Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element

G. W. Burr, R. M. Shelby, C. di Nolfo, J. W. Jang‡, R. S. Shenoy, P. Narayanan,
K. Virwani, E. U. Giacometti, B. Kurdi, and H. Hwang‡

IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, Tel: (408) 927–1512, E-mail: *gwburr@us.ibm.com*
‡Pohang University of Science and Technology, Pohang, South Korea

Abstract

Using 2 phase-change memory (PCM) devices per synapse, a 3–layer perceptron network with 164,885 synapses is trained on a subset (5000 examples) of the MNIST database of handwritten digits using a backpropagation variant suitable for NVM+selector crossbar arrays, obtaining a training (generalization) accuracy of 82.2% (82.9%). Using a neural network (NN) simulator matched to the experimental demonstrator, extensive tolerancing is performed with respect to NVM variability, yield, and the stochasticity, linearity and asymmetry of NVM-conductance response.

Introduction

Dense arrays of nonvolatile memory (NVM) and selector device-pairs (Fig. 1) can implement neuro-inspired non-Von Neumann computing [1,2], using pairs [2] of NVM devices as programmable (plastic) bipolar synapses. Work to date has emphasized the Spike-Timing-Dependent-Plasticity (STDP) algorithm [1,2], motivated by synaptic measurements in real brains, yet experimental NVM demonstrations have been limited in size ($\leq$100 synapses).

Unlike STDP, backpropagation [3] is a widely-used, well-studied NN, offering benchmark-able performance on datasets such as handwritten digits (MNIST) [4]. In forward evaluation of a multilayer perceptron, each layer's inputs (x_i) drive the next layer's neurons through weights w_{ij} and a nonlinearity $f()$ (Fig. 2). Supervised learning occurs (Fig. 3) by back-propagating error terms δ_j to adjust each weight w_{ij}. A 3–layer network is capable of accuracies, on previously unseen 'test' images (*generalization*), of ~97% [4] (Fig. 4); even higher accuracy is possible by first "pre-training" the weights in each layer [5]. Like STDP, low-power neurons should be achievable by emphasizing brief spikes[7] and local-only clocking.

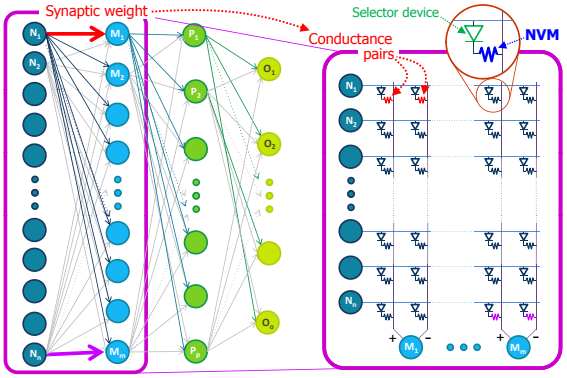

Fig. 1 Neuro-inspired non-Von Neumann computing [1,2], in which neurons activate each other through dense networks of programmable synaptic weights, can be implemented using dense crossbar arrays of nonvolatile memory (NVM) and selector device-pairs.

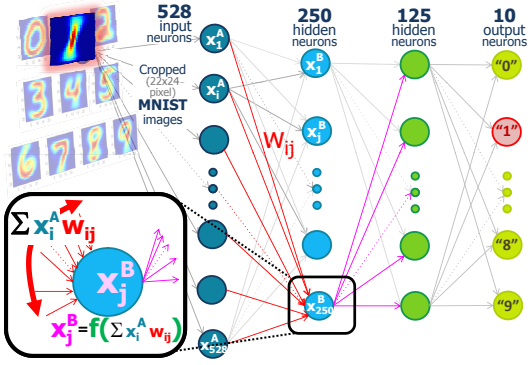

Fig. 2 In forward evaluation of a multilayer perceptron, each layer's neurons drive the next layer through weights w_{ij} and a nonlinearity $f()$. Input neurons are driven by pixels from successive MNIST images (cropped to 22×24); the 10 output neurons identify which digit was presented.

Considerations for a crossbar implementation

By encoding synaptic weight in conductance difference between paired NVMs, $w_{ij} = G^+ - G^-$ [2], forward propagation simply compares total read signal on bitlines (Fig. 5). However, backpropagation [3] calls for weight updates $\Delta w \propto x_i \delta_j$ (Fig. 6), requiring upstream i and downstream j neurons to exchange information for each synapse. In a crossbar, learning becomes much more efficient when neurons modify weights in parallel, by firing pulses whose overlap at the various NVM devices implements training [1] (Fig. 7). Fig. 8 shows, using a simulation of the NN in Figs. 2,3, that this adaptation for NVM implementation has no effect on accuracy.

However, the conductance response of any real NVM device exhibits imperfections that could still decidedly affect NN performance, including nonlinearity, stochasticity, varying maxima, asymmetry between increasing/decreasing responses, and non-responsive devices at low or high conductance (Fig. 9). **This paper explores the relative importance of each of these factors.**

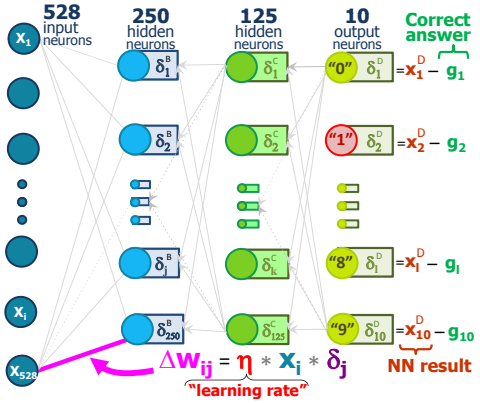

Fig. 3 In supervised learning, error terms δ_j are back-propagated, adjusting each weight w_{ij} to minimize an "energy" function by gradient descent, reducing classification error between computed (x_l^D) and desired output vectors (g_l).

