Biol. Cybern. 65, 55–63 (1991)

Biological Cybernetics

On the rate of convergence in topology preserving neural networks

Z.-P. Lo and B. Bavarian

Department of Electrical and Computer Engineering, University of California, Irvine, CA 92717, USA

Received September 18, 1990/Accepted in revised form February 13, 1991

Abstract. A formal analysis of the neighborhood interaction function selection in the topology preserving unsupervised neural network is presented in this paper. The definition of the neighborhood interaction function is motivated by anatomical evidence as opposed to what is currently used, which is a uniform neighborhood interaction set. By selecting a neighborhood interaction function with a neighborhood amplitude of interaction which is decreasing in spatial domain the topological order is always enforced and the rate of self-organization to final equilibrium state is improved. Several simulations are carried out to show the improvement in rate between using a neighborhood interaction function vs. using a neighborhood interaction set. An error measure functional is further defined to compare the two approaches quantitatively.

1 Introduction

Artificial Neural Networks (ANN) are providing the possibility of alternative solutions to many old problems with improved performance. As a recently revitalized area, many basic problems are yet to be solved before we can realize the implementation of ANN. The unsupervised self-organization or the property of learning and extracting important features of a given sensory input or information set is one of the attractive properties of neural networks. This concept is the main motivation behind Topology-Preserving Networks. The feature map is formed in a topologically ordered fashion, i.e., similar inputs are mapped into neighboring neurons in the network. This follows closely the observations in biological systems in which different sensory inputs are known to be mapped onto neighboring areas of the brain in an orderly fashion. Several approaches have been proposed to form such mapping (Takeuchi and Amari 1979; Willshaw and Malsburg 1976, 1979; Kohonen 1982a). In this work we consider the Kohonen approach.

Kohonen has formulated a learning rule or weight adaptation algorithm for the self-organizing feature map network (Kohonen 1982a). The network can automatically form one or many-dimensional maps of intrinsic features of the input data. The input data is presented in mixed random order to the network. Kohonen has shown that self-organizing networks with two dimensional topology are capable of learning complicated hierarchical relations of high-dimensional space by many simulations (Kohonen 1982b). The topology preserving network has been applied to many problems. Speech recognition is one of the first problems that has been tested on a self-organizing neural network (Kohonen 1984). In that work the neural network develops the order maps of the phonemes and recognizes the words in speech. Ritter and Schulten (1986a) used the network for learning the motor tasks, and extended the work to learn more complex motor actions such as ballistic movements (Ritter and Schulten 1986b).

In general, the neural network learning rule is very slow and extensive number of training input vectors are required for convergence to equilibrium state. Recently several researchers (Ritter and Schulten 1986c, 1988; Cottrell and Fort 1986) have investigated convergence properties of the Kohonen network. Ritter and Schulten (1986c) derived an equation for the equilibrium state of Kohonen network and gave an analytical expression of the local magnification factor in terms of the probability density of the input for one dimensional case. However analysis of the neighborhood selection in the Kohonen weight adaptation algorithm and its effect on the convergence or rate of convergence has not been addressed in the literature. In this paper we define the parameters of the neuron's neighborhood interactions and present a formal analysis of the effects of those parameters on rate of convergence of the network in the training phase. The analysis also provides systematic approach for selecting neighborhood interaction parameters to enforce ordered structure. We will define two classes of neighborhood interaction, one using neighborhood interaction set and the other using neighborhood interaction function. The topology

preserving properties of the neuron neighborhood interaction function are formulated in three theorems with a corollary. Furthermore the convergence properties of the network using neighborhood interaction function and neighborhood interaction set which is mostly used in the previous works are compared through several simulations and formal definition of convergence error.

The remainder of the paper is organized as follows. In Sect. 2 the Kohonen topology preserving network is briefly reviewed, the algorithm for the weight adaptation is presented. The formal analysis of neighborhood selection for the neuron in the learning phase is provided in Sect. 3. In Sect. 4 the simulations and error analysis are presented to clarify and support the main ideas of the paper.

2 Kohonen network

Kohonen has proposed an unsupervised self-organizing training algorithm for generating a mapping of an input signal vector $\mathbf{x}$ from a high dimensional space $\Re^n$ onto a one or two dimensional topological space of a neural network. The map is generated by developing a correspondence between input and signal vectors $\mathbf{x} \in \Re^n$, and weight vectors of neurons, $\mathbf{m} \in \Re^n$, such that the topological relationships of weight vectors faithfully preserve the essential features of the inputs. An element of the neuron weight vector is analogous to the synaptic efficacy which modulates the innervated neural signal received from another neuron. Associated with each synaptic connection, where output of a neuron j innervate neuron i is a weight efficacy m_{ij}, which is referred to as the Long Term Memory (LTM) and is responsible for encoding the input/output association. Kohonen has suggested the following simple form for m_{ij}'s adaptation (Kohonen 1988);

$$\frac{dm_{ij}}{dt} = \alpha(t) y_i (x_j - m_{ij}) \tag{1}$$

Where x_j is the jth element of the input vector, y_i is the output of the ith neuron, m_{ij} is the connection weight and $\alpha(t)$ is the adpatation gain in general. This weight adaptation rule can be written in discrete form and updated at discrete time t_k. Furthermore, the output of the neuron y_i is assigned a binary state, i.e. either the neuron is firing, $y_i = 1$, or is not firing, $y_i = 0$. Hence the weight adaptation is activated for only the neurons whose outputs are high.

