2005 RISP International Workshop on Nonlinear
Circuit and Signal Processing (NCSP'05)
Hawaii, USA, Mar. 4-6, 2005.

Research on Improvement in Self-Organization Capability Using Two SOMs

Haruna MATSUSHITA† , Yoko UWATE† and Yoshifumi NISHIO†

† Department of Electrical and Electronics Engineering
Tokushima University
2-1 Minami-Josanjima, Tokushima, Japan
Phone:+81-88-656-7470, Fax:+81-88-656-7471
Email: {haruna, uwate, nishio}@ee.tokushima-u.ac.jp

1. Introduction

The Self-Organizing Map (SOM) is unsupervised neural networks introduced by Kohonen in 1982 [1]. Self-organization is to change an internal structure to adjust to the signal from the outside though teacher signals are not given from the outside. SOM is a model simplifying self-organization process of the brain. SOM obtains a statistical feature of input data and is applied to a wide field of data classifications. And, the effectiveness is said to be very high [2]-[7].

SOM has the position relation between neurons with d-dimensions, and the neuron which won a competition, and some neurons around the winner learn. For example, if input data are 2-dimensional random data which are uniformly-distributed between 0 and 1, SOM is distributed like a square. Thus, a distribution of input data can be expressed with a weight vector by learning. However, when input data is nonuniform, some experience is needed to extract the dense part.

In this research, in order to solve this problem, we propose a method of using simultaneously two kinds of SOMs whose features are different. One is distributed in the area for which input data gathers and the other over the whole. As a result, the feature of data can be extracted more effectively by using two kinds of SOMs.

2. Self-Organizing Map

We concretely explain the learning method of SOM. m neurons are arranged on a 2-dimensional grid as shown in Fig. 1(a). The range of the elements of l-dimensional input vectors $\boldsymbol{x}_j = (x_{j1}, x_{j2}, \cdots, x_{jl})(j = 1, 2, \cdots, N)$ are assumed to be from 0 to 1.

(SOM1) The initial values of all the weight vectors $\boldsymbol{w}$ 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) The internal activity degree net_i^j is calculated from the distance between the input vector $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_i = (w_{i1}, w_{i2}, \cdots, w_{il})$ $(i = 1, 2, \cdots, m)$ of the neuron i, according to:

$$net_i^j = \|\boldsymbol{w}_i - \boldsymbol{x}_j\|^{-1}. \quad (1)$$

In this study, Euclidean distance is used for (1).

(SOM4) The neuron with the maximum internal activity degree is winner neuron c. In other words, the winner neuron c is the neuron with the weight vector nearest to the input vector.

(SOM5) The weight vector of the winner neuron c and the neighborhood neuron N_c (as Fig. 1(b)) are updated as:

$$\boldsymbol{w}_i(t+1) = \boldsymbol{w}_i(t) + \alpha(t)(\boldsymbol{x}_j - \boldsymbol{w}_i(t)),\ i \in N_c(t), \quad (2)$$

where $\alpha(t)$ is the learning coefficient. $\alpha(t)$ and $N_c(t)$ are decreased with time according to the following equations:

$$\alpha(t) = \alpha_0\,(1 - t/T)\,, \quad (3)$$

$$N_c(t) = [N_c(0)\,(1 - t/T)]\,, \quad (4)$$

where [ ] denotes the Gauss' notation, T is the maximum number of learning, α_0 is the initial value of the learning coefficient, $N_c(0)$ is the initial value of the neighborhood coefficient.

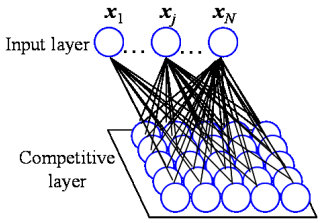
(SOM6) The steps from (SOM2) to (SOM5) are repeated for all the input vectors.

3. Two SOMs

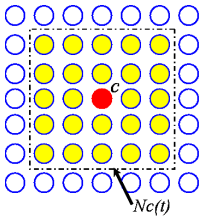
In this study, we propose a method of extracting the feature of the input data more effectively using two kinds of SOMs whose features are different. Namely, one self-organizes the area on which input data are concentrated, and the other self-organizes the whole of the input space. We call the former $\mathrm{SOM_L}$ and the latter $\mathrm{SOM_G}$.

We explain the learning method of the proposed SOMs.

(2-SOM1) The initial values of all the weight vectors $\boldsymbol{w}_L$ of $\mathrm{SOM_L}$ are given between 0 and 1 at random. The



(a) Structure of SOM.



(b) Example of neighborhood area.

Figure 1: Self-organiging map.

