PHYSICAL REVIEW E **70**, 066137 (2004)

Designing asymmetric neural networks with associative memory

Hong Zhao

Department of Physics, Xiamen University, Xiamen 361005, China

(Received 22 September 2004; published 27 December 2004)

A strategy for designing asymmetric neural networks of associative memory with controllable degree of symmetry and controllable basins of attraction is presented. It is shown that the performance of the networks depends on the degree of the symmetry, and by adjusting the degree of the symmetry the spurious memories or unwanted attractors can be suppressed completely.

DOI: 10.1103/PhysRevE.70.066137 PACS number(s): 84.35.+i, 87.18.Sn, 07.05.Mh, 05.45.−a

Neural networks [1] with associative memory can be classified as symmetric [2–7] and asymmetric networks [8–12] depending on whether the synaptic couplings, J_{ij}, between pairs of neurons, satisfy $J_{ij}=J_{ji}$, $i\neq j$. This kind of network has been studied extensively during the past two decades. However, some basic problems still remain unsolved. First, a network is very likely to be trapped in the vast spurious memories and therefore fails to recall the correct memories. To suppress the spurious memories, various algorithms have been presented [2,3,7–19], by which the unwanted attractors of spurious memories can be diminished to some extent. However, whether the spurious memories can be completely suppressed by suitable design is unknown. Second, asymmetric neural networks have much practical importance since the synaptic couplings are in general asymmetric in physiological nervous systems. It has been found that the asymmetric neural networks show some better performances than the symmetric ones, such as in recalling temporal patterns [8] and in suppressing spurious memories [12]. Nevertheless, a systematic algorithm for designing general asymmetric neural networks is still lacking and how to establish a connection between the performance of a neural network and the degree of the symmetry is unclear. Finally, the quality of the retrieval of a memory pattern depends on its basin of attraction, but in previous studies, the target of the design is aimed at finding a dynamical system whose fixed points involve the desired memory patterns. Therefore, an algorithm that can be further applied to control the attraction basins of memory patterns should be useful in controlling the quality of the retrieval.

In this paper I present an optimal strategy to design asymmetric neural networks with controllable degree of symmetry and controllable basins of attraction. I will particularly focus on illustrating a favorable property of the asymmetric networks, i.e., by adjusting the degree of symmetry suitably, spurious memories can be suppressed completely. I employ the Hopfield neural network

$$s_i(t+1)=sgn(h_i),\ h_i=\sum_{j=1}^{N}J_{ij}s_j(t),\quad i,j=1,\ldots,N, \qquad (1)$$

to introduce my idea. In this network the variable s_i takes the values 1 or −1. The goal of the design is to find the coupling matrix $\mathbf{J}$ (the diagonal elements J_{ii} of $\mathbf{J}$ are commonly set to 0) for which a given set of memory patterns $\{\xi_i^\mu\}$ ($\mu=1,\ldots,p$) are fixed points of (1). To be a fixed point, $\{\xi_i^\mu\}$ should satisfy $h_i^\mu\xi_i^\mu=\sum_{j=1}^{N}J_{ij}\xi_j^\mu$, or, equivalently,

$$h_i^\mu=\xi_i^\mu\sum_{j=1}^{N}J_{ij}\xi_j^\mu, \qquad (2)$$

with $h_i^\mu\geqslant 0$. The condition $h_i^\mu\geqslant 0$ ($i=1,\ldots,N$) is then referred to the fixed-point condition in the following. However, being stored as a fixed point is just the primary requirement as a memory pattern. An efficiently stored pattern should be retrievable practically. The quality of the retrieval is determined by the attraction basin of the fixed point, i.e., the larger the attraction basin, the easier the retrieval. To understand what determines the size of the attraction basin of a stored pattern, let us study a state obtained by proceeding with a flip $\xi_k^\mu\to-\xi_k^\mu$ on the kth bite of the μth pattern. For the new state, as has been pointed out in Ref. [1], it has $\tilde{h}_i^\mu=h_i^\mu-2J_{ik}\xi_i^\mu\xi_k^\mu$. If $\tilde{h}_i^u(i=1,\ldots,N)$ keep positive, the new state will be attracted to the μth pattern after only one step of evolution of (1). This observation can be extended to simultaneous multiple flips. A state obtained by proceeding with a set of flips on the μth pattern will be attracted to the original one after one step of the evolution, provided that $\tilde{h}_i^\mu=h_i^\mu-2\xi_i^\mu\Sigma_{\{k\}}J_{ik}\xi_k^\mu$ are positive, where $\{k\}$ represents the indexes of the flips. States of this kind form the core of the attraction basin of the μth pattern. To keep $\tilde{h}_i^\mu$ positive under a number of simultaneous flips, h_i^μ should be as big as possible, while $|\xi_i^\mu\Sigma_{\{k\}}J_{ik}\xi_k^\mu|$ should be as small as possible. Thus designing a neural network with associative memory is a typical problem of constrained optimization.

