On-Chip Optimization and Deep Reinforcement Learning in Memristor Based Computing

Shahanur Alam, Chris Yakopcic, and Tarek M. Taha
Dept. of Electrical and Computer Engineering, University of Dayton, OH, USA
{alamm8, cyakopcic1, tarek.taha}@udayton.edu

ABSTRACT

Reinforcement learning (RL) has shown its viability to learn when an agent interacts continually with the environment to optimize a policy. This work presents a memristor-based deep reinforcement learning (Mem-DRL) system for on-chip training, where the learning process takes place in a dynamic cartpole environment. Memristor device variability is taken into account to make the study more realistic. The proposed system utilized an analog ReLu module to reduce analog to digital converter usage. The analog Mem-DRL system consumed 191 times less energy than an optimized digital FP16 computing system. Our Mem-DRL system reduced the ADC usages by 40%, which led to reduced the overall system energy by 42%. Mem-DRL is 2.4 times faster than the FP16 system and performs 9.27 GOPS during DRL training. The system exhibited an energy efficiency of 23.8 TOPS/W.

CCS CONCEPTS

• Hardware • Emerging Technologies • Analysis and design of emerging devices and systems • Emerging Architecture

KEYWORDS

Reinforcement learning, Memristor, Online learning, in-memory computing, Analog computing

1 Introduction

Reinforcement Learning (RL) has attracted significant attention for training autonomous systems, as it enables the system to navigate an unknown dynamic environment based on experience. Unlike supervised and unsupervised learning, RL is motivated by cognitive neuroscience and offers a decision-making process that learns from the environment. Supervised or unsupervised systems provide a static solution, but an RL system can continuously evolve and has the potential to adapt to the environment. RL systems become more powerful when combined with a deep neural network and are referred to as Deep Reinforcement Learning (DRL) systems. For instance, the AlphaGo DRL system [1] beat human level capability in the game of Go.

It is common practice to train and run neural network-based systems on Graphics Processing Units (GPU) or Central Processing Units (CPU), which tend to be very energy hungry. For example, the first generation of AlphaGo trained using 280 GPUs, which consumed a peak power of 0.5 MW [2]. Edge DRL applications, such as autonomous drones, prosthetics joints, and autonomous robotic navigation, are generally battery powered and so need much more energy austere training on the device.

Several application-specific integrated circuits (ASICs) have been developed recently to run deep learning algorithms efficiently. They however, still require a high volume of memory and suffer from data latency for accessing off-chip memory modules. Neuromorphic and Computing-in-Memory (CIM) are emerging paradigms that dramatically reduce data transfer bottlenecks for these applications. CIM studies are advancing in industry and academia, with these architectures being investigated for increasing widespread internet-connected IoT and edge devices. Most of the CIM research uses static RAM (SRAM) based memory cells to store quantized binary weights to perform AI inference on edge devices [3]. The majority of CIM studies and products are for inference only, with only a handful of academic studies looking at training.

Training generally requires higher weight precision than for inference. An advantage of SRAM based CIM systems is that they can store quantized digital weight representations [3] that have sufficient precision for online training. However, state-of-the-art memristor devices can now be programmed up to thousands of states [15] and thus are also well suited for on-chip training systems.

The majority of memristor based CIM studies for on-chip training look at classification applications. Only a few works demonstrate RL or Q-learning on memristor-based in-memory spiking [5] and analog [2] domains. However, these works relied more on digital computing components and utilized high bit-width data conversion units, leading to more expensive on-chip training systems. Moreover, prior work on memristor implementations [2] used 16-bit ADC for output quantization and did not investigate the model performance to establish the post-learning success of RL in memristor neuromorphic systems.

This work proposes an extremely low power DRL system with online learning capabilities. The system utilizes emerging memristor devices for developing CIM neuromorphic processors for on-chip training. The memristor crossbar circuit is capable of computing multiplication and addition simultaneously in a highly parallel fashion to perform the dot-product of artificial neural networks. This work used a transposable single column memristive circuit with complementary inputs for accommodating negative parts of

the weight. Analog-to-digital converters (ADCs), digital-to-analog converters (DACs), and on-chip memory units are also needed to perform on-chip training in the analog domain. We developed a custom python-numpy platform to determine the training accuracy in such memristor systems. We compared the performance of the memristor system with a highly energy efficient digital computing system that would be computed in 16-bit floating point (FP16) precision. We assumed a 40 nm process technology was used for both memristor and digital systems.

