MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES
3 4456 0146027 4

ORNL/TM-10329
(CESAR-86/49)
(CSNP-86/01)

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

On the Stability, Storage Capacity and Design of Nonlinear Continuous Neural Networks

A. Guez
V. Protopopescu
J. Barhen

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A03; Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-10329
CESAR-86/49
CSNP-86/01

Engineering Physics and Mathematics Division

ON THE STABILITY, STORAGE CAPACITY AND DESIGN OF NONLINEAR CONTINUOUS NEURAL NETWORKS

A. Guezt
V. Protopopescu‡
J. Barhen**

Date Published - February 1987

Research sponsored by the
U.S. Air Force Wright Aeronautical Laboratories
under Interagency Agreement DOE-1579-85,
and for the Offices of Military Applications and
Basic Energy Sciences (Engineering Research Program)
of the U.S. Department of Energy

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
operated by
Martin Marietta Energy Systems, Inc.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400

†Visiting scientist from the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104.
‡Center for the Study of Nonlinear Phenomena.
**Center for Engineering Systems Advanced Research.

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES
3 4456 0146027 4

TABLE OF CONTENTS

Section		Page
I.	INTRODUCTION	1
II.	THE NEURAL NETWORK MODEL	5
III.	ASYMPTOTIC STABILITY OF THE NETWORKS' EQUILIBRIA	7
IV.	CAPACITY	13
V.	NEURAL NETWORKS CAN DESIGN NEURAL NETWORKS	19
VI.	DISCUSSION	23
	REFERENCES	25

LIST OF FIGURES

Page

Fig. 1. Case (a): the firing rate is a monotonic, convex function of the mean soma potential..... 15

Fig. 2. Case (b): the firing rate is a monotonic, nonconvex function of the mean soma potential..... 16

Fig. 3. Schematic architecture of a learning/production network..... 21

ABSTRACT

The stability, capacity, and design of a nonlinear, continuous neural network are analyzed. Sufficient conditions for existence and asymptotic stability of the network's equilibria are reduced to a set of piecewise linear inequality relations which can be solved by a feedforward binary network, or by methods such as Fourier Elimination. The stability and capacity of the network is characterized by the post synaptic firing rate function. An N neuron network with sigmoidal firing function is shown to have up to 3^N equilibrium points of which N+1 arbitrary points may always be made stable. This offers a higher capacity than the (0.1-0.2)N obtained in the binary Hopfield network. Moreover, it is shown that by a proper selection of the postsynaptic firing rate function, one can significantly extend the capacity storage of the network.

I. INTRODUCTION

A neural network is a network of a large number of neuron-like subsystems which are dynamically coupled, and exhibit via their collective behavior useful computational features. The resemblance of the neuron-like subsystem model to the actual behavior of the human brain neuron is not essential at this stage[21], mainly due to our as yet limited knowledge about the human brain and our emphasis on computational properties. We will thus follow the tradition established in the relatively successful applications of such networks to the solution of classically complex problems, e.g., sensory-motor control[6], the Traveling Salesman Problem[13], linear programming[30], feature discovery[26], and perceptual inference[9], and refer to them as neural networks, or when used as a general purpose architecture - neurocomputers[8].

Many models have been suggested for neural networks[7,10,11,12,16,19, 21,22,27]. When all the neuron subsystems update their state simultaneously the network is called synchronous, otherwise, we have an asynchronous network. If the state of each neuron is represented with finite resolution we denote it a finite state neural network; else it is a continuous network.

Neural networks may also be classified by their principal operation phases. The "production" phase is the one in which the time evolution of the network's state manifests the useful computational properties sought for. For instance, when a neural network is used in an "associative memory" (i.e., retrieval of information by content), then the convergence of the network state to a stable attractor is the useful activity which, accordingly, is called the "production" phase. The learning/adaptation/

design phase of a neural network, is the stage in which the network "learns", modifies, or designs its internal architecture (see Section 2), as a result of its interaction with the environment (external input) and according to "metarules" which are inherent to the global context within which the network is to be useful[10]. Early neural networks models[16] separate between these operation phases. First the network is operated in a learning mode, where the network state is not allowed to change, but the neuron interconnections (architecture) are modified (designed); then learning ceases and "production" is initiated by exposing the network to external excitation, which yields "useful" (converging) state trajectories. In more recent models[6-8], the two phases are intermixed, allowing the network architecture and state to concurrently evolve, i.e., the network is simultaneously adaptive and productive.

What makes a problem a "good candidate" for a solution approach based on neural networks? This is, of course, strongly related to the so called useful computational features of such networks. Based on the review of many neural networks applications[7,10,13,15,16,17,24,25,26,31], a common set of features was identified. These problems' features are complexity, redundancy, and speed. Specifically, the problems possess a very high dimension variables' space with complex interactions among the variables; the feasible solutions space is large, i.e., the problem has many, almost equally useful different solutions (redundancy); and a solution must be obtained relatively fast, often due to interaction with the real world under real-time constraints.

Although many models have been suggested for neural networks and despite the fact that neural networks models appeared over forty years

ago[20], relatively little work has been done in analysis and synthesis of continuous neural networks models. Reference [4] is to the best of our knowledge, the most general result available to date.

The use and application of neural networks have often been heuristic and "opportunistic" in nature, i.e., the network model has been customized to serve the needs of the task at hand[13,26,30,31]. In other cases, probabilistic or fuzzy approaches have been taken[2,3,14,17,22,23], where average performance and the performance variance of a neural network were analyzed, rather than its exact evolution.

Binary neural networks are extensively discussed in the literature, in particular with respect to their design[10,16], storage capacity, and stability[1,2,3,12,32]. Open issues still prevail however, regarding the design efficiency, implementation, elimination of spurious stable attractors, in the application of binary neural networks.

