Pattern recognition and learning in bistable CAM networks

Vladimir Chinarov (1), Martin Dudziak (2), Yuri Kyrpach (1)

(1) Nonlinear Dynamics Lab, Technolog SIE, 253144, Kiev-144, Ukraine
(2) Silicon Dominion, Inc., Richmond, USA

Abstract. The present study concerns the problem of learning, pattern recognition and computational abilities of a homogeneous network composed from coupled bistable units. New possibilities for pattern recognition may be realized due to the developed technique that permits a reconstruction of a dynamical system using the distributions of its attractors. In both cases the updating procedure for the coupling matrix uses the minimization of least-mean-square errors between the applied and desired patterns.

I. Introduction

The neural network approach is a traditional paradigm for simulating and analyzing the computational aspects of the dynamical behavior of complex neural-like networks concerning the problems of learning and pattern recognition in neural networks [1, 7, 8, 11]. Among others the Hopfield type models in which every fixed point corresponds to one of the stored patterns, are very convenient for the purposes of pattern recognition because of their high memory capacity and fast association.

Several dynamical models were also suggested in which network realizes other principles and ideas such as: a) gradient competitive dynamics [6] and its further generalizations that include short-range diffusive interactions and subthreshold periodic forcing [2, 13]; b) genetic algorithm that was used to evolve cellular automata to perform a particular computational task [11]; c) bistable network approach [3-5] which exploits the idea that ionic channel in biological membranes is a self-organized non-equilibrium dynamic system functioning in a multistable regime; oscillator network models such as relaxation oscillators with two time scales models [4, 5, 14] and stochastic bistable oscillator Hopfield-type network model [10].

In this study we propose a neural-like network model of a content-addressable memory which exploits the dynamics of coupled overdamped oscillators that move in a double-well potential. The main goal of this paper is to describe the dynamic properties of attractors that have been observed in numerical simulations. The proposed model describes the dynamics of a network composed of bistable elements. Coupling coefficients characterize pair-wise nature of interactions between network elements with all-to-all connections.

II. The model

Description of dynamics of the model is based upon the following equations:

$$\frac{d}{dt}\vec{x}(t) = -\frac{\partial H}{\partial \vec{x}}. \quad (1)$$

The model accounts for the connectivity of all elements of the network. We choose the effective free energy functional,

$$H = \overline{H} + H_{\text{int}}, \quad (2)$$

containing the homogeneous terms representing cubic forces acting on each element, $\overline{H}$, and the interaction H_{int}, terms, respectively

$$\overline{H} = -\frac{1}{4}\sum_{i=1}^{N} x_i^2 (2 - x_i^2), \quad (3)$$

$$H_{\text{int}} = -\frac{1}{2}\sum_{i,j=1}^{N} w_{ij} x_j x_i. \quad (4)$$

The network is composed of *N* coupled bistable elements that may be considered as neural-like network activities. Each element is an overdamped non-linear oscillator moving in a double-well potential $\overline{H}$, pair-wise interactions between all elements are given by Eq. (4). The network has the gradient dynamics with all limit configurations contained within the set of fixed point attractors. For a given coupling matrix w_{ij}, the network evolves towards such limit configurations starting from any initial input configurations of bistable elements (the elements are distributed among left and right wells of the potential with negative and positive values, respectively).

Let us consider several stable limit configurations as patterns to be stored by such a network. In the traditional neurodynamics paradigm the corresponding coupling matrix may be constructed according to the rule which is referred as Hebb's rule [9]

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu, \quad (5)$$

where *p* is a number of stored patterns, or the network may be trained (learned) using some iteration procedure in which a set of given patterns presented repeatedly to the network and coupling matrix elements adjusted according definite learning rule. For the considered network, we have used both these schemes.

III. Learning algorithm

In the framework of the standard steepest descent approach to construct the learning rule [9, 12], updating procedure for the coupling matrix may be obtained using an algorithm that minimize the mean squared error ε (MSE) between a given (desired) pattern d_i and some limit configuration $\bar{x}_i$

$$\varepsilon = \frac{1}{2}\sum_i \varepsilon_i^2 = \frac{1}{2}\sum_{i=1}^{N}(d_i - \bar{x}_i)^2 \tag{6}$$

In this approach the coupling matrix w_{ij} has to be updated via the following scheme

$$w_{ij}(k+1) = w_{ij}(k) + \eta\sum_r \varepsilon_r (L^{-1})_{ri}\,\bar{x}_j(k+1). \tag{7}$$

where k is the iteration step, parameter η determines the rate of learning [9], and matrix L is defined as

$$L_{ik} = \delta_{ik}(3\bar{x}_k^2 - 1) - w_{ik}\,, \tag{8}$$

This updating algorithm will be used for the network learning as well as for retrieving the stored memories when applied patterns are just corrupted memorized ones. The actual values of elements in all the patterns are not the bipolar (-1 or +1) ones, but they are real values established in the network when it reaches its final state. These values are stable states that correspond to the minima of the double-well potentials for each oscillator. It should be underlined here, that fixed-point attractors in our case do not coincide, like in all Hopfieldian-type networks, with the corners of hypercube,.

