Self-Organizing Map
Considering False Neighboring Neuron

Haruna MATSUSHITA* and Yoshifumi NISHIO*

* Department of Electrical and Electronic Engineering, Tokushima University
2–1 Minami-Josanjima, Tokushima 770–8506, Japan
Telephone: +81–88–656–7470, Fax: +81–88–656–7471,
Email: {haruna, nishio}@ee.tokushima-u.ac.jp

***Abstract*— In the real world, it is not always true that the next-door house is close to my house, in other words, "neighbors" are not always "true neighbors". In this study, we propose a new Self-Organizing Map (SOM) algorithm which considers the False Neighboring Neuron (called FNN-SOM). The FNN-SOM self-organizes with considering the real neighboring relation. The behavior of FNN-SOM is investigated with learning for various input data. We confirm that we can obtain the more effective map reflecting the distribution state of input data than the conventional SOM.**

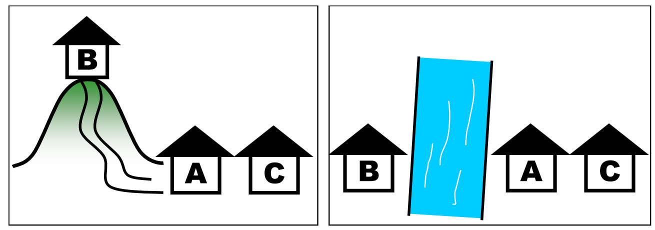
I. Introduction

Since we can accumulate a huge amount of data in recent years, it is important to investigate various clustering methods. Then, the Self-Organizing Map (SOM) has attracted attention for its clustering properties. SOM is an unsupervised neural network introduced by Kohonen in 1982 [1] and is a model simplifying self-organization process of the brain. SOM obtains statistical feature of input data and is applied to a wide field of data classifications. We can obtain the map reflecting the distribution state of input data using SOM. In the learning algorithm of SOM, a winner, which is a neuron with the weight vector closest to the input vector, and its neighboring neuron are updated, regardless of the distance between the input vector and the neighboring neuron. For this reason, if we apply SOM to clustering of the input data which includes some clusters located at distant location, there are some inactive neurons between clusters. Because inactive neurons are on a part without the input data, we are misled into thinking that there are some input data between clusters.

Meanwhile, in the real world, it is not always true that the next-door house is close to my house. For example, the case that the next-door house is at the top of a mountain whereas my house is at the foot (as Fig. 1(a)), and other case that there is a river, which does not have a bridge, between my house and my next-door house (as Fig. 1(b)). This means, "neighbors" are not always "true neighbors".

In addition, the synaptic strength is not constant in the brain. There is scarcely any research changing the synaptic strength.

In this study, we propose a new SOM algorithm which considers the False Neighboring Neuron (called FNN-SOM). A false-neighbor degree is allocated between each neuron of FNN-SOM. The neuron, which is the most distant from the input data in a set of direct topological neighbors of winner,



(a) (b)

Fig. 1. What is the "neighbors"? The house B and C is A's next-door neighbor on the left and on the right, respectively. (a) The house B is at the top of a mountain. (b) The river between A and B does not have a bridge.

is said to be "false neighboring neuron (FNN)". The false-neighbor degrees of FNN and its neighbors are increased, and the false-neighbor degrees act as a burden of the distance between each map node when the weight vectors of neurons are updated. Therefore, the FNN-SOM self-organizes with considering the real neighboring relation.

In Section II, we explain the learning algorithm of the conventional SOM. In Section III, the learning algorithm of the proposed SOM, FNN-SOM, is explained in detail. The learning behaviors of FNN-SOM for various input data are investigated in Section IV. We can see that there are no inactive neurons using FNN-SOM, and FNN-SOM can obtain the more effective map reflecting the distribution state of input data than the conventional SOM. In Section V, FNN-SOM are applied to clustering. Furthermore, clustering ability is evaluated by visualization of results for the iris data.

II. 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 $\boldsymbol{x}_j$;

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

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

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

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

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

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;

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

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

III. SOM considering False Neighboring Neuron (FNN-SOM)

We explain a proposed new SOM algorithm, FNN-SOM. A false-neighbor degree $C_{f(c,i)}$ is allocated between each neuron of FNN-SOM. The initial values of all of the false-neighbor degree are set to zero, and the initial values of all the weight vectors are given over the input space at random.

