ANN Training: A Survey of Classical & Soft Computing Approaches

Ashima Kalra¹, Shakti Kumar² and Sukhbir Singh Walia³

ABSTRACT

Neural networks are inspired from the structure of brain and have the capability to solve various complex classification problems. Various learning algorithms are available in the literature to train the neural networks. An extensive survey has been carried out for the available approaches. It has been observed that no single learning approach is suitable for all types of search and optimization problems. So for every new kind of problem new algorithm may be needed. The different available traditional methods of learning have a drawback of poor convergence speed and getting trapped to local optimum. So to avoid this, soft computing based approaches are to be preferred for global optimization. This paper presents the classical as well as soft computing based search and optimization algorithms in existing literature for training neural networks and the qualitative comparison among them will enable the researchers to develop new algorithms either hybrid or stand alone for ANN model identification.

Index terms: Back propagation, artificial neural network, learning algorithm, soft computing.

I. INTRODUCTION

An Artificial neural network (ANN) is a massively parallel, distributed processing system made up of simple processing elements which has the natural ability for storing experiential knowledge and making it available for use when required. ANN is an information processing paradigm that is inspired by the structure of brain. Neural networks has a special ability to derive something meaningful from any complicated data which can be further used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer. ANN’s are typically applied for pattern classification [1] [2] and pattern recognition [3] [4]. They have been successfully used for stock market predictions [5], wear and manufacturing processes [6], speech recognition [7], business applications [8], control applications [9], time series modeling and estimation applications [10], medical diagnosis [11] [12] [13], aeronautics [14] etc.

Gallant S. [15] was the first to describe a system combining the domain expert knowledge with neural training. This connectionist expert system (CES) consists of an expert system implemented throughout a multi layer perceptron. Various authors in papers [16] [17] demonstrated the structural learning with forgetting for training neural networks. Melanie Hilario et.al [18] presented a strategy for Modular Integration of Connectionist and Symbolic Processing in Knowledge based systems. Authors in papers [19] [20] demonstrated the rules generation from trained network using fast pruning. A rich literature survey has been found on the extraction of rules from the trained neural networks in papers [21] [22]. Various authors in their papers [23]-[42] presented the various different techniques for extracting rules from trained neural networks. Thrun S. [43] presented extraction of rules from artificial neural networks with distributed representations. Different types of architectures and corresponding learning algorithms can be found in literature. Some of the widely used architectures along with their learning algorithms and applications are
given in Table I. There are basically two types of architectures: Feed forward networks, Feedback/Recurrent networks. Feed forward networks can further be classified as single layer perceptron, multi-layer perceptron (MLP) and radial basis function networks (RBFN). On the other hand recurrent/feedback type networks consist of networks like competitive networks, Kohenen’s self-organizing Maps (SOM), Hopfield networks (HN) and adaptive resonant theory (ART) models.

Learning is the backbone of any kind of neural network design. Learning is a process by which the free parameters of a neural network are adapted through a process of simulation by the environment in which the network is embedded. There are three types of learning algorithms: Supervised learning, unsupervised learning and hybrid learning. Supervised learning is the learning with a teacher. In this type of learning, a kind of target or desired output signal is present with which the computed output signal is compared to compute the error signal. Paradigms of supervised learning include Perceptron learning algorithm, error correction learning, stochastic learning, Adaline, Medialine Boltzmann learning algorithm, learning vector quantization etc. An unsupervised learning is a kind of learning without teacher i.e no desired output signal is available. This type of learning is based on the concept of using self organization to compute the output. Paradigms of unsupervised learning are Hebbian learning and competitive learning, Kohenen learning (SOM), ART1, ART2. Hybrid learning is the combination of two types of learning mentioned above. Paradigms of hybrid learning are radial basis. The essence of a learning algorithm is the learning rule, i.e., a weight-updating rule which determines how connection weights are changed. Different types of learning laws are used to update the synaptic weights like delta law, perceptron law, instar learning, outstar law etc. The training process can be on-line or batch training. In on line training, the weights are adjusted after the processing of a randomly selected training pattern while in batch training; the weights are adjusted after processing.

