Vol. 08, No. 1 (2026) 341-350, doi: 10.24874/PES08.01A.007

Proceedings on Engineering Sciences

www.pesjournal.net

SUBSYSTEM DESIGN FOR NEUROMORPHIC COMPUTING

Karpagam T V
Jayanth D
Anita Angeline[1]

Received 06.04.2025
Revised 08.06.2025
Accepted 13.07.2025

Keywords:

Activation Function, ReLU Function, Hyperbolic Tangent (tanh), Feature Extraction, Backpropagation

A B S T R A C T

This article involves the circuit design of Convolutional Neural Network (CNN) activation functions, crucial components of deep learning architectures, alongside Spike-Timing-Dependent Plasticity (STDP) circuitry. These activation functions play a vital role in network performance, and the integration of STDP enhances learning dynamics. By mimicking biological synaptic behavior, STDP facilitates unsupervised learning and improves adaptability to input variations. Various activation functions, including sigmoid, tanh, ReLU (Rectified Linear Unit), and linear, were implemented, revealing varying power consumption levels. Sigmoid exhibited the highest power consumption at 65.33uW, while linear had the lowest at 2.491uW. Integrating STDP into the activation function architecture enhances the CNN's ability to efficiently learn from input patterns, following a biologically inspired approach. The overarching goal is to enhance learning dynamics, robustness, low latency, and reduce hardware requirements for both STDP and activation functions within the CNN. Simulations were conducted using Cadence® Virtuoso ADEL environments with 90nm technology node library.

1. INTRODUCTION

Gaining insight into the mechanisms that govern synaptic plasticity and activation functions inside artificial neural networks (ANNs) is essential for solving complex problems related to learning and information processing. Out of these mechanisms, Spike Timing Dependent Plasticity (STDP) stands out as a fundamental concept inherited from the behaviour of biological synapses. Activation functions work as a nonlinear modification that enhances the expressiveness and complexity of neural network models simultaneously. This introduction delves into the intricate connection between STDP (Spike-Timing-Dependent Plasticity) and activation functions in the realm of neural network learning and computing. It examines the roles, relationships, and outcomes of this complicated relationship.

The temporal relationship between pre-synaptic spikes, which are input signals, and post-synaptic spikes, which are output responses, in biological brain networks is the basis for the synaptic plasticity rule called STDP. It implies that the timing and magnitude of changes in synaptic weights are influenced by the time interval between these spikes, which subsequently impacts the intensity of connections between neurons. The principle underlying STDP is that neurons that activate

[1] Corresponding author: Anita Angeline A
Email: anitaangeline.a@vit.ac.in

simultaneously establish connections with one other, while neurons that activate asynchronously lose their connection (Acciarito et al., 2016). This concept effectively describes how the brain may modify and restructure its neural connections in response to experiences. The underlying concept of STDP is derived from Hebbian learning, initially proposed by Donald Hebb in 1949. According to this theory, synapses develop stronger when pre-synaptic neuron often triggers post-synaptic neuron to fire. Conversely, asynchronous firing patterns result in the degradation of synapses. By adding accurate timing data, STDP improves on Hebbian learning by permitting fine-grained adjustments dependent on the relative timing of post and pre-synaptic spikes.

Artificial neural networks use activation functions as nonlinear alterations on the neuronal input signals. They provide neural network models with the essential nonlinearity, allowing them to understand and convey complicated relationships present in the data. The output of neurons is largely influenced by activation functions, which also affect the network's capacity to predict difficult functions and extract meaningful information from the input data. Over time, a variety of activation functions have been suggested, each with unique computing capabilities and features. The rectified linear unit (ReLU) and its derivatives, the hyperbolic tangent (tanh), and the sigmoid function are a few of the most regularly utilized activation functions. The training dynamics, convergence characteristics, and generalization capacities of the network are affected by saturating or non-saturating nonlinearities.

For instance, during back-propagation, activation functions with saturating nonlinearities, such tanh and sigmoid, may create vanishing gradients, which can induce gradient instability and delayed convergence. On the other hand, non-saturating activation functions such as ReLU provide more stable and effective learning dynamics by eliminating the vanishing gradient issue and permitting faster convergence.

The nonlinear features of the activation function impact synaptic plasticity, which is one component of this interaction. Post-synaptic potentials (PSPs) in response to pre-synaptic activity are controlled by the distribution and amplitude of distinct activation functions, which reveal unique input-output mappings. Since STDP depends on these PSPs' exact timing, the network's stability and learning dynamics can be altered by the activation function utilized.

Neural network learning and computation will be strongly impacted by the combination of STDP and activation functions. Determining how these systems interact is key to understanding neural network computational capabilities and learning dynamics. STDP and activation functions interact with one other in ways that go beyond synaptic plasticity to include network dynamics and information processing. The network's response characteristics are determined by the nonlinear transformations brought about by activation functions, which have an impact on the network's capacity to encode and transmit information. Consequently, the network's connection patterns are transformed by STDP-mediated plasticity, which boosts the network's computational capacity by utilizing prior data and input statistics.

1.1 Implication and Future

Neural network learning techniques and designs may be advanced in new ways thanks to the synergy between STDP and activation functions (Chaturvedi & Kurshid, 2015). Through the exploitation of the relationship between nonlinear activation and synaptic plasticity, scientists may construct novel learning rules and network designs that may efficiently capture and utilize the underlying structure of complicated information.

