2021

Long Short-Term Memory with Spin-Based Binary and Non-Binary Neurons

Meghana Reddy Vangala
University of Central Florida

LONG SHORT-TERM MEMORY WITH SPIN-BASED BINARY AND NON-BINARY NEURONS

by

MEGHANA REDDY VANGALA
B.Tech. Jawaharlal Nehru Institute of Technology, 2019

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Electrical & Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2021

Major Professor: Ronald F. DeMara

ABSTRACT

Research in the field of neural networks has shown advancement in the device technology and machine learning application platforms of use. Some of the major applications of neural network prominent in recent scenarios include image recognition, machine translation, text classification and object categorization. With these advancements, there is a need for more energy-efficient and low area overhead circuits in the hardware implementations. Previous works have concentrated primarily on CMOS technology-based implementations which can face challenges of high energy consumption, memory wall, and volatility complications for standby modes. We herein developed a low-power and area-efficient hardware implementation for Long Short-Term Memory (LSTM) networks as a type of Recurrent Neural Network (RNN). To achieve energy efficiency while maintaining comparable accuracy commensurate with the ideal case, the LSTM network herein uses Resistive Random-Access Memory (ReRAM) based synapses along with spin-based non-binary neurons. The proposed neuron has a novel activation mechanism that mimics the ideal hyperbolic tangent (tanh) and sigmoid activation functions with five levels of output accuracy. Using ideal, binary, and the proposed non-binary neurons, we investigated the performance of an LSTM network for name prediction dataset. The comparison of the results shows that our proposed neuron can achieve up to 85% accuracy and perplexity of 1.56, which attains performance similar to algorithmic expectations of near-ideal neurons. The simulations show that our proposed neuron achieves up to 34-fold improvement in energy efficiency and 2-fold area reduction compared to the CMOS-based non-binary designs.

Dedicated to my family and friends who have always supported me in the process of completing my thesis successfully.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Dr. DeMara who provided me an opportunity to join the Computer Architecture Laboratory (CAL) research team and guided me through my research under his supervision. He has kindly supported me during this project, and his insightful comments helped me to proceed my research along the right direction leading to several publications in prestigious conferences. I would also like to thank my committee members, Dr. Kalpathy Sundaram, and Dr. Zixia Song for their support.

Furthermore, this work was supported in part by the Center for Probabilistic Spin Logic for Low Energy Boolean and Non-Boolean Computing (CAPSL), one of the Nanoelectronic Computing Research (nCORE) Centers as task 2759.006, a Semiconductor Research Corporation (SRC) program sponsored by the NSF through CCF 1739635.

TABLE OF CONTENTS

LIST OF FIGURES

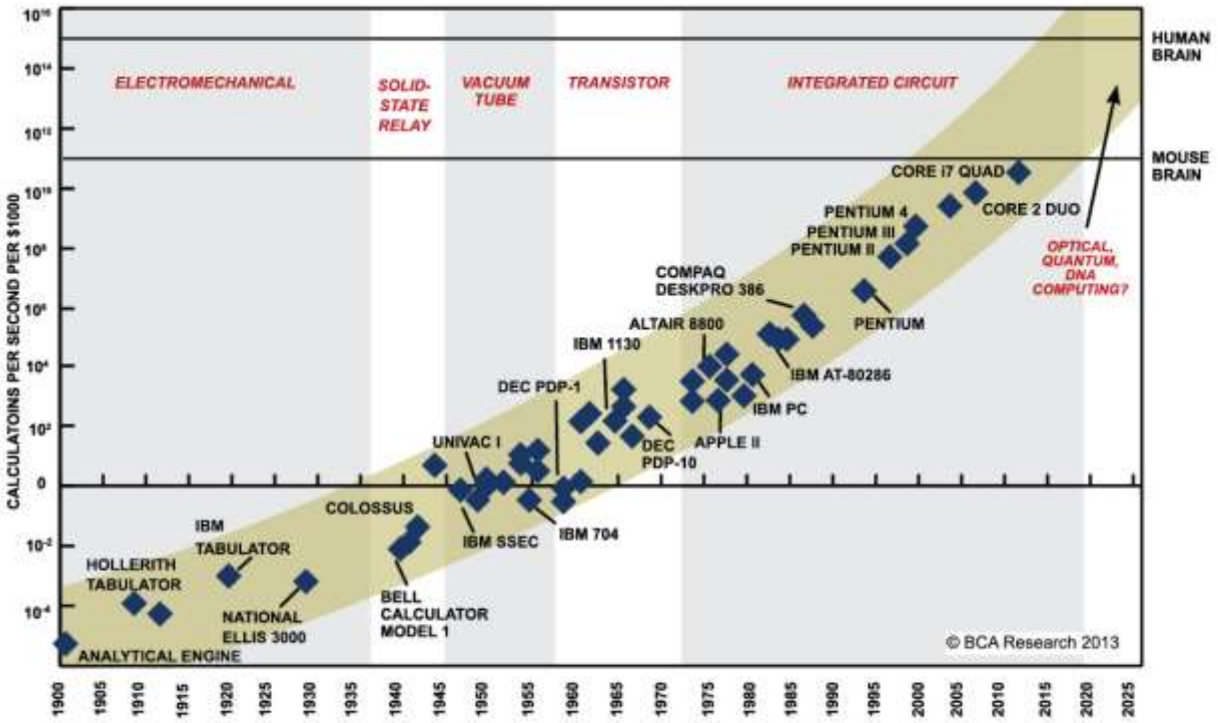
LIST OF TABLES

CHAPTER ONE: INTRODUCTION

Research Motivation

The number of transistors on integrated circuits has doubled every two years as stated by Moore's Law, shown in Figure 1. This increase has led to Complementary Metal-Oxide Semiconductor (CMOS) technologies beyond the 20nm process node. The transistor density of these highly scaled process technologies is enhanced while the nominal supply voltage is reduced. For CMOS devices and designers, power density and area have always been two major difficulties [1]. As the level of integration increases, many benefits and challenges arise [2]. The self-aligned double-gate MOSFET structure (FinFET) is one of the most promising semiconductor architectures for extending Moore's law to 20nm and beyond. FinFET transistors address difficulties including sub-threshold leakage, poor short-channel electrostatic behavior, and high device variability that affects traditional CMOS devices. Furthermore, due that it can run at a lower supply voltage, it saves both static and dynamic power [3]. Although challenges like Process Variation (PV) [4, 5], aging, and bias temperature and threshold voltage instability [6, 7] become more severe at increasing levels of integration, computing systems' capability is considerably boosted while their cost is reduced.

Miniaturization of CMOS technology has improved chip performance at a lower cost by increasing transistor density and switching speed, which has been the fundamental goal of silicon technology for decades [8]. However, as this technology is scaled down to nanoscales, a new challenge of leakage arises. In addition to this, CMOS technology scaling has other challenges such as limited gate control, increased lithography costs, high leakage currents, higher circuit noise sensitivity, and high-power density. Internet-of-Things (IoT) devices, on the other hand, have evolved into the most popular and rising technology in the last decade because of the convergence of several technologies such as embedded systems, machine learning, and cloud computing. IoT circuits offer a low cost, low overhead, and low-power data acquisition and processing capabilities [9]. Furthermore, due to the limited energy budget and the constraints posed by device scalability, achieving energy-efficient and high-performance computing is one of the primary goals of IoT applications. Considering these challenges, a hybrid and energy-efficient circuits can be designed by combining CMOS technology with emerging technologies popularly known as Spintronics [10, 11]. Because of the possibility of 3D integration at the back-end process, which can merge logic and memory and minimize dynamic power, Spintronic technology is specifically compatible with CMOS technology. Non-volatility, low area overhead, and near-zero static power are the major characteristics of Spintronic devices that make them a good choice for next-generation hybrid technologies. Furthermore, the non-volatility feature lowers standby power by preserving data even when the power is turned off. These features of Spintronics can be used to design energy-efficient circuits, arrays of non-volatile memory and novel activation function units for neuromorphic computational architectures, and area-efficient digital circuits [12].



SOURCE: RAY KURZWEIL, "THE SINGULARITY IS NEAR: WHEN HUMANS TRANSCEND BIOLOGY", P.67, *THE VIKING PRESS*, 2006. DATAPOINTS BETWEEN 2000 AND 2012 REPRESENT BCA ESTIMATES.

Figure 1: Moore's Law [13].

Need for Energy Efficient architectures in Neural Networks

With technological advancements and increasing production rates of electronic companies, the number of edge devices has increased significantly in recent years, and billions of connected devices are expected to generate vast amounts of raw data in the near future [14]. One of the key goals in creating next-generation IoT devices is to achieve battery-free computing by utilizing solely ambient sources of light, kinetic, thermal, and electromagnetic energy instead of batteries [15]. Machine learning approaches, on the other hand, have gotten a lot of attention and have made significant progress in areas like image recognition, machine translation, text-

Table 1:Energy consumption of various operations in 45nm CMOS processor [16]

Operations	**Energy (pJ)**	**Relative Energy Cost**
32-bit integer addition	0.1	1
32-bit floating-point addition	0.9	9
32-bit integer multiplication	3.1	31
32-bit floating-point multiplication	3.7	37
32-bit SRAM Access	5	50
32-bit DRAM Access	640	6400

classification, object categorization, and so on [17]. Larger model sizes and higher computing workload are required to achieve higher accuracy levels in various Neural Network (NN) applications. The inference process is typically performed on the cloud when running a NN on an IoT device. However, as it minimizes latency, improves privacy, and moderates execution time, conducting the inference on the edge device itself is gaining popularity [18, 19]. Edge devices, on the other hand, have limited on-chip cache memory capacity (usually <10 Mb), necessitating the use of off-chip main memory for high-performance variants [20]. For pre-processing huge data, there are numerous algorithms that run on general-purpose traditional CPUs. The enormous demand for data transportation between distinct processor and memory units, referred to as the memory bottleneck, prevents von Neumann computing architectures from processing huge data efficiently. According to studies, a 32-bit Dynamic Random-Access Memory (DRAM) read operation consumes higher energy than a 32-bit floating point multiplication when compared to on-chip processes, as shown in Table 1:Energy consumption of various operations in 45nm CMOS processor [16].

Contributions, Summary, and Organization of the Thesis

As a consequence of the motivations, this thesis mainly focuses on designing an energy-efficient and low area-overhead Long Short-Term Memory architecture using ReRAM-based synaptic crossbar arrays and spin-based non-binary neurons. In summary, the major contributions in this thesis are as follows:

1. First, a ReRAM based crossbar is designed which are inspired from the work proposed in [21] where they proposed a ReRAM-based crossbar RNN hardware implementation with feedback using spin-based Adjustable Probabilistic Activation Function (APAF) to achieve high energy- and area-efficiency, while keeping the accuracy loss and processing speed comparable with the baseline designs [21].
2. The proposed spin-based non-binary neuron activation function is designed using a spintronic device namely, probabilistic-bit (p-bit). Leveraging the stochastic property of a p-bit device, a spin-based activation function is designed that mimics the ideal sigmoidal and tanh activation functions thereby providing an energy efficient and low-area overhead design while maintaining comparable accuracy commensurate with the ideal case. Circuit-Level and experimental analysis is done for the design proposed and the results are further explained in future chapters.

This thesis is organized into five chapters. Figure 2: Organization of Thesis outlines the materials that each chapter covers. A quick overview of background of topics required for this thesis are provided in Chapter 2. In Chapter 3, all the previous works which include works on activation function unit, Recurrent Neural Network, and Long Short-Term Memory are elaborated. Chapter 4 includes our proposed spin-based LSTM network with non-binary neuron and its corresponding simulation and experimental results. This thesis then concludes in Chapter 5.

The material herein includes and/or extends the contents of research works and/or publications including [22], [21].

