Parallel Manner and Twist Measurement for Self-Organizing Maps

Michiharu Maeda† Hiromi Miyajima‡

†Kurume National College of Technology
Kurume, 830-8555 Japan

‡Faculty of Engineering, Kagoshima University
Kagoshima, 890-0065 Japan

Abstract: - This paper investigates a self-organizing map algorithm which is simultaneously given multiple inputs, and introduces a criterion of ordering process as the twist measurement for reference vectors in multidimensional array. The self-organizing algorithm and the criterion are termed the parallel self-organizing algorithm and the index of twist, respectively. The algorithm updates reference vectors corresponding to respective inputs at the same time, when multiple inputs are prepared at each step. The index of twist is the criterion to evaluate the multidimensional ordering of the topological array for reference vectors. When the parallel degree changes for the present algorithm, the topology preserving map after learning is evaluated by utilizing the index of twist. By discussing the formation rate and the average distortion of topology preserving map, the effectiveness of the present approach is examined.

Key-Words: - Neural networks, Self-organizing map, Parallel manner, Twist measurement Learning, Numerical experiment

1 Introduction

For neural networks, there exist supervised and unsupervised learnings [1]. Back-propagation algorithm [2] for the former and Kohonen's self-organizing map algorithm [3] for the latter are well known. Kohonen's self-organizing map is the neural network which adaptively classifies input patterns given to the first layer, and topologically organizes output nodes, in the two-layer structure [4]. The algorithm sequentially deals with the unit in the second layer which responds to an input data randomly chosen at each step. The algorithm is termed the sequential self-organizing algorithm and its network, the sequential model. In the application of self-organizing algorithm, for example, the traveling salesman problem can be solved in excellent results by taking the elastic ring method [5],[6]. In the meantime, the parallel network which permits multiple elements to transit at each step is discussed for Hopfield network [7]. It leads to different results against the sequential operation [8]. Concerning the parallel processing, the speed-up of the self-organizing map is attempted by the arrangement on linear chain and two-dimensional mesh of the transputer [9]. In the parallel vector quantization, a fine result is shown in the average distortion, in which the network has the partial and zero neighboring relations of reference vectors. In this algorithm, almost equivalent results to the sequential operation has been obtained [10]. However, the parallel manner of topology preserving map is not argued, and the convergence of reference vectors is not mentioned.

In this paper, a parallel self-organizing map algorithm which is simultaneously given multiple inputs is presented. The algorithm updates reference vectors corresponding to respective inputs at the same time, when multiple inputs are prepared at each step. Furthermore, an index of twist is introduced in order to evaluate the ordering process for reference vectors. The index is the criterion to evaluate multidimensional ordering of the topological array. The index of twist becomes zero if the topology is perfectly preserved. When the parallel degree changes for the present algorithm, the topology preserving map after learning is evaluated by using the index of twist. For one input and multiple inputs at each step, though the adaptation process generally yields different results in neural networks, the self-organizing algorithm produces almost same results for the present algorithm. By discussing the formation rate and the average distortion of topology preserving map, the effectiveness of the present approach is examined.

2 Self-Organizing Map by Kohonen Model

In Kohonen's algorithm, the updating of reference vectors is modified to involve neighboring relations in the output array. In the vector space R^n, the input $\boldsymbol{x}$, which is generated on the probability density function $p(\boldsymbol{x})$, is defined. Thus, $\boldsymbol{x}$ has components from x_1 to x_n. The output unit y_i is generally arranged in an array of one- or two-dimensional maps, and is completely connected to the inputs by way of w_{ij}.

Let $\boldsymbol{x}(t)$ be an input vector at step t and let $\boldsymbol{w}_i(0)$ be reference vectors composed of k at initial values in R^n space. For input vector $\boldsymbol{x}(t)$, we calculate the distance between $\boldsymbol{x}(t)$ and the reference vector $\boldsymbol{w}_i(t)$, and select the reference vector as a winner c minimizing the distance. The process is written as follows:

$$c = \arg\min_i\{\|\boldsymbol{x} - \boldsymbol{w}_i\|\}, \tag{1}$$

where arg$(\cdot)$ gives an index c of the winner.

