Neural Comput & Applic (2003)11:129–136
Ownership and Copyright
© 2003 Springer-Verlag London Limited

Neural Computing
& Applications

Dynamical Responses of Chaotic Memory Dynamics to Weak Input in a Recurrent Neural Network Model

S. Mikami[1] and S. Nara[2]

[1]Information Engineering Major of the Graduate, School of Engineering, Hiroshima University, Higashi-Hiroshima, Japan;
[2]Department of Electrical & Electronic Engineering, Faculty of Engineering, Okayama University, Tsushima-naka, Okayama, Japan

Chaotic dynamics in a recurrent neural network model, in which limit cycle memory attractors are stored, is investigated by means of numerical methods. In particular, we focus on quick and sensitive response characteristics of chaotic memory dynamics to external input, which consists of part of an embedded memory attractor. We have calculated the correlation functions between the firing activities of neurons to understand the dynamical mechanisms of rapid responses. The results of the latter calculation show that quite strong correlations occur very quickly between almost all neurons within 1 ~ 2 updating steps after applying a partial input. They suggest that the existence of dynamical correlations or, in other words, transient correlations in chaos, play a very important role in quick and/or sensitive responses.

Keywords: Chaotic memory dynamics; Correlation function; Instantaneous memory search; Recurrent neural network; Response property

1. Introduction

For several decades, neuroscience and modern technology have been developing very rapidly, and have enabled the quantitative measurement of spacio-temporal brain activities and their analysis [1,2]. In particular, the discovery of chaotic dynamics in the brain had a big impact, and has been attracting a great deal of interest of many scientists. One of the important points is how chaotic dynamics is related to the excellent information processing or control functioning realised in brain. However, it is extremely difficult to understand the functional role of complex dynamics occurring in the brain because of the enormous complexity that originates from dynamics in a system with large but finite degrees of freedom like brain. There have been several pioneering works about brain functioning from the viewpoint of a complex, nonlinear dynamical system, such as Aihara, Tsuda, Fujii et al. [3–5].

One of the typical chaotic dynamics in neural activities of the brain was discovered by Freeman et al. [6] in the brain of a rat. Their experiment gave us motivation to study the dynamical response properties of chaotic dynamics in a neural system to weak external input. Our approach is based on the model and numerical method proposed by Nara and Davis [7–12] from the viewpoint of investigating the functional role of chaotic dynamics. We develop their treatment, and study the quick and sensitive response characteristics of chaotic dynamics to external input; in particular, we focus on the dynamical mechanism by calculating the correlation functions between the firing activities of neurons.

2. Memory Attractors and Chaotic Dynamics

Let us introduce a recurrent neural network model, and define the synchronous updating rule as follows:

Correspondence and offprint requests to: Professor S. Nara, Department of Electrical & Electronic Engineering, Faculty of Engineering, Okayama University, Tsushima-naka 3-1-1, Okayama, 700–8530, Japan. Email: nara@elec.okayama-u.ac.jp

$$S_i(t+1) = \operatorname{sgn}\left(\sum_{j\in G(r)} W_{ij}\, S_j(t) + \alpha\, I_{i\in F(l)}\right) \qquad (1)$$

where $S_i(t) = \pm 1 (i = 1 \ldots N)$ represents the firing state of a neuron specified by index i at time t, and the function $\operatorname{sgn}(x)$ takes 1 (if $x \geq 0$) or -1 (if $x < 0$). W_{ij} is a connection weight (synaptic weight) from the neuron S_j to neuron S_i, where W_{ii} is taken to be 0. $G(r)$ means a connectivity configuration set, where each neuron has r connectivities (randomly located fan-in number for each neuron), and the transmission of signals from the other $(N - r)$ connectivities are assumed to be blocked by a certain inhibitory action. It should be noted that configuration set $G(r)$ indicates spacial configurations of connectivity r, where the numbers of combinations are ${}_N C_r$. $F(l)$ means a set of the partial pixels chosen to apply external input, and the number of the pixels is l ($l \ll N$). In our model, the long-term behaviour of $S(t)$ is determined depending on a given set of connection matrix $\{W_{ij}\}$, and as is well known, an appropriately determined $\{W_{ij}\}$ enables us to make arbitrary chosen state vectors $\{\xi\}$ into multiple stationary states in the time development of $S(t)$, which is equivalent to storing memory states. In our study, W_{ij} are taken as follows:

$$W_{ij} = \sum_{\mu=1}^{L} \sum_{\lambda=1}^{K} (\xi_\mu^{\lambda+1})_i \cdot (\xi_\mu^{\lambda\dagger})_j \qquad (2)$$

