Neural Networks, Orientations of the Hypercube, and Algebraic Threshold Functions

PIERRE BALDI

Abstract — **A class of possible generalizations of current neural networks models is described using local improvement algorithms and orientations of graphs. A notation of dynamical capacity is defined and, by computing bounds on the number of algebraic threshold functions, it is proven that for neural networks of size n and energy function of degree d, this capacity is $O(n^{d+1})$. Stable states are studied and it is shown that for the same networks the storage capacity is $O(n^{d-1})$. In the case of random orientations, it is proven that the expected number of stable states is exponential. Applications to coding theory are indicated and it is shown that usual codes can be embedded in neural networks but only at high cost. Cycles and their storage are also examined.**

I. Introduction

THE MAIN purpose of this paper is to present some results on the computational capabilities of a class of neural-type automata which are generalizations of the Hopfield model using nonquadratic energy functions. More precisely, we first describe the Hopfield model in a more general context of computations and dynamical systems. In Section II and based on this general context, we show how to construct several extensions of the original model and define a notion of dynamical capacity for a class of automata. In Section III we derive bounds on the number of threshold functions of degree d and on the dynamical capacity of the corresponding neural networks. In Section IV, we study stable states and storage capacities. In the last section we examine the questions of implementing codes and cycles in these networks.

There exist at least two very different approaches to computations. In mathematics these are in general "algebraic" maps between appropriate spaces, satisfying a set of formal axioms. For instance, the usual addition is a map from $\mathbb{R}^2$ to $\mathbb{R}$. However, in the physics of computation, computations are regarded as physical events occurring in some "hardware" according to the laws of nature. Very schematically, computations of all kinds (from digital to analog, from Boolean to arithmetic, from optimization to error correction, etc.) can be represented as trajectories in the space of states of a dynamical system. The dynamical system, in turn, can be emulated, and often at several levels, by different types of "hardware." This can be abstract hardware, in the sense of the evolution rule of an automaton or of a set of differential equations, or real hardware: abacus, silicon chip, brain, or even billiard table [10]! The introduction of trajectory, as an object *per se*, has the essential advantage of "decoupling" computations from hardware. For instance, given a class of computations, we can ask what is a corresponding reasonable set of trajectories. Or given a set of trajectories, what could they be good for computationally or what kind of natural or artificial system could generate them? This point of view has led to the creation of new algorithms, such as simulated annealing [13], where optimization strategies are embedded into the trajectories of statistical mechanical systems. These questions are also relevant in neurobiology, for instance, where trajectories can sometimes be recorded, without our complete understanding of the algorithmic level nor of all the details of the underlying biophysical phenomena. Computationally, two discrete automata with essentially the same trajectories are not distinguishable. Therefore, a notion of capacity for a class of such systems can be introduced based on the number of different dynamical behaviors they can exhibit.

Manuscript received July 9, 1986; revised June 9, 1987.

The author was with the California Institute of Technology, Pasadena, CA. He is now with the Department of Mathematics, University of California, San Diego, La Jolla, CA 92093.

IEEE Log Number 8821202.

In 1982, Hopfield [11] presented a model to attempt to understand the emergence of collective computational abilities in physical systems and networks of neurons. The original simple idea, and several of its derivations, have already been applied to a variety of contexts ranging, among others, from content-addressable memories and circuit architectures to learning algorithms and combinatorial optimization. In the original Hopfield discrete model a neural network consists of n pairwise interconnected devices or neurons, each one being in one of two possible states, $+1$ or -1. The state of the system is therefore represented by a vector $X = (x_1, \cdots, x_n)$ belonging to the hypercube H^n. The synaptic connections are described by a real symmetric matrix $\alpha = (\alpha_{ij})$ with the additional property: $\alpha_{ii} = 0$ for any i. Moreover, to each neuron i is associated a real threshold t_i. Randomly and asynchronously, each neuron changes its state according to the following rule. If x_i^+ is the state of neuron i after the corresponding updating step, then

$$x_i^+ = \operatorname{sgn}\left(\sum_{j=1}^{n} \alpha_{ij} x_j - t_i \right).$$

If the linear input minus the threshold is equal to zero, then $x_i^+ = x_i$.

This system possesses the fundamental property that, no matter what the starting state is and no matter in which fashion the neurons "decide" to update themselves, it will always converge to a stable state. The reason behind this key fact is the existence of an energy function for the

0018-9448/88/0500-0523$01.00 ©1988 IEEE

network, one which is decreasing when the algorithm is applied. If we consider the quadratic form

$$E(X) = -\sum_{i \le j} \alpha_{ij} x_i x_j + \sum_{i=1}^{n} t_i x_i,$$

then with the obvious notation we see that

$$E^+ - E^- = -(x_i^+ - x_i^-)\left(\sum_{j=1}^{n} \alpha_{ij} x_j - t_i \right).$$

Therefore, $E^+ - E^- \le 0$, and since there are only 2^n possible states, the system must end in a stable state. Two basic classes of computations have been associated to the trajectories of this system: a) error correction (EC) or content-addressable memory (CAM) operations, where the starting state is seen as a noisy or incomplete version of the final corresponding stable state; and b) optimization, in the sense that the network tends, at least locally, to minimize a certain quadratic function.

II. Generalizations

In this section we shall introduce neural networks with nonquadratic energy functions. However, we shall do so in a much more general setting, corresponding to a "topological" version of the Hopfield model. The reason is that with very little extra effort it is possible to achieve a broad and unifying view of several families of algorithms and automata, only a few of which have already been explored. It is also this setting that leads to a correct definition of dynamical information capacity.

