Journal of Computer Science and Cybernetics, V.30, N.3 (2014), 278–290
DOI:10.15625/1813-9663/30/3/4080

HIERARCHY SUPERVISED SOM NEURAL NETWORK APPLIED FOR CLASSIFICATION PROBLEM

LE ANH TU[1], NGUYEN QUANG HOAN[2], LE SON THAI[1]

[1] *University of Information and Communication Technology;*
latu@ictu.edu.vn, lesonthai@gmail.com
[2] *Posts and Telecommunications Institute of Technology.*
quanghoanptit@yahoo.com.vn

Tóm tắt. Trong bài báo này chúng tôi đề xuất các mạng nơron SOM có giám sát, gồm S-SOM và S-SOM$^+$ áp dụng cho bài toán phân lớp. Các mạng này được cải tiến từ các mô hình SOM không giám sát và có giám sát đã được đề xuất bởi Kohonen và các tác giả khác. Sau đó, chúng tôi tiếp tục đề xuất các mô hình SOM có giám sát phân tầng cải tiến từ S-SOM và S-SOM$^+$, gọi là HS-SOM và HS-SOM$^+$. Cải tiến của chúng tôi xuất phát từ ý tưởng xác định các nơron phân loại mẫu sai, từ đó phát triển các nhánh huấn luyện bổ sung đối với các mẫu dữ liệu được đại diện bởi các nơron này. Chúng tôi đã tiến hành thực nghiệm trên 11 tập dữ liệu phân lớp đơn nhãn đã được công bố. Kết quả thực nghiệm cho thấy mô hình đề xuất của chúng tôi phân loại mẫu đạt mức độ chính xác từ 92% tới 100%.

Từ khóa. Bản đồ tự tổ chức, học có giám sát, phân cụm dữ liệu, Kohonen, mạng nơron nhân tạo.

Abstract. In this paper, supervised SOM neural network was suggested, with S-SOM and S-SOM$^+$ applied for classification problems. These networks were developed from supervised and unsupervised SOM model by Kohonen and other researchers. Hierarchy supervised SOM models were developed from the S-SOM and S-SOM$^+$, called HS-SOM and HS-SOM$^+$. Our improvement was inspired by the idea of finding neurons that wrongly classify samples, which created extra training branches for the representative samples of these neurons. Experiments on 11 single-label classification datasets were executed. The results showed that the suggested model classified samples with high accuracy, from 92% to 100%.

Keywords. Self-organizing map, supervised learning, clustering, classification, Kohonen, neural network.

1. INTRODUCTION

Up to now, the conventional multivariate statistical techniques (cluster analysis, linear discriminant analysis) including unsupervised (Kohonen's network) and supervised (Bayesian network) artificial neural networks were compared for as general tools for the classification and identification problem [18–22]. One of those is the Self-Organizing Map (SOM) which was proposed by Tevou Kohonen [14]. SOM is a feedforward neural network using an unsupervised learning algorithm. It allows mapping data from multidimensional space to less dimensional space (normally two-dimension). The SOM structure consists of input signal and Kohonen layer (output). After training, Kohonen layer displays data features or feature map. In which, data with close features will be represented by the same neuron or neighboring neurons. To observe this feature map, visual techniques were used [31], for instance, visualization using U-matrix [11]. However, visual techniques did not determine to which kinds data belongs.

Unsupervised SOM is normally applied for data clustering problems [4, 6]. Essentially, this is a grouping of Kohonen's neurons because each neuron (after training process) is the representative of one or some patterns. In case, datasets are unlabeled, grouping is based on the differences of neurons' features (weight vector), for example, forming groups based on an agglomerative algorithm [30] or using splitting threshold [9]. In contrast, the dataset is labeled (single-label), grouping can be based on data's labels [3, 12]. Nonetheless, it is impossible to confirm that if the clustering results are optimized or not. This is because that SOM network used unsupervised learning algorithm. Consequently, clustering results are normally used to observe and analyze data features.

When applying SOM for classification (single label training datasets), the accuracy was not high. In fact, a neuron can be assigned many distinct labels. That means this neuron cannot classify samples. The unsupervised SOM experiment was conducted to classify Iris dataset [17]. The classification accuracy was only from about 75.0% to 78.35%. There were some reasons for this. Firstly, the network has not been trained completely since the network's initial parameters are not suitable. This is the challenge of the neural network in general and SOM network in particular, since the choice of parameters is often based on experience from trying-error. Secondly, the nature of unsupervised learning only updates input without updating expected output. That means features map is formed naturally from input data without orientating or adjusting of expected output data. This made labels assigned wrongly to neurons. To solve this problem, the supervised SOM model should be used. That means network needs training with both input samples and their corresponding labels.

