References

[1] M. Hewish and L. Ness, *Mine-Detection Technologies*, Int. Defense Rev., Nov. 1995.

[2] M. Bower, E. Cloud, H. Duvoisin, D. Long, and J. Hackett "Development of automatic target recognition for infrared sensor-based close range land mine detector" Martin Marietta Technol. Inc., Orlando, FL, 32812-5837, Tech. Rep. MP-040, 1995.

[3] A. Filippidis, L. C. Jain, and N. Martin, "Fuzzy fusion for surface landmines detection," in *ICONIP'97*, Dunedin, New Zealand, Nov. 1997.

[4] G. Carpenter and S. Grossberg, *Pattern recognition by self-organizing neural networks*. New York: Academic, 1987, pp. 399–410.

[5] K. Fuelop and J. Hall "Thermal infrared landmine detection," Jan. 1996, Tech. Rep. DSTO-TR-0295 AR-009-485, pp. 3–19.

[6] N. Stacy, R. Smith, and G. Nash, "Automatic target recognition for the INGARRA airborne radar surveillance system," D.S.T.O Microwave Radar Div. Int. Rep., Aug. 1994, pp. 1–12.

[7] P. J. Whitbread, *Multispectral Texture*, Ph.D. dissertation, Univ. Adelaide, Australia, Oct. 1992.

[8] R. M. Harlick, K. Shunmuhham, and I. Distein, "Textural features for image classification," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-3, no. 6, Nov. 1973.

[9] "Neuro genetic optimizer version 32202," BioComp Syst., Inc. 2871, Redmond, WA.

[10] B. Ben-Dor, U. P. Oppenheim, and L. S. Balfour, "Polarization properties of targets in backgrounds in the infrared," *SPIE*, vol. 1971, 8th Meet. Opt. Eng. Israel, 1992, pp. 68–77.

[11] M. B. Abdulghafour, "Data fusion through fuzzy reasoning applied to feature extraction from multisensory images," Ph.D. dissertation, Univ. Tennessee, Knoxville, Dec. 1992, pp. 41–96.

[12] A. Filippidis and L. C. Jain, "Identity attribute information in a multiband aerial photo image using neural networks to automatically detect targets" *Int. J. Knowledge-Based Intell. Eng. Syst.*, vol. 1, no. 1, Jan. 1997.

Hierarchical Overlapped SOM's for Pattern Classification

P. N. Suganthan

***Abstract*— We develop a multilayer overlapped self-organizing maps (SOM's) with limited structure adaptation capabilities, and associated learning scheme for labeled pattern classification applications. The learning algorithm consists of the standard unsupervised SOM learning of synaptic weights as well as the supervised learning vector quantization (LVQ) 2 learning. As higher layer SOM's overlap, the final classification is made by fusing the classifications of top-level overlapped SOM's. We obtained the best results ever reported for any SOM-based numerals classification system.**

***Index Terms*— Handwritten character recognition, hierarchical self-organizig maps, numerical recognition, self-organizing maps.**

I. Introduction

The SOM has been applied in the study of complex problems such as vector quantization, speech recognition, combinatorial optimization, control, pattern recognition, and modeling the structure of the visual cortex. Most of the work carried out so far on SOM's has concentrated on systems with a single self-organizing layer. This self-organizing layer has generally a fixed number of neurons. However, in recent years, some structure adaptive/multilayer SOM models have also been proposed. Lee *et al.* [8] proposed a self-development neural network for adaptive vector quantization. The network has one self-organizing layer and two levels of adaptations—namely structure and parameter (synaptic vectors) levels. Kohonen's topology preserving mapping algorithm was used for parameter adaptation. Structure adaptation includes neuron generation, neuron annihilation, and neuron merging. If the distortion error of a neuron exceeds a predefined threshold, the neuron is split to generate another neuron in the vicinity. If a neuron is referenced infrequently, then the neuron is removed. If two neurons are similar, they are merged into one neuron. Obviously, maintaining a regular connectivity has proved to be a difficult task during the structure adaptation [8]. Wu *et al.* [9] investigated a supervised two self-organizing layer SOM. However, their method did not employ any structure adaptation scheme.

