Relation Organization of SOM Initial Map by Improved Node Exchange

MIYOSHI Tsutomu
Department of Information and Electronics, Tottori University, Tottori, JAPAN
Email: mijosxi@ike.tottori-u.ac.jp

Abstract—**The Self Organizing Map (SOM) involves neural networks, that learns the features of input data thorough unsupervised, competitive neighborhood learning. In the SOM learning algorithm, connection weights in a SOM feature map are initialized at random values, which also sets nodes at random locations in the feature map independent of input data space. The move distance of output nodes increases, slowing learning convergence. As precedence research, we proposed the method to improve this problem, initial node exchange by using a part of feature map. In this paper, we propose two improved exchange method, node exchange with fixed neighbor area and spiral node exchange. The node exchange with fixed neighbor area uses fixed position of winner node and fixed initial size of neighbor area that sets to cover whole feature map. We investigate how average move distance of all nodes and average deviation of move distance would change with the differences by type of fixed neighbor area in node exchange process. The spiral node exchange is used instead of neighbor area reduction reputation of former method. By spiral node exchange, repetition by node exchange process becomes needless and can expect speed up of total processing.**

Index Terms—**first Self-organizing map, feature map, node exchange, fixed neighbor area, spiral exchange.**

I. Introduction

Kohonen's Self Organizing Map (SOM) [1] involves neural networks [2], for which an algorithm learns the feature of input data thorough unsupervised, competitive neighbourhood learning. The SOM is applied in many fields and has been widely studies [3]. Learning efficiency or speed is essential to putting the SOM to practical use, as proposed by some techniques [4-6]. But studies on improvement based on conventional SOM learning algorithms have been left out of main stream.

Based on the conventional SOM learning algorithm, SOM learning is influenced by the sequence of learning data and the initial feature map. We previously reported the influence of the sequence of learning data [7-9], finding that a difference occurred in learning speed due to the sequence of learning data even if all learning data and initial feature map are the same [7], and explained part of this phenomenon using mathematical analysis [8-9].

We reported that the initial feature map also influenced SOM learning efficiency. In initialization, connection weights in a feature map are initialized at random values, but this also sets nodes to random point in the feature map independent of input data space, slowing learning speed or convergence, because this relation is self-organized when convergence is completed. To solve this problem, we proposed initial node exchange using a part of learning data [10-12]. This method cut both the average of move distance of all nodes and the time reduced to complete learning almost in half [10]. And sufficient effect is acquired using few number of learning data in node exchange process [11-12]. In addition, we proposed initial node exchange using a part of initialized nodes connection weights [13], that is handled without the input from the outside. This method is effective by about 5% smaller number of input data for exchange, but peek performance is about 6% inferior, comparing with initial node exchange using learning data.

Since node are exchanged by adding to the conventional SOM learning algorithm, the processing load of node exchange should be minimized. The processing load is related to 3 factors, the amount of input data (factor 1), the initial size of neighbor area (factor 2), and the reduction speed of the neighbor area (factor 3). We clarified factor 1 in earlier research [11-13], and clarified factor 2 in another earlier research [14]. From the results, we found that sufficient effect is acquired, if a total of neighbor areas becomes wider than the feature map.

Through these researches, we got new ideas, fixed size of neighbor area for node exchange, fixed position of winner node and fixed initial size of neighbor area that sets to cover whole feature map at the beginning.

In this paper, we investigate how average move distance of all nodes and average deviation of move distance would change with the differences by type of fixed neighbor area in node exchange process.

Hereafter, Chapter 2 explains SOM, Chapter 3 describes about initial node exchange, Chapter 4 describes experiments, and Chapter 5 describes a conclusion.

II. Self Organizing Map

Kohonen's self-organizing map (SOM) involves of neural networks, that learn the features of input data through unsupervised, competitive neighborhood learning. The SOM is mapping from high to low

dimensional space, usually as a two-dimensional (2D) map. It provides a feature map that arranges similar classes nearer to one another to visualize high-dimensional information in a 2D feature map. Map representation makes it easier to understand relations between data.

The SOM generally has two layers, an input layer and an output layer. Output layer nodes usually form a 2D grid and input layer nodes are fully connected with those at output layer nodes. Each connection has a connection weight, so all output layer nodes have patterns or vectors to be learned.

After learning, each node represents a group of individuals with similar features, the individual corresponding to the same node or to neighboring nodes. That is, the SOM configures output nodes into a topological representation of original data through a process called self-organization (Fig.1).

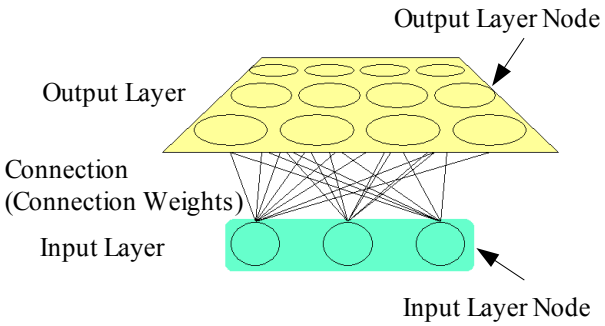


Figure 1. SOM configuration

A. SOM Learning Algorithm

In learning process, when an input pattern or input vector is presented to the input layer as learning data, output layer nodes compete mutually for the right to be declared the winner. The winning node is the output layer node whose incoming connection weights are the closest to the input pattern in Euclidean distance. The connection weights of the winning node and its neighbor nodes are then adjusted, i.e., moved closer toward the input pattern.