Recently, some supervised SOMs have been proposed. These models are often called supervised Kohonen networks, containing CPN (Counter Propagation network) [5, 7, 10, 13], SKN (Supervised Kohonen Network) [8, 14, 16], XYF (X–Y Fused Network) and BDK (Bi-Directional Kohonen network) [15]. Whereas, unsupervised SOM only updates input samples (signed X) to create features map of input data, supervised Kohonen updates both input samples (X) and output sample (Y) to form two features maps of input data (Xmap) and of output data (Ymap). This allows supervised Kohonen to represent the single or double dimension relationships between input and output data. Consequently, it is suitable with problems of identifying output sample from one unknown input and vice versa, for example, forecasting, controlling and voice recognition. Supervised SOM models are presented in section 2.

Another SOM's disadvantage is that the map's size must be defined in advance and suitable with the data set. However, to the large data sets, it is very difficult to choose a correct size. Growing and hierarchical unsupervised SOM models were proposed to solve this problem. For instance, GSOM (Growing SOM) [23] grow the map's size in the training process. HSOM (Hierarchical SOM) [24] is the structure layer model (the number of layers and dimension of maps is defined a priori). GHSOM (Growing Hierarchical SOM) [25–28], GHTSOM (Growing Hierarchical Tree SOM) [29] are the hybrid of GSOM and HSOM, which grow both dimensions of maps and hierarchy based on the quantization error. In the paper [2], we proposed top-down hierarchical tree structure which has map's size in the same branch reduced gradually and clusters are separated in detail from upper layers to lower layers by decreasing gradually the split threshold. In fact, both the quantization error and the split threshold are identified based on the dissimilar features of data. The common characteristic of the above models is that nodes are unsupervised SOMs. Therefore, the main object of the hierarchical unsupervised SOM is to represent the hierarchy of the data.

In this paper, two hierarchies supervised SOM models are proposed to apply for classification problems, called HS-SOM and HS-SOM$^+$. These two models are trained and correct error using architecture hierarchy tree [2], where each node is a supervised SOM, called S-SOM (or S-SOM$^+$ developed from S-SOM). To upgrade the effectiveness of clustering, S-SOM and S-SOM$^+$ have the capability of identifying neurons that wrongly classify samples. Each error unit (neuron) of parent node creates a child node which is trained by its corresponding wrong classified samples.

To prove the effectiveness of the suggestion, some experiments on the assumed datasets XOR [15] and 10 real-world datasets [32] were conducted. The classification results were correct from 92% to 100%.

In comparison with hierarchical unsupervised SOM, hierarchical supervised SOM models proposed in this paper have three main differences. Firstly, the main object of hierarchical supervised SOM is classifying data. Secondly, clustering based on output data (labels) since each node of hierarchical structure is a supervised SOM. Thirdly, lower layer's nodes are formed for extra training for the units which classify samples incorrectly (this is the network's supervising).

The rest of the paper includes: part 2 presents the overview of unsupervised and supervised Kohonen networks; part 3 displays the hierarchy supervised SOM network; part 4 shows the experimental results and finally some comments, evaluation of the suggested solution are presented.

2. UNSUPERVISED AND SUPERVISED SOM NETWORKS

2.1. Self-organizing Map (SOM)

The SOM neural network includes an input signal layer and an output layer called Kohonen layer. Kohonen layer is often organized as a two dimensional matrix of neurons. Each unit i (neuron) in the Kohonen layer is attached to a weight vector $w_i = [w_{i1}, w_{i2}, .., w_{in}]$, where n is the input vector size, w_{ij} is the weight of neuron i corresponding to the input j. Network training process is repeated several times, at iteration t, three steps are done:

- *Step 1*- determining the BMU: randomly select an input v from the data set, determine c neuron that has the smallest distance function ($dist$) in the Kohonen matrix (frequently use functions Euclidian, Manhattan or Vector Dot Product). c neuron is called Best Matching Unit (BMU).

