Variable-Density Self-Organizing Map for Incremental Learning

Atsushi Shimada and Rin-ichiro Taniguchi
Department of Intelligent Systems, Kyushu University, Japan
email: {atsushi, rin}@limu.is.kyushu-u.ac.jp

Keywords: Self-Organizing Map, Incremental Learning

Abstract— We propose a new incremental learning method of Self-Organizing Map. Basically, there are three problems in the incremental learning of Self-Organizing Map: 1. depletion of neurons, 2. oblivion of training data previously given, 3. destruction of topological relationship among training samples. Weight-fixed neurons and weight-quasi-fixed neurons are very effective for the second problem. However the other problems still remain. Therefore, we improve the incremental learning method with weight-fixed neurons and weight-quasi-fixed neurons. We solve the problems by introducing a mechanism to increase the number of neurons effectively in the incremental learning process.

1 Introduction

Self-Organizing Map (SOM) proposed by Kohonen[1, 2] is one of the most widely used artificial neural network algorithms that uses unsupervised learning. There are two types of SOM learning algorithms. One is batch learning and the other is incremental learning. Generally, the batch learning algorithm is usually used for SOM's learning and well studied. However, the incremental learning algorithm should be used when learning data are input into SOM sequentially. In the incremental leaning of SOM, it is very difficult to determine the number of neurons in the map since the number of training samples is usually unknown. In addition, a new training sample has much effect on the neurons' weights of SOM, and, therefore, the map tends to forget training data previously given. Moreover, the feature map can not always keep the topological relation among training data. As just described, there are several problems in the incremental learning of SOM. In this paper, we analyze characteristics of previous approaches for the incremental learning of SOM and propose a new approach to achieve simple and effective incremental learning.

2 Incremental learning of SOM

Basically, there are three problems in incremental learning of SOM.

(1) the number of neurons becomes insufficient
(2) training data previously given is forgotten
(3) topology among training data is destroyed

The first problem arises when the number of training samples comes close to or exceeds the number of neurons. It is necessary to increase the number of neurons in the learning process since it is very difficult to determine the number of neurons in advance. The second problem means that the map tends to forget training data previously given since a new training sample has much effect on the neurons' weights of SOM. The third problem is caused when the number of neurons is insufficient. In such situation the feature map can not always keep the topological relation among training data.

Michihata et al. proposed an incremental learning method based on abnormal states of SOM[3, 4]. They proposed three types of indicators to detect the abnormal states of SOM.

1 There are many neurons whose weights are much different from a new training sample.
2 There are many neurons which do not become activated at all.
3 There are neurons which are repeatedly selected as the winner.

When an abnormal state is detected by these indicators, all of the training data are relearned without initializing the weight of neurons. In this method, the neurons after incremental learning forget the previous training data since the winner neurons which become activated for the training data are changed by the incremental learning.

Shimasaki et al. proposed two types of incremental learning methods[5]. One is "read type" incremental learning. In this method, a SOM accepts not only newly added training samples, but also accepts, as training data, the weights of neurons which are acquired based on the previous training samples. The other is "storage type" incremental learning method. In this method, the averages of the training samples which are mapped to labeled neurons. However, these methods are not essentially incremantal learning but relearning.

Yamada et al. proposed a incremental learning method with weight-fixed neurons and weight-quasi-fixed neurons[6]. The weight-fixed neurons are not updated in the learning process. Therefore, the map does not forget the previous training data. The weight-fixed neurons are not updated easily in the learning process. This is effective for spreading the winner neurons throughout the map.

We summarized previous approaches for incremental learning of SOM in TABLE 1. There is no approach which

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

solve all the three problems. Therefore, a new incremental learning method should be developed.

Table 1: Comparison between traditional approaches.