In this paper, our purpose is to present some results concerning the stability and storage capacity as well as some preliminary tools and concepts for the design of a large class of continuous, nonlinear neural networks. Section II defines a general dynamic model for the network. In Section III, we reduce the problem of designing a set of stable network attractors to a set of linear equations and of piecewise linear inequality conditions. Section IV analyzes the information capacity of the network and demonstrates the results on a simple, two-neuron example. Section V describes our proposed architecture for the combined design/production system. A discussion of future work and of current results is given in Section VI.

II. THE NEURAL NETWORK MODEL

In this section we define the dynamic model of the neural network, following the simple models given in [10] and [12] (but relaxing important constraints on the network's architecture, such as symmetry).

The equations of motion of a N-neuron network are:

$$\frac{dx(t)}{dt} = -A(t)x(t) + Tg(x(t)) + I \tag{1}$$

where:

$x(t)=(x_1(t), x_2(t), .., x_N(t))^T \in R^N$, and $x_i(t)$ is the mean soma potential of the i-th neuron; $g(x(t)) = (g_1(x_1(t)), g_2(x_2(t)),...,g_N(x_N(t)))^T$, with $g_i(x_i)$ a differentiable function, represents the short term average of the firing rate of the i-th neuron as a function of its potential; and $I=(I_1, I_2,...I_N)^T \in R^N$ is the constant external input to the network. A and T are $N \times N$ constant matrices with real entries, A is diagonal. Its diagonal elements, $a_i > 0$, represent the time constant of the rate of change for the i-th neuron's potential. The element T_{ij} is the synaptic efficacy of the j-th neuron potential transmitted to the i-th neuron. The asymmetry of this efficacy is supported by experimental evidence and is reflected in the not necessarily symmetric structure of the matrix T. Symmetry is not required here, since we will not use a Lyapunov functional; moreover, the importance of asymmetry in temporal association and pattern generation/recognition has become increasingly clear[28]. In our approach, $g_i(x_i)$ is not necessarily a sigmoid function. Also, our notation above differs from those given in [12] by the absence of a constant scaling factor C_i, the i-th neuron capacitance.

III. ASYMPTOTIC STABILITY OF THE NETWORKS' EQUILIBRIA

In this section we present sufficient conditions, in the form of linear equations and piecewise linear inequality relations, for the asymptotic stability in the neighborhood of each member of an arbitrary set of state vectors, which are specifically designed to be equilibria points of the network.

Let $x^e \in R^N$ be an arbitrary, fixed vector, which we want to be a stable equilibrium point of the network. The following is a theorem regarding the network architecture which guarantees stability.

<u>Theorem 1</u> $x^e \in R^N$ is an asymptotically stable equilibrium of the network (1) if a_i, T_{ij}, $i,j = 1, .. N$ satisfy:

$$Ax^e - Tg(x^e) = I \tag{2}$$

$$T_{ii} \left.\frac{\partial g_i(x_i)}{\partial x_i}\right|_{x_i=x_i^e} - a_i < 0 \quad , \quad i=1, \ldots, N \ . \tag{3}$$

$$\sum_{\substack{j=1 \\ j\neq i}}^{N} \left| T_{ij} \left.\frac{\partial g_i(x_j)}{\partial x_j}\right|_{x_j=x_j^e} \right| - \left| T_{ii} \left.\frac{\partial g_i(x_i)}{\partial x_i}\right|_{x_i=x_i^e} - a_i \right| < 0 \ , \ i = 1, \ldots, N \ . \tag{4}$$

<u>Proof</u>: Substitute I given by (2) in (1) with $x=x^e$ then

$$\frac{dx^e}{dt} = - Ax^e + Tg(x^e) + Ax^e - Tg(x^e) = 0$$

implying the x^e is an equilibrium point of the network. To guarantee local asymptotic stability about $x=x^e$, we expand the right hand side of Eq. (1) in Taylor series about $x=x^e$, to obtain:

$$\frac{dx(t)}{dt} = -Ax^e+Tg(x^e)+I+(TJ(x^e)-A)(x-x^e)+O(|x-x^e|^2) \tag{5}$$

where $J(x^e)$ is the $N \times N$ Jacobian matrix of $g(x)$ evaluated at $x=x^e$ with $J_{ij}(x)=\frac{\partial g_i(x_i)}{\partial x_j}$, and $O(|x-x^e|^2)$ are terms of second and higher order which are vanishingly small as $x \to x^e$. Recalling that I is fixed, we use (2) to obtain:

$$\frac{dx(t)}{dt} = (TJ(x^e) - A)(x-x^e) + O(|x-x^e|^2) \tag{6}$$

Define $y = x-x^e$, then clearly $y = 0$ for $x=x^e$, and, in terms of y, (6) yields

$$\frac{dy}{dt} = \frac{dx}{dt} - \frac{dx^e}{dt} = \frac{dx}{dt} = (TJ(x^e) - A)y + O(|y|^2) \tag{7}$$

The origin $y=0$ of the system (7) is locally asymptotically stable if the eigenvalues of the constant matrix $TJ(x^e)-A$ are all in the left half plane (LHP)[5]. Thus a sufficient condition for $x=x^e$ to be an asymptotically stable equilibrium of the network is that the eigenvalues of $TJ(x^e)-A$, denoted by $\lambda_i(TJ(x^e)-A)$, i-1, .. N be located in the LHP.

Now we use a famous eigenvalues localization theorem by Gersgorin[18], to obtain conditions on the T_{ij}'s and a_i's to guarantee stability.

