



ISSN: 0975-766X  
CODEN: IJPTFI  
Research Article

Available Online through  
[www.ijptonline.com](http://www.ijptonline.com)

## A REVIEW OF HEURISTIC GLOBAL OPTIMIZATION BASED ARTIFICIAL NEURAL NETWORK TRAINING APPROACHES

Geraldine Bessie Amali.D<sup>+</sup>\*, Dinakaran.M<sup>++</sup>

<sup>+</sup>Assistant Professor, School of Computer Science and Engineering, VIT University, Vellore.

<sup>++</sup>Associate Professor, School of Information Technology, VIT University, Vellore.

Email: geraldine.mithra@gmail.com

Received on: 18.10.2016

Accepted on: 11.11.2016

### Abstract:

Artificial Neural Networks have earned popularity in recent years because of their ability to approximate nonlinear functions. Training a neural network involves minimizing the mean square error between the target and network output. The error surface is nonconvex and highly multimodal. Finding the minimum of a multimodal function is a NP complete problem and cannot be solved completely. Thus application of heuristic global optimization algorithms that computes a good global minimum to neural network training is of interest. This paper reviews the various heuristic global optimization algorithms used for training feedforward neural networks and recurrent neural networks. The training algorithms are compared in terms of the learning rate, convergence speed and accuracy of the output produced by the neural network. The paper concludes by suggesting directions for novel ANN training algorithms based on recent advances in global optimization.

**Keywords:** Artificial Neural Networks, Feedforward Neural Networks, Recurrent Neural Networks, Optimization Algorithms, Training Algorithms.

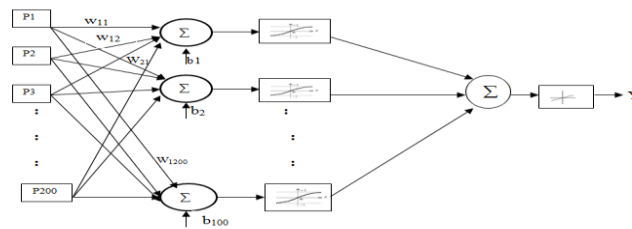
### Introduction:

Artificial Neural Network (ANN) is a mathematical model of the biological nervous system. A neural network consists of sets of adaptive weights, i.e. numerical parameters that are tuned by a learning algorithm, and are capable of approximating non-linear functions<sup>18</sup>. It can also be used to solve various problems including pattern recognition, classification, and function approximation. A neural network is an inter connection of neurons arranged in layers. It consists of an input layer, an output layer and zero or more hidden layers. The input layers and the subsequent layers are connected by links with weights. The strength of the link depends on the weight. The performance of the neural network lies in the number of neurons in each layer and also the weights. The goal of any training algorithm used by

the neural network is to determine the weights of the links so that it reduces the error between the output produced by the neural network and the ideal output.

**Feed forward Neural Networks:**

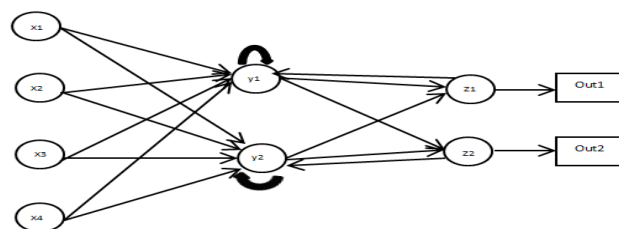
In typical feedforward architecture the neurons are arranged in cascaded layers where all the neurons in one layer are connected to all the neurons in the adjacent layer<sup>18</sup>. However, the neurons are not connected to other neurons in the same layer. The branching between the neurons is unidirectional therefore information can pass only in one direction and there is no feedback. The output of the first layer is presented as input to the second layer. The branches have weights associated with them which can be adjusted by the learning algorithm. A multi-layer perceptron with adequate number of neurons and one hidden layer can approximate a nonlinear function<sup>18</sup>. Figure 1 shows the architecture of a feedforward neural network with one hidden layer.



