REM-ART: Reward-based Electromagnetic Adaptive Resonance Theory

Gyeong-Moon Park, Yong-Ho Yoo, and Jong-Hwan Kim
School of Electrical Engineering, KAIST, Daejeon, Republic of Korea
Email: {gmpark, yhyoo, johkim} @rit.kaist.ac.kr

Abstract— *This paper proposes a Reward-based Electromagnetic Adaptive Resonance Theory (REM-ART) mainly for reducing the retrieval error, which can store and retrieve episodes consisting of a temporal sequence of events. With the existing EM-ART it is difficult to predict a correct episode when there are noisy inputs that are sparse or distorted. To overcome this problem, the proposed REM-ART has a reward channel along with an event channel, lending it robustness to noisy inputs. The reward channel generates a reward signal to reinforce the activation of a proper episode node. The proposed REM-ART is applied to predict and retrieve a correct episode from noisy inputs, out of episodes that are stored through the visual demonstration using an RGB-D camera.*

Keywords: EM-ART, episode retrieval, episodic memory, fusion ART, REM-ART, reward channel.

1. Introduction

Semantic memory and episodic memory are the main categories of the declarative memory in the human brain. These concepts have been illustrated in detail by Tulving [1], [2]. Semantic memory represents generalized knowledge and meaning, and it is not related with individual experiences. Semantic memory is objective information such that anyone can share the meaning, akin to a pen. Episodic memory, on the other hand, is the subjective memory of an individual, which is composed of his or her own experiences, like the first day of attending school. This memory contains contextual connections of several events. For this reason, episodic memory can be implemented for multiple cognitive capabilities, such as sensing, reasoning, and learning [3], [4]. It is helpful for the human to retrieve whole sequential events, i.e., an episode at a specific time from only partial events observed [5].

Recently, many researchers have introduced computational models of episodic memory. Ho *et al.* [6] proposed an autonomous agent having a sequential character of episodic memory. Mueller and Shiffrin presented a model representing the relation between semantic knowledge and episodic memory [7]. The neural network models focusing on the retrieval capability of episodic memory have been designed [8]–[10]. These models represent relations between events, but they have difficulty in describing complicated orders of time sequential events.

EM-ART is a recent neural network model for representing intricate sequences of events [11]. It is based on fusion ART, which has multi-channel inputs in fuzzy ART, to encode each event from input data [12]. The activation values of recognized events are decayed and buffered in the event field by following the time sequence. This is the main difference between this model and the other episodic memory models. The model encodes an episode using a set of temporal events, and thus it can easily retrieve all sequential events by a readout process.

EM-ART can retrieve a correct episode from sparse events or some noisy inputs. In particular, the closer input data are to whole sequential events, the more robust this model is to noisy inputs such as unrelated events. However, if input data are too sparse to predict a correct episode, the model will be affected by the unbalance caused by the number of all committed events constituting an episode. If the model encodes two episodes with different numbers of sequential events, the activation value of an event that belongs to the episode with a smaller number of elements (events) is higher than that from an event that belongs to the other episode with a larger number of elements. This causes a retrieval error when EM-ART receives sequential inputs in real time.

To make the ART model with a more precise retrieval capability from partial input data, we propose Reward-based EM-ART (REM-ART). REM-ART has a multi-channel architecture not only in the first layer but also in the second layer, unlike other ART models. The proposed model has a reward channel in the second layer, which generates a reward signal to compensate recognition errors. The magnitude of the reward signal is proportional to the number of input events under the corresponding episode. REM-ART is learned using both unsupervised learning for encoding events and episodes and reinforcement learning for the reward signal. The sequential inputs to demonstrate the effectiveness of the proposed REM-ART are obtained through a RGB-D camera. For the demonstration, four scenarios are provided and the sequences of events are learned using REM-ART. The experimental results show stable and robust retrieval performance to noisy inputs and incomplete cues.

This paper is organized as follows. Section 2 introduces conventional fusion ART and EM-ART models. Section 3 proposes REM-ART with a detailed description. Section 4 presents experiment results along with a discussion. Finally, concluding remarks follow in Section 5.

2. Episodic Memory Learning

When a human performs a specific task, his or her episodic memory plays an important role in carrying out the task, referring to the memory that stores the sequential events for the task [3], [14]. A robot can also learn and memorize a series of events efficiently using an ART neural network model, which helps the robot recall time sequential events. In this section, fusion ART and EM-ART models are briefly reviewed, and how the robot can learn episodes by these models is described.