<u>Gersgorin's Theorem</u>. The eigenvalues of a real $N \times N$ matrix B are contained in the union of the N discs of the complex λ-plane.

$$|\lambda - B_{ii}| \leq \sum_{\substack{j=1 \\ j \neq i}}^{N} |B_{ij}| \quad . \tag{8}$$

Let $B=TJ(x^e)-A$, since g_k is a function of x_k only, we obtain

$$J_{ij}(x^e)=\begin{cases}\left.\dfrac{\partial g_i(x_i)}{\partial x_i}\right|_{x_i=x_i^e} & , \quad i=j\\ 0 & , \quad i\neq j\end{cases}$$

Therefore,

$$B_{ij}=\begin{cases}T_{ii}\left.\dfrac{\partial g_i(x_i)}{\partial x_i}\right|_{x_i=x_i^e}-a_i & , \quad i=j\\ T_{ij}\left.\dfrac{\partial g_j(x_j)}{\partial x_j}\right|_{x_j=x_j^e} & , \quad i\neq j\end{cases} \tag{9}$$

Applying Gersgorin's Theorem to the matrix B, a sufficient condition for $\lambda_i(B)$ to be in the LHP is that all the discs $\left|\lambda-B_{ii}\right| \underset{\substack{j=1\\ j\neq i}}{\overset{N}{\leq\Sigma}} \left|B_{ij}\right|$ $i=1, \ldots, N$ be in the LHP, i.e., $B_{ii}<0$ and $\left|B_{ii}\right| \underset{j\neq 1}{>\Sigma} \left|B_{ij}\right|$, or

$$T_{ii}\left.\frac{\partial g_i(x_i)}{\partial x_i}\right|_{x_i=x_i^e}-a_i<0, \quad i=1, \ldots, N$$

and

$$\sum_{\substack{j=1\\ j\neq i}}^{N}\left|T_{ij}\left.\frac{\partial g_j(x_j)}{\partial x_j}\right|_{x_j=x_j^e}\right|-\left|T_{ii}\left.\frac{\partial g_i(x_i)}{\partial x_i}\right|_{x_i=x_i^e}-a_i\right|<0$$

$$i=1, \ldots, N \quad ;$$

which are inequalities (3) and (4).

Theorem 1 implies that by selecting the network's architecture, T_{ij} and a_i that satisfy conditions (2), (3) and (4), the vector x^e is guaranteed to be a locally, asymptotically stable equilibrium of the network.

This result may be extended to any finite, arbitrary set of given fixed vectors. Let the set X^e be given as: $x^{e_1}, x^{e_2}, .. x^{e_M}$, sufficient conditions for X^e to be an asymptotically stable equilibria set are given in Lemma 1.

<u>Lemma 1</u>. $X^e = \{x^{e_k}\}_{k=1}^{M}$, $x^{e_k} \in R^N$ is an asymptotically stable equlibria set of the network (1) if a_i, T_{ij}, $i, j = 1, .. N$, satisfy for each $k = 1, 2, .. M$:

$$Ax^{e_k} - Tg(x^{e_k}) = I \tag{10}$$

$$T_{ii} \left. \frac{\partial g_i(x_i)}{\partial x_i} \right|_{x_i = x_i^{e_k}} - a_i < 0 \quad , i = 1, .. N \tag{11}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{N} \left| T_{ij} \left. \frac{\partial g_j(x_j)}{\partial x_j} \right|_{x_j = x_i^{e_k}} \right| - \left| T_{ii} \left. \frac{\partial g_i(x_i)}{\partial x_i} \right|_{x_i = x_i^{e_k}} - a_i \right| < 0 \ , \ i=1 .. N \tag{12}$$

The proof to Lemma 1 is similar to that of Theorem 1 and will thus be omitted.

Notice that the only assumption made regarding $g_i(x_i)$ is its differentiability, thus Lemma 1 is valid for a large class of nonlinear continuous networks.

Conditions (10), (11) and (12) represent NM linear equations and 2NM piecewise linear inequalities in the N^2+N unknowns $\{T_{ij}, a_i\}$, $i, j = 1, \ldots, N$. Their solution can be implemented by Fourier Elimination method[29] or by another neural network as proposed in Section V.

IV. CAPACITY

We shall discuss in this section the storage capacity (memory) of the neural network in relation with different choices of the function g that models the firing rate of the system.

Let us choose a certain $x = x^e$ to be a stable equilibrium of the system (1). Then Eq. (2) provides at most N conditions for the $N^2 + N$ unknown $\{T_{ij}, a_i\}$, therefore the manifold of solutions is at least N^2 dimensional. We remark that the inequalities (3-4) do not reduce further the dimension of this manifold but they affect only its size. Let us pick a solution $\{T^*_{ij}, a^*_i\}$ belonging to this manifold. Plugging it back in Eqs. (2-4) and solving for x we shall find, of course, our original choice x^e, but, in general, we shall find some other solutions as well - all of them compatible with the architecture $\{T^*_{ij}, a^*_j\}$. The number and location of these stable equilibria will now depend only on g. One can proceed analogously by starting from the very beginning with M ascribed stable equilibria $x^{e1}, \ldots x^{eM}$, where M is chosen such as to ensure the compatibility/determination of the system (2-4). Then, one finds a unique solution $\{T_{ij}, a_j\}$ which admits $x^{e1}, \ldots x^{eM}$ as stable equilibria. However, when solving Eqs. (2-4) for x, one finds that, in general, there will be some other solutions as well, $x^{eM+1}, \ldots x^{eL}$, whose number and position depend now only on g.

An important practical question is "how many (stable) equilibria does a certain system have?" For binary systems, the upper bound of the number of equilibria is N and the lower bound is zero[1]. By using continuous g-functions, we want to show that one can increase these bounds, especially the upper bound.

To give a flavor of the various possibilities, let us describe a simplified situation:

Suppose T^{-1} exists, $I \equiv 0$, $T_{ii} = 0$, and $\left.\frac{\partial g_i(x_i)}{\partial x_i}\right|_{x^{e_i}} \cong 0$. Then Eqs. (3) and (4) are automatically satisfied and Eq. (2) can be written in the more convenient form $(T^{-1}Ax)_i = g_i(x_i)$, $i=1,\ldots,N$.

