A 1 TOPS/W Analog Deep Machine-Learning Engine With Floating-Gate Storage in 0.13 μm CMOS

Junjie Lu, *Student Member, IEEE*, Steven Young, *Student Member, IEEE*, Itamar Arel, *Senior Member, IEEE*, and Jeremy Holleman, *Member, IEEE*

***Abstract*—An analog implementation of a deep machine-learning system for efficient feature extraction is presented in this work. It features online unsupervised trainability and non-volatile floating-gate analog storage. It utilizes a massively parallel reconfigurable current-mode analog architecture to realize efficient computation, and leverages algorithm-level feedback to provide robustness to circuit imperfections in analog signal processing. A 3-layer, 7-node analog deep machine-learning engine was fabricated in a 0.13 μm standard CMOS process, occupying 0.36 mm^2 active area. At a processing speed of 8300 input vectors per second, it consumes 11.4 μW from the 3 V supply, achieving 1×10^{12} operation per second per Watt of peak energy efficiency. Measurement demonstrates real-time cluster analysis, and feature extraction for pattern recognition with 8-fold dimension reduction with an accuracy comparable to the floating-point software simulation baseline.**

***Index Terms*—Analog signal processing, current mode arithmetic, deep machine learning, floating gate, neuromorphic engineering, translinear circuits.**

I. Introduction

MACHINE-LEARNING systems provide automated data processing and see a wide range of applications from computer vision, data mining, natural language processing, to economics and biology [1]. When a machine learning system is used to process high-dimensional data such as raw images and videos, a difficulty known as the "curse of dimensionality" [2] arises. It stems from the fact that as the dimensionality increases, the volume of the input space increases exponentially. In order to maintain the same predictive power, a machine learning system requires exponentially larger training data set and computational power. Therefore, when dealing with such high dimensional data, it is often necessary to pre-process the data to reduce its dimensionality to what can be efficiently processed, while still preserving its essence, a technique known as feature extraction. Deep machine learning (DML) architectures have recently emerged as a promising bio-inspired framework, which mimics the hierarchical presentation of information in the human brain to achieve robust automated feature extraction [3].

While these deep layered architectures offer excellent performance attributes, the computation requirements involved grow dramatically as the dimensionality of input increases. GPU-based platforms have been proposed to provide the required parallel computation [4], but they are prohibitively power hungry, making them impractical in power-constrained environments and limiting their large-scale implementations. Custom analog circuitry presents a means of overcoming this limitation. By exploiting the computational primitives inherent in the physics of the devices, and representing the information with multi-bit encoding, analog signal processing (ASP) systems have the potential to achieve much higher energy efficiency compared to their digital counterpart [5]. Therefore, analog and mixed-mode signal processing is widely employed in ultra-low-power circuits and systems such as vision processors [6], adaptive filters [7], and biomedical sensors [8]. In [9]–[11], analog circuits are embedded in digital systems to implement efficient non-linear functions. The other advantage of ASP is that it interfaces directly with sensor output. By performing pre-processing and compression of the sensory data at the front-end, the accuracy and bandwidth requirement of subsequent blocks can be relaxed, increasing the overall system efficiency [12].

ASP has been successfully applied to build machine learning systems and its building blocks [13]–[18]. But many of them do not have on-chip learning capability; therefore software emulation is needed to obtain the parameters which will then be programmed into the chip [13], [15], [16]. This limits the system to the specific task or dataset it was pre-programmed to process. An on-chip trainable machine learning system is described in [14]. It is based on supervised learning and relies on a human expert to label the input data for training. An unsupervised learning system that is able to learn from the data continuously without any external assistance is more desirable in many applications.

The other important component of a learning system is the memory, which stores the previous learned knowledge. Digital memory requires A/D/A conversions to interface with analog circuits, consuming area and power headroom [9]–[11], [14], especially in a system with distributed memories where the data converters cannot be shared. Capacitors can be used for analog storage [17], but require constant refreshing and are prone to

Manuscript received April 22, 2014; revised July 13, 2014; accepted August 20, 2014. Date of publication October 09, 2014; date of current version December 24, 2014. This paper was approved by Guest Editor Yogesh Ramadass. This work was supported in part by the National Science Foundation under Grant CCF-1218492, and by the Defense Advanced Research Projects Agency under contract #HR0011-13-2-0016. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of DARPA, the NSF, or the U.S. Government.

The authors are with the Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2014.2356197

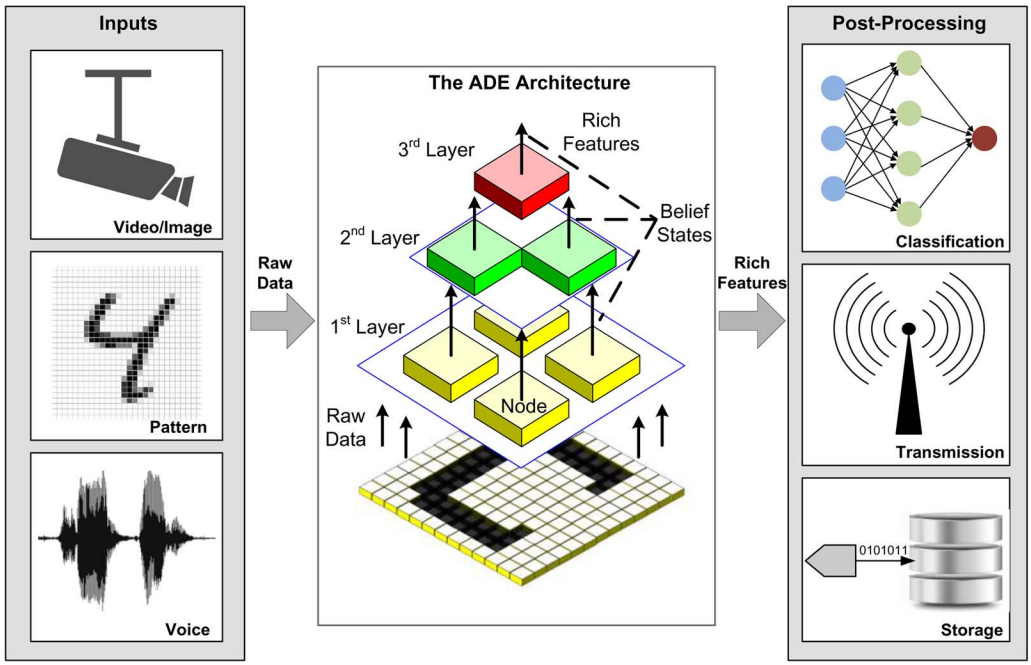


Fig. 1. Analog deep learning engine architecture and possible application scenarios.

long-term drift due to the leakage current, notably large in deep-sub-micron processes. In addition, both the digital and capacitor storage discussed above are volatile, which lose their states without power. This precludes their use in intermittently powered devices such as those depending on scavenged power, where blackout is common [19].

The purpose of this work is to develop an analog implementation of a deep machine-learning system [20]. It features unsupervised online trainability driven by the input data only. This ability to learn from the input data in real time without external intervention is essential for fully-autonomous systems. The proposed Analog Deep Learning Engine (ADE) utilizes floating-gate memory to provide non-volatile storage, facilitating operation with harvested energy. The memory has analog current output, interfacing naturally with the other components in the system, and is compatible with standard digital CMOS process. And the system architecture is designed for scaling. To maximize energy efficiency, several strategies are pursued at the system level. 1) The architecture adopts massively parallel computation, and the power-delay product is minimized by biasing transistors deep in weak inversion. 2) The feedback inherent in the learning algorithm is exploited to de-sensitize the system to inaccuracy such as mismatch, allowing aggressive area and bias current scaling-down with negligible performance penalty. 3) Current-mode circuits are extensively employed to realize efficient arithmetic. 4) Distributed memories are kept local to the computational elements, minimizing their access energy. 5) System power management applies power gating to the inactive circuits.

The rest of this paper is organized as follows: Section II presents the architecture of the system, as well as the algorithm it implements. Section III discusses the details of circuit implementation. The measurement results are reported in Section IV, and Section V concludes this paper.