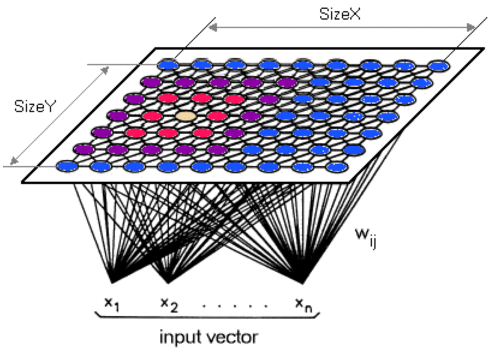

Figure 1: Illustration of SOM

$$dist = ||v - w_c|| = \min_i\{||v - w_i||\} \tag{1}$$

- *Step 2*- defining the neighboring radius of the BMU: $N_c(t) = N_0 \exp\left[-\frac{t}{\lambda}\right]$ is an interpolation function of radius (decreasing as the numbers of iterations), where N_0 is the initial radius; time constant $\lambda = \frac{K}{\log(N_0)}$, where K is the total number of iterations.
- *Step 3*- updating the weights of the neurons in the neighboring radius of the BMU in a trend closer to the input vector v:

$$w_i(t+1) = w_i(t) + N_c(t)h_{ci}(t)[v - w_i(t)] \tag{2}$$

where, $h_{ci}(t)$ is the interpolation function over learning times, shows the effect of distance to the learning process, can be calculated by the formula $h_{ci}(t) = \exp\left[-\frac{\|r_c - r_i\|^2}{2N_c^2(t)}\right]$ where r_c and r_i position of the neuron c and neuron i in the Kohonen matrix.

Obviously, the learning process only updates input data without updating expected corresponding output data, so that the Kohonen feature map is in fact an input data feature map. Therefore, the application of unsupervised SOM for classification problem is not effective.

The supervised SOM models developed from the unsupervised SOM model will be presented in the next sections.

2.2. CPN network

The CPN neural network (Counter Propagation Network) is in fact an enlarged SOM network [5, 10]. Besides Kohonen layer, the network was attached an extra output Ymap with the same Kohonen layer's size (Figure 2). Kohonen (Xmap) is still trained with an unsupervised algorithm as SOM model. In the training process, with each input sample and its corresponding output sample couple (X, Y), Xmap and Ymap are updated simultaneously. In which, BMU and its neighbors on Xmap are updated with X vector, and the neurons on the corresponding position on Ymap are updated with Y vector in the same SOM model's way. This allows CPN presenting one-way relationship between input X and output Y [13]. That means the output is formed from input, whereas, the input formation is not affected by output. Therefore, CPN is considered pseudo-supervised. CPN is normally applied for forecasting and controlling [7, 10].

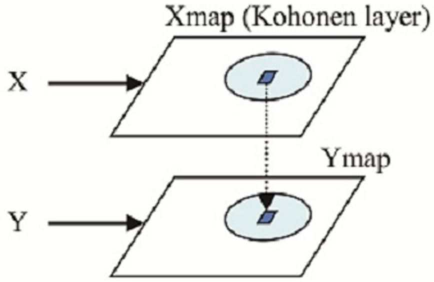

Figure 2: Illustration of CPN

2.3. SKN network

The SKN network (Supervised Kohonen Network) is the supervised SOM model by Kohonen [14]. SKN is in fact SOM model, but its inputs re-adjust. During training, input vector X and its corresponding output vector Y are connected together to make up a common input vector XY. Therefore, the weight vector of each neuron in the Kohonen layer has the same size as XY. However, to get a better feature map, the rate of the features of X and Y needs considering. When identifying an unknown X sample, X is only compared to the corresponding part in the weight vector of each neuron.

Kohonen used SKN to recognize voice [14]. The input included 34 features made up from two vectors x_s and x_u. Where, x_s is 15-component short-time acoustic spectrum vector computed over 10 milliseconds; x_u is the corresponding phonemic vector of x_s, containing 19 features.

2.4. XYF and BDK networks

The XYF (XY Fused) was proposed by W. Melssen [15]. This was developed from CPN network (Figure 3). It improved the way to define BMU. Melssen created a fuse similar matrix (fused matrix) of X and Y with Xmap and Ymap. With each input couple (X_i, Y_i), the value of unit k of the fused matrix, signed $S_{Fused}(i, k)$ calculated as (3):

$$S_{Fused}(i,k) = \alpha(t)S(X_i, Xmap_k) + (1 - \alpha(t))S(Y_i, Ymap_k) \tag{3}$$

where, $k = 1..m \times n$, with m, n are the size of Xmap and Ymap, $S(X_i, Xmap_k)$ is the similarity measure of X_i with unit k on Xmap, $S(Y_i, Ymap_k)$ is the similarity measure of Y_i with unit k on Ymap, $\alpha(t)$ is linear time-reducing.

BMU on both Xmap and Ymap are neurons which have a corresponding location of the smallest element of fused matrix. Units on both Xmap and Ymap (including BMU, and neurons in the neighboring radius) are updated simultaneously.

