NWC TP 7106

Neuron Requirements for Classification

by
W. O. Alltop
Research Department

JANUARY 1991

NAVAL WEAPONS CENTER
CHINA LAKE, CA 93555-6001

Naval Weapons Center

FOREWORD

This report presents geometric results applicable to the design and implementation of layered neural networks. The work was performed during 1989 and 1990 as part of the Naval Weapons Center Independent Research Program.

This report was reviewed for technical accuracy by J. M. Martin.

Approved by
R. L. DERR, *Head*
Research Department
26 November 90

Under the authority of
D. W. COOK
Capt., U. S. Navy
Commander

Released for publication by
W. B. PORTER
Technical Director

NWC Technical Publication 7106

Published by ..... Technical Information Department
Collation ..... Cover, 17 leaves
First printing ..... 66 copies

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	January 1991	Final, May 1989 to July 1990

4. TITLE AND SUBTITLE
NEURON REQUIREMENTS FOR CLASSIFICATION

5. FUNDING NUMBERS
PE 61152N
PR RR00NW
TA RR00NW
WU 13807002

6. AUTHOR(S)
William O. Alltop

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Weapons Center
China Lake, CA 93555-6001

8. PERFORMING ORGANIZATION REPORT NUMBER
NWC TP 7106

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Arlington, VA 22217

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
A Statement; public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

(U) The feed forward layered neural network holds great promise for application to classification problems. Determination of the sizes of the layers is an important network design problem. This report treats the neuron requirement question from the geometric viewpoint. Threshold neurons correspond to cutting planes in the Euclidean space of input patterns. Bounds on the minimum number of first-layer neurons are determined as functions of the partition sizes of the training data sets. Bounds are also proved for convex pattern classes. Measures of separability of the training data are defined in order tc emphasize the dependence of the design parameters upon the geometry of the classes.

14. SUBJECT TERMS
Neural Networks, Pattern Recognition, Geometry

15. NUMBER OF PAGES
32

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

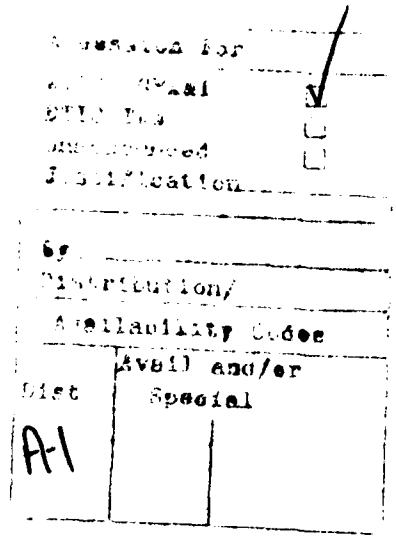


TABLE OF CONTENTS

ACKNOWLEDGEMENTS

The author is grateful to J. M. Martin for many helpful suggestions regarding this research effort and to Teri Hines for preparing the manuscript.

INTRODUCTION

Classification and taxonomy problems are pervasive in the design of weapons systems. The increasing capabilities of sensors and signal processing computers require ever more sophisticated algorithms. Two-dimensional signals are becoming available in many missile systems. Infrared and inverse synthetic aperture radar provide two-dimensional signals for guidance and target selection systems. One-dimensional RF signals arise from range only radar profiles (active RF) and emitter recognition (passive RF). Regardless of the signal domain involved, classification of the unknown object, or objects, generating or reflecting the signal is often the primary task of the signal processing algorithms.

The classification procedure assumed here consists of the design and implementation of a transformation, equivalently a function or mapping, from the set of input patterns to a set of desired outputs. The resulting transformation is called a classifier. This conforms to the types of models usually assumed in the pattern recognition literature. The following mathematical definitions give simplified versions of two basic pattern classification models (References 1 and 2).

Model 1. The discrete model.

U is the universe of objects (patterns).

$\mathcal{U}$ is a partition $\{U_1, ..., U_K\}$ of U into disjoint subsets.

$\mathcal{X}$ is a collection $\{X_1, ..., X_K\}$ of finite sets satisfying $X_i \subseteq U_i$ for $1 \le i \le K$.

$\mathcal{X}$ is known and $\mathcal{U}$ is unknown.

Classification consists of the design and implementation of algorithms for determining to which U_i unlabeled objects u belong.

Model 2. The statistical model.

U is the universe of objects (patterns).

$\mathcal{F}$ is a collection $\{f_1, ..., f_K\}$ of probability density functions (pdfs) on U.

For $1 \le i \le K$, X_i is a finite sample from the pdf f_i, and $\mathcal{X}$ is the collection of X_i's.

$\mathcal{X}$ is known and $\mathcal{F}$ is unknown.

Classification consists of the design and implementation of algorithms for determining from which f_i objects u were sampled.

The geometric results presented in this report are motivated primarily by Model 1. Model 2 is the more appealing in many classification applications. However, there is an enormous gap between the known data set $\mathcal{X}$ and the unknown pdfs. One must make assumptions regarding the types of distributions

which are possible for the f_i's. A good understanding of the underlying physical process usually helps determine the pdfs. In the presence of such understanding, the data set serves to validate the hypotheses regarding the pdfs. Model 1 assumes that the data set X is the only information available.

The set of input patterns is usually a real vector space, while the final outputs consist of a discrete set of class labels. Typically, the classifier employs intermediate outputs which also lie in a real vector space. The major analysis task is to determine a single mapping that sends the input patterns from distinct classes into well-separated, easily recognizable regions of the intermediate output space. From the intermediate output space, the mapping to the class labels should then be trivial. In the past most classifiers were implemented on von Neumann computers. The advances of parallel processing technology suggest new approaches to classifier design.

Of the many types of neurocomputing devices currently discussed in the engineering literature, perhaps the simplest is the feed forward layered neural network (LNN). This network is an obvious candidate for application to classification problems where the input patterns reside in a real vector space of fixed dimension.

The LNN takes as input the d coordinates of the pattern x and produces an m-dimensional output vector u. The output vectors are selected so as to facilitate the decision regarding the class to which x belongs.

For a K-class problem, one typically takes the output dimension m to be K and the desired output vectors to be the K elementary unit vectors

$$e_i = (0, 0, ..., 0, 1, 0, ... 0)^T$$

with 1 in the i^{th} coordinate. The network is designed and the weights are determined in order to map every member of the i^{th} class into a small neighborhood of e_i. These two steps—network design and weight assignment—give rise to interesting problems in the geometry of finite dimensional Euclidean spaces.

Step 1. Design the network; that is, determine the number of layers and the number of neurons in each layer.

Step 2. Determine the weights (one for each connection in the network) so as to produce the desired network mapping.

The two steps are clearly related. If the network does not accommodate the complexity of the classification problem, then Step 2 will be impossible. Here we give no precise definition of complexity. Roughly speaking, the problem complexity grows with the number of classes, the numbers of clusters within the classes, and the number of surfaces required to separate all pairs of interclass clusters (References 1 and 3). Appropriate weights may not exist if the number of connections in the network is too small (Reference 4).

We will discuss in some detail the number of first-layer neurons required by a threshold network to separate K convex classes in d-dimensional space. It will be shown in Sections 3 and 4 that this number can range from lg(K) to at most $(K^2 - K)/2$, when $K \leq d + 2$. For $K > d + 2$, the upper bound is at least

$$(d + 1)K - (d + 1)(d + 2)/2.$$

For K = 10 and d = 8, this gives a range of 4 to 45 first-layer neurons. For nonconvex classes, there is no upper bound.

The number of classes is usually fixed. Although the dimension of the raw input patterns is also fixed, the number d, of inputs to the LNN, may depend upon preprocessing and/or feature selection. These procedures generally reduce the input dimension, whereas addition of monomial features increases the input dimension (References 1 and 5).

The space of mappings $R^{(d)} \rightarrow R^{(m)}$ associated with a fixed network architecture is parameterized by the neuron transfer function s, called the squashing function, and the weights on the connections. Results presented here will pertain mostly to threshold transfer functions.

Section 2 presents some basic notation and terminology. Properties of decompositions by hyperplanes and their relevance to networks of threshold neurons are discussed in Section 3. Separation properties of disjoint convex sets are presented in Section 4.

NOTATION

$R^{(d)}$ is the space of real d-dimensional column vectors

$$y = (y_1, y_2, ..., y_d)^T.$$

For every $y \in R^{(d)}$, we define the extended column vector y^+ by

$$y^+ = (y_1, y_2, ..., y_d, 1)^T.$$

A λ by d + 1 matrix A defines an affine mapping $y \rightarrow A(y)$ from $R^{(d)}$ to $R^{(\lambda)}$ as follows:

$$A(y) = Ay^+.$$

Formally, we define the architecture of a LNN to be an ordered triple (s, h, Λ), where s is the neuron transfer function, h is a positive integer, and Λ is an (h + 1)-tuple

$$(\lambda_0, \lambda_1, ..., \lambda_h)$$

of positive integers. The function s satisfies

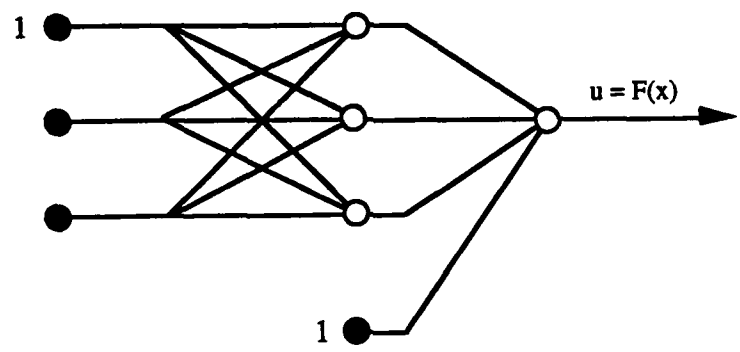
$$-1 \leq s(t) \leq 1 \text{ for all real } t,$$

$$t < u \Rightarrow s(t) \leq s(u) \text{ for all real } t, u,$$

$$\lim[s(t) : t \rightarrow -\infty] = -1,$$

$$\lim[s(t) : t \rightarrow -\infty] = 1.$$

By a Λ-network we mean a LNN, perhaps with unspecified neuron transfer function, having layers described by Λ. Figure 1 shows a (2, 3, 1) network. The extra nodes at layers 0 and 1 have constant value 1 in order to allow affine mappings (with nonzero bias) rather than pure linear mappings.



