

Networking Logistic Neurons can Yield Chaotic and Pattern Recognition Properties

Ke Qin

School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, 611731, China
Email: thinker_qk@hotmail.com

B. John Oommen, *IEEE Fellow*

School of Computer Science
Carleton University
1125 Colonel By Dr., Ottawa, ON, K1S 5B6, Canada
Email: oommen@scs.carleton.ca

Abstract—Over the last few years, the field of Chaotic Neural Networks (CNNs) has been extensively studied because of their potential applications in the understanding/recognition of patterns and images, their associative memory properties, their relationship to complex dynamic system control, and their capabilities in the modeling and analysis of other measurement systems. However, the results concerning CNNs which can demonstrate chaos, quasi-chaos, Associative Memory (AM), and Pattern Recognition (PR) are scanty. In this paper, we consider the consequences of networking a set of Logistic Neurons (LNs). By appropriately defining the input/output characteristics of a fully connected network of LNs, and by defining their set of weights and output functions, we have succeeded in designing a Logistic Neural Network (LNN) possessing some of these properties. The chaotic properties of a single-neuron have been formally proven, and those of the entire network have also been alluded to. Indeed, by appropriately setting the parameters of the LNN, we show that the LNN can yield AM, chaotic and PR properties for different settings. As far as we know, the results presented here are novel, and the chaotic PR properties of such a network are unreported.

I. INTRODUCTION

As a soft computing methodology, Neural Networks (NNs) are one of the fundamental strategies to tackle adaptable and flexible solutions for control systems, and for modeling and measurement systems. Further, as a sub-branch of NNs, Chaotic Neural Networks (CNNs) possess more useful properties in the area of dynamic system control and measurement systems. Historically, the concept of CNNs originates from the so called phenomena of Physical-Biology. 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[1]. Thus, as applied scientists, if we are able to develop a system which mimics the brain to achieve chaos and PR, it could lead to a new model of NN, which is the primary goal of Chaotic Neural Networks (CNNs), and chaotic PR. In turn, the modeling and measurements achieved

using such a methodology can be further employed to control the underlying system by making inferences based on the classification and PR.

CNNs which also possessed PR were first proposed by Adachi and his co-authors[2], [3], [4], [5], [6]. They designed a new simple neuron 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 Associative Memory (AM) and PR properties. In the next year, the author of [7] proposed two methods of controlling chaos with 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 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, or to obtain Associative Memory (AM) and/or PR properties. An interesting step in this regard was the work reported in [8], where the authors utilized the delayed feedback and Ikeda map to design a CNN to mimic the biological phenomena observed by Freeman[1]. Thereafter, in [9], [10], [11] Hiura and Tanaka investigated several CNNs based on a piecewise sine map or the Duffing's equation to solve the TSP. Their simulations showed that the latter model 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 [12], 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.

As a tool in the area of intelligent measurement systems, CNNs have also been utilized for system optimization. Chen and Aihara [13] demonstrated that their CNN-based search algorithm had better search ability for solving combinatorial optimization problems than the Hopfield-Tank approach. In 2007, the authors of [14] proposed a CNN-based predictive

The work of the first author was supported by the Fundamental Research Funds for the Central Universities in China: ZYGX2009J064. The second author is a *Chancellor's Professor*, a *Fellow: IEEE* and a *Fellow: IAPR*. He is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway.

control strategy by using ‘‘Tent-map’’ Chaotic Particle Swarm Optimization (TCPSO). Indeed, results describing the utilization of the theory and applications of CNNs in measurement systems can be found in [15], [16], [17].

The primary aim of this paper is to use a completely different neuron as a primitive element in the network, and to see if we can obtain chaotic, PR and AM properties. Aiming to develop a completely new chaotic PR system, in this paper, we present a CNN which is founded on the *Logistic Map*. Based on numerous simulation results, we demonstrate that this new NN possesses the desirable properties of AM, *Chaos* and *PR* for different settings. As far as we know, this is novel to the field of CNNs. Hopefully, the proposal presents in this article can be applied in measurement systems, especially for the image understanding and recognition.

