Memristive Neural Networks Application In Predicting of Health Disorders

Isha Baokar and Lili He, *Member, IAENG*

***Abstract*— This project focuses on designing a Memristive neural network (MNN) which proves to offer the same level of accuracy when compared with CMOS based DNN but consumes low power. The designed MNNs are simulated using modern CAD simulation frameworks for 22nm technology node and compared for the total leakage power with existing CMOS based neural network systems. The simulated MNNs estimate a leakage power of ~131 µW which is less compared to CMOS based neural network's power and an area of ~10mm^2 which is comparable with the size constraints for CMOS based implementations as studied.**

***Index Terms*—Memristor, Neural networks, CMOS based implantation**

I. Introduction

Memristors are the newest added element of the electric circuitry in addition to the resistor, capacitor and inductor. "Memristor" as the name signifies are "memory resistors" that have a prominent characteristic of maintaining the relationship between current and voltage across different time integrals in a two-terminal element thereby making them a passive circuit. The resistance of memristor is based on the amount of previous charge that flowed through it. Memristors have another characteristic of polarity, and if current is flowing in the same direction, the resistance tends to decrease but if we reverse the current flow the resistance tends to increase. Memristors were first identified by Leon Chua, in the year 1971. Chua is a non-linear circuit theorist from UC Berkeley. He discussed his findings in a research paper that he presented on Transactions on Circuit Theory [1-2]. Chua suggested that there is a significant relationship between flux and voltage change in any passive circuit. According to him, there are four basic fundamental variables, voltage (v), charge (q), current(i) and flux linkage (ϕ). The resistor defines the relation between (v) & (i), the capacitor defines the relation between (q) & (v), the inductor between (i) & (ϕ) the only relationship which remained unexplained was that of between (ϕ) & (q) which was the basis of the discovery of the Memristor. Since their discovery, it was in the year 2008 that a team at HP Labs led by Stanley Williams developed the first working memristor prototype [2].

Manuscript received March 6, 2023; revised May 30, 2023.
Ms. Isha Baokar (email: isha.baokar@sjsu.edu) was graduate student at The Department of Electrical Engineering, San Jose State University, San Jose, CA, USA
Dr. Lili He (email: lili.he@sjsu.edu) is a professor at the Department of Electrical Engineering, San Jose State University, San Jose, California, USA.

Memristors gained popularity owing to its capability to store state and perform efficient computing which make them a best candidate for high volume data processing used widely in AI and ML applications. Since memristors have the capability to retain their current state even when there is not power supply, makes them a great source for implementing NVM (Non-Volatile Memory).

The relevance of Moore's law seems to fade over a past few years due to the CMOS scaling limits being reached due to the transistor size reaching the atomic limits. To overcome this hurdle the semiconductor industry has moved on to idea of "Beyond CMOS" since the early 2000's which rely on devices like CNT's (Carbon Nano Tubes), several 2D materials and Memristors which currently do not face scaling issues and posses' quality of hybrid integrations [3]. Other devices like RRAM's (Resistive Random-Access Memory) have gained popularity due to their ability to handle future storage and high compute. Another important scaling roadblock faced by the semiconductor industry is famously known as - "Von-Neumann Bottleneck". With the growing trend of big data Analytics, a new form of computing has rose to popularity widely known as "Neuromorphic Computing" which overcomes the scaling and bottleneck challenges as described above by making use of devices such as Memristors, which have the ability to compute and co-locate memory in same physical device. Also, memristors have the ability of performing analog switching making them the best candidate for performing neuromorphic computing. Fig.1 shows a typical crossbar circuit for memristor A tabular representation of the used parameter values for the VTEAM model is shown in Table 2 below. Fig. 6 is the I-V plot using the data from Table 2.

.

