# Bounds on the Number of Hidden Neurons in Multilayer Perceptrons

Shih-Chi Huang and Yih-Fang Huang, *Member, IEEE*

*Abstract*—This paper investigates some fundamental issues concerning the capability of *multilayer perceptrons* with one hidden layer. The studies are focused on realizations of functions which map from a finite subset of $E^n$ into $E^d$. Both real-valued and binary-valued functions are considered. In particular, a least upper bound is derived for the number of hidden neurons needed to realize an arbitrary function which maps from a finite subset of $E^n$ into $E^d$. A nontrivial lower bound is also obtained for realizations of injective functions. This result will be useful in studying pattern recognition and database retrieval. In addition, an upper bound is given for realizing binary-valued functions that are related to pattern classification problems.

## I. INTRODUCTION

A FUNDAMENTAL question that is often raised in the applications of neural networks is "*how large does the network have to be to perform a desired task?*". Answers to this question are directly related to the capability of neural networks, and should be given independently of the learning algorithms employed. This paper investigates such issues of capability for a class of neural networks, i.e., multilayer perceptrons (MLP) with one hidden layer [1]. This investigation is focused on function realization by an MLP.

The input set considered in this paper is restricted to a finite subset of $E^n$ for the following two reasons. First, the theoretical results given in [2], [3] show that, if an MLP with one hidden layer is capable of realizing *arbitrary continuous functions* defined on a hypercube, the number of hidden neurons needed is infinite. Realistically, however, one can not build a device with infinite components. Second, if the MLP is trained by a learning-by-rote algorithm [4], e.g., the backpropagation algorithm, one can only expect the algorithm to have good performance with a finite training set. Furthermore, for those learning algorithms which update selectively by screening the input data [5]–[7], convergence to a desired solution in a *finite* number of updates can only be achieved when the training set is finite.

With regards to realization of an arbitrary function which maps from a finite subset $S$ of $E^n$ into $E^d$, the problem studied here is essentially the following: "*Given a finite set S, how many hidden neurons are needed for an MLP (with one hidden layer) to realize arbitrary functions defined on S?*". Section II provides an answer to this question by presenting an upper bound, which turns out to be consistent with the optimal number of hidden neurons determined by experimental results in [8], [9]. It is also shown that this upper bound is the least upper bound (LUB).

Section III then shows how to construct an MLP to realize arbitrary functions defined on a finite set. Interestingly enough, with the knowledge of the upper bound given in Section II, function realization by an MLP can be accomplished by simply assigning values of connection weights. Learning algorithms of the gradient descent type are needed only when such knowledge is unavailable and the number of hidden neurons is insufficient.

It is clear that a trivial lower bound on the number of hidden neurons of an MLP for realizing an arbitrary function is one. To obtain a nontrivial bound, Section IV addresses realizations of a particular class of functions, i.e., the injective functions, which nullify the possibility of grouping inputs with the same output in one region. For realizing such one-to-one functions, the problem studied here is "*how many hidden neurons does an MLP need to partition the input space such that different inputs will be contained in different regions?*". A lower bound on the number of hidden neurons is herewith derived by employing the function counting theorem [10]–[13]. However, an MLP with hidden neurons equal to the given lower bound can only separate certain kinds of input sets. A necessary condition of the distributed patterns for such sets is also given in Section IV.

The problem of realizing an arbitrary binary-valued function defined on a finite set is essentially that of constructing an arbitrary dichotomy on that set. It will be seen that, for dichotomy construction on arbitrary finite sets, the least upper bound on the number of hidden neurons is the same as the LUB given in Section II for realizing arbitrary functions. However, the general position [8], [14] condition may be imposed on the input set to obtain a tighter bound. In Section V, the input set is decomposed into a sequence of subsets, in which every subset is in general position and is contained in a linear variety [15]. With this decomposition, a different bound, somewhat tighter than the one obtained with the general position condition, is obtained for the case that the input set may not be in general position. Moreover, if the input set is continually decomposed, this upper bound will eventually approach the LUB.

Section VI concludes this paper by discussing the capability of a fixed size MLP. In particular, the upper (lower) bounds on the number of hidden neurons considered on a $k$-element input set given in the preceding sections will be converted into the lower (upper) bounds on the number of elements in a finite input domain on which the concerned collection of functions can be realized by an MLP with $m$ hidden neurons.

## II. THE LUB ON THE NUMBER OF HIDDEN NEURONS

Because this notation will be used throughout the paper, it is useful to explain what each term means in the context of the multilayer perceptron with one hidden layer. The input is represented by an $n$-dimensional vector $x$ with components $x(j)$. In the first layer, the input $x$ goes to each of $m$ perceptrons. The

$i$-th perceptron has weights and threshold represented by an ($n$ + 1)-dimensional vector $w(i) = [w(i1), w(i2), \cdots, w(in), t(i)]^T$ with scalar components $w(ij)$ and $t(i)$. In the second layer, there is one linear perceptron which has weights and threshold represented by an ($m$ + 1)-dimensional vector $w^* = [w^*(1), w^*(2), \cdots, w^*(m), t^*]^T$ with scalar components $w^*(i)$ and $t^*$. The vector $\theta = [w^T(1), w^T(2), \cdots, w^T(m), w^{*T}]^T$ represents the totality of all the weights $w(i)$ on the first layer and all the weights $w^*$ on the second layer. The function $M(\theta, x)$ represents the scalar output of the second layer of the multilayer perceptron with weight vector $\theta$ and input vector $x$ as shown in Fig. 1, where

$$h(i) = \text{sgn}\left[\sum_{j=1}^{n} x(j)w(ij) - t(i)\right] \qquad (1)$$

$$M(\theta, x) = \sum_{i=1}^{m} h(i)w^*(i) - t^*. \qquad (2)$$

Function realization studied here is formulated as follows: Let $f$ be a function defined on a finite set $S$ in the $n$-dimensional Euclidean space. Let $M(\theta)$ be an MLP with an adjustable parameter vector $\theta$. The $M(\theta)$ is said to be capable of realizing $f$ if there exists a $\theta$ such that $M(\theta, x) = f(x)$, for any $x \in S$. The problem is thus to determine $m$, the number of hidden neurons, such that $M(\theta)$ is capable of realizing arbitrary functions $f$ defined on a $k$-element set $S$, where $k$ is a positive integer greater than one. Extensive simulations in [8], [9] using different learning algorithms suggest that, in terms of learning efficiency, the *optimal* number of hidden neurons to realize a binary-valued function is $m = k - 1$. In this section, $m = k - 1$ is shown to be the least upper bound on the number of hidden neurons needed to realize an arbitrary real-valued function defined on $S$.

*Definition 1:* Let $S_1$ be a subset of a finite set $S$ in $E^n$, and let $S_2 = S - S_1$ be the complement of $S_1$ in $S$. Consider a hyperplane $H = \{x: a^T x = c\}$ in $E^n$ which partitions $E^n$ into two open halfspaces, i.e., $H_+ = \{x: a^T x > c\}$ and $H_- = \{x: a^T x < c\}$. The hyperplane $H$ is said to *separate* $S_1$ and $S_2$, if $S_1$ is contained in one of the halfspaces and $S_2$ is contained in the other one. If there exists such a separating hyperplane $H$, $S_1$ and $S_2$ are called (linearly) *separable* subsets of $S$.

*Definition 2:* An element $x_i \in S$ is a (linearly) *separable element* in $S$ if there exists a separating hyperplane which separates $\{x_i\}$ and $S - \{x_i\}$.

In this paper, it is assumed that the hidden neurons are characterized by the *signum* function, and that *separable* means specifically *linearly separable*.

*Lemma 1:* If $S$ is a nonempty finite set in $E^n$, there is at least one separable element in $S$.

