PHYSICAL REVIEW E **72**, 066111 (2005)

Pattern recognition using asymmetric attractor neural networks

Tao Jin
Physics Department of Lanzhou University, Lanzhou 730000, China

Hong Zhao*
Physics Department of Xiamen University, Xiamen 361005, China
(Received 13 April 2005; revised manuscript received 24 October 2005; published 9 December 2005)

The asymmetric attractor neural networks designed by the Monte Carlo– (MC-) adaptation rule are shown to be promising candidates for pattern recognition. In such a neural network with relatively low symmetry, when the members of a set of template patterns are stored as fixed-point attractors, their attraction basins are shown to be isolated islands embedded in a "chaotic sea." The sizes of these islands can be controlled by a single parameter. We show that these properties can be used for effective pattern recognition and rejection. In our method, the pattern to be identified is attracted to a template pattern or a chaotic attractor. If the difference between the pattern to be identified and the template pattern is smaller than a predescribed threshold, the pattern is attracted to the template pattern automatically and thus is identified as belonging to this template pattern. Otherwise, it wanders in a chaotic attractor for ever and thus is rejected as an unknown pattern. The maximum sizes of these islands allowed by this kind of neural networks are determined by a modified MC-adaptation rule which are shown to be able to dramatically enlarge the sizes of the islands. We illustrate the use of our method for pattern recognition and rejection with an example of recognizing a set of Chinese characters.

DOI: 10.1103/PhysRevE.72.066111 PACS number(s): 84.35.+i, 42.30.Sy, 05.45.−a, 07.05.Mh

I. INTRODUCTION

Pattern recognition and its many different applications have long been studied [1–8]. Many methods and techniques have been proposed [9–18]. Among them, the template matching, syntactic or structural matching, statistical classification, and neural network approaches are well known. A brief description and comparison of these approaches were mentioned in Ref. [1].

In recent years, the neural network approach [2,3,5,7,9,10,12,18–20,22–27] has drawn more and more attention because of its several remarkable advantages: (1) This approach deals with information parallelly, which is much more effective than other approaches that deal with information serially. In fact, the neural network approach deals with information mimicing the real neural networks of living things. (2) The training algorithms or learning rules are quite universal, by which the artificial neural networks used for pattern recognition are established. That is to say that these learning rules depend insensitively on domain-specific knowledge and thus can be easily applied to many different problems. (3) The neural networks have the capability of learning nonlinear input-output relationships, offering the potential to solve complicated problems.

The popular neural networks employed in pattern recognition are the feed-forward networks [7,20,22]. To design such a neural network, the training process uses all of the available training samples. For example, if such a network is designed to identify four types of tanks, four sets of training samples should be provided. Each set is obtained by coding different photos of one type of tanks. When the training samples are true representatives of the targets and their amount is large enough, the feed-forward networks show a powerful capability in pattern recognition. However, the training set is usually too small in practice and the neural network designed by them is fallacious.

As another important family of artificial neural networks, the attractor neural networks [2,9,19,23–27] also have potential applications to pattern recognition. An attractor neural network is a dynamic system. For a general symmetric network, the possible attractors of the system are the fixed-point attractors or the period-2 periodic attractors; for a general asymmetric network, the fixed-point attractors, long-period periodic attractors, and chaotic attractors may coexist. To employ such a network to identify the four types of tanks, four corresponding template patterns are stored as four fixed-point attractors. When a pattern to be identified is input as an initial state, the system will relax to a template pattern if it has higher similarity. In other words, if this initial state belongs to the attraction basin of a fixed-point attractor, the network will converge to the corresponding template pattern. In this way, recognition can be done. Thus, to design an attractor neural network for pattern recognition, one can use only one sample for each type of tanks.

However, the attractor neural networks have been rarely used in pattern recognition because of the following reasons. First, a huge number of unwanted fixed-point attractors coexist with the attractors of the template patterns. As a result, the pattern to be identified has a great probability of being attracted to unwanted attractors (i.e., wrong recognition) even though it shows high similarity with one of the template patterns. These unwanted attractors are an awkward problem in attractor neural networks. Several strategies have been proposed to suppress this kind of attractors [23–27], but no

*Electronic address: zhaoh@xmu.edu.cn.

one can eliminate them completely. Second, a rejection mechanism is absent for the existing attractor neural networks. Because of lacking a rejection mechanism, a pattern to be identified may be attracted to a template pattern though it shows low similarity with this template pattern. Rejection is an essential factor in pattern recognition. A preferable recognition algorithm should be able to not only recognize a pattern if it has high similarity with a template pattern but also reject recognizing a pattern as any template pattern if it has low similarity with them. The threshold of similarity, above which a pattern to be identified should be recognized, is dependent on the specific applications and highly expected to be controllable.

