DESIGN AND STABILITY OF HOPFIELD ASSOCIATIVE MEMORY

A THESIS
SUBMITTED TO THE DEPARTMENT OF ELECTRICAL
AND ELECTRONICS ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCES
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
M. Erkan SAVRAN
September 1991

DESIGN AND STABILITY OF HOPFIELD ASSOCIATIVE MEMORY

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

M. Erkan Savran

September 1991

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Ömer Morgül(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Mehmet Ali Tan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Kemal Oflazer

Approved for the Institute of Engineering and Sciences:

Prof. Dr. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

DESIGN AND STABILITY OF HOPFIELD ASSOCIATIVE MEMORY

M. Erkan Savran
M.S. in Electrical and Electronics Engineering
Supervisor: Assist. Prof. Dr. Ömer Morgül
September 1991

This thesis is concerned with the selection of connection weights of Hopfield neural network model so that the network functions as a content addressable memory (CAM). We deal with both the discrete and the continuous-time versions of the model using hard-limiter and sigmoid type nonlinearities in the neuron outputs. The analysis can be employed if any other invertible nonlinearity is used. The general characterization of connection weights for fixed-point programming and a condition for asymptotic stability of these fixed points are presented. The general form of connection weights is then inserted in the condition to obtain a design rule. The characterization procedure is also employed for discrete-time cellular neural networks.

Keywords: Hopfield neural network, content addressable memory, fixed-point programming, cellular neural networks.

ÖZET

HOPFIELD ÇAĞRIŞIMSAL BELLEK TASARIMI VE KARARLILIĞI

M. Erkan Savran
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Yard. Doç. Dr. Ömer Morgül
Eylül 1991

Bu tez Hopfield sinirsel ağ modelinin içerik adreslenebilir bellek olarak çalışabilmesi için bağlantı ağırlıklarının seçimi ile ilgilidir. Sinir çıktılarında sigmoid ve signum türü fonksiyonlar kullanılarak, modelin eşzamanlı yenileme kuralı ile çalışan ayrık ve sürekli zaman halleri incelenmiştir. Analiz herhangi tersi olan bir fonksiyon kullanıldığında da uygulanabilir. Sabit nokta programlaması için bağlantı ağırlıklarının genel yapısı ve bir asimtotik kararlılık şartı verilmiştir. Genel yapı daha sonra bu şarta konarak bir tasarım kuralı elde edilmiştir. Genel yapı prosedürü ayrık zamanlı hücresel sinir ağlarında da kullanılmıştır.

Anahtar Sözcükler: Hopfield sinirsel ağı, içerik adreslenebilir bellek, sabit nokta programlaması, hücresel sinir ağları.

To my family,

ACKNOWLEDGMENT

I am grateful to Assist. Prof. Ömer Morgül, for his supervision, guidance, encouragement and patience during the development of this thesis. I am indebted to the members of the thesis committee: Assoc. Prof. Mehmet Ali Tan and Assist. Prof. Kemal Oflazer.

I want to thank to my family for their constant support and to all my friends, especially B. Fırat Kılıç, Atilla Malaş and Taner Oğuzer for their many valuable discussions.

Contents

List of Figures

Chapter 1

Introduction

In recent years, the neural network model proposed by Hopfield has attracted a great deal of interest among researchers from various fields. This is due to a number of attractive features of these networks such as collective computation capabilities, massively parallel processing, etc., and these properties could be used in areas like control & robotics, pattern recognition and content addressable memory (CAM) design.

The Hopfield model consists of neurons, which are multi-input, single-output, nonlinear processing units, and a large number of interconnections between them. The model has a feedback structure so that each neuron can have information about the outputs of the other neurons. It is this high degree of connectivity that makes the neural networks computationally attractive. Hopfield has showed that with a proper choice of connection weights, the network can perform well as a CAM or can be used in solving difficult optimization problems such as traveling salesman problem (TSP) [1], [3].

Many researchers have argued certain aspects of Hopfield neural network and proposed methods of adjusting the connection weights for specific tasks. For the associative memory design, still most of these methods suffer from significant drawbacks, such as the existence of a large number of spurious memory vectors, or some conditions that should be posed on memory vectors so that they can be stored by those methods. In [2], Hopfield has used outer product rule, which is one of the most widely used methods, to store a given set of memory vectors. In [12] and [15], memory vectors have been chosen linearly independent and successfully stored. McEliece in [14], has proposed memory vectors to be eigenvectors of connection matrix with positive eigenvalues and Michel in [5], has synthesized the connection matrix so that the memory vectors become the eigenvectors of that matrix with a single degenerate positive eigenvalue (see also [8]). In [9], [10] and [11], Bruck and Goles have investigated the convergence properties of the network depending on the topology and the mode of operation (either serial or parallel). The work done by these people will be explained in detail in Chapter 3.

The subject of this thesis is to make a general characterization of all possible connection weights so that the network performs as an associative memory. A stability analysis is also done for a large class of Hopfield neural networks that contain functions having continuous first derivatives in their neuron outputs. The characterization process is then carried out to cellular neural networks proposed by Chua and Yang [20], which have important applications in areas such as image processing and pattern recognition. In cellular neural networks memory patterns are matrices rather than vectors and the topology is different since a certain node can communicate with a predefined neighborhood only.

In this study, we consider both discrete and continuous-time Hopfield neural networks with synchronous update rule and discrete-time cellular neural networks. We give a general characterization of all possible connection weights which store a given set of patterns into the neural net (i.e. each pattern becomes a fixed point of the network). We use several class of functions in the neurons and investigate the stability properties of the memory vectors for which the functions used in the neurons have continuous first derivatives. The analysis is supported by examples and some widely used methods are also discussed.

The thesis is organized as follows. In Chapter 2, we present the mathematical tools employed in the study such as singular value decomposition and matrix norms. In Chapter 3, we summarize the past work on Hopfield model and present the material which have been most important for us. In Chapter 4, we define our design problem (together with the motivation for it) for the discrete-time case first and give solutions to the problem for the cases depending on the nonlinearity used in the neuron outputs. We then give some results about the stability of the equilibria when sigmoid type nonlinearity is used. We redefine the design problem for the continuous-time case, give solutions to the problem depending on the nonlinearity and provide results concerning the stability of the equilibria. Finally, the design process is used for cellular neural networks. In Chapter 5, we give concluding remarks.

Chapter 2

Mathematical Preliminaries

2.1 Singular Value Decomposition

This chapter will serve as an auxiliary one in which we are going to make definitions and present some mathematical tools that will be used in the subsequent chapters.

Let $\mathbf{A} \in \Re^{m\times n}$, since $\mathbf{A}^T\mathbf{A}$ is a symmetric positive semidefinite matrix, it has n non-negative eigenvalues, $\sigma_1^2, \ldots, \sigma_n^2$, with n corresponding orthonormal eigenvectors, $\mathbf{v}_1, \ldots, \mathbf{v}_n$. Assume that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$; $\sigma_j = 0,\ j = r+1, \ldots, n$. Obviously $r = rank(\mathbf{A})$ and we have

$$\mathbf{A}^T\mathbf{A}\mathbf{v}_i = \sigma_i^2\mathbf{v}_i\,,\ \ i = 1, \ldots, n \tag{2.1}$$

Note that, $\mathbf{v}_{r+1}, \ldots, \mathbf{v}_n$ belong to the nullspace of $\mathbf{A}$, they are orthonormal and arbitrary otherwise. We define the vectors $\mathbf{u}_i$ as follows:

$$\mathbf{A}\mathbf{v}_i = \sigma_i\mathbf{u}_i\,,\ \ i = 1, \ldots, r \tag{2.2}$$

It is easy to see that the vectors $\mathbf{u}_i$ are orthonormal:

$$\mathbf{u}_i^T\mathbf{u}_j = \frac{1}{\sigma_i}\frac{1}{\sigma_j}(\mathbf{A}\mathbf{v}_i)^T\mathbf{A}\mathbf{v}_j = \frac{\sigma_j}{\sigma_i}\mathbf{v}_i^T\mathbf{v}_j\,,\ i, j = 1, \ldots, r \tag{2.3}$$

Let $\mathbf{u}_i, i = r+1, \ldots, m$, denote an arbitrary orthonormal complement of the vectors $\mathbf{u}_i, i = 1, \ldots, r$, defined by (2.3). Define $\mathbf{U} = [\mathbf{u}_1 \ldots \mathbf{u}_m], \boldsymbol{\Sigma} = \begin{pmatrix} \mathbf{D} & 0 \\ 0 & 0 \end{pmatrix}$, with $\mathbf{D} = diag(\sigma_1, \ldots, \sigma_r)$, and $\mathbf{V} = [\mathbf{v}_1 \ldots \mathbf{v}_n]$. Then, it is easy to see that we have the following decomposition

$$\mathbf{A} = \mathbf{U\Sigma V}^T \tag{2.4}$$

To summarize, we have the following theorem; square brackets denote the complex case [23].

Theorem 1: Let $\mathbf{A} \in \Re^{m\times n}$ $[C^{m\times n}]$. Then there exist orthogonal [unitary] matrices $\mathbf{U} \in \Re^{m\times m}$ $[C^{m\times m}]$ and $\mathbf{V} \in \Re^{n\times n}$ $[C^{n\times n}]$ such that

$$\mathbf{A} = \mathbf{U\Sigma V}^T \; [\mathbf{U\Sigma V}^H] \tag{2.5}$$

where $\mathbf{V}^T$ and $\mathbf{V}^H$ denote the transpose and conjugate transpose of $\mathbf{V}$, respectively, and

$$\Sigma = \begin{pmatrix} \mathbf{D} & 0 \\ & \\ 0 & 0 \end{pmatrix}$$

$\mathbf{D} = diag(\sigma_1, \ldots, \sigma_r)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, where $r = rank(\mathbf{A})$.

Proof: See [23].

The numbers $\sigma_1, \ldots, \sigma_r$ together with $\sigma_{r+1} = 0, \ldots, \sigma_n = 0$ are called the singular values of $\mathbf{A}$ and they are the positive square roots of the eigenvalues (which are non-negative) of $\mathbf{A}^T\mathbf{A}$ $[\mathbf{A}^H\mathbf{A}]$. The columns of $\mathbf{U}$ are called the left singular vectors of $\mathbf{A}$ (the orthonormal eigenvectors of $\mathbf{AA}^T$ $[\mathbf{AA}^H]$) while the columns of $\mathbf{V}$ are called the right singular vectors of $\mathbf{A}$ (the orthonormal eigenvectors of $\mathbf{A}^T\mathbf{A}$ $[\mathbf{A}^H\mathbf{A}]$). The matrix $\mathbf{A}^T$ $[\mathbf{A}^H]$ has m singular values, the positive square roots of the eigenvalues of $\mathbf{AA}^T$ $[\mathbf{AA}^H]$. The r (=rank ($\mathbf{A}$)) nonzero singular values of $\mathbf{A}$ and $\mathbf{A}^T$ $[\mathbf{A}^H]$ are, of course, the same. The choice of $\mathbf{A}^T\mathbf{A}$ $[\mathbf{A}^H\mathbf{A}]$ rather than $\mathbf{AA}^T$ $[\mathbf{AA}^H]$ in the definition of singular values is arbitrary.

