Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Image and Signal Processing (CIISP 2007)

Life-long Semi-supervised Learning: Continuation of Both Learning and Recognition

Youki Kamiya and Toshiaki Ishii
Department of Computational Intelligence and Systems Science,
Interdisciplinary Graduate School of Science and Engineering,
Tokyo Institute of Technology
Kanagawa, 226–8503, JAPAN
Email: {youki, ishikoro}@isl.titech.ac.jp

Osamu Hasegawa
Imaging Science and Engineering Laboratory,
Tokyo Institute of Technology
Phone: +81-45-924-5180
Fax: +81-45-924-5715
Email: hasegawa@isl.titech.ac.jp

***Abstract*— This paper presents a new method for continuous and incremental learning and recognition based on self-organized incremental neural networks. It is available in the fluctuating environment where the number of recognition classes cannot be defined. In this method, the learning process and recognition process are not separated. This method can acquire concept when multiple feature vectors of new input object come, and then can recognize it using previously acquired concept. We experiment an examples of life-long semi-supervised learning tasks in real world. In the result, the proposed method was able to learn and recognize 104 objects incrementally, non-stop, and in real time.**

I. Introduction

Incremental learning techniques were presented to learn new information from new input data. ARTMAP [1] is a popular method of incremental learning, but it is sensitive to selection of parameters, order of training data and amount of noise. Other techniques based on insertion criterion were proposed: Growing Neural Gas (GNG) [2], Dynamic Cell Structures (DCS) [3], Life-long Learning Cell Structures (LLCS) [4] and Self-Organized Incremental Neural Networks (SOINN) [5]. These techniques try to tackle main difficulty in incremental learning, Stability-Plasticity Dilemma [6]. This is how to learn new knowledge without forgetting previous knowledge.

Labeled instances are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Semi-supervised learning addresses this problem by using large amount of unlabeled data, together with the labeled data, to build better classifiers. Because semi-supervised learning requires less human effort and gives higher accuracy, it is of great interest both in theory and in practice. Many types of method were proposed: EM with generative mixture models [7], self-training [8], co-training [9], Transductive Support Vector Machines (TSVMs) [10] and graph-based methods [11] [12]. However, semi-supervised learning methods make strong model assumptions. Ideally one should use a method whose assumptions fit the problem structure. This may be difficult in reality [13].

Life-long learning, also termed continuous learning, emphasizes learning through the entire lifespan of a system. In life-long learning, system learns open data set influenced by noise and other factors. And system preserves old learned knowledge in changing environment if it does not contradict the current task. Surely labeled data rarely come among enormous quantity of input data. Thus, life-long learning includes both issues of incremental learning and semi-supervised learning. However, above mentioned methods consider either one issue and no method exists yet for such learning task.

In this paper, we propose a novel method for life-long semi-supervised learning. This method can learn new patterns easily without destroying old learned information. On account of nonstop learning, it can continue to recognize input patterns concurrently with learning them. And it can adjust old learned information based on scarce labeled data.

II. Proposed Method

A. Overview of the Proposed Method

The targets of the proposed algorithm are: (1) Without prior conditions such as the number of nodes or a good network structure to conduct life-long semi-supervised learning, (2) To learn new information without destroying old learned information, (3) To continue to recognize input patterns concurrently with learning, (4) To continuously adjust old learned information based on scarce labeled data and thereby decrease misrecognition.

Figure 1 shows overview of the proposed method. This method consists of two layers: a data analysis layer and a concept acquisition layer. The data analysis layer is based on neural networks to represent data structure. We adopt SOINN [5] to discriminate input data and eliminate noise. This layer has multiple SOINNs which deal with different type of feature vectors independently.

In this paper, we define *concept* and *signal*. An input pattern is converted a *signal* S that is a combination of reacted cluster number cl_i in each SOINN; $S = (cl_1, cl_2, \cdots, cl_n)$. And a *concept* c maintain the signal S that is a combination of cluster number cl_i of relevant node a_i in each SOINN, i.e., $S_c = \{cl_1(a_1), cl_2(a_2), \cdots, cl_n(a_n)\}$.

The concept acquisition layer makes new concept incrementally when the system judges that an input signal belongs to none of old created concepts. Concurrently with that, the system outputs the recognition result "unknown". If another input signal is judged as belonging to one of previous concepts, the system outputs any of following two recognition results.

