Emergence of Multi-Step Discrete State Transition through Reinforcement Learning with a Recurrent Neural Network

Mohamad Faizal Samsudin* †, Yoshito Sawatsubashi* and Katsunari Shibata*

Department of Electrical and Electronic Engineering, Oita University,
700, Dannoharu ,870-1192 Oita JAPAN*
School of Mechatronic, Universiti Malaysia Perlis,
02600 Arau, Perlis MALAYSIA †
ballack83@hotmail.co.jp†
shibata@oita-u.ac.jp*

Abstract. For developing a robot that learns long and complicated action sequences act in the real-world, autonomous learning of multi-step discrete state transition is significant. It is generally thought to be difficult to achieve both holding and transition of states through learning in a recurrent neural network. In this paper, only through the reinforcement learning using rewards and punishments in a simple learning system consisting of a recurrent neural network, it is shown that a multi-step discrete state transition emerged through learning in a continuous state-action space. It is shown that of the two-switch task, two states transition represented by the two types of hidden nodes emerged through the learning. In addition, it is shown that the contribution of the dynamics in the RNN based on the discrete state transitions leads to repetition of the interesting behavior when no reward is given at the goal.

Keywords: Recurrent neural network, multi-step discrete state transition, reinforcement learning

1 Introduction

For human being, long and complicated action's success does depend on how well the action is sorted to some discrete state transitions and how well the states are held in memory. For instance, in order to drive a car, we usually "take the key", "open the door", "start the engine" and "drive the car". We can see here that in order to reach the goal, humans must hold the state of "take the key" at first, before moving to the state of "open the door". It is a lot easier to memorize the discretized states and form the transition between the states rather than memorize all of the continuous states to perform appropriate action. When we want a robot to behave like that, the user usually tries to describe and perform all the states by developing the program. However, it is not good enough to act flexibly in the real-world environment. Therefore, it would be a significant step forward in the development of a potential human-like robot, if such state

representation and action planning based unit could be learned autonomously through experiences.

It has been said that neural networks are good at the continuous function approximation, but not good at the discrete state representation, and the research has not well-progressed. However, it has been well researched by using the recurrent neural network (RNN), the necessary information is memorized and is utilized in the action have been acquired through learning with the reinforcement learning[1][2][3]. In addition, the authors group has profounded that the combination of reinforcement learning and neural network lead to the emergence of various functions purposively and harmoniously[4].

However, it is recognized to be quite difficult to realize the multi-step discrete state transition in a neural network. That is because, the neural network needs to hold the state basically while a prompt transition between states must be performed as needed. When the attractors with a strong entrainment are formed to hold the states, it is thought to be difficult to move from that attractor to another in order to achieve the state transitions. If the entrainment of the attractors is too week, the transitions may occur inessential. Even to learn the counter task that counting the input signals by supervised learning is reported to be difficult[5]. Therefore, it is thought to be difficult to achieve both holding and transition of states through learning based on the reward and punishment that are scalar signals.

It is considered that in the learning of memory-required task as mentioned above[1][2][3], a single-step state transition based on the memorization of necessary information emerges by using a recurrent neural network with a reinforcement learning. However, there are no research that shows the emergence of multi-step discrete state transition through reinforcement learning using a recurrent neural network as far as the authors know. The authors believe that overcoming this problem opens the way to develop the human-like robot. Thus, in this paper, only by using the reward and punishment through the reinforcement learning, it is investigated whether multi-step discrete state transition emerges or not. Furthermore, it is confirmed that the representation of two states transition has been acquired through learning. In addition, the behavior of the robot when no reward is given at the goal is observed and discussed.

2 Reinforcement Learning with a Recurrent Neural Network

Reinforcement learning is autonomous and purposive learning based on trial and errors, and a neural network (NN) is usually used as a non-linear function approximator to avoid the state explosion due to the curse of dimensionality. The combination of reinforcement learning and neural network seems promising in the autonomous learning field and it was observed by the work of the author's group that the combination leads to the emergence of various necessary functions such as recognition, prediction, memory and communication[4].