It is not generally a good idea to compute the singular values of $\mathbf{A}$ by first finding the eigenvalues of $\mathbf{A}^T\mathbf{A}$. As in the case of linear least squares problems, the computation of $\mathbf{A}^T\mathbf{A}$ involves unnecessary numerical inaccuracy. For example, let

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ \beta & 0 \\ 0 & \beta \end{pmatrix}$$

then

$$\mathbf{A}^T\mathbf{A} = \begin{pmatrix} 1+\beta^2 & 1 \\ 1 & 1+\beta^2 \end{pmatrix}$$

so that

$$\sigma_1(\mathbf{A}) = (2+\beta^2)^{\frac{1}{2}},\ \sigma_2(\mathbf{A}) = |\beta|.$$

If $\beta^2 < \varepsilon_0$, the machine precision, the computed $\mathbf{A}^T\mathbf{A}$ has the form $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$, and one obtains from diagonalization $\tilde{\sigma_1}(\mathbf{A}) = \sqrt{2}$, $\tilde{\sigma_2}(\mathbf{A}) = 0$.

Fortunately, Golub and Reinsch [22] have developed an extremely efficient and stable algorithm for computing the SVD which does not suffer from the above defect. Golub and Reinsch employs Householder transformations to reduce $\mathbf{A}$ to bidiagonal form, and then the QR algorithm to find the singular values of the bidiagonal matrix. These two phases properly combined produce the singular value decomposition of $\mathbf{A}$.

By using Householder transformations, the matrix $\mathbf{A}$ is first transformed into a matrix $\mathbf{J}^{(0)}$ having the same singular values as $\mathbf{A}$, i.e. if the SVD of $\mathbf{A}$ is $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$, that of $\mathbf{J}^{(0)}$ is $\mathbf{J}^{(0)} = \mathbf{G}\Sigma\mathbf{H}^T$. $\mathbf{J}^{(0)}$ matrix, which is in bidiagonal form, is then iteratively diagonalized using a variant of QR algorithm so that

$$\mathbf{J}^{(0)} \rightarrow \mathbf{J}^{(1)} \rightarrow \ldots \rightarrow \Sigma.$$

For details, see [22].

2.2 Induced Matrix Norms and Jacobian of a Function

Definition: Let $||.||$ be a given norm on C^n. Then for each matrix $\mathbf{A} \in C^{n\times n}$, the quantity $||\mathbf{A}||_i$ defined by

$$||\mathbf{A}||_i = \sup_{\mathbf{x}\neq 0, \mathbf{x}\in C^{n\times n}} \frac{||\mathbf{A}\mathbf{x}||}{||\mathbf{x}||} = \sup_{||\mathbf{x}||=1} ||\mathbf{A}\mathbf{x}|| = \sup_{||\mathbf{x}||\leq 1} ||\mathbf{A}\mathbf{x}|| \tag{2.6}$$

is called the induced (matrix) norm of $\mathbf{A}$ corresponding to the vector norm $||.||$ [17].

It should be noted that there are two distinct functions involved in definition (2.6): One is the norm function $\|.\|$ mapping C^n into $\Re$, and the other is the norm function $\|.\|_i$ mapping $C^{n\times n}$ into $\Re$.

The induced norm of a matrix $\mathbf{A}$ (or the induced norm of a linear mapping $\mathbf{A}$) can be given a simple geometric interpretation. Equation (2.6) shows that $\|\mathbf{A}\|_i$ is the least upper bound of the ratio $\|\mathbf{Ax}\|/\|\mathbf{x}\|$ as $\mathbf{x}$ varies over C^n. In this sense, $\|\mathbf{A}\|_i$ can be thought of as the maximum "gain" of the mapping $\mathbf{A}$.

The induced norms have the special feature that they are submultiplicative; i.e.

$$\|\mathbf{AB}\|_i \leq \|\mathbf{A}\|_i\|\mathbf{B}\|_i \ \forall \mathbf{A}, \mathbf{B} \in C^{n\times n} \tag{2.7}$$

Another useful identity is

$$\|\mathbf{A}+\mathbf{B}\|_i \leq \|\mathbf{A}\|_i + \|\mathbf{B}\|_i \ \forall \mathbf{A}, \mathbf{B} \in C^{n\times n} \tag{2.8}$$

The induced matrix norms corresponding to the vector norms $\|.\|_\infty$, $\|.\|_1$, and $\|.\|_2$ respectively, are known and are displayed below.

$$i)\ \|\mathbf{x}\|_\infty = \max_i |\mathbf{x}_i|,\ \ \|\mathbf{A}\|_{i\infty} = \max_i \sum_j |a_{ij}|\ \ (max.rowsum)$$

$$ii)\ \|\mathbf{x}\|_1 = \sum_i |x_i|,\ \ \|\mathbf{A}\|_{i1} = \max_j \sum_i |a_{ij}|\ \ (max.columnsum)$$

$$iii)\ \|\mathbf{x}\|_2 = (\sum_i |x_i|^2)^{\frac{1}{2}},\ \ \|\mathbf{A}\|_{i2} = [\lambda_{max}(\mathbf{A}^H\mathbf{A})]^{\frac{1}{2}} = \sigma_{max}(\mathbf{A})$$

where $\lambda_{max}(\mathbf{A}^H\mathbf{A})$ =maximum eigenvalue of $\mathbf{A}^H\mathbf{A}$. Similarly $\|\mathbf{A}^{-1}\|_{i2} = 1/\sigma_{min}(\mathbf{A})$.

Lastly, we define the Jacobian of a vector function. Let $\mathbf{f}(\mathbf{x}) : \Re^n \rightarrow \Re^m$ be a differentiable vector function.

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \mathbf{f}(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_m(\mathbf{x}) \end{pmatrix}$$

The Jacobian $\partial \mathrm{f}/\partial \mathrm{x} : \Re^n \rightarrow \Re^{m \times n}$ is defined as:

$$\frac{\partial \mathrm{f}}{\partial \mathrm{x}} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$

Chapter 3

Past Work on Hopfield Model

In 1982, Hopfield proposed a system, based on biological nervous systems, which performs as an associative memory. He has called the processing devices "neurons" and each neuron i had two states: either $V_i = 0$ ("not firing") or $V_i = 1$ ("firing at maximum rate").

In Hopfield model, each neuron i has a connection made to it from neuron j, the strength of connection is defined as T_{ij}. The instantaneous state of the system is specified by listing the N values of V_i, so it is represented by a binary word of N bits. In general, the next state of the network is computed from the current states at a subset of the nodes of the network, to be denoted by S [9]. The modes of operation are determined by the method by which the set S is selected in each time interval. If the computation is performed at a single node in any time interval that is, $|S| = 1$ ($|S|$ denoting the number of nodes in the set S) then the network is operating in fully asynchronous mode or serial mode and if the computation is performed in all nodes in the same that is, $|S| = N$ (N is the total number of nodes in the network) , then the network is said to be operating in synchronous mode or fully parallel mode. All the other cases, that is, $1 < |S| < N$, will be called asynchronous or parallel modes of operation. The set S can be chosen randomly or according to some deterministic rule.

In Hopfield model, each neuron i readjusts its state randomly in time, setting

$$V_i \to 1 \text{ if } \sum_{j \neq i} T_{ij} V_j > 0$$

$$V_i \rightarrow 0 \text{ if } \sum_{j \neq i} T_{ij} V_j < 0$$

Thus each neuron randomly and asynchronously evaluates whether it is above or below 0 and readjusts accordingly.

Associative memory problem is to store a given set of states to the network, i.e. states should be fixed points of the network. For the associative memory problem, Hopfield has suggested the rule to store the set of states V^s, $s = 1, \ldots, n$:

$$T_{ij} = \sum_s (2V_i^s - 1)(2V_j^s - 1) \tag{3.1}$$

but with $T_{ii} = 0$. This model has stable limit points. Hopfield considers the special case $T_{ij} = T_{ji}$, and defines

$$E = -\frac{1}{2} \sum \sum_{i \neq j} T_{ij} V_i V_j \tag{3.2}$$

ΔE due to ΔV_i is given by

$$\Delta E = -\Delta V_i \sum_{j \neq i} T_{ij} V_j \tag{3.3}$$

Thus, the algorithm for altering V_i causes E to be monotonically decreasing function. State changes will continue until a minimum (local) E is reached.

Apart from the discrete stochastic model above, Hopfield has also proposed a continuous deterministic model for the associative memory problem based on continuous variables yet keeping all the significant behaviour of the original discrete model. He has constructed an electronic model using capacitors, resistors and amplifiers each of which represented a certain element of the nerve system.

The equation governing the electronic circuit is nonlinear and is given as follows

$$C_i(du_i/dt) = \sum_j T_{ij} g_i(u_j) - u_i/R_i + I_i \tag{3.4}$$

where u_i is the input to nonlinear amplifier, $g_i(u_j)$ represents the input-output characteristics of the nonlinear amplifier, C_i, R_i represent the capacitor and resistor respectively, and I_i is any other fixed input current.

Hopfield again employs a Lyapunov (or energy) function approach to investigate the stability properties of the network. He chooses the Lyapunov function as

$$E = -\frac{1}{2}\sum_i \sum_j T_{ij} V_i V_j + \sum_i (1/R_i) \int_0^{V_i} g_i^{-1}(V) dV + \sum_i I_i V_i \tag{3.5}$$

Its time derivative for a symmetric $\mathbf{T}$ is

$$dE/dt = -\sum_i dV_i/dt(\sum_j T_{ij} V_j - u_i/R_i + I_i) \tag{3.6}$$

The parenthesis is the right-hand side of (3.4), so

$$dE/dt = -\sum_i C_i(d(g_i^{-1}(V_i))/dV_i)(dV_i/dt)^2 \tag{3.7}$$

Since $g_i^{-1}(V_i)$ is a monotone increasing function and C_i is positive, each term in this sum is nonnegative. Therefore $dE/dt \leq 0$, and by LaSalle's theorem

$$dE/dt = 0 \rightarrow dV_i/dt = 0 \text{ for all i.}$$

Together with the boundedness of E, equation above shows that the time evolution of the system is a motion in state-space that seeks out minima in E and comes to a stop at such points. However, there is no guarantee that the network will find the best minimum.

In [1], Hopfield has also investigated the relation between the stable states of his two models. He has concluded that the only stable points of the extremely high gain continuous deterministic system corresponds to the stable points of the stochastic system.

Many scientists have made research on different aspects and applications of Hopfield model such as content addressable memory (CAM), some optimization problems, and capacity of the model. Here, a few of them will be explained but according to me, they are the most suitable ones to mention here.

