Analog Neuromorphic System based on Multi Input Floating Gate MOS Neuron Model

Ankit Tripathi[1], Mehdi Arabizadeh[2], Sourabh Khandelwal[2], and Chetan Singh Thakur[1]

[1]Department of Electronic Systems Engineering, Indian Institute of Science (IISc), Bengaluru, India
[2]School of Engineering, Macquarie University, Sydney, Australia

***Abstract*—This paper introduces a novel implementation of the low-power analog artificial neural network (ANN) using Multiple Input Floating Gate MOS (MIFGMOS) transistor for machine learning applications. The number of inputs to a neuron in an ANN is the major bottleneck in building a large scale analog system. The proposed MIFGMOS transistor enables to build a large scale system by combining multiple inputs in a single transistor with a small silicon footprint. Here, we show the MIFGMOS based implementation of the Extreme Learning Machine (ELM) architecture using the receptive field approach with transistor operating in the sub-threshold region. The MIFGMOS produces output current as a function of the weighted combination of the voltage applied to its gate terminals. In the ELM architecture, the weights between the input and the hidden layer are random and this allows exploiting the random device mismatch due to the fabrication process, for building Integrated Circuits (IC) based on ELM architecture. Thus, we use implicit random weights present due to device mismatch, and there is no need to store the input weights. We have verified our architecture using circuit simulations on regression and various classification problems such as on the MNIST data-set and a few UCI data-sets. The proposed MIFGMOS enables combining multiple inputs in a single transistor and will thus pave the way to build large scale deep learning neural networks.**

***Index Terms*—MIFGMOS, Sub-threshold region, ELM**

I. Introduction

Recent years have witnessed the hardware implementation of various machine learning algorithms for their ease of exploiting random device mismatches to overcome the complexity of software computations [1] [2] [3]. These hardware implementations enable the deployment of deep neural networks (DNN) for tasks of high complexity such as image classification, pattern recognition and object detection, but this ease of computation comes at the cost of a trade-off between low power and computation capability. Out of various architectures that supports DNNs, such as CPU, GPU, FPGA, and ASIC, realization in the analog domain is preferred owing to benefits of compactness and low power. In the analog context, Floating Gate Metal Oxide Semiconductor (FGMOS) devices have demonstrated the capability of low-power analog computation and the possibility of uses in the neuromorphic circuits. These have been used in a wide range of circuit applications related to learning and interfering, such as in an activation function generator [4], neuron modelling [5] [6], competitive learning system [7], implementation of a neuromorphic learning algorithm [8], pattern classification problem [9], and implementation of an online unsupervised deep learning system [10]. However, all these implementations can further be improved by using the multi-input variant of the FGMOS, which works in the same way as the FGMOS does but having multiple inputs. An earlier work [11] introduces a trainable analog block (TAB) which operates in the sub-threshold region and exploits device mismatches for computation. The TAB uses an additional circuit, referred to as the Weighted Average Circuit (WAC), for summation of inputs. As an extension to this work, the die size can be reduced considerably by using a multiple-input floating-gate MOS (MIFGMOS), which because of taking a weighted summation of its inputs can act as a single unit for exhibiting random oxide thickness as input weights and producing an output current as activation when being a part of a differential pair.

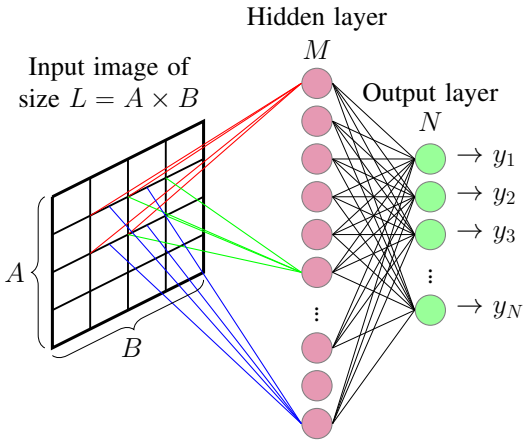

Fig. 1: Local receptive fields based feedforward neural network.