	Problem 1	Problem 2	Problem 3
Anomaly	▲	–	◎
Read	▲	–	◎
Storage	▲	–	◎
weight-fixed	–	◎	–

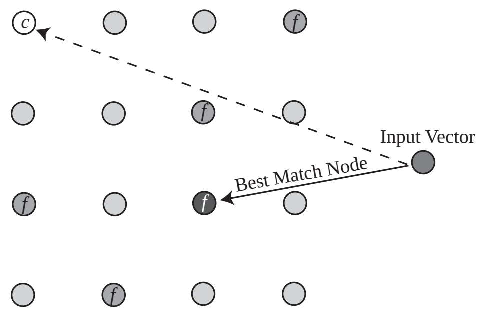

Figure 1: Example of topology destroyed problem

3 Variable-density SOM

A learning method with weight-fixed and weight-quasi-fixed neurons is very effective for the incremental learning of SOM since it does not forget the previous training data. Therefore, the incremental learning of SOM is achieved if we solve the other problems (problem 1 and problem 3). In the following subsections, first of all, we overview the learning algorithm with weight-fixed and weight-quasi-fixed neurons. Then, we propose a new incremental learning algorithm with a variable-density SOM.

3.1 Weight fixed neurons and weight-quasi-fixed neurons

In this subsection, we explain the incremental learning algorithm with weight-fixed and weight-quasi-fixed neurons.

Step 1 Initializing the weights of neurons randomly.

Step 2 A neuron c that satisfies equation (1) is selected as the winner neuron and the distance between the input vector $\boldsymbol{I}$.

$$||\boldsymbol{I} - \boldsymbol{W}_c|| = \min_u(||\boldsymbol{I} - \boldsymbol{W}_u||) \tag{1}$$

Step 3 Weight vectors are updated by using equation (2). However, the weight fixed neuron f is not updated.

$$\begin{aligned}\boldsymbol{W}_u(t+1) = \\ \boldsymbol{W}_u(t) + H(d)\alpha(t)h_{c,u}(\boldsymbol{I}(t) - \boldsymbol{W}_u(t))\end{aligned} \tag{2}$$

$h_{c,u}$ is a neighborhood kernel defined by equation (3).

$$h_{cu} = \exp\left(-\frac{||x_c - x_u||^2}{2\sigma^2(t)}\right) \tag{3}$$

$\alpha(t)$ and $\sigma(t)$ are monotonically decreasing functions calculated as follows.

$$\alpha(t) = \frac{-\alpha_0(t-T)}{T} \tag{4}$$

$$\sigma(t) = \sigma_i\left(\frac{\sigma_f}{\sigma_i}\right)^{t/T} \tag{5}$$

T is the limit of iterations, α_0 is the initial value of training rate, σ_i is the initial value of $\sigma(t)$ and σ_f is the last value of $\sigma(t)$. $H(d)$ is the function which defines the distribution weight-quasi-fixed neurons.

$$H(d) = \frac{1 - \exp(-d \cdot k)}{1 + \exp(-d \cdot k)} \tag{6}$$

d is the distance between a neuron u and a weight-fixed neuron nearest to the neuron u. k is a coefficient to determine the slope of the function $H(d)$. This function is about 0 in the neighborhood of already established weight-fixed neurons, and, therefore, the weights of neurons near to the neuron f are not updated easily. On the other hand, if the value of d is large, the function $H(d)$ will be about 1. Therefore, the weights of neurons are updated in a similar way to the case of traditional learning. This function enables SOM to disperse weight-fixed neurons all over the map.

Step 4 The same training sample is learned by Step 2 and Step 3 iteratively until satisfying the weight fixed condition; $U < d_f$, where U is the distance between the input vector and the weight of the winner neuron and d_f is a threshold.

Step 5 If one of the neurons satisfies the weight fix condition, a new training sample is learned.

The SOM does not forget the previous training data since the weight-fixed neuron f is no longer updated in Step 3. However, the problems that the number of neuron becomes insufficient in the process of the incremental learning and that topology among training data is destroyed still remain. The topology destruction problem is explained with Fig. 1. We consider that the winner neuron for the current training sample is neuron f. In this case, the weights of the neuron f are not updated and the weights of neurons around the neuron f are not updated easily. Therefore, the winner neuron c is selected relatively far from the neuron f. This is how the topological relation is destroyed.

