International Conference on Control, Automation and Systems 2010
Oct. 27-30, 2010 in KITEX, Gyeonggi-do, Korea

A Reinforcement Learning System Embedded Agent with Neural Network-Based Multi-Valued Pattern Memory Structure

Masanao Obayashi[1], Tomohiro Nishida[1], Takashi Kuremoto[1], Kunikazu Kobayashi[1] and Liang-Bing Feng[1]

[1] Division of Computer Science & Design Engineering, Yamaguchi University, Ube, Japan
{m.obayas, wu, koba, n007we}@yamaguchi-u.ac.jp)

Abstract: This paper concerns about a way of intellectualization of robots (called "agent" here). Human learns incidents by own actions and reflects them on the subsequent actions as own experiences. These experiences are memorized in his/her brain and recollected and reused if necessary. This research incorporates such an intelligent information processing mechanism, and applies it to an autonomous agent that has three main functions that is, learning, memorization and associative recollection. In the proposed system, an actor-critic type reinforcement learning method is used for learning. For memorization, we introduce the chaotic auto-associative model that is proposed by Chartier, and that is also used like mutual associative memory system. Moreover, to deal with the increase of information, the memory part has an adaptive hierarchical layered structure of the memory module that consists of chaotic neural networks, especially for multi-valued pattern. Finally, the effectiveness of this proposed method is verified through the simulation applied to the maze-searching problem.

Keywords: Reinforcement learning, Multi-valued associative memory, Chaotic neural network, Intelligent agent

1. INTRODUCTION

Reinforcement learning (RL) is a framework for an agent to learn the choice of an optimal action based on a reinforcement signal [1]. It has been applied to a variety of problem such as autonomous robot navigation and nonlinear control and so on. On the other hand, there are many researches for associative memory (ASM) using chaotic neural network [2], one of the mechanism for intellectualization of the robots (called "agent" here). However, there are few researches about intellectualization of agents using both RL and ASM. One of such researches is the research of Obayashi, *et al*. [3]. Regrettably, there exists a practical problem in their proposed system, that is, it can't deal with multi-valued patterns required in the real environment.

In this study, we deal with the multi-valued patterns using the associative chaotic neural network (ACNN) proposed by Chartier *et al.* [4,5] as a storage mechanism of results of RL. However, the storage capacity of ACNN is small, it is not suitable for working alone. So, we made up the hierarchical memory structure by making use of ACNN. Finally it is verified that our proposed method is useful through the computer simulation for a maze searching problem.

2. PROPOSED SYSTEM STRUCTURE

The proposed system consists of two parts: learning and memory. Fig. 1 shows its overall structure. The memory consists of short-term memory (STM) and long-term memory (LTM).

Learning sector: actor-critic system is adopted. It learns the choice of action to maximize the total predictive rewards obtained over the future considering the environmental information (s) and reward (r) as result of action (a).

STM sector: it memorizes the learning path of the information (environmental information and action) obtained in Learning part. Unnecessary information is forgotten and useful information is stored.

LTM sector: it memorizes only the enough sophisticated and useful experience in STM.

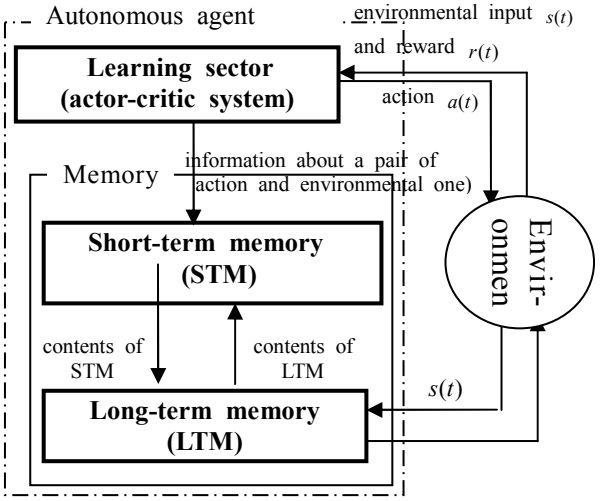


Fig. 1 Our proposed system

3. ACTOR-CRITIC REINFORCEMENT LEARNING SYSTEM

The actor-critic reinforcement learning system is shown in Fig. 2.