Since ANN based systems are highly complex and non-linear systems, we divide the learning algorithms in the following two classes: a) Classical Learning algorithms b) Soft Computing based algorithms. We define soft computing based algorithms as the one’s that uses approximate reasoning in ANN training. These algorithms include fuzzy logic based approaches, swarm based approaches and the other approaches based upon certain other nature inspired computing approaches.

Section II of this paper presents a quick glance on classical approaches used for ANN training available in literature. Section III reviews soft computing based approaches found in the literature. Section IV of this paper compares the two approaches and Section V concludes the paper.

II. CLASSICAL LEARNING APPROACHES

The main aim of training is to minimize the error function. For the ANN modeling, it is required to decide the architecture first with consideration to the number of hidden layers and hidden neurons as well. Once the Architecture is decided then the next step is weights learning i.e to update the connection weights. Feed-forward neural networks are typically applied for pattern classification and supervised learning is a convenient way to train them.

These approaches can be classified into two classes depending on their basic strategy. First order methods include the computation of error gradient through EBP along with simple modifications of the algorithm like the use of a momentum term or the adaptive step methods using a variable step size which is heuristically adapted to the error surface. There may either be one global learning rate for all weights or individual learning rates for each single weight. Second order methods compute a quadratic approximation of the error surface which is then minimized in order to reach the minimum of the actual error function iteratively.

Literature is rich with training & design approaches for ANN. There are many literatures address in the training and designing of ANN system. Various classical as well as soft computing based search and optimization methods are available in literature for the training of neural networks. The classical training
approaches available in literature are shown in Table II. Werbos, [44] introduced EBP for the first time as the roots of propagation. Rumelhart [45] further elaborated it as the learning representations of NN by back propagating errors. This algorithm includes different versions like standard or incremental back propagation (IBP), Freeman and Skappura [46] in which the network weights are updated after presenting each pattern from the learning data set, rather than once per iteration and batch back propagation (BBP), Hagan et.al [47] in which the network weights update takes place once per iteration, while all learning data pattern are processed through the network and quick propagation (QP), Fahlmann [48] which is a heuristic modification of the back propagation algorithm. To improve the convergence speed of EBP, Rumelhart extended his work by introducing the momentum term [49] [50] [51]. The various adaptive learning methods like Delta –Bar –Delta (DBD), R A Jacobs, Tollenaere [52] [54], Extended Delta Bar Delta (EDBD), Minai and Williams [56], SSAB (Super-SAB), Tollenaere [57], resilient propagation(RPROP), Riedmiller and Braun [58] and the generalized no-decrease adaptive method (GNDAM), R. Allard, J. Faubert [59] have been developed to self adjust the LR(learning rate) and to just get rid of the slow convergence problem thereby obtaining the optimal solution. The hybrid learning schemes have been proposed to incorporate second derivative related information in Rprop, such as the QRprop [60], which is the combination of RPROP with the one dimensional secant steps of quick algorithm and the Diagonal Estimation Rprop–DERprop [61], which directly computes the diagonal elements of the Hessian matrix. Also approaches inspired from global optimization theory have been developed to equip Rprop with annealing strategies, such as the Simulated Annealing Rprop–SARprop and the Restart mode Simulated Annealing Rprop i.e ReSARprop [61] in order to escape from shallow local minima. Another improvement was proposed as Improved Rprop (iRprop) algorithm, C Igel and M Husken [62] [63] [64] which applies a backtracking strategy (i.e. it decides whether to take back a step along a weight direction or not by means of a heuristic), has shown improved convergence speed when compared against existing Rprop variants, as well as other training methods. Aristoklis D A [65] [66] proposed another algorithm G Rprop which was modification over iRprop and exhibited better convergence speed and stability than Rprop and iRprop.