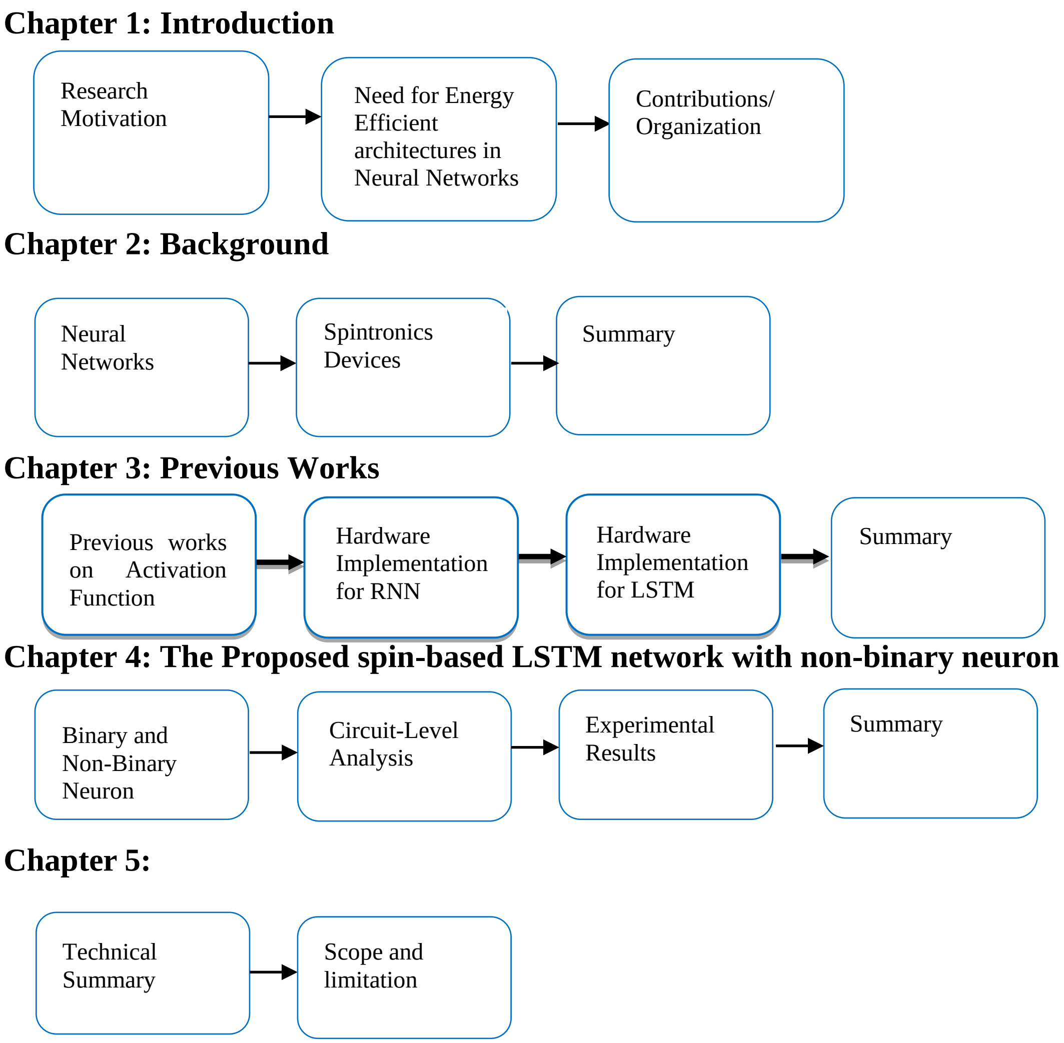

Chapter 1: Introduction
Research Motivation
Need for Energy Efficient architectures in Neural Networks
Contributions/ Organization
Chapter 2: Background
Neural Networks
Spintronics Devices
Summary
Chapter 3: Previous Works
Previous works on Activation Function
Hardware Implementation for RNN
Hardware Implementation for LSTM
Summary
Chapter 4: The Proposed spin-based LSTM network with non-binary neuron
Binary and Non-Binary Neuron
Circuit-Level Analysis
Experimental Results
Summary
Chapter 5:
Technical Summary
Scope and limitation

Figure 2: Organization of Thesis

CHAPTER TWO: BACKGROUND

Neural Networks

Reccurent Neural Network

Recurrent Neural Networks (RNNs) have illustrated eminent accomplishments in machine learning applications including classification, speech acknowledgment, machine translation, and static image processing due to their ability to construct the input data over time [23]. As a group of Artificial Neural Networks (ANNs) centering on successive information, RNNs are based on a repetitive way of the data stream as appeared in Figure 3. Unlike feedforward ANNs, the yield of RNNs depends both on the current input and the past computation outcomes. In this way, the feedback, as a vital and interesting component, gives the memory to capture the computed data in RNNs [23].

In RNNs, as shown in Figure 4, a directed chart is shaped along with transient groupings with an association between its nodes. The input vectors (x(t)) are fed into the network one at a time during forward propagation, controlled towards the neurons within the hidden layer. The states of the hidden neurons are updated upon entry of the input vectors and comparing neural connection weights. The updated neuron state is held for use upon entry of consequent input designs. With the entry of a new input vector at the proceeding time step, the neurons within the hidden layer compute a modern state vector based on the new input vector and the held state vector [24]. Anticipating that the W matrix in Figure 4. represents the repeating input neural connections framework within the hidden layer, Equation (1) and Equation (2) can allow a numerical representation of RNN updating the neuron state over time:

$$h(t) = f(U.x(t) + W.h(t - 1) + b_h) \tag{1}$$

$$y(t) = V.h(t) \tag{2}$$

where $h(t)$ represents the hidden neuron state and $y(t)$ denotes the output neurons state at time step t. f is the activation function in the hidden layer. U and V both signify the feedforward synapse networks matrices, where U holds the synapses from the input layer to the hidden layer and V represents the neural connections from the hidden layer to the output layer. Finally, b_h signifies the predisposition within the neurons of the hidden layer. The synaptic weights and the predisposition vectors are initialized sometime recently preparing based on the network execution [24].

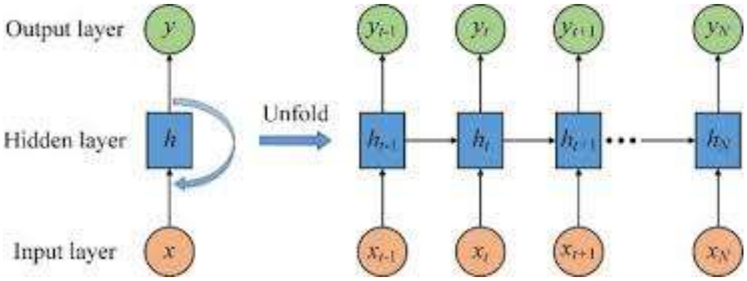


Figure 3: Folded and Unfolded RNN structures [25]

Long Short-Term Memory

Long Short-Term Memory (LSTM) is a specific type of RNN that is designed to overcome some of the drawbacks of RNN. Figure 4 (a) shows the basic RNN structure. RNN output depends on both the current sample (i_t) and the previously calculated network state (w_t)

as the network input. Unlike ANN, RNN has a feedback loop which gives RNN the capability to store the previous states and make future decision based on the previous values. The computational equations of a basic RNN cell are given below:

$$w_t = tanh\ (i_tU + w_{t-1}W + bias) \qquad (3)$$

$$y_t \ = \ softmax(w_tV) \qquad (4)$$

where i_t, w_t, and y_t are the current input, hidden state, and output for the current input, respectively; V, W, and U contain trainable parameter matrices. With the feedback loop, RNN is expected to handle long-term dependencies but this is not true for practical application. RNN can't handle long-term dependencies in practice due to the vanishing gradient problem [26]. LSTM is a special kind of RNN which tries to solve the problem of vanishing gradients that is encountered during the backpropagation technique of neural networks [27]. Figure 4.(b) shows an LSTM cell which contains three gates: input gate x_t, forget gate f_t, and output gate o_t. The forget gate decides which information from the previous cell state to be preserved and which must be forgotten. This decision is taken using a sigmoid layer which gives output between 0 and 1 [28]. The input gate decides which of the new cell contents should be written to the cell state. It has two parts: the sigmoid layer decides which values of input (concatenation of new input values and output values from previous states) to update, and the tanh layer generates a vector of new candidate values. The output gate decides which content of the cell to output based on given inputs and previous state values. The output vector is obtained by multiplying a new cell state which is normalized to values between -1 to 1 using tanh activation function and output of sigmoid layer that decides which part of cell state are given to output. The dimensions of all the

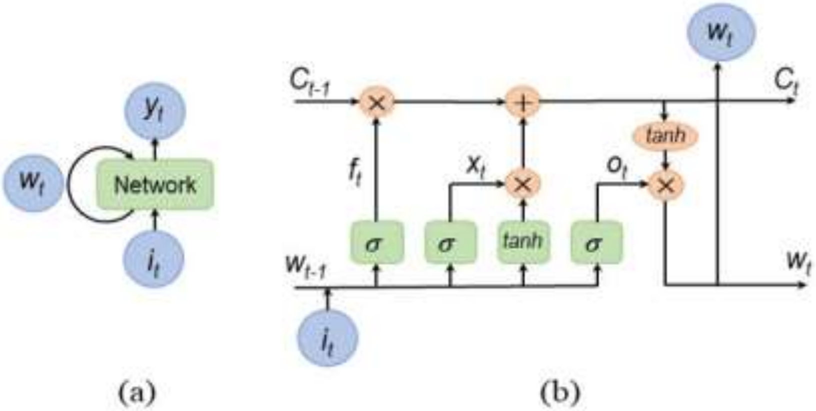


Figure 4: (a) Basic RNN structure, (b) LSTM

gates are same as the dimensions of hidden state. The computational equations of LSTM are given below:

$$x = \boldsymbol{\sigma}(i_t U^x + w_{t-1} W^x + b_x) \quad (5)$$

$$f = \boldsymbol{\sigma}(i_t U^f + w_{t-1} W^f + b_f) \quad (6)$$

$$o = \boldsymbol{\sigma}(i_t U^o + w_{t-1} W^o + b_o) \quad (7)$$

$$g = tanh\,(i_t U^g + w_{t-1} W^g + b_g) \quad (8)$$

$$c_t = c_{t-1} \odot f + g \odot x \quad (9)$$

$$w_t = tanh(c_t) \odot o \quad (10)$$

Three main operation types can be observed from the above equations: nonlinear functions (sigmoid $\boldsymbol{\sigma}$ and hyperbolic tangent $tanh$), matrix-vector multiplication (e.g., $w_{t-1}W^x$ and i_tU^x), and element-wise multiplication (e.g., $g \odot x$) [29].

LSTM Activation Functions: The conventional activation functions used in an LSTM are sigmoid or logistic-sigmoid and hyperbolic tangent in short $tanh$ activation functions. A sigmoid activation function alters any input value to value between 0 and 1. Similarly, a $tanh$ activation function alters any input value to a value between -1 and 1 [30]. This will help to allow or not allow the flow of information through the LSTM gates. The equations of these functions are given below:

$$\sigma(x) = 1/(1 + exp(-x)) \qquad (11)$$

$$\tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x}) \qquad (12)$$

where x is the input, $\sigma(x)$ is the sigmoid function, and $\tanh(x)$ is the hyperbolic tangent function.