where K is the number of states included in a cycle ($\xi_\mu^{K+1} = \xi_\mu^1$) and L is the number of cyclic memories. If connectivity r is large, with $r \simeq N$, the sequences of patterns used to construct the memory matrix are attracting sequences. Therefore, in the absence of input $\{I_i\}$, the network can then function as a conventional associative memory. If $S(t)$ is one of the memory patterns, ξ_μ^λ say, then $S(t + 1)$ will be the next memory pattern in the cycle, $\xi_\mu^{\lambda+1}$. If $S(t)$ is near one of the memory patterns ξ_μ^λ, then the sequence $S(t + kM)$ ($k = 1, 2, 3, \ldots$) generated by the M-step map will converge to the memory pattern ξ_μ^λ. More specifically, for each memory pattern ξ_μ^λ, there is a set of states $B_{\mu\lambda}$, called a memory basin, such that if $S(t)$ is in $B_{\mu\lambda}$, then $S(t + kM)$ ($k = 1, 2, 3, \ldots$) will converge to ξ_μ^λ. In Eq. (2), $\xi_\mu^{\lambda\dagger}$ is the conjugate vector of ξ_μ^λ which satisfies $\xi_\mu^{\lambda\dagger} \cdot \xi_\mu^\lambda = \delta'_{\mu\mu}\delta'_{\lambda\lambda}$, and is introduced to enable us to avoid increasing spurious memories [7].

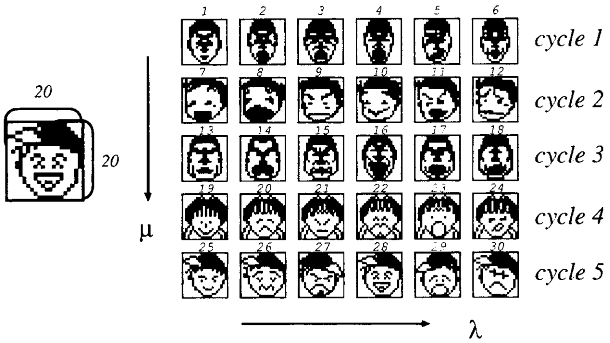

Fig. 1. 30 patterns for memory, where each memory pattern consists of 400 black-and-white pixels, which correspond to $(\xi_\mu^\lambda)_i = 1, -1$, respectively, represented by 20×20 pixel pattern, so that $i = 1 \sim 400$.

In our actual simulation, we employ $N = 400$, $L = 5$, $K = 6$, and the actual memory patterns are shown in Fig. 1, where 30 patterns are classified into five groups, and each group includes six patterns. In Fig. 1, the intra-group patterns have a strong overlap between them, but the inter-group patterns have less overlap, where an overlap is defined by

$$O^{\alpha\beta} = \frac{1}{N}\xi^\alpha \cdot \xi^\beta = \frac{1}{N}\sum_{i=1}^{N} \xi_i^\alpha\, \xi_i^\beta \quad (\alpha, \beta = 1, \ldots, 30) \qquad (3)$$

where, for convenience of description, the suffixes are tentatively changed.

Next, we introduce a certain system parameter and make these multi-stable attractors destabilise. An idea is to reduce the amount of synaptic connectivity r, in the other words, the fan-in number. When r becomes smaller and smaller, each basin volume gradually decreases. Finally, when r reaches some critical connectivity r_c ($r < r_c \cong 30$), each basin vanishes and the attractor becomes unstable. Therefore, if the amount of connectivity r is sufficiently reduced, updated states of the network do not converge to any cycle, even if updated for a long time. Then it can be observed that the network dynamics becomes itinerant in the 400 dimensional state space consisting of the 2^{400} points of a 400-dimensional hyper cube. So we call such a dynamics 'chaotic wandering'. Though dynamics is obviously generated by the perfectly deterministic rule defined in Eq. (1), and never contains any probabilistic property, the resulting network dynamics seems to be very complex and chaotic. It should be noted that configuration sets of $G(r)$ have a strong influence on the dynamical structures of chaos, and a quite different type of dynamics can be generated by different $G(r)$, even for the same initial pattern. These phenomena are typical even if we employ different memory patterns. To investigate the dynamical structure, we have calculated the basin visiting distribution in 5000 updating step intervals. Figure 2 shows the results of $r = 11$, where the vertical axis

