SAND2019-8543C

Benchmarking Event-Driven Neuromorphic Architectures

Craig M. Vineyard
Sam Green
William M. Severa
cmviney@sandia.gov
Sandia National Laboratories
Albuquerque, New Mexico, USA

Çetin Kaya Koç
cetinkoc@ucsb.edu
University of California Santa Barbara
Santa Barbara, USA

ABSTRACT

Neuromorphic architectures are represented by a broad class of hardware, with artificial neural network (ANN) architectures at one extreme and event-driven spiking architectures at another. Algorithms and applications efficiently processed by one neuromorphic architecture may be unsuitable for another, but it is challenging to compare various neuromorphic architectures among themselves and with traditional computer architectures. In this position paper, we take inspiration from architectural characterizations in scientific computing and motivate the need for neuromorphic architecture comparison techniques, outline relevant performance metrics and analysis tools, and describe cognitive workloads to meaningfully exercise neuromorphic architectures. Additionally, we propose a simulation-based framework for benchmarking a wide range of neuromorphic workloads. While this work is applicable to neuromorphic development in general, we focus on event-driven architectures, as they offer both unique performance characteristics and evaluation challenges.

KEYWORDS

Neural Network Architectures; Neural Network Accelerators; Neuromorphic Computing; Spiking Neural Networks

ACM Reference Format:
Craig M. Vineyard, Sam Green, William M. Severa, and Çetin Kaya Koç. 2019. Benchmarking Event-Driven Neuromorphic Architectures. In *ICONS '19: International Conference on Neuromorphic Systems, July 23–25, 2019, Oak Ridge, TN*. ACM, New York, NY, USA, 5 pages.

1 INTRODUCTION

Neuromorphic computing encompasses algorithms and architectures taking inspiration from the brain to perform computation. Neuromorphic algorithms may be executed on both von Neumann architectures (VA) and non-von Neumann architectures (NVA), or a combination of the two. NVAs have the potential to be more efficient than VAs at brain-inspired computations. This is due to NVAs typically being highly connected and parallel, potentially low-power, and collocating memory and processing. In this work we focus on a specific type of NVA: event-driven architectures, e.g. based on spiking neural networks, which are more biologically plausible than other mainstream NVAs, like digital artificial neural network accelerators [15].

The high performance of artificial neural networks (ANN) for image classification triggered a surge of interest in specialized architectures. For example, totaled across the top architecture conferences[1] only <u>two</u> neuromorphic papers were published in 2014, then <u>64</u> in 2016, and <u>122</u> in 2018 [26]. These numbers specifically represent the growth of ANN architectures.

From an applications perspective, the focus on ANN architectures at most recent VLSI conferences seems justifiable, as the performance of ANNs is now sufficiently good that they are being used in safety-critical applications like autonomous driving, and they are computationally taxing on traditional VA's, motivating a need for alternative neuromorphic approaches. However, ANNs have only loose biological plausibility and they are only good at a narrow range of cognitive tasks. Attention [27] and Capsule Networks [20] are two recent examples which attempt to augment ANNs with greater biological plausibility, and we expect to see more examples in the future.

Given that the number and variety of possible neuromorphic approaches is unbounded, how are architecture design decisions to be made? Rigorous benchmarking has been foundational in advancing traditional computer architecture, however, as NVAs employ alternative paradigms from VAs, it is challenging if not meaningless to try and compare these architectures using solely the same metrics. Different architectural approaches are optimized for different benefits, so appropriate metrics are necessary to provide full understanding of the trade-offs and advantages each affords.

The mainstream ANN community has begun developing strong benchmarking efforts to highlight their advantages[2]. It is the intention of this work to outline benchmarking goals for the neuromorphic community. Our focus is on event-driven architectures, but the guidelines presented here may be applied to neuromorphic architecture evaluation in general. Highlighted by Fig. 1, we propose more extensive architectural evaluation metrics, analogous to how modern nutrition understanding has progressed to include more than just calorie counts. For example, rather than looking at single metrics like operation counts, a more complete understanding of micro-nutrients enables a greater nutritional understanding. Similarly, a more advanced understanding of multiple factors of architecture operation are needed to compare the strengths and weaknesses of computational architectures.