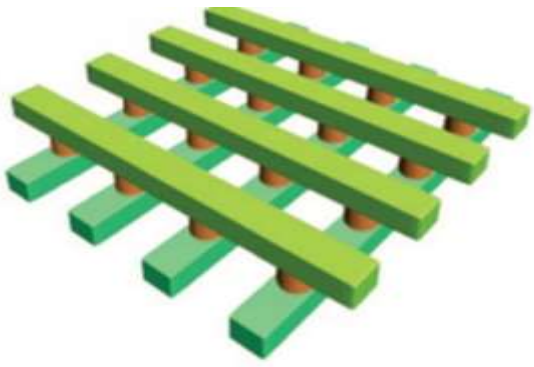

Fig. 1. A crossbar Circuit for Memristor

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. Memristor Devices

The most widely used device for memristor fabrication in an MIM setting is WO_x memristor. The W which is known as the bottom electrode has the function to form a partially oxidized WO_x layer which is the switching layer as shown in figure 3. The elements like Pd (Palladium) and Au (Gold) are deposited on top electrode. Finally, the silicon dioxide is applied to provide good area for the TE at the different cross points along with prevention from resistive switching regions [4]. For the standard memristor device fabrication process, a deposit of thin W film of around 60nm on a silicon substrate is done using processes like radio frequency and sputtering which are performed at ambient temperatures.

As discussed in the above sections memristor hold different properties depending on the level of oxidation and the material used for fabrication. Apart from that there are several widely used memristor models for carrying out simulations and establishing linear or non-linear relationship between state change and the current or voltage in the memristor. Depending on the application requirement different models are used to tune the memristor parameters and study the change of these on the memristor performance. The different memristor models are explained in the below sections.

(A)Linear Ion Drift Memristor Model

The linear ion drift model was established and developed by R.S Williams in HP Labs [5]. This model is based on the fact that there exists a linear relationship between the voltage and the derivative of state change. As shown in Fig.2, the width (D) of the model is divided into two regions comprising of doped and undoped TiO2, where a small amount of the total width (w) ha positive doped ions and has low resistance which in turn results in high conductance and the remaining region is left undoped.

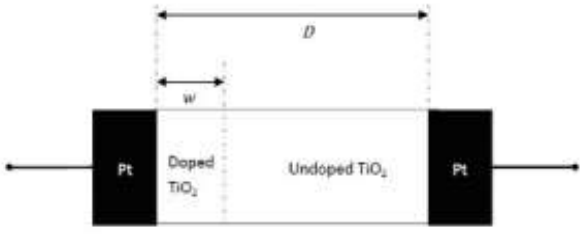

Fig. 2. Linear Ion Drift Memristor Model

In this model, it is assumed that the memristor have a certain level of ohmic conductance along with average ion mobility and a uniform field. Furthermore, the linear ion drift model predicts an inverse relationship between the applied voltage ($V_{o)}$ and the switching time (T_o). Hence the relationship can be mathematically put as $V_o \propto 1/T_o$.

(B)Non-Linear Ion Drift Memristor Model

This model is based on the fact that even though the linear ion drift model satisfies the relationship between the current/voltage and state change but still there exist a few non-linearity in the fabricated memristor devices. The non-linearity of the fabricated device is due to the change in voltage levels making this model widely used for modelling logic gates. This model is the most accurate when it comes to predicting both the static and dynamic behavior of the memristor. The model proves that there exists a non-linear relationship between the state change and voltage/current in an asymmetric switching behavior. This model highlights the hysteresis characteristic of the memristor. Finally, this model is mathematically formulated on the fact that the memristor is voltage-controlled device which has non-linear dependency with voltage and the state derivative which is given by

$$i(t) = wn\,(t)\beta \sinh(\alpha v(t)) + \chi\,[\exp(\gamma v\,(t) - 1]$$

Where the α, β, γ, and χ are known as the fitting parameters which are found experimentally and n determines the effect of state variable on the current.

