

XOR learning by spiking neural network with infrared communications

Kazuki Matsumoto*, Hiroyuki Torikai† and Hiroo Sekiya*

* Graduate School of Advanced Integration Science,
Chiba University, Japan

E-mail: matsumoto@chiba-u.jp Tel/FAX:+81-43-290-3258/+81-43-290-3269

† Department of Electrical and Electronic Engineering, Hosei University, Japan

Abstract—A Spiking Neural Network (SNN), which expresses information by spike trains, has an ability to process information with low energy like a human brain. Hardware implementation of a SNN is an important research problem. If the neurons are linked by wireless communications, SNNs can obtain the spatial degree of freedom, which may extend application area dramatically. Additionally, such SNNs can process information with low energy, owing to wireless communication by the spike trains. Therefore, it is regarded as low power-consumption wireless sensor networks (WSNs) with adding the functions of SNN neurons to wireless sensor nodes. This “Wireless Neural Sensor Networks” can distribute information processing like a brain on the WSN nodes. This paper presents a SNN with infrared(IR) communications as the first step of the above concept. Neurons are implemented by field programmable gate array, which are linked by IR communications. The implemented SNN succeeded in acquiring the XOR function through reinforcement learning.

I. INTRODUCTION

Biological information processing, which can high processing performance with low power consumption like a human brain, attracts many researchers attentions in recent years. Neural networks, which are models of brain information processing, are adapted in various field applications such as image processing and speech recognition. Spiking Neural Network (SNN) consists of spiking neurons, for example Izhikevich model [1], Hodgkin-Huxley model [2], and Integrate and Fire (IF) model [3]. The spiking neurons, which communicate information one another by spikes, imitates biological experimental-observed dynamics of neuron. Spike trains generated by each neuron have information in spike timing and rate of spikes [4]–[6]. For example, the SNNs can learn patterns of spike trains by the modifications of synaptic weights as a function of relative timing of pre and postsynaptic spikes, which is called Spike Timing Dependent Plasticity (STDP) [7]–[15].

A hardware implementation of a SNN is an important research topic. Theoretically, the SNN hardware can achieve high speed and low energy-consumption information processing [16], [17]. Actually, many SNN hardwares were proposed with wired connections among neurons [18], [19]. If the neurons are linked by wireless communications, neurons can be located arbitrary with spatial, which may extend application area of the SNNs dramatically. The SNNs can transmit spike information by wireless communications with low energy. Therefore, such SNNs are regarded as low power-consumption

wireless sensor networks (WSNs) with adding the functions of SNN neurons to wireless sensor nodes. Additionally, this “Wireless Neural Sensor Network” (WNSN) provide an ability of distributed information processing on the WSNs.

The Field Programmable Gate Array (FPGA) is the circuit devise, which can reconfigure a circuit topology by programmable software. Modern FPGAs include large number of logic gates and physical memories. Because it is possible to achieve parallel computations, FPGAs are effective tool for hardware implementation of the neural networks [20]–[22].

This paper presents a SNN implementation on FPGAs, which are linked by infrared (IR) communications as the first step for realizing our WNSN concept. The XOR function was acquired on the implemented SNN with reinforcement learning, which showed the ability and potential to realize the WNSN.

II. PROPOSED SNN

A. Neuron model

In this paper, a SNN is constructed by IF neurons [3] in discrete time. Figure 1 shows a schematic diagram of presynaptic and postsynaptic neurons and an example of the membrane potential dynamics. Synaptic weights between neurons are defined as w_{ij} , where j and i are labels of presynaptic and postsynaptic neurons, respectively, as shown in Fig. 1(a). The increase in postsynaptic membrane potential is caused by receiving presynaptic neuron spikes. The membrane potential values depend on neurons. The membrane potential of the neuron i is expressed by

$$v_i(t) = v_r + (v_i(t - \delta) - v_r) \exp(-\delta/\tau) + \sum_j w_{ij}(t) f_j(t - \delta),$$

$$\begin{aligned} &\text{if } v_i(t) \geq v_\theta, \text{ then } v_i(t) = v_r \text{ and } f_i(t) = 1, \\ &\text{if } v_i(t) < v_\theta, \text{ then } f_i(t) = 0, \end{aligned} \quad (1)$$

where v_r is the reset membrane potential value, δ is the time step, τ is the time constant for the exponential decay, and $f_i(t)$ expresses the state of fire of neuron i . The membrane potential decreases according to the exponential decay. If the membrane potential is larger than the threshold, the postsynaptic neuron fires and the membrane potential resets to the fixed value as shown in Fig 1(b).

