Context-based Word Recognition through a Coupling of Q-learning and Recurrent Neural Network

○Ahmad Afif Mohd Faudzi and Katsunari Shibata, Oita University

Abstract: **In the previous research, it was shown that by employing Actor-Q learning method, a learning of active perception and recognition problem using a real camera was successfully verified. However, because it was trained by a regular layered NN, context-based recognition was not trained. In this paper, the authors applied a RNN that is trained by Q-learning method and observing whether a context-based recognition function can be acquired. A simulation was done on 6 words. As a result, it was verified that flexible context-based word recognition and recognition timing can be learned through a coupling of Q-learning and recurrent neural network.**

1 Introduction

Among the human sensory organs, vision is probably the most informative perception. The eye movement and recognition in human seem very flexible and intelligent. There is a lot of evidence shows that in flexible recognition, contextual information can provide more relevant information for the recognition of an object than the intrinsic object information [1]. For an example, when we read a book, we do not usually seem to read it by recognizing each character one by one. We would predict a word from the first two or several characters, and would utilize the story context to expect the next character or word. The way human moves the eyes, extracts individual character or recognizes patterns, does not seem according to a simple formulaic routine but very flexible. Such flexible eye movement and recognition is achieved by our parallel and flexible brain where learning plays an important role to it.

Recently, the coupling of a neural network (NN) and reinforcement learning (RL) is considered useful in machine learning because its autonomous, adaptive and flexible learning ability. In the research by the authors [2], it was verified that the appropriate camera motion, recognition and recognition timing were successfully acquired through Actor-Q learning method. However, because it was trained by a regular layered NN, the context-based recognition and movement function was not trained. Furthermore, only two patterns were used.

In this paper, context-based word recognition task was trained. A simulation on data that capture by real camera is used and RNN with Q-learning method is applied. By applying RNN, it is verified that recognition and memory function emerge [3]. Here, the authors would like to verify not only just a normal recognition but also whether flexible recognition emerges through this context-based word recognition learning.

2 Learning System for Context-based Recognition

As shown in Fig. 1, a context-based word recognition task using a movable camera as a visual sensor and a monitor to display words are assumed here. Several words are prepared, and the system needs to identify which word is presented.

In the learning process, for every trial, a pattern was chosen randomly. The initial position of the camera is fixed at the left edge of presented word. As shown in Fig. 1, the sensor's field is too small to identify the presented word. In this task, camera can make only a horizontal movement and scanning from left to right with a constant interval. However, to avoid from spending a lot of time for learning, instead of using a real camera, a simulation based on real camera movement was done. Fig. 2 shows the 6 words that were used in the learning process and all

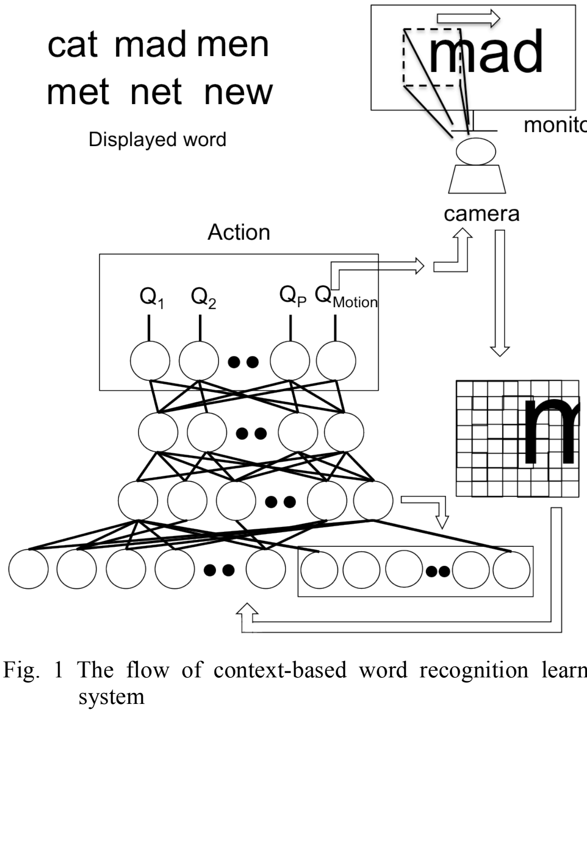


Fig. 1 The flow of context-based word recognition learning system

their partial images. The authors remake these images in order to remove all the noises including background and light effect.