(FNN-SOM1) An input vector $\boldsymbol{x}_j$ is inputted to all the neurons at the same time in parallel.
(FNN-SOM2) 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).
(FNN-SOM3) A False Neighboring Neuron l (called FNN) is found. FNN is the most distant from $\boldsymbol{x}_j$ in Nc_1. However, if k_l is smaller than the number of Nc_1, the FNN l dose not exist.

$$l = \arg\max_i \{\|\boldsymbol{w}_i - \boldsymbol{x}_j\|\}, \qquad i \in Nc_1, \qquad (5)$$

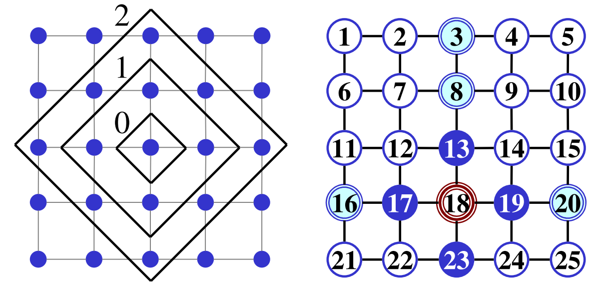
where Nc_1 is the set of direct topological neighbors of c as Fig. 2(a).
The false-neighbor degree between c and l is increased;

$$C_{f(c,l)} = C_{f(c,l)} + 1. \qquad (6)$$

Furthermore, the false-neighbor degrees between c and the set of neurons S_l, which are located beyond l as Fig. 2(b), are also increased;

$$C_{f(c,i)} = C_{f(c,i)} + 1, \qquad i \in S_l. \qquad (7)$$

(FNN-SOM4) The false-neighbor degrees between c and its



(a) (b)

Fig. 2. Neighborhood distances of the rectangular grid. (a) Discrete neighborhoods Nc_0, Nc_1 and Nc_2 of the centermost neuron. (b) Neighboring references of $c = 18$. $Nc_1 = [13, 17, 19, 23]$. If $l = 13$, $S_l = [3, 8]$ and $S_c = [16, 20]$. If $l = 17$, $S_l = 16$ and $S_c = [3, 8, 20]$. If $l = 19$, $S_l = 20$ and $S_c = [3, 8, 16]$. If $l = 23$, S_l does not exist and $S_c = [3, 8, 16, 20]$.

neighbors Nc_1 without l are set to zero (this means "refresh" the connection reference).

$$C_{f(c,i)} = 0, \qquad i \in Nc_1 \cap i \neq l. \qquad (8)$$

Furthermore, the false-neighbor degrees between c and the set of neurons S_c, which are located beyond the neurons in Nc_1 except l as Fig. 2(b), are also set to zero;

$$C_{f(c,i)} = 0, \qquad i \in S_c. \qquad (9)$$

(FNN-SOM5) The weight vectors of the neurons are updated as;

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + hn_{c,i}(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)), \qquad (10)$$

where $hn_{c,i}(t)$ is the neighborhood function of the proposed SOM;

$$hn_{c,i}(t) = \alpha(t)\exp\left(-\frac{\lambda_{c,i}}{4\sigma^2(t)}\right), \qquad (11)$$

where

$$\lambda_{c,i} = 0.5k_i + (\|\boldsymbol{r}_i - \boldsymbol{r}_c\|^2 + C_{f(c,i)}). \qquad (12)$$

(FNN-SOM6) The steps from (FNN-SOM1) to (FNN-SOM5) are repeated for all the input data.

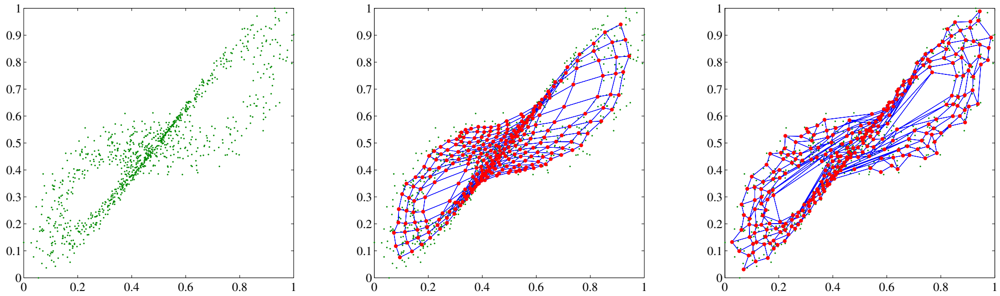
IV. Application to Feature Extraction

A. For Lorenz data

We carry out learning simulation for the data generated by Lorenz equations which is three dimensional nonlinear differential equations and is a chaotic system described by Edward Lorenz [2].

