Minimum-Seeking Properties of Analog Neural Networks with Multilinear Objective Functions

M. Vidyasagar, *Fellow, IEEE*

***Abstract*—In this paper, we study the problem of minimzing a multilinear objective function over the discrete set {0, 1}^{n}. This is an extension of an earlier work addressed to the problem of minimizing a quadratic function over {0, 1}^{n}. A gradient-type neural network is proposed to perform the optimization. A novel feature of the network is the introduction of a so-called bias vector. The network is operated in the high-gain region of the sigmoidal nonlinearities. The following comprehensive theorem is proved: For all sufficiently small bias vectors except those belonging to a set of measure zero, for all sufficiently large sigmoidal gains, for all initial conditions except those belonging to a nowhere dense set, the state of the network converges to a local minimum of the objective function. This is a considerable generalization of earlier results for quadratic objective functions. Moreover, the proofs here are completely rigorous. The neural network-based approach to optimization is briefly compared to the so-called interior-point methods of nonlinear programming, as exemplified by Karmarkar's algorithm. Some problems for future research are suggested.**

I. Introduction

Most of the current research into feedback neural networks is concentrated on networks with linear interconnections and quadratic energy functions. In his trend-setting paper [14], Hopfield considers two-state networks described by

$$x_i(t+1) = \text{sign}\left[\sum_{j=1}^{n} w_{ij}x_j(t) + \theta_i\right], \qquad i = 1,\cdots,n \tag{1.1}$$

where n is the number of neurons, $x_i(t) \in \{-1, 1\}$ is the state of neuron i at time t, w_{ij} is the weight of the interconnection from neuron j to neuron i, and $-\theta_i$ is the firing threshold of neuron i. Hopfield [14] defines the energy of the network as

$$E = -\tfrac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_ix_j - \sum_{i=1}^{n} x_i\theta_i \tag{1.2}$$

and proves the following the property: Suppose $w_{ji} = w_{ij}$ for all i, j (symmetric interactions) and $w_{ii} = 0$ for all i (no self-interactions). Finally, suppose the neural states are updated asynchronously as follows: At each (discrete) instant of time t, select an integer $i \in \{1,\cdots,n\}$ in sequence, compute $x_i(t+1)$ in accordance with (1.1), but leave $x_j(t)$ unchanged for all $j \neq i$. In this mode of operation, it is true that

$$E[\boldsymbol{x}(t+1)] \leq E[\boldsymbol{x}(t)] \tag{1.3}$$

where $\boldsymbol{x} = [x_1 \cdots x_n]^t$. Thus, in an asynchronous mode of operation, the neural network will eventually reach a fixed point of the network, that is, a vector $\boldsymbol{x}_0$ with the property that

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 \Rightarrow \boldsymbol{x}(t+1) = \boldsymbol{x}_0 \tag{1.4}$$

irrespective of which neuron is updated at time t. Hence, if it is desired to optimize a quadratic objective function over the discrete set $\{-1, 1\}^n$, one can use a neural network of form (1.1). Such a network will, within a finite number of time steps, reach a local minimum of E. Some practical problems that can be translated as the minimization of a quadratic objective function include the Traveling Salesman Problem [16] and linear programming [26].

While the above results are very impressive, the asynchronous mode of operation is vulnerable to the criticism that the convergence to a local minimum will be slow, because out of the n neurons, only one neuron changes its state at a time. If one thinks of the state space of the neural network as the set of 2^n corners of the hypercube $[-1, 1]^n$, then asynchronous updating changes the state of the network only to an adjacent state at each time step. Indeed, the results of [25] can be interpreted to mean that, in the worst case, a discrete-state Hopfield network can take an exponential number of steps to reach even a local minimum. It is of course possible to operate the network in a synchronous mode; in other words, at time t, the state of each neuron is updated according to (1.1). If this is done, however, then (1.3) is no longer true, and the network may not converge to a fixed point. In fact, it is shown in [11] that in the synchronous mode, the network can go into a limit cycle of length two.

In an attempt to speed up the operation of the network, Hopfield [15] considers analog neural networks where the constituent neurons have graded responses. Such networks are described by

$$C_i\dot{u}_i = -\frac{1}{R_i}u_i + \sum_{j=1}^{n} w_{ij}v_j + \theta_i, \quad v_i = g_i(\lambda u_i),$$
$$i = 1,\cdots,n \tag{1.5}$$

where n is the number of neurons; v_i is the neural current and u_i is the neural voltage, θ_i is the external current input to the ith neuron, C_i is the membrane capacitance and R_i is the neural resistance, λ is a scaling parameter, and w_{ij} is

Manuscript received April 9, 1993; revised November 30, 1994. Recommended by Past Associate Editor, W. S. Wong.

The author is with the Centre for Artificial Intelligence and Robotics, Raj Bhavan Circle, High Grounds, Bangalore 560 001, India.

IEEE Log Number 9412440.

0018-9286/95$04.00 © 1995 IEEE

the weight of the interconnection from neuron j to neuron i. The function g_i: $\Re \rightarrow (0,1)$ is continuously differentiable, strictly increasing, $g_i(x) \rightarrow 1$ as $x \rightarrow \infty$, and $g_i(x) \rightarrow 0$ as $x \rightarrow -\infty$. Note that as the scaling parameter $\lambda \rightarrow \infty$, the function $x \mapsto g_i(\lambda x)$ approaches the "saturation" function sat (x), which equals zero if $x < 0$ and equals one if $x > 0$. In [15], Hopfield assumes that $w_{ij} = w_{ji}$ for all i, j and that $w_{ii} = 0$ for all i. He then proposes the energy function

$$E_c(\boldsymbol{v}) = \sum_{i=1}^{n} \left[\frac{1}{\lambda R_i} \int_0^{v_i} g_i^{-1}(v)\, dv - \theta_i v_i - \frac{1}{2} \sum_{j=1}^{n} w_{ij} v_i v_j \right] \tag{1.6}$$

and shows that

$$\frac{d}{dt} E_c[\boldsymbol{v}(t)] \leq 0 \tag{1.7}$$

along trajectories of system (1.5). Hirsch [12] shows, using relationship (1.7), that system (1.5) is totally stable; that is, every solution trajectory approaches an equilibrium. Salam *et al.* [23] fix a small technical flaw in the reasoning of [12] by establishing that no solution trajectory of (1.5) escapes to infinity (in the $\boldsymbol{u}$ space).

The results of [12] and [23] combined show that every solution trajectory of (1.5), irrespective of the initial condition, approaches an equilibrium. Thus it becomes important to analyze the location and stability status of the equilibria of system (1.5). Such an analysis is carried out in [27] for a very general class of neural networks in the high gain limit, i.e., as $\lambda \rightarrow \infty$. The analysis of [27] applies also to the neural network (1.5). It shows that, if $w_{ii} = 0$ for all $i, w_{ij} = w_{ji}$ for all i, j, and certain additional assumptions hold, then "almost all" solution trajectories of the network approach a corner of the unit hypercube $[0,1]^n$. Note that, as $\lambda \rightarrow \infty$, the energy function E_c of (1.6) approaches the energy function E of (1.2). Hence, by operating the analog network (1.5) near the high gain limit, it is possible to minimize E over the discrete set $\{0, 1\}^n$. Note that the operation of the analog network (1.5) is "synchronous" in the sense that the states of all neurons are changing at the same time. It can therefore be argued that the analog implementation combines the speed of synchronous operation with the minimum-seeking property of asynchronous operation.

Thus, in the case where the objective function to be minimized is a quadratic, the situation is quite well understood. There are situations, however, in which it is more natural to use an objective function which is a polynomial of degree three or higher. One such example is given in [21], wherein the problem of checking whether of not there exists a truth assignment on a set of Boolean variables that makes each of a set of formulas true (commonly known as the "satisfiability problem" and the "original" NP-complete problem) is formulated as a minimization problem over the set $\{0,1\}^n$ where n is the number of literals and the degree of the objective function is the length of the longest clause in the set of formulas. Another example is given in [5], wherein the problem of algebraic block-decoding is formulated as that of maximizing a polynomial over $\{-1, 1\}^n$, where the number of neurons n equals the length of the encoded words and the degree of the objective function is equal to the number of information bits.

The objective of the present paper is to generalize the system description (1.5) to the case of an objective function that is not necessarily quadratic. A natural way to do this is to replace the term $\Sigma_{j=1}^n w_{ij} v_j + \theta_i$ in (1.5) by the negative gradient of the objective function. This, plus the introduction of a so-called "bias" vector, is the essence of the approach proposed here.

Next, the organization of the paper is described. The analysis of the neural networks can be divided into two distinct parts. The first part of the analysis applies to any neural network described by a set of equations of the form

$$\dot{u}_i = -\frac{u_i}{\alpha_i} - f_i(\boldsymbol{x}) + b_i, x_i = g_i(\lambda u_i), \qquad i = 1, \cdots n \tag{1.8}$$

where $\boldsymbol{f}$: $[0,1]^n \rightarrow \Re^n$ is continuously differentiable. In other words, it is not necessary that $\boldsymbol{f}$ be the gradient of a scalar-valued function. Even in this quite general case, it is possible to carry out a fairly comprehensive analysis of the equilibria of the network. This is done in Section II. The results of this section are similar to those in [27], except that the proofs are completely rigorous. This is in contrast to [27], where there is a certain amount of "hand waving."

The second part of the analysis applies to the specific class of neural networks described by

$$\dot{u}_i = -\frac{u_i}{\alpha_i} - [\nabla E(\boldsymbol{x})]_i + b_i, x_i = g_i(\lambda u_i), \qquad i = 1, \cdots, n \tag{1.9}$$

where E is the objective function to be minimized over $\{0,1\}^n$. It is assumed that the function E satisfies the condition

$$\frac{\partial^2 E}{\partial x_i^2} = 0 \text{ for all } i. \tag{1.10}$$

It is easy to see that the above condition is equivalent to requiring $E(x_1, \cdots, x_n)$ to be an affine function of x_i when all other $x_j, j \neq i$ are fixed, i.e., requiring that E be a multilinear polynomial. This condition assures that the right side of (1.9) is independent of x_i and can be thought of as a nonlinear version of the "no self-interactions" assumption made by Hopfield in [15]. In particular, if E is of the form (1.2), then E satisfies (1.10) if and only if $w_{ii} = 0$ for all i. Thus the class of objective functions studied here includes those of Hopfield as a special case.

The class of networks described by (1.9) is analyzed in Section III. The key to the analysis is the introduction of the so-called "bias" vector $\boldsymbol{b}$ into the dynamics (1.9). The main conclusions of the analysis are that, for "almost all" sufficiently small bias vectors $\boldsymbol{b}$, for "almost all" initial conditions $\boldsymbol{x}(0)$, and all sufficiently high values of the sigmoidal gain λ, the solution trajectory of (1.9) approaches a corner of the hypercube $[0,1]^n$ and that this corner is a local minimum of E over $\{0,1\}^n$.

In Section IV, a few simple examples are presented to show that the introduction of the bias vector is really necessary, in the sense that the theorems are false without these features.

In Section V, the neural network-based approach analyzed here is briefly compared to the so-called interior-point methods for nonlinear programming, as exemplified by Karmarkar's algorithm [18]. Special attention is paid to the so-called affine scaling vector field and another approach suggested by Faybusovich [7].

Section VI contains the concluding remarks. One of the conclusions is that there is a need to develop a suitable theory of computational complexity for analog computation. The paper is rounded out by two brief appendixes. The first summarizes optimality conditions in the case where the objective function is multilinear, while the second presents an open problem related to the strict local minima of multilinear polynomials.

Now a brief word about the notation. For the most part, upper-case bold-face letters indicate matrices, while lower-case bold-face letters indicate vectors. The symbol $\boldsymbol{o}(\lambda^{-1})$ indicates a vector with the property that

$$\lim_{\lambda\to\infty} \lambda\, \boldsymbol{o}(\lambda^{-1}) = 0 \tag{1.11}$$

while the symbol $\boldsymbol{O}(\lambda^{-1})$ indicates a matrix with the property that

$$\lim_{\lambda\to\infty} \lambda\, \boldsymbol{O}(\lambda^{-1}) = 0. \tag{1.12}$$

Thus the only difference between $\boldsymbol{o}(\lambda^{-1})$ and $\boldsymbol{O}(\lambda^{-1})$ is that the former is a vector while the latter is a matrix. In particular, there is no distinction as to the rate of convergence to zero as $\lambda \to \infty$. This usage is a little different from the standard convention.

II. Some General Results

Throughout this section, we study a very general class of nonlinear neural networks described by

$$\dot{u}_i = -\frac{u_i}{\alpha_i} - f_i(\boldsymbol{x}) + b_i, x_i = g_i(\lambda u_i), \qquad i = 1, \cdots, n \tag{2.1}$$

where $\boldsymbol{f}$: $[0, 1]^n \to \Re^n$ is continuously differentiable.[1] Note that $\boldsymbol{f}$ need not be the gradient of any function. Thus the above class of networks is much more general than the gradient neural networks studied in earlier papers and than (1.9). The object of the study is to determine what happens to the location of the equilibria of the network as the sigmoidal gain λ approaches infinity. It turns out that, under very mild conditions, network (2.1) has only a finite number of equilibria for each fixed value of λ. As $\lambda \to \infty$, these equilibria cluster at various points in the hypercube $[0, 1]^n$. The objective of this section is to obtain an explicit characterization of these cluster points. The culmination of the study undertaken in this section is Theorem 2.6, given at the end of the section.

To proceed further, we introduce an assumption on the sigmoidal functions $g_i(\cdot)$.

***A*)** For each i, the function $g_i(\cdot)$: $\Re \to (0, 1)$ is continuously differentiable, strictly increasing, and satisfies

$$g_i(u) \to 0 \text{ as } u \to -\infty, \quad g_i(u) \to 1 \text{ as } u \to \infty \tag{2.2}$$

and

$$\lim_{\lambda\to\infty} \lambda g_i'(\lambda u) = 0 \tag{2.3}$$

uniformly in u over the complement of any open neighborhood of zero.

Condition (2.3) means that, given any $\epsilon, \delta > 0$, however small, there exists a number M such that

$$|u| \geq \delta, \quad \lambda \geq M \Rightarrow |\lambda g_i'(\lambda u)| \leq \epsilon. \tag{2.4}$$

This is a very mild assumption. The fact that $g_i(u)$ has a definite limit as $|u| \to \infty$ implies that

$$\liminf_{|u|\to\infty} u g_i'(u) = 0 \tag{2.5}$$

because the function $u \mapsto 1/u$ is not integrable over any infinite interval. Hence, for any fixed $u \neq 0$, we have

$$\liminf_{\lambda\to\infty} \lambda g_i(\lambda u) = 0. \tag{2.6}$$

So the essence of the assumption is to i) replace "lim inf" by "lim" in (2.6) and ii) to make the limit uniform with respect to u, as u varies over the complement of a neighborhood of zero. Note that the commonly used sigmoid function $g_i(u) = 1/(1 + e^{-u})$ satisfies this assumption.

For purely technical reasons, it is assumed that the gain λ assumes one of only countably many values. To be precise, we suppose that $\{\lambda_j\}$ is a sequence of positive numbers approaching infinity and study what happens when $\lambda = \lambda_j$, $j = 1, 2, \cdots$.

Let us begin by recalling a few standard concepts and facts. Suppose $\boldsymbol{h}$: $\Re^n \to \Re^n$ is continuously differentiable. A vector $\boldsymbol{x} \in \Re^n$ is called a critical point of the map $\boldsymbol{h}$ if the Jacobian matrix $\nabla \boldsymbol{h}(\boldsymbol{x})$ is singular; otherwise, it is called a regular point of $\boldsymbol{h}$. A vector $\boldsymbol{y} \in \Re^n$ is called a regular value of the map $\boldsymbol{h}$ if every point in $\boldsymbol{h}^{-1}(\boldsymbol{y})$ is a regular point of $\boldsymbol{h}$; otherwise, $\boldsymbol{y}$ is called a critical value of $\boldsymbol{h}$. If $\boldsymbol{h}^{-1}(\boldsymbol{y})$ is empty, then $\boldsymbol{y}$ is vacuously deemed to be a regular value. Note that if $\boldsymbol{y}$ is a regular value of $\boldsymbol{h}$, then every point in the set $\boldsymbol{h}^{-1}(\boldsymbol{y})$ is isolated; that is, every point $\boldsymbol{x}$ in $\boldsymbol{h}^{-1}(\boldsymbol{y})$ has a neighborhood $N(\boldsymbol{x})$ that does not contain any other point of $\boldsymbol{h}^{-1}(\boldsymbol{y})$. This is a ready consequence of the fact that $\nabla \boldsymbol{h}(\boldsymbol{x})$ is nonsingular and the inverse function theorem. A standard argument based on the Heine–Borel theorem shows that if $\boldsymbol{y}$ is a regular value of $\boldsymbol{h}$ and $S \subseteq \Re^n$ is compact, then S can contain at most a finite number of points of $\boldsymbol{h}^{-1}(\boldsymbol{y})$. In other words, the set S contains at most a finite number of solutions $\boldsymbol{x}$ of the equation $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{y}$. The famous theorem of Sard [24] states, quite simply, that the set of critical values of any differentiable map has measure zero. Thus, for any differentiable map $\boldsymbol{h}$, it is the case that the equation $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{y}$ has only a finite number of solutions in any compact set $S \subseteq \Re^n$, for "almost all" values of $\boldsymbol{y}$.

