IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-4, April 2019 E-ISSN: 2347-2693

Optimizing Error Function of Backpropagation Neural Network

Munmi Gogoi[1], Ashim Jyoti Gogoi[2], Shahin Ara Begum[1*]

[1]Dept. of Computer Science, Assam University, Silchar, Assam, India
[2]Dept. of Electronics and Communication Engineering, National Institute of Technology Silchar, Assam, India

[*]*Corresponding Author: shahin.begum.ara@gmail.com, Tel.: +91-9435171932*

DOI: https://doi.org/10.26438/ijcse/v7i4.10111016 | Available online at: www.ijcseonline.org

Accepted: 22/Apr/2019, Published: 30/Apr/2019

Abstract - Backpropagation algorithm (BP) is one of the most popularized and effective learning algorithm for learning neural networks, starting with Multilayer perceptron's (MLP's) to today's Deep learning models in the domain of Artificial Intelligence (AI). Backpropagation algorithm works on two phases. The forward phase feed the network with input and communication links with synaptic weights, the activation function decides whether the hidden neurons fire or not. The primary focus of the present work is on the backpropagation error, which decides the amount of weight updating based on the errors. The driving force of the algorithm is to minimize the error by gradient descent where we differentiate the error function to get the gradient of the error and update the weights to reduce the error. In this paper, our approach is to reduce the error of Backpropagation neural network (BPNN) based on constraints using swarm intelligence based optimization method. For this, the optimization problem has been formulated mathematically with subjected constraints under the acceptable range of network parameters. This research investigation presents a comparison of results obtained from solving the minimization problem with different variants of swarm intelligence technique such as PSO, HBPSO, and ALCPSO.

Keywords— Backpropagation, Deep learning, PSO, HBPSO, ALCPSO

I. INTRODUCTION

Neural network (NN) is the most popular non-linear statistical data modeling tools and has shown to be universal approximators based on which today's numerous machine learning tools have been developed including deep learning networks [1-3]. Backpropagation (BP) algorithm is one of the most popularized and effective learning algorithms for learning MLP or more often MLP with two or more hidden layers are considered as deep networks or DNN. Backpropagation is a supervised learning algorithm proposed by [4], which can approximate a function to an arbitrary degree of accuracy based on an acceptable range of tuning parameters. Slow convergence, lengthier training time, network paralysis, local minima, etc. are the major issues associated with BP learning. Different approaches and improvement have been carried out on the BP algorithm to address the issues when dealing with very large problems [5]. But still, the definite global optimum solution needs to be responded. MLP with many hidden layers or today's deep learning models such as DNN are the trending machine learning models which have been developed to address the complications of high dimensional data related to the challenging domain of AI. [6]. Backpropagation learning did not work well in practice in deep models with multiple hidden layers due to the pervasive presence of local optima and other optimization challenges in the non-convex objective function of the deep network[7]. In connection to Backpropagation learning in deep network model [8-9] introduces Deep Belief Net or DBN which is a greedy layer by layer learning algorithm and with weight initialization on the deep network through a properly configured DBN produces better results than random weight initialization in MLP's. As such DBN-pre training is used to learn multiple layer networks or DNN's followed by back propagation fine tuning.

This paper focuses on the improvement of Backpropagation learning. The remainder of the paper is organized as follows. In section II the related works of the paper are discussed, where the section gives a brief description of previous work done on BP learning. The system model is presented in section III; The problem formulation is done in section IV, where the formulation of the optimization problem is presented mathematically into a minimization problem with some acceptable range of parameters. Discussions about the optimization algorithms are done in section V. Results and discussions are presented in section VI and finally, conclusions are drawn in section VII with results acquired from solving the minimization problem with different variants of swarm intelligence technique such as PSO, HBPSO, and ALCPSO.

