Learning of Situation Dependent Prediction toward Acquiring Physical Causality

Masaki Ogino* **Tetsuya Fujita**** **Sawa Fuke**** **Minoru Asada***,****

*JST ERATO Asada Synergistic Intelligence Project
Yamadaoka 2-1, Suita, Osaka 565-0871, Japan

**Osaka University
Graduate School of Engineering, Department of Adaptive Machine Systems,
Yamadaoka 2-1, Suita, Osaka 565-0871, Japan

Abstract

Physical causality is one of the most important knowledge that human babies learn first after birth through interaction with the surrounding environment. The properties of object movement changes depending on the situation, and so the agent should change its prediction. This paper proposes a learning model which predicts the movement of an attended object depending on the environment around the object. The predictor is formed by three main layered associative modules: (a) an *environment* module, which recognizes the attended object and its surrounding environment; (b) a *predictor module*, which anticipates the movement of the attended object depending on the surrounding environment; (c) an *attention module* which implements bottom-up and top-down attention processes. The proposed method is applied to the robot, and its prediction faculty and adaptability are examined in the simulation and actual environment.

1. Introduction

All infants are physicists. From the day of the birth, they begin to learn the fundamental properties of the world step by step through the interaction with the surrounding environments. The one of the important indices for their progress is "object permanence"; even when an attended object will be occluded from the view by an obstacle, they can understand the object will not be lost from the world and remain behind obstacles. Although Piaget firstly proposed that infants can acquire object permanence after 18 month old (Piaget, 1954), other researchers has shown that infants can pass object permanence task before one year old (Baillargeon et al., 1985) (Baillargeon and DeVos, 1991). In developmental cognitive robotics, some learning models are proposed to explain the results shown in these experiments with more restricted facilities (Schlesinger, 2003) (Lovett and Scasselatti, 2004). Although an actual mechanism that enables an infant to show these behaviors even in such a early stage is still unknown, how such higher concepts about the world as object continuity and impossibility can be learned autonomously is also an interesting problem in a robot area (Fitzpatrick et al., 2008). One of the fundamental faculties to realize the higher concepts about the world is to model the phenomena effectively for appropriate prediction. In this paper, we propose a learning model that enables a robot to learn the prediction of the object movement depending on the situation. For this purpose, multiple Restricted Boltzmann Machines (RBMs) (Hinton et al., 2006) are adopted, which can be used for both unsupervised and supervised learnings.

Moreover, with this learning model we treat the problem on the relationship between attention and learning. Attention is thought to consist of two processes; bottom-up and top-down attention (Knudsen, 2007). Whereas bottom-up attention is modeled well by the intrinsic features of the input image, top-down attention is affected by the experience. So, what to be attended is affected by learning, on the other hand what is learned is affected by attention. We set the attention level based on the prediction error and how the attention level affects to the learning.

2. Situation dependent predictor with Restricted Boltzmann Machine

2.1 Overview

In order to realize a situation dependent predictor, the prediction of the attended object should be well merged with recognition of the environment.

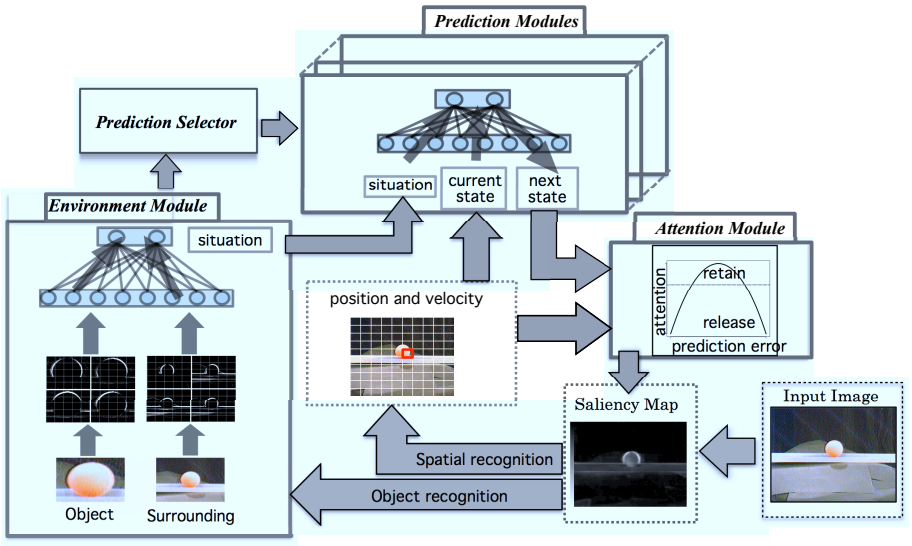


Figure 1: Overview of situation-dependent predictor

This is also an interesting problem as the model of integration of two visual pathways (where pathway and what pathway) in the brain, where appropriate self-organization for information compression and integration should be realized. For that purpose, we apply Restricted Boltzmann Machine (RBM) (Hinton et al., 2006) because this network model possesses good features as a building unit to make a larger system.

