

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,100

Open access books available

149,000

International authors and editors

185M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Designing Artificial Neural Network Using Particle Swarm Optimization: A Survey

Pooria Mazaheri, Shahryar Rahnamayan and Azam Asilian Bidgoli

Abstract

Neural network modeling has become a special interest for many engineers and scientists to be utilized in different types of data as time series, regression, and classification and have been used to solve complicated practical problems in different areas, such as medicine, engineering, manufacturing, military, business. To utilize a prediction model that is based upon artificial neural network (ANN), some challenges should be addressed that optimal designing and training of ANN are major ones. ANN can be defined as an optimization task because it has many hyper parameters and weights that can be optimized. Metaheuristic algorithms such as swarm intelligence-based methods are a category of optimization methods that aim to find an optimal structure of ANN and to train the network by optimizing the weights. One of the commonly used swarm intelligence-based algorithms is particle swarm optimization (PSO) that can be used for optimizing ANN. In this study, we review the conducted research works on optimizing the ANNs using PSO. All studies are reviewed from two different perspectives: optimization of weights and optimization of structure and hyper parameters.

Keywords: particle swarm optimization, artificial neural network, swarm intelligence, optimization, evolutionary algorithms

1. Introduction

ANN has been considered as an intelligent universal mechanism of dealing with function approximation, optimal design, process estimation, and prediction, pattern recognition, and other applications. Because of ANNs adaptability over a range of problems that involve decision making in uncertain situations, it is very attractive and popular amongst researchers. An ANN with many layers between the input layer and output layer is called Deep Neural Network (DNN). A large DNN may have millions of parameters that result in its learning process can take several days or even a month and need powerful hardware facilities. Also, there are several challenges which are required to address. For instance, the selection of the parameters, the structure of the networks, the selection of the initial values and the selection of the learning samples. If

ANN is designed with suitable parameters, it can be a powerful tool and lead to reducing learning time, minimizing loss function and make our predictions as accurate as possible. At this time, optimizers come to our aid. The optimizer helps us to build a better model, to improve the training process and some of them prevent to get trap in local optima.

Various methods exist to optimize a NNs. Backpropagation (BP) is one of them and it is used for optimizing Neural Networks [1–5]. BP training algorithm has different forms such as Gradient Descent, Levenberg-Marquardt, Conjugate Gradient Descent, Bayesian Regularization, Resilient, and One-Step Secant [6, 7]. For these algorithms, computational and storage requirements are different, some of these are suitable for an approximation of function and others for recognition of pattern, but they have disadvantages in a way or another such as the size of NN and storage requirements associated with them.

Another method is meta-heuristic algorithms. The objective of meta-heuristic algorithms is to discover global or local optimal solutions that are optimal with low cost. Meta-heuristic algorithms generally rely on various agents such as particles, chromosomes, and fireflies, searching iteratively to discover the global optimum or local optimum. Meta-heuristic is a collective concept of a series of algorithms such as evolutionary algorithm like Genetic Algorithm (GA) [8], naturally inspired algorithms such as PSO [9], trajectory algorithm like Tabu search [10], and etc.

In this paper, the focus is on PSO which is a nature-inspired algorithm for global optimization which can be utilized for solving the black-box optimization problem. Particle swarm is based upon simulation of the behavior of a school of fish or flock of birds. The use of active communication in such schools or swarms is a key concept. PSO like a GA is an optimization tool based upon population (swarm).

The goal of the study is to survey the papers which use PSO for optimizing ANN based on optimizing weights and biases and optimizing hyper parameters. There are some other surveys in this field, optimizing NN with evolutionary algorithms [11, 12], conventional and metaheuristic approaches [13], but this study only focuses on optimization of NN using PSO. In this survey, we try to categorize the existing methods for optimizing NN with PSO and show the role of hybrid and non-hybrid methods in optimization NN with PSO. The paper is organized as follows: In Section 2, Background Review, the architecture of Artificial Neural Network is explained with the backward and forward path for the BP method. Next, a brief overview of the PSO and its implementation is explained. Section 3, presents a review of the previous research related to optimizing ANN using PSO based on two categorizations. Section 4 will review challenges and gaps and finally, Section 5 will draw the Conclusion.

2. Background review

In this section, ANN, PSO, and the learning process in ANNs are reviewed.

2.1 Artificial neural network (ANN)

ANNs is considered a type of computational intelligence that is inspired by biological human systems like the brain process information [14]. ANNs are learned by instance and are configured for specific types of applications and problems through a learning system [15]. One of the most widely applied NN models is BP Neural

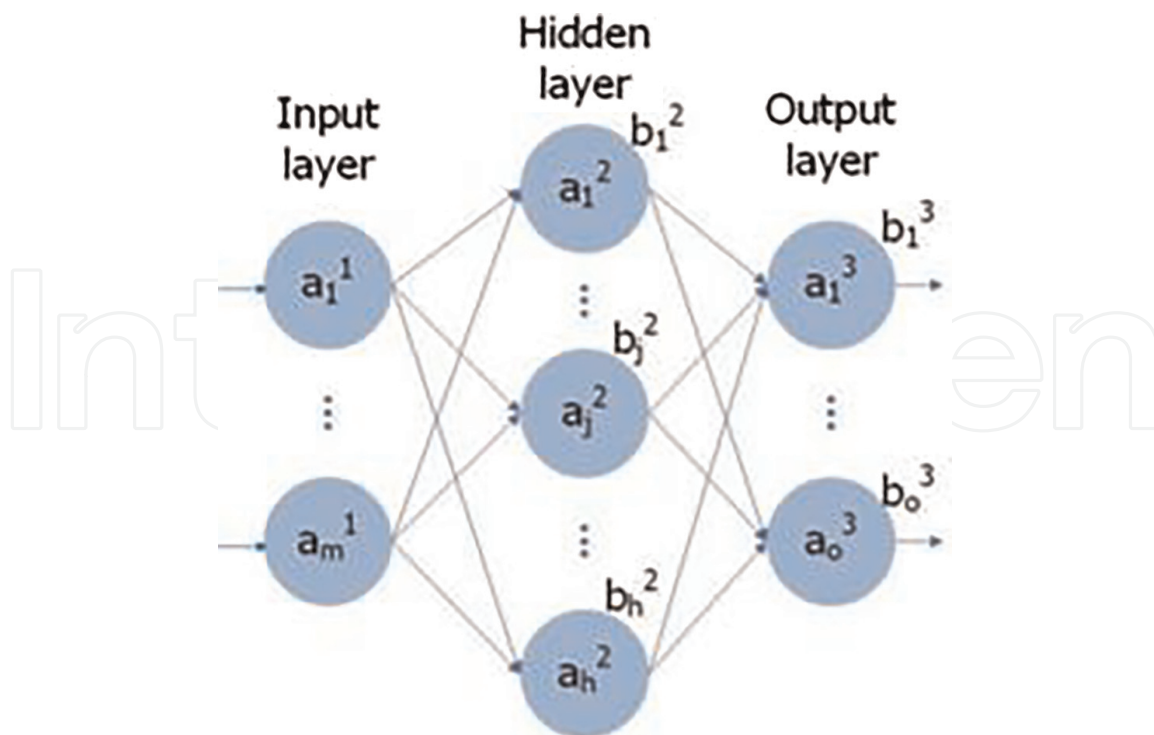


Figure 1.
 Three-layer topological structure of BPNN.