II. RELATED WORK

A number of researches have been done to address the problem of Backpropagation learning in different ways. Many improvements have been made on the BP algorithm till now but the performance of these algorithms cannot be generalized and still, the question of the global optimum solution is to be responded. Choosing better activation function [10], improvement in hyperparameters such as dynamic learning rate, momentum factor [11-12], retaining adaptive rules other than gradient descent [13] are the different ways of improvement on BP learning. Over the past few years, numerous research had been done on adaptive momentum modification [14]. The concept of keeping momentum coefficient fixed or static in back-propagation learning (BPFM) is discussed in [15-16], but later on study reveals that dynamically changing momentum value in back propagation shows better performance in case of weight updating so it is essential that instead of fixing the momentum–coefficient it should be adjusted adaptively. In relation to Backpropagation learning in Deep network, [8, 17] proposed DBN pre-training for Deep neural network (DNN) followed by Backpropagation fine tuning. In [18] to pre-train a deep neural network another energy-based model with unsupervised learning has been projected which can be effectively used for pre-training a DNN, much like DBN followed by BP fine tuning. In recent time [19] propose a new random synaptic feedback weights support error backpropagation for deep learning where author demonstrates that the strong architectural constraint of the deep network is not required for effective error propagation, they extent a modest mechanism that allots blame by multiplying errors by even random synaptic weights. As backpropagation learning algorithm, this technique can also achieves very good results on variety of tasks by spreading teaching signals across multiple layers of neurons.

III. SYSTEM MODEL

Neural Network comprises of a layered set of neurons or nodes interconnected by communication links with some synaptic weights, and an activation function processes the total weighted sum. Initially to solve a problem of linearly separable domain hidden layers was not used but the concept of multiple hidden layers or multilayer perceptron arises and complexity upsurges when difficulties related to arbitrary decision boundary to arbitrary accuracy with rational activation functions are come across.

Here, the network can be considered as input, hidden and one output layer respectively in Figure 1. There are 10 inputs in the input layer and one output. The statistics of hidden layer neurons are kept 2/3 or 70% of the input layer [20-21]. The network is trained by Backpropagation algorithm and it is the common method of training a network is used in combination with an optimization technique such as gradient descent. The algorithm comprises two sequences, propagation, and weight update. After the forward phase, the network computes the error of the network using "sum of square error function", which calculates the difference between target t and network output y for each node, square them together and adds them altogether [22].

$$E(t,y) = \frac{1}{2}\sum_{k-1}^{n}(t_k - y_k)^2 \tag{1}$$

where t is the target and y is the network output.

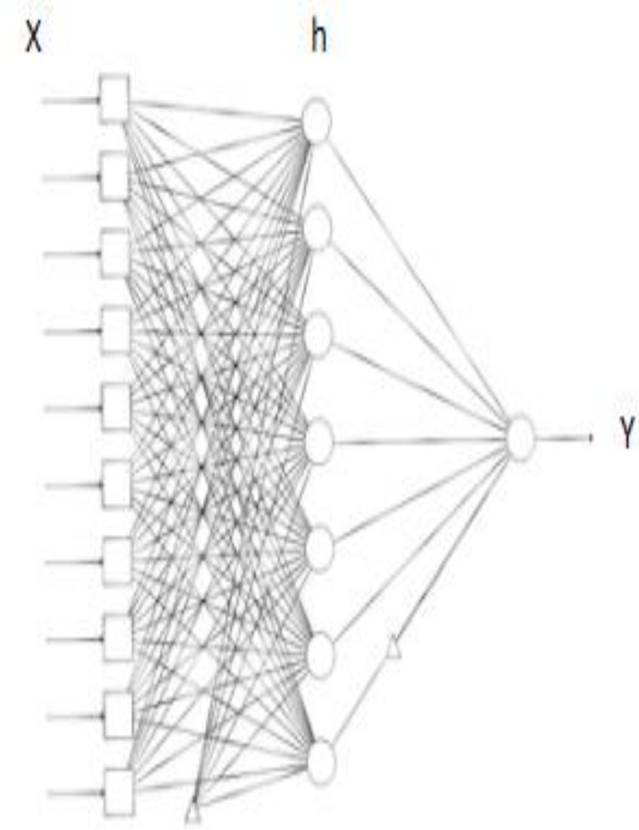

Figure 1: Network with 10 inputs, 7 hidden nodes and 1 output including the bias term connecting to each hidden neurons