II. Architecture and Algorithm

The analog deep learning engine (ADE) implements Deep Spatiotemporal Inference Network (DesTIN) [21], a state-of-the-art compositional DML framework, the architecture of which is shown in Fig. 1. Seven identical cortical circuits (*nodes*) form a 4-2-1 hierarchy. Each *node* captures the regularities in its inputs through an unsupervised learning process. The lowest layer receives the raw data (e.g. the pixels of an image), and continuously constructs *belief states* as its outputs to characterize the sequence observed. The inputs of *nodes* on the 2nd and 3rd layers are the outputs of the *nodes* on their respective lower layers. *Beliefs* extracted from the lower layers characterize local features and *beliefs* from higher layers characterize global features. From bottom to top, the abstraction level of information increases while the dimensionality of the data decreases. The *beliefs* formed at the top layer are then used as rich features with reduced dimensionality for post-processing.

The *node* learns through an online k-means clustering algorithm [22], which extracts the salient features of the inputs by recognizing spatial density patterns (clusters) in the input space. Each recognized cluster is represented in the circuit with a centroid, which is characterized by the estimated center of mass (centroid mean $\widehat{\mu}$) and spread (centroid variance $\widehat{\sigma}^2$). The architecture of the *node* is shown in Fig. 2(a). It incorporates an 8×4 array of reconfigurable analog computation cells (RAC), grouped into 4 centroids, each with 8-dimensional input. The centroids' parameters $\widehat{\mu}$ and $\widehat{\sigma}^2$ are stored in their respective floating gate memories (FGM). The input of the *node* is an 8-D observation vector sequence $o[n]$, presented row-parallel to the RAC array.

A training cycle begins with the *classification* phase (Fig. 2(b)). The analog arithmetic element (AAE) in the RAC

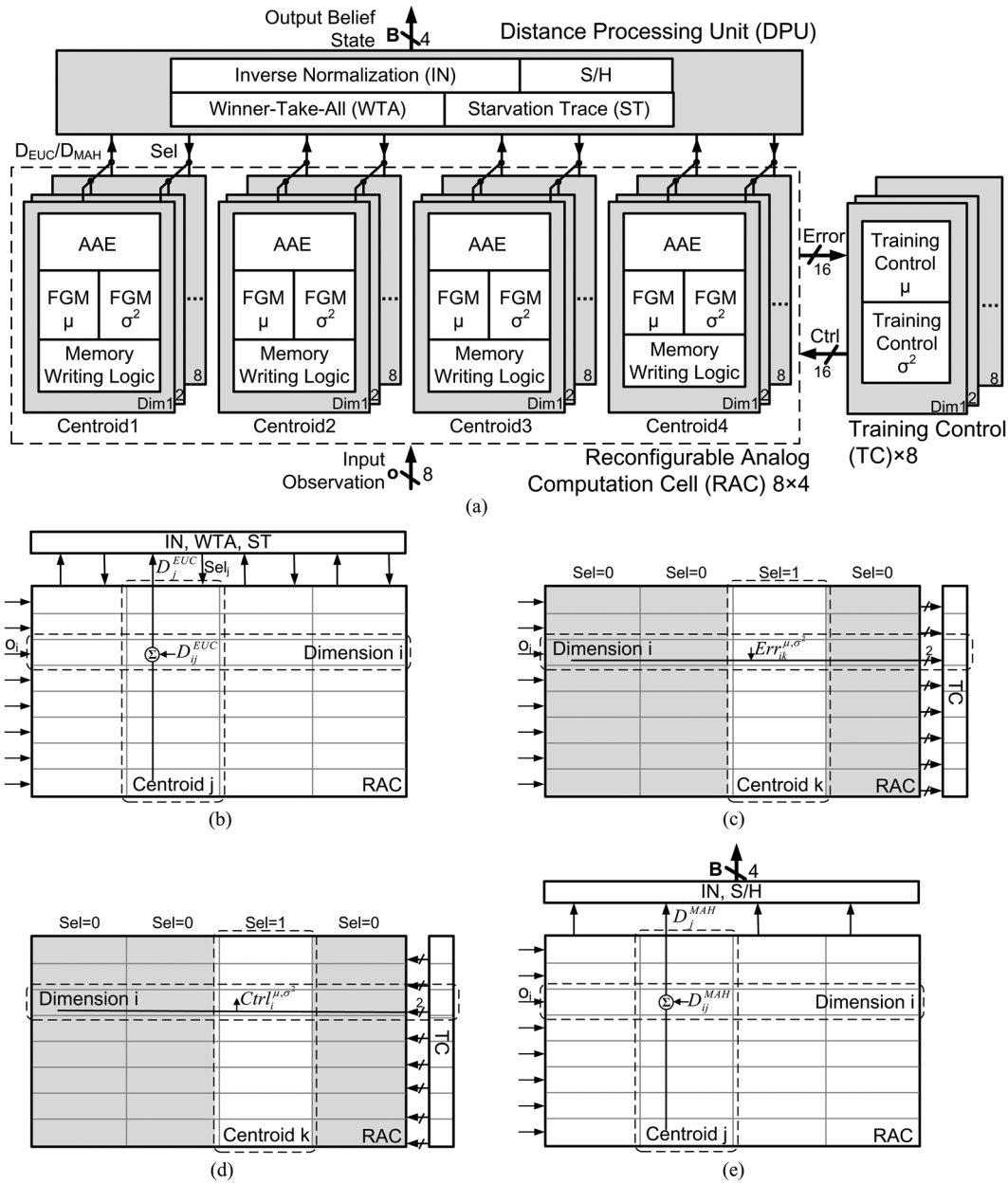


Fig. 2. (a) *Node* architecture. The clustering algorithm implemented by the *node* is illustrated in (b)–(e). In each cycle, the *node* goes through 4 phases: (b) classification, (c) training load, (d) memory write and (e) belief construction.

corresponding to the ith dimension of the centroid j calculates the 1-D squared Euclidean distance D_{ij}^{EUC} from its own centroid mean to the input element o_i. The Euclidean distance from each centroid to $\boldsymbol{o}$ in the 8-D space is obtained by wire-summing all the RAC output currents along the column:

$$\mathrm{D}_j^{EUC} = \sum_i D_{ij}^{EUC} = \sum_i (o_i - \hat{\mu}_{ij})^2. \tag{1}$$

Then a winner-take-all (WTA) network in the distance processing unit (DPU) searches for the best-matching centroid k with the minimum Euclidean distance to $\boldsymbol{o}$:

$$k = \arg\min_j \left(D_j^{EUC}\right) \tag{2}$$

and selects it by asserting Sel_k. For robust learning against unfavorable initial conditions, a starvation trace (ST) [23] circuit in the DPU monitors and corrects situations wherein some centroids are initialized too far away from populated regions of the inputs and never get selected, or "starved".

In the next phase (Fig. 2(c)), the selected centroid k propagates its mean and variance memory error vectors to the training control (TC) block. The ith elements of the 8-D error vectors are given by

$$\begin{aligned} Err_{ik}^{\mu} &= o_i - \hat{\mu}_{ik} \\ Err_{ik}^{\sigma^2} &= (o_i - \hat{\mu}_{ik})^2 - \hat{\sigma}_{ik}^2. \end{aligned} \tag{3}$$

The TC is shared across all centroids because only one centroid is selected for training each cycle. After the TC loads the

errors, it generates memory writing control signals ***Ctrl*** for both mean and variance memories in the selected centroid, respectively. ***Ctrl*** is broadcasted along the row, the memory writing logic ensures that only the memories in the centroid selected in the *classification* phase get updated (Fig. 2(d)). The magnitudes of update are proportional to the errors in (3):

$$\hat{\mu}_{ik}[n+1] = \hat{\mu}_{ik}[n] + \alpha Err_{ik}^{\mu}$$
$$\hat{\sigma}_{ik}^{2}[n+1] = \hat{\sigma}_{ik}^{2}[n] + \beta Err_{ik}^{\sigma^2} \quad (4)$$

where α and β are the learning rates. The proportional updates cause the centroid means and variances to follow exponential moving averages and converge to the true statistics of the data clusters. All the memories are written simultaneously. Finally, the 4-D *belief state* ***B*** is constructed, which represents the probability that the input vector belongs to each of the 4 centroids (Fig. 2(e)). Simplified 8-D squared Mahalanobis distances (assuming diagonal covariance matrix) from each centroid to the input are calculated in a way similar to (1):

$$\mathrm{D}_j^{MAH} = \sum_i D_{ij}^{MAH} = \sum_i \frac{(o_i - \hat{\mu}_{ij})^2}{\hat{\sigma}_{ij}^2}. \quad (5)$$

Compared to the Euclidean distance, the Mahalanobis distance is a better metric of statistical similarity in that it takes both the mean distance and spread of data into account. The inverse-normalization (IN) block in the DPU converts $\boldsymbol{D^{MAH}}$ to valid probability distribution ***B***, which satisfies:

$$B_j = \frac{\lambda}{D_j^{MAH}}$$
$$\sum_j B_j = 1 \quad (6)$$

where λ is the normalization constant. A sample and hold (S/H) holds ***B*** for the rest of the cycle to allow parallel operation across the hierarchy. After the training converges, the ADE can operate in recognition mode, in which the memory adaptation is disabled to save power and the ADE continuously extracts rich features from the input based on its previously learned model parameters.

