Lazy Self-Organizing Map and its Behaviors

Taku Haraguchi, *Student member, IEEE*, Haruna Matsushita, *Student member, IEEE*,
and Yoshifumi Nishio, *Member, IEEE*

Abstract— The Self-Organizing Map (SOM) is a famous algorithm for the unsupervised learning and visualization introduced by Teuvo Kohonen. This study proposes the Lazy Self-Organizing Map (LSOM) algorithm which reflects the world of worker ants. In LSOM, three kinds of neurons exist: worker neurons, lazy neurons and indecisive neurons. We apply LSOM to various input data set and confirm that LSOM can obtain a more effective map reflecting the distribution state of the input data than the conventional SOM.

I. INTRODUCTION

CLUSTERING is one of typical analysis techniques and is studied for many applications, such as statement, pattern recognition, data mining. In recent years, the Self-Organizing Map (SOM) [1] has attracted attention for the study on clustering [2]. SOM is an unsupervised neural network introduced by Kohonen in 1982 and is a simplified model of the self-organization process of the brain. SOM can classify input data according to similarities, which are obtained by the distance between neurons, and is applied to wide fields of data classifications. In the learning algorithm of SOM, a winner neuron and its neighboring neurons are updated. A learning rate, which denotes the updating degree of neurons, decreases with time, and the neuron nearer to the winner is significantly updated. This means that all the neurons can be updated, in so far as it is near to the input data or the winner neuron. In other words, all the neurons of the conventional SOM are good worker neurons.

Meanwhile, it is believed that God does not create an unnecessary thing even if it seems to useless. There is a report that 20 percent of worker ants are "lazy" (as Fig. 1) [3]. These ants keep still or stay around their nests. However, the researchers think that the lazy ants have certain role. Because, in another experiment, the ants group, which includes the lazy ants at food collections, can collect more foods than the group which includes only the worker ants. The worker ants can collect food efficiently, however, it may be hard to find new foods just because they consider the efficiency. In addition, it is thought that chances of finding foods increase because the lazy ants maunder without working. In other words, we can obtain better results in the case that useless things exist than only excellent things.

In this study, we propose a new type of SOM algorithm, which is called Lazy SOM (LSOM) algorithm. The important feature of LSOM is that three kinds of neurons exist; *worker*

Taku Haraguchi, Haruna Matsushita and Yoshifumi Nishio are with the Department of Electrical and Engineering, University of Tokushima, Tokushima, Japan (phone: +81–88–656–7470; fax: +81–88–656–7471; email: {taku, haruna, nishio}@ee.tokushima-u.ac.jp).
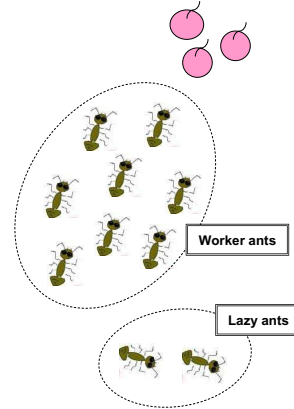
Fig. 1. World of worker ants. Some worker ants are "lazy".

neurons, *lazy neurons*, which do not work and *indecisive neurons* which are the neighborhoods of the lazy neurons. The learning rate of the lazy neurons is smaller than the ones of the worker neurons. The learning rate of the indecisive neurons become small due to the lazy neurons. We can say that LSOM carries out learning reflecting the world of worker ants.

We explain the learning algorithm of LSOM in detail in Section III. In Section IV, we apply LSOM to 2-dimensional and 3-dimensional input data, which have some clustering problems. Furthermore, we explain the learning behaviors of LSOM in detail. Learning performances are evaluated both visually and quantitatively using two measurements and are compared with the conventional SOM. We confirm that LSOM can obtain a more effective map reflecting the distribution state of the input data than the conventional SOM.

