The Science of Making *ERORS*: What Error Tolerance Implies for Capacity in Neural Networks

Santosh S. Venkatesh, *Member, IEEE*

***Abstract*—How does an allowed tolerance for error in output affect the computational capability of a neural network and the ability of the network to learn an underlying problem structure? The subject of this communication is the development of formal protocols for handling error tolerance which allow of a precise determination of the computational gains that may be expected. The error protocols are illustrated in the framework of a densely interconnected neural network architecture (with associative memory the putative application), and rigourous calculations of capacity are shown. Explicit capacities are also derived for the case of feedforward neural network configurations.**

***Index Terms*—Associative memory, capacity, error tolerance, neural networks, neuron.**

I. Introduction

HOW does an allowed tolerance for error in output affect the computational capability of a neural network and the ability of the network to learn an underlying problem structure? In this paper we attempt to mathematically codify the computational gains that are realized when errors are permitted in the network output. Such error tolerance may occur naturally in applications such as associative memory or pattern classification where we do not insist on accurately classifying every feature; alternatively, error tolerance may be forced upon us[1] by the inability of the neural architecture under consideration to respond accurately to *every* instance of a specific problem: as in, for example, attempting to classify two nonlinearly separable classes of points with a single separating plane (or equivalently, a single McCulloch-Pitts neuron).[2]

Anecdotal evidence exists for the premise that an allowed error tolerance can have a significant effect on computational capability. Consider, for instance, an associative memory application where it is desired to store memories as attractors in recurrent neural networks[3] whereby a linear number of component errors in any memory are corrected and the memory retrieved. Rigourous investigations by McEliece *et al.* [2] and Komlós and Paturi [3] show that the outer-product algorithm for storing memories in a recurrent network of n neurons stores exactly of the order of $n/\log n$ memories when it is required that each memory be retrieved *exactly* with no component errors. However, in earlier work, Hopfield [4] reports empirical evidence indicating that it may be possible to store a number of memories linear in n with the outer-product algorithm if errors are permitted in the retrieved memories; this was formally verified subsequently by Newman [5] who demonstrated a lower bound linear in n on the memory storage capacity if errors are permitted in retrieval. Thus, an allowed error tolerance effects a substantial improvement in storage capacity from sublinear in n to at least linear in n for the outer-product algorithm.

Our purpose here is to attempt to quantify the maximal gains in capacity that can accrue for *any* algorithm if errors are allowed in the outputs of a given neural network architecture. In particular, for recurrent neural networks of n neurons (and any choice of algorithm) we settle the issue of whether Newman's linear lower bounds can be substantially improved upon to allow memory storage capacities which increase faster than linearly in n if errors are permitted in the recall of the stored memories.

Let us consider the associative memory application in some more detail. Let $\boldsymbol{u}^1, \cdots, \boldsymbol{u}^m$ be a random selection of m memories drawn from the vertices of the cube $\{-1, 1\}^n$. For associative memory it is desired that a random probe differing from a memory in no more than ρn components (for some choice of $0 \leq \rho < 1/2$) is mapped into the memory. In other words, we require the memories to be *attractors* over a Hamming ball of radius ρn.

Consider now a recurrent neural network of n neurons where, at any epoch, the binary n-tuple of neural outputs is fed back to constitute the neural inputs for the next epoch. The degrees of freedom in this dynamical system reside in the specification of the weighting factors linking neural outputs to inputs. Each specification of interconnection weights results in the specification of a dynamics characterized by a family of trajectories in a state space of the vertices of the cube $\{-1, 1\}^n$. The storage of memories as attractors in such a structure is accomplished if, for a choice of interconnection weights, trajectories originating in a Hamming ball of radius ρn at the memories ultimately terminate at the memories. As we will see later in Corollary 4.2, it is not possible to store more than a linear number (in n) of random memories as attractors in a recurrent neural network of n neurons.

Manuscript received July 1, 1991. This work was supported by the Air Force Office of Scientific Research under Grant AFOSR 89-0523. This paper was presented in part at the Workshop on Neural Networks for Computing, Snowbird, UT, 1986 [1].

The author is with the Department of Electrical Engineering, University of Pennsylvania, Philadelphia, PA 19104.

IEEE Log Number 9106254.

[1] Some men are born into errors, others achieve errors, and some have errors thrust upon 'em.

[2] In the classical modeling of McCulloch and Pitts, a formal neuron is a linear threshold element which produces a binary output according to the sign of a linear form of the inputs.

[3] We restrict attention to the subclass of fully-interconnected recurrent neural networks where the output of each neuron in the network is fed back as input to all neurons in the network.

An allowed error tolerance in this context permits a fraction ϵ, $0 \leq \epsilon \leq \rho < 1/2$ of errors in the retrieval of any memory. This situation corresponds to requiring points in the Hamming ball of radius ρn at a memory to be ultimately mapped into a (smaller) ball of radius ϵn at the memory. To obviate technical nuisances, we also insist that trajectories remain confined to the ϵn ball at a memory once they enter the ball; in particular, trajectories originating at a point within the ϵn ball at a memory never leave the ball. This error tolerance scenario is indicated schematically in Fig. 1.

The capacity question is now to determine the largest allowable rate of growth of the number of memories m with the number of neurons n such that it can be guaranteed with high probability that trajectories originating in a Hamming ball of radius ρn at any memory are ultimately confined within a Hamming ball of radius ϵn at the memory. The following plausibility argument indicates that the flexibility inherent in allowing error tolerance may result in substantial gains in capacity over the error free case.

Let us simplify the problem by considering trajectories originating at a memory, say $\boldsymbol{u}^\alpha \in \{-1,1\}^n$. If the trajectory is to be confined within an ϵn ball at the memory, then clearly a necessary condition is that the *first* synchronous step in the trajectory not lead to a point outside the ball of radius ϵn at the memory. However, any transition $\boldsymbol{u}^\alpha \mapsto \boldsymbol{v}^\alpha$ where $\boldsymbol{v}^\alpha$ differs from the memory $\boldsymbol{u}^\alpha$ in no more than ϵn components is an admissible transition (see Fig. 2).

Thus, for the m memories, there are a total of $\left[\sum_{j=0}^{\epsilon n}\binom{n}{j}\right]^m$ admissible m-sets of first synchronous transitions originating at the memories corresponding to the number of different ways of specifying "first transition points" in the m Hamming balls of radius ϵn at the memories. Let us now concentrate on the problem of realizing an admissible set of m first transitions (one for each memory) within a recurrent neural network structure. This is clearly a necessary prelude to the larger problem of associative memory with error tolerance in the following sense: if $C_\epsilon(n)$ is the largest rate of growth of m with n for which it can be guaranteed with high probability that there exists a set of neural interconnection weights for a recurrent neural network which yields an admissible m-set of first synchronous transitions originating at the memories, then $C_\epsilon(n)$ gives an upper bound for the number of memories that can be stored with an error tolerance of ϵn components in retrieval.

Our basic question now is the following: For a random selection of m memories, $\boldsymbol{u}^\alpha \in \{-1,1\}^n, \alpha = 1, \cdots, m$, and a given (fractional) error tolerance $0 \leq \epsilon < 1/2$, is there any choice of neural interconnection weights for a recurrent neural network which will result in an admissible m-set of first synchronous transitions, $\boldsymbol{u}^\alpha \mapsto \boldsymbol{v}^\alpha, \alpha = 1, \cdots, m$, where each $\boldsymbol{v}^\alpha$ differs from the corresponding memory $\boldsymbol{u}^\alpha$ in no more than ϵn components? Clearly now, $C_\epsilon(n)$ is at least as large as $C_0(n)$, the largest rate of growth of m with n for which, with high probability, there exists a choice of interconnections for a recurrent neural network such that the m randomly chosen memories are fixed points of the network, i.e., can be retrieved with no component errors. Consider now a choice of

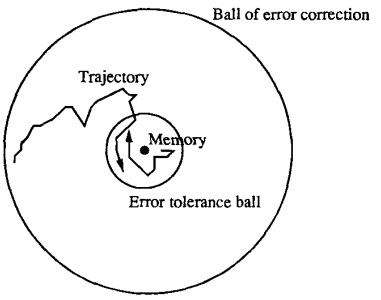

Fig. 1. A schematic demonstrating error correction (attraction) and error tolerance for a memory. Points in the Hamming ball of radius ρn at the memory lie on trajectories which are eventually confined in the (smaller) Hamming ball of radius ϵn at the memory.