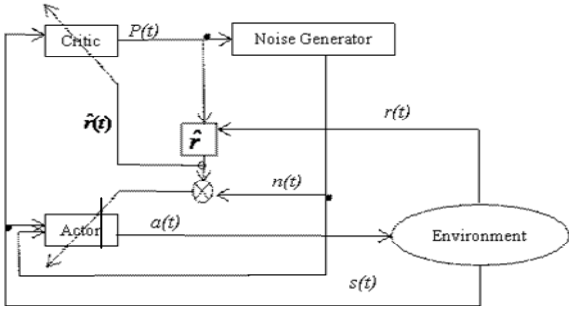


Fig. 2 Structure of the actor-critic system

3.1. Structure and learning of critic

3.1.1 Structure

Function of the critic is calculation of $P(t)$: the

prediction value of sum of the discounted rewards that will be gotten over the future and of its prediction error. These are shortly explained as follows:

The sum of the discounted rewards that will be gotten over the future is defined as $V(t)$.

$$V(t) \equiv \sum_{n=0}^{\infty} \gamma^n \cdot r(t+n), \quad (1)$$

where γ ($0 \le \gamma < 1$) is constant called discount rate. Eq. (1) is rewritten as

$$V(t) = r(t) + \gamma V(t+1). \quad (2)$$

Here the prediction value of $V(t)$ is defined as $P(t)$. The prediction error $\hat{r}(t)$ is expressed as follows:

$$\hat{r}(t) = r(t) + \gamma P(t+1) - P(t). \quad (3)$$

The parameters of the critic are adjusted to reduce this prediction error $\hat{r}(t)$. The prediction error $P(t)$ is calculated as follows:

$$P(t) = \sum_{j=0}^{J} w_j y_j(t). \quad (4)$$

Here J : number of nodes in the middle layer of the critic, w_j : weight of the jth output, y_j : jth output of the middle layer of the critic. The construction of the critic is also consisted of the RBFN as shown in Fig. 3.

3.1.2 Learning

Learning of critic is done by using commonly used Back Propagation method which makes prediction error $\hat{r}(t)$ goes to zero. Updating rule of parameters are as follows:

$$\Delta\omega_i^c = -\eta_c \cdot \frac{\partial \hat{r}_t^2}{\partial \omega_i^c}, \quad (i = 1, \cdots, J). \quad (5)$$

3.2. Structure and learning of actor

3.2.1 Structure

Fig.4 shows the construction of the actor. The actor is basically consisted of Radial Basis Function Network. The jth basis function of the middle layer node is as follows:

$$y_j = \exp\left[-\sum_{i=1}^{n} (x_i - m_{ij})^2 \Big/ \sigma_{ij}^2 \right], \quad (6)$$

$$u_k(t) = \sum_{j=1}^{J} w_{kj} y_j(t) + n(t), \quad (k = 1, \cdots, K). \quad (7)$$

Here y_j : jth output of the middle layer, m_{ij} , σ_{ij}^2 : center, dispersion for ith input of jth basis function respectively. K: number of the actions, n_k : additive noise to kth output, u_k : representative value of kth action, w_{kj} : connection weight from jth node of the middle layer to kth output.

3.2.2 Noise generator

Noise generator let the output of the actor have the diversity by adding the noise to it. It comes to realize the learning of the trial and error. Calculation of the noise $n(t)$ is as follows:

$$n(t) = n_t = noise_t \cdot \min(1, \exp(-P(t))), \quad (8)$$

where $noise_t$ is uniformly random number of $[-1, 1]$. As the $P(t)$ will be bigger, the noise will be smaller. This leads to the stable learning of the actor.

3.2.3 Learning

Parameters of actor, ω_{kj}^a ($j = 1, \cdots, J, k = 1, \cdots, K$) , are adjusted by using output u_k of actor and noise n.

$$\Delta\omega_{kj}^a = \eta_a \cdot n_t \cdot \hat{r}_t \cdot \frac{\partial u_k}{\partial \omega_{kj}^a}, \quad (9)$$

η_a (> 0) is the learning coefficient. Eq. (9) means that $(-n_t \cdot \hat{r}_t)$ is considered as error, ω_i^a is adjusted opposite to sign of $(-n_t \cdot \hat{r}_t)$.

