Associative Cellular Neural Networks
with Adaptive Architecture[*]

Oleksiy Dekhtyarenko, Alexander Reznik, Alexander Sitchov

Institute of Mathematical Machines and Systems,
Glushkov ave., 42, Kyiv 03187, Ukraine
Phone/fax: +380-44-266221/2666457, email: olexii@mail.ru

ABSTRACT: In this paper we introduce a new model of associative memory based on cellular neural network with adaptive architecture. This model outperforms networks with a fixed architecture that is set independently on data; in addition it has better associative recall properties than the fully connected network with weight selection procedure. Apart from the direct experimental comparison of considered learning algorithms an indirect comparison based on weight matrix spectra is made.

1. Introduction

Cellular Neural Networks are composed of simple partially connected threshold units (neurons). Due to partial connectedness, this type of network is considered to be hardware friendly (see e.g. [1], [2]). If neurons can have various sets of weights (unlike a network consisting of completely identical neurons, [3]) then such rather simple model can exhibit nontrivial functions of associative memory. One of the first examples in this field [4] uses matrix pseudoinversion technique. In our work [5] we studied in detail this algorithm and showed that it can be modified by abandoning the step that provides symmetry of weight matrix. It achieves better network performance without loosing its stability for feasible range of parameters and also eliminates the requirement for the network architecture to be symmetrical.

Weight selection algorithm (WS) in [6] is another leading motive for the given work. It allows a fully connected Hopfield network to perform well even after the removal of 80% of certain neuron connections. The location of the remaining connections is apparently of great importance for the associative properties of the network and reflects some hidden interrelationships in stored data. This selection rule can be used to set the architecture of the cellular network and we can expect that such a network will have better characteristics than a network with some other architecture set in advance.

2. Algorithm

Fully connected Hopfield network [7] is the first neural implementation of associative memory. It uses multistability inherent to a system with feedback connections. Stable states correspond to stored data. The network starts from some initial state during the so-called convergence process (subsequent changes of states) and ends up in one of stable states that is considered as an autoassociative response.

For a network with n neurons and state vector x the convergence process can be expressed as:

$$x_{t+1} = f(Cx_t), \qquad (1)$$

where C – $(n{\times}n)$ weight matrix of the network;
f – neuron activation function.

The best associative recall of a Hopfield network nowadays is achieved using a projective learning rule [8]. In this rule the weight matrix is found as a solution of stability equation

$$CV = V, \qquad (2)$$

where $V = \{v_1,...,v_m\}$ – matrix formed by m stored data vectors.
Solution of (2) can be found as:

$$C = VV^+, \qquad (3)$$

where operator "$^+$" stands for a pseudoinversion operation (see e.g. [9]). The weight matrix found using (3) equals to the projective matrix onto a subspace spanned on stored vectors.

WS algorithm [6] lies in the construction of a weight matrix using (3) and further retaining only those weights that have the maximum absolute values. The rest of the weights are set to zero. It has been shown in [6] that fully connected Hopfield network with small memory infill ($m \cong 0.1n$) still possesses associative memory functions having only 20% connections if they are selected using this principle.

Unlike WS algorithm where the allocation of neuron connections is known only after the data set to be stored in the network is available, a pseudoinverse (PI) learning rule for cellular neural network [4] allows forming weight matrix using net architecture specified in advance. Let this architecture be represented by configuration matrix W. This is $(n{\times}n)$ matrix whose elements have values $\{0, 1\}$ depending on absence or presence of a connection between correspondent neurons. In the given work for the sake of simplicity we do not consider neuron thresholds (it has been shown in [5] that introducing one more variable into equations (7) using thresholds allows to store exactly one more data vector without deterioration of associative recall quality).

For the i-th neuron the stability equation (3) has a form

$$C^i V = V^i, \qquad (5)$$

where
C^i – row vector of i-th neuron weight coefficients;
V^i – row vector formed from stored data.

