Theoretical Computer Science 407 (2008) 474–487

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Exponential lower bounds on the size of constant-depth threshold circuits with small energy complexity

Kei Uchizawa [a,*], Eiji Takimoto [b]

[a] Graduate School of Information Sciences, Tohoku University, 6-3-09, Aramaki aza Aoba, Aoba-ku, Sendai 980-8579, Japan

[b] Department of Informatics, Faculty/Graduate School of Information Science and Electrical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan

ARTICLE INFO

Article history:
Received 10 August 2007
Received in revised form 24 July 2008
Accepted 24 July 2008
Communicated by A. Razborov

Keywords:
Threshold circuits
Neural networks
Communication complexity
Energy complexity

ABSTRACT

A complexity measure for threshold circuits, called the energy complexity, has been proposed to measure an amount of energy consumed during computation in the brain. Biological neurons need more energy to transmit a “spike” than not to transmit one, and hence the energy complexity of a threshold circuit is defined as the number of gates in the circuit that output “1” during computation. Since the firing activity of neurons in the brain is quite sparse, the following question arises: what Boolean functions can or cannot be computed by threshold circuits with small energy complexity. In the paper, we partially answer the question, that is, we show that there exists a trade-off among three complexity measures of threshold circuits: the energy complexity, size, and depth. The trade-off implies an exponential lower bound on the size of constant-depth threshold circuits with small energy complexity for a large class of Boolean functions.

1. Introduction

A biological neuron in the brain can be modeled by a threshold gate, and a neural network by a circuit composed of threshold gates, which has been extensively studied so far [10,13,16]. A neural network in the brain has an interesting feature of energy consumption: a biological neuron needs more energy to transmit a “spike” than not to transmit one [8]. The feature contrasts with electrical circuits, in which a gate outputting a signal “1” consumes almost the same amount of power as a gate outputting “0” (see [6,15] for example). Therefore, many neuroscientists consider that computations in the brain usually result from sparse firing activity of neurons [2,8,9,12]. We thus confront the following natural question from the point of view of computational complexity: what Boolean functions can or cannot be computed by reasonably small threshold circuits with few firing gates? A firing gate means a gate outputting “1”.

Uchizawa, Douglas and Maass formalize the question posed above by introducing a new complexity measure called the energy complexity of threshold circuits, and obtain sufficient conditions on functions that can be computed by threshold circuits with small energy complexity [17]. More precisely, they define the *energy complexity* of a circuit of n input variables as the maximum or expected number, taken over all input assignments of n variables, of firing gates in the circuit during computation, and they show that every threshold decision tree of polynomial size can be converted to a threshold circuit of polynomial size with energy complexity $O(\log n)$, where a threshold decision tree is a binary decision tree such that the classification rule at each internal node is defined by a threshold function. Thus, a threshold circuit of polynomial size with sparse activity has fairly large computational power. However, the threshold circuit converted from a decision

* Corresponding author. Tel.: +81 22 715 7163.
E-mail addresses: uchizawa@ecei.tohoku.ac.jp (K. Uchizawa), eiji@i.kyushu-u.ac.jp (E. Takimoto).

0304-3975/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.07.028

tree has a large depth. Hence, a question arises: does a threshold circuit of polynomial size with sparse activity have large computational power even if the depth of circuits is restricted to, say, a constant?

The paper answers the question above. We show that a threshold circuit of polynomial size with sparse firing activity does not have large computational power if the depth of circuit is restricted to a constant. More precisely, we investigate the relationship among three complexity measures of circuits: the maximum energy complexity, the size, and the depth. In particular, as a main theorem, we obtain an upper bound on the "communication complexity" of a Boolean function computed by a threshold circuit. Our bound is expressed in terms of the three complexity measures of a circuit and is monotonically increasing with respect to each of them. Since many Boolean functions have large communication complexity, our upper bound implies that there is a trade-off among the three complexity measures. The trade-off implies an exponential lower bound on the size of a constant-depth threshold circuit with small energy complexity for a large class of Boolean functions, including the Inner-Product. The bound is exponential in the number n of input variables. To the best of our knowledge, this is the first exponential lower bound for threshold circuits of *arbitrary* constant depth, although superpolynomial lower bounds have been obtained for threshold circuits of depth two or three under some restriction other than the small energy complexity [1,3–5,14]. An early version of the paper was presented in [18].

In Section 2, we define some terms and present some known results on threshold circuits and communication complexity, including Nisan's upper bound on the communication complexity of a Boolean function. In Section 3, we first present our main theorem, that is, an upper bound on the communication complexity of a Boolean function, and then present three corollaries of the theorem. In Section 4, we first present a lemma, which is one of our main results and implies that every Boolean function can be approximately computed by a "small" threshold circuit. Using the lemma together with Nisan's bound, we then prove the main theorem. In Section 5, we prove the lemma. Finally in Section 6 we conclude with some remarks.

2. Preliminaries

In Section 2.1 we define some terms on threshold circuits. In Section 2.2 we define some terms on communication complexity and present some known results.

2.1. Threshold circuits

In the paper, we consider a threshold gate having an arbitrary number p of inputs. For every input $\boldsymbol{z} = (z_1, z_2, \ldots, z_p) \in \{0, 1\}^p$, a *threshold gate* g (with *weights* $w_1, w_2, \ldots, w_p$ and a *threshold* t) computes

$$g(\boldsymbol{z}) = \mathrm{sign}\left(\sum_{i=1}^{p} w_i z_i - t\right) = \begin{cases} 1 & \text{if } \sum_{i=1}^{p} w_i z_i \geq t; \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathrm{sign}(z) = 1$ if $z \geq 0$ and $\mathrm{sign}(z) = 0$ if $z < 0$. We assume throughout the paper that the weights and threshold of every threshold gate are integers. A *threshold function* is similarly defined.

A *threshold circuit* C with n input variables is represented by a directed acyclic graph; the graph has exactly n nodes of in-degree 0, each associated with an input variable and called an *input node*; each of the other nodes represents a threshold gate. (See Fig. 1.) For an assignment $\boldsymbol{x} \in \{0, 1\}^n$ to the n input variables, the output of all gates in C are computed in topological order of the nodes in the directed acyclic graph. For a gate g in C, we denote by $g[\boldsymbol{x}]$ the output of g for an input $\boldsymbol{x}$ to circuit C (although the actual input to gate g will in general consist of just some variables from $\boldsymbol{x}$, and in addition, or even exclusively, of outputs of other gates in C). We say that a gate g is *fired* by $\boldsymbol{x}$ if $g[\boldsymbol{x}] = 1$. Since we consider only a threshold circuit that computes a Boolean function, one may assume without loss of generality that the circuit has exactly one gate of out-degree 0, called the *top gate*. We denote by $C(\boldsymbol{x})$ the output of the top gate of C for $\boldsymbol{x}$. We say that a threshold circuit C *computes* a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ if $C(\boldsymbol{x}) = f(\boldsymbol{x})$ for every input $\boldsymbol{x} \in \{0, 1\}^n$. Hereafter, we often use the term "function" to refer to a Boolean function.

The *size* of a threshold circuit C is the number of gates in C, and is denoted by $\mathrm{size}(C)$. One may assume without loss of generality that the in-degree of every gate is one or more in the directed acyclic graph. Therefore, for every gate g in C, there is a directed path to g from some input node. The *level* of a gate g in C is the length of the longest directed path to g from an input node. The *depth* of C is the level of the top gate of C. The *energy complexity*, or more precisely the *maximum energy complexity*, of C is the maximum number of gates fired by inputs $\boldsymbol{x}$ in C, where the maximum is taken over all the 2^n inputs $\boldsymbol{x} \in \{0, 1\}^n$ [17]. We say that a threshold circuit C has *small energy complexity* if the energy complexity e satisfies $e = n^{o(1)}$, that is, $e = n^{h(n)}$ for some function $h(n) = o(1)$. Thus $(\log n)^{\log \log n} = n^{o(1)}$, and $(\log n)^c = n^{o(1)}$ for any large number $c (\neq \infty)$, while $n^c \neq n^{o(1)}$ for any small number $c > 0$. In the paper, every "log" is to base 2.

