Neurocomputing 72 (2009) 1910–1919

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Incremental learning of sequence patterns with a modular network model

Ichiro Igari *, Jun Tani

Brain Science Institute, RIKEN, 2-1 Hirosawa, Wako-shi, Saitama 351-0198, Japan

ARTICLE INFO

Article history:
Received 24 November 2006
Received in revised form
31 March 2008
Accepted 18 May 2008
Communicated by R. Kozma
Available online 16 July 2008

Keywords:
Dynamic systems approach
Motor learning
Incremental learning
Recurrent neural network
Parametric bias
Mixture of experts architecture

ABSTRACT

The relationship between generalization and differentiation fluctuates depending on the ongoing context, which is extracted by the current adaptive capability of the learner. In the present report, we numerically examine the relationship between generalization and differentiation using a novel connectionist model. The simulation results of incremental learning indicate that the newly added sequence modifies the previously learned internal representations in a different manner, depending on the inconsistency with the preceding task. This observation supports our assertion that it is fundamentally important to investigate how the transition dynamics of learning toward a goal affects the finally acquired structure of the learner.

1. Introduction

The incremental learning process is essential for developing organisms. In some cases, the incremental learning process exists as the developmental change of behavior. In other cases, it is regarded as new skill acquisition. The characteristic time scale of incremental learning ranges from a few seconds to dozens of years. Adaptation to an unknown and fluid environment continuously modifies how each individual reacts to the world.

The transformation of acquired behavior and skills has interested researchers in various fields, such as developmental psychology, psycho-physics, linguistics, ethology, developmental robotics, machine learning theory, and artificial intelligence. Some of these approaches emphasize a phased development or phased strategy change, as criticized in [12,16]. According to the perspective of the approach, if a small number of effective parameters can be extracted to characterize the skill for each phase, then the main interest will become the specification of the timing for switching the phase and the investigation of the basis of each behavioral function. However, recently, we have observed numbers of experiments and numerical simulations that highlight a gradual aspect of development rather than phased transition in learning. In the following, several studies are cited in relation to these observations.

1.1. Evidence from psycho-physics experiments

The experimental design of our computational study is directly inspired by psycho-physics research on consolidation [2,11,17]. In order to make our motivation clear and to understand the difference between incremental concept of these psycho-physics experiments and our numerical experiment, we briefly survey the previous studies.

A previous study [2] reported that after 4 h, the skill of learners at motor task *P* was not disrupted when the learner learned another task *Q*. This depends on the interval time length, which determines whether the internal model for initial task *P* has been consolidated.

Another experiment [11] examined whether the passage of time makes the internal model less fragile. The participant was asked to make rapid reaching movements to a series of targets with a robotic hand. A force field is applied by the torque motors of the robotic arm during the task. The previous authors found that for a few hours after the completion of practice, the internal model became less fragile with respect to behavioral interference. They also investigated positron emission tomography scans in the task, and as a result suggested that within 6 h after the completion of practice the representation of a motor skill is reorganized in the brain.

In another previous study [17], the finger tapping motor skill task was performed. The participants were instructed to press four numeric keys using their fingers following a five-element sequence presented to them repeatedly. The authors found that waking reactivation can return a previously consolidated memory

* Corresponding author. Tel.: +81 48 462 1111x7417; fax: +81 48 467 7248.
E-mail addresses: igari@brain.riken.go.jp, igari@bdc.brain.riken.go.jp (I. Igari), tani@brain.riken.go.jp (J. Tani).

0925-2312/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.05.002

to a fragile state, making it susceptible to interference. They argued the different forms of consolidation corresponding to waking and sleeping reactivation during the interval time.

At present, it is clear that simply asking "WHAT is given to the learner after completion of training?" is not sufficient for the prediction of whether the previously consolidated skill can be interfered or improved. The above psycho-physics experiments indicate that in order to determine the "stiffness" of an acquired skill the essential question is "HOW is the task presented?" These studies are good examples suggesting that previously learned functional modules should be modified when they are activated in some cases. The modeling of the present study is applied under the same premise, in which the history-dependency of incremental learning is regarded as crucial.

1.2. A computational model: the mixture of experts system

In the following, we briefly describe the "mixture of experts" type architecture [5,6,18] as a part of the theoretical background of the present study. This architecture supplies one of the computational frameworks by which to implement differentiation in learning.

The mixture of experts consists of learning units and a gating mechanism. At the end of successful learning, each lower-level unit works as an expert for a specific task and only one gate for the best unit dominates the entire system dynamics during the task. As a result, the system finally grows locally represented knowledge for each skill within a lower-level unit.

For instance, this method was applied to a mobile robot experiment for a sensory-motor online navigation task [15]. In the numerical experiment, the two-wheeled agent learns to move through a two-dimensional maze with the recurrent neural network (RNN) controller as a lower-level unit. Finally, the agent can automatically open and close suitable gates corresponding to lower-level units by detecting different types of the ongoing environment (i.e., wall configuration) in an online manner. In this case, only one lower-level unit governs the resulting agent motion and the dominant unit is sometimes switched when a significant environmental change takes place. This behavior is called "winner-takes-all" gating dynamics, which is typically observed for a successfully designed mixture of experts system.

Another study evaluated the mixture of experts system by applying it to analyze the result of a visuomotor learning task by human subjects [4]. The subjects were exposed to opposite prism-like visuomotor remappings on a two-dimensional monitor screen, which generates conflicting pairing between the visual and motor spaces. The authors showed that the mixture of experts framework successfully gives a quantitative account of the modular decomposition of learned mappings.

The goal of the present study is to build a plausible numerical model of an agent that extracts the essence of human incremental learning ability. Unfortunately, for this purpose, we cannot simply apply the mixture of experts because of the "inflexibility" of lower-level units. In a sense, lower-level units of this architecture can be self-organized, providing a seemingly useful tool. However, in previous studies, whether the inflexibility is assumed to be explicit or implicit, the theoretical frameworks share the same principle: once a lower-level unit has acquired a suitable function for the task, the unit (one of the "experts") stops learning and the structure should be protected from the other tasks. When we consider the incremental learning paradigm within the architecture, the previously acquired skills are assumed to be frozen and stable with respect to tasks that are given afterward.

The previously described experimental results of psycho-physics [2,11,17], however, provide a striking contrast to the above-mentioned principle. These experiments revealed that whether the well-organized modular motor skill becomes plastic depends on time spent in specific brain states between training and reactivation events. Thus, the mixture of experts type system is a good candidate for modeling the modular nature of skills or memory, although it is incompatible with the learning module plasticity, which allows each functional module to be modified even after completion of the first skill training. The difference between the present model and the original mixture of experts will be explained in the following two sections.