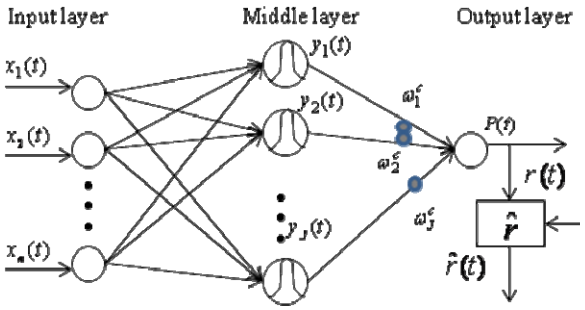


Fig. 3 Construction of critic

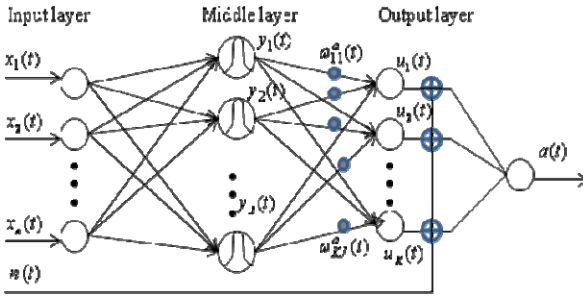


Fig. 4 Construction of actor

3.3 Action selection

The action a_b on time t is selected stochastically using Gibbs distribution Eq. (10).

$$P(a_b \mid \mathbf{x}(t)) = \frac{\exp(u_b(t)/T)}{\sum_{k=1}^{K} \exp(u_k(t)/T)}. \quad (10)$$

Here $P(a_b \mid \mathbf{x}(t))$: bth action selection probability, T : positive constant called temperature constant.

4. MEMORY SYSTEM

In this paper, we use an associative memory model as a tool to memorize the optimal path of the maze to be learned. To memorize the plural path and retrieve them, the memory model should behave itself chaotically.

4.1 Associative Chaotic Neural Network

Charties's chaotic neural network model shown in Fig. 5 can deal with the multi-valued patterns using the associative chaotic neural network (ACNN) [4]. The dynamical equation of it is described as Eq. (11).

$$\forall_{i,\ldots,N}, \mathbf{X}_{i[t+1]} = f(\mathbf{b}_i)$$

$$= \begin{cases} 1, & if\ \mathbf{b}_i > 1 \\ -1, & if\ \mathbf{b}_i < -1 \\ (\delta+1)\,\mathbf{b}_i - \delta\,\mathbf{b}_i^3 & \text{otherwise} \end{cases}, \quad (11),$$

where $\mathbf{b}$ is activated vector described as

$$\mathbf{b} = \mathbf{W}_{[k]}\mathbf{X}_{[t]} \quad . \quad (12)$$

$X_{i[0]}$: initial value of i-th neuron, $X_{i[t]}$: value of i-th neuron at time t, N: number of neurons of the network, δ: positive parameter of neuron, $W_{[k]}$: connection weight matrix consisting of stored patterns, k: number of learning. Fig. 6 shows shapes of output functions of Eq. (11) for δ=0.2 and δ=1.7. However, the Chartier's model does not behave chaotically for any values of δ. Therefore we improved Chartier's model so as to work chaotically as following next section.

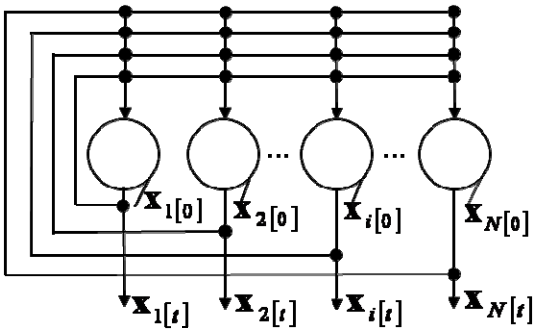


Fig. 5 Structure of Chartier's auto associative memory model

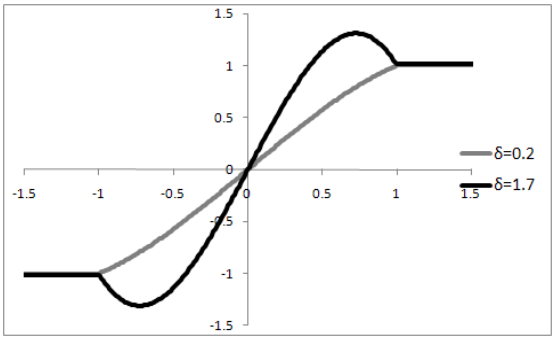


Fig. 6 Output function of Chartier's unit

4.2 Improved Associative Chaotic Neural Network

The bidirectional associative chaotic neural network proposed by Chartier *et al.* [4, 5] is one of the multi-valued pattern treatable CNN. However, their network doesn't work chaotically in the case of their construction as auto-associative chaotic neural network. So, we improve it so as to work chaotically. The improved dynamics of Chartier's auto-associative memory model are as follows,

