Implementing Analog Artificial Neural Networks for Enhanced Energy Efficiency and Speed in Machine Learning Applications

Sabri My Abdelouahed
Computer science department
Faculty of sciences, University Sidi Mohamed Ben Abdallah
Fes, Morocco
abdelouahed.sabri@usmba.ac.ma

El Hafci Mohamed
Computer science department
Faculty of sciences, University Sidi Mohamed Ben Abdallah
Fes, Morocco
mohamed.elhafci@usmba.ac.ma

Aarab Abdellah
Computer science department
Faculty of sciences, University Sidi Mohamed Ben Abdallah
Fes, Morocco
abdellah.aarab@usmba.ac.ma

2024 International Conference on Intelligent Systems and Computer Vision (ISCV) | 979-8-3503-5018-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ISCV60512.2024.10620095

***Abstract*— This paper presents the development and simulation of an analog artificial neural network (ANN) aimed at improving energy efficiency and processing speed for machine learning tasks. We detail the architectural design of the ANN, encompassing input, hidden, and output layers, along with the incorporation of resistors and operational amplifiers as key components. Through hardware implementation, the ANN utilizes continuous voltage generators for input representation and resistance values for storing weights and biases. Our simulations demonstrate the model's accuracy and minimal error margins compared to expected outcomes. Additionally, we explore the implementation of activation functions such as ReLU and Sigmoid using specific components to further minimize energy consumption, showcasing the potential of analog ANNs in future machine learning hardware.**

Keywords—Analog Artificial Neural Networks, ANN, ReLU, Sigmoid, Transistor, Intensity

I. INTRODUCTION (*HEADING 1*)

In the rapidly evolving field of machine learning and artificial intelligence, the quest for more efficient computational models has become increasingly crucial. Traditional digital computation, while versatile and powerful, often struggles with the high energy consumption and slower processing speeds when dealing with complex algorithms and large datasets [1]. This challenge has spurred interest in alternative computing paradigms, among which analog artificial neural networks (ANNs) stand out as a promising solution.

Analog computation, leveraging continuous signal processing, offers a pathway to significantly reduce energy consumption while accelerating data processing tasks. The inherent parallelism and energy efficiency of analog circuits align well with the operational principles of neural networks, making them an ideal candidate for hardware implementation of machine learning models [2]. This paper delves into the development and simulation of an analog ANN designed for machine learning applications, aiming to harness these advantages.

Our exploration begins with a brief overview of the fundamental differences between analog and digital computation, particularly in the context of neural network implementation. We then present the architecture of our proposed analog ANN, which includes innovative approaches to represent inputs, compute weights, and apply activation functions using analog components. The motivation behind this research is rooted in the growing demand for more efficient AI applications, where reducing power consumption and improving computational speed are paramount. By offering a detailed simulation study, we aim to demonstrate the feasibility and benefits of analog ANNs, paving the way for their broader adoption in future machine learning hardware solutions.

The development of this analog ANN model reflects a broader shift towards energy-efficient and high-speed computing methodologies. As we push the boundaries of what's possible with analog circuits, this paper seeks to contribute to the ongoing discourse on the role of analog computation in the next generation of machine learning technologies.

II. BACKGROUND AND RELATED WORK

The exploration of analog computing in the domain of artificial intelligence is not a novel concept but has gained renewed interest due to modern technological advances and the increasing demands for energy efficiency and processing speed in AI applications [3]. Analog computation, characterized by its use of continuous signals for processing information, presents a contrasting approach to the binary nature of digital computation. This section reviews the evolution of analog computing, its integration into neural network models, and the current state of research in this area, highlighting both the historical context and recent advancements.

A. Historical Perspective

The idea of analog computing dates back to the early 20th century, with the development of mechanical and electronic analog computers used for a variety of scientific and military applications [4]. Despite the shift towards digital computing in the latter half of the century, the principles of analog computation have remained relevant, particularly in specialized applications where the nuances of continuous data processing can be leveraged for more naturalistic and efficient computation.