To allow for structural restrictions imposed by cellular architecture we consider the selection operator that changes the second dimension of a matrix:

$$W^i : (l \times n) \rightarrow (l \times q(i))$$

$$\forall l \in N, \; q(i) = \sum_{j=1}^{n} W_{ij} \qquad (6)$$

Operator W^i retains only those columns of its matrix argument that correspond to nonzero elements of i-th row of configuration matrix W. The stability equation (5) can be rewritten as following:

$$W^i(C^i) \cdot W^i(V^T)^T = V^i \qquad (7)$$

The optimal solution of (7) (with the minimum norm subject to the minimum discrepancy) can be found by using a pseudoinversion operator:

$$W^i(C^i) = V^i \cdot \left(W^i(V^T)^T\right)^+. \tag{8}$$

The proposed algorithm is an extension of PI algorithm. It is further referred to as PI adaptive algorithm; it lies in setting the architecture using a WS algorithm and further network training using PI learning rule for cellular neural network (8).

2. Experimental Set-Up

Experimental comparisons of the algorithms were carried out using 256 dimensional data vectors ($n = 256$) with random independent components having equiprobable values {-1, 1}. The sign function with {-1, 1} domain was used as activation function in (1). Network performance is characterized by its attraction radius – the maximum extent of input data corruption that can be completely removed by the network during the convergence process. Input data are corrupted by reversing sign of r randomly selected components.

The attraction radius was determined experimentally. In order to find it the stored dataset was replicated 100 times with the random distortion of given intensity. Corrupted data were fed to the neural network input and after the stop of convergence process the intensity of corruption was increased or decreased depending on absence or presence of error in the network's output. The search of attraction radius was carried out using a bisection method. After getting its value there was a check to determine if the network was also capable of tolerating the distortions with intensities of r-1, r-2, r-3. This was done to take into account some slight non-monotonous of dependence of final network error on the intensity of corruption. The non-monotonous is caused by the amount of considered variants of distortion being substantially less than the amount of all possible distortions.

It has been shown in [11] that the weakening of neuron direct feedback connections (represented by diagonal elements of weight matrix C) can increase the attraction radius of fully connected networks. This approach was called the desaturation technique. Neuron feedback weights are weakened using multiplication by $0 < \alpha < 1$ value that is called desaturation coefficient. Its optimal value for the fully connected network learnt by the projective rule is $\alpha = 0.15$. However too small value of desaturation coefficient leads to the network instability (i.e. infinite convergence process (1)) and, consequentially, results in the deterioration of associative properties of the network.

For cellular networks it also makes sense to use the desaturation technique, but the appropriate value of α depends on network configuration and infill, therefore it should be chosen experimentally. As it is rather tedious process we used the following empirical estimation:

$$\alpha = 0.15m/trace(C), \tag{9}$$

This formula gives $\alpha = 0.15$ for the projective network (as it has $trace(C) = m$, see next chapter) and value that is inversely proportional to weight matrix trace for the PI network. Such estimation appeared to be close to the optimal α value that has been proven experimentally.

In the experimental part of this work while testing all networks trained in accordance with considered algorithms we used desaturation technique with α values being set using (9).

3. Experimental Results

The dependence of attraction radius on network connectivity degree (relative amount of connections that are allowed to exist by given architecture) was determined in the course of the experiment. At first such dependencies were obtained for networks trained using WS algorithm and PI adaptive algorithm. Networks were trained to store $m = 26$, 51 and 77 data vectors (memory infill 10%, 20% and 30% of network dimension).