*Proof:* Let $\max_{x_i \in S} \|x_i\| = r$. Construct a closed ball $\bar{B}(o, r)$ in $E^n$ with radius $r$ and center at the origin. Since $S$ is a finite set, there is an element in $S$, say $x_1$, such that $\|x_1\| = r$. Then, $x_1$ lies on the shell of $\bar{B}(o, r)$. Consider $H$, a tangent hyperplane to $\bar{B}(o, r)$ at $x_1$, which partitions the space into two open halfspaces $H_+$ and $H_-$ such that $\bar{B}(o, r) - \{x_1\}$ is contained in $H_-$.

Let $\min_{x_i, x_j \in S} \|x_i - x_j\| = \delta$. Construct a closed ball $\bar{B}(x_1, \delta)$ in $E^n$ with radius $\delta$ and center at $x_1$. Let $A$ be the intersection of the shell of $\bar{B}(o, r)$ and the shell of $\bar{B}(x_1, \delta)$. And, let $d$ be the distance between $H$ and an arbitrary element in $A$. If $H$ is shifted into $H_-$ with distance less than $d$, all the elements in $S$ are in $H_-$, except $x_1$ which is in $H_+$. So, $x_1$ is a separable element in $S$. $\qquad \nabla\nabla$
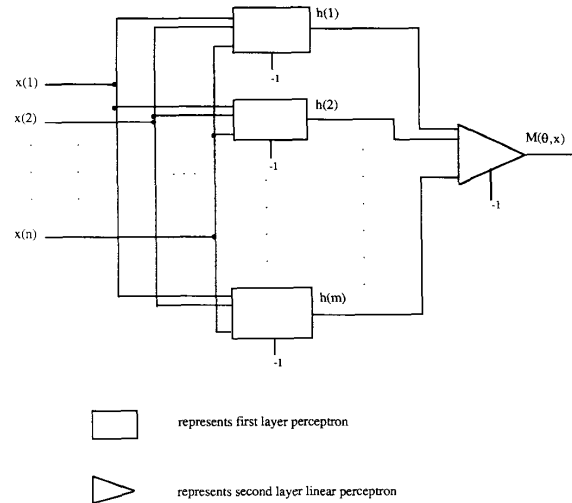


Fig. 1. An MLP with $m$ hidden neurons.

*Theorem 1:* Let $S$ be a $k$-element of $E^n$ where $k > 1$ is an integer, and let $f$ be an arbitrary real-valued function defined on $S$, i.e., $f: S \rightarrow E$. An MLP with $k - 1$ hidden neurons is capable of realizing $f$.

*Proof:* By Lemma 1, there is at least one separable element in $S$. Let $x_1$ be one such element in $S$. Let $H_1$ be a separating hyperplane of $\{x_1\}$ and $S_1 = S - \{x_1\}$ such that $x_1 \in H_{1+}$. Clearly, $S_1$ is also a finite set, and thus there is a separable element $x_2$ in $S_1$. Construct a separating hyperplane $H_2$ to separate $\{x_2\}$ and $S_2 = S_1 - \{x_2\}$ such that $x_2 \in H_{2+}$. Repeat this procedure till $H_{k-1}$ is obtained.

An MLP with $k - 1$ hidden neurons can now be constructed as follows: Let the vector $a_j$ and scalar $c_j$ of $H_j$ be, respectively, the values of connection weights and threshold of the $j$-th hidden neurons, $1 \leq j \leq k - 1$. Specifically, $[a_j^T, c_j]^T = w(j)$. Consider a $k$-by-$k$ matrix $D$ as follows: Let the $j$-th column of $D$ be the $j$-th hidden neuron outputs with respect to inputs $x_1$, $x_2, \cdots, x_k$, $1 \leq j \leq k - 1$. Furthermore, let every element in the $k$-th column of $D$ be 1. The hyperplane $H_j$ is constructed to separate $\{x_j\}$ and $\{x_{j+1}, \cdots, x_k\}$ such that $x_j$ is in $H_{j+}$, so $D$ is an upper triangular matrix with value 1 for each diagonal element (Fig. 2). All $x$ in Fig. 2 are undetermined elements which might be 1 or 0. Therefore, $D$ is nonsingular.

Now, define a vector $F = [f(x_1), f(x_2), \cdots, -f(x_k)]^T$ and a vector $P = D^{-1}F$, where the $k$-tuple vector $P$ is composed of the connection weights and threshold of the output neuron. Then, this MLP realizes the function $f$. $\qquad \nabla\nabla$

*Corollary 1:* An MLP with $k - 1$ hidden neurons is capable of realizing an arbitrary vector-valued function $g: S \rightarrow E^d$.

*Proof:* For any element $x_i$ in $S$, let $g(x_i) = [g_1(x_i), g_2(x_i), \cdots, g_d(x_i)]^T$, where $g_j: S \rightarrow E$, $1 \leq j \leq d$. The matrix $D$ constructed in Theorem 1 is independent of the function. Therefore, as long as the $j$-th output neuron is constructed to realize $g_j$, the overall MLP realizes $g$. $\qquad \nabla\nabla$

*Corollary 2:* Consider a function $h: \beta \rightarrow N$, where every element in $\beta$ is a doublet consisting of a particular $k$-element set $S$ and a function $f$ defined on $S$, and $N$ is the set of natural numbers. Define $h(S, f)$ to be the minimum number of hidden neurons in an MLP needed to realize the function $f$ defined on $S$. Then, $h(S, f) \leq k - 1$.

□ represents first layer perceptron

▷ represents second layer linear perceptron

$$\begin{bmatrix} 1x\,x. & \dots\ x\,1 \\ 0\,1\,x. & \dots\ x\,1 \\ 0\,0\,1. & \dots\ x\,1 \\ \cdots & \cdots \\ \cdots & \cdots \\ \cdots & \cdots \\ 0\,0\,0. & \dots\ 1\,1 \\ 0\,0\,0. & \dots\ 0\,1 \end{bmatrix}$$

Fig. 2. A $k$-by-$k$ matrix $D$.

Proof of this corollary can be easily inferred from Theorem 1. In fact, $k - 1$ is the least upper bound (LUB) on $h(S, f)$. To see this, consider the following example: Given a $k$-element set $S$ where $x_1, x_2, \cdots, x_k$ lie along a line in $E^n$, let $f$ be a function defined on $S$ such that $f(x_i) \neq f(x_{i+1})$, for any $1 \leq i \leq k - 1$. To realize $f$, at least $k - 1$ hyperplanes are needed to separate those $k$ elements from one another, as any hyperplane in $E^n$ that does not contain this line can only have at most one intersection with this line segment. So, $k - 1$ hidden neurons are needed to realize $f$.

## III. FUNCTION REALIZATION

In Section II, it was shown that an MLP with $k - 1$ hidden neurons can realize arbitrary functions defined on a $k$-element set. This section now presents a methodology for accomplishing such realizations. Section III-A shows how to construct an arbitrary switching function by such an MLP, while Section III-B discusses realization of an arbitrary function.

### A. Realization of Switching Functions

Let $S$ be the $2^n$-element set in $E^n$ where every element in $S$ lies on a corner of a unit hypercube, and let $S_1$ be an arbitrary $k$-element subset of $S$. Consider a switching function $f: S_1 \to \{0, 1\}$. If there exists a function $g: S \to E$ which is equivalent to $f$ on those $k$ elements in $S_1$ and $g$ is realized by a network $M$, we say that $M$ also realizes $f$. In the sequel, an MLP as shown in Fig. 1 is used to realize $f$, and two methods for assigning the values of connection weights and thresholds are discussed.

i) In this method, every hidden neuron is associated with a single element in $S_1$ and the relationship is given by the following observation.

*Observation 1:* Let $S$ be the $2^n$ element set in $E^n$ where every element in $S$ lies on a corner of a unit hypercube. Then, every element in $S$ is a separable element.

