Artif Life Robotics (2004) 7:145–150
DOI 10.1007/s10015-003-0269-1

ORIGINAL ARTICLE

Katsunari Shibata · Masanori Sugisaka

Dynamics of a recurrent neural network acquired through learning a context-based attention task

Received and accepted: July 11, 2003

Abstract Selective attention is a very important function for robots acting in the real world. In this function, not only attention itself, but also context extraction and retention are very intelligent processes and are not easily realized. In this article, an attention task is presented in which context information must be extracted from the first pattern presented, and using the context information, a recognition response must be generated from the second pattern presented. An Elman-type recurrent neural network is used to extract and retain the context information. The reinforcement signal that indicates whether the response is correct or not is the only signal given to the system during learning. By this simple learning process, the necessary context information got to be extracted and retained, and then the system changed to generate the correct responses. The function of associative memory was also observed in the feedback-loop in the Elman-type neural network. Furthermore, the adaptive formation of basins was examined by varying the learning conditions.

Key words Attention · Associative memory · Context extraction · Recurrent neural network · Adaptive basin formation · Reinforcement signal

1 Introduction

It is necessary for robots acting in the real world to handle huge quantities of sensor signals. In order to extract the necessary information for a given task, both "active perception" and "selective attention" are indispensable. Some context information is often used to suggest which information is necessary in the present sensor signals. However, the context information itself should be extracted from the series of past information, and it should also be retained until the time when the context information is needed.

Sakaguchi[1] proposed selecting the information source by which the entropy of an object model decreases the most. However, the handling of the context information was not mentioned. McCallum[2] mentioned both selective attention and short-term memory. In that article, attention and memory meant that the state could be identified by the previous sensor signals. In this article, attention means to focus on one part of many sensor signals using some context information.

On the other hand, in associative memories using a mutually connected neural network, the coding of information to be memorized is usually given in advance. In order to obtain the associative memory function adaptively, the Hebb rule, or an extension of the Hebb rule such as covariance learning, is often employed. In such learning, however, a pattern that is presented frequently is memorized regardless of whether the pattern is necessary for the task or not. Furthermore, small images are often used directly as memorized patterns. However, since visual sensory information has huge number of signals in most cases, it is not effective to memorize all the signals directly without any processing; it is necessary to compress them by extracting only the necessary information.

In the field of information compression, the coding of compressed information has often been decided on the basis of exactly how the original input pattern can be restored, such as principle component analysis, or a sand-glass neural network in which an identical mapping is learned. However, when some context information is memorized after being compressed, the restoration itself is not usually the main purpose in the robotics field, and such coding often includes unnecessary information. The coding should be decided by necessity for given tasks.

Zipser focused on a delayed match to sample tasks using monkeys.[3] In these tasks, since the monkey is required to indicate which pattern was presented earlier, it has to retain

K. Shibata (✉) · M. Sugisaka
Department of Electrical and Electronic Engineering, Oita University, 700 Dannoharu, Oita 870-1192, Japan
Tel. +81-97-554-7832; Fax +81-97-554-7832
e-mail: shibata@cc.oita-u.ac.jp

This work was presented in part at the 7th International Symposium on Artificial Life and Robotics, Oita, Japan, January 16–18, 2002

information about the pattern presented. He proposed using an Elman-type recurrent neural network[4] trained by back propagation through time (BPTT)[5] as a model of the monkey.[6] An analog signal and a gate signal were the inputs to the neural network. The network was trained to behave like a flip-flip, so that the analog input the last time the gate input was activated continues to be output as long as the gate signal is not activated. It was then shown that the neural network picks a new value when the gate signal is activated, keeps that value when the gate signal is not activated, and acquires a dynamic to converge to a fixed point. It was also shown that the output pattern is similar to the activating patterns of real neurons in the 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 the multiple dimensions of input, such as an image, and it does not need to be compressed to be memorized.

We have dealt with a simple image as the input signal, and shown that the function of context extraction, associative memory, and attention can be acquired through the learning of a delayed recognition or attention task using a recurrent neural network.[7]

In these tasks, the outputs of the neural network have one-to-one correspondence with the patterns presented. In the delayed recognition task, a pattern is presented first, and after some time, the network is trained according to the training signals in which only the corresponding output to the presented pattern is large. In the delayed-attention task, an arrow pattern was presented first, and after some time, another pattern was presented in which some small subpatterns were joined. Then the network was trained according to the training signal that indicated the subpattern in the corner that was pointed out by the first arrow pattern.

