristic search optimization approaches, all SI algorithms performed better. Specifically, the average F1-score was best produced by the improved R-ABC, whereas the

minimum F1-score was greatest by the conventional R-ABC. All algorithms showed comparable maximum F1-scores with the exception of SGD. Regardless of the statistical criteria, SGD always came out on the bottom. With its few dimensions and clear species-level differences, the iris dataset is ideal for most algorithms to get the best possible results. But in a few trials, SGD and Adam's solutions became stuck, which may explain why their minimum F1-scores were low and their standard deviations were high. Using the balanced scale dataset, PSO fared better than Adam on most statistical metrics, with the exception of the highest F1-score. When looking at average, minimum, and standard deviation values, the improved R-ABCs came in second place, while the conventional R-ABCs took third. But across the board, SGD had the worst outcomes in terms of statistical significance. In keeping with the trends seen in the iris dataset, SGD displayed high scores for both methods, but Adam's minimum F1-score results were shockingly low. This suggests that a few trials were stuck in local minima. Following trends seen in the balancing scale dataset, R-ABC and enhanced R-ABC were the best performers in the heart dataset across most statistical metrics. However, Adam fared better than R-ABC on the maximum F1-score. With respect to average, minimum, and maximum F1-scores, the enhanced R-ABC performed best in the seed and wine datasets. On the wine dataset, Adam and PSO showed the best standard deviations, whereas on the seed dataset, PSO showed the best SD. Contrarily, PSO had the worst performance in the seed dataset, while SGD had the worse performance across all statistical values in the wine dataset. In contrast, SGD outperformed PSO on most statistical measures of the F1-score in the liver disease dataset, with the exception of the highest F1-score, where PSO excelled. Among the SI methods, improved R-ABC came in third place in terms of F1-score values, while PSO came in second. When it comes to the runtime of artificial neural networks (ANNs) using the SGD and Adam optimizers from the TensorFlow library, it is not much affected by the amount of instances and dimensions in each benchmark dataset. On the other side, as the dataset's dimensions and number of instances grew, the ANN's runtime for the PSO and R-ABC optimizers grew. Since PSO updates all weight dimensions on each iteration, as opposed to R-ABC which changes a single weight dimension during the observing bee phase, it takes somewhat longer than R-ABC and enhanced R-ABC. So, even with the identical swarm and colony sizes, PSO took longer to execute than R-ABC. In the seed dataset (consisting of around 200 instances) and the wine dataset (containing about 200 instances), PSO and R-ABC optimizers used less time than SGD and Adam, as shown in Table 5. For identical swarm and particle sizes in the iris dataset, R-ABC required less time than PSO. With respect to

PSO took a little slower, with 345 cases in the liver disease dataset, but all optimization methods had comparable runtimes on the heart dataset, which had 303 occurrences. Given that the balanced scale dataset had 625 instances, the R-ABC method took around twice as long as the SGD optimizer from the TensorFlow package, while the PSO method took almost four times as long. Counting instances affects the R-ABC approach's internal assessment process, which is called the function evaluation. One method that greatly helps in handling complicated and big datasets is TensorFlow, a multidimensional array technology that is used in many different applications to

speed up algorithms [51].

In the iris and wine datasets, the enhanced R-ABC ranks first for the Friedman ranking ($P < 0.05$). However, in the heart and balance scale datasets, the efficiency of each technique is not substantially higher than the others. On the other hand, PSO ranks first in the Friedman test and produces the best results in the majority of statistical tests. When compared to the previous approaches, the modified R-ABC method performs much better in solving simple problems, according to the Friedman ranking findings. Due to the complicated nature of the other datasets, the suggested strategy falls short of competing solutions when used there.

Each benchmark dataset's F1-score development is shown in Figures 5, 6, 7, 8, 9, and 10. Backpropagation using SGD and Adam optimizer approaches gets good F1-scores at the beginning of the learning process, according to the graphs. But, some runs become stuck at local optima, which stops them from becoming better. However, there was a noticeable uptick in the F1-scores obtained by SI methods. The F1-scores skyrocket in the first few rounds of training on the seed, iris, balance scale, liver disease, and liver disorder datasets, and then gradually rise thereafter. All classification algorithms show a steady improvement in F1-scores on the heart dataset, but the enhanced R-ABC performs the best. In the beginning, the F1-scores for the wine dataset went down, but eventually, they started to go back up for all of the methods. In particular, the F1-scores showed somewhat better performance after backpropagation with the Adam optimizer, PSO, and conventional R-ABC on the wine dataset, but backpropagation with the SGD optimizer and enhanced R-ABC kept improving their results. Figure 8 demonstrates that compared to backpropagation using the SGD optimizer, the improved R-ABC yields superior results.