Every Boolean function f can be computed by a threshold circuit C such that both the depth and the energy complexity of C are two but the size may be exponential in n. One can construct such a circuit C from the truth table of f. Each gate on level 1 corresponds to an input $\boldsymbol{x}$ such that $f(\boldsymbol{x}) = 1$, and the threshold of the gate is equal to the number of 1's in $\boldsymbol{x}$. For every input $\boldsymbol{x} \in \{0, 1\}^n$, at most one of the gates of level 1 in C is fired by $\boldsymbol{x}$ and the top gate is fired only when one of the gates of level 1 is fired. Thus, the energy complexity e of C is 2. Fig. 1 illustrates such a circuit for a function f of three variables x_1, x_2 and x_3.

x_1	x_2	x_3	$f(\boldsymbol{x})$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

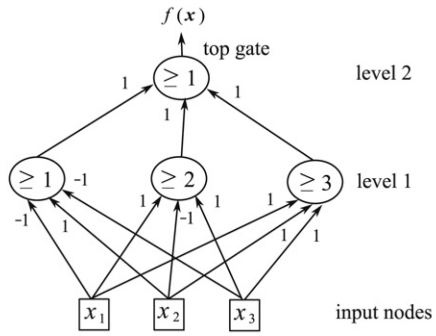

Fig. 1. Truth table of a Boolean function f, and a threshold circuit of size 4, depth 2, and energy complexity 2 for the function f.

2.2. Communication complexity

Consider a game of two players, say Alice and Bob. Assume that $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ is a Boolean function, and that Alice and Bob have unlimited computational power. Alice receives an input $\boldsymbol{x} \in \{0, 1\}^n$ and Bob receives an input $\boldsymbol{y} \in \{0, 1\}^n$. Alice and Bob try to compute the value $f(\boldsymbol{x}, \boldsymbol{y})$; they wish to exchange the least possible number of bits. The two players communicate with each other according to a randomized protocol. We assume that both Alice and Bob can use the same random bit string without communication. For each real number ϵ, $0 \leq \epsilon < 1/2$, the *communication complexity* of $f(\boldsymbol{x}, \boldsymbol{y})$, denoted by $R_\epsilon(f)$, is defined to be the maximum number of bits needed to be exchanged for the best randomized protocol to make the two players compute a value of $f(\boldsymbol{x}, \boldsymbol{y})$ correctly with probability $1 - \epsilon$ for every input $\boldsymbol{x} \times \boldsymbol{y} \in \{0, 1\}^n \times \{0, 1\}^n$; thus ϵ is the *error probability*. The maximum is taken over all input assignments and over all random bit strings, while the probability is taken over all random strings. Clearly, $R_\epsilon(f) \leq n + 1$. (For each real number δ, $0 < \delta \leq 1/2$, $R_{1/2-\delta}(f)$ is the maximum number of bits needed to be exchanged so that the two players compute a value of f correctly with probability $1/2 + \delta$.)

Nisan obtains the following upper bound on the communication complexity $R_\epsilon(f)$ of a function f computed by a threshold circuit [11], which we will use to prove our main theorem.

Lemma 1 (*[11]*). *If a Boolean function f of $2n$ variables can be computed by a threshold circuit of size s, then*

$$R_\epsilon(f) = O\left(s\left(\log n + \log \frac{s}{\epsilon}\right)\right)$$

for every number ϵ, $0 \leq \epsilon < 1/2$, that is, there is a randomized protocol to make two players compute a value of f with probability $1 - \epsilon$ by exchanging

$$O\left(s\left(\log n + \log \frac{s}{\epsilon}\right)\right)$$

bits.

Almost all functions f of $2n$ variables have large communication complexity:

$$R_{\frac{1}{2}-\delta}(f) = \Omega(n + \log \delta)$$

for every number δ, $0 < \delta \leq 1/2$ [7]. One of these functions is the Inner-Product of $2n$ variables [7], denoted by IP_n, which is defined as

$$IP_n(x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n) = x_1y_1 \oplus x_2y_2 \oplus \cdots \oplus x_ny_n$$

where $\oplus$ denotes the XOR function.

3. Main result

Our main result is the following theorem, which expresses an upper bound on the communication complexity of a function f in terms of the size s, depth d and energy complexity e of a threshold circuit computing f. In what follows, we may assume without loss of generality that $s, d, e \geq 1$.

Theorem 1. *If a Boolean function $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ can be computed by a threshold circuit of size s, depth d and energy complexity e, then*

$$R_{\frac{1}{2}-\delta}(f) = O((e + d)(\log n + (e + 1)^d \log s)) \tag{1}$$

for a number

$$\delta = \frac{1}{4s^{3(e+1)^d}}. \tag{2}$$

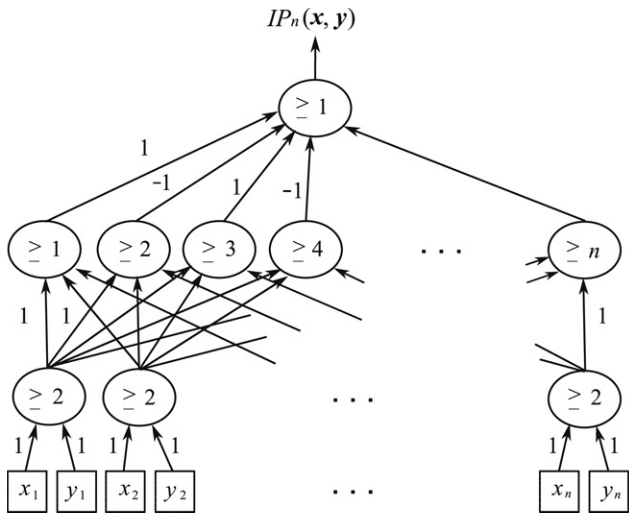

Fig. 2. Threshold circuit computing IP_n.

Many functions f have large communication complexity, that is, $R_{1/2-\delta}(f) = \Omega(n + \log \delta)$ for every number δ, $0 < \delta \leq 1/2$ [7]. Therefore, Theorem 1 implies that there is a trade-off among the size s, depth d, and energy complexity e of a threshold circuit computing f. In particular, we can derive from Theorem 1 an exponential lower bound on the size of threshold circuits for many functions, as in the following Corollaries 1–3.

Corollary 1. *Assume that f is a Boolean function of $2n$ variables and there is a number $\gamma > 0$ such that*

$$R_{\frac{1}{2}-\delta}(f) = \Omega(n^{\gamma} + \log \delta) \tag{3}$$

for every number δ, $0 < \delta \leq 1/2$. If f can be computed by a threshold circuit C of constant depth d with small energy complexity $e = n^{o(1)}$, that is, $e = n^{h(n)}$ for some function $h(n) = o(1)$, then

$$\mathrm{size}(C) = \exp(\Omega(n^{\gamma-(d+1)h(n)})) = \exp(\Omega(n^{\gamma-o(1)})).$$

Proof. Let $s = \mathrm{size}(C)$. Let δ be the number satisfying Eq. (2), then $0 < \delta < 1/2$. Eq. (3) for the number δ implies that

$$c(n^{\gamma} + \log \delta) \leq R_{\frac{1}{2}-\delta}(f) \tag{4}$$

for some constant c. On the other hand, Theorem 1 implies that

$$R_{\frac{1}{2}-\delta}(f) \quad \leq \quad c'(e+d)(\log n + (e+1)^d \log s) \tag{5}$$

for some constant c'. Substituting Eq. (2) into Eq. (4) and combining the resulting equation with Eq. (5), we have

$$\frac{c(n^{\gamma} - 2) - c'(e+d)\log n}{c'(e+d)(e+1)^d + 3c(e+1)^d} \leq \log s. \tag{6}$$

Clearly, the numerator of the left side of Eq. (6) is $\Omega(n^{\gamma})$. Since d is a constant and $e = n^{h(n)}$, the denominator of the left side of Eq. (6) is bounded above by $c''n^{(d+1)h(n)} (= n^{o(1)})$ for some constant c''. We thus have

$$s = \exp(\Omega(n^{\gamma-(d+1)h(n)}) = \exp(\Omega(n^{\gamma-o(1)})). \quad \square$$

Corollary 1 implies an exponential lower bound on size(C).