3.2 Proposed learning algorithm

In this subsection, we propose a new incremental learning algorithm with weight-fixed neurons and weight-quasi-

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

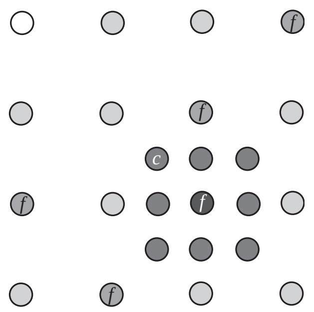

Figure 2: Variable-density SOM

Figure 3: Color data used for training

fixed neurons. We assume two dimensional SOM for simplicity, but proposed algorithm can be extended to higher dimensional SOM. Our approach can increase the number of neurons and put them into appropriate positions of the current SOM. Following Step 2-A and Step 2-B are inserted between Step 2 and Step 3 mentioned above.

Step 2-A If the weight of the winner neuron c is fixed, eight neurons are inserted around the neuron c.

Step 2-B The initial weights of the new eight neurons are calculated as follow. If the neuron f which is nearest from the neuron n is the neuron c,

$$\boldsymbol{W}_n = h_{c,n}\boldsymbol{W}_c \tag{7}$$

otherwise,

$$\boldsymbol{W}_n = \frac{h_{c,n}\boldsymbol{W}_c + h_{f,n}\boldsymbol{W}_f}{h_{c,n} + h_{f,n}} \tag{8}$$

Our approach adds neurons in a small area of the map, and, therefore, a variably densed map, i.e., variable-density map, described in Fig. 2 is generated. The insufficiency of the number of neuron is solved since the number of neurons is increased adaptively in the process of the incremental learning. Moreover, the problem of topology destruction is also solved since the added neurons have similar weights to those of the neuron f.

4 Experimental results

We made a couple of experiments to investigate the effectiveness of our approach.

4.1 Condition

We established an initial map with the size of 8×8 (64 neurons) and configured each parameter as follows; $\alpha_0 = 0.3, \sigma_i = 1.5, \sigma_f = 0.3, k = 0.6, T = 1000$. We made a comparison among three learning methods: the batch learning method (method 1), the incremental learning method proposed by Yamada (method 2), and our proposed method (method 3).

Figure 4: Topological map after learning by method 1

4.2 Result 1

The training samples were 20 kinds of color values, each of which is represented in an (R, G, B) color vector (see Fig. 3). The sequence of training was from Black, Red, ... to Beige. Fig. 4 shows the learning result with the batch learning, and each neuron is colored by a color value, to which the neuron is mapped. Neurons which are mapped to the training data homogeneously spread throughout the map. We can see smooth color gradation among colored neurons. On the other hand, Fig. 5 shows the learning result with weight-fixed neurons and weight-quasi-fixed neurons, and the color gradation is not smooth. This shows that this learning method can not keep topological relation, or color similarity, among training data. Finally, Fig. 6 shows the learning result with the variable-density SOM. Compared with Fig. 5, the color gradation is definetely smooth. This shows our approach can achieve the incremental learning without destroying the topological relationship.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

