On the Design of Dynamic Associative Neural Memories

M. Erkan Savran, *Member, IEEE* and Ömer Morgül, *Member, IEEE*

***Abstract*— We consider the design problem for a class of discrete-time and continuous-time neural networks. We obtain a characterization of all connection weights that store a given set of vectors into the network; that is, each given vector becomes an equilibrium point of the network. We also give sufficient conditions that guarantee the asymptotic stability of these equilibrium points.**

I. 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 exploited in areas like pattern recognition and associative memory design, see [8], [7]. The Hopfield model consists of neurons that 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. It is this high degree of connectivity that makes the neural networks computationally attractive. Hopfield showed that with a proper choice of connection weights, the network can perform well as an associative memory or can be used in solving difficult optimization problems such as the travelling salesman problem, [9], [1].

Many researchers proposed various methods to obtain suitable connection weights for specific tasks. In [7], Hopfield used the outer product rule to store a given set of vectors. In [11] and [3], memory vectors were chosen to be linearly independent, in [4] and [10] memory vectors were chosen to be eigenvectors of the connection matrix with positive eigenvalues, and in [2] a design technique based on the construction of an appropriate energy function is introduced.

In this paper we consider a class of discrete-time and continuous-time neural networks. The design problem we consider is to give a characterization of all connection weights that store a given set of vectors into the neural network; that is, each vector becomes an equilibrium point of the network. We obtain such a characterization for both cases and give sufficient conditions that guarantee the asymptotic stability of these equilibrium points.

This paper is organized as follows. In Section II we give the neural network models considered in this paper and state the design problem. In Section III we investigate the design problem for the discrete-time case and give a sufficient condition to ensure asymptotic stability, and in Section IV we investigate the same problem for the continuous time case. Finally, in Section V we give some concluding remarks.

Manuscript received August 14, 1991; revised October 23, 1992.
The authors are with the Department of Electrical and Electronics Engineering, Bilkent University, 06533, Bilkent, Ankara, Turkey.
IEEE Log Number 9207240.

II. Neural Network Models and Problem Statement

We consider both discrete-time and continuous-time neural networks. In the discrete-time case, we consider the following neural network model:

$$x_{n+1} = f(Tx_n), \quad n \in \mathbf{N} \tag{1}$$

where $x \in \mathbf{R}^N$ for some $N \in \mathbf{N}$, which is the number of neurons in the network, $T \in \mathbf{R}^{N\times N}$ is the connection matrix and $f : \mathbf{R} \to \mathbf{R}$ is a nonlinear function. Here, for a vector $v = (v_1 \cdots v_N)' \in \mathbf{R}^N$ the vector $f(v) \in \mathbf{R}^N$ is defined as $f(v) = (f(v_1)\cdots f(v_N))' \in \mathbf{R}^N$ where $'$ stands for transpose. Typically, $f(\cdot)$ is a sigmoid type nonlinearity that is given by:

$$f(x) = \frac{1-e^{-kx}}{1+e^{-kx}} \quad x \in \mathbf{R} \quad k > 0 \tag{2}$$

or a hard-limiter that is given by:

$$f(x) = \begin{cases} 1 & x \geq 0 \\ -1 & x < 0 \end{cases}. \tag{3}$$

Note that (3) may be considered as a limiting case of (2) for $k \to \infty$.

In the continuous-time case, we consider the following neural network model:

$$\dot{x} = -\frac{1}{\tau}x + Tf(x) \tag{4}$$

where $x \in \mathbf{R}^N$, $\tau > 0$ is the time constant of the network, $T \in \mathbf{R}^{N\times N}$ is the connection matrix, $f : \mathbf{R} \to \mathbf{R}$ is a nonlinear function, typically of the sigmoid type given by (2), and a dot denotes the time derivative.

