SAND2019-7007C

A Resurgence in Neuromorphic Architectures Enabling Remote Sensing Computation

Craig M. Vineyard, William M. Severa, Matthew J. Kagie, Andrew J. Scholand, and Park Hays
Sandia National Laboratories
Albuquerque, NM, USA
Email: {cmviney, wmsever, mjkagie, ajschol, phays}@sandia.gov

***Abstract*—Technological advances have enabled exponential growth in both sensor data collection, as well as computational processing. However, as a limiting factor, the transmission bandwidth in between a space-based sensor and a ground station processing center has not seen the same growth. A resolution to this bandwidth limitation is to move the processing to the sensor, but doing so faces size, weight, and power operational constraints. Different physical constraints on processor manufacturing are spurring a resurgence in neuromorphic approaches amenable to the space-based operational environment. Here we describe historical trends in computer architecture and the implications for neuromorphic computing, as well as give an overview of how remote sensing applications may be impacted by this emerging direction for computing.**

***Keywords*-Neuromorphic computing; Neural Networks; Computer Architectures; Deep Learning; Remote Sensing**

I. Introduction

In a classic computer architecture, the von Neumann bottleneck describes the computing limitation arising due to communication limits between processing and memory. This need to move data from one location to another for ensuing computation is analogous to a key challenge facing space-based remote sensing paradigms. Great advances in sensor technologies have enabled faster collection of larger regions at greater resolution. And while computational advances have also seen exponential growth, the limiting factor is transmitting information from the space-born sensor to earth for the subsequent signal processing.

Rather than the von Neumann bottleneck, space-based remote sensing faces a transmission bottleneck from the sensor to ground station processing. A remedy to this challenge would be to move the computation to the sensor, but doing so faces size, weight, and power (SWaP) constraints. The large scale data centers enabling impressive computing capabilities require resource budgets unreasonable for space collection assets.

Due to the arrival of physical scaling limits, the design of computer architectures is now looking to novel, more efficient designs. By reflecting upon trends in the history of computer architecture, we see how these advances are enabling the resurgence in neuromorphic computing and how it can play an enabling role in remote sensing. As follows, we provide a brief historical synopsis of the trends in computing followed by a description of remote sensing applications and how the future directions of neuromorphic computing can impact remote sensing computation with some benchmark results highlighting the potential impact.

II. Background

A. Computer Architecture History

With the invention of the transistor and the ensuing growth in the microelectronics industry, the field of computer architecture saw a flurry of activity in the 1960s through early 2000s exploring how to maximize the utility of the underlying hardware. Central to these advances, the Moore's Law phenomenology enabled a doubling of transistor density every two years (note Moore's original forecast stated every year but was revised in 1975) [1]. Coupled with this growth in computational density, Dennard scaling observed that with the reduction in scale of transistor dimensions, power density stays constant [2]. Consequently, together this equates to doubling computational performance for the same power budget.

Over this time, computer architects explored several design choices for how to best utilize this boon in underlying computation infrastructure. One of the first design stabilizations was the development of a core instruction set architecture (ISA). An ISA provides a standard by which software communicates with and specifies the actions for the underlying hardware to take. Integral to the ISA is a coupling with the design of system infrastructure such as the width of communication buses, the size of addressable memory, as well as how many unique instructions may be encoded.

Intrinsic to the architectural exploration of how many instructions may be incorporated into a model is the complexity of the instructions. In other words, simpler instructions encapsulate simpler operations such as additions, and more complex instructions often necessitate a compound set of simple instructions such as performing multiplication as a sequence of additions. This complexity argument took hold in the form of whether to use a reduced instruction set computer (RISC) or a complex instruction set computer (CISC), and has performance implications for the execution hardware. In particular, this impacts the size of the

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.

instruction store, sizes of memory caches, the sophistication of the program interpreter, as well as runtimes. The Intel x86 architecture is a prominent CISC architecture, and many mobile and non-personal computer devices employ some form of RISC processor. Other designs have also explored specialized instruction types, such as very long instruction word (VLIW), to optimize hardware designs and often target the acceleration of certain applications.

