Designing an Analog Crossbar based Neuromorphic Accelerator

Sapan Agarwal[1], Alexander Hsia[2], Robin Jacobs-Gedrim[2], David R. Hughart[2], Steven J. Plimpton[2]
Conrad D. James[2], Matthew J. Marinella[2]
[1]Sandia National Laboratories, Livermore, CA, USA, sagarwa@sandia.gov
[2]Sandia National Laboratories, Albuquerque, NM, USA

Resistive memory crossbars can dramatically reduce the energy required to perform computations in neural algorithms by three orders of magnitude when compared to an optimized digital ASIC [1]. For data intensive applications, the computational energy is dominated by moving data between the processor, SRAM, and DRAM. Analog crossbars overcome this by allowing data to be processed directly at each memory element. Analog crossbars accelerate three key operations that are the bulk of the computation in a neural network as illustrated in Fig 1: vector matrix multiplies (VMM), matrix vector multiplies (MVM), and outer product rank 1 updates (OPU)[2]. For an NxN crossbar the energy for each operation scales as the number of memory elements $O(N^2)$ [2]. This is because the crossbar performs its entire computation in one step, charging all the capacitances only once. Thus the CV^2 energy of the array scales as array size. This fundamentally better than trying to read or write a digital memory. Each row of any NxN digital memory must be accessed one at a time, resulting in N columns of length O(N) being charged N times, requiring $O(N^3)$ energy to read a digital memory. Thus an analog crossbar has a fundamental O(N) energy scaling advantage over a digital system. Furthermore, if the read operation is done at low voltage and is therefore noise limited, the read energy can even be independent of the crossbar size, O(1) [2].

Many different algorithms can be built on these kernels (VMM, MVM, OPU) including backpropagation [3], sparse coding [2], liquid state machines, restricted Boltzmann machines and more. The key design considerations for a neural algorithm accelerator is that it should both reduce the computational energy and delay by orders of magnitude, and it should be flexible enough that it can run many different neural algorithms. The difference in the implementation of many algorithms is how the inputs and outputs of a crossbar are processed. An NxN crossbar accelerates $O(N^2)$ operations, while it has O(N) inputs or outputs. This means that the energy to process an input or output can cost O(N) times more than the energy to read or write a single resistive memory element without significantly increasing the system energy. This key insight allows us to optimize the tradeoff between energy efficiency and system flexibility. A crossbar based neural core should be used to perform the parallel vector matrix multiply and outer product update, while a more general purpose digital core can be used to process the inputs and outputs of the crossbar as illustrated in Fig 2.

In order to interface between analog and digital logic, analog to digital (ADC) and digital to analog (DAC) converters will be needed at the inputs and outputs of the Crossbar as shown in Fig 2(b). As the energy and delay of these converters increases exponentially with the number of bits, it is important to understand exactly what impact the bit precision has on the energy, delay and accuracy of an algorithm. For instance, in [1] an analog accelerator core is designed with all the required ADCs and DACs for 8 bit, 4 bit and 2 bit precision. It is then compared against an optimized digital accelerator with a multiplier placed directly next to a digital cache. The results of the energy, latency and area analysis is plotted in Fig 3. At 8 bits, the analog accelerator has a 430X gain in energy and a 35X gain in latency. Reducing the ADC and DAC bit precision from 8 bits to 4 bits can give an additional 10X gain in energy and latency. In order to enable these energy gains, a high resistance (~100 MΩ) ReRAM cell needs to be developed. Fig 4(a) shows that reducing from 8 bits of precision to 4 bit of precision reduces the accuracy from 98% to 94% when training a 784×300×10 neural network on MNIST[4]. Probabilistically rounding quantized numbers to the nearest value [5] significantly increases the accuracy for the 4 bit systems to 96.6%.

Any analog device that behaves like a programmable resistor can be used to build the crossbar. For instance, resistive memories will change resistance when a large write voltage is applied, allowing the resistance to be programmed. At lower voltages, the state does not change and the device can be read out. The resistance acts like a weight that modulates the voltage applied to it.

Unfortunately, analog devices are noisy and suffer from many non-idealities including read noise, write noise and write nonlinearity. Analog arrays suffer from parasitic voltage drops. Furthermore, analog systems tend to have limited bit precision on the inputs and outputs to a crossbar, with the fewer bits used, the faster and more energy efficient an analog system is. All of these issues will impact the final classification accuracy of a neural network. To compensate for these issues and take advantage of large gains in energy and latency enabled by analog systems, neural algorithms will need to be designed specifically to overcome the hardware limitations. This will require new

co-design tools where the impact of device level properties can be evaluated on algorithmic performance and the requirements for new analog devices will be driven by algorithmic considerations.

