Algebraic Techniques for Constructing Minimal Weight Threshold Functions *

Vasken Bohossian Jehoshua Bruck

California Institute of Technology
Mail Code 136-93
Pasadena, CA 91125
E-mail: {vincent, bruck}@paradise.caltech.edu

Abstract

A linear threshold element computes a function that is a sign of a weighted sum of the input variables. The weights are arbitrary integers; actually, they can be very big integers—exponential in the number of the input variables. While in the present literature a distinction is made between the two extreme cases of linear threshold functions with polynomial-size weights as opposed to those with exponential-size weights, the best known lower bounds on the size of threshold circuits are for depth-2 circuits with small weights. Our main contributions are devising two distinct methods for constructing threshold functions with minimal weights and filling up the gap between polynomial and exponential weight growth by further refining the separation. Namely, we prove that the class of linear threshold functions with polynomial-size weights can be divided into subclasses according to the degree of the polynomial. In fact, we prove a more general result—that there exists a minimal weight linear threshold function for any arbitrary number of inputs and any weight size.

1 Introduction

The present paper focuses on the study of a single linear threshold gate with binary inputs and output as well as integer weights. Such a gate is mathematically described by a *linear threshold function.*

Definition 1 (Linear Threshold Function) *A linear threshold function of n variables is a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ that can be written, for any $X \in \{0,1\}^n$ and a fixed $W \in Z^{n+1}$, as :*

$$f(X) = sgn(F(X)) = \begin{cases} 1 & , \textit{for } F(X) \geq 0 \\ 0 & , \textit{otherwise} \end{cases}$$

$$\textit{where } F(X) = W \cdot (-1, X) = -w_0 + \sum_{i=1}^{n} w_i x_i$$

Although we could allow the weights, w_i, to be real numbers, it is known [Muroga 71], that for a binary input neuron, one needs $O(n \log n)$ bits per weight, where n is the number of inputs. So in

*This work was supported in part by the NSF Young Investigator Award CCR-9457811 and by the Sloan Research Fellowship.

the rest of the paper, we will assume without loss of generality that all weights are integers. Also, notice that a linear threshold function can be implemented as :

$$f : \{-1, 1\}^n \rightarrow \{0, 1\}$$

We will address both the $\{0, 1\}$ and the $\{-1, 1\}$ representations.

1.1 Motivation

Many experimental results in the areas of threshold circuits and neural networks have indicated that the magnitudes of the coefficients in the linear threshold elements grow very fast with the size of the inputs and therefore limit the practical use of the network. One natural question to ask is the following. How limited is the computational power of the network if one limits oneself to threshold elements with only "small" growth in the size of the coefficients? To answer that question we have to define a measure of the magnitudes of the weights. Note that, given a function f, the weight vector W is not unique (see Example 1 below).

Definition 2 (Weight Space) *Given a linear threshold function f we define $\mathcal{W}$ as the set of all weights that satisfy Definition 1, that is*

$$\mathcal{W} = \{W \in Z^n : \forall X \in \{0, 1\}^n, sgn(W \cdot (-1, X)) = f(X)\}$$

Here follows a measure of the size of the weights.

Definition 3 (Minimal Weight Size) *We define the size of a weight vector as the sum of the absolute values of the weights. The minimal weight size of a linear threshold function is defined as*

$$S[f] = \min_{W \in \mathcal{W}} (\sum_{i=0}^{n} |w_i|)$$

The particular vector that achieves the minimum is called a minimal weight vector.

Naturally, $S[f]$ is a function of n. It has been shown [Hastad 94], [Myhill 61], [Shawe-Taylor 92], [Siu 91] that there exists a linear threshold function that can be implemented by a single threshold element with exponentially growing weights, $S[f] \sim 2^n$, but cannot be implemented by a threshold element with smaller : polynomialy growing weights, $S[f] \sim n^d$, d constant. In light of that result the above question was dealt with by defining a class within the set of linear threshold functions : the class of functions with "small" (i.e. polynomialy growing) weights [Siu 91]. Most of the recent research focuses on the power of circuits with small weights, relative to circuits with arbitrary weights [Goldmann 92], [Goldman 93]. Rather than dealing with circuits we are interested in studying a single threshold gate. The main contribution of the present paper is to further refine the division of small versus arbitrary weights. We separate the set of functions with small weights into classes indexed by d, the degree of polynomial growth and show that all of them are non-empty. In particular, we develop a technique for proving that a weight vector is minimal. We use that technique to construct a function of size $S[f] = s$ for an arbitrary s. The only known lower bounds for threshold circuits involve small weights [Hajnal 93]. Our techniques might help in improving the results in that domain.

