

On the Convergence Properties of the Hopfield Model

JEHOSHUA BRUCK

The main contribution is showing that the known convergence properties of the Hopfield model can be reduced to a very simple case, for which we have an elementary proof. The convergence properties of the Hopfield model are dependent on the structure of the interconnections matrix W and the method by which the nodes are updated. Three cases are known: (1) convergence to a stable state when operating in a serial mode with symmetric W , (2) convergence to a cycle of length at most 2 when operating in a fully parallel mode with symmetric W , and (3) convergence to a cycle of length 4 when operating in a fully parallel mode with anti-symmetric W . We review the three known results and prove that the fully parallel mode of operation is a special case of the serial mode of operation, for which we present an elementary proof. The elementary proof (one which does not involve the concept of an energy function) follows from the relations between the model and cuts in the graph. We also prove that the three known cases are the only interesting ones by exhibiting exponential lower bounds on the length of the cycles in the other cases.

I. INTRODUCTION

A. What is a (Neural) Network?

The neural network model considered is the one suggested by Hopfield in 1982 [1]. It is a discrete-time system that can be represented by a weighted graph. A weight is attached to each edge of the graph and a threshold value is attached to each node (neuron) of the graph. The order of the network is the number of nodes in the corresponding graph. Let N be a neural network of order n ; then N is uniquely defined by (W, T) where:

- W is an $n \times n$ matrix, with element w_{ij} equal to the weight attached to edge (i, j)
- T is a vector of dimension n , where element t_i 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 denoted by $v_i(t)$. The state of the neural network at time t is the vector $V(t) = (v_1(t), v_2(t), \dots, v_n(t))$.

The state of a node at time $(t + 1)$ is computed by

$$v_i(t + 1) = \text{sgn}(H_i(t)) = \begin{cases} 1 & \text{if } H_i(t) \geq 0 \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

Manuscript received July 17, 1989; revised February 21, 1990.
J. Bruck is with the IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099, USA.
IEEE Log Number 9037383.

where

$$H_i(t) = \sum_{j=1}^n w_{j,i} v_j(t) - t_i.$$

Note that every node in the network is actually a linear threshold (LT) element with the states of the other nodes being its inputs and the threshold being t_i .

The next state of the network, that is, $V(t + 1)$, is computed from the current state by performing the evaluation (1) at a subset of the nodes of the network, to be denoted by S . 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, $|S| = 1$ ($|S|$ denotes the number of nodes in the set S), then we will say that the network is operating in a *serial* mode, and if the computation is performed in all nodes in the same time, that is, $|S| = n$, then we will say that the network is operating in a *fully parallel* mode. All the other cases, that is, $1 < |S| < n$, will be called *parallel* modes of operation. The set S can be chosen at random or according to some deterministic rule.

A state $V(t)$ is called *stable* iff $V(t) = \text{sgn}(WV(t) - T)$, that is, there is no change in the state of the network no matter what the mode of operation is. The set of stable states of a network N is denoted by M_N . A set of distinct states $\{V_1, \dots, V_k\}$ is a cycle of length k if a sequence of evaluations results in the sequence of states: $V_1, \dots, V_k, V_1, \dots$ repeating forever.

B. Three Simple Examples

To make the foregoing definitions clear, we consider three simple examples. The networks considered in these examples consist of two nodes only, like the one in Fig. 1.

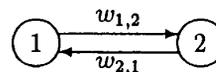


Fig. 1. A network with two nodes.

Example 1: serial operation, symmetric W : Consider the network $N = (W, T)$ with

$$W = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

and T being the 0 vector. It can be verified that when N is operating in a serial mode $M_N = \{(-1, 1), (1, -1)\}$.

Example 2: fully-parallel operation, symmetric W : Consider the network $N = (W, T)$, with

$$W = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix}$$

and T being the 0 vector. It can be verified that when N is operating in a fully parallel mode $M_N = \{(-1, 1), (1, -1)\}$ and that $\{(1, 1), (-1, -1)\}$ is a cycle of length 2.

Example 3: fully-parallel operation, antisymmetric W : Consider the network $N = (W, T)$ with

$$W = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

and T being the 0 vector. It can be verified that when N is operating in a fully parallel mode there are no stable states and the set of states

$$\{(1, 1), (-1, 1), (-1, -1), (1, -1)\}$$

is a cycle of length 4.

