Efficient Implementation of a Fully Analog Neural Network on a Reconfigurable Platform

Afolabi Ige
School of Eletrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA
aige3@gatech.edu

Jennifer Hasler
School of Eletrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA
jennifer.hasler@ece.gatech.edu

***Abstract*—This paper investigates the potential of floating gate field-effect transistors (FETs) as primitives for subthreshold computation in analog neural networks. By leveraging the inherent properties of these transistors, we demonstrate their suitability for constructing neural network activation functions, such as sigmoid and rectified linear units (ReLUs), as well as winner-take-all (WTA) circuits for softmax activation. Our end-to-end analog implementation successfully classifies the concentric circles problem, illustrating the advantages of maintaining an analog signal chain throughout the process.**

***Index Terms*—Analog computing, Analog neural networks, Computing-in-memory**

I. The Argument for Analog

As the need for computational complexity in deep neural networks intensifies, especially for edge computing devices, the field of analog computing has witnessed a renewed interest. Emerging devices have demonstrated efficient in-memory computing crossbar networks with floating gate field-effect transistors (FG FETs) being among the first devices investigated for these applications [1]. While contemporary approaches have exhibited fully analog-mode crossbars for vector matrix multiplication, end-to-end analog hardware implementations from input to classification remain rare. Results are frequently reported based on software simulations, which can overestimate hardware performance due to optimistic modeling of non-idealities. This paper addresses this gap by presenting a fully implemented and measured analog neural network, leveraging the energy efficiency of the analog domain without limitations imposed by digital or mixed-signal devices at the crossbar's periphery.

Our method utilizes FG FETs available on an in-house system-on-chip (SoC) field programmable analog array (FPAA), a versatile reconfigurable computing platform. This platform allows for arbitrary analog circuit experimentation and provides >12-bit precision synaptic weights [2]. Section II explores the computation primitives, followed by the introduction of analog activation circuits and their measured data in Section III. Section IV discusses the implementation of two integrated neural networks classifying the concentric circles problem using a sigmoid and Rectified Linear Unit (ReLU) activation function, while Section V concludes the discussion.

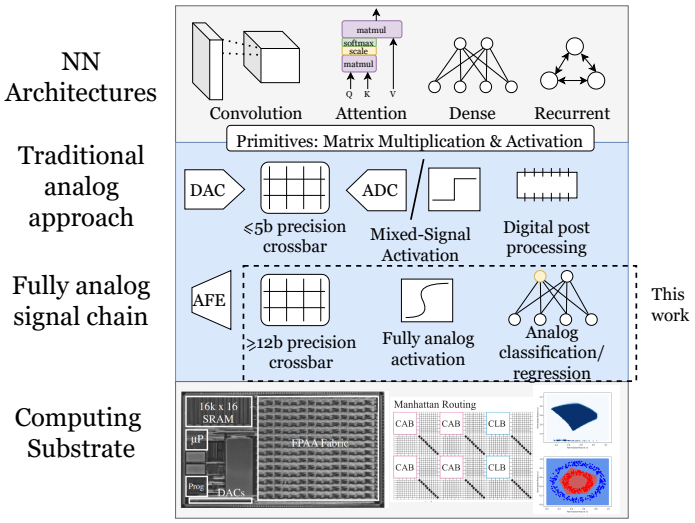

Fig. 1: Most neural network architectures share the same primitives of matrix multiplication and neuron activation. A fully analog signal chain allows the implementation to retain the majority of the computing efficiency as opposed to paying the cost overhead of an analog matrix crossbar with mixed signal neuron activation.

II. Computation Primitives

Floating gate (FG) transistors have since been known as powerful primitives in the design of vector matrix multiplier (VMM) crossbar arrays [1], providing an innovative solution to some of the most pressing challenges in modern computing. At the heart of their functionality lies the ability to store non-volatile charge, which corresponds to the weights of the neural network (Fig. 2a). This feature not only ensures stability but also enables them to operate in the subthreshold regime, thereby significantly reducing power consumption compared to traditional digital devices. To further understand the advantages of utilizing floating gate transistors in this context, it is essential to examine the EKV model [3], [4], which sheds light on the exponential multiplication between the gate voltage and source voltage in subthreshold p-type MOSFETs, given that other variables are held relatively constant and the transistor is in saturation (eq 1). Here, U_t is thermal voltage, κ is the

