2007 International Symposium on Nonlinear Theory and its Applications
NOLTA'07, Vancouver, Canada, September 16-19, 2007

Gray Scale Display of Input Data Using Shooting SOM

Masato Tomita, Haruna Matsushita and Yoshifumi Nishio

Department of Electrical and Electronic Engineering, Tokushima University
2–1 Minami-Josanjima Tokushima 770-8506, JAPAN
Email: {tomita, haruna, nishio}@ee.tokushima-u.ac.jp

Abstract—The Self-Organizing Map (SOM) is popular algorithm for unsupervised learning and is widely applied for many applications. In the previous study, we have proposed a new type of SOM algorithm, which is called Shooting SOM (SSOM) algorithm. The important feature of SSOM is that the neurons move like aiming at a target, namely, only some neurons near the cluster move toward the cluster to hit the area where input data are concentrated and 1-neighborhood neurons of the winner neuron get away a fraction of an inch from the cluster. Because of this feature, SSOM tends to self-organize each cluster along the figure of each cluster. We investigate the behavior of SSOM and apply SSOM to data visualization problems.

1. Introduction

In data mining, data visualization is one of the most important issues and is very useful for many applications, such as engineering applications covering. For instance, areas like pattern recognition, image analysis, process monitoring and control, and fault diagnosis. Then, the Self-Organizing Map (SOM) attracts attentions for the visualization methods in recent years. SOM is popular tools for visualization and clustering of high-dimensional data [1]. SOM is unsupervised neural network introduced by Kohonen in 1982 [2] and is a model simplifying self-organization process of the brain. SOM maps multidimensional data onto a 2-dimensional grid. SOM can classify input data according to similarities and patterns which are obtained by the distance between neurons and is applied to wide fields of data classifications. Although many methods to visualize the relationship of input data by using SOM have been proposed [3]-[7], it seems to be very difficult to construct a simple method using SOM for universal input data. On the one hand, in the world, the amount and complexity of data increase from year to year and control of large-volume data attracts a lot of attention. Therefore, it is important to investigate various visualization methods of input data.

In the previous study, we have proposed a new type SOM algorithm, which is called Shooting SOM (SSOM) algorithm [8]. The important feature of SSOM is that the neurons move like aiming at a target, namely, only some neurons near the cluster move toward the cluster to hit the area where input data are concentrated, 1-neighborhood neurons of the winner neuron move slightly from the cluster, and other neurons do not move at all from the initial position. Because of this feature, SSOM tends to find near the center of the area where input data are concentrated.

In this study, by using SSOM, we propose a gray scale display method of input data. The proposed display method is that the shading is reflected by the number of becoming the winner. By learning the input data, we think that the proposed method can simplify and visualize the information of the input data that it is difficult to judge by appearance. We investigate the behavior of SSOM and apply SSOM to the proposed display method. The efficiencies of SSOM are confirmed by several simulation results.

In the Section II, we explain the learning algorithm of SSOM algorithm. In the Section III, the behavior of SSOM is explained with some simulation results. In the Section IV, we apply SSOM to the proposed display method.

2. Shooting SOM (SSOM)

In the previous research, we have proposed SSOM. In this section we introduce SSOM algorithm in order to make clear new points of the proposed application method.

We explain the learning algorithm of SSOM. The learning algorithm of SSOM is largely similar to the conventional SOM. In SSOM, m neurons are arranged as a regular 2-dimensional grid. The range of the elements of d-dimensional input data $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ ($j = 1, 2, \cdots, N$) are assumed.

(SSOM1) The initial values of all the weight vectors $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ ($i = 1, 2, \cdots, m$) of the neurons are given between 0 and 1 in a reticular pattern.
(SSOM2) An input data $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.
(SSOM3) We find the winner neuron c by calculating the distances between $\boldsymbol{x}_j$ and $\boldsymbol{w}_i$ according to;

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

In other words, the winner neuron c is the neuron with the weight vector nearest to the input vector $\boldsymbol{x}_j$. In this study, Euclidean distance is used for Eq. (1).
(SSOM4) We measure whether the winner neuron or 1-neighborhood of the winner neuron or otherwise and show in Fig. 1.

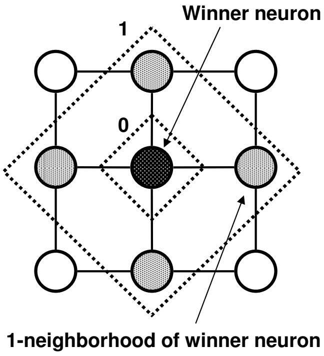

Figure 1: Winner neuron and 1-neighborhood neurons of the winner neuron.

Furthermore, we determine the update rate A_i for each cases as follows;

$$A_i = \begin{cases} 1, & \text{if } i = c \\ -a, & \text{if } i \text{ is 1-neighborhood of } c \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

where a is the constant number and usually set the smaller value. Namely, if the neuron i is the winner neuron, the neuron i is updated like the winner neuron of the conventional SOM. If the neuron i is 1-neighborhood neuron, the neuron i get away a fraction of an inch from the input data because a is negative number. And if the neuron i is otherwise, the neuron i is not updated from initial position.

(SSOM5) The weight vectors of all the neurons are updated as;

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + A_i \times \alpha(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)), \tag{3}$$

where t is the learning step and $\alpha(t)$ is the learning rate. $\alpha(t)$ decrease with time according to the following equations;

$$\alpha(t) = \alpha(0)\,(1 - t/T)\,, \tag{4}$$

where T is the maximum number of the learning.

(SSOM6) The steps from (SSOM2) to (SSOM5) are repeated for all the input data, namely, from $j = 1$ to $j = N$.

3. Learning Results

3.1. 2-dimensional input data

We consider the 2-dimensional input data of 1200 points whose distribution is as Fig. 2(1a). 800 points are distributed within a circular range from 0.1 to 0.5 horizontally and from 0.1 to 0.5 vertically. 400 points are distributed within a rectangular range from 0.7 to 0.9 horizontally and from 0.7 to 0.9 vertically. We consider the conventional SOM and the proposed SSOM with 400 neurons (20×20). The parameters for the learning of the conventional SOM and SSOM are chosen as follows;

$$\alpha(0) = 0.9,\ \sigma(0) = 1/16,\ \text{a}= -0.005, \tag{5}$$

We execute the learning for all input data 3 times. The simulation results of SSOM is shown in Fig. 2(1c). SSOM self-organizes only near the center of the area where the input data are concentrated.

Let us examine the behavior of SSOM in more detail. As we can see from Fig. 2(1b) and Fig. 2(1c), the conventional SOM self-organizes input data of two clusters with all neurons. However, we can see that SSOM moves like aiming at a target from learning process. Only some neurons near the cluster move toward the cluster to hit the area where input data are concentrated, 1-neighborhood neurons of the winner neuron move slightly from the cluster, and other neurons do not move at all from the initial position. Therefore, only neurons near a cluster self-organize the cluster. In addition, in the interior of the cluster, some neurons which distribute the outside of the cluster can not move into the interior of the cluster because only neurons of the interior can self-organize the input data of the cluster. Consequently, SSOM can effectively self-organize each cluster along the shape of the cluster.

3.2. 3-dimensional input data

We consider the 3-dimensional input data of 1800 points whose distribution is as Fig. 2(2a). 1200 points are distributed within a sphere with radius 0.3. 600 points are distributed within a cube, 0.3 on a side. We consider the conventional SOM and the proposed SSOM with 400 neurons (20×20). The parameters for the learning of the conventional SOM and SSOM are chosen as follows;

$$\alpha(0) = 0.9,\ \sigma(0) = 1/16,\ \text{a}= -0.003, \tag{6}$$

We execute the learning for all input data 3 times. As we can see from Fig. 2(2b) and Fig. 4(2c), SSOM can effectively self-organize each cluster, just like in the 2-dimensional input data.

4. Gray Scale Display

In this study, we apply SSOM to the proposed display method for the simulation results. And we apply the conventional SOM to the conventional display method.

4.1. Display method

We propose the gray scale display method as follows. The proposed method is that the shading is reflected by the number of becoming the winner as shown in Fig. 3. For example, because the neuron 5 becomes a winner 5 times, the shading of the circle color is the deepest. In contrast, because neurons 7 and 8 have never been a winner, the shading of these circles is the lightest. Namely, the more number of becoming the winner is deeper color. Therefore, we can visually see the positional relationship of input data and rough density.

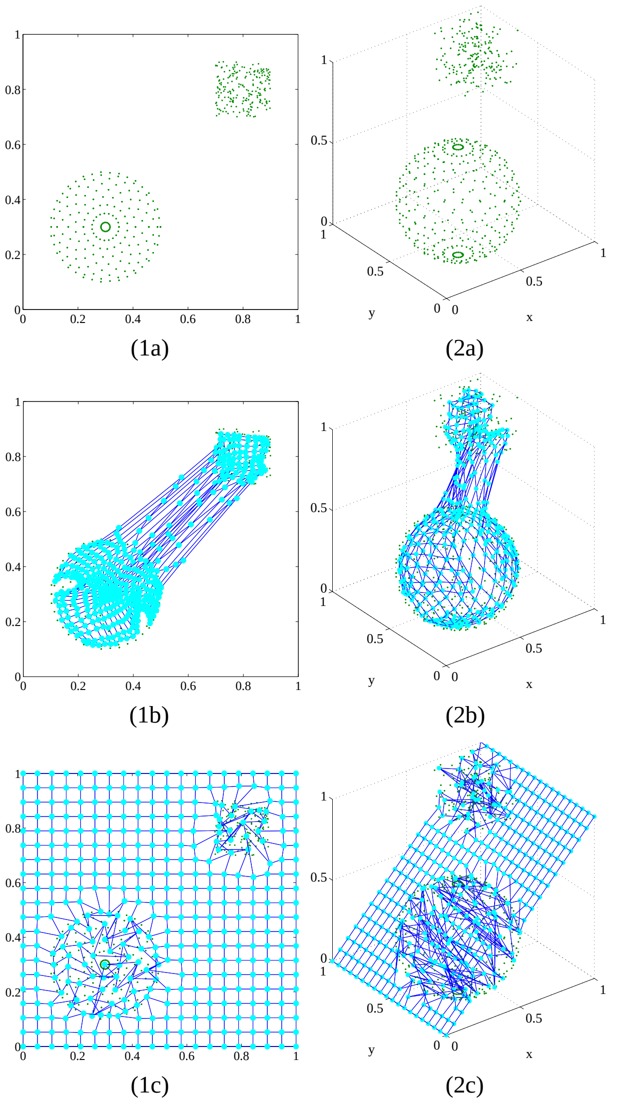

Figure 2: (1) 2-dimensional input data. (2) 3-dimensional input data. (a) Input data. (b) Learning result of the conventional SOM. (c) Learning result of SSOM.

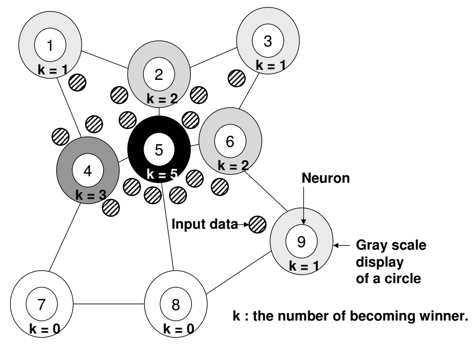

Figure 3: Gray scale display method

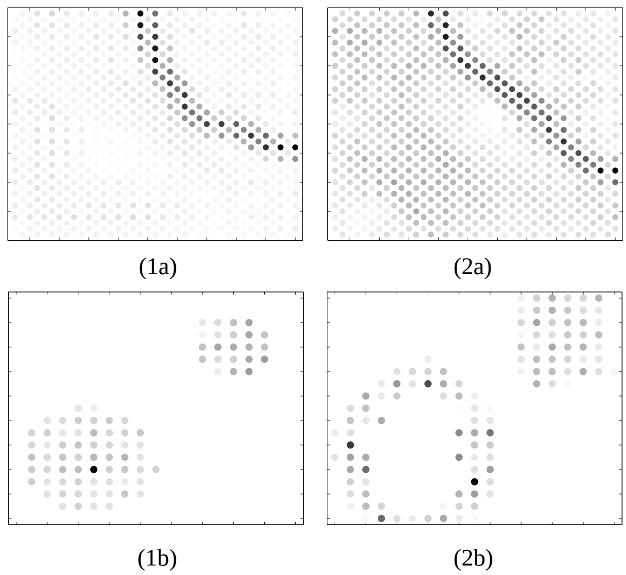

Figure 4: (1) 2-dimensional input data. (2) 3-dimensional input data. (a) Learning result of the conventional SOM. (b) Learning result of SSOM.

4.2. Example 1 (Result in Fig. 2)

We apply the gray scale display method to the simulation results of Fig. 2. The results are shown in Fig. 4. As we can see from Fig. 4(1a) and Fig. 4(2a), the conventional display method using the conventional SOM can only visualize the boundary line between clusters as usual. However, we can see that the proposed gray scale method using SSOM visualizes the positional relationship of input data as well as the number of clusters from Fig. 4(1b) and Fig. 4(2b). Therefore, not only visualizing the shape of each cluster, the proposed method can visualize the rough density by learning.

4.3. Example 2 (Lorenz model)

In this subsection, we apply the proposed gray scale display to the Lorenz chaotic attractor. We prepare the 2-dimensional and the 3-dimensional input data of the Lorenz attractor as shown in Fig. 5(1a) and Fig. 5(2a). Fig. 5(b) to Fig. 5(c) show the learning results of the conventional SOM and SSOM, respectively. By using these learning results and the proposed gray scale display, we obtained the results of the gray scale visualization as shown in Fig. 6. From the results, we can not visualize the shape and the relationship for the conventional SOM. However, for the proposed display method and SSOM, we can see the relationship of input data and the shape of each cluster.

5. Conclusions

In this study, we have discovered the new property of the Shooting SOM (SSOM) proposed in the previous study and proposed the gray scale display method. We have explained the differences between the conventional SOM and SSOM

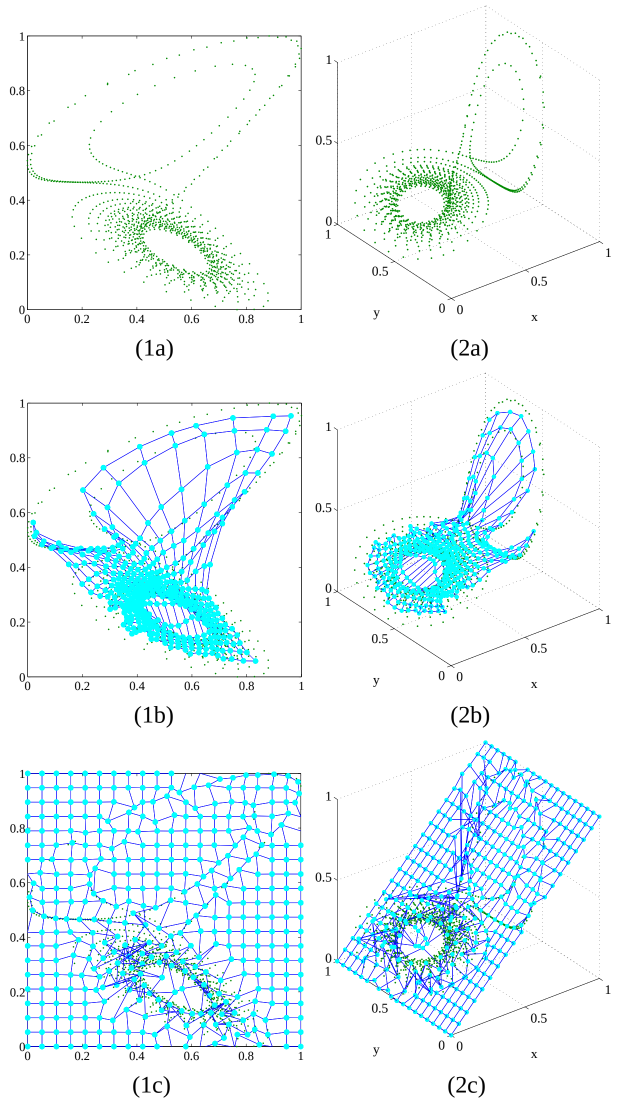

Figure 5: (1) 2-dimensional. (2) 3-dimensional. (a) Input data. (b) Learning result of the conventional SOM. (c) Learning result of SSOM.

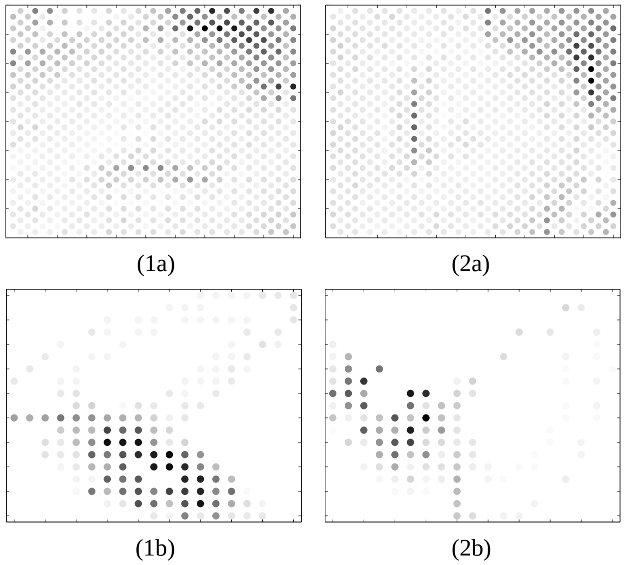

Figure 6: (1) 2-dimensional Lorenz model. (2) 3-dimensional lorenz model. (a) Simulation result of the conventional SOM. (b) Simulation result of SSOM.

have investigated its behavior. Furthermore, we have applied SSOM to visualize the relationship of input data and have confirmed its efficiency by combining the best properties of SSOM and the proposed display method.

In the future, we try to extend to higher dimensional input data because we think that the proposed method can show maximize the benefit by learning more than 3-dimensional input data. We try to discover new applications of SSOM in diverse fields such as sound data processing.

Acknowledgments

This work was partly supported by Yamaha Music Foundation.

References

[1] T. Kohonen, “The Self-Organizing Maps,” *Neurocomputing*, vol. 21, pp. 1-6, 1998.

[2] T. Kohonen, *Self-Organizing Maps*, Berlin, Springer, vol. 30, 1995.

[3] J. Vesanto and E. Alhoniemi, “Clustering of the Self-Organizing Map,” *Neural Networks*, vol. 11, pp. 586-600, 2000.

[4] D. Alahakoon, S. Halgamuge and B. Srinivasan, “Dynamic Self-Organizing Maps with Controlled Growth for Knowledge Discovery,” *Neural Networks*, vol. 11, pp. 601-614, 2000.

[5] J. Laaksonen, M. Koskela and E. Oja, “PicSOM-Self-Organizing Image Retrieval With MPEG-7 Content Descriptors,” *Neural Networks*, vol. 13, pp. 841-853, 2002.

[6] K. Simunic and J. Novak, “Combining Visualization and Interactive Clusterung for Exploring Large Document Pools,” *Proc. of VIIP’04*, 2004.

[7] C. Hsu, “Generalizing Self-Organizing Map for Categorical Data,” *Neural Networks*, vol. 17, pp. 294-304, 2006.

[8] M. Tomita, H. Matsushita and Y. Nishio, “Shooting SOM and its Application for Clustering,” *Proc. of NOLTA’06*, pp. 199-202, 2006.