In this paper, Actor-Critic with TD-learning[6] is used as a reinforcement learning algorithm. On the other hand, the network architecture used in the simulation here is an Elman-type recurrent neural network (RNN)[7] whose hidden outputs are fed back as a part of the input at the next time step. The present observation vector $\boldsymbol{x}_t$ is the external input. The outputs of the RNN are divided into a critic (state value) C and two actor outputs (motor command) $\boldsymbol{A}$. For learning, TD-error is represented as

$$\hat{r}_t = r_{t+1} + \gamma C(\boldsymbol{x}_{t+1}) - C(\boldsymbol{x}_t) \tag{1}$$

where r_{t+1} indicates a given reward at time step $t+1$, γ is a discount factor, $\hat{r}_t$ is the TD error, and $C(\boldsymbol{x}_t)$ is the critic output when $\boldsymbol{x}_t$ is the input of the RNN. After the forward computation in the RNN for the new input signal $\boldsymbol{x}_{t+1}$, the training signal for the critic $C_{d,t}$ is generated autonomously based on Temporal Difference learning and $\boldsymbol{A}_{d,t}$ for the actor output vector as

$$C_{d,t} = C(\boldsymbol{x}_t) + \hat{r}_t = r_{t+1} + \gamma C(\boldsymbol{x}_{t+1}) \tag{2}$$

$$\boldsymbol{A}_{d,t} = \boldsymbol{A}(\boldsymbol{x}_t) + \boldsymbol{rnd}_t \cdot \hat{r}_t \tag{3}$$

where $\boldsymbol{A}(\boldsymbol{x}_t)$ is the actor output and $\boldsymbol{rnd}_t$ is the uniform random number vector that was added to $\boldsymbol{A}(\boldsymbol{x}_t)$ as an exploration factor. Then, the RNN with the input $\boldsymbol{x}_t$ is trained by Back Propagation Through Time (BPTT)[8]. In Eq.(1), 0.5 is added to the value of the critic output, and 0.5 is subtracted from the derived training signal in Eq.(2) in order to adjust the value range of RNN output to the actual critic value.

3 Learning of Multi-step Discrete State Transition

3.1 System Architecture and Robot Learning Task

Fig.1 shows the system architecture and robot learning task. There are a robot, two switches and a goal on two dimensional 6 x 6 continuous, flat square space of arbitrary distance units. They are located randomly at the beginning of every episode. The shape of the goal and the switches is a circle of radius 0.5. They do not overlap with each other. The robot has to step on both of the switches in succession from switch 1 to switch 2 before it approaches to the goal. The input and output representations used in this simulation were straightforward. As shown in Fig.1(a), the observation vector has 11 elements in total. 3 signals represent the distances, 6 signals represent the angles to the goal and both of the switches. 2 signals represent the flag information for each of the switches. Only when the robot steps on the switch, it can perceive the flag signal whose value is 1. When the robot is not on the switch, the flag signal is always 0. All of these 11 signals are inputted to the RNN.

The robot gets a reward of 0.9 when it reaches the circle of the goal after stepping on both switches in a fixed order as shown in Fig.1(b). However, if the robot goes to the goal without stepping both switches or in a wrong order, a

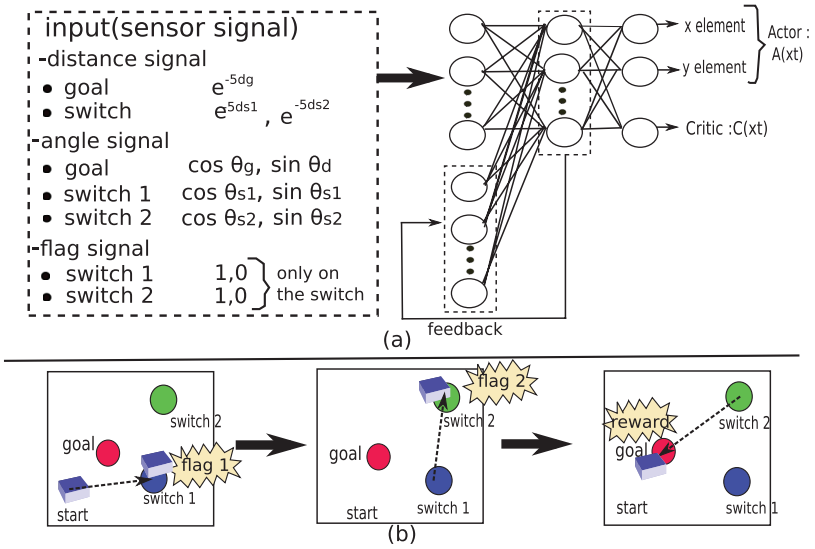

Fig. 1. System architecture and robot learning task. a) Observation vectors are the RNN inputs and the output nodes are divided into a critic (state value) and two actor outputs. Here, the two actor outputs represent step motion in x and y direction. b) The mission of the robot is to step on both switches in succession from switch 1 to switch 2 and then go to the goal.

punishment of -0.1 is imposed, and the episode is terminated. Moreover, if the robot collides to the wall, it is brought back to the place at the previous time step and a small punishment of -0.1 is also imposed.