The patterns (key patterns $\xi_{0\mu}$) from which the coupling matrix w_{ij}^0 is constructed using Hebb's rule given by Eq. (5), differ from the limit solutions $\bar{x}_j$ of Eq. (1).

During the learning phase based on a scheme given by Eq. (15) the key patterns $\xi_{0\mu}$ were taken as the target patterns and stored patterns with distortions (the signs of 10% of elements were inverted and small random noise was add to all the elements) were taken as the applied patterns. In this case the initial values of the weight coefficients were constructed using the applied patterns according to Hebb's rule given by Eq. (5). The iteration learning procedure is constructed in such a way that applied patterns repeatedly presented one by one, but the next is presented only after the weight coefficients are adjusted according to the learning rule given by Eq. (7). The learning procedure lasts until MSE criterion ε defined by Eq. (6) will be less or equal some given small value. In all our simulations the obtained matrix practically coincides with the matrix w_{ij}^0 constructed from the key patterns $\xi_{0\mu}$. We should underline here that in the coupling matrix constructed so far, all diagonal matrix elements w_{ii} are nonzero values.

In Fig. 1 the dependence of MSE criterion ε on the number of iterations needed to learn the coupling matrix W_{ij} for a network with N=20 elements is shown. In Fig. 2 the dependence of the sum of all matrix elements $\sum_{i,j} W_{ij}$ on the number of iterations for the network with N=20 units when only first pattern ξ_1 is used to learn the network is given. Both dependencies characterize the rate of convergence process during the learning phase.

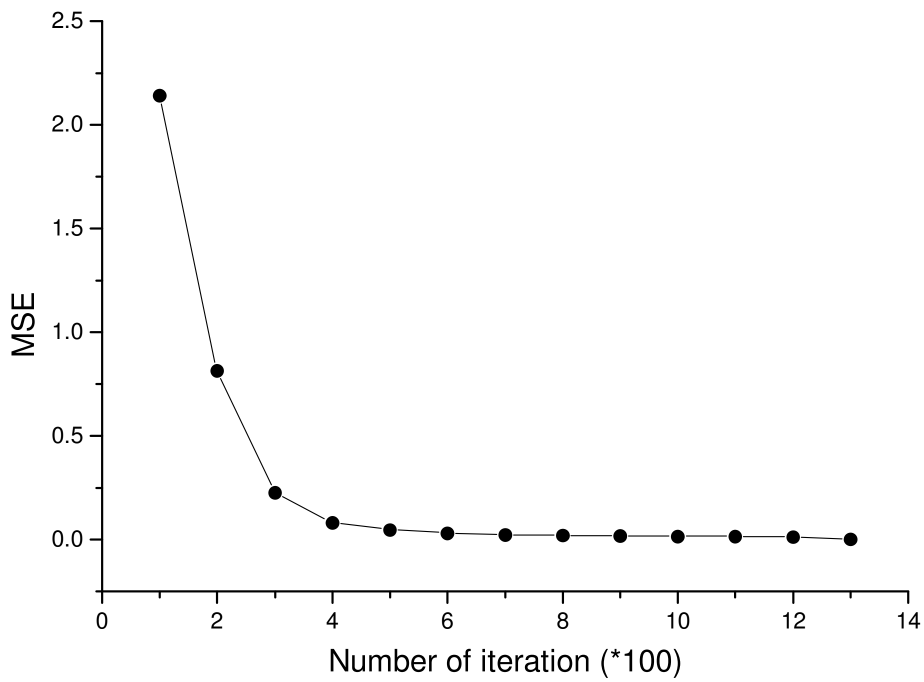

Fig. 1. Dependence of MSE on the number of iterations needed to learn the coupling matrix W_{ij} for a network with N=20 elements