A. Michel in his several papers has analyzed a certain class of nonlinear, autonomous, differential and difference equations containing the models of Hopfield. He deals with differential equations of the form (see [6])

$$\frac{d\mathrm{x}}{dt} = \mathbf{F}(\mathrm{x}) = \mathbf{A}\mathrm{x} + \mathbf{T}\mathrm{S}(\mathrm{x}) + \mathbf{I} \tag{3.8}$$

where $\mathbf{x} = (x_1, \ldots, x_n)^T \in \Re^n$; $\mathbf{F} : \Re^n \to \Re^n$ is a measurable function, $\mathbf{A} = diag(-\rho_1, \ldots, -\rho_n)$ is an $n \times n$ constant matrix with $\rho_i > 0$, $\mathbf{T} = [T_{ij}]$ is an $n \times n$ constant matrix, $\mathbf{I} = (I_1, \ldots, I_n)^T$ is a constant vector and $\mathbf{S} : \Re^n \to \Re^n$ is a measurable function defined by $\mathrm{S}(\mathrm{x}) = (g(x_1), \ldots, g(x_n))^T$, $g(x)$ being a monotonically nondecreasing, continuously differentiable function with $g(0) = 0$ and $|g(x)| \leq 1$. His difference equations are in the form of (see [7])

$$\mathrm{u}(k+1) = \mathbf{T}\mathbf{g}(\mathrm{u}(k)) + \mathbf{A}\mathrm{u}(k) + \mathbf{I} \tag{3.9}$$

where $\mathbf{T} = [T_{ij}] \in \Re^{n\times n}$, $\mathbf{A} = diag(A_{ii}) \in \Re^{n\times n}$, $\mathbf{I} = [I_i] \in \Re^n$, and $\mathrm{g} = [g_1, \ldots, g_n]^T : \Re^n \to \Re^n$, g_i's being monotonically nondecreasing, continuously differentiable function with $g_i(0) = 0$ and $|g_i(x)| \leq 1$.

For both of the differential and difference equations, some conditions are imposed on $\mathbf{T}$ and $\mathbf{A}$ matrices so that an appropriate analysis can be done. In the analysis, Michel develops the solutions of the systems, makes the concept of equilibrium point precise, gives bounds on the number of asymptotically stable equilibrium points and discusses the distribution of the equilibrium points in the state-space. He also constructs an energy function to investigate the stability properties of the equilibrium points.

For the synthesis procedure, he defines the set B^n as $B^n = \{\mathrm{x} \in \Re^n : -1 \leq x_i \leq 1,\ i = 1, \ldots, n\}$. Given m vectors in B^n, say $\alpha_1, \ldots, \alpha_\mathrm{m}$, it is desired to design $\mathbf{T}$, $\mathbf{I}$ (for the system (3.8) or (3.9)) such that

1) $\alpha_1, \ldots, \alpha_\mathrm{m}$ are stable output vectors of the system.

2) The system has no periodic output sequences.

3) The total number of stable output vectors of the system in the set $B^n - \{\alpha_1, \ldots, \alpha_\mathrm{m}\}$ is as small as possible.

4) The domain of attraction of each α_i is as large as possible.

Here as an example, we give Michel's synthesis procedure, which fulfills the design objectives stated above, for the continuous-time system

$$\frac{d\mathbf{x}}{dt} = -\mathbf{x} + \mathbf{T}\mathbf{S}(\mathbf{x}) + \mathbf{I} \tag{3.10}$$

where $\mathbf{S}$ is the function defined in (3.8). Suppose we are given m vectors $\alpha_1, \ldots, \alpha_m$ in B^n which are to be stored as stable otuput vectors for (3.10). We proceed as follows:

1) Compute the $n \times (m-1)$ matrix

$$\mathbf{Y} = [\alpha_1 - \alpha_m, \ldots, \alpha_{m-1} - \alpha_m]$$

2) Perform a singular value decomposition of $\mathbf{Y}$ and obtain the matrices $\mathbf{U}$, $\mathbf{V}$ and Σ such that $\mathbf{Y} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and where Σ is a diagonal matrix with the singular values of $\mathbf{Y}$ on its diagonal (this can be accomplished by standard computer routines). Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_{m-1}]$, $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n]$ and k =dimension of Span$(\mathbf{y}_1, \ldots, \mathbf{y}_{m-1})$. From the properties of singular value decomposition, we know that k =rank of Σ, $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ is an orthonormal basis of Span$(\mathbf{y}_1, \ldots, \mathbf{y}_{m-1})$ and $\{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ is an orthonormal basis of $\Re^n$.

3) Compute

$$\mathbf{T}^+ = [T^+_{ij}] = \sum_{i=1}^{k} \mathbf{u}_i \mathbf{u}_i^T, \quad \mathbf{T}^- = [T^-_{ij}] = \sum_{i=k+1}^{n} \mathbf{u}_i \mathbf{u}_i^T$$

4) Choose a value for the parameter $\tau > -1$ and compute

$$\mathbf{T}_\tau = \mathbf{T}^+ - \tau\mathbf{T}^-, \quad \mathbf{I}_\tau = \alpha_m - \mathbf{T}_\tau \alpha_m$$

In particular, the optimal pair $\{\mathbf{T}_{op}, \mathbf{I}_{op}\}$ is given by

$$\mathbf{T}_{op} = \mathbf{T}^+ - \tau_{op}\mathbf{T}^-, \quad \mathbf{I}_{op} = \alpha_m - \mathbf{T}_{op}\alpha_m$$

where $\tau_{op} = min\{T^+_{ii}/T^-_{ii}, 1 \le i \le n\}$.

Michel's work is essentially on analysis and design of Hopfield type systems. Bruck and Goles deal with rather a different aspect of Hopfield model using energy (Lyapunov) functions and a graph theoretic approach. They show that the known convergence properties of the Hopfield model can be reduced to a very simple case and the fully parallel mode of operation is a special case of the serial mode of operation. We begin presenting their work by some definitions and notation.

The order of the Hopfield network is the number of nodes in the corresponding weighted graph (by a weighted graph, we mean a set of nodes together with a set of weighted branches with the condition that each branch terminates at each end into a node). Let N be a neural network of order n; then N is uniquely defined by $(\mathbf{T}, \mathbf{i}^b)$ where:

•$\mathbf{T}$ is an $n \times n$ matrix, with element t_{ij} equal to the weight attached to edge (i, j).

•$\mathbf{i}^b$ is a vector of dimension n, where element i_i^b denotes the threshold attached to node i.

Every node (neuron) can be in one of two possible states, either 1 or -1. The state of node i at time t is denoted by $v_i(t)$. The state of the neural network at time t is the vector $\mathbf{V}(t) = (v_1(t), \ldots, v_n(t))$. The state of a node at time $(t+1)$ is computed by

$$v_i(t+1) = Sign(H_i(t)) = Sign(\sum_{j=1}^{n} t_{ji} v_j(t) - i_i^b) = \begin{cases} 1 & \text{if } H_i(t) \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

A set of distinct states $\{\mathbf{V}_1, \ldots, \mathbf{V}_k\}$ is a cycle of length k if a sequence of evaluations results in the sequence of states: $\mathbf{V}_1, \ldots, \mathbf{V}_k, \mathbf{V}_1, \ldots$ repeating forever.

The main contribution of Bruck and Goles [9] is the following theorem.

Theorem 2: Let $N = (\mathbf{T}, \mathbf{i}^b)$ be a neural network. Then the following hold:

i) If N is operating in a serial mode and $\mathbf{T}$ is a symmetric matrix with zero diagonal, then the network will always converge to a stable state.

ii) If N is operating in a serial mode and $\mathbf{T}$ is a symmetric matrix with nonnegative elements on the diagonal, then the network will always converge to a stable state.

iii) If N is operating in a fully parallel mode then, for an arbitrary symmetric matrix $\mathbf{T}$, the network will always converge to a stable state or a cycle of length 2; that is, the cycles in the state-space are length ≤ 2.

iv) If N is operating in a fully parallel mode then, for an antisymmetric matrix $\mathbf{T}$ with zero diagonal, with $\mathbf{i}^b = 0$, the network will always converge to a cycle of length 4.

Other subjects about content addressability and Hopfield model that are studied by Abu-Mostafa [13], McEliece [14], Venkatesh [12] and Dembo [15] are the capacity of Hopfield model and alternative construction schemes of connection weight matrices for content addressability.

Venkatesh and Dembo [15] have chosen the memory vectors linearly independent in order to store them successfully by their construction scheme. McEliece [14] has proposed memory vectors be eigenvectors of connection weight matrix with positive eigenvalues. These schemes guarantee the storage of memory vectors as fixed points.

Although different definitions exist for the capacity of a neural network, generally we may define the capacity as the maximum number K such that any K vectors of N entries can be made stable in a network of N neurons by the proper choice of connection weight matrix and the threshold vector attached to the neurons.

For the above definition, Abu-Mostafa [13] has shown that the capacity of Hopfield model (using hard-limiter functions in the neurons) is n for an n neuron network.

Using an alternative approach to capacity, McEliece [14] has shown that the capacity of a Hopfield neural associative memory of n neurons with sum-of-outer product interconnections is $m = n/2 \log n$ if we are willing to give up being able to remember a small fraction of the m memories, and $m = n/4 \log n$ if all memories must be remembered.

Chapter 4

Associative Memory Design for a Class of Neural Nets

4.1 Content Addressable Memory and Hopfield Model

In this section, we give brief information about content addressable memories and Hopfield neural networks.

4.1.1 Content Addressable Memory (CAM)

Human brain has an associative property that when a reasonable partial information of a vast set of incidents is supplied, the huge set of information is automatically invoked or synthesized. It has been observed that highly interconnected neural networks have this collective property known as CAM in literature [19]. This property arises from the fact that the motion of a system has stable points in state-space which can be thought of as a kind of memory. From an initial state which contains partial information about a particular memory (i.e. a stable state), the system asymptotically converges to that memory. The memory is reached by knowing its partial content rather than its address, hence called CAM.

Convergent flow to stored memory vectors (i.e. stable states) is the essential feature of this CAM operation. For normal operation of CAM, the memory items should have reasonable regions of attraction, so beginning from any initial point within the region of attraction of a particular item, we should be able to reach that item. Therefore, while designing a content addressable memory, one should take into account the asymptotic stabilities of memory items. The items which are much alike, like the photographs of twins, may cause problems in the separation of regions of attraction, i.e. their stable storage in the memory. The existence of spurious items is another fact that should be eliminated as much as possible. Sometimes, the CAM operation converges to a memory which does not belong to the set of memory patterns stored beforehand. These spurious patterns are generally the linear combinations and complements of stored patterns.

The main approach of Hopfield and many others for the convergence properties to stable states is to define a so called energy function and to show that this energy function is nondecreasing when the state of the network changes. Since the energy function is bounded from above, it follows that it will converge to some value. In other words, memory vectors or the states to which the network converges are local minima of this energy function. This means that the networks are performing optimization of a well defined function. Unfortunately, there is no guarantee that the network will find the best minimum. Initial states, which are queries to memory vectors, contain only partial or erroneous information about memory vectors. And the network through iteration simply finds the minimum that best fits the query.

In this thesis, we will not employ a Lyapunov or energy function approach to investigate the design and stability properties of Hopfield neural networks but rather use more direct methods.

