Energy and Area Efficiency in Neuromorphic Computing for Resource Constrained Devices

Gangotree Chakma, Nicholas D. Skuda, Catherine D. Schuman, James S. Plank, Mark E. Dean and Garrett S. Rose

ACM Great Lakes Symposium on VLSI (GLSVLSI), Chicago, IL, USA, May 2018.

DOI: https://doi.org/10.1145/3194554.3194611

Energy and Area Efficiency in Neuromorphic Computing for Resource Constrained Devices

Gangotree Chakma,
Nicholas D. Skuda
University of Tennessee
Knoxville, TN
[gchakma,nskuda]@utk.edu

Catherine D. Schuman
Oak Ridge National Laboratory
Oak Ridge, TN
schumancd@ornl.gov

James S. Plank, Mark E. Dean,
and Garrett S. Rose
University of Tennessee
Knoxville, TN
[jplank,markdean,garose]@utk.edu

ABSTRACT

Resource constrained devices are the building blocks of the internet of things (IoT) era. Since the idea behind IoT is to develop an interconnected environment where the devices are tiny enough to operate with limited resources, several control systems have been built to maintain low energy and area consumption while operating as IoT edge devices. Several researchers have begun work on implementing control systems built from resource constrained devices using machine learning. However, there are many ways such devices can achieve lower power consumption and area utilization while maximizing application efficiency. Spiky neuromorphic computing (SNC) is an emerging paradigm that can be leveraged in resource constrained devices for several emerging applications. While delivering the benefits of machine learning, SNC also helps minimize power consumption. For example, low energy memory devices (memristors) are often used to achieve low power operation and also help in reducing system area. In total, we anticipate SNC will provide computational efficiency approaching that of deep learning while using low power, resource constrained devices.

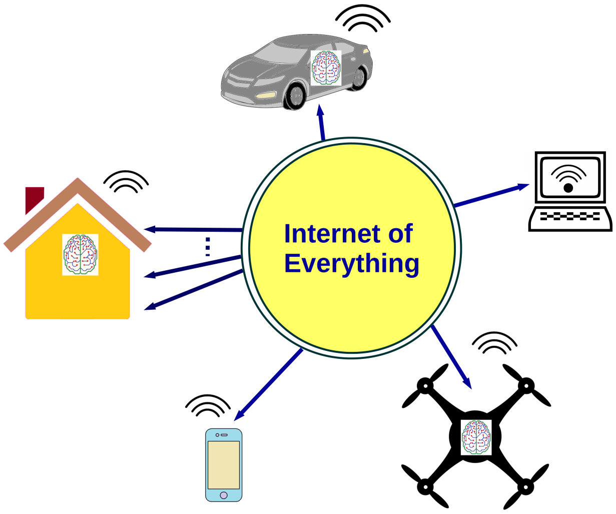

Figure 1: Neuromorphic computing in IoT edge devices.

1 INTRODUCTION

Internet of things applications (IoT) (Fig. 1) are emerging in which many resource constrained devices that compose such systems can benefit from machine learning capabilities. For example, autonomous drone applications can include simple neural networks which actively learn and adapt to subtle variations that will occur as they navigate their environment. Such capability can be particularly advantageous for applications where the drones are used to search for survivors in a disaster zone (e.g., a collapsed building), to provide one example. While such capability would certainly be advantageous, a real challenge exists in providing such on-the-fly learning in systems with limited area and power supply.

Several recent works have explored how deep learning neural networks can be leveraged for resource constrained systems. For example, Leroux *et al.* specifically demonstrated how cascaded neural network layers can achieve small error rates when classifying MNIST characters [9] using IoT devices [11]. Similarly, Motamedi *et al.* demonstrate a system-on-chip (SoC) implementation for convolutional neural networks (CNN) that is particularly useful at reducing the number of threads, and thus power, required for deep learning applications [15]. The prior work in this area has made some important steps toward constructing more efficient deep learning systems for IoT. However, it is the premise of this work that the extreme resource constrained nature IoT demands a more careful study of neuromorphic approaches that are inherently more area and energy efficient than any deep learning approach. Thus, we focus our attention on sparse recurrent neural networks (RNN) that exhibit significant area and energy savings relative to their CNN counterparts.