**Fig.1. A feedforward neural network with one hidden layer.**

**Recurrent Neural Networks:**

In feedforward neural networks the weights associated with the branches are fixed and therefore the state of the neuron solely depends upon the input given to the neuron<sup>18</sup>. This is a static model because it does not depend on the past state of the neurons. Recurrent neural networks on the other hand utilize feedback. Unlike feedforward neural networks this architecture uses nonlinear processing units, is fully connected and is fault tolerant<sup>18</sup>. Due to their dynamic nature and temporal behaviour recurrent neural networks are used in high intelligent systems with applications in symbolic reasoning. Figure 2 shows the architecture of recurrent neural network with four neurons in the input layer, two in the hidden layer and two output neurons.



**Fig. 2. A recurrent neural network with one hidden layer.**

## Learning in Artificial Neural Networks:

Learning also called as training a neural net is a very crucial process and is done by systematically adjusting the connection weights. The ANN learning can be one of the following types namely supervised, unsupervised or reinforcement learning<sup>21</sup>. In supervised learning the neural net is trained by input and corresponding pairs. The connection weights are adjusted in such a way so as to reduce the error. Unsupervised learning however trains the neural net based on the correlation of data. Reinforcement learning is a special type of learning which employs feedback from the environment. The neural net is trained with positive rewards and negative rewards based on the performance<sup>21</sup>.

## Global Optimization:

Mathematical optimization is minimization or maximization of a real valued function by selecting the best solution from an available set of feasible solutions<sup>32</sup>. In the case of minimizing a real valued function also called as the cost function the goal is to determine the value of the input which minimizes the function the most. An optimization problem can be formulated as follows minimize  $f(x)$  Subject to  $x \in \Omega$  Where  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is the function that is to be minimized and  $n$  is the dimensionality of the vector  $x$ . The set  $\Omega$  is a subset of  $\mathbb{R}^n$ . When the set  $\Omega$  is the whole of  $\mathbb{R}^n$  the optimization is said to be unconstrained.

In this paper an unconstrained optimization problem is considered. The optimization problem above can be viewed as finding the vector  $x^*$  from the domain  $\Omega$  such that that  $f(x^*) \leq f(x)$ . A point  $x^* \in \Omega$  is a local minimum of fover  $\Omega$  if there exists  $\varepsilon > 0$  such that  $f(x^*) \leq f(x)$  for all  $x \in \Omega \setminus \{x^*\}$  and  $\|x - x^*\| < \varepsilon$ . On the other hand a point A point  $x^* \in \Omega$  is a global minimum of fover  $\Omega$  if  $f(x^*) \leq f(x)$  for all  $x \in \Omega$ . In general, it is only practical to compute a good local minimum as the problem of nonlinear global optimization is NP-complete. Optimization problems can either be convex or non-convex. A Convex function has one local minimum which is also the global minimum. Highly efficient algorithms like the interior point algorithm exist to compute the global minimum for convex optimization problems<sup>32</sup>.

Non-convex functions on the other hand may contain numerous local minima which makes the problem of locating the global minimum either very difficult or impossible. In many instances optimization algorithms get stuck in local minimum without converging to the global minimum. Thus only good local minimum can be computed in general. However a majority of important problems in engineering like filter design involve non-convex global optimization. Hence development of heuristic random search algorithms inspired by biology like PSO and GA which can compute

near optimal solutions are of interest<sup>20</sup>. The rest of the paper is organized as follows. The following section presents the evolution of the training algorithms and compares the various optimization algorithms used in training feedforward neural networks followed by a review of the algorithms used in training recurrent neural networks. Conclusion is presented in the last section.