Beyond how to encode instructions and the accompanying storage and communication choices, computer architectures in this era also explored parallelization of processing. This lead to the development of instruction level parallelism (ILP), as well as pipelining. These techniques strive to alleviate the imbalance between processing speeds and communication limitations by queuing up predicted next data or instructions for the processor to compute upon. Doing so introduces a great deal of sophistication for forecasting the program and data flow of a computation. Additionally, the expansion to multiple cores furthers parallelism. However, not all computation can be fully parallelized and rather has some serial steps which limit the achievable maximum parallelization of the entire application. This bound is observed as Amdahyl's Law.

Over time, the enabling physical laws reached an end, or are nearing one. Around 2010, Dennard scaling reached its pinnacle. And Moore's Law has been slowing significantly from the year 2000 onwards. Together, while these two laws were both incredibly enabling in revolutionary computer architectures, they also are still shaping current computer designs today. The pursuit of parallelization of processing as a method of addressing limits to clock frequency is constrained by the ability to power a large amount of cores. Unable to advance computing by simply performing more computation over time, whether due to the speed of execution or the parallelization of the computation, introduces a pursuit of alternative computing paradigms. In particular, next we will describe neuromorphic computing and how such approaches may be impactful for the domain of remote sensing. For a more detailed history of computer architecture refer to [3], [4].

B. Neuromorphic Computing

The history of neuromorphic computing parallels much of the progression of the field of computing at large. Inspired by the computational feats brains perform, researchers have explored creating hardware and algorithms mimicking neural principles as long as computing machines have been constructed. Figure 1 illustrates a timeline highlighting several prominent neuromorphic pursuits over time.

In 1943, mathematician and logician researchers Warren McCulloch and Walter Pitt created a foundational mathematical model of neural processing, the namesake McCulloch-Pitt neuron [5]. And subsequently, hardware was developed using manufacturing capabilities of the time instantiating their model. Extending this model lead to the Perceptron model and its hardware instantiation in the form of the Mark 1 perceptron [6]. With mathematical advances resolving critiques presented by Minsky and Papert [7], the 1980s saw the prominence of the backpropagation learning rule [8], as well as the development of the foundational convolutional neural network model [9]. The 1980s and early 1990s saw a corresponding resurgence in neuromorphic hardware development with several efforts lead by Bell Labs [10] [11]. Additionally, Carver Mead pioneered the usage of advanced manufacturing technologies such as analog VLSI to develop neuromorphic systems [12]. In the 2000s, there has been a global resurgence in large scale neuromorphic systems. These efforts include SpiNNaker [13], IBM TrueNorth [14], and Intel Loihi [15]. Additionally, many domain specific architectures and accelerators have also been emerging such as the Google Tensor Processing Unit and the Intel Neural Compute Stick.

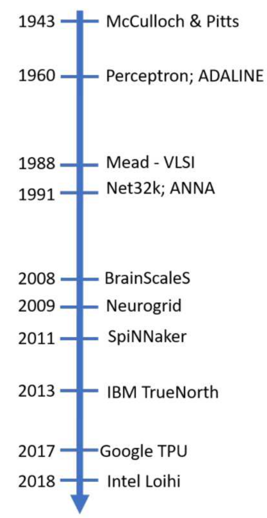

Figure 1: A non-exhaustive timeline depicting several prominent architectural approaches in the pursuit of neuromorphic computing

The incredible advances general purpose computing approaches have achieved in parallel to the pursuit of neuromorphic approaches have historically overshadowed neural-inspired computing due to being the prominent mode of computing. However, the progression of advances employed in general processors has followed many of the trends fundamental to neuromorphic, and computing pioneer John von Neumann identified many of these tenets in his unfinished 1958 last work "The Computer and the Brain" [16].

Similar to the CISC versus RISC explorations, various neuromorphic architectures explore how complex of a neuron is needed. In neuromorphic approaches, this exploration includes neuron models (such as variations to the leaky-integrate and fire (LIF) neuron), as well as the degree of connectivity between these computational units. The

TrueNorth architecture provides one million neurons per chip, but constrains the connectivity from one neuron to another to 256. Alternatively, the SpiNNaker architecture allows arbitrary connectivity. And the Loihi architecture employs configurable complexity as a resource tradeoff enabling a greater number of simpler neurons or a smaller number of more complex neurons.