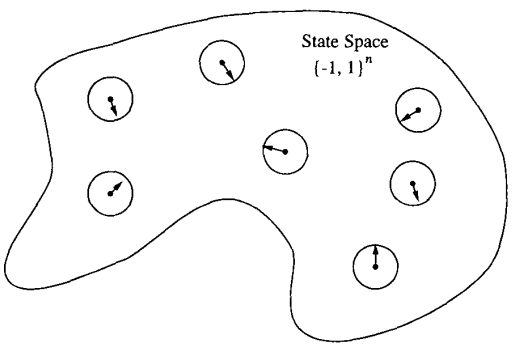

Fig. 2. A schematic showing a set of "admissible" transitions starting from a memory. Each such transition from a memory must result in a new vertex of the cube $\{-1,1\}^n$ which differs from the memory in no more than ϵn components.

m much larger than $C_0(n)$. For such a choice of m, any given m-set of admissible transitions, $\boldsymbol{u}^1 \mapsto \boldsymbol{v}^1, \cdots, \boldsymbol{u}^m \mapsto \boldsymbol{v}^m$, will have only a small probability of being realized within a recurrent neural network architecture. However, as is easy to verify using Stirling's formula, the number of admissible m-sets of first synchronous transitions from the memories is $\Omega(2^{c(\epsilon)mn})$, where $c(\epsilon)$ is a fixed positive function of ϵ. Thus, there are an exponential number of possible m-sets of admissible transitions so that the probability that one or more of these sets of transitions is realizable in a recurrent network may be large even if the probability of realizing any individual set of transitions is small. It hence appears plausible that $C_\epsilon(n) \gg C_0(n)$, i.e., that potentially large gains in capacity may be possible if errors are allowed in retrieving memories.

A similar plausibility argument could be made for the improvement in capacity of any neural network architecture if a fraction ϵ of the network outputs can be in error. For instance, if we are interested in realizing a set of desired assignments $\boldsymbol{u}^\alpha \mapsto f(\boldsymbol{u}^\alpha), \alpha = 1, \cdots, m$ (where f is some underlying function from which random examples are drawn) in a feedforward neural architecture, allowing up to ϵm of these assignments to be in error again gives an exponential (in m) number of choices for specifying incorrect assignments.

The above specious arguments notwithstanding, the main

results of this paper indicate, however, that error tolerance does not buy order of magnitude improvements in capacity over the error free case in neural network architectures. In brief (pending a formal definition of capacity) error tolerance results in an improvement in the multiplicative constants, but does not change the rate of growth of capacity: specifically, the capacity $C_\epsilon(n)$ of a neural network when a fraction ϵ of the outputs are allowed to be in error is no more than $k(\epsilon)C_0(n)$ for a fixed positive function $k(\epsilon) \geq 1$.

Organization: In the next section we set up the formal neural model in the framework of a fully-interconnected network architecture for definiteness. We also introduce two protocols for error—a random and an exhaustive error protocol—and define the formal notion of capacity. The definitions here follow those developed in Venkatesh and Psaltis [6] in the analysis of reliability and error tolerance issues for computations with a single neuron. In Section III we state some preliminary technical lemmas which are central to the ensuing development. In Section IV we state and prove the main theorems on the capacity of fully-interconnected, recurrent neural networks when there is a tolerance for output error. Finally, in Section V we briefly indicate the extensions of these results to feedforward neural network architectures.

On Notation: $\mathbb{B}$ denotes the set $\{-1, 1\}$; the function $\operatorname{sgn} : \mathbb{R} \to \mathbb{B}$ is defined by $\operatorname{sgn} x = x/|x|$ if $x \neq 0$, with the nonce convention $\operatorname{sgn} 0 = 1$; logarithms are to base e; for any positive integer k, $[k]$ denotes the set $\{1, \cdots, k\}$; $c_1, c_2, c_3, \cdots$ represent absolute positive constants; for any $\boldsymbol{u} \in \mathbb{B}^n$ and any $r > 0$, $B(\boldsymbol{u}, r)$ denotes the Hamming ball of radius r at $\boldsymbol{u}$, i.e., the set of all points in $\mathbb{B}^n$ which differ from $\boldsymbol{u}$ in r or fewer components; the probability that N Bernoulli trials with probabilities p for success and $1 - p$ for failure result in k successes and $N - k$ failures is denoted by $b(k; N, p)$:

$$b(k; N, p) = \binom{N}{k} p^k (1-p)^{N-k}.$$

We will also have recourse to the following asymptotic order notations. If $\{x_n\}$ and $\{y_n\}$ are positive sequences, we denote: $x_n = O(y_n)$ if there exists $K < \infty$ such that $x_n/y_n \leq K$ for all n; $x_n = \Omega(y_n)$ if there exists $L > 0$ such that $x_n/y_n \geq L$ for all n; and $x_n = o(y_n)$ if $x_n/y_n \to 0$ as $n \to \infty$.

II. Error Tolerance: Protocols

A. Formal Neurons and Networks

A formal neuron is a linear threshold element accepting n inputs which produces a binary output according to the sign of a linear form of the inputs. In particular, a neuron is characterized by a vector of n real *weights*, $\boldsymbol{w} = (w_1 \cdots w_n)$, and, given as input a vector $\boldsymbol{u} = (u_1 \cdots u_n)$, produces a binary output $v = \operatorname{sgn} \sum_{i=1}^n w_i u_i$.[4] The neuron hence associates a *decision* or *classification*, -1 or $+1$, with each point in n-space. For any given set of points then, the neuron dichotomizes the set of points into two classes—those points mapped to $+1$ and those mapped to -1. In the geometric analogue the neuron represents a separating hyperplane in n-space defined by the (orthogonal) weight vector $\boldsymbol{w}$.

Let $\boldsymbol{U} = \{\boldsymbol{u}^1, \cdots, \boldsymbol{u}^m\}$ be an m-set of points in n-space, and, corresponding to each $\boldsymbol{u}^\alpha \in \boldsymbol{U}$, let $v^\alpha \in \mathbb{B}$ be a desired decision. The m-set of decisions naturally dichotomizes $\boldsymbol{U}$ into two sets $(\boldsymbol{U}^+, \boldsymbol{U}^-)$, where $\boldsymbol{u}^\alpha \in \boldsymbol{U}^+$ if $v^\alpha = 1$ and $\boldsymbol{u}^\alpha \in \boldsymbol{U}^-$ if $v^\alpha = -1$. We say that the dichotomy $(\boldsymbol{U}^+, \boldsymbol{U}^-)$ is *separable by a neuron* iff there exists a weight vector $\boldsymbol{w} \in \mathbb{R}^n$ such that

$$\sum_{i=1}^n w_i u_i^\alpha \begin{cases} \geq 0 & \text{if } \boldsymbol{u}^\alpha = (u_1^\alpha \cdots u_n^\alpha) \in \boldsymbol{U}^+ \\ < 0 & \text{if } \boldsymbol{u}^\alpha = (u_1^\alpha \cdots u_n^\alpha) \in \boldsymbol{U}^-, \end{cases}$$

i.e., $\operatorname{sgn} \sum_{i=1}^n w_i u_i^\alpha = v^\alpha$ for $\alpha = 1, \cdots, m$.