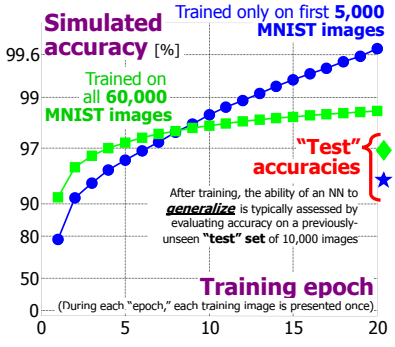

Fig. 4 A 3–layer perceptron network can classify previously unseen ('test') MNIST handwritten digits with up to ~97% accuracy[4]. Training on a subset of the images sacrifices some generalization accuracy but speeds up training.

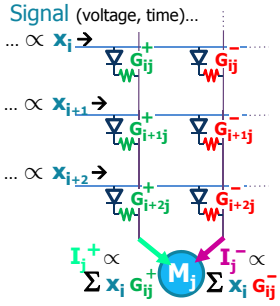

Fig. 5 By comparing total read signal between pairs of bitlines, summation of synaptic weights (encoded as conductance differences, $w_{ij} = G^+ - G^-$) is highly parallel.

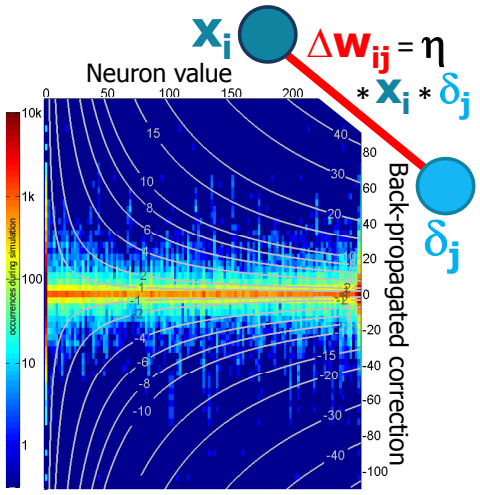

Fig. 6 Back-propagation calls for each weight to be updated by $\Delta w = \eta x_i w_{ij}$, where η is the *learning rate*. Colormap shows log(occurrences), in the 1^{st} layer, during NN training (blue curve, Fig. 4); white contours identify the quantized increase in the integer weight.

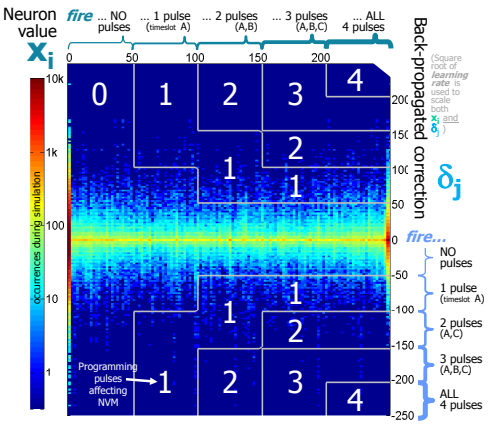

Fig. 7 In a crossbar, efficient learning requires neurons to update weights in parallel, firing pulses whose overlap at the various NVM devices implements training.

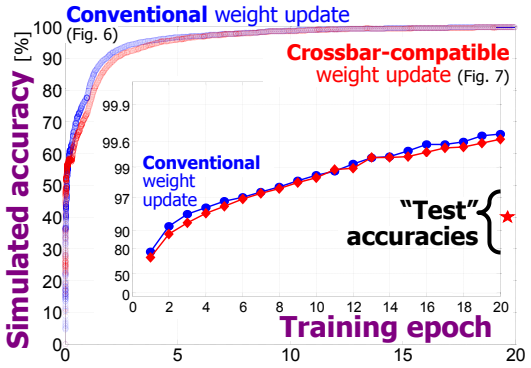

Fig. 8 Computer NN simulations show that a crossbar-compatible weight-update rule (Fig. 7) is just as effective as the conventional update rule (Fig. 6).

While bounding G values reduces NN training accuracy slightly (Fig. 10), unidirectionality and nonlinearity in the G-response strongly degrade accuracy. Figure insets (Fig. 10) map NVM-pair synapse states on a diamond-shaped plot of G^+ vs. G^- (weight is vertical position). In this context (Fig. 11), a synapse with a highly *asymmetric* G-response moves only unidirectionally, from left-to-right. Once one G is saturated, subsequent training can only increase the other G value, reducing weight magnitude, deleting trained information, and degrading accuracy. *Nonlinearity* in G-response further encourages weights of low value (Fig. 11), which can lead to network "freeze-out" (no weight changes, Fig. 10 inset). One solution to the highly asymmetric response of PCM devices is occasional RESET [2], moving synapses back to the left edge of the "G-diamond" while preserving weight value (with iterative SETs, Fig. 12 inset).

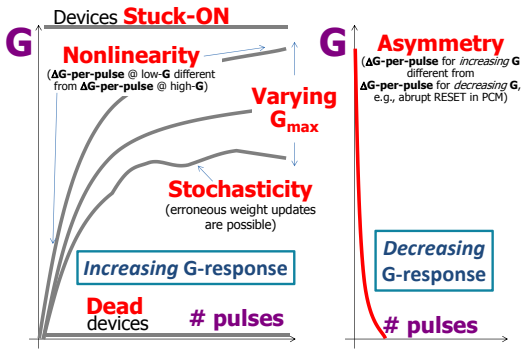

Fig. 9 The conductance response of an NVM device exhibits imperfections, including nonlinearity, stochasticity, varying maxima, asymmetry between increasing/decreasing responses, and non-responsive devices (at low or high G).

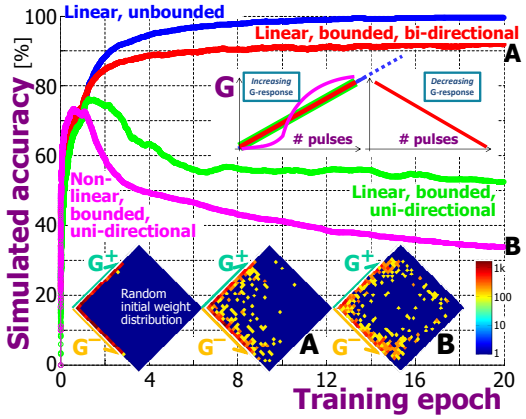

Fig. 10 Bounding G values reduces NN training accuracy slightly, but unidirectionality and nonlinearity in G-response strongly degrade accuracy. Figure insets map NVM-pair synapse states on a diamond-shaped plot of G^+ vs. G^- (weight is vertical position) for a sampled subset of the weights.

