Engineering Applications of Artificial Intelligence 67 (2018) 246–259

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Optimization of neural network using kidney-inspired algorithm with control of filtration rate and chaotic map for real-world rainfall forecasting

Najmeh Sadat Jaddi, Salwani Abdullah *

Data Mining and Optimization Research Group (DMO), Center for Artificial Intelligence Technology, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

A R T I C L E   I N F O

Keywords:
Kidney-inspired algorithm
Artificial neural network
Filtration rate control
Chaotic map
Classification
Time series prediction
Real-world rainfall forecasting

A B S T R A C T

A broad variety of real-world problems have been solved using multilayer perceptron (MLP) artificial neural networks (ANNs). Optimization techniques aid ANNs to select suitable weights and achieve correct results. Recently, the kidney-inspired algorithm (KA) has been proposed for optimization problems. This algorithm is based on the filtration, reabsorption, secretion, and excretion processes that take place in the kidneys of the human body. In the KA, the value of α in the filtration rate formula is a constant value in the range of [0, 1] that is set in the initialization stage of the algorithm. In this paper, an improved KA for optimization of the ANN model is presented in which the filtration rate is controlled by changing the value of α from minimum to maximum during the search process, which helps in achieving a better balance between exploration and exploitation in the algorithm. In this algorithm if more solutes are filtered and moved to filtered blood it means that the algorithm has more exploration. In contrast, if more solutes move to waste it means that more exploitation is performed by the algorithm. In addition, the separate use of three chaotic maps instead of a random number in the movement formula of the modified KA is investigated in order to assess the ability of each map to help to achieve superior results. The proposed method is tested on benchmark classification and time series prediction problems. The method is also applied to a real-world rainfall forecasting problem. The results of a statistical analysis prove the ability of the method.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

In recent years, the use of computational intelligence has become commonplace in handling complex real-world problems. This intelligence is based on concepts such as decision-making, fuzzy logic, metaheuristics, and artificial neural networks (ANNs). Artificial neural networks are the most powerful learning models currently available for estimating unknown functions. The best known and most widely used topology is the multilayer perceptron (MLP). Various researchers have proposed optimization algorithms to train MLPs (Yang et al., 2005; Nasseri et al., 2008; Tripathy et al., 2010). However, classical techniques frequently face difficulties in solving optimization problems in the real world; they tend to require a large amount of computational time, large amount of memory, become trapped in local optima and produce poor-quality solutions. In order to overcome these difficulties, metaheuristic algorithms are increasingly been used to train ANNs (see for example, Islam et al., 2009; Oh et al., 2009; Curry and Morgan, 2010; Kaylani et al., 2010; Mantzaris et al., 2011; Zanchettin et al., 2011; Yang and Chen, 2012; Jaddi et al., 2013, 2015a,b; Jaddi and Abdullah, 2017; Jaddi et al., 2017a). A review of the design of feed forward neural networks is given in Ojha et al. (2017).

Metaheuristic algorithms have been widely employed to train MLPs. The algorithm most commonly used for MLP optimization in the literature is the genetic algorithm (GA) (Oh et al., 2009; Jassadapakorn and Chongstitvatana, 2011; Mantzaris et al., 2011; Mehrabi et al., 2012; Jaddi et al., 2013, 2016). Recently, particle swarm optimization (PSO) has also been used for MLP optimization (Min et al., 2011; Aladag et al., 2013). Other algorithms such as the cuckoo search (CS) (Valian et al., 2011; Nawi et al., 2013), bat algorithm (BA) (Mishra et al., 2012; Jaddi et al., 2015a,b), artificial bee colony (ABC) (Hsieh et al., 2011), ant colony optimization (ACO) (Mavrovouniotis and Yang, 2015; Salama and Abdelbar, 2015), and fish swarm algorithm (FSA) (Song et al., 2014) have also been applied to train MLPs. A comparison of the BA, GA, PSO, BP, and LM for training feed forward neural networks can be found in Khan and Sahai (2012).

* Corresponding author.
E-mail addresses: najmehjaddi@gmail.com (N.S. Jaddi), salwani@ukm.edu.my (S. Abdullah).

https://doi.org/10.1016/j.engappai.2017.09.012
Received 20 December 2016; Received in revised form 27 August 2017; Accepted 11 September 2017
Available online 5 November 2017
0952-1976/© 2017 Elsevier Ltd. All rights reserved.

In this paper, we propose an optimization methodology based on the kidney-inspired algorithm (KA). This algorithm was recently proposed in Jaddi et al. (2017b). The KA is inspired by the four steps of urine formation in the kidneys of the human body. The KA begins with an initial population that contains units of water and solutes (solutions). In each iteration, filtration of the solutes in the population is performed by using a filtration rate that is calculated based on the mean of the objective functions (MOF) of all the solutes. The filtered blood (FB) retains the solutes that are filtered and the rest of the solutes are transferred to waste (W). A solution allocated to W is reabsorbed if after affecting the reabsorption operator it can become a member of FB, otherwise it is excreted from W. On the other hand, if a solution allocated to FB is not better than the worst solution in FB, it is secreted. The best solution is ranked after checking all the solutions in the population. Then W and FB are merged and the filtration rate is updated.