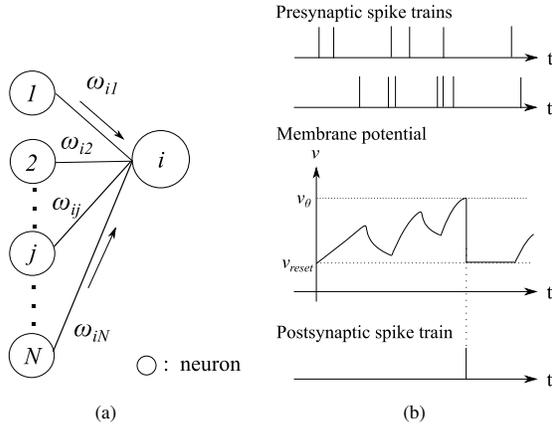


Fig. 1: Schematic of feed forward neural network. (a) Presynaptic and postsynaptic neurons. (b) Dynamics example of membrane potential

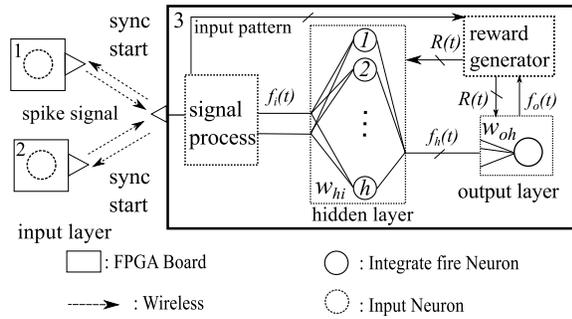


Fig. 2: Proposed system diagram

B. Proposed SNN implementation

Figure 2 shows a diagram of the proposed system. In the proposed system, a SNN is implemented on FPGAs, which are linked by IR communications. Table I gives FPGA and IR communication module types, which are used in this paper. The implemented SNN has two input neurons, 14 hidden neurons, one output neuron, which are linked with forward connections. Two neurons of the input layer are implemented on individual FPGAs. Hidden neurons, output neuron, and reward generator are installed in one FPGA. The reward generator was carried out information processing for reinforcement learning. The links between input to hidden layer neurons are achieved by the IR communications.

C. Infrared communication

In the proposed system, IR communication is adapted for between input-layer neurons and hidden-layer neurons. Two input units generate a spike train according to input signals.

In this paper, the communication multiplexing from two input units to one information processing unit was applied to the Time Division Multiple Access (TDMA). Two time

TABLE I: FPGA and Communication Modules

FPGA board	Xilinx basys-3 Artix-7
Receiver module	7.4*6.2*5.25mm Infrared Receiver Module
Transmitter module	5mm Round Infrared

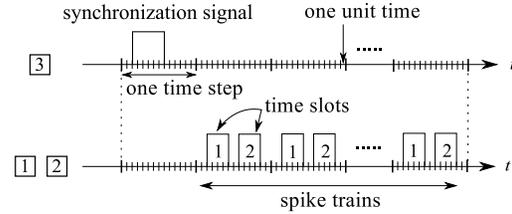


Fig. 3: Structure of time slots

slots are prepared for two input neuron spike signals. Figure 3 shows structure of time slots. For making time synchronization among FPGAs, the synchronization signal is transmitted from the information processing unit to input units. Input units transmit spike signals from the next time step of the synchronization signal. The spike signals from input units can be distributed to all the hidden-layer neurons, which are transmitted in the duration of the time slots. In this system, one time step is divided into 100 unit time. One unit time is 0.026 ($= 1/(37.9 \times 1000)$) second because cut-off frequency of the band-pass filtering at the IR receiver module is 37.9 kHz.