With the use of the winner c, the reference vector $\boldsymbol{w}_i(t)$ is updated as follows:

$$\Delta\boldsymbol{w}_i = \begin{cases} \alpha(t)\,(\boldsymbol{x} - \boldsymbol{w}_i) & (i \in N_c(t)), \\ \mathbf{0} & \text{(otherwise)}, \end{cases} \tag{2}$$

where $\alpha(t)$ is the learning rate and is a decreasing function of time ($0 < \alpha(t) < 1$). $N_c(t)$ has a set of indexes of topological neighborhoods for the winner c at step t. If $N_c(t)$ has an index of the winner only, the algorithm becomes the standard competitive learning.

3 Parallel Manner According to Self-Organizing Map

In this section, a parallel self-organizing algorithm is presented. The algorithm updates reference vectors corresponding to respective input vectors at the same time when multiple input vectors are prepared at each step. To begin with, let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_l\}$ be the set of l input vectors selected at step t, and let $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_m\}$ be the set of reference vectors had ready beforehand. The set of winner vectors $\{c_1, c_2, \cdots, c_l\}$ corresponding to respective input vectors are selected as follows:

$$c_i = \arg\min_j\{\|\boldsymbol{x}_i - \boldsymbol{w}_j\|\}. \tag{3}$$

Using the winner c_i, the reference vector $\boldsymbol{w}_i$ is updated at each step according to the following equation.

$$\Delta\boldsymbol{w}_i = \sum_{j=1}^{l} \boldsymbol{v}_{ij}, \tag{4}$$

where the updating value $\boldsymbol{v}_{ij}$ of the reference vector $\boldsymbol{w}_i$ is given as follows:

$$\boldsymbol{v}_{ij} = \begin{cases} \alpha(t)\,(\boldsymbol{x}_j - \boldsymbol{w}_i) & (i \in N_{c_i}(t)), \\ \mathbf{0} & \text{(otherwise)}, \end{cases} \tag{5}$$

where $N_{c_i}(t)$ is the set of the winner c_i and its topological neighborhood reference vectors, and is a decreasing function of time.

In the above updating process, since not only a winner and its neighborhood vectors but also multiple winners and their neighborhood vectors are updated at each step, reference vectors at each step are calculated as the composition of their vectors. If $c_i = c_j$, the value $\boldsymbol{w}_i$ is also updated as the composition under the influence of their vectors. Then we term it the parallel self-organizing algorithm (or the parallel updating process) and its network, the parallel model.

In the updating process, one step of the above parallel model corresponds to l steps of the sequential model. Therefore, the maximum speed-up is at l times in the parallel model if it is possible to obtain similar results such that in the sequential model, where l is the number of input vectors. Here, we define that l is possible to be changed into some numbers, where l is smaller than or equal to the number of reference vectors prepared in advance. Therefore we propose another parallel model which updates only l reference vectors corresponding to $\lfloor m/h \rfloor$ pieces of input data at each step, where h is a positive integer, $\lfloor m/h \rfloor$ is a reasonable number to perform the parallel process at a time, and $\lfloor m \rfloor$ means a maximum integer smaller than or equal to a real number m. Then we also term it the parallel self-organizing algorithm and its network, the parallel model.

[Parallel self-organizing algorithm]

Step 1 Initialization:
Give initial reference vectors $\{\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_m\}$, neighborhood set N_i for each reference vector $\boldsymbol{w}_i$, and parallel degree l. Set $t \leftarrow 0$.

Step 2 Selection of input vectors:
Select the set of input vectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_l\}$ in the input space at random.