Consequently, we have developed a new open source simulation tool called CrossSim [6] that allows for the impact of device level properties on algorithmic performance to be quantified. For algorithm designers this tool is designed to abstract away the crossbar to three key noisy mathematical operations: VMM, MVM and outer product update (OPU). The impact of different device properties or crossbar designs can then be studied by simply changing a few input parameters. Similarly, device designers can specify their measured device properties as input parameters and see quickly see how a new device would impact the accuracy of a neural network algorithm. CrossSim focuses on modelling the crossbar while allowing for arbitrary neuron models. This allows for many different algorithms to be built on top of the key computation kernels. The exact result of a matrix vector multiplication can be passed into a user specified neuron model, or analog to digital converter models can be used to quantize the output. The process of measuring a TaOx device, gathering statistics on it and simulating training is illustrated in Fig 5[7]. This methodology can be used to evaluate new programmable memory devices [8, 9].

Using CrossSim, new algorithmic techniques to compensate for device nonidealities such as *periodic carry* can be developed. Periodic carry uses multiple devices to represent a weight. Each device is used to represent increasing significance in a place value based number (i.e. base 10) that represents the weight while maintaining the benefit of a parallel update[10]. Using period carry allows an analog TaO_x ReRAM to reach within 1% of numerical accuracy as shown in Fig 6.

Overall we see that using analog crossbars can provide a fundamental O(N) energy scaling advantage over digital memories. Analyzing a particular 8-bit design shows that an analog accelerator has a 430X gain in energy and a 35X gain in latency over an optimized digital ASIC. In order to model the impact of non-ideal analog devices we have developed an open source simulation tool CrossSim that can be used to quickly compare new devices or test new algorithms. This allowed the development of a new technique called periodic carry that allows noisy TaOx ReRAM devices that would only train to 80% accuracy to now train to 97% accuracy, just 1% away from the ideal accuracy of 98%.

Acknowledgment:
This work was funded by Sandia National Laboratories Hardware Acceleration of Adaptive Neural Algorithms (HAANA) Grand Challenge Laboratory Directed Research and Development (LDRD) Project. Sandia National Laboratories is 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-NA0003525.

References:

[1] M. J. Marinella *et al.*, "Multiscale Co-Design Analysis of Energy, Latency, Area, and Accuracy of a ReRAM Analog Neural Training Accelerator," *arXiv preprint arXiv:1707.09952,* 2017.
[2] S. Agarwal *et al.*, "Energy Scaling Advantages of Resistive Memory Crossbar Based Computation and its Application to Sparse Coding," *Frontiers in Neuroscience,* vol. 9, p. 484, 2016, Art. no. 484.
[3] S. Agarwal *et al.*, "Resistive memory device requirements for a neural algorithm accelerator," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 929-938.
[4] Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits [Online]. Available: http://yann.lecun.com/exdb/mnist
[5] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *arXiv preprint arXiv:1602.02830,* 2016.
[6] S. Agarwal *et al.* (2017). *CrossSim*. Available: http://cross-sim.sandia.gov
[7] R. B. Jacobs-Gedrim *et al.*, "Impact of Linearity and Write Noise of Analog Resistive Memory Devices in a Neural Algorithm Accelerator," presented at the IEEE International Conference on Rebooting Computing (ICRC) Washington, DC, November 2017.
[8] Y. van de Burgt *et al.*, "A non-volatile organic electrochemical device as a low-voltage artificial synapse for neuromorphic computing," *Nat Mater,* Letter vol. 16, no. 4, pp. 414-418, 2017.
[9] E. J. Fuller *et al.*, "Li-Ion Synaptic Transistor for Low Power Analog Computing," *Advanced Materials,* vol. 29, no. 4, p. 1604310, 2017.
[10] S. Agarwal *et al.*, "Achieving ideal accuracies in analog neuromorphic computing using periodic carry," in *VLSI Technology, 2017 Symposium on*, 2017, pp. T174-T175: IEEE.

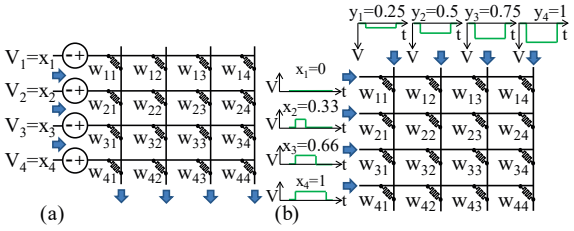

Fig. 1. (a) Analog crossbars reduce the energy of a vector-matrix multiply. The conductance of each resistive memory represents a weight. Analog input vector values are represented by pulse lengths, and output vector values are represented by integrated currents. This allows all the read operations, multiplication operations and sum operations to occur in a single step. A matrix-vector multiply is performed by driving the columns and reading the currents on the rows. (b) A parallel write is illustrated. Weight W_{ij} is updated by $x_i \times y_j$. In order to achieve a multiplicative effect the x_i are encoded in time while the yj are encoded in the height of a voltage pulse.

