

# On the Issue of Neighborhood in Self-Organizing Maps

††Hua Yang, ‡M. Palaniswami

‡La Trobe University, Victoria, Australia

†The University of Melbourne, Victoria, Australia

## Abstract

The neighborhood relation in a self-organizing neural network is discussed. A friendship concept is introduced to describe the connection structure of the network. In particular, a self-organizing neural network with high dimensional lattice connection is given with a convergence result for the associated learning algorithm.

## Introduction

Self-organizing maps[5] can discover the topological relations and other abstract structures in the input signals. It is extremely effective in creating spatially organized representation of the various features in the input patterns on a one- or two-dimensional array of neurons [7]. The idea of self-organizing topological map was proposed originally by Willshaw and von der Malsburg [11], [12]. The convergence properties and dynamical stability of the Willshaw and Malsburg model was analyzed by Amari and Takeuchi [1] and [10]. Kohonen[5] generalized this idea and proposed a simplified model in which the topologically correct feature map reveals abstract and conceptual structures in the input patterns. In [7], the simulation results demonstrate the usage of the feature maps in the self-construction and self-ordering of the neural maps. An example is the self-organizing formation of a frequency map which provides a model for the tonotopic map[7]. Some convergence results of the self-organizing algorithms (SOA) are in [9] and [8]. In this paper, we will consider some issues regarding the neighborhood relation in the self-organizing maps and present some new results on the convergence of SOA.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1992 ACM 0-89791-502-X/92/0002/0412...\$1.50

## Structure of self-organizing map

Let  $N$  be a set of  $n$  formal neurons which is called representation layer or neural network in this paper. Let  $U$  be the input sample space which is an  $m$ -dimensional space  $R^m$  and there is an array of  $m$  input nodes. There is a connection weight between every input node and every neuron in  $N$ . These connection weights can be modified by using Kohonen self-organizing algorithm[7].

Suppose that  $N$  is a subset of a metric space  $M$  and the neurons in  $N$  are numbered with an index  $i$ . The activity of the  $i$ -th neuron in  $N$  is  $f_i$ . Let  $(w_{ij})_{n \times m}$  be the connection weight matrix between input nodes and the neurons in  $N$ . Given a connection matrix, we have a map from  $R^m$  to  $R^n$ :  $F = (f_1(u), \dots, f_n(u))$  which is the response of the neurons in  $N$  corresponding to the input pattern  $u$ . After the model is trained by using Kohonen self-organizing algorithm [6], this map can find important features in the input patterns and preserve the neighborhood relations in the input space, where the concept of the neighborhood relation in the space  $N$  is specified by the distance in the space  $M$ .

## Friendship and neighborhood

For the purpose of discussing friendship and neighborhood relation in  $N$ , it is useful to observe that there are connections among the neurons in  $N$ . The connection structure of the neural network  $N$  can be described by a directed graph  $G = (V, E)$ , where  $V = \{v_i\}$  is  $N$ . The directed edge from the vertex  $v_i$  to the vertex  $v_j$  is represented by an ordered pair  $(v_i, v_j)$ . Here, it should be noted that there are two kinds of directed edges connecting vertices: excitatory edge and inhibitory edge. We can also define the length of the edge  $(v_i, v_j)$ , which is denoted by  $|(v_i, v_j)|$ , to be the distance  $d(v_i, v_j)$  between the points  $v_i$  and  $v_j$  in  $M$ . The connection cost of the directed graph  $G$  is defined by the following summation:

$$C(G) = \sum_{(v_i, v_j) \in E} d(v_i, v_j).$$

If only the connection structure is considered, there is no need to consider the locations of the neurons. But, when the connection cost is considered, we must consider the positions of the neurons in the space  $M$ . In order to decrease the connection cost, one can rearrange the vertices. However, there are several constraints in rearranging the vertices, such as: keeping the same connection structure in the rearranged network, and leaving certain distance between any two neurons because each neuron will take some physical space in the layout of the network. The optimal layout is the one with minimum connection cost.