Network (**Figure 1**). The framework of BP Neural Network is made of three kinds of layers, input layer, hidden layer, and output layer. The input layer and output layer are representatives of input variables and output variables, so that the number of input and output variables is equal to the number of neurons; depending on the specific problem, there may be one or more hidden layers. An ANN is called a Deep Neural Network when it is made up of more than three layers it means an input layer, multiple hidden layers, and an output layer. In different layers, the neuron junctions have their own weight, each output neuron is multiplied by a given weight and after summing up the result is used as the input to the next neuron. In the next step, the neurons generate the output signals by computations that are based upon the function of transfer, and then the gradient descent method is used to minimize the error function in order that the inferred network value be similar to the value of the target output as far as possible [16].

The learning process in a network consists of two steps: Feedforward (FF) and BP. The key principle is using the gradient descent method to minimize the error function and make a small change to the weights of the network [17].

The learning process is usually implemented in ANNs by instances; the learning process of ANNs has three types: supervised learning (SL), unsupervised learning (UL), and semi-supervised learning. The first type of learning process is SL that is based upon the direct comparison between the expected and actual output. The optimization algorithms are based upon gradient descent like BP algorithm, they can be used to iteratively modify the connection weights hence minimizing the error. UL is the second type that is based upon the correlation of the input data. The learning rule is the most important factor in the learning algorithm and can determine the weight update rules. Some popular learning rules are the Competitive Learning rule, Hebbian rule, and Delta rule [11]. The third type of learning process is semi-supervised learning. In this approach, a large amount of unlabeled data is combined

with a small amount of labeled data. In fact, it can be said that semi-supervised learning falls between SL and UL.

2.2 Particle swarm optimization

The algorithm of PSO is used to optimize continuous nonlinear functions. It was proposed by J Kennedy and R Eberhart [18] and inspired by observations of collective and social behavior. PSO algorithm is considered a metaphor of social behavior. The social behavior is inspired by the movement of the flock to find food for the case of a bird flocking.

One of the advantages of PSO is the ability to deal with problems of multi-modal (i.e., multiple local optima) optimization and its simple implementation compared to associated strategies such as GA. PSO is used in various fields and has successfully been applied by several researchers to quantitative structure-activity relationship modeling, including kernel regression and k-nearest neighbor [19], minimum spanning tree for partial least squares modeling [20], piecewise modeling, and Neural Network training [21].

At first, the system will have a population of randomly created candidate solutions. Each candidate solution is called a particle, and it will throw into the problem space and will be given a random velocity. Each particle has memory and keeps track of previous corresponding fitness and best position. p_{best} call the previous best value. Therefore, p_{best} is associated only with a particular particle. The best value that exists between all the particles p_{best} in the swarm is g_{best} . The basic concept of the PSO technique is the acceleration of every particle toward its p_{best} and the g_{best} locations at every time step. Acceleration weights are random for both g_{best} and p_{best} locations. **Figure 2** indicates the concept of PSO. In this figure, P^k , P^{k+1} , V_{ini} , and V_{mod} are the current position, modified position, initial velocity, and modified velocity, respectively. $V_{p_{best}}$ is velocity considering $V_{g_{best}}$, and p_{best} is velocity considering g_{best} .

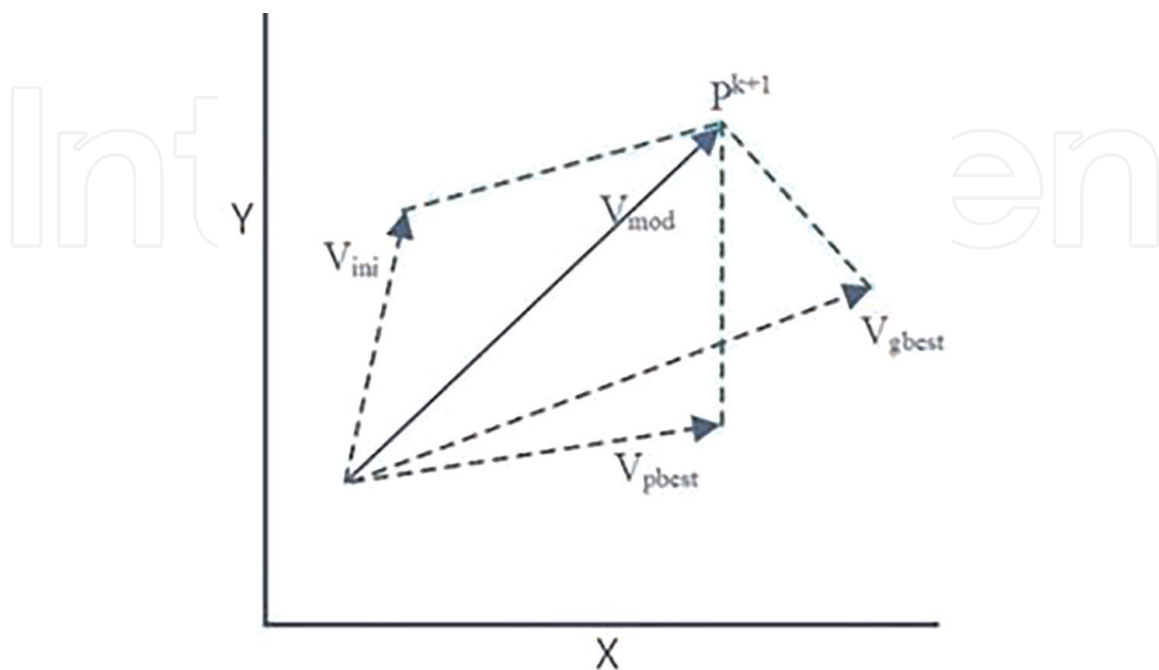


Figure 2.
Concept of changing a particle's position in PSO [22].