*Proof:* Let $x_j = (x_j(1), x_j(2), \cdots, x_j(n))$ be an arbitrary element in $S$, where $x_j(i)$ is either 0 or 1, $1 \leq i \leq n$. Consider a hyperplane $H_j$, where the $(n + 1)$-tuple characteristic vector, $[a_j^T, c_j] = [a_j(1), a_j(2), \cdots, a_j(n), c_j]$, of $H_j$ is

$$a_j(i) = 2(x_j(i) - 0.5), \qquad 1 \leq i \leq n \qquad (3.a)$$

and

$$c_j = -0.5 + \sum_{i=1}^{n} x_j(i). \qquad (3.b)$$

Given an arbitrary element $x_k$ in $S$,

$$a_j^T x_k - c_j = 2 \sum_{i=1}^{n} x_j(i) x_k(i) - \sum_{i=1}^{n} x_j(i) - \sum_{i=1}^{n} x_k(i) + 0.5. \qquad (4)$$

Since the components of $x_j$ and $x_k$ are either 0 or 1,

$$\sum_{i=1}^{n} x_j(i) x_k(i) \leq \sum_{i=1}^{n} x_j(i) \qquad \text{for all } x_j, x_k \in S, \qquad (5)$$

so

$$2 \sum_{i=1}^{n} x_j(i) x_k(i) - \sum_{i=1}^{n} x_j(i)$$
$$- \sum_{i=1}^{n} x_k(i) \begin{cases} \leq -1, & \text{if } x_j \neq x_k \\ = 0, & \text{if } x_j = x_k. \end{cases}$$
$$(6)$$

Combining (4) and (6), we have that $x_j$ is in $H_{j+}$ and all the other elements of $S$ are in $H_{j-}$. So, $H_j$ separates $x_j$ from all the other elements in $S$. Therefore, every element in $S$ is a separable element. $\nabla\nabla$

Let $x_j$ be an element in $S_1$. Following Observation 1, construct $H_j$ as the separating hyperplane of $x_j$ such that $x_j \in H_{j+}$. Furthermore, let $a_j$ and $c_j$, defined in (3), be the connection weights and threshold of the $j$-th hidden neuron, respectively, $1 \leq j \leq k - 1$. Let $f(x_j)$ be the output of the switching function with respect to the input $x_j$. To determine the values of the connection weights and threshold of the output neuron, i.e., $w^* = [w^*(1), w^*(2), \cdots, w^*(k), t^*]^T$, we first let $t^* = -f(x_k)$ where $x_k$ is the $k$-th element in $S_1$. And, let

$$w^*(j) = f(x_j) + t^*, \qquad 1 \leq j \leq k - 1. \qquad (7)$$

Then, this MLP maps every element in $S_1$ into its desired output and maps every element in $S_1^c = S - S_1$ into $f(x_k)$, and $f$ is realized by the MLP.

ii) In the previous method, each hidden neuron corresponds to a separating hyperplane for an element in $S_1$. Now, consider a case that each hidden neuron constructs a separating hyperplane for a separable subset of $S$. For any element $x_j = (x_j(1), x_j(2), \cdots, x_j(n))$ in $S_1$, consider a hyperplane $H_j$ whose characteristic vector is

$$a_j(i) = x_j(i), \qquad 1 \leq i \leq n \qquad (8.a)$$

and

$$c_j = -0.5 + \sum_{i=1}^{n} x_j(i). \qquad (8.b)$$

Given an element $x_k$ in $S$,

$$a_j^T x_k - c_j = \sum_{i=1}^{n} x_j(i) x_k(i) - \sum_{i=1}^{n} x_j(i) + 0.5. \qquad (9)$$

Following (5), $H_j$ separates the elements in $S$ in such a way that

$$\text{if } \sum_{i=1}^{n} x_j(i) x_k(i) < \sum_{i=1}^{n} x_j(i), \qquad \text{then } x_k \in H_{j-},$$

and

$$\text{if } \sum_{i=1}^{n} x_j(i) x_k(i) = \sum_{i=1}^{n} x_j(i), \qquad \text{then } x_k \in H_{j+}. \qquad (10)$$

In (10), an $x_k$ is in $H_{j+}$ if and only if all the nonzero components of $x_j$ have nonzero counterparts in $x_k$. Thus, the $j$-th hidden neuron performs an AND operation on the nonzero components of $x_j$. The AND operation performed is independent of those components of $x_j$ that are zero, so the connections from those components to the $j$-th hidden neuron can be removed. With this argument, for the case that $S_1 = S$, only $n(2^{n-1} - 1)$ connection weights, in contrast to $n(2^n - 1)$ as in Method i), are

needed for the hidden neurons in Method ii). Moreover, for the case that $S_1$ is a $k$-element subset of $S$, if every element in $S_1$ is chosen independently and with equal probability, Method ii) can save approximately half of the connection weights for the hidden neurons.

What needs to be done with Method ii) now is the determination of $w^*$, i.e., the connection weights between the hidden neurons and the output neuron. The network under consideration is essentially a higher order network [16], or a functional link net [17]. In [16] and [17], those connection weights are determined by the gradient descent type of learning algorithms, while a recursive method is used in [18]. In the following, a more efficient method is used.

Consider the situation that a switching function $f$ defined on a $k$-element input set $S_1$ is realized by an MLP. Suppose that, for some reasons, an element in $S_1^c = S - S_1$ becomes one in $S_1$ and, thus, a hidden neuron is added to the MLP to realize this $(k + 1)$-th input. If the MLP is constructed by Method i), $w(k)$ and $w^*(k)$, i.e., the connection weights and thresholds in the MLP related to the $(k + 1)$-th input, can be assigned using (3) and (7) without affecting the rest of the connection weights and thresholds.

If, on the other hand, the MLP is constructed by Method ii), $[a_{k+1}^T, c_{k+1}]$ can be assigned by (8) without changing other connection weights and thresholds in the first layer. However, most of the components in $w^*$ have to be changed so that the MLP can realize the $(k + 1)$-th input. As such, one may use the parallel network shown in Fig. 3 to calculate all the $k + 1$ recursive equations. The purpose of this network is to make a possible on-line assignment for Method ii) in the situation mentioned above. This network is similar to the Hopfield neural network for matrix inversion [19]. However, not every invertible matrix can be inverted by this network because the interaction between neurons caused by feedbacks may drive some of the operational amplifiers (neurons) out of the linear region. Fortunately, the connection matrix $[T_{ij}]$ in this network is a triangular one. As such, the weights can be calculated by a set of recursive equations and this circumvents the aforementioned problem. Therefore, this parallel network is capable of solving the $k + 1$ recursive equations and calculating the values of connection weights of the output neuron for Method ii).

### B. Realization of Arbitrary Functions

In this section, realization of arbitrary real-valued functions is considered. To realize an arbitrary function defined on a $k$-element set $S$, we first need to determine the values of connection weights and thresholds of the hidden neurons. Assume that $\min_{x_i, x_j \in S} \| x_i - x_j \| = \delta$ and that $\min_{x_i \in S} \| x_i \| = \mu$ are known. Arbitrarily pick an $x_1 \in S$ with $\| x_1 \| = r_1$. Let the characteristic vector of the first hidden neuron be $a_1 = x_1 / r_1$ and $c_1 = r_1 - \delta^2 / 4r_1$. Repeat this procedure until every element in $S$, except one element $x_i$ with $a_i^T x_i = \mu$, is chosen.

As in Theorem 1, construct a matrix $D$ which is nonsingular. Consider a $k$-by-$k$ matrix $D'$ as follows: Let the $j$-th row of $D'$ be the $(k + 1 - j)$-th row of $D^T$, $1 \leq j \leq k$. Then, if we let $[T_{ij}] = D'$, the connection weights and threshold of the output neuron can be determined by the parallel network given in Fig. 3. And the MLP realizes the given function.