Since $R_{1/2-\delta}(IP_n) = \Omega(n + \log \delta)$ for every number δ, $0 < \delta \leq 1/2$ [7], we immediately have the following Corollary 2 from Corollary 1.

Corollary 2. *If a constant-depth threshold circuit C with small energy complexity $e = n^{o(1)}$ computes the Inner-Product of $2n$ variables, then*

$$\mathrm{size}(C) = \exp(\Omega(n^{1-o(1)})).$$

One can easily observe that IP_n can be computed by a threshold circuit of $2n + 1$ gates, depth 3 and energy complexity $2n + 1 (= O(n))$, as illustrated in Fig. 2. Therefore, the restriction $e = n^{o(1)}$ on the energy complexity in Corollary 1 cannot be relaxed to $e = O(n)$.

The size s of a threshold circuit is often greater than the number n of input variables. An exponential bound holds even if the energy e is bounded above in terms of s instead of n, say $e = (\log s)^c$ for a constant c, as follows.

Corollary 3. *Assume that f is a Boolean function of $2n$ variables and there is a number $\gamma > 0$ such that*

$$R_{\frac{1}{2}-\delta}(f) = \Omega(n^{\gamma} + \log \delta)$$

for every number δ, $0 < \delta \le 1/2$. If f can be computed by a threshold circuit C of constant depth d with energy complexity $e = (\log s)^c$ for a constant c, then

$$\mathrm{size}(C) \ge exp(\Omega(n^{\gamma'}))$$

for some constant γ', $0 < \gamma' < \gamma$.

Proof. Similar to the proof of Corollary 1. □

4. Proof of Theorem 1

In this section, we prove Theorem 1. We use Lemma 1 together with the following Lemma 2, which is one of our main results and will be proved in Section 5.

Lemma 2. *If a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ can be computed by a threshold circuit C of size s, depth d and energy complexity e, then f can be represented by a threshold function and a number k of threshold circuits $C_1, C_2, \ldots, C_k$, that is,*

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{k} w_i C_i(\boldsymbol{x})\right) \tag{7}$$

for every input $\boldsymbol{x} \in \{0, 1\}^n$ (see Fig. 3), where

(a)

$$k = 2(e+1)^{d-1}\binom{s-1}{e}; \tag{8}$$

(b) *the weights $w_1, w_2, \ldots, w_k$ are not zero, and satisfy*

$$\sum_{i=1}^{k} w_i = 0 \tag{9}$$

and

$$\sum_{i=1}^{k} |w_i| \le 2s^{3(e+1)^d}; \tag{10}$$

(c) *for every index i, $1 \le i \le k$,*

$$\mathrm{size}(C_i) \le e + 3d - 2; \tag{11}$$

and

(d) *for every input $\boldsymbol{x}$,*

$$\sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) \ne 0. \tag{12}$$

We are now ready to prove Theorem 1.

Proof (of Theorem 1). Let $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$. We denote by $f(\boldsymbol{x}, \boldsymbol{y})$ the value of f for $\boldsymbol{x} \in \{0, 1\}^n$ and $\boldsymbol{y} \in \{0, 1\}^n$. Assume that f can be computed by a threshold circuit C of size s, depth d and energy complexity e. Then Lemma 2 implies that f can be represented by a threshold function (with weights $w_1, w_2, \ldots, w_k$) and threshold circuits $C_1, C_2, \ldots, C_k$ satisfying Eqs. (7)–(12). Let

$$W = \sum_{i=1}^{k} |w_i| \tag{13}$$

and

$$s_{\max} = \max_{1 \le i \le k} \mathrm{size}(C_i). \tag{14}$$

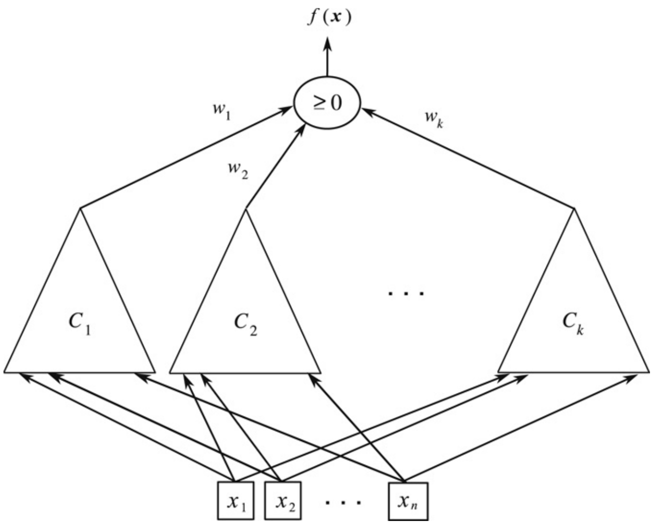

Fig. 3. A new circuit computing f.

Then, by Eqs. (10), (11) and (13), we have

$$W \leq 2s^{3(e+1)^d} \tag{15}$$

and

$$s_{\max} \leq e + 3d - 2. \tag{16}$$

We first show that it suffices to prove the following claim:

There is a randomized protocol to make Alice and Bob compute $f(\boldsymbol{x}, \boldsymbol{y})$ by exchanging

$$O(s_{\max}(\log n + \log(2Ws_{\max}))) \tag{17}$$

bits with error probability

$$\epsilon \leq \frac{1}{2} - \frac{1}{2W}. \tag{18}$$

Suppose that the claim above holds true. Then Eqs. (15)–(17) imply that the number of bits exchanged by the protocol is

$$\begin{aligned} O\left(s_{\max}(\log n + \log(2Ws_{\max}))\right) &= O\left((e+3d-2)(\log n + 2 + 3(e+1)^d \log s + \log(e+3d-2))\right) \\ &= O\left((e+d)(\log n + (e+1)^d \log s)\right). \end{aligned}$$

Eqs. (15) and (18) imply that the error probability is

$$\epsilon \leq \frac{1}{2} - \frac{1}{2W} \leq \frac{1}{2} - \frac{1}{4s^{3(e+1)^d}}.$$

Thus Theorem 1 holds if the claim above holds.

In what follows, we prove the claim above. Our protocol consists of the following three steps, and makes Alice and Bob compute a Boolean value $b \in \{0, 1\}$ as $f(\boldsymbol{x}, \boldsymbol{y})$ for every $\boldsymbol{x} \times \boldsymbol{y} \in \{0, 1\}^n \times \{0, 1\}^n$.

[Step 1]

Using a random bit string, Alice and Bob choose one of the k indices, $1 \leq i \leq k$, so that

$$\Pr[\text{Alice and Bob choose } i] = \frac{|w_i|}{W} \tag{19}$$

where the probability is taken over all random strings. Suppose that they choose an index j, $1 \leq j \leq k$, and let f_j be the function such that $f_j(\boldsymbol{x}, \boldsymbol{y}) = C_j(\boldsymbol{x}, \boldsymbol{y})$ for every $\boldsymbol{x} \times \boldsymbol{y} \in \{0, 1\}^n \times \{0, 1\}^n$.

