Ideal Chaotic Pattern Recognition Is Achievable: The Ideal-M-AdNN - Its Design and Properties

Ke Qin[1,⋆] and B. John Oommen[2,⋆⋆]

[1] School of Computer Science and Engineering
University of Electronic Science & Technology of China
Chengdu, China, 610054
qinke@uestc.edu.cn
[2] School of Computer Science, Carleton University,
1125 Colonel By Dr., Ottawa, ON, K1S 5B6, Canada
oommen@scs.carleton.ca

Abstract. This paper deals with the relatively new field of designing a *Chaotic* Pattern Recognition (PR) system. The benchmark of such a system is the following: First of all, one must be able to train the system with a set of "training" patterns. Subsequently, as long as there is no testing pattern, the system must be chaotic. However, if the system is, thereafter, presented with an unknown testing pattern, the behavior must ideally be as follows. If the testing pattern is not one of the trained patterns, the system must continue to be chaotic. As opposed to this, if the testing pattern is truly one of the trained patterns (or a noisy version of a trained pattern), the system must switch to being *periodic*, with the specific trained pattern appearing periodically at the output. This is truly an ambitious goal, with the requirement of switching from chaos to periodicity being the most demanding. Some related work has been done in this regard. The Adachi Neural Network (AdNN) [1–5] has properties which are pseudo-chaotic, but it also possesses *limited* PR characteristics. As opposed to this, the Modified Adachi Neural Network (M-AdNN) proposed by Calitoiu *et al* [6], is a fascinating NN which has been shown to possess the required periodicity property desirable for PR applications. However, in this paper, we shall demonstrate that the PR properties claimed in [6] are not as powerful as originally reported. Indeed, the claim of the authors of [6] is true, in that it resonates periodically for *trained* input patterns. But unfortunately, the M-AdNN also resonates for unknown patterns and produces these unknown patterns at the output periodically. However, we describe how the parameters of the M-AdNN for its weights, steepness and external inputs, can be specified so as to yield a new NN, which we shall refer to as the Ideal-M-AdNN.

⋆ The work of this author was partially supported by grant No. 2012HH0003 and 9140A17060411DZ02.

⋆⋆ *Chancellor's Professor* ; *Fellow : IEEE* and *Fellow : IAPR*. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. The work of this author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada.

N.T. Nguyen (Ed.): Transactions on CCI XI, LNCS 8065, pp. 22–51, 2013.

Using a rigorous Lyapunov analysis, we shall analyze the chaotic properties of the Ideal-M-AdNN, and demonstrate its chaotic characteristics. Thereafter, we shall verify that the system is also truly chaotic for untrained patterns. But most importantly, we demonstrate that it is able to *switch to being periodic* whenever it encounters patterns with which it was trained. Apart from being quite fascinating, as far as we know, the theoretical and experimental results presented here are both unreported and novel. Indeed, we are not aware of any NN that possesses these properties!

Keywords: Chaotic Neural Networks, Chaotic Pattern Recognition, Adachi-like Neural Networks.

1 Introduction

Pattern Recognition (PR) has numerous well-established sub-areas such as statistical, syntactic, structural and neural. The field of *Chaotic* PR is, however, relatively new and is founded on the principles of chaos theory. It is also based on a distinct phenomenon, namely that of *switching* from *chaos* to *periodicity*. Indeed, Freeman's clinical work has clearly demonstrated that the brain, at the individual neural level and at the global level, possesses chaotic properties. He showed that the quiescent state of the brain is chaos. However, during perception, when attention is focused on any sensory stimulus, the brain activity becomes more periodic [7].

If the brain is capable of displaying both chaotic and periodic behavior, the premise of this paper is that it is expedient to devise artificial Neural Network (NN) systems that can display these properties too. Thus, the primary goal of *chaotic* PR is to develop a system which mimics the brain to achieve chaos and PR, and to consequently develop a new PR paradigm in itself.

Philosophically, the question of whether we can achieve *chaotic* PR by using non-dynamic methods seems to have an unequivocal negative response. The subsequent question of whether we can accomplish the goal of designing a *chaotic* PR system with a *single* computing element (or neuron) also seems improbable. It appears as if we have to resort to a *network* of neurons, and any such network which possesses *chaotic* properties is referred to as a Chaotic Neural Network (CNN). The fundamental issue, really, is whether we can drive such a NN from chaos to periodicity and *vice versa* by merely controlling the input patterns.

At a very fundamental level, the field of NNs deals with understanding the brain as an information processing machine. Conversely, from a computational perspective, it concerns utilizing the knowledge of the (human) brain to build more intelligent computational *models* and computer systems. Thus, NNs have been widely and successfully applied to an ensemble of information processing problems, and the areas of PR and forecasting are rather primary application domains.

Before we proceed, it is fitting for us to submit a consistent terminology, i.e., to distinguish between systems that possess Associative Memory (AM) and

Pattern Recognition (PR). The reader should note that our primary concern is the latter. The literature reports that AM has two forms: auto-association and hetero-association. In auto-association, a NN is required to store a set of patterns. Whenever a single pattern or its "distorted" version is presented to the NN, a system possessing auto-association should be able to recall the particular pattern correctly. Hetero-association differs from auto-association in that an arbitrary set of input patterns is paired with another arbitrary set of output patterns. As opposed to these, the phenomenon of PR is mainly concerned with "classification".

CNNs which also possessed a *weak* form of PR were first proposed by Adachi and his co-authors [1–5]. It would be fair to state (and give them the honor) that they pioneered this "new" field of CNNs. In their papers, by modifying the discrete-time neuron model of Caianiello, Nagumo and Sato, they designed a novel and simple neural model possessing chaotic dynamics, and an Artificial Neural Network (ANN) composed of such chaotic neurons. Their experimental results demonstrated that such a CNN (referred to the AdNN in this paper) possesses both AM and PR properties, as will be illustrated in Section 2.

Historically, the initial and pioneering results concerning these CNNs were presented in [1–5]. In the next year, the author of [8] proposed two methods of controlling chaos by introducing a small perturbation in continuous time, i.e., by invoking a combined feedback with the use of a specially-designed external oscillator or by a delayed self-controlling feedback without the use of any external force. The reason for the introduction of this perturbation was to stabilize the unstable periodic orbit of the chaotic system. Subsequently, motivated by the work of Adachi, Aihara and Pyragas, various types of CNNs have been proposed to solve a number of optimization problems (such as the Traveling Salesman Problem, (TSP)), or to obtain AM and/or PR properties. Later, the author of [9] proposed a NN model based on the Globally Coupled Map (GCM) derived by modifying the Hopfield network. It was reported that this model was capable of information processing and solving the TSP. Chen and Aihara reported a new method referred to as chaotic simulated annealing based on a NN model with transient chaos properties [10]. They also numerically investigated the transiently chaotic neurodynamics with examples of a single neuron model and the TSP. Their conclusion was that this new model possessed a higher searching ability for solving combinatorial optimization problems when compared to both the Hopfield-Tank approach and a stochastic simulated annealing scheme. An interesting step in this regard was the work reported in [11], where the authors utilized the delayed feedback and the Ikeda map to design a CNN to mimic the biological phenomena observed by Freeman [7].

From a different perspective, in [12–14], Hiura and Tanaka investigated several CNNs based on a piecewise sine map or the Duffing's equation to solve the TSP. Their results showed that the latter yielded a better performance than the former.

More recently, based on the AdNN, Calitoiu and his co-authors made some interesting modifications to the basic network connections so as to obtain PR

properties and "blurring". In [15], they showed that by binding the state variables to those associated with *certain* states, one could obtain PR phenomena. However, by modifying the manner in which the state variables were bound, they designed a newly-created machine, the so-called Mb-AdNN, which was also capable of justifying "blurring" from a NN perspective.

While all of the above are both novel and interesting, since most of these CNNs are *completely*-connected graphs, the computational burden is rather intensive. Aiming to reduce the computational cost, in our previous paper [16], we proposed a mechanism (the Linearized AdNN (L-AdNN)) to reduce the computational load of the AdNN. This was achieved by using a spanning tree of the complete graph, and invoking a gradient search algorithm to compute the edge weights, thus minimizing the computation time to be linear.

Although it was initially claimed that the AdNN and M-AdNN possessed "pure" (i.e., periodic) PR properties, in actuality, this claim is not as precise as the authors claimed. We shall now clarify this. As explained in detail in [16, 17], if the AdNN weights the external inputs with the quantities determined by $a_i = 2 + 6x_i$, the properties of the system become closer to mimicing a PR phenomenon. Adachi *et al.* claimed that such a NN yields the stored pattern at the output periodically, with a short transient phase and a small periodicity. However, it turns out that if *untrained* patterns are presented to the system, they also appear periodically, although with a longer periodicity. Besides, the untrained input patterns can lead to having the system occasionally output *trained* patterns. Although the former can be considered to be a PR phenomenon, the latter is a handicap because we would rather prefer the system to be chaotic for all untrained patterns - which the AdNN cannot achieve for these settings. This phenomenon is also pertinent for the M-AdNN, i.e., the output can be periodic for both trained and untrained inputs.

To complete this historical overview, we mention that in [16], we showed that the AdNN goes through a spectrum of characteristics (i.e., AM, *quasi*-chaotic, and PR) as one of its crucial parameters, α, changes. In particular, it is even capable of recognizing masked or occluded patterns.

The primary aim of this paper is to show that the M-AdNN, when tuned appropriately, is capable of demonstrating ideal PR capabilities. Indeed, we shall argue that this NN, referred to as the Ideal-M-AdNN:

- Can be trained by a set of patterns. This will determine the weights of the CNN.
- When the Ideal-M-AdNN is presented with a trained pattern (or its noisy version) at the input, it resonates and yields *only this* pattern at the output periodically. We emphasize that, in this case, no other trained pattern is observed at the output.
- When the Ideal-M-AdNN is presented with an untrained pattern at the input, it continues to behave chaotically. In this case, none of the trained patterns is observed at the output - thus yielding the gold standard of *chaotic* PR.

