



HAL
open science

Structure and dynamics of random recurrent neural networks

Hugues Berry, Mathias Quoy

► **To cite this version:**

Hugues Berry, Mathias Quoy. Structure and dynamics of random recurrent neural networks. *Adaptive Behavior*, 2006, 14, pp.129-137. inria-00001065

HAL Id: inria-00001065

<https://inria.hal.science/inria-00001065v1>

Submitted on 29 Jan 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Structure and dynamics of random recurrent neural networks

Hugues Berry (1) and Mathias Quoy (2)

(1) INRIA Futurs - 4 rue J. Monod 91893 Orsay, France

(2) ETIS UMR8051 - IUT de Cergy-Pontoise - ENSEA, 6, av. du ponceau, 95014
Cergy-Pontoise, France

Abstract. In contradiction with Hopfield-like networks, random recurrent neural networks (RRNN), where the couplings are random, exhibit complex dynamics (limit cycles, chaos). It is possible to store information in these networks through hebbian learning. Eventually, learning “destroys” the dynamics and leads to a fixed point attractor. We investigate here the structural change in the networks through learning, and show a “small-world” effect.

1 Introduction

Using random recurrent neural networks is part of the dynamical systems approach to the simulation of some cognitive functions. The start of our work is the understanding of Freeman’s experiments on the olfactory bulb, where the dimension of the dynamics attractor reduces on a more simple attractor (limit cycle) when a known odor is recognized [12,16,23]. We have been able to replicate these findings on random recurrent neural networks (RRNN) with a classical hebbian learning rule [7]. Following Amari [1], and using a mean-field theory, we have also been able to theoretically study the dynamics of the system depending on the various parameters [5]. Using a modified hebbian learning rule, we have shown that this kind of network is able to store and retrieve complex spatial (static) and temporal (sequences) patterns [8].

Parallel to this approach, recent studies have focused on the topological structure of large networks using graph theory approaches. They have proven successful in understanding the global properties of several complex systems originating from highly disparate fields, from the biological to social and technological domain. Hence the same (or similar) reasonings can be applied to understand cell metabolism [13], the citation of scientific articles [14], software architecture [20], the Internet [2] or electronic circuits [4]. The most common statistical structures are the so-called *small-world* and *scale-free* networks. Small-world properties characterize networks with both small average shortest path and a large degree of clustering, while scale-free networks are defined by a connectivity probability distribution that decreases as a power law (see section 5 for more formal definitions). At a much coarser grain, graph theory methods have recently been applied to networks of cortical areas [18,11] i.e., not networks of neurons but

networks of neuron *areas*, with the prospect of understanding the network functions.

We propose here to use a dynamical system and a graph theory approach in order to understand the dynamical and structural changes through learning in our model. So, in the following, we first present our model (2), then show some typical dynamical behaviors (3), before introducing the learning rule (4) and study the structure of the network before and after learning (5). Finally we conclude (6).

2 Model

A random recurrent neural network is a set of N fully connected neurons. The connection weights are randomly drawn according to a Gaussian law $\mathcal{N}(0, J^2/N)$, where J is the standard deviation (chosen here to be 1). The neuron states are $\mathbf{x}(t)$. The state dynamics is given by the following set of discrete time recurrent equations:

$$\forall t \geq 0, \quad x_i(t+1) = f\left(\sum_{j=1}^N J_{ij}x_j(t) + I_i(t) - \theta_i\right) \quad (1)$$

f is a sigmoidal function with slope g in 0. The thresholds θ_i are randomly chosen according to a Gaussian law $\mathcal{N}(\bar{\theta}, \sigma_\theta^2)$. $\mathbf{I}(t)$ is a N dimensional input vector. Hence the parameters of the system are: $g, J, \bar{\theta}, \sigma_\theta$ and $\mathbf{I}(t)$. J is set to 1, and $\mathbf{I}(t), \bar{\theta}, \sigma_\theta$ to 0.

3 Dynamics

Depending on the parameter g (J being set to 1), various dynamical behaviors may arise. First, when g is low, the system exhibits a unique fixed point. When g is increased, bifurcations occur leading to a limit cycle (see fig. 1), a torus (see fig. 2), frequency locking and finally to chaos (see fig. 3). This corresponds to a quasi-periodicity route to chaos [10]. Observation of the network activity is performed through the mean activity of all neurons:

$$m_{net}(t) = \frac{1}{N} \sum_{i=1}^N x_i(t) \quad (2)$$

Note that not all neurons behave the same way. Some neurons are either damped near 0, or saturated near 1. Some others really oscillate in the range from 0 to 1. We will call in the following the saturated and the oscillating neurons *active neurons or active population*. This set of active neurons is precisely the part of the networks that sustains the dynamics. They are sequentially activated, so that there is a small cluster of neurons activated at each time step. This set of neurons is different for different inputs [7].