1.3. The parametric bias method

The RNN has a variety of applications and is attractive as an apparatus to deal with tasks that include time series sequences. In the present report, the RNN is considered only with regard to its capacity as a prediction device for short periodic time sequences [10]. An RNN that has learned successfully can sequentially output predicted motor values by supplying motor values of the current time step. However, there are two methods by which to evaluate the degree of success in learning a time sequence.

One method involves the summation of predicted output errors over the sequence by supplying "correct" input values in the teacher signal. This method is referred to as an "open-loop condition" method. Another method that is similar to the open-loop condition, but which substitutes the predicted output of the previous time step for the input value of the current time step. This method is referred to as a "closed-loop condition" method. Closed-loop learning is more difficult for a network than open-loop learning because a very small deviation in the predicted output at an early time step in a sequence might develop into a large error from the ideal trajectory after several steps.

A single RNN can potentially predict a variety of sequences if correct input (i.e., input having exactly the same value in the teacher signal sequences) is given at every time step. Back-propagation through time (BPTT) algorithm [10] enables an RNN to acquire several trajectories in terms of the open-loop condition. However, in the present model, a more stringent condition is assumed in the prediction process. That is, the predicted sequence should be calculated by iteratively giving the output value of the previous step as the input of the next step. This self-looping of the prediction process is naturally required by the human cognitive ability, so that we, as humans, can internally recollect and follow a series of events or movements that was experienced previously.

Under this closed-loop condition for the prediction process, an RNN usually fails to self-organize with the existing gradient-descent method when the number of time series patterns is submitted for learning, because the task implies that the achieved RNN should form a multiple-limit-cycle-attractor state, which is difficult to self-organize. The parametric bias (PB) method has been developed to improve the RNN capability for multiple time series patterns [14,13]. The basic concept of the PB method is that PB values, being maintained constant over a sequence, provide the information needed in order to differentiate between sequences. Note that the PB value itself is self-organized simultaneously with the other network parameters and the gating profile. With the help of the PB method, a single RNN can easily construct internal models for more diverse time-dependent patterns.

1.4. The model design policy

From a model study standpoint, there are a number of attractive approaches to the incremental learning issue that are similar to the present assumption regarding the plasticity of the learner. For example, Kuniyoshi and Berthouze [8,1] discussed the importance of the developmental aspect in building sensory-motor coordination of a vision robot. Kaplan and Oudeyer [7,9] proposed a learning system driven by an internal novelty reward, which is dependent on the current internal state. They applied this learning system to a mobile robot that explored an artificially designed environment. However, in the present study, we concentrate on the history-dependency of motor learning, rather than on the self-organization of sensory-motor coupling or the novelty reward.

Our simulated agent and training environment do not have a consolidation factor, but rather partly share the incremental learning setup and periodic nature of the motor task with real experiments for human subjects [2,11,17]. This fact leads our numerical experiment to an abstraction of reconsolidation process rather than consolidation process [3]. We designed a model to satisfy the requirements of (i) modularity for learnable units and (ii) rewritability of the learnable unit. Below, we explain how these two assumptions take shape in the current context.

The concept of modularity of skill learning (i) is not new and has often been assumed in artificial intelligence studies and other machine learning theories. According to this concept, each learnable unit is attributed to each skill or knowledge after successful acquirement. The proposed model follows the general framework of the modular learnable unit, but has a critical difference from the traditional models in the sense that the proposed machine does not have an ideal state into which to converge. The acquired structure can be stable, although the state is not regarded as optimization for a particular task set—this implication in the presented model is introduced by the above rewritability assumption (ii).

If we assume that modular learnable units can be reorganized at any time, even after successful skill acquirement, the agent uses a single method to "differentiate" and "generalize" incoming stimuli. In order to clarify this method, we consider the simple case in which an agent has first learned sequence A_1, followed by sequence A_2. The agent is considered to "differentiate" sequences A_1 and A_2 if two different learning modules acquired these sequences separately. Similarly, the agent is considered to "generalize" sequences A_1 and A_2 if one of the learning modules, which has learned sequence A_1, also acquires sequence A_2.

This definition implies that generalization and differentiation are two different methods of accepting the newly exposed stimulus A_2 for the agent. Note that the previously established module for sequence A_1 works as a standard when the entire system infers whether sequence A_2 should be grouped into the same category as those for A_1. In the same manner, we can imagine other succeeding sequences $A_3, A_4, \ldots$ after sequence A_2. At any time, the previously acquired structure of learnable modules always determines how to differentiate/generalize the currently facing stimulus.

The border between the differentiation and the generalization processes is clear if the designer of the model gives a fixed definition of the categories for a sequence. In the example discussed above, the border fluctuates depending on the accumulation of past experience. In this sense, the method of differentiation or generalization is not given a priori, but is indivisibly embedded in the never-converging modular structure of an agent.

In the preceding two sections, we described the mixture of experts architecture and the PB method as a theoretical framework. By combining these two methods, we can enhance the generalization capability of the learnable module in the system. This novel architecture is introduced at the request of the model design policy discussed in this section and the incremental learning task setup described in the following section.

2. Learner model

2.1. Assumed situation

In the present numerical model, we assume that there is a subject (an agent) who is asked to learn groups of motor sequences. The motor sequence is given as the coordinates of a small circle on a monitor screen. The circle moves only horizontally for a short time. The subject has to predict the position of the circle at the next step. The predicted position is indicated by moving a mouse cursor to the position before the next circle appears. Namely, this is a trajectory tracking task in one-dimensional continuous space in discrete time.

The agent is modeled by a neural network and motor sequence patterns are set as a teacher signal. The learning system consists of lower-level networks combined by a gating module (see Fig. 1). The experiment is simulated in a supervised learning framework.

2.2. Lower-level network with PB

The lower-level network (the subnet) is a three-layer RNN with PB inputs (Fig. 2). The network accepts one-dimensional time series $X(n)$ and PB vector $p_q^i(k)$ as input, where n is the time step, i denotes each subnet, q is a sequence identifier and k is a node number. The context units in the current step are copied to the input in the next step. We apply the sigmoid function in updating each unit in order to limit the unit value. Each subnet can receive an input sequence regardless of the gate opening. The output unit value $y^i(n+1)$ is regarded as a predicted position of the next step. The numbers of PB units, mid units, and context units are 2, 5, and 3, respectively. It is assumed that the PB unit remains constant during a sequence. Note that the subnet independently holds the PB vector set, in which each PB vector is attributed to each sequence in the entire task. Along with other network weights, the PB vector is trained by maximizing the likelihood function, which will be described in detail in the following section.

