Scientia Iranica, Vol. 8, No. 4, pp 277-286

SCIENTIA
IRANICA

Automatic Adjustment of Learning Rates of the Self-Organizing Feature Map

H. Shah-Hosseini* and R. Safabakhsh[1]

Time-decreasing learning rate and neighborhood functions of the conventional SOFM (Self-Organizing Feature Map) algorithm are two factors that reduce the capability of this map to adapt weights for different environments. Consequently, parameters for each environment have to be selected empirically, which is a very time-consuming process. In this paper, for dealing with non-stationary input distributions and varied environments, a SOFM algorithm is proposed that automatically adjusts the learning rate of each neuron independently. The learning rate is adjusted by the function of distance between an input vector and the weight vectors. The learning rate modification rule for each output neuron maximizes the correlation between a normalized error of the output neuron and its learning rate parameter. It is also demonstrated that learning rates are able to adjust themselves to vary between zero and one according to their surrounding conditions. With these modifications, the proposed SOFM algorithm virtually has no initial condition requirements crucial to its success when it is used as a Vector Quantizer (VQ) network. Moreover, the proposed network has some degree of incremental learning capability and converges to a topographic feature map representing the distribution of input vectors. Experimental results illustrate the superiority of the proposed SOFM algorithm in learning samples of non-stationary distributions. They also indicate that the proposed algorithm speeds up the SOFM convergence and stabilizes with lower distortion values. Moreover, it is shown that with a time-decreasing exponential neighborhood function, the proposed SOFM converges to the topographic map of the input samples.

INTRODUCTION

Self Organizing Feature Map (SOFM), originally developed by Kohonen [1], transforms incoming samples (signals) of arbitrary dimensions to a one- or two-dimensional discrete map in an adaptive fashion. This network has been used in applications such as vector quantization [2], texture segmentation [3], brain modeling [4], phonetic typewriter [5] and image compression [6].

SOFM uses a Hebb-like learning rule with time-decreasing learning parameters. The learning rate should begin with a value close to unity that decreases gradually, but stays above 0.1. It is during this initial phase of the algorithm that the topological ordering of the weight vectors $\mathbf{w}_j(n)$ takes place. This phase is called the *ordering phase*. The remaining iterations of the algorithm are needed for the fine-tuning of the computational map, called the *convergence phase*.

In order to observe the topological ordering of the weight vectors, the neighborhood function usually begins with all the neurons in the network and, then, gradually shrinks with time. A popular choice for the dependence of the learning rate (and the width $\sigma(n)$ of the neighborhood function) on time n is the exponential decay [4] described as:

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{\tau}\right), \qquad (1)$$

where τ is the time-constant and η_0 is the initial value of the learning rate parameter [4,7].

In fact, these parameters are at their highest values at the beginning of learning. Then, they decrease with time so that the feature map stabilizes and learns the topographic map of the incoming samples. At the final step, the learning rate parameter usually has a very small value and so does the neighborhood function. Therefore, the SOFM algorithm cannot learn the

*. *Corresponding Author, Department of Computer Engineering, Amir Kabir University of Technology, Tehran, I.R. Iran.*

1. *Department of Computer Engineering, Amir Kabir University of Technology, Tehran, I.R. Iran.*

new incoming samples that might be different in statistical characteristics from the previously learnt samples with adequate speed. In other words, the learning process is incapable of responding appropriately to a different environment that embodies incoming samples.

On the other hand, specific methods for determining the exact form of the neighborhood function and adjusting the learning rate parameter for the SOFM are not known; therefore, this task is usually performed through experiments. However, some efforts have been made to tackle this problem. In [8], the model of Kalman filters is used to automatically adjust the learning parameters, which of course is valid only within the system model. Moreover, it is computationally expensive and cannot adapt itself to varied environments. In [9], an individual neighborhood size, which is a function of distance between the input vector and the relevant weight vector is assumed with no suggestion for the learning rate parameter adjustment. In a different work, a vector quantization method is introduced for image coding, which updates the weight vectors of the SOFM with a variance-based adaptive learning rate to preserve edges [10].

A suggestion for resolving the aforementioned problem is to choose time-independent learning parameters that change their values according to the conditions of incoming samples, not to the elapse of time. These parameters should increase the capability of the SOFM in dealing with varied environments, but they should not decrease the speed of convergence of the SOFM algorithm. In this paper, new rules for adjusting the learning rate parameters in the SOFM algorithm are proposed that include the above features. The performance of the algorithm is compared with that of the conventional SOFM and the experimental results are presented.

The proposed SOFM algorithm is described in the next section. Then, the proposed learning rate modification rule is analyzed and it is shown that the learning rate modification rule attempts to minimize an error function. Consequently, experimental results are presented and concluding remarks are given in the final section of the paper.

THE PROPOSED SELF-ORGANIZING FEATURE MAP ALGORITHM

The learning rate parameter of the standard SOFM is only a function of time, n, and gradually decreases. The parameter is chosen this way to assure the stabilization of synaptic weights, with the assumption that the input sample vectors are from a specific stationary distribution. There is no mechanism for understanding whether the input distribution is changing or not. Any fundamental change in the input distribution causes severe problems for the SOFM and the learning rule cannot change the synaptic weights of the network with adequate speed. On the other hand, since the neighborhood function does not follow changes of the environment, the neurons of the network are unable to modify the topographic map based on the changed input vectors and thus the feature map cannot take the appropriate form.

Here, a modified SOFM algorithm is proposed that automatically adjusts the learning rate parameters and incorporates possible changes of the input distribution in updating the synaptic weights [11]. For this purpose, the learning rate of each neuron is considered to follow the values of a function of distance between the input vector and its synaptic weight vector. In this way, the parameter will be changed independently for each neuron and the number of parameters will be equal to the number of output neurons. The learning rate parameter, $\eta_{i(x)}(n)$, of each neuron $i(\mathbf{x})$ for the input vector $\mathbf{x}$ is updated by the following formula:

$$\eta_{i(x)}(n+1) = \eta_{i(x)}(n) + \alpha \left(f(||\mathbf{x}(n) - \mathbf{w}_{i(x)}(n)||) - \eta_{i(x)}(n)\right), \tag{2}$$

where α is a positive constant such that $0 < \alpha \leq 1$ and $||.||$ stands for Euclidean norm. The function $f(.)$ should have the following properties:

- $f(0) = 0.$
- $0 \leq f(z) \leq 1 \quad \forall z \geq 0.$
- $\frac{df(z)}{dz} \geq 0, \quad \forall z \geq 0.$ (3)

An example of such a function is $f_1(z) = \frac{2}{1+e^{-z}} - 1$. For faster responses of the network to the changes of the environment, the function $f_2(z) = 1 - \frac{1}{1+z}$ can be used.

