Modular Neural Associative Memory Capable of Storage of Large Amounts of Data*

A.M. Reznik and O.K. Dekhtyarenko
The Institute of the Mathematical Machines and Systems, Ukrainian National Academy of Science
03187, 42 Glushkov Str, Kiev, Ukraine
neuro@immsp.kiev.ua

***Abstract*–A new neural net architecture based on the Hopfield network is proposed. This architecture overcomes the memory limitation that is peculiar to a single network at the cost of moderate computational expenses. Parameters' influence on read-write processes is considered, possible read errors are defined and estimations for associative recall effectiveness as a function of search complexity are given. Theoretical estimations are in close correspondence with experimental results obtained for random vectors dataset.**

INTRODUCTION

Associative memory (AM) provides data access using data content, in contrast to addressable memory that uses address value instead. More specifically, AM is divided into auto- and heteroassociative memory types. The first type stores keys and can be used in filtering and information recovery tasks, while the second one is able to store key-value pairs and is used in various classification and mapping problems.

If AM is robust with respect to possible distortions in input data, then it can operate with incomplete or inexact information. Such properties are inherent to neural AM based on the Hopfield network model [1,2], which is a multistable feedback system. The net output X, starting from a state defined by the net input, evolves as follows:

$$X_{i+1} = sign(CX_i), \qquad (1)$$

where C is a weight (synaptic) matrix $(n \times n)$;
n – net dimension (number of neurons);
sign – sign function with $\{-1,1\}$ codomain.

Subject to positive diagonal elements of C and its symmetry, the process of net state change, called convergence, always stops in stable state – the attractor. Having attractors coincide with memorized data, the convergence process from the initial state (net input) to the nearest attractor performs function of associative search for the best match among stored data.

The first algorithm for C calculation was suggested by Hopfield and has limitation for number of stored data vectors: $m < 0.14n$. Violation of this ratio leads to the appearance of false attractors (i.e. stable states that do not correspond to any of stores vectors) and destruction of the associative memory. Pseudoinverse (projective) learning algorithm [3], based on exact solution of net stability equation, allows an increase in this ratio to $m < 0.25n$ that is half of theoretical limit $m = 0.5n$ [4]. Later, the desaturation method for synaptic matrix has moved this theoretical limit to $m = n$ [5] and provided neural AM operation with $m \le 0.75n$ [6].

A basic drawback of neural AM lies in the dependence on data dimension of maximum memory capacity. In order to increase this capacity, common practice is to synthetically extend stored data dimension. This results in an impetuous growth of demands to physical memory and computational resources needed (quadratic dependence relative to n).

This drawback can be overcome by substitution of one big network for a set of smaller ones, with some manner of information distribution among them. Such a principle is used in the suggested modular associative neural network.

MODULE NETWORK READ-WRITE ALGORITHMS

Modular Associative Neural Network (MANN) is a set of Hopfield networks combined in a binary tree structure (Fig. 1). Each Hopfield network, termed module, is learnt using pseudoinverse algorithm, thus having weights matrix equal to projective matrix on a linear subspace spanned onto stored vectors. Difference coefficient d is used as a criterion for data distribution among modules. It characterizes the norm of orthogonal component of input vector relatively to linear subspace of vectors already stored in the module. For the i-th module and input vector X it is defined as:

$$d_i(X) = \|X - C_i X\|^2 / \|X\|^2 = (X \cdot (I - C_i)X)/n, \qquad (2)$$

where the last expression uses projective properties of C and bipolar values of X's components. The codomain of d is $[0,1]$.

Each module stores not more than m vectors, and net infill starts from the root module. To memorize each new vector X, one has to find a module in which to store it. In order to do

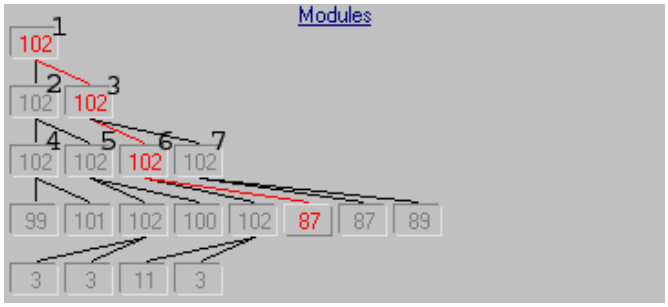

Fig. 1. The structure of module tree.