To adjust the weights we first find the output layer error of the network given as in (2)

$$\partial_{ok} = \left(t_k - y_k\right) y_k \left(1 - y_k\right) \tag{2}$$

After computing the error of the output layer, the weights are updated as shown in (3).

$$w_{jk} = w_{jk} + \eta \partial_{ok} a_j \tag{3}$$

where, η is the learning rate and a_j is the hidden layer activations. Accordingly the hidden layer error can be computed as in (4).

$$\partial_{hj} = a_j \left(1 - a_j\right) \sum_{k} w_{jk} \partial_{ok} \tag{4}$$

Where a_j is hidden layer activation, w_{jk} is the weight and is multiplied with hidden layer error given in (2). Based on the computed errors weights are then updated to minimize the error of the network. From the equations, it is noticed that the aim is to minimize the error depending upon some network parameters such as η (learning rate), activation function, hidden layer size, etc. So it is desirable to have less error in each layer while training the network.

IV. Problem Formulation

As mentioned earlier, during training the network, the backward phase of the backpropagation algorithm computes the errors of the network and update weights accordingly to make the error smaller. At the same time to get the accuracy of the network, some parameters such as learning rate, momentum, hidden layer nodes, etc. are also considered. To minimize the error of the network the network parameters are kept within a suitable range.

Mathematically the optimization problem can be formulated as

$$MinError = \frac{1}{1+exp\sum x_i * v_{ij}} * \left(1 - \frac{1}{1+exp\sum x_i * v_{ij}}\right)\sum w_{jk} * (t_k - y_k)\, y_k\,(1-y_k) \qquad (5)$$

Subject to the constraint,

$$1/\sqrt{n} \le w \le -1/\sqrt{n}$$

$$w_l \le w_{jk} \le w_u$$

where $1/\sqrt{n} \le w \le -1/\sqrt{n}$ is the range of weight initialization based on the number of nodes in the input layer [*22*], and n is the number of nodes in the hidden layer. And w_{jk} is the range of weights of the hidden layer synaptic weights. w_l is lower and w_u is the upper limits of hidden layer weight respectively.

V. Optimization Algorithms

Swarm intelligence based optimization methods such as PSO, HBPSO. ALCPSO are population based stochastic optimization technique which simulates the common communication manners of birds flocking and fish schooling. This technique is anticipated by Kennedy and E berhart [23-24]. The advantages of these techniques include computational efficiency, better time complexity and faster convergence [25-28]. To solve the optimization problem, these optimization techniques have been used in our work and this section includes a brief discussion about these techniques.

A. Particle Swarm Optimization(PSO)

PSO [26] mimic the common behavior of bird flocks probing for the most favourable and best possible location in an n-dimensional search space. Each particle of the swarm i.e. the *N* agents move through the search space *D* to achieve the global optimal solution for the objective function. All particle of the swarm maintains two vectors position(x) and velocity(v) [29]. Let the i^{th} particle of position(x_i) and velocity(v_i) in dimension D can be stated as

$x_i=[x_{i1}, x_{i2}, x_{i3}, \dots, x_{id}, \dots, x_{iD}]$
$v_i = [v_{i1}, v_{i2}, v_{i3} \dots, v_{id}, \dots, v_{iD}]$
The particles having the best fitness may be either maximum or minimum giving the optimal result, so in our case, the best fitness would be the one having the least error.