The output activation during the learning period is the key concept for the topology preserving network. It is defined in terms of a neighborhood set $N_I(t_k)$. The $N_I(t_k)$ is the set of neurons considered to be in the neighborhood of neuron I at time t_k, i.e. when the neuron I is maximally responding, the topological neighborhood set N_I of neuron I will be selected and activated. N_I is defined to be all neurons which lie within a certain radius from neuron I. Thus all the neurons inside the N_I set are excited with their outputs at constant value 1 when I is selected and outside of the neighborhood set the outputs are zero, i.e. when $y_I(t) = 1$ then

$$y_i(t) = \begin{cases} 1 & i \in N_I \\ 0 & \text{otherwise} \end{cases}$$

We define this neighborhood activation as the neighborhood interaction set. It incorporates two aspects into weight adaptation rule. First a set of topologically neighboring neurons is selected for weight adaptation and secondly all the synaptic weights are updated at the same rate. Equation (1), the learning rule, then can be written as

$$\frac{dm_{ij}}{dt} = \alpha(t)(x_j - m_{ij}) \quad \text{for } i \in N_I(t) \tag{2}$$

$$\frac{dm_{ij}}{dt} = 0 \quad \text{for } i \notin N_I(t) \tag{3}$$

This is referred to as the Kohonen learning law. It can be further written in discrete form and updated at discrete time t_k.

Typically the size of $N_I(t_k)$ starts large and slowly decreases over time. The shape of $N_I(t_k)$ depends on the topology of the network defined. For linear network, for example, where each neuron has two neighbors only a decaying linear array neighborhood is selected. For a two dimensional network, a square array or hexagonal array topological neighborhood shape can be selected. Figure 1 shows examples of such $N_I(t_k)$'s.

The Kohonen learning rule can be written into a step by step procedure (Lippmann 1987) for computer simulation;

Step 1: Initialization; select the size and structure of the network. Initialize $m_{ij}(t_0)$ with small random values. Initialize $N_I(t_0)$ to a large size. Typically the set $N_I(t_0)$ includes all the neurons in the network at the initial time.

Step 2: Present a new input $\mathbf{x}(t_k) \in \Re^n$.

Step 3: Compute the distance between $\mathbf{x}(t_k)$ to all neurons, or the matching scores d_i's,

$$d_i = \|\mathbf{x}(t_k) - \mathbf{m}_i(t_k)\|,$$

where $\|\cdot\|$ is Euclidean norm and $\mathbf{m}_i(t_k)$ is the weight vector of the ith neuron; $\mathbf{m}_i(t_k) \in \Re^n$.

Step 4: Select the Ith neuron closest to $\mathbf{x}(t_k)$, or minimum distance d_I;

$$d_I = \min_i (d_i). \tag{4}$$

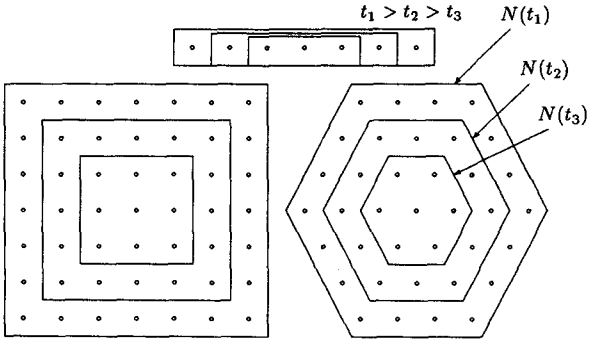

Fig. 1. The examples of topological neighborhood

Step 5: Update weight vectors to the Ith neuron and its neighbors;

$$\mathbf{m}_i(t_{k+1}) = \mathbf{m}_i(t_k) + \alpha(t_k)(\mathbf{x}(t_k) - \mathbf{m}_i(t_k)),$$

otherwise

$$\mathbf{m}_i(t_{k+1}) = \mathbf{m}_i(t_k).$$

where the adaption gain $\alpha(t_k)$ is usually a slowly decreasing function of time and $0 < \alpha(t_k) < 1$.
Step 6: Repeat by going to step 2.

The Kohonen learning rule is similar to the Adaptive Resonance Theory for analog inputs (ART2) developed by Carpenter and Grossberg (1987) except that in the topology preserving networks the neighborhood neurons also learn the input pattern. In the ART2 network only the winner updates its weight vector. In a recent work (Fujita and Bavarian 1991) these two learning concepts are combined to form a new two layer pattern classifier neural network.

The result of the above learning procedure is stated by the following proposition *(Proposition 5.1, Kohonen 1988)*;

"*The $\mathbf{m}_i$ vectors tend to be ordered according to their mutual similarity, and the asymptotic local point density of the $\mathbf{m}_i$, in the average sense, is of the form $f(p(\mathbf{x}))$, where f is some continuous, monotonically increasing function, and $p(\mathbf{x})$ is a stationary probability density function of the input vectors.*"

This proposition has been proved only for some special cases such as when the topology of the network is one dimensional.

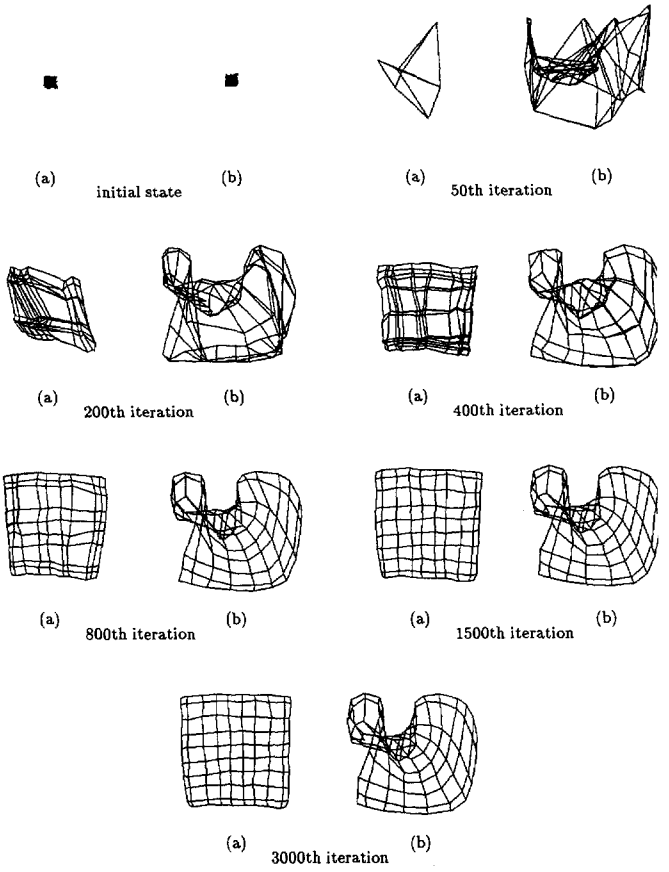

Fig. 2. Training a 10×10 neural network for 3000 iterations **a** Successful global ordering; **b** Unsuccessful global ordering