2.1 Event Learning

Fusion ART is an extended model of fuzzy ART. This model can receive several types of inputs at a time, since it has a multi-channel in the first layer. The architecture is useful to classify an event, since the event normally consists of several inputs, such as objects and related actions. Therefore, fusion ART is used for the basic structure to learn needed events. Fig. 1 shows the fusion ART architecture.

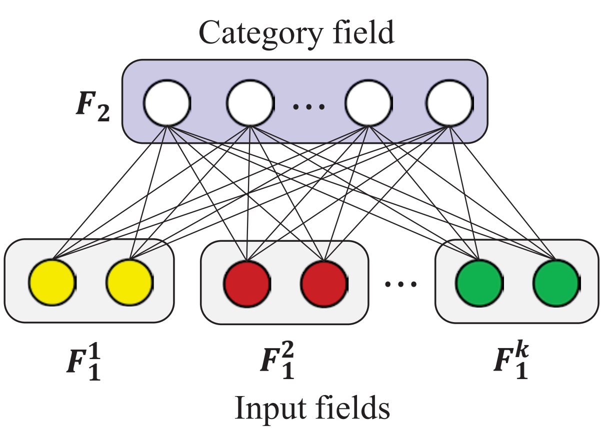


Fig. 1: Fusion ART architecture.

A summary of the fusion ART procedure is given below.

1) Complement coding: Each input field F_1^k receives an input vector $\mathbf{I}^k = (I_1^k, I_2^k, \dots, I_n^k)$, where $I_i^k \in [0,1]$ indicates the input i to channel k, for $k = 1, \dots, n$. The input vector $\mathbf{I}^k$ is converted to the activity vector $\mathbf{x}^k$ by a concatenation of the input vector $\mathbf{I}^k$ and its complements $\bar{\mathbf{I}}^k = 1 - \mathbf{I}^k$.

2) Code activation: The F_2 has one channel, which is represented as $\mathbf{y} = [y_1, y_2, \cdots, y_m]$, where m is the number of nodes. The jth output node in the category field is activated by the choice function:

$$T_j = \sum_{k=1}^{n} \gamma^k \frac{\left|\mathbf{x}^k \wedge \mathbf{w}_j^k\right|}{\alpha^k + \left|\mathbf{w}_j^k\right|} \tag{1}$$

where γ^k is a contribution parameter, $\mathbf{w}_j^k$ is a weight vector associated with the category j, α^k is a choice parameter, the fuzzy AND operator $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv min(p_i, q_i)$ and the norm $|.|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$ for vectors $\mathbf{p}$ and $\mathbf{q}$.

3) Code competition: A code competition is a selection process to distinguish the largest activation value in F_2 layer. The largest activation value is indexed at J as follows:

$$T_J = max\left\{T_j : \text{for all } F_2 \text{ node } j\right\}. \tag{2}$$

Once the maximum activation value is found among all F_2 nodes, this output value is set to one, and all the other values are set to zero. This is the winner-take-all strategy.

4) Template matching: For each channel k, the template matching process checks the resonance between each channel and all F_2 nodes. The resonance is defined as the similarity between the activity vector $\mathbf{x}^k$ and the weight vector $\mathbf{w}_J^k$ associated with the selected execution node, which is calculated using the following function:

$$m_J^k = \frac{\left|\mathbf{x}^k \wedge \mathbf{w}_J^k\right|}{\left|\mathbf{x}^k\right|} \geq \rho^k \tag{3}$$

where ρ^k is a vigilance parameter that is a constraint for the network resonance. If there are no matched nodes in F_2, an uncommitted node is added in F_2 as a new category node.

5) Template learning: Once the resonance occurs, the weight vector w_J^k for each channel k is updated by the following learning rule:

$$\mathbf{w}_J^{k(new)} = (1-\beta^k)\mathbf{w}_J^{k(old)} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(old)}) \tag{4}$$

where β^k is the learning rate. If the weight vector is initialized as $J_{1,2n}$, which is the one row of all-ones matrix, it can be arranged by the following procedure:

$$\begin{aligned}
\mathbf{w}_J^{k(1)} &= (1-\beta^k)\mathbf{w}_J^{k(0)} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(0)}) \\
&= (1-\beta^k)J_{1,2n} + \beta^k\mathbf{x}^k \\
\mathbf{w}_J^{k(2)} &= (1-\beta^k)\mathbf{w}_J^{k(1)} + \beta^k(\mathbf{x}^k \wedge \mathbf{w}_J^{k(1)}) \\
&= (1-\beta^k)^2 J_{1,2n} + (1-\beta^k)\beta^k\mathbf{x}^k + \beta^k\mathbf{x}^k \\
&= (1-\beta^k)^2 J_{1,2n} + (1-(1-\beta^k)^2)\mathbf{x}^k \\
&\vdots \\
\mathbf{w}_J^{k(n+1)} &= (1-\beta^k)^n J_{1,2n} + (1-(1-\beta^k)^n)\mathbf{x}^k
\end{aligned} \tag{5}$$

$$\lim_{n\to\infty} \mathbf{w}_J^{k(n)} = \mathbf{x}^k \tag{6}$$

Eq. (6) means the learned weight vector $\mathbf{w}_J^k$ converges to $\mathbf{x}^k$. It is applied to readout inputs from the category field such that $\mathbf{x}^{k(new)} = \mathbf{w}_J^k$.

2.2 Episode Learning

To recognize and learn episodes from continuous events, the feature of the time sequential representation of events is crucial. EM-ART is the neural network model for qualifying this character to fusion ART [11]. Fig. 2 shows a generalized EM-ART model for episodic memory learning.

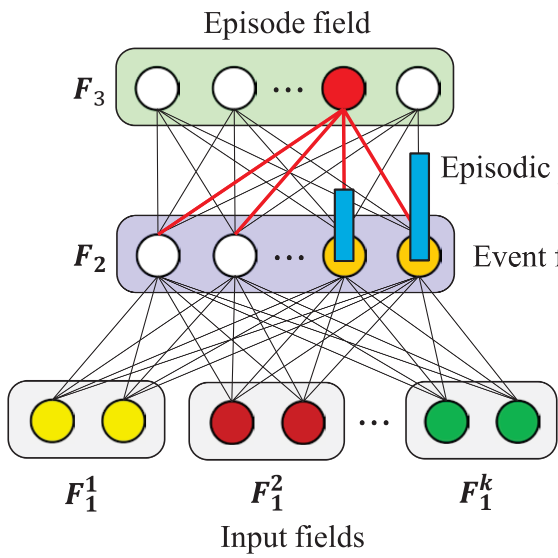


Fig. 2: EM-ART architecture.

As shown in Fig. 2, an episodic pattern is introduced to represent a temporal sequence of events. The encoding process of an episode by using a sequence of events as an input is the same as the encoding process of an event from an input vector, except that episodic encoding uses a time decaying factor to make a sequential input vector. In event field F_2, the activation values of recognized events are successively multiplied by the time decaying factor to keep track of the sequential order. If a set $\mathbf{y}$ represents the output vector of the event field, the first activation value of $\mathbf{y}$ is set to one. The index of the activated node is memorized until the next activation occurs in the event field. When a new activation occurs, y_j is decayed by multiplying the decaying factor using the following equation:

$$y_j^{new} = y_j^{old}(1-\tau) \tag{7}$$

where τ is the decaying factor. It makes an event encoded earlier than the other events have a smaller activation value to represent time decay. After finishing the matching process of event field nodes, non-activated nodes are set to zero. The output vector $\mathbf{y}$ containing temporal information is then the new input for the next additional layer.

2.3 Episode Retrieval

Once EM-ART is learned, it can retrieve episodes and time sequential events of each episode from time sequential inputs. Complete or incomplete cues can be the input vector for EM-ART. When input fields get cues composed of partial events, the template matching process checks the resonance in F_2. After recognizing events, the template matching between F_2 and F_3 is processed in the same manner to retrieve the matched episode. Every time sequential inputs are entered into input fields, EM-ART predicts an episode matched with inputs. This allows the model to retrieve a target episode from complete or incomplete cues in real time.

After recognizing the episode, EM-ART should be able to readout all sequential events matched with the episode. First, the activated node in F_3 readouts the input vector in F_2 using complement coding and learned weight vector. In the retrieved $\mathbf{y}$ vector, the highest value is selected and this value readouts the input vector in F_1. Successively, the value is set to zero and the next highest value is selected as the next input vector. After this process, EM-ART can readout all time sequential events of the selected episode.