The foregoing algorithm consists of a sequence of nonlinear operations describing the evolution of the neural network from the point of view of the "hardware," i.e., the neurons or the circuit which simulates them. Yet a different but totally equivalent description can be given using the trajectories in the space of states, the hypercube H^n of n-tuples of $(1, -1)$ coordinates.

Starting from one state X, with energy $E(X)$,

a) choose a new neighbor Y;
b) compute $E(Y)$ (or better, the difference $E(X) - E(Y)$);
c) if $E(Y) < E(X)$, move to Y; otherwise, go to step a).

It is essential to notice for our point that the discrete dynamical behavior does not depend at all on the actual value of the energies, but only on the partial ordering they induced on the vertices of the hypercube. This leads to the following construction. Let S be any (finite) set. A *neighborhood* on S is just a map N: $S \to 2^S$. Though this is not absolutely necessary, we shall add a condition of symmetry, namely, that if X belongs to $N(Y)$, then Y belongs to $N(X)$. As a consequence, we can always introduce a graph structure $G(S, N)$ in S by taking S as set of vertices and joining vertices which are neighbors for N. A partial ordering $\le_S$ on S is said to be *compatible* with N if and only if any two distinct neighbors X and Y in S are comparable (i.e., $X \le_S Y$ or $Y \le_S X$). In particular, any linear order on S is compatible with any N. Also if f is a function from S into a partially ordered set T, we can define a partial order induced by f on S by letting $X \le_S^f Y$ if and only if $f(X) \ne f(Y)$ and $f(X) \le_T f(Y)$. f is compatible if the induced order is compatible. For any partial ordering on S compatible with N we can define a *local improvement algorithm*.

Starting from a state X,

a) choose a new vertex Y such that $Y \in N(X)$;
b) If $Y \le_S X$, move to Y; otherwise, go to step a).

Again, if S is finite, this algorithm must halt. This definition is obviously incomplete since we have not specified the choice function for Y. This can be done in deterministic or probabilistic fashion. For most of our considerations here this choice function does not really matter. However, to fix his ideas the reader may assume a uniform probability distribution over the set of neighbors of the current point X, i.e., Y is chosen with probability equal to $|N(X)|^{-1}$.

If $\le_S$ is compatible with N, we can also orient any edge of $G(S, N)$ by $X \to Y$ if and only if $X \le_S Y$. It is easy to check that this yields an acyclic orientation (AO) of the graph G. In particular, any linear ordering of S leads to an AO of G, and, conversely, any AO of G can be obtained from a linear ordering on S or from an injective function f mapping S into a linear ordered set (see Fig. 1). It may be useful to think of the AO as a "landscape" on G and of the function f as an "energy" function. We can naturally define a *stable point* to be a vertex of G with outdegree 0. If X is stable, the *basin of attraction* of X is the set of all points Y such that all directed paths starting at Y end in X (i.e., the local algorithm started at Y always halts at X). Using the metric induced by the shortest path on G, there is a maximal sphere of radius $R(X)$ contained in the basin. Every stable point is, therefore, surrounded by three increasing sets: its sphere of attraction, its basin of attraction, and the set of points that are connected to X by at least one directed path (see Fig. 2). Occasionally, a basin of attraction may be spherical. However, it is essential to notice that two basins of attraction cannot be connected by a directed path.

Computationally, local improvement algorithms have been used to tackle discrete optimization problems. Though they are, of course, plagued by the problem of local minima, they are not necessarily worthless strategies as exemplified by the λ-opt algorithm for the traveling salesman problem [14]. In addition, in a context of CAM or EC, a very different point of view can be adopted in which local minima or stable points are seen as "codewords" and local improvement algorithms as "decoding." It is possible to generate in this fashion several new "codes," though it remains to be seen whether they could have practical applications. We shall just give an example. If F is a finite field of cardinal m and a is a generator of its multiplicative group, we can order F by $0 < 1 < a < \cdots < a^{m-2}$ and then consider polynomials f: $F^n \to F$. For instance, if $F = \{0, 1, a, a^2\}$, take $f(x, y) = ax + a^2 y$ with $N(x, y) = \{x\} \times F \cup F \times \{y\}$. It is easy to check that there are four

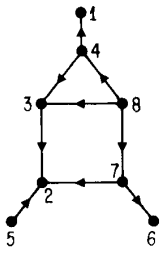

Fig. 1. Graph with ranking of vertices which induces an AO. Vertices ranked 1 and 2 are stable.

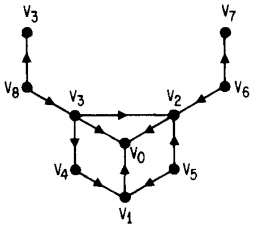

Fig. 2. Vertex V_0 is stable. Its basin of attraction is $\{v_0, v_1, v_2, v_3, v_4, v_5\}$ and its sphere of attraction $\{v_0, v_1, v_2, v_3\}$ has radius one.

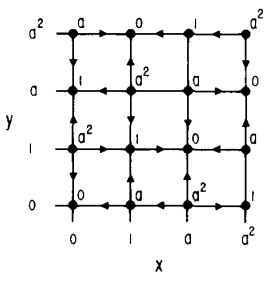

Fig. 3. $F = \{0, 1, a, a^2\}$, $0 < 1 < a < a^2$. AO of F^2 using linear form $f(x, y) = ax + a^2y$ (0,0), $(1, a^2)$, $(a, 1)$, and (a^2, a) are stable.

"codewords" corresponding to the kernel of the linear form f (see Fig. 3).

From a traditional standpoint these "codes" have two related problems: the possibly nondeterministic behavior on the boundaries of the basins of attraction and the fact that, in general, it is not possible to tightly pack spheres on G using AO. We shall examine these points in more detail for neural networks in Section V.