(C) Crossbar Arrays

The crossbar array is a circuit combination which was first introduced by HP Labs on a computer called "Teramac". The design of the computer was based on the fact that as we lower the technology node to the nm scale, it becomes seemingly difficult to obtain a high number of fabrication yield, and with increased density of components on the chip, it often becomes difficult to achieve a zero dppm due to the immense wiring between different components on the chip. To overcome this issue, crossbar array posed to be a new architectural design with a possibility to provide new techniques to achieve network connectivity and efficient computation ability. Amongst the first few designs proposed, was the tree architecture designs, more precisely known as the fat-tree design where the interconnections between different nodes pose a lot of redundancy. Whereas each of the interconnects have the ability to be programmed subject to the algorithm requirements and the defect status at each interconnection. Fig. 3 shows the diagrammatical representation of how each interconnection point can be programmed.

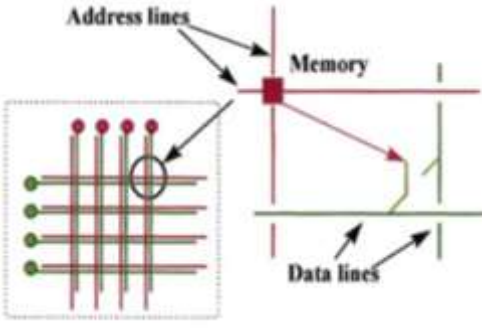

Fig. 3. Crossbar Architecture

This figure shows the enlarged view of the single layer and the junction node. The red block in Fig. 3 shows memory as a 3-terminal device which controls connection between the red and green lines depending upon the input combinations in the above figure. In the ideal computer system by HP Labs, this architecture was implemented using

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the six, input combinations and a lookup table to see the value of the output. The main idea behind this architecture was to stack up several layers of architecture which proved to be highly fault tolerant to fabricated defects.

III. Memristive Crossbar Arrays

Memristive crossbar arrays are formed with wires interconnecting each other at orthogonal intersection with nano-scale sized memristor devices at the intersection. The Memristive crossbar arrays have several advantages depending upon their architectural setting, they are very dense and small which makes them of great use for developing more capacity of memories and for computational purposes. One great advantage of these crossbars array is that they can be put on top of conventional CMOS based chips, wherein the CMOS circuits are responsible for complex processing tasks while the crossbar array is responsible for easy and parallel computation tasks. Combining these two architectures or techniques has led to immense development in terms of computation and hardware acceleration. Further this has led to development of artificial neural networks which are fast but consume very less power along with for other machine learning algorithms.

Memristive programming is basically divided into two broad categories – Unregulated and Regulated Write.

1)Unregulated Write – In this method of programming the memristor, the pulse is propagated into the device without observing the change in value of conductance of the memristor. This method of programming is helpful when memristors have 2-states of digital memory. It however poses a challenge when it comes to devices with more than two bits or analog signals.

2)Regulated Write- In this method of programming, the state of the device is already updated beforehand using certain level of programming and subsequently each pulse is observed after the read operation to verify whether the desired conductance value is attained or not. This is a more sophisticated approach to programming memristors and is often not economically feasible for crossbar array configuration of memristors.

A generic RC combination forming an Memristive crossbar array is shown in figure 10 below.

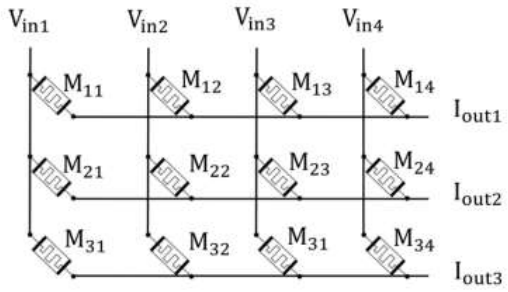

Fig 4. A Memristor Crossbar (3x4)

In fig. 4, input is applied to the top of crossbar which can be digital or analog depending on the application it is being used for. Now, if we held the output nodes at a constant of zero volts, the putout current equation can be given as –

$$I_{out1} = V_{in1}/ M_{11} + V_{in2}/ M_{12} + V_{in3}/ M_{13} + V_{in4} /M_{14}$$

Where M_{xy} is memristor resistance. The resistance in a memristor is equivalent to weights in a neural network, where the high resistance equals to the low weight. This crossbar architecture is suitable for complex and compact designs when there a multi-layer network, wherein the outputs can be given to different activation functions of a single neuron in the neuron layer [8]. Significantly speaking a single layer in any neural network equals to one crossbar which allows the efficient computation of matrix multiplication though its analog changes in voltage and resistance values. That said, multiple crossbars can be used to create a complex neural network or a feed forward network of different RC combinations.