After learning, the pattern could be classified into one of two categories. The input signal had multiple dimensions, and the system was required to reduce the dimensions to keep the information in a limited number of hidden units. It was found that the dynamics of the neural network was fixed-point convergent, and for any input patterns, including the patterns that were not used in learning, the hidden state converged to one of the four fixed points, each of which corresponded to an arrow direction. This means that the system could not only learn to pay attention according to the context, but could also learn that the arrow direction is important for the given task. Adaptability in the form of the basin was also observed.

We also showed a similar result for the case where only a reinforcement signal, which represents whether the classification response is correct or not, is given. Other knowledge, for example, how to extract the necessary information, whether the system should retain some information or not, how to code the stored information in the hidden layer, and how to use the context information to generate an appropriate response from the second pattern presented, was not given. Through a simulation, we checked whether only the arrow direction was extracted and retained by learning based on BPTT even though the arrow patterns varied even for the same arrow direction. It was also confirmed that the system classified the corresponding subpattern correctly using the context information, and that the hidden state had the dynamics of fixed-point convergence, in other words, the function of associative memory was observed in the hidden layer.[8]

In this work, the network dynamics acquired through learning was observed in detail. We also report the results of an examination of the flexibility in the basin formation.

2 Context-based attention task

Figure 1 shows the attention task and system employed in this work. One arrow pattern pointing in the direction of one of four corners is first presented as on the left-hand side of Fig. 1. After a while, another pattern, which consists of four small subpatterns, is presented on the same visual sensor as on the right-hand side of Fig. 1. The system is then required to classify the subpattern at the corner where the first arrow pattern pointed. The subpattern is classified into one of three categories.

The arrow direction presented first can be upper right, upper left, lower right, or lower left. The size of the original arrow image is $7 \times 7 = 49$. One pixel value, selected randomly, is inverted as noise with a probability of 0.5. The visual sensor consists of $5 \times 5 = 25$ visual cells, and one part of the original arrow image is cropped. So a total of $3 \times 3 = 9$ patterns can be presented for each arrow direction if the noise is not added. For the noise, $9 \times (25 + 1) = 234$ patterns can be presented for each arrow direction. A 5×5 image is put into an Elman-type recurrent neural network. The input signal is -1.0 for the white pixel, and 1.0 for the black one. The time is set as $t = 0$.

At a randomly selected time in the range from 5 to 14 at each trial, which is denoted by T, a pattern that consists of four small subpatterns is presented on the same visual sensor that the first arrow pattern was presented on. The size of the subpattern is 3×3, and it can be a square, cross, or a plus. Since the sensor size is 5×5, the subpatterns overlap each other in the middle row and the middle column. In such areas, the sensor signal is the average value of the overlapping pixels.

There are three output units, each of which corresponds to one of the subpatterns. The response from the system is decided according to the probability, which is proportional to the sum of each output and 0.5. The output function of each unit in the network is a sigmoid function whose value ranges from -0.5 to 0.5 except for the input layer. When the response is correct, the system gets a reward of 1.0, otherwise it gets a penalty of -1.0 as a reinforcement signal, r. The training signal for the output corresponding to the response is set at $0.4 \times r$. The training signal for the other outputs is set at -0.4 when the response is correct, and otherwise it is not given. This means that the system cannot know the correct answer directly when the answer is not correct.

At every time step from $t = 1$ to $t = T - 1$, 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 $t = 0$. The initial weight

