Mini Review

How do artificial neural networks lead to developing an optimization method?

*Corresponding author: Ali Sadollah, Department of Mechanical Engineering, University of Science and Culture, Tehran, Iran, Tel: +989123313591; E-mail: sadollah@usc.ac.ir; ali_sadollah@yahoo.com

Check for updates

Ali Sadollah*

Department of Mechanical Engineering, University of Science and Culture, Tehran, Iran

Abstract

This concise paper explains the inspiration of AI particularly artificial neural networks (ANNs) for developing new metaheuristics. Using the unique concept of ANNs and its wide applications in different fields of study, how the ANNs can be utilized for solving real life and complex optimization problems? This paper briefly links the inspiration to a practical model in order to build an optimization strategy.

Objectives

Simply looking at artificial neural networks (ANNs), we can see there exists many potentials from this mathematical model. Many applications and improvement have been inspired using the ANNs. Question of how ANNs inspire a new optimizer is underlined in its unique structure. Why the most important task of ANNs is prediction or regression. Why not using the concept of ANNs for the sake of optimization? Indeed, there are some versions of ANNs such as Hopfield neural networks in which are used for optimization, however, they mostly act as local search approaches and they are not considered as global optimizers. In this brief review paper, it is going to answer this question.

The main contribution in this brief review paper is to shortly introduce the proposed metaheuristic named as neural network algorithm (NNA) along with its applications up to now. For sure, in near future, more applications and utilization of this optimizer will be seen. It is worth mention that the NNA is different with ANNs in terms of concept and application. Indeed, the idea and inspiration of NNA is obtained using the unique strategy and structure used in the ANNs. Hence, as for other metaheuristics, the NNA is a metaheuristic optimization algorithm utilized for solving optimization problems unlike ANNs in which applies for regression and prediction purposes.

Artificial Neural Networks (ANNs)

Artificial neural networks (ANNs) are computing structures inspired by biological neural network structure and/or functional aspects. The ANNs are comprised of dense, interlinked, biologically motivated nervous system computing units (i.e., artificial neurons) [1]. The network feature is primarily influenced by interactions between units, as can be seen in nature.

ANNs can be divided into the following two groups according to their communication pattern (architecture), *a*) neural feed networks: these are networks whose architecture contains no loops. *b*) recurrent networks: these are networks in which loops arise due to feedback interactions and where a parameter of time falls into the model, the neural networks are complex in this way [2].

Figure 1 demonstrates two standard feed-forward and recurrent neural architectures for the ANNs. The question is how to connect the ANNs to form an efficient metaheuristic optimizer? First, let's redefine the meaning of a metaheuristic. More details of ANNs processes are given in Section 4 along with the proposed NNA.

Metaheuristic optimization algorithm

In computer science, a metaheuristic technique is a higher level or heuristic approach that attempts to discover, produce or pick a heuristic that can provide a satisfactory solution to a problem of optimization, in particular with insufficient knowledge or reduced computer power. Metaheuristics sample a sub-set of solutions that are too large to be listed or discussed.
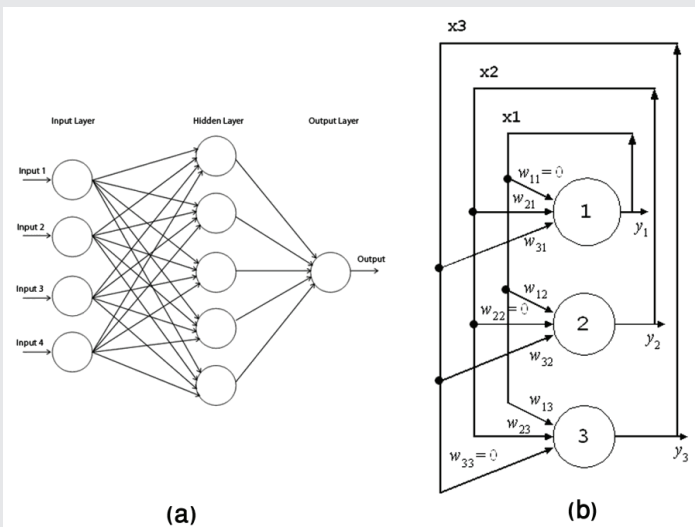
067

Figure 1: Schematic vision of an ANNs with: (*a*) feed forward neural networks, (*b*) recurrent neural networks.

Metaheuristics can make relatively few predictions about the solution of the optimization problem and can hence be used for a range of problems.