(A)Neural Networks

Neural Networks also known as NN's are basically described as a collection of algorithms which are loosely modelled resembling the human brain structure with a basic function to recognize patterns. Nowadays NN's are mostly associated with artificial intelligence and machine learning. The McCulloch-Pitts model is the first model which forms the basis of Neural Networks and basic formulations in terms of neural activity calculated, which is used in a varying set of fields such as artificial intelligence, machine learning, computer science etc. Neural Networks comprise of three layers which are input layer, hidden layer and output layer, and as the name suggests the input layer consists of the input data to the neural network, the hidden layers are where the actual computation of the neural network takes place and finally the output layer is where the computation results are produced by the neural network. Figure 5 describes the layers in a Neural Network.

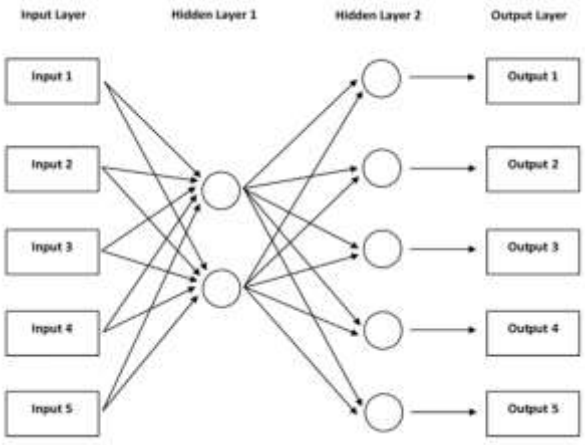

Fig 5. NN Layers

The neural networks are huge networks with several inputs and are trained first on dense datasets which consist of high number of datapoints and features. After the neural networks are trained on the samples of data, it learns from that and helps in predicting a similar pattern for future data. Neural networks have great scope in applications ranging from artificial intelligence to machine learning. Neural networks are used in predicting weather forecast or face and voice recognition.

A widely known application of neural networks is in

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the healthcare industry where neural networks are used to understand specific patterns of symptoms owing to a disease to development of new drugs for cure of several diseases. With the advancement of different types of neural networks, DNN's (Deep Neural Network) are widely used to understand medical scans from pathology tests, endoscopy and electrocardiograms. The sensors in the smart watches detects the heart beats in the user and any deviation from normal activity records an alert to the user. Several algorithms are used to track and examine the glucose levels in patients with diabetes and alert if any irregular observation is registered. As the medical industry is expanding in terms of providing more precise and accurate medical diagnosis through new technology advances in terms of biomedical devices, imaging plays a major role in source of medical diagnosis may it be in the form of X-ray, CT scan, MRI or mammogram. The medical industry analyzed that imaging can detect any disease way before the actual symptoms start to show, and for certain life-threatening diseases like cancer, early prevention and diagnosis can increases the chances of survival. In this context, CNN's or Convolutional Neural Networks have proved to be a great aid.

The CNN algorithm is designed to handle complex multi-class classification problem and a binary classification problem. There are several such life-threatening diseases such as osteoarthritis and diabetic retinopathy which are detected way ahead in the diagnosis cycle using several deep learning neural network architectures which makes them of great use in the healthcare industry.

(*B) Simulation Framework for Memristive Neural Network*

As described in the above sections, memristive crossbar architecture have been widely used in designing neural networks since they reduce time and complexity of several vector matrix multiplications from $O(n^2)$ to $O(n)$. This reduction has helped in accelerating and improving the power efficiency of the neural network. There are a few available open-sourced simulation frameworks to compute the efficiency of any memristive neural network along with several other features. Below is a comparison table of the different simulation frameworks for memristive DNNs in terms of ability to compute power and area of the network, capability to support non-linear properties and ability to convert a pretrained DNN to a memristive based DNN.

