



HAL
open science

Real-time implementation of ReSuMe learning in Spiking Neural Network

Yang Xia, Seiji Uenohara, Kazuyuki Aihara, Timothée Levi

► **To cite this version:**

Yang Xia, Seiji Uenohara, Kazuyuki Aihara, Timothée Levi. Real-time implementation of ReSuMe learning in Spiking Neural Network. ICAROB, Jan 2019, Beppu, Japan. hal-02484014

HAL Id: hal-02484014

<https://hal.science/hal-02484014>

Submitted on 19 Feb 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Real-time implementation of ReSuMe learning in Spiking Neural Network

Yang Xia*

*Graduate School of Engineering, The University of Tokyo, Japan
Tokyo, 4-7-1 Komaba, Meguro-ku, 153-8505, Japan*

Seiji Uenohara

*IIS, The University of Tokyo, Japan
Tokyo, 4-7-1 Komaba, Meguro-ku, 153-8505, Japan*

Kazuyuki Aihara

*IIS, The University of Tokyo, Japan
Tokyo, 4-7-1 Komaba, Meguro-ku, 153-8505, Japan*

Timothée Levi

*IIS, The University of Tokyo, Japan
Tokyo, 4-7-1 Komaba, Meguro-ku, 153-8505, Japan*

*E-mail: xiayang@sat.t.u-tokyo.ac.jp, uenohara@sat.t.u-tokyo.ac.jp, aihara@sat.t.u-tokyo.ac.jp, levi@sat.t.u-tokyo.ac.jp
www.iis.u-tokyo.ac.jp/ja/*

Abstract

Neuromorphic systems are designed by mimicking or being inspired by the nervous system, which realizes robust, autonomous, and power-efficient information processing by highly parallel architecture. Supervised learning was proposed as a successful concept of information processing in neural network. Recently, there has been an increasing body of evidence that instruction-based learning is also exploited by the brain. ReSuMe is a proposed algorithm by Ponulak and Kasinski in 2010. It proposes a supervised learning for biologically plausible neurons that reproduce template signals (instructions) or patterns encoded in precisely timed sequences of spikes. Here, we present a real-time ReSuMe learning implementation on FPGA using Leaky Integrate-and-fire (LIF) Spiking Neural Network (SNN). FPGA allows real-time implementation and embedded system. We show that this implementation can make successful the learning on a specific pattern.

Keywords: Spiking neural network, ReSuMe, LIF, FPGA

1. Introduction

Neuromorphic systems are designed by mimicking or being inspired by the nervous system, which realizes

robust, autonomous, and power-efficient information processing by highly parallel architecture. There are three common methods to realize the neuromorphic circuits, which are software^{1 2 3}, analog hardware^{4 5 6 7} and digital

*Typeset names in 10 pt Times Roman, uppercase. Use the footnote to indicate the present or permanent address of the author.

© The 2019 International Conference on Artificial Life and Robotics (ICAROB2019), Jan. 10-13, B-Con Plaza, Beppu, Oita, Japan

hardware^{8 9 10 11 12}. Software can implement simple neuron model but a large scale neural network with complex neuron model cannot be realized in real-time. The power consumption is also quite important (kW for supercomputer). For hardware implementation, compared to analog circuits, digital implementations consume more power but they are convenient to modify, more portable and lower cost for implementation with FPGA devices.

Supervised learning was proposed as a successful concept of information processing in neural network¹³. Recently, there has been an increasing body of evidence that instruction-based learning is also exploited by the brain.

Remote Supervised Method (ReSuMe) is a new supervised learning method for Spiking Neural Networks.

The main reason for the study of ReSuMe is the need to invent an effective learning method to control the movement of people with physical disabilities. However, the in-depth analysis of ReSuMe method shows that this method is not only suitable for motion control tasks, but also suitable for other practical applications, including modeling, identification and control of various non-stationary and non-linear objects^{14 15}.

