A Two Layer Paradigm Capable of Forming Arbitrary Decision Regions in Input Space *

VINAY DEOLALIKAR
Information Theory Research Group
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304
Email: vinayd@exch.hpl.hp.com
Tel: (650) 857 8605 Fax: (650) 852 3791

Abstract

It is well known that a 2-layer perceptron network with threshold neurons is incapable of forming arbitrary decision regions in input space, while a 3-layer perceptron has that capability. In this paper, the effect of replacing the output neuron in a 2-layer perceptron by a bithreshold element is studied. The limitations of this modified 2-layer perceptron are observed. Results on the separating capabilities of a pair of parallel hyperplanes are obtained. Based on these, a new 2-layer neural paradigm based on increasing the dimensionality of the output of the first layer is proposed and is shown to be capable of forming any arbitrary decision region in input space. Then a type of logic called bithreshold logic, based on the bithreshold neuron transfer function, is studied. Results on the limits of switching function realizability using bithreshold gates are obtained.

Keywords

Artificial Neural Networks, Classification Regions, 2-layer Networks, Bithreshold Neuron, Bithreshold Logic.

I. Introduction

Multi Layer Perceptrons (MLPs) have demonstrated very promising performance as compared to classical Von Neumann machines in several areas like function approximation, pattern recognition, speech recognition, *etc.* An excellent concise introduction to this field can be found in [7]. More classical treatises on the foundations of this subject are [9, 10, 12].

The classical MLP is made up of layers of neurons. Each neuron has a pair $(\mathbf{w}, t)$ associated to it, where $\mathbf{w}$ and t are called its weight vector and threshold, respectively. Let $\mathbf{x}$ be the input vector to a neuron, of the same dimension as $\mathbf{w}$. Then the output of the neuron is defined by

$$f_{\mathbf{w},t}(\mathbf{x}) = \begin{cases} +1, & \text{if } \mathbf{w}.\mathbf{x} > t \\ -1, & \text{if } \mathbf{w}.\mathbf{x} \le t. \end{cases}$$

The quantity $\mathbf{w}.\mathbf{x}$ is called the *activation* of the neuron. Each neuron in a layer receives as input the outputs of all the neurons in the previous layer and it feeds its output to every neuron in the next layer and so on. There are no interconnections between neurons within a layer. The output layer consists of only one neuron, called the *output neuron*.

The network as a whole performs a "classification" of $\mathbf{R}^n$ by mapping every vector in $\mathbf{R}^n$ to a +1 or a -1. The subset of $\mathbf{R}^n$ that is mapped to +1 is called the network's *decision region.* In general, a subset of $\mathbf{R}^n$ is said to be *classifiable* by a network if it can be made the decision region of the network (usually by appropriately changing the weight vectors).

It is known that a MLP with 3-layers is a *universal classifier*, *i.e.*, it can form any arbitrary decision region in input space. Many studies on the limitations of a 2-layer network in this regard have been done. We shall first study the 2-layer network in some detail.

Let there be m neurons with a fixed ordering in the first layer operating on a set of n-dimensional inputs. Geometrically, each of the neurons has a hyperplane associated with it given by

$$H_{\mathbf{w},t} = \{\mathbf{x} : \mathbf{w}.\mathbf{x} = t\}.$$

This hyperplane then divides $\mathbf{R}^n$ into two halfspaces

$$H^+_{\mathbf{w},t} = \{\mathbf{x} : \mathbf{w}.\mathbf{x} > t\} \text{ and } H^-_{\mathbf{w},t} = \{\mathbf{x} : \mathbf{w}.\mathbf{x} \leq t\},$$

also called its positive and negative halfspace, respectively. For the rest of the paper, we will continue to use the above notation to denote hyperplanes and their positive and negative halfspaces. The m such hyperplanes corresponding to the m first layer neurons exhaustively divide $\mathbf{R}^n$ into disjoint *basic convex polytopes.* Let the pair $\{\mathbf{w}_j, t_j\}$ be associated with the j^{th} neuron in the first layer. Then each basic convex polytope is given by an intersection of m halfspaces as

$$C_l = \bigcap_{j=1}^{m} H^{i_j}_{\mathbf{w_j},t_j},\ 1 \leq l \leq 2^m,$$

where $i_j = \pm$ depending on l and j. The index l is bounded by 2^m because m hyperplanes can divide $\mathbf{R}^n$ into at most 2^m regions. Each basic convex polytope C_l has a unique (for the ordering of neurons) m-dimensional Boolean representation given by $\mathbf{c}_l = (c_{l1}, \ldots, c_{lm})$ where $c_{lj} = +1$ if $i_j = +$, and $c_{lj} = -1$ if $i_j = -$.

