Exploring NoC jitter effect on simulation of spiking neural networks

Sergei Dytckov, Sushri Sunita Purohit, Masoud Daneshtalab, Juha Plosila
Department of Information Technology
University of Turku
Turku, Finland
{serdyt, susupu, masdan, juplos}@utu.fi

Hannu Tenhunen
School of Information and Communication Technology
Royal Institute of Technology
Stokholm, Sweden
hannu@kth.se

***Abstract*—The major bottleneck in simulation of large-scale neural networks is the communication problem due to one-to-many neuron connectivity. Network-on-Chip concept has been proposed to address the problem. This work explores the drawback that is introduced by interconnection networks – a delay jitter. The preliminary experiment is held in the spiking neural network simulator introducing variable communicational delay to the simulation. The performance degradation is reported.**

Keywords-spiking neural networks; self-organizing maps; network-on-chip

I. INTRODUCTION

Human brain has around 100 billion neurons which are chemically connected to each other. A neuron can be connected to around 10,000 neurons in the circuit. Those connections are called synapses which are usually formed from axons to dendrites. Fig.1 shows the structure of a biological neuron where Axon is the longest nerve fibre that transmits the electrical pulses or spikes away from the neuron cell body to other neurons which then extends into thousands of branches called Dendrites. Synapse converts an activity into electrical effects that excites the activity in the connected neurons. Large-scale artificial neural networks (ANNs) have been used to emulate the information processing function of the brain [1]. Spiking neural networks (SNNs) [2] are a type of ANN, which emulate real biological neural networks, conveying information through the communication of short transient pulses (spikes) between neurons via their synaptic connections. Each neuron maintains an internal membrane potential, which is a function of several parameters as input spikes, associated synaptic weights, current membrane potential, and a constant membrane potential leakage coefficient [2], [3]. A neuron fires (emits a spike to all connected synapses/neurons) when its membrane potential exceeds the neuron's firing threshold value. Understanding and emulating the behavior of the brain has received much attention not only from neuroscientists but also from engineers and computer scientists. While neuroscientists are interested in biophysical models, engineers and computer scientists are more interested in utilizing the brain's powerful computing capability. The dramatic developments in brain science and neuroscience over the past few decades, together with the formidable developments in hardware and software technology, have brought us to the edge of building brain-like functioning devices and systems [4].

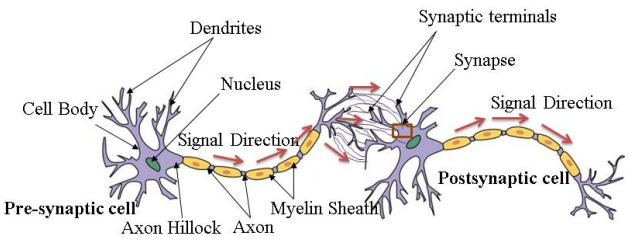

Fig. 1. Simple representation of two biological neuron cells.

Simulation of large-scale networks requires high-performance computational systems. Software based systems provide low speed and don't scale efficiently. The complexity of neuron interconnection is a fundamental bottleneck in hardware emulation platforms. Traditional bus-based and direct wired connections cannot provide mechanisms to overcome this problem. Recently, the network-on-chip (NoC) paradigm has emerged as a promising solution to solve the on-chip communication problems revealed in many-core system-on-chip. NoC architectures are composed of cores, routers, and links which are arranged in a specific topology. In the context of SNNs, the cores refer to the spiking neurons attached to NoC routers and the NoC topology refers to the way those neurons are interconnected across the network. NoC provides parallel transmission of packets. Packet transfer via shared on chip resources leads to varying latency values for each packet transfer. This variation in packet transfer latency is called jitter. The impact of jitter in SNN application is very prominent as it alters the arrival time of the spike packet at the destination neuron and causes information distortion within SNN, which eventually affect the reliability and efficiency of the SNN application. This work tends to explore how the jitter affects the computational properties of SNN.

II. RELATED WORK

Hardware implementation commonly assumes trade-offs between variable precision versus cost and speed. In the neural networks, synaptic weights are the most important variables,

especially in context of learning. Quite a few researches explored reduced synaptic weights precision and the learning properties of it for the classical ANNs [20-22]. SNN as a relatively new concept is less studied in this sense. One work explores genetic algorithm for three bit weights and delays precision [23]. FACETS project claims that four bit synaptic weights are enough for biological simulations [24]. Other study shows that biologically plausible learning remains functional down to two bits weight resolution [25]. From the point of hardware drawbacks, weight resolution is the most explored topic. However, we suppose that communication delay introduces larger error, but it remains unexplored.