In this paper, we present a real-time ReSuMe learning implementation on FPGA using Leaky Integrate-and-fire (LIF) Spiking Neural Network (SNN). FPGA allows real-time implementation and embedded system¹⁶.

We show that this implementation can make successful the learning on a specific pattern.

2. Method

This section proposed three methods that applied to the ReSuMe learning implementation on FPGA, which are LIF-neuron model, Postsynaptic potential (PSP) and Spike response model (SRM), as well as ReSuMe algorithm.

2.1. LIF neuron model

The LIF neuron is one of the simplest spiking neuron models. Due to the convenience with which it can be analyzed, simulated especially implemented in digital silicon neural network, the LIF neuron is very popular¹⁷. A neuron is modeled as a “leaky integrator” of its input $I(t)$:

$$\tau_m \frac{dv}{dt} = -v(t) + RI(t) \quad (1)$$

where $v(t)$ represents the membrane potential at time t , τ_m is the membrane time constant and R is the membrane resistance. This equation describes a simple resistor-capacitor (RC) circuit where the leakage term is due to the resistor and the integration of $I(t)$ is due to the capacitor that is in parallel to the resistor. The spiking events are not explicitly modeled in the LIF model. Instead, when the membrane potential $v(t)$ reaches a certain threshold v_{th} (spiking threshold), it is instantaneously reset to a lower value v_r (reset potential) and the leaky integration process described by Eq. (1) starts a new with the initial value v_r .

Consider the case of constant input: $I(t) = I$. We assume $v_r = 0$. The solution of Eq. (1) is then given by:

$$v(t) = RI \left[1 - \exp\left(-\frac{t}{\tau_m}\right) \right] \quad (2)$$

Here $v(t)$ is in an exponential decay. In discrete digital sequential circuit, a linear decay method is usually used to optimize computing process for saving hardware resources.

$$dv = [-v + RI] \frac{dt}{\tau_m} \quad (3)$$

Eq. (3) describes the computing equation of dv , then solution $v = v + dv$ obviously.

2.2. Postsynaptic Potential and Spike Response Model

By considering a single postsynaptic neuron i with a membrane potential u_i at time t , a simplified SRM is defined¹⁷.

$$u_i(t|X, y_i) := \sum_j w_{ij} \sum_{t_j^f \in x_j} \epsilon(t - t_j^f) \quad (4)$$

This SRM signifies a dependence of the neuron’s membrane potential on its presynaptic input pattern X from n_i synapses. An output spike occurs at a time t_j^f . The term of Eq. (4) describes a weighted summation of the pre-synaptic input: the w_{ij} corresponds to the synaptic weight from a presynaptic neuron j , the kernel ϵ refers to the shape of an evoked PSP. The PSP kernel evolves according to

$$\epsilon(s) = \frac{1}{c} \int_{s'=0}^{\infty} \exp\left(-\frac{s'}{\tau_m}\right) \alpha(s - s') ds' \Theta(s) \quad (5)$$

The term $\Theta(s)$ is the Heaviside step function defined such that $\Theta(s) = 1$ for $s \geq 0$ and $\Theta(s) = 0$. Here we

approximate the postsynaptic current's time course by an exponential decay¹⁸.

$$\alpha(s) = \frac{q}{\tau_s} \exp\left(-\frac{s}{\tau_s}\right) \theta(s) \quad (6)$$

For a further simplified computer in digital circuit, the exponential decay $\alpha(s)$ substitute the PSP kernel evolves $\epsilon(s)$ approximately.

2.3. ReSuMe architecture and algorithm

An implementation of ReSuMe in the Liquid State Machine (LSM) architecture is proposed as an example¹⁹. The Liquid State Machine consists of a large, fixed "reservoir" network - the neural microcircuit (NMC) from which the desired output is obtained by training the suitable output connection weights.