One of the fascinating properties of the network model is the fact that these three examples are special cases of a general property—the convergence property. Note that since the state-space of a network is finite, the network will always converge to the stable states/cycles in the state-space.

C. Convergence Properties

The convergence properties are dependent on the structure of W and the method by which the nodes are updated. The three foregoing examples are special cases of the following three known results:

- 1) Convergence to a stable state when operating in a serial mode with symmetric nonnegative diagonal W [1].
- 2) Convergence to a cycle of length at most 2 when operating in a fully-parallel mode with symmetric W [2].
- 3) Convergence to a cycle of length 4 when operating in a fully-parallel mode with antisymmetric W [3].

The main idea in the proof of the convergence properties is to define a so-called *energy function* and to show that this energy function is nondecreasing when the state of the network changes as a result of computation. Since the energy function is bounded from above it follows that it will converge to some value. Our approach here is to get a proof that does not involve the concept of an energy function.

In Section II it is shown that finding the global maximum of the energy function associated with the network operating in a serial mode is *equivalent* to finding the minimum cut in the *undirected* graph associated with the network. We then use this relation to derive an elementary proof (one that does not involve the concept of an energy function) for convergence in the serial mode (see the Appendix for a proof of this result that uses the concept of an energy function). Getting an elementary proof using the relation to cuts in a graph led to the following question:

Is the convergence property unique? Three convergence properties were described in Section I-C. The idea in the proofs of those results is to use the concept of an energy

function. Two different energy functions were used for those three results. The question is whether the three convergence properties are inherently different. The answer is no. In Section III it is shown that the two properties of convergence in a fully parallel mode are special cases of the convergence in a serial mode for which we have an elementary proof. Also we prove that those are the only three interesting cases by proving exponential lower bounds on the length of the cycles in the other cases.

II. THE MODEL AND CUTS IN A GRAPH

The idea in this section is to establish the relation between the method of computation performed by the network to a particular problem in graph theory, namely, the problem of finding a minimum cut (MC) in an undirected graph. In fact, it is shown that finding a global maximum of the energy function associated with a network operating in a serial mode is *equivalent* to find a minimum cut in the undirected graph associated with the network. We use this relation to get a very elementary proof for convergence in a network operating in a serial mode. It also possible to consider directed graphs and show how to program a network to perform a local random search for the MC [4].

A. The Equivalence with the Undirected Case

The neural network model, when operating in a serial mode, is actually performing a local search for a maximum of the energy function denoted by E_1 :

$$E_1(t) = V^T(t)WV(t) - 2V^T(t)T. \quad (2)$$

Theorem 8 (see the Appendix) implies that a network, when operating in a serial mode, will always get to a stable state which corresponds to a local maximum in the energy function E_1 . This property suggests the use of the network as a device for performing a local search algorithm in order to find a local maximal value of the energy function E_1 [5]. The value of E_1 that corresponds to the initial state is improved by performing a sequence of random serial iterations until the network reaches a local maximum. The local search algorithm performed by the neural network model is imposed by the way the network is operating. Consider a network N operating in a serial mode, and let L_N denote the local search algorithm performed by the network N .

Algorithm L_N for $\max(E_1(V))$ is:

- 1) Start with a random assignment $V \in \{-1, 1\}^n$.
- 2) Choose a node $i \in \{1 \cdots n\}$ at random.
- 3) Try to improve E_1 by performing the evaluation

$$v_i = \text{sgn} \left(\sum_{j=1}^n w_{j,i} v_j - t_i \right).$$

- 4) Go to step 2.

The class of optimization problems that can be represented by quadratic functions is very rich [6]. One problem, which is not only representable by a quadratic function but actually is equivalent, is the problem of finding an MC in a graph [6]–[8]. To make the above statements clear, let us start by defining the term “cut in a graph.”

Definition: Let $G = (V, E)$ be a weighted and undirected graph, with W being an $n \times n$ symmetric matrix of weights of the edges of G . Let V_1 be a subset of V , and let $V_{-1} = V - V_1$. The set of edges incident at one node in V_1 and at one

node in V_{-1} is called a *cut* of the graph G . A *minimum cut* in a graph is a cut for which the sum of the corresponding edge weights is minimal over all V_1 .

