Emergence of Discrete and Abstract State Representation through Reinforcement Learning in a Continuous Input Task

Yoshito Sawatsubashi[†] and Mohamad Faizal bin Samusudin[† ††], and Katsunari Shibata[†]

[†]Department of Electrical and Electronic Engineering, Oita University, 700, Dannoharu, 870-1124 Oita JAPAN
[††]Department of Mechatronic, Universiti Malaysia Perlis, 02600 Arau, Perlis MALAYSIA
bashis8@yahoo.co.jp
ballack83@hotmail.co.jp
shibata@oita-u.ac.jp

Abstract. "Concept" is a kind of discrete and abstract state representation, and is considered useful for efficient action planning. However, it is supposed to emerge in our brain as a parallel processing and learning system through learning based on a variety of experiences, and so it is difficult to be developed by hand-coding. In this paper, as a previous step of the "concept formation", it is investigated whether the discrete and abstract state representation is formed or not through learning in a task with multi-step state transitions using Actor-Q learning method and a recurrent neural network. After learning, an agent repeated a sequence two times, in which it pushed a button to open a door and moved to the next room, and finally arrived at the third room to get a reward. In two hidden neurons, discrete and abstract state representation not depending on the door opening pattern was observed. The result of another learning with two recurrent neural networks that are for Q-values and for Actors suggested that the state representation emerged to generate appropriate Q-values.

1 Introduction

While we get a huge amount of sensor signals with eyes, ears and other sensors, we can evaluate a state accurately and act appropriately. However, we are not conscious of each individual sensor signal, but rather represent a state with an abstract representation such as "room" and "corridor", and make an action plan as "open the door, go out the room, and walk along the corridor". Such discrete and abstract state representation must make our learning efficient and can be an origin of intelligence.

First of all, why can we recognize a "room" even though there are many kinds of rooms? For example, if the wall color is different, the sensor signal will

be different completely. What we usually see with our eyes is only part of a room, and the sensor signals change largely by eye, head or body movement. For these reasons, it is highly unlikely that such abstract representation is formed only from the huge amount of time series sensor signals. We probably recognize the "room" through the accumulation of action learning in a room where we "work or rest", "open a door to go out the room" and "open a window on a hot day". The authors call such discrete and abstract representation "concept". If "concept" as a high-order function is given to a robot, it is expected to behave appropriately even in the real world with various situations.

However, it is hardly possible for us to design each "concept" manually, because it is difficult to define it in words. For example, the place where there are walls, doors and windows is not always a "room". We recognize a place as a "room" based on the parallel consideration from many aspects, and learning from experiences with our brain as a massively parallel processing and learning system seems to enable it. Therefore, the concept formation is difficult to be achieved by hand-coding.

Tani et al. have proposed a method by which abstract state representation from the time series data as inputs emerge through learning. In the method, the next sensor signals are predicted from the present sensor signals and motor commands, and modular recurrent neural networks (RNNs)[1] or one RNN consisting of neurons with different time constants[2] are introduced. However, generating appropriate actions or achieving a goal is not considered. In their paper, the input was simple sensor signals, but if a visual sensor is used in the real world that is full of trivial information, it must be difficult for the system to predict all the input at every moment. Furthermore, as mentioned above, it seems impossible to divide the time series of huge sensor data into states such as "room" and "corridor" without considering the necessity for action generation.

Therefore, a method is suggested in which a neural network (NN) and reinforcement learning (RL) are combined for the emergence of abstract representation. RL enables to learn appropriate actions for a purpose such as getting a reward and avoiding a penalty. Therefore if RL is combined with a NN with a parallel structure, the NN is optimized based on RL, and the necessary information is extracted in the NN without any explicit directions. It is expected that the discrete and abstract state representation emerges in the NN as extracted necessary information. The authors group has been aiming it already, but in the previous works, binary signals that respond at a state transition were given as inputs, and/or a task with one-step state transition was employed[5][6].

In this paper, as a previous step of the "concept formation", it is aimed to form a discrete and abstract state representation in a task with multi-step state transition. Then, a simulation task is employed in which an agent moves in an environment with several rooms and doors, and the emergence of discrete and abstract state representation in which the state changes by the door opening is investigated in the hidden layers in a RNN after the mapping from the continuous sensor inputs to the action is learned in it by RL.

