Energy and Depth of Threshold Circuits

Kei Uchizawa,*† Takao Nishizeki*† and Eiji Takimoto‡§

Abstract

Suppose that a Boolean function f can be computed by a threshold circuit C of energy complexity e. Thus, at most e threshold gates in C output "1" for any input to C. We then prove that the function f can be computed also by a threshold circuit C' of depth $2e+1$. If the size of C is s, that is, there are s threshold gates in C, then the size of C' is $2es+1$. The proof is constructive, and hence C' can be immediately constructed from C.

1 Introduction

A threshold (logic) gate is a theoretical model of a neuron, and a threshold (logic) circuit, which is a combinatorial circuit consisting of threshold gates, is a theoretical model of a neural circuit in the brain. A threshold circuit is intensively studied for a few decades [8, 9, 10, 11, 14]. Information processing in a neural circuit results from "firing" of neurons. Recent studies in biology report that a neuron consumes a large amount of energy for firing, and consequently the firing rate of neurons is quiet low[6, 7]. Based on the fact above, the *energy complexity* e of a threshold circuit C is defined as the maximum number of threshold gates outputting "1" over all inputs to C[16]. There have been known several results on the energy complexity of threshold circuits [16, 17, 18], and it turns out that the energy complexity e of C has a simple relationship with other major complexity

*Graduate School of Information Sciences, Tohoku University, Aramaki Aoba-aza 6-6-05, Aoba-ku, Sendai, 980-8579, Japan.

†{uchizawa, nishi}@ecei.tohoku.ac.jp

‡Department of Informatics, Graduate School of Information Science and Electrical Engineering, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan.

§eiji@i.kyushu-u.ac.jp

measures such as the size s and the depth d; the *size* s of C is the number of threshold gates in C; and the *depth* d of C is the length of the longest directed path going from an input node to the output gate in C, and corresponds to the parallel computation time. In particular, there is a tradeoff $n \leq s^e$ between the size s and the energy complexity e of threshold circuits computing the PARITY function of n variables [18]. On the other hand, there is a tradeoff $n \leq 2s^{d-1}$ between the size s and the depth d of threshold circuits computing the PARITY function [5], and a tradeoff $n \leq (s/d)^{d-\epsilon}$ also holds for any $\epsilon > 0$ [15]. In all these tradeoffs, the left hand side is the number n of input variables, while either the energy complexity e or the depth d appears in the exponent of s in the right hand side. Thus the two measures, energy e and depth d, play the similar role in the tradeoffs at least for circuits computing the PARITY function, although the two measures have completely different physical meanings. Thus, the energy complexity e seems to have a close relationship with the depth d not only for the circuits computing the PARITY function but also for circuits computing any Boolean function.

In the paper, we investigate a relationship between the energy complexity and the depth of threshold circuits computing any Boolean function, and obtain the following result as the main theorem: if a Boolean function f can be computed by a threshold circuit C of energy complexity e, then f can be computed also by a threshold circuit C' of depth $d' = 2e + 1$. If C has size s, then C' has size $s' = 2es + 1$. Thus, if a Boolean function of n variables can be computed by a threshold circuit C of constant energy complexity and polynomial size in n, then f can be computed also by a threshold circuit C' of constant depth and polynomial size. Since the proof of the main theorem is constructive, a threshold circuit C' of shallow depth can be immediately obtained if a circuit C of small energy complexity is given for computing a function. The theorem immediately implies that a very simple relationship $d(f) \leq 2e(f) + 1$ holds for every Boolean function f, where $e(f)$ is the energy complexity of f defined as the minimum energy complexity of polynomial-size threshold circuits computing f and $d(f)$ is the depth complexity of f defined similarly as $e(f)$.

The rest of the paper is organized as follows. In Section 2, we first define some terms on threshold circuits, and then present the main theorem and corollaries. In Section 3, we first construct C' from C, then present a lemma on C', and finally prove the main theorem. In Section 4, we conclude with some remarks.

2 Definitions and main theorem

In the section, we first define some terms on threshold circuits, and then present the main theorem and corollaries.

A *threshold gate* in the paper is the so-called linear threshold logic gate, and can have an arbitrary number k of inputs. For every input $\boldsymbol{z} = (z_1, z_2, \cdots, z_k) \in \{0,1\}^k$ to a threshold gate g with weights $w_1, w_2, \cdots, w_k$ and a threshold t, the output $g(\boldsymbol{z})$ of the gate g for $\boldsymbol{z}$ is defined as follows:

$$g(\boldsymbol{z}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{k} w_i z_i \geq t; \\ 0 & \text{otherwise.} \end{cases}$$

We assume that the weights $w_1, w_2, \cdots, w_k$ and the threshold t are arbitrary real numbers.

A *threshold (logic) circuit* C is a combinatorial circuit consisting of threshold gates, and is represented by a directed acyclic graph, as illustrated by Fig. 1. We denote by n the number of inputs to C, and by $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$ the input variables to C. The underlying directed acyclic graph of C has n nodes of in-degree 0, each of which corresponds to one of the n input variables and is called an *input node*. The *size* s of a threshold circuit C is the number of threshold gates in C. Figure 1 depicts a threshold circuit with $n = 3$ and $s = 5$. All the wires with weight zero are not drawn in Fig. 1.

Let C be a threshold circuit of size s, and let $g_1, g_2, \cdots, g_s$ be the s gates in C. Then the input $\boldsymbol{z}_i$ to a gate g_i, $1 \leq i \leq s$, either consists of the inputs $x_1, x_2, \cdots, x_n$ to C and the outputs of the gates other than g_i or consists of some of them. However, we denote by $g_i[\boldsymbol{x}]$ the output $g_i(\boldsymbol{z}_i)$ of g_i for $\boldsymbol{z}_i$, because $\boldsymbol{x}$ decides $g_i(\boldsymbol{z}_i)$. Thus $g_i[\boldsymbol{x}] = g_i(\boldsymbol{z}_i)$.

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function of n variables. (Our main theorem can be immediately generalized to an m-output Boolean function $f : \{0,1\}^n \to \{0,1\}^m$ for any positive integer m, as stated in Section 4.) Let g_s be a gate of out-degree 0 in C, and let the output $g_s[\boldsymbol{x}]$ of g_s be the *output* $C(\boldsymbol{x})$ *of* C. Thus, $C(\boldsymbol{x}) = g_s[\boldsymbol{x}]$ for every input $\boldsymbol{x} \in \{0,1\}^n$. The gate g_s is called the *output gate* of C. 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 $\boldsymbol{x} \in \{0,1\}^n$.