Careful considerations at architecture and algorithm level facilitate scaling, and improve area and energy efficiency. First, each *node* is identical and operates in parallel and independent to each other, making it easy to scale up the system for deeper hierarchy and larger input dimensionality to solve more complex problems. Second, the DPU and TC are shared along the columns and rows, respectively, and kept peripheral to the computation array, so that their area and power scales up more slowly. Third, the similarity metrics used in the algorithm ($\mathrm{D^{EUC}}/\mathrm{D^{MAH}}$) allow easier scaling of input dimension. These distances are summed in current to accommodate a multivariate distribution: the increased current level reduces the time constant at the summing node, and all the 1-D elements are computed in parallel.

The ADE goes through four distinct operation phases in each cycle, and in each phase only a part of the system is active. Based on this observation, the circuits are partitioned into several power domains based on their functionality, and power gating is applied whenever possible to save biasing power. The resulting timing diagram of the flexible intra-cycle power gating is shown in Fig. 3. Measurement results show a reduction of power consumption by 22% in training mode and 37% in recognition mode due to this scheme.

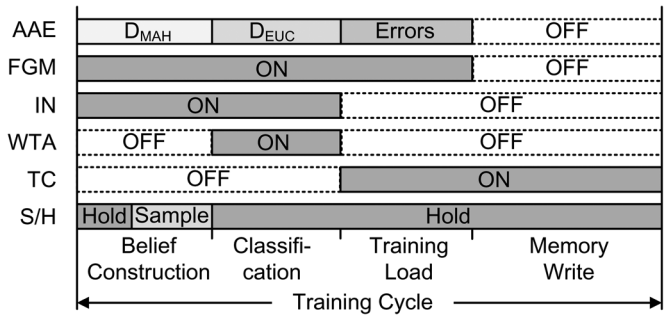


Fig. 3. Timing diagram of the intra-cycle power gating.

III. Circuit Implementation

A. Floating-Gate Analog Memory (FGM)

The FGM provides non-volatile storage for the centroid parameters. It can be accessed by on-chip circuits, as well as from off-chip through scanning registers for initialization. Its schematic is shown in Fig. 4(a). The negative feedback formed by the inverting amplifier M1/M2 and the feedback capacitor $\mathrm{C_F}$ keeps the floating gate voltage V_{FG} constant to achieve output-independent pulse-width controlled updates [7]. In the read mode, the source of $\mathrm{M_{INJ}}$ is at ground, so no channel current flows through it. The source of M1 is held at 3 V to keep V_{FG} at a relative high level, disabling tunneling. The write operation is controlled by pulses at the source of transistor $\mathrm{M_{INJ}}$ or M1. To achieve injection, 3 V pulses are applied to the source of $\mathrm{M_{INJ}}$. The holes at its drain end have enough kinetic energy to generate hot electrons, which are then injected onto the floating gate, reducing the memory output current. For tunneling, the source of M1 is lowered from 3 V to 1 V. This 2 V decrease causes a similar decrease in the floating gate voltage and thus an increase in the voltage across the gate oxide of $\mathrm{M_{TUN}}$, resulting in dramatic increase in tunneling current due to the steep relationship between gate current and oxide voltage in Fowler-Nordheim tunneling. Tunneling electrons off of the floating gate increases the memory output current. For both injecting and tunneling, the amount of charge added or removed from the floating gate depends solely on the pulse width applied, if neglecting 2nd order effects. This scheme allows random-accessible control of both tunneling and injection without high-voltage switches, charge pumps or complex routing, and is compatible with standard digital CMOS [24]. A two-transistor V-I converter, modified from that in [25], is employed to provide current output, and sigmoid update rules (Fig. 4(b)). It also reduces the swing at *Vout* to ensure the saturation of M2 with reduced supply during tunneling. Compared to a differential pair, this structure provides similar transfer function, while occupying less than 50% of area, and eliminates static bias current. The entire FGM consumes 0.5 nA of bias current, and shows 8-bit programming accuracy

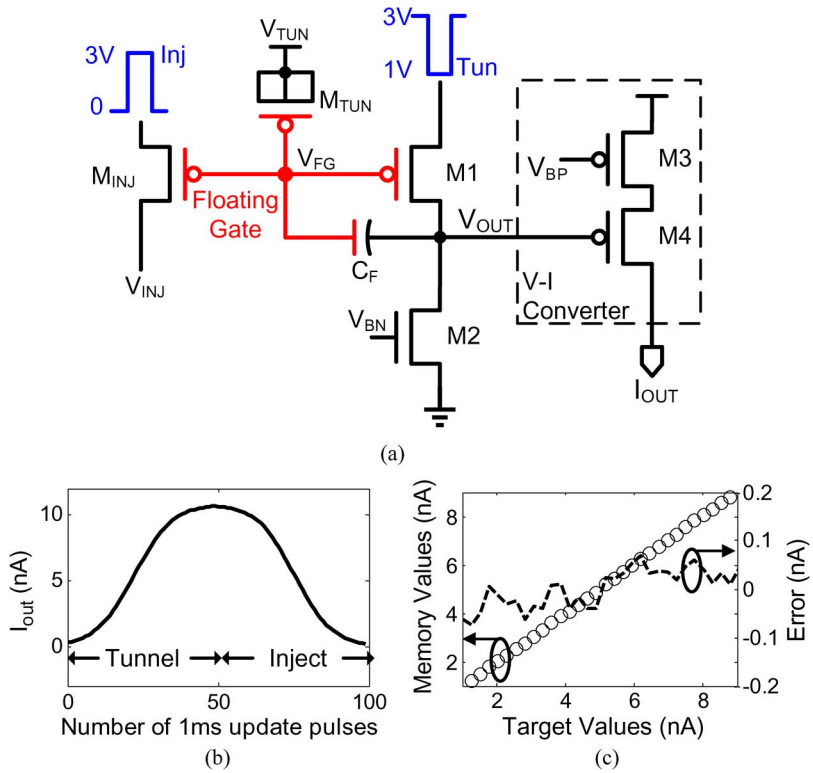


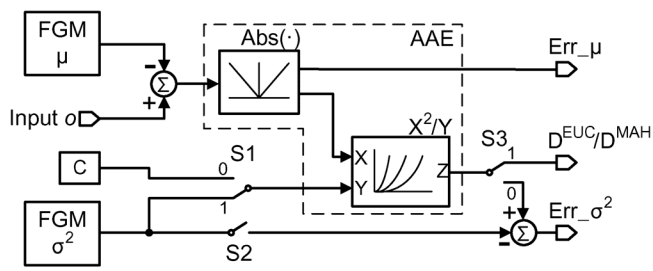
Fig. 4. (a) Schematic of the floating gate analog memory showing the update scheme. (b) The update rule measured by first ramping up then ramping down the memory with fixed pulse width. (c) Measured programming accuracy of 30 linearly spaced values. The standard deviation of the programming error is 40.2 pA.

(Fig. 4(c)), limited by testing equipment, and 46 dB SNR at full scale output.

B. Reconfigurable Analog Computation (RAC)

The RAC is the most computation-intensive block in the system and utilizes subthreshold current-mode computation to implement efficient arithmetic functions. It performs three different operations through reconfigurable current routing. The schematic and the current switch configurations for the three modes are shown in Fig. 5. The input current o and the centroid mean $\widehat{\mu}$ stored in the FGM_μ are added with opposite polarity and the difference current $o - \widehat{\mu}$ is rectified by the absolute value circuit Abs. The unidirectional output current is then fed into the X^2/Y operator circuit, where the Y component can be either the centroid variance memory output $\widehat{\sigma}^2$, or a constant C, depending on whether D^{MAH} or D^{EUC} is required. In *training load* phase, the Abs circuit duplicates its X input to get Err_μ, and the difference current between D^{EUC} and $\widehat{\sigma}^2$ forms Err_σ^2. The input o is used as the target value for mean memory training while the Euclidean distance is used as the target value for variance memory training because it has the same square error form as in (3). The reconfigurability of the RAC allows the computational circuits to be reused for different operations, therefore saving area and reducing the number of error sources in the circuit. In addition, use of the same circuit in the memory training and feature extraction tasks causes errors associated with these tasks to be correlated, which reduces the system's sensitivity to mismatch errors.