[1] ASPDAC, ASPLOS, DAC, DATE, FPGA, HotChips, HPCA, ICCAD, ISCA, ISSCC, MICRO, VLSI

[2] https://mlperf.org/

ICONS '19, July 23–25, 2019, Oak Ridge, TN

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.

In the remainder of the paper, we expand on why we are focused on benchmarking event-driven architectures, intrinsic and extrinsic metrics, and benchmark candidates for neuromorphic processors.

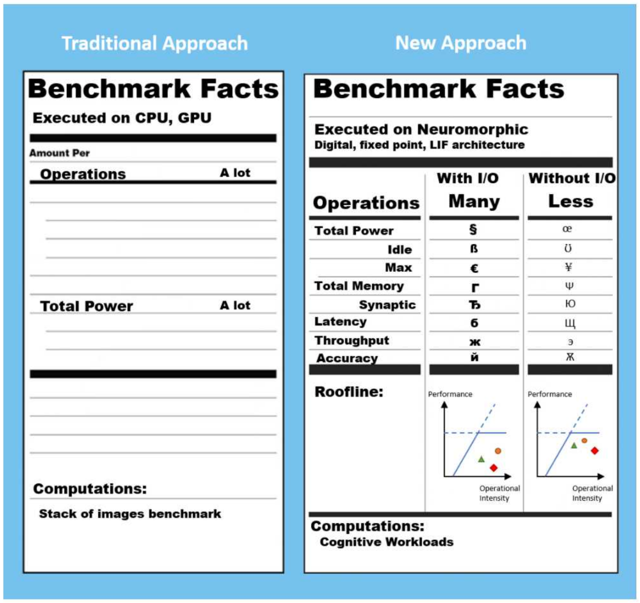

Figure 1: Illustration of traditional and proposed approaches to architectural benchmarking.

2 EVENT-DRIVEN NEUROMORPHIC ARCHITECTURES

Event-driven neuromorphic architectures (EDNA), often modeled on spiking neuron models, are more biologically plausible than ANNs and offer the promise of higher efficiency for certain applications. These architectures are often able to take advantage of sparse connectivity and communication. Industry research platforms and academically available ASIC implementations of event-driven architectures currently include IBM's TrueNorth [3], University of Manchester's SpiNNaker [10], the Human Brain Project's BrainScaleS [21], and Intel's Loihi [8]. There are other ASIC and FPGA implementations, and many architectures that have yet to be physically realized [22].

3 METRICS

Evaluating a neuromorphic processor is nuanced. For example, the literature (or advertising material) for a processor may report "low power", but it may not report benchmarks for a dataset or a task of interest. Furthermore, other published architecture details may lack information required to compare a potential processor to its alternatives. Due to this nuance, we suggest two high-level categories of metrics: extrinsic metrics and intrinsic metrics, with a metric's category dependent on whether or not a workload must be processed to measure the metric. In this work, we provide recommendations for a variety of extrinsic and intrinsic metrics. These metrics may be used to compare and improve architecture designs.

3.1 Intrinsic metrics

Intrinsic metrics may be measured without executing a workload on a processor. These metrics are simple to collect or may be gathered directly from technical manuals or publications, however, they do not provide sufficient information for a researcher to understand workload-dependent performance comparisons. They may not even indicate whether the architecture is likely to meet minimum specifications or performance requirements on tasks of interest.

Intrinsic metrics include *hardware metrics*, e.g. maximum power, idle power, silicon area, process size, clock speed, package dimensions, weight, memory, and time to reconfigure; *architecture metrics*, e.g. connectivity limits, communication limitations, reconfigurability, bit-precision options, IP protection (e.g. encryption), on-device learning availability, and built-in algorithm support; and *metadata metrics*, e.g. maturity, country of origin, access to design files, programming support, and manufacturer.

