NAWCWPNS TP 8071

Piecewise Linear Neural Networks

by
W. O. Alltop
Applied Mathematics Resources Group
Research Department

AUGUST 1992

NAVAL AIR WARFARE CENTER WEAPONS DIVISION
CHINA LAKE, CA 93555-6001

Approved for public release; distribution is unlimited.

FOREWORD

This report presents results pertaining to the mapping capabilities of layered networks of piecewise linear neurons. The work was performed during 1991 and 1992 as part of the Office of Naval Research Independent Research Program.

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

Approved by
R. L. DERR, *Head*
Research Department
14 August 1992

Under authority of
W. E. NEWMAN
RAdm., U.S. Navy
Commander

Released for publication by
W. B. PORTER
Deputy Commander for Research & Development

NAWCWPNS Technical Publication 8071

Published by . . . . .Technical Information Department
Collation . . . . Cover, 11 leaves
First printing . . . . 25 copies

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
August 1992

3. REPORT TYPE AND DATES COVERED
Interim 1991-1992

4. TITLE AND SUBTITLE
PIECEWISE LINEAR NEURAL NETWORKS

5. FUNDING NUMBERS
PE 61152N
PR R00N0
WU 13807002

6. AUTHOR(S)
W. O. Alltop

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

8. PERFORMING ORGANIZATION REPORT NUMBER
NAWCWPNS TP 8071

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) Modern weapons systems often require pattern recognition devices and algorithms. The three-layer neural network—consisting of input layer, hidden layer, and output layer—has great potential as a pattern classifier. Determining the size of the hidden layer and computing the weights are two of the primary design problems associated with layered networks. This report gives a lower bound on the number of hidden neurons as a function of the input and output dimensions and of the number of pattern prototypes. The neurons are assumed to have piecewise linear transfer functions.

14. SUBJECT TERMS
Neural Networks
Pattern Recognition
Geometry

15. NUMBER OF PAGES
20

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
UNCLASSIFIED

18. SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

19. SECURITY CLASSIFICATION OF ABSTRACT
UNCLASSIFIED

20. LIMITATION OF ABSTRACT
UL

INTRODUCTION

Modern weapons systems often require pattern recognition devices and algorithms. The three-layer neural network—consisting of input layer, hidden layer, and output layer—has great potential as a pattern classifier. Determining the size of the hidden layer and computing the weights are two of the primary design problems associated with layered networks. This report considers only piecewise linear networks; that is, networks with piecewise linear transfer functions. General introductions to layered networks appear in References 1 and 2.

Using linear algebraic methods, we determine a lower bound on the number of hidden neurons as a function of the input and output dimensions and of the number of pattern prototypes. Appropriate weights for matching the desired input/output pairs are constructed.

The network of interest is called a (d,L,m) network, where

d = dimension of the input space (number of input nodes)

L = dimension of the intermediate space (number of hidden neurons)

m = dimension of the output space (number of output nodes).

The network transfer function is denoted F_W, where W is the 'weight vector.' Actually, W consists of two matrices A and B and two vectors a and b.

A is L by d, a is L by 1

B is m by L, b is m by 1

$A^+(x) = Ax + a$ for $x \in R^{(d)}$

$B^+(u) = Bu + b$ for $u \in R^{(L)}$

$A^+ : R^{(d)} \longrightarrow R^{(L)}$

$B^+ : R^{(L)} \longrightarrow R^{(m)}$.

A^+ and B^+ are affine transformations, while A and B are linear. The weight vector W can be considered a member of $R^{(W)}$, where

$$w = dL + L + Lm + m \,,$$

the total number of entries (parameters) in A, a, B, and b. A typical weight initialization task involves finding a $W \in R^{(w)}$ for which

$$F_W(x_j) = y_j \text{ for } 1 \leq j \leq N \,,$$

where the (x_j, y_j)'s are desired input/output pairs.

In Reference 2 it is shown that one cannot find such a W for 'general' families of N input/output pairs when

$$N > w/m = L + 1 + (dL + L)/m \,. \tag{1}$$

For example, the number of general pairs for which the solution vector W exists in a (20, 30, 5) network is at most 157. The Inequality 1 holds for general sigmoidal neuron transfer functions.

This 'dimensional bound' on the number of general input/output pairs that a layered network can accommodate follows from comparing the space of output sets with the weight space. The space of all possible N-tuples $Y = (y_1, y_2, ..., y_N)$ of desired outputs has dimension mN. For a fixed N-tuple $X = (x_1, x_2, ..., x_N)$ of inputs, the set $S(X) = \{F_W(X) : W \in R^{(w)}\}$ of all output images of X must cover the mN-dimensional space of possible outputs. In the presence of appropriate hypotheses regarding the neuron transfer function it follows that

$$w \geq mN \,.$$

That is, the weight space, which is the domain of the function $F(X) : W \rightarrow F_W(X)$, must have dimension at least as great as that of the range, $R^{(mN)}$, which is the space of N-sets of outputs. It follows that

$$N \leq w/m \,.$$

It is shown in Reference 3 that the (d,L,m) network can accommodate at least d+1 input/output pairs when $d \geq L \geq m$ and at least L+1 pairs in any case. For $L \geq 3m/2$, the algorithm described in Section 3 allows one to raise the lower bound on the number of input/output pairs to d + L - 1.

