



INFORMATION PROCESSING IN 1-D AND 2-D MAP: RECURRENT AND CELLULAR NEURAL NETWORKS IMPLEMENTATION.

Yu. V. Andreyev, A. S. Dmitriev, D. A. Kuminov, V. V. Pavlov.

Institute of RadioEngineering and Electronics of the Russian Academy of Sciences,
Mokhovaya St., 11, 103907 Moscow, Russia.
Phone: (095)203-4693, FAX: (095) 203-84-14, E-mail: dmitr@ire.rc.ac.ru

ABSTRACT: *Mathematical models of neural networks, based on one-dimensional and two-dimensional maps are proposed, in which complex dynamics and chaos are used for storing and processing information.*

1. Introduction.

A possibility of a use of chaos and complicated dynamics in neural nets for a solution of problems coupled with information processing, was discussed already in the literature [1-3].

It should be noted however, that in most papers connected with a use of chaos and complex dynamics in neural networks, the neural network itself plays a role of a "black box" which is trained according to certain rules in order to obtain a desired response to a certain signal at its input.

The aim of this report is to demonstrate an application of complex dynamics and chaos to storing and processing information in recurrent neural networks, used as a hardware realization of simple mathematical models of a kind of 1-D and 2-D maps, which were used for storing and retrieving information [4,5]. Evidently this concept substantially differs from the presentation of a neural network as a "black box". We shall show below that an existence of a mathematical model realized in neural network of this kind leads to striking computational efficiency. Another important property of such networks is the limited number of interconnections. Therefore they can be considered as a special kind of CNN.

2. Storing and Retrieving Information in 1-D And 2-D Maps.

The procedure of storing and retrieving information on the basis of limit cycles of 1-D dynamical systems was introduced in [4].

Let there be given a sequence of symbols (information block)

$$a_1 a_2 \dots a_n, \quad (1)$$

each element a_j of which belongs to an alphabet composed of N symbols.

1-D map of a segment into itself is designed for this sequence, possessing a stable limit cycle of the period n , the elements of which are in mutual unambiguous correspondence with the elements of the sequence (1). In the simplest case each element of the alphabet is related to its own value of the mapping variable and to an interval with the length $1/N$ of the mapping variable. The example of the map with storing information block **174** is shown in Fig.1.

The retrieval of an information block is performed by setting initial conditions within one of the segments on the unit interval $[0,1]$ corresponding to the symbols of the information block, and by further transformation of the sequence of numbers occurring by iteration of the map into the sequence of symbols. If the initial conditions are arbitrary, the convergence of the system trajectory to the limit cycle is preceded by a chaotic transient process.

The method of storing and retrieving was generalized to storing information in 2-D maps [5]. It is shown, that use of two dimension systems for information processing led to high information capacity.

For store of the information block (1) on 2-D map of an individual square $[0,1] \times [0,1]$ into self, it is necessary to transform (1) as pairs of elements

$$(a_1, a_2), (a_2, a_3), \dots, (a_n, a_1). \quad (2)$$

To each element j from the given alphabet of length N is set in mutual correspondence intervals on axes X and Y

$$I_j^x = \left[\frac{j}{N}, \frac{j+1}{N} \right), \quad I_j^y = \left[\frac{j}{N}, \frac{j+1}{N} \right), \quad j = 1, \dots, N, \quad (3)$$

and points in middle of these intervals $\frac{j+0.5}{N}$. In this case to a pair of elements (a_m, a_{m+1}) from a sequence (2) there will correspond a square