$$\forall_{i,\ldots,N}, \mathbf{X}_{i[t+1]} = f(\mathbf{b}_i)$$

$$= \begin{cases} 1, & if\ \mathbf{d}_i > 1 \text{ and } (\mathbf{b_i} < -1 or 1 < \mathbf{b_i}) \\ -1, & if\ \mathbf{d}_i < -1 \text{ and } (\mathbf{b_i} < -1 or 1 < \mathbf{b_i}), \\ \mathbf{d}_i & \text{otherwise} \end{cases}$$

(13)

$$\mathbf{b} = \mathbf{W}\mathbf{X}_{[t]}, \quad (14)$$

$$\mathbf{d}_i = (\delta+1)\,\mathbf{b}_i - \delta\,\mathbf{b}_i^3 \ . \quad (15)$$

Figs. 7 ~ 10 show the output functions and its attractors. From these figures, we find that, in the case of $\delta = 0.2$, the model works non-chaotically, however in the case of $\delta = 1.7$, it works chaotically. Fig. 11 shows the stored patterns for Chartier's network model and our improved network model. Behavior of Chartier's model is shown in Fig. 12, it shows the network converges to one of stored patterns. On the other hand, our Chartier's improved model works chaotically as shown in Fig. 13.

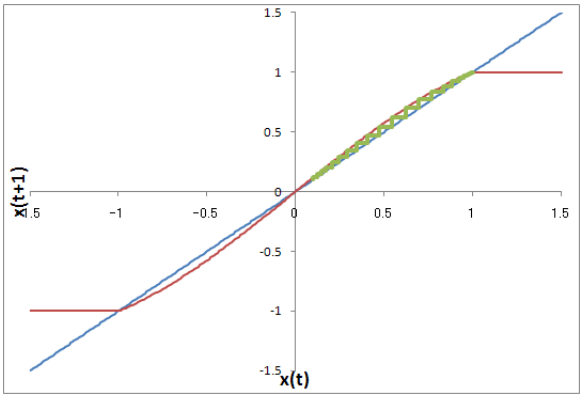


Fig. 7 Output function of our Chartier's improved unit ($\delta = 0.2$)

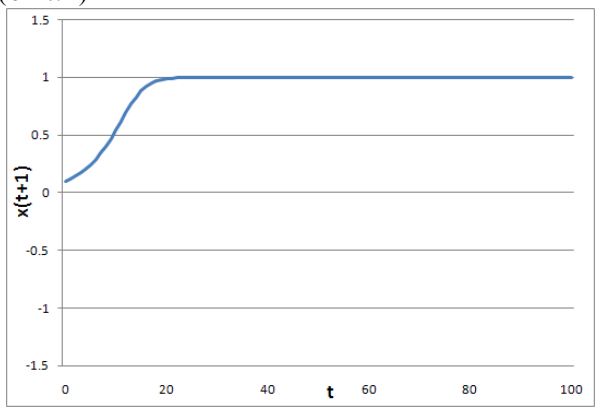


Fig. 8 Attractor of our Cartier's improved unit ($\delta = 0.2$)

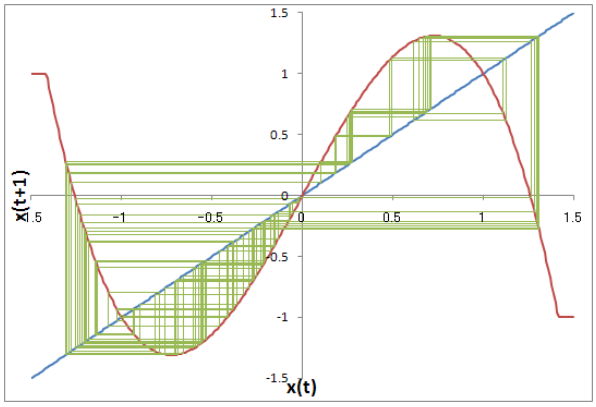


Fig. 9 Output function of our Chartier's improved unit ($\delta = 1.7$)