BACKGROUND AND NOTATION

The piecewise linear neuron transfer function p is defined by

$$p(t) = \left.\begin{array}{lll} -1 & \text{for} & t < -1 \\ t & \text{for} & -1 \le t \le 1 \\ 1 & \text{for} & 1 < t \end{array}\right\} .$$

The network transfer function satisfies

$$F_W(x) = B\, p^{(L)}\, (Ax + a) + b .$$

Equivalently,

$$F_W = B^+ \text{ o } p^{(L)} \text{ o } A^+ ,$$

where $p^{(L)}$ is the extension of p to $R^{(L)}$,

$$p^{(L)}(u) = (p(u_1), p(u_2), ..., p(u_L))^T ,$$

$$u = (u_1, u_2, ..., u_L)^T ,$$

and o denotes function composition.

The following definitions from affine and linear geometry are useful.

Affine space. H is a k-dimensional affine subspace of $R^{(d)}$ provided $H = H_0 + a$, where $a \in R^{(d)}$ and H_0 is a k-dimensional linear subspace of $R^{(d)}$; i.e., H must be a translate of a k-dimensional linear subspace.

Affine equivalence. Two ordered k-subsets X and Z of $R^{(d)}$ are affine equivalent provided there is an invertible affine mapping

$$A^+ : R^{(d)} \rightarrow R^{(d)}$$

for which $A^+(x_j) = A^+(z_j)$ for $1 \le j \le k$.

Affine generation. For a finite subset,

$$X = \{x_1. x_2, ..., x_k\}$$

of $R^{(d)}$, the affine subspace generated by X, denoted <<X>>, is defined by

$$<<X>> = \{ \sum_j \alpha_j x_j : \sum_j \alpha_j = 1\} .$$

General position. A subset X of $R^{(d)}$ is in general position provided every k-subset S of X generates a (k-1)-dimensional affine subspace of $R^{(d)}$ for $1 \leq k \leq d+1$.

Convexity. A subset C of $R^{(d)}$ is convex provided $\lambda x + (1 - \lambda) y \in C$ whenever x, $y \in C$, and $0 \leq \lambda \leq 1$. Equivalently, a set is convex if it contains every line segment joining two of its members.

REMARKS. The affine subspace of $R^{(d)}$ generated by the set $X = \{x_1, x_2, ..., x_k\}$ is also equal to $< X_0 > + x_k$, where $X_0 = \{x_1, x_k, x_2 - x_k, ..., x_{k-1} - x_k\}$ and $< X_0 >$ denotes the linear subspace generated by X_0.

The close relationship between convexity and affine generation should be noted. The convex hull of a subset X of $R_{(d)}$ is defined by

$$\mathrm{Hull}(X) = \{ \sum_j \alpha_j x_j : \sum_j \alpha_j = 1 \text{ and all } \alpha_j \geq 0\} .$$

Thus, the convex hull is that part of the affine subspace determined by non-negative coefficients. The affine subspace generated by X is the intersection of all affine subspaces containing X. Similarly, the convex hull of X is the intersection of all convex sets containing X.

The importance of affine geometry and affine equivalence stems from the fact that each layer of weights defines an affine transformation. Thus, if there exist weights that map the inputs $X = \{x_i\}$ into the outputs $Y = \{y_i\}$, then there also exist weights for the pairs in X', Y' whenever X' is affine equivalent to X and Y' is affine equivalent to Y.

A basic result from convexity theory, which we state here without proof, is

RADON'S THEOREM. If X is a set of d+2 points in general position in $R^{(d)}$, then there is a unique partition $X = S \cup T$ of X into two proper, disjoint subsets such that $\mathrm{Hull}(S) \cap \mathrm{Hull}(T) \neq \phi$.

PROJECTIONS OF FINITE SETS

The network transfer function F_W of a (d,L,m) network consists of two affine mappings with a piecewise linear truncation (or squash) in between. The output of the hidden layer is the result of the affine mapping $x \rightarrow Ax + a$ followed by the L-dimensional truncation $p^{(L)}$. Thus, the ith coordinate $u^{(i)}(x)$ of the intermediate output is given by

$$u^{(i)}(x) = p(A_i x + a_i) ,$$

where A_i is the ith row of A and a_i is the ith coordinate of a. The mapping $u^{(i)}$: $R^{(d)} \rightarrow R$ is constant in each of the half-spaces D_- and D_+, where

$$D_- = \{x : A_i x + a_i < -1\}$$

$$D_+ = \{x : A_i x + a_i > 1\} .$$

In the infinite strip $D_0 = R^{(d)} \backslash (D_- \cup D_+)$,

$$u^{(i)}(x) = A_i x + a_i .$$

Thus, the mapping $u^{(i)}$ is piecewise affine on $R^{(d)}$, with its three affine parts determined by the affine projection $x \rightarrow A_i x + a_i$.

The following lemmas provide helpful tools for constructing the affine mappings required in a piecewise linear network.