In dimension *D* search space, for each iteration '*j*', particles update its position(x) and velocity(v) vector of the i^{th} particle based on two factors such: its own historical most excellent position and best position initiated by the swarm so far. The updating rules are as follows [30]

$$v_{id}^{j+1} = w.v_{id}^{j} + c_1.rand_1^{j}.(pbest_{id}^{j} - x_{id}^{j} + c_2.rand_2^{j}.\left(gbest_d^{j} - x_{id}^{j}\right) \qquad (6)$$

$$x_{id}^{j+1} = x_{id}^{j} + v_{id}^{j+1} \qquad (7)$$

Here, $pbest_{id}$ ($pbest^1_{id}$, $pbest^2_{id}, \dots\dots\dots$ $Pbest^j_{id}$,)is the historically best position of particle *i*, and this *i*, *j*, and *d* indices varies their range from{(*i*=1,2,…, *N*)},{(*d*=1,2……*maximum D dimension*)}, {(*j*=1, 2,……. j^{th} number of iteration)}. $gbest_{id}$ ($gbest^1_{id}$, $gbest^2_{id}, \dots\dots\dots$ $Gbest^j_{id}$,) denotes globally best position of the entire population, $rand_1$ and $rand_2$ are two random numbers distributed uniformly between {0,1}, *w* is inertia factor which controls the quantity the current velocity affects the velocity of the subsequent time stamp, and c_1, c_2 denotes acceleration coefficients to evaluate the relative significance of *pbest* and *gbest* correspondingly. Consequently for a minimization problem, we have,

$$gbest(t)=argmin\{f(pbest^1),f(pbest^2),\dots\dots\dots\dots\dots f(pbest^n)\} \qquad (8)$$

Though PSO includes a good computational efficiency and straightforward implementation approach, but still it may suffers from the problem of stagnation effect [25]. So as a solution researcher have proposed variants of PSO algorithms.

B. Human Behavior Based Particle Swarm Optimization (HBPSO)

HBPSO is first introduced by Liu *et. al.* in the year 2014 [27]. Human behavior based PSO is integrated to enhance the performance of PSO but the working principle has remained the same as in PSO. In general human beings are influenced by good and bad habits of peoples around them, learning from bad habits is always harmful to us, and on the contrary, if we can take these bad habits as a warning then it may advance maturity in our activity. Therefore, while modeling group activities we are required to give a rational observation on these bad behaviors. In the velocity updating rule of equation 7, the term gworst (Global worst) represents the worst fitness among the entire populace and is defined as in (9):

$$gworst(t) = \arg\min \{ f(pbest_1), f(pbest_2), .., f(pbest_n)\} \quad (9)$$

Here *f (.)* denotes fitness value of the particle. A new random variable $rand_3$ has been introduced by authors in HBPSO such that $rand_3 \in N(0, 1)$ with some condition *if* $rand_3>0$ then it will enhance the "flying "velocity of the particle and *if* $rand_3<0$ then it will decrease the "flying" velocity or the vice versa. This means that the conditions can be considered as impelled learning coefficient and penalized learning coefficient respectively, and has an adverse effect on enhancing the employment of the particle. Along with *if* $rand_3=0$, than there is no effect on the particle. The velocity updating rule of HBPSO is in (9) where the acceleration coefficient (c_1, c_2) of (6) has been removed. The position update rule remains the same as in (7) of PSO. Therefore, the updating rule of HBPSO is as in (10).

$$v_{id}^{j+1} = w.v_{id}^{j} + rand_1^{j}.(pbest_{id}^{j} - x_{id}^{j}) + rand_2^{j}.\left(gbest_d^{j} - x_{id}^{j}\right) + rand_3^{j}\left(gworst_d^{j} - x_{id}^{j}\right)$$
(10)

C. Aging Leader and Challengers Particle Swarm Optimization (ALCPS)

ALCPSO approach is employed to boost a pertinent leader which can lead the swarm through aging, here a new term Leader had been introduced with a fine lifetime and based on the leading command the lifetime can be adapted consequently [28]. Once the lifetime is exhausted of a leader, it is replaced by a new particle. The velocity updating rule of ALCPSO is given in (11).

$$v_{id}^{j+1} = w.v_{id}^{j} + c_1 rand_1^{j}.(pbest_{id}^{j} - x_{id}^{j}) + c_2 rand_2^{j}.\left(Leader_d^{j} - x_{id}^{j}\right) \quad (11)$$

The flowchart of the optimization process using the optimization algorithms in this work is shown in Fig. 2. The position update processes for each algorithm is carried out several times until the best solution for the respective algorithm satisfying all the constraints of the optimization problem is attained.