### IV. A LOWER BOUND FOR INJECTIVE FUNCTIONS

For the MLP shown in Fig. 1, every hidden neuron constructs a hyperplane decision surface in the input space. An MLP with $m$ hidden neurons will partition the input space into a cer-
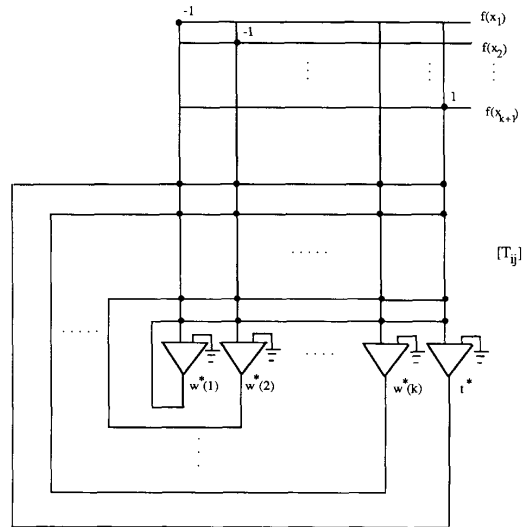


Fig. 3. A parallel network for calculating $(k + 1)$ recursive equations.

tain number of nonempty polytopes. A polytope is an intersection of a finite number of closed halfspaces. Thus, a polytope can be empty, bounded, or unbounded. An example is shown in Fig. 4(a) in which three lines partition $E^2$ into twenty-one unbounded polytopes and one bounded polytope. In this paper, an element is said to be contained in a polytope if this element is an interior point of the polytope. Also, $m$ hyperplanes partitioning $E^n$ may create some polytopes which is an intersection of fewer than $m$ closed halfspaces. For example, the shaded region in Fig. 4(b) is an unbounded polytope constructed by only two lines. To specify the nonempty polytopes constructed by $m$ hyperplanes, the following notion of $m$-polytope is needed for subsequent discussions.

*Definition 3:* An $m$-polytope is a nonempty intersection of $m$ closed halfspaces.

Consider $h(S, f)$ in Corollary 2, where $f$ is an arbitrary function. A trivial lower bound on $h(S, f)$ is one, which is the smallest natural number. In this section, $f$ is restricted to be an injective function defined on $S$. For the MLP shown in Fig. 1, inputs in the same $m$-polytope will map to the same output. Therefore, to realize such a one-to-one function, the number of hidden neurons has to be large enough so that every element in $S$ is contained in a different $m$-polytope. With this condition, we investigate the lower bound on the number of hidden neurons, namely, "*what is the minimum number of hyperplanes, denoted by $m_1$, needed to partition $E^n$ into more than $k$ $m_1$-polytopes?*". Answers to this question will provide useful information of the capability of MLP in pattern recognition and database retrieval [20]. The *function-counting theorem* and its extensions have been employed to provide an answer to this question [10]-[13]. The maximum number of $m$-polytopes obtained by partitioning $E^n$ with $m$ hyperplanes is

$$P(m, n) = \sum_{i=0}^{n} \binom{m}{i} \tag{11}$$

where

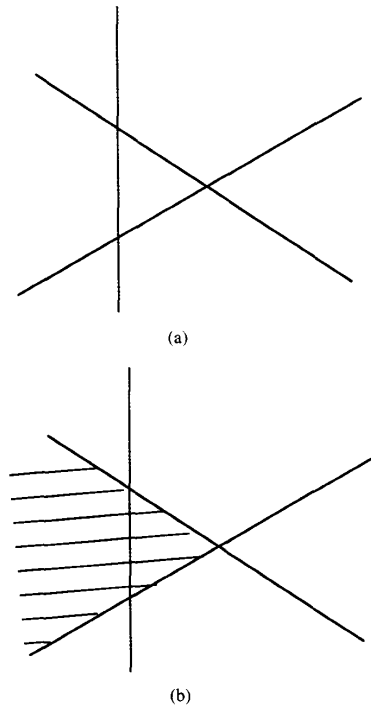$$\binom{m}{i} = 0, \quad \text{if } m < i.$$

Fig. 4. (a) Three line partition the two-dimensional space into twenty-four unbounded polytopes and one bounded polytope. (b) An unbounded polytope constructed by only two lines.
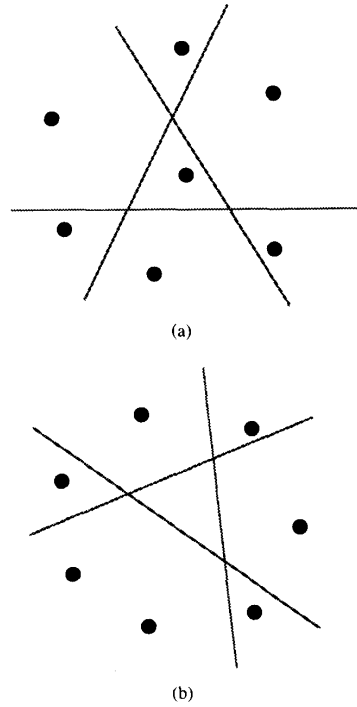


Fig. 5. (a) A seven-element set in the two-dimensional space which can be contained in seven 3-polytopes. (b) A seven-element set in the two-dimensional space which can not be contained in seven 3-polytopes.

Thus, for any given $k$,

$$m_1 = \min_m \left\{ P(m, n) \geq k \right\}. \tag{12}$$

The number of $m$-polytopes given by the function-counting theorem is an upper bound, and it can only be achieved when the $m$ hyperplanes partition the space in some particular ways. Therefore, the $k$ elements in $S$ are separated from one another by an MLP with $m_1$ hidden neurons only when $S$ fits those particular patterns. Consider, for instance, the seven-element sets $S_1$ in Fig. 5(a) and $S_2$ in Fig. 5(b). There exists a set of three lines which partition $E^2$ into seven 3-polytopes such that every element in $S_1$ is contained in a different 3-polytope. On the other hand, no matter how $E^2$ is partitioned by any set of three lines, there are two elements in $S_2$ which are contained in the same 3-polytope. The question here is "*what kind of $S$ will satisfy the restriction?*".

Note that an $m$-polytope can be either bounded or unbounded. The number of bounded $m$-polytopes $P_b(m, n)$ and that of unbounded $m$-polytopes $P_u(m, n)$ are given in [10] as

$$P_b(m, n) = \begin{cases} 0 & \text{if } m \leq n \\ \binom{m-1}{n} & \text{if } m > n \end{cases} \tag{13}$$

and

$$P_u(m, n) = \begin{cases} 2^m & \text{if } m \leq n \\ 2 \sum_{i=0}^{n-1} \binom{m-1}{i} & \text{if } m > n. \end{cases}$$

For $m \leq n$, $P_u(m, n)$ is achieved when the $m$-th hyperplane divides each of the $P_u(m - 1, n)$ $(m - 1)$-polytopes into two regions [12]. Therefore, when the number of elements $k \leq 2^n$, $S$ can be *continually dichotomized* into equal element subsets by hyperplanes, then every element in $S$ can be contained in one of the $P_u(m_1, n)$ $m_1$-polytopes where $m_1$ is defined in (12). An example for this is the $2^n$ element input set of switching functions. The following theorem, preceded by a lemma and two observations, gives a necessary condition for $S$ to satisfy the restriction. To begin, let $S$ be a nonempty finite subset of $E^n$, and a *convex hull* of $S$, denoted by $Co(S)$, is the intersection of all closed convex sets containing $S$. In this paper, a convex hull $Co(S)$ is a convex polyhedron in $E^n$ as $S$ is a finite set and thus an *extreme point* of $Co(S)$ is a vertex of the polyhedron [15].

*Lemma 2:* Let $S$ be a nonempty finite subset of $E^n$ and let $Co(S)$ be the convex hull of $S$. An element $x$ in $S$ is a separable element if and only if $x$ is an extreme point of $Co(S)$.

*Proof:* Let $x_1$ be an extreme point of $Co(S)$ and let $S_1 = S - \{x_1\}$. Then $x_1$ is an exterior point of $Co(S_1)$. According to the separating hyperplane theorem in [15], there is a hyperplane which separates $\{x_1\}$ and $Co(S_1)$. So, $x_1$ is a separable element in $S$.