The schematic of the analog arithmetic element (AAE) is shown in Fig. 6(a). The absolute value circuit utilizes the



Phase	Switch Positions			Output
	S1	S2	S3	
Classification	0	Open	1	D^{EUC}
Training Load	0	Close	0	Err_μ,σ^2
Belief Construction	1	Open	1	D^{MAH}

Fig. 5. Schematic of the reconfigurable analog computation cell and the switch positions for three different operation modes.

class-B structure M1/M2 and the current mirror M3/M4 to rectify the bidirectional input current. Amplifier A reduces the input impedance by providing a virtual ground, allowing high-speed resolution of small current differences. The X^2/Y operator circuit employs translinear principle [26] to implement efficient current-mode squaring and division. The translinear loop formed by M5–8 (denoted by the arrow) gives

$$Z = I_{D8} = \frac{I_{D5} I_{D6}}{I_{D7}} = \frac{X^2}{Y} \tag{7}$$

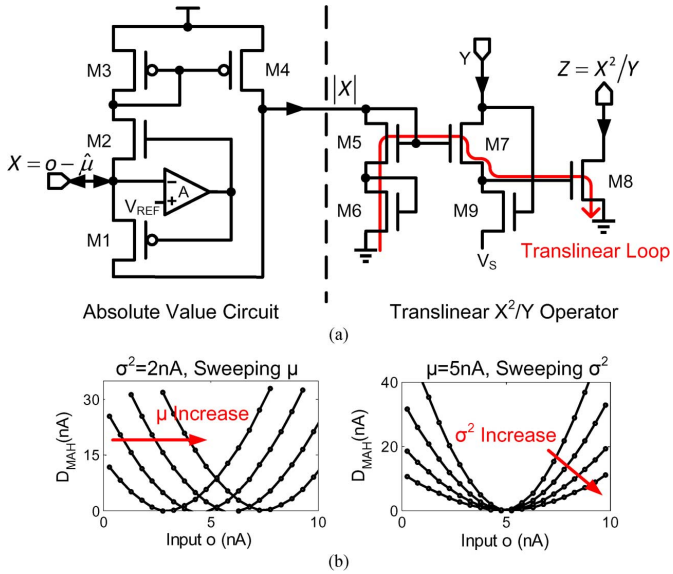


Fig. 6. (a) Schematic of the analog arithmetic element. (b) Measured transfer functions with the RAC configured to *belief construction* mode.

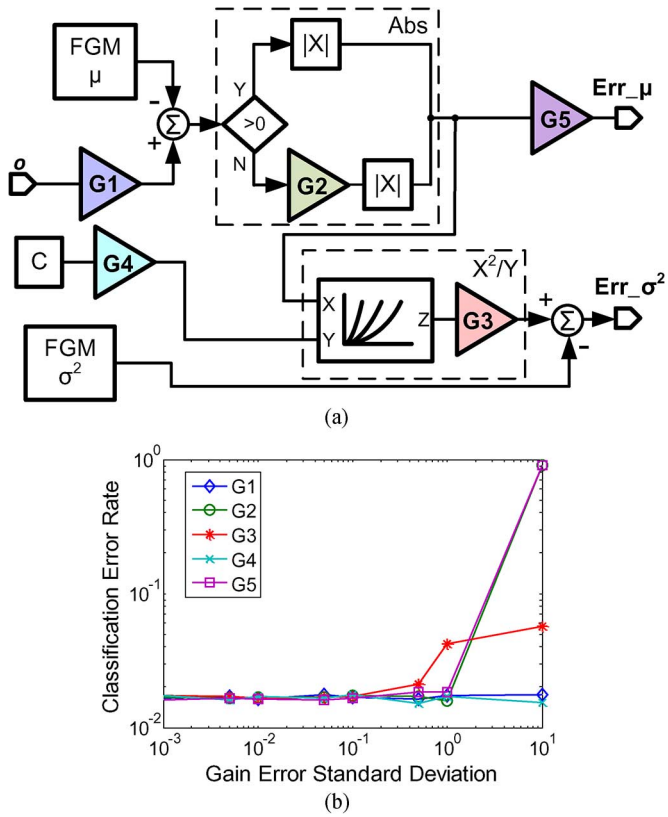


Fig. 7. (a) Behavioral model of the RAC with gain errors due to mismatch (*training load* mode is shown). (b) System's classification error rate as a function of each mismatch error.

assuming M5–8 are identical and biased in sub-threshold, and neglecting body effect. M9 modulates the source voltage of M7 to force current Y into its drain. The measurement results of the AAE are plotted in Fig. 6(b), showing variable center and width of the quadratic transfer functions by varying $\widehat{\mu}$ and $\widehat{\sigma}^2$.

ASP suffers from circuit imperfections such as noise and mismatch due to its lack of restoring mechanisms found in digital logic. Any ASP-based system needs to address these non-idealities, without excessively affecting the other performances metrics. The current noise power of transistors biased in sub-threshold is given by $2qI_D\Delta f$ [27], where Δf is the noise bandwidth, proportional to g_m of the transistors (the relative contribution of flicker noise is negligible at very low current level). As the g_m/I_D ratio is fairly flat in the subthreshold region, the computational throughput of a current-mode circuit biased in sub-threshold grows roughly linearly with the signal current level (or power consumption) while the system SNR remains nearly constant. Mismatch and efficiency place two contradictory requirements to the circuit design: device matching can be improved by increasing the areas of the devices [28], at the cost of both area and energy efficiency. Because computational throughput depends primarily on the ratio of transconductance to parasitic capacitance g_m/C_{GS}, and transconductance efficiency g_m/I_D is roughly geometry independent in the subthreshold region, therefore computational efficiency (operations/Joule) is roughly proportional to $g_m/(I_D C_{GS})$ and decreases as transistors are made larger. Fortunately, the learning algorithm used in this work provides robustness to mismatch by desensitizing the system to static errors using algorithm-level feedback [29]. To take full advantage of this robustness, a behavioral model of the RAC is built to include the mismatch error components found in the circuit. In weak inversion, the threshold voltage mismatch is the dominant source of mismatch, which manifests as gain errors in current-mode circuits. In the model shown in Fig. 7(a), each gain block Gx corresponds to the gain error introduced by a sub-circuit. System simulations were performed with progressively increasing gain errors to evaluate the effect of each of these errors on the ADE system performance. The results are plotted in Fig. 7(b). It can be seen that the system performance does not degrade until the errors are quite large, showing the robustness of the algorithm. The knowledge of the system sensitivities and tolerances allows aggressive reduction of the device sizes to place each gain error around its knee point of the performance curve in Fig. 7(b), maximizing efficiency with negligible performance penalty.

C. Distance Processing Unit (DPU)

The distance processing unit (DPU) performs various operations on the 8-D distance outputs from the four centroids. It has a modular design with four identical channels interconnected, one for each centroid. And it performs collective operations such as IN and WTA with a single communication wire along all the channels. Both facilitate scaling of the system to larger numbers of centroids. The simplified schematic of one channel is shown in Fig. 8. In *belief construction* phase, the IN block converts Mahalanobis distance $\boldsymbol{D^{MAH}}$ to *belief state* $\boldsymbol{B}$. The algorithm requires these two values to follow (6), as $\boldsymbol{B}$ represents collectively exhaustive probability measures of the input's similarity to each centroid. The translinear loop formed by M1 and M2 (denoted by the arrow) causes the product of the two drain currents to be a function of the difference between the voltage on the communication wire V_C and the bias voltage V_B, $I_{IN} \cdot I_{OUT} = f(V_C - V_B)$. Since all the channels share the same V_C and V_B, they all have: $I_{IN} \cdot I_{OUT} = \lambda$,