Fig. 1 shows a proposed system which consists of 4 modules; attention module, environment recognition module, predictor selector and motion predictor. The attention module determines the attention area in the environment. From the attended area, the geometrical information of an attended object and its surrounding images are extracted and self organized by RBM in the environment recognition module. The self organized information is associated with the information of the movement of the attended object in the prediction module. Based on the association memory feature of RBM, the prediction module can reconstruct the next position of the attended object based on the current position and the current environmental situation. In this section, first, the learning algorithm of the restricted Boltzmann machine is explained. Second, it is explained how RBMs are used in the environment recognition module and prediction module. Then, the attention module is explained

2.2 Restricted Boltzmann Machine

Restricted Boltzmann Machine (Hinton, 2007) (Hinton et al., 2006) is a neural network consisting of two layers, input (visible) layer and 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 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 biases for activation.

The learning of RBM is processed by the calculation process called *reconstruction*. First, the activation lavel of the hidden layer, h_j, are calculated by the forward calculation based on the input data v_i, the connection weights w_{ij} and biases β_{v_i} with eq. (1). Then the activation level of the visible layer, v_i, is calculated again with the activation level of the hidden layer, h_j with eq. (2). In the following, this re-calculated activation level of the visible layer is called reconstruction data. This calculation process can be proceeded repeatedly. The superscript of the unit v_i and h_j mentions the number of the repeated calculation between layers. When the probabilistic distribution of the input data and the reconstructed data after ∞ repeats of reconstruction are $\hat{p}(\mathbf{v})$ and $\hat{p}(\mathbf{v}|\mathbf{w})$, respectively, the purpose of the learning is to adjust the connection weights, w_{ij}, to minimize the difference of the distribution 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 the 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 probability of the realization of the state $(\mathbf{v}, \mathbf{h})$ is proportional to the total energy,

$$p(\mathbf{v}, \mathbf{h}|\mathbf{w}) \propto e^{-E(\mathbf{v},\mathbf{h}|\mathbf{w})}. \quad (6)$$

Thus, when the function of the right side of the equation is described as f like,

$$f(\mathbf{v}, \mathbf{h}|\mathbf{w}) = e^{-E(\mathbf{v},\mathbf{h}|\mathbf{w})} \quad (7)$$

$$f(\mathbf{v}|\mathbf{w}) = \sum_{\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h}|\mathbf{w})}, \quad (8)$$

then the probabilities of the realization of the state $(\mathbf{v}, \mathbf{h})$ and bfv given the weights bfw can be written with f as

$$p(\mathbf{v}, \mathbf{h}) = \frac{f(\mathbf{v}, \mathbf{h}|\mathbf{w})}{\sum_{\mathbf{v},\mathbf{h}} f(\mathbf{v}, \mathbf{h}|\mathbf{w})} \quad (9)$$

$$p(\mathbf{v}) = \frac{\sum_{\mathbf{h}} f(\mathbf{v}, \mathbf{h}|\mathbf{w})}{\sum_{v,h} f(\mathbf{v}, \mathbf{h}|\mathbf{w})} = \frac{f(\mathbf{v}|\mathbf{w})}{\sum_{\mathbf{v}} f(\mathbf{v}|\mathbf{w})}. \quad (10)$$

Applying the relation $\log p(\mathbf{v}|\mathbf{w}) = \log f(\mathbf{v}|\mathbf{w}) - \sum_v \log f(\mathbf{v}|\mathbf{w})$, the derivation of the cross entropy error, 4, can be transformed as follows,

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}} &= \langle \frac{\partial}{\partial \mathbf{w}} \log f(\mathbf{v}|\mathbf{w}) - \sum_{\mathbf{v}} \frac{f(\mathbf{v}|\mathbf{w})}{Z} \frac{\partial}{\partial \mathbf{w}} \log f(\mathbf{v}|\mathbf{w}) \rangle_{\hat{p}(\mathbf{x})} \\
&= \langle \frac{\partial}{\partial \mathbf{w}} \log f(\mathbf{v}|\mathbf{w}) - \sum_{\mathbf{v}} p(\mathbf{v}|\mathbf{w}) \frac{\partial}{\partial \mathbf{w}} \log f(\mathbf{v}|\mathbf{w}) \rangle_{\hat{p}(\mathbf{x})} \\
&= \langle \frac{\partial}{\partial \mathbf{w}} \log f(\mathbf{v}|\mathbf{w}) \rangle_{\hat{p}(\mathbf{x})} - \langle \frac{\partial}{\partial \mathbf{w}} \log f(\mathbf{v}|\mathbf{w}) \rangle_{p(\mathbf{v}|\mathbf{w})} \\
&= \langle \frac{\partial}{\partial \mathbf{w}} \log f(\mathbf{v}|\mathbf{w}) \rangle_{p_0} - \langle \frac{\partial}{\partial \mathbf{w}} \log f(\mathbf{v}|\mathbf{w}) \rangle_{p_\infty}
\end{aligned}$$

