31.2 A 42pJ/Decision 3.12TOPS/W Robust In-Memory Machine Learning Classifier with On-Chip Training

Sujan Kumar Gonugondla, Mingu Kang, Naresh Shanbhag

University of Illinois at Urbana-Champaign, IL

Embedded sensory systems (Fig. 31.2.1) continuously acquire and process data for inference and decision-making purposes under stringent energy constraints. These *always-ON* systems need to track changing data statistics and environmental conditions, such as temperature, with minimal energy consumption. Digital inference architectures [1,2] are not well-suited for such energy-constrained sensory systems due to their high energy consumption, which is dominated (>75%) by the energy cost of memory read accesses and digital computations. In-memory architectures [3,4] significantly reduce the energy cost by embedding pitch-matched analog computations in the periphery of the SRAM bitcell array (BCA). However, their analog nature combined with stringent area constraints makes these architectures susceptible to process, voltage, and temperature (PVT) variation. Previously, off-chip training [4] has been shown to be effective in compensating for PVT variations of in-memory architectures. However, PVT variations are die-specific and data statistics in *always-ON* sensory systems can change over time. Thus, on-chip training is critical to address both sources of variation and to enable the design of energy efficient *always-ON* sensory systems based on in-memory architectures. The stochastic gradient descent (SGD) algorithm is widely used to train machine learning algorithms such as support vector machines (SVMs), deep neural networks (DNNs) and others. This paper demonstrates the use of on-chip SGD-based training to compensate for PVT and data statistics variation to design a robust in-memory SVM classifier.

Figure 31.2.2 shows the system architecture with an analog in-memory (IMCORE) block, a digital trainer, a control block (CTRL) for timing and mode selection, and a normal SRAM R/W interface. The system can operate in three modes: as a conventional SRAM, for in-memory inference, and in training mode. IMCORE comprises of a conventional 512×256 6T SRAM BCA and in-memory computation circuitry: 1) pulse width modulated (PWM) WL drivers to realize functional read (FR), 2) BL processors (BLPs) implementing signed multiplication, 3) cross BLP (CBLP) implementing summation, and 4) an A2D converter and a comparator bank to generate final decisions. While the IMCORE implements feedforward computations of the SVM algorithm, the trainer implements a batch mode SGD algorithm (update equations shown in Fig. 31.2.2) to train the SVM weights $\boldsymbol{W}$ stored in the BCA. The input vectors $\boldsymbol{X}$ are streamed into the input buffers in the trainer. A gradient estimate (Δ) is accumulated for each input, based on the label y_n and outputs $\delta_{1,n}$ *and* $\delta_{-1,n}$ of IMCORE. At the end of each batch, the accumulated gradient estimate (Δ) is used to update the weights in BCA via the SRAM's conventional R/W interface. While 16b weights are used in the trainer during the weight update, only 8b weights are used for feedforward/inference. The learning rate (γ) and the regularization factor (α) can be reconfigured in powers of 2.

During feedforward computations, $\boldsymbol{W}$ is read in the analog domain on the BLs and the input vectors ($\boldsymbol{X}$) are transferred to the BLP via a 256b bus. The mixed-signal capacitive multiplier in the BLP realizes multiplication via sequential charge sharing, similar to the one introduced in [3]. Based on the sign of the weights, the multiplier outputs are charge shared either on the positive or on the negative CBLP rails across the BLs. The voltage difference of the negative and positive rails is proportional to the dot product $\boldsymbol{W^TX}$. The rail values are either sampled and converted to a digital value by an ADC pair, or a decision is obtained directly via a comparator bank. Three comparators are used, where one generates the decision ($\hat{y}$) while the other two comparators implement a SVM margin detector that triggers a gradient estimate update.

Functional read (Fig. 31.2.3) uses 4-parallel pulse-width and amplitude-modulated (PWAM) WL enable signals resulting in the BL discharge ΔV_{BL} (or ΔV_{BLB}) proportional to the 4b W_i's, stored in a column-major format (Fig 31.2.3), in one precharge cycle. The BL discharges (V_{BL}) proportional to 4b words read in the adjacent BLs are combined in a 1:16 ratio to realize an 8b read out. This enables 128-dimensional 8b vector processing per access. The weights are represented in 2's complement. A comparator detects the sign of W_i, which is then used to select its magnitude, both of which are passed on to the signed multipliers. Spatial variations impacting ΔV_{BL} is measured across 30 randomly chosen 4-row groups. When the maximum ΔV_{BL} ($\Delta V_{BL,\max}$), corresponding to 4b $W_i = 15$, is set to 320mV the maximum variation in ΔV_{BL} $((\sigma/\mu)_{\max})$, across all 16 values, is found to be 16% vs. 7% at $\Delta V_{BL,\max}$ = 560mV. This increase in variation leads to an increase in the misclassification rate: from 4% to 18%.