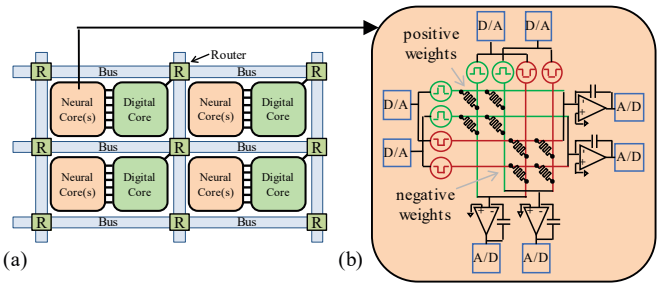

Fig. 2. (a) A general purpose neuromorphic architecture is shown. The neural cores perform matrix operations while the digital cores perform vector operations. (b) A neural core is illustrated. The inputs and outputs are converted to/from digital. Negative weights are modelled by taking the difference of two weights.

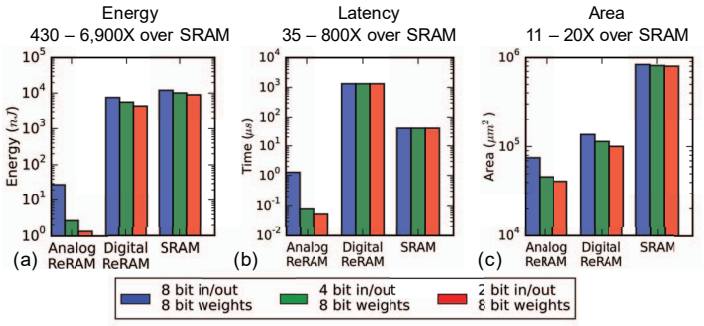

Fig 3: The (a) energy, (b) latency, and (c) area advantages for an analog accelerator from [1] are shown. The energy and latency for computing the three key kernel operations (VMM, MVM and OPU) are summed for a full 1024x1024 matrix. For the digital comparison it is assumed that the weights are eight bits.

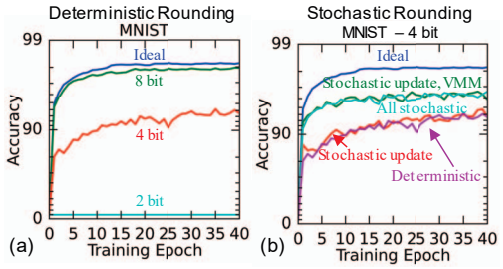

Fig 4: (a) The impact of reducing bit precision on the MNIST data sets is shown. (b) Using stochastic rounding can significantly increase the accuracy.

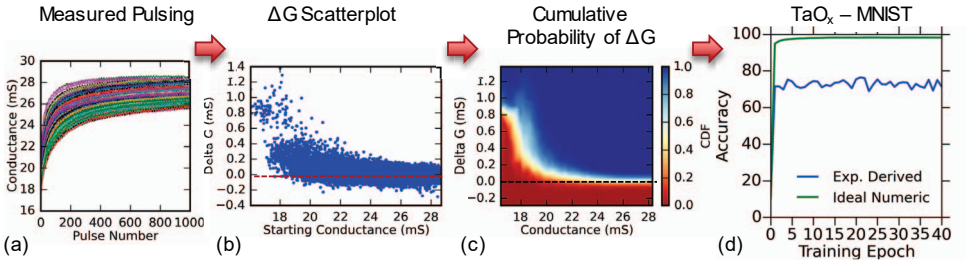

Fig 5: (a) A series of positive write pulses were applied to a TaOx ReRAM to gather write statistics (b) The measured change in conductance for a positive write pulse is shown vs starting state. (c) The measured change in conductance is used to create a cumulative distribution function (CDF) that CrossSim can use to model both the write noise and write nonlinearity. (d) The collected write statistics are used to simulate training in a neural network.

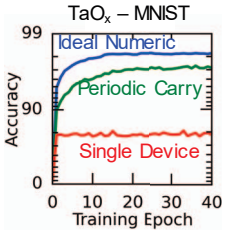

Fig 6: Using Periodic Carry can boost the accuracy from 80% to 97%, nearly ideal.