The other two second order approaches: conjugate gradient and Quasi Newton have been reported as the most successfully applied to the training of feed forward neural networks amongst all those using second order information. Conjugate gradient method was first initiated by Hestenes and Stiefel(1952) for linear functions and then based on this work, Fletcher and Reeves(1964) further extended it as conjugate algorithm for non linear functions. Afterwards Beale, 1972 proposed conjugate gradient method with the provision of restarting direction procedure. Navon and Legler in paper [67] has presented a review of various conjugate gradient algorithms for large scale minimization where they covered almost four types of Conjugate Gradient algorithms and compared their advantages as well as shortcomings. Johansson et al. in paper [68] proposed a conjugate gradient with line search (CGL) method where a step size is approximated with line search by avoiding the calculation of Hessian Matrix. M F Moller, 1991[69] proposed a Scaled conjugate gradient (SCG) method for fast supervised learning and was found to be more faster than CGL and BP. A M S Barreto, C.W. Anderson [70] proposed a restricted Gradient descent (RGD) algorithm for training local RBF networks in the context of reinforcement learning. BFGS Quasi Newton optimization approach with limited memory was first proposed by R. Battiti and F. Masulli [71]. There were basically two update approaches for quasi Newton’s method – BFGS (Broyden, Fletcher, Goldfarb and Shanno) update, Battiti [71][72] and the DFP(named for Davidon, Fletcher and Powell) update, Waterous [73]. J E Dennis and J J More in paper [74] presented the survey with justification of use of Quasi Newton methods over Newton method for general and gradient non linear systems and proved it more computational efficient than Newton method.

A Likas, A Stafylopatis in paper [75] presented the training of random neural network using Quasi newton methods. For fast and efficient training, second order learning algorithms have to be used. The most effective method is Levenberg Marquardt (LM) algorithm, which is a derivative of the Newton method. It is a good combination of Newton’s method and steepest descent The Levenberg-Marquardt algorithm
(LM), M. T. Hagan and M. B. Menhaj [76] [77], is also used as another algorithm to increase the convergence speed. But the LM algorithm becomes impractical for large sized neural networks so another modification of LM i.e. TREAT algorithm, Y Chan [78] is then preferred for large sized networks. So Traditional training algorithms, such as EBP and LM have been successfully applied to train neural networks by some authors in papers [79]-[83]. But still these algorithms require more memory storage, computation, and there is always a risk of getting trapped in local minimum, as they are not derivative free. So this could be a biggest reason to move towards another nature inspired or soft computing based search and optimization approaches.

III. SOFT COMPUTING OPTIMIZATION APPROACHES

Learning most often is modelled as an optimization process wherein the error is minimized as the learning takes place. The heuristic approach like nature inspired or soft computing based algorithms are much superior in solving complex optimization problems where traditional or classical problem solving methods fail. These approaches are broadly classified into four - Evolutionary computing (EC), Swarm intelligence (SI), Bio inspired Non-SI and physics or chemistry based algorithms. Various soft computing based Search and optimizations based approaches available in literature and are shown in Table 3 and Table 4. Evolutionary Computing is based on the biological evolution process in nature. Swarm intelligence based algorithms are based upon collective social behavior of organisms. Bio inspired Non SI optimization algorithms are bio inspired or ecology based but are not inspired by the cooperative behavior of any organisms. Physics or chemistry based algorithms are actually inspired by certain physical or chemistry laws like electric charges, gravity, theory of universe etc. Literature is also rich for soft computing based search and optimization approaches. ANN learning model, based on EA’s, Yao [82] [83], based upon GA’s, Holland [87][88], based upon GP, John Koza [89] [90], based upon BBO, Dan Simon 2008 [91], based upon ACO, Dorigo and Gambardella [93] [94], based upon PSO, Eberhart and Kenedy [95], based upon DE, Storn and Price [96], based upon FA, Yang [102] ,based upon ABC, Karaboga and Basturk [103]-[104] , based upon IWO, Mehrabian and Lucas [105][133], based on AIS, Dasgupta [107] [108], based upon EEIM Birbil & Fang [109]-[111], based upon BFOA, Passino [112]-[114], based upon GSA, Rashiedi et.al [115]-[116], based upon FSA, Li Xiao-lei et.al [117] and based upon BB-BC, O K. Erol, I Eksin [118][119] are available in literature. These soft computing based approaches have been successfully applied for ANN model identification. S. Kumar et.al [120] presented ANN model identification for rapid battery Charger. Parallel BB-BC algorithm was a multi-population algorithm proposed by S Kumar et.al [122]. S Kumar et.al [123] further presented overall rating and evaluation of institutions of higher learning using BBBC and parallel BB BC based on fuzzy model identification. Similar kinds of techniques are available for fuzzy model identification also. Fusion of Artificial Neural Networks (ANN) and Fuzzy Inference Systems (FIS) have attracted the growing interest of researchers in various scientific and engineering areas due to the growing need of adaptive intelligent systems to solve the real world problems. Neuro Fuzzy (NF) computing is a popular framework for solving complex problems. Various authors discussed the design of neuro fuzzy controllers [124]-[128] and adaptive neuro fuzzy systems [129][130].

