Linear and Logarithmic Capacities in Associative Neural Networks

SANTOSH S. VENKATESH, MEMBER, IEEE, AND DEMETRI PSALTIS, MEMBER, IEEE

Abstract —**A model of associative memory incorporating global linearity and pointwise nonlinearities in a state space of n-dimensional binary vectors is considered. Attention is focused on the ability to store a prescribed set of state vectors as attractors within the model. Within the framework of such associative nets, a specific strategy for information storage that utilizes the spectrum of a linear operator is considered in some detail. Comparisons are made between this spectral strategy and a prior proposed scheme which utilizes the sum of Kronecker outer products of the prescribed set of state vectors which are to function nominally as memories. The storage capacity of the spectral strategy is linear in n, the dimension of the state space under consideration, while an asymptotic result of $n/4\log n$ holds for the storage capacity of the outer product scheme. Computer-simulated results are quoted in support of the analysis to show that the spectral strategy stores information more efficiently than the outer product scheme. Estimates of the preprocessing costs incurred in the two algorithms are provided, and recursive strategies are developed for their computation.**

I. Introduction

A. Neural Networks

We WILL CONSIDER two models of associative memory based upon ideas from neural nets and characterize their performance. A neural network consists of a highly interconnected agglomerate of cells called neurons. The neurons generate action trains dependent upon the strengths of the *synaptic* interconnections. The instantaneous state of the system is described by the collective states of each of the individual neurons (firing or not firing). Models of learning (the Hebbian hypothesis [1]), and associative recall based on such networks (cf. [2] for instance), have been developed and illustrate how distributed computational properties become evident as a collective consequence of the interaction of a large number of simple elements (the neurons). The success of these biological models for memory has sparked considerable interest in developing powerful distributed processing systems utilizing the neural network concept. Central features of such systems include a high degree of parallelism, distributed storage of information, robustness, and very simple basic elements performing tasks of low computational complexity.

Our focus in this paper is on the distributed computational aspects evidenced in such networks. In particular, we consider the capacity of two specific neural networks for storage of randomly specified data and their capability for error correction (or, equivalently, nearest neighbor search).

We will be concerned with a specific neural network structure. We assume a densely interconnected network with neurons communicating with each other through linear synaptic connections. We will consider two state transition mechanisms for the system. In the synchronous mode each neuron updates its state simultaneously (at clocked intervals, for instance). In the asynchronous mode, each neuron updates its state at random times independent of the update times of the other neurons; in essence, with high probability, at most one neuron updates its state at any given instant. Mathematically speaking, in either mode the state vector is operated upon by a global linear operation followed by a pointwise nonlinear operation. We will consider two specific constructive schemes for generation of the matrix of synaptic weights corresponding to the global linear transformation, and utilize a thresholding decision rule.

B. Organization

In the next section, we will describe the neural network structure from a system-theoretic point of view and define the parameters that are important for the system to function as an efficient associative memory. In Section III we briefly review a scheme for generation of the matrix of synaptic weights using outer products of the prescribed datum vectors. This scheme has been well outlined in the literature [3]–[6]. Our focus in this section is on the statement of a new result concerning the storage capacity of the outer product scheme.

In Section IV we outline a technique for generating the weight matrix by tailoring the linear transformation to obtain a desired spectrum. We analyze the performance of this scheme as an associative memory in some detail, and obtain estimates of its performance from purely theoretical considerations. We will use $\boldsymbol{W}$ to represent the linear operation for both schemes; in the event that it is impor-

Manuscript received March 20, 1985; revised January 20, 1988. This work was supported in part by the National Science Foundation under Grant EET-8709198. The material in this paper was partially presented at the Workshop on Neural Networks for Computing, Santa Barbara, CA, April 1985.

S. S. Venkatesh was with the California Institute of Technology, Pasadena, CA. He is now with the Moore School of Electrical Engineering, University of Pennsylvania, 200 South 33rd Street, Philadelphia, PA 19104.

D. Psaltis is with the Department of Electrical Engineering, California Institute of Technology, Mail Stop 116-81, 391 South Holliston Street, Pasadena, CA 91125.

IEEE Log Number 8928183.

0018-9448/89/0500-0558$01.00 ©1989 IEEE

tant to discriminate between the linear operators of the two schemes, we use W^{op} for the outer product scheme, and W^{s} for the spectral scheme. Sections V and VI are devoted to computer simulations of the two techniques, *ad hoc* modifications, and discussions.

Formal proofs of quoted results can be found in the Appendices. Propositions 1 and 2 are proved in Appendix I. Theorems 4 and 5 and Corollary 3 are proved in Appendix II.

II. Linear Nets with Pointwise Decision Rules

A. Description of the Model

We consider a system of n neurons, each capable of assuming two values: -1 (off) and 1 (on). The instantaneous state of the system is denoted by a binary n-tuple $u \in \mathbb{B}^n$, where $\mathbb{B} = \{-1,1\}$, and the components u_i, $i = 1,\cdots,n$, of u denote the state of the ith neuron. The adaptation of the system with time, or the flow in state space, is governed by two mathematical operations: 1) a globally acting linear transformation W: $\mathbb{R}^n \to \mathbb{R}^n$ (corresponding to *fixed* synaptic strengths w_{ij}), and 2) a pointwise thresholding operation Δ: $\mathbb{R}^n \to \mathbb{B}^n$.

Two distinct modes of operation are possible: *synchronous*, where the entire state vector updates itself, and *asynchronous*, where state changes are synonymous with bit changes and only a single randomly chosen neuron updates its state per adaptation. For the sake of notational simplicity, we use Δ to denote both these modes of operation, and it will be clear from the context which mode we are actually referring to at any given time.

We fix the thresholding level at zero, so that the thresholding operation can be described mathematically as follows: for each $v \in \mathbb{R}^n$,

$$\Delta(v) = u \in \mathbb{B}^n$$

where

$$u_i = \operatorname{sgn}(v_i) = \begin{cases} 1, & \text{if } v_i \geq 0 \\ -1, & \text{if } v_i < 0. \end{cases}$$

For the synchronous algorithm, the thresholding is done for each neuron, $i = 1,\cdots,n$, whereas for the asynchronous algorithm, only the ith neuron (corresponding to some randomly chosen $i \in \{1,2,\cdots,n\}$) is updated per adaptation, with the others held fixed.

We can clearly consider just the restriction of W to $\mathbb{B}^n$, as wandering in the state space is confined to binary n-tuples by the nature of the algorithm. The total adaptation algorithm can hence be described as a cascade of operations $\Delta \circ W$: $\mathbb{B}^n \to \mathbb{B}^n$.

Other processing modes are feasible in such neural networks. Models based on linear mappings, for instance, have been examined by Longuet-Higgins [7] and Gabor [8], while Poggio [9] has considered certain polynomial mappings.

B. Storage Capacity and Attraction

Our concern in this paper is with the information storage capability of the specified structure. For it to function as an associative memory, we require that almost all prescribed m-sets of state vectors $\{u^{(1)}, u^{(2)}, \cdots, u^{(m)}\} \in \mathbb{B}^n$ are storable in the network and that these state vectors are invokable by any input that is sufficiently close to any of the stored vectors in some sense, i.e., these states function as *attractors*. We shall soon make these intuitive notions more precise.

We henceforth refer to the prescribed set of state vectors as *datums* to distinguish them from all other states of the system. Now, it is a desideratum that the prescribed datums $u^{(1)}, u^{(2)}, \cdots, u^{(m)} \in \mathbb{B}^n$ are self-perpetuating, i.e., are stable. We shall say that a datum $u^{(r)} \in \mathbb{B}^n$ is strictly stable (or simply stable) if $(\Delta \circ W)u^{(r)} = u^{(r)}$. Thus a state is strictly stable if it is a fixed point of the neural network. Clearly, for this definition of stability, it is immaterial whether the system is synchronous or asynchronous.

We define capacity to be a rate of growth rather than an exact number as in traditional channel capacity in information theory. Specifically, consider an algorithm X for storing prescribed datums in a neural network of the type we consider. We will assume that the components of the datums are chosen from a sequence of symmetric Bernoulli trials.

Definition: A sequence of integers $\{C(n)\}_{n=1}^{\infty}$ is a sequence of *capacities* for algorithm X if and only if, for every $\lambda \in (0,1)$, as $n \to \infty$ the probability that each of the datums is strictly stable approaches one whenever $m \leq (1-\lambda)C(n)$, and zero whenever $m \geq (1+\lambda)C(n)$.

The above definition of capacity was formally arrived at by a consideration of lower and upper limits for capacity and has been found to be particularly well-suited in the present context [10]. Fig. 1 schematically illustrates the 0–1 behavior required by the definition of capacity. A consequence of the above definition is that if a sequence of capacities does exist, then it is not unique; however, any two sequences of capacity are asymptotically equivalent.

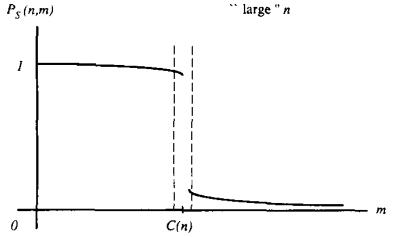

Fig. 1. 0–1 behavior of capacity for large n: schematic plot of probability that all datums are stored as fixed points by algorithm X (indicated by $P_S(n,m)$ in figure) is shown versus number of datums m.

In the structure of association, we would like the stored datums to have a region of influence around themselves, so that if an input vector is "sufficiently close" to a datum (in the Hamming distance sense), the adaptive algorithm will cause the neural network to settle into a stable state centered at that datum. The determination of attraction behavior in general depends on the specific structure of the linear transformation $\boldsymbol{W}$. For the specific case where $\boldsymbol{W}$ is symmetric, however, we can demonstrate Lyapunov functions for the system, and this suffices as an indicator of attraction behavior.

Let E: $\mathbb{B}^n \to \mathbb{R}$ be the quadratic form

$$E(\boldsymbol{u}) = -\frac{1}{2}\langle \boldsymbol{u}, \boldsymbol{W}\boldsymbol{u}\rangle = -\frac{1}{2}\sum_{i,j=1}^{n} w_{ij}u_i u_j. \quad (2.1)$$

By analogy with physical systems we refer to E as the energy of the current system state. As in physical systems, the system dynamics proceeds in the direction of decreasing energy [6], [12] (also cf. Section V-D). However, the result does not hold for arbitrary symmetric weight matrices in a synchronous mode of operation. For such cases the functional F: $\mathbb{B}^n \to \mathbb{R}$ defined by

$$F(\boldsymbol{u}) = -\sum_{i=1}^{n}\left|\sum_{j=1}^{n} w_{ij}u_j\right| \quad (2.2)$$

can be shown to be a Lyapunov function for the system [11].

In either case, fixed points of the system reside as minima of either E or F for systems with symmetric weight matrices. If the datums are programmed to be fixed-points by suitable choice of symmetric $\boldsymbol{W}$, then under the above conditions, trajectories in state space in the vicinity of the datums will tend to settle into strictly stable states at the datums thus establishing basins of attraction around the datums. (Limit cycles are possible with either E or F identically zero for those states. This, however, has small probability in most cases.)

C. Algorithm Complexity and Recursive Constructions

An algorithm for storing datums in a neural network is hence simply a prescription for determining the weights w_{ij}. We will characterize the algorithm preprocessing cost by the number of elementary operations required to compute the matrix of weights $[w_{ij}]$. For our purposes, we define an elementary operation to be the multiplication (or the addition) of two real quantities.

Another facet of computational importance is whether a recursion can be set up for the algorithm whereby weight matrices could be simply updated whenever a new datum is to be added to the existing set of datums. Let $\mathbb{M}^{(n)}$ denote the family of $n \times n$ matrices with real components. Consider an algorithm X for generating interconnection weight matrices. Let $\{\boldsymbol{u}^{(j)}\}$ be a sequence of datums, and let $\{\boldsymbol{W}^X[j]\}$ be a sequence of weight matrices in $\mathbb{M}^{(n)}$, where $\boldsymbol{W}^X[j]$ denotes the weight matrix generated by X for storing the first j datums $\boldsymbol{u}^{(1)},\cdots,\boldsymbol{u}^{(j)}$. For convenience, we set $\boldsymbol{W}^X[0]=\boldsymbol{0}$.

Definition: An algorithm X for generating weight matrices is memoryless if and only if there is a function f^X: $\mathbb{M}^{(n)} \times \mathbb{B}^n \to \mathbb{M}^{(n)}$ such that for any choice of datums $\{\boldsymbol{u}^{(j)}\}$, the sequence of weight matrices $\{W^X[j]\}$ generated by X satisfies

$$\boldsymbol{W}^X[k] = f^X(\boldsymbol{W}^X[k-1], \boldsymbol{u}^{(k)}), \qquad k \geq 1. \quad (2.3)$$

Thus, for memoryless algorithms, a new weight matrix can be generated given only knowledge of the previous weight matrix and the new datum. The terminology "memoryless" refers to the fact that, for storage algorithms satisfying this property, a set of datums can be essentially "forgotten" once a matrix of weights has been generated for their storage; updates of the weight matrix for the storage of new datums access information about the previous datums only through the generated matrix of weights. This is a computationally useful feature as the necessity of keeping track of the stored datums in some external storage medium is obviated.

Definition: A storage algorithm X is (additively) local if and only if there are functions g_{ij}^X: $\mathbb{R} \to \mathbb{R}$ and h_{ij}^X: $\mathbb{B}^n \to \mathbb{R}$, $i, j = 1,\cdots,n$, such that, for any choice of datums $\{\boldsymbol{u}^{(k)}\}$, the components $w_{ij}^X[k]$ of the generated sequence of weight matrices satisfy

$$w_{ij}^X[k] = g_{ij}^X\left(w_{ij}^X[k-1]\right) + h_{ij}^X(\boldsymbol{u}^{(k)}), \qquad i,j=1,\cdots,n. \quad (2.4)$$

Algorithm locality is a particularly nice feature to have as it almost invariably implies low computational requirements for matrix updates. Another appealing feature of local algorithms is that component updates can be done "in place." Particular simplicity results if the functions $g_{ij}^X = Id$. In this case,

$$\boldsymbol{W}^X[k] = \boldsymbol{W}^X[k-1] + g^X(\boldsymbol{u}^{(k)})$$

where g^X: $\mathbb{B}^n \to \mathbb{M}^{(n)}$.

More general definitions of locality are, of course, possible, but additive locality suffices for our purposes. Clearly, every local algorithm is memoryless.

III. The Outer-Product Algorithm

A. The Model

We review here a correlation-based scheme for generating the linear transformation $\boldsymbol{W}$. The scheme is based upon the sum of the outer products of the datum vectors and has been well-documented in the literature. Nakano [3] coined the term "associatron" for the technique and demonstrated how a linear net constructed using outer products of prescribed state vectors could be combined with a pointwise thresholding rule to obtain a time sequence of associations, with some ability for recall and error correction. More recent papers emphasize the role of the nonlinearity in the scheme and include both synchronous and asynchronous approaches. The conditions under which long-term correlations can exist in memory have been investigated by Little [4] and Little and Shaw [5] utilizing a synchronous model. Using an asynchronous model, Hopfield [6] demonstrated that the flow in state space was such that it minimized a bounded "energy" functional and that associative recall of chosen datums was hence feasible with a measure of error correction.

We now describe the model. We assume that m datums $\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \cdots, \boldsymbol{u}^{(m)} \in \mathbb{B}^n$ have been chosen randomly. The matrix of weights is constructed according to the following prescription: for $i, j = 1, \cdots, n$, set

$$w_{ij}^{\mathrm{op}} = \begin{cases} \sum_{r=1}^{m} u_i^{(r)} u_j^{(r)}, & \text{if } i \neq j \\ 0, & \text{if } i = j. \end{cases} \tag{3.1}$$

Thus $[w_{ij}^{\mathrm{op}}]$ is a symmetric zero-diagonal matrix of weights. We shall briefly review the question of stability and attractors in the model before quoting the result on the storage capacity of the network with this particular choice of linear transformation.

We first demonstrate that the datums are stable (at least in a probabilistic sense). Assume that one of the datums $\boldsymbol{u}^{(r)}$ is the initial state of the neural network. For each $i = 1, \cdots, n$, we have

$$\begin{aligned}(\boldsymbol{W}^{\mathrm{op}}\boldsymbol{u}^{(r)})_i &= \sum_{j=1}^{n} w_{ij}^{\mathrm{op}} u_j^{(r)} = \sum_{\substack{j=1 \\ j \neq i}}^{n} \sum_{s=1}^{m} u_i^{(s)} u_j^{(s)} u_j^{(r)} \\ &= (n-1) u_i^{(r)} + \sum_{s \neq r} \sum_{j \neq i} u_i^{(s)} u_j^{(s)} u_j^{(r)}. \end{aligned} \tag{3.2}$$

We assume that the datum components, $u_i^{(r)}$, $i = 1, \cdots, n$, $r = 1, \cdots, m$, are generated from a sequence of symmetric Bernoulli trials; specifically, $\boldsymbol{P}\{u_i^{(r)} = -1\} = \boldsymbol{P}\{u_i^{(r)} = 1\} = 1/2$. It then follows that the second term of (3.2) has zero mean and variance equal to $(n-1)(m-1)$, while the first term is simply $(n-1)$ times the sign of $u_i^{(r)}$. The terms w_{ij} from (3.1) are the sum of m independent random variables and are hence asymptotically normal vide the central limit theorem. As the second term in (3.2) is asymptotically normal, we have that the bit $u_i^{(r)}$ will be stable only if the mean to standard deviation given by $(n-1)^{1/2}/(m-1)^{1/2}$ is large. Thus, as long as the storage capacity of the system is not overloaded, we expect the datums to be stable in probability. Note that the simple argument used above immediately implies that we require $m = o(n)$. Stable datums tend to exhibit attraction basins for this model as the interaction matrix is symmetric.

B. Storage Capacity of the Outer-Product Scheme

Let the sequence $\{m_n\}_{n=1}^{\infty}$ denote explicitly the number of datums as a function of the number of neurons n. We define the sequence of probabilities $\{P_s(n)\}_{n=1}^{\infty}$ by

$$P_S(n) = \boldsymbol{P}\{\boldsymbol{u}^{(r)} \text{ is a stable state, } r = 1, \cdots, m_n\}. \tag{3.3}$$

The following results follow from [10] and [12]. All logarithms are to base e.

Theorem 1: Let δ be a parameter with $(\log n)^{-1} \leq \delta \leq \log n$. If

$$m_n = \frac{n}{4 \log n}\left[1 + \frac{3 \log\log n + \log(128\pi\delta^2)}{4 \log n} + o\left(\frac{1}{\log n}\right)\right], \quad \text{as } n \to \infty, \tag{3.4}$$

then the probability that each of the m_n datums is strictly stable is asymptotically $e^{-\delta}$, i.e., $\boldsymbol{P}_s(n) \sim e^{-\delta}$ as $n \to \infty$.

Corollary 1: $C(n) = n/4 \log n$ is the storage capacity of the outer-product algorithm.

The capacity result was based on the requirement that all of the datums be strictly stable. A recomputation of capacity based on the less stringent requirement that one of the datums be strictly stable with high probability yields a capacity of $n/2 \log n$. (This latter requirement yields that the expected number of strictly stable datums is $m - o(m)$.) Thus requiring all the datums to be stable instead of just one reduces capacity by only a factor of a half.

C. Update Rule and Preprocessing Cost

Let $\boldsymbol{U} = [\boldsymbol{u}^{(1)}, \cdots, \boldsymbol{u}^{(m)}]$ be the $n \times m$ matrix of datums to be stored. From (3.1) we have that for direct construction of the matrix of weights,

$$\boldsymbol{W}^{\mathrm{op}} = \boldsymbol{W}^{\mathrm{op}}[m] = \boldsymbol{U}\boldsymbol{U}^T - m\boldsymbol{I} \tag{3.5}$$

where $\boldsymbol{I}$ is the $n \times n$ identity matrix. The following assertion follows directly.

Theorem 2: The outer-product storage algorithm is local; specifically,

$$\boldsymbol{W}^{\mathrm{op}}[k] = \boldsymbol{W}^{\mathrm{op}}[k-1] + \boldsymbol{u}^{(k)}\boldsymbol{u}^{(k)^T} - \boldsymbol{I}, \qquad k \geq 1, \tag{3.6}$$

with $\boldsymbol{W}^{\mathrm{op}}[0] = \boldsymbol{0}$.

Matrix updates to include new datums can hence be done in place at low computational expense for the outer-product algorithm. In fact, let N^{op} be the number of elementary operations required to compute the weight matrix $\boldsymbol{W}^{\mathrm{op}}$ when all the m datums are to be directly stored by the outer-product algorithm according to (3.5): also let $N^{\mathrm{op}}[k]$ denote the number of elementary operations needed to compute the update of the weight matrix according to (3.6) when datum $\boldsymbol{u}^{(k)}$ is to be included in the stored set of datums $\boldsymbol{u}^{(1)}, \cdots, \boldsymbol{u}^{(k-1)}$. We now have the following cost estimates which follow directly from (3.5) and (3.6), and the observation that the matrices are symmetric and zero-diagonal.

Corollary 2:

$$N^{\mathrm{op}} = \frac{mn^2}{2} - \frac{mn}{2},$$

$$N^{\mathrm{op}}[k] = \frac{n^2}{2} - \frac{n}{2}, \qquad k \geq 1.$$

In the above assertion we estimated the number of multiplications. Using real additions as a measure of elementary operations yields the same order of magnitude of required elementary operations. Also note that $mN^{\mathrm{op}}[k] = N^{\mathrm{op}}$, so that for this instance the total computational labor involved is the same whether we compute $\boldsymbol{W}^{\mathrm{op}}$ directly or recursively. Providing an update capability thus does not cost any more.

IV. Log Capacity and Linear Capacity

Each datum comprises n bits so that we can store up to $n^2/4\log n$ bits using the outer-product algorithm. In the densely interconnected model of neural network that we utilize, we have n^2 possible interconnections. The outer-product scheme is hence capable of storing of the order of $1/4\log n$ bits of information per connection.

Thus, while the number of datums that can be stored grows at a reasonable rate with n, in terms of interconnections, however, an ever-decreasing amount of information is stored per interconnection with every added interconnection. This is of concern, especially in digital implementations, where the cost and circuit complexity of VLSI circuits increases rapidly with the number of interconnections required [13].

We have n^2 *degrees of freedom* corresponding to the matrix of weights $[w_{ij}]$ with which to tailor a network for storage of a given set of datums. The intuitive idea that the storage capacity of the network is a monotone function of the available degrees of freedom—in this case the total number of possible interconnections—has been previously remarked upon by Little and Shaw [5], and quantified by Abu-Mostafa and St. Jacques [14]. The latter authors define the *information capacity* of the network as the logarithm of the total number of distinct mappings of $\mathbb{B}^n$ to $\mathbb{B}^n$ that can be made by all possible threshold logic operations of the form $\Delta \circ W$. (All such distinct mappings can be regarded as informative transitions.) They deduce the information capacity of the network to be exactly of the order of n^3 bits, and, as a corollary, conclude that the storage capacity is at best of the order of n^2.

However, the n^2 bound does not apply with the imposition of the additional requirement that the stored datums be stable. Each neuron can be formally considered to be a *threshold logic unit* [15] within the model we consider, so that the neural network is simply a large interconnected network of threshold logic units. Using this fact and ideas from combinatorial geometry, we can demonstrate that if *all* possible neural networks are allowed for consideration (i.e., we are allowed to examine all choices of matrix W), then the maximal algorithm independent storage capacity of neural networks is of the order of $2n$ datums [10], [16], [17]. Specifically, let $P(m,n)$ represent the probability that each of m randomly chosen associations of the form $u \mapsto v$ can be stored for some choice of neural network. We quote the following result (cf. [10], [16], [17]) without proof.

Theorem 3: For every $\lambda \in (0,1)$, as $n \to \infty$, the following hold:

a) $P(m,n) \to 1$ whenever $m \le 2n(1-\lambda)$, and
b) $P(m,n) \to 0$ whenever $m \ge 2n(1+\lambda)$.

For the specific case of autoassociation some strictures apply on allowable choices for W (cf. [10], [16] for details).

In [14] an alternate deterministic formulation of capacity is used wherein it is required that, for *every* choice of m datums with m less than capacity, there must exist some neural network (restricted to symmetric zero-diagonal weight matrices) in which all the m datums are stable. With this definition the authors derive an upper bound of n for capacity. However, it can be demonstrated that this bound is rather loose for the adopted definition. In particular, any two datums differing in precisely one component cannot be jointly stored as stable states in any neural network with a symmetric zero-diagonal weight matrix (cf. also [18]).

The probabilistic definition of capacity that we adopted effectively relaxes the requirement that all choices of m datums (with m less than capacity) be storable to the requirement that *almost all* choices of m datums be storable. Pathological examples of choices of datums with m less than capacity that cannot be stored form a set whose size is small compared to $\binom{2^n}{m}$ and are hence effectively ignored by our definition. Furthermore, the maximal capacity of $2n$ is a tight (probablistic) upper bound which can actually be realized in networks of sufficient size.

With the added requirement of attraction, we conjecture that the storage capacity of $2n$ datums becomes of the order of about 1 bit of information per interconnection. We anticipate that, by careful choice of linear transformation, we can store up to about 1 bit of information per interconnection.

V. Spectral Algorithms

A. A New Perspective of the Outer-Product Scheme

We again assume that m datums $u^{(1)}, u^{(2)}, \cdots, u^{(m)} \in \mathbb{B}^n$ have been chosen randomly. For strict stability, we require that $(\Delta \circ W)(u^{(r)}) = u^{(r)}$ for $r = 1, \cdots, m$. Specifically, if $Wu^{(r)} = v^{(r)} \in \mathbb{R}^n$, we require that $\operatorname{sgn}(v_i^{(r)}) = u_i^{(r)}$ for each $i = 1, \cdots, n$.

For the outer-product scheme for generating the elements of the weight matrix, we have from (3.1)

$$\begin{aligned}(W^{\text{op}}u^{(r)})_i &= \sum_{j=1}^{n} w_{ij}^{\text{op}} u_j^{(r)} \\ &= \sum_{\substack{j=1\\ j\neq i}}^{n} \sum_{s=1}^{m} u_i^{(s)} u_j^{(s)} u_j^{(r)} \\ &= (n-1)u_i^{(r)} + \sum_{s\neq r}\sum_{j \neq i} u_i^{(s)} u_j^{(s)} u_j^{(r)} \\ &= (n-1)u_i^{(r)} + \delta u_i^{(r)} \end{aligned} \tag{5.1}$$

where $E(\delta u_i^{(r)}) = 0$, $\operatorname{var}(\delta u_i^{(r)}) = (n-1)(m-1)$. Hence

$$\frac{E\left(|(n-1)u_i^{(r)}|\right)}{\left(\operatorname{var}\left(\delta u_i^{(r)}\right)\right)^{1/2}} = \frac{(n-1)^{1/2}}{(m-1)^{1/2}} \to \infty \text{ as } n \to \infty,$$

where we require that $m = o(n)$ from Corollary 1 so that the datums are stable with high probability. Hence we can write

$$W^{\text{op}}u^{(r)} = (n-1)u^{(r)} + \delta u^{(r)}$$

where $\delta u^{(r)}$ has components $\delta u_i^{(r)}$ whose contributions are small compared to $u_i^{(r)}$, at least in a probabilistic sense. In essence then, the datums $u^{(r)}$ are "eigenvectors-in-mean" or "pseudo-eigenvectors" of the linear operator W^{op}, with "pseudo-eigenvalues" $n-1$.

B. Constructive Spectral Approaches

In this section we demonstrate constructive schemes for the generation of the weight matrix which yield a larger capacity than the outer-product scheme. This construction ensures that the given set of datums is stable under the algorithm; specifically, we obtain linear operators W^s which ensure that the conditions $\operatorname{sgn}(W^s u^{(r)})_i = u_i^{(r)}$, $i = 1,\cdots,n$, $r = 1,\cdots,m$, are satisfied for $m \le n$. The construction entails an extension of the approach outlined in the previous section so that the datums $u^{(r)}$ are true eigenvectors of the linear operator W^s [10], [19]. Related approaches include those of Kohonen [20], who considers a purely linear mapping that is optimal in the mean-square sense, and Poggio's polynomial mapping technique [9]. Other schemes formally related to our approach are the interesting orthogonalization techniques proposed by Amari [21] and Personnaz *et al.* [22]. We now utilize a result due to Komlós on binary n-tuples, to establish two results which have a direct bearing on the construction of the weight matrix.

Proposition 1:

a) For all randomly chosen binary $(-1,1)$ n-tuples $u^{(1)}, u^{(2)}, \cdots, u^{(m)} \in \mathbb{B}^n$ with $m \le n$, define the $n \times m(-1,1)$ matrix $U = [u^{(1)}u^{(2)} \cdots u^{(m)}]$. Then $P\{\operatorname{rank}(U) = m\} \to 1$ as $n \to \infty$.

b) Let Ξ_n be the family of bases for $\mathbb{R}^n$ with all basis elements constrained to be binary n-tuples; (i.e., $E = \{e_1, e_2, \cdots, e_n\} \in \Xi_n$ if and only if $e_1 e_2, \cdots, e_n \in \mathbb{B}^n$ are linearly independent). Then asymptotically as $n \to \infty$, almost all vectors $u \in \mathbb{B}^n$ have a representation of the form

$$u = \sum_{j=1}^{n} \alpha_j e_j, \qquad \alpha_j \ne 0 \text{ for each } j = 1,\cdots,n, \quad (5.2)$$

for almost all bases E in Ξ_n.

We use these results to establish the validity of the following schemes for constructing the weight matrix W_s. Fix $m \le n$, and let $\lambda^{(1)}, \lambda^{(2)}, \cdots, \lambda^{(m)} \in \mathbb{R}^+$ be fixed (but arbitrary) positive real numbers. Let $u^{(1)}, u^{(2)}, \cdots, u^{(m)} \in \mathbb{B}^n$ be the m randomly chosen datums to be stored in the memory. In what follows we formally define two "spectral" formulations for the interconnection matrix.

Strategy 1: Define the $m \times m$ diagonal matrix $\Lambda = dg[\lambda^{(1)}, \lambda^{(2)}, \cdots, \lambda^{(m)}]$, and the $n \times m(-1,1)$ matrix of datums $U = [u^{(1)}u^{(2)} \cdots u^{(m)}]$. Set $W^s = U\Lambda(U^T U)^{-1}U^T$.

Strategy 2: Choose any $(n-m)$ vectors $u^{(m+1)}, u^{(m+2)}, \cdots, u^{(n)} \in \mathbb{B}^n$, such that the vectors $u^{(1)}, \cdots, u^{(m)}, u^{(m+1)}, \cdots, u^{(n)}$ are linearly independent. Define the augmented $n \times n$ diagonal matrix Λ_a, and the augmented $n \times n(-1,1)$ matrix U_a by $\Lambda_a = dg[\lambda^{(1)}, \cdots, \lambda^{(m)}, 0, \cdots, 0]$, and $U_a = [u^{(1)} \cdots u^{(m)}u^{(m+1)} \cdots u^{(n)}]$. Set $W^s = U_a \Lambda_a U_a^{-1}$.

The crucial assumption of linear independence of the datums in the formal definitions above is justified by Proposition 1-a). Specifically, $\operatorname{rank}(U) = m$ and $\operatorname{rank}(U_a) = n$ for almost any choice of datums, so that the inverses are well defined.

Note that in both strategies, $\{\lambda^{(1)}, \lambda^{(2)}, \cdots, \lambda^{(m)}\}$ is the spectrum of the linear operator W^s, and the datums $u^{(1)}, u^{(2)}, \cdots, u^{(m)}$ are the corresponding eigenvectors. Alternative schemes can also be obtained by combining the two strategies.

Theorem 4: The storage capacity of all spectral strategies is linear in n; specifically, $C(n) = n$ for all spectral strategies.

Remarks

1) Additional stable states are created by both strategies: For simplicity, let us consider the eigenvalues $\lambda^{(r)}$ to be equal to some value $\lambda > 0$. Let $\Gamma = \operatorname{span}\{u^{(1)}, u^{(2)}, \cdots, u^{(m)}\} \subset \mathbb{R}^n$. Clearly, if u belongs to $\Gamma \cap \mathbb{B}^n$, then u is also stable for both strategies. By Proposition 1-b), however, there will not be many such stable states created if $m < n$. In addition there will be some more stable states created in more or less random fashion in both strategies. Such stable states satisfy the more general stability requirement: $\operatorname{sgn}(W^s u)_i = u_i$ for each $i = 1,\cdots,n$, and are not eigenvectors of the linear operator W^s.

2) Both strategies have some capacity for positive recognition of unfamiliar starting states: Let $\Phi \subset \mathbb{R}^n$ denote the null space of W^s. For strategy 1, Φ is the orthogonal subspace to Γ, while for strategy 2, $\Phi = \operatorname{span}\{u^{(m+1)}, u^{(m+2)}, \cdots, u^{(n)}\}$. If $u \in \Phi$, we have $W^s u = \mathbf{0}$. Consequently, at least for a synchronous algorithm, $(\Delta \circ W^s)$ will iteratively map u to some vector $u^{(0)} \in \mathbb{B}^n$ for all $u \in \Phi$. The vector $u^{(0)}$ in this case represents a positive indication that the starting state was not familiar.

C. Recursive Constructions and Cost

Note that there is a computational advantage in choosing strategy 1 as it involves just an $m \times m$ matrix inversion as opposed to the $n \times n$ matrix inversion required in strategy 2. In what follows we assume that we construct W^s according to the prescription of strategy 1.

The linear capacity evidenced in the spectral schemes yields considerable improvement over the (inverse-) logarithmic capacity of the outer-product algorithm. The improvement in capacity, however, is at the cost of increased complexity in the construction of the weight matrix. In general, this increased complexity implies that simple update rules like (3.6) cannot be found. However, for the particular (but important) case where the spectrum is chosen to be m-fold degenerate, some simplicity is attained.

Theorem 5: For a constant choice of eigenvalues, $\lambda^{(k)} = \lambda > 0$, $k \ge 1$, the pseudo-inverse spectral storage algorithm of strategy 1 is memoryless; specifically, let $e^{(k)}$ be the n-vector given by

$$e^{(k)} = (\lambda I - W^s[k-1])u^{(k)}.$$

Then

$$\boldsymbol{W}^s[k] = \boldsymbol{W}^s[k-1] + \frac{\boldsymbol{e}^{(k)}\boldsymbol{e}^{(k)T}}{\boldsymbol{u}^{(k)^T}\boldsymbol{e}^{(k)}}, \qquad k \geq 1 \qquad (5.3)$$

with $\boldsymbol{W}^s[0] = \boldsymbol{0}$.

Note that the algorithm is not local—each component of the updated matrix requires knowledge of the entire previous weight matrix—so that in-place updates are not possible. For the general spectral algorithm with unequal eigenvalues, a recursion can still be provided (cf. Appendix II) to construct updated matrices. However, the spectral algorithm is not memoryless for the general case.

The matrix inversions that have to be performed for the spectral strategies pose a much more involved computational task than the simple Kronecker products required in the outer-product algorithm. Symmetries in the structure can, however, be utilized to ease the computational burden. We again restrict our attention to the pseudo-inverse formulation of strategy 1.

Let N^s denote the number of elementary operations required to compute the weight matrix $\boldsymbol{W}^s$ directly from the m datums to be stored, and let $N^s[k]$ denote the number of elementary operations needed to compute the update of the weight matrix according to (5.3). Again counting the number of multiplications (the number of additions is of the same order), we get the following cost estimates.

Corollary 3:

$$N^s = mn^2 + m^2n + \frac{m^3}{2} + O(n^2)$$

$$N^s[k] = 2n^2 + 2n, \qquad k \geq 1.$$

Note that for all choices of $m < n$, we have $mN^s[k] \gtrsim N^s$, so that, especially for large n, the recursive construction of $\boldsymbol{W}^s$ through the updates (5.3) is computationally more expensive than the direct estimation of $\boldsymbol{W}^s$. There is thus an additional cost to be paid if updating capability is desired.

D. Exhibition of Attractorlike Behavior by Datums

We now probe the question of whether or not a region of attraction exists around each datum. We will restrict ourselves to the case where $\boldsymbol{W}$ has an m-fold degenerate spectrum. For definiteness, we consider variants of the matrix $\boldsymbol{W}$ chosen according to the pseudoinverse scheme of strategy 1.

As in the case of the outer product algorithm, the signal-to-noise ratio (SNR) serves as a good *ad hoc* measure of attraction capability. Specifically, let $\lambda > 0$ be the m-fold degenerate eigenvalue of $\boldsymbol{W}$. Then we claim that $\|\boldsymbol{Wx}\| \leq \lambda\|\boldsymbol{x}\|$ for all $\boldsymbol{x} \in \mathbb{R}^n$. To see this we write $\boldsymbol{x}$ in the form $\boldsymbol{x} = \boldsymbol{x}_1 + \boldsymbol{x}_2$, where $\boldsymbol{x}_1 \in \Gamma$ and $\boldsymbol{x}_2 \in \Phi$. (Recall that we defined $\Gamma = \operatorname{span}\{\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \cdots, \boldsymbol{u}^{(m)}\}$, and Φ was the orthogonal subspace to Γ.) Then $\boldsymbol{Wx} = \boldsymbol{Wx}_1 = \lambda\boldsymbol{x}_1$. Also $\|\boldsymbol{x}\|^2 = \|\boldsymbol{x}_1\|^2 + \|\boldsymbol{x}_2\|^2 \geq \|\boldsymbol{x}_1\|^2$. Hence $\|\boldsymbol{Wx}\| = \lambda\|\boldsymbol{x}_1\| \leq \lambda\|\boldsymbol{x}\|$.

Now, if $\boldsymbol{u}^{(r)}$ is a datum and $\boldsymbol{u} = \boldsymbol{u}^{(r)} + \delta\boldsymbol{u}$, then $\boldsymbol{Wu} = \lambda\boldsymbol{u}^{(r)} + \boldsymbol{W}\delta\boldsymbol{u}$, so that $\boldsymbol{u}$ will be mapped into $\boldsymbol{u}^{(r)}$ by the adaptation algorithm only if the perturbation team $\boldsymbol{W}\delta\boldsymbol{u}$ is sufficiently small. As a measure of the strength of the perturbation, we define the SNR by $\|\boldsymbol{Wu}^{(r)}\|/\|\boldsymbol{W}\delta\boldsymbol{u}\| = \lambda\sqrt{n}/\|\boldsymbol{W}\delta\boldsymbol{u}\|$; a high SNR implies that the perturbation term is weak, and conversely. From the discussion in the preceding paragraph, we have that the SNR $\geq \sqrt{n}/\|\delta\boldsymbol{u}\|$. If d denotes the Hamming distance between $\boldsymbol{u}$ and $\boldsymbol{u}^{(r)}$, then $\|\delta\boldsymbol{u}\| = 2\sqrt{d}$. For vectors $\boldsymbol{u}$ in the immediate neighborhood of $\boldsymbol{u}^{(r)}$, we have $d \ll n$. We hence obtain a large SNR which is lower-bounded by $\sqrt{n}/2\sqrt{d}$, which is indicative of a small perturbation term (compared to the "signal" term).

The SNR argument provides a quantitative measure of the attraction radius. The existence of attraction basins is, however, ensured only in probability, insofar as we accept the SNR as an accurate barometer of attraction behavior. For the case where $\boldsymbol{W}$ has an m-fold degenerate spectrum, a direct analytical argument can be supplied for the existence of a flow in the state space towards stable states whatever the mode of operation adopted [23]. We use the following facts.

Fact: If the spectrum of $\boldsymbol{W}$ generated by strategy 1 is m-fold degenerate, then $\boldsymbol{W}$ is symmetric, nonnegative definite.

Proof: Let $\lambda > 0$ be the m-fold degenerate eigenvalue of $\boldsymbol{W}$. Then $\boldsymbol{W}^T = \boldsymbol{W} = \lambda\boldsymbol{U}(\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T$, so that $\boldsymbol{W}$ is symmetric. Furthermore, for any $\boldsymbol{u}$, set $\boldsymbol{u} = \boldsymbol{u}_1 + \boldsymbol{u}_2$, where $\boldsymbol{u}_1$ lies in the degenerate subspace of eigenvectors Γ and $\boldsymbol{u}_2$ lies in the orthogonal subspace to Γ. Then $\langle \boldsymbol{u}, \boldsymbol{Wu} \rangle = \lambda\|\boldsymbol{u}_1\|^2 \geq 0$.

Now, for any mode of operation, and for each state $\boldsymbol{u}$, let the algorithm result in a flow in state space defined by $\boldsymbol{u} \mapsto \boldsymbol{u} + \delta\boldsymbol{u}$, where $\delta\boldsymbol{u}$ is an n-vector whose components take on values -2, 0, and $+2$ only. The change in energy $\delta E(\boldsymbol{u}) = E(\boldsymbol{u} + \delta\boldsymbol{u}) - E(\boldsymbol{u})$ is then given by

$$\begin{aligned} \delta E(\boldsymbol{u}) &= -\frac{1}{2}[\langle \delta\boldsymbol{u}, \boldsymbol{Wu} \rangle + \langle \boldsymbol{u}, \boldsymbol{W}\delta\boldsymbol{u} \rangle + \langle \delta\boldsymbol{u}, \boldsymbol{W}\delta\boldsymbol{u} \rangle] \\ &= -\langle \delta\boldsymbol{u}, \boldsymbol{Wu} \rangle - \frac{1}{2}\langle \delta\boldsymbol{u}, \boldsymbol{W}\delta\boldsymbol{u} \rangle, \end{aligned} \qquad (5.4)$$

as $\boldsymbol{W}$ is symmetric. Now, every nonzero component of $\delta\boldsymbol{u}$ has the same sign as the corresponding component of $\boldsymbol{Wu}$ by the prescription for state changes, so that $\langle \delta\boldsymbol{u}, \boldsymbol{Wu} \rangle \geq 0$. Further, $\langle \delta\boldsymbol{u}, \boldsymbol{W}\delta\boldsymbol{u} \rangle \geq 0$ as $\boldsymbol{W}$ is nonnegative definite. Hence $\delta E(\boldsymbol{u}) \leq 0$ for every $\boldsymbol{u} \in \mathbb{B}^n$. It then follows that model trajectories in state space follow contours of decreasing energy. As the energy attains (local) minima at stable datums, basins of attraction are typically established. In the general case, however, this does not preclude the possibility of lower energy stable states being incidentally created close to a datum (cf. remarks following Theorem 4), so that the attractive flow in the region is dominated by the extraneous stable state.

Proposition 2: Global energy minima are formed at the datums for the m-fold degenerate spectral scheme of strategy 1.

This result is not true in general for the outer product scheme.

Note that as W is nonnegative definite, the energy is always nonpositive. All vectors in the null space of W have zero energy so that the flow in state space is typically away from these vectors. Vectors in the null space hence constitute *repellor states.*

When the spectrum of W is not degenerate, however, the above argument does not hold, and the algorithm does not always generate flows that decrease energy. However, a statistical argument can be adduced instead of the analytical argument above to show that such flows are typically the case.

VI. Computer Simulations

Trends observed in computer simulations for systems with state vectors of between 32 and 64 bits have bolstered our intuitive supposition that the increased storage capacity of the spectral approach (*vis-à-vis* the outer product scheme) results in significantly improved performance as an associative memory. A typical comparative plot between the two schemes is shown in Fig. 2.

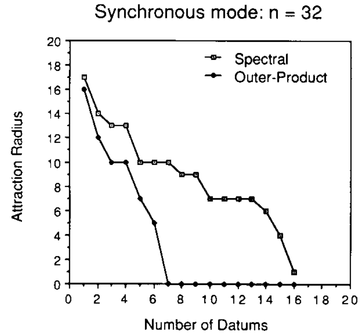

Fig. 2. Error correction in outer-product scheme compared with that in spectral scheme using equal eigenvalues.

To test the robustness of the scheme to changes in the weight matrix, we considered a modified spectral weight matrix whose elements were hard-limited to have binary values. Comparisons with hard-limited versions of the outer-product algorithm showed some superiority in attraction radius for the spectral algorithm in the cases considered, with qualitative similarity to the behavior for the non-hard-limited case (barring a slight decrease in storage capacity) as illustrated in Fig. 3.

While the datums are stable up to $m = n$ for the spectral scheme, the SNR argument indicates that attraction behavior secures for $m \leq kn$ with $k < 1$. As illustrated in Fig. 4, datums stored using the spectral algorithm were seen to exhibit some attraction behavior for $m \leq n/2$.

Small random perturbations did not have a significant effect on performance of the spectral scheme. As anticipated by the SNR argument, decreasing a datum's eigenvalue (relative to the mean) in general caused a decrease in the corresponding radius of attraction, while substantially increasing a datum's eigenvalue usually increases the radius of attraction (Fig. 5).

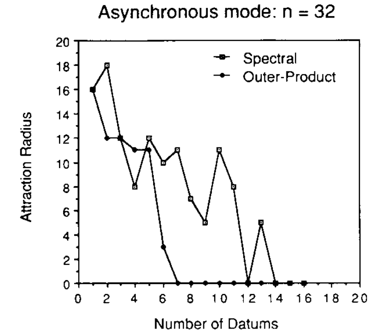

Fig. 3. Error correction in hard-limited outer-product scheme compared with that in hard-limited spectral scheme with equal eigenvalues: hard-limited version of both schemes is generated by replacing each individual component of original weight matrix by sign (− 1 or 1) of that component.

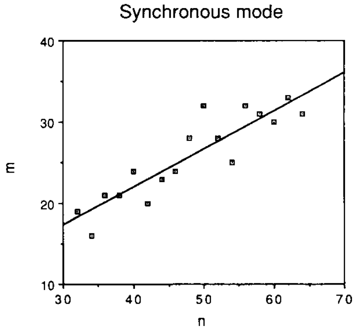

Fig. 4. Number of datums m that can be stored in spectral scheme (using equal eigenvalues) with attraction over unit Hamming distance plotted as function of number of neurons n.

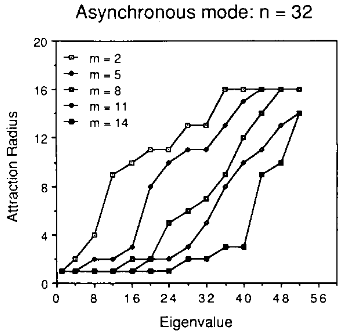

Fig. 5. Attraction radius of typical datum plotted as function of its eigenvalue in spectral scheme. Several curves are generated by varying number of datums stored as parameter, fixing eigenvalues of all other datums equal to n (in this case 32).

The figures are plotted for a typical choice of datum and "error" vectors chosen randomly at the prescribed Hamming distances. However, simulations on a variety of datums with different choices of error vectors indicate that the plots (sans the fluctuations) are quite representative of the average attraction behavior of datums under the algorithm. Also, synchronous and asynchronous modes of operation were seen to yield virtually identical attraction behavior.

VII. Conclusion

A number of viable implementations ranging from conventional digital circuitry to analog systems can be envisaged for such associative memories based on simple neural models. Recently proposed optical implementations of such models [24] are particularly exciting in this regard. For large systems, digital implementations may founder upon the problems of full interconnection since the cost and circuit complexity of VLSI circuits usually are driven by the wiring or interconnection problem [13]. Optical systems, in contrast, have a built-in capacity for global communication; accordingly, large associative nets with high performance and rapid convergence may be realized optically.

The relatively simple construction of the linear transformation by means of outer products yields surprisingly good performance and has a reasonably large storage capacity of $(n/4\log n)$. The spectral approach to constructing the linear transformation is more complex in structure, but results in considerable improvement in performance, with a storage capacity linear in n. While the spectral capacity is nearly optimal, the question remains whether or not different (more optimal) choices of linear transformation could affect a substantial improvement in attraction performance.

The larger storage capacity of the spectral scheme is reflected in increased preprocessing costs for computing the components of the weight matrix. For m of the order of $n/4\log n$, however, the spectral algorithm requires only about twice as many elementary operations as the outer-product algorithm; for m of the order of n, the spectral algorithm requires about five times as many elementary operations as the outer-product algorithm which, however, does not function well for this range of m.

Acknowledgment

We wish to thank our colleagues Profs. Y. S. Abu-Mostafa, N. Farhat, J. J. Hopfield, R. J. McEliece, and E. C. Posner, and Mr. J. Hong with whom we have had many illuminating discussions on this subject. Our thanks also go to Dr. R. Winslow who kindly made available to us an analytical proof demonstrating that the degenerate spectral scheme always enters a stable state, and to our reviewers who made several useful suggestions for improving the clarity of the presentation.

Appendix I

Proof of Proposition 1

a) This is essentially Komlós' result [25]. Let A_n denote the number of singular $n \times n$ matrices with binary elements $(-1,1)$. Komlós demonstrated that

$$\lim_{n\to\infty} \frac{A_n}{2^{n^2}} = 0. \tag{A.1}$$

(Komlós' result was for $n \times n$ (0,1) matrices, but it holds equally well for $n \times n$ $(-1,1)$ matrices.) Let $A_{n,m}$ denote the number of $n \times m(-1,1)$ matrices with rank strictly less than m. We have that $A_{n,m}2^{n(n-m)} \le A_n$, so that, from (A.1), $A_{n,m}2^{-nm} \to 0$ as $n \to \infty$. It then follows that asymptotically as $n \to \infty$, almost all $n \times m(-1,1)$ matrices with $m \le n$ are full rank. This proves the first part of the proposition.

b) We first estimate the cardinality of Ξ_n as follows. Let $\Theta_n = \{T = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n\} \subset \mathbb{B}^n\text{: } T \text{ is a linearly dependent set}\}$. We have

$$\begin{aligned}|\Xi_n| &= \binom{2^n}{n} - |\Theta_n| \\ &= \binom{2^n}{n}\left[1 - \frac{n!|\Theta_n|}{2^{n^2}\left(1-\frac{1}{2^n}\right)\left(1-\frac{2}{2^n}\right)\cdots\left(1-\frac{n-1}{2^n}\right)}\right].\end{aligned} \tag{A.2}$$

Let $T = \{\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n\} \in \Theta_n$. Then $[\boldsymbol{d}_1\boldsymbol{d}_2 \cdots \boldsymbol{d}_n]$ is a singular matrix. Each permutation of the column vectors $\boldsymbol{d}_1, \boldsymbol{d}_2, \cdots, \boldsymbol{d}_n$ yields another distinct singular matrix. Since the column vectors are all distinct, we have $n!|\Theta_n| \le A_n$. We further have

$$\begin{aligned}&\left(1-\frac{1}{2^n}\right)\left(1-\frac{2}{2^n}\right)\cdots\left(1-\frac{n-1}{2^n}\right)\\ &\qquad > \left(1-\frac{n-1}{2^n}\right)^n > \left(1-\frac{n(n-1)}{2^{n-1}}\right).\end{aligned}$$

Combining these results with (A.2) we get

$$1 - \frac{A_n}{2^{n^2}}\frac{1}{\left(1-\frac{n(n-1)}{2^n-1}\right)} \le \frac{|\Xi_n|}{\binom{2^n}{n}} \le 1.$$

Define the sequence $\{\kappa_n\}$ by

$$\kappa_n = 1 - \frac{A_n}{2^{n^2}}\frac{1}{\left(1-\frac{n(n-1)}{2^{n-1}}\right)}. \tag{A.3}$$

Then from (A.1) we have that $\kappa_n \to 1$ as $n \to \infty$.

Define a sequence of random variables $\{S_n\}_{n=1}^{\infty}$ such that S_n takes on the value 0 if a randomly chosen binary n-tuple $\boldsymbol{u} \in \mathbb{B}^n$ has the representation (5.2) in a randomly chosen basis $E \in \Xi_n$, and 1 otherwise. To complete the proof, it suffices to show that $\boldsymbol{E}\{S_n\} = \boldsymbol{P}\{S_n = 1\} \to 0$ as $n \to \infty$.

Fix $\boldsymbol{u} \in \mathbb{B}^n$, $E \in \Xi_n$, and assume that (5.2) does not hold. Then $\exists j \in \{1, 2, \cdots, n\}$ such that $\alpha_j = 0$. Assume without loss of generality that $\alpha_n = 0$. Then

$$\boldsymbol{u} = \sum_{j=1}^{n-1} \alpha_j \boldsymbol{e}_j, \qquad \alpha_j \ge 0. \tag{A.4}$$

We hence have that $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_{n-1}, \boldsymbol{u}\} \in \boldsymbol{\Theta}_n$. An overestimate for the number of choices of $\boldsymbol{u}$ and E such that (A.4) holds is $\binom{2^n}{n-1}2^n$. Also, the total number of ways that we can choose $E \in \Xi_n$, and $\boldsymbol{u} \in \mathbb{B}^n$ is $|\Xi_n|2^n$. Hence, from this and (A.3), we have

$$\boldsymbol{P}\{S_n = 1\} \le \frac{\binom{2^n}{n-1}}{|\Xi_n|} \le \frac{\binom{2^n}{n-1}}{\binom{2^n}{n}}\frac{1}{\kappa_n} \le \frac{n2^{-n}}{\kappa_n\left(1-\frac{n-1}{2^n}\right)}.$$

By definition of κ_n, we then have that $\boldsymbol{P}\{S_n = 1\} \to 0$ as $n \to \infty$. □

Proof of Proposition 2

For each datum $\boldsymbol{u}^{(r)}$, the energy is given by

$$E(\boldsymbol{u}^{(r)}) = -\frac{1}{2}\left\langle \boldsymbol{u}^{(r)}, \boldsymbol{W}\boldsymbol{u}^{(r)}\right\rangle = -\frac{\lambda n}{2}.$$

Let $\boldsymbol{u} \in \mathbb{B}^n$ be arbitrary. We can write $\boldsymbol{u}$ in the form $\boldsymbol{u} = \sum_{r=1}^{m} \alpha^{(r)}\boldsymbol{u}^{(r)} + \boldsymbol{u}_0$, where $\boldsymbol{u}_0$ is a vector in the subspace orthogonal to the space spanned by the m datums $\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \cdots, \boldsymbol{u}^{(m)}$, and $\alpha^{(r)}$ are real scalars. Then the energy is given by

$$\begin{aligned} E(\boldsymbol{u}) &= -\frac{1}{2}\langle \boldsymbol{u}, \boldsymbol{W}\boldsymbol{u}\rangle = -\frac{1}{2}\left\langle \sum_{r=1}^{m} \alpha^{(r)}\boldsymbol{u}^{(r)} + \boldsymbol{u}_0, \sum_{r=1}^{m} \lambda\alpha^{(r)}\boldsymbol{u}^{(r)}\right\rangle \\ &= -\frac{\lambda}{2}\left\| \sum_{r=1}^{m} \alpha^{(r)}\boldsymbol{u}^{(r)}\right\|^2 \\ &\geq -\frac{1}{2}\lambda\|\boldsymbol{u}\|^2 \\ &= -\frac{\lambda n}{2}, \end{aligned}$$

since

$$\|\boldsymbol{u}\|^2 = \| \sum_{r=1}^{m} \alpha^{(r)}\boldsymbol{u}^{(r)}\|^2 + \|\boldsymbol{u}_0\|^2$$

by the Pythagorean theorem. □

Appendix II

Proof of Theorem 4

Assume the datums $\boldsymbol{u}^{(1)}, \cdots, \boldsymbol{u}^{(m)}$, are linearly independent (over $\mathbb{R}$). Then for strategy 1 we have

$$\begin{aligned} (\Delta \circ \boldsymbol{W}^s)\boldsymbol{u}^{(r)} &= \left(\Delta \circ \left(\boldsymbol{U}\Lambda(\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T\right)\right)\boldsymbol{u}^{(r)} \\ &= \Delta(\lambda^{(r)}\boldsymbol{u}^{(r)}) = \boldsymbol{u}^{(r)}, \qquad r = 1, \cdots, m, \end{aligned}$$

as $\lambda^{(r)} > 0$ so that $\operatorname{sgn}(\lambda^{(r)}u_i^{(r)}) = u_i^{(r)}$. Similarly, for strategy 2 we have

$$\Delta \circ \boldsymbol{W}^s\boldsymbol{u}^{(r)} = \left(\Delta \circ \left(\boldsymbol{U}_a\Lambda_a\boldsymbol{U}_a^{-1}\right)\right)\boldsymbol{u}^{(r)} = \Delta(\lambda^{(r)}\boldsymbol{u}^{(r)}) = \boldsymbol{u}^{(r)}.$$

Thus the datums are stable regardless of the strategy adopted.

The assumption of linear independence holds with probability one for large n by proposition 2. Hence almost all choices of datums are stable regardless of the adopted spectral strategy. The capacity result follows because a linear transformation can have at most n linearly independent eigenvectors in n-space. □

For the proof of Theorem 5 we will need the following lemma due to Greville [27] which yields a simple recursive construction for the pseudo-inverse of a matrix.

Lemma 1: Let $\boldsymbol{U}$ be a real $n \times m$ matrix of full rank with $m < n$, and let $\boldsymbol{U}[k]$, $k \leq m$, denote the $n \times k$ submatrix formed from the first k columns of $\boldsymbol{U}$. To each $\boldsymbol{U}[k]$ associate the $m \times n$ matrix $\boldsymbol{V}[k]$ (the pseudo-inverse of $\boldsymbol{U}[k]$) given by

$$\boldsymbol{V}[k] = \left(\boldsymbol{U}[k]^T\boldsymbol{U}[k]\right)^{-1}\boldsymbol{U}[k]^T.$$

Let $\boldsymbol{u}^{(k)}$ be the kth column vector of $\boldsymbol{U}$. Then, if $\boldsymbol{U}[k]\boldsymbol{V}[k] \neq \boldsymbol{I}$,

$$\boldsymbol{V}[1] = \frac{1}{n}\boldsymbol{u}^{(1)^T}$$

and

$$\boldsymbol{V}[k] = \begin{bmatrix} \boldsymbol{V}[k-1](\boldsymbol{I} - \boldsymbol{u}^{(k)}\boldsymbol{x}^{(k)^T}) \\ \boldsymbol{x}^{(k)^T} \end{bmatrix}, \qquad k = 2, \cdots, m \quad \text{(B.1)}$$

where $\boldsymbol{x}^{(k)}$ is an n-vector defined by

$$\boldsymbol{x}^{(k)} = \frac{(\boldsymbol{I} - \boldsymbol{U}[k-1]\boldsymbol{V}[k-1])\boldsymbol{u}^{(k)}}{\boldsymbol{u}^{(k)^T}(\boldsymbol{I} - \boldsymbol{U}[k-1]\boldsymbol{V}[k-1])\boldsymbol{u}^{(k)}}. \qquad \text{(B.2)}$$

Proof of Theorem 5

For $m < n$, let $\boldsymbol{u}^{(1)}, \cdots, \boldsymbol{u}^{(m)}$ be a choice of datums. (With the same justification as before, we assume these to be linearly independent.) For $k = 1, \cdots, m$, let $\boldsymbol{U}[k]$ be the $n \times k$ matrix formed from the first k datums: $\boldsymbol{U} = [\boldsymbol{u}^{(1)} \cdots \boldsymbol{u}^{(k)}]$. Let $\boldsymbol{V}[k] = (\boldsymbol{U}[k]^T\boldsymbol{U}[k])^{-1}\boldsymbol{U}[k]$ be the pseudo-inverse of $\boldsymbol{U}[k]$, and let $\Lambda[k] = \boldsymbol{dg}(\lambda^{(1)}, \cdots, \lambda^{(k)})$ be the $k \times k$ diagonal matrix whose diagonal comprises of the (positive) eigenvalues for the first k datums. For the pseudo-inverse spectral strategy, we have

$$\boldsymbol{W}^s[k] = \boldsymbol{U}[k]\Lambda[k]\boldsymbol{V}[k].$$

By construction we have

$$\boldsymbol{U}[k] = [\boldsymbol{U}[k-1]\boldsymbol{u}^{(k)}]$$

and

$$\Lambda[k] = \begin{bmatrix} \Lambda[k-1] & \boldsymbol{0} \\ \boldsymbol{0} & \lambda^{(k)} \end{bmatrix}.$$