The design of nature–inspired metaheuristics is a very active field of study. Many recent metaheuristics are motivated by natural processes, in particular by evolutionary computational algorithms. Nature serves as a source of ideas, processes and principles in the design of computer systems that solve complex computer problems. Such metaheuristic methods include simulation of annealing [3], genetic algorithms [4] (e. g., classified as evolutionary algorithms), ant colony optimization [5], the optimization of particle swarm [6] and water cycle algorithm [7].

Neural Network Algorithm (NNA)

A modern metaheuristic optimization algorithm has been developed with the aid of ANNs principle and the biological nervous system to optimally solve problems, known as the neural network algorithm (NNA) [8]. The proposed NNA incorporates the framework and definition of ANNs to build new solution candidates and utilized the other ANNs operators for its searching strategy. The NNA benefits from the dynamic nature of ANNs and their mechanism in order to generate new solutions in the search domain.

The NNA begins with an initial population called the pattern solution population, as for other metaheuristics. Inspired by the ANNs, at each iteration in the NNA, target data is considered as the best solution obtained (i.e., time–optimal solution) and the goal is to reduce the error between target data and other predicted pattern solutions (i.e., moving other predicted patterns to the target solution). In fact, the NNA is designed to minimize the problem (i.e., reduce the error between the target and the predicted solutions). It should be stressed that this aim solution has been revised with each iteration.

There may be multiple input directions for the artificial neurons or processing units in the ANNs, referring to the

dendrites. The unit averages the weighted values of these input paths using a straightforward summation. The effect is an internal degree of operation for the unit [9]. Initial weights in ANNs are random numbers which will be modified considering the measured network error as the iteration number increases. Back to the NNA, initial weights are defined as a weight matrix (initial random numbers between zero and one) which is altered when iteration number is increased. Then, by matrix multiplying the weight matrix to initial population, new solutions can be obtained.

Next, these solutions are added to the current population to form the new positions of new candidate solution. Note that there is a limitation on weight values. Indeed, the restriction imposed is that the summation of weights does not exceed one which is due to escaping from local optima during search [8]. Figure 2 displays the schematic view of the aforementioned rule in practical.

After generating new solutions, search operators such as bias and transfer function operators play main important roles after creating a new update solution with the special ANNs structure. In the complexities of the neural networks model, the bias present plays a crucial role. In the current population of pattern solutions and a modernized weights matrix the bias operator (acting as noise) modifies a certain portion of pattern solutions in the NNA.

In the NNA, the transfer function operator, unlike the ANNs, is moving from existing search positions to the new positions of the population to the target solution so that higher performing solutions are revised and implemented against the target solution. The schematic view of the NNA entire processes can be observed in Figure 3 showing the local and global feedbacks along with used searching operators.

It can be seen that a simple structure of ANNs (i.e., neuron and layers) can lead to design a metaheuristic. The findings of numerical optimization obtained from many optimization problems clearly show the attractiveness and competition
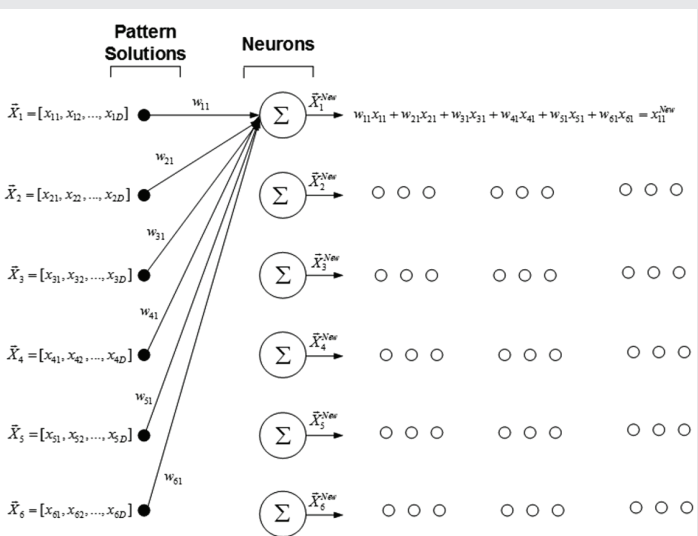