II. STATE OF THE ART

A. 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 $\xi_i(t)$ ($i = 1 \dots N$) at time t . The output of the i^{th} neuron, $x_i(t)$, is given by the so-called *Logistic Function* which will also be used in this paper.

Before we proceed, we emphasize that in this paper, we differentiate between the terms *Logistic Function* and *Logistic Map*. The former is defined by $y = \frac{1}{1+e^{-x}}$ (also referred to as the *Sigmoid Function*), and the latter is defined by equation (3), explained later.

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 behave like a PR system, although with some weaknesses, as illustrated in [18].

By invoking a Lyapunov Exponents analysis (LE), one can show that the AdNN has $2N$ negative LEs. Unlike the AdNN, which incorporates all the internal states to achieve the dynamical behavior, the M-AdNN uses two global internal states which are both associated with a *single* neuron, for example, the N^{th} neuron.

By resorting to this modification, the M-AdNN has the two positive LEs: $\lambda_N = \ln k_f + \frac{1}{2} \ln N$ and $\lambda_{2N} = \ln k_r + \frac{1}{2} \ln N$, which renders the M-AdNN to be truly chaotic.

Calitoui and his co-authors [19] also proposed a new approach for modeling the problem of blurring or inaccurate perception, and demonstrated that the quality of a system can be modified without manipulating the quality of the stimulus.

B. Our Previous Work

More recently, in our previous paper[20], we presented a collection of previously unreported properties of the AdNN.

We have shown that it goes through a spectrum of characteristics as one of its crucial parameters, α , changes. As α increases, it is first an AM, and it then becomes *quasi*-chaotic. The system is subsequently distinguished by two phases which really do not have clear boundaries of demarcation, where in the former it is *quasi*-chaotic for some patterns and periodic for others, and in the latter, it exhibits PR properties. It is fascinating that the AdNN also possesses the capability to recognize masked or occluded patterns, and even patterns which are completely inverted.

Later, we investigated problem of reducing the computational cost of the AdNN and its variants. Because their structures involve a completely connected graph, the computational complexity of the AdNN (and its variants) is quadratic in the number of neurons. In [18], we considered how the computations can be significantly reduced by merely using a linear number of inter-neuron connections. To achieve this, we extracted from the original completely connected graph, *one* of its spanning trees, and then computed the best weights for *this* spanning tree by using a gradient-based algorithm. By a detailed experimental analysis, we showed that the new linear-time AdNN-like network possesses chaotic and PR properties for different settings.

III. THE LOGISTIC NN: A NEW MODEL OF CNNs

Consider the discrete Hopfield NN model, characterized by the following equations:

$$y_i(t+1) = ky_i(t) + \alpha \left(\sum_j w_{ij} x_j(t) + a_i \right). \quad (1)$$

In an attempt to obtain AM and PR properties, we shall modify the structure by introducing a logistic *feedback* component. Therefore, our new network possesses a Present-State/Next-State function, and a State/Output function, which are described by means of the following equations relating the only internal state $\eta_i(t)$ and the output $x_i(t)$ as follows:

$$\begin{aligned} \eta_i(t+1) &= k\eta_i(t) + \alpha \left(\sum_j w_{ij} x_j(t) + a_i \right) \\ &\quad - \beta z_i(t) x_i(t), \end{aligned} \quad (2)$$

