Analog Inference Circuits for Deep Learning

Jeremy Holleman, Itamar Arel
Department of Electrical Engineering & Computer Science
University of Tennessee, Knoxville
Knoxville, TN, USA
{jhollema, arel}@utk.edu

Junjie Lu
BroadCom Corporation
Irvine, CA, USA

Steven Young
Oak Ridge National Lab
Oak Ridge, TN, USA

***Abstract*— Deep Machine Learning (DML) algorithms have proven to be highly successful at challenging, high-dimensional learning problems, but their widespread deployment is limited by their heavy computational requirements and the associated power consumption. Analog computational circuits offer the potential for large improvements in power efficiency, but noise, mismatch, and other effects cause deviations from ideal computations. In this paper we describe circuits useful for DML algorithms, including a tunable-width bump circuit and a configurable distance calculator. We also discuss the impacts of computational errors on learning performance. Finally we will describe a complete deep learning engine implemented using current-mode analog circuits and compare its performance to digital equivalents.**

Keywords—analog CMOS; deep learning; neuromorphic computing; error modeling;

I. Introduction

The proliferation of various types sensors in recent years, ranging from camera phones and traffic cameras to implanted medical monitoring devices, has generated vast quantities of data. The high dimensionality of the raw data makes processing and transmission very costly in terms of computational resources and, ultimately, power. Deep Machine Learning (DML), insired by the structure of mammalian cortex, has recently emerged as a promising approach to extract meaningful features from high-dimensional data [1].

DML algorithms are highly parallel and computationally intensive, restricting their use to resource-rich platforms such as grid-connected clusters. Analog computation offers an avenue to reduce the power consumption of DML systems and expand their use to a wide variety of mobile and implanted sensing platforms. In this paper, we describe micro-power analog circuits for computing distance, one of the key operations in DML algorithms. We also describe a modeling approach to account for analog error sources and a system implementation of a complete analog DML engine.

II. Distance Computation

The calculation of the distance between two objects, or equivalently, their similarity, is one of the critical operations common to many machine learning tasks. While there are many metrics that can be used to evaluate similarity in learning tasks, two of the most popular classes of distances are those based on probability distributions, such as a Gaussian kernel, and those based on p-norms [2], such as the Euclidean distance. In this section we will discuss examples of these two classes of distance measurements and analog CMOS circuit implementations.

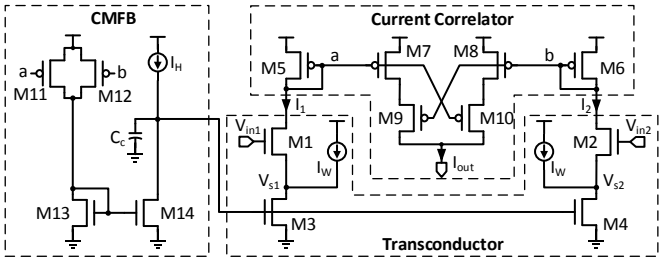


Fig. 1. Schematic of the proposed tunable bump circuit.

A. Bump Circuits

Statistical learning techniques often rely on maximizing the likelihood of an observation given a class assignment and must therefore find the likelihood that an observation could have been drawn from a random distribution centered at a location corresponding to a class prototype or template. This likelihood corresponds to a Guassian kernel as a distance if one assumes that the underlying probability distribution is Gaussian. The "bump circuit" [3] is a well-known circuit that computes a similarity between two inputs with a function similar to a Gaussian distribution. The most basic implementation of the bump circuit includes a differential pair and a current correlator. The circuit shown in Fig. 1 [4] extends the concept of the original bump circuit by allowing the height and width of the output curve to be independently adjusted, as shown in Fig. 2. This feature allows statistical inference systems built using the circuit to incorporate variance information into statistical calculations.

B. Euclidean and Related Distances

The Euclidean distance d_E is another popular distance.

$$d_E = \sqrt{\sum_i (x_i - y_i)^2}$$

Related distances include the Manhattan distance d_{Man} and the Mahalanobis distance d_{Mah}.

$$d_{Man} = \sqrt{\sum_i |x_i - y_i|}$$

This work was partially supported by the NSF through grant #CCF-1218492 and by DARPA through agreement #HR0011-13-2-0016.

