ISSN: 2454-4248
49 – 57

A Review on Optimizing Radial Basis Function Neural Network using Nature Inspired Algorithm

Ms. Tanvi Gupta [1], Supriya P Panda[1], S.S. Handa[2] and M.V.Ramana Murthy [3]
[1]Manav Rachna International University [2]SRM University Manav Rachna International University [3] Department of Mathematics and Computer Science, foundation for scientific Research and Technological innovation, (Osmania University), Hyderabad -102
Email: *[1]tanvigupta.fet@mriu.edu.in [2]rohit.handa@gmail.com [3] mv.rm50@gmail.com*

Abstract: Radial Basis Function (RBF) is a type of feed forward neural network .This function can be applied to interpolation, chaotic time-series modeling, control engineering, image restoration, data fusion etc. In RBF network, parameters of basis functions (such as width, the position and number of centers) in the nonlinear hidden layer have great influence on the performance of the network. Common RBF training algorithms cannot possibly find the global optima of nonlinear parameters in the hidden layer, and often have too many hidden units to reach certain approximation abilities, which will lead to too large a scale for the network and decline of generalization ability. Also, RBF neural network has the advantages of adaptive and self-learning ability, but it is difficult to determine the number of hidden layer neurons, and the weights learning ability from hidden layer to the output layer is low; these deficiencies easily lead to decreasing learning ability and recognition precision. Secondly, the Swarm Intelligence Algorithms are (Meta-Heuristic) development Algorithms, which attracted much attention and appeared its ability in the last ten years within many applications such as data mining, scheduling, improve the performance of artificial neural networks (ANN) and classification. So, in this paper the work of Artificial Bee Colony (ABC), Genetic algorithm(GA), Particle swarm optimization(PSO) and Bat algorithm(BA) have been reviewed, which optimized the RBF neural network in their own terms.

Keywords: *RBF, neural network, swarm intelligence, ABC algorithm, bat algorithm*

*****

I. Introduction

Radial basis function

Neural networks are non-linear statistical data modeling tools and can be used to model complex relationships between inputs and outputs or to find patterns in a dataset. RBF**[9]** network is a type of feed forward neural network composed of three layers, namely the input layer, the hidden layer and the output layer. Each of these layers has different tasks. A general block diagram of an RBF network is as depicted in Figure 1 below:

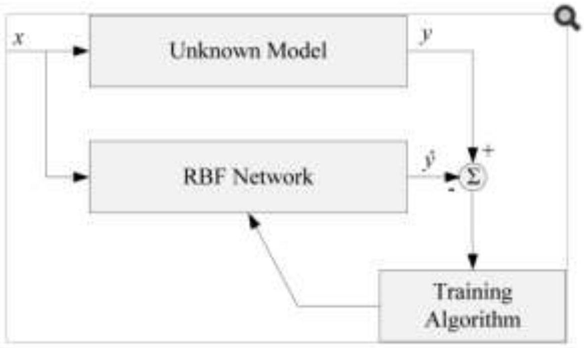

Figure 1 : Block Diagram of RBF

In RBF networks, the outputs of the input layer are determined by calculating the distance between the network inputs and hidden layer centers. The second layer is the linear hidden layer and outputs of this layer are weighted forms of the input layer outputs. Each neuron of the hidden layer has a parameter vector called center. Therefore, a general expression of the network can be given as

$$\hat{y}_j = \sum_{i=1}^{I} w_{ij}\, \phi\left(\| \mathbf{x} - \mathbf{c}_i \|\right) + \beta_j \qquad \text{--(1)}$$

The norm is usually taken to be the Euclidean distance and the RBF is also taken to be Gaussian function and defined as follows:

$$\varphi(r) = \exp\left(-\alpha_i \cdot \| \mathbf{x} - \mathbf{c}_i \|^2\right) \qquad \text{-(2)}$$

