Acquisition of Context-based Active Word Recognition by Q-Learning Using a Recurrent Neural Network

Ahmad Afif Mohd Faudzi[1,2] and Katsunari Shibata[3]

[1] Kyushu University, Fukuoka, Japan,
[2] Universiti Malaysia Pahang, Malaysia,
afif@cig.ees.kyushu-u.ac.jp
[3] Oita University, Oita, Japan,
shibata@oita-u.ac.jp

Abstract. In the real world, where there is a large amount of information, humans recognize an object efficiently by moving their sensors, and if it is supported by context information, a better result could be produced. In this paper, the emergence of sensor motion and a context-based recognition function are expected. The sensor-equipped recognition learning system has a very simple and general architecture that is consisted of one recurrent neural network and trained by reinforcement learning. The proposed learning system learns to move a visual sensor intentionally and to classify a word from the series of partial information simultaneously only based on the reward and punishment generated from the recognition result. After learning, it was verified that the context-based word recognition could be achieved. All words were correctly recognized at the appropriate timing by actively moving the sensors not depending on the initial sensor location.

Keywords: reinforcement learning, neural network, active perception, word recognition

1 Introduction

Among the human sensory organs, vision is probably the most informative perception. The eye movement and recognition capability in humans seem very flexible and intelligent. It has been shown that, in object recognition, compared to the case where only the information of the object is provided, a better recognition could be done if we were also supported by other information such as past knowledge and contextual information.

Imagine that we were presented with several new patterns or words repeatedly until we learned their shape and color in detail. At one time, we could recognize all the patterns correctly even if only some parts of these recognized objects that can be distinguished each other are visible to us. Even when perceptual aliasing, a condition where the same input stimuli belong to different words occurs, we can still recognize it by memorizing the different past input

stimuli or by moving our eyes actively[1]. This is because human has the ability to extract the necessary information, e.g., contextual information, and the ability to memorize it. It has also the ability to move sensors efficiently, which is called active perception. Our flexible brain achieves such abilities and learning must play an important role to acquire them.

Learning-based active perception/recognition methods have been proposed[2]-[6] including the works for object tracking[4][6]. In [3][5][6], considering probability distribution of each possible presented pattern or the state of the pattern from the image sequences explicitly, the appropriate viewpoint is learned to reduce the uncertainty in the distribution. In [4], conventional image processing methods are used and the system is complicatedly designed. In [2], by reinforcement learning, the system learns to choose to make a response that the target pattern presented or to move the sensor, but classification of the presented pattern is not necessary. Except for [2], the system and method is designed especially for active recognition, and recognition and sensor motion generation are separately performed. Therefore, it is concerned that the approach is not suitable for developing human-like intelligence by integrating with other functions.

Furthermore, to solve the partial observation problem, past information is utilized in several ways. In [6], useful features are extracted from all the captured images, and also in [2], memory-based reinforcement learning is employed. However, no mechanism has been proposed that learns to extract important information from the captured image at each time step, to hold it, and to utilize it for recognition and sensor motion according to the necessity.

The authors have thought that for developing flexible intelligent robots like humans, the entire process from the sensors to actuators should be learned harmoniously in total[7]. In the machine-learning field, many researchers are positioning reinforcement learning (RL) as learning specific for actions in the total process, and the NN as a non-linear function approximator. Based on the above standpoint, the authors' group has used a recurrent neural network that connects from sensors to actuators and learns autonomously based on reinforcement learning. It has been shown that necessary functions emerge according to necessity through learning. In this framework, it was shown that recognition and sensor motion are learned simultaneously using a neural network[8, 9]. It was verified through simulations and also using a real camera that the appropriate camera motion, recognition and recognition timing were successfully acquired[9]. However, the samples are limited and because the learning system was trained using a regular layered neural network, the recognition and sensor movement function were limited to the case where the presented pattern can be classified from the present captured image. On the other hand, it was shown that by using a recurrent neural network (RNN), contextual behavior, memory or prediction function emerged through reinforcement learning[10]-[12].