○ = node (neuron) with transfer function s

● = node with identity transfer function

FIGURE 1. (2, 3, 1) LNN.

The network mapping F is the composition of h single-layer mappings, each of which is the composition of an affine mapping and a 'squashing' function S.

$$F = f_h \circ f_{h-1} \circ \ldots \circ f_2 \circ f_1,$$

where

$$f_h = S \circ A_h,$$

$$S = (s, s, \ldots, s),$$

$$A_h = \begin{bmatrix} a_h(1, 1) & a_h(1, 2) & \cdots & a_h(1, \lambda_{h-1} + 1) \\ a_h(2, 1) & a_h(2, 2) & \cdots & a_h(2, \lambda_{h-1} + 1) \\ \vdots & \vdots & & \vdots \\ a_h(\lambda_h, 1) & a_h(\lambda_h, 2) & \cdots & a_h(\lambda_h, \lambda_{h-1} + 1) \end{bmatrix},$$

and the j^{th} coordinate of $A_h(x)$ is

$$a_h(j, \lambda_{h-1} + 1) + \sum_{i=1}^{\lambda_{h-1}} a_h(j, i)x_i$$

for $1 \le j \le \lambda_h$.

S is a vector of functions each of whose coordinates is s. For $y = (y_1, y_2, \ldots, y_\lambda) \in R^{(\lambda)}$,

$$S(y) = (s(y_1), s(y_2), \ldots, s(y_\lambda)).$$

Technically, we have a sequence of functions $S_1, S_2, \ldots$, where

$$S_\lambda : R^{(\lambda)} \rightarrow I^{(\lambda)}$$

and I denotes the unit interval [0, 1]. In order to simplify the notation, we use the unsubscripted S for all S_λ when the dimension of the domain is understood. For the (2, 3, 1) network of Figure 1, the network mapping F is given by

$$u = F(x) = S(A_2(S(A_1(x)))),$$

where

$$A_1 = \begin{bmatrix} a_1(1, 1) & a_1(1, 2) & a_1(1, 3) \\ a_1(2, 1) & a_1(2, 2) & a_1(2, 3) \\ a_1(3, 1) & a_1(3, 2) & a_1(3, 3) \end{bmatrix}$$

and

$$A_2 = \begin{bmatrix} a_2(1, 1) & a_2(1,2) & a_2(1, 3) & a_2(1, 4) \end{bmatrix}.$$

For $X \subseteq R^{(d)}$, Hull(X) denotes the convex hull of X. An extreme point of a convex set C is a point c satisfying

$$E \subseteq C \text{ and } c \in \text{Hull}(E) \Rightarrow c \in E.$$

That is, c is not a convex combination of other members of C.

A convex polytope is the convex hull of a finite set of points. For X finite and P = Hull(X), the set of vertices V of P is the set of extreme points of P. A convex N-gon is a convex polytope in $R^{(2)}$. A finite set X in $R^{(d)}$ is the set of vertices of a convex polytope if—and only if—

$$X \cap \text{Hull}(Y) = Y$$

for all $Y \subseteq X$.

$\lceil t \rceil$ denotes the smallest integer not smaller than t

and

$\lfloor t \rfloor$ denotes the largest integer not larger than t for all $t \in R$. For $y \in R^{(\lambda)}$, $\|y\|$ is the Euclidean norm:

$$\|y\| = (y_1^2 + y_2^2 + \ldots + y_\lambda^2)^{1/2}.$$

HYPERPLANES AND THRESHOLD NEURONS

Our objective in this section is to describe, using results from combinatorial geometry, how neuron requirements depend upon the configuration of input patterns. Reference 6 establishes a lower bound on the number of first-layer threshold neurons required, using the formula of Theorem 1 below. We extend this result by showing that the first-layer neuron requirement is problem dependent. That is, the lower bound on the number of first-layer threshold neurons, as a function of the set of data points, becomes much larger than that of Reference 6.

Combinatorial analysis of arrangements of hyperplanes in $R^{(d)}$ provides the foundation for threshold LNNs. References 1, 7 and 8 contain good introductions to pattern recognition, combinatorial geometry, and convexity, respectively. We assume here that the reader is familiar with the basics of linear algebra (Reference 9). However, for the reader's convenience, we present some standard definitions and well-known facts.

An affine subspace in $R^{(d)}$ is a translate of a (linear) subspace. A k-flat in $R^{(d)}$ is a k-dimensional affine subspace. That is, H is a k-flat provided $H = U + v$ for some k-dimensional subspace U and some $v \in R$.

We adopt the convention that the empty set is the unique k-flat for all negative k, whereas every singleton {y} is a 0-flat.

DEFINITION 1.1. A hyperspace in $R^{(d)}$ is a linear subspace of dimension d - 1.

DEFINITION 1.2. A hyperplane in $R^{(d)}$ is a translate of a hyperspace. That is, H is a hyperplane in $R^{(d)}$ provided there exist a hyperspace U and a vector v, both in $R^{(d)}$, such that $H = U + v$.

FACT 1. U is a hyperspace in $R^{(d)}$ provided there exists a nonzero vector a in $R^{(d)}$ such that

$$U = \{y \in R^{(d)} : a \cdot y = 0\},$$

where $a \cdot y$ denotes the inner product of a and y,

$$a \cdot y = \sum_{i=1}^{d} a_i y_i$$

FACT 2. H is a hyperplane in $R^{(d)}$ provided there exists a nonzero vector a in $R^{(d)}$ and a scalar b such that

$$H = \{y \in R^{(d)} : a \cdot y - b = 0\}.$$

It will prove helpful to view the set of hyperplanes as characterized both by Definition 1.2 and Fact 2.

Suppose $\mathcal{H}$ is finite set of hyperplanes in $R^{(d)}$, and $1 \le e \le d+1$. We say that $\mathcal{H}$ is in general position provided that the dimension of the intersection of any e-subset of $\mathcal{H}$ is d - e.

DEFINITION 2.1. An open half-space H^O in $R^{(d)}$ is a subset satisfying

$$H^O = \{y \in R^{(d)} : a \cdot y - b < 0\}$$

where $a \in R^{(d)}$ and $b \in R$. Similarly a closed half-space H^- in $R^{(d)}$ is a subset satisfying

$$H^- = \{y \in R^{(d)} : a \cdot y - b \geq 0\}.$$

DEFINITION 2.2. A convex polyhedron is the intersection of a finite number of closed half-spaces.

From the preceding definition it follows that a convex is polytope is a convex polyhedron. However, a convex polyhedron need not be a polytope, since polytopes must be compact (bounded). A compact convex polyhedron is convex polytope.

Now consider a LNN $\mathcal{T} = (\tau, h\ \Lambda)$, where τ is the threshold transfer function satisfying

$$\tau(t) = \left.\begin{array}{l} -1 \text{ for } t < 0 \\ \\ 1 \text{ for } 0 \leq t \end{array}\right\}.$$

For weight matrices $A_1, A_2, ..., A_h$, the network mapping F is given by

$$F(x) = f_h(f_{h-1}(... f_2(f_1(x)) ...)),$$
$$f_j = T \circ A_j, \text{ for } 1 \leq j \leq h.$$
$$T = (\tau, \tau, ..., \tau)^T.$$

Let G(x) denote the result of mapping x through only the first layer of neurons, i.e., the second layer of nodes. Equivalently, G equals f_1 and is the mapping for the network $(\tau, 1, M)$, where $M = (d, \lambda)$, $d = \lambda_0$, and $\lambda = \lambda_1$. We focus attention on the network M and its mapping G for the following two reasons. If $G(x) = G(y)$ for $x, y \in R^{(d)}$, then $F(x) = F(y)$. Thus, input patterns that are to be mapped into different outputs must be separated by G. This applies to all LNNs, but is not of great consequence when s is injective (one-to-one). The second reason applies specifically to threshold neurons; namely, the range of the set-valued function.

$$G^{-1} : R^{(\lambda)} \rightarrow 2^{(R^{(d)})}$$

consists of a finite number of disjoint convex polyhedra in $R^{(d)}$. Therefore, many properties of threshold networks depend largely upon decompositions of $R^{(d)}$ into polyhedra.

For the remainder of this section, we let $A = A_1$, the single λ by $d + 1$ weight matrix that defines G.

The following theorems and corollaries illuminate the relationship between the weight matrix A and the decomposition of the input space $R^{(d)}$. We let

$$G(x) = (G_1(x), G_2(x), ..., G_\lambda(x)),$$

where

$$G_j(x) = \tau(a(j) \cdot x^+),$$

and a(j) is the j^{th} row of A. Recalling that the rows a(j) of A are (d + 1)-vectors, we denote by $\tilde{a}(j)$ the truncated d-dimensional row vector:

$$\tilde{a}(j) = (a(j, 1), a(j, 2), ..., a(j, d)).$$

In order to avoid discussing degenerate configurations, we assume that the truncated row vectors $\tilde{a}(j)$, $1 \le j \le \lambda$, lie in general position in $R^{(d)}$. That is, for $1 \le e \le d$, every e-subset of the $\tilde{a}(j)$s is linearly independent.