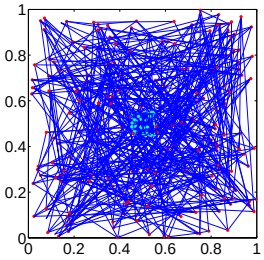


Figure 2: Example of initial states of two SOMs

initial values of all the weight vectors $\boldsymbol{w}_G$ of $\mathrm{SOM_G}$ are given between 0.4 and 0.6 at random, as Fig. 2.

(2-SOM2) An input vector $\boldsymbol{x}_j$ is inputted to all the neurons of $\mathrm{SOM_G}$ and $\mathrm{SOM_L}$ at the same time in parallel.

(2-SOM3) The distance between $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_G = (w_{Gi1}, w_{Gi2}, \cdots, w_{Gil})$ of the neuron i of $\mathrm{SOM_G}$, and the distance between $\boldsymbol{x}_j$ and the weight vector $\boldsymbol{w}_L = (w_{Li1}, w_{Li2}, \cdots, w_{Lil})$ of the neuron i of $\mathrm{SOM_L}$ are calculated. The internal activity degree $net_{G_i}^j$ and $net_{L_i}^j$ are obtained as:

$$net_{G_i}^j = \|\boldsymbol{w}_{Gi} - \boldsymbol{x}_j\|^{-1}, \tag{5}$$

$$net_{L_i}^j = \|\boldsymbol{w}_{Li} - \boldsymbol{x}_j\|^{-1}. \tag{6}$$

(2-SOM4) The winner neuron c is the neuron with the maximum internal activity degree in all net_G and net_L.

(2-SOM5) If the winner neuron c is the neuron in $\mathrm{SOM_G}$, the weight vector of the winner neuron and the neighborhood neuron N_{Gc} are updated as:

$$\boldsymbol{w}_{Gi}(t+1) = \boldsymbol{w}_{Gi}(t) + \alpha_G(t)(\boldsymbol{x}_j - \boldsymbol{w}_{Gi}(t)),\ i \in N_{Gc}(t), \tag{7}$$

where $\alpha_G(t)$ is the learning coefficient.

If the winner neuron c is in $\mathrm{SOM_L}$, the weight vector of the winner neuron and the neighborhood neuron N_{Lc} are updated as:

$$\boldsymbol{w}_{Li}(t+1) = \boldsymbol{w}_{Li}(t) + \alpha_L(t)(\boldsymbol{x}_j - \boldsymbol{w}_{Li}(t)),\ i \in N_{Lc}(t), \tag{8}$$

where $\alpha_L(t)$ is the learning coefficient. $\alpha_G(t)$, $\alpha_L(t)$, N_{Gc}, and N_{Lc} are decided as:

$$\alpha_G(t) = \alpha_{G0}\,(1 - t/T)\,, \tag{9}$$

$$\alpha_L(t) = \alpha_{L0}\left\{1 - (t/T)^{\frac{1}{2}}\right\}, \tag{10}$$

$$N_{Gc}(t) = \left[N_{Gc}(0)\left(\frac{N_{Gc}(0)-1}{N_{Gc}(0)-n}\right)(1 - t/T)\right], \quad N_{Gc}(t) < N_{Gc}(0), \tag{11}$$

$$N_{Lc}(t) = \left[N_{Lc}(0)\,(1 - t/T)\right], \tag{12}$$

where α_{G0} and α_{L0} are the initial values of learning coefficients, $N_{Gc}(0)$ and $N_{Lc}(0)$ are the initial values of neighborhood coefficients and n is the number of all SOMs.

(2-SOM6) The steps from (2-SOM2) to (2-SOM5) are repeated for all the input vectors.

(2-SOM7) Furthermore, only $\mathrm{SOM_G}$ learns. Time is reset as $t = 0$. We compute $net_{G_i}^j$ and $net_{L_i}^j$ according to the steps (2-SOM2) and (2-SOM3).

(2-SOM8) If the distance between $\boldsymbol{x}_j$ and the neuron of $\mathrm{SOM_L}$ with the maximum net_L is larger than a small distance ε, the weight vector of the neuron of $\mathrm{SOM_G}$ with the maximum net_G and the neighborhood neuron N_{Gsc} are updated as:

$$\boldsymbol{w}_{Gi}(t+1) = \boldsymbol{w}_{Gi}(t) + \alpha_{Gs}(t)(\boldsymbol{x}_j - \boldsymbol{w}_{Gi}(t)),\quad i \in N_{Gsc}(t),\ \ \max(net_{L_i}^j) \le \tfrac{1}{\varepsilon}, \tag{13}$$

where $\alpha_{Gs}(t)$ is the learning coefficient.