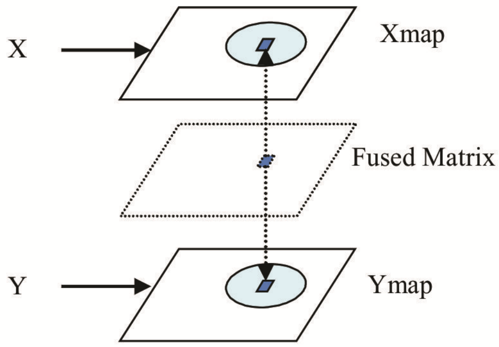

Figure 3: Illustration of XYF

Melssen suggested another improved BDK model from XYF (Figure 4). BDK used two fused similar matrixes, Xmatrix and Ymatrix, to define distinct BMUs on Xmap and Ymap instead of using a common fused matrix as XYF. BMU of Xmap is identified based on the smallest unit on Ymatrix and vice versa.

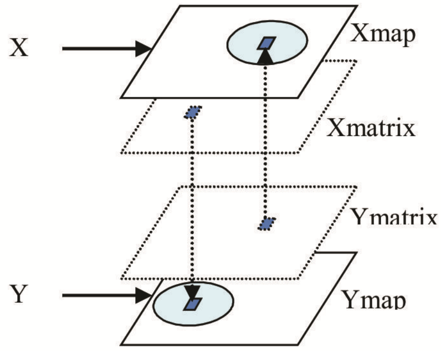

Figure 4: Illustration of BDK

With each input couple (X_i, Y_i), first calculate units of Ymatrix by (4):

$$S_{Ymatrix}(i,k) = (1-\alpha(t))S(X_i, Xmap_k) + \alpha(t)S(Y_i, Ymap_k) \tag{4}$$

Next, update Xmap with BMU position defined by Ymatrix and calculate the units of Xmatrix by (5):

$$S_{Xmatrix}(i,k) = \alpha(t)S(X_i, Xmap_k) + (1-\alpha(t))S(Y_i, Ymap_k) \tag{5}$$

After that, update Ymap with BMU position defined by Xmatrix. So, Xmap and Ymap updated alternately, which is different from XYF model.

3. HIERARCHY SUPERVISED SOM (HS-SOM)

The HS-SOM was developed from hierarchy training ideas presented in the paper [2]. HS-SOM is a hierarchy training tree (Figure 7), where each node is an S-SOM. Each child node was trained by the samples which were wrongly classified, featured by an error unit of the parent node.

3.1. S-SOM

S-SOM was similar to CPN, containing an input Kohonen layer (Xmap) and a same size output Ymap layer (label map) $m \times n$ (Figure 5). Ymap presents the label distribution of data, or called label map. Each training sample couple (X, Y), X was updated to Xmap and Y was updated to Ymap. S-SOM's training algorithm consists of two steps:

- *Step 1*: Finding BMU of X on Xmap and then update X to BMU, and the neurons in the neighboring radius of BMU using an unsupervised learning algorithm (section 2.1). Simultaneously, on Ymap, assign the Y label to units which have a corresponding location to BMU, and neighbor of BMU (note, the final labels of each element on Ymap are the last labels assigned). Figure 5 illustrated Ymap with four assigned labels A, B, C (labels in the training data set) and E (the error label updated in step 2). In fact, these labels should use numeric values.
- *Step 2*: On Xmap, defining neurons which wrongly classified samples by re-checking all training data set once more, with each input X:
  - Find BMU on Xmap. Add X to sample set characterized by BMU (signed IBMU).
  - If Y (label of X) does not match with a Ymap's unit which has a corresponding position to BMU, assign E value to this unit (E is a special label which does not belong to the dataset used to mark the unit that wrongly classified samples). Neurons with label E need extra training. Extra training technique is presented in the section 3.3.

If there exists $Y_i \in Ymap$ which is not assigned in the learning process, Y_i is assigned a label following agglomerative algorithm [30]. That means $Y_i = Y_j$ (with $Y_j \in Ymap$), if it satisfies the conditions: weight vector of neuron N_i is the closest to weight vector of neuron N_j (with $N_i, N_j \in Xmap$), and Y_j is assigned a label, where $i, j = 1, 2, ..., m \times n$.

3.2. S-SOM$^+$

Obviously, the step one of the S-SOM's training algorithm is similar to the CPN. That means Xmap only updated input X without expecting output Y. Therefore, S-SOM was improved by replacing a Kohonen layer (Xmap) with SKN layer. The replacing model was called S-SOM$^+$ (Figure 6). S-SOM$^+$ can be considered as a hybrid model between CPN and SKN.