To achieve the goal of the constrained optimization, a simple way is to limit J_{ij} to $|J_{ij}|\leqslant d$ (thus $|\xi_i^\mu\Sigma_{\{k\}}J_{ik}\xi_k^\mu|\leqslant kd$) and search for the optimal solution which makes h_i^μ as big as possible. Here I take the simplest case of $|J_{ij}|=1$ to explain my approach. Notice from Eq. (2) that h_i^μ is determined merely by the ith row of $\mathbf{J}$, and thus the matrix can be designed row by row independently. My basic idea is to find the binary configuration $\{J_{ij},j=1,\ldots,N,j\neq i\}$ which leads the minimum of h_i^μ ($\mu=1,2,\ldots,p$), denoted by $h_i^{\min}$, taking a value as big as possible.

To start the design procedure, I endow binary numbers ±1 randomly to J_{ij} and set $J_{ii}=0$ for a fixed i. In this case, h_i^μ calculated following (2) distributes around 0. The task of the design procedure is to "drive" the set $\{h_i^\mu, \mu=1,\ldots,p\}$ to the positive region by continuous adaptations of $J_{ij}\to -J_{ij}$. I apply three steps to carry out such an adaptation.

At the first step, I compute $\{h_i^\mu, \mu=1,\ldots,p\}$ to find the minimum $h_i^{\min}$ of h_i^μ. There are usually multiple terms taking the same minimum, and the goal of this step is to find the set $\{h_i^{\mu_1},\ldots,h_i^{\mu_m}\}$ satisfying the condition $h_i^\mu = h_i^{\min}$ for $\mu \in \{\mu_1,\ldots,\mu_m\}$. At the second step, I calculate the sets $\{\xi_i^\mu J_{ij}^\mu \xi_j^\mu, \mu=\mu_1,\ldots,\mu_m\}$ for $j=1,2,\ldots,N$ and count the number m_i^j of the negative terms in each set. Let $m_i^{\max}$ represent the biggest one of m_i^j. Again multiple terms of m_i^j may take the same value $m_i^{\max}$. Let $\{j_1,\ldots,j_k\}$ record the indexes of j satisfying $m_i^j = m_i^{\max}$. At the third step, I randomly pick up an index j from the set $\{j_1,\ldots,j_k\}$ and make an adaptation $J_{ij}\to -J_{ij}$. This adaptation changes the sign of $\xi_i^\mu J_{ij}\xi_j^\mu$. As a result, there will be $m_i^{\max}$ terms in $\{\xi_i^\mu J_{ij}^\mu \xi_j^\mu, \mu=\mu_1,\ldots,\mu_m\}$ changing from the negative to the positive, and then $m_i^{\max}$ terms in $\{h_i^{\mu_1},\ldots,h_i^{\mu_m}\}$ are shifted to $h_i^{\min}+2$. In the mean time, other terms in $\{h_i^{\mu_1},\ldots,h_i^{\mu_m}\}$, with an amount of $m - m_i^{\max}$, will be moved to $h_i^{\min}-2$. At the beginning, since J_{ij} is set randomly, it has $m_i^j \sim m/2$ statistically, and thus $m_i^{\max} \geqslant m/2$ in general. Therefore, by proceeding the adaptations continuously, the set $\{h_i^\mu, \mu=1,\ldots,p\}$ will be shifted towards the positive direction continuously until an equilibrium is approached with $m_i^{\max}=m/2$ statistically. In practice, one can stop the procedure by applying a program-stop condition $h_i^\mu \geqslant c$, i.e., the adaptations are stopped once this criterion is satisfied for $\mu=1,\ldots,p$.