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 trial. Furthermore, as the set of word only contains 6 words, we can expect context-based flexible word recognition. For example, in the case of the word "cat", 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 hold by next partial images to distinguish it from "men" and "met" that started with same 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 is inputted. However, for the word "met" and "net", since the same images inputted from the step 4. Here, in order to distinguish it, the system needs to memorize whether the first character was "m" or "n". Through this learning, the system not only can recognize all tested words correctly but also can recognize them at the appropriate timing.

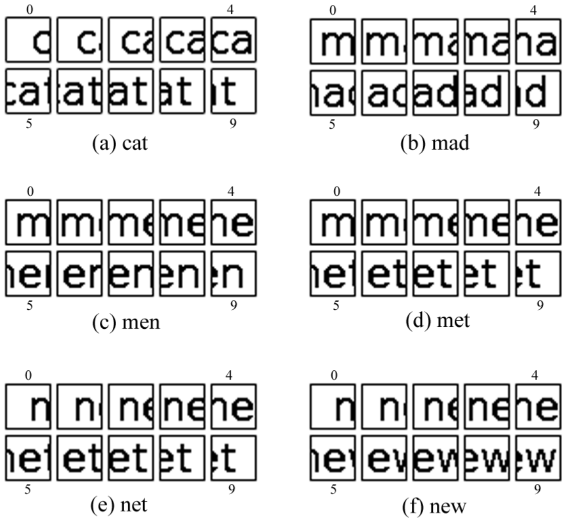
3 Learning Method

As a learning method, one of the reinforcement learning methods, Q-Learning is used [4]. Here, it will computes a teaching signal for recurrent network training. In Q-learning, state-action pairs are evaluated, and the evaluation value is called Q-value. An agent chooses an action with the probability that is calculated from the Q-values.

The algorithm of Q-learning is as follows.

1) An agent observes a current state s_t.
2) The agent selects and executes an action a_t according to Q-values for s_t.
3) The agent observes the state after the transition s_{t+1}.
4) Evaluation is done. A reward or punishment r_{t+1} is given. Only the output for selected action is modified. The teaching signal $T_{a_t,t}$ for corresponding output is computed as
   $$T_{a_t,t} = r_{t+1} + \gamma \ \max_a \mathrm{Q}(s_{t+1}, a) \quad (1)$$
   where γ is a discount factor $(0 \leq \gamma < 1)$.
5) $t \rightarrow t + 1$, and it is returned to the step 2).

In this learning system, the state s is the direct image signals from camera, and the action a is a discrete decision making.



(a) cat

(b) mad

(c) men

(d) met

(e) net

(f) new

Fig. 2 The word samples and all their partial images.

The action a is selected based on ε-greedy. ε is reduced according to the progress of learning using a exponential function in eq. (2)

$$\varepsilon = \exp(-trial \times 0.0005) \quad (2)$$

Trial is the number of the current episode. 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 outputs" for each of the 6 word is 0.1/3 and for "camera movement" is 0.80.

The learning system has a 4-layer Elman-type recurrent neural network. In this type of network, the hidden outputs are fed back to the input layer at the next time step. The inputs are the raw image signals that are linearly converted to a value between -0.5 and 0.5. There are seven output neurons. Each of them represents the Q-value corresponding to one of the seven actions, which are six "recognition result outputs" and "camera movement". The number of neurons in each layer is 576-100-15-7 from the input layer to the output layer.

The output function of each neuron is the sigmoid function with the range from -0.5 to 0.5. However, all the outputs are used after adding 0.5. When the correct result is outputted, the corresponding Q-value is trained to be 0.9. However, if it is not correct, the training signal is smaller by 0.1 than the current Q-value. 0.5 is subtracted from the training signal before using it as the actual training signals in BPTT. 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. That enables the error signal to propagate the past states effectively in BPTT without diverging.