Where,

I Number of neurons in the hidden layer i€{1,2,.....I}

J Number of neurons in the output layer j€(1,2,……J}

w_{ij} *weight of the ith neuron and jth output*

Φ Radial Basis Function

a_i *Spread parameter of the ith neuron*

International Journal on Future Revolution in Computer Science & Communication Engineering
Volume: 3 Issue: 10
ISSN: 2454-4248
49 – 57

x *input data vector*

c_i *center vector of the ith neuron*

β_j *bias value of the output jth neuron*

$\hat{y}_j$ *network output of the jth neuron*

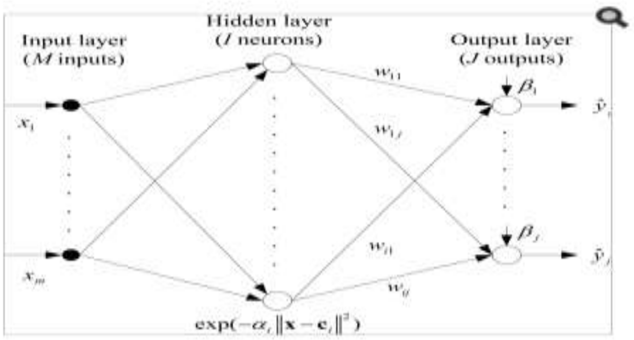

Figure 2 : Network Architecture of RBF

The detailed architecture of an RBF network is shown above in Figure 2. M -dimensional inputs $(x_1,\ldots,x_m)$ are located in the input layer, which broadcast the inputs to the hidden layer. The hidden layer includes I neurons and each neuron in this layer calculates the Euclidean distance between the centers and the inputs. A neuron in the hidden layer has an activation function called the basis function. In the literature, the Gaussian function is frequently chosen as the radial basis function and it has a spread parameter to shape the curve $(\alpha_1,\ldots,\alpha_i)$. The weighted $(w_{11},\ldots,w_{ij})$ outputs of the hidden layer are transmitted to the output layer. Here, I $(i \in \{1,2,\ldots,I\})$ denotes the number of neurons in the hidden layer and J $(j \in \{1,2,\ldots,J\})$ denotes the dimension of the output. The output layer calculates the linear combination of hidden layer outputs and bias parameters $(\beta_1,\ldots,\beta_j)$. Finally, the outputs of the RBF network are obtained $(\hat{y}_1,\ldots,\hat{y}_j)$.

The design procedure of the RBF neural network includes determining the number of neurons in the hidden layer. Then, in order to obtain the desired output of the RBF neural network w, α, c and β parameters might be adjusted properly. Reference based error metrics such as mean square error (MSE) or sum square error (SSE) can be used to evaluate the performance of the network. Error expression for the RBF network can be defined as follows:

$$E^{SSE}(w,a,c,\beta) = \sum_{j=1}^{J} (y_j - \hat{y}_j)^2$$

-(3)

Here, y_j indicates the desired output and $\hat{y}_j$ indicates the RBF neural network output. The training procedure of the RBF neural network involves minimizing the error function.

Radial Basis Function Network Model

The RBF network topological structure is shown below in Figure3. The network consists of three layers, namely the input layer, radial basic function hidden layer and output layer. The input part does not transform the signals but only dispatches the input vector to the radial basic layer. The function in a hidden layer node (also called nucleus function) responds partly to the input signals, i.e. when the input function is close to the center range of the nucleus function, the hidden layer will produce a larger output. The output layer makes output values through a linear combination of outputs from the hidden layer.

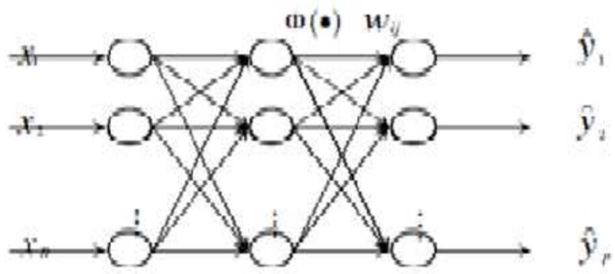

Figure 3: Structure of RBF neural network

Here input vector X= $\left[x_1, x_2, \ldots, x_R\right]$;C_i-the center of RBF neural network, a constant vector with the same dimension as X;R-the dimension of input network ,M-neurons number of hidden

 ISSN: 2454-4248
49 – 57

layer; $\Phi(\cdot)$ - RBF , $\|X - c_i\|$ -Eucledian distance between X and C_i; j -output node ,j=1,2,…..,P; W_{ij}-the weight value which connected the i^{-th} hidden node with j^{-th} output node.