The PSO algorithm contains the following steps:

1. A population of particles initialized with random velocities and positions of d-dimension in the problem space.
2. Evaluate the desired optimization fitness function in terms of d variables for each particle.
3. *pbest* compare with particle's fitness evaluation. If *pbest* is worse than the current value then set *pbest* value and *pbest* location equal to the current value and the current location, respectively in d-dimensional space.
4. The population's overall previous best compare with fitness evaluation. When the *gbest* is worse than the current value then *gbest* changes to the current particle's array value and index.
5. Change the position and velocity of the particle according to Eqs. (1) and (2), respectively. $rand_1$ and $rand_2$ are two uniform random vectors. X_{id} and V_{id} also show the position and velocity of ith particle that has d-dimension, respectively.

$$V_{id} = (V_{id} * W) + (rand_1 * c_1 * (P_{bestid} - X_{id})) + rand_2 * c_2 * (G_{bestid} - X_{id}) \quad (1)$$

$$X_{id} = (V_{id} + X_{id}) \quad (2)$$

6. Step (2) is repeated until a criterion is met. This criterion usually is a maximum number of iterations or sufficiently suitable fitness calls.

PSO has several control parameters: W is the weight inertia that controls the exploitation and exploration of the search because it adjusts velocity dynamically. Asynchronous updates are less costly than synchronous updates. V_{max} is the largest velocity that is possible for the particles, if V_{max} is less than the velocity particle, the velocity of the particle decreases to V_{max} . Therefore, the fitness of search and resolution is directly affected by V_{max} . Particles are trapped in local minima when V_{max} is too low, and particles will move beyond good solution if V_{max} is too high. c_1 (cognition) and c_2 (social components) are the constants of acceleration. They change a particle velocity toward *gbest* and *pbest*. The tension is determined by velocity in the system. In a search space, a swarm of particles can be used globally or locally. In the PSO's local version, the entire procedure is the same and the *gbest* is replaced by the l_{best} .

3. The optimization of ANNs based on PSO algorithm

Methods with the aim of optimal design of an ANN utilizing PSO have been divided into two main categories: optimizing weights and optimizing structure and hyper parameters. These categories are further divided into two subcategories, including non-hybrid optimization and hybrid optimization, which in former authors only used PSO to optimize ANNs weights, in latter, hybrid methods have been utilized. Both subcategories have been reviewed in the following subsections A and B.

3.1 Weights and biases optimization

Some papers focused on weights for optimizing ANNs. They can be divided into two categories. First, those related to Non-Hybrid Optimization, and second, those related to Hybrid Optimization.

3.1.1 Non-hybrid optimization

Some studies used classical PSO to optimize NN and for showing their accuracy, they compared their solution with conventional approach optimization like BP. The first paper that falls into this category is from Gudise and Venayagamoorthy [23] published in 2003. They made a comparative study on the computational requirements of the BP and PSO algorithm for NN as training algorithms. They presented results for an FFNN learning a nonlinear function and indicated that the FFNN weights converge faster when the PSO is used instead of the BP algorithm. Later, in 2005, a modified PSO was presented by Zhao et al. [24], which adjusts the velocities and positions of the particle on the basis of the best positions that are earlier visited by other particles and themselves, and includes the method of diversifying the population to prevent premature convergence. In this paper, PSO is compared with the conventional BP to learn a nonlinear function for training a FFNN. The considered problem is how accurate and fast can the weights of NN be determined by BP and PSO to learn a common function. Another research that compared the PSO and BP for optimization NN was proposed by Ni et al. [25] in 2014. They introduced PSO for stochastic global optimization in NN training to solve the flaws of the traditional BP network in cementing prediction. They showed their method's training time is shorter than BP network and also the prediction accuracy that they obtained is high. Following by that, Liu et al. [26] to predict the high-speed grinding temperature used a BP NN based upon PSO algorithm (PSO-BP). They compared their method with gradient descent training BP NN which trained based upon Levenberg Marquardt (LM) algorithm and showed that PSO- BP performs better than the other methods in predicting the grinding temperature. In this paper, the authors used PSO algorithm for training BP NN to obtain a set of weights and biases, which could minimize the Mean Square Error (MSE).

In some studies, firstly PSO was improved and then used for optimizing NN. First, Bai et al. [27] used improved PSO- BP NN to improve the prediction accuracy of pest occurrence cycle. Their method used inertia weight to improve the PSO algorithm. Next, they used improved PSO to optimize the thresholds and weights of BP NN. Then, they established a pest prediction model using a rough set and an improved PSO- BP network. Their research showed that the number of iterations can be reduced by the improved PSO algorithm. Second, Liu and Yin [28] optimized BP NN with using an improved PSO. In the new algorithm, PSO used enhanced adaptive acceleration factor and also enhanced adaptive inertia weight to justify the initial weight value and biases of BP NN. At the end, simulation results indicated that the new algorithm is able to enhance convergence rate and precision of prediction of BP NN, that decreases the error of prediction. Later on, Nandi and Jana [29] rectified the problem by formulating a new inertia weight strategy for PSO called PPSO which balanced the exploitation and exploration properly while training ANN and compared their model with 4 other training algorithms. For all benchmark datasets, PPSO showed better performance with regard to avoiding local minima and convergence rate as well as

better accuracy. The proposed PPSO reduced the trapping risk in local minima with a very well convergence rate.

In some works, PSO was employed to optimize NN in different fields such as medical imaging, energy consumption, civil engineering, etc. For example, in medical imaging, Wang et al. [30] introduced a method of relatively recent image enhancement for improving the brain image contrast. Then, they presented the Predator-Prey PSO (PP-PSO), which is a modification of traditional PSO to train weights of single-hidden layer NN. In their method, they utilized the MSE as an objective function. Later on, Zhang et al. [31] developed a technique that could automatically establish diagnoses from the brain magnetic resonance images. First, the processing brain imaging was implemented. Second, from the volumetric image, one axial slice was selected. Third, a single-hidden layer NN was utilized as a classifier. Finally, for training the weights and biases of the classifier, a predator-prey PSO was proposed. Their method performs better than the human observers and 10 state-of-the-art approaches. Also, in energy consumption, Le et al. [32] proposed four novel AI techniques. They utilized these models for predicting the heating load of buildings' energy efficiency. Their model was based upon meta-heuristics algorithms and the potential of ANN, including Imperialist Competitive Algorithm (ICA), Artificial Bee Colony optimization (ABC), GA, and PSO. For the buildings prediction of the heating load of energy efficiency with PSO-ANN model, the parameters of the PSO algorithm were set up before optimization of the ANN model consisting of the number of particle swarms, maximum particle's velocity, individual cognitive, group cognitive, inertia weight, and maximum number of iterations. Then, PSO algorithm optimized the biases and weights of the initialized ANN. The best PSO-ANN model was determined with the lowest Root Mean Squared Error (RMSE). The GA provided the highest performance in optimizing the ANN model, to forecast the HL of EEB systems. The remaining meta-heuristics algorithms provided more unsatisfactory performance, in contrast to the performance of the ICA-ANN, PSO-ANN, and ABC-ANN models. In the civil engineering field, Chatterjee et al. [33] proposed a PSO-based approach to train the NN for predicting structural failure of the reinforced concrete buildings. In order to find the optimal weights for the NN classifier, the PSO algorithm was involved. In the first phase, NN training, PSO minimizes the RMSE to achieve the optimal input weight vector to the input layer of the ANN. Next, to get ingenuity, the NN-PSO model was compared with MLP-FFN classifier (multilayer perceptron FF network) and NN. Finally, the supremacy of the presented NN-PSO in comparison to the NN and MLP-FFN classifiers was shown by the experimental results.

