*Bull. Nov. Comp. Center*, Comp. Science, 9 (1998), 107–112


# Limiting capability of cellular-neural associative memory

S. Pudov

The problem of achieving Cellular-Neural Associative Memory (CNAM) limiting capability is considered. At first a CNAM learning method based on the idea of Perceptron Learning Rule which provides maximal ability to restore distorted patterns is suggested. Next, expressions for determining self-connection weight values which increase attractivity and decrease the number of oscillation states are obtained. Finally, influence of neuron threshold on basic characteristics of CNAM is investigated. It is shown that CNAM is capable to store more than $2q$ patterns where $q$ is the cardinality of neuron neighborhood.

## 1. Introduction

Cellular-neural associative memory is the associative memory by Hopfield [1] with connection structure like that of cellular automaton. Restriction of connection number greatly simplifies implementation of CNAM but it accordingly decreases storing and retrieving capability of such network. There are only three ways of influence on CNAM characteristics when connection structure is fixed. The first is to use an appropriate learning rule, for example [2]. This method is very complex since it requires singular matrix decomposition. The second is to correct neurons self-connection weights [3] which for some learning rule improves network capability to restore distortions. The third is to define a threshold value for each neuron – it can be equal to zero or not. It can be noticed that at first each of this ways does not exclude the use of others. Secondly, the first two ways have been studied more than the last while non-zero threshold was chosen arbitrarily in most cases depending on learning rule used.

In this paper the results of investigation of capabilities of cellular-neural associative memory are presented. After CNAM formal model presentation (Section 2) and problem statement (Section 3) a CNAM learning method based on the idea of Perceptron learning rule [4] is suggested in Subsection 4.1. It increases the network capability to restore stored pattern (*prototype*) distortions. This method inherits such properties as simple realization and guarantees the prototype's individual stability. Next, an algorithm for neuron self-connection weights calculation is suggested in Subsection 4.2. It can be used for already learned CNAM and provides acceleration of convergence time, reduction of oscillations, and improvement of the ability to restore distortions. Finally (Subsection 4.3), the results obtained from investigation and comparison of properties of two CNAM models: with zero and non-zero threshold respectively, are described. Main accent was made on local properties of a neuron but not on a global network behavior.

## 2. Formal model representation

Let us describe the associative memory model with the help of cellular-neural network formalisms [5]. Following accepted notions CNAM is defined by a set: $\boldsymbol{N} = \langle \boldsymbol{C}, \boldsymbol{W}, \boldsymbol{\Phi} \rangle$, where $\boldsymbol{C}$ is a rectangular $m \times n$ array consisting of cells (or neurons) with the states $c_{ij} \in \{-1, 1\}$; $\boldsymbol{W} = \{W_{ij}\}$ is a set of *weight vectors* of the form $W_{ij} = (w_1, \dots, w_q)$, $w_k$ denotes a real number characterizing the connection between the neuron with the coordinates $(i, j)$ and its $k$-th neighbor; $\boldsymbol{\Phi}$ is the CNAM functioning rule.

For each neuron $(i, j)$ a set of other neurons, which communicate with it, forms its neighborhood consisting of $q$ pairwise different neighbors without itself. The states of any neuron $(i, j)$ neighbors are represented as a state neighborhood vector $C_{ij} = (c_1, \dots, c_q)$. Further, a vector $D_{ij} = c_{ij}C_{ij} = (d_1, \dots, d_q)$ called *normalized state neighborhood* (*normalized neighborhood* for short) of neuron $(i, j)$ will be also used. Both vectors $C_{ij}$ and $W_{ij}$ have $q$ components each with their numeration in agreement ($w_i$ is a connection weight with the $i$-th neighbor), therefore their scalar product can be defined as $\langle C_{ij}, W_{ij} \rangle = \sum_l c_l w_l$.

The rule $\boldsymbol{\Phi}$ of CNAM operation is described by a following iterative procedure:

**Procedure.** Let $C(t)$ be an array after $t$-th iteration, $C_{ij}$ and $W_{ij}$ are neighbor states and weight vector of a neuron $(i, j)$, respectively. Then:

Step 1. Each neuron in $C(t)$ computes the following function:

$$f(C_{ij}, W_{ij}) = \begin{cases} 1 & \text{if } g_{ij} \geq 0, \\ -1 & \text{otherwise,} \end{cases}$$

and its result becomes the state of this neuron $(i, j)$ at the iteration $t + 1$.