Theorem 1: [6], [8] Let $G = (V, E)$ be a weighted and undirected graph, with W being the matrix of its edge weights. Then the MC problem in G is equivalent to $\max Q_G(X)$, where $X \in \{-1, 1\}^n$, and:

$$Q_G(X) \stackrel{\text{def}}{=} \sum_{i=1}^n \sum_{j=1}^n w_{i,j} x_i x_j.$$

Proof: Assign a variable x_i to every node $i \in V$. Let W^{++} denote the sum of the weights of edges in G with both end points equal to 1, and let W^{--} and W^{+-} denote the corresponding sums of the other two cases. Thus,

$$Q_G = 2(W^{++} + W^{--} - W^{+-})$$

which also can be written as

$$\begin{aligned} Q_G &= 2(W^{++} + W^{--} + W^{+-}) - 4W^{+-} \\ &= \sum_{i=1}^n \sum_{j=1}^n w_{i,j} - 4W^{+-}. \end{aligned} \quad (3)$$

The first term in (3) above is constant (equals the sum of weights of edges in G); hence, maximization of Q_G is equivalent to minimization of W^{+-} is actually a weight of a cut in G with V_1 being the set of nodes in G that correspond to variables that are equal to 1. \square

The above theorem can be applied to get the equivalence with the energy associated with the network.

Theorem 2: Let $N = (W, T)$ be a network with W being an $n \times n$ symmetric zero diagonal matrix. Let G be a weighted graph with $(n + 1)$ nodes, with its weight matrix W_G being

$$W_G = \begin{pmatrix} W & T \\ T^T & 0 \end{pmatrix}.$$

The problem of finding a state V in N for which E_1 is a global maximum is equivalent to the MC problem in the corresponding graph G .

Proof: Note that the graph G is built out of N by adding one node to N and connecting it to the other n nodes, with the edge connected to node i having a weight t_i (the corresponding threshold). Clearly, if the state of the added node is constrained to -1 , then for all $X \in \{-1, 1\}^n$

$$Q_G(X, -1) = E_1(X).$$

Hence, the equivalence follows from Theorem 1. Note that the state of node $(n + 1)$ need not be constrained to -1 . There is a symmetry in the cut; that is $Q_G(X) = Q_G(-X)$ for all $X \in \{-1, 1\}^{n+1}$. Thus, if a minimum cut is achieved with the state of node $(n + 1)$ being 1, then a minimum is also achieved by the cut obtained by interchanging V_1 and V_{-1} (resulting in $x_{n+1} = -1$). \square

B. A Simple Proof for Convergence

The relation between neural networks and the MC problem leads to the following nice interpretation of the algorithm L_N performed by the model.

Algorithm L_N for the MC problem is:

- 1) Start with a random cut.
- 2) Choose a node k at random.
- 3) Compare the sum of weights of the edges which belong to the cut and incident at node k with the

sum of weights of the other edges which are incident at node k . Move node k to the side of the cut which will result in a decrease in the weight of the cut. Ties (the case of equality) are broken by placing node k in V_1 .

- 4) Go to step 2.

Hence, we have an elementary proof for convergence:

Theorem 3: Let $N = (W, T)$ be a network with $T = 0$ and W be a symmetric zero-diagonal matrix. If N is operating in a serial mode then it will always converge to a stable state.

Proof: By the foregoing derivation we can consider the operation of N as the running of algorithm L_N for the minimum cut in N . In each iteration the value of the cut is non-increasing (ties are broken as described above); thus, the algorithm will always stop resulting in a cut whose weight is a local minimum. \square

Clearly, the proof above is for a special case of a network (that is why it is simple). In Section III we show that *all* the other general cases can be reduced to the foregoing simple case.

III. A UNIFIED APPROACH TO CONVERGENCE

Convergence to a stable state/cycle of a certain length is one of the most important properties of the neural network model. The convergence properties are dependent on the structure of the interconnection matrix W and the method by which the nodes are updated. The three known cases are mentioned in the Section I-B. Those results were proved by using the concept of an energy function. In this section we answer the three following questions: (i) Is the convergence property unique? (ii) Is there an elementary proof for convergence (one that does not involve the concept of an energy function)? (iii) Are there any interesting cases besides the three known cases?