*This research was supported by INTAS-01-0257

this the path is built starting from the root according to the following rule:

$$i := \begin{cases} 2i, & d_i(X) < t \\ 2i+1, & d_i(X) \geq t \end{cases}, \quad (3)$$

where i – number of module concerned (root has $i = 1$);
t – fixed threshold value.

The search is carried out until it reaches the first partially filled module (with number of stored vectors less then m). This is the module to store input vector X.

At MANN reading phase for the given input vector X one has to find the module that may contain this vector. The search tree is built analogously to write phase, but now branching is allowed – after each considered i-th module one or two next level modules are included into tree:

$$i := \begin{cases} 2i, & d_i(X) < t - \varepsilon \\ 2i+1, & d_i(X) \geq t + \varepsilon \\ \{2i, 2i+1\}, & d_i(X) \in [t-\varepsilon, t+\varepsilon) \end{cases}, \quad (4)$$

where ε is a half-width of uncertainty interval.

A module from the search subtree, which has the smallest d value, is considered as a module containing the prototype of input vector X. After this module is found, usual associative recall procedure is carried out with it and input vector X.

Values of t and ε parameters influence AM read-write processes. The value of t defines the extent of module tree balancing. Values too small would lead to right subtree domination relative to any module (including the root module), too big – left subtree. Such a situation is disadvantageous as it results in extensive search subtrees at read phase, hence more computational resources are needed. The median of d's probability distribution may be used as an optimal value for parameter t, but this value essentially depends on the nature of stored data.

The value of ε defines the branching intensity of the search subtree at AM read phase. There is no branching when $\varepsilon = 0$, while $\varepsilon = 1$ results in the search subtree coinciding with entire module tree.

As long as input vector X can be inexact at AM read phase, values d_i for all modules of the net may be different from those calculated at the write phase. This may lead to incorrect module selection and, thus, erroneous net output. Following two reasons may cause incorrect selection:

1. Path error – when the search subtree does not pass through module containing input vector;
2. Belonging error – when module containing input vector is included into the search subtree, but is not selected as module with a minimum d value.

Selection of ε value affects probabilities of these errors. The larger ε value is, the greater search subtree will be, resulting in lower probability of path error and higher probability of belonging error.

DISTRIBUTION OF DIFFERENCE COEFFICIENT

The MANN read-write processes depends on the nature of input data and in general case it is hard to predict it *a priory*. Nevertheless it is possible to make some estimations for the particular type of data, that is often used as a model example in associative memory investigations. These estimations can help to reveal some common regularities of MANN functioning.

Let us assume that a set of data being memorized consists of n-dimensional vectors with independent random components having equiprobable values *{-1,1}*. Each module stores m vectors. With pseudoinverse learning rule weight matrix elements have normal distribution [6]. The mean value of C's diagonal elements is m/n, non-diagonal – 0. Element dispersion is defined in [7] as:

$$D(c_{ij}) = m(m-n)/n^3. \quad (5)$$

It is possible to find distribution of values derived from difference coefficient d using the distribution of C elements. These values are defined by input data, so, according to limit theorem with large enough data dimension n, their distributions can be considered as normal ones.

If $G = I - C$ is a designation for projective matrix onto orthogonal supplement of stored vectors, then the first two moment coefficients for difference coefficient d are:

$$\begin{aligned} E(d) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{n} E(x_i g_{ij} x_j) = 1 - m/n; \\ D(d) &= \frac{1}{n^2} D\left(\sum_{i=1}^{n}\sum_{j=1}^{n} x_i g_{ij} x_j\right) = D(c_{ij}) = m(n-m)/n^3. \end{aligned} \quad (6)$$

As long as the median of normal distribution is equal to its average, $E(d)$ defines optimal threshold t value, providing balances module tree formation at writing phase.

Network behavior at the read phase is dependent on characteristics of d change under the influence of noise. Suppose that input vector X, that is not contained in a given module, has been affected by the noise of intensity h, i.e. sign of its random h components has been reversed. The noisy vector can be represented as $X + S$, where vector S has exactly h nonzero components with absolute values of 2 and signs opposite to the signs of corresponding components of vector X. The increment of d is:

$$\Delta d = \left(\|G(X+S)\|^2 - \|GX\|^2\right)/n = \left((S \cdot GS) + 2(X \cdot GS)\right)/n. \quad (7)$$