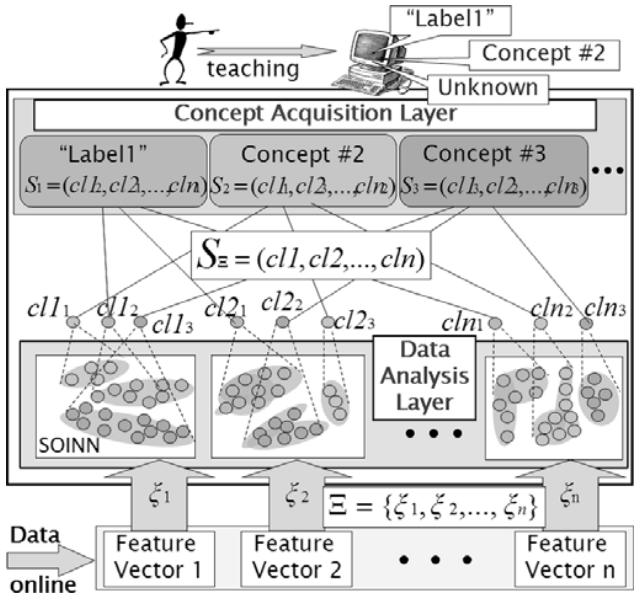

Fig. 1. Overview of the proposed method

If the concept has a label, the system outputs the label. If not, the system outputs the result that the input pattern is related to the concept. Thus, the system outputs any of three types of recognition result at the same time as learning each input pattern.

When the system gets a teaching label from outside teacher such as human, the system compares the teaching label with a label of the concept reacted to the input pattern. If these labels are same, the system only gives reacted node in each SOINN the label. If not, it remediates the clusters of each SOINN based on the information of labeled nodes.

B. Complete Algorithm

Along with the analysis of Section 2.1, we present the complete algorithm here. In this paper, we describe the algorithm if set the number of SOINN in data analysis layer to m.

Notations to be used in the algorithm

W_i	n-dimension weight vector of node i.
A_i	Node set in i-th SOINN, used to store nodes.
N_i	Set of direct topological neighbors of node i. If a node links with node i by an edge, we say the node is the neighbor of node i.
M_i	Local accumulated number of signals of node i.
E_i	Edge set (or connection set) in i-th SOINN, used to store connections between nodes.
$age_{(i,j)}$	Age of the edge between node i and node j.
C	Concept set, used to store concepts in concept acquisition layer.
$cl_i(j)$	Cluster number of node j in i-th SOINN.
S_i	Signal of concept i. Each concept has one signal, which is the combination of reactive cluster numbers of each data analysis layer. I.e. $S_i = (cl_1, cl_2, \cdots, cl_m)$.
L_i	Label corresponding to node or concept i. While concept does not have a label, this value means cluster number.
T_i	Similarity threshold. If the distance between an input pattern and node i is greater than T_i, the input pattern is a new node.

Procedure on online semi-supervised learning

0) Initialize node set A_i to contain two nodes, a_1 and a_2 with weight vectors selected randomly. Initialize connection set E_i, $E_i \subset A_i \times A_i$, to the empty set. Also initialize concept set C to the empty set.

(A) *Data Analysis Layer* : From step 1 to step 10 is the process in data analysis layer. Input pattern $\Xi = (\xi_1, \xi_2, \cdots, \xi_m)$ is separated into m parts and each part is processed equally in m SOINNs $(i = 1, 2, \cdots, m)$.

1) Input new pattern $\xi_i \in R^{n_i}$ $(\xi_i \in \Xi)$. The dimension of the input pattern is individually defined in each SOINN.
2) Search the nearest node (winner) $s1_i$, and the second-nearest node (second winner) $s2_i$ by

$$s1_i = \arg \min_{a \in A_i} \|\xi - W_a\| \quad (1)$$

$$s2_i = \arg \min_{a \in A_i \setminus \{s1\}} \|\xi - W_a\| \quad (2)$$

If the distance between ξ_i and $s1_i$ or $s2_i$ is greater than similarity threshold T_{s1_i} or T_{s2_i}, the input signal is a new node; add it to A_i and go to Step9 to discriminate the input signal. The similarity threshold T_j is;

$$T_j = \begin{cases} \max_{c \in N_j} \| W_j - W_c \| & (|N_j| \neq 0) \\ \min_{c \in A_i \setminus \{j\}} \| W_j - W_c \| & (|N_j| = 0) \end{cases} \quad (3)$$

