SAND2019-3140C

Sandia National Laboratories

Exploring Applications of Random Walks Using Neural Algorithms

Leah Reeder*, Aaron J. Hill, James B. Aimone, William Severa

Introduction

Neuromorphic computing combines neural inspired algorithms with brain-like architectures to achieve low-power solutions to challenging computing problems [3, 4].

These computing problems have been on the forefront of scientific computing for decades, and with the power benefits and efficiencies that non-traditional computing gives, methods like neuromorphic computing can revolutionize the industry [1].

Background

• Random Walks

A random walk is the movement a particle takes in a diffusion scheme. If a particle is at a location x, then it will move to the left ($x - 1$) with a certain probability p, and to the right ($x + 1$) with a probability $1 - p = q$.

This motion can be modeled with a well known Partial Differential Equation (PDE), known as the diffusion (or heat) equation. This is some summarized steps of Chapter 5 of [6]. We will show this relationship here:

It can be shown that the 1-D random walk case can be expressed with a binomial distribution. The probability event that the particle has taken m steps to the right at time n can be denoted as

$$p(m,n) = \binom{n}{r} p^r q^{n-r}, \text{ where } r = \frac{1}{2}(n - m).$$

Let the function $u(x, t)$ represent the probability that the particle lies in an interval centered at x at time t with a width $2\Delta x$. Then, we can denote $u(x, t)$ as the space- and time-scaled probability function

$$u(x,t) = \frac{p(\frac{x}{\Delta x}, \frac{t}{\Delta t})}{2\,(\Delta x)}$$

We can then use the Central Limit Theorem and approximate this with the normal distribution. The resulting gaussian,

$$u(x,t) = \frac{e^{-\frac{x^2}{4Dt}}}{\sqrt{4\pi D t}}$$

Where D is the diffusion coefficient and is defined as $D = \frac{\Delta x^2}{2(\Delta t)}$. This is a solution to the diffusion equation,

$$\frac{\partial u}{\partial t} = D\,\frac{\partial^2 u}{\partial x^2}$$

• Spiking Neural Algorithm – Density Method

The spiking neural algorithm discussed here was first presented in [5]. This algorithm was designed to conduct random walks, inherently solving the diffusion equation stochastically using spiking neuromorphic architectures. The density method, depicted in Figure 1, works by placing an embedded circuit of neurons at each node in the graph. Rather than always tracking where each particle is, we can count the number of particles at each node at any given timestep.

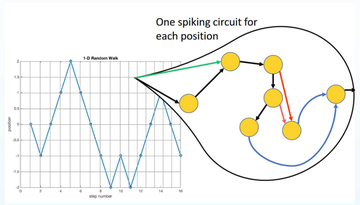


Figure 1 Depiction of how density method is applied on a graph

The random walk is conducted by initially placing walkers at selected points in the graph. When a walker is at a node, the embedded circuit determines which node the walker will go to at the next time step. Every circuit contains a set of stochastic neurons that each correspond to a possible direction. A particle is signaled to move when one of those neurons spikes.

Random Walk Applications

• Image Analysis

By using an image as the underlying graph structure, we can conduct random walks on images. We can conduct random walks on images. We can use the walkers to replicate the image by using the color of each pixel to determine the directional probability at each node. A noise constant can be added in order to create a biased effect for the walkers.

An example of a random walk conducted on an image using the density method can be seen in Figure 2. The darker color corresponds to a larger number of walkers at that node.

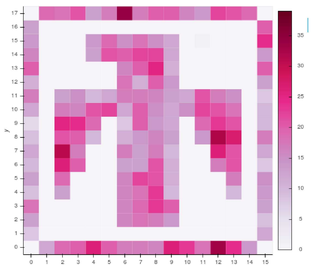


Figure 2 Random walk replication of an image. Darker colors correspond to more walkers at that node.

• Capacitance

It can be shown in [2] that a random walk can be used to find the charge density of a capacitor. Using the charge density, we can find capacitance. This random walk application is different from the others because it requires high connectivity. This would require a significant amount of neurons, and is not very feasible with the way our algorithm is set up. Figure 3 contains an example of how a random walk can be conducted to calculate capacitance.

Due to this potential memory problem, we are considering designing alternate hierarchical network connections, which could also be implemented in previous applications as well in order to speed up calculations.

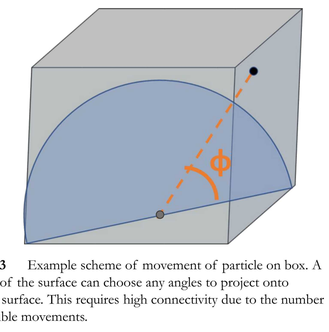


Figure 3 Example scheme of movement of particle on box. A particle of the surface can choose any angles to project onto another surface. This requires high connectivity due to the number of possible movements.

• Radiation Transport

For a 1-D radiation transport scheme, we consider random movement throughout a slab. Particles are either absorbed or reflected at random spatial intervals throughout the slab. This results in 3 random numbers: distance traveled, direction, and rate at which the particle is absorbed.

We are working on mapping this to our density method, where population of neurons correspond to spatial locations. The spikes between locations represents particle movement. This has potential for a lot of non-local jumps, which is challenging for our model. There are also multiple random decisions to consider. Figure 4 contains a potential neural scheme for 1D radiation transport.