II. SELF-ORGANIZING MAP (SOM)

SOM has a two-layer structure of the input layer and the competitive layer. In the input layer, there are d-dimensional input vectors $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, N)$. In the competitive layer, M neurons are arranged on the 2-dimensional grid. Each neuron has a weight vectors $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, M)$ with the same dimension as the input vector. The range of the input vector is assumed to be between 0 and 1. The initial values of all the weight vectors are given between 0 and 1 at random.
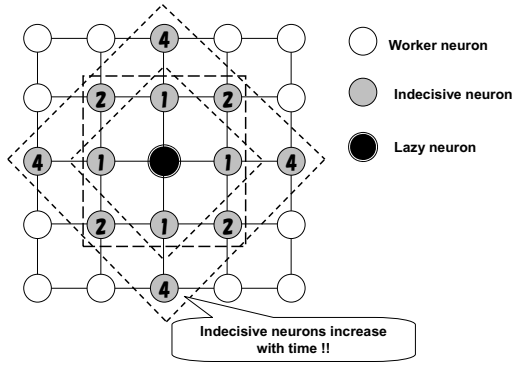
2276

Fig. 2. LSOM contains three kinds of neurons: *worker neuron*, *lazy neuron* and *indecisive neuron*. Each number denotes neighborhood distance between lazy neuron and each neuron.

(SOM1) We input an input vector $\boldsymbol{x}_j$ to all the neurons at same time in parallel.

(SOM2) We find a winner neuron by calculating the distances between the input vector $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_i$ of neuron i. The winner neuron c is the neuron with the weight vector nearest to the input vector $\boldsymbol{x}_j$;

$$c = \arg\min_i\{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}, \qquad (1)$$

where $\|\cdot\|$ is the distance measure, in this study, we use Euclidean distance.

(SOM3) The weight vector of all the neurons are updated as

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + h_{c,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)), \qquad (2)$$

where t is the learning step. $h_{c,i}(t)$ is called the neighborhood function and is described as

$$h_{c,i}(t) = \alpha(t)\exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_c\|^2}{2\sigma^2(t)}\right), \qquad (3)$$

where $\boldsymbol{r}_i$ and $\boldsymbol{r}_c$ are the vectorial locations on the display grid, $\alpha(t)$ is called the learning rate, and $\sigma(t)$ corresponds to the widths of the neighborhood function. Both $\alpha(t)$ and $\sigma(t)$ decrease monotonically with time, in this study, we use the following;

$$\alpha(t) = \alpha(0)\left(1 - \frac{t}{T}\right), \quad \sigma(t) = \sigma(0)\left(1 - \frac{t}{T}\right), \qquad (4)$$

where T is the maximum number of the learning.

(SOM4) The steps from (SOM1) to (SOM3) are repeated for all the input data.

III. LAZY SELF-ORGANIZING MAP (LSOM)

In this study, we propose Lazy SOM (LSOM). The important feature of LSOM is that three kinds of neurons exist (as Fig. 2): *worker neurons*, *lazy neurons*, which do not work, and *indecisive neurons* which are neighborhoods of the lazy neuron. The updating degree of the lazy neurons is
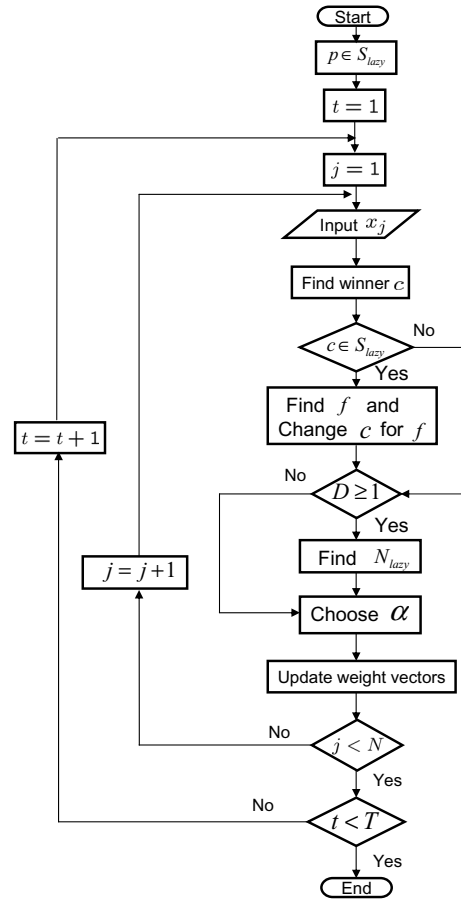