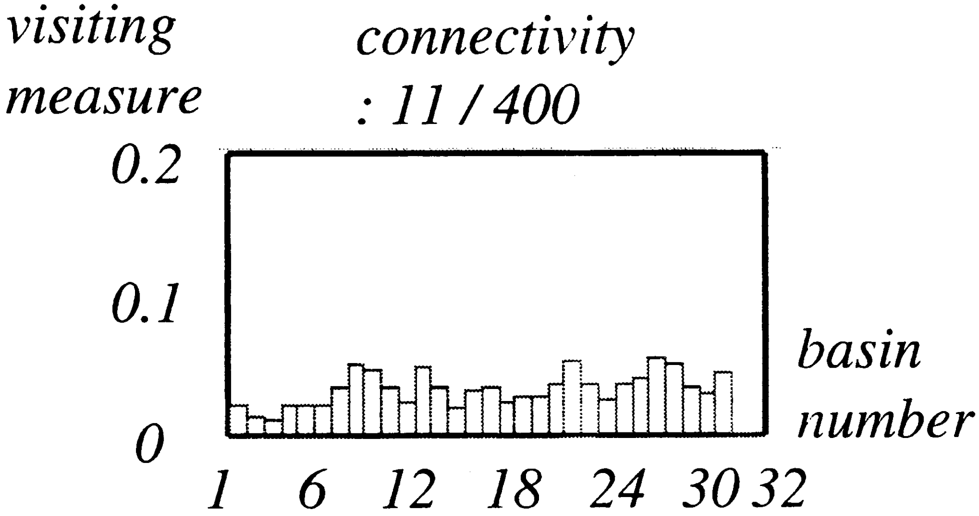

Fig. 2. Basin visiting measure, where the vertical axis is the normalised basin visiting averaged over 5000 updating steps, and the horizontal axis is the basin number.

is the normalised ratio of basin visiting measure and the horizontal axis is the numbering of the attractor basins (1–30), which the state $S(t)$ visits during time averaging of visiting frequencies. The number 31 in Fig. 2 means that $S(t)$ visits a spurious attractor basin, and the number 32 denotes that there is $S(t)$ which visits none of the stored attractor basins or spurious attractor basins. It is omitted to describe the method of determining the passing basins [7–12].

In Fig. 2, it can be observed that the basin visiting measure is distributed to all the basins of attraction, which suggests that the trajectory can pass the whole 400-dimensional state space. Considering this result, when connectivity r is sufficiently decreased, all the memory cycle attractors become unstable, and the network shows a highly developed chaotic dynamics.

In closing, it should be noted that the results obtained in this section are not particularly sensitive to the size of the neuron number N. In the present simulation, we report only the results of taking $N = 400$, but one of the authors and his collaborators has done the simulations for various cases of parameter values, for instance, the number of neurons N (200, 800), the number of embedded limit cycles L, the number of intracycle patterns K. All of the results show that the results obtained in this and later sections do not change so much in the sense of qualitative properties. Therefore, our results can be said to be typical in this type of neural network model, which could be extended to more generic systems having very large but finite degrees of freedom.

3. Sensitive Response Characteristics to Partial External Input

3.1. Outline of the Simulation

In this section, from a functional point of view, we evaluate the sensitivity of response characteristics to partial input. The motivation was partly activated by the work of Skarda and Freeman [6]. They stated that ‘chaotic activities allow rapid and unbiased access to every limit cycle attractor on every inhalation, so that the entire repertoire of leaned discriminanda is available to the animal at all times for instantaneous access. There is no search through a memory store.’ Our model is too simple to relate with or to correspond to their experimental facts of the animal brain, however from an heuristic viewpoint, it could be possible to suggest such a complex function based on a drastically simplified model, including applications to instantaneous access of memory or very quick memory search in an ill-posed setting [7–12].

A very rough description of our simulation is as follows. Keeping an itinerant (chaotic) state by taking a small connectivity (for instance, $r = 8$), once a certain input (consisting of a memory fragment) is applied, the dynamical state can make an almost instantaneous transition to a very near state containing the target memory, even if the dynamical state is wandering and located at any position in the high dimensional state space in the present model. The problem is how we can quantify and evaluate the performance of the rapidness. The important points are as follows:

1. To choose an input pattern as an input signal.
2. To specify the position of the state vector in the state space at the time step just before applying the input.
3. To begin applying the input with a certain strength.
4. To specify the position of the state vector at every time step after the input was applied.

Also, the important points of the simulation setting are

(a) The input should be applied to certain partial neurons, the number of which is one order of magnitude smaller than the total number of neurons.

(b) The input should be a memory fragment of the stored (memorised) patterns.

The following sections describe this process in more detail.

3.2. Very Rapid Reaching to the Target Basin from any Point in the State Space

Let us describe each setting (a), (b) and the algorithm for (2), (3) and (4) stated in the previous section.

