Abstract We present an analog implementation of the DeSTIN deep learning architecture. It utilizes floating-gate analog memories to store parameters and current-mode computation and features online unsupervised trainability. The circuit occupies 0.36 mm² in a 0.13 μm CMOS process. At 8.3 kHz, it consumes 11.4 μW from a 3 V supply, achieving 1×10¹² ops/s/W. Measurement demonstrates accuracy comparable to software.

Keywords: Analog signal processing; deep machine learning; neuromorphic engineering

Introduction
Automated processing of large data sets has a wide range of applications [1], but the high dimensionality necessitates unsupervised automated feature extraction. Deep machine learning (DML) architectures have recently emerged as a promising bio-inspired framework, which mimics the hierarchical representation of information in the neocortex to achieve robust automated feature extraction [2]. However, their computational requirements result in prohibitive power consumption when implemented on conventional digital processors or GPUs. Custom analog circuitry provides a means to dramatically reduce the energy costs of the moderate-resolution computation needed for learning applications [3-8]. In this paper we describe an analog implementation of a deep machine learning system first presented in [9]. It features unsupervised online trainability driven by the input data only. 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 technology.

Architecture and Algorithm
The analog deep machine learning engine (ADE) implements Deep Spatiotemporal Inference Network (DesTIN) [10], a state-of-art compositional DML framework, illustrated in Figure 1. Seven identical nodes form a 4-2-1 hierarchy. Each node captures the regularities in its inputs through an unsupervised learning process and constructs belief states, which it passes up to the layer above. The bottom layer acts on raw data (e.g. pixels of an image) and the information becomes increasingly sparse and abstract as it moves up through the hierarchy. 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 [11], which extracts the salient features of the inputs by recognizing patterns of similarity in the input space. Each recognized cluster is represented with a centroid, characterized by estimated mean and variance for each dimension. The node, shown in Figure 2, incorporates an 8×4 array of reconfigurable analog computation cells (RAC), grouped into 4 centroids, each with 8-dimensional input.

A training cycle begins with the classification phase, in which an input vector is assigned to the nearest centroid. The RACs calculate the 1-D Euclidean distance between the centroid mean and the input element oᵢ, which are then wire-summed in current form to yield the total Euclidean distances between the observation and each centroid. The nearest centroid is then selected and its mean and variance estimates are updated such that the FGMs calculate an exponential moving average estimate of the cluster mean and variance. The RAC then incorporates the centroids’ variances to compute the Mahalanobis distance, which is used to probabilistically assign the observation to each centroid.

To take full advantage of the algorithm’s robustness and avoid performance degradation due to analog error sources, extensive system-level simulations were performed [12]. The results
informed the noise and matching specifications for the computational circuits.

**Circuit Implementation**

The Reconfigurable Analog Computation cell (RAC) performs three different operations using subthreshold current-mode computation and reconfigurable current routing. The input current $o$ and the centroid mean $\bar{\mu}$ stored in the FGM-$\mu$ are added with opposite polarity and the difference current $o - \bar{\mu}$ 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 $\hat{\sigma}^2$, or a constant $C$ to compute $D_{MAH}$ or $D_{EUC}$, respectively. The absolute error $|o - \bar{\mu}|$ is used to generate the update pulse for the mean memory, while the quantity $(|o - \bar{\mu}|^2 - \hat{\sigma}^2)$ is computed to determine the update for the variance memory.

The schematic of the analog arithmetic element (AAE) is shown in Figure 3. Amplifier A and the class-B structure M1/M2 provide a virtual ground at the input, allowing high-speed resolution of small current differences. The $X^2/Y$ operator circuit employs the translinear principle [13] to implement efficient current-mode arithmetic. Parameters are stored in an analog floating-gate memory (FGM)[14].

![Figure 2](image2.png)  
**Figure 2.** The node architecture. Each node includes four 8-D centroids as well as control and processing circuitry.

![Figure 3](image3.png)  
**Figure 3.** Schematic of analog arithmetic element.