We can view the outputs of the first layer neurons collectively as a m-dimensional vector. Then, $\mathbf{c}_l$ forms the output vector of the first layer neurons when an input vector falls in the region C_l. Let Q^m denote the set of vertices of the m-cube $\{-1, 1\}^m$. Since $\mathbf{c}_l \in Q^m$, there is a 1-1 (but not necessarily onto) mapping from the set of basic convex polytopes $\{C_l\}$ to Q^m given by $g : C_l \mapsto \mathbf{c}_l$. In effect, it is this mapping that is performed by the first layer of the network. In general, the image of this map, $im(g)$, is not all of Q^m. The vertices of the m-cube that lie in $im(g)$ are then the possible inputs to the output neuron.

Now the output neuron also has a unique hyperplane associated with it given by

$$H_{\mathbf{w_o},t_o} = \{\mathbf{c} : \mathbf{w_o}.\mathbf{y} = t_o\},$$

where $\mathbf{y}$ is a generic m-dimensional vector. This hyperplane divides Q^m into two sets of vertices, one falling in $H^+_{\mathbf{w_o},t_o}$ and the other in $H^-_{\mathbf{w_o},t_o}$. Thus, if the output neuron receives as its input a vertex $\mathbf{v}$ in $H^+_{\mathbf{w_o},t_o}$, the network outputs +1. If, on the other hand, the output neuron receives as its input a vertex $\mathbf{v}$ in $H^-_{\mathbf{w_o},t_o}$, the network outputs -1. The network's decision region is given by the union of all the basic convex polytopes whose Boolean representations fall in $H^+_{\mathbf{w_o},t_o}$.

From this discussion, it is clear that a certain decision region is implementable by a 2-layer network if the vertices of Q^m corresponding to the individual convex polytopes that comprise this region are *linearly separable* from their complement in $im(g)$.

Definition 1 *Let Q^m denote the set of vertices of a m-cube. A dichotomy $\{Q^{m+}, Q^{m-}\}$ of Q^m is said to be linearly separable if there exists a pair $\{\mathbf{w}, t\}$ such that*

$$Q^{m+} \subset H^+_{\mathbf{w},t} \text{ and } Q^{m-} \subset H^-_{\mathbf{w},t}.$$

The problem of ascertaining whether or not an arbitrary dichotomy of Q^m is linearly separable is known to be hard [2]. This implies that the problem of determining whether or not an arbitrary decision region is implementable by a 2-layer network is hard as well.

II. Available results on 2 and 3-layer classifiability

At first, it was thought that the decision regions of 2-layer perceptrons could only be convex polytopes. Later, nonconvex decision regions were shown to be 2-layer classifiable, but the condition of connectedness was added [6]. However, it was demonstrated later that even unions of disconnected convex regions could be 2-layer classifiable [8]. Subsequently, CoRD (Convex Recursive Deletion) regions have been shown to be 2-layer classifiable [13]. The reader should not, however, believe that there is any chance of traditional 2-layer networks having the capability to form arbitrary decision regions in input space, as is equivalent to saying that every dichotomy of Q^m is linearly separable. We can find many counterexamples - for $m > 1$, a simple one being the dichotomy $\{Q^{m+}, Q^{m-}\}$ with Q^{m+} comprising just a pair of antipodal points. However, it is easily shown that any arbitrary decision region in input space is 3-layer classifiable [7]. It is perhaps this result that has led to a somewhat diminished interest in the problem of 2-layer classifiability. There is, however, a result by Cybenko [4] that proves that 2-layers are sufficient to *approximate* arbitrary decision regions in input space.

III. The bithreshold neuron model

While threshold models for neurons are most widely used in existing literature, there has also been some effort devoted to studying multi-threshold neuron models. For example, in [11], expressions for the separating capacity of a multi-threshold gate acting upon several points which are assumed to be in general position have been derived. For our purposes, these results will not be very useful since the points that we seek to separate are not in general position, but are vertices of Q^m. We will focus our attention on the simplest case of a multi-threshold neuron, namely, the bithreshold neuron (BN). A BN is defined by a triple $(\mathbf{w}, t_1, t_2)$, where $t_1 < t_2$ with its output $f_{\mathbf{w},t_1,t_2}$ given by (see also Figure 1)

$$f_{\mathbf{w},t_1,t_2} = \begin{cases} +1, & \text{if } t_1 < \mathbf{w}.\mathbf{x} \leq t_2, \\ -1, & \text{otherwise.} \end{cases}$$

Geometrically, the bithreshold neuron has two separating surfaces that define its decision region, as opposed to just one separating surface that defined the decision region of the traditional threshold neuron. These are in the form of two parallel hyperplanes given by

$$H_{\mathbf{w},t_1} = \{\mathbf{x} : \mathbf{w}.\mathbf{x} = t_1\} \text{ and } H_{\mathbf{w},t_2} = \{\mathbf{x} : \mathbf{w}.\mathbf{x} = t_2\}.$$