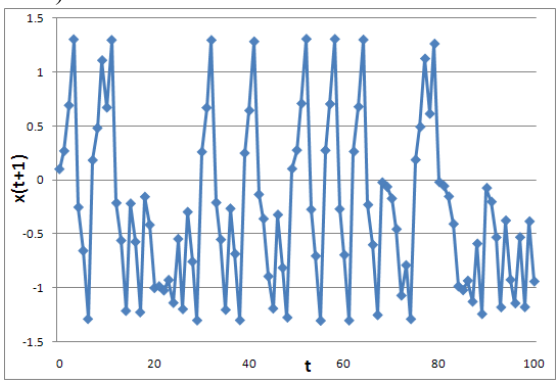


Fig. 10 Attractor of our Cartier's improved unit ($\delta = 1.7$)

4.3 Network control

Here, network control is defined as control which makes transition of network from chaotic state to non-chaotic one and vice versa. The network control algorithm of the improved ACNN is shown in Fig. 14. The state of the IACNN is calculated by $\Delta x(t)$, total change of internal state $x(t)$ temporally, and when $\Delta x(t)$ is less than a threshold value θ, the chaotic retrieval of the IACNN is stopped by changing values of parameter δ from 1.7 to 0.2. As a result, network converges to a stored pattern near the present network state.

4.4 Improved mutual associative type CNN

We make use of auto-associative type CNN as an improved mutual associative type CNN (**IMACNN**), namely, auto-associative matrix consisting of stored patterns is constructed with environmental inputs $\mathbf{s}$ and their corresponding actions $\mathbf{a}$. When $\mathbf{s}$ is set as a part of initial state of **IMACNN**, **IMACNN** retrieves action $\mathbf{a}$ with state $\mathbf{s}$. The memory matrix $\mathbf{W}$ is described as Eq. (16), here, λ is a forgetting coefficient, and η is a learning coefficient.

$$\mathbf{W}^{new} = \lambda \cdot \mathbf{W}^{old} + \eta \cdot (\mathbf{X}_{[0]}\mathbf{X}_{[0]}^{\mathrm{T}} - \mathbf{X}_{[1]}\mathbf{X}_{[1]}^{\mathrm{T}}) . \qquad (16)$$

Using this update rule, the information in STM becomes refined one, according to proceed the process of reinforcement learning. Here, $\mathbf{X}_{[0]} = [\mathbf{s}^{\mathrm{T}}\ \mathbf{z}^{\mathrm{T}}\ \mathbf{a}^{\mathrm{T}}]^{\mathrm{T}}$ is a stored pattern, $\mathbf{s}$: state vector of environment, $\mathbf{a}$: action vector, $\mathbf{z}$: random vector to weaken the correlation between $\mathbf{s}$ and $\mathbf{a}$.

STM as one unit consists of plural IMACNNs, and one IMACNN memorizes information for one environmental input pattern (refer to Fig. 15). STM has path information from start to goal of only one maze searching problem.

Fig. 11 Stored patterns for Cartier's and our improved network

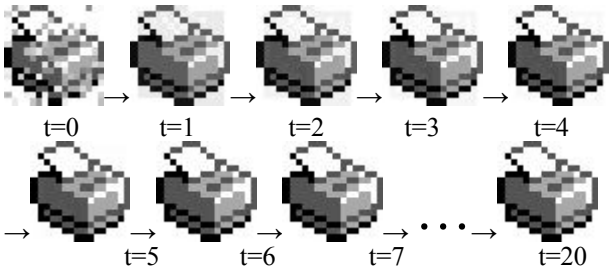


Fig. 12 Behavior of Chartier's model (δ =1.7)