There are five kinds of connection patterns as shown in Figure 1. We call the one-edge pattern as a single-connection and the two-edge pattern as a double-connection.

In this paper, two neurons are called friends if they are linked by a double-connection. This defines the friendship relation between two neurons. To minimize the connection cost, the friendly neurons should be put as close as possible in the layout of the network because there are two edges in one double-connection.

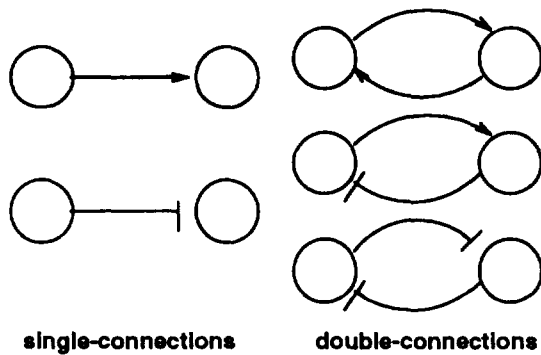


Figure 1: Connection patterns

If all single-connections are deleted from the graph  $G$ , we may get a disconnected graph  $G'$ . A maximal connected subgraph of the graph  $G'$  is called its component. In each component of the graph  $G'$ , the neurons are linked by double-connections. Two neurons will be in the same component of  $G'$  if they are friends. Taking each component as one vertex, we can form a new graph  $G'' = (V'', E'')$  from  $G'$ . In  $G''$ ,  $V'' = (v_i'')$  are all components of  $G'$ . An edge  $e_{ij}'' = (v_i'', v_j'')$  in  $G''$  is a bundle of single-connections which link the neurons in  $v_i''$  and the ones in  $v_j''$ . The length of the edge  $e_{ij}''$  is defined as  $\sum_{v_k \in v_i'', v_l \in v_j''} |(v_k, v_l)|$ , which is the summation of all lengths of the edges in the bundle. One way to minimize the total connection cost for  $G$  is to do it for each component first then do it for the induced graph  $G''$ . This method may not find the optimal solution in minimizing the connection cost for  $G$ , but it can find a satisfactory one.

In the following sections, we assume that  $G'$  has only one component. Graphically, an edge without direction represents a double-connection. Given a layout of the network  $G$  in the space  $M$ , the neighborhood of a neuron is a small region around it. All neurons in the region are its neighbors. A neuron may have more than one nearest neighbor in the space  $M$ .

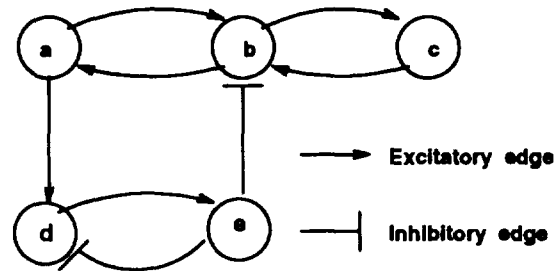


Figure 2: Friends and non-friends

In Figure 2, the node a and b, b and c, and d and e are friends, but a and d, and b and e are not friends. There are two components in this graph: one formed by the nodes a, b and c, and the other formed by the nodes d and e.

In Figure 3, the friendship and neighborhood relation coincide, but these two relations are different in Figure 4.

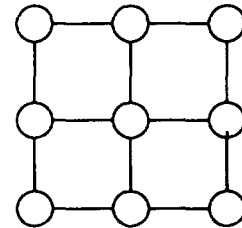


Figure 3: Neighbors are also friends

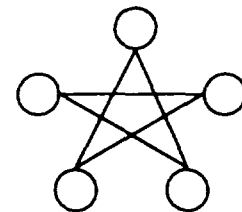


Figure 4: Neighbors are not friends

The friendship relation is important for the training of the self-organizing map as it brings the cooperation

of the neurons into the network.

## Neighborhood on a lattice

In this section, we introduce a lattice connection structure in which the two relations, friendship relation and the neighborhood relation coincide.

