Dynamics of a Recurrent Neural Network Acquired through the Learning of a Context-based Attention Task

Katsunari Shibata & Masanori Sugisaka
Dept. of Electrical & Electronics Engineering, Oita University, 700 Dannoharu, Oita 870-1192, Japan.
shibata@cc.oita-u.ac.jp

Abstract

A context-based attention task is employed in this paper. An Elman-type recurrent neural network is utilized to extract and keep the context information, and only the reinforcement signal that indicates whether the answer is correct or not is given. Through this learning, the function of an associative memory is observed in the Elman-type neural network. Adaptive formation of the basins are examined by varying the learning conditions.

Keywords: attention, associative memory, reinforcement learning, recurrent neural network, adaptive basin formation

1 Introduction

Selective Attention is one of the important research items for robots in the real world. There are huge pieces of information, and necessary part of them should be extracted. Some context information is often utilized to suggest which part is necessary. Further, the context information itself should be extracted from the past series of information and be kept until the time when the context information is utilized.

Zipser showed that associative memory, in other words, fixed-point convergence dynamics can be obtained through learning in an Elman-type recurrent network with one hidden unit[2]. The hidden unit becomes equal to one input signal when the other signal is activated, and otherwise the hidden unit keep its value like a flipflop.

The authors showed that when there are more than one input signals and more than one hidden units, the recurrent network forms some basins, and each of the basins corresponds to one of the categories required in the task[3]. The coding of the category is decided through the learning, while in the conventional associative memory, the coding has been decided by the designer.

In this paper, the network dynamics obtained through learning is observed in detail. Furthermore, the result is reported to examine how flexible the basins are formed.

2 Context-Based Attention Task

Fig. 1 shows the attention task and system employed in this paper. One arrow pattern whose direction is one of four corners is presented at first as the left half of Fig. 1. After some while, another pattern, which consists of 4 small sub-patterns, is presented on the same visual sensor as the right half of Fig. 1. Then, the system is required to classify the sub-pattern at the corner where the first presented arrow pattern pointed. The sub-pattern can be classified into one of three categories.

The arrow direction presented at first can be "upper right", "upper left", "lower right" or "lower left". The size of the original arrow image is $7 \times 7 = 49$. As a noise, one pixel value is inverted randomly with the ratio of one-half. The visual sensor consists of $5 \times 5 = 25$ visual cells, and one part of the original arrow image is cropped. So totally $3 \times 3 = 9$ patterns can be presented for each arrow direction if the noise is not added. Considering the noise, $9 \times (25 + 1) = 234$ patterns can be presented for each arrow direction. 5×5 image is put into 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 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 sensor size is 5×5, the sub-patterns are overlapped with each other at the middle row and column. In such areas, the sensor signal is the average value of the overlapped pixels.

There are three output units, each of which is corresponding to each sub-pattern. The 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 in the network is sigmoid function whose value range is -0.5 to 0.5 except for the input layer. When the answer is correct, the system obtains a reward 1.0, otherwise it obtains a penalty -1.0 as a reinforcement signal r. The output corresponding to the answer 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. This

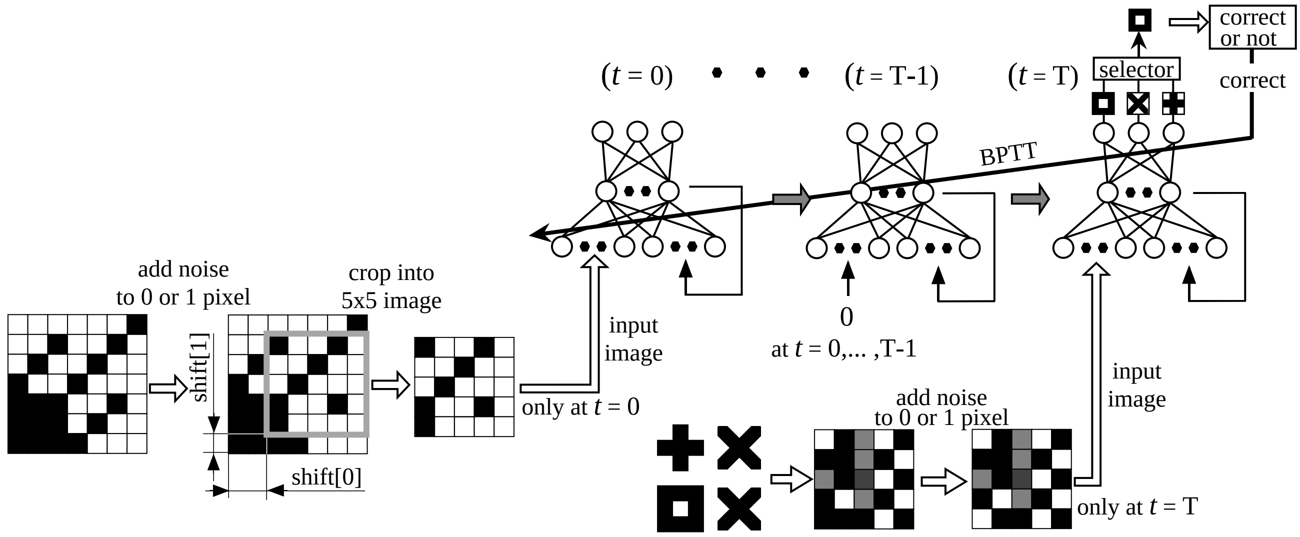

Figure 1: The flow of the context-based Attention task employed in this paper.

means that the system cannot know the correct answer directly when the answer is not correct.