In [18], Schuman and Birdwell present a genetic algorithm based technique, evolutionary optimization (EO), tailored for implementing area-efficient spiky neuromorphic networks. Typically, EO produces sparse RNN neural networks (small, energy-efficient) that run on the corresponding neuro-inspired dynamic architecture (NIDA) [17]. Further, simplicity in the number and type of parameters is a key feature of the NIDA architecture, making is particularly area and energy efficient relative to other neuromorphic approaches. Thus, NIDA-style systems are particularly well suited for resource constrained IoT systems. While NIDA is itself a high-level architecture (typically only simulated), our group has also begun development on a NIDA-based memristive dynamic adaptive neural

Notice: This manuscript has been authored in part by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

network architecture (mrDANNA) [4] that can one day be leveraged in IoT hardware implementations.

Normally for analysis of large or streaming datasets like those from the sensors within IoT devices, data analysis is performed using deep neural networks (DNN's). Traditionally, to avoid needing massive compute power at edge devices, the data is compressed, encrypted, and then broadcast to a data center to handle the calculation, and the response is then sent back to the device. Performing the connection remotely uses much less power than doing the calculation at the endpoint, but this can be slow, insecure, and unreliable in areas without a strong network infrastructure [19]. To avoid these problems, some devices include more powerful processors and more efficient deep learning algorithms to either do all of the deep learning work on-device or process some layers of the DNN on device before sending a smaller message to the remote compute facility [12, 19]. Adding more traditional computing power to a given IoT device may not be desirable or even possible, so work on small, low-power devices is needed.

2 BACKGROUND

2.1 Nano-scale Memristors

Leon O. Chua first introduced the theory behind "memristors" (or "memory resistors") [6] in 1971. A memristor in more recent terms typically refers to a two terminal nanoscale non-volatile device whose resistance can be modulated by the magnitude of voltage across the device and the time for which the voltage is applied. A memristor can attain multiple resistance levels between the two bounds known as the low resistance state (LRS) and the high resistance state (HRS). The LRS and HRS of any memristor is dependent on the switching material, process conditions, noise and environmental conditions. Since memristors are very small in size, they are particularly attractive for designing area efficient systems. Moreover, the energy consumption of a memristor tends to be very low as well, especially for devices whose LRS is relatively large. Hence, memristors have been attractive to the energy efficient neuromorphic chip designer. The used-based switching characteristics of memristors also make them useful for implementing synaptic circuits. Several neuromorphic architectures have been proposed in recent years which leverage a variety of memristive switching materials [10, 13, 20].

The memristor model we have used in this paper is based on [1]. This model was developed from on experimental results for nanoscale hafnium-oxide (HfO_x) memristors fabricated within a 65 nm CMOS process [2, 3]. Fig. 2 shows the current-voltage relationship for an example HfO_x memristor, specifically illustrating the characteristics of the device when switching between the LRS and HRS resistance states. When a voltage greater than a threshold of $1V$ is applied across the memristor, the device will switch from HRS to LRS. Likewise, an applied voltage less than the negative threshold of about $-0.7V$ will cause the device to switch from LRS to HRS. In both cases, the voltage applied should also be held for at least a minimum "switching time" (typically $10 - 100ns$) in order to switch fully between the two extreme resistance states, HRS and LRS. Resistance states between LRS and HRS are also achievable by applying short, nanosecond pulses that essentially "nudge" the resistance. This variety and range of resistance is particularly useful for representing synaptic weights in the spiky neuromorphic model considered. The ability to change these resistance values with controlled pulses is also leveraged to implement online learning mechanisms such as spike time-dependent plasticity (STDP) [4].

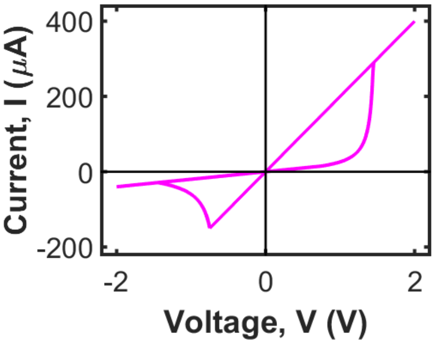

Figure 2: I-V characteristics of memristor model.