One should remember that the method for setting $\{w_{i,j}\}$ and $\{a_i\}$ were already specified for the AdNN in the paper by Adachi and Aihara. But the reader should remember that we are not working with the AdNN, but rather with the M-AdNN, in which the state bindings are significantly different. Apart from making the changes in the structure, the authors of the M-AdNN also made changes in the assignment strategies. Our task in this paper is: (a) To demonstrate that these latter changes are not warranted; (b) To argue that in the paper due to Adachi and Aihara, the choice of $a = 2$ is not the best value that can yield AM properties; and (c) To demonstrate that the value of one of its parameters, ε (explained presently) that was used, is just a *single* one of the choices that can force the system to be chaotic.Thus, **the primary contributions of the paper** are:

1. We formalize the requirements of a PR system which is founded on the theory of chaotic NNs;
2. We enhance the M-AdNN to yield the Ideal-M-AdNN, so that it does, indeed, possess *Chaotic* PR properties;
3. We show that the Ideal-M-AdNN does switch from chaos to periodicity when it encounters a trained pattern, but that it is truly chaotic for all other input patterns;
4. We present results concerning the stability of the network and its transient and dynamic retrieval characteristics. This analysis is achieved using eigenvalue considerations, and the Lyapunov exponents;
5. We provide explicit experimental results justifying the claims that have been made.

We conclude by mentioning that in this paper, we were not so concerned about the application domain. Rather, our work is intended to be of an investigatory sort.

2 State of the Art

The AdNN and Its Variants: The AdNN is a network of neurons with weights associated with the edges, a well-defined Present-State/Next-State function, and a well-defined State/Output function. It is composed of N neurons which are topologically arranged as a completely connected graph. A neuron, identified by the index i, is characterized by two internal states, $\eta_i(t)$ and $\xi_i(t)$ $(i = 1...N)$ at time t, whose values are defined by Equations (1) and (2) respectively. The output of the i^{th} neuron, $x_i(t)$, is given by Equation (3), which specifies the so-called *Logistic Function*.

$$\eta_i(t+1) = k_f \eta_i(t) + \sum_{j=1}^{N} w_{ij} x_j(t), \tag{1}$$

$$\xi_i(t+1) = k_r \xi_i(t) - \alpha x_i(t) + a_i. \tag{2}$$

$$x_i(t+1) = f(\eta_i(t+1) + \xi_i(t+1)). \tag{3}$$

where $f(\cdot)$ is defined by $f(x) = 1/(1+e^{-x})$. Under certain settings, the AdNN can behave as a dynamic AM. It can dynamically recall all the memorized patterns as a consequence of an input which serves as a "trigger". If the external stimulations correspond to trained patterns, the AdNN can also behave like a PR system, although with some limitations and weaknesses, as will be explained below, and as also illustrated in [18].

By invoking a Lyapunov Exponents (LE) analysis, one can show that the AdNN has $2N$ *negative* LEs. In order to obtain a system with *positive* LEs, Calitoiu *et al* [15] proposed a model of CNNs, called the M-AdNN, which modifies the AdNN to enhance its PR capabilities.

The fundamental difference between the AdNN and the M-AdNN in terms of their Present-State/Next-State equations is that the latter has only a *single* global neuron (and its corresponding two global states) which is used for the state updating criterion for *all* the neurons. Thus, $\eta_i(t)$ and $\xi_i(t)$ are updated as per:

$$\eta_i(t+1) = k_f \eta_m(t) + \sum_{j=1}^{N} w_{ij} x_j(t) \tag{4}$$

$$\xi_i(t+1) = k_r \xi_m(t) - \alpha x_i(t) + a_i. \tag{5}$$

$$x_i(t+1) = f(\eta_i(t+1) + \xi_i(t+1)). \tag{6}$$

Observe that at time $t+1$, the global states are updated with the values of the states of *the single* neuron, $\eta_m(t)$ and $\xi_m(t)$. The value of m is set to be N. This is in contrast to the AdNN in which the updating at time $t+1$ uses the internal state values of *all* the neurons at time t. The weight assignment rule for the M-AdNN is the classical variant $w_{ij} = \frac{1}{p}\sum_{s=1}^{p} x_i^s x_j^s$. This is in contrast to the AdNN which uses $w_{ij} = \frac{1}{p}\sum_{s=1}^{p}(2x_i^s - 1)(2x_j^s - 1)$. Additionally, Calitoiu *et al.* set $a_i = x_i^s$, as opposed to the AdNN in which $a_i = 2 + 6x_i^s$. Thus, as the authors demonstrated, the M-AdNN will be more "receptive" to external inputs, leading to a chaotic PR system. However, we will illustrate that the parameters Calitoiu *et al.* set were not the appropriate ones to yield PR, and show how these settings must be tuned to yield a more superior PR system. This will be done in Sections 3.1 and 3.2.

We list below the relevant salient features of the M-AdNN in the interest of completeness and continuity:

1. Being a variant of the AdNN, the M-AdNN is topologically a completely-connected NN;
2. The main difference between the M-AdNN and the classical auto-AM model is that unlike the latter, the M-AdNN was seen to be chaotic, this behavior being a consequence of the dynamics of the underlying system. The M-AdNN seems to be more representative of the way by which the brain achieves PR;

3. The most significant difference between the AdNN and the M-AdNN is the way by which the values of the internal state(s) are updated. The M-AdNN uses two global internal states, both of which are associated with a *single* neuron. By using these global states, the transient phase is reduced to be approximately 30 [6], which is in contrast to the AdNN's transient phase – which can be as large $21,000$;
4. The authors of [6] show that the largest Lyapunov exponent of the AdNN is $\lambda = \log k_r < 0$ when $k_r = 0.9$, which implies that AdNN is not really chaotic. In contrast, the M-AdNN has a positive largest Lyapunov exponent given by $\lambda = \frac{1}{2}\log N + \log k_r$, rendering it chaotic.
5. The experimental results reported in [6] claimed that the M-AdNN responds periodically for a trained input pattern. However, it turns out that this periodic output occurs for both trained and untrained patterns, as can be seen in [16].

3 The Ideal-M-AdNN

Traditionally, PR systems work with the following model: Given a set of training patterns, a PR system learns the characteristics of the class of the patterns, and retains this information either parametrically or non-parametrically. When a testing sample is presented to the system, a decision of the identity of the sample class is made using the corresponding "discriminant" function, and this class is "proclaimed" by the system as the identity of the pattern.

The goal of the field of Chaotic PR systems can be expressed as follows: We do not intend a chaotic PR system to report the identity of a testing pattern with such a "proclamation". Rather, what we want to achieve for a the chaotic PR system are the following phenomena:

- The system must yield a strong *periodic* signal when a trained pattern is to be recognized.
- Further, between two consecutive recognized patterns, none of the trained patterns must be recalled.
- On the other hand, and most importantly, if an untrained pattern is presented, the system must be chaotic.

Calitoiu *et al* were the first researchers who recorded the potential of chaotic NNs to achieve PR. But unfortunately, their model, as presented in [6], named the M-AdNN, was not capable of demonstrating *all* the PR properties mentioned above. To be more accurate:

- For trained patterns, the M-AdNN reacts exactly as what we want: The specific input patterns appear at the output *periodically*.
- However, for untrained patterns, the M-AdNN still yields the unexpected *untrained* pattern periodically.

– In summary, the M-AdNN yields at the output whatever is presented to the system as its input[1], and therefore we claim that the M-AdNN is not good enough for PR since it is not able to resonate *only* for the trained patterns.

Nevertheless, the M-AdNN is still a fascinating network since it can produce periodic output while it is still in the midst of chaotic iterations, which, since it was previously unreported in the literature, is really a ground-breaking work in the area of chaotic NNs. Based on the results of Calitoiu *et al*, we improve the M-AdNN for it to truly possess *Chaotic* PR properties. The new model proposed here is referred as the "Ideal-M-AdNN".

The topology of the Ideal-M-AdNN is exactly the same as that of the M-AdNN, which is, on deeper consideration, seen to actually also be a Hopfield-like model. Structurally, it is also composed of N neurons, topologically arranged as a completely connected graph. Again, each neuron has two internal states $\eta_i(t)$ and $\xi_i(t)$, and an output $x_i(t)$. Just like the M-AdNN, the Present-State/Next-State equations of the Ideal-M-AdNN are defined in terms of only a *single* global neuron (and its corresponding two global states), which, in turn, is used for the state updating criterion for *all* the neurons. Thus, $\eta(t)$ and $\xi(t)$ are updated as per:

$$\eta_i(t+1) = k_f \eta_m(t) + \sum_{j=1}^{N} w_{ij} x_j(t), \tag{7}$$

$$\xi_i(t+1) = k_r \xi_m(t) - \alpha x_i(t) + a_i, \tag{8}$$

$$x_i(t+1) = f(\eta_i(t+1) + \xi_i(t+1)), \tag{9}$$

where m is the index of this so-called "global" neuron.

We shall now concentrate on the differences between the two models, which are the parameters: $\{w_{ij}\}$, ε and a_i. In the work reported in [6], these parameters were arbitrarily set to be $w_{ij} = \frac{1}{4}\Sigma_{s=1}^{4} x_i^s x_j^s$, $\varepsilon = 0.00015$ and $a_i = x_i$. In this paper, we will show that these values must not be set arbitrarily. Rather, we shall address the issue of how these parameters must be assigned their respective values so as to yield all the properties required of a *Chaotic* PR system.

