Learning of Action Generation from Raw Camera Images in a Real-World-Like Environment by Simple Coupling of Reinforcement Learning and a Neural Network

Katsunari Shibata and Tomohiko Kawano

Oita University, 700 Dannoharu, Oita, Japan
shibata@cc.oita-u.ac.jp
http://shws.cc.oita-u.ac.jp/~shibata/home.html

Abstract. For the development of human-like intelligent robots, we have asserted the significance to introduce a general and autonomous learning system in which one neural network simply connects from sensors to actuators, and which is trained by reinforcement learning. However, it has not been believed yet that such a simple learning system actually works in the real world. In this paper, we show that without giving any prior knowledge about image processing or task, a robot could learn to approach and kiss another robot appropriately from the inputs of 6240 color visual signals in a real-world-like environment where light conditions, backgrounds, and the orientations of and distances to the target robot varied. Hidden representations that seem useful to detect the target were found. We position this work as the first step towards taking applications of the simple learning system away from "toy problems".

1 Introduction

In order to develop human-like intelligent robots, researchers have been trying to introduce sophisticated human functions into them. They have modularized the whole process into some functional modules such as recognition, action planning, and control at first, then developed each module individually, and finally connected them sequentially. However, the brain is massively parallel and cohesively flexible, and much of the brain functions seem to be performed subconsciously. Therefore, we think there is a limitation for humans to know exactly how the brain really works by the guess of the functions through the sequential consciousness. For the optimization and consistency of the entire system, we think that even though the understandability for humans is sacrificed, we should try not to interfere to the robot, but to leave everything to the robot's optimization through autonomous learning. From these discussions, we have suggested a general and autonomous learning system in which a neural network (NN) simply connects from sensors to actuators in a robot and is trained by reinforcement learning (RL) as shown in Fig. 1 [1][2].

M. Köppen et al. (Eds.): ICONIP 2008, Part I, LNCS 5506, pp. 754–761, 2009.

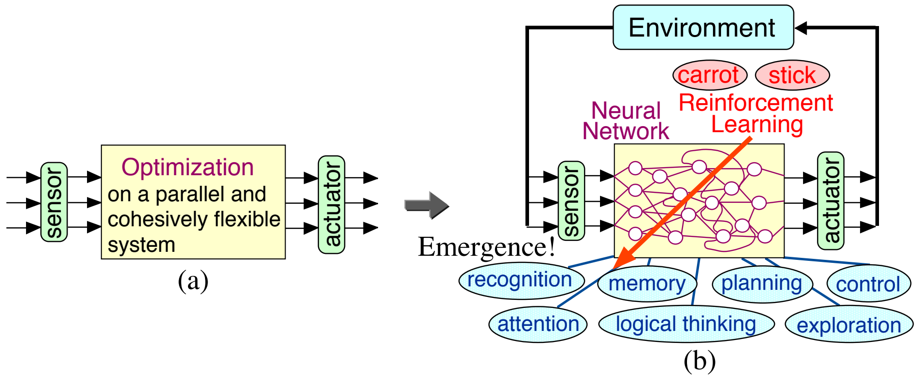


Fig. 1. (a) The objective in robots. (b) Parallel and cohesively flexible learning of the entire process from sensors to actuators by coupling of RL and NN.

In our previous work[3], acquisition of flexible image recognition by coupling of RL and a NN was shown, but the robot was static at the same place and could move its head to one of only nine discrete states. This means that the target recognition could be achieved easily by a "template matching" technique, and actually, in the hidden layers, many neurons seemed to work as a "template". In this experiment, since one robot was placed at a different initial location at each trial (episode) and walked, the number of possible views of the other robot could not be counted. The robot has to find the other robot in the camera images under various conditions as shown in Fig. 4, and has to approach and kiss it. One can understand that it is not so easy to develop appropriate programs for this task in the real-world-like environment. To learn a task with such a huge state space, the point is whether effective abstraction could be obtained autonomously through learning. We position this work as the first step towards taking applications of the simple learning system away from "toy problems".

2 Experiment

Here an experiment using two AIBO robots, a white one and a black one, is introduced. The two AIBOs are placed in a $1.5m \times 1.6m$ field. The mission of the black AIBO is to approach the white AIBO and kiss it as shown in Fig. 2. Before each episode, the black AIBO is located manually within the range of about ±45 degrees from the front line of the white AIBO and on the condition that the white AIBO is caught in the camera of the black AIBO. The black AIBO captures a color image with its camera at its nose after moving every step. The number of pixels is around 350,000, but when the AIBO sends the image to the computer, it is reduced to $52 \times 40 = 2080$ by pixel skipping. The white AIBO sits on a transparent plastic bottle, and is fixed at the same place during one episode. However, the location of the white AIBO was sometimes changed between episodes.