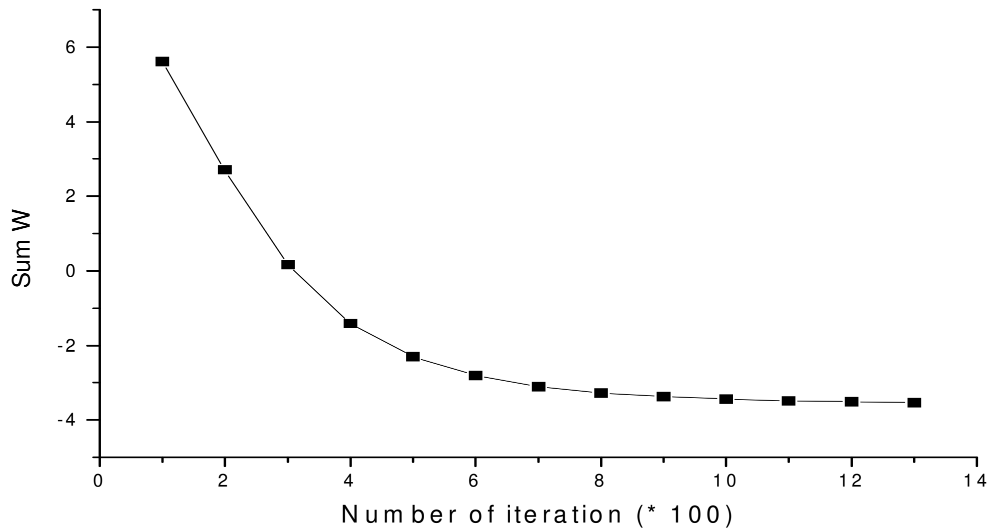

Fig. 2. Rate of convergence of the learning phase. Dependence of the sum of all matrix elements $\sum_{i,j} W_{ij}$ on the number of iterations for the network with N=20 units when first pattern ξ_1 is used to learn the network.

IV. Network Performance

The configuration of N elements that have positive and negative values is taken as a pattern. In Fig. 3 a first example of the network performance is shown. Seven patterns ($\xi_1,.....,\xi_7$) were memorized in the network consisting of N=8 coupled bistable elements. Black circles depict element's negative value (left well of the double-well potential is occupied by this element), white circles depict element's positive value (right well of the double-well potential).

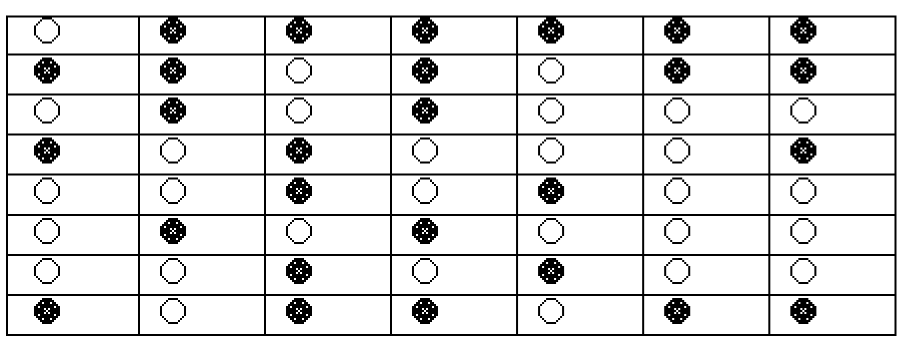

Fig. 3. Example of seven patterns memorized in the network consisting of N=8 elements. Black circles depict element's negative values (left wells of the double-well potential), white circles depict element's positive values (right wells of the double-well potential).

Note that the initial values for each element within input configurations that were used to learn the network to memorize these seven patterns differ from their final values. Being memorized, each of these patterns could be easily retrieved when an input sequence with random values of elements is applied to the network. In all our simulations we have used normally distributed random values with zero mean. We note that besides these memorized patterns only the patterns fully antisymmetrical to those shown in Fig. 3 could be retrieved. The latter have inverted signs of all elements with respect to the original pattern and therefore have the same energy. No other spurious states corresponding to linear combinations of stored patterns will appear during retrieval process. During the retrieve phase we have applied also a pattern that was a result of overlap of all stored patterns. In this case the network may recall only one pattern possessing the smallest energy or corresponding antisymmetrical one.

The network is characterized also by extremely fast convergence to the desired patterns when elements of the input patterns are distorted up to 30% (by inversion of their signs) in case of using the fixed learned coupling matrix. These distortions should remain each pattern within the basin of attraction of corresponding fixed-point attractor. In this case the network immediately will reach the latter.