The key idea of Extreme Learning Machine (ELM) [12] that the input layer weights and hidden layer parameters need not required to be tuned is well suited for our arrangement. This eliminates the need of storage units for input layer weights. Our work is novel in that a hidden layer neuron is realized by a MIFGMOS differential amplifier, which acts as a single unit for taking inputs, calculating their weighted sum, and producing a corresponding output current that serves as an activation function. The full architecture of the ELM was simulated using the Cadence analog design environment

on 65nm CMOS technology, with output weights having a resolution of 8-bits, to validate the system's ability to perform regression and classification.

II. Theory & Network Architecture

A. Extreme Learning Machine (ELM)

The system (Figure (1)) implements the following equation of the ELM algorithm [12].

$$m_{ki} = z(\mathrm{x}_k.\mathrm{w}_i^{(1)} + b_i) \qquad i = 1, \cdots, M \tag{1}$$

where m_{ki} is the output of the i^{th} hidden layer neuron for the k_{th} sample, $z(x)$ is the neuron's activation function and is kept same for all M neurons, $\mathrm{w}_i^{(1)} \in \mathcal{R}^L$ is the vector of the weights connecting inputs of size L to the i^{th} hidden layer neuron and b_i is the bias to the i^{th} hidden layer neuron. x_k is the input to the network where k denotes the k^{th} sample. The activation function of our system is Sigmoid, as shown in equation below.

$$z(g_i) = \frac{1}{1 + e^{-g_i}}. \tag{2}$$

Here, instead of having full connectivity between the input and hidden layer, we have shown local receptive fields based connectivity, which captures local correlation to map the input to higher dimensional space [13]. The output of the network for x_k is $\mathrm{y}_k \in \mathcal{R}^N$, the j_{th} component of which is

$$y_{kj} = \sum_{n=1}^{M} m_{kn} w_{jn}^{(2)} \tag{3}$$

where m_{kn} is the output of the n_{th} neuron as in equation (1) , $w_{jn}^{(2)}$ is the output weight connecting the n_{th} hidden layer neuron to the j_{th} output layer neuron. For C distinct samples, there are $C \times N$ equations of the form of equation (4). These equations can be written compactly as

$$HW^{(2)} = Y \tag{4}$$

where

$$H = \begin{bmatrix} z(\mathrm{x}_1.\mathrm{w}_1^{(1)} + b_i) & \cdots & z(\mathrm{x}_1.\mathrm{w}_M^{(1)} + b_i) \\ \vdots & \cdots & \vdots \\ z(\mathrm{x}_C.\mathrm{w}_1^{(1)} + b_i) & \cdots & z(\mathrm{x}_C.\mathrm{w}_M^{(1)} + b_i) \end{bmatrix}_{C \times M}$$

$$W^{(2)} = \begin{bmatrix} \mathrm{w}_1^{(2)} \\ \vdots \\ \mathrm{w}_N^{(2)} \end{bmatrix}_{N \times M}^T \quad and \quad Y = \begin{bmatrix} \mathrm{y}_1 \\ \vdots \\ \mathrm{y}_C \end{bmatrix}_{C \times N}$$

$$\mathrm{w}_k^{(2)} = \begin{bmatrix} w_{k1}^{(2)} & w_{k2}^{(2)} & \cdots & w_{kM}^{(2)} \end{bmatrix}_{1 \times M}^T$$

$\mathrm{w}_k^{(2)}$ is $M \times 1$ vector of weights connecting the k_{th} output layer neuron to the hidden layer neurons.

Calculation of output weights in the ELM is solving a generalized linear problem [14], as given in equation (4), where H is the output of the hidden layer, $W^{(2)}$ is the weight matrix connecting the hidden layer to the output layer and Y is the output matrix. Solving this problem is equivalent to finding the $W_0^{(2)}$ achieving the minimum of the following least-square problem.

$$\left|\left| HW_0^{(2)} - Y \right|\right| = \min_{W^{(2)}} \left|\left| HW^{(2)} - Y \right|\right|$$

Out of all the solutions to this least square problem, the one which gives the smallest norm is the following unique solution.

$$\widetilde{W}_0^{(2)} = H^{\dagger} Y$$

where $H^{\dagger}$ is the Moore-Penrose inverse of H.