3) If an edge between $s1_i$ and $s2_i$ does not exist, create it. Set the age of the edge between $s1_i$ and $s2_i$ to zero.
4) Increment the age of all edges emanating from $s1_i$ by 1.
5) Add 1 to the local accumulated number of signals M_{s1_i}.
6) Adapt the weight vectors of the winner and its direct topological neighbors by fraction ϵ_1 and ϵ_2 of the total distance to the input signal,

$$\Delta W_{s1_i} = \epsilon_1(\xi_i - W_{s1_i}) \quad (4)$$

$$\Delta W_a = \epsilon_2(\xi_i - W_a) \quad (\forall a \in N_{s1_i}) \quad (5)$$

We adopt a scheme to adapt the learning rate over time by $\epsilon_1 = 1/M_{s1_i}$ and $\epsilon_2 = 1/100M_{s1_i}$.
7) Remove edges with an age greater than a predefined threshold age_{dead}. If this creates nodes having no more emanating edges, remove them as well.
8) If the number of input signals generated up to that time is an integer multiple of parameter λ, delete nodes caused by noise as follows: for all nodes in A_i, search for nodes having no neighbor or only one neighbor, then remove them.
I.e., if $|N_a| = 1 or 0$, then $A_i = A_i \setminus \{a\} (\forall a \in A_i)$.
If node or edge is added or deleted in past steps, go to step 9. If not, go to step 10.
9) Cluster all nodes in A_i as the following procedure. Here, we define that the path between node a and node b exists

if we can follow from node a to node b along some edges.

a) Initialize all nodes to be unclassified. $k = 0$..
b) Choose certain node u which previous cluster number is k among unclassified nodes. If previous cluster number of any unclassified nodes is not k, choose certain node $u = \min_{a \in A_i} cl_i^{old}(a)$.
c) Set new cluster number of node u to k in order to discriminate the node, i.e. $cl_i(u) = k$.
d) Search nodes that has the path between node u. Set new cluster number of the nodes to the same cluster number as node u in order to discriminate them.
e) If there are still some unclassified nodes, increment k by 1 and go back to (a). If not, go to step 10.

10) If some nodes with different labels exist in cluster D, divide the cluster D as the following procedure.
a) Initialize all nodes in the cluster D to be unclassified and $k = 0$. Initialize terminal node set B_{end} to the emptyset and search node set B_k as follows;

$$B_k = B_0 = \{a | \forall a (a \in D, L_a \neq \emptyset)\} \quad (6)$$

Change cluster numbers of node with label into new numbers which are different from each other. I.e. $cl_i(u) = j$, $cl_i(u) \neq cl_i(v)$ $[j \neq cl^i(a) \ \ (\forall a \in A_i \setminus B_0), \{u, v\} \subseteq B_0]$.
b) Search unclassified nodes that are neighbors of node u in B_k. Set these nodes to B_{k+1}.

$$B_{k+1} = \{a | \forall a \in D, a \in N_u \ \ (\forall u \in B_k)\} \quad (7)$$

Set new cluster number of all nodes in B_{k+1} to that of node u.
c) Remove nodes from B_{k+1}, whose local accumulated number is less than that of all its neighbors.
d) Increment k by 1. If B_k is not the empty set, go back to (b). If not, $B_k = B_{end}$.
e) Search unclassified nodes which are neighbors of the nodes in B_k. Set new cluster number of them to that of neighbor node in B_k. And add them to B_k.
f) If there are still some unclassified nodes, go back to (e). If not, go to step 11.

(B) *Concept Acquisition Layer* : From step 11 to step 15 is the process in concept acquisition layer. Acquire and adapt information of concept based on information from m SOINNs against input signal ($i = 1, 2, \cdots, m$).

11) Based on change of cluster in data analysis layer, adapt signals of concepts and remove concepts if necessary.
a) Initialize i and j to zero.
b) Choose certain node u ($\in A_i$) which satisfy following equations: $cl_i^{old}(u) = j$, $cl_i^{new}(u) \neq j$. If any nodes cannot satisfy, go to (d).
c) Change elements of signals which satisfy equations, $cl_i^{old}(u) = cl_i \quad (cl_i \in S_c, c \in C)$, into $cl_i = cl_i^{new}(u)$.