The decision region is the intersection of the positive halfspace $H^+_{\mathbf{w},t_1}$ of the first hyperplane corresponding to the lower threshold limit and the negative halfspace $H^-_{\mathbf{w},t_2}$ of the second hyperplane corresponding to the upper threshold limit. We denote this decision region by $P_{\mathbf{w},t_1,t_2}$ where

$$P_{\mathbf{w},t_1,t_2} = H^+_{\mathbf{w},t_1} \cap H^-_{\mathbf{w},t_2}.$$

We denote the complement of the decision region by

$$N_{\mathbf{w},t_1,t_2} = H^-_{\mathbf{w},t_1} \cup H^+_{\mathbf{w},t_2}.$$

Definition 2 *Let $\{Q^{m+}, Q^{m-}\}$ be a dichotomy of Q^m. It is said to be P-separable if there exists a triple $\{\mathbf{w}, t_1, t_2\}$ such that $Q^{m+} \subset P_{\mathbf{w},t_1,t_2}$ and $Q^{m-} \subset N_{\mathbf{w},t_1,t_2}$. It is said to be N-separable if $Q^{m+} \subset N_{\mathbf{w},t_1,t_2}$ and $Q^{m-} \subset P_{\mathbf{w},t_1,t_2}$. A dichotomy of Q^m is said to be bithreshold-separable if it is either P-separable or N-separable (or both).*

Proposition 1 *P-separability does not imply N-separability and vice versa.*

Proof. Let Q^{3+} be the set of two diagonally opposite points of the same face and Q^{3-} be its complement. Then, the dichotomy $\{Q^{3+}, Q^{3-}\}$ is P-separable but not N-separable, while the dichotomy $\{Q^{3-}, Q^{3+}\}$ is N-separable but not P-separable. $\square$

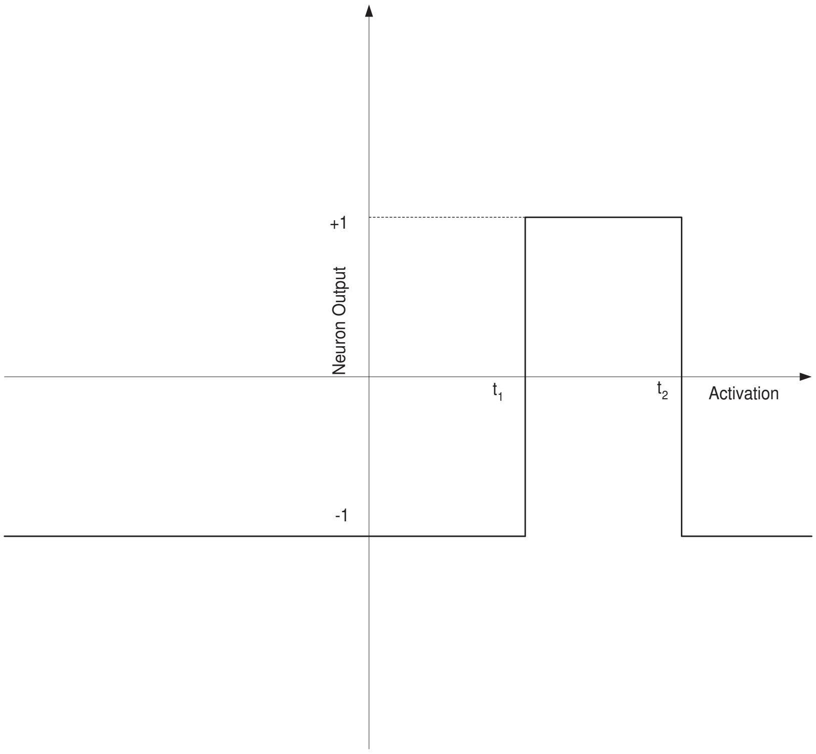


Figure 1: The transfer function for the bithreshold neuron model. This type of characteristic can be obtained by tying the outputs of two open-collector amplifiers, one of which compares the input with t_1 and the other with t_2.

IV. The Capabilities of a 2-layer Perceptron with a BN Output

We now study the capabilities of a 2-layer perceptron in which the output neuron has been replaced by a bithreshold neuron. The rest of the network remains as earlier. Such a network will be said to have a BN output and will be referred to as a *modified 2-layer perceptron.*

The basic theorem which we use to tackle the problem of separating sets of vertices using two hyperplanes is stated below. Even though we only need the use of this theorem where S and S' are known to be subsets of Q^m, we will prove it for the more general case where they are arbitrary finite subsets of $\mathbf{R}^m$. In what follows, let $C(S)$ denote the convex polytope defined by points in S and $\mathbf{A}(S)$ denote the affine subspace defined by the points of S.