Recently, an algorithm named the Monte Carlo– (MC-) adaptation rule was developed in Ref. [9] to design asymmetric attractor neural networks. By applying this rule, the performance of the networks can be controlled by a parameter c. It is found that the neural networks show different dynamic behavior as c changes. In the range of $c \leqslant c_1$, the attraction basins of the template patterns are embedded into a "chaotic sea." That is, in this range one chaotic attractor coexists with those fixed-point attractors acting as template patterns. This range is called the "chaos phase," and neural networks with this phase have a low degree of symmetry. In the range of $c_1 < c \leqslant c_2$, those template patterns are the only attractors and any initial state will be attracted to one of them. This range is referred as the "memory phase" and neural networks with this phase have a moderate degree of symmetry. When the parameter c exceeds c_2, one encounters the so-called "mixture phase." In this phase, the unwanted fixed-point attractors appear and coexist with those of the template patterns. The degree of symmetry in this range is relatively high.

The chaos phase and the memory phase are a particular behavior of the attractor neural networks designed by the MC-adaptation rule. Since the unwanted attractors are eliminated completely, the neural networks with memory phase are more suitable for storing memory patterns. However, for the purpose of pattern recognition, this phase is not good enough because any initial state will be attracted to a template pattern and the rejection will never be carried out.

In this paper, we show that the neural networks with chaos phase are intrinsically suitable for pattern recognition. When a pattern to be identified has high similarity with a template pattern, it will be attracted to the template pattern and thus be recognized. When it has no similarity or has a low degree of similarity with any template pattern, it will wander in the chaotic attractor forever. That is, it will never be attracted to a template pattern and result in a wrong recognition. In this way, the neural networks with chaos phase overcome the two limitations of the usual attractor neural networks for pattern recognition.

The threshold of similarity above which the initial pattern should be recognized as the corresponding template pattern can be controlled by the parameter c in the range of $c \leqslant c_1$. However, this controllable range is unexpectedly narrow in the neural networks designed by the original MC-adaptation rule. This drawback limits the application scope of this approach. We therefore develop a modified version of the MC-adaptation rule in present paper to extend the controllable range.

The rest of this paper is organized as following. We introduce the basic idea of the MC-adaptation rule in Sec. II and present a modified version in Sec. III. Why and how the neural networks with chaos phase can be used for pattern recognition are explained carefully in Sec. IV. As an example of an application, in this section, we use the neural network designed by the modified MC-adaptation rule to identify the printed Chinese characters which are distorted by random noise. The last section is the conclusion.

II. BRIEF INTRODUCTION OF THE ORIGINAL MC-ADAPTATION RULE

An attractor neural network with N neurons may be described by

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

Here, $s_i(t) \in \{+1, -1\}$ represents the state of the ith neuron at time t and $h_i(t)$ is the local field acting on it. The synaptic coupling between the ith and jth neurons is represented by J_{ij}. To design an attractor neural network is to find a coupling matrix $\mathbf{J}$, by which a given set of template patterns $\{\xi_i^\mu, \mu=1, \ldots, p\}$ with $\xi_i^\mu = \pm 1$ is stored as a set of fixed-point attractors. As a fixed point, the local field of the μth pattern h_i^μ must satisfy $\bar{h}_i^\mu \xi_i^\mu = \Sigma_{j=1}^N J_{ij} \xi_j^\mu$ or, equivalently,

$$\bar{h}_i^\mu = \xi_i^\mu \sum_{j=1}^{N} J_{ij} \xi_j^\mu, \tag{2}$$

with $\bar{h}_i^\mu \geqslant 0$. For the purpose of memory retrieval and pattern recognition, it is desired that the attraction basins of these fixed-point attractors be big enough. In other words, a good design procedure should not only can store the given template patterns as fixed-point attractors but also can control their attraction basins.

The attraction basin of a fixed-point attractor is related to the core which is defined as a set of initial states attracted to this attractor by only one step of evolution. Roughly, the larger the core, the bigger the attraction basin. To understand what determines the size of the core, let us consider a state $\{\tilde{\xi}_i^\mu\}$ obtained by making n mutations $\xi_{j_k}^\mu \to -\xi_{j_k}^\mu$ on the μth pattern, where $\{j_k, k=1, \ldots, n\}$ represent the positions of these mutations. One may easily derive the local fields related to this mutated pattern: $\tilde{h}_i^\mu = \bar{h}_i^\mu - 2\xi_i^\mu \Sigma_{k=1}^n J_{ij_k} \xi_{j_k}^\mu$. It is clear that the new state will stop at $\{\xi_i^\mu\}$ after only one step of evolution if $\tilde{h}_i^\mu$ is positive for $i=1, \ldots, N$. Therefore, the size of the core is related to the maximum of n, denoted by n_{max}, below which $\{\tilde{h}_i^\mu\}$ remain positive. There are no less than $2^{n_{max}}$ states belonging to the core. The maximum of n_{max} is obtained by making $\bar{h}_i^\mu$ as big as possible and $|\xi_i^\mu \Sigma_{k=1}^n J_{ij_k} \xi_{j_k}^\mu|$ as small as possible. This is a typical problem of constrained optimization, and the MC-adaptation rule provides an effective procedure to solve it.