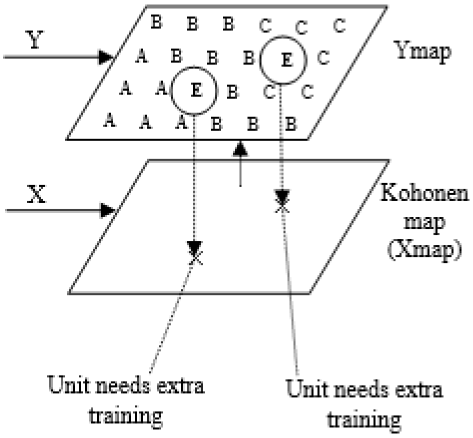

Figure 5: Illustration of S-SOM

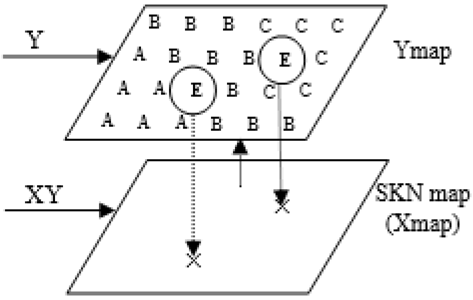

Figure 6: Illustration of S-SOM$^+$

The training algorithm of S-SOM$^+$ is only adjusting step one of S-SOM. The input of SKN map is a combination vector of X and Y, signed XY. Weight vector of each neuron on Xmap has the size of XY. According to Kohonen, to increase the supervising, the rate between the number of features of X and Y can be adjusted. So, p Y labels can be added to the end of the feature set of X. The more p increases, the higher tendency of the same label samples gathers.

3.3. HS-SOM and HS-SOM$^+$

Initially, HS-SOM has only one S-SOM (root node) with the size of $n \times n$. The root node was trained by the all samples in the dataset (signed I). After training, I was divided into s subsets $I = \{I_1, I_2, ..., I_s\}$, where I_i (with $i = 1..s$) was the subset represented by neuron N_i of Xmap (with $s = n \times n$).

Each error unit (neuron) N_k on Xmap (element k on Ymap assigned with label E) formed a child node and this child node was trained by I_k represented by neuron N_k with $k = 1, 2, ..., s$. Each child node continued developing new child node in the same way of parent.

Each node development of HS-SOM conducted as the rules:

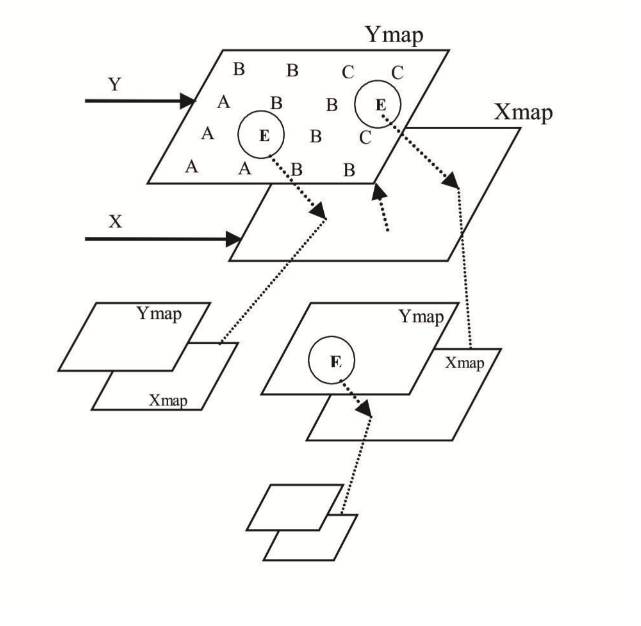

Figure 7: Illustration of HS-SOM

- If $|I_{child}| \neq |I_{parent}|$, a child node formed with a more reducing size than its parent's following the formula (6):

$$n_{child} = ceil\left(\left(\frac{|I_{child}|}{|I_{\text{parent}}|}\right)^{\beta} n_{\text{parent}}\right) \tag{6}$$

  where, *ceil ()* is a round up function; |.| represents the number of elements in a set; the parameter β warrants child node's size which cannot be so smaller than the parent node.

- Whereas, if there exists $|I_{child}| = |I_{parent}|$, re-initialize the current node with more suitable size and weight (current node is the node trained by I_{parent}). This only occurs when the current node is not the root node, and the number of training elements is small and quite close. Therefore, the current node's size was adjusted approximately with the number of labels in I_{parent} following the formula:

$$n_{\text{parent}} = ceil(\sqrt{q}) \tag{7}$$

  where, q is the total of labels in I_{parent}. The weight vector of each neuron in this node is initialized using the input vector providing that vector samples with different labels are updated to the network.

