Enhanced Self Organizing Map (SOM) and Particle Swarm Optimization (PSO) for Classification

Shafaatunnur Hasan[1], Siti Mariyam Shamsuddin[2], and Bariah binti Yusob[3]

***Abstract*— Hybrid technique for Self Organizing Map and Particle Swarm Optimization approach is commonly implemented in clustering area. In this paper, a hybrid approach that is based on Enhanced Self Organizing Map and Particle Swarm Optimization (ESOM/PSO) for classification is proposed. Enhanced Self Organization map which based on Kohonen network structure is to improve the quality of the data classification and labeling. New formulation of hexagonal lattice area is used for the enhancement Self Organizing Map structure. The proposed hybrid ESOM/PSO algorithm uses PSO to evolve the weights for ESOM. The weights are trained by ESOM in the first stage. In the second stage, they are optimized by PSO. In the proposed algorithm, the result is measured by using a classification accuracy and quantization error techniques.**

***Keywords*— Enhanced Self Organizing Map (SOM), Particle Swarm Optimization (PSO), classification.**

I. INTRODUCTION

Classification is a process of separating a large class of objects into smaller classes, together with the criterion for determining classes of the objects. This process can be very complicated due to the difficulty of identifying the criteria accordingly, especially whenever it involves complex data structures. Therefore, artificial intelligence techniques have been introduced by many researchers, such as neural network, fuzzy set, genetic algorithm, swarm intelligence and rough set. Kohonen Self Organizing Map (SOM) which is one of the neural network techniques has been implemented extensively in this study. Learning enhancement of SOM is explored to improve the quality of data classification and labeling by proposing a new formulation of the hexagonal lattice area. However, larger grid size in SOM means increase in training time. The larger is the lattice, the more nodes should be considered for BMU calculation, thus leading to higher operating cost for the algorithm [1]. In order to solve this problem, the training weight of SOM algorithm need to be optimizes by Particle Swarm Optimization (PSO).

Early studies show that combination of SOM-PSO approach. First approach is comes from Shi Eberhart with modified particle swarm optimizer [2]. In [3], O'Neill and Brabazon adopt PSO as unsupervised SOM algorithm. The algorithm produces competitive result for data classification. However, the author suggests applying different distance metric in calculating the distance between input vectors and each member of the swarm. Different swarm sizes and different velocity update formulations can be implemented in future studies.

PSO also has been used for clustering purpose. In [4][5], Xiao et. al use PSO to refine the weight vectors for a SOM obtained after an initial application of a standard SOM training methodology. The research design for this study is quite similar to this method. The difference is this study used enhanced SOM algorithm for classification purpose.

Recently, Ozift *et al* proposed PSO in the optimization of SOM algorithm to reduce the training time without quality loss in clustering [1]. The author stated that the size of lattice is related to the clustering quality of SOM. This optimization technique successfully by reducing the numbers of nodes that finds the BMU for a particular input.

The proposed supervised SOM with Particle Swarm Optimization (PSO) is tested on different types of data, and these include cancer, iris and XOR data. There are 5 sections in the study. Section 1 is the introduction of PSO and SOM. In section 2 explain the related previous work in SOM-PSO. In section 3, we explain the method and algorithm for ESOM and PSO. In Section 4 we present the results and discuss about the application of the techniques under study. Finally, Section 5 presents some concluding remarks.

II. PROPOSED METHODOLOGY

In this section, we describe the research design that has been conducted in this study.

A. Dataset

The dataset is required to represent the problem. A universal data has been used for training of the network which are Iris, Cancer and XOR. dataset can be downloaded from UCI Machine Learning database [6].

[1]Shafaatunnur Hasan was in Universiti Teknologi Malaysia (e-mail: shafaatunnur@gmail.com).

[2]Siti Mariyam Shamsuddin is professor at Department of Computer Graphics & Multimedia, Faculty of Computer Science & Information Systems, Universiti Teknologi Malaysia (e-mail: mariyam@utm.my/ sitimariyams@gmail.com).

[3]Bariah binti Yusob is Lecturer at Faculty of Computer Systems & Software Engineering, Universiti Malaysia Pahang (e-mail: bariahyusob@ump.edu.my).

Iris

IRIS dataset is the most famous real data benchmark in machine learning. IRIS dataset was proposed by Fisher in 1936 . This dataset is formed by 150 points that belong to three different classes. One class is linearly separable from the other two, but the other two are not linearly separable from each other. Since the dimension of IRIS dataset is 4, IRIS dataset is usually represented by projecting the data along their principal components.