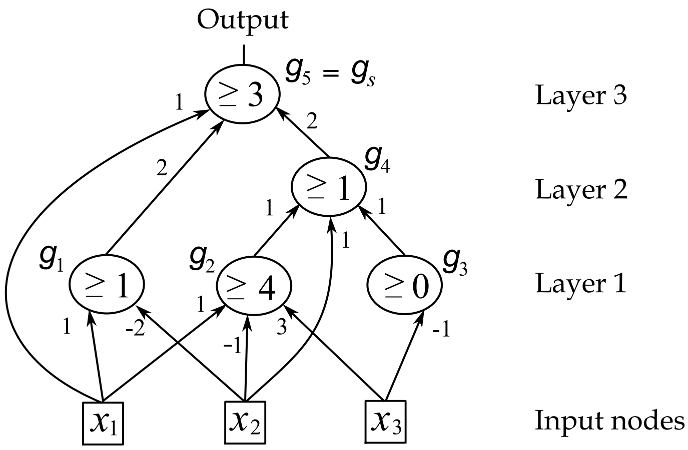

Figure 1: Threshold circuit C of $n = 3$ input nodes, size $s = 5$ and depth $d_C = 3$.

We say that a gate g_i, $1 \leq i \leq s$, is *in the l-th layer* of a circuit C if there are l gates (including g_i) on the longest path from an input node to g_i in the underlying graph of a circuit C. The *depth* d of C is the number of gates on the longest path to the output gate g_s. The circuit C in Fig. 1 has depth 3, the gates g_1, g_2 and g_3 are in the first layer, g_4 is in the second layer, and the output gate g_5 is in the third layer.

For each input $\boldsymbol{x} \in \{0,1\}^n$ to a circuit C, we denote by $e_C(\boldsymbol{x})$ the number of gates fired by $\boldsymbol{x}$, that is,

$$e_C(\boldsymbol{x}) = \sum_{i=1}^{s} g_i[\boldsymbol{x}].$$

We then define the *energy complexity* e_C of C as

$$e_C = \max_{\boldsymbol{x} \in \{0,1\}^n} e_C(\boldsymbol{x}).$$

Thus, the energy complexity e_C is the maximum number of gates outputting “1” over all inputs $\boldsymbol{x} \in \{0,1\}^n$. Obviously $0 \leq e_C \leq s$. We often denote $e_C(\boldsymbol{x})$ and e_C simply by $e(\boldsymbol{x})$ and e, respectively.

We are now ready to present our main result as the following theorem, whose proof will be given in the next section.

Theorem 1. *If a Boolean function f can be computed by a threshold circuit C of energy complexity e and size s, then f can be computed also by a threshold circuit C' of depth $d' = 2e + 1$ and size $2es + 1$.*

We immediately obtain the following corollary from Theorem 1.

Corollary 1. *If a Boolean function f of n variables can be computed by a threshold circuit C of constant energy complexity and polynomial size in n, then f can be computed also by a threshold circuit C' of constant depth and polynomial size in n.*

We now define the *energy complexity* $e(f)$ *of a Boolean function* f as the minimum energy complexity e_C among all the polynomial-size threshold circuits C computing f. Similarly we define the *depth complexity* $d(f)$ of a Boolean function f. The two complexities $e(f)$ and $d(f)$ have completely different physical meanings, but Theorem 1 immediately implies that there is a simple relationship between them, as follows.

Corollary 2. *For every Boolean function f, $d(f) \leq 2e(f) + 1$.*

A Boolean function f is *non-trivial* if $f(\boldsymbol{x}) = 1$ for some $\boldsymbol{x} \in \{0,1\}^n$ and $f(\boldsymbol{x}') = 0$ for some $\boldsymbol{x}' \in \{0,1\}^n$. Thus, if f is non-trivial then $e(f) \geq 1$, while if f is trivial then $e(f) \leq 1$ and $d(f) = 1$. The upper bound $d(f) \leq 2e(f) + 1$ on $d(f)$ in Corollary 1 can be improved to $d(f) \leq 2e(f)$ if f is non-trivial, as stated in Section 4. The bound $d(f) \leq 2e(f)$ cannot be improved to $d(f) \leq (2-\epsilon)e(f)$ for any number $\epsilon > 0$, as follows. Let n be the number of input variables, and let $n \geq 2$. Let a be an integer such that $0 < a < n$, and let f be

$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} x_i = a; \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Then f is non-trivial, and can be computed by the threshold circuit C in Fig. 2, which has the energy complexity $e = 1$. However, f cannot be computed by any threshold circuit C' of depth $d' = 1 \leq (2-\epsilon)e$.

3 Proof of Theorem 1

We prove Theorem 1 in this section.

Suppose that a Boolean function f can be computed by a threshold circuit C of energy complexity e and size s. In Section 3.1, we construct a threshold circuit C' computing f, and show that C' has depth $d' = 2e + 1$ and size $s' = 2es + 1$. In Section 3.2, we prove that C' computes f.

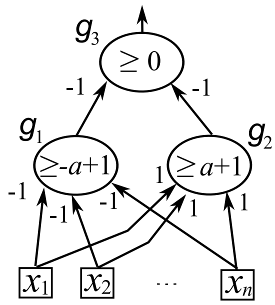

Figure 2: Threshold circuit computing f in Eq. (1).

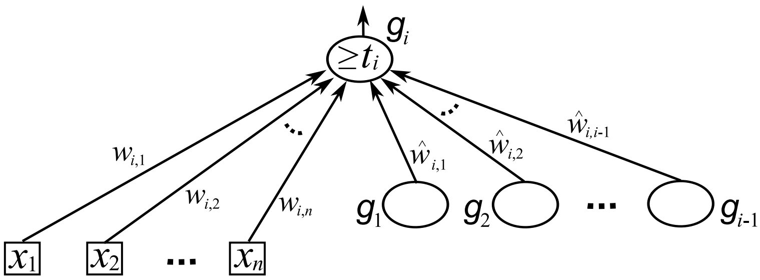

Figure 3: Weights of g_i in C.