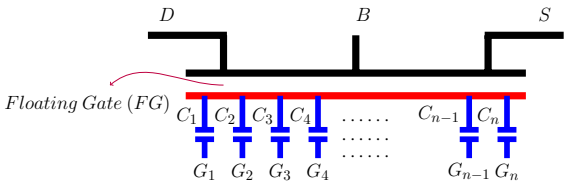

Fig. 2: A multiple input floating gate MOS (MIFGMOS). The value of C_i depends on the respective gate size.

B. Multiple Input Floating Gate MOS (MIFGMOS)

Figure (1) shows the inputs to the hidden layer neurons via local receptive fields [13]. In theory all the inputs can be connected to a hidden layer neuron but here the MIFGMOS (Figure 2) poses a restriction on the number of inputs that can be connected to a hidden layer neuron. The MIFGMOS that we used has a maximum of 9 inputs which can take connections from a receptive field of size 3×3. A MIFGMOS [5] [15] calculates a weighted summation of its inputs in the following way [16].

$$V_{FG} = \frac{\left(Q_{FG} + C_{FGD}V_D + C_{FGS}V_S + C_{FGB}V_B + \sum_{i=1}^{n} C_i V_{G_i}\right)}{C_{Total}} \tag{5}$$

where,

$$C_{Total} = C_{FGD} + C_{FGS} + C_{FGB} + \sum_{i=1}^{n} C_i$$

FG denotes the floating gate. $D, S, B,$ and G_i denote drain, source, body, and gate terminals respectively, n denotes the number of gate terminals. Q_{FG} is the charge on the floating gate. Rest of the symbols have their usual meaning.

These gate terminals can be thought of as inputs to a particular hidden layer neuron. The capacitive coupling factor C_i, which shows the relative strength by which the i^{th} gate terminal affects the potential of the floating gate, is the input layer weight and is random in nature because of the fabrication process. The MIFGMOS that we incorporated is based on the UC Berkeley BSIM6 Verilog-A model.

III. Circuit Implementation

A. Neuron Block

The circuit that we employed to get a Sigmoid activation is the MIFGMOS differential pair [16] as shown in Figure (3). This circuit when biased in the sub-threshold region with $v_{DS} > 5V_T$ for each transistor, where V_T is the thermal voltage, gives the following approximate expression for the drain currents I_{D1} and I_{D2}.

$$I_{D1} = \frac{I_{bias}}{1 + exp[-(V_{FG} - V_{ref})/\eta V_T]} \tag{6}$$

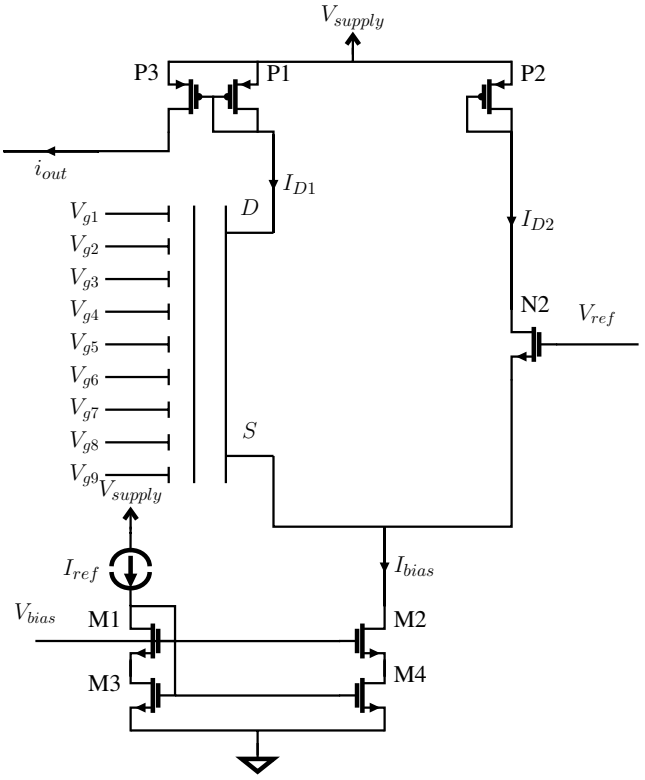

Fig. 3: Schematic of a neuron block based on MIFGMOS differential pair.