### **Review of optimization algorithms used for training feedforward neural networks:**

Training ANNs using back propagation algorithm had limitations in terms of overfitting, increase in learning time with the size of the training data, and most importantly the risk of getting stuck in the flat regions of the search space thereby converging to a local minimum and not finding the global minimum<sup>1</sup>. Therefore, biologically inspired optimization algorithms like Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) were used to train ANNs.

GA is inspired by natural evolution and adopts the principles of selection, crossover and mutation<sup>24</sup>. It is stochastic and derivative free and therefore can be applied to both continuous and discrete optimization problems. In<sup>2</sup> the authors have used GA with crossover to calculate the weights of the feedforward neural network (FNN)<sup>23</sup>. He has demonstrated in his work<sup>3</sup> that GA outperforms back propagation algorithm. Training neural networks with distributed GA reinforced by perceptron learning rule was proposed and applied by Olikier et al in<sup>4</sup>. A version of GA known as soft algorithm is combined with backpropagation and soft-bp is applied to train ANNs in<sup>19</sup> by Adawy et.al. This algorithm obtains a good weight vector thereby reducing the error of the output. The parallel version of GA has been used in<sup>22</sup> for time series prediction.

PSO is inspired by the swarm behaviour of flock of birds or school of fishes. It was proposed by Kennedy and Eberhart in<sup>5</sup> and has been used for training neural networks by Gudise and Venayagamoorthy in<sup>6</sup>. The performance of the PSO algorithm is compared with the backpropagation algorithm by training the neural network to learn a nonlinear function. It was found the PSO was faster between the two algorithms to learn the nonlinear function as it required less number of computations than BP to attain the same error goal<sup>6</sup>. PSO algorithm itself has certain limits in terms of convergence, precision and parameter selection. It was slower during the final stages of evolution and had lower precision.

Therefore, Chen et al in<sup>31</sup> proposed an algorithm called Artificial Fish Swarm Algorithm (AFSA) - PSO parallel hybrid evolutionary algorithm (APPHE) for training FNNs. This algorithm divides the PSO population into two sub populations. PSO is executed in one sub population and AFSA in the other in parallel. The best solution of both the

sub population is given back to the swarm and PSO is now executed in both sub populations. The algorithm terminates when a termination criterion is satisfied. The authors tested the performance of the proposed APPHE algorithm with the Levenberg- Marquardt Back propagation (LMBP) algorithm by training the neural network in iris data classification. The neural network trained using the APPHE algorithm did better than LMBP in terms of faster convergence to the global minimum and accuracy of the result. Hybrid algorithms that combine global optimization algorithm and local search algorithms were used to train ANNs. Hybrid Artificial Bee Colony Algorithm that combines the Artificial Bee Colony algorithm (ABC) and the Levenberg- Marquardt(LM) is used for training ANN in <sup>7</sup>. ABC is a global optimization algorithm and finds the global minimum whereas LM is used to exploit the local minimum. Therefore a hybrid algorithm that combines the exploration ability of the ABC and exploitation ability of the LM has been proposed by Ozturk in <sup>7</sup>. The hybrid algorithm performs better than the algorithms by themselves. A modified LM algorithm which addressed the demand of memory for large jacobians and the need for inverting the large matrices was proposed by Wilamowski and Chen in <sup>11</sup>. Their proposed algorithm used a new performing index which reduces the size of the matrix that is to be inverted thereby increasing the computation speed. Simulated annealing is a global search heuristic that is inspired by annealing in metallurgy. Metallurgy is a physical process of heating metal to very high temperatures and then cooling it very slowly. This helps remove the defects in the crystals formed. Simulated annealing is combined with local gradient search algorithm (Rprop) in <sup>8</sup> and in <sup>9</sup> with tabu search to train ANNs. A composite squared error algorithm was proposed and applied to train ANNs by Gonzaga et al. in <sup>11</sup>. In this algorithm the first part of training uses the linear error while the second part uses the nonlinear. By doing so the algorithm escapes the suboptimal solutions and converges to the optimal solution faster than the backpropagation algorithm.