Some large-scale projects have indicated the jitter problem. The Fast Analog Computing with Emergent Transient States project (FACETS) is based on mixed (analog-digital) approach [5]. The HICANN (High Input Count Analog Neural Network) is a building block of a system incorporating 512 analog neurons and more than 131,072 synapses with 4-bit SRAMs to store weights. Up to 384 HICANN chips are placed on the wafer and connected through hierarchical busses. Each chip has access to 256 2-bit bus lanes, 8-bit packets with neuron addresses are transmitted serially through them. FACETS network allocates a time slot for a neuron on a specific lane. A receiving neuron determines spike time by a delivery time, the delay is said to introduce an error, but the real impact is not reported.

The Spiking Neural Network Architecture project (SpiNNaker) [6] is based on utilizing Multiprocessor-based approaches. The building block of the system comprises 18 ARM968 processor cores. Each building block can emulate 16,000 biologically plausible neurons with STDP learning in real-time. The interconnection between each node is handled by a NoC using six links, which is wrapped into a triangular lattice; this lattice is then folded onto a surface of a toroid. Spikes are transmitted, as 40-bit packets, serially through the asynchronous multicasting network [15]. SpiNNaker network uses the spike discard policy in case of congestion or significant delay. The spike traffic patterns effect on the network performance is studied [16], but not vice-versa.

EMBRACE (Emulating Biologically InspiRed ArChitectures in hardwarE) utilizes hierarchical (H-NOC) approach, which gives a good trade-off between scalability and power consumption [17]. The H-NOC approach offers a high throughput of spikes per second along with low power consumption of 13mW for a single cluster facility. However, each module contains a fixed amount of neurons but not an optimized amount. Last work on EMBRACE project identifies the problem of a network jitter [7]. They show that jitter creates spike rate deviation that affects the SNN data flow. The unidirectional ring topology is proposed to reach a fixed delay regime for local communications. Rings can be combined into mesh topology for scalability with the assumption that cortical-like modular SNN architecture is used, where neurons in one module mostly have only local connections, thus the traffic in mesh is kept low.

The researches mentioned above allow some error for simulation, but the exact effect is not explored. In this work, we explore how NoC jitter affects a specific SNN architecture

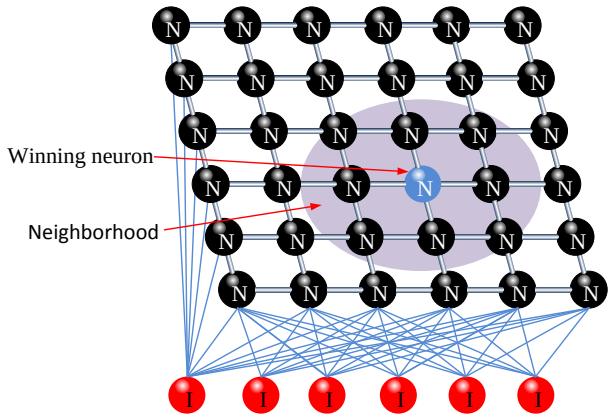

Fig. 2. Self-Organizing Map architecture

called Self-Organizing Maps (SOM). We perform classification experiment with different delay conditions to find the operational range that hardware NoC should provide.

III. SELF-ORGANIZING MAPS

Self-Organizing Maps (SOM) is a computational method for the visualization and analysis of high-dimensional data, especially experimentally acquired information [8]. It was first developed in 1982 by Teuvo Kohonen, therefore SOM is popularly called Kohonen maps. SOM employ an unsupervised learning technique to achieve data classification, data segmentation or vector quantization. SOM typically consist of an input layer and a two-dimensional map of neurons, as shown in Fig. 2. Each neuron in a map also has a neighborhood relation which determines the map's topology. Based on the topology, distance in the map can be defined. SOM provides a topology preserving mapping from the input space to the two-dimensional grid of neurons, which means that the relative distance between the neurons is preserved [9].