These soft computing based approaches have been successfully applied for ANN model identification. S. Kumar et.al [216] presented ANN model identification for rapid battery Charger. Parallel BB-BC algorithm was a multi-population algorithm and first proposed by S Kumar et.al [218]. S Kumar et.al [219] proposed overall rating and evaluation of institutions of higher learning using BBBC and parallel BB BC based on fuzzy model identification. Similar kinds of techniques are available for fuzzy model identification also. Fusion of Artificial Neural Networks (ANN) and Fuzzy Inference Systems (FIS) have attracted the growing interest of researchers in various scientific and engineering areas due to the growing need of adaptive intelligent systems to solve the real world problems. Neuro Fuzzy (NF) computing is a popular framework for solving complex problems. Various authors in paper discussed the design of neuro fuzzy controller [221]-[225] and adaptive neuro fuzzy systems [226] [227].
IV. CLASSICAL VS. SOFT COMPUTING APPROACHES

Various authors have compared the soft computing based approaches with classical learning approaches as well as hybrid techniques for NN learning. Various comparisons have been made in literature among classical learning approaches too. Marcus Pfister et.al [228] also compared five algorithms so as to speed up the backpropagation i.e gradient reuse algorithm, DBD, Extended DBD, dynamic adaptation, quick prop and extended quick prop for five different benchmark problems. This paper also concluded that a learning algorithm that may be proved very faster for one problem, may fail in another case. The results show that Quickprop was the one that performed very well in all the benchmark problems while Extended QP was a big failure. For smaller problems gradient reuse algorithm is faster than BP but even much slower in case of complex problems. Remy Allard et.al in his paper [229] compared and tested the various adaptive learning methods i.e MOM(momentum), DBD(Delta Bar Delta), SSAB(Super SAB), RPROP(Resilent Prop), and GNDAM(Generalized no-decrease adaptive method) on four benchmark problems i.e parity-bit, encoder, texture detection and luminance. It clearly shows that a single AM approach cannot be proved overall best for all the tasks but the results may vary depending upon the task. MOM and DBD had a similar behaviour when they were used on the luminance, encoder and parity task. The only task of which they clearly differed was the texture task where MOM never solved the task as opposed to DBD. RPROP showed a net advantage over SSAB for the parity-bit and luminance tasks detection. A numbers of soft computing based approaches are compared against BP or LM for various benchmark problems in the literature.

G V R Sagar[230] also proposed an EA for Connection Weights in Artificial Neural Networks and compared it with BP algorithm for a X-OR benchmark problem. It was shown that EA-ANN approach gave zero mean square error than the (BP) gradient descent method datasets, and the results did not depend on the initial choice of weights. It gives the increased performance of the network in terms of accuracy. A Jagtap [231] proposed a quantum based method i.e QNN method for four well-known benchmark classification problems, namely breast cancer and iris, heart, and diabetes problems. QNN is helpful to provide a set of appropriate weights when evolving the network structure and to alleviate the noisy fitness evaluation problem.

H Kitanao [232] compared genetic algorithms(GA) with BP(back propagation) and presented and hybrid approach GA-BP which was proved faster than GA alone. The author also stated that the GA is equally efficient to the faster variants of BP in small scale networks but found less efficient in large networks. J.N.D Gupta et.al [233] compared Standard EBP with GA for optimizing artificial neural networks. The empirical results showed that the GA is superior to BP in effectiveness, ease-of-use and efficiency in training NNs. Further Zhen Guo Che et.al in his paper [234] compared BP with GA and drawn conclusion that BP is much superior and having faster training speed than GA with a drawback of having overtraining which GA doesn’t have. Paul Batchis in his paper [235] compared EA with BP using Weka Knowledge Explorer software package on three classification problems. In this, EA is found to outperform the BP method. Asha et.al [236] compared ABC with BP for classification task using four benchmark datasets availed from the UCI machine learning. This paper implemented ABC for optimizing the connection weights and concluded that ABC performance is found to be better for the four datasets as compared to BPN performance.