As learning process progresses, the learning rate and the size of the neighbor area around the winning node decreases, so in an early stage of learning process, large numbers of output layer nodes are adjusted strongly and, in the final stage, the winning node alone is adjusted weakly.

SOM learning consists of a double loop. Learning data is input sequentially and connection weights are adjusted by the inside loop. The learning rate and neighbor area size are decreased by the outside loop and the inside loop is repeated until learning is completed.

The inside loop learning algorithm is as follows:

1: Select input data from learning data.
2: Present input data to input nodes.
3: Calculate the distance between input data and all output nodes.
4: Sequence output nodes by distance.
5: Set the first order node as the winning node.
6: Adjust node connection weights.
7: Select a neighbor node of the winning node in the feature map.
8: Adjust node connection weights.
9: Repeat 7 to 8 until all neighbor nodes are processed.
10: Repeat 1 to 9 until all data is performed.

In steps 6 and 8 of the inside loop in the SOM learning algorithm, connection weights of the winner and its neighbors are moved closer toward learning data or values of input nodes. Each connection weight is calculated by a equation (1).

$$x_n = x_{n-1} - g(x_{n-1} - y_n) \quad (1)$$

x_n : Connection weight learned by n-th learning data.
y_n : n-th learning data.
g : Learning rate. $(0 < g < 1)$

The outside loop of the learning algorithm is as follows:

1: Initialize all connection weights at random.
2: Initialize the learning rate and neighbor area size.
3: Execute the inside loop.
4: Decrease the learning rate and neighbor area size.
5: Repeat 3 to 4 until learning is completed.

B. Measure of Learning

In the conventional SOM algorithm, the convergence of learning is mainly determined by the following two measures:

1: The number of repetitions exceeds the threshold.
2: The largest distance in all distances between learning data and its winning node becomes smaller than the threshold.

Measure 1 emphasizes learning convergence without considering learning speed. A long time or a large number of repetitions is set as the threshold, because it is the sufficient length for learning convergence. Learning takes the same time or the same repetition limited by the threshold, regardless of learning algorithm performance.

Measure 2 uses the distance between learning data and farthest winning node as a measure of learning. Since attention is paid to the node with the slowest convergence, convergence of the entire feature map or all nodes is not measured.

In addition to these measures, we proposed the average move distance of all nodes (AMD) [10] to consider both learning speed and convergence of the entire feature map. Since AMD becomes a constant if learning converges, this measures the speed of convergence. Because AMD is the average move distance instead of the move distance of a specific node, this measures the entire feature map.

AMD at s-th learning loop $M_{AMD}(s)$ is calculated by equation (2).

$$M_{AMD}(s) = \sum_{k=1}^{s} \sum_{j=1}^{r} \sum_{i=1}^{q} d_e(x_{i,j,k}, x_{i,j-1,k}) \quad (2)$$

$d_e(a,b)$: Euclidean distance.
$x_{i,j,k}$: Output node vector of
i-th location of feature map learned by
j-th learning data in

k-th learning loop.
q : Total number of output nodes.
r : Total number of learning data.
s : Number of learning loops.

III. Initial Node Exchange

A. Basic Concepts

In the initialization process of SOM learning, step 1 in the outside loop in the previous section, all connection weights in the feature map are initialized at random values within the domain of each dimension, so no relation exists between locations of input data space and the feature map.

Unfortunately, learning becomes slow for this missing relation. In the SOM learning algorithm, step 7 of the inside loop, neighbor nodes are selected based on the location on the feature map, although the neighbor node that should be selected is based on the location of input data space. As learning progresses, the relation between input data space and the feature map is self-organized, but the influence of this missing relation is particularly serious in early learning and the distance nodes in input data space move increases.

The initial node exchange concept holds that neighbor nodes in input data space are gathered into neighbor area in the feature map by node exchange, without adjusting connection weights. Nodes in the neighbor area in the feature map are exchanged for nodes in the neighbor area in input data space. This way, the relation between input data space and the feature map is organized based on input data.

