A Low-Power Actor-Critic Framework Based on Memristive Spiking Neural Network

Yaozhong Zhang¹,a, Xiaofang Hu¹, 2, *, Yue Zhou², 3, b and Wenbo Song¹, c

¹College of Computer and Information Science, Southwest University, Chongqing 400715, China
²Brain-inspired Computing & Intelligent Control of Chongqing Key Lab, Chongqing 400715, China
³College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China

*Corresponding author e-mail: huxf@swu.edu.com, azhangyaozhong9@126.com, bzhouyuenju@163.com, cwbsongwu@163.com

Abstract. Traditional deep reinforcement learning (DRL) algorithms consume much energy. Energy-efficient spiking neural networks (SNNs) are promising technologies to build a low-power reinforcement learning architecture. In this paper, an actor-critic framework based on memristive SNN is proposed. To convey and process information in SNN, spike encoding and decoding systems are created. Then, an improved learning algorithm based on spike-timing-dependent plasticity (STDP) learning rule is designed to combine actor-critic method with SNN. Moreover, this learning algorithm is also hardware-friendly. Besides, memristive synapse is designed to accelerate this learning algorithm. Finally, a continuous control problem is applied to illustrate the effectiveness of the proposed framework. The results show the proposed framework is prior to traditional methods.

1. Introduction

Reinforcement learning (RL) is one of the most crucial branches of machine learning. It has been widely used in various control problems [1, 2]. The two main methods of reinforcement learning are policy iteration and value iteration. However, the learning efficiency of these two methods is low due to their characteristics. Policy iteration evaluates policies in each of iterations, which leads to slow convergence speed [3]. As for the value iteration, since the data samples are highly correlated, it is likely to result in an unsuccessful learning [4]. However, Actor-Critic method is able to combine the value iteration with the policy iteration and avoids their disadvantages [3].

Traditional approaches for implementing the actor-critic method are using deep neural networks (DNNs). However, DNN consumes much energy. For instance, AlphaGo Fan and AlphaGo Lee were distributed over 176 GPUs and 48 TPUs, respectively [5]. As a result, traditional deep reinforcement learning is not suitable for large-scale hardware implementation.

The third-generation neural network: spiking neural networks (SNNs) [6] are promising technologies to build a low power actor-critic architecture. SNN is more energy-efficient compared with DNN because of its event-driven nature [7]. Besides, the basic learning rule of SNN is similar to the learning
mechanism of mammals [8]. This bionic characteristic may provide innovations to the artificial intelligence research.

Spike-timing-dependent plasticity (STDP) is the basic learning rule of SNN. Markram defined the STDP learning rule in 1997[9]. The basic STDP learning rule is unsupervised and mainly applied to some specified problems. To combine STDP with RL, a simple and highly efficient learning algorithms must be designed.

To achieve hardware acceleration for SNN, the novel nanoscale electric element: memristor is introduced in this paper. Experiments showed that memristors can simulate the features of biological synapses. There are various implementations for applying memristors in spiking neural network recently. Hu et al proposed a memristor-based dynamic synapse design to realize STDP learning in single memristor [10]. Zheng et al created a methodology to design STDP learning systems based on a memristor crossbar structure [11].

This paper proposes a novel actor-critic framework based on memristive spiking neural network (MSAC): In the second part, data-spike transformation systems are created to transmit and process information in SNN. On account of implementation difficulties of the existing STDP-based learning algorithms, an improved learning algorithm is designed. This improved learning algorithm is combined with actor-critic method effectively. In the third part, memristive synapse is designed based on the proposed learning algorithm. Finally, the simulation of continuous control verifies that the performance of MSAC is better than the conventional one.