The contributions of this work are to implement memristor DRL (Mem-DRL) on-chip training systems that have incorporated the following circuit-level implementations:

1. This is the first study to examine the use of analog ReLu circuits for DRL memristor circuits. To ensure accurate evaluation, we implemented the analog ReLu circuit in SPICE and compared the linearity with traditional ReLu. Analog ReLu reduced ADC usage by about 40%. This reduced overall system energy by about 42%.

2. We have shown that low precision can be used for analog training. We used 4-bit ADCs and DACs for on-chip analog training operations where other studies used 16-bit ADCs [2]. To evaluate the training accuracy impact of this, we developed a Mem-DRL on-chip training simulator on the Python-Numpy platform that utilized state-of-the-art device parameters.

The combined effect of these two contributions is a major reduction in the energy consumption and increase in speed of the MEM-DRL system. Comparing our memristor systems with an optimized digital system (FP16), shows about 192 times lower energy while computing 2.4 times faster. Additionally, our Mem-DRL system consumed several orders less energy than the system studied in [2].

We show the proposed Mem-DRL system uses online training to learn in an unknown dynamical environment. This task has many potential applications, such as robotics and healthcare (programming a prosthetic limb [6]), or industry [7] (Unmanned Aerial Vehicle (UAV) flight training [8] and mining operations). For instance, the prosthetic limb application requires the RL chip to be fast (to keep up with real time use), produce low heat (as the system would be attached to the human body), and have low power (to keep battery weight and size low). This motivates the need for a fast, low power memristor DRL system.

The rest of the article is organized as follows: Section 2 describes related works. Section 3 describes in-memory RL systems, and section 4 presents the environment setup for the DRL system. Section 5 describes on-chip training, and section 6 presents the experimental setup for the memristor-based system. Section 7 presents and describes all the results on Mem-DRL, and section 8 describes energy and timing analysis. At the end, section 9 presents a brief conclusion on this article.

2 Related Work

Hardware implementations of CIM systems can be found in the literature [3]. Many research groups and industries are developing and implementing SRAM-based CIMs mainly for edge inference with quantized binary weights stored in SRAM cells [3, 4]. A suitable in-memory learning system is still an open challenge to research communities. Emerging memristor devices are very suitable for developing in-memory computing systems for on-device training and inference. Memristors are well-known non-volatile devices that have been examined for implementing CIM systems [3,4].

The hardware implementation of ANN-RL has not been investigated as much compared to software implementations for many application domains. There are only a few works that have presented reinforcement learning in hardware. Field Programmable Gate Array (FPGA) based deep reinforcement learning has been presented in [9]. The FPGA-based system has to frequently access memory for data, thus causing latency and area overhead. TIME is a memristor-based training in-memory architecture that proposed a CIM reinforcement learning framework [10].

A memristor spiking neural network (SNN) model was proposed for RL in acrobat systems [11,12]. The Remote Supervised Method (ReSuMe) combines SNNs with the basic RL algorithm SARSA [11]. The STDP learning rule is implemented for the SNN training [12]. The RL is implemented in a 1T1R memristor-CMOS hybrid system in [2]. This is an in-memory training system for a classic frictionless and noiseless ideal Cartpole system. The training utilizes off-chip pretraining to accelerate learning, but the memristor-based on-chip training mechanism is hard to determine, given the information provided.