- **The setting (a)**: as a sample set of external inputs, we prepare five patterns chosen from each memory cycle shown in Fig. 1.
- **The setting (b)**: each partial pattern consists of $8 \times 5 = 40$ pixels, the number of which is one order of magnitude smaller than the number of total neurons. The position of 40 pixels for each pattern is shown in Fig. 3, where the selected partial patterns have strong features in each intracycle pattern.
- **The algorithm of (2)**: the state space is extraordinarily wide (400-dimensions in the present model), so that, to specify the position of the state vector, $S(t)$ at time t, we use the basins of stored patterns, and represent the position of the state vector by specifying the passing basins, where a basin means that all of the states $\{X\}$ in each basin converge to one of the embedded attractor patterns if one updates the state vectors with the time development rule

$$X_i(t+1) = \mathrm{sgn}\left(\sum_{j=1}^{N} W_{ij}\, X_j(t)\right) \quad (i = 1 \sim N)$$

- Thus, the total state space can be divided into KL subspaces, where in the present simulation, $K = 6$, $L = 5$, so that it is divided into the 30 subspaces. It should be noted that one should take K-step checking in specifying the basins. With this method, we can check what basin $S(t)$ at time t is passing through by specifying the basin number ($1 \sim 30$). This check is done at the time step just before input is applied.
- **The algorithm of (3)**: with setting a strength parameter α of the input, we begin to apply the chosen input at time t as

$$S_i(t+1) = \mathrm{sgn}\left(\sum_{j\in G(r)} W_{ij}\, S_j(t) + \alpha\, I_{i\in F(l)}\right)$$

- **The algorithm of (4)**: we check what basin $S(t{+}1)$ belongs to, with the method of specifying a basin. Then we can recognise that the input makes the state vector $S(t)$ jump to the other position in the state space specified by the basin number ($1 \sim 30$).

If the transition to the target basin, which contains the input, occurrs, then the input makes the dynamical state instantaniously jump to the near state of the memorised patterns. Actual simulations show that, in an averaged sense, it takes a few steps until it reaches the target basin. We have repeated the simulation for several patterns of input, changing the connectivities r : ($8 \sim 14$), and changing the spacial configuration of connectivity $G(r)$, and the strength of input α ($15 \sim 65$). Let us show an example, which gives one of the best performances in our simulation (Fig. 4).

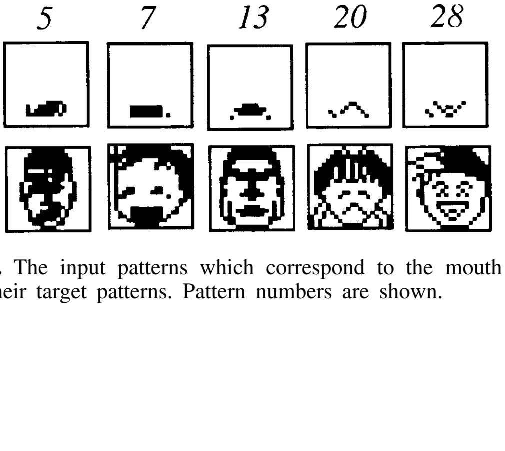

Fig. 3. The input patterns which correspond to the mouth part and their target patterns. Pattern numbers are shown.

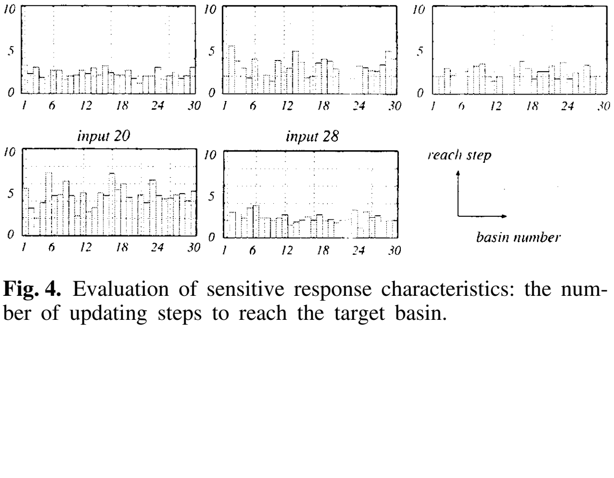

Fig. 4. Evaluation of sensitive response characteristics: the number of updating steps to reach the target basin.