Subsequent investigations can concentrate on explaining the foundations impacting the relationship between STDP and activation functions in various neural network topologies and learning assignments (Cruz et al., 2011; Yildiz et al., 2019). Furthermore, exploring biologically inspired learning mechanisms and putting them into artificial neural networks may result in learning algorithms that are more effective and adaptive, reducing the gap between artificial and biological intelligence.

In summary, the combination of STDP and activation functions offers a viable way to further expand our comprehension of neural network computation and learning. Researchers can obtain new insights into the principles of learning and information processing in biological and artificial neural networks by figuring out the complex interactions between synaptic plasticity and nonlinear transformations. This will open the door to the creation of more effective, adaptable, and intelligent systems.

2. LITERATURE REVIEW

In various literatures, the activation function circuit is segmented into three components: the positive half-axis calculation circuit, negative half-axis mirror circuit, and error compensation unit is synthesized in TSMC 180nm technology within a small area at high frequency, showing significant improvements in precision and area utilization (Liu et al., 2021). This circuit is capable of computing both sigmoid and tanh activation function. The switching is achieved by use of a multiplexer. Experimental results demonstrate almost no accuracy loss.

The work of Shakiba & Zhou (2020) proposed a design that divides the hyperbolic tangent function into two subcircuits for the positive and negative halves. It utilizes a P-type metal oxide semiconductor (PMOS) transistor and a resistor to approximate the mathematical Tanh activation function effectively. The study highlights the compatibility of the design with recent memristive neural network architectures and validation methods for activation function neuron design. This paper is aimed at implementing the Tanh exponential activation function on FPGA boards, particularly the Artix-7 and Zynq-7000. It outlines the use of piecewise linear and quadratic polynomial approximations, alongside IEEE754 2008 floating-point representation, to evaluate efficiency. Two methods of approximation are compared for speed, accuracy, and hardware resource usage. A hybrid approach is proposed, utilizing quadratic approximation in one interval and linear approximation elsewhere, aiming to balance resource utilization and speed. The implementation includes a multilayer neural network utilizing this combined approximation.

The work of Yildiz et al. (2019) is a digital circuit design in Figure 1. technique for hyperbolic tangent sigmoid functions—a popular activation function for neural networks—is presented (Hajduk et al., 2018; Lin et al., 2008). The design strategy for such a nonlinear function is to first approximate the function of its first-order derivative with piece-wise linear functions, and then use a digital circuit to integrate the approximated function of the first-order derivative to obtain an estimate of the original function. In the software simulation, the suggested approximation approach's average error and maximum error are in the range of 10^-3 and 10^-2, respectively. With the use of resource sharing, the suggested method's hardware implementation only needs one addition/subtraction and one multiplication ALU. A neural network has verified the circuit's functionality for a system.

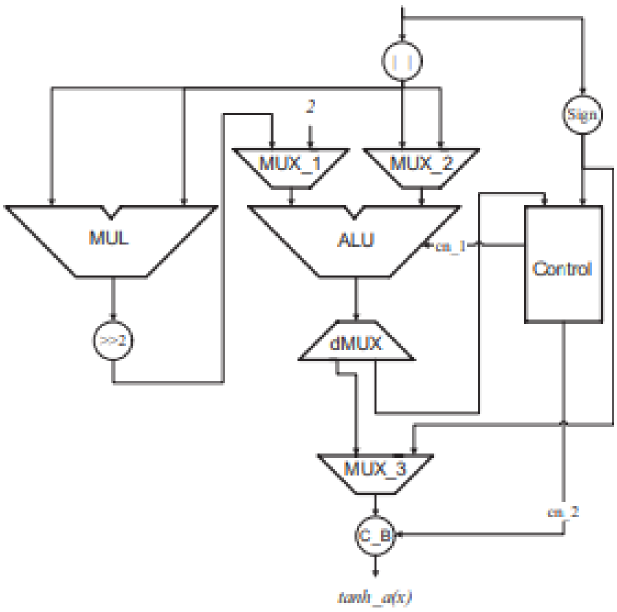

Figure 1. ALU

The work of Lin et al. (2008) employs a dendritic spine-like topology to represent weighting coefficients and utilize a dual structure neural network to segregate signal values into positive and negative ranges. The simulation results of digit recognition conducted on the standard MNIST database demonstrate the capability of attaining adequate precision through the utilization of 4-bit binary weighting coefficients. These simulations assume ideal memristor elements for the memristive implementation of the neural network. Do not position a heading at the bottom of a column, as this results in the following text beginning at the top of the subsequent page or column.

The work of Chaturvedi & Kurshid (2015) explores the efficacy of third-generation spiking neural networks (SNNs), which utilize individual spikes for increased biological realism. It demonstrates their application in character identification and classification, employing ASIC technology for large-scale simulations of the Izhikevich model while reducing dynamic power through RTL clock gating. Emphasis is placed on lowering system costs and power usage for mass production. The SNN model, executed on an ASIC utilizing a 90 nm process, demonstrates suitability for extensive cortical simulations, attributed to its precision and operational efficiency. MATLAB simulations show high classification accuracy, with an ASIC consuming 2.8 mW power and occupying 120312 µm² area for English character recognition. The study underscores the potential of SNNs for character recognition with custom implementation and reduced power consumption.