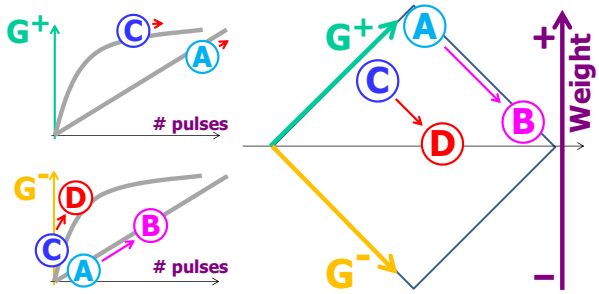

Fig. 11 If G values can only be increased (asymmetric G-response), a synapse at point **A** (G^+ saturated) can only increase G^-, leading to a low weight value (**B**). If response at small G values differs from that at large G (nonlinear G-response), alternating weight updates can no longer cancel. As synapses tend to get herded into the same portion of the G-diamond (**C** → **D**), the decrease in average weight can lead to network freeze-out.

However, if this is not done frequently enough, weight stagnation will degrade NN accuracy (Fig. 12).

Experimental results

We implemented a 3–layer perceptron of 164,885 synapses (Figs. 2,3) on a 500 × 661 array of mushroom-cell [6], 1T1R PCM devices (180nm node, Fig. 13). While the update algorithm (Fig. 7) is fully compatible with a crossbar implementation, our hardware allows only sequential access to each PCM device (Fig. 14). For read, a sense amplifier measures G values and thus weights for the software-based neurons, mimicking column- and row-based integrations. Weights are increased (decreased) by identical "partial–SET" pulses (Fig. 7) to increase G^+ (G^-) (Fig. 15). The deviation

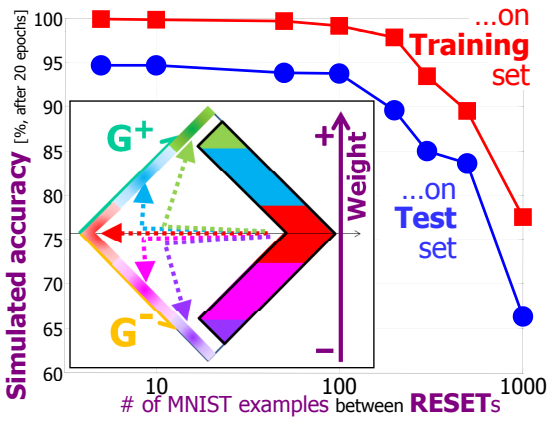

Fig. 12 Synapses with large conductance values (inset, right edge of G-diamond) can be refreshed (moved left) while preserving the weight (to some accuracy), by RESETs to both G followed by a partial SET of one. If such RESETs are too infrequent, weight evolution stagnates and NN accuracy degrades.

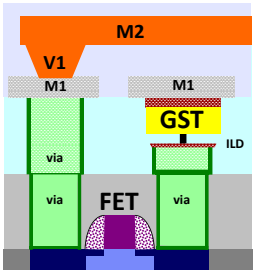

Fig. 13 Mushroom-cell [6], 1T1R PCM devices (180nm node) with 2 metal interconnect layers enable 512×1024 arrays.

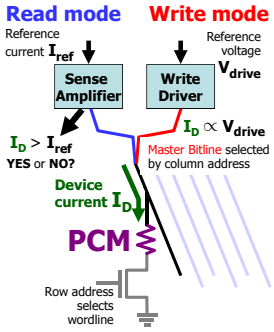

Fig. 14 A 1-bit sense amplifier measures G values, passing the data to software-based neurons. Conductances are increased by identical 25ns "partial–SET" pulses to increase G^+ (G^-) (Fig. 7), or by RESETs to both G followed by an iterative SET procedure (Fig. 12).

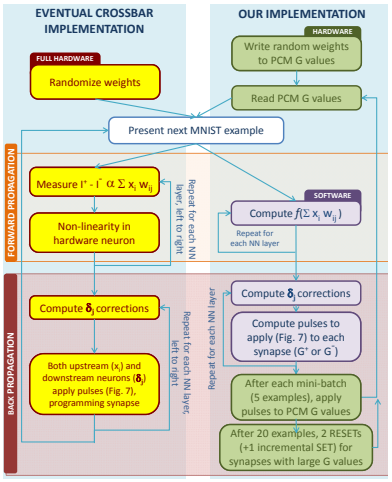

Fig. 15 Although G values are measured sequentially, weight summation and weight update procedures in our software-based neurons closely mimic the column (and row) integrations and pulse-overlap programming needed for parallel operations across a crossbar array. However, since occasional RESET is triggered when both G^+ and G^- are large, serial device access is required to obtain and then re-program individual conductances.

from true crossbar implementation occurs upon occasional RESET (Fig. 12), triggered when either G^+ or G^- are large, thus requiring both knowledge of and control over individual G values.