2 The learning system

Here, as shown in Fig. 1, a 5-layer Elman-type RNN is used to learn a task with multi-step state transition. The RNN is trained with back propagation through time (BPTT)[3] using the training signals generated by RL. As for RL, Actor-Q[4] is used in which Q-learning for discrete action selection and Actor-Critic for the continuous motion are combined. Therefore, in the RNN, there are two types of outputs: for the Q-values and for the Actors. At first, a discreate actions is selected based on the Q-values, and then if the action needs a continuous motion, the motion is generated based on the Actors. The Q-value for the previous action is updated using the Q-value for the present action and reward. The training signal for the Q-value $Q_{d\ a_t,t}$ is generated as

$$Q_{d\ a_t,t} = Q_{a_t}(\mathbf{s}_t) + TDerror_t = r_{t+1} + \gamma \max_a Q_a(\mathbf{s}_{t+1}) \tag{1}$$

$$TDerror_t = r_{t+1} + \gamma \max_a Q_a(\mathbf{s}_{t+1}) - Q_{a_t}(\mathbf{s}_t) \tag{2}$$

where γ indicates a discount factor, and r_t, $\mathbf{s}_t$, a_t indicates a reward, state vector, and action at time t respectively. The Actor output is updated using the TD-error derived from the Q-value on behalf of the critic in Actor-Critic. The training signal for the Actor $\mathbf{A}_{d,t}$ is generated as

$$\mathbf{A}_{d,t} = \mathbf{A}(\mathbf{s}_t) + TDerror_t \times \mathbf{rnd}_t \tag{3}$$

where $\mathbf{A}(\mathbf{s}_t)$, $\mathbf{rnd}_t$ indicates the Actor output vector and a random number vector that is added to the Actor output vector for exploration to generate the actual motion commands.

3 Task

As a previous step of the "concept-formation", a task with multi-step state transition is learned to examine the emergence of discrete and abstract representation from continuous inputs. As shown in Fig. 2, there are several rooms and doors in the task environment, and an agent is initially located at a random place in the central room. When it pushes a switch located at the center of each room, one of the surrounding doors that connects the present room with a next room is opened at random. An episode terminates when the agent passes two rooms and finally arrives at the third room. The agent gets a reward when it arrives at the third room. In the early stage of learning, the agent takes an action almost at random because all the outputs are 0.0 by setting all the initial connection weights to the output layer to 0.0. Through leaning, only with the reward at the goal, the agent is expected to generate appropriate actions and the abstract representation that is useful to achieve the goal.

As shown in Fig. 1, the agent detects the distance to the walls using 8 sensors. Each sensor signal represents the distance from the agent to the wall in one of the eight directions in the form of e^{-3d_i} where d_i indicates the distance to the wall

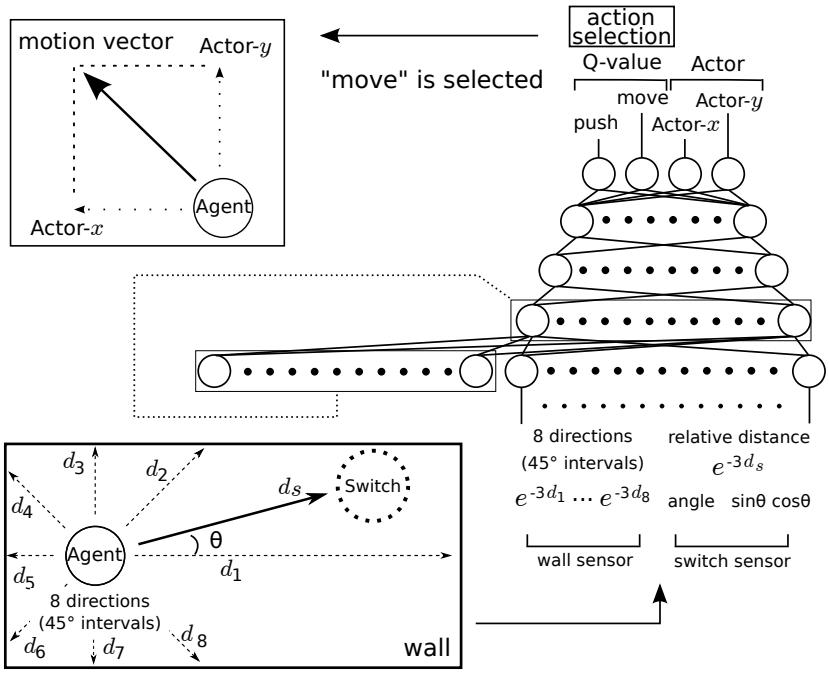


Fig. 1. Learning system consisted of a 5-layer Elman-type RNN, and its inputs and outputs.