Many implementations of image processing have been realized on hardware systems such as Field Programmable Gate Array (FPGA) that demonstrate their efficiency in term of precision, processing speed and energy efficiency [5][6]

B. Analog Computation in Neural Networks

In the context of neural networks, analog computation offers a potential solution to some of the inherent limitations faced by digital systems, such as high-power consumption and

latency in data processing [2][4]. The parallel nature of analog circuits, coupled with their ability to handle continuous signals, aligns well with the operational requirements of neural networks, which often process complex, multidimensional data. Recent research has focused on harnessing these properties to develop analog neural network models that can operate more efficiently than their digital counterparts. Studies have shown that analog ANNs can achieve significant reductions in power consumption while maintaining competitive accuracy levels in tasks such as image recognition, signal processing, and pattern recognition [7].

C. Advancements in Analog ANN Technologies

The resurgence of interest in analog ANNs has been fueled by advancements in semiconductor technologies and fabrication techniques, allowing for more precise and reliable analog circuits [8]. Innovations such as floating-gate transistors, memristors, and other non-volatile memory technologies have opened new pathways for implementing the synaptic weights and activation functions critical to neural network operations in an analog format. Furthermore, research into hybrid systems that combine the best aspects of analog and digital computing presents a promising avenue for overcoming the scalability and variability challenges traditionally associated with analog circuits [9]. These hybrid models aim to leverage the energy efficiency and speed of analog processing for certain computations while relying on digital components for tasks requiring high precision and flexibility.

D. Challenges and Future Directions

Despite the potential advantages, the development of analog ANNs faces several challenges, including issues related to the variability of analog components, the difficulty of precisely controlling and updating synaptic weights, and the integration of analog circuits with existing digital infrastructure. Ongoing research efforts are directed towards addressing these challenges, with a focus on developing more robust, scalable, and adaptable analog neural network architectures. The exploration of new materials, circuit designs, and computational models, alongside advances in machine learning algorithms, holds the promise of further enhancing the performance and applicability of analog ANNs. As the field progresses, it is expected that analog neural networks will play an increasingly significant role in the next generation of energy-efficient and high-speed AI systems.

III. System Architecture and Design

The analog artificial neural network (ANN) presented in this work is structured around a simplified yet effective architecture designed to exploit the advantages of analog computation. The architecture consists of three main layers: an input layer, a single hidden layer, and an output layer. This design choice is informed by our objective to demonstrate the feasibility and efficiency of analog computation in executing neural network tasks, with a focus on optimizing for both energy consumption and processing speed.

The simulated Artificial Neural Network (ANN) is presented in the figure Fig. 1.

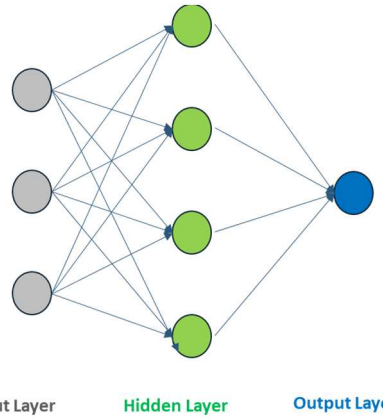

Fig. 1. The simulated Artificial Neural Network (ANN)

Training and validation of our analog artificial neural network (ANN) were conducted in an offline environment, utilizing computational resources to optimize the network's weights and biases before their application in the analog model. This approach allows for the thorough and efficient calibration of the network, ensuring that it is finely tuned to perform its designated tasks with high accuracy and reliability.

- **Offline Training Process:** The network's training was carried out using a digital simulation on a computer, where a dataset representative of the problem space was used to iteratively adjust the network's synaptic weights and biases. This process employed backpropagation and gradient descent algorithms to minimize the error between the network's outputs and the expected outcomes. By performing these computations in a digital environment, we could leverage the computational power and precision of digital processors, facilitating a more controlled and flexible training phase.
- **Application of Weights and Biases:** Upon the completion of the training phase, the optimized weights and biases were directly translated into the analog domain. The weights were implemented using programmable resistors, while biases were implemented using a DC voltage in series with a resistor. This direct application method ensures that the analog ANN accurately reflects the learned parameters, allowing it to perform as intended based on the training it received.