$$z_i(t+1) = 4z_i(t)(1 - z_i(t)), \quad (3)$$

$$x_i(t+1) = \frac{1}{1 + e^{-\eta_i(t+1)/\varepsilon}}. \quad (4)$$

Observe that the new model is composed of N neurons, topologically arranged as a completely connected graph. Each neuron i , $i = 1, 2, \dots, N$, has an internal state $\eta_i(t)$ and an output $x_i(t)$. With regard to the Present-State/Next-State function, at time instant $t+1$, the internal state $\eta_i(t+1)$ is determined by previous internal state $\eta_i(t)$, the external stimulus and the net input which is obtained via the feedback $x_i(t)$. In the equations, k is the damping factor (of the ‘‘nerve’’ membrane), where $0 < k \leq 1$. Also, since α and β are constants, we have set α to be unity, and used β to be the parameter which is tuned to get the desired performance. Also,

while $z_i(t)$ is a chaotic feedback factor characterizing the Logistic map given by equation (3), the weights $\{w_{ij}\}$ are the edge weights obtained by the classic definition as in [3]. Observe that this is a one-shot assignment and that it, in and of itself, does not include an additional training phase.

IV. LYAPUNOV ANALYSIS OF A SINGLE LOGISTIC NEURON

We first undertake the Lyapunov analysis of a single neuron. Indeed, it can be easily proven that a single neuron is chaotic by considering its Jacobian matrix and its QR decomposition[21].

Consider the primitive component of the LNN, where the model of a *single* neuron can be described as¹:

$$\eta(t+1) = k\eta(t) + \alpha(x(t) + a) - \beta z(t)x(t), \quad (5)$$

$$z(t+1) = 4z(t)(1 - z(t)), \quad (6)$$

$$x(t+1) = \frac{1}{1 + e^{-\eta(t+1)/\varepsilon}}. \quad (7)$$

As is well known, the LEs are defined from the Jacobian matrix as:

$$J^t(\xi_0) = \frac{df^t(\xi)}{dx} \Big|_{\xi_0} \quad (8)$$

In this case, the LEs are defined by

$$\lambda_i(\xi_0) = \log \Lambda_i(\xi_0), \quad (9)$$

where $\Lambda_i(\xi_0)$ are the eigenvalues of the limit defined by:

$$L(\xi_0) = \lim_{t \rightarrow \infty} (J^t \cdot (J^t)^T)^{\frac{1}{2t}}, \quad (10)$$

where $(J^t)^T$ denotes the transpose of J^t .

In order to avoiding the complicated task of computing the limit of equation (10), we resort to the strategy proposed by Eckmann and Sandri in [21], [22]. To achieve this, we calculate the LEs of this discrete dynamical system in another way, that is, by resorting to the QR decomposition.

We rewrite $J(\xi_0)$ using its QR decomposition as $J(\xi_0) = Q_1 R_1$. We now define $J_k^* = J(f^{k-1}(\xi_0))Q_{k-1}$, and decompose $J_k^* = Q_k R_k$. Consequently, we obtain $J(f^{k-1}(\xi_0)) = Q_k R_k Q_{k-1}^{-1}$. By applying this equation to the chain rule, the differential $Df^t(\xi_0)$ can be transformed to be:

$$\begin{aligned} Df^t(\xi_0) &= Q_t R_t Q_{t-1}^{-1} \cdot Q_{t-1} R_{t-1} Q_{t-2}^{-1} \cdots R_1 \\ &= Q_t R_t \cdots R_1 \end{aligned} \quad (11)$$

where Q_i is an orthogonal matrix, and R_i is upper triangular matrix. The LEs, $\{\lambda_i\}$ can then be obtained as:

$$\lim_{t \rightarrow \infty} \frac{1}{t} \ln |v_{ii}^{(t)}| = \lambda_i, \quad (12)$$

where $\{v_{ii}\}$ are the diagonal elements of the upper-triangular product $\Upsilon^{(t)} = R_t \cdots R_1$.