Cancer

The Cancer dataset requires the decision maker to correctly diagnose breast lumps as either benign or malignant based on data from automated microscopic examination of cells collected by needle aspiration. The dataset includes nine inputs and one output. The exemplars are split with 599 for training, and 100 for testing, totaling 699 exemplars. All inputs are continuous variables and 65.5% of the examples are benign. The data set was originally generated at hospitals at the University of Wisconsin Madison, by Dr. William H. Wolberg. In this study, 150 data patterns are used in both algorithms.

XOR

A connective in logic known as the "exclusive or" or exclusive disjunction is a logical operation on two operands that results in a logical value of true if and only if one of the operands but not both has a value of true. XOR is a basic dataset that widely use to train and test NN. In this study, 8 data patterns are used in both algorithms.

B. Enhanced Self Organizing Map (ESOM)

Self Organizing Map (SOM) first introduced by von der Malsburg (1973) and presented by Professor Teuvo Kohonen, (1982). The goal of SOM is to convert a high dimensional input signal into a simpler low dimensional discrete map. SOM are based on competitive learning, where the output nodes compete among themselves to be winning node and the only node to be activated by a particular input observation [7]. Lattice structure is important to determine the quality of Kohonen map. This is because weight for each neuron in the neighborhood will be updated. Hexagonal lattice structure is good for image processing. The structure can make the image pixel uniform to each other [8].

Therefore, a new formulation of hexagonal lattice area is proposed in equation 1. The traditional hexagonal lattice is shown in equation 2. A neighborhood is hexagonal where Influence *N(j,t)*, is used instead of width of neighborhood, *N(j)* since *D(t)* is a threshold value decreased via a schedule as training progresses. For this neighborhood function the distance is determined considering the distance in the lattice in each dimension, and the one having the maximum value is chosen as distance a node from BMU, *d(j)*, *N(j)* corresponds to a hexagonal around n_{win} having neighborhood width as:

1) Right border x = WinNode x;
2) Right border y =WinNode y + R_{new} ;
3) Left_border x = WinNode x;
4) Left_border y = WinNode y - R_{new} ;

with $R_{new} = (2r+1)^2 + 2r^2$ (1)

$$R = 3r\sqrt{\left(r^2\right) - \left(\frac{1}{4}r^2\right)} \quad (2)$$

Where R_{new} = Enhanced hexagonal lattice, R = traditional hexagonal lattice.

$$N(j,t)\begin{cases} 1 \rightarrow d(j) \le D(t) \\ 0 \rightarrow d(j) > D(t) \end{cases} \quad (3)$$

The weights of all neuron within this hexagon are updated with *N(j)*=1, while the others remaining unchanged. As the training progresses, this neighborhood gets smaller and smaller, resulting in that only the neurons very close to the winner are updated towards the end of the training. The training end as remains no more neuron in the neighborhood. Usually, the neighborhood function, *N(j)* is chosen as an L-dimensional Gaussion function as equation below:

$$N(j,t) = \exp\frac{-d(j)^2}{2\sigma(t)^2} \quad (4)$$

The algorithm shows below:

1) Initialization
   Set initial synaptic weights to small random values, say in a interval [0,1], and assign a small positive value to the learning rate parameter
2) Competition.
   For each output node j, calculate the value $D(V-W_j)$ of the scoring function. For example, for Euclidean distance, $D(V-W_j) = \sqrt{\sum_{i=0}^{n}\left(V_i - W_{ij}\right)^2}$ (1)
   Find the winning node J that minimizes $D(V-W_j)$ overall output nodes.
3) Cooperation.
   Identify all output nodes *j* within the neighborhood of *J* defined by the neighborhood size *R*. For these nodes, do the following for all input records fields. Reduce the radius with exponential decay function:
   $$\sigma(t) = \sigma_0 \exp\left(-\frac{1}{\lambda}\right), t = 1,2,3,... \quad (2)$$
   Where σ_0 = initial radius λ = maximum iteration, t = current iteration.
   New formulation of hexagonal lattice:
   $$R_{new} = (2r+1)^2 + 2r^2 \quad (3)$$
   Where r = neighborhood radius
4) Adaptation
   Adjust the weights:
   $$W(t+1) = W(t) + \Theta(t)L(t)V(t) - W(t) \quad (4)$$
   Where L = learning rate, Θ = influence a node's distance from the BMU.

$$L(t) = L_0 \exp\left(-\frac{t}{\lambda}\right), t = 1, 2, 3, ... \quad (5)$$

Where L_0 = initial learning rate.

$$\Theta(t) = \exp\left(-\frac{dist^2}{2\sigma^2(t)}\right), t = 1, 2, 3, ... \quad (6)$$

Where *dist* = distance a node from BMU, σ = width of neighborhood.