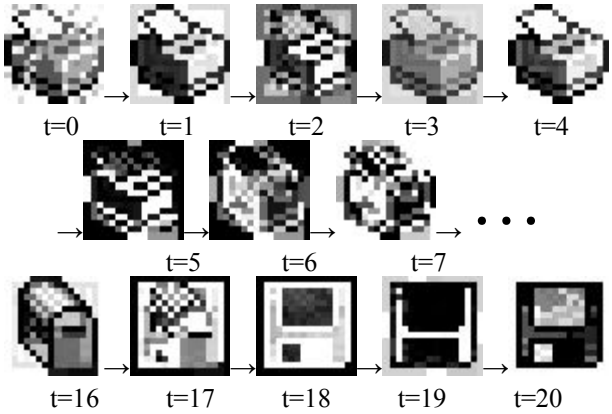


Fig. 13 Behavior of our improved Chartier's model ($\delta = 1.7$)

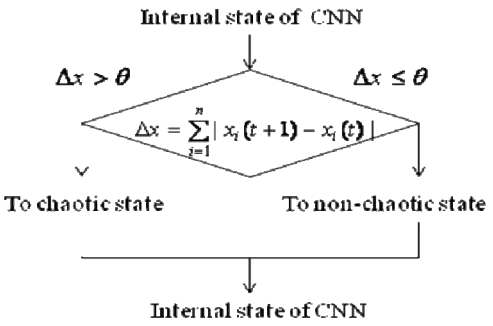
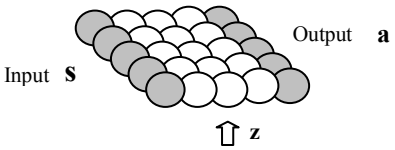


Fig. 14 Network control algorithm



$\mathbf{z}$: additional random memory units for weakening correlations between $\mathbf{s}$ and $\mathbf{a}$

Fig. 15 Memory configuration of IMACNN

4.5 Adaptive hierarchical memory structure

Fig. 16 shows a part of configuration of an adaptive hierarchical memory structure. When an environmental state is input to agent, at first it is sent to the LTM for confirming if it is the stored information or not. If it is the stored information, the obtained action corresponding to it is executed, otherwise, it is used to train the actor-critic system. The pair of enough refined and trained environmental state $\mathbf{s}$ and action $\mathbf{a}$ in the STM is sent to the LTM to be stored.

4.6 Memory set and its use algorithm

4.6.1 Memory set algorithm

In the case of no experience and no stored patterns corresponding to the present environment, the agent has to learn them and set to Memory sector.

Memory set algorithm is as follows,

Step 1: Receive a pair of the state of the environment and action, i.e., a stored pattern $\mathbf{x}_{[0]} = \left[\mathbf{s}^{\mathrm{T}}\ \mathbf{z}^{\mathrm{T}}\ \mathbf{a}^{\mathrm{T}}\right]^{\mathrm{T}}$ from learning sector.

Step 2: Using the observation of the state of the environment, memory sector selects the IMACNN corresponding to the environment.

Step 3: Calculate the next state by Eq. (13) using $\mathbf{x}_{[0]}$ and renew the memory matrix W by Eq. (16).

This memory set algorithm should be done for all the process of the reinforcement learning.

4.6.2 Memory use algorithm

In the case of existing of experience and stored patterns corresponding to the present environment, the agent acts by making use of the stored patterns.

Memory use algorithm is as follows,

Step 1: Observe the state of the environment.

Step 2: Using the observation of the state of the environment, memory sector selects the IMACNN corresponding to the environment.

Step 3: Given and fixed the present state, i.e., a part of the $\mathbf{x}_{[0]}$, to the IMACNN, execute the operation of the IMCCNN, i.e., Eq. (13) till the network state converges to a certain stored pattern, according to the network control (refer to **4.3**).

Step 4: Do the action gotten by the information at Step 3.

Repeat Step 1 ~ 4 till the agent arrives to the goal.

Table 1 Parameters used in the simulations

ACTOR-CRITIC			
σ	0.1	ξ	0.7
η	0.1	γ	0.98
T	1.0	-	-
Mutual Associative Chaotic Neural Network			
N	20	λ	0.9
η	0.1	δ	0.005
Network control parameters of MACNN			
	Chaos		Non-chaos
δ	1.7	δ	0.005