Besides, some studies have focused on only a specific version of NN like random FF NN (RFNN) and tried to use PSO to optimize them. For example, Xu and Shu [34] at the beginning, considered the advantages of both PSO and non-iterative learning to train RFNN. Pacifico and Ludermir [35] presented to utilize PSO and clustering analysis to optimize RFNN input weights and biases. In this study, they employed a local best neighborhood scheme for PSO population updating where each individual only followed some members belongs to its immediate neighborhood. Following by that, an improved PSO was proposed by Ling et al. [36], which encoded the input-to-output sensitivity information of RFNN to optimize the input weights and biases.

Some researchers to find a better answer for their problems, used different types of PSO such as cooperative PSO, Cultural Cooperative Particle Swarm Optimization (CCPSO), and multi-phase PSO. The cooperative PSO is an enhanced PSO that was presented by Van den bergh and Engelbrecht [37]. They obtained good results by applying this method on NN training. In this method, input vectors are divided into

several sub vectors that are optimized in their own swarms cooperatively. In this case, performance is improved due to splitting the main vector into several sub vectors which in turn results in better credit assignments and decreases the chance to omit a possible good solution for a certain component in the vector. Lin et al. proposed [38] a CCPSO approach that a collection of multiple swarms which interact by exchanging information. They applied CCPSO for optimizing a fuzzy NN and result in it performed better than BP and GA. Next, Multi-phase PSO (MPPSO) was proposed by Al-kazemi and Mohan [39] in 2002. Training of ANNS by MPPSO is another variation which evolves simultaneously multiple groups of particles that change the direction of search in different phases of the algorithm. Each particle in this method is in a specific group and phase at a given time. MPPSO boosts the broader exploration of the search space, increases population diversity, and prevents premature convergences. Furthermore, MPPSO has different update equations comparing to the basic PSO and permits changes to the locations of the particle that only lead to some improvements. Many researchers chose a different path and have used multiobjective PSO for optimizing NN. For example, Carlos Coello et al. [40] proposed Multiobjective Particle Swarm Optimization (MOPSO) and used this method as a searching strategy for improving NN.

Some studies utilized PSO for solving large-scale problems. For instance, a novel study for high-dimensional datasets was proposed to optimize the weights of NN with PSO and some other Evolutionary Computation (EC) methods. Xue et al. [41] presented a self-adaptive parameter and strategy-based PSO (SPS-PSO) algorithm and then they used this method to optimize FFNN with feature selection. The authors divided the experiments into two groups. They utilized SPS-PSO and three other evolutionary computation methods, GA, PSO, and biogeography-based optimization for directly optimizing the FNN's weights in the first group. In the other group, firstly, they employed SPS-PSO-based feature selection on the initial datasets and obtained eight comparatively smaller datasets with the K-Nearest Neighbor (KNN). Then, the new datasets were utilized as the inputs for FNN. They optimized the FNN weights one more time by SPS-PSO and three other evolutionary computation methods. The experimental findings showed that SPS-PSO had the vantage to optimize the FNN weights in comparison to the other methods of EC. Meanwhile, the feature selection based upon SPS-PSO can decrease the size of solution and computational complexity, whereas ensuring the accuracy of classification, it is utilized for preprocessing the datasets for FNN.

3.1.2 Hybrid optimization

In this subcategory, authors used hybrid methods to optimize weights of ANNs.

Some studies combining GA and PSO for optimizing ANN's weights. For instance, in 2018, Anand and Suganthi [42] optimized ANN with using a hybrid algorithm of PSO and GA. Then, they used this model to enhance the measurement of electricity demand in India. Their model has higher performance and reliable accuracy than ANN-PSO or ANN- GA that are single optimization models. They used hyperbolic tangent and identity as activation function in hidden layer and output layer, respectively, the sum of squares as error function and mean absolute percentage as an indicator of the quality of prediction. PSO by using linear and quadratic regression models together, optimized the weights of socio-economic indicators and performs a search for the best fitted members that lessen the error. Also, Ma [43] developed a short time traffic flow prediction software on the basis of BP NN that could be used

for predicting urban short-term traffic flow. The GA-based improved PSO was utilized for optimizing BP NN weight threshold to improve BP NN prediction accuracy. The results showed that this software could accurately and quickly predict the information of road traffic flow at the next moment, which could extremely reduce urban road traffic pressure. Next, Xiao et al. [44] proposed a new three-stage nonlinear ensemble model. In this model, three various types of NN based models, including elman network, generalized regression NN, and wavelet NN built by three non-overlapping training sets. The results of the study showed the ensemble ANNs-PSO-GA method enhanced the prediction performance over other linear combination and individual models.

In some works, researchers preferred combining PSO and wavelet to obtain a better answer. In 2015, Zhang et al. [45] with using Wavelet Entropy (WE) proposed a novel computer-aided diagnosis system to extract some features from Magnetic Resonance (MR) brain images, followed by FFNN with training method of a Hybridization of PSO and biogeography-based optimization (HBP), which combined the exploration ability of biogeography-based optimization and exploitation ability of PSO. They used MSE as an objective function to optimize weights with PSO. The proposed WE+HBP-FNN method obtain nearly perfect detection pathological brains in MRI scanning. Next, a novel hybrid approach called Switching PSO-Wavelet Neural Network (WNN) was proposed by Yang Lu et al. [46] in 2015 to enhance recognition accuracy in face recognition that is one of the important research problems in computer vision. They used the algorithm of the recently proposed Switching PSO (SPSO) for optimizing the weight parameters, translation factors, scale factors, and threshold in WNN. The proposed method, SPSO- WNN, has a higher learning ability and fast convergence speed than conventional WNN. Especially, for overcoming the difference between the local search and the global search, which facilitates jumping the local minimum, a velocity-updating equation depended on mode with Markovian switching parameters is presented in SPSO. They showed their method has a much better performance compared to PSO-WNN, GA-WNN, and WNN.