The proposed SOFM may be summarized as follows:

1. Initialization. First, random values for the initial weight vectors $\mathbf{w}_j(0)$ are chosen where $j = 1, 2, ..., N$; N is the number of neurons in the lattice. These random values should be small and different. The learning rate parameters, $\eta_j(n)$, should be initialized with values close to unity. The parameter α can have any value between zero and one. If fast responses to environmental changes are required, α should be chosen close to one; otherwise, small positive values guarantee slow and accurate responses to possible environmental changes. Then, the neighborhood function parameters are initialized.
2. Sampling. A sample-input vector $\mathbf{x}$ is drawn from the input distribution with a certain probability.

3. Similarity matching. The best-matching or winning neuron $i(\mathbf{x})$ is found at time n, using the minimum distance Euclidean norm as the matching measure:

$$i(\mathbf{x}) = \arg_j \min ||\mathbf{x}(n) - \mathbf{w}_j|| \quad j = 1, 2, ..., N. \tag{4}$$

4. Updating the learning rate parameters. The learning rate parameters $\eta_j(n)$ are adjusted in the neighborhood $\Lambda_{i(\mathbf{x})}(n)$ of the winning neuron $i(\mathbf{x})$ by the following formula:

$$\eta_j(n+1) = \eta_j(n) + \alpha(f(||\mathbf{x}(n) - \mathbf{w}_j(n)||) - \eta_j(n)) \quad \text{for } j \in \Lambda_{i(x)}(n). \tag{5}$$

The learning rate parameters of the other neurons do not change.

5. Updating the synaptic weights. The synaptic weight vectors of all output neurons are adjusted in the neighborhood $\Lambda_{i(\mathbf{x})}(n)$, using the following update rule:

$$\mathbf{w}_j(n+1) = \begin{cases} \mathbf{w}_j(n) + \eta_j(n+1)[\mathbf{x}(n) - \mathbf{w}_j(n)] & j \in \Lambda_{i(\mathbf{x})}(n), \\ \mathbf{w}_j(n) & \text{otherwise} \end{cases} \tag{6}$$

where $\eta_j(n+1)$ is the learning rate parameter and $\Lambda_{i(\mathbf{x})}(n)$ is a neighborhood function centered on the winning neuron $i(\mathbf{x})$. A popular choice for the form of the neighborhood function is $\exp(-\frac{d_{j,i}^2}{2\sigma^2(n)})$ with the width $\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau}\right)$ which decreases with time.

6. Continuation. Step 2 is continued until the synaptic weights stabilize and no noticeable change in the feature map is observed.

The above algorithm resembles the conventional SOFM, except for Step 4 which is a new step added to update the learning rates.

ANALYSIS OF THE PROPOSED SOFM'S LEARNING RATE RULE

In this section, the proposed learning rate modification rule is further investigated and its behavior is analyzed. By moving $\eta_{i(\mathbf{x})}(n)$ of Equation 5 to the left hand side and dividing by $n+1-n$, it is obtained that:

$$\frac{\eta_{i(\mathbf{x})}(n+1) - \eta_{i(\mathbf{x})}(n)}{n+1-n} = -\alpha\eta_{i(\mathbf{x})}(n) + \alpha f(||\mathbf{x}(n) - \mathbf{w}_{i(\mathbf{x})}(n)||). \tag{7}$$

In continuous form, this equation converts to:

$$\frac{\eta_{i(\mathbf{x})}(t+\Delta t) - \eta_{i(\mathbf{x})}(t)}{(t+\Delta t) - t} = -\alpha\eta_{i(\mathbf{x})}(t) + \alpha f(||\mathbf{x}(t) - \mathbf{w}_{i(\mathbf{x})}(t)||), \tag{8}$$

or equivalently,

$$\frac{\Delta\eta_{i(\mathbf{x})}(t)}{\Delta t} = -\alpha\eta_{i(\mathbf{x})}(t) + \alpha f(||\mathbf{x}(t) - \mathbf{w}_{i(\mathbf{x})}(t)||). \tag{9}$$

With the assumption that Δt is a very small positive value,

$$\frac{d\eta_{i(\mathbf{x})}(t)}{dt} = -\alpha\eta_{i(\mathbf{x})}(t) + \alpha f(||\mathbf{x}(t) - \mathbf{w}_{i(\mathbf{x})}(t)||), \tag{10}$$

and assuming that the changes of $\mathbf{x}(t)$ and $\mathbf{w}_{i(\mathbf{x})}(t)$ with time are much smaller than that of the learning rate $\eta_{i(\mathbf{x})}$, the above non-homogeneous first order differential equation can be solved:

$$\eta_i(t) = \eta_i(0)e^{-\alpha t} + \alpha e^{-\alpha t}\int_0^t e^{\alpha s} f(||\mathbf{x}(s) - \mathbf{w}_i(s)||)ds. \tag{11}$$

The first term on the right hand side decreases exponentially to zero as time reaches infinity. Therefore, without the second term, the learning rate always decreases with time, as in the standard SOFM. The second term on the right hand side of the equation requires a more careful examination. In fact, function $f(||\mathbf{x}(t) - \mathbf{w}_{i(\mathbf{x})}(t)||)$ provides some normalized error of the neuron i for the input vector $\mathbf{x}$ and its value always lies between zero and one. Therefore, it is possible to write:

$$0 \leq \text{ the second\_term} = \alpha e^{-\alpha t}\int_0^t e^{\alpha s} f(||\mathbf{x}(s) - \mathbf{w}_i(s)||)ds \leq \alpha e^{-\alpha t}\int_0^t e^{\alpha s} ds, \tag{12}$$

or equivalently,

$$0 \leq \text{ the second\_ term } \leq \alpha e^{-\alpha t}\left[\frac{1}{\alpha}e^{\alpha s}\right]_0^t, \tag{13}$$

or,

$$0 \leq \text{ the second\_ term } \leq (1 - e^{-\alpha t}). \tag{14}$$

Considering Equations 11 and 14, the following conclusions can be drawn. When time reaches infinity and the normalized error $f(||\mathbf{x}(t) - \mathbf{w}_{i(x)}(t)||)$ is high (near one), the second term on the left hand side of Equation 11

approches one, its upper limit and the first term goes to zero. Therefore, when the error is high, the learning rate approaches one, its highest value.