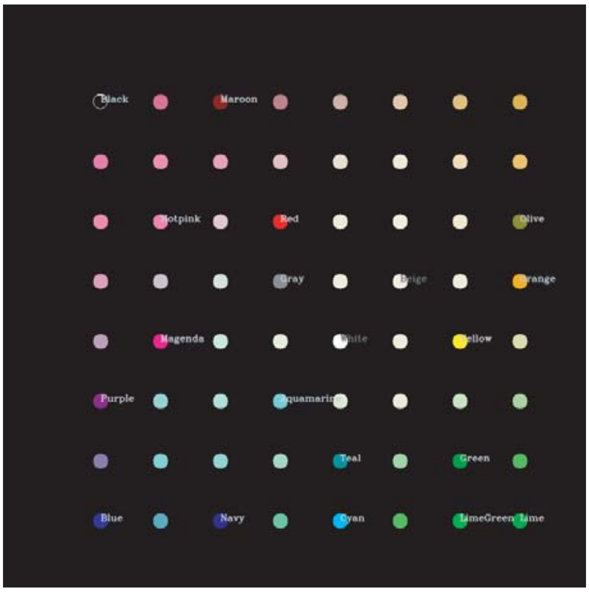

Figure 5: Topological map after learning by method 2

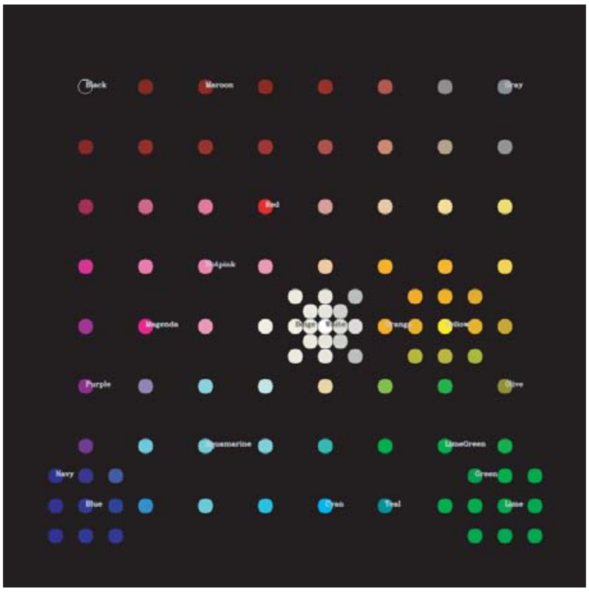

Figure 6: Topological map after learning by method 3

4.3 Result 2

The input data were a set of images consisting of 26 kinds of alphabets shown in Fig. 7. The size of images was 7×7, and each pixel had a binary value. The sequence of training was in alphabetical order, i.e., from A to Z.

Fig. 8 shows the learning result with the batch learning. We put each alphabet label on its corresponding neuron. There are several cases in which two different alphabets are mapped to one neuron: "E and F", "H and U", "M and N", "Q and V". This is a sign that the number of neurons was insufficient. We enlarged the size of SOM so that each alphabet was assigned to a different neuron. When the map size became 10×10, each alphabet was assigned to a different neuron. Fig. 9 shows this result.

Fig. 10 shows the learning result with weight-fixed and weight-quasi-fixed neurons. Neurons being mapped to an alphabet are unevenly distributed in the lower right part of the map. In addition, the alphabets "C", "G" and "O", whose shapes are visually similar, are respectively mapped to neurons which are far from one another in Fig. 10. On the contrary, they are mapped to neurons which are relatively near from one another in the batch learning (see Fig. 9). It shows that the incremental learning method with weight-fixed and weight-quasi-fixed neurons can not keep the topological relation of training data.

Fig. 11 shows the learning result with the variable-density SOM. The number of neurons after learning was 104. This is almost the same as the number of neurons in Fig. 9. This means that the number of neurons inserted into the map seems to be sound. Also, the alphabets "C", "G" and "O" were mapped to neurons which were near from one another. Therefore, we can say that our approach can keep the topological relationship. Fig. 12 shows the result with the variable-density SOM when the SOM learned training data in a random order. In this case, again, neurons labeled as "C", "G" and "O" were mapped nearby in the map.