Just as parallelism has been pursued in traditional architectures, the intrinsic parallelism of the brain motivates neuromorphic architectures with many simple, parallel processing neural units. To make use of these parallel units, analogous to the architectural efforts of pipelining and ILP, neuromorphic architecture considerations often explore costs associated with moving around data or weight parameters [17]. For example, in a convolutional neural network (CNN), computing the namesake convolution operation requires either keeping the convolution filter weights in memory and passing a cascade of inputs through, or alternatively performing all of the processing for the various filters on an input patch before loading the next input patch. Additionally, many emerging architectural approaches are exploring the co-location of memory and processing as a non von Neumann computing architecture.

While digital CMOS technology has been the prevalent device technology in computing, neuromorphic approaches have historically explored the merits of alternatives and continue to do so as digital limitations are being reached. Accordingly, neuromorphic approaches are being pursued based upon both digital and analog computation, as well as looking towards novel technologies such as memristors and optical approaches. These varying architectural pursuits strive after different computational advantages. For example, analog processing as well as optical communication both pursue accelerated computation. Additionally, rather that operating in a traditional clocked manner, some neuromorphic architectures operate asynchronously or in an event-driven manner. Doing so often enables energy savings as the architectures are able to only perform computation as needed. Relatedly, spiking neuromorphic architectures employ an event-driven communication paradigm where neurons which exceed their activation threshold then transmit single bits, spikes. Additional information may be represented by the timing of spikes as well as the associated encoding scheme employed.

Effectively, while classic approaches to computation have reached limitations to scaling laws, driving the need for alternative non von Neumann approaches and domain specific architectures, neuromorphic computing has experienced a resurgence as a promising answer. While not necessarily an optimal approach to all computational needs, the advantageous properties of neuromorphic computing, such as energy efficient computation, make it a promising transformative computational approach for domains such as remote sensing. For more detailed review of neuromorphic computing see [18] [19] [20].

III. Remote Sensing

As an alternative paradigm to computing, rather than executing legacy algorithms employed on conventional architectures, the novel architectures neuromorphic approaches provide require alternative algorithmic formulations to leverage the capabilities of the emerging architectures. In this section we describe the primary classes of sensors and applications that space-based remote sensing employs, providing an understanding of how novel algorithms may be developed to utilize these neuromorphic architectures.

A. Sensors

In space-based sensing, there are three general classes of sensors: non-imaging sensors, framing focal plane arrays, and event-driven focal plane arrays. Non-imaging sensors are used in situations where spectral and/or intensity information is predominantly of interest, and spatial information is less important or can be inferred by means other than a single sensor's measurement. Non-imaging sensors can have very high sampling rates, and therefore excellent temporal resolution, without overwhelming downlink data constraints. Focal plane arrays contain multiple sensing elements or pixels, and hence add spatial information to spectral and intensity information. A frame focal plane array collects data much like a movie camera. In this sense, a snapshot or readout is taken every time interval, creating a 2D array of timestamped pixel measurements. A disadvantage of this design is that the volume of sensor data increases rapidly as the sampling rate increases. One data volume mitigation strategy is to identify localized regions of interest within the larger sensor, and read out only information from this sub-array. This windowing strategy requires additional system complexity to trigger on events of interest and appropriately shape the sub-array.

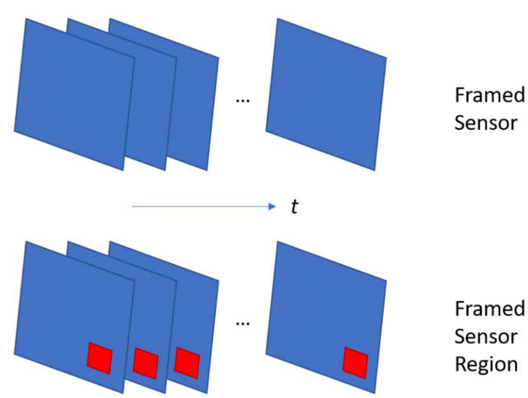

Figure 2: Illustration of the framed sensor paradigm where 2D regions are collected repeatedly over time. As shown in the bottom half of the figure, a region of interest may be focused upon to enable greater transmission throughput of just the region of interest.