The MIT CBCL face detection data set is used for testing: it consists of 4000 training images and 858 test images. During training, input vectors are randomly sampled, with replacement, from the training set. At the end of each batch, the classifier is tested on the test set to obtain the misclassification (error) rate. Figure 31.2.4 shows the benefits of on-chip learning to overcoming process and data variations, and the need for learning chip-specific weights. Beginning with random initial weights and a $\Delta V_{BL,\max}$ = 560mV, the learning curves converge to within 1% of floating-point accuracy in 400 batch updates for learning rates $\gamma \geq 2^{-4}$. The misclassification rate increases dramatically to 18%, when $\Delta V_{BL,\max}$ is reduced to 320mV, at batch number 400 due to the increased impact of process variations during FR. Continued on-chip learning reduces this misclassification rate down to 8% for $\gamma \geq 2^{-4}$. Similar results are observed when the illumination changes abruptly at batch number 400 indicating robustness to variation in data statistics. The table in Fig. 31.2.4 shows the misclassification rate measured across 5 chips when the weights are trained on one chip and used in others. The use of chip-specific weights (diagonal) results in an average misclassification rate of 8.4% vs. 43% when they are not, highlighting the need for on-chip learning.

Figure 31.2.5 shows the trade-off between the misclassification rate, IMCORE energy, and $\Delta V_{BL,\max}$. On-chip training enables the IC to achieve a misclassification rate below 8% at a 38% lower $\Delta V_{BL,\max}$ (320mV) and a lower IMCORE supply $V_{DD,\text{IMCORE}}$ (0.675V), compared to the use of weights obtained at $\Delta V_{BL,\max}$ of 560mV and a $V_{DD,\text{IMCORE}}$ of 0.925V. Thus, the IMCORE energy is reduced by 2.4× without any loss in accuracy. The energy cost of training is dominated by SRAM writes of updated weights done once per batch. This cost reduces with the batch size (N) reaching 26% of the total energy cost, for a batch size of 128. At this batch size, 60% of the total energy is due to the control block; this energy overhead reduces with increasing SRAM size.

Figure 31.2.6 shows an IMCORE energy efficiency of 42pJ/decision at a throughput of 32M decisions/s, which corresponds to a computational energy efficiency of 3.12TOPS/W (1OP is a single 8b×8b MAC operation). This work achieves the lowest reported precision-scaled MAC energy, as well as the lowest reported MAC energy when SRAM memory access costs are included. Energy consumption of a digital architecture [1,2] to realize the 128-dimensional SVM algorithm of this work is estimated from their MAC energy, which shows a savings of greater than 7×, thereby demonstrating the suitability of this work for energy-constrained sensory applications.

The die micrograph of the 65nm CMOS IC and performance summary is shown in Fig. 31.2.7.

Acknowledgements:
This work was supported in part by Systems On Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA. The authors would like to acknowledge constructive discussions with Professors Pavan Hanumolu, Naveen Verma, Boris Murmann, and David Blaauw.

References:

[1] Y.H. Chen, et al., "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *ISSCC*, pp. 262-263, 2016.
[2] P.N. Whatmough, et al., "A 28nm SoC with a 1.2GHz 568nJ/prediction sparse deep-neural-network engine with >0.1 timing error rate tolerance for IoT applications," *ISSCC*, pp. 242-243, 2017.
[3] M. Kang, et al., "A 481pJ/decision 3.4 M decision/s multifunctional deep in-memory inference processor using standard 6T SRAM array," *arXiv:1610.07501*, 2016, https://arxiv.org/abs/1610.07501.
[4] J. Zhang, et al., "In-memory computation of a machine learning classifier in a standard 6T SRAM array," *JSSC,* vol. 52, no. 4, pp. 915-924, April 2017.
[5] E.H. Lee, et al., "A 2.5GHz 7.7TOPS/W switched-capacitor matrix multiplier with co-designed local memory in 40nm," *ISSCC*, pp. 418-419, 2016.
[6] S. Joshi, et al., "2pJ/MAC 14b 8×8 linear transform mixed-signal spatial filter in 65nm CMOS with 84dB interference suppression," *ISSCC*, pp. 364-365, 2017.