In the next experiment, we compared time spent in the incremental learning with ones in other methods. The result is shown in Fig. 13. The red colored point shows the learning time required in the batch learning (method 1). When the training samples were learned by weight-fixed and weight-quasi-fixed neurons (method 2) and by the variable-density SOM (method 3), each plotted point shows accumulative learning time. Also, accumulative learning time when the SOM relearned all the previously given alphabets at each time. (we call it "sequential learning") is shown for comparison. We can see that it takes much longer time to learn the training data with the sequential learning. Compared with this, the incremental learning can finish the learning in shorter time. Moreover, learning time with the variable-density SOM is shorter than the time with weight-fixed and weight-quasi-fixed neurons though the number of neurons with the variable-density SOM is larger than the one with weight-fixed and weight-quasi-fixed neurons. This reason can be explained in Fig. 14, which shows the number of iterations required to satisfy the weight fix condition. The number of iterations with the variable-density SOM is smaller than the one with weight-fixed and weight-quasi-fixed neurons. This is because the inserted neurons had similar weights with the weight-fixed neuron.

Finally, we examined how many neurons are mapped to the training samples. Fig. 15 shows spatial distribution of neurons mapped to each alphabet. The distribution value is 0 if a training sample is mapped to only one neuron. If a training sample is mapped to more than 1 neurons, the

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

Figure 7: Alphabet images used for training

Figure 8: Topological map after learning by method 1

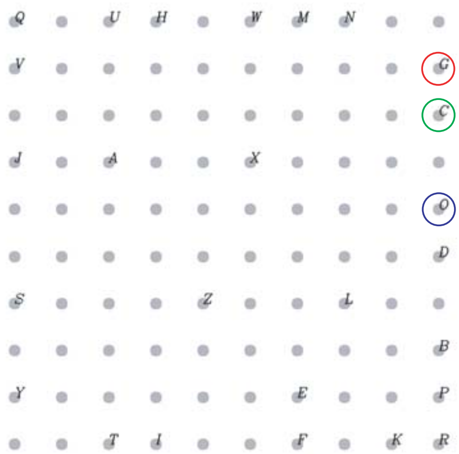

Figure 9: Topological map after learning by method 1 (the number of neurons: 100)

Figure 10: Topological map after learning by method 2

distribution value is calculated by the neurons' 2-D coordinates, or neurons' positions, in the map. In the method with weight-fixed and weight-quasi-fixed neurons, neurons mapped to training samples given in the latter stage such as "Z" had the larger distribution and most of training samples given in the early stage were mapped to only one neuron. This is because the latter training samples have to be mapped to non weight-fixed neurons, which are spatially dispersed in a latter stage. Compared with this, in the variable-density SOM, the training samples are remembered in some neurons even if the data are learned in the early stage. Insertion of additional neurons makes the number of iterations in learning phase smaller, and, thus, the weights of neurons are not modified too much.

5 Conclusion

In this paper, we have proposed a new incremental learning method of Self-Organizing Map. There are three problems in the incremental learning of Self-Organizing Map: 1. depletion of neurons, 2. oblivion of training data previously given, 3. destruction of topological relationship among training samples. Our approach can solve these problems. Moreover, the training time is shorter than the traditional incremental learning method. From several experiments, we have found that our approach is simple and effective for the incremental learning of SOM.

References

[1] T. Kohonen. "Self-Organization and Associative Memory". *Springer-Verlag*, 1989.

[2] T. Kohonen. "Self-Organizing Maps". *Springer Series in Information Science*, 1995.

[3] Masahiro Mitihata, Tsutomu Miyoshi, Hiroshi Masuyama. "A Consideration on the labels of Self-Organizing Map after refresh learning". The 16th Fuzzy System Symposium TC2-1, pp. 383–386, 2000.

[4] Masahiro Mitihata, Tsutomu Miyoshi, Hiroshi Masuyama. "Automatic Incremental Learning System on Self-Organizing Map". Journal of Japan Society for Fuzzy Theory and Systems, Vol. 14, No. 2, pp. 147–156, 2002.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