1.2 Organization

Here follows a brief outline of the rest of the paper. In section 2 we show some of the difficulties one faces when minimizing the weights as well as how the latter are affected by the choice of input domain. In section 3 we consider functions defined over $\{-1, 1\}$. We limit ourselves to functions with no threshold (generalized majority function) and we show how to construct such functions with minimal weights. In section 4 we present another way of constructing minimal functions that allows us to deal with any threshold function defined over $\{0, 1\}$.

2 Preliminaries and Examples

In this section we illustrate some of the difficulties one faces when trying to minimize the weights of a threshold function. We also show how the input domain (i.e. $\{0, 1\}$ versus $\{-1, 1\}$) affects the size of the weights. See [Krause 95] for related results.

2.1 Minimizing the weights

The main difficulty in analyzing the size of the weights of a threshold element is due to the fact that a single linear threshold function can be implemented by different sets of weights as shown in the following example.

Example 1 (A Threshold Function with Minimal Weights) *Let us consider the following two sets of weights (weight vectors).*

$$W_1 = (4\ \ 1\ \ 2\ \ 5),\ F_1(X) = -4 + x_1 + 2x_2 + 5x_3$$

$$W_2 = (8\ \ 2\ \ 4\ \ 10),\ F_2(X) = -6 + 2x_1 + 4x_2 + 10x_3$$

They both implement the same threshold function

$$f(X) = sgn(F_2(X)) = sgn(2F_1(X)) = sgn(F_1(X))$$

A closer look reveals that $f(X) = sgn(-1 + X_3)$, implying that none of the above weight vectors has minimal size. Indeed, the minimal one is $W_3 = (1\ \ 0\ \ 0\ \ 1)$ and $S[f] = 2$.

To determine if a given set of weights is minimal is in general a difficult problem, [Amaldi 93], [Willis 63]. Our technique consists of constructing weight vectors whose minimality is easily established. We then show how to modify them, while keeping them minimal, in order to get to a larger set of functions.

2.2 $\{0, 1\}$ versus $\{-1, 1\}$

Suppose we implement the same function over $\{0, 1\}$ and over $\{-1, 1\}$. How are the weights affected? Let us look at an example.

Example 2 (The OR function)

1. *Let $x_i \in \{0, 1\}$,*

$$OR(x_1, ..., x_n) = sgn(-1 + x_1 + ...x_n)$$

The size of the weights is $S = n + 1$. Those weights are minimal.
Proof: *The weights are integers. Reducing their size implies reseting one or more of them to 0, which will violate the definition of OR.* □

2. Now, let $x_i \in \{-1,1\}$,

$$OR(x_1,...,x_n) = sgn(n-2+x_1+...+x_n)$$

The size of the weights is $S = 2n-2$. Those weights are minimal as well.
Proof: *Any weights that implement OR have to be positive. Suppose there exist weights of size $S' < 2n-2$. No weight can be 0, so $\sum_1^n w' \geq n$, implying that the threshold $-w_0 < (2n-2)-n = n-2$. Let w_i' be the smallest weight. Set $x_i = 1$ and all other inputs to -1. $\sum_1^n w' < -w_i(n-2)$ so that $F(X) < 0$ violating the definition of OR.* □

It appears from this example that the $\{0,1\}$ implementation has smaller weight size than the $\{-1,1\}$ representation. Is that true in general?

Example 3 (The Majority (MAJ) function) *Let the number of variables, n, be odd. The majority function outputs true if more than half of its inputs are true.*

- *Let $x_i \in \{0,1\}$,*
$$MAJ(x_1,...,x_n) = sgn(-\frac{n+1}{2}+x_1+...+x_n)$$
The size of the weights is $S = \frac{3n+1}{2}$. We show they are minimal by a proof similar to case 2, above.