For achieving global optimization, various soft computing based global optimization algorithms can be used standalone or in a hybrid manner in which some local search phenomena like BP or LM is hybridized with some soft computing based global optimization algorithm. Such hybrid approaches are also compared with standalone soft computing based approaches. Enrique Alba and J. Francisco Chicano, in paper [237] proposed training Neural Networks with GA Hybrid Algorithms. They suggested the concept of weak hybridization (just the combination of two algorithms) by introducing and testing GA with the BP algorithm (GABP), and a GA with LM (GALM). J Zhang et.al in paper [238] proposed a hybrid PSO-BP algorithm where it was shown that the PSO–BP algorithm uses less CPU time to get higher training accuracy than the PSO algorithm as well as the BP algorithm. So the hybrid of PSO-BP is better than using BP or PSO alone.
Another hybrid learning approach ACO-BP was proposed by L Yan Peng et al. [239] and their results show that the ACO-BP is more effective and efficient than the standalone BP algorithm. It is also concluded that with the variation of the number of hidden nodes, the performance of ACO-BP became stable compared to ACO or BP alone. Mavrovouniotis and Yang [240][241] proposed NNACO-BP for different real-world benchmark datasets taken from the UCI repository. This paper compared the performance of ACO and ACO-BP training against: two classical learning approaches i.e. BP and LM, RCH (ACO training without pheromone consideration), a standalone ACO and a hybrid ACO i.e., ACOR and ACOR-BP, respectively, and four soft computing based approaches i.e GA, PSO, ABC and DE. The author concluded that ACO was a good choice for selecting good values for the BP. The standalone ACO training was outperformed by the standalone ACOR training whereas the hybrid ACO-BP showed superior performance, especially on large problem instances. Secondly, the performance of gradient descent methods is degraded as the problem size increases when compared with the hybrid ACO-BP training algorithm. Third, gradient descent methods usually have better performance than a standalone GA, PSO, ABC, ACO and DE training. ACO has a relatively good performance when compared with other network training algorithms for pattern classification. Huadong Chen et al. in his paper [242] proposed a hybrid of AFSA-PSO for feed forward Neural Network Training and showed that hybrid FSA-PSO has better global astringency and stability than standalone FSA and standalone PSO. S. Nandy et al. in paper [243] compared the performance hybrid ABC-BP with hybrid GA-BP on the basis of three parameters i.e. SSE (sum of squared error), convergence speed and stability on optimum solution for four data sets (iris, wine, soya bean and glass). It showed that ABC-BP is better than GA-BP with increased efficiency.

Various soft computing based approaches for NN learning are also compared with each other. Yun Cai [245] proposed Artificial Fish School Algorithm (FSA) for Combinatorial Optimization Problem and stated that the algorithm has better convergence performance than GA and ACO. Basturk and Karaboga in paper [246] compared the performance of ABC algorithm with GA, PSO and Particle Swarm Inspired Evolutionary Algorithm (PS-EA). The results showed that ABC outperforms the other algorithms. Basturk and Karaboga further in paper [247] compared the performance of ABC algorithm with that of DE, PSO and EA for a set of well known test functions. Simulation results show that ABC algorithm performs better than the mentioned algorithms and can be efficiently employed to solve the multimodal engineering problems with high dimensionality. D Karaboga and B Akay in paper [248] compared ABC with GA, PSO, DE and ES for optimizing a large set of numerical test. Results show that the performance of the ABC is better than or similar to those of other population-based algorithms with the advantage of employing fewer control parameters. V. Saishanmuga Raja et al. in paper [249] compared three optimization techniques GA, ACO and PSO in biomedical application based on processing time, accuracy and time taken to train Neural Networks. The paper concluded that GA outperformed the other two algorithms-ACO and PSO and is most suitable for training the neural network with minimum time and minimum mean square error. Dipti Srinivasan et al. in paper [250] proposed particle swarm inspired EA (PS-EA) and compared it with Genetic Algorithm (GA) and PSO. It is found that PS-EA is much superior over typical GA and PSO for complex multi-modal functions like Rosenbrock, Schwefel and Rastrigrin functions. A. Ghaffari et al. in his paper [251] presented the comparison of five training algorithms-two versions of gradient descent-IBP (incremental), BBP (Batch) and LM,QP ,GA with reference to the predicting ability. The convergence speed of BBP is three to four times higher than IBP. The performances in terms of precision of predictive ability were in the order of: IBP, BBP > LM > QP (quick propagation) > GA. Zhang et al. in his paper [252] presented application of bacteria foraging optimized neural network (BFO NN) for short term electric load forecast. This paper used BFO to find optimized weights of neural network while minimizing the MSE. Simulation results also showed that BFO NN converges more quickly than Genetic algorithm optimized neural network (GANN).