Cho [2] proposed a structure-adaptive SOM with a single self-organizing layer. The network is first initialized with a 4×4 map and trained using the SOM algorithm [6]. The map is calibrated using training data and the nodes which do not produce unique class label is replaced with a 2×2 submap. If a node does not activate for a long time, the node is deleted. The resulting map has an irregular connectivity. Bauer *et al.* [1] presented a growing self-organizing map (GSOM) algorithm. The GSOM has a general hypercubical shape which is adapted during the learning. Bauer *et al.* [1] also showed that the GSOM produces maps which preserve neighborhoods in a nearly optimal fashion. Fritzke [3] proposed a structure adaptation algorithm for SOM which has one self-organizing layer with sophisticated multidimensional lattice topologies. The algorithm includes cell insertion and removal based on reference frequency. Although several hierarchical SOM models have been proposed [5], [7], [9], the novelty of our approach is to allow a degree of overlap between adjacent higher-level SOM's. This overlapping structure possesses several beneficial properties.

In this paper, we develop a hierarchical overlapped self-organizing map (HOSOM) with limited structure adaptation capabilities. Our approach retains the essence of Kohonen's SOM. In general every class is represented by several neurons. Further, as every training sample is used develop a number of different upper layer maps, we achieve a degree of overlap in the upper-level SOM's. This multiplicity allows us to make the final decision by fusing the classification of several maps for every training and testing sample. In addition, every higher-level map is trained using a different subset of the training data. As the top-layer maps are pruned by merging and removing neurons, the problem of over-training is curtailed. The resulting HOSOM network offers the best performance for any SOM-based classifier and somewhat comparable performance to the best multilayer backpropagation network-based classifier. Further, the HOSOM may be regarded as an efficient alternative to the k-nearest neighbor type algorithms. This paper is organized as follows. We present the HOSOM neural network learning algorithm in the next section. In Section III, experimental results are presented and compared with other systems. The letter is concluded in Section IV.

II. HOSOM

The HOSOM has a hierarchical and overlapped structure. However, the network is initialized with just one layer first. The number of neurons in the layer has to be chosen. If there are too few neurons in the first layer, the network may have to be grown to have several

Manuscript received November 10, 1997; revised October 25, 1998.

The author is with the Department of Computer Science and Electrical Engineering, University of Queensland, St. Lucia QLD 4072, Australia.

Publisher Item Identifier S 1045-9227(99)00892-9.

TABLE I
THE UNSUPERVISED SOM ALGORITHM

USOM1: Initialise the weights for the given size map. First layer weights are randomly initialised. Subsequent layers are initialised around the root node. Initialise the learning rate parameter, neighbourhood size and set the number of unsupervised learning iterations.
USOM2: Present the input feature vector $x = [x_1, x_2, ..., x_n, ..., x_N]$ in the training data set of the root neuron.
USOM3: Determine the winner node c such that $\|x - w_c\| = \min_i\{\|x - w_i\|\}$
USOM4: Update the weights within the neighbourhood of node c, $N_c(t)$ using the standard update rule: $w_i(t+1) = w_i(t) + \alpha(t)[x_n - w_i(t)]$ where $i \in N_c(t)$. The neighbourhood wraps around at edges, i.e. column and row indices are in modulo representation.
USOM5: Update learning rate and neighbourhood size. $\alpha(t+1) = \alpha(0)\{1 - \frac{t}{K}\}$; $|N_c(t+1)| = |N_c(0)|\{1 - \frac{t}{K}\}$, where K is a constant and is usually set to be equal to the total number of iterations in the self-organising phase.
USOM6: Repeat USOM2-5 for the specified number of unsupervised learning iterations.

TABLE II
THE SUPERVISED LVQ 2 LEARNING ALGORITHM FOR SOM

SSOM1: Present the input feature vector $x = [x_1, x_2, \cdots, x_n, \cdots, x_N]$ in the training data set.
SSOM2: Locate the winner node c such that $\|x - w_c\| = \min_i\{\|x - w_i\|\}$.
SSOM3: If the winning neuron has the same lavel as the training example, update weights of the winning neuron only using the standard update rule: $w_c(t+1) = w_c(t) + \beta[x_n - w_c(t)]$.
If the winning neuron has a different label, then 1) update the weights of the winning neuron only using a small negative learning rate β_1 as follows: $w_c(t+1) = w_c(t) + \beta_1[x_n - w_c(t)]$ and 2) Locate the closest neuron with the same label as the training sample and update its weights uring the update equation with a positive learning rate of β_2.
SSOM4: Repeat **SSOM1-3** for the specified number of supervised learning iterations.

layers. If there are too many neurons in the first layer, computational advantage of hierarchical architecture may be compromised [9]. In pattern recognition applications, the number of training samples may be considered in the selection of initial lattice size.