Theorem 1 *Let S and S' be finite subsets of $\mathbf{R}^m$. If $|S| \leq m-1$, and $S' \cap C(S) = \phi$, there exists a bithreshold neuron defined by a triple $\{\mathbf{w}, t_1, t_2\}$ such that $S \subset P_{\mathbf{w},t_1,t_2}$ and $S' \subset N_{\mathbf{w},t_1,t_2}$.*

Proof. Firstly, note that if $|S| < m-1$, we could add points to S that are arbitrarily close to those already in S to make $|S| = m-1$. Thus, without loss of generality, we assume that $|S| = m-1$. Let $|S'| = k$. There are k hyperplanes uniquely defined by the k sets of m points formed by adding only one of the k points in S' to the m-1 points in S. Take a point p in $\mathbf{R}^m$ not lying on any of these k hyperplanes. Now along with the m-1 points in S it forms a set of m points. Let the hyperplane uniquely defined in $\mathbf{R}^m$ by these m points be $H_{\mathbf{a},b}$. This hyperplane then does not pass through any of the k points of S', if they do not lie in $\mathbf{A}(S)$. Then two hyperplanes given by $H_{\mathbf{a},b-\epsilon}$ and $H_{\mathbf{a},b+\epsilon}$, where ϵ is less than the perpendicular distance from $H_{\mathbf{a},b}$ of the nearest point in S', perform the desired partition. All the m-1 points of S lie in $P_{\mathbf{a},b+\epsilon,b-\epsilon,}$ while all the k points in S' lie in $N_{\mathbf{a},b+\epsilon,b-\epsilon,}$.

If any of the points belonging to S' lie in $\mathbf{A}(S)$, then $H_{\mathbf{a},b}$ will pass through them also. In that case, we can perform the above procedure with S replaced by a set S_1 that contains $m-2$ points from S and one more point not on $\mathbf{A}(S)$. We ensure that the distance from $\mathbf{A}(S_1)$ of the point of S left out is less than that of any point in S'. This is always possible since $S' \cap C(S) = \phi$. This completes the proof. □

Lemma 1 *Let $\mathbf{v} \in Q^m$ and $V \subset Q^m$ with $\mathbf{v} \notin V$. Then $\mathbf{v} \notin C(V)$.*

Proof. Since this is a geometric statement, we can relabel the vertices in Q^m such that $\mathbf{v} = (1, \ldots, 1)$. Clearly, $C(V) \subseteq C(Q^m \setminus \mathbf{v})$. Now consider the hyperplane $H_{\mathbf{v},m-\epsilon}$ where $0 < \epsilon < 2$. Then clearly, $\mathbf{v} \in H^+_{\mathbf{v},m-\epsilon}$ and the entire set of vertices $Q^m \setminus \mathbf{v}$ lies in $H^-_{\mathbf{v},m-\epsilon}$, and therefore so do $C(Q^m \setminus \mathbf{v})$ and $C(V)$. Thus the hyperplane $H_{\mathbf{v},m-\epsilon}$ has separated $\mathbf{v}$ from $C(V)$. By the Hahn-Banach Theorem [1], the vertex $\mathbf{v}$ cannot lie in $C(V)$. □

Theorem 2 *The modified 2-layer perceptron can implement any decision region in $\mathbf{R}^n$ that is formed by the union of $\leq (m-1)$ basic convex polytopes, where m is the number of neurons in the first layer.*

Proof. Follows immediately from Theorem 1 and Lemma 1 by letting Q^{m+} comprise the vertices of Q^m corresponding to the stated basic convex polytopes and Q^{m-} be their complement. □

Corollary 1 *The modified 2-layer perceptron can classify any decision region in $\mathbf{R}^n$ that is formed by the union of $\geq 2^m - (m-1)$ basic convex polytopes in $\mathbf{R}^n$, where m is the number of neurons in the first layer.*

Proposition 2 *A linearly separable dichotomy of Q^m is always N and P-separable.*

Proof. Let $\{\mathbf{w}, t_1, t_2\}$ be the triple associated with the BN. The result follows by making $t_1 - t_2 > 2m$ and performing partitions of Q^m with just one of the two separating hyperplanes. This also implies that decision regions implementable by a standard 2-layer network are always implementable by a 2-layer network with a BN output. Moreover, there are bithreshold-separable dichotomies of Q^m that are not linearly separable, resulting in decision regions that can be implemented by a modified 2-layer perceptron but not by a traditional 2-layer perceptron (see also Figure 2). □

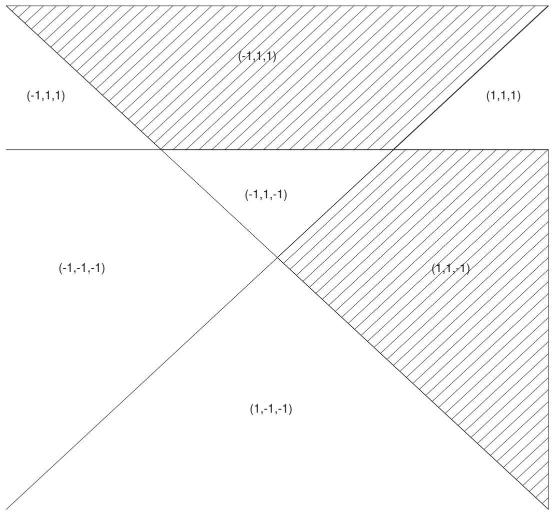


Figure 2: Example of a decision region in 2-dimensional input space implementable by a 2-layer network with 3 first layer neurons and a bithreshold neuron output (without further addition of first layer neurons to the existing network) that is not implementable by a 2-layer perceptron with 3 first layer neurons with threshold output. The vertices of the 3-cube that correspond to the basic convex polytopes are also indicated in the figure.