Ivona BRAJEVIC et al. [253] presented Training Feed-Forward Neural Networks Using Firefly Algorithm (FA) for classification purpose. This paper compares FA with GA and ABC for three well
known classification transfer function. The parameters used for comparison are MMSE: Mean of Mean Squared Errors, SDMSE: Standard Deviation of Mean Squared Errors, MC: Mean of Cycle Numbers, SDC: Standard Deviation of Cycle Numbers. It shows that FA performs better than GA algorithm, but worse than ABC algorithm for the majority of benchmark problems. It also stated that the choice of transfer functions may strongly influence the performance of neural networks, so it also compared the FA results obtained by using traditional sigmoid transfer function with another by using sine transfer function and showed that FA implemented using sine transfer function is much efficient with fast convergence speed. Ritwick in paper [254] proposed a modified version of Invasive Weed Optimization (MIWO) for training the feed-forward Artificial Neural Networks (ANNs) by adjusting the weights and biases of the neural network. In this, Modified IWO was compared with DE, BP, One step secant learning and RPROP based on MSE. Modified IWO is performed better than DE and other classical gradient-based optimization algorithms mentioned in terms of learning rate and solution quality. BBO has been further compared with another optimization algorithms like PSO, ACO and found better for the detection of abnormal growth of tissues in MRI image segmentation by Harpreet Kaur et. al [255]. S. Mirjalili et. al in paper [256] proposed hybrid PSO-GSA for training neural networks and proved it to outperform other optimization algorithms such as PSO and ACO in terms of converging speed and local minima avoidance. Saeide Sheikhpour in paper [257] proposed a hybrid GSA-GA for neural network training that uses the GSA(Gravitational Search algorithm) to do global search in the beginning of stage, and then uses the GA(Genetic Algorithm) to do local search around the global optimum and proved it more efficient than standard GSA and back propagation algorithm. Bao-Chang Xu et. al in paper [258] proposed an Improved Gravitational Search Algorithm (IGSA) for Dynamic Neural Network Identification. It showed the best performance when compared with the system identification based on gravitational search algorithm neural network (GSANN) and other conventional methods like BPNN and GANN. Saeed Ayat et. al in paper [259] compared various ANN learning algorithms in which 12 algorithms concerned with Perceptron multilayer neural networks were studied and 6 classical learning algorithms(Gdx, cgb, lm, oss, cgf, cgp) have presented an acceptable percentage of accountability. Conjugate gradient and LM is found to have better efficiency in reaction to the given training as compared to others. The paper concluded that LM is the most convergent and represented better predict of average. Nasser Mohammadi et. al in paper [260] compared the PSO with the variants of back propagation techniques (LM, GD, GDM, GDA, GDMA) based on mean square error (MSE) and accuracy. The paper concluded that LM is found to have better performance than other variants of BP but PSO is more superior to LM and other variants of BP. The performance level is found to be PSO>LM>other BP variants.

V. CONCLUSIONS

An extensive survey has been carried out on the available classical as well as the soft computing based approaches available in literature. In case of classical learning approaches, it is evident that not a single training algorithm can be proved best for all the test or benchmark problems. In fact it is the problem dependant. It is found that as the classical learning approach either EBP or LM is having the poor convergence speed so the soft computing based approaches are to be preferred as the global optimization approaches. These soft computing based approaches can be used to evolve ANN architecture or the synaptic weights. These approaches can be used standalone or in a hybrid manner with EBP or LM. It is evident that the hybrid techniques are found to be more efficient than standalone soft computing approaches. This survey will cover all the available classical and soft computing approaches to evolve ANN and will be beneficial for researchers to carry out their research further on evolving ANN preferring soft computing based approaches. BB BC or parallel BB BC is a new approach among soft computing which could be further utilized broadly for new ANN model identification algorithm.
REFERENCES


[34] Zhi-Hua Zhou, “Rule Extraction: Using Neural Networks or For Neural Networks?”, National Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China


[38] GG Towell and J W Shivalik, Knowledge-Based Artificial Neural Networks”, Appears in Artificial Intelligence, volume 69 or 70., Submitted 1/92, Final pre-publication revisions 8/94


[75] Hao Yu, Bogdan M. Wilamowski, Member IEEE “Levenberg–Marquardt Training”.