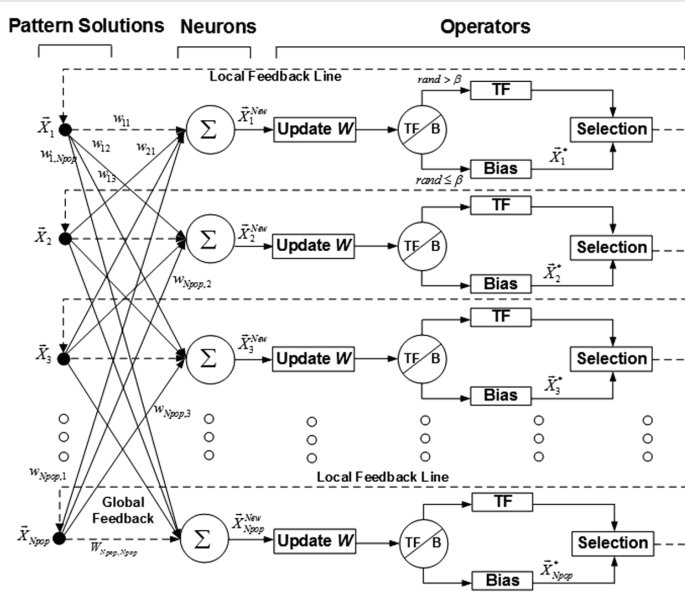**Figure 2:** Schematic vision of new pattern solution generation [8].

Figure 3: Schematic view for the performance of the NNA [8].

characteristics of the proposed NNA for dealing with unconstrained and restricted real life issues with several design variables relative to recent and well-used optimizers [10-14].

It is worth mentioning that developing a new metaheuristic with a new metaphor by itself does not consider a novel contribution in optimization community. The proposed new optimizer should have examined first over well-known and valid mathematical benchmarks, then should be applied for solving real-world, practical, and large-scale optimization problems. Only in this situation, developing new metaheuristic based on new metaphor inspired by nature can be accepted as it increases the accuracy of obtained optimum solution and it may reduce computational time as it maturely and quickly converges to global optimum solution.

References

1. Hassoun MH (1995) Fundamentals of artificial neural networks. The MIT Press, Campridge. **Link:** https://bit.ly/35fyaYp

2. Smith KA (1999) Neural networks for combinatorial optimization: A review on more than a decade of research. Informs J Comput 11: 15-34. **Link:** https://bit.ly/3ka3UEk

3. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220: 671-680. **Link:** https://bit.ly/359BXGx

4. Goldberg D (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, MA. **Link:** https://bit.ly/3keNlXV

5. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics Part B 26: 29-41. **Link:** https://bit.ly/3lWZiSw

6. Kennedy J, Eberhart R (1995) Particle swarm optimization. IEEE IJCNN, Perth, Australia 4: 1942-1948. **Link:** https://bit.ly/3kciZoT

7. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm – A novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110−111: 151-166. **Link:** https://bit.ly/354xBAg

8. Sadollah A, Sayyaadi H, Yadav A (2018) A dynamic metaheuristic optimization model inspired by biological nervous systems: Neural network algorithm. Appl Soft Comput 71: 747-782. **Link:** https://bit.ly/35e35UD

9. Topping BHV, Khan AI, Bahreininejad A (1997) Parallel training of neural networks for finite element mesh decomposition. Comput Struct 63: 693-707. **Link:** https://bit.ly/2TanDI0

10. Yiying Z, Zhigang J, Ye C (2020) Hybrid teaching−learning-based optimization and neural network algorithm for engineering design optimization problems. Knowledge-Based Systems 187: 104836. **Link:** https://bit.ly/3o6ONhx

11. Yiying Z, Zhigang J, Ye C (2020) Hybridizing grey wolf optimization with neural network algorithm for global numerical optimization problems. Neural Computing and Applications 32: 10451−10470. **Link:** https://bit.ly/3jcSVsm

12. AbouOmar MS, Zhang HJ, Su XY (2019) Fractional order fuzzy PID control of automotive PEM fuel cell air feed system using neural network optimization algorithm. Energies 12: 1435. **Link:** https://bit.ly/3o646qw

13. Chohan JS, Mittal N, Kumar R, Singh S, Sharma S, et al. (2020) Mechanical strength enhancement of 3d printed acrylonitrile butadiene styrene polymer components using neural network optimization algorithm. Polymers 12: 2250. **Link:** https://bit.ly/3ja7pcA

14. Fawzi M, El-Fergany AA, Hasanien HM (2019) Effective methodology based on neural network optimizer for extracting model parameters of PEM fuel cells, Int J Energy Res 1−12. **Link:** https://bit.ly/37iIykA