Generation of condition-dependent reaching movements based on layered associative networks

Masaki Ogino
Faculty of Informatics, Kansai University
2-1-1 Ryozenji, Takatsuki,
OSAKA 569-1095 JAPAN
Email: ogino@res.kutc.kansai-u.ac.jp

Mai Hikita Sawa Fuke Minoru Asada
Department of Adaptive Machine Systems,
Graduate School of Engineering,Osaka University
2-1 Yamadaoka, Suita, Osaka 565-0871 Japan
Email: {hikita, fuke}@er.ams.eng.osaka-u.ac.jp
asada@ams.eng.osaka-u.ac.jp

***Abstract*—This paper proposes a hierarchical network that enables an information processing nesting structure for tool-use that represents a relationship between a hand end-effector and a tool target when reaching for the tool, and an additional relationship between the tip of the tool end-effector and an object target after taking the tool. The network consists of associative networks whose lower layer networks associate the multimodal information under various conditions. The higher layer network associates the temporal order of the lower network states. A simulation experiment shows that the proposed network successfully generates reaching movements regardless of the conditions.**

I. Introduction

We can adaptively use various kinds of tools adaptively as if they were part of our bodies. Iriki et al. [9] investigated the activities of bimodal neurons in macaque monkeys before and after tool-use learning (the neurons are called bimodal because they are activated during both visual and tactile sensing). They showed that, after learning, the receptive field of the bimodal neurons extended from hand to tool.

In the cognitive developmental robotics area, many models have been developed to explain how body representation is acquired. Hikita et al. proposed a learning model that associates visual information with the somatic senses during the the co-occurrence of touch sensing on the basis of bottom-up attention [7]. Nabeshima proposed a model for tool-use in which the inertia parameters of the tools are estimated and used for adaptive handling of a never-before-seen tool [13]. These models attempt to adaptively associate the position of the end-effector and the posture information in one system where the end-effector can be the hand or the tool.

However, even though the associations between the end-effector and the targets are learned, the problem of how appropriate motions can be generated depending on various conditions still exists. Note that even if the macaque monkeys have a tool in hand, the bimodal neurons does not extend to the tool if they do not want to take food with the tool [10].

Module selection and identification for control (MOSAIC) [17] is a possible mechanism for situation-dependent motion generators. MOSAIC prepares multiple modules, each of which is a combination of a prediction model and a control model for different motion primitives. The higher layer selects the appropriate module that yields the best prediction in the current situation. Tani et al. proposed a recurrent neural network with parametric bias (RNNPB), in which the same recurrent network generates different dynamics output depending on the parametric bias [16]. Nishide et al. proposed a multiple time-scales recurrent neural network (MTRNN) that is an extension of RNNPB with a layered network, in which different layers have different time scales [14]. They showed that the network can generate the appropriate motion patterns depending on the given target tools that are placed in the same location. However, these models did not realize the sequential procedures depending on the network situations.

The change in the body representation can be regarded as that in the combinations of an end-effector and a target object. For example, when food is within close reach, the end-effector is the hand and the target object is the food. In a situation where the food can be reached with a tool, it is necessary to plan the motion procedures — first take the tool with the hand and then take the food with the tool. In the first procedure, the end-effector is the hand and the target object is the tool. In the second procedure, the end-effector is the tool and the target object is the food.

This paper proposes a layered network that generates the appropriate procedures for multiple end-effectors (hand and tool) on the basis of the nesting structure of the end-effectors and the targets. We investigate how the layered networks interact with each other, and determine the appropriate tool-use status when the associative network is layered.

We first introduce the Restricted Boltzmann Machine (RBM) as a unit of the proposed layered network. Second, the details of the task setting for the tool-use are explained. Then, the proposed hierarchical network is introduced. Finally, the problems with regard to the layered network are discussed.