The answer to the first question is yes; in fact we prove that convergence in a fully parallel mode is a special case of convergence in a serial mode. In Section II we presented a proof (see Theorem 3) for convergence in a very simple network based on the relations between networks and cuts in a graph. In this section we show that all the other cases are special cases of this simple network; hence, we have an elementary proof for the convergence properties (a positive answer to (ii)). We also consider other cases and exhibit exponentially (in the number of nodes) long cycles in networks that are not of the three known cases; hence we have a negative answer to (iii).

A. Convergence Theorems

One of the most important properties of the model is the fact that in certain cases it always converges, as summarized by the following theorem. Notice that these three cases correspond to the three simple examples in the Section I-B.

Theorem 4: Let $N = (W, T)$ be a neural network, then:

- 1) Assume N is operating in a serial mode and W is a symmetric matrix with the elements of the diagonal being nonnegative. Then the network will always converge to a stable state, that is, there are no cycles in the state space [1].
- 2) Assume N is operating in a fully parallel mode and W is a symmetric matrix. Then the network will always converge to a stable state or to a cycle of

length 2, that is, the cycles in the state space are of length ≤ 2 [2].

- 3) Assume N is operating in a fully parallel mode and W is an antisymmetric matrix with zero diagonal and let $T = 0$. Then the network will always converge to a cycle of length 4 [3].

The main idea in the proof of the three parts of the theorem 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 the energy will converge to some value. An important note is that originally the energy function was defined by others such that it is nonincreasing [1]-[3]; we changed it to be nondecreasing such that the value of the energy will comply with some known graph problems (for example, MC, see Section II). The second step in the proof is to show that constant energy implies in the first case a stable state, in the second a cycle of length ≤ 2 , and in the third a cycle of length 4 (see the Appendix for a proof of part 1 of the theorem that involves the concept of an energy function). Two different energy functions were defined:

$$\begin{aligned} E_1(t) &= V^T(t)WV(t) - (V(t) + V(t))^T T \\ E_2(t) &= V^T(t)WV(t-1) - (V(t) + V(t-1))^T T. \end{aligned} \quad (4)$$

The energy function $E_1(t)$ was used to prove the first part of the theorem and $E_2(t)$ was used to prove the second and third parts of the theorem. In the next section we reveal the relation between the three cases.

B. A Unified Theorem via Reductions

In this section we prove that the three cases of Theorem 4 are special cases of a network operating in a serial mode with W being a symmetric zero-diagonal matrix. Notice that the reduction is in the sense that it is possible to derive the state of one network given the state of the other network. The first lemma presents the two reductions associated with W being a symmetric matrix [4].

Lemma 1: Let $N = (W, T)$ be a neural network where W is a symmetric matrix. Let $\hat{N} = (\hat{W}, \hat{T})$ be obtained from N as follows: \hat{N} is a bipartite graph, with

$$\hat{W} = \begin{pmatrix} 0 & W \\ W & 0 \end{pmatrix}$$

and

$$\hat{T} = \begin{pmatrix} T \\ T \end{pmatrix}.$$

- (a) For any serial mode of operation in N there exists an equivalent serial mode of operation in \hat{N} , provided W has a nonnegative diagonal.
 (b) There exists a serial mode of operation in \hat{N} which is equivalent to a fully parallel mode of operation in N .

Proof: The new network \hat{N} is a bipartite graph with $2n$ nodes. The set of nodes of \hat{N} can be subdivided into two sets: let P_1 and P_2 denote the set of the first and the last n nodes, respectively. Clearly, no two nodes of P_1 (and also P_2) are connected by an edge; that is, both P_1 and P_2 are independent sets of nodes in \hat{N} . Another observation is that P_1 and P_2 are symmetric in the sense that a node $i \in P_1$ has an edge set similar to that of a node $(i + n) \in P_2$.

