Trading Weight Size for Circuit Depth: An $\widehat{LT}_2$ Circuit for Comparison*

Vasken Bohossian, Marc D. Riedel and Jehoshua Bruck

California Institute of Technology
Mail Code: 136-93
Pasadena, CA 91125

Email: {vincent,riedel,bruck}@paradise.caltech.edu

November 13, 1998

Abstract

We present an explicit construction of a circuit for the COMPARISON function in $\widehat{LT}_2$, the class of polynomial-size linear threshold circuits of depth two with polynomially growing weights. Goldmann and Karpinski proved that $LT_1 \subset \widehat{LT}_2$ in [4]. Hofmeister presented a simplified version of the same result in [6]. We have further simplified the results of these two papers by limiting ourselves to the simulation of COMPARISON. Our construction has size $O(n^4 \log n)$, a significant improvement on the general bound of $O(n^{12} \log^{11} n)$ in [6].

*Supported in part by an NSF Young Investigator Award (CCR-9457811), by a Sloan Research Fellowship, by an IBM Partnership Award and by DARPA through an agreement with NASA/OSAT.

1 Introduction

A *linear threshold function* $f(X)$ is a Boolean-valued function with Boolean inputs $X = x_1, x_2, \cdots, x_n$, such that

$$f(X) = \mathrm{sgn}[F(X)] = \begin{cases} 1 & \text{for } F(X) \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\text{where } F(X) = w_0 + \sum_{i=1}^{n} w_i x_i$$

for some fixed *weights* $w_i \in \mathbf{Z}$, $0 \leq i \leq n$.

A *linear threshold gate* is a device that computes a linear threshold function. The class LT consists of Boolean functions that are computable with a single linear threshold gate. The class LT_d consists of Boolean functions that are computable by an unbounded fan-in, polynomial-size, depth d circuit of linear threshold gates. (The size of the circuit is the number of wires.)

Threshold circuits have been widely studied; surveys of the topic can be found in [7] and [9]. A question that arises is how powerful threshold circuit are if one limits oneself to threshold gates with only "small" growth in the size of the weights [2], [11]. It has been shown that there exist functions that can be implemented by a linear threshold gate with exponentially growing weights, but cannot be implemented by one with polynomially growing weights [5], [8], [10], [11].

In light of this result, a subclass of LT has been defined: the class $\widehat{LT}$ of functions with "small" weights. Each function

$$f(X) = \mathrm{sgn}\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)$$

in $\widehat{LT}$ is characterized by the property that the weights are integers bounded by a polynomial in n, i.e., $|w_i| \leq n^c$ for some constant $c > 0$. Siu and Bruck proved that $LT_d \subset \widehat{LT}_{2d+1}$ [11]. Goldmann and Karpinski improved the bound to $LT_d \subset \widehat{LT}_{d+1}$ by showing that $LT_1 \subset \widehat{LT}_2$ and generalizing to arbitrary depth [4]. Hofmeister presented a simplified version of the proof that $LT_1 \subset \widehat{LT}_2$ [6]. The idea is to use two operations in order to reduce the weights: divide them by powers of two and divide them modulo a prime. The resulting "small"-weight gates are connected into a circuit that produces the correct output if enough primes are used.

We have further simplified the results presented in [4] and [6] by limiting ourselves to the simulation of a particular large weight function: COMPARISON. Our construction has size $O(n^4 \log n)$, a significant improvement on the general bound of $O(n^{12} \log^{11} n)$ in [6].

We note that Alon and Bruck have presented a sophisticated construction for the COMPARISON function in [1], based upon a discrete version of a sparse "delta polynomial" (one that has a large absolute value for a single assignment and extremely small absolute values for all other assignments). The construction we present here is considerably simpler, and readily implementable for practical values of n.

2 $\widehat{LT}_2$ Circuit for Comparison

Let X_1 and X_2 be two n-bit numbers, $X_1 = x_1, x_3, \ldots, x_{2n-1}$, $X_2 = x_2, x_4, \ldots, x_{2n}$, where $x_i \in \{0,1\}$, for $1 \leq i \leq 2n$. The integer values represented by X_1 and X_2 are equal to $\sum_{i=1}^{n} x_{2i-1} 2^{i-1}$ and $\sum_{i=1}^{n} x_{2i} 2^{i-1}$, respectively. The COMPARISON function is defined as

$$C(X_1, X_2) = \begin{cases} 1 & X_1 > X_2 \\ 0 & \text{otherwise.} \end{cases}$$

In other words,

$$\begin{aligned} C(X_1, X_2) &= \text{sgn}[X_1 - X_2] \\ &= \text{sgn}\left[\sum_{i=1}^{n} 2^{i-1}(x_{2i-1} - x_{2i})\right]. \end{aligned}$$

The COMPARISON function has the interesting property that it belongs to LT_1, but not to $\widehat{LT}_1$. Using tools from harmonic analysis, it is shown in [11] that COMPARISON is in $\widehat{LT}_2$. We provide an explicit construction of an $\widehat{LT}_2$ circuit for COMPARISON, inspired by the method in [6].

