Improving Performance of Radial Basis Function Network based with Particle Swarm Optimization

Sultan Noman Qasem & Siti Mariyam Hj. Shamsuddin
Soft Computing Research Group
Faculty of Computer Science and Information System
University Technology Malaysia
Skudai, Johor, Malaysia
sultanoman@yahoo.com, mariyam@utm.my

Abstract—in conventional RBF Network structure, different layers perform different tasks. Hence, it is useful to split the optimization process of hidden layer and output layer of the network accordingly. This study proposes hybrid learning of RBF Network with Particle Swarm Optimization (PSO) for better convergence, error rates and classification results. The hybrid learning of RBF Network involves two phases. The first phase is a structure identification, in which unsupervised learning is exploited to determine the RBF centers and widths. This is done by executing different algorithms such as k-mean clustering and standard derivation respectively. The second phase is parameters estimation, in which supervised learning is implemented to establish the connections weights between the hidden layer and the output layer. This is done by performing different algorithms such as Least Mean Squares (LMS) and gradient based methods. The incorporation of PSO in hybrid learning of RBF Network is accomplished by optimizing the centers, the widths and the weights of RBF Network. The results for training, testing and validation of five datasets (XOR, Balloon, Cancer, Iris and Ionosphere) illustrate the effectiveness of PSO in enhancing RBF Network learning compared to conventional Backpropogation.

Keywords-component; Hybrid learning; Radial basis function network; K-means; Least mean squares; Backpropogation; Particle swarm optimization; Unsupervised and supervised learning.

I. INTRODUCTION

Radial Basis Function (RBF) Networks form a class of Artificial Neural Networks (ANNs), which has certain advantages over other types of ANNs. It has three layers feed forward fully connected network, which uses RBFs as the only nonlinearity in the hidden layer neurons. The output layer has no nonlinearity and the connections of the output layer are only weighted, the connections from the input to the hidden layer are not weighted [1]. RBF Networks have been widely applied in many science and engineering fields. It is a feedback network of three layers, where each hidden unit implements a radial activation function and each output unit implements a weighted sum of hidden units' outputs. Its training procedure is usually divided into two stages. The first stage includes determination of centers and widths of the RBFs by clustering algorithms such as K-means. The second stage involves weights establishment by connecting the hidden layer with the output layer. This is determined by algorithms such as Least Mean Squares [2]. Clustering algorithms have been successfully used in training RBF Networks to determine the centers and widths of RBFs. In most traditional algorithms, such as the K-means, the number of cluster centers need to be predetermined, which restricts the real applications of the algorithms [4].

In this paper, PSO is explored to enhance hybrid learning of RBF Network. The paper is structured as follows. In Section II, RBF Network model and parameter selection is introduced. Section III presents related work about RBF Network training. BP-RBF Network model is given in section IV. Section V describes PSO algorithm. Section VI describes our proposed approach of hybrid learning of RBF Network. Sections VII and VIII gives the experiments setup, results and validation results of the proposed model on datasets respectively. The comparison of classification accuracy between PSO-RBF Network and BP-RBF Network is presented in section IX and finally, the paper is concluded in Section X.

II. ARCHITECTURE OF RBF NETWORK

RBF Network is structured by embedding radial basis function in hidden layer and it has three layers feed-forward neural network. Such a network is characterized by a set of inputs and a set of outputs. It is used in function approximation, time series prediction, and control, and the network architecture is constructed with three layers: input layer, hidden layer, and output layer. The input layer is made up of source nodes that connect the network to its environment. The second layer, the only hidden layer of the network, applies a non-linear transformation from the input space to a hidden space. The nodes in the hidden layer are associated with centers that determine the behavior structure of network. The response from the hidden unit is activated through RBF using Gaussian function or other functions. The output layer provides the response of the network to the activation pattern of the input layer that serves as a summation unit.