(a) Suppose $g_i(x_i), i=1,\ldots N$ are odd, monotonic, convex functions approaching a saturation level (i.e., the response of each neuron increases with the excitation and, eventually, saturates). The sigmoid ($\tan^{-1}$ - like) function[12] is a generic candidate, but any other similar form can be used (e.g., $g(x) = -1$, $x < -1$, $g(x) = x$, $|x| < 1$, $g(x) = 1$, $x > 1$). Then, for a fixed set $\{T^*_{ij}, a^*_j\}$, Eq. (2) cannot have any other solution except the originally ascribed $x^{e_1},\ldots,x^{e_M}$: each $x_i^{e_j}$ is uniquely determined by the intersection of the hyperplane $(T^{-1}Ax^{e_j})_i$ and $g_i\ (x_i)$. (See Fig. 1). Thus, in this case L=M. How large can M be? In general, Eq. (2) represents N independent linear conditions on $\{T_{ij}, a_j\}$ • In order to fully determine the architecture, a minimum of N+1 ascribed <u>stable</u> equilibria is required. Therefore, $M \leq N+1$. However, because of the additional conditions (3)-(4), we may run into an incompatible system even for M=2. The <u>total</u> number of equilibria, P, is much larger. If g_i is like (a), then for a given architecture there are 3^N equilibria (positive, negative, and zero).

A simple way to increase capacity is to select other functions g. For instance:

(b) Suppose $g_i(x_i)$ is monotonic, nonconvex. This type of function may model a more refined type of response allowing for intermediate quasi-saturation, before the eventual saturation sets in.

ORNL/DWG-86-18228

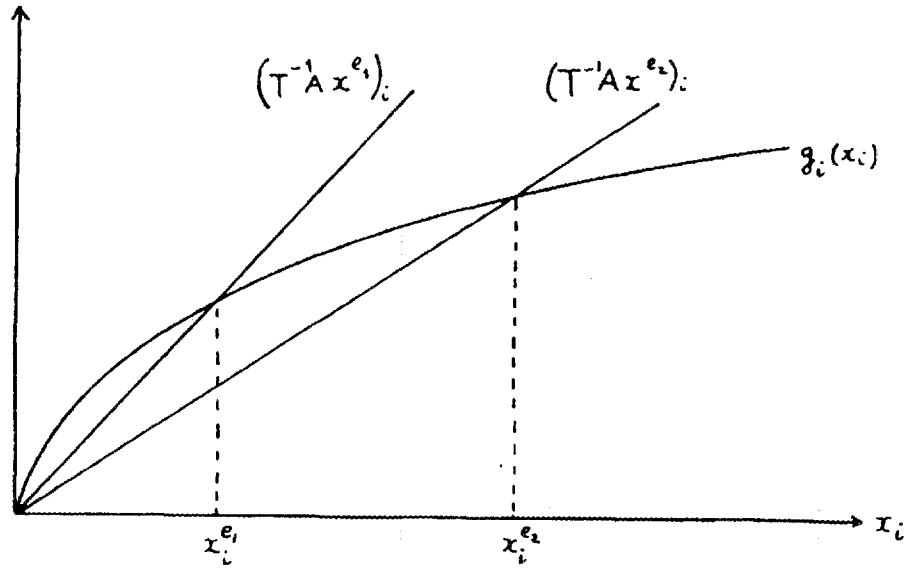

Fig. 1. Case (a): the firing rate is a monotonic, convex function of the mean soma potential.

If x^{e1} has been the originally ascribed stable equilibrium, then x^{e2}, x^{e3},..x^{e1}, x^{e2},... represent acceptable solutions as well. The points x^{e1}, x^{e2}...are stable equilibria only if (4) is satisfied with the corresponding values of $\left.\frac{\partial g_i(x_i)}{\partial x_i}\right|_{x_i = \tilde{x}_i^e}$. (See Fig. 2).

(c) Suppose $g_i(x_i)$ is nonmonotonic, nonconvex. This function can model responses which, temporarily, react inhibitorily to an increase of the input. Although excitation-inhibition modeling is done by the matrix T (the sign of the matrix elements T_{ij}), it may be sometimes convenient to have it included in the function g_i itself.

Depending on the actual form of g_i, we see that the number of equilibria increases and it may be designed as to increase without bound. Yet, there is a trade-off between the increase of capacity and its usefulness insofar as access time and reliable retrievability are concerned. We can summarize and generalize our results in the following way:

ORNL/DWG-86-18229

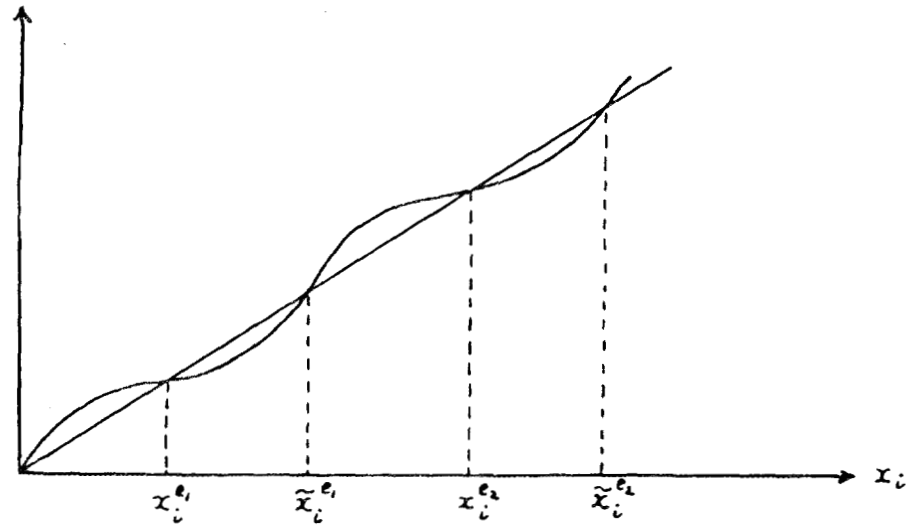

Fig. 2. Case (b): the firing rate is a monotonic, nonconvex function of the mean soma potential.