Step 3 Determination of winners:
Determine the winner c_i corresponding to $\boldsymbol{x}_i$ $(i = 1, 2, \cdots, l)$, according to Eq. (3).

Step 4 Update of reference vectors:
Update $\boldsymbol{w}_i$ according to Eqs. (5) and (4), by using c_i as determined in Step 3. Set $t \leftarrow t + 1$.

Step 5 Termination Condition:
If $t = T_{max}$ then terminate, otherwise go to Step 2.

Table 1: Sequential and parallel models.

Sequential		$l = 1$	i
Parallel	A	$1 < l < m$	ii
	B	$l = m$	iii

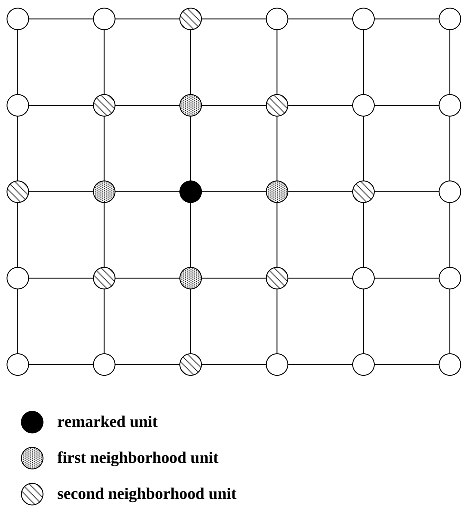


Figure 1: Concept of the index of twist.

According to the above-mentioned discussion, three models are considered, as shown in Table 1. "Sequential model" is the conventional technique dealt with sequentially. For "Parallel model", let A be the case where l has any size of m $(l = \lfloor m/h \rfloor)$, and let B be the case where l has all numbers of m $(l = m)$. The case of i is well known as Kohonen's model. Then, we will consider models i, ii, and iii in the following section.

4 Twist Measurement

For evaluating self-organizing map formed by learning, two measures for reference vectors are considered as follows [3]:

I Ordering stage.

II Convergence phase.

The degree of ordering is expressed as the "index of disorder" in terms of the way of I [11]. For the way of II, asymptotic distributions are theoretically discussed as the adaptive vector quantization [12]–[16]. After reference vectors become ordered, their final convergence are discussed. In this section, we consider the way of I. Since the conventional index of disorder [4] is adjusted to one-dimensional case only, the expansion is not found to multi-dimensional case so that the index observes the perfect ordering of reference vectors.

In order to evaluate the degree of ordering for reference vectors in the updating process, we define a positive integer C as follows:

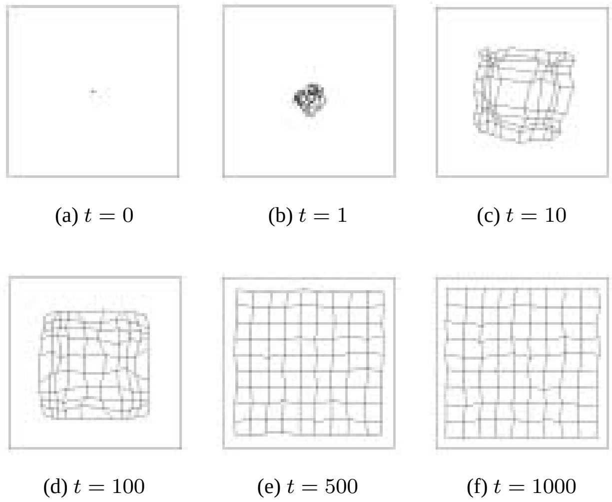
$$C = \sum_{i=1}^{n} H(d_{ij} - d_{ik} > 0) \quad j \in N_i^1, k \notin N_i^2, \tag{6}$$

where $d_{ij} = \|\boldsymbol{w}_i - \boldsymbol{w}_j\|$. N_i^1 and N_i^2 are the topological neighborhood sets of vector $\boldsymbol{w}_i$, and $H(\theta)$ is an indicator function as follows:

$$H(\theta) = \begin{cases} 1 & \theta \text{ is true,} \\ 0 & \theta \text{ is false.} \end{cases}$$