Observing each histogram, it can be said that the step numbers necessary to reach the target basin after input is applied is, in an averaged sense, almost equal. In other words, from any position in the state space, chaotic dynamics can reach every target basin within a few steps after input is applied. The averaging is 2.93 steps in the case of Fig. 4. The above result does not change so much within the given region of parameter $r(8 \sim 14)$, but considerably depends on spatial configuration of connectivity r. To confirm this more quantitatively, we have calculated the number of steps necessary for 200 samples of different spacial configurations of $G(r = 8)$, where the input strength is always kept to be $\alpha = 35$. Figure 5 shows the averaged number of steps as a function of s_u (1–10), where s_u means the upper-bound step to check the passing memory basins after the input was applied, and we show only two cases: '*chaos*(s1)' represents the best case among the 200 samples of $G(r = 8)$; and '*chaos*(f1)' represents the worst case. For comparison, we have evaluated the same performance with the use of a complete random walk instead of using chaotic dynamics. The result is shown as '*random*' in Fig 5. Comparing the three cases in Fig. 5, the performances of averaged reaching steps are considerably different between *chaos*(s1) and *chaos*(f1). It should be noted that the performance of *chaos*(s1) is far superior to that of *random*.

3.3. Performance Evaluation in Terms of Success Rate

Let us show the evaluation of the sensitive response characteristics in terms of the success rate to reach the target basin, with the same condition in the previous sections. To obtain the evaluation of difference between 'effective chaos (*chaos*(s1)' and 'less effective chaos (*chaos*(f1)' shown in Fig 5. which would be dependent on the spatial configuration of connectivity r, we report only the results of calculating the success rate for the two configurations used in Fig 5. Now, Fig. 6 shows the success rate of reaching a target memory basin as a function of s_u, where s_u means the upper-bound step to check the passing memory basins after the input was applied. The success rate is the average over 200 samples, where each data is taken at the different locations of wandering in chaotic memory dynamics. It can be clearly observed that the spatial configuration of $G(r = 8)$, which shows the best performance in averaged reaching steps, also gives a high performance in the success rate. Comparing the two success rates of *chaos*(s1) and *random*, the performance obtained with the use of chaotic memory dynamics is far superior to the performance of the random walk, particularly at $s_u \cong 3, 4$, so it can be said that depending upon the choice of spatial configuration of $G(r)$, chaotic dynamics is far superior to random walk with respect to the sensitive response characteristics.

The data in Fig. 7 support the above statement, where Fig. 7(a) indicates the basin visiting measure of *chaos*(s1) constrained by external input, and Fig. 7(b) indicates the cross-section of memory basin volume. The cross-section of memory basin volume means the subspace volume measure of a memory basin evaluated under the condition that the states of the 40 neurons where input is applied are fixed to be the same state with input (memory fragment). Comparing Figs 7(a), (b) for each target pattern, we can clearly recognise that the case in Fig. 7(a) (chaotic dynamics of the effective *chaos*(s1)) has

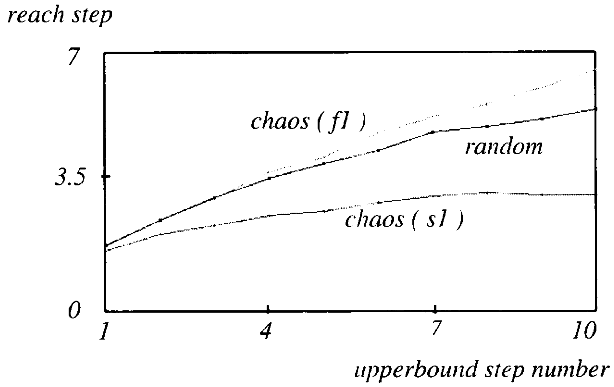

Fig. 5. Evaluation of the sensitivity response characteristics: the averaged reach step number as the upper bound step s_u changes from 1 to 10, where the upper bound s_u means the constraint that the updated state $S(t)$ after the input is applied can reach the target basin within the updating step number s_u.

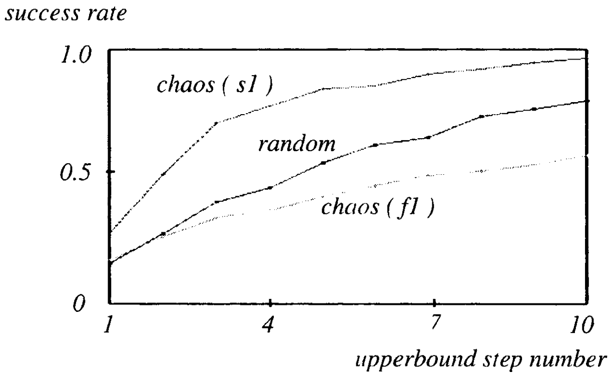

Fig. 6. Evaluation of sensitive response characteristics: the success rate of reaching the target basin as a function of the upper bound step s_u from 1 to 10 (about s_u, see the caption of the previous figure).