Let $X = x_1, x_2, \ldots x_{2n}$ and let $F(X) = \sum_{i=1}^{2n} w_i x_i$ be the linear combination corresponding to $C(X_1, X_2)$:

$$F(X) = C(X_1, X_2) = \sum_{i=1}^{n} 2^{i-1}(x_{2i-1} - x_{2i}).$$

From $F(X)$ we form a new function $F_1(X)$ with weights w'_i by dividing each weight in half and rounding down:

$$w'_i = \left\lfloor \frac{w_i}{2} \right\rfloor$$

Note that the division is equivalent to "left-shifting" both X_1 and X_2:

$$F_1(X) = C\left(\left\lfloor \frac{X_1}{2} \right\rfloor, \left\lfloor \frac{X_2}{2} \right\rfloor\right).$$

We repeat this process to form a sequence of functions $F_l(X)$, for $0 \leq l \leq n$. After n steps, the division process yields $F_n(X)$ which is identically zero.

$$\begin{aligned} F_0(X) &= x_1 - x_2 + 2x_3 - 2x_4 + \cdots + 2^{n-1}x_{2n-1} - 2^{n-1}x_{2n} \\ F_1(X) &= x_3 - x_4 + \cdots + 2^{n-2}x_{2n-1} - 2^{n-2}x_{2n} \\ \vdots &= \vdots \\ F_{n-1}(X) &= x_{2n-1} - x_{2n} \\ F_n(X) &= 0. \end{aligned}$$

Lemma 2.1

$$\begin{aligned} F(X) \geq 0 \quad &\Rightarrow \quad \forall\, l : F_l(X) \geq 0, \\ F(X) \leq 0 \quad &\Rightarrow \quad \forall\, l : F_l(X) \leq 0. \end{aligned}$$

Proof: Note that

$$X_1 \geq X_2 \quad \Rightarrow \quad \left\lfloor \frac{X_1}{2} \right\rfloor \geq \left\lfloor \frac{X_2}{2} \right\rfloor$$

and similarly,

$$X_1 \leq X_2 \quad \Rightarrow \quad \left\lfloor \frac{X_2}{2} \right\rfloor \leq \left\lfloor \frac{X_1}{2} \right\rfloor$$

(see Appendix). The two statements of the lemma follow. □

Lemma 2.2

$F(X) > 0 \quad \Leftrightarrow \quad \exists\, l : F_l(X) = 1.$

Proof: Note that the value of COMPARISON can be determined by examining the highest-order bit position in which X_1 and X_2 differ. (If they do not differ, then the value of COMPARISON is defined to be zero). Suppose that $X_1 \neq X_2$ and the highest-order bit in which they differ is the k-th bit. If the k-th bit of X_1 is 1 and the k-th bit of X_2 is 0, then $C(X_1, X_2) = 1$ and $F(X) > 0$. Now, suppose that we "left-shift" X_1 and X_2 by k bits, obtaining X_1' and X_2'. That is to say, we divide each of the weights of X_1 and X_2 by 2^k and take the floor to obtain the corresponding weights of X_1' and X_2'. If $F(X) > 0$, then $F_k(X) = X_1' - X_2' = 1$. The lemma follows. □

Example 2.1

Suppose $X_1 = 0, 1, 1, 1$ (i.e., 14) and $X_2 = 0, 1, 0, 1$ (i.e., 10). Thus, $F(X) = 4$.

$$\begin{aligned}
F_0(X) &= 0 - 0 + 2(1) - 2(1) + 4(1) - 4(0) + 8(1) - 8(1) = 4 \\
F_1(X) &= 1 - 1 + 2(1) - 2(0) + 4(1) - 4(1) = 2 \\
F_2(X) &= 1 - 0 + 2(1) - 2(1) = 1 \\
F_3(X) &= 1 - 1 = 0 \\
F_4(X) &= 0
\end{aligned}$$

Note that $F(X) > 0$ and we have $\forall\, l : F_l(X) \geq 0$. In particular, we have $F_2(X) = 1$.

Define the "test" function for each $0 \leq l < n$ as follows:

$$T_l(X) = \begin{cases} 1 & \text{if } F_l(X) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 2.2 may be expressed as

$$F(X) > 0 \;\Leftrightarrow\; \bigvee_{l=0}^{n-1} T_l(X) = 1.$$

Although trivial in itself, this notation becomes useful when we introduce the idea of computing modulo prime numbers.