For the neural network models given by (1) or (4), the design problem we consider is the following:

A. Problem (Design)

Let, for some $M \in \mathbf{N}$, the vectors $\mathbf{m}_i \in \mathbf{R}^N$, $i = 1,\cdots,M$, be given. In the discrete-time case we assume that for f given by (2) we have $\mathbf{m}_i \in (-1,1)^N$, and for f given by (3) we have $\mathbf{m}_i \in \{-1,1\}^N$, for $i = 1,\cdots,M$. Find, if possible, all connection matrices T that store the given vectors into the network; that is, each vector $\mathbf{m}_i$ becomes an equilibrium point of the network. □

III. Design of Discrete-Time Neural Networks

We first consider the system given by (1) and (2). Let $\mathbf{m}_i \in (-1, 1)^N$, $i = 1, \cdots, M$ be the vectors to be stored in the network as equilibrium points. Placing these vectors as columns of a matrix, we obtain a matrix $A \in \mathbf{R}^{N \times M}$, which is defined as:

$$A = [\mathbf{m}_1 \mathbf{m}_2 \cdots \mathbf{m}_M]$$

Then, for each $\mathbf{m}_i$ to be an equilibrium of (1), the T matrix must satisfy the following:

$$TA = f^{-1}(A) \tag{5}$$

where $[f^{-1}(A)]_{ij} = f^{-1}(a_{ij})$, $i = 1, \cdots, N$, $j = 1, \cdots, M$.

To find all connection matrices T that satisfy (5), we apply singular value decomposition (SVD) to A as follows:

$$A = U \Sigma V' \tag{6}$$

where $U \in \mathbf{R}^{N \times N}$, $\Sigma \in \mathbf{R}^{N \times M}$, $V \in \mathbf{R}^{M \times M}$; U and V are orthogonal matrices, Σ is a block-diagonal matrix containing the singular values of A; for more information on SVD, see [5]. We partition U, Σ, and V as follows:

$$U = [U_1 U_2], \quad \Sigma = diag\{D, 0\}, \quad V = [V_1 V_2] \tag{7}$$

where $r = \text{rank}\ (A)$, $U_1 \in \mathbf{R}^{N \times r}$, $D \in \mathbf{R}^{r \times r}$, $V_1 \in \mathbf{R}^{M \times r}$, and $D = diag\{\sigma_1, \cdots, \sigma_r\}$, $\sigma_1 \geq \sigma_2 \cdots \sigma_r > 0$, and σ_i denotes the singular values of A. Then by using (6) and (7) in (5), we obtain

$$TU_1 = f^{-1}(A) V_1 D^{-1}. \tag{8}$$

To find all matrices that satisfy (8), we concatenate U_1 with U_2, which results in the following equation:

$$T[U_1 U_2] = [f^{-1}(A) V_1 D^{-1} X] \tag{9}$$

where X is an appropriate $N \times (N - r)$ matrix. Hence we obtain:

$$T = f^{-1}(A) V_1 D^{-1} U_1' + X U_2'. \tag{10}$$

From the above development it is obvious that any matrix T that satisfies (5) is of the form (10) with $X = TU_2$. By reversing the argument, we see that for an arbitrary X, the matrix T given by (10) satisfies (5). Hence we conclude that the equation (10) gives all possible solutions to the design problem stated in this section, with X being a $N \times (N - r)$ arbitrary real matrix.

In order to function as an associative memory, the neural network must be able to recover the full information from a reasonable partial information, hence each stored vector must be an asymptotically stable equilibrium point of the network. Although equation (10) gives all possible connection matrices T that store the given vectors, the stability of these vectors as equilibrium points are not guaranteed *a priori*. In the sequel we give a partial answer to this question and present a sufficient condition to ensure stability.

The stability analysis presented here is based on the linearization of (1) about each equilibrium point. Let $g : \mathbf{R}^N \rightarrow \mathbf{R}^N$ be defined as:

$$g(x) = (f(x_1) \cdots f(x_N))' \tag{11}$$

where $x = (x_1 \cdots x_N)'$ and f is given by (2). We define the matrices F_i as

$$F_i = \frac{\partial g}{\partial x}(f^{-1}(\mathbf{m}_i)), \quad i = 1, \cdots, M. \tag{12}$$