[1] This means that $\boldsymbol{f}$ is continuously differentiable over some open set containing $[0, 1]^n$.

The objective of this section is to examine the behavior of the equilibria of neural network (2.1) as $\lambda \to \infty$. Clearly, a vector $\boldsymbol{u}$ is an equilibrium of the network (2.1) if and only if

$$0 = -\frac{u_i}{\alpha_i} - f_i(\boldsymbol{x}) + b_i, \quad x_i = g_i(\lambda u_i), \qquad i = 1, \cdots, n. \tag{2.7}$$

This can be expressed more compactly. Define $\boldsymbol{A} \in \Re^{n \times n}$ to be the diagonal matrix Diag $\{\alpha_1, \cdots, \alpha_n\}$, and define maps $\boldsymbol{f}$: $[0,1]^n \to \Re^n, \boldsymbol{g}$: $\Re^n \to (0,1)^n$, and the vector $\boldsymbol{b} \in \Re^n$ in the obvious way. Then (2.7) can be rewritten as

$$0 = -\boldsymbol{A}^{-1}\boldsymbol{u} - \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{b}, \quad \boldsymbol{x} = \boldsymbol{g}(\lambda \boldsymbol{u}). \tag{2.8}$$

Proposition 2.1: There exists a number μ such that, if $\boldsymbol{u} \in \Re^n$ satisfies (2.8) for some λ, then $\|\boldsymbol{u}\| \leq \mu$.

Proof: Rewrite (2.8) as

$$\boldsymbol{u} = \boldsymbol{A}[\boldsymbol{b} - \boldsymbol{f}(\boldsymbol{x})]. \tag{2.9}$$

Since $\boldsymbol{f}$ is continuous and $\boldsymbol{x} \in [0,1]^n$, there is a finite constant M such that $\|\boldsymbol{f}(\boldsymbol{x})\| \leq M$ for all $\boldsymbol{x} \in [0,1]^n$. Thus (2.9) implies that

$$\|\boldsymbol{u} \leq \|\boldsymbol{A}\| \cdot [\|\boldsymbol{b}\| + M] \tag{2.10}$$

where $\|\boldsymbol{A}\|$ denotes the induced matrix norm of $\boldsymbol{A}$, i.e., the largest singular value of $\boldsymbol{A}$.

Now we focus on the case where $\lambda = \lambda_j$ for some index j, and $\{\lambda_j\}$ is a sequence of positive numbers approaching infinity. In other words, λ is restricted to any one of a countable set of values. Define $\mathcal{E}_j$ to be the set of solutions $\boldsymbol{u}$ of the equation

$$0 = -\boldsymbol{A}^{-1}\boldsymbol{u} - \boldsymbol{f}[\boldsymbol{g}(\lambda_j \boldsymbol{u})] + \boldsymbol{b}. \tag{2.11}$$

Thus, $\mathcal{E}_j$ is the set of equilibria (in the $\boldsymbol{u}$-space) of network (2.1) when λ is set equal to λ_j. Next, define

$$\mathcal{E} = \bigcup_j \mathcal{E}_j. \tag{2.12}$$

Thus $\mathcal{E}$ is the set of all equilibria (in the $\boldsymbol{u}$-space) of (2.1), for whatever λ. The sets $\mathcal{E}_j$ and $\mathcal{E}$ pertain to the $\boldsymbol{u}$-space and are subsets of $\Re^n$. It is actually more convenient to work in the $\boldsymbol{x}$-space. Define $\mathcal{S}_j$ to be the set of solutions $\boldsymbol{x}$ of the equation

$$0 = -\lambda_j^{-1}\boldsymbol{A}^{-1}\boldsymbol{g}^{-1}(\boldsymbol{x}) - f(x) + \boldsymbol{b} \tag{2.13}$$

and define

$$\mathcal{S} = \bigcup_j \mathcal{S}_j. \tag{2.14}$$

Observe that for each j, the map $\boldsymbol{u} \mapsto \boldsymbol{g}(\lambda_j \boldsymbol{u})$ is a diffeomorphism of $\Re^n$ onto $(0,1)^n$. Hence there is a one-to-one correspondence between the sets $\mathcal{E}_j$ and $\mathcal{S}_j$. But the same need not be true of the sets $\mathcal{E}$ and $\mathcal{S}$, as these are unions of sets corresponding to different values of λ.

Proposition 2.2: For all $\boldsymbol{b} \in \Re^n$ except those belonging to a set of measure zero, each set $\mathcal{E}_j$ contains only a finite number of points.

Proof: Define $\mathcal{C}_j$ to be the set of critical values of the map $\boldsymbol{u} \mapsto \boldsymbol{A}^{-1}\boldsymbol{u} + \boldsymbol{f}[\boldsymbol{g}(\lambda \boldsymbol{u})]$, and observe that each $\mathcal{C}_j$ has measure zero, by Sard's theorem. Therefore

$$\mathcal{C} = \bigcup_j \mathcal{C}_j \tag{2.15}$$

also has measure zero, because $\mathcal{C}$ is a countable union of sets of measure zero.[2] Thus, if $\boldsymbol{b} \notin \mathcal{C}$, then $\boldsymbol{b}$ is a regular value of each map $\boldsymbol{u} \mapsto \boldsymbol{A}^{-1}\boldsymbol{u} + \boldsymbol{f}[\boldsymbol{g}(\lambda \boldsymbol{u})]$. Hence (2.11) has only a finite number of solutions in the sphere $\|\boldsymbol{u}\| \leq \mu$. From Proposition 2.1, these are all the solutions of (2.11).

Throughout the rest of the section, it is assumed that $\boldsymbol{b} \notin \mathcal{C}$. Additional assumptions on $\boldsymbol{b}$ are added as we go on.

Proposition 2.2 shows that, for almost all $\boldsymbol{b} \in \Re^n$, the set $\mathcal{S}$ is countable. Let us now study the cluster points of the set $\mathcal{S}$. Recall that a vector $\boldsymbol{p} \in \Re^n$ is defined to be a cluster point of a set $S \subseteq \Re^n$ if each neighborhood of $\boldsymbol{p}$ contains an element of S not equal to $\boldsymbol{p}$. Equivalently, $\boldsymbol{p}$ is a cluster point of S if and only if there exists a sequence $\{\boldsymbol{s}_i\}$ in S converging to $\boldsymbol{p}$ such that the sequence $\{\boldsymbol{s}_i\}$ is pairwise distinct, that is, $\boldsymbol{s}_i \neq \boldsymbol{s}_j$ if $i \neq j$.

Note that a countable set can have an uncountable number of cluster points. For example, let S be the set of all rational numbers in the interval [0, 1]; then the set of cluster points of S is the set of all numbers in [0, 1]. The objective of the remainder of the section is to show that, for the set $\mathcal{S}$ of equilibria of neural network (2.1), such a thing does not happen. In fact, for almost all bias vectors $\boldsymbol{b}$, there are only a finite number of cluster points, and these can be described explicitly. Moreover, as λ_j becomes large, network (2.1) has exactly as many equilibria as the number of cluster points, and each equilibrium of (2.1) is "close" to the corresponding cluster point. To arrive at this comprehensive characterization of the behavior of the equilibria of (2.1), we go through several intermediate steps. The first step is to categorize the cluster points of the set $\mathcal{S}$ into three groups. Note that, since $\mathcal{S} \subseteq [0,1]^n$, all cluster points of $\mathcal{S}$ must belong to $[0,1]^n$.

Definition 2.1: Suppose $\boldsymbol{v} \in [0,1]^n$ is a cluster point of $\mathcal{S}$. Then $\boldsymbol{v}$ is said to be an interior point if $\boldsymbol{v} \in (0,1)^n$, a corner point if $\boldsymbol{v} \in \{0,1\}^n$, and a face point otherwise.

Suppose $\boldsymbol{v} \in [0,1]^n$ is a cluster point of $\mathcal{S}$. Then there exists a pairwise distinct sequence $\{\boldsymbol{v}_i\}$ in $\mathcal{S}$ converging to $\boldsymbol{v}$. In other words, there exists a sequence of indexes $\{j_i\}$ such that $\boldsymbol{v}_i \in \mathcal{S}_{j_i}$ and $\boldsymbol{v}_i \to \boldsymbol{v}$. Equivalently

$$0 = -\lambda_{j_i}^{-1}\boldsymbol{A}^{-1}\boldsymbol{g}^{-1}(\boldsymbol{v}_i) - f(\boldsymbol{v}_i) + \boldsymbol{b} \tag{2.16}$$

and $\boldsymbol{v}_i \to \boldsymbol{v}$. Now, because the sequence $\{\boldsymbol{v}_i\}$ is pairwise distinct and each set $\mathcal{S}_i$ is finite, it is clear that the sequence $\{j_i\}$ cannot contain any integer more than a finite number of times; thus $j_i \to \infty$. Therefore, without loss of generality, we can renumber the sequence $\{\lambda_{j_i}\}$ as $\{\lambda_i\}$ and observe that the renumbered sequence $\{\lambda_i\}$ also approaches infinity. With the renumbering, (2.16) becomes

$$0 = -\lambda_i^{-1}\boldsymbol{A}^{-1}\boldsymbol{g}^{-1}(\boldsymbol{v}_i) - \boldsymbol{f}(\boldsymbol{v}_i) + \boldsymbol{b}. \tag{2.17}$$

[2] This is why λ is restricted to take on any one of a countable set of values.

Of course, the renumbering depends on the cluster point. But it simplifies notation by eliminating the double subscripts.

The next three propositions give explicit characterizations of each of the three types of cluster points.

Proposition 2.3: Suppose $\boldsymbol{b}$ is a regular value of the function $\boldsymbol{f}$. Then $\boldsymbol{v} \in (0,1)^n$ is a cluster point of $\mathcal{S}$ if and only if

$$\boldsymbol{f}(\boldsymbol{v}) = \boldsymbol{b}. \tag{2.18}$$

Proof: Suppose $\boldsymbol{v} \in (0,1)^n$ is a cluster point of $\mathcal{S}$; it is shown that $\boldsymbol{v}$ satisfies (2.18). Choose $\{\boldsymbol{v}_i\}, \{\lambda_i\}$ such that (2.17) holds, and $\boldsymbol{v}_i \to \boldsymbol{v}$. Also, assume without loss of generality that the components of $\boldsymbol{v}_i$ are bounded away from zero and one for all i; that is, suppose that there exists an $\epsilon > 0$ such that

$$\epsilon \le (\boldsymbol{v}_i)_k \le 1 - \epsilon, \qquad k = 1, \cdots, n, \text{ for all } i. \tag{2.19}$$

This is possible because $\boldsymbol{v}$ lies in the interior of $[0,1]^n$ and $\boldsymbol{v}_i \to \boldsymbol{v}$. Now (2.19) implies that $\|\boldsymbol{g}^{-1}(\boldsymbol{v}_i)\|$ can be bounded with respect to i by some finite constant. Letting $i \to \infty$ in (2.17) establishes (2.18).

Conversely, suppose $\boldsymbol{v} \in (0,1)^n$ satisfies (2.18); it is shown that $\boldsymbol{v}$ is a cluster point of $\mathcal{S}$. Since $\boldsymbol{b}$ is a regular value of $\boldsymbol{f}$, it follows that $\boldsymbol{v}$ is a regular point of $\boldsymbol{f}$, i.e., that $\nabla \boldsymbol{f}(\boldsymbol{v})$ is nonsingular. Choose a $\delta \in (0, \epsilon)$ such that $\nabla \boldsymbol{f}(\boldsymbol{x})$ is nonsingular and $\|[\nabla \boldsymbol{f}(\boldsymbol{x})]^{-1}\|$ is bounded above whenever $\boldsymbol{x} \in B(\boldsymbol{v}, \delta)$, where $B(\boldsymbol{v}, \delta)$ denotes the ball $\{\boldsymbol{x}\colon \|\boldsymbol{x} - \boldsymbol{v}\| < \delta\}$. Now consider the equation

$$\boldsymbol{f}(\boldsymbol{x}) + \lambda^{-1} \boldsymbol{A}^{-1} \boldsymbol{g}^{-1}(\boldsymbol{x}) = \boldsymbol{b}. \tag{2.20}$$

The closed ball $B = B(\boldsymbol{v}, \delta)$ lies inside the interior of the open hypercube $(0,1)^n$. Hence $\|\nabla \boldsymbol{g}^{-1}(\boldsymbol{x})\|$ is uniformly bounded as $\boldsymbol{x}$ varies over B. Now a routine application of the contraction mapping theorem shows that (2.20) has a unique solution for $\boldsymbol{x}$ whenever λ is sufficiently large. This can be seen by defining $\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$ and rewriting (2.20) as

$$\boldsymbol{y} + \lambda^{-1} \boldsymbol{A}^{-1} \boldsymbol{g}^{-1}[\boldsymbol{f}^{-1}(\boldsymbol{y})] = \boldsymbol{b} \tag{2.21}$$

or equivalently

$$\boldsymbol{y} = \boldsymbol{b} - \lambda^{-1} \boldsymbol{A}^{-1} \boldsymbol{g}^{-1}[\boldsymbol{f}^{-1}(\boldsymbol{y})]. \tag{2.22}$$

The right side of (2.22) is a contraction whenever λ^{-1} is sufficiently small. Hence (2.21) has a unique solution for $\boldsymbol{y}$, and (2.20) has a unique solution for $\boldsymbol{x}$, whenever λ exceeds some number, call it M. To complete the argument, select a subsequence of the original $\{\lambda_i\}$ such that $\lambda_i > M$ for all i; renumber this subsequence as $\{\lambda_i\}$; let $\boldsymbol{v}_i$ denote the corresponding unique solution of (2.21) inside the ball B. Then $\boldsymbol{v} \in \mathcal{S}$ and $\boldsymbol{v}_i \to \boldsymbol{v}$. Thus $\boldsymbol{v}$ is a cluster point of $\mathcal{S}$.

Proposition 2.4: Suppose $\boldsymbol{e} \in \{0,1\}^n$, and define

$$\boldsymbol{z} = \boldsymbol{b} - \boldsymbol{f}(\boldsymbol{e}). \tag{2.23}$$

Choose $\boldsymbol{b}$ such that no component of $\boldsymbol{z}$ is zero. Then $\boldsymbol{e}$ is a cluster point of $\mathcal{S}$ if and only if the vector $\boldsymbol{z}$ satisfies the parity condition

$$z_k < 0 \text{ if } e_k = 0, \quad z_k > 0 \text{ if } e_k = 1, \qquad k = 1, \cdots, n. \tag{2.24}$$

Proof: Suppose $\boldsymbol{e} \in \{0,1\}^n$ is a cluster point of $\mathcal{S}$; it is shown that the parity condition (2.24) is satisfied. Select $\lambda_i, \boldsymbol{v}_i$ such that (2.17) holds and such that $\boldsymbol{v}_i \to \boldsymbol{e}$. Define $\boldsymbol{u}_i = \lambda_i^{-1} \boldsymbol{g}^{-1}(\boldsymbol{v}_i)$. Then $\boldsymbol{u}_i$ satisfies the $\boldsymbol{u}$-space counterpart of (2.17), namely

$$\boldsymbol{A}^{-1} \boldsymbol{u}_i = -\boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{u}_i)] + \boldsymbol{b}. \tag{2.25}$$

Now let $i \to \infty$. Then $\boldsymbol{g}(\lambda_i \boldsymbol{u}_i) = \boldsymbol{v}_i \to \boldsymbol{e}$, so that

$$\boldsymbol{A}^{-1} \boldsymbol{u}_i \to -\boldsymbol{f}(e) + \boldsymbol{b} = \boldsymbol{z}. \tag{2.26}$$

In other words

$$\boldsymbol{u}_i \to \boldsymbol{A}\boldsymbol{z}. \tag{2.27}$$

Now, by assumption, each component of $\boldsymbol{z}$ is nonzero. Hence each component of $\boldsymbol{u}_i$ approaches a nonzero number. The only remaining question is the sign of this limit. Fix an index $k \in \{1, \cdots, n\}$, and let t_k denote the limit of the sequence $\{(\boldsymbol{u}_i)_k\}$. In other words

$$\boldsymbol{t} = \boldsymbol{A}\boldsymbol{z}. \tag{2.28}$$

By assumption, $t_k \neq 0$. Thus, since $\lambda_i \to \infty$, it follows that

$$g_k[\lambda_i (\boldsymbol{u}_i)_k] \to 0 \text{ if } t_k < 0, \to 1 \text{ if } t_k > 0. \tag{2.29}$$

In other words

$$t_k < 0 \text{ if } e_k = 0, \quad t_k > 0 \text{ if } e_k = 1. \tag{2.30}$$

Now (2.24) follows from (2.27) and (2.30), and observing that $t_k = \alpha_k z_k$.