For the initial adaptation of the synaptic weight vectors, we employ Kohonen's SOM algorithm. The algorithm is applied to the topmost layers which were grown during the last structure adaptation epoch.[1] The SOM algorithm is summarized in Table I.

Having completed the unsupervised SOM learning, the neurons in the topmost layers are labeled using a simple voting mechanism. Then the supervised LVQ 2 [6] algorithm given in Table II is applied to fine-tune the prototype vectors. We apply the following structure adaptation techniques just after applying the supervised LVQ 2 algorithm.

1) *Growing a Layer:* The network may be grown, until either a satisfactory recognition rate is achieved OR a predefined level of structural complexity is reached by the network. The complexity may be defined in terms of number neurons or layers.
2) *Merging/Removing Neurons:* The merger operation is essential in particular in the final layer which is not to be grown further. We employ a simple scheme. If an end-node neuron represents a few (in our experiments less than 3) training samples, that neuron is merged with another neuron which is the closest with the same label. If there is no other neuron with the same label, the neuron is removed. It should be noted that merging operation improves the performance on test data set, in case the network has been over-trained or over-specialized [6], [9].

[1] Here we assume that the environment does not change. If the environment is nonstationary, synaptic vectors of all neurons should be adapted, not just the newly grown layers.

It was indicated earlier that overlapped SOM's are obtained at the completion of training. The overlapping is achieved by duplicating every training sample to train several upper-level SOM's. That is, the winning neuron as well as a number of runners-up neurons make use of the same training sample to train the higher-level maps grown from those neurons. By duplicating the training samples in the upper-level SOM's, we obtain overlapped SOM's. The testing samples are also duplicated, but to a lesser degree. Hence, the testing samples fit well inside the feature maps developed using the best matching and several runners-up in the training data. In addition, this duplication of samples allows us to employ a voting scheme to obtain the final classification. More sophisticated decision fusion techniques such as fuzzy decision fusion [2] may be employed to improve the final decision based on these multiple overlapped SOM's. The proposed HOSOM algorithm for pattern recognition is presented in Table III. Fig. 1 shows the first-layer SOM and two instances of second-layer SOM's grown out of nodes A and B. The figure also shows the overlap in the feature space of the two second-level maps conceptually.

III. EXPERIMENTAL RESULTS

We performed handwritten digit recognition experiments to test the proposed classification method. The handwritten digits samples were extracted from the NIST database 19. A feature extraction algorithm similar to the one proposed by Wu *et al.* [9] was used. The algorithm is summarized as follows: 1) Each character image is noise-cleaned. Noise-cleaning process involves two steps. First, vertical and horizontal projections are obtained to detect isolated blobs. All isolated blobs which are 15% or smaller than the size of the biggest blob is removed. Second, isolated "1" and "0" pixels were removed. 2) Image was centered. 3) Each character image is then rescaled to 88 $\times$ 72 pixels. 4) Each scaled image is divided into 8 $\times$ 8 blocks and pixel values in each block are summed to result in an 11 $\times$ 9 image

TABLE III
THE HOSOM ALGORITHM

HOSOM1: Apply USOM (Table 1)

HOSOM2: Label all output nodes using a simple voting scheme.

HOSOM3: Apply SSOM (Table 2)

HOSOM4: Merge/remove neurons

HOSOM5: Apply SSOM (Table 2)

HOSOM6: Obtain recognition rates on training data.

HOSOM7: Grow an additional layer and repeat HOSOM1-6 until a satisfactory recognition rate is achieved OR maximum complexity level is reached.