A novel ant algorithm proposed Dorigo in <sup>28</sup> is used by Li and Liu in <sup>27</sup> and applied to train feedforward neural net for call admission control. Ant colony optimization (ACO) is a global optimization algorithm that is inspired by the swarm behaviour of ants following a path seeking food from their colonies <sup>28</sup>. The ants have to perform two tasks. First they have to select the path which they want to follow and secondly adjust their pheromone level along the chosen path. A version of the aforementioned ACO is used for training a neural net in <sup>27</sup>. This ACO trained neural network performed well when compared to BP however its performance degrades with the increase in the number of inputs due to the communication overhead.

Quantum Computing with optimization algorithms started to evolve. Quantum inspired GA<sup>13</sup> a quantum inspired parallel GA was proposed in<sup>14</sup>. A Quantum Shuffled Frog Leaping Algorithm (QSFLA) was proposed and used for training ANNs by Liu and Zhang in<sup>15</sup>. This algorithm efficiently solved continuous optimization problem in high dimensional space and did better than the BP algorithm in terms of convergence and accuracy.

### **Review of optimization algorithms used for training recurrent artificial neural networks:**

Breeding swarm algorithm is a hybrid of GA and PSO. This was proposed by Matthew et al in<sup>10</sup> and used it to training of ANNs. Their algorithm uses the classical PSO formula for updating the velocity and the positions of the particles, and uses the selection, mutation and crossover principles from GA. In addition the authors have also introduced a parameter called the breeding parameter, which determines the population size that should undergo breeding. Since breeding swarms algorithm was a general population based algorithm, when it was used to train recurrent neural networks it was found that the algorithm was able to scale better. In<sup>16</sup> a hybrid Bayesian learning method which combines Markov chain Monte Carlo methods with fuzzy membership functions and GA is used by Kocadagli for training Bayesian neural networks. The author has addressed the problems of complexity in choosing the parameters of the model, the training time associated with the Bayesian neural networks. He argues that the proposed hybrid model can overcome the problems faced with normal training algorithms. A hybrid model that combines the gradient descent and metaheuristics is proposed and used in<sup>17</sup>.

An improved version of PSO with time varying parameter and constriction helps in improving the search ability and convergence. In order to prevent overfitting a cross validation method is also included in the algorithm. A variant of PSO called the modified binary Particle Swarm Optimization (MPSO) was proposed by Eberhart for binary problems<sup>21</sup>. This version of PSO was used to train recurrent neural networks in<sup>29</sup> for decoding of 1/n rate convolutional codes. This approach provided low latency and converged to a global minimum thereby making it more practicable.

### **Discussion:**

A review of the literature shows that optimization algorithms approximate nonlinear functions and provide near accurate solutions. Biologically inspired optimization algorithms like GA, PSO, ACO, ABC,AFSA are stochastic in nature. This helps these algorithms explore the search space and escape from local minima. Backpropagation however uses gradient descent and might get stuck in the local minimum. Literature shows that variants of the optimization algorithms especially hybrid algorithms that combine global and local search heuristic perform better

than the algorithms by themselves. These hybrid algorithms performed better than the algorithms by themselves in terms of the learning rate, accuracy and convergence. Because these hybrid algorithms combine the exploratory ability of the global optimization algorithms and the exploitation ability of the local search algorithms they provide better results. Literature also confirms that hybrid algorithms outperform the algorithms by themselves in training recurrent neural networks. In recent years novel heuristic global optimization algorithms that outperform current state of the art optimization algorithms have been proposed<sup>32</sup>. In the approach proposed in <sup>32</sup>alternating cycles of exploration and exploitation are used to achieve a compromise between exploration of new solutions and exploitation of existing solutions and avoid premature convergence to local minima. Gradient based algorithms like Backpropagation have a tendency to get stuck in local minima leading to poor performance of the ANN. Thus application of novel heuristic global optimization algorithms like the Galactic Swarm Optimization (GSO) algorithm<sup>32</sup> to ANN training is of interest. Comparison of different heuristic global optimization algorithms such as<sup>32</sup> and <sup>33</sup> on benchmark ANN training problems can be considered for future work. Table 1 shows the comparison of the various algorithms used for training neural networks.

