2007 IEEE Workshop on Nonlinear Circuit Networks
NCN'07, Tokushima, Japan, Dec 7--8, 2007

Clustering of Self-Organizing Map Considering Winning Frequency

Taku Haraguchi, Masato Tomita, Haruna Matsushita and Yoshifumi Nishio
Department of Electrical and Electronic Engineering,
Tokushima University
Email: taku@ee.tokushima-u.ac.jp, tomita@ee.tokushima-u.ac.jp,
haruna@ee.tokushima-u.ac.jp, nishio@ee.tokushima-u.ac.jp

***Abstract*— In this study, we propose a new Self-Organizing Map (SOM) algorithm considering Winning Frequency (called WF-SOM). The WF-SOM is that each neuron has individuality by considering winning frequency. We confirm its exact and quick self-organization in the case of the small number of learning.**

I. Introduction

The Self-Organizing Map (SOM) [1][2] is an unsupervised neural network introduced by Teuvo Kohonen. SOM has attracted attention in studies of clustering in recent years. In this study, we propose a new type of SOM algorithm, which is called SOM considering Winning Frequency (WF-SOM) algorithm. The most important feature of WF-SOM is that each neuron has individuality by considering winning frequency. If a neuron becomes a winner less frequency and is nearer to the winner neuron, the neuron is updated more significantly. We investigate the behavior of WF-SOM and apply WF-SOM to clustering problems. The efficiencies of WF-SOM are confirmed by several simulation results.

II. Self-Organizing Map (SOM)

SOM has two-layer structure of the input layer and the competitive layer. In the input layer, there are d-dimensional input vectors $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jd})$ $(j = 1, 2, \cdots, M)$. In the competitive layer, m neurons are arranged on the 2-dimensional grid. Besides, each neuron has a weight vectors $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{id})$ $(i = 1, 2, \cdots, M)$ with the same dimension as the input vector. The range of the input vector is assumed to be between 0 and 1. The initial values of all the weight vectors are given between 0 and 1 at random.

(SOM1) An input vector $\boldsymbol{x}_j$ is inputted to all the neurons at same time in parallel.
(SOM2) We find the winner neuron by calculating the distances between the input vector $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_i$ of neuron i. Winner neuron c is the neuron with the weight vector nearest 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, Euclidean distance.
(SOM3) The weight vector of all the neuron 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 follows;

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

where $\alpha(t)$ is called the learning rate, $\boldsymbol{r}_i$ and $\boldsymbol{r}_c$ are the vectorial locations on the display grid, and $\sigma(t)$ corresponds to the widths of the neighborhood function. Both $\alpha(t)$ and $\sigma(t)$ decrease monotonically with time as follows;

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

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

III. SOM considering Winning Frequency (WF-SOM)

We explain the learning algorithm of WF-SOM in detail. In the WF-SOM algorithm, if the winning frequency of a neuron is smaller and the neuron is nearer to the winner neuron, the neuron is updated more significantly. Namely, even if it is seldom a winner, though it is near the winner neuron, it is updated significantly. Slick neurons always exist. This is similar to an unreasonable situation at human society. The initial values of all the weight vectors are given between 0 and 1 at random. The winning frequency of all the neurons are set to zero : S_c=0.

(WF-SOM1) An input data is inputted to all the neurons at the same time in parallel.
(WF-SOM2) We find the winner neuron c by calculating the distance between $\boldsymbol{x}_j$ and $\boldsymbol{w}_i$ according to Eq. (1).
(WF-SOM3) The winning frequency of winner c is increased by

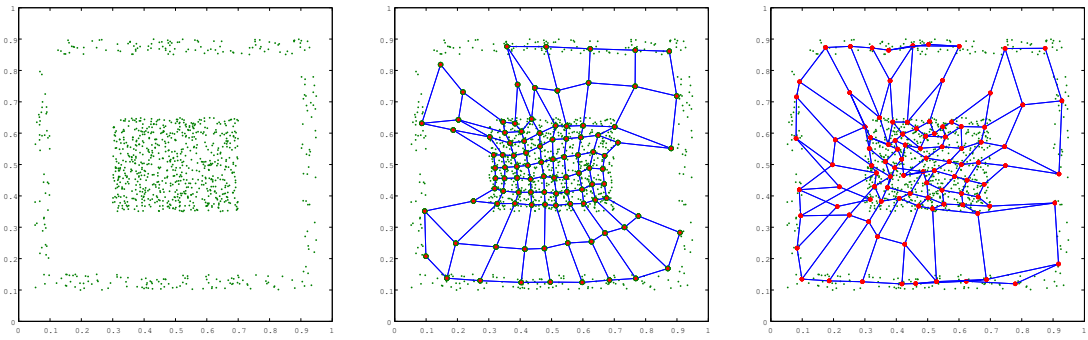
$$S_c^{\text{new}} = S_c^{\text{old}} + 1. \tag{5}$$