Following by that, some studies tried to use a hybrid model to propose better models compare to BP. Firstly, in 2008, Chen et al. [47] used a hybrid evolutionary algorithm that is based upon PSO and AFSA, also referred to as AFSA-PSO- parallel-hybrid evolutionary (APPHE) algorithm in FFNN training. They showed that FFNN training by the novel hybrid evolutionary algorithm compared to FFNN trained by Levenberg-Marquardt BP (LMBP) algorithm, show high stability toward the optimal position, satisfactory performance, convergent accuracy and converges quickly. In this research, both the output transfer function and the hidden transfer function were sigmoid function. Secondly, a hybrid crop classifier was presented by Zhang and Wu [48] for polarimetric synthetic aperture radar images in 2011. The feature sets included the cloude decomposition known as H/A/ α decomposition, span image, and the gray-level co-occurrence matrix-based texture features. Then, Principle Component Analysis (PCA) reduced the features. Lastly, an FNN was built and trained by Adaptive Chaotic PSO (ACPSO). The results on flevoland sites showed the superiority of ACPSO to BP and adaptive BP.

Some works prefer to combine BP and PSO to make a hybrid model for optimizing weights of NN. In 2007, Zhang et al. [49] proposed a hybrid algorithm combining BP with PSO algorithm. For training the weights of FFNN, the hybrid algorithm can benefit from employing strong global searching and local searching ability of the PSO and the BP algorithm, respectively. Firstly, in the PSOBP algorithm, a heuristic algorithm was adopted by them to give a transition from PSO to gradient descending

search. Also, they gave three kinds of encoding strategy of particles and gave the different problem areas that every encoding strategy was actively used in. They showed that in terms of accuracy and convergent speed, the proposed hybrid PSOBP algorithm performs better than the adaptive PSO and BP algorithm. Following by that, in 2011, Yaghini et al. [50] proposed a hybrid improved opposition-based algorithm that is based upon PSO and GA (HIOPGA) methods and then compared BP algorithm with their method on several benchmark problems. In fact, their method combined ability of two algorithms. This algorithm began training using a particle population. During the algorithm iteration, when improved opposition-based PSO cannot improve some particles' position, a subpopulation of such NNs is created and sent to GA. Now, the HIOPGA can find better NN to replace in the population by utilizing the GA operators, mutation, and crossover. Also, Kartheeswaran and Durairaj [51] in 2017, for image reconstruction, presented the sequential and parallel data implementing the decomposition strategies on a PSO algorithm based ANN weights optimization. They utilized a hybrid algorithm combining BP with PSO algorithm. They used PSO with BP-ANN for optimizing the different parameters including hidden layer sizes, number of hidden nodes, and optimize the network connection's weights. In fact, this study, by optimizing the weights of connection, presented the application of a hybrid model for the reconstruction of Shepp-Logan head phantom image.

3.2 Optimizing structure and hyper parameters

In this category, there are a few papers that have focused on optimizing hyper parameters. There are two subcategories: first Non-Hybrid Optimization, second, Hybrid Optimization.

3.2.1 Non-hybrid optimization

In this subcategory, the authors used non-hybrid methods to optimize structure and hyper parameters.

In 2000, Zhang and Shao [52] were the first authors that presented a PSONN system for evolving network architecture and the weights of ANNs, alternately. They used evolved ANNs in modeling product quality estimator for a fractionator of the hydrocracking unit in the oil refining industry. Carvalho and Ludermir [53] proposed another study that was inspired by Zhang and Shaos methodology but introduces the weight decay heuristic in the weight adjustment process in an attempt to obtain more generalization control. They analyzed the use of the PSO for the optimization of architectures and weights of NN with the aim of the performance of better generalization by making a compromise between low training errors and low architectural complexity and utilized them for specific problems in the medical field that fall within benchmark classification category. The results that they obtained, showed that a PSO-PSO based method indicates an acceptable alternative for optimizing architectures and weights of NNs of MLP. Xue et al. [54] similar to Carvalho and Ludermir tried to optimize weight and architecture simultaneously. They found a variable-length PSO to optimize both the number of hidden nodes and input weights, simultaneously. Particles with various lengths which showed various network configurations can be solved with a new particle update strategy presented in this study.

Many researchers improved the algorithms themselves to optimize architecture. Here are some examples: Carvalho [55], proposed a PSO-PSO method, in which a PSO was employed for optimizing weights that were nested under another PSO which was

employed to optimize the architecture of FNN by deleting or adding hidden nodes. Next, in 2009, Kiranyaz et al. [56] proposed a multidimensional PSO approach to construct FNN by utilizing an architectural space, automatically. Furthermore, the individuals in the swarm population have been designed in a way that it optimized both the weights and architecture of an individual in every iteration.

PSO for optimizing NN's architecture used by researchers in different areas and topics such as communication theory, civil and medical engineering. PSO has been utilized widely to address the optimization problems existing in communication theory. Das et al. [57] optimized ANN by using PSO for the problem of channel equalization in 2013. In this paper, they used PSO algorithm to optimize all the variables including network parameters and network weights. In fact, they used the PSO to optimize the number of input neurons, hidden neurons, the type of transfer functions, and the number of layers. The novelty in this paper is that they take care of suitable network topology. Extensive simulations proposed in this research showed that, as compared to other ANN-based equalizers as well as neuro-fuzzy equalizers, the proposed equalizer performs better in all noise conditions. An interesting application area of PSO is civil engineering. The application of an improved PSO technique was proposed by Asadnia et al. [58] for training an ANN to predict water levels for the Heshui Watershed. The results showed that the PSO-based ANNs performed better to predict the peak and low water levels compare with the LM-NN model. Additionally, IPSONN had a quicker convergence rate in comparison with CPSONN. In medical engineering, an adaptive CPSO was developed by Zhang et al. [59] to train the parameters of FFNN, with the purpose of accurate classification of magnetic resonance (MR) brain images. The classification accuracy of the presented technique was 98.75% on 160 images.