III. LEARNING MECHANISM

A. Reinforcement learning

A reinforcement learning is one of learning methods of the SNNs. In the reinforcement learning, the reward signals, which are generated by reward generator according to output neuron state and input neuron spike trains, are distributes to all neurons. The reinforcement learning algorithm in [7], which is called the modulated STDP with eligibility trace (MSTDPET) [7], was installed in the implemented SNN. The synapse weights are renewed according to reward signals. The calculation flow of renewing synapse weights is as follows.

Firstly, the influences of postsynaptic and presynaptic are obtained from

$$P_{ij}^+(t) = P_{ij}^+(t - \delta) \exp\left(-\frac{\delta}{\tau_+}\right) + A_+ f_j(t), \quad (2)$$

and

$$P_{ij}^-(t) = P_{ij}^-(t - \delta) \exp\left(-\frac{\delta}{\tau_-}\right) + A_- f_i(t), \quad (3)$$

respectively, where τ_\pm and A_\pm are the constant value parameters and A_+ and A_- have a positive value and a negative one, respectively.

Secondly, the synaptic efficacy notation is calculated by

$$\xi_{ij}(t) = P_{ij}^+(t) f_i(t) + P_{ij}^-(t) f_j(t), \quad (4)$$

At the next step, eligibility trace is obtained from

$$z_{ij}(t + \delta) = z_{ij}(t) \exp\left(-\frac{\delta}{\tau_z}\right) + \frac{\xi_{ij}(t)}{\tau_z}, \quad (5)$$

where τ_z is the time constant for exponential decay. In the SNN, eligibility trace keeps decaying memory of the synaptic efficacy.

Finally, synaptic weight is renewed as

$$w_{ij}(t + \delta) = w_{ij}(t) + \eta R(t) z_{ij}(t + \delta), \quad (6)$$

where η is the learning rate and R is a reward function which is defined from a given task for the SNN. By renewing synaptic weights successively, the desired output, which reflects the input patterns, can be obtained.

B. XOR learning

For showing that the proposed system has an ability of the SNN, the XOR function, which is a classical benchmark problem of the NN, was learned and acquired in the proposed system.

The XOR function relationships between inputs and output are given in Table II. Input signals 0 and 1 were coded by two distinct spike trains of 500 time steps in lengths, which randomly include 50 spikes. The spike intervals follow Poisson distribution. Output signals were coded by the firing rate. The output firing rates for the output ‘1’ is higher than the output firing rate for the output ‘0’. In one learning epoch, the four patterns of (0, 0), (0, 1), (1, 0) and (1, 1) are randomly input for 500 time steps every epoch. Namely, one epoch has 2000 time steps. 200 epochs are repeated for learning.

From the output neuron state and the correct output at t , reward function R is obtained from

$$R(t) = \begin{cases} 1 & f_{output}(t) = 1 \text{ and correct output is } 1 \\ -1 & f_{output}(t) = 1 \text{ and correct output is } 0 \\ 0 & f_{output}(t) = 0, \end{cases} \quad (7)$$

where f_{output} expresses the state of fire of output neuron and correct output is obtained by knowing accurate value. For that, time slots for transmitting of correct input value are prepared.

The synaptic weights between the input and the hidden layers in the range of $-15.0 \leq w_{hi} < 15.0$, where i and h are labels of input and hidden layer neuron respectively. The weights of the synapses between the hidden layer and the output layer in the range of $0.0 \leq w_{oh} < 15.0$, where o is the label of hidden layer neuron. The initial values of the synaptic weights are set with random values in the above ranges by using Linear Feedback Shift Register(LFSR). The LFSR consists of shift registers and XOR logic gates, it is a pseudo random number generator.

IV. DISCUSSION OF EXPERIMENT RESULTS

This section shows experimental results of XOR learning by MSTDPET. Table III gives the values of experiment parameters. Figure 4 shows the proposed system used for experiments. The distances between transmitter and receiver of infrared communication are 0.5 m and there are no obstacle.