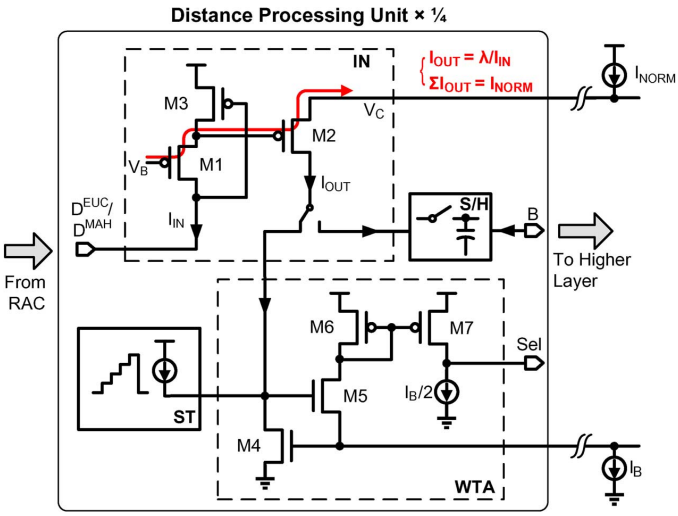


Fig. 8. Schematic of one channel of the distance processing unit.

where λ is equal across the four channels. In addition, the sum of the four output currents is dictated by the normalization current I_{NORM}, common to all the channels. Thus the inverse normalization function is implemented with only 3 transistors per channel without any additional biasing. The output *belief states* are sampled then held for the rest of the cycle to enable parallel operation of all the layers. The sampling of $\boldsymbol{B}$ starts from the top layer and propagates to the bottom, opposite to the data path; this pipelined processing eliminates the need to wait for the data to settle before sampling, improving the throughput. In *classification* phase, reconfigurable current routing allows the IN circuits to be reused together with the WTA to yield a loser-take-all function to find the centroid with the minimum Euclidean distance. The WTA (M4-M7) is based on the design in [30]. The voltage on the common wire is determined by the cell with the largest input current (winner). And the entire biasing current I_B will flow through M5/6 in the winner cell, making its output *Sel* high. A starvation trace (ST) circuit is implemented to inject current into the WTA input when the centroid is starved.

The schematic of the current mode sample and hold (S/H) is shown in Fig. 9(a). To maximize the power efficiency, the holding capacitor C_{HOLD} is realized entirely with the wiring parasitic capacitances between *nodes*. These wires are carefully laid-out to shield them from noisy signals, and a low-charge-injection switch is designed to mitigate the charge injection errors exacerbated by low-value C_{HOLD} and current-mode sub-threshold operation. During sample mode, S/H is low and the switch M3 is turned-on with near-minimum necessary V_{GS} to minimize its channel charge. This V_{GS} is generated by the diode-connected PMOS M1: body effect causes it to have slightly higher V_{TH} than M3, ensuring reliable turn-on in worst case mismatch situation. The post-layout simulation results are shown in Fig. 9(b). The S/H achieves less than 0.7 mV of charge injection error and less than 17 μV of droop across a cycle with about 80 fF C_{HOLD}.

The schematic of the starvation trace circuit (ST) is shown in Fig. 10, together with the typical timing diagram. C1, D1 and M1 form a charge pump, which removes a certain amount of charge from the storage capacitor C2 at every negative edge of

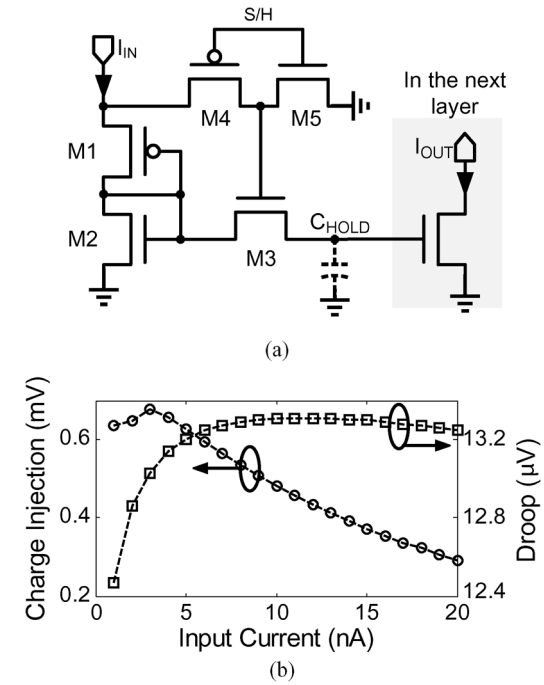


Fig. 9. (a) Schematic of the sample and hold and (b) simulated charge injection and droop errors.

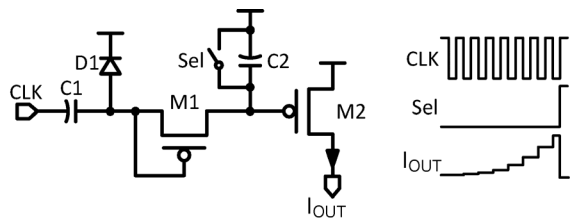


Fig. 10. Schematic and timing diagram of the starvation trace circuit.

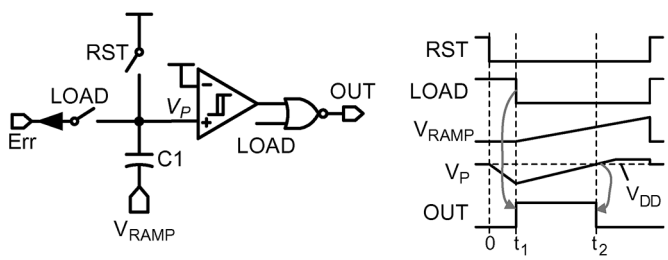


Fig. 11. Schematic and timing diagram of the training control circuit.

the system clock, increasing the output current I_{OUT}. Since the ST output adds to the inversed Euclidean distance, this current progressively decreases the apparent distance from the starved centroid to the input, forcing it to be selected for update and pulling it toward more populated areas of the input space. The ST current is reset once the centroid is selected (*Sel* = 1).

D. Training Control (TC)

The training control circuit converts the memory error current to pulse width to control the memory adaptation. For each dimension, two TC cells are implemented, one for mean and one for variance, shared across centroids. The schematic and timing diagram of one cell is shown in Fig. 11. The unidirectional error

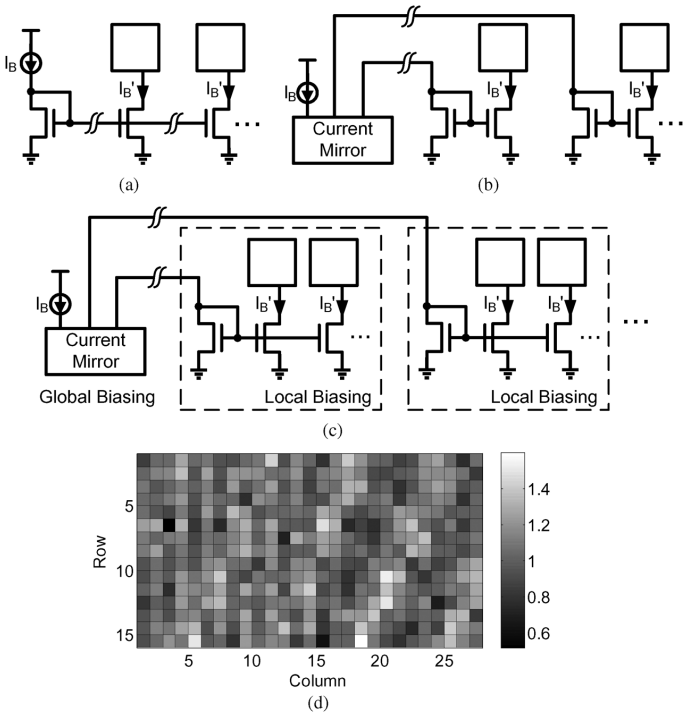
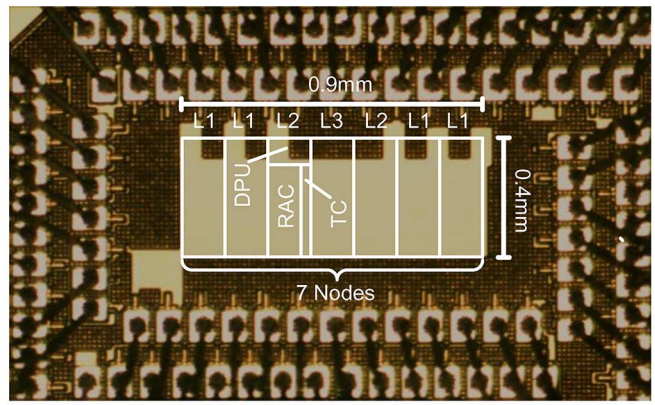


Fig. 12. (a) Voltage distribution of biasing. (b) Current distribution of biasing. (c) The hybrid biasing scheme used in this design. (d) Measured normalized mismatch of biasing.

current Err first discharges C1 from V_{DD} for a fixed time period of t_1. Then the non-inverting input of the comparator V_P is ramped up by the external voltage V_{RAMP} at the bottom plate of C1, until V_P gets back to V_{DD} at t_2. The update pulse is defined by $t_2 - t_1$, which is proportional to the input error current, so that the memory values converge to the input data statistics.