- *Now, let $x_i \in \{-1,1\}$,*
$$MAJ(x_1,...,x_n) = sgn(x_1+...x_n)$$
Those weights are minimal since reducing them would imply reseting one or more of them to 0, which will violate the definition of MAJ. The size of the weights is $S = n$.

This second example shows that in general we cannot tell which implementation $\{0,1\}$ or $\{-1,1\}$ will produce a function with smaller weights.

3 Generalized Majority Function over $\{-1,1\}$

In this section we study the following model :

$$f : \{-1,1\} \to \{0,1\}$$

$$f(X) = sgn(\sum_1^n w_i x_i)$$

Notice that there is no threshold; we are looking at a majority function with arbitrary weights. We address the problem of constructing functions with minimal weights. In particular, our goal is that for a given number of inputs n and size S, we find a function.

3.1 Mathematical setting

We are interested in constructing functions for which the minimal weight is easily determined. Finding the minimal weight involves a search, we are therefore interested in finding functions with a constrained weight spaces. The following tools allows us to put constraints on W.

Definition 4 (Root Space of a Boolean Function) *A vector $\vec{v} \in \{-1,1\}^n$ such that $f(\vec{v}) = f(-\vec{v})$ is called a root of f. We define the root space, R, as the set of all roots of f.*

Definition 5 (Root Generator Matrix) *For a given weight vector $\vec{w} \in W$ and a root $\vec{v} \in R$, the root generator matrix, $G = (g_{ij})$, is a $(n \times k)$-matrix, with entries in $\{-1, 0, 1\}$, whose rows $\vec{g}$ are orthogonal to $\vec{w}$ and equal to $\vec{v}$ at all non-zero coordinates, namely,*

1. $G\vec{w} = \vec{0}$
2. $g_{ij} = 0$ *or* $g_{ij} = v_j$ *for all i and j.*

Example 4 (Root Generator Matrix) *Suppose that we are given a linear threshold function specified by a weight vector $\vec{w} = (1, 1, 2, 4, 1, 1, 2, 4)$. By inspection we determine one root $\vec{v} = (1, 1, 1, 1, -1, -1, -1, -1)$. Notice that $w_1 + w_2 - w_7 = 0$ which can be written as $\vec{g} \cdot \vec{w} = 0$, where $\vec{g} = (1, 1, 0, 0, 0, 0, -1, 0)$ is a row of G. Set $\vec{r} = \vec{v} - 2\vec{g}$. Since $\vec{g}$ is equal to $\vec{v}$ at all non-zero coordinates, $\vec{r} \in \{-1, 1\}^n$. Also $\vec{r} \cdot \vec{w} = \vec{v} \cdot \vec{w} + \vec{g} \cdot \vec{w} = 0$. We have generated a new root : $\vec{r} = (-1, -1, 1, 1, -1, -1, 1, -1)$.*

Lemma 1 (Orthogonality of G and W) *For a given weight vector $\vec{w} \in W$ and a root $\vec{v} \in R$, $\vec{u}G^T = \vec{0}$ holds for any weight vector $\vec{u} \in W$.*

Proof: For an arbitrary $\vec{u} \in W$ and an arbitrary row, $\vec{g}_i$, of G, let $\vec{v}' = \vec{v} - 2\vec{g}_i$. By definition of $\vec{g}_i$, $\vec{v}' \in \{-1, 1\}^n$ and $\vec{v}' \cdot \vec{w} = 0$. That implies $f(\vec{v}') = f(-\vec{v}')$: $\vec{v}'$ is a root of f. For any weight vector $\vec{u} \in W$, $sgn(\vec{u} \cdot \vec{v}') = sgn(-\vec{u} \cdot \vec{v}')$. Therefore $\vec{u} \cdot (\vec{v} - 2\vec{g}_i) = 0$ and finally, since $\vec{v} \cdot \vec{u} = 0$ we get $\vec{u} \cdot \vec{g}_i = 0$. □