**Table-1. Comparison of global optimization algorithms used for training neural networks.**

<b>ANN Training Algorithm</b>	<b>Successes</b>	<b>Challenges</b>
Genetic Algorithm	Explores large and complex search space	Slower convergence
Particle Swarm Optimization	Fewer number of computation required to learn	Slower convergence
Hybrid Artificial Bee Colony	Accuracy of results	Not practicable for high dimensional classification problems
Hybrid PSO with stop criteria	Faster convergence	Less exploration of the search space
Ant Colony Optimization	Accuracy of result	Performance degrades with the increase in the number of inputs.
Artificial Fish	Faster	Limited

---

Swarm Optimization	convergence	applications
Hybrid Simulated Annealing	Accuracy of results	Overfitting

---

### **Conclusion:**

This paper reviews the global optimization algorithms used for training feedforward and recurrent neural networks. ANNS and most of the global optimization algorithms are biologically inspired and they borrow ideas from the social behaviour and biological structure of the individuals. Thereby it can be positively stated that training neural nets with biologically inspired optimization algorithms will provide a more complete learning. A review of the literature proves that due to the stochastic nature of these algorithms, hybrid algorithms which combine the global and local optimization algorithms outperform the algorithms by themselves in terms of faster convergence and accuracy of the output. In recent years novel heuristic global optimization algorithms that outperform current state of the art optimization algorithms have been proposed. These algorithms can be considered for ANN training in the future.

### **References:**

1. M. Gori and A. Tesi, "On the problem of local minima in backpropagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.14, pp.76-86, Jan 1992.
2. D. Whiteley, "Applying Genetic Algorithms to Neural Networks Learning", *Proceedings of 7th Conference of the Society of Artificial Intelligence and Simulation of Behavior*, Sussex, England: Pitman Publishing, pp. 137-144, 1989.
3. D. Whiteley, T. Starkweather and C. Bogart, "Genetic Algorithms and neural Networks: Optimizing Connections and Connectivity", *Parallel Computing*, vol. 14, pp. 347-361, 1990.
4. Olikar, S, et al., "Design architectures and training of neural networks with a distributed genetic algorithm", *IEEE International Conference on Neural Networks*, vol.1, pp.199-202, 1993.
5. James Kennedy and R. Eberhart, "Particle swarm optimization", *IEEE International Conf. on Neural Networks*, Perth, Australia. Vol. IV, pp. 1942-1948, 1995.
6. Gudise and Venayagamoorthy G.K., "Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks," *Proceedings of the 2003 IEEE in Swarm Intelligence Symposium*, 2003, pp.110-117, 24-26 April 2003.



