A Memory-Efficient Routing Method for Large-Scale Spiking Neural Networks

Saber Moradi[1], Nabil Imam[2], Rajit Manohar[2], Giacomo Indiveri[1]
[1]Institute of Neuroinformatics
University of Zurich and ETH Zurich, Switzerland
[2]Computer Systems Laboratory
Cornell University, Ithaca, NY 14853, U.S.A.
{saber, ni49, rajit}@csl.cornell.edu, giacomo@ini.uzh.ch

Abstract—**Progress in VLSI technologies is enabling the integration of large numbers of spiking neural network processing modules into compact systems. Asynchronous routing circuits are typically employed to efficiently interface these modules, and configurable memory is usually used to implement synaptic connectivity among them. However, supporting arbitrary network connectivity with conventional routing methods would require prohibitively large memory resources. We propose a two stage routing scheme which minimizes the memory requirements needed to implement scalable and reconfigurable spiking neural networks with bounded connectivity. Our routing methodology trades off network configuration flexibility for routing memory demands and is optimized for the most common and anatomically realistic neural network topologies. We describe and analyze our routing method and present a case study with a large neural network.**

I. Introduction

Promising approaches are being proposed for reducing power consumption in hardware implementations of spiking neural networks and for dramatically increasing their integration densities, for example with the use of memristive devices and advanced Very Large Scale Integration (VLSI) processes [1,2]. These approaches are enabling the development of large-scale neural network systems, composed of a large number of homogeneous neural processors interfaced among each other [3]. Both for multi-chip systems [5-7] and for System-on-Chip (SoC) solutions, an efficient way to route spikes among neurons within and across these processors is to use asynchronous event-based communication protocols. The methodology that is emerging as a common standard for spiking neural network hardware is based on the Address Event Representation (AER) [8,9].

In biological neural systems, each neuron on average makes thousands of connections (synapses) with other neurons. In neuromorphic VLSI, implementing this dense connectivity using dedicated wires is infeasible for any moderately-sized network since it would lead to excessive area requirements for on-chip networks and a prohibitely large number of input-output pads for multi-chip systems. However, semiconductor wire bandwidths (several hundred MHzs to a few GHzs) are orders of magnitude higher than typical operating speeds of neurons (a few Hzs). AER leverages this discrepancy and employs time-division multiplexing to implement dense neural connectivity in silicon using a limited number of shared wires.

In AER, digital address-event packets encode the address and time of spiking neurons. These packets are communicated between neurons as asynchronous streams of binary words. The destinations of AER packets are stored in routing tables (implemented as memory arrays) whose entries define the network connectivity. Typically, synapses outnumber neurons by three or four orders of magnitude and therefore the size of this routing table is a critical factor in determining the area requirements, the power consumption, and the operation speed of a neuromorphic system [4].

In order to determine the scalability of the system, it is particularly important to analyze how the size of the routing table varies with network size. Designing routing schemes that support an arbitrary number of connections results in prohibitively large routing tables. On the other hand putting hard bounds on connectivity numbers in conventional routing methods restricts the type of networks that can be implemented. In this paper, we propose a novel two-stage, tag-based approach to AER routing that reduces memory usage without severely constraining the supported networks.

II. Background

Neural connectivity through AER packets have been previously implemented through shared bus networks [10]. However, bandwidth restrictions limit the scalability of such approaches. Alternatively, local buses between adjacent neurons arranged in one-dimensional [11] or two-dimensional [12] grid network have been proposed. Such an approach can be generalized to multiple hierarchical levels [13], with neurons in different levels of the hierarchy connected via a shared bus.

The memory requirement for storing network connectivity information is one of main obstacles in hardware implementations of large highly-interconnected neural networks. Routing methods can be broadly categorized into two classes based on where routing information is stored: source routing, and destination-tag/distributed routing. In source routing [15], information about the destination of a spike is stored at its source and is attached to the outgoing routing packet. The routers in the routing fabric use this information to direct the packet to its destination. In distributed routing [16], the source address of a spike is attached to the outgoing routing packet and the routing information to reach destinations is stored at the routers. As the packet traverses the routing fabric, it reaches only those nodes that have subscribed to that address.