3 Analysis of the neighborhood selection

The adaptation gain, $\alpha(t_k)$, and neighborhood selection, play critical roles for convergence of the topology preserving mapping network. However, there has not been a formal analysis to show the effects on convergence and to serve as a guide for selecting these parameters. Figure 2 shows two simulations of a two dimensional grid network with 10×10, i.e., 100 neurons. During training two dimensional input vectors, which are points from a unit square with uniform distribution are presented to the two networks. Figure 2 shows the plots of seven intermediate states of the neuron weight vectors. Since the weight vectors are two-dimensional, they are plotted in cartesian coordinates with each point of the grid representing a neuron weight vector. The neighboring neurons (grid points in the graph) are connected as shown in Fig. 2. The learning is carried out for 3000 training vectors. Since the training vectors are from a uniform distribution unit square we expect both networks to converge to uniform grid. The initial values for the weight vectors are the same for both cases. Figure 2 part (a) shows a successful convergence to an approximated grid and Fig. 2 part (b) shows the unsuccessful case. There are mainly three reasons why the network does not evolve into a globally ordered structure:

1. The adaptation gain, $\alpha(t)$, decays too fast.
2. The neighborhood set, $N_I(t_k)$, shrinks too fast.
3. Initial value of $N_I(t_k)$ is too small.

Thus, in order to get an ordered network, $N_I(t_k)$ needs to be selected large initially, and both $N_I(t_k)$, and $\alpha(t)$ decay gradually with time. However, this will result in slow convergence. This raises the issue of optimum selection of the parameters. Furthermore, even if the decay and initial size of the parameters are selected optimally, since all the neurons in the neighborhood update their weight equally, the convergence will be slow. This latter phenomenon is evident in all the simulations reported in previous papers in which extremely large number of training data are used for the network to converge to an ordered structure, and will be formally analyzed later.

We propose a neighborhood interaction function instead of the neighborhood interaction set for the weight adaptation. This closely follows the work by Ritter and Schulten (1986c) in which they have used the Gaussian type neighborhood interaction to find the equilibrium state of the Kohonen network as mentioned earlier. This was also investigated in a certain sense by Cottrell and Fort (1986). Here, the neighborhood interaction function are compared in the context of rate of convergence of the network, and its effect on enforcing topological order.

This is motivated by both anatomical and physiological evidence of the way neurons in mammalian nervous system interact laterally. The interactions are excitatory in the short range reaching up to a radius of 50 to 100 μm. The excitatory area is surrounded by a penumbra of inhibitory action reaching up to a radius of 200 to 500 μm (Kohonen 1988). Furthermore, a neuron which is firing excites the closer neighboring neurons more than the neurons which are further. Kohonen originally defined this type of lateral interaction. However, he used a neighborhood interaction set approximation for simulation of the network in which the neighborhood amplitude of interaction is set to 1 in the neighborhood set, i.e. all the neurons in the neighborhood set were firing at the same rate, so the interaction was independent of lateral distance. This approximate interaction is shown in Fig. 3.

On the other hand, the neighborhood amplitude of interaction can be selected as the Gaussian function to more closely model the anatomical evidence. The amplitude is then defined as a function, not a constant, and for a one dimensional topology is given by

$$A(x, \sigma) = a + b \exp\left(-\frac{cx^2}{2\sigma^2}\right) \tag{5}$$

Where a, b and c are constants. The shape of the neighborhood amplitude interaction is shown in Fig. 4.

First we define the neighborhood interaction with two parameters. One is the neighborhood set which gives the extent of the lateral excitatory interaction, and the second parameter is the amplitude of the interaction. For a neuron which is maximally excited, the set defines the number of additional neighboring neurons around it which will adapt their weight vectors, and the amplitude defines the amount of adaptation.

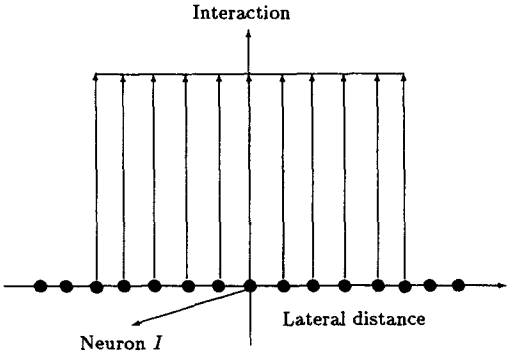

Fig. 3. Constant lateral interaction in one dimensional network

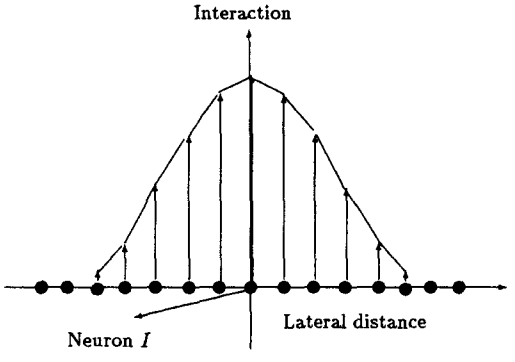

Fig. 4. Gaussian type lateral interaction in one dimensional network

Now we can define the neighborhood interaction set and the neighborhood interaction function formally.

Definition 1. The neighborhood interaction set is defined by a neighborhood set which may stay the same or decay with time, and a neighborhood amplitude of interaction which is unity.

Definition 2. The neighborhood interaction function is defined by a neighborhood set which may stay the same or decay with time, and a neighborhood amplitude of interaction which decreases in spatial domain.