[Step 2]

They apply Nisan's protocol in Lemma 1 for the function f_j with setting the error probability ϵ' as

$$\epsilon' = \frac{1}{2W}.$$

We denote by $N(\boldsymbol{x}, \boldsymbol{y})$ the Boolean value which Nisan's protocol makes Alice and Bob compute as $f_j(\boldsymbol{x}, \boldsymbol{y})$. Then

$$\Pr[N(\boldsymbol{x}, \boldsymbol{y}) \neq f_j(\boldsymbol{x}, \boldsymbol{y})] \leq \epsilon' = \frac{1}{2W}. \tag{20}$$

[Step 3]

Our protocol makes Alice and Bob compute, as $f(\boldsymbol{x}, \boldsymbol{y})$, a Boolean value $b \in \{0, 1\}$ such that

$$b = \begin{cases} N(\boldsymbol{x}, \boldsymbol{y}) & \text{if } w_j > 0; \\ \overline{N(\boldsymbol{x}, \boldsymbol{y})} & \text{if } w_j < 0, \end{cases} \tag{21}$$

where $\overline{N(\boldsymbol{x}, \boldsymbol{y})}$ is the negation of the Boolean value $N(\boldsymbol{x}, \boldsymbol{y})$.

Lemma 1, Eqs. (14) and (20) imply that the number of bits exchanged between Alice and Bob by our protocol above is

$$O\left(s_{\max}(\log n + \log(2Ws_{\max}))\right).$$

We have thus verified Eq. (17).

We then prove Eq. (18), that is,

$$\epsilon = \Pr[b \neq f(\boldsymbol{x}, \boldsymbol{y})] \leq \frac{1}{2} - \frac{1}{2W}.$$

For an input $\boldsymbol{x} \times \boldsymbol{y}$, we partition the set $\{1, 2, \dots, k\}$ of indices into four subsets I_0^+, I_1^+, I_0^- and I_1^-, as follows:

$$I_0^+ = \{i \in \{1, 2, \dots, k\} \mid C_i(\boldsymbol{x}, \boldsymbol{y}) = 0 \text{ and } w_i > 0\};$$
$$I_1^+ = \{i \in \{1, 2, \dots, k\} \mid C_i(\boldsymbol{x}, \boldsymbol{y}) = 1 \text{ and } w_i > 0\};$$
$$I_0^- = \{i \in \{1, 2, \dots, k\} \mid C_i(\boldsymbol{x}, \boldsymbol{y}) = 0 \text{ and } w_i < 0\};$$

and

$$I_1^- = \{i \in \{1, 2, \dots, k\} \mid C_i(\boldsymbol{x}, \boldsymbol{y}) = 1 \text{ and } w_i < 0\}.$$

Clearly

$$W = \sum_{i \in I_0^+} w_i + \sum_{i \in I_1^+} w_i + \sum_{i \in I_0^-} |w_i| + \sum_{i \in I_1^-} |w_i|. \tag{22}$$

We define a Boolean value $b' \in \{0, 1\}$ as

$$b' = \begin{cases} f_j(\boldsymbol{x}, \boldsymbol{y}) & \text{if } w_j > 0; \\ \overline{f_j(\boldsymbol{x}, \boldsymbol{y})} & \text{if } w_j < 0. \end{cases} \tag{23}$$

Eqs. (21) and (23) imply that if $b \neq f(\boldsymbol{x}, \boldsymbol{y})$ then either $b' \neq f(\boldsymbol{x}, \boldsymbol{y})$ or $N(\boldsymbol{x}, \boldsymbol{y}) \neq f_j(\boldsymbol{x}, \boldsymbol{y})$. Therefore, by Eq. (20) we have

$$\epsilon = \Pr[b \neq f(\boldsymbol{x}, \boldsymbol{y})] \leq \Pr[b' \neq f(\boldsymbol{x}, \boldsymbol{y})] + \epsilon'. \tag{24}$$

If

$$\Pr[b' \neq f(\boldsymbol{x}, \boldsymbol{y})] \leq \frac{1}{2} - \frac{1}{W}, \tag{25}$$

then by Eqs. (20) and (24) we have

$$\begin{aligned} \epsilon &\leq \left(\frac{1}{2} - \frac{1}{W}\right) + \frac{1}{2W} \\ &= \frac{1}{2} - \frac{1}{2W}, \end{aligned}$$

and hence Eq. (18) holds. Thus we shall prove Eq. (25).

There are two cases to consider.

Case 1: $f(\boldsymbol{x}, \boldsymbol{y}) = 1$.

In this case we have

$$\Pr[b' \neq f(\boldsymbol{x}, \boldsymbol{y})] = \Pr[b' = 0].$$

Eq. (23) implies that $b' = 0$ if and only if either $f_j(\boldsymbol{x}, \boldsymbol{y}) = 0$ and $w_j > 0$ or $f_j(\boldsymbol{x}, \boldsymbol{y}) = 1$ and $w_j < 0$, and hence $b' = 0$ if and only if $j \in I_0^+ \cup I_1^-$. Therefore by Eq. (19) we have

$$\Pr[b' \neq f(\boldsymbol{x}, \boldsymbol{y})] = \frac{\sum_{i \in I_0^+} w_i + \sum_{i \in I_1^-} |w_i|}{W}. \tag{26}$$

Using Eq. (22), we have

$$\sum_{i\in I_0^+} w_i + \sum_{i\in I_1^-} |w_i| = \frac{W}{2} - \sum_{i\in I_1^+} \frac{w_i}{2} - \sum_{i\in I_0^-} \frac{|w_i|}{2} + \sum_{i\in I_0^+} \frac{w_i}{2} + \sum_{i\in I_1^-} \frac{|w_i|}{2}. \tag{27}$$

Using Eqs. (9) and (27) we have

$$\begin{aligned}\sum_{i\in I_0^+} w_i + \sum_{i\in I_1^-} |w_i| &= \frac{W}{2} - \sum_{i\in I_1^+} \frac{w_i}{2} - \sum_{i\in I_0^-} \frac{|w_i|}{2} + \sum_{i\in I_0^+} \frac{w_i}{2} + \sum_{i\in I_1^-} \frac{|w_i|}{2} - \sum_{i=1}^{k} \frac{w_i}{2} \\ &= \frac{W}{2} - \left(\sum_{i\in I_1^+} w_i - \sum_{i\in I_1^-} |w_i|\right). \end{aligned} \tag{28}$$

Since $f(\boldsymbol{x}, \boldsymbol{y}) = 1$, by Eqs. (7) and (12) we have

$$1 \le \sum_{i=1}^{k} w_i C_i(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i\in I_1^+} w_i - \sum_{i\in I_1^-} |w_i|. \tag{29}$$

By Eqs. (26), (28) and (29) we have

$$\begin{aligned}\Pr[b' \ne f(\boldsymbol{x}, \boldsymbol{y})] &= \frac{1}{W}\left(\frac{W}{2} - \left(\sum_{i\in I_1^+} w_i - \sum_{i\in I_1^-} |w_i|\right)\right) \\ &\le \frac{1}{2} - \frac{1}{W}.\end{aligned}$$

We have thus verified Eq. (25) for Case 1.

Case 2: $f(\boldsymbol{x}, \boldsymbol{y}) = 0$.