To prove the converse, suppose $\boldsymbol{e} \in \{0,1\}^n$, define $\boldsymbol{z}$ as in (2.23), and suppose the parity condition (2.24) is satisfied; it is shown that $\boldsymbol{e}$ is a cluster point of $\mathcal{S}$. Define

$$\boldsymbol{u}_0 = \boldsymbol{A}\boldsymbol{z} = \boldsymbol{A}\boldsymbol{b} - \boldsymbol{A}\boldsymbol{f}(\boldsymbol{e}). \tag{2.31}$$

Then no component of $\boldsymbol{u}_0$ is zero, by assumption. Pick an $\epsilon > 0$ such that

$$\|\boldsymbol{u}\| \ge \|\boldsymbol{u}_0\| - \epsilon \Rightarrow u_k \neq 0 \text{ for all } k. \tag{2.32}$$

Define the set

$$U = \{\boldsymbol{u} \in \Re^n\colon \|\boldsymbol{u}\| \ge \|\boldsymbol{u}_0\| - \epsilon\}. \tag{2.33}$$

Then U is the complement of the open ball $\{\boldsymbol{u}\colon \|\boldsymbol{u}\| < \|\boldsymbol{u}_0\| - \epsilon\}$ that contains the origin. Moreover, since U is a closed subset of $\Re^n$, it is a complete metric space in its own right. Now by Assumption A) on the sigmoidal functions

$$\lim_{i \to \infty} \|\nabla \boldsymbol{g}(\lambda_i \boldsymbol{u})\| \to 0, \text{uniformly for } \boldsymbol{u} \in U. \tag{2.34}$$

Also, since the parity condition is satisfied, it follows that

$$\lim_{i \to \infty} \|\boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{u}_0)] - \boldsymbol{f}(\boldsymbol{e})\| = 0. \tag{2.35}$$

Hence it is possible to find an index N such that, whenever $i > N$, we have

$$\|\boldsymbol{A}\| \cdot \|\boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{u})]\| \|\lambda_i \nabla \boldsymbol{g}(\lambda_i \boldsymbol{u})\| \le 1 - \rho, \text{ for all } \boldsymbol{u} \in U, \tag{2.36}$$

$$\|\boldsymbol{A}\| \cdot \|\boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{u}_0)] - \boldsymbol{f}(\boldsymbol{e})\| \le \epsilon\rho \tag{2.37}$$

where $\rho \in (0, 1)$ is arbitrary. Now look at the equation

$$\boldsymbol{u} = \boldsymbol{A}\boldsymbol{b} - \boldsymbol{A}\boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{u})]. \tag{2.38}$$

For each fixed i, the map $\boldsymbol{u} \mapsto \boldsymbol{A}\boldsymbol{b} - \boldsymbol{A}\boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{u}] =: T_i \boldsymbol{u}$ is a contractive map on U, in view of (2.36). If we define

$$\boldsymbol{u}_i^{(1)} = T_i \boldsymbol{u}_0 = \boldsymbol{A}\boldsymbol{b} - \boldsymbol{A}\boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{u}_0)] \tag{2.39}$$

then

$$\|\boldsymbol{u}_i^{(1)} - \boldsymbol{u}_0\| = \|\boldsymbol{A}\{\boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{u}_0)] - \boldsymbol{f}(\boldsymbol{e})\}\| \le \epsilon\rho \tag{2.40}$$

from (2.31) and (2.37). Now rewrite (2.40) as

$$\frac{\|\boldsymbol{u}_i^{(1)} - \boldsymbol{u}_0\|}{\rho} \le \epsilon. \tag{2.41}$$

By the local version of the contraction mapping theorem [28, Theorem 2.3.15], it follows that (2.38) has a unique solution in U; call it $\boldsymbol{u}_i$. Define $\boldsymbol{v}_i = \boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{u}_i)]$. Now it is easy to see that $\boldsymbol{u}_i \to \boldsymbol{u}_0$ and that $\boldsymbol{v}_i \to \boldsymbol{e}$. Moreover, $\boldsymbol{u}_i \in \mathcal{E}_i$ and $\boldsymbol{v}_i \in \mathcal{S}_i$. Hence $\boldsymbol{e}$ is a cluster point of $\mathcal{S}$.

Thus far we have examined the cluster points of $\mathcal{S}$ in the interior of $[0, 1]^n$ and in the corners of $[0, 1]^n$. Next we examine the cluster points in the faces of $[0, 1]^n$. For this purpose, it is useful to have another interpretation of a face. Let I be a nonempty proper subset of $\{1, \cdots, n\}$, and let $\boldsymbol{v}_2 \in \{0, 1\}^{|I|}$, where $|I|$ is the cardinality of the set I. Then the set

$$\mathcal{F}_I = \{\boldsymbol{x} \in [0, 1]^n \colon x_i = (\boldsymbol{v}_2)_i \text{ if } i \in I, x_i \in (0, 1) \text{ if } i \notin I\} \tag{2.42}$$

is an (open) face of the hypercube $[0, 1]^n$. Once the index set I is chosen, there are $2^{|I|}$ ways of choosing the binary vector $\boldsymbol{v}_2$. Each choice of I and a corresponding choice of $\boldsymbol{v}_2$ defines an open face of $[0, 1]^n$. Conversely, each open face has associated with it a unique set I and a unique binary vector $\boldsymbol{v}_2$. It is easy to see that the union of the interior of $[0, 1]^n$, the corners of $[0, 1]^n$, and the faces of $[0, 1]^n$ is precisely $[0, 1]^n$. Hence, a cluster point of S that is neither an interior point nor a corner must belong to a face.

As one would expect, the conditions for a vector $\boldsymbol{v}$ belonging to a face of $[0, 1]^n$ to be a cluster point of $\mathcal{S}$ are a combination of Propositions 2.3 and 2.4. But the notation is a little cumbersome. Suppose $\boldsymbol{v}$ belongs to a face of $[0, 1]^n$. Let I denote the index set associated with this face. We can define the vectors

$$\begin{aligned} \boldsymbol{v}_1 &= \{v_i, i \notin I\} \in (0, 1)^{n-|I|}, \\ \boldsymbol{v}_2 &= \{v_i, i \in I\} \in \{0, 1\}^{|I|}. \end{aligned} \tag{2.43}$$

Proposition 2.5: Suppose $\boldsymbol{v}$ belongs to a face of $[0, 1]^n$, and define $\boldsymbol{v}_1$ as in (2.43). Partition $\boldsymbol{f}$ and $\boldsymbol{b}$ commensurately with $\boldsymbol{v}$. Suppose the bias vector $\boldsymbol{b}$ has two properties: i) $\boldsymbol{b}_1$ is a regular value of the map $\boldsymbol{x}_1 \mapsto \boldsymbol{f}_1(\boldsymbol{x}_1, \boldsymbol{v}_2)$, and ii) the vector

$$\boldsymbol{z}_2 = -\boldsymbol{f}_2(\boldsymbol{v}_1, \boldsymbol{v}_2) + \boldsymbol{b}_2 \tag{2.44}$$

has all nonzero components. Under these conditions, $\boldsymbol{v}$ is a cluster point of $\mathcal{S}$ if and only if i) $\boldsymbol{v}_1$ is a solution for $\boldsymbol{x}_1$ of the equation

$$\boldsymbol{f}_1(\boldsymbol{x}_1 . \boldsymbol{v}_2) = \boldsymbol{b}_1 \tag{2.45}$$

and ii) the vector $\boldsymbol{z}_2$ satisfies the parity condition

$$(\boldsymbol{z}_2)_i < 0 \text{ if } (\boldsymbol{v}_2)_i = 0, \quad (\boldsymbol{z}_2)_i > 0 \text{ if } (\boldsymbol{v}_2)_i = 1. \tag{2.46}$$

The proof is essentially a combination of those of Propositions 2.3 and 2.4 and is therefore omitted.

All of the above preceding results are now combined into a single statement.

Theorem 2.6: Suppose $\boldsymbol{b} \in \Re^n$ satisfies the following conditions:

1) $\boldsymbol{b}$ is a regular value of the map $\boldsymbol{u} \mapsto \boldsymbol{A}^{-1}\boldsymbol{u} + \boldsymbol{f}[\boldsymbol{g}(\lambda_i \boldsymbol{i})]$ for all i.
2) $\boldsymbol{b}$ is a regular value of the map $\boldsymbol{x} \mapsto \boldsymbol{f}(\boldsymbol{x})$.
3) For each binary vector $\boldsymbol{e} \in \{0, 1\}^n$, the vector $\boldsymbol{b} - \boldsymbol{f}(\boldsymbol{e})$ has all nonzero components.
4) For each nonempty proper subset I of $\{1, \cdots, n\}$ and each binary vector $\boldsymbol{v}_2 \in \{0, 1\}^{|I|}$, the vector $\boldsymbol{b}_1$ is a regular value of the map $\boldsymbol{x}_1 \mapsto \boldsymbol{f}_1(\boldsymbol{x}_1, \boldsymbol{v}_2)$.
5) For each solution $\boldsymbol{v}_1$ of the equation $\boldsymbol{f}_1(\boldsymbol{v}_1, \boldsymbol{v}_2) = \boldsymbol{b}_1$, the vector $\boldsymbol{b}_2 - \boldsymbol{f}_2(\boldsymbol{v}_1, \boldsymbol{v}_2)$ has all nonzero components.

Under these conditions, the following conclusions hold:

A) The cluster points of $\mathcal{S}$ in $(0, 1)^n$ are precisely the vectors $\boldsymbol{v}$ that satisfy the equation $\boldsymbol{f}(\boldsymbol{v}) = \boldsymbol{b}$, and these are finite in number.

B) The cluster points of $\mathcal{S}$ in $\{0, 1\}^n$ are precisely the binary vectors that satisfy parity condition (2.24).

C) The cluster points of $\mathcal{S}$ in the faces of $[0, 1]^n$ are precisely those vectors $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ that satisfy (2.45) and (2.46), and these are finite in number.

D) Let $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_r$ denote the cluster points of $\mathcal{S}$. Then there exists an integer N and a $\delta > 0$ such that, for each $j > N$, (2.13) has precisely r solutions; call them $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_r$. Moreover, these solutions can be numbered in such a way that $\|\boldsymbol{x}_k - \boldsymbol{v}_k\| < \delta$ for all k.

Proof: The hypotheses of the theorem include those of Propositions 2.3–2.5. Hence Conclusions A)-C) follow from these propositions. The only conclusion that remains to be proved is Conclusion D).

Let $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_r$ denote the cluster points of the set $\mathcal{S}$. From the proofs of Propositions 2.3–2.5, it follows that, for δ sufficiently small and j sufficiently large, each ball $B(\boldsymbol{v}_k, \delta)$ contains exactly one solution of (2.13). By hypothesis 1), (2.13) has a finite number of solutions for each j. It is of course possible that, for some j, (2.13) has some other solutions outside the balls $B(\boldsymbol{v}_k, \delta)$. This cannot happen, however, for infinitely many values of j, because in that case $\mathcal{S}$ would contain infinitely many points outside the union of the balls $B(\boldsymbol{v}_k, \delta)$, and would therefore have another cluster point besides $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_r$.

Remarks:

1) The hypotheses of Theorem 2.6 are just the union of the hypotheses of Propositions 2.3–2.5. The important point to note is that these five hypotheses hold for all $\boldsymbol{b} \in \Re^n$ except those belonging to a set of measure zero. The first three conclusions of the theorem are again a repeat of those of Propositions 2.3–2.5. The extra conclusion is the last one, which states that, for sufficiently large λ. the number of equilibria of network (2.1) equals

the number of cluster points and that there is a one-to-one correspondence between them. Thus, for almost all choices of the bias vector, the "high gain" behavior of network (2.1) is quite easy to analyze. As shown by Examples 4.1–4.3, however, the "natural choice" of $\boldsymbol{b} = 0$ just may belong to the exceptional set. Therefore, the bias term is very important to be able to predict the high gain behavior of network (2.1).

2) If one wishes to determine the cluster points of (2.1) corresponding to a given bias vector $\boldsymbol{b}$, one could proceed as follows, in principle at least: First, find all solutions of $\boldsymbol{f}(\boldsymbol{v}) = \boldsymbol{b}$. These are the interior cluster points. Next, check the 2^n corner points of $\{0, 1\}^n$ to see whether they satisfy parity condition (2.24). These are the corner cluster points. Finally, for each nonempty proper subset I of $\{1, \cdots, n\}$ and each binary vector $\boldsymbol{v}_2 \in \{0, 1\}^{|I|}$, do two things: Solve the equation $\boldsymbol{f}_1(\boldsymbol{v}_1, \boldsymbol{v}_2) = \boldsymbol{b}_1$ for $\boldsymbol{v}_1$. For each solution $\boldsymbol{v}_1$, check parity condition (2.46). Those pairs $(\boldsymbol{v}_1, \boldsymbol{v}_2)$ that satisfy (2.46) are the cluster points in the faces. In practice, however, the above procedure is not recommended, as it is computationally expensive. Just testing the parity condition at the various corners of $\{0, 1\}^n$ alone requires 2^n computations!

III. Main Results

Suppose the function E: $[0, 1]^n \rightarrow \Re$ is a multilinear polynomial and that it is desired to minimize $E(\boldsymbol{x})$ as $\boldsymbol{x}$ varies over the discrete-set $\{0, 1\}^n$. For this purpose, we propose an analog neural network described by

$$\begin{aligned} \dot{u}_i &= -\frac{u_i}{\alpha_i} - [\nabla E(\boldsymbol{x})]_i + b_i, \\ x_i &= g_i(\lambda u_i), \qquad i = 1, \cdots, n. \end{aligned} \tag{3.1}$$

Network (3.1) is of the form (2.1), with $\boldsymbol{f}$ replaced by ∇E. Hence Theorem 2.6 applies to this network as well. Moreover, in view of the special (gradient) nature of $\boldsymbol{f}$, the analysis can be refined further. In particular, the network is totally stable, and every solution trajectory approaches an equilibrium. (This is true even if E is not a polynomial.) When E is a multilinear polynomial, the equilibria of (3.1) that cluster in the interior of $[0, 1]^n$ turn out to be hyperbolic and unstable for sufficiently large values of λ, as do the equilibria that cluster in the faces of $[0. 1]^n$. Only the equilibria that approach a corner of $[0. 1]^n$ turn out to be asymptotically stable. Hence all solution trajectories, except for those originating in the stable manifolds of the unstable equilibria, approach the "corner" equilibria, which correspond precisely to the local minima of the objective function E. This fact, described in Theorem 3.2, is the main result of the paper.

Comparing (3.1) with (1.5), we can observe two important differences: First, the two "physical" constants C_i and R_i have been combined into a single time constant α_i. This is a very minor difference. Second, a "bias" input b_i is introduced into each of the differential equations. This is a substantial difference and is a novel feature of the network proposed here. The role of the bias term is explained below, just after the statement of Theorem 3.2. If the function $E(\boldsymbol{x})$ is quadratic of the form (1.2), then (3.1) reduces to (1.5), except for the bias term. Also, note that the function $E(\boldsymbol{x})$ of (1.2) is multilinear if and only if $w_{ii} = 0$ for all i, i.e., there are no self-interactions.

Proposition 3.1: Neural network (3.1) is totally stable.

Proof: It is first shown, as in [23], that the solution trajectories of (3.1) are eventually confined to a bounded region in $\boldsymbol{u}$-space; in other words, the network is Lagrange stable.

Let $r(t) = \|\boldsymbol{u}(t)\|^2$. Then, from (3.1)

$$\frac{dr}{dt} = 2\boldsymbol{u}^t\dot{\boldsymbol{u}} = -2[\boldsymbol{u}^t A^{-1}\boldsymbol{u} + \boldsymbol{u}^t\nabla E(\boldsymbol{x}) - \boldsymbol{u}^t\boldsymbol{b}] \tag{3.2}$$

where the argument t is suppressed for clarity. Let

$$\alpha_0 = \max_i \alpha_i \tag{3.3}$$

and observe that, since E is a polynomial, there exists a constant M such that

$$\|\nabla E(\boldsymbol{x})\| \leq M, \text{ for all } \boldsymbol{x} \in (0, 1)^n. \tag{3.4}$$

Thus, from (3.1) and (3.2), we get

$$\frac{1}{2}\frac{dr}{dt} \leq -\frac{\|\boldsymbol{u}\|^2}{\alpha_0} + \|\boldsymbol{u}\|(M + \|\boldsymbol{b}\|) < 0 \text{ if } \|\boldsymbol{u}\| > \alpha_0(M + \|\boldsymbol{b}\|). \tag{3.5}$$

Hence all solution trajectories eventually enter into the closed ball (in the $\boldsymbol{u}$-space) of radius $\alpha_0(M + \|\boldsymbol{b}\|)$ centered at $\boldsymbol{0}$ and stay there. In the $\boldsymbol{x}$-space, there is a corresponding compact subset of $(0, 1)^n$ to which all solution trajectories are eventually confined, for each fixed value of λ.