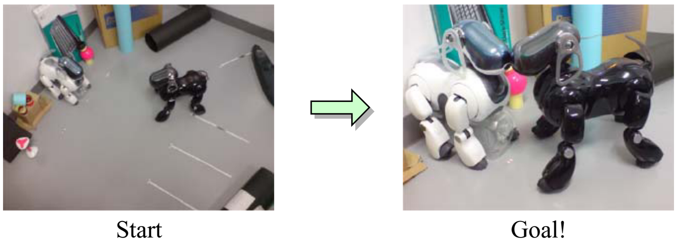


Fig. 2. The mission of the black AIBO is to walk and kiss the white AIBO

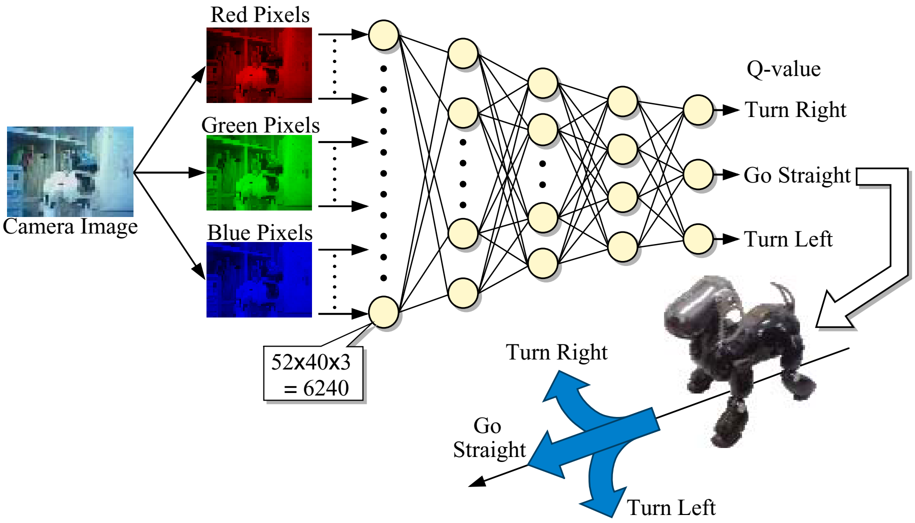
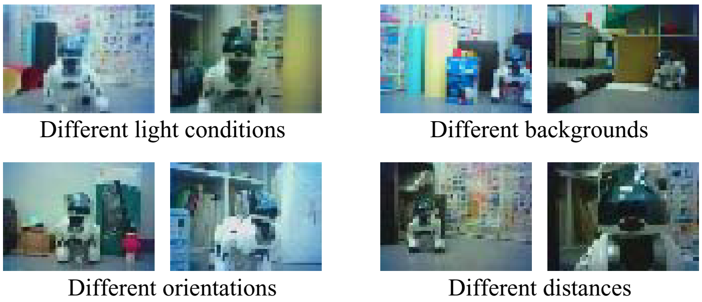


Fig. 3. The processing in the black AIBO. The visual signals are inputted to the NN directly and the black AIBO chooses its action according to the outputs from three possible actions, “turn right”, “turn left” and “go forward”.

The processing in the black AIBO is shown in Fig. 3. The 52×40 image is inputted into the 5-layer NN after inverting and normalizing each pixel value between 0.0 and 1.0. Since each pixel is represented by the gradation of RGB colors, the number of inputs to the NN is 6240 in total. The number of the output neurons is three in accordance with the number of actions that the black AIBO can choose. The three actions are “turn right”, “turn left” and “go forward”. The output is dealt with as the Q-value for the corresponding action, and ϵ-greedy ($\epsilon = 0.13$ here) is employed as the action selection[4]. The actions are based on the software provided by Ito and Kakiuchi[5]. When the robot chooses the action “turn”, the turning radius and angle are $22.5cm$ and $9.5°$ respectively. When choosing the action “go forward”, it advances by 6cm. However, the angles and distances actually vary at each time, since it is a real robot.



Different light conditions

Different backgrounds

Different orientations

Different distances

Fig. 4. Some sample images captured under different light conditions with different backgrounds, different orientations of and different distances to the white AIBO. They were used actually in learning.

An experimenter always monitored the situation and the camera image during the experiment. When it judged that the black AIBO kissed the white AIBO, a reward was given to the black AIBO. When it judged that the black AIBO lost sight of the white AIBO, a penalty was given. In both cases, the episode terminated. For acceleration of learning, some small rewards or penalties were imposed depending on the situation without termination of the episode, but they were not imposed after 2000 episodes of learning.