LEMMA 1. $G_j^{-1}(-1)$ and $G_j^{-1}(1)$ are complementary open and closed half-spaces in $R^{(d)}$, respectively, for $1 \le j \le \lambda$.

LEMMA 2. For $\sigma \in \{-1, 1\}^{\lambda}$, the closure of $G^{-1}(\sigma)$ is a convex polyhedron in $R^{(d)}$.

PROOF. $G^{-1}(\sigma) = \bigcap_{j=1}^{\lambda} G_j^{-1}(\sigma_j)$. Since $G^{-1}(\sigma)$ is the intersection of finitely many half-spaces, its closure is a convex polyhedron. (We consider the empty set to be a convex polyhedron.)

THEOREM 1 (References 6 and 7). Let $\mathcal{H}$ be a set of λ hyperplanes in general position in $R^{(d)}$, then $R^{(d)} - \cup\mathcal{H}$ is the union of $Reg_d(\lambda)$ connected components, each of which is a convex polyhedron, where

$$Reg_d(\lambda) = \binom{\lambda}{0} + \binom{\lambda}{1} + ... + \binom{\lambda}{d}.$$

DEFINITION 3. For $X \subseteq R^{(d)}$, we say that a partition $\{X_1, X_2\}$ of X into two disjoint subsets is linearly separable if there exists a hyperplane H that separates every pair of points $\{x_1, x_2\}$ with $x_1 \in X_1$, $x_2 \in X_2$

FACT 3. $\{X_1, X_2\}$ is linearly separable provided there exists a vector a and a scalar b such that

$$a \cdot x - b \quad \begin{cases} < 0 \text{ if } x \in X_1 \\ > 0 \text{ if } x \in X_2 \end{cases}.$$

DEFINITION 4. Suppose that $\mathcal{X} = \{X_1, X_2, ..., X_K\}$ is a finite family of subsets of $R^{(d)}$ and $\mathcal{H} = \{H_1, H_2, ..., H_\lambda\}$ is a family of hyperplanes in $R^{(d)}$. We say that $\mathcal{H}$ separates $\mathcal{X}$ if for every $x_i \in X_i$, $x_j \in X_j$, $1 \le i < j \le K$, there is at least one member of $\mathcal{H}$ that separates x_i and x_j.

FACT 4. A finite set $\mathcal{H}$ of hyperplanes separates a finite family $\mathcal{X}$ of sets in $R^{(d)}$ if and only if every connected component of $R^{(d)} - \cup\mathcal{H}$ contains members of at most one member of $\mathcal{X}$ and $\cup\mathcal{H}$ is disjoint form $\cup\mathcal{X}$.

THEOREM 2 (Reference 10). Let X be a set of N points in general position in $R^{(d)}$, then the number of linearly separable partitions of X into two disjoint subsets is $Sep_d(N)$, where

$$Sep_d(N) = \binom{N-1}{0} + \binom{N-1}{1} + \ldots + \binom{N-1}{d}.$$

REMARKS. Theorems 1 and 2 are 'almost dual' to one another. By moving from $R^{(d)}$ to projective space $P^{(d)}$ with appropriately modified definitions, lines and planes may be interchanged by projective duality. The discrepancy between the formulas for $Reg_d(\lambda)$ and $Sep_d(N)$ results from the different topologies of $R^{(d)}$ and $P^{(d)}$. This difference is exemplified by the fact that a projective hyperplane does not disconnect $P^{(d)}$, while two projective hyperplanes decompose $P^{(d)}$ into two disjoint components. Of the $Reg_d(\lambda)$ components in $R^{(d)}$ determined by λ hyperplanes, $2Reg_{d-1}(\lambda-1)$ of them are infinite. These infinite regions occur in $Reg_{d-1}(\lambda-1)$ pairs that are connected when transformed into projective space $P^{(d)}$. Thus, the number of connected components determined by λ lines in $P^{(d)}$ is, in fact, $Sep_d(\lambda)$, as one would expect from duality.

EXAMPLE 1. For $d = 2$ and $\lambda = 4$, we have

$$Reg_2(4) = \binom{4}{0} + \binom{4}{1} + \binom{4}{2} = 11.$$

Figure 2 shows four lines in the plane and the resulting decomposition into 11 regions; three finite and eight infinite.

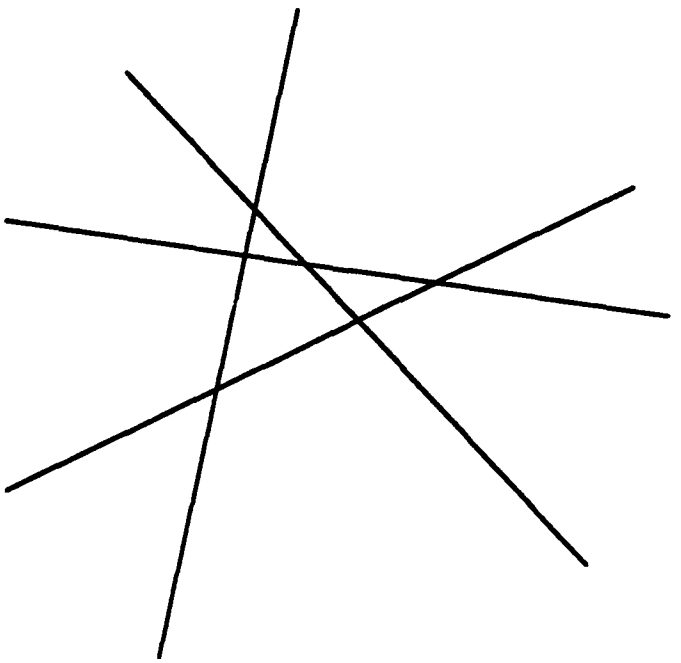


FIGURE 2. Eleven Regions in $R^{(2)}$ Determined by Four Lines.

EXAMPLE 2. Figure 3(a) shows a set $Y = \{y_1, y_2, y_3, y_4\}$ of four points in $R^{(2)}$. Table 1(a) shows the seven linear separations of Y.

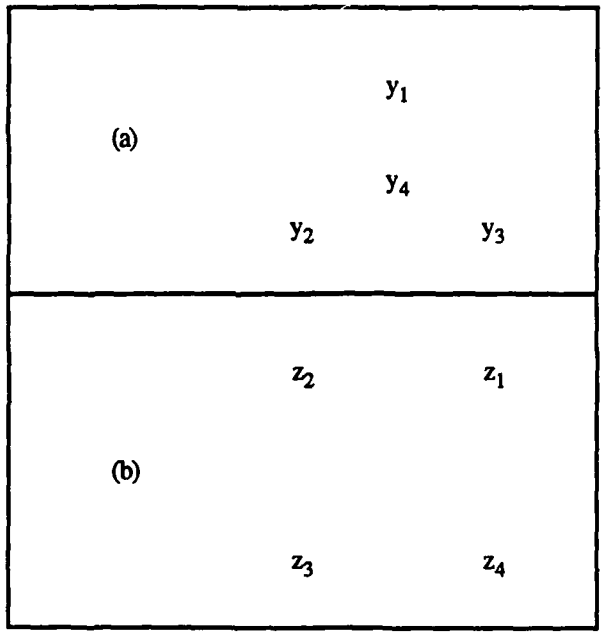


FIGURE 3. Quadruples Y and Z in $R^{(2)}$.

TABLE 1(a). Separable Partitions of Y in Figure 3(a).

Y_1	Y_2
Ø	$\{y_1\ y_2\ y_3\ y_4\}$
$\{y_1\}$	$\{y_2\ y_3\ y_4\}$
$\{y_2\}$	$\{y_1\ y_3\ y_4\}$
$\{y_3\}$	$\{y_1\ y_2\ y_4\}$
$\{y_1\ y_2\}$	$\{y_3\ y_4\}$
$\{y_1\ y_4\}$	$\{y_2\ y_3\}$
$\{y_1\ y_3\}$	$\{y_2\ y_4\}$

EXAMPLE 3. Figure 3(b) shows a set $Z = \{z_1, z_2, z_3, z_4\}$ of four points in $R^{(d)}$. Table 1(b) shows the seven linear separations of Z.

TABLE 1(b). Separable Partitions of Z in Figure 3(b).

Z_1	Z_2
$\emptyset$	$\{z_1\, z_2\, z_3\, z_4\}$
$\{z_1\}$	$\{z_2\, z_3\, z_4\}$
$\{z_2\}$	$\{z_1\, z_3\, z_4\}$
$\{z_3\}$	$\{z_1\, z_2\, z_4\}$
$\{z_4\}$	$\{z_1\, z_2\, z_3\}$
$\{z_1\, z_4\}$	$\{z_2\, z_3\}$
$\{z_1\, z_2\}$	$\{z_3\, z_4\}$

REMARKS. The 4-sets of Examples 2 and 3 both admit seven linearly separable partitions as indicated by Theorem 2. However, the sets of partitions are not isomorphic. That is, there exists no mapping from Y to Z that sends the linearly separable partitions of Y into those of Z. This stems from the fact that Y and Z represent different order types in $R^{(2)}$. Reference 11 presents definitions and basic results on order types in Euclidean spaces.

The fact that the partition

$$\{z_1, z_3\} \quad \{z_2, z_4\}$$

is not linearly separable is what prevents one from 'solving' the exclusive-or problem with $\lambda = 1$ (Reference 12).