3.1 The Weights of the Ideal-M-AdNN

In the interest of clarity, we first argue that the weights assignment Equation (10) for the M-AdNN is not appropriate. We rather advocate another form of the Hebbian rule given by Equation (15) instead. The reason is explained as below: As we have observed above, the M-AdNN uses a form of the Hebbian rule to determine the weights of the connections in the network. This rule is defined by the following equation:

[1] This phenomenon was not observed nor reported in [6].

$$w_{ij} = \frac{1}{p} \sum_{s=1}^{p} P_i^s P_j^s, \tag{10}$$

where $\{P\}$ are the training patterns, P_i^s denotes the i^{th} neuron of the s^{th} pattern P^s, and p is the number of training patterns.

At this juncture, we emphasize that the Hebbian rule, Equation (10), is founded on one fundamental premise: Any pair of learning vectors, P and Q, must be orthogonal, and thus, for all P and Q, if P^T is the transpose of P:

$$P^T Q = \begin{cases} 0, (P \neq Q) \\ N, (P = Q). \end{cases} \tag{11}$$

The reasons why the formula given by Equation (10) is based on the above, is because of the following:

If P denotes an N-by-1 vector (e.g, the $P1$ of Figure 8 (a) is a 100-by-1 vector), Equation (10) can be rewritten as:

$$W = \frac{1}{p} \sum_{s=1}^{p} P^s (P^s)^T. \tag{12}$$

Thus, the output for any given input vector P^k is:

$$\begin{aligned} O = W P^k &= \frac{1}{p} \sum_{s=1}^{p} P^s (P^s)^T P^k \\ &= \frac{1}{p} \sum_{s=1}^{p} P^s \left[(P^s)^T P^k \right] = \frac{N}{p} P^k \end{aligned} \tag{13}$$

As per Equation (11), the output $O = \frac{N}{p} P^k$ is a scalar[2] of the input P^k. Otherwise, if the learning vectors $\{P\}$ are not orthogonal, Equation (13) has the form:

$$O = \frac{1}{p} \sum_{s=1}^{p} P^s [(P^s)^T P^k] = \frac{1}{p} \left(N P^k + P^{noise} \right), \tag{14}$$

where $P^{noise} = \sum_{s=1, s \neq k}^{p} b_s P^s$ and $b_s = (P^s)^T P^k$. Obviously, $O = \frac{N}{p} P^k$ if and only if $P^{noise} = 0$, explaining the rationale for orthogonality.

The need for orthogonality is not so stringent, because, as per the recorded research, when the number of neurons is much larger than the number of patterns, and the learning vectors are randomly chosen from a large set, the probability of having the learning vectors to be orthogonal is very high. Consequently, Equation (11) is true, albeit probabilistically.

Summary: Based on the above observations, we conclude that for the M-AdNN, we should not use the Hebbian rule as dictated by the form given in Equation

[2] Observe that without loss of generality, we can easily force $O = P^k$ instead of $O = \frac{N}{p} P^k$, by virtue of a straightforward normalization.

(10), since the data sets used by both Adachi *et al* and Calitoiu *et al* are defined on $\{0,1\}^N$, and the output is further restricted to be in $[0,1]$ by virtue of the logistic function. We may easily verify that any pair of the given patterns in Figure 8 are NOT orthogonal. In fact, this is why Adachi and Aihara computed the weight by scaling all the patterns to -1 and 1 using the formula given by Equation (15) instead of (10):

$$w_{ij} = \frac{1}{p}\sum_{s=1}^{p}(2P_i^s - 1)(2P_j^s - 1). \tag{15}$$

By virtue of this argument, in this paper, we advocate the use of Equation (15), to determine the network's weights.

It is pertinent to mention that since the patterns P are scaled to be in the range between -1 and 1, it *does* change the corresponding property of orthogonality. Actually, it is very interesting to compare the inner product of the so-called Adachi dataset before and after scaling (please see Table 1). From it we see that the inner products of any pair of *scaled* patterns while not being exactly 0, are very close to 0 – i.e., these patterns are *almost* orthogonal. In contrast, the inner products of the corresponding *unscaled* patterns are very large. Again, this table confirms that Equation (15) is a more suitable (and reasonable) expression than Equation (10).

Table 1. The inner products of the patterns of the Adachi set. In the left table, all patterns are scaled to -1 and 1 by $2P - 1$. In the table on the right, all the patterns are defined on $\{0,1\}^N$. The inner products in the table on the left are much smaller and almost 0, implying that the patterns are *almost* orthogonal.

	P1	P2	P3	P4
P1	100	10	10	6
P2	10	100	-4	4
P3	10	-4	100	12
P4	6	4	12	100

	P1	P2	P3	P4
P1	49	27	27	26
P2	27	50	24	26
P3	27	24	50	28
P4	26	26	28	50

3.2 The Steepness, Refractory and Stimulus Parameters

Significance of ε for the AdNN. The next issue that we need to consider concerns the value of the steepness parameter ε of the output function. From Section 2 we see that the output function is defined by the Logistic function $f(x) = \frac{1}{1+e^{-x/\varepsilon}}$, which is a typical sigmoid function.

One can see that ε controls the steepness of the output. If $\varepsilon = 0.01$, then $f(x)$ is a normal sigmoid function. If ε is too small, say, 0.0001, the Logistic function almost degrades to become a unit step function (see Figure 1). The question, really, is one of knowing how to set the "optimal" value for ε. To provide a rationale for determining the best value of ε, we concentrate on the Adachi's neural model [2] defined by:

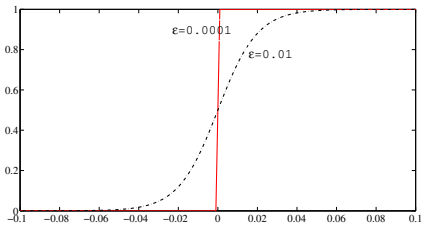


Fig. 1. A graph demonstrating the effect of the parameter ε to change the steepness of the sigmoidal function. The ε for the dash and solid lines are 0.01 and 0.0001 respectively.

$$y(t+1) = ky(t) - \alpha f(y(t)) + a, \tag{16}$$

where $f(\cdot)$ is a continuous differentiable function, which as per Adachi *et al* [2], is the Logistic function[3].

The properties of Equation (16) greatly depend on the parameters k, α, a and $f(\cdot)$. In order to obtain the full spectrum of the properties represented by Equation (16), it is beneficial for us to first consider $f(\cdot)$ in terms of a unit step function, and to work with a fixed point analysis. To do this, we consider a few distinct cases listed below:

1. If the system has only *one* fixed point.

- In this case, obviously, $y(t) = 0$ is not a fixed point of Equation (16) unless $a = 0$.
- If Equation (16) has only one positive fixed point, say y_0, then the value of this point can be resolved from Equation (16) as $y_0 = \frac{a-\alpha}{1-k}$. In the paper by Nagumo and Sato [19], the authors set $\alpha = 1$, $k < 1$, and let a vary from 0 to 1. Apparently, the above value of y_0 is always negative, which contradicts our assumption that y_0 is a positive fixed point. Similarly, we can see that Equation (16) does not have any negative fixed points either. In other words, we conclude that Equation (16) does not have any fixed point.

2. If the system has period-2 points, say y_1 and y_2 (where $y_1 \neq y_2$), we see that these points must satisfy:

$$\begin{cases} y_2 = ky_1 - \alpha f(y_1) + a \\ y_1 = ky_2 - \alpha f(y_2) + a. \end{cases} \tag{17}$$

- If 0 is one of the period-2 points (without loss of generality, $y_1 = 0, y_2 \neq 0$), then we can solve the above equations to yield the two solutions as:

[3] Historically, the original form of this equation initially appeared in the paper by Nagumo and Sato [19]. The only difference between the function that these researchers used and the one discussed here is the function $f(\cdot)$, since in [19], they utilized the unit step function instead of the logistic function.

$$\begin{cases} y_2 = a \\ y_1 = (k+1)a - \alpha. \end{cases}$$

From this, we see that the two points exist if and only if $a \neq 0$ and $\alpha = (1+k)a$.

– If the system has two positive period-2 points (assume $y_1 > 0$, $y_2 > 0$), we see that the system can be rewritten as:

$$\begin{cases} y_2 = ky_1 - \alpha + a \\ y_1 = ky_2 - \alpha + a, \end{cases}$$

which is equivalent to implying that $y_2 - y_1 = k(y_1 - y_2)$. Clearly, this equality cannot hold since $y_1 \neq y_2$ and $k > 0$. In an analogous manner we can conclude that the system does not have two negative period-2 points either.

– If the system has two period-2 points with different signs (without loss of generality we can assume that $y_1 > 0$, $y_2 < 0$), we solve Equation (17) as:

$$\begin{cases} y_1 = \frac{a(k+1)-k\alpha}{1-k^2} \\ y_2 = \frac{a(k+1)-\alpha}{1-k^2}. \end{cases} \tag{18}$$

Since $y_1 > 0$ and $y_2 < 0$, this yields the inequality:

$$\frac{k\alpha}{1+k} < a < \frac{\alpha}{1+k}. \tag{19}$$

In Nagumo's paper [19], the authors set $k = 0.5$ and $\alpha = 1$, implying that the system has period-2 fixed points only when $1/3 < a < 2/3$.
From the above, we know the Nagumo-Sato model does not have any fixed points, but that it rather does have period-2 points. The next question that we have to resolve is whether it possesses chaotic properties? To determine this, we have to investigate what happens to the iteration orbits when the initial point is not exactly on, but close to a period-2 point? Indeed, this question can be answered by considering the inequality:

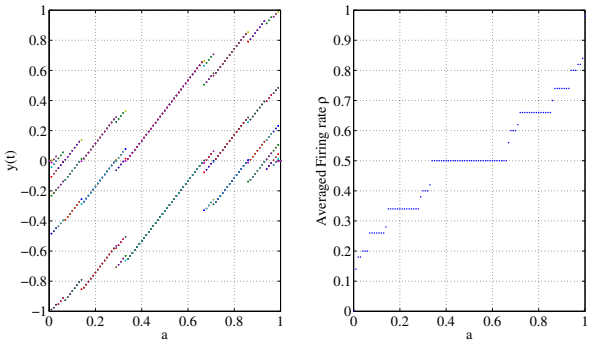
$$\left| \frac{\partial g^{(2)}(y)}{\partial y} \right|_{y=y_1} < 1, \tag{20}$$

where $g(y) = ky - \alpha f(y) + a$, and $f(\cdot)$ is still a unit step function. A simple algebraic manipulation displays that the inequality given by Equation (20) is always true whenever $k < 1$, implying that whatever the initial condition, after a sufficiently large number of iterations, the orbits converge to the period-2 points. In other words, all the period-2 points are attracting, implying that the system does not demonstrate any chaotic phenomena.