Similarly as in Eq. (26), we have

$$\Pr[b' \ne f(\boldsymbol{x}, \boldsymbol{y})] = \frac{\sum_{i\in I_1^+} w_i + \sum_{i\in I_0^-} |w_i|}{W}. \tag{30}$$

Similarly as in Eq. (28), we have

$$\sum_{i\in I_1^+} w_i + \sum_{i\in I_0^-} |w_i| = \frac{W}{2} + \left(\sum_{i\in I_1^+} w_i - \sum_{i\in I_1^-} |w_i|\right). \tag{31}$$

Since $f(\boldsymbol{x}, \boldsymbol{y}) = 0$, by Eqs. (7) and (12) we have

$$\sum_{i=1}^{k} w_i C_i(\boldsymbol{x}, \boldsymbol{y}) = \sum_{i\in I_1^+} w_i - \sum_{i\in I_1^-} |w_i| \le -1. \tag{32}$$

Therefore by Eqs. (30)–(32) we have

$$\begin{aligned}\Pr[b' \ne f(\boldsymbol{x}, \boldsymbol{y})] &= \frac{1}{W}\left(\frac{W}{2} + \left(\sum_{i\in I_1^+} w_i - \sum_{i\in I_1^-} |w_i|\right)\right) \\ &\le \frac{1}{2} - \frac{1}{W}.\end{aligned}$$

We have thus verified Eq. (25) for Case 2. $\square$

5. Proof of Lemma 2

In this section we prove Lemma 2. Assume that a function $f : \{0, 1\}^n \to \{0, 1\}$ can be computed by a threshold circuit C of size s, depth d, and energy complexity e. Let G be the set of all gates in C, and let $g \in G$ be the top gate of C.

We first prove Lemma 2 for the case where $d = 1$. In this case, $e \le s = 1$, $k = 2$ and $G = \{g\}$. We construct two circuits C_1 and C_2. The circuit C_1 is the same as C. The circuit C_2 is a circuit which outputs the negation $\overline{C(\boldsymbol{x})}$ of the Boolean value $C(\boldsymbol{x})$ for every input $\boldsymbol{x} \in \{0, 1\}^n$. We can construct the circuit C_2 from C by replacing g by a new gate; if g has weights $w_{g,i}$, $1 \le i \le n$, and a threshold t_g, then the new gate has weights $-2w_{g,i}$, $1 \le i \le n$, and a threshold $-2t_g + 1$. Clearly $C_2(\boldsymbol{x}) = \overline{C(\boldsymbol{x})}$ for every input $\boldsymbol{x}$. Let $w_1 = 1$ and $w_2 = -1$. Then clearly Eqs. (7)–(12) hold.

In what follows, we prove Lemma 2 for the case where $d \ge 2$. In this case, $s \ge d \ge 2$. In Section 5.1, we construct the k circuits $C_1, C_2, \ldots, C_k$ from C, and prove Eq. (11). In Section 5.2, we decide the k weights $w_1, w_2, \ldots, w_k$, and prove Eqs. (9) and (10). In Section 5.3, we prove Eqs. (7) and (12).

5.1. Circuits $C_1, C_2, \ldots, C_k$

In this section, we show how to construct circuits $C_1, C_2, \ldots, C_k$, and prove Eq. (11). The construction consists of the following Step 1 and Step 2.

[Step 1]

Let g be the top gate of C. Define a family $\mathbb{S}$ of subsets of G as follows:

$$\mathbb{S} = \{S \subseteq G \mid g \in S, \ |S| = e + 1\}.$$

Let

$$m = |\mathbb{S}|,$$

then

$$m = \binom{s-1}{e}.$$

(If $e = s$, then let $\mathbb{S} = \{G\}$ and we define $m = \binom{s-1}{s} = 1$.) We construct two circuits C_S^+ and C_S^- for each set $S \in \mathbb{S}$, and hence we construct $2m$ circuits in total in Step 1.

C_S^+ is a sub-circuit of C induced by all the gates in S and the n input nodes. More precisely, C_S^+ consists of all the gates in S and the n input nodes together with the wires connecting them in C, and the weights and threshold of every gate in C_S^+ are the same as those of the gate in C. Thus $\mathrm{size}(C_S^+) = e + 1$. (If $e = s$, then $\mathrm{size}(C_S^+) = e$.) We denote by g_S the gate g contained in C_S^+. Clearly g_S has out-degree zero in C_S^+. Circuit C_S^+ may have two or more gates having out-degree zero. However, we regard g_S as the top gate of circuit C_S^+, and hence $C_S^+(\boldsymbol{x}) = g_S[\boldsymbol{x}]$ for every input $\boldsymbol{x} \in \{0, 1\}^n$. If S contains all the gates that are fired by an input $\boldsymbol{x} \in \{0, 1\}^n$ in C, then

$$C_S^+(\boldsymbol{x}) = C(\boldsymbol{x}). \tag{33}$$

Note that, for every gate $h \in S$, h is fired by $\boldsymbol{x}$ in C_S^+ if and only if h is fired by $\boldsymbol{x}$ in C. For every input $\boldsymbol{x} \in \{0, 1\}^n$, at least one of the m sets in $\mathbb{S}$ contains all the gates fired by $\boldsymbol{x}$ in C, because at most e gates of C are fired by $\boldsymbol{x}$.

C_S^- is a circuit which outputs the negation $\overline{C_S^+(\boldsymbol{x})}$ of the Boolean value $C_S^+(\boldsymbol{x})$ for every input $\boldsymbol{x}$. The circuit C_S^- is obtained from C_S^+ by replacing the top gate g_S by a new top gate which outputs the negation $\overline{g_S[\boldsymbol{x}]}$ of $g_S[\boldsymbol{x}]$ for every $\boldsymbol{x} \in \{0, 1\}^n$. The new top gate can be obtained from g_S as follows: if g_S has weights $w_{S,i}$, $1 \le i \le p_S$, and a threshold t_S, then the new top gate has weights $-2w_{S,i}$, $1 \le i \le p_S$, and a threshold $-2t_S + 1$. Clearly, $C_S^-(\boldsymbol{x}) = \overline{C_S^+(\boldsymbol{x})}$ for every input $\boldsymbol{x}$. Furthermore

$$C_S^-(\boldsymbol{x}) = \overline{C(\boldsymbol{x})} \tag{34}$$

for an input $\boldsymbol{x}$ if S contains all the gates that are fired by $\boldsymbol{x}$ in C.

[Step 2]

In Step 2, we construct k circuits $C_1, C_2, \ldots, C_k$ from the $2m$ circuits C_S^+ and C_S^- constructed in Step 1.

Let $U = \{0, 1, \ldots, e\}^{d-1}$, then $|U| = (e+1)^{d-1}$. For every pair $(S, \boldsymbol{u}) \in \mathbb{S} \times U$, we construct a circuit $C_{S,\boldsymbol{u}}^+$ from C_S^+, as follows. Let $\boldsymbol{u} = (u_1, u_2, \ldots, u_{d-1}) \in U$, then $0 \le u_i \le e$ for every i, $1 \le i \le d-1$. For each index l, $1 \le l \le d$, we denote by K_l the set of all gates of C having level l in C. Then $K_d = \{g\}$. The inputs of a gate in K_l come from gates in $K_1 \cup K_2 \cup \cdots K_{l-1}$ and/or input nodes. For each l, $1 \le l \le d-1$, we add three gates g_l^1, g_l^2 and g_l^3 to C_S^+, as illustrated in Fig. 4. The gate g_l^1 has exactly $|K_l \cap S|$ inputs, all the weights are 1, and the threshold of g_l^1 is u_l. The output of each gate g' in $K_l \cap S$ is connected to one of the $|K_l \cap S|$ inputs of g_l^1. Thus g_l^1 computes

$$g_l^1[\boldsymbol{x}] = \mathrm{sign}\left(\sum_{g' \in K_l \cap S} g'[\boldsymbol{x}] - u_l\right).$$

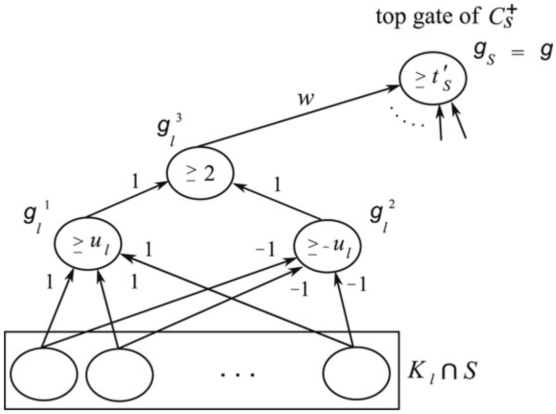

Fig. 4. Three gates g_l^1, g_l^2, g_l^3 added to C_S^+ for level l, and the top gate g_S of C_S^+.