2.2 Neuromorphic ASIC Design

Resource constrained IoT devices are always in need of devices and techniques that consume less power and energy, as these are the major constraints for compact systems with limited battery supply. Hence, memristor-based spiky neuromorphic computing architectures are particularly attractive as viable solutions for IoT based machine learning. Moreover, several emergent nano-scale devices (memristors included) are being leveraged in such systems that promise lower area and power consumption [7, 8, 16]. For our part, non-volatile memristors have been used to design synapses that help to ensure low energy consumption while storing synaptic weights. Different materials and their corresponding devices will exhibit differences in energy consumption. For instance, if we consider HfO_x, the energy while the synapse is active, idle and learning is detailed on Table 1. Here, the active phase of the synapse refers to the mode of operation; the idle phase defines the inactive condition and the learning phase includes both the potentiation and the depression of the synaptic weights based on the neuron fires. Each of these synapse states consumes energy depending on the type of memristive device used and the peripheral control circuitry.

Table 1: Energy values for HfO_x synapses [4]

Synapse state	Energy per spike (pJ)
Active	0.48
Idle	0.002
Learning (IncreaseWeight)	0.26
Learning (DecreaseWeight)	0.13

Mixed-signal CMOS neurons can also be designed for energy and area efficiency, specifically when using CMOS integrate-and-fire neurons consisting of very few transistors [5]. The primary concern in terms of area is the size of the capacitors. Capacitor area can also be mitigated via the use of different memristors that will in turn ensure proper incoming current flow into the neuron. The mixed-signal neuron operates in three different phases: *accumulation*, *idle* and the *firing* phases. Energy consumed by the neuron during these phases are listed on Table 2. Here the accumulation energy refers to the energy consumed while accumulating incoming charge/spikes from the synaptic weights. The idle energy is comparatively low since the neuron's functionality is mostly inactive with the exception of peripheral circuitry. Unlike the synapse phases, the neuron does not consume energy specifically relating to the learning process. Instead, the neuron consumes energy while it produces firing spikes. This energy involves the generation of a post-neuron fire when the accumulated charge crosses over the threshold of that particular neuron. Moreover, the shaping of the firing spike is also an important factor for energy consumption since the generated firing pulses would be fed into the next stage of neuromorphic cores.

Table 2: Energy values for CMOS neurons [4]

Neuron phase	Energy per spike (pJ)
Accumulation	9.81
Idle	7.2
Firing	12.5

The neuromorphic architecture (mrDANNA) we consider is mixed-signal in nature. Hence the architecture is significant for low power and area efficiency and includes both the synapse and neuron model described. Several synapses and neurons are gathered to turn into some memristive neuromorphic core which we call mrDANNA cores. Each core contains energy efficient memristive synapses and an analog IAF neuron. The total layout of the design (shown in Figure 3) consists of 36 placements of mrDANNA cores on the right side of the design with the left side containing a digital implementation of the architecture. The advantage of a mixed-signal design is that the connections with the outside signal are fully digital whereas the integration within the core itself is analog. Hence, the mixed-signal memristive neuromorphic system discussed here is digital between cores and analog within the core. Moreover, the mixed-signal approach is also more energy efficient relative to other digital implementations. Since mixed-signal models provide the opportunity to implement synapse and neuron models with better area and power efficiency, we are inclined towards improving the design of our neuromorphic cores for use in multiple different applications including classification, control and anomaly detection.

3 EXAMPLE: SMALL ROBOT NAVIGATION

One example of a resource constrained environment for computation is the navigation system on an autonomous robot. A robot usually has fixed energy storage and has limits on size and weight to allow it to travel. In this paper we consider the NeoN robot design [14] using a mrDANNA hardware chip. As described by Mitchell et. al, the output spikes from this network are used to signal the motor controller, while input spikes are mostly generated by the robot's sensors (in this case, LIDAR sensors on a servo, which take five measurements in an arc, and limit switches). The goal of the autonomous robot is to explore as much of an area as possible while avoiding any obstacles. In particular, the robot must be able to navigate unfamiliar environments, where the layout of the environment and the obstacles within the environment is unknown.