The elements of S can also be seen as representing the states of an automaton and the local improvement algorithm as describing its transitions in the course of computations. Two energy functions (or two automata) which induce the same AO on G cannot be distinguished from the statistics of their dynamical behavior. We can accordingly introduce an equivalence relation on N-compatible functions by $f \equiv g$ whenever f and g induce the same AO. If $A(G)$ is the set of all AO's on G, we can define the *dynamical capacity* $C(G)$ of G in bits to be the logarithm base 2 of $|A(G)|$. Similarly, if $\mathcal{F}$ is a class of compatible functions, the dynamical capacity $C^{\mathcal{F}}(G)$ of the class is the logarithm base 2 of the number of distinct AO induced by members of $\mathcal{F}$.

The Hopfield model is a special case of the previous analysis with $G = H^n$ and AO induced by quadratic real-valued energy functions. As we have just seen, it can be extended using other classes of compatible functions. The original model has an additional desirable feature: the local improvement algorithm is matched with a local updating rule at the "hardware" level. This remains true if we consider energy functions which are polynomial. There are many other reasons for considering such polynomials extensions: they can be implemented in hardware (see [21] for an optical one) and they appear very naturally in a context of optimization (see Appendix I for a family of examples). Finally, as we shall see in Section IV, quadratic forms are very limited in storage capacity, and therefore, it might be interesting and important to explore the different degrees of flexibility and programming capability realizable with polynomials of increasing degrees (see also [15]).

To fix the notation, if $X = (x_1, \cdots, x_n)(x_i = \pm 1)$, and if $\mathcal{I}$ is a family of subsets of $N = \{1, \cdots, n\}$, then an algebraic form in n variables based on $\mathcal{I}$ with coefficients in A is a polynomial expression of the type

$$P_n(X) = \sum_{I \in \mathcal{I}} \alpha_I X^I \qquad (\alpha_I \in A)$$

where $X^I = \prod_{i \in I} x_i$ and $x^\phi = 1$. Notice that on H^n only 0 and 1 powers of the variables need to be considered. The degree of the form is the cardinal of the largest I in $\mathcal{I}$, and the form is said to be homogeneous of degree d if $|I| = d$ for any I. In general, we shall use "bars" ($\bar{P}$) for the homogeneous case. Typically, a form of degree d depends on $r(n, d) = \sum_{k=0}^{d} \binom{n}{k}$ coefficients and a homogeneous form on $\binom{n}{d}$ coefficients. If P_n^d is viewed as the energy function of degree d of a generalized neural network, then we can write

$$P_n^d = x_i P_{n-1}^{d-1} + Q_{n-1}^d$$

where P_{n-1}^{d-1} is a form of degree $d-1$ in the variables $x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n$ and Q_{n-1}^d is a form of degree d in the same $n-1$ variables. The updating rule for unit i is: $x_i^+ = \operatorname{sgn}(-P_{n-1}^{d-1})$. As in the quadratic case, this forces the monotonicity of P_n^d. From a technical point of view a form could be (even if rarely) noncompatible on H^n for it could take the same value on two adjacent vertices. In addition, situations exist where one would like to restrict the coefficients to be rationals or integers. These two points are settled by the following Proposition.

Proposition 1: 1) For any form P_n^d with real coefficients, we can construct a form Q_n^d with real coefficients such that $Q_n^d(X) \neq Q_n^d(Y)$ for any two vertices X and Y. Also, whenever $P_n^d(X) < P_n^d(Y)$, then $Q_n^d(X) < Q_n^d(Y)$.

2) For any form Q_n^d with real coefficients and such that $Q_n^d(X) \neq Q_n^d(Y)$ for any two vertices X and Y, we can construct a form R_n^d with rational coefficients (and hence a form S_n^d with integer coefficients) such that $Q_n^d(X) < Q_n^d(Y)$ iff $R_n^d(X) < R_n^d(Y)$ (and iff $S_n^d(X) < S_n^d(Y)$).

Proof: See Appendix II.

From now on we shall only consider compatible forms and will assume, without loss of generality, that the coefficients are real numbers. In particular, there is no restriction on their size or precision. Notice, finally, that each unit needs only to locally compute the sign of an algebraic form of degree $d-1$ in $n-1$ variables. A *switching function* $f(x_1,\cdots,x_n)$ of n binary variables is a function f: $H^n \to \{0,1\}$. f is *separable* by a form of degree d if we can find a form P_n^d such that the algebraic surface $P_n^d = 0$ separates the "on" set $f^{-1}(1)$ from the "off" set $f^{-1}(0)$. A switching function which is separable by a form of degree d is called an *algebraic threshold function* of degree d. Therefore, every unit acts as an algebraic threshold function of degree $d-1$. $T_n^{\mathscr{F}}, T_n^d$, (resp. $\overline{T}_n^d$ for the homogeneous case) will represent the set of algebraic threshold functions with real coefficients based on $\mathscr{F}$ of degree d, and C_n^d (resp. $\overline{C}_n^d$) the dynamical capacity of neural networks with energy function of degree d. Notice that the definition of information capacity for the Hopfield model introduced in [1] is a special case of ours and corresponds to C_n^2.

III. Dynamical Capacities and the Number of Algebraic Threshold Functions

We shall first derive upper bounds and then lower bounds on the number of threshold functions and dynamical capacities. In a generalized neural network with energy function of degree d, we have seen that every unit acts as an algebraic threshold function of degree $d-1$ in $n-1$ variables. Therefore, the total number of such automata (or AO) is upper bounded by $|T_{n-1}^{d-1}|^n$. A well-known bound (see, for instance, [8]) exists on $|T_n^1|$:

$$|T_n^1| \le 2\sum_{i=0}^{n}\binom{2^n-1}{i} \le 2^{n^2}.$$

Combining these two facts, it is shown in [1] that $C_n^2 \le n(n-1)^2$.