Lemma 2 *Let S be a set of points lying on a hyperplane $H_{\mathbf{a},b}$. Let S' be a set of points such that $S' \cap H_{\mathbf{a},b} = \phi$. Then $\{S, S'\}$ is bithreshold-separable.*

Proof. The set of parallel hyperplanes $H_{\mathbf{a},b-\epsilon}$ and $H_{\mathbf{a},b+\epsilon}$ will perform the required separation by setting $\epsilon < d$ where d is the perpendicular distance from $H_{\mathbf{a},b}$ of the closest point in S'. $\square$

Proposition 3 *For $m = 1, 2$, all dichotomies of Q^m are bithreshold-separable. For all $m \geq 3$, there exist dichotomies of Q^m that are not bithreshold-separable.*

Proof. The cases $m = 1, 2$ are trivial. Consider $m = 3$. For a particular labelling of the vertices, the dichotomy $\{Q^{m+}, Q^{m-}\}$ where $Q^{m+} = \{(-1,-1,-1),(-1,1,1),(1,-1,1),(1,1,-1)\}$ and Q^{m-} is its complement, is not bithrehold-separable. There are in all four such dichotomies corresponding to the eight different ways of labelling the cube and equating the dichotomies obtained by labelling w.r.t. a vertex and its antipodal vertex. To see this, observe that the m-cube has a natural "layering" of its vertices, such that the i^{th} layer, $0 \leq i \leq m$, comprises those vertices which have exactly i -1's in their coordinates. A dichotomy that is not bithreshold-separable for $m \geq 3$ is obtained by letting Q^{m+} be the union of layers having even parity, and Q^{m-} be the union of layers having odd parity (see also Figure 3(a)). $\square$

V. A 2-layer paradigm capable of forming arbitrary decision regions

A Theoretical Framework

Consider a dichotomy $\{Q^{m+}, Q^{m-}\}$ of Q^m. By Theorem 1, we know that if $|Q^{m+}| \leq (m-1)$ or $|Q^{m+}| \geq 2^m - m + 1$, the dichotomy is bithreshold-separable. For $m - 1 < |Q^{m+}| < 2^m - m + 1$, we cannot guarantee the bithreshold-separability of $\{Q^{m+}, Q^{m-}\}$. Let $|Q^{m+}| = p$, with $m - 1 < p < 2^m - m + 1$. Let $\mathbf{v} = (v_1, \ldots, v_m)$ be a generic vertex of Q^m. We now define a map

$$\begin{array}{rcl} e_k^m : Q^m & \hookrightarrow & Q^{m+1} \\ \mathbf{v} & \mapsto & \mathbf{v}' = (v_1, \ldots, v_m, v_k). \end{array}$$

Here v_i is the k^{th} component $(1 \leq k \leq m)$ of the vertex $\mathbf{v}$. Thus, the map e_k^m appends to each of the vertices in Q^{m+} its own k^{th} component mapping it to a vertex in Q^{m+1}.

Now consider the sequence of maps

$$Q^m \overset{e_k^m}{\hookrightarrow} Q^{m+1} \overset{e_k^{m+1}}{\hookrightarrow} \ldots \overset{e_k^p}{\hookrightarrow} Q^{p+1}.$$

The image of the set Q^{m+} under this sequence of maps is a set of vertices of Q^{p+1} of cardinality p. But by Theorem 1, this set is bithreshold-separable from its complement in Q^{p+1}. This is the theoretical framework that leads to our neural architecture.

B The New Network Paradigm

Consider the 2-layer modified MLP with a BN as its output neuron. Let there be m neurons in its first layer, each of them receiving n-dimensional inputs. The hyperplanes corresponding to the m first layer neurons form $C(m,n)$ basic convex polytopes in $\mathbf{R}^n$, where $C(m,n)$ is given by [16]

$$C(m,n) = \sum_{i=0}^{min(m,n)} \binom{m}{i} = \begin{cases} 2^m, & m \leq n, \\ \sum_{i=0}^{n} \binom{m}{i}, & m > n. \end{cases}$$

Then consider a decision region comprising a union of p basic convex polytopes. If $p \leq m-1$ or $p \geq 2^m - m + 1$, it can be realized by the 2-layer modified MLP by force of Theorem 1.

If $m - 1 < p < 2^m - m + 1$, to the first layer of m neurons, add a neuron defined by the same pair $\{\mathbf{w}, t\}$ as the j^{th} existing neuron of the first layer. Thus now, we have $m+1$ neurons in the first layer. The added neuron's output is identical to the j^{th} existing neuron's output.

Geometrically, it means that the hyperplane separating surface of the added neuron exactly coincides with the hyperplane separating surface of the j^{th} existing neuron. Thus, by addition of this neuron, *no new* regions are formed in the input space. The input space partitioning into basic convex polytopes remains exactly the same as before. In particular, the required decision region still comprises the union of only p basic convex polytopes.