A typical exponential decay function which is often used in the literature for the neighborhood set is given by:

$$N_I(t_k) = a + b \exp(-ct_k), \tag{6}$$

Where a, b and c are some positive constants. $N_I(t_k)$ is the same for both neighborhood interaction set and the neighborhood interaction function. However, the amplitude of the interaction is unity for the neighborhood interaction set; i.e.

$$A_I(i, t_k) = 1, \quad \text{for } i \in N_I(t_k) \tag{7}$$

and it can be selected as a Gaussian type function as defined in (5) for the neighborhood interaction function; i.e.

$$A_i(i, t_k) = c + d \exp\left(-\frac{h(i-I)^2}{2\sigma^2}\right), \quad \text{for } i \in N_I(t_k) \tag{8}$$

Where c, d and h are constants, and the variance σ is set equal to the neighborhood set, i.e.

$$\sigma = N_I(t_k) \tag{9}$$

The parameters in (8) are selected such that $0 < A_I < 1$ at all times. Notice that this is a decaying function in spatial domain for neighboring neurons and also in time domain its variance, or domain of action, decreases which provides for the focusing property of the network. Now the learning rule in the Kohonen algorithm can be written more generally as:

$$\mathbf{m}_i(t_{k+1}) = \mathbf{m}_i(t_k) + \alpha(t_k) A_I(i, t_k)[\mathbf{x}(t_k) - \mathbf{m}_i(t_k)] \quad \text{for } i \in N_I(t_k) \tag{10}$$

$$\mathbf{m}_i(t_{k+1}) = \mathbf{m}_i(t_k) \quad \text{for } i \notin N_I(t_k) \tag{11}$$

and depending on whether one uses neighborhood interaction set or neighborhood interaction function the proper $A_I(i, t_k)$ is selected.

Next, we present the formal analysis of the neighborhood interaction for enforcing the topological order in the network. Before stating the theorems the network topological order is defined:

Definition 3. Topologically ordered network: Let $\mathbf{x}$ be an

input to a network and pick any $I \in \Upsilon$, a neuron which is closest to $\mathbf{x}$, i.e.

$$\Upsilon = \{i | \forall j \|\mathbf{x} - \mathbf{m}_i\| \leq \|\mathbf{x} - \mathbf{m}_j\|\}$$

The network is topologically ordered if for any two neurons i_1 and i_2 in the neighborhood of neuron I satisfy

$$\|\mathbf{x} - \mathbf{m}_{i_1}\| < \|\mathbf{x} - \mathbf{m}_{i_2}\|$$

when i_1 is closer to I than i_2 topologically.

Thus a network is ordered if the order in the neurons topological space is similar to the order in the neurons weight vector space. Figures 5 and 6 help to visualize the two cases of ordered and disordered network. The following two theorems presents the main ideas of the paper.

Theorem 1. *The learning rule of (10) and (11), using a neighborhood amplitude interaction which is decreasing in spatial domain enforces the topological order in the neighborhood set for every iteration.*

We prove the theorem by calculating and comparing the percentage of the change in the norm of weight vectors during an adaptation step.

Proof. Let us consider two neurons i_1 and i_2 in the neighborhood of neuron I whose weight vector is closest to the current input vector $\mathbf{x}(t_k)$ as defined in Definition 3. For neuron i_1, the weight adaptation results in:

$$\mathbf{m}_{i_1}(t_{k+1}) = \mathbf{m}_{i_1}(t_k) + \alpha(t_k)A_I(i_1, t_k)[\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_k)]$$

To find the percentage change in weight adaptation, both sides are subtracted from $\mathbf{x}(t_k)$ and terms are

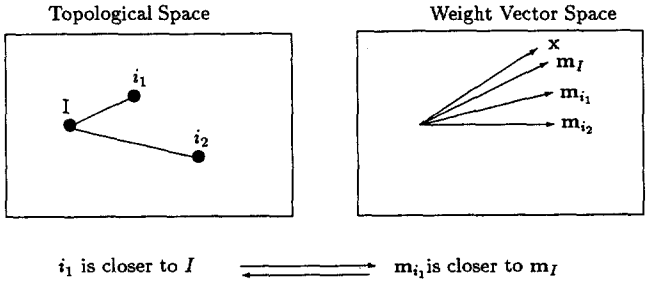

Fig. 5. Illustration of ordered neurons

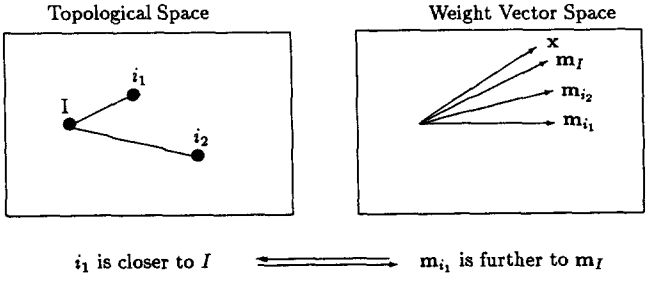

Fig. 6. Illustration of disordered neurons

rearranged to yield

$$\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_{k+1}) = (1 - \alpha(t_k)A_I(i_1, t_k))(\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_k)) \quad (12)$$

Define the two distances of the weight vector before and after adaptation to the input $\mathbf{x}(t_k)$ by

$$d_{i_1,\text{before}} = \|\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_k)\|$$

$$d_{i_1,\text{after}} = \|\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_{k+1})\|$$

Then taking the norm of both sides of (12) and since $0 < \alpha(t_k) \leq 1$, $0 < A_I(i_1, t_k) \leq 1$, we have

$$d_{i_1,\text{after}}/d_{i_1,\text{before}} = 1 - \alpha(t_k)A(i_1, t_k)$$

or

$$\gamma_{i_1} = 100(d_{i_1,\text{before}} - d_{i_1,\text{after}})/d_{i_1,\text{before}} = 100\alpha(t_k)A(i_1, t_k) \quad (13)$$

where γ_{i_1} is the percentage of change in the l^2-norm of the neuron weight vectors in one adaptation iteration for the neuron i_1. Similarly for i_2 we have

$$\gamma_{i_2} = 100(d_{i_2,\text{before}} - d_{i_2,\text{after}})/d_{i_2,\text{before}} = 100\alpha(t_k)A(i_2, t_k) \quad (14)$$