The model is given by the following equation for the jth output $y_j(i)$:

$$y_j(i) = \sum_{k=1}^{K} w_{jk} \Phi \left[ x(i), c_k, \sigma_k \right]$$  \hspace{1cm} (1)
Where \(j=1,2,\ldots,n\), \(n\) is the number of output nodes in the output layer, \(i=1,2,\ldots,N\), \(N\) is the number of input nodes in the input layer, \(K\) is the number of RBFs used, and \(C_1 \in \mathbb{R}^m, \sigma_k \in \mathbb{R}^m\), are the center value vector, the width value vector of RBFs and \(\{w_{jk}|k=1,2,\ldots,K\}\) are the weights of RBFs connected with the \(j^{th}\) Output, respectively.

Fig. 1 shows the structure of RBF Network. Each term \(\Phi(.)\) forms the activation function in a unit of the hidden layer. The output layer then implements a linear combination of this new space.

\[
\Phi(x, c_k, \sigma_k) = \prod_{i=1}^{m} \phi(x_i, c_{ki}, \sigma_{ki})
\]

(2)

Moreover, the most popular choice for \(\phi(.)\) is the Gaussian Function form defined as

\[
\phi(x_i, c_{ki}, \sigma_{ki}) = \exp\left(-\frac{(x_i - c_{ki})^2}{2 \sigma_{ki}^2}\right)
\]

(3)

In (3), \(\sigma_k\) indicates the width of the \(k^{th}\) Gaussian RBF functions. One of the \(\sigma_k\) selection methods is shown as follows:

\[
\sigma_k^2 = \frac{1}{m_k} \sum_{x_i \in \Theta_k} \|x_i - c_k\|
\]

(4)

Where \(\Theta_k\) is the \(k^{th}\) cluster of training is set and \(m_k\) is the number of sample data in the \(k^{th}\) cluster.

Figure 1. Structure of RBF Network

There are three types of parameters in RBF model with Gaussian basis functions: RBF centers (hidden layer neurons); Widths of RBFs (standard deviations in the case of a Gaussian RBF); and Output layer weights.

III. RELATED WORK

Although there are many studies in RBF Network learning but research on leaning of RBF Network with PSO is still fresh. This section presents some existing work of RBF Network learning based on Evolutionary Algorithms (EAs) such as PSO especially based on unsupervised learning only (Clustering).

In [11], a PSO learning algorithm has been proposed to automate the design of RBF Networks, to solve pattern classification problems. Thus, PSO-RBF finds the size of the network and the parameters that configure each neuron: center and width of its basis function. Supervised mean subtractive clustering algorithm has been proposed [3] to evolve RBF Networks and the evolved RBF (function not network) acts as fitness evaluation function of PSO algorithm for feature selection. The method performs feature selection and RBF learning simultaneously. PSO algorithm has been introduced [2] to train RBF Network related to automatic configuration of network architecture related to centers of RBF. Two learning algorithms were compared. One was PSO algorithm. The other was newrb routine that was included in Matlab neural networks toolbox as standard training algorithm for RBF Network.

A hybrid PSO (HPSO) has been proposed [13] with simulated annealing and Chaos search technique to train RBF Network. The HPSO algorithm combined the strong ability of PSO, SA, and Chaos. An innovative Hybrid Recursive Particle Swarm Optimization (HRPSO) learning algorithm with normalized fuzzy c-mean (NFCM) clustering, PSO and Recursive Least Squares (RLS) has been presented [15] to generate RBF Networks modeling system with small numbers of descriptive RBFs for fast approximating two complex and nonlinear functions. On other hand, a newly evolutionary search technique called Quantum-Behaved Particle Swarm Optimization, in RBF Network learning has been used [14]. The proposed QPSO-Trained RBF Network was test on nonlinear system identification problem.

Unlike previous studies that mentioned earlier, this research shares consideration of parameters of RBF (unsupervised learning) which are centers and length of width or spread of RBFs with different algorithms such as K-means and K-nearest neighbors or standard deviations algorithms respectively. However, training of RBF Network need to enhance with PSO to optimize the centers and widths values which are obtained from the clustering algorithms and PSO also used to optimize the weights which connect between hidden layer and output layer (supervised learning). Also this paper has been presented to train, test and validate the PSO-RBF Network on the datasets.

