Task Segmentation in a Mobile Robot by mnSOM and Hierarchical Clustering

M. Aziz Muslim, Masumi Ishikawa, and Tetsuo Furukawa
Dept. of Brain Science and Engineering, Kyushu Institute of Technology, Japan
2-4 Hibikino, Wakamatsu, Kitakyushu, Japan 808-0196
email: aziz-muslim@edu.brain.kyutech.ac.jp ,{ishikawa, furukawa}@brain.kyutech.ac.jp

Keywords: mnSOM, task segmentation, hierarchical clustering

Abstract**—**Our previous studies assigned labels to mnSOM modules based on the assumption that winner modules corresponding to subsequences in the same class share the same label. We propose segmentation using hierarchical clustering based on the resulting mnSOM. Since it does not need the above unrealistic assumption, it gains practical importance at the sacrifice of the deterioration of the segmentation performance by 1.2%. We compare the performance of task segmentation for two kinds of module architecture in mnSOM. The result is that module architecture with sensory-motor signals as target outputs has superior performance to that with only sensory signals as target outputs.

1 Introduction

Task segmentation in navigation of a mobile robot based on sensory-motor signals is important for realizing efficient navigation, hence has attracted wide attention. Tani et al. proposed to generate a series of actions based on sensory-motor signals using a forward model represented by a recurrent neural network [2]. Tani and Nolfi [3] proposed 2-level hierarchical mixture of recurrent experts (MRE), which is an extension of the network architecture proposed by Jacobs et al. [8]. Tani et al. also proposed 2-level prediction networks for extracting spatio-temporal regularities [11]. Wolpert and Kawato [1] proposed MOSAIC architecture for motor control with the soft-max function for assigning responsibility signal to each module.

In the conventional competitive learning, only a winner module or unit is highlighted, accordingly similarity between modules or units, and interpolation among them are not taken into account. There are two types of “interpolation:” the one is creating an output which is an interpolation of outputs of multiple modules, and the other is creating a module which is an interpolation of multiple modules. Let the former be called “output interpolation” and the latter be called “module interpolation.” The present study focuses on the latter.

The soft-max [1] is an improvement over the conventional competitive learning in that the output interpolation is possible based on the responsibility signals produced by the soft-max function. Similarity between modules, however, is not explicitly represented. Furthermore, the soft-max function and segmentation do not generally coexist; only when the soft-max function is asymptotically equivalent to winner-take-all, segmentation is possible at the sacrifice of interpolation. Tani et al. proposed a recurrent neural network with a parametric bias [4]. It has the ability of the output interpolation, but has no longer the capability of segmentation.

Self Organizing Maps (SOM)[12] is a popular method for classification, while preserving topological relationship between data. The resulting topological maps demonstrate the unit interpolation among units on a competitive layer. In contrast to SOM using a vector unit as its element, a modular network SOM (mnSOM) uses a function module as its element to increase its representation and learning capability [13][14][5]. Owing to competitive learning among function modules, mnSOM is capable of segmentation. Owing to topographic mapping of function modules on a competitive layer, neighboring function modules tend to have similar characteristics. Hence, interpolation among function modules becomes possible. The simultaneous realization of segmentation and interpolation is unique and unparalleled characteristics of mnSOM. It has also an advantage of computational stability in contrast to competitive learning due to careful assignment of learning rate to modules and data.

We proposed to use mnSOM for task segmentation in navigation of a mobile robot [6][7]. In case of a mobile robot, however, the standard mnSOM is not applicable as it is, because it is based on the assumption that class labels are known *a priori*. In a mobile robot, however, only an unsegmented sequence of data is available. Hence, we proposed to decompose it into many subsequences, supposing that a class label does not change within a subsequence. Accordingly, training of mnSOM is done for each subsequence in contrast to that for each class in the standard mnSOM.

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

Our previous studies assigned labels to mnSOM modules based on the assumption that winner modules corresponding to subsequences in the same class share the same label. We propose segmentation using hierarchical clustering based on the resulting mnSOM. Since it does not need the above unrealistic assumption, it is expected to gain practical importance.