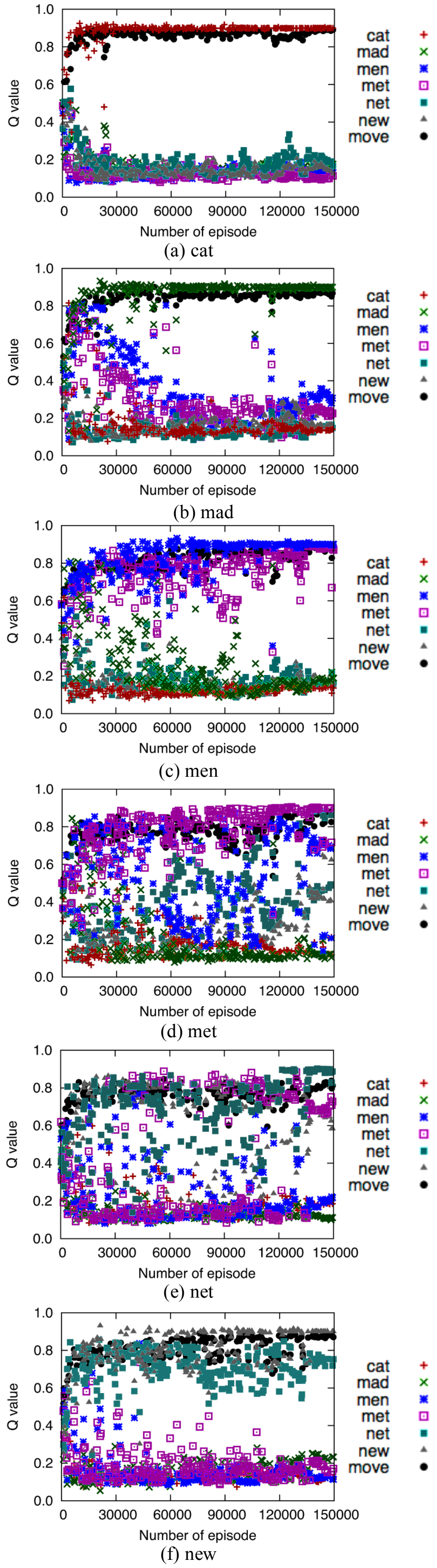
3 Result

As a result, after 150,000 trials, the context-based flexible word recognition function can be achieved. Fig. 3 shows the Q-values for each action at the end of every trial for the presentation of words (a) “cat”, (b) “mad”, (c) “men” (d) “met” (e) “net” and (f) “new” samples respectively.

As shown in Fig. 3(a) “cat”, at first, the Q-value of each action were started to change from 0.0. During learning, the red point (+), which is the Q-value corresponding to the word “cat”, increased and became higher than the other Q-values. At the same time, other Q-values that correspond to the other words were decreased. By these big differences of Q-values, the system can make recognition correctly. This learning curve shown in Fig. 3(b) to (f) also show that the system also can make recognition on other provided words.

Fig. 3 also shows that different word required a different time for learning. As shown in Fig. 2, at one time, the system can only capture a partial of the whole word. So, since the word “cat” is start with character “c” that is different compared to other words, the system can distinguish it even though only the first partial was captured. Here, Fig. 3(a) “cat” clearly shows that Q-value corresponding to the word “cat” became stable earlier compared to others. While Fig. 3(b) “mad” that corresponds to word “mad” shows that the green point (x) was the second one that became stable. It happened because the system need a longer learning to recognize the second character, “a” since the first character is same with word “men” and “met”. Then, it was followed by Q-values in Fig. 3(c) “men” and (d) “met”. For these two words, the system need to recognize all characters since the first and the second characters were the same.

Finally, in Fig. 3(e) “net” and (f) “new”, even though the context are just same like “men” and “met”, we can observe that the system required a longer learning to recognize the word “net” rather than the word “new”. In order to recognize the word “new”, the system need to distinguish it with the word “net” which required the system to recognize it until the third character. When the third character was captured, the system can recognize the word “new”. However, it is hard to recognize the word “net” when the current information is same with the word “met”. Here the system needs the information whether the first character was character “m” or “n”



(a) cat

(b) mad

(c) men

(d) met

(e) net

(f) new

Fig. 3. The changes of Q-values at the end of every trial.

which needed a memory function that required longer learning. The same condition was supposed to happen when the word "met" is presented. However, the reason why the learning for the word "met" was faster than the word "net" is still unknown.