IV. BP-BASED TRAINING RBF NETWORK

In our paper, the standard BP is selected as the simplest and most widely used algorithm to train feed-forward RBF Networks and considered for the full-training paradigm; customizing it for half-training is straightforward and can be done simply by eliminating gradient calculations and weight-updating corresponding to the appropriate parameters. The following is the procedure of BP-RBF Network algorithm:

1) Initialize network.
2) Forward pass: Insert the input and the desired output; compute the network outputs by proceeding forward through the network, layer by layer.
3) Backward pass: Calculate the error gradients versus the parameters, layer by layer, starting from the output layer and proceeding backwards.
4) Update parameters: (weight, center and width of the RBF Network respectively).
   
   
   \[ w_{jk}(t+1) = w_{jk}(t) + \Delta w_{jk}(t+1) \]
   
   \[ \Delta w_{jk}(t+1) = \eta_j \delta_j o_k \]
   
   With
   
   \[ \delta_j = o_j (1 - o_j)(t_j - o_j) \]
   
   \[ o_k = \exp \left(-\frac{(x - c_k)^2}{2\sigma_k^2}\right) \]  

   Where \( w_{jk}[t] \) is the weight from node j to node k at time t, \( \Delta w_{jk} \) is the weight adjustment, \( \eta_j \) is the learning rate, \( \delta_j \) is error at node j, \( o_j \) is the actual network output at node k, \( o_j \) is the actual network output at node j and \( t_j \) is the target output value at node j.

   
   \[ c_{ki}(t+1) = c_{ki}(t) + \Delta c_{ki}(t+1) \]
   
   \[ \Delta c_{ki}(t+1) = \eta_j \delta_j w_{jk} / \sigma_k^2 (x_k - c_{ki}) \]
   
   Where \( c_{ki}[t] \) is the centre from node k to node i at time t, \( \Delta c_{ki} \) is the center adjustment, \( \eta_j \) is the learning rate, \( \delta_j \) is error at node j, \( o_j \) is the actual network output at node k, \( \sigma_k \) is the width at node k, \( w_{jk} \) is the weight connected between node k and j and \( x_k \) is the input node i to node k.

   
   \[ \sigma_k(t+1) = \sigma_k(t) + \Delta \sigma_k(t+1) \]
   
   \[ \Delta \sigma_k(t+1) = \eta_j \delta_j w_{jk} / \sigma_k^2 (x_k - c_{ki})^2 \]
   
   Where \( \sigma_k[t] \) is the width of node k at time t, \( \Delta \sigma_k \) is the width adjustment, \( \eta_j \) is the learning rate, \( \delta_j \) is error at node j, \( o_j \) is the actual network output at node k, \( \sigma_k \) is the width at node k, \( w_{jk} \) is the weight connected between node j and k, \( x_k \) is the input at node i and \( \eta_j, \eta_j, \eta \) are learning rate factors in the range [0, 1].

5) Repeat the algorithm for all training inputs. If one epoch of training is finished, repeat the training for another epoch.

RBF is Radial Basis Function in each node of the hidden layer and has three different activation functions e.g. Gaussian Function. MLP is Multi-Layer Perceptron which is a kind of Artificial Neural Networks (ANNs). BP-RBF Network doesn’t need the momentum term as it is common for the MLP. It does not help in training of the RBF Network [14].

V. PARTICLE SWARM OPTIMIZATION

Particle Swarm Optimization (PSO) algorithm, originally introduced by Kennedy and Eberhart in 1995 [5], simulates the knowledge evolvement of a social organism, in which each individual is treated as an infinitesimal particle in the ndimensional space, with the position vector and velocity vector of particle i being represented as \( X_i(t) = (X_{i1}(t), X_{i2}(t), \ldots, X_{in}(t)) \) and \( V_i(t) = (V_{i1}(t), V_{i2}(t), \ldots, V_{in}(t)) \). The particles move according to the following equations:

\[ X_i(t+1) = X_i(t) + V_i(t+1) \]

\[ V_i(t+1) = w_i V_i(t) + c_1 r_1 (p_i(t) - X_i(t)) + c_2 r_2 (g(t) - X_i(t)) \]  