We are now in a position to compute the LEs of the system described by equations (5) and (6). By a straightforward computation we see that the Jacobian matrix of the neuron is:

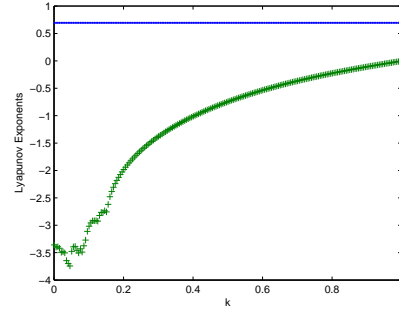


Fig. 1. The variation of the LEs of a single neuron with the parameter k .

$$J = \begin{pmatrix} \frac{\partial \eta(t+1)}{\partial \eta(t)} & \frac{\partial \eta(t+1)}{\partial z(t)} \\ \frac{\partial z(t+1)}{\partial \eta(t)} & \frac{\partial z(t+1)}{\partial z(t)} \end{pmatrix} = \begin{pmatrix} k + \frac{1}{\varepsilon}(\alpha - z(t)) \cdot x(t) \cdot (1 - x(t)) & -\beta x(t) \\ 0 & 4 - 8z(t) \end{pmatrix}.$$

One observes that J is an upper triangular. Thus, quite simply, we have $J_0 = Q_1 R_1$ where Q_1 is a unit matrix and R_1 is J_0 itself. Similarly, for all $i = 1, 2, \dots, t$, we have $R_i = J_{i-1}$. As a result, Υ can be rewritten as $\Upsilon = J_{t-1} \cdot J_{t-2} \cdots J_0$, which is also an upper triangular matrix, implying that the eigenvalues of Υ are the diagonal elements of Υ themselves.

Figure 1 shows the variation of the LEs of a single neuron with the parameter k . A single neuron has two LEs since it is defined by a two-dimension discrete system (equations (5) and (6)). From this figure we can see that one of the LEs is positive (approximately 0.6918), which implies that the behavior of every single neuron is chaotic. We also see from the figure that the positive eigenvalue approaches a constant value because it is only determined by $z(t)$. As we know, $z(t)$ is a chaotic map possessing the positive LE $\log 2 \approx 0.6932$! Consequently, we claim that a single LN is truly chaotic² because of the Logistic feedback factor $z(t)$.

In the same way as we used in our paper [18], the LEs of the entire network can be obtained easily. The details of Lyapunov analysis of the LNN has been eliminated for the interest of brevity.

V. CHAOTIC AND PR PROPERTIES OF THE LNN

We shall now report the AM and PR properties of the LNN. 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. In this regard, we mention that the experiments were conducted using the data set used by Adachi *et al* given in Figure 2.

A. AM Properties

We discuss the properties of the LNN in three different settings. In all of the three cases, the parameters were set to be $k = 0.55$, $\beta = 30$, $a_i = 2$ and $z_0 = 0.3$. We should also point out that we assumed that $z_1(0) = z_2(0) = \dots =$

²One should also observe that the other LE is negative. This LE corresponds to the first eigenvalue of the Jacobian matrix.

¹It should be observed that $w_{ii} = 1, i = 1, 2, \dots, N$.



Fig. 2. The 10×10 patterns used by Adachi *et al.*. The first four patterns are used to train the network. The fifth pattern is obtained from the fourth pattern by including 15% noise. The sixth pattern is the untrained pattern.

$z_{100}(0) = 0.3$. In other words, we assumed that the initial values of all the neurons were forced to be the same. Also, to emphasize, these AM properties were demonstrated by setting each “external stimulus” to be $a_i = 2$.

We now discuss the AM-related results of the LNN for the three scenarios, i.e., for trained inputs, for noisy inputs and for untrained (unknown) inputs respectively. In each case, as mentioned above, we report only a few results, i.e., for the case when the original pattern and the noisy version are related to P4. The results obtained for the other patterns are identical, and omitted here in the interest of brevity.