where p_0 is the input data (0−th reconstruction data) and p_∞ is the ∞-th reconstruction data. For actual calculation, instead of p_∞, $1 - st$ reconstruction data, p_1, is used for minimization. Then, the derivation can be simplified as

$$\begin{aligned}
&\langle \frac{\partial}{\partial w_{ij}} \log f(\mathbf{v}|w_{ij}) \rangle_{p_0} - \langle \frac{\partial}{\partial w_{ij}} \log f(\mathbf{v}|w_{ij}) \rangle_{p_1} \\
&= \langle \frac{\partial}{\partial w_{ij}} \sum_{i,j} v_i h_j w_{ij} \rangle_{p_0} - \langle \frac{\partial}{\partial w_{ij}} \sum_{i,j} v_i h_j w_{ij} \rangle_{p_1} \quad (11) \\
&= \langle v_i h_j \rangle_{p_0} - \langle v_i h_j \rangle_{p_1} \quad (12)
\end{aligned}$$

Thus, the update learning rule for minimizing the cross entropy error can be derived as

$$\Delta w_{ij} = \epsilon(v_i^0 \mathbf{P}(h_j^0 = 1) - \mathbf{P}(v_i^1 = 1)\mathbf{P}(h_j^1 = 1)). \quad (13)$$

In the same way, the learning rule for biases can be derived as

$$\Delta \beta_{h_j} = \epsilon(\mathbf{P}(h_j^0 = 1) - \mathbf{P}(h_j^1 = 1)) \quad (14)$$

$$\Delta \beta_{v_i} = \epsilon(\mathbf{P}(v_i^0 = 1) - \mathbf{P}(v_i^1 = 1)) \quad (15)$$

In the actual learning, the input data are divided into several groups and the parameters are updated group by group to avoid the over learning. Moreover, we added the additional of learning rule to limit the activation rate of each unit. This sparseness constraint seems to be important to describe the input data with more compact patterns of activations in hidden layers (Lee et al., 2008). The convergence of the learning is evaluated by the total error between input data and the reconstruction data,

$$err = v_i^0 - \mathbf{P}(v_i^1 = 1). \quad (16)$$

After learning, the reconstruction process can be used for reconstructing complete data set from the incomplete data. This feature can be used for association of the given multiple data sets. Moreover, when the number of the units in the hidden layer is less than that in the visible layer, the extraction of the important features of the input data can be expected. Hinton stresses that this characteristic of RBM favorable for avoiding local minima in learning of deep layered network. Thus, RBM has both features of supervised and unsupervised self-organization learning properties.

2.3 *Environment Module*

An object will change the movement depending on the environment where the object is put. The properties of the movement will be affected by the shape of the object. For example, we expect a ball shape will be expected to move easily but not for a square object. And the same object will change its movement depending on the pathway the object is put on. The environment module categories visual information of an attended object and its surroundings.

To extract the geometrical information from the images of an attended object and its surroundings, the results of the gabor filters of these images are input to RBM. The result images of the gabor filters of $\psi = [0°, 45°, 90°, 135°]$ are segmented into 5×5 units. In each unit, the pixel values are summed and normalized to the attended area. The activation level of the j-th unit, I_j, is determined by the normalized summed value a_j and some threshold,

$$I_j = \begin{cases} 1 & (a_j > th) \\ 0 & (a_j \leq th) \end{cases}. \quad (17)$$

The input to the environment module RBM, $\mathbf{v}^{<env>}$, is the combination of the vectors of the gabor filter

results for the object image, $\mathbf{I}^{<obj>}$, and the vectors of the gabor filter results for the surrounding image, $\mathbf{I}^{<around>}$,

$$\mathbf{v}^{<env>} = (\mathbf{I}^{<obj>}, \mathbf{I}^{<around>}). \quad (18)$$

Fig. 2 shows the flow chart of the processing.