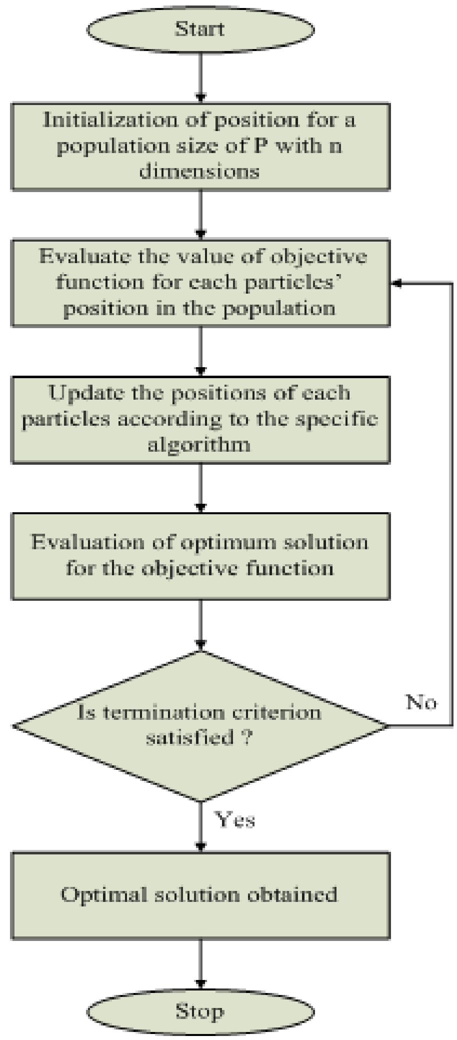

Figure.2 Flowchart of the optimization process [29]

VI. RESULTS AND DISCUSSIONS

Error minimization is the prime goal of the Network computed through the Backpropagation algorithm under the subjected constraints of the minimization problem. In order to understand the error minimization of Back-propagation algorithm formulated here against the subjected constraints, we simulate it in a model of MLP's with 3 layers of network .The parameter values of the network for the optimization process is taken depending on the architecture considered in this paper as given in Table 1. The hardware specification of core *i7* intel processor, 8 GB RAM, with 2GB *Nvidia Geforce GTX* graphics card has been used to carried out the experimentation in a MATLAB environment. The PSO technique and its variants such as HBPSO, ALCPSO has been used to optimize the errors computed through backpropagation. The comparative results of these three optimization algorithms implemented in this work and the convergence plot of the algorithms are shown in Fig 3. The outcomes of the optimization process shown in Table 2, where the ALCPSO presents minimum error in comparison to HBPSO and PSO.

Table 1. Parameters and Values

Parameters	***Values***
W_l	**-1**
W_u	1
N	7
V_i	0.4
T_k	0.1
Y_k	0.46

Table 2. Results of Optimization Process

Parameters	*PSO based optimum values*		*HPSO based optimum values*		*ALCPSO based optimum values*	
	values of parameters	*Error*	*values of parameters*	*Error*	*values of parameters*	*Error*
W	0.0083	-0.154	0.0021	-0.155	-0.0041	-0.157
W_{jk}	0.893		0.991		1	

From Table 2 it is seen that error of ALPSO based optimum value is lesser than PSO and HBPSO based optimum values i.e. -0.157, where the values of the parameters of W and W_{jk} is -0.0041 and 1 respectively. The parameter value W represents the range of weight initialization based on the number of nodes in the input layer as mentioned in the constraints formulated in (5) i.e. $1/\sqrt{n} \le w \le -1/\sqrt{n}$ and W_{jk} is the range of weights of the hidden layer synaptic weights.