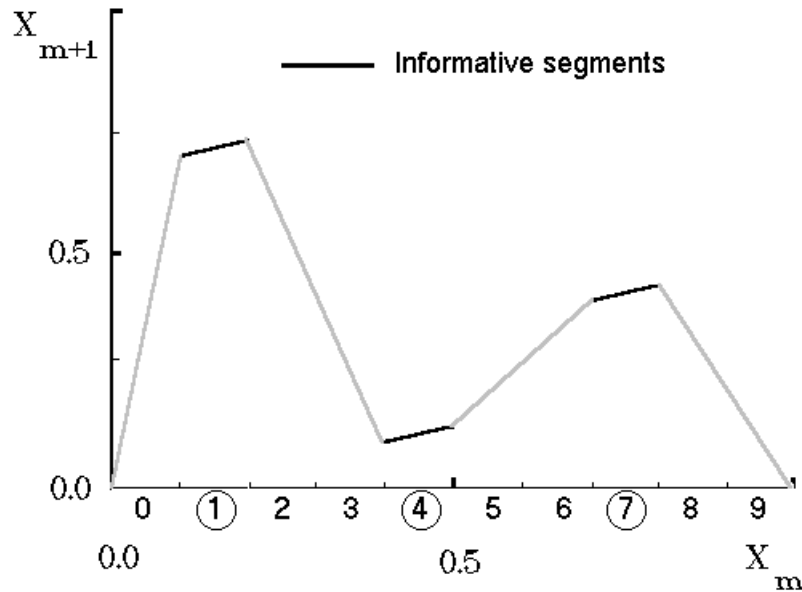


Fig.1. 1-D map with storing information block 174.

$$I_{a_m} \times I_{a_{m+1}} = \left[\frac{a_m}{N}, \frac{a_m + 1}{N} \right) \times \left[\frac{a_{m+1}}{N}, \frac{a_{m+1} + 1}{N} \right). \quad (4)$$

Is shown, that required 2-D map should look like

$$\begin{aligned} x_{m+1} &= y_m \\ y_{m+1} &= f(x_m, y_m) \end{aligned} \quad (5)$$

The function $f(x, y)$ is built so that the map (5) had a unique limiting cycle (2).

In the squares through which passes a cycle (2) (informative squares) function $f(x, y)$ looks like

$$f(x, y) = 2 \cdot D \cdot \left(x - \frac{i+0.5}{N} \right) + \frac{k+0.5}{N}, \quad 2D < 1. \quad (6)$$

In the other squares

$$f(x, y) = \frac{x - \frac{i}{N}}{\frac{1}{N}} \left(1 - \frac{1}{N} \right) + \frac{e}{N}, \quad e \sim 10^{-2}. \quad (7)$$

An example of a 2-D map, in which string 174 is stored, is shown in Fig.2.

3. Realization of the Dynamics of 1-D and 2-D Maps with Stored Information Using Neural Networks.

Let us consider a network composed of three layers of neurons (an input layer, a hidden layer, an output layer) and a feedback loop between the output layer and the input layer. The three layers of the neural network are used to emulate a 1-D map function, and the feedback loop with a unit delay is to organize iteration process [6-8].

The input and output layers contain one element each. There are no couplings between the elements of the hidden layer.

A signal corresponding to the initial conditions is applied to the input element. The signal from the output of this element is sent to the hidden layer neurons. The sum of the signals from the hidden layer neurons is applied to the input of the output layer neuron. The coefficients of the couplings between the layers of the neural network are determined by the function of a map with stored information. The signal at the network output represents the value of the map next iteration.

All the neural-like elements have the same structure and the same piece-wise linear characteristics. The threshold values for the elements of the input and output layers are zero, the threshold values for the elements of the hidden layer are determined by the form of the function which is realized in the network.

An arbitrary 1-D piece-wise linear map $M(x)$ defined at the unit interval $I = [0, 1]$ by the set of its points