Consider an $x_2 \in S$ which is not an extreme point of $Co(S)$. Then, every hyperplane $H$ in $E^n$ containing $x_2$ will either intersect at least one of the edges or contain one of the edges of $Co(S)$. For either cases, $S - \{x_2\}$ can not be contained in $H_+$ or in $H_-$. So, $x_2$ is not a separable element in $S$. $\nabla\nabla$

*Remarks:*
1. The above lemma says that the number of separable ele-

ments in a finite set $S$ is equal to the number of vertices of $Co(S)$.

2. It has been shown in the previous section that every element in the input set of switching functions is a separable element. It is easy to see that those elements are extreme points in the convex hull of the input set.

A supporting hyperplane $H$ of $Co(S)$ is a hyperplane containing $Co(S)$ in $\overline{H}_+ = \{x: a^T x \geq c\}$ and containing a boundary point of $Co(S)$ [15]. In this paper, we only consider the case that $Co(S)$ is a polyhedron and an example is shown in Fig. 6 where the shaded $Co(S)$ has five supporting hyperplanes. Let $x$ be an extreme point of $Co(S)$. The opposite polytope $p$ of $x$ for $Co(S)$ is defined to be $p = \bigcap_j \overline{H}_{j-}$ where $H_j$ is a supporting hyperplane of $Co(S)$ which also contains $x$, and the intersection is over all such $H_j$. In the following observation, we will see that $p$ is an unbounded polytope.

*Observation 2:* Let $Co(S)$ be the convex hull of a nonempty finite set $S$ and let $x$ be an extreme point of $Co(S)$. Then, the opposite polytope $p$ of $x$ for $Co(S)$ is an unbounded polytope.

*Proof:* Consider an interior point $x_1$ of $Co(S)$. Let $x' = x_1 - x$, then, for every supporting hyperplane $H_j$ of $Co(S)$ containing $x$, we have

$$a_j^T x' = a_j^T x_1 - a_j^T x$$
$$= a_j^T x_1 - c_j$$
$$> 0$$

because $x_1$ is in $\overline{H}_{j+}$. Consider a point $x_2 = x - dx'$ where $d$ is a positive real number. Then, for every $H_j$, we have

$$a_j^T x_2 = a_j^T x - da_j^T x'$$
$$= c_j - da_j^T x'$$
$$< c_j.$$

So, $x_2$ is in $\overline{H}_{j-}$. However, $p = \bigcap_j \overline{H}_{j-}$ and thus $x_2 \in p$ for any $d > 0$. This argument holds when $d \to \infty$, so $p$ is an unbounded polytope. ▽▽

*Observation 3:* Given $m$ hyperplanes in $E^n$ which partition the space into a certain number of $m$-polytopes, consider $m'$ hyperplanes of those $m$ hyperplanes which partition $E^n$ into a certain number of $m'$-polytopes. Let $q$ be an unbounded $m'$-polytope. Then, $q$ contains at least one unbounded $m$-polytope.

*Proof:* Consider a hyperplane $H$ which is one of those $m$ hyperplanes but is not one of those $m'$ hyperplanes. If either $\overline{H}_+ \cap q$ or $\overline{H}_- \cap q$ is empty, $q$ is also an unbounded $(m' + 1)$-polytope. Otherwise, either $\overline{H}_+ \cap q$ or $\overline{H}_- \cap q$ is an unbounded $(m' + 1)$-polytope and it is contained in $q$. Repeat this procedure to check all the other hyperplanes. Then, $q$ contains at least one unbounded $m$-polytope. ▽▽

*Theorem 2:* Let $S$ be a $k$-element set in $E^n$. Consider $m$ hyperplanes in $E^n$ which partition $E^n$ into $P(m, n)$ $m$-polytopes such that every element in $S$ is contained in a different $m$-polytope. Then, the number of separable elements in $S$ is no more than $P_u(m, n)$.

*Proof:* For $m \leq n$, $P_u(m, n) = P(m, n)$. Since every element in $S$ is contained in a different $m$-polytope, $k \leq P(m, n)$. Therefore, the number of separable elements in $S$ is no more than $P_u(m, n)$.

For $m > n$, $P(m, n) = P_u(m, n) + P_b(m, n)$. Let $g$ be one of the $P_b(m, n)$ bounded $m$-polytopes. For any vertex $y_i$ in $g$, consider the opposite polytope $p_i$. According to Observations 2 and 3, either $p_i$ is one of those $P_u(m, n)$ unbounded $m$-polytopes or $p_i$ will contain at least one of the $P_u(m, n)$ unbounded $m$-polytopes.
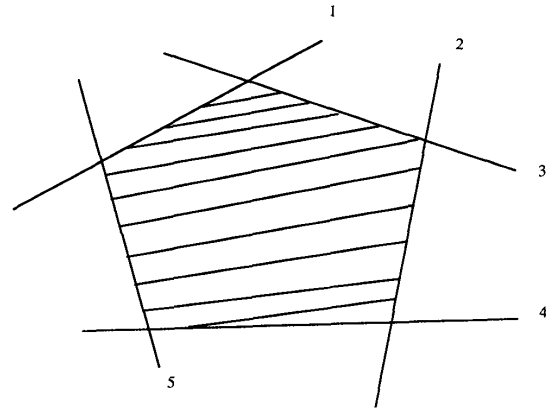


Fig. 6. A convex hull for five supporting hyperplanes.

Let $x_1$ be a separable element in $S$ which is contained in a bounded $m$-polytope $g_1$. Then, at least one of the vertices in $g_1$ is not contained in $Co(S)$ which implies that at least one of the opposite polytopes of those vertices contains no elements in $S$. Otherwise, $g_1$ will be contained in $Co(S)$ and $x_1$ will not be an extreme point in $S$. This contradicts the assumption that $x_1$ is a separable element, by Lemma 2.

The opposite polytope is unique in the sense that it is constructed by the supporting hyperplanes of a particular bounded $m$-polytope containing a particular vertex. Therefore, even for two different polyhedrons which are adjacent to each other, the opposite polytopes for the same vertex are still different. Thus, for every separable element in $S$ contained in a bounded $m$-polytope, there is at least one opposite unbounded $m$-polytope containing no elements in $S$. And every separable element in $S$ is contained either in an unbounded $m$-polytope or in a bound $m$-polytope. So the number of separable elements in $S$ is no more than $P_u(m, n)$. ▽▽

*Remark:* Consider a $k$-element set $S$ with $k_1$ separable elements. Let $m_1 = \min_m \{P(m, n) \geq k\}$ and let $m_2 = \min_m \{P_u(m, n) \geq k_1\}$. If every element in $S$ is contained in a different $m$-polytope, $m \geq \max \{m_1, m_2\}$.

## V. An Upper Bound for Binary-Valued Functions

In the literature, most of the results on capability of the MLP deal with realization of binary-valued functions, see e.g., [8], [9], [14], [21]. Such problems are basically two-category classification problems. Let the two disjoint categories be $A$ and $B$. Then, an MLP $M(\theta)$ as shown in Fig. 1 is said to be capable of constructing the classifier for $A$ and $B$ if there is a $\theta$ such that

$$M(\theta, x) > \tau, \quad \text{if } x \in A, \quad \text{and}$$
$$M(\theta, x) < \tau, \quad \text{if } x \in B, \tag{14}$$

where $\tau$ (the threshold) is a constant. Without loss of generality, $\tau$ can always be set to zero for the classifiers realized by $M(\theta)$. So, the goal here is to find a $\theta$ such that $\{x \mid M(\theta, x) > 0\} \supset A$ and $\{x \mid M(\theta, x) < 0\} \supset B$. In Fig. 1, $h = [h(1), h(2), \cdots, h(m)]$ is an $m$-tuple binary vector. As long as none of the $m$-tuple binary vectors $h$ will make $M(\theta, x) = 0$ for a given $\theta$, both $\{x \mid M(\theta, x) > 0\}$ and $\{x \mid M(\theta, x) < 0\}$ are the unions of polytopes in the input domain. As such, the sets $A$ and $B$ can be covered by unions of polytopes.