TABLE II: XOR function relationships

input 1	input 2	output
0	0	0
0	1	1
1	0	1
1	1	0

TABLE III: Experiment of parameters

parameter	value
η	0.125
δ	1.0
τ	20.0
τ_z	25.0
τ_+	20.0
τ_-	20.0
v_r	-54.0
v_θ	-70.0
A_+	2.0
A_-	-1.0

Figure 5(a) shows the firing rates of the output neurons as a function of epoch number for the fixed input patterns. It is seen from this figure, the fire rate converged when the epoch number was approximately 160. Figure 5(b) shows that the number of total reward as a function of learning epoch. It is seen from Fig. 5(b), number of total reward increased as the epoch number increased. Additionally, the total reward also converged when the epoch number was approximately 160 because fire rates of each input patterns are converged.

Before learning, the output fire rates for a certain input are the almost same as for other inputs. It is seen from Figure 5(a) that the input pattern (0, 1) and (1, 0) fires more than (0, 0) and (1,1) posterior to learning.

The implemented SNN acquired the XOR function successfully in 100% of 20 experiments.

It could be confirmed from above results that the proposed wireless SNN obtained the same result as the wired SNNs. This showed an ability and a possibility to realize WNSN concept in the near future.

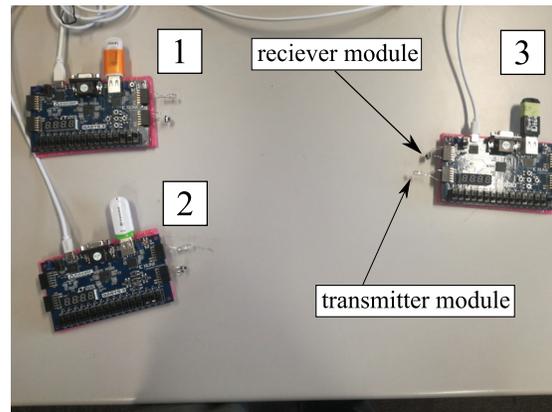


Fig. 4: Proposed system used for experiments

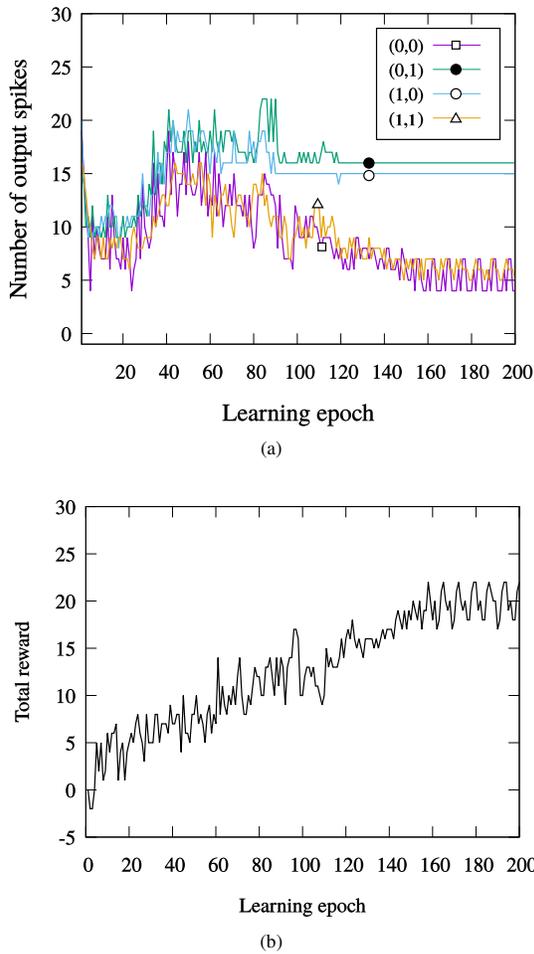


Fig. 5: Experimental result of XOR learning. (a) The number of output neuron spiked as a function of the learning epoch. (b) Total reward as a function of learning epoch.

V. CONCLUSIONS

In this paper, the SNN, which is linked by IR communications, has been proposed. The XOR function could be acquired by applying reinforcement learning, which showed the ability and potential to realize our WNSN concept. For future works to build WNSN, it is necessary to adopt high speed wireless communication module for improving the capacity of spike signal communication and increasing transmission distance.