5) Iteration
Adjust the learning rate and neighborhood size, as needed until no changes occur in the feature map. Repeat to step (2) and stop when the termination criteria are met.

C. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is one of the Swarm Intelligence (SI) technique that inspired by social behavior of bird flocking and fish schooling. The pioneers of the PSO algorithm are James Kennedy and Russell C. Eberhart in 1995. PSO is a global optimization, population based evolutionary algorithm for dealing with problems in which a best solution can be presented as a point or surface in an n-dimensional space. Hypotheses are plotted in this space and seeded with an initial velocity, as well as a communication between the particles.

To explain how the algorithm works in solving an optimization problem, suppose that we are trying to choose D continuous variables *x1*, . . . , *xD* to maximize a function

$$f(x1, ..., xD) \quad (7)$$

Suppose also that we create a swarm of *i* = 1, . . . ,N particles. At all points in time, each particle *i* have:

1) A current position *Xi* or *Xn = (xi1, . . . , xiD)*
2) A record of the direction it followed to get to that position *Vi* or *Vn= (vi1, . . . , viD)*
3) A record of its own best previous position P_{best} = (P_{best} *1, . . . ,* P_{best} *D)*
4) A record of the best previous position of any member in its group g_{best} = (g_{best} *1, . . .* g_{best} *D)*

Given the current position of each particle, as well as the other information, the problem then becomes one of determining the direction of change for the particles. As mentioned above, this is done by reference to each particle's own experience and the experience of other members of its group. Its own experience includes the direction it came from *Vi* and its own best previous position. The experience of others is represented by the best previous position for any member in its group. This suggests that each particle might move in:

1) The same direction that it came from *Vi*
2) The direction of its best previous position $P_{best} - Xi$
3) The direction of the best previous position of any member in its group $g_{best} - Xi$.

The algorithm supposes that the actual direction of change for particle *i* will be a weighted combination of these

$$V_n = W \cdot V_n + C1 * rand1 * \left(G_{best,n} - X_n\right) + C2 * rand2 * \left(P_{best,n} - X_n\right) \quad ... (8)$$

Where r1 and r2 are uniform [0,1] random numbers, c1 > 0 and c2 > 0 are constants called the *cognitive* and *social* parameters and w > 0 is a constant called the *inertia* parameter. For their part, n and n+1 index successive periods (generations). Given the direction of change, the new position of the particle will simply be:

$$X_n = X_n + V_n \quad (9)$$

Given initial values for *Xi*, *Vi*, P_{best} and g_{best}, equations (8) and (9) will determine the subsequent path that each particle in the swarm will follow.

III. RESULT AND DISCUSSION

The experiments are conducted on XOR, Cancer and Iris dataset. The weights are trained by ESOM in the first stage. In the second stage, they are optimized by PSO. The results are validated by executing the convergence errors and quantization errors. Convergence error describes how accurately the particle tends towards a stable position (velocity tends towards zero) [9]. Quantization error describes how accurately the neurons respond to the given dataset. For example, if the reference vector of the BMU calculated for a given testing vector xi is exactly the same xi, the error in precision is then 0. Table 1 depicts the quality of map and classification accuracy of SOM and ESOMPSO. The table shows that ESOMPSO is better than standard SOM.

A. Results on XOR Dataset

The network size that has been used to train the XOR problem consists of 3 input layer nodes, 2-d mapping layer for output nodes. 8 data patterns used to train the network. For PSO parameters, C1 and C2 = 2, $\Delta t = 0.1$, the minimum value of weight is 0.40 and the maximum value are 0.90. The population of particles was set as 10 and problem dimension as 100 (a 10*10 grid structure) maximum iteration of 10385. The experimental results are shown in Table 1 and Figure 1.