SOM has a broad area of application in numerous fields of science and technology where it is used for different pattern recognition, data analysis and data classification. Most promising fields of application are of SOM are: data mining, visualization of statistical data, process analysis, biomedical applications, data analysis. A research was done in the searching of patterns that are used to detect trading signals in Taiwan Stock Index (TAIEX). 36 patterns were established by implementing a 6 by 6 two-dimensional SOM to a time series data of TAIEX. The patterns were analyzed by using a normalized equity curve, for several days to verify whether they transmit profitable signals [10]. SOM was used for the edge detection process in order to reduce image intensity levels. Since the edge detection procedure is a critical step in biomedical image analysis, an efficient mechanism with the satisfying quality of outputs has been proposed. The outputs were verified using the high-resolution computed tomography images [11]. Viscovery SOMine is an explorative data mining commercial application, based on SOMs and statistics. It is a desktop application for explorative data mining, visual cluster analysis, statistical profiling and classification based on SOM and classical statistics in an intuitive workflow environment. In addition to a large number of enterprises, consultants and labs,

hundreds of universities worldwide make use of SOMine for their analytical tasks [12].

IV. Experiment

We perform an experiment on the adaptation of SOM for spiking neurons, which mostly resembles [18]. The simplest Leaky Integrate and Fire neuron model was used

$$\tau_m \frac{dV}{dt} = I(t) - V$$
$$V = 0, \; if \; V \geq \theta \tag{1}$$

where V is membrane potential, $I(t)$ is the current input, τ_{ij} time constant for the membrane potential, θ is threshold for generating a spike.

The input layer consists of the banks of ten spiking neurons that represent one input value. The input values are transformed into a temporal spike sequence within each bank. During one input presentation to the network, every neuron of the input layer spikes only once. Input values are normalized on the [0;1] interval. Each neuron in the input bank is tuned around a point in that interval. The closer the input value to the tuned point of a neuron, the earlier it spikes. This spike sequence then drives the SOM layer. Neurons in map layer have lateral connections to the neighbor, the weights of which are defined with Mexican-hat like function

$$w_{ij} = (1+a)G(\|i-j\|, r) - aG(\|i-j\|, br) \tag{2}$$

where w_{ij} is synaptic connection strength between neuron i and j, a is a magnitude of the negative component of the function, b is a decay of the negative component of the function, r is a radius of the positive component of the function, G is Gaussian function of the distance between neuron i and j. The lateral connections implement the winner-take-all mechanism. A winning neuron, which fires first, excites closest neighbors and suppresses firing in remote neurons. Thus, every input pattern activates some particular area on the map layer, forming the representation of input data on it.

The learning is implemented with Spike-Timing Dependent Plasticity rule (STDP), which is applied to connections between input and map layers. STDP provides a function for the long-term potentiation (LTP) and depression (LTD) of synapses based on the time difference between a single pair of pre and postsynaptic spikes, according to [19]

$$\Delta w_{ij} = \begin{cases} w_{ij} A_+ \exp(-\Delta t / \tau_+), \; if \; \Delta t > 0 \\ -w_{ij} A_- \exp(\Delta t / \tau_-), \; if \; \Delta t < 0 \end{cases} \tag{3}$$

where, A_+ and A_- are both positive and determine the maximum amount of synaptic strengthening and weakening that can occur, respectively, τ_+ and τ_- are time constants determining the range of time in which synaptic strengthening and weakening will occur. The winning neuron typically fires after receiving about a half of spikes from input banks. These connections fall into LTP region, whereas late spikes depress their synapses. To the detailed explanation of input encoding and learning mechanism used, refer to original work [18].

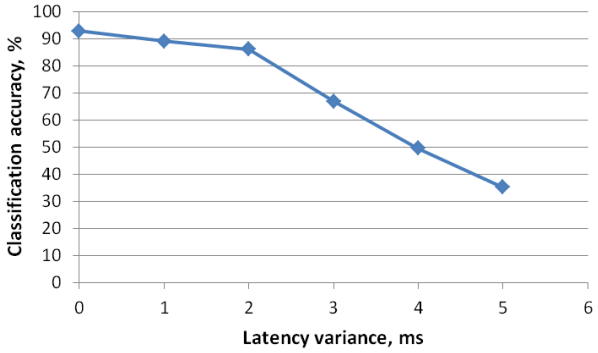

Fig. 3. Classification accuracy for tests with different delay jitter.

The network described above was applied for Iris dataset classification. The dataset consists of three iris plant classes of 50 instances each [13]. In the experiment, the dataset was divided into two equal parts. The one half was used for training the network with 20 000 of input presentations in total, and another for the classification test. The experiment was performed in a Brian simulator [14]. A communicational delay was introduced to the simulation to represent hardware communicational jitter. Every spike had normal distributed delay with a zero mean and variable variance. Fig. 3 shows the classification results for simulations with different delay variances. The results show tolerable accuracy loss with the delay variance until 2ms. Further increase in the variance degrades the classification accuracy significantly.