The work of Acciarito et al. (2016) introduces a circuit implementation of a memristor-based neural network, featuring a driving circuit model capable of adjusting synaptic strength without specific input pulse shapes. Moreover, the prototype circuit emulates standard Spike Timing Dependent Plasticity (STDP) behavior, enabling controlled synaptic weight changes. Preliminary experimental results validate the effectiveness of the proposed driving circuit. The study proposes an analog VLSI driving circuit for memristor-based synapses to mimic and regulate STDP, validated through simulations of a system with three neurons and two synapses. This approach serves as a foundation for designing more complex neural networks using memristors, with ongoing research focusing on real-world applications.

The work of Cruz et al. (2012) is inspired by the inherent energy efficiency of biological neurons and synapses, energy-efficient integrated circuits that emulate the behavior of brain neurons and synapses are designed. These circuits highlight spike-based communication, which uses distinct electrical spikes to convey information, much like brain neurons do. The circuits use binary-amplitude signals, which simplify the design and lower energy requirements, to achieve low power consumption. Adaptive behaviors, like Spike Timing Dependent Plasticity (STDP), replicate the brain's learning mechanisms by enabling the circuits to modify and learn in response to spike timing. Important elements of the design include voltage-controlled current sources, which regulate current flow based on input voltage, further decreasing power consumption, and hysteresis comparators, which stabilize neural output and guarantee the circuit appropriately responds to spikes. By modeling these biological processes, the circuits operate efficiently while maintaining the ability to learn and adapt in real-time.

The work of Hajduk (2018) research on the hardware implementation of activation functions specifically, sigmoid and hyperbolic tangent(Tanh) functions indicates that using Field-Programmable Gate Arrays (FPGAs) to improve neural network calculations' accuracy and efficiency is becoming more and more popular. Neural networks require activation functions to introduce non-linearities, which allows the networks to learn intricate patterns. For real-time applications, particularly in fields where low-latency processing is required, their accurate and effective computation is crucial. The implementation of these activation functions in hardware has evolved significantly, with recent studies employing Algorithmic State Machine (ASM) diagrams as a systematic approach to design. ASM diagrams provide a visual representation of the sequential logic needed for implementing activation functions, facilitating the translation of high-level algorithms into efficient hardware structures. This structured design process is vital for ensuring that the circuits operate correctly and efficiently. One noteworthy technique covered is approximating exponential functions using McLaurin interpolation, which is essential for computing activation functions such as sigmoid and tanh. Polynomial approximations are able to balance accuracy with computing economy. Exponential functions can be represented as polynomials, which makes hardware calculations easier to handle and quicker. Simulators and tests on FPGA boards are among the stringent verification methods used to confirm the correctness of the implemented functionality. The relevance of accuracy verification in ensuring that hardware implementations meet theoretical expectations is emphasized throughout the text. Previous research on this topic emphasizes the possible differences that might occur between theoretical models and real-world applications, which could have a negative impact on neural network performance. Therefore, before a circuit is really deployed on FPGA hardware, designers may evaluate its performance under various scenarios using simulations. Another important consideration in these systems is the optimization of arithmetic operations. The paper describes a number of performance-enhancing tactics, including the use of floating-point (FP) arithmetic blocks that increase computation accuracy. When working with activation functions that need precise representation of minute input fluctuations, this is very crucial. Furthermore, the use of parallel calculations is emphasized as a way to speed up execution even further and increase system efficiency in general.

The work of Indiveri et al. (2013) focuses on co-localization of memory and computation in artificial neural processing components is the main topic of discussion in this article about the integration of nanoscale memristor synapses in neuromorphic computing systems (Indiveri et al., 2013). It emphasizes how crucial it is to process sensory data quickly by utilizing circuitry with physiologically believable temporal constants. Memory and computation are colocalized in some parts of massively parallel artificial neural processing. For neuromorphic systems to handle real-world sensory signals effectively, circuits with physiologically realistic temporal constants are required. In order to arrange a high number of memristive synapses in a small area, cross-bar arrays are frequently utilized. For long-term synaptic state storage in brain-inspired probabilistic computing systems, memristors provide a compact solution.

Deep learning architectures and activation functions lead to optimization of Hardware Design for the particular application (Li et al., 2016). Prior research has demonstrated the efficacy of different CNN architectures, including ResNet and Inception (GoogLeNet), which use novel techniques like residual connections and multi-scale feature extraction to enhance image classification performance on datasets like MNIST, CIFAR-10, and CIFAR-100. Due to their efficacy and efficiency in many settings, traditional activation functions, namely Rectified Linear Units (ReLU), have dominated the area. However, they have several drawbacks, such as the possibility of gradient saturation and problems with dying neurons. In an effort to address these issues, solutions like Leaky ReLU and Exponential Linear Units (ELU) have been investigated. A new method that expands on these earlier discoveries is the introduction of Hyperbolic Linear Units (HLUs), which provide a mathematically based activation function intended to improve gradient flow and speed up learning in deep CNNs. The research presents empirical data showing that models with HLUs perform much better than those with typical activation functions, suggesting that HLUs have the potential to enhance the accuracy of classification and training efficiency of state-of-the-architecture. This study contributes to the body of knowledge about activation functions and opens up new directions for advancement of deep learning techniques in the future.

