Open Source Journeys With KLM

Akke Toeter[1]

1 Radboud University, Nijmegen, Netherlands

Radboud University

ABSTRACT

Abstract

This research sets out to test whether principles of brain adaptation can be leveraged to increase the radiation resistance of neuromorphic space hardware. Neuromorphic architectures provide energy efficient platforms for AI applications in space. Structural similarities between neuromorphic architectures and the brain may allow them to benefit from brain-inspired design on the topic of damage recovery. Space environments can provide challenging conditions with significant radiation exposure, that may damage neuromorphic space hardware. To explore approaches to mitigate such damages, a simulated radiation test with- and without a brain inspired implementation adaptation is investigated. These adaptations are applied to a spiking neural network (SNN) implementation of a distributed minimum dominating set approximation algorithm by Alipour et al. The experiment is performed using Intel's Lava 0.3.0 Framework.

METHODOLOGY

1 Methodology

The research methodology starts in section 1.1 with a description of the minimum dominating set (MDS) approximation algorithm by Alipour, the SNN implementation of this algorithm by Diehl et al., and the implemented form of brain adaptation [?][?]. Since the overarching research project aims at performing physical radiation tests, section 1.2 specifies the hardware that is used for testing, and how the radiation effects are simulated. The test procedure is detailed in section 1.3.

1.1 Algorithm Selection

The MDS approximation as presented by Alipour et al., is specified in Alg. 1. Next, an SNN implementation of this algorithm is generated using Leaky-Integrate-and-Fire (LIF) neurons. This implementation is created by Diehl et al. [?] using the open-source Lava software framework by Intel. This implementation takes as input connected, triangle-free, planar graphs (E.g. section 1.1). Then it converts these graphs into the specification of an SNN that is encoded in a new graph (E.g. section 1.1). A recursive method then takes a single neuron and converts the encoded SNN that is encoded in the graph into an actual functional SNN that can be run on the simulated on a regular computer.

Fig. 1: Input Graph

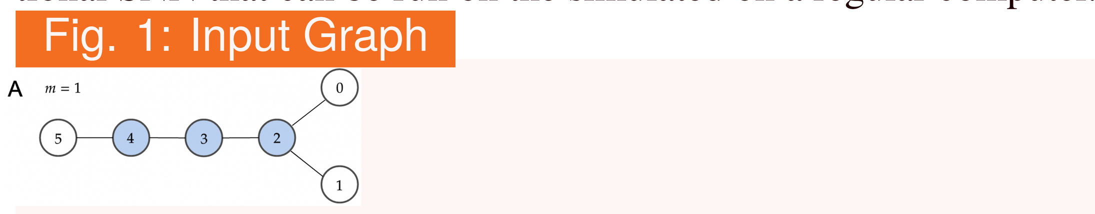
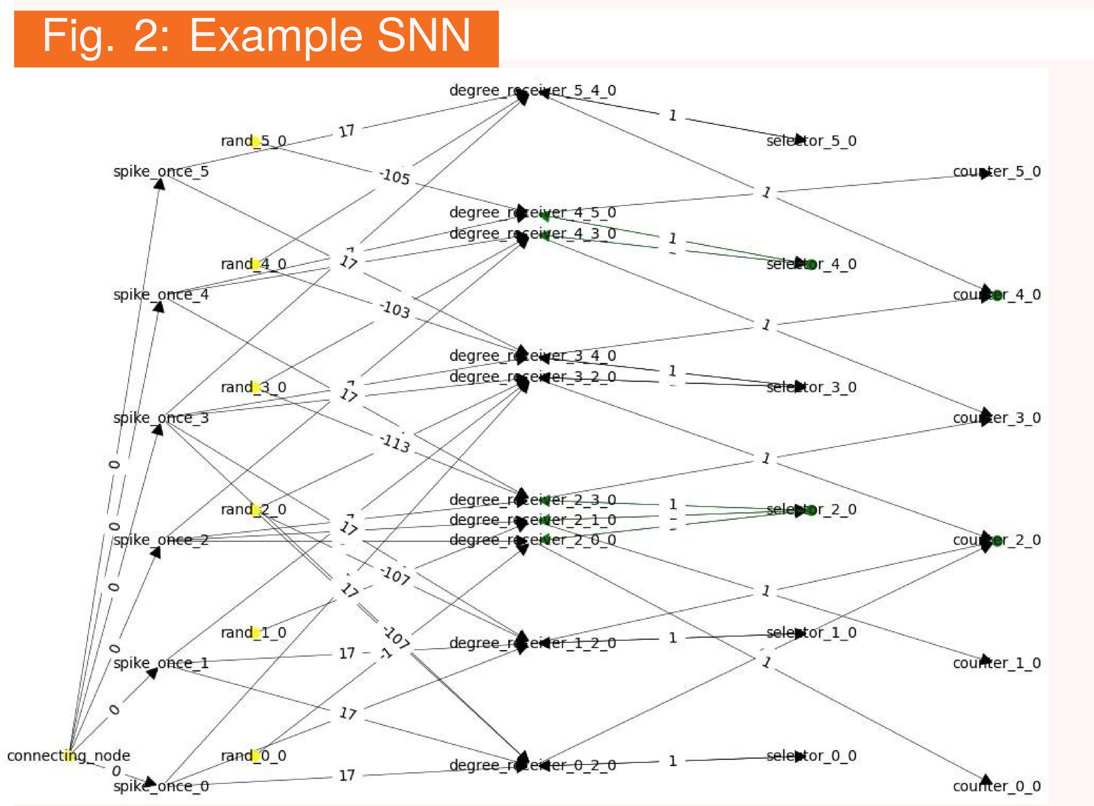