$$I_{D2} = \frac{I_{bias}}{1 + exp[-(V_{ref} - V_{FG})/\eta V_T]} \tag{7}$$

where V_{FG} is as defined in equation (5). η is the sub-threshold slope factor. I_{bias} is the biasing current which is copied from I_{ref} by means of a high swing current mirror formed by $M1, M2, M3$ and $M4$. V_{bias} biases the gate voltage of $M1$ & $M2$ for the proper operation of the current mirror. The diode connected MOS $P1$ and $P2$ are the current source loads. The current I_{D1} is mirrored by a current mirror formed by $P1$ and $P3$ to I_{out} which is the output of a neuron. The form of equation (6) is similar to the Sigmoid function (eqation (2)) with scaling and shifting. The parameter V_{ref} is the systematic offset which gives additional variability among activations of neurons in case there is not enough random mismatch [17]. The output of a hidden layer neuron block is shown in Figure (4).

The Figure (4) shows the activation of a neuron for all gate terminals as their voltage is varied one at a time with random coupling capacitance thickness, which is a configurable parameter of our setup. The heterogeneity among activations with respect to different gate terminals is evident from the figure. These activations act as a random basis covering the whole input space, which helps to create non-linear classification boundary and learning non-linear regression function.

B. Output Weight Block

The output synaptic weights are realized by an 8-bits M-2M Digital to Analog Converter (DAC) [18] [19] [20]. A 3-bit version of this DAC is shown in Figure (6) for clarity. In the ELM architecture, there is full connectivity between the hidden layer neurons and output neurons in conjunction with the fact that the output layer neurons are linear. The number of output weight blocks are $M \times N$. The input current I_{in} shown in the figure is the output of a hidden layer neuron block, as shown in Figure (3). This DAC reduces the input current by a constant factor in each successive branch and the current in each

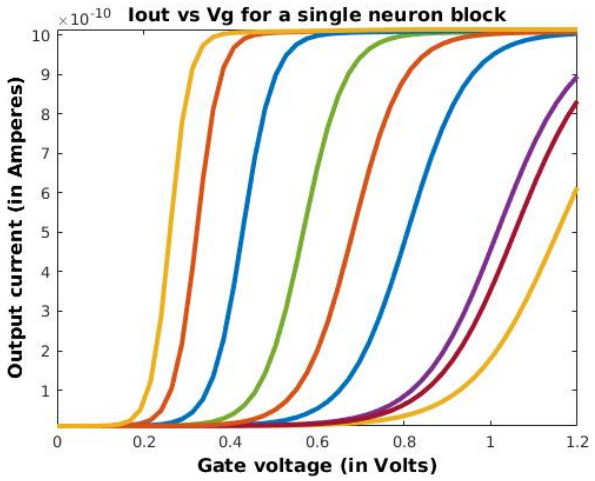

Fig. 4: Activation of a single neuron block for different gate voltages over random weights. Only one gate voltage is varying at a time.

branch is routed to either I_{retain} or I_{dump} depending on whether its weight bit is high or low. The current I_{retain} is copied via a current mirror and is further routed either via transistor M_{pos} or M_{neg}, out of both transistors, only one is active at a time depending on the value of the sign bit, which corresponds to the value of weight being positive or negative, and hence changes the direction of current accordingly. The current I_{dump} is dumped to the ground by passing it through a diode-connected MOS so that the impedance that I_{retain} & I_{dump} sees are the same, and hence a better accuracy is achieved. The gate of transistors of the upper part of the DAC (denoted as N) is biased by a master bias voltage [21]. Since its output current magnitude is always less or equal than the magnitude of the input current it is necessary to have output weights confined to [-1,1] and being taken care in offline learning process. We have employed an 8-bit DAC, which realizes 255 different current levels and an additional bit for sign of the weights.