The work of Li et.al 2021 outlines the historical development of CNN architectures, highlighting important discoveries and seminal contributions. It follows the path from the research explores the distinct contributions of several CNN designs, highlighting important elements including pooling algorithms that reduce dimensionality without sacrificing important information and convolutional layers that automate feature extraction (Li et al., 2012; Pan et al., 2021). The writers talk on multi-scale processing, especially as it relates to architectures like Inception that improve model performance by capturing characteristics at many sizes. Key obstacles in deep learning, including as computing efficiency and the difficulties of training deeper networks, are addressed by these architectural improvements. Additionally, it demonstrates how well CNNs recognize and localize objects in photos, which has important ramifications for surveillance and autonomous driving systems. CNNs are contributing significantly to medical imaging by using radiological image analysis to help with diagnosis and therapy planning. The writers emphasize how CNNs have improved diagnostic speed and accuracy, revolutionizing medical procedures. the difficulties associated with fine-tuning hyperparameters in machine learning models, highlighting the exponential rise in trials using more hyperparameters. It draws attention to the drawbacks of grid search and offers an easier method for advancing from coarse to fine search. Because the number of tests increases exponentially with the number of hyperparameters, hyperparameter tuning can be difficult. For adjusting many hyperparameters, grid search is not practicable, thus more effective approaches such as switching from coarse to fine search are required. Random search or small-step grid search can be used in a small area identified by coarse grid search to pinpoint the optimal hyperparameters.

The work by Pan et al. (2020) investigates ways for improving the performance of convolutional neural networks (CNNs) using memristor-based hardware by addressing challenges such as limited N state, and asymmetric write nonlinearities. Techniques that boost network performance include limiting weight ranges, carrying out unique update methods, and using several memristors for each kernel piece. The intent of the weight range approach is to make the most use of the memristors' possible conductance states (Nstate). The network can more effectively use the discrete conductance states and achieve higher accuracy by restricting the weight range. The update plan is meant to lessen the impact of nonlinearities in asymmetric writes. Weight updates may become inaccurate due to asymmetric write nonlinearities; thus, this approach helps to balance the updates and enhances overall efficiency. Each Kernel Element has many memristors: The authors recommend utilizing numerous memristors for each kernel element in order to reduce the influence of cycle-to-cycle (C2C) variance. CNN's dependability and accuracy are increased by this redundancy, which helps to average out variations. These techniques may greatly increase memristor-based CNNs' recognition accuracy, bringing them up to 95.25% to 96.81% accuracy levels.

The work Saxena et al. (2017) explains the creation of neuromorphic computing architectures that use cutting-edge memory technologies, such as RRAM, to do cognitive computing tasks that need little energy. RRAM devices are attractive candidates for electronic synapses because they show conductance modification properties comparable to STDP. By eliminating the requirement for external memory transfer and drastically lowering energy consumption, neuromorphic systems may be able to accomplish local, on-chip learning capabilities through the use of this bio-plausible learning method. Saxena presents a hybrid CMOS-RRAM Very-Large-Scale Integration (VLSI) circuit architecture that enables dense integration of artificial neurons and synapses, leveraging the nanometer scaling capabilities of silicon processing technology. Such integration is key to realizing energy-efficient spiking neuromorphic systems, which can operate at ultra-low power levels compared to traditional von Neumann architectures. The document outlines the requirements and obstacles such as non-volatility, high neuro-synaptic density, localized learning algorithms, and ultra-low power operation - for the development of neuromorphic hardware capable of deep learning.

Hybrid CMOS-RRAM VLSI circuits, which use nanoscale silicon processing methods, it has been proposed to enable dense integration of CMOS neurons and future devices for brain-inspired computer processors. The creation of neuromorphic system-on-a-chip (NeuSoC) architectures is anticipated to pave the way for chip-scale form factors that save significant amounts of energy and enable deep learning.

In few literatures, passive resistive-type neuron that may be used as the activation function in artificial neural networks (ANNs) to produce the hyperbolic tangent function M (Shamsi et al., 2015, Zaki et al., 2019). The suggested neuron uses less power because it doesn't require any biasing voltage to function. It is simulated using 180 nm CMOS technology and exhibits low errors and an excellent approximation to the ideal hyperbolic tangent function. Through simulations and performance comparisons with conventional neuron models, the authors show that their suggested model uses a great deal less energy while achieving comparable accuracy levels in tasks like image categorization. The hyperbolic tangent function is intended to be implemented by the passive resistive-type neuron as the activation function in ANN. It uses only 62.5 μW of total power and no standby power because it doesn't need any biasing voltage to function. When the neuron is simulated using 180 nm CMOS technology, the average error and highest error are 6.88% and 19.7%, respectively. This is a decent approximation to the ideal hyperbolic tangent function. Compared to earlier analog hyperbolic tangent constructed neurons, the suggested neuron uses around 59.86% less power when implemented in a pattern recognition neural network.