In the implementation of ReSuMe method, the original LSM approach has been modified. The modified architecture consists of a set of input neurons N^{in} , the NMC structure, a set of learning neurons N^l with a total number k and a corresponding set of teacher neurons N^d (see Fig.1). NMC receives signal $s^{in}(t)$ from N^{in} and transforms it into a vector of signals $\hat{S}_i^{in}(t)$ which i is presented to the learning neurons $n_i^l \in N^l$. The teacher neurons N^d are not directly connected with any other structure.

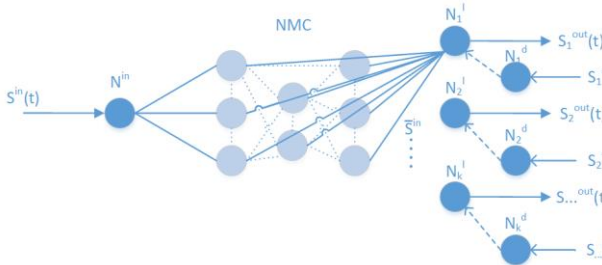


Fig. 1. ReSuMe implemented in the modified Liquid State Machine Architecture.

Since we focus more on the ReSuMe learning implementation itself on this paper, so we generated pre-synapse to the learning neuron as NMC output.

The modification algorithm, which adjusts weights between pre-spike and post-neuron, is applied according to the following simplified equation:

$$\frac{d}{dt} w_{ki}(t) = [S^d(t) - S^l(t)] * [a^d + \int_0^\infty W^d(s^d) S^{in}(t - s^d) ds^d] \quad (5)$$

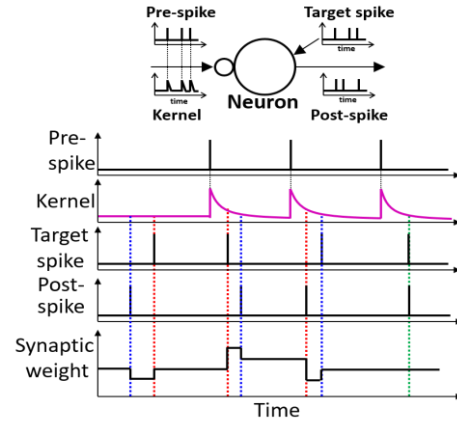


Fig. 2. An example waveform of synaptic weight updating in ReSuMe learning process.

The Fig.2 shows a specific weights update process.

3. Implementation

These section proposed implementation of above methods we introduced above with results showed in waveforms.

3.1. Implementation of LIF neuron

We implement LIF Neuron with VHDL language in FPGA. By adjusting the size of dt and matching different time constant, our LIF Neuron can work at very high clock frequency (10 kHz), which means that its calculation accuracy is very high and the real-time requirement is realized.

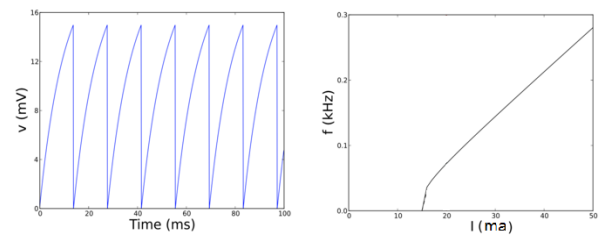


Fig. 3. Stimulation of a LIF neuron by a constant input current: the time-course of the membrane potential, $v(t)$ with 15 mV threshold and 20 mA current I (left); $f-I$ curve for a LIF neuron (right).

3.2. Implementation of PSP and SRM

As we introduced in 2.2, we use simplified exponential decay to achieve RSM and PSP. Because post neurons are connected to 500 pre-synaptic inputs, hardware resources are still unacceptable if 500 exponential operations (even in linear decay) are performed in the same clock cycle. We adopt time division multiplexing, and use two-stage pipeline to complete 500 sets of PSP operations in 1000 clock cycles (actually 501 cycles, remaining standby, theoretically supporting input ceiling of 999), just using one multiplier and one adder.