Applying the same procedure to each row of **J** one obtains a network with $h_i^\mu \geqslant c$ for $i=1,\ldots,N$ and $\mu=1,\ldots,p$. Since the adaptations are randomly proceeded for J_{ij} with $j \in \{j_1,\ldots,j_k\}$, I call this design procedure "Monte-Carlo-adaptation (MC-adaptation) rule."

For a matrix **J** in general it is difficult to quantify the degree of the symmetry. In the case of $|J_{ij}|=1$, nevertheless, it is easy to do so. In this case, a symmetric matrix has totally $N(N-1)/2$ symmetric elements that satisfy $J_{ij}=J_{ji}$, and a random matrix has $N(N-1)/4$ symmetric elements statistically since J_{ij} takes the same sign as J_{ji} with a probability of 0.5. In general, one can count the number of the symmetric elements, denoted as Γ, and define a symmetricity constant as

$$\sigma = \frac{2\Gamma}{N(N-1)}. \quad (3)$$

According to this definition, $\sigma=0,1$ define the antisymmetry and symmetry limits, respectively, and $\sigma=0.5$ quantifies the case of random couplings.

The MC-adaptation rule is indeed an optimal design strategy, and the quality of the networks can be controlled by the parameter c. Only if c is positive all the p patterns are stored as fixed points, and when c increases the cores of the attraction basins of these fixed-point attractors increase. Furthermore, it will show that the degree of the symmetry of a network is also determined by c. Obviously, the MC-adaptation rule is different from the previous algorithms, such as the Hebb rule [2], the pseudoinverse rule [5], the Dale hypothesis [9] and the perceptron type algorithms [20,21]. These algorithms calculate J_{ij} in anticipation of storing memories as fixed points of the networks, but do not pursue large attraction basins explicitly in their prescriptions. A common disadvantage of these algorithms is the large amount of coexisting spurious memories. To show how serious the problem of spurious memories is, let us consider the local learning rule—Hebb rule $J_{ij}=(1/N)\Sigma_{\mu=1}^{p}\xi_i^\mu\xi_j^\mu$—and the global learning rule—pseudoinverse rule [5] $J_{ij}=(1/N)\Sigma_{\mu,\nu=1}^{p}\xi_i^\mu(\mathbf{C}^{-1})_{\mu\nu}\xi_j^\nu$, where $\mathbf{C}^{-1}$ is the inverse of the overlap matrix **C** defined by the rule $C_{\mu\nu}=(1/N)\Sigma_{i=1}^{N}\xi_i^\mu\xi_i^\nu$, as examples. The number of patterns that can be stored as fixed points without errors is found to be $p=\alpha N$. The ratio α is very low in the former case ($\alpha < 1/(2\ln(N))$ [4]), while it is extremely high in the latter case ($\alpha<1$ [6,7]). But whatever the maximum storage ratios are, spurious memories exist in both cases. In Fig. 1(a), in the case of $n=1000$, I plot P_{total}, the total percentage of initial states attracted to the memory patterns, against the number p of the memory patterns for networks designed by the Hebb rule and the pseudoinverse rule correspondingly. For each p, I average P_{total} for ten sets of randomly selected memory patterns, and, for each set, 10 000 initial states are checked. As usual, the pattern $\{-\xi i\mu\}$ is equated to $\{\xi_i^\mu\}$ because of the symmetric property of the dynamics (1). For both rules, P_{total} decreases quickly, e.g., at $p=50$, P_{total} is already smaller than 0.1, which means that most of the initial states are attracted to unwanted attractors.