In this section, we only consider the case that $S = A \cup B$ is a $k$-element subset of $E^n$. Then, an MLP is capable of constructing a classifier for any two subsets $A$ and $B$ of $S$ if and only if the MLP is capable of constructing arbitrary dichotomies on $S$. In [8], [14], an upper bound on the number of hidden neurons for constructing arbitrary dichotomies is given for the case that $S$ is in *general position*. The upper bound is based on the following two propositions. To begin with, the notion of general position should be specified:

*Definition 4:* A $k$-element set $S$ in $E^n$ is in general position if no $(j + 1)$ elements in $S$ are in a $(j - 1)$-dimensional linear variety of $E^n$ for any $j$ such that $2 \leq j \leq n$.

A $(j - 1)$-dimensional linear variety is a hyperplane in $E^j$ [15]. It is easy to see that Definition 4 is equivalent to the one given in [8], [21] when $k \geq n + 1$, which is slightly more restrictive than the notion of general position defined in [11], [12].

*Proposition 1:* Let $S$ be a finite set of $E^n$ and let the elements of $S$ be in general position. Then, for any $k$-element subset $S_1$ of $S$, where $k \leq n$, there is a hyperplane, which is an $(n - 1)$-dimensional linear variety of $E^n$, containing $S_1$ and no other elements in $S$.

*Proposition 2:* Let $h = [h(1), h(2), \cdots, h(m)]^T$ be an $m$-tuple binary vector and let $I(j)$ be a collection of such vectors that $\sum_{i=1}^{m} h(i) = j$, where $0 \leq j \leq m$. Then, setting $w^* = [1, 1, \cdots, 1, j - 0.5]^T$ will make

$$\sum_{i=1}^{m} h(i)w^*(i) - t^* > 0, \quad \text{for } h \in \left( \bigcup_{k=j}^{m} I(k) \right)$$

and

$$\sum_{i=1}^{m} h(i)w^*(i) - t^* < 0, \quad \text{for } h \notin \left( \bigcup_{k=j}^{m} I(k) \right). \quad (15)$$

Let the number of elements in $A$ and $B$ be $k_1$ and $k_2$, respectively. Then $k_1 + k_2 = k$, as $A \cup B = S$. Without loss of generality, let $k_1 \leq k/2$. According to Proposition 1, $[\![(k_1 - 1)/n + 1]\!]$ hyperplanes can contain all those $k_1$ elements in $A$ and no elements in $B$. Note that $[\![x]\!]$ is the largest integer which is less than or equal to $x$. Consider one of the $[\![(k_1 - 1)/n + 1]\!]$ hyperplanes, say $H$, which is shifted infinitesimally into $H_+$ and into $H_-$ to become $H'$ and $H''$, respectively, such that every element contained in $H$ is in $H'_+ \cap H''_+$. And every element in $S$ which is not contained in $H$ is either in $H'_+ \cap H''_+$ or in $H'_- \cap H''_-$. Let all $[\![(k_1 - 1)/n + 1]\!]$ hyperplanes be shifted that way and let $m = 2[\![(k_1 - 1)/n + 1]\!]$ hidden neurons in an MLP construct those hyperplanes. Then, the output vector of hidden neurons for every element in $A$ belongs to $I(m/2 + 1)$ and that for every element in $B$ belongs to $I(m/2)$. According to Proposition 2, this MLP is capable of constructing the dichotomies $A$ and $B$. For a $k$-element set, the upper bound on the number of hidden neurons needed to construct arbitrary dichotomies is thus $m = 2[\![(k - 2)/2n + 1]\!]$. The derivation of this upper bound is based on the results given in [8], [14], but a slight difference has resulted from the fact that the number of hidden neurons here has to be an even integer.

One of the drawbacks for the above approach is that it is not suitable for input sets which are not in general position. In the following, input sets are decomposed into a sequence of subsets such that every subset is contained in a linear variety. With this decomposition, an upper bound is derived on the number of hidden neuron. Then, we show how to extend this approach to the case where the input sets are not in general position. Moreover, when the subsets are continually decomposed into lower

dimensional linear varieties, this upper bound will eventually approach the LUB.

*Definition 5:* Let $S$ be a finite set in $E^n$. A hyperplane $H$ which has $S$ contained entirely in either $H_+$ or in $H_-$ is called a *nil-separating* hyperplane of $S$. Any hyperplane in $E^n$ is a nil-separating hyperplane of the empty set.

*Lemma 3:* If an $n$-element set $S$ in $E^n$ is in general position as defined in Definition 4, then every subset of $S$ is a separable subset.

*Proof:* The characteristic vector of a particular hyperplane is an $(n + 1)$-tuple vector $[a^T, c]$. For a given $x_i \in S$, $a^T x_i - c = 0$ is a homogeneous hyperplane in the $(n + 1)$-dimensional weight-threshold space. $S$ is in general position, so $n$ homogeneous hyperplanes partition $E^{n+1}$ into $2^n$ unbounded $n$-polytopes as in (13) [12]. Every $[a^T, c]$ in the interior of a different $n$-polytope represents the characteristic vector of a hyperplane $H$ which separates a subset of $S$ and its complement, so that the subset is in $H_+$. Since $S$ has $2^n$ subsets, every subset of $S$ is a separable subset. ▽▽

*Remarks:*

1. According to Lemma 3 and (13), every subset of a $j$-element set in general position, $1 \leq j \leq n$, is a separable subset.

2. For a finite set $S$, there are hyperplanes which separates $S$ and its complement $\varnothing$. To distinguish those hyperplanes from a hyperplane which separates two nonempty subsets of $S$, they are called nil-separating hyperplanes in Definition 5.

*Definition 6:* If $S = S_m \supset S_{m-1} \supset \cdots \supset S_1 \supset S_0 = \varnothing$ and every $S_i$ is a separable subset of $S$, the sequence $\{S_i\}$ is said to be a *sequence of separable subsets* of $S$.

*Theorem 3:* An MLP with $[\![(k - 2)/n + 1]\!]$ hidden neurons is capable of constructing arbitrary dichotomies for a $k$-element set $S$ in $E^n$, if there is a sequence of separable subsets $\{S_i\}$ of $S$ such that 1) every $S'_i = S_i - S_{i-1}$, $1 \leq i \leq m$, is in general position and has $n$ elements except that $S'_m$ may have less than $n$ elements, and 2) $H_i$, the hyperplane containing $S'_i$ in $E^n$, is a nil-separating hyperplane of both $S_{i-1}$ and $S - S_i$.

*Proof:* According to Proposition 1, there is a hyperplane $H_i$ which contains all elements in $S'_i$. Let $S'$ and $S - S'$ be an arbitrary dichotomy of $S$. Consider the $n$ points in $H_1$ with $n'_1$ elements in $S'$ and $n - n'_1$ elements in $S - S'$. According to Lemma 3, $H_1$ can be rotated infinitesimally to separate those $n'_1$ in $H_{1+}$ and those $n - n'_1$ elements of $S'_1$ in $H_{1-}$. $H_1$ is a nil-separating hyperplane of $S - S_1$, so $S - S_1$ can be contained in $H_{1-}$.

Consider the $n$ elements in $S'_2$ with $n'_2$ elements in $S'$ and $n - n'_2$ elements in $S - S'$. As before, construct a hyperplane $H_2$ which separates those $n - n'_2$ elements of $S'_2$ in $H_{2+}$ and those $n'_2$ elements in $H_{2-}$. Because $H_2$ is a nil-separating hyperplane of $S_1$ and $S - S_2$, $S_1$ can be contained in $H_{2+}$ and $S - S_2$ can be contained in $H_{2-}$. Repeat this procedure until $H_m$, where $m = [\![(k - 1)/n + 1]\!]$.