II. Restricted Boltzmann Machine

RBM [8] is a neural network consisting of two layers, i.e., the input (visible) layer and the hidden layer. There are no connections among units within each layer. Each unit in the visible layer, v_i, has a symmetrical connection weight, w_{ij}, to each unit in the hidden unit, h_j. Each unit is activated by the

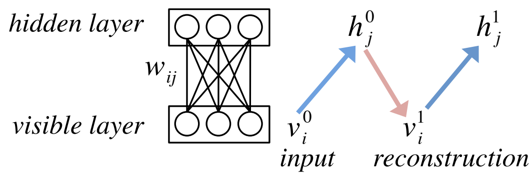


Fig. 1. Restricted Boltzmann Machine

following probabilities:

$$\mathbf{P}(h_j = 1) = \frac{1}{1 + exp(-\sum_i v_i w_{ij} - \beta_{h_j})} \quad (1)$$

$$\mathbf{P}(v_i = 1) = \frac{1}{1 + exp(-\sum_j h_j w_{ij} - \beta_{v_i})}, \quad (2)$$

where β_{v_i} and β_{h_j} are activation biases.

RBM learning is accomplished via a calculation process called *reconstruction*. First, the activation level of the hidden layer, h_j, is calculated via the forward calculation on the basis of the input data v_i, the connection weights, w_{ij}, and biases, β_{v_i}, using eq. (1). Then, the activation level of the visible layer, v_i, is calculated again with that of the hidden layer, h_j, with eq. (2). This calculation process can be executed repeatedly. The superscript of the units v_i and h_j indicate the number of repeated calculations between layers. If the probabilistic distributions of the input data and the reconstructed data after ∞ reconstruction repeats are denoted by $\hat{p}(\mathbf{v})$ and $\hat{p}(\mathbf{v}|\mathbf{w})$, respectively, learning consists of adjusting the connection weights, w_{ij}, to minimize the difference between $\hat{p}(\mathbf{v})$ and $\hat{p}(\mathbf{v}|\mathbf{w})$. The distance between two distributions can be measured by the cross-entropy error, which is defined by the following equation:

$$L = \langle \log(p(\mathbf{v}|\mathbf{w})) \rangle_{\hat{p}(\mathbf{x})} \quad (3)$$

$$= -\sum_{i=0}^{N} \hat{p}(\mathbf{v}_i) \log(p(\mathbf{v}_i|\mathbf{w})). \quad (4)$$

The total energy of the RBM network with the activation level, $(\mathbf{v}, \mathbf{h})$ in both layers can be defined by the following equation:

$$E(\mathbf{v}, \mathbf{h}|\mathbf{w}) = -\sum_{i,j} v_i h_j w_{ij}. \quad (5)$$

The connection weights are trained such that this energy is lowered. After learning, the reconstruction process can be used to reconstruct a complete dataset from the incomplete data. This feature can be used to associate the given multiple datasets. Moreover, by comparing with the network energy, it is possible to evaluate how close the input data are to the learning data.

III. Task Setting of Reaching with Tools

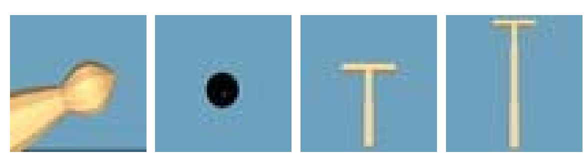
Nesting is an important information process that enables higher cognitive abilities such as language and tool-use [4]. To realize nesting in a layered network, it is important that the higher layer controls the temporal sequences of the lower layer network states. In this paper, taking a reaching task with a tool as an example, we consider how the layered associative network makes it possible to proceed with the nesting task.

Although there are many abilities involved in tool-use, we concentrate on the reaching task, in which an agent must first reach for a tool with his/her hand, and then reach for a target object with the tool. Specifically, an agent should modify the target object posture to reach the target depending on what is (or is not) in his/her hand. Note that we concentrate on the problem of how the end posture is generated and not on how the motor commands during reaching are generated.