We call C an "index of twist." In case of the square arrangement as shown in Fig. 1, N_i^1 and N_i^2 are the set to the first neighborhood and the set to the second neighborhood, respectively. In this figure, each point shows reference vector and the line which connects $\boldsymbol{w}_i$ with $\boldsymbol{w}_j$ means the nearest neighborhood to each other. N_i^1 is the set composed of the remarked vector and its first neighborhood vectors, because the square arrangement is considered in this case. N_i^2 is added the set of the second neighborhood vectors to N_i^1. Then there is a region composed of the remarked vector $\boldsymbol{w}_i$ and its first neighborhoods. The index of twist shows the amount in which other reference vectors enter this region, except for the remarked, its first neighborhood, and its second neighborhood vectors. When the configuration of reference vectors are changed, the ordering process may be shown in the index of twist by employing N_i^1 and N_i^2 suitably.

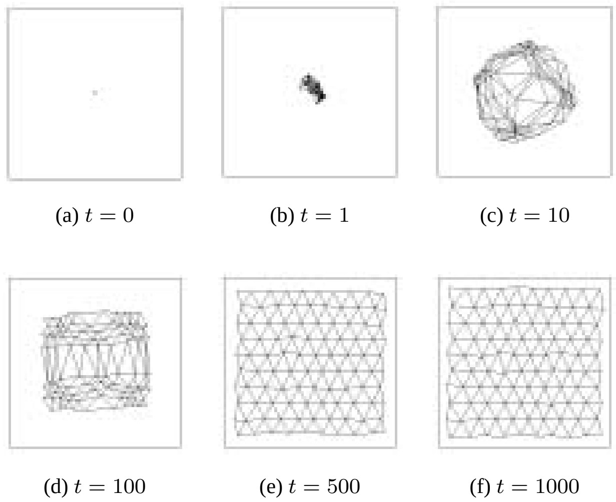
Any value of C is always greater than or equal to zero ($C \geq 0$). If reference vectors are completely ordered, then $C = 0$ holds. And if there are topological defects in the updating process, a value C in proportion to the rate of them is given. Almost any of C will decrease in the learning progress. Therefore, it is considered as an index which generalized the index of disorder. C is a good criterion to show the ordering process of reference vectors.



(a) $t = 0$ (b) $t = 1$ (c) $t = 10$

(d) $t = 100$ (e) $t = 500$ (f) $t = 1000$

Figure 2: Movement of reference vectors in the rectangle input space for the parallel model ($l = 100$).



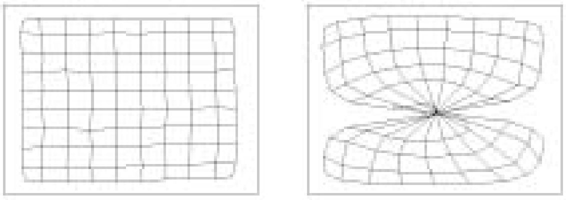
(a) $t = 0$ (b) $t = 1$ (c) $t = 10$

(d) $t = 100$ (e) $t = 500$ (f) $t = 1000$

Figure 3: Movement of reference vectors in the hexagonal input space for the parallel model ($l = 100$).

5 Numerical Experiments

For the numerical experiments, input patterns are uniformly prepared from the entire spece. Input vectors are randomly assigned with $[0, 1]$ on



(a) Topological map (TM) (b) Non-topological map (NTM)

Figure 4: Figures to explain of the topological map and non-topological map.

Table 2: Number of TM for each model.