We define the modulus operation to return values in a symmetric interval centered at zero, i.e., for an integer Z and a positive integer k, let Z mod $k = t$, where $t \equiv Z \pmod{k}$ and $-\left\lfloor \frac{k}{2} \right\rfloor \leq t \leq \left\lfloor \frac{k}{2} \right\rfloor$.

Given a prime p, define a "test" operation modulo p for each $0 \leq l < n$ as follows:

$$T_{p,l}(X) = \begin{cases} 1 & \text{if } F_l(X) \bmod p = 1 \\ 0 & \text{otherwise.} \end{cases}$$

For a given $X \in \{0,1\}^{2n}$ and a given prime p, suppose that we compute $T_{p,l}(X)$ for all functions $F_l(X)$ in the sequence. This would not be sufficient, since the test operation modulo a prime p does not necessarily give the correct answer. However, the following lemma tells us that if we repeat the process for enough prime numbers, say r many, then we will obtain the correct answer most of the time.

Lemma 2.3

Let $p_1 < p_2 < \cdots$ be consecutive primes greater than 3. Let s be the minimum integer that satisfies $p_1 p_2 \cdots p_s \geq 2^{n+1} - 1$. Then for every integer Z, where $|Z| \leq 2^n - 1$,

$$\begin{aligned} Z = 1 \quad &\Rightarrow \quad Z \bmod p_i = 1 \text{ for all primes } p_i \geq 3, \\ Z \neq 1 \quad &\Rightarrow \quad Z \bmod p_i = 1 \text{ for less than } 3 \cdot s \text{ many primes } p_i > 3. \end{aligned}$$

Proof: The first statement is trivially true. The second follows from Lemma 2 in [6], based on the Chinese Remainder Theorem. Note that $s = O(n \log n)$. □

For a given $X \in \{0,1\}^{2n}$ and a set of primes $p_1, p_2, \ldots, p_r$, suppose that we have an array of elements that compute $T_{p_i,l}(X)$ for $1 \leq i \leq r$ and $0 \leq l < n$, i.e.,

$T_{p_1,0}(X)$	$T_{p_1,1}(X)$	$\cdots$	$T_{p_1,n-1}(X)$
$T_{p_2,0}(X)$	$T_{p_2,1}(X)$	$\cdots$	$T_{p_2,n-1}(X)$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$T_{p_r,0}(X)$	$T_{p_r,1}(X)$	$\cdots$	$T_{p_r,n-1}(X)$

Define a "false" positive to be the event that an element returns a 1 when $F(X) < 0$. Define a "true" positive to be the event that an element returns a 1 when $F(X) > 0$.

- When $F(X) > 0$, Lemma 2.2 tells us that $\exists l : F_l(X)$. Thus, there is at least one true positive per row. Therefore, there are at least r true positives in the array in total.
- When $F(X) \leq 0$, Lemma 2.1 tells us that $\forall\, l : F_l(X) \neq 1$. Thus, Lemma 2.3 tells us that there are fewer than $3 \cdot s$ false positives per column. Therefore, there are less than $3 \cdot s \cdot n$ false positives in the array in total.

If we choose $r = 3 \cdot s \cdot n$, then the number of elements returning 1's in the case where $F(X) \leq 0$ will always be less than r, whereas the number returning 1's in the case where $F(X) > 0$ will always be greater than or equal to r. The key here is that the upper bound on the number of false positives is independent of the number of rows, whereas the lower bound on the number of true positives is independent of the number of columns.

Example 2.2

For $n = 3$, $F(X)$ can assume values in the range $[-7, 7]$. The following table shows these values reduced modulo 5 and modulo 7:

	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7
mod 5	-2	-1	-0	1	2	-2	-1	0	1	2	-2	-1	0	1	2
mod 7	0	1	2	3	-3	-2	-1	0	1	2	3	-3	-2	-1	0

If we repeatedly reduce a number in the range $[1, 7]$ modulo 5 and modulo 7, we eventually get 1 in both cases. Therefore, at least two test elements return one if $F(X) > 0$. If we repeatedly reduce a number in the range $[-7, 0]$ modulo 5 and modulo 7, we may get 1 in one case but not the other. Therefore, at most one test element returns one if $F(X) \leq 0$.

To obtain a circuit for COMPARISON, we connect the test elements as inputs to an LT gate and set the threshold of the gate to r. It remains to show how to realize the test elements using a single layer of thresholds gates with small weights.

The approach is a standard one in threshold circuit theory. For $1 \leq i \leq r$ and $0 \leq l < n$, define $F_{p_i,l}(X)$ to be the linear combination obtained by reducing the weights of $F_l(X)$ mod p_i. Note that for each $X \in \{0,1\}^{2n}$,

$$T_{p_i,l}(X) = 1 \Leftrightarrow F_{p_i,l}(X) \bmod p = 1.$$

Now $F_{p_i,l}(X)$ assumes at most $n \cdot p_i$ different values. At most n of these are equal to 1 when taken modulo p_i. Denote the values of $F_{p_i,l}(X)$ that equal 1 when reduced modulo p_i as $v_1, v_2, \ldots, v_n$. For each v_j, $1 \leq j \leq n$, we place two LT gates in the first layer. Call them $G_j^{(1)}(X)$ and $G_j^{(2)}(X)$.