2.3. Mixture of experts system and gate function

The mixture of experts architecture accepts a one-dimensional time series $X(n)$ as an input and outputs $Y(n+1)$ as a predicted position in the next step:

$$Y(n+1) = \sum_{i=1}^{N^{\mathrm{subnet}}} g_q^i \cdot y^i(n+1), \tag{1}$$

where N^{subnet} is the number of subnets and g_q^i denotes the gate opening value of subnet i for the q-th sequence. In the following experiments, we set $N^{\mathrm{subnet}} = 2$. Gate opening g_q^i is given by the soft-max activation function:

$$g_q^i = \frac{e^{s_q^i}}{\sum_{j=1}^{N^{\mathrm{subnet}}} e^{s_q^j}}, \tag{2}$$

where s_q^i denotes a gate activation variable. The proposed system has a gating profile, which supplies the gate activation s_q^i for each subnet. Note that this profile is attributed to each sequence and the gate activation is kept constant during a single sequence.

In order to update the network weights of a subnet, the gating profile and the PB, the likelihood function $\ln L_q$ for the q-th

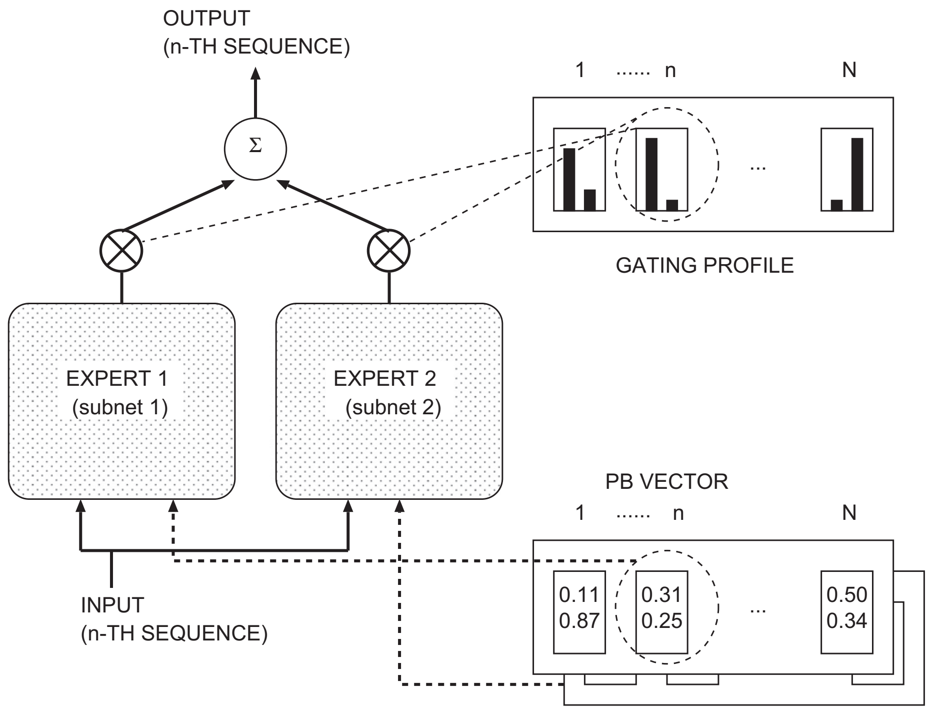


Fig. 1. System architecture of mixture of experts with parametric bias.

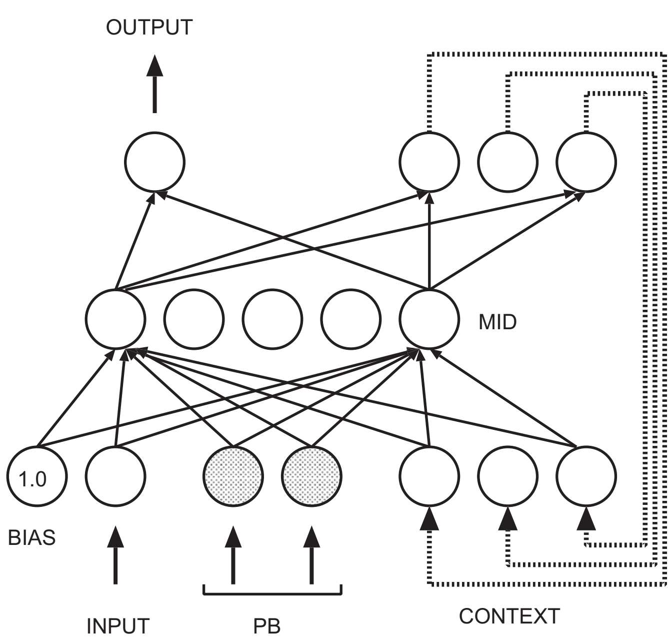


Fig. 2. Subnet architecture. The PB vector input is given as constant throughout a sequence.

sequence is introduced by the following formulation in [5]:

$$\ln L_q = \ln \sum_{m=1}^{N^T} \sum_{i=1}^{N^{\text{subnet}}} g_q^i \cdot \exp\left[\frac{-|y^*(m+1) - y^i(m+1)|^2}{2\sigma^2}\right], \tag{3}$$

where N^T is the length of a sequence, $y^*(m+1)$ is the teacher signal for m-th input, and σ is the exclusiveness parameter.

The proposed system performs learning by following the likelihood maximization principle. Update rules for the network connection weight w_{kl}, the PB vector, and the gate activation variable are given by

$$\delta w_{kl} = \varepsilon_w \frac{\partial \ln L_q}{\partial w_{kl}} + \xi_w \cdot \delta w_{kl}^{\text{prev}}, \tag{4}$$

$$\delta p_q^i(k) = \varepsilon_p \frac{\partial \ln L_q}{\partial p_q^i(k)} + \xi_p \cdot \delta p_q^i(k)^{\text{prev}}, \tag{5}$$

$$\delta s_q^i = \varepsilon_s \frac{\partial \ln L_q}{\partial s_q^i} + \xi_s \cdot \delta s_q^{i\ \text{prev}}, \tag{6}$$

where ε_w, ε_p, and ε_q are learning rates, ξ_w, ξ_p, and ξ_s are inertia coefficients, and $\delta w_{kl}^{\text{prev}}$, $\delta p_q^i(k)^{\text{prev}}$, and $\delta s_q^{i\ \text{prev}}$ are updates for the previous training step. Actual computation of these updates is performed using the BPTT algorithm. In general cases, the teacher signal set includes more than one sequence, and these updates are averaged over sequence q. The delta rule is iteratively applied in the training process.