2.4 The Limitation of Episode Retrieval

The episode is made of time sequential events, and each episode can have a different number of events for its elements. This implies that the contribution of each event belonging to the different episodes for the activation is different. In particular, the activation value from an event belonging to the episode with a smaller number of events is larger than that from an event belonging to the episode with a larger number of events. It can be shown from the choice function in F_2:

$$T_j = \frac{|\mathbf{y} \wedge \mathbf{w}_j|}{|\mathbf{w}_j|} \tag{8}$$

where k is omitted because F_2 is a single channel. In this equation, the choice parameter α is set to zero and the contribution parameter γ is set to one. As the learning iteration is increased, the learned weight vector $\mathbf{w}_j$ is almost the same as the input vector $\mathbf{y}_j$. Therefore, the summation of all elements in the weight vector $\mathbf{w}_j$ is computed as follows:

$$\begin{aligned} |\mathbf{w}_j| &\simeq |\mathbf{y}_j| \\ &= \sum_{n=0}^{h_j-1}(1-\tau)^n + \sum_{n=0}^{h_j-1}(1-(1-\tau)^n) + (N-h_j) \\ &= \sum_{n=0}^{h_j-1}(1) + (N-h_j) \\ &= N \end{aligned} \tag{9}$$

where h_j is the number of events corresponding to the jth episode, and N is the number of all learned events. This equation shows $|\mathbf{w}_j|$ is always constant at N. To compare the activation values calculated from an event, consider that the event field gets only one event corresponding with the jth episode for the input. In this case, the choice functions of the ith and jth nodes are respectively as follows:

$$
\begin{aligned}
T_i &= \frac{|\mathbf{y} \wedge \mathbf{y}_i|}{N} \\
&= (\sum_{n=0}^{h_i-1}(1-(1-\tau)^n) + (N-h_i-1))/N \\
T_j &= \frac{|\mathbf{y} \wedge \mathbf{y}_j|}{N} \\
&= (\sum_{n=0}^{h_j-1}(1-(1-\tau)^n) + 2(1-\tau)^p + (N-h_j-1))/N
\end{aligned} \tag{10}
$$

where $p \in [0, h_j - 1]$ denotes the order of a selected event, and $h_j > h_i$. Therefore, the difference between them is as follows:

$$
\begin{aligned}
T_j - T_i &= \frac{|\mathbf{y} \wedge \mathbf{y}_j|}{N} - \frac{|\mathbf{y} \wedge \mathbf{y}_i|}{N} \\
&= (2(1-\tau)^p - \sum_{n=h_i}^{h_j-1}(1-\tau)^n)/N.
\end{aligned} \tag{11}
$$

The difference can be a negative value under the condition $h_j > h_i$. This means the result of episode retrieval may be the ith episode, although the input event belongs to the jth episode as well. This demonstrates that the number of events in each episode affects the activation value. Therefore, the noisy input vector having mixed events may lead to retrieval of a wrong episode, which is critical in reproducing a complete pattern of sequential events.

3. Proposed REM-ART

EM-ART performs well when the input vector in F_1 is exactly matched with time sequential events, or for little noisy inputs. However, this model does not easily preserve robustness against noisy inputs. Moreover, if the number of time sequential events in each episode is different, EM-ART may not retrieve a target episode from the noisy inputs. To overcome this problem, we propose Reward-based EM-ART (REM-ART) consisting of input, category, episode fields. The category field is composed of event and reward channels. The role of the reward channel is to generate a reward that makes the network more robust to noisy inputs.

3.1 Architecture of REM-ART

REM-ART network has multi-channels in F_2 as well as in F_1. Fig. 3 shows the architecture of the model. Reward channel F_2^r in Fig. 3 receives the information about recognized events from event field F_2^e to generate the reward signal. This architecture is similar to the model having two connected fusion ART in the first and the second layers. However, the REM-ART structure in F_2 is different from the conventional fusion ART in that each channel is connected unidirectionally from one to the other channel.

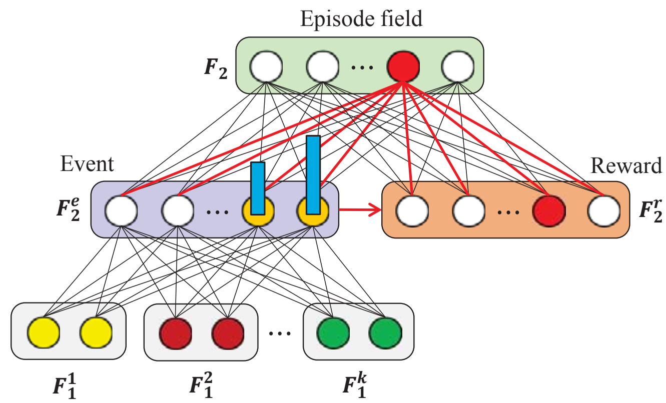


Fig. 3: REM-ART architecture.