Now consider the case where the error function of neuron i is low (near zero). The second term in this case will be near zero and as time reaches infinity, the first term also approches zero. Therefore, the conclusion can be drawn that the learning rate reaches zero when the error is low.

Based on the above discussion, it might be stated that the first term of Equation 11 plays the role that a standard SOFM algorithm demands: gradual exponential decay with time. On the other hand, the second term of the equation assumes a time varying environment and thus, it is always ready to adjust learning rates according to the present conditions of the environment. At the same time, the second term causes no problem for a stable environment. Therefore, the learning rates are automatically adjusted between zero and one and the value of parameter α is not crucial to these adjustments.

MINIMIZATION OF A COST FUNCTION

Consider the continuous form of the learning rate modification rule of Equation 5 as:

$$\frac{d\eta_{i(\mathbf{x})}(t)}{dx} = \alpha(f(||\mathbf{x}(t) - \mathbf{w}_{i(\mathbf{x})}(t)||) - \eta_{i(\mathbf{x})}(t)). \tag{15}$$

It is possible to assume that the above rule tries to minimize a cost (error) function, say $e_{i(\mathbf{x})}$. The above rule is, then, equal to the negative of the gradient with respect to the learning rate $\eta_{i(\mathbf{x})}$ of the cost function. Specifically,

$$\frac{de_{i(\mathbf{x})}}{d\eta_i} = -\alpha(f(||\mathbf{x}(t) - \mathbf{w}_{i(\mathbf{x})}(t)||) - \eta_{i(\mathbf{x})}(t)). \tag{16}$$

For simplicity, i is used instead of $i(\mathbf{x})$ and η_i instead of $\eta_{i(\mathbf{x})}(t)$. By integrating both sides and ignoring constant terms:

$$e_i = -\alpha \int f(||\mathbf{x}(t) - \mathbf{w}_i(t)||)d\eta_i + \alpha \int \eta_i d\eta_i, \tag{17}$$

calculating the second term on the right hand side of the equation gives:

$$e_i = -\alpha \int f(||\mathbf{x}(t) - \mathbf{w}_i(t)||)d\eta_i + \alpha\frac{\eta_i^2}{2}. \tag{18}$$

With the assumption that the changes of the normalized error $f(||\mathbf{x}(t) - \mathbf{w}_i(t)||)$ with respect to η_i are very small, the error is taken out of the integral to obtain:

$$e_i = -\alpha f(||\mathbf{x}(t) - \mathbf{w}_i(t)||) \int d\eta_i + \alpha\frac{\eta_i^2}{2}, \tag{19}$$

and finally:

$$e_i = -\alpha f(||\mathbf{x}(t) - \mathbf{w}_i(t)||)\eta_i + \alpha\frac{\eta_i^2}{2}. \tag{20}$$

This is the cost function for neuron i. The cost function $e(t)$ at time t for all of the neurons is the summation of individual cost functions e_i:

$$e(t) = \sum_{i=1}^{N} e_i. \tag{21}$$

Replacing e_i with their equivalents in Equation 20, it is obtained that:

$$e(t) = -\alpha \sum_{i=1}^{N} f(||\mathbf{x}(t) - \mathbf{w}_i(t)||)\eta_i + \frac{\alpha}{2}\sum_{i=1}^{N}\eta_i^2, \tag{22}$$

the total error e_{total} of the network is the expectation of $e(t)$ over time. Thus,

$$e_{\text{total}} = \lim_{T\to\infty}\frac{1}{T}\sum_{t=0}^{T} e(t) = E(e(t)), \tag{23}$$

or,

$$e_{\text{total}} = -\alpha \sum_{i} E(f(||\mathbf{x} - \mathbf{w}_i||)\eta_i) + \frac{\alpha}{2}\sum_{i} E(\eta_i^2). \tag{24}$$

According to this equation, the total error e_{total} is minimized when each $E(f(||\mathbf{x} - \mathbf{w}_i||)\eta_i)$ and each $E(\eta_i^2)$ are minimized. The former is a cross-correlation between the normalized error and the learning rate of the same neuron. Therefore, the total error is reduced when the cross-correlation increases. In other words, the learning rate modification rule tries to maximize the correlation between the normalized error and the learning rate and, thus, the learning rate should follow the normalized error. On the other hand, $E(\eta_i^2)$ in Equation 24 is the time-average of the square of learning rate that the learning rate modification rule tends to minimize.

EXPERIMENTAL RESULTS

To compare the performance of the proposed and the conventional SOFMs, a network of two layers is considered: an input layer of two nodes and a one-dimensional lattice of output neurons. In the first experiment, the neighborhood function has the smallest size; i.e., it includes only the winning neuron. Consequently, it has no effect on the learning process and, thus, the two SOFMs behave as VQ networks [12]. In addition, the value of τ in Equation 1 for the conventional SOFM is considered equal to 5000 and the value of α in the proposed SOFM equals to unity.

For uniformly distributed random two-dimensional input vectors, the synaptic weights, which are initially set equal to positive values around (0.5, 0.5), converge to their final values after 15000 iterations of the conventional and proposed SOFMs, as shown in Figures 1 and 2, respectively. The distortion values are also depicted in Figure 3. It is seen that the synaptic weights of both algorithms are scattered evenly in the weight space, but the proposed SOFM converges faster than the conventional algorithm. If the input distribution changes to two different uniform distributions, one in the $[0,1] \times [0,1]$ region and the other in the $[1,2] \times [1,2]$ region, the new converged weights of the two algorithms would be as shown in Figures 4 and 5, respectively. According to the changes of distortion values illustrated in Figure 6, the proposed SOFM converges much faster to its final state than the conventional one. Moreover, it stabilizes with lower distortion values than the conventional SOFM. Thus, it

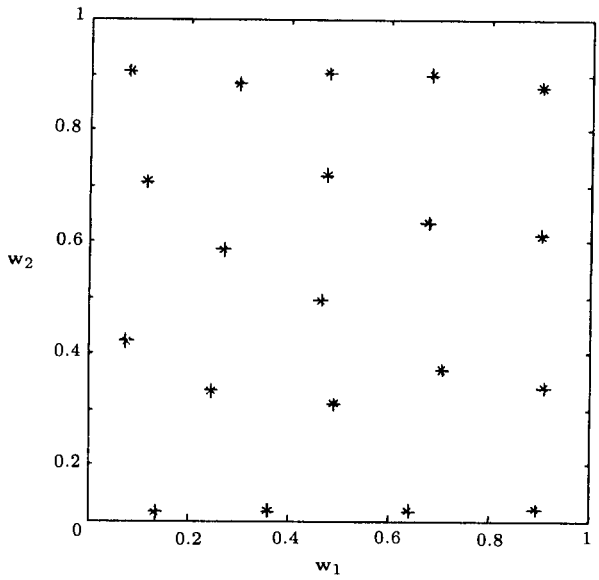

Figure 1. The feature map of the conventional SOFM algorithm after 15000 iterations for uniformly distributed input samples in the region $[0,1] \times [0,1]$ and the smallest neighborhood size.