Event-driven focal plane arrays are one implementation of this windowing strategy. The logical extreme of an event-driven focal plane array is a spiking sensor, where each pixel simply fires once the environmental energy it absorbs and (leakily) stores exceeds a threshold.

B. Data

The three broad types of space-based sensors provide different data products. Notional examples of this data are illustrated in Figure 3. The left-most plot shows the amplitude versus time data typical of non-imaging sensors, shown for two different spectral bands. This sort of representation is actually a post-processed output, where sensor readouts across time have been accumulated. An instantaneous readout will yield only a single value. As mentioned above, very high temporal resolution is much more achievable with non-imaging sensors given typical communication and readout constraints for space-based systems. This comes at the cost of losing spatial information (from one sensor).

The center image shows a single instant in time of a framing sensor, with varying input levels shown as various shades of grey between pure white and black. Physical sensors must read values off the array, often sequentially, so the 'instant' associated with a given data frame may differ slightly across the array. (In digital photography, this is termed the 'rolling shutter' effect.) Due to readout and communication constraints, space-based full framing sensors typically operate much slower than their non-imaging counterparts. Furthermore, the higher the spatial resolution of the sensor (the more pixels in the array), the more data is generated. This can lead to a trade-off between spatial and temporal resolution of the data. One way to get around this is to only read off areas of interest from the array. Typically, these regions of interest are defined by regions of rapid change. As such, this requires much more sophistication in the on-array processing to isolate and readout the 'interesting' data. However, the benefits of this method allow for much higher temporal resolution without reducing spatial resolution.

The right-hand plot is also an instantaneous representation of an entire focal plane array. However, this array is a spike-based sensing array, meaning that the produced data is event-driven and single bit. Individual pixels can either produce a spike (shown as white) if information needs communicating, or nothing (black). Depending on the design of the spiking sensor, the output of spikes from an individual pixel can either be clocked or asynchronous. If the spikes are output on a clock, then the figure represents one time instant. However, if the sensor has asynchronous spiking capabilities, this image actually represents a small time interval, rather than a true 'instant' as spikes typically occur as soon as a per-pixel threshold is exceeded, rather than on an external polling. Hence spikes are not edge aligned in time, and the activation of multiple spikes as shown in the plot implies we are looking at the array over some (small) time interval. This method can provide readout of the entire array at very high speeds, as data is only generated when necessary, and it is single bit. This is very amenable to space-based readout and communication constraints.

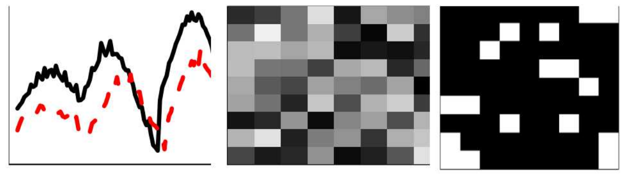

Figure 3: Notional illustration of the form of data from the three primary types of remote sensors: typeNon-imaging (left), Framing (middle), and Spiking Data (right).

With the growing interest in applications of machine learning advances to remote sensing data, various publicly available datasets (some with challenges) are available. For example, SpaceNet is a collection of satellite imagery encompassing several large cities around the world collected by DigitalGlobe satellites. Examples of non-imaging (time-series) data include the Kaggle Exoplanet search [21] and the StarLightCurves dataset [22]. Due to the relatively recent development of event-based focal plane arrays for space-based systems [23], the authors of this paper are not aware of a publicly available dataset for spiking sensors, as of yet.