Let $g_i(x_i)$ be differentiable, bounded and not identically constant, and let T be invertible. Then the <u>total</u> number of equilibria is contained between N+1 and $\prod_{i=1}^{N} k_i$ where

$$k_i = \max_{\alpha,\beta,x_j \neq x_i} \{\text{number of solutions of the equation } g_i(x_i) = \alpha x+\beta\} \quad . \tag{13}$$

The number of <u>stable</u> equilibria is contained between 1 and $\prod_{i=1}^{N} k_i$. Indeed, Eq. (2) that determines the equilibria represents a system of linear equations for the N^2+N unknown $\{T_{ij}, a_j\}$. The minimum number of equations in order to determine them is N^2+N, which implies that we have to specify at least N+1 equilibria. In order that the equilibria be stable we have to take into account also (3) and (4) which impose additional restrictions. One stable equilibrium can always be accommodated, but if we choose x^{e_1} and x^{e_2} oddly enough, it may turn out that there is no architecture that can accommodate them.

The upper bound is obtained by the following reasoning. Equation (2) can be written

$$[T^{-1}(Ax-I)]_i = g_i(x_i) \quad i=1,\ldots, N \quad . \tag{14}$$

By (13), the number of solutions of the i-th equation (14) cannot exceed k_i. The total number of combinations is $\prod_{i=1}^{N} k_i$, which gives the number of equilibria. This is an upper bound also for the stable equilibria, since it may happen that (3) and (4) are finally satisfied.

Remarks. Whether the upper bound for stable equilibria is actually reachable for a given g_i is yet to be determined.

If $\frac{\partial g_i}{\partial x_i}$ is unbounded or identically constant on some intervals, then the stable equilibria may form lines, and $k_i = \infty$. We postpone a more careful discussion of this case for a later publication.

Example. Let $N = 2$, $g_i(x_i) = \tan^{-1}(x_i)$; $x^e = (1, 0)^T$, $I = 0$ then from Eq. (2) we obtain:

$$\begin{cases} -a_1 + T_{11}\frac{\pi}{4} = 0 \\ \\ T_{21}\frac{\pi}{4} = 0 \end{cases}$$

from (3) and (4) we obtain:

$$\frac{1}{2}T_{11} - a, < 0 \quad ; \quad |T_{12}| - |\frac{1}{2}T_{11} - a_1| < 0$$

$$T_{22} - a_2 < 0 \quad ; \quad |\frac{1}{2}T_{21}| - |T_{22} - a_2| < 0$$

for which one solution is: $T_{11} = \frac{4}{\pi}$; $T_{21} = T_{12} = 0$; $T_{22} = \frac{2}{\pi}$;

$a_1 = a_2 = 1$.

For this architecture there are only three equilibrium points, since from (2) we have:

$$\begin{cases} 0 = -x_1 + \frac{4}{\pi}\tan^{-1}(x_1) \\ \\ 0 = -x_2 + \frac{2}{\pi}\tan^{-1}(x_2) \end{cases}$$

or $\{(x_1,x_2)|x_1 = 0 , \pm 1 \ ; x_2 = 0\}$. Among these equilibria only two, namely $\{(x_1, x_2)| \ x_1 = \pm 1; x_2=0\}$, are known to be stable, since the Jacobian

$$\left[\frac{\partial(\dot{x}_i)}{\partial x_j}\right] = \begin{bmatrix} -1 + (\frac{4}{\pi})\frac{1}{1+x_1^2} & ; & 0 \\ 0 & ; & -1 + (\frac{2}{\pi})\frac{1}{1+x_2^2} \end{bmatrix} \quad i,j = 1,2$$

is negative definite there. Following the results of Sections 3 and 4, one cannot guarantee the stability of more than one arbitrary vector. Indeed, if the vectors $x^{e1} = (1,0)$ and $x^{e2} = (0,1)$ are given, then conditions (2), (3) and (4) when applied to this example lead to a contradiction. On the other hand, since in this example $k_1 = 3$, $k_2 = 1$, we indeed have $k_1k_2 = 3$ as the networks maximum storage capacity.

V. NEURAL NETWORKS CAN DESIGN NEURAL NETWORKS

As mentioned in the introduction, two phases exist in the simplest mode of operation of a neural network, namely, the "learning/adaptation/design" phase and the "production" or "useful" phase. In the first, which we hereafter refer to as the design phase, the network architecture, T, A, is to be specified. We assume that the fixed parameters, N, M, the equilibria set X^e and the functions $g_i(x_i)$ are determined by a higher hierarchical level, which determines the scope and application of the network.

An important step in the design phase is the selection of T and A which guarantee the stability of the M patterns in the set X^e. In Section III, it was found that this could be done by the solution of a set of piecewise linear relations in the variables $\{T_{ij}, a_i\}$. However, for a N-neuron network we have N(N+1) unknown variables, which for N in the range 10^3-10^4[16] yields a 10^6-10^8 dimensional problem; this is the "curse of dimensionality" of the network design problem.

A second aspect of the learning/design phase is that it may have to be performed in real time (or near real-time), since information obtained from the environment has its own dynamic flow.

A third, but not the least important feature of the design problem, is its redundancy, i.e., there is a large multiplicity of networks architectures which satisfy the important design feature of stable equilibria. More specifically, for given M arbitrary patterns, the dimension of the solution space (T, A), of a N-neurons network is at least N(N+1)-3NM; for M in the order of N/10[8], we obtain a redundancy factor in the order of N^2.