Spintronic Devices

Spin Hall Effect- Magnetic Tunnel Junction (SHE-MTJ)

In comparison to Spin Transfer Torque – Magnetic Tunnel Junctions (STT-MTJ), a Spin Hall Effect-Magnetic Tunnel Junction (SHE-MTJ) is a 3-terminal device having segregated paths for write and read operations and lower switching energy. As identified in [14], it comprises of a Heavy Metal (HM) nanowire beneath an MTJ with two ferromagnetic layers separated by a thin oxide barrier, referred as the pinned and free layers [31]. The parallel (P) and antiparallel (AP) magnetization orientations of the MTJ free layer give two different levels of resistance for this device. The HM can have different electrical properties such as β -tungsten (β -W) or β-tantalum (β -Ta) [32]. The device can be designed with tungsten because of the larger positive Spin Hall angle achieved with this material [32]. The free-layer magnetization must be changed in order to store data in the SHE-MTJ. As shown in Figure 5 (a), this is performed by injecting a charge current (I_c) into HM in the +x (/-x) direction. I_c causes an accumulation of oppositely directed spin vectors on both surfaces of

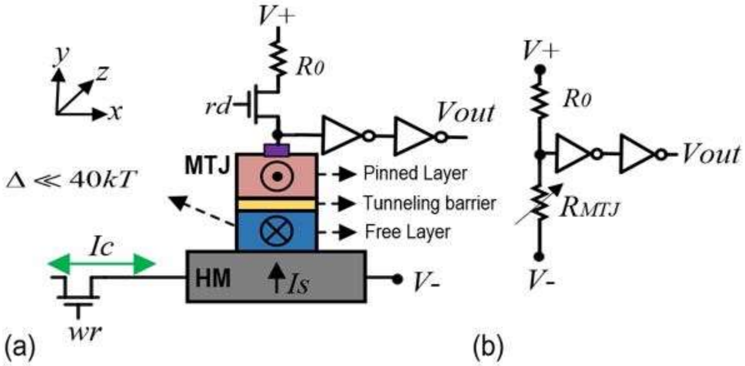


Figure 5: (a) Structure of SHE-MTJ (b) Resistive equivalent read circuit of SHE-MTJ [33].

the HM due to the spin Hall effect, which generates a spin current (Is) and a Spin-Orbit Torque (SOT) in the +y (/-y) direction. According to the direction of the charge current, the spin current will modify the magnetization configuration of the free layer in the ±z direction [34]. The spin Hall injection efficiency (PSHE) is calculated as:

$$P_{SHE} = \frac{I_S}{I_C} = \theta_{SH}\,\frac{A_{FM}}{A_{HM}}\left(1 - \operatorname{sech}\left(\frac{t_{HM}}{\lambda_{sf}}\right)\right) \qquad (13)$$

where A_{FM} and A_{HM} respectively denote the adjacent free layer area and the cross-sectional area of HM. The spin Hall angle is defined as the ratio of generated spin current density to charge current density in equation $P_{SHE} = \frac{I_S}{I_C} = \theta_{SH}\,\frac{A_{FM}}{A_{HM}}\left(1 - \operatorname{sech}\left(\frac{t_{HM}}{\lambda_{sf}}\right)\right)$ (13). t_{HM} and λsf are the HM substrate thickness and spin flip length, respectively [35].

Probabilistic Spintronics Device (p-bit)

The Probabilistic Spintronics Device (p-bit) device has the same structure as Figure 5(a), which consists of a Spin Hall Effect Magnetic Tunnel Junction (SHE-MTJ) with a circular unstable (low energy barrier) nanomagnet ($\Delta \ll 40kT$) [33, 36], with two CMOS inverters amplifying the output. In comparison to Spin Transfer Torque-MTJs, the SHE-MTJ-based p-bit is a 3-terminal device with segregated read (rd) and write (wr) paths and lower switching energy [35, 37]. It comprises of an unstable MTJ with two ferromagnetic layers (pinned layer and free layer) which are divided by a thin oxide barrier on top of a Heavy Metal (HM) nanowire [31] composed of β-tantalum (β-Ta) or β-tungsten (β-W) [32]. The pinned layer has a fixed orientation making it stable nanomagnet, while the MTJ's free layer can be oriented in two ways: parallel (P) or antiparallel (AP), offering two degrees of resistance. The resistance level can be adjusted by introducing a charge current (I_c) in the +x (/-x) direction into the HM, as shown in Figure 5 (a). This charge current causes oppositely directed spin vectors to accumulate on each surface of the HM, resulting in a spin current (I_s) and a Spin-Orbit Torque (SOT) in the +y (/-y) direction. The magnetization configuration of the free layer is changed by spin current in the ±z direction according to the direction of the charge current [34]. Because of its inherent physics, the spin-current driven low energy barrier nanomagnet delivers the sigmoidal function by taking a long-time average of magnetization fluctuations. A SHE-MTJ based p-bit's read circuit is shown in Figure 5 (b). A minimal read voltage is provided to the MTJ (V+ and V- terminals) to evaluate its resistance (R_{MTJ}) in order to read the data. The R_{MTJ} and the reference resistor R0 are then used to form a resistive voltage divider. This reference resistor is adjusted to the MTJ average conductance (R_0 -1 = GP + GAP/2), where GP and GAP stand for parallel (P) and anti-parallel (AP) state conductance, respectively. The corresponding voltage is sent into the CMOS inverters' inputs, which are set to their DC operation's middle point. As a result, the output voltage (V_{out}) will stochastically oscillate between “0” and “1,” with the input charge current regulating the probability of either value [38]. The p-bit device produces a stochastic output that behaves similarly to a sigmoid activation function, with an input current modulating the steady-state probability. For example, if the input current is a huge positive amount, the device's stochastic output will almost certainly be "0" If no input current is present, the output will swing randomly between “0” and “1” with a probability of 0.5.

The circuit simulations were performed using the SPICE platform, and the device features were extracted from the experimentally benchmarked models in [39]. We want to define the output's time-averaged behavior as an analytical approximation. To begin, we connect the flowing charge current in the spin Hall layer to the magnet's absorbed spin current. For the sake of simplicity, we will assume short-circuit conditions, i.e., the FM absorbs 100 percent of the spin:

$$P_{SHE} = \frac{I_S}{I_C} = \theta_{SH} \, \frac{A_{FM}}{A_{HM}} (1 \, - \, \text{sech} \, (\frac{t_{HM}}{\lambda_{sf}})) \tag{14}$$

The free layer area and the cross-sectional area of HM, respectively, are represented by A_{FM} and A_{HM}. θ_{SH} denotes spin hall angle, t_{HM} represents the thickness of the HM substrate, and λ_{sf} represents the spin flip length. The magnitude of a spin-current can be dominant over the "gain" generated by the charge current by choosing an acceptable quantity for the A_{FM} and A_{HM}. As a result, the magnetization behavior is calculated using a function of input spin-current oriented in the ±z direction. A distribution function for a magnet in steady state with Perpendicular Magnetic Anisotropy (PMA) and spin current in the ±z direction can be stated analytically as follows:

$$\rho(m_z) = \frac{1}{z}\exp(\,\Delta m_z^2 + 2i_s m_z\,) \qquad (15)$$

where mz is the +z direction magnetization, z is a normalization constant, Δ is a thermal barrier for nanomagnet, and i_s is the spin-current normalization quantity, which can be defined as $i_s = I_S/(\frac{4\mathrm{q}}{\hbar\alpha\mathrm{kT}})$, where $\hbar$ the reduced Planck constant, α is the magnets' damping coefficient, and q is the electron charge. Equation (15) can be used to obtain an average magnetization as follows: $< m_z > = \int_{-1}^{+1} dm_z\, m_z \rho(m_z) / \int_{-1}^{+1} dm_z\, \rho(m_z)$. As $\Delta \ll kT$, the Langevin function $< m_z > = L(i_s)$ is given by $< mz >$, where $L(x) = \frac{1}{x} - coth\frac{1}{x}$. This illustrates an accurate average magnetization description around a z-directed spin-current for a low energy barrier PMA magnet [38]. However, because the p-bit device nanomagnet has a substantial in-plane anisotropy as well as a circular shape, we are unable to develop a straightforward analytical formula in this work. As a result, we use a fitting parameter in the Langevin function to change the normalization current by a factor η so that the altered normalization constant is translated to $(4q/\hbar\alpha kT)$ (η). This factor grows when the shape anisotropy is increased $(Hd \sim 4\pi Ms)$ and equals "1" when there is no shape anisotropy. When the magnetization and charge currents are coupled, a phenomenological equation can be used to approximate the CMOS inverter output probability in addition to the fitting parameter χ, as follows: $\mathrm{p} = \frac{V_{OUT}}{V_{DD}} \approx \frac{1}{2}\,[Sa1 - \tanh\,(\chi\, <\, \mathrm{mz}\, >)]$. By using physical parameters, this equation can be utilized to link the output probability to the input charge current. Figure 6 depicts a comparison of the Spice model with the aforementioned analytical equivalences. This supports the agreement between both η with the magnetization and χ with CMOS components [38].

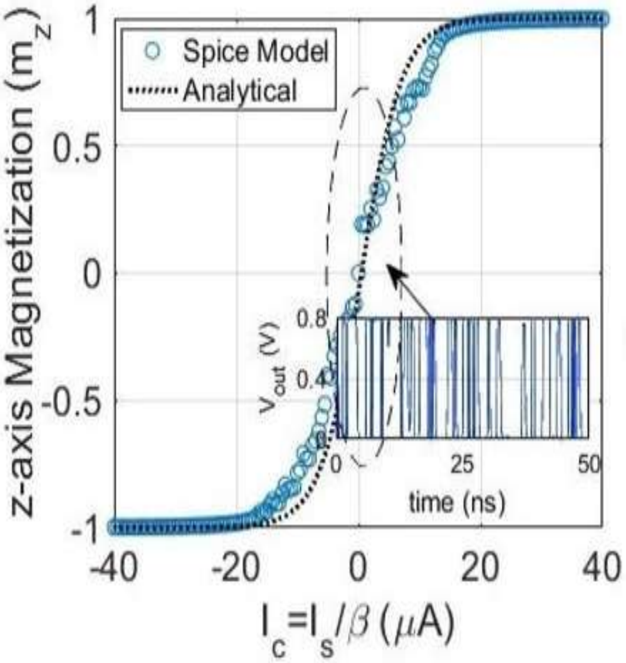


Figure 6: Time-averaged behavior of the SHE-MTJ based p-bit device showing the magnetization fluctuations.

Summary

Machine learning technology is one of the quickly growing technology in the recent scenarios. This technology focuses on applications such as computer vision, natural language processing, semantic analysis, and prediction. Neural Networks is an advancing subset of machine learning that has captured the interest of many researchers from past few years. Some of these works are presented in CHAPTER THREE: PREVIOUS WORKS. Neural networks are a part of Deep Learning which has got algorithmic advances from Artificial Neural Networks (ANN) feedforward propagation to present Recurrent Neural Networks (RNN) feedback propagation. With the advancement in the applications used, these RNN are further developed to Long Short-Term Memory and Gated Recurrent Units (GRU). RNNs are based on a repetitive way of the data stream as shown in Figure 3. Unlike feedforward ANNs, the yield of RNNs depends both on the current input and the past computation comes about. In this way, the feedback, as a vital and interesting component, gives the memory to capture the computed data in RNNs [23]. LSTM is a special kind of RNN which tries to solve the problem of vanishing gradients that is encountered during the backpropagation technique of neural networks [27]. The GRU is similar to a LSTM with a forget gate, but it lacks an output gate, hence it has fewer parameters [40].

Spintronic devices are emerging technology that has emerged from spintronics which is the study of the electron's intrinsic spin and associated magnetic moment in solid-state devices [41]. Spintronic devices are an advanced in non-volatility, low area-overhead, less power consumption. The Spintronic devices mentioned in this chapter are Spin Hall Effect-Magnetic Tunnel Junction (SHE-MTJ) and Probabilistic Spintronics Device (p-bit). In comparison to Spin Transfer Torque – Magnetic Tunnel Junctions (STT-MTJ), a Spin Hall Effect-Magnetic Tunnel Junction (SHE-MTJ) is a 3-terminal device having segregated paths for write and read operations and lower switching energy. A p-bit device works stochastically and is not certain to give same output each time however, it gives an output of either “0” or “1” with some probability. In the proposed design a spin hall effect switching based probabilistic bit is used to mimic sigmoid or tanh activation function.

