SAND2020-2406C

Evaluating complexity and resilience trade-offs in emerging memory inference machines

Christopher H. Bennett*, Ryan Dellana*, T. Patrick Xiao, Ben Feinberg, Sapan Agarwal,
Suma Cardwell, Matthew J. Marinella, William Severa, Brad Aimone
Center for Computing Research, Sandia National Laboratorie, Albuquerque, New Mexico
Email: cbennet,jbaimon@sandia.gov
*These authors contributed equally

***Abstract*—Neuromorphic engineering only works well if limited hardware resources are maximized properly, e.g. memory and computational elements, scale efficiently as the number of parameters relative to potential disturbance. In this work, we use realistic crossbar simulations to highlight a significant trade-off between the complexity of deep neural networks and their susceptibility to collapse from internal system disturbances. Although the simplest models are the most resilient, they cannot achieve competitive results. Our work proposes a middle path towards high performance and moderate resilience utilizing the *Mosaics* framework, by re-using synaptic connections in a recurrent neural network implementation.**

I. Introduction

A variety of neuromorphic systems have been implemented using emerging memory devices, but few perform at industrial levels due to the difficulties of implementing software-quality synapses and gradients. These challenges are somewhat simplified in the inference-only application, where imported weights from a separately pre-trained network are used to accelerate predictions on a known task. Nevertheless, electrical issues such as parasitic voltage drops, parasitic resistances, and cycle-to-cycle read noise may limit the ultimate size and accuracy of inference-only neuromorphic accelerators [1], [2]. These issues can quickly degrade the performance of deep networks with hundreds of thousands of parameters. Accelerators have been proposed to leverage memristive crossbars such as PUMA [3], ISAAC, [4], but they typically discuss the implications of these issues at minimal length. We explore the idea that these systems may be unexpectedly fragile to perturbations, and seek mitigations.

The use of temporal neural networks is a legitimately brain-inspired computing approach that may yield new sources of power and resilience [5]. In order to efficiently mapp standard networks to temporal ones, the efficiency of fundamental operations and components in a given graph must be considered. As illustrated in Fig. 4, while the overall size of an artificial neural network (ANN) graph may be very large, parts of the necessary computation are done repeatedly. *Mosaics* in this context refers to a temporal form of neuromorphic multiplexing, whereby certain computations (in this example, the neurons, which are often a limiting resource) can be reused for different stages of an algorithm; allowing larger scale algorithms to be implemented on a resource restricted neuromorphic platform. By exploring the partitioning of a neural graph into sub-graphs that enable the computation to be performed in isolation on a smaller subset of available computing nodes - trading space for time in computation - we open a new avenue for failure and resilience analysis. Concretely, in the following sections we explore this contrast through three relatively well-known ANNs: a) a simple, low parameter multi-layer perceptron, b) a complex, medium parameter convolutional neural network (CNN), and c) a medium-parameter, medium complexity recurrent neural network (RNN) inspired by *Mosaics* re-use concepts. In general, RNNs are an attractive emerging option for the demonstration of on-chip learning or inference, as they are powerful general computational models [6] . Recently, long-short-term memory (LSTM) networks have been the most heavily considered for implementation with dense non-volatile memory arrays [7]; however, such schemes involve complex crossbar partioning and the hardware implementation of transcendental functions. Feasibly, LSTM implementations can be done on-chip during inference (prediction) only mode [8], yet the exact energy and complexity overhead of such systems are unknown. Drawing on previous work which shows that an RNN network using apprpriate activation functions can still perform competitively to an LSTM on certain tasks [9], in the following sections we demonstrate that a carefully designed RNN system can perform competitively to a comparable CNN on two state of the art machine learning (ML) tasks, at a lower energy budget.

II. Methodology

In order to implement the previously described neural network models, we imported pre-trained Keras models [10] into CrossSim trained on the iconic MNIST data task [11] as well as the newer fashion-MNIST data task [12]. CrossSim is a crossbar simulation tool that helps model resisitve memory crossbars as highly parameterizable neural cores [13], and which has recently been extended to perform physics-rich analysis of inference operations. Our imported models for CNNs contain 119,322 trainable parameters and contain both convolutional and fully connected (FC) layers, each followed by a bounded rectified linear function (ReLU) as visible in Fig. 2(a); exact model is given in Table 1. [14] and as visible in Fig. 2(a). Our imported MLP models are standard consisting of one hidden layer (128 ReLU functions) and a logit of 10; this model has 101,770 trainable parameters. Our recurrent neural

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.