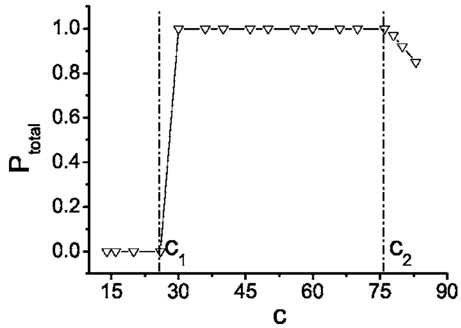

FIG. 1. P_{total} against c for the neural network designed by the original MC-adaptation rule with $N=1000$ and $p=30$.

The basic idea of the MC-adaptation rule is to limit $|J_{ij}|\leqslant d$ (thus $|\xi_i^\mu \Sigma_{k=1}^n J_{ij_k}\xi_{j_k}^\mu|\leqslant nd$) and push the set of $\{\bar{h}_i^\mu\}$ to the positive region by continuously adapting $\{J_{ij}\}$. In this paper, for the sake of simplicity, we restrict $|J_{ij}|=1$ and the adaptation is degenerated as $J_{ij}\to -J_{ij}$. The adaptation process is stopped when the condition of $\bar{h}_i^\mu \geqslant c$ is satisfied for $\mu=1,\ldots,p$ and $i=1,\ldots,N$. It is easy to obtain $n_{max}\approx c/2d$. Therefore, the size of the core can be controlled by the parameter c and so are the attraction basins of the template patterns. When $c>0$, all the given patterns are surely stored as fixed-point attractors and have nonvanishing attraction basins.

To estimate the relative size of the attraction basin of the μth template pattern, one may measure the percentage of random initial states attracted to this pattern. It is denoted by P_μ. Therefore, $P_{total}=\Sigma_{\mu=1}^p P_\mu$ gives the total percentage of these initial states attracted to the set of template patterns. This quantity has been commonly employed to measure the quality of the neural networks in previous works [9,21].

The relationship between P_{total} and c is shown in Fig. 1 for the neural network designed by the original MC-adaptation rule with $N=1000$ and $p=30$. For each point, we average P_{total} over 10 sets of randomly selected template patterns; for each set, 10 000 randomly selected initial states are checked out. As usual, the pattern $\{-\xi_i^\mu\}$ is equated to $\{\xi_i^\mu\}$ because of the symmetric property of the dynamics, Eq. (1). Figure 1 clearly indicates that the parameter space of c is divided into three ranges by two turning points, which will be denoted in the following as c_1 and c_2, respectively. Around the first turning point, P_{total} changes from $P_{total}=0$ at $c=26$ to $P_{total}=1$ at $c=28$. Notice that the parameter c only takes even integers in the situation of $|J_{ij}|=1$; this change is quite sharp. The last value of c that satisfies $P_{total}(c)=0$ is defined as c_1. At the second turning point, P_{total} changes from $P_{total}=1$ at $c=76$ to $P_{total}<1$ at $c=78$. Similarly, the last value of c that satisfies $P_{total}(c)=1$ is defined as c_2. According to these definitions, one obtains $c_1=26$ and $c_2=76$ in Fig. 1.

The parameter c determines the dynamic behavior of the system. In a different range of c, the system has different dynamic behavior. In the range of $c\leqslant c_1$, it has $P_{total}=0$. In this range, a chaotic attractor coexists with the p template patterns. We will further discuss the dynamical behavior in this range in Sec. IV. In the range of $c_1<c\leqslant c_2$, it has $P_{total}=1$. This fact indicates that the template patterns are the only attractors and the unwanted attractors are suppressed completely. In the range of $c>c_2$, it has $P_{total}<1$. This result reveals that the unwanted attractors appear and coexist with those of the template patterns. The three ranges have been discussed in detail in Ref. [9] and named the "chaos phase," the "memory phase," and the "mixture phase," respectively.

III. MODIFIED MC-ADAPTATION RULE

The original MC-adaptation rule leads to $\bar{h}_i^\mu \geqslant c$. As a result, the local fields of all the template patterns distribute over the half-closed intervals of $h_i^\mu \leqslant -c$ and $h_i^\mu \geqslant c$ for $\xi_i^\mu=1$ and -1, respectively. In this section, a modified version of the MC-adaptation rule is presented in order to further control the distribution of the local fields $\{h_i^\mu\}$. The next section will illustrate the benefit of this modification for pattern recognition.