CHAPTER THREE: PREVIOUS WORKS

Even though Long Short-Term Memory (LSTM) is an algorithm-level advancement for Recurrent Neural Networks (RNN), there is still need for energy efficient hardware implementations of LSTM. The LSTM hardware implementations on FPGA [42], memristor-based [43], and SHE-MTJ [14] have been investigated in prior works. Neural Networks based on von-Neumann architecture, despite significant improvements in performance, nevertheless face the well-known memory-wall challenge, which is caused by limited memory bandwidth, high energy consumption for data transportation between memory and processing units, and long memory access latency [44]. Computing-in-Memory (CiM) architectures provide a realistic non-von-Neumann infrastructure to boost parallelism and reduce the data movement issue, avoiding the memory-wall challenge, allowing for an efficient neural network hardware implementation. Various CiM platforms have proven data-level parallelism as well as increased processing speed and energy efficiency [45]. However, implementing these designs above volatile memories such as SRAM/DRAM necessitates large complex circuitry that consume significant switching energy in order to perform Multiplication and Accumulation (MAC) and activation functions, which are the fundamental operations of neural network [14, 45].

Alternatively, emerging Non-Volatile Memory (NVM) devices such as Phase Change Memory (PCM) [46], Resistive Random-Access Memory (ReRAM) [47], and Spin-Transfer Torque Magnetic Random-Access Memories (STT-MRAMs) [48], have been investigated to implement MAC operation using CiM cross-bar architecture's intrinsic weighted summation property. ReRAM crossbar accelerators have received a lot of attention recently because of their high ultra-low power consumption, Ron/Roff ratio (~106), and switching speed and high scalability [47, 49] for realizing feedforward neural networks like Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs) [50], Multi-layer Perceptron (MLP), and so on. However, due to two limitations, there are just a few studies on ReRAM-based LSTMs. First, implementing a feedback component as a necessary component of LSTMs demands few unavoidable write-back operations which are considered as an inefficient, energy-intensive and high-latency (>20ns [49]), operations of NVMs. Second, the structure of the used activation function has a significant impact on the accuracy of RNNs. The non-linear sigmoid and tanh activation functions, which are the principal thresholding functions employed in LSTMs, necessitate a large area and power budget in a CMOS architecture. In the neuromorphic computing paradigm, minimizing these aspects is an important yet understudied subject. In this chapter, we will go over various previous works explored on Activation Function Unit, hardware implementation of RNN and LSTM.

Previous works on Activation Function Unit

One of the challenging research goals in neural networks is to implement an optimum low-power activation function with a low area overhead. Various activation function designs have been presented so far, leveraging both CMOS-based and emerging device-based technologies. However, due to the enormous number of activation functions used in each layer of neural networks, these designs still have high energy consumption or take up a lot of space, making them unsuitable for growing complex multi-layer networks. Here, we look at a few of these activation functions in more detail. In [51], a tanh activation function is designed using CMOS-based stochastic design with Finite State Machines as its building block, with the goal of lowering power consumption and area overhead by using simpler stochastic arithmetic. However, to implement the probabilistic behavior, this architecture necessitates long bit-stream lengths generated by Linear Feedback Shift Registers (LFSRs) and CMOS pseudo-random number generators, resulting in longer latencies and higher energy consumption.

According to the authors in [52], implementing a sigmoid function results in unnecessary energy and area overheads, therefore they have designed a hardware implementation based on subsampling and approximation which can achieve higher energy efficiency while causing only minimal accuracy loss. Although this solution is quite practical, the implemented activation function still has large energy and area overheads because it uses logic gates for its approximation unit and a 64x16 lookup table on top of a pseudo random number generator. A CMOS-based activation function with a large circuit footprint and significant energy consumption is shown in [24]. It consists of four independent parts: a current generator, a function generator, a pulse generator, and a digital controller. To build a tanh activation function, the special function unit in [29] uses the Chebyshev approximation [53] technique, which has relatively high power and area compared to other similar approaches. In this technique, the CPU calculates the coefficients first and stores them into the local register. The unit will then read the register and calculate the nonlinear function in RNN computing mode. Other efforts, on the other hand, are based on hybrid spin-CMOS p-bit devices, which use the intrinsic physical characteristics of micro magnets to conduct computation [38]. Although this stochastic activation function has a very small footprint and uses very little power, the circuit's output is probabilistic binary (either "0" or "1"), which makes it impossible to employ in LSTMs with deterministic sigmoid and tanh functions. As a result, we were inspired to present a novel activation function based on a p-bit device with software assistance that can perform non-linear functions in a semi-probabilistic manner while maintaining high accuracy.

Hardware Implementation for RNN

Most previous approaches for neural network hardware implementation use CMOS-based activation functions with a built-in truth table [29], however such design leads to a large area overhead and increased clock cycles to evaluate the desired function. ReRAM crossbar arrays are used as synapses in the RNN implementation in [24], coupled with CMOS-based activation functions. Even though this work presents a detailed hardware implementation for RNNs and provides an effective synaptic connection, the CMOS-based neuron is a large complex circuit consisting of four independent parts. All these components eventually cause the overall design to consume a lot of energy. In [29], the author provides a detailed design of RNN using ReRAM-based CiM architecture with a unique processing engine that uses three distinct subarrays for data processing including specialized functional units, a multiplier, and the use of a ReRAM-based crossbar. However, the neuron design, on the other hand, takes up a lot of silicon space and consumes a lot of energy. Previous research have looked into RNN hardware implementations on FPGA [54], ASIC [55], and GPU [56]. Previous research have also focused on speeding up the training or inference phases of typical RNNs. [57] [58] [59] are instances of representative work. [57] focuses on implementing an RNN-based multiuser detection

(MUD) for CDMA, whereas [58] focuses on speeding up a novel RNN training strategy on FPGAs. Li et al. present a framework for training an RNN-based language model in [59]. All these designs use fixed-point data, and they perform fairly well. However, the RNN models that are implemented do not perform well enough in terms of accuracy prediction, preventing them from being used in real life applications.

The authors in [55] show an 8.1TOPS/W reconfigurable CNN-RNN processor with three major characteristics in [55]:

1. To support general-purpose RNNs, a reconfigurable heterogeneous architecture with a CL processor (CP) and an FC-RL processor (FRP) is used [55].
2. To get the most out of kernel reuse in the CP, a LUT-based reconfigurable multiplier is optimized for the dynamic fixed point with on-chip adaptation via overflow monitoring [55].
3. Matrix multiplication based on quantization table (Q-table) to decrease off-chip memory access and eliminate redundant multiplications in the FRP [55].

For Convolutional Neural Networks (CNNs), a variety of specialized accelerators have been proposed. To map massive Deep Neural Network (DNN) onto its core architecture, DianNao [60] uses an array of multiply-add units. Because of the restricted SRAM resources, off-chip DRAM traffic consumes majority of the energy. By having all weights on-chip, DaDianNao [61] and ShiDianNao [62] eliminate DRAM access (eDRAM or SRAM). These

DianNao-series designs, on the other hand, were proposed to speed up CNN, and the weights are uncompressed and stored in a dense manner.

Compared to all the other works previously proposed, the work in [21] suggested an energy-efficient RNN platform with ReRAM crossbar to fully implement the network's low-latency feedback component and low-area-overhead activation function. This work is the motivation for the design proposed in this thesis. To learn in detailed about the work in [21], Figure 7 depicts a comprehensive illustration of the concept presented in [21] for CiM architecture. By partitioning each memory chip into numerous memory banks, this architecture is effectively built on top of the 1T1R-resistive main memory architecture [47, 63]. As shown in Figure 7, each memory bank is subsequently partitioned into several computing sub-arrays utilizing Resistive Crossbars. They organized the resistive crossbar units at the bank level into sets of three interconnected subunits indicated by U-Array, W-Array, and V-Array in order to make the ReRAM-based accelerator appropriate for RNN computation. The circuit and interconnection strategy for the sub-units is shown in Figure 8. The Source-Line (SL) is shared among the resistive synapses connected to neurons in the same column, and the Bit-line (BL) and Word-line (WL) are shared among the synapses in the same row in each crossbar. As a result, three signals govern each synapse in the resistive crossbar. To save energy and reduce area overhead, the W and V resistive crossbar arrays are coupled to a shared activation function unit through the SL peripheral. However, the U-array is only connected to the internal memory bus. All sub-arrays additionally include a buffer component (Buf in Figure 7) connected to the SL peripherals to hold the output value before passing it to the activation functions.

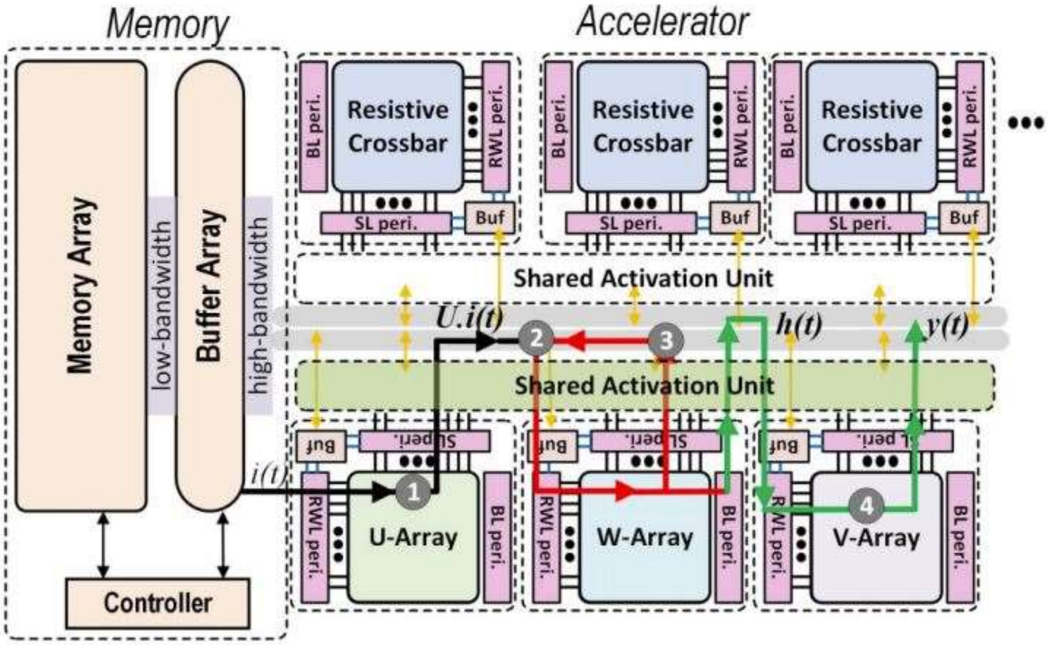


Figure 7: RNN CiM accelerator architecture [21]

The activation function designed in [47] is a spin-based activation function using stochastic p-bit device. The design is an Adjustable Probabilistic Activation Function (APAF) that extracts the stochastic behavior of a p-bit device and stores the output of each execution of p-bit device which is run with the same input for multiple intervals of time. These symmetric range of output voltages are mapped to a low-overhead Look Up Table (LUT). When compared to its CMOS equivalent, this approach permits the p-bit to work in an enhanced non-binary state while keeping its low-power and low-area qualities. The p-bit stochastic activation function was improved by adding three components for hardware implementation. To latch the output voltage of the p-bit circuit (out array), a 2^n-bit buffer is added first. Second, to sum up and compress the saved binary data in out_array a compressor unit (cmp) made up of CMOS full adders is used. Finally, the activation function output is generated using a LUT. This design is further developed and is utilized in our proposed design.