At every time step except $t = 0$ or $t = T$, all the input signals are 0.0. The number of the hidden units is 20, and the values of the hidden units are 0.0 at $t = 0$. The initial weight values are 0.0 for the hidden-output connections, and are decided randomly from -1.0 to 1.0 for the input-hidden 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 around input$=$ 0.0, and the error signal goes backward effectively through time without diverging because $0.25 \times 4.0 = 1.0$.

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. Hebb learning that is often employed for the learning of associative memory cannot realize asymmetrical connections. However, here, no such constraint is given in advance.

3 Simulation Result

Some simulation results after 1000000 trials of learning are shown in this section. The sequence from the presentation of the arrow pattern to the answer and learning is defined as one trial. 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.

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

Fig. 2 shows the change of the standard deviation of each category σ_i and the distance between the centers of two categories d_{ij}. These values are shown for the input pattern at $t = 0$, and the hidden pattern at $t = 0$ and $t = T - 1$ after being normalized by the standard deviation of all the patterns σ to observe the relative relation. 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 at the hidden layer than at the input layer at $t = 0$ for any combinations of the categories. It becomes larger also through time. While, the standard deviation σ_i in one category becomes almost 0.0 at $t = T - 1$ for any categories, and as a result, it is far smaller than the minimum distance between categories $min_{i,j}d_{ij}$. This means that the dynamics of the recurrent network is almost fixed-point convergence and one fixed-point is formed 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 more, the system can generate the correct answer.

Fig. 3 shows the change of the average pattern for each category. It is shown for the input pattern at $t =$ 0, and the hidden pattern 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

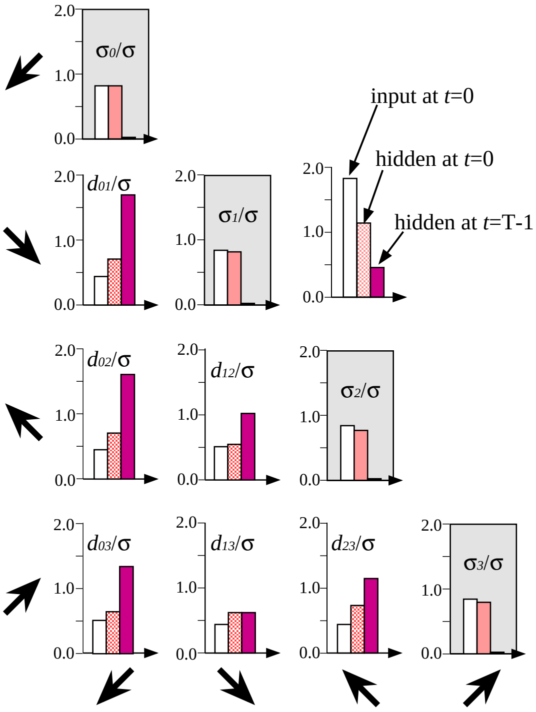

Figure 2: The change of the normalized distances between categories and the standard deviation of one category through time.

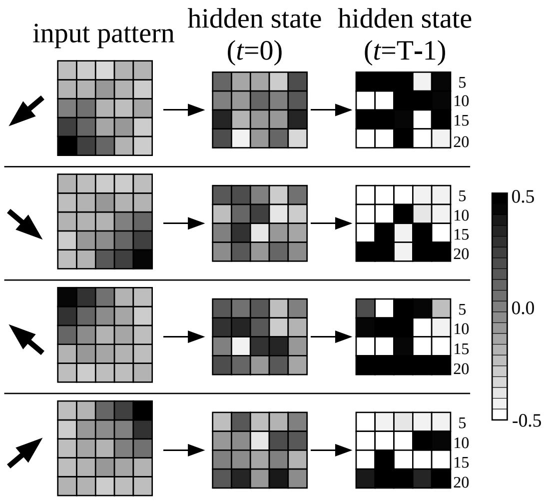

Figure 3: The change of the average pattern of input and hidden layers.

Table 1: The size of the basin for each category in the whole input space.

condition \ category	0	1	2	3	others
normal	1075	3853	1160	3912	0
no_shift (category 2,3)	2039	7527	248	186	0
no_shift (all categories)	841	468	(3528)	4670	493
3 categories	2138	2203	5627	-	32
fixed time (T=10)	1286	(5719)	1306	1679	10

added to one pixel, it is not always 1.0. In the average hidden pattern at $t = 0$, no values are so close to 0.5 or -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 be observed also from this figure.