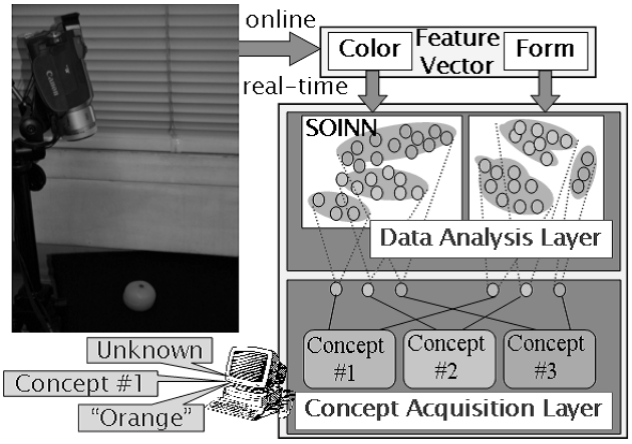

Fig. 2. Environment of our experiment

d) If $j < cl_i^{old}(a) \ \ (\forall a \in A_i)$, then increment j by 1 and go back to (b).
e) If i is larger than m, then increment i by 1 and go back to (b).

In the result of this step, if at least one winner $s1_i$ has been deleted, go to step 1. If not, go to step 12.

12) Set signal S against the input pattern Ξ to;

$$S_\Xi = (cl_1(s1_1), cl_2(s1_2), \cdots, cl_m(s1_m)) \quad (8)$$

13) Search the concept γ which has the same signal as that of S_Ξ. If any concepts is not relevant, add new concept to concept set C. I.e. If $S_\Xi = S_c \ \ (\exists c \in C)$, then $\gamma = c$, else

$$C = C \cup \{\gamma\}, \ S_\gamma = S_\Xi, \ L_\gamma = |C| \quad (9)$$

14) If system has got teaching label L_t, set label of γ to L_t. Also set that of the winner in each SOINN to L_t.

$$L_\gamma = L_t, \quad L_{s1_i} = L_t \ \ (1 \leq i \leq m) \quad (10)$$

15) Output information of γ against input pattern Ξ. I.e., output a label if γ has, and output concept number if not. Go to step 1 to continue the life-long semi-supervised learning and recognition.

III. Experiment

A. Outline of the Experiment

We simulate life-long semi-supervised learning tasks to verify the availability of the proposed method. Figure 2 shows the situation we set for our experiment. A fixed camera is set up in doors, we put objects on the black sheet sequentially. An object image is extracted by background subtraction, and two different types of feature vectors are extracted from the image, frame by frame. A pair of extracted vectors is used as an input signal, and the system learns and recognizes it. We teach labels of each object several times, and the system adjusts the previous learned knowledge based on them. This situation requires the method which can learn incrementally, semi-supervised, continuously, and in real time.

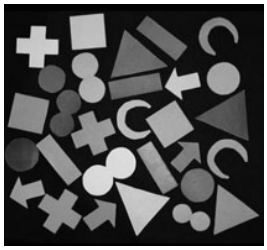

Fig. 3. Examples of artificial objects

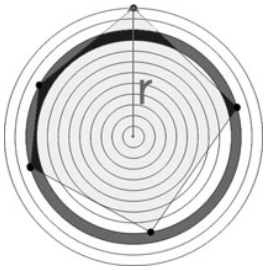

Fig. 4. Shape feature evaluation method: In this case, $\mathbf{Shape}[10] = S1/(S1 + S2)$, where S1 denotes the black area and S2 the area of the dark gray part

In this paper, we use artificial objects shown in Figure 3. Color and shape features are adopted as two different types of input vectors. The data analysis layer includes two SOINNs. For the color feature, we evaluate average values of RGB intensity in pixels of the object image (three dimensions). And the shape feature is evaluated by following procedure (ten dimensions);

1) Find the furthest point from the center of the object; let r denote their distance.
2) Divide r by 12 to obtain 12 concentric circles as Figure 4 illustrates. Their respective radii are $\frac{i \times r}{12} (1 \leq i \leq 12)$.
3) Sequentially from the center, calculate the respective areas between the two concentric circles, denoted as S_j. In this figure, $S_j \longleftarrow$ the total area of the black part and the dark gray part.
4) Evaluate the area T_j that the object occupies in S_j, i.e. $T_j \longleftarrow$ the area of the black part in the figure.
5) $\mathbf{Shape}[j] \longleftarrow \frac{T_{j+2}}{S_{j+2}} (1 \leq j \leq 10)$.