Our analog ANN model incorporates a total of three layers, each serving a distinct function within the neural network:

- **Input Layer**: Comprising 3 neurons, the input layer is tasked with receiving the analog signals that represent the data to be processed. These signals are then forwarded to the hidden layer. The use of three neurons corresponds to the dimensionality of the input data, ensuring that each input feature can be directly mapped to a neuron.
- **Hidden Layer**: The hidden layer, consisting of 4 neurons, plays a crucial role in the network's ability to learn and model complex relationships. Each neuron in this layer is connected to all neurons in the input layer, allowing it to synthesize the input signals in a manner that captures the underlying patterns and

dependencies. The decision to use a single hidden layer with four neurons strikes a balance between computational simplicity and the network's learning capability.

- **Output Layer**: The network culminates in an output layer with a single neuron, responsible for producing the final output of the ANN. This output represents the network's prediction or decision based on the input data it has processed.

The analog ANN network scheme is presented in the figure Fig. 2.

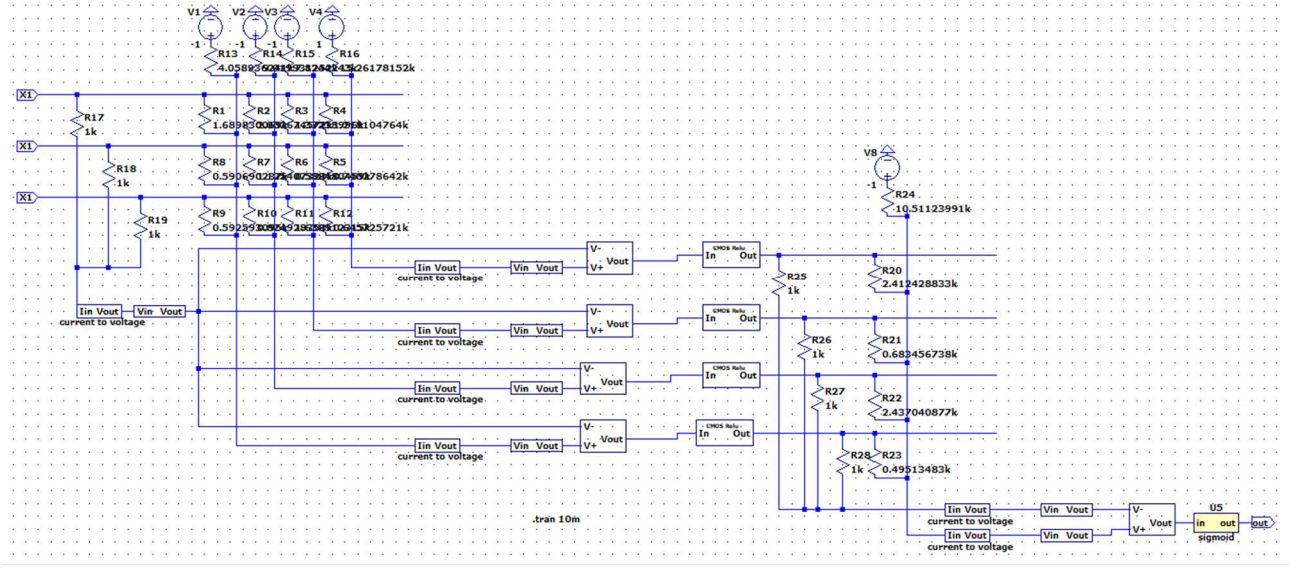

Fig. 2. The analog ANN architecture

Analog circuits simulate activation functions to introduce necessary non-linearity. Specifically, the network utilizes CMOS Technology for ReLU functions in the hidden layer, allowing it to effectively capture complex data patterns in an energy-efficient manner (Fig.3). The sigmoid activation function used in the output layer was simulated using Spice code.

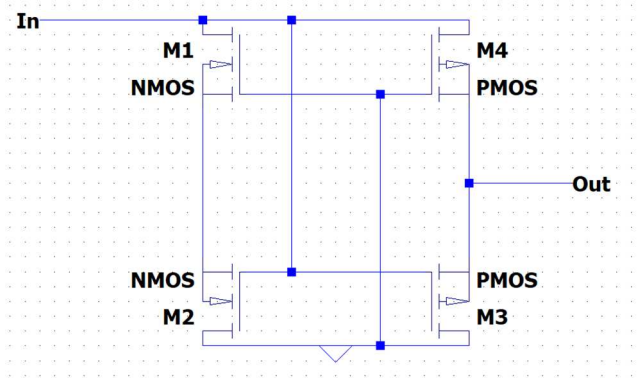

Fig. 3. Analog ReLU activation function scheme

IV. METHODOLGY

The methodology behind the development, training, and validation of our analog artificial neural network (ANN) model is designed to ensure that the model is both accurate in its predictions and efficient in its operation. This section outlines the comprehensive approach taken from the model's conceptualization to its final implementation and testing.