V. Conclusion and Future Work

We have indicated the problem of variable latency in NoC that can potentially harm the results of simulation of SNN. The problem is only slightly explored by the community and the exact impact is unknown. In this work, we make the experiment with SOM, introducing variable delay into simulation. The result show that the variable delay may be considered as tolerable until 2ms variance, but degrades significantly afterwards. The quite high level of delay tolerance

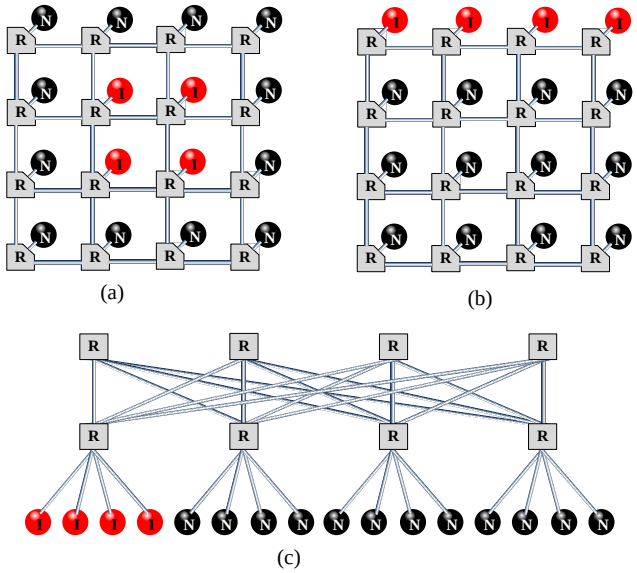

Fig. 4. Mapping of SOM into NoC. (a)-(b) 2-D mesh. (c) Fat tree.

can be explained by the input representation mechanism. Further exploration is required for different neural network architectures with different data encoding techniques.

Another work direction is to perform NoC simulations to explore how different topologies and routing algorithms affect SNN simulation. Our next experiment is to perform simulation of the same SOM architecture mapped on different NoC topologies. We are planning to compare the 2-D mesh and the fat tree topologies. Typical 2-D mesh consists of routers that are connected to four nearest neighbors, forming regular and highly scalable topology. Each router is connected to one local processing unit. In tree topologies, network routers form a tree structure and computing units are connected only to the leaves of a tree. Fat tree increases the number of routers and communication lanes moving up to the root. Fig. 4. shows the principle of the mapping of SOM into hardware NoC. In Fig. 4(c) the mapping of input neurons into one branch helps to transmit spikes in the correct order. Mapping to the mesh topology is somewhat tricky. We identify two major possibilities: (i) to place input neurons in the center, as in Fig. 4(a), (ii) to place the neurons of the map layer preserving the neighborhood connectivity, as in Fig. 4(b). The mapping (i) minimizes the average transmission distance, as during most of training or testing periods all the input neurons fire, but only a winner neuron and its neighbors fire in the map layer. However, at the beginning of a training several neurons can fire close in time. Thus, the mapping (ii) tries to minimize the training error that can be introduced by the spike delays in the lateral connections. As the result of the whole work, the community can get better understanding of communication networks requirements and hardware drawbacks in simulations of SNN.

REFERENCES

[1] S. Grossberg, W. Maass, and H. Markram, “Introduction: spiking neurons in neuroscience and technology,” Neural Networks, vol. 14, no. 6–7, p. 587, 2001.

[2] W. Maass, “Networks of spiking neurons: the third generation of neural network models,” Neural Networks, vol. 10, pp. 1659–1671, 1997.

[3] W. Gerstner, and W. M. Kistler, Spiking Neuron Models Single Neurons, Populations, Plasticity. Cambridge, U.K., Cambridge University Press, 2002.

[4] T. Trappenberg, Fundamentals of Computational Neuroscience, 2nd ed., Oxford, U.K.: Oxford University Press, 2010.

[5] J. Schemmel, J. Fieres, and K. Meier: "Wafer-scale integration of analog neural networks," in *International Joint Conference on Neural Networks*, Hong Kong, (IJCNN), 2008, pp.431-438.

[6] S.B. Furber, D.R. Lester, L.A. Plana, J.D. Garside, E. Painkras, S. Temple, and A.D. Brown: "Overview of the SpiNNaker system architecture," Computers, IEEE Transactions on, vol. 62, no. 12, pp. 2454-2467, Dec. 2013.