![Figure 4](image4.png)  
**Figure 4.** Schematic of one channel of the distance processing unit. The translinear loop formed by M1 and M2 yields an inverse relationship between their two drain currents, while the current source $I_{Norm}$ constrains all output currents $I_{Out}$ to be normalized to a constant sum. The Distance Processing Unit (DPU) performs inverse normalization and a winner-take-all operation [16] on the combined distance outputs from the four centroids. The simplified schematic of one channel is shown in Figure 4. The DPU also implements the starvation trace [15], which allows poorly initialized centroids to be slowly drawn towards populated areas of the data space.

The training control circuit converts the memory error current to a 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.
Measurement Results

The ADE occupies an active area of 0.36 mm² in a 0.13 μm CMOS process. With a 3 V power supply, it consumes 27 μW in training mode, and 11.4 μW in recognition mode.

Noise Performance To measure input-referred noise, we construct a histogram of classifications near a decision boundary with adaptation disabled. Assuming additive Gaussian noise, the relative frequency of one of the two classification results approximates the cumulative density function (cdf) of a normal distribution with standard deviation equal to the RMS current noise. The measured input-referred current noise is 56.23 pA rms, which with an input full scale range of 10 nA corresponds to an SNR of 45 dB, or 7.5 bit resolution.

Clustering Test To test the clustering performance of the node, an input dataset of 40,000 8-D vectors was drawn from 4 underlying Gaussian distributions with different mean and variance. During the test, the centroid means were read out every 0.5 s; the locations are overlaid on the data scatter in Figure 5. For easier visual interpretation, 2-D results are shown. The performance of the starvation trace is verified by presenting the node with an ill-posed clustering problem where one centroid is initialized far from the input data, and is therefore never updated without the ST enabled (Figure 5, left). 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 Figure 5 (right).

Feature Extraction Test We demonstrate the full functionality of the chip by performing feature extraction for pattern recognition with the setup shown in Figure 6(a). The input patterns are 16×16 bitmaps corrupted by random pixel errors. An 8×4 moving window selects the ADE’s 32 inputs. The ADE is first trained on unlabeled patterns. After training, adaptation can be disabled and the circuit operates in recognition mode. The 4 belief states from the top layer 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 Figure 6(d), demonstrating robustness to the non-idealities of analog computation.

Efficiency Comparison The measured performance of the ADE is summarized in Table 1. It achieves an energy efficiency of 480 GOPS/W in training mode and 1.04 TOPS/W in recognition mode. Table 2 shows that, compared to other mixed-signal computational circuits, this work achieves very high energy efficiency in both modes.

<table>
<thead>
<tr>
<th>Table 1. Performance Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Technology</td>
</tr>
<tr>
<td>Power Supply</td>
</tr>
<tr>
<td>Active Area</td>
</tr>
<tr>
<td>Memory</td>
</tr>
<tr>
<td>Memory SNR</td>
</tr>
<tr>
<td>Training Algorithm</td>
</tr>
<tr>
<td>Input Referred Noise</td>
</tr>
<tr>
<td>System SNR</td>
</tr>
<tr>
<td>I/O Type</td>
</tr>
<tr>
<td>Operating Frequency</td>
</tr>
<tr>
<td>Power Consumption</td>
</tr>
<tr>
<td>Energy Efficiency</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Figure 5. Clustering test results with bad initial condition without (left) and with the starvation trace enabled.

Figure 6. (a) Feature extraction test setup. (b) The convergence of centroids during training. (c) Rich features output from the top layer, showing the effectiveness of normalization. (d) Measured classification accuracy using the features extracted by the chip. The plot on the right shows the mean accuracy and 95% confidence interval (2) from the three chips tested, compared to the software baseline.
Additionally, a digital equivalent of the ADE was implemented in the same process using logic synthesized from 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, 288 times lower than this analog implementation.

Conclusions

In this paper, we describe an analog deep machine-learning system. It overcomes the limitations of conventional digital implementations through the efficiency of analog signal processing while exploiting algorithmic robustness to mitigate the effects of the non-idealities of analog computation. It demonstrates efficiency more than two orders of magnitude greater than custom digital VLSI with accuracy comparable to a floating-point software implementation. The low power consumption and non-volatile memory make it attractive for autonomous sensory applications and as a building block for large-scale learning systems.

Acknowledgements

This work was partially supported by the NSF through grant #CCF-1218492 and by DARPA through agreement #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.

REFERENCES