To complete the proof of total stability, define the Lyapunov function

$$E_c(\boldsymbol{x}) = E(\boldsymbol{x}) - \boldsymbol{b}^t\boldsymbol{x} + \sum_{i=1}^{n} \frac{1}{\lambda\alpha_i}\int_0^{x_i} g_i^{-1}(x)\, dx. \tag{3.6}$$

The function $\dot{E}_c(\boldsymbol{x})$ equals

$$\dot{E}_c(\boldsymbol{x}) = [\nabla E_c(\boldsymbol{x})]^t\dot{\boldsymbol{x}}. \tag{3.7}$$

Now $\dot{\boldsymbol{x}}$ can be computed from (3.1) as

$$\dot{\boldsymbol{x}} = \boldsymbol{J}(\lambda\boldsymbol{u})\lambda\dot{\boldsymbol{u}} = -\lambda\boldsymbol{J}(\lambda\boldsymbol{u})[A^{-1}\boldsymbol{u} + \nabla E(\boldsymbol{x}) - \boldsymbol{b}] \tag{3.8}$$

where $\boldsymbol{J}$ is the Jacobian matrix of the map $\boldsymbol{g}$. Next, from (3.6) and (3.8)

$$\begin{aligned} \nabla E_c(\boldsymbol{x}) &= \nabla E(\boldsymbol{x}) - \boldsymbol{b} + \lambda A^{-1}\boldsymbol{g}^{-1}(\boldsymbol{x}) \\ &= \nabla E(\boldsymbol{x}) - \boldsymbol{b} + A^{-1}\boldsymbol{u} \\ &= -\lambda^{-1}\boldsymbol{J}^{-1}(\lambda\boldsymbol{u})\dot{\boldsymbol{x}}. \end{aligned} \tag{3.9}$$

Hence, from (3.7)

$$\dot{E}_c(\boldsymbol{x}) = -\lambda^{-1}\dot{\boldsymbol{x}}^t\boldsymbol{J}^{-1}(\lambda\boldsymbol{u})\dot{\boldsymbol{x}}. \tag{3.10}$$

Observe that $\boldsymbol{J}^{-1}$ is positive definite. Therefore

$$\dot{E}_c(\boldsymbol{x}) \leq 0 \text{ for all } \boldsymbol{x}, \quad \dot{E}_c(\boldsymbol{x}) < 0 \text{ if } \dot{\boldsymbol{x}} \neq \boldsymbol{0}. \tag{3.11}$$

This completes the proof of total stability.

Now we come to the main result of the paper.

Theorem 3.2: Suppose E is a multilinear polynomial on $\Re^n$ and that the sigmoidal nonlinearities $g_i(\cdot)$ satisfy Assumption A). Let $\{\lambda_j\}$ be any sequence of positive numbers approaching infinity. Under these conditions, there exists an $\epsilon > 0$ such that, for all $\boldsymbol{b}$ with $\|\boldsymbol{b}\| < \epsilon$ except for those belonging to a set of measure zero, for all sufficiently large indexes j, for all initial conditions $\boldsymbol{x}(0)$ except for those belonging to a nowhere dense set, the solution trajectory $\boldsymbol{x}(t)$ converges to a vector $\boldsymbol{e} + o(1/\lambda_j)$, where $\boldsymbol{e} \in \{0,1\}^n$ and $\boldsymbol{e}$ is a local minimum of E over $\{0,1\}^n$.

Remarks:

1) Informally, Theorem 3.2 can be stated as follows: For almost all sufficiently small bias vectors $\boldsymbol{b}$, for all sufficiently large sigmoidal gains λ, solution trajectories of (3.1) starting from almost all initial conditions converge to a corner of the hypercube $[0,1]^n$. Moreover, any such corner is a local minimum of the objective function E over the discrete set $\{0,1\}^n$.
2) The assumption that the objective function E is multilinear is not particularly restrictive. In the two applications cited in the Introduction ([21], [5]), the objective function is already a multilinear polynomial.
3) The bias vector $\boldsymbol{b}$ is very important. Theorem 3.2 states that almost all sufficiently small bias vectors will do the job. But it can happen that the "natural choice" $\boldsymbol{b} = \boldsymbol{0}$ belongs to the exceptional set. It is shown subsequently that the problem of minimizing $E(\boldsymbol{x}) = x_1x_2x_3$ over the corners of the cube $\{0,1\}^3$, which is one of the easiest problems imaginable, requires a nonzero bias vector (see Example 4.1). In general, it appears that whenever the objective function assumes only integer values (as is the case in the application presented in [21] and [5]), the bias vector plays a crucial role.
4) In practice, one would use Theorem 3.2 as follows: To minimize E over $\{0,1\}^n$, one would choose a "small" bias vector $\boldsymbol{b}$, a "large" sigmoidal gain λ, a "random" initial condition $\boldsymbol{u}(0)$ and set network (3.1) in motion. When the vector $\boldsymbol{x}(t)$ appears to be approaching some corner of the hypercube $[0,1]^n$, this vector is rounded off to the nearest binary vector $\boldsymbol{e}$. Then, using Proposition A.1 in the Appendix, one would check whether $\boldsymbol{e}$ satisfies the parity condition (A.5). If so, a local minimum of E will have been found. If the parity condition is not satisfied, then the process is repeated by increasing λ, and only as a last resort, changing the bias vector $\boldsymbol{b}$. Thus, in practice, neural network (3.1) generates candidate local minima, which must then be tested to determine whether they really are local minima. It may be pointed out that this is a common feature of almost all "interior point" methods of discrete optimization.

The proof of Theorem 3.2 proceeds through several stages. Theorem 2.6 forms the starting point of our study. The bias vector $\boldsymbol{b}$ is chosen so as to satisfy all five hypotheses of this theorem, with $\boldsymbol{f}(\boldsymbol{x}) = \nabla E(\boldsymbol{x})$. By dropping a finite number of terms from the sequence $\{\lambda_i\}$ if necessary, we can make the following assumptions:

1) The set S has exactly r cluster points, $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_r$. These are divided into three groups: $\boldsymbol{p}_1, \cdots, \boldsymbol{p}_s \in (0,1)^n$, $\boldsymbol{e}_1, \cdots, \boldsymbol{e}_c \in \{0,1\}^n$, and $\boldsymbol{q}_1, \cdots, \boldsymbol{q}_f$ belong to the faces of $[0,1]^n$.
2) The balls $B(\boldsymbol{v}_k, \delta)$ are pairwise disjoint, and each ball contains exactly one solution of (2.13), for each j.

Before proceeding further, a few standard concepts are recalled. For further details, see [13].

Definition 3.1: A matrix $\boldsymbol{M}$ is said to be hyperbolic if it has no eigenvalues with a zero real part.

Definition 3.2: Consider a differential equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{h}[\boldsymbol{x}(t)] \tag{3.12}$$

where $\boldsymbol{h}$ is continuously differentiable. Suppose $\boldsymbol{x}_0$ is an equilibrium of (3.1), i.e., that $\boldsymbol{h}(\boldsymbol{x}_0) = \boldsymbol{0}$. Then the equilibrium $\boldsymbol{x}_0$ is said to be hyperbolic if the matrix

$$\boldsymbol{H} := \nabla \boldsymbol{h}(\boldsymbol{x}_0) \tag{3.13}$$

is hyperbolic.

Hyperbolic equilibria have several advantageous features, one of which is that their stable and unstable manifolds have complementary dimensions and intersect transversally.

The next result is at the level of a homework problem, but it does not seem to be stated in this form anywhere.

Proposition 3.3: Suppose $\boldsymbol{P}, \boldsymbol{Q}$ are symmetric matrices, with $\boldsymbol{Q}$ positive definite. Then $\boldsymbol{PQ}$ has only real eigenvalues. If $\boldsymbol{P}$ is nonsingular, then $\boldsymbol{PQ}$ is hyperbolic.

Proof: Let $\boldsymbol{Q}^{1/2}$ denote the symmetric square root of $\boldsymbol{Q}$ and observe that

$$\boldsymbol{PQ} = \boldsymbol{Q}^{-1/2}\boldsymbol{Q}^{1/2}\boldsymbol{PQ}^{1/2}\boldsymbol{Q}^{1/2} \tag{3.14}$$

is similar to the symmetric matrix $\boldsymbol{Q}^{1/2}\boldsymbol{PQ}^{1/2}$. Hence $\boldsymbol{PQ}$ has only real eigenvalues. If $\boldsymbol{P}$ is nonsingular, then so is $\boldsymbol{PQ}$. Hence zero is not an eigenvalue of $\boldsymbol{PQ}$. Since all eigenvalues of $\boldsymbol{PQ}$ are real, this implies that $\boldsymbol{PQ}$ is hyperbolic.

Proposition 3.4: For all j sufficiently large, the solution of (2.13) in the ball $B(\boldsymbol{p}_k, \delta)$ is a hyperbolic equilibrium of network (3.1) with $\lambda = \lambda_j$, and it is unstable.

Remarks: The proposition states that the equilibria that cluster in the interior of $[0,1]^n$ are hyperbolic and unstable for λ_j sufficiently large.

Proof: For clarity, hereafter we drop the subscript j on λ_j and say "for sufficiently large λ" to mean "for sufficiently large index j." In the same way, we drop the subscript k on $\boldsymbol{p}_k$. Thus the analysis below holds for a fixed value $\lambda = \lambda_j$, which can be made arbitrarily large as desired and in the vicinity of a fixed cluster point $\boldsymbol{p}_k$.

For convenience, define

$$\boldsymbol{H}(\boldsymbol{x}) = \nabla^2 E(\boldsymbol{x}) \tag{3.15}$$

to be the Hessian matrix of E. Thus

$$[\boldsymbol{H}(\boldsymbol{x})]_{ij} = \frac{\partial^2 E}{\partial x_i \partial x_j}. \tag{3.16}$$

Observe that $\boldsymbol{H}(\boldsymbol{x})$ is always a symmetric matrix. Moreover, if E is a multilinear polynomial, then

$$h_{ii}(\boldsymbol{x}) = 0 \text{ for all } i. \tag{3.17}$$

Suppose $\boldsymbol{x}_0 \in B(\boldsymbol{p}, \delta)$ is an equilibrium (in the $\boldsymbol{x}$-space) of network (3.1), for the fixed value of λ. Then $\lambda^{-1}\boldsymbol{g}^{-1}(\boldsymbol{x}_0)$ is an equilibrium of (3.1) in the $\boldsymbol{u}$-space. To determine the stability status of this equilibrium, we make two transformations of variables. First, instead of working in the $\boldsymbol{u}$-space, we work in the $\bar{\boldsymbol{u}} := \lambda \boldsymbol{u}$ space. Second, we change the time variable from t to $\tau = t/\lambda$. This reflects the fact that, as $\lambda \to \infty$, the time scale gets correspondingly foreshortened. Now (3.1) leads to

$$\frac{d(\lambda \boldsymbol{u})}{dt} = -\boldsymbol{A}^{-1}(\lambda \boldsymbol{u}) - \lambda \nabla E(\boldsymbol{x}) + \lambda \boldsymbol{b}, \quad \boldsymbol{x} = \boldsymbol{g}(\lambda \boldsymbol{u}), \tag{3.18}$$

$$\frac{d\bar{\boldsymbol{u}}}{d\tau} = -\lambda^{-1}\boldsymbol{A}^{-1}\bar{\boldsymbol{u}} - \nabla E[\boldsymbol{g}(\bar{\boldsymbol{u}})] + \boldsymbol{b}. \tag{3.19}$$

By assumption, this system has an equilibrium (in the $\bar{\boldsymbol{u}}$-space) at $\boldsymbol{d} := \boldsymbol{g}^{-1}(\boldsymbol{x}_0)$. To determine the stability status of this equilibrium, we linearize (2.13) around $\bar{\boldsymbol{u}} = \boldsymbol{d}$. This gives the linearization matrix

$$\begin{aligned}\boldsymbol{M} &= \frac{d}{d\bar{\boldsymbol{u}}}\{-\lambda^{-1}\boldsymbol{A}^{-1}\bar{\boldsymbol{u}} - \nabla E[\boldsymbol{g}(\bar{\boldsymbol{u}})]\}_{\bar{\boldsymbol{u}}=\boldsymbol{d}} \\ &= -\lambda^{-1}\boldsymbol{A}^{-1} - \nabla^2 E[\boldsymbol{g}(\boldsymbol{d})]\boldsymbol{J}(\boldsymbol{d})\end{aligned} \tag{3.20}$$

where $\boldsymbol{J}(\bar{\boldsymbol{u}}) = \nabla \boldsymbol{g}(\bar{\boldsymbol{u}})$ is the Jacobian matrix of the map $\boldsymbol{g}$. Now $\boldsymbol{M}$ is in reality a function of λ. As $\lambda \to \infty, \boldsymbol{g}(\boldsymbol{d}) = \boldsymbol{x}_0 \to \boldsymbol{p}$, while $\boldsymbol{d} \to \boldsymbol{g}^{-1}(\boldsymbol{p})$. Of course, $\lambda^{-1}\boldsymbol{A}^{-1} \to \boldsymbol{0}$. Hence

$$\boldsymbol{M} = -\boldsymbol{H}(\boldsymbol{p})\boldsymbol{J}[\boldsymbol{g}^{-1}(\boldsymbol{p})] + O(\lambda^{-1}) \tag{3.21}$$

where $\boldsymbol{H} = \nabla^2 E$.

Now recall that, by assumption, $\boldsymbol{b}$ is a regular value of the map ∇E. Hence $\boldsymbol{H}(\boldsymbol{p})$ is nonsingular. Since the Jacobian matrix $\boldsymbol{J}$ is positive definite, it follows from Proposition 3.3 that the product

$$\boldsymbol{M}_0 = -\boldsymbol{H}(\boldsymbol{p})\boldsymbol{J}[\boldsymbol{g}^{-1}(\boldsymbol{p})] \tag{3.22}$$

is hyperbolic. It is also independent of λ. Hence $\boldsymbol{M}$ is also hyperbolic for sufficiently large λ. This shows that the equilibrium $\boldsymbol{d}$ (in the $\bar{\boldsymbol{u}}$-space) is hyperbolic. The property of hyperbolicity is preserved under a diffeomorphic coordinate transformation, so the corresponding equilibria in the $\boldsymbol{u}$-space and $\boldsymbol{v}$-space are also hyperbolic.

To show that the equilibrium is unstable, observe that, because E is multilinear, all diagonal elements of $\boldsymbol{H}(\boldsymbol{p})$ are zero [cf., (3.17)]. The same is true of $\boldsymbol{M}_0$, since $\boldsymbol{J}$ is a diagonal matrix. Hence the trace of $\boldsymbol{M}_0$, equal to the sum of the eigenvalues of $\boldsymbol{M}_0$, is also zero. Since $\boldsymbol{M}_0$ is hyperbolic and has only real eigenvalues, it has at least one positive eigenvalue. Hence the same is true of $\boldsymbol{M}$ for sufficiently large λ. By the linearization theorem [28, Theorem 5.5.27], it follows that the equilibrium at $\boldsymbol{d}$ is unstable for large enough λ.

Note that the argument in the proof collapses if $\boldsymbol{H}(\boldsymbol{p})$ is singular. This is one reason for introducing the bias vector $\boldsymbol{b}$. Also, the analysis shows that, for almost all initial conditions $\boldsymbol{x}(0)$ near $\boldsymbol{x}_0$, the solution trajectory "flies away" from the equilibrium in the "fast" time scale $\tau = t/\lambda$—the bigger we make λ, the faster the trajectories move away from equilibria that cluster in the interior $(0, 1)^n$.

Proposition 3.5: For all i sufficiently large, the solution of (2.13) in the ball $B(\boldsymbol{e}, \delta)$ is an exponentially stable equilibrium of the neural network (3.1) with $\lambda = \lambda_j$.