3.2 Extrinsic metrics

Extrinsic metrics require a specific workload to be executed on an architecture. By "workload", we are referring to the combination of a specific algorithm processing a specific set of data. If the input data changes, this may lead to different processing flows, and therefore to different extrinsic metrics. Extrinsic metrics include power, latency, throughput, accuracy, and roofline analysis.

The workload used to generate extrinsic metrics would ideally be matched to the type of workload for which the architecture was designed. For example, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) benchmark is a popular data set for workloads in the ANN community, but it may be an incongruous workload for most EDNAs [19]. Ideally, the neuromorphic community should have access to a suite of benchmarks which represent different brain-inspired tasks. This would allow researchers to select tasks tailored to their design, and also compare how their design performs relative to other designs on the same workload. We recognize that benchmarking event-driven systems could require hardware or datasets which are not yet widely available.

3.2.1 Roofline analysis. Roofline plots are visual aids to understand performance of a workload on a particular architecture. This tool was developed for analysis of the interaction between system memory, processor performance, and application efficiency in high-performance computing (HPC) [28]. Roofline plots are espoused by a popular computer architecture book and were recently used for analysing systolic and dataflow ANN accelerators [5, 12, 14]. These plots are similarly useful for EDNAs.

Illustrated in Fig. 2, the "roof" of the roofline plot is the maximum theoretical throughput at which a processor can perform some operation. The definition of "operation" is flexible. The HPC community uses floating-point operations (FLOPs). The ANN community may use multiply-accumulate operations (MACs). The neuromorphic community could use events (such as synapticops). Also, higher order operations may be defined, e.g. FLOPs/Watt.

A processor receives data from memory (or directly from a sensor). Memory (or sensor) bandwidth can be saturated if a processor demands too much data, too often. The slope of the roofline indicates memory saturation at various workload intensities, where

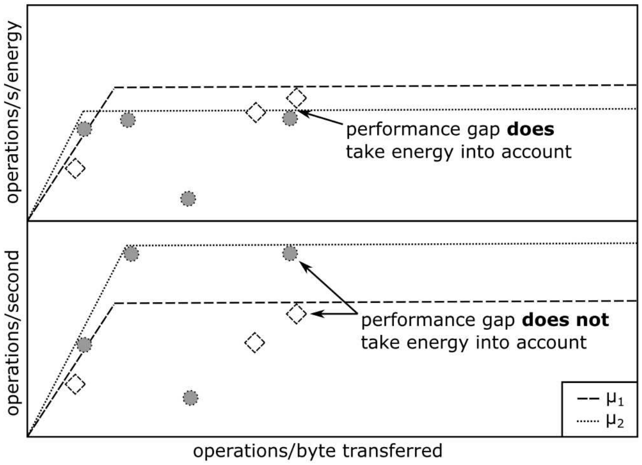

Figure 2: Illustrative roofline plots of two exemplar architectures (μ_1 and μ_2) comparing how various workloads utilize computational resources under two evaluation criteria. The bottom figure does not normalize for energy, but the top figure does. By normalizing for energy we can get a more accurate understanding of efficiency.

"intensity" is from the perspective of a processor. A low intensity workload would finish quickly and require data quickly.

Generating a roofline plot requires a specific workload. Creating a roofline plot for an ANN would require the ANN and all of the training data. After a workload is selected, it can be analyzed: how many operations are required, how many bytes of data for all the code and data, how long does it actually take to run on the hardware. This information is then plotted on the roofline plot.

Roofline analysis can indicate whether a workload is processor or memory-bound or if the workload's implementation is suboptimal for the architecture. In other words, it explains how efficiently a particular workload executes on a particular piece of hardware. A roofline plot may thus be used to compare the efficiency of two architectures for processing equivalent workloads, and it may be used to understand how a system should be optimized to increase performance.

Accordingly, roofline analysis not only illustrates how well workloads are implemented on and suitable for an architecture, but, by specifying the metric(s) of interest constituting an "operation", they can also provide greater insight into the function of the architecture. Amdahl et al., in their foundational paper which first specified the notion of a computer architecture, observed that the utility of an architecture comes from problems solved rather than bits-per-microsecond [4]. This premise is demonstrated by the mathematical optimization of ANNs showing comparable performance can be achieved with computationally simpler models using fewer computations. Next we describe workloads which we propose provide greater insight into the advantages of EDNA processing than singular metrics such as FLOPs often used to measure VAs.