3. Experimental results

Two numerical experiments are designed to apply our network architecture to different learning situations. First, the basic behavior of the learning system is investigated. Then, we discuss the effect of incremental learning of motor sequences.

3.1. Experiment I: fixed motor sequence set learning

The teacher signal set shown to the agent consists of a variety of one-dimensional sequence patterns. In this experiment, the target patterns are fixed and do not change throughout the learning process.

3.1.1. Definition of task group

Each sequence pattern is periodic: an eight-period one-dimensional sequence is repeated four times to form one

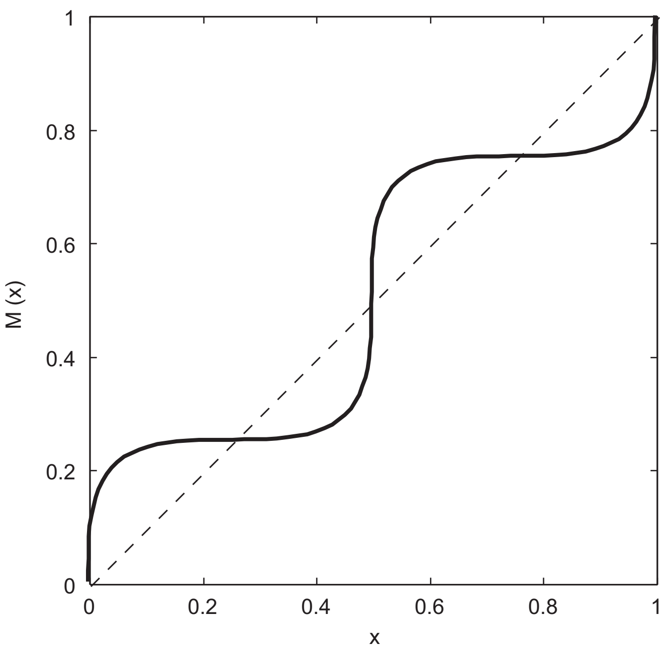


Fig. 3. Example of modification function $M(x)$ (solid line). In this figure, the parameters a and n_p are set as 1.0 and 4, respectively.

sequence having 32 points. A task group is generated by applying the modification function $M(x)$ to a particular prototype trajectory, $X_i(n)$, where i and n denote the sequence number and time step, respectively. In addition, $M(x)$ is a sinusoid function rotated about the origin and defined by

$$\frac{1}{\sqrt{2}}(-x+M(x)) = \frac{2\sqrt{2}a}{\pi n_p}\sin\left\{\frac{\pi n_p}{2}(x+M(x))\right\}, \tag{7}$$

where the parameters are limited to $-1.0<a<1.0$, $n_p = 1, 2, \ldots$. A modified sequence is generated by substituting one prototype $X(n)$ into x of $M(x)$. One example of the modification function $M(x)$ is shown in Fig. 3. By fixing n_p to be constant and changing parameter a gradually, we obtain a similar yet distinct series of sequences.

The entire teacher signal consists of two task groups generated by different prototype trajectories.

3.1.2. *Introduction of distance measure between two sequences*

How can we characterize the entire teacher signal? In this simulation, there are two different task groups, and the members of each task group share the same prototype trajectory within a group, thus quantifying both the similarity between two groups and the similarity within a group is feasible.

Here, we introduce the distance measure for two sequence patterns given by Eq. (8). The definition reflects that the structure to be learned by an agent is periodic and is not related to the initial point in a sequence, e.g., the distance between the period-3 sequences $(0.1, 0.2, 0.3, 0.1, 0.2, 0.3, \ldots)$ and $(0.2, 0.3, 0.1, 0.2, 0.3, 0.1, \ldots)$ should be zero:

$$D_{ij} = \min\left\{(1/L_s)\sum_{n=0}^{L_s-1}|x_i(n) - x_j(n+m)|;\quad m = 0, \ldots, (L_s-1)\right\}, \tag{8}$$

where i and j specify sequence numbers and L_s denotes the periodicity of the sequence ($=8$). In measuring D_{ij}, the averaged Euclidean distance is calculated for each phase offset m. When a particular m_0 results in the smallest distance, these two sequences should be compared by sliding the pattern $x_j(n)$ by m_0.

Using the above distance measure, two values are calculated: the maximum distance within a task group $D_P^{\text{max\_within}}$ and the minimum distance between two groups $D_{PQ}^{\text{min\_between}}$, where P and Q denote task groups:

$$D_P^{\text{max\_within}} = \max\{D_{ij}; i, j \in P\}, \tag{9}$$

$$D_{PQ}^{\text{min\_between}} = \min\{D_{ij}; i \in P, j \in Q\}. \tag{10}$$

These computed values are useful for characterizing an entire teacher signal. Suppose that the entire teacher signal consists of two task groups, A and B. If $D_{AB}^{\text{min\_between}} - D_A^{\text{max\_within}}$ is a negative value, then it is difficult for an agent to perfectly separate group A from B. Since some of the members in group A, there are more similar sequences in group B than the most dissimilar sequence in group A. Thus, two values, $D_{AB}^{\text{min\_between}} - D_A^{\text{max\_within}}$ and $D_{AB}^{\text{min\_between}} - D_B^{\text{max\_within}}$, reflect how clearly these two task groups are divided in terms of the distance measure D_{ij}.

3.1.3. *Task procedure and simulation results*

In Experiment I, we investigate how the proposed learning system works for a fixed teacher signal set. Fig. 4 shows the prototype trajectories for task groups A and B. The entire teacher signal is given by applying Eq. (7) to the two prototypes and does not change throughout the task. Each task group consists of 11 sequences. The task is batch learning, and all of the sequences in a set are shown equally to the learner.

Typically, the learning of task groups is performed as follows. At first, the gate profile is set provisionally, based on the current error of each network, which means that it is based on badly organized network weights and the PB vector. As the network weights and the PBs grow systematically, the gate profile tends to stabilize. The gate learning is unstable as long as each network is not supplied with consistent teacher signals. The instability of the gate profile can give rise to a catastrophic change in the network growth, and vice versa. Generally, in the final stage, self-organization of the gate profile precedes that of each network weight and PB vector. In this way, a subnet chooses which sequence to learn based on the current predictability by the opening gate, and these processes are unified in the likelihood function.

Next, we consider the conditions for successful learning. The agent is required to simultaneously generalize a variety of trajectories centered around one prototype and to differentiate members of two task groups. As stated earlier, consideration of the distance between/within groups helps to analyze this point.