The neighborhood relation in the input space  $U$  is well defined because  $U$  is a metric space. If there is a distance in space  $M$ , we can define the neighborhood relation. However, the distance in space  $M$  can be defined in several ways. For example, if the neurons in  $N$  are arranged in a one-dimensional array we can use an integer index  $i$  to number the neurons as  $N = \{0, 1, \dots, n-1\}$ . In this case,  $N$  is called an one-dimensional lattice. The distance between  $i$ -th neuron and  $j$ -th neuron is defined by  $|i - j|$ . Two neurons are neighbors if the distance between them is 1. In the one-dimensional lattice, every neuron has two neighbors except the neurons on the boundary which have only one neighbor. Similarly, if the neurons in  $N$  are arranged into a two-dimensional array, which is called two-dimensional lattice, then each internal neuron which is not on the boundary has four neighbors. A pair of integers  $(i, j)$  is used to label the neurons. The distance between neuron  $(i, j)$  and  $(k, l)$  is  $|i - k| + |j - l|$ .

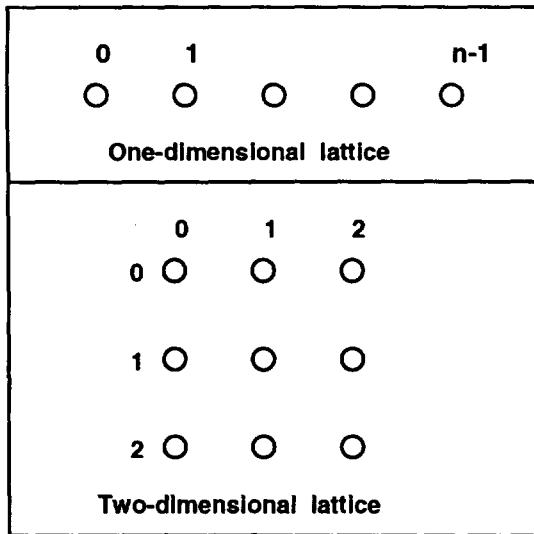


Figure 5: One- and two-dimensional lattice

In these two cases, the neighborhood relation is determined by the positions of the neurons in space  $M$ .

If some neurons need to have more than four neighbors, then we need higher dimensional lattice to represent the neighborhood relation. Let us assume that there are  $n = L^k$  neurons in  $N$ . Then, the  $k$ -dimensional lattice is defined by the following:

$$N = \{(i_1, \dots, i_k) : 0 \leq i_1, \dots, i_k < L\}.$$

We can also define the distance on the lattice  $N$ . If a neuron is an inner node, it has  $2k$  neighbors in a  $k$ -dimensional lattice.

The friendship relation or the connection structure of  $N$  in one-, two-, three- and four-dimensional lattices is characterized by the graphs in Figure 6.

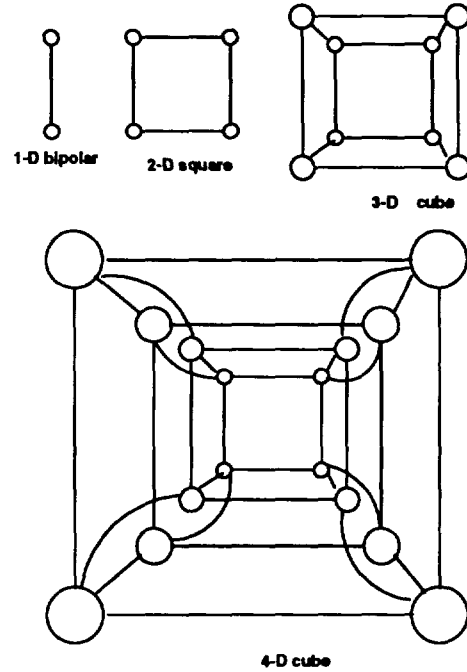


Figure 6: Connections on the cubes

Suppose that there is a connection structure of  $N$  which can be represented by a directed graph. If a node in this graph fires, its activity will be propagated to the nodes which have connections from this node. The behavior of a neuron is strongly affected by its friends. In the Kohonen algorithm, if one neuron becomes active, then the weights pointing to this active neuron and its friends will be updated after competition. The cooperation effect is brought into the system in this way.