3.1 Construction of C'

Suppose that a threshold circuit C computing f consists of s threshold gates $g_1, g_2, \cdots, g_s$, and that g_s is the output gate of C. One may assume that $g_1, g_2, \cdots, g_s$ are topologically ordered with respect to the underlying acyclic graph of C, as illustrated in Fig. 1. Thus, for each i, $1 \le i \le s$, the input $\boldsymbol{z}_i$ to a gate g_i either consists of all the n inputs $x_1, x_2, \cdots, x_n$ to C and the outputs of all the $i-1$ gates $g_1, g_2, \cdots, g_{i-1}$ preceding g_i or consists of some of them. We denote the weights of g_i for inputs $x_1, x_2, \cdots, x_n$ by $w_{i,1}, w_{i,2}, \cdots, w_{i,n}$, respectively, and denote the weights of g_i for the outputs of $g_1, g_2, \cdots, g_{i-1}$ by $\hat{w}_{i,1}, \hat{w}_{i,2}, \cdots, \hat{w}_{i,i-1}$, respectively, as illustrated in Fig. 3. Some of the weights may be zero. Let t_i be the threshold of g_i. Then the output $g_i[\boldsymbol{x}]$ of g_i is

$$g_i[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{i,j} x_j + \sum_{k=1}^{i-1} \hat{w}_{i,k} g_k[\boldsymbol{x}] - t_i\right). \tag{2}$$

Figure 4 illustrates the circuit C' which we are going to construct. The circuit C' has depth $d' = 2e + 1$. There are exactly s gates $g_1^l, g_2^l, \cdots, g_s^l$ in the l-th layer of C' for each integer l,

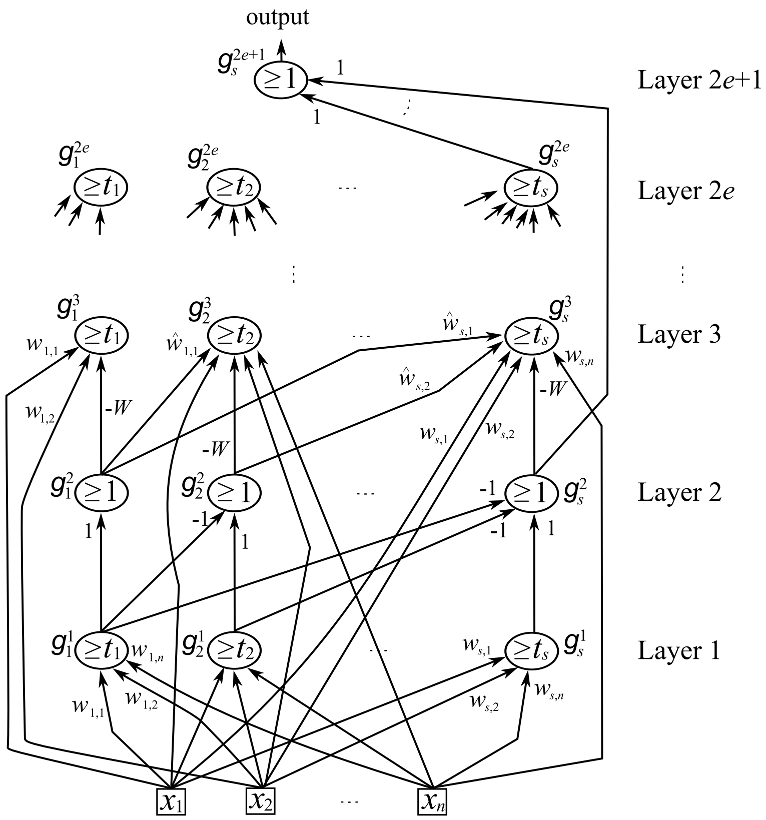

Figure 4: Sketch of C'.

$1 \leq l \leq 2e$, and there is only the output gate g_s^{2e+1} of C' in the top $(2e+1)$-st layer of C'. Thus C' has size $s' = 2es + 1$.

Intuitively speaking, each pair of layers of C' ”finds” the next gate with output 1 in C. More precisely, the circuit Csatisfies the following lemma, whose proof is omitted in this extended abstract due to the page limitation.

Lemma 1. *Let $\boldsymbol{x} \in \{0,1\}^n$ be an arbitrary input to C. Let $g_{a_1}, g_{a_2}, \cdots, g_{a_{e(\boldsymbol{x})}}$ be the $e(\boldsymbol{x})$ gates outputting "1" for $\boldsymbol{x}$, and let $1 \leq a_1 < a_2 < \cdots < a_{e(\boldsymbol{x})} \leq s$. Thus g_{a_1} fires first, g_{a_2} fires second, and subsequently $g_{a_{e(\boldsymbol{x})}}$ fires last for $\boldsymbol{x}$ in C, as illustrated in Fig. 5(a) for $\boldsymbol{x}$ such that $f(\boldsymbol{x}) = g_s(\boldsymbol{x}) = 1$. Then the following (a) and (b) hold. (See Fig. 5(b).)*

output	0	0	$\cdots$	**1**	0	$\cdots$	**1**	$\cdots$	**1**
gate no.	1	2	$\cdots$	a_1	$\cdots$		a_2	$\cdots$	$s = a_{e(\boldsymbol{x})}$

(a) Circuit C

layer \ gate no.	1	2	$\cdots$	a_1		$\cdots$	a_2			$\cdots$	$s = a_{e(\boldsymbol{x})}$	
$2e_C+1$											**1**	
$2e_C$	0	0	$\cdots$	0	0	$\cdots$	0	0	0	$\cdots$	0	Lemma 1(b)
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$		$\vdots$	$\vdots$	$\vdots$		$\vdots$	
$2e(\boldsymbol{x})+1$	0	0	$\cdots$	0	0	$\cdots$	0	0	0	$\cdots$	0	
$2e(\boldsymbol{x})$	0	0	$\cdots$	0	0	$\cdots$	0	0	0	$\cdots$	**1**	Lemma 1(a)
$2e(\boldsymbol{x})-1$	0	0	$\cdots$	0	0	$\cdots$	0	0	0	$\cdots$	**1**	
$\vdots$												
4	0	0	$\cdots$	0	0	$\cdots$	**1**	0	0	$\cdots$	0	
3	0	0	$\cdots$	0	0	$\cdots$	**1**	$*$	$*$	$\cdots$	$*$	
2	0	0	$\cdots$	**1**	0	$\cdots$	0	0	0	$\cdots$	0	
1	0	0	$\cdots$	**1**	$*$	$\cdots$	$*$	$*$	$*$	$\cdots$	$*$	

(b) Circuit C'

Figure 5: Firing patterns of (a) C and (b) C' for $\boldsymbol{x}$ such that $f(\boldsymbol{x}) = 1$ and $1 \le e(\boldsymbol{x}) < e_C$, where $*$ means 0 or 1.

(a) For every integer l, $1 \le l \le e(\boldsymbol{x})$, and every index i, $1 \le i \le s$,

$$g_i^{2l-1}[\boldsymbol{x}] = \begin{cases} 0 & \textit{if } 1 \le i \le a_l - 1; \\ 1 & \textit{if } i = a_l \end{cases} \tag{3}$$

and

$$g_i^{2l}[\boldsymbol{x}] = \begin{cases} 1 & \textit{if } i = a_l; \\ 0 & \textit{otherwise.} \end{cases} \tag{4}$$

(b) For every integer l, $e(\boldsymbol{x}) + 1 \le l \le e_C$, and every index i, $1 \le i \le s$,

$$g_i^{2l-1}[\boldsymbol{x}] = 0 \tag{5}$$

and

$$g_i^{2l}[\boldsymbol{x}] = 0. \tag{6}$$

We now show how to construct C' by separating the $2e+1$ layers into the following four sets of layers.