Many works used basic PSO to optimize NN's architecture. In a study by Chunkai et al. [60], in 2000, the network structure is adaptively adjusted and the PSO algorithm is applied to evolve the nodes of the NN with a specific generated structure. The techniques such as the combination of partial training and evolving added nodes are used to generate the desired architecture and then PSO is employed to evolve the nodes of the predefined structure. In another study in 2013, Wang et al. [61] used the BP NN to build an estimation model for the cost of plastic injection modeling parts to decrease the complication of the conventional procedures of estimating all the costs. They have made an estimation model for costs on the basis of the superior capability in forecasting and diagnosis for BP NN, and the capability of the great solution caused by PSO was utilized to get the parameters for BP NN, such as the number of hidden nodes and layers, initial weight, learning rate, hence learning and training for the network were made to perform better and be more precise. In this study, the sigmoid function was utilized as activation function and transfer function. In 2018, Qi et al. [62] presented a combination of ANN and PSO for forecasting the unconfined compressive strength of Cemented Paste Backfill (CPB). The authors used ANN for non-linear relationships modeling and also utilize PSO for tuning the ANN architecture. In fact, in this work, PSO optimized the number of neurons and hidden layers. The findings indicated that PSO was efficient for optimizing the ANN architecture. Also, comparing the values of forecast UCS with experimental values indicated that the model of optimal ANN was very precise to predict the strength of CPB.

3.2.2 Hybrid optimization

In this subcategory, authors employed hybrid methods to optimize the structure and hyper parameters of NNs.

In a study, J Yu et al. [63] presented a new evolutionary ANN algorithm called IPSONet. This algorithm was based on an improved PSO. The improved algorithm utilized parameter automation strategy, mutations, crossover, and velocity resetting to enhance the performance of the classical PSO in fine-tuning of the solutions and global search. To solve the design problem of FFNN, the improved PSO was used by IPSONet. They used the improved PSO to evolve simultaneously weights and structure of ANNs by the evolutionary scheme and a specific individual representation. Next, researchers employed hybrid GA and PSO to optimize structure and hyper parameters to obtain a better answer. For example, Juang [64] in 2004, presented a modified PSO Hybrid of GA and PSO (HGAPSO) method that was employed to design NN. In this method, the individuals of the next generation are created not only by crossover and mutation operators but also by PSO. The upper half of the best performing individuals in a population are enhanced using PSO and the other half is generated by applying the crossover and mutations. Unlike GA, HGAPSO removes the restrictions of evolving the individuals within the same generation. In this article, the proposed method is another variation of PSO for fixed structure ANNs where only weights are adjusted.

4. Challenges and gaps

Particle Swarm Optimization is a heuristic optimization method that performs well for various optimization problems. But like other swarm intelligence-based optimization technique, PSO has some disadvantages including sensitivity to parameters, high computational complexity, slow convergence. The first reason is that PSO is unable to employ the crossover operator as utilized in genetic algorithm or differential Evolution. Therefore, the distribution of suitable information between candidates is not at an essential level. Another factor can be the fact that PSO is unable to handle appropriately the relationship between exploration and exploitation, in fact, local search and global search, so it often converges to a local minimum quickly. One of the solutions that can address these problems is hybridization. Numerous optimization algorithms have been utilized for ANN optimization like GA that some of them can be seen in this paper. For future work, PSO can be hybridized with some of these optimization algorithms like GA, SA, TS, DE, ABC, and ACO to develop hybrid approaches in order to achieve better exploration ability.

Another challenge is that study of PSO for optimizing NN had great achievements but there is no in-depth research on theoretical aspects. So, we think it can be interesting to conduct another study of both the run-time and convergence properties of PSO for optimizing NN. In addition, there are not many works related to PSO implemented in parallel for optimizing NN. Thus, it can be a potential path for future research. Moreover, considering other Deep Learning.

Finally, stream data poses significant challenges in this area. In a non-stationary environment, like weather forecasting and stock-price market, data comes in the stream. So, it can be a good topic to design strategies for the dynamic training of NN using PSO.

5. Conclusion

ANN as a fertile approach to developing an intelligent information processing system has been introduced. Specifically, ANNs have been seen as a powerful tool in

No.	Author/Authors	Year	Optimization task	Types
1	Zhang and Shao [52]	2000	Optimizing weights and structure	Non-Hybrid
2	Chunkai et al. [60]	2000	Optimizing structure	Non-Hybrid
3	Al-kazemi et al. [39]	2002	Optimizing weights	Non-Hybrid
4	Gudise and Venayag [23]	2003	Optimizing weights	Non-Hybrid
5	Vandenbergh and Engelbrecht [37]	2004	Optimizing weights	Non-Hybrid
6	Coello et al. [40]	2004	Optimizing weights	Non-Hybrid
7	Juang et al. [64]	2004	Optimizing structure	Hybrid
8	Meissner et al. [65]	2005	Optimizing weights	Non-Hybrid
9	Zhao et al. [24]	2005	Optimizing weights	Non-Hybrid
10	Carvalho and Ludermir [66]	2006	Optimizing weights	Hybrid
11	Xu and Shu [34]	2006	Optimizing weights	Non-Hybrid
12	J Yu et al. [63]	2007	Optimizing weights and structure	Hybrid
13	Carvalho and Ludermir [55]	2007	Optimizing weights structure	Non-Hybrid
14	Carvalho [53]	2007	Optimizing weights and structure	Non-Hybrid
15	Zhang et al. [49]	2007	Optimizing weights	Hybrid
16	Lin et al. [38]	2008	Optimizing weights	Non-Hybrid
17	Chen et al. [47]	2008	Optimizing weights	Hybrid
18	Kiranyaz et al. [56]	2009	Optimizing structure	Non-Hybrid
19	Zhang et al. [59]	2010	Optimizing structure	Non-Hybrid
20	Zhang and Wu [48]	2011	Optimizing weights	Hybrid
21	Yaghini et al. [50]	2011	Optimizing weights	Hybrid
22	Zhao [67]	2012	Optimizing weights	Non-Hybrid
23	Wang et al. [61]	2013	Optimizing structure	Non-Hybrid
24	Armaghani et al. [68]	2013	Optimizing weights	Non-Hybrid
25	Das et al. [57]	2013	Optimizing weights and structure	Non-Hybrid
26	Pacifico and Ludermir [35]	2013	Optimizing weights	Non-Hybrid
27	Xue et al. [54]	2013	Optimizing weights and structure	Non-Hybrid
28	Asadnia et al. [58]	2014	Optimizing structure	Non-Hybrid
29	Xiao et al. [44]	2014	Optimizing weights	Hybrid
30	Bai et al. [27]	2014	Optimizing weights	Non-Hybrid
31	Ni et al. [25]	2014	Optimizing weights	Non-Hybrid
32	Yang Lu et al. [46]	2015	Optimizing weights and scale factors	Hybrid
33	Zhang et al. [45]	2015	Optimizing weights	Hybrid
34	Liu et al. [26]	2016	Optimizing weights	Non-Hybrid
35	Wang et al. [30]	2016	Optimizing weights	Non-Hybrid
36	Chatterjee et al. [33]	2016	Optimizing weights	Non-Hybrid
37	Liu and Yin [28]	2016	Optimizing weights	Non-Hybrid
38	Zhang et al. [31]	2017	Optimizing weights	Non-Hybrid