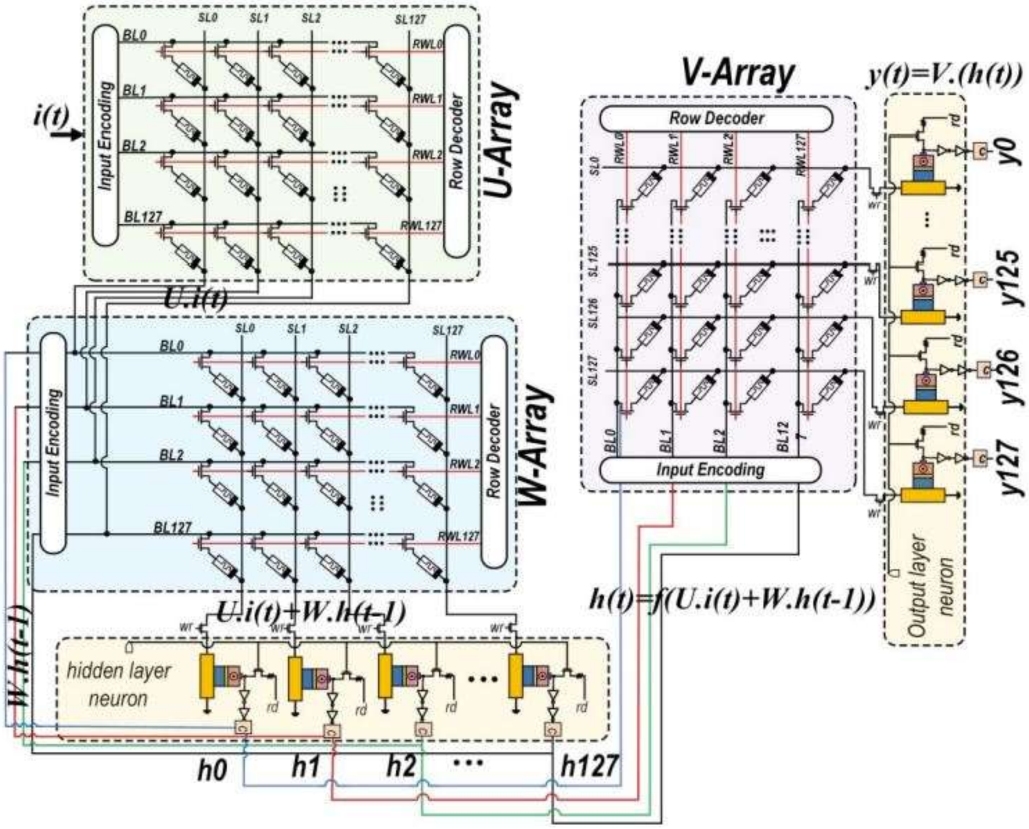


Figure 8: ReRAM-based RNN architecture with stochastic activation functions as neurons [21]

Hardware Implementation for LSTM

A Memristor is the fourth fundamental two-terminal circuit element that can be used as a non-volatile memory electronic memory device [64]. For the implementation of LSTM, a memristor crossbar array is used to do matrix-vector multiplication.

Analog LSTM circuit: The whole LSTM architecture is designed based on different CMOS circuits such as CMOS-memristor crossbar array is used for matrix-vector multiplication and

CMOS-based activation function sigmoid and tanh functions. Earlier work [65] offered CMOS-memristor analog circuit design of current-based LSTM cell architecture which used current mirrors and current-based activation function circuits. To improve the accuracy voltage-based circuits are designed [43]. Comparison between these two designs is given in Table 2 [66].

Table 2: Comparison between voltage-based and current -based analog LSTM circuit [66]:

LSTM architecture	**Power consumption**	**Area**	**RMSE (software)**	**RMSE (circuit)**
Current-based LSTM	105.9 mW	77.33 µm2	55.26%	47.33%
Voltage-based LSTM	225.67 mW	108.60 µm2	10.05%	10.99%

Ta/HfO_2 memristive crossbar array: In this structure, fully-connected layers and LSTM are implemented using a 1 transistor 1 resistor crossbar array with Ta/HfO_2 memristors placed on top of it.The main limitation of memristor-based LSTM networks is they need to be updated in the pre-training of networks' weight values [66].

FPGA-based LSTM is the growing research area and many researchers have proposed different architectures for it. The designs in [67] and [68] are considered to be representational work. The goal of [67] is to replace the LSTM training method with simultaneous perturbation stochastic approximation (SPSA), which is more suited to FPGA implementation. However, work in [42] focuses on speeding up the inference phase so that LSTM-RNN may be used in real-world applications. Chang et al. propose an FPGA-based 2-layer LSTM-RNN accelerator in [68]. During LSTM-RNN inference, their accelerator examines coarse-grained computing parallelism, and the data format chosen is 16-bit fixed point. In comparison to the work proposed by Chang, [42] employs floating point data, investigates both computational and communication optimizations, and obtains better results. Chang proposed a hardware version of LSTM network on Zynq 7020 FPGA from Xilinx with 2 layers and 128 hidden units in hardware to study the parallelism for RNN/LSTM [68]. This implementation performs 21 times faster than the Zynq 7020 FPGA's ARM Cortex-A9 CPU. Lee used highly parallel processing elements (PEs) to accelerate RNNs on an FPGA for low latency and high throughput [69]. These implementations did not support sparse LSTM networks, but the work in [54] supports sparse LSTM and can produce faster results.

The fully-memristive architectures limit to give endurance and accessible signal gain, to overcome these other resistive paradigms such as spintronic devices can be considered . SHE-MTJ is one such spintronic device that can be used in the architecture of LSTM to give better performance in predicting the output. One such SHE-MTJ based short term memory and long term memory circuit are designed in [14] that gives faster and more reliable functionality of a given circuit and also low energy consumption. If such a device is used in LSTM, the performance of the LSTM can be improved while having an energy efficient and low-area overhead circuit.

Summary

Numerous previous works on activation function unit, RNN, and LSTM are discussed in this chapter. These works mainly focused on hardware implementations using CMOS technology[51] [52], FPGA [54], ASIC [55], GPU [56], and some based on emerging technology [14] [21]. All these works overcome past challenges and proposed an improved version of the design, but these designs lead to further drawbacks which make a significant impact on the design. Drawbacks of most of the designs are mentioned in this chapter. Designs based on FPGA might lead to lower throughput due to low clock frequency. Designs using CMOS technology-based implementations have the drawback of high energy consumption and high area-overhead which encourage the use of emerging technology-based hardware implementations. Hence, this thesis developed a spin-based non-binary neuron which is discussed in CHAPTER FOUR: PROPOSED SPIN-BASED LSTM NETWORK WITH NON-BINARY NEURONS.

CHAPTER FOUR: PROPOSED SPIN-BASED LSTM NETWORK WITH NON-BINARY NEURONS

Binary and Non-Binary Neuron

For multiple gating applications, LSTM networks require sigmoid and tanh-based neurons. In an average time interval, the current-controlled p-bit device shows an analogous behavior to the sigmoid function. Circuit implementation of p-bit device is shown in Figure 9 (a). By connecting an inverter to VDD and GND, a sigmoidal behavior can be achieved. The sigmoid function output is indicated by the black dotted curve, and the p-bit output average is indicated by the red-circle curve in Figure 10 (b) which is almost the identical to sigmoid output curve. Similarly, the nonlinear hyperbolic tangent or tanh function can be designed using a sigmoid function like $\tanh(x) = 2\sigma(2x) - 1$ where output values lie between “+1” and “-1”. This can be obtained by connecting an inverter to VDD and -VDD in the p-bit device. In Figure 10 (c), the tanh function output is shown by the black dotted curve and the green-circle curve shows the time-averaged output of modified p-bit device (*tanh* (I_c)). The output of p-bit at each time step depends on the input; a zero input gives an output of either “-1” or “+1” with equal probability, a positive input I_c gives high probability to output a positive value and vice versa, as shown by the Figure 10.

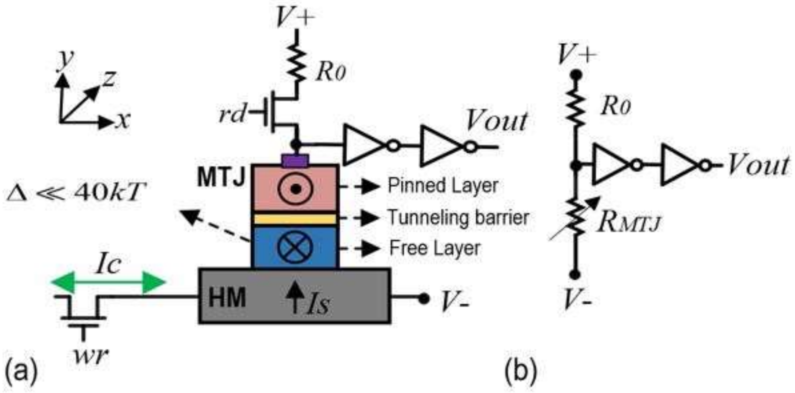


Figure 9: (a) The building block of non-binary neuron (p-bit), and (b) the equivalent read circuit [33]

Therefore, the time-averaged output of the p-bit device can provide both sigmoid and tanh function behaviors with slight modifications to the circuit designs. However, for practical purposes, p-bit device gives a binary output of either “0” or “1” at a given instance. On the contrary, ideal sigmoid and tanh functions do not have a limited binary state as outputs but vary within a limited range as per the input value. A novel complementary activation circuit and mechanism is required to use a p-bit device as a viable activation function to reach improved accuracy levels. The stochasticity of any p-bit device is highest for input current values near zero and decreases when the input current values reach their maximum or lowest levels. A non-binary neuron can be implemented using the behavior of a p-bit. The suggested design extracts this characteristic by repeatedly operating the p-bit device for the same input, resulting in a symmetric range of output voltages.

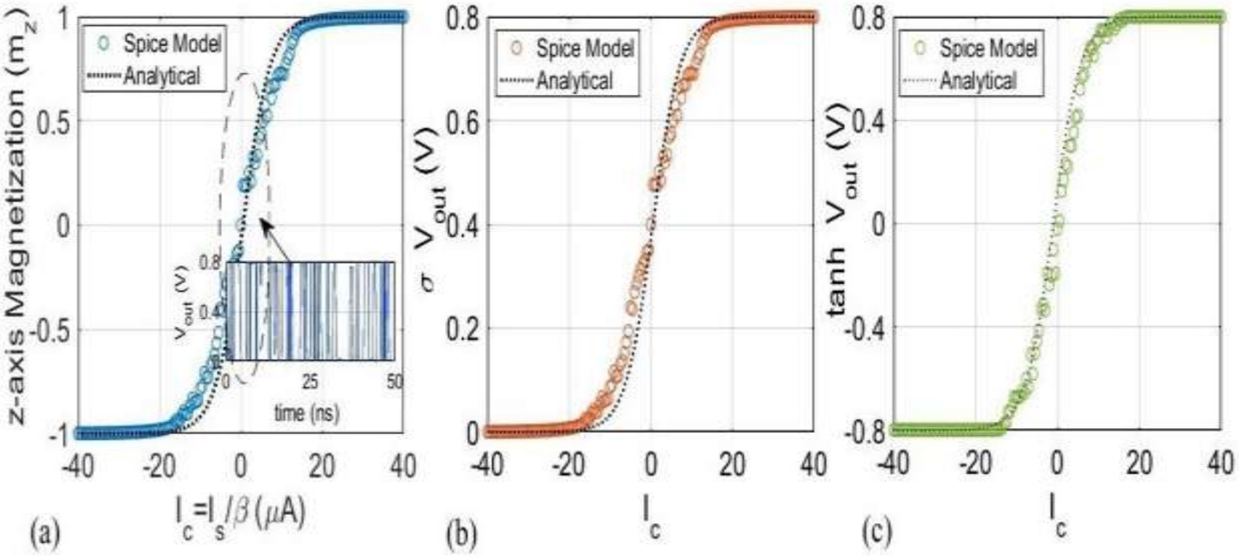


Figure 10: Time-averaged behavior of SHE-MTJ based p-bit device. (a) magnetazitaion fluctuation. (b) and (c) are the implemented sigmoid and tanh behaviors respectively.