As shown above in the Figure 3, ideal output y_j (j=1,2,,…,P), the actual output $\hat{y}_j$ and the weight value of the output layer W_{ij} can be obtained by RBF neural network.

Choosing Gaussian function

$$\Phi_i(x) = \exp\left(-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right)$$

as RBF , the actual output $\hat{y}_j$ is calculated by the following formula:

$$\hat{y}_j = \sum_{i=1}^{M} w_{ij}\Phi_i(x) = \sum_{i=1}^{M} w_{ij}\exp\left(-\frac{\|x - c_i\|^2}{2\sigma_i^2}\right) \qquad \text{-(4)}$$

Then, the weight value W_{ij} is adjusted to satisfy the following formula, from which the final result of the RBF neural network can be obtained.

$$E = \sum_{j=1}^{P}\left(y_j - \hat{y}_j\right)^2 = \sum_{j=1}^{P}\left(y_j - \sum_{i=1}^{M} w_{ij}\Phi_i(x)\right)^2 \qquad \text{--(5)}$$

II. Literature Survey

A. RBF Neural network trained using ABC firefly algorithm

Tuba kurban.et.al [9] suggested that training of an RBF neural network can be obtained with the selection of the optimal values for the following parameters:

- weights between the hidden layer and the output layer (w)
- spread parameters of the hidden layer base function (α)
- center vectors of the hidden layer (c)
- bias parameters of the neurons of the output layer (β)

The number of neurons in the hidden layer is very important in neural networks. Using more neurons than that is needed causes an over learned network and moreover, increases the complexity. Therefore, it has to be investigated how the numbers of neurons affect the network's performance.

The individuals of the population of ABC include the parameters of the weight ($\vec{w}$), spread ($\vec{\alpha}$), center ($\vec{c}$) and bias ($\vec{\beta}$) vectors. An individual of the population of ABC algorithm can be expressed as:

$$Pi=[\vec{w} \quad \vec{\alpha} \quad \vec{c} \quad \vec{\beta}] \text{---(6)}$$

The quality of the individuals (possible solutions) can be calculated using an appropriate cost function. In the implementation, SSE between the actual output of the RBF network and the desired output is adopted as the fitness function:

$$f = E^{SSE} \qquad \text{----(7)}$$

B. RBF Neural network trained using Genetic Algorithm(GA)

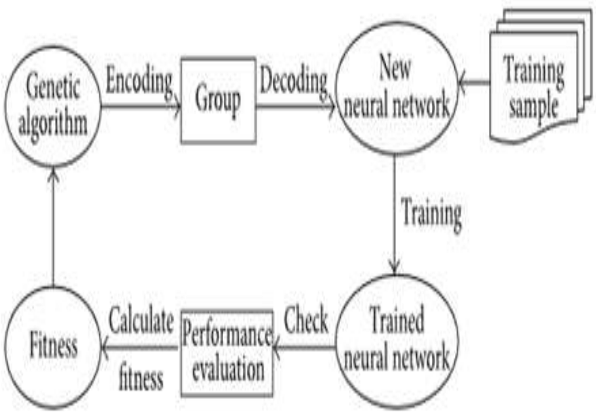

Figure 4: The flowchart of GA-RBF algorithm[10]

 ISSN: 2454-4248
49 – 57

Weikuan Jia.et.al[10] have suggested some steps of GA-RBF neural network.

The GA–RBF algorithm neural network basic steps are described as follows:

Step1: Set the RBF neural network, according to the maximum number of neurons in the hidden layer; using K-clustering algorithm to obtain the center of basis function $\sigma = \frac{d}{\sqrt{2s}}$. to calculate the width of the center.