- 1) The input of the network is a known pattern, say P4. The observation that we report is that during the first 1,000 iterations, the network can dynamically retrieve all the known patterns. This can be seen in Figure 3 in which we plot the Hamming distance of the output pattern with P4. Observe if the identical pattern is observed the Hamming distance is 0, and if the inverse version is observed, the Hamming distance is 100.
- 2) The input of the network is a noisy pattern, in this case P5, which is a noisy version of P4. In this case, the network can still dynamically retrieve all the known patterns and their inverse versions during the first 1,000 iterations³. This can be seen in Figure 3 (b). The reader should observe that although the initial input is a *noisy* pattern, the four memorized patterns (and their inverses) appear in the output frequently. Based on the above observation and this present one, we can easily conclude that the four patterns and their inverse versions are the attractors of the network. Indeed, in both these cases we see that independent of the initial input, the network will be attracted to these attractors, even though the network will not stay at a fixed or periodic state – which is a typical phenomenon of chaotic systems.
- 3) The input of the network is an unknown pattern, P6. If the initial input is a completely unknown pattern, say, P6, a system possessing AM should still be able to reproduce/retrieve all the memorized patterns. On the other hand, the system should never yield an unknown pattern. This phenomena can be seen from Figure 4. Here the untrained pattern never appears in the output.

³From Figure 3 we see that the known pattern P2 is never recalled because the time frame is too short. Indeed, if the time-frame is expanded for 10,000 iterations, we see that the pattern P2 appears 5 times. But even in this expanded time-frame, P5 and P6 *never* appear

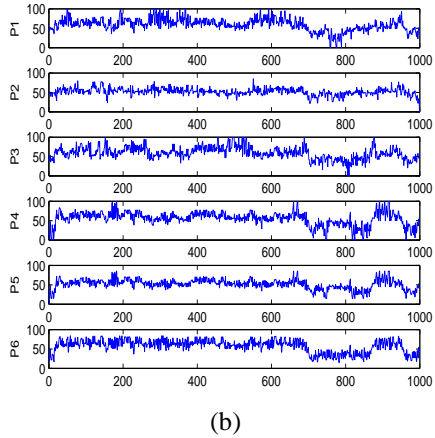
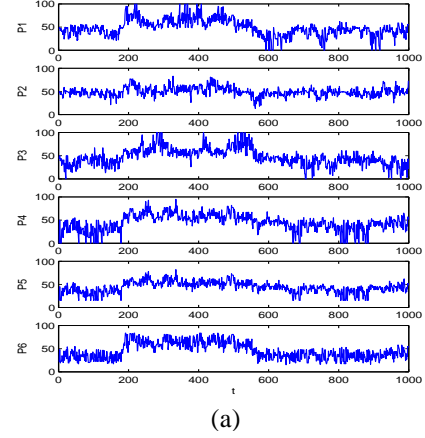


Fig. 3. AM properties: The Hamming distance between the output and the trained patterns. In (a) and (b), the input patterns are the fourth and fifth patterns of Figure 2 (a) respectively.

A record of the statistics (frequencies) of all the above three cases is catalogues in Table I.

B. PR Properties

Similar to the results reported in Section V-A, we now present an in-depth report of the LNN’s PR properties by using Hamming distance-based analyses. The parameters that we used were: $k = 0.55$, $\beta = 30$, $a_i = 2x_i$, and $z_0 = 0.3$. The difference between these experiments and the AM-related ones involve the “external stimulus” $\{a_i\}$, which is not a constant, but a scaled version of the $\{x_i\}$. The PR-related results of the LNN are reported for the three scenarios, i.e., for trained

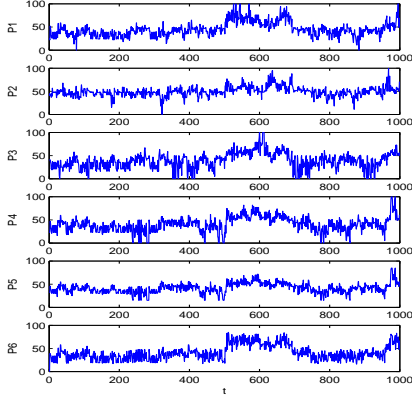


Fig. 4. AM properties: The Hamming distance between the output and the trained patterns. In this figure, the input pattern is the sixth pattern of Fig. 2