4.1.2 Hopfield Model

Hopfield network is constructed by connecting a large number of simple processing units to each other by links having fixed weights. These weights are modeled after the synaptic connections between the real neurons. The processing units stand for the real neurons where each one of them is a nonlinear function chosen to be a monotonously increasing and usually continuous function of its arguments.

Each neuron has a current state (or input) and an output. It has feedback connections from the outputs of all the other neurons as well as itself and its current state is a weighted sum of all the outputs of the previous instant. This current state is processed by the neuron i.e. mathematically, undergoes a nonlinear function and the value of the function is simply the output of the neuron. The function in the neuron is usually the hard-limiter function or a sigmoid type nonlinearity as shown in Fig. 4.1.

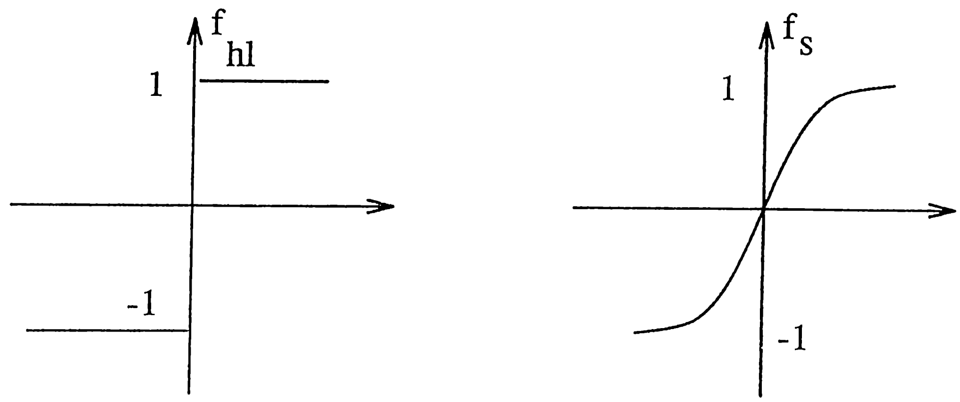

Figure 4.1: Hard-Limiter and Sigmoid functions

The output of the j^{th} neuron is fed to the input of the i^{th} neuron by a connection weight t_{ij}. Therefore the output of the j^{th} neuron is first multiplied by t_{ij} and then fed back to the input of the i^{th} one and the total input of the i^{th} neuron is the summation over all feedbacks from the other neurons including itself. In addition each neuron has an offset bias of i_i^b fed to its input.

Hopfield model has two versions:
i)Discrete-time
ii)Continuous-time

In discrete-time version, the outputs of the neurons are updated at discrete time instants whereas in continuous-time, they are updated continuously. We use synchronous update rule for both of the versions in this paper.

4.2 Motivation

Although Hopfield model and its content addressability have been studied before, the main motivation for this study has been the lack of existence of a general associative memory design and the conditions to check the quality of the design.

We begin with asking some questions:

1) What are the necessary and sufficient conditions on $\mathbf{T}$ such that for the discrete-time model (using hard-limiter functions in neurons), all possible 2^N binary vectors of dimension N are the fixed points of $\mathbf{T}$?

2) Given $\{\mathbf{v}_1, \ldots, \mathbf{v}_M\}$ binary memory vectors ($M \leq 2^N$), find possibly all

$\mathbf{T}$ matrices which store this set. For a specific $\mathbf{T}$, what can be said about the stability of memory vectors?

3) Apply the second question to the continuous-time case.

4) In discrete-time model, instead of hard-limiter function, use sigmoid type nonlinearities and apply the questions 1 and 2.

Above questions will be the subject of the subsequent sections.

4.3 Discrete-Time Case

Let's denote the outputs of the neurons by the vector $\mathbf{x}$, the connection weights by the matrix $\mathbf{T}$, and the offset biases by the vector $\mathbf{i}^b$, i.e. for an N neuron network $\mathbf{T}$ is an $N \times N$ matrix with real components t_{ij} (t_{ij} being the connection weight from the output of j^{th} neuron to the input of i^{th} neuron), $\mathbf{x}$ is a real n_vector with components x_i (x_i being the output of i^{th} neuron) and $\mathbf{i}^b$ is a real n_vector with components $\mathbf{i}^b$ (i_i^b being the offset bias of i^{th} neuron). Let $f(.) : \Re \rightarrow \Re$ be a function and let $\mathbf{v} \in \Re^n$ be a vector. By f(v), it is meant that f is applied to all of the entries of v.

Then the dynamics of the discrete-time network is described by the following difference equation

$$\mathbf{x}_{n+1} = f(\mathbf{T}\mathbf{x}_n + \mathbf{i}^b) \tag{4.1}$$

where f(.) is a sigmoid type nonlinearity or signum function. The aim is to determine $\mathbf{T}$ and $\mathbf{i}^b$ for the particular choice of application.

Here we deal with discrete-time synchronous Hopfield model without the threshold vector $\mathbf{i}^b$, i.e.

$$\mathbf{x}_{n+1} = f(\mathbf{T}\mathbf{x}_n) \tag{4.2}$$

where $\mathbf{x} \in \Re^N$, $\mathbf{T} \in \Re^{N \times N}$ (i.e. the number of the neurons in the network is N). Before defining the design problem, the first question, in the previous section, is solved.

<u>Theorem 3</u>: All 2^N binary vectors of dimension N are the fixed points of (4.2), where f is the hard-limiter (signum) function, if and only if $\mathbf{T}$ (connection

weight matrix) satisfies the following row dominancy condition:

$$t_{ii} > \sum_{j=1, j\neq i}^{N} |t_{ij}|, \quad i = 1, \ldots, N$$

Proof: To be a fixed point, a binary vector must stay in the same quadrant after it is multiplied by $\mathbf{T}$. For i^{th} row of $\mathbf{T}$ ($i = 1, \ldots, N$), the maximum of $\sum_{j=1, j\neq i}^{N} x_j t_{ij}$ (where x_j is 1 or -1) is $\sum_{j=1, j\neq i}^{N} |t_{ij}|$ (since all 2^N binary vectors are considered, this maximum is achieved for each row) and $x_i = sign(\sum_{j=1}^{N} x_j t_{ij})$ iff the above row dominancy condition is satisfied. So to store all 2^N binary vectors, $\mathbf{T}$ should satisfy

$$t_{ii} > \sum_{j=1, j\neq i}^{N} |t_{ij}|, \quad i = 1, \ldots, N$$

If $f(.) : \Re \rightarrow \Re$ is a sigmoid type nonlinearity, it is defined as

$$f(x) = \frac{1 - e^{-kx}}{1 + e^{-kx}} \tag{4.3}$$

where $k > 0$, $k \in \Re$. We consider the following problem.

Design problem: Given a set of vectors $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_M\}$, $\mathbf{m}_i \in (-1, 1)^N$, $i = 1, \ldots, M$, find, if possible, all matrices $\mathbf{T}$ which store $\mathcal{M}$ into the network (i.e. for $i = 1, \ldots, M$; $\mathbf{m}_i$ becomes a fixed point of (4.2) where f is given in (4.3)).

This problem is referred to as fixed-point programming and has been investigated for similar nonlinearities by several researchers, see [2], [4], [5], [12], [15]. The most famous one of these schemes is the outer product rule

$$\mathbf{T} = \sum_{i=1}^{M} \mathbf{m}_i \mathbf{m}_i^T \tag{4.4}$$

which poses some conditions on the memory vectors to be stored. We'll analyze the outer product rule and derive these conditions in section 3.

The solutions found in the literature to the design problem posed above are special solutions, and a characterization of all matrices $\mathbf{T}$ solving the problem stated above has not yet been given.

Let the number of the neurons in the network be N and let $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_M\}$ be the set of vectors we want to store as fixed points. Placing them in columns of a matrix, it is obtained $\mathbf{A} = [\mathbf{m}_1\mathbf{m}_2 \ldots \mathbf{m}_M]$ where $\mathbf{A} \in \Re^{N\times M}$. Then the $\mathbf{T}$ matrix should satisfy the following equation

$$\mathbf{TA} = f^{-1}(\mathbf{A}) \tag{4.5}$$

where $f^{-1}(\mathbf{A}) \in \Re^{N\times M}$ is defined as:

$$[f^{-1}(\mathbf{A})]_{ij} = f^{-1}(a_{ij}) \; i = 1, \ldots, N; j = 1, \ldots, M$$

(i.e. $f^{-1}(.) : (-1, 1) \to \Re$ is applied to all of the entries of the matrix $\mathbf{A}$). Since f is given by (4.3), f^{-1} is defined as

$$f^{-1}(x) = \frac{1}{k} ln(\frac{1+x}{1-x}) \tag{4.6}$$

Applying a singular value decomposition to $\mathbf{A}$ as follows

$$\mathbf{A} = \mathbf{U\Sigma V}^T \tag{4.7}$$

where $\mathbf{U} \in \Re^{N\times N}$, $\mathbf{\Sigma} \in \Re^{N\times M}$,$\mathbf{V} \in \Re^{M\times M}$; $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and $\mathbf{\Sigma}$ is a block-diagonal matrix containing the singular values of $\mathbf{A}$. Partitioning $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}^T$ as

$$\mathbf{U} = [\mathbf{U}_1\,\mathbf{U}_2], \mathbf{\Sigma} = \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \mathbf{V}^T = \begin{pmatrix} \mathbf{V}_1^T \\ \mathbf{V}_2^T \end{pmatrix} \tag{4.8}$$

where $r = rank(\mathbf{A})$, $\mathbf{U}_1 \in \Re^{N\times r}$, $\mathbf{U}_2 \in \Re^{N\times(N-r)}$, $\mathbf{D} \in \Re^{r\times r}$, $\mathbf{V}_1^T \in \Re^{r\times M}$,$\mathbf{V}_2^T \in \Re^{(M-r)\times M}$, $\mathbf{D} = diag(\sigma_1, \ldots, \sigma_r)$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, σ_i's are the singular values of $\mathbf{A}$.

Then, combining (4.7) & (4.8) and putting in (4.5) will yield

$$\mathbf{TU}_1 = f^{-1}(\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1} \tag{4.9}$$

In order to find the matrices $\mathbf{T}$ which satisfy (4.9), $\mathbf{U}_1$ is concatenated with $\mathbf{U}_2$, which results in the following equation:

$$\mathbf{T}[\mathbf{U}_1\,\mathbf{U}_2] = [f^{-1}(\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1}\;\; \mathbf{U}'] \tag{4.10}$$

Here $\mathbf{U}'$ is any real $N \times (N-r)$ matrix. Hence it is obtained

$$\mathbf{T} = f^{-1}(\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1}\mathbf{U}_1^T + \mathbf{U}'\mathbf{U}_2^T \tag{4.11}$$

The Eq. (4.11) characterizes all possible solutions of the design problem stated in this section. It is obvious that, with this choice of $\mathbf{T}$, the vectors $\mathbf{m}_i$, $i = 1, \ldots, M$, become fixed points of (4.2), but the stability of these equilibria is not guaranteed a priori.