After learning, the activation pattern in the hidden layer, $\mathbf{h}^{<\mathbf{env}>}$, is expected to describe self organized information of the input images. Thus, these information is used in the prediction module for prediction of the movement of the attended object.

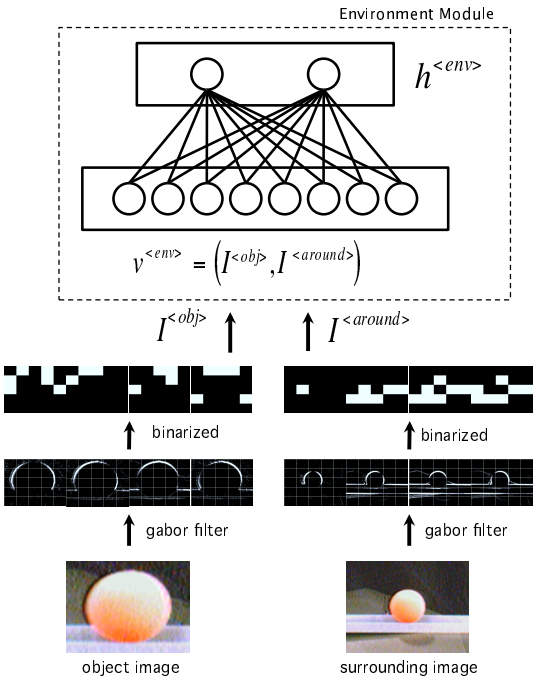


Figure 2: The environment module

2.4 Prediction Module

Fig. 3 shows the schema of the prediction module. The RBM in prediction module associates the current movement information $S(t)$, the previous movement information $S(t-1)$, and the situation information $\mathbf{h}^{<env>}$. The position information consists of the position and the velocity of an attended object,

$$\mathbf{S}(t) = (x_0, x_1, ..., x_{n-1}, y_0, y_1, ..., y_{m-1}, dx_0, dx_1, ..., dx_{2n-1}, dy_0.dy_1, ..., dy_{2m-1}).$$

When the image size is $W \times H$ and it is divided into $n \times m$, and the coordinates of the attended object are (x, y), then the position nodes are determined by the following equations,

$$x_i = \begin{cases} 1 & (\frac{x}{W/n} \leq i < \frac{x}{W/n} + 1) \\ 0 & else \end{cases} \quad (19)$$

and

$$y_j = \begin{cases} 1 & (\frac{y}{H/m} \leq j < \frac{y}{H/m} + 1) \\ 0 & else \end{cases}. \quad (20)$$

When the shift of the attended object between observed steps is (dx, dy), the velocity nodes are determined by

$$dx_i = \begin{cases} 1 & (\frac{dx}{W/n} + n - 1 \leq i < \frac{dx}{W/n} + n) \\ 0 & else \end{cases} \quad (21)$$

and

$$dy_j = \begin{cases} 1 & (\frac{dy}{H/m} + m - 1 \leq j < \frac{dy}{H/m} + m) \\ 0 & else \end{cases}. \quad (22)$$

In order to realize the prediction in the various time scales and spatial frames, several RBM with various kinds of time scale and spatial segmentation sizes are prepared. Among them, the appropriate predictor is selected based on the reliability of the predictors. The reliability of i-th predictor, c_i, is calculated based on the hidden layer of the environment recognition module, $\mathbf{h}^{<env>}$, as

$$c_i = \sum_j w_{ij}^s \times h_j^{<env>}. \quad (23)$$

The connection weights, w_{ij}^s, is learned based on the following Hebbian learning,

$$\Delta w_{ij}^s = \epsilon(e^{-\Delta r_i} \times h_j^{<env>}) \quad (24)$$

where Δr_i is the prediction error of the movement in i-th prediction module, ϵ is the learning rate. The activation level of each RBM, $a_i^{<RBM>}$, is calculated based on the reliability c_i,

$$a_i^{<RBM>} = \frac{1}{1 + exp(-\sum_i c_i)} \quad (25)$$

and the RBM that has the maximum value is selected as the predictor under the current situation.