2. Combine Actor-Critic with SNN

2.1. Actor-Critic Method

The target of actor-critic method is to maximize the returns $G_t$, which is the sum of discounted reward:

$$G_t = \sum_{k=1}^{\infty} \gamma^{k-1} r_{t+k}$$

(1)

Where $\gamma \in [0,1]$ is the discount factor. $r_{t+k}$ is the reward that agent receives in time step $t+k$. If $\gamma = 0$, the learning process will only depend on the $t+1$ rewards. $\gamma$ is usually set to 0.9 to balance the $t+1$ rewards and rewards that will be received in future time steps. The equation (1) can get its maximum values by solving the Bellman equation:

$$V(s_t) = E[r_{t+1} + \gamma V(s_{t+1})]$$

(2)

Where $V(s_t)$ denotes the value function in state $s_t$. The actor-critic method uses two structures to solve this equation: actor and critic. They are corresponding to policy iteration and value iteration, respectively. The actor is used to output actions while the critic criticizes actor by TD error. TD error is defined as follows [3]:

$$TD = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$$

(3)

The role of the actor network and the critic network is different. The main goal of the critic network is to minimize the absolute value of the TD error. For the actor neural network, it outputs action $a_t$. Then it will adjust its weights by TD error to maximize the value function $V$. 
2.2. Spiking Neural Network

Similar to the previous artificial neural networks (ANNs), spiking neural networks (SNNs) are composed of neurons and synapses. This paper uses typical leaky integrate-and-fire (LIF) neuron model and the synapses model based on STDP learning rule to construct SNN.

The membrane potential \( u(t) \) of LIF model is described by

\[
C \frac{du(t)}{dt} + \frac{u(t)}{R} = I(t)
\]

(4)

Where \( I \) is the input current and \( C \) is the membrane capacitor of LIF model. \( R \) is the membrane resistance of neuron. Besides, the input current usually comes from presynaptic spike strains in LIF model. Thus, the input current of neuron \( j \) can be expressed as follows:

\[
I_j(t) = \sum_i w_{ij} \sum f(t - t_i^{(f)})
\]

(5)

Where \( w_{ij} \) is the synaptic weight between neuron \( i \) and \( j \). \( t_i^{(f)} \) is the firing time of neuron \( i \) for the \( f \)th spike. \( \delta(t - t_i^{(f)}) \) is the presynaptic spike trains.

As one of the basic learning rules of SNN, STDP learning rule adjusts synaptic weights by the time difference \( \Delta t \) of presynaptic and postsynaptic weights. The STDP learning window function \( \xi(\Delta t) \) is defined as follows:

\[
\xi(\Delta t) = \begin{cases} 
A^+ e^{-\Delta t/\tau_{\text{pre}}} , & \Delta t \geq 0 \\
A^- e^{\Delta t/\tau_{\text{post}}} , & \Delta t < 0 
\end{cases}
\]

(6)

Where \( \Delta t = t_{\text{post}} - t_{\text{pre}} \), \( t_{\text{post}} \) and \( t_{\text{pre}} \) are the firing time of postsynaptic and presynaptic neurons. \( \tau_{\text{post}} \) and \( \tau_{\text{pre}} \) are postsynaptic and presynaptic time constants. Coefficients \( A^+ > 0 \), \( A^- < 0 \) for STDP and \( A^+ < 0 \), \( A^- > 0 \) for anti-STDP. In the training process of STDP, presynaptic neuron fires input spike \( \delta_i(t) \) and postsynaptic neuron is forced to fire target spike \( \delta_t(t) \).

2.3. Spike-Data Transformation and Improved STDP Learning Algorithm

The encoding is the transformation from analog data to spike and the decoding is the inverse process in this paper. In addition, to simplify design of MSAC, we restrict that all neurons in SNN can only fire one spike only when they are activated, which can be realized by setting refractory periods. A time window \( T \) is set to be 10 ms.

Biological research shows that spike coding is related to the timing of first spike [12]. Therefore, we designed an input spike encoding method, which is expressed as follows:

\[
t(z) = \frac{T(z - z_{\text{min}})}{z_{\text{max}} - z_{\text{min}}}
\]

(7)

Where \( z \in [z_{\text{min}}, z_{\text{max}}] \) is the input data, \( t(z) \in [0, T] \) is the timing of the first spike after encoding, \( z \) is state \( s \), continuous action \( a \) or value function \( V \). Then, the input current of the postsynaptic neuron is given based on equation (5):
Furthermore, the decoding method is proposed based on (7):

\[ z = \frac{t(z) + (z_{\text{max}} - z_{\text{min}}) + T \cdot z_{\text{min}}}{T} \]  

(9)

State data \( s \) is the input data and does not need decoding. As a result, \( z \neq s \) in (9).