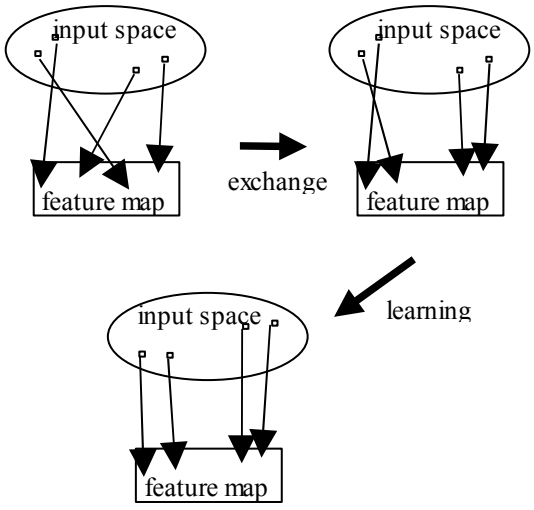


Figure 2. image of node exchange process

Nodes must be exchanged using selected learning data as input data, because many mapping possibilities exist from input data space to the feature map and desirable mapping changes based on learning data. Even when enough input data is chosen at random from learning data, the relation between the location of input data space and that of the feature map must be made by node exchange, without adjusting connection weights (Fig.2).

B. Former Method : initial node exchange

The scheme of initial node exchange process as follows:

1: Select input data.
2: Present input data to input nodes.
3: Calculate the distance between input data and all output nodes.
4: Sequence output nodes by distance.
5: Set the first order node as the winning node.
6: Select a neighbor node of the winning node in the feature map.
7: Select next order node.
8: Exchange their location in the feature map or exchange connection weights.
9: Repeat 6 to 8 until all neighbor node are processed.
10: Repeat 1 to 9 until enough data are performed.

It is clear from this that the initial node exchange process is easy to implement because the inside loop scheme of the learning algorithm resembles the initial node exchange process. The only differences are step 6 and step 8 of the inside loop. In initial node exchange process, step 6 of the inside loop is removed because it involves connection weight adjustment of the winning node, and step 8 of the inside loop is rewritten from connection weight adjustment of neighbor nodes to select neighbor nodes in input data space (step 7 in the exchange) and exchange it for the neighbor node in the feature map (step 8 in the exchange).

The learning algorithm with initial node exchange is easily realized by inserting the node exchange process between feature map initialization and connection weight adjustment in the outside loop of learning.

The outside loop of our proposal is as follows:

1: Initialize all connection weights at random.
e1: Initialize the learning rate and neighbor area size.
e2: Execute initial node exchange.
e3: Decrease the learning rate and neighbor area size.
e4: Repeat e2 to e3 until relation is made.
2: Initialize the learning rate and neighbor area size.
3: Execute the inside loop.
4: Decrease the learning rate and neighbor area size.
5: Repeat 3 to 4 until learning is completed.

Because nodes are exchanged by adding to the conventional SOM learning algorithm, the processing load should be minimized. The processing load depends on the number of exchange nodes, so the number of input data (factor 1), the initial size of the neighbor area (factor 2), and the reduction speed of the neighbor area (factor 3) are influenced by the load.

In early works [11-13], we studied how AMD changed with differences in the number of input data (factor 1) under the conditions that the initial size of the neighbor area and the reduction speed of the neighbor area were fixed to the same value as in learning process. We found that the effect was highest for 5% to 10% of the number of input data in the output node of the feature map, and AMD was shortened to about 70%.

It also studied about 2nd factor by our another work [14]. From this research, We found that sufficient effect

is gained, if a total of initial neighbor areas becomes wider than feature map.

C. Proposed Method 1 : Fixed Size Neighbor Area

Through these researches, we got new ideas, fixed size neighbor area for node exchange, fixed position of winner node and fixed initial size of neighbor area, that initial size of neighbor area sets to fixed value to cover whole feature map, and center node of initial neighbor area is used as winner node. Because, we clarified in earlier research [11-14], that sufficient effect is acquired, if a total of neighbor areas becomes wider than feature map. In former method, node exchange algorithm resembles to learning algorithm because of easy implementation, winner nodes are random point because input data are selected from random point of feature map [13-14], and initial size of neighbor area is about 50% of feature map [13].

The algorithm of former method is modified for proposed method 1. The scheme of initial node exchange process is shown as following:

1: Select center node of fixed neighbor area as input data.
2: Present input data to input nodes.
3: Calculate the distance between input data and all output nodes.
4: Sequence output nodes by distance.
5: Set the first order node as the winning node.
6: Select a neighbor node of the winning node in the feature map.
7: Select next order node.
8: Exchange their location in the feature map or exchange connection weights.
9: Repeat 6 to 8 until all neighbor node are processed.
10: Repeat 1 to 9 until all fixed neighbor area are performed.

The differences are only step 1 and step 10 of former method. Step 1 is rewrote from random node in feature map to center node of one of fixed neighbor area, Step 10 is rewrote from enough data to all fixed neighbor area.

The initialization of neighbor area in the outside loop scheme is also modified from default size to fixed size.

Total number of node exchange L_{fix} is calculated by a equation (3).

$$L_{fix} = d \sum_{i=1}^{\lfloor \frac{m}{2} \rfloor} (2i+1)^2 \qquad (3)$$

d : number of input data
m : initial size of neighbor area
n : size of feature map

D. Proposed Method 2 : Spiral Node Exchange

In learning process, neighbor area reduction are required, because by node adjustment and neighbor area reduction SOM realized self-organization, Mexican hat distribution for each winner node. In contrast, because no node adjustment in node exchange process, a value of a node is not changed. Value of all nodes are fixed, and distance for a winner node is constant from beginning to end. Therefore it is not required neighbor area reduction. The process can exchange nodes so that a node of a near value comes near a winner node according to a fixed value.