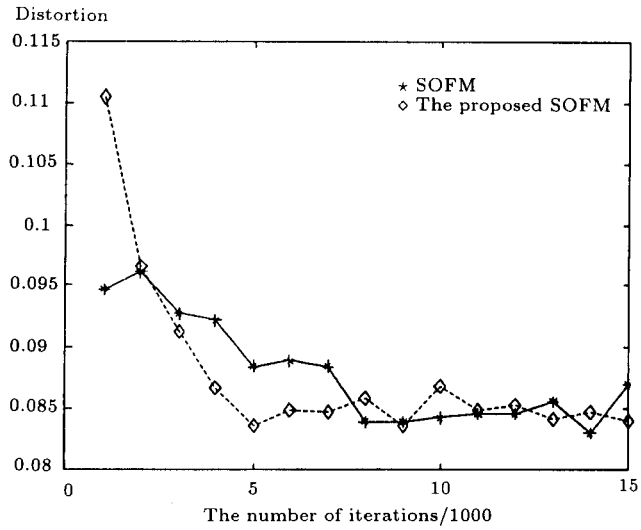

Figure 3. The distortion values of the learning processes of the conventional SOFM and the proposed SOFM for uniformly distributed random two-dimensional vectors as input samples.

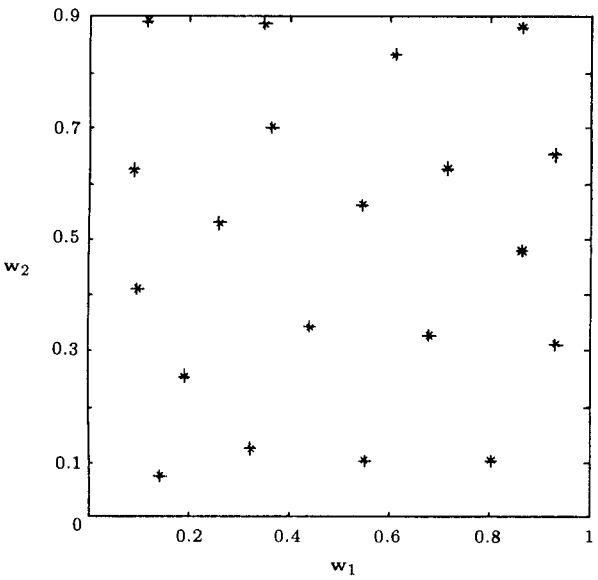

Figure 2. The feature map of the proposed SOFM algorithm after 15000 iterations for uniformly distributed input samples in the region $[0,1] \times [0,1]$ and the smallest neighborhood size.

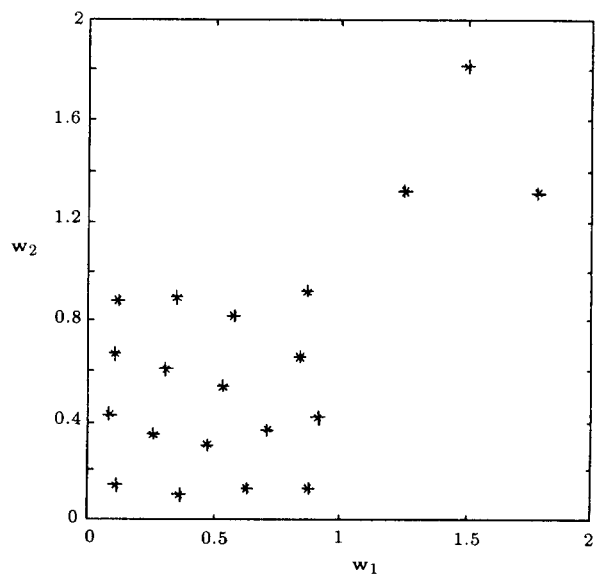

Figure 4. The feature map of the conventional SOFM algorithm after 15000 iterations with initial parameters and weights of the converged network of Figure 1 and the smallest neighborhood size.

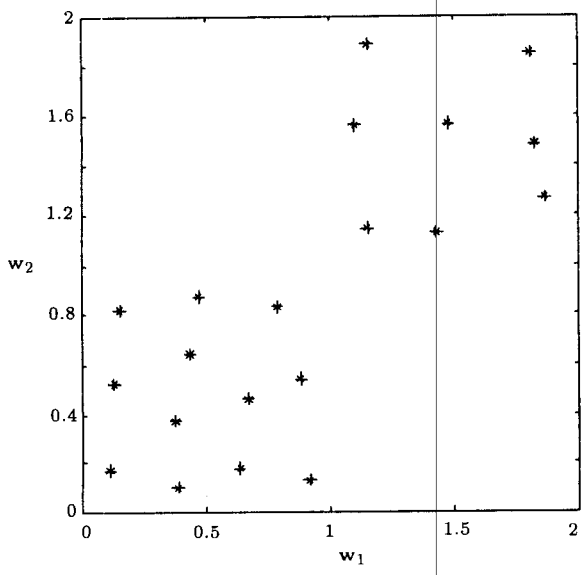

Figure 5. The feature map of the proposed SOFM algorithm after 15000 iterations with initial parameters and weights of the converged network of Figure 1 and the smallest neighborhood size.