Thus, three features were identified as characteristics of the network design problem, namely, the high dimensionality, the requirement for real time solution and the redundancy in the solution space. Notice that these three features are exactly the three ingredients which, as described in Section I, are needed to make a problem a good candidate for a neurocomputer application.

We therefore propose to use neural networks for the design of neural networks. Although recent neural networks models allow for the on-line modification of the network architecture* (via a set of differential equations to be satisfied by the synaptic weights)[4,6,7,8,10], no specific design guidelines (to our best knowledge), such as those given in Sections III and IV, are yet available.

A preliminary architecture for the combined learning/production networks system is described in Fig. 3. The networks system is interacting with its environment and higher hierarchical systems that may exist via the input to the learning/design network and the output from the production network. No direct feedback from the last to the first is assumed; however, indirectly, through its effect on the environment and communication with other network systems the production network may signal to the learning/design network the need for redesign.

Due to their different roles, the two networks differ in their dimensions and architectures. The production network, to be useful, must exhibit the performance of a well trained, mature "expert", which we

*Despite the undoubted contribution of evolution (via genetic coding) to the metastructure of mammal brains, the brain "architecture" is constantly under modification[21], i.e., the learning and "production" phases are intermixed.

ORNL/DWG-86-18230

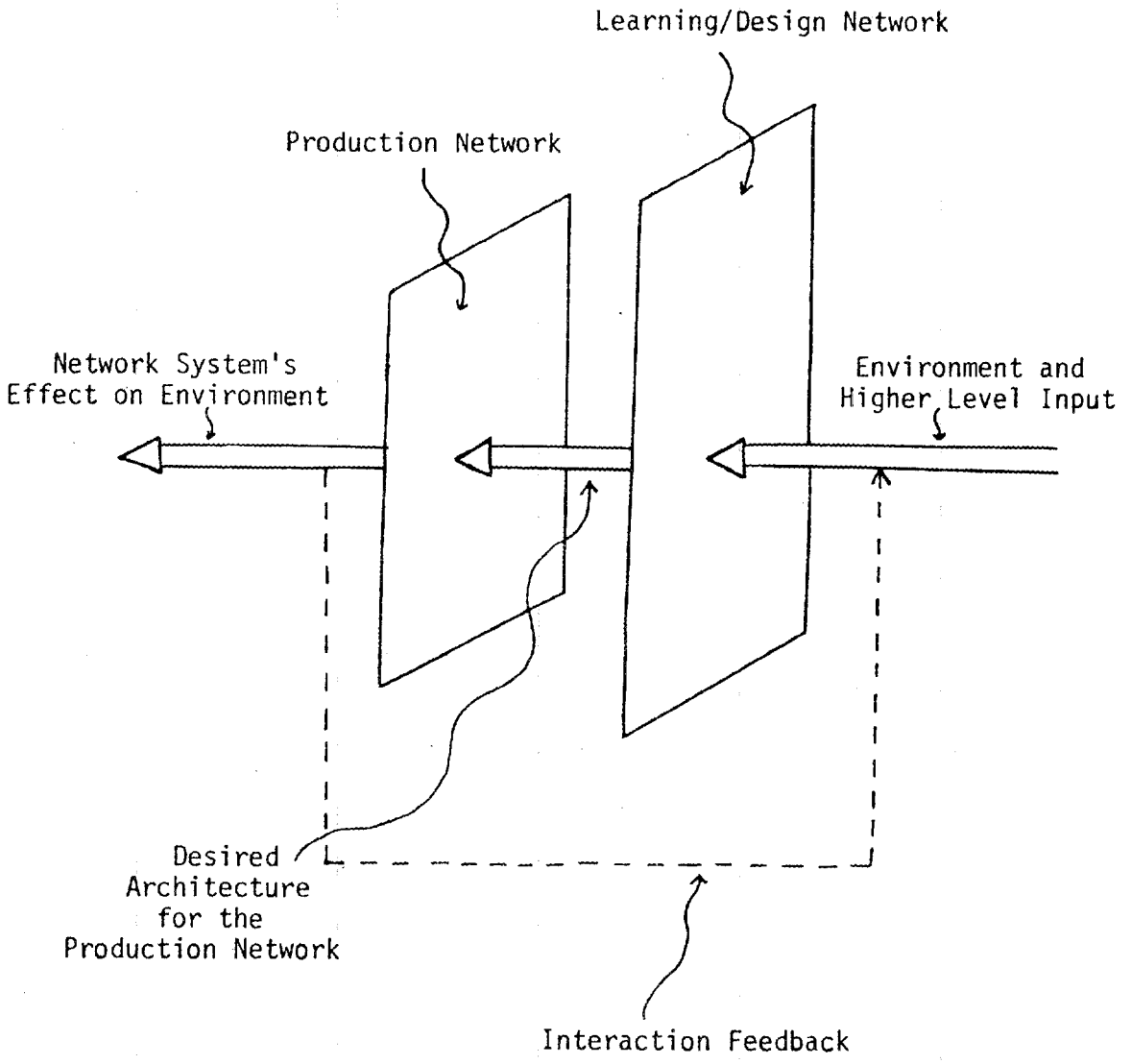

Fig. 3. Schematic architecture of a learning/production network.

interpret as constraints for its design, i.e., fast dynamics/fast convergence (strong stability), continuous representation (high resolution) and nonlinear (high capacity) model. The learning/design network may be slower (as implied by the performance of a "beginning student") and of low resolution (initially only general concepts are learned, specialization is postponed to an expert role); however, its dimension (number of neurons) is high (the primary school syllabus is composed of a large variety of subjects). These properties suggest that a binary model may suffice for the learning/design network model.