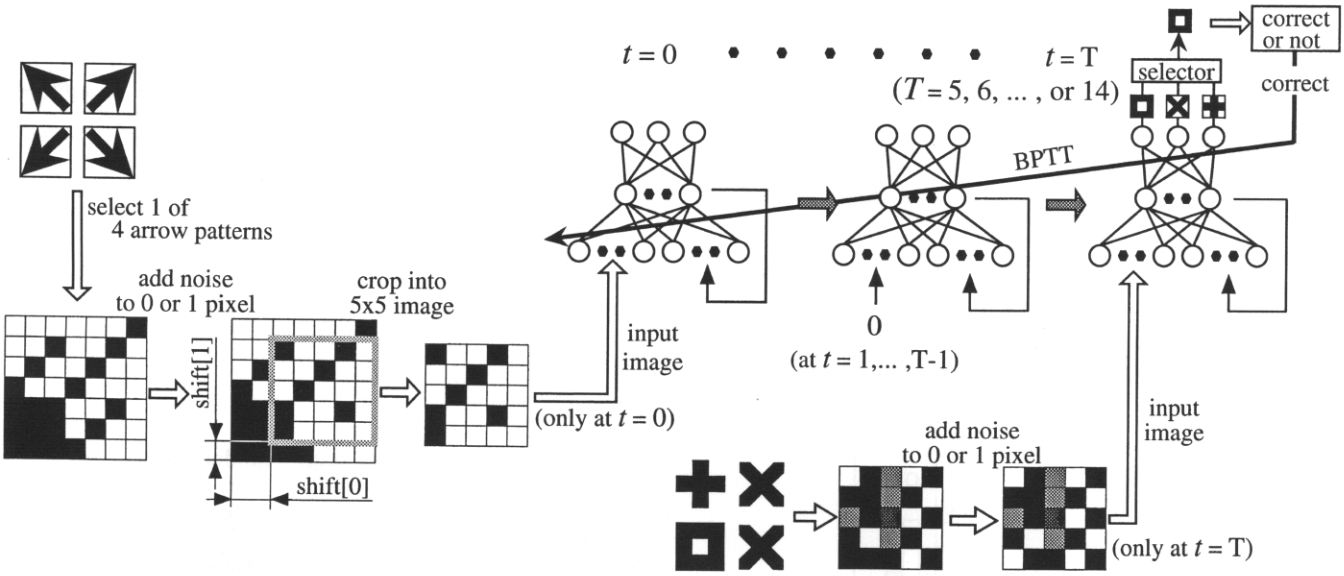

Fig. 1. The flow of the context-based attention task discussed in this article. Two 5×5 binary patterns are presented, with an interval T that is selected in the range from 5 to 14 time-units. The first pattern is part of one of four 7×7 arrow patterns. The second pattern consists of four small 3×3 subpatterns. The system is required to state which is the subpattern at the corner pointed at by the first arrow pattern. Thus, the system should extract the direction of the arrow, but it is only told whether the recognition response is correct or not

values are 0.0 for the hidden-output connections, and are decided randomly from -1.0 to 1.0 for the hidden-input connections. For the hidden–hidden feedback connections, the weight value is 4.0 for the self-feedback connections, and 0.0 for the others. The self-feedback connection weight is set at 4.0 because the maximum derivative of the output function is 0.25 around $input = 0.0$, and the error signal effectively goes backward through time without diverging because $0.25 \times 4.0 = 1.0$.

When the mutually connected neural network is used for an associative memory, the connection weights are usually symmetrical, because the network dynamics always becomes fixed-point convergent when the weights are symmetrical. Hebb learning, which is often employed for the learning of associative memory, cannot realize asymmetrical connections. Here, although the initial connections are symmetrical, no such constraint is imposed during learning.

3 Simulation result

Some simulation results after 1000000 learning trials are shown in this section. One trial is defined as the sequence from the presentation of the arrow pattern to the response and learning. 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 response, when the output is larger than 0.4, the error is 0.0, 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 over 10000 trials was computed. If the maximum output is supposed to be the correct response, a wrong response appeared about once per 10000 trials. Depending on the initial connection weight values in the neural network, it sometimes fails to learn.

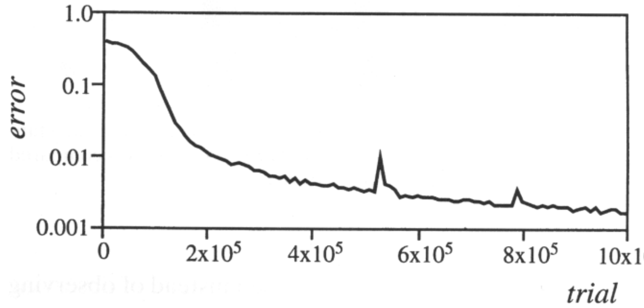

Fig. 2. Learning curve

At the next stage, the context extraction and associative memory function are observed. The first patterns presented should be classified into one of four categories because only the direction of the arrow pattern is needed when considering the second pattern presented. As mentioned above, a total of 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 in each unit.