HS-SOM will be more effective if the S-SOM is replaced by S-SOM$^+$. This replacing model is called HS-SOM$^+$. The experimental results are displayed in the next section.

4. EXPERIMENTAL RESULTS

The experiments on both HS-SOM and HS-SOM$^+$ with 11 issued labeled datasets were executed, including: XOR, Iris, Glass, Aggregation, Compound, Pathbased, Spiral, D31, R15, Jain, and Flame. In which, XOR is an assumed dataset [15], with feature values belonging to the domain of [0,1]. The remaining datasets are real-world data [32], with feature values taken from the real world (not in the same domain).

When cross-checking method applied, two parameters were used to evaluate the capability of classification [1]: *Precision and Sensitivity*

$$Precision = TP/(TP + FP) \tag{8}$$

$$Sensitivity = TP/(TP + FN) \tag{9}$$

where, *TP - True Positive* (for instance, A is defined A), *FP - False Positive* (for instance: not A but be defined A), *FN – False Negative* (for instance: A but not be defined A). Hence, *precision* stands for the correct defining rate A in the total of samples considered as A and *sensitivity* stands for the correct defining rate A in the total of samples A needed defining.

Each dataset was divided into five parts. Testing was conducted five times; each time parts swapped their roles. In which, four parts were trained and the last part was tested. Each test calculated two values *precision* and *sensitivity* for each cluster. Therefore, the final *precision* and *sensitivity* values of each cluster were the average values of five tests. *Precision* and *sensitivity* values of a dataset were the average values of the *precision* and *sensitivity* of all clusters. Initial parameters were defined in Table 1.

Parameter	HS-SOM	HS-SOM$^+$
Root node's initial size	15×15	15×15
Vector weight's size	$\|X\|$	$\|X\| + p$
Initial radius N_0	15	15
Formula (1)	Euclidian	Euclidian
β	0.3	0.3
p		$\|X\|$

Table 1: Initial parameters

Table 2 showed the experimental results of HS-SOM and HS-SOM$^+$ on the experimental datasets when using the initial parameters as in Table 1.

The *precision* and *sensitivity* values were calculated by percent; values could change ±2% per each test (because the weight vector of each neuron initialized randomly in the domain of dataset samples). In all tests, HS-SOM$^+$ had higher results than the HS-SOM with about 5%. Especially, with four datasets XOR, Iris, Jain, R15, HS-SOM$^+$ model classified 100% fully and exactly all samples (when applying CPN, SKN, XYF, BDK models on the XOR, the accuracy is from 85% to 92% [15]). With the remaining datasets, the classification result can be improved by normalizing features to the same value domain. Moreover, with each problem and dataset, it is necessary to identify the importance of each feature to integrate a corresponding adjusting parameter when computing the distance function (1). For example, a Euclidian distance function is adjusted as (10).

$$dist(v, \mathrm{w}) = \sqrt{\sum_{i=1}^{k} \mu_i (v_i - \mathrm{w}_i)^2} \tag{10}$$

where, *dist(v,w)* is the distance between vector v and vector w; μ_i is the adjusting parameter to feature i.

Dataset	Samples	Features	Clusters	HS-SOM		HS-SOM$^+$	
				Precision	Sensitivity	Precision	Sensitivity
XOR	4125	2	2	99.9275	99.9274	100	100
Iris	150	4	3	95.7239	95.3333	100	100
Glass	214	9	6	94.0933	92.9563	98.0556	98.1250
Flame	240	2	2	97.3396	97.2729	99.4737	99.6667
Pathbased	300	2	3	97.1393	97.0603	99.4203	99.3159
Spiral	312	2	3	96.9171	96.7619	96.9554	96.7937
Jain	373	2	2	99.8246	99.4737	100	100
Compound	399	2	6	94.4959	93.2910	97.7822	96.2963
Aggregation	788	2	7	98.2602	98.2926	99.5918	99.9160
R15	600	2	15	98.3392	98.1667	100	100
D31	3100	2	31	95.7267	95.5161	99.9693	99.9677

Table 2: Experimental results

p value was continuously increased and applied to un-exact 100% data sample classification. The more p increased, the more classification accuracy increased. With each dataset, when p increased to a certain value, the classification reached 100% (Table 3).

HS-SOM$^+$ was continuously applied for the datasets which has less classification result than 100% with a gradually increasing p. The more p increased, the more classification accuracy increased. Table 3 showed the suitable p values which helped HS-SOM$^+$ classify 100% correctly.