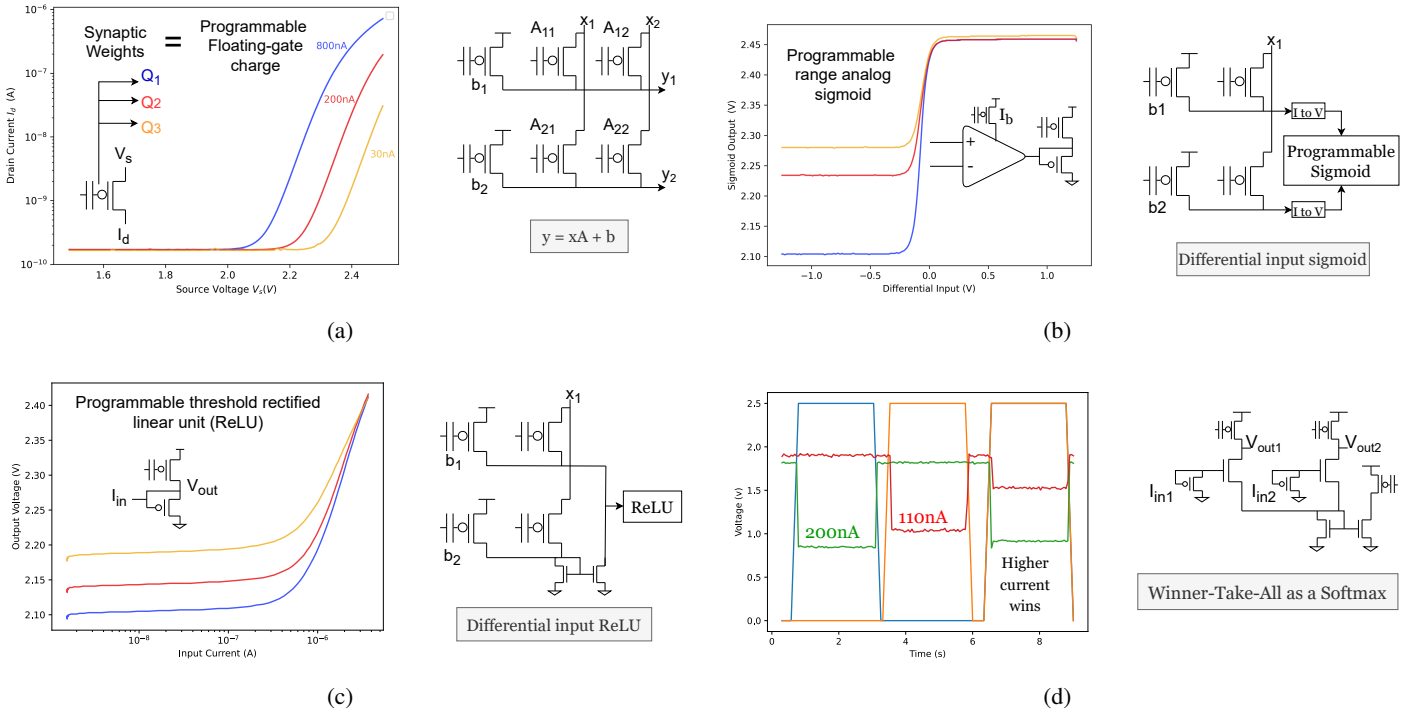

Fig. 2: (a) Synaptic weights can be mapped to charge stored at the floating gate node of FG FETs. This is the basis of in-memory computing. When placed in a crossbar, it forms a vector matrix multiplication circuit with voltage input and current computation. (b) A sigmoid function can be achieved with a differential pair found in an OTA. The additional transistors set the rails which are required to interface with subsequent source-input crossbars. (c) A ReLU analog circuit has its threshold current set by the bias FET and linearly increases output voltages for currents above that value. (d) A two input Winner Take All Network. The graphs show each individual input winning in isolation as well as the larger current winning in simultaneous operation.

coupling from gate voltage to surface potential of the channel ψ.