LEMMA 1. If $S = \{s_j\}$ is a set of d+1 points in general position in $R^{(d)}$ and $x = (z_j)$ is a vector of d+1 reals, then there exists a unique affine functional $x \rightarrow f(x) = a x + \alpha$, satisfying $f(s_j) = z_j$, $1 \leq j \leq d+1$. Here a is a d-dimensional row vector and α is a real scalar.

PROOF. Let $t_j = s_j - s_{d+1}$ *and* $v_j = z_j - z_{d+1}$ *for* $1 \leq j \leq d$. Now let

$$T = [t_1, t_2, ..., t_d]$$

and

$$v = (v_1, v_2, ..., v_d) .$$

Since S is in general position, the t_j's form a basis. Thus, T is nonsingular and the equation $yT = v$ has a unique solution $y = a$ in $R^{(d)}$. Then the desired affine functional is $f(x) = ax + \alpha$, where $\alpha = z_{d+1} - as_{d+1}$. Uniqueness of f also follows from the nonsingularity of T.

Lemma 1 is the affine version of the fact that the linear functional mapping d independent vectors in d-space onto a prescribed set of d numbers exists and is unique.

LEMMA 2. Suppose X is a finite set in general position in $R^{(d)}$, $S = [s_1\ s_2, ..., s_d\}$ is a d-subset of X, K is a positive real, and $z = (z_1, z_2, ..., z_d)$ is a real d-vector. Then there exists an affine functional $f : x \rightarrow ax + \alpha$, satisfying

$$f(s_j) = z_j, \text{ for } 1 \leq j \leq d$$

and

$$| f(x) | \geq K, \text{ for } x \in X \backslash S .$$

PROOF. First note that the result is trivial if X = S. For $^{\mathrm{Y}}$ ≠ S, X\S is nonempty, and we let x_0 be a fixed member of X\S. Next, we apply Lemma 1 to the (d+1)-set $\{x_0\} \cup S$ twice: we first obtain the affine functional g satisfying

$$g(s_j) = z_j \text{ for } 1 \leq j \leq d$$

$$g(x_0) = 0$$

and then obtain the affine functional h satisfying

$$h(s_j) = 0 \text{ for } 1 \leq j \leq d$$

$$h(x_0) = 1 .$$

The desired affine functional is f = g + Mh, where

$$M = \max\left\{\frac{K + |\, g(x)\,|}{|\, h(x)\,|} : x \in X \setminus S\right\} .$$

Note that h maps all d members of S into 0. However, h is not the zero-mapping since $h(x_0) = 1$. It follows that the kernel of h is the hyperplane H_0 containing S. Since X is in general position, no members of X (other than those in S) lie in H_0. Thus, $h(x) \neq 0$ for all $x \in X\setminus S$ and M is well-defined. The summands g and Mh of f have the following properties:

g maps the members of S into the desired outputs

Mh spreads all the members of X\S away from 0 while preserving the desired values $g(s_j)$.

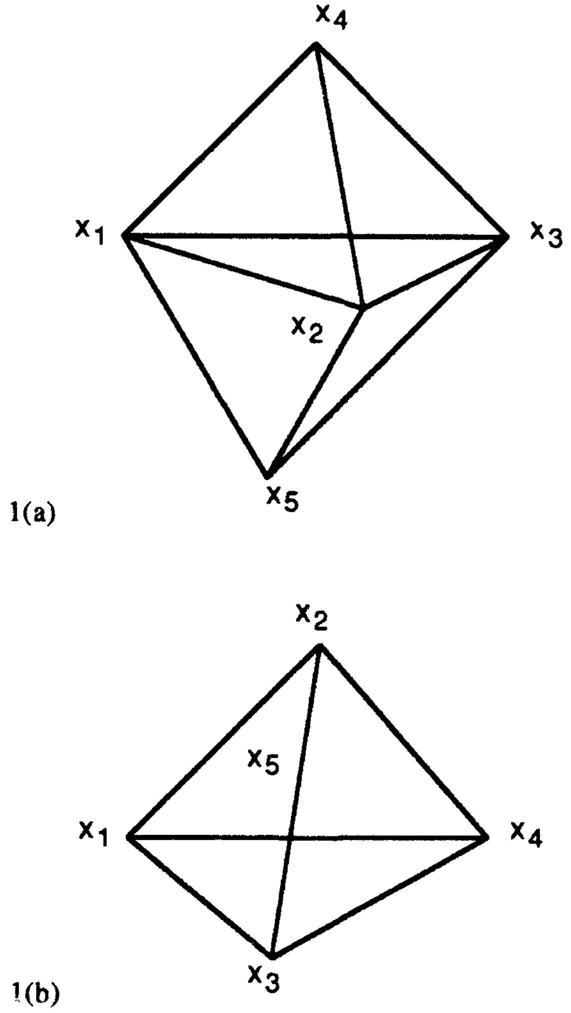
Repeated application of Lemma 2 to the L real outputs of the hidden layer allows one to send some inputs to the corners of the unit cube while placing the others at arbitrary locations in the unit cube. The combinatorial configuration of inputs in $R^{(d)}$ determines how the points in X\S are separated by the hyperplane through S. This technique is best understood by analyzing the following examples.