The work of Kuzum et al. (2012) delves into the creation of phase change materials (PCMs), like Ge2Sb2Te5 (GST), which are extensively implemented in optical data storage and non-volatile memory applications, for the design and development of nanoscale electronic synapses (Kuzum et al., 2012). In order to enable brain-inspired computing systems that go beyond the conventional digital logic paradigm—which has grown inefficient with increasing complexity and input size—researchers are focusing on imitating the behavior of biological synapses. Future neuromorphic systems must include vital properties of biological synapses, including ultrahigh density, energy efficiency, and parallelism. The goal of these systems is to imitate the brain's durability, flexibility, and low energy consumption in information processing—tasks that are considerably above the capabilities of existing digital computers. The study highlights the difficulties encountered by traditional CMOS-based designs, which are constrained in their capacity to scale and use less energy despite their best efforts.

The main contribution of the study is the development of synaptic plasticity-replicating programmable electronic synapses based on PCMs, which includes the crucial learning mechanism of STDP. The gadgets shown by the authors mimic the analog nature of synaptic weight change by exhibiting continuous resistance shifts. The devices can provide a scaling potential for brain-inspired designs by accurately controlling synaptic strength with picojoule-level energy consumption by using the intermediate resistance states of PCMs. The benefits of PCM-based synapses over earlier CMOS techniques are also highlighted in the research, including the possibility of huge parallelism and less area occupation. Because of the PCM synapses' resilience and programmability, different STDP learning rules may be implemented, providing cognitive functions that are similar to those of biological systems.

The work by Zaki et al. (2019) suggests a unique technique for sigmoid function approximation with the goal of enhancing neural network implementations on Field-Programmable Gate Arrays (FPGAs). The study of the literature focuses on how activation functions specifically, the sigmoid contribute to non-linearity in neural networks. The sigmoid function is essential for backpropagation, but hardware implementations face difficulties due to its computational complexity, particularly on resource-constrained FPGAs.

With features like energy economy and parallel processing, FPGAs are perfect for real-time neural network applications like image recognition. However, the direct implementation of complicated functions such as the sigmoid is wasteful due to their restricted resources. Many approximation methods, such as polynomial approximations, lookup tables, and piecewise linear functions, have been investigated to address this. These techniques have disadvantages and sacrifice accuracy for hardware efficiency. While lookup tables demand a large amount of memory, piecewise linear approximations simplify processing but may introduce mistakes. Larger networks cannot use polynomial approximations since they reduce complexity but still require hardware resources like multipliers and adders. The analysis focuses on previous efforts that sought to strike a compromise between hardware efficiency and accuracy, but it also points out that many of these methods either sacrifice too much accuracy or need too many resources for widespread FPGA implementation. Furthermore, the majority of methods are restricted to certain neural network topologies, which limits their wider use. The paper's conclusion highlights the need for a more effective, broadly applicable sigmoid approximation method that strikes a compromise between hardware economy and accuracy. This requirement drives the paper's new methodology, which aims to fill in the gaps in the body of research.

3. RELATED WORK

Many research was done previously on several strategies for optimizing both ANN activation function computation and STDP implementation in hardware. In the field of ANNs, software-based solutions including quantization, trimming, and approximation have been proposed to reduce computational complexity. However, these solutions may impair accuracy or increase network design complexity. Hardware acceleration has emerged as an attractive path due to its capability of large speed increase and energy economy. Different techniques for implementing STDP in hardware, including analog, digital, and hybrid circuitry. While analog circuits offer low power consumption and great parallelism, they may lack precision and scalability. Conversely, digital circuits provide more precision and flexibility but often need large computing resources and energy. Hybrid techniques try to leverage the advantages of both analog and digital circuits while resolving their limitations. Nonetheless, creating efficient and scalable circuits for both ANN activation functions and STDP remains an ongoing research issue.

3.1 Working of the Activation Functions

The hyperbolic tangent, or tanh function, is a prominent activation function used in neural networks. The tanh function gives output values in the range [-1, 1], comparable to the sigmoid function. However, unlike the sigmoid function, the tanh function is zero-centered, that means its output is centered around zero. This characteristic can improve the learning process in neural networks by keeping the activations more balanced. As illustrated in Figure 2. the M1 and M2 are MOSFET transistors form a differential pair, where the difference between their gate voltages controls the output. When the input current (“I_{in}”) fluctuates, it modifies the gate voltages of M1 and M2. The differential pair amplifies the difference between these gate voltages and the output voltage (“V_{out}”) is proportional to the difference between M1 and M2 currents. This behavior mirrors the tanh function’s curve, which transitions smoothly between -1 and 1.

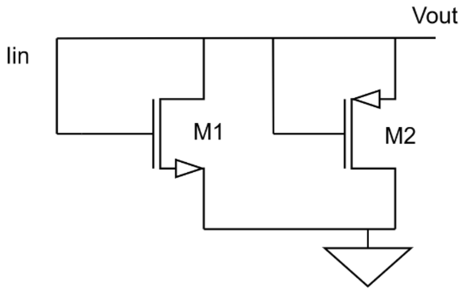

Figure 2. Hyperbolic Tangent Function

The Rectified Linear Unit (ReLU) activation function is sought after in neural networks, especially in deep learning models. It's defined as:

$$f(x)=\max(0, x)$$

In other words, the ReLU function gives 0 for negative inputs and 1 for positive inputs. The key advantage of ReLU is its comprehensibility and effectiveness in training deep neural networks. It helps address the vanishing gradient problem encountered with sigmoid and tanh activation functions, where gradients become very small in deep networks, hindering learning. ReLU avoids this problem by ensuring that gradients are non-zero for positive inputs, facilitating faster and more stable training. ReLU also allows for faster computation compared to sigmoid and tanh, as it involves only a simple thresholding operation.