We sketch the steps of a similar proof for the general case (details of the derivation can be found in [3]). First, it is easy to see that for every subset $\mathscr{F}$ of $\mathscr{P}(N)$ containing the empty set and for $1 \le d \le n$ we have $|T_n^{\mathscr{F}}| \le T_{|\mathscr{F}|-1}^1|$. As a consequence, $|T_n^{\mathscr{F}}| \le B_{|\mathscr{F}|}^{2^n} = 2\Sigma_0^{|\mathscr{F}|-1}\binom{2^n-1}{k}$. Using the simple bounds $|T_n^{\mathscr{F}}| < 2|\mathscr{F}|\binom{2^n-1}{|I|-1}$ for $|\mathscr{F}| \le (2^n+1)/2$ and $|T_n^{\mathscr{F}}| < (2^n-1)^{|\mathscr{F}|-1}$ if, in addition, $5 \le |\mathscr{F}|$, we can derive the following Theorem.

Theorem 1: We have

1) $|\overline{T}_n^d| < 2\dfrac{n^{d+1}}{d!}$ as soon as $\binom{n}{d} \ge 5$

2) $|T_n^d| < 2\dfrac{n^{d+1}}{(d-1)!}$ as soon as $r(n,d) \ge 5$ and $d \le \left\lfloor \dfrac{n}{2} \right\rfloor$.

A similar result can be found in [9]. Using the symmetry of homogeneous forms (i.e., 2^{n-1} hyperplanes instead of 2^n) does not improve the second bound. Notice also that, essentially, a one-to-one correspondence exists between forms of degree d and $n-d$. Namely, if $P_n^d = \Sigma_{|I|=d}\alpha_I x^I$, define Q_n^{n-d} by $Q_n^{n-d} = \Sigma_I \alpha_I x^{N-I}$. We can now prove an upper bound on the number of acyclic orientations or capacities corresponding to different classes of generalized networks.

Theorem 2: For $1 \le d \le n$:

1) $$C_n^1 = \overline{C}_n^1 = n;$$

2) $$\overline{C}_n^d < \frac{n^{d+1}}{(d-1)!}, \qquad \text{for } 5 \le \binom{n-1}{d-1};$$

3) $$C_n^d < \frac{n^{d+1}}{(d-2)!}, \qquad \text{for } 5 \le r(n-1,d-1) \text{ and } d \le \left\lfloor \frac{n+1}{2} \right\rfloor;$$

4) $$C_n^n < n2^{n-1}.$$

Proof: 1) This is clear since the orientation, by a form of degree one, of any edge parallel to the ith coordinate axis depends only on the sign of $\alpha_{\{i\}}$. Notice that this enumeration is for linear forms such that $\alpha_{\{i\}} \neq 0$ for any i. These are the only linear forms which define properly an AO of H^n. They have an additional property of a unique local-global minimum. (If one allows 0 coefficients in the form, then some ambiguities arise in the orientation of corresponding edges, and we then have $C_n^1 = n\log_2(3)$.)

The proof of 2) is similar to 3).

3) Given a neural network with energy P_n^d, recall that each unit simulates an algebraic threshold function in T_{n-1}^{d-1}. Therefore, $C_n^d \le n\log_2|T_{n-1}^{d-1}|$. We then apply Theorem 1.

4) The total number of orientations (including cyclic ones) is $2^{|E|} = 2^{n2^{n-1}}$. This bound can be microscopically improved by subtracting, for instance, the number of cyclic orientations where a given fixed face receives a cyclic orientation. Because of 4), the bound in 2), for instance, is interesting only if $n^{d+1}/(d-1)! \le n2^{n-1}$ or $n^d \le (d-1)!2^{n-1}$.

We turn now to lower bounds. The threshold functions corresponding to the different neurons are obviously dependent. Yet for $d \le \lfloor n/2 \rfloor$ we can study the collection of networks where the first $\lfloor n/2 \rfloor$ neurons simulate independent threshold functions of degree $d-1$ in the remaining $\lfloor n/2 \rfloor$ variables. Since here we are mainly interested in asymptotic values, we shall not distinguish the cases n even or odd because that leads only to trivial improvements. We know that in some sense $T_n^{\mathscr{F}} \subset T_{|\mathscr{F}|-1}^1$, and in [18] a construction shows that $|T_n^1| > 2(n(n-3)/2)+8$ for $9 \le n$. For the special importance case of $d=2$, this yields

$$\left(\frac{\left\lfloor \frac{n}{2} \right\rfloor\left(\left\lfloor \frac{n}{2} \right\rfloor - 3\right)}{2} + 8\right)\left\lfloor \frac{n}{2} \right\rfloor < C_n^2 < n(n-1)^2.$$

Hence the capacity C_n^2 is exactly of the order of n^3 bits, which is the result of [1]. However, for $2 < d$ we cannot

reason in the same fashion because the inclusion above is strict. Yet the following is true.