As we see from table 1, Memtorch and DNN+Neurosim are open-sourced simulation frameworks available. As researched from previous works, DNN+Neurosim [17] integrate the PyTorch and TensorFlow to provide capability of non-ideal characteristics of MNNs. On the other hand, Memtorch [20] provides co-simulation of the non-ideal device characteristics along with the common device models for higher flexibility and provide parameters for process variance. The Memtorch simulation framework is designed using C++, Python and CUDA, performance sensitive tasks are executed using C++ or CUDA for GPU specific execution. Memtorch is extensively based on PyTorch [21]. However, Memtorch does not provide the functionality to estimate the area and leakage power of the network which is overcome by DNN+ Neurosim [17]. DNN+ Neurosim is a compute in memory framework for DNNs, providing design options ranging from device to algorithm level. It includes a python wrapper similar to Memtorch to interface with PyTorch. DNN + Neurosim provides algorithm to hardware mapping along with evaluation of chip area, throughput of the network and energy efficiency during training and inference and also during developing hardware constraints for the network [17]. It provides support for designing flexible DNNs using device technologies such as CMOS based and beyond CMOS based. The work presented in this report focuses on using DNN+ Neurosim simulation framework over Memtorch, because the former has the capability to compute chip area and power leakage of the circuit along with, behavior modeling in terms of non-linearity and asymmetry, provide device-to-device and cycle-to-cycle variation for optimization of nonlinearities in networks which makes it a better candidate in simulating analog synaptic device based neural networks.

Table 1. Comparison of different simulation framework

Simulation Framework	Pretrained DNN Conversiom	Non-Liner Property	Device Faults	Open Source	Built-in Area/Power estimation
DNN+Neurosim [17]	Yes	Partial	No	Yes	Yes
MnSIM[18]	Yes	Partial	No	No	No
DL-RSIM[19]	Yes	Yes	No	No	No
Memtorch[20]	Yes	Partial	Yes	Yes	No
ReRAM-based DNN Accelerator	Yes	Yes	No	No	No
RAPIDNN[22]	Yes	No	No	No	No
PUMA[20]	Yes	No	Yes	No	No

IV. METHODOLOGY

(A)Dataset Selection and Description

In order to design and simulate a memristive neural network, the first step is to perform dataset selection in terms of type of dataset, number of input and output parameters, number of datapoints and features. As described in the previous sections, two simulation frameworks are narrowed down to build and simulate memristive neural networks i.e., MemTorch and DNN+ Neurosim. Since the scope of this project is designing memristive neural networks for predicting healthcare disorders, datasets targeting basic health disorders are selected and described in the below sections.

(B)Diabetes Prediction Dataset

The "PIMA Indian Diabetes Dataset" is taken from the National Institute of Diabetes and Digestive and Kidney Diseases. The dataset aims to classify a yes or no for a patient in terms of his diabetes levels, i.e., does the patient being diagnosed has diabetes or not. In order to predict or classify the diabetes condition of a patient many parameters are taken into account. The PIMA dataset focuses basically on females of minimum age of twenty-one years of the Indian heritage. This dataset has eight features or parameters which are – Glucose levels, Pregnancies which indicate the number of times the females has got pregnant, the blood pressure levels, insulin levels, skin thickness in terms of skin folds for triceps, BMI ratio and diabetes function age [22].

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

C)Human Activity Recognition Dataset

The HAR dataset is taken from University of California, Irvine which addresses the basic human activities in a modern day or ambient living environment. This dataset is collected by recording the basic everyday activities which include laying, sitting, walking etc.[23]. These recordings are collected by a smartphone which is given to the patients, this smartphone has several features such as highly sensitive sensors to catch any slightest activity of the patient. The patients participated in the data collection have an age range between nineteen to forty-eight years of age. The signals obtained from the smartphone are polished before proceeding with data cleaning. Finally, the dataset is divided into one twenty-eight datapoints and divided in two parts for test and train data.