C. Applications

As is often the case with interdisciplinary efforts, the fields of machine learning, remote sensing, and signal processing have some overlapping terminology to describe the fundamental problems of space-based remote sensing applications which perform computation upon the sensed signal data. Here we provide an overview of the application areas so that neuromorphic research can develop algorithms to address the fundamental problems rather than focusing upon specific canonical algorithmic approaches. And likewise, remote sensing researchers can utilize the signal processing breakthroughs being made in fields such as machine learning (ML) and deep neural networks (DNN) in particular.

Rather than characterizing application areas by the means in which a computation is achieved, instead we describe three broad application areas which jointly capture the problem space of remote sensing tasks and computational techniques ML techniques can address. These are: *signal processing, signal classification, and signal understanding*. We denote each of these classes in terms of signals, to broadly capture that they can encompass a variety of modalities of interest such as the various sensor types described above. Others have provided similar taxonomies with greater granularity focusing upon specific applications such as in [24]. As follows, we will describe each of these classes of application with examples to further articulate each of them.

*1) **Signal Processing**:* By signal processing we are referring to the manipulation or transformation of a sensed signal. An abundance of mathematical transformations exist, and the desired outcomes of the signal processing computation includes manipulating the signal to a representation more amenable to subsequent processing. Examples include noise reduction or dimensionality manipulation. For space-based remote sensing this includes tasks such as the alignment of sensed images or cleaning up signal to noise ratios.

Neural network techniques are increasingly being employed such as for signal denoising or enhancement, sharpening, dimensionality reductions and encoding, etc. In some cases, these signal processing steps are innate to the early processing layers of a DNN. Figure 4 illustrates an example of a neural-inspired denoising approach from [25]. Part a) of the figure shows the original image from the CalTech 101 dataset, and b) is a noisy version of the original image with 10% noise added. Subfigures c) and d) then show the result of two denoising approaches. Namely, in c) a traditional median-filtering technique has been applied, and in d) a spiking neural approach is employed.

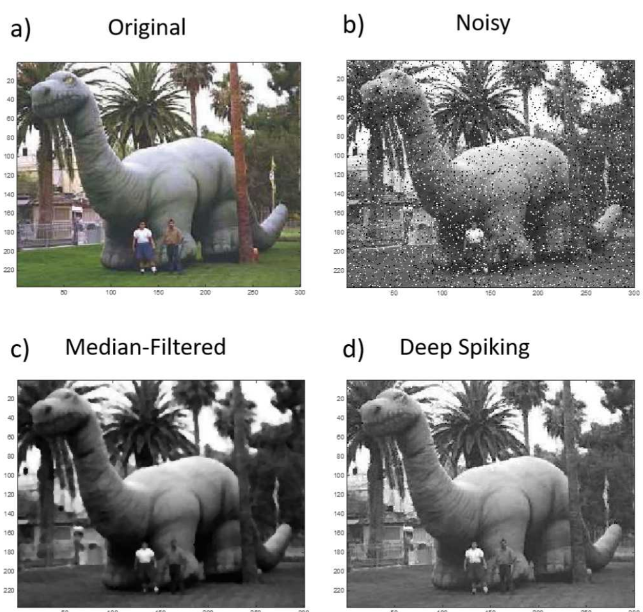

Figure 4: Example illustrating denoising neural network processing

*2) **Signal Classification**:* By signal classification we are referring to the general suite of operations which attribute a label to a signal. Both ML and signal processing have developed an abundance of such techniques. In the scope of space-based remote sensing, signal classification applications are broad but generally consist of the detection of items of interest in a sensed area. This may be object detection, land use determination, single pixel labeling, etc.

Signal classification is one of the most well-known applications of neural networks. Advances in DNNs have increased the accuracy of such techniques, motivating the development of novel algorithm architectures classifying every single pixel, such as semantic segmentation shown in Fig. 5, as well as operating on high dimensional inputs such as hyperspectral. Convolutional neural networks (CNNs) often serve as the core computational approach of signal classification DNNs.

Figure 5: Semantic segmentation examples applied to COCO dataset trained to label the individual pixels of people in the various scenes

*3) **Signal Understanding**:* By signal understanding we are referring to the determination of higher order effects in a sensed signal. Beyond simply determining the presence of a feature in a signal, this includes relationships across space and time. For example, in remote sensing this includes tasks such as tracking items over time or space as well as higher level understanding of a sensed domain based upon the composition or interactions of detected items.