As shown in Figures 5, 6, 7, 8, 9, and 10, SGD and Adam speed up the process of finding beneficial solutions by making use of their heuristic optimization search skills, which efficiently converge to optimum answers. The problem, however, is that these techniques may easily get mired in local optima. Their ability to quickly converge to optimum solutions is due to their heuristic optimization search skills, which is responsible for their fast advancement. Nevertheless, these strategies may get trapped in local optima, which hinders their capacity to seek and discover the global optimum.

Conclusions

This study employed advanced SI optimization techniques to optimize weight adjustments in ANNs. The R-ABC algorithm was enhanced by improving the onlooker bee phase to adjust across all dimensions rather than a single randomly chosen dimension, closely mirroring the heuristic search method in backpropagation. In addition, the position-update formula integrates neighboring food sources to improve the exploitation capabilities of the R-ABC. The proposed method was evaluated using the F1-score and compared its performance with established optimization approaches such as SGD and Adam, as well as with other SI techniques such as PSO and traditional R-ABC.

This work introduces key advancements in the use of SI methods for ANN optimization, including refining the onlooker bee phase in R-ABC for comprehensive weight adjustment and incorporating neighboring positions for better exploitative behavior. These contri-

butions demonstrate the capability of SI approaches to consistently surpass traditional optimization methods such as SGD and Adam, particularly in avoiding local optima. The findings also establish the potential of SI optimization to improve ANN performance across various datasets and problem domains. The outcomes indicate that PSO and the improved R-ABC consistently outperform other methods in most benchmark solutions, continuing to improve without getting trapped in local optima. In the iris dataset, all SI approaches in this experiment attained F1-scores exceeding 0.94 in all statistical outcomes, surpassing the performances of both SGD and Adam. Furthermore, in various other datasets, the SI approach consistently outperformed SGD and Adam in most statistical evaluations, except for the liver disorder dataset, in which SGD exhibited the best performance. However, PSO and improved R-ABC also delivered performances closely comparable to SGD and retained the potential to further enhance performance with an appropriate variable setup in the PSO and R-ABC approaches. Nevertheless, it is worth noting that the PSO and R-ABC approaches require more runtime when compared to SGD and Adam from the TensorFlow library because TensorFlow is a Python library designed to optimize computational time using multidimensional arrays. Interestingly, when we developed and compared the ANN structure with an SGD optimization approach that does not rely on the TensorFlow library, the results demonstrated that PSO not only provides superior performance but is also faster than the self-developed SGD approach. These discoveries highlight the capacity of SI techniques to assist ANNs in achieving optimal solutions.

The practical benefits of this research are significant for ANN applications. The enhanced R-ABC and PSO approaches offer superior performance in optimizing neural network weights across a variety of datasets, particularly for benchmark problems, and surpass traditional methods such as SGD and Adam in statistical evaluations. Although PSO and R-ABC require more runtime compared to SGD and Adam when utilizing TensorFlow, their superior outcomes justify their use in scenarios where performance is a priority. Moreover, the integration of TensorFlow concepts into SI approaches can further enhance efficiency and open new avenues for addressing complex problems. Future research can extend the capabilities of these approaches in solving challenging problems in ANN training by exploring other methodologies, such as an ACO, and optimizing SI parameters.

REFERENCES

1 “Using deep learning models for COVID-19 related sentiment analysis on Twitter data” (W. H. Bangyal, S. Amina, R. Shakir, G. Ubakanma, and M. Iqbal, 2023 International Conference on Human-Centered Cognitive Systems (HCCS), pp. 1-6, Cardiff, UK, December 2023.

Two authors, namely W. H. Bangyal, R. Shakir, A. Ashraf, and Z. U. Qayyum, and Abstract: “Automatic detection of diabetic retinopathy from fundus images using machine learning based approaches” was presented at the 2023 25th International Multitopic Conference (INMIC) in Lahore, Pakistan in November 2023. Page numbers range from 1 to 6.

“Analysis of brain imaging data for the detection of early age autism spectrum disorder using transfer learning approaches for internet of things,” published in the IEEE Transactions on Consumer Electronics in 2024, was written by A. Ashraf, Z. Qingjie, W. H. K. Bangyal, and M. Iqbal.

The article “Introduction to artificial neural network” was published in the International Journal of Engineering and Innovative Technology in 2012 and was written by A. Dongare, R. R. Kharde, and A. D. Kachare.

“A beginner’s guide to neural networks and deep learning pathmind,” [5] Pathmind A neural network wiki page: <https://wiki.pathmind.com/neural-network>.

[6] “Learning Representations by back-propagation. errors,” Nature 533, 1986, by D. E. Rumelhart, G. E. Hinton, and R. J. Williams.

A. Ge’ron, “Neural networks and deep learning,” Oreilly Publishing, 2018, p. 354.

In order to circumvent the local minima issue while learning with artificial neural networks, global descent was proposed by B. C. Cetin, J. W. Burdick, and J. Barhen in their 1993 paper published in the IEEE International Conference on Neural Networks, volume 2, pages 836-842.

