Self-Organizing Map with Weighted Connections Avoiding False-Neighbor Effects

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— **This study proposes the Self-Organizing Map with Weighted Connections avoiding false-neighbor effects (WC-SOM). We investigate the effectiveness of WC-SOM in comparison with the conventional SOM, Growing Grid and FN-SOM. We confirm that WC-SOM enables the most flexible self-organization among the four algorithms and can obtain the effective map reflecting the distribution state of the input data using fewer neurons.**

I. INTRODUCTION

The Self-Organizing Map (SOM) is an unsupervised neural network [1] and has attracted attention for its clustering properties [2]. In the learning algorithm of SOM, a winner, which is a neuron closest to the input data, and its neighboring neurons are updated, regardless of the distance between the input data and the neighboring neurons. For this reason, if we apply SOM to clustering of the input data including some clusters located at distant locations, there are some inactive neurons between clusters. Because the inactive neurons are on a part without the input data, we are misled into thinking that there are some input data between clusters. Furthermore, because the simulation time depends on the number of neurons, it is important to utilize the used neurons effectively by reducing the inactive neurons.

In the real world, it is not always true that neighboring houses are physically adjacent or close to each other. In addition, the relationship between neighborhoods is not fixed, but keeps changing with time. It is important to change the neighborhood relationship flexibly according to the situation.

Meanwhile, the synaptic strength is not constant in the brain. So far, the Growing Grid network was proposed in 1985 [3]. Growing Grid increases the neighborhood distance between neurons by increasing the number of neurons. However, there is not much research changing the synaptic strength itself.

In our past study, we proposed the SOM with False-Neighbor degree between neurons (called FN-SOM) [5]. False-neighbor degrees (FNDs) are allocated between adjacent rows and adjacent columns of FN-SOM. FNDs act as a burden of the distance between map nodes when the weight vectors of neurons are updated. FN-SOM can greatly reduce the inactive neurons, however, the algorithm has following problems. All the FNDs between the neurons on same line are increased simultaneously and forcibly. It often produces the increase of

FNDs between correct-neighboring neurons, namely, FALSE false-neighbor. Then, it produces the twist of the map.

In this study, we propose a new SOM algorithm, SOM with Weighted Connections avoiding false-neighbor effects (WC-SOM). In WC-SOM, all the connections between adjacent neurons are weighted to avoid false-neighbor effects unlike FN-SOM. This weights are called as false-neighbor weights (FNWs). In the algorithm of WC-SOM, we find winless neurons and its "false neighbors", and FNWs between these neurons are increased. The initial values of all of FNWs are set to zero, however, they are increased with learning. In this way, WC-SOM changes the neighborhood relationship more flexibly according to the situation and the shape of data.

We compare WC-SOM with the conventional SOM, Growing Grid and FN-SOM, and effectiveness of WC-SOM are investigated by applying to various input data. We confirm that WC-SOM can obtain the effective map reflecting the distribution state of the input data using fewer neurons.

II. SOM WITH WEIGHTED CONNECTIONS AVOIDING FALSE-NEIGHBOR EFFECTS (WC-SOM)

The previous method, FN-SOM, has false-neighbor degrees (FNDs) between neurons. However, all the FNDs between neurons at same line on the map are forcibly the same value. It often produces the twist of the map.

In this study, we propose SOM with Weighted Connections avoiding false-neighbor effects (WC-SOM). We explain WC-SOM in detail in this section. WC-SOM consists of $n \times m$ neurons located at 2-dimensional rectangular grid. Each neuron i has a d-dimensional weight vector $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, nm)$. The range of the elements of the 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.

In WC-SOM, a false-neighbor weight (FNW) denoted by $n_{f(i,k)}$ is allocated between directly-connected neurons i and k, and we propose a new neighborhood distance considering FNWs. The initial values of all FNWs are set to zero, and the initial values of all the weight vectors are given over the input space at random. Moreover, a winning frequency γ_i is associated with each neuron i and is set to zero initially.

Learning Step

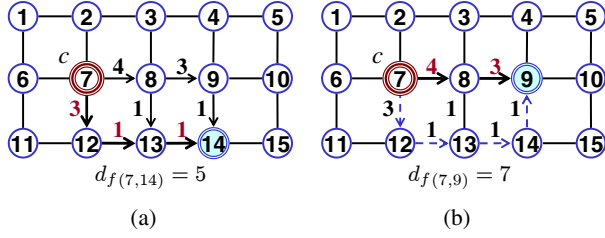**(Step1)** Input an input vector $\boldsymbol{x}_j$ to all the neurons.