For a routing fabric to accommodate arbitrary connectivity among n neurons, $O(n)$ entries are required in the routing table, to specify the connections of each neuron. However, the flexibility offered by this kind of routing fabric can be quite wasteful for systems designed to emulate real brain networks. Such networks have common structures and specific features in their connectivity profiles [14,17] that can be exploited to optimize resource usage in routing networks. One of the most notable features in these types of networks is their high degree of clustering, with nodes (neurons) connecting preferentially to others in their local neighborhood. The large density of local connections in brain networks may have several functional and evolutionary benefits such as enhanced communication speeds and minimal wiring and metabolic costs. As we describe in the following section, this feature can have beneficial effects also in the design of memory-efficient routing networks for neuromorphic systems.

III. The Proposed Routing Scheme

Let's consider a network of spiking neurons in which the number of neurons is N and the fan-out of each neuron is F. In a standard routing method, each destination would be encoded with $\log_2(N)$ bits. With F possible destinations, the storage requirement would be of $F\log_2(N)$ bits per neuron. The total number of bits required for such scheme would therefore be $NF\log_2(N)$.

A. Two-stage Routing

In the proposed routing scheme neurons are grouped in clusters of size C, resulting in N/C clusters. To reduce memory requirements, while still supporting large fan-out per neuron, we divide the fan-out operation into two stages. For a fan-out of F per neuron, the first stage is responsible for fan-out of F/M, and the second stage implements a fan-out of M. This two-stage routing scheme is illustrated in Fig. 1. The following steps describe the routing scheme:

- Each neuron transmits F/M copies of its AER packet to an equivalent number of intermediate nodes, using point-to-point routing.
- Each of the N/C intermediate nodes broadcasts its AER packet to C neurons in its end-point cluster.
- Each neuron in the end-point cluster has a set of K *tags* (there are K unique tags per cluster). If the AER packet received matches one of the neuron's tags then the packet is accepted. In this way, an M-way fan-out can be implemented within each cluster ($M \leq F$, $M \leq C$)

Total tag memory: If within each cluster the tags were uniformly distributed, then each tag would be replicated M times, for a total number of KM tag entries. Hence each neuron would contain KM/C tags, each requiring $\log_2(K)$ bits. Note that an alternative is to simply have a bit-vector for each neuron to determine which tag a neuron subscribes to. The storage requirements for this would be K bits.

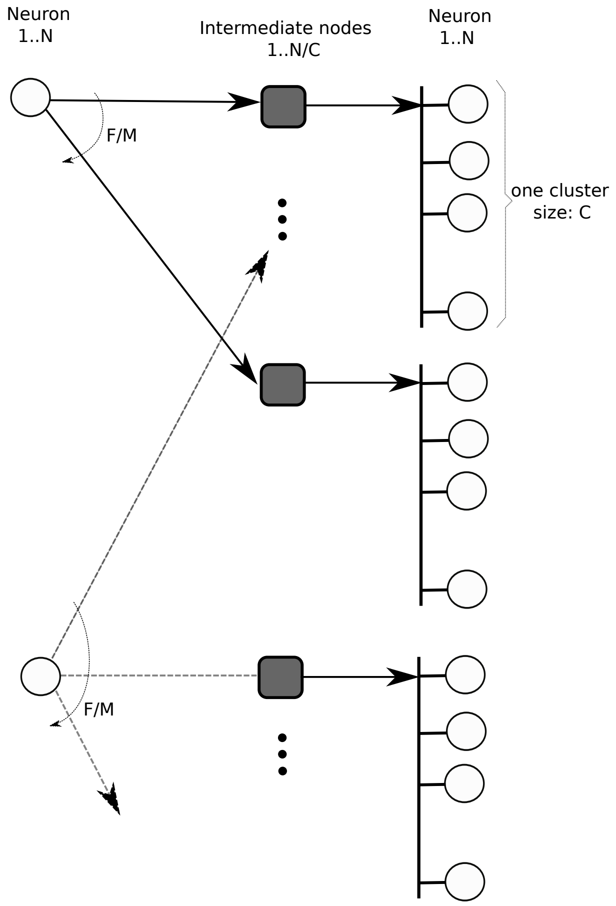

Fig. 1. Two-stage tag-based Routing. Parameter N is the total number of neurons and C is cluster size. Each tag is replicated M times within a cluster