Lemma 2 (Minimality) *For a given weight vector $\vec{w} \in W$ and a root $\vec{v} \in R$ if $rank(G) = n - 1$ (i.e. G has $n - 1$ independent rows) and $|w_i| = 1$ for some i, then $\vec{w}$ is the minimal weight vector.*

Proof: From Lemma 1 any weight vector $\vec{u}$ satisfies $\vec{u}G^T = \vec{0}$. $rank(G) = n - 1$ implies that $dim(W) = 1$, i.e. all possible weight vectors are integer multiples of each other. Since $|w_i| = 1$, all vectors are of the form $\vec{u} = k\vec{w}$, for $k \geq 1$. Therefore $\vec{w}$ has the smallest size. □

We complete Example 4 with an application of Lemma 2.

Example 5 (Minimality) *Given $\vec{w} = (1, 1, 2, 4, 1, 1, 2, 4)$ and $\vec{v} = (1, 1, 1, 1, -1, -1, -1, -1)$ we can construct :*

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

It is easy to verify that $rank(G) = n - 1 = 7$ and therefore, by Lemma 2, $\vec{w}$ is minimal and $S[f] = 16$.

3.2 Weight Vectors

In Example 5 we saw how, given a weight vector, one can show that it is minimal. In this section we present an example of a linear threshold function with minimal weight size, with an arbitrary number of input variables.

We would like to construct a weight vector and show that it is minimal. Let the number of inputs, n, be even. Let $\vec{w}$ consist of two identical blocks : $(w_1, w_2, ..., w_{n/2}, w_1, w_2, ..., w_{n/2})$. Clearly,

$\vec{v} = (1, 1, ..., 1, -1, -1, ..., -1)$ is a root and G is the corresponding generator matrix.

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & ... & 0 & 0 & 0 & -1 & 0 & 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & ... & 0 & 0 & 0 & 0 & -1 & 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & ... & 0 & 0 & 0 & 0 & 0 & -1 & 0 & ... & 0 & 0 & 0 \\ \vdots & & & & & & & & & & & & & & & \vdots \\ 0 & 0 & 0 & 0 & ... & 0 & 1 & 0 & 0 & 0 & 0 & 0 & ... & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & ... & 0 & 0 & 1 & 0 & 0 & 0 & 0 & ... & 0 & 0 & -1 \end{pmatrix}$$

3.3 Construction

The following theorem states that given an integer s and a number of variables n there exists a function of n variables and minimal weight size s.

Theorem 3 (Main Result) *For any pair (s, n) that satisfies*

1. $n \leq s \leq \begin{cases} 2^{\frac{n}{2}} & \text{, for } n \text{ even} \\ 2^{\frac{n-1}{2}} + 2^{\frac{n-3}{2}} & \text{, for } n \text{ odd} \end{cases}$

2. *s even*

there exists a linear threshold function of n variables, f, with minimal weight size $S[f] = s$.

Proof: Given a pair (s, n), that satisfies the above conditions we first construct a weight vector $\vec{w}$ that satisfies $\sum_{i=1}^{n} |w_i| = s$, then show that it is the minimal weight vector of the function $f(x) = sgn(\vec{w} \cdot \vec{x})$. The proof is shown only for n even.

CONSTRUCTION.

1. Define $(a_1, a_2, ..., a_{n/2}) = (1, 1, ..., 1)$.
2. If $\sum_{i=1}^{n/2} a_i < s/2$ then increase by one the smallest a_i such that $a_i < 2^{i-2}$. (In the case of a tie take the w_i with smallest index i).
3. Repeat the previous step until $\sum_{i=1}^{n/2} a_i = s/2$ or $(a_1, a_2, ..., a_N) = (1, 1, 2, 4, ..., 2^{\frac{n}{2}-2})$.
4. Set $\vec{w} = (a_1, a_2, ..., a_{n/2}, a_1, a_2, ..., a_{n/2})$.

Because we increase the size by one unit at a time the algorithm will converge to the desired result for any integer s that satisfies $n \leq s \leq 2^{\frac{n}{2}}$. We have a construction for any valid (s, n) pair. Let us show that $\vec{w}$ is minimal.