We will be concerned here with a fully-interconnected, recurrent network of n neurons where the neural outputs at any epoch are fed back to constitute the neural inputs for the next epoch. In particular, for any $i \in [n]$, neuron i is characterized by a set of $n - 1$ weights $\{w_{ij} : j \neq i, j \in [n]\}$, and if the vector $\boldsymbol{u} = (u_1 \cdots u_n) \in \mathbb{B}^n$ denotes the neural outputs at any epoch, at the next epoch the ith neuron then produces a binary output $u_i' = \operatorname{sgn} \sum_{j \neq i} w_{ij} u_j$.[5] The system hence evolves (synchronously) in a state space of vertices of the cube $\mathbb{B}^n$, and is completely characterized by the zero-diagonal matrix of neural interconnection weights $[w_{ij}]$.

Let $\boldsymbol{u}^1, \cdots, \boldsymbol{u}^m$ be a random m-set of memories chosen independently from the vertices of the cube. In particular, the memory components $\{u_i^\alpha : i \in [n], \alpha \in [m]\}$ are i.i.d. random variables drawn from a sequence of symmetric Bernoulli trials:

$$\boldsymbol{P}\{u_i^\alpha = 1\} = \boldsymbol{P}\{u_i^\alpha = -1\} = 1/2, \qquad i \in [n], \quad \alpha \in [m].$$

In an associative memory application, a basic deisderatum would be that all the memories are fixed points—$\boldsymbol{u}^\alpha \mapsto \boldsymbol{u}^\alpha, \alpha = 1, \cdots, m$—of the network, i.e., that there exists a zero diagonal matrix of weights $[w_{ij}]$ such that

$$\operatorname{sgn} \sum_{j \in [n] \setminus \{i\}} w_{ij} u_j^\alpha = u_i^\alpha, \qquad i \in [n], \quad \alpha \in [m].$$

If now there is an allowed tolerance for error, the fixed point requirement for the memories could be relaxed to allow "admissible" first synchronous transitions of the form $\boldsymbol{u}^\alpha \mapsto \boldsymbol{v}^\alpha$. By "admissible" we mean as before that the number of components in which the points $\boldsymbol{v}^\alpha \in \mathbb{B}^n$ are allowed to differ from the corresponding memories $\boldsymbol{u}^\alpha$ must be within a given error tolerance. We are interested in estimating the largest allowable rate of growth of m with n for which there exists a zero-diagonal network which realizes m admissible synchronous transitions $\boldsymbol{u}^\alpha \mapsto \boldsymbol{v}^\alpha$ with high probability as n grows large. In the following we define two formal error protocols which provide different notions of "admissibility."

B. Random Error Protocol

We begin by defining a protocol which randomly specifies which components of a memory are allowed to be in error by randomly labeling a set of memory components as *don't-cares*. Let $0 \leq \epsilon < 1/2$ be the fraction of errors that we

[4] This is the model of McCulloch and Pitts. A real threshold is allowed within the model but is not critical to the present discussion.

[5] To avoid trivial complications, we assume that there is no self feedback from the output of any neuron to its input. This corresponds to setting the weights $w_{ii} \equiv 0$ for $i = 1, \cdots, n$.

are willing to allow in the retrieval of any memory. For each $i \in [n]$, let $\{D_i^\alpha : \alpha \in [m]\}$ be the outcomes of m identical and independent experiments whose outcomes are subsets of $\{-1, 1\}$, and such that

$$D_i^\alpha = \begin{cases} \{u_i^\alpha\} & \text{with probability } 1 - 2\epsilon \\ \{-1, 1\} & \text{with probability } 2\epsilon. \end{cases}$$

If a sample outcome $D_i^\alpha = \{u_i^\alpha\}$, then we require that the ith neuron retrieve the ith component of memory $\boldsymbol{u}^\alpha$; if, however, the sample outcome $D_i^\alpha = \mathbb{B}$, then we associate a don't-care decision with point $\boldsymbol{u}^\alpha$: the neuron can result in either -1 or 1 as output when $\boldsymbol{u}^\alpha$ is input. We call D_i^α the *decision set* associated with memory component u_i^α; we say that D_i^α is *normal* if $D_i^\alpha = \{u_i^\alpha\}$ (i.e., memory component u_i^α has to be retrieved by the ith neuron), and D_i^α is *exceptional* if $D_i^\alpha = \mathbb{B}$ (i.e., the decision is don't-care). The idea behind defining the decision sets in this fashion is the following. For each $\alpha \in [m]$, the decision sets $\{D_i^\alpha : i \in [n]\}$ are generated independently according to the above prescription so that the expected number of normal decision sets is $(1 - 2\epsilon)n$ while the expected number of exceptional decision sets is $2\epsilon n$ for each memory. If now, ignoring components corresponding to exceptional decision sets, we design a zero-diagonal weight matrix to retrieve all components corresponding to the normal decision sets, then on average one-half of the components corresponding to exceptional decision sets will also be retrieved so that the expected number of errors in the retrieval of any memory will be only ϵn.

Definition 2.1: For each $i \in [n]$, let $\{w_{ij} : j \neq i\}$ be the set of $n - 1$ weights corresponding to the ith neuron. We then say that the ith neuron makes *ϵ-reliable decisions* on the set of memories $\{\boldsymbol{u}^1, \cdots, \boldsymbol{u}^m\}$ if

$$\operatorname{sgn} \sum_{j \in [n]}^{\{i\}} w_{ij} u_j^\alpha \in D_i^\alpha, \qquad \alpha = 1, \cdots, m.$$

If all the neurons make ϵ-reliable decisions, then, by the Borel strong law, the fraction of component errors in the retrieval of any memory is ϵ almost surely. We are hence interested in the following attribute of the m-set of memories.

EVENT $\mathcal{R}_\epsilon(n, m)$ [Random Error Protocol with Parameter ϵ]: *For each $i \in [n]$, there is a choice of weights for the ith neuron such that the neuron makes ϵ-reliable decisions on the set of memories $\{\boldsymbol{u}^1, \cdots, \boldsymbol{u}^m\}$.*

The attribute $\mathcal{R}_\epsilon(n, m)$ deals with the notion of random synchronous transitions from the memories by specifying a random choice of (on average $2\epsilon n$) don't-care components—resulting effectively in an average of ϵn component errors—for each memory. The computational gains we may expect from this protocol arise from the large number of "typical" transitions that can be specified.

C. Exhaustive Error Protocol

Consider now a protocol where the number of component errors in a memory in a single synchronous transition is not permitted to exceed ϵn; however, there is no specification or constraint on which components in a memory are allowed to be in error and we are free to examine all alternatives of specifying ϵn or fewer component errors in each memory in one synchronous step. This protocol leads to a consideration of the following attribute of the m-set of memories.

EVENT $\mathcal{E}_\epsilon(n, m)$ [Exhaustive Error Protocol with Parameter ϵ]: *For each $\alpha \in [m]$, there exists a vertex $\boldsymbol{v}^\alpha \in B(\boldsymbol{u}^\alpha, \epsilon n)$ such that the set of synchronous transitions $\{\boldsymbol{u}^\alpha \mapsto \boldsymbol{v}^\alpha : \alpha \in [m]\}$ is realized for some choice of zero-diagonal weight matrix $[w_{ij}]$ for a fully-interconnected network of n neurons.*

The attribute $\mathcal{E}_\epsilon(n, m)$ is somewhat stronger than the attribute $\mathcal{R}_\epsilon(n, m)$; instead of specifying a random set of essentially $2\epsilon n$ don't-care components for a memory (resulting in essentially ϵn component errors), we are now allowed to examine all transitions resulting in ϵn or fewer component errors for each memory and choose the most favorable specification of errors. As we saw in the Introduction, this allows us any choice from among an exponentially large number of admissible m-sets of first synchronous transitions originating at the memories.