Stability: For the Hopfield model to function as a CAM, obviously it must be able to recover an original memory vector when presented with a probe vector close to it (usually in terms of Hamming distance). If the probe vector is considered as a corrupted version of the original memory vector, then it is possible to view the network's operation as a form of error correction. Therefore, each memory vector should be an asymptotically stable equilibrium point of (4.2). Hence each of these memory vectors will have a certain region of attraction, so that any probe vector in that region of attraction will converge to the memory vector.

To determine the stability of the fixed points, we make use of the following theorem.

Theorem 4: Consider the following system

$$\mathbf{x}_{n+1} = \mathbf{f}(\mathbf{x}_n) \tag{4.12}$$

where $\mathbf{f} : \Re^N \rightarrow \Re^N$ is a differentiable function. Let $\mathbf{x}_e$ be an equilibrium of this system (i.e. $\mathbf{f}(\mathbf{x}_e) = \mathbf{x}_e$). If all the eigenvalues of $\partial \mathbf{f}/\partial \mathbf{x}$ at $\mathbf{x} = \mathbf{x}_e$ are inside the unit disc (i.e. less than 1 in norm), then the equilibrium point $\mathbf{x}_e$ of (4.12) is asymptotically stable.

Proof: See [16].

The following corollary easily follows from Theorem 1:

Corollary 1: Let $\mathbf{x}_e$ be an equilibrium of (4.2). Let

$\mathbf{F} = \partial f/\partial \mathbf{x}|_{\mathbf{x}=f^{-1}(\mathbf{x}_e)}$, where $\partial f/\partial \mathbf{x}$ is the Jacobian given by $\partial f/\partial \mathbf{x} = diag(\partial f/\partial x_1, \ldots, \partial f/\partial x_N)$.

If all the eigenvalues of $\mathbf{FT}$ are inside the unit disc, then $\mathbf{x}_e$ is an asymptotically stable equilibrium point.

Proof: See Theorem 4 and (4.2).

To make use of the corollary 1 in the stability analysis, we need the following theorem:

<u>Theorem 5</u>: For any $N \times N$ complex matrix $\mathbf{M}$, we have the following interlacing property

$$\sigma_{min}(\mathbf{M}) \leq |\lambda_{min}(\mathbf{M})| \leq |\lambda_{max}(\mathbf{M})| \leq \sigma_{max}(\mathbf{M}) \tag{4.13}$$

where σ_{min}, σ_{max} are the minimum and maximum singular values of $\mathbf{M}$, respectively, and λ_{min}, λ_{max} are the minimum and maximum eigenvalues of $\mathbf{M}$, in absolute value, respectively. We note that $\sigma_{min} = \sqrt{\lambda_{min}(\mathbf{M}^T\mathbf{M})}$, $\sigma_{max} = \sqrt{\lambda_{max}(\mathbf{M}^T\mathbf{M})}$.

Proof: See [18].

As a result,we can write

$$|\lambda_{max}(\mathbf{FT})| \leq \sigma_{max}(\mathbf{FT}) = \|\mathbf{FT}\|_2 \leq \|\mathbf{F}\|_2\|\mathbf{T}\|_2 = \lambda_{max}(\mathbf{F})\sigma_{max}(\mathbf{T}) \tag{4.14}$$

where $\|\mathbf{A}\|_2$ denotes induced 2-norm of the matrix $\mathbf{A}$, and $\|\mathbf{A}\|_2 = \sigma_{max}(\mathbf{A})$, (see [17]).

If we choose

$$\sigma_{max}(\mathbf{T}) \leq \frac{1}{\lambda_{max}(\mathbf{F})} \tag{4.15}$$

then the asymptotic stability of the equilibrium $\mathbf{x}_e$ is guaranteed. Of course, this condition is only a sufficient condition which guarantees the asymptotic stability.

The above condition holds for a single memory vector. Thus, we should generalize for a set of memory vectors. For $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_M\}$, let $\mathbf{F}_i = \partial f/\partial \mathbf{x}|_{\mathbf{x}=f^{-1}(\mathbf{m}_i)}$, $i = 1, \ldots, M$; and

$$\lambda_{max}(\mathbf{F}) = \max_i \lambda_{max}(\mathbf{F}_i) \tag{4.16}$$

Then

$$\sigma_{max}(\mathbf{T}) \leq \frac{1}{\lambda_{max}(\mathbf{F})} \tag{4.17}$$

Since $\mathbf{T} = [f^{-1}(\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1}\ \mathbf{U}']\mathbf{U}^T$, $\sigma_{max}(\mathbf{T}) \leq \|[f^{-1}(\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1}\ \mathbf{U}']\|_2\|\mathbf{U}^T\|_2$. $\mathbf{U}^T$ is unitary, thus $\|\mathbf{U}^T\|_2 = 1$. Also

$$\sigma_{max}(\mathbf{T}) \leq \|[f^{-1}(\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1}\ \mathbf{U}']\|_2 \leq \|f^{-1}(\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1}\|_2 + \|\mathbf{U}'\|_2 \tag{4.18}$$

Hence if we can choose $\mathbf{U}'$ as

$$\|\mathbf{U}'\|_2 \leq \frac{1}{\lambda_{max}(\mathbf{F})} - \|f^{-1}(\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1}\|_2 \tag{4.19}$$

then we guarantee the asymptotic stability of the memory vectors. For the existence of $\mathbf{U}'$, the right side of the above inequality should be non-negative. As the relationship between $1/\lambda_{max}(\mathbf{F})$ and $\|f^{-1}(\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1}\|_2$ is highly nonlinear, it needs further investigation.

Remark 1: To increase the bound on $\sigma_{max}(\mathbf{T})$, one should decrease $\lambda_{max}(\mathbf{F})$, which in turn means one should choose equilibrium points where the slope is small on the sigmoid, i.e. near the asymptotes 1 & -1 (provided a $\mathbf{U}'$ exists for (4.19)).

We note that the same design procedure can be applied if the nonlinearity used in (4.2) is different than the sigmoid type. For example, consider the design problem for the following system:

$$\mathbf{x}_{n+1} = sign(\mathbf{T}\mathbf{x}_n) \tag{4.20}$$

where $\mathbf{x} \in \Re^N$, $\mathbf{T} \in \Re^{N\times N}$ and the signum function is defined as:

$$sign(u) = \begin{cases} 1 & u \geq 0 \\ -1 & u < 0 \end{cases} \tag{4.21}$$

Let $\{\mathbf{m}_1, \ldots, \mathbf{m}_M\}$ be a set of binary vectors to be stored in the neural network given by (4.20). As before, we set $\mathbf{A} = [\mathbf{m}_1\mathbf{m}_2\ldots\mathbf{m}_M]$. Then the $\mathbf{T}$ matrix should satisfy:

$$\mathbf{TA} = \mathbf{P} \tag{4.22}$$

Here $\mathbf{P} \in \Re^{N\times M}$, and should satisfy

1) $sign(a_{ij}) = sign(p_{ij}) \quad i = 1, \ldots, N \; ; \; j = 1, \ldots, M$

2) Row space of $\mathbf{A}$ spans the row space of $\mathbf{P}$.

Applying the singular value decomposition to $\mathbf{A}$

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{4.23}$$

where $\mathbf{U} \in \Re^{N\times N}$, $\mathbf{\Sigma} \in \Re^{N\times M}$,$\mathbf{V} \in \Re^{M\times M}$; $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and $\mathbf{\Sigma}$ is a block-diagonal matrix containing the singular values of $\mathbf{A}$. Partitioning $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}^T$ as

$$\mathbf{U} = [\mathbf{U}_1\, \mathbf{U}_2], \mathbf{\Sigma} = \begin{pmatrix} \mathbf{D} & 0 \\ & \\ 0 & 0 \end{pmatrix}, \mathbf{V}^T = \begin{pmatrix} \mathbf{V}_1^T \\ \\ \mathbf{V}_2^T \end{pmatrix} \tag{4.24}$$

where $r = rank(\mathbf{A})$, $\mathbf{U}_1 \in \Re^{N\times r}$, $\mathbf{U}_2 \in \Re^{N\times(N-r)}$, $\mathbf{D} \in \Re^{r\times r}$, $\mathbf{V}_1^T \in \Re^{r\times M}$, $\mathbf{V}_2^T \in \Re^{(M-r)\times M}$, $\mathbf{D} = diag(\sigma_1, \ldots, \sigma_r)$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, σ_i's are the singular values of $\mathbf{A}$. Applying the same steps in the preceding analysis, it is obtained

$$\mathbf{T} = \mathbf{P}\mathbf{V}_1\mathbf{D}^{-1}\mathbf{U}_1^T + \mathbf{U}'\mathbf{U}_2^T \tag{4.25}$$

where $\mathbf{U}'$ is any real $N \times (N-r)$ matrix. The Eq. (4.25) characterizes all possible solutions of the design problem for the signum type nonlinearity. Note that $\mathbf{P}$ should satisfy the above conditions and $\mathbf{U}'$ is arbitrary. A particular choice is $\mathbf{P} = \tau_1\mathbf{A}$ and $\mathbf{U}' = -\tau_2\mathbf{U}_2$, where τ_1 and τ_2 are arbitrary positive constants. This choice yields $\mathbf{T} = \tau_1\mathbf{U}_1\mathbf{U}_1^T - \tau_2\mathbf{U}_2\mathbf{U}_2^T$ which is the form of $\mathbf{T}$ given in [5]. Observe that $\mathbf{P} = \tau_1\mathbf{A}$ means all the memory vectors are eigenvectors of $\mathbf{T}$ with a single positive eigenvalue τ_1 (see Eq. (4.22)).

In the analysis above, the threshold vector $\mathbf{i}^b$ is not used. Now, $\mathbf{i}^b$ will be inserted into the model to see whether it can be used as a control to reduce the number of spurious memory vectors. With $\mathbf{i}^b$, the model becomes

$$\mathbf{x}_{n+1} = sign(\mathbf{T}\mathbf{x}_n + \mathbf{i}^b) \tag{4.26}$$

Now the question is: given $\mathbf{T}$, the memory vectors $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_M\}$ and the spurious memory vectors $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_S\}$, can $\mathbf{i}^b$ used to reduce the number of spurious memory vectors without disturbing the memory vectors?

We have a simple routine for the solution of this problem: memory vectors are put in columns of a matrix **A**, and spurious memory vectors in columns of a matrix **B**.

$$\mathbf{A} = [\mathbf{m}_1\mathbf{m}_2 \ldots \mathbf{m}_M], \mathbf{B} = [\mathbf{s}_1\mathbf{s}_2 \ldots \mathbf{s}_S]$$

Then

$$\mathbf{T}[\mathbf{A}|\mathbf{B}] = \left(\begin{array}{cccc|ccc} p_{11} & p_{12} & & p_{1M} & r_{11} & & r_{1S} \\ p_{21} & p_{22} & & p_{2M} & r_{21} & & r_{2S} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ p_{N1} & p_{N2} & \ldots & p_{NM} & r_{N1} & \ldots & r_{NS} \end{array}\right)$$

The basic idea of the routine is to give the maximum sign change to the entries of the rows of **TB** by using $\mathbf{i}^b$ without changing the signs of the entries of **TA**.