Let $m$ hidden neurons construct $H_1, H_2, \cdots, H_m$. Then, elements in $S$ are transformed into $m$-tuple binary vectors. Let $h_j = [h_j(1), h_j(2), \cdots, h_j(m)]^T$, $1 \leq j \leq m$, be a binary vector with $h_j(i) = 0$ for $i < j$ and $h_j(i) = 1$ for $i \geq j$. In the outputs of the hidden neurons, every element in $S'$ is a binary vector $h_j$ where $j$ is odd while every element in $S - S'$ is a binary vector $h_j$ where $j$ is even. Let $w^* = [w^*(1), w^*(2), \cdots, w^*(m), t^*]^T$ be $w^*(1) = 1$ and $w^*(i) = -w^*(i - 1)$, $2 \leq i \leq m$, and $t^* = 0.5w^*(m)$. Then, the overall MLP constructs the dichotomies: $S'$ and $S - S'$.

Because $H_i$ can be chosen to have either those $n'_i$ elements or those $n - n'_i$ elements in $H_{i+}$, only $m - 1$ hidden neurons are needed for the case that $S'_m$ is a single-element set. Therefore,

an MLP with $[(k - 2)/n + 1]$ hidden neurons is capable of constructing arbitrary dichotomies for a $k$-element set in $E^n$.

$$\triangledown\triangledown$$

Consider realization of switching functions by an MLP. When $n = 2$, $S$ is a four-element set in the two-dimensional space. Let the sequence of separable subets of $S$ be $S_1 = \{(0, 0), (1, 0)\}$ and $S_2 = S$. Then, this sequence satisfies the conditions in Theorem 3, and $m = [(4 - 2)/2 + 1] = 2$. Because XOR can be realized by an MLP with two hidden neurons, $m = 2$ is also the LUB in this case. When $n = 3$, $S$ is an eight-element set. We first consider a seven-element set $S_1$ as given in Fig. 7. Then, an MLP with two hidden neurons is sufficient to realize $S_1$. Consider the hyperplane which separates $S_1$ from $S$. So, $m = 3$ is sufficient to realize $S$. When $n = 4$, $S$ is a sixteen-element set. Consider hyperplanes $x(1) + x(2) + x(3) + x(4) - 1 = 0$, $x(1) + x(2) + x(3) + x(4) - 2 = 0$, and $x(1) + x(2) + x(3) + x(4) - 3 = 0$ which contain $S_1 = \{(0, 0, 0, 1), (0, 0, 1, 0), (0, 1, 0, 0), (1, 0, 0, 0)\}$, $S_2' = \{(0, 0, 1, 1), (0, 1, 1, 0), (1, 1, 0, 0), (1, 0, 0, 1), (0, 1, 0, 1), (1, 0, 1, 0)\}$, and $S_3' = \{(1, 1, 1, 0), (1, 1, 0, 1), (1, 0, 1, 1), (0, 1, 1, 1)\}$, respectively. $S_1$ and $S_3'$ are in general position. So, $x(1) + x(2) + x(3) + x(4) - 1 = 0$ ($x(1), x(2) + x(3) + x(4) - 3 = 0$) can be rotated infinitesimally to separate the elements in $S_1$ ($S_3'$) with the same output as $(0, 0, 0, 0)$ $((1, 1, 1, 1))$ in one side. Moreover, $x(1) + x(2) + x(3) + x(4) - 2 = 0$ is a three-dimensional linear variety. And, $S_2'$ is a six-element set in the linear variety. Consider two two-dimensional linear varieties which are the intersections of $x(1) = 0$ and $x(1) + x(2) + x(3) + x(4) - 2 = 0$, and $x(1) = 1$ and $x(1) + x(2) + x(3) + x(4) - 2 = 0$. Each one of the two-dimensional linear varieties contains three elements of $S_2'$ which satisfies the conditions given in Theorem 3. Then, an MLP with two hidden neurons is sufficient to realize $S_2'$. Thus, $m = 4$ is sufficient to realize the sixteen-element set $S$. In contrast, if the results of Section II were employed, the upper bound would have been 15.

In Theorem 3, we require that every $S_i'$, instead of $S$, be in general position. The example shown in Fig. 7 illustrates how to slice an input set $S$ that is not in general position to meet the conditions. Also, Fig. 8 shows an input set $S$ in general position which does not satisfy the second condition in Theorem 3. From these two examples, we see that the general position condition, Definition 4, does not imply the two conditions specified in Theorem 3, and neither does vice versa. In the following theorem, the approach in Theorem 3 is extended such that input sets in general position also satisfy the decomposition conditions.

*Theorem 4:* Let $S_j'$ be a separable subset of $S_j = S - \bigcup_{i=0}^{j-1} S_i'$, $1 \leq j \leq d$, where $S_0' = \varnothing$. An MLP with $m = 2[(k - 2)/n + 1] - 1$ hidden neurons is capable of constructing arbitrary dichotomies for a $k$-element set $S$ in $E^n$, if there is a sequence of subsets $\{S_i\}$ of $S$ such that 1) every $S_j'$ is in general position and has $n$ elements except that $S_d'$ may have less than $n$ elements, and 2) $H_j$, the hyperplane containing $S_j'$ in $E^n$, is a nil-separating hyperplane of $S_j - S_j'$.

*Proof:* Since $S_j'$ is a separable subset of $S_j$, let $H_j'$ be the separating hyperplane of $S_j'$ and $S_j - S_j'$ such that $S_j'$ is contained in $H_{j+}'$. $H_j$ is a nil-separating hyperplane of $S_j - S_j'$, so $S_j - S_j'$ can be contained in $H_{j-}$. Let the hidden neurons construct $H_1$, $H_1'$, $H_2$, $H_2'$, $\cdots$, $H_{d-1}'$, $H_d$, where $d = [(k - 2)/n + 1]$. Rotate $H_j$, $1 \leq j \leq d$, as in Theorem 3, to separate those elements in $S_j'$ into two dichotomies. Then, elements in $S \cap H_{1+}$ have same outputs for the hidden neurons. And, elements in $S \cap H_{j+}' \cap H_{j-}$ have same outputs for the hidden neurons. Also, elements in $S \cap H_{j-}' \cap H_{(j+1)+}$ have same outputs for the
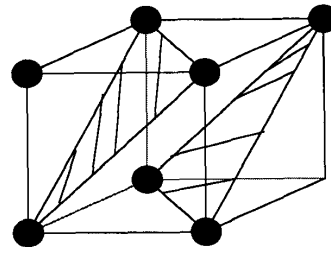


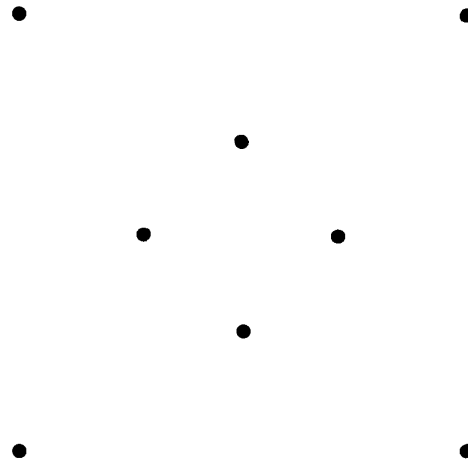Fig. 7. Two three-element subsets of a seven-element set contained in two-dimensional linear varieties.



Fig. 8. An eight-element set in general position in the two-dimensional space.

hidden neurons, $1 \leq j \leq d - 1$. If elements in $S$ with the same outputs for the hidden neurons are treated as one element, this problem will become a realization of a binary function defined on a $2d$-element set. According to Theorem 1, an MLP with $m = 2[(k - 2)/n + 1] - 1$ hidden neurons is capable of constructing arbitrary dichotomies on a $k$-element set $S$ which satisfies the conditions.

$$\triangledown\triangledown$$

*Remark:* Theorem 4 also applies when the $k$-element set $S$ is in an $n$-dimensional linear variety, because a linear variety can be shifted to be a space.

*Theorem 5:* Let $S$ be a $k$-element subset of $E^n$ in general position, where $k \geq n$. Then, there is an $n$-element separable subset $S_1$ of $S$ such that $H_1$, the hyperplane containing $S_1$ in $E^n$, is a nil-separating hyperplane of $S - S_1$.