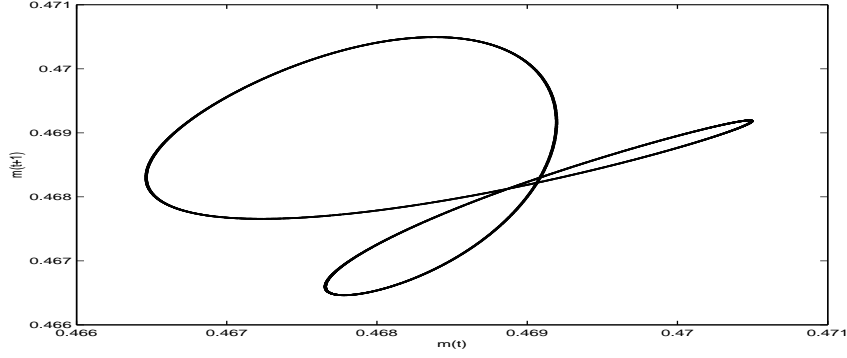


Fig. 1. Limit cycle ($g = 6, N = 100$) represented in phase space ($m_{net}(t + 1)$ versus $m_{net}(t)$). See text for explanations on m_{net} .

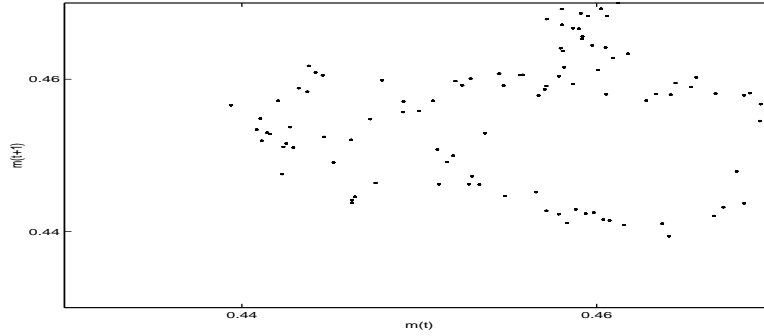


Fig. 2. T2 Torus ($g = 6.85, N = 100$). See figure 1 for explanations.

4 Learning rule

Learning is performed using the following hebbian rule:

$$\begin{aligned}
 &\forall t \geq 1, \\
 &\text{if } x_j(t) \geq 0.5 \text{ then} \\
 &J_{ij}(t + 1) = J_{ij}(t) + \alpha \cdot x_i(t + 1) \cdot x_j(t)
 \end{aligned} \tag{3}$$

In addition a weight may not change its sign (it stays positive or negative). It is easily seen from this learning rule that only weights between active neurons will change. More complicated learning rule may be used [7], but this one is the most simple and is enough for the study we want to perform.

Learning an input modifies the weights, and therefore the dynamics of the system. Hence, during learning, we observe an inverse quasi-periodicity route to chaos, eg. from chaos to a fixed point. If learning is stopped when the dynamics

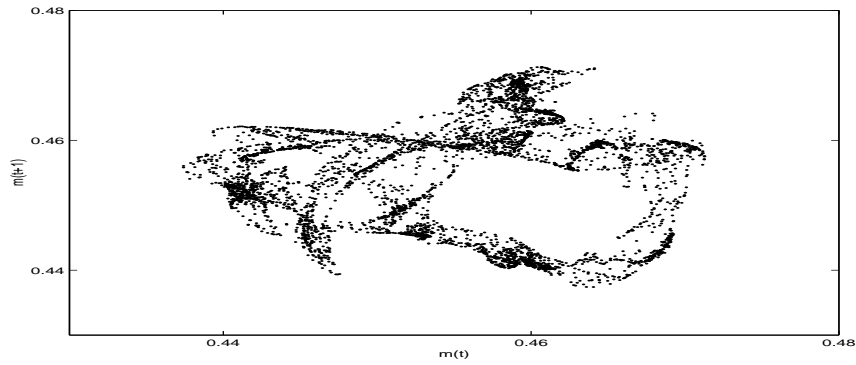


Fig. 3. Chaos ($g = 7, N = 100$). See figure 1 for explanations.