Fig. 16 shows measured accuracies for a hardware-synapse NN, with **all weight operations taking place on PCM devices**. To reduce test time, weight updates for each *mini-batch* of 5 MNIST examples were applied together. Fig. 17 plots measured G-response, stochasticity, variability, stuck-ON pixel rate, and RESET accuracy. By matching all parameters including stochasticity (Fig. 18) to those measured during the experiment, a NN computer simulation can

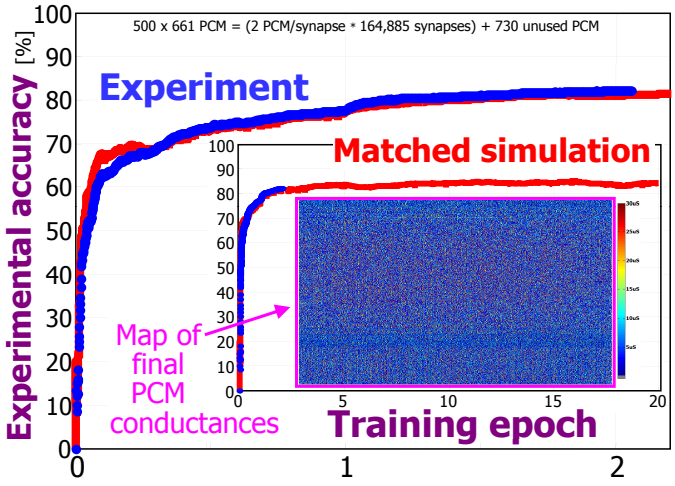

Fig. 16 Training and test accuracy for a 3–layer perceptron of 164,885 hardware-synapses, with all weight operations taking place on a 500 × 661 array of mushroom-cell [6] PCM devices (Fig. 13). Also shown is a matched computer simulation of this NN, using parameters extracted from the experiment.

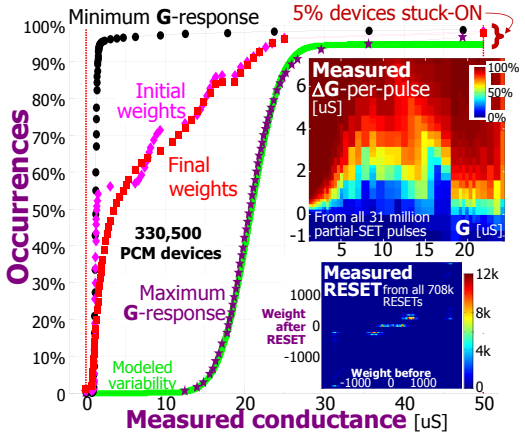

Fig. 17 50-point cumulative distributions of experimentally measured conductances for the 500 × 661 PCM array, showing variability and stuck-ON pixel rate. Insets shows the measured RESET accuracy, and the rate and stochasticity of G-response, plotted as a colormap of ΔG-per-pulse vs. G.

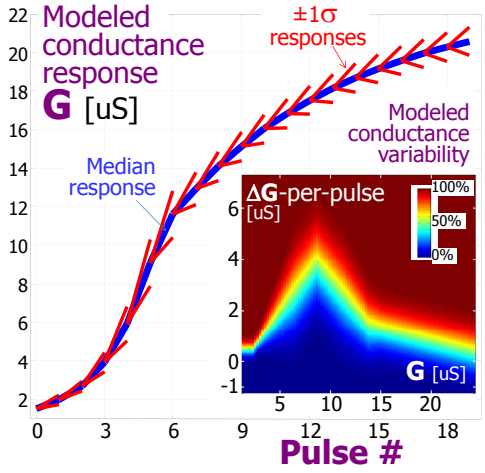

Fig. 18 Fitted G-response vs. # of pulses (blue average, red $\pm 1\sigma$ responses), obtained from our computer model (inset) for the rate and stochasticity of G-response (ΔG-per-pulse vs. G) matched to experiment (Fig. 17).

precisely reproduce the measured accuracy trends (Fig. 16).

Tolerancing and power considerations

We use this matched NN simulation to explore the importance

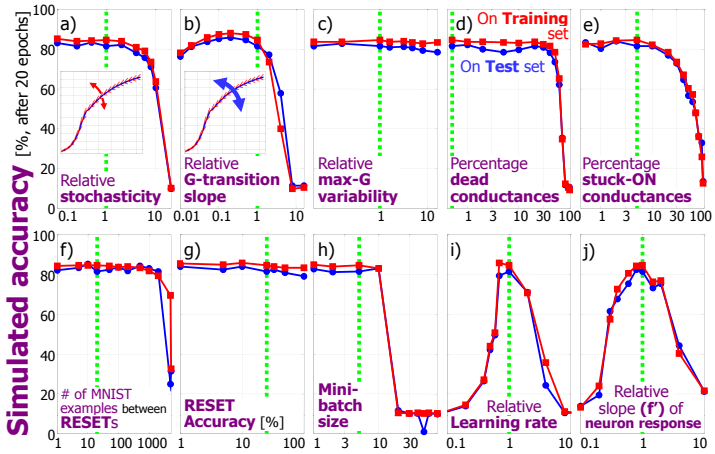

Fig. 19 Matched simulations show that an NVM-based NN is highly robust to stochasticity, variable maxima, the presence of non-responsive devices, and infrequent or inaccurate RESETs. Mini-batch size = 1 avoids the need to accumulate weight updates before applying them. However, the nonlinear and asymmetric G-response limits accuracy to ∼85%, and require learning-rate and neuron-response (f') to be precisely tuned.

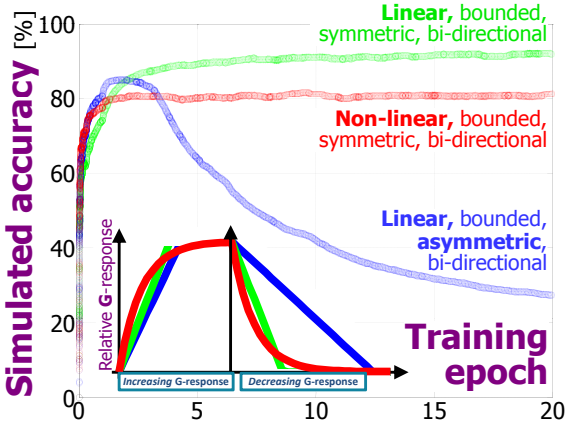

Fig. 20 NN performance is improved if G-response is linear and symmetric (green curve) rather than nonlinear (red). However, asymmetry between the up- and down-going G-responses (blue), if not corrected in the weight-update rule (Fig. 7), can strongly degrade performance by favoring particular regions of the G-diamond (Figs. 10,11).

of NVM imperfections. Fig. 19 shows final training (test) accuracy as a function of variations in NVM and NN parameters. NN performance is highly robust to stochasticity, variable maxima, the presence of non-responsive devices, and infrequent RESETs. A mini-batch of size 1 allows weight updates to be applied immediately. However, as mentioned earlier, nonlinearity and asymmetry in G-response limit the maximum possible accuracy (here, to ∼85%), and require precise tuning of the learning rate and neuron-response (f'). Too low a learning rate and no weights receive any updates; too high, and the imperfections in the NVM response generate chaos.