Fig. 2: Example SNN



This SSN implementation of the MDS approximation algorithm is onwards referred to as the default network. This default network is enhanced with strategically placed redundant neurons that are inhibited by the default network neurons. Next, space radiation damage is simulated on the Loihi 2 in the form of random neuron deaths. These neuron deaths imply that a neuron is removed from the SNN, which can occur in both the default network and the set of redundancy neurons. If a default network neuron dies, the inhibition to the redundant neuron should be removed, causing the SNN to use the alternative neural pathway to recover from this damage. This usage of the alternative neural pathway simulates the multi-pathway structures found in brains of higher vertebrates and is proposed as the main adaptive mechanism in response to damage.

1.2 Hardware

At the time of writing, no single-event effect (SEEs) propagation mechanisms are identified for space radiation exposure on the Loihi 1 & 2 neuromorphic chips, a high-level software simulation of these single-event effects is performed. This is done by assuming that the non-neuromorphic components of the chips are performing nominally, and that the SEEs propagate from, for example, transient bit-flips, towards neuronal and synaptic parameter changes. The first assumptions may be accurate if local radiation hardening and redundancy is applied to the non-neuromorphic components. Weight and/or energy saving could be a motivation to apply these radiation counter-measures sparsely. The second assumption is based on the digital nature of the components that make up the neural components of the Loihi.

1.3 Testing

To test whether the brain adaptation implementation may be used to increase radiation robustness of neuromorphic space hardware, it is compared to a baseline without brain adaptation.

1.3.1 Metrics

The metrics of the comparison are:

1. *Radiation Robustness* - a percentage score indicating the ratio of successful solution generation on random input graphs.
2. *Neuronal & Synaptic Overcapacity* - a factor from 0 to n, indicating the ratio of redundant neurons and synapses with respect to the original implementation without adaptation implementation.
3. *Energy Efficiency* - the number of spikes consumed by implementations.