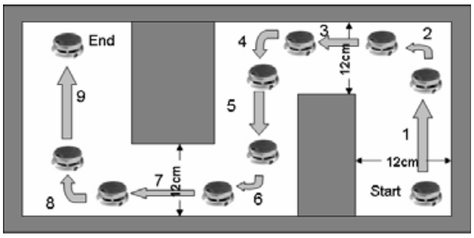
We also compare the performance of task segmentation for two kinds of module architecture in mnSOM: sensory-motor signals at the next time step as target outputs and sensory signals at the next time as target outputs.

Section 2 briefly explains algorithms for mnSOM and hierarchical clustering. Section 3 describes experimental results. Section 4 provides conclusions and discussions.

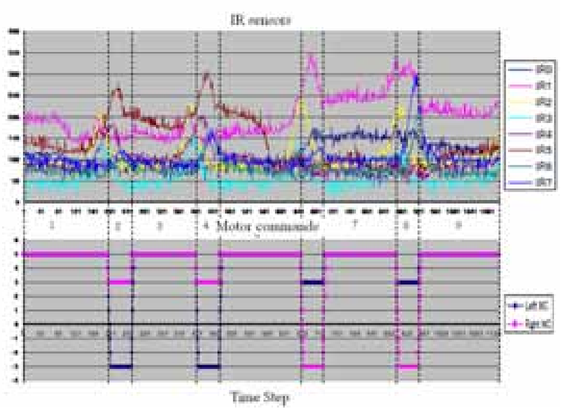
2 Task segmentation by mnSOM

2.1 Data Segmentation

Fig.1(a) illustrates the robotic field. The robot moves from the start position to the end position by wall following. During movement, the robot turns left twice and turns right twice. When the robot moves in the reverse direction, it experiences similar movements.



(a)



(b)

Fig.1: Mobile robot and data segmentation
(a) the mobile robot in the robotic field. (b) manual segmentation of sensory-motor signals

The whole dataset are manually segmented into 9 sequences based on motor commands as in Fig.1(b). Manual segmentation provides true class information, and is used for evaluation of the resulting segmentation performance, not for training of mnSOM. Sequences 1, 3, 5, 7 and 9 correspond to a class of forward movement, sequences 2 and 4 correspond to a class of left turn, and sequences 6 and 8 correspond to a class of right turn.

As mentioned in Introduction, we decompose the whole sequence into many subsequences of a uniform length, supposing that a class label does not change within a subsequence [6][7]. Each subsequence is assigned a label of the corresponding sequence. Due to uniform splitting of the whole dataset, however, some subsequences stretch over two consecutive sequences (i.e., a forward movement sequence and a left turn sequence). They are called "transition" subsequences, and constitute virtual classes.

2.2 Algorithm of mnSOM

An important issue in the design of mnSOM architecture is to choose appropriate function modules and similarity measure between modules. To deal with dynamical systems, recurrent neural networks (RNN) are suitable as function modules [2]. Learning of each module is done by backpropagation through time (BPTT) [9].

mnSOM learns an internal model of robot-environment interaction by minimizing mean prediction error in sensory or sensory-motor signals at the next time step, given the current sensory-motor signals. We present a brief overview of an mnSOM algorithm [5][13][14].

An mnSOM algorithm comprises the following 4 processes.

(1) In an evaluative process, subsequences are fed into mnSOM modules, and their outputs are evaluated. Let $\{(x_i(t), y_i(t))\}$ $(i = 1 \ldots M)$ be a pair of current sensory-motor signals and the sensory-motor signals at the next time step, where M is the number of subsequences and t is the iteration number in mnSOM.

(2) In a competitive process, the module with the minimum prediction error is determined as a winner.

(3) In a cooperative process, the learning rate, $\Psi_i^{(k)}(t)$, for module k and subsequence i is defined by the following normalized neighborhood function centered at the winner module.

$$\Psi_i^{(k)}(t) = \frac{\phi(r(k, v_i^*); t)}{\sum_{i'=1}^{M} \phi(r(k, v_i^*); t)} \qquad (1)$$

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

$$\phi(r;t) = \exp\left[-\frac{r^2}{2\sigma^2(t)}\right] \quad (2)$$

$$\sigma(t) = \sigma_{\min} + (\sigma_{\max} - \sigma_{\min})e^{-\frac{t}{\tau}} \quad (3)$$

where $r(k, v_i^*)$ stands for the distance between module k and the winner module v_i^*, $\sigma_{\min}$ is the minimum neighborhood radius, $\sigma_{\max}$ is the maximum neighborhood radius, and τ is a neighborhood decay rate.
(4) In an adaptive process, connection weights of module k, $\mathbf{w}^{(k)}$, are modified by the following BPTT algorithm,

$$\Delta\mathbf{w}^{(k)} = \sum_{i=1}^{M} \Psi_i^{(k)}(t)\left(-\eta \frac{\partial E_i^{(k)}}{\partial \mathbf{w}^{(k)}}\right) \quad (4)$$

where $E_i^{(k)}$ is the output error of module k in subsequence i.