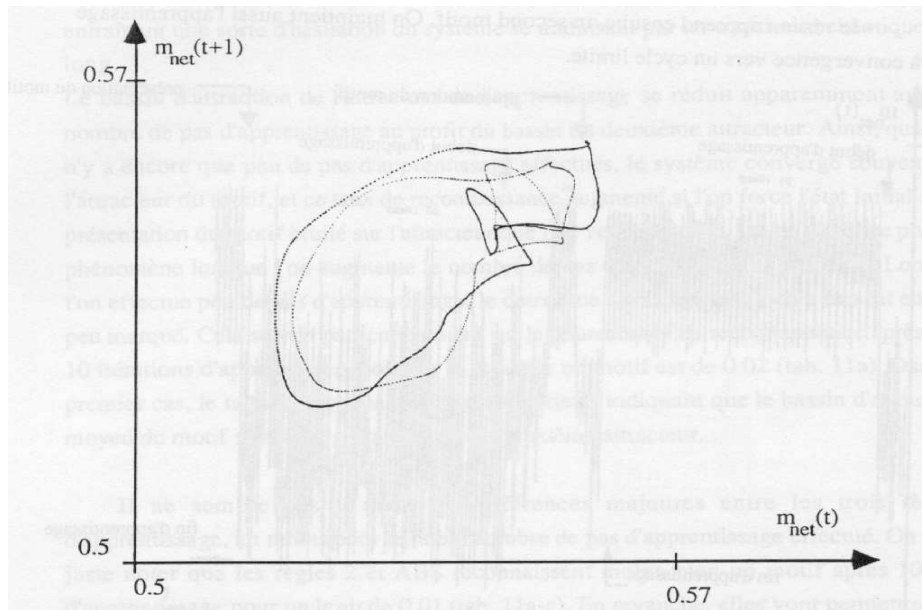


Fig. 4. The attractor linked with the same input \mathbf{I}^1 after learning of \mathbf{I}^1 alone (light line), and after learning of a different input \mathbf{I}^2 (darker line). The attractor is slightly different because the weights have changed through learning. However, the same group of neurons are active [7].

settles on a limit cycle, this cycle is specific of the learned input. Namely, the dynamics without input is chaotic, and presenting the input reduces the dynamics on the corresponding limit cycle. This procedure is similar to the results observed by Freeman [12].

Each input vector leads to a different attractor because an input is similar to a

threshold, and thus may be seen as a *parameter* of the system. When learning an other input, the weights are again modified. Thus the attractor corresponding to the first input is modified. However it stays similar in terms of frequency, rotation number or center. But more important, almost the same neurons are activated (see fig. 4) [7]. Hence, applying a learning rule relating these active neurons to a decision neuron on an other layer will lead to the activation of the same decision neuron. We use this property for the motor command of a mobile robot [8].

Note that unlike Hopfield networks, we do not have several distinct attractors at the same time. For each set of parameters, we have only *one* attractor on which any initial condition converges. Because inputs are parameters, there are different attractors for different inputs. But in this case, an input is *not* the initial condition $x(0)$ (which is chosen randomly). Thus our system is different from others where the chaotic dynamics is the exploration dynamics of several different attractors of the *same* dynamical system like chaotic itinerancy for instance [19].

5 Learning and structure

Because the law of the weights is changing through learning, we have not yet been able to theoretically describe why the dynamics is reducing when learning. However, simulations and statistics on the weight matrix give us some insights. We have studied the structure of the network before and after learning with graph theory tools [3]. The results are reported in the following.

Let $\mathcal{W} = \{w_{ij}\}$ be the weight matrix and $\mathcal{A} = \{a_{ij}\}$ its corresponding (possibly thresholded) adjacency matrix, i.e.

$$a_{ij} = \Theta(w_{ij} - \epsilon) \quad (4)$$

where ϵ is the threshold and $\Theta(\dots)$ the Heavyside step function. Note that we are interested here in the strength of the connection between two neurons, regardless of their inhibitory/excitatory nature. Thus we restrain our analysis to positive weights, retaining only their absolute values, i.e. $w_{ij} \leftarrow |w_{ij}|$.