We did not use T_i and S_i for $i = 1, 2$ because we assume that the shape feature of an object seldom occurs at its interior part, except when it is hollow.

Under these experimental conditions, we evaluate the progress of change in the learned knowledge and the recognition results. The learned knowledge means the number of nodes, clusters, and concepts. As mentioned in section 2.1, the system outputs any of the three types of recognition results against each input pattern. Because of that, we define two values, known ratio and callable ratio, and we evaluate the progress of their changes.

Known ratio The rate of recognition result that any concepts react to the input pattern in last 1,000 results.

Callable ratio The rate of the recognition result that the concept with correct label reacts to the input pattern in last 1,000 results.

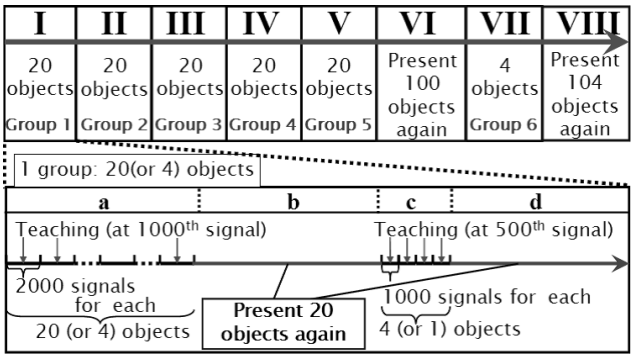

Fig. 5. Sequence of presentation

In this experiment, we set the four parameters as follows: $\lambda = 100$, $age_{dead} = 100$.

B. Experiment: An Example of Life-long Learning Tasks Using Artificial Objects

1) Experimental Conditions: In this experiment, we use the artificial objects shown in Figure 3, and evaluate the progress of learning and recognition. We use 13 colors (red, blue, purple, etc) and eight figures (circle, square, etc) for all the objects, thus the 104 objects are used ($13 \times 8 = 104$). We associate each object with different labels.

104 objects are divided into six groups. Five groups (Group 1–5) include 20 objects each and one group (Group 6) includes four objects. Then we assume the sequence that includes eight environments (Figure 5). These environments include different appearing groups each. There are following two different meanings of these environments.

- Appearance of new kind of objects (I–V, VII)
  In this environment, new objects appear sequentially. In the environment I, objects that belong to Group 1 and those of Group 2 appear in the environment II, etc. We give system the labels of objects once each object.
- Appearance of all kinds of objects (VI, VIII)
  In this environment, all objects appear sequentially again, that have appeared already. We do not give any labels.

Moreover, in the environments where the new group appears, we set similar sub-environments as follows;

a) All 20 (or four) objects appear sequentially. We give the labels of 16 (or three) objects that are chosen randomly.
b) All objects appear again. We do not give any labels.
c) Four (or one) objects appear again, of which we have not given the label. And we give their labels.
d) All objects appear again. We do not give any labels.

We set the number of input signals to 2,000 signals at the first appearance of each object, and 1,000 signals at the other appearances. And we teach at 1,000th signal in the section (a),