The planar exclusive-or problem leads us naturally into pattern recognition in Euclidean spaces. We adopt the following simple model. We are given a family

$$\mathcal{X} = \{X_1, X_2, ..., X_K\}$$

of K disjoint finite subsets of $R^{(d)}$, with

$$X = X_1 \cup X_2 \cup ... \cup X_K$$

$$|X_j| = n_j,\ 1 \le j \le K$$

$$|X| = N = n_1 + {}_2 + ... + n_K.$$

This corresponds to a K-class problem for which X_i is the training sample for Class i. The task is to define a neural network (or some other type of classification device) whose mapping F satisfies

$$\|F(x_i) - u_i\| \le \varepsilon, \text{ for all } x_i \in X_i. \tag{1.1}$$

These conditions force F to map X_i into a sphere of radius ε centered at u_i. The u_i are distinct points in $R^{(m)}$, the output space, and ε is some small allowable error, in particular

$$2\varepsilon < \|u_i - u_j\| \text{ for } 1 \le i < j \le K, \tag{1.2}$$

so that the K target spheres in $R^{(m)}$ are disjoint.

Our results on neuron requirements pertain only to the number $\lambda = \lambda_1$ of first-layer threshold neurons required in a threshold network for the network mapping F to achieve the classification objective. We appeal to the following obvious fact.

FACT 5. If the network mapping F of a threshold network satisfies Equations 1.1 and 1.2, then for all $x_i \in X_i$, and $x_j \in X_j$, $1 \le i < j \le K$:

$$G(x_i) \neq G(x_j).$$

Lemma 3 follows immediately.

LEMMA 3. If there exist weights for a threshold network $\mathcal{T} = (\tau, h, \Lambda)$ such that the resulting F satisfies Equation 1, then the set of hyperplanes determined in $R^{(d)}$ by the λ_1 first-layer neurons separates $\mathcal{X}$.

In order to relate threshold neuron requirements to sets of training data, we introduce six combinatorial functions. In the following definition, point sets and sets of hyperplanes are assumed to be in general position in $R^{(d)}$.

DEFINITION 5. For X a finite subset of $R^{(d)}$, and $\mathcal{X} = \{X_1, X_2, ..., X_K\}$ a partition of X into K disjoint subsets:

$$\lambda_{min}(X/\mathcal{X}) = \min \{\lambda : \text{there exists a set } \mathcal{H} \text{ of } \lambda \text{ hyperplanes which separates } \mathcal{X}\}. \tag{5.1}$$

For a partition $\mathcal{N} = \{n_1, n_2, ..., n_K\}$ of N into K positive integers,

$$\lambda_{minmin}(d, N, \mathcal{N}) = \min \{\lambda_{min}(X/\mathcal{X})\} \tag{5.2}$$

$$\lambda_{maxmin}(d, N, \mathcal{N}) = \max \{\lambda_{min}(X/\mathcal{X})\} \tag{5.3}$$

where the minimum in Equation 5.2 and the maximum in Equation 5.3 are taken over all X, $\mathcal{X}$ such that X is an N-subset of $R^{(d)}$ and $\mathcal{X}$ is a partition of X into K disjoint subsets with cardinalities n_i, $1 \le i \le K$

$$\lambda_{min}(X) = \lambda_{min}(X/\mathcal{S})$$

where $\mathcal{S}$ denotes the family of singletons $\{\{x\} : x \in X\}$ (5.4)

$$\lambda_{minmin}(d, N) = \min \{\lambda_{min}(X) : X \subset R^{(d)} \text{ and } |X| = N\} \tag{5.5}$$

$$\lambda_{maxmin}(d, N) = \max \{\lambda_{min}(X) : X \subset R^{(d)} \text{ and } |X| = N\}. \tag{5.6}$$

For a K-class training set X, with

$$\mathcal{X} = \{X_1, X_2, ..., X_K\},$$

the number of first-layer threshold neurons required is at least $\lambda_{min}(X/\mathcal{X})$. If each X_i contains a single prototype, then at least $\lambda_{min}(X)$ first-layer threshold neurons are required.

LEMMA 4 (Reference 6). If X is an N-subset of $R^{(d)}$, then

$$\lambda_{min}(X) \geq \min \{\lambda : Reg_d(\lambda) \geq N\}.$$

PROOF. Suppose $Reg_d(\mu) < N$. A set $\mathcal{H}$ of μ hyperplanes in $R^{(d)}$ decomposes $R^{(d)}$ into less than N regions, so at least two of the N members of X must lie in the same region. Thus, X is not separated by $\mathcal{H}$. The conclusion of the lemma follows immediately.

Lemma 4 gives a lower bound for $\lambda_{min}(X)$ in terms of $|X|$. Thus, we have a lower bound for $\lambda_{minmin}(d, N)$. This lower bound is, in fact, sharp.

THEOREM 3.

$$\lambda_{minmin}(d, N) = \min \{\lambda : Reg_d(\lambda) \geq N\}.$$

PROOF. Let $\mathcal{H}$ be a set of μ hyperplanes in $R^{(d)}$, where

$$\mu = \min \{\lambda : Reg_d(\lambda) \geq N\}.$$

$\mathcal{H}$ decomposes $R^{(d)}$ into $r = Reg_d(\mu)$ disjoint regions. Select a point from each of the regions, and let X be an N-subset of the selected r-set. This is possible because $r \geq N$. $\mathcal{H}$ separates X, so $\lambda_{min}(X) \leq \mu$. But $\lambda_{min}(X) \geq \mu$ by Lemma 4. Thus, $\lambda_{min}(X) = \mu$, and the theorem follows.

EXAMPLE 4. Figure 4(a) shows a set W_1 of 11 point-classes (prototypes) for which $\lambda_{min}(W_1) = 4$. Since $Reg_2(4) = 11$, no set of 11 points in $R^{(2)}$ can be separated by fewer than four lines. Figure 4(b) shows a set W_2 of 11 point-classes for which $\lambda_{min}(W_2) = 6$.

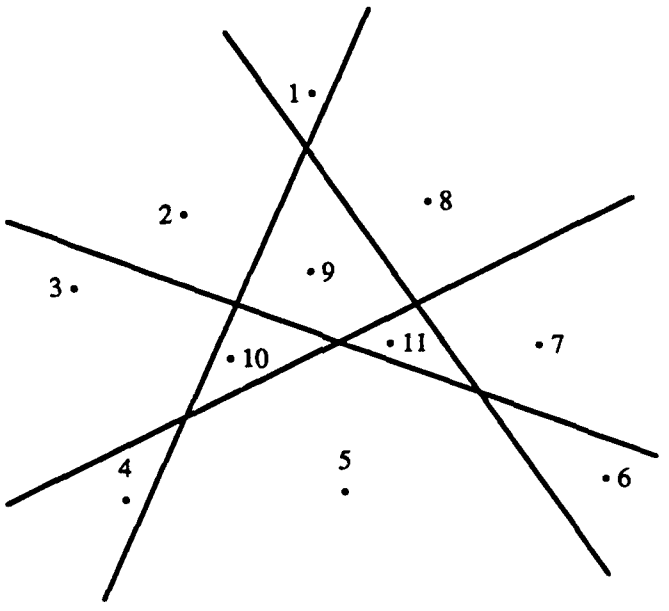


FIGURE 4(a). Eleven Points in $R^{(2)}$ Separated by Four Lines.

Lemma 4 and Theorem 3 determine $\lambda_{minmin}(d, N)$ exactly. Clearly, $\lambda_{minmin}(d, N)$ is also the lower bound on the number of threshold neurons required to separate any N-set X in $R^{(d)}$. As X varies through the N-subsets of $R^{(d)}$, $\lambda_{min}(X)$ is bounded above by $\lambda_{maxmin}(d, N)$. The set W_2 in Figure 4(b) shows that $\lambda_{maxmin}(2, 11) \geq 6$. This is a special case of the following theorem.

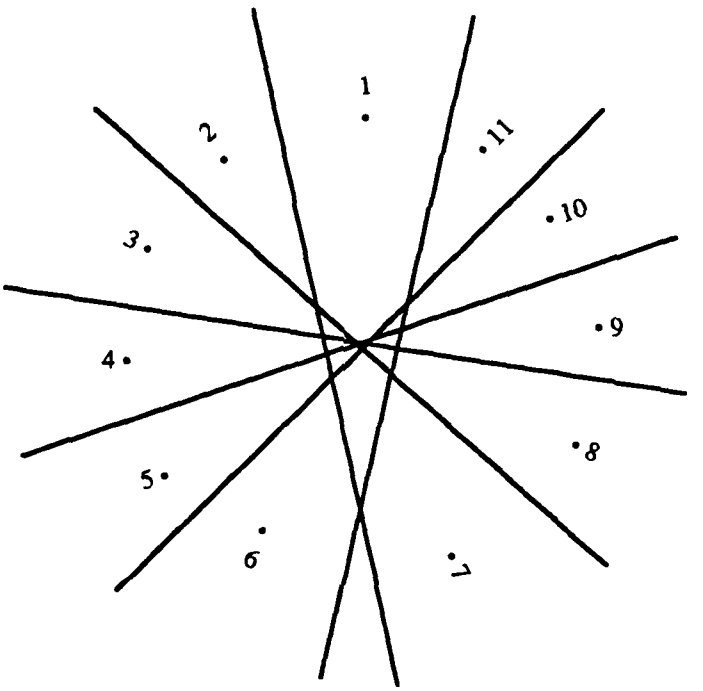


FIGURE 4(b). Eleven points in $R^{(2)}$ Requiring Six Lines for Separation.