(2-SOM9) The steps from (2-SOM7) to (2-SOM8) are repeated for all the input vectors. $\alpha_{Gs}(t)$ and $N_{Gsc}(t)$ are decreased with time according to the following equations:

$$\alpha_{Gs}(t) = \alpha_{Gs0}\,(1 - t/T_s)\,, \tag{14}$$

$$N_{Gsc}(t) = \left[N_{Gsc}(0)\,(1 - t/T_s)\right], \tag{15}$$

where T_s is the maximum number of learning since step (2-SOM7),

4. Simulation Results

4.1. Input Data

Input data is 2-dimensional random data of 450 points whose distribution is non-uniform as Fig. 3. 225 points are distributed within a small range from 0.2 to 0.45 horizontally and from 0.7 to 0.95 vertically. The remaining 225 points are uniformly distributed between 0 and 1.

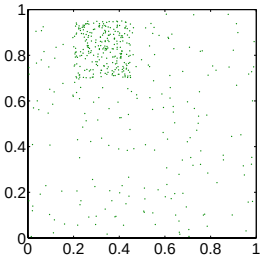


Figure 3: 2-dimensional input data.

4.2. Setting

4.2.1. One SOM

SOM has 441 neurons (21×21). The initial values of the learning coefficient and the neighborhood coefficient are chosen as follows:

$$\alpha_0 = 0.9, \quad N_c(0) = 7.$$

In the learning process, all the input data are given 8 times repeatedly.

4.2.2. Two SOMs

$\mathrm{SOM_G}$ and $\mathrm{SOM_L}$ have 225 neurons (15×15) each, namely, 2 SOMs have totally 445 neurons. The initial values of the learning coefficients, the neighborhood coefficients and the small distance are chosen as follows:

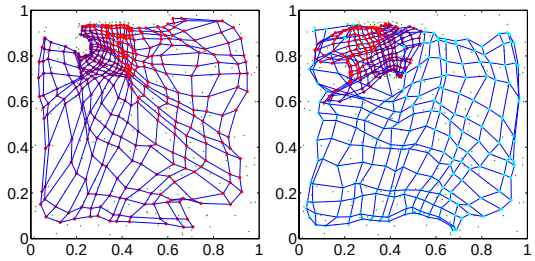
$$\alpha_{G0} = \alpha_{L0} = 0.9, \quad N_{Gc}(0) = N_{Lc}(0) = 6,$$

$$\alpha_{Gs0} = 0.7, \quad N_{Gsc}(0) = 4, \quad \varepsilon = 0.02.$$

After we give all the input data to $\mathrm{SOM_G}$ and $\mathrm{SOM_L}$ 4 times, furthermore we give them for learning only of $\mathrm{SOM_G}$ 4 times repeatedly.

4.3. Results

The simulation result is shown in Fig. 4. From Fig. 4(a), we can say that One SOM is self-organized. However, Two SOMs can be self-organized more effectively as shown in Fig. 4(b). In addition, $\mathrm{SOM_L}$ is self-organized in the area for which input data concentrate, and $\mathrm{SOM_G}$ is self-organized over the whole input space.



(a) One SOM. (b) Two SOMs.

Figure 4: Simulation Results.

5. *n* SOMs

The number of $\mathrm{SOM_L}$ can be increased depending on distribution of input data. We carry out simulation for various input data. For instance, when the data of Fig. 5(a) are input, one $\mathrm{SOM_G}$ and two $\mathrm{SOM_L}$ are effective. The learning method for 2 or more $\mathrm{SOM_L}$ is similar to the case of Two SOMs.

Figure 6 shows the respective results. In all the results, $\mathrm{SOM_L}$ is self-organized in the area for which input data concentrate, and $\mathrm{SOM_G}$ is self-organized over the whole input space.

We consider that the concept using two kinds of SOMs can be used to extract the data only in a dense part of the input data, because the $\mathrm{SOM_L}$ can find such an area by themselves. Figures 7(a) and (b) show one of the $\mathrm{SOM_L}$ and the data whose nearest neuron is in the $\mathrm{SOM_L}$, respectively, after the self-organization in the previous simulation of Fig. 6(c). The data in Fig. 7(b) is almost same as one of the dense parts of the input data in Fig. 5(c). We can also extract other dense parts of the input data in Fig. 5(c) using the other $\mathrm{SOM_L}$ as shown in Fig. 8.