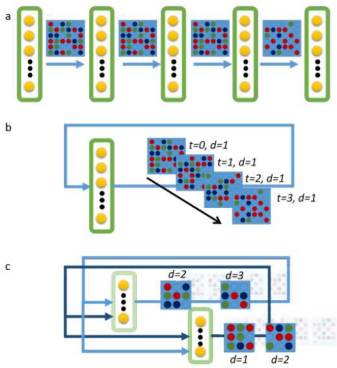

Fig. 1. (**a**) Simple ANN depiction with yellow sub-graphs as neurons and blue tiles as weights. (**b**) A recurrent Mosaic implementation, with processing neurons recurrently connected to a dynamically used single set of weights. (**c**) Future *Mosaics* implementation, with each layer broken into component sub-layers. This requires that the connection matrices be partitioned with different delays to permit connections to 'skip' layers to target appropriate node

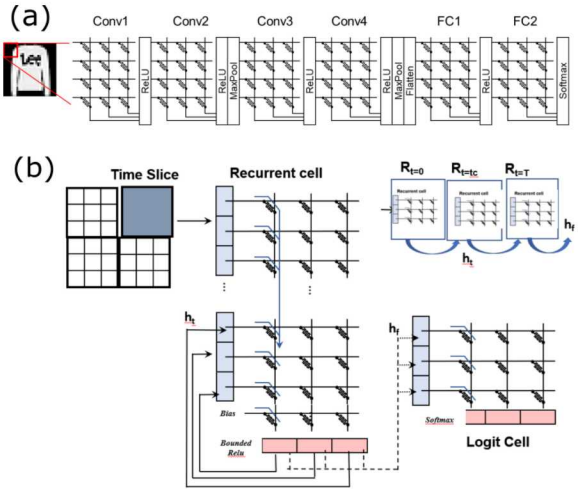

Fig. 2. (**a**) A CNN architecture is demonstrated; parts of images are fed to ANN sequentially. (**b**) As in the CNN, pieces of the image are fed to the recurrent cell until the entire image has been presented; now, the read-out or fully connected logit system only activates on the last time-step.

network design, newly developed for this work and visible in Fig. 2(b), consists of a recurrent matrix which is re-used during multiple time steps. Notably it is partioned into two cores, a core which always receives a component of the input and a part which receives the hidden layer activation h_t from the previous time step $t-1$ (or null if$t=0$) . There are 400 total hidden (input) neuron units, and depending on the chosen time steps t, the number of differentiable hidden nodes D_{hl} is given as $400 - M/t$, where $M = 784$ is the dimensionality of our task. The number of parameters of our RNN networks vary from 118,37 trainable parameters at $t = 7$ up to 158,656 parameters at $t = 56$ (time step integers are divisors of M). The exact paratmers used for all three major network varieties are given in Table 1.

The RNN core crossbar is connected to a read-out/logit crossbar of dimensions D_{hl} x L, where L is the number

TABLE I
PRIMARY SYSTEM ARCHITECTURES.

Neural Net Design	**Layout**
MLP	785x300, 301x10
RNN	301x400, (t time steps), 401x10 4
CNN	C3/3-C3/3-MP2-C3/6-C3/6-D100-D10

TABLE II
STANDARD RESILIENCE RESULTS ($t = 7$)

Architecture	Noise Scenario		
	Internal (σ_{syn}*)	External (σ_{te}*)	Both Effects
MLP- MNIST	96.8%	94.1%	93.1%
RNN - MNIST	97.4%	95.1%	94.9%
CNN-MNIST	98.5%	96.7%	96.05%
MLP- f-MNIST	82.2%	69.91%	62.35%
RNN - f-MNIST	86.3%	84.22%	81.11%
CNN-f-MNIST*	85.1%	57.91%	42.35%

*In all cases, $\sigma_{\text{syn}} = \sigma_{\text{te}} = 0.025$

of classes (here $L = 10$); critically, the second core is only activated at the final time step (every t steps). In our resilience testing strategy, we have considered two classes of pre-trained networks: noise-prepared or regularized networks, which train with jittered gaussian filters on every hidden neuron's ReLu function during training following the scheme given in [14] and standard/un-prepared networks which have been trained without noise. As first suggested in [15], the use of noise regularization provides a definitive improvement in inference (Test) performance of models to internal and external noise or perturbation effects. During training phase, noise centered around 0 with a distribution width σ_{neu} is injected into every ReLu neuron; during testing phase σ_{syn} is then injected on a synapse-per-synase basis . This device-by-device synaptic dispersion most closely relates to intrinsic noise effects that would occur during the operation of large arrays (Johnson-Nyquist noise), but can approximate shot noise or device-specific perturbations or weight variance as suggested in [1]. In addition, given the test set $\mathcal{Y}$, we have also considered the addition of additive gaussian noise at dispersion σ_{te} where $\mathcal{Y}_{\text{noisy}} = \mathcal{Y} + y(\sigma_{\text{te}})$. We have perturbed all weight matrices in every system consistently based on the generation of random numbers with seeds different in every run.