In order to show the variety of the captured images depending on the light condition, background, and orientation of and distance to the white AIBO, Fig. 4 shows some samples. The light conditions change according to the hour when the experiment was done. The location of the white AIBO and also the backgrounds were changed by the experimenter. The black AIBO not only approach from the front of the white AIBO, but also was located diagonally to the front of the white AIBO as an initial state though it was not located at its side or back.

As for the NN, the number of neurons in each layer is 6240-600-150-40-3 from input layer to output layer. The training signal is generated using Q-learning[6] and the NN is trained by Error Back Propagation[7]. When one of the actions is chosen at time t, the robot walked, and forward computation of NN at time $t+1$ is performed with the new camera image S_{t+1} captured after the walk. After that, the previous input S_t are inputted into the NN again, and the training signal is provided only to the output corresponding to the executed action a_t at time t according to

$$T_{a_t,t} = r_{t+1} + \gamma \max_a O_a(S_{t+1}) \qquad (1)$$

where γ is a discount factor (here 0.9 is used), $O_a(S_t)$ is the a-th output of the NN when the captured image at time t are entered as sensor signal inputs. Here, the sigmoid function used as the output function of each neuron ranges from -0.5 to 0.5. To adjust the offset between Q-value and NN output, a linear transformation is done. Here, since Q-value 0.0 corresponds to output -0.4 and Q-value 0.8 corresponds to output 0.4, 0.4 is added or subtracted actually for the transformation. When the episode terminates, the second term of right-hand

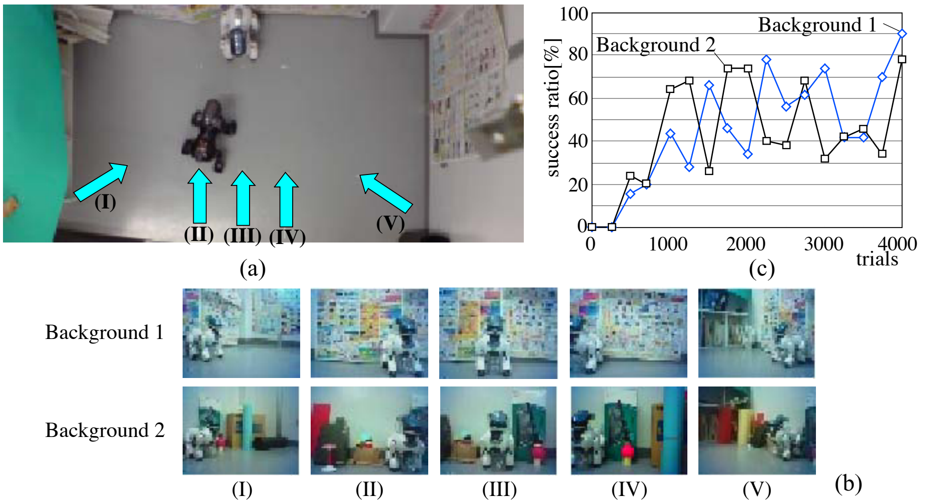


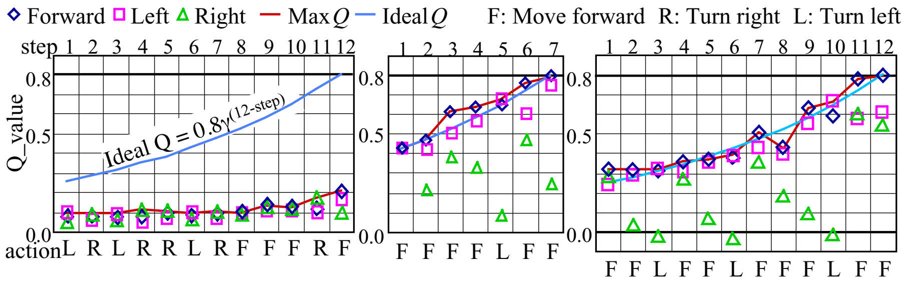
Fig. 5. (a) The environment for the test trials and (b) the camera images at the five initial locations from (I) to (V) for two kinds of backgrounds. (c) The learning curve shows the change of success ratio according to the number of trials.

side of Eq. (1) is set to be 0. The reward for the kiss is 0.8, while the training signal for the missing is equal to the Q-value output minus 0.02. The reward and penalty at non-terminal state is 0.04 and -0.04 respectively.