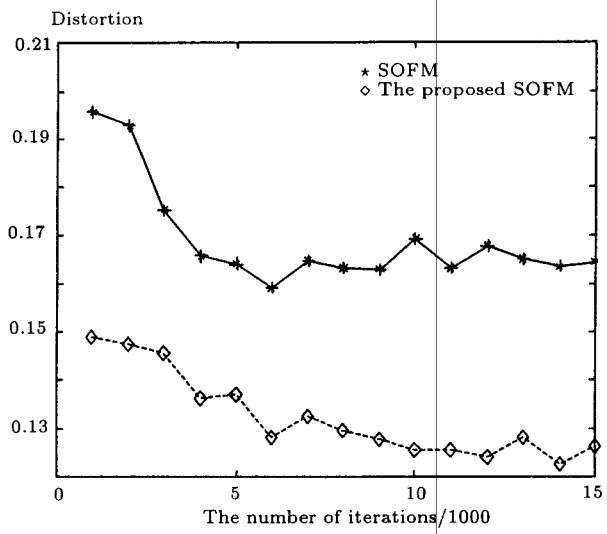

Figure 6. The distortion values of the learning processes of the conventional SOFM and the proposed SOFM for an input distribution composed of two different uniform distributions, one in the region $[0,1] \times [0,1]$ and the other in the region $[1,2] \times [1,2]$.

might be concluded that the proposed SOFM algorithm shows some degree of incremental learning in its process and is more suitable than the conventional one for non-stationary input distributions and varied environments. In these two experiments, the distortion measure $Q = \frac{1}{M}\sum_{j=1}^{M} d(\mathbf{x}_j, \mathbf{w}_{i(\mathbf{x}_j)})$ has been used to evaluate the clustering performance of the algorithms [7]. Here, M is the number of the training input vectors, $d(.,.)$ is a distance function and $\mathbf{x}_j$ is an input vector while $\mathbf{w}_{i(\mathbf{x}_j)}$ is the synaptic weight vector of the winning neuron $i(\mathbf{x})$ for the input vector $\mathbf{x}_j$.

Assume again that the two-dimensional uniform distribution is in the region $[0,1] \times [0,1]$ for the input vectors and the initial weight vectors of the proposed SOFM are around point (0.5, 0.5). The proposed SOFM, with 100 output neurons, $\alpha = 1$, function $f_2(.)$ and exponential time-decreasing neighborhood function $\exp(-\frac{d_{j,i}^2}{2\sigma^2(n)})$ with the width $\sigma(n) = \sigma_0 \exp\left(-\frac{n}{\tau}\right)$, where $\sigma_0 = 50$ and $\tau = 1000$, converges to the one-dimensional feature map shown in Figure 7 after 6000 iterations. It has been assumed that there is no neuron before neuron 1 or after neuron N. The converged synaptic weights occupy the input vector space almost evenly, and the feature map preserves topological ordering with no unfolding in the map. If 100 neurons with the same conditions are organized into a two-dimensional lattice, the algorithm converges to the two-dimensional feature map shown in Figure 8 after 5000 iterations. It is seen that it preserves topological ordering and neurons are distributed uniformly in the map.

The two-dimensional map coordinates for the Kodiak Island shoreline [13], shown in Figure 9, are used to train the two SOFM algorithms. In this experiment, each SOFM network has a one-dimensional lattice of 25 neurons and an input layer of two neurons. For both networks, the exponential time-decreasing neighborhood function with parameters $\sigma_0 = 25$ and $\tau = 1000$ is used. The learning rate for the SOFM is decreased according to Equation 1 with parameters

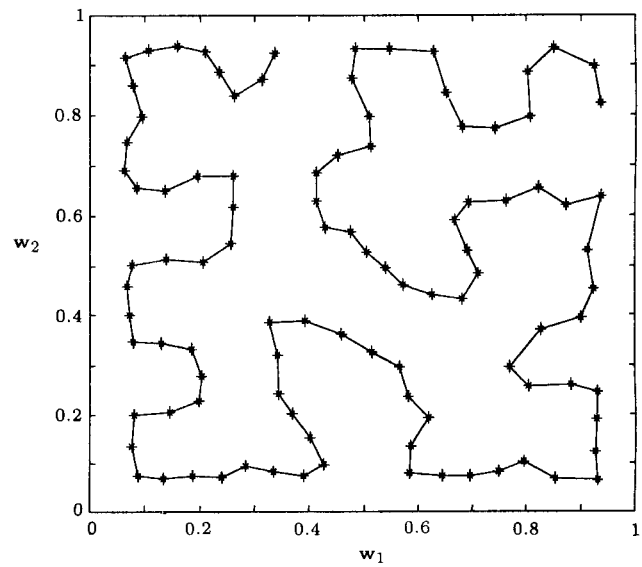

Figure 7. The topographic map of the proposed SOFM with a time-decreasing exponential neighborhood function and the proposed learning rate adjustment after 6000 iterations for the uniform distribution of region $[0,1] \times [0,1]$.

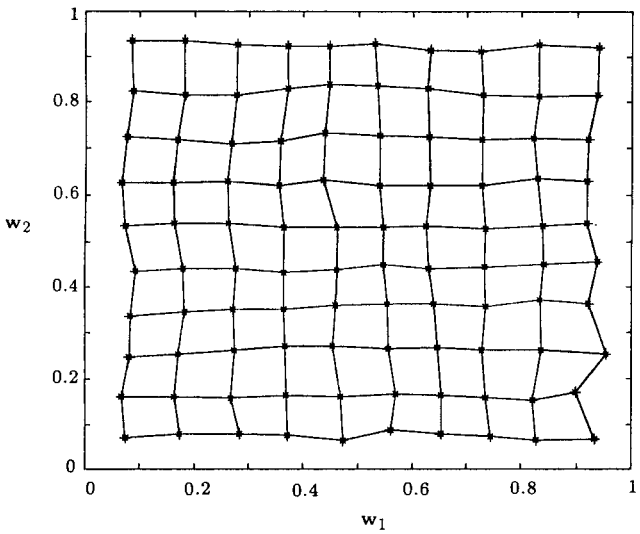

Figure 8. The two-dimensional lattice of neurons after 5000 iterations for the uniform distribution of region $[0, 1] \times [0, 1]$.

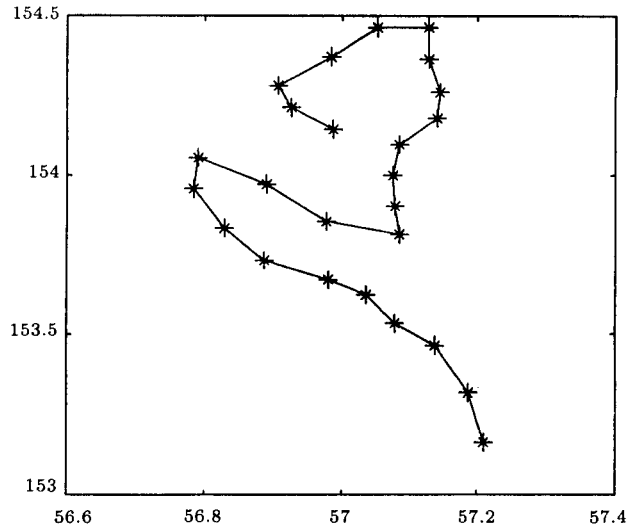

Figure 10. The weight vectors of the standard SOFM with 25 neurons after clustering the Kodiak data with quantization error 0.0464.

