Formation of Attention and Associative Memory based on Reinforcement Learning

Katsunari Shibata
Department of Electrical & Electronic Engineering, Oita University, Oita 870-1192, Japan
(Tel&Fax: +81-97-554-7832; Email:shibata@cc.oita-u.ac.jp)

Abstract: An attention task, in which context information should be extracted from the first presented pattern, and the recognition answer of the second presented pattern should be generated using the context information, is employed in this paper. An Elman-type recurrent neural network is utilized to extract and keep the context information. A reinforcement signal that indicates whether the answer is correct or not, is only a signal that the system can obtain for the learning. Only by this learning, necessary context information became to be extracted and kept, and the system became to generate the correct answers. Furthermore, the function of an associative memory is observed in the feedback loop in the Elman-type neural network.

Keywords: attention, associative memory, reinforcement learning, recurrent neural network, context information

1. Introduction

It is required for robots in the real world to manage a huge amount of sensory signals. In order to select the necessary information among them, "active perception" and "selective attention" are useful. In the case of selective attention, context information is often utilized. It is difficult to select necessary information according to the context information, but it is also difficult to decide which information should be extracted as the context information from huge sensory signals in the past.

Sakaguchi et al. proposed to select the information source by which the entropy of the object model decreases the most[1]. However, the handling of the context information was not mentioned. McCallum mentioned both selective attention and short-term memory[2]. In the paper, attention and memory mean that the state can be identified by the past sensory signals. In this paper, attention means to focus on a part of the huge sensory signals using context information.

On the other hand, in associative memories using a mutually-connected neural network (NN), the coding of the information is usually given in advance. In order to obtain the associative memory adaptively, Hebb rule or an extension of Hebb rule, such as covariance learning, is often employed. In such a learning, however, a pattern that is presented frequently is memorized not depending on whether the pattern is necessary for the task or not. Furthermore, small images are often used directly as memorized patterns. However, since vision sensory information has huge signals actually, it is not effective to memorize them directly and it is inevitable to compress them by extracting only necessary information.

The coding of the compressed information has been often decided on the basis of how exactly the original input pattern can be restored, such as principle component analysis, or sand-glass neural network in which an identical mapping is learned. However, the restoration itself is not the purpose usually, and such coding sometimes includes unnecessary information for its purpose. The coding should be decided by the necessity in a given task.

Zipser focused on delayed match to sample tasks using monkeys[3]. In this task, since the monkey has to answer the presented pattern after some while, and so it has to keep the information about the presented pattern. They proposed an Elman-type recurrent neural network (NN)[4] trained by BPTT (Back Propagation through Time) as a model of the monkey[5]. An analog signal and a gate signal are the inputs of the NN, and the NN is trained so as that the analog input when the gate input was activated, is maintained like a flip-flop. Then it was shown that the NN picks a new value when the gate signal is activated, keeps the value when the gate signal is not activated, and obtains a dynamics to converge a fixed point. It was also shown that the output pattern is similar to the activating patterns of real neurons in the actual brain. However, recognition, attention, and autonomous coding in associative memory were not considered because the input signal is only one analog signal in spite of a multiple dimensions of input like an image, and it is not required to be compressed to be memorized.

The author has dealt with a simple image as input signals, and shown that the function of context extraction, associative memory, and attention can be obtained through the learning of delayed recognition or attention task using a recurrent NN[6]. Fig. 1 shows the system employed in this paper, and is used also for the explanation of the author's previous works. In the delayed recognition task, one arrow pattern that points one of the four corners is presented at first as the left half of Fig. 1, and after some while, it is required to output the classification result of the arrow direction. In the attention task, one arrow pattern is also presented at first

A part of this research was supported by the Scientific Research Foundation of the Ministry of Education, Culture, Sports, Science and Technology of Japan (#13780295)