Total sender memory: Sender nodes require F/M entries, each having to route a tag ($\log_2(K)$ bits) to N/C intermediate neurons ($\log_2(N/C)$ bits). The total memory for the sender side of this two-stage routing scheme is therefore $(F/M)(\log_2(K) + \log_2(N/C))$

B. Routing flexibility versus memory trade-offs

For a fixed choice of parameters, any network that can be implemented with K tags can be embedded into one that uses $K+1$ tags. Therefore any network that can be implemented with clusters of size rC can be embedded into one with cluster size C ($r > 1$, $r \in \mathbb{N}$) if both networks have the same number of tags. Hence while larger clusters and fewer tags lead to reduced storage requirements, both larger clusters and fewer tags reduce the flexibility in routing. There is another trade-off between cluster size and number of tags that is captured by the following observation: any network that can be implemented with K tags and clusters of size C can be also embedded in a network with rK tags and clusters of size rC. Hence, it is the ratio $\alpha = K/C$ that matters. If M (the amount of fan-out internal to a cluster) increases, then it is reasonable to expect that the size of the cluster C should also increase. Hence, we

Routing	Storage/neuron
standard	$F\log_2(N)$
two-stage-log	$\sqrt{F\log_2(N)}\cdot 2\sqrt{\log_2(C)}$

TABLE I
ROUTING TABLE STORAGE REQUIREMENTS.

assume that $M = \gamma C$. The parameter $\gamma \in [0,1]$ captures the mean "utilization" of the cluster. In other words, high values of γ imply that an input spike to the cluster is a valid input for most of the neurons in the cluster.

C. Logarithmically encoded tags

For logarithmically encoded tags, the total memory requirement per neuron would be:

$$\begin{aligned} bits &= \frac{KM}{C}\log_2(K) + \frac{F}{M}\left(\log_2(K) + \log_2(N/C)\right) \\ &= \frac{KM}{C}\log_2(K) + \frac{F}{M}\log_2\left(\frac{KN}{C}\right) \\ &= \alpha M\log_2(\alpha C) + \frac{F}{M}\log_2(\alpha N) \end{aligned}$$

Larger values of C increase storage requirements, but they also increase the flexibility supported by the routing network. The parameter M determines the trade-off between point-to-point copying versus flooding. We can minimize the storage as a function of M by differentiating this w.r.t. M. At the optimal point M^* we get:

$$\begin{aligned} 0 &= \alpha\log_2(\alpha C) - \frac{F}{M^{*2}}\log_2(\alpha N) \\ M^* &= \sqrt{\frac{F}{\alpha}\frac{\log_2(\alpha N)}{\log_2(\alpha C)}} \end{aligned}$$

The total number of bits required for this choice of M are:

$$2\sqrt{\alpha F\log_2(\alpha C)\log_2(\alpha N)}$$

If for example we pick the design point $K = C$ ($\alpha = 1$), we have:

$$\begin{aligned} storage/neuron &= 2\sqrt{F\log_2(C)\log_2(N)} \\ M^* &= \sqrt{F\log_2(N)/\log_2(C)} \end{aligned}$$

Although this routing scheme is not as flexible as the standard one it requires significantly fewer bits to represent destination addresses.

However, there is a problem when $M^* > F$, that is when:

$$\sqrt{\frac{F}{\alpha}\frac{\log_2(\alpha N)}{\log_2(\alpha C)}} \quad > \quad F$$

This condition can be true only if $N^{1/F} > C$, when $\alpha = 1$. So, if the clusters are chosen so that $C \geq N^{1/F}$, then we can always pick M^* as a valid design point. This is a very safe constraint: for example even when the total neuron count in the 10^{10} range, a fan-out as small as 10 would require a cluster size of $C \geq 10$ to be able to have an optimal choice of M^*. Since typical fan-out values are actually in the 10^3–10^4 range, this requirement imposes very few constraints on the cluster size. The total number of neurons N would have to be larger than 10^{10^3} before the right hand side of the constraint would be 10 or larger.