Proof: We use the same notational simplifications as in the proof of Proposition 3.4 by denoting $\lambda_j, \boldsymbol{e}_k$ as λ and $\boldsymbol{e}$, respectively. Let $\boldsymbol{x}_0 \in B(\boldsymbol{e}, \delta)$ denote the equilibrium in the $\boldsymbol{x}$-space. Then we know from the proof of Proposition 2.4 that there is an equilibrium in the $\boldsymbol{u}$-space near

$$\boldsymbol{u}^* := \boldsymbol{A}[\boldsymbol{b} - \nabla E(\boldsymbol{e})]. \tag{3.23}$$

Let $\boldsymbol{u}_0 = \lambda^{-1}\boldsymbol{g}^{-1}(\boldsymbol{x}_0)$ denote the equilibrium of (3.1) in the $\boldsymbol{u}$-space. Then $\boldsymbol{u}_0 \to \boldsymbol{u}^*$ as $\lambda \to \infty$. Now linearize the right side of (3.1) at $\boldsymbol{u} = \boldsymbol{u}_0$. This gives the linearization matrix

$$\begin{aligned}\boldsymbol{M} &= \frac{d}{d\boldsymbol{u}}\{-\boldsymbol{A}^{-1}\boldsymbol{u} - \nabla E[\boldsymbol{g}(\lambda \boldsymbol{u})]\}_{\boldsymbol{u}=\boldsymbol{u}_0} \\ &= -\boldsymbol{A}^{-1} - \boldsymbol{H}[\boldsymbol{g}(\lambda \boldsymbol{u}_0)]\lambda \boldsymbol{J}(\lambda \boldsymbol{u}_0).\end{aligned} \tag{3.24}$$

Now $\boldsymbol{u}_0 \to \boldsymbol{u}^*$, which is a finite vector with all nonzero components. Hence, by Assumption A), $\|\lambda \boldsymbol{J}(\lambda \boldsymbol{u}_0)\| \to 0$, even allowing for the fact that $\boldsymbol{u}_0$ depends on λ. Of course, $\|\boldsymbol{H}[\boldsymbol{g}(\lambda \boldsymbol{u}_0)]\|$ is bounded. Hence

$$\boldsymbol{M} = -\boldsymbol{A}^{-1} + O(\lambda^{-1}). \tag{3.25}$$

Note that all eigenvalues of $\boldsymbol{A}^{-1}$ are negative; hence all eigenvalues of $\boldsymbol{M}$ have negative real parts for λ sufficiently large. It follows from the linearization theorem [28, Corollary 5.5.26] that the equilibrium at $\boldsymbol{x}_0$ is exponentially stable.

Proposition 3.6: For all j sufficiently large, the solution of (2.13) in the ball $B(\boldsymbol{q}_k, \delta)$ is a hyperbolic equilibrium of (3.1) with $\lambda = \lambda_j$, and it is unstable.

Proof: Once again, we replace $\lambda_j, \boldsymbol{q}_k$ by λ and $\boldsymbol{q}$, respectively. Also, for notational convenience, suppose the vector $\boldsymbol{q}$ belongs to the face

$$\begin{aligned}\{\boldsymbol{v} \in [0, 1]^n \colon & v_i \in (0, 1) \text{ for } i = 1, \cdots, k, \\ & v_i = 0 \text{ or } 1 \text{ for } i = k+1, \cdots, n\}.\end{aligned} \tag{3.26}$$

Once the vector $\boldsymbol{q}$ is fixed, this assumption can always be satisfied by renumbering the neurons if necessary. Define

$$\begin{aligned}\boldsymbol{q}_1 &= [q_1 \cdots q_k]^t \in (0, 1)^k, \\ \boldsymbol{q}_2 &= [q_{k+1} \cdots q_n]^t \in \{0, 1\}^{n-k}\end{aligned} \tag{3.27}$$

and partition all other vectors commensurately. Thus the subscript "1" denotes the first k components of a vector and the subscript "2" denotes the last $n - k$ components of a vector. Matrices are also partitioned commensurately, with the obvious meanings assigned to the subscripts 11–22.

Let $\boldsymbol{x}_0 \in B(\boldsymbol{q}, \delta)$ denote the equilibrium whose stability status is to be determined. We define the auxiliary variable

$$\bar{\boldsymbol{u}} = \begin{bmatrix} \lambda \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{bmatrix} = \Lambda \boldsymbol{u} \tag{3.28}$$

where

$$\Lambda = \begin{bmatrix} \lambda I_k & \boldsymbol{0} \\ \boldsymbol{0} & I_{n-k} \end{bmatrix}. \tag{3.29}$$

Also, we study the vector

$$\begin{bmatrix} \lambda^{-1} & d\bar{\boldsymbol{u}}_1/dt \\ & d\bar{\boldsymbol{u}}_2/dt \end{bmatrix} = \Lambda^{-1}\frac{d\bar{\boldsymbol{u}}}{dt}. \tag{3.30}$$

It turns out that this vector is well behaved as $\lambda \to \infty$, suggesting that the trajectories of the neural network in the vicinity of the equilibrium clustering in the faces of $[0,1]^n$ exhibit two time scale behavior. Now (3.1) leads to

$$\frac{d\bar{\boldsymbol{u}}}{dt} = \Lambda\dot{\boldsymbol{u}} = -\boldsymbol{A}^{-1}\bar{\boldsymbol{u}} - \Lambda\nabla E[\boldsymbol{g}(\lambda\Lambda^{-1}\bar{\boldsymbol{u}})] + \Lambda\boldsymbol{b} \tag{3.31}$$

where we take advantage of the fact that Λ and $\boldsymbol{A}^{-1}$ commute, both being diagonal matrices. Hence

$$\Lambda^{-1}\frac{d\bar{\boldsymbol{u}}}{dt} = -\Lambda^{-1}\boldsymbol{A}^{-1}\bar{\boldsymbol{u}} - \nabla E[\boldsymbol{g}(\lambda\Lambda^{-1}\bar{\boldsymbol{u}})] + \boldsymbol{b}. \tag{3.32}$$

Let us linearize the right side of (3.32) around the equilibrium in the $\bar{\boldsymbol{u}}$-space. This equilibrium is at

$$\bar{\boldsymbol{u}}_0 = \begin{bmatrix} \boldsymbol{g}_1^{-1}(\boldsymbol{x}_{10}) \\ \boldsymbol{A}_2\{\boldsymbol{b}_2 - [\nabla E(\boldsymbol{x}_0)]_2\} \end{bmatrix}. \tag{3.33}$$

This can be seen by analogy with the proofs of Propositions 3.4 and 3.5. Hence the linearization matrix is

$$\begin{aligned} \boldsymbol{M} &= \frac{d}{d\bar{\boldsymbol{u}}}\{-\Lambda^{-1}\boldsymbol{A}^{-1}\bar{\boldsymbol{u}} - \nabla E[\boldsymbol{g}(\lambda\Lambda^{-1}\bar{\boldsymbol{u}})]\}_{\bar{\boldsymbol{u}}=\bar{\boldsymbol{u}}_0} \\ &= -\Lambda^{-1}\boldsymbol{A}^{-1} - \nabla^2 E[\boldsymbol{g}(\lambda\Lambda^{-1}\bar{\boldsymbol{u}}_0)]\lambda\Lambda^{-1}\boldsymbol{J}(\lambda\Lambda^{-1}\bar{\boldsymbol{u}}_0). \end{aligned} \tag{3.34}$$

Now note that

$$\lambda\Lambda^{-1} = \begin{bmatrix} I_k & \boldsymbol{0} \\ \boldsymbol{0} & \lambda I_{n-k} \end{bmatrix}. \tag{3.35}$$

Also, by assumption

$$\boldsymbol{g}(\lambda\Lambda^{-1}\bar{\boldsymbol{u}}_0) = \boldsymbol{x}_0 = \boldsymbol{q} + \boldsymbol{o}(\lambda^{-1}). \tag{3.36}$$

Therefore

$$\nabla^2 E[\boldsymbol{g}(\lambda\Lambda^{-1}\bar{\boldsymbol{u}}_0)] = \boldsymbol{H}(\boldsymbol{q}) + \boldsymbol{O}(\lambda^{-1}) \tag{3.37}$$

where $\boldsymbol{H} = \nabla E$. Next, by Assumption A)

$$\begin{aligned} \boldsymbol{J}(\lambda\Lambda^{-1}\bar{\boldsymbol{u}}_0) &= \begin{bmatrix} \boldsymbol{J}_{11}(\bar{\boldsymbol{u}}_{01}) & \boldsymbol{0} \\ \boldsymbol{0} & \lambda\boldsymbol{J}_{22}(\lambda\bar{\boldsymbol{u}}_{20}) \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{J}_{11}(\boldsymbol{d}_1) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \boldsymbol{O}(\lambda^{-1}) \end{aligned} \tag{3.38}$$

where

$$\boldsymbol{d}_1 = \boldsymbol{g}_1^{-1}(\boldsymbol{q}_1) \in \Re^k. \tag{3.39}$$

Substituting from (3.37) and (3.38) into (3.34) leads to

$$\begin{aligned} \boldsymbol{M} &= -\begin{bmatrix} \lambda^{-1}\boldsymbol{A}_{11}^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{A}_{22}^{-1} \end{bmatrix} - \begin{bmatrix} \boldsymbol{H}_{11}(\boldsymbol{q}) & \boldsymbol{H}_{12}(\boldsymbol{q}) \\ \boldsymbol{H}_{21}(\boldsymbol{q}) & \boldsymbol{H}_{22}(\boldsymbol{q}) \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} \boldsymbol{J}_{11}(\boldsymbol{d}_1) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} + \boldsymbol{O}(\lambda^{-1}) \\ &= -\begin{bmatrix} \boldsymbol{H}_{11}(\boldsymbol{q})\boldsymbol{J}_{11}(\boldsymbol{d}_1) & \boldsymbol{0} \\ \boldsymbol{H}_{21}(\boldsymbol{q})\boldsymbol{J}_{11}(\boldsymbol{d}_1) & \boldsymbol{A}_{22}^{-1} \end{bmatrix} + \boldsymbol{O}(\lambda^{-1}) \end{aligned} \tag{3.40}$$

where the matrix $\lambda^{-1}\boldsymbol{A}_{11}^{-1}$ is absorbed into the $\boldsymbol{O}(\lambda^{-1})$ term. Now define

$$\boldsymbol{M}_0 = -\begin{bmatrix} \boldsymbol{H}_{11}(\boldsymbol{q})\boldsymbol{J}_{11}(\boldsymbol{d}_1) & \boldsymbol{0} \\ \boldsymbol{H}_{21}(\boldsymbol{q})\boldsymbol{J}_{11}(\boldsymbol{d}_1) & \boldsymbol{A}_{22}^{-1} \end{bmatrix}. \tag{3.41}$$

This matrix is independent of λ. The set of eigenvalues of $\boldsymbol{M}_0$, denoted by spec $(\boldsymbol{M}_0)$, is given by

$$\text{spec}(\boldsymbol{M}_0) = \text{spec}[-\boldsymbol{H}_{11}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{d}_1)] \bigcup \text{spec}(-\boldsymbol{A}_{22}^{-1}) \tag{3.42}$$

because $\boldsymbol{M}_0$ is block triangular. By assumption (see Hypothesis 4 of Theorem 2.6), $\boldsymbol{q}_1$ is a regular value of the map $\boldsymbol{x}_1 \mapsto \nabla E(\boldsymbol{x}_1, \boldsymbol{q}_2)$. Hence $\boldsymbol{H}_{11}(\boldsymbol{q})$ is nonsingular. By Proposition 3.3, the product $-\boldsymbol{H}_{11}(\boldsymbol{q}\boldsymbol{J}_{11}(\boldsymbol{d}_1)$ is hyperbolic. Clearly so is the matrix $\boldsymbol{A}_{22}^{-1}$. Hence $\boldsymbol{M}_0$ is also hyperbolic. Since $\boldsymbol{M}_0$ is independent of λ and $\boldsymbol{M} = \boldsymbol{M}_0 + \boldsymbol{O}(\lambda^{-1})$, it follows that $\boldsymbol{M}$ is also hyperbolic for sufficiently large λ. Hence the equilibrium at $\boldsymbol{x}_0$ is hyperbolic.

To prove that $\boldsymbol{x}_0$ is an unstable equilibrium, we proceed exactly as in the proof of Proposition 3.4. The matrix $\boldsymbol{H}_{11}(\boldsymbol{q})$ has zero diagonal elements; hence the trace of $-\boldsymbol{H}_{11}(\boldsymbol{q})\boldsymbol{J}_{11}(\boldsymbol{d}_1)$ is also zero. Since the matrix is hyperbolic and has only real eigenvalues, it follows that at least one eigenvalue is positive. Hence $\boldsymbol{x}_0$ is an unstable equilibrium.

The proof of Proposition 3.6 sheds some light on the behavior of the solution trajectories near the "face" equilibria. Let $\boldsymbol{x}_0 = \boldsymbol{x}_0(\lambda)$ denote the equilibrium of (3.1) that approaches the "face" vector $\boldsymbol{q}$. If we define the new variable $\bar{\boldsymbol{u}}$ as in (3.28) and linearize around the equilibrium $\boldsymbol{u}_0$ (with the partitioning of the time variable into a fast and a slow time scale, as in (3.30)), then the linearization matrix approaches the block-triangular matrix $\boldsymbol{M}_0$. Note that the set $\mathcal{M} = \{\boldsymbol{u} \in \Re^n : \boldsymbol{u}_2 = \boldsymbol{0}\}$ is invariant under $\boldsymbol{M}_0$; that is, $\boldsymbol{u} \in \mathcal{M}$ implies that $\boldsymbol{M}_0\boldsymbol{u} \in \mathcal{M}$. Hence, by standard arguments in the theory of singular perturbations (see e.g., [8] and [17]), there exists, for each sufficiently large λ, an $(n-k)$-dimensional manifold M_λ containing $\boldsymbol{x}_0$ that is invariant under the flow of the differential equation (3.1). Moreover, this invariant manifold M_λ "approaches" $\mathcal{M}$ as $\lambda \to \infty$. If we examine the vector field defined by (3.1) when restricted to the lower-dimensional invariant manifold M_λ, then $\boldsymbol{x}_{20}$ is an equilibrium of this restricted vector field. Moreover, if this restricted vector field is linearized around the equilibrium $\boldsymbol{x}_{20}$, then the resulting matrix equals $-\boldsymbol{A}_{22}^{-1} + \boldsymbol{O}(\lambda^{-1})$ [see (3.42)]. On the other hand, the matrix $-\boldsymbol{H}_{11}(\boldsymbol{q})\boldsymbol{J}_{11}(\boldsymbol{d}_1)$ has at least one positive eigenvalue. Hence, at least one of the components of the "fast" variables "flies away" from the equilibrium $\boldsymbol{x}_{10}$ in the fast time scale $\tau = t/\lambda$. Hence, for large enough λ, roughly speaking, the last $(n-k)$ components of $\boldsymbol{x}(t)$ converge exponentially fast in the normal time scale t to $\boldsymbol{x}_{20}$, while at least one of the first k components of $\boldsymbol{x}(t)$ flies away from $\boldsymbol{x}_{10}$ in the fast time scale τ.

At last we come to the proof of the main result.

Proof of Theorem 3.2: Propositions 3.4, 3.5, and 3.6 together show that, for almost all $\boldsymbol{b} \in \Re^n$:

i) The neural network has only a finite number of equilibria.

ii) All equilibria are hyperbolic.

iii) The equilibria can be grouped unambiguously into three classes, namely: Those approaching the interior of $[0, 1]^n$. those approaching the faces of $[0, 1]^n$. and those approaching the corners of $[0, 1]^n$.

iv) Of these, the equilibria in the first two classes are unstable, while those in the last class are exponentially stable.

Now, the stable manifold of each unstable equilibrium (in the $\boldsymbol{x}$-space) is a nowhere dense set because it is a manifold of dimension lower than n. Since there are only a finite number of such unstable equilibria, the union of the stable manifolds of all unstable equilibria is also a nowhere dense set. Let V denote this union. Since the network is totally stable by Proposition 3.1, it follows that, whenever $\boldsymbol{x}(0) \notin V$, the resulting solution trajectory must converge to one of the "corner" equilibria, i.e., to a vector of the form $\boldsymbol{e} + \boldsymbol{o}(\lambda^{-1})$, where $\boldsymbol{e} \in \{0, 1\}^n$.

