2007 RISP International Workshop on Nonlinear Circuits and Signal Processing (NCSP'07)
Shanghai Jiao Tong University, Shanghai, China, Mar. 3-6, 2007

Disconnecting Self-Organizing Map

Haruna MATSUSHITA and Yoshifumi NISHIO

Tokushima University
2-1 Minami-Josanjima, Tokushima 770-8506, Japan
Phone:+81-88-656-7470 FAX:+81-88-656-7470
Email: {haruna, nishio}@ee.tokushima-u.ac.jp

Abstract

The Self-Organizing Map (SOM) has problems with some inactive neurons which have affected a result of clustering. In this study, we propose a new SOM algorithm which is the Disconnecting Self-Organizing Map (DSOM). In the initial state, all neurons of DSOM are directly or indirectly connected each other. However, connections between neurons located at distant locations are cut with learning. We can confirm that the result of using DSOM includes no inactive neurons and DSOM can obtain a more effective map reflecting the distribution state of input data than the conventional SOM.

1. Introduction

In recent years, the Self-Organizing Map (SOM) is widely used in classification and feature extraction tasks [1]-[5]. SOM is one of the unsupervised neural networks introduced by Kohonen in 1982 [6] and is a model simplifying self-organization process of the brain. In the learning algorithm of SOM, a winner, which is a neuron with the weight vector closest to the input vector, and its neighboring neurons are updated, regardless of the distance between the input vector and the neighboring neurons. For this reason, if we apply SOM to clustering of input data which includes some clusters located at distant locations, there are some inactive neurons between the clusters. Because there are inactive neurons on an area without the input data, we are misled into thinking that there are some input data between clusters.

In this study, we propose a new SOM algorithm which is the Disconnecting Self-Organizing Map (DSOM). In the initial state, all neurons of DSOM are directly or indirectly connected each other. However, connections between neurons located at distant locations are cut with learning. We explain contours of the learning algorithm of DSOM. Each connection between neurons of DSOM has a connection strength. The connection strength between the winner and the neuron, which is the most distant from the input data in a set of direct topological neighbors of the winner, is decreased with each learning. The neuron is disconnected from the winner if the connection strength between them becomes zero.

In Section II, we explain the learning algorithm of the conventional SOM. In Section III, the learning algorithm of DSOM in detail. The learning process and the behaviors of DSOM are investigated in Section IV with an application to clustering of 2-dimensional input data. The clustering ability is evaluated by the visualization of the result. We can confirm that the result of using DSOM includes no inactive neurons. Furthermore, in Section V, we apply DSOM to feature extraction. We can see that DSOM can obtain a more effective map reflecting the distribution state of input data than the conventional SOM.

2. Self-Organizing Map (SOM)

We explain the learning algorithm of the conventional SOM. SOM consists of m neurons located at a regular low-dimensional grid, usually a 2-D grid. The basic SOM algorithm is iterative. Each neuron i has a d-dimensional weight vector $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, m)$. The initial values of all the weight vectors are given over the input space at random. 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) An input vector $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.
(SOM2) Distances between $\boldsymbol{x}_j$ and all the weight vectors are calculated. The winner, denoted by c, is the neuron with the weight vector closest to the input vector $\boldsymbol{x}_j$;

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

where $\|\cdot\|$ is the distance measure, in this study, Euclidean distance.
(SOM3) The weight vectors of the neurons are updated as;

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

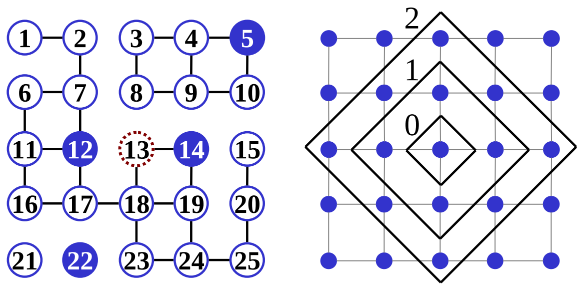
where t is the learning step. $h_{c,i}(t)$ is called the neighborhood function and is described as a Gaussian function;

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

where $\|\boldsymbol{r}_i - \boldsymbol{r}_c\|$ is the distance between map nodes c and i on the map grid, $\alpha(t)$ is the learning rate, and $\sigma(t)$ corresponds to the width of the neighborhood function. Both $\alpha(t)$ and $\sigma(t)$ decrease with time as follows;

$$\begin{aligned} \alpha(t) &= \alpha(0)\left(\frac{\alpha(T)}{\alpha(0)}\right)^{t/T}, \\ \sigma(t) &= \sigma(0)\left(\frac{\sigma(T)}{\sigma(0)}\right)^{t/T}, \end{aligned} \tag{4}$$

where T is the maximum number of the learning.
(SOM4) The steps from (SOM1) to (SOM3) are repeated for all the input data.