Choosing the n-vector $\boldsymbol{x}^{(k)}$ according to the prescription (B.2) of Lemma 1, we hence have

$$\begin{aligned} \boldsymbol{W}^s[k] &= [\boldsymbol{U}[k-1]\boldsymbol{u}^{(k)}]\begin{bmatrix} \boldsymbol{U}[k-1] & \boldsymbol{0} \\ \boldsymbol{0} & \lambda^{(k)} \end{bmatrix} \\ &\quad \cdot \begin{bmatrix} \boldsymbol{V}[k-1](\boldsymbol{I} - \boldsymbol{u}^{(k)}\boldsymbol{x}^{(k)^T}) \\ \boldsymbol{x}^{(k)^T} \end{bmatrix} \\ &= [\boldsymbol{U}[k-1]\Lambda[k-1] \quad \lambda^{(k)}\boldsymbol{u}^{(k)}] \\ &\quad \cdot \begin{bmatrix} \boldsymbol{V}[k-1](\boldsymbol{I} - \boldsymbol{u}^{(k)}\boldsymbol{x}^{(k)^T}) \\ \boldsymbol{x}^{(k)^T} \end{bmatrix} \\ &= \boldsymbol{W}^s[k-1](\boldsymbol{I} - \boldsymbol{u}^{(k)}\boldsymbol{x}^{(k)^T}) + \lambda^{(k)}\boldsymbol{u}^{(k)}\boldsymbol{x}^{(k)^T} \\ &= \boldsymbol{W}^s[k-1] + (\lambda^{(k)}\boldsymbol{I} - \boldsymbol{W}^s[k-1])\boldsymbol{u}^{(k)}\boldsymbol{x}^{(k)^T}. \quad \text{(B.3)} \end{aligned}$$

By (B.2) we have

$$\boldsymbol{u}^{(k)}\boldsymbol{x}^{(k)^T} = \frac{\boldsymbol{u}^{(k)}\boldsymbol{u}^{(k)^T}(\boldsymbol{I} - \boldsymbol{U}[k-1]\boldsymbol{V}[k-1])}{\boldsymbol{u}^{(k)^T}(\boldsymbol{I} - \boldsymbol{U}[k-1]\boldsymbol{V}[k-1])\boldsymbol{u}^{(k)}}.$$

Now, for the particular case where the eigenvalues are all equal, $\lambda^{(j)} = \lambda > 0$, $j \geq 1$, we have

$$\boldsymbol{W}^s[j] = \lambda\boldsymbol{U}[j]\boldsymbol{V}[j].$$

Hence

$$\boldsymbol{u}^{(k)}\boldsymbol{x}^{(k)^T} = \frac{\boldsymbol{u}^{(k)}\boldsymbol{u}^{(k)^T}(\lambda\boldsymbol{I} - \boldsymbol{W}^s[k-1])}{\boldsymbol{u}^{(k)^T}(\lambda\boldsymbol{I} - \boldsymbol{W}^s[k-1])\boldsymbol{u}^{(k)}}.$$

Setting

$$\boldsymbol{e}^{(k)} = (\lambda\boldsymbol{I} - \boldsymbol{W}^s[k-1])\boldsymbol{u}^{(k)}$$

in (B.3) we finally have

$$\boldsymbol{W}^s[k] = \boldsymbol{W}^s[k-1] + \frac{\boldsymbol{e}^{(k)}\boldsymbol{e}^{(k)^T}}{\boldsymbol{u}^{(k)^T}\boldsymbol{e}^{(k)}}. \qquad □$$

We will utilize the number of real multiplications as a measure of preprocessing cost and consider the pseudo-inverse spectral algorithm.

Proof of Corollary 3

We first estimate the number of elementary operations N^s required to directly compute the weight matrix $\boldsymbol{W}^s = \lambda\boldsymbol{U}(\boldsymbol{U}^T\boldsymbol{U})^{-1}\boldsymbol{U}^T$, where $\lambda > 0$ is the positive eigenvalue corresponding to the m-fold degenerate spectrum of $\boldsymbol{W}^s$. The matrix product $\boldsymbol{U}^T\boldsymbol{U}$ requires $mn(n-1)/2$ elementary operations as it is a symmetric matrix with constant diagonal n. Since $\boldsymbol{U}^T\boldsymbol{U}$ is symmetric positive-definite, its inverse can be efficiently performed using the Choleski decomposition (cf. [28]). This computation requires $m^3/2+2m^2+5m/2$ real multiplications. (This can be computed directly from the inversion algorithm given in [28].) Using the fact that $\boldsymbol{W}^s$ is symmetric, the remaining matrix products require $mn^2/2+m^2n+mn/2+n$ multiplications. The direct computation of $\boldsymbol{W}^s$ hence requires $mn^2+m^2n+m^3/2+O(n^2)$ elementary operations.

To compute $\boldsymbol{W}^s[k]$ note that the outer product $\boldsymbol{e}^{(k)}\boldsymbol{e}^{(k)^T}$, and the division by $\boldsymbol{u}^{(k)^T}\boldsymbol{e}^{(k)}$ in each require $n(n+1)/2$ multiplications, the inner product $\boldsymbol{u}^{(k)^T}\boldsymbol{e}^{(k)}$ requires n multiplications, and the estimation of $\boldsymbol{e}^{(k)} = (\lambda\boldsymbol{I} - \boldsymbol{W}^s[k-1])\boldsymbol{u}^{(k)}$ requires n^2 multiplications. □

REFERENCES

[1] D. O. Hebb, *The Organization of Behavior.* New York: Wiley, 1949.

[2] G. E. Hinton and J. A. Anderson, *Parallel Models of Associative Memory.* Hillsdale, NJ: Erlbaum, 1981.

[3] K. Nakano, "Association—A model of associative memory," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-2, no. 3, pp. 380–388, July 1972.

[4] W. A. Little, "The existence of persistent states in the brain," *Math. Biosci.*, vol. 19, pp. 101–120, 1974.

[5] W. A. Little and G. L. Shaw, "Analytic study of the memory storage capacity of a neural network," *Math. Biosci.*, vol. 39, pp. 281–290, 1978.

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

[7] H. C. Longuet-Higgins, "The non-local storage of temporal information," *Proc. Roy. Soc. B*, vol. 171, pp. 327–334, 1968.

[8] D. Gabor, "Associative holographic memories," *IBM J. Res. Devel.*, vol. 13, pp. 156–159, 1969.

[9] T. Poggio, "An optimal nonlinear associative recall," *Biol. Cybern.*, vol. 19, pp. 201–209, 1975.

[10] S. S. Venkatesh, "Linear maps with point rules: Applications to pattern classification and associative memory," Ph.D. dissertation, California Inst. Technol., Pasadena, 1987.

[11] E. Goles and G. Y. Vichniac, "Lyapunov function for parallel neural networks," in *Neural Networks for Computing: AIP Conf. Proc.*, vol. 151, 1986, pp. 165–181.

[12] 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, 1987.

[13] I. E. Sutherland and C. A. Mead, "Microelectronics and computer science," *Sci. Amer.*, vol. 237, pp. 210–228, 1977.

[14] Y. S. Abu-Mostafa and J. S. Jacques, "Information capacity of the Hopfield model," *IEEE Trans. Inform. Theory*, vol. IT-31, pp. 451–464, 1985.

[15] P. M. Lewis and C. L. Coates, *Threshold Logic.* New York: Wiley, 1967.

[16] S. S. Venkatesh, "Epsilon capacity of neural networks," in *Neural Networks for Computing: AIP Conf. Proc.*, vol. 151, 1986, pp. 440–445.

[17] S. S. Venkatesh and D. Psaltis, "Error-tolerance and epsilon capacity in neural networks," in review.

[18] S. S. Venkatesh and P. Baldi, "Fixed points for systems characterized by algebraic Hamiltonians of high order," in review.

[19] S. S. Venkatesh and D. Psaltis, "Efficient strategies for information storage and retrieval in associative neural nets," presented at Workshop on Neural Networks for Computing, Santa Barbara, CA, Apr. 1985.

[20] T. Kohonen, *Associative Memory: A System-Theoretical Approach.* Berlin, Germany: Springer-Verlag, 1977.

[21] S. Amari, "Neural theory of association and concept formation," *Biol. Cybern.*, vol. 26, pp. 175–185, 1977.

[22] L. Personnaz, I. Guyon, and G. Dreyfus, "Information storage and retrieval in spin-glass like neural networks," *J. Phys. Lett.*, vol. 46, pp. L359–L365, 1985.

[23] R. Winslow, private communication.

[24] D. Psaltis and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," *Opt. Lett.*, vol. 10, no. 2, pp. 98–100, Feb. 1985.

[25] J. Komlós, "On the determinant of (0,1) matrices," *Studia Sci. Math. Hungarica*, vol. 2, pp. 7–21, 1967.

[26] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. I. New York: Wiley, 1968.

[27] T. N. E. Greville, "Some applications of the pseudoinverse of a matrix," *SIAM Rev.*, vol. 2, pp. 15–22, Jan. 1960.

[28] A. Ralston, *A First Course in Numerical Analysis.* New York: McGraw-Hill, 1965.