A Reinforcement Learning System Based on State Space Construction Using Fuzzy ART

Kunikazu Kobayashi[1], Shotaro Mizuno[2], Takashi Kuremoto[1], Masanao Obayashi[1]

1 Yamaguchi University, 2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan.
2 Graduate School of Yamaguchi University, 2-16-1 Tokiwadai, Ube, Yamaguchi 755-8611, Japan.
koba@yamaguchi-u.ac.jp

Abstract: A new reinforcement learning system using fuzzy ART (adaptive resonance theory) is proposed. In the proposed method, fuzzy ART is used to classify observed information and to construct effective state space. Then, profit sharing is employed as a reinforcement learning method. Furthermore, the proposed system is extended to the hierarchical structures for solving partially observable Markov decision process (POMDP) problems. Through various computer simulations using maze problems, it is confirmed that the proposed methods are effective to solve POMDP problems.

Keywords: Reinforcement learning, Fuzzy ART, MDP, POMDP, Profit sharing, Maze problem

1. Introduction

Reinforcement learning (RL) is learning from interaction with an environment, from the consequences of action, rather than from explicit teaching[1, 2]. The purpose of RL is to obtain as many as rewards as soon as possible.

Most RL is conducted within the mathematical framework of Markov decision processes (MDPs). MDPs involve a decision-making agent interacting with its environment so as to maximize the cumulative reward it receives over time. The agent observes aspects of the environment's state and selects actions. The agent may estimate a value function and use it to construct better and better decision-making policies over time. The above framework, however, cannot directly apply to partially observable MDPs (POMDPs)[3].

Recently, some researches for POMDPs are conducted[4, 5]. Most systems, however, are controlled by action groups designed by human. Therefore, as tasks and interactions between agents are more complicated, it is difficult to design the controller. The agent with self-learning and self-adapting ability is highly desirable.

In addition, if observed information has analog values, it is very time consuming that an agent obtains a desired value function. As a result, the construction of an appropriate state space for an agent becomes difficult.

In this paper, in order to avoid the above problems, a new RL system using fuzzy ART (Adaptive Resonance Theory)[6, 7] is proposed. In the proposed system, fuzzy ART is used to classify observed information and to construct effective state space. Then, profit sharing[8], which is one of the experience based RL methods, is employed as a learning method. Furthermore, the proposed system is extended to the hierarchical structures using HQ-learning (hierarchical Q-learning)[9] and an AQ controller (abstract Q controller) used in SSS (self-segmentation of sequences) algorithm[10] for solving POMDP problems. Through various computer simulations using maze problems, it is confirmed that the proposed methods are effective to solve POMDP problems.

In the reminder of this paper, RL systems for dealing with POMDP problems will be developed. In section 2, the outline of fuzzy ART is introduced. Then, in section 3, a RL system using fuzzy ART is proposed and its performance is evaluated through computer simulation. Furthermore, in section 4, hierarchical RL systems using HQ-learning and the AQ controller are proposed and their performance is also evaluated through computer simulation. Finally, section 5 concludes and gives a direction for future research.

2. Fuzzy ART

ART was developed by S. Grossberg as a theory of human cognitive information processing to solve the learning instability problem suffered by standard feed-forward networks[6]. Fuzzy ART[7] is a synthesis of ART and fuzzy logic[11]. It consists of an input layer (F_1) and a category layer (F_2) as shown in Fig. 1. The neuron i in F_1 layer is connected to all the neurons in F_2 layer through a top-down weight w_i and also the neuron j in F_2 layer is connected to all the neurons in F_1 layer through a bottom-up weight w_j.