The main idea for designing an improved STDP learning algorithm is to combine a third signal with basic STDP. In actor-critic method, both actor and critic use the TD error for adjusting weights. As a result, TD error can be used as the third signal. Positive TD error suggests that the target output is better than the current outputs, so synaptic weight should increase and vice versa. So the proposed learning algorithm, which can be called TD-STDP, is shown below:

\[ \Delta w_{ij} = TD \cdot \xi(\Delta t) \]  

(10)

3. Hardware Implementation

3.1. Memristive Synapse

The HP memristor model is shown in Fig.1. \( D \) is both the depth of the titanium dioxide and the full length of the memristor. \( W \) is the width of the doped area which is defined by the charge passes through the memristor. The total memristance is expressed as follows:

\[ R_{\text{mem}} = R_{\text{on}} \cdot \frac{W}{D} + R_{\text{off}} \cdot (1 - \frac{W}{D}) \]  

(11)

![Figure 1. HP memristor model. (a) Physical model; (b) circuit symbol.](image)

Panwar et al proved that \( pt \) based memristor can simulate arbitrary STDP behaviors by analog waveform engineering [13]. There are two steps to simulate the STDP window function \( \xi(\Delta t) \) in the memristor. First, set the voltage of the memristor \( V_{\text{mem}}(t) \) as follows:

\[ V_{\text{mem}}(t) = spk(t) - spk(t + \Delta t) \]  

(12)

Where \( spk(t) \) is the spikes fired by neurons. Then, the change of conductance of the memristor \( \% \Delta G_{\text{mem}} \) is computed as follows:

\[ \xi(\Delta t) = \% \Delta G_{\text{mem}}(x) = \text{sign} \* \left( \frac{G_{\text{mem},f}(x) - G_{\text{mem},i}(x)}{\text{min}(G_{\text{mem},i}(x), G_{\text{mem},f}(x))} \right) \* 100 \]  

(13)
In equation (13), $G_{mem,i}(x)$ is the initial conductance before applying $n^{th}$ STDP cycle, $G_{mem,f}(x)$ is the final conductance after applying $n^{th}$ STDP cycle and $\min(G_{mem,i}(x), G_{mem,f}(x))$ is the minimum of $G_{mem,i}(x)$ and $G_{mem,f}(x)$. We set $\text{sign} = 1$ for anti-STDP and $\text{sign} = -1$ for STDP. Furthermore, the weight updating formula based on equation (10) and equation (13) is shown below:

$$\Delta w_{ij} = TD \times \% \Delta G_{mem}(x)$$

(14)

3.2. Hardware implementation and Full Algorithm

The hardware implementation based on previous parts is given in Fig.2. As shown in the picture, the actor and the critic memristive spiking neural networks are both fully connected two-layer neural networks.

There are two modes for this architecture:

(1) Testing mode

Input state data $s$ is transformed into state spikes by spike encoders. Then state spikes propagate to the spike processor along memristive synapses. The spike processor generates input current $I_j(t)$ and sends $I_j(t)$ to the output neuron. $I_j(t)$ charges the $RC$ circuit which leads to neurons fire. Finally, the spike decoder is selected and the generated output spike is transformed to output data (blue arrow in Fig.2).

(2) Training mode

The operation steps of the input layer are same as that of in the input layer. However, operation steps change in the output layer. The spike encoder is selected and the data (green arrow in Fig.2) are transformed to spike. The spike back propagates to the right side of memristive synapses. The TD-STDp learning process will start when the memristive synapses receive spikes in both sides.

In summary, the detailed algorithm is shown as follows:
4. Experiment

A car is positioned in the mountain valley, as shown in Fig.3. The goal for this car is to drive up to the top of the mountain on the right with least energy. However, the engine of the car is not strong enough to reach the target directly. Therefore, it must drive back and forth to gain momentum. The car will get greater reward if it spends less energy to reach the goal.

To verify the effectiveness of MSAC, we compare its performance with the actor critic algorithm based on traditional artificial neural network (AC). Hyperparameters settings are: For MSAC, $A^+ = 1$, $A^- = -0.5$, $\tau_{pre} = \tau_{post} = 1$ms. For AC, the learning rate of the actor network is 0.01 and the critic is 0.1. $\gamma = 0.9$ for both two algorithms. We execute two algorithms for 300 episodes independently. To show the results clearly, we compute the average value every 5 episodes (one training epoch). The results are shown in Fig.4.