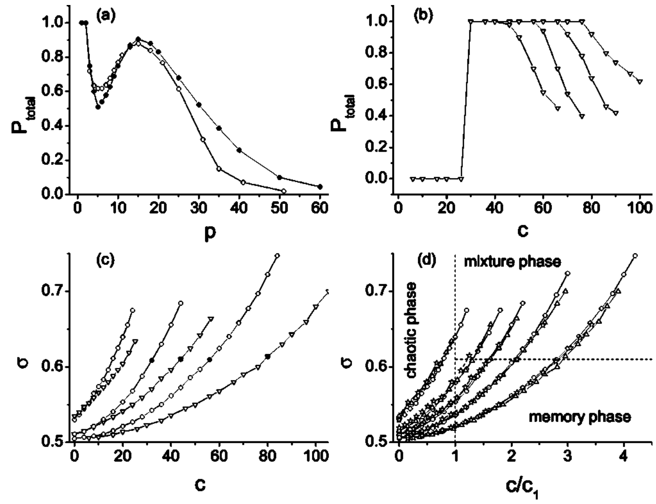

FIG. 1. (a) P_{total} against p for the neural networks designed by the Hebb rule (solid circles) and the pseudoinverse rule (circles) for $N=1000$. (b) P_{total} against c for the neural networks designed by the MC-adaptation rule for $N=1000$. From right to left: $\alpha=0.03$, 0.05, 0.07 and 0.09, correspondingly. (c) The symmetricity constant σ against c for $N=500$ (circles) and $N=1000$ (triangles). From right to left: $\alpha=0.03$, 0.07, and 0.15, correspondingly. The solid symbols represent the second turning points c_2. (d) The symmetricity constant σ against c/c_1 for $N=500$ (circles), $N=1000$ (triangles), and $N=2000$ (stars). From right to left: $\alpha=0.03$, 0.05, 0.07, 0.09, and 0.15, correspondingly. The lines in each plot are shown for guarding the eye.

On the contrary, the spurious memories may disappear totally in the networks designed by the MC-adaptation rule. Figure 1(b) shows P_{total} against c in the case of $n=1000$ for

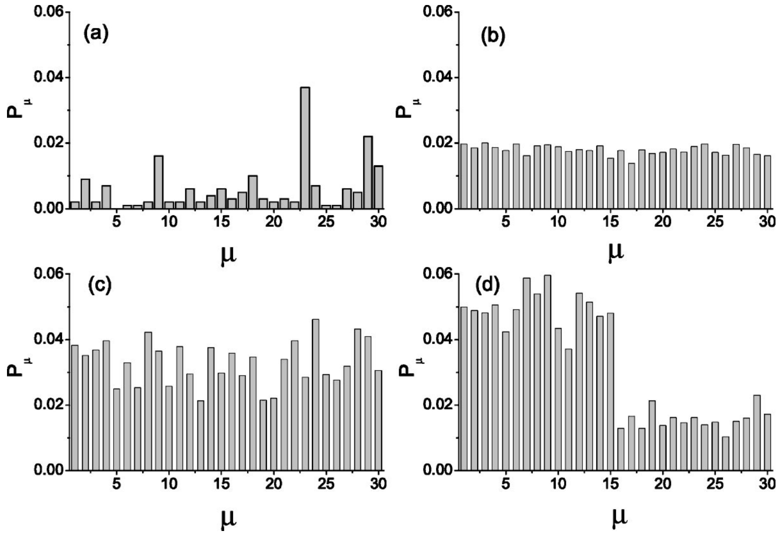

FIG. 2. P_μ against μ for a set of 30 memory patterns for neural networks design by (a) the Hebb rule, (b) the pseudoinverse rule, (c) the MC-adaptation rule with $c=70$, and (d) the MC-adaptation rule with $c=74$ for the first 15 patterns and $c=70$ for the last 15 patterns.

several p. One can find two turning points, c_1 and c_2, from each curve. In the region $c<c_1$, it has $P_{total}=0$, i.e., almost no initial state is attracted to the memory patterns. Tracing the evolution of the initial states, one can find that the initial states are attracted to chaotic orbits, and therefore this parameter regime is named as "chaotic phase." In the interval $c_1<c<c_2$, it has $P_{total}=1$, which indicates that almost all the initial states are attracted to the memory patterns, i.e., there is no spurious memories at all, and I call this regime the "memory phase." When $c>c_2$, P_{total} decreases as c increases, which indicates that the unwanted attractors appear and increase with c (I checked these attractors and confirmed that there are periodic orbits). This regime is called the "mixture phase."