EXAMPLE 1. We consider a (3,3,2) network acting on five points in $R^{(3)}$. Let X = $\{x_1, x_2, x_3, x_4, x_5\}$ be a set of inputs in general position in $R^{(3)}$ and let Ext(X) be the set of extreme points (vertices) of Hull(X). At least one of the ten segments joining pairs of points from X meets the interior of Hull(X). More precisely, if $|\,\mathrm{Ext}(X)\,| = 4$, then there are four such interior segments, while there is only one if $|\,\mathrm{Ext}(X)\,| = 5$. (This follows from Radon's Theorem as stated in the previous section.)

Without loss of generality we may assume that $\{x_4, x_5\}$ bounds an interior segment of Hull(X). Figure 1 shows two different configurations of five points in

$R^{(3)}$ with $\{x_4, x_5\}$ an interior segment. Choose a basis $\{b_1, b_2, b_3\}$ for $R^{(3)}$ with $b_1 = x_4 - x_5$. Let $Q : R^{(3)} \rightarrow R^{(3)}$ be the linear transformation defined by

$$Qb_j = \left. \begin{array}{l} 0 \text{ for } j = 1 \\ \\ b_j \text{ for } j = 2 \text{ or } 3 \end{array} \right\}$$



1(a)

1(b)

FIGURE 1. Two Configurations in 3-Space With (x_4, x_5) an Internal Edge.

and let $q_j = Q\,x_j$. The mapping Q is a rank 2 mapping from $R^{(3)}$ onto the two-dimensional subspace Range(Q). The four points q_1, q_2, q_3, q_4, are distinct while $q_5 = q_4$. Moreover, since $\{x_4, x_5\}$ bounds an interior segment in Hull(X), q_4 must lie in the two-dimensional interior of Hull($\{q_1, q_2, q_3, q_4\}$). It follows that q_4 lies inside the triangle bounded by $\{q_1, q_2, q_3,\}$. Figure 2 shows the configuration of q's in $R^{(2)}$. The purpose of analyzing the image of X under Q is to select hyperplanes in $R^{(3)}$ for application of Lemma 2. The three lines in Range(Q) joining q_4 to q_1, q_2, and q_3 correspond to the three planes in $R^{(3)}$ determined by the triples $\{x_1, x_4, x_5,\}$, $\{x_2, x_4, x_5\}$, and $\{x_3, x_4, x_5\}$. The position of q_4 relative to $\{q_1, q_2, q_3\}$ determines how the three hyperplanes partition the remaining points. Indeed the partitions by the hyperplanes are identical to those of the lines in Range(Q). Let L_i denote the line through q_i and q_4 and let H_i denote the plane through x_i, x_4, and x_5, for $1 \leq i \leq 3$. The plane H_i separates the remaining two x's in $R^{(3)}$ if and onl·· if the line L_i separates the remaining two q's in Range(Q). Therefore, each plane H_i separates the remaining two points in $R^{(3)}$. Moreover, this fact is a consequence of our choosing an interior segment to define Q. Choice of an exterior edge to define Q would have resulted in only one of three hyperplanes separating the remaining two points. This geometry forms the basis for Example 2.

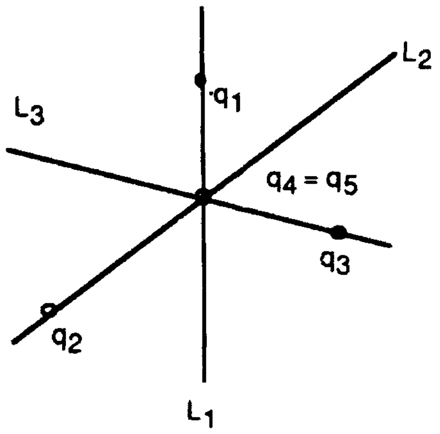


FIGURE 2. Projection Into Plane
For Example 1.

Lemma 2 allows us to define an affine mapping A_i^+ corresponding to the plane H_i, $1 \leq i \leq 3$. In each case, we select the values on the triple S to lie in the interval [-1,1] and set the lower bound K = 1. Thus, of the five outputs $u_{ij} = p(A_i^+(x_j))$, $1 \leq j \leq 5$, of the ith hidden neuron, $1 \leq i \leq 3$, three may be placed arbitrarily in

[-1,1], while the other two must be -1 and 1. Table 1 lists the outputs $u_j^T = (u_{1j}, u_{2j}, u_{3j})$, $1 \leq j \leq 5$ at the hidden layer.

TABLE 1. Coordinates of u_j for Example 1.

j	u_j
1	$(u_{11}\ \ -1\ \ 1)^T$
2	$(1\ \ u_{22}\ \ -1)^T$
3	$(-1\ \ 1\ \ u_{33})^T$
4	$(u_{14}\ \ u_{24}\ \ u_{34})^T$
5	$(u_{15}\ \ u_{25}\ \ u_{35})^T$

The nine variables u_{ij} in Table 1 can be independently chosen in the interval [-1,1]. This provides considerable flexibility in positioning the images u_j of the five inputs at the hidden layer so as to facilitate the final mapping into the desired outputs y_j. Figure 3 shows the five u's in $R^{(3)}$. The points u_1, u_2, and u_3 lie on edges of the cube. This follows from the fact that each of them has one variable coordinate, as shown in Table 1. Similarly, u_4 and u_5 each have all three coordinates variable, which allows them to be placed anywhere in the cube.