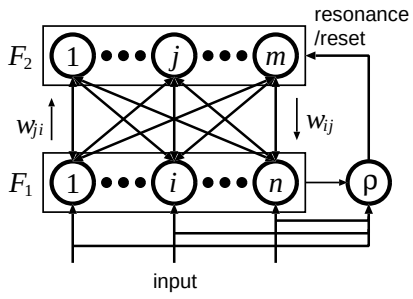


Fig. 1: Architecture of fuzzy ART

The algorithm of fuzzy ART is described briefly. Then, a choice function for category neuron j in F_2 layer is calculated by:

$$T_j = \frac{|I \wedge w_j|}{\alpha + |w_j|} \qquad (j = 1, \cdots, m), \tag{1}$$

where I is an input given to F_1 layer, a is a positive choice parameter, $\wedge$ means the fuzzy AND operator, i.e. $p \wedge q =$

$\min(p, q)$, and $|u|$ refers to the norm of a vector u, i.e. $|u| = \sum_{i=1}^{n} u_i$. The winner category neuron j in F_2, which has the maximum T_j, is only activated. After that, resonance occurs if a match function A_J in the orienting subsystem meets the vigilance criteria:

$$A_J = \frac{|I \wedge w_J|}{\alpha + |I|} \geq \rho, \tag{2}$$

where $\rho \in [0, 1]$ is a vigilance parameter. Otherwise, a reset of F_2 occurs, i.e. the winner category neuron J is deactivated and a new category which meets the above vigilance criteria is searched. Finally, weight w_J is updated by:

$$w_J \leftarrow \beta(I \wedge w_J) + (1 - \beta)w_J, \tag{3}$$

where β is a learning rate parameter.

In general, the input I is preprocessed to avoid category proliferation using complement coding. For input $a \in [0, 1]^n$, the complement coded input, I, is defined as follows:

$$I = [a_1, \cdots, a_n, a_1^c, \cdots, a_n^c] \in R^{2n}, \tag{4}$$

where $a_i^c \equiv 1 - a_i$ and the relation $|I| = n$ holds.

3. Reinforcement Learning System Using Fuzzy ART

3.1 Architecture

The architecture of a RL system proposed in this paper is illustrated in Fig. 2. It comprises three layers, i.e., perceptual input layer, state category layer, and action layer. The first two layers, perceptual input and state category layers, correspond to fuzzy ART. The enormous inputs from an environment are categorized and consequently the state space of inputs is compressed. Then, the action learning is conducted in the action layer using profit sharing.

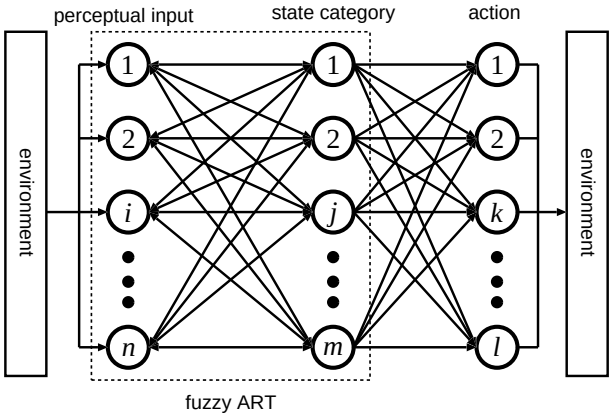


Fig. 2: Architecture of the proposed RL system using fuzzy ART

3.2 Action selection

After the winner category neuron J is selected in fuzzy ART, an action is selected based on softmax action selection. In this paper, Boltzmann distribution defined in eqn. (5) is employed as softmax action selection.

$$p(w_{kJ}) = \frac{\exp(w_{kJ}/T)}{\sum_k \exp(w_{kJ}/T)}, \tag{5}$$

where w_{kJ} is a weight between action neuron k and category neuron J, and T is a temperature parameter. High temperatures cause the actions to be nearly equi-probable. In the limit as $T \rightarrow \infty$, it becomes the same as random action selection. Low temperatures cause a greater difference in selection probability for actions that differ in their value estimates. In the limit as $T \rightarrow 0$, it becomes the same as greedy action selection.