The symmetricity constant σ is closely related to c. My numerical analysis shows that σ increases with c, as shown in Fig. 1(c), where I plot σ against c for $\alpha=0.03$, 0.07 and 0.15 in the case of $N=1000$ and $N=500$ correspondingly. Each value of σ in the figure is obtained by averaging the values of σ for ten sets of randomly selected memory patterns with fixed p and N, but I would like to point out that σ is insensitive to the detail of the memory patterns.

The value of c_1 appears to be only determined by the system size N, e.g., in Fig. 1(b) $c_1=28$ for different p. Indeed, further calculations show that c_1 behaves as $c_1\sim N^{1/2}$. The second turning point is found to always take place at $\sigma(c_2)\approx 0.61$. In Fig. 2(c), solid symbols represent those points of $\sigma=\sigma(c_2)$, which confirms that σ takes roughly the same value at c_2 for different N and α.

There is a well-defined scaling property between σ and c. In Fig. 1(d) I plot σ against the rescaled parameter c/c_1, which shows that the curves with the same value of α collapse into a universal curve. As a result, one obtains a universal phase diagram that is independent of the system size N. This phase diagram is divided into three regimes with different properties, i.e., the chaotic phase $(0<c/c_1<1)$, the memory phase $(c/c_1\geqslant 1, 0.5<\sigma\leqslant 0.61)$, and the mixture phase $(c/c_1\geqslant 1, \sigma>0.61)$.

The memory phase is obviously preferable for the purpose of storing and retrieval since any initial states can not be trapped into unwanted attractors. This is an important advantage of the asymmetric neural networks designed by the present algorithm. However, the other two "phases" may also have available properties. To stress this statement let us analyze the "chaotic phase" as an example. Exactly, a neural network designed in this regime has a nonvanishing attraction-basin for each memory pattern. The reason is as follows. For $c=20$ in the case of $N=1000$, for instance, it has $h_i^\mu\geqslant 20$ for all the memory patterns. Thus, $\tilde{h}_i^\mu(=h_i^\mu-2\xi_i^\mu\Sigma_{\{k\}}J_{ik}\xi_k^\mu)$ should remain positive for k simultaneous flips if only $k\leqslant 10$, which implies that there are at least 2^{80} states belonging to the attraction basin of a memory pattern. I have not found that the random initial states are attracted to the memory patterns [i.e., $P_{total}=0$ for $c<c_1$ in Fig. 1(b)] just because that the attraction basins with this size are negligibly small compared with the huge configuration space with 2^{1000} states. But neural networks designed in this parameter region may be applied for sensitive recognition, i.e., an initial state is attracted to a memory pattern only if it has a high similarity with the pattern, otherwise it will wander in a chaotic orbit for ever.

I show in the following how to control the attraction basins of individual memories by ending different values of c to different memories. I fix a set of 30 quenched patterns as the memory patterns in the case of $N=1000$. Figures 2(a)–2(c) illustrate P_μ against μ for the networks designed using the Hebb rule, the pseudoinverse rule and the MC-adaptation rule with $c=70$, respectively, where P_μ is the percentage of 10 000 initial states attracted to the μth pattern. It can be found that P_μ are different significantly in the first plot while they are roughly uniformly distributed in the second and third plots, which indicates that the attraction basins of the individual patterns are dramatically different in the Hebb rule case while the difference is slight in the last two cases. The advantage of the MC-adaptation rule over the pseudoinverse rule is that there is no spurious memory and, also importantly, the sizes of the attraction basins are controllable by endowing different c's to different patterns. Figure 2(d) shows an example applying $h_i^\mu\geqslant 74$ to the first 15 patterns while $h_i^\mu\geqslant 70$ is applied to the last 15 ones. This simple treatment results in the increase (decrease) of the attraction basins of the first (last) part dramatically.