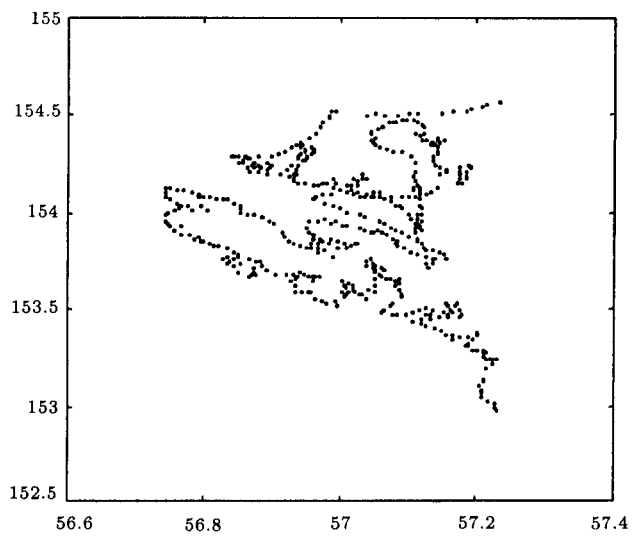

Figure 9. The 500 map coordinates of Kodiak Island shoreline.

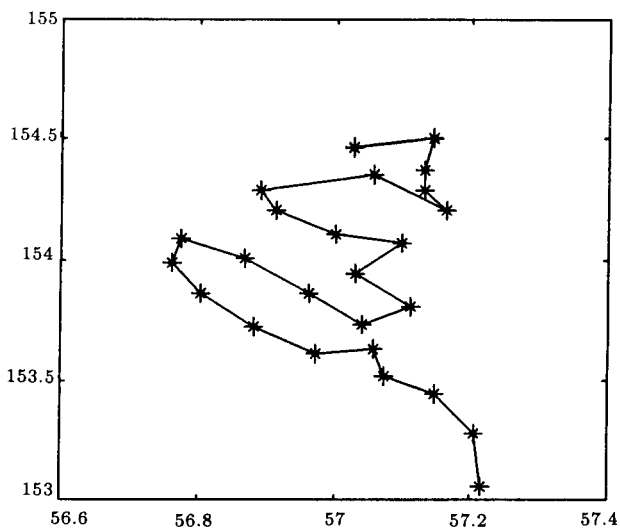

Figure 11. The weight vectors of the proposed SOFM with 25 neurons after clustering the Kodiak data with quantization error 0.0439.

$\eta_0 = 0.9$ and $\tau = 1000$. These parameters are selected for best performance in terms of lower quantization (distortion) errors. The two networks are trained with 500 samples of the Kodiak data set. The weight vectors of the standard and proposed SOFMs after weight convergence are depicted in Figures 10 and 11, respectively. The quantization error for the SOFM is 0.0464 while this error is 0.0439 for the proposed SOFM. Consequently, the proposed SOFM shows better clustering performance than the standard SOFM. Both networks preserve topological ordering in this experiment.

Quadrature Amplitude Modulation (QAM) is used for transmission of quantized signals. In QAM, the transmission channel is divided into two subchannels I (in-phase) and Q (quadrature). In 16QAM, each sample of quantized signal is transmitted with four bits: two bits in the I part and the other two bits in the Q part. Consequently, there are 16 different symbols for level transmission of signals. A square lattice of 4×4 neurons of an SOFM network with two-dimensional input vectors has been used at the receiving end as an adaptive detector for the QAM system [14]. Each neuron of the SOFM is supposed to follow its own cluster under various deformations such as corner and lattice collapse situations. Moreover, noise can be superimposed on the transmitted symbols. Because this environment is time-varying and non-stationary, it is suggested that the SOFM should keep its learning rate at a constant value after finishing the training phase [14].

Here, such a changing environment is simulated and the two SOFMs are used for the QAM as an adaptive detector. Random number generation is employed to produce quantized signals. These quantized signals are added with noise, and then delivered to an SOFM network at the receiving end. The winning neuron for the present input signal identifies the received symbol. At the same time, this input vector is used for learning of the SOFM network in order to follow changes in the environment.

Again, the exponential time-decreasing neighborhood function is used as the neighborhood function for both networks with parameters $\sigma_0 = 4$ and $\tau = 500$. The lowest value of the neighborhood function is kept at 3.8 for both networks. The learning rate for the SOFM is decreased according to Equation 1 with parameters $\eta_0 = 0.9$, $\tau = 500$ and the lowest learning rate of 0.1. As before, the learning rate for the proposed SOFM changes according to Equation 5 with function $f_2(.)$ and $\alpha = 1$. Both networks have two-dimensional input vectors, an output layer of 4×4 neurons and are trained with the noisy symbols. The weight vectors, after 1000 iterations, along with the input vectors are shown in Figures 12 and 13. In this case, both networks perform equally well. Now, the corner deformation is simulated with a rotation of 10 degrees about the origin. The results for the standard and the proposed SOFMs are shown in Figures 14 and 15, respectively. It is seen that the conventional SOFM is not able to follow this deformation, while the proposed SOFM accurately follows the simulated deformation. This result is repeated when the collapse deformation is simulated with a shear transformation. The weights of the algorithms along with the input vectors for this case are depicted in Figures 16 and 17.

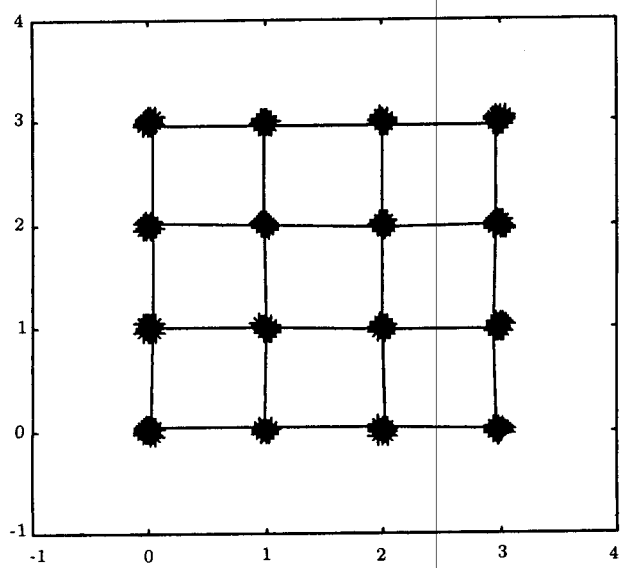

Figure 12. The weight vectors of the standard SOFM along with the noisy symbols in the QAM.