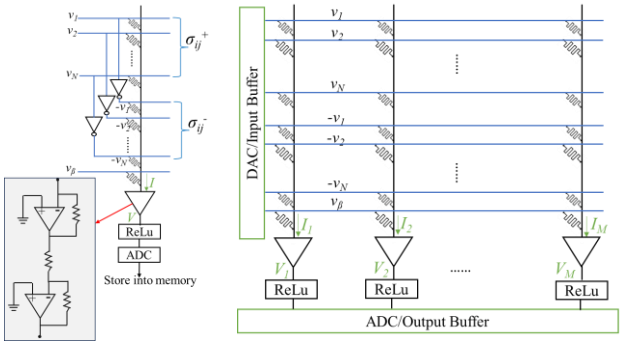
Alternatively, our proposed work complements the previous papers in this area, as we present a complete method for in-situ learning in a memristor crossbar-based RL circuit. We have presented a Mem-DRL hardware model for on-chip RL training and inference with a memristor-based Multi-Layered Perceptron (MLP) model. Error backpropagation method is programmed, and the same crossbar circuit was used for backpropagation. Memristor crossbars were updated with pulse update via the write circuit. As a result, we successfully learned Cartpole-v0, as our system produced a score above 195 over 100 consecutive trials. This is the recommended metric for success for this problem, as described in [13,14].

We show that memristor-based on-chip and in-memory computing can be performed even after adding in device variability. This work looks at a broad application of reinforcement learning as a proof of concept for applications such as autonomous UAV training and prosthetics applications.

3 In-Memory RL in Memristor

Memristors are resistive memory devices whose resistive state can be programmed and which retain this resistance level when powered off. They are often used in crossbar

circuits to perform neural network computational primitives of Matrix-Vector-Multiplication (MVM) in the analog domain. An artificial neural network model can be mapped onto multiple inter-connected crossbar circuits. The crossbar maps each neural network layer for computing MVM operation in one shot. A transposable crossbar circuit is implemented for performing the training operations. Figure 2a is the crossbar representation of a single neuron with complementary inputs by connecting inverter circuits to the original inputs. The complementary input strategy also helps to reduce the input buffer memory and DAC usage. The neuron circuit has negative and positive weight representations with σ_{ij}^{+} and σ_{ij}^{-}, and the actual MVM output is the algebraic sum of $v_i\sigma_{ij}^{+}$ and $v_i\sigma_{ij}^{-}$, where v_i is the element of the input feature map. Eq. (1) presents output voltage representations of j^{th} neuron in a crossbar circuit (Figure (1a,1b). The forward propagation of a transposable crossbar system is presented in Figure 1b. The same circuit is used during error backpropagation, but the inputs are inserted from the transposed direction. The crossbar circuit is orchestrated with digital-to-analog converters (DAC), ReLu activation, analog-to-digital converters (ADC), weight update circuits, and buffer storage for storing quantized outputs and rewards in each time step.



(a) (b)

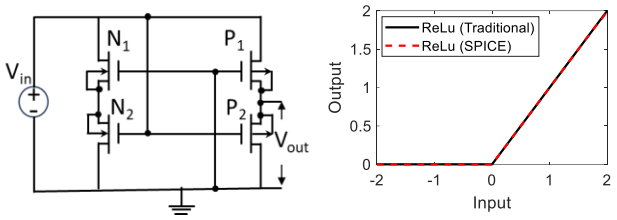
Figure 1: Memristor crossbar circuits, (a) a single neuron, (b) a neural network layer with N inputs and M outputs. The changes of σ_{ij}^{+} and σ_{ij}^{-} are limited between σ_{min} and σ_{max}.

Op-amp circuits are used as summing amplifiers for carrying out the resultant dot-product of the neural system and give a corresponding voltage output. The crossbar circuit outputs can be represented in Eq. (1), which is analogous to the computing primitive of conventional neural networks, as shown in Eq.(2), where x_i, w_{ij}, and b represent respectively inputs weight matrix and bias of a conventional neural network layer.

$$V_j = R[\sum_{i=1}^{N}(v_i\sigma_{ij}^{+} - v_i\sigma_{ij}^{-}) + v_\beta\sigma_\beta] \quad (1)$$

$$Y_j = \sum_{i=1}^{N} x_i.w_{ij} + b \quad (2)$$

An analog ReLu circuit [21] (see Figure 2a) is implemented in SPICE and analyzed with a DC sweep within the the range of -2V to 2V to show the linearity of the circuit to perform ReLu activation. The ReLu circuit works as a half-wave rectifier. For $V_{in}>0$, P_1 and N_2 transistors are turned on and give a linear output, and for $V_{in}\leq 0$, the output remains at ground level through P_2. The linearity of the ReLu circuit follows the traditional ReLu activation with less than 0.01% error margin. The Mem-DRL simulation is set to 0V to 2V for activation output to mimic the actual ReLu output.