$\langle 1 \rangle$ *First layer*

As illustrated in Fig. 4, each gate g_i^1, $1 \le i \le s$, in the first layer of C' has the same threshold t_i as g_i in C, and receives inputs only from the input nodes $x_1, x_2, \cdots, x_n$ with the same weights as g_i. Thus, the output $g_i^1[\boldsymbol{x}]$ of g_i^1 is

$$g_i^1[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j - t_i\right) \tag{7}$$

for every input $\boldsymbol{x} \in \{0,1\}^n$. From Eq. (2) and Eq. (7), we have

$$g_i^1[\boldsymbol{x}] = g_i[\boldsymbol{x}] \quad \text{if} \;\; g_1[\boldsymbol{x}] = g_2[\boldsymbol{x}] = \cdots = g_{i-1}[\boldsymbol{x}] = 0. \tag{8}$$

If $e(\boldsymbol{x}) \ge 1$, then the gate g_{a_1} fires first for $\boldsymbol{x}$ in C and hence we have from (8)

$$g_i^1(\boldsymbol{x}) = \begin{cases} 0 & \text{if } 1 \le i \le a_1 - 1; \\ 1 & \text{if } i = a_1. \end{cases} \tag{9}$$

Thus Eq. (3) holds for $l = 1$. If $e(\boldsymbol{x}) = 0$, then $g_i[\boldsymbol{x}] = 0$ for every i, $1 \le i \le s$, and hence by (8)

$$g_i^1[\boldsymbol{x}] = 0 \tag{10}$$

for every i, $1 \le i \le s$. Thus Eq. (5) holds for $l = 1$.

$\langle 2 \rangle$ *Even-numbered layers*

We design gates g_i^{2l}, $1 \le i \le s$, in the $2l$-th layer, $1 \le l \le e$, as follows. The gate g_i^{2l} receives, as inputs, only the outputs of i gates $g_1^{2l-1}, g_2^{2l-1}, \cdots, g_i^{2l-1}$ in the $(2l-1)$-th layer, as illustrated in Fig. 6. The weights for the outputs of $g_1^{2l-1}, g_2^{2l-1}, \cdots, g_{i-1}^{2l-1}$ are -1's, and the weight for the output of g_i^{2l-1} is 1. The gate g_i^{2l} has a threshold 1. Thus, the output $g_i^{2l}[\boldsymbol{x}]$ of g_i^{2l} is

$$g_i^{2l}[\boldsymbol{x}] = \text{sign}\left(-\sum_{k=1}^{i-1} g_k^{2l-1}[\boldsymbol{x}] + g_i^{2l-1}[\boldsymbol{x}] - 1\right) \tag{11}$$

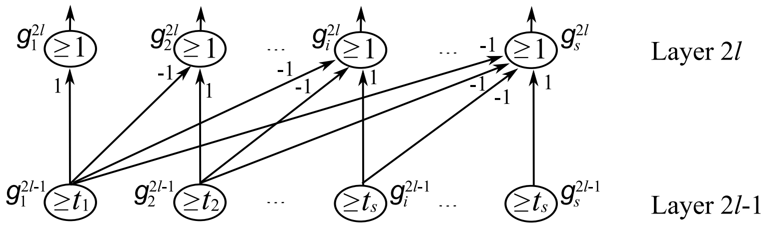

Figure 6: Construction of an even-numbered layer in C'.

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

$$g_i^{2l}[\boldsymbol{x}] = 1 \text{ if and only if} \qquad g_1^{2l-1}[\boldsymbol{x}] = g_2^{2l-1}[\boldsymbol{x}] = \cdots = g_{i-1}^{2l-1}[\boldsymbol{x}] = 0 \text{ and } g_i^{2l-1}[\boldsymbol{x}] = 1. \tag{12}$$

Hence, if g_i^{2l-1} fires first among the s gates in the $(2l-1)$-th layer, then only g_i^{2l} fires among the s gates in the $2l$-th layer.

Let $l = 1$, and consider gates g_i^2, $1 \le i \le s$, in the second layer. If $e(\boldsymbol{x}) \ge 1$, then Eqs. (9) and (12) imply

$$g_i^2[\boldsymbol{x}] = \begin{cases} 1 & \text{if } i = a_1; \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

Thus Eq. (4) holds for $l = 1$. If $e(\boldsymbol{x}) = 0$, then Eqs. (10) and (12) imply that $g_i^2[\boldsymbol{x}] = 0$ for every i, $1 \le i \le s$. Thus Eq. (6) holds for $l = 1$.

$\langle 3 \rangle$ *Odd-numbered layers*

We now design gate g_i^{2l-1}, $1 \le i \le s$, in the $(2l-1)$-th layer, $2 \le l \le e$. As illustrated in Fig. 7, the gate g_i^{2l-1} has the same threshold t_i as g_i in C, and receives inputs $x_1, x_2, \cdots, x_n$ with the same weights as g_i in C. Thus the weights of g_i^{2l-1} for $x_1, x_2, \cdots, x_n$ are $w_{i,1}, w_{i,2}, \cdots, w_{i,n}$, respectively. The gate g_i^{2l-1} receives, as inputs, also the outputs of gates $g_1^{2m}, g_2^{2m}, \cdots, g_{i-1}^{2m}$ in the $2m$-th layer for each m, $1 \le m \le l-1$, with weights $\hat{w}_{i,1}, \hat{w}_{i,2}, \cdots, \hat{w}_{i,i-1}$, respectively. In addition, g_i^{2l-1} receives the output of g_i^{2m} with weight $-W$ for each m, $1 \le m \le l-1$, where W is a sufficiently large positive integer. For example, we choose W so that

$$W > \max_{1 \le i \le s} \max_{\boldsymbol{x} \in \{0,1\}^n} \left(\sum_{j=1}^{n} w_{i,j} x_j + \sum_{k=1}^{i-1} \hat{w}_{i,k} g_k[\boldsymbol{x}] - t_i \right). \tag{14}$$

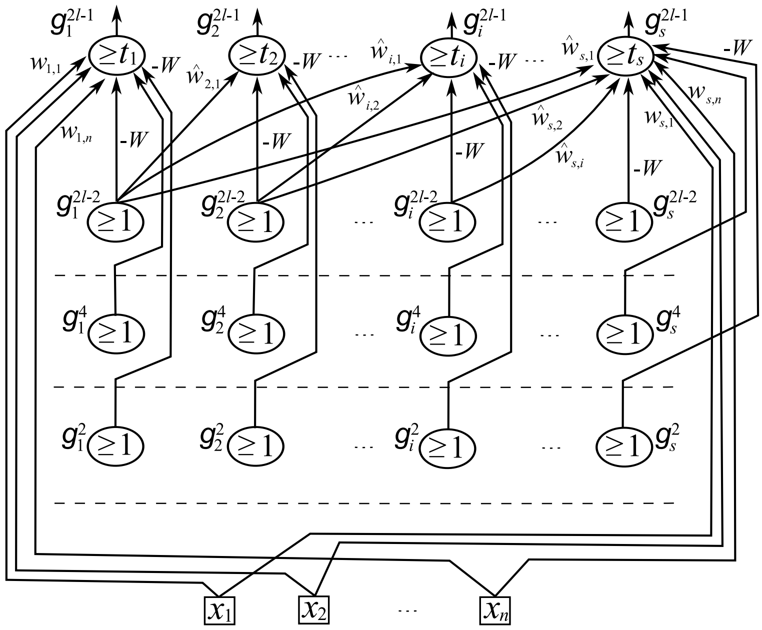

Figure 7: Construction of an odd-numbered layer in C'.