3. If we now consider the period-n orbits, one can see that it is not possible to resolve the set of equations that generalize from Equations (18) and (19) for n variables. However, the analysis that corresponds to generalizing the *stability*

equation is exactly the same as in Equation (20), and in every case, one can again show that the inequality is true[4]. This again leads to the conclusion that all the period-n orbits are also attracting. This can be verified by considering the graphs in Figure 2.

The situation is, however, completely different if the unit step function is replaced by the sigmoid function. This is a relevant issue because, as stated earlier, the difference between the Nagumo-Sato model and the Adachi model lies in the output function $f(\cdot)$, where the former uses the unit step function and the latter uses a sigmoid function. The issue is further accentuated because the logistic function may degrade to an unit step function if an inappropriate value of ε is utilized, as shown in Figure 1. Indeed, when ε is very small, the sigmoid function will "degrade" to the unit step function, leading us to the analysis which we have just concluded.



(a) (b)

Fig. 2. The period-n orbits (a) and the averaged firing rate (b) of the Nagumo-Sato model, in which a varies from 0 to 1.

Significance of ε for the M-AdNN. Since the M-AdNN is a modified version of the AdNN (which uses the Logistic function as the output function), setting the value of ε is also quite pertinent in our case. As we shall argue now, in order to enable the system to have chaotic properties, the parameter ε should not be close to zero.

To demonstrate the role of parameter ε of the Adachi model, as before, we analyze the behavior of the model starting from fixed points or period-n points.

Let us first analyze whether the Adachi neuron has any fixed points, i.e., if

$$y(t+1) = y(t) = ky(t) - \alpha f(y(t)) + a$$

[4] This has been done for values of n which equal 3, 4 and 5. The general analysis for arbitrary values of n can also be done using the Schwarz derivative and the period doubling bifurcation theorem. However, this goes beyond the scope of this paper and is thus omitted here.

is ever satisfied. Since this equation is a transcendental equation, obtaining the exact fixed point(s) is far from trivial. Nevertheless, we are still able to achieve some analysis on the bifurcation parameters.

As we know, if a fixed point(s) y^* exists, it should satisfy:

$$\alpha f(y^*) = ky^* - y^* + a.$$

Our first task is to see if a fixed point y^* does exist.

To do this, we compose:

$$F(y) = -\alpha f(y) + (k-1)y + a, \tag{21}$$

which is done so that the root of $F(y)$ corresponds to the fixed point of y.

It is easy to observe that following: For any given negative y, $f(y)$ is almost zero by virtue of its sigmoidal nature. Thus, $F(y) \approx (k-1)y + a > 0$. Similarly, for any given positive y, (for example, y=1), $f(y)$ is almost unity, again by virtue of its sigmoidal nature. Thus, $F(y) \approx -\alpha + k - 1 + a < 0$. Since $F(y)$ is a continuous function over $(-\infty, \infty)$, and since it changes its sign, there must exist at least one y^* satisfying $F(y^*) = 0$ implying that this value of y^* is the fixed point.

The next question is: How many fixed points does the system have? This question can be answered by verifying the monotonicity of Equation (21). Since $f'(y) = -\frac{1}{\varepsilon} f(y)[1 - f(y)]$, we get:

$$F'(y) = -\frac{\alpha}{\varepsilon} f(y)[1 - f(y)] + k - 1. \tag{22}$$

Again, since we always have $f(y)[1-f(y)] \approx 0$, it implies that $F'(y) \approx k-1 < 0$, which, in turn means that $F(y)$ is a decreasing function. As a result of the fact that it is positive on one side of the fixed point, negative on the other side, and simultaneously always has a negative derivative, we conclude that the Adachi model has only *a single* fixed point, y^*.

If y^* is stable, it should satisfy the condition

$$\left| \frac{dy(t+1)}{dy(t)} \right|_{y=y^*} < 1. \tag{23}$$

From the dynamical form of $y(t+1)$, this derivative can be rewritten as

$$\left| k - \frac{\alpha}{\varepsilon} f(y^*)(1 - f(y^*)) \right| < 1, \tag{24}$$

which is equivalent to:

$$\begin{cases} \alpha f^2(y) - \alpha f(y) + \varepsilon(k+1) > 0 \\ \alpha f^2(y) - \alpha f(y) + \varepsilon(k-1) < 0. \end{cases} \tag{25}$$

Since each neuron's output is between 0 and 1, we see that for all values in (0,1) the inequalities given by Equation (25) are always true. This leads us to the discriminants:

$$\begin{cases} \Delta_1 = \alpha^2 - 4\alpha\varepsilon(k+1) < 0 \\ \Delta_2 = \alpha^2 - 4\alpha\varepsilon(k-1) > 0. \end{cases} \tag{26}$$

Obviously, since $k - 1 < 0$, $\alpha > 0$ and $\varepsilon > 0$, the second inequality is always true. Concentrating now on the first inequality, we can solve for ε and see that it has to satisfy:

$$\varepsilon > \frac{\alpha}{4(k+1)}. \tag{27}$$

Briefly, this states that if the fixed point is stable, ε should be greater than $\alpha/4(k+1)$; otherwise, it is not stable.

This condition is experimentally verified in Figures 3 (a), (b) and (c). In our experiment, the parameters were set to be $\alpha = 1, k = 0.5$. In this case, the "tipping point" for ε is $1/6 \approx 0.1667$. As one can very clearly see, if $\varepsilon = 0.18 > 0.1667$, all of the fixed points are stable (Figure 3 (a)); Otherwise, if $\varepsilon = 0.15 < 0.1667$, there exist period doubling bifurcations[5] (Figure 3 (b)). As ε is further decreased, one can observe chaotic windows (Figure 3 (c)).

We conclude this section by emphasizing that ε cannot be too small, for if it were, the Adachi neural model would degrade to the Nagumo-Santo model, which does not demonstrate any chaotic behavior. This is also seen from Figures 4 (a) and (b). In (a), where the parameter ε is 0.015, we can see that the orbit of the iteration is non-periodic, while in (b), in which ε is 0.00015, the orbit of the trajectory is forced to have a periodicity of 5. In fact, we have calculated the whole LE spectrum of the system (16) when it concerns the parameters ε, as shown in Figure 5 (a). In order to get a better view, we have amplified the graph in the area $\varepsilon \in [0, 0.04]$ and in the area $\varepsilon \in [-1.5, 0.5]$ in (b). As one can see when $\varepsilon = 0.015$, the LE is positive, and has a value which is approximately 0.3, which indicates chaos. On the other hand, when $\varepsilon = 0.00015$, the LE is negative. Of course, there are some other optional values which also lead to chaos, such as $\varepsilon = 0.007, 0.008, 0.012$ etc can also lead to chaos.

Indeed, our arguments show that the value of ε as set in [15] to be $\varepsilon = 0.00015$, is not appropriate. Rather, to develop the Ideal-M-AdNN, we still opted to use a value of ε which is *two* orders of magnitude larger, i.e., $\varepsilon = 0.015$.

4 Lyapunov Exponents Analysis of the Ideal-M-AdNN

We shall now analyze the Ideal-M-AdNN, both from the perspective of a *single* neuron and of the entire network.

As is well known, the LEs describe the behavior of a dynamical system. There are many ways, both numerically and analytically, to compute the LEs of a dynamical system. Generally, for systems with a small dimension, the best way is to analytically compute it using its formal definition. As opposed to this, for systems with a high dimension, it is usually not easy to obtain the entire LE spectrum in an analytic manner. In this case, we have several other alternatives to judge whether a system is chaotic. One of these is to merely determine the largest LE (instead of computing the entire LE spectrum) since the existence

[5] In this paper, we analyze the bifurcation in a rather simple way. The reader must note that this could also be achieved by using the Schwarz derivative and the period doubling bifurcation theorem.

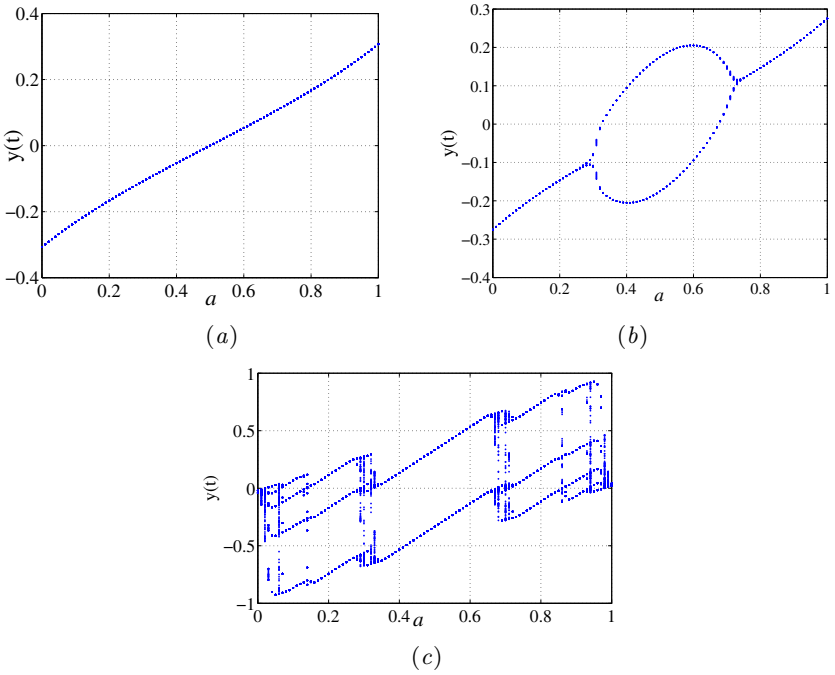


Fig. 3. This figure shows that as the steepness parameter, ε, varies, the dynamics of the Adachi neuron changes significantly. In the three figures, the values of ε are 0.18, 0.15 and 0.015 respectively.