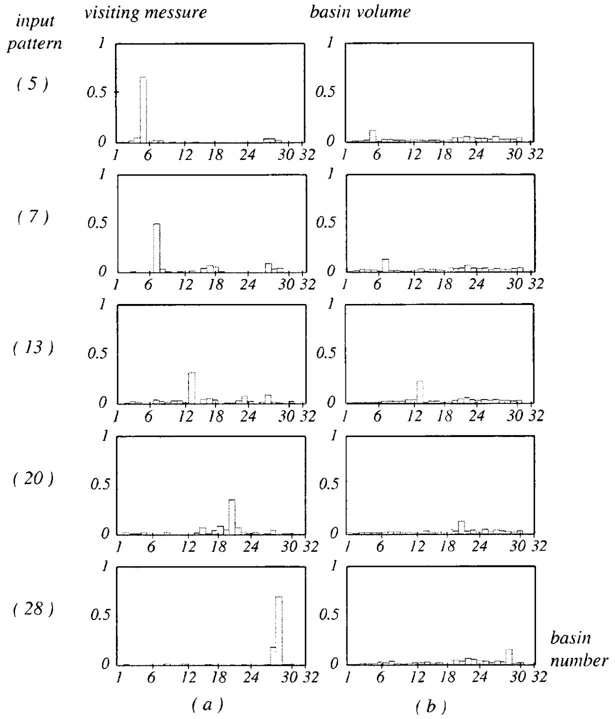

Fig. 7. The basin visiting measure constrained with external input (left) and the cross-section of basin volume (right).

the stronger localisation in the memory basin to which the input pattern belongs.

Next, in Fig. 8 we show the constrained basin visiting measure of *chaos*(*f*1) under the existence of external input. In contrast to Fig. 7(a), this figure obviously shows that the dynamical trajectories of the less effective *chaos*(*f*1) are *not* localising in the memory basin to which the input belongs. Considering these results, the performance of sensitive response characteristics strongly depends upon which direction the effect of external input makes the trajectories shift, to the target basin or the other basins. We have confirmed by further simulation that the discussion so far is true in the following cases. Connectivity *r*: 8, 10, 12, 14; external input strength α: 15, 25, 35, 45, 55, 65; part of external input $I_{i \in F(l)}$: 40 pixels corresponding to the 'eye' part and the 'mouth' part in the face patterns; the number of samples: 200, 400, 600. Therefore, our results could be generic in the present chaotic system.

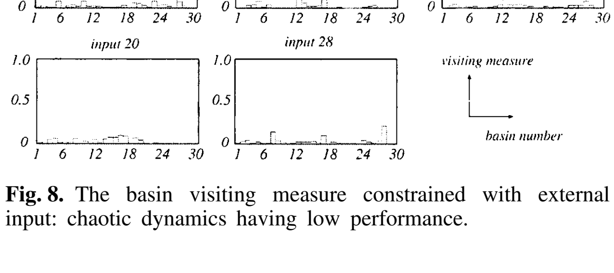

Fig. 8. The basin visiting measure constrained with external input: chaotic dynamics having low performance.

4. Almost Instantaneous Emergence of Correlation between Firing Activities of Neurons

Based on the results in the previous section, we further investigated the dynamical structures that give sensitive response characteristics occurring just after applying input. In more detail, we have evaluated the time-dependent correlation functions between the firing of neurons S_i and S_j. First, we define the correlation function $F_{i,j}(t',t)$ as follows:

$$\begin{aligned} F_{i,j}(t', t) &= \langle S_i(\tau + t')\, S_j(\tau + t) \rangle_\tau \\ &= \frac{1}{T}\int_{\kappa}^{\kappa+T} S_i(\tau + t')\, S_j(\tau + t) d\tau \\ &= \frac{1}{T}\sum_{\tau=\kappa}^{\tau=\kappa+T} S_i(\tau + t')\, S_j(\tau + t) \qquad (4) \end{aligned}$$

where *i,j* indicates neurons S_i, S_j and t', t, the two time points when the correlations of the firing S_i, S_j are taken, respectively. $\langle f(\tau) \rangle_\tau$ means the time averaging of $f(\tau)$ over $\kappa \leq \tau \leq \kappa + T$. Let us note the meaning of $F_{i,j}(t', t)$. Specifying the parameters i, j, t', t, the correlation function $F_{i,j}(t', t)$ represents the degree of spatio-temporal firing synchronisation (the strength of dynamical correlation) between the two neurons $S_i(t')$ and $S_j(t)$. Thus, $F_{i,j}\ (t', t) \simeq \pm 1$ means that two neurons $S_i(t')$ and $S_j(t)$ keep quite synchronised 'in-phase' or 'anti-phase' firing over the time interval T. Decreasing $|F_{i,j}(t', t)|$ means that the degree of synchronisation between $S_i(t')$ and $S_j\ (t)$ becomes weaker and weaker. When $|F_{i,j}(t', t)|$ vanishes, the correlation of firing between $S_i(t')$ and $S_j(t)$ also vanishes, and these two firings look completely random. In the following calculations of the spatio-temporal correlation functions $F_{i,j}(t', t)$ with changing the parameters i, j, t', t, let us specify two kinds of cases: an auto correlation function ($i = j$), and a pair correlation function ($i \neq j$). In our actual simulation, the notation $F_{i,j}(t)$ is used, omitting t' in the original definition, since we always keep $t' = 0$.