We thus have

$$g_i^{2l-1}[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{m=1}^{l-1}\sum_{k=1}^{i-1} \hat{w}_{i,k}g_k^{2m}[\boldsymbol{x}] - \sum_{m=1}^{l-1} W g_i^{2m}[\boldsymbol{x}] - t_i\right). \tag{15}$$

Hence, g_i^{2l-1} does not fire if at least one of the $l-1$ gates $g_i^2, g_i^4, \cdots, g_i^{2(l-1)}$ fires.

$\langle 4 \rangle$ *Top layer*

There is only the output gate g_s^{2e+1} in the top $(2e+1)$-th layer of C'. The threshold of the gate g_s^{2e+1} is 1, and g_s^{2e+1} receives the outputs of e gates $g_s^2, g_s^4, \cdots, g_s^{2e}$ with weights 1, as illustrated in Fig. 8. Thus

$$g_s^{2e+1}[\boldsymbol{x}] = \text{sign}\left(\sum_{l=1}^{e} g_s^{2l}[\boldsymbol{x}] - 1\right). \tag{16}$$

Hence, g_s^{2e+1} computes the OR of outputs of $g_s^2, g_s^4, \cdots, g_s^{2e}$.

We have thus completed the construction of C'.

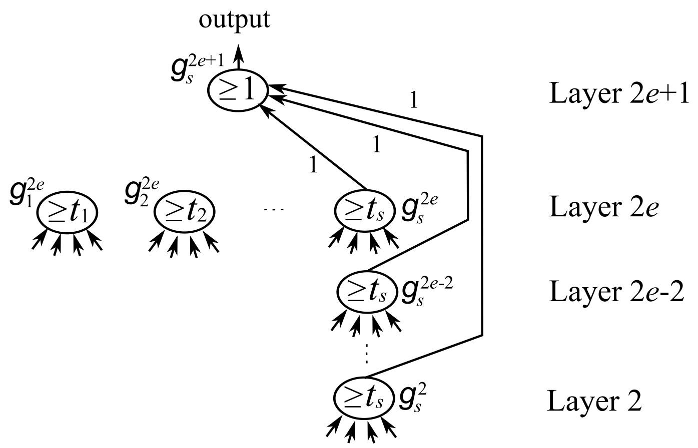

Figure 8: Connections to the output gate g_s^{2e+1} in C'.

3.2 C' computes f

In the section, we prove that the circuit C' constructed in Section 3.1 computes f, that is, $C'(\boldsymbol{x}) = f(\boldsymbol{x})$ for every input $\boldsymbol{x} \in \{0,1\}^n$. We separate the proof into two cases, $f(\boldsymbol{x}) = 1$ and $f(\boldsymbol{x}) = 0$, as follows.

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

In the case, $f(\boldsymbol{x}) = C(\boldsymbol{x}) = g_s[\boldsymbol{x}] = 1$, and hence $e(\boldsymbol{x}) \geq 1$ and $a_{e(\boldsymbol{x})} = s$ as illustrated in Fig. 5(a). Substituting $i = s$ and $l = 1, 2, \cdots, e(\boldsymbol{x})$ in Eq. (4), we obtain

$$g_s^2[\boldsymbol{x}] = g_s^4[\boldsymbol{x}] = \cdots = g_s^{2(e(\boldsymbol{x})-1)}[\boldsymbol{x}] = 0 \tag{17}$$

and

$$g_s^{2e(\boldsymbol{x})}[\boldsymbol{x}] = 1. \tag{18}$$

Substituting $i = s$ and $l = e(\boldsymbol{x}) + 1, e(\boldsymbol{x}) + 2, \cdots, e$ in Eq. (6), we have

$$g_s^{2(e(\boldsymbol{x})+1)}[\boldsymbol{x}] = g_s^{2(e(\boldsymbol{x})+2)}[\boldsymbol{x}] = \cdots = g_s^{2e}[\boldsymbol{x}] = 0. \tag{19}$$

By Eqs. (17)−(19), we have

$$\sum_{l=1}^{e} g_s^{2l}[\boldsymbol{x}] = 1. \tag{20}$$

Equations (16) and (20) imply that

$$C'(\boldsymbol{x}) = g_s^{2e+1}[\boldsymbol{x}] = \text{sign}(0) = 1 = f(\boldsymbol{x}).$$

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

In the case, $f(\boldsymbol{x}) = C(\boldsymbol{x}) = g_s[\boldsymbol{x}] = 0$ and hence $a_1, a_2, \cdots, a_{e(\boldsymbol{x})} < s$. Therefore, by Eq. (4) and Eq. (6), we have

$$\sum_{l=1}^{e} g_s^{2l}[\boldsymbol{x}] = 0. \tag{21}$$

Equations (16) and (21) imply that

$$C'(\boldsymbol{x}) = g_s^{2e+1}[\boldsymbol{x}] = \text{sign}(-1) = 0 = f(\boldsymbol{x}).$$

4 Conclusions

In the paper, we prove that a very simple relationship $d(f) \le 2e(f) + 1$ holds for every Boolean function f, where $e(f)$ is the energy complexity and $d(f)$ is the depth complexity of f. More precisely, we prove that if a function f can be computed by a threshold circuit C of energy complexity e and size s then the function f can be computed also by a threshold circuit C' of depth $d' = 2e + 1$ and $s' = 2es + 1$. Lemma 1 implies that the energy complexity e' of C' satisfies $e' \le e(s+1)+1$. Thus, the energy complexity e' of C' is not necessarily small even if the energy complexity e of C is small. Let n_{wire} be the number of wires in C, that is, n_{wire} is the number of all the non-zero weights $w_{i,j}$ and $\hat{w}_{i,k}$, $1 \le i \le s$, $1 \le j \le n$, $1 \le k \le i-1$, in C. Then the number n'_{wire} of wires in C' is $n'_{\text{wire}} \le es^2 + e^2 n_{\text{wire}}$. Let n_{in} be the maximum fan-in of gates in C, then the maximum fan-in n'_{in} of gates in C' is $n'_{\text{in}} \le \max\{s, e(n_{\text{in}} + 1)\}$. If all the weights and thresholds in C are integers, then all of them in C' are integers, too.