Now comparing (13) and (14), if i_1 is closer to I than i_2 then from (8) neighborhood interaction amplitude function

$$A(i_1, t_k) > A(i_2, t_k)$$

hence

$$\gamma_{i_1} > \gamma_{i_2} e$$

i.e., the percentage change in the weight vectors which are topologically closer to I in the activated neighborhood is larger than those which are further and vice versa. Hence for each $\mathbf{x}(t_k)$, training iteration the topological ordered structure is enforced by adapting the weight vectors of the topologically closer neurons more than the neurons which are far apart. Thus, the theorem is proved.

Notice that regardless of the initial state of the network using neighborhood interaction function enforces the topological order. So even in the disordered case due to the above property the order is achieved after several training iterations. This is demonstrated in the simulations later on. The next theorem complements the first theorem in proving that once the topological order is established the neighborhood interaction function helps to preserve it.

Theorem 2. *If the network is topologically ordered, then the learning rule of (10) and (11) with a neighborhood amplitude of interaction which is decreasing in spatial domain will preserve the order.*

Proof. For an input vector $\mathbf{x}(t_k)$, according to the nearest match, the Ith neuron is chosen, then the weight vectors of the neurons in its neighborhood at time t_{k+1} are updated by (10) and (11). Since explicit values of the weight vector adaptations are needed, let's

examine the Euclidean distance between $\mathbf{x}(t_k)$ and $\mathbf{m}_i(t_{k+1})$, the updated weight vector, i.e.

$$\begin{aligned} d_i &= \|\mathbf{x}(t_k) - \mathbf{m}_i(t_{k+1})\| \\ &= \|\mathbf{x}(t_k) - (\mathbf{m}_i(t_k) + \alpha(t_k)A_I(i, t_k)[\mathbf{x}(t_k) - \mathbf{m}_i(t_k)])\| \\ &= \|\mathbf{x}(t_k) - \mathbf{m}_i(t_k)\|(1 - \alpha(t_k)A_I(i, t_k)) \end{aligned} \tag{15}$$

Then, the difference of the Euclidean distances between two neurons i_1 and i_2 in the neighborhood of the winner neuron I is given by,

$$\begin{aligned} d_{i_1} - d_{i_2} = &\|\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_k)\|[1 - \alpha(t_k)A_I(i_1, t_k)] \\ &- \|\mathbf{x}(t_k) - \mathbf{m}_{i_2}(t_k)\|[1 - \alpha(t_k)A_I(i_2, t_k)], \end{aligned} \tag{16}$$

Since the network is ordered, two cases may occur:

Case 1: If

$$\|\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_k)\| > \|\mathbf{x}(t_k) - \mathbf{m}_{i_2}(t_k)\|,$$

and

$$A_I(i_1, t_k) < A_I(i_2, t_k),$$

since $0 < \alpha(t_k) \leqslant 1$, $0 < A_I \leqslant 1$ and the norm part is positive, then

$$\begin{aligned} \|\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_k)\|[1 - \alpha(t_k)A_I(i_1, t_k)] > &\|\mathbf{x}(t_k) \\ &- \mathbf{m}_{i_2}(t_k)\|[1 - \alpha(t_k)A_I(i_2, t_k)], \end{aligned}$$

thus

$$A_I(i_1, t_k) < A_I(i_2, t_k) \Rightarrow d_{i_1} > d_{i_2}$$

i.e. the order is preserved.

Case 2: If

$$\|\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_k)\| < \|\mathbf{x}(t_k) - \mathbf{m}_{i_2}(t_k)\|,$$

and

$$A_I(i_1, t_k) > A_I(i_2, t_k),$$

since $0 < \alpha(t_k) \leqslant 1$, $0 < A_I \leqslant 1$ and the norm part is positive, then

$$\begin{aligned} \|\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_k)\|[1 - \alpha(t_k)A_I(i_1, t_k)] < &\|\mathbf{x}(t_k) \\ &- \mathbf{m}_{i_2}(t_k)\|[1 - \alpha(t_k)A_I(i_2, t_k)], \end{aligned}$$

thus

$$A_I(i_1, t_k) > A_I(i_2, t_k) \Rightarrow d_{i_1} < d_{i_2}$$

again the order is preserved and the theorem is proved.

That is, the weight vectors are arranged in such a way that a topologically closer neuron to the Ith neuron has an updated weight vector closer to $\mathbf{m}_I$ in weight vector space compared to far-off neurons. As the weight vectors self-organize into the shape of the input distribution, the topologically close neurons maintain "close" weight vectors in l^2 sense and at the same time the neighborhood shrinks containing those neurons with "close" weights.

Theorem 3. *If all the weight vectors in the neighborhood set are equal, then the learning rule of (10) and (11) with a neighborhood amplitude of interaction which is decreasing in spatial domain establishes the order in one iteration.*

Proof. For any two neurons i_1, and i_2 in the neighborhood set, let

$$\mathbf{m}_{i_1}(t_k) = \mathbf{m}_{i_2}(t_k) = \mathbf{s},$$

Then,

$$\|\mathbf{x}(t_k) - \mathbf{m}_{i_1}(t_k)\| = \|\mathbf{x}(t_k) - \mathbf{m}_{i_2}(t_k)\| = \|\mathbf{x}(t_k) - \mathbf{s}\|,$$

and from (16)

$$d_{i_1} - d_{i_2} = \|\mathbf{x}(t_k) - \mathbf{s}\|\alpha(t_k)[A_I(i_2, t_k) - A_I(i_1, t_k)] \tag{17}$$

Since the norm part and $\alpha(t_k)$ are positive

$$A_I(i_1, t_k) > A_I(i_2, t_k) \rightarrow d_{i_1} > d_{i_2} \tag{18}$$

and

$$A_I(i_1, t_k) < A_I(i_2, t_k) d_{i_1} < d_{i_2} \tag{19}$$

i.e., the order is established in one iteration and the theorem is proved.