By fixing n_p to be one and gradually changing the maximum value of $|a|$ in Eq. (7), we obtain sets of entire teacher signals that have different $(D_{AB}^{\text{min\_between}} - D_A^{\text{max\_within}}, D_{AB}^{\text{min\_between}} - D_B^{\text{max\_within}})$ pairs. The maximum values of $|a|$ for task groups A and B are set as $(0.0, 0.1, \ldots, 0.9)$ for each group, and $10 \times 10 = 100$ entire teacher signal sets are tested. In this simulation, the parameters are set as follows. The exclusiveness parameter for the likelihood function is $\sigma = 0.180$. The learning coefficients ε_w, ξ_w, ε_p, ξ_p, ε_s, ξ_s are set to $0.06, 0.9, 1.14, 0.82, 0.24$, and 0.12, respectively. The weight of each subnet is limited within the range of $[-5.5, 5.5]$. The gate activation value s_q^i is limited within the range of $[-4.0, 4.0]$.

Fig. 5 illustrates the success rate diagram, and for each teacher signal set, the success of the learning is enumerated starting from 100 different initial conditions for the network weight. In each learning process, 1500 training steps are iterated.

We regard a particular learning as being successful when the following conditions are satisfied: (i) subnet 1 becomes an expert

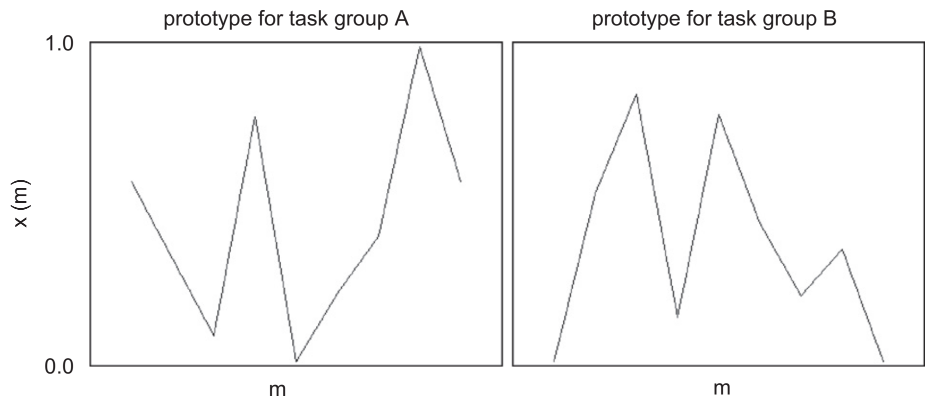


Fig. 4. Prototype trajectories for two task groups.

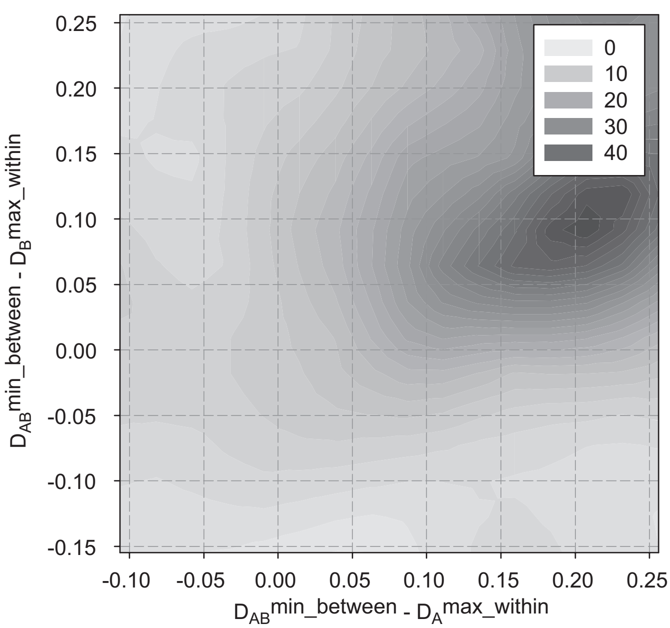


Fig. 5. Learning success rate diagrams for different task groups. The number of successes out of 100 initial conditions is plotted. The x- and y-axes correspond to the ease of grouping for task groups A and B, respectively.

of task group A (or B) and subnet 2 becomes an expert of task group B (or A) and (ii) all of the gate opening values for the "winner" sequence are greater than 0.85. The numerical result supports the previous speculation on separation of task groups. The successful learning rate is higher when parameters $D_{AB}^{\text{min\_between}} - D_A^{\text{max\_within}}$ and $D_{AB}^{\text{min\_between}} - D_B^{\text{max\_within}}$ are both large positive.

In addition, an asymmetry exists between task groups A and B in the success rate diagram, which can be explained by the learning-algorithm-specific and network-structure-specific factors. If we randomly generate two prototype sequences, there is a slight difference in the learnability of the sequences. This factor is not controlled in this simulation. However, the asymmetry can be neglected in the present discussion.

Here it should be noticed that low success rate (no more than 50% at any point) shown in Fig. 5 does not mean low convergence rate. For any teacher signal set chosen from Fig. 5, at least 90 of 100 trials converge, achieve high likelihood value, and satisfy the aforementioned condition (ii). In this case, the learning process finishes successfully in terms of maximization of likelihood function. This shows one of the advantages of the combination of mixture of experts and PB architecture. Comparatively low success rate in Fig. 5 just reflects that the way the agent separates sequences into two groups is sometimes different from what an experimenter expect.

3.2. *Experiment II: incremental learning*

In the following experiment, the structural change of the system in incremental learning is investigated. The teacher signal set shown to the agent is switched with another set during the learning process.

3.2.1. *Task design*

Fig. 6 illustrates the experimental design forming three stages.

In the first session, a teacher signal set is provided to the agent in the same way as in Experiment I, and training is performed 1500 times.

In the second session, another teacher signal set is prepared. Task groups Ax and Bx ($x = 1, 2$) indicate the task groups for the x-th session. In Fig. 6, the teacher signal sets for both sessions are shown. Task groups Ax and Bx share the same prototype trajectory. In generating the task group for the second session, the maximum value of parameter $|a|$ in Eq. (7) is the same as that for the first session. The critical difference in constructing task groups $A2$ and $B2$ is the parameter $n_p : n_p = 2$ for $A2$ and $n_p = 1$ for $B2$. As n_p increases, the generated trajectories become more twisted. At the twisted point, the new trajectory can change its shape from a decreasing function to an increasing function, and vice versa. As a result, higher inconsistency with the previous task is introduced in task group $A2$, compared to task group $B2$. Task group $B2$ appears to be identical to $B1$, although different trajectories are chosen.