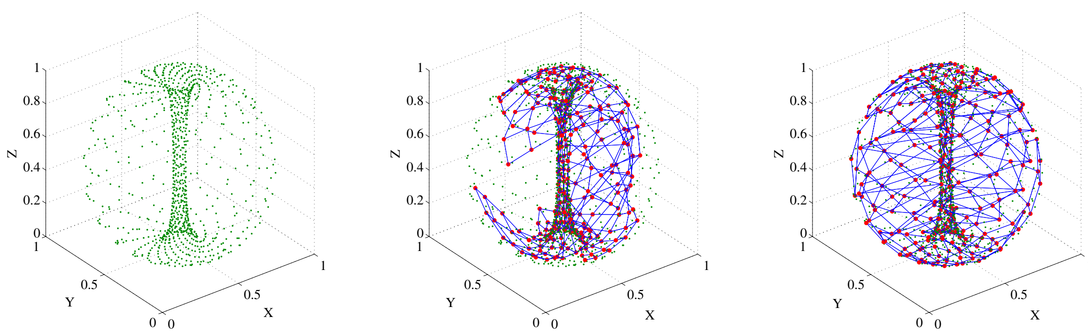
In this study, we use 1_{st} and 2_{nd} dimensional data of Lorenz attractor shown in Fig. 3(a), and is like a butterfly. The total number of data N is 1000. All the input data are sorted by random.

The learning conditions are as follows. Both the conventional SOM and FNN-SOM has $m = 289$ (17×17). We repeat the learning 15 times for all input data, namely $T = 15000$. The parameters of the learning are chosen as follows;
(For SOM)



(a) (b) (c)

Fig. 3. Learning for 2-D data generated by Lorenz equations. (a) Input data. (b) Simulation result of conventional SOM. (c) Simulation result of FNN-SOM.



(a) (b) (c)

Fig. 4. Learning for 3-D data generated by Langford equations. (b) Simulation result of conventional SOM. (c) Simulation result of FNN-SOM.

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

(For FNN-SOM)

$$\alpha(0) = 0.5,\ \alpha(T) = 0.05,\ \sigma(0) = 10,\ \sigma(T) = 0.01.$$

The results of SOM and FNN-SOM are shown in Figs. 3(b) and (c). The conventional SOM can not self-organizes the edge data of the input space because the center are denser area than the edge. However, FNN-SOM self-organizes the center data as well as the edge data of the input space because the neuron located at distant location from winner is defined FNN.

B. For Langford data

Next, we carry out learning simulation for the data generated by Langford equation [3]. This attractor is 2-torus and has the cavity. Figure 4(a) shows the input data. The total number of data N is 1000.

The learning results of SOM and FNN-SOM are shown in Figs. 4(b) and (c). We can see that FNN-SOM can obtain the more effective map reflecting the distribution state of input data than SOM.

V. Application to Clustering

We apply FNN-SOM to clustering.

A. For 2-dimensional input data

First, we consider 2-dimensional input data generated by the probability density function as shown in Fig. 5(a). This data has 5 clusters, however, it is difficult to find 5 clusters because the each cluster is close to each other and the density of each clusters is different. Total number of the input data N is 700. The learning conditions are the same used in Fig. 3.

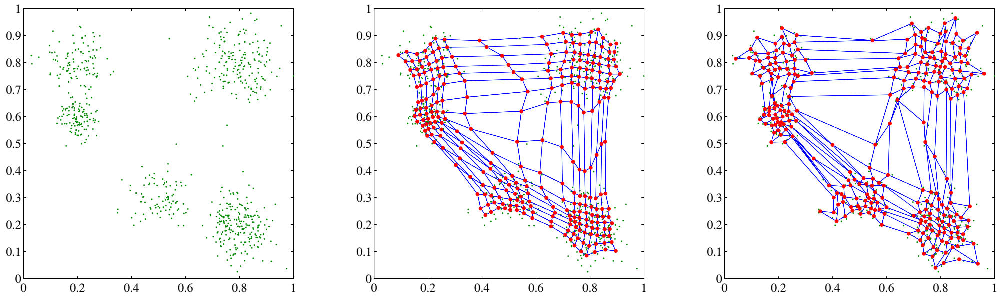
The simulation results of the conventional SOM and FNN-SOM are shown in Figs. 5(b) and (c). We can see that there are some inactive neurons between two clusters in the result of SOM, however, there are few inactive neurons in the result of FNN-SOM. This is because the neurons of FNN-SOM are not affected by FNN, so, the neurons can learn more distant for the distant input data, than SOM learning.

Figure 6 shows distances between neighboring neurons and 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. Each map is normalized by each distance, respectively.

We can see that the boundary line of FNN-SOM is clearer than the conventional SOM because FNN-SOM has few inactive neurons between clusters. Therefore, we can confirm that the input data has 5 clusters from Fig. 6(b).