These 4 processes are iterated, and mnSOM terminates when connection weights converge and the resulting mnSOM becomes stable.

2.3 Clustering

We apply hierarchical clustering to the resulting mnSOM to provide labels to modules without the assumption that winner modules corresponding to subsequences in the same class share the same label.

The procedure of hierarchical clustering [10] is:
1) Let each module form a separate cluster.
2) Merge two clusters with the minimum distance.
3) Recalculate the distance between clusters.
4) Repeat steps 2 and 3 until the minimum distance between clusters exceeds a given threshold or the number of clusters reaches a given number of clusters.

An essential issue in hierarchical clustering is the definition of the distance. First we select a subsequence which minimizes the mean square error of module *k*.

$$m_k = \arg\min_i MSE(i,k) \quad (5)$$

where *MSE(i,k)* stands for the mean square error of module *k* given input subsequence *i*. The distance between modules k_1 and k_2 is defined by:

$$d_{k_1k_2} = \sqrt{\begin{aligned}&\left(MSE(m_{k_1},k_2) - MSE(m_{k_1},k_1)\right)^2 \\ &+ \left(MSE(m_{k_2},k_1) - MSE(m_{k_2},k_2)\right)^2\end{aligned}} \quad (6)$$

The inclusion of only the subsequences m_{k_1} and m_{k_2} in the definition is to prevent the distance from being blurred by many less relevant subsequences.

We then define the distance between clusters K_1 and K_2. Suppose that the cluster K_1 is composed of modules, $M_1 \cdots M_{R_{K_1}}$, and the cluster *J* is composed of modules, $M_1 \cdots M_{R_{K_2}}$. The distance between them is defined by,

$$D_{K_1K_2} = \frac{1}{R_{K_1}R_{K_2}} \sum_{k_1=1}^{R_{K_1}} \sum_{k_2=1}^{R_{K_2}} d_{k_1k_2} \quad (7)$$

where $d_{k_1k_2}$ is the distance between two modules k_1 and k_2.

3 Experimental Results

Experiments are carried out using a Khepera II robot moving in the robotic field in Fig. 1 (a). It has 8 infra-red (IR) proximity sensors and 2 separately controlled DC motors. The sensors can detect an obstacle within 5 cm. The robot, performing wall following behavior, is controlled by a PC via serial connection.

3.1 Task Segmentation by SOM

As a preliminary study, a SOM with 10x10 nodes is employed to do task segmentation. Concatenation of 10 dimensional raw sensory-motor signals at current time step and 10 dimensional raw sensory-motor signals at the next time step are used as input to SOM. Fig. 2 depicts the resulting SOM after 10000 iterations. We then label nodes in SOM under the assumption that winner nodes corresponding to input data in the same class share the same label. Those nodes which never become winners are labeled by the label of input data which is closest to the node concerned. Based on the resulting task map, we perform task segmentation. The segmentation rate is 91.8% for training dataset and 90.5% for novel dataset.

3.2 Task Segmentation by mnSOM

We employ an mnSOM with 10x10 modules. Each module in mnSOM is a fully connected recurrent neural network (FRNN). All units in FRNN have sigmoidal activation functions. External input units in FRNN correspond to 8 IR sensors and 2 motor commands. To evaluate the role of motor commands, two kinds of target outputs are used: the former is the use of sensory signals at the next time step, and the latter is the use of sensory-motor signals at the next time step.