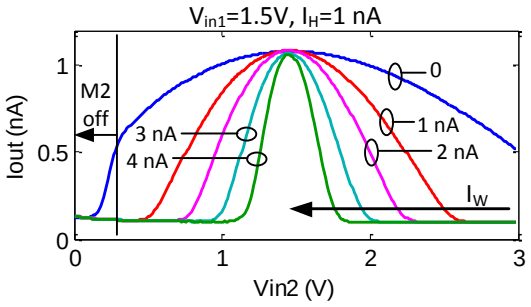
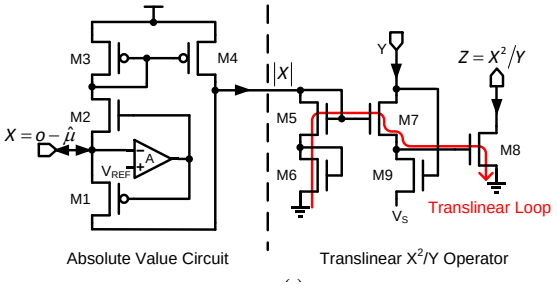
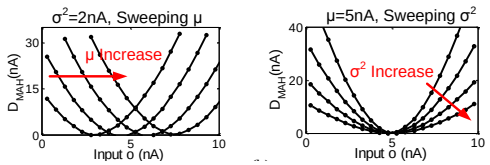


Fig. 2. The effect of shifting (a) one input and (b) the width-tuning current.



(a)



(b)

Fig. 3. (a) Schematic of the absoluate value and x2/y circuits. (b) The effect of shifting one input (left) and the width-tuning current (right).

$$d_{Mah} = \sqrt{(\boldsymbol{x}-\boldsymbol{y})^T S^{-1}(\boldsymbol{x}-\boldsymbol{y})} \approx \sqrt{\sum_i \frac{(x_i - y_i)^2}{\sigma_i^2}}$$

The Manhattan distance is a computationally simple distance that is also proportional to the required update magnitude in online mean estimation. In the Mahalanobis distance formulation S^{-1} is the inverse of the covariance matrix of the underlying multidimensional distribution and the approximation holds if S is diagonal, which is true when correlations between dimensions are negligible. As with the adjustable-width bump, the Mahalanobis distance provides the capability to incorporate variance information into statistical calculations. All of these distances share the property that multi-dimensional distances can be computed on a per-dimension basis and then aggregated across dimensions through simple summation.

Figure 3 illustrates an analog arithmetic element (AAE) [5], a circuit that computes all three of these distances using configurable current-mode circuits. The AAE comprises an absolute value circuit and an x^2/y circuit. The absolute value circuit utilizes an amplifier that senses the polarity of the input current to provide a sign output and selectively route negative currents through a current mirror, inverting their sign. It also acts as a feedback amplifier with M1/M2, synthesizing a low-impedance input to reduce the settling time relative to a simple current-mirror input. When the Manhattan distance is desired, the output is taken directly from the absolute value circuit and aggregated across dimensions.

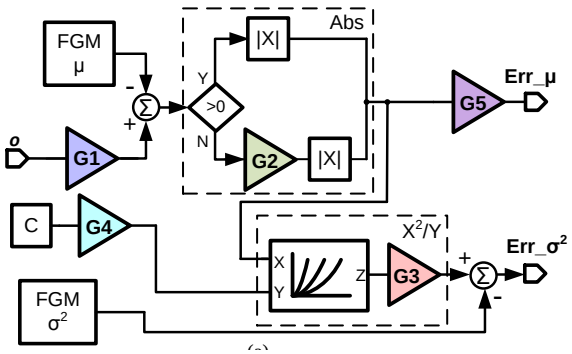


Fig. 4. Error model of the configurable distance calculator. Mean and variance are stored in floating-gate memories (FGM).

The x^2/y block squares the absolute distance (x) and divides by the variance (y) resulting in the Mahalanobis distance. When a Euclidean distance is desired, the y input is connected to a constant unit current. The block operates on the translinear principle, equating the sum of V_{GS5} and V_{GS6} with the sum of V_{GS7} and V_{GS8}, which results in the equality of the current products $I_{D5}I_{D6}$ and $I_{D7}I_{D8}$ when the transistors are biased in weak inversion. The output can then be expressed as $I_{D8}=I_{D5}I_{D6}/I_{D7}$.

III. Error Sensitivity

Analog computational circuits achieve excellent energy efficiency by exploiting the computation intrinsic to the physical operation of transistors and utilizing the continuous range of signal levels in a voltage or current. These same properties also make analog circuits susceptible to error sources, such as noise and mismatch. Fortunately, machine learning algorithms are robust to many error sources due to feedback intrinsic to the algorithms. In order to avoid degrading system accuracy with excessive computational errors while also minimizing power consumption, system-level simulations can be carried out to determine the effect of circuit-level errors on system-level performance [6].