Just before beginning the second session, only the gate profile and the PB acquired in the first session are eliminated. The weight for each network is preserved and carried over in the second session. This assumption is derived from the fact that the network weights grow slowly and represent the basic ability to perform the task. In contrast, the PB vector and gate profile grow quickly and represent the ability to adapt to the ongoing task using current abilities.

After obtaining the PB vector and the gate profile, all of the parameters are once again self-organized for the remainder of the session.

The third session is prepared for retesting, and learning of the agent is not performed. The purpose of this session is to quantify how the previously acquired performance for task groups $A1$ and $B1$ improves or deteriorates after the incremental learning.

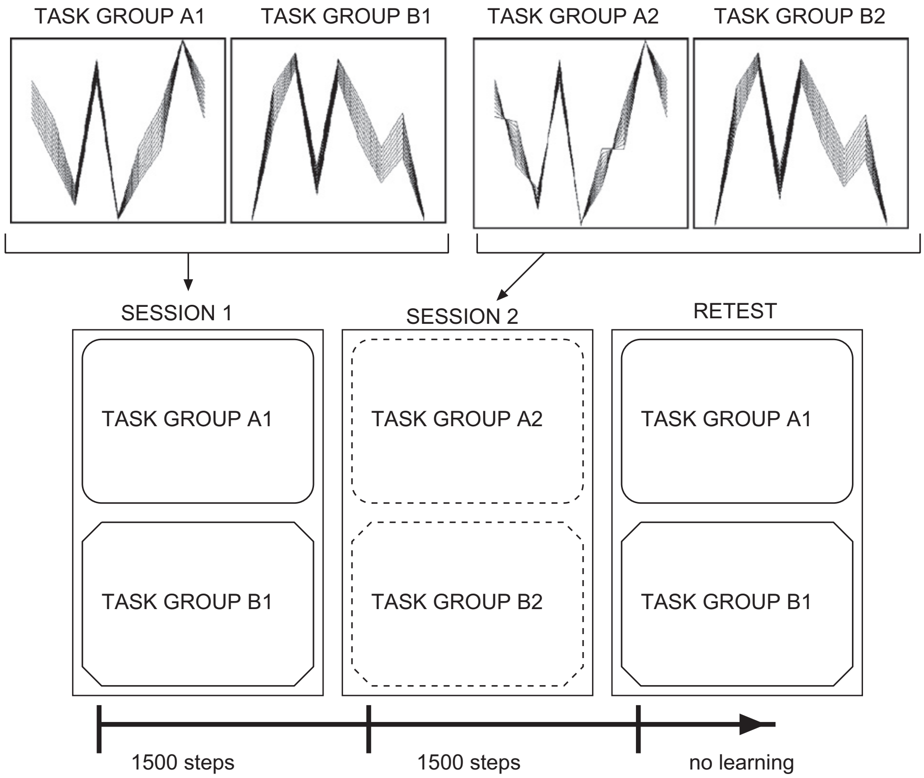


Fig. 6. Three-stage task design of Experiment II. The unit sequences shown at the top of the figure are iterated four times and form a periodic sequence. Only the first two sessions include the learning process. In the RETEST session, the system parameters are fixed and the error for the task in the first session is estimated.

3.2.2. *Simulation results*

In this experiment, by measuring the distance between every sequence pair in generated task groups, we get the following values: $D_{A1}^{\text{max\_within}} = 0.122$, $D_{B1}^{\text{max\_within}} = 0.153$, $D_{A1B1}^{\text{min\_between}} = 0.231$ for session 1, and $D_{A2}^{\text{max\_within}} = 0.137$, $D_{B2}^{\text{max\_within}} = 0.152$, $D_{A2B2}^{\text{min\_between}} = 0.236$ for session 2. The other parameters for learning are the same as those in Experiment I.

Fig. 7 represents the record of learning process in the first and second sessions. The horizontal axis of all plots in this figure corresponds to the training time step.

In Fig. 7, the likelihood, average likelihood and total likelihood are renormalized and plotted within the range of [0.0, 1.0]. The leftmost likelihood plots show the likelihood value for each sequence, which is acquired by each subnet. The average likelihood plots show the likelihood value for each sequence, which is averaged over two subnets. The total likelihood plot indicates the likelihood value, which is defined by Eq. (3). For instance, when one of the subnets takes a higher value than another subnet in the leftmost likelihood plot, the former subnet is expertized for the particular sequence. The total likelihood plot reflects the success of the entire system in acquiring the task set. The output error of each subnet for each sequence is also plotted in Fig. 7. This error plot has one-to-one correspondence to the likelihood plot for each sequence. The gate opening plot shows the change of the gate opening g_q^i in Eq. (2).

Until the end of first session, the entire system successfully learns the task set by the expertization of subnet 1 for sequences 0–9, and subnet 2 for sequence 10–19. Here, note that the sequence is numbered systematically for convenience. However, the learner (the modular network model) is not exposed to any cue that indicates which sequence belongs to a particular task group.

When the second session starts at a training time = 1500 steps, the likelihood first declines, but soon after, the gate profile returns to the previous value learned in the first session and the likelihood again increases. This "recovery" process happens within a small number of training step and does not cause a drastic change in the network weight. Thus, the system preserves the capabilities of task groups $A1$ and $B1$, even after finishing the second session.

Self-organization of the PB vector is important for a single subnet to adapt to multiple sequences [14]. When a subnet has a very small gate opening for a particular sequence, its corresponding PB vector does not affect the prediction result. Fig. 8 shows the self-organized values of PB units after the second session. Note that different vectors are set for each sequence, which produces better likelihood, as shown in Fig. 7. In addition, when the gate for subnet i ($i = 1$ or 2) is opening for a particular sequence, only the self-organized PB vector for subnet i effectively works in the entire system, and the PB vector for another subnet is not related to the result of prediction task.

The numerical result of retest in the third session is shown in Fig. 9 and is compared with the errors at the end of first and second sessions. The error is the RMS error estimated within each subnet and is not related to the gate opening. In these two learning sessions, subnets 1 and 2 become experts of task groups Ax and Bx, respectively. First, it is observed in Fig. 9 that the error of subnet 1 for task group Bx and the error of subnet 2 for task group Ax are large, that is, they are greater than 0.1 in all three sessions. However, in practice, these bad performances do not affect the following discussion because such an unsuitable subnet is blocked by the gating system.

It must be emphasized that the error in the retest session shows a clear contrast between the reorganization of subnets 1 and 2. When we observe the performance of subnet 1 for task group $A1$, the errors increase at the retest of task group $A1$ after incremental learning of $A2$. On the other hand, the errors for task group $B1$ decrease after the learning of $B2$.