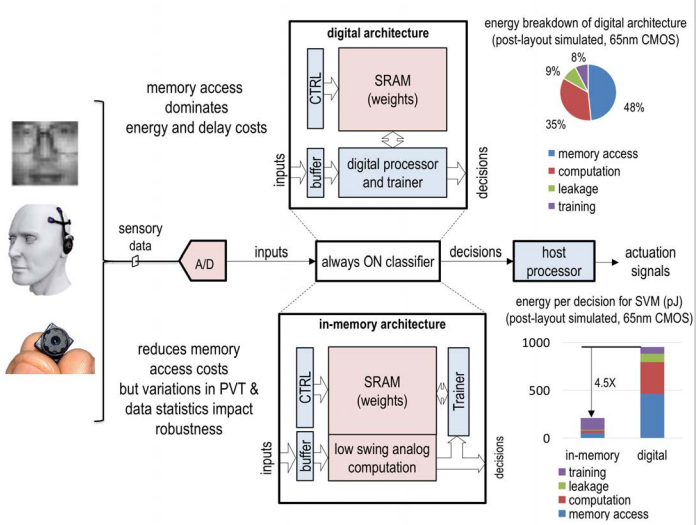


Figure 31.2.1: An SGD-based on-chip learning system for robust energy efficient always-ON classifiers.

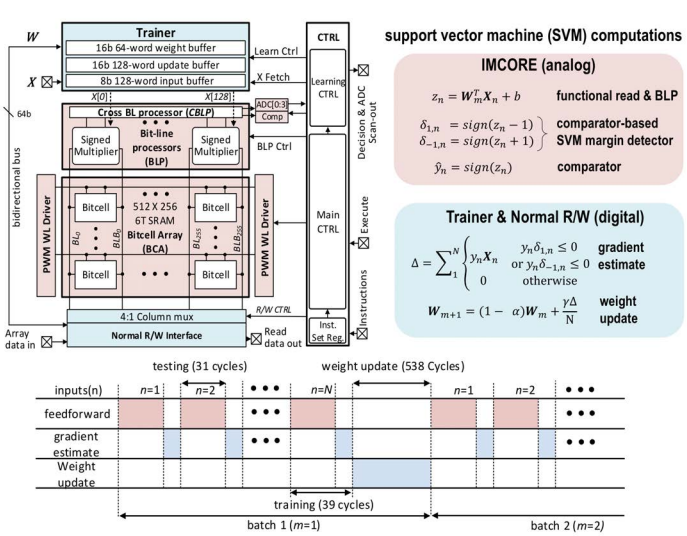


Figure 31.2.2: Proposed SGD-based in-memory classifier architecture.

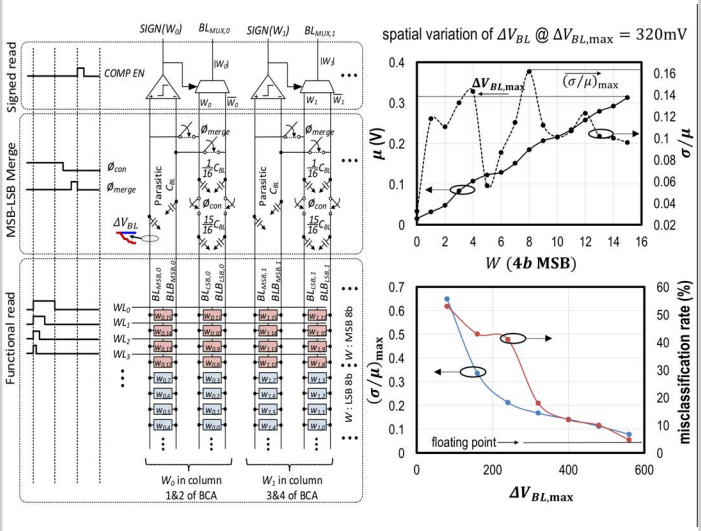


Figure 31.2.3: In-memory functional read, measured spatial variations on BL swing, and its impact on the measured SVM misclassification rate.

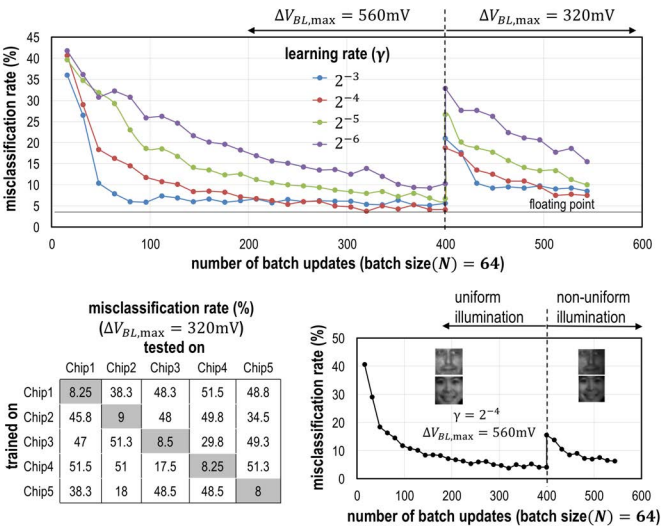


Figure 31.2.4: Measured robustness to spatial variations and non-stationary data.