Fig. 4 shows the changes of Q-values for every word after learning process. The *y*-axis indicates the Q-value, while *x*-axis indicates the number of step. 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. However, in other graphs, it was very small. While in Fig. 4(b), the Q-value corresponds to "move" action is higher at step 0 because the information is still not enough and required the information of the second character of the word. After that the Q-value corresponding to the word "mad" became higher at the step 1 and successfully output the result.

As shown in Fig. 4(c) and (d), the Q-values changes from step 0 to step 4 were the same. However, at the step 5, the Q-values started to split depending on the presented word and successfully output the result. Fig. 4(e) and Fig. 4(f) also show the Q-values changes from step 0 to step 4 were the same because the first and second characters were the same. However, the system takes seven and eight step to recognize the words "net" and "new" respectively. It probably suggest that more learning is required to make the system outputted the result earlier.

In Fig. 4(d) and (e), at the step 5, we can see that Q-values correspond to the words "met" and "net" were high. It happened because at the step 5, the current information is completely the same for both cases. At this time, we could say that the system manages to recognize both the words by understanding the context of the word by memorizing the earlier information.

4 Conclusion

In this paper, a simulation of context-based word recognition through a coupling of RNN and Q-learning was verified. The learning was successful and the system can recognize all the tested word. Through this learning, the system also managed to understand the context of all the tested word and was capable to make recognition in an appropriate timing.

We are currently working on a system that not only could decide a discrete intention but also can make a continuous movement. In addition, in order to investigate the performances in a more difficult task, more word samples and real capture images will be used.

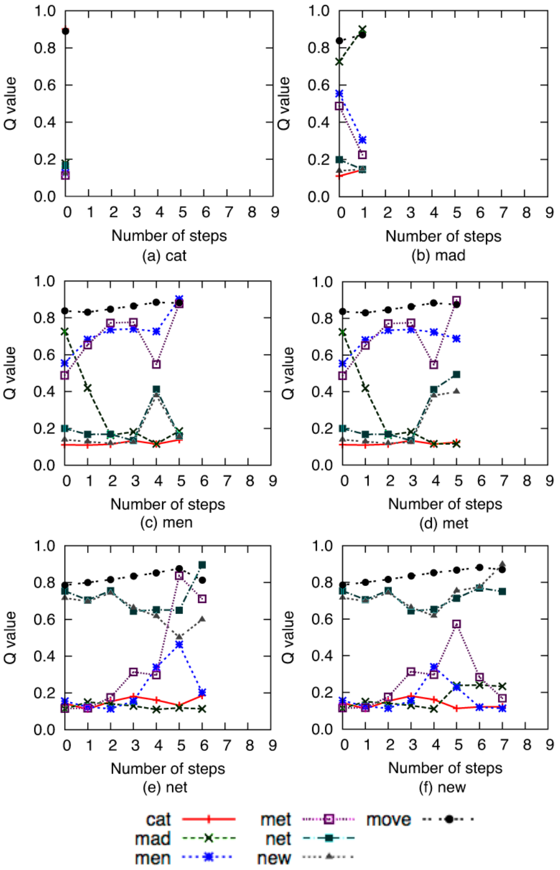


Fig. 4. The changes of all Q-values for each word after learning.

Acknowledgment

This research was supported by JSPS Grant in-Aid for Scientific Research #19300070 and #23500245. Special thanks to all members of Shibata Laboratory for support.

References

[1] A. Torralba. Contextual priming for object detection. Intl. J. Computer Vision, 53(2):153–167, 2003

[2] A.A.M. Faudzi and K. Shibata, Acquisition of active perception and recognition through Actor-Q learning using a movable camera, Proc. of SICE Annual Conf. 2010, FB03-2.pdf, 2010.

[3] K. Goto and K. Shibata, Emergence of prediction by reinforcement learning using a recurrent neural network, Journal of Robotics, Vol. 2010, Article ID 437654, 2010.

[4] Watkins C.J.C.H and Dayan P, "Q-learning", Machine Learning, Vol. 8, pp. 279-292, 1992.