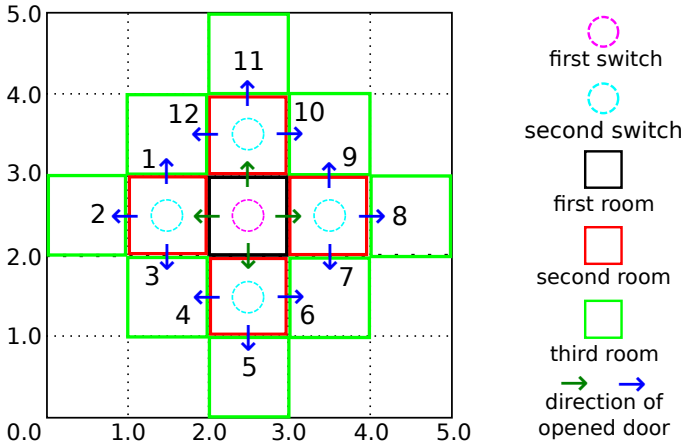


Fig. 2. Task environment with rooms and switches

in the i-th direction. The agent also detects the information about the closest switch. The distance from the agent to the switch is also represented as e^{-3d_s} and the angle of the switch direction was given as $sin\theta$ and $cos\theta$. Thus, the number of inputs in the RNN was 11 in total. This task needs memory because after opening a door in the second room, the agent cannot discriminate the first room and the third room without referring the past wall sensor signals.

As shown in Fig. 1, the number of outputs in the RNN is 4. Two of them are used as the Q-values of "move" and "push", and the other two are used as the Actors for the agent move in the x- and y-directions when the action "move" is selected. The size of the move vector is limited in the circle with the radius of 0.5 around the agent.

When the agent moves and arrives at the third room within 35 steps for 30,000 consecutive episodes, learning finishes. If the agent does not arrive at the third room as the goal within 200 steps, the episode is terminated. As an action selection method, "Boltzmann selection" is used, and the temperature is gradually decreased from 1.00 to 0.01 as the learning progresses. The value range of sigmoid function that is used as an output function in each hidden or output neuron is from -0.5 to 0.5. To transform between Q-value output [-0.5, 0.5] in the RNN and actual Q-value [0.0, 1.0], the value is shifted by -0.5 or 0.5. The other parameter settings are shown in Table 1.

Table 1. Parameter setting

initial position of the agent (x,y)	(2.0∼3.0, 2.0∼3.0)
radius of the switch	0.2
number of layers	5
number of neurons in each layer	11(input)-80-40-20-4(output)
constant input for bias	0.1
reward r at the third room	0.9
punishment for "push" at the outside of the switch	-0.1
discount factor γ	0.9
thr range of exploration vector **rnd** (rnd_x,rnd_y)	-0.5∼0.5 → -0.2∼0.2
traced back time in BPTT	30
initial connection weight	
input - hidden1, hidden1 - hidden2	-0.1∼0.1, -0.2∼0.2
hidden2 - hidden3, hidden3 - output	-0.5∼0.5, 0.0
self-feedback, other-feedback	4.0, 0.0
learning rate	
for feedback connections	0.0125
for other connections	0.5

4 Learning Result

The learning finished after 760, 675 episodes. After learning, the agent can reach the third room from any grid point with the interval of 0.01 in the central room for any pattern of the room appearance. However, depending on the random sequence used in learning, a few failures occured.

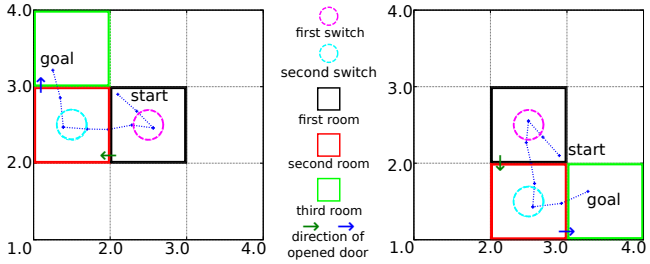
Fig. 3∼Fig. 6 show two sample agent behaviors after learning. In (a), the initial position is (2.1, 2.9), and the direction of the opened door is left at first and then upward in order, and in (b), the initial position is (2.9, 2.1), and the direction of the opened door is downward at first and then right in order. Fig. 3∼Fig. 6 show the agent trajectory, the Q-value outputs, and the Actor outputs, a part of the outputs in the top hidden layer at each step respectively.