As to the legitimate question: "Who" designs the design network? We observe the following. The role of the design network was presented in Sections III and IV, as a solution of a set of piecewise linear inequalities which may be solved via linear programming (with arbitrary cost functions) methods, constraints satisfaction algorithms and categorization/classification techniques. Tank and Hopfield[30] showed how to design a neural network for the solution of a linear programming problem with fast convergence. Although their method was demonstrated on a two variable problem, we expect their technique to be applicable to a much higher dimensional problem since we are only looking for a feasible solution of Eqs. (10-12) and not necessarily for an optimal one. In Refs. [9] and [31] a neural network was used to solve a constraint satisfaction problem. References [9] and [26] use neural networks for classification/categorization (feature discovery and perceptual inference) problems. Thus, a few schemes already exist (and many more will undoubtedly appear) for the implementation of the learning/design network.

VI. DISCUSSION

We have analyzed the stability, capacity and design of nonlinear, continuous neural networks. Sufficient conditions for existence and asymptotic stability of the network's equilibria have been reduced to a set of piecewise linear inequality relations which can be solved by a feedforward binary network or by other methods such as Fourier Elimination. In our model, the stability and capacity of the network depend almost entirely on the postsynaptic firing rate function. An N neuron network with sigmoidal firing function is shown to have up to 3^N equilibria points of which N+1 arbitrary points may always be made stable. This offers a higher capacity than the (0.1-0.2)N obtained in the binary Hopfield network. Moreover, it is shown that by a proper selection of the postsynaptic firing rate function one can significantly extend the capacity storage of the network.

Neural networks, defined as adaptive dynamical systems, can carry out useful information processing by means of their state response to initial or continuous input. Their intrinsic properties, i.e., high connectivity, absence of a system-wide clock (asynchronicity), high degree of fault-tolerence, and "global" speed, emphasize their potential role for advanced robotics applications.

REFERENCES

1. Abu Mustafa, Y. and J. M. Jacques, "Information Capacity of the Hopfield Model," IEEE Tran. Information Theory **IT-31**(4), 461 (1985).

2. Amit, D. J., H. Gutfreund and H. Sompolinsky, "Storing Infinite Numbers of Patterns in a Spin Glass Model of Neural Networks," Physical Review Letters **55**(14), 1530 (1985).

3. Amit, D., H. Gutfreund and H. Sompolinsky, "Spin Glass Models of Neural Networks," Phys. Rev. A **32**(2), 1007 (1985).

4. Cohen, M. A. and S. Grossberg, "Absolute Stability of Global Pattern Formation and Parallel Memory Storage by Competitive Neural Networks," IEEE Trans. Syst. Man Cyber., **SMC-13**(5), 815 (1983).

5. Gibson, J. E., Nonlinear Automatic Control, McGraw-Hill (1963).

6. Grossberg, S. and M. Kuperstein, Neural Dynamics and Adaptive Sensory-Motor Control, North-Holland, Amsterdam (1986).

7. Grossberg, S., Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition and Motor Control, Reidel Press, Boston (1982).

8. Hecht-Nielsen, R., "Performance Limits of Optical, Electro-Optical and Electronic Neurocomputers," Proc. SPIE Adv. Inst. on Hybrid and Optical Computing SPIE **634**, 1986.

9. Hinton, G. E. and T. Sejnnowski, "Optimal Perceptual Inference," Procs. IEEE Conf. Computer Vision and Pattern Recognition, pp. 448-453 (1983).

10. Hinton, G. E. and J. A. Anderson, Parallel Models of an Associative Memory, Lawrence Erlbaum, Hillsdale, NJ (1981).

11. Hopfield, J. J., "Neural Networks and Physical Systems with Emergent Collective Computational Abilities," Proc. Nat. Acad. Sci. USA **79**, 2554 (1982).

12. Hopfield, J. J., "Neurons with Graded Response Have Collective Computational Properties Like Those of Two State Neurons," Proc. Nat. Acad., Sci. USA **81**, 3088 (1984).

13. Hopfield, J. J. and D. W. Tank, "Neural Computations of Decisions in Optimization Problems," Biol. Cyber. **52**, 141 (1985).

14. Kandel, A. and S. C. Lee, Fuzzy Switching and Automata, pp. 208-231, Crane-Russak, New York (1980).

REFERENCES (cont'd)

15. Kinzel, W., "Learning and Pattern Recognition in Spin Glass Models," Z. Phys. B **60**, 205 (1985).

16. Kohonen, T., Self Organization and Associative Memory, Springer-Verlag, New York (1984).

17. Lee, S. C. and E. T. Lee, "Fuzzy Sets and Neural Networks," Journal of Cybernetics **4**(2), 83 (1974).

18. Marcus, M. and H. Minc, "A Survey of Matrix Theory and Matrix Inequalities," Allyn and Bacon, Boston (1964).

19. Maxwell, T., C. Giles, Y. Lee and H. Chen, "Nonlinear Dynamics of Artificial Neural Systems," in Procs. Second Conf. on Neural Networks for Computing, American Institute of Physics (1986, in press).

20. McCullouch,W. S. and W. Pitts, "A Logical Calculus of the Ideas Imminent in Nervous Activity," Bull. Math Biophys. **5** (1983).

21. Palm, G., "Neural Assemblies," Springer-Verlag, New York (1982).

22. Peretto, P. and J. J. Niez, "Stochastic Dynamics of Neural Networks," IEEE Trans. Syst. Man Cyber. **SMC-16**(1), 73 (1986).

23. Peretto, P. and J. J. Niez, "Long Term Memory Storage Capacity of Multiconnected Neural Networks," Biol. Cybern. **54**, 53 (1986).

24. Personnaz, L., I. Guyon and G. Dreyfus, "Information Storage and Retrieval in Spin-Glass Like Neural Networks," J. Physique Lett. **46**, L359 (1985).