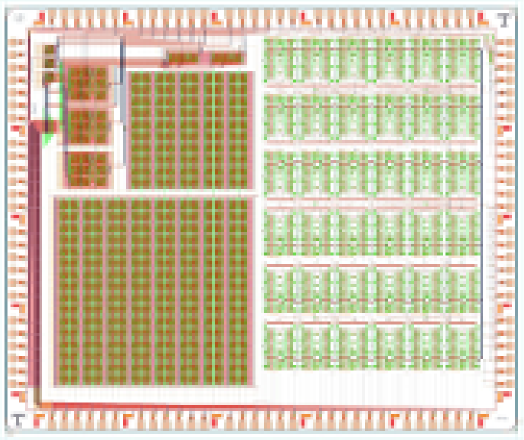

Figure 3: Final layout of first generation mrDANNA test chip submitted for fabrication [3, 4].

Networks are developed and trained using evolutionary optimization (EO); in particular, a set of fixed room configurations are used to evaluate how well the network is able to control the robot to perform the objective of covering as much ground as possible while also avoiding obstacles. During the EO, we simulate the neuromorphic system, the robot, and the environments in which the robot navigates (see Figure 4 for a visualization of the simulation), rather than using the actual robot in real environments in training. Though the simulator is relatively primitive, we have deployed network trained in simulation using EO onto another neuromorphic architecture (DANNA), which has then been used to successfully control the real robot in unfamiliar environments [14]. As such, we are confident that the resulting mrDANNA network will also perform well on the robot itself in navigating new environments.

The example network generated using the EO to operate the robot uses 31 neurons (9 input neurons, 4 output neurons, and 18 hidden neurons) and utilizes 119 of the synapses for communication (see Figure 5), roughly twice as many as the DANNA implementation described by Parker et. al [14]. The nine input neurons are where the LIDAR sensor information is fed in (five inputs), along with the robot's limit switches information (two inputs), and a bias and random value to help drive activity. The four outputs correspond to forward and backward for both the left and right motors. The network shown in Figure 5 represents a single, sparse neuron layer to handle processing the input and decision-making, and includes many recurrent connections. This different style of handling network layers (as opposed to traditional feed-forward neural networks) makes this network type much smaller than the networks used in traditional deep learning methods.

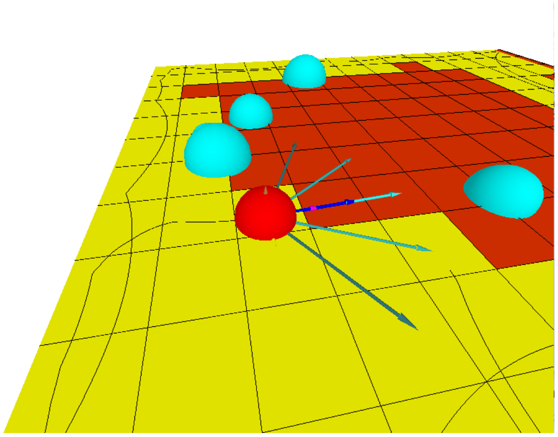

Figure 4: This is the simulator visualization for robot navigation, the floor is divided into a grid – red boxes are unvisited, yellow boxes have been visited. The red sphere is a simple representation of the robot, and the five blue rays represent its sensors. The teal spheres represent obstacles that must be avoided. The black path on the floor is the path that the robot has taken.

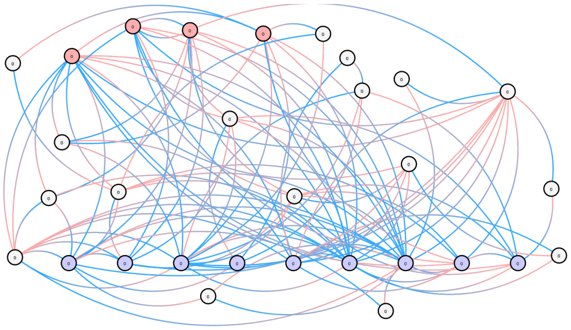

Figure 5: A visualization of the example network. Neurons are represented by colored circles: Blue are input neurons, red are output neurons, and white are hidden neurons. The arcs represent the synapses of the network with the blue end being the pre-neuron and the pink end being the post-neuron.