At first, the simulation robot moves its hand randomly on the table on which the tools or food are placed. When the robot detects the touch senses, RBM modules associate the joint angles of the arm, those of the eyes, and the image features of the targets and end-effectors.

The robot uses only the left arm that has five degrees of freedoms (DOFS). The table size in front of the robot is 0.5 [m] wide $\times$ 0.2 [m] deep.

The range of each joint angle is equally segmented into 16 divisions, each of which is the receptive field of one neuron. The neuron output is binary; only one neuron corresponding to the current joint angle among 16 neurons is activated at one time. Thus, $16x5 = 90$ neurons are prepared for the joint angles of the arm. The pan and the tilt angles of the two eyes are input to the network when an object (hand ,tool and target) is captured in the center of the eye image (this is ”attention” to the object). As well as for the arm, the range of each joint angle is segmented into 10 divisions. Thus, 10x4=40 neurons are prepared for the joint angles of the eyes. The input eye image is segmented into $5x5$ areas, in each of which the results of 4 types of Gabor filters are binarized with certain threshold. Thus, the $5x5x4 = 100$ neurons are prepared for the image features. In the reconstruction process, the result of the Gabor filters is reconstructed and compared with input data in this level (not the image itself). Note that for the image feature inputs, we take the constant values prepared in advance on the basis of the images shown in Fig. 2. The robot explores in four environments, as shown in Fig. 3.



(a) hand (b) food (c) tool1 (d) tool2

Fig. 2. Input images of end-effectors and target objects.

IV. Condition-dependent Reaching Control System

A. Overview of the System

Figure 4 shows an overview of the proposed system. The sensor information of the joint angles of the arm, those of the eyes, and the feature values of the visual image during reaching is input to the lower layer of the network. The lower layer consists of three RBMs: *reaching motion module*, *reaching area module*, and *attention image module*. *The reaching*

Environments	Food	Tool1	Tool2	motion order
1	Reachable	—	—	hand food
2	Reachable with tool 1	Reachable with hand	—	hand tool1 food
3	Reachable with tool 2	—	Reachable with hand	hand tool2 food
4	Reachable with tool 2	Reachable with hand	Reachable with tool 1	hand tool1 tool2 food

Fig. 3. Conditions of simulation experiment.

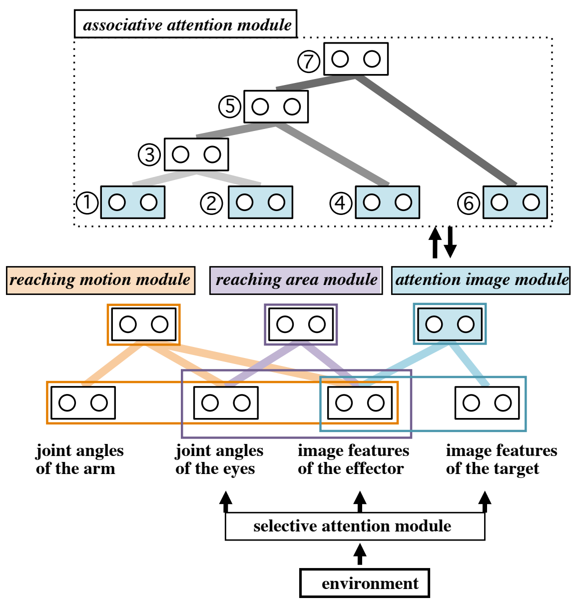


Fig. 4. Overview of layered network.