Our focus is developing an analog ANN to tackle a regression task characterized by the equation:

$$y = f(X) = W*X + b \quad (1)$$

where X denotes the input vector (x1, x2, x3), W is the layers weights, and b the layers biases.

This model encapsulates the network's ability to predict continuous values, demonstrating the efficacy of ANNs in mapping complex input patterns to a numerical output. The backbone of our model's training regime is a dataset composed of 1000 entries, each consisting of a three input values and their respective numerical output. This substantial dataset enables the precise adjustment of the network's weights (W) and bias (b), ensuring the model's adeptness at capturing and predicting the underlying trends in the data.

A. Model development

A crucial part of our methodology is the offline training process, which leverages the computational advantages of digital systems for the precise optimization of the model's parameters.

The training involved adjusting the synaptic weights and biases using algorithms like backpropagation and gradient descent to minimize the error between the network's outputs and the expected results. This approach allowed for a detailed and controlled optimization process, essential for the success of the analog implementation.

- **Data Preparation:** Prior to training, the dataset was preprocessed to ensure compatibility with the analog model's input requirements. This included adapting the data to match the voltage levels used in the analog inputs.
- **Parameter Optimization:** The training process involved iterative adjustments to the model's parameters to achieve optimal performance. This was facilitated by a simulation environment that closely mimicked the analog network's behavior.

- **Transfer to Analog Domain:** After the training phase, the optimized parameters were translated into their analog equivalents. Weights were implemented using programmable resistors, while biases were set using voltage offsets.

B. Analog Implementation

Following the offline training, the next step was the physical implementation of the trained model in the analog domain. This involved setting up the analog circuitry to reflect the trained network's structure and parameters accurately. Each neuron's behavior was emulated using resistors crossbar and operational amplifiers. Where weights are represented as resistor values, operational amplifiers are used to perform various necessary operations on the electrical signal output from the cell.

The output of each neuron is represented by the current I, where :

$$I = V_w x \frac{1}{R_w} + V_b x \frac{1}{R_b} \quad (2)$$

The weight values generated by the training are not directly implemented in the circuit, but go through two transformations. The first is a positive shift and the second is the inversion of the values to obtain the corresponding value of the resistors.

As far as the biases are concerned, we carry out a single transformation by inverting the bias values to obtain the resistor values. For the negative bias, we use a DC voltage generator with a value of -1 volts (-b=(-1)/R) [10].

C. Model Validation and Testing

The final phase of our methodology focused on validating and testing the analog ANN model to ensure its accuracy and efficiency. Using a separate dataset, the model's performance was evaluated to assess its generalization capabilities and accuracy in new, unseen scenarios. We will use the same inputs for both analog and digital prediction to compare them and especially the effectiveness of the analog calculation.

V. Implementation and Simulation Results

In this work, we have successfully simulated an analog artificial neural network (ANN) architecture, targeting hardware acceleration for Machine Learning applications.

A. Implementation

Our approach utilizes DC voltage generators to represent inputs, capturing the nuances of real-world data in an efficient analog format. Weight and bias parameters are ingeniously stored as resistor values within the circuit, enabling dynamic adjustment essential for the training and operation of the ANN.

To address the inherent challenge of incorporating negative weight values in analog circuits, we devised a novel solution by translating the weight values in each layer. This is achieved by adding a constant value to all weights, thus eliminating negative values without compromising the model's integrity. Post-computation and prior to the activation function, the constant is subtracted to preserve the original behavior of our model. This translation technique is pivotal, ensuring the analog system accurately reflects the computational model.

The selection of the translation constant is critical to the system's performance; in our implementation, we settled on a value of 1, equivalent to 1 KOhm, guided by the average weight values across the network. This choice is instrumental in maintaining the balance between operational integrity and the practicalities of hardware implementation, highlighting our commitment to developing a robust and efficient analog computing solution for machine learning acceleration.