(a) (b)

Figure 1: Neighborhood distances of the rectangular grid. (a) Neighborhood distances $n_{c,i}$ of $c = 13$. $n_{c,14} = 1$, $n_{c,12} = 3$, $n_{c,5} = n_{c,22} = m$ (namely, 25). (b) Discrete neighborhoods (size 0, 1 and 2) of the centermost neuron when all the neurons are connected.

3. Disconnecting SOM (DSOM)

We explain a proposed new SOM algorithm, DSOM. The initial state of all neurons of DSOM are directly or indirectly connected to all neuron. However, the connection between neurons located at distant locations is cut with learning. The connection between each neuron of DSOM has a connection strength $Cs_{(c,i)}$. The initial values of all the weight vectors are given over the input space at random.

(DSOM1) An input vector $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.
(DSOM2) Distances between $\boldsymbol{x}_j$ and all the weight vectors are calculated, and the rank order of distances, denoted by $k = 0, \cdots, m-1$, is calculated. k_i is the rank of w_i, namely, $k_c = 0$ is the rank of the winner c which is closest to $\boldsymbol{x}_j$ according to Eq. (1).
(DSOM3) The weight vectors of the neurons are updated as;

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + h_{Dc,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)), \tag{5}$$

where $h_{Dc,i}(t)$ is the neighborhood function of DSOM;

$$h_{Dc,i}(t) = \alpha(t) \exp\left(-\frac{k_i + n_{c,i}}{\sigma^2(t)}\right), \tag{6}$$

where $n_{c,i}$ is the neighborhood distance between c and each neuron i. The neighborhood distances are defined as shortest-path distances between connected map nodes as Fig. 1(a). If a neuron i is not connected directory or indirectly to c, $n_{c,i}$ is equal to the number of neurons m.
(DSOM4) A connection between c and a neuron p which is closest to $\boldsymbol{x}_j$ in Nc_1 is created, if it does not exist.

$$p = \arg\min_i \{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}, \quad i \in Nc_1, \tag{7}$$

where Nc_1 is the set of direct topological neighbors of c as Fig. 1(b),

The connection strength between c and p is set to initial value (this means "refresh" the connection reference);

$$Cs_{(c,p)} = Cs_0, \tag{8}$$

where Cs_0 is the initial value of Cs.
(DSOM5) A disconnecting neuron q is found. q is the most distant from $\boldsymbol{x}_j$ in Nc_1. However, if k_q is smaller than the number of Nc_1, the disconnecting neuron q does not exist, and we perform (DSOM7).

$$q = \arg\max_i \{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}, \quad i \in Nc_1. \tag{9}$$

The connection strength between c and q is decreased;

$$Cs_{(c,q)} = Cs_{(c,q)} - \|\boldsymbol{w}_q - \boldsymbol{x}_j\|^2. \tag{10}$$

(DSOM6) The disconnecting neuron q is disconnected from the winner c if its connection strength becomes smaller than zero.
(DSOM7) The steps from (DSOM1) to (DSOM6) are repeated for all the input data.

4. Application to Clustering

We consider 2-dimensional input data as shown in Fig. 2(a) to clustering. The input data is generated artificially as follows. Total number of the input data N is 15000, and the input data include two clusters. 7500 data are distributed within a range from 0.2 to 0.8 horizontally and from 0.1 to 0.3 vertically. The remaining 7500 data are distributed within a range from 0.2 to 0.8 horizontally and from 0.7 to 0.9 vertically. All the input data are sorted at random.

Both the conventional SOM and DSOM has $m = 100$ neurons (10×10). We repeat the learning until all input data is