Theorem 3: We have

$$|T_n^{\mathcal{F}}| > 2^{|\mathcal{F}|}.$$

Proof: Consider the square matrix M with 2^n rows indexed by the vertices $X=(x_1,\cdots,x_n)$ of H^n, 2^n columns indexed by the subsets I of $N=\{1,2,\cdots,n\}$, and such that $M(X,I)=x^I=\prod_{i\in I}x_I(x^\Phi=1)$. It is easy to check that the scalar product of any two rows of M is zero. Therefore, M is in fact a Hadamard matrix and so is invertible. In particular, for any set $\mathcal{F}\subset\mathcal{P}(N)$ we can find $|\mathcal{F}|$ vectors on H^n such that the corresponding $|\mathcal{F}|\times|\mathcal{F}|$ submatrix has full rank. Let $M_{\mathcal{F}}$ be such a matrix, corresponding to vectors $Y_1,\cdots,Y_{|\mathcal{F}|}$ of H^n. If $\alpha=(\alpha_I)$ is the column vector representing the coefficient of a form based on $\mathcal{F}$, the system $M_I\alpha=\beta$ has a unique solution for any vector β. In particular, for any subset J of $\{1,2,\cdots,|\mathcal{F}|\}$ we can find an element T in $T_n^{\mathcal{F}}$ such that $T(Y_i)>0$ if $i\in J$ and $T(Y_i)<0$ if $i\in\{1,2,\cdots,|\mathcal{F}|\}-J$. Therefore, $2^{|\mathcal{F}|}<|T_n^{\mathcal{F}}|$.

As a straightforward application $|\bar{T}_n^d|>2\binom{n}{d}$ and $|T_n^d|>2^{r(n,d)}$. Because of $C_n^d>\lfloor n/2\rfloor\log_2|T_{\lfloor n/2\rfloor}^{d-1}|$ and of $\bar{C}_n^d>\lfloor n/2\rfloor\log_2|\bar{T}_{\lfloor n/2\rfloor}^{d-1}|$, we have the following.

Theorem 4: For $d\le\lfloor n/2\rfloor$

1) $$\bar{C}_n^d>\left\lfloor\frac{n}{2}\right\rfloor\binom{\lfloor\frac{n}{2}\rfloor}{d-1}$$

2) $$C_n^d>\left\lfloor\frac{n}{2}\right\rfloor r\left(\left\lfloor\frac{n}{2}\right\rfloor,d-1\right).$$

Using simple bounds in these exponents, we get

$$C_n^d>\frac{1}{(d-1)!}\left\lfloor\frac{n}{2}\right\rfloor\left(\left\lfloor\frac{n}{2}\right\rfloor-d+2\right)^{d-1}$$

and

$$C_n^n>C_n^{\lfloor n/2\rfloor}>\left\lfloor\frac{n}{2}\right\rfloor(2^{\lfloor n/2\rfloor}-1).$$

We can summarize part of our results with the following theorem.

Theorem 5: We have

1) $$\left\lfloor\frac{n}{2}\right\rfloor(2^{\lfloor n/2\rfloor}-1)<C_n^n(H^n)<n2^{n-1}.$$

2) Moreover, for $d\le\lfloor n/2\rfloor$,

$$\frac{1}{(d-1)!}\left\lfloor\frac{n}{2}\right\rfloor\left(\left\lfloor\frac{n}{2}\right\rfloor-d+2\right)^{d-1}<C_n^d<\frac{n^{d+1}}{(d-2)!}.$$

IV. Stable States

We have just analyzed a notion of capacity for generalized neural networks, exploiting their rich dynamics. However, other complementary definitions of capacity are possible. In the context of error correcting codes or of content-addressable memories, one might be intersted in the stable points of these systems and how to program them by selecting the appropriate energy function. We can, therefore, measure other capacities using the number and/or the structure of the programmable stable states (or of other types of transitions) and of their basins of attraction. We shall now partially examine generalized neural networks from this point of view. Given a generalized neural network defined by an energy function P_n^d, its stable states are the local minima of P_n^d on H^n, i.e., points X such that $P(X)\le P(Y)$ for any neighbor Y of X. This is also equivalent to the statement $x_i=\operatorname{sgn}(-P_{n-1}^{d-1})$ for $i=1,\cdots,n$ (notice that the form P_{n-1}^{d-1} depends on i). There are at least two distinct classes of orientations considered in the literature:

1) random classes of orientations,
2) specific classes of orientations;

and the results in general are exact or of an asymptotic nature ($n\to\infty$).

Random Classes of Orientations

Different notions of randomness can be introduced.

1) If we choose a random orientation for each edge with probability $p=0.5$, then in general we do not get an AO. However, the expected number of stable states is readily seen to be equal to 1 with Poisson asymptotic distribution.

2) Assume now we want to define a probability distribution on the set of AO's of the hypercube. One possibility is to put a uniform distribution on the set of linear orderings of the vertices of the hypercube.

Proposition 2: With a uniform distribution on the set of linear orderings of the vertices of H^n, the expected number of stable points is $2^n/n+1$.

Proof: The expected number of stable points is 2^n times the probability that a fixed vertex X is stable. X is stable if its "energy" is less than that of its n neighbors, an event of probability $1/(n+1)$. It is also possible to show that the variance is given by $2^{n-1}(n-1)/(n+1)^2$ and that a central limit theorem holds for the distribution of the stable states (see [6]).

A third possibility is to consider generalized neural networks where the coefficients of the energy function are chosen according to some probability distribution. For forms $\bar{P}_n^d$ with coefficients selected using $\binom{n}{d}$ independent zero-mean Gaussian random variables with same variance, the expected number of stable points is asymptotically equal to $k_d 2^{c_d n}$, where $2\le d$ and c_d, k_d are computable parameters which depend only on d. This is essentially a generalization of the classical spin glass case of statistical mechanics (for more details see [5]).