In all optimization algorithms the major aim is to find the global optimum. To achieve this aim, it is crucial to obtain a balance between exploration and exploitation. In the KA, the movement of solutes and the filtration process provides the algorithm with a good exploration capability. In addition, exploitation in the KA is the result of reabsorption. To improve the balance between exploration and exploitation, in this paper we propose a controlled filtration rate that is provided by changing the value of α in each iteration of the search process. In the basic KA, the value of α in the filtration rate (*fr*) formula is a fixed value (in the range of [0, 1]), which is set in advance in the initialization stage of the algorithm. As the choice of the value of α has a direct effect on exploration and exploitation in the search space, in this paper the value of α is set to the minimum value of zero at initialization to get more exploration in the early iterations and it is increased during the search to get more exploitation in the later iterations until it reaches the maximum value in the last iteration. This modification makes the algorithm a self-adaptive algorithm that does not require parameter tuning in advance. Furthermore, the convergence ability of the modified algorithm is enhanced by achieving a better balance between exploration and exploitation.

In addition, in this paper the separate use of three chaotic maps to generate a chaotic sequence is investigated as an alternative to using a random number in the movement formula of the modified KA in order to assess the ability of each map to help to achieve superior results. In the context of this area of research, chaos can be defined as a random-like procedure that is set up in dynamic and nonlinear systems. Generally speaking, the basic metaheuristic algorithms in the literature make use of uniform probability to produce random numbers. In the proposed modified KA, a random number can be generated by iterating one action of the chosen chaotic map. A chaotic map is started from a random initial value and then is generated in each iteration by using one action of a chaotic map. Recently, some researchers (Charrier et al., 2010; Talatahari et al., 2012) have employed the chaotic map for optimization problems. Others (Min et al., 2011; Pan et al., 2011; Jaddi et al., 2015a,b) have presented the use of a chaotic map instead of a random number for many applications. In light of the potential of this approach, a combination of a chaotic map instead of a random number with the proposed modified KA is presented in this paper.

The rest of this paper is organized as follows: Section 2 provides a description of the KA. Then Section 3 explains the proposed modified KA in detail. This is followed by Section 4 which presents and discusses the results of applying the proposed method to some benchmark datasets and to real-world rainfall data. Finally, Section 5 summarizes the work and draws some conclusions.

2. Kidney-inspired algorithm

The KA was first proposed in Jaddi et al., (2017b). It is based on the function of the kidneys in the human body. Kidneys, which are the main biological structure in the human urinary system, filter blood so as to handle the amount of ions in the blood and also to eliminate excess water and wastes that are then excreted via urine. The filtration process in the kidneys starts in the glomerular capillaries where dissolved substances are moved into the tubules and reabsorption is performed. By reabsorption the solutes from the tubules are returned to the bloodstream. The movement of the solutes towards the renal tubule is considered as a secretion process as the excess is excreted in the urine. The kidney process can be summarized as four steps: filtration, reabsorption, secretion, and excretion.

In the initialization stage of the KA, as in other population-based algorithms, a random population of solutions is generated and their objective functions are computed. In each iteration, a new solution is created for all solutions by moving in the direction of the best solution found so far. In this algorithm, the movement of a solution is formulated as follows:

$$Sol_{i+1} = Sol_i + rand(Sol_{best} - Sol_i) \tag{1}$$

where *Sol* is a solution in the population of the KA, Sol_i is a solution in the ith iteration, *rand* is a random number, and Sol_{best} is the best solution found so far.

The solutions in the population that are of higher quality are filtered into FB by applying the filtration operator and the rest of the solutions are moved into W. In the KA, filtration is performed by using a filtration rate that is calculated and updated in each iteration. The filtration rate is calculated as follows:

$$fr = \alpha \times \frac{\sum_{i=1}^{p} f(x_i)}{p} \tag{2}$$

where *fr* represents the filtration rate, α denotes a constant value in the range of [0, 1], p is the population size, and $f(x_i)$ represents the objective function of solution x in the ith iteration. The algorithm follows the rule below to determine whether to accept the solution as a member of FB or W:

- If the quality of the solution $> fr$, accept the solution as a member of FB.
- If the quality of the solution $< fr$, accept the solution as a member of W.

If a solution is allocated to W, another opportunity is given to the solution to improve itself so that it can then be moved into FB. If this chance is not satisfied, excretion of the solution from W is performed and another random solution is inserted into W instead. On the other hand, if, after the filtration operation, a solution is allocated to FB and the quality of that solution is not better than the worst solution in FB, it is secreted (removed) from FB. However, if the solution is better than the worst solution in FB, the worst solution is secreted. Then the solutions in FB are ranked and the best solution found so far is updated. After that, FB and W are merged and the filtration rate is updated. This process is repeated until the termination criterion is reached. The pseudocode of the KA is shown in Fig. 1.

3. Modification of the KA

3.1. Control of filtration rate

In any optimization algorithm, exploration and exploitation are two fundamental concepts for finding the optimum solution. Exploration involves searching a large segment of the search space with the hope of finding other promising solutions that are yet to be refined. This process diversifies the search in order to prevent the algorithm from becoming trapped in a local optimum, and thus a global search is performed. Exploitation involves searching a limited and promising region of the search space with the hope of improving a promising solution that is already on hand. This process intensifies the search, and thus a local search is performed. In the KA, the role of the filtration process is to separate high-quality candidate solutions from low-quality ones. Therefore, in this algorithm, if more solutes are filtered and moved to

Download English Version:

https://daneshyari.com/en/article/6854302

Download Persian Version:

https://daneshyari.com/article/6854302

Daneshyari.com