As described in the above sections, two simulation frameworks are identified that support converting the neural network layer defined in PyTorch to their corresponding memristor based layers. In general, the linear and convolutional layers in the neural network can be implemented using a combination of MAC (multiple and accumulate) and VMM (vector matrix multiplication) operations. These specific operations can be calculated in memory using memristive crossbar arrays. In such arrays, the weight corresponding to the neurons can be modelled as resistance or conductance values and inputs to these neurons are modelled as *wl* voltages or power supply voltages.

Several CAD tools have been developed to convert neural network to memristive neural network or memristor tied architectures. SPICE simulation tools can also be used, but they are more commonly used for analog circuit simulations. In order to simulate large crossbar arrays, CAD tools that support conversion of neural network layers defined in modern ML frameworks like PyTorch and TensorFlow are usually preferred to simulated MNNs. CAD tools can also be sued to simulate the training and inference processes in traditional DNNs, and to estimate the area and power characteristics. These allow us to approximate the overall efficiency of the simulated MNN, apart from the accuracy of the network in comparison to CMOS- based DNNs.

Based on the above research, MemTorch and DNN + Neurosim are the two selected simulation frameworks. The purpose of selecting these is to prove than MNNs hold a promising avenue as an alternative to CMOS implemented DNNs without a significant loss in accuracy. The application of such MNNs can be useful in the healthcare industry or in wearable health devices, where chip area, accuracy and power consumption play a key role. The area and power characteristics are estimated using DNN + Neurosim framework where the memristor specific device properties are referred from the default values taken in [17].

V. Simulation

This section describes the training of neural networks and the transformation of these conventional networks into memristive neural networks. The development of neural networks and their conversion into MNNs is done using two datasets and the mentioned simulation frameworks described in [17] and [20]. As per the datasets described in section 5.1, neural network models are trained with models implemented using the PyTorch framework. The goal of the neural network model is not to achieve or reproduce the highest accuracy presented in research, but to design models with linear and convolution layers, and achieve similar accuracies when the PyTorch model is transformed to a memristive neural network architecture using memristor characteristics used in past research.

For the first dataset, Pima Indian Diabetes Dataset, a DNN model is constructed using layers connected linearly, and a sigmoid function is applied on each neuron. The dataset is split into separate training and test datasets in a 2:1 ratio. Using 16 hidden nodes and 100000 epochs, the best accuracy derived from the implemented model is 67.2% when inferred from the test dataset.

```
import torch.nn as nn
import torch


class Model(nn.Module):

    def __init__(self, nodes):
        super().__init__()
        self.hidden_linear = nn.Linear(8, nodes)
        self.output_linear = nn.Linear(nodes, 1)
        self.sigmoid = nn.Sigmoid()

    def forward(self, X):
        hidden_output = self.sigmoid(self.hidden_linear(X))
        out = self.sigmoid(self.output_linear(hidden_output))
        return out
```

Fig 6. DNN Model Implementation

A tabular representation of the used parameter values for the VTEAM model is shown in Table 2 below. Fig. 6 is the I-V plot using the data from Table 2.

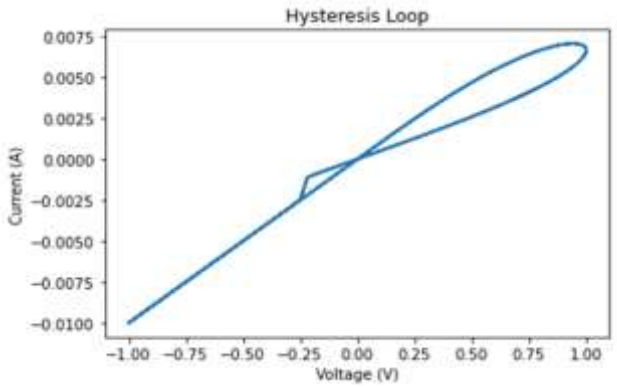

Fig 7. I-V Characteristic using parameters in Table-2

Fig. 5 shows the accuracy of the model trained and inferred on the split dataset. The model is saved for the maximum accuracy obtained at epoch 10000. This model is then transformed with MemTorch by patching the linear layers to their equivalent memristor behavior. Several behavioral memristor models are supported by MemTorch, including the LinearIonDrift, VTEAM, Verilog-A ReRAM, and Stanford PKU models. The VTEAM model is used with the behavioral parameters of the Pt-Hf-Ti memristor.