To test the association performance of the network we use different applied (test) patterns that differ from the stored (desired) patterns by opposite values of some elements. Examples of the restoration of different distorted patterns (memorized patterns ξ_7, ξ_4, and ξ_6) in the network with N=8 elements by updating the coupling matrix w_{ij} are shown in Figures 4 and 5. In all these cases applied patterns have 50% of distorted elements (their signs are inverted). In each panel first column corresponds to the desired pattern and second column to the applied one. Other columns depict configurations of elements in each successive iteration of w_{ij} that is needed to achieve target state.

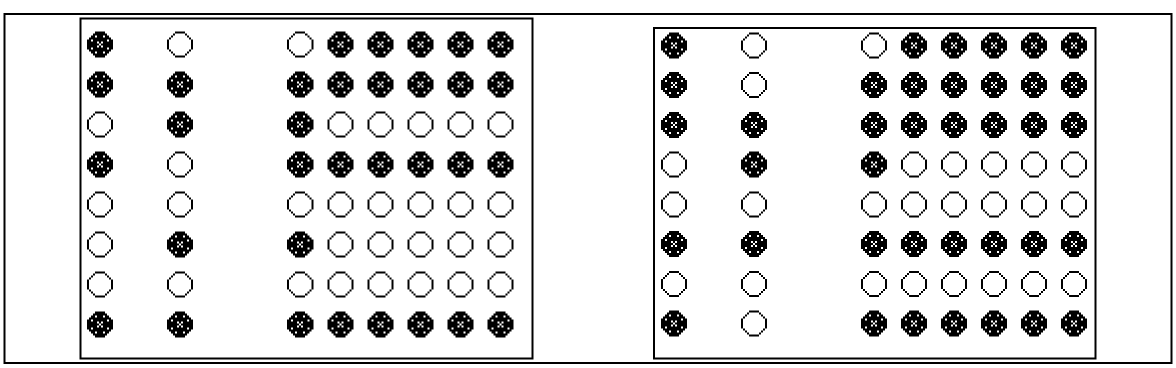

Fig. 4. Example of restoration of different distorted patterns (ξ_7 – left panel, ξ_4 - right panel) by updating the coupling matrix w_{ij}. In these applied patterns 50% of elements are distorted (signs are inverted). In each panel first column corresponds to desired pattern and second column - to applied

pattern. Other columns depict configurations of elements in each successive iteration that is needed to achieve target state (desired pattern).

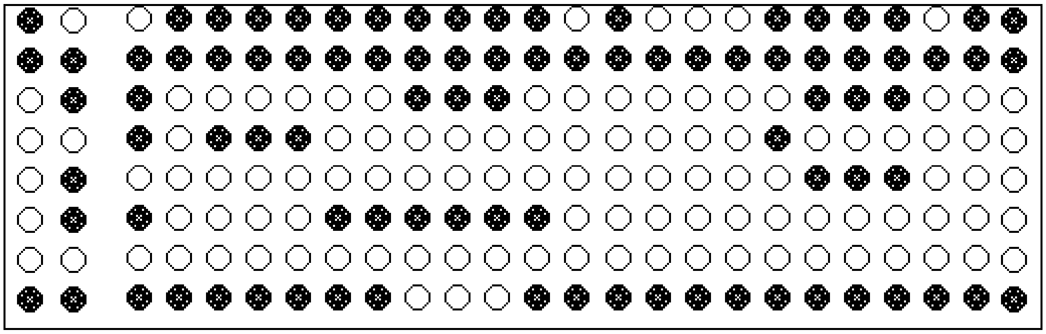

Fig. 5. Example of restoration of sixth distorted pattern (ξ_6) with 50% of distortions.

Figure 6 shows the result of restoration of the sixth pattern (ξ_6 – first column) corrupted with 62% of distortions (second column). Top and bottom panels show all the iterations from the beginning to the end. More iterations are needed in this case in comparison with previous cases shown in Figures 4 and 5 where 50% of distortions of input patterns were taken.