Step2: Set the parameters of GA, the population size, crossover rate, mutation rate, selection mechanism, crossover operator, mutation operator, the objective function error, and the maximum number of operations.

Step3: Initialize population P randomly; its size is N (the number of RBF neural network is N); the corresponding network to each individual is encoded by formula

$$c_1c_2\cdots c_s w_{11}w_{21}\cdots w_{s1}w_{12}w_{22}\cdots w_{s2}\cdots w_{1m}w_{2m}\cdots w_{sm}\theta_1\theta_2\cdots\theta_m.$$

Step 4: Use the training sample to train the initial constructed RBF neural network, whose amount is N; use formula $e = \sum_{k=1}^{n}(t_k - y_k)^2.$ to calculate the network's output error E.

Step 5: According to the training error E and the number of hidden layer neurons s , use $F = C - E \times \frac{s}{s_{\max}}.$ to calculate corresponding chromosome fitness to each network.

Step 6: According to the fitness value , sort the chromosome; select the best fitness of the population, denoted by $F_{b;}$ verify $E<E_{min}$ or $G>G_{max}$;if yes, turn to Step 9;otherwise turn to Step 7.

Step 7: Select several best individual to be reserved to the next generation NewP directly.

Step 8: Select a pair of chromosomes for single-point crossover, to generate two new individuals as members of next generation; repeat this procedure, until the new generation reaches the maximum size of population Ps; at this time the coding will be done separately.

Step9: Mutate the population of new generation; binary coding part and real number coding part should use different mutation strategies. Then the new population is generated; set P=NewP, G=G+1, return to step 4.

Step 10: Get the optimal neural network structure, and the iteration of genetic algorithm is terminated, which means the optimizing stopped.

Step 11: The new neural networks weight learning is not sufficient, so use LMS method to further learn the weights. End of the algorithm.

C. RBF Neural Network Based on Particle Swarm Optimization

Yuxiang Shao.et.al [12] stated that an RBF neural network, whose parameters including clustering centers, variances of Radial Basis Function and weights are optimized by PSO algorithm. Therefore it has not only simplified the structure of RBF neural network, but also enhanced training speed and mapping accurate. The performance and effectiveness of the proposed method are evaluated by using function simulation and compared with RBF neural network. The result shows that the optimized RBF neural network has significant advantages inspect of fast convergence speed, good generalization ability and not easy to yield minimal local results.

ISSN: 2454-4248
49 – 57

In [12] a hybrid RBF training method combining PSO algorithm is proposed. In this method, PSO algorithm is used to determine the structure and parameters in RBF hidden layer, and RBF linear output weights as well. The experiments showed that this method improved effectively the convergence speed of the algorithm and overcomes the problem of immature convergence, and the method integrating RBF network and PSO algorithm are effective and feasible.

In the paper written by Yuxiang Shao.et.al [12] PSO algorithm uses real-coded , makes C_i, σ_i

And w_{ij} of RBF network as a particle.The whole integration step is summarized as following:

Step1: Initialize swarm, including swarm size, each particle's position and velocity; give initial value:w_{max} , w_{min} and generation=0.

Step 2: Set the range of w_{ij} is [w_{min},w_{max}] makes C_i, σ_i , and w_{ij} of RBF network as a particle. Thus, build the initial swarm.

Step 3: Calculate individual firtness, decode the individual, assign them to C_i, σ_i , and w_{ij} of RBF network. Calculate the study sample' output error. The fitness of particle *a* is defined as:

$$f(a)=\sum_{i=1}^{p}\sum_{j=1}^{n}\left(y_{ij}-t_{ij}\right)^2 \quad -(8)$$

Where ,y_{ij} is calculated output of individual network I, t_{ij} is the expected output , n is the number of training set examples, and P is the number of output nodes.

Step 4: Determine whether meet the conditions to terminate the algorithm.

Step5: If it meets the condition , it will go to step 6, else generate next swarm by PSO algorithm, find new P_i and P_g, update P_g of the swarm and P_i of each particle , goto step3.

Step6: Decode the optimal individual searched by PSO algorithm assign them to C_i, σ_i ,W_{ij} of RBF Network.