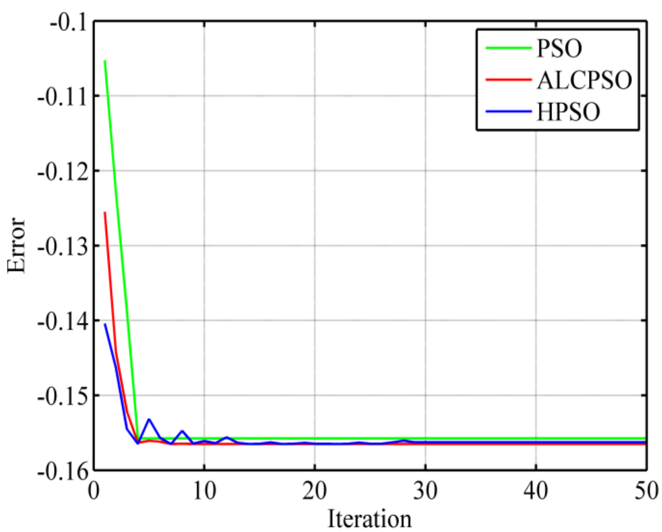

Figure 3. Convergence graph of algorithms used in the network

The convergence plots of the aforementioned algorithms based on error and iteration is shown in Figure 3, which reveals that the system optimized with ALCPSO algorithm exhibits less error than the other two algorithms while computed through backpropagation. So, the system optimized with ALCPSO algorithm is more efficient than that of the other algorithms to be computed with backpropagation algorithm.

VII. CONCLUSION AND FUTUREWORK

Backpropagation algorithm is a supervised learning algorithm widely used to learn the Neural Network model based on which different machine learning model have been established to address the real world problems around the globe. Though it suffers from the difficulties of slow convergence, local minima, network paralysis, etc., it has been used to learn networks starting from MLPs to deep network while back propagation fine-tuning has been done on DNN followed by DBN pre-training. In the present paper an experiment has been carried out to enhance the BP algorithm by formulating the optimization problem mathematically to minimize the errors. The optimization of the minimization problem has been carried out using PSO algorithm and a comparison between the variants of PSO algorithm has been presented. Further research can be done in this field by implementing this algorithm in a deep network though this experimentation has been done in a three layered network model.

REFERENCES

[1] S. I. Gallant, *"Perceptron-based learning algorithms"*, IEEE Trans. on Neural Networks, vol. 1, no. 2, pp. 179-191, 1990.

[2] K. Hornik, M. Stinchcombe, H. White, *"Multilayer feedforward networks are universal approximators*", Neural Networks, vol. 2, no. 5, pp. 359-366, 1989.

[3] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models", in Advances in Neural Information Processing Systems, pp. 3581–3589, 2014.

[4] D. E. Rumelhart , G.E Hinton, R. J. Williams, "Learning internal representations by error propagation", in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, 1986.

[5] Saduf, M.A. Wani, A. Wani. "Comparative study of back propagation learning algorithms for neural networks" ,International Journal of Advanced Research in Computer Science and Software Engineering 3.12 ,**2013**.

[6] Y. Le Cun, Y. Bengio,G. Hinton, "Deep Learning", Nature. 521. P.436, 2015.

[7] N.S. Lele, "*Image Classification Using Convolutional Neural Network*", International Journal of Scientific Research in Computer Science and Engineering, Vol.6, Issue.3, pp.22-26, 2018. .

[8] G. E Hinton, S. Osindero, Y. W. The, "A fast learning algorithm for deep belief nets", Neural computation, vol. 18, no. 7, pp. 1527-1554.

[9] G.E. Hinton, and R.R Salakhutdinov, "Reducing the dimensionality of data with neural networks", Science, vol. 313, no. 57, pp.504-507, 2006.

[10] P, Chandra, Y. Singh, *"An activation function adapting training algorithm for sigmoid feed forward networks"*, Neurocomputing vol. 61, pp. 429–437, 2004.

[11] N. K. Bose and P. Liang, "Neural Network Fundamentals with Graph Algorithms and Applications", McGraw-Hill, 1996

[12] Y. H. Zweiri, L D. Seneviratne, K. Althoefer, "Stability analysis of three term back propagation algorithm", Neural Networks, vol. 18, pp. 1341-1347, 2005.