Figure 3 shows the change in the standard deviation of each category σ_i and the distance between the centers of two categories d_{ij}. Figure 4 shows a rough image of the variables. These variables are shown for the input pattern at $t = 0$, and the hidden patterns at $t = 0$ and $t = T - 1$ after being normalized by the standard deviation of all the patterns σ in order to observe the relative relations. For simplicity, the

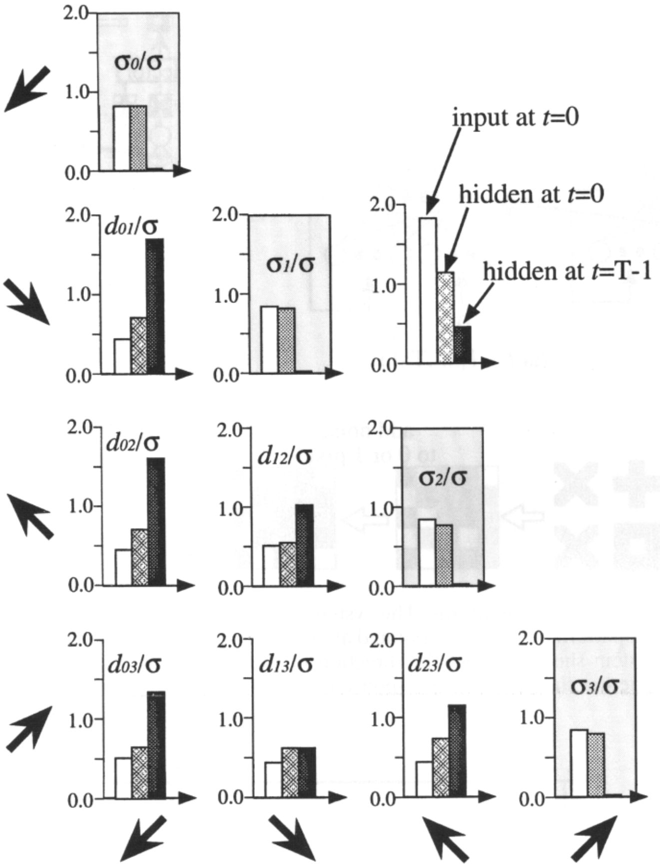

Fig. 3. The change in the distances between categories and the standard deviation of one category. All values are relative values compared with the standard deviation of all patterns

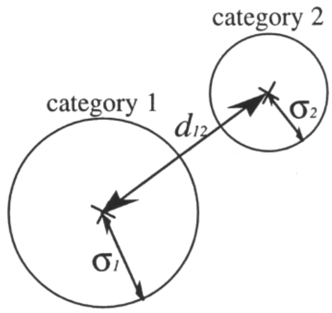

Fig. 4. The variables used in Fig. 3

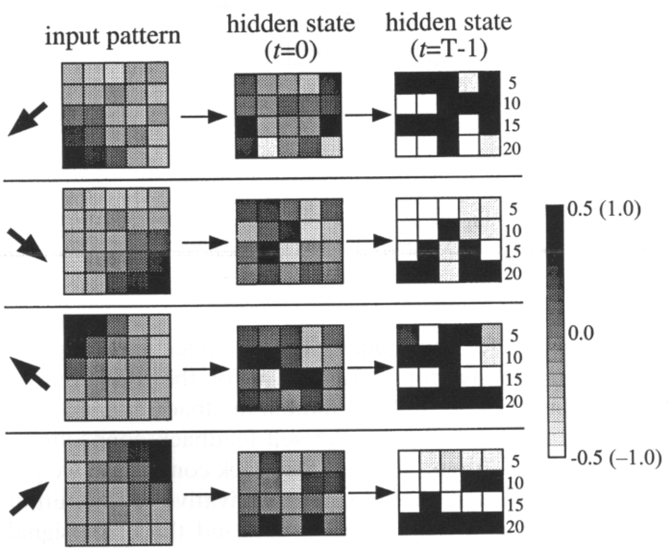

Fig. 5. The changes in the average patterns of input and hidden layers for each category

data from the last 1000 trials were used instead of observing all the possible input patterns.

When the variables for the input pattern at $t = 0$ are seen, the standard deviation of each pattern is larger than the maximum distance between two categories, $\max_{i,j}(d_{ij})$. The distance d_{ij} becomes larger at the hidden layer than at the input layer at $t = 0$ for any combinations of categories. The distance between the hidden patterns also becomes larger through time. While the standard deviation σ_i in each category becomes almost 0.0 at $t = T - 1$, it is far smaller than the minimum distance between two categories, $\min_{i,j}(d_{ij})$. This means that the dynamics of the recurrent network is almost fixed-point convergent, and one fixed point is formed for each category. In the cases where the system gave an incorrect response, the interval T was 5 or 6. It is presumed that if the reminder time is longer, the system could generate the correct answer.