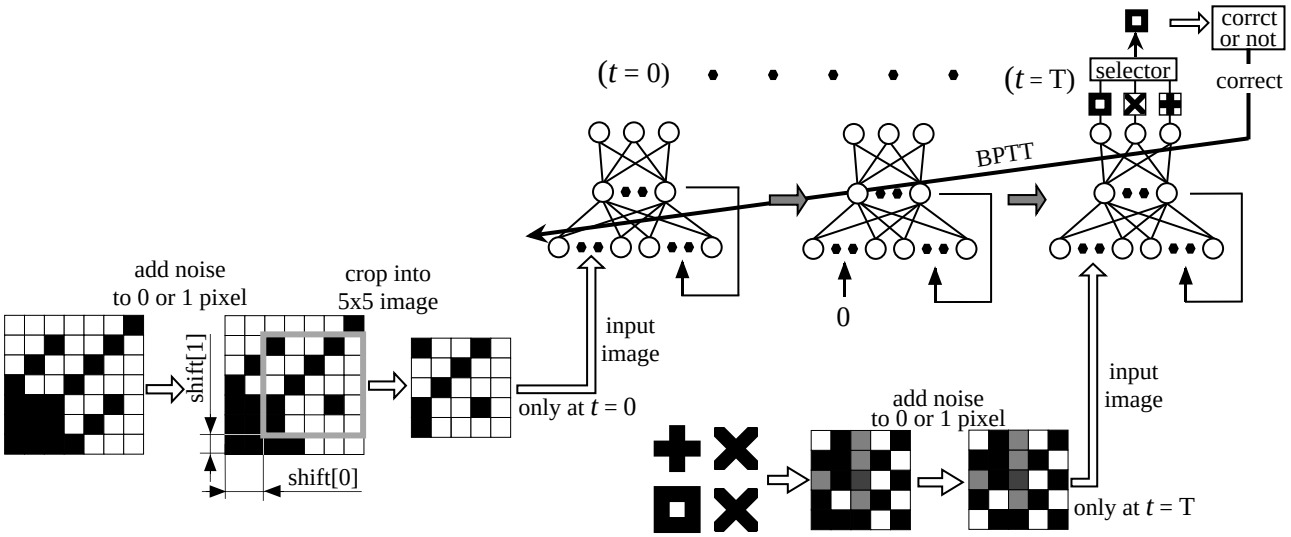

Fig. 1. An attention task that requires context information.

without any noises. After some while, another pattern, which consists of 4 small sub-patterns, is presented on the same visual sensor as Fig. 1. Then, the system is required to classify the sub-pattern at the corner where the first presented arrow pattern pointed. The pattern can be classified into one of two categories. The input signal has multiple dimensions, and system is required to reduce the dimensions to keep the information in the limited number of hidden units. It has been seen that the dynamics of the NN is fixed-point convergence, and for any input patterns including the patterns that are not used in the learning, the hidden state converged to one of the 4 fixed-points, each of which corresponds to each arrow direction. The dynamics was formed so adaptively that the time when the second pattern was presented was fixed, it was observed that the hidden state became the state corresponding to the arrow direction at the fixed time. But, after that, the state changed and finally converged to the other state that corresponded to the other arrow direction. Further when the variation for each arrow direction was only one, the other basin appeared that did not correspond to any arrow directions.

In the task here, only a reinforcement signal, which represents whether the classification is correct or not, is given for the classification answer made by the system. The other knowledge is not given for the learning including how to extract the necessary information, whether the system should keep some information or not, how to code the stored information in the hidden layer, and how to use the context information to classify the second presented pattern. The sub-pattern should be classified into one of three patterns. Furthermore, one pixel is inverted at random as a noise even in the first presented arrow pattern as well as the second presented pattern. It is examined through a simulation whether only the arrow direction becomes to be extracted and kept by the learning based on BPTT even though some kinds of arrow patterns for each direction are presented. Then, it is confirmed that the system classified the corresponding sub-pattern correctly using context information and the hidden state has the dynamics of fixed-point convergence, in other words, the function of associative memory is observed in the hidden layer.

2. Attention Task

Here, an attention task is explained in detail. As mentioned above, it requires context information, and is more difficult than a delayed match to sample task. Fig. 1 shows the flow of the task. A part of an arrow image is presented, and it is put into an Elman-type recurrent NN at $t = 0$. The size of the original arrow image is 7×7, and its direction can be one of 4 directions, "upper right", "upper left", "lower right", and "lower left". With the ratio of one-half, the value of only one pixel is inverted randomly as a noise. The visual sensor consists of $5 \times 5 = 25$ visual cells, and one part of the presented arrow image is cropped. So totally $3 \times 3 = 9$ patterns can be presented for each direction of the arrow if the noise is not added. Considering the noise, $9 \times (25+1) = 234$ patterns can be presented for each arrow direction. The input signal is -1.0 for the white pixel, and 1.0 for the black one. At a randomly selected time from 5 to 14, which is denoted by T, a pattern that consists of 4 small sub-patterns is presented on the same visual sensor. The size of the sub-pattern is 3×3, and it can be "square", "cross", or "plus". Since the size of the visual sensor is 5×5, at the middle row and middle column, the sub-patterns are overlapped. The sensory signal from such sensory cell is the average of the overlapped sub-patterns. For example, in Fig. 1, the direction of the arrow is "lower-left", and in the pattern presented at $t = T$, the shape at the lower-left corner is "square".