Specific Classes of Orientations

We consider now generalized networks where the coefficients of the energy function are carefully constructed to force a precise set of transitions. The most simple case is

when we fix in advance a set of vertices of H^n and look for a network that includes these points among its stable states. A review of possible storage rules, some of their basic properties, together with exact results on the structure of their stable states can be found in [4]. Asymptotic results are also derived in [2], [5], and [16]. We now derive exact bounds on the number of programmable stable states regardless of the particular storage scheme adopted. In [1] a theorem is presented showing that if we want to be able to store *any* set of k vectors using a quadratic energy function, then $k \leq n$. As pointed out by several authors, the result as stated is problematic for it is easy to find small configurations of vectors (in fact, with $k=2$) that cannot be stored. However, these counterexamples are artificial and often use vectors which are at Hamming distance 1. Because of the very nature of acyclic orientations, stable points must be by definition at distance (for the metric induced by the shortest path on the corresponding graph) at least 2 from each other and at least 3 if we require that each stable point be surrounded by a sphere of attraction of radius at least 1. A lower bound of at least 2 must therefore be imposed on the minimal distance occurring between the vertices that one would like to store. This is still not sufficient to remove all pathological cases and leads to the following technical definition. For a fixed set of vectors $M^1, \cdots, M^k$ and degree d, we consider the submatrix $S^d(M)$ of the matrix M appearing in the proof of Theorem 3, corresponding to the columns indexed by subsets I with $|I| \leq d$ and rows indexed by all the vectors N such that $d(N, M^i) \leq 1$ for some i. (If the minimal distance between the vectors $M^1, \cdots, M^k$ is at least 3, then $S^d(M)$ is a $k(n+1) \times r(n,d)$ matrix). The vectors $M^1, \cdots, M^k$ are said to be *well conditioned* iff $S^d(M)$ has full rank. We then have the following theorem.

Theorem 6: Suppose that for any k vectors $M^1, \cdots, M^k$ of H^n with minimal Hamming distance of at least 2 we can find a form P_n^d such that $M^1, \cdots, M^k$ are local minima of P_n^d on H^n. Then

$$k \leq r(n-1, d-1) = \sum_{i=0}^{d-1} \binom{n-1}{i}.$$

Moreover, if they are well conditioned, then

$$\frac{r(n,d)-1}{n+1} \leq k.$$

In particular, if $d=2$,

$$\frac{n}{2} \leq k \leq n.$$

Proof: Consider k vectors $M^1, \cdots, M^k$. Fix their $n-1$ last coordinates such that they are all different ($k < 2^{n-1}$). Consider the threshold function corresponding to the first coordinate. For any of the 2^k possible choices for the components $M_1^1, \cdots, M_1^k$ we must find an algebraic form P_{n-1}^{d-1} satisfying $\operatorname{sgn}(P_{n-1}^{d-1}(M_2^i, \cdots, M_n^i)) = M_1^i$ for $i = 1, \cdots, k$. For any i the coefficients $M_2^i, \cdots, M_n^i$ define a hyperplane in the space F_{n-1}^{d-1}. There are k such hyperplanes, and we have seen in Section III that they determine at most $B_{r(n-1,d-1)}^k$ regions. Therefore, $2^k \leq 2\sum_{i=0}^{r(n-1,d-1)-1} \binom{k-1}{i}$. If $r(n-1,d-1) < k$, then we would have $2^k < 2\sum_{i=0}^{k-1}\binom{k-1}{i} = 2^k$, a contradiction. Therefore, $k \leq r(n-1, d-1)$.

To get the lower bound, notice that we need only to find a polynomial of degree d taking a value a on the vectors to be stored, and a value $b > a$ on the at most nk points surrounding them. This yields $(n+1)k$ linear equations in the coefficients of the polynomial to be found. Notice that the matrix of this linear system is exactly $S^d(M)$. It has $r(n,d)$ columns and at most $(n+1)k$ rows. Therefore, if $(n+1)k < r(n,d)$, the linear system is always solvable and its resolution yields the coefficients of a polynomial that has $M^1, \cdots, M^k$ among its local minima. It should be noticed that the condition of Theorem 6 is a strong one and more precise bounds can be derived for specific storage formulas. However, all the additional known results confirm that under a variety of different assumptions, the storage capacity of neural networks with energy function of degree d is at most of the order of $O(n^{d-1})$. Therefore, for large n the number of bits that can be stored per connection weight is essentially a constant (which may depend on the storage rule adopted, the degree $d, \cdots$).

V. Applications: Codes and Cycles

A natural question to raise (and partially examined in [7]) is whether traditional codes could be implemented using generalized neural networks. Since any two basins of attraction cannot be connected by a directed path, it is obvious that no disjoint packing of the hypercube into spheres can be realized using an AO. Therefore there is, for example, no AO on H^7 corresponding to the (7,4) Hamming code. However, if we view this code as essentially made of 16 stable points, each with a radius of attraction of at least 1, then it can be implemented with an AO on a higher dimensional cube satisfying $16(n+1) < r(n,d)$. More generally, the arguments in the proof of the lower bound in the previous theorem can be used to find conditions on n and d to construct an AO with given properties. For instance, if we want to have four stable points with radius at least 1 and four stable points with radius at least 2, in general, it will be sufficient to have $4(2+2n+\binom{n}{2}) < r(n,d)$, which is possible with a form of degree 3 as soon as n is large enough.