THEOREM 4.

$$\lambda_{maxmin}(2, N) \geq \lceil N/2 \rceil .$$

PROOF. Let X be the set of N vertices of a convex N-gon C. Let B denote the boundary of C. B is a simple, closed polygonal curve containing N vertices, the members of X, and N edges, the line *segments joining consecutive members of X*. Every line in $R^{(2)}$ intersects B in at most two points. Since C is convex, every line in $R^{(2)}$ intersects at most two of the edges in B. Let $\mathcal{H}$ be a set of λ lines in $R^{(2)}$ that separate X. Since each member of $\mathcal{H}$ meets at most two edges in B, $\cup\mathcal{H}$ meets at most 2λ members of B. But every edge must be cut by at least one line since X is separated. It follows that $2\lambda \geq N$ and $\lambda \geq \lceil N/2 \rceil$.

The N-gon in $R^{(2)}$ provides an N-set that is difficult to separate, with difficulty measured by the number of lines required. This is a special case of N-sets lying on the moment curve in $R^{(d)}$ (Reference 7).

DEFINITION 6. The moment curve in $R^{(d)}$ is the set $M^{(d)}$ defined by

$$M^{(d)} = \{(t, t^2, t^3, ..., t^d) : t \in R\}.$$

Finite subsets of $M^{(d)}$ provide interesting examples in the study of convex polytopes. If X is a finite subset of $M^{(d)}$, then every point of X is an extreme point of Hull(X). Furthermore, X is difficult to separate, which is the property of interest here.

LEMMA 5. A hyperplane in $R^{(d)}$ cuts $M^{(d)}$ in at most d points.

PROOF. Let H be a hyperplane in $R^{(d)}$. From Fact 2, there exist a nonzero vector a and a scalar b such that

$$H = \{y \in R : a \cdot y - b = 0\}.$$

Suppose $y_i \in H \cap M^{(d)}$. Since $y_i \in M^{(d)}$, there exists $t_i \in R$, such that $y_i = (t_i, t_i^2, \ldots, t_i^d)$. Since $y_i \in H$, we have

$$a \cdot y_i - b = a_1 t_i + a_2 t_i^2 + \ldots a_d t_i^d - b = 0.$$

Thus, every $y_i \in H \cap M^{(d)}$ corresponds to a root of the polynomial

$$f_H(t) = a_1 t + a_2 t^2 + \ldots + a_d t^d - b.$$

The lemma follows from the fact that $f_H(t)$ has at most d roots.

LEMMA 6. Suppose $X \subset M^{(d)}$ and $|X| = N$. Then $\lambda_{min}(X) \geq \lceil (N - 1)/d \rceil$.

PROOF. We may assume the members x_i of X satisfy

$$x_i = (t_i, t_i^2, \ldots, t_i^d),$$

where

$$t_1 < t_2 < \ldots < t_N.$$

Let M_i be the open interval in $M^{(d)}$, connecting t_i and t_{i+1} for $1 \leq i \leq N - 1$. That is,

$$M_i = \{(t, t^2, \ldots, t^d) : t_i < t < t_{i+1}\}.$$

The ordering of the t_is guarantees that the M_is are disjoint. Now suppose that $\mathcal{H}$ is a family of λ hyperplanes that separates X. From Lemma 5 we know that each member H_j, $1 \leq j \leq \lambda$, of $\mathcal{H}$ cuts $M^{(d)}$ in at most d points. In order for X to be separated, each of the N - 1 segments M_i must be cut by a member of $\mathcal{H}$. It follows that

$$\lambda d \geq N - 1$$

and

$$\lambda \geq \lceil (N - 1)/d \rceil.$$

THEOREM 5.

$$\lambda_{maxmin}(d, N) \geq L_{maxmin}(d, N),$$

where

$$L_{maxmin}(d, N) = \lceil (N - 1)/d \rceil .$$

PROOF. This inequality follows directly from Lemma 6 and the definition of $\lambda_{maxmin}(d, N)$.

THEOREM 6. Discrete Ham Sandwich Theorem (Reference 7).

Suppose we have d sets $X_1, X_2, ..., X_d$ in $R^{(d)}$, with $|X_i| = n_i$. Then there exists a hyperplane H such that H bisects every X_i. That is, for $1 \le i \le d$, $|X_i \cap H^+|$ and $|X_i \cap H^-|$ differ by at most 1. For n_i odd the difference is 1, and for n_i even the difference is 0.

PROOF. See the Appendix.

DEFINITION 7. For positive integers d, N,

$$r = 1 + \lfloor \lg(d) \rfloor$$

and

$$U_{maxmin} = \begin{cases} \lceil \lg(N) \rceil \text{ if } N \le d+1 \\ r + \left\lceil \dfrac{N - 2^r}{d} \right\rceil \text{ if } N \ge d+2. \end{cases}$$

Theorem 6 enables us to bound $\lambda_{maxmin}(d, N)$ above. The upper bound, $U_{maxmin}(d, N)$, is established by an algorithm. The idea is best seen by working through an example.

EXAMPLE 5. Let X be a set of 45 points in $R^{(4)}$. We invoke Theorem 6 repeatedly to define a sequence $H_1, H_2, ..., H_{13}$ of hyperplanes that separate X. Initially, the best we can do is select H_1 to bisect X. This gives $X_1^{(1)}$, $X_2^{(1)}$ of cardinalities 23 and 22. Next, we select H_2 to simultaneously bisect $X_1^{(1)}$, $X_2^{(1)}$, giving subsets $X_1^{(2)}$, $X_2^{(2)}$, $X_3^{(2)}$, $X_4^{(2)}$ of cardinalities 12, 11, 11, 11. Next, we select H_3 to simultaneously bisect all four $X_j^{(2)}$'s. This gives eight subsets $X_j^{(3)}$'s of cardinalities 6, 6, 6, 5, 6, 5, 6, 5. At each remaining step we can bisect four of the existing components. We select the largest four at each step. Thus, at step 4 we bisect four of the five $X_j^{(3)}$'s of cardinality 6. This gives twelve components of cardinalities

3, 3, 3, 3, 3, 3, 5, 3, 3, 5, 6, 5.

The next four to be bisected have cardinalities 5, 5, 6, 5. The process yields 45 components after nine more steps, since the number of components increases by four at each step. Therefore,

$$\lambda_{min}(4, 45) \le 13 = U_{maxmin}(4, 45).$$

Note that at each step we may have created more components that those guaranteed by Theorem 6. We ignore this possibility and continue bisecting subsets as if they had not been cut by any earlier hyperplanes. The general algorithm follows.

SEPARATION ALGORITHM.

INPUT: An N-subset X of $R^{(d)}$

OUTPUT: A set $\mathcal{H} = \{H_1, H_2, \ldots H_U\}$

of U hyperplanes that separates X,

$$U = U_{maxmin}(d, N).$$

For r as defined above,

$$2^{r-1} \le d,$$

$$2^r > d.$$

If $N \le d$, then each application of Theorem 6, except perhaps the last, doubles the number of components. Thus, we obtain a separating set of size $\lceil \lg(N) \rceil$. The algorithm requires two parts: Steps A and B when $N > d$.

Step A consists of Steps A.k, $1 \le k \le r$.

Step A

Step A.1: Choose H_1 to bisect X giving components $X_1^{(1)}, X_2^{(1)}$.

⋮

Step A.k: Choose H_k to bisect the 2^{k-1} current components $X_j^{(k-1)}$.

After Step A, we have 2^r components $X_j^{(r)}$.

Step B consists of Steps B.k, $1 \le k \le s$, where

$$s = \left\lceil \frac{N - 2^r}{d} \right\rceil.$$

Step B

⋮

Step B.k: Choose H_{r+k} to bisect the d largest of the current

components $X_j^{(r+k-1)}$,

$1 \le j \le 2^r + d(k - 1)$.

⋮

After Step B.k, there are at least $2^r + dk$ components. Thus, the algorithm terminates after Step B.s, where s is the smallest integer satisfying

$$2^r + ds \ge N.$$

This gives a total of

$$U = r + s = r + \left\lceil \frac{N - 2^r}{d} \right\rceil \text{ hyperplanes}$$

that separate X. It follows that

$$\lambda_{min}(X) \le U_{maxmin}(d, N)$$

for $X \subseteq R^{(d)}$, and $|X| = N$. This result, together with Theorems 3 and 5, gives the following.

THEOREM 7. The number of hyperplanes required to separate N points in $R^{(d)}$ always lies between $\lambda_{minmin}(d, N)$ and $U_{maxmin}(d, N)$ with λ_{minmin} and U_{maxmin} as defined in Theorem 3 and Definition 7.

The lower bound is sharp, and the upper bound cannot be reduced below

$$L_{maxmin}(d, N) = \left\lceil \frac{N - 1}{d} \right\rceil.$$

Equivalently,

$$L_{maxmin}(d, N) \le \lambda_{maxmin}(d, N) \le U_{maxmin}(d, N).$$

Table 2 shows $\lambda_{minmin}(d, 200)$, $L_{maxmin}(d, 200)$, and $U_{maxmin}(d, 200)$ for several values of d.

TABLE 2. Bounds for $\lambda_{min}(d, 200)$.

d	2	3	4	5	10	15	20	25
λ_{minmin}	20	11	9	8	8	8	8	8
L_{maxmin}	100	67	50	40	20	14	10	8
U_{maxmin}	100	68	51	42	23	17	14	12

Although $\lambda_{maxmin}(d, N, \mathcal{N})$ appears to be more complex than $\lambda_{maxmin}(d, N)$, because of the additional argument $\mathcal{N}$, it can be bounded using the preceding results together with another example from the moment curve.