4 RESULTS

In order to understand how this network would perform on physical mrDANNA hardware, we analyzed the network's performance and measured various types of activity in the network. We use the measurements shown in Tables 1 and 2 to then estimate the power consumption of the network on a physical chip. Analyzing the network's performance showed an average of 4425 spikes in the network per second. When considered with the 20MHz clock frequency, this results in the network being idle the vast majority of the time during this real-time task, as the robot only polls to make a decision five times per second. The average power used by the network is approximately 142.7 μW (see Table 3). It is worth noting that this value is only measuring the core logic of the neuromorphic system; neither the costs of generating spikes from the input sensors nor converting output spikes to signals to the motor controller are factored into this value, both of which may be non-trivial.

Table 3: A description of a NeoN mrDANNA network

Number of Neurons	31
Number of Synapses	119
Average Spikes per Second	4425
Power Usage (Core Logic)	142.7 μW

The generated network is generally very sparse but there are some heavily connected nodes, with individual neurons having out-degrees up to 13 and in-degrees up to 11. The connectivity is primarily directly between input and outputs neurons (including many input-to-input connections), with the hidden neurons usually having relatively few synapses.

5 FUTURE WORK AND CONCLUSIONS

The era of IoT devices has compelled us to consider such technologies with opportunities that introduce resource constrained devices having area and power efficiency. Since we are looking at neuromorphic computing with emergent devices, our primary goal is to ensure energy and area efficiency with proper architecture. In this work, we have shown that a network for a neuromorphic architecture using emerging devices (mrDANNA) can be generated using evolutionary optimization to perform an autonomous robot navigation task. The resulting network is relatively small and sparse when compared with most deep learning networks, resulting in area efficiency, while also maintaining power efficiency (142.7 μW).

There is much future work to be done with respect to neuromorphic computing and IoT devices. One potential future direction is the aggregation of data communication among devices connected to each other. Because of the potential power of spiky neuromorphic systems in analyzing streaming data, we also anticipate that neuromorphic systems will play a large role in data analysis at the "edge." Our utilization of EO for training has tended to produce smaller, sparser networks than what is typically seen in deep learning networks, making them well-suited for deployment onto physical, resource-constrained devices. Moreover, we also intend to explore utilizing on-chip plasticity mechanisms to continue learning or training on the device itself. These types of mechanisms will potentially allow for adapting and self-healing systems in the future, which will be especially important for IoT devices that are in remote areas and not easily accessible. We envision to continue our research in the area of big data analysis while ensuring low energy and area consumption for largely connected IoT devices.

6 ACKNOWLEDGEMENTS

The authors would like to thank Grant Bruer, Sagarvarma Sayyaparaju and Sherif Amer from the University of Tennessee, Knoxville for interesting and useful discussions on this topic.

This material is based on research sponsored by Air Force Research Laboratory under agreement number FA8750-16-1-0065 and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under contract number DE-AC05-00OR22725. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

REFERENCES

[1] Sherif Amer, Sagarvarma Sayyaparaju, Garrett S. Rose, Karsten Beckmann, and Nathaniel C. Cady. [n. d.]. A Practical Hafnium-Oxide Memristor Model Suitable for Circuit Design and Simulation. In *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*.

[2] Karsten Beckmann, Josh Holt, Harika Manem, Joseph Van Nostrand, and Nathaniel C Cady. 2016. Nanoscale Hafnium Oxide RRAM Devices Exhibit Pulse Dependent Behavior and Multi-level Resistance Capability. *MRS Advances* (2016), 1–6.

[3] Nathaniel Cady, Karsten Beckmann, Wilkie Olin-Ammentorp, Gangotree Chakma, Sherif Amer, Ryan Weiss, Sagarvarma Sayyaparaju, Md. Musabbir Adnan, John Murray, Mark Dean, James Plank, Garrett S. Rose, and Joseph E. Van Nostrand. 2018. Full CMOS-Memristor Implementation of a Dynamic Neuromorphic Architecture. In *Proceedings of the Government Microcircuit Applications and Critical Technology Conference (GOMACTech)*.

[4] Gangotree Chakma, Md. Musabbir Adnan, Austin R. Wyer, Ryan Weiss, Catherine D. Schuman, and Garrett S. Rose. 2017. Memristive Mixed-Signal Neuromorphic Systems: Energy-Efficient Learning at the Circuit-Level. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems (JETCAS)* PP, 99 (2017). https://doi.org/10.1109/JETCAS.2017.2777181

[5] G. Chakma, S. Sayyaparaju, R. Weiss, and G. S. Rose. 2017. A Mixed-Signal Approach to Memristive Neuromorphic System Design. In *60th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. Boston, MA.