No.	Author/Authors	Year	Optimization task	Types
39	Kartheeswaran and Durairaj [51]	2017	Optimizing weights	Hybrid
40	Pradeepkumar and Ravi [69]	2017	Optimizing weights	Hybrid
41	Anand and Suganthi [42]	2017	Optimizing weights	Hybrid
42	Ling et al. [36]	2017	Optimizing weights	Non-Hybrid
43	Qi et al. [62]	2018	Optimizing structure	Non-Hybrid
44	Yang and Jiang [70]	2018	Optimizing weights	Non-Hybrid
45	Kong et al. [71]	2019	Optimizing weights	Non-Hybrid
46	Ma [43]	2019	Optimizing weights	Hybrid
47	Xue et al. [41]	2019	Optimizing weights	Non-Hybrid
48	Le et al. [32]	2019	Optimizing weights	Non-Hybrid
49	Chen et al. [72]	2019	Optimizing weights	Non-Hybrid
50	Nandi and Jana [29]	2019	Optimizing weights	Non-Hybrid

Table 1.
Optimization types that researchers used.

modern AI techniques. To utilize a prediction model based upon ANN, we face some challenges that ANN training is one of the major of them. For training ANN, conventional algorithms are used which results in researchers faced some problems. These conventional algorithms like backpropagation, are local search methods that exploit the current solution to produce a new solution. However, they lack exploration ability, hence, they often, finds local minima of an optimization problem. Unlike conventional approaches, metaheuristics like PSO are good at both exploration and exploitation and are able to solve simultaneous adaptation in each component of NN. In this paper, we present a survey of optimizing and training ANNs with using PSO that is one of the best metaheuristic algorithms for optimizing ANN. We try to review some studies conducted on optimizing ANN using PSO for different goals including comparing different methods results and solving various types of problems. In this study, all the papers are grouped into categories including the kind of PSO, year of publication, activation fitness function types, and what has been optimized. Findings in this study provide future direction for further work on optimizing ANN with using PSO (Table 1).

IntechOpen


IntechOpen

Author details

Pooria Mazaheri*, Shahryar Rahnamayan and Azam Asilian Bidgoli
Nature Inspired Computational Intelligence (NICI) Lab, Ontario Tech University,
Ontario, Canada

*Address all correspondence to: pooria.mazaheri@ontariotechu.ca

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Werbos PJ. Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*. 1990;**78**(10): 1550-1560
- [2] Werbos PJ. *The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting*. Vol. 1. John Wiley & Sons; 1994
- [3] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation *Tech. Rep.* 1985
- [4] Rumelhart DE, Hinton GE, Williams RJ. Learning representations by back-propagating errors. *Nature*. 1986; **323**(6088):533-536
- [5] Hagan M, Demuth H, Design MBNN. Boston, MA. 1996
- [6] Hagan MT, Menhaj MB. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*. 1994;**5**(6):989-993
- [7] Chen C, Lai H. An empirical study of the gradient descent and the conjugate gradient backpropagation neural networks
- [8] Goldberg DE, Holland JH. 1988
- [9] Tian Z, Fong S. *Optimization algorithms methods and Applications*. 2016
- [10] Glover F. Tabu search—part I. *ORSA Journal on Computing*. 1989;**1**(3): 190-206
- [11] Ding S, Li H, Su C, Yu J, Jin F. Evolutionary artificial neural networks: A review. *Artificial Intelligence Review*. 2013;**39**(3):251-260
- [12] Wistuba M, Rawat A, Pedapati T. ar Xiv preprint ar Xiv: 1905.01392. 2019
- [13] Ojha VK, Abraham A, Snášel V. Metaheuristic design of feedforward neural networks: A review of two decades of research. *Engineering Applications of Artificial Intelligence*. 2017;**60**:97-116
- [14] Lee KY, Cha YT, Park JH. Short-term load forecasting using an artificial neural network. *IEEE Transactions on Power Systems*. 1992;**7**(1):124-132
- [15] Banda E. Department of Electrical Engineering, University of Cape Town, Student thesis. 2006
- [16] Liu CL, Yang TY. Study on method of GPS height fitting based on BP artificial neural network. *Journal of Southwest Jiaotong University*. 2007;**2**(5)
- [17] Wu JB, Li WJ. Study on textile industry using BP neural networks. *Progress in Text Science and Technology*. 2007;**2**(2):7-10
- [18] Eberhart RC, Shi Y, Kennedy J. *Swarm Intelligence*. Morgan Kaufmann Publishers; 2001
- [19] Cedeño W, Agrafiotis DK. Using particle swarms for the development of QSAR models based on K-nearest neighbor and kernel regression. *Journal of Computer-Aided Molecular Design*. 2003;**17**(2):255-263
- [20] Lin WQ, Jiang JH, Shen Q, Shen GL, Yu RQ. Optimized block-wise variable combination by particle swarm optimization for partial least squares modeling in quantitative structure-activity relationship studies. *Journal of*

Chemical Information and Modeling.
2005;**45**(2):486-493

[21] Shen Q, Jiang JH, Jiao CX, Lin WQ, Shen GL, Yu RQ. Hybridized particle swarm algorithm for adaptive structure training of multilayer feed-forward neural network: QSAR studies of bioactivity of organic compounds. *Journal of Computational Chemistry*. 2004;**25**(14):1726-1735

[22] Yoshida H, Kawata K, Fukuyama Y, Takayama S, Nakanishi Y. A particle swarm optimization for reactive power and voltage control considering voltage security assessment. *IEEE Transactions on Power Systems*. 2000;**15**(4): 1232-1239

[23] Gudise VG, Venayagamoorthy GK. Comparison of PSO and backpropagation as training algorithms for neural networks

[24] Zhao F, Ren Z, Yu D, Yang Y. Application of an improved particle swarm optimization algorithm for neural network training

[25] Ni HM, Yi Z, Li PC, Tong XF. Application of BP network based on PSO algorithm in cementing quality prediction

[26] Liu C, Ding W, Li Z, Yang C. *The International Journal of Advanced Manufacturing Technology*. 2017;**89**: 2277-2285

[27] Bai T, Meng H, Yao J. *Neural Computing and Applications*. 2014;**25**: 1699-1707