THEOREM 8. Suppose that $\mathcal{N} = (n_1, n_2, ..., n_K)$ is a partition of N, with the parts n_i labeled so that

$$n_1 \geq n_2 \geq ... \geq n_K.$$

Then

$$\lambda_{maxmin}(d, N, \mathcal{N}) \leq U_{maxmin}(d, N) \tag{8.1}$$

$$\lambda_{maxmin}(d, N, \mathcal{N}) \geq L_{maxmin}(d, N) \qquad \text{if } 2n_1 \leq N + 1, \tag{8.2}$$

$$\lambda_{maxmin}(d, N, \mathcal{N}) \geq \left\lceil \frac{2N - 2n_1}{d} \right\rceil \qquad \text{if } 2n_1 > N + 1, \tag{8.3}$$

and

$$\lambda_{minmin}(d, N, \mathcal{N}) \geq \lambda_{minmin}(d, K). \tag{8.4}$$

PROOF. Any set of hyperplanes that separates $(X, \mathcal{S})$, where $\mathcal{S}$ is the set of singletons from X, also separates $(X, \mathcal{N})$. Thus, Equation 8.1 follows from Theorem 7. Similarly Equation 8.4 follows from the fact that the separating hyperplanes must form at least K distinct regions in $R^{(d)}$.

The lower bounds of Equations 8.2 and 8.3 are obtained from subsets of the moment curve $M^{(d)}$. Let

$$x_i = (t_i, t_i^2, ..., t_i^d) \qquad \text{for } 1 \leq i \leq N,$$

where $t_1 < t_2 ... < t_N$. The x_i's are consecutive points in $M^{(d)}$. We now assign the labels 1, 2, ..., K to the x_i's with frequencies $n_1, n_2, ..., n_K$, respectively. Let $b = (b_1, b_2, ..., b_N)$ be the sequence consisting of n_1 1's followed by n_2 2's followed by n_3 3's, etc. The sequence c of labels is defined by

$$c = (b_1, b_h, b_2, b_{h+1}, b_3, b_{h+2}, ...),$$

where $h = \lfloor (N + 3)/2 \rfloor$. For example, with d = 3, K = 4, N = 15, and $\mathcal{N} = (6, 5, 2, 2)$, we have

$$b = (1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 3, 3, 4, 4)$$

and

$$c = (1, 2, 1, 2, 1, 2, 1, 3, 1, 3, 1, 4, 2, 4, 2).$$

In this case, $2n_1 = 12 < 16 = N + 1$, and at least $5 = \lceil 14/3 \rceil$ hyperplanes are required to separate all pairs with different labels.

If $2n_1 \geq N + 1$, then $n_1 > N - n_1$. Thus, the number of 1's exceeds the number of remaining symbols. Placing the remaining symbols in separate intervals between 1's produces a sequence requiring two cuts for each symbol different from 1. The number of cuts is $2(N - n_1)$, which proves Equation 8.3. As an example, suppose K = 4, N = 15, and $\mathcal{N} = (10, 2, 2, 1)$. There are nine intervals separating the ten 1's, and $126 = \binom{9}{5}$ ways of placing the remaining symbols in five separate intervals. Each of the resulting sequences requires 10 cuts. One such sequence is

$$c = (1, 1, 1, 2, 1, 2, 1, 3, 1, 3, 1, 4, 1, 1, 1).$$

REMARKS. For most pattern recognition problems the training set $(X/\mathcal{X})$ satisfies $2n_1 \leq N + 1$. We have treated the case $2n_1 > N + 1$ for the sake of mathematical completeness. The extreme of the latter case occurs with $K = 2$, and $n_2 = 1$. Here, $\lambda_{min}(X/\mathcal{X}) = 1$ or 2. Let y be the single member of X_2. If y is an extreme point of Hull(X), then $\lambda_{min}(X/\mathcal{X}) = 1$. Otherwise, y can be separated from X_1 by a pair of parallel hyperplanes. This is accomplished by first choosing any hyperplane H through y that is disjoint from X_1. The two hyperplanes are then chosen parallel to H and on opposite sides of H, sufficiently close together so that no point of X_1 lies between them. This construction generalizes in the following way, and yields an improvement on Equation 8.1. Suppose $Y \subseteq X_j$ and $|Y| \leq d$. There exists a hyperplane H (which is unique if $|Y| = d$) which contains Y and is disjoint from X - Y. Since X - Y is finite, one can select two hyperplanes parallel to H and on opposite sides of H with no member of X - Y lying between them. Repeating this contruction on d-subsets of $X_2, X_3, ..., X_K$, one obtains a family $\mathcal{H}$ of λ hyperplanes that separate $(X/\mathcal{X})$ where

$$\lambda = 2\left(\left\lceil \frac{n_2}{d} \right\rceil + \left\lceil \frac{n_3}{d} \right\rceil + ... + \left\lceil \frac{n_K}{d} \right\rceil\right).$$

This provides a better upper bound for $\lambda_{maxmin}(d, N, \mathcal{N})$ when n_1 is sufficiently large.

EXAMPLE 8. Suppose $d = 3$, $K = 4$, $N = 50$, and $\mathcal{N} = (32, 6, 6, 6)$. Then the construction above shows that

$$\lambda_{min}(3, 50, \mathcal{N}) \leq 2\left(\left\lceil \frac{6}{3} \right\rceil + \left\lceil \frac{6}{3} \right\rceil + \left\lceil \frac{6}{3} \right\rceil\right) = 12,$$

whereas the upper bound provided by Therorem 8 is 18.

SEPARATION OF CONVEX SETS

We assume in this section that the classes to be separated occupy disjoint regions of the input space $R^{(d)}$. The task now becomes separation of regions rather than finite sets of points. This approach is useful in conjunction with certain data analysis techniques. Both cluster analysis and density estimation yield regions of the input space that are associated with single classes. Separating these regions at the first layer of a LNN becomes a requirement of the network mapping. Linear separability of finite sets generalizes to linear separability of subsets.

DEFINITION 8. Two subsets C_1, C_2 of $R^{(d)}$ are linearly separable provided there exists a hyperplane H that cuts every segment joining a point in C_1 to a point in C_2; equivalently, C_1 and C_2 lie in different components of $R^{(d)}$ - H.

DEFINITION 9. A set $\mathcal{H}$ of hyperplanes in $R^{(d)}$ separates a family $\mathcal{C}$ of disjoint subsets of $R^{(d)}$ provided that every segment, which joins two points lying in different members of $\mathcal{C}$, is cut by at least one member of $\mathcal{H}$.

FACT 6. Two finite subsets of $R^{(d)}$ are linearly separable if and only if their convex hulls are disjoint.

FACT 7. Two compact convex subsets of $R^{(d)}$ are linearly separable if and only if they are disjoint.

The preceding definitions and facts are used implicitly in the discussion below.

Clustering within classes generates partitions of each class into disjoint subsets. In general, there is no guarantee that the regions for Class i are disjoint from those of Class j. Successive refinements of the clusterings will, however, yield disjoint sets.

EXAMPLE 6. For a K-class problem, we have training data $(X/\mathcal{X})$ where

$$\mathcal{X} = \{X_1, X_2, ..., X_K\}$$

and

$$X = X_1 \cup X_2 \cup ... \cup X_K.$$

At step 1, each X_i is clustered to obtain a partition $\mathcal{X}(i, 1)$ of X_i. Replacing each element of $\mathcal{X}(i, 1)$ with its convex hull gives a covering $\mathcal{C}(i, 1)$ of X_i with convex sets. At step r, the partition $\mathcal{X}(i, r - 1)$ is refined (i.e., each of its components is partitioned) to obtain a partition $\mathcal{X}(i, r)$ of X_i. Again the convex hulls of the members of $\mathcal{X}(i, r)$ give a covering $\mathcal{C}(i, r)$ of X_i by convex sets. This procedure is continued through step s, when all of the members of the total covering

$$\mathcal{C}(s) = \mathcal{C}(1, s) \cup \mathcal{C}(2, s) \cup ... \cup \mathcal{C}(K, s)$$

are pairwise disjoint. This is possible, since the partition of each X_i into singletons satisfies the disjointness requirement.

REMARK. It should be noted that a set $\mathcal{H}$ of hyperplanes that separates every pair of sets in $\mathcal{C}(s)$ must also separate $(X/\mathcal{X})$. Conversely, if we have a set $\mathcal{H}$ of hyperplanes that separate $(X/\mathcal{X})$, the convex hulls of the unseparated subsets of X form a covering of X by disjoint convex sets. An obvious question is: Why cluster the data, rather than proceed directly to a search for a separating set $\mathcal{H}$ of hyperplanes? We have no precise answer. However, it may well be that a good way to start the search for $\mathcal{H}$ is to look for clusters.

Regions may also be associated with individual classes when the training data is used to define density functions. For K density functions $p_1, p_2, ..., p_K$, one may assign disjoint regions $R_1, R_2, ..., R_K$ to the classes in such a way as to maximize (some function of) the K associated probabilities:

$$P_i = \text{Prob}[x \in R_i \text{ given } x \in \text{Class i}];$$

for such a model, the problem ultimately becomes the separation of the K regions. As with the clustering model, each R_i may be replaced by a finite family of convex sets that cover R_i, so that all of the resulting sets are pairwise disjoint.

Thus, we proceed to consider the problem of separation of disjoint convex sets.

DEFINITION 10. For $\mathcal{C}$ a finite family of disjoint compact convex subsets of $R^{(d)}$, we define

$$\gamma_{min}(\mathcal{C}) = \min\{\gamma : \text{there exists a set } \mathcal{H} \text{ of } \gamma \text{ hyperplanes which separates } \mathcal{C}\} \quad (10.1)$$

$$\gamma_{minmin}(d, N) = \min\{\gamma_{min}(\mathcal{C})\} \quad (10.2)$$

$$\gamma_{maxmin}(d, N) = \max\{\gamma_{min}(\mathcal{C})\}, \tag{10.3}$$

where the minimum in Equation 10.2 and the maximum in Equation 10.3 are taken over all families $\mathcal{C}$ of N disjoint convex subsets of $R^{(d)}$.