By linearizing (1) about $\mathbf{m}_i$, $i = 1, \cdots, M$, we see that if the eigenvalues of $F_i T$ are inside the unit disc (i.e., less than one in absolute value), then from standard stability theory we conclude that $\mathbf{m}_i$ is an asymptotically stable equilibrium point of (1) (see [6]). The following upper bound on the maximum eigenvalue of $F_i T$ in absolute value can be easily obtained:

$$\lambda_{max}(F_i T) \leq \|F_i\|_2 \|T\|_2 = \lambda_{max}(F_i) \sigma_{max}(T) \tag{13}$$

where $\lambda_{max}(\cdot)$, $\|\cdot\|_2$, and $\sigma_{max}(\cdot)$ denote the maximum eigenvalue in absolute value, the induced 2 norm, and the maximum singular value of a given matrix, respectively. In deriving (13) we used the fact that for any given square matrix Q, $\lambda_{max}(Q) \leq \sigma_{max}(Q)$ and $\|Q\|_2 = \sigma_{max}(Q)$ (see [5]).

Let the quantity Λ_{max} be defined as

$$\Lambda_{max} = \max\{\lambda_{max}(F_1), \cdots, \lambda_{max}(F_M)\}. \tag{14}$$

By using (13), (10), and the fact that $\|U_1\|_2 = \|V_1\|_2 = 1$, we conclude that if X can be chosen as

$$\|X\|_2 < \frac{1}{\Lambda_{max}} - \frac{\|f^{-1}(A)\|_2}{\sigma_{min}(A)}, \tag{15}$$

then all equilibrium points are asymptotically stable, where $\sigma_{min}(A)$ denotes the minimum singular value of A.

For the existence of an X that satisfies (15), the right-hand side of (15) must be strictly positive. For simplicity, assume that $N = 1$ and let $\mathbf{m} \in (-1, 1)$ be the vector to be stored. Let $y \in \mathbf{R}$ be defined as $y = f^{-1}(\mathbf{m})$. In this case, the right-hand side of (15) is always positive if

$$\frac{|\, y \,|\, e^{-y}}{(1 + e^{-y})^2} < 0.5.$$

Since $y = \frac{1}{k} \ln \frac{(1+\mathbf{m})}{(1-\mathbf{m})}$, it follows that this latter inequality is always satisfied if $\mathbf{m}$ is close to ± 1.

Example: Consider the model given by (1) and (2) with $N = 2$, $k = 1$ and $M = 1$.

1) Let $\mathbf{m}_1 = (0.95\ \ 0.9)'$ be the vector to be stored. The matrices T given by (10) can be computed as

$$T = \begin{bmatrix} 2.0325 - 0.6877x_1 & 1.9253 + 0.762x_1 \\ 1.6335 - 0.6877x_2 & 1.5474 + 0.762x_2 \end{bmatrix}$$

where $X = (x_1 x_2)'$. For stability, the inequality (15) yields $x_1^2 + x_2^2 < 47.755$.

2) Let $\mathbf{m}_1 = (0.2\ \ 0.1)'$ be the vector to be stored. The matrices T given by (10) can be computed as:

$$T = \begin{bmatrix} 0.3627 - 0.4472x_1 & 0.1813 + 0.8944x_1 \\ 0.1795 - 0.4472x_2 & 0.0898 + 0.8944x_2 \end{bmatrix}$$

where $X = (x_1 x_2)'$. In this case, the right-hand side of (15) is negative. As argued above, this is due to the fact that the components of $\mathbf{m}_1$ are not close to ± 1. However, straightforward calculations show that for small x_1 and x_2 (e.g., $x_1 = x_2 = \alpha, 0 \le \alpha \le 1$), $\mathbf{m}_1$ is an asymptotically stable equilibrium point. This shows that the bound given by (15) for stability is very conservative. □

Next we consider the system given by (1) and (3); i.e., the nonlinearity is of the hard-limiter type. As before, let $\mathbf{m}_i$, $i = 1, \cdots, M$, be the *binary* vectors to be stored; i.e., $\mathbf{m}_i \in \{-1, 1\}^N$, $i = 1, \cdots, M$. We set $A = [\mathbf{m}_1 \cdots \mathbf{m}_M]$. Similar to (5), for the vectors $\mathbf{m}_i$ to be the equilibrium points of (1), (3), the matrix T must satisfy the following:

$$TA = P \tag{16}$$

where $P \in \mathbf{R}^{N \times M}$, and must satisfy the following requirements:

1) $f(p_{ij}) = f(a_{ij})$, $i = 1, \cdots, N$, $j = 1, \cdots, M$; here f is given by (3), and $p_{ij} = [P]_{ij}$;
2) Row space of A spans the row space of P; i.e., for an arbitrary matrix $K \in \mathbf{R}^{N \times N}$, we have $P = KA$.