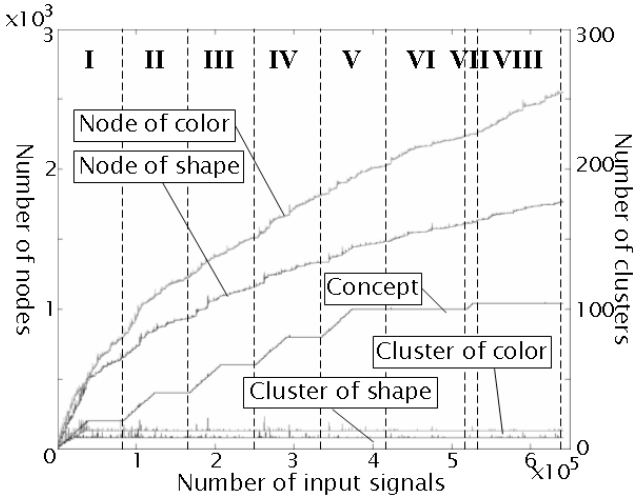

Fig. 6. Change of the number of nodes, clusters and concepts

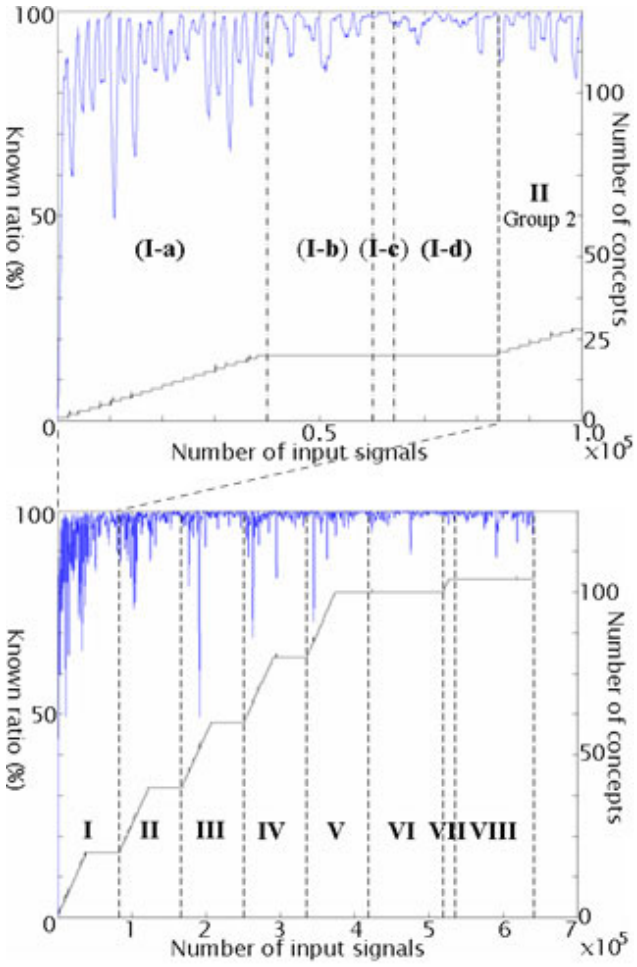

Fig. 7. Change of known ratio: Graphs show the result in Environment I (upper part) and the whole result (under part)

and at 500th signal in the section (c). However, these settings are not necessary for operation of the proposed method. We set them only to verify the experimental result at a glance.

2) Result and Discussion: Figure 6 shows the progress of change in the number of nodes, clusters, and concepts. Insertion of nodes occurs when it is necessary to represent new data structure. In the beginning, the number of nodes increases by the necessity to adapt to new environment. Then the increases get smaller and smaller as the adaptation proceeds. The numbers of clusters also increase in the beginning, then they are stable at 13 clusters of color and eight clusters of shape. Concepts are created when the new objects appear (I–V, VII), and no superfluous concept is created when the objects appear again. At the end, the proposed method has 104 concepts. These results show the proposed method can adjust the number of nodes, clusters, and concepts, as circumstances demand.

Figure 7 and 8 show the progress of the change in the known ratio and the callable ratio. For the following explanation, the progress of the change in the number of concept is shown with the known ratio.

Upper part of Figure 7 shows the result of the known ratio under the environment I. In the section (I–a), the first appearance of 20 objects, the system creates concepts sequentially. So the known ratio begins to increase when the new concept is created, while it begins to decrease when the new object appears. In the subsequent environments (from I–b to I–d), the system constantly indicates a high known ratio. Under part of Figure 7 shows that the results in the environment II–V and VII denote the same tendency of that in the environment I. In the environment VI and VIII, where the all objects appear again that have appeared already, the system also indicates the high known ratio. That means the proposed method can learn information of 104 objects incrementally and can preserve it.

Upper part of Figure 8 shows the result of the callable ratio under the environment I. In the section (I–a), where 16 labels of the objects are given, the proposed method associates the created concepts with the given labels. So the callable ratio begins to increase each time the label is given. Of course the proposed method cannot call the labels of four objects about which the labels are not given. After the section (c), the system constantly indicates the high callable ratio. These results show the proposed method can flexibly correspond to the various timing of teaching. Under part of Figure 8 shows the results in the environment II–V and VII denote the same tendency of that in the environment I. And in the environment VI and VIII, the system also indicates the high callable ratio. That means the proposed method can adjust relations between the 104 concepts and the input data structures based on these labels, and can preserve all learned information.