Figure 5 shows the changes in the average patterns for each category for three cases, i.e., for the input pattern at $t = 0$, and the hidden patterns at $t = 0$ and $t = T - 1$. In the average input pattern, one pixel value at a corner takes 1.0 with a high probability, but since a noise is added to one pixel, it is not 1.0 exactly. In the average hidden pattern at $t = 0$, no values are close to the maximum value 0.5 or the minimum value -0.5, while at $t = T - 1$, almost all the values are close to 0.5 or -0.5. The dynamics of fixed-point convergence can also be seen in this figure.

Figure 6 shows two examples of the dynamics. Here, 15 pixels in a total of 25 pixels have a different value between the two input patterns even though both were generated from the same original arrow pattern. Even in the hidden patterns at $t = 0$, the values are still different by more than 0.5 in 15 units of a total 20 units. However, as in the lower part of the figure, both hidden states converge to the same hidden state.

In order to find the size of the basin corresponding to each category, the input signals were set randomly and the hidden state at $t = 100$ 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, because 1.9 is the maximum distance from the average hidden state to one hidden state in the same category at $t = T - 1$. It can be seen that the number, in other words the size of the basin, varies considerably depending on the category. The variation depends on the initial connection weights of the neural network.

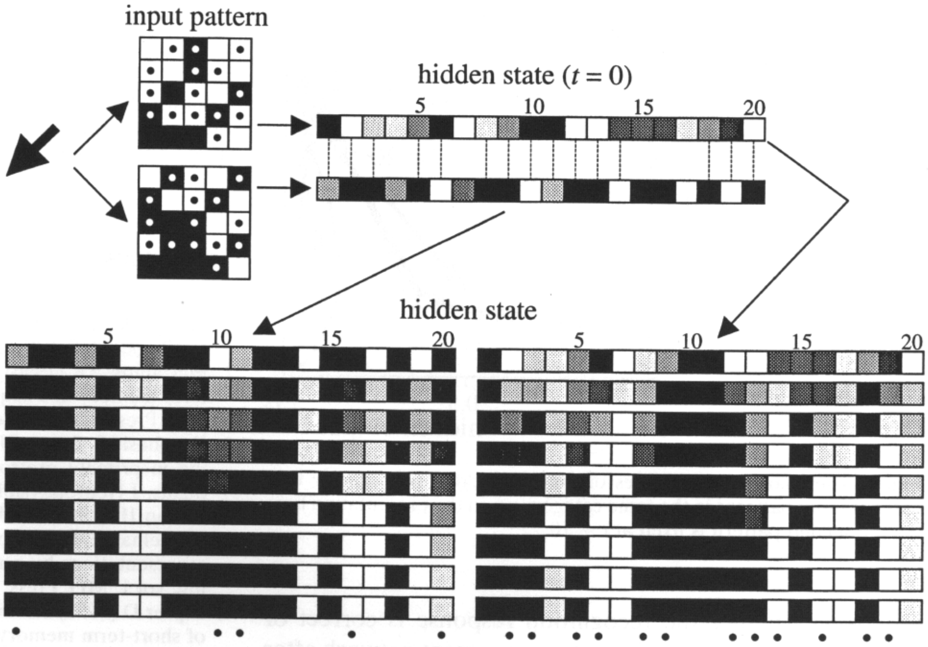

Fig. 6. Two examples of the hidden-state change. The dynamics of fixed-point convergence can be observed. The *dots* in the input patterns indicate that the values are different between corresponding pixels. The *dots* below parts of the hidden states indicate that the value changed by more than 0.5 over time. Both input patterns originated from the lower left directed *arrow*, but more than half of the pixels are different

Table 1. Adaptive change in the basin size for each category in the whole input space according to the learning condition. The parentheses indicate cases where the hidden state at $t = T - 1$ is different from the convergent hidden state

Condition	Category				
	0	1	2	3	Others
Normal	1075	3853	1160	3912	0
Fixed_sensor (categories 2, 3)	2039	7527	248	186	0
Fixed_sensor (all categories)	841	468	(3528)	4670	493
3 categories	2138	2203	5627	–	32
Fixed interval ($T = 10$)	1286	(5719)	1306	1679	10

When the number of cropping ways into a 5×5 image from the original 7×7 arrow pattern is limited to only one in spite of $3 \times 3 = 9$ for two of the four categories, the basins change, as in the second row in Table 1. It can be seen that the basin becomes smaller for the category for which the cropping way was limited. In the other two simulations with different initial connection weights, a difference can be observed, but is not so clear. There is one case in which the basin is larger in one category of limited cropping way than in one category of the normal way.