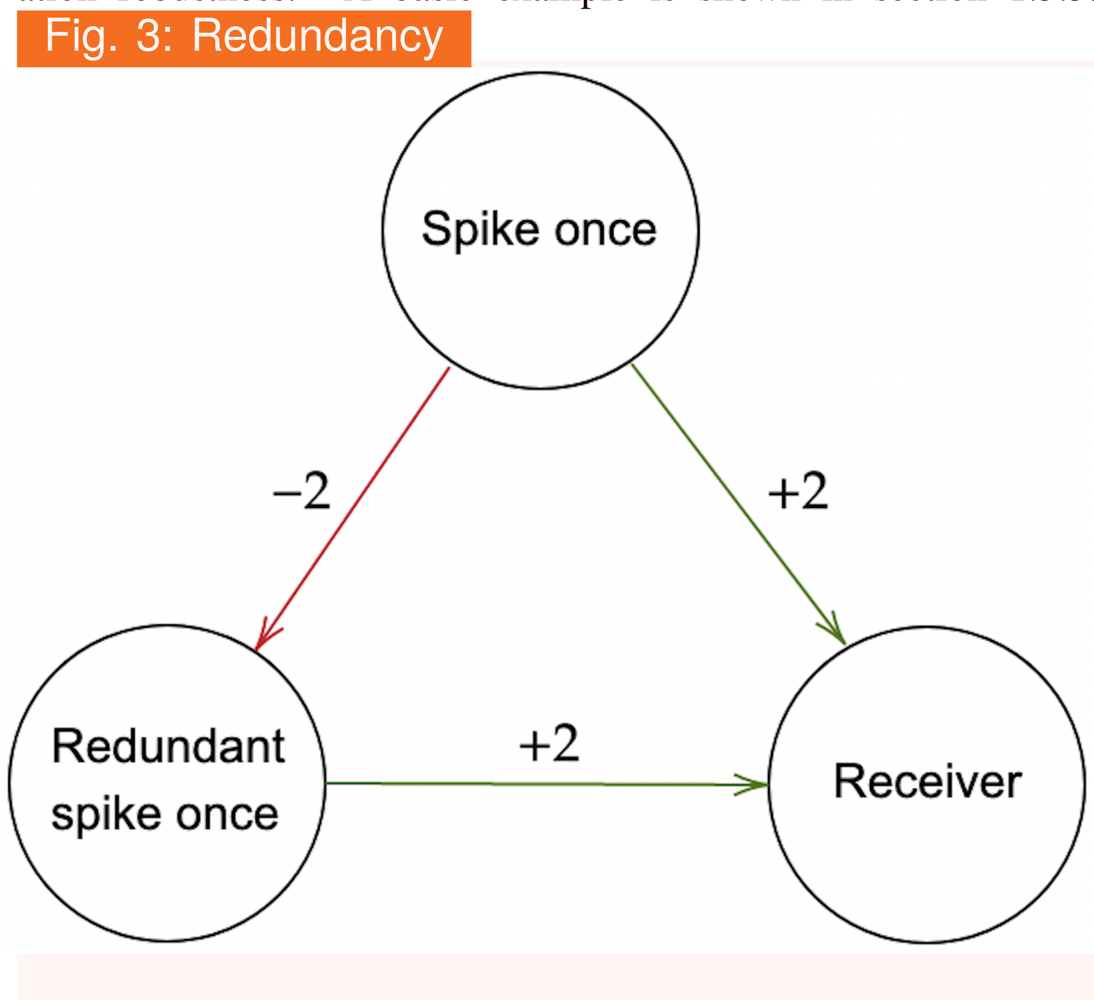
1.3.2 Simulated Radiation Damage

This work simulates radiation damage propagation of SEEs as neuron deaths by setting the neuron thresholds $vth = 1000[V]$ at from the start of the simulation. The 1000 [V] is arbitrary yet large enough to prevent spiking. Transient effects are ignored along with neuron property changes in $\delta u, \delta v, bias$, synaptic death, and synaptic property changes in: $sign, weight$.

1.3.3 Brain Adaptation Mechanism

The selected SNN implementation is enhanced with redundant neurons and neuronal pathways to realise simulated radiation robustness. A basic example is shown in section 1.3.3.

Fig. 3: Redundancy



The alternative neural pathway renders a single neuron $original_i$ with a input synapses, and b output synapses, redundant at the cost of $2a + 2b$ synapses and 2 neurons. This cost can be reduced by implementing a controller that scans the network and manually redirects neural pathways to a smaller buffer network. However, that shifts the radiation robustness problem as that controller may also endure SEEs. One can also add triple/n-factor redundancy by including more redundant spike_once neurons that are inhibited by the original spike_once neuron and each other. This induces an additional delay of t=1 time step per redundant neuron.

RESULTS

2 Results

2.1 Radiation Robustness

Table 1: Average fraction of correct outputs of the SNN implementation with and without adaptation. Neuron deaths can occur in default and redundant LIF neurons.

	Neuron Death Percentage in SNN		
Adaptation	0% 5%	10%	25%
Without	1	1	0
With	1	1	0.5

2.2 Neuronal, Synaptic & Energy Costs

Table 2: The multiple of the amount of neurons and synapses used by adaptation implementation, compared to the default SNN implementation.

	Average Cost of Adaptation Mechanism		
Graph Size	Neuronal	Synaptic	Spikes
3	5.7	8.3	32
4	5.8	10	37.5
5	5.8	10.2	64

Note this is the average overcapacity for $m = 0$ and $m = 1$ combined.

POLICY RECOMMENDATIONS

To facilitate a smooth transformation towards a COSS business model the following policy recommendations are included:

- Apply the temporary reduction of workforce size to nimbly convert KLM towards an COSS business model, incorporate the COSS business model in the hiring strategy when KLM is able to expand again and hire an additional cyber security-, data analytics- and legal team to ensure a safe transition from proprietary codebase to open source codebase.
- Design the COSS to set up open application programmable interfaces (APIs) in cooperation with the open source community to enable companies and community to interact with KLM through the APIs.
- Convert the annual amortisation method of software assets in KLM from the straight line method to the dedicated software valuation model for software as provided by M. Ben-Menachem and I. Gavious [1].

These three steps enable KLM to pivot to a COSS business model to successfully navigate the challenges of the early 21st century and reaffirm its pioneering position.

References

1. M. Ben-Menachem and I. Gavious, 2007, Accounting Software Assets:A Valuation Model for Software, Ben-Gurion University.
2. Royal Dutch Airlines KLM Annual Report 2019.
3. S. Shahrivar et al., 2018, A business model for commercial open source software: A systematic literature review, Tarbiat Modares University.
4. G. Schryen and R. Kadura, 2009, Open Source vs. Closed Source Software: Towards Measuring Security.

Acknowledgments: Dr. W.F.G. Haselager for suggesting open source software as topic for a research brief.

e-mail: akke.toeter@radboud.student.nl