[7] S. Pande, F. Morgan, G. Smit, T Bruintjes, J. Rutgers, B. McGinley, S. Cawleys, J. Harkin, and L. McDaid: “Fixed latency on-chip interconnect for hardware spiking neural network architectures,” Parallel Computing, vol. 39, no. 9, pp. 357-371, September 2013.

[8] T. Kohonen, and T. Honkela: “Kohonen network,” Scholarpedia, vol. 2, no. 1, pp. 1568. [Online]. http://dx.doi.org/10.4249/scholarpedia.1568

[9] J. Hollmen: ”Process modeling using the self organizing map,” M.S. thesis, Dep. Comp. Sci., Hels. Univ. Tech., Helsinki, 1996.

[10] S. H. Chen, and H. He, “Searching financial patterns with self-organizing maps,” in *Computational Intelligence in Economics and Finance*, New York, Springer, pp. 203-216, 2003.

[11] L. Gráfová, J. Mareš, A. Procházka, P. Konopásek: “Edge detection in biomedical images using self-organizing maps,” *in Artificial Neural Networks - Architectures and Applications*, Rijeka, Croatia, InTech, January, 2013.

[12] “Viscovery SOMine” Viscovery SOMine 6. [Online]. Available: www.viscovery.net/somine

[13] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” Ann. Eugen., vol. 7, no. 2, pp. 179–188, 1936.

[14] D. Goodman, and R. Brette “Brian: a simulator for spiking neural networks in Python,” Front. Neuroinform., vol. 2, no. 5, 2008.

[15] L.A. Plana, S.B. Furber, S. Temple, M. Khan, Y. Shi, J. Wu, and S. Yang, "A GALS infrastructure for a massively parallel multiprocessor," Design & Test of Computers, IEEE, vol. 24, no. 5, pp. 454-463, 2007.

[16] J. Navaridas, L. A. Plana, J. Miguel-Alonso, M. Luján, and S.B. Furber, “SpiNNaker: impact of traffic locality, causality and burstiness on the performance of the interconnection network,” in *Proceedings of the 7th ACM international conference on Computing frontiers* (CF '10), New York, USA, 2010, pp. 11-20.

[17] S. Carrillo, J. Harkin, L.J. McDaid, F. Morgan, S. Pande, S. Cawley, and B. McGinley: "Scalable hierarchical network-on-chip architecture for spiking neural network hardware implementations," Parallel and Distributed Systems, IEEE Transactions on, vol. 24, no. 12, pp. 2451-2461, 2013

[18] T. Rumbell, S.L. Denham, T. Wennekers: “A spiking self-organizing map combining STDP, oscillations, and continuous learning,” Neural Networks and Learning Systems, IEEE Transactions on, unpublished.

[19] S. Song, K.D. Miller, L.F. Abbott: “Competitive Hebbian learning through spike-timing-dependent synaptic plasticity,” Nat Neurosci., vol. 3, no. 9, pp. 919-926, Sept., 2000.

[20] V.P. Plagianakos, and M.N. Vrahatis, “Training neural networks with threshold activation functions and constrained integer weights,” in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks* (IJCNN), Como, 2000, vol.5, pp.161-166.

[21] B. Jian, C. Yu, and Y. JinShou, “Neural networks with limited precision weights and its application in embedded systems,” in *Second International Workshop on Education Technology and Computer Science* (ETCS), 2010, vol.1, pp.86-91.

[22] M. Hoehfeld, and S.E. Fahlman, “Learning with limited numerical precision using the cascade-correlation algorithm,” Neural Networks, IEEE Transactions on, vol. 3, no. 4, pp.602-611, Jul 1992.

[23] E. Stromatias, “Developing a supervised training algorithm for limited precision feed-forward spiking neural networks,” M.S. thesis, Microelectron. Sys., Univ. of Liverpool, Liverpool, 2011.

[24] T. Pheil, T.C. Potjans, S. Schrader, W. Potjans, J. Schemmel, M. Diesmann, and K. Meier, “Is a 4-bit synaptic weight resolution enough? – constraints on enabling spike-timing dependent plasticity in neuromorphic hardware,” Frontiers in Neuroscience, vol. 6, no. 90, 2012.

[25] D. Roclin, O. Bichler, C. Gamrat, S.J. Thorpe, J.-O. Klein: “Design study of efficient digital order-based STDP neuron implementations for extracting temporal features,” in *International Joint Conference on Neural Networks* (IJCNN), Dallas, 2013 , pp.1-7.