$$I_s = I_{th} e^{(\kappa(V_{dd}-V_{fg}-V_g-V_{th})-(V_{dd}-V_s))/U_t}$$
$$I_s \approx I_{th} e^{V_{fg}} * e^{V_{dd}-V_s} \quad (1)$$

Building upon the merits of FG transistors in VMM crossbar arrays, it is crucial to address the practical aspects of computation and communication within the system. Although the core computation is executed in the current domain, it is advantageous to convert the resulting currents to voltages for communication purposes. This is primarily because voltages offer a more efficient means of broadcasting signals across the network, ensuring lower power dissipation and higher signal integrity. Typically, this current-to-voltage conversion takes place at the activation function stage, where the output currents are transformed into corresponding voltage values.

III. Analog Activation Functions

Having explored the matrix multiplication aspect of vector matrix multiplier crossbar arrays, it is essential to delve into the fundamental activation functions that play a pivotal role in the functionality of neural networks. The three primary activation functions – Sigmoid/tanh (hyperbolic tangent), ReLU (Rectified Linear Unit), and softmax – serve as building blocks for a wide range of neural network architectures. By designing these activation functions as continuous-value programmable analog circuits, we can achieve high efficiency and seamless integration with the crossbar arrays of subsequent layers.

1) Sigmoid: The primary distinction between a sigmoid function and a tanh function lies in their output range; while the tanh function yields outputs between -1 and 1, the sigmoid function generates strictly positive values. To minimize hardware complexity, the sigmoid function was chosen for implementation. As depicted in Fig 2b, the differential input of an Operational Transconductance Amplifier (OTA) facilitates the creation of exponential swings between two voltage rails. The bias of the OTA is governed by an FG FET, which is programmed to produce sufficient current to elevate the output node to the upper rail. Conversely, the lower rail is determined by the FG FET acting as an external bias for the circuit. By carefully balancing the current provided by the FET with the current sunk by the OTA, the lower rail can be adjusted to align with the designer's intent. Finally, the resulting current is converted to voltage across a diode-connected p-type FET. The voltage rails are important because these activation functions feed the subsequent crossbar layer which is the source input