In former method, I applied neighbor area reduction reputation to node exchange process because of easy implementation. In proposed method 2, I use spiral node exchange instead of neighbor area reduction reputation. By this method, the repetition by node exchange process becomes needless and can expect speed up of processing.

At the algorithm of the proposed method 2, nodes are exchanged by the order of distance, exchange is earlier so as to be nearer. The process is performed in a spiral toward the outside from the center. So we call this spiral node exchange.

The algorithm of former method is modified for proposed method 2. The scheme of initial node exchange process is shown as following:

1: Select input data.
2: Present input data to input nodes.
3: Calculate the distance between input data and all output nodes.
4: Sequence output nodes by distance.
5: Set the first order node as the winning node.
6: Select a neighbor node of the winning node that is the closest and still not exchanged.
7: Select next order node.
8: Exchange their location in the feature map or exchange connection weights.
9: Repeat 6 to 8 until all neighbor node are processed.
10: Repeat 1 to 9 until enough data are performed.

The differences are only step 6 of former method. Step 6 is rewrote from random node in neighbor area to the node that is the closest and still not exchanged.

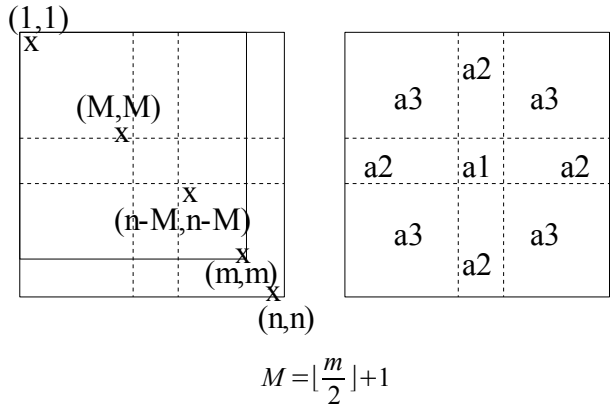


Figure 3. feature map and initial neighbor area

The outside loop scheme is also modified. The outside loop of our proposal is as follows:

1: Initialize all connection weights at random.
e1: Initialize the learning rate and neighbor area size.
e2: Execute initial node exchange.
2: Initialize the learning rate and neighbor area size.
3: Execute the inside loop.
4: Decrease the learning rate and neighbor area size.
5: Repeat 3 to 4 until learning is completed.

The differences are only removing step e3 and e4 because neighbor area reduction reputation becomes needless.

Let's estimate number of node exchange. The neighbor area size is cut off by the edge of the feature map if the winner node locate near a edge. When the neighbor size is smaller than the size of feature map, the feature map divides 3 kind of 9 areas shown in fig.3 because of symmetricalness of the feature map. If the winner node is in "a1" area, the neighbor area is not cut off, then the number of node exchange is m^2 . The area size of "a1" is $(n-2\mathrm{M}(m))^2$ so the possibility that winner node is in "a1" is $(n-2\mathrm{M}(m))^2/n^2$. If winner node is in "a2" area, neighbor area is cut off in one direction, then the number of node exchange is calculated by $m(m-i)$, where $m-i$ is the distance from a edge. The length of $m-i$ part of "a2" is $n-2\mathrm{M}(m)$ so the possibility that winner node is in $m-i$ part of "a2" is $(n-2\mathrm{M})/n^2$. If winner node is in "a3" area, neighbor area is cut off in two directions, then the number of node exchange is calculated by $(m-i)(m-j)$, where $m-i$ and $m-j$ are the distance from each edge. The possibility that winner node is in $m-i$ and $m-j$ point of "a3" is $1/n^2$. So, the expectation of the number of node exchange, when neighbor size is m , is calculated by a formula (4).

in case of $m<n$

$$N(m)=\frac{(n-2\mathrm{M}(m))^2}{n^2}m^2+4\frac{n-2\mathrm{M}(m)}{n^2}\sum_{i=1}^{M(m)}m(m-i)+\frac{4}{n^2}\sum_{j=1}^{M(m)}\sum_{k=1}^{M(m)}(m-j)(m-k) \quad (4)$$

where $M(m)=\lfloor\frac{m}{2}\rfloor+1$

In same way, the expect ion, when the neighbor size is bigger than the size of feature map, is calculated by a formula (5).

in case of $m\geq n$

$$N(m)=\frac{(m-2\mathrm{M})^2}{n^2}n^2+4\frac{m-2\mathrm{M}}{n^2}\sum_{i=1}^{M}n(n-i)+\frac{4}{n^2}\sum_{j=1}^{M}\sum_{k=1}^{M}(n-j)(n-k) \quad (5)$$

where $M=\lfloor\frac{n}{2}\rfloor+1$

Total number of node exchange L_{nex} is calculated by a formula (6).