As shown in Fig. 3, the agent moved on switches and pushed them and arrived at the third room. In both cases, when the agent pushed the second switch, the agent cannot know which door is newly opened only from the present sensor signals. However, from the fact that the agent arrived at the third room without being at a loss, it is considered that the agent could act based on memory. In Fig. 4, as the number of steps increased, the larger of the two Q-values at each step increased monotonically, and is close to the ideal curve. When the agent was located on a switch at the 2nd or 7th step, the value for the "push" action increased even though no explicit signal is given to identify that the agent is on a switch. This means that the agent could recognize it from the continuous inputs. In Fig. 5, it is known that when a new room appears, the Actor changes so as that the movement of the agent is directed to the new room.

Fig. 6 shows the output of characteristic neurons in the top hidden layer. A hidden neuron 1 changed its output only when the agent pushed the first switch. The hidden neuron 2 changed its output only when the agent pushed the second switch. The interesting point is that the output change of the hidden neuron at the door opening is very large and not depending on the direction in which the new room appears.

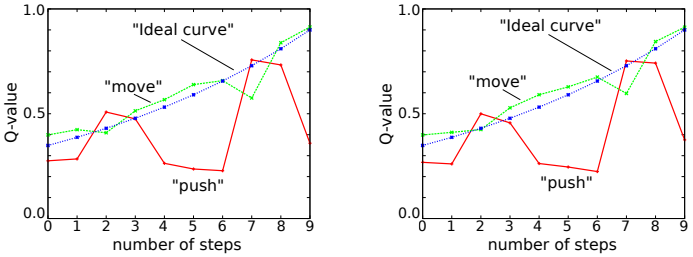
In a past study[6], it was reported that a discrete representation emerged through the learning in the task that an agent passed on a switch and arrived at a goal. In this case, a binary input indicating whether the agent is on the switch or not is given, so it is rather easy to form discrete state representation by holding the binary input in the RNN. However, in this task, although one of the distance inputs changes discretely when the agent opens the door, it changes only from 0.22 to 0.01 in the range from 0.0 to 1.0. Therefore, it is difficult to explain the sudden change in the hidden neuron only from the input signal, but the change should be explained from the necessity of generating Q-values or Actor outputs.

Then, to examine which output needs such discrete and abstract representation in the hidden neurons, Q-values and Actor outputs are learned separately using two RNNs in the learning of the same task. As the result, the discrete and abstract state representation was formed only in the RNN for Q-values. Fig. 7 and 8 show the Q-value outputs and a part of the output in the top hidden layer at each step respectively in the RNN for Q-values. Note that, the agent pushed



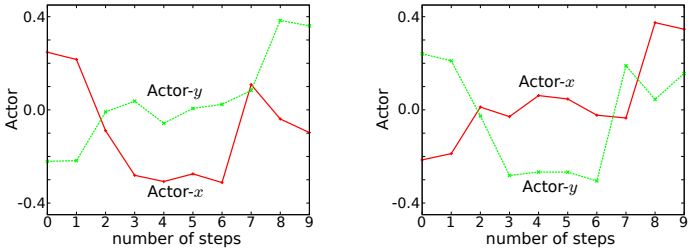
(a) (b)

Fig. 3. Two sample trajectories of the agent after learning.



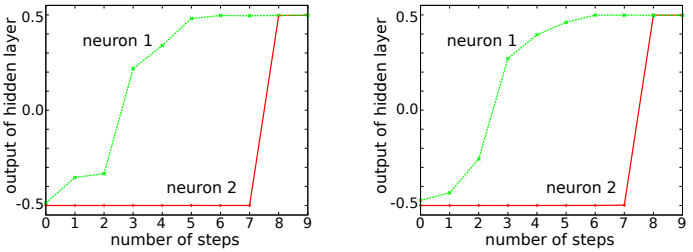
(a) (b)

Fig. 4. Change of the Q-values in one episode.



(a) (b)

Fig. 5. Change of the Actors in one episode.