4 COGNITIVE WORKLOADS

In the following subsections we outline high-level cognitive applications we expect to see more brain-inspired neuromorphic systems attempting to solve in the near future. The application areas are drawn from [2] and range from basic sensory processing and pattern recognition to long-term planning at multiple timescales. We model our approach after the Seven Motifs of Scientific Computing [7], which delineates the seven basic kernels of scientific computing: structured grids, unstructured grids, dense linear algebra, sparse linear algebra, fast Fourier transforms, particles, and Monte Carlo. Each of these core algorithms have different hardware and memory access patterns, and they are instantiated in a number of open source benchmark packages. Similarly [18] identifies six kernels ubiquitous in space applications: matrix addition, fast Fourier transforms, matrix multiplication, matrix convolution, Jacobi transformation, and Kronecker product.

Realistic and interesting workloads should be processed in order to exercise a system for generation of extrinsic metrics. To understand this claim, observe that an EDNA processes potentially sparse graphs. Assuming even a deterministic architecture, a single event change in an input may lead to numerous different downstream events. And while it is possible to generate arbitrary inputs and algorithms to stress specific aspects of an architecture, we consider it to be more meaningful if such tests are tied to problems which the community is interested in solving.

Similar to how scientific software is usually composed of multiple kernels discussed above, future cognitive systems will most likely combine two or more of the following.

4.1 Feed-forward sensory processing

Cognitive systems need to perform pattern classification and regression from potentially multimodal sensory inputs. Modes may include vision, audio, tactile, sonar, radar, or other more abstract data like sales transaction information. Basic, but high accuracy, classification of static images began the current trend in ANN popularity and remains a mainstay of machine learning systems [11, 17].

4.2 Recurrent sensory processing

Success at feed-forward sensory processing implies that the current observation contains all the information needed for prediction. However, for systems with temporal dynamics or other time-dependent behavior, some type of memory is needed for accurate prediction. Simple memory is often achieved through network recurrency, where intermediate information is retained locally and processed alongside new information, allowing temporal dependencies to be learned. See [6, 13, 27] for examples.

4.3 Top-down processing

ANNs gradually build up features in a bottom-up approach. For example, filters from early layers in convolutional neural network learn to detect edges, while later layers learn to detect entire shapes. This is not how mammalian brains process sensory data in general. A more biologically plausible approach at feature extraction allows higher-level processing to affect lower-level processing. This top-down approach may be modeled with Bayesian algorithms. For some efforts in this direction, please see [1, 24, 25].

4.4 Dynamical memory and control algorithms

Biological neurons, and groups of neurons, and various regions of the brain can be modeled as multiple dynamical systems. This is something that neither ANNs nor current EDNA architectures commonly do. The neuromorphic architecture community must wait for tractable models to become available before tackling problems in this space. One existing effort along these lines is [9].

4.5 Cognitive inference algorithms, self-organizing algorithms and beyond

The frontal and subcortical parts of the brain are responsible for long-term planning from earlier processed information. Popular reinforcement learning methods represent a simple example of long-term decision making. As progress continues with more capable feed-forward sensory processing, recurrent sensory processing, Bayesian neural algorithms, and dynamical memory and control, we expect to also see progress in their consolidation in the form of powerful long-term planning algorithms. This, as well as using these subsystems for life-long learning across multiple timescales, will represent much of the future effort for the neuromorphic algorithm community. For some interesting concepts on these and other ideas, refer to [2, 16, 23].

5 THE ROLES OF SIMULATION AND EMULATION

In the previous section, we outlined five high-level cognitive application areas. Each of these areas are currently being studied across the neuromorphic computing spectrum. The ANN community has an advantage in the space, as back-propagation performs so well in so many problem areas that these systems have become useful for commercial, scientific, military, and medical applications. The ANN community arrived at this point by applying massive amounts of compute to massive amounts of data. On the other hand, despite having theoretically computational benefit, there are no training algorithms which have yet given event-driven systems the type of workload performance as has been seen in the ANN community.