3.3 Learning method

In this paper, profit sharing[8] is employed as a weight updating method. Here, profit sharing is one of the reinforcement learning methods that allows agents to learn effective behaviors from their experiences within dynamic environments. Profit sharing is different from other reinforcement learning methods such as Q-learning[12], which is one of the dynamic programming based reinforcement learning methods and makes the assumption that an environment can be modeled by a MDP. Rules on an episode, the weight w_{kJ}, is reinforced by:

$$w_{kJ} \leftarrow w_{kJ} + \alpha_{PS}\gamma_{PS}^{h-1} r_{PS}, \tag{6}$$

where α_{PS} represents a learning rate, r_{PS} is a reward, γ_{PS} is a decreasing rate, and h is the length of a PS (profit sharing) table. The PS table contains a history of pairs of category and action.

3.4 Computer simulation

The performance of the proposed system is evaluated by comparison with a Q(λ)-learning[12] system using fuzzy ART and a normal Q(λ)-learning system through computer simulation. Note that Q(λ)-learning is an extension of Q-learning incorporating eligibility traces[2].

In this simulation, two maze problems shown in Figs. 3 and 4 are used for evaluation. The maze problem is to find a shortest route from a starting point S to a goal G. In Figs. 3 and 4, solid and open squares mean walls and passages, respectively.

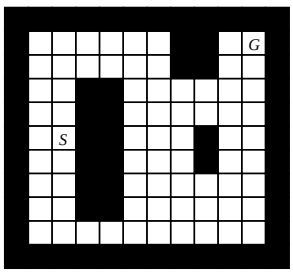


Fig. 3: Maze #1 (11x12)

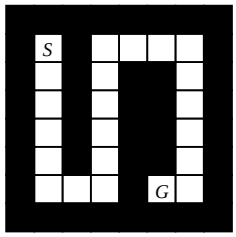
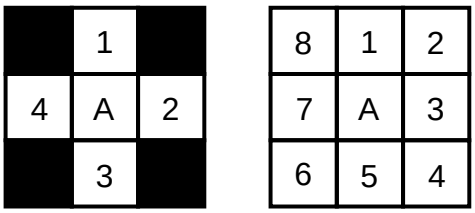


Fig. 4: Maze #2 (8x8)

It is assumed that the agent can perceive 4-neighbors (Fig. 5(a)) or 8-neighbors (Fig. 5(b)) but cannot recognize

the absolute coordination. Both mazes contain POMDP environment if the agent can only perceive 4-neighbors. If the agent can perceive 8-neighbors, maze #1 becomes MDP environment but maze #2 is still a POMDP. Furthermore, it is assumed that the agent can move one square in any of four directions (north, west, south and east) at each step, but it cannot change its orientation according to the movement direction.



(a) 4-neighbors (b) 8-neighbors

Fig. 5: Perceivable grids of agent A

The value of parameters is set as follows: vigilance parameter; $\rho = 0.8$, choice parameter; $\alpha = 0.01$, learning rate parameter; $\beta = 1.0$, learning rate; $\alpha_{PS} = 0.1$, decreasing rate; $\gamma_{PS} = 0.8$, eligibility rate; $\lambda = 0.9$. The reward r_{PS} is defined as 10 if the agent arrives at the goal, and $r_{PS} = -0.1$ otherwise.

