Title	Energy and Depth of Threshold Circuits Computing Parity Function (Recent Trends in Algorithms and Computation)
Author(s)	Uchizawa, Kei
Citation	数理解析研究所講究録 = RIMS Kokyuroku (2019), 2132: 20-22
Issue Date	2019-10
URL	http://hdl.handle.net/2433/254785
Right	
Type	Departmental Bulletin Paper
Textversion	publisher

Energy and Depth of Threshold Circuits Computing Parity Function

Kei Uchizawa

Graduate School of Science and Engineering, Yamagata University, Jonan 4-3-16, Yonezawa-shi Yamagata, 992-8510 Japan.

1 Introduction

A logic circuit consists of a number of basic computational elements (i.e., gates), and computes a Boolean function. In the standard model of a logic circuit, a gate computes AND, OR or NOT function. A threshold circuit is a logic circuit consisting of gates computing linear threshold functions: a gate g with n input variables has weights $w_1, w_2, \dots, w_n$ with a threshold t, and computes $g(\mathbf{x}) = \mathrm{sign}(\sum_{i=1}^{n} w_i x_i - t)$ for $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \{0,1\}^n$.

The computational power of a threshold circuit is well-studied in the literature, and is shown to be very expressive even if the size (i.e., the number of gate) is bounded to be a polynomial, and the depth to be constant. Even a single threshold gate is able to compute the majority function which the standard logic circuit of constant depth needs an exponential size to compute [4]. Also, a threshold circuit of a polynomial size and constant depth computes a number of arithmetic functions such as addiction, multiplication and division [9]. These results leave us a question: What Boolean function is not computable by threshold circuits of polynomial size and constant depth? This is one of the cutting-edge questions in circuit complexity, and we can not even rule out the possibility that every problem in $\mathrm{EXP}^{\mathrm{NP}}$ is solvable by a threshold circuit of polynomial size and depth 2 (See, for example, the paper [1]).

Research on biology also motivates a study on threshold circuit, since a threshold gate is traditionally considered to be a theoretical model of a biological neuron. As a neuron receive inputs from other neurons, and emits an electrical signal (i.e., a spike), a threshold gate receives inputs, and emits the value one. The aim of this line of the research is to find out computational principles of the brain, and understand how the brain carry out interesting and difficult information processing. We focus on a biological fact about a neuron: a biological neuron needs more energy to emit a spike than not to transmit one, and consequently their computation results from a relatively small number of simultaneously active neurons out of a large population in the nervous system [6, 8]. This fact is sometimes called a sparse coding, and affects a recent progress of the so-called deep learning method [3, 5, 7].

In this paper, we employ the energy complexity as a measure for sparsity, and investigate the computational power of a threshold circuit with sparse activity. For a threshold circuit C, the energy of C is defined to be the maximum number of gates outputting "1" in C, where the maximum is taken over all the input assignments to C [11]. In the previous research, it turns out that the energy relates to other major complexity measures of threshold circuits. In particular, it is known that there exists a Boolean function f that any threshold circuit of constant depth and energy $n^{o(1)}$ requires an exponential number of gates to compute [12]. This result is shown by the communication complexity argument: they show that f has high communication complexity, while every Boolean function computable by a threshold circuit of small energy has low communication complexity. Thus, we can apply the lower bound to any Boolean function of high communication complexity.

However, other results on the energy complexity suggests that even low communication complexity Boolean function such as the n-variable parity function is hard for threshold circuits of small depth and energy. In the case of energy $e = 1$, it is known that any threshold circuit needs size 2^{n-1} to compute [10]. For $e \geq 2$, there is a lower bound $\Omega(n^{1/e})$ on the size of threshold circuit of energy e [13], Although it is shown that the parity function is computable by a threshold circuit of size $O(en^{1/(e-1)})$ and energy

e [10], looking into the construction given in [10] shows that their circuit has large depth to have small energy.

There are other good lower bounds for threshold circuits computing the parity function. It is known that any threshold circuit of depth d has size $(n/2)^{1/2(d-1)}$ to compute the parity function [2], and that any threshold circuit of depth 2 needs size $\Omega(\sqrt{n})$ even to approximate the function [14]. However, these results are of energy-independent, and hence do not rule out the possibility that the parity function is computable by a threshold circuit of size $\sqrt{n}$, depth 2 and energy 2.

Here we show a statement that for any threshold circuit of size s, depth d and energy e, it holds that $n/(e2^{e+d}\log^e n) \le \log s$. This inequality implies that a threshold circuit of constant depth and energy e requires size $2^{\Omega(n/e\log^e n)}$ to compute the n-variable parity function. Thus, although the parity function has low communication complexity, an exponential number of gates is required for threshold circuits of small depth and energy.

2 Definitions

A *threshold gate* g is a logic gate computing a linear threshold function of an arbitrary integer z of inputs, which is identified by weights $w_1, w_2, \ldots, w_z$ for the z input variables and a threshold t. We define the output $g(\mathbf{x})$ of g as follows:

$$g(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{z} w_i x_i \ge t; \\ 0 & \text{otherwise.} \end{cases}$$

A *threshold circuit* C is a feedforward circuit consisting of threshold gates, and is expressed by a directed acyclic graph. Let n be the number of inputs to C, then C has n input nodes of in-degree 0, each of which corresponds to one of the n input variables $x_1, x_2, \ldots x_n$, while the other nodes correspond to threshold gates. The inputs to a gate g in C consists of the inputs $x_1, x_2, \ldots x_n$ and the outputs of some gates directed to g.