TABLE IV
NUMERALS RECOGNITION RESULTS

First layer map size	Overlap in training samples	Overlap in testing samples	Training accuracy (%)	Testing accuracy (%)
10x10	3	1	99.82	98.10
10x10	5	1	99.92	98.17
15x15	7	3	99.93	98.58
15x15	9	5	99.97	98.87
15x15	11	7	99.95	98.90
15x15	13	9	99.96	98.91

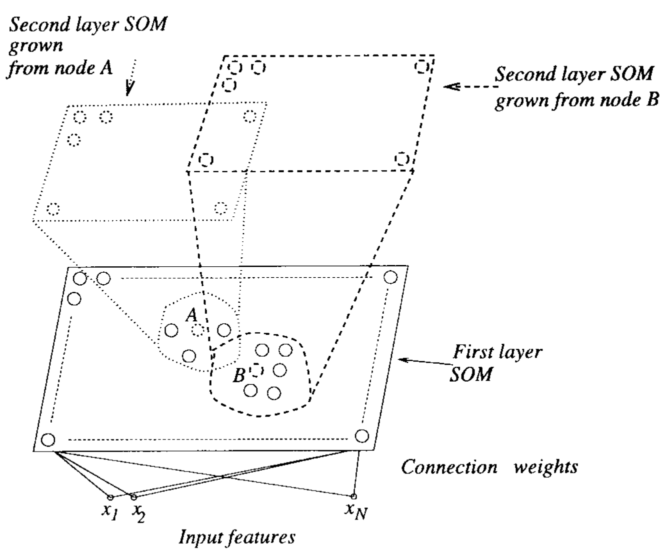

Fig. 1. The HOSOM structure.

with pixel values in the range of [0, 64]. This preprocessing yields a feature vector with 99 elements. Further, the extracted samples are divided into two sets—namely training and test sets.

In our experiments, we restricted the number of layers to two. The first layer has 10×10 or 15×15 lattice. The number of rows (columns) in the second layer is determined using the number of training samples available for training by the following expression: $\max[5, \text{sqrt}(\text{Number\_of\_training\_samples}/8)]$. As indicated earlier, we assume that the environment is stationary and apply the Kohonen's algorithm to the most recently created top layers. The initial neighborhood size is set to two-thirds of the map size for all layers. The number of iterations needed to form the topological map using the unsupervised SOM learning is set to around six times the number of training samples. Initial learning rate was set to 0.6.

Training a HOSOM with 15×15 first-layer neurons involves iterating Table III once to obtain the first-layer SOM and iterating Table III 15×15 times to obtain 225 second-layer overlapped SOM's and hence, the two self-organizing layer HOSOM network. As there are 226 SOM's altogether, training the HOSOM can be computationally demanding. However, testing speed may not be reduced, as it involves searching at most nine closest neurons in the first-level map and nine winner neurons in nine different second-level SOM's (this corresponds to the last row in Table IV). If we have an overlap of 13 for training data as in the last row of Table IV, every training sample is used to train 13 second-level SOM's grown from the best matching 13 neurons. During the testing phase, we obtain nine labels from the second-level maps grown out of the best matching nine units in the first level to make a classification decision.

The data sets have 23 000 training samples and 23 000 testing samples. The best results obtained using the SOM and its variants (to our knowledge) are as follows: Wu *et al.* [9]: 97.3% and Cho [2]: 96.05%. Apparently, the recognition rates of our method (given in Table IV) is far superior to other SOM-based classifiers due to primarily the overlapped multilayer structure. To our knowledge, only the rates of Ha *et al.* [4] are better than ours. However, Ha *et al.* designed two recognition systems using two distinct feature vectors—namely, 48-elements-long projection features and 99-elements-long contour features. The two individual classifiers are fused to obtain a rate of 99.34%. We used a primitive feature extraction method—namely, the normalized character matrix. A more meaningful comparison would be possible between the classifiers when we also use the identical data sets and feature vectors. The standard nonstructure-adaptive implementation without overlap achieved a recognition rate of 99.5 and 97.6% on training and testing data, respectively.

By examining Table IV, we observe that the recognition performance improves substantially as the degree of overlap is increased to nine and five for training and testing data, respectively. Further increase in the overlap did not offer any substantial improvements in the performance. We also experimented with different combinations of overlap in the training and testing data such as (11, 5), (13, 7), and (13, 5). These combinations yielded performances lower than those presented in the table for the same degree of training sample overlaps (refer to the last two rows in the table). Hence, the results clearly show the advantages and a consistent trend in performance variations with the degree of training and testing sample overlap.

IV. Conclusion

In this paper, we proposed a structure-adaptive multilayer overlapped SOM networks for labeled pattern classification. The proposed approach possesses several attractive features such as the presence of multiple hierarchical overlapped SOM's, and an ability to avoid overfitting by merging and pruning operations. The overlap allows us to employ a decision fusion scheme by a majority voting mechanism to improve the final classification performance. Further, different higher-level maps are trained using a different subset of the training data. We tested the developed network, HOSOM, on handwritten numerals and obtained the best results ever for any SOM-based classifier to our knowledge.