One can indeed decrease the depth d' of the circuit C' in Theorem 1 by 1 if f is non-trivial and hence $e \ge 1$, as follows. Since the $s-1$ gates $g_1^{2e}, g_2^{2e}, \cdots, g_{s-1}^{2e}$ in the $2e$-th layer have out-degree 0 as illustrated in Fig. 5, these $s-1$ gates can be removed from C'. The two gates g_s^{2e} and g_s^{2e+1} can be merged into a single output gate of C. One can thus construct a circuit C' of depth $d' = 2e$

and size $s' = s(2e-1)+1$. We hence have $d(f) \leq 2e(f)$ for every non-trivial Boolean function f.

One can easily generalize Theorem 1 for an m-output Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}^m$, where m is any positive integer, as follows. If such a function f can be computed by a threshold circuit C of energy complexity e and size s, then f can be computed also by a threshold circuit C' of depth $d' = 2e+1$ and size $s' = 2es+m$. The construction of C' is similar to that in Section 3.1 except for the top layer, in which there are m output gates of C', each corresponds to one of the m output gates in C and is designed similarly as g_s^{2e+1}.

One would expect that the following proposition, which is a converse proposition of Corollary 2, holds: $e(f) = O(d(f))$ for every Boolean function f. However, the proposition does not hold, as follows. The addition of two n-bit numbers can be computed by a threshold circuit C of depth $d = 2$ [14], while every circuit C computing the addition has energy complexity $e \geq n$.

A threshold circuit of constant depth and polynomial size has fairly big computational power; for example, not only the addition but also the multiplication of two n-bit numbers can be computed by such a circuit [12, 13, 19]. On the other hand, some functions cannot be computed by any threshold circuit of polynomial size and depth 2 or 3 under some restrictions on weights, thresholds, fan-ins, etc [1, 2, 3, 4]. It is interesting to know whether there is a function $f : \{0,1\}^n \rightarrow \{0,1\}$ which cannot be computed by any threshold circuit of polynomial size and constant energy complexity.

References

[1] K. Amano and A. Maruoka. On the complexity of depth-2 circuits with threshold gates. *Proceedings of the 30th MFCS*, LNCS 3618:107–118, 2005.

[2] J. Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65:612–625, 2002.

[3] A. Hajnal, W. Maass, P. Pudlák, M. Szegedy, and G. Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46:129–154, 1993.

[4] J. Håstad and M. Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1993.

[5] R. Impagliazzo, R. Paturi, and M. E. Saks. Size-depth trade-offs for threshold circuits. *SIAM Journal on Computing*, 26(3):693–707, 1997.

[6] P. Lennie. The cost of cortical computation. *Current Biology*, 13:493–497, 2003.

[7] T. W. Margrie, M. Brecht, and B. Sakmann. In vivo, low-resistance, whole-cell recordings from neurons in the anaesthetized and awake mammalian brain. *Pflugers Arch.*, 444(4):491–498, 2002.

[8] M. Minsky and S. Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, Cambridge, MA, 1988.

[9] I. Parberry. *Circuit Complexity and Neural Networks*. MIT Press, Cambridge, MA, 1994.

[10] S. Shao-Chin and T. Nishino. The complexity of threshold circuits for parity functions. *IEICE Transactions on Information and Systems*, 80(1):91–93, 1997.

[11] J. Sima and P. Orponen. General-purpose computation with neural networks: A survey of complexity theoretic result. *Neural Computation*, 15:2727–2778, 2003.

[12] K. Y. Siu, J Bruck, T Kailath, and T. Hofmeister. Depth efficient neural networks for division and related problems. *IEEE Transaction on Infromation theory*, 39:946–956, 1992.

[13] K. Y. Siu and V. Roychowdhury. On optimal depth threshold circuits for multiplication and related problems. *SIAM Journal on Discrete Mathematics*, 7(2):284–292, 1994.

[14] K. Y. Siu, V. Roychowdhury, and T. Kailath. *Discrete Neural Computation; A Theoretical Foundation*. Prentice-Hall, Inc., Upper Saddle River, NJ, 1995.

[15] K. Y. Siu, V. P. Roychowdhury, and T. Kailath. Rational approximation techniques for analysis of neural networks. *IEEE Transactions on Information Theory*, 40(2):455–466, 1994.

[16] K. Uchizawa, R. Douglas, and W. Maass. On the computational power of threshold circuits with sparse activity. *Neural Computation*, 18(12):2994–3008, 2006.

[17] K. Uchizawa and E. Takimoto. Exponential lower bounds on the size of threshold circuits with small energy complexity. *Theoretical Computer Science*, 407:474–487, 2008.

[18] K. Uchizawa, E Takimoto, and T Nishizeki. Size-energy tradeoff for threshold logic circuits computing mod functions. *IEICE Technical Report*, 108(237):63–69, 2008.

[19] C. H. Yeh and E. A. Varvarigos. Depth-efficient threshold circuits for multiplication and symmetric function computation. *Proceedings of the 2nd COCOON*, LNCS 1090:231–240, 1992.

Appendix: Proof of Lemma 1

In the section, we prove Lemma 1 by induction on l.

Let $A = \{a_1, a_2, \cdots, a_{e(\boldsymbol{x})}\}$, and let $A_i = \{a \in A \mid a \leq i-1\}$ for every integer i, $1 \leq i \leq s$. Then $A_1 = \emptyset$, and if $e(\boldsymbol{x}) = 0$ then $A = A_1 = A_2 = \cdots = A_s = \emptyset$. Clearly, $g_k[\boldsymbol{x}] = 1$, $1 \leq k \leq s$, if and only if $k \in A$. Therefore, Eq. (2) can be rewritten as follows:

$$g_i[\boldsymbol{x}] \quad = \quad \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{a \in A_i} \hat{w}_{i,a} - t_i\right). \tag{22}$$

We prove Lemma 1(a) in Section A.1 and Lemma 1(b) in Section A.2.

A.1 Proof of Lemma 1(a)

We have already proved in $\langle 1 \rangle$ and $\langle 2 \rangle$ of Section 3.1 that Eq. (3) and Eq. (4) hold for $l = 1$.

Let l be an integer such that $2 \leq l \leq e(\boldsymbol{x})$, and assume that Eq. (3) and Eq. (4) hold for every integer m, $1 \leq m \leq l-1$. We then prove that Eq. (3) and Eq. (4) hold for the integer l. Equation (4) immediately follow from Eq. (3) and (12). It thus suffices to prove only Eq. (3). Therefore, one may assume that $1 \leq i \leq a_l$.