For each row do the following:

i) Find the minimum absolute valued one of the positive elements of p_{ij}'s ($j = 1, \ldots, M$).

ii) Find the minimum absolute valued one of the negative elements of p_{ij}'s ($j = 1, \ldots, M$).

iii) For the positive elements of r_{ij}'s ($j = 1, \ldots, S$), find the number of ones which are smaller in absolute value than the minimum found in part i.

iv) For the negative elements of r_{ij}'s ($j = 1, \ldots, S$), find the number of ones which are smaller in absolute value than the minimum found in part ii.

v) Take the maximum of the numbers found in part iii and part iv:

v1. if part iii is larger than part iv, choose $i_i^b < 0$ and choose it so that $|p_{im+}| > |i_i^b| > |r_{ij1}| \ldots > |r_{ijk}|$ where p_{im+} is the minimum absolute valued of positive p_{ij}'s and $r_{ij1}, \ldots, r_{ijk}$ are the positive ones of r_{ij}'s which are smaller in absolute value than p_{im+}.

v2. if part iv is larger than part iii, choose $i_i^b > 0$ and choose it so that $|p_{im-}| > |i_i^b| > |r_{ij1}| \ldots > |r_{ijk}|$ where p_{im-} is the minimum absolute valued of negative p_{ij}'s and $r_{ij1}, \ldots, r_{ijk}$ are the negative ones of r_{ij}'s which are smaller in absolute value than p_{im-}.

Example: Let

$\mathbf{m}_1 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ 1 \end{pmatrix}$, $\mathbf{m}_2 = \begin{pmatrix} 1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$ be the memory vectors and

$\mathbf{s}_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$, $\mathbf{s}_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$, $\mathbf{s}_3 = \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix}$, $\mathbf{s}_4 = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix}$ be the spurious ones.

Using the singular value decomposition techniques, $\mathbf{T}$ is chosen as

$$\mathbf{T} = \begin{pmatrix} 0.8 & 0.5 & 0.0167 & 0.45 \\ -0.35 & 1.45 & 0.0167 & 0.25 \\ -0.3 & -0.25 & 1.3167 & -0.2 \\ -0.225 & 0.1 & -0.35 & 1.625 \end{pmatrix}$$

<u>Without the threshold vector $\mathbf{i}^b$:</u>

$\mathbf{r}_1 = (-1\ 1\ 1\ 1)^T$ is converging to $\mathbf{s}_1$, and $\mathbf{r}_2 = (1\ -1\ 1\ -1)^T$ is converging to $-\mathbf{m}_1$. Now

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & -1 \end{pmatrix}$$

$$\mathbf{T}[\mathbf{A}|\mathbf{B}] = \left(\begin{array}{cc|cccc} 1.73 & 0.76 & 1.76 & 0.86 & 0.83 & -0.73 \\ 1.33 & -1.53 & 1.36 & 0.86 & 0.83 & 1.56 \\ -2.06 & 1.06 & 0.56 & 0.96 & -1.66 & 1.56 \\ 1.85 & 0.95 & 1.15 & -2.1 & -1.4 & -1.65 \end{array}\right)$$

1) For the first row, minimum is 0.76, all the positive elements are greater than 0.76 on the right. So $i_1^b = 0$.

2) For 1.33, 0.86 & 0.83 are smaller. For -1.53, there are no negative elements on the right. So choose $i_2^b < 0$ and $1.33 > |i_2^b| > 0.86 > 0.83$. $i_2^b = -1$.

3) For the third row -2.06 is greater in absolute value than -1. 1.06 is greater than 0.56 & 0.96. Thus choose $i_3^b < 0$ and $1.06 > |i_3^b| > 0.96 > 0.56$. $i_3^b = -1$.

4) For the fourth row, minimum is 0.95, but the only positive element 1.15 on the right is greater than 0.95. So $i_4^b = 0$. $\mathbf{i}^b$ becomes $\mathbf{i}^b = (0\ \ -1\ \ -1\ \ 0)^T$.

With the threshold vector $\mathbf{i}^b$, the network becomes

$$\mathbf{x}_{n+1} = Sign(\mathbf{T}\mathbf{x}_n + \mathbf{i}^b)$$

$\mathbf{r}_1 = (-1\ 1\ 1\ 1)^T$ and $\mathbf{s}_1 = (1\ 1\ 1\ 1)^T$ is now converging to $\mathbf{m}_1$. As a result, with the insertion of a proper $\mathbf{i}^b$, it is achieved to reduce the number of spurious memory vectors and also enlarge the region of attraction of memory vectors.

4.3.1 Outer Product Rule

In this subsection, in view of the analysis given above, we analyze the performance of the outer product rule, and give a sufficient condition for this method to work as a design method.

Let the neural net be given by (4.20). Let $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_M\}$ be a set of binary vectors to be stored in the neural net. According to outer product rule, one chooses the following connection weight matrix:

$$\mathbf{T} = \sum_{i=1}^{M} \mathbf{m}_i \mathbf{m}_i^T \tag{4.27}$$

Note that, with this choice, the storage of $\mathbf{m}_i$'s in the neural network as memory vectors is not guaranteed a priori. Our aim is to give sufficient conditions which guarantees this property.

Forming the matrix $\mathbf{A}$ as $\mathbf{A} = [\mathbf{m}_1 \mathbf{m}_2 \ldots \mathbf{m}_M]$ where $\mathbf{A} \in \Re^{N \times M}$, we obtain

$$\mathbf{T} = \sum_{i=1}^{M} \mathbf{m}_i \mathbf{m}_i^T = \mathbf{A}\mathbf{A}^T \tag{4.28}$$

If $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_M$ are orthogonal to each other then $\mathbf{A}^T\mathbf{A}$ is diagonal with the single eigenvalue N on the diagonal. Then

$$\mathbf{T}\mathbf{A} = \mathbf{A}\mathbf{A}^T\mathbf{A} = N\mathbf{A} \tag{4.29}$$

which means, the memory vectors are successfully stored. But the orthogonality condition is a very stringent condition posed on the memory vectors.

Comparing the Equations (4.29) and (4.22), it is seen that $\mathbf{P} = \mathbf{A}\mathbf{A}^T\mathbf{A}$ in the outer product rule. Hence for the method to work, one needs to check the following:

i) Row space of $\mathbf{A}$ spans the row space of $\mathbf{P}$.

ii) $sign(a_{ij}) = sign(p_{ij}) \quad i = 1, \ldots, N \,; j = 1, \ldots, M$

The first of the above conditions is readily satisfied with this choice of $\mathbf{P}$. To check the second condition, the following observation is made:

$$\mathbf{m}_i^T \mathbf{m}_j = N - 2h_{ij} \tag{4.30}$$

where h_{ij} is the Hamming distance between the memory vectors $\mathbf{m}_i$ and $\mathbf{m}_j$. Hence for $i = 1, \ldots, M$; $j = 1, \ldots, M$, we have:

$$[\mathbf{A}^T\mathbf{A}]_{ij} = \begin{cases} N & i = j \\ N - 2h_{ij} & i \neq j \end{cases} \tag{4.31}$$

For the second condition stated above to be satisfied, $\mathbf{TA}$ and $\mathbf{A}$ should be in the same quadrant of $\Re^N$, columnwise. A sufficient condition which guarantees this is the following

$$\sum_{k=1, k\neq i}^{M} |N - 2h_{ki}| < N, \;\; i = 1, \ldots, M \tag{4.32}$$

Note that for two orthogonal vectors, the Hamming distance between them is $N/2$. Thus for a set of orthogonal memory vectors, the above inequality is readily satisfied. The above analysis then implies that, for the outer product rule to be used as a design method, the memory vectors should have pairwise Hamming distances close to $N/2$, that is they should be nearly orthogonal.

4.4 Continuous-Time Case

Hopfield [1] has also constructed a model that has been based on continuous variables and responses but retained all the significant behaviour of the original discrete model. He has realized an analog electronic counterpart of the nerve system using capacitors, resistors and amplifiers. The equation governing the electronic circuit is nonlinear and is given as follows

$$C_i(du_i/dt) = \sum_j t_{ij} g(u_j) - u_i/R_i + I_i \tag{4.33}$$

where u_i is the input to nonlinear amplifier modeled after the real neuron or more explicitly for neurons exhibiting action potentials, u_i could be thought of as the mean soma potential of a neuron from the total effect of its excitatory and inhibitory inputs. I_i is any other (fixed) input current to neuron i which can be regarded as the offset bias of the neurons. C_i and R_i stand for the input capacitance of the cell membranes and the transmembrane resistance, respectively. t_{ij} is again the connection weight from output of j^{th} neuron to the input of i^{th} one , i.e. a finite impedance in the electronic model.

Collectively denoting the inputs by the vector $\mathbf{u}$, and using the same time constant for all neurons, the dynamics of the network is described by the following differential equation

$$\dot{\mathbf{u}} = -\frac{1}{\tau}\mathbf{u} + \mathbf{T}g(\mathbf{u}) + \mathbf{i}^b \tag{4.34}$$

where $g(.) : \Re \rightarrow \Re$ is a sigmoid type nonlinearity or signum function. For the continuous-time model, we deal with the synchronous update rule, without the threshold vector $\mathbf{i}^b$, i.e.

$$\dot{\mathbf{u}} = -\frac{1}{\tau}\mathbf{u} + \mathbf{T}g(\mathbf{u}) \tag{4.35}$$

Let's consider the same design problem in the previous section. For the equilibrium condition, the differential equation is set to 0.

$$\dot{\mathbf{u}} = -\frac{1}{\tau}\mathbf{u} + \mathbf{T}g(\mathbf{u}) = 0 \tag{4.36}$$

Let $\mathbf{A}$ again be $\mathbf{A} = [\mathbf{m}_1\mathbf{m}_2 \ldots \mathbf{m}_M]$, where $\mathbf{m}_i \in \Re^N$, $i = 1, \ldots, N$, are the memory vectors to be stored in the network as equilibrium points.Then

$$-\frac{1}{\tau}\mathbf{A} + \mathbf{T}g(\mathbf{A}) = 0 \tag{4.37}$$

or equivalently, $\mathbf{T}g(\mathbf{A}) = (1/\tau)\mathbf{A}$. Applying the singular value decomposition to g($\mathbf{A}$), we obtain:

$$g(\mathbf{A}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{4.38}$$

where $\mathbf{U} \in \Re^{N\times N}$, $\mathbf{\Sigma} \in \Re^{N\times M}$,$\mathbf{V} \in \Re^{M\times M}$; $\mathbf{U}$ and $\mathbf{V}$ are unitary matrices and $\mathbf{\Sigma}$ is a block-diagonal matrix containing the singular values of $g(\mathbf{A})$. Partitioning $\mathbf{U}$, $\mathbf{\Sigma}$ and $\mathbf{V}^T$ as before

$$\mathbf{U} = [\mathbf{U}_1\,\mathbf{U}_2], \mathbf{\Sigma} = \begin{pmatrix} \mathbf{D} & \mathbf{0} \\ & \\ 0 & 0 \end{pmatrix}, \mathbf{V}^T = \begin{pmatrix} \mathbf{V}_1^T \\ \\ \mathbf{V}_2^T \end{pmatrix} \tag{4.39}$$

where $r = rank(\mathbf{A})$, $\mathbf{U}_1 \in \Re^{N\times r}$, $\mathbf{U}_2 \in \Re^{N\times(N-r)}$, $\mathbf{D} \in \Re^{r\times r}$, $\mathbf{V}_1^T \in \Re^{r\times M}$, $\mathbf{V}_2^T \in \Re^{(M-r)\times M}$, $\mathbf{D} = diag(\sigma_1, \ldots, \sigma_r)$, $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$, σ_i's are the singular values of $g(\mathbf{A})$. Hence, using the methods employed in the previous sections, it is obtained:

$$\mathbf{T} = \frac{1}{\tau}\mathbf{A}\mathbf{V}_1\mathbf{D}^{-1}\mathbf{U}_1^T + \mathbf{U}'\mathbf{U}_2^T \tag{4.40}$$

Here, again $\mathbf{U}'$ is any real $N\times(N-r)$ matrix. The above equation characterizes all possible solutions of the design problem for continuous-time model.