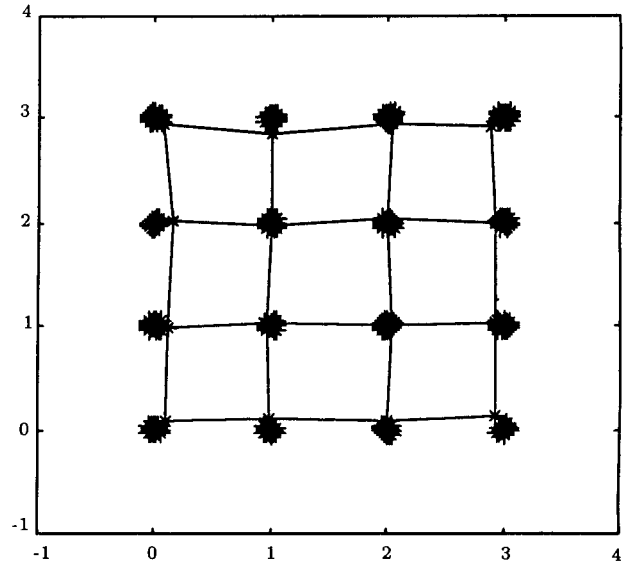

Figure 13. The weight vectors of the proposed SOFM along with the noisy symbols in the QAM.

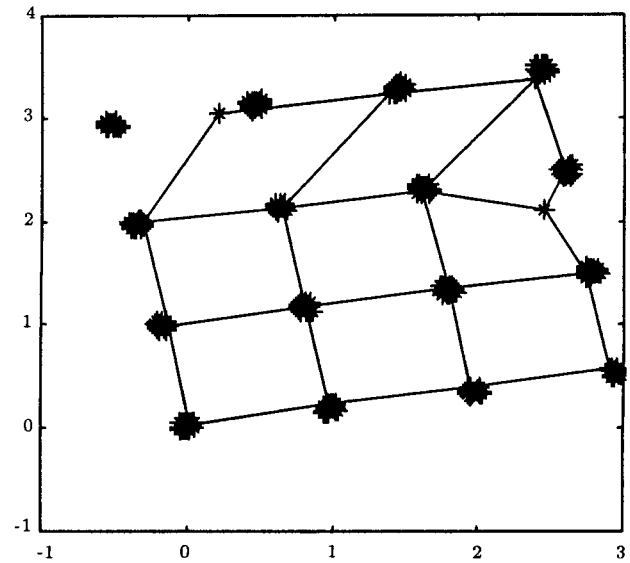

Figure 14. The weight vectors of the standard SOFM along with the noisy symbols containing the corner deformation in the QAM.

The superiority of the proposed SOFM appears here again as it thoroughly follows this deformation.

The behavior of the two algorithms is better compared when the decoding errors for every 100 receiving symbols are computed and shown in Figures 18 and 19 for the standard and the proposed SOFMs, respectively. For the noisy symbols, the two algorithms perform similar to each other. When the noisy symbols also contain the collapse deformation, the decoding error of the proposed SOFM after a few iterations becomes zero. The standard SOFM is not able to move its weights to the desired position and, thus, its decoding error remains non-zero. A similar situation is

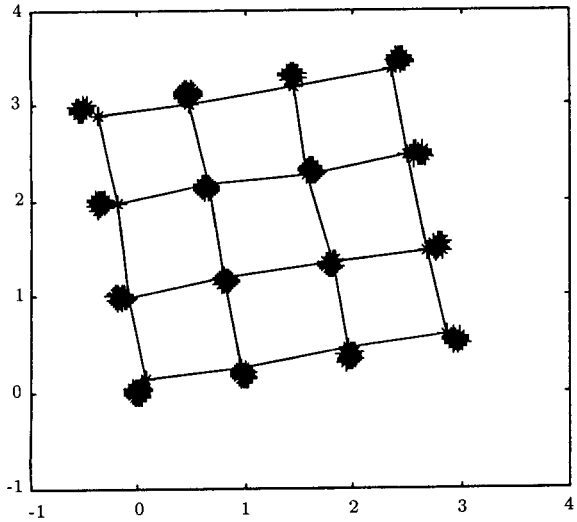

Figure 15. The weight vectors of the proposed SOFM along with the noisy symbols containing the corner deformation in the QAM.

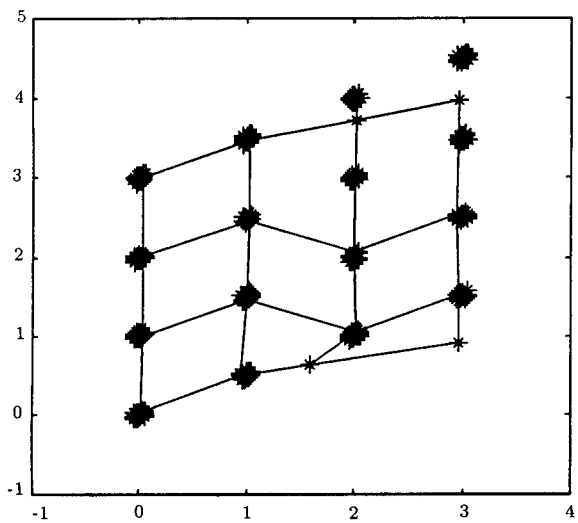

Figure 16. The weight vectors of the standard SOFM along with the noisy symbols containing the collapse deformation in the QAM.

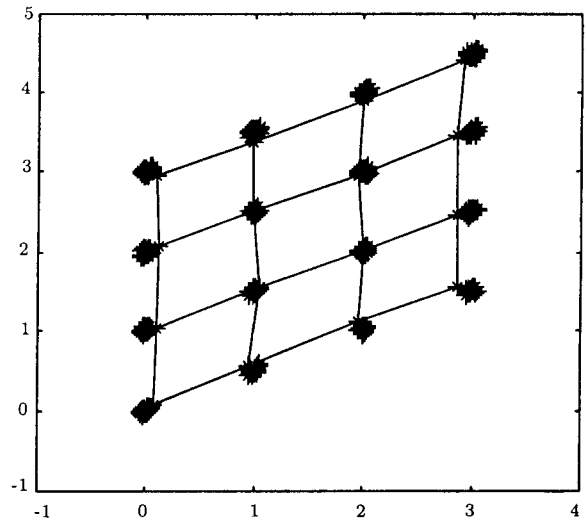

Figure 17. The weight vectors of the proposed SOFM along with the noisy symbols containing the collapse deformation in the QAM.