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

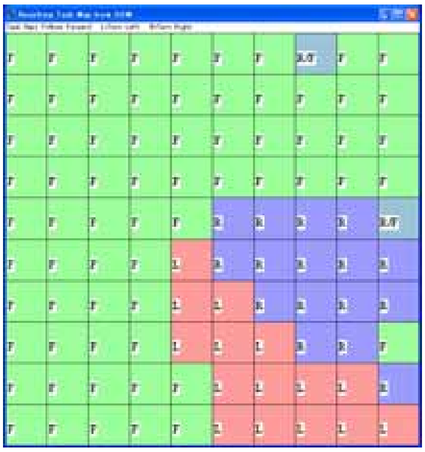


Fig.2: The resulting task map using SOM (Label “F”, “L” and “R” correspond to winner of forward movement, left turn and right turn samples, respectively)

mnSOM modules learn internal models of nonlinear dynamics of robot-environment interaction by minimizing mean prediction error of sensory or sensory-motor signals at the next time step, given the past sensory-motor signals. The resulting mnSOM provides a label to each module by the procedure in [6][7], under the assumption that winner modules corresponding to subsequences in the same class share the same label. The labeling procedure is approximately the following. For a winner module the label is given by the label of the corresponding input subsequence. For a non-winner module the lebel is given by the label of the input subsequence with the minimum MSE.

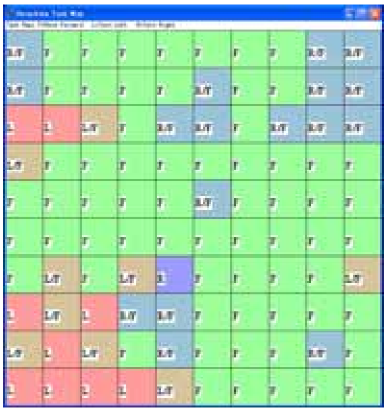


Fig.3: The resulting task map by mnSOM with sensory signals as target outputs

Fig. 3 illustrates the resulting task map with sensory signals as the target outputs, and Fig. 4 depicts that with sensory-motor signals as the target outputs. The modules in green color are the winners for forward movement subsequences, the modules in red color are the winners for left turn subsequences, and the modules in blue color are the winners for right turn subsequences. Let the modules in red and blue be called “turning modules.” The other two types of modules are transition modules: the modules in light red color for the transition between left turn and forward movements, and the modules in light blue color for the transition between right turn and forward movement.

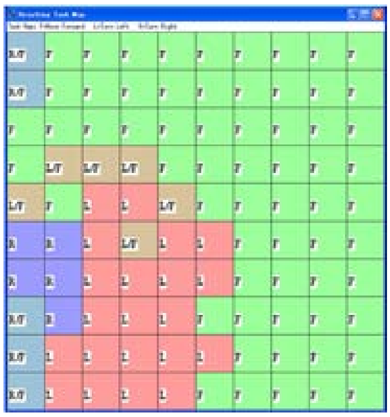


Fig.4: The resulting task map by mnSOM with sensory-motor signals as target outputs

To evaluate the segmentation ability of these task maps, training datasets as well as novel dataset are given to them. Table 1 summarizes the correct segmentation rate. It indicates that mnSOM with sensory-motor signals as target outputs has superior performance to mnSOM with only sensory signals as target outputs.

Table 1: Summary of segmentation performance

target output for mnSOM	correct segmentation rate (%)	
	training datasets	novel Dataset
sensory signals	80.35	82.14
sensory-motor signals	96.73	94.05

3.3 Hierarchical clustering

We apply the standard hierarchical clustering to assign label to modules. Fig. 5 illustrates the result of clustering with 5 clusters including 2 virtual classes. Fig.6 illustrates the result of clustering with 3 clusters corresponding to 3 basic classes: forward movement, left turn and right turn. Two virtual classes in Fig. 5 (represented by light blue and light red modules) are absorbed into right turn modules in Fig. 6.