Because we deal with undirected graphs, incoming and outgoing connections are distinguished. Thus, we should ideally study each statistical indicator in triplicate: one concerning outgoing links only, one for incoming links only, and one dealing with the total links (incoming + outgoing). In this case, dealing with multiple statistical indicators becomes rapidly difficult to handle. We thus chose a tradeoff solution, considering the graph as almost undirected most of the times, while taking directionality into account, especially concerning the clustering index.

We define the global weight $w_{i,j}$ as the maximal value of the incoming and outgoing weights:

$$w_{i,j} = \max(w_{ij}, w_{ji}) \quad (5)$$

Likewise, we note

$$a_{i,j} = \max(a_{ij}, a_{ji}) \quad (6)$$

We then define the degree (connectivity) of node i as:

$$k_i = \sum_{j=1}^N a_{i,j} \quad (7)$$

The average clustering coefficient C expresses the probability that two nodes connected to a third one are also connected together (degree of cliquishness). The clustering index C_i of node number i is:

$$C_i = \frac{1}{k_i(k_i - 1)} \sum_{j,h} a_{i,j} a_{i,h} (a_{jh} + a_{hj}) \quad (8)$$

With this definition at hand, the clustering index takes link directionality (reciprocal links) into account, but only when reciprocal links are between i 's neighbors (see Fig 5). Reciprocal links linking i to its neighbors are thus not explicitly accounted for in C_i . This information can be found in the average density of reciprocal links.

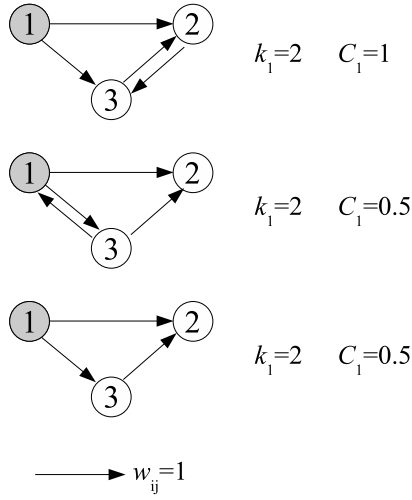


Fig. 5.

We then classically define the clustering index C , as C_i average over the network

$$C = N^{-1} \sum_{i=1}^N C_i \quad (9)$$

Let $d(i, j)$ be the shortest path (in number of neurons) between neuron i and j , then the mean shortest path (MSP) is its average over the network

$$MSP = 1 / (N^2 - N) \sum_{i,j} d(i, j) \quad (10)$$

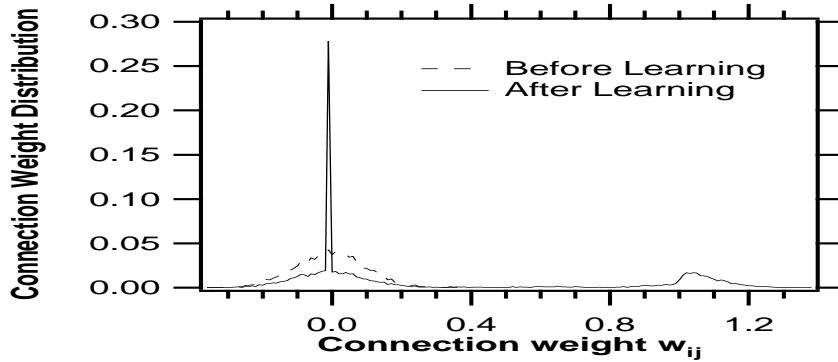


Fig. 6. Weight distribution before and after learning.

The weight distribution before and after learning is displayed on figure 6. Initially, the distribution follows the Gaussian law described above (section 3). After learning there is a clear distinction between weights around 0, and weights around 1. Weights around 1 are those corresponding to learned connections linking together active neurons (see section 4). Weights around 0 are corresponding to negative weights that are bounded by the condition that a weight may not change its sign. However the peak near 1 was not that much expected because the positive weights distribution could have been almost uniform. Setting properly the threshold ϵ to obtain the nonrecurrent adjacency matrix from the weight matrix, enables to gradually isolate the active neuron network from the inactive part.

Using increasing thresholds (thus gradually isolating the active neuron network) We observe a slight increase (less than 10%) of the MSP after learning (see fig. 7). This indicates that the average degree of separation between two neurons of the active population increases only slightly. Thus the random structure is largely maintained in the active neuron network.

However, the average clustering index increases up to 50% (as compared to purely random network see fig. 8). This denotes that after learning (and thresholding) the active neurons are more connected to each other than to other neurons. Further measurements, such as the clustering index probability distribution or the connectivity correlation function [21] have also been computed and will be reported elsewhere.