REFERENCES

[1] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003.
 [2] a. L. Hodgkin and a. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *J Physiol*, vol. 117, no. 4, pp. 500–544, 1952.
 [3] W. Gerstner and W. Kistler, *Spiking Neuron Models: An Introduction*. New York, NY, USA: Cambridge University Press, 2002.
 [4] S. M. Bohte, "The evidence for neural information processing with precise spike-times: A survey," *Natural Computing*, vol. 3, pp. 195–206, 2004.

[5] W. Gerstner, A. K. Kreiter, H. Markram, and A. V. M. Herz, "Neural codes: Firing rates and beyond," *Proceedings of the National Academy of Sciences*, vol. 94, pp. 12740–12741, nov 1997.
 [6] G. B. Stanley, "Reading and writing the neural code," *Nature Neuroscience*, vol. 16, no. 3, pp. 259–263, 2013.
 [7] R. V. Florian, "Reinforcement Learning Through Modulation of Spike-Timing-Dependent Synaptic Plasticity," *Neural Computation*, vol. 19, no. 6, pp. 1468–1502, 2007.
 [8] R. Florian, "A reinforcement learning algorithm for spiking neural networks," in *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNAS'05)*, p. 8 pp., 2005.
 [9] R. V. Florian, "The chronotron: A neuron that learns to fire temporally precise spike patterns," *PLoS ONE*, vol. 7, no. 8, 2012.
 [10] R. Legenstein, D. Pecevski, and W. Maass, "A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback," *PLoS Computational Biology*, vol. 4, no. 10, p. e1000180, 2008.
 [11] M. A. Farries and A. L. Fairhall, "Reinforcement Learning With Modulated Spike Timing Dependent Synaptic Plasticity," *Journal of Neurophysiology*, vol. 98, no. 6, pp. 3648–3665, 2007.
 [12] P. L. Bartlett and J. Baxter, "A Biologically Plausible and Locally Optimal Learning Algorithm for Spiking Neurons," *Information Sciences*, 2000.
 [13] N. Caporale and Y. Dan, "Spike Timing-Dependent Plasticity: A Hebbian Learning Rule," *Annual Review of Neuroscience*, vol. 31, no. 1, pp. 25–46, 2008.
 [14] R. Legenstein, C. Naeger, and W. Maass, "What Can a Neuron Learn with Spike-Timing-Dependent Plasticity?," *Neural Computation*, vol. 17, no. 11, pp. 2337–2382, 2005.
 [15] R. Rubin, R. Monasson, and H. Sompolinsky, "Theory of spike timing-based neural classifiers," *Physical Review Letters*, vol. 105, no. 21, p. 4, 2010.
 [16] M. Shim and P. Li, "Biologically inspired reinforcement learning for mobile robot collision avoidance," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2017-May, pp. 3098–3105, 2017.
 [17] A. Mahadevuni and P. Li, "Navigating mobile robots to target in near shortest time using reinforcement learning with spiking neural networks," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 2017-May, pp. 2243–2250, 2017.
 [18] I. E. Ebgong and P. Mazumder, "CMOS and memristor-based neural network design for position detection," *Proceedings of the IEEE*, vol. 100, no. 6, pp. 2050–2060, 2012.
 [19] N. Zheng and P. Mazumder, "A Low-Power Hardware Architecture for On-Line Supervised Learning in Multi-Layer Spiking Neural Networks," pp. 1–5, 2018.
 [20] L. P. Maguire, T. M. McGinnity, B. Glackin, A. Ghani, A. Belatreche, and J. Harkin, "Challenges for large-scale implementations of spiking neural networks on FPGAs," *Neurocomputing*, vol. 71, pp. 13–29, dec 2007.
 [21] K. Cheung, S. R. Schultz, and W. Luk, "A large-scale spiking neural network accelerator for FPGA systems," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7552 LNCS, no. PART 1, pp. 113–120, 2012.
 [22] D. Neil and S. C. Liu, "Minitaur, an event-driven FPGA-based spiking network accelerator," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 12, pp. 2621–2628, 2014.