(a) (b)

Figure 2: Analog ReLu implementation, (a) ReLu activation circuit, (b) ReLu linearity compared to traditional ReLu.

4 Environment Setup of Mem-DRL

The conceptual schematic of the Mem-DRL with MLP neural network is presented in Figure 3. The MLP neural network learns the policy function for a particular environment state (s). The reward prediction for action is based on the forward passes of the perceptron network and historical observations repeatedly replayed from the experience to optimize the network parameter θ to make the best decision in the unknown dynamic environment. The policy $\pi_\theta(s,a)$ is a Markov decision process that dictates the action which is taken by the agent regarding the state and environment by looking one step ahead to the next state.

Table 1: Parameters for Basic Cartpole system

Parameter	***Magnitude***
Mass of the Cart	1 kg
Mass of the Pole	0.1 kg
Total Mass	1.1 kg
Length of the Pole	1 m
Force	10 N
Cart Friction Coefficient (μ_c)	5×10^{-4}
Pole Friction Coefficient (μ_p)	2×10^{-6}
Interval Between State Update	0.02 s
Reward in Each Time Step	1
Network Learning Rate	0.001
Discount Parameter (γ)	0.997
Optimum Average Score	>195

This work adopts a cartpole environment to examine the Mem-DRL in a memristor neuromorphic CIM. The kinematic relations of a cartpole system are given by A. Barto et al. [16]. A cartoon model of the cartpole is presented in Figure 3(a) with the conventional parameters, and Figure 3(b) presents the Mem-DRL training model. The cartpole environment generates a random four-dimensional Markov state vector s(x, $dx/dt, \theta$, $d\theta/dt$) as input in each time step, x and θ represent position and angle of the pole, respectively. The inputs are applied to the Mem-DRL Q-learning network. In Q-learning, the agent interacts with the environment through a sequence of experience replay, state, action, and reward. The process is schematically presented in Figure 3(b). Table 1 presents the basic environmental setup of the cartpole system.

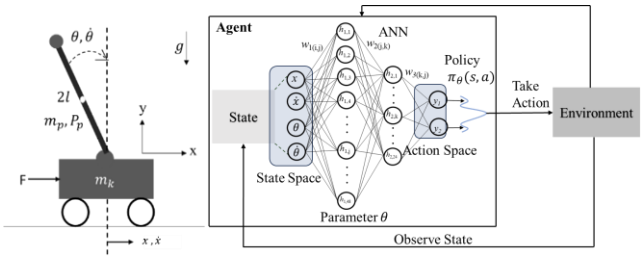


Figure 3: (a) A standard cartpole model with usual parameters, (b) DRL learning setup with state-action representation.

5 Mem-DRL On-Chip Training

In this experiment, the cartpole agent is physically implemented with a memristor based deep Q-network. Q-learning is performed by the agent, which randomly samples from a fixed size pool of transitions (s_t, a_t r_t, s_{t+1}) at each time-step, where s_t, a_t r_t, *and* s_{t+1} represent the state, action, reward, and next state, respectively. The stored transitions are defined as experience and used to train the agent to make future decisions by experience replay.

The physical Mem-DRL training process is presented in Figure 4. The four-dimensional state vector is applied to the memristor crossbar array with complement inputs, as shown in Figure 2. The op-amp circuit in the neuron accumulates MVM results and produces a voltage output. The ReLu removes all negative voltages, ADCs quantize output voltages, and the quantized outputs are stored in the output buffer for the training operation.

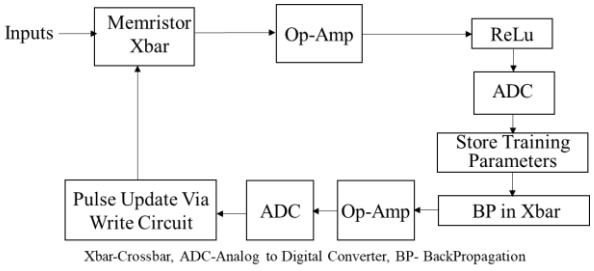


Figure 4: Mem-DRL training procedure.