The first step in performing such a system-level evaluation is the construction of an error model. An error model of the configurable distance circuits (Fig. 3(a)) is shown in Fig. 4. Gain errors are injected into the computation after several key blocks. G1 models the error in copying the input current to different units, which results from threshold variations in the input current mirrors. Similarly, the absolute value circuit contributes a gain error G2 when it reverses the polarity for the signal using a current mirror, but not when it simply routes the

input signal to the output. G3 and G4 model variation in the *y* input and output transistor of the x^2/y circuit (M8 in Fig. 3(a)). G5 models the variation in the circuitry used to convey the single-dimension Manhattan distance to the update circuitry.

There can also be errors in the parameter update mechanisms. The update rate can vary in magnitude, causing some elements to update slower or faster than intended, or exhibit asymmetry, where positive and negative updates are of unequal magnitude. Asymmetry will cause the learned centroid to move away from the true mean so that the weaker update occurs more often. The deviation from the true mean will increase with the magnitude of the asymmetry, until the sum of the increments and decrements are equal and equilibrium is achieved.

Fig. 5 shows the impact of these errors, as measured by mean absolute error (MAE) with respect to the same clustering algorithm with no errors introduced. The system is particularly robust to errors in the update mechanism. Input errors simply shift patterns in the input space without reducing their separability and are thus also well tolerated. Because errors in the distance calculation can result in the incorrect cluster being chosen they are somewhat more harmful. However, none of the deterministic error sources contribute to noticeable performance degradation when their standard deviation is less than 10% of the full-scale range, indicating that only around 3.3 bits of matching is required. The most destructive artifact by far was noise, which began to degrade performance with an RMS value of about 1% of full scale.

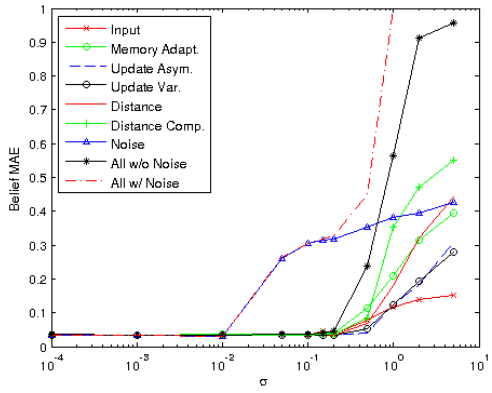


Fig. 5. The Accuracy versus level of error (σ). Gain errors on clean dataset (top) and noisy dataset (bottom).

IV. Complete Learning System

In this section, we describe an example of a complete learning system that implements the DeSTIN architecture. The core computation is performed by the AAE described in II-B with parameters stored in non-volatile floating-gate memories (FGM)[7].

A. Architecture & Circuit Design

The analog deep machine learning engine (ADE) implements Deep Spatiotemporal Inference Network (DesTIN) [1]. Seven identical nodes form a 4-2-1 hierarchy. Each node captures structure in its inputs through a clustering process and constructs belief states about the current input, 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 abstract as it moves up through the hierarchy. The beliefs formed at the top layer are then used as rich features for classification.

The node learns through an online k-means clustering algorithm [8]. Each recognized cluster is represented with a centroid, characterized by estimated mean μ_i and variance σ_i^2 for each dimension. The node, shown in Figure 6, 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 D_{ij}^{EUC} between the i^{th} elements of the j^{th} centroid mean and the input o_i, which are

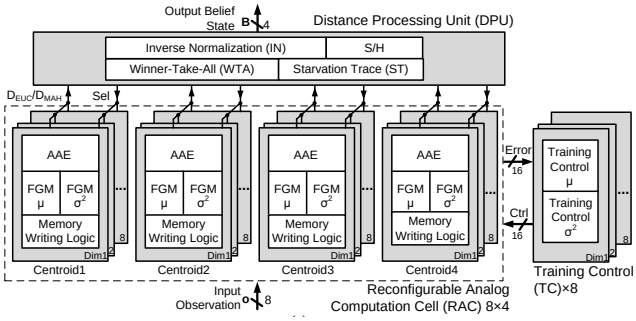


Fig. 6. Block diagram of a single node of the ADE.

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 to calculate an exponential moving average estimate of the cluster mean and variance. The RAC then incorporates the centroids' variances to compute the Mahalanobis distance and its inverse, which is the belief passed to the next layer in the hierarchy.

The Reconfigurable Analog Computation cell (RAC) comprising the AAE and two floating-gate memories performs the core distance calculations. The Distance Processing Unit (DPU) shown in Fig. 7 performs inverse normalization and a winner-take-all operation on the combined distance outputs from the four centroids. The inverse is calculated by fixing the sum of gate-source voltages corresponding to the input and output so that the output current varies inversely with the input current. The sum across centroids of the inverted distance currents is constrained to a constant I_{Norm} so that the inverse distances act as a valid probability mass function. The DPU also implements the starvation trace [8], which allows poorly initialized centroids to be slowly drawn towards populated areas of the data space.