In order to show progress according to some metric (e.g. power) it must be possible to processes some meaningful workload with high performance. Unfortunately it is expensive, in terms of dollars and time, to generate large amounts of event-driven I/O for sensory data – an area where EDNA architectures should excel at processing. We propose the development of a physics-based simulation system designed to benchmark and compare ANN, EDNA, and other neuromorphic systems. The simulation system would have the following basic characteristics:

- Ability to generate multimodal physics simulations in both the standard spatial domain for ANNs and spatiotemporal domain for EDNAs. For example, both an RGB and DVS camera could be modeled by the simulator. To accurately account for full operation costs, the simulator needs to account of differences such as in bandwidth/transmission rates or power consumption associated with the different paradigms.
- If the neuromorphic hardware has an emulator then it may be used for training directly from the simulator outputs. This approach would allow for massive parallelization for the development of neuromorphic algorithms and the collection of many extrinsic metrics, e.g. spiking events per workload.
- If the neuromorphic hardware is a low-power physical system, a simulator interface board could be developed. The interface board would translate I/O between the simulator and the neuromorphic hardware. The I/O could include both analog and digital channels. Additionally, the interface board could be designed to provide power to the low-power system, thus it would be possible to measure the neuromorphic system's power consumption.
- If the neuromorphic hardware is actually a cluster, e.g. SpiNNaker and BrainScales, then the simulator's interface board would only communicate with the neuromorphic hardware, without measuring power. If the cluster has the ability to be partitioned, then multiple simulators could be connected.

Widely available access to appropriate workloads and computational resources is currently preventing both accurate comparisons between various ANNs and EDNAs, as well as participation from a wider community. Our simulation proposal aims to create a rich, flexible, physics-based environment. Once such an environment is available, then various challenge workloads may be created, e.g. controlling an autonomous vehicle with event-based sensors or performing robotic manipulation with event-based tactile input. Once appropriate workloads are created, algorithms and architectures may be developed and applied using either hardware emulation or domain appropriate I/O. Execution of the attempted solutions will then enable collection of extrinsic metrics, which will enable benchmarking for design improvement and comparison.

6 CONCLUSION

Event-driven neuromorphic architectures offer more promising performance-per-Watt operation than currently-popular ANNs, but it has been difficult to compare various EDNA realizations – both among themselves and to ANNs. In this work, we define various metrics which may be used to evaluate EDNA performance. Taking inspiration from techniques employed by conventional computer architecture, we present how roofline analysis may be utilized in conjunction with other advanced performance measures to develop an understanding of the strengths and weaknesses of different architectures on common workloads. Our approach emphasizes the importance of the algorithms and data being processed for the collection of meaningful and actionable metrics. Additionally, we motivate a need for the development of cognitive workloads, inspired by the motifs of scientific computing. Such cognitive workloads will include datasets to process, but they offer more than a simple data science challenge, as their combination with a task creates a computational requirement which stresses the architectures and articulates their merits, rather than simply providing a one-dimensional metric like floating-point operation counts. To address these needs, we also propose a simulation framework that may be used to efficiently train and evaluate EDNAs on cognitive tasks.

7 ACKNOWLEDGMENTS

The authors thank James Aimone for critical comments and discussions regarding the manuscript. This work was supported by the DOE Advanced Simulation and Computing program, and the Laboratory Directed Research and Development program at Sandia National Laboratories. Sandia National Laboratories is a multiprogram 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. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

REFERENCES

[1] Subutai Ahmad and Jeff Hawkins. 2015. Properties of sparse distributed representations and their application to hierarchical temporal memory. *arXiv preprint arXiv:1503.07469* (2015).

[2] James B. Aimone. 2019. Neural Algorithms and Computing Beyond Moore's Law. *Commun. ACM* 62, 4 (April 2019), 110.