Fig. 3. Flowchart of LSOM learning.

smaller than the ones of the worker neurons. Furthermore, the updating degrees of the lazy neurons' neighbors become small due to the lazy neuron, hence the name is "indecisive". The lazy neuron becomes the worker neuron whenever it becomes the winner, and another neuron becomes lazy. The number of the indecisive neuron increases with time.

A. Learning Algorithm

We explain the learning algorithm of LSOM in detail. A flowchart of the learning algorithm is shown in Fig. 3. In LSOM, M neurons are arranged as a regular 2-dimensional grid. p neurons are classified into a set of the lazy neurons S_{lazy} at random.

(LSOM1) An input data $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.

(LSOM2) We find the winner neuron c according to Eq. (1). If $c \in S_{\text{lazy}}$, we perform (LSOM3). If not, we perform (LSOM4).

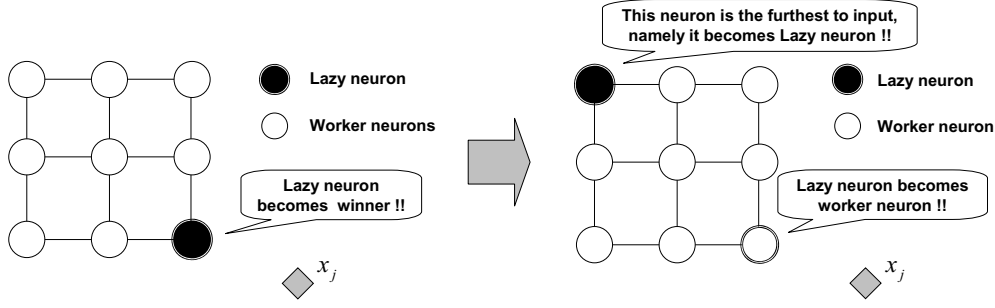**(LSOM3)** The lazy neuron, which is the winner c, is ex-

Fig. 4. Lazy neuron becomes worker neuron whenever it becomes winner, and another neuron becomes lazy.

cluded from the set of the lazy neuron S_{lazy}, and a neuron f, which is farthest from the input data $\boldsymbol{x}_j$ and is not in S_{lazy}, is selected to become a member of S_{lazy};

$$f = \arg\max_i\{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}, \quad i \notin S_{\text{lazy}}. \qquad (5)$$

In other words, the lazy neuron becomes worker if it becomes the winner c, and another neuron f becomes lazy shown as Fig. 4.

(LSOM4) We find the indecisive neurons. A set of the indecisive neurons N_{lazy} are the neighborhoods of each lazy neuron l in S_{lazy}.

$$N_{\text{lazy}} = \{i \mid \quad \|\boldsymbol{r}_i - \boldsymbol{r}_l\|^2 \le D(t),$$
$$i \ne c, \ i \notin S_{\text{lazy}}, \ l \in S_{\text{lazy}}\}, \qquad (6)$$

where $\|\boldsymbol{r}_i - \boldsymbol{r}_l\|$ is the neighborhood distance between map nodes i and l on the map grid, and $D(t)$ corresponds to the neighborhood size, namely the quantity of the indecisive neurons. $D(t)$ increases with time according to the following equation;

$$D(t) = \left[D_{\max}\frac{t}{T}\right], \qquad (7)$$

where $[\ ]$ denotes the Gauss' notation and $D_{\max}$ is a fixed parameter deciding the max value of $D(t)$.