The memristor-based Q-network works as the Q-function approximator and approximates the Q-function for the current and future states. The results of the forward pass are used in the Bellman equation [2] to compute the loss function. The computed error is applied from the transposed direction of the crossbar circuit and computes the error gradient. The error gradient is quantized and generates a pulse to update the memristor conductance, which is applied to the crossbar through a weight-update circuit along with the previously stored layer input.

6 Experimental Setup

In this study, we assumed a memristor and digital system for evaluation. All the hardware parameters were estimated based on a 40 nm process technology. The memristor system was specified in detail (see below) while the digital system is described at the end of this section.

The crossbar sizes for MLP circuits are (4×2+1)×48, (48×2+1)×24, and (24×2+1) ×2 for 4□48□24□2 fully connected MLP network. The MLP network needs a total of 2858 memristor devices in the crossbar circuits. The ADC and DAC bit-widths are set to 4-bits for both forward and backward propagation. The minimum and maximum conductive states of the memristor devices are considered 0.7 μS and 210μS with a ratio of 300. The weight update process strictly bound the conductive state within this range. The training system assumes there are *M* ADCs for the faster training process. The ADCs are connected to the neuron after ReLu activation, reducing the ADC access by about 40% as negative voltages become zero, thus reducing the timing and energy consumption, as shown in Figure 5.

The use of ADCs in the memristor circuits leads to a quantization of the op-amp outputs and reduces the training accuracy of the memristor system compared to a digital training system. It is essential to capture this effect to ensure accurate training modeling in memristor circuits. Thus, we developed a python-numpy based deep learning training software that modeled the training of the MLP network in our memristor crossbar based training circuits. This software is flexible enough to model other types of networks and ADC/DAC bit widths.

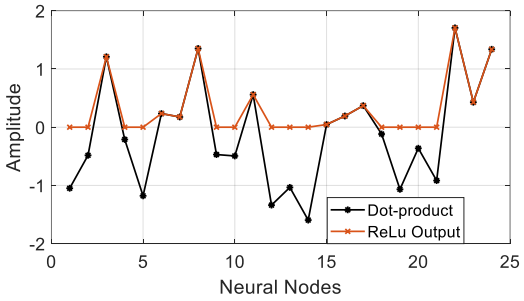


Figure 5: Application of ReLu circuit in the hidden layer 2.

We considered only the memory or computation units for energy and timing estimation for the digital system. We ignored all other energies, including control. Thus, our digital system energy would be the equivalent of a highly optimized digital system. All digital computations were assumed to be done in FP16 for energy and timing considerations. However, the actual training is performed in a general-purpose x86 computer using FP64.

The computing speed of the FP16 system was estimated using parameters from L. Li et al. [17]. The memory energy consumption was estimated with a 40 nm ultra-low leakage memory SRAM memory design by J. Wang et al. [18], and the memory area was estimated with Hewlett Packard's CACTI-P memory estimation software [19]. The ADCs are often one of the most energy-hungry pieces of hardware in analog processors. The ADC energy consumption and area were scaled and estimated based on the results from S. Yu et al. [20].

7 Results and Discussion

The Mem-DRL system is examined with the cartpole-v0 agent for on-chip training and testing. The results of the DRL experiment using a purely digital approach are displayed in Figure 6 compared to the proposed Mem-DRL design, using the same network and hyperparameters. Successful training requires an average score greater than 195 over 100 consecutive trials, and the maximum possible score is 200 for a single trial [13]. Figure 6 presents the raw scores, and Figure 7 presents the moving average of the last 100 trials. The Mem-DRL system took 172 episodes to reach the cut-off reward, whereas the digital system spent 161 episodes.

The injected noise makes the system more realistic and adds to the variability of the memristor devices. We have introduced noise as the randomly generated signal multiplied by a certain percent of the minimum resistance level of the memristor devices. With increasing noise, learning of Mem-DRL system becomes challenging and increases the play time to reach the required score for a successful training operation. The Mem-DRL took 179 and 197 trials to achieve 195 average rewards when applied at a 2% and 4% noise level. For 6% noise, the training does not reach the required score level for successful training.