The second requirement is that $C \geq M^*$, otherwise the cluster would not have a sufficient number of neurons to support the fan-out anticipated. This means:

$$C \quad \geq \quad \sqrt{\frac{F}{\alpha}\frac{\log_2(\alpha N)}{\log_2(\alpha C)}}$$

which leads to

$$C\sqrt{\log_2(C)} \quad \geq \quad \sqrt{F\log_2(N)} \qquad \text{for } \alpha = 1$$

This constraint is much more restrictive than the first one. For example, if we take typical values of $F = 5000$, and $N = 10^{10}$, then clusters need to be $C \geq 152$. Conversely, if we picked a cluster size $C = 256$ with $\alpha = 1$ (i.e., with 256 tags), then the optimal value of M is $M^* = 144$. The network would require a first-level fan-out of 35, followed by a second cluster-level fan-out of 144 for a total maximum fan-out of 5040 and the storage per neuron would be $424.26\sqrt{\log_2 N}$ bits.

IV. BIOLOGICALLY REALISTIC NETWORKS

The routing scheme we described is suitable to implement networks that have dense local connectivity, such as small-world networks [17] and locally connected random networks (LCRN) [14]. Here, we study an example of a LCRN derived from data of layer II/III of the rat visual cortex [14]. In this area, the probability of having local connections between two neurons is given by a Gaussian function with standard deviation of σ= 3 mm; the neuron density is approximately 75000 neurons/mm^3; the thickness of the cortical layer is 0.3 mm. Therefore the total number of neurons in an area of 5 mm×5 mm is approximately $700,000$. In our analysis, we map the network onto a two-dimensional grid and assume that each neuron makes connections within a circle area with diameter of 5σ around it.

Suppose each neuron has fan-out of 4000. To implement this network with the proposed routing scheme, we first calculate M^* and then the *storage/neuron* figure, for different cluster sizes (C). The results are summarized in Table II. As expected, the memory requirements and the size of the copy circuits increase with increasing cluster size.

In a standard routing algorithm, a fan-out of 5000 per neuron would require 5000 entries for each neuron in the routing table. As illustrated in Fig. 2, our two-stage routing method requires a significantly lower amount of memory even for large network sizes and cluster sizes. For a fixed cluster size, the amount of routing memory required per neuron decreases as

Cluster Size (C)	**Copy-1**	**Copy-2**	**Storage/neuron**
127	38	106	1.42 Kbits
256	40	98	1.53 Kbits
512	43	92	1.63 Kbits
1024	45	88	1.72 Kbits

TABLE II
STORAGE/NEURON REQUIREMENTS FOR DIFFERENT CLUSTER SIZES.

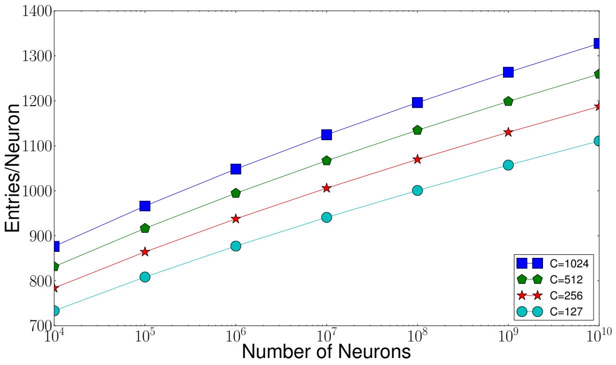

Fig. 2. Number of memory entries per neuron for different cluster sizes (C) in the proposed two-stage routing scheme while $\alpha = 1$ and $F = 5000$.

the network size decreases. The amount of memory required also decreases with decreasing cluster size (with $\alpha = 1$) for a given network size. Smaller cluster sizes, however, provide less connection flexibility in the system.

V. Conclusion

Synapses outnumber neurons by two or three orders of magnitude in typical brain networks. The scalability of neuromorphic systems are therefore restricted by the routing memory required to implement highly interconnected neural networks. In this paper we have presented a novel two-stage routing architecture that minimizes these memory requirements. Our routing method utilizes the existence of dense clusters in typical brain networks to optimize routing memory usage in neuromorphic systems. Using the connectivity profile of the rat visual cortex we showed that a network consisting of 700,000 neurons with 4000 connections each can be implementing in our routing architecture using only 1.72 $Kbits$ of storage per neuron.