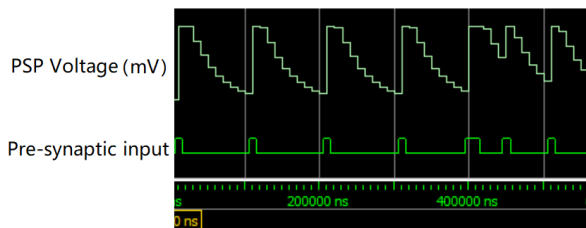


Fig. 4. Waveform of PSP process in which indicates once pre-synaptic input spikes, the PSP starts an exponential decay.

3.3. Implementation of ReSuMe learning

ReSuMe we implemented includes an exponential attenuation (linear attenuation) which attribute to change of parameter k . Each time teacher input spike or post-synaptic neuron spike arrives, weight is updated. 500 exponential operations are performed. We use 500 clock cycles, time division multiplexing to achieve this change, and update weight in real-time so that PSP and RSM modules are used to calculate the correct weight instantly.

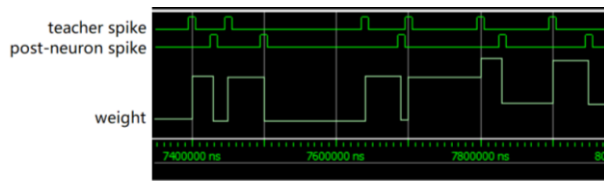


Fig.5. Waveform of ReSuMe weight adjusting process which achieves the ideal functions as showed in Fig.2

3.4. Architecture of ReSuMe learning

The overall hardware architecture of ReSuMe learning is shown in Fig. 6. We use LIF Neuron as post-synaptic neuron, equipped with ReSuMe learning module, and

500 pre-synaptic inputs are connected to post-synaptic neuron. Each connection is operated by PSP and

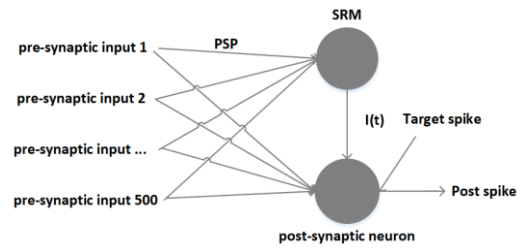


Fig. 6. Architecture of ReSuMe learning.

summarized by RSM.

3.5. Results and discussion

Firstly, we simulate the actual learning waveform of ReSuMe. As shown in Fig. 7, we have completed the learning of 500 input data containing target patterns and random patterns. Post neurons can spike the location of each pattern with a minimal delay in 2-4 time steps. Here time step means cycle of computing and input patterns. For example, if clock frequency is 100 MHz and computing needs 1000 cycles under our implementation, the time step is 0.01 ms. Then we tested different input pattern groups, and completed real-time test and verification on the FPGA.

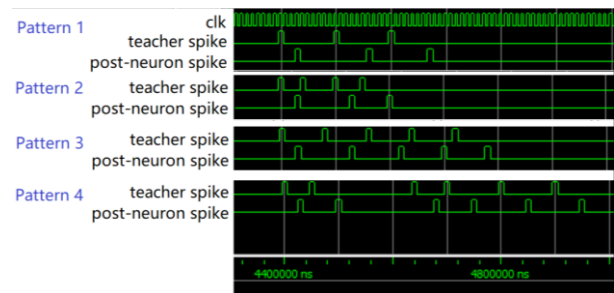


Fig. 7. Waveform of different input patterns and their ReSuMe learning process. The third result shows that the minimum spike interval is 5 time steps.

4. Conclusion

This paper introduces the advantages of using devices such as FPGA to realize the digital neural network. Methods of LIF neuron, PSP and RSM module as well as ReSuMe learning are described and illustrated on hardware implementation. Then the overall framework of ReSuMe learning is elaborated, with the different output data due to different inputs analyzed.

Acknowledgements

Authors would like to thank Prof. Takashi Kohno, from IIS, the University of Tokyo.