In summary, the MC-adaptation rule can be applied to design neural networks with controllable degree of symmetry and controllable basins of attraction. For a fixed storage ratio α, the symmetricity constant σ is a universal function of the rescaled control parameter c/c_1 and independent of the system size N. The performance of the neural networks is closely related to the degree of the symmetricity. In the low-symmetry region of $0<c/c_1<1$, small-scale attraction basins of the memory patterns are embedded into the "chaotic sea." In the moderate-symmetry region of $1<c/c_1<c_2/c_1$ with $\sigma\leqslant 0.61$, any initial state will be attracted to one of the memory patterns and spurious memories disappear completely. In the high-symmetry case of $\sigma>0.61$, on the contrary, memory patterns and spurious memories coexist, and the amount of spurious memories increases quickly with σ. Thus, for suppressing spurious memories the networks with moderate symmetry (which is corresponding to intermediate value of c) are preferable. Further increase of the threshold c can increase the sizes of the cores of the attraction basins of the memory patterns, but may also induce spurious memories. The physics behind these findings needs to be clarified in the future.

From Figs. 1(b) and 1(d) one can realized that the memory phase exists in the condition of $\alpha<0.1$. This value of α thus gives a threshold of perfect storage capacity without being trapped into any spurious memories. As to the capacity of storing memories as fixed points, Fig. 1(d) indicates that a storage capacity with $\alpha\leqslant 0.15$ can be approached at least. I would like to point out that the present algorithm can be applied directly to improve the storage ratio of any neural networks designed by other algorithms. The approach is very simple: Starting the design procedure by applying the connection matrix initially obtained by a specific algorithm, one can optimize the design and further improve its storage capacity. Finally, I would like to point out that the present design strategy can be easily extended to more general cases, such as continuous couplings, more general gain functions, and multistate neurons, etc.

Particular thanks are given to Professor Schuster from whom I got a lot of useful ideas and suggestions related to this work. This work is supported in part by the Major State Research Development 973 project of nonlinear science in China, the National Natural Science Foundation of China under Grant No. 10475067, and the Distinguished Visiting Scholar Program of the Chinese Government.

[1] H. G. Schuster, *Complex Adaptive Systems* (Scator Verlag, Saarbrücken, 2001).
[2] J. J. Hopfield, Proc. Natl. Acad. Sci. U.S.A. **79**, 2554 (1982); **81**, 3088 (1984); J. J. Hopfield, D. I. Feinstein, and R. G. Palmer, Nature (London) **304**, 159 (1983).
[3] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Phys. Rev. A **32**, 1007 (1985); Phys. Rev. Lett. **55**, 1530 (1985).
[4] R. J. McEliced, E. C. Posner, E. R. Rodemick, and S. S. Venkatesh, IEEE Trans. Inf. Theory **IT-33**, 461 (1987).
[5] L. Personnaz, I. Guyon, and G. Dreyfus, J. Phys. (Paris), Lett. **46**, L-359 (1985).
[6] I. Kanter and H. Sompolinsky, Phys. Rev. A **35**, 380 (1987).
[7] C. M. Marcus, F. R. Waugh, and R. M. Westervelt, Phys. Rev. A **41**, 3355 (1990).
[8] H. Sompolinsky and I. Kanter, Phys. Rev. Lett. **57**, 2861 (1986).
[9] T. Fukai and M. Shiino, Phys. Rev. Lett. **64**, 1465 (1990).
[10] I. Kanter and E. Eisenstein, Phys. Rev. Lett. **65**, 520 (1990).
[11] B. Derrida, E. Gardner, and A. Zippelius, Europhys. Lett. **4**, 167 (1987).
[12] G. Parisi, J. Phys. A **19**, L675 (1986).
[13] F. R. Waugh, C. M. Marcus, and R. M. Westervelt, Phys. Rev. Lett. **64**, 1986 (1990).
[14] J. L. vanHemmen and R. Kuehn, Phys. Rev. Lett. **57**, 913 (1986).
[15] P. N. McGraw and Michael Menzinger, Phys. Rev. E **67**, 016118 (2003).
[16] P. R. Krebs and W. K. Thenmann, Phys. Rev. E **60**, 4580 (1999).
[17] D. Bolle and J. Huyghebaert, Phys. Rev. E **48**, 2250 (1993).
[18] M. Morita, Neural Networks **6**, 115 (1993).
[19] S. Yoshizawa *et al.*, Neural Networks **6**, 167 (1993).
[20] S. Diederich and M. Opper, Phys. Rev. Lett. **58**, 949 (1987).
[21] E. Gardner, Europhys. Lett. **4**, 481 (1987).