Since each episode takes a long time, simulation was also run unsupervised. All the visual signals and chosen actions were stored for every episode during the experiment, and in the simulation, the NN was trained by using the stored data up to that time. However, since actions could not be chosen, the same episodes appear. The simulation was run 13 times during the course of 4000 experimental episodes. In each simulation, 10000 episodes of learning were executed.

Fig. 5(c) shows the learning curve. All the connection weights in the NN are stored every 250 episodes of learning. Two backgrounds were prepared for the test, and five initial locations were selected as shown in Fig. 5(a). A sample image from each initial state is also shown in Fig. 5(b). Five trials were performed for each background and initial location set with the greedy action selection, and the success ratio for each background was observed as a learning curve. Using the connection weight set before learning, it could not kiss the white AIBO from any initial positions in either background. The ratio increased gradually at first, and did not increase very much after around 1000 trials. However, eventually, the success ratio reached between 78% to 90% even though the black AIBO was not exposed to the same background in the learning phase.

In order to examine how learning progressed, the time-series of Q-values are observed. Fig. 6 shows the change of Q-values for the two cases of using the connection weight set (a) after 1000 trials and (b)(c) after 4000 trials. If learning converged successfully, it is expected that the maximum Q-values at each step



(a) After 1000 trials from the position (III)

(b) After 4000 trials from the position (III)

(c) After 4000 trials from the position (I)

Fig. 6. Some samples of Q-value change in one trial. The weight sets after 1000 trials are used in the case of (a), and that after 4000 trials are used in the case of (b) and (c). The starting location is different between (b) and (c), and refer Fig. 5 for it.

would increase as the ideal Q-value that is the exponential curve decided from the discount factor γ and the final reward 0.8. It can be seen by comparing Fig. 6(a) and (b) that in the case of after 1000 trials, the episode length is larger than the case of after 4000 trials. The series of actions chosen can be seen at the bottom of each figure. The black AIBO chose the "turn left" action even in the case (b) of the front start after 4000 trials. The reason for this is that the black AIBO is likely to turn right slightly even when it chooses the action "go forward", and so compensated the shift. Furthermore, in the case (a) of after 1000 trials, the Q values are very different from the ideal, but in the cases (b) and (c) of after 4000 trials, the maximum Q-value is very similar to the ideal. From this result, it can be inferred that learning actually progressed even though the success ratio did not seem to change pronouncedly after 1000 trials.

Next, the role of the hidden neuron is guessed from the connection weights. Because the number of connections in each lowest hidden neuron is the same as the number of inputs, the weight pattern after a linear transformation can be observed as a color image whose size is 52 × 40. The weight patterns seem random due to the influence of the initial connection weight that were determined randomly. However, revealing patterns could be found when the change of each weight from the initial value was observed. The linear transformation from each weight value to the corresponding pixel value is as

$$pixel_{color,i,j} = \frac{w_{after,color,i,j} - w_{before,color,i,j}}{\max_{color,i,j} |w_{after,color,i,j} - w_{before,i,j}|} \times 127 + 128, \qquad (2)$$

where w_{after}, w_{before} indicate the weight after and before learning respectively. The color can be R, G, or B, and i, j indicates the row and column number of a pixel in the image respectively.

Most of the weight images are very vague, but in some of them, the image of the white AIBO is suggested as in Fig. 7(a-1,2). The upper images of Fig. 7(a-1,2) are similar camera images selected from actually captured ones. The representations look different from those observed in our previous experiment[3]

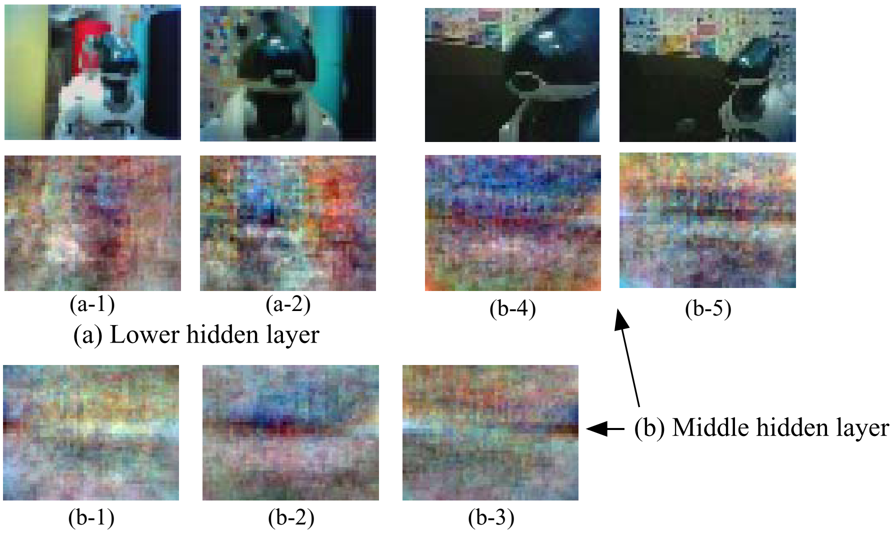


Fig. 7. Visualization of the hidden neurons' role generated from the connection weights change from the initial values. The images for the lowest hidden neurons are generated by Eq. (2), and those for the middle hidden neurons are generated as the weighted sum of the images for the lowest hidden neurons. The four images at the top are actual images that seem relevant to the weight images.