In the previous work, it is examined that using a RNN, both sensor motion and context-based word recognition functions emerge through reinforcement learning. At that time, the initial camera facing direction was fixed at the left edge of the presented word[13]. However, in this paper, the initial camera facing

direction at every episode is randomly set up. The input of the RNN is just the present image, and so the RNN has to learn what information should be extracted and memorized from the image for appropriate recognition and sensor motion to get a reward. The learning is very simple and general, and so it is easy to extend to integrate with other functions acquired on the same basis of reward and punishment.

This paper is organized as follows. In the next section, we will describe the basic idea of the learning system. In Section 3, we will explain the learning method and the task settings. The learning results are written in Section 4, and Section 5 states the conclusions.

2 Learning of Context-based Active Word Recognition

As shown in Fig. 1(A), the context-based active word recognition learning system has a movable camera as a visual sensor and a monitor. The monitor will display a word that is randomly chosen, and the system needs to identify which word is displayed. In every episode, only one word will be displayed. The camera can make a horizontal movement either to the left or to the right with a constant interval. In this paper, the initial camera facing direction for every episode is randomly set up, and so the camera sometimes should be moved to a different direction even though the camera catches the same potion of an image. In order to avoid from spending a lot of time on learning, instead of using a real camera, a simulation based on real camera movement was done using the images remade from the captured ones by the camera. Fig. 1(B) shows the 6 words that were used in the learning process and their partial images for all the camera directions.

As shown in Fig. 1(A), the sensor's field is too small to identify the presented word. In order to recognize the words correctly, the system has to memorize the information of the partial images that had been captured in the same episode. For example, in the case of the word 'cat' and the initial camera direction is 0, since it is the only word that starts with character 'c', the system can judge and recognize it from the first partial image. While in the case of word 'mad', the system needs more information about the second character that is held by the next partial image to distinguish it from the words 'men' and 'met' that have the same initial character. In the case of the other 4 words, the system is expected to recognize them when the partial images that hold the last character information are inputted. However, for the words 'met' and 'net', since the same images are inputted after the 4th camera direction, in order to distinguish them, the system needs to memorize whether the first character was 'm' or 'n'. Through this learning, the system is expected to recognize all the prepared words correctly at the appropriate timing by flexibly moving the camera.

3 The Learning Method and The Task Settings

Here, a 4-layer Elman-type RNN is used as shown in Fig. 1. The RNN is trained with back propagation through time (BPTT)[14] using the training signals gen-

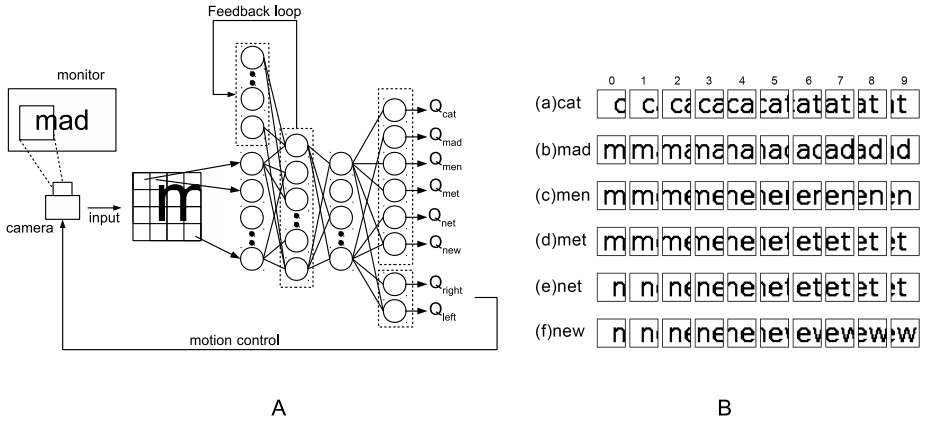


Fig. 1: **A:** Context-based word recognition learning system architecture. There is a monitor to display all the prepared words, and a camera as a visual sensor for the system. **B:** Six prepared words and all their partial images. These images were prepared base on the camera movement from "direction 0" to "direction 9" with a constant interval.