Models	l	T_{max}	Number of TM	
			Rect.	Hexa.
Sequential	1	100000	942	936
Parallel	10	10000	929	919
	100	1000	914	906

the x and y axes, and reference vectors are distributed around the central space as initial values at random. The performance comparisons are executed for each model described in the previous section. The ordering process of reference vectors for each model is evaluated by the index of twist. Here, the number of reference vectors is 100 (i.e., $m = 100$), and the parallel degree l is an integer. As discussed above, l will be decided for each model. The initial topological neighborhood $N(0)$ has a set of indexes the first to the fifth topological neighborhoods including the winner. $N(t)$ is gradually decreased with time, and has a winner only after the half of the maximum learning iterations. Then the parameters are chosen as follows: $\alpha(t) = \alpha_0(1 - t/T_{max})$, $\alpha_0 = 0.05$, and $T_{max} = 100000/l$.

The movements of reference vectors in the rectangle and hexagonal arrangements for the parallel model ($l = 100$) are shown in Figs. 2 and 3, respectively. As learning progresses, reference vectors approximate the input space and form their maps in each arrangement.

Here we define a topological map (denoted by TM), in detail, it is the "maximally ordered" state map [17] or the topology preserving map [18]. Furthermore we define a non-topological map (denoted by NTM) which has the "topological defects" [1]. Two examples, TM and NTM, are shown in Fig. 4. Table 2 shows the number of TM, which is performed on 1000 trials for initial values given at random, for each of the sequential and parallel ($l = 10$ and 100) models. This table gives notice that about a same number of TM is obtained in both arrangements. However, there seems to be general trend that the number of TM decreases with a greater parallel degree.

Figure 5 shows the values for the index of twist versus the learning iterations. The models are (a) sequential and (b) parallel models in the rectangle arrangement, (c) sequential and (d) parallel models in the hexagonal arrangement. For all the models, the value of the index of twist rapidly drops at first, and gradually approaches zero afterwards. After the ordering ends, reference vectors approximate the input space.

For evaluating the approximate accuracy, the average distortion is presented as follows:

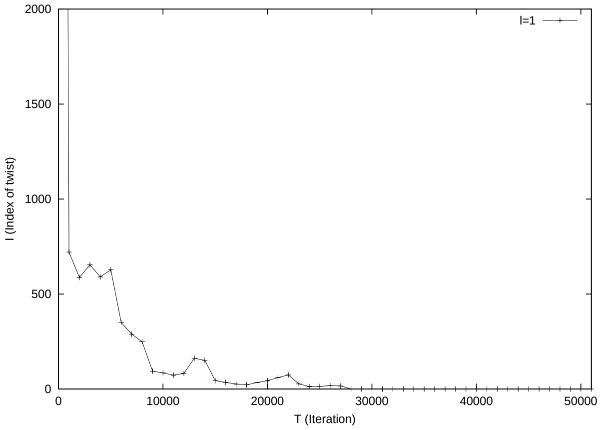
$$E = \frac{1}{M}\sum_{i=1}^{k} D_i, \tag{7}$$

where M is the total number of input vectors, and the i-th partition error D_i is given by the following equation:

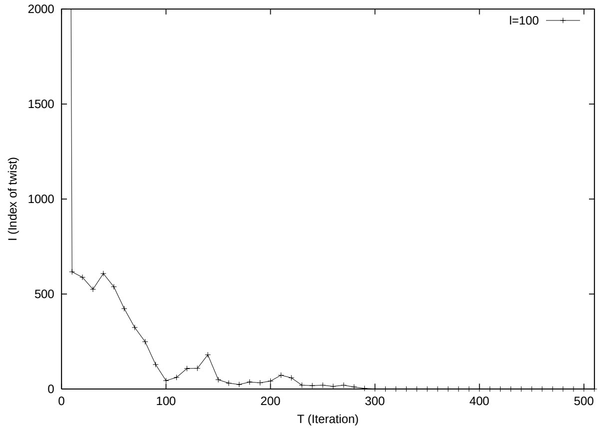
$$D_i = \frac{1}{n}\sum_{\boldsymbol{x}\in S_i} d(\boldsymbol{x}, \boldsymbol{w}_i), \tag{8}$$

where n is the dimension of the input vector, and $d(\boldsymbol{x},\boldsymbol{w}_i)$ is the square of the Euclidean distance between the input vector $\boldsymbol{x}$ and the reference vector $\boldsymbol{w}_i$ (i.e., $d(\boldsymbol{x},\boldsymbol{w}_i) = \|\boldsymbol{x}-\boldsymbol{w}_i\|^2$).