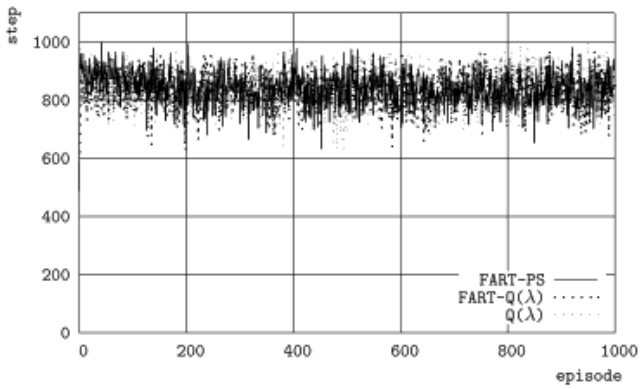


Fig. 6: Results of three systems with four inputs for maze #1

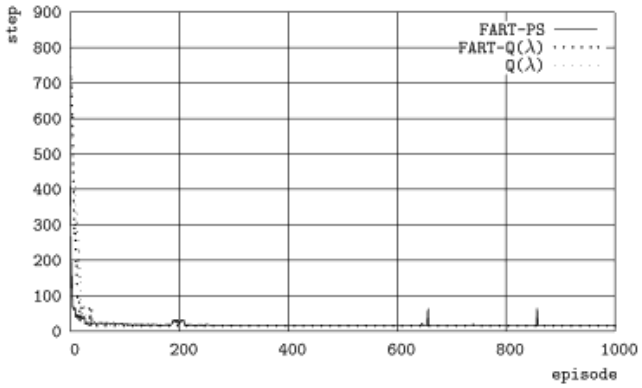


Fig. 7: Results of three systems with eight inputs for maze #1

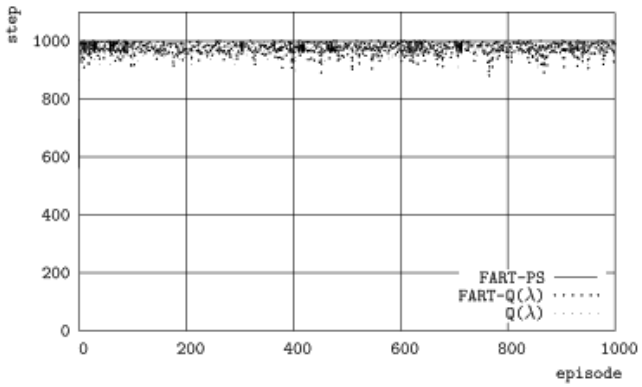


Fig. 8: Results of three systems with four inputs for maze #2

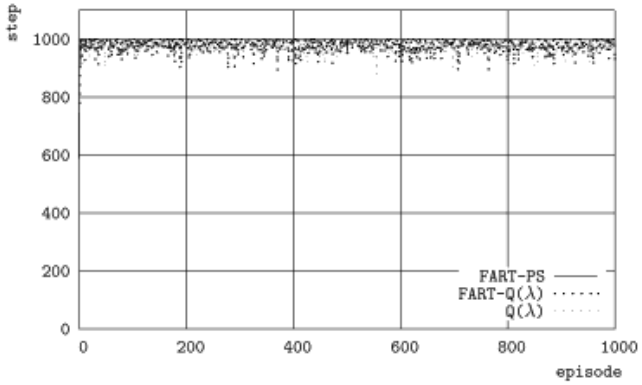


Fig. 9: Results of three systems with eight inputs for maze #2

The simulation results of three systems are shown in Figs. 6 to 9. Figures 6 and 7 depict the curves for maze #1 and Figs. 8 and 9 are those for maze #2. The curve illustrates the number of steps that the agent reaches the goal against the number of episodes. In these figures, the proposed system is denoted as FART-PS, and the Q(λ)-learning system using fuzzy ART and the normal Q(λ))-learning system are FART- Q(λ)) and Q(λ)), respectively. The number of steps to reach the goal is averaged in 20 trials.

For maze #1, all the system with four inputs is not converged at all (See Fig. 6) but those with eight inputs is converged quickly because the environment is an MDP (See Fig. 7). On the other hand, for maze #2, all the three systems with both four and eight inputs cannot solve the problem (See Figs. 8 and 9). From the above results, it is shown that the proposed system cannot solve a difficult POMDP problem. In the next section, hierarchical RL system is proposed to solve such problem.

4. Hierarchical Reinforcement Learning System

A POMDP problem may be divided into some MDP problems in order to solve an aliasing problem. In this paper, a hierarchical RL system is proposed as one of the above solution.

4.1 Introducing a HQ-table

To extend the proposed system to a hierarchical structure, a HQ-table in HQ-learning (hierarchical Q-learning)[9] is introduced. Note that HQ-learning is a hierarchical extension of Q(λ)-learning[12] designed to solve certain types of POMDP problems.