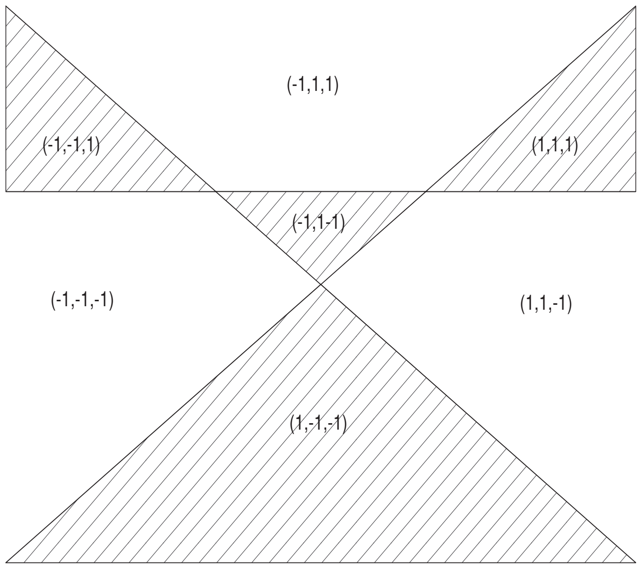
But now, the output BN receives as input the vertices of Q^{m+1}, instead of Q^m as earlier. From these vertices, the BN is required to separate p vertices corresponding to the p convex polytopes whose union is our desired decision region.

Repeat this addition of neurons. With each successive addition, the output BN is required to separate out p vertices from a higher dimensional cube. In particular, after $p - m + 1$ neurons have been added to the first layer, the output BN is required to separate out p vertices in Q^{p+1} from their complement. This it can accomplish, by force of Theorem 1. Thus, we will have to add at most $p - m + 1$ neurons to the first layer to implement a decision region that is the union of p basic convex polytopes in $\mathbf{R}^n$. This paradigm is illustrated for the case of $n = 2, m = 3, p = 4$ in Figure 3.

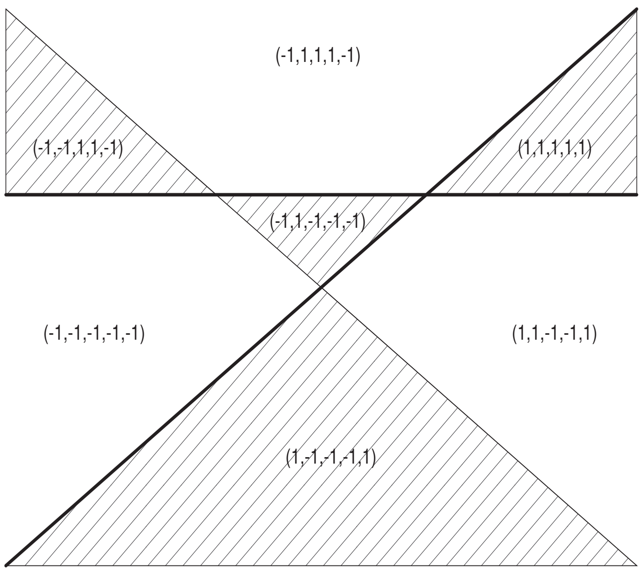
C Upper bound on the number of first layer neurons

Let m be the minimum number of hyperplanes required to obtain the p basic convex polytopes whose union is the desired decision region. Then these m hyperplanes form $C(m,n)$ regions which correspond to $C(m,n)$ vertices of the m-cube. It is these vertices only that will be possible inputs to the output BN. From these $C(m,n)$ it will have to separate out p vertices from all the others.

In the worst case, $p = C(m,n)/2$, for if $p > C(m,n)/2$ we can take the complement of our decision region and separate out the $C(m,n) - p$ vertices corresponding to this union of complementary basic convex polytopes. Then we have to have at most $p + 1 = C(m,n)/2 + 1$ neurons in the first layer. Thus, the number of neurons in the first layer is bounded by $C(m,n)/2 + 1$ where m is the minimum number of hyperplanes required to form the basic convex polytopes whose union is our desired decision region. Note that the result holds for any arbitrary choice of basic convex polytopes.



(a)



(b)

Figure 3: A demonstration of the new network architecture. (**a**) A decision region not implementable with a 2-layer network by just replacing the output neuron with a BN. This network has 3 first layer neurons. (**b**) Add two neurons to the first layer whose decision regions coincide with the first and third existing neurons (signified by the two darker lines). The partitions of input space remain the same, and the new vertices of the 5-cube that correspond to the desired decision region are separable from their complementary set by force of Theorem 1.

VI. Bithreshold Logic

In this section we shall examine whether the logic that can be implemented by bithreshold gates - bithreshold logic - offers any considerable advantages over threshold logic. To this end, we will estimate the number of Boolean switching functions that are implementable with a single BN.

There is a well known upper bound on the number of implementable threshold Boolean switching functions of m Boolean variables defined on r points ($r \leq 2^m$) given by [15]

$$B_r^m \leq 2 \sum_{i=0}^{m-1} \binom{r-1}{i} .$$

Similarly, we seek an estimate of the number of Boolean switching functions of m variables defined on r points and realizable by a single bithreshold gate. By comparing the two, we can get an idea of the enhancement in switching function realization capacity offered by the bithreshold gate.