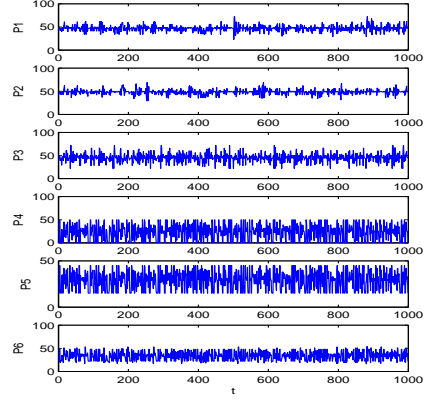
TABLE I

THE STATISTICS (FREQUENCIES) OF THE OCCURRENCES OF THE VARIOUS PATTERNS DURING THE FIRST 10,000 ITERATIONS. IN THE TABLE, $*P_i$ SIGNIFIES THE INVERSE VERSION OF THE PATTERN P_i .

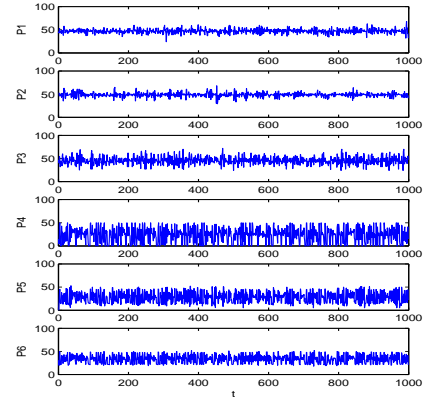
		Input Patterns					
		P_1	P_2	P_3	P_4	P_5	P_6
Frequency	P_1	99	96	108	102	92	89
	$*P_1$	187	85	201	197	191	186
	P_2	38	45	32	34	38	31
	$*P_2$	31	19	41	41	31	32
	P_3	190	190	185	174	189	222
	$*P_3$	235	90	219	219	236	218
Statistic	P_4	98	98	127	135	84	95
	$*P_4$	40	18	30	30	43	33
	P_5	0	0	0	0	1	0
	$*P_5$	0	0	0	0	0	0
	P_6	0	0	0	0	0	1
	$*P_6$	0	0	0	0	0	0

inputs, for noisy inputs and for untrained (unknown) inputs respectively. As in the AM case, we report only the results for the setting when the original pattern and the noisy version are related to P_4 . The results obtained for the other patterns are identical, and omitted here for brevity.

- 1) The input of the network is a known pattern, say P_4 .
To report the results for this scenario, we request the reader to observe Figure 5, where in (a) we can find that P_4 is retrieved very frequently as a response of the input pattern. This occurs 197 times (almost 1/5) in the first 1,000 iterations. On the other hand, the other three patterns never appear in the output sequence. From Figures 5 (a), we can conclude that the input can be recognized successfully if it is one of the known patterns.
- 2) The input of the network is a noisy pattern, in this case P_5 , which is a noisy version of P_4 .
Even when the external stimulus is a garbled version of a known pattern (in this case P_5 which contains 15% noise), it is interesting to see that *only* the original pattern P_4 is recalled frequently (as high as 187 times in the first 1,000 iterations). In contrast, the others three known patterns are never recalled. This phenomena can



(a)



(b)

Fig. 5. PR properties: The Hamming distance between the output and the trained patterns. The input patterns are the P_4 and P_5 patterns of Figure 2 (a) respectively. Observe that P_5 is a 15%-noisy version of P_4 .

be seen from the Figure 5 (b). By comparing (a) and (b) of Figure 5, we can conclude that the LNN can achieve chaotic PR even in the presence of noise and distortion. Even if the external stimulus contains some noise, the LNN is still able to recognize it correctly.