III. RESULTS

A. Resilience to noise sources

First, we consider the raw resilience of MLP, RNN, and CNN networks to noise at physically plausible values. As in Table II, all considered systems perform best with just the internal noise, second best with just external noise, and suffer the most when the effects are compounded. There is a significant task-dependence, with CNN systems performing best on MNIST (easier task) even in the worst-case, and with RNN systems performing by far the best on the fashion-MNIST (harder task) case. Of particular concern/interest is

the degradation of the CNN f-MNIST models to test-set noise degradation and combined noise degradatio, which does even worse than the MLP system. Although the possibility of catastrophic responses of CNN networks to adversarial perturbations are welll-known, this effect can probably be at least partially lessened by a more complex CNN architecture with more parameters and/or larger filter sizes [16].

Next, we consider broader sweeps of the internal noise parameter σ_{syn} along with a consideration of the effect of noise-regularization during inference time for the two best-performing systems overall (CNN, RNN). As visible in Figure 3, in both the MNIST (a,b) and f-MNIST (c,d) cases, the stochastic ReLU behavior during training helps the CNN systems far more than the RNN systems. For MNIST, the regularization helps the CNN achieve best overall performance at both the purely internal and combined noise source cases (blue improves to orange), while the regularization assists the RNN but not as dramatically (green to red). This same trend holds in fashion MNIST with only internal noise; however, in the combined noise case for fashion-MNIST, at $\sigma_{\text{syn}} < 0.075$ the regularized and non-regularized models are broadly superior to CNN approaches. Overall, the current results support the argument that RNNs seems to have greater intrinsic noise immunity as compared to CNNs, while CNNs benefit far more from a standard neuron-level regularization approach.

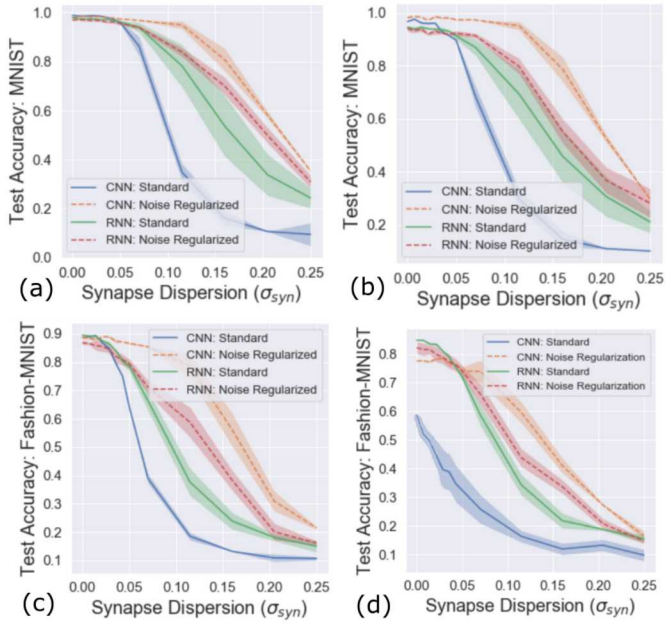

Fig. 3. (**a**) Test inference accuracy on the MNIST task for regularized and non-regularized CNN and RNN systems as a function of progressively increased σ_{syn} value; (**b**)) MNIST performance on highlighted systems given both internal and noisy test set scenario ($\sigma_{\text{te}} = 0.025$). (**c**) fashion-MNIST performance given internal noise and (**d**) both internal and test-set noise fashion-MNIST degradation (again $\sigma_{\text{te}} = 0.025$).

B. Temporal stacking sensitivity of RNN

The considered RNN approach can treat the number of time-steps of input presentation , which are automatically concatenated at the output of the RNN core, as a free parameter. As noted in [9], extending t too far can cause problems in learning convergence since the temporal dependencies (trace) between the eventual output and input stream can become too weak. We find a similar result however greater sensitivity in the NVM system. As shown in Figure 4, when $t = 26$ a slight degradation is visible, and at $t = 56$ inference at the task even at no noise loses $8 - 10\%$ depending on the task.