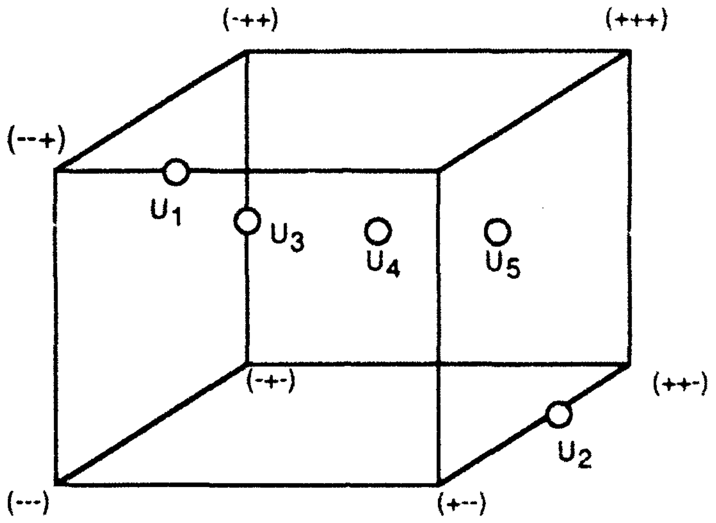


FIGURE 3. Five Points u_j of Example 1 in the 3-Cube; u_4, u_5 Can Be Anywhere.

EXAMPLE 2. Again, we consider a (3,3,2) network acting on the set X = (x_1, x_2, x_3, x_4, x_5) in $R^{(3)}$. At least six of the ten segments joining pairs of points of X must be edges in the boundary of Hull(X). In this example, we assume that (x_4, x_5) determines an exterior edge. Defining b_j, q_j, and Q as in Example 1 makes q_4 an extreme point of the boundary of Hull(q_1, q_2, q_3, q_4). It follows that only one of the three lines through the pairs (q_1, q_4), (q_2, q_4), and (q_3, q_4) separates the remaining two q's. We may assume that (q_2, q_4) is the line. Let H_1 be a plane in $R^{(3)}$, which intersects Hull(X) only in the line joining x_4 and x_5. Such a plane exists because (x_4, x_5) is an exterior edge of Hull(X). Next, we let H_2 and H_3 be the planes in $R^{(3)}$ through (x_1, x_4, x_5) and (x_3, x_4, x_5), respectively. The plane H_j corresponds to the line L_j; the three lines L_j are shown in Figure 4.

Applying Lemma 2, we again obtain a mapping A_i^+ corresponding to the plane H_i, $1 \leq i \leq 3$. Table 2 shows the five outputs u_j of the hidden layer, where $u_{ij} = p(A_i^+(x_j))$, $1 \leq j \leq 5$, as in Example 1.

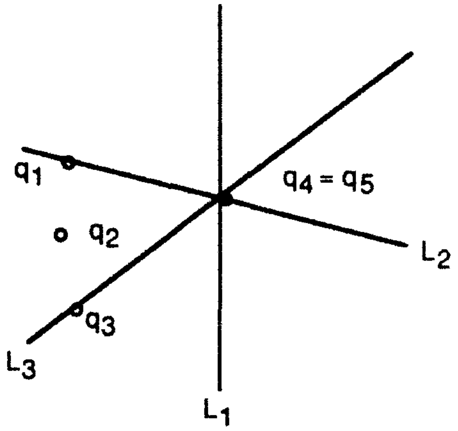


FIGURE 4. Projection Into Plane of Example 2.

In this geometry, eight variables u_{ij} in Table 2 can be chosen independently in the interval [-1,1]. Figure 5 shows the five u's in $R^{(3)}$: u_2 lies at the corner $(1, 1, -1)^T$; u_1 and u_3 lie on edges; and u_4 and u_5 can be anywhere. A critical feature of this geometry is that u_1, u_2, and u_3 can be placed arbitrarily close together by letting u_{21} approach 1 and u_{33} approach -1.

TABLE 2. Coordinates of u_j for Example 2.

j	u_j
1	$(1 \quad u_{21} \quad -1)$
2	$(1 \quad 1 \quad -1)^T$
3	$(1 \quad 1 \quad u_{33})$
4	$(u_{14} \quad u_{24} \quad u_{34})$
5	$(u_{15} \quad u_{25} \quad u_{35})$

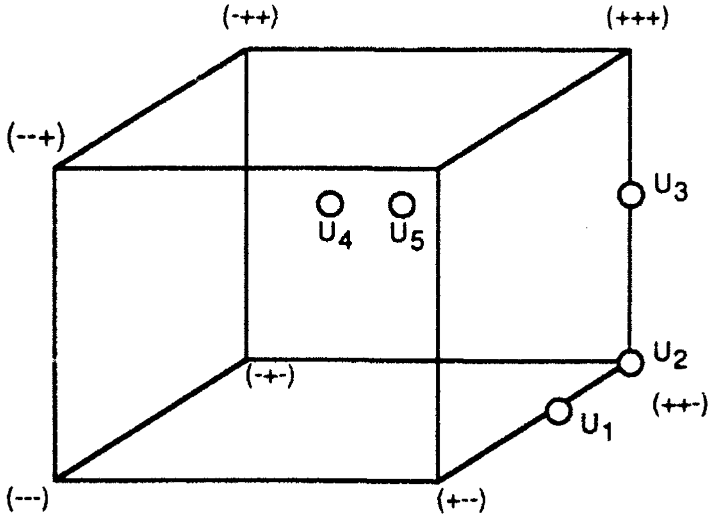


FIGURE 5. Five Points u_j of Example 2 in the 3-Cube; u_4, u_5 Can Be Anywhere.