$$L_{nex}=d\sum_{i=1}^{\lfloor\frac{m}{2}\rfloor}N(2\mathrm{i}+1) \quad (6)$$

d : number of input data
m : initial size of neighbor area
n : size of feature map

Total number of exchange in former method is calculated by a formula (6), and in spiral node exchange, sum in formula (6) becomes needless because neighbor area reduction reputation becomes needless. Using formula (4) and formula (5), total number of exchange in spiral node exchange L_{sp} is calculated by a formula (7).

$$L_{sp}=d\,N(m) \quad (7)$$

d : number of input data
m : initial size of neighbor area

VI. Experiments

An experiments were performed to examine the effectiveness of proposed methods.

A. Proposed Method 1 : Fixed Size Neighbor Area

For former method, initial node exchange with part of initial output nodes as input data for initial node exchange process (indicated by "node"), the initial size of neighbor area is 7x7, and random 10 output nodes as input data are used in initial node exchange.

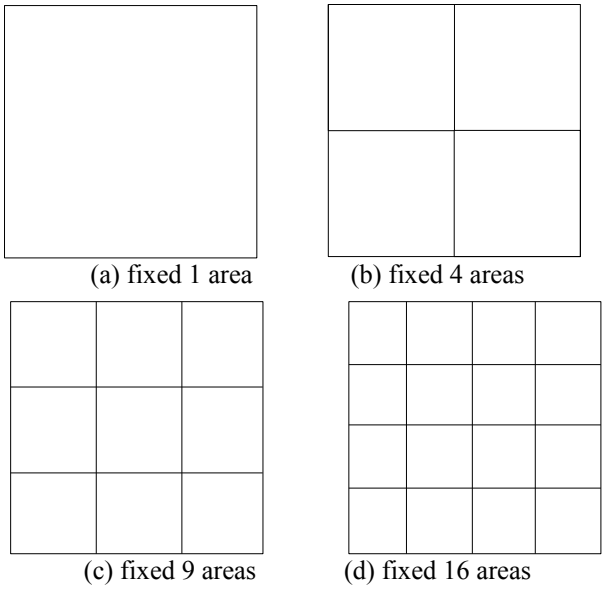


Figure 4. 4 types of initial size and position

To examine a difference about the size of fixed neighbor areas, 4 types of fixed neighbor areas are used in proposed method, fixed 1, 4, 9, and 16 neighbor areas. The examples are shown in Fig.4.

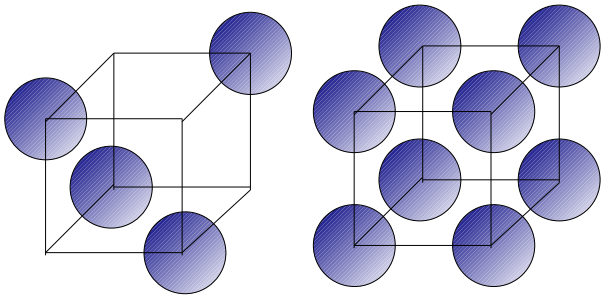
TABLE I. Total Number of Node Exchange

	n	m	L	
node	10	7	830	100%
fixed1	1	10	264	32%
fixed4	4	5	136	16%
fixed9	9	4	285	37%
fixed16	16	3	123	17%

Total number of node exchange using the parameters in the experiment is calculated by equation (3), and the result are shown in Tab.1. From Tab.1, by proposed methods, total number of node exchange is decreased to 16% at the maximum, in comparison with former method.

To examine a difference about the type of leaning data, 3 variation of leaning data are used for all experiments.

Learning data is created randomly into 1, 4, and 8 classes. In 1 class, total 160 data are used. In 4 classes, the center of classes is one of the farthest corner of 3 dimensional cube, and 40 data per class for a total of 160 data are used. The image of learning data distribution is shown in Fig.5(a). In 8 classes, the center of classes is one of the corner of 3 dimensional cube, and 20 data per class for a total 160 of data are used. The image of learning data distribution is shown in Fig.5(b).



(a) 4 classes (b) 8 classes

Figure 5. distribution image of learning data

For experiments we used common parameters, 10x10 of 100 nodes two dimensional feature map, initializing its connection weights at random values, with the learning rate gradually converging at 0.0001 from 0.01 as learning progresses, neighbor size of learning also converges from 7x7 area to single node.

```
. 5 4 7 1 . 4 6 2 .
. 2 . . 4 2 2 7 0 2
2 2 1 2 3 5 5 6 . 4
1 5 . 6 1 5 1 . . .
1 6 5 0 0 6 . 0 . 7
3 4 . 5 . 1 . . 6 2
5 6 1 4 1 . 1 1 6 7
. . 5 . 2 5 5 1 2 .
2 7 6 . . 3 . 4 3 7
5 . . 7 7 . 2 . 3 5
```