The second term in the parenthesis of the right hand side of Eq. (15) can be expanded as follows:

$$\sum_{m=1}^{l-1}\sum_{k=1}^{i-1} \hat{w}_{i,k} g_k^{2m}[\boldsymbol{x}] \quad = \quad \sum_{k=1}^{i-1} \hat{w}_{i,k} g_k^{2}[\boldsymbol{x}] + \sum_{k=1}^{i-1} \hat{w}_{i,k} g_k^{4}[\boldsymbol{x}] + \cdots + \sum_{k=1}^{i-1} \hat{w}_{i,k} g_k^{2l-2}[\boldsymbol{x}]. \tag{23}$$

Since we assume that Eq. (4) holds for every integer m, $1 \leq m \leq l-1$,

$$g_k^{2m}[\boldsymbol{x}] = \begin{cases} 1 & \text{if } k = a_m \\ 0 & \text{otherwise} \end{cases}$$

for every k, $1 \leq k \leq s$. Therefore, the m-th term in the right hand side of Eq. (23) is

$$\sum_{k=1}^{i-1} \hat{w}_{i,k} g_k^{2m}[\boldsymbol{x}] = \begin{cases} \hat{w}_{i,a_m} & \text{if } a_m \leq i-1; \\ 0 & \text{otherwise.} \end{cases} \tag{24}$$

Since $1 \leq i \leq a_l$, the equation $a_m \leq i-1$ implies $m \leq l-1$. Therefore, Eqs. (23) and (24) imply that

$$\sum_{m=1}^{l-1}\sum_{k=1}^{i-1} \hat{w}_{i,k} g_k^{2m}[\boldsymbol{x}] = \sum_{a \in A_i} \hat{w}_{i,a}. \tag{25}$$

By Eqs. (15) and (25), we have

$$g_i^{2l-1}[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{a\in A_i} \hat{w}_{i,a} - \sum_{m=1}^{l-1} W g_i^{2m}[\boldsymbol{x}] - t_i\right). \tag{26}$$

We separate the proof of Eq. (3) into two cases, $i \notin A$ and $i \in A$, as follows.

Case 1: $i \notin A$.

In this case, we have $1 \le i \le a_l - 1$ because $1 \le i \le a_l$ and $i \notin A$. Therefore, we shall prove that $g_i^{2l-1}[\boldsymbol{x}] = 0$. Since $i \notin A$, the gate g_i in C does not fire for $\boldsymbol{x}$ and hence by Eq. (22)

$$g_i[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{a\in A_i} \hat{w}_{i,a} - t_i\right) = 0. \tag{27}$$

Since $i \notin A$ and we assume that Eq. (4) holds for every integer m, $1 \le m \le l-1$, we have $g_i^{2m}[\boldsymbol{x}] = 0$ for every integer m, $1 \le m \le l-1$, and hence

$$\sum_{m=1}^{l-1} W g_i^{2m}[\boldsymbol{x}] = 0. \tag{28}$$

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

$$g_i^{2l-1}[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{a\in A_i} \hat{w}_{i,a} - t_i\right). \tag{29}$$

Equations (27) and (29) imply that $g_i^{2l-1}[\boldsymbol{x}] = g_i[\boldsymbol{x}] = 0$.

Case 2: $i \in A$.

In this case, we shall prove that

$$g_{a_l}^{2l-1}[\boldsymbol{x}] = 1 \tag{30}$$

and

$$g_i^{2l-1}[\boldsymbol{x}] = 0 \tag{31}$$

if $1 \le i \le a_l - 1$.

We first prove Eq. (30). Substituting $i = a_l$ in Eq. (22), we have

$$g_{a_l}[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{a_l,j}x_j + \sum_{a\in A_{a_l}} \hat{w}_{a_l,a} - t_{a_l}\right) = 1. \tag{32}$$

Since Eq. (4) holds for every integer m, $1 \le m \le l-1$, we have

$$\sum_{m=1}^{l-1} g_{a_l}^{2m}[\boldsymbol{x}] = 0. \tag{33}$$

Substituting $i = a_l$ in Eq. (26) and then using Eq. (33), we have

$$g_{a_l}^{2l-1}[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{a_l,j}x_j + \sum_{a\in A_{a_l}} \hat{w}_{a_l,a} - t_{a_l}\right). \tag{34}$$

Equations (32) and (34) imply that $g_{a_l}^{2l-1}[\boldsymbol{x}] = g_{a_l}[\boldsymbol{x}] = 1$, and hence Eq. (30) holds.

We then prove Eq. (31). Since $i \in A$ and $1 \le i \le a_l - 1$, we have $i = a_r$ for some index r, $1 \le r \le l-1$. Substituting $i = a_r$ in Eq. (4) for each integer m, $1 \le m \le l-1$, we obtain

$$g_{a_r}^{2}[\boldsymbol{x}] = g_{a_r}^{4}[\boldsymbol{x}] = \cdots = g_{a_r}^{2(r-1)}[\boldsymbol{x}] = 0, \tag{35}$$

$$g_{a_r}^{2r}[\boldsymbol{x}] = 1 \tag{36}$$

and

$$g_{a_r}^{2(r+1)}[\boldsymbol{x}] = g_{a_r}^{2(r+2)}[\boldsymbol{x}] = \cdots = g_{a_r}^{2l-2}[\boldsymbol{x}] = 0. \tag{37}$$

Equations (35)$-$(37) imply that

$$-\sum_{m=1}^{l-1} W g_{a_r}^{2m}[\boldsymbol{x}] = -W g_{a_r}^{2r}[\boldsymbol{x}] = -W. \tag{38}$$

Substituting $i = a_r$ and Eq. (38) in the parenthesis in the right hand side of Eq. (26) and then using Eq. (14), we have

$$\begin{aligned}\sum_{j=1}^{n} w_{a_r,j}x_j + \sum_{a\in A_{a_r}} \hat{w}_{a_r,a} - \sum_{m=1}^{l-1} W g_{a_r}^{2m}[\boldsymbol{x}] - t_{a_r}\\ = \left(\sum_{j=1}^{n} w_{a_r,j}x_j + \sum_{a\in A_{a_r}} \hat{w}_{a_r,a} - t_{a_r}\right) - W < 0.\end{aligned}$$

Thus Eq. (26) implies that $g_{a_r}^{2l-1}[\boldsymbol{x}] = 0$, and hence Eq. (31) holds.

A.2 Proof of Lemma 1(b)

Equation (6) immediately follows from Eq. (5) and (12). We thus prove only Eq. (5) by induction on l.