Proof of (a): Let V_0 be an initial state of N , and let (i_1, i_2, \dots) be the order by which the states of the nodes are evaluated in a serial mode in N . We will show that starting from the initial state (V_0, V_0) in \hat{N} (the state of both P_1 and P_2 is V_0) and using the order $(i_1, (i_1 + n), i_2, (i_2 + n), \dots)$ for the evaluation of states will result in:

- 1) The state of P_1 will be equal to the state of P_2 in \hat{N} after an arbitrary even number of evaluations.
- 2) The state of N at time k is equal to the state of P_1 at time $2k$, for an arbitrary k .

The proof of (1) is by induction. Given that at some arbitrary time k the state of P_1 is equal to the state of P_2 , it will be shown that after performing the evaluation at node i and then at node $(n + i)$ the states of P_1 and P_2 remain equal.

There are two cases:

- If the state of node i does not change as a result of evaluation, then by the symmetry of \hat{N} there will be no change in the state of node $(n + i)$.
- If there is a change in the state of node i , then because $\hat{W}_{i, n+i}$ is nonnegative it follows that there will be a change in the state of node $(n + i)$ (the proof is straightforward and won't be presented).

The proof of (2) follows from (1): by (1) the state of P_1 is equal to the state of P_2 right before the evaluation at a node of P_1 . The proof is by induction: assume that the current state of N is the same as the state of P_1 in \hat{N} . Then an evaluation performed at a node $i \in P_1$ will have the same result as an evaluation performed at node $i \in N$. \square

Proof of (b): Let's assume as in part (a) that \hat{N} has the initial state (V_0, V_0) . Clearly, performing the evaluation at all nodes belonging to P_1 (in parallel) and then at all nodes belonging to P_2 , and continuing with this alternating order is equivalent to a fully parallel mode of operation in N . The equivalence is in the sense that the state of N is equal to the state of the subset of nodes (either P_1 or P_2) of \hat{N} at which the last evaluation was performed. A key observation is that P_1 and P_2 are independent sets of nodes, and a parallel evaluation at an independent set of nodes is equivalent to a serial evaluation of all the nodes in the set. Thus, the fully parallel mode of operation in N is equivalent to a serial mode of operation in \hat{N} . \square

The second lemma presents the reduction associated with W being an antisymmetric matrix.

Lemma 2: Let $N = (W, T)$ be a neural network where W is an antisymmetric matrix with zero diagonal and $T = 0$. Assume that WV has no zero for all $V \in \{1, -1\}^n$. Let $\hat{N} = (\hat{W}, \hat{T})$ be obtained from N as follows: \hat{N} is a bipartite graph, with

$$\hat{W} = \begin{pmatrix} 0 & 0 & 0 & -W \\ 0 & 0 & W & 0 \\ 0 & -W & 0 & 0 \\ W & 0 & 0 & 0 \end{pmatrix}$$

and $\hat{T} = 0$. There exists a serial mode of operation in \hat{N} which is equivalent to a fully parallel mode of operation in N .

Proof: The new network \hat{N} is a bipartite graph with $4n$ nodes. The set of nodes of \hat{N} can be subdivided into four sets, to be denoted by $P_1, P_2, P_3,$ and P_4 , that correspond to the first, second, third, and fourth sets of n nodes of \hat{N} , respectively. Note that P_1 is connected only to P_4 and that

P_2 is connected only to P_3 . Another observation is that \tilde{W} is a symmetric matrix. This follows from the assumption that $W^T = -W$. We consider fully parallel iterations in the network N and denote that state of N after k iterations by V_k . In the network \tilde{N} we consider parallel iterations at the sets P_i (which are in fact serial iterations), and denote the state of \tilde{N} by a vector which is a concatenation of the states of the P_i s.

Let us assume that the initial state of N is V_0 . We show that if the initial state of \tilde{N} is $(-V_0, V_0, V_0, -V_0)$ and the order of evaluation is $P_1, P_2, P_3, P_4, P_1, P_2, \dots$, then the network \tilde{N} simulates the network N . Note that after an even number of iterations in \tilde{N} , the state of P_1 is the complement of the state of P_2 and the state of P_3 is the complement of the state of P_4 .