The RNN has three layers consisting of 11 inputs, 40 hidden nodes and 3 outputs. The maximum time steps traced back for BPTT is 20. The discount factor γ is 0.96. The initial weight for each hidden-output connection is 0.0, and for each non-feedback input-hidden connection is chosen randomly from -1.0 to 1.0. In order to make the learning of the memory function easy, the initial weight for the self-feedback connections is set to 4.0, while the other feedback connections is 0.0. The learning rate is 0.05 for feedback connections, and 0.2 for the others.

3.2 Simulation Results

The learning results were similar to each other when more than 10 sets of random number sequences for initial connection weights and exploration are tested. One of them is shown in the following. Learning terminates after 250,000 episodes. As shown in Fig.2, the number of steps to the goal during learning is plotted at every episodes when the horizontal axis shows the number of episodes and the vertical axis shows the number of steps to the goal. In order to show how the multi-step discrete state transition emerges during the learning, 1000 different initial

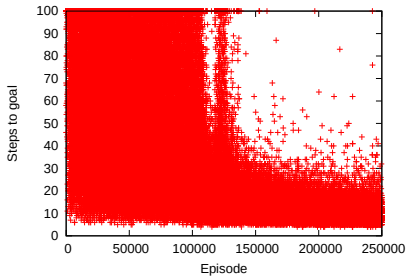

Fig. 2. The number of steps to the goal during learning is plotted at every episodes when the horizontal axis shows the number of episodes and the vertical axis shows the number of steps to the goal. The steps to the goal is decreased while the episode is increased during the learning.

location of starting point, switches and goal are tested and the temporal change in hidden nodes is observed. Fig.3a) shows two examples of robot trajectory after learning which is taken from various allocation of goal, switches and starting point. It can be seen from Fig.3a) that the robot stepped on the switch 1 at the time step 3 and 5 respectively. In Fig.2b), the critic output is increased while the actor x and y that represent the step motion of x and y direction are also changed drastically. It can see that the robot went to the right direction according to the positive value to the actor-x and actor-y. Then, the robot changed the direction because the actor-y changed to a negative value. The same can be said to the rest of the trajectory.

Furthermore, it is recognized that there are 2 types of nodes that seem to represent a transition between states although the number of hidden nodes that belong to each type is not so large, only 3 for type 1 and 1 for the type 2 among 40 hidden nodes. Fig.3c) and d) shows the temporal changed in hidden 2nd, 7th, 26th and 40th nodes in each episode. The type 1 nodes shown in Fig.3c) changed its output when entering the area of switch 1 and kept their output until the reward is given. On the other hand, the type 2 node shown in Fig.3d) seems to memorize the flag 2.

3.3 Test Performance and Observation of Temporal Change in Hidden Nodes.

Furthermore, in order to observe the robot's behavior, no reward was given and the episode was not terminated even though the robot succeeded to reach the goal. It is quite interesting to observe that the robot returned to search for both of the switches again even though no clue is given to the robot. As shown in Fig.4a), after stepping into the area of goal, the robot changed its direction to the switch 1 and 2 for several times. In addition, we could see some oscillation in the temporal changed for type 1 and 2 in hidden nodes as shown in Fig.4b) and c).