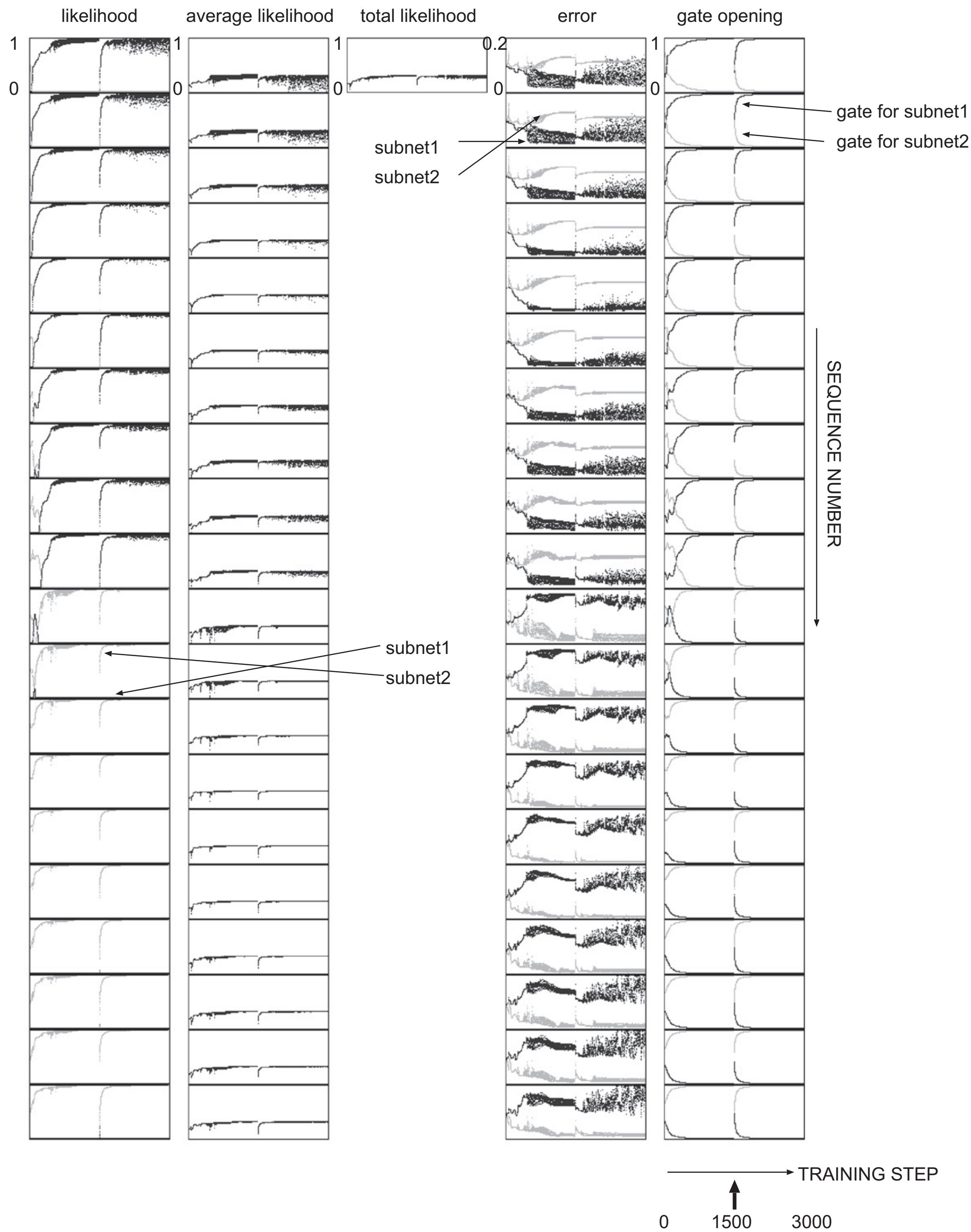


Fig. 7. Likelihood, error, and gate opening for each sequence with respect to training time. In the likelihood plots, the error plots, and the gate opening plots, black lines indicate corresponding values for subnet 1 and gray lines indicate corresponding values for subnet 2.

This observation is interpreted as the system modifying its structure differently when it adapts to the task group changes of ($A1 \to A2$) and ($B1 \to B2$). In the second session, the newly provided teacher signals work (i) to preserve the method by which to differentiate two task groups by two subnets and (ii) to interfere with how to generalize sequences in task group $A1$ (by subnet 1) and to preserve how to generalize sequences in task group $B1$ (by subnet 2). This result is due to the difference in the inconsistency between the task change for Ax and that for Bx, as previously introduced in the task design section.

4. Concluding remarks

In this report, we have presented a novel architecture based on the mixture of experts model and the PB method that demonstrates the generalization and differentiation capabilities for motor sequence learning. A distance measure for task groups is proposed in order to quantify the difficulty balance between grouping and separating task sets. The incremental learning experiment demonstrates that the newly assigned task group interferes with the previous learning in one case and preserves it

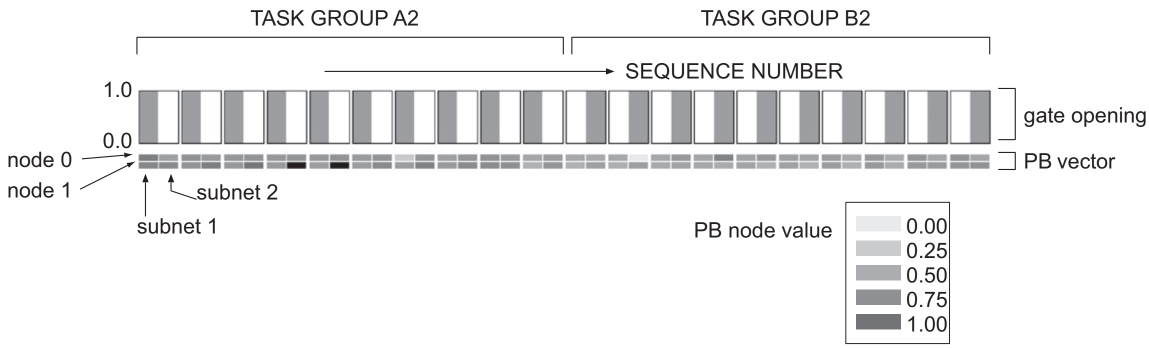


Fig. 8. Self-organized gate profile and independent growth of PB vectors after the second session. Acquired PB vector values are represented in gray scale.