IV. Results

The number of connections to a hidden layer neuron are configurable to a maximum of nine which is a degree of freedom facilitated by our model, but can be scaled in the future model. We have tested the system for two-dimensional regression, binary classification, and MNIST [22] digit recognition problem. For regression two inputs instantiation of 100 neuron blocks, each cascaded with M-2M DAC were used. The activations were collected in the simulator, and the output weights were calculated off the simulator. These output weights were then stored by means of 8-bit shift register, outputs of which are inputs to the M-2M DAC, labelled as wt_i, $i = 1, \cdots, 8$ in

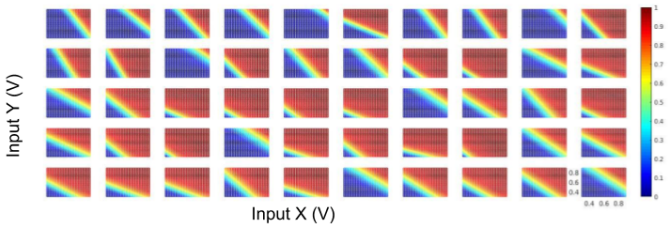

Fig. 5: Activations of 50 neuron blocks. These neuron blocks are two inputs instantiation of circuit of Figure (3).

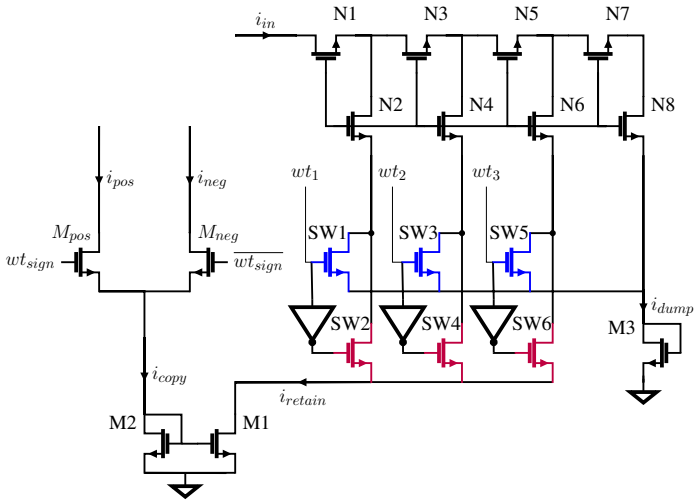

Fig. 6: Current mode M-2M DAC.

Figure (6). The system is able to learn two-variable algebraic function such as cubic $x^3 + y^3$ with an error percentage of 1.69%, the results of which are shown in Figure (7). Here, error is defined as the ratio of RMS value of the difference between the target and learnt function, and the RMS value of the target function.

The ability of our system to perform classification was tested using small data-sets chosen from the UCI machine learning repository. The circuit simulation results are shown in Table (I) along with the software simulation results. The two data-sets, Banknote Authentication and Pima Indians Diabetes, have different feature sizes for which different inputs instance of MIFGMOS were used.

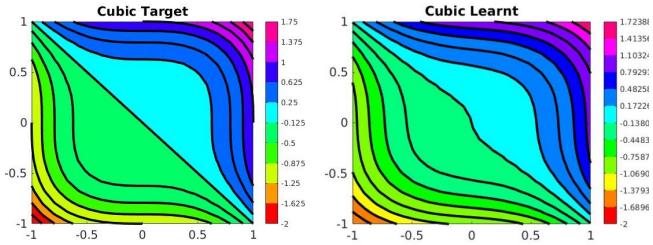

Fig. 7: Regression on $x^3 + y^3$ for 100 neuron blocks.