in which a clearer AIBO image could be seen. In the experiment, the black AIBO was fixed and only the head was rotated. Furthermore, the head angle could be one of only 9 states. This means that the number of views of the white AIBO is limited, which was enough for each hidden neuron to represent only one of the views of the target AIBO. However, in this experiment, the number of hidden neurons is not sufficient for each neuron to represent only one view because the number of views is far larger than the number of hidden neurons.

In order to roughly see the role of each neuron in the middle hidden layer, the weighted sum of the weight changes of the lowest hidden neurons by the connection weights between the lowest and middle hidden layers is observed. The word "roughly" is used because the non-linear factor in the lowest hidden layer neurons is neglected. Fig. 7 (b-1,...,5) shows the visualized weight image of some neurons in the middle hidden layer after normalization from 0 to 255. In most of the images, the dark or bright thin band-like area spread laterally in the middle section of the image is clearly visible in Fig. 7(b-1,...,5). It can be inferred that neurons such as (b-1,2,3) contribute to detect the lateral location of the target AIBO. In most of them, by the lateral extension of the band-like area, the color changes from dark to bright or from bright to dark at the same height. It is inferred that this emphasizes the contrast in the neuron's output according to the lateral location of the target AIBO. When comparing with the weight images of (b-4,5), the dark blue area is spread wider to the upper area of the image in (b-4), while the dark area is thin in (b-5). Referring from the actual camera images above these weight images, it can be inferred that such neurons contribute to detect the distance to the target AIBO. The representations also look different from those observed in our previous experiment in which fat AIBO images often appeared in the images for the neurons in the middle hidden layer.

However, the detection of the lateral or forward distance is only a part of the functions that the NN acquired. Neurons in this experiment must acquire a variety of functions such as compensation of light conditions, neglect of background and some other functions that we were not aware of from the connection weights easily. This is very similar to the situation where we cannot understand exactly how the brain functions when we see the excitation pattern of real neurons in the brain.

3 Conclusion

It was shown that although our proposed learning system that consists of one neural network trained by reinforcement learning is very simple, it worked on a real walking robot in a real-world-like environment. Many hidden neurons seem to obtain useful feature extraction abilities through learning without any suggestions from the others. The image processing acquired in this task seems different from the template-like image processing that was found in our previous work in which the robot did not walk and the head could rotate to one of only 9 states. This difference shows the flexible and purposive function emergence in a neural network by our proposed approach.

Acknowledgments. This research was supported by JSPS Grant in-Aid for Scientific Research #19300070. We would like to express our gratitude to Dr. Ito and Mr. Kakiuchi in Naruto University of Education for permitting us to employ their software.

References

1. Shibata, K., Okabe, Y.: Reinforcement Learning When the Visual Signals are Directly Given as Inputs. In: Proc. of ICNN 1997, vol. 3, pp. 1716–1720 (1997)
2. Shibata, K., Iida, M.: Acquisition of Box Pushing by Direct-Vision-Based Reinforcement Learning. In: Proc. of SICE Annual Conf. 2003, 0324.pdf, pp. 1378–1383 (2003)
3. Shibata, K., Kawano, T.: Acquisition of Flexible Image Recognition by Coupling of Reinforcement Learning and a Neural Network. The SICE Journal of Control, Measurement, and System Integration 2(2) (to appear, 2009)
4. Sutton, R.S., Barto, A.: Reinforcement Learning: An Introduction, A Bradford Book. MIT Press, Cambridge (1998)
5. Ito, Y., Kakiuchi, H., Oizumi, K., Kikuchi, A.: Development of Educational Software to Control Walking Motion for Four-legged Robot. The Japanese Society of Technology Education 49(1), 1–9 (2007)
6. Watkins, C.J.C.H.: Learning from Delayed Rewards, PhD thesis, Cambridge University, Cambridge, England (1989)
7. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Internal Representation by Error Propagation. In: Parallel Distributed Processing, vol. 1, pp. 318–364. MIT Press, Cambridge (1986)