Phase transition in sparse associative neural networks

Oleksiy Dekhtyarenko[1]*, Valery Tereshko[2] and Colin Fyfe[2]

1- Institute of Mathematical Machines and Systems - Dept. of Neurotechnologies
Glushkov Ave. 42, Kyiv 03187 - Ukraine
2- University of Paisley - School of Computing
Paisley PA1 2BE - Scotland, UK

Abstract. We study the phenomenon of phase transition occurring in sparse associative neural networks, which is characterized by the abrupt emergence of associative properties with the growth of network connectivity. It is shown that this discontinuous behaviour is caused by the specific way of architecture selection. Based on empirical results the relationship among critical parameters is suggested.

1 Introduction

Sparse (or diluted) associative neural networks attract a lot of attention for a number or reasons: they are more suitable for hardware implementation than their fully-connected counterparts; they require less memory and are faster to operate during computer simulations; lastly, their architecture looks more feasible from a biological point of view.

Though much research has been done in this field, the phenomenon of phase transition that occurs during change of architecture was discovered only recently.

In [1], the authors consider a McCulloch-Pitts model with a random architecture characterized by a connectivity parameter. They show that network activity (fraction of active neurons when network reaches its steady state) undergoes a discontinuous change from zero to a finite value as a critical value of connectivity is reached.

The efficiency of the network as an associative memory is studied in [2], [3]. The network is based on a small-world architecture and is shown to exhibit a phase transition as the network architecture changes from a regular lattice to a random graph.

Like the authors of [2], [3], we are interested in studying sparse recurrent networks as a model of associative memory. But we take a different approach to selecting network architecture. For a particular value of the network connectivity, we select the architecture that depends on a given set of desirable memory patterns. This method of architecture selection combined with the Pseudo Inverse learning rule was first suggested in [4] and allows us to improve network performance subject to a limited number of connections. A very distinct transition between non-remembering and remembering phases of a network occurs as connectivity grows. We will show that the architecture selection criterion causes this phenomenon.

* Supported by INTAS YSF 03-55-795.

2 The Associative Model

2.1 The Network

We consider a Hopfield-type sparse associative neural network, consisting of *n* neurons. Neuron *j* affects neuron *i* if and only if

$$j \in N_i$$

where $N_i \subset \{1,...,n\}$ is a subset of unique indices.

The network architecture is characterized by the density of connections, or *connectivity*:

$$\rho = \sum_{i=1}^{n} |N_i| \Big/ n^2$$

There is no direct connection from a neuron to itself: $i \notin N_i, \forall i$.

The neuron input, or *local field* of the *i*-th neuron, is calculated as a weighted sum of net outputs:

$$h_i = \sum_{j \in N_i} w_{ij} s_j$$

where *W* is the (*n*×*n*) weight matrix of interneuron connections.

During the *convergence process* the neuron output at the next time step is obtained after applying some nonlinear *activation function* to the neuron input at the current time step:

$$s_i^{t+1} = f\left(h_i^t\right) = \mathrm{sgn}\left(h_i^t\right)$$

In the given work as an activation function we use the sign function with codomain {-1, +1}. Therefore the network stores bipolar vectors with {-1, +1} components.

Neuron states can be updated synchronously or asynchronously. We use a synchronous update mode that favours parallel processing in a hardware implementation and offers better associative properties to the network.

2.2 Associative Properties

We want the network to store a set of *n*-ary, bipolar training vectors

$$\left\{\xi^p\right\},\ p = 1...m,\ \xi^p \in \left\{-1,+1\right\}^n$$

The network is said to have *associative properties* with respect to the training set $\{\xi^p\}$ if for some value of *r* the convergence process starting from any point within Hamming distance *r* from one of the training vectors ξ^p always finishes exactly at ξ^p.

The maximum value of *r* satisfying this criterion is called the *Attraction Radius* (AR) of the network.

2.3 Kappa estimation of associative performance

It is always possible to find AR experimentally. But it can be computationally expensive to obtain a reliable value for AR for high dimensional networks. In the situation when one is not interested in quantitative estimation of attractor performance but rather wants to compare different networks, another approach can be used.

The *aligned local field* for the i-th neuron and bipolar vector ξ is a value of $h_i\xi_i$. The bipolar vector is a stable point for the network if and only if all aligned local fields are positive.

An upward scaling of the weight matrix increases all aligned local fields, but obviously does not affect the associative properties of the network. Optimal performance can be achieved with the maximization of the aligned local fields with respect to the size of weights. It can be done maximizing the n*ormalized stability measure* [5]:

$$\gamma_i = h_i\xi_i \Big/ \left\| W^i \right\|$$

where W^i is the vector of weight coefficients of neuron i (the i-th row of W).