Hence, the gate g_l^1 outputs 1 if and only if at least u_l gates in $K_l \cap S$ are fired by $\boldsymbol{x}$ in C_S^+. We then add a gate g_l^2 to C_S^+ which computes

$$g_l^2[\boldsymbol{x}] = \mathrm{sign}\left(-\sum_{g' \in K_l \cap S} g'[\boldsymbol{x}] + u_l\right).$$

The gate g_l^2 outputs 1 if and only if at most u_l gates in $K_l \cap S$ are fired by $\boldsymbol{x}$ in C_S^+. We finally add a gate g_l^3 having two inputs connected to the outputs of g_l^1 and g_l^2, which computes

$$g_l^3[\boldsymbol{x}] = \mathrm{sign}(g_l^1[\boldsymbol{x}] + g_l^2[\boldsymbol{x}] - 2).$$

Clearly, g_l^3 computes AND of Boolean values $g_l^1[\boldsymbol{x}]$ and $g_l^2[\boldsymbol{x}]$. Therefore, g_l^3 outputs 1 if and only if exactly u_l gates in $K_l \cap S$ are fired by $\boldsymbol{x}$ in C_S^+. Let the top gate $g_S(= g)$ of C_S^+ have weights $w_{S,1}, w_{S,2}, \ldots, w_{S,p_S}$ and a threshold t_S. Let w be an arbitrary integer such that

$$w > \sum_{i=1}^{p_S} |w_{S,i}| + |t_S|.$$

We connect the outputs of all the $d-1$ gates g_l^3, $1 \leq l \leq d-1$, to g_S with weight w, and replace the threshold t_S of g_S with a new threshold t'_S

$$t'_S = t_S + (d-1)w.$$

The resulting circuit is $C_{S,\boldsymbol{u}}^+$. Clearly, $C_{S,\boldsymbol{u}}^+(\boldsymbol{x}) = 1$ if and only if $C_S^+(\boldsymbol{x}) = 1$ and exactly u_l gates in $K_l \cap S$ are fired by $\boldsymbol{x}$ in C_S^+ for every l, $1 \leq l \leq d-1$.

For a set $S \in \mathbb{S}$ and an input $\boldsymbol{x} \in \{0,1\}^n$, we say that $\boldsymbol{u} = (u_1, u_2, \ldots, u_{d-1}) \in U$ is the *signature of* $\boldsymbol{x}$ for circuit C_S^+ if, for each l, $1 \leq l \leq d-1$, exactly u_l gates in $K_l \cap S$ are fired by $\boldsymbol{x}$ in C_S^+. Then one can easily know that the circuit $C_{S,\boldsymbol{u}}^+$ computes

$$C_{S,\boldsymbol{u}}^+(\boldsymbol{x}) = \begin{cases} C_S^+(\boldsymbol{x}) & \text{if } \boldsymbol{u} \text{ is the signature of } \boldsymbol{x} \text{ for } C_S^+; \\ 0 & \text{otherwise.} \end{cases} \tag{35}$$

Similarly as above, for each pair $(S, \boldsymbol{u}) \in \mathbb{S} \times U$, we construct from C_S^- a circuit $C_{S,\boldsymbol{u}}^-$ which computes

$$C_{S,\boldsymbol{u}}^-(\boldsymbol{x}) = \begin{cases} \overline{C_S^+(\boldsymbol{x})} & \text{if } \boldsymbol{u} \text{ is the signature of } \boldsymbol{x} \text{ for } C_S^-; \\ 0 & \text{otherwise.} \end{cases} \tag{36}$$

It should be noted that exactly u_l gates in $K_l \cap S$ are fired by $\boldsymbol{x}$ in C_S^+ for each l, $1 \leq l \leq d-1$, if and only if exactly u_l gates in $K_l \cap S$ are fired by $\boldsymbol{x}$ in C_S^-. We can thus denote by $\boldsymbol{u}_S$ both the signature of $\boldsymbol{x}$ for C_S^+ and that for C_S^-.

In Steps 1 and 2 we have constructed two circuits $C_{S,\boldsymbol{u}}^+$ and $C_{S,\boldsymbol{u}}^-$ for each pair $(S, \boldsymbol{u}) \in \mathbb{S} \times U$. These are the k circuits $C_1, C_2, \ldots, C_k$ that we are constructing. Clearly, for every i, $1 \leq i \leq k$,

$$\mathrm{size}(C_i) \leq (e+1) + 3(d-1) = e + 3d - 2.$$

(If $e = s$, then $\mathrm{size}(C_i) = e + 3d - 3$.) We have thus verified Eq. (11).

5.2. Weights $w_1, w_2, \ldots, w_k$

In this section, we decide the weights $w_1, w_2, \ldots, w_k$ and prove Eqs. (9) and (10).

For each $\boldsymbol{u} = (u_1, u_2, \ldots, u_{d-1}) \in U$, we denote by $[\boldsymbol{u}]$ the integer whose $(e+1)$-ary representation is $\boldsymbol{u}$:

$$[\boldsymbol{u}] = \sum_{i=1}^{d-1} u_i (e+1)^{d-1-i}.$$

For each pair $(S, \boldsymbol{u})$, let

$$w_{S,\boldsymbol{u}} = m^{[\boldsymbol{u}]} \tag{37}$$

where $m = \binom{s-1}{e}$.

For each $\boldsymbol{u} = (u_1, u_2, \ldots, u_{d-1}) \in U$, we decide the weight for the circuit $C^+_{S,\boldsymbol{u}}$ as $w_{S,\boldsymbol{u}}$ and for the circuit $C^-_{S,\boldsymbol{u}}$ as $-w_{S,\boldsymbol{u}}$. All these k weights $w_1, w_2, \ldots, w_k$ are not zero, and

$$\sum_{i=1}^{k} w_i = \sum_{S\in\mathbb{S}} \sum_{\boldsymbol{u}\in U} \left(w_{S,\boldsymbol{u}} - w_{S,\boldsymbol{u}}\right) = 0,$$

and hence Eq. (9) holds. Clearly,

$$m^{[\boldsymbol{u}]} \le m^{(e+1)^{d-1}}$$

and

$$m = \binom{s-1}{e} \le s^e.$$

We thus have

$$\begin{aligned}
\sum_{i=1}^{k} |w_i| &= \sum_{S\in\mathbb{S}} \sum_{\boldsymbol{u}\in U} (w_{S,\boldsymbol{u}} + w_{S,\boldsymbol{u}}) \\
&= 2\binom{s-1}{e} \sum_{\boldsymbol{u}\in U} m^{[\boldsymbol{u}]} \\
&\le 2s^e m^{(e+1)^{d-1}} |U| \\
&\le 2s^e s^{e(e+1)^{d-1}} (s+1)^{d-1} \\
&\le 2s^{3(e+1)^d}.
\end{aligned}$$

Note that $2 \le d \le s$ and $1 \le e$. We have thus proved Eq. (10).