TABLE 1. RESULT OF ESOMPSO AND SOM ON XOR DATASET

	ESOMPSO	SOM
Learning Iteration	78	388
Convergence Error	0.0048279	0.00499
Quantization Error	01916	0.2060
Classification (%)	97.49	96.73

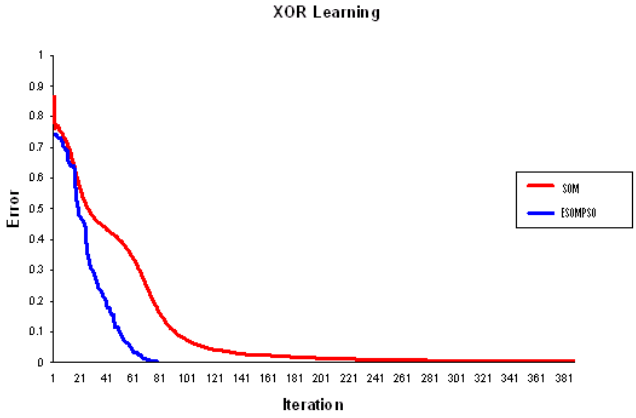

Figure 1. Convergence of XOR dataset

From Table 1, correct classification percentage shows that ESOMPSO result is better than SOM with 97.49 % compared to 96.73%. Figure 1 shows the learning process where both algorithms attempt to reach the learning stop condition. In ESOMPSO, particles work together to find the lowest error (gbest) at each iteration and consistently reduce the error at each iteration. While in SOM, it seems that the error is decreasing constantly when it reach the 341 iterations, and stop at a specified condition on the last iteration.

B. *Results on Cancer Dataset*

For Cancer problems, 380 data patterns have been used where the network size consists of 9 nodes in the input layer, 2-d mapping layer for output nodes. For PSO parameters, C1 and C2 = 2, Δt = 0.1, The minimum value of weight is 0.40 and the maximum value are 0.90. The population of particles was set as 10 and problem dimension as 100 (a 10*10 grid structure) maximum iteration of 10385. The experimental results are shown in Table 2 and Figure 2.

TABLE 2. RESULT OF ESOMPSO AND SOM ON CANCER DATASET

	ESOMPSO	SOM
Learning Iteration	489	118
Convergence Error	0.00497508	0.00495
Quantization Error	0.4422	0.4924
Classification (%)	99.77	99.69

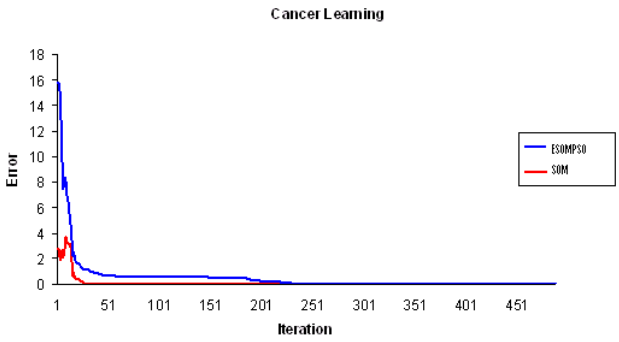

Figure 2. Convergence of Cancer dataset

In Cancer learning process, the correct classification percentage shows that ESOMPSO result is better than SOM with 99.77% compared to 99.69%.

C. *Results on Iris Dataset*

The network architecture used for Iris dataset consists of 4 input nodes and 2-d mapping layer for output nodes. 120 data patterns used to train the network. For PSO parameters, C1 and C2 = 2, Δt = 0.1, the minimum value of weight is 0.40 and the maximum value are 0.90. The population of particles was set as 10 and problem dimension as 100 (a 10*10 grid structure) maximum iteration of 10000. The experimental results are shown in Table 3 and Figure 3.

TABLE 3. RESULT OF ESOMPSO AND SOM ON IRIS DATASET

	ESOMPSO	SOM
Learning Iteration	10000	1000
Convergence Error	1.88831	4.18404
Quantization Error	0.0243	0.0318
Classification (%)	97.72	92.11

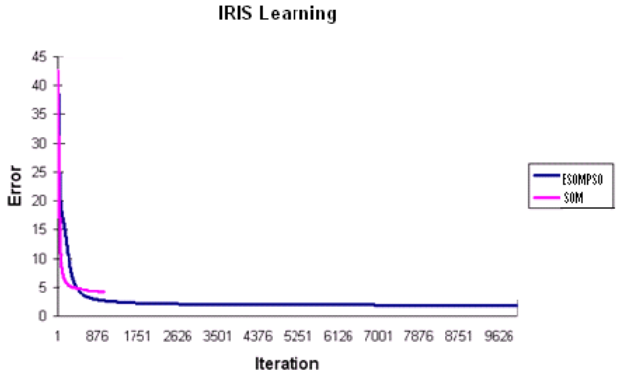

Figure 3. Convergence of Iris dataset

For Iris learning, both algorithms converge using the maximum number of pre-specified iteration. SOM converge at minimum error of 4.18404 while minimum error for ESOMPSO is 1.88831 at 10000 iterations. Table 4.3 shows that classification of ESOMPSO is better than SOM with 97.72% compared to 92.11%.