Suppose that the neurons are physically located in some space  $M$  in which the connection structure specifies the friendship relation. Without changing the connection structure or the friendship relation, any network on  $M$  can be rearranged into one-dimensional array, the 1-D layout of the neurons in  $M$ . In Figure 7, two graphs represent the same connection structure. However, the 1-D layout needs much more connection cost. For non-linear connections, it is better to pack the neurons into two- or three-dimensional array to minimize the connection cost.

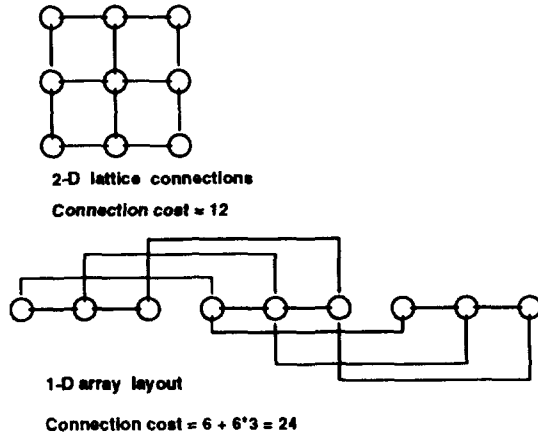


Figure 7: 1-D layout for a two-dimensional lattice

In the following sections, we will assume that  $N$  is a  $k$ -dimensional lattice. For each neuron  $i$  in  $N$ , its neighborhood is a set  $N_i$ . It is a set value function from  $N$  to  $2^N$  which is used in the self-organizing algorithm. We will call the function  $N_i$  a neighborhood function.

### Self-organizing algorithm

For each node  $i$  in  $N$ , there is a weight vector  $W_i = (w_{i1}, \dots, w_{im})$  pointing to the neuron  $i$ , where  $m$  is the dimension of the input space  $U$ . Figure 8 shows the structure of the self-organizing map. The connection plate realizes the lattice connection in  $N$

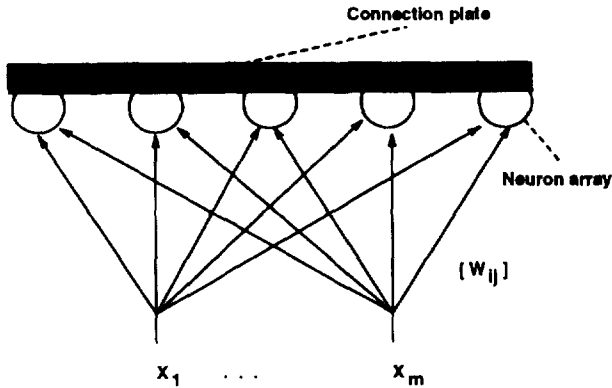


Figure 8: The structure of the self-organizing map

Let  $N_c$  be a neighborhood set around a neuron  $c$  which is an active node in  $N$  at the moment. The Kohonen self-organizing algorithm is :

$$W_i^{t+1} = \begin{cases} W_i^t, & i \notin N_c, \\ (1 - \lambda(t))W_i^t + \lambda(t)S(t), & i \in N_c, \end{cases} \quad (1)$$

where  $S(t) = (s_1(t), \dots, s_m(t))^T$  is the input vector. In this paper, if  $Y$  is a vector or matrix,  $Y^T$  means transposition of  $Y$ .

In this learning algorithm, if a weight vector  $W_i^t$  points to the active node or its neighbors in  $N_c$  at time  $t$ , then this vector will be updated at the time  $t+1$ .