Simple examples of this type show that because of the very nature of AO's implementation of traditional codes via neural networks is rather expensive. Typically, for any codeword with a ball of attraction of radius R, the points immediately surrounding this ball cannot belong to the sphere of attraction of other codewords and therefore are "wasted." On H^n most of the points of a sphere are located on its surface, and this leads to important losses. For coding applications it may therefore be of interest to look at AO's on graphs with lower connectivity than the hypercube, for instance, the usual lattices where the ratio

of the surface of a sphere to its volume behaves typically like cst/R. To generate an AO on the square lattice, for instance, one can, of course, use functions $f\colon \mathbb{Z}^2 \to \mathbb{R}$. Another more interesting way is to use a Gray code along each coordinate axis which yields a locally isometric embedding of H^{a+b} in a $2^a \times 2^b$ piece of lattice. Finally, applications of neural network ideas to analog decoding can be found in [20].

So far we have concentrated on orientations which had the property of being acyclic, forcing the local algorithm to converge. However, it is not unreasonable to conjecture, for instance, that the command organ of a complex system (living organism, robot, etc.) might cycle through a sequence of states during certain periodic activities of the system (gait, etc.). Consequently, it is of interest to explore classes of orientations which are not even acyclic. It is still possible to orient the hypercube in a nonacyclic fashion using a neural type of network. To create such a network of "degree d," it suffices to assign to each device or neuron an algebraic threshold function in T_{n-1}^{d-1}. Yet this time the algebraic functions can be uncorrelated and the interactions between d (or fewer) units are not symmetric, as in the original Hopfield model, but may depend on their ordering. If O_n^d represents the family of orientations we can define in this manner, a glance at the proofs of Section III will convince the reader that exactly the same bounds (as in Theorem 5, for instance) hold. Therefore,

$$\frac{1}{(d-1)!}\left\lfloor \frac{n}{2} \right\rfloor \left(\left\lfloor \frac{n}{2} \right\rfloor - d + 2 \right)^{d-1} < \log_2 |O_n^d| < \frac{n^{d+1}}{(d-2)!}.$$

(Obviously, $|O_n^n| = 2^{n2^{n-1}}$.) Sharper bounds are required to separate C_n^d from $\log_2 |O_n^d|$. It is known [22] that the number of acyclic orientations of a graph is equal to the value of its chromatic polynomial evaluated at -1. It remains to be investigated whether combinatorial techniques of this type may be of help in these matters.

To evaluate the number of cycles that can be stored, similar arguments to those used for the lower bound of Theorem 6 can be applied again. For each unit i in T_{n-1}^{d-1} we have $r(n-1, d-1)$ degrees of freedom. Any time we fix the orientation of an edge of H^n parallel to the ith coordinate, we get a linear inequality (which can be arbitrarily turned into an equality) in the $r(n-1, d-1)$ coefficients. Therefore, in general we can fix the orientation of at least $r(n-1, d-1)-1$ edges parallel to the ith coordinate, and this can be repeated independently for each unit or direction. In this fashion we can implement cycles (or more generally sets of transitions) even with the requirement that they be edge disjoint, and find lower bounds. We shall give two examples.

Example 1: Consider the task of storing a cycle of length $2m$ as an "attractor" (i.e., all edges adjacent to the cycle must lead to it). We need to fix the orientation of $2m$ edges in the cycle and of $2m(n-1)$ edges adjacent to the cycle. In any given direction i we have at most $2m$ equations. Therefore, the problem can be solved by constructing the cycle in an H^m subcube provided we have $2m < r(n-1, d-1)$.

Example 2: Assume now we want to store k cycles of length $2m$ (not necessarily as attractors). We can consider $\lfloor n/m \rfloor$ independent m-dimensional subcubes of H^n and store edge-disjoint cycles in each one of them. Each cycle of length $2m$ yields two equations for each unit. It should be possible to implement in this manner at least $1/2\lfloor n/m \rfloor (r(n-1, d-1)-1)$ such cycles.

A final remark is that the issue of timing, i.e., how the network is updated, is more crucial in the case of cycles than in the case of an AO. With the appropriate choice of sequence of updates and by combining cycles belonging to different subcubes one can obtain complicated long cycle structures. On the other hand, if the cycles are not attractors and if the updating steps are completely stochastic, it is hard to imagine how these cycles could be used to process information. One solution is to have a deterministic mode of operation, sequential or synchronous (see, for instance, [19]). It is also important to provide mechanisms that allow one to exit from the cycles. One possible way is precisely by changing the mode of updates, shifting, for instance, from simultaneous to asynchronous.

Acknowledgment

I would like to thank John Hopfield, Carver Mead, and Yaser Abu Mostafa for useful insights, one of the referees for pointing out Cover's work, and Mario Blaum, Jorge Sanz and Jehoshua Bruck for discussions related to coding theory.

Appendix I

We give a simple example of a family of problems which naturally leads to the optimization of forms P_n^4 over H^n. Fundamental questions in combinatorial designs (such as the existence of a projective plane of order 10) and coding theory amount to the construction of an $m \times n$ matrix $A = (a_{ij})$ with $a_{ij} = \pm 1$ (or, equivalently, 0-1) with prescribed Hamming distances d_{kl} between row k and row l. This condition can be rewritten as

$$\sum_{j=1}^{n} a_{kj} a_{lj} = n - 2d_{kl}.$$

Therefore, A can be seen as the minimization over the hypercube of the quartic form

$$\sum_{k,l} \left(\sum_{j=1}^{n} a_{kj} a_{lj} - n + 2d_{kl} \right)^2.$$

Appendix II
Proof of Proposition 1

1) Let us assume that $P_n^d(X) = P_n^d(Y)$. Let $c = \min |P_n^d(Z) - P_n^d(T)|$, the minimum being taken over all vertices Z, T such that $P_n^d(Z) \neq P_n^d(T)$. Then if for any U $P_n^d(U) = \sum_{|I| \leq d} \alpha_I U^I$, we can construct a new form $\tilde{Q}_n^d(U) = \sum_I (\alpha_I + h_I) U^I$ with $|h_I| < c/2r(n, d)$ and $\sum_I h_I X^I \neq \sum_I h_I Y^I$ (for instance, if X and Y differ in position i, take $h_{\{i\}} = c/3r(n, d)$ and $h_I = 0$ for any $I \neq \{i\}$). Then, obviously, $\tilde{Q}_n^d(X) - \tilde{Q}_n^d(Y) \neq 0$. In addition, if Z and T