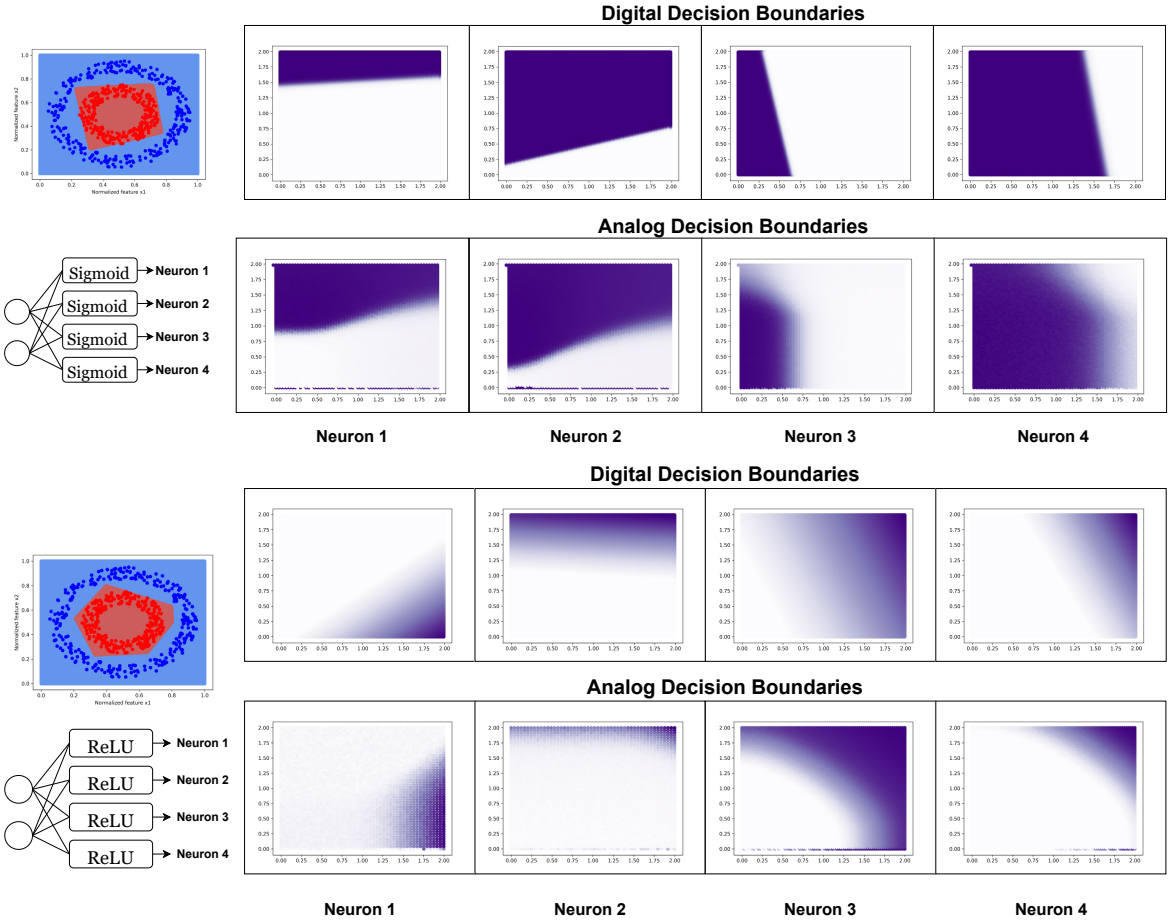

Fig. 3: Measured analog decision boundaries (hyperplanes) of individual neurons and how they compare to their digital counterparts for both a sigmoid and ReLU activation network

of an FG pfet. The output must fall within the active region and tunable rails allow that to happen.

2) Rectified Linear Unit (ReLU): The ReLU circuit operates on a concept similar to the sigmoid function, but without the need for an OTA. Instead, an external FG p-type FET establishes a programmable threshold current. Consequently, any current below this threshold leads to an insignificant shift in the output voltage (Fig 2c). When the summation from the VMM row surpasses this threshold, the excess current is linearly converted to voltage across the diode-connected p-type FET. For the differential operation of the circuit, a negative row connecting to the same column vector input feeds a current mirror that will uphold kirchoffs current law (KCL) and subtract that amount of current from the total. This straightforward mechanism enables the ReLU function to effectively introduce nonlinearity into the neural network while maintaining low hardware complexity and power consumption.

3) Winner-Take-All (WTA) as Softmax: The WTA circuit, originally introduced by Lazzaro [5], aimed to approximate the lateral inhibition observed in neurons, wherein the firing of a specific neuron suppresses the activity of surrounding neurons. This behavior is well-suited for the softmax operation, which involves an exponential function applied to a set of input values, divided by the sum of the exponentials. Softmax serves to amplify the differences between inputs and normalize them to a consistent range. The circuit shown in Fig 2d is a slightly modified version of Lazzaro's original WTA circuit, retaining a similar functionality by converting input currents to gate voltages in the differential pair. The lower bias is programmed to sink just enough current for a single branch, ensuring that the stronger input captures the majority of the current and causes the output voltage to drop. This phenomenon can be observed in Figure 2d; when both inputs are high, the output voltage of the higher input current decreases significantly. This circuit can be extended to N-outputs by simply stacking the branches in parallel. A digital interface is easily integrated by placing an inverter at the output of the branch for a sharper decision boundary.

IV. Neural Network Integration

To demonstrate the capabilities of our hardware, we selected the well-known concentric circles classification problem and trained two neural networks to tackle the task. Fig. 3 presents a side-by-side comparison of the computed digital decision