References

[1] H. U. Bauer and T. Villmann, "Growing a hypercubical output space in a self-organizing feature map," *IEEE Trans. Neural Networks*, vol. 8, pp. 218–226, 1997.

[2] S. B. Cho, "Neural-network classifiers for recognising totally unconstrained handwritten numerals," *IEEE Trans. Neural Networks*, vol. 8, pp. 43–53, 1997.

[3] B. Fritzke, "Growing cell structures—A self-organizing network for unsupervised and supervised learning," *Neural Networks*, vol. 7, no. 9, pp. 1441–1460, 1994.

[4] T. M. Ha and H. Bunke, "Off-line, handwritten numeral recognition by perturbation method," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 535–539, 1997.

[5] M. Herrmann, R. Der, and G. Balzuweit, "Hierarchical feature maps and nonlinear component analysis," in *Proc. ICNN'96*, Texas, vol. 2, pp. 1390–1394.

[6] T. Kohonen, *Self-Organizing Maps*, 2nd ed. Berlin, Germany: Springer-Verlag, 1997; also "The self-organizing maps," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480.

[7] J. Lampinen and E. Oja, "Clustering properties of hierarchical self-organizing maps," *J. Math. Imaging Vision*, vol. 2, no. 3, pp. 261–272, 1992.

[8] T. C. Lee and A. M. Peterson, "Adaptive vector quantization using a self-development neural network," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1458–1471, 1990.

[9] J. Wu, H. Yan, and A. Chalmers, "Handwritten digit recognition using two-layer self-organizing maps," *Int. J. Neural Syst.*, vol. 5, no. 4, pp. 357–362, 1994.

A "Mutual Update" Training Algorithm for Fuzzy Adaptive Logic Control/Decision Network (FALCON)

Sinan Altug, H. Joel Trussell, and Mo-Yuen Chow

***Abstract*— The conventional two-stage training algorithm of the fuzzy/neural architecture called FALCON may not provide accurate results for certain type of problems, due to the implicit assumption of independence that this training makes about parameters of the underlying fuzzy inference system. In this correspondence, a training scheme is proposed for this fuzzy/neural architecture, which is based on line search methods that have long been used in iterative optimization problems. This scheme involves synchronous update of the parameters of the architecture corresponding to input and output space partitions and rules defining the underlying mapping; the magnitude and direction of the update at each iteration is determined using the Armijo rule. In our motor fault detection study case, the mutual update algorithm arrived at the steady-state error of the conventional FALCON training algorithm as twice as fast and produced a lower steady-state error by an order of magnitude.**

***Index Terms*— Adaptation, artificial neural networks, fault detection and diagnosis, mutual update, training.**

I. Introduction

The fuzzy/neural (FZ/NN) architecture proposed by Lin and Lee [1] has been popular with numerous applications in the literature [1], [2]. In this architecture, two sets of parameters are of interest: *membership function parameters and rule parameters* (or *weights*). These correspond to the fuzzy partitions of the input and output spaces, and the relative importance of the rules underlying the architecture, respectively.

The training of the FZ/NN architecture involves modification of these parameters. In one stage, the rule parameters are updated using a modified competitive learning algorithm while the membership function parameters are fixed. In the other stage, the membership function parameters are updated by gradient descent while the rule parameters are fixed to the values found in the first stage.

This two-stage training method may not provide satisfactory results for certain problems, as the training implicitly assumes that the two sets of parameters are almost independent. Usually, the two sets are highly interdependent [1], [3], [4], and this training may not provide accurate results for the cases where this implicit assumption does not hold.

In this correspondence, a method for training this FZ/NN architecture is proposed, which involves synchronous update of the two sets of parameters, and eliminates the requirement for a two-stage training scheme. The proposed training scheme is the *mutual update algorithm*, where at each step, an update is performed in a way to alter either the membership function parameters, or the rule parameters, or both. The magnitude and the direction at each iteration are computed by examining the *sufficient decrease* condition in the objective function.

Manuscript received March 10, 1998; revised September 29, 1998. This work was supported by National Science Foundation under Grant ECS-9 521 609.

The authors are with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695-7911 USA.

Publisher Item Identifier S 1045-9227(99)00638-4.