25. Port, O., "Computers that Come Awfully Close to Thinking," Business Week, pp. 32-35 (June 2, 1986).

26. Rumelhart, D. and D. Zipser, "Feature Discovery by Competitive Learning," Cognitive Science **9**, 75 (1985).

27. Selverston, A. I., ed., Model Neural Networks and Behavior, Plenum Press, New York (1985).

28. Sompolinsky, H. and I. Kanter, "Temporal Association in Asymmetric Neural Networks," Phys. Rev. Lett. **57**, 2861 (1986).

29. Stoer, J. and C. Witzgall, Convexity and Optimization in Finite Dimensions I, Springer-Verlag, New York (1970).

30. Tank, D. W. and J. J. Hopfield, "Simple Neural Optimization Networks: An A/D Converter, Signal Decision Circuit and a Linear Programming Circuit," IEEE Trans. Circuit and Systems **CAS-33**(5), 533 (1986).

REFERENCES (cont'd)

31. Venkatasubramanian, V., "Inexact Reasoning in Expert Systems: A Stochastic Parallel Network Approach," Procs. Second IEEE Conf. on Artificial Intelligence, pp. 13-15, Miami, Florida (December 1985).

32. Weisbuch, G. and F. Fogelman-Soulié, "Scaling Laws for the Attractors of Hopfield Networks," J. Physique Lett. **46**(14), L623 (1985).

ORNL/TM-10329
CESAR-86/49
CSNP-86/01

INTERNAL DISTRIBUTION

1. V. A. Alexiades
2. R. G. Alsmiller
3. S. M. Babcock
4-8. J. Barhen
9. D. L. Barnett
10. D. E. Bartine
11-15. D. G. Cacuci
16. K. M. Clinard
17. G. de Saussure
18. W. B. Dress
19. J. R. Einstein
20. W. E. Fulkerson
21. L. J. Gray
22. P. M. Haas
23. W. R. Hamel
24. M. R. Hillard
25. E. B. Harris
26. C. C. Jorgensen
27. G. E. Liepins
28. A. S. Loebl
29. J. L. Lucius
30-34. F. C. Maienschein
35. B. F. Maskewitz
36. S. A. Meacham
37. F. R. Mynatt
38. E. M. Oblow
39. R. W. Peelle
40. F. G. Perey
41. R. B. Perez
42. L. M. Petrie
43. F. G. Pin
44-48. V. Protopopescu
49. D. B. Reister
50. R. E. Ricks
51. R. T. Santoro
52. A. D. Solomon
53. N. Toomarian
54. L. D. Trowbridge
55. R. C. Ward
56-60. C. R. Weisbin
61. R. M. Westfall
62. A. Zucker
63. P. W. Dickson, Jr. (Consultant)
64. G. H. Golub (Consultant)
65. R. M. Haralick (Consultant)
66. D. Steiner (Consultant)
67-71. Central Research Library
72. Y-12 Technical Library
Document Reference Section
73-77. Laboratory Records Department
78. Laboratory Records ORNL, RC
79. ORNL Patent Office
80-84. EPMD Reports Office

EXTERNAL DISTRIBUTION

85. Office of the Assistant Manager for Energy Research and Development, Oak Ridge Operations Office, U.S. Department of Energy, Oak Ridge, TN 37831

86. O. P. Manley, Division of Engineering and Geosciences, Office of Basic Energy Sciences, U.S. Department of Energy - Germantown, Washington, DC 20545

87. R. Greenberg, Office of Military Applications, U.S. Department of Energy - Germantown, DP-225, Washington, DC 20545

88. R. LeChevalier, San Francisco Operations Office, U.S. Department of Energy, 1333 Broadway, Oakland, CA 94612

89. J. S. Coleman, Division of Engineering, Mathematical, and Geosciences, Office of Basic Energy Sciences, ER-15, U.S. Department of Energy - Germantown, Washington, DC 20545

EXTERNAL DISTRIBUTION (cont'd)

90. D. M. Austin, Office of Scientific Computing, U.S. Department of Energy, Washington, DC 20545

Air Force Wright Aeronautical Laboratories, Wright Patterson AFB, Ohio 45433

91. G. Waggoner
92. R. Nieter, AART/2
93. R. Bryan, AART/2
94. B. Lem

Rome Air Development Center, Griffiss AFB, NY 13441

95. Lt. Col. C. E. Anderson
96. D. L. Harvey, OCTS
97. Lt. A. Williams, COTC

U.S. Army Human Engineering Laboratory, Aberdeen Proving Grounds, MD 21005-5001

98. J. Weisz
99. J. Stephens
100. C. M. Shoemaker
101. Maj. F. Holly

Jet Propulsion Laboratory, 4800 Oak Grove Drive, Pasadena, CA 91109

102. A. K. Bejczy
103. G. Rodriguez
104. B. Belaram
105. S. Szermay

106. Wayne Book, Department of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332

107. Ewald Heer, Heer Associates, Inc., 5329 Crown Avenue, La Canada, CA 91011

108. John F. Palmer, Chairman, NCUBE Corporation, 915 E. La Vieve Lane, Tempe, AZ 85284

109. Harry E. Pople, Decision Systems Laboratory, University of Pittsburgh, School of Medicine, Pittsburgh, PA 15261

110. Mike Rabins, Dean, Mechanical Engineering Department, Wayne State University, Detroit, MI 48202

111. Karl Reid, Dean, College of Engineering, Architecture and Technology, Oklahoma State University, Stillwater, OK 74074

112-116. A. Guez, Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA 19104

117. J. T. Poore, Chairman, Department of Computer Science, University of Tennessee, Knoxville, TN 37916

118. W. L. Greene, Chairman, Department of Electrical Engineering, University of Tennessee, Knoxville, TN 37916

119-148. Technical Information Center