Large-scale neural networks implemented with non-volatile memory as the synaptic weight element: comparative performance analysis (accuracy, speed, and power)

G.W. Burr, P. Narayanan, R. M. Shelby, S. Sidler, I. Boybat, C. di Nolfo, and Y. Leblebici†
IBM Research–Almaden, 650 Harry Road, San Jose, CA 95120, Tel: (408) 927–1512, E-mail: gwbur@us.ibm.com
†EPFL, Lausanne, CH–1015 Switzerland

Abstract
We review our work towards achieving competitive performance (classification accuracies) for on-chip machine learning (ML) of large-scale artificial neural networks (ANN) using Non-Volatile Memory (NVM)-based synapses, despite the inherent random and deterministic imperfections of such devices. We then show that such systems could potentially offer faster (up to $25 \times$) and lower-power (from $120–2850 \times$) ML training than GPU–based hardware.

Introduction
By performing computation at the location of data, non-Von Neumann (non–VN) computing offered to provide power and speed benefits (Fig. 1). For one such non–VN approach — on-chip training of large-scale ANN using NVM-based synapses [1,2] — viability will require (at least) two things. First, despite the inherent imperfections of NVM devices such as Phase Change Memory (PCM)[1] or Resistive RAM (RRAM)[3], such NVM-based networks must achieve competitive performance levels (e.g., classification accuracies) vs. ANN trained using CPUs or GPUs. Second, the benefits of performing computation at the data (Fig. 2) must confer a decided advantage in either training power or speed (or preferably, both).

We review our work [1-3] towards competitive training accuracies, and provide an initial assessment of the feasibility of achieving lower power and/or faster training with on-chip NVM-based ML.

Potential for competitive classification accuracies
Using 2 phase-change memory (PCM) devices per synapse, we demonstrated a 3–layer perceptron with 164,885 synapses[1], trained with backpropagation[4] on a subset (5000 examples) of the MNIST database of handwritten digits[5] (Fig 3), using a modified weight-update rule compatible with NVM+selector crossbar arrays[1]. This modification does not degrade the high “test” (generalization) accuracies such a 3–layer network inherently delivers on this problem when trained in software (ranging from >94% with 5000 training examples, up to >97% with the full set of 60,000 examples, Fig. 4) [1]. However, nonlinearity and asymmetry in PCM conductance response limited both “training” and “test” accuracy in these initial experiments to 82–83% [1] (Fig. 5).

Asymmetry (between the gentle conductance increases of PCM partial–SET and the abruptness of PCM RESET) was mitigated by an occasional RESET strategy, which could be both infrequent and inaccurate [1]. While in these initial experiments, network parameters such as learning rate $\eta$ had to be tuned very carefully, we have developed a modified ‘LG’ algorithm which offers wider tolerance to $\eta$, higher classification accuracies, and lower training energy (Fig. 6) [1]. (Hardware experiments are in progress, and will be published in due course.)
Fig. 5 Training accuracy for a 3-layer perceptron of 164,885 hardware-synapses [1], with all weight operations taking place on a 500 × 661 array of mushroom-cell PCM devices. Also shown is a matched computer simulation of this NN, using parameters extracted from the experiment [1].

Tolerancing results showed that all NVM-based ANN can be expected to be highly resilient to random effects (NVM variability, yield, and stochasticity), but highly sensitive to “gradient” effects that act to steer all synaptic weights [1]. We showed that a bidirectional NVM with a symmetric, linear conductance response of finite but large dynamic range (e.g., each conductance step is relatively small) can deliver the same high classification accuracies on the MNIST digits as a conventional, software-based implementation (Figs. 7, 8). The ‘LG’ algorithm, possibly with other approaches, then helps a nonlinear, asymmetric NVM (such as PCM) act more like this ideal linear, bidirectional NVM, potentially leading to similarly high accuracies. The key is avoiding constraints on weight magnitude that arise when the two conductances are either both small or both large — e.g., synapses should remain in the center stripe of the “G-diamond” (Fig. 9).

Comparative analysis of speed and power

In PCM–based ML, weights are efficiently summed by column-wise reads in forward propagation (row-wise reads for backpropagation) (Fig. 10), with \( x \) and \( \delta \) values encoded by a mixture of read-voltage and -duration (Fig. 11). The large aggregate read currents (~5uA) can be readily integrated, followed by quick digitization that need not be overly precise. Data is then passed to the block(s) representing the next (previous) ANN layer. Parallel write is fast because partial-SET pulses can be quite short (in [1], 25ns), although long delays will be incurred on the small subset of rows affected by occasional RESET (Fig. 11). Critical is the design of area-efficient read/write circuitry, so that many copies of this circuitry operate in parallel (each handling a small number of columns (rows), \( t \)). Energy is needed to activate the long bitlines and wordlines, and to perform the digitization, but write energy can be kept reasonable if learning rate is low (minimal # of devices written per example or in need of occasional RESET) (Fig. 12).