Where \( c_1 \) and \( c_2 \) are the acceleration coefficients, Vector \( P_i = (P_{i1}, P_{i2}, \ldots, P_{in}) \) is the best previous position (the position giving the best fitness value) of particle i known as the personal best position (pbest); Vector \( P_g = (P_{g1}, P_{g2}, \ldots, P_{gn}) \) is the best position among the personal best positions of the particles in the population and is known as the global best position (gbest). The parameters \( r_1 \) and \( r_2 \) are two random numbers distributed uniformly in (0, 1). Generally, the value of \( V_i \) is restricted in the interval \([-V_{max}, V_{max}]\). Inertia weight \( w \) was first introduced by Shi and Eberhart in order to accelerate the convergence speed of the algorithm [6].

VI. PSO-BASED TRAINING RBF NETWORK

PSO has been applied to improve RBF Network in various aspects such as network connection (centers, weights), network architecture and learning algorithm. The main process in this paper is to employ PSO-based training algorithm on center, width and weight of RBF network, and investigate the efficiency of PSO in enhancing RBF Network training. Every single solution of PSO called (a particle) represents a single RBF Network and flies over the solution space in search for the optimal solution. The particles are evaluated using a fitness function to seek the optimal solution. Particles have dimension which is the set of connections (centers, widths) values are initialized from the k-means algorithm while (weights, bias) values are initialized randomly or from LMS algorithm. The particles are updated accordingly using (13) and (14).

The procedure for implementing PSO global version (gbest) is given in the next pseudo code. The fitness value of each particle (member) is the value of the error function.
evaluated at the current position of the particle. The pseudo code of the procedure is as follows:

```plaintext
For each particle do
    Initialize particle position and velocity
End for

While stopping criteria are not fulfilled do
    For each particle do
        Calculate fitness value (MSE in RBF Network)
        If fitness value is better than best fitness value pBest in particle history then
            Set current position as pBest
        End if
    End for
    Choose as gBest the particle with best fitness value among all particles in current iteration
    For each particle do
        Calculate particle velocity based on (13).
        Update particle position (center, width and weight) based on (14).
    End for
End while
```

VII. EXPERIMENTS

A. Experimental Setup

The experiments of this work included the standard PSO and BP for RBF Network training. For evaluating all of these algorithms we used five benchmark classification problems obtained from the machine learning repository [9]. The parameters of the PSO algorithm were set as: weight w decreasing linearly between 0.9 and 0.4 [6]. The population size used by PSO was constant. The algorithm stopped when a predefined number of iterations have been reached. Values selected for parameters were shown in Table I. The $C_1$ and $C_2$ constant was set to 2.0 as suggested by [3] [11] [16].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population Size</td>
<td>20</td>
</tr>
<tr>
<td>Iterations</td>
<td>10000</td>
</tr>
<tr>
<td>$W$</td>
<td>[0.9,0.4]</td>
</tr>
<tr>
<td>$C_1$</td>
<td>2.0</td>
</tr>
<tr>
<td>$C_2$</td>
<td>2.0</td>
</tr>
</tbody>
</table>

The XOR data set (3 features and 8 examples) is a logical operation on three operands that results in a logical value of true if and only if one of the operands but not both have a value of true. The Balloon data set (4 features and 16 examples) is used in cognitive psychology experiment. There are four data sets representing different conditions of an experiment. All have the same attributes. The Iris dataset (4 features and 150 examples) is used for classifying all the information into three classes. The Ionosphere dataset (34 features and 351 examples) is radar data was collected by a system in Goose Bay; Labrador is used for classifying all the information into “Good” or “Bad” results. Finally, the cancer dataset (9 features and 699 examples) is related to the diagnosis of breast cancer in benign or malignant.