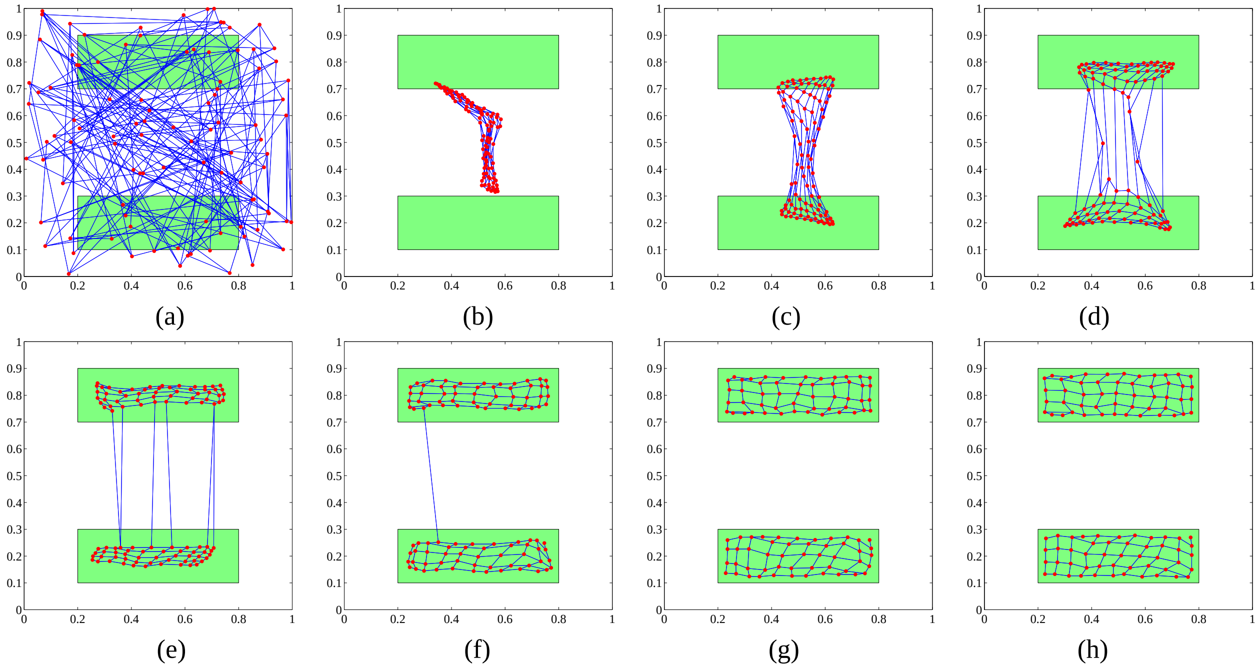


Figure 3: Learning process of DSOM. (a) Initial state ($t = 0$). (b) $t = 1000$. (c) $t = 3000$. (d) $t = 5000$. (e) $t = 7500$. (f) $t = 9500$. (g) $t = 12500$. (f) Learning result ($t = 15000$).

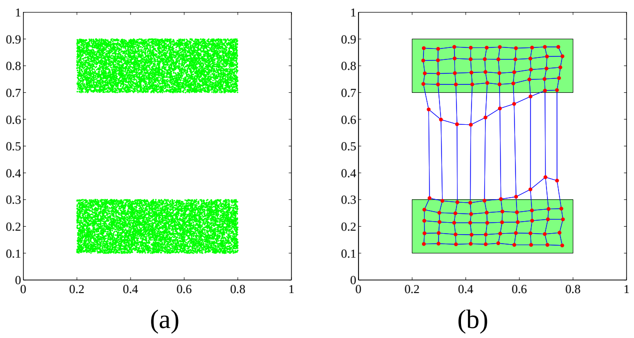


Figure 2: Clustering of 2-dimensional input data. (a) Input data. (b) Learning result of the conventional SOM.

inputted, namely $T = 15000$. The parameters of the learning are chosen as follows;

(For SOM)

$$\alpha(0) = 0.5, \alpha(T) = 0, \sigma(0) = 5, \sigma(T) = 0.$$

(For DSOM)

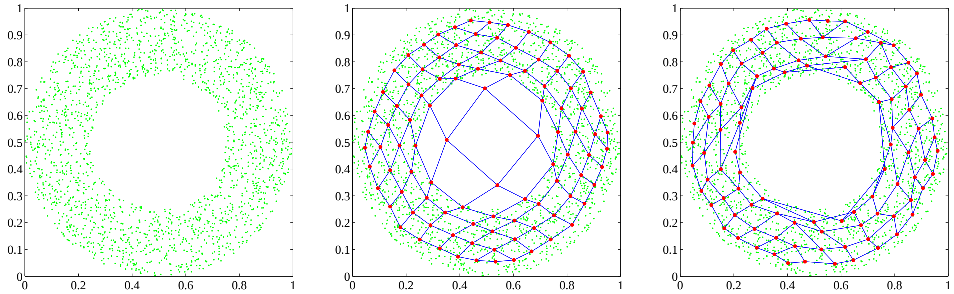
$$\alpha(0) = 0.5, \alpha(T) = 0.05, \sigma(0) = 40, \sigma(T) = 0.5, Cs_0 = 12.$$

The learning result of the conventional SOM is shown in Fig. 2(b). We can see that there are some inactive neurons between the two clusters.