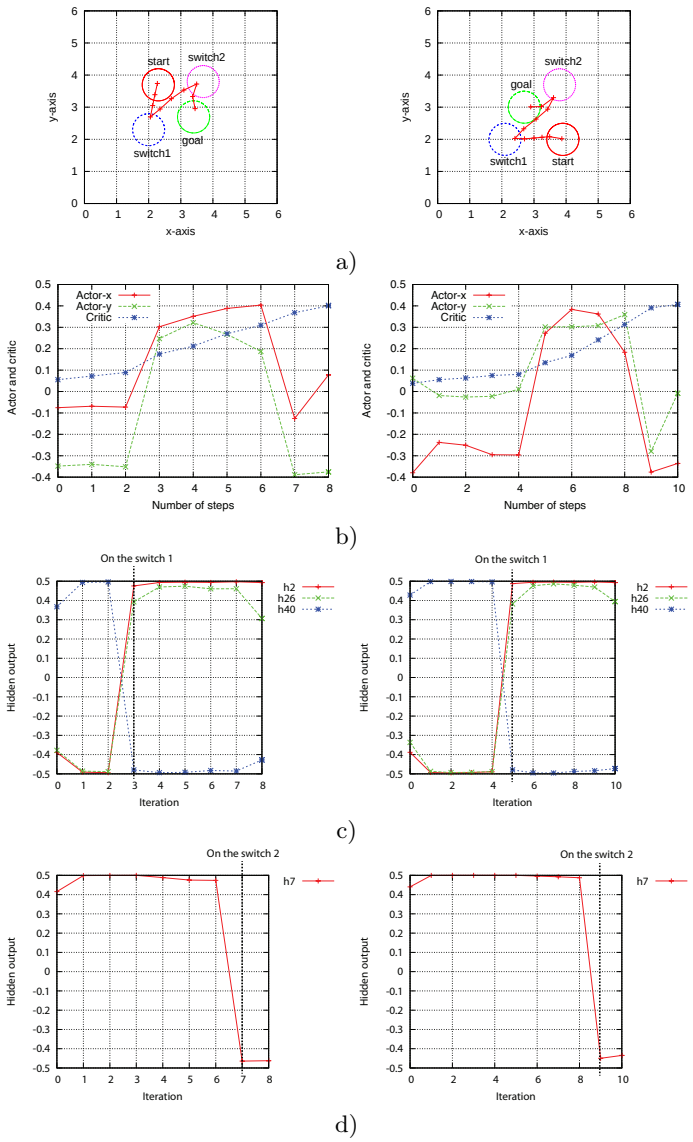

Fig. 3. a) Some examples of different robot trajectory. b) The change of critic and actor output for each trajectory in a). c) The change of type 1 hidden node's output that seems to respond to state 1 d) The change of type 2 that seems to respond to state 2

In the robot's experiences, when it succeeds to reach the goal with stepping the switches in a fixed order, the episode always terminated with a reward. However, when the reward did not perceive, it is thought that due to the gener-