- 3) The input of the network is an unknown pattern, P_6 .
In this case we investigate whether the LNN is capable of distinguishing between known and unknown patterns. Thus, we attempt to stimulate the network with a completely unknown pattern. We used the pattern P_6 of Figure 2 initially used by Adachi *et al.* From Figure 6 we see that some of the known patterns (P_1 , P_3 and P_4) are retrieved several times. As opposed to this, the noisy pattern P_5 and the unknown pattern P_6 never appear.

The statistical frequencies of the Hamming distances for the Adachi dataset for all the three cases are listed in Table II. We submit that is also a convincing proof of the fact that the LNN can achieve chaotic PR.

VI. CONCLUSIONS

In this paper we have concentrated on the field of Chaotic Pattern Recognition (PR), which is a relatively new sub-field of

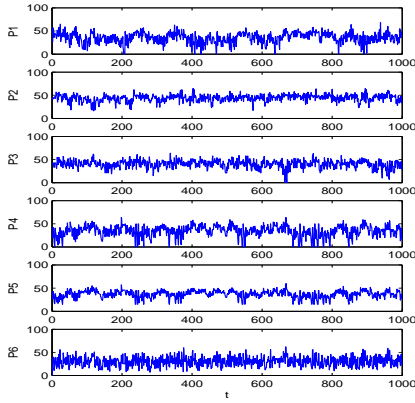


Fig. 6. PR properties: The Hamming distance between the output and the trained patterns. The input pattern is the sixth pattern of Figure 2. P_6 is an unknown pattern.

TABLE II
FREQUENCY-BASED STATISTICS FOR THE ADACHI DATA SET, WHICH DEMONSTRATES THE LNN'S PR CAPABILITIES FOR FIGURES 5 AND 6.

		Input Patterns					
		P_1	P_2	P_3	P_4	P_5	P_6
Frequency Statistic	P_1	188	0	0	0	0	16
	P_2	0	232	0	0	0	0
	P_3	0	0	204	0	0	3
	P_4	0	0	0	197	187	36
	P_5	0	0	0	0	1	0
	P_6	0	0	0	0	0	1

PR. The network which we have designed and investigated is one that has not been investigated before, namely the Logistic NN based on the chaotic Logistic Map. By appropriately defining the input/output characteristics of a fully connected LNN, and by defining their set of weights and output functions, we have succeeded in demonstrating that it possesses AM and PR properties for different inputs.

We summarize the LNN's AM and PR properties as follows:

- Case I — AM: When exposed to an external stimulus with $a_i = 2$, it performs as an AM. In other words, it is able to retrieve all the known patterns independent of what the initial input pattern is.
- Case II — PR: When exposed to an external stimulus $a_i = 2x_i$, it achieves PR. Further, in this case:
 - If x_i corresponds to a known pattern or a noisy pattern, the LNN can recognize it correctly.
 - If the x_i corresponds to an unknown pattern, the LNN can retrieve all the known patterns (akin, to some degree, to an AM phenomenon), but the frequency with which the patterns occur is much lower than in Case I.

Finally, we observe that the AM and PR characteristics of the LNN depending on the settings and the inputs. By the well-defined and well-trained parameters, the LNN can be applied for image understanding and recognition, which is crucial for measurement systems.