D. RBF optimization using bat algorithm

Ruba Talal [8] states that the process of training RBF network occurs using bat algorithm (BA) by choosing the optimal parameters of the weights between the hidden layer and the layer output (w) and the parameter spread (α) to function mainly in the hidden layer, Centers hidden layer (µ) and bias of the cells in a layer output (β).

The determine number of cells in the hidden layer is very important in RBF network, if the number of cells is few that leads to slow speed of convergence However, if number is a large that leads to the complexity of the network structure, so it was in the paper (2,3,4,5,7,11)choose the number of neurons in the hidden layer of the RBF network. The possible solutions are calculated by the fitness function (MSE).

$$f(\varepsilon, y_n) = MSE(y) = \sqrt{\frac{\sum_{j=1}^{n}\sum_{k=1}^{m}\left(T_{jk}-Y_{jk}\right)^2}{nm}} \quad -(9)$$

Figure 5 below shows the BA algorithm for training the RBF network, where the algorithm begins to read the data set and then set the required parameters of the RBF network in terms of the number of hidden cells, and the maximum generations . The next step is to determine the parameters controlling of BA algorithm. In each generation, every particle evaluated based on a scale MSE as well as dependence on the values of w and µ and ß and α.

ISSN: 2454-4248
49 – 57

- BAT is initializes and passes the best weights to RBF
- Load the training data
- While MSE < Stopping Criteria
- Initialize all BAT Population
- Bat Population finds the best weight and ε in Equation 11 and pass it on to the network in Equation 1 and Equation 2.
- RBF neural network runs using the weights and ε initialized with BAT
- Bat keeps on calculating the best possible weight,α,β,μ at each period until the network is converged.
- End While.

Figure 5: Improving Radial Basis function using Bat Algorithm(BA)

III. Conclusion

1) GA is population based evolutional optimization algorithm and it has been used for training RBF network. It is more robust for finding the global minimum. However, population based methods have a disadvantage such as slow converging rate. ABC is an evolutional optimization algorithm that is inspired by the foraging behavior of honey bees. It is a very simple and robust optimization technique. The results show that the ABC algorithm is more robust than GA because of the changes in standard deviations. Average results of the training results show that the ABC algorithm is better than the other methods. Results show that randomly selected data do not affect the performance of the ABC algorithm.
2) Particle Swarm Optimization (PSO) is a relatively recent heuristic search method that is based on the idea of collaborative behavior and swarming in biological populations. PSO is similar to the Genetic Algorithm (GA) in the sense that they are both population-based search approaches and that they both depend on information sharing among their population members to enhance their search processes using a combination of deterministic and probabilistic rules.

Characteristics of the UCI dataset

	Inputs	Outputs	Total Samples	Training Samples	Test Samples
Iris	4	3	150	90	60
Wine	13	3	178	106	72
Glass	9	2	214	128	86

Table 1: UCI Dataset [9]

Control parameters of GA.

Population size	Parameter count
Number of generations	4,000
Selection type	Roulette
Mutation type	Uniform
Mutation rate	0.05
Crossover type	Single point
Crossover ratio	0.8

Figure 6: Control Parameters of GA[9]

Control parameters of ABC.

Population Size	Parameter count
Number of generations/cycles	4,000
Limit (ABC)	400

Figure 7: Control Parameters of ABC[9]

In the experiments percent of correctly classified samples (PCCS) metric is used as the performance measure:

$$PCCS = \frac{Correctly\ Classified\ Samples}{Total\ Samples} \times 100 \qquad -(10)$$

ISSN: 2454-4248
49 – 57

Statistical PCCS results of Iris dataset.