Let the r points in Q^m on which the switching function is defined be $\mathbf{u}_1, \ldots, \mathbf{u}_r$. Transform each point $\mathbf{u}$ in $\mathbf{R}^m$ to a point in $\mathbf{R}^{m+1}$ given by $\mathbf{u}' = (\mathbf{u}, -1)$. Now consider the hyperplane in $\mathbf{R}^{m+1}$ given by

$$H_{\mathbf{u}'_i,0} = \{(\mathbf{x}, t) : \mathbf{x}.\mathbf{u}_i - t = 0\},$$

where the m weights plus the threshold make up the $m + 1$ dimensions.

Then in $\mathbf{R}^{m+1}$, points lying in the positive halfspace $H^{+}_{\mathbf{u}'_i,0}$ of this hyperplane represent values of $\mathbf{w}$ and t that would make the threshold function at $\mathbf{u}_i$ negative and points in the negative halfspace represent values of $\mathbf{w}$ and t that would make the threshold function at $\mathbf{u}_i$ positive. Each of the r points gives a similar hyperplane. To calculate the number of threshold functions all we have to do is to count the regions into which the r hyperplanes divide the whole of $\mathbf{R}^{m+1}$ [16].

The number of bithreshold functions is arrived at in a slightly different manner. Two points in different regions correspond to threshold functions differing on at least one of the r points. Consider a point in any of the regions. This corresponds to a threshold function. We can also view it as a hyperplane in $\mathbf{R}^m$. Now, consider a point in another region. This also corresponds to a hyperplane in $\mathbf{R}^m$ which differs from the first on at least one of the r vertices of the cube in the sense that at least one of the r vertices is not on the same side of both of these hyperplanes.

Now if these two points had differed only in their t coordinate, then the two hyperplanes in $\mathbf{R}^m$ corresponding to them would have been parallel, *i.e.*, together they would have represented a bithreshold function. Thus, directed line segments in $\mathbf{R}^{m+1}$ parallel to the t axis represent bithreshold functions. Two directed line segments represent the same bithreshold functions iff the two segments begin in the same region and also end in the same region. We seek to estimate the number of such directed line segments.

We now make the following estimate. If there are B_r^m such regions in $m + 1$ dimensions and they are uniformly distributed, roughly ${B_r^m}^{\frac{1}{m+1}}$ will be "stacked up" in any one coordinate. Thus for each of the B_r^m regions, roughly ${B_r^m}^{\frac{1}{m+1}}$ can be reached by just changing the t coordinate. So in all we have roughly ${B_r^m}^{\frac{m+2}{m+1}}$ of such directed line segments parallel to the t axis.

Thus, the number of different switching functions of m variables defined on r points and implementable by a single bithreshold gate is $\sim {B_r^m}^{\frac{m+2}{m+1}}$ as compared to B_r^m by a single threshold gate.

Proposition 4 *If $r > 3m$, single bithreshold gate realizability is unlikely.*

Proof. The proof is similar to Winder's [15] proof of the result for a single threshold gate. We know that

$$B_r^m < \frac{2r^m}{m!} < r^m.$$

Also, the total number of switching functions on r points is 2^r. So we can estimate the ratio

$$\begin{aligned} S &= \left[\tfrac{2r^m}{m!}\right]^{\frac{m+2}{m+1}} 2^{-r} \\ \text{(Using Stirling's approximation)} &= \left[\frac{2r^m}{\sqrt{2\pi m}(\frac{m}{e})^m}\right]^{\frac{m+2}{m+1}} 2^{-r} \end{aligned}$$

Taking log to the base 2 and letting $\alpha = r/m$,

$$\begin{aligned}\log S &= -\alpha m + \tfrac{m+2}{m+1}\left[1 + m\log\alpha e - \tfrac{\log 2\pi m}{2}\right] \\ &= m[-\alpha + \tfrac{m+2}{m+1}\log\alpha e + \tfrac{m+2}{m(m+1)}[1 - \tfrac{\log 2\pi m}{2}]]\end{aligned}$$

as $m \longrightarrow \infty$,

$$S = 2\{exp[-\alpha + \log\alpha e]\}^m$$

which goes to zero for $\alpha > 3$. □

Proposition 5 *If a switching function is randomly defined on r randomly chosen points, then as $m \longrightarrow \infty$, the probability of the function being realizable by a single bithreshold gate $\longrightarrow 1$ for $r < 2m$; 0 for $r > 2m$; and $1/2$ for $r = 2m$.*

Proof. We seek to show that

$$B_r^{m\,\frac{m+2}{m+1}}/2^r \stackrel{m\to\infty}{\longrightarrow} 1 \;\text{ for } \alpha < 2,$$

$$B_r^{m\,\frac{m+2}{m+1}}/2^r \stackrel{m\to\infty}{\longrightarrow} 0 \;\text{ for } \alpha > 2,$$

$$B_r^{m\,\frac{m+2}{m+1}}/2^r \stackrel{m\to\infty}{\longrightarrow} 1/2 \;\text{ for } \alpha = 2.$$