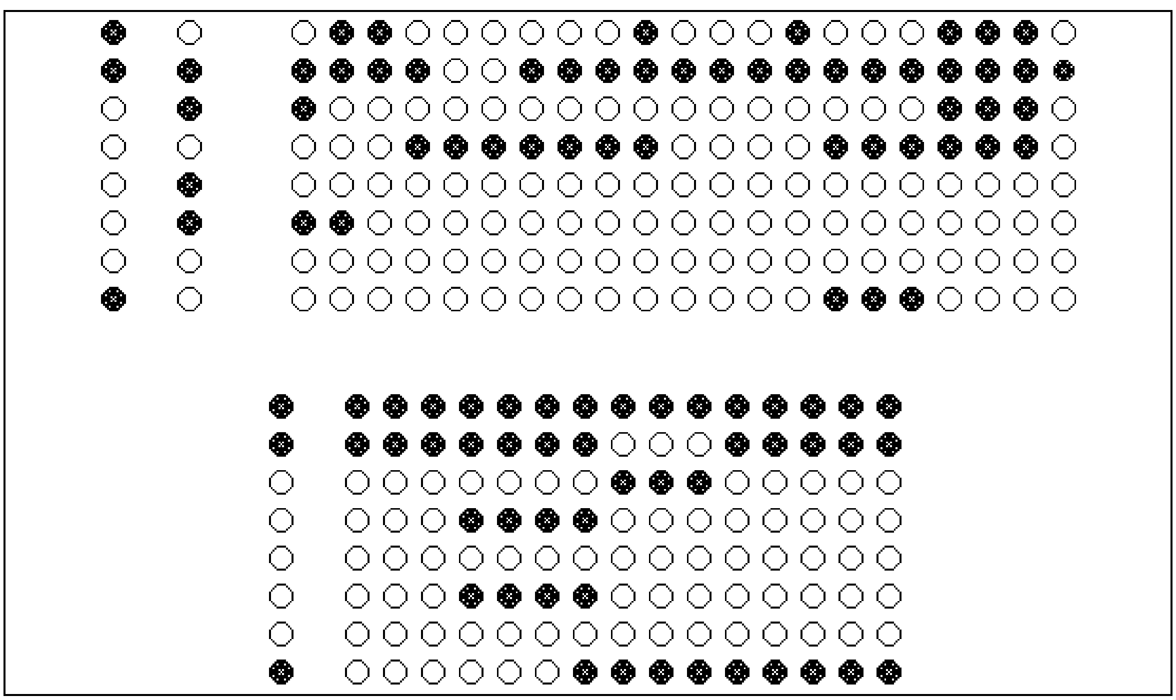

Fig. 6. Example of restoration of the sixth pattern (ξ_6) with 62% of distortions (second column). Top and bottom panels show all the iterations from the beginning to the end.

Only a few iterations are needed to restore within the accuracy defined by a given MSE all the values of the elements of each pattern. The number of iterations needed to restore only signs of each element is substantially less then in the case when the values of the elements are restored, not only their signs. Next example of the network performance is shown in Fig. 7. Five patterns ($\xi_1,\ldots,\xi_5$) were memorized in the network consisting of N=20 coupled bistable elements.

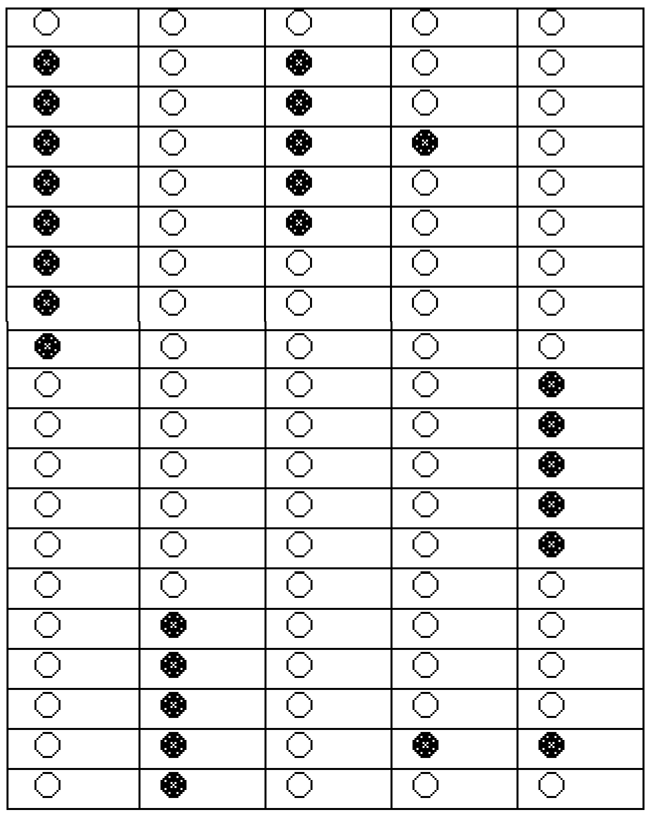

Fig. 7. Example of five stored patterns ($\xi_1,\ldots,\xi_5$) in the network consisting of N=20 elements. Each column represent one pattern. Black circles depict element's negative values, white circles - element's positive values.