By trial and error, the settings and parameters of RBF Network are identified. The architecture of the RBF Network was fixed in one hidden layer (number of inputs of the problem - 2 hidden units - 1 output units) in XOR, Balloon, Cancer, Ionosphere and (number of inputs of the problem - 3 hidden units - 3 output units) in Iris dataset. The parameters of the experiments are described in Table II.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train data</td>
<td>XOR Balloon Cancer Iris Ionosphere</td>
</tr>
<tr>
<td>Test data</td>
<td>3 4 175 120 251</td>
</tr>
<tr>
<td>Validation data</td>
<td>8 16 175 150 351</td>
</tr>
<tr>
<td>Input dimension</td>
<td>3 4 9 4 34</td>
</tr>
<tr>
<td>Output neurons</td>
<td>1 1 1 3 1</td>
</tr>
<tr>
<td>Network Structure</td>
<td>3-2-1 4-2-1 9-2-1 4-3-3 34-2-1</td>
</tr>
</tbody>
</table>

The number of maximum iterations is set differently to bind the number of forward propagations to $4 \times 10^4$. The maximum iterations in BP is set to $2 \times 10^4$ (number of forward propagations $= 2 \times$ maximum number of iterations), while the maximum number of iterations in PSO is set to 10000 (number of forward propagations $= $ swarm size $\times$ maximum number of iterations) [8].

B. Experimental Results

This section presents the results of the study on PSO-trained RBF Network and BP-trained RBF Network. The experiments are conducted by using five datasets: XOR, Balloon, Iris, Ionosphere and Cancer. The results for each dataset are compared and analysed based on the convergence, error and classification performance.

a) XOR Dataset

A connective in logic known as the "exclusive or" or exclusive disjunction is a logical operation on three operands. Two algorithms used to train and test of RBF Network. The stopping conditions of PSO-RBFN are set as minimum error of 0.005 or maximum iteration of 10000. On the other hand, the stopping conditions for BP-RBFN are set as the minimum error of 0.005 or the iterations have reached to 20000. The results for PSO and BP for RBFN are illustrated in Table III and Figure 2.

From Table III, PSO-RBFN converges at 93 iterations compared to BP-RBFN with 5250 iterations for the whole learning process. Both algorithms are converged with given minimum error. For the classification, it shows that BP-RBFN is better than PSO-RBFN with 93.88% compared to 93.51%.
However, PSO-RBFN converges faster compared to BP-RBFN. However, the classification rate for testing of XOR problem is not good due to smaller amount of data to be learned by the network. Figure 2 illustrates that PSO-RBFN significantly reduces the error with minimum iterations compared to BP-RBFN.

TABLE III. RESULT OF BP-RBFN AND PSO-RBFN ON XOR

<table>
<thead>
<tr>
<th></th>
<th>BP-RBFN</th>
<th>PSO-RBFN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Iteration</td>
<td>5250</td>
<td>1</td>
</tr>
<tr>
<td>Error Convergence</td>
<td>0.00500</td>
<td>0.32972</td>
</tr>
<tr>
<td>Classification (%)</td>
<td>93.88</td>
<td>64.72</td>
</tr>
</tbody>
</table>

$b)$ Balloon Dataset

This data is used in cognitive psychology experiment. There are four data sets representing different conditions of an experiment. All have the same attributes. It contains 4 attributes and 16 instances. The stopping conditions of PSO-RBFN are set to a minimum error of 0.005 or maximum iteration of 10000. Conversely, the stopping conditions for BP-RBFN are set to the minimum error of 0.005 or the network has reached maximum iteration of 20000.

TABLE IV. RESULT OF BP-RBFN AND PSO-RBFN ON BALLOON

<table>
<thead>
<tr>
<th></th>
<th>BP-RBFN</th>
<th>PSO-RBFN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Iteration</td>
<td>20000</td>
<td>3161</td>
</tr>
<tr>
<td>Error Convergence</td>
<td>0.01212</td>
<td>0.23767</td>
</tr>
<tr>
<td>Classification (%)</td>
<td>91.27</td>
<td>75.41</td>
</tr>
</tbody>
</table>

$c)$ Iris Dataset

The Iris dataset is used for classifying all the information into three classes which are iris setosa, iris versicolor, and iris virginica. The classification is based on its four input patterns which are sepal length, sepal width, petal length and petal width. Each class refers to type of iris plant contain 50 instances. For Iris dataset, the minimum error of PSO-RBFN is set to 0.05 or maximum iteration of 10000. While, the minimum error for BP-RBFN is set to 0.05 or the network has reached maximum iteration of 20000. Table V shows that BP-RBFN is better than PSO-RBFN with an accuracy of 95.66% compared to 95.48%. However, PSO-RBFN converges faster at 3774 iterations compared to 10162 iterations in BP-RBFN.