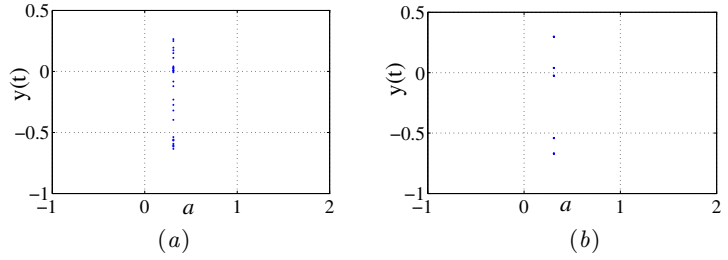


Fig. 4. This figure shows that if the value of ε is too small, it will force the orbit of the trajectory to be periodic (b). The values of ε for (a) and (b) are 0.015 and 0.00015 respectively.

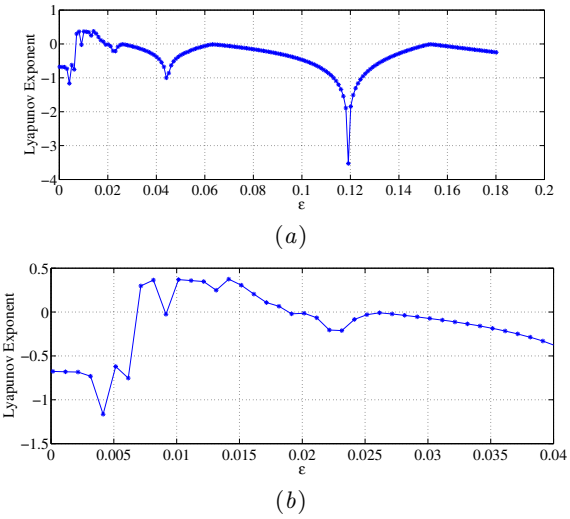


Fig. 5. This figure shows the LE spectrum of the system. Some values between 0.007 and 0.02 may cause the system to be chaotic. The reader should observe that other parametric settings are: $k = 0.5$, $a = 0.31$. The reason why we have used such settings is because one can observe chaotic windows under these conditions, as shown in Figure 3 (c).

of a single positive LE indicates chaotic behavior. In this setting, this is usually achieved by using a numerical scheme as described in Algorithm 1. In this algorithm, the basic idea is to follow two orbits that are close to each other, and to calculate their average logarithmic rate of separation [20–22].

In practice, this algorithm is both simple and convenient if we have the right to access the equations that govern the system. Furthermore, if it is easy to obtain the partial derivatives of the system, we can also calculate the LE spectrum by QR decomposition [20, 23, 22].

We now present the theoretical basis for obtaining the LE spectrum using the QR decomposition. To do this, in the interest of compactness, we use the notation that J_t represents the Jacobian matrix at time t, i.e., $J_t = J(t)$, whose transpose is written as J_t^T. Also, x_0 is assumed to be a randomly chosen starting point, and $\{f(x_0), f_2(x_0), f_3(x_0) \cdots f_t(x_0)\}$ denotes the trajectory obtained as a consequence of iterating as per the system's dynamical equation. Again, to simplify the notation, we shall use the notation that $x_k = f_k(x_0)$.

As we know, the LEs of a network are defined by the eigenvalues of the limit of the matrix[6]

$$\Lambda = \lim_{t \to \infty} [J_t \cdot J_t^T]^{\frac{1}{2t}}. \tag{28}$$

Generally, it is hard to compute the limit of Equation (28), which is why one often invokes the QR decomposition. To achieve this, we note that by the chain rule,

[6] The conditions for the existence of the limit are given by the Oseledec theorem.

Algorithm 1. Numerical Calculation of the Lyapunov Exponent

1: Start with any initial condition in the basin of attraction.
2: Iterate until the orbit is on the attractor.
3: Select a nearby point (separated by d_0, as indicated in [21]. In our work, we set $d_0 = 10^{-12}$).
4: Advance both orbits one iteration and calculate the new separation d_1.
5: Evaluate $\log |d_1/d_0|$ in any convenient base.
6: Readjust one orbit so that its separation is d_0 in the same direction as d_1.
7: Repeat Steps 4-6 many times and calculate the average of Step 5.

$$Df_t(x_0) = J(f_{t-1}(x_0)) \cdots J(f(x_0)) \cdot J(x_0),$$

where $J(x_i) = Df(x)|_{x=x_i} = J(f_i(x_0))$. As clarified in [20, 22], we can rewrite $J_0(x_0)$ using its QR decomposition as $J_0(x_0) = Q_1 R_1$, where Q_1 is orthogonal implying that $Q_1 \cdot Q_1^T = I$. If we define $J_k^* = J(f_{k-1}(x_0))Q_{k-1}$, we can write *its* QR decomposition as $J_k^* = Q_k R_k$. Thus, $J(f_{k-1}(x_0)) \cdot Q_{k-1} = Q_k R_k$. Consequently, we obtain $J(f_{k-1}(x_0)) = Q_k R_k Q_{k-1}^{-1}$. By applying this equation to the chain rule, the differential $Df_t(x_0)$ can be transformed to be:

$$\begin{aligned} Df_t(x_0) &= Q_t R_t Q_{t-1}^{-1} \cdot Q_{t-1} R_{t-1} Q_{t-2}^{-1} \cdots R_1 \\ &= Q_t R_t \cdots R_1. \end{aligned} \tag{29}$$

The LEs, $\{\lambda_i\}$, can then be obtained as:

$$\lim_{t \to \infty} \frac{1}{t} \ln |v_{ii}(t)| = \lambda_i, \tag{30}$$

where $\{v_{ii}(t)\}$ are the diagonal elements of the product $R_{t-1} R_{t-2} \cdots R_1$.

Again, for the reader's convenience, the algorithmic details are formally given in Algorithm 2.

4.1 Numeric Lyapunov Analysis of a Ideal-M-AdNN

We shall now undertake a Lyapunov Analysis of the Ideal-M-AdNN. To do this, we first undertake the Lyapunov analysis of a single neuron in the interest of simplicity. Indeed, it can be easily proven that a single neuron is chaotic when the parameters are properly set.

Consider a primitive component of the Ideal-M-AdNN, where the model of a *single* neuron can be described as[7]:

$$\eta(t+1) = k_f \eta(t) + wx(t), \tag{31}$$

$$\xi(t+1) = k_r \xi(t) - \alpha x(t) + a, \tag{32}$$

$$x(t+1) = \frac{1}{1 + e^{-(\eta(t+1) + \xi(t+1))/\varepsilon}}. \tag{33}$$

[7] It should be observed that $w_{ii} = 1, i = 1, 2, \cdots, N$ for the entire network. For a single neuron, the value of w should be $w = 1$. We should observe that, for a primitive neuron, there is no difference between the AdNN, the M-AdNN and the Ideal-M-AdNN.

Algorithm 2. Calculation of the Lyapunov Exponent by QR Decomposition

1: Choose a starting point x_0 and compute the Jacobian matrix $J_0(x_0) = \frac{df(x)}{dx}$ at x_0.
2: Decompose $J_0(x_0)$ by the QR decomposition: $J_0(x_0) = Q_1 \cdot R_1$ where Q_1 is an orthogonal matrix and R_1 is an upper triangular matrix. Let $\Upsilon = R_1$.
3: Compute the Jacobian matrix at x_1, $J_1(x_1) = \frac{df(x_1)}{dx_1}$, where $x_1 = f(x_0)$.
4: Let $J_2^* = J_1 \cdot Q_1$ and decompose J_2^* using the QR decomposition: $J_2^* = Q_2 \cdot R_2$ where Q_2 is also an orthogonal matrix and R_2 is an upper triangular matrix. Let $\Upsilon = R_2 \cdot \Upsilon$.
5: Repeat Steps 3 and 4 $t-1$ times till we finally obtain $\Upsilon = R_t R_{t-1} \cdots R_1$.
6: The LE spectrum is defined by the logarithm of the diagonal elements υ_{ii} of Υ: $\lambda_i = \frac{1}{t} \log |\Upsilon_{ii}|$.

As we know from the records of [2, 5], the external stimulus a is considered as a constant, i.e., $a = 2$. In [2, 5], the authors have shown the LE spectrum of Equation (16) while a varies from 0 to 1. However, for a single Adachi neuron, the role of a has not yet been discussed. As a matter of fact, most of papers which concern the Lyapunov analysis of the AdNN and it's variants focus on the parameter k_f and k_r [2, 15, 6, 18, 24]. Indeed, the role of α and a have never been discussed seriously in the literature.