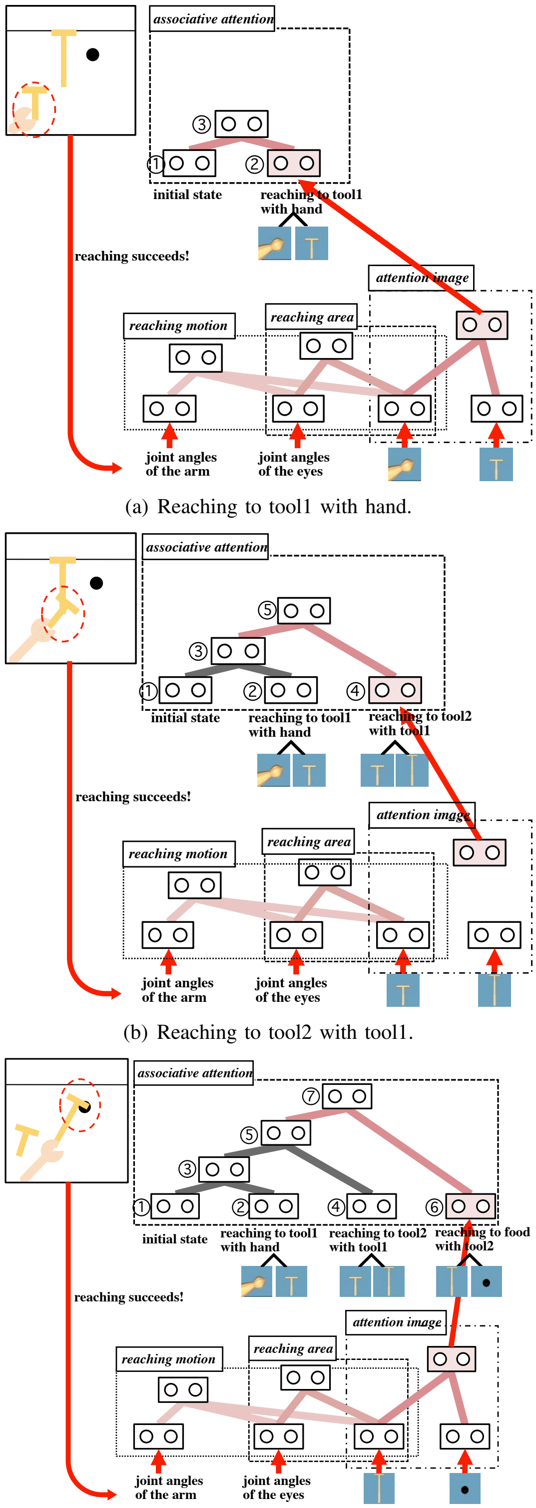
motion module (140 neurons (40 for eyes, 100 for image) for input layer and 30 neurons for output layer) associates the input data of the joint angles of the arm, those of the eyes, and the visual image of the end-effector when the agent attends to the reaching target. *The reaching area module* (230 neurons (40 for eyes, 100 for image, 90 for arm) for input layer and 85 neurons for output layer) associates the input data of the joint angles of the two eyes with image features of the end-effector when the agent explores the front space with the end-effector (hand or tool). Thus, after learning, the network information can be used to evaluate whether the attended position is reachable with the current end-effector. *The attention image module* (200 neurons (100 for effector image, 100 for target image) for input layer and 50 neurons for output layer) associates the image features of the end-effectors with the target objects during reaching. The training data are gathered when the hand is reached to objects (target and tools) that are put in various places on the table. Because the receptive field of neurons are not overlapped in this setting, the network could not interpolate the untrained data.

The upper layer network, *the associative attention module*, learns the temporal transition of the lower network. The hidden layer unit activation patterns in the attention image module are used as lower network representational data. For the upper layer of the network, the number of the neurons are 50 for input layer and 10 for hidden layer. Only the success data (reaching with (hand/short tool/ long tool) to (target/short tool/long tool)) are given as the training data. The sequential data of the lower layer network are input to the visual layer of the upper layer network depending on temporal order. For example, when an agent takes the food with a tool such as a rake, first he/she takes the tool with his/her hand, and then the food with the tool. In the first procedure, the activation patterns of the hidden layer of the attention image module in the initial state (no reaching) and reaching for the tool with the hand are input to the visual layer of the upper layer (① and ② in Fig. 4, respectively). Then, during the second tool reaching, the activation pattern of the attention image module, in which the image features of the tool and food are input to the end-effector and the target nodes, respectively, is input to the upper layer(④ in Fig. 4), and associated with the pattern of the hidden layer of ③.