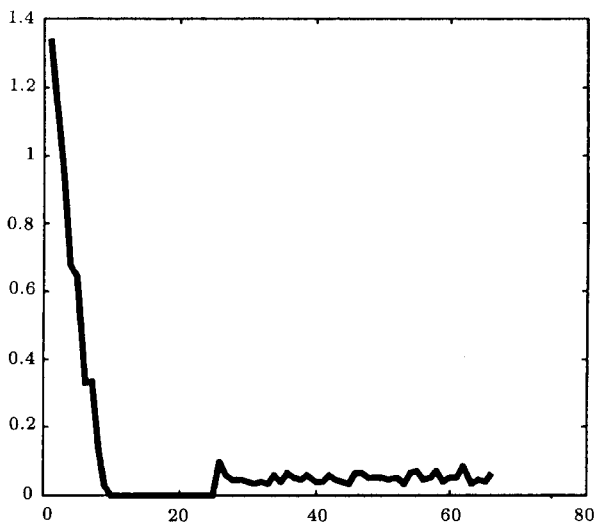

Figure 18. The decoding error of the standard SOFM with the noisy symbols. After a while, the collapse deformation is also superimposed on the noisy symbols.

also observed in cases where the noisy symbols contain corner deformation.

The proposed algorithm can effectively be used in other real-world applications. An example is presented in [15], where the algorithm is used for bilevel image thresholding and has proved itself a powerful tool for image segmentation.

It can be concluded that the proposed SOFM often converges faster than the conventional one as a VQ network, behaves more adaptively than the conventional SOFM in changing environments, needs almost no manual learning rate adjustment crucial to its learning success or failure and, finally, can preserve topological ordering using a time-decreasing exponential neighborhood function.

CONCLUDING REMARKS

The decreasing time-dependent learning parameters of the SOFM lower the adaptation and incremental learning capability of the algorithm in response to varied environments. In this paper, a new SOFM algorithm was proposed which automatically adjusts the learning rate parameter of the output neurons. Each output neuron is assumed to have its own learning rate and this rate is updated repeatedly in the proposed SOFM algorithm in response to new input samples.

Experimental results demonstrate that, as VQ networks, the proposed SOFM converges faster and with lower distortion values than the conventional SOFM. It also appears so in response to rapid changes

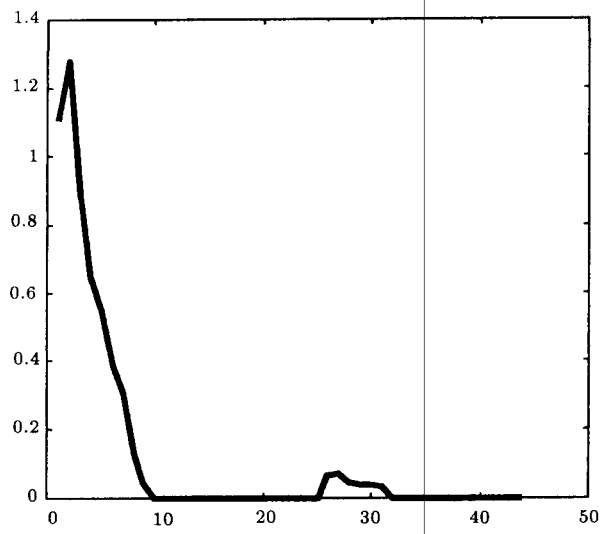

Figure 19. The decoding error of the proposed SOFM with the noisy symbols. After a while, the collapse deformation is also superimposed on the noisy symbols.

of the input distribution. Therefore, the proposed SOFM can be used in varied environments for non-stationary input distributions.

The learning rate modification of each output neuron was shown to maximize the correlation between a normalized error of the output neuron and its learning rate parameter. Also, it was demonstrated that learning rates are automatically adjusted between zero and one and the value of parameter α is not crucial to these adjustments.

With a time-decreasing exponential neighborhood function, the proposed SOFM converges to the topographic map of the input distribution and clusters the input space better than the standard SOFM. It can be used as an adaptive detector in the QAM system at the receiving end for symbol decoding in non-stationary environments. In this application, the proposed SOFM performs much better than the standard SOFM.

REFERENCES

1. Kohonen, T. "Self-organized formation of topologically correct feature maps", *Biological Cybernetics*, **43**, pp 59-69 (1982).
2. Chiuch, T.D. et al. "Vector quantization using tree-structured self-organizing feature maps", in *Applications of Neural Networks to Telecommunications*, J. Alspector, R. Goodman and T.X. Brown, Eds., pp 259-265, Hillsdale, NJ, USA (1993).
3. Oja, E. "Self-organizing maps and computer vision", in *Neural Networks for Perception*, H. Wechsler, Ed., San Diego, CA, Academic Press, **1**, pp 368-385 (1992), USA.
4. Ritter, H.J. et al., *Neural Computation and Self-Organizing Maps: An Introduction*, Reading, MA, Addison Wesley (1992).
5. Kohonen, T. "The neural phonetic typewriter", *Computer*, **21**, pp 11-22 (1988).
6. Amerijckx, C. et al. "Image compression by self-organized Kohonen map", *IEEE Trans. on Neural Networks*, **9**(3), pp 503-507 (1998).
7. Haykin, S., *Neural Networks*, Prentice Hall (1999).
8. Haese, K. "Self-organizing feature maps with self-adjusting learning parameters", *IEEE Trans. On Neural Networks*, **9**(6), pp 1270-1278 (1998).
9. Maillard, E. and Gresser, J. "Reduced risk of Kohonen's feature map non-convergence by an individual size of the neighborhood", *IEEE International Conference on Neural Networks*, **2**, pp 704-707 (1994).
10. Kim, Y.K. and Ra, J.B. "Adaptive learning method in self-organizing map for edge preserving vector quantization", *IEEE Trans. on Neural Networks*, **6**(1), pp 278-280 (1995).
11. Shah-Hosseini, H. and Safabakhsh, R. "A learning rule Modification in the self-organizing feature map algorithm", *4th Int. CSI Computer Conference*, Tehran, Iran, pp 1-9 (1999).
12. Patterson, D.W., *Artificial Neural Networks: Theory and Applications*, Prentice Hall (1996).
13. Kodiak at http://lib.stat.cmu.edu.
14. Kohonen, T., *Self-Organizing Maps*, Springer (1997).
15. Shah-Hosseini, H. and Safabakhsh, R. "The time adaptive self-organizing map with neighborhood functions for bilevel thresholding", *AIS'2000 Conference*, Arizona, USA, pp 123-128 (2000).