[6] L. O. Chua. 1971. Memristor-The missing circuit element. *IEEE Transactions on Circuit Theory* 18, 5 (September 1971), 507–519.

[7] Miao Hu, Hai Li, Yiran Chen, Qing Wu, Garrett S. Rose, and Richard W. Linderman. 2014. Memristor Crossbar-Based Neuromorphic Computing System: A Case Study. *IEEE Trans. Neural Netw. Learning Syst.* 25, 10 (October 2014), 1864–1878. https://doi.org/10.1109/TNNLS.2013.2296777

[8] Giacomo Indiveri, Bernabé Linares-Barranco, Robert Legenstein, George Deligeorgis, and Themistoklis Prodromakis. 2013. Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology* 24, 38 (2013), 384010.

[9] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (November 1998), 2278–2324.

[10] HY Lee, YS Chen, PS Chen, TY Wu, F Chen, CC Wang, PJ Tzeng, M-J Tsai, and C Lien. 2010. Low-power and nanosecond switching in robust hafnium oxide resistive memory with a thin Ti cap. *IEEE Electron Device Letters* 31, 1 (2010), 44–46.

[11] Sam Leroux, Steven Bohez, Tim Verbelen, Bert Vankeirsbilck, Pieter Simoens, and Bart Dhoedt. 2015. Resource-constrained classification using a cascade of neural network layers. In *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 1–7.

[12] H. Li, K. Ota, and M. Dong. 2018. Learning IoT in Edge: Deep Learning for the Internet of Things with Edge Computing. *IEEE Network* 32, 1 (Jan 2018), 96–101. https://doi.org/10.1109/MNET.2018.1700202

[13] Gilberto Medeiros-Ribeiro, Frederick Perner, Richard Carter, Hisham Abdalla, Matthew D Pickett, and R Stanley Williams. 2011. Lognormal switching times for titanium dioxide bipolar memristors: origin and resolution. *Nanotechnology* 22, 9 (2011), 095702.

[14] J. P. Mitchell, G. Bruer, M. E. Dean, J. S. Plank, G. S. Rose, and C. D. Schuman. 2017. NeoN: Neuromorphic control for autonomous robotic navigation. In *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*. 136–142. https://doi.org/10.1109/IRIS.2017.8250111

[15] Mohammad Motamedi, Daniel Fong, and Soheil Ghiasi. 2017. Machine Intelligence on Resource-Constrained IoT Devices: The Case of Thread Granularity Optimization for CNN Inference. *ACM Transactions on Embedded Computing Systems (TECS)* 16, 5s (2017), 151.

[16] Mirko Prezioso, Farnood Merrikh-Bayat, BD Hoskins, GC Adam, Konstantin K Likharev, and Dmitri B Strukov. 2015. Training and operation of an integrated neuromorphic network based on metal-oxide memristors. *Nature* 521, 7550 (2015), 61.

[17] C.D. Schuman, J.D. Birdwell, and M. Dean. 2014. Neuroscience-inspired dynamic architectures. In *Biomedical Science and Engineering Center Conference (BSEC), 2014 Annual Oak Ridge National Laboratory*. 1–4. https://doi.org/10.1109/BSEC.2014.6867735

[18] Catherine D. Schuman and J. Douglas Birdwell. 2013. Dynamic Artificial Neural Networks with Affective Systems. *PLoS ONE* 8, 11 (November 2013), e80455. https://doi.org/10.1371/journal.pone.0080455

[19] J. Tang, D. Sun, S. Liu, and J. L. Gaudiot. 2017. Enabling Deep Learning on IoT Devices. *Computer* 50, 10 (2017), 92–96. https://doi.org/10.1109/MC.2017.3641648

[20] J Joshua Yang, MX Zhang, John Paul Strachan, Feng Miao, Matthew D Pickett, Ronald D Kelley, G Medeiros-Ribeiro, and R Stanley Williams. 2010. High switching endurance in TaOx memristive devices. *Applied Physics Letters* 97, 23 (2010), 232102.