Fig. 1. The connection weight $d_{f_{(c,i)}}$ between the winner $c(=7)$ and the neuron i. $d_{f_{(c,i)}}$ is the minimum of sum-of-FNWs on the shortest-path from c to i. Values in circles mean respective neuron numbers. Values between neurons mean FNWs n_f between directly-connected neurons. (a) Let $i = 14$. The possible shortest path routes from c to i are [7 8 9 14], [7 8 13 14] and [7 12 13 14], and its sum of FNWs are 8, 6 and 5, respectively. Then, $d_{f_{(7,9)}} = 5$ which is a minimum value. (b) Let $i = 9$. Then, $d_{f_{(7,9)}} = 7$ because the possible shortest-path route is only [7 8 9]. Note that a route [7 12 13 14 9] is NOT adopted because it is not the shortest-path route although the sum of FNWs is 6 which is smaller than 7.

(Step2) Calculate distances between x_j and all the weight vectors. Find a winner, denoted by c, which is the neuron with the weight vector closest to the input vector x_j.

$$c = \arg \min_i \{\|w_i - x_j\|\}, \qquad (1)$$

where $\| \cdot \|$ is Euclidean distance measure.

(Step3) Increase the winning frequency of the winner c by $\gamma_c^{\text{new}} = \gamma_c^{\text{old}} + 1$.

(Step4) Calculate the neighboring distance $dis(c, i)$ between the winner c and each neuron i by considering FNWs n_f as the following measure ;

$$dis(c, i) = \|r_i - r_c\|^2 + d_{f(c,i)}, \qquad (2)$$

where $\|r_i - r_c\|$ is the Euclidean distance between map nodes c and i on the map grid. $d_{f(c,i)}$ is the connection weight between c and i, and it is defined as the minimum of sum-of-FNWs on the shortest-path from c to i (as shown in Fig. 1).

(Step5) Update the weight vectors of all the neurons:

$$w_i(t+1) = w_i(t) + h_{Fc,i}(t)(x_j - w_i(t)), \qquad (3)$$

where $h_{Fc,i}(t)$ is the neighborhood function of WC-SOM:

$$h_{Fc,i}(t) = \alpha(t) \exp\left(-\frac{dis(c,i)}{2\sigma^2(t)}\right). \qquad (4)$$

where $\alpha(t)$ is the learning rate, and $\sigma(t)$ corresponds to the width of the neighborhood function. Both $\alpha(t)$ and $\sigma(t)$ decrease with time, in this study, we use following equations;

$$\alpha(t) = \alpha_0(1 - t/t_{\max}), \quad \sigma(t) = \sigma_0(1 - t/t_{\max}), \qquad (5)$$

where α_0 and σ_0 are initial values of α and σ, respectively, and t_{max} is the maximum number of the learning.

(Step6) If $\sum_{i=1}^{nm} \gamma_i \geq \lambda$ is satisfied, find the false-neighbor and increase FNWs n_f, according to steps from (Step7) to (Step10). If not, perform step (Step11).

Consideration of False-Neighbor

(Step7) Find a set of neurons S which have never become the winner: $S = \{i \mid \gamma_i = 0\}$. If the winless neuron does not

exist, namely $S = \emptyset$, return to (Step1) without considering the false-neighbor.

(Step8) Choose a false-neighbor f_q of each neuron q in S from the set of direct topological neighbors of q denoted as N_{q_1}. f_q is a neuron whose weight vector is most distant from q:

$$f_q = \arg \max_i \{\|w_i - w_q\|\}, \quad q \in S, \ i \in N_{q_1}. \qquad (6)$$

(Step9) Increase FNW $n_{f(q,f_q)}$ between each q and its false-neighbor f_q as

$$n_{f(q,f_q)} = n_{f(q,f_q)} + 1, \qquad (7)$$

where $n_{f(q,f_q)} = n_{f(f_q,q)}$.

(Step10) Reset the winning frequencies of all the neurons to zero: $\gamma_i = 0$.

(Step11) Repeat the steps from (Step1) to (Step10) for all the input data.