erated by reinforcement learning (RL). As for RL, Q-learning, which is a popular learning algorithm for discrete action selection problem, is used[15]. In Q-learning, state-action pairs are evaluated, and the evaluation value that will determine the action is called Q-value. The Q-value for the previous action is updated using the Q-value from the present action and reward. The training signal for the Q-value $T_{a_t,t}$ is generated as

$$T_{a_t,t} = r_{t+1} + \gamma max_a Q(s_{t+1}, a) \tag{1}$$

where γ indicates a discount factor ($0 \leqslant \gamma < 1$), and r_t, s_t, a_t indicates the reward, state, and action at time t respectively. In this learning system, the state s is the direct image signals from camera, and the action a is a discrete decision making. The action a is selected based on ε-greedy. In greedy selection, there is no probabilistic factor and the action with the maximum Q-value is always selected. Otherwise, the action a is selected randomly while the probability for "recognition result output" for each of the 6 words is 0.1/3 and for "camera movement" is 0.80. The current episode is terminated when the system makes a recognition or when the system moved the camera more than 25 times, i.e., $t = 25$. For the new episode, another word will be displayed and the camera is set up to a new facing direction.

As for the RNN, the inputs are the raw image signals that are linearly converted to a value between -0.5 and 0.5. There are eight output neurons. Each of them represents the Q-value corresponding to one of the eight actions, which are six "recognition result outputs" and two "camera movement" actions. The number of neurons in each layer is 576-100-15-8 from the input layer to the out-

put layer. The number of neurons in the input layer does not includ the hidden outputs that are fed back to the input layer at the next time step.

The output function of each neuron is a sigmoid function with the range from -0.5 to 0.5. However, all the outputs are used after adding 0.5. When the system chooses to output the result and it is correct, the training signal is added by 0.9. However, if it is not correct, the training signal will be the value of corresponding Q-value deducted by 0.2. 0.5 is subtracted from the training signal before using it as the actual training signals in BPTT. On the other hand, if the system chooses to move the camera, no reward is given.

Initial connection weights from the external input to the hidden layer are random numbers from -1.0 to 1.0. The initial connection weights from the hidden layer to the output layer are all 0.0, and as for the feedback connection of the hidden layer, initial weights are 4.0 for the self-feedback and 0.0 for the other feedback. This enables the error signal to propagate the past states effectively in BPTT without diverging.

4 Learning Result

As a result, after 150,000 episodes, both flexible sensor motion and context-based word recognition function can be achieved. Fig. 2 shows the Q-values for each action when every episode was finished or terminated for the presentation of words (a)'cat', (b)'mad', (c)'men', (d)'met', (e)'net' and (f)'new' respectively. The highest value for each episode indicates what action was taken if the action selection was greedy.

As shown in Fig. 2(a)'cat', at first, the red point (+), which is the Q-value corresponding to the word 'cat', increased soon and became higher than the other Q-values. At the same time, except for the black point, the other Q-values that correspond to the other words had decreased. By these big differences of Q-values, the system can make recognition correctly. On the other hand, the black point, which is the Q-value that corresponds to camera motion, was also high showing that if the system moved the camera, it can recognize the word correctly at the next step. This characteristic is also shown in Fig. 2(b) to (f) indicating that the system also successfully recognized the other words provided.

Fig. 2 also shows that different words required different time of learning. As shown in Fig. 1, at one time, the system can only capture a partial of the whole word. Since all the partial image of the word 'cat' does not identical to any of the other words, the system can distinguish it when only the first partial was captured. Here, Fig. 2(a) 'cat' clearly shows that the Q-value corresponding to the word 'cat' became high while the other Q-values became low at earlier stage of learning. While in Fig. 2(b), which corresponds to the word 'mad', shows that the green point (x) was also high, however there were also some corresponding points that were not high. This happened possibly due to the insufficient learning depending on the initial camera direction. The system need a longer learning time to recognize the second character, 'a', since the first character, 'm', is the same with words 'men' and 'met'. For the words(c)'men' and (f)'new', the partial

images from the 0th to 4th are identical to those for the word ‘met’ or‘new’, but those from the 5th to 9th are not identical to any other partial images.

Finally, in Fig. 2(d)‘met’ and (f)‘net’, it can be observed that the system required a longer learning time to recognize them compared to other 4 words. All the partial images of ‘met’ or ‘net’ have the identical image in another word. That means that the words cannot be classified correctly only from the present image, and the system has to classify them from both present and past images. It is thought that it took a long time to learn to memorize and use the past image even though it was not taught that the memorization of the information about the past image was necessary. The slower learning for the word ‘met’ than for the word ‘net’ may be because the number of words that start with the character ‘m’ is more than that with the character ‘n’.