The output voltages are stored and mapped to a low-overhead Look-Up Table (LUT) which consists of the voltage values. The low-power and low-area qualities of p-bit are preserved in this technology, which is used to generate an upgraded non-binary state. Figure 11 shows how we optimized the p-bit based stochastic neuron for hardware implementation by adding two components. A 4-bit buffer is inserted first to latch the output voltage of the p-bit circuit. The neuron output is formed using a LUT next. We synchronized the write/read access transistors on the p-bit device to avoid multiple crossbar computations. This strategy enables the design to maintain a consistent crossbar output current and apply it to the neuron unit as required. We consider two complement signals for wr and rd, as shown in Figure 11. The wr signal goes high for each sample, and the p-bit device is programmed based on the crossbar output current. The wr signal drops low and the rd signal goes high to read out the p-bit resistance and produce the output bit. The sampled floating-point activation values corresponding to output combinations in the buffer are then sent to the converter LUT, which is prestored with the 4-bit buffered data. If the buffer content is 001, for example, the LUT chooses -0.4 as the output. This value can be triggered by p-bit output bitstreams 0001/0010/0100/1000. This type of non-binary neuron design can be used in a wide range of ANN applications that require non-linear and deterministic tanh and sigmoid activation functions.

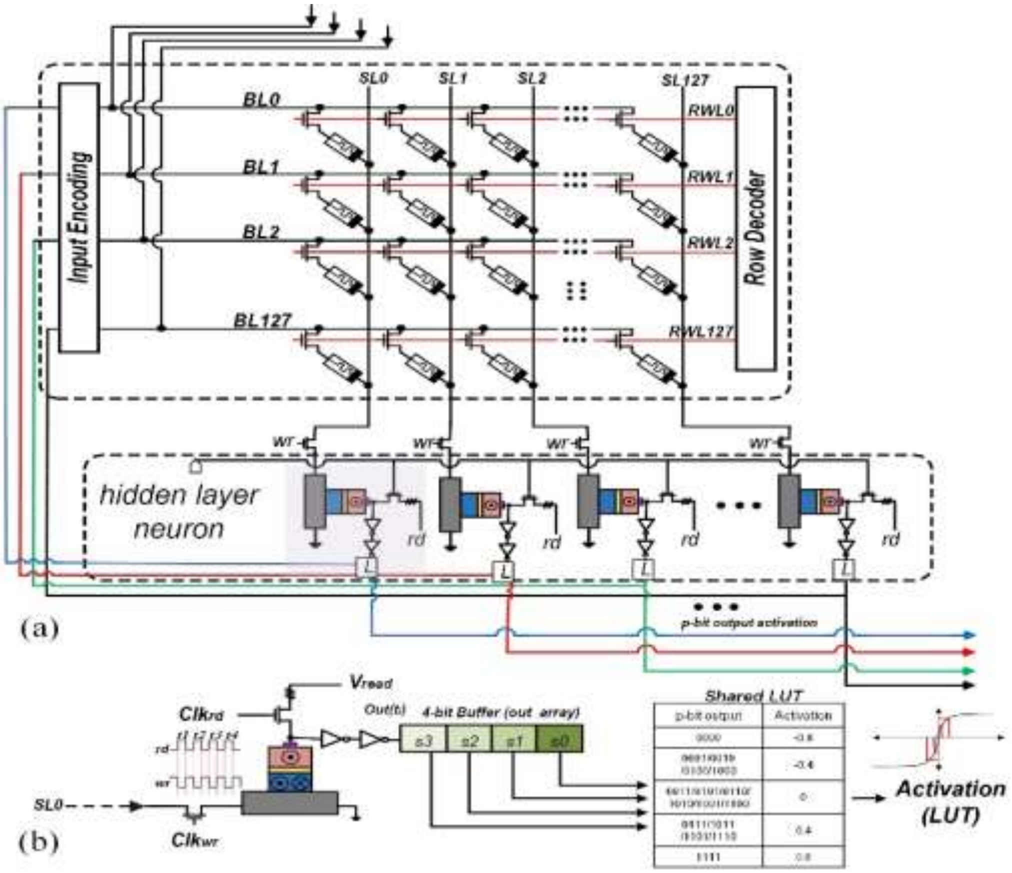


Figure 11: The proposed spin-based LSTM network with non-binary neurons

Circuit-Level Analysis

Starting with device-level modelling of memristive synapse and p-bit based neuron components, we analyze the suggested LSTM design performance. We used the Ag-Si memristor device parameters from the SPICE model for memristors. [70]. The Landau Lifshitz–Gilbert (LLG) equation is used to represent free layer magnetization dynamics, and the nonequilibrium Green's function (NEGF) is used to estimate the resistance range in the SHE-MTJ model (RP, RAP). The SPICE models of CMOS transistors and memristors are then combined under the 14nm PTM-MG library. In HSPICE, we created crossbar arrays of p-bit neurons in two sizes

(32x32, 128x128) at the circuit level. In Synopsys Design Compiler, we implemented all peripheral circuits, including row address decoders, array controllers, and so on. We used the popular names dataset accessible as a national data set to build three separate name predictor LSTM networks using ideal, binary, and the proposed non-binary neuron as application-level analysis.

Table 3: The Comparison of proposed non-binary neuron with CMOS-based designs

	32x32			**128x128**		
xbar Size	[52]	[51]	Here	[52]	[51]	Here
xbar #	68	68	68	5	5	5
Area (mm^2)	0.17	0.07	0.06	0.06	0.02	0.02
Energy (uJ)	N/A	4.04	0.14	N/A	1.03	0.03

Figure 12 shows the SPICE simulation waveforms of the p-bit based non-binary neuron, verifying its functionality. Under five input currents for four clock cycles, we assess the neuron output twice (p-bit 1 to p-bit 2). I_{sum} signifies the weighted sum of input currents realized by the resistive sub-array and flowing into the p-bit device, ranging from -50A to +50A. When the I_{sum} is -50A or +50A, the output of both p-bit devices is “1” or “0” for the full four clock cycles, confirming the neuron's deterministic behavior based on these charge currents. The LUT will eventually use these outputs to signify 0.8v and -0.8v. We see various outputs for each p-bit device when the I_{sum} is -5A. However, both outputs will be mapped to a common value in the future (-0.4v).

Additionally, we compared the area and energy consumption of the proposed design with [52] and [51] CMOS-based designs, under two distinct sub-array sizes as tabulated in Table 3: The Comparison of proposed non-binary neuron with CMOS-based designs. In comparison to CMOS-based non-binary designs, simulations demonstrate that our suggested neuron delivers up to a 34 percent increase in energy efficiency and a 2 percent reduction in area. The energy consumption results for [52] could not be appropriately reported.

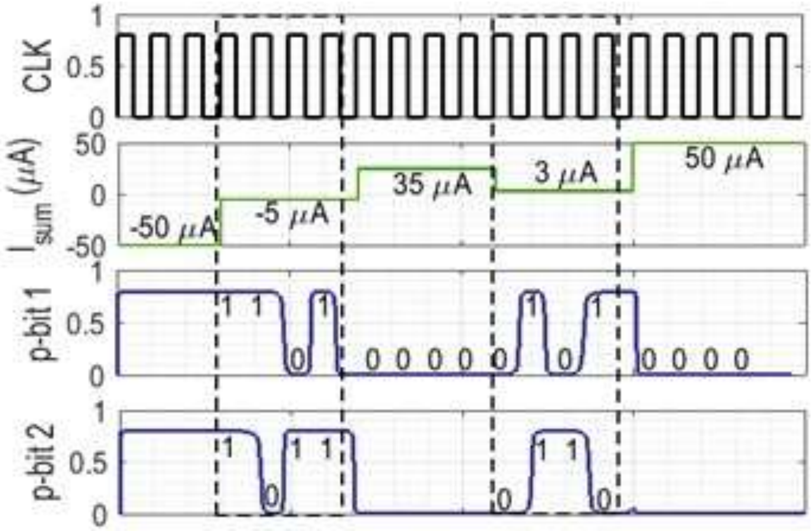


Figure 12: The transient simulation result of the neuron based on the crossbar SL current

Experimental Results

Figure 13 shows the experimental results for three distinct neuron designs including loss, perplexity, and accuracy fluctuations for all cases. Lower values are favored for loss and confusion parameters, unlike accuracy. An average of 30 training sample batch sets is plotted. The accuracy graph depicts the neural network's performance, whereas the perplexity graph assesses the network's current implementation in terms of sample data modelling. The ideal sigmoid neuron displays the limits on possibility with an approximation for all plots in Figure 13 (a). In the binary neuron case shown in Figure 13 (b), there is a sharp rise in accuracy in the first sets of batches. However, it initially does not reach the performance of the ideal sigmoidal model (Figure 13 (a)). As a result, the binary case's results show a lengthy tail that begins around set number 50, with the system steadily improving as it approaches the conclusion of the batches.

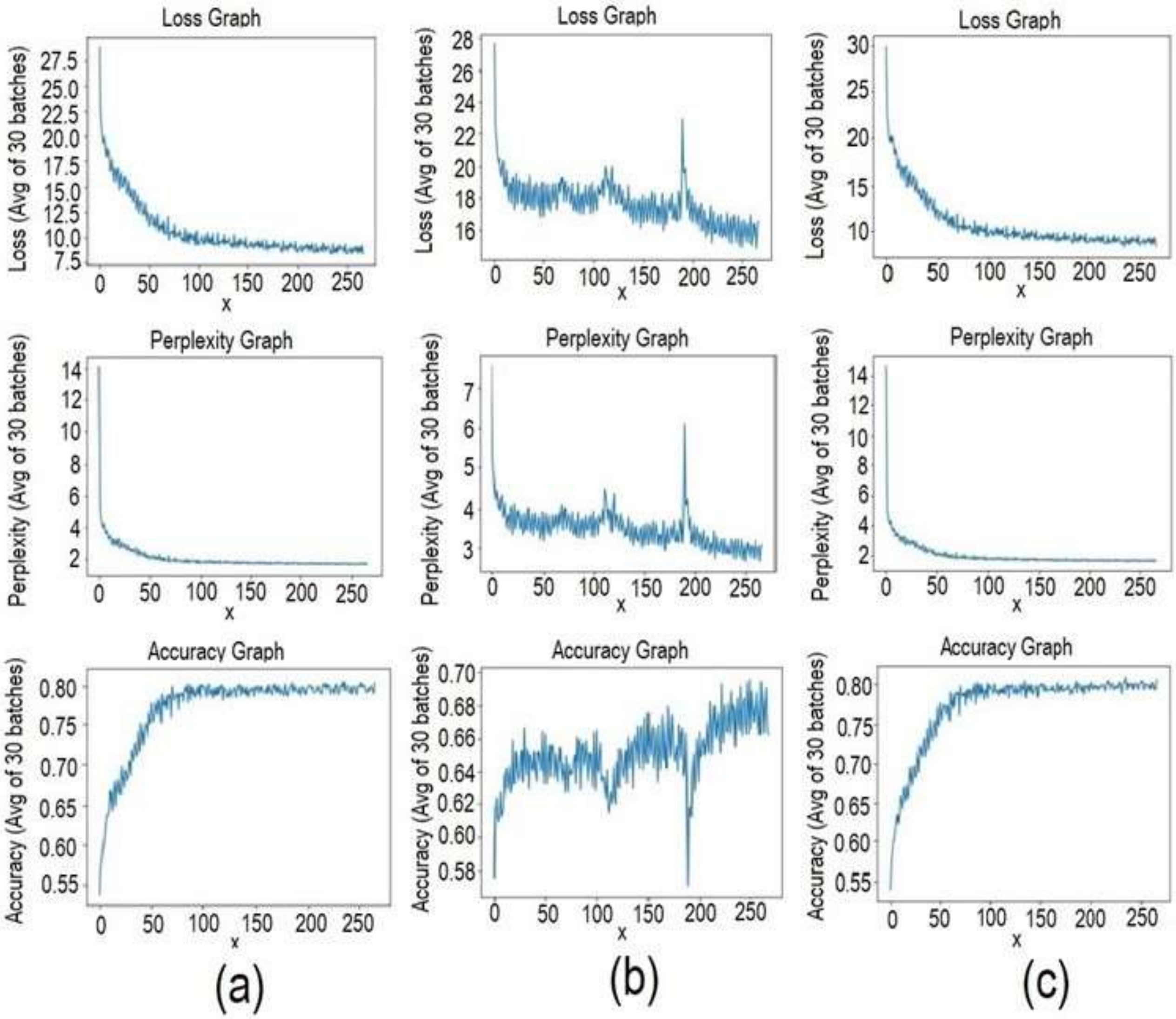


Figure 13: The experimental results of the LSTM network with (a) ideal, (b) binary and (c) proposed non-binary neuron.