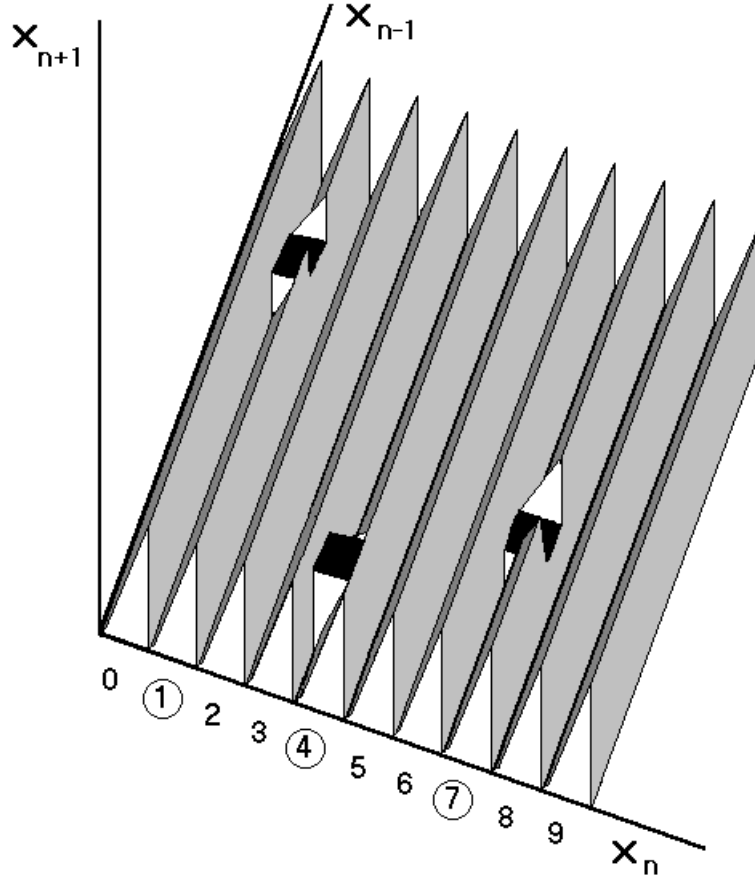


Fig.2. 2-D map with storing information block 174.

$$(x_1, y_1), (x_2, y_2), \dots, (x_{p+1}, y_{p+1}), \quad (8)$$

where p is the number of linear segments in the 1-D map; (x_i, y_i) are the coordinates of the left and right endpoints of the linear segments, can be precisely realized as a neural network, composed of the elements with identical piece-wise linear characteristics:

$$f(x) = \begin{cases} 0, & x \leq 0, \\ x, & 0 < x \leq 1, \\ 1, & x > 1. \end{cases} \quad (9)$$

The map function $M(x)$ may be represented as

$$M(x) = \sum_{i=1}^N \mathbf{b}_i \cdot f((x - x_i) / \mathbf{a}_i) = \sum_{i=1}^N T_i^2 f(T_i^1 x - \mathbf{q}_i), \quad (10)$$

where $\mathbf{a}_i = x_{i+1} - x_i$, $\mathbf{b}_i = y_{i+1} - y_i$, $T_i^1 = 1 / \mathbf{a}_i$ are the couplings from the input layer to the hidden layer, $T_i^2 = \mathbf{b}_i$ are the couplings from the hidden layer to the output layer, $\mathbf{q}_i = x_i / \mathbf{a}_i$ are the thresholds of the elements of the hidden layer. In Fig.3 an example of a neural net which realizes the dynamics of 1-D map storing the string 174 is presented.

In the case of piece-wise linear map realization as a neural net, some of the neural elements are used to realize nonlocal coupling between input and output elements, and nonlocal connections between these elements and the elements of the hidden layer.

To emulate a 2-D map function neural network more complex than for 1-D map structure is used.

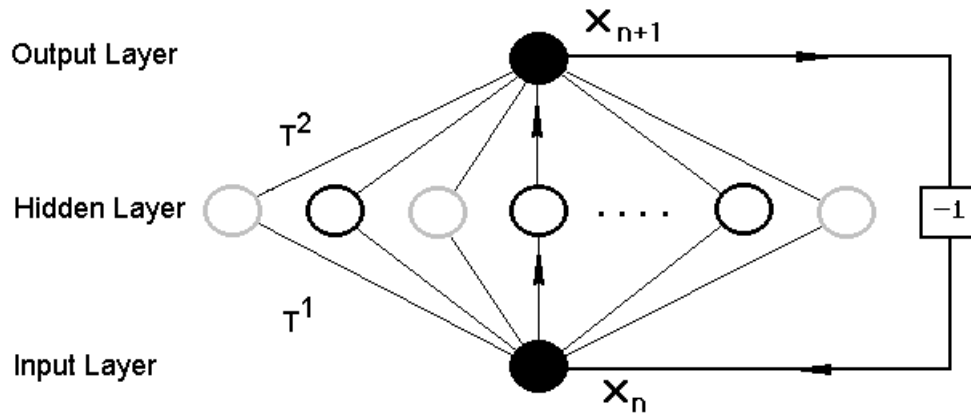


Fig.3. Neural net which realizes the dynamics of 1-D map storing the string 174.

The neural network is built as follows. From Fig.2 it is visible, that 2-D map can be constructed from segments of three types (one for informative squares and two for noninformative). Each of these segments are realized with the help of neural units of simple structure. An example of the net which is approximate the 2-D map is shown in Fig.4. Three modules in the right part of the network on Fig.4 realize informative squares, other realize noninformative squares. The training of the network is reduced to analytical formulas.

An important property of such a neural networks is a limited number of interconnections. This is the same property which we have in cellular neural networks.