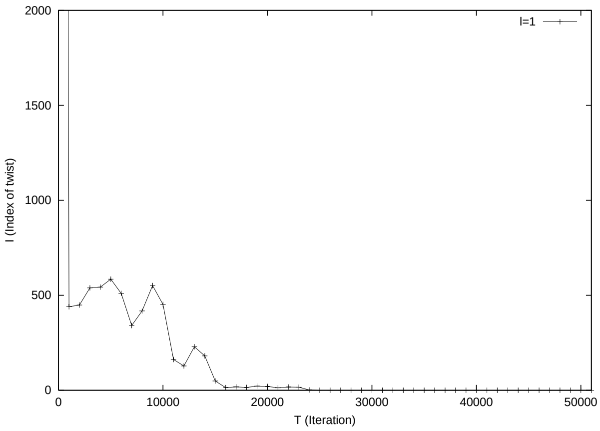
In Fig. 6, the average distortion in the rectangle and the hexagonal arrangements are shown when the parallel degree changes. Here, we obtained the average distortion when the number of input vectors requires 5000 chosen from the input space randomly. The results are the averages of 1000 trials. In this case, it is proven that the average distortion does not change almost, even if the parallel degree changes. It is also proven that



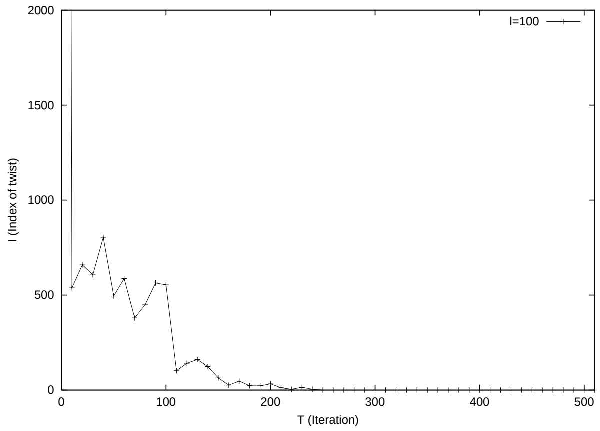
(a) Sequential ($l = 1$) in rectangle



(b) Parallel ($l = 100$) in rectangle



(c) Sequential ($l = 1$) in hexagonal



(d) Parallel ($l = 100$) in hexagonal

Figure 5: Index of twist versus learning iteration for each model. The models are (a) sequential and (b) parallel models in the rectangle arrangement, (c) sequential and (d) parallel models in hexagonal arrangement.

the average distortion is the hexagonal arrangement is lower than that in the rectangle arrangement. This means that the rectangle arrangement is not always optimum for the approximation of the input space. For the standard deviation of subdistortions, however, the hexagonal arrangement rises from the rectangle arrangement, and the dispersion of the hexagonal arrangement is intensified as shown in Fig. 7.

Figure 8 shows the number of TM in the rectangle and hexagonal arrangements. The topology preserving map is formed at the probability of about 90%, even if the parallel degree changes.

The average distortion in the rectangle and hexagonal arrangements in changing the initial value of the topological neighborhoods to 10 from 5 in Fig. 9. In this case, the topology preserving maps are formed at 100%. Unlike in the case of Fig. 6, the average distortion rises as a whole, and the tendency of the average distortion in the rectangle and hexagonal arrangements is also different. It seems that neighboring effects are strong and lead to excellent results for the formation of the topology preserving maps, but they bring on reverse effects for the average distortion.