Repeated application of DNN signal classifiers to a series of inputs is one method of observing trends for signal understanding. Additionally, more advanced network structures can intrinsically include recurrence or other features which endow the network with the ability to process sequences of inputs. Figure 6 depicts the output of tracking a golf club head over the course of a swing.

A wealth of emerging research has focused upon applying DNN or deep learning techniques to satellite data across the three general classes of applications described here.

Figure 6: Example of three frames from the HMDB51 dataset in which the red boxes are tracking the golf club head over time across the swing of the golfer

Excitingly, such techniques are seeing compelling results in terms of accuracy but are often computationally expensive and best suited for ground station processing. Alternatively, neuromorphic approaches provide promise for moving such computation to a sensor. An example monitoring processing pipeline is shown in Fig. 7. In this illustration, a weather monitoring system might be observing meteorology trends over time. First pixel data is remotely sensed by a satellite. Then under the category of signal classification, the pixel data is analyzed to identify events which and subsequently classified. This might entail capturing sudden light intensity variations during a lightning storm and observing their signature pattern compared with general background luminescence variability. Comprehending the temporal persistence of illumination variation can then provide signal understanding of a lightning storm. Lastly, rather than transmitting all collected pixel data down to earth, only the reconstructed pixels pertaining to lightning storms can be transmitted as a result of performing processing at the sensor as an enabling capability of neuromorphic processing.

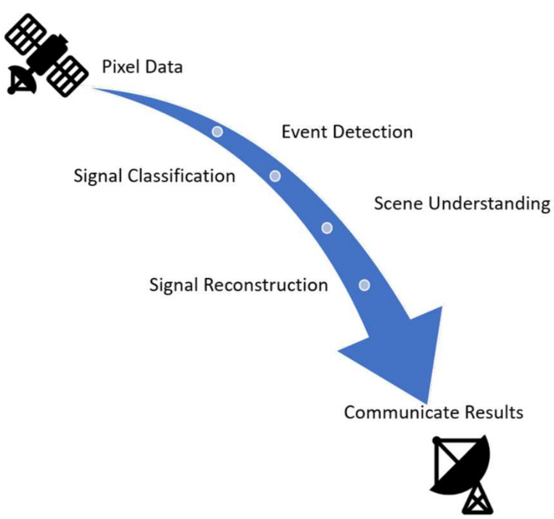

Figure 7: Illustration of a space-based remote sensing pipeline beginning with pixel collection and the ensuing processing steps to communicate a result of the observed phenomenology

For an extensive survey of deep learning approaches and applications to remote sensing see [26].

IV. Results

An example of the potential impact neuromorphic approaches may have for remote sensing is illustrated by Fig. 8 depicting performance in frames per second processed per watt bench-marking across a suite of architectures. In this assessment, the various architectures are computing an image processing task on 28x28 tiles representative of identifying important chips out of a large field of view in a space-based remote sensing task.

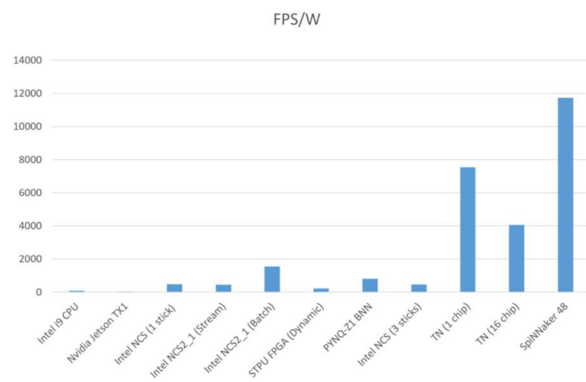

Figure 8: Comparison of various computational architectures in terms of frames per second per watt in a small chip image processing task. Intel NCS denotes the Intel Neural Compute Stick. Parenthetical denotations indicate a single stick as well as using three sticks. NCS2 denotes the second version of the Intel Neural Compute Stick with results for both streaming and batching modes. The STPU FPGA is a Sandia Labs developed Spiking-Temporal Processing Unit architecture implemented on FPGA [27]. The PYNQ-Z1 BNN is a FPGA implementation of FINN [28]. TN denotes the IBM TrueNorth architecture with a single chip as well as chips.