The minimum over all neurons and all training patterns gives a measure of attractor performance (κ-*measure*):

$$\kappa = \min_{\substack{i=1\ldots n \\ p=1\ldots m}} \gamma_i^p$$

3 Learning Algorithm

In [4] we studied the influence of an architecture selection algorithm on the associative properties of the network. Given a certain number of inter-neuron connections we wanted to find the best way to allocate them among neurons, i.e. we wanted to find the network architecture that would enable us to obtain the sparse network with the largest values of AR. The best results were obtained with a two-phase algorithm. The first phase is architecture selection and the second one is assigning weights to the connections using the noniterative Pseudo Inverse learning rule (PI LR) [6].

To select network architecture we use an approach based on the Weight Selection algorithm [7]. For a given set of training vectors we construct the projective matrix onto this dataset. Then for a given connectivity ρ of the network we select the location of ρn^2 connections corresponding to the projective matrix elements with the largest absolute values. It should be noted that using PI LR instead of reusing the subset of projective weights provides the improvement of associative properties [4].

For sufficiently connected networks ($\rho \sim m/n$) and non-singular training data $\{\xi^p\}$ PI LR is guaranteed to match its learning criterion – the equality of all aligned local fields to 1. The algorithm works as follows.

To allow for structural restrictions imposed by the cellular architecture we introduce a selection operator that sparsifies the columns of a matrix:

$$S^i : (l \times n) \rightarrow (l \times |N_i|)$$

Operator S^i retains only those columns of its matrix argument that correspond to indices contained in N_i.

Denoting the i-th row of the training data matrix as $\{\xi^p\}^i$, the weights of the i-th neuron are calculated as a solution of the following "fixed point" equation:

$$S^i\left(W^i\right) \cdot S^i\left(\left\{\xi^p\right\}^T\right)^T = \left\{\xi^p\right\}^i \qquad (1)$$

This solution can be found using matrix pseudo inversion operator:

$$S^i\left(W^i\right)=\left\{\xi^p\right\}^i\cdot\left(S^i\left(\left(\left\{\xi^p\right\}^T\right)^T\right)\right)^+$$

4 Numerical Simulations

4.1 Experimental Setup

The training set $\{\xi^p\}$ was composed of random data vectors with independent and equiprobable components {-1, +1}. The size of the data set is 20 and its dimension is 200 (n = 200, m = 20)

In order to estimate the attraction radius r for each value of connectivity there were 100 tests of convergence for training vectors corrupted with a certain level h of noise. The value of h corresponding to AR was found using the bisection method.

4.2 Evolution of associative properties with increasing connectivity

To verify that the architecture selection algorithm is crucial for the emergence of phase transition we measured the evolution of AR with the increase in connectivity for four networks. All these networks were trained using PI LR but they use different rules for architecture selection.

- *PI WS* – The main model described in this paper. It uses the Weight Selection algorithm for setting up the architecture.
- *PI Random* – The model with random architecture (no diagonal connections, the rest of connections are chosen randomly subject to the required connectivity).
- *PI Cell* – The model with cellular (regular lattice) architecture. This architecture corresponds to 1D cellular network with the following connectivity criterion:

$$j\in N_i \quad\Leftrightarrow\quad i\neq j\ \&\ |i-j|\leq \mathit{connection\ radius}$$

- *PI WS Reverse* – This approach, like PI WS, also uses projective matrix for architecture selection. But, so as to consider probably the worst choice of the connectivity pattern, it uses location of elements of projective matrix with the *smallest* absolute values.

Due to the usage of PI LR all of these networks starting from connectivity $\rho \geq 0.16$ have sufficiently complex architectures to find sets of weights providing all aligned local fields being equal to 1. In spite of very much the same behaviour in terms of learning, the networks shows considerably different results when it comes to their performance as associative memory. This allows us to conclude that the difference we see in the associative properties of the networks is determined by the way their architectures are selected.

As Figure 1a shows, phase transition from non-remembering to remembering state is observed only for the PI WS model (at the value of connectivity $\rho = 0.16$). The discontinuous behaviour of κ-measure (Figure 1b) also corresponds to this fact. Only the network trained with the PI WS algorithm exhibits a sharp increase in the positive codomain (0.15; 0.05) → (0.16; 0.5).

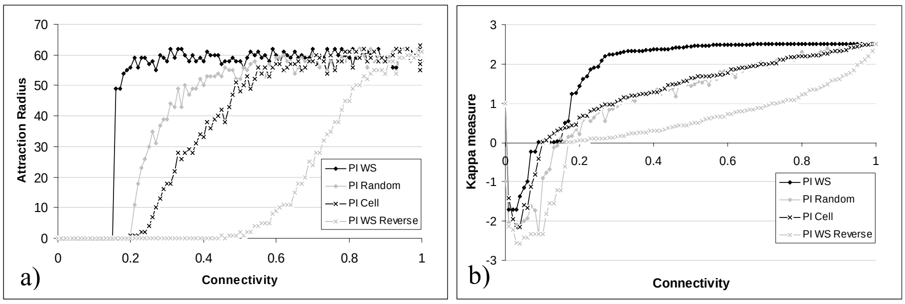

Fig. 1: Attraction Radius and κ-measure vs. network connectivity.