To investigate the asymptotic stability of the equilibrium points, we make use of the following theorem:

Theorem 6: Let

$$\dot{\mathbf{u}} = \mathrm{g}(\mathbf{u}) \tag{4.41}$$

where $\mathrm{g} : \Re^N \to \Re^N$ is a differentiable function. Let $\mathbf{u}_e$ be an equilibrium of this system (i.e. $\mathrm{g}(\mathbf{u}_e) = 0$). If all the eigenvalues of $\partial \mathrm{g}/\partial \mathbf{u}$ at $\mathbf{u} = \mathbf{u}_e$ have negative real parts, then the equilibrium point $\mathbf{u}_e$ of (4.41) is asymptotically stable.

Proof: See [16].

The following corollary easily follows from the Theorem 6:

Corollary 2: Let $\mathbf{u}_e$ be an equilibrium of the system (4.35), where g is a sigmoid type nonlinearity. Then if all the eigenvalues of $(-1/\tau)\mathbf{I}+\mathbf{T}\partial g/\partial \mathbf{u}|_{\mathbf{u}=\mathbf{u}_e}$ have negative real parts, then $\mathbf{u}_e$ is asymptotically stable. Here $\partial g/\partial \mathbf{u}$ is the Jacobian given by $\partial g/\partial \mathbf{u} = diag(\partial g/\partial u_1, \ldots, \partial g/\partial u_N)$.

Proof: See Theorem 6 and (4.35).

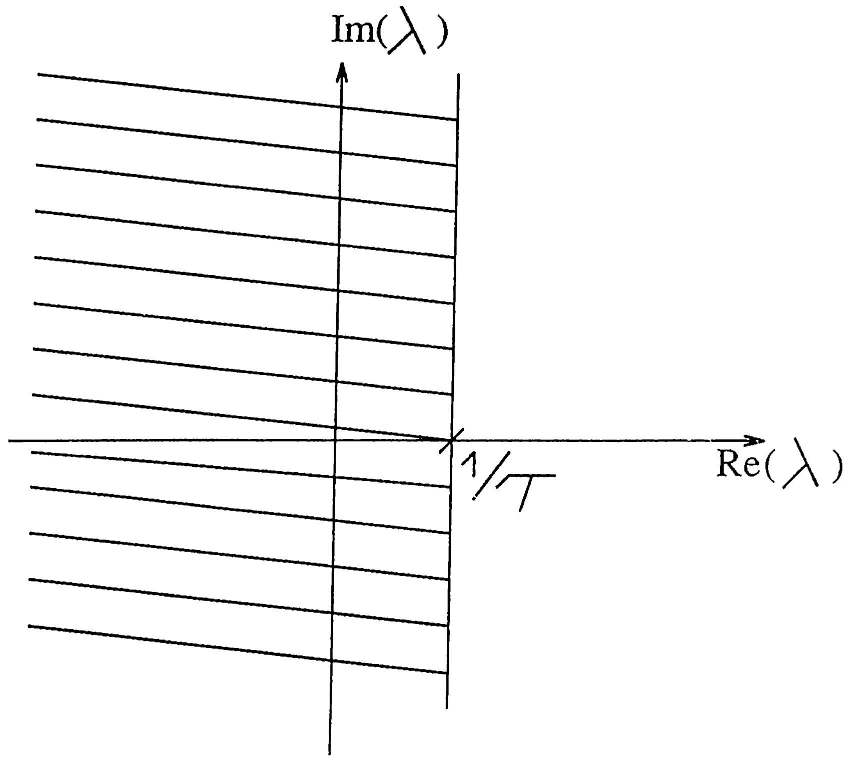

Figure 4.2: Safe region for the eigenvalues of $\mathbf{TG}$

Let $\mathbf{G} = \partial g/\partial \mathbf{u}|_{\mathbf{u}=\mathbf{u}_e}$. Then the eigenvalues of $((-1/\tau)\mathbf{I} + \mathbf{TG})$ should be in the left-half of the complex plane. Since the eigenvalues of $((-1/\tau)\mathbf{I} + \mathbf{TG})$ are the eigenvalues of $\mathbf{TG}$ shifted to the left by $1/\tau$, the eigenvalues of $\mathbf{TG}$ should be to the left of the line $Re(\lambda) = 1/\tau$, as shown in Fig. 4.2.

Hence, the smaller the τ is, the greater degree of freedom one has in choosing the connection weight matrix $\mathbf{T}$ without altering asymptotic stability.

Using the Theorem 5, it is obtained:

$$|\lambda_{max}(\mathbf{TG})| \leq \sigma_{max}(\mathbf{TG}) = \|\mathbf{TG}\|_2 \leq \|\mathbf{T}\|_2\|\mathbf{G}\|_2 = \sigma_{max}(\mathbf{T})\lambda_{max}(\mathbf{G}) \quad (4.42)$$

If one chooses

$$\sigma_{max}(\mathbf{T})\lambda_{max}(\mathbf{G}) \leq \frac{1}{\tau} \longrightarrow \sigma_{max}(\mathbf{T}) \leq \frac{1}{\tau\lambda_{max}(\mathbf{G})} \quad (4.43)$$

then the asymptotic stability of the equilibrium $\mathbf{u}_e$ is guaranteed. In fact, with this choice, the eigenvalues of $\mathbf{TG}$ will be confined in the disc whose center is at the origin and whose radius is $1/\tau$.

To generalize for a set of memory vectors $\mathcal{M} = \{\mathbf{m}_1, \ldots, \mathbf{m}_M\}$, define

$\mathbf{G}_i = \partial g/\partial \mathbf{u}|_{\mathbf{u}=\mathbf{m}_i}$, $i = 1, \ldots, M$; and

$$\lambda_{max}(\mathbf{G}) = \max_i \lambda_{max}(\mathbf{G}_i) \tag{4.44}$$

Then

$$\sigma_{max}(\mathbf{T}) \leq \frac{1}{\tau \lambda_{max}(\mathbf{G})} \tag{4.45}$$

Since $\mathbf{T} = [(1/\tau)\mathbf{A}\mathbf{V}_1\mathbf{D}^{-1}\ \mathbf{U}']\mathbf{U}^T$, $\sigma_{max}(\mathbf{T}) \leq \|[(1/\tau)\mathbf{A}\mathbf{V}_1\mathbf{D}^{-1}\ \mathbf{U}']\|_2\|\mathbf{U}^T\|_2$. $\mathbf{U}^T$ is unitary, thus $\|\mathbf{U}^T\|_2 = 1$. Also

$$\sigma_{max}(\mathbf{T}) \leq \|[\frac{1}{\tau}\mathbf{A}\mathbf{V}_1\mathbf{D}^{-1}\ \mathbf{U}']\|_2 \leq \|\frac{1}{\tau}\mathbf{A})\mathbf{V}_1\mathbf{D}^{-1}\|_2 + \|\mathbf{U}'\|_2 \tag{4.46}$$

As a result, if one can choose $\mathbf{U}'$ as

$$\|\mathbf{U}'\|_2 \leq \frac{1}{\tau\lambda_{max}(\mathbf{G})} - \|\frac{1}{\tau}\mathbf{A}\mathbf{V}_1\mathbf{D}^{-1}\|_2 \tag{4.47}$$

the memory vectors are guaranteed to be asymptotically stable. For the existence of $\mathbf{U}'$, the right side of the above inequality should be non-negative which needs further investigation.

Remark 2: Decreasing $\lambda_{max}(\mathbf{G})$ will increase the degree of freedom one has in choosing τ without endangering asymptotic stability. To decrease $\lambda_{max}(\mathbf{G})$, one should choose the equilibrium points on the sigmoid where the slope is small, i.e. close to the asymptotes 1 and -1 (provided a $\mathbf{U}'$ exists for (4.47)).

As before, note that the condition given above is only a sufficient condition which guarantees the asymptotic stability. One interesting thing about the conditions for stability is that they are almost the same for continuous and discrete-time models. The bound on the norm of connection matrix $\mathbf{T}$ is inversely proportional to the maximum eigenvalue of the Jacobian of the nonlinearity (evaluated at the equilibrium points) for both of the models.

Example: Let $k = 1$ in (4), $\tau = 1$ in (4.37) and

$$\mathbf{x}_{e_1} = \begin{pmatrix} 0.9 \\ 0.95 \end{pmatrix}, \quad \mathbf{x}_{e_2} = \begin{pmatrix} 0.95 \\ -0.9 \end{pmatrix}$$

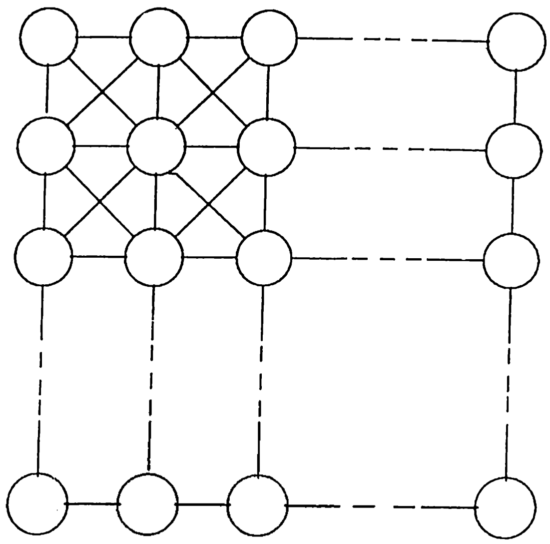

Figure 4.3: Topology of a cellular neural network

$$\mathbf{A} = \begin{pmatrix} 0.9 & 0.95 \\ 0.95 & -0.9 \end{pmatrix}, \; g(\mathbf{A}) = \begin{pmatrix} 0.4219 & 0.4422 \\ 0.4422 & -0.4219 \end{pmatrix}, \; \mathbf{T} = \begin{pmatrix} 2.1411 & -0.0076 \\ 0.0076 & 2.1411 \end{pmatrix}$$

Eigenvalues of $\mathbf{TG}_1$ and $\mathbf{TG}_2$ come out to be 0.8617, 0.8795 which are to the left of the line $Re(\lambda) = 1/\tau = 1$. Hence, the above condition is satisfied. With initial condition $\mathbf{x}_{01} = (0.925\ 0.925)^T$ for $\mathbf{x}_{e_1}$ and $\mathbf{x}_{02} = (0.9\ \ -0.95)^T$ for $\mathbf{x}_{e_2}$, simulations have showed convergence.