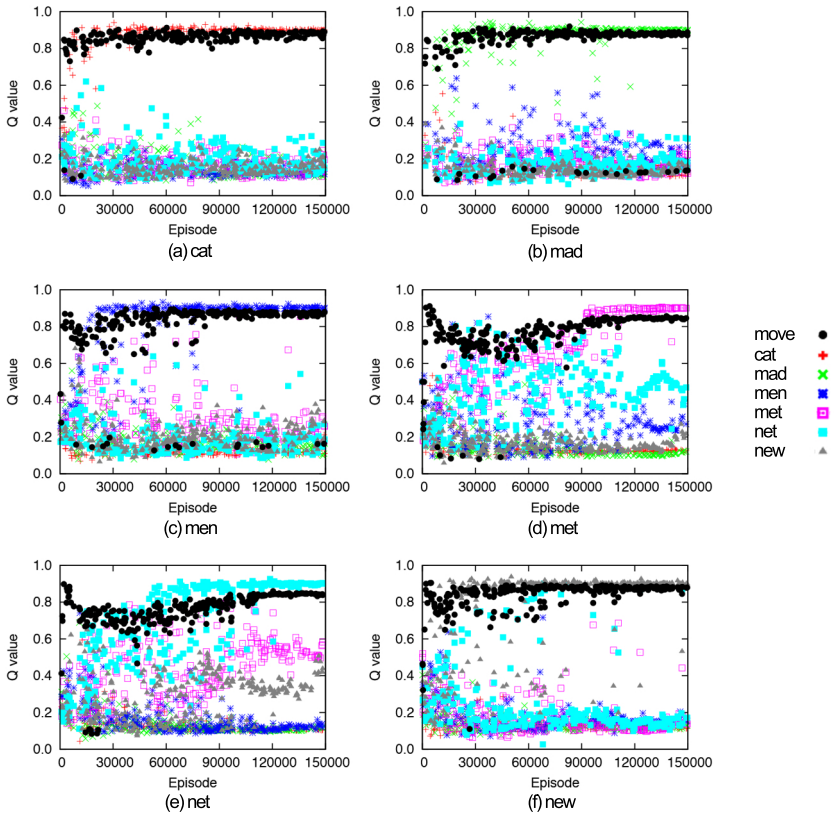
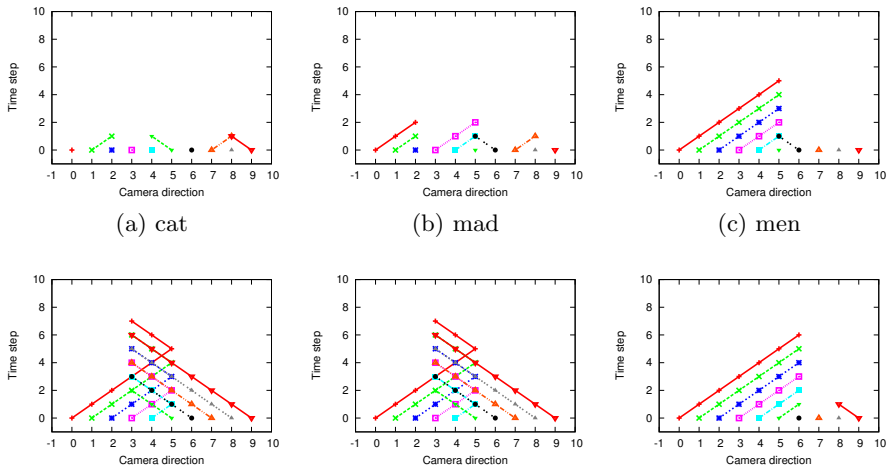


Fig. 2: The learning curve for each presented word. The plots show the Q-values for all actions at the end of every episode during the learning simulation for each word (e.g., (a) The learning curve when the word ‘cat’ was displayed on the monitor).