3.2 Episode Learning

To encode events, REM-ART uses the basic fusion ART between input fields F_1 and event channel F_2^e. This is the same as the encoding process of EM-ART. After perceiving events, the event channel makes time sequential inputs for episode field, and simultaneously the episode field informs the reward channel of the number of sequential events of the encoded episode. The reward channel then makes the following reward values:

$$
\begin{aligned}
\mathbf{y}^r &= \{y_1^r, y_2^r, \cdots, y_f^r, \cdots, y_F^r\} \\
&= \left\{\frac{n_1}{h}, \frac{n_2}{h}, \cdots, \frac{n_f}{h}, \cdots, \frac{n_F}{h}\right\}
\end{aligned} \tag{12}
$$

where $\mathbf{y}^r$ is a reward vector, F is the number of learned episodes, n_f is the number of events of the fth episode among all input events, and $h = \sum_{f=1}^{F} n_f$. In the network learning process, the reward signal is always one because $n_f = h$. The other reward signals are zero because all $n_i = 0$, when $i \neq f$. The reward vector $\mathbf{y}^r$ and the time sequential input vector $\mathbf{y}^e$ constitute multi-channel inputs for encoding each episode. After the learning process, the learned reward weight vector asymptotically converges to

$$
\mathbf{w}^{r(n+1)} = (1-\beta^r)^n J_{1,2n} + (1-(1-\beta^r)^n)\mathbf{y}^r \simeq \mathbf{y}^r \tag{13}
$$

3.3 Episode Retrieval

Learned REM-ART memorizes the weight vectors for the corresponding episodes. It can be used to retrieve all time sequential events from the observed partial cues. When REM-ART recognizes an episode, only the weight vector is required to recall time sequential events. This can be done simply by substituting $\mathbf{y}^e$ into $\mathbf{w}^e$, such as $\mathbf{y}^e = \mathbf{w}^e$. The reward weight vector $\mathbf{w}^r$ is independent of $\mathbf{w}^e$, and thus it is not needed when retrieving events. The event node with the highest activation value retrieves the input vector in F_1. The activation value of the retrieved node then becomes zero, and the next highest activation node retrieves the next sequential input recursively.

When REM-ART tries to predict an episode from incomplete cues, it recognizes events in F_2^e first. Recognized events in F_2^e are forwarded to reward channel in real time. At the same time, REM-ART loads all learned events and gives them to the reward channel. Then the reward channel generates the reward signals using both sets of data. Since h is the number of recognized events, it increases as a new input comes to input field F_1. To calculate n_f, the reward channel matches recognized events with learned events. For instance, input events matched with the fth episode increase the value of n_f. After the matching process, y_f^r can be calculated using Eq. (12), and the multi-channel input vector of $F_2 = \{F_2^e, F_2^r\}$ is used to calculate activation values for each episode. y_f^r is closer to one when events corresponding the fth episode enter the network. This generates a larger activation value for the fth episode, since the reward value y_f^r reinforces the activation of the fth episode node. This makes the REM-ART network more robust against noisy inputs.

To compare the activation values with EM-ART in the case when the input vector $\mathbf{y}$ in F_2 has only one event for the jth episode, the choice functions of the ith and the jth episodes in REM-ART are respectively as follows:

$$
\begin{aligned}
{}^rT_i &= \sum_{k=1}^{2} \gamma^k \frac{|\mathbf{y}^k \wedge \mathbf{w}_i^k|}{\alpha + |\mathbf{w}_i^k|} \\
&= \gamma^1 \frac{|\mathbf{y}^1 \wedge \mathbf{w}_i^1|}{N_1} + \gamma^2 \frac{|\mathbf{y}^2 \wedge \mathbf{w}_i^2|}{N_2} \\
&= \gamma^1 T_i + \gamma^2 \frac{N_2 - 2}{N_2} \\
{}^rT_j &= \sum_{k=1}^{2} \gamma^k \frac{|\mathbf{y}^k \wedge \mathbf{w}_j^k|}{\alpha + |\mathbf{w}_j^k|} \\
&= \gamma^1 \frac{|\mathbf{y}^1 \wedge \mathbf{w}_j^1|}{N_1} + \gamma^2 \frac{|\mathbf{y}^2 \wedge \mathbf{w}_j^2|}{N_2} \\
&= \gamma^1 T_j + \gamma^2 \\
\therefore {}^rT_j - {}^rT_i &= \gamma^1 (T_j - T_i) + \gamma^2 \frac{2}{N_2}
\end{aligned}
\tag{14}
$$

where T_i and T_j are the activation values of the ith and the jth episodes, respectively, in EM-ART. This shows the difference between T_i and T_j is decreased by γ^1, and the second term related with the reward signal is added to make the difference positive. Therefore, the reward signal weakens the activation of the episode not concerned with the target episode, while it reinforces the activation of the target episode.