7. Ozturk and Karaboga, "Hybrid Artificial Bee Colony algorithm for neural network training," 2011 IEEE Congress on Evolutionary Computation (CEC), pp.84-88, 5-8 June 2011.
8. N. Treadgold and T. Gedeon, "Simulated annealing and weight decay in adaptive learning: the sarprop algorithm," IEEE Transactions on Neural Networks, vol. 9, pp. 662–668, 1998.
9. T. Ludermir et al., "An optimization methodology for neural network weights and architectures," IEEE Transactions on Neural Networks, vol. 17(5), pp. 1452–1460, 2006
10. Settles M et al., "Breeding swarms: a new approach to recurrent neural network training", In Proceedings of the 7th annual conference on Genetic and evolutionary computation ACM , pp. 185-192, Jun 25 2005.
11. Gonzaga D et al., "Composite squared-error algorithm for training feedforward neural networks", Advances in Digital Filtering and Signal Processing IEEE Symposium, pp. 116-120, Jun 5 1998.
12. Wilamowski et al., "Efficient algorithm for training neural networks with one hidden layer". In Proc. IJCNN 1999 Jul 10, vol. 3, pp. 1725-728.
13. Narayanan A and Moore M., "Quantum-inspired genetic algorithms", In Evolutionary Computation, 1996., Proceedings of IEEE International Conference, 1996 May 20, pp. 61-66.
14. Han KH et al., "Parallel quantum-inspired genetic algorithm for combinatorial optimization problem", In Evolutionary Computation, 2001 Proceedings of the 2001 Congress, 2001, vol. 2, pp. 1422-1429.
15. Liu L and Zhang Q, "Training and application of process neural network based on quantum shuffled frog leaping algorithm", In Computer Science and Network Technology (ICCSNT), 2013 3rd International Conference, 2013 Oct 12 pp. 829-833.
16. Kocadağlı O, "A novel hybrid learning algorithm for full Bayesian approach of artificial neural networks", Applied Soft Computing, 2015, pp. 52-65.
17. Yaghini M et al., "A hybrid algorithm for artificial neural network training", Engineering Applications of Artificial Intelligence, 2013 Jan 31, vol. 26(1), pp. 293-301.
18. Ham FM, Kostanic I "Principles of neurocomputing for science and engineering", McGraw-Hill Higher Education 2000 Sep 1.
19. El Adawy et al., "A SOFT-backpropagation algorithm for training neural networks", In Radio Science Conference, 2002 (NRSC 2002). Proceedings of the Nineteenth National 2002 , pp. 397-404.

20. Noel MM, "A new gradient based particle swarm optimization algorithm for accurate computation of global minimum", *Applied Soft Computing.*, 2012 Jan 31, vol.12, pp.353-9.
21. Kennedy J and Eberhart RC, "A discrete binary version of the particle swarm algorithm",. In *Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation.*, 1997 IEEE International Conference, 1997, vol. 5, pp. 4104-4108.
22. Mendivil SG et al., "Optimization of artificial neural network architectures for time series prediction using parallel genetic algorithms", In *Soft Computing for Hybrid Intelligent System*, 2008 Jan 1, pp. 387-399.
23. Zi-wu RE and Ye SA, "Improvement of Real-valued Genetic Algorithm and Performance Study [J]", *Acta Electronica Sinica.*, 2007 vol. 2, 017.
24. Guo Z and Uhrig RE, "Using genetic algorithms to select inputs for neural networks", In *Combinations of Genetic Algorithms and Neural Networks, 1992.*, COGANN-92. International Workshop on 1992 Jun 6, pp. 223-234.
25. Kitano H, "Designing neural networks using genetic algorithms with graph generation system", *Complex Systems Journal*, 1990, vol. 4, pp. 461-76.
26. Angeline PJ et al., "An evolutionary algorithm that constructs recurrent neural networks", *IEEE transactions on Neural Networks*, 1994, vol. 1, pp. 54-65.
27. Li S and Liu Z, "A general CAC approach using novel ant algorithm training based neural network", In *Neural Networks, 1999. IJCNN'99. International Joint Conference on 1999*, vol. 3, pp. 1885-1888.
28. Dorigo M and Gambardella LM, "Ant colony system: a cooperative learning approach to the traveling salesman problem", *IEEE Transactions on Evolutionary Computation*, 1997 vol. 1, pp. 53-66.
29. Asvadi R and Ahmadian M "Modified binary PSO training of recurrent neural network for 1/n rate convolutional decoders", In *Performance Evaluation of Computer and Telecommunication Systems, 2008. SPECTS 2008. International Symposium*, 2008 Jun 16, pp. 30-35.
30. Luitel B and Venayagamoorthy GK., "A PSO with quantum infusion algorithm for training simultaneous recurrent neural networks", In *Neural Networks, 2009. IJCNN 2009. International Joint Conference*, 2009 Jun 14, pp. 1923-1930.