There are three output units, and answer of the system is decided according to the probability that is proportional to the sum of the output and 0.5. The output function of each unit is sigmoid function whose value range is from -0.5 to 0.5. When the answer is correct, the system obtains a reward 1.0, while when not correct, it obtains a penalty -1.0 as a reinforcement signal r. The corresponding output is trained to be $0.4 \times r$. The other outputs are trained to be -0.4 when the answer is correct, and otherwise they are not trained. So when the answer is not correct, the system cannot know directly which is the correct answer. At the other time steps, all the input signals are 0.0. The number of hidden units is 20, and the values of the hidden units are 0.0 at $t = 0$. Initial weight values are 0.0 for the hidden-output connections. As for the feedback connection of the hidden-hidden connections, the weight value is 4.0 for the self-feedback connections, and 0.0 for the others. The reason why the self-feedback connection weight is set to be 4.0 is that the maximum derivative of the output function is 0.25, and the error signal propagates backward effectively through time without diverging because $0.25 \times 4.0 = 1.0$. For the input-hidden connections, it is decided randomly from -1.0 to 1.0. When only the hidden layer is seen, there is a mutually-connected NN with 20 units. When the mutually-connected NN is utilized for an associative memory, the connection weights are always symmetrical, because the network dynamics always becomes fixed-point convergence when the weights are symmetrical. However, here, no such constraint is given in advance.

3. Simulation Result

1000000 trials of learning was done. After learning, if the maximum output supposed to be the answer, a wrong answer appears about once per 10000 trials. Depending on the initial connection weight values in the NN, it sometimes fails to learn. The learning curve is shown in Fig. 2. The value of the vertical axis shows the average error. For the output corresponding to the correct answer, the error is 0.0 when the output is larger than 0.4, and otherwise it is the square of the difference from 0.4. For the other outputs, the error is 0.0 when the output is smaller than -0.4, and otherwise it is the square of the difference from -0.4. The sum of the errors for three output units was computed and then the average of the sum was computed over 10000 trials. The average error is also plotted for the case that the supervised signals for all the outputs were given. The initial weights and learning rate is the same. It is seen that supervised learning is faster than reinforcement learning. If the number of the outputs becomes larger, the difference of the learning speed is thought to become larger.

At the next, the context extraction and associative memory function is observed. The first presented patterns should be classified into 4 categories, because only

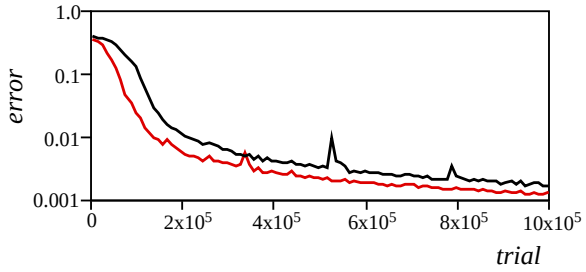

Fig. 2. Learning curves for the cases of supervised learning and reinforcement learning.

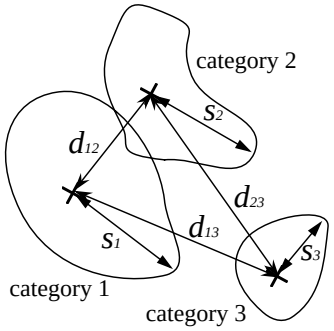

Fig. 3. Some defined variables.

the direction of the arrow pattern is utilized to give an attention to the second presented pattern. As mentioned above, totally 234 patterns can be taken as one category. Here, the distance between two patterns in a layer is defined as the sum of the absolute value of the difference of each unit. Then some variables are defined as shown in Fig. 3, but for the case of three categories and two-dimensional space for simplicity. s_i indicates the maximum distance from the average pattern over the whole the patterns in the category i. d_{ij} indicates the distance of the average patterns between category i and j. σ_i indicates the standard deviation of the patterns in the category i, and σ indicates the standard deviation of whole the patterns.

Fig. 4 shows the change of s_i and d_{ij} from $t = 0$ to $t = T - 1$ after being normalized by σ to see the relative distance. For simplicity, the data from the last 1000 trials were utilized on behalf of observing over all the possible input patterns. The distance between the category i and j d_{ij} becomes larger through time for any combinations of the categories. While, the maximum distance from the average s_i in one category becomes smaller through time for any categories, that is seen on the dotted background. It is also seen that at $t = T - 1$, $\min_{i,j} d_{ij}$ is larger than the twice of $\max_i s_i$. This means that the sphere whose center is the average hidden pattern and whose radius is $\max_i s_i$, includes whole the hidden patterns belonging to the category and also does not includes any hidden patterns belonging to the other categories. While, at $t = 0$, the maximum distance from the average in one