Corollary 1. *The learning rule of (10) and (11) using a neighborhood amplitude of interaction which is decreasing in spatial domain will have faster convergence than using a neighborhood interaction set when the neighborhood set decay function is the same for both.*

The proof follows the theorem 1. The percentage of change in the norm of weight vectors for the training algorithm using neighborhood interaction function amplitude is given by (13) and (14). For the learning rule with neighborhood interaction set (13) and (14) become

$$\gamma_{i_1} = \gamma_{i_2} = 100\alpha(t_k)$$

i.e. the percentage of change in the norm of the weight vectors for all neurons in the neighborhood set are the same, while using neighborhood interaction function with decreasing amplitude in spatial domain enforces the order in every training iteration, hence it rates up the self-organization. Furthermore in case of symmetry when all the weight vectors are initially equal, i.e.

$$\mathbf{m}_{i_1} = \mathbf{m}_{i_2} = \mathbf{s}$$

and for neighborhood set fixed to the size of the network, after any number of adaptation, the distances for any two neuron stay the same according to (16) i.e.

$$d_{i_1} = d_{i_2}$$

for all t_k and order is not formed at all, when using the neighborhood interaction set, while from Theorem 1 and 3 the order is established using neighborhood interaction function only in one iteration.

The above argument may raise the issue of how the Kohonen network converges to the equilibrium state with the neighborhood interaction set as has been shown in the literature. The answer is basically in Kohonen proposition 5.1 (Kohonen 1988) stated earlier, and due to the shrinking neighborhood set in time. Since a group of neurons weight vector are adapted each time and for every training iteration randomly different groups are selected, statistically in time the

order is formed through a very large number of training iterations. Using a neighborhood interaction function with decreasing amplitude in spatial domain not only have this characteristic but also as proved above enforces and preserve topological order in every iteration.

In summary selecting neighborhood interaction function enforces the topological ordered structure in the weight space in every training sample which results in faster convergence as compared to the neighborhood interaction set which relies on large number of training samples to form the order structure. In the next section, several simulations are presented to show these properties clearly.

4 Simulations

We have carried out many simulations to investigate the rate of convergence comparing the learning rule with neighborhood interaction function and the neighborhood interaction set. All the simulations show that the network using the neighborhood interaction function has faster convergence than the network using the neighborhood interaction set. Here we present two cases to illustrate various characteristics discussed in the last section.

The learning adaptation gain $\alpha(t)$ is usually assigned a decaying function in time to ensure the termination of training. It is set to be the same for all the simulations here.

Case 1

In this case two simulations are presented to show the convergence of the network using the neighborhood interaction function when the initial weight vectors are all set equal to the same value, i.e., the condition of complete weight vector symmetry. The neighborhood set is also fixed in all the simulation to the size of the network, i.e.

$$N_I(t_k) = \textit{size-of-networks} \tag{20}$$

Thus, the only difference in the simulations is the neighborhood amplitude interaction. The first simulation is training an 8×8 two dimensional square topology network. The training vectors are chosen from a uniform distribution over a letter Y shape. Figure 7 part (a) shows the plot of the weight vectors of the neurons for the network using the neighborhood interaction function. The simulation of the network using the neighborhood interaction set shows no change at all due to the weight symmetry as discussed in the corollary 1. The second example is an 8×8 two dimensional square topology network. The training vectors are chosen from a uniform distribution over a triangle. Again only the network using the neighborhood interaction function converges to the ordered structure, as shown in Fig. 7 part (b).

Case 2

In this case the initial weight vectors are assigned randomly and not equal. The neighborhood set is selected to be a slow exponential decay function for all

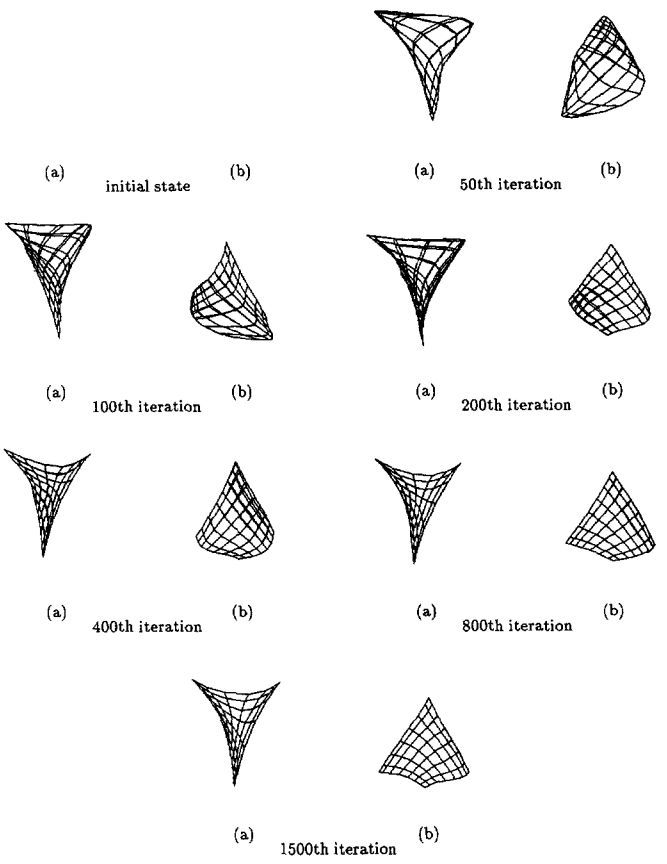

Fig. 7. Training an 8×8 neural network for 1500 iterations; **(a)** input letter Y shape distribution with fixed neighborhood set radius; **(b)** input triangular shape distribution with fixed neighborhood set radius