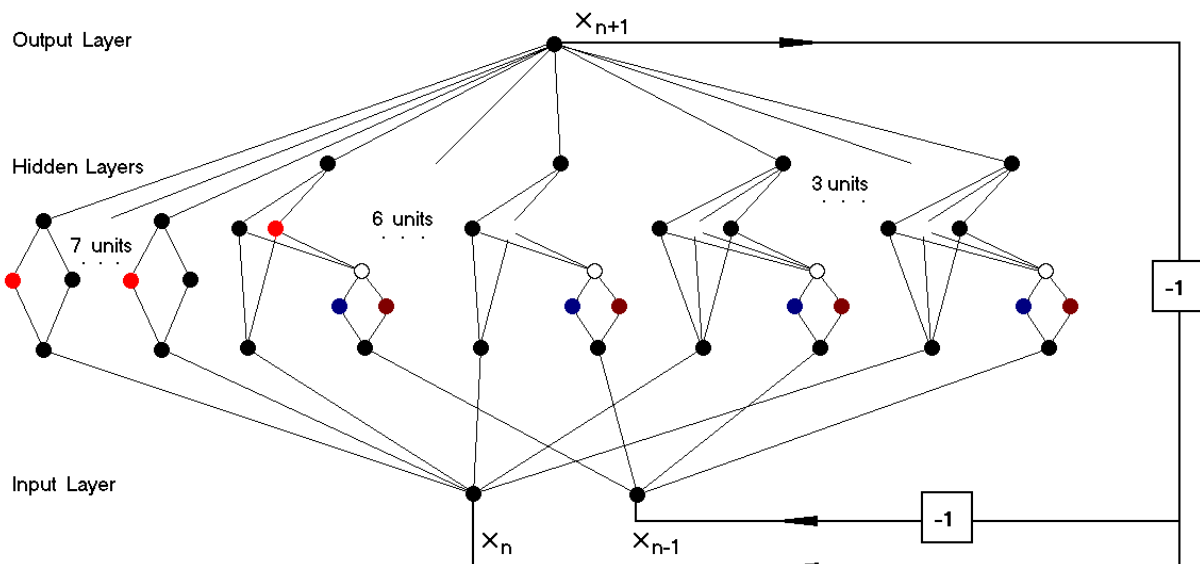


Fig.4. Neural net which realizes the dynamics of 2-D map storing the string 174.

Acknowledgment

This study is supported by an INTAS Grant (INTAS-94-2899).

References

- [1] I. Guyon, L. Personnaz, J. P. Nadal, G. Dreyfus: "Storing and retrieval of complex sequences in neural networks". *Phys. Rev. A*, Vol.38, pp.6365-6372, 1988.
- [2] D. Servan-Schreiber, A. Cluremans, J. L. McClelland: "Learning sequential structure in simple recurrent networks". In *Advances in neural information processing systems I*, ed. by D. S. Touretzky, Morgan Kaufman Publisher Inc., Palo Alto, CA, 1989.
- [3] W. J. Freeman: "Tutorial on neurobiology: from single neurons to brain chaos". *Int. J. Bifurcation and Chaos*, Vol.2, pp.451-482, 1992.
- [4] S. Dmitriev, A. I. Panas, S. O. Starkov: "Storing and recognition information based on stable cycles of one-dimensional maps". *Physics Letters A*, Vol.155, pp. 494-499, 1991.
- [5] Yu. V. Andreyev, Yu. L. Belsky, A. S. Dmitriev: "Storing and recognition of information using stable cycles of 2-D and multi-dimensional maps". *Radiotekhnika i Elektronika*, Vol.39, pp.114-123, 1994 (in Russian).
- [6] Yu. V. Andreyev, Yu. L. Belsky, A. S. Dmitriev, D. A. Kuminov: "Dynamic systems with chaos as a medium for storing and processing information". *Izvestiya Vuzov Radiophysica*, Vol.37, pp.1003-1019, 1994 (in Russian).
- [7] A. S. Dmitriev, D. A. Kuminov: "Recurrent neural networks with chaos for storing and retrieving information". *Proceeding of 2nd Int. Symposium on Neuroinformatic and Neurocomputers*, Rostov-Don, Russia, pp.24-31, 1995.
- [8] Yu. V. Andreyev, Yu. L. Belsky, A. S. Dmitriev, D. A. Kuminov: "Information processing using dynamic chaos. Neural networks implementation". *IEEE Transactions on Neural Networks*, Vol.7, pp.1-11, 1996.