Now we claim that: (i) After $8k$ iterations at \tilde{N} the state of P_3 is equal to the state of N after $4k$ evaluations and the state of P_1 is equal to the state of N after $4k - 1$ evaluations. (ii) After $8k + 4$ iterations the state of P_4 is equal to the state of N after $4k + 2$ evaluations and the state of P_2 is equal to the state of N after $4k + 1$ evaluations. The proof of those two claims is by induction on k . Clearly, after 4 iterations in \tilde{N} we have that the state of the network is $(-V_1, V_1, -V_2, V_2)$, which establishes the basis of the induction (one can consider the next 4 iterations and get the basis for (ii)). We assume that (i) and (ii) are true for k and prove (i) and (ii) for $k + 1$. Here we present only the proof of (i). By the assumption, after $8k + 4$ iterations the state of \tilde{N} is $(-V_{4k+1}, V_{4k+1}, -V_{4k+2}, V_{4k+2})$. Hence, after $8(k + 1)$ iterations the state of \tilde{N} is $(V_{4k+3}, -V_{4k+3}, V_{4(k+1)}, -V_{4(k+1)})$, which established (i).

Note that evaluation at a set $P_i, 1 \leq i \leq 4$, is equivalent to a serial evaluation of the nodes in the set, since the sets are independent sets of nodes. Hence, the network \tilde{N} operating in a serial mode can simulate the network N operating in a fully parallel mode. \square

Using the transformations suggested by the above lemmas we show that the three known convergence properties are special cases of convergence in a network operating in a serial mode with W being a symmetric zero-diagonal matrix.

Theorem 5: Let $N = (W, T)$ be a neural network. Given (1), then (2), (3), and (4) below hold.

- 1) If N is operating in a serial mode and W is a symmetric matrix with zero diagonal, then the network will always converge to a stable state.
- 2) If N is operating in a serial mode and W is a symmetric matrix with nonnegative elements on the diagonal, then the network will always converge to a stable state.
- 3) If N is operating in a fully parallel mode then, for an arbitrary symmetric matrix W , 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 .
- 4) If N is operating in a fully parallel mode then, for an antisymmetric matrix W with zero diagonal, with $T = 0$, the network will always converge to a cycle of length 4.

Proof: The proof is based on Lemma 1 and Lemma 2.

(2) is implied by (1): By Lemma 1 part (a), every network with nonnegative diagonal symmetric matrix W which is operating in a serial mode can be transformed to an equiv-

alent network to be denoted by \hat{N} , which is operating in a serial mode with \hat{W} being a symmetric zero-diagonal symmetric matrix. \hat{N} will converge to a stable state (by (1)); hence, N will also converge to a stable state which will be equal to the state of P_1 . Note that trivially (1) is implied by (2).

(3) is implied by (1): By Lemma 1 part (b), every network operating in a fully parallel mode can be transformed to an equivalent network to be denoted by \hat{N} , which is operating in a serial mode with \hat{W} , being a symmetric zero-diagonal matrix. \hat{N} will converge to a stable state (by (1)). When \hat{N} reaches a stable state there are two cases:

1. The state of P_1 is equal to the state of P_2 ; in this case N will converge to a stable state which is equal to the state of P_1 .
2. The states of P_1 and P_2 are distinct; in this case N will oscillate between the two states defined by P_1 and P_2 , that is, N will converge to a cycle of length 2.

(4) is implied by (1): By Lemma 2 every network operating in a fully parallel mode can be transformed to an equivalent network, to be denoted by \hat{N} , which is operating in a serial mode, with \hat{W} being a symmetric zero-diagonal matrix. \hat{N} will converge to a stable state (by (1)). We denote this stable state by (U_1, U_2, U_3, U_4) with U_i corresponding to the state of P_i . We claim that the U_i are distinct, hence, the stable state in \hat{N} corresponds to a cycle of length 4 in N .