[13] S. Haykin, “Neural Networks: A Comprehensive Foundation”, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[14] C. C. Yu, Bin Da Liu, “A backpropogation algorithm with adaptive learning rate and momentum coefficient”, in IEEE International Joint Conference on Neural Networks, vol. 2, pp. 1218-1223, 2002.

[15] H. Shao, H. Zheng, “A new bp algorithm with adaptive momentum for FNNs training”, in: GCIS 2009, Xiamen, China, pp. 16—20, 2009.

[16] M. Z. Rehman, N. M. Nawi, M. I. Ghazali, “Noise-induced hearing loss (NIHL) prediction in humans using a modified back propagation neural network,” in 2nd International Conference on Science Engineering and Technology, pp. 185—189, 2011.

[17] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, *“Greedy layer-wise training of deep networks”,* in Advances in neural information processing systems, pp. 153-160, 2007.

[18] C. Poultney, S. Chopra, Y. L. Cun, *“Efficient learning of sparse representations with an energy-based model”,* in Advances in neural information processing systems, pp. 1137-1144, 2007.

[19] X. Kai, “Research on the Improvement of BP Neural Network Algorithm and its Application”, Advanced Materials Research, vol. 926, pp. 3216-3219, 2014

[20] Z. Boger, H. Guterman, *“Knowledge extraction from artificial neural network models”,* in IEEE international conference on Systems, Man, and Cybernetics Orlando, FL, USA 1997.

[21] M.J.A. Berry, G. Linoff, “Data Mining Techniques”, NY: John Wiley & Sons 1997.

[22] S. Marsland *“Machine Learning an Algorithmic Prospective”* CRC Press,Tylor and Francis group, London, Newyork 2009.

[23] J. Kennedy and R. C. Eberhart, “Particle swarm optimization”, in Proceedings of the IEEE International Conference on Neural Networks, pp 1942-1948, Dec. 1995.

[24] J. Kennedy, “Particle swarm optimization”, Encyclopedia of machine learning. Springer US, pp. 760-766, 2011.

[25] S. S. Rao “Engineering optimization: theory and practice”, John Wiley and Sons, 2009.

[26] K. Kamiyama, “Particle Swarm Optimization-A Survey”, ICICE transactions on information and systems”., vol. 92, no. 7, 2009.

[27] L. Hao, “Human behavior-based particle swarm optimization”, The Scientific World Journal 2014.

[28] Wei-Neng Chen, “Particle swarm optimization with an aging leader and challengers”, IEEE Transactions on Evolutionary Computation 17.2 (2013): 241-258.

[29] A. J. Gogoi, N. M. LASKAR, Ch L.Singh, K. L. Baishnab, and K. Guha, “Throughput Optimization in Multi-user Cognitive Radio Network using Swarm Intelligence Techniques”, Journal of Information Science & Engineering, vol.34, no. 4, 2018.

Authors Profile

Ms.Munmi Gogoi pursed Bachelor of Computer Application from Punjab Technical University in 2010 and Master of Computer Application from Dibrugarh University Assam,India in the year 2013. She is currently pursuing Ph.D. degree in Department of Computer Sciences, Assam University Silchar,India,since 2015. Her main research work focuses on Artificial Neural Network, Deep Learning,and Image classification.

Mr.Ashim Jyoti Gogoi received his B.E. degree from Gauhati University and M. Tech degree from NERIST, India in 2010 and 2014, respectively. He is currently pursuing Ph.D. in National Institute of Technology Silchar, India. His research interest includes cognitive radio network, image processing, MIMO communications and Modern Optimization methods.

Dr(Mrs) Shahin Ara Begum pursed her Bachelor of Science from Benglore University and Master of Science from Jamia Milia Islamia University India. She has pursed her Ph.D. from Assam University Silchar and currently working as Associate Professor in the Department of Computer Science,Assam University Silchar since 1997. She has published more than 30 research papers in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences as well. Her main research work focuses on Machine Learning, Deep learning Soft Computing Techniques, Pattern Recognition and Data Mining.