NN performance with NVM-based synapses offers high accuracy if G-response is linear and symmetric (Fig. 20, green curve) rather than nonlinear (red curve). Asymmetry in G-response (blue curve) strongly degrades performance. While the asymmetric G-response of PCM makes it necessary to occasionally stop training, measure all conductances, and apply RESETs and iterative SETs, energy usage can be reasonable if RESETs are infrequent (Fig. 21, inset), and learning rate is low (Fig. 21).

Conclusions

Using 2 phase-change memory (PCM) devices per synapse, a 3–layer perceptron with 164,885 synapses was trained with back-propagation on a subset (5000 examples) of the MNIST database of handwritten digits to high accuracy of (82.2%, 82.9% on test set). A weight-update rule compatible for NVM+selector crossbar arrays was developed; the "G-diamond" concept illustrates issues created by nonlinearity and asymmetry in NVM conductance response. Using a neural network (NN) simulator matched to the experimental demonstrator, extensive tolerancing was performed (Fig. 22). NVM-based NN are **highly resilient to random effects** (NVM variability, yield, and stochasticity), but **highly sensitive to "gradient" effects that act to steer all synaptic weights**. A learning-rate just high enough to avoid network "freeze-out" is shown to be advantageous for both high accuracy and low training energy.

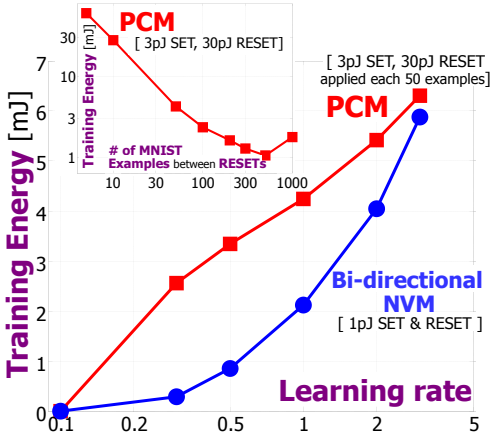

Fig. 21 Despite the higher power involved in RESET rather than partial-SET (30pJ and 3pJ for highly-scaled PCM [1]), total energy costs of training can be minimized if RESETs are sufficiently infrequent (inset). Low-energy training requires low learning rates, which minimizes the number of synaptic programming pulses. At higher learning rates, even a bi-directional, linear RRAM requiring no RESET and offering low-power (1pJ per pulse) can lead to large training energy.

Table of conclusions

Large 3-layer network with 2 PCM devices/synapse. (Figs. 1-3, 5) Back-propagation weight update rule compatible with crossbar array. (Figs. 6-8)	Moderately high accuracy (82%) achieved on MNIST handwritten digit recognition with two training epochs. (Fig. 16)
NVM models identified issues for training: Conductance bounds, nonlinearity, and asymmetry must be considered. (Figs. 9-10, 20)	PCM response and asymmetry mitigated by RESET strategy, mapping of response, choice of update pulse. (Figs. 11, 12, 17)
Model of PCM allows well-matched simulation of experiment (Figs. 16-18), variation of network parameters allows tolerancing. (Fig. 19)	NN is resilient to NVM variations (Figs. 19a-e) and RESET strategy (Figs. 19f-g), but sensitive to learning rate and neuron response function. (Figs. 19i-j)
Bidirectional NVM with no special RESET strategy and good performance requires scheme for symmetric response. (Fig. 20)	For PCM, keeping RESET frequency down and learning rate above "freeze-out" threshold allows reasonable training energy. (Fig. 21)

Fig. 22 NN built with NVM-based synapses tend to be highly sensitive to "gradient" effects (nonlinearity and asymmetry in G-response) that "steer" all synaptic weights towards either high or low values, yet are highly resilient to random effects (NVM variability, yield, and stochasticity).

References

[1] B. Jackson et al., *ACM J. Emerg. Tech. Comput. Syst.*, **9**(2), 12 (2013).
[2] M. Suri et al., *IEDM Tech. Digest*, 4.4 (2011).
[3] D. Rumelhart et al., *Parallel Distributed Processing*, MIT Press (1986).
[4] Y. LeCun et al., *Proc. IEEE*, **86**(11), 2278 (1998).
[5] G. Hinton et al., *Science*, **313**(5786) 504 (2006).
[6] M. Breitwisch et al., *VLSI Tech. Symp.*, T6B-3 (2007).
[7] B. Rajendran et al., *IEEE Trans. Elect. Dev.*, **60**(1), 246 (2013).

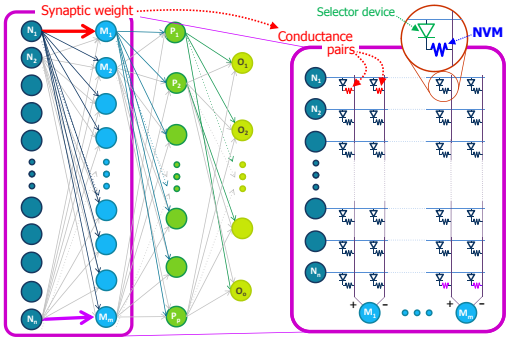

Fig. 1 Neuro-inspired non-Von Neumann computing [1,2], in which neurons activate each other through dense networks of programmable synaptic weights, can be implemented using dense crossbar arrays of non-volatile memory (NVM) and selector device-pairs.

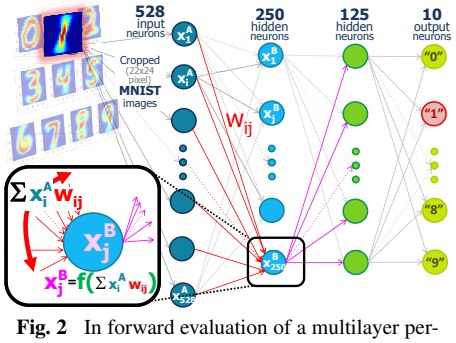

Fig. 2 In forward evaluation of a multilayer perceptron, each layer's neurons drive the next layer through weights w_{ij} and a nonlinearity $f()$. Input neurons are driven by pixels from successive MNIST images (cropped to 22×24); the 10 output neurons identify which digit was presented.