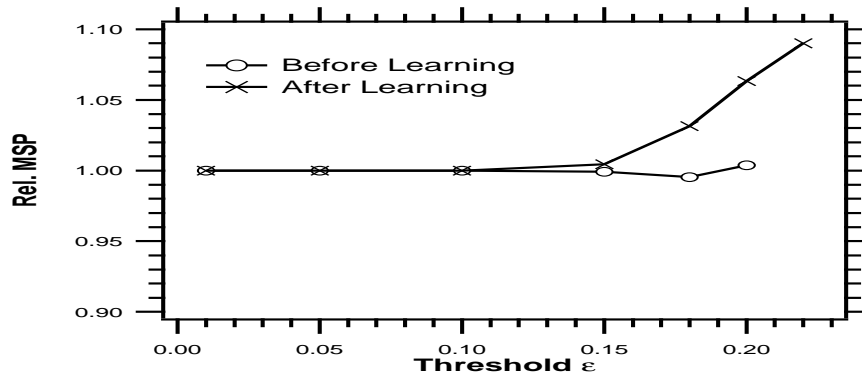


Fig. 7. Mean shortest path versus threshold ϵ . Values are normalized by those observed on a random network with same number of neurons and connections.

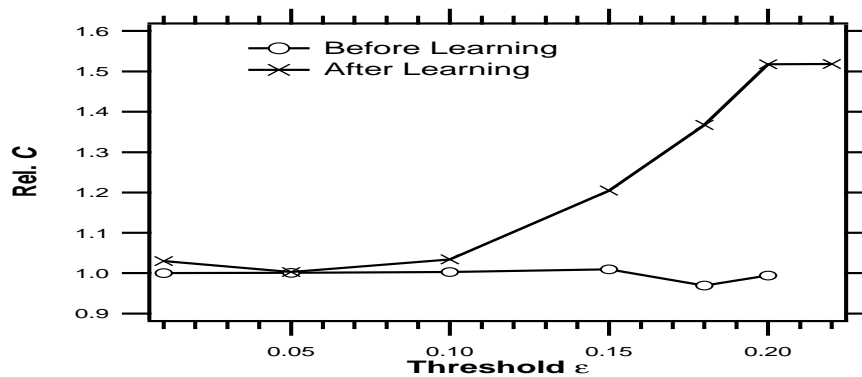


Fig. 8. Average clustering index versus threshold ϵ . Values are normalized by those observed on a random network with same number of neurons and connections.

The two previous properties (high clustering and short distances) are the signature of a small-world network [22] also encountered in many complex networks: cell metabolism [13], the citation of scientific articles [14], software architecture [20], the Internet [2] or electronic circuits [4]. They mean that the network of active neurons after learning is not totally random anymore. It attracted much heavier weights and these highly weighted connections have spontaneously organized into a small-world network.

6 Conclusion

We show that beginning with a randomly connected neural network, and running a hebbian learning rule produces the following behaviors:

- the dynamics reduces from chaos to a limit cycle and finally a fixed point

- the set of “active” neurons is reinforced, and particular for every input
- the resulting graph structure is looking like a small-world one

However the effects observed on the graph structure are obtained after hundreds of learning steps, whereas only a few dozen are enough to reduce the dynamics and recognize a learned pattern [7]. This means first that the observed structure after learning may only be the result of a long-term learning shaping the global structure of the network. Second, this also means that we have to take into account the dynamics of the network and the “dynamic topology” (as opposed to the structural topology). This dynamic topology makes a relation between neurons responding to the same input. This response may be very particular to the frequency, and does not depend on the existence of a direct link between the two neurons [6]. Thus, one further step will be the investigation of the influence of learning on the ability to transmit a signal from one neuron to the other one.

The graph study has only been performed after learning one input. It is now necessary to investigate this structure after learning several inputs. We expect to see the same kind of small-world structure.

Our system is a discrete time one. We can expand it into two directions. The first one is dealing with “spikes” rather than firing rates [9]. There is no qualitative change compared with the properties exposed here. The other direction is introducing a continuous time dynamics instead of a discrete time one. This is a very important issue for at least two reasons. First, though synchronizing processes occur, the natural brain does not compute with fixed discrete time steps. There are some results concerning chaos in the same kind of dynamical systems [17]. We also performed stability simulations in the continuous time version of our equation. We obtain similar results as in the discrete time case, but in another range of parameters [15]. Thus, bifurcations and chaos are also observed. However, it remains unclear whether we will keep the same learning and retrieval properties. Indeed, our learning scheme mainly relies on the correlation between active neurons at two successive time steps. In the discrete time system, these active neurons are changing quite rapidly because the dynamics goes at each time step from one part of the attractor to another one. In the continuous time system, we will lose this property. Therefore, learning may have to be adapted to this different behavior. This remains to be inquired.