The article “A stochastic approximation method” was published in 1951 in the Annals of Mathematical Statistics, volume 22, issue 3, pages 400–407, and was written by H. Robbins and S. Monro.

[10] In “Optimization methods for supervised machine learning: from linear models to deep learning,” F. E. Curtis and K. Scheinberg (2017) discuss several approaches in The Operations Research Revolution, volume 89, pages 89-113.

In November 2003, at the 29th Annual Conference of the IEEE Industrial Electronics Society in Roanoke, VA, USA, S. H. Ling, H. K. Lam, F. H. F. Leung, and Y. S. Lee presented their work on a genetic algorithm based neural-tuned neural network.

The paper “An improved genetic algorithm and its application” was presented at the 2005 5th International Conference on Information Communications and Signal Processing in Bangkok, Thailand, in December 2005. The authors of the paper are Q. Gao, K. Qi, Y. Lei, and Z. He. The paper is on pages 357–360.

[13] “Tuning the structure and parameters of a neural network by using hybrid taguchi- genetic algorithm,” published in 2006 by IEEE Transactions on Neural Networks, volume 17, issue 1, pages 69–80. J. T. Tsai, J. H. Chou, and T. K. Liu were the authors.

The paper “Random maximum 2 satisfiability logic in discrete

hopfield neural network incorporating improved election algorithm” was published in Mathematics in 2022 and was co-authored by V. Someetheram, M. F. Marsani, and M. S. M. Kasihmuddin.

“Tuning the structure and parameters of a neural network by using co-operative quantum particle swarm algorithm,” published in Applied Mechanics and Materials in 2011, was written by Q. Tang, L. Zhao, R. Qi, H. Cheng, and F. Qian.

Applied Mathematics and Computation published a paper in 2007 titled “A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training” [16]. The authors are J. R. Zhang, J. Zhang, T. M. Lok, and M. R. Lyu.

in “Antares: an ant-inspired P2P information system for a self-structured grid,” (pp. 151-158), published in 2007 in Budapest, Hungary, in the Proceedings of the 2nd Bio-Inspired Models of Network, Information and Computing Systems, edited by A. Forestiero, C. Mastroianni, and G. Spezzano.

[18] In A. Forestiero’s 2017 article “Bio-inspired algorithm for outliers detection” published in Multimedia Tools and Applications, volume 76, issue 24, pages 25659–25677, the author discusses [this topic].

The article “Self-organizing anomaly detection in data streams” was published in 2016 by Information Sciences and was written by A. Forestiero.

D. Karaboga, “An idea based on honey bee swarm for numerical optimization,” 2005, https://abc.erciyes.edu.tr/pub/tr06_2005.pdf, cited in ().

[21] “Hybrid artificial bee colony algorithm for transit network design” (WSZETO & Y. Jiang, 2012) in the TR: Journal of the TRB (vol. 2284, no. 1), pp. 47–56.

“Multi-discrete genetic algorithm in hopfield neural network with weighted random k satisfiability,” published in Neural Computing and Applications, volume 34, issue 21, pages 19283–19311,

2022, was written by N. E. Zamri, S. A. Azhar, S. S. M. Sidik, and others. Researchers S. S. M. Sidik, N. E. Zamri, and M. S. M. Kasihmuddin

The authors of the 2022 article “Non-systematic weighted satisfiability in discrete hopfield neural network using binary artificial bee colony optimization” are H. A. Wahab, Y. Guo, and M. A. Mansor. The article appears in the journal Mathematics, volume 10, issue 1129, papers 1–27.

The paper “Gbest-guided artificial bee colony algorithm for numerical function optimization” was published in 2010 by Applied Mathematical and Computation and was written by G. Zhu and S. Kwong.

[25] “The best-so-far selection in Artificial Bee Colony algorithm,” Applied Soft Computing, vol. 11, no. 2, pp. 2888-2901, 2011, by A. Banharnsakun, T. Achalakul, and B. Sirinaovakul.

Applied Soft Computing, 2016, vol. 49, pp. 423-436, article by D. Karaboga and E. Kaya titled “An adaptive and hybrid artificial bee colony algorithm (aABC) for ANFIS training” [26].

D. Ozdemir, S. Donderler, and D. Aydın, “A new improved neural computing and applications, 2022, vol. 34, no. 20, pp. 17455-17471, discusses an artificial bee colony algorithm that is used to predict energy consumption.

[28] “An adaptive search equation-” by D. Ozdemir and S. Donderler in the Turkish Journal of Electrical Engineering and Computer Sciences, volume 30, issue 4, pages 1251–1268, 2022, based on an artificial bee colony method for transportation energy demand forecasting.

Reinforcement learning for solution updating in artificial bee colony, S. Faiee, S. Prom-On, and B. Sirinaovakul, 2018, PLoS One, vol. 13, no. 7, pp. e0200738-38.

In their paper titled “A hyperheuristic with Q-learning for the multiobjective energy-efficient distributed,” Zhao, Di, and Wang (2019) publish their find.