5.3. Proof of Eqs. (7) and (12)

In this section we prove Eqs. (7) and (12), that is,

$$f(\boldsymbol{x}) = \mathrm{sign}\left(\sum_{S\in\mathbb{S}} \sum_{u\in U} \left(w_{S,\boldsymbol{u}} C^+_{S,\boldsymbol{u}}(\boldsymbol{x}) - w_{S,\boldsymbol{u}} C^-_{S,\boldsymbol{u}}(\boldsymbol{x})\right)\right)$$

and

$$\sum_{S\in\mathbb{S}} \sum_{u\in U} \left(w_{S,\boldsymbol{u}} C^+_{S,\boldsymbol{u}}(\boldsymbol{x}) - w_{S,\boldsymbol{u}} C^-_{S,\boldsymbol{u}}(\boldsymbol{x})\right) \ne 0$$

for every $\boldsymbol{x} \in \{0, 1\}^n$.

Consider a fixed input $\boldsymbol{x} \in \{0, 1\}^n$ till the end of the proof of Lemma 3.

For each set $S \in \mathbb{S}$, Eqs. (35) and (36) imply that

$$C^+_{S,\boldsymbol{u}_S}(\boldsymbol{x}) = C^+_S(\boldsymbol{x}) \tag{38}$$

and

$$C^-_{S,\boldsymbol{u}_S}(\boldsymbol{x}) = \overline{C^+_S(\boldsymbol{x})} \tag{39}$$

where $\boldsymbol{u}_S$ is the signature of $\boldsymbol{x}$ for C^+_S. On the other hand

$$C^+_{S,\boldsymbol{u}}(\boldsymbol{x}) = C^-_{S,\boldsymbol{u}}(\boldsymbol{x}) = 0 \tag{40}$$

for every $\boldsymbol{u} \in U \backslash \{\boldsymbol{u}_S\}$. Eqs. (38)–(40) imply that exactly one of the two circuits $C^+_{S,\boldsymbol{u}_S}$ and $C^-_{S,\boldsymbol{u}_S}$ outputs 1 for $\boldsymbol{x}$, and hence exactly one of the $2|U|$ circuits $C^+_{S,\boldsymbol{u}}$ and $C^-_{S,\boldsymbol{u}}$, $\boldsymbol{u} \in U$, outputs 1 for $\boldsymbol{x}$.

Let

$$U_{\mathrm{sig}} = \{\boldsymbol{u}_S \mid S \in \mathbb{S} \text{ and } \boldsymbol{u}_S \text{ is the signature of } \boldsymbol{x} \text{ for } C^+_S\},$$

and let $\boldsymbol{u}^* = (u^*_1, u^*_2, \ldots, u^*_{d-1})$ be the lexicographically largest signature in U_{sig}. Then $\sum_{l=1}^{d-1} u^*_l \le e$, since $|\cup_{1\le l\le d-1} S \cap K_l| \le e$ for every $S \in \mathbb{S}$. Let

$$\mathbb{S}^* = \{S \in \mathbb{S} \mid \boldsymbol{u}_S = \boldsymbol{u}^*\}.$$

We then have the following lemma.

Lemma 3. *For every set $S \in \mathbb{S}^*$,*

$$C^+_{S,\boldsymbol{u}^*}(\boldsymbol{x}) = C(\boldsymbol{x}).$$

Proof. Let $S \in \mathbb{S}^*$. Let F be the set of all the gates of C that are fired by $\boldsymbol{x}$ in C. By Eqs. (33) and (38), it suffices to prove that $F \subseteq S$.

Let $n_l = |F \cap K_l|$ for each index l, $1 \le l \le d-1$. We will prove by induction on l, $1 \le l \le d-1$, that

$$n_l = u^*_l$$

and

$$F \cap (K_1 \cup K_2 \cup \cdots \cup K_l) \subseteq S. \tag{41}$$

If Eq. (41) holds for $l = d-1$, then $F \subseteq S$, because S contains the top gate g of C on level d.

1° For the basis of induction, we prove $n_1 = u^*_1$ and $F \cap K_1 \subseteq S$. Obviously $n_1 = |F \cap K_1| \le e$. For every gate $h \in K_1 \cap S$, h is fired by $\boldsymbol{x}$ in C^+_S if and only if h is fired by $\boldsymbol{x}$ in C, since the inputs of h come only from the n input nodes. Since $\boldsymbol{u}^*(= \boldsymbol{u}_S)$ is the lexicographically largest in U_{sig}, we have $n_1 = u^*_1$ and $F \cap K_1 \subseteq S$.

2° For the induction hypothesis, we assume that $2 \le l \le d-1$, $F \cap (K_1 \cup K_2 \cup \cdots \cup K_{l-1}) \subseteq S$, and $n_i = u^*_i$ for each index i, $1 \le i \le l-1$.

3° Since $F \cap (K_1 \cup K_2 \cup \cdots \cup K_{l-1}) \subseteq S$, one can observe that, for every gate $h \in K_l \cap S$, h is fired by $\boldsymbol{x}$ in C^+_S if and only if h is fired by $\boldsymbol{x}$ in C. It should be noted that, both in C and C^+_S, the inputs of h come only from some of the gates in $K_1 \cup K_2 \cup \cdots \cup K_{l-1}$ and the n input nodes, although the topological order of the gates in the directed acyclic graph corresponding to the circuit C^+_S is not necessarily consistent with that of circuit C. Since $n_1 = u^*_1, n_2 = u^*_2, \ldots, n_{l-1} = u^*_{l-1}$, $\sum_{i=1}^{l} n_i \le e$ and $\boldsymbol{u}^*(= \boldsymbol{u}_S)$ is the lexicographically largest, we have $n_l = u^*_l$ and $F \cap K_l \subseteq S$. Hence $F \cap (K_1 \cup K_2 \cup \cdots \cup K_l) \subseteq S$. □

We are now ready to prove Eqs. (7) and (12). Note that $f(\boldsymbol{x}) = C(\boldsymbol{x})$ for every input $\boldsymbol{x} \in \{0,1\}^n$, since C computes f. There are two cases to consider.

Case 1: $f(\boldsymbol{x}) = C(\boldsymbol{x}) = 1$.

If

$$\sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) > 0, \tag{42}$$

then we have

$$f(\boldsymbol{x}) = 1 = \mathrm{sign}\left(\sum_{i=1}^{k} w_i C_i(\boldsymbol{x})\right)$$

and

$$\sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) \ne 0,$$

and hence Eqs. (7) and (12) hold. Thus it suffices to prove Eq. (42).

By Eq. (40) we have

$$\begin{aligned}\sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) &= \sum_{S\in\mathbb{S}} \sum_{\boldsymbol{u}\in U} \left(w_{S,\boldsymbol{u}} C^+_{S,\boldsymbol{u}}(\boldsymbol{x}) - w_{S,\boldsymbol{u}_S} C^-_{S,\boldsymbol{u}_S}(\boldsymbol{x})\right) \\ &= \sum_{S\in\mathbb{S}} \left(w_{S,\boldsymbol{u}_S} C^+_{S,\boldsymbol{u}_S}(\boldsymbol{x}) - w_{S,\boldsymbol{u}_S} C^-_{S,\boldsymbol{u}_S}(\boldsymbol{x})\right).\end{aligned} \tag{43}$$