Let us assume that only one neuron is active at time  $t$ . Denote  $N(t) = N_c$  if the neuron  $c$  is active at time  $t$ . We call  $N(t)$  the neighborhood movement. We can change the neighborhood function  $N_i$  to improve the training. But we can not control the neighborhood movement, as it depends on the sequence of the inputs and the initial connection weights  $(w_{ij})$ . In this paper, we assume that the learning rate  $\{\lambda(t)\}$  satisfies the following:

$$\lambda(t) > 0, \quad \sum_t \lambda(t) = \infty, \quad \text{and} \quad \sum_t \lambda^2(t) < \infty, \quad (2)$$

### Convergence of the algorithm

The system expressed in equation (1) can be considered for each input dimension. However, here we only analyze the convergence of this algorithm with the input  $S(t)$  being in a one-dimensional space.

In (1),  $(w_i^t)$  is a random process because the input process  $S(t)$  is . In order to get convergence of the algorithm, we need the following convergence result for the Robbins-Monro algorithm [3]:

**Gladyshev's Theorem** (see Theorem 2.2 in [3]) Assume  $h(x) : R^n \rightarrow R^m$  is a function with a unique zero point  $x^0$ , i.e.  $h(x^0) = 0$ . Let  $Y^{t+1} = h(X^t) + n_{t+1}$  be the observed value of the function  $h(\cdot)$  at  $X^t$  with a random error  $n_{t+1}$ . The approximation of  $x^0$  is  $X^{t+1} = X^t - \lambda(t)Y^{t+1}$ , where  $\{\lambda(t)\}$  satisfies (2). Suppose there is a positive definite matrix  $U$ , such that  $(x - x^0)^T U h(x) > 0$ ,  $\forall x \neq x^0$ , and

$$|h(x)|^2 + E|n_t|^2 \leq K(1 + |x|^2),$$

where  $E|n_t|^2$  is a mathematical expectation and  $K$  is a constant. Then  $X^t \rightarrow x^0$  a.s. as  $t \rightarrow \infty$ .

Assume that the representation layer  $N$  is a  $k$ -dimensional lattice,

$$N = \{(i_1, \dots, i_k) : 0 \leq i_1, \dots, i_k < L\}.$$

Define an index map  $l$  from  $\{0, 1, \dots, L-1\}^K$  to  $\{0, 1, \dots, L^k-1\}$  as

$$l(i_1, \dots, i_k) = i_1 * L^{k-1} + i_2 * L^{k-2} + \dots + i_k.$$

By using this index map, we can put a  $k$ -dimensional array  $(x_{i_1, \dots, i_k})$  into a vector

$(x_{0,0,\dots,0}, x_{0,0,\dots,1}, \dots, x_{i_1,\dots,i_k}, \dots, x_{L-1,\dots,L-1})$ . Denote this vector as  $\vec{x}$  or  $(x_{i_1,\dots,i_k})$ . The  $l(i_1, \dots, i_k)$ -th component of the vector  $\vec{x}$  is  $x_{i_1,\dots,i_k}$ . If the neuron  $c = (i_1, \dots, i_k)$  is active at time  $t$ , let  $N(c)$  denote its neighborhood. We define a  $k$ -dimensional array  $G = (g(i_1, \dots, i_k))$ , where  $g(i_1, \dots, i_k) = 1$  if  $(i_1, \dots, i_k)$  is in  $N(c)$ , otherwise  $g(i_1, \dots, i_k) = 0$ .

Let  $(I_1(t), \dots, I_k(t))$  be the movement of the active node. Set  $G(t) = (g(I_1(t), \dots, I_k(t)))$ . The equation (1) can be rewritten as

$$\vec{w}^{t+1} = \vec{w}^t - \lambda(t)H(t)\vec{w}^t + \lambda(t)S(t)\vec{G}(t),$$

where  $H(t)$  is a diagonal matrix. The diagonal of  $H(t)$  is the vector  $\vec{G}(t)$  whose elements are zero or one. Both  $H(t)$  and  $\vec{G}(t)$  are stochastic processes because they depend on the movement of the active node. Let the elements on the diagonal be  $h_0(t), h_1(t), \dots, h_{L^k-1}(t)$ .