[28] Liu T, Yin S. *Multimedia Tools and Applications*. 2017;**76**:11961-11974

[29] Nandi A, Jana ND. ar Xiv preprint ar Xiv: 1905.04522. 2019

[30] Wang H, Lv Y, Chen H, Li Y, Zhang Y, Lu Z. *Multimedia Tools and Applications*. 2018;**77**:3871-3885

[31] Zhang Y, Wang S, Sui Y, Yang M, Liu B, Sun J, et al. *Journal of Alzheimer's Disease*. 2018;**65**:855-869

[32] Le LT, Nguyen H, Dou J, Zhou J, et al. *Applied Sciences*. 2019;**9**:2630

[33] Chatterjee S, Sarkar S, Hore S, Dey N, Ashour AS, Balas VE. *Neural Computing and Applications*. 2017;**28**: 2005-2016

[34] Xu Y, Shu Y. *Evolutionary Extreme Learning Machine-based on Particle Swarm Optimization*. Springer; 2006. pp. 644-652

[35] Pacifico LD, Ludermir TB. Evolutionary extreme learning machine based on particle swarm optimization and clustering strategies. In: 2013 International Joint Conference on Neural Networks (IJCNN) (IEEE). 2013. pp. 1-6

[36] Ling QH, Song YQ, Han F, Lu H. An improved evolutionary random neural networks based on particle swarm optimization and input- to-output sensitivity. In: *International Conference on Intelligent Computing*. Springer; 2017. pp. 121-127

[37] Van den Bergh F, Engelbrecht AP. *IEEE Transactions on Evolutionary Computation*. 2004;**8**:225-239

[38] Lin CJ, Chen CH, Lin CT. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 2008;**39**:55-68

[39] Al-Kazemi B, Mohan CK. Training feedforward neural networks using multi-phase particle swarm optimization

- [40] Coello CAC, Pulido GT, Lechuga MS. *IEEE Transactions on Evolutionary Computation*. 2004;**8**: 256-279
- [41] Xue Y, Tang T, Liu AX. Large-scale feedforward neural network optimization by a self-adaptive strategy and parameter based particle swarm optimization. *IEEE Access*. 2019;**7**: 52473-52483
- [42] Anand A, Suganthi L. Forecasting of electricity demand by hybrid ann-pso models
- [43] Ma Q. Design of BP neural network urban short-term traffic flow prediction software based on improved particle swarm optimization
- [44] Xiao Y, Xiao J, Lu F, Wang S. *International Journal of Computational Intelligence Systems*. 2014;**7**:272-290
- [45] Zhang YD, Wang S, Dong Z, Phillip P, Ji G, Yang J. *Progress In Electromagnetics Research*. 2015;**152**: 41-58
- [46] Lu Y, Zeng N, Liu Y, Zhang N. A hybrid wavelet neural network and switching particle swarm optimization algorithm for face direction recognition. *Neurocomputing*. 2015;**155**:219-224
- [47] Chen X, Wang J, Sun D and Liang J 2008 A novel hybrid evolutionary algorithm based on pso and AFSA for feedforward neural network training
- [48] Zhang Y, Wu L. Crop classification by forward neural network with adaptive chaotic particle swarm optimization. *Sensors*. 2011;**11**:4721-4743
- [49] Zhang JR, Zhang J, Lok TM, Lyu MR. *Applied Mathematics and Computation*. 2007;**185**:1026-1037
- [50] Yaghini M, Khoshraftar MM, Fallahi M. Hiopga: A new hybrid meta- heuristic algorithm to train feedforward neural networks for prediction
- [51] Kartheeswaran S, Durairaj DDC. *Informatics in Medicine Unlocked*. 2017; **8**:21-31
- [52] Zhang C, Shao H. An Ann's Evolved by a New Evolutionary System and Its Application. Vol. 4. *IEEE*; 2000. pp. 3562-3563
- [53] Carvalho M, Ludermitr TB. Particle swarm optimization of feed- forward neural networks with weight decay. In: 2006 Sixth International Conference on Hybrid Intelligent Systems (HIS'06). *IEEE*; 2006. p. 5
- [54] Xue B, Ma X, Gu J, Li Y. An Improved Extreme Learning Machine Based on Variable-length Particle Swarm Optimization. *IEEE*; 2013. pp. 1030-1035
- [55] Carvalho M, Ludermitr TB. Particle swarm optimization of neural network architectures and weights
- [56] Kiranyaz S, Ince T, Yildirim A, Gabbouj M. *Neural Networks*. 2009;**22**: 1448-1462
- [57] Das G, Pattnaik PK, Padhy SK. *Expert Systems with Applications*. 2014; **41**:3491-3496
- [58] Asadnia M, Chua LH, Qin X, Talei A. *Journal of Hydrologic Engineering*. 2014; **19**:1320-1329
- [59] Zhang YD, Wang S, Wu L. *Progress in Electromagnetics Research*. 2010;**109**: 325-343
- [60] Chunkai Z, Yu L, Huihe S. A New Evolved Artificial Neural Network and

Its Application. Vol. vol 2. IEEE; 2000.
pp. 1065-1068

Science and Technology. 2019;33:
5137-5145

[61] Wang H, Wang Y, Wang Y. Expert Systems with Applications. 2013;40: 418-428

[72] Chen B, Zhang H, Li M. Prediction of pK (a) values of neutral and alkaline drugs with particle swarm optimization algorithm and artificial neural network. Neural Computing and Applications. 2019

[62] Qi C, Fourie A, Chen Q. Construction and Building Materials. 2018;159:473-478

[63] Yu J, Xi L, Wang S. An improved particle swarm optimization for evolving feedforward artificial neural networks. Neural Processing Letters. 2007;26: 217-231

[64] Juang CF. IEEE Transactions on Systems, Man, and Cybernetics (Part B) (Cybernetics). 2004;34:997-1006

[65] Meissner M, Schmuker M, Schneider G. BMC Bioinformatics. 2006; 7:125

[66] Carvalho M, Ludermir TB. An Analysis of PSO Hybrid Algorithms for Feed-forward Neural Networks Training. IEEE; 2006. pp. 6-11

[67] Zhao W. BP Neural Network based on PSO Algorithm for Temperature Characteristics of Gas Nanosensor JCP. Vol. 72012. pp. 2318-2323

[68] Armaghani DJ, Hajihassani M, Mohamad ET, Marto A, Noorani S. Arabian Journal of Geosciences. 2014;7: 5383-5396

[69] Pradeepkumar D, Ravi V. Applied Soft Computing. 2017;58:35-52

[70] Yang J, Wang L, Jiang Q. Ford vehicle identification via shallow neural network trained by particle swarm optimization

[71] Kong Y, Abdullah S, Schramm D, Omar M, Haris S. Journal of Mechanical