Firstly, observe that $\frac{m+2}{m+1}$ goes to 1 as $m \longrightarrow \infty$. Thus the ratios we seek to evaluate as $m \longrightarrow \infty$ are the same as the ratios of $B_r^m/2^r$ for the limits in question. But these are just the corresponding ratios for single threshold gate realizability, and so the result we seek to prove for bithreshold gate realizability is the same as the existing ones for threshold gate realizability [15]. □

This result tells us that the capabilities of a bithreshold gate are asymptotically the same as a threshold gate. This may seem slightly surprising at first, but is actually not so in light of Cover's [3] results on the separating capacities of surfaces. Cover showed that the natural separating capacity of a surface with m degrees of freedom is $2m$. If however, there are k independent constraints on the surface, its separating capacity reduces to $2m - k$.

We may view a pair of hyperplanes in $\mathbf{R}^m$ as a single surface with $2m$ degrees of freedom. However, if we insist that the hyperplanes be parallel, then after we have fixed the first hyperplane, we have only one degree of freedom left for the second hyperplane - its distance from the first. This leads to a total of $m + 1$ degrees of freedom, which $\approx m$ for large m. The result can easily be extended to multi-threshold gates as well.

However, for practical applications with smaller number of inputs, the bithreshold gate provides a significantly improved capability. Perhaps more importantly, it allows us to separate certain geometric structures of the hypercube, like its major diagonals, which could not be separated with a threshold gate.

VII. Conclusion

In this paper, we have shown that a bithreshold neuron, when used as the output neuron of a 2-layer network, significantly improves its classification capability. We provided a new paradigm for a 2-layer network based on increasing the dimensionality of the input to the output neuron. In most neural learning paradigms, the number of neurons in the various layers and their interconnections remain fixed while the weights vary. In our paradigm, we vary the number of neurons in the first layer as well. This paradigm can achieve universal classification capability for a 2-layer network. We also studied the realizability of Boolean functions using bithreshold gates and showed that asymptotically, threshold and bithreshold gates have the same capacity.

Acknowledgements

I thank P. G. Poonacha for many interesting conversations and P. C. Sharma for all his help in making the final manuscript of this paper.

References

[1] A. Balakrishnan, *Applied functional analysis*, New York: Springer-Verlag, 1976.

[2] A. Blum and R. Rivest, "Training a 3-node network is NP complete," *Neural Networks*, vol. 5, pp. 117-127, 1992.

[3] T. Cover, "Geometric and statistical properties of systems of linear inequalities with applications in pattern recognition," *IEEE Transactions on Electronic Computers*, vol. EC-14, pp. 326-334, 1965.

[4] G. Cybenko, "Approximation by superpositions of a sigmoidal function," Unpublished manuscript, October 1988.

[5] V. Deolalikar, *New approaches to learning and classification in feedforward neural networks*, M. Tech. Dissertation, Department of Electrical Engineering, I.I.T. Bombay, July 1994.

[6] G. Gibson and C. Cowan, "On the decision regions of multilayer perceptrons," *Proceedings of the IEEE*, vol. 78, No. 10, pp. 1590-1594, October 1990.

[7] R. Lippman, "An introduction to computing with neural nets," *IEEE ASSP magazine*, pp. 4-22, April 1987.

[8] J. Makhoul and A. Jeroudi, "Formation of disconnected decision regions with a single hidden layer," *Proc. Int. Joint Conf. on Neural Networks*, vol. 1, pp. 455-460, 1989.

[9] W. McCulloch and W. Pitts, "A logical calculus of the ideas imminent in nervous activity," *Bulletin of Mathematical Biophysics*, vol. 5, pp. 115-133, 1943.

[10] M. Minsky and S. Papert, *Perceptrons: an introduction to computational geometry*, MIT Press, 1969.

[11] S. 'Olafson and Y. Abu-Mostafa, "The capacity of multilevel threshold functions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-10, No. 2, March 1988.

[12] D. Rumelhart, and J. McLelland, *Parallel distributed processing: explorations into the microstructure of cognition*, MIT Press, 1986.

[13] R. Shonkwiler, "Separating the vertices of N-cubes by hyperplanes and its application to artificial neural networks," *IEEE Transactions on Neural Networks*, vol. TNN-4, No. 2, pp. 343-347, March 1993.

[14] A. Weiland and R. Leighton, "Geometric analysis of network capabilities," *IEEE Int. Conf. on Neural Networks*, vol. 3, pp. 385-392, June 1987.

[15] R. Winder, "Bounds on threshold gate realizability," *IEEE Transactions on Electronic Computers*, vol. EC-12, pp. 561-564, 1963.

[16] R. Winder, "Partitions of N-space by hyperplanes," *J. SIAM Appl. Math.*, vol. 14, No. 4, pp. 811-818, July 1966.