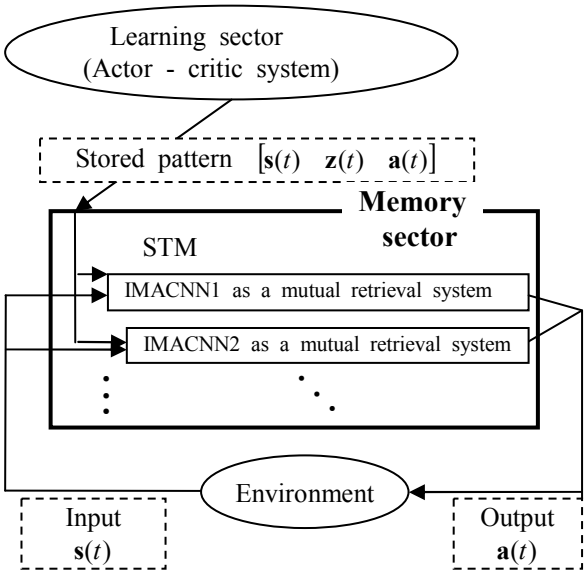


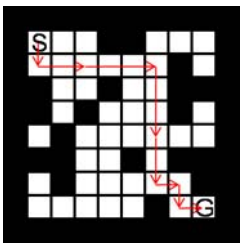
Fig. 16 Adaptive hierarchical memory structure

5. COMPUTER SIMULATION

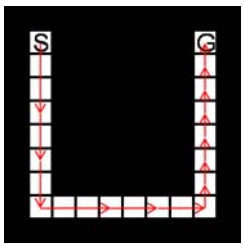
We treat the maze problems shown in Fig. 17. The multi-valued state and scope of the sensor of the agent are shown in Fig. 18. As shown in Fig. 18, agent can perceive the wall at 8 directions around itself. Therefore in actor-critic, state vector $\mathbf{s}$ of environment consists of 8 inputs (= n). We define the value of each state such that when the number of cells from agent to the wall are 1, 2, 3, or more than 4, the agent perceives them as 1, 1/3, -1/3, -1, respectively. For example, in Fig. 18-(left), when an agent is on A, the agent perceves around as Fig. 18-(right). On the other hand, agent can move appropriate cells between 1 cell and 3 cells by one action to the direction of forward, back, left, right, i.e., kinds of actions is 12 (3 times 4 = K in Fig. 4). When an agent gets the goal, the agent is given a reward, 1.0. For the case of collision with wall, reward is -1.0, and for each action except collision is - 0.1. Other parameters used in the simulations are shown in Table 1.

Results of the simulation using the maze 1 so-called medium sized maze are shown in Fig. 17- (a) and Fig.19. The learned optimal path is shown in Fig.17-(a) using red line with arrows. The length of arrow means a moving distance by one action. It says the action is used more efficiently. Results of the simulation using the maze 2 so-called the maze with aliasing are shown in Fig. 17- (b) and Fig. 20. In this case, it is found that the agent solved the aliasing problem by a choice of the action making use of the characteristic of the associative chaotic neural networks. Results of the simulation using the maze 3 so-called the large sized maze are shown in Fig. 17- (c) and Fig.21. Results of the simulation using the maze 4 so-called the semi-Markov typed maze is shown in Fig. 17- (d) and Fig. 22. From Fig. 17-(d) it finds that the choice of the action is optimal and agent reached the goal by shortest actions.

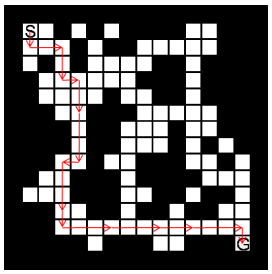
These results show that our proposed RL and the associative chaotic neural network with multi-valued patterns work well.



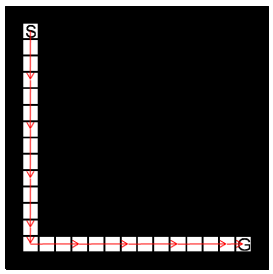
(a) Learned path at maze 1



(b) Learned path at maze 2

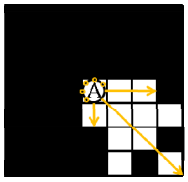


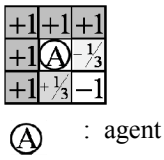
(c) Learned path at maze 3



(d) Learned path at maze 4

Fig. 17 Experimental mazes and results of learning





Ⓐ : agent

Fig. 18 The multi-valued state (right) and scope of the sensor (left) at the start position for maze1 (Fig. 17(a))