2.5 Attention Module

We hypothesized that attention consists of three processes; catch, retain and release. First, in the catch process, the attended point is selected based on the saliency (Itti et al., 2003). For that purpose, the saliency map is calculated with regard to various image features such as intensity, color, motion, etc. Once the attended point is decided, the attended object area is evaluated as the set of pixels that have the same color of attended point. Then, the attended object area is segmented and used for the template for pattern matching. The object are is normalized and binarized as the input for the attention module, $I^{<obj>}$ (Fig. 2). The surrounding image of the attended object whose size is the half of the camera image is normalized and binarized as the input for the

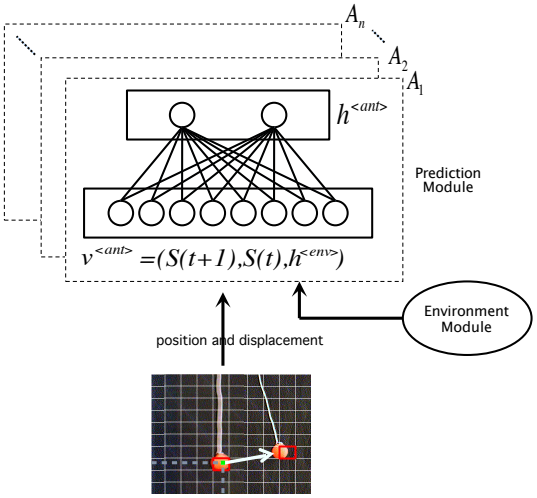


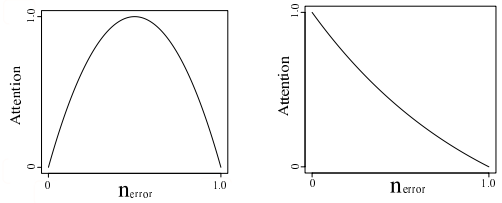
Figure 3: The prediction module

attention module, $I^{<around>}$. The attention point is retained for the learning until the trigger for releasing attended point is given. For effective learning, it is supposed that the attended points whose movement can be predicted completely should be released. The points whose movement are random should also be released earlier because such points may well be noise. On the other hand, the points whose movement can be partly predicted should be retained long for learning more. For that purpose, we compared two kinds of functions that decide the probabilities to release the attention points.

$$attention1 = \frac{-e^{n_{error}} - e^{1-n_{error}} + e + 1}{-2e^{0.5} + e + 1} \quad (26)$$

$$attention2 = \frac{e^{1-n_{error}} - 1}{e - 1} \quad (27)$$

where n_{error} is the rate of the number of the prediction modules that fails to predict. The graphs are shown in Figs. 4. The attention is released when the above attention level becomes less than some threshold.



(a) Attention function1 (b) Attention function2

Figure 4: Attention function

3. Experiments

3.1 Learning Prediction without attention

To validate the prediction faculty, the proposed system is applied to the real robot. Fig. 5 shows the robot *FK* used in the experiment. Although this robot has two IEEE 1394 cameras and 2 degrees of freedom (pan and tilt) to move the camera, only the right camera is used with eye position fixed. The camera image is captured with 33 [frames/sec].

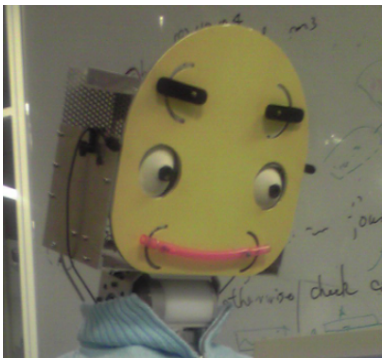
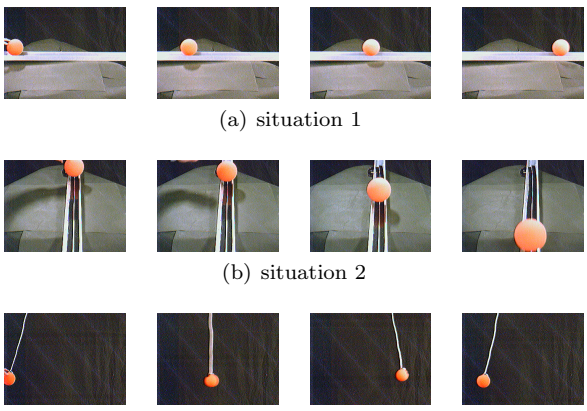


Figure 5: robot *FK*

To validate the prediction faculty in various situations, three kinds of situations are prepared; a ball on the holizontal rail, a ball on the vertical rail and a ball in the pendulum 6. In each situation, 4 tri-



(a) situation 1

(b) situation 2

(c) situation 3

Figure 6: Situations of experiments

als are recorded, each of which has about 90 steps. For the prediction module, 6 RBMs are prepared (2 kinds of segmentations ($40 \times 40, 10 \times 10$) and 3 kinds of time steps ($5, 10, 20$ steps)). The numbers of the visible and hidden units of the environment module are 200 and 50, respectively. The numbers of the visible and hidden units of the prediction modules are 368 and 92 for the segment size 40×40, 128 and 32 for the segment size 10×10. The attended area to be