After the learning process had finished, a performance test was done. For each word, the system was tested by initializing the camera at each of the 10 directions. In this test, the system chose to move the camera's direction either to the right or to the left greedily according to the Q-values. Fig. 3 shows the camera movement for each initial direction. The x-axis indicates the camera direction that corresponds to the partial images in Fig. 1(b). y-axis indicates the number of step, and so x at y=0 indicates the initial direction. Here, when the system chose one of the "move the camera" actions, y will increase and x will show the current direction. A line shows the history of camera movement, and the final plot on the line shows the direction where the system chose to output the recognition result.

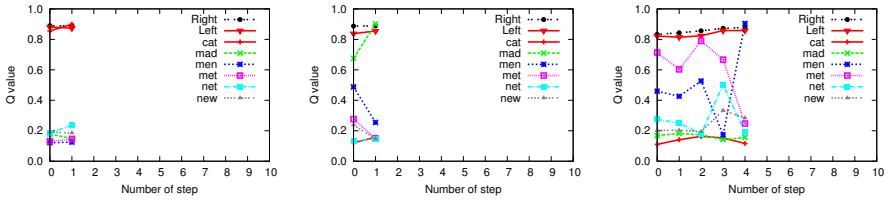
As shown in Fig. 3(a), since the word 'cat' is much different compared to others, from every initial direction, one or no movement is required for the system to recognize the word. As for the word 'mad', Fig. 3(b) shows that when the camera started from the leftmost side, the system outputs the recognition result at 'direction 2' where the information of the second character 'a' is inputted. This happened because the first character 'm' is same with the words 'men' and 'met'. We can also see that, when the camera started from the rightmost side, it only took one or no movement to make recognition. As for the word 'men' whose first two characters were same for the word 'met', Fig. 3(c) shows, for the initial direction 0 to 6, the recognition result is output at the direction 5. The correct recognitions were done at direction 5 even though there was a slight difference compared to the word 'met'. When the initial direction was at direction 7 to 9,



(a) cat (b) mad (c) men

(d) met (e) net (f) new

Fig. 3: The performance test result-1: Camera movement when initial direction is set to all directions, "direction 0" to "direction 9". The x-axis indicates the initial direction, while y-axis indicates the number of step.



(a) cat (b) mad (c) men

(d) met (e) net (f) new

Fig. 4: The performance test result-2: The changes of Q-value when the initial direction for camera was set to 'direction 1'.

the system could output the result immediately. It is because the other words do no have 'n' as the third character. The same graph characteristic was shown in Fig. 3(f) when the word 'new' is tested. Finally, as shown in Fig. 3(d) and Fig. 3(e), for both words 'met' and 'net', we could observed that the system makes the recognition at direction 3 regardless of the camera initial direction. In order to recognizes the word 'met' with the initial direction 0, the system at least need to move the camera until direction 5 to acquire the third character information to distinguish the word 'met' from the word 'men'. Since the learning system has a memory function, it is expected that the system make recognition at this direction. However, it is observed that, the system returned to direction 3 and outputs the recognition result there. Therefore, even though the system manages to recognize both words correctly, the recognition timing is not optimal. That may be because it is more difficult for the system to recognize the words at any direction where the word can be classified than to recognize them after moving to the direction where the word can be classified easily. The hypothesis can explain that the system has preferable directions to recognize even in the word 'cat'. Longer learning might improve the performance, but it is interesting that the movement back to the left when the character 't' appears at the direction 5 looks as if the system returns to make sure. When the initial direction was on the rightmost side, the system moved the camera to the left until the first character appears.

During the performance test that was shown in Fig. 3, the change of Q-values for all actions when the initial direction was set up at 'direction 1' were recorded and plotted for each word as shown in Fig. 4. The y-axis indicates the Q-value,

while x-axis indicates the number of step in the episode. This graphs show how the system makes the correct recognition. In Fig. 4(a), when the word 'cat' is tested, it is already clear from step 0 that the Q-value corresponding to the word 'cat' is higher than other Q-values and in (b)-(f), Q-value for the word 'cat' was very small. While in Fig. 4(b), the Q-value corresponds to the 'move' action is clearly higher at step 0 because the information is still not enough and that requires the information of the second character of the word. After that, the Q-value corresponding to the word 'mad' became higher at step 1 where the image for the direction 2 was inputted, and the system output the result successfully.