		Hidden Layer Neurons							
		1	2	3	4	5	6	7	8
GA	Train	63,5 (12,3)	89,9 (8,6)	94,1 (3,8)	96,1 (2,0)	96,0 (1,7)	96,6 (1,9)	97,4 (1,4)	97,1 (1,1)
	Test	58,9 (12,3)	88,1 (9,6)	91,9 (5,3)	94,3 (3,9)	95,5 (2,9)	94,6 (3,9)	95,9 (2,2)	96,1 (2,5)
ABC	Train	70,6 (5,2)	96,1	97,1	97,9	97,5	97,8	98,0	98,0

Table 2: Statistical PCCS results of Iris dataset [9]

3) RBF neural network system which is optimized by PSO algorithm has better convergence rate and higher learning precision. Meanwhile RBF neural network system which is optimized by PSO algorithm can obtain better simulation results compared with RBF neural network.

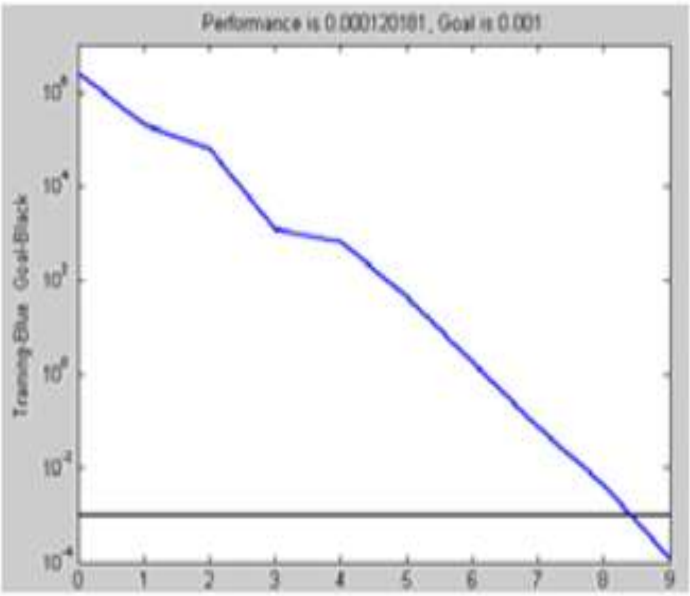

Figure 8: Training curve of RBF network[12]

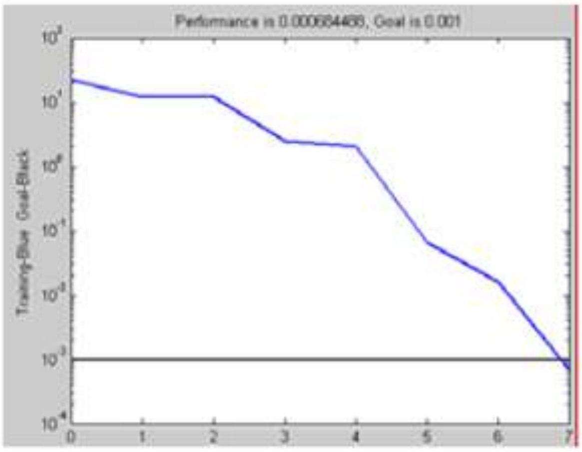

Figure 9: Training Curve of RBF network optimized by pso algorithm[12]

4) Although the PSO algorithm has the power to find a Global Minimum, but its society has a slow rate of convergence in finding from optimal solution, therefore the Bat algorithm is better because it's based on the principle of frequency tuning and change the emission rate of impulses which lead to the good affinity from ideal solutions, in addition to the creation process of a balance between exploration and exploitation and accelerate the training time, which led to increase network efficiency and reduce the fall errors and thus the algorithm is very efficient in multiple applications, such as image processing and clustering.

dataset	No. of samples	No. of properties	No. of classes
Iris	150	4	3
Wine	178	13	3
Glass	214	9	7

Table3: shows the standard features of these data[8]

dataset	(MSE)	standard deviation	Time in seconds
Iris	2.43	4.87	6.5
Wine	36.36	2.00	
Glass	54.30	20.03	

Table 4: Shows the (MSE) and (SD) & Time for PSO[8]

dataset	(MSE)	standard deviation	Time in seconds
Iris	2.14	1.09	3.6
Wine	23.80	1.73	
Glass	18.29	1.00	

Table 5: Shows the (MSE) &(SD)&Time for BA[8]

Table 4 and Table 5 show that BA is better than PSO algorithm in the classification of dataset Iris and Wine and plant Glass and achieve scale error and a standard deviation less than the PSO algorithm during a few time of training.