Dataset	Glass	Flame	Pathbased	Spiral	Compound	Aggregation	D31
p	27	11	8	48	7	8	130

Table 3: p values which helped HS-SOM$^+$ classify samples 100% correctly.

In fact, to avoid adding so many labels to sample vector (p is too large), which increased time calculation, an adjusting parameter for the added label should be applied (formula (10)).

5. CONCLUSION

This paper suggested two supervised SOM neural networks applied to the data clustering problem, including S-SOM and S-SOM$^+$. Error detection mechanism and error correction by extra training of these two networks allowed to form two corresponding hierarchical supervised SOMs, HS-SOM and HS-SOM$^+$. These improvements were aimed at overcoming the weak points of unsupervised and supervised SOM models issued by other researchers (section 2). The effectiveness of the suggested models was evaluated through the results presented in section 4. Although the classification results on the datasets were quite high. They can be further improved by standardizing datasets or adjusting the distance function more suitably with each problem. Furthermore, parameter, p contributed greatly to the classification results of HS-SOM^{+}.

As analyzed in the paper [2], the hierarchical architecture allowed data-partition and network-partition for parallel processing. Instead of initializing a network's big-enough size to match the dataset, a smaller size should be chosen (the root node's size is 15×15 for all above experimental datasets). After that, network self-develops necessary child nodes, with reducing the size to assure that network converges. So, an initial size can be chosen easily instead of trying-error many times to get a suitable network size. The size of each small node means shortened processing time. Besides, each node can be installed for parallel processing. Therefore, HS-SOM and HS-SOM^{+} are better at classification and more dominant in time calculating than SOM, CPN, SKN, XYF, and BDK when used on big datasets.

REFERENCES

[1] T. Fawcelt, “An Introduction to ROC Analysis", *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006

[2] Le Anh Tu, Nguyen Quang Hoan, Le Son Thai, “Clustering hierarchical data using SOM neural network", *ICCASA 2012*, LNICST-Springer, vol. 109, pp. 282–289, 2013.

[3] T. Aroui, Y. Koubaa, A. Toumi, “Clustering of the Self-Organizing Map based Approach in Induction Machine Rotor Faults Diagnostics", *Leonardo Journal of Sciences*, ISSN 1583-0233, Issue 15, July-December, pp.1-14, 2009.

[4] Juha Vesanto and Esa Alhoniemi, “Clustering of the Self-Organizing Map", *IEEE Transactions on Neural Networks*, Vol. 11, No. 3, May 2000, 586-600.

[5] R. Hecht-Nielsen,“ Counter propagation networks", *Appl. Opt.* vo.26, no.23, pp. 4979 – 4984, 1987.

[6] J. Ong and S. S. R. Abidi, “Data Mining Using Self-Organizing Kohonen maps: A Technique for Effective Data Clustering & Visualisation", June 28-July 1, in *International Conference on Artificial Intelligence (IC-AI'99)*, Las Vegas, 1999.

[7] G. Carrera and J. Aires-de-Sousa, “Estimation of melting points of pyridinium bromides ionic liquids with decision trees and neural networks", *Green Chem.* vol. 7, no. 1, pp. 20-27, 2005.

[8] E. Bayram, P. Santiago II, R. Harris, Y.D. Xiao, A.J. Clauset, and J.D. Schmitt, “Genetic algorithms and self-organizing maps: a powerful combination for modelling complex QSAR and QSPR problems", *J. Comp. Aid. Mol. Des.* 18 (2004) 483– 493.

[9] Le Anh Tu, Nguyen Quang Hoan, and Le Son Thai, “Improving SOM neural network algorithm applied to the clustering problem color image", Oct 7-8 2011. *In Proceeding of the @2011.* Can Tho-VietNam, Issue 4 pp. 54-63.

[10] J. Zupan M. Novic and I. Ruisánchez, “Kohonen and counterpropagation artificial neural networks in analytical chemistry", *Chemometrics and Intelligent Laboratory Systems* 38 1-23, 1997.

[11] Alfred Ultsch and H. Peter Siemon, Kohonen's self-organizing feature maps for exploratory data analysis. *In Proceedings of the International Neural Network Conference (INNC'90)*, pages 305-308. Kluwer, 1990.

[12] Andreas Rauber, LabelSOM: On the Labeling of Self-Organizing Maps (1999), in *Proc. International Joint Conference on Neural Networks.*

[13] J. A. Freeman, D.M. Skapura, Neural Networks: Algorithms, Applications and Progamming Techniques, *Addison & Wesley*, New York, 1991.