TABLE V. RESULT OF BP-RBFN AND PSO-RBFN ON IRIS

<table>
<thead>
<tr>
<th></th>
<th>BP-RBFN</th>
<th>PSO-RBFN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Iteration</td>
<td>10162</td>
<td>3774</td>
</tr>
<tr>
<td>Error Convergence</td>
<td>0.05000</td>
<td>0.04205</td>
</tr>
<tr>
<td>Classification (%)</td>
<td>95.66</td>
<td>95.48</td>
</tr>
</tbody>
</table>

For Iris learning, both algorithms converge using the maximum number of pre-specified iteration. PSO-RBFN takes 3774 iterations to converge at a minimum error of 0.0499949 while minimum error for BP-RBFN is 0.05000 with 10162 iterations. Figure 4 shows that PSO-RBFN reduces the error with minimum iterations compared to BP-RBFN.
d) Ionosphere Dataset

This radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. For Ionosphere problems, the stopping conditions for BP-RBFN is minimum error of 0.05 or maximum iteration of 20000. The minimum error of PSO-RBFN is 0.05 or maximum iteration of 10000. The experimental results for PSO-based RBFN and BP-based RBFN are shown in Table VI and Figure 5.

![Figure 5. Convergence of Ionosphere dataset](image)

| TABLE VI. RESULT OF BP-RBFN AND PSO-RBFN ON IONOSPHERE |
|----------------|----------------|
|                | **BP-RBFN**    | **PSO-RBFN**  |
| **Training**   | **Testing**    | **Training**  | **Testing**  |
| Learning Iteration | 20000          | 5888          |
| Error Convergence  | 0.18884        | 0.0499999     |
| Classification (%) | 62.27          | 87.24         |

In Ionosphere learning process, Table VI shows PSO-RBFN takes 5888 iterations compared to 20000 iterations in BP-RBFN to converge. In this experiment, PSO-RBFN is managed to converge using minimum error at iteration of 5888, while BP-RBFN trapped at the local minima and converges at a maximum iteration of 20000. For the correct classification percentage, it shows that PSO-RBFN result is better than BP-RBFN with 87.24% compared to 62.27%. Figure 5 shows PSO-RBFN significantly reduce the error with small number of iterations compared to BP-RBFN.

e) Cancer Dataset

The purpose of the breast cancer data set is to classify a tumour as either benign or malignant based on cell descriptions gathered by microscopic examination. It contains 9 attributes and 699 examples of which 485 are benign examples and 241 are malignant examples. The first 349 examples of the whole data set were used for training, the following 175 examples for validation, and the final 175 examples for testing [8]. The ending conditions of PSO-RBFN are set to minimum error of 0.005 or maximum iteration of 10000. Alternatively, the stopping conditions for BP-RBFN are set to a minimum error of 0.005 or maximum iteration of 20000 has been achieved.

![Figure 6. Convergence of Cancer dataset](image)

| TABLE VII. RESULT OF BP-RBFN AND PSO-RBFN ON CANCER |
|----------------|----------------|
| **Training**   | **Testing**    |
| Learning Iteration | 20000          | 10000         |
| Error Convergence  | 0.03417        | 0.0181167     |
| Classification (%) | 92.80          | 97.65         |

In Cancer learning process, from Table VII shows PSO-RBFN takes 10000 iterations compared to 20000 iterations in BP-RBFN to converge. In this experiment, PSO-RBFN is managed to converge at iteration 10000, while BP-RBFN converges at a maximum iteration of 20000 and illustrates that PSO-RBFN is better than BP-RBFN with an accuracy of 97.65% and 92.80%. Figure 6 shows PSO-RBFN significantly reduce the error with small number of iterations compared to BP-RBFN.