.

ISBN: 978-988-14049-4-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Table 2. Memristor model characteristics

Characteristics of the Model	Value Taken
Alpha_on	1
Alpha_off	3
V_off	0.5 V
V_on	-0.53V
R_off	2500 ohms
R_on	100 ohms
K_off	$4.03x10^{-8}$
K_on	-80

VI. Conclusion

In conclusion of this study, a review of some recent work has been done in the memristor application was conducted. the memristors are the perfect candidates to implement neural networks on chip, due to their capability of being grouped to represent weights and kernels thereby having a simple and compact structure. This project is invested in exploring the use of memristors in neural networks contributing to the healthcare industry. The findings in the project demonstrate that memristive neural networks hold a great promise to provide effective computation rates at the cost of low power consumption as compared to a conventional CMOS base deep learning system. Using MNNs or memristive neural networks in the wearable healthcare industry holds a promising future to make healthcare devices more efficient and productive. The project points out to the new simulation frameworks for memristors which provide mapping of MNNs from algorithm to hardware level and not constraint to only SPICE model simulations. MemTorch and DNN+ Neurosim are two such frameworks used in this project to compute chip area and power consumption of neural networks using memristors, and research provided through literature in the report proves that MNNs compute less power as compared to CMOS based DL systems.

References

[1] Azghadi, Mostafa Rahimi, Corey Lammie, Jason K. Eshraghian, Melika Payvand, Elisa Donati, Bernabe Linares-Barranco, and Giacomo Indiveri. "Hardware implementation of deep network accelerators towards healthcare and biomedical applications." IEEE Transactions on Biomedical Circuits and Systems 14, no. 6 (2020): 1138-1159.

[2] L.Chua, "Memristor-The missing circuit element," in IEEE Transactions on Circuit Theory, vol. 18, no. 5, pp. 507-519, September,1971

[3] W.M. Holt, "1.1 Moore's Law: A Path Going Forward," Proc. IEEE Int'l SolidState Circuits Conf., 2016, pp. 8–13.

[4] Meng, F.-Y & Duan, Shukai & Wang, L.-D & Hu, Xiaofang & Dong, Zhekang. (2015). An improved WOx memristor model with synapse characteristic analysis. Wuli Xuebao/Acta Physica Sinica. 64. 10.7498/aps.64.148501

[5] Kvatinsky, Shahar & Friedman, E.G. & Kolodny, Avinoam & Uri. (2013). TEAM: ThrEshold adaptive memristor model. Circuits and Systems I: Regular Papers, IEEE Transactions on. 60. 211-221 10.1109/TCSI.2012.2215714.

[6] Jeong, H., and Shi, L. (2018). Memristor devices for neural networks. J. Phys. D Appl. Phys. 52:023003. doi: 10.1088/1361-6463/aae223

[7] D. Kudithipudi, Q. Saleh, C. Merkel, J. Thesing, and B. Wysocki, "Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing," Frontiers in neuroscience, vol. 9, 2015.

[8] Warren S. McCulloch and Walter Pitts," A logical calculus of the ideas immanent in nervous activity", Neurocomputing: foundations of research, James A. Anderson and Edward Rosenfeld (Eds.). MIT Press, Cambridge, MA, USA 15-27, 1988.