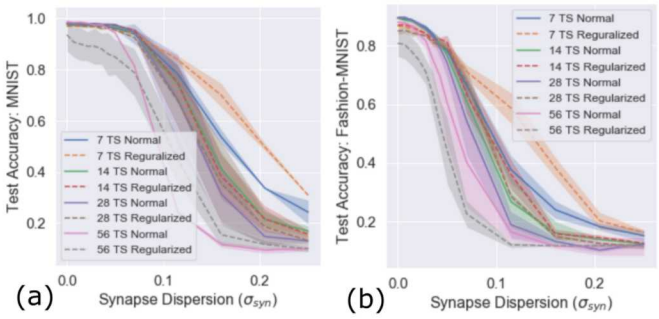

Fig. 4. (**a**) Test inference accuracy on the MNIST task as a function of the internally applied perturbation parameter for all of the considered time-step values , in both regularized and non-regularized procedures (**b**) Same analysis is presented for all pre-trained models on the fashion-MNIST task.

TABLE III
ENERGY OVERHEAD PER INFERENCE OP IN CONSIDERED SYSTEMS

Noise Mode	Synapse Type		
	Total Energy/Op	VMM Op	Neuron Activation Op
MLP ReRAM*	4.24 nJ	4.22nJ	15pJ
RNN ReRAM* †	35.6nJ	35.5nJ	66pJ
CNN ReRAM*	480 nJ	479 nJ	358 pJ
MLP SONOS*	6.04 nJ	6.02nJ	15pJ
RNN SONOS* †	42.7nJ	42.7nJ	66pJ
CNN SONOS*	2.084 μJ	2.084μJ	358 pJ

*We have assumed that softmax (all systems) and maxpool (CNN only) operations energy costs are negligible.
†RNN energy estimates are worst-case ($t = 7$). Note benefits are greater for greater number of t.

C. Energy overhead considerations

Lastly we combine elementary energy costs and system dimensions to estimate total energy overhead. Notably, there are two energy costs driving inference: the vector-matrix-multiply (VMM) operation, which requires signficant energy to charge and read out (given ADC costs) the memristive array, and the neuron activation energy costs. Using the gseneral anaylsis proposed in [17], the CMOS circuit for the ReLU function given in [18], and considering two strong device candidates for inference - optimized filamentary resistive RAM (ReRAM) alongside charge-trap based ultra-low energy SONOS memory [19]- we find that emerging memory CNN systems are energy hungry relative to the other options (Table III). Our proposed RNN system consumes only 7x more than the MLP system by exploiting the weight re-use or temporal encoding strategy, and saves between 13-38x energy compared to the competitor CNN system.

IV. Discussion

One phenomenon suggested but not proven in our analysis is the idea that noise regularization is still helpful , but less critical than in more complex CNN models, due to the presence of attractor basins in recurrent architectures which help to provide some intrinsic level of noise resilience [20]. We have also discovered that binary stochastic neurons may not be sufficient for full RNN regularization; a more complex function or method may be require. One interesting method suggested in [21] would be the use of temporal skip connections. Another fruitful extension of this work could be the exploration of how recurrent , convoutional , or mixed architectures using effective regularization mtehods (tuned based on the time-step dynamics of networks) could provide defensive properties against adversarial noise or perturbation effects [22], [23]. Finally, while we have showed the *Mosaics* concept implemented in the temporal domain in this work, the efficient parallel or horizontal stacking of convolution operations may yield more efficient and/or resilient convolutional operations (Figure 4c).

V. Conclusion

Making emerging memory inference systems more noise/perturbation reslient is an important goal for the neuromorphic engineering field, but results in this direction so far have focused almost exclusively on the limitations of CNN systems. In this work, we have put CNN resilience in conversation with other approaches and in particular our novel RNN approach. We have shown that an ostensibly simple recurrent neural network design efficiently implements the idea of temporal stacking or weight re-use necessary to reduce energy costs by at least 13.5x while still achieving results that are competitive with CNNs - at least on tasks of intermediate complexity. In the future, we plan to extend our physics-aware methods to analyse the cross-over points at which various deeper ANNs scale to state-of-the-art tasks given both temporally and physically sequential sub-systems.

Acknowledgment

Sandia National Laboratories is a multimission laboratory managed and operated by NTESS, 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. 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] S. Agarwal, S. J. Plimpton, D. R. Hughart, A. H. Hsia, I. Richter, J. A. Cox, C. D. James, and M. J. Marinella, "Resistive memory device requirements for a neural algorithm accelerator," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 929–938.

[2] M. Bavandpour, M. Mahmoodi, H. Nili, F. M. Bayat, M. Prezioso, A. Vincent, D. Strukov, and K. Likharev, "Mixed-signal neuromorphic inference accelerators: Recent results and future prospects," in *2018 IEEE International Electron Devices Meeting (IEDM)*. IEEE, 2018, pp. 20–4.