Finally, we implemented the system against a binary version of the handwritten digit database MNIST where pixel values above and below a certain threshold were mapped to one and zero, respectively. The resulting images were fed to the network in a convolutional neural network (CNN) manner, where the size of the receptive field is 3×3 owing to the maximum number of inputs to the hidden layer neurons being nine. The weights of all the fields were random and distinct. For this implementation, we simulated the system with 676 hidden layer neurons at a time, without using the output weight block

Database	Training (Simulator)	Testing (Simulator)	Training (Software)	Testing (Software)
Banknote Authentication	71.83	70.23	74.45	73.86
Pima Indians Diabetes	84.76	82.58	89.62	87.93

TABLE I: Accuracy over binary databases (in %) for 100 neuron blocks.

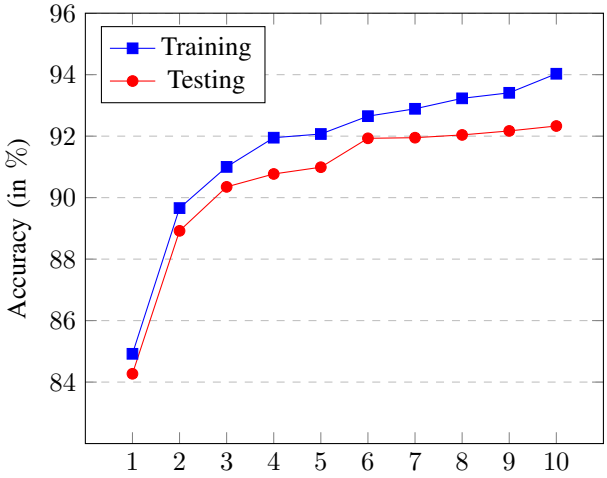

Fig. 8: Accuracy of the system with the binary MNIST dataset.

in the simulator because of limitation on the simulation environment. All the simulations with 676 neurons were simulated with distinct random weights. The accuracy of the system for the different number of hidden layer neurons in multiples of 676 is shown in the Figure (8).

V. Conclusion

We presented a novel method to realizing the ELM in the analog domain by employing the MIFGMOS, which utilizes mismatches to perform computations. The MIFGMOS is a substantial improvement over the previous work [11] [23] in terms of optimizing the die area because it obviates the need of using Weighted Average Circuit (WAC) for performing weighted summation of inputs. This incorporation of an MIFGMOS based neuron has an additional advantage of realizing a large number of neuron blocks because ELM requires a large number of hidden layer neurons relative to sample size for effective computation that gives better accuracy. This along with the fact that analog domain implementation facilitates low power consumption is an additional edge. Our software simulations are performed on 65nm technology, which aids heterogeneity among neuronal curves even further. The diversity among neuronal curves is a serious concern where there is a limitation on the die area. In that case, an upper limit on the number of hidden layer neurons requires the variance among the input layer weights to be as large as possible. This can not always be guaranteed because there is an upper and lower bound on oxide thickness which manifests itself as input weights. In that case, a deliberate offset between the reference voltage and bias current of neuron block would serve as an additional parameter of variation.

The ability of system to perform regression and classification is also presented. The accuracy of the system on the MNIST data-set shows that the system is able to perform image classification tasks quite efficiently. With as little as 676 neurons which is less than the feature size of the input (784), the accuracy is greater than 84%. This system thus provides a favorable alternative for image classification tasks, on account of its small size, cost efficiency and better accuracy (on account of large number of input layer connections).

VI. Acknowledgement

This work was supported by the Pratiksha Trust grant (Pratiksha-YI/2017-8512), Indian Institute of Science.

REFERENCES

[1] R. Perfetti and E. Ricci, "Analog neural network for support vector machine learning," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 1085–1091, July 2006.

[2] M. Kawaguchi, M. Umeno, and N. Ishii, "The two-stage analog neural network model and hardware implementation," in *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, Aug 2014, pp. 936–941.

[3] L. Gatet, H. Tap-Beteille, and M. Lescure, "Analog neural network implementation for a real-time surface classification application," *IEEE Sensors Journal*, vol. 8, no. 8, pp. 1413–1421, Aug 2008.

[4] I. Bayraktaroglu, A. S. Ogrenci, G. Dundar, S. Balkir, and E. Alpaydin, "Annsys (an analog neural network synthesis system)," in *Proc. Int. Conf. Neural Networks (ICNN'97)*, vol. 2, Jun. 1997, pp. 910–915 vol.2.