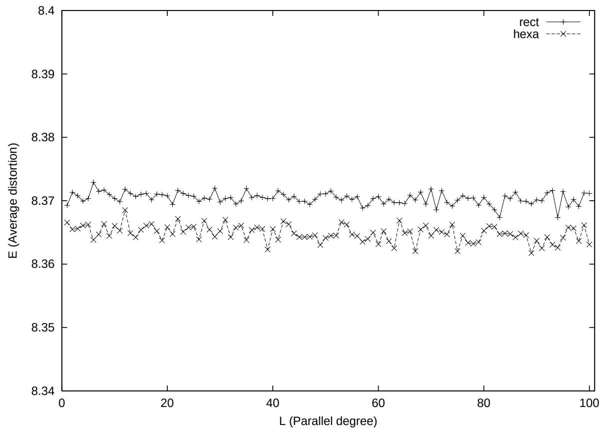


Figure 6: Average distortion of the rectangle and hexagonal neighborhood arrangements. The results are the averages of 1000 trials, randomly chosen among 5000 data.

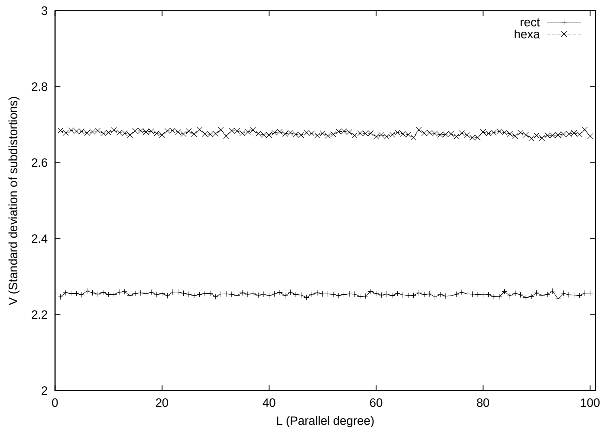


Figure 7: Standard deviation among sub-distortions of the rectangle and hexagonal neighborhood arrangements.

6 Conclusions

In this paper, we have presented the parallel self-organizing models. In the parallel model, multiple input vectors are given at each step and reference vectors are updated as the composition of their vectors under the influence of all vectors. Specifically, we have evaluated the ordering process for each model, making use of the index of twist. As a result, it is shown that Kohonen's self-organizing algorithm for the feature map can also be performed in the parallel manners. Though the adaptation process for one input and multiple inputs at each step generally yields different results in neural networks, it is useful that the self-organizing algorithm produces almost same results. For the future works, we will study theoretical considerations and practical applications of the present models.

References

[1] T. Geszti, *Physical models of neural networks*, World Scientific, 1990.

[2] J. Hertz, A. Krogh, and R.G. Palmer, *Introduction to the theory of neural computation*, Addison-Wesley, 1991.

[3] T. Kohonen, *Self-organization and associative memory*, Springer-Verlag Berlin, 1989.

[4] T. Kohonen, The self-organizing map, *Proc. IEEE*, vol.78, 1990, pp.1464–1480.

[5] R. Durbin and D. Willshaw, An analogue approach to the traveling salesman problem using an elastic net method, *Nature*, vol.326, 1987, pp.689–691.

[6] B. Angéniol, G. de La C. Vaubois, and J.-Y. Le Texier, Self-organizing feature maps and the traveling salesman problem, *Neural Networks*, vol.1, 1988, pp.289–293.

[7] E. Goles, F. Fogelman, and D. Pellegrin, Decreasing energy functions as a tool for studying threshold networks, *Discrete Appl. Math.*, vol.12, 1985, pp.261–277.