In Figure 3. when the input voltage (V_{DD}) increases M3 allows more current (I_{in}) to flow from drain to source. This results in an increase in the output voltage (“V_{out}”). The relationship between input and output is linear due to the MOSFET’s behavior. This helps in obtaining a linear activation function curve.

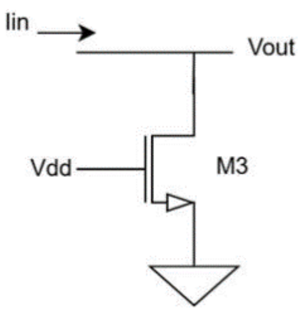

Figure 3. Linear Activation Function

The sigmoid activation function is a commonly used nonlinear function in neural networks. It's characterized by its S-shaped curve, which maps input values to an output range between 0 and 1. The sigmoid function in Figure 4. is particularly useful in binary classification tasks where the output needs to be interpreted as a probability. Values near to 0 indicate one class, whereas values close to 1 indicate the opposite class.

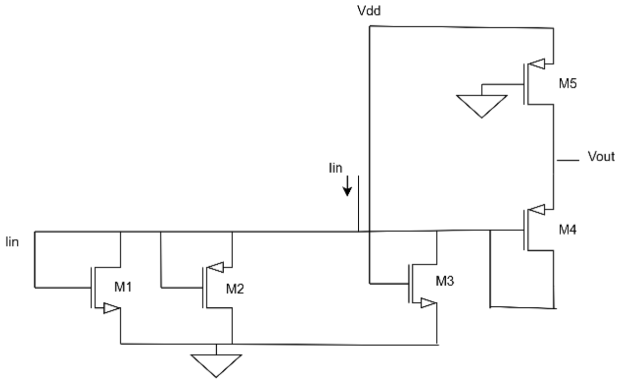

Figure 4. Sigmoid Activation Function

ReLU is the most often used activation function in neural networks, particularly convolutional neural networks (CNNs). The mathematical definition is as follows: y = max (0, x), x ∈ R. We attempted to design a ReLU circuit as shown in the Figure 5. using the input-output characteristic curve of a source follower, as the curve closely resembles the ReLU function. When a suitable DC operating voltage is given to the input, the NMOS transistor operates in saturation mode. The DC operating voltage affects both drain current and the linearity of its input-output characteristics. Proper DC biasing is necessary to approximate the ReLU function.

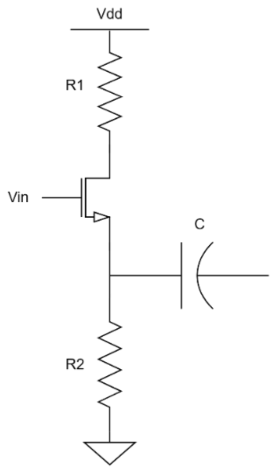

Figure 5. ReLU Activation Function

3.2 Working of the STDP Circuit

The STDP weight update block transforms the variance in time between pre and post spikes ($\Delta t = t_{post} - t_{pre}$) into a change in VG, impacting synaptic weight. Figure 6 depicts the synapse that converts input pre and post pulses into voltage traces ($V_{p,exp}$ and $V_{m,exp}$) via two Exponential Decay Circuit (EDC) blocks. The EDC outputs are transformed to current via the common Gm, which is subsequently integrated on C1. The exponential trace is created with active resistors ($R_{p,m}$) and time constants ($\tau_{p,m} = R_{p,m}C_{p,m}$) that may be altered independently. Transient simulations show how synaptic weight shifts during STDP learning events, including as short-term potentiation and depression.

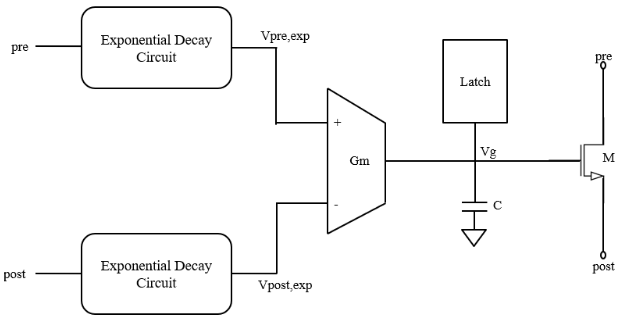

Figure 6. STDP Synapse

The circuit exhibits long-term bistability through a weak latch mechanism, allowing the synapse to retain binary conductance states. This bistability ensures persistent weights after training, enabling efficient read-out and system-level exploration in NeuSoC architectures. The circuit has long-term bistability due to a weak latch mechanism, which allows the synapse to maintain binary conductance states. This bistability guarantees that weights survive after training, allowing for fast readout and system-level exploration in NeuSoC designs.