The goal of the modified rule is to constrain $\{\bar{h}_i^\mu\}$ to the interval of $[c,c']$ with $c'>c>0$ and to make the width of this interval as small as possible. The procedure used to achieve this goal is described in the following. For the sake of simplicity, we restrict $J_{ij}=\pm 1$ as mentioned above. The design procedure starts with randomly endowing J_{ij} with ±1. Then, a set of p patterns is randomly selected as the template patterns to be stored. In the beginning, since J_{ij} is random, the local fields $\{h_i^\mu\}$ satisfy a Gaussian distribution with zero mean. Our task is to "drive" $\{\bar{h}_i^\mu\}$ into the closed interval $[c,c']$ by continual adaptation of $J_{ij}\to -J_{ij}$. For the convenience of description, we define a new parameter

$$b_i^\mu = -|\bar{h}_i^\mu - (c+c')/2| + (c'-c)/2. \tag{3}$$

If $b_i^\mu \geqslant 0$, one obtains $c\leqslant \bar{h}_i^\mu \leqslant c'$. Therefore, $b_i^\mu \geqslant 0$ is the criterion that the design goal is achieved. In other words, when $b_i^\mu \geqslant 0$ is approached, $\{h_i^\mu\}$ are limited in the interval of $[c,c']$ for $\xi_i^\mu=+1$ or the interval of $[-c',-c]$ for $\xi_i^\mu=-1$.

Notice from Eq. (2) that $\bar{h}_i^\mu$ is merely determined by the ith row of $\mathbf{J}$, so the coupling matrix can be designed row by row independently. We apply the following three steps repeatedly to design each row of $\mathbf{J}$. At the first step, we calculate $\{b_i^\mu,\mu=1,\ldots,p\}$ and find the minimum b_i^{min} of this set. There are usually many terms taking the same minimum, and the goal of this step is to search the set $\{b_i^{\mu_1},\ldots,b_i^{\mu_m}\}$ satisfying the condition of $b_i^\mu = b_i^{min}$ for $\mu\in\{\mu_1,\ldots,\mu_m\}$. At the second step, for each $J_{ij}\in\{J_{ij};j=1,\ldots,N,j\neq i\}$ we calculate $\{C_i^\mu \xi_i^\mu J_{ij}\xi_j^\mu,\mu=\mu_1,\ldots,\mu_m\}$ and count the number of the negative terms m_i^j in each subset. Here, $C_i^\mu=-1$ for $\bar{h}_i^\mu>c$ and $C_i^\mu=+1$ for others. Let m_i^{max} represent the maximum of m_i^j. Again many terms of m_i^j may take the same value of m_i^{max}. To obtain the set of index $\{j_1,\ldots,j_n\}$ satisfying $m_i^j=m_i^{max}$ is the goal of this step. At the third step, we randomly pick up an index j from the set of $\{j_1,\ldots,j_n\}$ 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, m_i^{max} terms in $\{\bar{h}_i^{\mu_1},\ldots,\bar{h}_i^{\mu_m}\}$ will be pushed towards the interval $[c,c']$. It is obvious that $m_i^{max}\geqslant n/2$ in general. Therefore, $\{\bar{h}_i^\mu\}$ will be pushed to-

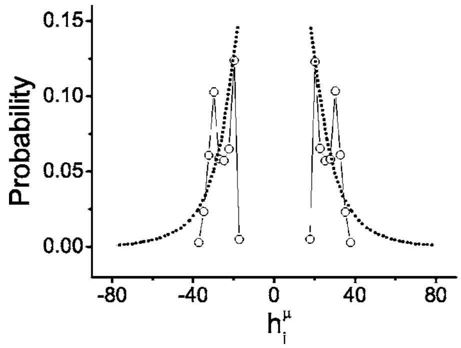

FIG. 2. The probability distributions of h_i^μ obtained by the modified MC-adaptation rule (circles) and the original MC-adaptation rule (short-dashed line) with $N=1000$, $p=30$, and $c=18$.

wards the interval $[c, c']$ gradually by continuously repeating the three steps. The design procedure stops when the criterion $b_i^\mu \geqslant 0$ is satisfied for $\mu=1,\ldots,p$.