Assume

$$\prod_{i=0}^{L^k-1} Eh_i(t) > 0,$$

where the  $Eh_i(t)$  is the mathematical expectation of  $h_i(t)$ . This assumption is true if the active node moves evenly on  $N$ . In addition, suppose the input process  $S(t)$  is stationary, then we can prove that  $\vec{w}^t$  is convergent to a vector  $\xi$  which is a solution to the following equation:

$$EH(t)\xi = ES(t)\vec{G}(t).$$

Here the meaning of the convergence is the almost sure convergence of stochastic process.

We can prove this by applying Gladyshev's Theorem. Set  $U = EH(t)$  and  $b = ES(t)\vec{G}(t)$ , let  $\xi = U^{-1}b$ ,  $\vec{w}^{t+1} = \vec{w}^t - \lambda(t)\vec{Y}^{t+1}$ , where  $\vec{Y}^{t+1} = H(t)\vec{w}^t - S(t)\vec{G}(t) = U\vec{w}^t - b + n_t$ , where  $n_t = (H(t) - EH(t))\vec{w}^t + (S(t)\vec{G}(t) - ES(t)\vec{G}(t))$ . Choose  $h(x) = Ux - b$ .  $\xi$  is the unique zero point of the function  $h(x)$ . Then  $(x - \xi)^T U h(x) = (x - U^{-1}b)^T U (Ux - b) = (Ux - b)^T (Ux - b) > 0$ , for any  $x \neq \xi$ . So by Gladyshev's Theorem,  $\vec{w}^t \rightarrow \xi$ , a.s., as  $t \rightarrow \infty$ , where a.s. means almost sure convergence.

## Conclusions

The friendship and neighborhood relations are introduced in this paper to characterize connection structure of the representation layer in the self-organizing maps. The lattice structure is introduced in such way that the two relations coincide. When the connection structure in the representation layer is described by a lattice, the convergence result is given for the Kohonen self-organizing algorithm.

## References

- [1] S. Amari (1980), Topographic Organization of Nerve Fields. *Bull. Math. Biol.* 42, 339-364.
- [2] G.A. Carpenter and S. Grossberg (1987), ART2: Self-organization of Stable Category Recognition Codes for Analog Input Patterns. Vol. 26, No. 23, *Applied Optics*, pp. 4919-4930.
- [3] Han-Fu Chen (1985), Recursive Estimation and Control for Stochastic Systems, John Wiley & Sons, Inc.
- [4] M. Cottrell and J. C. Fort (1986), A Stochastic Model of Retinotopy: A Self-organizing Process, *Biol. Cybern.* 53, 405-411.
- [5] T. Kohonen (1982a), Self-organized Formation of Topologically Correct Feature Maps, *Biol. Cybern.* 43, 59-69.
- [6] T. Kohonen (1990), The self-organizing Map, *Proceedings of The IEEE*, Vol. 78, No. 9, September.
- [7] T. Kohonen (1989), Self-organization and Associative Memory, Third Edition, Springer-Verlag.
- [8] Chung-Ming Kuan and Kurt Hornik, Convergence of Learning Algorithms with Constant Learning Rates, *IEEE Tran. on Neural Networks*, Vol. 2, No. 5, September 1991.
- [9] H. Ritter and K. Schulten (1989), Convergence properties of Kohonen's topology conserving maps: fluctuations, stability, and dimension selection, *Biol. Cybern.* 60, pp. 59-71.
- [10] A. Takeuchi and S. Amari, Formation of topographic maps and columnar microstructures, *Biol. Cybernetics*, vol. 35, pp. 63-72, 1979.
- [11] D.J. Willshaw and C. von der Malsburg (1976), How patterned neural connections can be set up by self-organization, *Proc. Roy. Soc.*, vol. B-194, pp. 431-445.
- [12] D.J. Willshaw and C. von der Malsburg (1979), A marker induction for the establishment of ordered neural mapping: its application to the retino-tectal problem. *Philos. Trans. R. Soc. London*, 287: B1021.