4. SIMULATION AND RESULTS

In this study, we conducted simulations to analyze the performance of various activation functions using 90nm technology process. We implemented the circuit design of sigmoid, ReLU, tanh and linear activation functions to assess their suitability for neural network applications. A quantitative comparison of power, energy, and voltage levels for normal and high Vt conditions is summarized in Table 1. Each activation function was simulated under different input conditions to evaluate its response characteristics and computational efficiency. The output graphs for the same are shown below in the Figure 7.

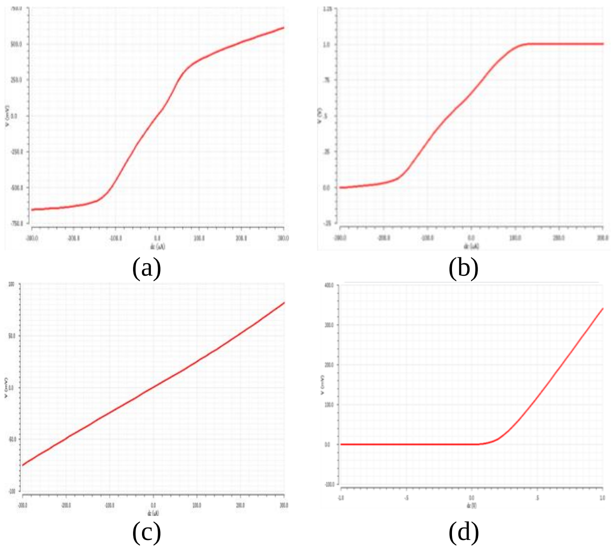

Figure 7. a). Tanh graph b). Sigmoid graph c). Linear graph d). ReLU graph

Table 1. Comparison of power, energy and voltage with respect to Normal and High V_t.

Activation Function	Normal V_t			High V_t		
	Power (in uW)	Energy (in pJ)	Voltage (in mV)	Power (in uW)	Energy (in pJ)	Voltage (in mV)
Linear	2.491	0.1246	-85 to 85	3.787	0.189	-110 to 130
Tanh	38.54	1.927	-650 to 650	56.06	2.901	-650 to 800
Sigmoid	65.33	3.267	0 to 1V	96.34	4.817	100 to 1000
ReLu	56.84	2.842	560	42.07	2.103	410

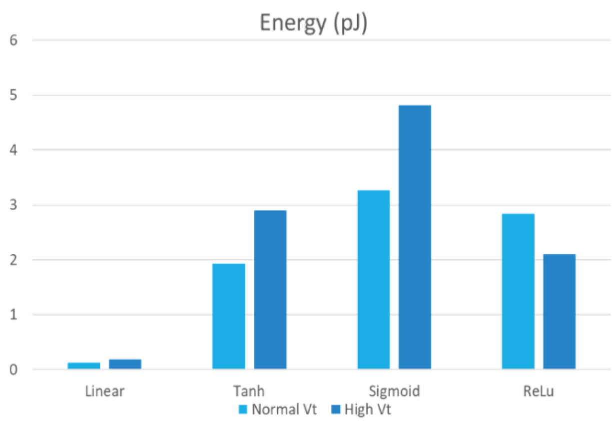

Figure 8. Power Analysis

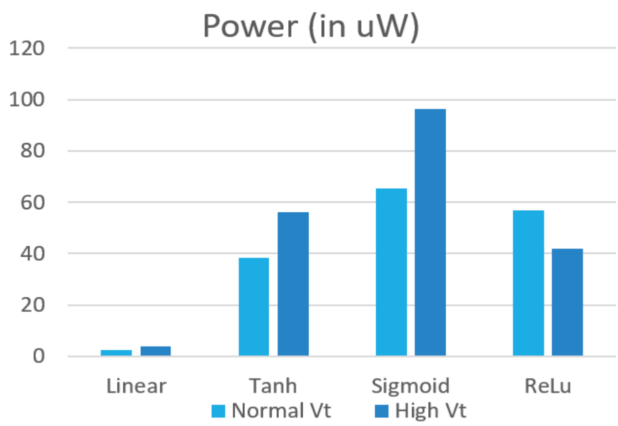

Figure 9. Energy Analysis

The simulation results revealed distinct behaviors for each activation function. As illustrated in Figure 8, the power consumption varies significantly across activation functions, with the sigmoid function exhibiting the highest power consumption and the linear activation function showing the lowest. Similarly, Figure 9 presents the energy consumption analysis, confirming that the sigmoid activation function consumes the highest energy, while the linear function is the most energy-efficient. It can be observed that the power and energy consumption of the sigmoid activation function are the highest, whereas those of the linear activation function are the lowest. Overall, these results highlight the performance trade-offs that must be considered when selecting activation functions for neural network applications.

5. CONCLUSION

Finally, our research shed light on how activation functions and Spike Timing Dependent Plasticity (STDP) were implemented. We conducted comprehensive simulations to assess the efficacy of several activation functions in digital circuit designs, including sigmoid, ReLU, linear and tanh. Furthermore, we looked at the feasibility of implementing STDP in hardware, which is critical for synaptic plasticity in neural networks. The power consumption of the circuit is found to be 194.1 nW. Overall, our study contributes to the understanding of activation functions and STDP implementation in advanced semiconductor processes, laying the groundwork for future developments in neural network hardware design.