Distribution of Δd has conditional nature but for the sake of simplicity we neglect its dependence on the initial value of $d(X)$. Having equiprobable signs Δd must have zero average with the dispersion:

$$D(\Delta d) = \frac{1}{n^2}\left(D\left(\sum_{k=1}^{h}\sum_{l=1}^{h} s_{i_k} g_{i_k j_l} s_{j_l} \right) + 4D\left(\sum_{i=1}^{n}\sum_{l=1}^{h} x_i g_{ij_l} s_{j_l} \right)\right)$$
$$= \frac{16\left(h^2 + nh\right)}{n^2} D\left(c_{ij}\right). \quad (8)$$

If vector X is stored in given module, then $G \cdot X = 0$, and Δd_0 distribution has following parameters:

$$\Delta d_0 = \left(s \cdot Gs\right)/n;$$
$$E\left(\Delta d_0\right) = \frac{1}{n} E\left(\sum_{k=1}^{h}\sum_{l=1}^{h} s_{i_k} g_{i_k j_l} s_{j_l} \right) = \frac{1}{n} E\left(\sum_{k=1}^{h} s_{i_k}^2 g_{i_k i_k} \right) = \frac{4h}{n}\left(1 - \frac{m}{n}\right);$$
$$D(\Delta d_0) = \frac{1}{n^2} D\left(\sum_{k=1}^{h}\sum_{l=1}^{h} s_{i_k} g_{i_k j_l} s_{j_l} \right) = \frac{16h^2}{n^2} D\left(c_{ij}\right). \quad (9)$$

Probabilistic Estimations of Read Process

Once we know the probability distribution of *d*-associated values, it is possible to find probabilities of basic events playing an important role in MANN read-write processes.

A path error appears if the following event occurs in at least one module from the search subtree:

$$\begin{bmatrix} d(X) < t,\ d(X+S) \geq t + \varepsilon \\ d(X) \geq t,\ d(X+S) < t - \varepsilon \end{bmatrix}$$

The probability of such an event (jump) is

$$P_j = \int_0^t f_d(y)\left[\int_{t+\varepsilon-y}^{1} f_{\Delta d}(z)dz \right]dy + \int_t^1 f_d(y)\left[\int_{-1}^{y-(t-\varepsilon)} f_{\Delta d}(z)dz \right]dy \quad (10)$$
$$= 2\int_0^t f_d(y)\left[\int_{t+\varepsilon-y}^{1} f_{\Delta d}(z)dz \right]dy.$$

Now let the search subtree contain *i*-th module with the prototype of input vector $X + S$. Belonging error occurs if at least one of the rest search subtree modules has a difference coefficient value less than the *i*-th module has. It happens with a probability

$$P_b = P\left\{d_j(X+S) < d_i(X+S)\right\} = \int_0^1 f_{\Delta d_0}(y)\left[\int_0^y f_d(z)dz \right]dy. \quad (11)$$

If V_i denotes the subset of modules from the *i*-th level in a tree that has *l* levels, then path error probability for a random stored vector is:

$$1 - P_{path} = \sum_{i=1}^{l} \left(1 - P_j\right)^{i-1} P\{x \in V_i\}$$
$$= \frac{1}{2^{l-1}} \sum_{i=1}^{l} \left(1 - P_j\right)^{i-1} 2^{i-1} = \frac{\left[2\left(1 - P_j\right)\right]^l - 1}{\left(2^l - 1\right)\left(1 - 2P_j\right)}. \quad (12)$$

If no path error occurs when a search tree is constructed, then the probability of belonging error during module selection is:

$$1 - P_{belonging} = \left(1 - P_b\right)^{r-1}, \quad (13)$$

where *r* is a number of modules in the search subtree.

In some modules the search subtree can split with a probability of:

$$P_s = \int_{t-\varepsilon}^{t+\varepsilon} f_d(y)dy \quad (14)$$

and expectable search subtree size is:

$$r = \sum_{i=1}^{l} \left(1 + P_s\right)^{i-1} = \frac{\left(1 + P_s\right)^l - 1}{P_s}. \quad (15)$$

This value defines the computational complexity of MANN read process, as value *d* has to be computed for every module of the search subtree except leaf ones. Thus it takes the order of rn^2 operations for the complete read process to be executed.

Experimental Results

Expressions obtained for error probabilities and read process complexity have been experimentally verified with model dataset using NeuroLand neurocomputing program [8].