(a) (b)

Fig. 6. Discrete change of the outputs in two neurons in the top hidden layer.

the second switch at one step earlier than the previous case. The hidden neurons that change suddenly at the door opening not depending on its direction have the largest connection weight with the Q-value output for the "move" action. It was reported that if the two training signal patterns are similar, the representation in the hidden layer that is close to the output layer is likely to be closer to each other through learning even though the input patterns are not close[7]. As shown in Fig. 4 (a) and (b), since the Q-values always change in the same way not depending on which door is opened, it can be explained that similar representation not depending on the door opening direction emerges. The reason of emergence of sudden change in the hidden neurons is suggested as follows. In the hidden neurons, the recognition of whether the agent is on a switch or not emerged from the necessity of raising the Q-value for the "push" action and its representation should be discrete to realize the sudden change in the Q-values at the step 2 or 7. The recognition result is held in another hidden neuron to represent the difference in Q-values between before and after the door opening. However, it has not been examined yet, and that is left as a future work.

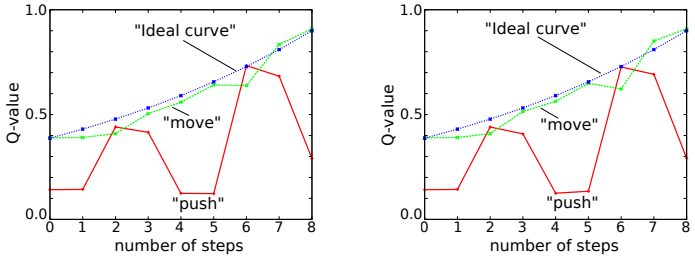


Fig. 7. Change of the Q-values in one episode.

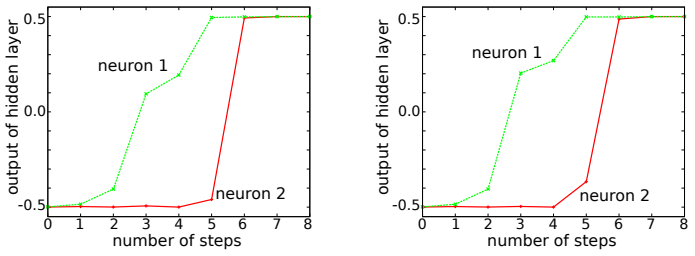


Fig. 8. Discrete change of the outputs in two neurons in the top hidden layer.

5 Discussion & Conclusion

In this paper, it was shown that discrete and abstract state representation emerged through Actor-Q reinforcement learning using a recurrent neural net-

work in doors-and-rooms environment. It is interesting that a different input signal changed depending on the opened door, but the representations in the top hidden neurons were almost the same. It is suggested from the learning using two recurrent neural networks that such representation emerge to generate Q-values that is also not depending on the door opening direction.

Although the discrete and abstract representation emerged, emergence of "concept" still seems to be out of reach. In order to form the "concept", it is required to extract necessary information and to recognize its state flexibly from more kinds of sensor signals that includes trivial information by considering many things simultaneously taking advantage of parallel processing. The large gap between the sensor signals and the state representation requires more intelligence, and that permit us to call a "concept", the authors hope. From this viewpoint, a visual sensor with many visual cells will be used in a future work.

Anyway, the authors hope that the work in this paper shows the usefulness of the combination of reinforcement learning and a recurrent neural network towards the emergence of "concept", and becomes a foothold for it.

Acknowledgment

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

References

1. Tani, J. & Nolfi, S. (1999) Learning to Perceive the World as Articulated: An Approach for Hierarchical Learning in Sensory-Motor Systems, Neural Networks, **12**, pp. 1131-1141
2. Y. Yamashita and J. Tani (2008) Emergence of functional hierarchy in a multiple timescale neural network model: a humanoid robot experiment, PLoS Computational Biology, **4**(11), e100220
3. Rumelhart, D.E, Hinton, G.E., and Williams, R.J.: Learning Internal Representations by Error Propagation, Parallel Distributed Processing, The MIT Press, pp. 318-362 (1986)
4. Shibata. K, Tetsuo Nishino & Yoichi Okabe (2002) Active Perception and Recognition Learning System Based on Actor-Q Architecture Systems and Computers in Japan, **33**(14), pp. 12-22
5. M, F, Samsudin. & Shibata, K. (2012), Emergence of Multi-Step Discrete State Transition through Reinforcement Learning with a Recurrent Neural Network, Proc. of ICONIP 2012 (to appear)
6. Utsunomiya, H. & Shibata, K. (2009). Contextual Behaviors 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, Vol.5507, pp. 970-978, 5507-0970. pdf (CD-ROM)
7. Shibata, K. & Ito, K. (2007). Adaptive Space Reconstruction on Hidden Layer and Knowledge Transfer based on Hidden-level Generalization in Layered Neural Networks, Trans. SICE, Vol. 43, No.1, pp. 54-63 (in Japanese).