References:

Acciarito, S., Cristini, A., Di Nunzio, L., Khanal, G. M., & Susi, G. (2016, June). A VLSI driving circuit for memristor-based STDP. In *2016 12th Conference on Ph. D. Research in Microelectronics and Electronics (PRIME)* (pp. 1-4). IEEE. https://doi.org/10.1109/PRIME.2016.7519503

Chaturvedi, M. S., & Kurshid, A. A. (2015). ASIC implementation for improved character recognition and classification using SNN model. *Procedia Computer Science*, *62*, 151-158. https://doi.org/10.1016/j.procs.2015.08.428

Cruz-Albrecht, J. M., Yung, M. W., & Srinivasa, N. (2012). Energy-efficient neuron, synapse and STDP integrated circuits. *IEEE transactions on biomedical circuits and systems*, *6*(3), 246-256. https://doi.org/10.1109/TBCAS.2011.2174152

Hajduk, Z. (2018). Hardware implementation of hyperbolic tangent and sigmoid activation functions. *Bulletin of the Polish Academy of Sciences. Technical Sciences*, *66*(5), 563-577.

Indiveri, G., Linares-Barranco, B., Legenstein, R., Deligeorgis, G., & Prodromakis, T. (2013). Integration of nanoscale memristor synapses in neuromorphic computing architectures. *Nanotechnology*, *24*(38), 384010. Integration of nanoscale memristor synapses in neuromorphic computing architectures - IOPscience

Kuzum, D., Jeyasingh, R. G., Lee, B., & Wong, H. S. P. (2012). Nanoelectronic programmable synapses based on phase change materials for brain-inspired computing. *Nano letters*, *12*(5), 2179-2186. Nanoelectronic Programmable Synapses Based on Phase Change Materials for Brain-Inspired Computing | Nano Letters (acs.org)

Li, J., Xu, H., Deng, J., & Sun, X. (2016, July). Hyperbolic linear units for deep convolutional neural networks. In 2016 International Joint Conference on Neural Networks (IJCNN) (pp. 353-359). IEEE. https://doi.org/10.1109/IJCNN.2016.7727220

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, *33*(12), 6999-7019. https://doi.org/10.1109/TNNLS.2021.3084827

Lin, C. W., & Wang, J. S. (2008, May). A digital circuit design of hyperbolic tangent sigmoid function for neural networks. In *2008 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 856-859). IEEE. https://doi.org/10.1109/ISCAS.2008.4541553

Liu, F., Zhang, B., Chen, G., Gong, G., Lu, H., & Li, W. (2021, November). A novel configurable high-precision and low-cost circuit design of sigmoid and tanh activation function. In *2021 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)* (pp. 222-223). IEEE. https://doi.org/10.1109/ICTA53157.2021.9661606

Pan, W. Q., Chen, J., Kuang, R., Li, Y., He, Y. H., Feng, G. R., ... & Miao, X. S. (2020). Strategies to improve the accuracy of memristor-based convolutional neural networks. *IEEE Transactions on Electron Devices*, *67*(3), 895-901. https://doi.org/10.1109/TED.2019.2963323

Saxena, V., Wu, X., Srivastava, I., & Zhu, K. (2017, September). Towards spiking neuromorphic system-on-a-chip with bio-plausible synapses using emerging devices. In *Proceedings of the 4th ACM International Conference on Nanoscale Computing and Communication* (pp. 1-6). https://doi.org/10.1145/3109453.3123961

Shakiba, F. M., & Zhou, M. (2020). Novel analog implementation of a hyperbolic tangent neuron in artificial neural networks. *IEEE Transactions on Industrial Electronics*, *68*(11), 10856-10867. https://doi.org/10.1109/TIE.2020.3034856

Shamsi, J., Amirsoleimani, A., Mirzakuchaki, S., Ahmade, A., Alirezaee, S., & Ahmadi, M. (2015, May). Hyperbolic tangent passive resistive-type neuron. In *2015 IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 581-584). IEEE. https://doi.org/10.1109/ISCAS.2015.7168700

Yildiz, H. A., Altun, M., Gungordu, A. D., & Stan, M. R. (2019, November). Analog neural network based on memristor crossbar arrays. In *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)* (pp. 358-361). IEEE. https://doi.org/10.23919/ELECO47770.2019.8990597

Zaki, P. W., Hashem, A. M., Fahim, E. A., Masnour, M. A., ElGenk, S. M., Mashaly, M., & Ismail, S. M. (2019, December). A novel sigmoid function approximation suitable for neural networks on FPGA. In *2019 15th International Computer Engineering Conference (ICENCO)* (pp. 95-99). IEEE. https://doi.org/10.1109/ICENCO48310.2019.9027479

Karpagam T. V.
Vellore Institute of Technology,
Chennai,
India
Karpagam.tv@vitstudent.ac.in
ORCID 0009-0006-2840-8499

Jayanth D.
Vellore Institute of Technology,
Chennai,
India
jayanth.d2023@vitstudent.ac.in
ORCID 0009-0007-1603-8408

Anita Angeline A.
Vellore Institute of Technology,
Chennai,
India
anitaangeline.a@vit.ac.in
ORCID 0000-0002-5603-0976