Acknowledgments This work is supported by a French ACI program (neurosciences intégratives et computationnelles) on the dynamics of biologically plausible neural networks in collaboration with M. Samuelides (SupAéro, Toulouse), G. Beslon (INSA, Lyon), and E. Dauté (Perception et mouvement, Marseille).

References

1. S. Amari. Characteristics of random nets of analog neuron-like elements. *IEEE Trans. Syst. Man. Cyb.*, 2(3), 1972.

2. A.L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.
3. A.L. Barabasi and Z.N. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews, Genetics*, 5:101–113, 2004.
4. R. Ferrer Cancho, C. Janssen, and R.V. Solé. The topology of technology graphs: the small-world patterns in electronic circuits. *Phys. Rev. E*, 64:32767, 2001.
5. B. Cessac, B. Doyon, M. Quoy, and M. Samuelides. Mean-field equations, bifurcation map and route to chaos in discrete time neural networks. *Physica D*, 74:24–44, 1994.
6. B. Cessac and J.A. Sepulchre. Stable resonances and signal propagation in a chaotic network of coupled units. *Phys. Rev. E*, 70:056111, 2004.
7. E. Daucé, M. Quoy, B. Cessac, B. Doyon, and M. Samuelides. Self-organization and pattern-induced reduction of dynamics in recurrent networks. *Neural Networks*, 11:521–533, 1998.
8. E. Daucé, M. Quoy, and B. Doyon. Resonant spatio-temporal learning in large random neural networks. *Biological Cybernetics*, 87:185–198, 2002.
9. E. Daucé. Short term memory in recurrent networks of spiking neurons. *Natural Computing*, 3:135–157, 2004.
10. B. Doyon, B. Cessac, M. Quoy, and M. Samuelides. Chaos in neural networks with random connectivity. *Int. Journ. of Bifurcation and Chaos*, 3(2):279–291, 1993.
11. V.M. Eguiluz, D.R. Chiavo, G. Cecchi, and A.V. Apkarian. Scale-free brain functional networks. *Phys. Rev. Lett.*, 94:018102, 2005.
12. W.J. Freeman. Simulation of chaotic eeg pattern with a dynamic model of the olfactory system. *Biol. Cyber.*, 56:139–150, 1987.
13. H. Jeong, B. Tombor, R. Albert, Z.N. Oltvai, and A.L. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407:651–654, 2000.
14. M.E.J. Newman. The structure of scientific collaboration networks. *Proc. Natl. Acad. Sci. USA*, 98:404–109, 2001.
15. M. Quoy. Bifurcation scheme of continuous time random recurrent neural networks. *internal report*, 2004.
16. C.A. Skarda and W.J. Freeman. How brains make chaos in order to make sense of the world. *Behav. Brain Sci.*, 10:161–195, 1987.
17. H. Sompolinsky, A. crisanti, and H.J. Sommers. Chaos in random neural networks. *Phys. Rev. Lett.*, (61):259–262, 1988.
18. O. Sporns, D.R. Chiavo, M. Kaiser, and C.C. Hilgetag. Organization, development and function of complex brain networks. *Trends in Cognitive Sciences*, 8(9):418–425, 2004.
19. I. Tsuda. Towards an interpretation of dynamic neural activity in terms of chaotic dynamical systems. *Behav. Brain Sc.*, 24:793–847, 2001.
20. S. Valverde, R. Ferrer Cancho, and R.V. Solé. Scale-free networks from optimal design. *Europhys. Lett.*, 60:512–517, 2002.
21. A. Vázquez, R. Pastor-Satorras, and A. Vespignani. Large-scale topological and dynamical properties of the internet. *Phys. Rev. E*, 65:066130, 2002.
22. D.J. Watts and S.H. Strogatz. Collective dynamics of "small-world" networks. *Nature*, 393:440–442, 1998.
23. Y. Yao and W.J. Freeman. Model of biological pattern recognition with spatially chaotic dynamics. *Neural Networks*, 3:153–170, 1990.