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

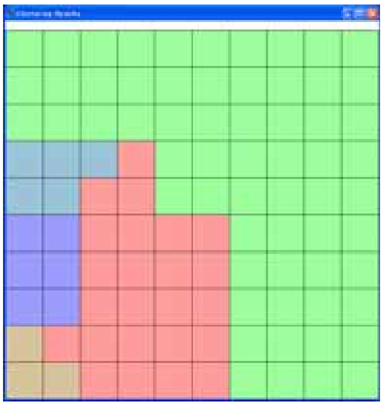


Fig.5: The result of clustering for the resulting mnSOM (5 clusters) with sensory-motor signals as target output

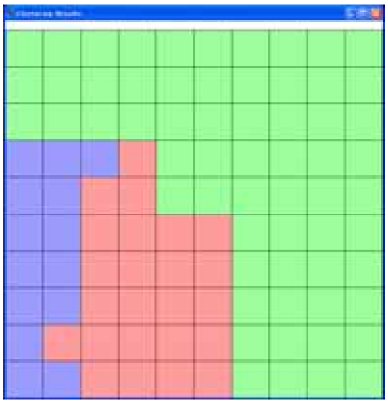


Fig.6: The result of clustering for the resulting mnSOM (3 clusters) with sensory-motor signals as target output

Comparison of Figs. 4 and 5 demonstrates the similarity of colors to some degree. Supposing that color in Figs. 4 and 5 has the same meaning, 85 modules (85%) share the same meaning.

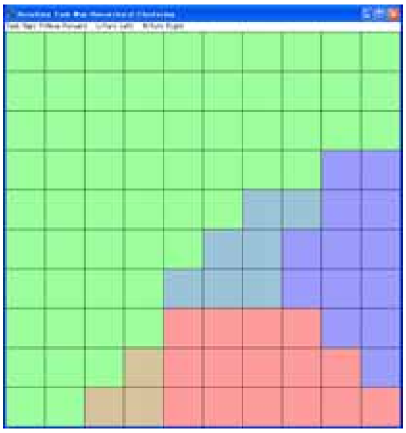


Fig.7: The result of clustering for the resulting SOM

Hierarchical clustering also can help in assigning node label of the resulting SOM, without using the assumption that winner nodes corresponding to input data in the same class share the same label. Fig. 7 illustrates the result. Comparison of Figs. 2 and 7 also demonstrates the similarity of colors to some degree.

3.4 Task Segmentation by clustering

Fig.8. illustrates the task segmentation by hierarchical clustering applied to the resulting mnSOM, and labels for novel subsequences. The colored numbers of mnSOM modules represent subsequence numbers for which they become winners. Incorrect task segmentation occurs when the color of subsequence number does not match with the color of the corresponding winner module.

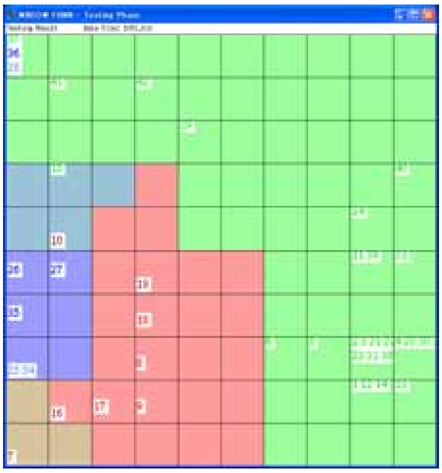


Fig.8: Result of clustering and labels for novel subsequences

Table 2 summarizes the segmentation performance by clustering applied to the resulting mnSOM as well as to the resulting SOM. It indicates that segmentation performance in mnSOM with 5 clusters is superior to that with 3 clusters. This is reasonable because the original number of clusters is 5. Comparison of Tables 1 and 2 demonstrate that segmentation rate by hierarchical clustering is slightly inferior to that by SOM or mnSOM [7] with the assumption that winner modules corresponding to subsequences in the same class share the same label. Since segmentation by clustering does not need the unrealistic assumption, it is expected to gain practical importance.

Current sensory-motor signals have very small fluctuation, hence the segmentation performance by clustering of the resulting SOM is very good both for training data and novel data. Suppose there exist a right turn movement for a very short period. It is easily interpreted as right turn. In this way, segmentation by SOM is easily affected by fluctuation and noise in data. In contrast to this, mnSOM is expected to be more robust than SOM, because it learns nonlinear dynamics of a mobile robot concerned. We

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

have not done segmentation using SOM for data corrupted by noise, which is left for future study.