III. EXPERIMENTAL RESULTS

We apply WC-SOM to various input data and compare it with the conventional SOM, Growing Grid and FN-SOM. For all experiments, SOM, FN-SOM and WC-SOM have $nm = 100$ neurons (10×10). Growing Grid starts learning with 2×2 neurons, and the maximum number of neurons is less than 100. The parameters of the learning for WC-SOM are chosen as $\alpha_0 = 0.3$, $\sigma_0 = 3$, $\lambda = 3000$, where we use the same α_0 for all the method and the same λ for FN-SOM and WC-SOM. For SOM, Growing Grid and FN-SOM, we use $\sigma_0 = 4$.

A. For Chainlink data set

We considered a 3-dimensional input data which is Chainlink data set [6] with clustering problems of the linearly-inseparable. The total number of the input data N is 1000, and the input data has two clusters. We repeated the learning 20 times for all the input data, namely $t_{\max} = 20000$.

Learning results of the conventional SOM, Growing Grid are shown in Figs. 2(a) and (b). We can see that there are a lot of inactive neurons between clusters. The other side, in the learned maps of FN-SOM and WC-SOM shown in Figs. 2(c) and (d), there are only a few inactive neurons between clusters.

Furthermore, in order to compare the learning performance of WC-SOM with the other SOMs numerically, we use the following well-used two measurements.

Quantization Error Qe measures the average distance between each input vector and its winner [1]. The small value Qe is more desirable.

Neuron Utilization U measures the percentage of neurons that are the winner of one or more input vectors in the map [4]. Thus, U nearer 1.0 is more desirable.

We carry out 30 simulations with different initial state of the weight vectors and different order of inputting. The averages of the two measurements over 30 independent runs are listed in Table I. The quantization error Qe of WC-SOM is the best value among the four algorithms and has improved by 12.3%
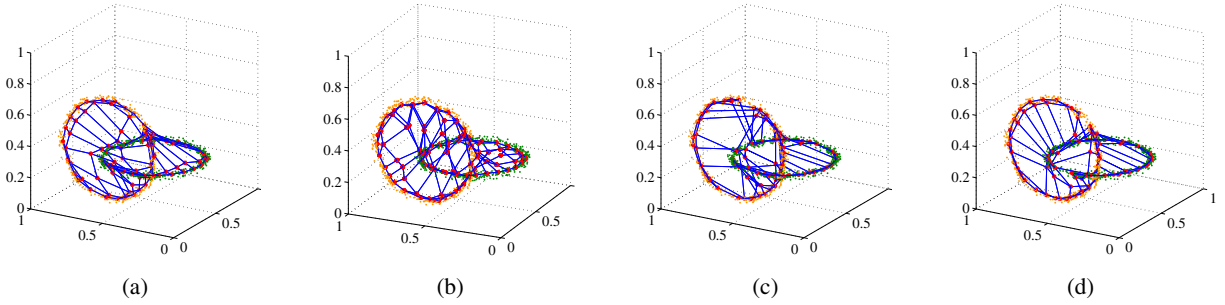
Fig. 2. Learning results of four algorithms for Chainlink data. (a) Conventional SOM. (b) Growing Grid. (c) FN-SOM. (d) WC-SOM.
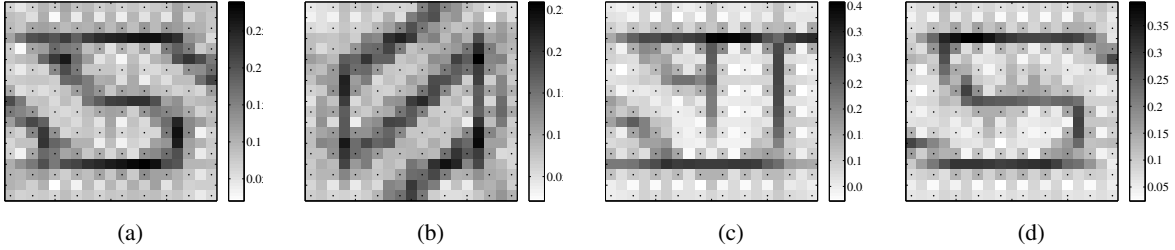

Fig. 3. U-Matrices of simulation results for Chainlink data. (a) Conventional SOM. (b) Growing Grid. (c) FN-SOM. (d) WC-SOM.

TABLE I
TWO MEASUREMENTS FOR CHAINLINK DATA.