B. Learning Reaching Sequences

In the associative attention module, a new RBM is composed layer by layer with the activation pattern of the hidden layer in the attention image module, until the final goal (reaching the food) is accomplished. These layers make up the nesting structure of the combinations of the end-effector and the target. In the following section, the details of the learning procedure are explained using the example shown in Fig. 5 (condition 4 in Fig. 3), in which a short tool exists in the area the agent can reach by hand, a long tool exists in the area in the area the agent can reach with the short tool, and the food exists in the area the agent can reach with the long tool.

First, the success of reaching the short tool by hand triggers the first module learning in the lower layer (the reaching motion module, the reaching area module, and the attention image module). That is, the lower layer where the joint angles of the arm, those of the eyes with which the agent focuses on the short tool, and the images of the short tool and the hand are associated. Then, the RBM in the upper layer is constructed with the visual layer of ① the hidden layer of the attention image module and ② the initial state in which all values are set to 0, and the hidden layer ③. Similarly, when the agent reaches for the long tool with the short tool, the second association learning is triggered in the reaching motion, the reaching area, and the attention image modules. The learning in the associative attention module is also proceeded with the visible layer of ④ the hidden layer of the attention image module, ③ the hidden layer of the first associative attention module, and the new hidden layer ⑥. When the agent finally reaches for the food with the long tool, the learning in the lower modules is triggered. A third RBM in the associative attention module is constructed with the visible layer of ⑥ the hidden layer of the attention image module, ⑤ the hidden layer of the second RBM in the associative attention module, and the new hidden layer ⑦.



(a) Reaching to tool1 with hand.

(b) Reaching to tool2 with tool1.

(c) Reaching to food with tool2.

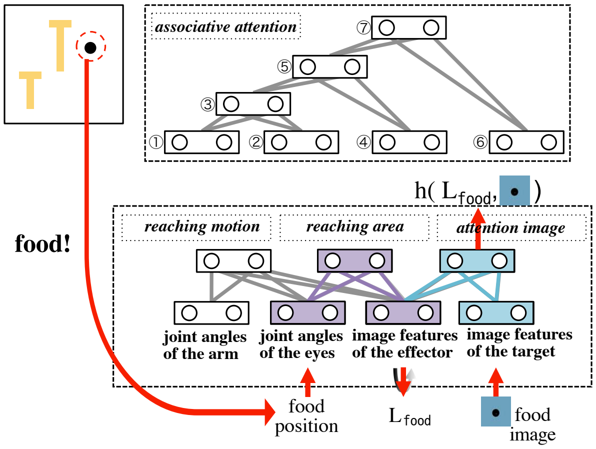
Fig. 5. Network procedures in learning phase.

C. Generating Sequential Reaching Motor Commands

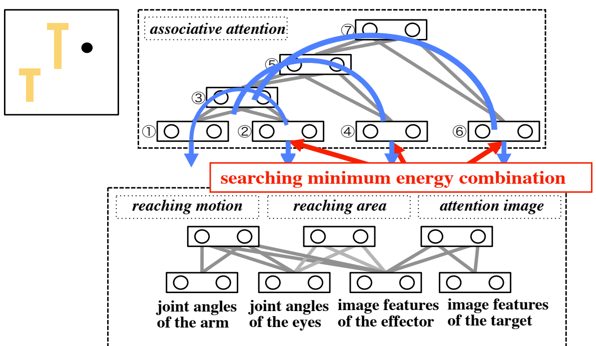
After the procedures for reaching are recorded in the layered network, it is important to note how the appropriate procedures are remembered with the layered associative network. In our network, the template for the end-effector and the target is prepared. In this framework, it is important to find the appropriate combinations of the end-effector and the target. Note that the same problem occurs in language recognition, in which we must decide whether the heard word is a verb or the target object, for example. In this subsection, we show how the network recalls the learning event and generates the appropriate motor commands. This is the same situation as in the previous subsection, in which a short tool, a long tool, and the food exist as shown in Fig. 6. First, among the observed objects, the image of the target (food) is selected and input to the visible layer of the reaching area module. In addition, the end-effector with which the agent could reach that position is reconstructed. Likewise, the images of the short tool and the long tool are input to the visible layer of the reaching area module, and the appropriate end-effector images for reaching are reconstructed.