The numerical experiment used a set of vectors with dimension $n = 256$ and with random independent components possessing equiprobable *{-1,1}* values. Each module stored $m = 102$ vectors that corresponded to 40% of the memory saturation. The desaturation coefficient [5] was set to *0.1*. Noise level $h = 33$ corresponds to the full attraction radius value of a single network, i.e. the maximum data deformation that can be removed by the net during the convergence process (note that the concept of full radius of attraction is different from the attraction radius used in [3,5,6] to denote the maximum Hamming distance overcome by the net at the last convergence step). This value of noise intensity allows to characterize MANN read quality using only module selection criterion. Threshold parameter value was assigned using (6). To construct theoretical dependencies using (12) and (15) the *l* value used was as follows:

$$l = \log_2\left(\left(M/m\right) - 1\right), \quad (17)$$

where *M* is a total number of vectors stored in the MANN.

The first series of experiments was aimed to obtain probability of incorrect module selection at MANN read phase as a function of the total number of vectors stored. The half-width of the uncertainty interval was set to $\varepsilon = 0.01$. The

following probabilities of key events were obtained (in parentheses experimental value is given) using (10,11,14):

$$P_j = 0.2477\,(0.2275);$$
$$P_b = 1.438 \cdot 10^{-17}\,(0);$$
$$P_s = 0.2557\,(0.1836).$$

Fig. 2 depicts the experimental and theoretical dependencies of read error probability. Theoretical dependence is slightly overestimated for large values of MANN infill. It is caused by a greater theoretical value of P_j than the experimental one. There were no belonging errors revealed during experimentation that corresponded to vanishing theoretical values of P_b.

The purpose of the second set of experiments was to investigate MANN behavior as the half-width of uncertainty interval ε changes. The growth of ε results in search subtree expansion. It leads to a decrease in the number of path errors, but at the same time, it may result in a greater probability of a belonging error. The growth of ε is also associated with greater search complexity, which is defined as a ratio of average search subtree size for a given value of ε to average search subtree size without branching, i.e. when $\varepsilon = 0$.

Experiments were carried out with a network containing $M = 3000$ vectors ($l \cong 5$, 35 modules). Data reading was performed for different ε values. Fig. 2 and 3 depict experimental and theoretical dependencies of error probability and search complexity as functions of ε. Comparison of these dependencies allows for selection of an acceptable ε value as a compromise between quality and complexity of the reading procedure.

As in the first set of experiments, there were no belonging errors revealed for any of the ε values. The theoretical path error estimation is also slightly greater than the experimental values and the relative difference increases along with the ε value.

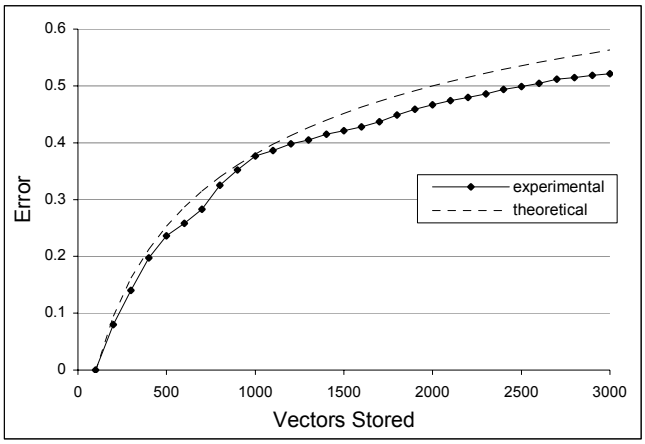

Fig. 2. Read error as a function of net infill

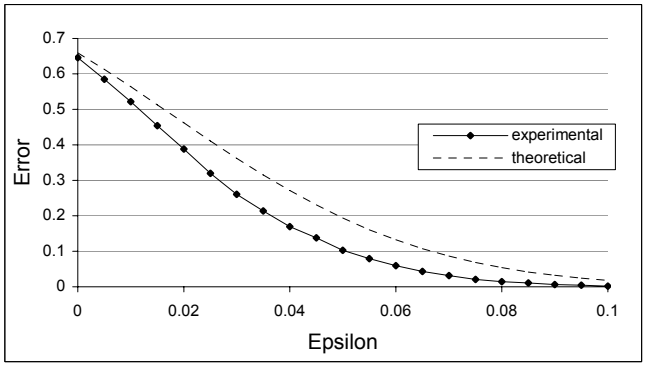

Fig. 3. Read error as a function of ε.