4.5 Cellular Neural Networks

In cellular neural networks [20], each unit is called a cell and the structure is similar to that found in cellular automata (see Fig. 4.3). Any cell in the network is connected only to its neighbor cells and they can interact directly with each other. Cells which are not neighbors can affect each other indirectly because of the propagation effects of the network. To make neighborhood clear, r-neighborhood of a cell C(i,j), in a cellular neural network is defined by $N_r(i,j) = \{C(k,l) \mid max|k-i|, |l-j| \leq r, 1 \leq k \leq M, 1 \leq l \leq N\}$ where r is a positive integer number.

Our concern with cellular neural networks is fixed-point programming [21], as in the case of Hopfield model. The design task in this case is finding the connection weights for a certain neighborhood so that a given set of binary patterns $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_K$ are fixed points of the net.

We use discrete time-model with synchronous update rule. At time t, each cell takes feedback from its neighbor cells and total feedback is passed from a signum function, the output of the function becoming the output of that cell. The feedback coming from a neighbor cell is simply the multiplication of the weight connecting the two cells by the output of that neighbor cell at time $(t-1)$. Here, the case $r = 1$ is considered.

For cellular neural network structures, patterns are in the form of $M \times N$ matrices (see Fig. 4.3). To be able to do the same analysis as we have done in the Hopfield case, we bring $M \times N$ matrices into vectors of $MN \times 1$ as below:

$$\mathbf{U}_i = [\mathbf{u}_{i1}\mathbf{u}_{i2}\ldots\mathbf{u}_{iN}] \tag{4.48}$$

$$\mathbf{U}'_i = \begin{pmatrix} \mathbf{u}_{i1} \\ \mathbf{u}_{i2} \\ \vdots \\ \mathbf{u}_{iN} \end{pmatrix} \tag{4.49}$$

Now the problem resembles to that of Hopfield but because of the restriction of communicating with only neighbors, $\mathbf{T}$ has the special form given below:

$$\mathbf{T} = \begin{pmatrix} \mathbf{T}_{11} & \mathbf{T}_{12} & 0 & 0 & 0 & \\ \mathbf{T}_{21} & \mathbf{T}_{22} & \mathbf{T}_{23} & 0 & 0 & \\ 0 & \mathbf{T}_{32} & \mathbf{T}_{33} & \mathbf{T}_{34} & 0 & \\ & & & & & \\ & & & & & \\ 0 & 0 & \ldots & \mathbf{T}_{(M-1)(M-2)} & \mathbf{T}_{(M-1)(M-1)} & \mathbf{T}_{(M-1)M} \\ 0 & 0 & 0 & & \mathbf{T}_{M(M-1)} & \mathbf{T}_{MM} \end{pmatrix} \tag{4.50}$$

where each nonzero block above is in the form of:

$$\mathbf{T}_{ij} = \begin{pmatrix} B_1 & C_1 & 0 & 0 & 0 & \\ A_2 & B_2 & C_2 & 0 & 0 & \\ 0 & A_3 & B_3 & C_3 & 0 & \\ & & & & & \\ \dots & \dots & \dots & \dots & & \\ & & & & & \\ 0 & 0 & \dots & A_{M-1} & B_{M-1} & C_{M-1} \\ 0 & 0 & 0 & & A_M & B_M \end{pmatrix} \qquad (4.51)$$

($\mathbf{T}_{ij}$ is one of the nonzero blocks given in Eq. (4.50)). An example is provided here to illustrate a specific choice of $\mathbf{T}$. The parameters of each nonzero block $\mathbf{T}_{ij}$ are chosen as:

If <u>M is even</u>

$B_1 = B_2 = \dots = B_M = C_1 = C_3 = \dots = C_{M-1} = A_2 = A_4 = \dots = A_M$
and

$C_2 = C_4 = \dots = C_{M-2} = A_3 = A_5 = \dots = A_{M-1} = 0$

If <u>M is odd</u>

$B_1 = B_2 = \dots = B_{M-1} = C_1 = C_3 = \dots = C_{M-2} = A_2 = A_4 = \dots = A_{M-1}$ and

$C_2 = C_4 = \dots = C_{M-1} = A_3 = A_5 = \dots = A_M = B_M = 0$

$$\mathbf{U}'_i = \begin{pmatrix} \mathbf{u}_{i1} \\ \mathbf{u}_{i2} \\ \vdots \\ \mathbf{u}_{iN} \end{pmatrix} \qquad (4.52)$$

For each vector $\mathbf{U}'_i$ above, partition each subblock into smaller blocks each one containing 2 elements as below (if M is odd, there will be a block containing only one element at the bottom).

$$\mathbf{u}_{ij} = \begin{pmatrix} * \\ * \\ \cdots \\ * \\ * \\ \cdots \\ \\ \cdots \\ * \\ * \end{pmatrix} \tag{4.53}$$

Choose these 2 elements opposite in sign, i.e., 1 and -1. Though this seems to be posing a condition on memory vectors that are to be stored, this is not very serious because there are $2^{[MN/2]}$ vectors which satisfy the above condition([.] denoting the greatest integer function).

Chapter 5

Conclusion

For the CAM design problem, we have presented a characterization of connection weights for the continuous and discrete Hopfield neural networks, using hyperbolic tangent and signum functions at the neuron outputs.Similar analysis can be used if any other (invertible) nonlinearity is used at the neuron outputs. Our analysis showed that the choice of connection weights depends on an arbitrary matrix when hyperbolic tangent type nonlinearity is used at the neuron outputs (see (4.11) and (4.40)), and depends on two arbitrary matrices when the signum function is used (see (4.25)). The choice of these matrices affects the performance of the neural network. For that purpose, we presented some sufficient conditions (see Corollaries 1, 2 and Equations (4.17), (4.45)) which guarantee the asymptotic stability of the equilibria of the neural network. These conditions pose an upper bound on the norm of the connection weight matrix. For cellular neural networks because of neighborhood constraint, the weight matrix has a special form (see (4.50)) and it brings a restriction on the patterns to be stored.

The choice of the arbitrary matrices mentioned above will affect the stability of the neural network. An interesting problem would be the selection of these matrices so that all the stored memory vectors become asymptotically stable equilibria of the neural network. Another research problem is to incorporate the design methodology presented here with a learning algorithm.

References

[1] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci. U.S.*, vol. 81, pp. 3088-3092, 1984.

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

[3] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biolog. Cybern.*, vol. 52, pp. 1-25, 1985.

[4] S. Aiyer, M. Niranjan, F. Fallside, "A theoretical investigation into the performance of the Hopfield model," *IEEE Trans. on Neu. Netw.*, vol. 1, no. 2, June 1990.

[5] A. Michel, J. Si, G. Yen, "Analysis and synthesis of a class of discrete-time neural networks described on hypercubes," *IEEE Trans. on Neu. Netw.*, vol. 2, no. 1, January 1991.

[6] J.-H. Li, A. Michel, and W. Porod, "Analysis and synthesis of a class of neural networks: Variable structure systems with infinite gains," *IEEE Trans. on Circ. Sys.*, vol. 36, no. 5, pp. 713-731, May 1989.

[7] A. Michel, J. Farrell, and H.-F. Sun, "Analysis and synthesis techniques for Hopfield type synchronous discrete neural networks with application to associative memory," *IEEE Trans. on Circ. Sys.*, vol. 37, no. 11, pp. 1356-1366, November 1990.

[8] G. Yen, A. N. Michel, "A learning and forgetting algorithm in associative memories: Results involving pseudo inverses," *IEEE Symp. Circ. Sys.*, Singapore, June 1991.

[9] J. Bruck, "On the convergence properties of the Hopfield model," *Proc. IEEE*, vol. 78, no. 10, October 1990.

[10] E. Goles, F. Fogelman, and D. Pellegrin, "Decreasing energy functions as a tool for studying threshold networks," *Discrete Appl. Math.*, vol. 12, pp. 261-277, 1985.

[11] E. Goles, "Antisymmetrical neural networks," *Discrete Appl. Math.*, vol. 13, pp. 97-100, 1986.

[12] S. S. Venkatesh, D. Psaltis, "Linear and logarithmic capacities in associative neural networks," *IEEE Trans. Inform. Theory*, vol. IT-35, no. 3, pp. 558-568, 1989.

[13] Y. S. Abu-Mostafa, J. St. Jacques, "Information capacity of the Hopfield model," *IEEE Trans. Inform. Theory*, vol. 31, no. 4, pp. 461-464, July 1985.

[14] R. J. McEliece, C. E. Posner, R. R. Rodemich and S. S. Venkatesh, "The capacity of the Hopfield associative memory," *IEEE Trans. Inform. Theory*, vol. IT-33, vol. 4, pp. 461-482, 1987.

[15] A. Dembo, "On the capacity of associative memories with linear threshold functions," *IEEE Trans. Inform. Theory*, vol. IT-35, no. 4, pp. 709-720, 1989.

[16] M. Hirsh and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra.* New York: Academic Press, 1974.

[17] M. Vidyasagar, *Nonlinear Systems Analysis.* New York: Prentice-Hall, 1978.

[18] D. H. Owens, "The numerical range: A tool for robust stability studies?," *Systems & Control Letters*, vol. 5, no. 3, December 1984.

[19] D. E. Rumelhart, "The appeal of PDP," in D.E.Rumelhart, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* Vol. 1: Foundations. MIT Press, 1986.

[20] L. Chua, L. Yang, "Cellular neural networks: Theory," *IEEE Trans. on Circ. Sys.*, vol. 35, no. 10, October 1988.

[21] S. Tan, J. Hao, J. Vandewalle, "Cellular neural networks as a model of associative memories," *Proc. IEEE CNNA*, pp. 26-35, 1990.

[22] G. H. Golub, C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, pp. 403-420, 1970.

[23] Virginia C. Klema, Alan J. Laub, "The singular value decomposition: Its computation and some applications," *IEEE Trans. on AC.*, vol. AC-25, no. 2, pp. 164-176, 1980.

[24] M. Erkan Savran, Ömer Morgül, "On the design of Hopfield and cellular neural networks," to appear in *Proc. ISCIS*, Antalya-Turkey, October 1991.

[25] M. Erkan Savran, Ömer Morgül, "On the associative memory design for the Hopfield neural network," to appear in *IJCNN*, Singapore, November 1991.

[26] M. Erkan Savran, "Design and stability of Hopfield associative memory," M.S. Thesis, Elec. and Electro. Eng., Bilkent University, Ankara-Turkey, September 1991.