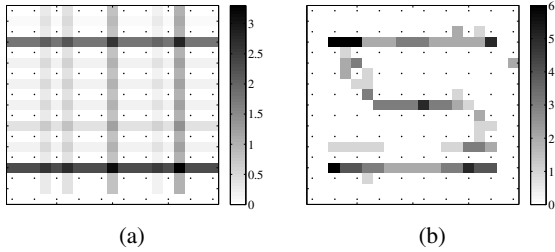|	SOM	Growing Grid	FN-SOM	WC-SOM
Qe	0.0243	0.0299	0.0221	0.0213
U	0.776	0.7049	0.869	0.8757

Fig. 4. Gray scale display of false-neighbor degrees and false-neighbor weights after learning for Chainlink data. (c) False-neighbor degrees of FN-SOM. (d) False-neighbor weights of WC-SOM.

compared to that of SOM. This is because the result of WC-SOM has few inactive neurons, therefore, more neurons can self-organize the input data. This is confirmed by the neuron utilization U. U of WC-SOM is the best value among the four algorithms. It means that 87.6% of the neurons of WC-SOM are the winner of one or more input data, in other words, there are only 12.4% inactive neurons.

In order to evaluate how well SOM preserves the topology of the data set, we calculate U-Matrix [6] which visualizes the cluster structure of the map by showing distances between neighboring neurons. Figure 3 shows U-Matrices of the four algorithms. We can see that the boundary lines of FN-SOM and WC-SOM are clearer than other two algorithms and it

is easy to distinguish between light areas (cluster) and dark areas (no input data) because there are few inactive neurons in the results of FN-SOM and WC-SOM. However, it should be noted that it is clear that U-Matrix of FN-SOM does not visualize the cluster structure correctly. It means that FN-SOM can not preserve the topology of the data set by increasing its FNDs forcibly.

This can be confirmed by differences between FND of FN-SOM and FNW of WC-SOM after learning. Figure 4 shows FND and FNW between the neurons displayed by gray-scale. From FNDs of FN-SOM as Fig. 4(a), we can clearly see that even FNDs between the correct-neighboring neurons were increased. This is because that all the FNDs between the neurons on same line are increased simultaneously and forcibly, in FN-SOM algorithm. On the other hand, in Fig. 4(b), WC-SOM can adapt its FNWs flexibly to the shape of the input data.

From these results measured in terms of the quantization error, the number of inactive neurons and the topographic error, we can say that WC-SOM enables more flexible self-organization than FN-SOM and other SOMs.

B. For Target data set

Next, we considered Target data set [6] which has a clustering problem of outliers. The input data is 2-dimension and has six clusters including 4 outliers, and the total number of the input data N is 770. We repeated the learning 26 times for all the input data, namely $t_{\max} = 20020$.

From the learning results of the four algorithms shown in Fig. 5, it is clear that FN-SOM and WC-SOM can greatly reduce the number of inactive neurons in comparison with the conventional SOM. It is conformed by the measurements
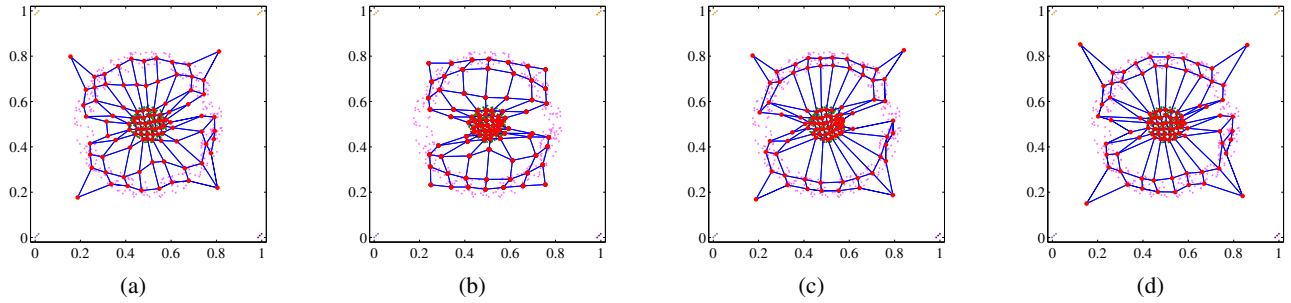
Fig. 5. Learning results of four algorithms for Target data. (a) Conventional SOM. (b) Growing Grid. (c) FN-SOM. (d) WC-SOM.
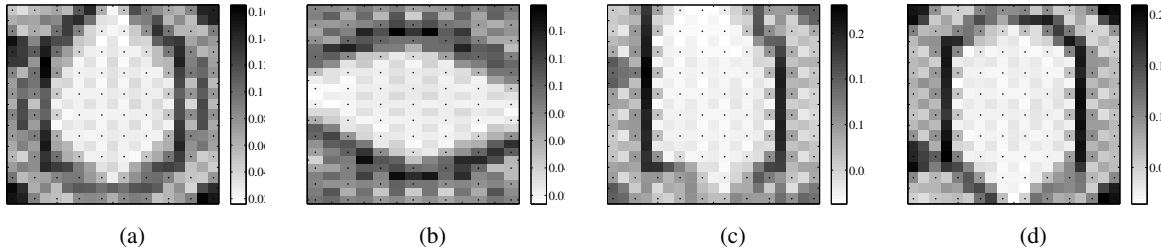