Dependencies depicted in fig.1 show that usage of PI adaptive algorithm provides better quality of associative recall compare to the network trained by the WS rule. This superiority takes place for network connectivity values up to 0.4 ($m = 26$) - 0.6 ($m = 77$). For higher connectivity values the difference between algorithms (in terms of attraction radius) vanishes. It corresponds to the fact that a PI algorithm, being formally applied to the fully connected network, would result in weight matrix that is equal to one obtained using the projective learning rule (3).
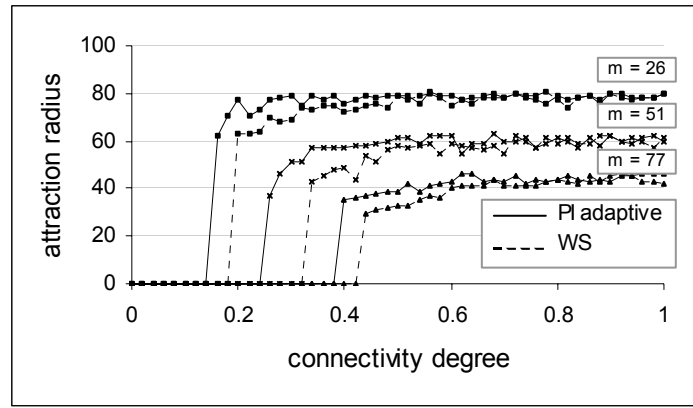

Figure 1: Attraction radius vs. connectivity degree

Also it should be mentioned that for the extremely low values of network connectivity the usage of PI adaptive algorithm allows to get a network with nonzero attraction radius using the amount of connections 5-15% less than it requires using WS rule.

To make sure of the significance of proper architecture selection, the same experiment with $m = 51$ data vectors was carried out for a network with the fixed architecture (one dimensional neuron allocation, only neurons within a certain distance from each other are connected).
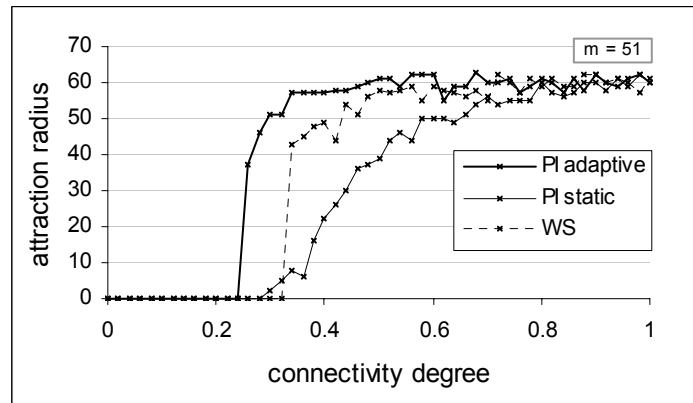

Figure 2: Comparison of three learning algorithms

It can be concluded from fig.2 that though such a network provides nonzero attraction radius for smaller values of connectivity degree compare to WS algorithm, but further improvement of its associative properties has almost linear behaviour.

Due to the symmetry of projective matrix used for weight selection the selected architecture is always symmetric too. But the usage of PI network training in a form of (8) abandons the symmetry of weight values and does cause network instability (absence of convergence) for high values of input distortions. Meanwhile there is a belief that within basins of attraction of stored vectors the network performs as strictly contracting mapping thus eliminating any possible instability for the amounts of input distortion it can tolerate. We are currently working on the theoretical proof of this hypothesis.

4. Indirect comparison of learning algorithms

The quality of associative recall for various learning algorithms and architectures can be compared indirectly by comparing ordered sets of weight matrix eigenvalues (the spectra). This technique was used in [12] to compare fully connected networks with projective learning rule. Spectrum of such networks can be considered as an ideal one. It consists of ones and zeros that correspond to stored data vectors and their orthogonal supplement respectively. During a step in the convergence process the component of current network state lying in the subspace of stored data remains unaffected, while its orthogonal supplement is suppressed. Restrictions on network architecture result in deviation from such shape; the stronger this deviation is, the worse network performance can be expected.