Furthermore, the perplexity graph demonstrates that the training algorithm struggles to represent the samples using the network due to the disruption caused by discontinuity of the binary activation. When compared to the ideal sigmoid neuron, the network containing the binary neuron shows a 58% decline in data modelling after 8,000 training examples.

Utilizing the proposed non-binary neuron, the results are very close to the ideal case as shown in Figure 13. (c). The enhanced activation mechanism allows it to mimic the ideal sigmoidal system. The perplexity graphs converging to similar values, with the proposed non-binary neuron showing only a 7% degradation when compared to the sigmoidal system, reflects the same. The proposed neuron, however, like the binary activation function, starts with a significantly slower training pace. This tail, on the other hand, is substantially shorter, lasting only about 1,050 training samples.

Summary

This chapter describes the proposed hardware implementation of LSTM using ReRAM-based synaptic crossbar and a spin-based non-binary neuron. ReRAM-based synaptic crossbar arrays are inspired from the work proposed in [21] where they proposed a ReRAM-based crossbar RNN hardware implementation with feedback using spin-based Adjustable Probabilistic Activation Function (APAF) to achieve high energy- and area-efficiency, while keeping the accuracy loss and processing speed comparable with the baseline designs [21]. The proposed spin-based non-binary neuron activation function is designed using a spintronic device namely, probabilistic-bit (p-bit). Leveraging the stochastic property of a p-bit device, a spin-based activation function is designed that mimics the ideal sigmoidal and tanh activation functions thereby providing an energy efficient and low-area overhead design while maintaining comparable accuracy commensurate with the ideal case.

To verify the functionality of a p-bit as a non-binary neuron, a circuit level analysis is done by running a SPICE simulation and the output waveform verifies the functionality of a p-bit as a non-binary neuron. Under five input currents for four clock cycles, we evaluated the neuron output twice for p-bit 1 and p-bit 2. Figure 12 shows the transient simulation result of the neuron based on the crossbar SL current. Additionally, we compared the area and energy consumption of the proposed design with [52] and [51] CMOS-based designs, under two distinct sub-array sizes as tabulated in Table 3: The Comparison of proposed non-binary neuron with CMOS-based designs. In comparison to CMOS-based non-binary designs, simulations demonstrate that our suggested neuron delivers up to a 34 percent increase in energy efficiency and a 2 percent reduction in area. As application-level analysis, we have built three LSTM networks for distinct name prediction for popular baby names dataset (US Baby's First Names). These three LSTM networks uses ideal, binary, and the proposed non-binary neurons. The performance evaluation of an LSTM network utilizing ideal, binary, and the proposed non-binary neuron shows that the proposed neuron can achieve up to 85% accuracy and perplexity of 1.56, similar to algorithmic expectations of near-ideal neurons.

CHAPTER FIVE: CONCLUSION

Neuromorphic computing paradigms have lately produced excellent results in algorithm-level experiments, as detailed in prior chapters. These models conduct computationally critical operations on huge datasets, such as Multiplication and Addition (MAC), resulting in excessive power consumption and the well-known memory-wall challenge. Energy-efficient non-von-Neuman computing architectures that can be implemented with low-power devices are necessary to tackle these issues. There are a few researchers that suggest an architecture-level solution than there are algorithm-level advances in various neuromorphic paradigms. In some of the previous works proposed [21], three energy-efficient accelerators have been proposed for Generative Adversarial Networks (GANs), biologically-inspired networks, and recurrent neural networks. However, some additional neural networks, such as Long-Short Term Memory networks (LSTMs) and Spiking neural networks, are still in desperate need of a suitable hardware implementation to improve the efficiency and speed of training. These networks use unsupervised neural networks which are used to make predictions based on the time sequence of input data and use a memory unit to anticipate the future data based on the previous input present in the memory unit. In comparison to previous artificial neural networks, these networks interact with larger datasets and have a more complicated network topology. As a result, these networks require a high-performance hardware design that allows them to train faster and includes a non-volatile memory unit for the feedback component. For such devices, MRAM-based architecture is an excellent contender for providing both power efficiency and non-volatility. Hence, we proposed a stochastic activation function using SHE-MTJ based p-bit device which gives analogous behavior of a sigmoid function.

Technical Summary and Insights gained

To summarize, the major contributions of defense are listed as follows:

1. This proposal is majorly concentrated on the hardware implementation of Long Short-Term Memory using emerging technology non-volatile memory.
2. Despite there are several circuit level advancements in the Recurrent Neural Networks, there is a need for LSTM as it solves the vanishing gradients problem that occurs while using the backpropagation technique in RNNs.
3. Hardware implementation of LSTM was previously mainly concentrated towards CMOS technology-based implementations which may result in facing challenges of high energy consumption, memory wall problem, and volatility complications for standby modes.
4. In this thesis, we proposed a Long Short-Term Memory networks as a form of Recurrent Neural Network, with ReRAM-based synaptic crossbar arrays and spin-based non-binary neurons.
5. ReRAM-based synaptic crossbar arrays are inspired from the work proposed in [21] where they proposed a ReRAM-based crossbar RNN hardware implementation with feedback using spin-based Adjustable Probabilistic Activation Function (APAF) to achieve high energy- and area-efficiency, while

keeping the accuracy loss and processing speed comparable with the baseline designs [21].

6. The proposed spin-based non-binary neuron is designed using a spintronic device namely, probabilistic-bit (p-bit).
7. A probabilistic-bit (p-bit) is derived from a Spin Hall Effect – Magnetic Tunnel Junction (SHE-MTJ) with a design of 1-Transistor-with-1- SHE-MTJ structure as shown in Figure 9.
8. A SHE-MTJ based p-bit equivalent read circuit is shown in Figure 9(b), in which the reading operation is performed by delivering a low read voltage to the MTJ terminals (V+ and V-) to sense its resistance (R_{MTJ}). The RMTJ and the reference resistor R0 are then used to build a resistive voltage divider, with the reference resistor set to the MTJ average conductance (R_0-1=GP+GAP/2), where GAP and GP are the AP and P state conductances, respectively. The voltage from the voltage divider is fed into the CMOS inverters, and the output voltage (V_{out}) will stochastically oscillate between "0" and "1," with the input charge current controlling the probability of each value.
9. The magnetization fluctuations of a SHE-MTJ based p-bit device when observed on a time- averaged behavior as shown in Figure 10 is analogous to the output of a sigmoid activation function whose steady-state probability is modulated by an input current.

10. The proposed approach extracts the stochastic behavior of a p-bit device by running it numerous times for the same input and generating a symmetric range of output voltages. These output voltages are recorded and mapped to the voltage values in a low-overhead Look-Up Table (LUT). The low-power and low-area qualities of p-bit are preserved in this technique, which is used to generate an upgraded non-binary state.
11. To verify the functionality of a p-bit as a non-binary neuron a circuit level analysis is done by running a SPICE simulation and the output waveform verifies the functionality of a p-bit as a non-binary neuron. Under five input currents for four clock cycles, we evaluated the neuron output twice for p-bit 1 and p-bit 2. Figure 12 shows the transient simulation result of the neuron based on the crossbar SL current.
12. Additionally, we compared the area and energy consumption of the proposed design with [52] and [51] CMOS-based designs, under two distinct sub-array sizes as tabulated in Table 3: The Comparison of proposed non-binary neuron with CMOS-based designs. In comparison to CMOS-based non-binary designs, simulations demonstrate that our suggested neuron delivers up to a 34 percent increase in energy efficiency and a 2 percent reduction in area.
13. As application-level analysis, we have built three LSTM networks for distinct name prediction for popular baby names dataset (US Baby's First Names). These three LSTM networks uses ideal, binary, and the proposed non-binary neurons.

The performance evaluation of an LSTM network utilizing ideal, binary, and the proposed non-binary neuron shows that the proposed neuron can achieve up to 85% accuracy and perplexity of 1.56, similar to algorithmic expectations of near-ideal neurons.

The pertinent research flow is summarized in Figure 14:

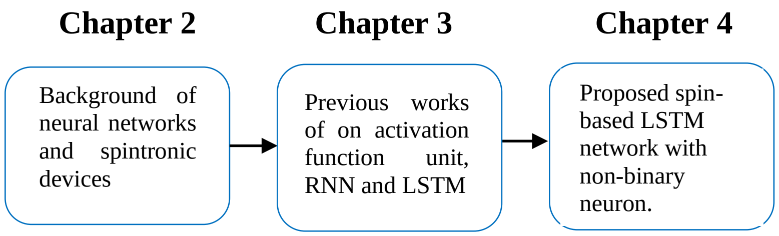

Chapter 2
Chapter 3
Chapter 4
Background of neural networks and spintronic devices
Previous works of on activation function unit, RNN and LSTM
Proposed spin-based LSTM network with non-binary neuron.

Figure 14: Logical organization of this thesis

Insights gained from the work presented are as follows:

- In their recurrent layer, both Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) networks feature a feedback loop to keep the information alive over time. LSTM, on the other hand, makes use of additional units such as a memory cell that can store data for long periods of time.
- When compared to CMOS-based technology architectures, spin-based architectures provide considerable energy efficiency benefits.
- Spin-based architectures offer reduction in area overhead when compared to CMOS-based technology architectures as they possess an ability to be fabricated vertically onto silicon and has high device density.

- A probabilistic-bit (p-bit) can be used to mimic an ideal sigmoid or hyperbolic tangent (tanh) in any neural networks.

Scope and Limitations

The scope of this proposal is to obtain a spin-based non-binary neuron for a LSTM network which performs in an efficient way when compared to CMOS technology-based LSTM network in the aspect of energy consumption and area-overhead. To obtain an energy efficient and low-area overhead design, we proposed a Long Short-Term Memory networks as a form of Recurrent Neural Network, with ReRAM-based synaptic crossbar arrays and spin-based non-binary neurons. A potential limitation of the design developed here could be its sensitivity to manufacturing process variation when produced on a commercial scale. Nonetheless, emerging technology will continue to become an increasingly relevant upcoming technology and as it becomes more commercially proven.