Examples of the restoration of the distorted patterns in the network with N=20 elements by using the algorithm of updating the coupling matrix w_{ij} are shown in Figures 8 and 9. These patterns differ from the memorized patterns ξ_2 and ξ_3 by inversion of the signs of 40 % (Fig. 8), or 50%. (Fig. 9) of elements. Second case (50%. of distortions) needs much more iterations to restore desired pattern.

Being restored at the *n*-th iteration the pattern remains stable in the succeeding iterations. On the other hand, during the retrieve performance one could frequently observe that at some iteration step pattern coincides with a desired one (like in the fourth column of Fig. 5) but the retrieval process does not stop at that moment. The reason is that in all considered restoration network performances not only the signs of all elements that constitute given pattern should be recovered, but also their values should be close to those from the desired pattern. The network meanwhile should move in the weight space along the gradient in the weight space w_{ij} until MSE criterion ε will exceed some small given value. Therefore, the iteration process will proceed further from this iteration point. In all the simulations shown in this section the value of ε was taken as 0.1. It is worthy to note that at such MSE values the network converges actually to its stable (limit) configuration that corresponds to the desired pattern. And the retrieve algorithm will stopped just at that moment of the iteration procedure.

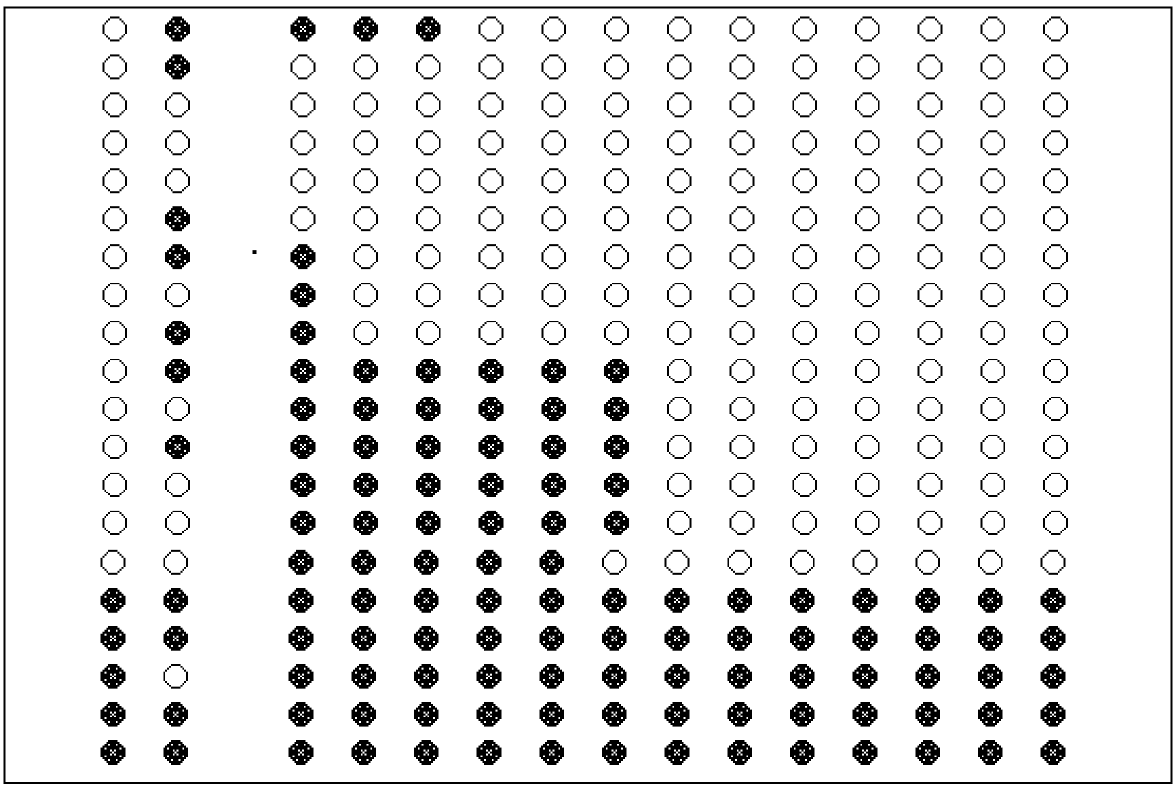

Fig . 8. Restoration of the second distorted pattern (ξ_2) for the network with *N*=20 elements. In applied patterns 8 elements (40%) are distorted (by inversion of their signs). First column - desired pattern, second column - applied pattern. Only several iterations are shown in other columns. Pattern is restored after 245 iterations (seventh column). Black circles depict element's negative value (left well), white circles depict element's positive value (right well).