WEIGHT ASSIGNMENT ALGORITHM

The geometry of Example 2 is the basis for the algorithm discussed in this section. We first present a brief description of the algorithm. Next, the algorithm will be applied to example 2, and finally the general algorithm will be described in detail.

Suppose that d, m, and N are given, i.e., we require a mapping from $R^{(d)}$ to $R^{(m)}$, which accommodates N given input/output pairs. The first task is to choose L. The following cases arise when considering how large L must be.

Case 1. $N \leq d + 1$.
Let $L = \min [d, m]$.

Case 2. $d + 2 \leq N \leq d + 3m/2 - 1$.
Let $L = 3n$, with $n = \lceil m/2 \rceil$.

Case 3. $d + 3m/2 \leq N$ and $m \geq 2$.
Let $L = 3n$, with $n = \lceil (N - d + 1)/3 \rceil$.

Case 4. $d + 3m/2 \leq N$ and $m = 1$.
Let $L = \min [L_1, L_2]$, where $L_1 = 3n$, with $n = \lceil (N - d + 1)/3 \rceil$, and $L_2 = \lceil N/2 \rceil$.

In Case 1, N is small enough to employ the methods of Reference 3. The algorithm presented here is intended for Cases 2, 3, and 4. When $L_2 < L_1$ in Case 4, a method presented in Reference 4 is used. In Cases 2, 3, and 4, the inequalities

$$N \leq d + L - 1$$

and

$$3\,m \leq 2L$$

both hold. These are the only requirements for the algorithm.

L = 3n in both Cases 2 and 3. If 3m < 2L, we increase m to 2L/3, adding new coordinates to each output while maintaining general position in $R^{(m)}$. Likewise, we add new input/output pairs, if necessary, to achieve N = d + L - 1. These modifications enable us to assume that N = d + L - 1, L = 3n and m = 2n. In the presence of these assumptions, the input to the algorithm consists of

(I1) Network parameters d, L, m, with L = 3n

(I2) Set X of N input points in general position in $R^{(d)}$, N = d + L - 1

(I3) Set Y of N desired outputs in general position in $R^{(m)}$.

We assume, of course, that y_j is the desired output for x_j. Hence, we seek a set W of weights for the (d, L, m) network satisfying

$$F_W(x_j) = y_j \text{ for } 1 \leq j \leq N\ .$$

The algorithm proceeds as follows. First one must determine a facet of a facet of Hull(X); i.e., a subset S of X for which there exists a facet F of X satisfying $S \subset F$, and $|S| = d - 1$. A facet F of X must be a d-subset, so a facet of F must be a (d-1)-subset. In Figure 1(a) all of the pairs of points except (x_4, x_5) are facets of facets. For example, (x_1, x_2, x_4) is a facet of Hull(X) and (x_1, x_2) is a facet of (x_1, x_2, x_4).

Assume, without loss of generality, that

$$S = (x_{3n+1}, x_{3n+2}, \ldots, x_N)\ .$$

Let G be the (d-2)-dimensional hyperplane through S and let P be the 2-dimensional linear subspace of $R^{(d)}$ perpendicular to G. Let Q be the orthogonal projection from $R^{(d)}$ to P. Q maps S onto a single point s in P.

Since S is a (d-2)-face of Hull(X), there is a hyperplane H_O through S (which must contain G), which does not separate X\S. H_O intersects P in a line L_O through s, which does not separate QX\{s} in P. Thus, there is a linear ordering on the members of $QX \backslash \{s\}$ determined by the angles between L_O and the vectors $Qx_j - s$ in P. We assume that $(x_1, x_2, ..., x_L)$ is the linear order. It should be noted here that a cyclic ordering is always induced on a set in the plane by specifying a center point. It is only when the center point is an extreme point of the hull that the 'endpoints' are uniquely defined.

The significance of the linear ordering is the following. The hyperplane H_j through $\{x_j\} \cup S$ decomposes $X \backslash (\{x_j\} \cup S)$ into

$$\{x_1, x_2, ..., x_{j-1}\} \cup \{x_{j+1}, x_{j+2}, ..., x_{N-d+1}\}$$

for $1 \leq j \leq N - d + 1$.

Each of the hyperplanes H_j, $0 \leq j \leq N - d + 1$, yields an affine functional via Lemma 2. Acting on the N points of X with these functionals and 'squashing' gives the $N - d + 2$ coordinates for each of the N points shown in Table 3. The image of X_j in $(N - d + 1)$-space is u_j. These points can be considered the output at a hidden layer containing $N - d + 2$ neurons. Those coordinates that can assume any value in $[-1, 1]$ are marked '*'.