The important idea is that partial input ('memory fragment') applied to neurons one order of magnitude smaller than the total number of neurons could result in a strong correlation between many neurons almost instantaneously through dynamical correlation between neurons existing in chaotic dynamics. The outline of our simulation is as follows. As a trial set of external inputs ('memory fragments), we employ the same patterns as shown in Fig. 3, and we take the parameters $i = 286$, $t' = 0$, $\tau = 1000$, respectively. The reason why we choose $i = 286$ as the neuron S_i in the calculation of $F_{ij}(t)$ is that S_{286} is one of the neurons belonging to the input set $I_{i \in F(l)}$, so that $F_{i,j}(t)$ directly tells us the degree of correlation with the input. Let us briefly describe the method of numerical evaluation of correlation functions:

1. Preparing an initial random pattern $S(0)$ using a random number generator.
2. Choosing a certain small value of connectivity r, we drive chaotic dynamics.
3. Choosing a certain value of input strength α, we begin to apply input (memory fragment) and update the network for 30 steps with external input, where the updating rule is given by Eq. (1). During updating, we record the firing states of all neurons.
4. Repeat the procedure 3 for the 1000 samples, keeping the same value of connectivity r and the input strength α, where the initial states are all different in the 1000 samples.

Using the recorded state of all neurons, we evaluate the correlation function $F_{286,j}(t)$ over neurons $1 \leq j \leq 400$ and over updating time $0 \leq t \leq 30$. Figure 9 indicates the absolute values of pair correlation function $|F_{i,j}(t)|$ evaluated by the above procedures, where $1 \leq j \leq 400$ and $0 \leq t \leq 9$. Figure 9(a) shows the correlation function without external input and Fig. 9(b) is with external input $\alpha = 35$, where we apply the external input consisting of a partial pattern belonging to the attractor number 28 at $t = 0$. In Fig. 9(a), one can obviously understand that the spatio-temporal correlations $|F_{i,j}(t)| \cong 0$ almost everywhere, since the firing dynamics is becoming sufficiently chaotic. On the other hand, Fig. 9(b) indicates that the quite strong correlations are occurring everywhere within 1, 2 updating steps after applying input.

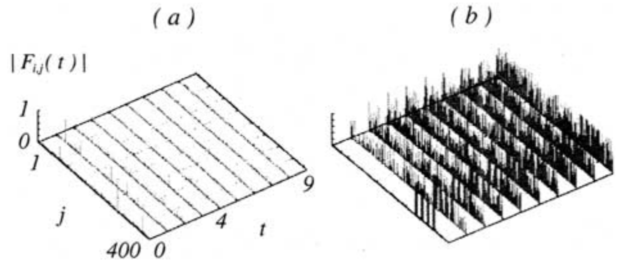

Fig. 9. Spatio-temporal dependent correlation functions just after applying input: (a) without external input, (b) with external input $\alpha = 35$.

These results can also be confirmed by taking the distributions of $|F_{i,j}(t)|$ just after applying input in Fig. 10. In this figure, t is the same as in Fig. 9 and $F_{i,j}(t)$, $\rho(|F_{i,j}(t)|)$ indicate the value of the pair correlation function and its distribution, respectively. It can be observed that the influence of partial input extends rapidly into all neurons. Considering these results, it can be said that the dynamical effect of the partial input transmits through all neurons within 1, 2 updating steps.

5. Summary and Concluding Remarks

Let us summarise the results obtained by our computer experiments, and give concluding remarks about the following two points.

We have evaluated the sensitive response characteristic of chaotic memory dynamics to partial external input (memory fragment). The averaged reaching steps to the target basin with respect to each input pattern are almost equal. The sensitive response characteristics of chaotic dynamics vary considerably, depending upon spatial configuration $G(r)$. The wandering (chaotic) dynamics, which shows a high performance of averaged reaching steps, also shows a high success rate. It can be said that, depending upon the choice of spatial configuration $G(r)$, chaotic dynamics is far superior to random walk with respect to the sensitive response characteristics having a high success rate. The performance of sensitive response characteristics is proportional to the localisation measure to the memory basin to which the input pattern (memory fragment) belongs. We have confirmed that the performance depends considerably upon the spatial configuration of $G(r)$. This indicates that $G(r)$ (spatial configuration of small connectivity) changes the localisation of the basin visiting measure.