[14] T. Kohonen, Self-Organizing Maps, 3rd Edition. *Springer-Verlag*, Berlin, 2001.

[15] W. Melssen and R. Wehrens and L. Buydens, "Supervised Kohonen networks for classification problems", *Chemometrics and Intelligent Laboratory Systems*, vol. 83 , pp. 99-113, 2006.

[16] Y.D. Xiao, A. Clauset, R. Harris, E. Bayram, P. Santiago II, J. and D. Schmitt, "Supervised self-organizing maps in drug discovery: 1. Robust behavior with overdetermined data sets", *J. Chem. Inf.* (2005) Model.

[17] Le Anh Tu, Le Son Thai, Nguyen Quang Hoan, "Visualizing and data clustering for SOM neural network using improved distance matrix", *Journal of Automation Today in VietNam*, ISSN 1859-0551, vol. 9, pp. 72-77, (2014) .

[18] P. Piraino, A. Ricciardi, G. Salzano, T. Zotta, and E. Parente, "Use of unsupervised and supervised artificial neural networks for the identification of lactic acid bacteria on the basis of SDS-PAGE patterns of whole cell proteins", *Journal of Microbiology Methods*, vol. 66, no. 2, pp. 336-46, 2006.

[19] Narine Manukyan, Margaret J. Eppstein, Donna M. Rizzo, "Data-Driven Cluster Reinforcement and Visualization in Sparsely-Matched Self-Organizing Maps", *IEEE transactions on neural networks and learning systems* 01/2012; 23(5), pp.846-852, 2012.

[20] Y.-H. Jin, A. Kawamura, S-C. Park, N. Nakagawa, H. Amaguchi, and J. Olsson, "Spatiotemporal classification of environmental monitoring data in the Yeongsan River basin, Korea, using self-organizing maps", *Journal of Environmental Monitoring* 09/2011; 13(10), pp. 2886-2894, 2011.

[21] G. Pierris and T. S. Dahl, "Humanoid actile Gesture production using a hierarchical SOM-based encoding, autonomous mental development", *IEEE Transactions*, vol. 6, no. 2, pp. 153 – 167, 2014.

[22] P. Andras, "Function approximation using combined unsupervised and supervised learning, neural networks and learning systems", *IEEE Transactions* vol. 25, Issue: 3, DOI: 10.1109/TNNLS.2013.2276044, pp. 495 – 505, 2014.

[23] D. Alahakoon, S. K. Halgamuge, and B. Sirinivasan, "A self growing cluster develop-ment approach to data mining", *In: Proc. of IEEE Inter. Conf. on Systems*, Man and Cybernetics, 1998.

[24] J. Blackmore and R. Mukkulainen, "Incremental grid growing: Encoding high-dimensional structure into a two-dimensional feature map", *In: Proc. of the IEEE Inter. Conf. on Neural Networks*, 1993.

[25] A. Rauber, D. Merkl, and M. Dittenbach, "The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data", *IEEE TransactionsonNeuralNetworks*, vol. 13, pp. 1331–1341, 2002.

[26] S. Bizzil, R. F. Harrison, D. N. Lerner, "The Growing Hierarchical Self-Organizing Map (GH-SOM) for analysing multi-dimensional stream habitat datasets", In: *18th World IMACS/-MODSIM Congress (2009).*

[27] E.-J1. Palomo, J. North, D. Elizondo, R. M. Luque, and T. Watson, “Application of growing hierarchical SOM for visualisation of network forensics traffic data”, *Neural Networks.* 2012 Aug;32:275-84. doi: 10.1016/j.neunet.2012.02.021. Epub 2012 Feb 14.

[28] P. B. Myszkowski, *Appl. Inf. Inst.*, Wroclaw Univ. of Technol., Wrocław, Poland; Buczek, B., “Growing hierarchical self-organizing map for searching documents using visual content, Computer Science and Information Systems (FedCSIS)", *2011 Federated Conference on IEEE.* 18-21 Sept. 2011, pp. 77 – 81.

[29] Alberto Forti, Gian Luca Foresti, “Growing hierarchical tree SOM: an unsupervised neural network with dynamic topology", *Neural Networks* vol. 19, pp. 1568–1580, 2006.

[30] https://rtmath.net/help/html/29f7cb00-39a1-4fc0-af60-52925f074edd.htm

[31] http://www.ifs.tuwien.ac.at/dm/somtoolbox/visualisations.html

[32] http://cs.joensuu.fi/sipu/datasets/

Received on June 03 - 2014
Revised on August 11 - 2014