(a) initial feature map

```
4 6 6 6 2 6 6 2 6 6      4 4 . 2 2 2 2 6 6 6
2 3 . 7 . 2 . 2 6 .      4 4 4 2 2 2 2 6 6 6
2 3 . . 5 5 0 3 5 .      4 4 4 2 2 2 . 1 1 1
2 7 5 1 . . . 1 7 5      4 4 . 2 2 2 1 1 1 1
2 5 . 5 1 1 1 . . 4      3 3 3 3 3 3 1 1 1 1
2 . 0 5 1 1 1 1 . 2      3 3 3 3 3 3 5 5 5 5
6 7 5 . 1 1 1 . . 5      3 3 3 3 3 3 5 5 5 5
2 4 3 5 5 . . . 0 7      3 3 3 3 7 7 5 5 5 5
4 4 . . . . 7 5 3 .      3 7 7 7 7 7 7 0 0 0
4 2 . 2 7 . 7 4 0 .      7 7 7 7 7 7 7 0 0 0
```

(b) after exchange (c) after learning

Figure 6. typical example of feature map

We conducted experiment for confirming that learning using our proposal is satisfactory.

An example of initial feature map by 8 classes leaning data is shown in fig.6(a). All connection weights are initialized at random values. The number, 0 to 7, shows the node belonging to the class in it, and a different number shows a different class. Fig.6(a) shows that nodes are distributed at random in the initial feature map.

Fig.6(b) and fig.6(c) show results of initializing the feature map in fig.6(a). The feature map after applying the initial node exchange is shown in fig.6(b), which indicates that, even though connection weights were not adjusted, nodes of the same class have gathered together in the initial node exchange. The feature map after applying all proposed method is shown in fig.6(c), which indicates that self-organization of the feature map is completed after our proposal is done.

This trend is seen in all experiment results so we concluded that our proposal enabled learning the same as the conventional SOM learning algorithm.

We conducted experiment for investigate how AMD would change by type of fixed neighbor area and the number of data classes.

Fig.7 shows average of 1000 times of experiments. Horizontal axis shows AMD by the percentage which set conventional SOM to 100. Vertical axis shows the number of data classes. "node" indicates former method, and "fixed1" to "fixed16" indicates the method using each type of neighbor area in initial node exchange process.

Fig.7 indicates that, initial node exchange is effective from the reason that AMD of all methods, including former method, are always smaller than the conventional method. In addition, fixed1 and fixed4 are smaller than former method, and fixed1 is the smallest in every number of data classes. The advantage against conventional method increases when data class, or randomness of learning data, increases. The difference between each method becomes smaller when data class increase with same reason.

From these, we concluded that fixed1 and fixed4 are improved than former method, and fixed1 is the best method in the viewpoint of AMD.

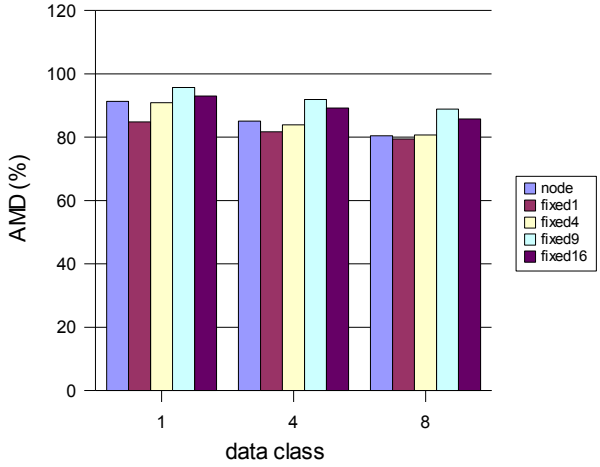


Figure 7. the average of move distance (AMD) for each method

We conducted experiment for investigate how average deviation of move distance would change by type of fixed neighbor area and the number of data classes.

Fig.8 shows average of 1000 times of experiments. Horizontal axis shows Average Deviation of Move Distance by the percentage which set conventional SOM

to 100. Vertical axis shows the number of data classes. "node" indicates former method, and "fixed1" to "fixed16" indicates the method using each type of neighbor area in initial node exchange process.

Fig.8 indicates that, except fixed1, average deviation are always bigger than the conventional method. The average deviation becomes bigger when data class increase. This can be explained form the reason that the difference or the distance between input data make deviation large, only fixed1 has one input data, there is on difference between input data, and others have multi data as input data.

From these, we concluded that in the viewpoint of average deviation, only fixed1 is improved than conventional method, although almost all methods are improved than former method.