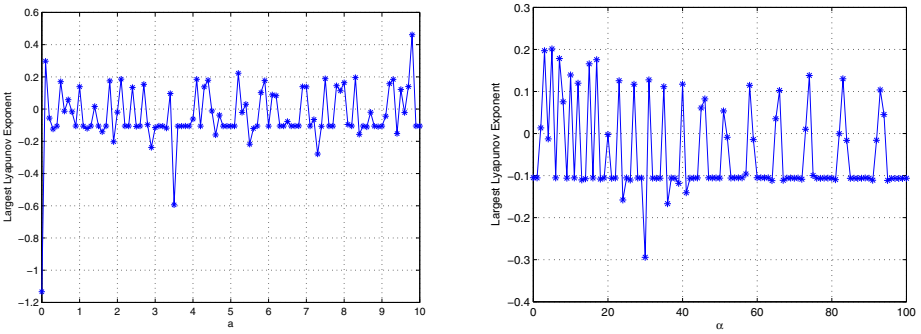
Figures 6 (a) and (b) show the largest LE spectrum as obtained by Equations (31)-(33). We can clearly observe that the largest LE fluctuates sharply with a and α: For some values of a and α, the largest LE is positive and for others it is negative. From this, we can understand that in all of the papers [2, 15, 6, 18, 24], the values of $a = 2$ and $\alpha = 10$ were really set arbitrarily. Indeed, these parameters could have been replaced by one of many other values. As a matter of fact, we have verified in a experimental way that the AdNN possesses *more powerful* AM properties when $a = 7$. As we can see from Figures 6 (a) and (b), when $a = 2$ and $\alpha = 10$, the largest LE is negative which indicates that there is no chaos. This also explains why the AdNN converges after a long transient phase — which is a phenomenon also reported in [2] — approximately $21,000$ iterations[8].

4.2 Lyapunov Analysis of the Ideal-M-AdNN

We shall now perform an analysis of the Ideal-M-AdNN (i.e., the entire network) from the viewpoint of its Lyapunov Exponents.

As we stated earlier in Section 3, the Ideal-M-AdNN originates from the AdNN and the M-AdNN. It differs from the M-AdNN in its parameter settings w_{ij}, ε and a_i, which is quite crucial, because the same network can exhibit

[8] Elsewhere, we have verified that the size of transient phase varies with the actual initial input[25].



(*a*) (*b*)

Fig. 6. The variation of the Largest Lyapunov Exponents of a single neuron with the parameters a and α. These two figures are plotted using Algorithm 1. In (a), a varies from 0 to 10 with a step size of 0.1. In (b), α varies from 0 to 100 with a step size of 1.

completely different phenomena dependent on the settings themselves. Consequently, the theoretical Lyapunov analysis of the Ideal-M-AdNN should be exactly the same as that of the M-AdNN, except that the LEs are evaluated at the *current* parameter settings.

As demonstrated in [6], the Jacobian matrix J of the Ideal-M-AdNN (and the M-AdNN) can be seen to have the form[9]:

$$J = \begin{pmatrix} J_{ij}^1 & J_{ij}^2 \\ J_{ij}^3 & J_{ij}^4 \end{pmatrix} = \begin{pmatrix} 0 \cdots 0 & k_f & 0 & \cdots 0 \\ \vdots \ddots \vdots & \vdots & \vdots & \ddots \vdots \\ 0 \cdots 0 & k_f & 0 & \cdots 0 \\ 0 \cdots 0 & 0 & \cdots 0 & k_r \\ \vdots \ddots \vdots & \vdots & \ddots \vdots & \vdots \\ 0 \cdots 0 & 0 & \cdots 0 & k_r \end{pmatrix} = \begin{cases} k_f, 0 < i \leq N, & j = N \\ k_r, \ N < i \leq 2N, j = 2N \\ 0, \quad otherwise \end{cases}$$

where J_{ij}^n is an N by N matrix, and $J_{ij}^1(t) = \frac{\partial \eta_i(t+1)}{\partial \eta_j(t)}$, $J_{ij}^2(t) = \frac{\partial \eta_i(t+1)}{\partial \xi_j(t)}$, $J_{ij}^3(t) = \frac{\partial \xi_i(t+1)}{\partial \eta_j(t)}$, $J_{ij}^4(t) = \frac{\partial \xi_i(t+1)}{\partial \xi_j(t)}$ respectively.

Generally speaking, it is not easy to compute the limit of Equation (28). However, due to the special form of J that we encounter here, we are able to calculate the value of Λ easily. As illustrated above,

[9] For the interest of brevity, we omit the detailed algebraic steps, but only present the final results. The details of these expressions can be found in [6]. Since we have chosen the N^{th} neuron as the global neuron, we see that only the N^{th} column vector of J_{ij}^1 and J_{ij}^4 is non-zero.

$$J_t = \begin{pmatrix} 0 \cdots 0 & k_f^t & 0 & \cdots & 0 \\ \vdots \ddots \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 \cdots 0 & k_f^t & 0 & \cdots & 0 \\ 0 \cdots 0 & 0 & \cdots & 0 & k_r^t \\ \vdots \ddots \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 \cdots 0 & 0 & \cdots & 0 & k_r^t \end{pmatrix}$$

whence we get:

$$\Lambda = \lim_{t\to\infty} [J_t \cdot (J_t)^T]^{\frac{1}{2t}} = \begin{pmatrix} k_f \cdots k_f & 0 & \cdots & 0 \\ \vdots \ddots \vdots & \vdots & \vdots & \ddots & \vdots \\ k_f \cdots k_f & 0 & \cdots & 0 \\ 0 \cdots 0 & k_r & \cdots & k_r \\ \vdots \ddots \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 \cdots 0 & k_r & \cdots & k_r \end{pmatrix}. \tag{34}$$

By a simple algebraic analysis we see that Λ has three different eigenvalues: Nk_f, Nk_r and 0. As a result, the Lyapunov Exponents are:

$$\lambda_1 = \cdots \lambda_{N-1} = -\infty,$$

$$\lambda_N = \log N + \log k_f > 0,$$

$$\lambda_{N+1} = \cdots \lambda_{2N-1} = -\infty,$$

$$\lambda_{2N} = \log N + \log k_r > 0.$$

In conclusion, the Ideal-M-AdNN has two positive LEs, which indicates that the network is truly a chaotic network!

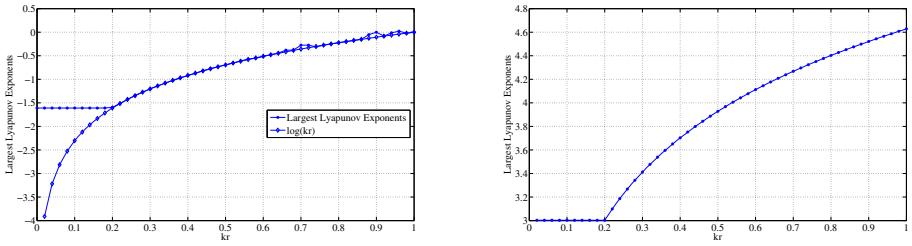
At this juncture, we would like to point out that in the work of [6], the authors took another form of Equation (28) as

$$\Lambda = \lim_{t\to\infty} [(J_t)^T \cdot J_t]^{\frac{1}{2t}}. \tag{35}$$

Consequently, the Largest LE is the maximum of the $\{\frac{1}{2}\log N + \log k_f, \frac{1}{2}\log N + \log k_r\}$, which is different from ours. However, from a realistic perspective, the choice of the definition does not matter. Indeed, independent of the definition we use, the result is consistent: The largest LE is positive which implies that the network is chaotic.

It's very interesting to compare this result with the one presented for the AdNN. Indeed, as we can see from [26], the AdNN has two different LEs: $\log k_f$ and $\log k_r$. Thus the largest LE is $\max\{\log k_f, \log k_r\}$. One can observe the following from Figure 7 (*a*): When $k_r < 0.2$, the largest LE is approximately -1.6, while $k_r > 0.2$, the largest LE is[10] $\log k_r$. As opposed to this, the largest LE of

[10] Please note that due to the fact that the computation is done numerically, there are some numerical errors, and thus the largest LE is not exactly, but very close to $\log k_r$ at some points.



(a) (b)

Fig. 7. The spectrum of the largest LEs of (a) the AdNN and of (b) the Ideal-M-AdNN, in which we vary k_r from 0 to 1 with a step size of 0.02, and where $k_f = 0.2$. Figure (a) is calculated using Algorithm 1 while (b) is calculated using Algorithm 2.

the Ideal-M-AdNN is always positive, as shown in (b). The difference is that by binding the states of all the neurons to a single "*global*" neuron, we force the Ideal-M-AdNN to have two positive LEs.

5 Chaotic and PR Properties of the Ideal-M-AdNN

We shall now report the properties of the Ideal-M-AdNN. The protocol of our experiments is quite simple: If the network output resonates a known input pattern with *any* periodicity, we deem this pattern to have been recognized from a chaotic PR perspective. Otherwise, we say that it is unrecognizable. These properties have been gleaned as a result of examining the Hamming distance between the input pattern and the patterns that appear at the output. As a comparison, we also list the simulation results of the M-AdNN. By comparing the Hamming distance of the Ideal-M-AdNN and the M-AdNN, we can then conclude which of the schemes performs better when it concerns PR properties. In this regard, we mention that the experiments were conducted using two data sets, namely the figures used by Adachi *et al* given in Figure 8 (a), and the numeral data sets used by Calitoiu *et al* [15, 6] given in Figure 8 (b). In both the cases, the patterns were described by 10×10 pixel images, and the networks thus had 100 neurons.

(a) (b)

Fig. 8. The 10×10 patterns used by Adachi *et al* (on the top) and Calitoiu *et al* (at the bottom). In both figures (a) and (b), the first four patterns are used to train the network. The fifth patterns are obtained from the corresponding fourth patterns by including 15% noise in (a) and (b) respectively. In each case, the sixth patterns are the untrained patterns.

Before we proceed, we remark that although the experiments were conducted for a variety of scenarios, in the interest of brevity, we present here only a few typical sets of results – essentially, to catalogue the overall conclusions of the investigation.