When the cropping way was limited to one for all four categories, learning was faster and more stable, and eight basins were formed. The dynamics seems complicated in this case. That may be because the basins are small and not so steep, and therefore they do not cover the whole input space. It takes a long time for the hidden pattern to converge. The numbers in the third row in Table 1 are counted when $t = 200$ for this condition only. Some of them do not come into any of the four categories. The number for category 2 is put in parentheses because the final convergence point is different from the hidden state at around $t = T - 1$.

When the number of categories is reduced to three, the basins change as in the fourth row of Table 1. It can be seen that the number of large basins is three, and another small basin was formed. In the other two simulations with a different random number sequence, the number of basins formed is only three. It is considered that the number of basins becomes equal to the number of categories when the input pattern used in learning varies to some degree.

When the presentation time of the second pattern is fixed at $t = 10$ while learning, the four main basins are formed as in the fifth row of Table 1, but for category 1, the hidden state at $t = 10$ is different from the final fixed point. In this case, the values of the three hidden neurons changed after $t = 10$. Figure 7 shows how such neurons change their values. The x-axis shows the output of one of the hidden neurons, and the y-axis shows the output of another. Each of the four lines in this figure shows the change in the hidden neurons' output for each of four input patterns in category 1. It can be seen that the hidden states were almost all the same at $t = 10$, and the speed of change in the hidden state became slow. However, after that, the change gradually became fast again, and finally converged to the real fixed point. In a simulation with different initial connection weights, four categories can be distinguished from each other, but only three basins are formed. This means that the hidden state at $t = 10$ for one category changes and finally converges to the fixed point for another category.

4 Conclusion

It has been shown that context extraction and short-term memory with the associative memory function can be acquired in a recurrent neural network through learning a delayed-attention task only by the reinforcement signal

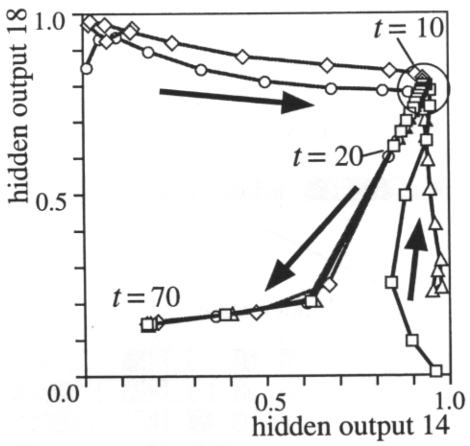

Fig. 7. The change in the states of the two hidden neurons for four cases of input patterns in the same category when the presentation time for the second pattern is fixed at $t = 10$

indicating whether the recognition response is correct or not. The dynamics of a Elman-type recurrent network after learning a context-based attention task was observed, and was found to be almost a fixed-point convergent. When the input patterns cover the input space to some extent, the number of basins becomes equal to the number of categories required for the task, even though the size of the basin varies considerably. The dynamics was rational and adaptive according to the learning conditions.

Acknowledgment Part of this research was supported by Grants-in-Aid for Scientific Research of the Ministry of Education, Culture, Sports, Science and Technology of Japan (No. 13780295).

References

1. Sakaguchi Y (1993) Sensory integration and active perception in tactile perception (in Japanese). J IEICE 76:1222–1227
2. McCallum AK (1996) Learning to use selective attention and short-term memory in a sequential task. From animals to animats 4, The MIT Press, Cambridge, MA, p 315–324
3. Funahashi S, Bruce CJ, Goldman-Rakic PS (1990) Visuospatial coding in primate prefrontal neurons revealed by oculomotor paradigms. J Neurophysiol 63:814–831
4. Elman JL (1990) Finding structure in time. Cogn Sci 14:179–211
5. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagating. Parallel distributed processing, vol 1. MIT Press, Cambridge, p 318–362
6. Zipser D (1991) Recurrent network model of the neural mechanism of short-term memory. Neural Comput 3:179–193
7. Shibata K, Ito K (2000) Formation of attention and associative memory based on recognition learning (in Japanese). IEICE Technical Report (NeuroComputing), NC99-137, p 153–160
8. Shibata K (2001) Formation of attention and associative memory based on reinforcement learning. Proceedings of the ICCAS (International Conf. on Control, Automation and Systems), ICASE, Jeju Island, Korea, p 9–12