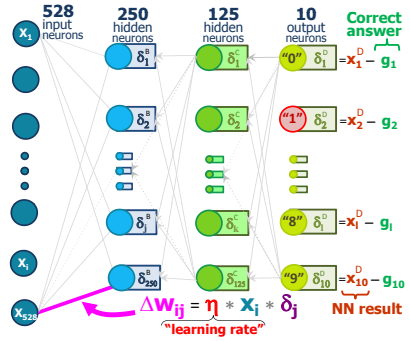

Fig. 3 In supervised learning, error terms δ_j are back-propagated, adjusting each weight w_{ij} to minimize an "energy" function by gradient descent, reducing classification error between computed (x_l^D) and desired output vectors (g_l).

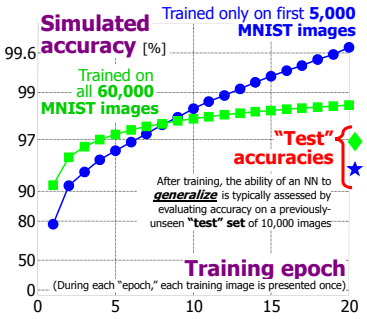

Fig. 4 A 3–layer perceptron network can classify previously unseen ('test') MNIST handwritten digits with up to ∼97% accuracy[4]. Training on a subset of the images sacrifices some generalization accuracy but speeds up training.

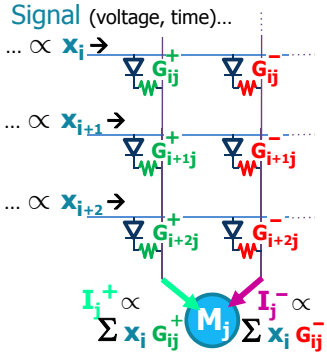

Fig. 5 By comparing total read signal between pairs of bitlines, summation of synaptic weights (encoded as conductance differences, $w_{ij} = G^+ - G^-$) is highly parallel.

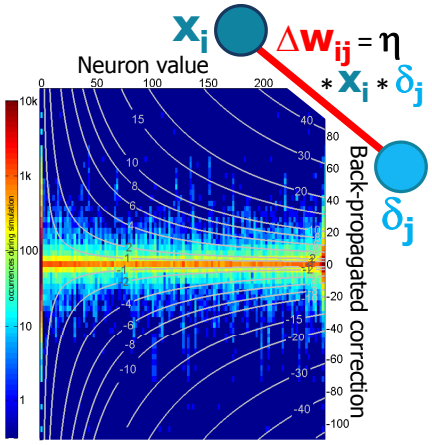

Fig. 6 Back-propagation calls for each weight to be updated by $\Delta w = \eta x_i w_{ij}$, where η is the *learning rate*. Colormap shows log(occurrences), in the 1^{st} layer, during NN training (blue curve, Fig. 4); white contours identify the quantized increase in the integer weight.

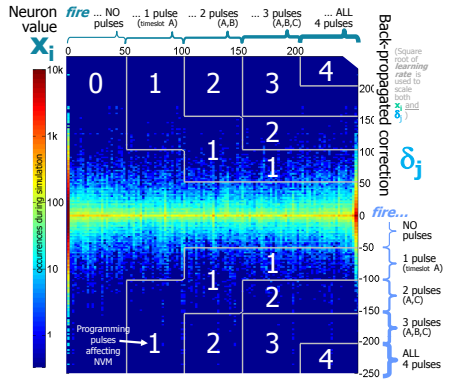

Fig. 7 In a crossbar, efficient learning requires neurons to update weights in parallel, firing pulses whose overlap at the various NVM devices implements training.

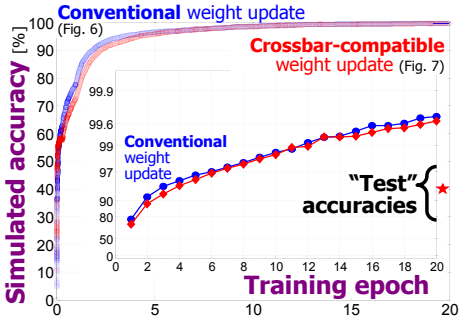

Fig. 8 Computer NN simulations show that a crossbar-compatible weight-update rule (Fig.7) is just as effective as the conventional update rule (Fig.6).

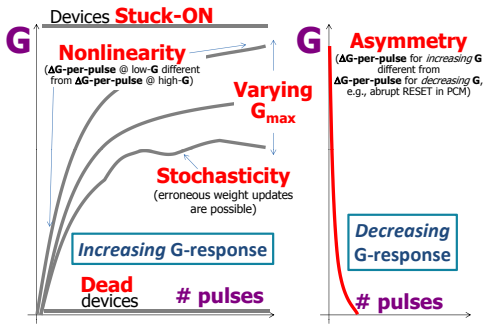

Fig. 9 The conductance response of an NVM device exhibits imperfections, including nonlinearity, stochasticity, varying maxima, asymmetry between increasing/decreasing responses, and non-responsive devices (at low or high G).

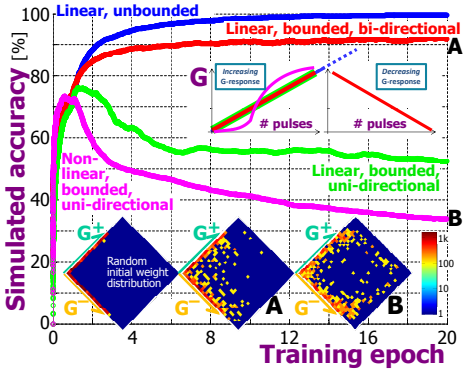

Fig. 10 Bounding G values reduces NN training accuracy slightly, but unidirectionality and nonlinearity in G-response strongly degrade accuracy. Figure insets map NVM-pair synapse states on a diamond-shaped plot of G^+ vs. G^- (weight is vertical position) for a sampled subset of the weights.