Step 2. If $C(t+1) = C(t)$ then $C(t) = \boldsymbol{\Phi}(C(0))$ is the result of calculations which corresponds to a stable state of CNAM.

Two models of CNAM to be investigated differ in the function $g_{ij}$ only, which in general looks like $g_{ij} = \langle W_{ij}, C_{ij} \rangle + B_{ij}$, where $B_{ij}$ is referred to as *bias*. In the neuron model called **Model 1**, $B_{ij} = 0$. In the other model called **Model 2**, $B_{ij}$ is non-zero. For uniformity, $B_{ij}$ is replaced by a fictitious neighbor with the state equal always to +1, the connection between neuron $(i, j)$ and this neighbor having the weight equal to $w_{ij} = B_{ij}$. So, in this way Model 2 is restricted to Model 1 with $q + 1$ neighbors (without bias!).

## 3. Problem definition

Since it was shown how using fictitious neighbor, Model 2 can be viewed as the Model 1, then without loss of generality further only neurons without bias will be considered. It is also suggested that CNAM connection structure and initial set of prototypes $P^0, \ldots, P^{L-1}$ are fixed. Ideally, the synthesis of CNAM (it consists of neuron models choosing, memory learning, and self-connection weights correction) should provide the following characteristics:

(1) *Individual stability*, i. e., each prototype should be a stable state of CNAM ($\Phi(P^K) = P^K$).

(2) *Attractivity*, i. e., for each prototype $P^K$ *the basin of attraction* (all patterns retrieved by network as $P^K$) should be maximal.

(3) *Minimum of oscillations*, i. e., the number of final network states like this: $\ldots, C^1, C^2, C^1, C^2, \ldots$ (it is a cycle consisting of two patterns $C^1$ and $C^2$) should be as small as possible.

From this requirements the problem was defined as follows: how can maximal CNAM attractivity be obtained? Also an additional question appears: how many prototypes can be stored in CNAM with predefined neuron neighborhood? The first question demanded considerable theoretical investigations is described further in this paper. The second question can be reduced to the following one: how many linear separable prototypes [6] exist for a chosen CNAM? The answer was found by simulation of learning process. Prototype set contained images of symbols from English and Russian alphabets and arabic numbers drawn in thin lines (1 pixel width). Simulations showed that CNAM can store more than 50 prototypes when cardinality of neuron neighborhood is equal to 24. In this case the relation (number of prototypes) / (number of neighbors) can be more than 2, but this network has relatively small attractivity. Particularly, the following experiment was made for CNAM consisting of $20 \times 20 = 400$ neurons: for each prototype 400 patterns differing from it in one neuron state were input in turn into CNAM for retrieving. The original prototype was correctly retrieved in 60-70% cases only, i. e., distortion of one neuron state leads retrieval process out of initial prototype in 30% cases.

So, one more question was added to two previously put: how many prototypes can be stored in CNAM so that it ability to restore 1-distortions can be guaranteed? Here the word "1-distortion" means a pattern differing from the appropriate prototype in not more than one cell state at each neuron neighborhood. It can be noticed that this two patterns can differ in states of $N/q$ cells. In this case the number of distortions can be sufficiently large but they are to be placed uniformly in such way that each neuron has one distortion in its neighborhood only. The answer on this question was found also by simulation of CNAM storing the set of patterns mentioned above. For CNAM in which each neuron has 24 neighbors the following result was obtained: such a network is capable to correct all 1-distortions when there are 10–12 prototypes stored in it, i. e., the relation $L/q$ is approximately equal to 1/2. In this experiment the method of learning and the algorithm for self-connection weight calculation given in the next section was used.

## 4. Learning of CNAM

**4.1. Method of learning.** Since the connection structure is fixed, the only way to provide the required CNAM characteristics is to determine connection weights between neurons. It concerns also two particular weights: a self-connection and a bias one. As compared with correction of self-connection weight or choosing neuron model, a method of CNAM learning maximally influences the network characteristics because it defines almost all neuron connection weights. So, the properties of learning method have a great significance, and besides one more requirements is added to those mentioned above:

(4) The method of learning should be *cellular-neural*: we require that the learning procedure is performed by the CNAM itself, i. e., there should be no global interactions during the learning process.