Acknowledgment

We acknowledge the financial support of the "SI-CODE" project of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of The European Commission, under FET-Open grant number: FP7-284553, as well as the support of the "neuroP" project of the European Research Council (ERC), grant number: FP7-257219.

References

[1] P. Merolla, *et al.*, "A digital neurosynaptic core using embedded crossbar memory with 45pj per spike in 45nm," in *Custom Integrated Circuits Conference (CICC), 2011 IEEE*, Sept. 2011, pp. 1–4.

[2] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, "STDP and STDP variations with memristors for spiking neuromorphic learning systems," *Frontiers in Neuroscience*, vol. 7, no. 2, 2013.

[3] J. V. Arthur, *et al.*, "Building block of a programmable neuromorphic substrate: A digital neurosynaptic core," in *International Joint Conference on Neural Networks, IJCNN 2012*. IEEE, Jun 2012, pp. 1946–1953.

[4] Bolotin, Evgeny and Cidon, Israel and Ginosar, Ran and Kolodny, Avinoam, "Routing table minimization for irregular mesh NoCs," in *Proceedings of the conference on Design, automation and test in Europe*. Nice, France, IEEE, 2007, pp. 942–947.

[5] R. Silver, K. Boahen, S. Grillner, N. Kopell, and K. Olsen, "Neurotech for neuroscience: unifying concepts, organizing principles, and emerging tools," *Journal of Neuroscience*, vol. 27, no. 44, p. 11807, 2007.

[6] X. Jin, *et al.*, "Modeling spiking neural networks on SpiNNaker," *Computing in Science & Engineering*, vol. 12, no. 5, pp. 91–97, September-October 2010.

[7] S. Choudhary, *et al.*, "Silicon neurons that compute," in *Artificial Neural Networks and Machine Learning – ICANN 2012*, ser. Lecture Notes in Computer Science, A. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, Eds. Springer Berlin / Heidelberg, 2012, vol. 7552, pp. 121–128.

[8] M. Mahowald, *An Analog VLSI System for Stereoscopic Vision*. Boston, MA: Kluwer, 1994.

[9] N. Imam and R. Manohar, "Address-event communication using token-ring mutual exclusion," in *Asynchronous Circuits and Systems (ASYNC), 2011 17th IEEE International Symposium on*. IEEE, 2011, pp. 99–108.

[10] E. Chicca, *et al.*, "A multi-chip pulse-based neuromorphic infrastructure and its application to a model of orientation selectivity," *IEEE Transactions on Circuits and Systems I*, vol. 5, no. 54, pp. 981–993, 2007.

[11] P. Merolla, J. Arthur, B. Shi, and K. Boahen, "Expandable networks for neuromorphic chips," *IEEE Transactions on Circuits and Systems I*, vol. 54, no. 2, pp. 301–311, Feb. 2007.

[12] L. A. Plana, *et al.*, "Spinnaker: Design and implementation of a GALS multicore system-on-chip," *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, vol. 7, no. 4, p. 17, 2011.

[13] S. Joshi, *et al.*, "Scalable event routing in hierarchical neural array architecture with global synaptic connectivity," in *Cellular Nanoscale Networks and Their Applications (CNNA), 2010 12th International Workshop on*. IEEE, 2010, pp. 1–6.

[14] C. Mehring, U. Hehl, M. Kubo, M. Diesmann, and A. Aertsen, "Activity dynamics and propagation of synchronous spiking in locally connected random networks," *Biological cybernetics*, vol. 88, no. 5, pp. 395–408, 2003.

[15] D. P. M. Northmore and J. G. Elias W. Maass and C. M. Bishop, "Pulsed Neural Networks," *MIT Press*, pp.135 -156 1998.

[16] Davies, S., Navaridas, J., Galluppi, F. and Furber, S, "Population-based routing in the SpiNNaker neuromorphic architecture," *IJCNN*, IEEE, , pp. 1-8, 2012 .

[17] D. S. Bassett and E. Bullmore, "Small-world brain networks," *The neuroscientist*, vol. 12, no. 6, pp. 512–523, 2006.