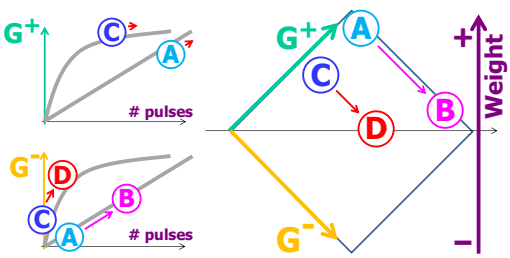

Fig. 11 If G values can only be increased (asymmetric G-response), a synapse at point **A** (G^+ saturated) can only increase G^-, leading to a low weight value (**B**). If response at small G values differs from that at large G (nonlinear G-response), alternating weight updates can no longer cancel. As synapses tend to get herded into the same portion of the G-diamond (**C** → **D**), the decrease in average weight can lead to network freeze-out.

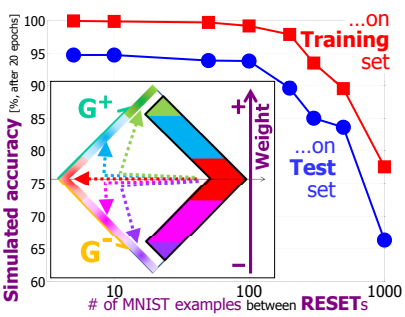

Fig. 12 Synapses with large conductance values (inset, right edge of G-diamond) can be refreshed (moved left) while preserving the weight (to some accuracy), by RESETs to both G followed by a partial SET of one. If such RESETs are too infrequent, weight evolution stagnates and NN accuracy degrades.

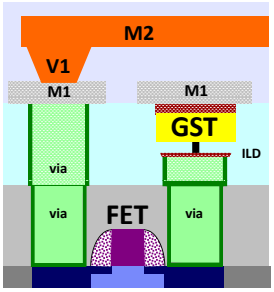

Fig. 13 Mushroom-cell [6], 1T1R PCM devices (180nm node) with 2 metal interconnect layers enable 512×1024 arrays.

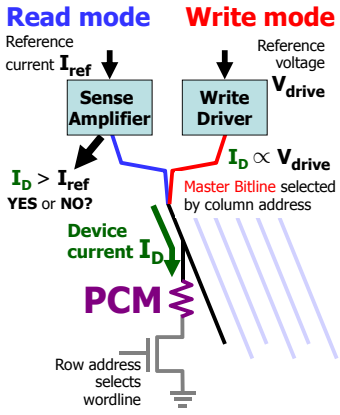

Fig. 14 A 1-bit sense amplifier measures G values, passing the data to software-based neurons. Conductances are increased by identical 25ns "partial–SET" pulses to increase G^+ (G^-) (Fig. 7), or by RESETs to both G followed by an iterative SET procedure (Fig. 12).

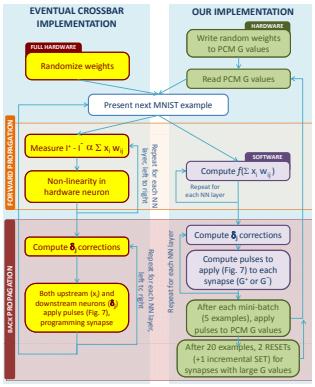

Fig. 15 Although G values are measured sequentially, weight summation and weight update procedures in our software-based neurons closely mimic the column (and row) integrations and pulse-overlap programming needed for parallel operations across a crossbar array. However, since occasional RESET is triggered when both G^+ and G^- are large, serial device access is required to obtain and then re-program individual conductances.

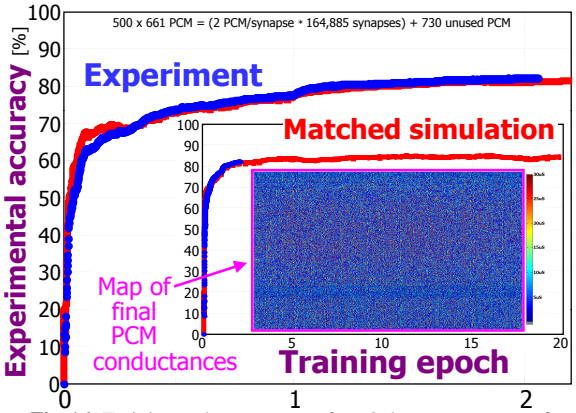

Fig. 16 Training and test accuracy for a 3–layer perceptron of 164,885 hardware-synapses, with all weight operations taking place on a 500 × 661 array of mushroom-cell [6] PCM devices (Fig. 13). Also shown is a matched computer simulation of this NN, using parameters extracted from the experiment.

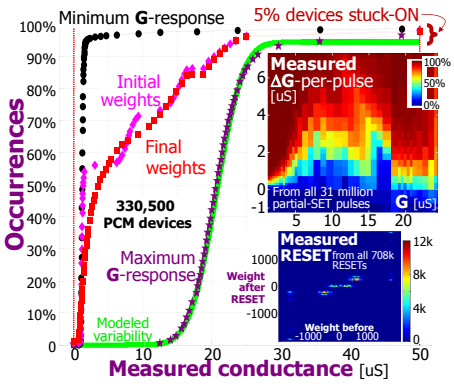

Fig. 17 50-point cumulative distributions of experimentally measured conductances for the 500 × 661 PCM array, showing variability and stuck-ON pixel rate. Insets shows the measured RESET accuracy, and the rate and stochasticity of G-response, plotted as a colormap of ΔG-per-pulse vs. G.