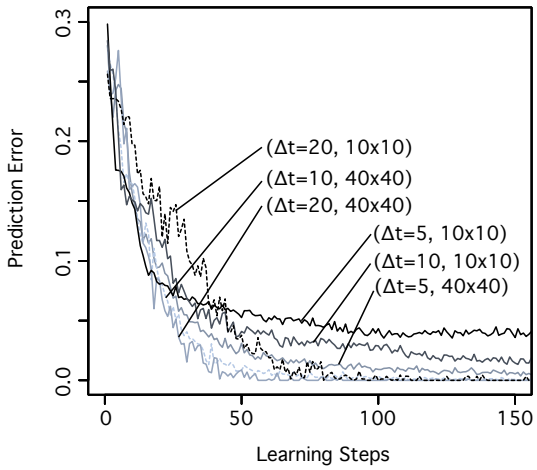
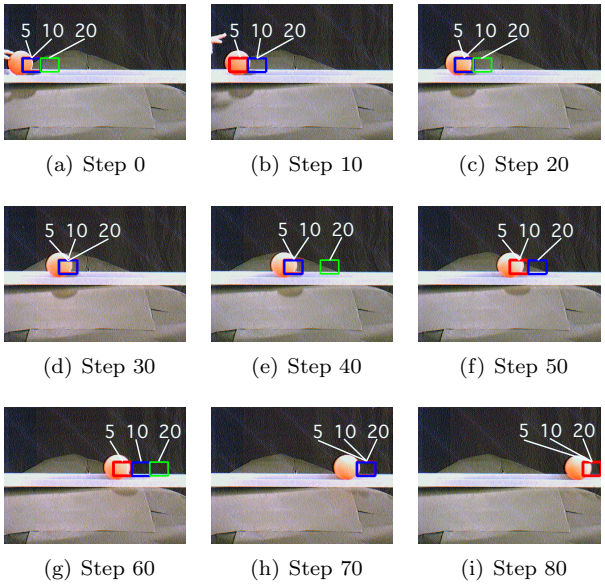


Figure 7: Prediction Error without attention

attended is given as the image template (orage ball) by the designer in this experiment.

Fig. 7 shows the learning error of all RBMs. The learning of each RBM converges within 100 learning steps. The examples of the prediction after learning is mentioned in Fig. 8. These are the predictions of RBMs that have 10×10 segments and 5, 10, 20 prediction time steps (20 step prediction is shown only in every 20 step).



(a) Step 0 (b) Step 10 (c) Step 20

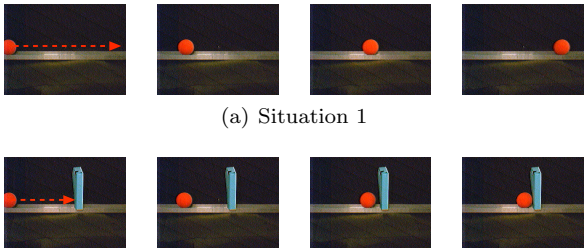
(d) Step 30 (e) Step 40 (f) Step 50

(g) Step 60 (h) Step 70 (i) Step 80

Figure 8: Examples of prediction of the movement after learning

3.1.1 Supplemental Learning

To validate the faculty of the predictor in additional learning, after learning in one situation (Fig. 9(a)), additional data in another situation (Fig. 9(b)) is



(a) Situation 1

(b) Situation 2

Figure 9: The training data for supplemental learning

given to the network. For each situation, 3 trials (each trial consists of 68 steps) are recorded for training data. The other conditions are the same as the previous subsection. The data of second situation is added to the training data of the situation network after the 250 steps of learning in the first situation. Fig. 10 shows the time courses of the averaged error rate per one node through the learning process (only 2 of 6 predictors are shown). Around the 250-th learning steps, the error rate rises when new data is added to the training data. However, in the following 100 steps, the error rate decreases to around 0.3 indicating that the network successfully represent both the new and old situation. Fig.

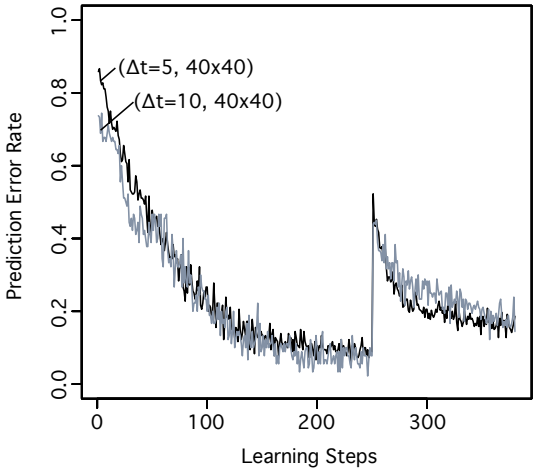
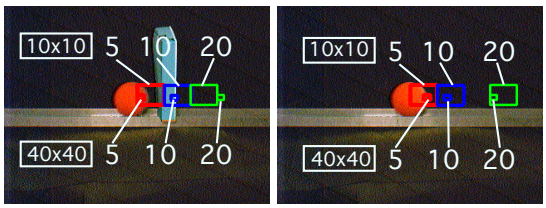