Table 2: Summary of segmentation performance by clustering

applied to	correct segmentation rate (%)	
	training datasets	Novel dataset
mnSOM (5clusters)	89.29	92.85
mnSOM (3 clusters)	83.93	89.26
SOM (5 clusters)	94.3	94.0

4 Conclusions and Discussions

Our previous studies on task segmentation adopted an assumption that winner modules corresponding to subsequences in the same class shared the same label. We proposed task segmentation using hierarchical clustering based on the resulting mnSOM. Since it does not need the above unrealistic assumption, it gains practical importance at the sacrifice of the deterioration of the segmentation performance by 1.2%.

We compared the performance of task segmentation for two kinds of module architecture in mnSOM. The result is that module architecture with sensory-motor signals as target outputs has superior performance to that with only sensory signals as target outputs.

In the current study sensory-motor signals are obtained from a real mobile robot off-line, and task segmentation is done successfully based on them. In a real world situation, the sequence proceeds as follows. Take an action, observe sensory signals, determine a winner module based on the criterion of the mean square error (MSE), determine motor-control signals from the output of the winner module, and so forth. As can be seen from this, goodness of motor-control signals depends on sensory-motor signals for training. How to select a set of sensory-motor signals is an issue yet to be solved and is left for future study.

Acknowledgements

This research was partially supported by the 21st Century COE (Center of Excellence) Program and by Grant-in-Aid for Scientific Research(C)(18500175) both from the Ministry of Education, Culture, Sports, Science and Technology(MEXT), Japan.

References

[1] D.M. Wolpert, and M. Kawato, “Multiple paired forward and inverse models for motor control”, *Neural Networks*, Vol.11, pp. 1317-1329, 1998.

[2] J. Tani, “Model-based Learning for Mobile Robot Navigation From Dynamical System Perspective”, *IEEE Transactions on System, Man and Cybernetics Part B*, 26(3), pp. 421-436, 1996.

[3] J. Tani, and S. Nolfi, “Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems”, *Neural Networks*, pp. 12, 1131-1141, 1999.

[4] J. Tani, M. Ito, and Y. Sugita, “Self-organization of distributedly represented multiple behavior schemata in a mirror system: reviews of robot experiments using RNNPB”, *Neural Networks*, Vol.17, pp. 1273-1289, 2004.

[5] K.Tokunaga, T. Furukawa, and S. Yasui, “Modular Network SOM: Extension of SOM to the realm of function space”, *Proc. of Workshop on Self Organizing Maps (WSOM2003)*, pp. 173-178, 2003.

[6] M. Aziz Muslim, Masumi Ishikawa, and Tetsuo Furukawa, “A New Approach to Task Segmentation in Mobile Robots by mnSOM”, *Proc. of 2006 IEEE World Congress on Computational Intelligence (IJCNN2006 Section)*, Canada, pp. 6542-6549, 2006.

[7] M. Aziz Muslim, Masumi Ishikawa, Tetsuo Furukawa, “Task Segmentation in a Mobile Robot by mnSOM : A New Approach To Training Expert Modules”, *Neural Computation and Application*, Springer, 2007 (accepted).

[8] R. Jacobs, M. Jordan, S. Nowlan, and G. Hinton, “Adaptive Mixtures of Local Experts”, *Neural Computation*, pp. 79-87, 1991.

[9] R.J.Williams, and D. Zipser, “Gradient-based learning algorithms for recurrent networks and their computational complexity”, *Backpropagation: Theory, Architectures and Applications*, Erlbaum, pp. 433-486, 1992.

[10] R.O. Duda, P.E. Hart, and D.G. Stork, “ Pattern Classification”, *Wiley-Interscience*, 2001.

[11] S. Nolfi, and J. Tani, “Extracting Regularities in Space and Time Through a Cascade of Prediction Networks : The Case of a Mobile Robot Navigating in a Structural Environtment”, *Connection Science*, (11)2, pp. 129-152, 1999.

[12] T. Kohonen,“Self-Organizing Maps”, *Springer*, 1995.

[13] T. Furukawa, K. Tokunaga, S. Kaneko, K. Kimotsuki, and S. Yasui, “Generalized self-organizing maps (mnSOM) for dealing with dynamical systems”, *Proc. of NOLTA2004*, Fukuoka, Japan, pp. 231-234, 2004.

[14] T. Furukawa, K. Tokunaga, K. Morishita, and S. Yasui, “Modular Network SOM (mnSOM): From Vector Space to Function Space”, *Proc. of IJCNN2005*, Canada, 2005.

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