Neurocomputing 000 (2017) 1–8

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Generating exponentially stable states for a Hopfield Neural Network

Erick Cabrera [a,b,*], Humberto Sossa [a]

[a] Instituto Politécnico Nacional (CIC), Av. Juan de Dios Bátiz, S/N, Col. Nva. Industrial Vallejo, Mexico D.F. 07738, Mexico
[b] Universidad Autónoma del Estado de Mexico, Av. Bordo de Xochiaca S/N, Col. Benito Juárez, Estado de Mexico 57000, Mexico

ARTICLE INFO

Article history:
Received 8 November 2016
Revised 19 April 2017
Accepted 28 August 2017
Available online xxx

Communicated by S. Arik

Keywords:
Exponential capacity
Hopfield network
Stable state
Associative memory

ABSTRACT

An algorithm that generates an exponential number of stable states for the very well-known Hopfield Neural Network (HNN) is introduced in this paper. We show that the quantity of stable states depends on the dimension and number of components of the input pattern supporting noise. Extensive tests verify that the states generated by our algorithm are stable states and show the exponential storage capacity of a HNN. This paper opens the possibility of designing improved HNNs able to achieve exponential storage, and thus find their applicability in complex real-world problems.

1. Introduction

Since several decades several brain models have been developed trying to describe short and long term memory nature, based on the fact that memory is the result of synaptic modification that depends on the correlation between presynaptic and postsynaptic firings. Through artificial neural networks, scientists are able to mimic certain behaviors of the human brain. One of the most important topics is related with the question of how the brain acquires and maintains new information, process known as learning and memory. An associative memory has the capacity to recall a stored pattern from a reasonable sub-set of its information. An associative memory is also known to function as an error corrector in the sense that it can eliminate inconsistent information of the signal that it represents. An associative memory is a dynamic information system composed of a set of stable states acting as attracting basins through which neighboring states evolve in time. This temporal evolution of a set of neuronal elements though an equilibrium point can be interpreted as the evolution of an imperfect pattern through a correct (stored) pattern. This way, association and information recall is simulated by the dynamic behavior of a non-linear system. The Hopfield associative memory model is known to exhibit these characteristics.

Hopfield Neural Networks (HNN) are an important class of recurrent neural networks. They have been extensively studied during decades. HNN have been applied in different areas such as problem optimization, pattern classification, signal processing, image segmentation and associative memories [1–6].

Despite huge efforts, the great majority of research papers still perform tests on artificial patterns without an application on a real-world problem [7–10].

The dynamic behavior of different classes of HNN has been a topic of great interest to many researchers having been extensively investigated. The high-order HNNs have attracted the attention of many authors due to the fact that high-order neural networks present stronger properties than low-order neural networks. Stability and dissipativity are two issues that arise from the dynamical behavior of high-order neural networks. In [11] new criteria for ensuring the existence and global exponential stability of almost periodic solution for delayed high-order Hopfield Neural Network were obtained. Duan et al. [12] studied the problem of robust dissipativity for a class of recurrent neural networks with time-varying delay and discontinuous activations showing that the dissipativity is important because it can generalize the idea of a Lyapunov function.

Among the important research topics the following can be mentioned: the problem of stability [13–17], storage capacity [18–23], equilibrium points (stable states) [24]. For example, in [7], a method for the design of a HNN is developed, by which a given set of patterns can be assigned as locally asymptotically network equilibria. Li et al. [25] have shown that a class of neural networks relatively close to the Hopfield model has at most 2^n asymptotically stable equilibrium points. This research shows the possibility

* Corresponding author at: Universidad Autónoma del Estado de Mexico, Av. Bordo de Xochiaca S/N, Col. Benito Juárez, Estado de Mexico 57000, Mexico.
E-mail address: cabrerickx@gmail.com (E. Cabrera).

http://dx.doi.org/10.1016/j.neucom.2017.08.032
0925-2312/

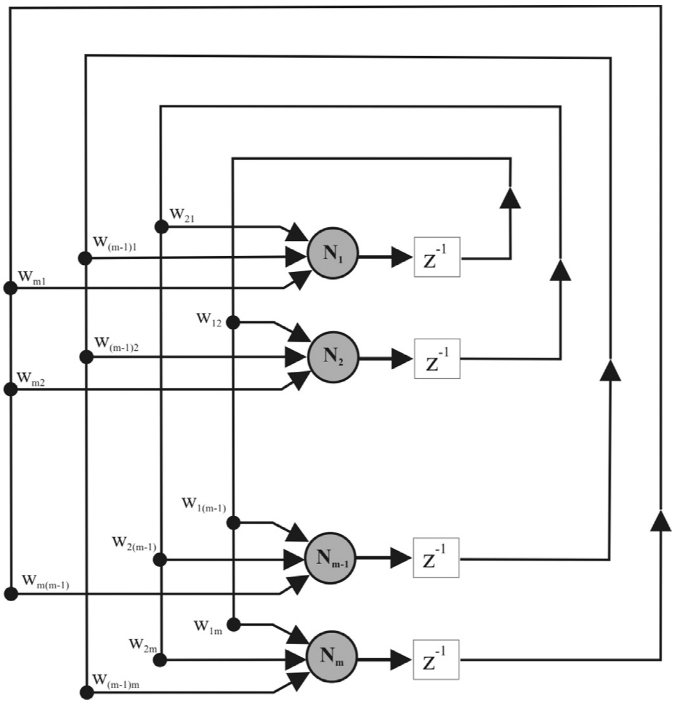

Fig. 1. Typical architecture of the Hopfield Neural Network.

of reaching exponentially storage capacity; although the results were obtained under a set of assumptions.