The change in Q-values from step 0 to step 3 were the same between Fig. 4(c) and (d). However, at the step 4, where the fifth image that holds the third character was inputted, as shown in Fig. 4(c), the Q-values started to split depending on the presented word and successfully recognizing the word 'men'. We can observe the same characteristic when the system tries to recognize the word 'new' in Fig. 4(f). Finally, as shown in Fig. 4(d) and (e), at 'step 4' the Q-values for both words were almost the same not depending on the past images. Here, rather than outputting the recognition result, the system chose to move the camera backward and correctly recognized the words at 'direction 3'. It is not the optimal, but this can be accepted since moving the camera backward seems to be a rational option rather than pushing itself to memorize everything.

5 Conclusions

In this paper, it was shown through simulations that both flexible sensor motion and context-based word recognition function learning could be confirmed through reinforcement learning based only from reward and punishment. The learning process was successful and the trained system was able to recognize all the tested words from any initial camera direction. After learning, the system also managed to understand the context of all of the tested words and was capable to make recognition by flexibly moving its sensor though the process is not always optimal.

There are still some limitations for current result. For future work, not only a horizontal discrete action but also a continuous and vertical movement should be considered. Furthermore, since only one type of font was used, the emergence of generalization function is not expected. Varying font type and size, and by using a real camera movement with the perspective distortion must promote its generalization.

Acknowledgment

This research was supported by JSPS Grant-in-Aid for Scientific Research #2350 0245.

References

1. Whitehead, S. D. & Ballard, D. H. (1990) Active Perception and Reinforcement Learning, *Neural Computation*, **2** (4), 409–419.
2. Darrell, T. (1998) Reinforcement Learning of Active Recognition Behavior, *Interval Research Technical Report*, 1997-045
3. Paletta, L. & Pinz, A. (2000) Active Object Recognition by View Integration and Reinforcement Learning, *Robotics and Automation Systems*, **31**, 71–86.
4. Lopez, M. T. et al. (2007) Dynamic Visual Attention Model in Images Sequences, *Image and Vision Computing*, **25**, 597–613.
5. Larochelle, H. & Hinton, G. (2010) Learning to combine foveal glimpses with a third-order Boltzmann machine, *Advances in Neural Information Processing*, **23**, pp. 1243-1251.
6. Denil, M., Bazzani, L. & de Freitas, N. (2012) Learning Where to Attend with Deep Architecture for Image Tracking, *Neural Computation*, **24** (8), 2151–2184
7. Shibata, K. (2011) Emergence of Intelligence through Reinforcement Learning with a Neural Network, *Advances in Reinforcement Learning*, Abdelhamid Mellouk (Ed.), InTech, pp.99-120.
8. Shibata, K., Nishino, T. & Okabe, Y. (2002) Active Perception and Recognition Learning System Based on Actor-Q Architecture, *System and Computers in Japan*, **33**, 12–22.
9. Faudzi, A.A.M. and Shibata, K. (2010) Acquisition of active perception and recognition through Actor-Q learning using a movable camera, *Proc. of SICE Annual Conf.*, FB03-2.pdf.
10. Utsunomiya, H. & K. Shibata (2009) 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*, **5507**, 970–978.
11. Shibata, K. & Utsunomiya, H. (2011) Discovery of Pattern Meaning from Delayed Rewards by Reinforcement Learning with a Recurrent Neural Network, *Proc. of IJCNN (Int'l Joint Conf. on Neural Networks)*. 2011, 1445–1452.
12. Goto, K. & Shibata, K. (2010) Emergence of prediction by reinforcement learning using a recurrent neural network, *Journal of Robotics*, **2010**, Article ID 437654.
13. Faudzi,A.A.M. & Shibata, K. (2011) Context-based Word Recognition through a Coupling of Q-Learning and Recurrent Neural Network, *Proc. of SICE Annual Conf. of the Kyushu Branch*, 155-158.
14. Rumelhart, D.E, Hinton, G.E., & Williams, R.J. (1986) Learning Internal Representations by Error Propagation, *Parallel Distributed Processing*, The MIT Press, 318–362.
15. Watkins C.J.C.H and Dayan P. (1992) Q-learning, *Machine Learning*, **8**, 279–292.