5) Tables 6 illustrates that, from the training success rate (the success times within 50 training times) aspect, GA optimized RBF algorithm is superior to the traditional RBF algorithm; from the training error and test error aspect, RBF and GA-RBF-L algorithm are equivalent, or slightly better than GA-RBF algorithm; from the operation time aspect, the operation time of GA optimized RBF algorithm is slightly longer, because running the genetic algorithm will take longer time; from the recognition precision aspect, the GA-RBF-L algorithm's classification precision is the best.

Neural networks algorithm	Traditional RBF	GA-RBF	GA-RBF-L
Training success rate, %	86	100	100
Training error	0.22	0.36	0.29
Test error	1.78	1.97	1.61
Number of hidden neurons	44	28	28
Operation time, s	1.21	1.62	1.62 + 0.22
Classification accuracy, %	89	87	97

Table 6: Comparison of the performance of each algorithm for waveform database

References

[1] Tuba kurban , Erkon besdog,"A Comparison of RBF Neural Network Training Algorithms for Inertial Sensor Based Terrain Classification", *Sensors (Basel)* ,2009,9(8),pp-6312-6329.

[2] Yuxiang Shao , Qing Chen, and Hong Jiang," RBF Neural Network Based on Particle Swarm Optimization", *Springer-Verlag Berlin Heidelberg ,2010,pp-169-176*.

[3] Weikuan Jia, Dean Zhao, Tian Shen, Chunyang Su, Chanli Hu, and Yuyan Zhao," A New Optimized GA-RBF Neural Network Algorithm ",*Computational Intelligence and Neuroscience,Volume 2014 (2014)*, *Article ID 982045, 6 pages*

[4] Ruba Talal ," Comparative Study between the (BA) Algorithm and (PSO) Algorithm to Train (RBF) Network at Data Classification**",** *International Journal of Computer Applications (0975 – 8887) Volume 92 – No.5, April 2014.*

[5] Chen, S., Grant, P.M., Cowan, C.F.N," Orthogonal Least-Squares Algorithm for Training Multi output Radial Basis Function Networks", *IEEE Transactions on Neural Networks 6, 378–384 (1992).*

[6] Knopf, G.K., Sangole, A," Interpolating Scattered Data Using 2D Self-Organizing Feature Maps", *J. Graphical Models 66, 50–69 (2004).*

[7] Dr.laheeb M. Ibrahim, Hanan H. Ali," Using of Neocognitron Artificial Neural Network To

Recognize handwritten Arabic numbers". *The first scientific conference for information technology - Iraq - Mosul University 22 to 23 December 2008.*

[8] Nazri Mohd. Nawi, Abdullah Khan, and Mohammad Zubair Rehman:" A New Back-Propagation Neural Network Optimized with Cuckoo Search Algorithm". *Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia (UTHM).SPRINGER. pp. 413–426, 2013.*

[9] Diptam Dutta, Argha Roy, Kaustav Choudhury." Training Artificial Neural Network using Particle Swarm Optimization Algorithm". *International Journal of Advanced Research in Computer Science and Software Engineering. Volume 3, Issue 3, March 2013.*

[10] Horng, M.H., Performance Evaluation of Multiple Classification of the Ultrasonic Supraspinatus Image by using ML, RBFNN and SVM Classifier, Expert Systems with Applications, *2010, Vol. 37, pp. 4146-4155.*

[11] Wu, D., Warwick, K., Ma, Z., Gasson, M. N., Burgess, J. G., Pan, S. & Aziz, T.," A Prediction of Parkinson's Disease Tremor Onset Using Radial Basis Function Neural Networks, Expert Systems with Applications", *Vol .37, pp.2923-2928.*

[12] Feng, M.H. & Chou, H.C. "Evolutional RBFNs Prediction Systems Generation in the Applications of Financial Time Seriies Data. Expert Systems with Applications", Vol.38, 8285-8292.