[5] T. Shibata and T. Ohmi, "A functional MOS transistor featuring gate-level weighted sum and threshold operations," *IEEE Transactions on Electron Devices*, vol. 39, no. 6, pp. 1444–1455, Jun. 1992.

[6] V. Kornijcuk, H. Lim, J. Y. Seok, G. Kim, S. K. Kim, I. Kim, B. J. Choi, and D. S. Jeong, "Leaky integrate-and-fire neuron circuit based on floating-gate integrator," *Frontiers in Neuroscience*, vol. 10, may 2016.

[7] D. Hsu, M. Figueroa, and C. Diorio, "Competitive learning with floating-gate circuits," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 732–744, May 2002.

[8] P. Hasler and J. Dugger, "An analog floating-gate node for supervised learning," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 5, pp. 834–845, May 2005.

[9] F. Merrikh-Bayat, X. Guo, M. Klachko, M. Prezioso, K. K. Likharev, and D. B. Strukov, "Sub-1-us, sub-20-nj pattern classification in a mixed-signal circuit based on embedded 180-nm floating-gate memory cell arrays," *CoRR*, vol. abs/1610.02091, 2016.

[10] J. Lu, S. Young, I. Arel, and J. Holleman, "A 1 Tops/w analog deep machine-learning engine with floating-gate storage in 0.13 μm CMOS," *IEEE Journal of Solid-State Circuits*, vol. 50, no. 1, pp. 270–281, Jan. 2015.

[11] C. S. Thakur, R. Wang, T. J. Hamilton, R. Etienne-Cummings, J. Tapson, and A. van Schaik, "An analogue neuromorphic co-processor that utilizes device mismatch for learning applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 65, no. 4, pp. 1174–1184, Apr. 2018.

[12] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Proc. IEEE Int. Joint Conf. Neural Networks (IEEE Cat. No.04CH37541)*, vol. 2, Jul. 2004, pp. 985–990 vol.2.

[13] G. Huang, Z. Bai, L. L. C. Kasun, and C. M. Vong, "Local receptive fields based extreme learning machine," *IEEE Computational Intelligence Magazine*, vol. 10, no. 2, pp. 18–29, May 2015.

[14] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, dec 2006.

[15] L. Topór-Kaminski and P. Holajn, "Multiple-input floating-gate mos transistor in analogue electronics circuit," *TECHNICAL SCIENCES*, vol. 52, no. 3, 2004.

[16] K. Yang and A. G. Andreou, "A multiple input differential amplifier based on charge sharing on a floating-gate mosfet," *Analog Integrated Circuits and Signal Processing*, vol. 6, no. 3, pp. 197–208, Nov 1994. [Online]. Available: https://doi.org/10.1007/BF01238888

[17] C. S. Thakur, T. J. Hamilton, R. Wang, J. Tapson, and A. van Schaik, "A neuromorphic hardware framework based on population coding," in *2015 International Joint Conference on Neural Networks (IJCNN)*, July 2015, pp. 1–8.

[18] K. Bult and G. Geelen, "An inherently linear and compact most-only current-division technique," in *1992 IEEE International Solid-State Circuits Conference Digest of Technical Papers*, Feb 1992, pp. 198–199.

[19] C. M. Hammerschmied and Q. Huang, "Design and implementation of an untrimmed mosfet-only 10-bit a/d converter with -79-db thd," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 8, pp. 1148–1157, Aug 1998.

[20] T. Lee and C. Lin, "Nonlinear r-2r transistor-only dac," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 10, pp. 2644–2653, Oct 2010.

[21] T. Delbruck and A. van Schaik, "Bias current generators with wide dynamic range," in *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*, vol. 1, May 2004, pp. I–I.

[22] Y. LeCun and C. Cortes, "MNIST handwritten digit database," 2010. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[23] C. S. Thakur, R. Wang, T. J. Hamilton, J. Tapson, and A. van Schaik, "A low power trainable neuromorphic integrated circuit that is tolerant to device mismatch," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 2, pp. 211–221, Feb 2016.