Applying the same procedure to each row of $\mathbf{J}$, one may obtain a network with $b_i^\mu \geqslant 0$ for $i=1,\ldots,N$ and $\mu=1,\ldots,p$. That is, all h_i^μ are pushed into the two prescribed intervals $[c, c']$ and $[-c', -c]$. The parameter c' can be set independently. In this paper, it is determined in the following way. Fixing c at a value and temporally fixing the interval $[c, c']$, we proceed with the designing procedure to push all of the $\bar{h}_i^\mu$ into this interval. Then decrease c' a little and repeat the same operation. This procedure of decrease is stopped when it fails to push all of the h_i^μ into the anticipated intervals. In this way, the two control parameters degenerate into one.

In order to make a comparison, Fig. 2 shows the probability distributions of $\{h_i^\mu\}$, which are obtained by the original MC-adaptation rule and the modified MC-adaptation rule with $c=18$ for $N=1000$ and $p=30$. The consequences of the two design procedures are obviously different.

The dynamical behavior of the neural networks designed by the modified rule is qualitatively the same as that of the neural networks designed by the original rule. Figure 3 shows P_{total} against c for several p in the case of $N=1000$. The results indicate that the primary advantages are inherited. For each p, the parameter c is also divided into three ranges—i.e., the chaos phase, the memory phase, and the

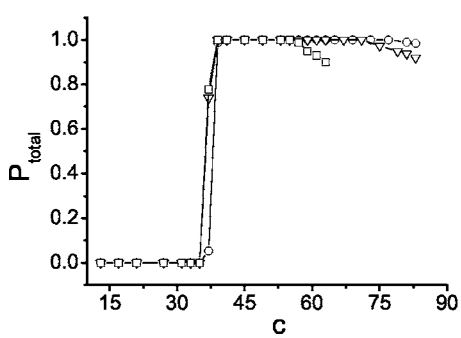

FIG. 3. P_{total} against c for the neural networks designed by the modified MC-adaptation rule with $N=1000$ and $p=20$ (circles), 30 (triangles), and 60 (squares), correspondingly.

mixture phase by two turning points c_1 and c_2.

The turning point c_1, which divides the chaos phase and the memory phase, appears as an universal constant independence of the storage capacity. That is, it remains almost unchanged for different p as shown in Fig. 3. The turning point c_2, where the P_{total} changes from $P_{total}=1$ to $P_{total}<1$, has different values depending on p. It decreases with an increase of p. These are similar to the neural networks designed by the original MC-adaptation rule [9].

However, there is an significant difference. By comparing Fig. 1 with Fig. 3, one may realize that the value of c_1 of the neural networks designed by the modified procedure is much bigger than that of the neural networks designed by the original procedure. At $N=1000$, the former has $c_1=36$, while the latter has $c_1=26$. In the next section, one will see that this improvement is crucial for practical applications.

IV. APPLICATION OF THE CHAOS PHASE TO PATTERN RECOGNITION

For the purpose of pattern recognition, as mentioned in the Introduction, we are more interested in the neural networks with chaos phase. Thus it is worthy of more detailed descriptions. It can be confirmed that a random initial state is not attracted to a limit cycle with long period because we have never observed its recurrence in a reasonable time scale. We believe that there exists a single chaotic attractor because we have checked that different initial states show the same asymptotic behavior: the Lyapunov exponents calculated by these states are positive and approach each other (this method is usually applied to test whether different initial orbits are attracted to the same chaotic attractor in low-dimensional systems). For neural networks, it is true that there is no exact chaotic orbit since the configuration space is finite. However, it is large enough (2^{1000} in our case) for practical applications. We would like to point out that the dynamic behaviors of the neural networks designed by the MC-adaptation rule are different from those of other asymmetric neural networks [28–30], in which various limit cycles with different lengths may coexist.

The result of $P_{total}=0$ in the chaos phase implies that the probability of a random initial state attracted to the template patterns is negligible or, in other words, such an initial state is almost surely attracted to the chaotic attractor. However, it does not mean that the attraction basins of those template patterns are vanishing. On the contrary, according to the analysis in Sec. II, the template patterns in this phase surely have nonvanishing attraction basins because of $c>0$.

Let m represent the normalized overlap of two system states, which describes their similarity. To measure the difference between them, one may introduce a new variable ds defined as

$$ds = \frac{1-m}{2}. \tag{4}$$

According to this definition, Nds gives the number of different elements between the two states.