Since $C(\boldsymbol{x}) = 1$, Lemma 3 implies that $C^{+}_{S,\boldsymbol{u}^*}(\boldsymbol{x}) = C(\boldsymbol{x}) = 1$ for every $S \in \mathbb{S}^*$, and hence by Eqs. (38) and (39) we have $C^{-}_{S,\boldsymbol{u}^*}(\boldsymbol{x}) = \overline{C^{+}_{S}(\boldsymbol{x})} = \overline{C^{+}_{S,\boldsymbol{u}^*}(\boldsymbol{x})} = \overline{C(\boldsymbol{x})} = 0$ for every $S \in \mathbb{S}^*$. Therefore, we have from Eq. (43)

$$\sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) = \sum_{S\in\mathbb{S}^*} w_{S,\boldsymbol{u}^*} + \sum_{S\in\mathbb{S}\setminus\mathbb{S}^*} \left(w_{S,\boldsymbol{u}_S} C^{+}_{S,\boldsymbol{u}_S}(\boldsymbol{x}) - w_{S,\boldsymbol{u}_S} C^{-}_{S,\boldsymbol{u}_S}(\boldsymbol{x}) \right). \tag{44}$$

Since $-w_{S,\boldsymbol{u}_S} < 0 < w_{S,\boldsymbol{u}_S}$ and exactly one of $C^{+}_{S,\boldsymbol{u}_S}(\boldsymbol{x})$ and $C^{-}_{S,\boldsymbol{u}_S}(\boldsymbol{x})$ is 1, we have from Eq. (44)

$$\sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) \geq \sum_{S\in\mathbb{S}^*} w_{S,\boldsymbol{u}^*} - \sum_{S\in\mathbb{S}\setminus\mathbb{S}^*} w_{S,\boldsymbol{u}_S}. \tag{45}$$

Let S^* be an arbitrary set in $\mathbb{S}^*$, then by Eq. (37) we have

$$\begin{aligned} \sum_{S\in\mathbb{S}^*} w_{S,\boldsymbol{u}^*} &\geq w_{S^*,\boldsymbol{u}^*} \\ &= m^{[\boldsymbol{u}^*]} \end{aligned} \tag{46}$$

and

$$-\sum_{S\in\mathbb{S}\setminus\mathbb{S}^*} w_{S,\boldsymbol{u}_S} = -\sum_{S\in\mathbb{S}\setminus\mathbb{S}^*} m^{[\boldsymbol{u}_S]}. \tag{47}$$

By Eqs. (45)–(47) we have

$$\sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) \geq m^{[\boldsymbol{u}^*]} - \sum_{S\in\mathbb{S}\setminus\mathbb{S}^*} m^{[\boldsymbol{u}_S]}. \tag{48}$$

Since $\boldsymbol{u}^*$ is the lexicographically largest in U_{sig}, we have $[\boldsymbol{u}^*] - 1 \geq [\boldsymbol{u}_S]$ for every $S \in \mathbb{S}\setminus\mathbb{S}^*$. Therefore, by Eq. (48) we have

$$\begin{aligned} \sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) &\geq m^{[\boldsymbol{u}^*]} - (m-1)m^{[\boldsymbol{u}^*]-1} \\ &= m^{[\boldsymbol{u}^*]-1} \\ &> 0, \end{aligned}$$

and hence Eq. (42) holds.

Case 2: $f(\boldsymbol{x}) = C(\boldsymbol{x}) = 0$.

In this case, similarly as in Case 1, we have

$$\begin{aligned} \sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) &\leq -\sum_{S\in\mathbb{S}^*} w_{S,\boldsymbol{u}^*} + \sum_{S\in\mathbb{S}\setminus\mathbb{S}^*} w_{S,\boldsymbol{u}_S} \\ &\leq -m^{[\boldsymbol{u}^*]} + \sum_{S\in\mathbb{S}\setminus\mathbb{S}^*} m^{[\boldsymbol{u}_S]}. \end{aligned} \tag{49}$$

By Eq. (49) we have

$$\begin{aligned} \sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) &\leq -m^{[\boldsymbol{u}^*]} + (m-1)m^{[\boldsymbol{u}^*]-1} \\ &= -m^{[\boldsymbol{u}^*]-1} \\ &< 0 \end{aligned}$$

and hence

$$f(\boldsymbol{x}) = 0 = \mathrm{sign}\left(\sum_{i=1}^{k} w_i C_i(\boldsymbol{x}) \right).$$

We have thus proved Eqs. (7) and (12) for Case 2.

6. Conclusions

In the paper, we showed that there exists a trade-off among three complexity measures of threshold circuits: the size, depth and energy complexity. Our trade-off implies exponential lower bounds on the size of constant-depth threshold circuits with small energy complexity for a large class of Boolean functions, including the Inner-Product IP_n. Since IP_n can be computed by a threshold circuit of $2n + 1$ gates, depth 3 and energy complexity $2n + 1$, our restriction on the energy complexity cannot be relaxed to $O(n)$.

Acknowledgments

We are grateful to Professor Takao Nishizeki of Tohoku University, the supervisor of the first author, for thorough rewriting and reorganization of the paper numerous times over half a year. We also thank Professor Kazuyuki Amano of Gunma University for fruitful discussions, and the referees for helpful comments. This work is supported by MEXT Grant-in-Aid for Scientific Research on Priority Area "New Horizon in Computing".

References

[1] K. Amano, A. Maruoka, On the complexity of depth-2 circuits with threshold gates, in: Proceedings of the 30th MFCS, in: LNCS, vol. 3618, 2005, pp. 107–118.
[2] P. Földiák, Sparse coding in the primate cortex, in: The Handbook of Brain Theory and Neural Networks, vol. 1, 2003, pp. 1064–1068.
[3] J. Forster, A linear lower bound on the unbounded error probabilistic communication complexity, Journal of Computer and System Sciences 65 (2002) 612–625.
[4] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, G. Turán, Threshold circuits of bounded depth, Journal of Computer and System Sciences 46 (1993) 129–154.
[5] J. Håstad, M. Goldmann, On the power of small-depth threshold circuits, Computational Complexity 1 (1993) 113–129.
[6] G. Kissin, Upper and lower bounds on switching energy in VLSI, Journal of the Association for Computing Machinery 38 (1991) 222–254.
[7] E. Kushilevitz, N. Nisan, Communication Complexity, Cambridge University Press, 1997.
[8] P. Lennie, The cost of cortical computation, Current Biology 13 (2003) 493–497.
[9] T. W. Margrie, M. Brecht, B. Sakmann, In vivo, low-resistance, whole-cell recordings from neurons in the anaesthetized and awake mammalian brain, Pflugers Arch. 444 (4) (2002) 491–498.
[10] M. Minsky, S. Papert, Perceptrons: An Introduction to Computational Geometry, MIT Press, 1988.
[11] N. Nisan, The communication complexity of threshold gates, in: D. Miklós, V. T. Sós, T. Szőnyi (Eds.), Combinatorics, Paul Erdős is Eighty, Volume 1, János Bolyai Mathematical Society, Budapest, 1993, pp. 301–315.
[12] B.A. Olshausen, D.J. Field, How close are we to understanding V1? Neural Computation 17 (2005) 1665–1699.
[13] I. Parberry, Circuit Complexity and Neural Networks, MIT Press, 1994.
[14] A. Razborov, A. Wigderson, $n^{\Omega(\log n)}$ lower bounds on the size of depth 3 threshold circuits with AND gates at the bottom, Information Processing Letters 45 (1993) 303–307.
[15] J.H. Reif, A. Tyagi, Energy complexity of optical computations, in: Proceedings of the Second IEEE Symposium on Parallel and Distributed Processing, 1990, pp. 14–21.
[16] K.Y. Siu, V. Roychowdhury, T. Kailath, Discrete Neural Computation; A Theoretical Foundation, Information and System Sciences Series, Prentice Hall, 1995.
[17] K. Uchizawa, R. Douglas, W. Maass, On the computational power of threshold circuits with sparse activity, Neural Computation 18 (12) (2006) 2994–3008.
[18] K. Uchizawa, E. Takimoto, An exponential lower bound on the size of threshold circuits with small energy complexity, in: Proceedings of the 22nd Annual IEEE Conference on Computational Complexity, 2007, pp. 169–178.