the simulations and the network topology is set to be square. Again the only difference in the simulation is the neighborhood interaction amplitude. Following the simulations presented by Kohonen (1988), the training samples are chosen from a uniform distribution over a unit square. The first example is for a 10×10 network and the second one is for a 32×32 network. The first example part (a) uses the neighborhood interaction set and part (b) uses the neighborhood interaction function. The training is performed for 1500 samples. Figure 8 shows the plots of seven intermediate states of the weight space of the 10×10 network. Figure 9 shows the same simulation for the 32×32 network. Since the input vectors are from a uniform distribution over a unit square, we expect the weight space of the network self-organize into a perfect square grid. As can be seen from the Figures, part (b), with the neighborhood interaction function, forms the square grid faster than part (a) with the neighborhood interaction set. However, a better quantitative measure is needed to compare the two. We define the sum of the square of errors of each neuron weight vector compared to the ideal square grid position to be the measure. Let's define the ideal square grid coordinates to be (μ_{i1}, μ_{i2}) and $\mathbf{m}_i = (m_{i1}, m_{i2})$, then the error measure is defined by

$$\textit{error}(t_k) = \sum_i [(m_{i1}(t_k) - \mu_{i1})^2 + (m_{i2}(t_k) - \mu_{i2})^2] \tag{21}$$

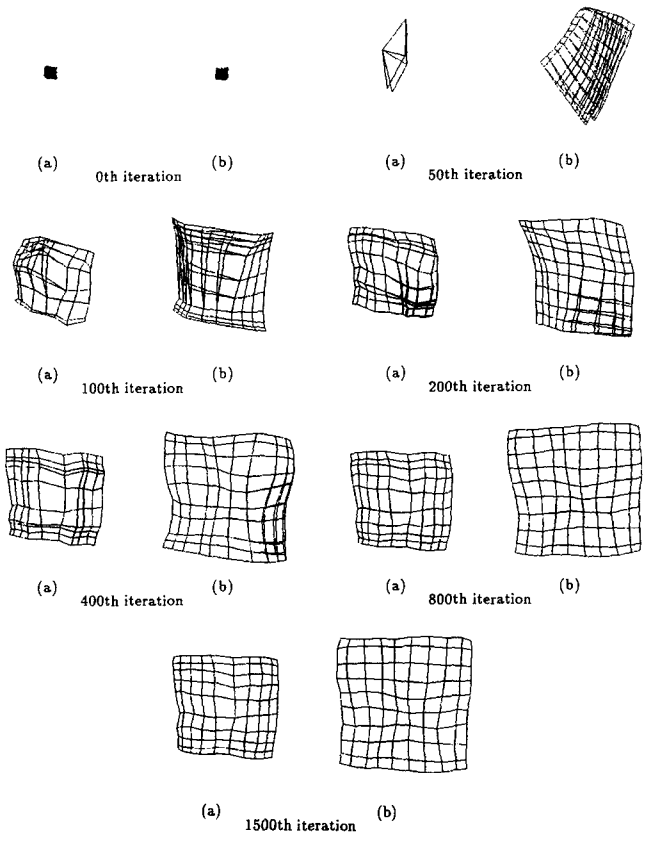

Fig. 8. Training a 10×10 neural network for 1500 iterations; (**a**) with the neighborhood interaction set; (**b**) with the neighborhood interaction function

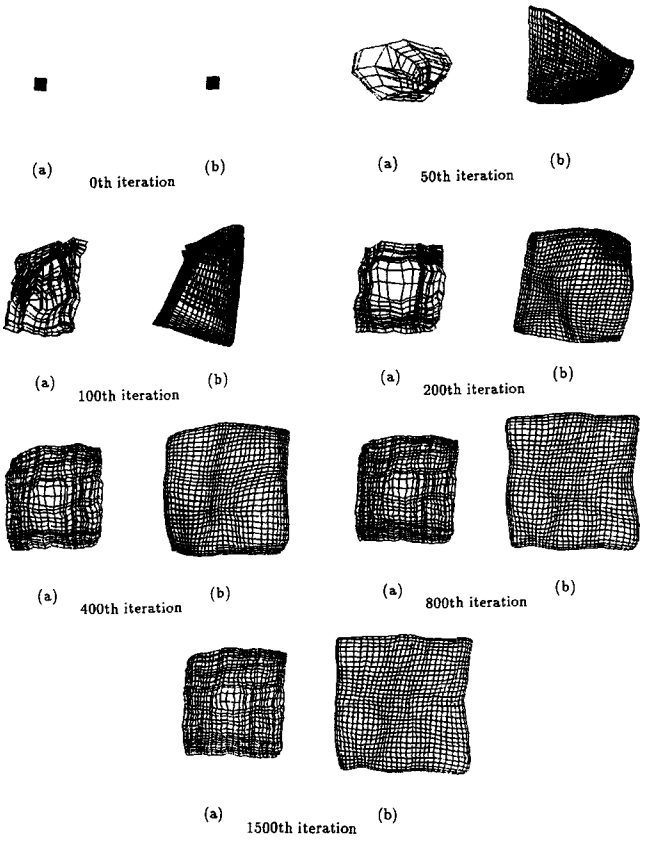

Fig. 9. Training a 32×32 neural network for 1500 iterations; (**a**) with the neighborhood interaction set; (**b**) with the neighborhood interaction function

Figures 10 and 11 show the error which is plotted as the function of iteration for the 10×10 and 32×32 networks respectively. In each figure two error measures are plotted: one for the network using the neighborhood interaction function and one for the neighborhood interaction set. From these graphs, it is clear that the former case converges faster. Furthermore, towards the end, when both weight spaces are almost close to the ideal square grid, the network with the neighborhood interaction set has substantial error. This is mainly due to the slow expansion of the network. The boundary neuron weight vectors are still away from the ideal position they have to approach.