In practice if we store a set of patterns in a HNN, it is very difficult to ensure if the HNN will converge to stable state that corresponds to the stored pattern, in this sense no article has been published telling the reader which set of points of the state space behave as stable states and which allow maximum retrieval capacity in the presence of noise.

2. Hopfield model: basics, training and retrieval

The discrete version of the HNN consists of n fully connected neurons as depicted in Fig. 1. Each neuron in a HNN has two states determined by the level of the induced local field acting on it. The "on" state of neuron i is denoted as $x_i = +1$, and the "off" state is represented by $x_i = -1$. These values represent the two neuron possible states, a state of inhibition (-1) or a state of activation $(+1)$ as in McCulloch and Pitts neuron [26]. In a HNN the connection strength is described by a $n \times n$ matrix; this matrix is called weight matrix. Each neuron also has a threshold. Each weight matrix defines a HNN in its discrete version. From a structural point of view, the general model of the HNN is fully connected and contains positive and negative feedback information.

For a network made up of such n neurons, a state of the network is defined by the vector:

$$\mathbf{x} = [x_1, x_2, \ldots, x_n] \tag{1}$$

The induced local field v_j of neuron j is defined by

$$v_j = \sum_{i=1}^{n} w_{ji} x_i + b_{j,} \tag{2}$$

where b_j is a fixed bias applied externally to neuron j and w_{ij} represents the synaptic weight from neuron i to neuron j. We refer to W as $n \times n$ synaptic HNN weight matrix. In a normal operation W satisfies two conditions: (1) $w_{ii} = 0$ for all i, and (2) $W^T = W$.

Neuron j modifies its state x_j, according to the following rule:

$$x_j = \begin{cases} -1 & \text{if } v_j < 0 \\ +1 & \text{if } v_j > 0 \end{cases} \tag{3}$$

Conveniently, if $v_j = 0$, then neuron j must remain in its previous state, regardless whether it is on or off. Equivalently:

$$x_j = sgn(v_j), \tag{4}$$

where sgn is the signum function [27].

A discrete HNN operates in two phases: one of storage and one of retrieval, both phases are briefly described next.

2.1. Storage phase

During this phase we are given a set of M patterns to be memorized in a HNN. Each pattern is represented by an n-dimensional vector. These vectors called fundamental memories are denoted by $\{\mathbf{x}_a : a = 1, 2, \ldots, M\}$. Let $x_{a,\,i}$ denote the ith component of the fundamental memory $\mathbf{x}_a$. According to generalized Hebb's learning rule, the synaptic weight from neuron i to neuron j is defined by

$$w_{ji} = \frac{1}{n} \sum_{a=1}^{M} x_{a,j} x_{a,i} \tag{5}$$

Also, we can write synaptic weight matrix $W = w_{ij}$ in matrix form as follows:

$$W = \frac{1}{n} \sum_{a=1}^{M} \mathbf{x}_a \mathbf{x}_a^T - M\mathbf{I} \tag{6}$$

In this case $\mathbf{x}_a \mathbf{x}_a^T$ represents the outer product of the vector $\mathbf{x}_a$ with itself and $\mathbf{I}$ is the identity matrix.

As stated in [27]: (1) The output of each neuron in the network is fed back to all other neurons, (2) there is no self-feedback in the network (i.e., $w_{ii} = 0$), and (3) the weight matrix of the network is symmetric (i.e., $\mathbf{W} = \mathbf{W}^T$).

2.2. Retrieval phase

During this phase, a n-dimensional vector x_{probe}, called probe, with elements equal to $+1$ are imposed to the HNN. Typically, this input pattern is a noisy version of one of the M fundamental memories: $\{\mathbf{x}_a : a = 1, 2, \ldots, M\}$. Pattern retrieval proceeds in accordance with a dynamic rule, applied over each network neuron at some fixed rate. This dynamic rule has two stages as follows:

1. *Initialization.* At $t = 0$, make $\mathbf{x}(t) = \mathbf{x}_{probe}$.
2. *Iteration until convergence.* Update each component of the state vector according to

$$x_j(t+1) = \begin{cases} -1 & \text{if } \sum_{i=1}^{n} w_{ji} x_i(t) < 0 \\ x_j(t) & \text{if } \sum_{i=1}^{n} w_{ji} x_i(t) = 0 \\ +1 & \text{if } \sum_{i=1}^{n} w_{ji} x_i(t) > 0 \end{cases}$$

   Repeat this process until the state vector $\mathbf{x}$ remains unchanged, that is until $x_j(t+1) = x_j(t)$ for all $j = 1, 2, \ldots, n$.
3. *Output state.* At the end, finally the network produces a time-invariant state vector $\mathbf{y}$. This vector $\mathbf{y}$ that satisfies the stability condition is called a stable state.

It is possible that during retrieval phase the HNN converges to stable state that does not correspond to the stored pattern. This state is called spurious state.

Please cite this article as: E. Cabrera, H. Sossa, Generating exponentially stable states for a Hopfield Neural Network, Neurocomputing (2017), http://dx.doi.org/10.1016/j.neucom.2017.08.032