MINIMALITY. Given that $\vec{w} = (a_1, a_2, ..., a_{n/2}, a_1, a_2, ..., a_{a/2})$ we find a root $\vec{v} = (1, 1, ..., 1, -1, -1, ..., -1)$ and $n/2$ rows of the generator matrix G corresponding to the equations $w_i = w_{i+\frac{n}{2}}$. To form additional rows note that the first k a_i's are powers of two (where k depends on s and n). Those can be written as $a_i = \sum_{j=1}^{i-1} a_j$ and generate $k-1$ rows. And finally note that all other a_i, $i > k$, are smaller than 2^{k+1}. Hence, they can be written as a binary expansion $a_i = \sum_{j=1}^{k} \alpha_{ij} a_j$ where $\alpha_{ij} \in \{0, 1\}$. There are $\frac{n}{2} - k$ such weights. G has a total of $n-1$ independent rows. $rank(G) = n-1$ and $w_1 = 1$, therefore by Lemma 2, $\vec{w}$ is minimal and $S[f] = s$. □

Example 6 (A Function of 10 variables and size 26) *We start with $\vec{a} = (1, 1, 1, 1, 1)$. We iterate:* $(1, 1, 2, 1, 1)$, $(1, 1, 2, 2, 1)$, $(1, 1, 2, 2, 2)$, $(1, 1, 2, 3, 2)$, $(1, 1, 2, 3, 3)$, $(1, 1, 2, 4, 3)$, $(1, 1, 2, 4, 4)$, *and finally the algorithm converges to* $\vec{a} = (1,1,2,4,5)$. *We claim that* $\vec{w} = (\vec{a},\vec{a}) = (1,1,2,4,5,1,1,2,4,5)$ *is minimal. Indeed,* $\vec{v} = (1,1,1,1,1,-1,-1,-1,-1,-1)$ *and*

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

is a matrix of rank 9.

Example 7 (Functions with Polynomial Size) *This example shows an application of Theorem 3. We define* $\widehat{LT}^{(d)}$ *as the set of linear threshold functions for which* $S[f] \leq n^d$. *The Theorem states that for any even* n *there exists a function* f *of* n *variables and minimum weight* $S[f] = n^d$. *The implication is that for all* d, $\widehat{LT}^{(d-1)}$ *is a proper subset of* $\widehat{LT}^{(d)}$.

4 Arbitrary Threshold Function over $\{0,1\}$

In this section we present a different technique for constructing threshold functions with minimal weights. It allows us to construct functions with any weight size and number of variables. We consider functions with input domain $\{0,1\}$, but as mentioned below, the argument holds for an arbitrary input space $\{a,b\}$.

4.1 Approach

The method we use is based on a result from [Willis 63]. We assume, without loss of generality, that the weights are strictly positive integers. Our goal is to minimize $S = \sum_0^n |w_i| = \sum_0^n w_i$. We know from [Muroga 71] that any other weights, U, implementing the same function have to be strictly positive. We will show that under certain conditions on W, $\sum_0^n w_i \geq \sum_0^n u_i$ for any U.

Consider input vectors X and Y for which the following equations hold :

$$F(X) = -w_0 + \sum_1^n w_i x_i = 0 \qquad\qquad F(Y) = -w_0 + \sum_1^n w_i y_i = -1$$

Let them define the rows of a matrix that we call A :

$$A = \begin{pmatrix} -1 & X^{(1)} \\ -1 & X^{(2)} \\ \vdots & \vdots \\ -1 & X^{(p)} \\ 1 & -Y^{(1)} \\ 1 & -Y^{(2)} \\ \vdots & \vdots \\ 1 & -Y^{(q)} \end{pmatrix} = \begin{pmatrix} -1 & x_1^{(1)} & x_2^{(1)} & \ldots & x_n^{(1)} \\ -1 & x_1^{(2)} & x_2^{(2)} & \ldots & x_n^{(2)} \\ \vdots & & & & \vdots \\ -1 & x_1^{(p)} & x_2^{(p)} & \ldots & x_n^{(p)} \\ 1 & -y_1^{(1)} & -y_2^{(1)} & \ldots & -y_n^{(1)} \\ 1 & -y_1^{(2)} & -y_2^{(2)} & \ldots & -y_n^{(2)} \\ \vdots & & & & \vdots \\ 1 & -y_1^{(q)} & -y_2^{(q)} & \ldots & -y_n^{(q)} \end{pmatrix}$$

We allow repetition of rows : we may have $X^{(i)} = X^{(j)} = \ldots = X^{(k)}$.