The other side, the result of DSOM and its learning process are shown in Fig. 3. We can see from Fig. 3(h) that there are no inactive neurons between the two clusters.

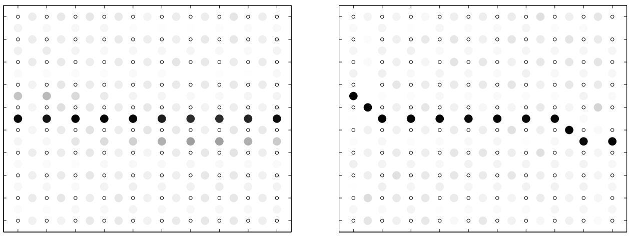
Let us consider the learning process. The initial states of weight vectors of neurons are random values, and all neurons are directly or indirectly connected each other as Fig. 3(a). In the early-stage of the learning as Figs. 3(b) and (c), all neurons are still connected mutually and all neurons are self-organizing all input data. Furthermore, from Fig. 3(d), there are some inactive neurons because the neurons located at distant locations are not disconnected. In the middle stage as Figs. 3(e) and (f), we can see that the connections between neurons of the two clusters are cut, and there are no inactive neurons. This is because the neurons of DSOM are not affected by neurons which are self-organizing another cluster, so, the neurons can learn more distant for the distant input data, than the conventional SOM. In the last stage as Figs. 3(g) and (h), we can confirm that DSOM are completely separated by two and each neuron group self-organizes each cluster.

Figure 4 shows distances between adjacent neurons of learning results of Figs. 2(b) and 3(h). This figure thus visualizes the cluster structure of the map. Black circles on this figure mean large distance between neighboring map nodes. Clusters are typically uniform areas of white circles. We can see that the boundary line of DSOM is clearer than the conventional SOM because DSOM has no inactive neurons.



(a) (b) (c)

Figure 5: Feature extraction for doughnut-shaped data. (a) Input data. (b) Learning result of conventional SOM. (c) Learning result of DSOM.



(a) (b)

Figure 4: Visualization of learning result. (a) Conventional SOM. (b) DSOM.

5. Application to Feature Extraction

Furthermore, we apply DSOM to feature extraction. We use the "doughnut"-shaped data shown as Fig. 5 for the input data. Total number of the input data N is 2000. All the input data are sorted at random.

We repeat the learning 7 times for all input data, namely $T = 14000$. We use the same parameters as for Fig. 2(a) except $Cs_0 = 11.5$ for DSOM.

Figures 5(b) and (c) show the learning results of the conventional SOM and DSOM, respectively. From the learning result of SOM, we can see that there are some inactive neurons in the doughnut hole. However, there are no inactive neurons in the result DSOM. This is because neurons located at distant locations is disconnected, so, the neurons can learn more distant for the distant input data. Therefore, DSOM can obtain the more effective map reflecting the distribution state of input data, than the conventional SOM.

6. Conclusions

In this study, we proposed a new SOM algorithm which is the Disconnecting Self-Organizing Map (DSOM). In the initial state, all neurons of DSOM are directly or indirectly connected each other. However, connections between neurons located at distant locations are cut with learning. Each connection between neurons of DSOM has a connection strength. The connection strength between the winner and the neuron, which is the most distant from the input data in a set of direct topological neighbors of the winner, is decreased with each learning. The neuron is disconnected from the winner if the connection strength between them becomes zero.

The learning process and the behaviors of DSOM were investigated with applications to clustering of 2-dimensional input data and feature extraction. We confirmed that the result of using DSOM includes no inactive neuron and DSOM can obtain the more effective map reflecting the distribution state of input data than the conventional SOM.

References

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

[2] 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.

[3] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586-600, 2002.

[4] I. Lapidot, H. Guterman and A. Cohen, "Unsupervised Speaker Recognition Based on Competition Between Self-Organizing Maps," *IEEE Transactions on Neural Networks*, vol. 13, no. 4, pp. 877-887, 2002.

[5] H. Matsushita and Y. Nishio, "Reunifying Self-Organizing Map," *Proc. of NOLTA'06*, pp. 207-210, 2006.

[6] T. Kohonen, *Self-organizing Maps*, Springer, Berlin, 1995.