[3] Filipp Akopyan, Jun Sawada, Andrew Cassidy, Rodrigo Alvarez-Icaza, John Arthur, Paul Merolla, Nabil Imam, Yutaka Nakamura, Pallab Datta, Gi-Joon Nam, et al. 2015. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 34, 10 (2015), 1537–1557.

[4] Gene M. Amdahl, Gerrit A. Blaauw, and FP Brooks. 1964. Architecture of the IBM System/360. *IBM Journal of Research and Development* 8, 2 (1964), 87–101.

[5] Yu-Hsin Chen, Joel Emer, and Vivienne Sze. 2018. Eyeriss v2: A Flexible and High-Performance Accelerator for Emerging Deep Neural Networks. arXiv:cs.DC/1807.07928

[6] Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2015. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*. 2067–2075.

[7] Phil Colella. 2004. Defining Software Requirements for Scientific Computing. (01 2004).

[8] Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. 2018. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 1 (2018), 82–99.

[9] Chris Eliasmith, Terrence C Stewart, Xuan Choo, Trevor Bekolay, Travis DeWolf, Yichuan Tang, and Daniel Rasmussen. 2012. A large-scale model of the functioning brain. *science* 338, 6111 (2012), 1202–1205.

[10] Steve B. Furber, Francesco Galluppi, Steve Temple, and Luis A. Plana. 2014. The spinnaker project. *Proc. IEEE* 102, 5 (2014), 652–665.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[12] John L. Hennessy and David A. Patterson. 2017. *Computer architecture: a quantitative approach*. Morgan Kaufmann.

[13] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[14] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, et al. 2017. In-datacenter performance analysis of a tensor processing unit. In *2017 ACM/IEEE 44th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 1–12.

[15] Jeffrey L. Krichmar, William Severa, Muhammad S. Khan, and James L. Olds. 2019. Making BREAD: Biomimetic Strategies for Artificial Intelligence Now and in the Future. *Frontiers in Neuroscience* 13 (2019), 666.

[16] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. 2017. Building machines that learn and think like people. *Behavioral and brain sciences* 40 (2017).

[17] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[18] Tyler M. Lovelly and Alan D. George. 2017. Comparative analysis of present and future space-grade processors with device metrics. *Journal of Aerospace Information Systems* 14, 3 (2017), 184–197.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, and et al. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115, 3 (Apr 2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[20] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic Routing Between Capsules. *CoRR* abs/1710.09829 (2017). arXiv:1710.09829 http://arxiv.org/abs/1710.09829

[21] Sebastian Schmitt, Johann Klähn, Guillaume Bellec, Andreas Grübl, Maurice Guettler, Andreas Hartel, Stephan Hartmann, Dan Husmann, Kai Husmann, Sebastian Jeltsch, et al. 2017. Neuromorphic hardware in the loop: Training a deep spiking network on the brainscales wafer-scale system. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2227–2234.

[22] Catherine D. Schuman, Thomas E. Potok, Robert M. Patton, J. Douglas Birdwell, Mark E. Dean, Garrett S. Rose, and James S. Plank. 2017. A Survey of Neuromorphic Computing and Neural Networks in Hardware. *CoRR* abs/1705.06963 (2017). arXiv:1705.06963 http://arxiv.org/abs/1705.06963

[23] P Taylor, JN Hobbs, J Burroni, and HT Siegelmann. 2015. The global landscape of cognition: hierarchical aggregation as an organizational principle of human cortical networks and functions. *Scientific reports* 5 (2015), 18112.

[24] Joshua B Tenenbaum and Fei Xu. 2000. Word learning as Bayesian inference. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, Vol. 22.

[25] Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. 2019. Learning to Infer and Execute 3D Shape Programs. *arXiv preprint arXiv:1901.02875* (2019).

[26] Fengbin Tu. 2019. Neural Networks on Silicon. https://github.com/fengbintu/Neural-Networks-on-Silicon

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017). arXiv:1706.03762 http://arxiv.org/abs/1706.03762

[28] Samuel Williams, Andrew Waterman, and David Patterson. 2009. *Roofline: An insightful visual performance model for floating-point programs and multicore architectures.* Technical Report. Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States).