It only remains to show that, whenever the norm $\|\boldsymbol{b}\|$ of the bias vector is sufficiently small, each such corner $\boldsymbol{e}$ is a local minimum of the objective function E. For this purpose, define

$$E_{\boldsymbol{b}}(\boldsymbol{x}) = E(\boldsymbol{x}) - \boldsymbol{b}^t\boldsymbol{x}. \tag{3.43}$$

Then $E_{\boldsymbol{b}}$ is also a multilinear polynomial. Now observe that

$$\nabla E_{\boldsymbol{b}}(\boldsymbol{x}) = \nabla E(\boldsymbol{x}) - \boldsymbol{b}. \tag{3.44}$$

By Proposition 2.4, $\boldsymbol{e}$ is a cluster point of S if and only if it satisfies parity condition (2.24) with $\boldsymbol{f}(\boldsymbol{x}) = \nabla E(\boldsymbol{x})$; that is

$$[\nabla E(\boldsymbol{x}) - \boldsymbol{b}]_i > 0 \text{ if } e_i = 0, < 0 \text{ if } e_i = 1. \tag{3.45}$$

From Proposition A.2, $\boldsymbol{e}$ satisfies (3.45) if and only if it is a strict local minimum of the function $E_{\boldsymbol{b}}$. Thus the proof of Theorem 3.2 is complete if it can be shown that, whenever $\|\boldsymbol{b}\|$ is sufficiently small, every strict local minimum of $E_{\boldsymbol{b}}$ is also a local minimum of E.

We prove the contrapositive, namely: If $\boldsymbol{x} \in \{0, 1\}^n$ is not a local minimum of E. then there exists an $\epsilon > 0$ such that $\boldsymbol{x}$ is also not a (strict) local minimum of $E_{\boldsymbol{b}}$ whenever $\|\boldsymbol{b}\|_\infty < \epsilon$. Suppose $\boldsymbol{x}$ is not a local minimum of E. Then there exists a neighbor $\boldsymbol{y} \in N(\boldsymbol{x})$ such that $E(\boldsymbol{x}) > E(\boldsymbol{y})$. Choose a $\boldsymbol{y} \in N(\boldsymbol{x})$ such that $E(\boldsymbol{x}) - E(\boldsymbol{y})$ is positive, but as small as possible. Define $g(\boldsymbol{x}) = E(\boldsymbol{x}) - E(\boldsymbol{y})$. Then $g(\boldsymbol{x}) > 0$. Moreover, since there are only a finite number of vectors $\boldsymbol{x}$. there exists an $\epsilon > 0$ such that $\epsilon \le g(\boldsymbol{x})$ whenever $\boldsymbol{x}$ is not a local minimum of E. Now suppose $|b_i| < \epsilon$ for each i. Let $\boldsymbol{x}$ be any nonlocal minimum of E. and choose $\boldsymbol{y} \in N(\boldsymbol{x})$ such that

$$E(\boldsymbol{x}) - E(\boldsymbol{y}) = g(\boldsymbol{x}) > 0. \tag{3.46}$$

Suppose $\boldsymbol{x}$ and $\boldsymbol{y}$ differ only in the ith component. Then

$$\begin{aligned} E_{\boldsymbol{b}}(\boldsymbol{x}) - E_{\boldsymbol{b}}(\boldsymbol{y}) &= E(\boldsymbol{x}) - E(\boldsymbol{y}) - \boldsymbol{b}^t(\boldsymbol{x} - \boldsymbol{y}) \\ &\ge g(\boldsymbol{x}) - |b_i| > 0 \end{aligned} \tag{3.47}$$

since $g(\boldsymbol{x}) \ge \epsilon$ and $|b_i| < \epsilon$. Hence $\boldsymbol{x}$ is not a local minimum of $E_{\boldsymbol{b}}$. This completes the proof.

The above proof suggests a way of making the phrase "sufficiently small bias vector $\boldsymbol{b}$" in the statement of Theorem 3.2 quantitatively precise. Suppose the objective function E assumes only integer values (as is the case in the applications suggested in [21] and [5], for example). Then it is easy to see that if $\boldsymbol{x}$ is not a local minimum of E. then there exists an adjacent vector $\boldsymbol{y} \in N(\boldsymbol{x})$ such that $E(\boldsymbol{y}) \le E(\boldsymbol{x}) - 1$. In other words, $g(\boldsymbol{x}) \ge 1$ for all $\boldsymbol{x}$ that are not local minima. Thus, to apply Theorem 3.2 to objective functions that assume only integer values, one can use any bias vector $\boldsymbol{b}$ such that $|b_i| < 1$ for all i, and almost all such bias vectors will work. Now consider the case of a general polynomial objective function E whose coefficients are rational numbers; this assumption applies to all but the most contrived situations. Let r denote the least common multiple of the denominators of all the coefficients of E. Then clearly the modified objective function $rE(\boldsymbol{x})$ assumes only integer values, and as a result, we have that $rg(\boldsymbol{x}) \ge 1$ for all $\boldsymbol{x}$ that are not local minima. Thus, in applying Theorem 3.2, we can choose any bias vector such that $|b_i| < 1/r$ for all i.

IV. Necessity of the Bias Vector

Neural network (3.1) has two distinctive features when compared with earlier works, including [27]. First, the sigmoidal gain λ is made to go to infinity via a sequence of values, rather than through a continuum of values. This is purely a technical device to enable the application of Sard's theorem. In practice, one would run network (3.1) with a particular value of λ, and if the results are not satisfactory, run it again with a higher value of λ, and so on. Thus, in practice, λ does indeed assume only a sequence of values. In fact, after a finite number of trials, one is guaranteed success, so actually λ assumes only a finite set of values. Hence the restriction on λ is quite justifiable. The second distinctive feature of network (3.1) is the introduction of the bias vector $\boldsymbol{b}$. Theorem 3.2 holds for "almost all" bias vectors $\boldsymbol{b}$, but the "natural choice" $\boldsymbol{b} = \boldsymbol{0}$ may not work, because it may belong to the exceptional set. This is illustrated through two simple examples.

Example 4.1: Suppose $n = 3$, and consider the problem of minimizing

$$E(\boldsymbol{x}) = x_1x_2x_3 \tag{4.1}$$

as $\boldsymbol{x}$ varies over $\{0, 1\}^3$. It is obvious that $E(\boldsymbol{x}) = 1$ if $\boldsymbol{x} = [1 \;\; 1 \;\; 1]^t$ and equals zero otherwise. Hence the function E has seven local minima. The trouble is that none of them is a strict local minimum.

Suppose we set up network (3.1) with $\boldsymbol{b} = \boldsymbol{0}$ and look at the cluster points of the resulting set of equilibria. Formally letting $\lambda_i \to \infty$ in (2.17) gives

$$\nabla E(\boldsymbol{x}) = \boldsymbol{0} \tag{4.2}$$

as the equation to be satisfied by the cluster points in the interior of $[0, 1]^n$. This equation has the solution set

$$\{\boldsymbol{x}{:}x_1 = x_2 = 0\} \bigcup \{\boldsymbol{x}{:}x_1 = x_3 = 0\} \bigcup \{\boldsymbol{x}{:}x_2 = x_3 = 0\} \tag{4.3}$$

which contains uncountably many points. All the analysis in the paper breaks down, because $\boldsymbol{b} = \boldsymbol{0}$ is a critical value of the map ∇E. One can also verify that none of the seven local minima of E satisfies parity condition (2.24), because the vector $\boldsymbol{z}$ contains at least one zero component in each case. This is because none of the seven local minima of E is a strict local minimum.

The introduction of the bias vector solves the problem. One can compute the number $g(\boldsymbol{x})$ introduced in the proof of Theorem 3.2. There is only one nonminimum, namely $\boldsymbol{x} = [1 \;\; 1 \;\; 1]^t$, and it is easy to see that $g(\boldsymbol{x}) = 1$. It is an easy exercise to verify that, if $\boldsymbol{b}$ is any vector with $|b_i| < 1$ for all i, then for each of the $2^3 = 8$ possible combinations of the signs of the vector $\boldsymbol{b}$, at least one of the seven local minima of E becomes a strict local minimum of $E_{\boldsymbol{b}}(\boldsymbol{x}) = E(\boldsymbol{x}) - \boldsymbol{b}^t\boldsymbol{x}$. Hence network (3.1) has an exponentially stable equilibrium near this binary vector. Also, the equation $\nabla E(\boldsymbol{x}) = \boldsymbol{b}$ has only a finite number of solutions (in most cases, zero) inside the set $(0, 1)^n$. Hence the introduction of the bias term greatly simplifies and clarifies the situation.

Example 4.2: Consider the Hopfield network (1.5) with $n = 3, C_i = R_i = 1$ for all i, and

$$\boldsymbol{W} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \theta = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{4.4}$$

This network is a special case of (3.1), with the bias vector $\boldsymbol{b} = 0$. Since $\boldsymbol{W}$ is singular, the equation $\boldsymbol{Wx} = \boldsymbol{b}$ has an uncountable number of solutions if $\boldsymbol{b} = \boldsymbol{0}$, each of which is potentially a cluster point of the equilibria of (3.1). So the situation is very messy. On the other hand, the range of the matrix $\boldsymbol{W}$ has dimension two; so, for almost all $\boldsymbol{b} \in \Re^3$, the equation $\boldsymbol{Wx} = \boldsymbol{b}$ has no solution, and as a result, the equilibria of (3.1) have no cluster point in the interior $(0, 1)^n$. Hence, once again, the introduction of the bias vector simplifies the dynamics considerably.

In [27], the possibility of the equation $\boldsymbol{Wx} = \boldsymbol{b}$ having an infinite number of solutions is avoided by assuming that all principal submatrices of $\boldsymbol{W}$ of size 2×2 or larger are hyperbolic (and therefore nonsingular). Hence the arguments of [27] are technically sound. But the assumptions on the weight matrix $\boldsymbol{W}$ are really not necessary—it is enough to introduce a bias vector. In contrast, no such assumptions on $\boldsymbol{W}$ are made in [15]. Hence, strictly speaking, the analysis in [15] is incorrect, though it is correct in spirit.

Example 4.3: In Proposition 4.4, the vector $\boldsymbol{b}$ is supposed to be chosen in such a way that no component of $\boldsymbol{z}$ is zero. One may be tempted to make the following conjecture: Suppose $\boldsymbol{e} \in \{0, 1\}^n$, and define $\boldsymbol{z} = \boldsymbol{b} - \boldsymbol{f}(\boldsymbol{e})$. Then $\boldsymbol{e}$ is a cluster point of $\mathcal{S}$ if and only if $\boldsymbol{z}$ satisfies the "relaxed" parity condition

$$z_i \leq 0 \text{ if } e_i = 0, \quad z_i \geq 0 \text{ if } e_i = 1. \tag{4.5}$$

But this conjecture is false. The objective of this example is to show that, if some component of $\boldsymbol{z}$ equals zero and $\boldsymbol{z}$ satisfies (4.5), then $\boldsymbol{e}$ may or may not be a cluster point of $\mathcal{S}$.

Suppose $n = 1$ (a single neuron), and let

$$f(x) = x(x - 1). \tag{4.6}$$

Suppose $g(\cdot)$ is a sigmoidal nonlinearity. A plot of $-f[g(\lambda u)]$ versus u is shown in Fig. 1 for increasing values of λ. Suppose the bias b is set equal to zero. Then both $e = 0$ and $e = 1$ satisfy the relaxed parity condition (4.5). But it turns out that $e = 0$ is not a cluster point of $\mathcal{S}$, while $e = 1$ is a cluster point. Thus, if some components of $\boldsymbol{z} = \boldsymbol{b} - \boldsymbol{f}(\boldsymbol{e})$ are zero, then there is no simple way to determine whether or not $\boldsymbol{e}$ is a cluster point of $\mathcal{S}$.

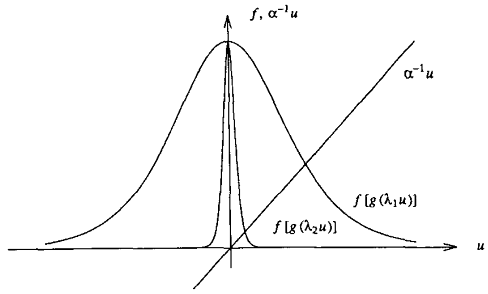

Fig. 1. Plot for Example 4.3; $\lambda_2 > \lambda_1$.

To prove the assertion in the preceding paragraph, observe that the equilibria of the network in the u-space are the solutions of

$$\alpha^{-1}u = -f[g(\lambda u)]. \tag{4.7}$$

Fig. 1 shows that, for each $\lambda > 0$, (4.7) has exactly one solution for u, call it u_λ. As $\lambda \to \infty, u_\lambda \to 0^+$ while $g(\lambda u) \to 1$. Thus $e = 1$ is a cluster point of $\mathcal{S}$ while $e = 0$ is not.

It is left to the reader to show that, if f is changed to

$$f(x) = x(1 - x) \tag{4.8}$$

then once again both $e = 0$ and $e = 1$ satisfy the relaxed parity condition (4.5). But now the situation is reversed: $e = 0$ is a cluster point of $\mathcal{S}$ while $e = 1$ is not.

The difficulty in trying to determine whether a given $\boldsymbol{e} \in \{0, 1\}^n$ is a cluster point when some component of $\boldsymbol{z}$ is zero is this: We can have a situation where some component of the equilibrium $\boldsymbol{u}_\lambda$ approaches zero, but does so more slowly than $1/\lambda$. In this case, the corresponding component of $\boldsymbol{g}(\lambda\boldsymbol{u})$ can approach zero or one even though the component of $\boldsymbol{u}$ itself approaches zero. The introduction of a bias term eliminates this difficulty, because for almost all $\boldsymbol{b}$, the vector $\boldsymbol{z}$ will have all nonzero components.

V. Connections to Interior-Point Methods of Optimization

In this section, a brief comparison is given of the neural network-based optimization methods discussed thus far with the so-called interior-point methods of linear and nonlinear programming. In any such comparison, the following point should be kept in mind: Nonlinear programming is a decades-old subject, and the recent interior-point methods (which are about a decade old) can be interpreted in terms of penalty function methods, which are nearly 30 years old. In contrast, the use of neural networks to perform combinatorial optimization is relatively new, and this paper is just about the first to give a rigorous analysis of the dynamical behavior of analog neural networks for optimization. Thus it would be unreasonable to expect that neural network-based methods would at once be competitive with nonlinear programming methods. Rather, the present paper can be said to open the door towards enhancing the competitiveness of neural network-based methods by presenting a very general set of analysis tools that can be applied to a very broad class of neural networks (see Section II). These tools are applied here to a particular network (3.1), but in principle future researchers could apply these analysis tools to other formulations as well. Using the results presented here, it should be possible to analyze the relative merits and demerits of neural network-based optimization methods vis-a-vis other methods. Of course, one cannot rule out the possibility that neural network-based methods will eventually turn out be demonstrably inferior to interior-point methods. If so, the present paper will have at least served to facilitate such a comparison.

A. Brief Summary of Interior-Point Methods for Linear Programming

The present generation of interior-point methods can be said to have been launched by the paper of Karmarkar [18], who proposed a new algorithm for linear programming. Since that time, more than 1000 papers have appeared on the subject. In particular, Renegar [22] has shown that Karmarkar's algorithm is closely related to the much older penalty function methods of Fiacco and McCormick [9], provided that i) the penalty function used is of the so-called "logarithmic barrier" type and ii) at each iteration, only a single Newton iteration is performed on the penalty-augmented objective function (as opposed to carrying out a complete minimization along a particular search direction). In this subsection, a brief review is attempted of interior-point methods with particular emphasis on those formulations that can be interpreted as dynamical systems.

Problem Formulation: The two most popular formulations of linear programming problems are as follows.

Canonical form: Given vectors $\boldsymbol{c} \in \Re^n, \boldsymbol{b} \in \Re^m$ and a matrix $A \in \Re^{m \times n}$, the problem is

$$\min \boldsymbol{c}^t\boldsymbol{x}, \text{ subject to} A\boldsymbol{x} = \boldsymbol{b}, x_i \geq 0 \text{ for all } i, \text{and} \quad (5.1)$$

$$\boldsymbol{e}^t\boldsymbol{x} = n, \text{ where } e = [1 \ 1 \cdots 1]^t \in \Re^n. \quad (5.2)$$

It is assumed that the choice $\boldsymbol{x} = \boldsymbol{e}$ is feasible. This form is well suited for the so-called projective scaling algorithms as exemplified by Karmarkar's original algorithm [18].

(Strict) Standard form: In this case the problem is given by (5.1), without the additional constraint (5.2). This is called the "standard" form. If, in addition, there exists a feasible point $\boldsymbol{x}$ with $x_i > 0$ for all i, then the problem is said to be in "strict standard" form. Note that the canonical form is in strict standard form, in view of the assumption that $\boldsymbol{x} = \boldsymbol{e}$ is feasible. This form is well suited for the so-called affine scaling algorithms; see, e.g., [1].

Both forms are equivalent for all practical purposes. The discussion below is in terms of the strict standard form, since it is closer to the optimization problem studied in Section III. Moreover, to avoid certain technical difficulties, it is further assumed that the feasible region is bounded and that the matrix A has full row rank (i.e., all constraints are independent). These assumptions are satisfied by the problem in Section III.