Example 8 (The matrix A) *Suppose we are given the following weights :* $W = (13\ \ 6\ \ 6\ \ 3\ \ 3\ \ 3\ \ 2\ \ 2\ \ 1\ \ 1)$ *Our goal is to show they are minimal. We need to first construct the matrix A. Here follows a candidate :*

$$A = \begin{pmatrix} -1 & X^{(1)} \\ -1 & X^{(2)} \\ 1 & -Y^{(1)} \\ 1 & -Y^{(2)} \end{pmatrix} = \begin{pmatrix} -1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ -1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \\ 1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \end{pmatrix}$$

There are many possible choices for A. The one shown above is not a good one as we will see. □

Theorem 4 (Condition for Minimality) *Given a weight vector W, we construct A as described above. If there exists $a > 0$, such that A satisfies :*

$$(1\ ...\ 1)A = (a\ ...\ a)$$

the weight vector W is minimal.

Proof: By definition of the X's and the Y's the matrix A satisfies :

$$A \cdot (w_0\ w_1\ w_2\ ...\ w_n)^T = (\overbrace{0\ 0\ ...\ 0\ 0}^{p}\ \overbrace{1\ 1\ ...\ 1\ 1}^{q})^T \tag{1}$$

Because $sgn(0) = 1$ and $sgn(-1) = 0$ any other weight vector, U, implementing the same function has to verify the above equalities with "$\geq$" instead of "$=$" :

$$A \cdot (u_0\ u_1\ u_2\ ...\ u_n)^T \geq (\overbrace{0\ 0\ ...\ 0\ 0}^{p}\ \overbrace{1\ 1\ ...\ 1\ 1}^{q})^T \tag{2}$$

Let $V = U - W$, and subtract Equations (1) from Inequalities (2), we get :

$$A \cdot (v_0\ v_1\ v_2\ ...\ v_n)^T \geq (\overbrace{0\ 0\ ...\ 0\ 0}^{p+q})^T \tag{3}$$

Now suppose A is such that :

$$(\overbrace{1\ 1\ ...\ 1\ 1}^{p+q}) \cdot A = (\overbrace{a\ a\ ...\ a\ a}^{n}) \tag{4}$$

Where a is a strictly positive integer. We multiply Inequalities (3) by the all 1 vector from the left and get :

$$(\overbrace{1\ 1\ ...\ 1\ 1}^{p+q}) \cdot A \cdot (v_0\ v_1\ v_2\ ...\ v_n)^T \geq (\overbrace{1\ 1\ ...\ 1\ 1}^{p+q}) \cdot (\overbrace{0\ 0\ ...\ 0\ 0}^{p+q})^T$$

$$(\overbrace{a\ a\ ...\ a\ a}^{n}) \cdot (v_0\ v_1\ v_2\ ...\ v_n)^T \geq 0$$

$$a \sum_0^n v_i \geq 0$$

And since $a > 0$, $w_i \geq 0$, $u_i \geq 0$ for all $i = 0, ..., n$ we know that : $\sum_0^n u_i \geq \sum_0^n w_i$ □

Notice that nowhere in the proof did we use the fact that the input domain is $\{0, 1\}$. Indeed, the above proof is valid for any input domain $\{a, b\}$. As you can see the proof relies on constructing A so that Equation (4) holds. To construct A we need appropriate X's and Y's which in turn depend on the choice W.