From the four requirements mentioned above the perceptron learning rule satisfies two of them: 1 and 4, so it was chosen as basis for the new method to be elaborated. Satisfaction of requirement 2, i. e., increasing attractivity, was the main problem. It was solved in the following way. As it is known [7] the greater the values

$$m_{ij}^K = \langle D_{ij}^K, W_{ij} \rangle,$$

in learned CNAM, the better is network correction of prototype distortion. From perceptron learning rule [4] one can see that it guarantees $m_{ij}^K > 0$ $\forall (i,j) \in M$, $\forall K = 0, \ldots, L-1$ in the case of learning process termination, i. e., when the prototypes are individually stable [4]. During learning by this method the value $m_{ij}^K$ is compared with zero; that is enough for ensuring individual stability of prototypes.

The idea of new learning method is to compare $m_{ij}^K$ not with zero but with some positive value $\alpha_{ij}$. In this case if learning process terminates then all $m_{ij}^K$ are greater than $\alpha_{ij}$, and a certain level of attractivity is guaranteed. The problem of choosing such an $\alpha_{ij}$ which provides the ability to restore distorted patterns as large as possible is solved in [8]. It can be noticed that the complexity of the method increases a little as compared with perceptron learning rule: each neuron computes two more values than earlier.

Simulations show that the suggested method ensures good optimization of $m_{ij}^K$ when a number of learning iteration is 2–3 times more than for perceptron learning rule termination.

**4.2. Calculation of self-connection weights.** So, the above learning method satisfies all requirements except the third one, because like perceptron learning rule it does not define self-connection weights which stabilize associative memory behavior. Therefore, the decision about self-connection weights calculation after learning process is completed, was accepted. This not only reduces the number of oscillations but also greatly increases the attractivity of already learned CNAM. Simulations showed that the ability of CNAM learned by proposed method to restore 1-distortions is increased up to 10–20 times for the set of patterns mentioned above. The method of self-connection weights calculation was described in details in [9, 10]; some simulation results was published in [8].

**4.3. Comparison of Model 1 with Model 2.** Between the results of investigation of neuron bias influence on main CNAM characteristics, the following can be distinguished. At first, it was shown that the addition of neuron bias to CNAM improves its abilities for data storing. Next, it was obtained that Model 2 (with bias) has greater ability to restore distortions as compared with Model 1. Finally, the learning method suggested in Subsection 4.1 was modified for learning Model 2, and its convergence was proved. Some simulation were made for investigation of influence of neuron bias on the ability to restore 1-distortion of stored patterns.

## 5. Conclusion

The problem of achieving CNAM limiting capability is solved in this paper. A method of learning which provides maximal ability to restore distorted patterns is suggested. Probability of improving of CNAM characteristic by self-connection weights correction and neuron bias taking into account was investigated. It was shown theoretically and by simulation that CNAM is capable to store more than $2q$ patterns where $q$ is the cardinality of neuron neighborhood. It is recommended that the relation $L/q$ should be about $1/2$ when $q = 16\ldots25$ for retrieving of 1-distortions of stored patterns.

# References


[1] J.J. Hopfield and D.W. Tank, *Computing with Neural Circuits: a Model*, Science, **233**, 1986.

[2] D. Liu and A.N. Michel, *Sparsely interconnected neural networks for associative memories with applications to cellular neural networks*, IEEE Transactions on Circuits and Systems–II: Analog and Digital Signal Processing, **41**, No. 4, 1994, 295–307.

[3] D.O. Gorodnichy, *Desaturating coefficient for projection learning rule*, Lecture Notes in Computer Science, **1112**, 1996, 469–476.

[4] F. Rosenblatt, *Principles of Neurodynamics*, Washington, Spartan, 1959.

[5] O.L. Bandman, *Cellular-neural computations. Formal model and possible applications*, Lecture Notes in Computer Science, **964**, 1995, 21–35.

[6] O.L. Bandman and S.G. Pudov, *Stability of stored patterns in cellular-neural associative memory*, Bulletin of the Novosibirsk Computer Center. Series: Computer Science, issue 4, 1996, 1–18.

[7] M. Cottrell, *Stability and attractability in associative memory networks*, Biological Cybernetics, **58**, 1988, 129–139.

[8] S.G. Pudov, *Learning of cellular-neural associative memory*, Avtometria, **2**, 1997, 107–120 (in Russian).

[9] S.G. Pudov, *Influence of self-connection weights on the stability of cellular-neural associative memory*, Avtometria, No. 1, 1998, 90–96 (in Russian).

[10] S.G. Pudov, *Influence of self-connection weights on cellular-neural network stability*, Lecture Notes in Computer Science, **1277**, 1997, 76–82.