VIII. VALIDATION RESULTS

In artificial neural network methodology, data samples are divided into three sets; training, validation and testing in order to obtain a network which is capable of generalizing and performing well with new cases. There is no precise rule on the optimum size of the three sets of data, although authors agree that the training set must be the largest. Validations are motivated by two fundamental problems either in model selection or in performance estimation.

To create a N-fold partition of the dataset we simplifies that for each of N experiments, use N-1 folds for training and the remaining one for testing and the true error is estimated as
the average error rate. The results demonstrate the evaluation of our algorithms with respect to the convergence rate on the training and testing dataset.

**TABLE VIII. VALIDATION RESULT OF BP-RBFN AND PSO-RBFN ON ALL DATASET**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BP-RBFN</th>
<th>PSO-RBFN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Training</td>
<td>Testing</td>
</tr>
<tr>
<td>XOR</td>
<td>0.11332</td>
<td>0.32864</td>
</tr>
<tr>
<td>Balloon</td>
<td>0.06004</td>
<td>0.33155</td>
</tr>
<tr>
<td>Iris</td>
<td>0.05000</td>
<td>0.06227</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>0.20743</td>
<td>0.22588</td>
</tr>
<tr>
<td>Cancer</td>
<td>0.03046</td>
<td>0.04233</td>
</tr>
</tbody>
</table>

These results for BP-RBFN and PSO-RBFN on all dataset are shown in Table VIII. On the whole data, the experiments showed that PSO-RBFN gives high performance for RBF Network training. PSO-RBFN reduces the error with minimum iterations compared to BP-RBFN.

**IX. COMPARISON BETWEEN PSO-RBF NETWORK AND BP-RBF NETWORK**

This analysis is carried out to compare the results between PSO-RBFN and BP-RBFN. The learning patterns for both algorithms in both experiments are compared using a five datasets. The classification results for all datasets are shown in Figure 7.

For Balloon, Cancer and Ionosphere dataset, the results show that PSO-RBFN is better in terms of convergence rate and correct classification. PSO-RBFN converges in a short time with high classification rates. For XOR and Iris dataset, both algorithms converge to the solution within specified minimum error; it shows that at this time, BP-RBFN classifications are better than PSO-RBFN. But in terms of convergence rate, it shows that PSO-RBFN is better than BP-RBFN, and PSO-RBFN significantly reduces the error with minimum iterations.

For overall performance, the experiments show that PSO-RBFN produces feasible results in terms of convergence rate and classification accuracy.

**X. CONCLUSION**

This paper proposes PSO based Hybrid Learning of RBF Network to optimize the centers, widths and weights of network. Based on the results, it is clear that PSO-RBFN is better than BP-RBFN in terms of convergence and error rate and PSO-RBFN reached optimum because it reduces the error with minimum iteration and obtains the optimal parameters of RBF Network. In PSO-RBFN, network architecture and selection of network parameters for the dataset influence the convergence and the performance of network learning.

In this paper, both the algorithms need to be used the same network architecture. Choosing PSO-RBFN parameters also depend on the problem and dataset to be optimized. These parameters can be adjusted accordingly to achieve better optimization. However, to have better comparison, the same parameters for all five datasets have been used. For BP-RBFN, the learning rate which is critical for standard BP network is provided with a set of weight. This is to ensure the convergence time is faster with better results. Although Standard BP learning becomes faster based on those parameters, the overall process including parameters selection in BP-RBFN takes lengthy time compared to the process in PSO-RBFN.

**ACKNOWLEDGMENT**

This work is supported by Ministry of Higher Education (MOHE) under Fundamental Research Grant Scheme (Vote No.78243). Authors would like to thank Research Management Centre (RMC) Universiti Teknologi Malaysia, for the research activities and Soft Computing Research Group (SCRG) for the support and incisive comments in making this study a success.

**REFERENCES**