4.2 Basic construction

In this section we introduce W, the weight vector for the general construction, and prove it is minimal by finding an appropriate matrix A. Let the threshold, w_0 be arbitrary. We choose $w_1 = \lfloor \frac{w_0}{2} \rfloor$, $w_3 = \lfloor \frac{w_0 - w_1}{2} \rfloor$, $w_5 = \lfloor \frac{w_0 - w_1 - w_3}{2} \rfloor$, ..., $w_{n-1} = 1$, and $w_{2i} = w_{2i-1}$ for $i = 1...n$. We choose n so that $\sum_{i=1}^{n} w_{2i-1} = w_0 - 1$. Let us look at an example :

Example 9 **($w_0 = 13$)** *Applying the above recursive definition we get the weight vector of Example 8 : $W = (13\ 6\ 6\ 3\ 3\ 2\ 2\ 1\ 1)$ Here follow the X and Y-type rows for A.*

$$\left.\begin{cases} -1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ -1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \end{cases}\right\} sumX_1 = (-2\ 1\ 1\ 1\ 1\ 1\ 1\ 2\ 2)$$

$$\left.\begin{cases} -1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{cases}\right\} sumX_2 = (-2\ 1\ 1\ 1\ 1\ 2\ 2\ 0\ 0)$$

$$\left.\begin{cases} -1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ -1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \end{cases}\right\} sumX_3 = (-2\ 1\ 1\ 2\ 2\ 0\ 0\ 1\ 1)$$

$$\left.\begin{cases} -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{cases}\right\} sumX_4 = (-2\ 2\ 2\ 0\ 0\ 0\ 0\ 1\ 1)$$

$$\underbrace{\begin{matrix} 1 & -1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & 0 & -1 & 0 & -1 \end{matrix}}_{sumY_1 = (2\ -1\ -1\ -1\ -1\ -1\ -1\ -1\ -1)}$$

We replicate rows and add them in order to get to the all 1 vector. Only odd numbered columns are shown.

$$\begin{pmatrix} -2 & 1 & 1 & 1 & 2 \\ -2 & 1 & 1 & 2 & 0 \\ -2 & 1 & 2 & 0 & 1 \\ -2 & 2 & 0 & 0 & 1 \\ -2 & -1 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 1 & -1 & -1 & 0 \\ -2 & -1 & -1 & -1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 2 & -1 & -1 & -1 & -1 \end{pmatrix}$$

The latter of which add up to the all 2 vector. □

Theorem 5 (Minimality of the Construction) *For any w_0 we can construct a threshold function with minimal weights of size $S = 3 * w_0 - 2$ and number of variables $n = \lceil \log_2 S \rceil$.*

Proof: We are going to construct A, show that it satisfies $\underline{1}A = a\underline{1}$ and apply Theorem 4. Only two Y-type vectors are needed for the construction of A :

$$\begin{pmatrix} 1 & -1 & 0 & -1 & 0 & ... & -1 & 0 \\ 1 & 0 & -1 & 0 & -1 & ... & 0 & -1 \end{pmatrix}$$

They add up to $(2\ \ -1\ \ ...\ \ -1)$. The X-type vectors, summed two by two, add up to two possible forms :

$$-2\ \ 1\ \ ...\ \ 1\ \ 2\ \ 0\ \ ...\ \ 0\ \ 0$$
$$\text{or}$$
$$-2\ \ 1\ \ ...\ \ 1\ \ 2\ \ 0\ \ ...\ \ 0\ \ 1$$

By repeating and adding those partial sums one can get to the all 1 vector. How do we do that? We produce the $(0, ..., 0, 1)$ vector by adding two Y and two X-type vectors.

$$\begin{pmatrix} 2 & -1 & ... & -1 & -1 \\ -2 & 1 & ... & 1 & 2 \end{pmatrix}$$

Let us denote by S_i, $i = 1..n$, the singleton vector $(0, ...0, 1, 0, ..., 0)$, where the 1 is in the i^{th} position. We use induction to show that we can get to all S_i by adding up X and Y-type vectors. Indeed, suppose we have obtained all S_j for $j = 1, ..., i-1$. We can produce S_i by adding two X and two Y-type vectors :

$$\begin{pmatrix} 2 & -1 & -1 & -1 & -1 & .. & -1 & -1 & -1 & -1 \\ -2 & 1 & .. & 1 & 2 & 0 & .. & 0 & 0 & 0 \\ 0 & 0 & .. & 0 & 0 & 1 & 0 & .. & 0 & 0 \\ 0 & 0 & .. & 0 & 0 & 0 & 1 & 0 & .. & 0 \\ \vdots & & & & & & & & & \vdots \\ 0 & 0 & .. & 0 & 0 & 0 & 0 & .. & 0 & 1 \end{pmatrix}$$

Once we have all S_i vectors, we add them up 3 times to $(2 \quad -1 \ \ldots \ -1)$ in order to get to the all 2 vector. $\Box$

4.3 Construction for arbitrary size and number of variables

In this section we show how to split a weight in order to get an additional variable. We also prove that adding one or two variables with unit weight results in a minimal function as well.