D. *Comparison between ESOMPSO and SOM*

This analysis is carried out to compare the results between ESOMPSO and SOM. To do this, the learning patterns for both algorithms are compared using all three datasets. The comparative correct classification percentage for all datasets is shown in Figure 4.

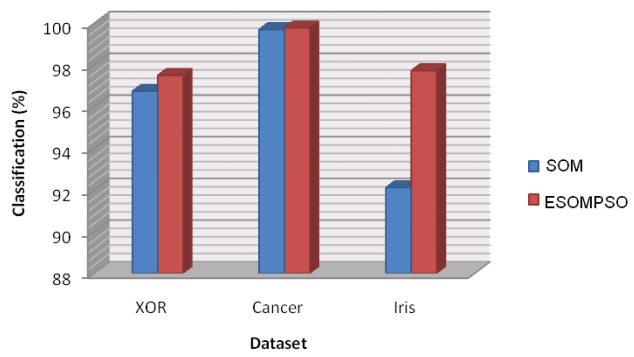

Figure 4. Comparative of correct classification percentage between ESOMPSO and SOM.

The results show that ESOMPSO has better results on convergence error and correct classification percentage. For overall performance, the experiments show that ESOMPSO produces feasible results in terms of quantization error, convergence error and classification percentage.

IV. CONCLUSION

The study being developed to reach it main objective that is to optimize the training weight of ESOM by integrating Particle Swarm Optimization. The project was carried out to analyze the optimization algorithm of PSO and ESOM to explore the classification accuracy and convergence rate compared to the standard Self Organization Map. Based on the result derived from the training of datasets, it is clear that ESOMPSO is better than standard SOM in term of classification accuracy percentage and convergence rate. Different distance measure such as Manhattan distance, Chebyshev distance and Bray Curtis distance can be used in ESOMPSO for future studies.

AKNOWLEDGMENT

Authors would like to thank Research Management Centre (RMC) Universiti Teknologi Malaysia, for the research activities, Soft Computing Research Group (SCRG) for the support in making this study a success.

REFERENCES

[1] Ozift, A., et al. “Swarm Optimized Organizing Map (SWOM): A swarm Intelligence based Optimization of Self Organizing Map”. Expert Systems with Applications (2009). doi:10.1016/i.eswa.2009.02.051.

[2] Shi, Y. and Eberhart, R. C (1998). A modified particle swarm optimizer. Proceedings of the IEEE International Conference on Evolutionary Computation pp. 69-73. IEEE Press, Piscataway, NJ

[3] Brabazon, A. and O’Neill, M. (2006). Biologically Inspired Algorithms for Financial Modelling, Berlin: Springer.

[4] Xiao, X., Dow, E.R., Eberhart, R., Miled, Z.B., Oppelt, R.J. (2004). A hybrid self-organizing maps and particle swarm optimization approach. Concurrency and Computation: Practice and Experience, 16(9):895-915.

[5] Xiao, X., Dow, E.R., Eberhart, R., Miled, Z.B., Oppelt, R.J. (2003). Gene-Clustering Using Self-Organizing Maps and Particle Swarm Optimization, in Proceedings of the IEEE International Parallel and Distributed Processing Symposium (IPDPS), 22-26 April 2003,Nice, France, IEEE Press.

[6] UCI Machine Learning Repository [Online]. Available: http://www.ics.uci.edu/~mlearn/MLRRepository.html

[7] Haykin, S (1999). Neural Networks : A Comprehensive Foundation. (2nd ed). Upper Saddle River, New Jersey.

[8] Bariah (2007). Pembaikan Struktur Kekisi Heksagon dalam Pembelajaran Rangkaian Kohonen, master’s thesis, Dept. Computer Science and Information Systems., University Technology Malaysia.

[9] C.Maurice (2006). Particle Swarm Optimization.(1st ed). Great Britain