4.3 Dynamics of transition connectivity

In the previous experiment the value of AR observed at the transition state is r_{trans} = 49. This value is fairly close to the AR of fully connected network which is r_{full} = 58. We can evaluate the steepness of transition as a ratio of these two AR values:

$$\eta = r_{trans} \big/ r_{full}$$

In the next experiment we calculate the values of transition connectivity ρ_{trans} and AR for different *n* and *m* in a similar experimental to the preceding. For each set of values of *n* and *m* we averaged ρ_{trans}, r_{trans} and r_{full} over three sets of randomly generated data. We can see from Figure 2 that the value of transition connectivity increases and transition behaviour itself becomes less explicit with decreasing network dimension (*n*) and increasing memory load (*m*).

Two values for (*n*; *m*) equal to (100; 100) and (100; 80) are missing as network exhibits no associative behaviour at these points ($r_{trans} = r_{full} = 0$).

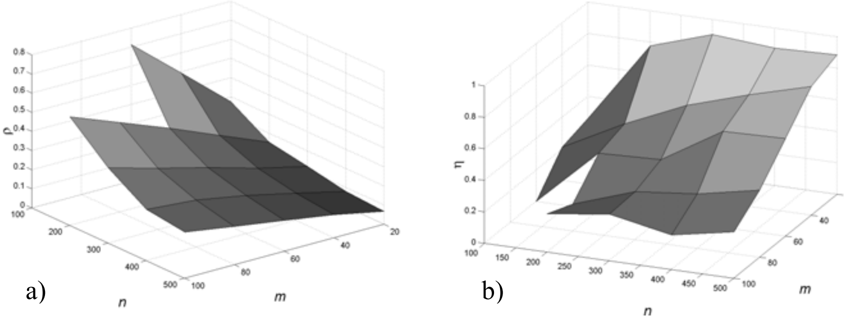

Fig. 2: Transition connectivity and steepness vs. dimension and memory load.

4.4 Estimation of transition connectivity

Normally it requires *m* weights for each neuron to memorize *m* patterns; hence we can expect the transition connectivity to be a function of normalized memory loading *m/n*.

Experimental values of ρ_{trans} obtained in the previous experiment are shown in Figure 3. Within the considered range of parameters the dependence appears to be linear with a high degree of correlation. The fact that it takes on average more than *m* weights per neuron to store the set of *m* training patterns (angular coefficient α = 1.1 > 1) is explained by the unequal number of selected weights among neurons.

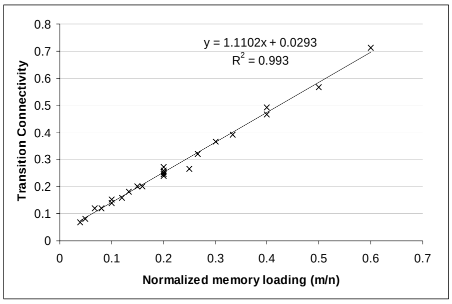

Fig. 3: Transition connectivity and steepness vs. normalized memory loading.

5 Conclusions

Experimental results show the importance of proper network architecture selection for the existence of phase transition. As a matter of fact, running the WS algorithm, we are performing the selection of variables (neuron weights) of equation (1) using the information about the constraints they must satisfy. This approach is more optimal for the solvability of (1), providing a lower-norm solution (higher value of κ-measure). Once the network has sufficient number of connections for the solution to exist, this solution is already good enough to provide high performance as an associative memory, thus causing the observed phase transition phenomenon.

The discovered behaviour is also important from a practical point of view – for a wide range of parameters it allows us to use a sparse network instead of a fully-connected one without sacrificing performance in associativity.

References

[1] A.-L. Barabasi, Dynamics of random networks: connectivity and first order phase transitions, *arxiv.org*, *Condensed Matter*, cond-mat/9511052.

[2] J. Bohland and A. Minai, Efficient associative memory using small-world architecture, *Neurocomputing*, 38:489-496, Elsevier, 2001.

[3] L. Morelli, G. Abramson and M. Kuperman, Associative memory on a small-world neural network, *The European Physical Journal B*, 38(3): 495-500, Springer-Verlag Heidelberg, 2004.

[4] O. Dekhtyarenko, A. Reznik and A. Sitchov, Associative cellular neural networks with adaptive architecture, proceedings of the *8th IEEE International Biannual Workshop on Cellular Neural Networks and their Application* (CNNA 2004), July 22-24, Budapest (Hungary), 2004.

[5] T.B. Kepler and L.F. Abbot, Domains of attraction in neural networks, *Journal de Physique*, 49: 1657-1662, Société Française de Physique, 1988.

[6] M. Brucoli, L. Carnimeo and G. Grassi, Discrete-time cellular neural networks for associative memories with learning and forgetting capabilities, *IEEE Transactions on Circuits and Systems*, 42(7):396-399, IEEE, 1995.

[7] A. Sitchov, Weights selection in neural networks, *Mathematical Machines and Systems* (in Ukrainian), 2:25-30, IMMSP, 1998.