[3] A. Ankit, I. E. Hajj, S. R. Chalamalasetti, G. Ndu, M. Foltin, R. S. Williams, P. Faraboschi, W.-m. W. Hwu, J. P. Strachan, K. Roy *et al.*, "Puma: A programmable ultra-efficient memristor-based accelerator for machine learning inference," in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 2019, pp. 715–731.

[4] M. N. Bojnordi and E. Ipek, "Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning," in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2016, pp. 1–13.

[5] J. B. Aimone, "Neural algorithms and computing beyond moore's law," *Communications of the ACM*, vol. 62, no. 4, pp. 110–110, 2019.

[6] B. Hammer, "On the approximation capability of recurrent neural networks," *Neurocomputing*, vol. 31, no. 1-4, pp. 107–123, 2000.

[7] T. Gokmen, M. J. Rasch, and W. Haensch, "Training lstm networks with resistive cross-point devices," *Frontiers in neuroscience*, vol. 12, p. 745, 2018.

[8] H. Tsai, S. Ambrogio, C. Mackin, P. Narayanan, R. Shelby, K. Rocki, A. Chen, and G. Burr, "Inference of long-short term memory networks at software-equivalent accuracy using 2.5 m analog phase change memory devices," in *2019 Symposium on VLSI Technology*. IEEE, 2019, pp. T82–T83.

[9] Q. V. Le, N. Jaitly, and G. E. Hinton, "A simple way to initialize recurrent networks of rectified linear units," *arXiv preprint arXiv:1504.00941*, 2015.

[10] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[11] Y. LeCun, C. Cortes, and C. Burges, "Mnist handwritten digit database," *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, vol. 2, p. 18, 2010.

[12] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[13] *Crossbar Simulator: CrossSim*, 2019, accessed October 2nd, 2019. [Online]. Available: https://cross-sim.sandia.gov

[14] C. Bennett, T. P. Xiao, R. Dellana, B. Feinberg, K. Prabhakar, V.and Ramkumar, V. Agrawal, L.Hinh, S.Saha, V. Raghavan, R. Chettuvetty, S. Agarwal, and M. Marinella, "Device-aware inference operations in sonos nonvolatile memory arrays," in *2020 IEEE International Reliability Physics Symposium (IRPS)*. IEEE, 2020, pp. 1–6.

[15] H. Noh, T. You, J. Mun, and B. Han, "Regularizing deep neural networks by noise: Its interpretation and optimization," in *Advances in Neural Information Processing Systems*, 2017, pp. 5109–5118.

[16] T.-J. Yang and V. Sze, "Design considerations for efficient deep neural networks on processing-in-memory accelerators," *arXiv preprint arXiv:1912.12167*, 2019.

[17] M. J. Marinella, S. Agarwal, A. Hsia, I. Richter, R. Jacobs-Gedrim, J. Niroula, S. J. Plimpton, E. Ipek, and C. D. James, "Multiscale co-design analysis of energy, latency, area, and accuracy of a reram analog neural training accelerator," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 1, pp. 86–101, 2018.

[18] P. Priyanka, G. Nisarga, and S. Raghuram, "Cmos implementations of rectified linear activation function," in *International Symposium on VLSI Design and Test*. Springer, 2018, pp. 121–129.

[19] S. Agarwal, R. B. Jacobs-Gedrim, C. Bennett, A. Hsia, M. S. Van Heukelom, D. Hughart, E. Fuller, Y. Li, A. A. Talin, and M. J. Marinella, "Designing and modeling analog neural network training accelerators," in *2019 International Symposium on VLSI Technology, Systems and Application (VLSI-TSA)*. IEEE, 2019, pp. 1–2.

[20] A. Ceni, P. Ashwin, and L. Livi, "Interpreting recurrent neural networks behaviour via excitable network attractors," *Cognitive Computation*, pp. 1–27, 2019.

[21] V. Campos, B. Jou, X. Giró-i Nieto, J. Torres, and S.-F. Chang, "Skip rnn: Learning to skip state updates in recurrent neural networks," *arXiv preprint arXiv:1708.06834*, 2017.

[22] Z. You, J. Ye, K. Li, Z. Xu, and P. Wang, "Adversarial noise layer: Regularize neural network by adding noise," in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 909–913.

[23] J. Jin, A. Dundar, and E. Culurciello, "Robust convolutional neural networks under adversarial noise," *arXiv preprint arXiv:1511.06306*, 2015.