[8] H. Miyajima, S. Yatsuki, and M. Maeda, Some characteristics of higher order neural networks with decreasing energy functions, *IEICE Trans. Fundamentals*, vol.E79-A, 1996, pp.1624–1629.

[9] C.-H. Wu, R.E. Hodges, and C.-J. Wang, Parallelizing the self-organizing feature map on multiprocessor systems, *Parallel Comput.*, vol.17, 1991, pp.821–832.

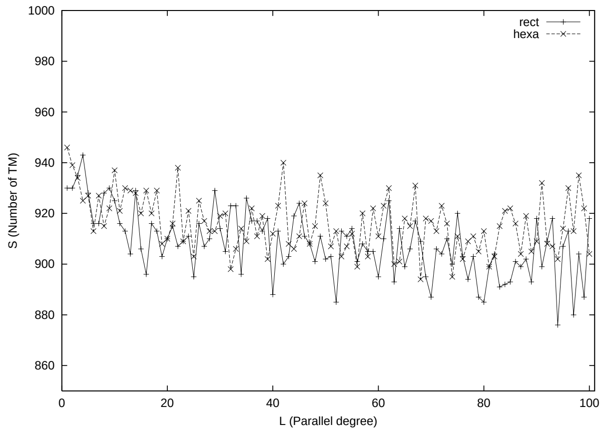


Figure 8: Number of TM in the rectangle and hexagonal neighborhood arrangements.

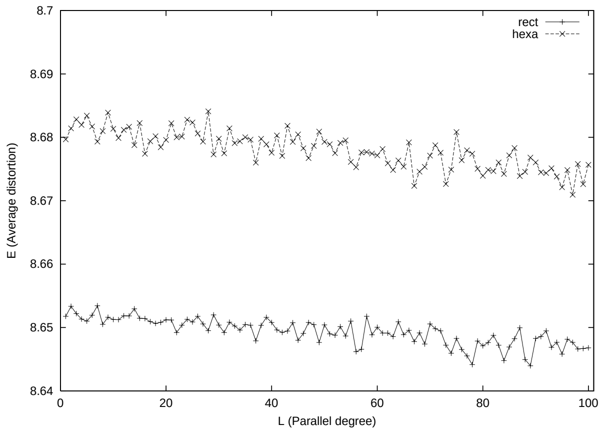


Figure 9: Average distortion of the rectangle and hexagonal neighborhood arrangements.

[10] K.K. Parhi, F.H. Wu, and K. Genesan, Sequential and parallel neural network vector quantizers, *IEEE Trans. Comput.*, vol.43, 1994, pp.104–109.

[11] T. Kohonen, Analysis of a simple self-organizing process, *Biol. Cybern.*, vol.44, 1982, pp.135–140.

[12] Y. Yamada, S. Tazaki, and R. Gray, Asymptotic performance of block quantizers with difference distortion measures, *IEEE Trans. Inform. Theory*, vol.26, 1980, pp.6–14.

[13] P.L. Zador, Asymptotic quantization error of continuous signals and the quantization dimension, *IEEE Trans. Inform. Theory*, vol.28, 1982, pp.139–149.

[14] H. Ritter and K. Schulten, On the stationary state of Kohonen's self-organizing sensory mapping, *Biol. Cybern.*, vol.54, 1986, pp.99–106.

[15] H. Ritter and K. Schulten, Convergence properties of Kohonen's topology conserving maps, Fluctuations, stability, and dimension selection, *Biol. Cybern.*, vol.60, 1988, pp.59–71.

[16] M. Maeda and H. Miyajima, Competitive learning methods with refractory and creative approaches, *IEICE Trans. Fundamentals*, vol.E82–A, 1999, pp.1825–1833.

[17] H. Ritter, T. Martinetz, and K. Schulten, Neural computation and self-organizing maps, an introduction, *Addison-Wesley*, 1992.

[18] T. Martinetz and K. Schulten, Topology representing networks, *Neural Networks*, vol.7, 1994, pp.507–522.