For the μth pattern $\{\xi_i^\mu\}$, the attraction basin can be estimated in the following way. Fixing a value of ds, we ran-

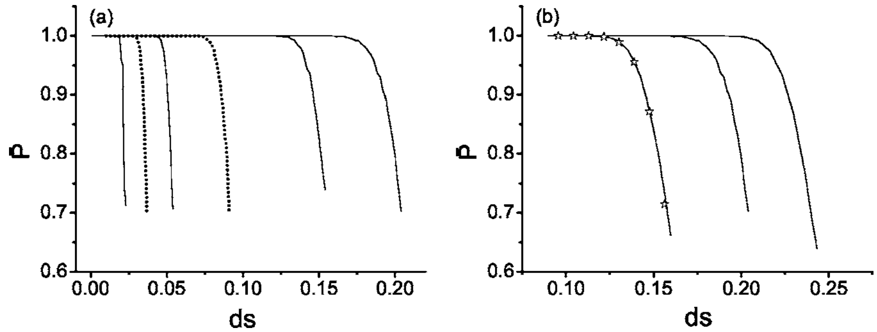

FIG. 4. (a) $\bar{P}$ against ds in the case of $N=1000$ and $p=30$. The solid lines represent the results obtained by the modified MC-adaptation rule with several c, from right to left: $c=36,34,26,18$; the short-dashed lines represent the results obtained by the original MC-adaptation rule with $c=26$ (the right one) and $c=20$ (the left one). (b) $\bar{P}$ against ds at $c=c_1$ for the neural networks designed by the modified MC-adaptation rule, from right to left: $N=1500$, 1000, and 576. The stars represent the results obtained by using the Chinese characters as the template patterns.

domly select Nds elements from this pattern and flip them by $\xi_i^\mu \rightarrow -\xi_i^\mu$. Then, we evolve the system using this mutated pattern as the initial state to find whether or not it is attracted to $\{\xi_i^\mu\}$. In this way, one may calculate the percentage of this kind of initial states being attracted to the μth pattern, which is denoted by $\bar{P}_\mu$ to distinguish from P_μ. We further apply $\bar{P}$ to represent the mean value of $\bar{P}_\mu$ obtained by averaging over all the template patterns. Figure 4(a) plots $\bar{P}$ against ds for several values of c in the case of $N=1000$ and $p=30$. The short-dashed lines give the results obtained by the original MC-adaptation rule, and the solid lines are obtained by the modified design procedure.

For each c, one may notice that $\bar{P}$ remains 1 when ds is small and declines quickly once ds exceeds a threshold value, denoted as ds_m. Therefore, one can estimate the states that are attracted to a template pattern by $\mathcal{V}=\Sigma_{i=1}^{N}\bar{P}C_N^i \approx \Sigma_{i=1}^{Nds_m}C_N^i$. The $\mathcal{V}$ is just the mean size of the attraction basin of a template pattern. As shown in Fig. 4(a), Nds_m is usually much smaller than N, so $\mathcal{V}$ is negligibly small compared with the huge configuration space which has 2^N states. As a result, a random initial state is most likely to be attracted to the chaotic attractor instead of any one of the template patterns. This fact suggest to us to image the relationship between the attraction basins of the template patterns and the attraction basin of the chaotic attractor as some isolated islands embedded into a chaotic sea. This property is much suitable for pattern recognition.

First, one can control the size of the attraction basins of the template patterns through the parameter c. In this way one can control the recognition capability of such a neural network.

In the terminology of pattern recognition, ds may be interpreted as the degree of difference between a template pattern and a pattern to be identified; $\bar{P}$ may be interpreted as the recognition rate of the template pattern with a certain disturbance ds. The results shown in Fig. 4(a) mean that a disturbed template pattern will be surely recognized if $ds < ds_m(c)$; otherwise, it may be rejected. The ds_m depends on c. The bigger the c, the larger the ds_m, and thus the better the recognition capability.

Since the chaos phase is restricted in the range of $c \leqslant c_1$, beyond which the memory phase appears, the maximum of ds_m is determined by c_1. For the neural network designed by the original MC-adaptation rule with $N=1000$, Fig. 1 shows $c_1 \sim 26$ and Fig. 4(a) indicates $ds_m(c_1) \sim 0.07$. It implies that this network can only recognize patterns with no more than a 7% difference from a template pattern as the template pattern. This limits the application scope of this type of neural networks.

We have shown that applying the modified MC-adaptation rule can extend c_1 from $c_1=26$ to $c_1=36$ in the case of $N=1000$. Further calculations indicate that ds_m depends on c as $ds_m \sim e^{\gamma c}$, which implies that a tiny increment of c_1 may lead to a remarkable increase of ds_m. Therefore, the modified MC-adaptation rule would remarkably enlarge the controllable range of ds_m. Figure 4(a) shows that this is the case: the maximum of ds_m is enlarged to $ds_m(c_1) \sim 0.18$. It means that the system can recognize a pattern as a template pattern even if it has a 18% difference from this template pattern. This is sufficient for most practical applications.

Second, the huge attraction basin of the chaotic attractor provides a credible mechanism of rejection. Rejection means that when a pattern does not belong to any one of the template patterns the system should reject it as an unknown one. This is crucial for certain applications, such as the identification of clients in bank services.