The implementation of the ReLU activation function can be achieved using operational amplifier-based circuits. However, we opt for a CMOS technology-based implementation utilizing four transistors (two PMOS and two NMOS) to reduce energy consumption in our circuit [11]. As for the sigmoid activation function, we implement it using Spice code, even though an implementation using operational amplifiers or MOSFET transistors is feasible. However, these approaches fail to replicate the sigmoid function's behavior accurately. In our circuit, we also employ two additional blocks: a current-to-voltage converter (CVC) to convert the current output from the resistors into a voltage used by other parts of the circuit, and an inverting amplifier, since the current-to-voltage converter generates a low inverse voltage. Finally, we perform a transient simulation to observe the output of our ANN. The simulation yields a voltage value in millivolts, aligning with the values obtained via the Python API with an error on the order of 10^{-6}.

B. Simulation results

After simulating the ANN network, we attempted to observe the behavior of our analog circuit based on 3 simulations. The following table presents the input values, the desired output, the simulated output, and the error. In the first part of the simulation, we used a function written in Spice code to simulate the ReLU and Sigmoid activation functions.

TABLE I. Results of ANN Simulations Using Spice Code for ReLU and Sigmoid Functions

N°	Input	Desired Output	Simulated Output (mV)	Error
1	0.874071773	0.4556881487369	455.68922	1.07E-06
	0.249668953			
	0.263247347			
2	0.000653390	0.6063608527183	606.35519	5.66E-06
	0.870669034			
	0.792594813			
3	0.627328935	0.4828004539012	482.80108	6.26E-07
	0.750265764			
	0.152048567			

In the second part of the simulation, we simulated the ReLU function with CMOS analog components and the Sigmoid function in Spice code.

TABLE II. Performance Comparison of ANN Simulations with CMOS Components for ReLU and Spice Code for Sigmoid Function

N°	Input	Desired Output	Simulated Output (mV)	Error
1	0.874071773	0.4556881487369	476.21387	2.05E-02
	0.249668953			
	0.263247347			
2	0.000653390	0.6063608527183	513.68099	9.27E-02
	0.870669034			
	0.792594813			
3	0.627328935	0.4828004539012	476.16714	6.63E-03
	0.750265764			
	0.152048567			

These simulations help illustrate the effectiveness and accuracy of our ANN's analog implementation in processing and predicting outcomes based on input data. The two approaches to simulating activation functions—Spice code for ReLU and Sigmoid, and CMOS components for ReLU—show distinct performance characteristics, as highlighted by the results and errors recorded in each scenario.

C. Simulation analysis

The simulation results from the artificial neural network (ANN) implemented with analog components reveal significant insights into the model's performance and accuracy. In the initial simulations, which utilized Spice code for both ReLU and Sigmoid activation functions, the ANN demonstrated exceptional precision, as indicated by the minimal errors observed across all three simulations. The errors were remarkably low, on the order of 10^{-6} to 10^{-7}, suggesting that the model's output closely aligns with the desired values. This high level of accuracy highlights the effectiveness of using Spice code to simulate activation functions in an analog ANN context.

In contrast, the second set of simulations, which employed CMOS components for the ReLU function and Spice code for the Sigmoid function, showed a noticeable increase in error rates. Although the errors remained relatively small, they were significantly higher than those observed in the first set of simulations, reaching up to 9.27E-02 in one instance. This increase suggests that while the CMOS-based implementation of ReLU is innovative and energy-efficient, it may not achieve the same level of precision as the purely Spice code-based approach.

The two simulation approaches underscore the trade-offs between energy efficiency and accuracy in analog ANN implementations. The use of CMOS components for activation functions like ReLU is promising for reducing energy consumption, a critical consideration for hardware acceleration in machine learning. However, this may come at the cost of slightly reduced precision, as demonstrated by the comparative analysis of simulation results.

Overall, these simulations provide valuable insights into the potential and challenges of analog ANNs for machine learning applications. The choice between Spice code simulation and CMOS components for activation functions will depend on the specific requirements of the application, including the need for precision versus energy efficiency. Future work could explore optimizing the CMOS-based implementation to improve accuracy while maintaining its energy-saving benefits.