On the other hand, the activation patterns in the hidden layer with inputs of images of the possible targets (short tool, long tool, and food) and the image of the end-effector that is reconstructed in the reaching area module are input in the visible layer of the attention image module.

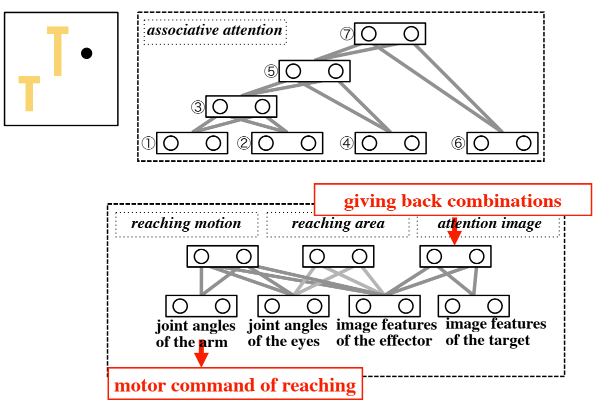
Then, the activation pattern of the hidden layer with the final target image is examined in the associative attention module. The pattern is input to ②, ④, and ⑤ in order, and the network energy (eq. (5)) is calculated. The network energy of the reconstruction pattern with the learned input data will be lower than that with the inexperienced data. For example, Fig. 7 shows the network energy of RBM of ⑤, ⑥, and ⑦ during training and reconstruction with the hidden layer of the attention image module when various combinations of end-effector and target image features are input. The graph shows that the network energy during reconstruction is lower for the same level of training. However, even though the network energy reconstructed data energy is lower, the data are declined if the reconstructed data are not found in the current environment. The exploration of the reaching procedures continues until the appropriate low energy combinations are found. Once the appropriate combinations are found, these datasets are transferred back to the lower layer network. The appropriate combinations are transferred to the hidden layer of the attention image module, and the necessary datasets for reaching, such as joint angles of the arm and eyes, are reconstructed in the three modules of the lower layer network. Figure 8 shows the generated reaching movements from the viewpoint of the robot under the various conditions corresponding to Fig. 3.



(a) Making combination of the attention images.



(b) Finding minimum energy state in the upper layer.



(c) Giving back biases to lower layer.

Fig. 6. Invoked network in control phase.

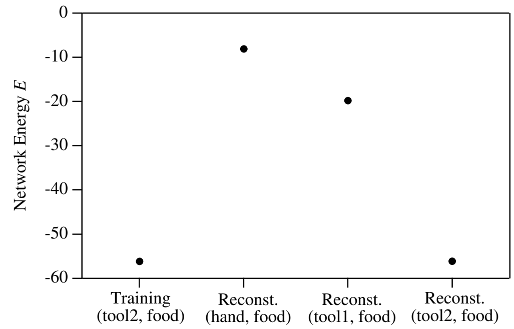
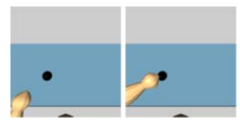
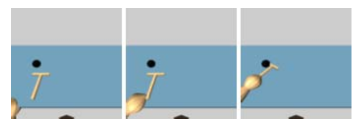


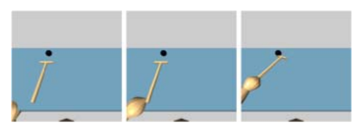
Fig. 7. Network energy with reconstruction data.