We discuss the properties of the Ideal-M-AdNN in two different settings: The NNs AM properties and its PR properties. In all cases, the parameters were set to be $k_f = 0.2$, $k_r = 0.9$, $\varepsilon = 0.015$, and all internal states $\eta_i(0)$ and $\xi_i(0)$ started from 0. Further, we catalogue our experimental protocols as follows:

1. AM properties: Although the AM properties are not the main issues of this paper, it is still interesting to examine whether the Ideal-M-AdNN possesses any AM-related properties. This is because the Ideal-M-AdNN is a modified version of the AdNN and the M-AdNN. By comparing the differences between the Ideal-M-AdNN and the AdNN or the M-AdNN, we can obtain a better understanding of its dynamical properties. In this case, the external stimulus a is a constant, i.e., $a = 2$.

2. PR properties: This investigation is really the primary intent of this paper. In this case, we investigate whether our new network is able to achieve pattern recognition. This is accomplished by checking whether the output can respond correctly to different inputs. In our experiments, we tested the network with known patterns, noisy patterns and with unknown patterns. In this case, the external stimulus was set to $a = 2 + 6P$, where P is the input pattern.

5.1 AM Properties

We now examine whether the Ideal-M-AdNN possesses any AM-related properties for certain scenarios, i.e., if we fix the external input $a_i = 2$ for all neurons. The observation that we report is that during the first 1,000 iterations (due to the limitations of the file size, we present here only the first 36 images), the network only repeats black and white image. This can be seen in Figure 9.

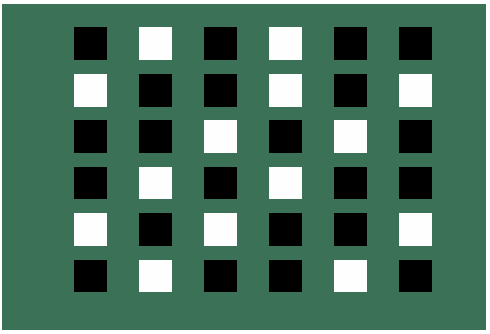


Fig. 9. The visualization of the output of the Ideal-M-AdNN under the external input $a_i = 2$. We see the output switches between images which are entirely only black or white.

It is very easy to understand this phenomenon: First of all, we see that all the neurons have the same output, 0, at time t (i.e, $t = 0$). This is true because we start the iteration with all internal states $\eta_i(0) = 0, \xi_i(0) = 0$ and $x_i(0) = 0$. As a result, the first image of Figure 9 is completely black[11]. At the next time step, $\eta_i(1) = 0$, $\xi_i(1) = a > 0$, which causes the output of all the neurons to be $x_i(1) = f(\eta+\xi) \approx 1$, which is why we see a completely white image. At the third time step, we first computed the summation $\Sigma_{j=1}^{j=N} w_{ij}x_j(1)$ where w_{ij} is defined by Equation (15). We can verify that the summation of each line of of the matrix w is either -0.5 or 0.5. Thus, $\eta_i(2)$ is either 0.5 or -0.5. Meanwhile, we must note that $\alpha = 10$, which results in a negative value for $\xi_i(2)$ and $\eta_i(2) + \xi_i(2)$, and consequently, $x_i(1) = f(\eta+\xi) \approx 0$. This is the reason why we see the third image of Figure 9 to be completely black. If we follow the same arguments, we see that at any time instant, the outputs of all the neurons only switch between 1 and 0 synchronically, which means the output image of Figure 9 switches between black and white, implying that the network possesses *no* AM properties at all.

As a comparison, we mention that the M-AdNN also does not possess any AM-related properties. In this regard, the M-AdNN and the Ideal-M-AdNN are similar.

5.2 PR Properties

The PR properties of the Ideal-M-AdNN are the main concern of this paper. As illustrated in Section 3, the goal of a chaotic PR system is the following: The system should respond periodically to trained input patterns, while it should respond chaotically (with chaotic outputs) to untrained input patterns. We now confirm that the Ideal-M-AdNN does, indeed, possess such phenomena.

We now present an in-depth report of the PR properties of the Ideal-M-AdNN's by using a Hamming distance-based analysis. The parameters that we used were: $k_f = 0.2$, $k_r = 0.9$, $\varepsilon = 0.015$ and $a_i = 2+6x_i$. The PR-related results of the Ideal-M-AdNN are reported for the three scenarios, i.e., for trained inputs, for noisy inputs and for untrained (unknown) inputs respectively. We report only the results for the setting when the original pattern and the noisy version are related to P4. The results obtained for the other patterns are identical, and omitted here for brevity.

1. The external input of the network corresponds to a known pattern, say P4. To report the results for this scenario, we request the reader to observe Figure 10, where we can find that P4 is retrieved periodically as a response to the input pattern. This occurs 391 times in the first 500 iterations. On the other hand, the other three patterns never appear in the output sequence. The phase diagrams of the internal states that correspond to Figure 10 are shown in Figure 11, where the x-axes are $\eta_{86}(t) + \xi_{86}(t)$ (the neuron with the index 86 has been set to be the global neuron), and where the y-axes are $\eta_i(t) + \xi_i(t)$ where the index $i = 80, \cdots, 88$ (i is chosen randomly) respectively.

[11] In our visualization, the value 0 means "black" while 1 means "white".

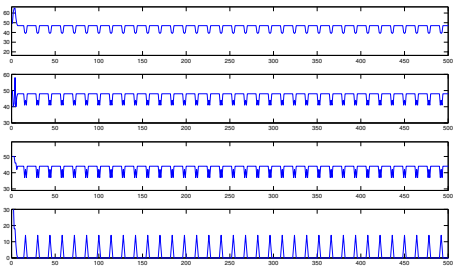


Fig. 10. PR properties: The Hamming distance between the output and the trained patterns. The input was the pattern P4. Note that P4 appears periodically, i.e., the Hamming distance is zero at periodic time instances.

From Figures 10 and 11, we can conclude that the input pattern can be recognized periodically if it is one of the known patterns. Furthermore, from Figure 11, we verify that the periodicity is 14, because all the phase plots have exactly 14 points.
As a comparison, we report that the M-AdNN is also able to recognize known patterns under *certain* conditions. However, this is not a phenomenon that we can "universally" ascribe to the M-AdNN because it fails to pass the necessary tests!

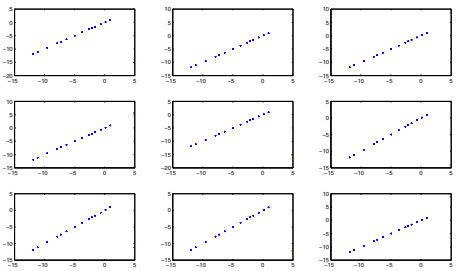


Fig. 11. PR properties: The phase diagrams of the internal states corresponding to Figure 10

2. The external input of the network corresponds to a noisy pattern, in this case P5, which is a noisy version of P4.
Even when the external stimulus is a garbled version of a known pattern (in this case P5 which contains 15% noise), it is interesting to see that *only* the original pattern P4 is recalled periodically. In contrast, the others three known patterns are *never* recalled. This phenomenon can be seen from the Figure 12. By comparing Figures 10 and 12, we can draw the conclusion that the Ideal-M-AdNN can achieve chaotic PR even in the presence of noise and distortion.

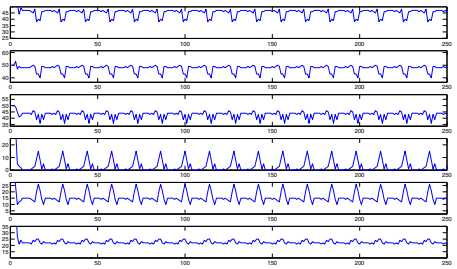


Fig. 12. PR properties: The Hamming distance between the output and the trained patterns. The input was the pattern P5. Note that P4 (not P5) appears periodically, i.e., the Hamming distance is zero at periodic time instances.

As in the previous case, the phase diagrams of the internal states that correspond to Figure 12 are shown in Figure 13, where the x-axes are $\eta_{86}(t)+\xi_{86}(t)$ (the neuron with the index 86 has been set to be the global neuron), and where the y-axes are $\eta_i(t) + \xi_i(t)$ where the index $i = 80, \cdots, 88$ (i is chosen randomly) respectively. As before, from Figure 13, we verify that the periodicity is 14, because all the phase plots have exactly 14 points.
Indeed, even if the external stimulus contains some noise, the Ideal-M-AdNN is still able to recognize it correctly, by resonating periodically! Furthermore, we have also tested the network using noisy patterns which contained up to 33% noise. We are pleased to report that it still resonates the input correctly, as shown in Figure 14. If the noise is even higher, i.e., more than 34%, the PR properties tend to gradually disappear. In this case, understandably, the simulation results are almost the same as when does the testing with *unknown* patterns.

3. The external input of the network corresponds to an unknown pattern, P6. In this case we investigate whether the Ideal-M-AdNN is capable of distinguishing between known and unknown patterns. Thus, we attempt to

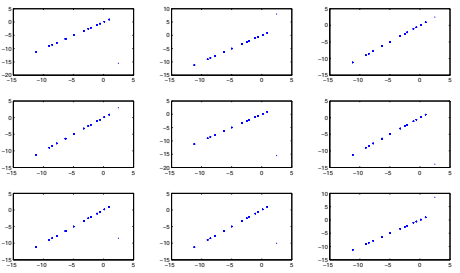


Fig. 13. PR properties: The phase diagrams of the internal states corresponding to Figure 12

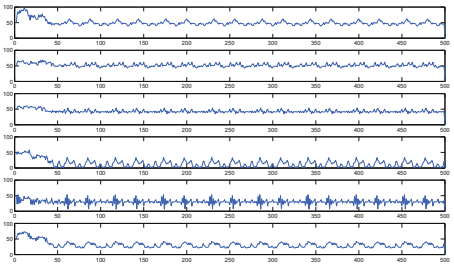


Fig. 14. The PR properties of the Ideal-M-AdNN when it encounters a noisy pattern which contain up to 33% noise. Observe that the system still resonates the corresponding known pattern correctly.