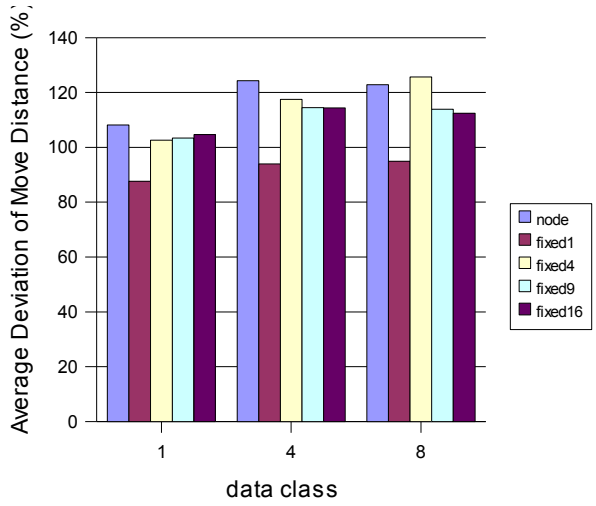


Figure 8. average deviation of move distance for each method

B. *Proposed Method 2 : Spiral Node Exchange*

For experiments we used common parameters, 10x10 of 100 nodes two dimensional feature map, initializing its connection weights at random values, with the learning rate gradually converging at 0.0001 from 0.01 as learning progresses, neighbor size of learning also converges from 7x7 area to single node.

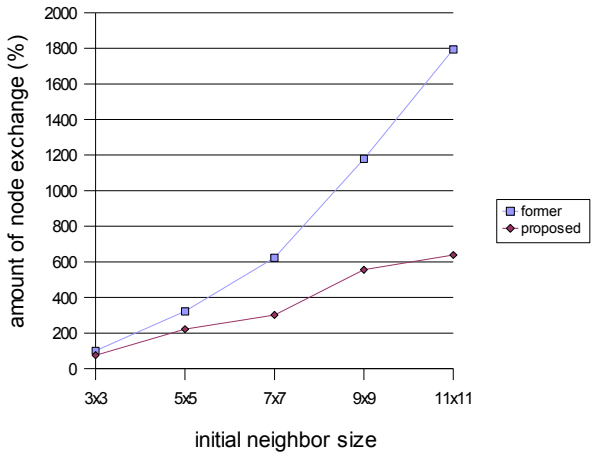


Figure 9. amount of node exchange

Learning data is created randomly into 8 classes. The center of classes is one of the corner of 3 dimensional cube, and 20 data per class for a total 160 of data are used. The image of learning data distribution is shown in Fig.5(b).

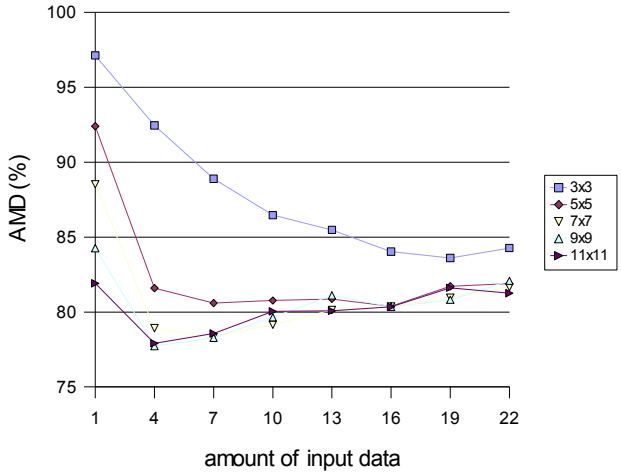


Figure 10. former method

Total number of node exchange using the parameters in the experiment is calculated by formula (6) and formula (7) are shown in Fig.9. Horizontal axis shows the calculation result of total amount of node exchange by the percentage which set 3x3 neighbor size of former method to 100. Vertical axis shows initial size of neighbor area.

Fig.9 indicates that, the proposed method are always small amount of node exchange than the former method, and at "11x11" proposed method is 1/3 times of the former method.

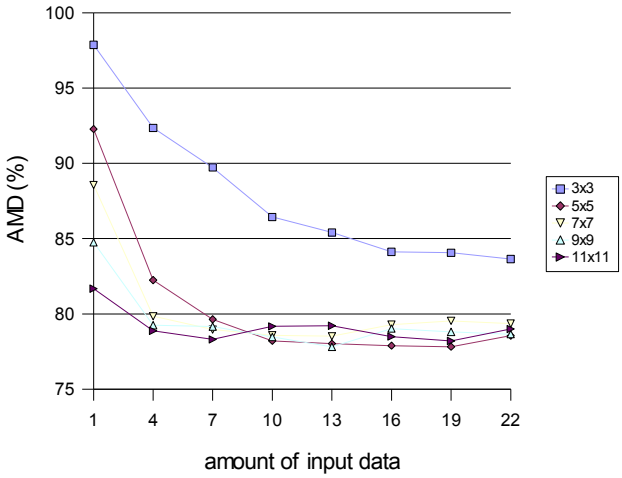


Figure 11. proposed method 2

We conducted experiment for investigate how AMD would change by type of node exchange.

Fig.10 and Fig.11 shows average of 500 times of experiments. Horizontal axis shows AMD by the percentage which set conventional SOM to 100. Vertical axis shows the amount of input data by the percentage which set all node in feature map to 100. "3x3" to

"11x11" indicates initial neighbor size in initial node exchange process.

Fig.10 is the result of former method, Fig.11 is the result of proposed method that uses spiral node exchange.