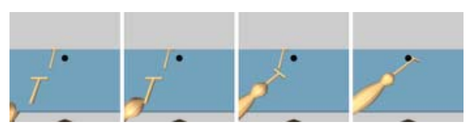
(a) Condition 1



(b) Condition 2



(c) Condition 3



(d) Condition 4

Fig. 8. Generated reaching movements under various conditions.

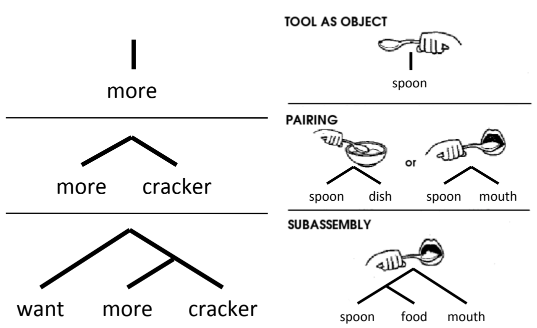
V. Discussion

In this paper, we proposed a hierarchical network that describes a sequential tool-use motion as multiple combinations of an end-effector and a target object, and generates appropriate combinations of an end-effector and joint angles for reaching. The proposed network makes use of the advantages of the restricted Boltzmann machine to generate situation-dependent motion. Learned data are generated from partly deficit data, and the distance from the current input data to the learned data is evaluated on the basis of the network energy. Another advantage of RBM is that the input data are self-organized in the hidden layer. In the proposed network, all modules are constructed with RBMs. Thus, information of all procedures to the final target is summarized as the activation pattern of each of the hidden layers in the upper network. The trained data in the hidden layer of the RBM are self-organized. This makes it possible to evaluate to what extent the current task is similar to the experienced task. The RBM reconstruction process can be utilized as the top down process to provide the bias with which the appropriate motor command is generated depending on the situations in the lower layer.

Several assumptions were made such that the whole network would work for autonomous learning and larger scale problems. Hereafter, we discuss some of the problems. First is the attention and working memory. In this study, the objects to be attended to or recognized are given in advance. The robot assigns these recognized objects to the meaning of the end-effector or the target. When attended objects are input to the network (image feature module), the object images are kept in temporal memory together with the position information (the joint angles of the eyes). When checking whether the remembered combinations are possible, the current possible combinations should be kept in the temporal memory. Thus, our system assumes that the temporal memory (or working memory) is stored at the input level of the network. In future research, we will attempt to combine this memory system with the existing working memory models [2] [3].

The second assumption concerns the activation control of the network. In the proposed network, the activation timing of the lower network and the higher network (e.g. Fig. 4) is given by the designer. Moreover, the network does not have a mechanism to check how long the current bias affects the lower layer. Because of the lack of the dynamics property in RBM, we did not treat the problem of how to control the dynamics in the hierarchical system. We are currently working on the design of an autonomous networkcontrol system in the hierarchical structure.

The third problem is the network structure. Greenfiled [4] and Molnar-Szakacs et al. [12] proposed that the complexity of the hand manipulation such as spoon and nesting cup relates to that of the acquired language. It is reported that the Broca area is activated in both tool-use and language tasks, although the details of the mechanism are unknown [6]. This suggests that the mirror neuron system plays an important role in the hierarchical structure of both language and tool-use. Figure 9 shows the hierarchical structure in terms of syntactic structure of language and spoon-use. Greenfield insists that "subassembly" mechanisms are common in both tasks. If the subassembly is interpreted as a temporal order of the combination of an end-effector and a target in a sequential procedure, the Greenfield model corresponds to the model proposed in this paper (Fig. 10).



(a) Hierarchical structure of language

(b) Hierarchical structure of spoon-use

Fig. 9. Development of hierarchical structure of language and tool-use (modified from Greenfield [4]).

However, it is not necessary that the nesting structure, such as a subassembly or a temporal procedure, should be realized with the network structure itself, as in the proposed network. The recurrent network is a possible structure that realizes the nesting information processing, although a plausible mechanism for additional learning is necessary [14].