(LSOM5) The weight vectors of all the neurons are updated as

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + h_{Lc,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)), \qquad (8)$$

where the function $h_{Lc,i}(t)$ is the neighborhood function of LSOM and is described as

$$h_{Lc,i}(t) = \alpha \exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_c\|^2}{2\sigma^2(t)}\right), \qquad (9)$$

where α is the learning rate of LSOM and is decided by each neuron's character;

$$\alpha = \begin{cases} \alpha_{\text{lazy}}, & \text{if } i = l, \ l \in S_{\text{lazy}} \\ \alpha_{\text{N}}, & \text{if } i \in N_{\text{lazy}} \\ \alpha_{\text{w}}, & \text{otherwise}, \end{cases} \qquad (10)$$

where α_{w} is the learning rate of the worker neurons, α_{lazy} is the learning rate of the lazy neurons, and α_{N} is the learning

rate of the indecisive neurons, namely $\alpha_{\text{lazy}} \le \alpha_{\text{N}} \le \alpha_{\text{w}}$.
(LSOM6) The steps from (LSOM1) to (LSOM5) are repeated for all the input data.

IV. EXPERIMENTAL RESULTS

A. For Target data

We consider a 2-dimensional input data: Target data set shown in Fig. 5(a). This data contains 770 points which has a clustering problem of outliers [4]. Both the conventional SOM and the proposed LSOM have 225 neurons (15×15) each. p=20 neurons, namely 20% of the neurons, are classified into a set of the lazy neurons. We repeat the learning 20 times for all input data, namely $T = 15400$. The parameters for the learning are chosen as follows;

(For SOM)
$$\alpha(0) = 0.3, \ \sigma(0) = 5.0,$$

(For LSOM)
$$\alpha_{\text{w}} = 0.3, \ \alpha_{\text{lazy}} = 0.03, \ \alpha_{\text{N}} = 0.15$$
$$\sigma(0) = 5.0, \ D_{\max} = 5.$$

where we use the same learning rate: $\alpha(0)$ and α_{w}, and $\sigma(0)$ for the comparison and the confirmation of the lazy neuron effect.
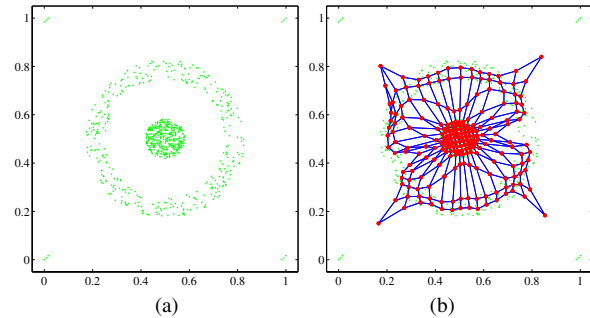