*Proof:* Consider the convex hull $Co(S)$ which is a polyhedron in $E^n$. Let $y_1$ be a vertex of $Co(S)$ and let $H_1$ be a supporting hyperplane of $y_1$. Then, $H_1$ contains a face of $Co(S)$ which implies that there are at least $n$ vertices of $Co(S)$ contained in $H_1$. Because $S$ is in general position, $H_1$ contains no other elements in $S$ except those $n$ vertices. Let $S_1$ be the set of those $n$ vertices which is a separable subset of $S$. And, $H_1$ is a nil-separating hyperplane of $S - S_1$.

$$\triangledown\triangledown$$

The conditions on input sets in Theorem 4 are a generalized version of those in Theorem 3. According to Theorem 5, a finite set $S$ in general position also satisfies the conditions in Theorem 4. Note that the least upper bound given in Section II can not

be reduced here, even though the output of the MLP is simply binary for two-category classification. However, $2\llbracket (k-2)/n + 1 \rrbracket - 1 \leq k - 1$ for $n \geq 2$, the upper bound given in Theorem 4 is thus nontrivial.

According to Theorem 5, there is a sequence of subsets $\{ S_j \}$ of any finite set $S$ which satisfies the second condition in Theorem 4. But, $S_j'$ may contain more than $n$ elements in $S$ and the first condition is violated. However, with the fact that a hyperplane is an $(n-1)$-dimensional linear variety of $E^n$, the finite set $S$ is decomposed into $d$ subsets in $(n-1)$-dimensional linear varieties. Also, for any $S_j$, $H_j'$, the hyperplane that separates $S_j'$ and $S_j - S_j'$, isolates $S_j'$ from $S_j$. Then, $S_j'$ is a finite subset of an $(n-1)$-dimensional linear variety $H_j$ and Theorem 4 is still suitable for every decomposed subset. If a finite set $S$ is continually decomposed to one-dimensional linear varieties and none of the decomposed subsets can satisfy the first condition, the bound in Theorem 4 will be eventually equivalent to that in Theorem 1. The example given after Corollary 2 is one such case. In that case, we only need those separating hyperplanes $H_j'$ in Theorem 4 and the number of those hyperplanes is $k - 1$.

## VI. CONCLUSION

It has been seen here that an MLP with $m$ hidden neurons (Fig. 1) will partition the input space into $N$ $m$-polytopes, where $(m + 1) \leq N \leq P(m, n)$, such that elements in the same $m$-polytope have the same outputs. In realization of arbitrary functions, we showed that at least $(m + 1)$ of $N$ $m$-polytopes can be mapped into arbitrary real-valued outputs. Therefore, the MLP is capable of realizing arbitrary functions defined on an $(m + 1)$-element set. In realization of one-to-one functions, the maximum number of $m$-polytopes that can be obtained by partitioning $E^n$ with $m$ hyperplanes is $P(m, n)$. Consider a $k$-element subset $S$ with $k_1$ separable elements. Two necessary conditions that every element in $S$ is contained in a different $m$-polytope are $k \leq P(m, n)$ and $k_1 \leq P_u(m, n)$. In realization of binary-valued functions, the MLP is capable of constructing arbitrary dichotomies for an $\{ n \llbracket (m + 1)/2 \rrbracket + 1 \}$-element set which satisfies the conditions in Theorem 4 and for an $(m + 1)$-element set without any conditions.

## ACKNOWLEDGMENT

The authors wish to thank Dr. Y. Inouye at Osaka University, Department of Control Engineering, and Dr. R. Liu at the University of Notre Dame, Department of Electrical Engineering, for their valuable comments.

## REFERENCES

[1] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *PDP: Vol. 1*, chap. 8, Eds. D. E. Rumelhart, J. L. McClelland. Cambridge, MA: M.I.T. Press, 1986.
[2] N. Arai, "Mapping abilities of three-layer neural networks," in *Proc. 1989 Int. Joint Conf. Neural Networks*, Washington, DC, July 1989, pp. 1-419-423.
[3] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Mathemat. Control, Signals, Systems*, vol. 2, pp. 303-314, 1989.
[4] E. B. Baum, J. Moody, and F. Wilczek, "Internal representations for associative memory," *Biol. Cybern.*, vol. 59, pp. 217-228, 1988.
[5] F. Rosenblatt, *Principles of Neurodynamics*. Washington, DC: Spartan, 1961.
[6] S. C. Huang and Y. F. Huang, "Learning algorithms for percep-
trons using back propagation with selective updates," *IEEE Control Systems Mag.*, pp. 56-61, Apr. 1990.
[7] S. C. Huang and Y. F. Huang, "Back propagation with selective updates," in *Proc. Conf. Inform. Sci. Syst.*, pp. 584-589, Johns Hopkins University, Baltimore, MD, 1989.
[8] V. C. Soon and Y. F. Huang, "Applications and analysis of second order neural networks," in *Proc. Contr. Decis. Conf.*, Austin, TX, Dec. 1988, pp. 348-349.
[9] S. Y. Kung and J. N. Hwang, "An algebraic projection analysis for optimal hidden unit size and learning rates in back-propagation learning," in *Proc. IEEE Int. Conf. Neural Networks*, San Diego, CA, 1988, pp. 363-370.
[10] L. Schläfli, *Gesammelte Mathematische Abhandlungen* I. Basel, Switzerland: Verlag Birkhäuser, pp. 209-212, 1950.
[11] T. M. Cover, "Geometrical and statistical properties of linear inequalities with applications in pattern recognition," *IEEE Trans. Elec. Comp.*, vol. EC-14, pp. 326-334, 1965.
[12] R. O. Winder, "Partitions of n-space by hyperplanes," *J. SIAM Appl. Math.*, vol. 14, no. 4, pp. 811-818, July 1966.
[13] G. Mirchandani and W. Cao, "On hidden nodes for neural nets," *IEEE Trans. Circ. Syst.*, vol. 36, pp. 661-664, May 1989.
[14] E. B. Baum, "On the capabilities of multilayer perceptrons," *J. Complexity*, vol. 4, pp. 193-215, 1988.
[15] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Second Edition. Reading, MA: Addison-Wesley, 1984.
[16] V. C. Soon and Y. F. Huang, "Artificial neural networks with second order discriminant functions," in *Proc. 1988 Conf. Inform. Sci. and Syst.*, pp. 308-313, Princeton University, Princeton, NJ, Mar. 1988.
[17] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.
[18] R. L. Kashyap, "Synthesis of switching functions by threshold elements," *IEEE Trans. Electron. Comput.*, vol. EC-15, no. 4, pp. 619-628, Aug. 1966.
[19] J. S. Jang, S. Y. Lee, and S. Y. Shin, "An optimization network for matrix inversion," *IEEE Conf. Neural Inform. Process. Syst.-Natural and Synthetic*, p. 55, Denver, CO, 1987.
[20] V. Cherkassky and N. Vassilas, "Performance of back propagation networks for associative database retrieval," in *Proc. 1989 Int. Joint Conf. Neural Networks*, pp. I-77-84, Washington, DC, July 1989.
[21] S. Olafsson and Y. S. Abu-Mostafa, "The capacity of multilevel threshold functions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 10, no. 2, pp. 277-281, March 1988.

**Shih-Chi Huang** received the B.S. degree from the National Chiao-Tung University, Hsing-Chu, Taiwan, in 1983 and the M.S. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1988. He is currently a Ph.D. candidate in the Department of Electrical Engineering at the University of Notre Dame.

His research interests are in neural networks, pattern recognition, and signal processing.

**Yih-Fang Huang** (S'80-M'82) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1976; the M.S.E.E. degree from the University of Notre Dame, Notre Dame, IN, in 1979; and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1982.

He has been on the Faculty of Electrical and Computer Engineering at University of Notre Dame since September 1982, and is currently Associate Professor of Electrical Engineering. His research interests are in the areas of statistical signal processing and artificial neural networks.

He is currently Associate Editor on Neural Networks and Signal Processing for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS.