2006 RISP International Workshop
on Nonlinear Circuits and Signal Processing (NCSP'06)
Waikiki Beach Marriott, Honolulu,
Hawaii, USA, March 3-5, 2006

Data Extraction by Self-Organizing Map Containing Neurons with Additional States

Masato Tomita†, Haruna Matsushita† and Yoshifumi Nishio†

†Tokushima University
2-1 Minami-Josanjima, Tokushima, Japan
Phone:+81-88-656-7470, FAX:+81-88-656-7471
Email: {tomita, haruna, nishio}@ee.tokushima-u.ac.jp

Abstract

The Self-Organizing Map (SOM) is unsupervised neural network introduced by Teuvo Kohonen. SOM attracts attentions for clustering in these years. In this study, we propose SOM containing neurons with additional states. Basic algorithm of proposed SOM is the same as conventional SOM. Each neuron of the proposed SOM has an additional state value. The value decreases according to the update of the weight vectors. Furthermore, as decreasing the value, the updated rate of neurons reduces. The proposed SOM self-organizes only cluster from the input data which include a lot of noises. The behavior of proposed SOM is investigated with an application to clustering input data including a lot of noises. We can see that the proposed SOM successfully extracts the cluster.

1. Introduction

In data mining, clustering is one of the most important issues and is also very useful for many applications, such as industrial engineering, image processing, biology and medicine. Then, the Self-Organizing Map (SOM) attracts attentions for clustering in recent years. SOM is unsupervised neural network introduced by Kohonen in 1982 [1] and is a model simplifying self-organization process of the brain. SOM maps multidimensional data onto a 2-dimensional grid. SOM can classify the 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 in order to extract clusters by using SOM have been proposed [2]-[8], 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. Therefore, it is important to investigate various extraction methods of clusters from data including a lot of noises.

In this study, we propose SOM containing neurons with additional states. The proposed SOM self-organizes only cluster from the input data including a lot of noises. The most important feature of the proposed SOM is that each neuron has an additional state value. The value decreases according to the update of the weight vectors. Furthermore, as decreasing the value, the updated rate of neurons reduces. From the feature, proposed SOM tends to self-organize the area where the input data are concentrated.

In the Section 2, the algorithm of the conventional SOM is introduced. In the Section 3, we explain the learning algorithm of the proposed SOM algorithm in detail. In the Section 4, we investigate the behavior of the proposed SOM with an application to clustering input data including a lot of noises. We can see that the proposed SOM successfully extracts the cluster.

2. Self-Organizing Map (SOM)

In this Section, we explain about the algorithm of the conventional SOM.

SOM has two-layer structure of the input layer and the competitive layer. In the competitive layer, 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 to be from 0 to 1.

(SOM1) The initial values of all the weight vectors $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, m)$ are given between 0 and 1 at random.

(SOM2) An input vector $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.

(SOM3) We find the winner neuron $c(j)$ by calculating the distance between the input vector $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_i$ of neuron i, according to;

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

In other words, the winner neuron $c(j)$ 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).

(SOM4) The weight vectors of the neurons are updated as;

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

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

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

where $\alpha(t)$ is the learning rate, $\boldsymbol{r}_i$ and $\boldsymbol{r}_{c(j)}$ are the vectorial locations on the display grid, and $\sigma(t)$ corresponds to the width of the neighborhood function. $\alpha(t)$ and $\sigma(t)$ decrease with time according to the following equations;

$$\begin{aligned} \alpha(t) &= \alpha(0)\,(1 - t/T)\,, \\ \sigma(t) &= \sigma(0)\,(1 - t/T)\,. \end{aligned} \tag{4}$$

where T is the maximum number of the learning.
(SOM5) The steps from (SOM2) to (SOM4) are repeated for all the input data.

3. SOM Containing Neurons with Additional States

In this study, we propose SOM containing neurons with additional states.

We assign a variable p_i ($i = 1, 2, \cdots, m$) to each neuron i of SOM. This additional state p_i is updated according to the following rule and also influences the update of the weight vectors.

$$p_i(t+1) = p_i(t) - 0.5 h_{c(j),i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)). \tag{5}$$

We set the initial value of p_i as 3.0.
(Case 1) For $2 < p_i \leq 3$:
The weight vector of neuron i is updated according to Eq. (2).
(Case 2) For $1 < p_i \leq 2$:
The weight vector of neuron i is updated according to;

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + 0.5 h_{c(j),i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)). \tag{6}$$

(Case 3) For $0 < p_i \leq 1$:
The weight vector of neuron i is updated according to;

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + 0.1 h_{c(j),i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)). \tag{7}$$

(Case 4) For $p_i \leq 0$:
p_i does not decrease anymore. The weight vector of neuron i is updated according to;

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + 0.1 h_{pc(j),i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)). \tag{8}$$

where $h_{pc(j),i}(t)$ is the neighborhood function described as follows;

$$h_{pc(j),i}(t) = \alpha_p(t) \exp\left(-\frac{\|\boldsymbol{r}_i - \boldsymbol{r}_{c(j)}\|^2}{2\sigma_p^2(t)}\right). \tag{9}$$

$$\begin{aligned} \alpha_p(t) &= \alpha_p(0)\,(1 - t/T)\,, \\ \sigma_p(t) &= \sigma_p(0)\,(1 - t/T)\,. \end{aligned} \tag{10}$$

The initial value of α_p and σ_p are chosen as follows;

$$\alpha_p(0) = 0.05,\ \sigma_p(0) = 10.0.$$

By virtue of this additional state, SOM is updated as follows. In the initial stage of the program, neuron i converges at once in the area of cluster. Furthermore, in the last stage of the program, neuron i does not move well to spread out of the cluster. Consequently we can gather neurons in the area of the cluster.

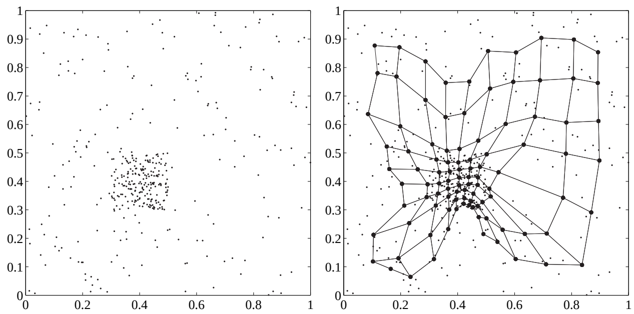
4. Simulation Results

4.1. 2-dimensional input data

Input data is 2-dimensional random data of 400 points whose distribution is non-uniform as Fig. 1(a). 200 points are distributed within a small range from 0.3 to 0.5 horizontally and from 0.3 to 0.5 vertically. The remaining 200 points are uniformly distributed between 0 and 1 at random. The conventional SOM and the proposed SOM have 100 neurons (10×10). The parameters for the learning are chosen as follows;

$$\alpha(0) = 0.7,\ \sigma(0) = 3.0.$$

We repeat the learning 5 times for all input data. The simulation result of the conventional SOM is shown in Fig. 1(b). Fig. 2 shows the learning process of the proposed SOM in detail. We can see that the conventional SOM self-organizes all the input data including a lot of noises. On the other hand, the proposed SOM self-organizes only the cluster and is not influenced by noises.



(a) (b)

Figure 1: Clustering of 2-dimensional input data. (a) Input data. (b) Learning results of conventional SOM.

The concept using the proposed SOM can be exploited to extract the data only in clusters of the input data including a lot of noises, because proposed SOM can find such areas by themselves. We carry out the extraction of cluster after Fig. 1(b) and Fig. 2(f). The extraction method is a relatively simple as follows. After learning, the input data which is

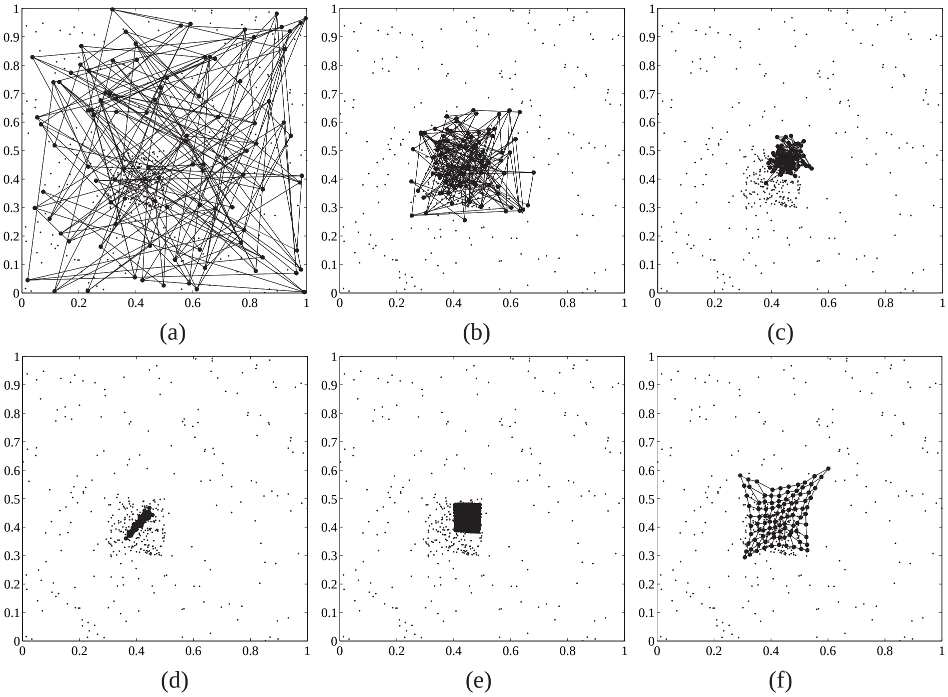


Figure 2: Learning process of proposed SOM. (a) Initial state (t = 0). (b) t = 20. (c) t = 50. (d) t = 100. (e) t = 200. (f) Learning results of proposed SOM (t = 400).

within a radius of R from any neurons on the map are classified into the cluster.

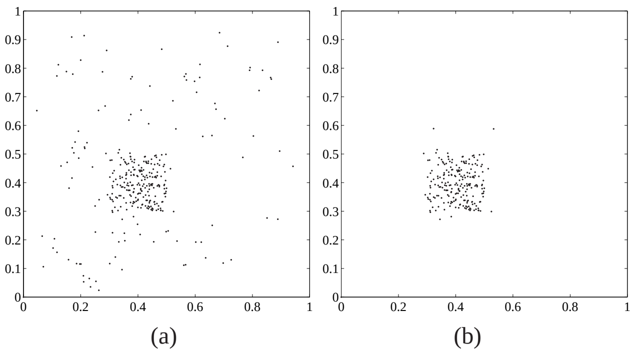


Figure 3: Extraction of cluster. (a) Cluster extracted by conventional SOM. (b) Cluster extracted by proposed SOM.

The simulation results of the conventional SOM and the proposed SOM are shown in Figs. 3(a) and (b), respectively (R = 0.05). As we can see from figures, the cluster obtained by the conventional SOM includes a lot of noises. However, the proposed SOM can successfully extract only the cluster.

4.2. 3-dimensional input data

Furthermore, we carry out simulation for 3-dimensional input data shown in Fig. 4. The input data is generated artificially as follows. Total number of the input data is 400 ($N = 400$). 200 points are distributed in the cluster whose values are distributed according to x-coordinate : $N(0.2, 0.085^2)$, y-coordinate : $N(0.3, 0.09^2)$, and z-coordinate : $N(0.7, 0.09^2)$. The remaining 200 points of the input data are distributed between 0 and 1 at random.

The conventional SOM and the proposed SOM have 100 neurons (10×10). The parameters for the learning are chosen as follows;

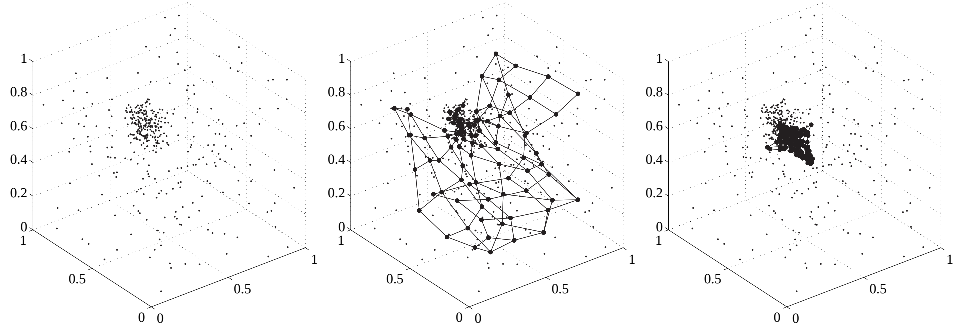
$$\alpha(0) = 0.7, \ \sigma(0) = 3.0.$$

We repeat the learning 5 times for all input data.

Figures 4(b) and (c) show the learning results of the conventional SOM and the proposed SOM, respectively (R = 0.1). We can see that SOM containing neurons with additional states self-organizes only the cluster. Figure 5(b) shows the extracted cluster by the proposed SOM method. We can confirm that the noises are removed and only cluster part can be extracted very well.

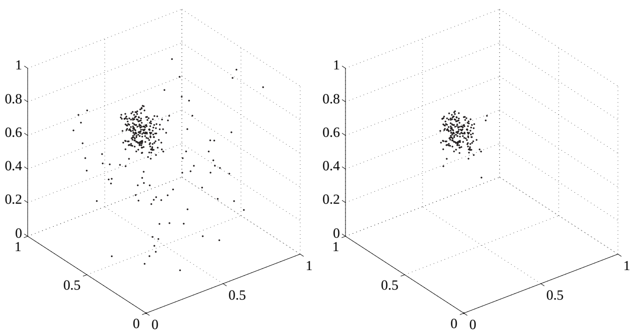
5. Conclusions

In this study, we have proposed the neuron with additional states of SOM. We have explained the differences between SOM and proposed SOM with learning algorithm and have investigated its behavior caused by the additional states. Furthermore, we have applied proposed SOM to extract clusters including a lot of noises and have confirmed the efficiency.



(a) (b) (c)

Figure 4: Clustering of 3-dimensional input data. (a) Input data. (b) Learning results of conventional SOM. (c) Learning results of proposed SOM.



(a) (b)

Figure 5: Extraction of cluster. (a) Cluster extracted by conventional SOM. (b) Cluster extracted by proposed SOM.

In the future, we try to discover a new way to more extract data than proposed SOM. And we try to utilize it in diverse fields such as sound data processing.

References

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

[2] Y. Cheng, "Clustering with Competing Self-Organizing Maps," *Proc. of IJCNN'92*, vol. Ⅳ, pp. 785-790, 1992.

[3] W. Wan and D. Fraser, "M2dSOMAP: Clustering and Classification of Remotely Sensed Imagery by Combining Multiple Kohonen Self-Organizing Maps and Associative Memory," *Proc. of IJCNN'93*, vol. Ⅲ, pp. 2464-2467, 1993.

[4] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586-600, 2000.

[5] C. Derek and E. Adrian, "Finding Curvilinear Features in Spatial Point Patterns: Principal Curve Clustering with noise." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, pp. 601-609, 2000.

[6] P. Doucette, P. Agouris and A. Stefanidis, "Self-Organized Clustering for Road Extraction in Classified Imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 55, pp. 347-358, 2001.

[7] M. Gervey and S. Lek, "Patterning and Clustering Ecological Assemblages," *Proc. of WSOM'05*, pp. 9-16, 2005.

[8] H. Matsushita and Y. Nishio, "Competing Behavior of Two Kinds of SOMs and its Application to Clustering," *Proc. of WSOM'05*, pp. 355-362, 2005.