REM-ART encodes and learns episodes in an unsupervised manner, and also it utilizes reinforcement learning for retrieval of the episodes. The reinforcement learning helps the network robustly retrieve the target episode exactly using the reward signal when it receives incomplete or noisy inputs.

4. Experiments

4.1 Experimental Setup

The proposed method was applied to predict which episode would be executed during a visual demonstration captured by a RGB-D camera, mounted on the robotic head of Mybot-KSR, developed in the RIT Lab, at KAIST. The robot is shown in Fig. 4. The Mybot-KSR recognizes the environment through the RGB-D camera. By assuming that objects were on the table, regions of the object hypothesis were detected by removing the plane that is detected using Random Sample Consensus (RANSAC) [13]. The feature of each object hypothesis was extracted by SIFT [15], [16]. The action on objects is related to a hand posture. Thus, the region of one hand near a certain object was found by a skin color detector and it was followed by feature extraction using SIFT. To enhance the accuracy of the action recognition, a speed term, given by the hand's distance between two consecutive frames, was concatenated to the SIFT feature.

The feature extractions of objects and an action were followed by general SVM for classification. The objects in this experiment were {*wateringpot, flowerpot, bottle, cup, toy, box, bread, toaster, dish*} and the actions were {*grasp, move, tilt, put down, push down*}. Using these recognition algorithms, a single image in video frames could be parsed into objects and their corresponding action, which is defined as an event. The task consists of a sequence of events. To evaluate how well the performed task is predicted, we defined four scenarios: 1) Water the flower; 2) Pour the contents of a bottle; 3) Sort the toys; and 4) Toast a slice of bread. Lists of time sequential events in individual episodes are provided in Table 1.

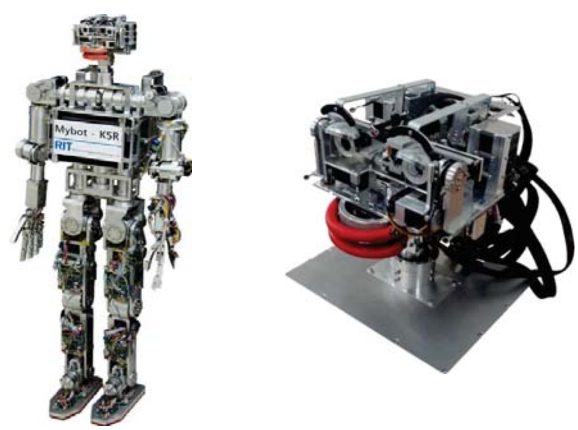


Fig. 4: Mybot-KSR and its robotic head.

The event used in these scenarios needed two input categories, the object and the action. Therefore, we used REM-ART architecture having two channels in the input

Table 1: The lists of scenarios and events.

Scenario	Event
Water the flower	Grasp a watering pot.
	Move the watering pot to the flowers.
	Tilt the watering pot to the flowers.
	Put down the watering pot.
Pour the contents of a bottle	Grasp a bottle.
	Move the bottle to the cup.
	Tilt the bottle to the cup.
	Put down the bottle.
Sort the toys	Grasp a toy.
	Move the toy to the box.
	Put down the toy in the box.
Toast a slice of bread	Grasp a bread on the dish.
	Move a bread to the knob of the toaster.
	Put down the bread in the toaster.
	Grasp the toaster.
	Push down the toaster.

field to store the information on objects and actions. The user acted the sequence of events in all scenarios in front of the humanoid robot. The robot then observed and perceived objects and actions. These sequential inputs are learned by REM-ART. After the episode learning, the user acted each scenario ten times again, and the robot predicted which episode was performed. For comparing the proposed REM-ART with EM-ART for the retrieval error rate, EM-ART was also learned using the same sequential inputs.

4.2 Experimental Results

Fig. 5 shows the recognized image from the RGB-D camera of the robotic head. Each row of Fig. 5 is the perceived sequential events when the user performed each scenario. The features extracted from RGB-D images were made of objects and actions, and they were used to learn and test the neural networks.