In Fig.4, it can be seen that MSAC can get a higher reward than AC. Besides, the absolute value of the average velocity of AC is much smaller than MSAC. The total average velocity of MSAC agent is -0.476 while AC agent is only 0.005. This result suggests that the car controlled by MSAC masters the skill to use reverse potential energy while AC not.

**Algorithm. Memristive Spiking Actor Critic (MSAC)**

Initialize synaptic weights $w_{ij}$

**For** all $episode = 1$ to $M$ **do**

  **Initialize state** $s$

  **For** $t = 1$ to $T$ **do**

    Input state $s_t$ to actor network, get action $a_t$
    Take $a_t$, get reward $r_{t+1}$, state $s_{t+1}$
    Input state $s_t$ and $s_{t+1}$ to critic network, get state value $V(s_t)$ and $V(s_{t+1})$ respectively
    Compute TD error: $TD = r_{t+1} + \gamma V(s_{t+1}) - V(s_t)$
    Train critic network and actor network by TD-STDP
    $s_t \leftarrow s_{t+1}$

  **End for**

**End for**

![Figure 3. MountainCar Continuous](image_url)
5. Conclusion
A novel actor critic framework based on memristive SNN is proposed in this paper. The simulation results show that the performance of the proposed framework is better than conventional approaches. However, the two-layer SNN cannot perform a non-linearly separable logical operation. In the future, we will research on the hardware architecture for deep spiking neural network.

Acknowledgments
This work was financially supported in part by the National Natural Science Foundation of China under Grants 61601376, in part by Fundamental Science and Advanced Technology Research Foundation of Chongqing under Grants cstc2016jcyjA0547, in part by China Postdoctoral Science Foundation Special Funded under Grant 2018T110937, and in part by Chongqing Postdoctoral Science Foundation Special Funded under Grant Xm2017039. Student's Platform for Innovation and Entrepreneurship Training Program 201810635017.

References
[1] C.-H. Wan, M.C. Hwang, Value-based deep reinforcement learning for adaptive isolated intersection signal control, IET Intell. Transp. Syst. 12 (2018) 1005-1010.
[2] K.S. Hwang, H.J. Chao, Adaptive reinforcement learning system for linearization control, IEEE Trans. Ind. Electron. 47 (2000) 1185-1188.
[3] R. Sutton, A. Barto, Reinforcement Learning: An Introduction. The MIT Press, London, 2005. Chapter 4.
[4] V. Mnih, K. Kavukcuoglu, D. Silver et al, Playing atari with deep reinforcement learning, in Proceedings of 26th Conference and Workshop on Neural Information Processing Systems, 2013.
[5] D. Silver, J Schrittwieser, K. Simonyan et al, Mastering the game of Go without human knowledge. Nature. 550 (2017) 354-358.
[6] W. Maass, Networks of spiking neurons: The third generation of neural network models, Neural Netw. 10 (1997) 1659-1671.
[7] Y. Cao, Y. Chen, and D. Khosla, Spiking deep convolutional neural networks for energy-efficient object recognition, Int. J. Comput. Vis. 113 (2015) 54-66.
[8] R.V. Florian, Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity, Neural Comput. 19 (2007) 1468-1502.
[9] H. Markram, J. Lubke, M. Frotscher et al, Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs, Science. 275 (1997) 213-215.
[10] M. Hu, Y. R. Chen, J. J. Yang et al, A compact memristor-based dynamic synapse for spiking neural networks, IEEE Trans. Comput-Aided Des. Integr. Circuits Syst. 1353 (2017) 1353-1366.
[11] N. Zheng, P. Mazumder, Learning in memristor crossbar-based spiking neural networks through modulation of weight-dependent spike-timing-dependent plasticity, IEEE Trans. Nanotechnol. 17 (2018) 520-532.
[12] T. Gollisch, M. Meister, Rapid neural coding in the retina with relative spike latencies, Science, 319 (2008) 1108-1111.

[13] N. Panwar, B. Rajendran, U. Ganguly, Arbitrary spike time dependent plasticity (STDP) in memristor by analog waveform engineering, IEEE Electron Device Lett. 38 (2017) 740-743.