[9] Leibin Ni, Hantao Huang, Zichuan Liu, Rajiv V. Joshi, and Hao Yu. 2017, "Distributed In-Memory Computing on Binary RRAM Crossbar",

[10] J. Emerg. Technol. Comput. Syst., Vol. 13, 3, Article 36, March 2017, 18 pages. DOI: https://doi.org/10.1145/ 2996192

[11] L. O. Chua, BMemristorVMissing circuit element, [IEEE Trans. Circuit Theory, vol. 18, no. 5, pp. 507–519, Sep. 1971

[12] Choi, S., Tan, S. H., Li, Z., Kim, Y., Choi, C., Chen, P. Y., et al. (2018). SiGe epitaxial memory for neuromorphic computing with reproducible high performance based on engineered dislocations. Nat. Mater. 17, 335–340

[13] Zhang, Y., Wang, Z., Zhu, J., Yang, Y., Rao, M., Song, W., et al. (2020). Braininspired computing with memristors: challenges in devices, circuits, and systems. Appl. Phys. Rev. 7:011308.

[14] Seok, J. Y., Song, S. J., Yoon, J. H., Yoon, K. J., Park, T. H., Kwon, D. E., et al. (2014). A review of three-dimensional resistive switching cross-bar array memories from the integration and materials property points of view. Adv. Funct. Mater. 24, 5316–5339.

[15] Kozhevnikov, D. D., and Krasilich, N. V. (2016). Memristor-based hardware neural networks modelling review and framework concept. Proc. Inst. Syst. Prog. RAS 28, 243–258.

[16] Ma, D., Wang, G., Han, C., Shen, Y., and Liang, Y. (2018). A memristive neural network model with associative memory for modeling affections. IEEE Access. 6, 61614–61622

[17] L. Gao, P.-Y. Chen, S. Yu, IEEE Electron Device Lett. 2016, 37, 870.

[18] X. Peng, S. Huang, Y. Luo, X. Sun, S. Yu, DNN+NeuroSim: An End to-End Benchmarking Framework for Compute-in-Memory Accelerators with Versatile Device Technologies, in: IEEE International Electron Devices Meeting, 2019.

[18] L. Xia, B. Li, T. Tang, P. Gu, P. Chen, S. Yu, Y. Cao, Y. Wang, Y. Xie, H. Yang, MNSIM: Simulation Platform for Memristor-Based Neuromorphic Computing System, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems 37 (2018) 1009–1022

[19] M. Lin, H. Cheng, W. Lin, T. Yang, I. Tseng, C. Yang, H. Hu, H. Chang, H. Li, M. Chang, DL-RSIM: A Simulation Framework to Enable Reliable ReRAM-based Accelerators for Deep Learning, in: Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Diego, CA, 2018, pp. 1–8.

[20] C. Lammie and M. R. Azghadi, "MemTorch: A Simulation Framework for Deep Memristive Cross-Bar Architectures," *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2020, pp. 1-5, doi: 10.1109/ISCAS45731.2020.9180810.

[21] Y. Zhang, Z. Jia, H. Du, R. Xue, Z. Shen and Z. Shao, "A Practical Highly Paralleled ReRAM-Based DNN Accelerator by Reusing Weight Pattern Repetitions," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 4, pp. 922-935, April 2022, doi: 10.1109/TCAD.2021.3071116

[22] M. Imani, M. Samragh, Y. Kim, S. Gupta, F. Koushanfar, T. Rosing, RAPIDNN: In-Memory Deep Neural Network Acceleration Framework, CoRR abs/1806.05794 (2018)

[23] A. Ankit, I. E. Hajj, S. R. Chalamalasetti, G. Ndu, M. Foltin, R. S. Williams, P. Faraboschi, W. Hwu, J. P. Strachan, K. Roy, D. S. Milojicic, PUMA: A Programmable Ultra-efficient Memristor-based Accelerator for Machine Learning Inference, CoRR abs/1901.10351 (2019).

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in PyTorch, in: NIPS Autodiff Workshop, 2017

[25] S. Kvatinsky, M. Ramadan, E. G. Friedman, A. Kolodny, VTEAM: A General Model for Voltage-Controlled Memristors, IEEE Transactions on Circuits and Systems II: Express Briefs 62 (2015) 786–790