As shown, the far left architectures (barely visible at under a hundred FPS/W) are classic architectures (CPUs and GPUs). Moving to the right, the next few architectures are neuromorphic accelerators as well as FPGA implementations of architectures. Finally, the rightmost benchmarks are two prominent spiking neuromorphic chips achieving several thousand FPS/W. Effectively, as shown, accelerators tailored to ML computational domains are able to improve upon general purpose traditional architectures, but more significantly, spiking neuromorphic approaches are showing on the order of 100x improvements.

As another example of the impact neuromorphic processing may have on remote sensing computation, consider scientific computing. Traditionally, large simulations require a HPC analogous to ground station processing data centers. The immense computational power afforded by these clusters requires significant energy consumption. As an example, Fig. 9 depicts scaling requirements of a particle diffusion simulation. Several thousand nodes are required to perform large scale, high fidelity simulations. Each of these nodes requires on the order of 80W each equating to a very large power budget. Alternatively, Severa et al. have shown that spiking neural algorithms may be employed to compute diffusion equations as a markov random process

[29]. Coupling such an algorithmic approach with neuromorphic hardware, designed to perform such computations efficiently, has the potential to enable comparable large scale computation on hardware consuming a fraction of the power. This shows evidence of how large scale computations traditionally requiring extensive ground station processing can rather be reduced to a deploy able resource constrained operational environment through advances in neural inspired algorithms and architectures.

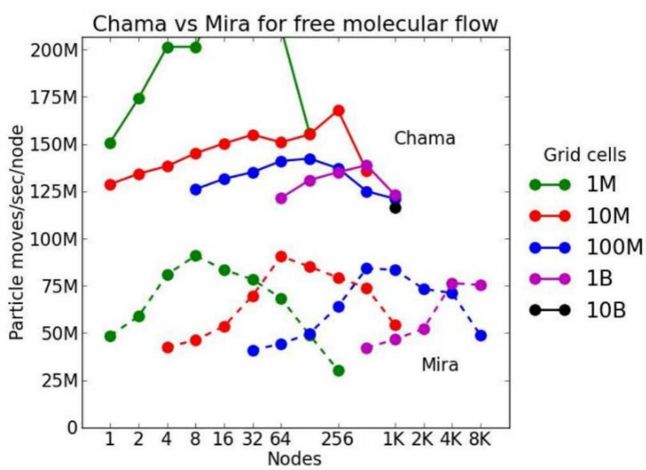

Figure 9: Plot of node scaling across cluster size as a function of particle movement on the SPARTA benchmark. Chama and Mira are two respective HPC architectures. For more information see: http://sparta.sandia.gov/bench.html

V. Conclusion

Meeting a need for alternative, non-conventional computing approaches, neuromorphic techniques are also emerging as a technique enabling computation to be performed under resource limitations in resource constrained conditions such as that of space-based remote sensing. These techniques provide room for growth in both algorithmic and architectural implementations. Several compelling neuromorphic architectures exist using technologies of today, but novel devices and architectures can also be incorporated in future generation neuromorphic architectures. Advances in DNNs are providing state-of-the-art performance across a growing suite of remote sensing applications within signal processing, signal classification, and signal understanding domains. In a co-design manner, these emerging algorithms and architectures can further inspire the development of additional novel approaches. In particular, to maximize the efficacy of such approaches, novel sensor technologies may also be incorporated. Event-driven sensors are already showing promise in enabling novel data representation and transmission methods which neuromorphic approaches are amenable to make use of. The integration of neural-inspired sensors, algorithms, and architectures offers an opportunity for complete system approaches. Following the trends in general purpose computing architectures, the field of neuromorphic computing is well posed to provide advanced processing under size, weight, and power constrains and enable transformative processing capability at the sensor for space-based remote computing.

Acknowledgment

Supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and 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-NA-0003525.