Figure 10: The error rate per one node in the supplemental learning

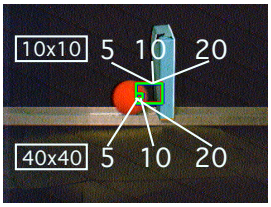
11 shows the predicted position of the attended object before and after the supplemental learning. (a) Before the supplemental learning, the robot predicts the attended ball will go through the wall because he does not experienced such kind of situation (squares are the predicted positions in the next 5, 10 and 20 steps. The big and small squares are the prediction in 10×10 and 40×40 segments.). (b) After the supplemental learning, the robot can make appropriate predictions depending on the situations. Before the supplemental learning, the robot predicts that

the ball will move through the wall in right direction as before (b). After the supplemental learning, the robot can predict that the ball will stop at the wall (c).



(a) Prediction before learning

(b) Prediction after learning (situaton 1)



(c) Prediction after learning (situation 2)

Figure 11: Prediction of the ball position before and after the supplemental learning

3.2 Learning Prediction with Attention

In the experiments of previous subsection, the object to be attended is given by the designer in advance. In order for a robot to learn the physical causality autonomously, it is important to implement an attention control system appropriately. For this purpose, we applied the attention module to the same situations as the experiments explained in the previous subsection. Each situation consists of 3 trials that have 90 steps. For the prediction module, 6 RBMs are prepared (2 kinds of segmentations $(40 \times 40, 10 \times 10)$ and 3 kinds of time steps $(5, 10, 20$ steps)) for the prediction modules.

Figs. 12 shows the time course of the error rate in the learning procedures with the attention function 1 (Fig. 12 (above)) and the attention function 2 (Fig. 12 (below)). Whereas the learning is not stable with the attention function 1, the learning with the attention function 2 converges to some stable state. This is because with the attention function 1 the robot easily change its attention to another point (often shiny noise point in the environment other than the object) when the first part of the movement can be learned. Figs. 13 show the timing when the robot changes its attention in the middle of the learning procedure for situation 3 with the function 1 (above) and with the function 2 (below). In these graphs, the gray line indicates the rate of the prediction failure modules, the black line indicates the attention level (calculated by eq. (26) and eq. (27)), and the dashed line indicates the threshold that the robot changes its attention (the arrows indicate the timing when the robot changes its attention). With the function 1, the robot predicts the first part of the movement successfully and loses its attention easily because the prediction is successfully done. This makes the learning unstable. On the other hand, with the function 2, the robot can keep its attention once the appropriate attention point (the object) is found. And the stable learning data can be obtained.

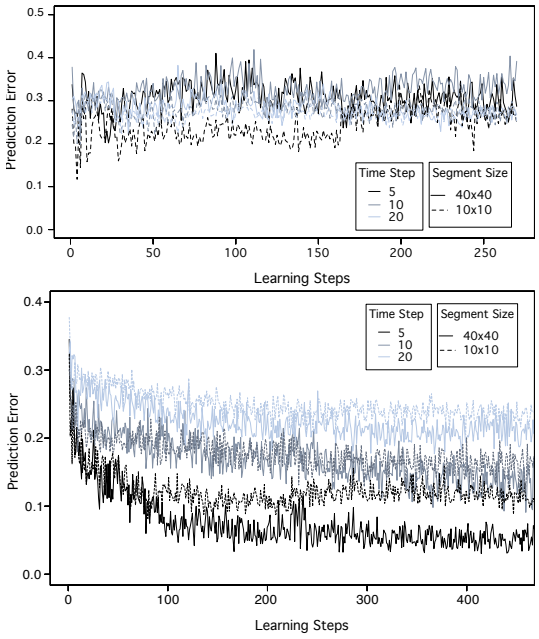


Figure 12: Prediction error based on the attention module with the function 1 (above) and the function 2 (below)