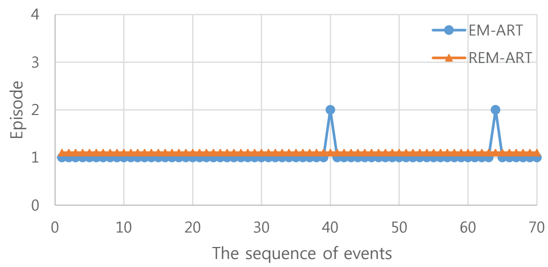
Fig. 5: The sequence of recognized images from *Water the flower* (top row), *Pour the contents of a bottle* (second row), *Sort the toys* (third row), and *Toast a slice of bread* (bottom row).

The prediction results of REM-ART and EM-ART are shown in Fig. 6. Since the sequential inputs are recognized from camera images, they contain many recognition errors. For instance, the robot can extract sequential inputs such as "Grasp a bottle", "Move a chair", and "Throw a pen" from the images. If the user acts scenario 1, these inputs are all noisy inputs. As shown in Fig. 6, EM-ART predicted wrong episodes as the sequence of events is provided to the inputs, because the noisy cues made episode prediction errors. However, REM-ART could predict the exact scenarios from these noisy inputs. The reward signal reinforced the activation of the desired episode node, and at the same time it attenuated the activation of the other episode nodes. This helps REM-ART to recognize the correct episode even though the input vector was disturbed from noises. The important point is that REM-ART becomes increasingly robust as the input sequence enters the input fields, because the reward signal of the desired episode is more and more strengthened whenever relevant sequential events enter. As shown in Fig. 6(b), the prediction result of REM-ART converged to scenario 2 immediately and it was not changed, but EM-ART predicted wrong scenarios after finding the correct scenario, because of error inputs.

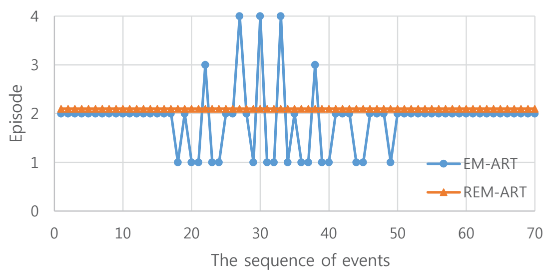
To analyze and compare the prediction accuracy of each model, the confusion matrices are provided in Fig. 7. The prediction accuracy of EM-ART is distributed from 52% to 97%. In particular, the prediction accuracy of scenario 4 is quite low, since this scenario has the largest number of events among all scenarios. As mentioned above, the events relevant to the target episode with a large number of events generate relatively small activation values for the episode. This leads to the other episode nodes being activated, and thus the prediction accuracy is reduced. Therefore, with the EM-ART it is difficult to predict the correct episode from all noisy inputs when the input vector is quite distorted in the real environment. On the other hand, the prediction accuracy of REM-ART is almost 100% for the four scenarios. This demonstrates that the REM-ART model is remarkably effective in predicting and retrieving the target episode from noisy inputs.

5. Conclusion

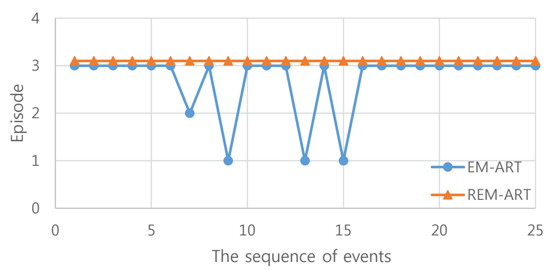
We proposed a novel Reward-based ART model called REM-ART. This model can store and retrieve episodes consisting of time sequential events. The proposed REM-ART could easily increase its memory size to learn events and episodes dynamically. The proposed REM-ART model has multi-channels for input and category fields, and thereby grants the neural network new functionalities. A newly added channel to a category field was designed to make the episodic memory learning model robust to noisy inputs. This channel is called the reward channel, which generates a reward signal. This reinforces the episode prediction accuracy of REM-ART, such that the robot can perceive contextual situations more robustly in the real environment. The experimental



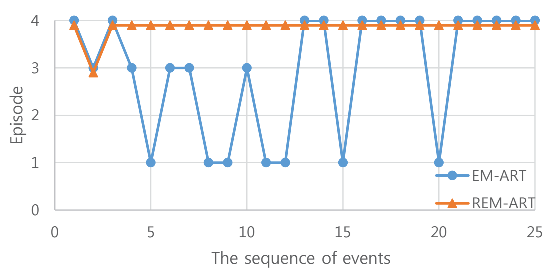
(a) Episode prediction in scenario 1.