REFERENCES

- [1] W. J. Freeman, "Tutorial on neurobiology: from single neurons to brain chaos," *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering*, vol. 2, pp. 451–482, 1992.
- [2] K. Adachi, M. Aihara and M. Kotani, "An analysis of associative dynamics in a chaotic neural network with external stimulation," in *Proceedings of the 1993 International Joint Conference on Neural Networks, Nagoya, Japan*, vol. 1, 1993, pp. 409–412.
- [3] M. Adachi and K. Aihara, "Associative dynamics in a chaotic neural network," *Neural Networks*, vol. 10, no. 1, pp. 83–98, 1997.
- [4] —, "Characteristics of associative chaotic neural networks with weighted pattern storage—a pattern is stored stronger than others," in *Proceedings Of The 6th International Conference On Neural Information, Perth Australia*, vol. 3, 1999, pp. 1028–1032.
- [5] M. Adachi, K. Aihara, and M. Kotani, "Pattern dynamics of chaotic neural networks with nearest-neighbor couplings," in *Proceedings of the 1991 IEEE International Symposium on Circuits and Systems, Westin Stanford and Westin Plaza, Singapore*, vol. 2, 1991, pp. 1180–1183.
- [6] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," *Physics Letters A*, vol. 144, no. 6-7, pp. 333–340, 1990.
- [7] K. Pyragas, "Continuous control of chaos by self-controlling feedback," *Physics Letters A*, vol. 170, no. 6, pp. 421–428, 1992.
- [8] A. P. M. Tsui and A. J. Jones, "Periodic response to external stimulation of a chaotic neural network with delayed feedback," *International Journal of Bifurcation and Chaos*, vol. 9, no. 4, pp. 713–722, 1999.
- [9] E. Hiura and T. Tanaka, "A chaotic neural network with duffing's equation," in *Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, 2007*, pp. 997–1001.
- [10] T. Tanaka and E. Hiura, "Computational abilities of a chaotic neural network," *Physics Letters A*, vol. 315, no. 3-4, pp. 225–230, 2003.
- [11] —, "Dynamical behavior of a chaotic neural network and its applications to optimization problem," in *Proceedings Of The International Joint Conference On Neural Network, Montreal Canada, 2005*, pp. 753–757.
- [12] D. Calitoiu, B. J. Oommen, and D. Nussbaum, "Desynchronizing of chaotic pattern recognition neural network to model inaccurate perception," *IEEE Transaction on Systems, Man, and Cybernetics-part B: Cybernetics*, vol. 37, no. 3, pp. 692–704, 2007.
- [13] L. Chen and K. Aihara, "Global searching ability of chaotic neural networks," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no. 8, pp. 974–993, 1999.
- [14] Y. Song, Z. Chen, and Z. Yuan, "New chaotic pso-based neural network predictive control for nonlinear process," *IEEE Transactions on Neural Networks*, vol. 18, no. 2, pp. 595–601, 2007.
- [15] Y. Hayakawa, A. Marumoto, and Y. Sawada, "Effects of the chaotic noise on the performance of a neural network model for optimization problems," *Phys. Rev. E*, vol. 51, no. 4, pp. 2693–2696, 1995.
- [16] T. Kwok and K. A. Smith, "Experimental analysis of chaotic neural network models for combinatorial optimization under a unifying framework," *Neural Networks*, vol. 13, no. 7, pp. 731–744, 2000.
- [17] L. Wang, S. Li, F. Tian, and X. Fu, "A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing," *IEEE Transactions on Systems, Man and Cybernetics, Part B: Cybernetics*, vol. 34, no. 5, pp. 2119–2125, 2004.
- [18] K. Qin and B. J. Oommen, "Adachi-like chaotic neural networks requiring Linear-time computations by enforcing a tree-shaped topology," *IEEE Transaction on Neural Networks*, vol. 20, no. 11, pp. 1797–1809, 2009.
- [19] D. Calitoiu, B. J. Oommen, and D. Nussbaum, "Periodicity and stability issues of a chaotic pattern recognition Neural Network," *Pattern Anal. Applic.*, vol. 10, no. 3, pp. 175–188, 2007.
- [20] Q. Ke and B. J. Oommen, "Chaotic pattern recognition: The spectrum of properties of the Adachi neural network," *Lecture Notes in Computer Science*, vol. 5342, pp. 540–550, 2008.
- [21] M. Sandri, "Numerical calculation of lyapunov exponents," *The Mathematica Journal*, pp. 78–84, 1996.
- [22] J. P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," *Reviews of Modern Physics*, vol. 57, no. 3, pp. 617–656, 1985.