D. Capacity Functions

The notion of capacity of a fully-interconnected neural network that we espouse is, loosely speaking, the "largest number" of random memories that can be "stored" in the network. The precise meaning we attach to "storage" of a memory depends upon the attribute $\mathcal{A}$ of interest, such as: all memories are fixed points; almost all memories are attractors over a radius ρn; there are no more than ϵn component errors in retrieving any memory. We will be interested in particular in the attributes $\mathcal{R}_\epsilon(n, m)$, the random error protocol with parameter ϵ, and $\mathcal{E}_\epsilon(n, m)$, the exhaustive error protocol with parameter ϵ. The following definition captures the notion of "largest number of memories" as a threshold function of a relevant attribute. We have explored the notion in somewhat greater generality in [7].

Definition 2.2: Let $\mathcal{A}(n, m)$ be an attribute of the m-set of memories $\{\boldsymbol{u}^1, \cdots, \boldsymbol{u}^m\}$. A sequence $C(n)$ is a *capacity function* for the attribute $\mathcal{A}(n, m)$ if, for $\lambda > 0$ arbitrarily small, as $n \to \infty$:

1) $\boldsymbol{P}\{\mathcal{A}(n, m)\} \to 1$ whenever $m \leq (1 - \lambda)C(n)$;
2) $\boldsymbol{P}\{\mathcal{A}(n, m)\} \to 0$ whenever $m \geq (1 + \lambda)C(n)$.

We say that $C(n)$ is a *lower capacity function* if it satisfies the first condition, and that $C(n)$ is an *upper capacity function* if it satisfies the second condition.

Capacity functions have been found for a variety of neural network architectures and algorithms (a survey can be found in Venkatesh [7]). These investigations into network capacity, however, have hitherto concentrated mainly on capacity functions for perfect decisions with no errors (cf. Komlós and Paturi [3] and Newman [5], however, for results on error tolerance in the outer-product algorithm). In the following we expand on our results in [1] and [6] and show capacity functions for the attributes $\mathcal{R}_\epsilon(n, m)$ and $\mathcal{E}_\epsilon(n, m)$.

III. Technical Preliminaries

Our basic technique is to replace the geometrical notion of trajectories within Hamming balls in n-space by calculations involving the tails of the binomial distribution. The following is a classical result due to Chernoff [8] which asserts exponential bounds for the binomial tails for linear deviations from the mean.

Lemma 3.1: Let $0 < p < 1$ be fixed, and let T_p and H be real-valued functions on the closed interval $[0,1]$ defined for $0 \leq c \leq 1$ by

$$T_p(c) = -c \log p - (1-c) \log(1-p),$$
$$H(c) = -c \log c - (1-c) \log(1-c).^6$$

Then for every choice of $c \in (p,1]$ and every integer N we have

$$\sum_{k=0}^{\lfloor cN \rfloor} b(k;N,p) \geq 1 - e^{-N[T_p(c)-H(c)]}.$$

Remarks: H is the *binary entropy function* which takes values in $[0, \log 2]$. Note that for any choices of c and p, Jensen's inequality yields

$$H(c) - T_p(c) = c \log \frac{p}{c} + (1-c) \log \frac{1-p}{1-c} \leq \log 1 = 0$$

with equality holding only when $c = p$. Hence, $T_p(c) > H(c)$ whenever $c \neq p$. Chernoff's bound hence yields exponentially small probabilities for the extreme tails of the binomial distribution. This bound can be shown to be exponentially tight (see Blake and Darabian [9], for instance).

For the special case $p = 1/2$, Chernoff's bound yields

$$\sum_{k=0}^{\lfloor cN \rfloor} b(k;N,0.5) \geq 1 - e^{-N[\log 2 - H(c)]}$$

for any choice $1/2 < c \leq 1$.

For any m-set of points $\boldsymbol{U} \in \mathbb{R}^N$, $|\boldsymbol{U}| = m$, let $\mathcal{D}(\boldsymbol{U})$ denote the family of dichotomies of $\boldsymbol{U}$ that can be separated by a neuron: a dichotomy $(\boldsymbol{U}^+, \boldsymbol{U}^-)$ of $\boldsymbol{U}$ is in $\mathcal{D}(\boldsymbol{U})$ if and only if there exists a weight vector $\boldsymbol{w} = (w_1 \cdots w_N) \in \mathbb{R}^N$ such that

$$\sum_{i=1}^{N} w_i u_i \begin{cases} \geq 0 & \text{if } \boldsymbol{u} = (u_1 \cdots u_N) \in \boldsymbol{U}^+ \\ < 0 & \text{if } \boldsymbol{u} = (u_1 \cdots u_N) \in \boldsymbol{U}^- . \end{cases}$$

The following estimate for the number of dichotomies separable by a neuron was given by Schläffli [10] using an elegant combinatorial argument. (For more accessible proofs of the result see Wendel [11] or Cover [12].)

Lemma 3.2: Let $\boldsymbol{U} \in \mathbb{R}^N$ be an m-set of points. Then the following estimate holds for the number of dichotomies of $\boldsymbol{U}$ separable by a neuron:

$$|\mathcal{D}(\boldsymbol{U})| \leq D(N,m) = 2 \sum_{i=0}^{N-1} \binom{m-1}{i}.$$

[6] We define $H(c) \equiv 0$ when $c = 0$ or $c = 1$.

Furthermore, the upper bound of $D(N,m)$ is achieved for m-sets of points in general position.[7]

A fundamental parameter of interest to us is the probability that a neuron can separate a random dichotomy of a random set of vertices of the cube $\mathbb{B}^N$. Let $\boldsymbol{u}^1, \cdots, \boldsymbol{u}^m$ be a randomly drawn set of patterns from the vertices of the cube $\mathbb{B}^N$, and let the pattern components $\{u_i^\alpha : i \in [N], \alpha \in [m]\}$ form a sequence of symmetric Bernoulli trials:

$$\boldsymbol{P}\{u_i^\alpha = -1\} = \boldsymbol{P}\{u_i^\alpha = +1\} = 1/2, \qquad i \in [N],\ \alpha \in [m].$$

To each pattern $\boldsymbol{u}^\alpha$ associate a desired classification $v^\alpha \in \mathbb{B}$ specified independently of $\boldsymbol{u}^\alpha$. We are interested in the probability $P(N,m)$ that there exists a choice of weight vector $\boldsymbol{w} \in \mathbb{R}^N$ such that

$$\operatorname{sgn}\left(\sum_{i=1}^{N} w_i u_i^\alpha\right) = v^\alpha, \qquad \alpha = 1, \cdots m.$$

The following asymptotic estimate for $P(N,m)$ was shown by Füredi [13].

Lemma 3.3: If $m = O(N)$ as $N \to \infty$ then

$$P(N,m) = \sum_{j=0}^{N-1} b(j; m-1, 0.5) - O\left(e^{-c_1 N}\right).$$

Remarks: Note that Lemma 3.2 guarantees that $2^{-m} \cdot D(N,m)$ is an upper bound for $P(N,m)$. The asymptotic order correction to this estimate in Lemma 3.3 arises because the probability that a random m-set of *vertices* from $\mathbb{B}^N$ is in general position rapidly becomes small when m exceeds N. The exponentially small order term quoted above is a strengthening of Füredi's original estimate of $O(N^{-1/2})$. The refinement was made possible by a new result due to Kahn, Komlós, and Szemerédi [14] which asserts that the probability that a random $N \times N \pm 1$ matrix is singular decreases exponentially fast with N. Incorporating this result in Füredi's proof (without any other change) gives the quoted improvement. We will need the stronger form of the result.

A direct application of Chernoff's bound to the above estimate for $P(N,m)$ yields the following

Lemma 3.4: For every fixed $\lambda > 0$ we can find absolute positive constants c_2 and c_3 such that as $N \to \infty$:

$$P(N, 2N(1-\lambda)) = 1 - O\left(e^{-c_2 N}\right),$$
$$P(N, 2N(1+\lambda)) = O\left(e^{-c_3 N}\right).$$

Remarks: If the points $\boldsymbol{u}^1, \cdots, \boldsymbol{u}^m$ are points in $\mathbb{R}^N$ drawn independently from an absolutely continuous distribution, then a well known result asserts that a formal neuron can separate a random dichotomy of up to $2N$ patterns. Lemma 3.4 asserts that the classical result holds true even when the points are chosen from the (discrete) uniform distribution on the vertices $\mathbb{B}^N$ of the cube.