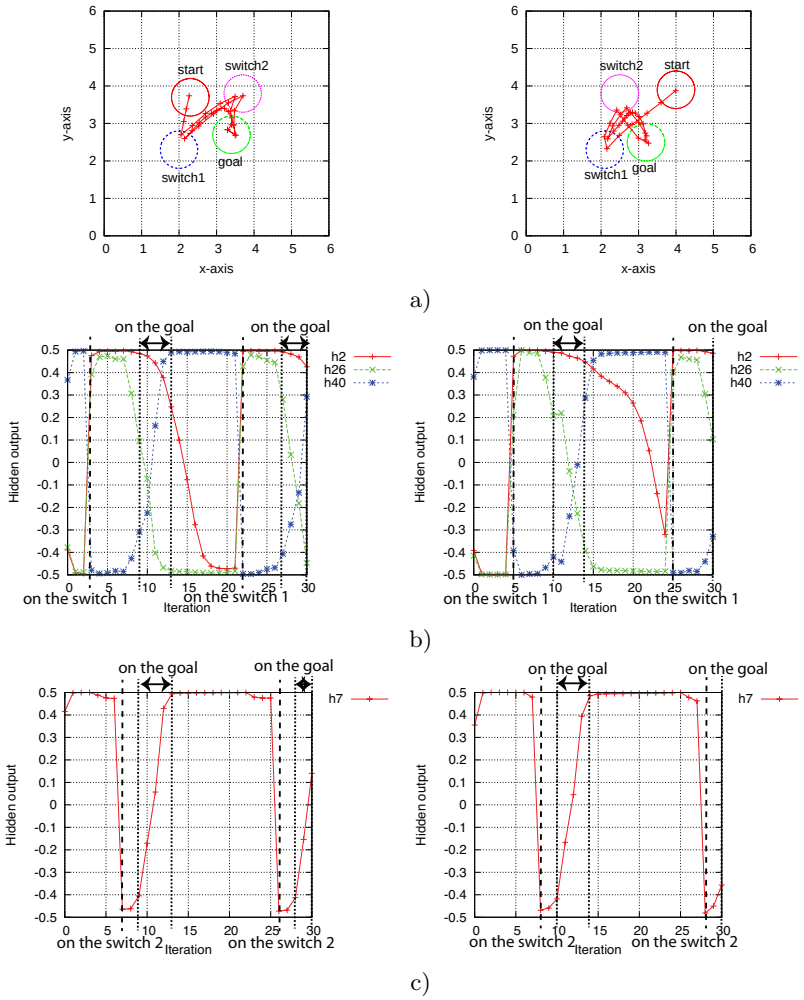

Fig. 4. a) The robot's behavior after no reward is given and the episode is not terminated. Robot kept going to the switch again even though no direction or clue is given. Some oscillation is observed to the nodes that seems to respond to the state 1 and 2 in b) and c)

alization from experience, the robot seems to think that it had forgotten to step on the switches even though it actually had stepped. Furthermore, it was shown that the robot took the trajectory with a proper order again and again. The discrete change in the type 1 and 2 hidden nodes also repeated synchronously to the behavior. The interesting behavior must be contributed by the dynamics based on the discrete state transition in the RNN. The type 1 and type 2 hidden nodes can be considered as control valuables and it is similar to the parametric bias proposed in [9].

4 Conclusion

In a memory-required task, a multi-step discrete transition emerged through a simple learning system consisting a recurrent neural network trained using a reinforcement learning in a continuous state-action space. From the simulation results, it was confirmed that two types of hidden nodes represented the state transition between before and after the pressing of one of the two switches. Furthermore, an interesting repetitive behavior of the robot was observed when no reward was given even the robot reached the goal. It was suggested that the dynamics in the recurrent neural network based on the discrete state transition is contributed to the behavior.

Acknowledgment

This work was supported by JSPS Grant-in-Aid for Scientific Research #23500245.

References

1. B.Bakker, V.Zhumatiy, G.Gruener, and J.Schmidhuber: "A Robot that Reinforcement-Learns to Identify and Memorize Important Previous Observations", Proc. of IROS 2003, 430-435 (2003)
2. H.Utsunomiya and K.Shibata: "Contextual Behavior and Internal Representations Acquired by Reinforcement Learning with a Recurrent Neural Network in a Continuous State and Action Space Task", Advances in Neuro-Information Processing, Lecture Notes in Computer Science, Proc. of ICONIP (Int'l Conf. on Neural Information Processing) 08, Vol. 5507, pp. 970–978, 5507-0970.pdf (CD-ROM), 2009
3. K.Shibata and H.Utsunomiya: "Discovery of Pattern Meaning from Delayed Rewards by Reinforcement Learning with a Recurrent Neural Network", Proc. of Int'l Joint Conf. on Neural Networks 2011, pp. 1445-1452, N-0311.pdf, 2011.
4. K.Shibata: "Emergence of Intelligence through Reinforcement Learning with a Neural Network", Advances in Reinforcement Learning, Abdelhamid Mellouk (Ed.), InTech, pp.99-120, 2011
5. Y.Taguchi and K.Shibata: "The Effect of the Initial Weight Values of the Learning Problem that Needs the Internal State Transition by a Recurrent Neural Network", (in japanese) Proc. of Kyushu Branch Annual Conf. of SICE, pp. 87-90, 2011.12
6. Barto, A.G., Sutton, R.S., Anderson, W.: "Neuronlike Adaptive Elements Can Solve Difficult Learning Control Problems", IEEE Trans. on Systems, Man, and Cybernetics 13(5), pp.834-846
7. Elman, J.L.: "Finding Structure in Time", Cognitive Science, 14, pp.179-211 (1990)
8. Rumelhart D.E., Hinton G.E. and Williams R.J.: "Learning Internal Representations by Errorpropagating", Parallel Distributed Processing, Vol. 1, MIT Press, pp.318-362 (1986)
9. J. Tani, M. Ito, Y. Sugita: "Self-organization of Distributedly Represented Multiple Behavior Schemata in a Mirror System: Reviews of Robot Experiments using RNNPB, Neural Networks (2004), 17, 1273-1289