Note that the above requirements are satisfied if we particularly choose $P = KA$, with K being a diagonal matrix with strictly positive elements on the diagonal.

To find all connection matrices T that satisfy (16), we apply SVD to A, [see (6)], and following the developments between (6)–(10), we obtain the following characterization of all matrices T that solve the design problem:

$$T = PV_1 D^{-1} U_1' + XU_2'. \tag{17}$$

As before, X is an arbitrary $N \times (N-r)$ matrix; P must satisfy requirements 1) and 2) stated above, arbitrary otherwise.

Remark 1: A particular choice of P and X in (17) is $P = \tau_1 A$ and $X = -\tau_2 U_2$, where $\tau_1 > 0$ and $\tau_2 \in \mathbf{R}$ is arbitrary. This choice, in (17), yields:

$$T = \tau_1 U_1 U_1' - \tau_2 U_2 U_2'. \tag{18}$$

which is the form of T given in [10]. Observe that $P = \tau_1 A$ means that all of the vectors to be stored are eigenvectors of T with a single positive eigenvalue τ_1, see (16). □

Remark 2: A well-known method used to form a T matrix to solve the design problem is the outer-product method, which is given by the following equation (see [7]):

$$T = \sum_{i=1}^{M} \mathbf{m}_i \mathbf{m}_i' - \alpha MI = AA' - \alpha MI \tag{19}$$

where $\alpha = 0$ or $\alpha = 1$, and I is the $N \times N$ identity matrix. This method gives a symmetric T and solves the design problem if the vectors to be stored are mutually *orthogonal*; in which case for $\alpha = 1$ the diagonal elements of T are nullified. Note that if the vectors to be stored are mutually *orthogonal*, by straightforward calculations it can be shown that the matrix T given by (19) is of the form given by (18) with $\tau_1 = N - \alpha M$, $\tau_2 = M$.

In case the vectors are not mutually orthogonal, it is not guaranteed *a priori* that the matrix T given by (19) solves the design problem. In the sequel we give a *sufficient* condition that guarantees that the matrix T given by (19) is a solution to the design problem. Comparing (16) and (19) we see that $P = (AA' - \alpha MI)A$; hence, for T to be a solution to the design problem, requirement 1) given after (16) must be satisfied. Let h_{ij} denote the Hamming distance between the vectors $\mathbf{m}_i$ and $\mathbf{m}_j$, $i = 1, \cdots, M$, $j = 1, \cdots, M$. Noting that $h_{ii} = 0$, we conclude that $[A'A]_{ij} = N - 2h_{ij}$. By straightforward calculations we obtain the following *sufficient* condition that guarantees that requirement 1) is satisfied:

$$\sum_{k=1, k \ne i}^{M} | N - 2h_{ki} | < N - \alpha M \quad i = 1, \cdots, M. \tag{20}$$

Hence, from the above arguments we conclude that if (20) is satisfied then the matrix T given by (19) is a solution to the design problem. Since the Hamming distance between two orthogonal vectors is $N/2$, (20) is readily satisfied for a set of mutually orthogonal vectors. The above analysis suggests that for the outer-product rule to be used as a design method, the vectors to be stored should have pairwise Hamming distances close to $N/2$; that is, they should be nearly orthogonal. □

IV. Design of Continuous-Time Neural Networks

We consider the neural network model given by (4) and (2). Let, as before, $\mathbf{m}_i \in \mathbf{R}^N$, $i = 1, \cdots, M$, denote the vectors to be stored and let $A = [\mathbf{m}_1 \cdots \mathbf{m}_M]$. These vectors are the equilibrium points of the neural network if the matrix T satisfies the following:

$$Tf(A) = \frac{1}{\tau} A. \tag{21}$$

To use the same technique used in Section III, we apply the SVD to $f(A)$ [see (6)]:

$$f(A) = U \Sigma V'. \tag{22}$$

We decompose U, Σ, and V as given by (7) with $r = \operatorname{rank} f(A)$. Following the developments between (8)–(10), we obtain the following characterization of all matrices T that satisfy (21):

$$T = \frac{1}{\tau} AV_1 D^{-1} U_1' + XU_2', \tag{23}$$

where $X \in \mathbf{R}^{N \times (N-r)}$ is an arbitrary matrix.