References

[1] G. E. Moore, "Moores law at 40," *Understanding Moores law: four decades of innovation*, pp. 67–84, 2006.

[2] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of ion-implanted mosfet's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, 1974.

[3] J. L. Hennessy and D. A. Patterson, *Computer architecture: a quantitative approach*. Elsevier, 2017.

[4] ——, "A new golden age for computer architecture," *Communications of the ACM*, vol. 62, no. 2, pp. 48–60, 2019.

[5] W. S. McCulloch and W. Pitts, "A logical calculus of hte ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[6] F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[7] M. Minsky and S. Papert, "An introduction to computational geometry," *Cambridge tiass., HIT*, 1969.

[8] K. GGCS, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 9, 1986.

[9] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[10] L. Jackel, R. Howard, H. Graf, B. Straughn, and J. Denker, "Artificial neural networks for computing," *Journal of Vacuum Science & Technology B: Microelectronics Processing and Phenomena*, vol. 4, no. 1, pp. 61–63, 1986.

[11] B. E. Boser, E. Sackinger, J. Bromley, Y. LeCun, and L. D. Jackel, "An analog neural network processor with programmable topology," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 12, pp. 2017–2025, 1991.

[12] C. Mead, "Analog vlsi and neural systems," vol. 90, 1989.

[13] 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," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2454–2467, 2013.

[14] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[15] D. Mike, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S. H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain *et al.*, "Loihi: a neuromorphic manycore processor with on-chip learning," *IEEE Micro*, vol. 38, no. 1, pp. 82–99, 2018.

[16] J. Von Neumann, *The computer and the brain.* Yale University Press, 2012.

[17] Y.-H. Chen, J. Emer, and V. Sze, "Eyeriss v2: A flexible and high-performance accelerator for emerging deep neural networks," *arXiv preprint arXiv:1807:07928*, 2018.

[18] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, "A survey of neuromorphic computing and neural networks in hardware," *arXiv preprint arXiv:1705:06963*, 2017.

[19] S. Furber, "Large-scale neuromorphic computing systems," *Journal of neural engineering*, vol. 13, no. 5, p. 051001, 2016.

[20] C. D. James, J. B. Aimone, N. E. Miner, C. M. Vineyard, F. H. Rothganger, K. D. Carlson, S. A. Mulder, T. J. Draelos, A. Faust, M. J. Marinella *et al.*, "A historical survey of algorithms and hardware architectures for neural-inspired and neuromorphic computing applications," vol. 19, pp. 49–64, 2017.

[21] Kaggle Inc. (2017) Exoplanet hunting in deep space. [Online]. Available: https://www.kaggle.com/keplersmachines/kepler-labelled-time-series-data

[22] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The UCR time series classification archive," October 2018, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[23] G. Cohen, S. Afshar, A. van Schaik, A. Wabnitz, T. Bessell, M. Rutten, and B. Morreale, "Event-based sensing for space situational awareness," 11 2017.

[24] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22–40, 2016.

[25] S. J. Verzi, C. M. Vineyard, and J. B. Aimone, "Neural-inspired anomaly detection," in *International Conference on Complex Systems*. Springer, 2018, pp. 202–209.

[26] J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community," *Journal of Applied Remote Sensing*, vol. 11, no. 4, p. 042609, 2017.

[27] A. J. Hill, J. W. Donaldson, F. H. Rothganger, C. M. Vineyard, D. R. Follett, P. L. Follett, M. R. Smith, S. J. Verzi, W. Severa, F. Wang *et al.*, "A spike-timing neuromorphic architecture," in *2017 IEEE International Conference on Rebooting Computing (ICRC)*. IEEE, 2017, pp. 1–8.

[28] Y. Umuroglu, N. J. Fraser, G. Gambardella, M. Blott, P. Leong, M. Jahre, and K. Vissers, "Finn: A framework for fast, scalable binarized neural network inference," in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '17. ACM, 2017, pp. 65–74.

[29] W. Severa, R. Lehoucq, O. Parekh, and J. B. Aimone, "Spiking neural algorithms for markov process random walk," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.