We define the *size* s of a circuit C as the number of gates, and denoted by $g_1, g_2, \ldots, g_s$ the gates in C where the gates are numbered in the topological order on the underlying directed acyclic graph of C. We regard the output $g_s(\mathbf{x})$ of g_s as the *output* $C(\mathbf{x})$ *of* C, that is, $C(\mathbf{x}) = g_s(\mathbf{x})$ for every input $\mathbf{x} \in \{0,1\}^n$. A threshold circuit C *computes* a Boolean function $f : \{0,1\}^n \to \{0,1\}$ if $C(\mathbf{x}) = f(\mathbf{x})$ for every $\mathbf{x} \in \{0,1\}^n$. We say that a gate g_i, $1 \le i \le s$, 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. We define the *energy* e of C as $e = \max_{\mathbf{x}\in\{0,1\}^n} \sum_{i=1}^{s} g_i(\mathbf{x})$.

For every $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \{0,1\}^n$, the n-variable parity function PAR_n is defined to be

$$\mathrm{PAR}_n(\mathbf{x}) = \begin{cases} 1 & \text{if } x_1 + x_2 + \cdots + x_n \text{ is odd;} \\ 0 & \text{if otherwise.} \end{cases}$$

3 Lower Bound

Theorem 1. *If a threshold circuit of size s, depth d and energy e computes* PAR_n *(or its complement), then it holds that $n/(e2^{e+d}\log^e n) \le \log s$.*

The proof of the theorem proceeds by induction on the energy and depth: For a circuit C of size s, depth d, energy e, we give a procedure that reduces either depth or energy by fixing part of the input variables. In particular, we use the following lemma to reduce the energy. For a threshold gate g of n variables, we define $S_1(g) = \{\mathbf{x} \in \{0,1\}^n \mid g(\mathbf{x}) = 1\}$.

Lemma 1. *For any threshold gate g with n input variables, there exists $I \subseteq \{1, 2, \ldots, n\}$ and $\mathbf{b} \in \{0,1\}^{[n]\setminus I}$ satisfying $\log|S_1(g)|/\log n \le |I|$ such that, by fixing the variables for $[n]\setminus I$ to $\mathbf{b}$, g outputs one for every $\mathbf{a} \in \{0,1\}^I$.*

It is well-known that we can force any threshold gate to be constant either zero or one by fixing at most half of input variables [9], but the above lemma states that if $S_1(g)$ is large, we can force g to be the constant one, while we leave a certain number of input variables. We then show that one of the following events must occur: (i) no gate in the first layer outputs one for inputs in a large Boolean cube; (ii) some gate in the first layer outputs one for a large number of inputs. Thus, by fixing appropriate set of input variables, we can obtain a circuit C' of depth $d-1$ for (i), or a circuit C' of energy $e-1$ for (ii), where C' computes $\mathrm{PAR}_{n'}$ or its complement, and n' is sufficiently large for our statement.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP16K00006 and the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University.

References

[1] L. Chen and R. Tell. Bootstrapping results for threshold circuits "just beyond" known lower bounds. ECCC, TR18-199, 1–43. 2018.

[2] R. Impagliazzo, R. Paturi , M. E. Saks. Size-depth tradeoffs for threshold circuits. *SIAM Journal on Computin*, 26(3):693–707. 1997.

[3] Y. He, K. Kavukcuoglu, Y. Wang, A. Szlam and Y. Qi. Unsupervised feature learning by deep sparse coding. *Proceedings of SIAM International Conference on Data Mining 2014*, 902–910, 2014.

[4] Stasys Jukna. Boolean Function Complexity: Advances and Frontiers, Algorithms and Combinatorics 27, Springer-Verlag Berlin Heidelberg, 2012.

[5] H. Lee, A. Battle, R. Raina, A. Y. Ng Efficient sparse coding algorithms. *Advances in Neural Information Processing Systems*, 19:801–808, 2006.

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

[7] A. Y. Ng Sparse autoencoder, CS294A Lecture notes, 2011.

[8] B. Olshausen and D. J. Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4):481–287, 2004.

[9] K. Y. Siu, V. Roychowdhury, T. Kailath, Discrete Neural Computation: A Theoretical Foundation, Information and System Sciences Series, Prentice Hall, 1995.

[10] A. Suzuki, K. Uchizawa, and X. Zhou. Energy-efficient threshold circuits computing mod functions. *International Journal of Foundations of Computer Science*, 24:15–29, 2013.

[11] K. Uchizawa, R. Douglas, and W. Maass. On the computational power of threshold circuits with sparse activity. *Neural Computation*, 18(12):2994–3008, 2008.

[12] K. Uchizawa and E. Takimoto. Exponential lower bounds on the size of constant-depth threshold circuits with small energy complexity. *Theoretical Computer Science*, 407(1-3):474–487, 2008.

[13] K. Uchizawa, E. Takimoto, and T. Nishizeki. Size-energy tradeoffs of unate circuits computing symmetric Boolean functions. *Theoretical Computer Science*, 412:773–782, 2011.

[14] D. M. Kane and R. Williams. Super-linear gate and super-quadratic wire lower bounds for depth-two and depth-three threshold circuits. Proc. of *the 48th annual ACM symposium on Theory of Computing*, 633–643, 2016.