(WF-SOM4) The weight vectors of all the neuron are updated with considering winning frequency;

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + h_c(S_c, t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)), \tag{6}$$

where $h_c(S_c, t)$ is a gaussian function described as

$$h_c(S_c, t) = p_c(S_c) \exp\left(-\frac{(\|\boldsymbol{r}_c - \boldsymbol{r}_i\|)^2}{2\sigma_{WF}^2(S_c, t)}\right), \tag{7}$$



(a) (b) (c)

Fig. 1. Learning Simulation. (a) Input data. (b) Simulation result of conventional SOM. (c) Simulation result of WF-SOM.

where $p_c(S_c)$ is described as

$$p_c(S_c) = \alpha_{WF}(S_c) \exp\left(-\frac{S_c}{M}\right), \tag{8}$$

where $\alpha(S_c)$ is the learning rate, and $\sigma(S_c, t)$ corresponds to the width of the neighborhood function. $\alpha(S_c)$ decrease with the number of being winner as follows and $\sigma(S_c, t)$ decrease with time and the number of being winner as follows;

$$\begin{aligned} \alpha_{WF}(S_c) &= \alpha_{WF}(0)\frac{S_c}{M}, \\ \sigma_{WF}(S_c, t) &= \sigma_{WF}(0)\frac{S_c}{t}, \end{aligned} \tag{9}$$

If the winning frequency of a neuron is smaller and the neuron is nearer to the winner neuron, the neuron is significantly updated.

(WF-SOM5) The steps from (WF-SOM1) to (WF-SOM4) are repeated for all the input data.

IV. Learning Simulation

Input data is 2-dimensional random data of 1000 points whose distribution is non-uniform as Fig.1(a). We repeat the learning 3 times for all the input data. This is smaller than the general simulation as the learning times. We consider the conventional SOM and the proposed WF-SOM with M=100 neurons (10×10). The parameters of the learning are chosen as follows;

(For SOM)

$$\alpha(0) = 0.9, \quad \sigma(0) = 3, \tag{10}$$

(For WF-SOM)

$$\alpha_{WF}(0) = 0.9, \quad \sigma_{WF}(0) = M/3, \tag{11}$$

Figures 1(b) and (c) show the learning results of the conventional SOM and WF-SOM, respectively. We can see that the conventional SOM does not self-organize all the corners of the input in the case of the small number of learning. On the other hand, WF-SOM can self-organize all the corners of the input although the number of learning is small. From these figures, we can say that WF-SOM can self-organize more exactly and quickly than the conventional SOM. However, WF-SOM tends to produce twists between the neurons.

In order to evaluate the mapping precision of WF-SOM, we define a quantization error e_q[2].

Quantization Error e_q: This measures the average distance between each input vector and its winner;

$$e_q = \frac{1}{N}\sum_{j=1}^{N} \|\boldsymbol{x}_j - \bar{\boldsymbol{w}}_j\|, \tag{12}$$

where $\bar{\boldsymbol{w}}_j$ is the weight vector of the corresponding winner of the input vector $\boldsymbol{x}_j$. Therefore, the small value e_q is more desirable. If the weight vector of the winner neuron is exactly the same as input data, the value of e_q is 0.

The calculated results are summarized in Table 1. We can evaluate the effectiveness of the method using WF-SOM. Furthermore, the improvement rate is 12.3 [%].

TABLE I

QUANTIZATION ERROR FOR 2-DIMENSIONAL INPUT DATA.

	Conventional SOM	WF-SOM
e_q	0.0244	0.0214

V. Conclusions

In this study, we have proposed WF-SOM. We have explained the differences between SOM and WF-SOM with learning algorithm and have investigated its behavior. Furthermore, we have applied the proposed WF-SOM to clustering problems in the case of the small number of learning and have confirmed its exact and quick self-organization.

In the future, we try to reduce the twists between neurons.

Acknowledgment

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.