VI. Conclusion

This investigation into the implementation of an analog artificial neural network (ANN) for machine learning has highlighted the potential and challenges of analog computation. Our study focused on simulating the ANN's performance, emphasizing how inputs are represented, and how weight and bias values are intricately stored and manipulated within the circuit, using analog components like CMOS transistors for activation functions and operational amplifiers for signal processing.

The results from our simulations, particularly those concerning the handling of weights, biases, inputs, and outputs, underscore the delicate balance required to achieve high accuracy and efficiency in analog ANNs. By managing negative weight values through value translation and utilizing DC voltage generators for input representation, we demonstrated the model's capability to closely mimic desired outputs with minimal error margins. These simulations not only affirmed the precision of our analog model but also illuminated the critical role of component selection and circuit design in optimizing ANN performance.

Reflecting on the broader implications, our work suggests that analog ANNs can offer significant advantages in terms of energy efficiency and processing speed. However, achieving these benefits without compromising accuracy necessitates innovative solutions, such as the ones we've explored for weight and bias management. This research contributes to a deeper understanding of the operational dynamics of analog ANNs and lays a foundation for future explorations aimed at enhancing their practicality for machine learning applications.

Future research directions should aim at refining the implementation of weights and biases in analog circuits, further reducing the error in input-output simulations, and exploring the scalability of these models for more complex tasks. Additionally, investigating the integration of analog and digital components could provide a hybrid solution that leverages the strengths of both realms, potentially leading to groundbreaking advancements in hardware acceleration for machine learning.

References

[1] Ayari, A., Mechergui, H., & Haddouk, A. (2013, March). Active power measurement comparison between analog and digital methods. In 2013 International Conference on Electrical Engineering and Software Applications (pp. 1-6). IEEE.

[2] Moons, B., & Verhelst, M. (2014). Energy-efficiency and accuracy of stochastic computing circuits in emerging technologies. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 4(4), 475-486.

[3] Wu, J., Lin, X., Guo, Y., Liu, J., Fang, L., Jiao, S., & Dai, Q. (2022). Analog optical computing for artificial intelligence. Engineering, 10, 133-145.

[4] Small, J. S. (1994). The Analogue Alternative: A Socio-Economic History of the Analogue Computer in Britain and the USA, 1930-1975. The University of Manchester (United Kingdom).

[5] El Khoukhi H., Idriss F.M., Yahyaouy A., Sabri M.A. (2020) An Efficiency Study of Adaptive Median Filtering for Image Denoising, Based on a Hardware Implementation. In: Bhateja V., Satapathy S., Satori H. (eds) Embedded Systems and Artificial Intelligence. Advances in Intelligent Systems and Computing, vol 1076. Springer, Singapore. https://doi.org/10.1007/978-981-15-0947-6_9

[6] Hasnae El Khoukhi; My Abdelouahed Sabri (2018). Comparative study between HDLs simulation and Matlab for image processing. 2018 International Conference on Intelligent Systems and Computer Vision (ISCV). Pages: 1-4. ISBN: 978-1-5386-4396-9. DOI: 10.1109/ISACV.2018.8354046. 2-4 April 2018. Fez, Morocco.

[7] MacLennan, B. J. (2007). A review of analog computing. Department of Electrical Engineering & Computer Science, University of Tennessee, Technical Report UT-CS-07-601 (September), 19798-19807.

[8] Ma, S., Wu, T., Chen, X., Wang, Y., Tang, H., Yao, Y., ... & Bao, W. (2022). An artificial neural network chip based on two-dimensional semiconductor. *Science bulletin*, *67*(3), 270-277.

[9] Houshmand, P., Sarda, G. M., Jain, V., Ueyoshi, K., Papistas, I. A., Shi, M., ... & Verhelst, M. (2022). Diana: An end-to-end hybrid digital and analog neural network soc for the edge. IEEE Journal of Solid-State Circuits, 58(1), 203-215.

[10] Han, G., Lee, C., Lee, JE. et al. Alternative negative weight for simpler hardware implementation of synapse device based neuromorphic

system. Sci Rep 11, 23198 (2021). https://doi.org/10.1038/s41598-021-02176-4

[11] Huang, Y., Yang, Z., Zhu, J., & Ye, T. T. (2020, September). Analog circuit implementation of neurons with multiply-accumulate and relu functions. In Proceedings of the 2020 on Great Lakes Symposium on VLSI (pp. 493-498).