Affine Scaling Vector Field: To solve the problem at hand, one defines a vector field on the relative interior of the feasible region. To describe this vector field, a little notation is introduced:

1) Let $\Re^n_+$ denote the nonnegative orthant in $\Re^n$, and let $\Re^n_o$ denote its interior. Thus

$$\Re^n_+ = \{\boldsymbol{x} \in \Re^n: x_i \geq 0 \text{ for all } i\}, \text{ and}$$
$$\Re^n_o = \{\boldsymbol{x} \in \Re^n: x_i > 0 \text{ for all } i\}. \quad (5.3)$$

Clearly $\Re^n_o$ is an n-dimensional manifold.

2) Given any vector $\boldsymbol{x} \in \Re^n$, let X denote the corresponding diagonal matrix with the elements of $\boldsymbol{x}$ on the diagonal. Thus, if $\boldsymbol{x} = [x_1 \cdots x_n]^t$, then $X = \text{Diag}\{x_1, \cdots, x_n\}$.

3) Given a matrix $M \in \Re^{m \times n}$, let $\Pi^\perp(M)$ denote the orthogonal projection of $\Re^n$ onto the orthogonal complement of the range of M. Note that if M has full row rank (so that MM^t is nonsingular), then

$$\Pi^\perp(M) = I - M^t(MM^t)^{-1}M. \quad (5.4)$$

With this notation, the affine scaling vector field on $\Re^n_o$ is defined by (see, e.g., [2, (2.8)])

$$\begin{aligned}\boldsymbol{f}(\boldsymbol{x}) &= -X\Pi^\perp(AX)X\boldsymbol{c} \\ &= -X[I - XA^t(AX^2A^t)^{-1}AX]X\boldsymbol{c}. \quad (5.5)\end{aligned}$$

Approaches to Optimization: At this point, one can identify three distinct approaches to the solution of the linear programming problem.

a) Path-following approach: One solves the differential equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}[\boldsymbol{x}(t)], \boldsymbol{x}(0) = \boldsymbol{x}_0 \quad (5.6)$$

where the vector field $\boldsymbol{f}$ is defined in (5.5), and $\boldsymbol{x}_0 \in \Re^n_o$ is feasible. This approach is closest in spirit to the neural networks approach suggested in Section III. If $\boldsymbol{x}_0$ is chosen as the so-called "center" of the feasible region, then under mild conditions the resulting solution trajectory $\boldsymbol{x}(t)$ approaches the optimal face of the LP problem [3, Theorem 9.2]. For the class of optimization problems discussed in Section III, the "center" $\boldsymbol{x}_0$ equals $[0.5 \cdots 0.5]^t$.

b) Predictor-corrector approach: Unlike the path-following approach, this is an iterative method. At the initial iteration, $\boldsymbol{x} = \boldsymbol{x}_0$, and one defines a search direction $\boldsymbol{v}_0 := \boldsymbol{f}(\boldsymbol{x}_0)$, where $\boldsymbol{f}$ is the vector field of (5.5). Then one takes a "moderate-sized" step along the direction $\boldsymbol{v}_0$. This is the predictor step. In doing so, one of course deviates from the "true" solution trajectory of (5.6). Let α_0 denote the step size, and define $\bar{\boldsymbol{x}}_1 = \boldsymbol{x}_0 + \alpha_0\boldsymbol{v}_0$. The vector $\bar{\boldsymbol{x}}_1$ is then "corrected" to a nearby point $\boldsymbol{x}_1$ on the "true" solution trajectory of (5.6). One then defines the next search direction $\boldsymbol{v}_1 = \boldsymbol{f}(\boldsymbol{x}_1)$ and repeats the procedure. There are several rules of thumb for selecting the step size, but they all have the flavor of requiring that the inner product between successive search directions (computed in accordance with some appropriate Riemannian metric) be bounded away from zero. From a theoretical standpoint, the validity of this approach can be established only in relatively simple cases (see, e.g., [19, Theorem 2]).

c) Large step-size approach: In this approach, one identifies a suitable search direction and takes as large a step as possible while still remaining within the feasible region. Dikin's method [6] is of this type. To define the method precisely, let P denote the feasible region of the problem, i.e.,

$$P = \{\boldsymbol{x} \in \Re^n_+ \colon A\boldsymbol{x} = \boldsymbol{b}\}. \tag{5.7}$$

At iteration i, one defines the search direction $\boldsymbol{v}_i$ as $\boldsymbol{f}(\boldsymbol{x}_i)$, and

$$\lambda_i = \max\{\lambda \colon \boldsymbol{x}_i + \lambda \boldsymbol{v}_i \in P\}. \tag{5.8}$$

Thus λ_i denotes the size of the largest step that one can take while staying within P. Now one defines $\boldsymbol{x}_{i+1} = \boldsymbol{x}_i + 0.95\lambda_i \boldsymbol{v}_i$, where the factor of 0.95 ensures that $\boldsymbol{x}_{i+1}$ is an interior point.

B. Interior-Point Methods for Nonlinear Programming with Linear Constraints

Next, consider the problem

$$\min\ E(\boldsymbol{x}) \text{ subject to } A\boldsymbol{x} = \boldsymbol{b}, x_i \geq 0 \text{ for all } i. \tag{5.9}$$

Now the objective function is nonlinear, but the constraints are still linear. At this stage one can identify two distinct approaches.

Affine Scaling Approach: Here one merely replaces the vector $\boldsymbol{c}$ in (5.5) by the gradient $\nabla E(\boldsymbol{x})$ of the objective function. Thus one sets up the differential equation

$$\dot{\boldsymbol{x}} = -X\Pi^{\perp}(AX)X\nabla E(\boldsymbol{x}). \tag{5.10}$$

Faybusovich's Approach [7]: Here one eliminates the constraint $\boldsymbol{x} \in \Re^n_+$ by introducing new variables as follows

$$x_i = y_i^2, \qquad i = 1, \cdots, n. \tag{5.11}$$

Let us write the above equation in brief as $\boldsymbol{x} = \phi(\boldsymbol{y})$. This transformation gets rid of the constraint $\boldsymbol{x} \in \Re^n_+$, but in the process the linear constraints on $\boldsymbol{x}$ get transformed into the set of nonlinear constraints

$$A\phi(\boldsymbol{y}) = \boldsymbol{b}. \tag{5.12}$$

The idea is to set up a vector field in the $\boldsymbol{y}$ variables that is everywhere tangent to the above constraint surface. For this purpose, one defines

$$M(\boldsymbol{y}) = [Y\boldsymbol{a}_1^t \cdots Y\boldsymbol{a}_m^t] \in \Re^{n\times m} \tag{5.13}$$

where $Y = \text{Diag}\,\{y_1, \cdots, y_n\}$ (compare the definition of X) and $\boldsymbol{a}_1, \cdots, \boldsymbol{a}_m$ are the rows of the matrix A. Then one defines the projection matrix

$$\Pi(\boldsymbol{y}) = I - M(\boldsymbol{y})[M^t(\boldsymbol{y})M(\boldsymbol{y})]^{-1}M^t(\boldsymbol{y}). \tag{5.14}$$

This matrix projects a vector $\boldsymbol{z} \in \Re^n$ onto its tangent (at $\boldsymbol{y}$) to the surface defined by (5.12). Now the objective function $E(\boldsymbol{x})$ also gets transformed into $\overline{E}(\boldsymbol{y}) := E[\phi(\boldsymbol{y})]$. Moreover

$$\nabla_{\boldsymbol{y}}\overline{E}(\boldsymbol{y}) = Y\nabla_{\boldsymbol{x}}E[\phi(\boldsymbol{y})]. \tag{5.15}$$

With this background, one defines the vector field

$$\boldsymbol{g}(\boldsymbol{y}) = -\Pi(\boldsymbol{y})Y\nabla_{\boldsymbol{x}}E[\phi(\boldsymbol{y})]. \tag{5.16}$$

It is also possible to retransform this vector field back into an equivalent vector field in the original $\boldsymbol{x}$ (see [7, (19)]), but this is not discussed here.

C. Applications to Function Minimization over $\{0,1\}^n$

Now we briefly discuss how the above ideas may be applied to the discrete optimization problem

$$\min\ E(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in \{0,1\}^n. \tag{5.17}$$

The first step is to consider the continuous relaxation

$$\min\ E(\boldsymbol{x}) \text{ subject to } \boldsymbol{x} \in [0,1]^n. \tag{5.18}$$

If a solution to (5.18) actually belongs to $\{0,1\}^n$, then we are done. In general, this is not very likely. The discussion in Section III, however, implies the following fact: Consider the augmented objective function

$$E_{\boldsymbol{b}}(\boldsymbol{x}) = E(\boldsymbol{x}) + \boldsymbol{b}^t\boldsymbol{x}. \tag{5.19}$$

Proposition 5.1: Suppose E is a multilinear polynomial. Then:

1) For all bias vectors $\boldsymbol{b}$ except those belonging to a set of measure zero, the function $E_{\boldsymbol{b}}$ is a Morse function at each of its stationary points in $(0,1)^n$. In other words, whenever $\boldsymbol{x} \in (0,1)^n$ satisfies $\nabla E_{\boldsymbol{b}}(\boldsymbol{x}) = \boldsymbol{0}$, it is true that $\nabla^2 E_{\boldsymbol{b}}(\boldsymbol{x})$ is nonsingular. Moreover, the trace of $\nabla^2 E_{\boldsymbol{b}}(\boldsymbol{x})$ equals zero. Hence $E_{\boldsymbol{b}}$ does not have any local minima in $(0,1)^n$. Similarly, it does not have any local minima in the faces of $[0,1]^n$. Hence any local minima of $E_{\boldsymbol{b}}$ over $[0,1]^n$ are the same as its local minima over $\{0,1\}^n$.
2) For all sufficiently small bias vectors $\boldsymbol{b}$ except those belong to a set of measure zero, every local minimum over $\{0,1\}^n$ of $E_{\boldsymbol{b}}$ is also a local minimum of E.

This proposition means that, for almost all sufficiently small bias vectors $\boldsymbol{b}$, original discrete optimization problem (5.17) can be solved by solving instead the continuous relaxation (5.18) (with E replaced by $E_{\boldsymbol{b}}$).

With this technicality out of the way, we can now focus on how the interior-point methods of Section V-B can be used to solve the continuous relaxation problem (5.18). In the discussion that follows, the objective function E should be understood as the augmented objective function of (5.19). The subscript $\boldsymbol{b}$ is omitted, however, to facilitate a comparison with the contents of Section V-B.

The constraints in (5.18) can be expressed as

$$x_i \geq 0, -x_i \geq -1, \qquad i = 1, \cdots, n. \tag{5.20}$$

These inequality constraints can be converted to equality constraints via the introduction of "slack" variables, as is by now standard. The details of these manipulations are omitted, as they are straightforward.

Affine-Scaling Vector Field: Here one can use the contents of [3, Section VIII]. The "logarithmic barrier function" of [3, (8.6)] becomes[3]

$$L(\boldsymbol{x}) = -\sum_{i=1}^{n} [\log\, x_i + \log(1 - x_i)]. \tag{5.21}$$

[3] Note that the symbol f_H in the cited equation is changed to L. to avoid confusion.

The Hessian of L is diagonal, and

$$(\nabla^2 L)_{ii}(\boldsymbol{x}) = \frac{1}{x_i^2} + \frac{1}{(1-x_i)^2}, \qquad i = 1, \cdots, n. \qquad (5.22)$$

Thus the affine-scaling vector field becomes (cf., [3, (8.11)])

$$\dot{x}_i = \frac{1}{(\nabla^2 L)_{ii}(\boldsymbol{x})} \frac{\partial E}{\partial x_i} = \frac{x_i^2(1-x_i)^2}{x_i^2 + (1-x_i)^2} \frac{\partial E}{\partial x_i}. \qquad (5.23)$$

This is equivalent to (5.10) in that both systems have the same trajectory, though the parameterization of "time" might be different—see [3, Theorem 8.1].

Faybusovich Vector Field: Define

$$z_{2i-1} = x_i, \quad z_{2i} = 1 - x_i, \qquad i = 1, \cdots, n. \qquad (5.24)$$

Then the constraints in (5.20) become

$$\begin{aligned} z_i &\geq 0, & i &= 1, \cdots, 2n, \quad \text{and} \\ z_{2i-1} + z_{2i} &= 1, & i &= 1, \cdots, n. \end{aligned} \qquad (5.25)$$

Now let

$$z_i = y_i^2, \qquad i = 1, \cdots, n. \qquad (5.26)$$

Then the projection matrix of (5.14) is easily verified to be

$$\Pi(\boldsymbol{y}) = \text{Block Diag}\{\Pi_1, \cdots, \Pi_n\} \qquad (5.27)$$

where

$$\Pi_i = \begin{bmatrix} 1 - x_i & -\sqrt{x_i}\sqrt{1-x_i} \\ -\sqrt{x_i}\sqrt{1-x_i} & x_i \end{bmatrix}. \qquad (5.28)$$

Now, $\overline{E}(\boldsymbol{y}) = E(\boldsymbol{x})$ depends only on y_{2i-1} and not on y_{2i}. Hence (5.16) becomes (on noting that $y_{2i-1} = \sqrt{x_i}$)

$$\begin{aligned} \begin{bmatrix} \dot{y}_{2i-1} \\ \dot{y}_{2i} \end{bmatrix} &= -\Pi_i \begin{bmatrix} y_{2i-1}(\partial E/\partial x_i) \\ 0 \end{bmatrix} \\ &= -\begin{bmatrix} \sqrt{x_i}(1-x_i)(\partial E/\partial x_i) \\ -x_i\sqrt{1-x_i}(\partial E/\partial x_i) \end{bmatrix}. \end{aligned} \qquad (5.29)$$

One can reduce these to a set of n equations (not $2n$) by noting that

$$\dot{x}_i = 2y_{2i-1}\dot{y}_{2i-1} = -2x_i(1-x_i)\frac{\partial E}{\partial x_i}. \qquad (5.30)$$

D. Comparison with Proposed Neural Network

Note that both the vector fields (5.23) and (5.30) are well defined for all $\boldsymbol{x} \in \Re^n$, and not just when $\boldsymbol{x} \in [0,1]^n$. (Recall that E is a polynomial and hence analytic.) Also, it is clear that each of the 2^n vectors in $\{0,1\}^n$ is an equilibrium of both vector fields. This is a clear difference from the neural network proposed here, which has an equilibrium near a vector $\boldsymbol{e} \in \{0,1\}^n$ if and only if $\boldsymbol{e}$ is a local minimum of E over $\{0,1\}^n$. On the other hand, it is easy to verify that an equilibrium of (5.23) or (5.30) at $\boldsymbol{e} \in \{0,1\}^n$ is exponentially stable if and only if $\boldsymbol{e}$ satisfies parity condition (A.10), i.e., is a strict local minimum of E over $\{0,1\}^n$. This can be shown as follows: Both the vector fields (5.23) and (5.30) have the form

$$\dot{x}_i = h(x_i)\frac{\partial E}{\partial x_i} =: g_i(\boldsymbol{x}) \qquad (5.31)$$

where $h(\cdot)$ satisfies

$$h(0) = h(1) = 0, h'(0) > 0, h'(1) < 0. \qquad (5.32)$$

Now linearize (5.31) around an equilibrium $\boldsymbol{e} \in \{0,1\}^n$. Then

$$\frac{\partial g_i}{\partial x_i} = h'(x_i)\frac{\partial E}{\partial x_i} \qquad (5.33)$$

$$\frac{\partial g_i}{\partial x_j} = h(x_i)\frac{\partial^2 E}{\partial x_i \partial x_j} = 0 \text{ if } x_i \in \{0,1\}. \qquad (5.34)$$

Here we use the fact that $\partial^2 E/\partial x_i^2 = 0$ because E is multilinear. Thus the matrix obtained by linearizing (5.31) around $\boldsymbol{x} = \boldsymbol{e}$ is diagonal. Moreover, every diagonal element is negative (i.e., the equilibrium is exponentially stable) if and only if

$$[\nabla E(\boldsymbol{e})]_i > 0 \text{ if } e_i = 0, \quad [\nabla E(\boldsymbol{e})]_i < 0 \text{ if } e_i = 1. \qquad (5.35)$$

But this is precisely parity condition (A.10).

In summary, it can be stated that compared to the neural network proposed here, both (5.23) (the affine scaling vector field) and (5.30) (Faybusovich's vector field) have more spurious equilibria, but these spurious equilibria are not exponentially stable. It is desirable to make a deeper comparison involving the actual behavior of the time trajectories. Such a comparison can be a topic for future researchers.

VI. Conclusions

In this paper, we have shown that an analog neural network can be used to find a local minimum of any multilinear polynomial over the discrete set $\{0,1\}^n$.

Comparing the present paper with earlier results, we can claim the following advantages: First, the treatment here is comprehensive and rigorous. Second, it has been shown that it is necessary to introduce a so-called "bias" vector, which is not done in earlier work. Without the bias vector, some of the claims of the earlier papers may not be valid.