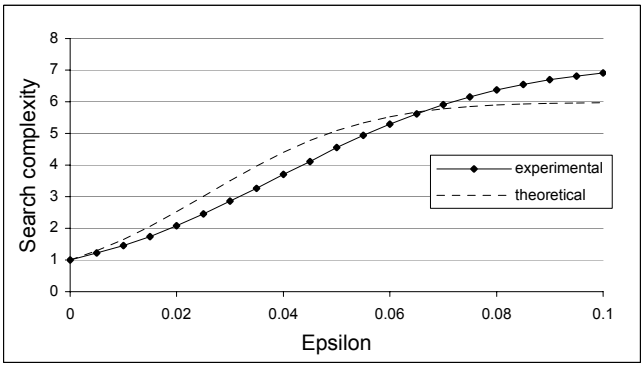

Fig. 4. Search complexity as a function of. ε.

CONCLUSIONS AND FUTURE WORK

The considered model of modular associative neural network provides nearly linear dependence of necessary physical memory resources from a number of stored vectors. In its ability to remove input data artifacts in the convergence process, it maintains the main advantage of the Hopfield network. Proposed model is superior to the cellular associative network, in which the number of connections number is also linearly dependent on net size [9]. Though sparse weight matrix of the cellular net can have a tape structure, the net capacity is defined by tape width and does not depend on net size [10]. Therefore, an associative cellular net is almost equally effective as fully connected Hopfield network.

Another important advantage of the modular associative memory is its ability to be used with modules of heteroassociative type. As the first layer of the two-layer heteroassociative network performs the autoassociative memory function, the nature of the module selection process during read-write operations remains the same. Having complete freedom in the selection of the second layer structure and functionality, it is possible to store any kind of data using binary keys for the associative search.

The expressions obtained allow estimation of character of read-write processes without direct implementation of the net. It can be used for quick net parameters selection, which provides optimal values for some particular task.

Nevertheless, the range of application of these expressions to real-world problems is unknown as their data can have asymmetry and/or strong correlation in contrast to the model data.

The proposed model obviously has great potential for further research and improvement. Other criteria of data distribution among modules that take into account input data properties should be considered. In addition, the tree formation process and its dependence on data order merit deeper investigation. Such a process, similar to the self-organization of the Kohonen's net, may turn out to be a more efficient clustering algorithm.

REFERENCES

[1] J.J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *in Proc. Nat. Acad. Sci.,* vol. 79, pp. 2554-2558, Apr. 1982.

[2] B. Kosko, "Bi-directional associative memories," *IEEE Trans. Syst., Man, Cybern.,* vol. 18, no. 1, pp. 49-60, Jan/Feb 1987.

[3] L. Personnaz, I. Guyon, G. Dreyfus, "Collective computational properties of neural networks: New learning mechanisms," *Phys. Rev. A.,* vol. 34, no. 5, pp. 4217-4228, 1986.

[4] M. Weinfield, "A fully digital integrated CMOS Hopfield network including learning algorithm," *in Proc. Int. Workshop WLSI Art. Intell.,* Univ. of Oxford, E1-E10, 1988.

[5] A.M. Reznik, D.O. Gorodnichy, A.S. Sitchov, "Regulating feedback bond in neural networks with learning projectional algorithm," *Cybernetics and system analysis*, vol. 32, no. 6, pp. 868-875, 1996.

[6] D.O Gorodnichy., A. M. Reznik, "Increasing attraction of pseudo-inverse autoassociative networks," *Neural Processing Letters,* vol. 5, no. 2, pp. 123-127, 1997.

[7] A.S. Sitchov, "Weight selection in neural networks with pseudoinverse learning rule," (in Russian) *Mathematical Machines and Systems,* vol.2, pp. 25-30, 1998.

[8] A. M. Reznik, E.A. Kalina, A.S. Sitchov, E.G. Sadovaya, O.K. Dekhtyarenko, A.A. Galinskaya, "Multifunctional neurocomputer NeuroLand," (in Russian) *in Proc. Int. Conf. Inductive Simulation,* Lviv, Ukraine, vol. 1(4), pp. 82-88, May 2002.

[9] M. Bruccoli, L. Carnimeo, G. Grassi, "Heteroassociative memories via cellular neural networks," *Int. J. Circuit Theory Appl.,* vol. 26, pp. 231-241, 1998.

[10] O.K. Dekhtyarenko, D.W. Nowicki, "Associative memory based on partially connected neural networks," (in Russian) *Proc. 8th all-Russian Conf. Neurocomp. Appl.,* Moscow, pp. 934-940, March 2002.