Fig. 5. Learning simulation for Target data. (a) Input data. (b) Learning result of the conventional SOM.
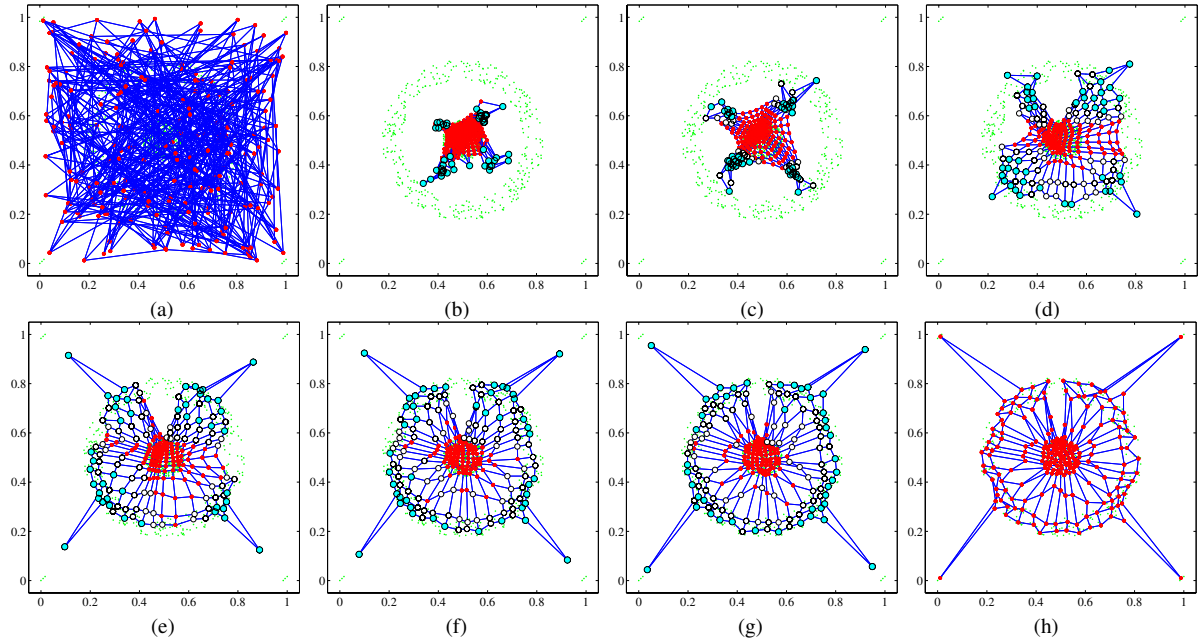
Fig. 6. Learning process of proposed LSOM. Red points, blue points and white points denote worker neurons, lazy neurons and indecisive neurons, respectively. (a) Initial state ($t = 0$). (b) $t = 500$. (c) $t = 4000$. (d) $t = 9000$. (e) $t = 11000$. (f) $t = 12000$. (g) $t = 13000$. (h) Learning results ($t = 15400$).

The simulation results of the conventional SOM is shown in Fig. 5(b). We can see that the conventional SOM does not self-organize up to all the outliers input data. Figures 6(a)-(h) show the learning process of LSOM, and the final result is shown in Fig. 6(h). We can see that LSOM can self-organize up to all the corner data. This is because LSOM has three kinds of neurons reflecting the world of worker ants.

1) Behavior of LSOM: Let us consider the learning process and behaviors of LSOM in more detail. In the early stage of learning as in Fig. 6(b), many worker neurons tend to gather at concentrated input data, namely the cluster area. Meanwhile, the lazy neurons tend to move to the outside of cluster because they are not significantly updated. However, as learning progresses, the lazy neurons self-organize other cluster as in Fig. 6(c) because they are in different area from the worker neurons. That is to say the lazy neurons can discover new input data since they maunder. In the middle stage of learning as Fig. 6(d), the indecisive neurons, whose learning rate are larger than the lazy neurons but smaller than the worker neurons, increase. The lazy neurons and the indecisive neurons self-organize outside the cluster. As more learning progresses, the lazy neurons self-organize the outliers as in Fig. 6(e). In the last stage of learning as in Figs. 6(f)-(h), the number of indecisive neurons increase further, and the map converges. From these figures, we can say that LSOM can self-organize in every corner of input data than the conventional SOM.

2) Comparison between SOM and LSOM: Furthermore, in order to compare the learning performance of LSOM with the conventional SOM numerically, we use the following well-used two measurements.

Quantization Error Q_e [1]: This measures the average distance between each input vector and its winner;

$$Qe = \frac{1}{N} \sum_{j=1}^{N} \|\boldsymbol{x}_j - \bar{\boldsymbol{w}}_j\|, \qquad (11)$$

where $\bar{\boldsymbol{w}}_j$ is the weight vector of the corresponding winner of the input vector $\boldsymbol{x}_j$. Therefore, the small value Qe is more desirable.