[96] Raymond Chiong, Member, IEEE, and Ooi Koon Beng, Member, IEAust43, A Comparison between Genetic Algorithms and Evolutionary Programming based on Cutting Stock Problem, Engineering Letters, 14:1, EL_14_1_14, Feb 2007.


[114] Min-Rong Chen, Xia Li, Xi Zhang, Yong-Zai Lu,”A novel particle swarm optimizer hybridized with extremal optimization”.


[127] Yana Mazwin Mohmad Hassim and Rozaida Ghazali, “Training a Functional Link Neural Network Using an Artificial Bee Colony for Solving a Classification Problems”.


[152] Jaeggi et al., “Multiobjective tabu Search for contrained optimization problems”.
[153] FRED GLOVER and BELEN MELIAN, “381 –TS-Intro to TS, Chapter Tabu Search”.
[171] Xin-She Yang1, “Harmony Search as a Metaheuristic Algorithm.


[237] Enrique Alba and J. Francisco Chicano, “Training Neural Networks with GA Hybrid Algorithms”.


[242] Huadong Chen et al., “A Hybrid of Artificial Fish Swarm Algorithm and Particle Swarm Optimization for Feedforward Neural Network Training”.


**APPENDIX**

**Table 1**

<table>
<thead>
<tr>
<th>Paradigm</th>
<th>learning Rule</th>
<th>Architecture</th>
<th>Learning Algorithm</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supervised Training</td>
<td>Error Correction</td>
<td>Single layer or Multilayer Perceptron</td>
<td>Perceptron learning Algorithm(LMS), BP, Adaline and Medaline</td>
<td>Function approximation, Prediction and control</td>
</tr>
<tr>
<td>Boltzmann</td>
<td></td>
<td>Recurrent</td>
<td>Boltzmann Learning algorithm</td>
<td>Pattern Classification</td>
</tr>
<tr>
<td>Hebbian learning</td>
<td></td>
<td>Multilayer feed forward</td>
<td>Linear discriminant Analysis</td>
<td>Pattern classification, data Analysis</td>
</tr>
<tr>
<td>Competitive Learning</td>
<td></td>
<td>Competitive</td>
<td>Learning Vector Quantization</td>
<td>within class categorization</td>
</tr>
<tr>
<td>ART networks</td>
<td>ART map</td>
<td></td>
<td></td>
<td>Pattern Classification, within class categorization</td>
</tr>
<tr>
<td>Un-Supervised Training</td>
<td>Error Correction</td>
<td>Multilayer Perceptron</td>
<td>Sammon’s Projection</td>
<td>Data Analysis</td>
</tr>
<tr>
<td>Hebbian learning</td>
<td></td>
<td>feed forward or Competitive; Hopefield network</td>
<td>Principal Component analysis; Associative Memory Learning</td>
<td>Data Analysis, Data compression; Associative Memory</td>
</tr>
<tr>
<td>Competitive Learning</td>
<td></td>
<td>Competitive</td>
<td>Vector Quantization</td>
<td>Categorization, Data Analysis</td>
</tr>
<tr>
<td>ART networks</td>
<td>ART-I, ART-II</td>
<td></td>
<td></td>
<td>Categorization</td>
</tr>
<tr>
<td>Hybrid</td>
<td>Error Correction and Competitive</td>
<td>RBF Networks</td>
<td>RBF Learning Algorithms</td>
<td>Pattern Classification, Function Approximation, Prediction &amp; Control.</td>
</tr>
</tbody>
</table>
## Table II
### Classical Learning Approaches