Although all matrices T that store the given vectors are characterized by (23), the asymptotic stability of these equilibrium points are not guaranteed *a priori*. Here we present a sufficient condition, similar to (15), that guarantees the asymptotic stability of these equilibrium points. The stability analysis used is, as in Section III, based on the linearization of (4) about each equilibrium point.

Let the function $g : \mathbf{R}^N \to \mathbf{R}^N$ be defined as given by (11). We define the matrices G_i as:

$$G_i = \frac{\partial g}{\partial x}(\mathbf{m}_i) \quad i = 1, \cdots, M. \tag{24}$$

Then by using the linearization of (4) about $\mathbf{m}_i$, we conclude that $\mathbf{m}_i$ is an asymptotically stable equilibrium point for (4) if all eigenvalues of the matrix $TG_i - \frac{1}{\tau}I$ are in the open left-half of the complex plane (see [6]). Since the eigenvalues of $TG_i - \frac{1}{\tau}I$ are the eigenvalues of TG_i shifted to the left by $\frac{1}{\tau}$, to guarantee stability, the eigenvalues of TG_i should have real parts less than $\frac{1}{\tau}$.

Let us define Λ^*_{max} as follows:

$$\Lambda^*_{max} = \max\{\lambda_{max}(G_1), \cdots, \lambda_{max}(G_M)\}. \tag{25}$$

Following the developments between (13), (14), we conclude that if X can be chosen as:

$$\|X\|_2 < \frac{1}{\tau\Lambda^*_{max}} - \frac{\sigma_{max}(A)}{\tau\sigma_{min}(A)} \tag{26}$$

then the stability of all the equilibrium points are guaranteed. In this case all eigenvalues of TG_i are confined in a disc of radius $\frac{1}{\tau}$ centered at the origin. Note that for the existence of an X that satisfies (26), the right-hand side of this inequality must be strictly positive. This is always true if $\sigma_{max}(A)\Lambda^*_{max} < \sigma_{min}(A)$, which is always guaranteed if the components of the vectors to be stored are sufficiently large—that is, if the *images* of the components of these vectors under the nonlinear function f are sufficiently close to ± 1. If this latter inequality is satisfied, then from (26) we conclude that the smaller the τ is, the bigger the right-hand side of (26), and hence the larger the degree of freedom in choosing T; see (23) and (26). □

V. Conclusion

In this paper we considered both discrete-time and continuous-time neural networks. The design problem we considered is to find all possible connection matrices that store a given set of vectors into the network; i.e., each given vector becomes an equilibrium point of the network. We obtained a characterization of all possible matrices for both discrete-time and continuous-time cases, [see (10), (17), and (23)]. The relation between the well-known outer-product method and our method is discussed in Remark 2. We also presented sufficient conditions for both the discrete-time and the continuous-time cases that guarantee the asymptotic stability of the equilibrium points. These conditions are satisfied if the components of the vectors to be stored are sufficiently close to ± 1 for the discrete-time case, and if the images of the components of the vectors to be stored under the nonlinear function f are sufficiently close to ± 1 for the continuous-time case.

References

[1] S. Aiyer, M. Niranjan, and F. Fallside, "A theoretical investigation into the performance of the Hopfield model," *IEEE Trans. Neural Networks*, vol. 1, no. 2, pp. 204–216, 1990.

[2] S. R. Das, "On the synthesis of nonlinear continuous neural networks," *IEEE Trans. Syst., Man Cybern.*, vol. 21, no. 2, pp. 413–418, 1991.

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

[4] 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. 33, no. 4, pp. 461–482, 1987.

[5] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore: Johns Hopkins Univ. Press, 1983.

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

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

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

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

[10] A. Michel, J. Si, and G. Yen, "Analysis and synthesis of a class of discrete-time neural networks described on hypercubes," *IEEE Trans. Neural Networks*, vol. 2, no. 1, pp. 32–47, 1991.

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