Our algorithm does not use all of the N-d+2 columns in Table 3 as neuron coordinates at the hidden layer. The appropriate array of coordinates is constructed by deleting columns numbered $3k-1$, $1 \leq k \leq n$, and duplicating the columns numbered $3k$, $1 \leq k \leq n-1$. This leaves an array with L columns corresponding to the L hidden neurons. One additional modification is required. The variable entry in one member of each pair of duplicated columns is fixed. The effect of this construction is the clustering of triples of outputs at n corners of the cube, $n = L/3$. The transpose of the L by N array of hidden layer coordinates is shown in Table 4 with the columns relabeled 1 to L. The rows are partitioned into triples to illustrate the clustering.

TABLE 3. Coordinates of X in (N-d+2)-Space.

j	u_j^T								
	H_0	H_1	H_2	H_3	H_4		H_{N-d-1}	H_{N-d}	H_{N-d+1}
1	1	*	-1	-1	-1	. . .	-1	-1	-1
2	1	1	*	-1	-1	. . .	-1	-1	-1
3	1	1	1	*	-1	. . .	-1	-1	-1
4	1	1	1	1	*	. . .	-1	-1	-1
			⋮					⋮	
N-d	1	1	1	1	1	. . .	1	*	-1
N-d+1	1	1	1	1	1	. . .	1	1	*
N-d+2	*	*	*	*	*	. . .	*	*	*
N-d+3	*	*	*	*	*	. . .	*	*	*
			⋮					⋮	
N-1	*	*	*	*	*	. . .	*	*	*
N	*	*	*	*	*	. . .	*	*	*

*Can be anything in [-1, 1].

TABLE 4. Coordinates of X in L-Space.

j	u_j^T									
	1	2	3	4	5	6		L-2	L-1	L
1	1	*	-1	-1	-1	-1	. . .	-1	-1	-1
2	1	1	-1	-1	-1	-1	. . .	-1	-1	-1
3	1	1	*	-1	-1	-1	. . .	-1	-1	-1
4	1	1	1	1	*	-1	. . .	-1	-1	-1
5	1	1	1	1	1	-1	. . .	-1	-1	-1
6	1	1	1	1	1	*	. . .	-1	-1	-1
			⋮						⋮	
L-3	1	1	1	1	1	1	. . .	-1	-1	-1
L-2	1	1	1	1	1	1	. . .	1	*	-1
L-1	1	1	1	1	1	1	. . .	1	1	-1
L	1	1	1	1	1	1	. . .	1	1	*
L+1	*	*	*	*	*	*	. . .	*	*	*
L+2	*	*	*	*	*	*	. . .	*	*	*
			⋮						⋮	
N-1	*	*	*	*	*	*	. . .	*	*	*
N	*	*	*	*	*	*	. . .	*	*	*

*Can be anything in [-1, 1].

Next, we replace every * in entry (3k-2, 3k-1) with the value $1 - \delta$, $0 < \delta < 1$. Similarly, the * in entry (3k, 3k) is replaced by $-1 + \delta$. The rows of the resulting array are denoted $u_j^T(\delta)$. The following equations now hold among the $u_j(\delta)$:

$$u_{3k-1}(\delta) = (\overbrace{1,\ 1,\ \dots,\ 1}^{3k-1},\ \overbrace{-1,\ -1,\ \dots,\ -1}^{3n-3k+1})^T$$

$$u_{3k-2}(\delta) = u_{3k-1} - \delta e_{3k-1}$$

$$u_{3k}(\delta) = u_{3k-1} + \delta e_{3k}\ ,$$

where $u_{3k-1} = u_{3k-1}(1)$ and e_j is the jth elementary vector,

$$e_j = (\overbrace{0,\ 0,\ \dots,\ 0}^{j-1},\ 1,\ \overbrace{0,\ 0,\ \dots,\ 0}^{L-j})^T\ .$$

As δ approaches 0, each of the $u_j(\delta)$ approaches one of the corners u_{3k-1}. The following decomposition of $R^{(L)}$ is critical to the algorithm:

$$R^{(L)} = U + E\ ,$$

where $(u_2, u_5, \dots, u_{3n-1})$ is a basis for U and $(e_2, e_3, e_5, e_6, \dots, e_{3n-1}, e_{3n})$ is a basis for E. Here we let $u_{3k-1} = u_{3k-1}(\delta)$, since $u_{3k-1}(\delta)$ is independent of δ, $1 \leq k \leq n$. For all $\delta > 0$, the $u_j(\delta)$'s form a basis for $R^{(L)}$. Therefore, there exists a linear transformation $T(\delta)$ that maps every $u_j(\delta)$ into y_j, $1 \leq j \leq 3n$. $T(\delta)$ decomposes naturally into the following sum:

$$T(\delta) = T_1 + \frac{1}{\delta} T_2\ .$$

The summands T_1 and T_2 are defined by

$$T_1\ :\ \left.\begin{array}{l} u_{3k-1} \rightarrow y_{3k-1} \\ e_{3k-1} \rightarrow 0 \\ e_{3k} \rightarrow 0 \end{array}\right\}\quad \text{for } 1 \leq k \leq n$$

$$T_2\ :\ \left.\begin{array}{lll} u_{3k-1} & \rightarrow & 0 \\ e_{3k-1} & \rightarrow & y_{3k-1} - y_{3k-2} \\ e_{3k} & \rightarrow & y_{3k} - y_{3k-1} \end{array}\right\}\quad \text{for } 1 \leq k \leq n\ .$$

Of course, the objective of the algorithm is to enable a mapping that also sends u_j into y_j for $3n+1 \leq j \leq N$. The d-1 remaining points u_j can be placed anywhere in the L-cube. Thus, it suffices to choose δ so as to guarantee a preimage in the L-cube for every y_j, $3n+1 \leq j \leq N$.