The architecture of the proposed hierarchical RL system using the HQ-table is illustrated in Fig. 10. In this figure, all the subagents are connected in a sequential way. Each subagent has the reinforcement learning system proposed in section 3, an HQ-table and a control transfer unit except for the last agent. The HQ-table stores estimated subgoal values and is used to generate a subgoal once the agent is activated.

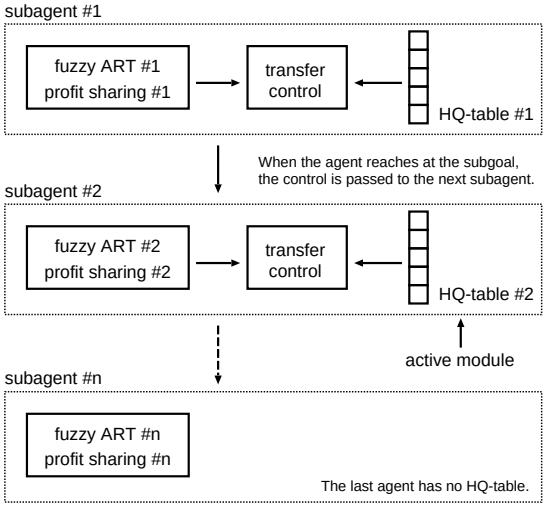


Fig. 10: Architecture of the proposed hierarchical RL system using the HQ-table

The algorithm of the proposed system is described below.

Algorithm #1

Step 1. Decide a subgoal.

Step 1-1. Observe an input I from the environment.

Step 1-2. Assume a winner category J selected by fuzzy ART as a state s.

Step 1-3. Select an action a based on an action selection method.

Step 1-4. Obtain a reward r from the environment. If the agent arrives at the goal or if it arrives at a subgoal and there is the next subagent, go to Step 2.

Step 1-5. Store a pair of state s and action a in a PS-table.

Step 1-6. If the number of steps reaches the maximum number go to Step 5, otherwise go back to Step 1-1.

Step 2. Update the rules contained in an episode using eqn. (6) and the PS-table.

Step 3. Initialize the PS-table.

Step 4. If there is the next subagent the control is passed to it and go back to Step 1.

Step 5. Evaluate the sequences on all the subgoals using HQ-tables.

For a maze with POMDP, some subgoals are placed on the maze so as to divide a POMDP problem into some MDP problems. As a result, the system could solve such maze.

4.2 Introducing an AQ controller

In addition, an AQ controller, which is used in SSS (self-segmentation of sequences) algorithm[10], is incorporated into the proposed system described in section 4.1. Note that SSS can segment action sequences to reduce non-Markov temporal dependencies and to facilitate the learning of the overall task. Furthermore, in this paper, an idea of eligibility traces is introduced. Here, eligibility traces are one of the basic mechanisms of reinforcement learning. In the popular TD(λ) algorithm[2], the refers to the use of an eligibility trace. Almost any temporal-difference methods, e.g., Q-learning[12] and Sarsa[13, 2], can be combined with eligibility traces to obtain a more general method that may learn more efficiently.

The architecture of the proposed hierarchical RL system using the AQ controller is illustrated in Fig. 11. As shown in this figure, the AQ controller is added to the hierarchical RL system in Fig. 10.

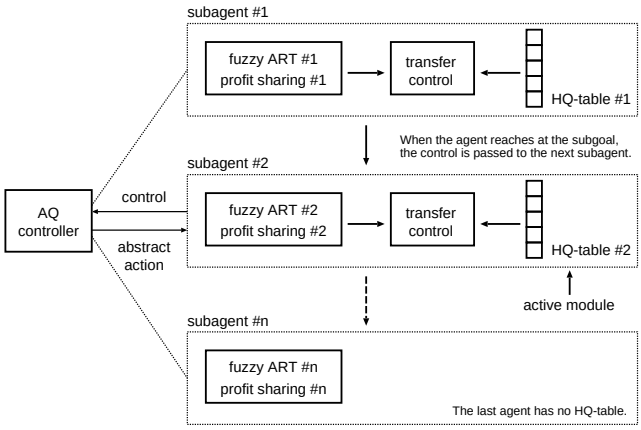


Fig. 11: Architecture of the proposed hierarchical RL system using the AQ controller

The algorithm of the proposed system is described below.