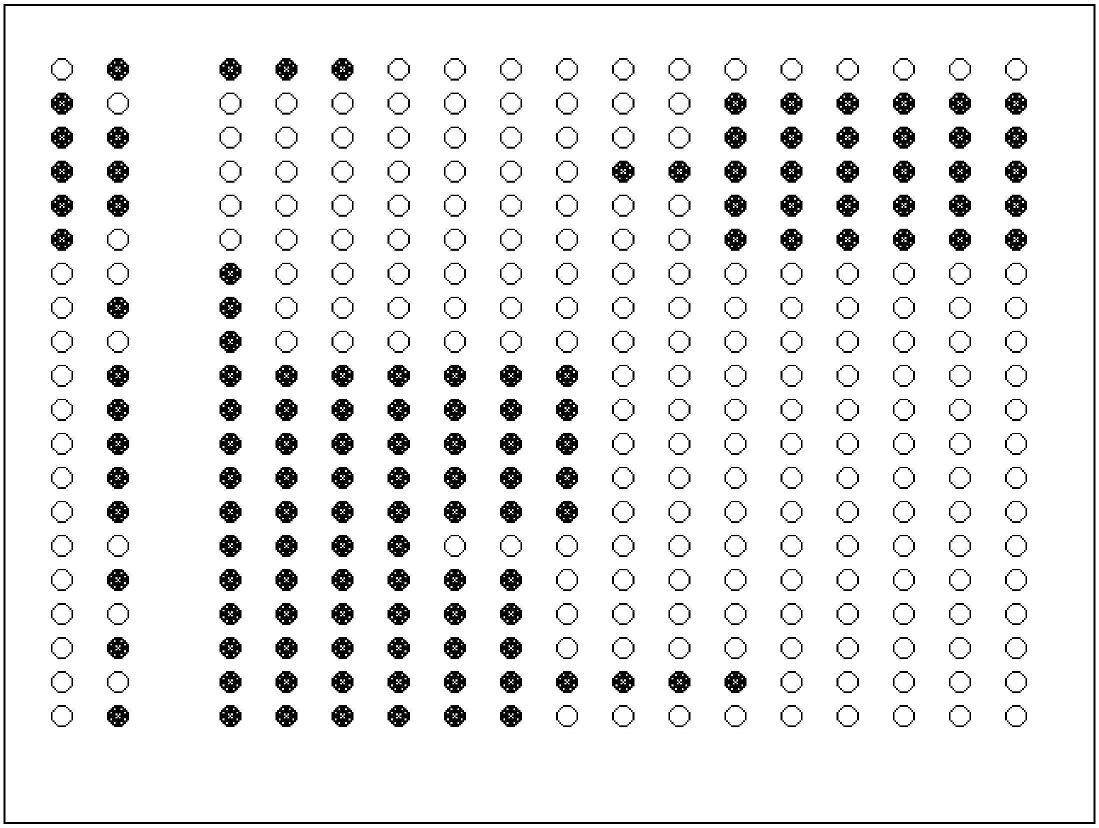

Fig . 9. Restoration of the third distorted pattern (ξ_3) for the network with N=20 elements. In applied patterns 60% of elements are distorted (by inversion of their signs). First column - desired pattern, second column - applied pattern. Only several iterations are shown in other columns. Pattern is restored after 1500 iterations (eleventh column).

The next point we wish to emphasize is that if the matrix w_{ij} will be multiplied by a coupling strength coefficient γ, the rate of convergence to the desired pattern during the retrieval phase may increase as γ increases. Unfortunately, such an effect depends on the relation between the values of strength coefficient γ, MSE criterion ε, and learning rate parameter η. One should look for the optimal choice of these parameters in each case trying to speed up the retrieval process. It could be shown also that the restoration of all (inverted) signs is achieved much faster than practically full restoration of the element's values when ε tends to zero.