It is seen from fig. 3 that weight selection algorithm affects both 0 and 1 values of the spectra while pseudoinverse learning algorithm preserves 1 value unaffected (due to the exact solution of (2)). Because of the unsymmetrical matrix of networks trained using PI there are complex values in the spectra, but due to a slight degree of asymmetry their imaginary parts are about 10% of the real ones (only real parts are depicted at fig. 3).
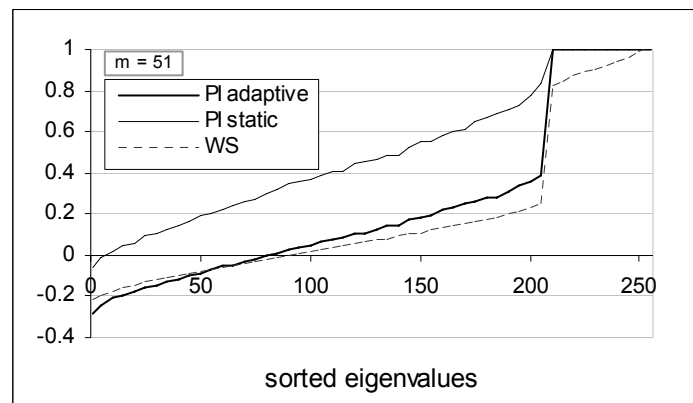

Figure 3: Weight matrix spectra

Among the considered algorithms, PI algorithm with static architecture differs the most from the step-like form spectra. It corresponds to the experimental results. PI adaptive algorithm has larger deviations in the nonzero part of the spectra compare to WS, but it has better associative recall properties because it provides the exact solution of (2).

5. Conclusion

Based on the results of the given work we can conclude that network architecture that is constructed using information about the data to be stored results in better network characteristics than using some other preset architecture.

The proposed algorithm can be applied to the hardware implementations of associative memory based on Cellular Neural Networks, however this implementation must allow the existence of auxiliary located connections. If architecture restrictions are not only of quantitative but also of topological nature (for example only nearby neurons can be connected) then this algorithm is not applicable in its pure form. Still if this rigid task formulation allows certain freedom in architecture selection then the proposed algorithm can be modified to provide a selection of the architecture that results in a network with the best associative recall properties.

References

[1] J. Austin: "The cellular neural network associative processor, C-NNAP". Proc. of 5th IEEE Int. Conf. on Image Processing and its Applications, Edinburgh, Scotland, pp. 622-626, July 1995.

[2] Sungjun Park, Joonho Lim and Soo-Ik Chae: "Digital Implementation of discrete-time cellular neural networks with distributed arithmetic". Proc. of Int. Conf. on Neural Networks, Houston, Texas, USA, pp. 959-963, June 1997.

[3] L. O. Chua, L. Yang: "Cellular Neural Networks: Theory". IEEE Trans. on Circuits and Systems, Vol.35, pp. 1257-1272, 1988.

[4] M. Brucoli, L. Carnimeo, G. Grassi: "Discrete-time Cellular Neural Networks for associative memories with learning and forgetting capabilities". IEEE Trans. on Circuits and Systems, CAS 42(7), pp. 396-399, 1995.

[5] O.K. Dekhtyarenko, D.W. Nowicki: "Cellular-structured neural associative memory". Mathematical Machines and Systems, Vol.3, pp. 37-44, 2002.

[6] A. Sitchov: "Weights selection in neural networks". Mathematical Machines and Systems, Vol.2, pp 25-30, 1998.

[7] J.J. Hopfield: "Neural networks and physical systems with emergent collective computational abilities". Proc. Nat. Acad. Sci., Vol.79, pp. 2554-2558, April 1982.

[8] L. Personnaz, I. Guyon, G. Dreyfus: "Collective computational properties of neural networks: New learning mechanisms". Phys. Rev. A, Vol.34(5), pp. 4217-4228, 1986.

[9] A. Albert: Regression and the Moore-Penrose Pseudoinverse, Academic Press, New York and London, 1972.

[10] A.M. Reznik: "Non-iterative learning for neural networks". Proc. of Int. Joint Conf. on Neural Networks, Washington, USA, Paper no. 548, July 1999.

[11] D.O. Gorodnichy, A.M. Reznik: "Increasing Attraction of Pseudo-Inverse Neural Networks". Neural Processing Letters, Vol.7, pp. 121-125, 1997.

[12] A.M. Reznik, A.S. Sitchov, O.K. Dekhtyarenko, D.W. Nowicki: "Associative Memories with "Killed" Neurons: the Methods of Recovery". Proc. of Int. Joint Conference on Neural Networks, Portland, Oregon, USA, pp. 2579-2582, July 2003.