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Author, Reference</th>
<th>Algorithm</th>
<th>Author, Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Steepest Gradient Method)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Standard or Incremental Backpropagation (IBP)</td>
<td>Freeman and Skappura, 1991[46]</td>
<td>Levenberg-Marquardt algorithm (LM), Menhaj, 1994[76]</td>
<td></td>
</tr>
<tr>
<td>Resilient PROP (RPROP)</td>
<td>Riedmiller and Braun, 1993[58]</td>
<td>Restricted Gradient</td>
<td>A M Salles Baretto, CW Anderson, 2008[70]</td>
</tr>
<tr>
<td>Diagonal Estimation</td>
<td>1998 [61], TREAT algorithm</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grprop</td>
<td>Aristoklis D A et al., 2004[65][66]</td>
<td>No-Prop algorithm</td>
<td>Bernard Widrow et al., 2013 [272]</td>
</tr>
</tbody>
</table>
### Table III

**Soft Computing Based Approaches**

#### Evolutionary Algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Author, Reference</th>
<th>Algorithm</th>
<th>Author, Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>EANN (Evolutionary ANNs)</td>
<td>Yao, 1999[84][85]</td>
<td>GA-BP and GA-LM</td>
<td>Enrique Alba and J. Francisco Chicano [237]</td>
</tr>
<tr>
<td>GA</td>
<td>Holland, 1975[90][91]</td>
<td>Breeder Genetic Programming (BGP)</td>
<td>B T Zang [98]</td>
</tr>
<tr>
<td>GP</td>
<td>John Koza, 1992[97]</td>
<td>Improved DE</td>
<td>Shamekhi [105]</td>
</tr>
<tr>
<td>ES</td>
<td>1965, 1975, Richenberg, Schewefel [99][100]</td>
<td>self-adaptive DE (SaDE)</td>
<td>Qin et al. [106]</td>
</tr>
</tbody>
</table>

#### Physics or Chemistry Based Approaches

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Author, Reference</th>
<th>Algorithm</th>
<th>Author, Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parallel BB-BC</td>
<td>S Kumar et. al. [218]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Soft Computing Based Approaches

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Author, Reference</th>
<th>Algorithm</th>
<th>Author, Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>BFOA</td>
<td>Passino, 2002[143]</td>
<td>PSO</td>
<td>Eberhart and Kennedy, 1995[113]</td>
</tr>
<tr>
<td>GA-BFOA</td>
<td>D H Kim, 2007[147]</td>
<td>ACO</td>
<td>Dorigo &amp; Gambardella [122]-[125]</td>
</tr>
<tr>
<td>BFONN</td>
<td>Zhang et al. [252]</td>
<td>FA</td>
<td>Yang, 2009 [129]</td>
</tr>
<tr>
<td>Eurygaster Algorithm</td>
<td>Fariborz Ahmadi et al, 2012 [156]</td>
<td>FSA-PSO</td>
<td>Huadong Chen et al [242]</td>
</tr>
<tr>
<td>Hybrid BA</td>
<td>Fister Jr.[139]</td>
<td>Virtual bee algorithm</td>
<td>X S Yang [158]</td>
</tr>
<tr>
<td>Enhanced Bat Algorithm(EB)</td>
<td>A. Kaveh and P. Zakian</td>
<td>quaternion bat algorithm(QBA)</td>
<td>Fister,2013[140]</td>
</tr>
</tbody>
</table>

## Bio Inspired Non-si Approaches

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Author, Reference</th>
<th>Algorithm</th>
<th>Author, Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>IWO</td>
<td>Mehrabian and Lucas (2006) [159], Flower pollination algorithm, (X. S. Yang, 2012)[177]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modified IWO (MIWO)</td>
<td>Ritwick, 2010[254], photosynthetic algorithm, (Murase, 2000) [199]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improved HGIWO</td>
<td>Zhi YIN,2012 [160], enzyme algorithm, (Yang, 2005) [180],</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BBO</td>
<td>Dan Simon 2008[161], cross-entropy algorithm, Rubinstein (1999; 2001) [181][182]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AIS</td>
<td>Dasgupta, in 1999 [166], GSO S.He,Q.H.Wu et al., 2006][173]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NN Immune System</td>
<td>Leandro Nunes de Castro[168], cultural algorithms (CA) R. G. Reynolds,1999) [175]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HS</td>
<td>Geem et al., 2001[169], monkey search (MS) Mucherino and O. Seref, 2007) [176]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DHS</td>
<td>Prithwish Chakraborty et al. [172]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>