It is interesting to underline that considered network may successfully provide perfect associative memories retrieve in a case when the coupling matrix w_{ij} that was previously learned to memorize several patterns, is distorted up to 20%-25%. The corresponding distortion was made in the following way: for a network with N=20 elements the coupling coefficients were affected by the white noise or put to zero for each site remote from the given one on more than 13 -15 neighboring sites.

It should be noted that considered bistable network may function successfully also in a case when all diagonal matrix elements w_{ii} have zero values (like in the traditional Hopfield-type networks).

V. Conclusion

The results presented in this paper concerns the computational possibilities of a network consisting of coupled bistable units that may store (for the fixed coupling matrix) much more memory patterns in comparison with the Hopfield-type neural networks. These new possibilities may be realized due to the proposed learning algorithm that may induce the system to learn efficiently. Using known patterns with up to 45%- 50% of distortions, the coupling matrix may be fully reconstructed. In some sense, the developed technique resembles a reconstruction of the dynamical system using its attractors.

For an example, a network composed of N coupled bistable units for a fixed coupling matrix has several stable fixed-point-like attractors that are associated with the memorized patterns. If these patterns and the coupling matrix are known, the applied patterns taken as the initial values for the dynamical system may be restored in few iterations. If some applied patterns belongs to another set (say, they were obtained as a fixed-point attractors for different coupling matrix), they would be easily recognized giving another resulting coupling matrix that is updated during the retrieval phase. Therefore, the identification of memories (attractors) could be easily done.

It was shown in our simulations that the applied patterns with 45% of distortions may be effectively restored. If only memorized patterns are known the coupling matrix will be reconstructed after several iterations. In both cases the updating procedure for the coupling matrix uses the minimization of the least-mean-squares errors between the applied and desired patterns.

From a computational point of view the proposed network offers definite advantages in comparison with the traditional Hopfield-type networks. It has good performance, it may learn efficiently, and has bigger memory capacity. In examples described above we have seen that the network with N=8 elements may store seven stable patterns that may be perfectly retrieved even when half of its elements are distorted by inversion of their signs. It should be said also that the performance of our learning algorithm depends on a judicious choice of the rule parameter η. It' s worthy to underline that the network may operate also in a noisy environment.

References

1. Bishop C.M. Neural Network for Pattern Recognition, Clarendon Press, Oxford, 1995.
2. Bressloff P.C., P. Popper. Stochastic dynamics of the diffusive Haken model with subthreshold periodic forcing, Phys. Rev. E 58, 2282-2287, 1998.
3. Chinarov V.A. *et al*. Self-organization of a membrane ion channel as a basis of neuronal computing, In: Proc. Intern. Symp. on Neural Networks and Neural Computing, pp. 65-67, September 1990, Prague.
4. Chinarov V.A. *et al*. Ion pores in biological membrane as a self-organized bistable system, Phys. Rev. A 46, 1992, 5232-5241.

5. Chinarov V. Modelling self-organization processes in non-equilibrium neural networks, SAMS, 30, 311-329, 1998.
6. Dudziak, M., Quantum Processes and Dynamic Networks in Physical and Biological Systems, The Union Institute, 1993
7. Dudziak, M. and Pitkanen, M., Quantum Self-Organization Theory with Applications to Hydrodynamics and the Simultaneity of Local Chaos and Global Particle-Like Behavior, Acta Polytechnica Scandinavica, 91, 1998
8. Haken H. Synergetic Computer and Cognition, Springer-Verlag, Berlin, 1991.
9. Halici U. Reinforcement learning in random neural networks for cascaded decisions, Journal of Biosystems, 40, 83-91, 1997.
10. Halici U., G. Ongun. Fingerprint classisfication through self organizing maps modified to treat uncertainties, Proc. IEEE, 84, 1497-1512, 1996.
11. Haykin S. Neural Networks, Macmillan College Pub. Co, Inc. New York, 1994.
12. Han S.K., W.S. Kim, H. Kook. Temporal segmentation of the stochastic oscillator neural network, Phys. Rev. E 58, 2325-2334,1998.
13. Mitchell M., J.P. Crutchfield, P.T. Hraber. Evolving cellular automata to perform computations: Mechanisms and impediments, Physica D 75, 361-391, 1994.
14. Pineda F.J. Dynamics and architecture in neural computation, Journ. of Complexity, 4, 216-245, 1988.
15. Shmutz M., W. Banzhaf, Robust competitive networks, Phys. Rev. A 45, 4132-4145, 1992.
16. Terman D., D.L. Wang. Global competition and local cooperation in a network of neural oscillators, Physica D 81, 148-176, 1995.