References

1. M.-O. Gewaltig and M. Diesmann, M., *NEST (NEural Simulation Tool)*, *Scholarpedia* 2:1430 (2007).
2. M.L. Hines and N.T. Carnevale, *NEURON: a tool for neuroscientists*, *The Neuroscientist* 7 (2001) pp. 123-135.
3. D. Goodman and R. Brette, *The Brian simulator*, *Frontiers in Neuroscience* 3(2) (2009) pp. 192-197.
4. A. Natarajan and J. Hasler, *Hodgkin-Huxley Neuron and FPAA Dynamics*, *IEEE Trans Biomed Circuits Systems*, 12(4) (2018), pp. 918-926.
5. F. Grassia, L. Buhry, T. Levi, J. Tomas, A. Destexhe and S. Saïghi, *Tunable neuromimetic integrated system for emulating cortical neuron models*, *Frontiers in Neurosciences*, 5:134 (2011).
6. N. Qiao, H. Mostafa, F. Corradi, M. Osswald, F. Stefanini, D. Sumislawska, G. Indiveri, *A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128K synapses*, *Frontiers in neuroscience*, 9:141(2015).
7. J. Schemmel, L. Kriener, P. Muller, K. Meier, *An Accelerated Analog Neuromorphic Hardware System Emulating NMDA- and Calcium-Based Non-Linear Dendrites*, *Proceedings of the International Joint Conference on Neural Networks* (2017)
8. R. Wang, G. Cohen, K. Stiefel, T. Hamilton, J. Tapson, and A. Van Schaik, A., *An FPGA Implementation of a Polychronous Spiking Neural Network with Delay Adaptation*, *Frontiers in Neuroscience*, 7:1-14(2013).
9. P. Merolla et al., *A million spiking-neuron integrated circuit with a scalable communication network and interface*, *Science*, 345:6197(2014) pp. 668-673.
10. M. Ambroise, T. Levi, S. Joucla, B. Yvert and S. Saïghi, *Real-time biomimetic Central Pattern Generators in an FPGA for hybrid experiments*, *Frontiers in Neurosciences*, 7:215(2013).
11. Grassia F, Kohno T, Levi T, *Digital hardware implementation of a stochastic two-dimensional neuron model*, *Journal of Physiology Paris*, 110:4(2016), pp. 409-416.
12. Grassia F, Levi T, Doukkali E, Kohno T, *Spike pattern recognition using artificial neuron and Spike-Timing-Dependent Plasticity implemented on a multi-core embedded platform*, *Journal of Artificial Life and Robotics*, 23:2(2018), 200-204
13. S. Ghosh-Dastidar, H. Adeli, *A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection*, *J. Neural Networks*, (2009).
14. A. Kasinski and F. Ponulak, *Experimental demonstration of learning properties of a new supervised learning method for the spiking neural networks*, *Lecture Notes in Computer Science*, Vol. 3696, (2005), pp. 145-153.
15. M. Kraft, A. Kasinski and F. Ponulak, *Design of the spiking neuron having learning capabilities based on FPGA circuits*, *Proceedings of the 3rd International IFAC Workshop on Discrete-Event System Design*, (Rydzyzna, Poland, 2006), pp. 301-306.
16. L. Kechiche, L. Touil, and B. Ouni, *Toward the Implementation of an ASIC-Like System on FPGA for Real-Time Video Processing with Power Reduction*, *International Journal of Reconfigurable Computing*, vol. 2018(2018)
17. W. Gerstner, and W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press (2002).
18. B.Gardner and A.Gruning, *PLOS ONE | DOI : 10.1371 / journal.pone.0161335*(2016).
19. W. Maass, T. Natschlaeger, H. Markram. *Computational models for generic cortical microcircuits*, *Computational Neuroscience: A Comprehensive Approach*, (Chapman and Hall/CRC, Boca Raton, 2004), pp. 575-605.