In our neural networks, if the pattern to be identified is a disturbed version of a template pattern, it has two possibilities. When the degree of the disturbance is small—i.e., $ds < ds_m$—the pattern can be recognized as the template pattern. When the disturbance is big—i.e., $ds > ds_m$—the pattern still has chance to be attracted to the template pattern and thus be recognized; if the pattern is not attracted to the template pattern, it will wander in the chaotic attractor for ever. This can be practically guaranteed because for other template patterns this disturbed pattern is equivalent to a random initial state. If the pattern to be identified is an unknown pattern whose template pattern has not be stored as a fixed-point attractor of the network, it corresponds to a random initial state for all of the stored template patterns. As a random initial state, it has $P_{total}=0$ in the chaos phase. Therefore, these patterns have no chance of being attracted to any one

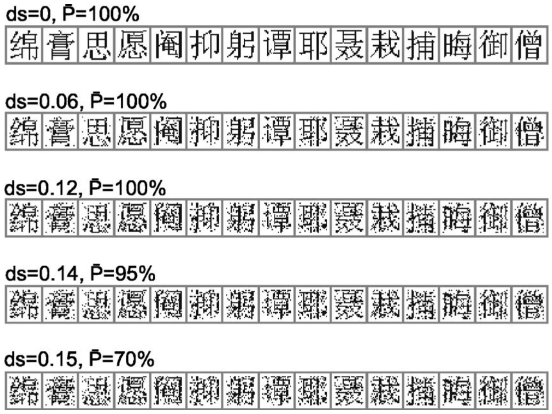

FIG. 5. An example of Chinese character recognition.

of the template patterns and thus result in wrong recognition. In other worlds, a neural network designed by the MC-adaptation rule realizes the rejection by attracting these patterns to the chaotic attractor with the probability of 100%.

Whether an initial state has converged to a template pattern or wanders in the chaotic attractor can be easily judged by calculating the overlap between two continuous states $\{s_i(t-1)\}$ and $\{s_i(t)\}$ of the trajectory. If the overlap satisfies $m=1$, one may affirm that the system reaches a fixed-point attractor. Since there are no unwanted fixed-point attractors in the chaos phase, this attractor must be a template pattern. If the overlap still varies after a sufficiently long time of evolution, one may judge that the system is wandering in the chaotic attractor. For practical applications, a time threshold of about $t \sim 10$ is large enough to make such a judgement.

Another relevant problem is the size effect. For pattern recognition, it is worth studying the size-dependent behavior of $ds_m(c_1)$ as a function of the system size N. Figure 4(b) illustrates the results for $N=576$, 1000, and 1500 with $p=15$, 30, and 45, respectively. In this case, the neural networks have the approximate storage ratio: i.e., $p/N \approx 0.03$. The c_1 is found to be 26, 36, and 42 for $N=576$, 1000, and 1500, respectively. It can be seen that $ds_m(c_1)$ increases with an increase of N. At $N=1500$, $ds_m(c_1) \approx 0.21$. This is a desired property in pattern recognition, since it implies that one can improve the recognition capacity by making use of a larger neural network.

In order to impressively show the feasibility of using attractor neural networks for pattern recognition, as an example we apply the modified MC-adaptation rule to printed Chinese character recognition. About 3775 Chinese characters are commonly applied in documents, and each of them is described as a black and white picture by a 24×24 matrix with the elements of +1 and −1 representing black and white. Then, each character can be coded by a 576-dimension vector. The first row of Fig. 5 shows 15 Chinese characters picked up randomly from the 3775 characters, which will serve as the template patterns in our experiment. We then designed a neural network with $N=576$ by using the modified MC-adaptation rule with $c=c_1$ to store the 15 characters as fixed-point attractors.

We exam the recognition rate $\bar{P}$ as a function of the disturbance degree ds and show the result in Fig. 4(b). It is almost exactly consistent with the result obtained by using the randomly selected patterns as the template patterns. In the other rows of Fig. 5, we show samples with different disturbance degrees. In the case of $ds < 0.14$ all samples are recognized correctly; one may even notice that the samples may be disturbed dramatically. In the case of $ds \geqslant 0.14$, the recognition rate decreases quickly. More importantly, our calculations indicate that when a sample is disturbed too much to be attracted to the corresponding template pattern, it will wander in the chaotic attractor for ever. We have never observed that such a sample is attracted to other template patterns and thus result in a wrong recognition.

V. CONCLUSION

The neural networks designed by the MC-adaptation rule with chaos phase are suitable for pattern recognition. Compared with traditional strategies, they exhibit some unique advantages. One is the presence of the chaotic attractor which provides a natural mechanism of rejection. By this mechanism, the system will never recognize unknown patterns and disturbed template patterns as a template pattern. In other words, these patterns have no chance of being attracted to any template pattern and result in a wrong recognition. This is the case since these patterns correspond to the random initial states and for the random state it has $P_{total}=0$ in the chaos phase. The other one is that the recognition capacity of the neural network is controllable through the parameter c. In this way, one may preset the recognition capacity of a neural network.