GPU execution time is estimated for five example ANN configurations (Fig. 13) from the associated memory and computation requirements with respect to the available GPU resources (Fig. 14).
For an ANN that occupies a small portion of its GPU, it seems that only a portion of the total bandwidth and computation are effectively available (Figs. 15, 16). Fig. 17 compares predicted training time (per ANN example) and power for two configurations of PCM–based on-chip machine learning against conventional GPU training. Under aggressive assumptions for parallel–read and –write speed, PCM-based on-chip machine learning can potentially offer lower power and faster training for both large and small networks. As circuit sharing, t, increases, the speed benefits of PCM-based on-chip machine learning disappear (Fig. 18). Performing occasional RESET too frequently can also degrade the speed benefits of PCM-based on-chip machine learning (Fig. 19).

Conclusions
We have assessed the potential advantages, in terms of speed and power, of on-chip machine learning (ML) of large-scale artificial neural networks (ANN) using Non-Volatile Memory (NVM)-based synapses, in comparison to conventional GPU–based hardware. We have (almost certainly) overlooked important considerations, such as how data might pass between (and be aggregated amongst) multiple PCM blocks, particularly since these blocks will need to be flexibly combined to form different ANN. However, since this quick analysis has shown some prospect for speed/power savings, we recommend further consideration and potentially different strategies for implementing on-chip machine learning.

Fig. 10 For an ANN with N neuron-layers, read of each (1000 × 1000) PCM block is parallel, then serialized across layers (N – 1 column-wise in forward propagation, N – 2 row-wise for backpropagation). Shared circuitry handles t columns (rows), integrating current (∼5uA) from 1000 devices, digitizing with nonlinearity, then passing to the next block. Conservative (aggressive) assumption: G+ and G– currents integrated and digitized serially (in parallel).

**Parallel reads**
Conservative: \((8\text{usec} \times t) \times (2N - 3)\)
Aggressive: \((400\text{nsec} \times t) \times (2N - 3)\)

**Fig. 11** During backpropagation, each block receives a series of four ∼25ns partial-SET pulses from downstream and upstream neurons, repeated t times because of the shared circuitry. After every R examples, all PCM elements must be individually read, with some 5% of rows receiving RESET pulses followed by a relatively long, multiple–SET procedure [1].

**Occasional RESET**
Conservative: \((1\text{usec} \times t \times 1000 \text{rows}) \times (10\text{usec} \times t \times 50 \text{rows})\)
Aggressive: \((400\text{nsec} \times t \times 1000 \text{rows}) \times (3\text{nsec} \times t \times 50 \text{rows})\)

**Parallel writes**
Conservative: \((1\text{usec} \times t \times (N - 1))\)
Aggressive: \((200\text{nsec} \times t \times (N - 1))\)

Fig. 12 For each (1000 × 1000) PCM block, parallel reads require bitline & wordline activation plus the digitization. Partial–SET write energy activates all wires, but only a fraction of devices are programmed for any one training example. Occasional RESET requires full array reads, but only a few devices receive RESETs followed by multiple SET pulses [1].
Fig. 17 Predicted training time (per ANN example) and power for 5 ANNs (Fig. 13), ranging from 0.2GB to nearly 6GB. Network #1 assumes Tesla K10 w/o momentum [6,7]; #2-#5, Tesla K20x and momentum [7]. Minibatch size is 1024, RESET interval $R=200$, $t=4$, solid (dotted) line assumes 50W (20W) idle GPU power. Under aggressive assumptions for parallel–read and –write speed, PCM–based on–chip machine learning could potentially offer lower power and faster training for both large and small networks.

Critical design considerations, such as the importance of minimal circuit sharing (low $t$), have been identified. Thus the design of highly area–efficient circuits enabling read and write of many synaptic bitlines (and during backpropagation, of many wordlines) in parallel will be critical. Continued improvements in ML training algorithms that can deliver competitively–high ANN classification accuracies despite the nonlinearity and asymmetry of NVM such as PCM will also be important.

Future work should explore important circuit–design tradeoffs (higher precision in $x$, $\delta$ and nonlinearity $f$ to maintain high accuracy vs. the consequences in circuit area and in training speed and power). The present work on multilayer perceptrons should also be extended to convolutional neural networks, both in terms of accuracy assessment when using NVM–based synapse arrays as well as a similar speed and power assessment.

References

Fig. 16 From the same set of measurements for network #1 [6], our simple GPU–model (Fig. 14) can also predict execution times reasonably well, for both computation–light and computation–heavy matrix–multiplication subtasks.

Fig. 18 As circuit sharing, $t$, increases, the speed benefits of PCM–based on–chip machine learning disappear. Thus the design of highly area–efficient circuits enabling read and write of many synaptic bitlines (and during backpropagation, wordlines) in parallel will be critical.

Fig. 19 Having to perform the ‘occasional RESET’ step too frequently can also degrade the speed benefits of PCM–based on–chip machine learning. However, a RESET interval of 200–500 is large enough that the time and energy spent on occasional RESET are minor compared to the partial–SET training.