Lemma 6 (Splitting a Weight) *Let* $W = (w_0, w_1, ..., w_n)$ *be minimal. Then* $\tilde{W} = (w_0, a, b, w_2, w_3, ..., w_{n+1})$ *where* $a + b = w_1$ *is also minimal.*

Proof: Construct A while duplicating the second column. $\Box$

Lemma 7 (Adding an input with unit weight) *Let* $W = (w_0, w_1, ..., w_n)$ *be minimal. Then* $\tilde{W} = (w_0, w_1, w_2, w_3, ..., w_{n+1})$ *where* $w_{n+1} = 1$ *is also minimal.*

Proof: Suppose it is not minimal, implying there exists a better choice for $\tilde{W}$, let us call it W'. There are two possibilities. Either $w'_{n+1} = 0$ or some of the w'_i for $i < n+1$ is smaller than the corresponding w_i. In the latter case, we set $x_{n+1} = 0$ and obtain the original function implemented with smaller weights, contradicting the hypothesis. Now suppose $w'_{n+1} = 0$, implying that $\tilde{f}$ does not depend on x_{n+1}. That in turn implies $\sum_0^n w_i x_i \geq 0$ or $\sum_0^n w_i x_i \leq -2$ for all inputs X. We can reduce w_0 by 1, implying the original function was not minimal. $\Box$

Using those two lemmas, the construction of functions with arbitrary size and number of variables is straightforward.

5 Conclusions

We presented two techniques for constructing minimal weight threshold functions of arbitrary weight size and number of inputs. We considered both the $\{0, 1\}$ and $\{-1, 1\}$ input domains. Using these techniques we further refined the separation between polynomialy and exponentially growing weights. The natural open problem is to find out if this new techniques are useful in extending the existing lower bounds [Hajnal 93] on circuit size to functions with arbitrary weights.

References

[Amaldi 93] E. Amaldi and V. Kann. The complexity and approximability of finding maximum feasible subsystems of linear relations. Ecole Polytechnique Federale De Lausanne Technical Report, *ORWP 93/11*, August 1993.

[Goldmann 92] M. Goldmann, J. Hastad, and A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, (2):277–300, 1992.

[Goldman 93] M. Goldmann and M. Karpinski. Simulating threshold circuits by majority circuits. In *Proc. 25th ACM STOC*, pp. 551–560, 1993.

[Hajnal 93] A. Hajnal, W. Maass, P. Pudlak, M. Szegedy and G. Turan. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, v46 (2), pp. 129 - 154, April 1993.

[Hastad 94] J. Hastad. On the size of weights for threshold gates. *SIAM. J. Disc. Math.*, 7:484–492, 1994.

[Krause 95] M. Krause and P. Pudlak. On Computing Boolean Functions by Sparse Real Polynomials. Proceedings of the 36th Annual Symposium on Foundations of Computer Science, pp. 682–691, October 1995.

[Muroga 71] M. Muroga. *Threshold Logic and its Applications.* Wiley-Interscience, 1971.

[Myhill 61] J. Myhill and W. H. Kautz. On the size of weights required for linear-input switching functions. *IRE Trans. Electronic Computers*, (EC10):pp. 288–290, 1961.

[Shawe-Taylor 92] J. S. Shawe-Taylor, M. H. G. Anthony, and W. Kern. Classes of feedforward neural networks and their circuit complexity. *Neural Networks*, Vol. 5:pp. 971–977, 1992.

[Siu 91] K. Siu and J. Bruck. On the power of threshold circuits with small weights. *SIAM J. Disc. Math.*, Vol. 4(No. 3):pp. 423–435, August 1991.

[Willis 63] D. G. Willis. Minimum weights for threshold switches. In *Switching Theory in Space Techniques.* Stanford University Press, Stanford, Calif., 1963.