Since a point in $R^{(d)}$ is a convex set, separation of convex sets includes separation of points as a special case. The following theorem is a consequence of Lemma 4 and Theorem 3.

THEOREM 9.

$$\gamma_{minmin}(d, N) = \min\{\gamma : Reg_d(\gamma) \geq N\}.$$

Theorem 9 says, in effect, that the easiest problems for convex sets are just the easy problems for points. However, the worst case cost of separability is much greater for general convex sets than for finite sets. Here, cost is measured by the number of hyperplanes required for separation. We employ two techniques to establish a lower bound for $\gamma_{minmin}(d, N)$. The first involves constructing families of convex sets as the Voronoi regions of finite families of points. The second technique replaces junction points in the arrangement of Voronoi regions with new convex sets.

LEMMA 7. $\gamma_{maxmin}(d, N) \leq \binom{N}{2}$.

PROOF. Lemma 7 follows from Fact 7. One hyperplane for each pair of disjoint convex sets suffices for separation.

DEFINITION 11. For $X = \{x_1, x_2, ..., x_N\}$ a finite subset of $R^{(d)}$, we define the Voronoi region V_i associated with x_i by

$$V_i = \{x \in R^{(d)} : \|x - x_i\| = \min_j \{\|x - x_j\|\}\}.$$

That is, V_i is the set of points whose nearest neighbor in X is x_i.

FACT 8. The interiors V_i^o of the N Voronoi regions for $\{x_1, x_2, ..., x_N\}$ are disjoint convex sets.

Voronoi regions are intimately related to linear discriminant functions. For $1 \leq i \leq N$, define the nearest prototype discriminant function F_i by

$$F_i(x) = \|x_i\|^2 - 2\, x \cdot x_i\,,$$

and for $1 \leq i, j \leq N$ let

$$D_{ij}(x) = F_j(x) - F_i(x).$$

$D_{ij}(x)$ is positive at those points that are closer to x_i than to x_j. Letting H_{ij}^+ be the set of points at which $D_{ij}(x)$ is positive, we have

$$V_i^o = \cap\{H_{ij}^+ : 1 \leq j \leq N, j \neq i\}.$$

Thus, each V_i^o is convex, since it is the intersection of half-spaces.

EXAMPLE 7. Figure 5 shows the Voronoi regions for a set X of four points in $R^{(2)}$. The points x_1, x_2, and x_3 are the vertices of a triangle, and $x_4 = (x_1 + x_2 + x_3)/3$ is the centroid of the triangle. For each pair x_i, x_j of points, there is a one-dimensional boundary separating their regions V_i^o and V_j^o. Therefore, separation of these four regions requires six hyperplanes.

The regions of Figure 5 generalize to $R^{(d)}$, $d \geq 3$. Let $x_1, x_2, ..., x_{d+1}$ be the vertices of a simplex in $R^{(d)}$, and let $x_{d+2} = (x_1 + x_2 + ... + x_{d+1})/(d + 1)$, the centroid of the simplex. Each of the $\binom{d+2}{2}$ pairs V_i^o, V_j^o of Voronoi regions shares a (d - 1)-dimensional boundary. Since no two of these boundary regions lie in the same hyperplane, $\binom{d+2}{2}$ hyperplanes are required to separate the d + 2 convex regions.

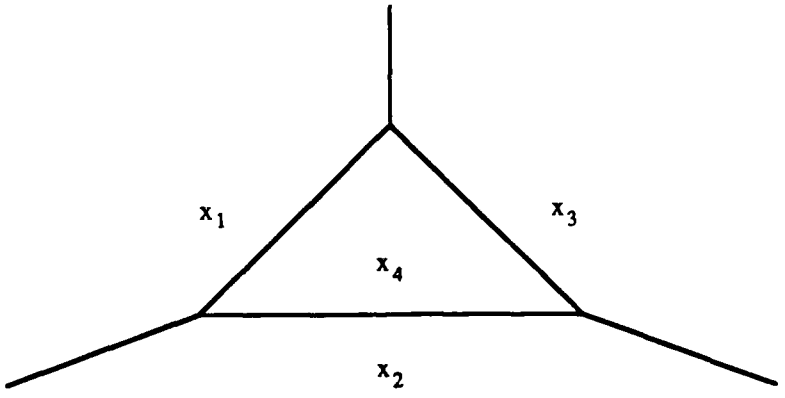


FIGURE 5. Four Voronoi Regions in $R^{(2)}$.

LEMMA 8. $\gamma_{maxmin}(d, d + 2) = \binom{d+2}{2}$.

PROOF. The simplex and its centroid are sufficient to show that $\gamma_{maxmin}(d, d + 2) \geq \binom{d+2}{2}$, and from Lemma 7 it follows that $\gamma_{maxmin}(d, d + 2) \leq \binom{d+2}{2}$.

Proceeding from the simplex construction, additional open convex regions may be added. The collective boundary of the d + 2 regions resulting from the simplex and its centroid contains d + 1 junction points, i.e., points at which d + 1 distinct (d - 1)-dimensional interfaces meet. By adding a new region, which is an open simplex containing a junction point, one obtains d + 3 regions requiring $\binom{d+2}{2} + d + 1$ hyperplanes. The d + 1 new hyperplanes contain the d + 1 boundaries of the new region. This construction also adds d + 1 new junction points while removing one. Thus, each additional region increases the number of junction points by d and the number of required hyperplanes by d + 1. After adding k new regions, there are dk + d + 1 junction points and d + k + 2 regions requiring a total of

$$\gamma = \binom{d+2}{2} + k(d+1)$$

hyperplanes for separation. Substituting N for d + k + 2 gives the following theorem.

THEOREM 10. $\gamma_{maxmin}(d, N) \geq (d+1)N - \binom{d+2}{2}$.

for $N \geq d + 2$.

Thus, the cost of separation grows as N/d for points, and at least as fast as N(d + 1) for convex sets.

CONJECTURE. We conjecture that 2d regions in $R^{(d)}$ may require $2d^2 - d$ hyperplanes, one for each pair of regions. If this is the case, then

$$\gamma_{maxmin}(d, N) \geq (d+1)N - 3d$$

for $N \geq 2d$.

SUMMARY

The feed-forward layered neural network is the simplest of the neural computing devices proposed for systems requiring pattern classification capabilities. The number of first-layer neurons imposes quantifiable limits on the amount of separation that the network can achieve in the pattern space. Conversely, the number and complexity of the pattern classes force minimal requirements on the size of the first layer of neurons.

This report establishes bounds on the number of first-layer neurons—in terms of input dimension and number of training patterns—required in a threshold network. Although, in general, neurons with continuous transfer functions are more versatile than threshold neurons, the separability capabilities of threshold neurons provide a baseline.

In order to completely separate N points (in general position) in d-dimensional Euclidean space, a threshold network may require as few as $Reg_d^{-1}(N)$ first-layer neurons or as many as N/d. $Reg_d^{-1}(N)$ is the minimum value of λ for which

$$\binom{\lambda}{0} + \binom{\lambda}{1} + \cdots + \binom{\lambda}{d} \geq N.$$

For example, the upper and lower bounds for 100 points in the plane are 14 and 50, respectively. For 5000 points in five-dimensional space, we obtain bounds of 16 and 1000. One does not expect to encounter either of these extremes in real applications. The upper bound, which arises from data lying on a one-dimensional curve in d-dimension space, is particularly unrealistic. On the other hand, it is unwise to offer a target number to cover all contingencies, since neuron requirements will obviously depend upon the application as well as upon the values of d and N. What is really required is either a fundamental understanding of the processes giving rise to the patterns, or exploratory data analysis of the training samples to determine their separability requirements. Either of these additional bodies of information will often yield not only the number of separating hyperplanes required, but the hyperplanes themselves. The hyperplanes in turn determine the weights on the first layer of connections.

As one might expect, the worst case separability requirements for finite families of convex sets are considerably greater, than for points. For N disjoint convex subsets of d-dimensional Euclidean space, the

number of hyperplanes required to separate every pair of subsets ranges from $Reg_d^{-1}(N)$ to $(N^2 - N)/2$. The lower bound is the same as for points, since points are convex sets. For $N \leq d + 2$, the upper bound is sharp. Moreover, for small (relative to the dimension d) families of sets, the upper bound is not totally unrealistic. For a family of multivariate distributions, the Voronoi regions of the class means may include sections from $(N^2 - N)/2$ hyperplanes among their boundaries. This upper bound has been proved here only for $N \leq d + 2$. We conjecture that it applies for $N \leq 2d$. In any event, the worst case hyperplane requirement for convex sets grows at least as fast as $(d + 1)N$, whereas the analogous growth for points is only N/d.

Sets of training data for supervised learning (pattern recognition) consist of disjoint unions of finite subsets of d-dimensional space. The distinct subsets of a training set are samples from a single class. The separation task for this problem is to create convex subsets of the pattern space each of which contains points of at most one class. Thus, separation of all pairs of points is not the objective. Surprisingly, in the worst case, all pairs of points must be separated in order to separate the classes. One would expect this type of situation to arise only when the underlying classes are nearly identical. That is, this extreme nonseparability among the classes of training samples indicates an unsolvable pattern recognition problem. Indeed, the number of hyperplanes required for class separation can be used as a criterion for solvability of the problem. As the requirements increase, solvability decreases.