The results presented here guarantee only convergence to a local minimum. To build a network that is guaranteed to converge to a global minimum, a natural guess would be to mimic the simulated annealing algorithm, by replacing the deterministic and constant bias vector by a stochastic and time-varying bias vector, and slowly reduce the variance of the bias vector to zero. The analysis of such networks is bound to be extremely complicated and is left to those more competent than the author in such matters.

At the current state of evolution of the theory of neural computation, it is no longer enough to show that some problems can (also) be solved using a neural network—it is necessary to analyze the complexity of the computation. For instance, in [5] it is shown that the MAX-CUT problem (which is NP-complete) can be formulated as that of minimizing a quadratic objective function over $\{0,1\}^n$ and can thus be solved using a conventional discrete Hopfield network. Since a discrete Hopfield network merely changes one component of the state vector at a time, it is difficult to believe that "neural" computation based on such an approach can be superior to other algorithms for discrete optimization. Indeed, the results of [25] can be interpreted to mean that, in the worst case, a discrete-state Hopfield network can take an exponential number of time steps ($O(2^n)$) to reach even a local minimum. Note that, by simply enumerating $E(\boldsymbol{x})$ at all the 2^n points in $\{0,1\}^n$, one can in fact find a global minimum of E.

The contents of the present paper highlight the need to develop a theory of complexity for computation using differential equations. The vast majority of complexity theory is addressed to the computing ability of algorithms/machines and to the expressive ability of languages, though some beginnings have been made in a complexity theory for continuous computation [4]. But it appears that an entirely new approach is needed to analyze the complexity of computation using analog neural networks.

Appendix A
Multilinear Polynomials

Suppose $E\colon [0,1]^n \to \Re$ is a multilinear polynomial and that it is desired to minimize $E(\boldsymbol{x})$ as $\boldsymbol{x}$ varies over the discrete set $\{0,1\}^n$. The objective of the appendix is to show that there exist very simple necessary and sufficient conditions for a vector $\boldsymbol{x} \in \{0,1\}^n$ to be a local minimum of E. Thus, given a "candidate minimum" $\boldsymbol{x} \in \{0,1\}^n$, it is a very simple matter to determine whether or not $\boldsymbol{x}$ is indeed a local minimum. This justifies the analog approach to discrete optimization, whereby the aim is to generate a sequence of candidate minima, which are then checked for optimality.

Definition A.1: A vector $\boldsymbol{x} \in \{0,1\}^n$ is said to be a local minimum of the objective function E if

$$E(\boldsymbol{x}) \leq E(\boldsymbol{y}), \quad \text{for all } \boldsymbol{y} \in N(\boldsymbol{x}) \tag{A.1}$$

where $N(\boldsymbol{x})$ denotes the set of all vectors in $\{0,1\}^n$ lying at a Hamming distance of one from $\boldsymbol{x}$. $\boldsymbol{x}$ is said to be a strict local minimum of E if

$$E(\boldsymbol{x}) < E(\boldsymbol{y}), \quad \text{for all } \boldsymbol{y} \in N(\boldsymbol{x}). \tag{A.2}$$

We begin with an obvious observation. For each $\boldsymbol{x} \in \Re^n$ and each index $i \in \{1,\cdots,n\}$, let $\bar{\boldsymbol{x}}_i$ denote the $(n-1)$-dimensional vector obtained by omitting the ith component of $\boldsymbol{x}$; that is

$$\bar{\boldsymbol{x}} = [x_1 \cdots x_{i-1} x_{i+1} \cdots x_n]^t. \tag{A.3}$$

Suppose $E(\boldsymbol{x})$ is a multilinear polynomial. Then it is easy to see that there exist functions $\overline{E}_i\colon \Re^{n-1} \to \Re$ and $c_i\colon \Re^{n-1} \to \Re$ such that

$$E(\boldsymbol{x}) = x_i \overline{E}_i(\bar{\boldsymbol{x}}_i) + c_i(\bar{\boldsymbol{x}}_i). \tag{A.4}$$

This property proves useful below.

Proposition A.1: Suppose E is a multilinear polynomial on $\Re^n$ and that $\boldsymbol{x} \in \{0,1\}^n$. Then the following statements are equivalent:

1) $\boldsymbol{x}$ is a local minimum of E.
2) $\boldsymbol{x}$ satisfies the "relaxed parity condition," that is

$$[\nabla E(\boldsymbol{x})]_i \leq 0 \text{ if } x_i = 1, \quad [\nabla E(\boldsymbol{x})]_i \geq 0 \text{ if } x_i = 0. \tag{A.5}$$

Proof: "(2) ⇒ (1)" Suppose Statement 2 is true. Select an arbitrary index $i \in \{1,\cdots,n\}$, and define $\boldsymbol{y} \in \{0,1\}^n$ by

$$y_i = 1 - x_i, \quad y_j = x_j \text{ for all } j \neq i. \tag{A.6}$$

Then $\boldsymbol{y} \in N(\boldsymbol{x})$, and $\bar{\boldsymbol{y}}_i = \bar{\boldsymbol{x}}_i$, where $\bar{\boldsymbol{x}}_i$ is defined in (A.3). Hence, from (A.4), it follows that

$$E(\boldsymbol{y}) = y_i \overline{E}(\bar{\boldsymbol{x}}_i) + c_i(\bar{\boldsymbol{x}}_i) = (1 - x_i)\overline{E}(\bar{\boldsymbol{x}}_i) + c_i(\bar{\boldsymbol{x}}_i). \tag{A.7}$$

Therefore

$$E(\boldsymbol{y}) - E(\boldsymbol{x}) = (1 - 2x_i)\overline{E}(\bar{\boldsymbol{x}}_i). \tag{A.8}$$

Now observe from (A.4) that

$$\overline{E}(\bar{\boldsymbol{x}}_i) = \frac{\partial E}{\partial x_i} = [\nabla E(\boldsymbol{x})]_i. \tag{A.9}$$

Hence, if $\boldsymbol{x}$ satisfies parity condition (A.5), it follows from (A.8) that $E(\boldsymbol{y}) \geq E(\boldsymbol{x})$. Since the index i is arbitrary, the conclusion is that $\boldsymbol{x}$ is a local minimum.

"(1) ⇒ (2)" We show instead that if Statement 2 is false, then so is Statement 1. Suppose accordingly that the parity condition (A.5) is violated for some i. Define $\boldsymbol{y}$ as in (A.6); then (A.8) shows that $E(\boldsymbol{y}) < E(\boldsymbol{x})$, so that $\boldsymbol{x}$ is not a local minimum.

Proposition A.2: Suppose E is a multilinear polynomial on $\Re^n$ and that $\boldsymbol{x} \in \{0,1\}^n$. Then the following statements are equivalent:

1) $\boldsymbol{x}$ is a strict local minimum of E.
2) $\boldsymbol{x}$ satisfies the "parity condition," that is, no component of $\nabla E(\boldsymbol{x})$ is zero, and

$$[\nabla E(\boldsymbol{x})]_i < 0 \text{ if } x_i = 1, \quad [\nabla E(\boldsymbol{x})]_i > 0 \text{ if } x_i = 0. \tag{A.10}$$

The proof is a routine modification of that of Proposition A.1 and is left to the reader.

Appendix B
An Open Problem

In this appendix, we postulate a conjecture on the strict local minima of multilinear polynomials. Though the results of the present paper hold with or without this conjecture, it nevertheless represents an interesting problem in combinatorics.

Suppose E is a given multilinear polynomial, $\boldsymbol{e} \in \{0,1\}^n$, and $\boldsymbol{b}$ is the bias vector. Suppose $\boldsymbol{b}$ is chosen such that no component of the vector $\boldsymbol{z} = \boldsymbol{b} - \boldsymbol{f}(\boldsymbol{e})$ is zero. Comparing Propositions A.2 and 2.4 shows that $\boldsymbol{e}$ is a cluster point of $\mathcal{S}$ if and only if $\boldsymbol{e}$ is a strict local minimum of the modified function $E_{\boldsymbol{b}}(\boldsymbol{x}) = E(\boldsymbol{x}) - \boldsymbol{b}^t\boldsymbol{x}$.

It is easy to see that a multilinear polynomial need not have any strict local minima. An extreme example is provided by the function $E(\boldsymbol{x}) = x_1 x_2 \cdots x_n$, which has $2^n - 1$ local minima, none of them strict. Suppose $\boldsymbol{e}$ is a local minimum of E. It is easy to show that $\boldsymbol{e}$ is a strict local minimum of $E_{\boldsymbol{b}}$ for a suitable choice of the bias vector $\boldsymbol{b}$. To see this, observe from Proposition A.1 that $\nabla E(\boldsymbol{e})$ satisfies the relaxed parity condition (A.5). Suppose some components of $\nabla E(\boldsymbol{e})$ are zero, so that $\boldsymbol{e}$ is not a strict local minimum. If $\boldsymbol{b}$ is chosen such that $b_i < 0$ if $e_i = 0$ and $[\nabla E(\boldsymbol{e})]_i = 0$, and $b_i > 0$ if

$e_i = 1$ and $[\nabla E(\boldsymbol{e})]_i = 0$, then $\boldsymbol{e}$ satisfies the strict parity condition of Proposition A.2 for the function $E_{\boldsymbol{b}}$ and is thus a strict local minimum of $E_{\boldsymbol{b}}$, provided that $\|\boldsymbol{b}\|$ is sufficiently small. But the open question is the converse, namely: Given E, is it true that $E_{\boldsymbol{b}}$ has a strict local minimum for almost all $\boldsymbol{b}$ of sufficiently small norm?

The answer to the question as stated is yes. This follows from Theorem 3.2. For almost all sufficiently small $\boldsymbol{b}$, at least one $\boldsymbol{e} \in \{0, 1\}^n$ must satisfy the conditions of Proposition 2.4; otherwise the network (3.1) will have only unstable equilibria, which contradicts the fact that the network is totally stable. This proof, however, is very indirect and unsatisfactory. The question is basically combinatorial in nature, and the answer should therefore have a combinatorial proof.

Now we formulate a conjecture that goes slightly beyond the above question. For this purpose, some terminology is introduced. A chain in $\{0, 1\}^n$ is a sequence $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k\}$ such that $\boldsymbol{x}_{i+1} \in N(\boldsymbol{x}_i)$ for $i = 1, \cdots, k-1$.[4] A set $S \subseteq \{0, 1\}^n$ is connected if there is a chain between every pair of points in S. Suppose $M \subseteq \{0, 1\}^n$ is the set of local minima of E. Then M can be divided into its connected components, call them $M_1, \cdots, M_c$. Moreover, it is easy to see that E has the same value at all vectors in a particular connected component.

Conjecture: There exists a number $\epsilon > 0$, dependent on E, such that, whenever $\|\boldsymbol{b}\| < \epsilon$ and $b_i \neq 0$ for all i, at least one vector in M_j is a strict local minimum of $E_{\boldsymbol{b}}$, for $j = 1, \cdots, c$.

Acknowledgment

The author thanks Prof. L. Faybusovich for suggesting that a comparison be made between neural network based methods and interior-point methods of optimization and Prof. V. Chandru for helpful discussions on interior-point methods.

References

[1] E. R. Barnes, "A variation on Karmarkar's algorithm for solving linear programming problems," *Math. Programming,* vol. 36, pp. 174–182, 1986.

[2] D. A. Bayer and J. C. Lagarias, "The nonlinear geometry of linear programming, I: Affine and projective scaling trajectories," *Trans. Amer. Math. Soc,* vol. 314, no. 2, pp. 499–526, Aug. 1989.

[3] ——, "The nonlinear geometry of linear programming, II: Legendre transform coordinates and central trajectories," *Trans. Amer. Math. Soc,* vol. 314, no. 2, pp. 527–582, Aug. 1989.

[4] L. Blum, M. Shub, and S. Smale, "On a theory of computation and complexity over the real numbers: NP-Completeness, recursive functions and universal machines," *Bull. Amer. Math. Soc.,* vol. 21, pp. 1–46, 1989.

[5] J. Bruck and M. Blum, "Neural networks, error-correcting codes, and polynomials over the binary n-cube," *IEEE Trans. Infor. Theory.,* vol. 35, no. 5, pp. 976–987, Sep. 1989.

[6] I. Dikin, "Iterative solution of problems of linear and quadratic programming," *Soviet Mathematics Doklady,* vol. 8, pp. 674–675, 1967.

[7] L. Faybusovich, "Dynamical systems which solve optimization problems with linear constraints," *IMA J. Math. Contr. Infor.,* vol. 8, pp. 135–149, 1991.

[8] N. Fenichel, "Geometric singular perturbation theory for ordinary differential equations," *J. Diff. Eqns.,* vol. 3, pp. 53–98, 1979.

[9] A. V. Fiacco and G. W. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* New York: Wiley, 1968.

[10] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* New York: W. H. Freeman, 1979.

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

[12] M. W. Hirsch, "Convergence in neural nets," *Proc. Int. Joint Conf. Neural Networks,* vol. II, pp. 115–125, 1987.

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

[14] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational capabilities," *Proc. Nat'l. Acad. Sci. (U.S.A.),* vol. 79, pp. 2554–2558, 1982.

[15] ——, "Neurons with graded response have collective computational capabilities like those of two-state neurons," *Proc. Nat'l. Acad. Sci. (U.S.A.),* vol. 81, pp. 3088–3092, 1984.

[16] J. J. Hopfield and D. W. Tank, " 'Neural' computation of decision optimization problems," *Biological Cybernetics,* vol. 52, pp. 141–152, 1985.

[17] F. Hoppensteadt, "Properties of solutions of ordinary differential equations with small parameters," *Comm. Pure and Appl. Math.,* vol. 34, pp. 807–840, 1971.

[18] N. Karmarkar, "A new polynomial-time algorithm for linear programming," *Combinatorica,* vol. 4, no. 4, pp. 373–395, 1984.

[19] ——, "Riemannian geometry underlying interior-point methods for linear programming," *Contemporary Math.,* vol. 114, pp. 51–75, 1990.

[20] M. W. Krental, "On finding locally optimal solutions," *SIAM J. Computing,* vol. 19, pp. 742–749, 1990.

[21] C. L. Masti and M. Vidyasagar, "A stochastic high-order connectionist network for solving inferencing problems," in *Proc. Int. Joint. Conf. Neural Networks,* Singapore, Nov. 1991, pp. 911–916.

[22] J. Renegar, "A polynomial-time algorithm based on Newton's method for linear programming," *Math. Programming,* vol. 40, no. 1, pp. 59–94, 1988.

[23] F. M. A. Salam, Y.-W. Wang, and M.-R. Choi, "On the analysis of dynamic feedback neural networks," *IEEE Trans. Circuits Syst.,* vol. CAS-38, pp. 196–201, 1991.

[24] A. Sard, "The measure of critical values of differentiable maps," *Bull. Amer. Math. Soc.,* vol. 48, pp. 883–890, 1942.

[25] A. A. Schäffer, and M. Yannakakis, "Simple local search problems that are hard to solve," *SIAM J. Computing,* vol. 20, pp. 56–87, 1991.

[26] D. W. Tank and J. J. Hopfield, "Simple 'neural' optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit," *IEEE Trans. Circuits Syst.,* vol. CAS-33, pp. 533–541, 1986.

[27] M. Vidyasagar, "Location and stability of the high-gain equilibria of nonlinear neural networks," *IEEE Trans. Neural Networks,* vol. 4, no. 4, pp. 660–672, July 1993.

[28] ——, *Nonlinear Systems Analysis,* 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1993.

M. Vidyasagar (S'69–M'69–SM'78–F'83) was born in Guntur, Andhra Pradesh on September 29, 1947. He received the B.S., M.S., and Ph.D. degrees, all in electrical engineering, from the University of Wisconsin, in 1965, 1967, and 1969, respectively.

He taught at Marquette University, Milwaukee, WI, from 1969–1970, Concordia University, Montreal, Canada, from 1970–1980, and the University of Waterloo, Waterloo, Canada from 1980–1989. Since June 1989, he has been the Director of the Centre for Artificial Intelligence and Robotics (under the Defence Research and Development Organisation) in Bangalore, India. In addition to the above, he has held visiting positions at several universities including Massachusetts Institute of Technology, University of California at Berkeley, University of California at Los Angeles, C.N.R.S. Toulouse, France, Indian Institute of Science, University of Minnesota, and Tokyo Institute of Technology. He is the author or co-author of six books and more than 100 papers in archival journals.

Dr. Vidyasagar has received several honors in recognition of his research activities and is a Fellow of the Indian Academy of Sciences, the Indian National Science Academy, and the Indian National Academy of Engineering.

[4] Recall that $N(\boldsymbol{x})$ is the set of all vectors in $\{0, 1\}^n$ that are at a Hamming distance of one from $\boldsymbol{x}$, i.e., the set of nearest neighbors of $\boldsymbol{x}$.