Using elementary binomial identities it is easy to verify the following "monotone property."

[7] An m-set of points in N-space is in *general position* iff any subset of N or fewer of the points is linearly independent.

Lemma 3.5: If $k = O(n)$ as $N \to \infty$ then

$$\left| P(N,k) - P(N,k+1) - 2^{-k}\binom{k-1}{N-1} \right| = O(e^{-c_4' N}).$$

In particular, $P(N,k)$ is a monotone nonincreasing function of k in the vicinity of $2N$ for large enough N. Note that Stirling's formula gives $|P(N,k) - P(N,k+1)| = O(e^{-c_4' N})$ when $k = (2-\delta)N$ or $k = (2+\delta)N$.

IV. Error Tolerance: Capacities

A. Random Error Protocol

We first consider the computational gains that are feasible under a random error protocol with parameter $\epsilon \in [0, 1/2)$. For any fixed $i \in [n]$, let $\{D_i^\alpha : \alpha \in [m]\}$ be the sequence of decision sets corresponding to neuron i. Recall that the decision sets are drawn independently according to a sequence of Bernoulli trials, and

$$Dia = \begin{cases} \{u_i^\alpha\} & \text{with probability } 1-2\epsilon \\ \{-1, 1\} & \text{with probability } 2\epsilon. \end{cases}$$

Theorem 4.1: For any fixed error parameter $0 \le \epsilon < 1/2$, the sequence $2n/(1-2\epsilon)$ is a capacity function for $\mathcal{R}_\epsilon(n,m)$, the random error protocol with parameter ϵ.

Proof: The ith neuron makes ϵ-reliable decisions if there is a choice of $n-1$ weights $\{w_{ij} : j \in [n] \setminus \{i\}\}$ such that

$$\operatorname{sgn}\left(\sum_{j \ne i} w_{ij} u_j^\alpha\right) \in D_i^\alpha, \qquad \alpha = 1, \cdots, m.$$

Alternatively, let $A = \{\alpha : D_i^\alpha \text{ is normal}\}$ be the (random) set of indexes identifying memories whose ith component must be retrieved. The above is then equivalent to requiring that

$$\operatorname{sgn}\left(\sum_{j \ne i} w_{ij} u_j^\alpha\right) = u_i^\alpha, \qquad \alpha \in A.$$

Note that as a consequence of the zero-diagonal nature of the network, the term u_i^α is absent in the sum above. By the independence of the memory components, if $|A| = k$ then the above is equivalent to finding a weight vector in $(n-1)$-space which separates a randomly and independently specified dichotomy of a set of k vertices chosen randomly from $\mathbf{B}^{n-1}$. It is hence clear that $P(n-1,k)$ is the probability that the ith neuron makes ϵ-reliable decisions *conditioned* upon there being k normal decision sets and $m-k$ exceptional decision sets. As the distribution of normal and exceptional decision sets is governed by the binomial distribution it follows that the probability $P_\epsilon(n,m)$ that the ith neuron makes ϵ-reliable decisions is given by

$$P_\epsilon(n,m) = \sum_{k=0}^{m} b(k; m, 1-2\epsilon)\, P(n-1,k). \tag{1}$$

By Boole's inequality, the probability $1 - \boldsymbol{P}\{\mathcal{R}_\epsilon(n,m)\}$ that one or more neurons fails to make ϵ-reliable decisions is bounded by

$$1 - \boldsymbol{P}\{\mathcal{R}_\epsilon(n,m)\} \le n[1 - P_\epsilon(n,m)].$$

Also, the probability $\boldsymbol{P}\{\mathcal{R}_\epsilon(n,m)\}$ that all the neurons make ϵ-reliable decisions is clearly bounded above by the probability $P_\epsilon(n,m)$ that a single neuron makes ϵ-reliable decisions. Combining this with the above inequality we have the two-sided bounds:

$$1 - n[1 - P_\epsilon(n,m)] \le \boldsymbol{P}\{\mathcal{R}_\epsilon(n,m)\} \le P_\epsilon(n,m). \tag{2}$$

Now let $0 < \lambda < \epsilon$ be fixed but arbitrary. With the choice

$$m = (1-\lambda)\frac{2n}{1-2\epsilon} \tag{3}$$

we have

$$\begin{aligned} P_\epsilon(n,m) &\ge \sum_{k=0}^{m(1-2\epsilon)(1+\lambda)} b(k; m, 1-2\epsilon)\, P(n-1,k) \\ &= 1 - O\left(e^{-c_5 n}\right) \qquad (n \to \infty). \end{aligned}$$

The first inequality is obvious as it arises from the deletion of terms in the sum in (1). This lower bound for $P_\epsilon(n,m)$ can be seen to approach 1 exponentially fast as asserted above by two appeals to Lemma 3.1: with m increasing with n as in (3), $P(n-1,k) = 1 - O\left(e^{-c_5' n}\right)$ for k in the range $0 \le k \le m(1-2\epsilon)(1+\lambda)$; further, $\sum_{k=0}^{m(1-2\epsilon)(1+\lambda)} b(k; m, 1-2\epsilon) = 1 - O\left(e^{-c_5'' n}\right)$. It follows from the lower bound of (2) that

$$\boldsymbol{P}\{\mathcal{R}_\epsilon(n,m)\} = 1 - O(n\, e^{-c_5 n}) \qquad (n \to \infty)$$

for m growing as in (3). As λ can be chosen arbitrarily small, $2n/(1-2\epsilon)$ is a lower capacity function for $\mathcal{R}_\epsilon(n,m)$.

Now choose m growing with n such that

$$m = (1+\lambda)\frac{2n}{1-2\epsilon}. \tag{4}$$

Partitioning the sum in (1) into two parts, we have

$$\begin{aligned} P_\epsilon(n,m) = &\sum_{k=0}^{m(1-2\epsilon)(1-\lambda)} b(k; m, 1-2\epsilon) P(n-1,k) \\ &+ \sum_{k=m(1-2\epsilon)(1-\lambda)+1}^{m} b(k; m, 1-2\epsilon) P(n-1,k). \end{aligned}$$

Obtain an upper bound by replacing $P(n-1,k)$ by 1 in the first sum, and, invoking Lemma 3.5, by replacing $P(n-1,k)$ by $P(n-1, m(1-2\epsilon)(1-\lambda)) + O(e^{-c_6' n})$ in the second

sum. As $n \to \infty$, we then have

$$\begin{aligned} P_\epsilon(n,m) &\le \sum_{k=0}^{m(1-2\epsilon)(1-\lambda)} b(k;m,1-2\epsilon) \\ &\quad + \left(P(n-1,m(1-2\epsilon)(1-\lambda)) + O\left(e^{-c_6' n}\right)\right) \\ &\quad \sum_{k=m(1-2\epsilon)(1-\lambda)+1}^{m} b(k;m,1-2\epsilon) \\ &\le \sum_{k=0}^{m(1-2\epsilon)(1-\lambda)} b(k;m,1-2\epsilon) \\ &\quad + P(n-1,m(1-2\epsilon)(1-\lambda)) + O\left(e^{-c_6' n}\right) \\ &= O\left(e^{-c_6 n}\right). \end{aligned}$$

The exponential decrease of the upper bound to zero is readily ascertained by applying Lemma 3.1 twice, as before. The upper bound of (2) hence yields

$$\boldsymbol{P}\{\mathcal{R}_\epsilon(n,m)\} = O\left(e^{-c_6 n}\right) \qquad (n \to \infty)$$

for m growing as in (4). As λ is arbitrary, $2n/(1-2\epsilon)$ is also an upper capacity function, hence a capacity function, for $\mathcal{R}_\epsilon(n,m)$. ∎

The case where each memory is required to be a fixed point of the network corresponds to the choice of error parameter $\epsilon = 0$. The following conclusion is hence immediate:

Corollary 4.2: The sequence $2n$ is a capacity function for the attribute $\mathcal{R}_0(n,m)$ that all the memories are fixed points of the network.