The training control (TC) 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.

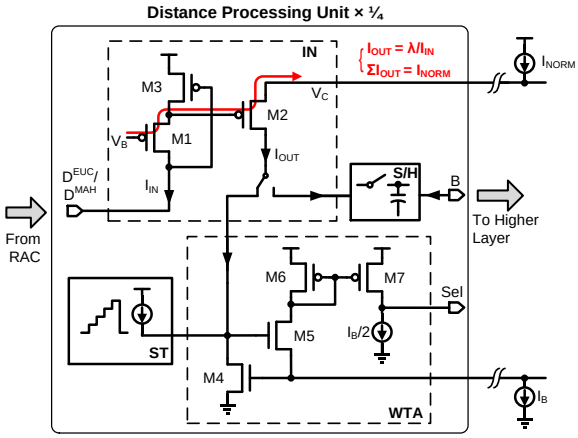


Fig. 7. Schematic of one channel of the distance processing unit. The translinear loop formed by M1 and M2 yields an inverse relationship between their two drain currents, while the current source I_{Norm} constrains all output currents I_{Out} to be normalized to a constant sum.

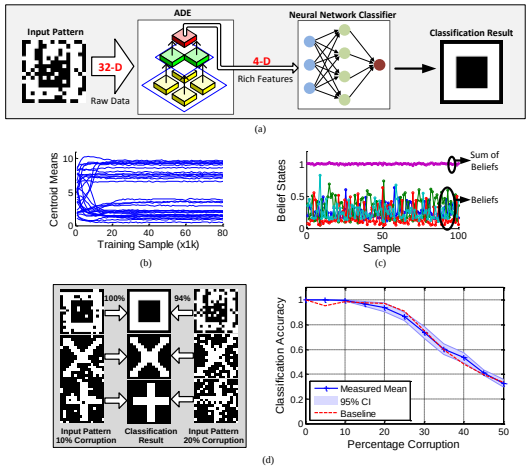


Fig. 8. (a) Feature extraction test setup. (b) The convergence of centroids during training. (c) Rich features output from the top layer, showing the effectiveness of normalization. (d) Measured classification accuracy using the features extracted by the chip. The plot on the right shows the mean accuracy and 95% confidence interval (2σ) from the three chips tested, compared to the software baseline.

B. Measurements

The ADE occupies an active area of 0.36 mm^2 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. The measured input-referred current noise is 56.23 pA_{RMS} with a full-scale range of 10 nA, corresponding to an SNR of 45 dB.

We demonstrate the full functionality of the chip by performing feature extraction for pattern recognition with the setup shown in Figure 8(a). The input patterns are 16×16 bitmaps corrupted by random pixel errors. A moving 8×4 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 features, achieving a dimension reduction from 32 to 4. A software neural network then classifies the four-element 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 8(d), demonstrating robustness to the non-idealities of analog computation.

The ADE achieves an energy efficiency of 480 GOPS/W in training mode and 1.04 TOPS/W in recognition mode. 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.

V. Conclusion

We have presented a variety of useful circuits suitable for low-power analog implementations of machine learning systems. While analog circuits contribute errors to a compuational system, error modeling can guide the circuit design to avoid performance degradation. Finally, a system implementation demonstrates that analog learning systems can exhibit accuracy comparable to floating-point software while consuming orders of magnitude less power than even a custom digital implementation.

References

[1] S. Young, A. Davis, A. Mishtal and I. Arel, "Hierarchical spatiotemporal feature extraction using recurrent online clustering," *Pattern Recognition Letters*, vol. 37, pp. 115-123, Feb. 2014.

[2] A.W. Naylor and G.R. Sell, *Linear Operator Theory in Engineering and Science*, Springer, 2000.

[3] T. Delbruck, "`Bump' circuits for computing similarity and dissimilarity of analog voltages," in *Proc. Int. Joint Conf. on Neural Networks*, Jul. 1991, pp. 475-479.

[4] J. Lu, T. Yang, M.S. Jahan, J. Holleman, "Nano-power tunable bump circuit using wide-input-range pseudo-differential transconductor," *Electronics Letters*, Vol. 50, No. 13, pp. 921-923, June 2014.

[5] J. Lu, S. Young, I. Arel, J. Holleman, "A 1 TOPS/W Analog Deep Machine-Learning Engine with Floating-Gate Storage in 0.13um CMOS," *IEEE Journal of Solid-State Circuits*. Vol. 50, Issue 1, pp. 270-281, Jan. 2015.

[6] S. Young, J. Lu, J. Holleman and I. Arel, "On the impact of approximate computation in an analog DeSTIN architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, no. 99, p. 1, Oct. 2013.

[7] 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.

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