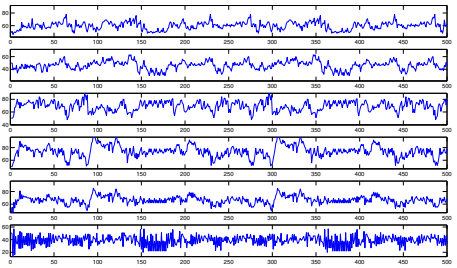


Fig. 15. PR properties: The Hamming distance between the output and the trained patterns. The input pattern was P6.

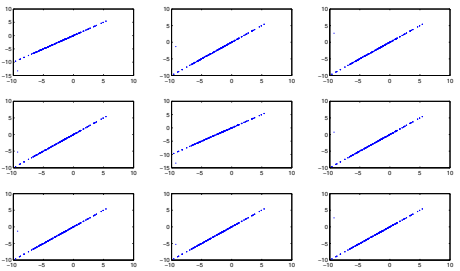


Fig. 16. PR properties: The phase diagrams of the internal states corresponding to Figure 15

stimulate the network with a completely unknown pattern. In our experiments, we used the pattern P6 of Figure 8 (a) initially used by Adachi *et al.* From Figure 15 we see that neither those known patterns nor unknown pattern appear.

As in the previous two cases, the phase diagrams of the internal states that correspond to Figure 15 are shown in Figure 16, where the x-axes are $\eta_{86}(t)+\xi_{86}(t)$ (where, as before, the neuron with the index 86 has been set to be the global neuron), and where the y-axes are $\eta_i(t)+\xi_i(t)$ where the index $i = 80, \cdots, 88$ (i is chosen randomly) respectively. The lack of periodicity can be observed from Figure 16, since the plots themselves are dense.

By way of comparison, we can see that the M-AdNN fails to pass this test, as can be seen from Figure 17. In this figure, the input is an unknown pattern, P6. We expect the network to respond chaotically to this input. Unfortunately, one sees that it still yields a periodic output. In short, no matter what the input is, the M-AdNN always yields the input pattern itself, and that, *periodically*. Obviously, this property cannot be deemed as PR. As opposed to this, the Ideal-M-AdNN responds intelligently to the various inputs with correspondingly different outputs, each resonating with the input that excites it – which is the crucial "golden" hallmark characteristic of a *Chaotic* PR system. Indeed, the switch between "order" (resonance) and "disorder" (chaos) seems to be consistent with Freeman's biological results – which, we believe, is quite fascinating!

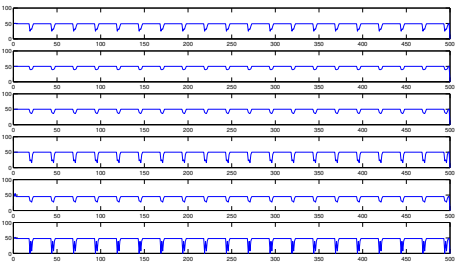


Fig. 17. The Hamming distance between the output and the trained patterns of the M-AdNN. The output repeats the input pattern, P6, periodically.

6 Conclusions

In this paper we have concentrated on the field of Chaotic Pattern Recognition (PR), which is a relatively new sub-field of PR. Such systems, which have only recently been investigated, demonstrate chaotic behavior under normal conditions, and "resonate" (i.e., by presenting at the output a specific pattern frequently) when it is presented with a pattern that it is trained with. This ambitious goal, with the requirement of switching from chaos to periodicity is, indeed, most demanding.

While the Adachi Neural Network (AdNN) [1–5] has properties which are pseudo-chaotic, it also possesses *limited* PR characteristics. As opposed to this, the Modified Adachi Neural Network (M-AdNN) proposed by Calitoiu *et al* [6], is a fascinating NN which has been shown to possess the required periodicity property desirable for PR. In this paper, we have explained why the PR properties claimed in [6] are not as powerful as originally claimed. Thereafter, we have argued for the basis for setting the parameters of the M-AdNN. By appropriately tuning the parameters of the M-AdNN for its weights, steepness and external inputs, we have obtained a new NN which we have called the Ideal-M-AdNN. Using a rigorous Lyapunov analysis, we have demonstrated the chaotic properties of the Ideal-M-AdNN. We have also verified that the system is truly chaotic for untrained patterns. But most importantly, we have shown that it is able to *switch to being periodic* whenever it encounters patterns with which it was trained (or noisy versions of the latter).

Apart from being quite fascinating, as far as we know, the theoretical and experimental results presented here are both unreported and novel. However, as the reader can observe, we were not so concerned about the application domain. Rather, our work was to be of an investigatory sort. Thus, although the data sets that we used were rather simplistic, they were used to submit a *prima facie* case. On one hand, if we could achieve our goal with these data sets, we believe that this research direction has the potential of initiating new research avenues in PR and AM. On the other hand, from a practical perspective, we believe that this is also beneficial for other computer science applications such as image searching, content addressed memory etc., and this is considered open at present. However, although we have not yet developed any immediate applications, we are willing to collaborate with others (and also provide them with our code) to do this.

References

1. Adachi, M., Aihara, K., Kotani, M.: An analysis of associative dynamics in a chaotic neural network with external stimulation. In: International Joint Conference on Neural Networks, Nagoya, Japan, vol. 1, pp. 409–412 (1993)
2. Adachi, M., Aihara, K.: Associative dynamics in a chaotic neural network. Neural Networks 10(1), 83–98 (1997)
3. Adachi, M., Aihara, K.: Characteristics of associative chaotic neural networks with weighted pattern storage-a pattern is stored stronger than others. In: The 6th International Conference On Neural Information, Perth Australia, vol. 3, pp. 1028–1032 (1999)
4. Adachi, M., Aihara, K., Kotani, M.: Pattern dynamics of chaotic neural networks with nearest-neighbor couplings. In: The IEEE International Sympoisum on Circuits and Systems, vol. 2, pp. 1180–1183. Westin Stanford and Westin Plaza, Singapore (1991)
5. Aihara, K., Takabe, T., Toyoda, M.: Chaotic neural networks. Physics Letters A 144(6-7), 333–340 (1990)
6. Calitoiu, D., Oommen, B.J., Nussbaum, D.: Periodicity and stability issues of a chaotic pattern recognition neural network. Pattern Analysis and Applications 10(3), 175–188 (2007)

7. Freeman, W.J.: Tutorial on neurobiology: from single neurons to brain chaos. International Journal of Bifurcation and Chaos in Applied Sciences and Engineering 2, 451–482 (1992)
8. Pyragas, K.: Continuous control of chaos by self-controlling feedback. Physics Letters A 170(6), 421–428 (1992)
9. Nozawa, H.: A Neural-Network Model as a Globally Coupled Map and Applications Based on Chaos. Chaos 2(3), 377–386 (1992)
10. Chen, L., Aihara, K.: Chaotic simulated annealing by a neural network model with transient chaos. Neural Networks 8(6), 915–930 (1995)
11. Tsui, A.P.M., Jones, A.J.: Periodic response to external stimulation of a chaotic neural network with delayed feedback. International Journal of Bifurcation and Chaos 9(4), 713–722 (1999)
12. Hiura, E., Tanaka, T.: A chaotic neural network with duffing's equation. In: Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, pp. 997–1001 (2007)
13. Tanaka, T., Hiura, E.: Computational abilities of a chaotic neural network. Physics Letters A 315(3-4), 225–230 (2003)
14. Tanaka, T., Hiura, E.: Dynamical behavior of a chaotic neural network and its applications to optimization problem. In: The International Joint Conference On Neural Network, Montreal, Canada, pp. 753–757 (2005)
15. Calitoiu, D., Oommen, B.J., Nussbaum, D.: Desynchronizing a chaotic pattern recognition neural network to model inaccurate perception. IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics 37(3), 692–704 (2007)
16. Qin, K., Oommen, B.J.: Chaotic pattern recognition: The spectrum of properties of the adachi neural network. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) S+SSPR 2008. LNCS, vol. 5342, pp. 540–550. Springer, Heidelberg (2008)
17. Qin, K., Oommen, B.J.: An enhanced tree-shaped adachi-like chaotic neural network requiring linear-time computations. In: The 2nd International Conference on Chaotic Modeling, Simulation and Applications, Chania, Greece, pp. 284–293 (2009)
18. Qin, K., Oommen, B.J.: Adachi-like chaotic neural networks requiring linear-time computations by enforcing a tree-shaped topology. IEEE Transactions on Neural Networks 20(11), 1797–1809 (2009)
19. Nagumo, J., Sato, S.: On a response characteristic of a mathematical neuron model. Biological Cybernetics 10(3), 155–164 (1971)
20. Eckmann, J.P., Rulle, D.: Ergodic theory of chaos and strange attractors. Reviews of Modern Physics 57(3), 617–656 (1985)
21. Sprott, J.C.: Numerical calculation of largest lyapunov exponent (1997)
22. Wolf, A., Swift, B.J., Swinney, L.H., Vastano, A.J.: Determining lyapunov exponent from a time series. Physica 16D, 285–317 (1985)
23. Sandri, M.: Numerical calculation of lyapunov exponents. The Mathematica Journal, 78–84 (1996)
24. Qin, K., Oommen, B.J.: Networking logistic neurons can yield chaotic and pattern recognition properties. In: IEEE International Conference on Computational Intelligence for Measure Systems and Applications, Ottawa, Ontario, Canada, pp. 134–139 (2011)
25. Luo, G.C., Ren, J.S., Qin, K.: Dynamical associative memory: The properties of the new weighted chaotic adachi neural network. IEICE Transactions on Information and Systems E95d(8), 2158–2162 (2012)
26. Adachi, M., Aihara, K.: An analysis on instantaneous stability of an associative chaotic neural network. International Journal of Bifurcation and Chaos 9(11), 2157–2163 (1999)