The turning point c_1 is crucial for pattern recognition. It determines the range of the chaos phase, so as to determine the maximal recognition capacity of the neural networks. The modified version of the MC-adaptation rule is proved to be efficient for enlarging the parameter range of the chaos phase and thus enlarges the domains of using such neural networks for pattern recognition. Our investigations of size-dependent behavior show that the recognition capacity can be further improved by increasing the size of the networks. For this purpose, one may expect that the recognition capacity can be further improved by releasing the limitation of $|J_{ij}|=1$. The example of printed Chinese character recognition indicates the practical application possibilities of the neural networks designed by our method.

ACKNOWLEDGMENTS

This work is supported in part by the National Natural Science Foundation of China under Grant No. 10475067, the Major State Research Development 973 Project of Nonlinear Science in China and the Doctor Education Fund of the Educational Department of China.

[1] A. K. Jain, R. P. W. Duin, and Mao Jianchang, IEEE Trans. Pattern Anal. Mach. Intell. **22**, 4 (2000).
[2] Yi Sun, IEEE Trans. Signal Process. **48**, 2105 (2000).
[3] Harold H. Szu, X. Y. Yang, Brian A. Telfer, and Yunlong Sheng, Phys. Rev. E **48**, 1497 (1993).
[4] George Nagy, IEEE Trans. Audio Electroacoust. **203**, 428 (1968).
[5] Frank C. Hoppensteadt and Eugene M. Izhikevich, Phys. Rev. E **62**, 4010 (2000).
[6] Keith L. Peterson, Phys. Rev. A **41**, 2457 (1990).
[7] J. M. Carison, J. S. Langer, and B. E. Shaw, Rev. Mod. Phys. **66**, 657 (1994).
[8] R. Palaniappan, IEE Proc.-A: Sci., Meas. Technol. **151**, 16 (2004).
[9] H. Zhao, Phys. Rev. E **70**, 066137 (2004).
[10] Andrzej Nowak, Maciej Lewenstein, and Wojciech Tarkowski, Phys. Rev. E **48**, 4091 (1993).
[11] J. L. van Hemmen and A. C. D. van Enter, Phys. Rev. A **34**, 2509 (1986).
[12] Z. Tan and M. K. Ali, Phys. Rev. E **58**, 3649 (1998).
[13] Ralf Schutzhold, Phys. Rev. A **67**, 062311 (2003).
[14] R. Stoop, J. Buchli, G. Keller, and W.-H. Steeb, Phys. Rev. E **67**, 061918 (2003).
[15] David Horn and Assaf Gottlieb, Phys. Rev. Lett. **88**, 018702 (2001).
[16] Toshiaki Aida, Phys. Rev. E **64**, 056128 (2001).
[17] Alan S. Perelson and Gerard Weisbuch, Rev. Mod. Phys. **69**, 1219 (1997).
[18] Rocio Alaiz-Rodriguez, Alicia Guerrero-Curieses, and Jesus Cid-Sueiro, Pattern Recogn. **38**, 29 (2005).
[19] H. Gutfreund and M. Mezard, Phys. Rev. Lett. **61**, 235 (1988).
[20] B. A. Huberman and T. Hogg, Phys. Rev. Lett. **52**, 1048 (1984).
[21] J. J. Hopfield, D. I. Feinsein, and R. G. Palmer, Nature (London) **304**, 158 (1983).
[22] G. C. Vasconcelos, M. C. Fairhurst, and D. L. Disset, Pattern Recogn. Lett. **16**, 207 (1995).
[23] F. R. Waugh, C. M. Marcus, and R. M. Westervelt, Phys. Rev. Lett. **64**, 1986 (1990).
[24] J. L. van Hemmen and R. Kuhn, Phys. Rev. Lett. **57**, 913 (1986).
[25] P. N. McGraw and Michael Menzinger, Phys. Rev. E **67**, 016118 (2003).
[26] P. R. Krebs and W. K. Theumann, Phys. Rev. E **60**, 4580 (1999).
[27] D. J. Amit, H. Gutfreund, and H. Sompolinsky, Phys. Rev. A **32**, 1007 (1985); Phys. Rev. Lett. **55**, 1530 (1985).
[28] H. Gutfreund, J. D. Reger, and A. P. Young, J. Phys. A **21**, 2775 (1988).
[29] Brigitte Quenet and David Horn, Neural Comput. **15**, 309 (2003).
[30] David Horn, Gideon Dror, and Brigitte Quenet, IEEE Trans. Neural Netw. **15**, 1002 (2004).