E. Biasing Design

Like other ASP systems, the ADE requires biasing in many blocks, for example, V_{BP} in the FGM sets the full scale output, and the amplifier in the Abs circuit requires tail current. Accurate and efficient distribution of biasing is important to the system's performance. A tight tolerance in biasing current allows less safety margin in the system design and operation, because the block with lowest biasing current is usually the performance bottleneck. Biasing can be distributed across the chip using voltage as in Fig. 12(a). However this scheme results in poor matching performance in large-scale systems due to process, stress and thermal gradients [28]. A current distribution scheme as in Fig. 12(b) achieves better immunity to gradients by keeping both sides of current mirror close, but consumes large biasing current and wiring overhead. The biasing scheme adopted in this design is a trade-off between the above two: current distribution is used for global biasing, and voltage distribution is used for local biasing, as shown in Fig. 12(c). The resulting biasing current accounts for only about 5% of the total current consumption, without observable gradient effects, shown in Fig. 12(d).



Block	RAC cell	DPU (per ch.)	TC cell
Transistor count	138	68	61

Fig. 13. Die micrograph of the analog deep learning engine and the transistor count of each block in the system.

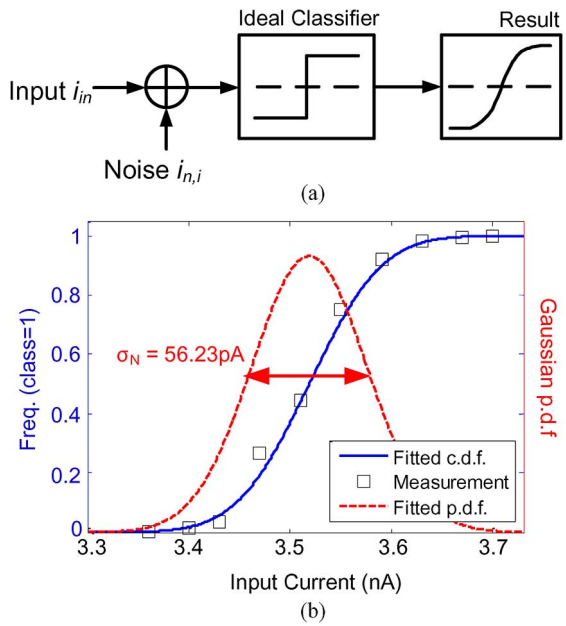


Fig. 14. (a) Model for input referred noise measurement. (b) Measured classification results and extracted Gaussian distribution.

IV. Measurement Results

The ADE was fabricated in a 0.13 μm standard CMOS process, and occupies an active area of 0.36 mm^2, including the biasing circuits and programming registers, shown in Fig. 13, together with the transistor count of each block in the system. Each RAC cell occupies 792 μm^2. Thick-oxide IO FETs are used to reduce charge leakage in the FGMs. With 3 V power supply, it consumes 27 μW in training mode, and 11.4 μW in recognition mode. To characterize the chip, a custom test board is developed with circuits to interface with the current-mode IOs. For practical use, the design is intended for system-on-chip applications where the input and output currents are generated and processed on-chip. The data is streamed between the chip and PC through data acquisition hardware, and the acquired data is post-processed on the PC.

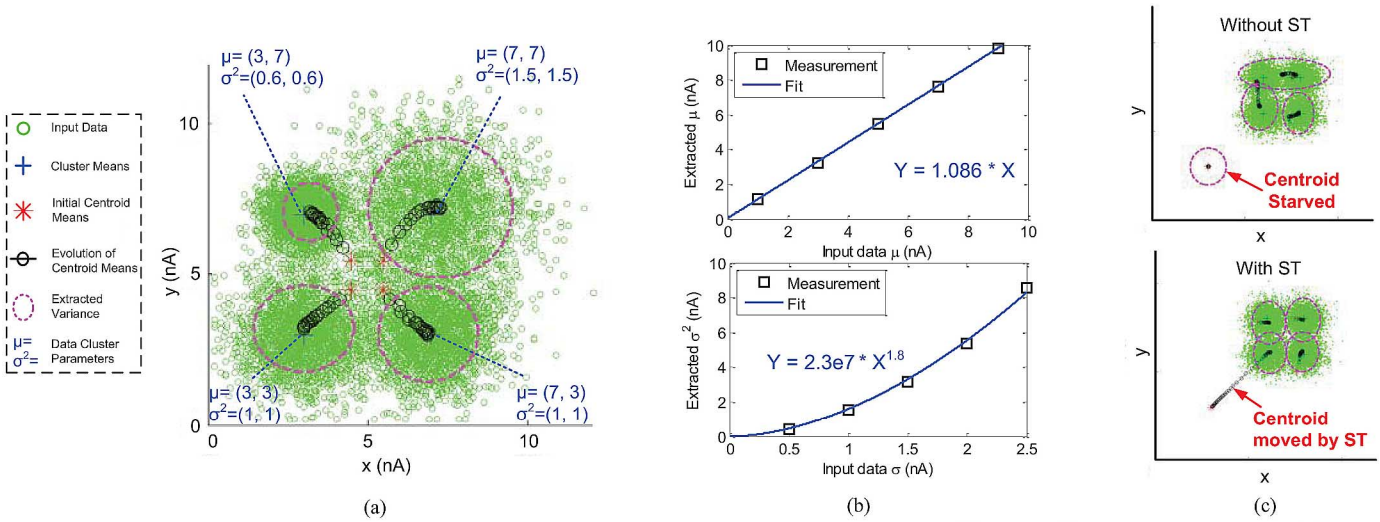


Fig. 15. (a) Clustering test results. (b) The extracted parameters are plotted versus their true values. The gain error in μ extraction is due to component mismatch; and the deviation of exponent from 2 in σ^2 extraction is due to body effect in the X^2/Y circuit; both can be tolerated by the algorithm. (c) Clustering results with bad initial condition without and with the starvation trace enabled.

A. Input Referred Noise

We use a statistical approach to measure the input referred noise of the non-linear ADE system. In the measurement, memory adaptation is disabled and the *node* is configured into a classifier, modeled as an ideal classifier with an input referred current noise (Fig. 14(a)). With two centroids competing, the circuit classifies the inputs to one centroid (class = 1) or the other (class = 0). When the inputs are close to the decision boundary and the classification is repeated for multiple times, the noise causes uncertainty in the outcome. Assuming additive Gaussian noise, it can be shown that the relative frequency of the event class = 1 approaches the cumulative density function (c.d.f.) of a normal distribution. The standard deviation σ_N of this distribution is extracted using curve fitting, shown in Fig. 14(b), and can be interpreted as the input-referred rms noise. The measured input-referred current noise is 56.23 pA_{rms} and with an input full scale of 10 nA, the system shows an SNR of 45 dB, or 7.5 bit resolution.

B. Clustering Test

The performance of the *node* is demonstrated with clustering tests. 40,000 8-D vectors are generated as the input dataset, consisting of four underlying clusters, each drawn from a Gaussian distribution with different mean and variance. The centroids are first initialized to separated means and a same variance (the initial condition is not critical since the circuit adaptively adjusts to the inputs). During the test, the centroid means are read out every 0.5 sec, plotted on top of the data scatter in Fig. 15(a), and shown together is the learned variance values at the end of test. For easier visual interpretation, 2-D results are shown. The extracted cluster means and variances from several tests are compared to the true values and show good agreement in Fig. 15(b). The performance of the starvation trace is verified by presenting the *node* with an ill-posed clustering problem. It can be seen that one of the centroids is initialized too far away from the input data, therefore never gets updated without the ST enabled. However, with the starvation trace enabled, the starved centroid is slowly pulled toward the area populated by the data, achieving a correct clustering result, shown in Fig. 15(c).