are two vertices such that $P_n^d(T) < P_n^d(Z)$, then $\tilde{Q}_n^d(Z) - \tilde{Q}_n^d(T) = P_n^d(Z) - P_n^d(T) + \Sigma_I h_I Z^I - \Sigma_I h_I T^I$. Now $P_n^d(Z) - P_n^d(T) \geq c$ by assumption, and, $|\Sigma_I h_I Z^I - \Sigma_I h_I T^I| \leq 2\Sigma_I |h_I| < c$. Therefore, $\tilde{Q}_n^d(Z) - \tilde{Q}_n^d(T) > 0$, and $\tilde{Q}_n^d$ preserves the partial ordering induced by P_n^d. Iterating this procedure a finite number of times yields the required form Q_n^d.

2) We use essentially the same procedure as in 1). If $Q_n^d(X) = \Sigma_I \alpha_I X^I$, let h_I be such that $\alpha_I + h_I$ is rational and $|h_I| < c/2r(n,d)$. Then $R_n^d(X) = \Sigma_I(\alpha_I + h_I)X^I$ has rational coefficients and preserves the ordering induced by Q_n^d. Finally, for any form P_n^d, the form $\lambda P_n^d + \mu$ ($\mu > 0$) induces exactly the same AO as P_n^d. Therefore, one needs only to multiply the rational form by the absolute value of the common denominator of all the coefficients to get an equivalent form with only integer coefficients.

REFERENCES

[1] Y. S. Abu Mostafa and J. M. St. Jaques, "Information capacity of the Hopfield model," *IEEE Trans. Inform. Theory*, vol. IT-31, no. 4, pp. 461–464, 1985.

[2] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Spin-glass models of neural networks," *Phys. Rev. A*, vol. 32, pp. 1107–1118, 1985.

[3] P. Baldi, "1) On a generalized family of colorings; 2) Some contributions to the theory of neural networks; 3) Embeddings of ultrametric spaces in finite dimensional structures," Ph.D. dissertation, California Institute of Technology, Pasadena, 1986.

[4] ____, "Group actions and learning for a family of automata," *J. Comput. Syst. Sci.*, vol. 36, no. 1, pp. 1–15, 1988.

[5] P. Baldi and S. S. Venkatesh, "Number of stable points for spin-glasses and neural networks," *Phys. Rev. Lett.*, vol. 58, no. 9, pp. 913–917, Mar. 1987.

[6] P. Baldi and Y. Rinott, "Asymptotic normality of some graph related statistics," *J. Appl. Prob.*, to be published.

[7] J. Bruck and J. Sanz "A study of neural network," IBM Almaden Research Center Internal Report, 1986.

[8] S. H. Cameron, "An estimate of the complexity requisite in a universal decision network," in *Proc. Bionics Symp.*, Wright Airforce Dev. Div. (WADD) Rep. 60-600, 1960, pp. 197–212.

[9] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IRE Trans. Electron. Comput.*, vol. EL-14, pp. 326–334, 1965.

[10] E. Fredkin and T. Toffoli, "Conservative logic," *Int. J. Theoretical Phys.*, vol. 21, nos. 3–4, pp. 219–225, 1982.

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

[12] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141–152, 1985.

[13] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[14] S. Lin, "Computer solutions to the traveling salesman problem," *Bell Syst. Tech. J.*, vol. 44, no. 10, pp. 2245–2269, 1965.

[15] T. Maxwell, C. L. Giles, Y. C. Lee, and H. H. Chen, "Nonlinear dynamics of artificial neural systems," in *Proc. Conf. Neural Networks for Computing*, Snowbird UT, Amer. Inst. Physics, 151, J. S. Denker, Ed., 1986, pp. 299–304.

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

[17] S. Muroga, "Lower bound on the number of threshold functions and a maximum weight," *IRE Trans. Electron. Comput.*, vol. EC-14, pp. 136–148, 1965.

[18] S. Muroga and I. Toda, "Lower bound on the number of threshold functions," *IRE Trans. Electron. Comput.*, vol. EC-15, pp. 805–806, 1966.

[19] L. Personnaz, I. Guyon, and G. Dreyfus, "Designing neural networks satisfying a given set of constraints," in *Proc. Conf. Neural Networks for Computing*, Snowbird, UT, Amer. Inst. Physics, 151, J. S. Denker, Ed., 1986, pp. 356–359.

[20] J. C. Platt and J. J. Hopfield, "Analog decoding using neural networks," in *Proc. Conf. Neural Networks for Computing*, Snowbird, UT, Amer. Inst. Physics, 151, J. S. Denker, Ed., 1986, pp. 364–369.

[21] D. Psaltis, and C. H. Park, "Nonlinear discriminant functions and associative memories," in *Proc. Conf. Neural Networks for Computing*, Snowbird, UT, Amer. Inst. Physics, 151, J. S. Denker, Ed., 1986, pp. 370–375.

[22] R. P. Stanley, "Acyclic orientations of graphs," *Discrete Math.*, vol. 5, pp. 171–178, 1973.

[23] R. O. Winder, "Bounds on threshold gate realizability," *IRE Trans. Electron. Comput.*, vol. EC-12, pp. 561–564, 1963.