Neuron Utilization U [5]: This measures the percentage of neurons that are the winner of one or more input vector in the map;

$$U = \frac{1}{nm} \sum_{i=1}^{nm} u_i, \qquad (12)$$

where $u_i = 1$ if the neuron i is the winner of one or more input data. Otherwise $u_i = 0$. Thus, U nearer 1.0 is more desirable.

The calculated two measures are shown in Table I. The quantization error Qe of LSOM is smaller than the conventional SOM, and by using LSOM, the quantization error Qe has improved by 45.3545% compared to that of SOM. In addition, the neuron utilization U of LSOM is larger than the conventional SOM, and by using LSOM, the neuron utilization U has improved by 11.6959%. This result means that the result of LSOM have fewer inactive neurons than the conventional SOM, and LSOM can obtain more exact map reflecting the distribution state of input data.
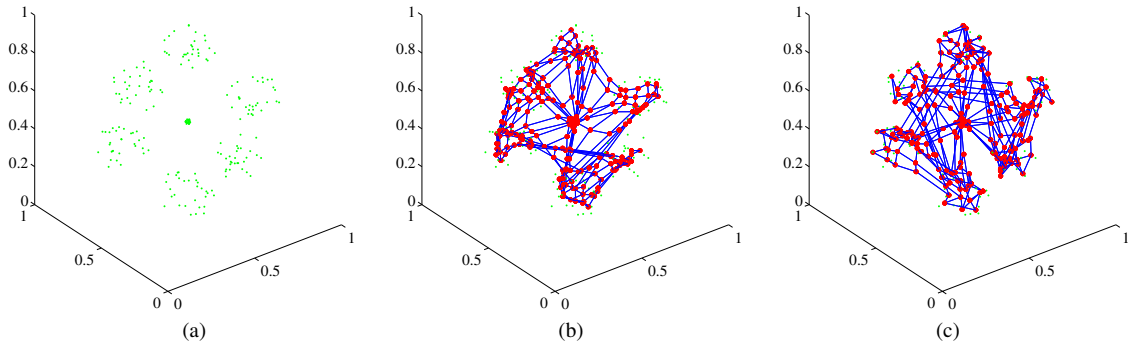
Fig. 7. Learning simulation for Hepta data. (a) Input Data. (b) Learning result of conventional SOM. (c) Learning result of LSOM.
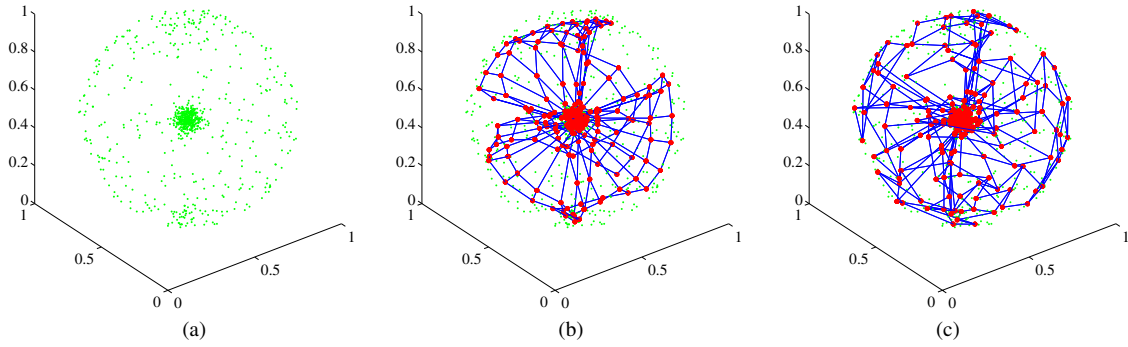

Fig. 8. Learning simulation for Atom data. (a) Input Data. (b) Learning simulation of conventional SOM. (c) Learning simulation of LSOM.