Fig. 6. U-Matrices of simulation results for Target data. (a) Conventional SOM. (b) Growing Grid. (c) FN-SOM. (d) WC-SOM.

TABLE II
TWO MEASUREMENTS FOR TARGET DATA.

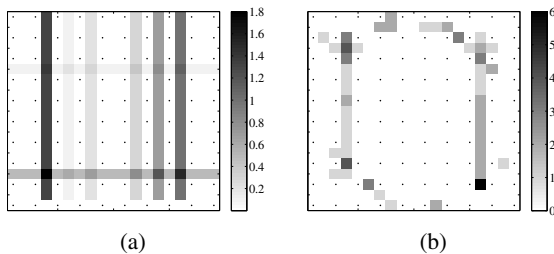|	SOM	Growing Grid	FN-SOM	WC-SOM
Qe	0.0196	0.0232	0.0190	0.0175
U	0.8113	0.8317	0.9263	0.8917

Fig. 7. Gray scale display of false-neighbor degrees and false-neighbor weights after learning for Target data. (c) False-neighbor degrees of FN-SOM. (d) False-neighbor weights of WC-SOM.

listed in Table II. It should be noted that the quantization error Qe of WC-SOM is better than FN-SOM although the neuron utilization U of FN-SOM is the best value. This means that there are few inactive neurons in the result of FN-SOM, however, FN-SOM has not self-organized the statistical features of the input data correctly. In other words, WC-SOM can obtain the clustering map reflecting the distribution state of the input data more effectively than FN-SOM.

This is backed by the U-Matrices shown in Fig. 6. The boundary lines of FN-SOM is clear, however, we can not say that it visualizes the cluster structure correctly. Meanwhile, U-Matrix of WC-SOM visualizes the cluster structure effectively, and its boundary lines are very clear. From FNDs and FNWs

after learning shown in Fig. 7, FNWs of WC-SOM has be increased flexibly according to the shape of the input data in comparison with FN-SOM.

IV. CONCLUSIONS

This study has proposed the Self-Organizing Map with Weighted Connections (WC-SOM). WC-SOM has false-neighbor weights (FNWs) allocated between connections of adjacent neurons to avoid false-neighbor effects. In the algorithm, the calculation method of the neighborhood distance has been proposed in accordance with FNWs. We have compared WC-SOM with the conventional SOM, Growing Grid and FN-SOM and have confirmed that WC-SOM enabled the most flexible self-organization by increasing FNWs with learning. Furthermore, we have confirmed visually and numerically that WC-SOM had few inactive neurons. These results mean that WC-SOM can obtain the effective map reflecting the distribution state of the input data using fewer neuron.

REFERENCES

[1] T. Kohonen, *Self-organizing Maps*, Berlin, Springer, 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] B. Fritzke, "Growing Grid – a self-organizing network with constant neighborhood range and adaptation strength," *Neural Processing Letters*, vol. 2, no. 5, pp. 9–13, 1995.
[4] Y. Cheung and L. Law, "Rival-Model Penalized Self-Organizing Map," *IEEE Trans. Neural Networks*, vol. 18, no. 1, pp. 289–295, 2007.
[5] H. Matsushita and Y. Nishio, "Self-Organizing Map with False-Neighbor Degree between Neurons for Effective Self-Organization," *IEICE Transactions on Fundamentals*, vol. E91-A, no. 6, pp. 1463–1469, 2008.
[6] A. Ultsch, "Clustering with SOM: U*C," *Proc. of WSOM'05*, pp. 75–82, 2005.