After solving the problem, the trained models were evaluated to check the performance of both digital and memristor-based Mem-DRL systems. Figure 8 presents the evaluation of the trained models. The systems played 500 test trials. The digital and ideal Mem-DRL system successfully played all trials and scored 100% accuracy. However, the Mem-DRL system with 4% noise failed to score in 16 trials where 195 is considered a passing score. Thus, the accuracy is about 97%.

8 Energy and Timing Analysis

Energy consumption and performance analysis are crucial for measuring the robustness of any hardware. We estimated the energy, power, and processor performance using detailed system evaluations. In the analog training processor design, the data conversion and memory modules are the most energy consuming hardware components. Mem-DRL requires about 1.66 KB of on-chip memory for training the cartpole agent. This is needed to store intermediate training parameters generated in the forward pass and to be consumed during the backward propagation. However, this work did not consider the energy consumption in the replay memory. Table II presents the timing, energy consumption, and performance of the cartpole Mem-DRL and digital systems. The Mem-DRL system is experimented with ReLu in an analog circuit. If activation is computed in a digital system then the energy consumption per time step is 0.147 nJ, which is reduced in the Mem-DRL system to 0.062 nJ. Thus, analog ReLu reduced the ADC usage by about 40% and the energy consumption by 42%.

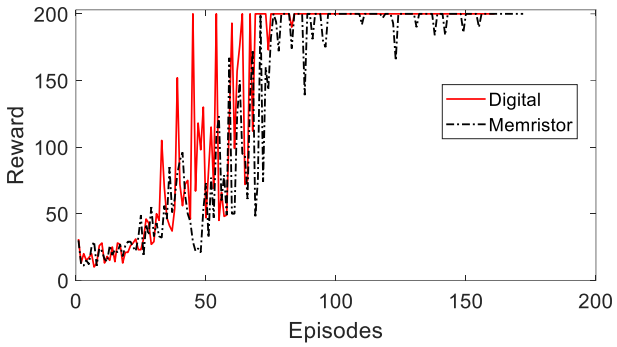


Figure 6: Reward vs. play episode during training.

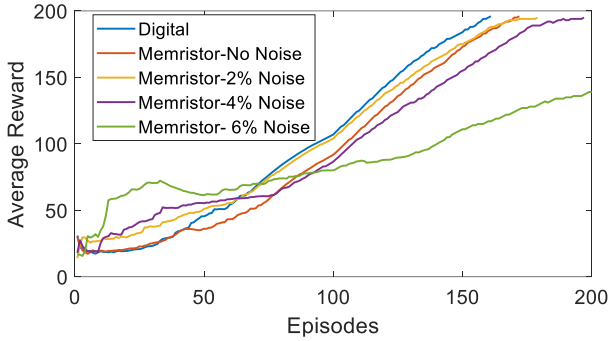


Figure 7: Average reward vs. episode to determine the success of cartpole training while injecting randomly distributed noise scaled with the percent of minimum memristor conductance.

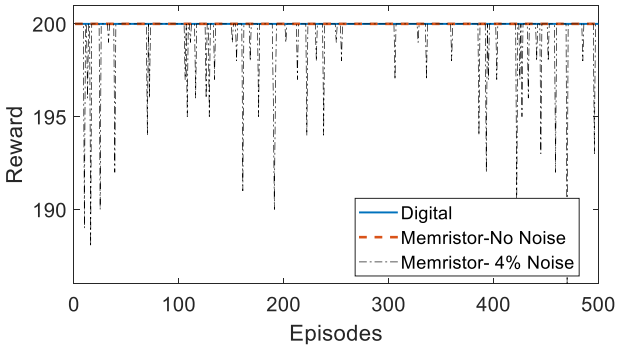


Figure 8: Evaluation of a trained model for digital and MEM-DRL systems.

At the end of the training process, Mem-DRL consumed 1.42 µJ, and the FP16 system consumed 271 µJ. The Mem-DRL system consumed about 191 times less energy than the FP16 system. Figure 9 presents the energy consumption of the cartpole-v0 agent to complete the successful training, where it achieved 195 average rewards over 100 successive trials. The Mem-DRL system performs 9.27 GOPS, which makes the system about 2.4 times faster than the FP16 system. The chip area of Mem-DRL is 1.19× smaller than the FP16 system. The chip area of the digital system is mainly dominated by memory, and the FP16 system requires 8.85KB of memory for cartpole DRL training, which occupies only 0.0145 mm^2. The

chip area of the analog system is dominated by the ADC, which occupies about 80% of the chip area. Finally, the Mem-DRL system exhibited 23.8 TOPS/W, whereas the FP16 system showed 0.123 TOPS/W in DRL training operation.