TABLE I

QUANTIZATION ERROR Qe AND NEURON UTILIZATION U FOR TARGET DATA.

	Qe	U
Conventional SOM	0.0147	0.7600
LSOM	0.0080	0.8489
Improvement rate [%]	45.3545	11.6959

TABLE II

QUANTIZATION ERROR Qe AND NEURON UTILIZATION U FOR HEPTA DATA.

	Qe	U
Conventional SOM	0.0279	0.5333
LSOM	0.0138	0.5600
Improvement rate [%]	50.4575	5.0

B. For Hepta data

Next, we considered a 3-dimensional input data called Hepta data set shown in Fig. 7(a), which has a clustering problem of different densities in clusters. The total number of the input data N is 212, and the input data has seven clusters.

We repeated the learning 70 times for all the input data, namely $T = 14840$. The learning conditions are the same as those used in Subsection IV-A.

Figures 7(b) and (c) show the learning results of the conventional SOM and LSOM, respectively. We can see that LSOM can self-organize up to edge data than the conventional SOM.

The calculated two measures are shown in Table II. The quantization error Qe of LSOM is smaller and the neuron utilization U of LSOM is larger than the conventional SOM. Because the inactive neurons have been reduced by 5.0%, more neurons are attracted to clusters and the quantization error Qe has been decreased by 50.4575%. It means that LSOM can extract the feature of the input data more effectively than the conventional SOM.

C. For Atom data

Furthermore, we considered a 3-dimensional input data called Atom data set shown in Fig. 8(a). This data set has clustering problems of linear not separable, different densities

	Qe	U
Conventional SOM	0.0493	0.7467
LSOM	0.0314	0.8267
Improvement rate [%]	36.2946	10.7143

and variances. The total number of the input data N is 800, and the input data has two clusters.

We repeated the learning 20 times for all input data, namely $T = 16000$. The learning conditions are the same as those used in Subsection IV-A.

The learning results of the conventional SOM and LSOM are shown in Figs. 8(b) and (c). We can see that LSOM can self-organize edge data more effective than the conventional SOM.

Table. III shows the calculated two measures. The quantization error Qe of LSOM is smaller than the conventional SOM, and the quantization error Qe has improved 36.2946% from using the conventional SOM. In addition, the neuron utilization U of LSOM is larger than the conventional SOM, and by using LSOM, the neuron utilization U has improved by 10.7143%. From these results, we can say that LSOM, which includes the lazy neurons, can self-organize more effectively than the conventional SOM which contains only worker neurons.

V. CONCLUSIONS

In this study, we have proposed the Lazy Self-Organizing Map (LSOM), which reflects the world of worker ants. LSOM contains three kinds of neurons: worker neurons, lazy neurons and indecisive neurons which are neighborhoods of the lazy neurons. We have investigated the behaviors of LSOM by applying it to some input data set. We have confirmed that LSOM can self-organize in every corner of the input data and can obtain more exact map reflecting the distribution state of input data than the conventional SOM. From these results, we can say that LSOM, which includes the lazy neurons, can self-organize more effectively than the conventional SOM which contains only worker neurons.

REFERENCES

[1] T. Kohonen, *Self-Organizing Maps*, Berlin, Springer, vol. 30, 1995.
[2] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Trans. Neural Networks*, vol. 11, no. 3, pp. 586–600, 2002.
[3] H. Hasegawa, "Optimization of GROUP Behavior," Japan Ethological Society Newsletter, no. 43, pp. 22–23, 2004.
[4] A. Ultsch, "Clustering with SOM: U*C", Proc. Workshop on Self-Organizing Maps, pp.75-82, 2005.
[5] Y. Cheung and L. Law, "Rival-Model Penalized Self-Organizing Map," *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 289–295, 2007.