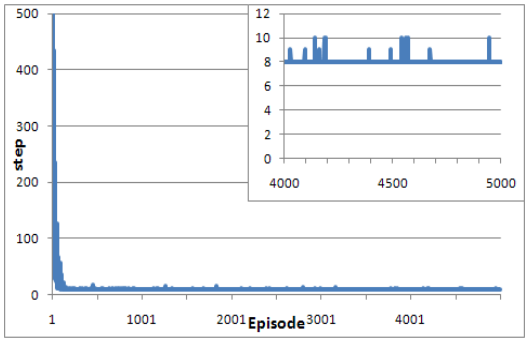


Fig. 19 Result of simulation 1 with use of maze 1

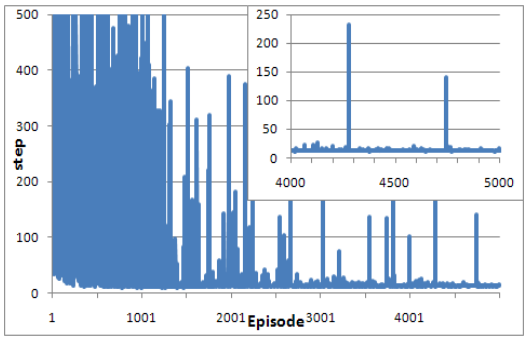


Fig. 20 Result of simulation 2 with use of maze 2

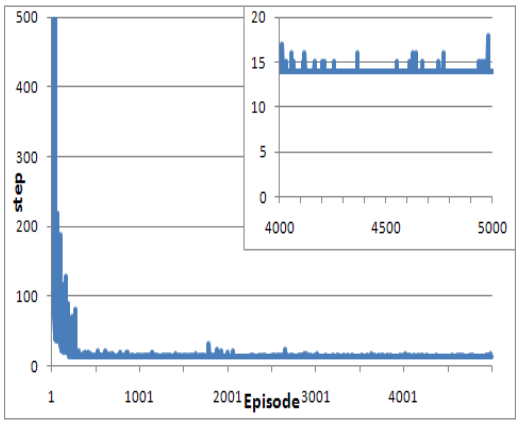


Fig. 21 Result of simulation 3 with use of maze 3

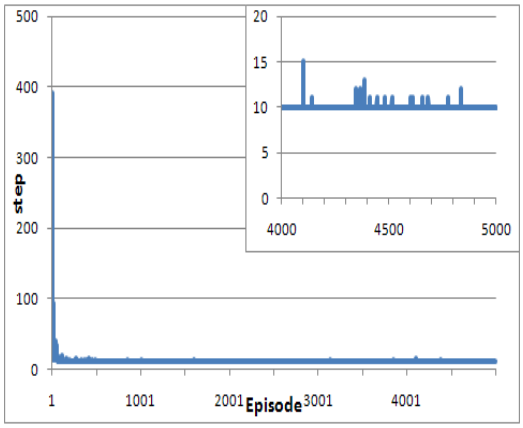


Fig. 22 Result of simulation 4 with use of maze 4

6. CONCLUSIONS

We proposed a reinforcement learning system embedded agent with neural network-based multi-valued pattern memory structure, so-called intelligent agent, and showed its effectiveness, especially for multi-valued memory and aliasing problem through the goal searching problem in plural mazes.

RERFERENCES

[1] R.S. Sutton, A.G. Barto: “Reinforcement Learning”, *The MIT Press,* 1998

[2] M. Adachi, K. Aihara: “Associative Dynamics in a Chaotic Neural Network”, *Neural Networks* ,Vol. 10, No. 1, pp.83-98, 1997

[3] M. Obayashi,*„et al.*:”A Reinforcement Learning System with Chaotic Neural Networks-Based Adaptive Hierarchical Memory Structure for Autonomous Robots”, *Proceedings of ICCAS2008,* pp. 69-74, 2008

[4] S. Chartier,*et al.*: “NDRAM : Nonlinear Dynamic Recurrent Associative memory for Learning Bipolar and Non-bipolar Correlated Patterns”, *IEEE Trans. on Neural Networks,* Vol.16, No.6, pp.1393-1400, 2005

[5] S. Chartier,*et al.*: “A nonlinear dynamic artificial neural network model of memory”, *New Ideas in Psychology* 26, pp.252-277, 2008