Table 2: Energy, Time, Performance and Power

Parameters	FP16	Mem-DRL
Time (µs)/step	0.38	0.158
Energy (nJ)/step	11.9	0.062
Power (mW)	31	0.389
Performance (GOPS)	3.85	9.27
Energy Efficiency (TOPS/W)	0.123	23.8
Chip Area (mm^2)	0.0078	0.014
Max Training E (µJ)/Trial	2.38	0.0123

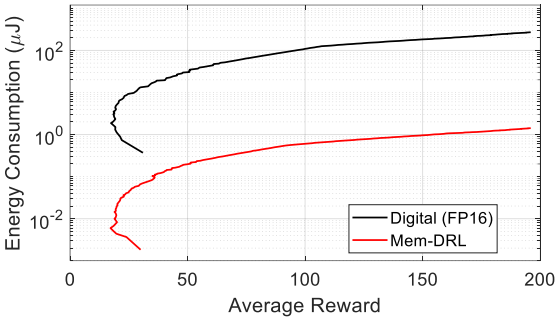


Figure 9: Energy consumption for successful training of cartpole in Mem-DRL and FP16 digital systems.

9 Conclusion

We have developed a low power memristor-based analog computing processor, Mem-DRL, for reinforcement learning applications. We also compared the Mem-DRL system with a digital FP16 system and showed that it achieves the same accuracy level with significantly lower energy costs. The memristor system consumed 191 times less energy than the FP16 system while computed about 2.4 times faster. However, the Mem-DRL has a 1.91 times bigger chip size than the FP16 system. The analog ReLu significantly reduced the memristor-based analog computing system's energy consumption, a major contribution of this work. This activation technique potentially can be implemented in many other analog computing system for low power training operations. Finally, the Mem-DRL shows 80 times more power efficiency than the FP16 system. In future work, the memristor-based ANN-RL may be utilized to develop in-memory and in-situ training systems for autonomous drones, power-constrained navigation robots, and prosthetics.

REFERENCES

[1] Silver, D., Schrittwieser, J., Simonyan, K. *et al.* 2017. Mastering the game of Go without human knowledge. *Nature* 550, 354–359. DOI: https://doi.org/10.1038/nature24270.

[2] Wang, Z., Li, C., Song, W. *et al. 2019.* Reinforcement learning with analogue memristor arrays. *Nat Electron* 2, 115–124. DOI: https://doi.org/10.1038/s41928-019-0221-6.

[3] S. Yu, H. Jiang, S. Huang, X. Peng and A. Lu, 2021. "Compute-in-Memory Chips for Deep Learning: Recent Trends and Prospects," in *IEEE Circuits and Systems Magazine*, vol. 21, no. 3, pp. 31-56, third quarter, DOI: 10.1109/MCAS.2021.3092533.

[4] Sebastian, A., Le Gallo, M., Khaddam-Aljameh, R. *et al. 2020.* Memory devices and applications for in-memory computing. *Nat. Nanotechnol.* 15, 529–544. https://doi.org/10.1038/s41565-020-0655-z.

[5] X. . Ji, Y. Zhang, C. Li, T. Wu, and X. Hu. 2019. "Reinforcement learning in memristive spiking neural networks through modulation of resume." In AIP Conference Proceedings, vol. 2073, no. 1, p. 020094. AIP Publishing LLC, DOI: https://doi.org/10.1063/1.5090748.

[6] S. Yu, H. Jiang, S. Huang, X. Peng and A. Lu, 2021. "Compute-in-Memory C. Yu, J. Liu, S. Nemati, and G. Yin. 2021. "Reinforcement learning in healthcare: A survey." ACM Computing Surveys (CSUR) 55, no. 1: 1-36. DOI: https://doi.org/10.1145/3477600.