Fig. 4 shows two examples of the dynamics. In this example, 15 pixels in total 25 pixels have a different value between the two input patterns even though both were generated from the same original arrow pattern. In the hidden patterns at $t = 0$, the values are different more than 0.5 in 15 units among total 20 units. However, as the lower part of the figure, the both hidden states converge to the same hidden state.

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

When the number of the cropping way into 5×5 image is limited to only one in spite of $3 \times 3 = 9$ for two of the 4 categories, the basins change as the second row of Table 1. It is seen that the size of the basin became smaller for the category for which the cropping way was limited. In the other two simulations with different initial connection weights, the 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 normal way.

When the cropping way was limited to only one for all the 4 categories, the learning was faster and more stable, and 8 basins were formed. The dynamics seems complicated in this case. That may be because the basins do not cover the whole input space. The numbers on the third row of Table 1 are counted when $t = 200$ only for this condition. Some of them do not correspond to any of the 4 categories. The reason why the number for the category 2 is put in parenthesis is that the final convergence point is different from the hidden state at $t = 5..14$.

When the number of the categories reduced to three, the basins changes as the fourth row of Table 1. It is seen that the number of large basins is three, and the other small basin was formed. In the other two simulations with a different random number sequence,

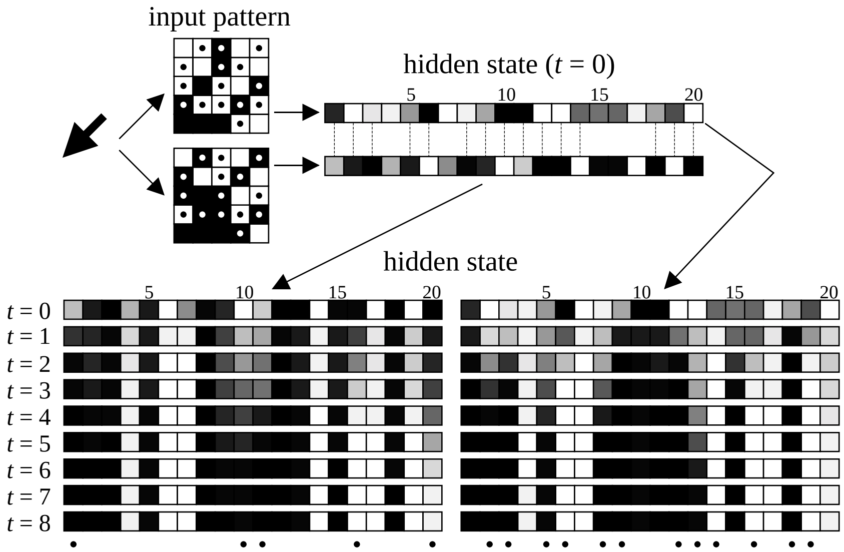

Figure 4: An example of the change of hidden state. The dynamics of fixed-point convergence can be observed.

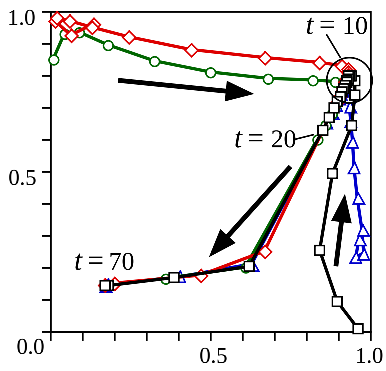

Figure 5: The change of two hidden neuron states when presentation time of the second pattern is fixed at $t = 10$.

the number of the formed basins is just three. It is considered that the the number of the basins becomes equal to the number of categories when the input pattern used in the learning varies in some degree.

When the presenting time of the second pattern is fixed at $t = 10$, the 4 main basins are formed as the fifth row of Table 1, but for the category 1, the hidden state at $t = 10$ is different from the final fixed point. In this case, the values of three hidden neurons changed after $t = 10$. Fig. 5 shows how such neurons change its values. X-axis shows the output of one such neurons, and y-axis the output of another one. Each of four lines in this figure shows the change of hidden neurons output for each of four input patterns in the category 1. It is seen that all the hidden states are the same at $t = 10$, and the speed of hidden state change becomes slow. However, after that, the state changes gradually, and finally converges to the real fixed-point. In the simulation with different initial connection weights, four categories can be distinguished with each other as well, but only 3 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

The dynamics after the learning of a context-based attention task using Elman-type recurrent network has been observed. When the input patterns cover the input space in some degree, the number of the basins becomes equal to the number of categories required in the task, even though the size of the basin is varied so much. The dynamics was rational and adaptive according to the learning conditions.

Acknowledgement

A 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 (#13780295)

References

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

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

[3] Shibata, K., "Formation of Attention and Associative Memory Based on Reinforcement Learning", *Proc. of ICCAS (Int'l Conf. on Control, Automation and Systems*, pp. 9–12, 2001.