Of greater interest than worst case neuron requirements are expected neuron requirements. Expected requirements, however, lead to the same dilemma that pervades computational complexity questions in theoretical computer science. Expectations are dependent upon assumptions regarding the distributions of the classes. Model 2 leads to questions of this type. We have treated the worst case problem at some length in this report for two reasons. The first is simply that these results follow easily from basic knowledge of convexity and combinatorial geometry. The second, more important, reason is the need to construct a firm mathematical framework that exhibits the intimate relationship between the pattern space and the role played by the first layer of neurons in the classification procedure. Understanding that such bounds exist is perhaps more helpful than knowing their exact integer values.

Appendix

DISCRETE HAM SANDWICH THEOREM

Our proof of the discrete Ham Sandwich Theorem uses the Borsuk Antipodal Mapping Theorem and several basic facts. The following definitions will prove helpful.

A median cut ofr a finite N-subset X of R(d) is a hyperplane H satisfying

$$|H^+ \cap X| \leq \lfloor N/2 \rfloor$$

and

$$|H^- \cap X| \leq \lfloor N/2 \rfloor.$$

The d-dimensional sphere $S^{(d)}$ is the boundary of the closed unit sphere in $R^{(d+1)}$; i.e.,

$$S^{(d)} = \{a \in R^{(d+1)} : \|a\| = 1\}.$$

For $u \in R^{(N)}$ and $1 \leq i \leq N$, $u_{(i)}$ denotes the value of the i^{th} smallest coordinate of u. Note that $u_{(i)}$ cannot always be associated with a unique coordinate of u.

EXAMPLE. If N = 6 and u = (4, -2, 1, 7, 3, 1), then

$u_{(1)} = -2$ $\qquad$ $u_{(4)} = 3$

$u_{(2)} = 1$ $\qquad$ $u_{(5)} = 4$

$u_{(3)} = 1$ $\qquad$ $u_{(6)} = 7$

Here, $u_{(2)}$ is equal to both u_3 and u_6.

FACT. For $1 \leq i \leq N$, the function $w_{(i)}^{(N)} : R^{(N)} \rightarrow R$, defined by

$$w_{(i)}^{(N)}(u) = u_{(i)},$$

is continuous.

We define the median, $Med^{(N)} : R^{(N)} \rightarrow R$, as follows:

$$Med^{(N)}(u) = \left\{ \begin{array}{ll} u_{(r)} & \text{for } N = 2r - 1 \\ \frac{1}{2}(u_{(r)} + u_{(r+1)}) & \text{for } N = 2r \end{array} \right\}.$$

FACT. Since $Med^{(N)} = w_{(i)}^{(N)}$ or $\frac{1}{2}(w_{(i)}^{(N)} + w_{(i+1)}^{(N)})$, $Med^{(N)}$ is continuous.

Now suppose that $X = \{x_1, x_2, ..., x_N\}$ is a finite subset of $R^{(d)}$ and $a \in S^{(d-1)}$. Define $f_X(a)$ by

$$f_X(a) = Med^{(N)}(P(x_1, a), P(x_2, a), ..., P(x_N, a)),$$

where P denotes inner product,

$$P(y, a) = y \cdot a.$$

Since $P : R^{(d)} \times S^{(d-1)} \rightarrow R$ is continuous, the mapping $f_X : S^{(d-1)} \rightarrow R$ is continuous for every finite X in $R^{(d)}$. The importance of the family of mappings f_X rests upon the following.

FACT. The hyperplane H perpendicular to a and passing through the point $f_X(a)a$ is a median cut for X.

Also of importance is the fact that f_X is antipodal, i.e., $f_X(-a) = -f_X(a)$ for all $a \in S$. This follows from the fact that

$$Med^N(-u) = -Med^N(u)$$

for all $u \in R^{(N)}$.

The following topological theorem provides the fundamental result required to prove the Discrete Ham Sandwich Theorem.

BORSUK ANTIPODAL MAPPING THEOREM. If F is a continuous mapping $S^{(d)}$ to $R^{(d)}$ satisfying $F(-a) = -F(a)$ for all $a \in S^{(d)}$, then 0 lies in the range of F. That is, for some $a \in S$

$$F(a) = (0, 0, ..., 0).$$

THEOREM 6. Discrete Ham Sandwich Theorem (Reference 7).

Suppose we have d sets X(1), X(2), ..., X(d) in $R^{(d)}$, with $|X(i)| = n_i$ and $\cup X(i)$ in general position. Then there exists a hyperplane H such that H bisects every X(i). That is, for $1 \leq i \leq d$, $|X(i) \cap H^+|$ and $|X(i) \cap H^-|$ differ by at most 1. For n_i odd the difference is 1, and for n_i even the difference is 0.

PROOF. For $1 \leq i \leq d-1$, define $F_i : S^{(d-1)} \rightarrow R$ by

$$F_i(a) = f_{X(i)}(a) - f_{X(d)}(a).$$

Since F_i is the difference of continuous functions from $S^{(d-1)}$ to R, F_i is continuous. Hence, the function F from $S^{(d-1)}$ to $R^{(d-1)}$ defined by

$$F(a) = (F_1(a), F_2(a), ..., F_{d-1}(a))$$

is also continuous. Moreover, $F(-a) = -F(a)$, since

$F_i(-a) = -F_i(a)$ for $1 \le i \le d-1$.

Applying the Borsuk Antipodal Mapping Theorem to F gives us a point $a \in S^{(d-1)}$ for which $F(a) = 0$. Thus, for $1 \le i \le d-1$, $f_{X(i)}(a) = f_{X(d)}(a)$. It follows that all $f_{X(i)}(a) = A$, where A is constant, and the hyperplane L through Aa perpendicular to a is a median cut for each of the sets X(i). Let $X = \bigcup_{i=1}^{d} X(i)$, $X^- = X \cap L^-$, $X^O = X \cap L$, $X^+ = X \cap L^+$. Similarly we let $X^-(i) = X(i) \cap L^-$, $X^O(i) = X(i) \cap L$, and $X^+(i) = X(i) \cap L^+$, for $1 \le i \le d$.

If X^O is empty, then H = L bisects all X(i). If not, we must rotate L in order to obtain the bisector H. Suppose X^O is non-empty. Let $n_i^- = |X^-(i)|$, $n_i^O = |X^O(i)|$, and $n_i^+ = |X^+(i)|$, for $1 \le i \le d$. Since X is in general position, X^O contains at most d points, and these points lie in general position in L. Thus, any split of X^O into two subsets may be effected by a (d - 1)-flat in L (the (d - 1)-flats are the hyperplanes of L). Our task is to select the split of X^O so as to bisect all of the X(i).

Let $m_i^- = \lfloor n_i/2 \rfloor - n_i^-$, $m_i^+ = \lceil n_i/2 \rceil - n_i^+$ Since L is a median cut for X(i), m_i^- and m_i^+ are non-negative. Moreover, $m^- + m^+ = n_i - n_i^- - n_i^+ = n_i^O$. Thus, we may partition X^O into Y^-, Y^+ as follows. Split X_i^O into disjoint subsets Y_i^-, Y_i^+ of cardinalities m_i^- and m_i^+, respectively, and let $Y^- = \bigcup_{i=1}^{d} Y_i^-$, $Y^+ = \bigcup_{i=1}^{d} Y_i^+$. Choose a (d - 1)-flat K in L which splits X^O into Y^- and Y^+. Let $\mathcal{H}$ be the family of hyperplanes in $R^{(d)}$ which contain K. Every member of $\mathcal{H}$, except L, splits X^O into Y^- and Y^+. Since X is finite, there is a neighborhood $\mathcal{N}$ of L in $\mathcal{H}$, all of those members split $X \setminus X^O$ into X^- and X^+. We choose H to be a member of $\mathcal{N} \setminus \{L\}$. Then H splits X into $X^- \cup Y^-$ and $X^+ \cup Y^+$. It follows that H bisects every X(i).

REFERENCES

1. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis.* New York, John Wiley and Sons, 1973.

2. K. Fukunaga. *Introduction to Statistical Pattern Recognition.* New York and London, Academic Press, 1972.

3. R. P. Lippmann. "An Introduction to Computing With Neural Nets," *IEEE ASSP Mag.* (April 1987), pp. 4-22.

4. Naval Weapons Center. *Equations of Learning and Capacity of Layered Neural Networks*, by Jorge M. Martin. China Lake, Calif., NWC, May 1989, 35 pp. (NWC TP 7013, publication UNCLASSIFIED.)

5. _____. *Neuro-Computing for Ship Classification*, by W. O. Alltop, P. R. Kersten and O. McNiel. China Lake, Calif., NWC, in process. (NWC TM 6772, publication UNCLASSIFIED.)

6. G. Mirchandani and W. Cao. "On Hidden Nodes for Neural Nets," *IEEE Trans. Circuits Syst*, CAS 36, No. 5 (May 1989), pp. 661-664.

7. H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Berlin, Springer-Verlag, 1987.

8. F. A. Valentine. *Convex Sets.* New York, McGraw-Hill, 1964.

9. B. Noble. *Applied Linear Algebra.* Englewood Cliffs, N. J., Prentice-Hall, 1969.

10. E. F. Harding. "The Number of Partitions of a Set of N Points in k Dimensions Induced by Hyperplanes," *Proc. Edinburgh Math. Soc*, Vol. 15 (1967), pp. 285-289.

11. J. E. Goodman and R. Pollack. "Multi-dimensional Sorting," *SIAM J. Computing*, Vol. 12, No. 3 (1983), pp. 484-507.

12. M. Minsky and S. Papert. *Perceptrons.* Cambridge, Mass., MIT Press, 1969.