Algorithm #2

Step 1. Except for the initial step, an eligibility trace function $\eta(\tilde{s}, a_i)$ is updated by:

$$\eta(\tilde{s}, a_i) = \begin{cases} 1 & (a_i = a) \\ 0 & (a_i \neq a) \end{cases} \tag{7}$$

Step 2. Check the active categories using fuzzy ART. If there is no active category a new one is created.

Step 3. Select an action a' based on Max-Boltzmann method: with probability $p_{\max}$ take an action according to greedy selection method, with probability $1 - p_{\max}$ take an action according to Boltzmann distribution in eqn. (5).

Step 4. Decide a state s' based on the action a'.

Step 5. The action-value function $AQ(s_i, a_k)$ and the eligibility trace function $\eta(s_i, a_k)$ is updated by:

$$AQ(s_i, a_k) \leftarrow AQ(s_i, a_k) + \alpha_{AQ}\delta\eta(s_i, a_k), \quad (8)$$
$$\eta(s_i, a_k) \leftarrow \gamma_{AQ}\lambda\eta(s_i, a_k), \quad (9)$$
$$\delta = R_i + \gamma_{AQ}^{m_i} AQ(s', a') - AQ(\tilde{s}, a_i), \quad (10)$$

where α_{AQ} is a learning rate, γ_{AQ} is a decreasing rate, λ is an eligibility rate, m_i is the number of steps requiring from the state $\tilde{s}$ to s', R_i is a discounted cumulative reward defined by:

$$R_i = r_t + \gamma_{AQ} r_{t+1} + \cdots + \gamma_{AQ}^{t+m_i-1} r_{t+m_i-1}. \quad (11)$$

If the number of steps reaches the maximum number go to Step 11.

Step 6. $\tilde{s} \leftarrow s'$ and $a \leftarrow a'$.

Step 7. Decide a subgoal.

Step 7-1. Observe an input I from the environment.

Step 7-2. Assume a winner category J selected by fuzzy ART as a state s.

Step 7-3. Select an action a based on an action selection method.

Step 7-4. Obtain a reward r from the environment. If the agent arrives at the goal or if it arrives at a subgoal and there is the next subagent, go to Step 8.

Step 7-5. Store a pair of state s and action a in a PS-table.

Step 7-6. If the number of steps reaches the maximum number go to Step 1, otherwise go back to Step 7-1.

Step 8. Update the rules contained in an episode using eqn. (6) and the PS-table.

Step 9. Initialize the PS-table.

Step 10. If there is the next subagent the control is passed to it and go back to Step 1.

Step 11. Evaluate the sequences on all the subgoals using HQ-tables.

The AQ controller decides which subagent should be activated under each situation. The AQ controller takes a crucial role in effectively reducing the number of subagents.

4.3 Computer simulation

In this simulation, a fairly complicated maze shown in Fig. 12 is used. This maze problem cannot be solved by single agent. Two proposed systems are compared with the standard HQ-learning system. Here, two proposed systems refer to the proposed hierarchical RL system using the HQ-table (See 4.1) and that using the AQ controller (See 4.2).

The value of parameters is set as follows: the number of subagents; $n = 4$, vigilance parameter; $\rho = 0.9$, choice parameter; $\alpha = 0.01$, learning rate parameter; $\beta = 1.0$,

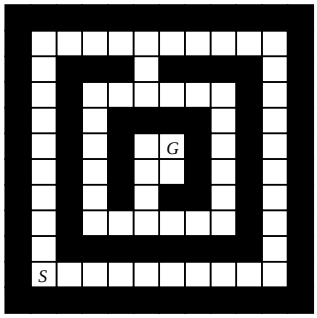


Fig. 12: Maze #3 (12x12)

learning rate; $\alpha_{PS} = 0.2$, $\alpha_{HQ} = 0.2$, $\alpha_{AQ} = 0.1$, decreasing rate; $\gamma_{PS} = 0.8$, $\gamma_{HQ} = 0.9$, $\gamma_{AQ} = 0.9$, eligibility rate; $\lambda_{HQ} = 0.9$, $\lambda_{AQ} = 0.9$. It is assumed that the reward r is 10 if the agent arrives at a subgoal or the goal, $r = -0.1$ otherwise and the reward for HQ-tables is 100 if the agent reaches the goal. The probability $p_{\max}$ is initialized as 0.9 and then is gradually increased to 1.0. The temperature parameter T is initialized as 0.1 and then is gradually decreased to 0.01.

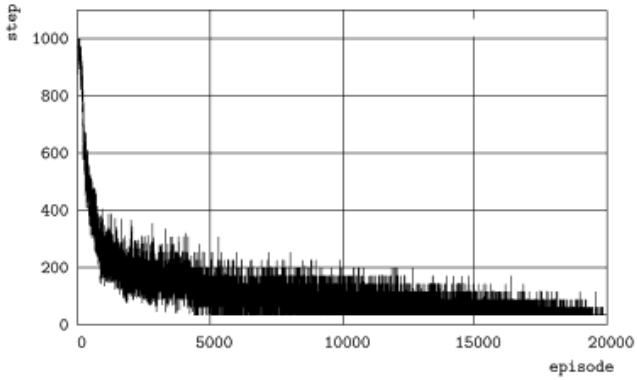


Fig. 13: Result of the proposed hierarchical RL system using the HQ-table (four subagents)

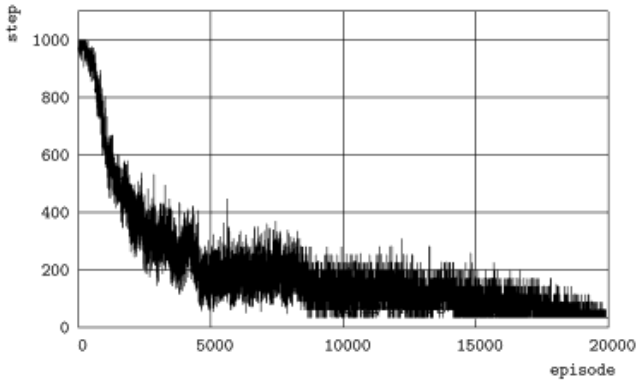


Fig. 14: Result of the proposed hierarchical RL system using the AQ controller (four subagents)

The simulation results of three systems are illustrated in Figs. 13 to 15. In these figures, the number of steps to reach the goal is averaged in 40 trials. As shown in these figures, the convergence speed of the proposed hierarchical RL sys-

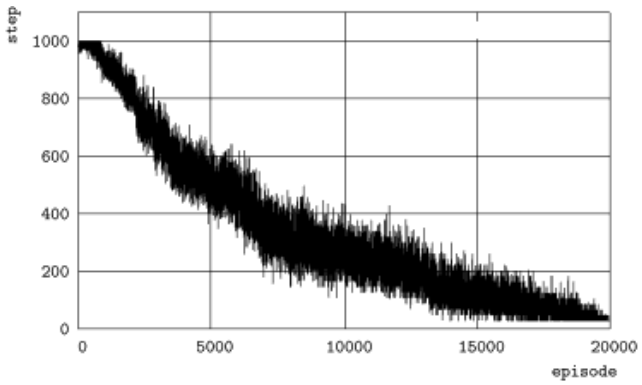


Fig. 15: Result of HQ-learning system (four subagents)