Fig.10 indicates that, when an initial neighbor size is "7x7" or bigger, the average of move distance of nodes is enough shorter at 4% of data. From this point, as the number of input data increases, AMD increases.

Fig.11 indicates that, when an initial neighbor size is "7x7" or bigger, the average of move distance of nodes is enough shorter (shorter than 80%) at 4% of data. And the line is stable (80% to 77.5%) independently of input data and initial neighbor size.

From these, we concluded that spiral node exchange is improved than former method, in the view point of both AMD and the amount of node exchange. Reduction performance of AMD is better and the amount of node exchange is less.

V. Conclusions

In this paper we proposed two improved methods, Fixed Size Neighbor Area (method 1) and Spiral Node Exchange (method 2).

By method 1, we have studied how AMD and average deviation of move distance changes with differences by type of fixed neighbor area and the number of data classes in node exchange process. As a result of experiments, the most effect is acquired by fixed1 method in the viewpoint of both AMD and average deviation. Using fixed1 method, the initial node exchange process reduced to about 32% in comparison with former method.

By method 2, we have studied how AMD changes with differences by type of node exchange and initial neighbor size in node exchange process. As a result of experiments, spiral node exchange is effective and is improved the performance in the view point of both AMD and the amount of node exchange.

Future work is to combine these two methods and to estimate its efficiency.

References

[1] Teuvo Kohonen, Self-Organizing Maps, Springer Verlag, ISBN 3540586008, 1995.

[2] Robert Heclt-Nielsen, Neurocomputing, Addison-Wesley Pub. Co., ISBN 0-201-09355-3, 1990.

[3] "Bibliography of SOM papers," Draft version 2006-03-27, http://www.cis.hut.fi/mpolla/sombibliography/

[4] Makoto KINOUCHI, KUDO Yoshihiro, "Much faster learning algorithm for batch-learning som and its application to bioinformatics." Proceedings of The Workshop on Self-Organizing Maps (WSOM03), 2003.

[5] Young Pyo Jun, Hyunsoo Yoon, Jung Wan Cho, "L* learning: a fast self-organizing feature map learning algorithm based on incremental ordering." IEICE Transactions on Information and Systems, Vol.E76-D, No.6, pp.698-706, 1993.

[6] Hakaru Tamukou, Keiichi Horio, Takeshi Yamakawa, "Fast learning algorithms for self-organizing map employing rough comparison WTA and its digital hardware implementation." IEICE Transactions on Electronics, Vol.E87-C, No.11, pp.1787-1794, 2004.

[7] Tsutomu Miyoshi, Hidenori Kawai, Hiroshi Masuyama, "Efficient SOM Learning by Data Order Adjustment," Proceedings of 2002 IEEE World Congress on Computational Intelligence (WCCI2002), pp.784-784, 2002.

[8] MIYOSHI Tsutomu, "Order of Learning Data and Convergence of SOM Learning," Proceedings of The 6th International Symposium on Advanced Intelligent Systems (ISIS2005), pp.756-759, 2005.

[9] MIYOSHI Tsutomu, "Learning Data Order and Convergence of SOM Learning," GESTS International Transactions on Computer Science and Engineering, Vol.22, No.1, pp.188-197, 2005.

[10] MIYOSHI Tsutomu, "Node Exchange for Improvement of SOM Learning," Proceedings of 9th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES2005), pp.569-574, 2005.

[11] MIYOSHI Tsutomu, "Initial Node Exchange and Convergence of SOM Learning," Proceedings of The 6th International Symposium on Advanced Intelligent Systems (ISIS2005), pp.316-319, 2005.

[12] MIYOSHI Tsutomu, "Initial Node Exchange Using Learning Data and Convergence of SOM Learning," GESTS International Transactions on Computer Science and Engineering, Vol.21, No.1, pp.216-224, 2005.

[13] MIYOSHI Tsutomu, "Relation Organization of Initial SOM by Node Exchange Using Connection Weights," Proceedings of 2006 IEEE International Conference on Systems, Man, and Cybernetics (SMC2006), pp.1554-1558, 2006.

[14] MIYOSHI Tsutomu, "Influence of Neighbor Size for Initial Node Exchange of SOM Learning," Proceedings of Joint 3rd International Conference on Soft Computing and Intelligent Systems and 7th International Symposium on advanced Intelligent Systems (SCIS&ISIS2006), pp.1117-1121, 2006.

MIYOSHI Tsutomu has currently been an Associate Professor at the Department of Information and Electronics, Graduate School of Engineering, Tottori University, Tottori, Japan since 1997. He was a Senior Researcher at Laboratory for International Fuzzy Engineering Research, Yokohama, Japan, from 1989 to 1993. He was a Researcher at SHARP Corporation, Nara, Japan, from 1985 to 1989 and from 1993 to 1997. He received the Doctor (Engineering) degree from Osaka University, Osaka, Japan, in 1994.

His research interests include Self-Organizing Map, data mining at Internet, and human-machine interaction.