We carried out many other simulations with differ-

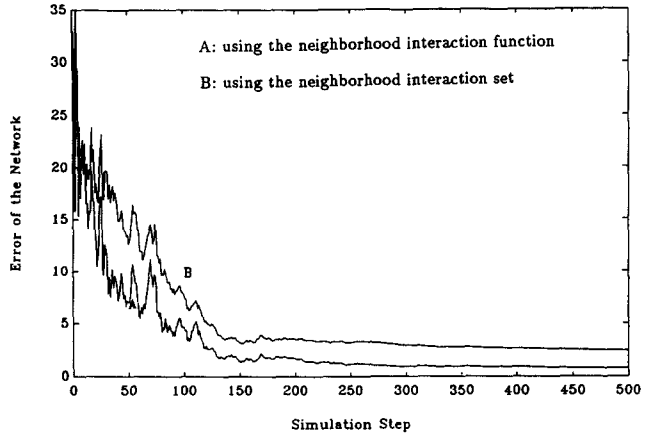

Fig. 10. Error of the 10×10 neural network

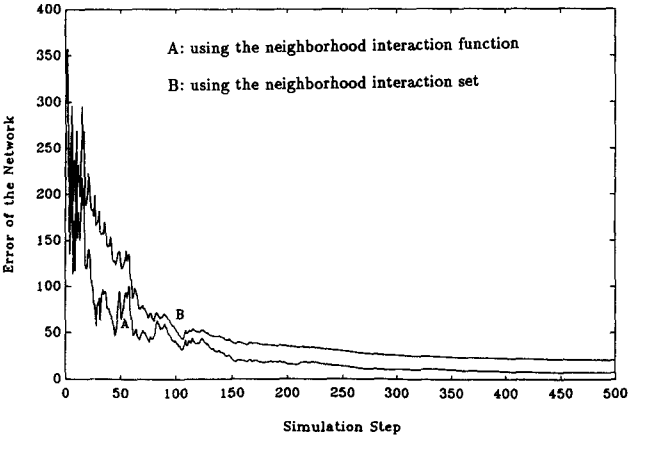

Fig. 11. Error of the 32×32 neural network

ent network dimensions and topology such as one dimensional network with linear array neighborhood topology, and two dimensional network with linear array neighborhood topology. All the simulations have shown fast convergence by using neighborhood interaction function.

5 Conclusion

We have investigated the selection of neighborhood interaction function in topology preserving neural networks. It is formally shown that selecting a neighborhood interaction function amplitude which decreases in spatial domain will enforce topological order self-organization of the neuron weight space as compared to the reported approaches which use the neighborhood interaction set. Formal analysis and illustrative simulations to support the analysis are presented.

There are several aspects of the Kohonen learning rule which need further study. Here we have only dealt with the effect of the neighborhood amplitude of interaction. Further analysis of the effect of the neighborhood set shrinking will shed light on the proper selection of this parameter.

There is a trade-off between rate of convergence to the equilibrium state and error of "closeness" of the equilibrium state to the intrinsic distribution of the input. This raises the question of optimum selection of the parameters, e.g., desirable neighborhood amplitude of interaction which are subjects of future research. Currently we are investigating a statistical energy approach to the problem. Also we are studying the different phenomena of these neural networks reported by Kohonen (1982) such as magnification factor, boundary effects, the "Pinch" phenomenon, the "Collapse" phenomenon and the "Focussing" phenmonenon using the formal analysis and simulations of this paper. The fact is that the Kohonen self-organizing network works and it has many application domain which need to be explored further.

References

Carpenter GA, Grossberg S (1987) ART2: self-organization of stable catagory recognition codes for analog input patterns. Appl Opt 26:4919–4930

Cottrell M, Fort JC (1986) A stochastic model of retinotopy: a self organizing process. Biol Cybern 53:405–411

Fujita M, Bavarian B (1991) An ART2-TPM neural network for pattern classification. IEEE Trans Neural Net (submitted for publication)

Kohonen T (1982a) Self-organized formation of topologically correct feature maps. Biol Cybern 43:59–69

Kohonen T (1982b) Clustering, taxonomy, and topological maps of patterns. Sixth Interational Conference on Pattern Recognition, Munich, Germany, October 19–22, 1982, pp 114–128

Kohonen T (1982c) Analysis of simple self-organizing process. Biol Cybern 44:135–140

Kohonen T (1984) Phonotopic maps – insightful representation of phonological feature for speech recognition. Proceedings of the Seventh International Conference on Pattern Recognition, Montreal, Canada, July 30–August 2, 1984, pp 114–128

Kohonen T (1988) Self-organization and associative memory. Springer Series in Information Sciences. Springer, Berlin Heidelberg New York

Lippmann P (1987) An introduction to computing with neural nets. IEEE ASSP Magazine 4–22

Ritter H, Schulten K (1986a) Topology conserving mapping for learning motor tasks. AIP Conference proceeding 151, Snowbird Utah, pp 376–380

Ritter H, Schulten K (1986b) Extending Kohonen's self-organizing mapping algorithm to learn ballistic movements. In: Eckmiller R, von der Malsburg C (eds) Neural computers. Springer, Berlin Heidelburg New York, pp 393–406

Ritter H, Schulten K (1986c) On the stationary state of Kohonen's self-organizing sensory mapping. Biol Cybern 54:99–106

Ritter H, Schulten K (1988) Convergence properties of Kohonen's topology conserving maps: fluctuations, stability, and dimension selection. Biol Cybern 60:59–71

Takeuchi A, Amari S (1979) Formation of topographic maps and columnar microstructures in nerve fields, Biol Cybern 35:63–72

Willshaw DJ, Malsburg C von der (1976) How patterned neural connections can be set up by self-organization. Proc R Soc London B194:431–445

Willshaw DJ, Malsburg C von der (1979) A marker induction mechanism for the establishment of ordered neural mapping: its application to the retinotectal problem. Proc R Soc London B287:203–243

Prof. Behnam Bavarian
Department of Electrical Engineering
University of California, Irvine
Irvine, CA 92717, USA