(b) Episode prediction in scenario 2.



(c) Episode prediction in scenario 3.



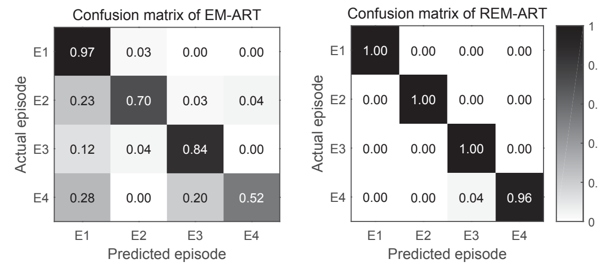
(d) Episode prediction in scenario 4.

Fig. 6: Episode prediction results from sequential input events of each scenario.

results demonstrated that the reward channel effectively reduces the prediction error of REM-ART regardless of noisy inputs.

Acknowledgement

This work was supported by the Technology Innovation Program, 10045252, funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).



(a) (b)

Fig. 7: Confusion matrices of the episode prediction using (a) EM-ART and (b) REM-ART.

References

[1] E. Tulving, *"Episodic and semantic memory 1,"* in *Organization of Memory*. New York: ACADEMIC PRESS, 1972.

[2] E. Tulving, *Elements of episodic memory*. New York: Oxford Univ. Press, 1983.

[3] A. Nuxoll and J. E. Laird, *"Extending cognitive architecture with episodic memory,"* in *Proc. 22nd Nat. Conf. Artif. Intell.*, 2007, pp. 1560-1564.

[4] A. Nuxoll and J. E. Laird, *"A cognitive model of episodic memory integrated with a general cognitive architecture,"* in *International Conference on Cognitive Modeling*, 2004.

[5] Matthew L. Lee and Anind K. Dey., *"Providing good memory cues for people with episodic memory impairment,"* in *Proceedings of the 9th international ACM SIGACCESS conference on Computers and accessibility*, ACM, 2007, pp. 131-138.

[6] W. C. Ho, K. Dautenhahn, and C. L. Nehaniv, *"Comparing different control architectures for autobiographic agents in static virtual environments,"* in *Proc. IVA*, 2003, pp. 182-191.

[7] S. T. Mueller and R. M. Shiffrin, *"REM-II: A model of the developmental co-evolution of episodic memory and semantic knowledge,"* in *Proc. Int. Conf. Develop. Learn.*, vol. 5. 2006, pp. 1-7.

[8] M. W. Howard and J. Lim, *"A distributed representation of temporal context,"* in *Journal of Mathematical Psychology*, vol. 46. 2001, pp. 269-299.

[9] A. B. Samsonovvich and G. A. Ascoli, *"A simple neural network model of the hippocampus suggesting its pathfinding role in episodic memory retrieval,"* in *Learn. Memory*, vol. 12, pp. 193-208, Mar. 2005.

[10] S. Becker and J. Lim, *"A computational model of prefrontal control in free recall: strategic memory use in the california verbal learning task,"* in *J. Cogn. Neurosci.*, vol. 15. 2005, pp. 821-832.

[11] W. Wang, B. Subagdja, A.-H. Tan, and J. A. Starzyk, *"Neural modeling of episodic memory: Encoding, retrieval, and forgetting,"* in *IEEE Trans. on Neural Netw. and Learning Syst.*, vol. 23, 2012.

[12] G. Carpenter, S. Grossberg, and D. Rosen, *"Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system,"* in *Neural Netw.*, vol. 4. 1991, pp. 759-771.

[13] M. A. Fischler and R. C. Bolles, *"Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,"* in *Commun. ACM*, vol. 24, no. 6, 1981, pp. 381-395.

[14] B.-S. Yoo, Y.-H. Yoo, W.-R. Ko, S.-J. Lee, S.-H. Cho and J.-H. Kim, *"Realization of task intelligence based on the iOA for assistive robots,"* in *Proc. International Conference on Artificial Intelligence (ICAI)*, Las Vegas, U.S.A., accepted, Jul, 2015.

[15] D. G. Lowe, *"Distinctive image features from scale-invariant keypoints,"* in *Int. J. Comput. Vision*, vol. 60, no. 2, 2004, pp. 91-110.

[16] Y.-H. Yoo and J.-H. Kim, *"Procedural memory learning from demonstration for task performance,"* in *Proc. IEEE. Sys. Man. Cybern*, Oct. 2015.