This fixed point capacity of $2n$ was also demonstrated by Venkatesh and Baldi [15] in the analysis of fixed points of higher order neural networks. Recall the classical result restated in Lemma 3.4 that $2n$ is a capacity function for a *single* neuron. (The relevant attribute here is the separation of a random dichotomy of a set of points (memories) in n-space by a neuron.) The corollary above asserts that there is no decrease in capacity for a zero-diagonal network of n neurons even though we now require n dichotomies of the set of memories to be *simultaneously* separated. (As seen in the proof of the theorem, neuron i, for instance, is required to dichotomize the set of memories according to the set of signs $\{u_i^\alpha : \alpha \in [m]\}$.)

Theorem 4.1 hence asserts that if the fixed point requirement on the memories is relaxed and it is only required that, starting at any memory, a synchronous state transition results in a new state no more than ϵn bits away from the memory *on average*, then the capacity increases by a constant multiplicative factor of $1/(1-2\epsilon)$. Note, however, that the rate of increase of the capacity function remains linear in n and is not improved in the random error protocol.

Theorem 4.1 remains true if we are interested in *hetero-associative* maps $\boldsymbol{u}^\alpha \mapsto \hat{\boldsymbol{u}}^\alpha$ rather than the *auto-associative* maps $\boldsymbol{u}^\alpha \mapsto \boldsymbol{u}^\alpha$ that we have hitherto considered. In particular, for $\alpha = 1, \cdots, m$ let the *associated memories* $\hat{\boldsymbol{u}}^\alpha$ be chosen independently from $\mathbf{B}^n$ and with components drawn from a sequence of symmetric Bernoulli trials (independent of the memories $\boldsymbol{u}^\alpha$). The decision sets D_i^α are now specified in natural fashion by a sequence of Bernoulli trials with

$$D_i^\alpha = \begin{cases} \{\hat{u}_i^\alpha\} & \text{with probability } 1-2\epsilon \\ \{-1,1\} & \text{with probability } 2\epsilon. \end{cases}$$

It is easy to see that the proof of the theorem carries over *in toto* for the hetero-associative case. A more direct proof can be crafted, however, with the observation that the independence of the components of the associated memories $\hat{\boldsymbol{u}}^\alpha$ yields

$$\boldsymbol{P}\{\mathcal{R}_\epsilon(n,m)\} = P_\epsilon(n,m)^n.$$

Lemma 3.1 now readily yields that $2n/(1-2\epsilon)$ is both a lower and an upper capacity function for $\mathcal{R}_\epsilon(n,m)$.[8]

B. Exhaustive Error Protocol

For ϵ close to 1/2 the multiplicative improvement (over the error-free case) of $1/(1-2\epsilon)$ to the capacity that arises for the random error protocol can become quite large; the gains may nonetheless be perceived as unsatisfactory as there is no improvement in the *rate* of increase of capacity with n. The exhaustive error protocol would seem to have a greater potential for substantially increased capacity as it would appear to confer even greater flexibility in the choice of errors than the random error protocol—one is allowed in principle to examine every admissible configuration of errors before selecting the most favorable configuration—; as argued heuristically in the Introduction, this might augur well for a large improvement in capacity. We show in this section, however, that while there is a further improvement in the multiplicative constant, the capacity function for the exhaustive error protocol is still linear in n.

As a first step let us show that, in accordance with intuition, the exhaustive error protocol attains capacities at least as large as those of the random error protocol.

Theorem 4.3: For any fixed error parameter $0 \le \epsilon < 1/2$, the sequence $2n/(1-2\epsilon)$ is a lower capacity function for $\mathcal{E}_\epsilon(n,m)$, the exhaustive error protocol with parameter ϵ.

Proof: If $\epsilon = 0$ there is nothing to prove. Let us hence assume $\epsilon > 0$. Now for any choice $0 \le \epsilon' < \epsilon$, the sequence $2n/(1-2\epsilon')$ is a capacity function for the random error protocol with parameter ϵ'. Invoking Lemma 3.1, within capacity the number of errors in each memory that result in the random error protocol is no more than $(\epsilon' + o(1))n$ with probability approaching one as $n \to \infty$. For any $\epsilon' < \epsilon$ the number of errors in each memory is hence less than ϵn with probability one. Consequently, $2n/(1-2\epsilon')$ is a lower capacity function for the exhaustive error protocol with parameter ϵ. As $\epsilon' < \epsilon$ can be chosen arbitrarily close to ϵ, it follows from the definition of capacity that $2n/(1-2\epsilon)$ is a lower capacity for the exhaustive error protocol with parameter ϵ. ∎

[8] We had presented these results without proof for the hetero-associative case in [1]. There we had assumed in addition that the memories $\boldsymbol{u}^\alpha$ were drawn from an absolutely continuous distribution in Euclidean n-space $\mathbb{R}^n$, and not from the vertices of the cube $\mathbf{B}^n$ as is more natural in the recurrent network context. The increased dependency structure in the problem makes the case of auto-association with memories drawn from $\mathbf{B}^n$ somewhat harder technically; the improvement to Füredi's lemma quoted in the text is necessary here.

Theorem 4.4: Let κ_ϵ be a function of the error tolerance ϵ defined by the unique solution of

$$H\left(\frac{1-2\epsilon}{2\kappa_\epsilon}\right) + H(\epsilon) = \log 2, \qquad 0 \le \epsilon < \frac{1}{2} \tag{5}$$

where H is the binary entropy function. Then the sequence $2\kappa_\epsilon n/(1-2\epsilon)$ is an upper capacity function for $\mathcal{E}_\epsilon(n,m)$, the exhaustive error protocol with parameter ϵ.

Remark: The function κ_ϵ is defined as we vary ϵ in the interval $0 \le \epsilon < 1/2$, and monotonically increases from a value of $+1$ at $\epsilon = 0$ to a value close to 505 as ϵ approaches 1/2. For small ϵ it remains close to +1, so that the capacity function for the attribute $\mathcal{E}_\epsilon(n,m)$ still behaves like $2n/(1-2\epsilon)$.

Proof: Assume that a particular choice of weights for the zero-diagonal network of neurons results in the synchronous transitions: $\boldsymbol{u}^\alpha \mapsto \boldsymbol{v}^\alpha, \alpha = 1, \cdots, m$. Recall that the ith neuron makes a decision error on memory $\boldsymbol{u}^\alpha$ if $v_i^\alpha \neq u_i^\alpha$, i.e., the ith component of memory $\boldsymbol{u}^\alpha$ is not retrieved. The key observation here is the following: if each $\boldsymbol{v}^\alpha$ differs from the corresponding memory $\boldsymbol{u}^\alpha$ in no more than ϵn components, then there exists at least one neuron which makes ϵm or fewer decision errors. In fact, if there is no neuron which makes ϵm or fewer decision errors, then the total number of component errors after one synchronous step summed over all the m memories will exceed ϵmn so that there has to be at least one memory $\boldsymbol{u}^\alpha$ which is mapped to a point $\boldsymbol{v}^\alpha$ which is at a Hamming distance larger than ϵn from $\boldsymbol{u}^\alpha$. But this contradicts the earlier assumption about the points $\boldsymbol{v}^\alpha$.