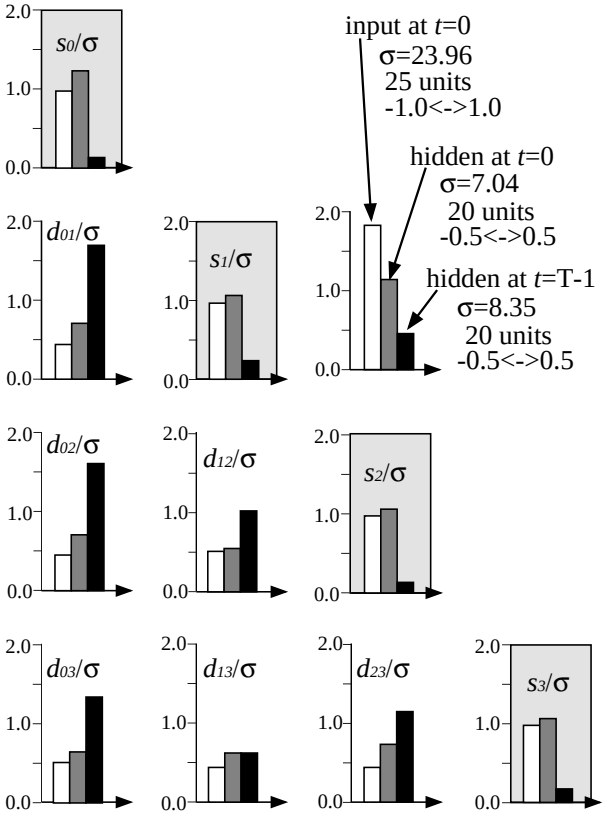

Fig. 4. The change of the normalized distances between categories and the size of each category through time.

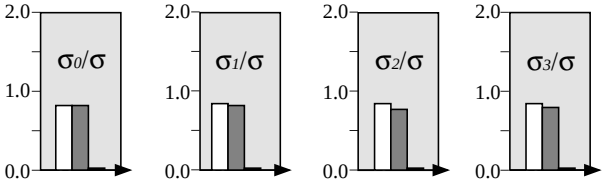

Fig. 5. The change of the standard deviation of the patterns in each category through time.

category is always larger than the distance between any combinations of categories.

Fig. 5 shows the change of σ_i from $t = 0$ to $t = T - 1$. From this figure, it is seen that the standard deviation became very small at $t = T - 1$. $\max_i \sigma_i$ was 0.26, while $\max_i s_i$ was 1.90. If only one of the hidden units takes different values and they are 0,4 and -0.4, the distance becomes 0.8. Accordingly the dynamics of the recurrent network is almost fixed-point convergence even though there are 234 patterns for each category. In the case when the system made a wrong answer, the interval T is 5 or 6. It is supposed that if the remind time is longer, the system can generate the correct answer.

In order to know the size of the basin corresponding to each category, input signals were set randomly and hidden state at $t = 13$ was observed. Table 1 shows the number of hidden states whose distance is less than 1.9 from the average hidden state of one category. As mentioned, 1.9 is $\max_i s_i$. It is seen that the number varies, in other words, the size of the basin varies very much depending on the category.

Table 1. The size of the basin in whole the input space. Input signals were set randomly, the hidden state at $t = 13$ is classified and the number for each category was counted when 10000 random patterns were presented.

category	0	1	2	3	misc
normal	1078	3828	1063	3862	169
asym_in	1006	5048	1178	2404	364

Finally, only for the "lower-left" arrow pattern, the variety of cropping ways of the arrow image is limited to only one in spite of $3 \times 3 = 9$. It was expected that the size of the basin changes adaptively, i.e., the basin for the "lower-left" arrow becomes small. The standard deviation of the hidden state for the "lower-left" arrow at $t = 0$ was 0.80, while that for the other arrows was more than 5.0. However, the size of the basin was not different so much from the normal case as the lower row of Table 1.

4. Conclusion

It has been shown that context extraction and short-term memory with the associative memory function can be obtained in a recurrent neural network through learning only by the reinforcement signal indicating whether recognition answer is correct or not. The dynamics of the recurrent network was almost fixed-point convergence.

References

[1] Sakaguchi, Y., "Sensory Integration and Active Perception in Tactile Perception", *J. IEICE*, **76**(11), pp.1222–1227, 1993 (in Japanese).

[2] McCallum, A. K., "Learning to Use Selective Attention and Short-Term Memory in Sequential Task", *From Animals To Animat*, pp.315–324, 1996.

[3] Funahashi, S., Bruce, C. J. & Goldman-Rakic, P. S., "Visuospatial Coding in primate prefrontal neurons revealed by oculomotor paradigms", *J. Neurophysiology*, **63**, pp. 814–831, 1990.

[4] Elman, J. L., "Finding Structure in Time", *Cognitive Science*, **14**, pp. 179–211, 1990.

[5] Zipser, D., "Recurrent Network Model of the Neural Mechanism of Shot-Term Memory" *Neural Computation*, **3**, pp. 179–193, 1991.

[6] Shibata, K. and Ito, K., "Formation of Attention and Associative Memory Based on Recognition Learning", *IEICE Technical Report (Neurocomputing)*, NC99-137, pp. 153–160, 2000 (in Japanese).