To prove that the U_i are distinct observe that, by Lemma 2, $U_1 = -U_2$ and $U_3 = -U_4$. Also, in a stable state $U_1 = \text{sgn}(WU_4)$ and $U_4 = \text{sgn}(-WU_1)$. Assume that $U_1 = U_4$; then $\text{sgn}(WU_1) = \text{sgn}(-WU_1)$. This is a contradiction. Also, when we assume that $U_1 = -U_4$, we reach a contradiction. Hence at a stable state the U_i are distinct. From Lemma 2 it follows that in the network N we have a cycle of length 4: U_2, U_4, U_1, U_3 . \square

An Elementary Proof for Convergence: To show that we have an elementary proof for all the cases, we have to reduce case 1 in Theorem 5 to the simple case considered in Theorem 3. Namely, we have to show that, the case where the network is operating in a serial mode with W a zero-diagonal symmetric matrix and T is an arbitrary vector, can be reduced to that in which $T = 0$. This reduction follows from Theorem 2.

To summarize, we showed that the three known cases of convergence can be reduced to a very simple case: a network operating in a serial mode with W a symmetric zero-diagonal matrix and $T = 0$. For this special case we have an elementary proof that uses the equivalence with cuts in a graph (see Section II). Next we show that, indeed, the three known cases are the only interesting ones.

C. Big Cycles

In the previous section we considered the convergence properties in three cases that can be characterized by the mode of operation and the structure of W : (i) serial mode of operation, symmetric W , (ii) fully parallel mode of operation, symmetric W , and (iii) fully parallel mode of operation, antisymmetric W . Using this characterization, there are three more cases that can be considered: (iv) serial mode of operation, antisymmetric W , (v) serial mode of operation, arbitrary W and (vi) fully parallel mode of operation, arbi-

trary W . In this section we prove that cases (iv)–(vi) are not interesting by showing that networks of these types can have exponentially (in the number of nodes) long cycles in their state space. First we show that cases (iv) and (v) can have an exponentially long cycle by considering a network with antisymmetric W .

Theorem 6: Let $n \geq 2$ be an even integer. There exists a network $N = (W, T)$ of order n with W antisymmetric which, when operating in a serial mode, has a cycle of length 2^n .

Proof: Consider the network $\hat{N} = (\hat{W}, \hat{T})$ defined by

$$\hat{W} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

and $\hat{T} = (0, 0)$. This is the same network as the one in Example 3 in Section I-B. When we consider serial mode of operation in \hat{N} we find that there is a cycle of length 4: $\{(1, 1), (-1, 1), (-1, -1), (1, -1)\}$. This cycle corresponds to the following order of evaluation: 1, 2, 1, 2 \dots .

To get the result we construct a network of order $2n$, to be denoted by N , simply taking n networks like \hat{N} . In N we can generate a cycle of length 2^{2n} by going through all the possible states. The idea is to consider the state of every one of the n subnetworks as a symbol over $GF(4)$ and to go through the possible states lexicographically. \square

Next we show that there is an exponentially long cycle also in case (vi):

Theorem 7: Let n be a positive integer. There exists a network $N = (W, T)$ of order $3n$, which when operating in a fully parallel mode has a cycle of length 2^n .

Proof: The idea in the proof is to construct a linear shift register [9] using linear threshold elements. A linear shift register device is simply a shift register in which the input of a cell is the output of the previous cell. The input to the first cell is a sum mod 2 of a certain subset of the cells. There is a way to select the subset of cells that sum up to be the input to the first cell in such a way that the shift register will go through all the possible states (2 to the number of cells). For more details on this subject see [9]. To construct a linear shift register using linear threshold elements we need to implement two basic operations: (i) IDENTITY—to implement the function of a single cell in a shift register, (ii) XOR—to implement the sum mod 2. Clearly, only the XOR is a problem. But XOR can be implemented by a depth 2 circuit of linear threshold elements (see [10]). For XOR of n variables we need n linear threshold elements. Since we have a depth 2 circuit we need to introduce a delay between any two cells in the shift register. For that we need n more elements. To summarize, we can implement a linear shift register device with n cells using $3n$ linear threshold elements. Hence, for every n , we have a network of linear threshold elements of order $3n$ which, when operating in a fully parallel mode, goes through a cycle of length 2^n . \square

IV. CONCLUDING REMARKS

We have presented a unified approach for proving convergence in the Hopfield model. The idea in our approach is to reduce all the known cases to a very simple case of convergence for which the proof is elementary. We also proved that those cases are the only interesting cases by proving exponential lower bounds on the size of cycles in the other cases.

V. APPENDIX: PROOF OF CONVERGENCE USING THE ENERGY FUNCTION

We consider the first convergence property (serial mode), and prove it using the concept of the energy function.