REFERENCES

1. Karnik, T., S. Borkar, and V. De. *Sub-90nm technologies: challenges and opportunities for CAD*. in *Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*. 2002.
2. Bahr, H. and R. DeMara, *OTBSAF Scalability on Pentium III/4 and Athlon 64/XP3000 Architectures.* MSIAC Modeling and Simulation Journal, on, 2005. **6**(3): p. 1-4.
3. Kawa, J., *FinFET design, manufacturability, and reliability.* Synopsys Design Ware Technical Bulletin, 2013.
4. Jang, H.B., et al., *Leveraging process variation for performance and energy: In the perspective of overclocking.* IEEE Transactions on Computers, 2012. **63**(5): p. 1316-1322.
5. Kawa, J. and A. Biddle, *Finfet: The promises and the challenges.* Synopsys Insight Newsletter, 2012(3).
6. De, V. *Energy efficient computing in nanoscale CMOS: Challenges and opportunities*. in *2014 IEEE Asian Solid-State Circuits Conference (A-SSCC)*. 2014. IEEE.
7. Khan, S. and S. Hamdioui. *NBTI Modeling in the Framework of Temperature Variation*. in *Proc. of Design and Test in Europe (DATE)*. 2010.
8. Davari, B., R.H. Dennard, and G.G. Shahidi, *CMOS scaling for high performance and low power-the next ten years.* Proceedings of the IEEE, 1995. **83**(4): p. 595-606.

9. Angizi, S. and D. Fan. *Redram: A reconfigurable processing-in-dram platform for accelerating bulk bit-wise operations*. in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. 2019. IEEE.

10. Fong, X., et al., *Spin-transfer torque devices for logic and memory: Prospects and perspectives.* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2015. **35**(1): p. 1-22.

11. Nikonov, D.E. and I.A. Young, *Overview of beyond-CMOS devices and a uniform methodology for their benchmarking.* Proceedings of the IEEE, 2013. **101**(12): p. 2498-2533.

12. Yu, S., *Neuro-inspired computing with emerging nonvolatile memorys.* Proceedings of the IEEE, 2018. **106**(2): p. 260-285.

13. Templeton, G., *What is Moore's Law?* 2015.

14. Sheikhfaal, S. and R.F. Demara, *Short-Term Long-Term Compute-in-Memory Architecture: A Hybrid Spin/CMOS Approach Supporting Intrinsic Consolidation.* IEEE Journal on Exploratory Solid-State Computational Devices and Circuits, 2020. **6**(1): p. 62-70.

15. Roohi, A. *Normally-off computing design methodology using spintronics: from device to architectures*. in *2020 11th International Green and Sustainable Computing Workshops (IGSC)*. 2020. IEEE.

16. Horowitz, M., *Energy table for 45nm process*, in *Stanford VLSI wiki*. 2014.

17. Silver, D., et al., *Mastering the game of go without human knowledge.* nature, 2017. **550**(7676): p. 354-359.

18. Chen, J. and X. Ran, *Deep Learning With Edge Computing: A Review.* Proceedings of the IEEE, 2019. **107**(8): p. 1655-1674.

19. Shi, W. and S. Dustdar, *The promise of edge computing.* Computer, 2016. **49**(5): p. 78-81.

20. He, Z., *Efficient and Secure Deep Learning Inference System: A Software and Hardware Co-Design Perspective*. 2020, Arizona State University.

21. Shadi Sheikhfaal, R.F.D., *Energy-Efficient Recurrent Neural Network with MRAM-based Probabilistic Activation Functions (revision pending).* IEEE Transactions on Emerging Topics in Computing (TETC), 2021.

22. Shadi Sheikhfaal, M.R.V., Adedoyin Adepegba, and Ronald F. DeMara, *Long Short-Term Memory with Spin-Based Binary and Non-Binary Neurons (accepted)*, in *Midwest Symposium on Circuits and Systems*. 2021, IEEE explore: Virtual.

23. Chung, J., et al., *Empirical evaluation of gated recurrent neural networks on sequence modeling.* arXiv preprint arXiv:1412.3555, 2014.

24. Long, Y., et al. *Reram crossbar based recurrent neural network for human activity detection*. in *2016 International Joint Conference on Neural Networks (IJCNN)*. 2016. IEEE.

25. Fang, Z., et al., *Predicting flood susceptibility using LSTM neural networks.* Journal of Hydrology, 2021. **594**: p. 125734.

26. Bengio, Y., P. Simard, and P. Frasconi, *Learning long-term dependencies with gradient descent is difficult.* IEEE transactions on neural networks, 1994. **5**(2): p. 157-166.

27. Greff, K., et al., *LSTM: A search space odyssey.* IEEE transactions on neural networks and learning systems, 2016. **28**(10): p. 2222-2232.

28. Smagulova, K., O. Krestinskaya, and A.P. James, *A memristor-based long short term memory circuit.* Analog Integrated Circuits and Signal Processing, 2018. **95**(3): p. 467-472.

29. Long, Y., T. Na, and S. Mukhopadhyay, *ReRAM-based processing-in-memory architecture for recurrent neural network acceleration.* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2018. **26**(12): p. 2781-2794.

30. Brownlee, J., *A gentle introduction to the rectified linear unit (ReLU).* Machine learning mastery, 2019. **6**.

31. Liu, L., et al., *Spin-torque ferromagnetic resonance induced by the spin Hall effect.* Physical review letters, 2011. **106**(3): p. 036601.

32. Pai, C.-F., et al., *Spin transfer torque devices utilizing the giant spin Hall effect of tungsten.* Applied Physics Letters, 2012. **101**(12): p. 122404.

33. Camsari, K.Y., et al., *Stochastic p-bits for invertible logic.* Physical Review X, 2017. **7**(3): p. 031014.

34. Roohi, A., et al., *Voltage-based concatenatable full adder using spin hall effect switching.* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2017. **36**(12): p. 2134-2138.

35. Manipatruni, S., D.E. Nikonov, and I.A. Young, *Energy-delay performance of giant spin Hall effect switching for dense magnetic memory.* Applied Physics Express, 2014. **7**(10): p. 103001.

36. Liu, L., et al., *Spin-torque switching with the giant spin Hall effect of tantalum.* Science, 2012. **336**(6081): p. 555-558.

37. Andrawis, R., A. Jaiswal, and K. Roy, *Design and comparative analysis of spintronic memories based on current and voltage driven switching.* IEEE Transactions on Electron Devices, 2018. **65**(7): p. 2682-2693.

38. Zand, R., et al. *Low-energy deep belief networks using intrinsic sigmoidal spintronic-based probabilistic neurons*. in *Proceedings of the 2018 on Great Lakes Symposium on VLSI*. 2018.

39. Camsari, K.Y., S. Ganguly, and S. Datta, *Modular approach to spintronics.* Scientific reports, 2015. **5**(1): p. 1-13.

40. contributors, W., *Gated recurrent unit. In Wikipedia, The Free Encyclopedia.* 2020. p. from https://en.wikipedia.org/w/index.php?title=Gated_recurrent_unit&oldid=997015931.

41. contributors, W., *Spintronics*. 2021, Wikipedia, The Free Encyclopedia. p. https://en.wikipedia.org/w/index.php?title=Spintronics&oldid=1025008108.

42. Guan, Y., et al. *FPGA-based accelerator for long short-term memory recurrent neural networks*. in *2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC)*. 2017. IEEE.

43. Adam, K., K. Smagulova, and A.P. James. *Memristive LSTM network hardware architecture for time-series predictive modeling problems*. in *2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*. 2018. IEEE.

44. Farmahini-Farahani, A., et al. *NDA: Near-DRAM acceleration architecture leveraging commodity DRAM devices and standard memory modules*. in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*. 2015. IEEE.

45. Li, S., et al. *Pinatubo: A processing-in-memory architecture for bulk bitwise operations in emerging non-volatile memories*. in *Proceedings of the 53rd Annual Design Automation Conference*. 2016.

46. Burr, G.W., et al., *Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) using phase-change memory as the synaptic weight element.* IEEE Transactions on Electron Devices, 2015. **62**(11): p. 3498-3507.

47. Wong, H.-S.P., et al., *Metal–oxide RRAM.* Proceedings of the IEEE, 2012. **100**(6): p. 1951-1970.

48. Vincent, A.F., et al., *Spin-transfer torque magnetic memory as a stochastic memristive synapse for neuromorphic systems.* IEEE transactions on biomedical circuits and systems, 2015. **9**(2): p. 166-174.

49. Angizi, S., et al. *Accelerating deep neural networks in processing-in-memory platforms: Analog or digital approach?* in *2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. 2019. IEEE.

50. Roohi, A., et al., *Apgan: Approximate gan for robust low energy learning from imprecise components.* IEEE Transactions on Computers, 2019. **69**(3): p. 349-360.

51. Ardakani, A., et al., *VLSI implementation of deep neural network using integral stochastic computing.* IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2017. **25**(10): p. 2688-2699.

52. Bojnordi, M.N. and E. Ipek. *Memristive boltzmann machine: A hardware accelerator for combinatorial optimization and deep learning*. in *2016 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. 2016. IEEE.

53. Price, M., J. Glass, and A.P. Chandrakasan. *14.4 A scalable speech recognizer with deep-neural-network acoustic models and voice-activated power gating*. in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 2017. IEEE.

54. Han, S., et al. *Ese: Efficient speech recognition engine with sparse lstm on fpga*. in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. 2017.

55. Shin, D., et al. *14.2 DNPU: An 8.1 TOPS/W reconfigurable CNN-RNN processor for general-purpose deep neural networks*. in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 2017. IEEE.

56. Sak, H., A. Senior, and F. Beaufays, *Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition.* arXiv preprint arXiv:1402.1128, 2014.

57. Teich, W., et al. *Towards an efficient hardware implementation of recurrent neural network based multiuser detection*. in *2000 IEEE Sixth International Symposium on Spread Spectrum Techniques and Applications. ISSTA 2000. Proceedings (Cat. No. 00TH8536)*. 2000. IEEE.

58. Maeda, Y. and M. Wakamura, *Simultaneous perturbation learning rule for recurrent neural networks and its FPGA implementation.* IEEE Transactions on Neural Networks, 2005. **16**(6): p. 1664-1672.

59. Li, S., et al. *Fpga acceleration of recurrent neural network based language model*. in *2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines*. 2015. IEEE.

60. Chen, T., et al., *Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning.* ACM SIGARCH Computer Architecture News, 2014. **42**(1): p. 269-284.

61. Chen, Y., et al. *Dadiannao: A machine-learning supercomputer*. in *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*. 2014. IEEE.

62. Du, Z., et al. *ShiDianNao: Shifting vision processing closer to the sensor*. in *Proceedings of the 42nd Annual International Symposium on Computer Architecture*. 2015.

63. Chi, P., et al., *Prime: A novel processing-in-memory architecture for neural network computation in reram-based main memory.* ACM SIGARCH Computer Architecture News, 2016. **44**(3): p. 27-39.

64. *Memristor*. p. available at https://www.nanowerk.com/memristor.php.

65. Smagulova, K., et al. *Design of cmos-memristor circuits for lstm architecture*. in *2018 IEEE international conference on electron devices and solid state circuits (EDSSC)*. 2018. IEEE.

66. Smagulova, K. and A.P. James, *A survey on LSTM memristive neural network architectures and applications.* The European Physical Journal Special Topics, 2019. **228**(10): p. 2313-2324.

67. Tavcar, R., et al., *Transforming the LSTM training algorithm for efficient FPGA-based adaptive control of nonlinear dynamic systems.* INFORMACIJE MIDEM-JOURNAL OF MICROELECTRONICS ELECTRONIC COMPONENTS AND MATERIALS, 2013. **43**(2): p. 131-138.

68. Chang, A.X.M., B. Martini, and E. Culurciello, *Recurrent neural networks hardware implementation on FPGA.* arXiv preprint arXiv:1511.05552, 2015.

69. Lee, M., et al. *FPGA-based low-power speech recognition with recurrent neural networks*. in *2016 IEEE International Workshop on Signal Processing Systems (SiPS)*. 2016. IEEE.

70. Gao, L., F. Alibart, and D.B. Strukov. *Analog-input analog-weight dot-product operation with Ag/a-Si/Pt memristive devices*. in *2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC)*. 2012. IEEE.