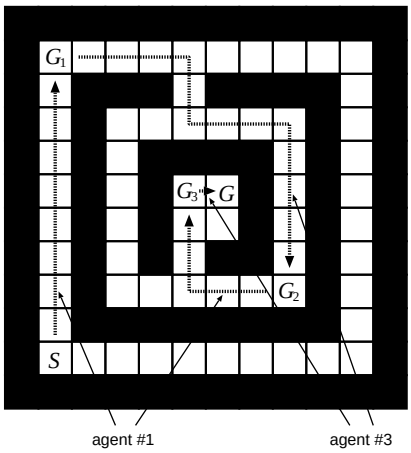


Fig. 16: A solution of the proposed hierarchical RL system using the AQ controller

tem using the HQ-table is the fastest among three systems and HQ-learning system shows the worst performance. Figure 16 shows a solution for maze #3 using the proposed hierarchical RL system using the AQ controller. This figure illustrates the system could find a route of maze #3 by only two subagents (#1 and #3) out of four subagents using the AQ controller. It is shown that the proposed hierarchical RL systems can solve POMDP problems.

5. Summary

In this paper, the RL system using fuzzy ART was proposed. Fuzzy ART is used to classify the enormous inputs from an environment and consequently to compress the state space of inputs.

In order to solve POMDP problems, the hierarchical RL systems incorporating the HQ-table and the AQ controller were also proposed. In the hierarchical RL systems, a POMDP problem is divided into some MDP ones in order to solve an aliasing problem.

Through various computer simulations using MDP and POMDP maze problems, the performance of the proposed system was evaluated. As a result, it is verified that the proposed systems are outperformed as compared with the conventional RL systems, i.e. the Q(λ)-learning system and the HQ-learning system.

In the future, the proposed system is applied to multi-agent problems.

References

[1] Kaelbling, L.P., Littman, M.L., Moore, A.W., Reinforcement Learning: A Survey, Journal of Artificial Intelligence Research, Vol. 4, pp. 237-285, 1996.

[2] Sutton, R.S., Barto, A.G., Reinforcement Learning: An Introduction, MIT Press, 1998.

[3] Monahan, G.E., A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms, Management Science, Vol. 28, No. 1, pp. 1-16, 1982.

[4] Kimura, H., Miyazaki, K., Kobayashi, S., Reinforcement Learning in POMDPs with Function Approximation, Proceedings of the 14th International Conference on Machine Learning, pp. 152-160, 1997.

[5] Kaelbling, L.P., Littman, M.L., Cassandra, A.R., Planning and Acting in Partially Observable Stochastic Domains, Artificial Intelligence, Vol. 101, pp. 99-134, 1998.

[6] Grossberg, S., Adaptive Pattern Classification and Universal Recoding, I: Parallel Development and Coding of Neural Feature Detectors, Biological Cybernetics, Vol. 23, pp. 121-134, 1976.

[7] Carpenter, G.A., Grossberg, S., Rosen, D. B., Fuzzy ART: Fast Stable Learning and Categorization of Analog Patterns by an Adaptive Resonance System, Neural Networks, Vol. 4, pp. 759-771, 1991.

[8] Grefenstette, J.J., Credit Assignment in Rule Discovery Systems Based on Genetic Algorithms, Machine Language, Vol. 3, pp. 225-245, 1988.

[9] Wiering, M., Schmidhuber, J., HQ-learning, Adaptive Behavior, Vol. 6, No. 2, pp. 219-246, 1997.

[10] Sun, R., Sessions, C., Self-segmentation of Sequences: Automatic Formation of Hierarchies of Sequential Behaviors, IEEE Transactions on Systems, Man, and Cybernetics: Part B Cybernetics, Vol. 30, No. 3, pp. 403-418, 2000.

[11] Zadeh, L.A., Outline of a New Approach to the Analysis of Complex Systems and Decision Process, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-3, No. 1, pp. 28-44, 1973.

[12] Watkins, C.J.C.H., Dayan, P., Q-Learning, Machine Learning, Vol. 8, pp. 55-68, 1992.

[13] Rummery, G.A., Niranjan, M., On-line Q-learning Using Connectionist Systems, Technical Report CUED/F-INFENG/ 166, Cambridge University Engineering Department, 1994.