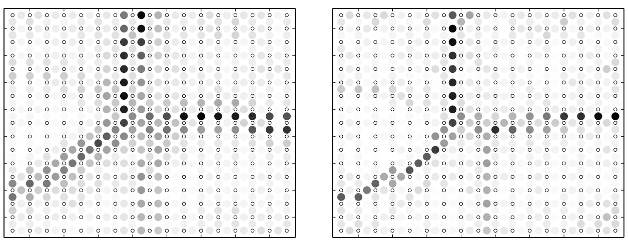
B. For Iris data

Furthermore, we apply FNN-SOM to the real world clustering problem. We use the Iris plant data [4] as real data. This data is one of the best known databased to be found in pattern recognition literatures [5]. The data set contains three clusters of 50 instances respectively, where each class refers to a type of iris plant. The number of attributes is four as the sepal length, the sepal width, the petal length and the petal width, namely, the input data are 4-dimension. The three classes correspond to *Iris setosa*, *Iris versicolor* and *Iris*



(a) (b) (c)

Fig. 5. Clustering for 2-dimensional input data. (a) Input data. (b) Simulation result of the conventional SOM. (c) Simulation result of FNN-SOM.



(a) (b)

Fig. 6. Visualization of result. (a) Conventional SOM. (b) FNN-SOM.

virginica, respectively. *Iris setosa* is linearly separable from the other two, however *Iris versicolor* and *Iris virginica* are not linearly separable from each other.

The learning conditions are as follows. Both the conventional SOM and FNN-SOM has $m = 100$ (10×10). We repeat the learning 100 times for all input data, namely $T = 15000$. The parameters of the learning are chosen as follows;
(For SOM)

$$\alpha(0) = 0.5,\ \sigma(0) = 3,\ \alpha(T) = \sigma(T) = 0$$

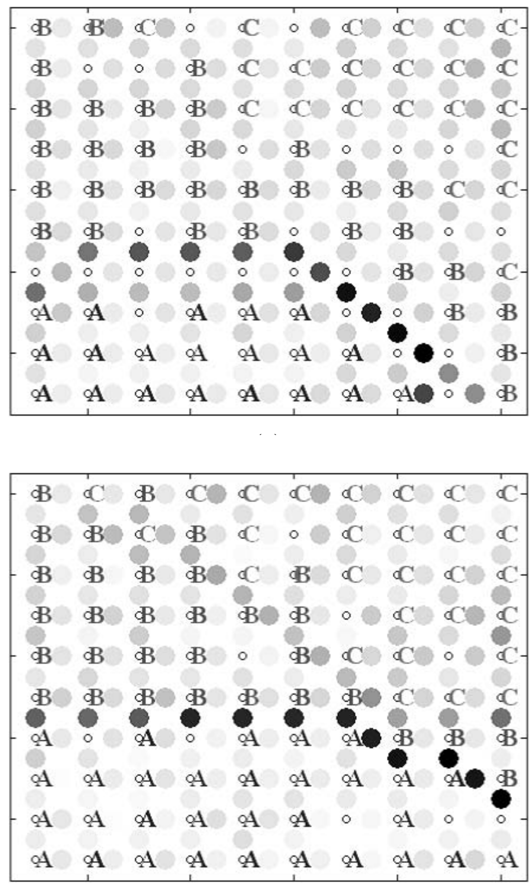
(For FNN-SOM)

$$\alpha(0) = 0.5,\ \alpha(T) = 0.05,\ \sigma(0) = 10,\ \sigma(T) = 0.01.$$

The visualizations of results are shown in Fig. 7. The boundary line of FNN-SOM is clearer than the conventional SOM, and from FNN-SOM result, we can confirm that 3 clusters exists.

VI. Conclusions

In this study, we have proposed a new SOM algorithm which considers the False Neighboring Neuron (called FNN-SOM). The neuron, which is the most distant from the input data in a set of direct topological neighbors of winner, is defined as "false neighboring neuron (FNN)". Furthermore, the false-neighbor degree is allocated between each neuron of FNN-SOM. The false-neighbor degrees of FNN and its neighbors are increased, and the false-neighbor degrees act as a burden of the distance between each map node when the weight vectors of neurons are updated. We have investigated its behaviors with learning for various input data and application to clustering. We have confirmed the efficiency of FNN-SOM.



(b)

Fig. 7. Visualization of result for Iris data. Label A, B and C correspond to *Iris setosa*, *Iris versicolor* and *Iris virginica*, respectively. (a) Conventional SOM. (b) FNN-SOM.

References

[1] T. Kohonen, *Self-organizing Maps*, Berlin, Springer, vol. 30, 1995.
[2] E. N. Lorenz, "Deterministic Nonperiodic Flow", *Journal of the Atmospheric Sciences*, vol.20, pp. 130-141, 1963.
[3] W. F. Langford, "Numerical Studies of torus bifurcations," *International Series of Numerical Mathematics*, vol.70, pp.285-295, 1984.
[4] UCI Repository of Machine Learning Databases and Domain Theories, FTP address: ftp://ftp.ics.uci.edu/pub/machine-learning-databases
[5] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annual Eugenics*, no.7, part II, pp. 179-188,1936.