IV. Conclusion

This paper presents a novel method for life-long semi-supervised learning. Through the learning using different type of feature vectors, the system can get the information that is the combinations of clusters in different feature spaces

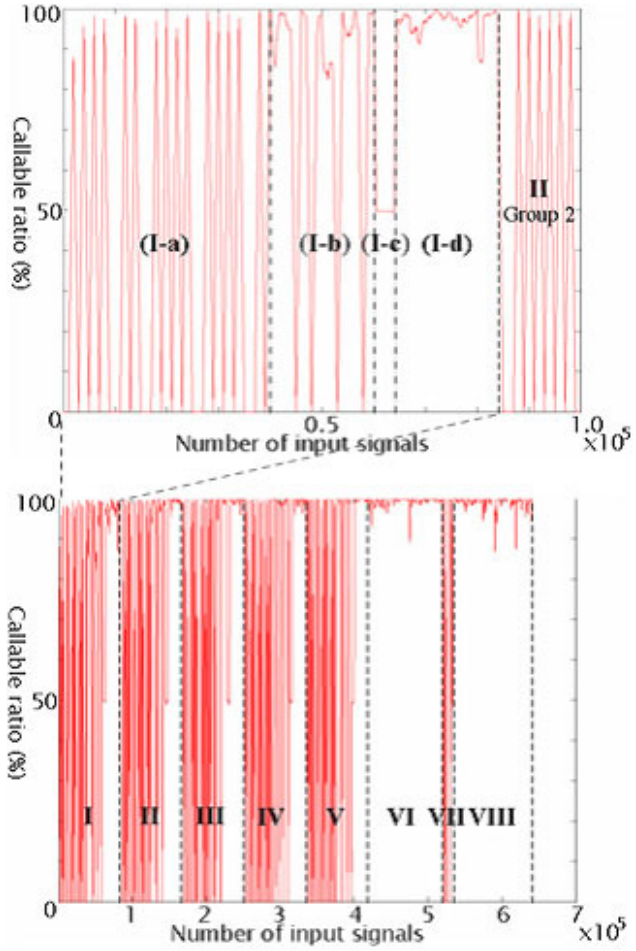

Fig. 8. Change of callable ratio: Graphs show the result in Environment I (upper part) and the whole result (under part)

incrementally. And it can adjust previous learned information based on scarce labeled data. It is independent of predefined optimal conditions. It can continue to recognize input patterns concurrently with learning. The experiment using 104 artificial objects shows that the proposed method can realize life-long learning with a high recognition ratio, nonstop, and even in real time.

We must mention that some other problems remain unsolved. For example, if the occurrence probability of labeled data of each class is different, the adjustment of previous learned information becomes difficult. It is difficult to define the criterion by which the system judges whether the labeled data is available or not. This problem remains as a subject for our future research.

ACKNOWLEDGMENT

This study was supported by the Industrial Technology Research Grant Program from the New Energy and Industrial Development Organization (NEDO) of Japan.

REFERENCES

[1] G. A. Carpenter and S. Grossberg, "The art of adaptive pattern recognition by a self-organizing neural network," *IEEE Compute*, vol. 21, no. 3, pp. 77–88, 1988.

[2] B. Fritzke, "A growing neural gas network learns topologies," in *Neural Information Processing Systems*, vol. 7. Denver, USA: MIT Press, 1995, pp. 625–632.

[3] J. Bruske and G. Sommer, "Dynamic cell structure learns perfectly topology preserving map," *Neural Computation*, vol. 7, no. 4, pp. 845–865, 1995.

[4] F. H. Hamker, "Life-long learning cell structures — continuously learning without catastrophic interference," *Neural Networks*, vol. 14, no. 4, pp. 551–572, 2001.

[5] Anonymous, "to appear in the final paper," *Neural Networks*, vol. 19, no. 1, 2006.

[6] S. Grossberg, "Nonlinear neural networks: principles, mechanisms, and architectures," *Neural Networks*, vol. 1, pp. 17–61, 1988.

[7] K. Nigam, A. K. McCallum, S. B. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine Learning*, vol. 39, no. 2, pp. 103–134, 2000.

[8] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *Seventh IEEE Workshop on Applications of Computer Vision*, 2005.

[9] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998.

[10] M. Seeger, "Learning with labeled and unlabeled data," University of Edinburgh, Tech. Rep., 2001.

[11] D. Zhow, O. Bousquet, T. N. Lal, J. Weton, and B. Scholkopf, "Learning with local and global consistency," in *Neural Information Processing Systems*, vol. 16. Cambridge, USA: MIT Press, 2004, pp. 321–328.

[12] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. 18th International Conference on Machine Learning*, 2001.

[13] X. Zhu, "Semi-supervised learning literature survey," University of Wisconsin, Tech. Rep., 2006.