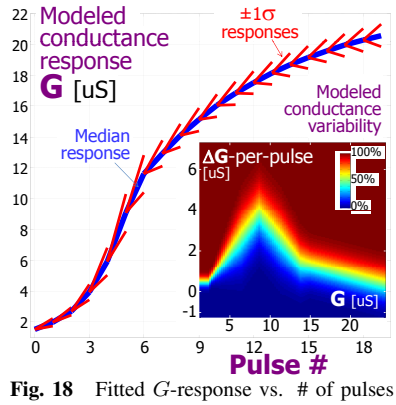

Fig. 18 Fitted G-response vs. # of pulses (blue average, red ±1σ responses), obtained from our computer model (inset) for the rate and stochasticity of G-response (ΔG-per-pulse vs. G) matched to experiment (Fig. 17).

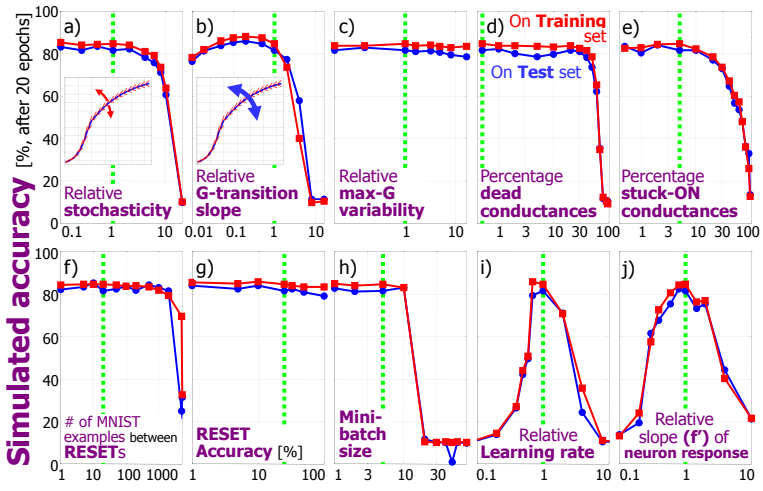

Fig. 19 Matched simulations show that an NVM-based NN is highly robust to stochasticity, variable maxima, the presence of non-responsive devices, and infrequent or inaccurate RESETs. Mini-batch size = 1 avoids the need to accumulate weight updates before applying them. However, the nonlinear and asymmetric G-response limits accuracy to ∼85%, and require learning-rate and neuron-response (f') to be precisely tuned.

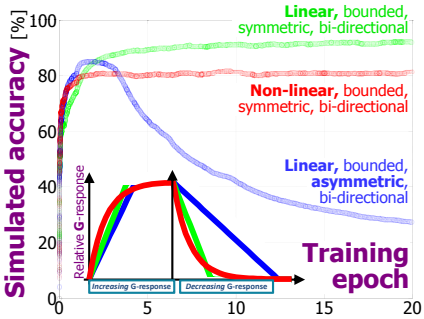

Fig. 20 NN performance is improved if G-response is linear and symmetric (green curve) rather than nonlinear (red). However, asymmetry between the up- and down-going G-responses (blue), if not corrected in the weight-update rule (Fig. 7), can strongly degrade performance by favoring particular regions of the G-diamond (Figs. 10,11).

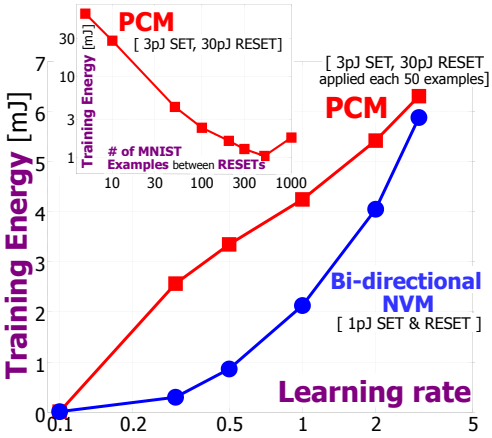

Fig. 21 Despite the higher power involved in RESET rather than partial-SET (30pJ and 3pJ for highly-scaled PCM [1]), total energy costs of training can be minimized if RESETs are sufficiently infrequent (inset). Low-energy training requires low learning rates, which minimizes the number of synaptic programming pulses. At higher learning rates, even a bi-directional, linear RRAM requiring no RESET and offering low-power (1pJ per pulse) can lead to large training energy.

Table of conclusions

Large 3-layer network with 2 PCM devices/synapse. (Figs. 1-3, 5) Back-propagation weight update rule compatible with crossbar array. (Figs. 6-8)	Moderately high accuracy (82%) achieved on MNIST handwritten digit recognition with two training epochs. (Fig. 16)
NVM models identified issues for training: Conductance bounds, nonlinearity, and asymmetry must be considered. (Figs. 9-10, 20)	PCM response and asymmetry mitigated by RESET strategy, mapping of response, choice of update pulse. (Figs. 11, 12, 17)
Model of PCM allows well-matched simulation of experiment (Figs. 16-18), variation of network parameters allows tolerancing. (Fig. 19)	NN is resilient to NVM variations (Figs. 19a-e) and RESET strategy (Figs. 19f-g), but sensitive to learning rate and neuron response function. (Figs. 19i-j)
Bidirectional NVM with no special RESET strategy and good performance requires scheme for symmetric response. (Fig. 20)	For PCM, keeping RESET frequency down and learning rate above "freeze-out" threshold allows reasonable training energy. (Fig. 21)

Fig. 22 NN built with NVM-based synapses tend to be highly sensitive to "gradient" effects (nonlinearity and asymmetry in G-response) that "steer" all synaptic weights towards either high or low values, yet are highly resilient to random effects (NVM variability, yield, and stochasticity).

References

[1] B. Jackson et al., *ACM J. Emerg. Tech. Comput. Syst.*, **9**(2), 12 (2013).

[2] M. Suri et al., *IEDM Tech. Digest*, 4.4 (2011).

[3] D. Rumelhart et al., *Parallel Distributed Processing*, MIT Press (1986).

[4] Y. LeCun et al., *Proc. IEEE*, **86**(11), 2278 (1998).

[5] G. Hinton et al., *Science*, **313**(5786) 504 (2006).

[6] M. Breitwisch et al., *VLSI Tech. Symp.*, T6B-3 (2007).

[7] B. Rajendran et al., *IEEE Trans. Elect. Dev.*, **60**(1), 246 (2013).