C. Feature Extraction Test

We demonstrate the full functionality of the chip by doing feature extraction for pattern recognition with the setup shown in Fig. 16(a). The input patterns are 16 × 16 symbol bitmaps corrupted by random pixel errors. An 8 × 4 moving window defines the pixels applied to the ADE's 32-D input. First the ADE is trained unsupervised with examples of patterns at 4.5 kHz. The training converges after about 30k samples (7 sec), as shown in Fig. 16(b). After the training converges, adaptation can be disabled and the circuit operates in recognition mode at 8.3 kHz. The 4 *belief states* from the top layer (shown in Fig. 16(c)) are used as rich features, achieving a dimension reduction from 32 to 4. A software neural network then classifies the reduced-dimension patterns. Three chips were tested and average recognition accuracies of 100% with pixel corruption level lower than 10%, and 94% with 20% corruption are obtained, which is comparable to the floating-point software baseline, as shown in Fig. 16(d), demonstrating robustness to the non-idealities of analog computation.

D. Performance Summary and Comparison

The measured performance of the ADE is summarized in Table I. It achieves an energy efficiency of 480 GOPS/W in training mode and 1.04 TOPS/W in recognition mode. The performance and energy breakdown in the training mode are shown in Fig. 17. Table II compares this work with state-of-the-art bio-inspired parallel processors utilizing analog computation. It can be seen that this work achieves very high energy efficiency in both modes. Although it operates relatively slow, the ultra-low power consumption, together with the advantages of nonvolatile memory and unsupervised online trainability make it ideal for autonomous sensory applications. Because this work is the first

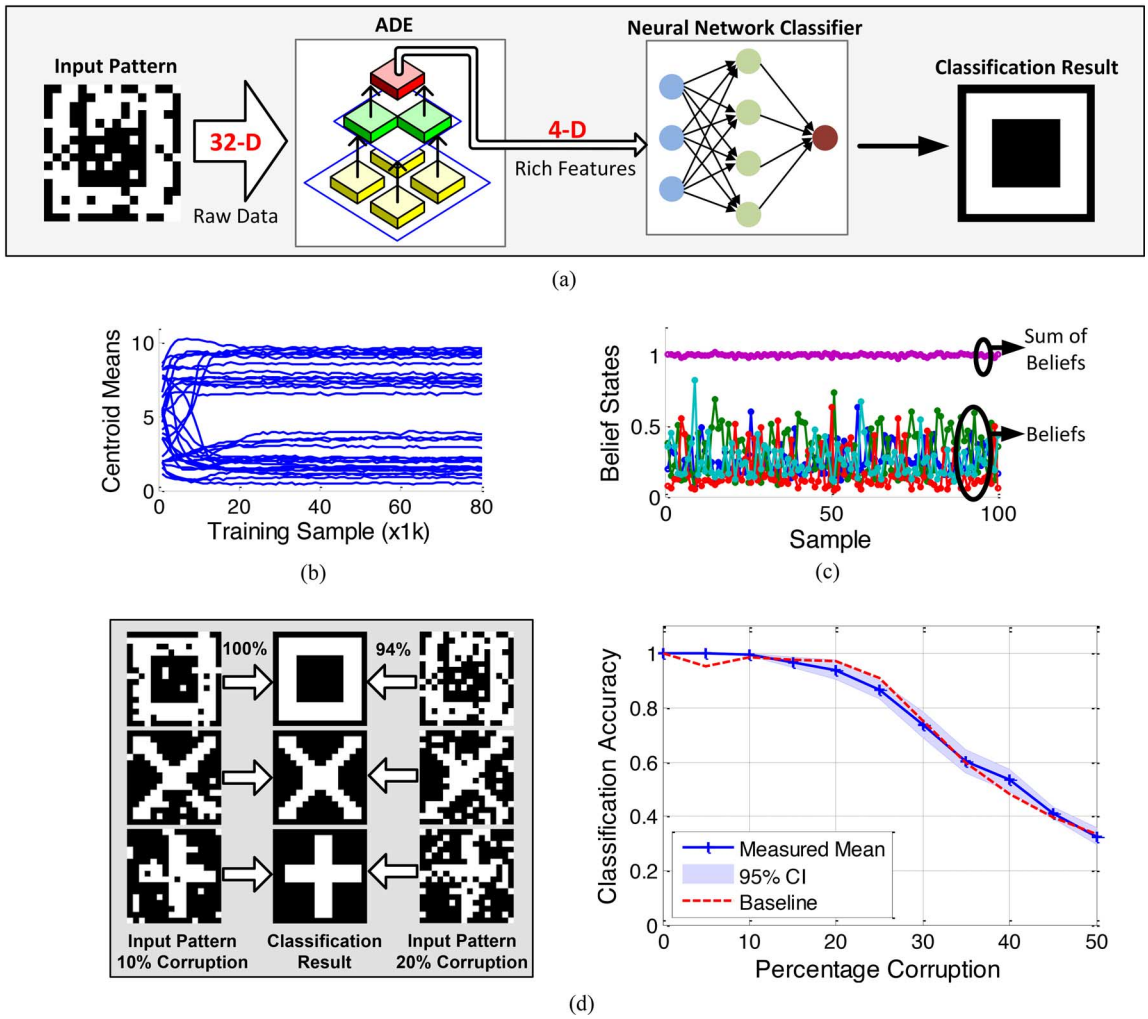


Fig. 16. (a) Feature extraction test setup. (b) The convergence of centroid during training. (c) Output rich feature from the top layer, showing the effectiveness of normalization. (d) Measured classification accuracy using the feature extracted by the chip. The plot on the right shows the mean accuracy and 95% confidence interval (2σ) from the three chips tested, compared to the software baseline.

TABLE I
PERFORMANCE SUMMARY

Techonology	1P8M 0.13µm CMOS	
Power Supply	3V	
Active Area	0.9mm×0.4mm	
Memory	Non-Volatile Floating Gate	
Memory SNR	46dB	
Training Algorithm	Unsupervised Online Clustering	
Input Referred Noise	56.23pA$_{rms}$	
System SNR	45dB	
I/O Type	Analog Current	
Operating Frequency	Training Mode	4.5kHz
	Recognition Mode	8.3kHz
Power Consumption	Training Mode	27µW
	Recognition Mode	11.4µW
Energy Efficiency	Training Mode	480GOPS/W
	Recognition Mode	1.04TOPS/W

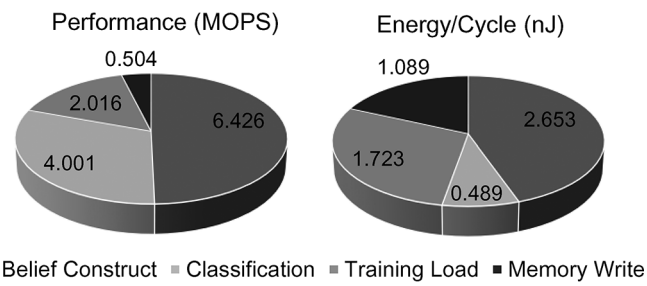


Fig. 17. Performance and energy breakdown in training mode.

reported analog DML system to the best of the authors' knowledge, the foregoing comparison is based only on elementary operations. To better assess the advantage of this design, a digital equivalent of the ADE was implemented in the same process using standard cells with 8-bit resolution and 12-bit memory width. According to post-layout power estimation, this digital equivalent running at 2 MHz in training mode consumes 3.46 W, yielding an energy efficiency of 1.66 GOPS/W, compared to which this work's energy efficiency is 288 times higher.

V. CONCLUSIONS

In this work, we develop an analog deep machine-learning system, the first reported in the literature to the best of the authors' knowledge. It overcomes the limitations of conventional

TABLE II
COMPARISON TO PRIOR WORKS

	This work	JSSC'13 [9]	ISSCC'13 [10]	JSSC'10 [11]
Process	**0.13µm**	0.13µm	0.13µm	0.13µm
Purpose	**DML Feature Extraction**	Neural-Fuzzy Processor	Object Recognition	Object Recognition
Non-volatile Memory	**Floating Gate**	NA	NA	NA
Power (W)	**11.4µW**	57mW	260mW	496mW
Peak Energy Efficiency	**1.04TOPS/W**	655GOPS/W	646GOPS/W	290GOPS/W

digital implementations by exploiting the efficiency of analog signal processing. Reconfigurable current-mode arithmetic realizes parallel computation. A floating-gate analog memory compatible with digital CMOS provides non-volatile storage. Algorithm-level feedback mitigates the effects of device mismatch. System-level power management applies power gating to inactive circuits. We demonstrate online cluster analysis with accurate parameter learning, and feature extraction in pattern recognition with dimension reduction by a factor of 8. In these tests, the ADE achieves a peak energy efficiency of 1 TOPS/W and accuracy in line with the floating-point software simulation. The system features unsupervised online trainability, nonvolatile memory and good efficiency and scalability, making it a general-purpose feature extraction engine ideal for autonomous sensory applications as well as a building block for large-scale learning systems.