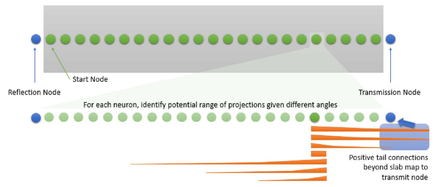


Figure 4 A potential neural model for radiation transport. For each neuron, identify all possible projections in the available range. If the tail is negative, it corresponds to a reflection where positive tail corresponds to transmitting node beyond slab.

Extending Algorithm

As presented, the density method supported 1- and 2-dimensional graphs, which corresponded to 2 and 4 possible directions for the walkers. We were able to generalize this to work on larger graphs.

To do this, we developed a stochastic algorithm for the spiking neurons in the embedded circuit. We created a probability tree, depicted in Figure 5, that propagates spikes down the tree until they reach the output gate neurons at the bottom. Figure 6 shows which neurons need to spike in order to obtain each direction.

The output neurons each correspond to a possible direction. Thus, the tree in Figure 5 corresponds to 8 possible directions. To ensure only one output neuron spikes, we created delays throughout the tree as well as negatively weighted spikes, to stop other neurons from spiking.

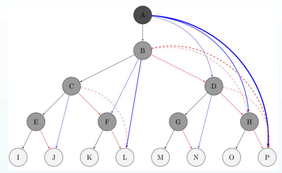


Figure 5 Probability tree used in stochastic algorithm. Nodes I-P correspond to 8 directions.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Direction
1	1	1	0	1	0	-	0	1	0	-	0	-	-	-	0	Direction 1
1	1	1	0	*	0	-	0	-	1	-	0	-	-	-	0	Direction 2
1	1	*	0	-	1	-	0	-	-	1	0	-	-	-	0	Direction 3
1	1	*	0	-	*	-	0	-	-	-	1	-	-	-	0	Direction 4
1	0	-	1	-	-	1	0	-	-	-	-	1	0	-	0	Direction 5
1	0	-	1	-	-	*	0	-	-	-	-	-	1	-	0	Direction 6
1	0	-	*	-	-	-	1	-	-	-	-	-	-	1	0	Direction 7
1	0	-	*	-	-	-	*	-	-	-	-	-	-	-	1	Direction 8

Figure 6 8 direction scheme showing which nodes need to spike (denoted 1) to signal a specific neuron/direction.

IBM's TrueNorth – Performance Results

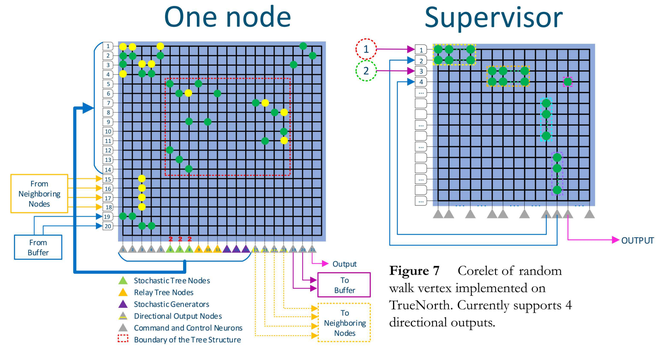


Figure 7 Corelet of random walk vertex implemented on TrueNorth. Currently supports 4 directional outputs.

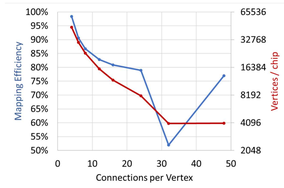
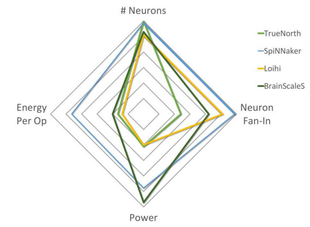


Figure 8 The total vertex size scales with the number of outputs desired. Mapping efficiency occurs for low connectivity – why current applications are challenging.

Figure 9 TrueNorth is efficient for implementing density method random walk algorithm. Ideal because of low power and low energy per operation with low neuron fan-in. Downsides are memory induced: high number of neurons required.



Citations

[1] A. Calimera, E. Mach, and M. Poncino, *The human brain project and neuromorphic computing*, Functional neurology, 28 (2013), p. 191.

[2] M. Mascagni and N. A. Simonov. *The random walk on the boundary method for calculating capacitance*, Journal of Computational physics, 195 (2004), pp. 465 - 473.

[3] D. Monroe, *Neuromorphic computing gets ready for the (really) big time*, Commun. ACM, 57 (2014), pp. 13-115

[4] C. D. Schuman, T. E. Potok, R. M. Paton, J. D. Birdwell, et al., *A survey of neuromorphic computing and neural networks in hardware*, arXiv preprint arXiv:1705.06963, (2017).

[5] W. Severa, R. Lehoucq, O. Parekh, and J.B. Aimone, *Spiking neural algorithms for markov process random walk*, arXiv preprint arXiv:1805.00509, (2018).

[6] L. Sjogren, *Chapter 2 : Random walks* [pdf].

[7] W. Zachary, *An information flow model for conflict and fission in small groups*, Journal of anthropological research, 33 (1976).

*Leah Reeder
lreeder@sandia.gov
720-239-3431

This research was funded by the laboratory Directed Research and Development program and Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology &Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND No. _____