We have evaluated the correlation functions between firing neurons, and have investigated how partial input transmits into the network. The results

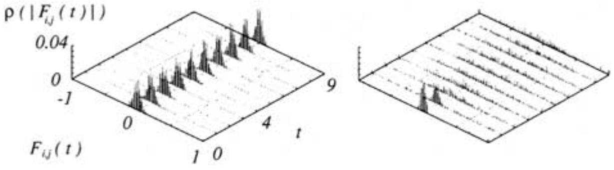

Fig. 10. Distributions of the absolute value of correlation functions just after applying input: without external input (left), with external input $\alpha = 35$ (right).

show that the quite strong correlations $|F_{i,j}(t| \cong 1$ occur everywhere within 1–2 updating steps after applying input. The signal from the partial input can reach any neuron within two steps of updating.

Acknowledgements. The authors thank their colleagues in the 'Dynamic Brain Group' (Prof. Fujii, Prof. Tsukada, Prof. Tsuda and Prof. Aihara) for their valuable comments on this work. This work has been supported partly by a Grant-in-Aid from the Ministry of Education, Science, Sports and Culture of the Japanese government, and partly by CREST (Core Research for Evolutional Science and Technology) of the Japan Science and Technology Corporation (JST).

References

1. Anderson JA, Rosenfeld E (Eds). Neurocomputing, MIT Press, 1988
2. Anderson JA, Rosenfeld E (Eds), Neurocomputing 2, MIT Press, 1990
3. Aihara K, Takabe T, Toyoda M. Chaotic neural networks. Phys Lett A 1990; 114: 333–340
4. Tsuda I. Chaotic itinerancy as a dynamical basis of hermeneutics of brain and mind. World Futures 1991; 32: 167–185, (also see the most recent publishing and the references therein; Kaneko K, Tsuda I, Complex systems: chaos and beyond, Springer-Verlag, 2000)
5. Fujii H, Itoh H, Ichinose N, Tsukada M. Dynamical cell assembly hypothesis – Theoretical possibility of spatio-temporal coding in the cortex. Neural Networks 1996; 1303–1350
6. Skarda CA, Freeman WJ. How brains make chaos in order to make sense of the world. Behav Brain Sci 1987; 10: 161–195
7. Mori T, Davis P, Nara S. Pattern retrieval in an asymmetric neural network with embedded limit cycles. J Phys A 1989; 22: L525–L532
8. Nara S, Davis P. Chaotic wandering and search in a cycle-memory neural network. Progress Theor Phys 1992; 88: 845–855
9. Nara S, Davis P Totsuji H. Memory search using complex dynamics in a recurrent neural network model. Neural Networks 1993; 6: 963–973
10. Nara S, Davis P, Kawachi M, Totsuji H. Chaotic memory dynamics in a recurrent neural network with cycle memories embedded by pseudo-inverse method. Int J Bifurcation Chaos 1995; 5: 1205–1212
11. Nara S, Davis P. Learning feature constraints in a chaotic neural memory. Phys Rev E 1997; 55: 826–830
12. Kuroiwa J, Nara S, Aihara K. Functional possibility of chaotic behaviors in a single chaotic neuron model for dynamical signal processing elements. Proceedings IEEE Int Conf SMC 1999; 1: 290–295
13. Peinke J, Parisi J, Rössler OE, Stoop R. Encounter with chaos, Springer-Verlag, 1992

Mathematical Symbols

$S_i(t)$:	the state of the ith neuron at time t.
$S(t)$:	state vector of the present neural network.
W_{ij}:	a synamptic connection matrix.
$G(r)$:	a special configuration set of a given connectivity r.
r:	a connectivity.
r_c:	the critical connectivity which gives chaotic dynamics begins.
α:	an input strength.
$I_{i \in F(l):}$	an external input which is applied to a certain set of partial neurons specified by $F(l)$.
sgn(x):	the signature function.
ξ_μ^λ:	a specified memory pattern.
$\xi_\mu^{\lambda\dagger}$:	a conjugate vector of ξ_μ^λ.
$O^{\alpha\beta}$:	an overlap matrix.
N:	the total number of neurons.
L:	the period length of a cyclic memory.
K:	the number of cycles.
$B_{\mu\lambda}$:	an attractor basin.
P_l, p_i:	a basin number.
s_u:	an upper bound of updating step number.
$F_{i,j}(t', t)$:	a correlation function between the ith neuron and the jth neuron, at time t' and t, respectively.
$\rho(\|F_{i,j}\ (t)\|$:	a distribution function of the values of correlation functions.