Theorem 8: Let $N = (W, T)$ be a neural network operating in a serial mode. Let W be a symmetric matrix with non-negative diagonal. Then the network will always converge to a stable state, i.e., there are no cycles in the state-space [1].

Proof: The energy function is defined as follows:

$$E_1(t) = V^T(t)WV(t) - 2V(t)^T T \quad (5)$$

where a superscript T indicates a matrix transpose. Let $\Delta E = E_1(t+1) - E_1(t)$ be the difference between the energies associated with two consecutive states, and let ΔV_k denote the difference between the next state and the current state of node k at some arbitrary time t . H_k is defined in (1). From (1) it follows that

$$\Delta V_k = \begin{cases} 0 & \text{if } V_k(t) = \text{sgn}(H_k(t)) \\ -2 & \text{if } V_k(t) = 1 \text{ and } \text{sgn}(H_k(t)) = -1 \\ 2 & \text{if } V_k(t) = -1 \text{ and } \text{sgn}(H_k(t)) = 1. \end{cases} \quad (6)$$

By the assumption (serial mode of operation) the computation is performed only at a single node at any given time. Suppose this computation is performed at an arbitrary node k ; then the energy difference resulting from this computation is

$$\Delta E = \Delta V_k \left(\sum_{i=1}^n W_{k,i} V_i + \sum_{i=1}^n W_{i,k} V_i \right) + W_{k,k} \Delta V_k^2 - 2 \Delta V_k T_k. \quad (7)$$

From the symmetry of W and the definition of H_k it follows that

$$\Delta E = 2 \Delta V_k H_k + W_{k,k} \Delta V_k^2. \quad (8)$$

Hence, since $\Delta V_k H_k \geq 0$ and $W_{k,k} \geq 0$ it follows that $\Delta E \geq 0$ for every k . Since E_1 is bounded from above, the value of the energy will converge.

The second step in the proof is to show that convergence of the value of the energy implies convergence to a stable state. The following two simple facts are helpful for this step:

- 1) If $\Delta V_k = 0$ then $\Delta E = 0$.
- 2) If $\Delta V_k \neq 0$ then $\Delta E = 0$ only if the change in V_k is from -1 to 1 , with $H_k = 0$.

Hence, once the energy in the network has converged, it is clear from the preceding facts that the network will reach a stable state after at most n^2 time intervals. \square

REFERENCES

- [1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci. USA*, vol. 79, pp. 2554–2558, 1982.
- [2] 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.
- [3] E. Goles, "Antisymmetrical neural networks," *Discrete Appl. Math.*, vol. 13, pp. 97–100, 1986.
- [4] J. Bruck and J. W. Goodman, "A generalized convergence theorem for neural networks," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1089–1092, Sept. 1988.

- [5] J. J. Hopfield and D. W. Tank, "Neural computations of decisions in optimization problems," *Biolog. Cybern.*, vol. 52, pp. 141-152, 1985.
- [6] P. L. Hammer and S. Rudeanu, *Boolean Methods in Operations Research*. New York: Springer-Verlag, 1968.
- [7] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [8] J. C. Picard and H. D. Ratliff, "Minimum cuts and related problems," *Networks*, vol. 5, pp. 357-370, 1974.
- [9] S. W. Golomb, *Shift Register Sequences*. CA: Aegean Park Press, 1982.
- [10] J. Bruck, "Harmonic analysis of polynomial threshold functions," *SIAM J. Dis. Math.*, vol. 3, no. 2, pp. 168-177, May 1990.



Jehoshua Bruck was born in Haifa, Israel, on April 19, 1956. He received the B.Sc. and M.Sc. degrees in electrical engineering from the Technion, Israel Institute of Technology, in 1982 and 1985, respectively, and the Ph.D. degree in electrical engineering from Stanford University in 1989.

From 1982 to 1985 he was with the IBM Haifa Scientific Center, Israel. In March, 1989, he joined the IBM Research Division at the Almaden Research Center, San Jose, CA, where he is presently a Research Staff Member.

Dr. Bruck's research interests include error-correcting codes, fault-tolerant computing, parallel computing, and neural networks.