(1) *Induction basis*

We prove that Eq. (5) holds for the integer $l = e(\boldsymbol{x}) + 1 \le e_C$.

Consider first the case where $e(\boldsymbol{x}) = 0$. In this case, we have $l = e(\boldsymbol{x}) + 1 = 1$. Since $e(\boldsymbol{x}) = 0$, Eq. (2) implies that

$$g_i[\boldsymbol{x}] = \text{sign}\left(\sum_{k=1}^{n} w_{i,k}x_k - t_i\right) = 0 \tag{39}$$

for every index i, $1 \le i \le s$. By Eq. (7) and Eq. (39), we have $g_i^1[\boldsymbol{x}] = g_i[\boldsymbol{x}] = 0$ for every index i, $1 \le i \le s$, and hence Eq. (5) holds for $l = e(\boldsymbol{x}) + 1$.

Consider next the case where $e(\boldsymbol{x}) \ge 1$. An argument similar to Eqs. (23)$-$(25) implies that

$$\sum_{m=1}^{e(\boldsymbol{x})} \sum_{k=1}^{i-1} \hat{w}_{i,k} g_k^{2m}[\boldsymbol{x}] = \sum_{a \in A_i} \hat{w}_{i,a} \tag{40}$$

for every index i, $1 \le i \le s$. Substituting $l = e(\boldsymbol{x}) + 1$ in Eq. (15) and then using Eq. (40), we have

$$g_i^{2e(\boldsymbol{x})+1}[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{a \in A_i} \hat{w}_{i,a} - \sum_{m=1}^{e(\boldsymbol{x})} W g_i^{2m}[\boldsymbol{x}] - t_i\right). \tag{41}$$

We separate the proof of Eq. (5) for $l = e(\boldsymbol{x}) + 1$ to the following two cases.

Case 1: $i \notin A$.

In this case, $g_i[\boldsymbol{x}] = 0$, and hence by Eq. (22) we have

$$g_i[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{a \in A_i} \hat{w}_{i,a} - t_i\right) = 0. \tag{42}$$

Equations (4) and $i \in A$ imply that $g_i^{2m}[\boldsymbol{x}] = 0$ for every m, $1 \le m \le e(\boldsymbol{x})$, and hence we have

$$\sum_{m=1}^{e(\boldsymbol{x})} W g_i^{2m}[\boldsymbol{x}] = 0. \tag{43}$$

By Eq. (41)−(43), we have

$$g_i^{2e(\boldsymbol{x})+1}[\boldsymbol{x}] = \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{a\in A_i} \hat{w}_{i,a} - t_i\right) = 0, \tag{44}$$

and hence Eq. (5) holds for $l = e(\boldsymbol{x}) + 1$.

Case 2: $i \in A$.

Let $i = a_r \in A$, where $1 \leq r \leq e(\boldsymbol{x}) = l - 1$. Since $l = e(\boldsymbol{x}) + 1$, Eq. (4) implies that

$$g_{a_r}^{2}[\boldsymbol{x}] = g_{a_r}^{4}[\boldsymbol{x}] = \cdots = g_{a_r}^{2(r-1)}[\boldsymbol{x}] = 0, \tag{45}$$

$$g_{a_r}^{2r}[\boldsymbol{x}] = 1, \tag{46}$$

and

$$g_{a_r}^{2(r+1)}[\boldsymbol{x}] = g_{a_r}^{2(r+2)}[\boldsymbol{x}] = \cdots = g_{a_r}^{2e(\boldsymbol{x})}[\boldsymbol{x}] = 0. \tag{47}$$

By Eqs. (45)−(47), we have

$$-\sum_{m=1}^{e(\boldsymbol{x})} W g_{a_r}^{2m}[\boldsymbol{x}] = -W g_{a_r}^{2r}[\boldsymbol{x}] = -W. \tag{48}$$

Substituting $i = a_r$ and Eq. (48) in the parenthesis in the right hand side of Eq. (41) and then using Eq. (14), we have

$$\begin{aligned} &\sum_{j=1}^{n} w_{a_r,j}x_j + \sum_{a\in A_i} \hat{w}_{a_r,a} - \sum_{m=1}^{e(\boldsymbol{x})} W g_{a_r}^{2m}[\boldsymbol{x}] - t_{a_r} \\ &= \left(\sum_{j=1}^{n} w_{a_r,j}x_j + \sum_{a\in A_{a_r}} \hat{w}_{a_r,a} - t_{a_r}\right) - W < 0. \end{aligned} \tag{49}$$

Equations (41) and (49) imply that $g_{a_r}^{2e(\boldsymbol{x})+1}[\boldsymbol{x}] = 0$, and hence Eq. (5) holds for $l = e(\boldsymbol{x}) + 1$.

(2) *Induction hypothesis*

Let l be an integer such that $e(\boldsymbol{x}) + 2 \leq l \leq e$, and assume that Eq. (5) and Eq. (6) hold for every integer m, $e(\boldsymbol{x}) + 1 \leq m \leq l - 1$.

(3) *Induction step for Eq. (5)*

We prove that Eq. (5) holds for the integer l. Since Eq. (6) is assumed to hold for every integer m, $e(\boldsymbol{x})+1 \leq m \leq l-1$, we have $g_i^{2m}[\boldsymbol{x}] = 0$ for every integer m, $e(\boldsymbol{x})+1 \leq m \leq l-1$, and every index i, $1 \leq i \leq s$. Thus, for every index i, $1 \leq i \leq s$, we have

$$\sum_{m=1}^{l-1}\sum_{k=1}^{i-1}\hat{w}_{i,k}g_k^{2m}[\boldsymbol{x}] = \sum_{m=1}^{e(\boldsymbol{x})}\sum_{k=1}^{i-1}\hat{w}_{i,k}g_k^{2m}[\boldsymbol{x}] \tag{50}$$

and

$$\sum_{m=1}^{l-1} W g_i^{2m}[\boldsymbol{x}] = \sum_{m=1}^{e(\boldsymbol{x})} W g_i^{2m}[\boldsymbol{x}]. \tag{51}$$

Using Eqs. (15), (50) and (51) and then applying the induction basis, we have

$$\begin{aligned} g_i^{2l-1}[\boldsymbol{x}] &= \text{sign}\left(\sum_{j=1}^{n} w_{i,j}x_j + \sum_{l=1}^{e(\boldsymbol{x})}\sum_{k=1}^{i-1}\hat{w}_{i,k}g_k^{2l}[\boldsymbol{x}] - \sum_{l=1}^{e(\boldsymbol{x})} W g_i^{2l}[\boldsymbol{x}] - t_i\right) \\ &= g_i^{2e(\boldsymbol{x})+1}[\boldsymbol{x}] \\ &= 0 \end{aligned}$$

and hence Eq. (5) hold for l.