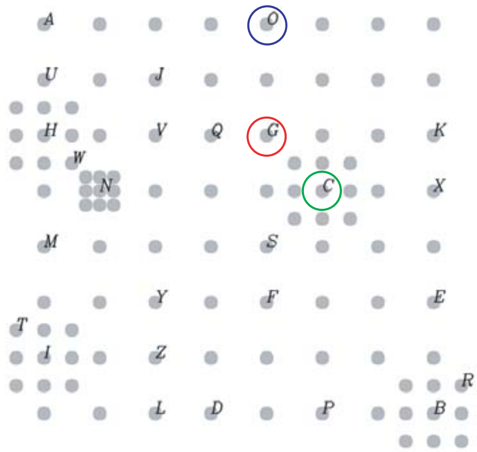

Figure 11: Topological map after learning by method 3

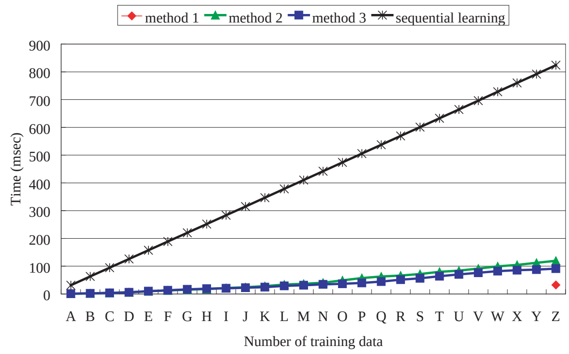

Figure 13: Learning time

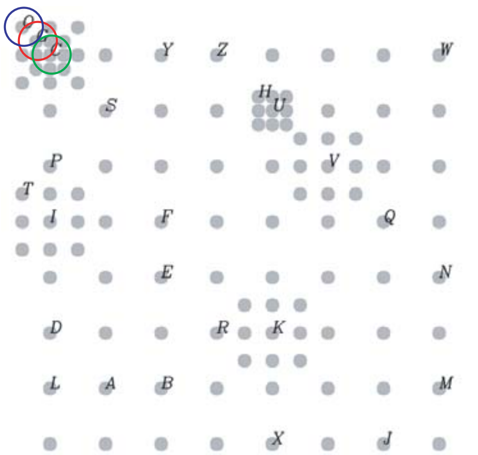

Figure 12: Topological map after learning by method 3 (random order)

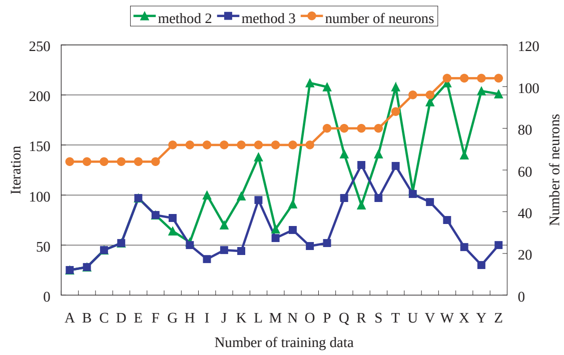

Figure 14: Iteration until satisfying the weight fixed condition

[5] Naohumi Shimasaki, Hiroto Segi, Yasaburo Matuura, Makoto Ohki and Masaaki Ohkita. "A proposal and Examination od Additional Learning Method For a Torus Typed Self-Organizing Map". The seventh SOM-Meeting, 2006.

[6] Takeo Yamada, Motonobu Hattori, Masayuki Morisawa and Hiroshi Ito. "Sequential Learning for Associative Memory using Kohonen Feature Map". Neuro Computing 98-90, pp. 93–100, 1999.

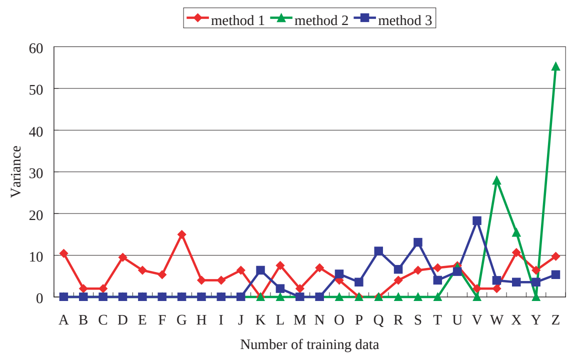

Figure 15: Distribution of neurons which remember training data

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de