- The weights on the input wires of both $G_j^{(1)}$ and $G_j^{(2)}$ are set equal to the corresponding weights of $F_{p_i,l}(X)$.
- The thresholds of $G_j^{(1)}$ and $G_j^{(2)}$ are set to v_j and v_{j+1}, respectively.
- The weights on the output wires of $G_j^{(1)}$ and $G_j^{(2)}$ are set to 1 and -1, respectively.

Clearly,

$$\sum_{j=1}^{n} (G_j^{(1)}(X) + G_j^{(2)}(X)) = T_{p_i,l}(X).$$

To summarize, we have a total of $3 \cdot s \cdot n^2$ test elements, each of which requires $2 \cdot n$ LT gates to realize. Thus, we require $6 \cdot s \cdot n^3$ LT gates in total. Since $s = O(n \log n)$, we conclude that our construction has size $O(n^4 \log n)$.

3 Final Remarks

We have presented a simple, explicit $\widehat{LT}_2$ construction for COMPARISON, a function not in LT_1. The number of gates in our construction, $O(n^4 \log n)$, is a significant improvement on the general

bound of $O(n^{12}\log^{11} n)$ in [6]. An interesting direction of further research is to derive the exact size of the $\widehat{LT}_2$ implementation of COMPARISON using the approach presented in [3].

References

[1] N. Alon and J. Bruck. Explicit constructions of depth-2 majority circuits for comparison and addition. In *SIAM J. Discrete Math*, Vol. 7, No. 1, pp. 1–8, 1994.

[2] V. Bohossian and J. Bruck. On neural networks with minimal weights. In *Proc. of Neural Information Processing Systems*, No. 8, 1995.

[3] V. Bohossian and J. Bruck. Algebraic techniques for constructing minimal weight threshold functions. To appear in *SIAM J. Discrete Math.* Available at *http://www.paradise.caltech.edu/ETR.html.*

[4] M. Goldmann and M. Karpinski. Simulating threshold circuits by majority circuits. In *SIAM J. Discrete Math*, Vol. 7, No. 1, pp. 230–246, 1998.

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

[6] T. Hofmeister, A note on the simulation of exponential threshold weights, *CONCOON conference*, 1996.

[7] M. Muroga. *Threshold logic and its applications.* Wiley-Interscience, 1971.

[8] J. Myhill and W. H. Kautz. On the size of weights required for linear-input switching functions. *IRE Trans. Electronic Computers*, EC. 10, pp. 288–290, 1961.

[9] A.A. Razborov. On small depth threshold circuits. *Proc. 3rd Scandinavian Workshop on Algorithm Theory*, Lecture Notes in Computer Science 621, pp. 42–52, Springer-Verlag, 1992.

[10] 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.

[11] 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.

Appendix

Proof of Lemma 2.1:

We show that

$$X_1 \geq X_2 \quad \Rightarrow \quad \left\lfloor \frac{X_1}{2} \right\rfloor \geq \left\lfloor \frac{X_2}{2} \right\rfloor$$

There are four cases to consider.

1. X_1 is even, X_2 is even:

$$X_1 \geq X_2 \quad \Rightarrow \quad \frac{X_1}{2} \geq \frac{X_2}{2} \quad \Rightarrow \quad \left\lfloor \frac{X_1}{2} \right\rfloor \geq \left\lfloor \frac{X_2}{2} \right\rfloor .$$

2. X_1 is odd, X_2 is odd:

$$X_1 \geq X_2 \quad \Rightarrow \quad \frac{X_1 - 1}{2} \geq \frac{X_2 - 1}{2} \quad \Rightarrow \quad \left\lfloor \frac{X_1}{2} \right\rfloor \geq \left\lfloor \frac{X_2}{2} \right\rfloor .$$

3. X_1 is even, X_2 is odd:

$$X_1 \geq X_2 \quad \Rightarrow \quad \frac{X_1}{2} \geq \frac{X_2 - 1}{2} \quad \Rightarrow \quad \left\lfloor \frac{X_1}{2} \right\rfloor \geq \left\lfloor \frac{X_2}{2} \right\rfloor .$$

4. X_1 is odd, X_2 is even:

$$X_1 \geq X_2 \quad \Rightarrow \quad \frac{X_1 - 1}{2} \geq \frac{X_2}{2} \quad \Rightarrow \quad \left\lfloor \frac{X_1}{2} \right\rfloor \geq \left\lfloor \frac{X_2}{2} \right\rfloor .$$

□