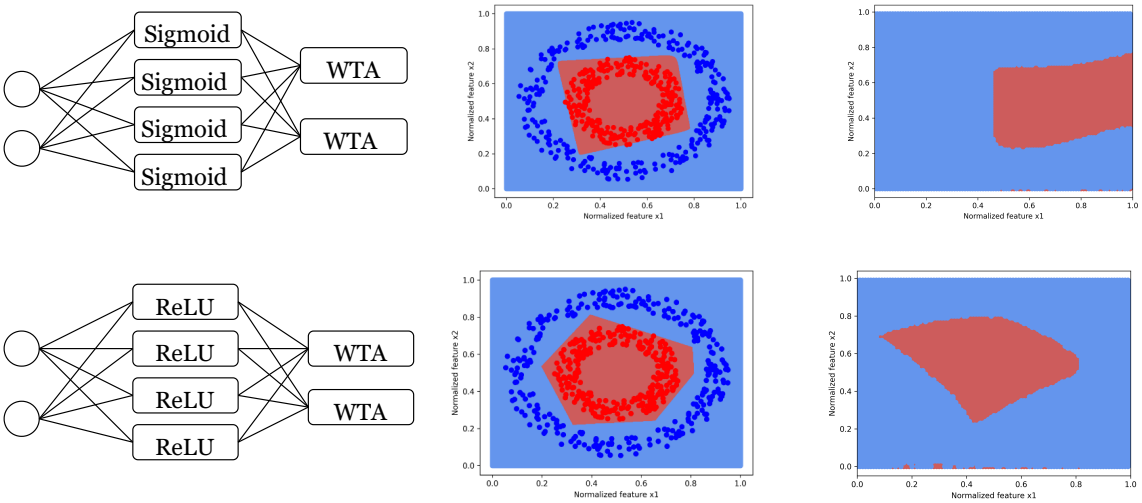

Fig. 4: Shows a linear layer with a sigmoid/ReLU activation feeding another linear layer with a winner-take-all circuit for classification in the left column. The digitally trained implementation is in the middle and measured analog decision boundaries on the right.

boundaries implemented by each individual neuron and their corresponding measured analog counterparts. The ReLU implementation was relatively straightforward and necessitated a shorter training time. Conversely, the sigmoid demanded more extended training epochs and mathematical modeling of the circuit implemented as a custom activation function.

The training process was carried out offline using the PyTorch library. Although online training is feasible, it merits a separate, comprehensive discussion to thoroughly examine its complexities. Following training, the weights of both networks were mapped and fine-tuned to accommodate the hardware.

Challenges were encountered during the offline training process. In addition to modeling the activation function, the exponential function preceding the multiplication step had to be accounted for. This was achieved by multiplying the weights in each layer to exceed the trained value, thereby allowing the exponential attenuation by source voltage to yield a correctly weighted output. The transistors utilized in the VMM also required calibration during programming to maintain accuracy across a range of currents. For further details on FG programming, see [6]. Notably, the exponential functions inherent in sigmoid and softmax activation functions, typically computationally expensive in digital implementations, are essentially cost-free in analog circuits. This efficiency is further augmented by the system's programmability, enabling it to address mismatch and designer intent.

Fig. 4 displays the fully integrated networks and their resulting classifications. The Sigmoid network encountered difficulties in generating the correct boundary wall properly. In contrast, the ReLU network successfully replicated the encapsulating shape of the digital implementation. The power consumption of the sigmoid network amounted to 20μW, while that of the ReLU network reached 80μW.

V. Conclusion

This work has delved into the utilization of floating gate FETs as primitives for subthreshold computation, highlighting their potential for constructing neural network activation functions. Furthermore, it demonstrated the implementation of two integrated networks by solving the concentric circles classification problem end-to-end using analog circuits. This exploration underscores the significant energy efficiency gains that can be achieved by maintaining an end-to-end analog signal chain, paving the way for future advancements in neural network design and the broader field of analog computation.

References

[1] R. Chawla, A. Bandyopadhyay, V. Srinivasan, and P. Hasler, "A 531 nW/MHz, 128x32 current-mode programmable analog vector-matrix multiplier with over two decades of linearity," in *CICC*, 2004, p. 651.

[2] S. Kim, J. Hasler, and S. George, "Integrated floating-gate programming environment for system-level ics," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 6, pp. 2244–2252, 2016.

[3] C. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley, 1989.

[4] C. Enz, F. Krummenacher, and E. Vittoz, "An analytical mos transistor model valid in all regions of operation and dedicated to low-voltage and low-current applications," 01 1995.

[5] J. Lazzaro, "Winner take all networks of o(n) complexity." [Online]. Available: https://authors.library.caltech.edu/52787/1/151-winner-take-all-networks-of-on-complexity.pdf

[6] S. Kim, S. Shah, and J. Hasler, "Calibration of floating-gate soc fpaa system," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 25, no. 9, pp. 2649–2657, 2017.