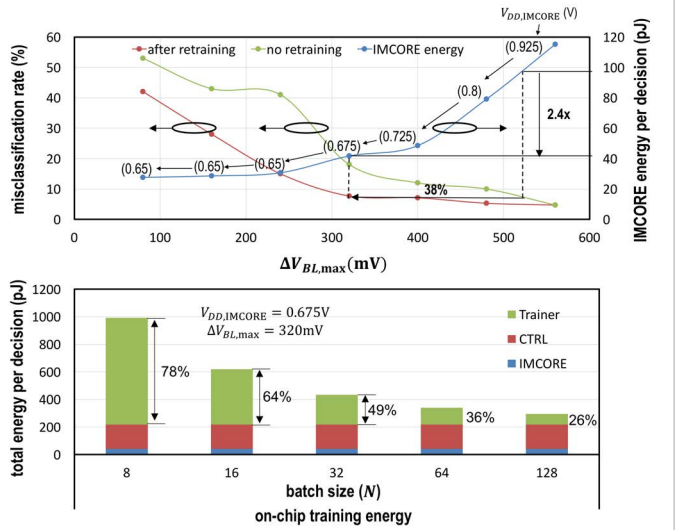
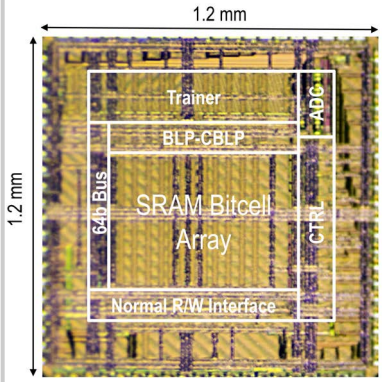


Figure 31.2.5: Measured energy via supply voltage and BL swing scaling. Energy cost of training.

	[1]	[2]	[5]	[6]	[3]	[4]	this work
Technology	65nm	28nm HPC	40nm	65nm	65nm	180nm	65nm
Algorithm	CNN	FC-DNN	matrix mult.	filtering	SVM	AdaBoost	SVM
Data set	ImageNet	MNIST			MIT-CBCL	MNIST	MIT-CBCL
Architecture	digital	digital	analog	analog	in-memory	in-memory	in-memory
On-chip learning	No	No	No	No	No	No	Yes
Total SRAM size (kb)	1449.2	9248	–	–	128	103.6	128
Energy/Decision	7.94mJ[d]	0.56uJ	–	–	0.4nJ	0.6nJ	0.042nJ
Decisions/s	35	28.8k[d]	–	–	9.2M	7.9M	32M
# of MACs/Decision	2663M	334k	–	–	512	–	128
Max. accuracy (%)	–	98	–	–	96	91	96
MAC level metrics							
MAC precision[a] ($B_x \times B_w$)	$16^s \times 16^s$	$8^s \times 8^s$	$3^s \times 6^s$	8×14^s	8×8	5×1	8×8^s
Efficiency (TOPS/W)	0.336[d]	0.56[d]	3.84[b]	0.5[b]	1.25	–	**3.125**
MAC energy (E_{MAC}) (pJ)	2.98[d]	1.79[d]	0.26[b]	2[b]	0.8	–	**0.32**
precision-scaled MAC energy[c] (fJ)	11.6	28	14.4[b]	17.857[b]	12.5	–	**4.9**
Estimated performance of prior art to realize SVM algorithm with vector dimension of 128							
Energy/Decision (nJ)	0.381	0.229	0.033[b]	0.256[b]	0.102	–	**0.042**
Decisions/s	250M	75M	19.5M	350k	36.8M	–	32M
# MACs per cycle	168	8	1	64	256	10,368	128

[a] s indicates signed; B_x: input precision; B_w: weight precision
[b] does not include SRAM memory access
[c] normalized to account for operand precision ($E_{MAC}/(B_x \times B_w)$)
[d] estimated from reported data

Figure 31.2.6: Comparison table.



Technology	65nm CMOS	
Die size	1.2 mm × 1.2mm	
Memory capacity	16KB (512 × 256)	
Nominal supply	1.0 V	
CTRL operating frequency	1 GHz	
Energy per decision (nJ)	Test	0.21
	Training	0.34
Average throughput (decision/s)	Test	32.3 M
	Training	21 M

Figure 31.2.7: Die micrograph and chip summary.

978-1-5090-4940-0/18/$31.00 ©2018 IEEE