Since the y_j's are in general position in $R^{(m)}$, the differences $y_{3k-2} - y_{3k-1}$ and $y_{3k-1} - y_{3k}$, $1 \leq k \leq n$, are linearly independent. It follows that rank $(T_2) = m = 2n$. Thus, the minimum singular value of $T_2 \mid E$ is positive:

$$\sigma = \underset{\min}{\sigma} (T_2 \mid E) > 0 \quad .$$

From this it also follows that

$$\| T_2 v \| \geq \sigma \| v \| \text{ for all } v \in E \quad .$$

We choose $\delta = \sigma / \max_j [\| y_j \|]$. This choice of δ yields a transformation $T(\delta)$ that admits preimages in the L-cube for all y_j, $3n+1 \leq j \leq N$. Indeed preimages exist in the intersection of E with the L-cube.

For $3n+1 \leq j \leq N$, choose $u_j \in E$ such that $T_2(u_j) = \delta y_j$. This is possible since $T_2 \mid E$ is a bijection $E \rightarrow R^{(m)}$. We have

$$\| \delta y_j \| = \| T_2(u_j) \| \geq \sigma \| u_j \| \, .$$

From this it follows that

$$\| u_j \| < \frac{\delta}{\sigma} \| y_j \| \leq 1 \, .$$

Thus, u_j lies in the L-cube. Finally for $3n+1$, $\leq j \leq N$,

$$T(\delta)(u_j) = (T_1 + \frac{1}{\delta} T_2)(u_j)$$

$$= \frac{1}{\delta} T_2(u_j) = y_j \, .$$

In summary, the determination of the weights proceeds as follows:

(1) Inputs

(1.1) d, L, m satisfying $N = d + L - 1$, $L = 3n$ and $m = 2n$.

(1.2) N pairs (x_j, y_j). The sets $X = \{x_j\}$ and $Y = \{y_j\}$ are in general position in $R^{(d)}$ and $R^{(m)}$, respectively.

(2) Find a facet F of Hull(X) and select a (d-1)-subset S of F. Let G be the (d-2)-dimensional affine subspace through S and let P be the 2-dimensional linear subspace orthogonal to G. Let Q denote the projection of X onto P. The entire set S projects onto a point s in P. Thus, $\mid Q \mid = N - d + 2 = 3n + 1$. Moreover, s is an extreme point of Hull(Q) in P. Let K be a directed line through s that does not separate $Q \backslash \{s\}$. For each q in $Q \backslash \{s\}$, the vector $q - s$ makes some angle $\theta(q)$

with K, $0 < \theta(q) < \pi$. This orders the q's. Relabel the pairs (x_j, y_j) so that $S = \{x_{3n+1}, x_{3n+2}, \ldots, x_N)$ and $\theta(q_1) < \theta(q_2) < \ldots < \theta(q_{3n})$.

(3) Determine the transformations T_1 and T_2 satisfying

$$T_1 + T_2 \,:\, u_j \rightarrow Y_j \;,\; 1 \leq j \leq 3n$$

$$T_1 \mid E = 0 \text{ and } T_2 \mid U = 0 \,.$$

Compute the minimum singular value, σ, of $T_2 \mid E$ and set $\delta = \sigma / \max\limits_j [\| y_j \|]$.

(4) Compute the preimage $u_j \in E$ of δy_j under T_2 for $3n+1 \leq j \leq N$:

$$u_j = \delta (T_2 \mid E)^{-1} y_j$$

for $3n+1 \leq j \leq N$. At this point all of the u_j's are known. For $1 \leq j \leq 3n$, they are the $u_j(\delta)$'s.

(5) Finally, compute the 3n affine functionals f_i (guaranteed by Lemma 2) that map the x_j's into the u_j's. The first layer of weights is determined by the f_i's, while the second layer is determined by $T_1 + \frac{1}{\delta} T_2$.

REFERENCES

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

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

3. ------. *On the Interpolation Properties of Feedforward Layered Neural Networks*, by Jorge M. Martin. China Lake, Calif., NWC, October 1990. (NWC TP 7094, publication UNCLASSIFIED.)

4. E. D. Sontag. "Feedforward Nets for Interpolation and Classification," to be published in *J. Comput. System Sci.*

INITIAL DISTRIBUTION

1 Commander in Chief, U. S. Pacific Fleet, Pearl Harbor (Code 325)
1 Commander, Third Fleet
1 Commander, Seventh Fleet
2 Naval Academy, Annapolis (Director of Research)
1 Naval War College, Newport (Library)
1 Air Force Intelligence Agency, Bolling Air Force Base (AFIA/INT, Maj. R. Esaw)
2 Defense Technical Information Center, Alexandria
1 Center for Naval Analyses, Alexandria, VA (Technical Library)