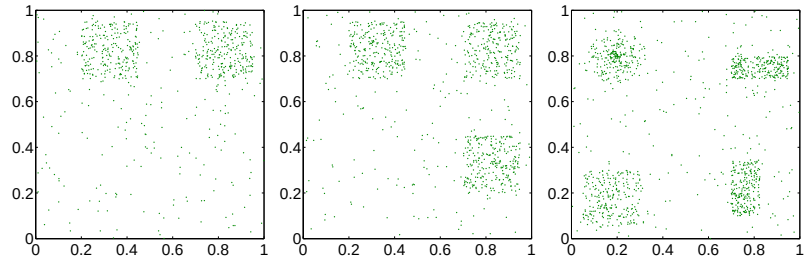
6. Conclusions

In this study, we proposed the method of extracting the feature of the input data more effectively using the two kinds of SOMs whose features are different. The number of $\mathrm{SOM_L}$ is increased depending distribution of input data. We confirmed by computer simulations that n SOMs could extract the dence parts of the input data. In the future, we hope to develop this method further and to apply it to various data.

References

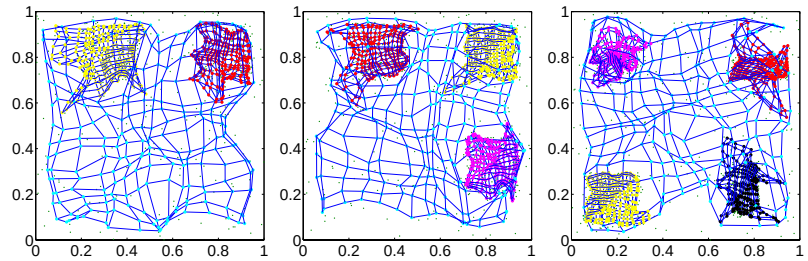
[1] T. Kohonen, "Self-Organising Maps," 2nd ed, Springer Series in Information Sciences, Vol. 30, Springer, Berlin, 1995.

[2] B. Angeniol, G. de La C. Vaubois and J. Y. Le Texier, "Self-Organizing Feature Maps and the Travelling Sales-



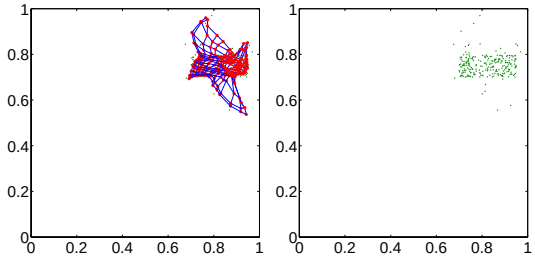
(a) With 2 dense parts. (b) With 3 dense parts. (c) With 4 dense parts.

Figure 5: Input data with various distributions.



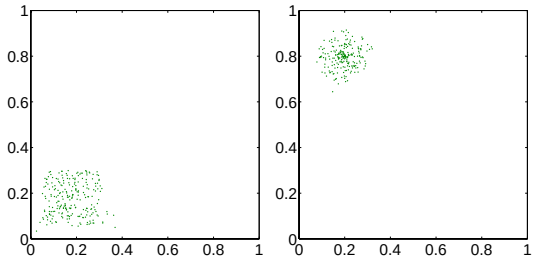
(a) 3 SOMs. (b) 4 SOMs. (c) 5 SOMs.

Figure 6: Self-organization by n SOMs.

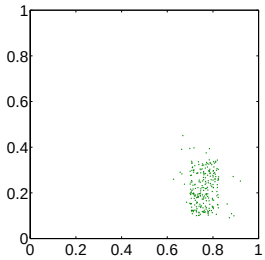


(a) $\mathrm{SOM}_{\mathrm{L}1}$. (b) Extracted data.

Figure 7: Extraction of dense parts of input data by $\mathrm{SOM}_{\mathrm{L}}$.



(a) Extraction by $\mathrm{SOM}_{\mathrm{L}2}$. (b) Extraction by $\mathrm{SOM}_{\mathrm{L}3}$.



(c) Extracted by $\mathrm{SOM}_{\mathrm{L}4}$.

Figure 8: Dense parts extraction.

man Problem," Neural Networks, Vol. 1, pp. 289-293, 1988

[3] H. Tokutaka, "Condensed Review of SOM and LVQ Research in Japan," Proc. of WSOM'97, pp. 322-329, 1997.

[4] M. Cottrell, E. de Bodt, M. Verleysen, "Kohonen Maps Versus Vector Quantization for Data Analysis," Proc. of ESANN'97, pp. 187-193, 1997.

[5] G. Simon, A. Lendasse, M. Cottrell, J.-C. Fort, M. Verleysen, "Double SOM for Long-term Time Series Prediction," Proc. of WSOM'03, pp. 35-40, 2003.

[6] M. Strickert and B. Hammer, "Neural Gas for Sequences," Proc. of WSOM'03, pp. 53-58, 2003.

[7] T. Ehara, H. Sasamura, T. Saito, "An Approach to the TSP based on Growing Self-Organizing Maps," Proc. of NOLTA'04, pp. 709-712, 2004.