REFERENCES

[1] Machine Learning Surveys. [Online]. Available: http://www.mlsurveys.com/

[2] R. Bellman, *Adaptive Control Processes: A Guided Tour*. Princeton, NJ, USA: Princeton University Press, 1961.

[3] I. Arel, D. Rose, and T. Karnowski, "Deep machine learning—A new frontier in artificial intelligence research," *IEEE Computational Intelligence Mag.*, vol. 5, no. 4, pp. 13–18, 2010.

[4] J. Bergstra *et al.*, "Theano: Deep learning on GPUs with Python," in *Big Learning Workshop, NIPS'11*, 2011.

[5] R. Sarpeshkar, "Analog versus digital: Extrapolating from electronics to neurobiology," *Neural Comput.*, vol. 10, pp. 1601–1638, Oct. 1998.

[6] N. Cottini, M. Gottardi, N. Massari, R. Passerone, and Z. Smilansky, "A 33 μW 64 x 64 pixel vision sensor embedding robust dynamic background subtraction for event detection and scene interpretation," *IEEE J. Solid-State Circuits*, vol. 48, no. 3, pp. 850–863, Mar. 2013.

[7] J. Holleman, S. Bridges, B. Otis, and C. Diorio, "A 3 μW CMOS true random number generator with adaptive floating-gate offset cancellation," *IEEE J. Solid-State Circuits*, vol. 43, no. 5, pp. 1324–1336, May 2008.

[8] J. Holleman, A. Mishra, C. Diorio, and B. Otis, "A micro-power neural spike detector and feature extractor in 0.13 μm CMOS," in *Proc. IEEE Custom Integrated Circuits Conf. (CICC)*, Sep. 2008, pp. 333–336.

[9] J. Oh, G. Kim, B.-G. Nam, and H.-J. Yoo, "A 57 mW 12.5 μJ/Epoch embedded mixed-mode neuro-fuzzy processor for mobile real-time object recognition," *IEEE J. Solid-State Circuits*, vol. 48, no. 11, pp. 2894–2907, Nov. 2013.

[10] J. Park *et al.*, "A 646GOPS/W multi-classifier many-core processor with cortex-like architecture for super-resolution recognition," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2013, pp. 17–21.

[11] J.-Y. Kim, M. Kim, S. Lee, J. Oh, K. Kim, and H.-J. Yoo, "A 201.4 GOPS 496 mW real-time multi-object recognition processor with bio-inspired neural perception engine," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 32–45, Jan. 2010.

[12] R. Robucci, J. Gray, L. K. Chiu, J. Romberg, and P. Hasler, "Compressive sensing on a CMOS separable-transform image sensor," *Proc. IEEE*, vol. 98, no. 6, pp. 1089–1101, Jun. 2010.

[13] S. Chakrabartty and G. Cauwenberghs, "Sub-microwatt analog VLSI trainable pattern classifier," *IEEE J. Solid-State Circuits*, vol. 42, no. 5, pp. 1169–1179, May 2007.

[14] K. Kang and T. Shibata, "An on-chip-trainable Gaussian-kernel analog support vector machine," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 7, pp. 1513–1524, Jul. 2010.

[15] T. Yamasaki and T. Shibata, "Analog soft-pattern-matching classifier using floating-gate MOS technology," *IEEE Trans. Neural Networks*, vol. 14, no. 5, pp. 1257–1265, Sep. 2003.

[16] S. Peng, P. Hasler, and D. V. Anderson, "An analog programmable multidimensional radial basis function based classifier," *IEEE Trans Circuits and Syst. I, Reg Papers*, vol. 54, no. 10, pp. 2148–2158, Oct. 2007.

[17] J. Lubkin and G. Cauwenberghs, "A micropower learning vector quantizer for parallel analog-to-digital data compression," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 1998, pp. 58–61.

[18] J. Lu, T. Yang, M. Jahan, and J. Holleman, "A nano-power tunable bump circuit using a wide-input-range pseudo-differential transconductor," *Electron. Lett.*, vol. 50, no. 13, pp. 921–923, June 2014.

[19] Y. Zhang *et al.*, "A batteryless 19 μW MICS/ISM-band energy harvesting body sensor node SoC for ExG applications," *IEEE J. Solid-State Circuits*, vol. 48, no. 1, pp. 199–213, Jan. 2013.

[20] J. Lu, S. Young, I. Arel, and J. Holleman, "A 1TOPS/W analog deep machine-learning engine with floating-gate storage in 0.13 μm CMOS," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, 2014, pp. 504–505.

[21] S. Young, A. Davis, A. Mishtal, and I. Arel, "Hierarchical spatiotemporal feature extraction using recurrent online clustering," *Pattern Recognit. Lett.*, vol. 37, pp. 115–123, Feb. 2014.

[22] J. Lu, S. Young, I. Arel, and J. Holleman, "An analog online clustering circuit in 130 nm CMOS," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, 2013, pp. 177–180.

[23] S. Young, I. Arel, T. Karnowski, and D. Rose, "A fast and stable incremental clustering algorithm," in *Proc. 7th Int. Conf. Information Technology*, Apr. 2010.

[24] J. Lu and J. Holleman, "A floating-gate analog memory with bidirectional sigmoid updates in a standard digital process," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, vol. 2, pp. 1600–1603.

[25] J. Mulder, M. van de Gevel, and A. van Roermund, "A reduced-area low-power low-voltage single-ended differential pair," *IEEE J. Solid-State Circuits*, vol. 32, no. 2, pp. 254–257, Feb. 1997.

[26] B. Gilbert, "Translinear circuits: A proposed classification," *Electron. Lett.*, vol. 11, no. 1, pp. 14–16, 1975.

[27] G. Reimbold and P. Gentil, "White noise of MOS transistors operating in weak inversion," *IEEE Trans. Electron Devices*, vol. 29, no. 11, pp. 1722–1725, Nov. 1982.

[28] M. Pelgrom, A. C. J. Duinmaijer, and A. Welbers, "Matching properties of MOS transistors," *IEEE J. Solid-State Circuits*, vol. 24, no. 5, pp. 1433–1439, Oct. 1989.

[29] S. Young, J. Lu, J. Holleman, and I. Arel, "On the impact of approximate computation in an analog DeSTIN architecture," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 5, pp. 934–946, May 2014.

[30] J. Lazzaro, S. Ryckebusch, M. A. Mahowald, and C. Mead, "Winner-take-all networks of O(n) complexity," in *Advances in Neural Information Processing Systems 1*. San Francisco, CA, USA: Morgan Kaufmann, 1989, pp. 703–711.

Junjie Lu (S'12) received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, China, in 2007, and the Ph.D. degree in electrical engineering from the University of Tennessee, Knoxville, TN, USA, in 2014.

He has previously worked for Philips and Siemens. He joined Broadcom Corporation, Irvine, CA, USA, in 2014 as a staff design engineer, working on low-power, high-precision analog and mixed-signal circuit design.

Steven Young (S'07) earned the B.S. degree in electrical engineering from the University of Tennessee, Knoxville, TN, USA, in 2010. He is currently pursuing the Ph.D. degree in computer engineering in the Machine Intelligence Laboratory at the University of Tennessee.

His current research interests include scalable machine learning with a focus on deep learning.

Itamar Arel (S'92–M'03–SM'06) received the B.S., M.S., and Ph.D. degrees in electrical and computer engineering and an M.B.A. degree, all from Ben-Gurion University, Israel.

He is an Associate Professor of electrical engineering and computer science and Director of the Machine Intelligence Laboratory at the University of Tennessee, Knoxville, TN, USA. His research focus is on high-performance machine intelligence, with emphasis on deep learning architectures, reinforcement learning and scalable decision making under uncertainty.

Jeremy Holleman (S'02–M'09) received the B.S. degree in electrical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 1997, and the M.S. and Ph.D. degrees in electrical engineering from the University of Washington, Seattle, WA, USA, in 2006 and 2009, respectively.

He joined the faculty of the Department of Electrical Engineering and Computer Science at the University of Tennessee, Knoxville, TN, USA, in 2009, where he is currently an Assistant Professor. He has previously worked for Data I/O and National Semiconductor. His research focuses on mixed-mode computation, neuromorphic engineering, and ultra-low-power integrated circuits for biomedical devices and other wireless sensing applications.