Now, for any selection of values $v_i^\alpha \in \mathbb{B}, \alpha = 1, \cdots, m$, the probability that the ith neuron can realize the maps $\boldsymbol{u}^\alpha \mapsto v_i^\alpha$ is exactly $P(n-1,m)$. The number of ways in which ϵm or fewer decision errors (i.e., the set $\{\alpha : v_i^\alpha \neq u_i^\alpha\}$) can occur is $\sum_{k=0}^{\epsilon m} \binom{m}{k}$. Combining Boole's inequality with the observation above, we hence have the upper bound

$$\boldsymbol{P}\{\mathcal{E}_\epsilon(n,m)\} \le n\, P(n-1,m) \sum_{k=0}^{\epsilon m} \binom{m}{k}.$$

Let $\lambda > 0$ be fixed, but arbitrary, and choose $m = 2\kappa_\epsilon n(1+\lambda)/(1-2\epsilon)$, where κ_ϵ is as defined in (5). Two applications of Lemma 3.1 yield

$$P(n-1,m) \le e^{-(m-1)[\log 2 - H\{(1-2\epsilon)/2\kappa_\epsilon(1+\lambda)\}]},$$

and

$$\sum_{k=0}^{\epsilon m} \binom{m}{k} \le e^{mH(\epsilon)}$$

for large enough n. Hence, for each choice of $0 \le \epsilon < 1/2$ and $\lambda > 0$, there is a choice of $\beta > 0$ such that

$$\boldsymbol{P}\{\mathcal{E}_\epsilon(n,m)\} \le \beta n\, e^{-m[\log 2 - H\{(1-2\epsilon)/2\kappa_\epsilon(1+\lambda)\} - H(\epsilon)]}.$$

The binary entropy function $H(c)$ increases monotonically from a value of 0 at $c = 0$ to a value of $\log 2$ at $c = 1/2$. Hence, with κ_ϵ as in (5), $H\{(1-2\epsilon)/2\kappa_\epsilon(1+\lambda)\} + H(\epsilon) < \log 2$. Let

$$\delta = \log 2 - H\{(1-2\epsilon)/2\kappa_\epsilon(1+\lambda)\} - H(\epsilon) > 0.$$

For every choice of $\lambda > 0$ and $m = 2\kappa_\epsilon n(1+\lambda)/(1-2\epsilon)$, we then have $\boldsymbol{P}\{\mathcal{E}_\epsilon(n,m)\} = O(ne^{-\delta m}) = o(1)$ as $n \to \infty$. ∎

The exhaustive error protocol allows any synchronous transition from a memory that does not leave the Hamming ball of radius ϵn at the memory. It is clear that this is a necessary condition that must be satisfied if we require confinement of trajectories within balls of radius ϵn at the memories. (Recall that in allowing an error tolerance of up to ϵn bits in the recall of any memory we required that state transitions be confined to the ball of radius ϵn at a memory once a transition leads within the ball.) Consequently, Theorem 4.4 implies the following result: *if memory components are drawn from a sequence of symmetric Bernoulli trials, then no algorithm for storing memories in a recurrent neural network can achieve a capacity which increases faster than linearly with n; in particular, if a linear number of errors ϵn is permitted in the recall of any memory, then $1010n/(1-2\epsilon)$ is an upper capacity function for any algorithm.*

V. Extensions

The error protocols that we had defined in Section II for fully-interconnected networks can be readily extended to arbitrary network architectures. We briefly derive here certain bounds on the capacity of feedforward neural networks when there is an allowed tolerance to output error.

An *L-layer feedforward neural network* is comprised of L ordered subcollections of neurons called *layers* with interconnections specified as follows: for $l = 2, \cdots, L$ the inputs to the lth layer are obtained from the outputs of the $(l-1)$th layer. The inputs to the first layer are the network inputs, and the outputs of the Lth layer are the network outputs. Let $\boldsymbol{n} = (n_0\, n_1 \cdots n_{l+1})$ denote a vector of positive integers. We use the nonce terminology *$\boldsymbol{n}$-network* to mean an $(l+1)$-layer feedforward neural network which has $n_0 \equiv n$ inputs and whose ith layer contains n_i neurons. For simplicity we will restrict ourselves to the case of a single output neuron, $n_{l+1} = 1$. An $\boldsymbol{n}$-network then realizes a Boolean function of $n_0 = n$ inputs. The error protocols are readily extended to this case.

Let $\boldsymbol{u}^1, \cdots, \boldsymbol{u}^m$ be any set of points from Euclidean n-space. To each point $\boldsymbol{u}^\alpha$ we assign a desired classification $v^\alpha \in \mathbb{B}$. We assume that the set of desired classifications $\{v^1, \cdots, v^m\}$ are drawn from a sequence of symmetric Bernoulli trials.

Analogously with the fully-interconnected case, in the random error protocol we independently assign decision sets D^α to each classification v^α with

$$D^\alpha = \begin{cases} \{v^\alpha\} & \text{with probability } 1-2\epsilon \\ \mathbb{B} & \text{with probability } 2\epsilon. \end{cases}$$

For a particular assignment of weights to the $\boldsymbol{n}$-network we say that the network makes ϵ-reliable decisions on the set of points $\{\boldsymbol{u}^\alpha : \alpha \in [m]\}$ if $\boldsymbol{u}^\alpha \mapsto D^\alpha$ for each $\alpha \in [m]$. The attribute of interest is

EVENT $\mathcal{R}_\epsilon(\boldsymbol{n}, m)$: *There is a choice of weights such that the $\boldsymbol{n}$-network makes ϵ-reliable decisions on the m-set of points $\{\boldsymbol{u}^\alpha : \alpha \in [m]\}$.*

A completely analogous development leads to the following attribute for the exhaustive error protocol.

EVENT $\mathcal{E}_\epsilon(\boldsymbol{n}, m)$: *There is a choice of weights such that the* $\boldsymbol{n}$*-network makes no more than* ϵm *classification errors on the* m*-set of points* $\{\boldsymbol{u}^\alpha : \alpha \in [m]\}$.

The notion of capacity is defined as before as a threshold function of an attribute of the m-set of points $\{\boldsymbol{u}^1, \cdots, p\boldsymbol{u}^m\}$ as the input dimension n becomes large. (Note that we tacitly assume a family of feedforward network architectures with the number of elements in each layer n_i a function of n.)

Now let $D(\boldsymbol{n}, m)$ denote the number of dichotomies of $\{\boldsymbol{u}^\alpha : \alpha \in [m]\}$ that can be separated by an $\boldsymbol{n}$-network. The following simple overestimate for $D(\boldsymbol{n}, m)$ is obtained by applying Lemma 3.2:

$$D(\boldsymbol{n}, m) \leq \prod_{i=0}^{l} D(n_i, m)^{n_{i+1}}.$$

As the classifications are symmetric Bernoulli it follows that the probability $P(\boldsymbol{n}, m)$ that a random dichotomy can be separated by the network can be bounded by

$$P(\boldsymbol{n}, m) = 2^{-m} D(\boldsymbol{n}, m) \leq 2^{-m} \prod_{i=0}^{l} D(n_i, m)^{n_{i+1}}$$
$$= \exp\left[-m \log 2 + \sum_{i=0}^{l} n_{i+1} \log D(n_i, m)\right].$$

Using the easy bound $D(n_i, m) < m^{n_i}$ we get

$$P(\boldsymbol{n}, m) < \exp\left[-m \log 2 + \left(\sum_{i=0}^{l} n_i n_{i+1}\right) \log m\right].$$

Define the function $C_0^{(1)}(\boldsymbol{n})$ by

$$C_0^{(1)}(\boldsymbol{n}) \log 2 = \left(\sum_{i=0}^{l} n_i n_{i+1}\right) \log C_0^{(1)}(\boldsymbol{n}).$$

Note that

$$C_0^{(1)}(\boldsymbol{n}) = \frac{1}{\log 2}\left(\sum_{i=0}^{l} n_i n_{i+1}\right) \cdot \log\left(\sum_{i=0}^{l} n_i n_{i+1}\right) \cdot \left[1 + O\left(\frac{\log\log \sum_{i=0}^{l} n_i n_{i+1}}{\log \sum_{i=0}^{l} n_i n_{i+1}}\right)\right] \qquad (n \to \infty).$$

It is clear that for any $\lambda > 0$, if $m \geq (1 + \lambda) C_0^{(1)}(\boldsymbol{n})$ then $P(\boldsymbol{n}, m) \to 0$ as $n \to \infty$. As before, we have

$$\boldsymbol{P}\{\mathcal{R}_\epsilon(\boldsymbol{n}, m)\} = \sum_{k=0}^{m} b(k; m, 1 - 2\epsilon)\, P(\boldsymbol{n}, k).$$

The same argument in the proof of Theorem 4.3 continues to work, so that we have proved the following

Theorem 5.1: The sequence $C_\epsilon^{(1)}(\boldsymbol{n}) = \frac{C_0^{(1)}(\boldsymbol{n})}{1-2\epsilon}$ is an upper capacity function for the attribute $\mathcal{R}_\epsilon(\boldsymbol{n}, m)$.

For the exhaustive error protocol, we have likewise

$$\boldsymbol{P}\{\mathcal{E}_\epsilon(\boldsymbol{n}, m)\} \leq P(\boldsymbol{n}, m) \sum_{k=0}^{\epsilon m} \binom{m}{k} < \exp\left[-m \log 2 + \left(\sum_{i=0}^{l} n_i n_{i+1}\right) \log m + m H(\epsilon)\right].$$

For a small enough choice of error parameter ϵ, let $C_\epsilon^{(2)}(\boldsymbol{n})$ satisfy

$$C_\epsilon^{(2)}(\boldsymbol{n})[\log 2 - H(\epsilon)] = \left(\sum_{i=0}^{l} n_i n_{i+1}\right) \log C_\epsilon^{(2)}(\boldsymbol{n}).$$

Again, as $n \to \infty$, we have

$$C_\epsilon^{(2)}(\boldsymbol{n}) = \frac{1}{\log 2 - H(\epsilon)}\left(\sum_{i=0}^{l} n_i n_{i+1}\right) \log\left(\sum_{i=0}^{l} n_i n_{i+1}\right) \cdot \left[1 + O\left(\frac{\log\log \sum_{i=0}^{l} n_i n_{i+1}}{\log \sum_{i=0}^{l} n_i n_{i+1}}\right)\right].$$

For any fixed $\lambda > 0$ if $m \geq (1 + \lambda) C_\epsilon^{(2)}(\boldsymbol{n})$ then $\boldsymbol{P}\{\mathcal{E}_\epsilon(\boldsymbol{n}, m)\} \to 0$ as $n \to \infty$. Hence, we have the following

Theorem 5.2: The sequence $C_\epsilon^{(2)}(\boldsymbol{n})$ is an upper capacity function for the attribute $\mathcal{E}_\epsilon(\boldsymbol{n}, m)$.

The results can be sharpened, for instance, by using the tighter bound $D(n_i, m) < 2m^{n_i - 1}/(n_i - 1)!$ which is valid if $m > 3n_i$ and $n_i > 3$ instead of the simpler estimate $D(n_i, m) < m^{n_i}$ used here. Unfortunately, good lower bounds are currently unavailable except for the case of one and two layer networks (cf., Venkatesh and Psaltis [6]; Baum [16]).

VI. Conclusions

The main contributions of this paper are the development of formal protocols for error tolerance, and the explicit computation of the gains that accrue for neural network capacity under these protocols when errors are permitted in the output. The principal result here is that error tolerance in a network situation results in gains in the multiplicative constants for the capacity, but leaves the rate of growth of capacity as input dimensionality increases unchanged. In particular, if a tolerance of $\epsilon \in [0, 1/2)$ is specified for a fully-interconnected network, then under the random error protocol there is a gain in capacity by a multiplicative factor of $1/(1 - 2\epsilon)$ over the error free case, while for the exhaustive error protocol the gain is no more than a multiplicative factor of $505/(1-2\epsilon)$. Similar gains accrue for feedforward network configurations.

The absence of more startling gains in capacity can be traced to the exponential decay of the relevant probabilities. These protocols are readily applicable in other computational paradigms. Following the analysis here, in a general computational scenario we would expect error tolerance to buy us order of magnitude improvements in computational capability only if the relevant probabilities decay sufficiently slowly.

As a final note, it is worth remarking upon the strong information theoretic flavor of the techniques in proof, in particular,

random coding and sphere hardening (cf., McEliece [17], for instance). While the notion of (computational) capacity analyzed here is somewhat different from the notion of channel capacity, the results are also very similar in spirit to analogous situations in information theory. The gentle reader is invited to verify, for instance, that the capacities under the random error protocol remain unaffected if $2\epsilon n$ don't-care components per memory are deterministically fixed in advance (or indeed, chosen from a variety of possible distributions—including the binomial, as in the text—for which the mean of the number of don't-care components per memory is $2\epsilon n$). This mirrors the following known result in information theory: the capacity of the binary erasure channel (with erasure locations not known in advance) is the same as the capacity if the erasure locations are known in advance.

ACKNOWLEDGMENT

The author thanks the referees for suggesting that the link with information theory be made explicit, and for catching the odd typo in a first draft of the manuscript.

REFERENCES

[1] S. S. Venkatesh, "Epsilon capacity of neural networks," in *Neural Networks for Computing*, J. Denker, Ed. New York: AIP, 1986.
[2] 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.
[3] J. Komlós and R. Paturi, "Convergence results in an associative memory model," *Neural Networks*, vol. 1, no. 3, pp. 239–250, 1988.
[4] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational properties," *Proc. Nat. Acad. Sci.*, vol. 79, pp. 2554–2558, 1982.
[5] C. M. Newman, "Memory capacity in neural network models: Rigorous lower bounds," *Neural Networks*, vol. 1, no. 3, pp. 223–238, 1988.
[6] S. S. Venkatesh and D. Psaltis, "On reliable computation with formal neurons," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, pp. 87–91, 1992.
[7] S. S. Venkatesh, "Computation and learning in the context of neural network capacity," in *Neural Networks for Perception*, H. Wechsler, Ed. New York: Academic, 1991.
[8] H. Chernoff, "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Ann. Math. Stat.*, vol. 23, pp. 493–507, 1952.
[9] I. F. Blake and H. Darabian, "Approximations for the probability in the tails of the binomial distribution," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 426–428, 1987.
[10] L. Schläfli, *Gesammelte Mathematische Abhandlungen I*. Basel, Switzerland: Verlag Birkhäuser, 1950, pp. 209–212.
[11] J. G. Wendel, "A problem in geometric probability," *Math. Scand.*, vol. 11, pp. 109–111, 1962.
[12] T. M. Cover, "Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Trans. Electron. Comput.*, vol. EC-14, pp. 326–334, 1965.
[13] Z. Füredi, "Random polytopes in the d-dimensional cube," *Discrete Comput. Geom.*, vol. 1, pp. 315–319, 1986.
[14] J. Kahn, J. Komlós, and E. Szemerédi, "On the determinant of random ± 1 matrices," preprint.
[15] S. S. Venkatesh and P. Baldi, "Programmed interactions in higher-order neural networks: Maximal capacity," *J. Compl.*, vol. 7, pp. 316–337, 1991.
[16] E. B. Baum, "On the capabilities of multi-layer perceptrons," *J. Compl.*, vol. 4, pp. 193–215, 1988.
[17] R. J. McEliece, *The Theory of Information and Coding*. Reading, MA: Addison-Wesley, 1977.

Santosh S. Venkatesh (S'81–M'86) was born on June 8, 1959 in Trichur, India. He received the B.Tech. degree with distinction from the Indian Institute of Technology, Bombay, in 1981, and the M.S. and Ph.D. degrees from the California Institute of Technology, Pasadena, in 1982 and 1986, respectively, all in electrical engineering.

Since 1986 he has been an Assistant Professor of Electrical Engineering and the Neurosciences Graduate Group at the University of Pennsylvania, Philadelphia, where he is also a member of the David Mahoney Institute for Neurological Sciences. He has been a Consultant for Dupont de Nemours, Inc. and AT&T Bell Laboratories. His research interest include pattern recognition, neural networks, complexity theory, and computational learning theory.