[7] R. Nian, J. Liu, and B. Huang. 2020. "A review on reinforcement learning: Introduction and applications in industrial process control." Computers & Chemical Engineering 139: 106886. DOI: 10.1016/j.compchemeng.2020.106886.

[8] A. T. Azar, A. Koubaa, N. A. Mohamed, H. A. Ibrahim, Z. F. Ibrahim, M. Kazim, A. Ammar et al. "Drone Deep Reinforcement Learning: A Review." Electronics 10, no. 9 (2021): 999. DOI: 10.3390/electronics10090999.

[9] D. P. Leal, M. Sugaya, H. Amano and T. Ohkawa, "FPGA Acceleration of ROS2-Based Reinforcement Learning Agents," *2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW)*, Naha, Japan, 2020, pp. 106-112, DOI: 10.1109/CANDARW51189.2020.00031.

[10] Ming Cheng *et al.*, "TIME: A training-in-memory architecture for memristor-based deep neural networks," *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, Austin, TX, USA, 2017, pp. 1-6, DOI: 10.1145/3061639.3062326.

[11] C. Shi, J. Lu, Y. Wang, P. Li and M. Tian, 2021. "Exploiting Memristors for Neuromorphic Reinforcement Learning," *2021 IEEE 3rd International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2021, pp. 1-4, DOI: 10.1109/AICAS51828.2021.9458542.

[12] X. Ji, Y. Zhang, C. Li, T. Wu, and X. Hu. 2019. "Reinforcement learning in memristive spiking neural networks through modulation of resume." In AIP Conference Proceedings, vol. 2073, no. 1, p. 020094. AIP Publishing LLC, DOI: https://doi.org/10.1063/1.5090748.

[13] Cartpole, Open AIGym, Accessed on: Aug 29, 2023, Available: CartpoleOpenAIGym:https://gym.openai.com/envs/CartPole-v0/.

[14] S. Kumar, "Balancing a CartPole System with Reinforcement Learning--A Tutorial." arXiv preprint arXiv:2006.04938 (2020).

[15] Rao, M., Tang, H., Wu, J. *et al. 2023.* Thousands of conductance levels in memristors integrated on CMOS. *Nature* 615, 823–829. https://doi.org/10.1038/s41586-023-05759-5.

[16] Barto, Andrew G., Richard S. Sutton, and Charles W. Anderson. "Neuronlike adaptive elements that can solve difficult learning control problems." *IEEE transactions on systems, man, and cybernetics* 5 (1983): 834-846. DOI: 10.1109/TSMC.1983.6313077.

[17] L. Li, S. Zhang and J. Wu, "Design and realization of deep learning coprocessor oriented to image recognition," *2017 IEEE 17th International Conference on Communication Technology (ICCT)*, Chengdu, China, 2017, pp. 1553-1559. DOI: 10.1109/ICCT.2017.8359892.

[18] J. Wang, H. An, Q. Zhang, H. S. Kim, D. Blaauw and D. Sylvester, "A 40-nm Ultra-Low Leakage Voltage-Stacked SRAM for Intelligent IoT Sensors," in *IEEE Solid-State Circuits Letters*, vol. 4, pp. 14-17, 2021. DOI: 10.1109/LSSC.2020.3043461.

[19] S. Li, K. Chen, J. H. Ahn, J. B. Brockman and N. P. Jouppi, "CACTI-P: Architecture-level modeling for SRAM-based structures with advanced leakage reduction techniques," *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, San Jose, CA, USA, 2011, pp. 694-701. DOI: 10.1109/ICCAD.2011.6105405.

[20] S. Yu, X. Sun, X. Peng and S. Huang, "Compute-in-Memory with Emerging Nonvolatile-Memories: Challenges and Prospects," *2020 IEEE Custom Integrated Circuits Conference (CICC)*, Boston, MA, USA, 2020, pp. 1-4, doi: 10.1109/CICC48029.2020.9075887.

[21] Priyanka, P., Nisarga, G.K. and Raghuram, S., 2019. CMOS implementations of rectified linear activation function. In *VLSI Design and Test: 22nd International Symposium, VDAT 2018, Madurai, India, June 28-30, 2018, Revised Selected Papers 22* (pp. 121-129).