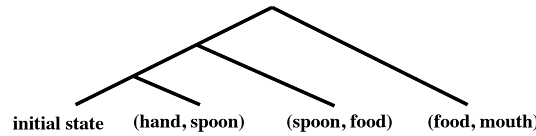


Fig. 10. Nesting structure of combinations of end-effector and target in spoon use.

As a next step, we will attempt to extend our model to the problem of syntactic processing and explore the common information processing mechanism of tool-use and language-use.

Acknowledgment

This work was supported by MEXT.KAKENHI 22700203 and 24000012.

References

[1] M. Asada, K. Hosoda, Y. Kuniyoshi, H. Ishiguro, T. Inui, Y. Yoshikawa, M. Ogino and C. Yoshida. "Cognitive developmental robotics: a survey." IEEE Transactions on Autonomous Mental Development, vol. 1, pp. 12-34, 2009.

[2] A. D. Baddeley. "The episodic buffer: a new component of working memory?" Trends in Cognitive Sciences, vol. 4, pp. 417-423, 2000.

[3] N. Cowan. "Working memory capacity," New York, NY: Psychology Press, 2005.

[4] P. M. Greenfield. "Language, tools and brain: The ontogeny and phylogeny of hierarchically organized sequential behavior." Behavioral and Brain Sciences, vol. 14, pp. 531-595, 1991.

[5] H. Head and G. Holmes. "Sensory disturbances from cerebral lesions." Brain, vol. 34, pp.102-245, 1911.

[6] S. Higuchi, T. Chaminate, H. Imamizu and M. Kawato. "Shared neural correlates for language and tool use in Broca's area." NeuroReport, Vol. 20, No. 15, pp. 1376-81, 2009.

[7] M. Hikita, S. Fuke, M. Ogino and M. Asada. "Cross-modal body representation based on visual attention by saliency." IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.2041-2046, 2008.

[8] G. E. Hinton, S. Osindero and Y. Teh. "A fast learning algorithm for deep belief nets." Neural Computation, vol. 18, pp. 1527-1554, 2006.

[9] A. Iriki, M. Tanaka and Y. Iwamura. "Coding of modified body schema during tool use by macaque postcentral neurones." Neuroreport, vol. 7, pp. 2325-2330, 1996.

[10] A. Maravita and A. Iriki. "Tools for the body (schema)." Trends in Cognitive Sciences, vol. 8, pp. 79-86, 2004.

[11] S. I. Maxim. "Body Image and Body Schema (edited by P. D. Helena)." John Benjamins Publishing Company, 2005.

[12] I. Molnar-Szakacs, J. Kaplan, P. M. Greenfield and M. Iacoboni. "Observing complex action sequences: The role of the fronto-parietal mirror neuron system." NeuroImage, vol. 33, pp. 923-935, 2006.

[13] C. Nabeshima and Y. Kuniyoshi. "A method for sustaining consistent sensory-motor coordination under body property changes including tool grasp/release," Advanced Robotics, Vol.24, No.5-6, pp.687-717(31), 2010.

[14] S. Nishide, T. Nakagawa, T. Ogata, J. Tani, T. Takahashi and H. G. Okuno. "Modeling Tool-Body Assimmilation using Second-order Recurrent Neural Network." IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5376-5381, 2009.

[15] E. Oztop, M. A. Arbib, and N. Bradley. "The development of grasping and the mirror system." In M. A. Arbib, editor, Action to Language via the Mirror Neuron System, Cambridge University Press, 2006.

[16] J. Tani and M. Ito. "Self-Rrganization of Behavioral Primitives as Multiple Attractor Dynamics: A Robot Experiment," IEEE Transactions on Systems Man and Cybernetics, vol. 33, pp. 481-488, 2003.

[17] D. M. Wolpert and M. Kawato. "Multiple paired forward and inverse models for motor control." Neural Networks, vol. 11, pp. 1317-1329, 1998.