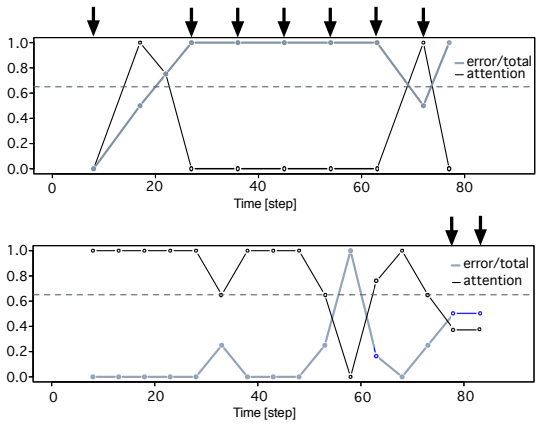


Figure 13: Attention level and attention changes based on the attention module with the function 1 (above) and the function 2 (below)

4. Discussion

In this paper, we proposed a layered associative network that can predict the movements of the observed object depending on the surrounding situation. The higher abstract concept such as "object permanence" can be acquired through the learning of many concrete phenomena in the real world. The proposed network could be extended to more higher representation of the world. Fig. 14 shows the result of the principle component analysis of the activation patterns in the hidden layer of the prediction module (the image segments are 10×10 and the prediction time step is 5). This graph shows the activation patterns can be self-organized depending on the situation. So, the information of the activation patterns can be used to discern the states such as "The ball is goes to left on the horizontal line.". This implies the possibility to construct higher abstract concept based on the self-organization of lower data through the bottom-up approach.

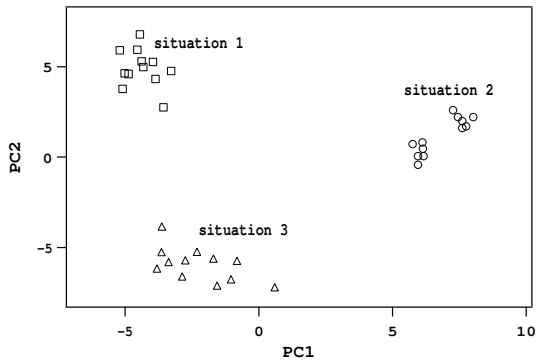


Figure 14: Principal component analysis of the activation patterns of hidden layer in prediction module

"Object permanence" is thought to be closely related to memory. To realize an object does not disappear behind an obstacle and will appear again, an agent should recognize that the reappeared object is the same one as the previous one. In the proposed network, if the attended object disappears behind some obstacle, the robot could not retain its attention because the robot will release its attention based on the attention level function 2. However, if the prediction module that enables long term prediction is available, the robot can retain its attention and relate the object behavior during disappearing and reappearing. The key faculty for this learning is how long working memory can record the series of events and how the prediction module will learn from the events in the working memory. In fact, it is reported that the working memory ability of infants is enhanced from 7.5 months (2 secs) to 12 months (12 sec) (Schwartz and Reznick, 1999) (Reznick et al., 2004). We are now conducting the experiments to relate the prediction ability of a disappeared object and the time length of memory.

References

Baillargeon, R. and DeVos, J. (1991). Object permanence in young infants: further evidence. *Child development*, 62:1227–1246.

Baillargeon, R., Spelke, E. S., and Wasserman, S. (1985). Object permanence in five-month-old infants. *Cognition*, 20:191–208.

Fitzpatrick, P., Needham, A., Natale, L., and Metta, G. (2008). Shared challenges in object perception for robots and infants. *Journal of Infant and Child Development*, 17(1):7–24.

Hinton, G. E. (2007). Learning multiple layers of representation. *TRENDS in Cognitive Sciences*, 11(10):428–434.

Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006). A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554.

Itti, L., Dhavale, N., and Pighin, F. (2003). Realistic avatar eye end head animation using a neurobiological model of visual attention. *Proceedings of SPIE*.

Knudsen, E. I. (2007). Fundamental components of attention. *Annual Review of Neuroscience*, 30:57–78.

Lee, H., Ekanadham, C., and Ng, A. Y. (2008). Sparse deep belief net model for visual area v2. In *Proceedings of the Neural Information Processing Systems (NIPS) 20*.

Lovett, A. and Scasselatti, B. (2004). Using a robot to reexamine looking time experiments. In *Proceedings of the 4th International Conference on Development and Learning (ICDL)*.

Piaget, J. (1954). *The construction of reality in the child*. basic books.

Reznick, J. S., Morrow, J. D., Goldman, B. D., and Snyder, J. (2004). The onset of working memory in infants. *Infancy*, 6(1).

Schlesinger, M. (2003). A lesson from robotics: Modeling infants as autonomous agents. *Adaptive Behavior*, 11(2).

Schwartz, B. B. and Reznick, J. S. (1999). Measuring infant spatial working memory using a modified delayed-response procedure. *Memory*, 7:1–17.