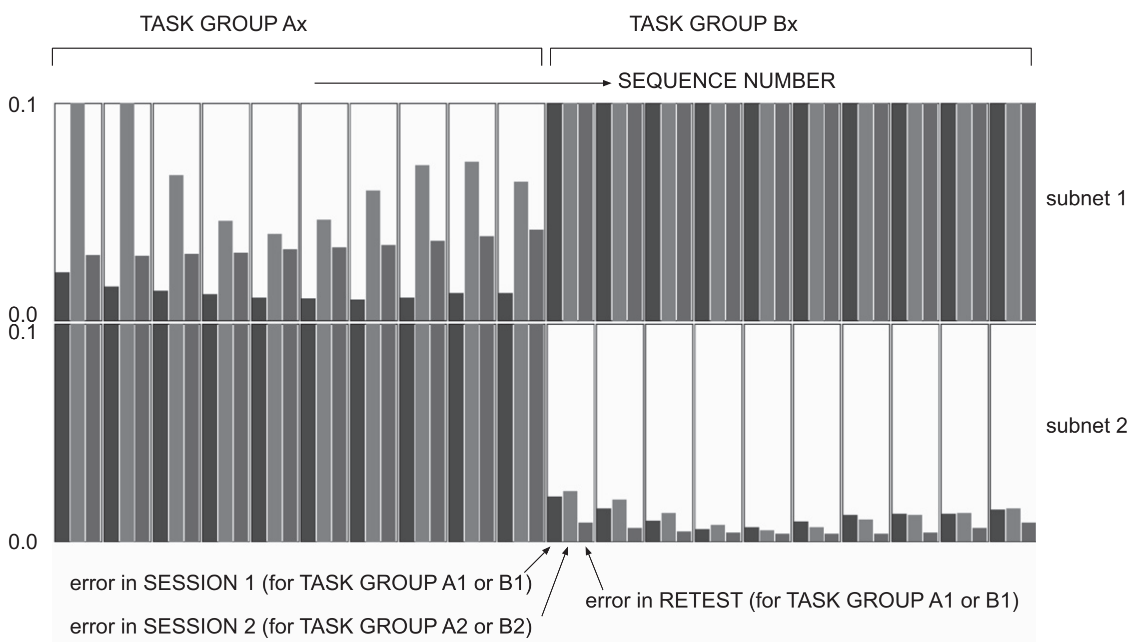


Fig. 9. Error comparison for three sessions.

in another case. Allowing the previously acquired subnet to be flexible in subsequent additional learning may provide a new perspective on the modularity of skill and knowledge. The acquired modular structure is the basis for processing ongoing information signals, but in some cases the structure itself is modified by the current task. The plasticity of a learner to a novel situation is a source of rich dynamics in the skill acquirement process.

References

[1] L. Berthouze, Y. Kuniyoshi, Emergence and categorization of coordinated visual behavior through embodied interaction, Mach. Learn. 31 (1998) 187–200.

[2] T. Brashers-Krug, R. Shadmehr, E. Bizzi, Consolidation in human motor memory, Nature 382 (1996) 252–255.

[3] Y. Dudai, The neurobiology of consolidations, or, how stable is the engram?, Ann. Rev. Psychol. 55 (2004) 51–86.

[4] Z. Ghahramani, D.M. Wolpert, Modular decomposition in visuomotor learning, Nature 386 (1997) 392–395.

[5] R. Jacobs, M. Jordan, S. Nowlan, G. Hinton, Adaptive mixtures of local experts, Neural Comput. 3 (1991) 79–87.

[6] M. Jordan, R. Jacobs, Hierarchical mixtures of experts and the EM algorithm, Neural Comput. 6 (1994) 181–214.

[7] F. Kaplan, P.-Y. Oudeyer, Maximizing learning progress: an internal reward system for development, in: F. Iida, R. Pfeifer, L. Steels, Y. Kuniyoshi (Eds.), Embodied Artificial Intelligence, Lecture Notes in Artificial Intelligence, vol. 3139, Springer, Berlin, 2004, pp. 259–270.

[8] Y. Kuniyoshi, L. Berthouze, Neural learning of embodied interaction dynamics, Neural Networks 11 (1998) 1259–1276.

[9] P.-Y. Oudeyer, F. Kaplan and V.V. Hafner, Intrinsic motivation systems for autonomous mental development, in: Intrinsic Motivation Systems for Autonomous Mental Development, IEEE Transactions on Evolutionary Computation, Special Issue on Autonomous Mental Development, to appear.

[10] D.E. Rummelhart, J.L. McClelland and PDP Research Group, Parallel Distributed Processing: Explorations in the Microstructures of Cognition, vols. 1, 2, MIT Press, Cambridge, MA, 1986.

[11] R. Shadmehr, H.H. Holcomb, Neural correlates of motor memory consolidation, Science 8 (1997) 821–825.

[12] L.B. Smith, E. Thelen, Development as a dynamic system, Trends Cognitive Sci. 7 (2003) 343–348.

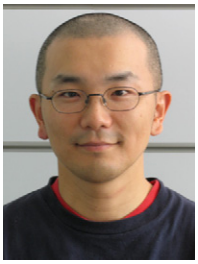
[13] Y. Sugita, J. Tani, Learning semantic combinatoriality from the interaction between linguistic and behavioral processes, Adaptive Behav. 13 (1) (2005) 33–52.

[14] J. Tani, M. Ito, Self-organization of behavioral primitives as multiple attractor dynamics: a robot experiment, IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans 33 (4) (2003) 481–488.

[15] J. Tani, S. Nolfi, Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems, Neural Networks 12 (1999) 1131–1141.

[16] E. Thelen, L.B. Smith, A Dynamic Systems Approach to the Development of Cognition and Action, MIT Press, Cambridge, MA, 1993.

[17] M.P. Walker, T. Brakefield, J.A. Hobson, R. Stickgold, Dissociable stages of human memory consolidation and reconsolidation, Nature 425 (2003) 616–620.
[18] D.M. Wolpert, M. Kawato, Multiple paired forward and inverse models for motor control, Neural Networks 11 (1998) 1317–1329.



Ichiro Igari is a researcher in the Brain Science Institute, RIKEN, Japan. He received doctorate from the University of Tokyo, Japan, with a thesis on "Numerical Studies of Language and Discourse Complexity: A Dynamical Systems Approach". His current research focuses on theories and numerical models for embodied cognition and memory dynamics.

Jun Tani was born in 1958 in Tokyo, Japan. He received the B.S. degree in mechanical engineering from Waseda University, M.S. degree in electrical engineering from Michigan University and Dr.Eng. from Sophia University. He is currently a team leader in Lab. For Behavior and Dynamic Cognition, Brain Science Institute, RIKEN in Tokyo, Japan. He is interested in embodied cognition, complex adaptive systems and brain science.