基于多层忆阻脉冲神经网络的强化学习及应用

张耀中 [1] 胡小方 [2,3] 周 跃 [3,4] 段书凯 [2,3]

摘 要 人工神经网络 (Artificial neural networks, ANNs) 与强化学习算法的结合显著增强了智能体的学习能力和效率. 然而, 这些算法需要消耗大量的计算资源, 且难以硬件实现. 而脉冲神经网络 (Spiking neural networks, SNNs) 使用脉冲信号来传递信息, 具有能量效率高、仿生特性强等特点, 且有利于进一步实现强化学习的硬件加速, 增强嵌入式智能体的自主学习能力. 不过, 目前脉冲神经网络的学习和训练过程较为复杂, 网络设计和实现方面存在较大挑战. 本文通过引入人工突触的理想实现元件 — 忆阻器, 提出了一种硬件友好的基于多层忆阻脉冲神经网络的强化学习算法. 特别地, 设计了用于数据−脉冲转换的脉冲神经元; 通过改进脉冲时间依赖可塑性 (Spiking-timing dependent plasticity, STDP) 规则, 使脉冲神经网络与强化学习算法有机结合, 并设计了对应的忆阻神经突触; 构建了可动态调整的网络结构, 以提高网络的学习效率; 最后, 以 Open AI Gym 中的 CartPole-v0 (倒立摆) 和 MountainCar-v0 (小车爬坡) 为例, 通过实验仿真和对比分析, 验证了方案的有效性和相对于传统强化学习方法的优势.

关键词 强化学习, 脉冲神经网络, 脉冲时间依赖可塑性规则, 忆阻器

引用格式 张耀中, 胡小方, 周跃, 段书凯. 基于多层忆阻脉冲神经网络的强化学习及应用. 自动化学报, 2019, **45**(8): 1536−1547

DOI 10.16383/j.aas.c180685

A Novel Reinforcement Learning Algorithm Based on Multilayer Memristive Spiking Neural Network With Applications

ZHANG Yao-Zhong[1] HU Xiao-Fang[2,3] ZHOU Yue[3,4] DUAN Shu-Kai[2,3]

Abstract The combination of reinforcement learning algorithms with artificial neural networks (ANNs) enhances the learning ability of agents effectively. However, these algorithms consume a large number of computing resources, which are unfavourable for hardware implementation. Bionic spiking neural networks (SNNs) convey information by spikes and possess energy-efficient and hardware-friendly features. It is promising to accelerate reinforcement learning and develop embedded self-learning agents based on SNNs. Nevertheless, SNNs lack efficient learning algorithms and their training processes are really complex. As a result, it is challenging to design and implement SNNs. This paper proposes a hardware-friendly reinforcement learning algorithm based on an SNN by introducing famous artificial synapse element: memristor. Data-spike switching spiking neurons are designed especially. Then, we improve spiking-timing-dependent plasticity (STDP) rule to combine the SNN with reinforcement learning organically and the corresponding memristive synapses are created. Besides, the dynamic adjustable network structure is created to increase learning efficiency. Finally, a series of simulations show the effectiveness and advantages of the proposed scheme over conventional reinforcement learning algorithms in applications of CartPole-v0 and MountainCar-v0 in Open AI Gym environment.

Key words Reinforcement learning, spiking neural network (SNN), spike-timing-dependent plasticity (STDP), memristor

Citation Zhang Yao-Zhong, Hu Xiao-Fang, Zhou Yue, Duan Shu-Kai. A novel reinforcement learning algorithm based on multilayer memristive spiking neural network with applications. *Acta Automatica Sinica*, 2019, **45**(8): 1536−1547

收稿日期 2018-10-22 录用日期 2018-12-26
Manuscript received October 22, 2018; accepted December 26, 2018
国家自然科学基金 (61601376, 61672436), 中央高校基本科研业务费 (XDJK2019C034), 重庆市基础与前沿技术研究专项 (cstc2016jcyjA0547), 中国博士后科学基金 (2018T110937), 重庆市博士后科学基金 (Xm2017039), 国家级大学生创新创业训练计划项目 (201810635017) 资助
Supported by National Natural Science Foundation of China (61601376, 61672436), Fundamental Research Funds for the Central Universities (XDJK2019C034), Fundamental Science and Advanced Technology Research Foundation of Chongqing (cstc2016jcyjA0547), Special Science Foundation of Chinese Postdoctoral Fellow (2018T110937), Special Foundation of Postdoctoral Fellow of Chongqing (Xm2017039), and National Student's Platform for Innovation and Entrepreneurship Training Program (201810635017)
本文责任编委 张敏灵
Recommended by Associate Editor ZHANG Min-Ling
1. 西南大学计算机与信息科学学院 重庆 400715 2. 西南大学人工智能学院 重庆 400715 3. 类脑计算与智能控制重庆市重点实验室 重庆 400715 4. 西南大学电子信息工程学院 重庆 400715
1. School of Computer and Information Science, Southwest University, Chongqing 400715 2. School of Artificial Intelligence, Southwest University, Chongqing 400715 3. Brain-inspired Computing and Intelligent Control of Chongqing Key Laboratory, Chongqing 400715 4. College of Electronic and Information Engineering, Southwest University, Chongqing 400715

强化学习, 是智能体通过与环境交互、试错的过程来学习的行为. 它是一种以环境反馈作为输入的自适应的机器学习方法[1], 目前已广泛应用于控制科学、运筹学等诸多领域[2−3]. 在强化学习过程中, 智能体最初对环境一无所知, 通过与环境交互的方式获取奖赏. 智能体在这个过程中学习策略, 使得最终能在某种评价体系下达到最优目标. Q 学习是一种典型的无需模型的强化学习算法, 智能体根据动作价值即 Q 值函数, 通过对状态−动作序列进行足够多的访问, 学习到最优策略[4]. 通常, 在 Q 学习任务中, Q 值函数由表格的方式实现, 在状态为连续值的情况下, 则通过离散化状态以存储动作价值, 然而传统的表格法有如下缺点: 1) 状态的离散度难以控制. 2) 状态维数较多时会导致维数灾难.

将神经网络作为 Q 值函数拟合器可以有效解决以上问题. 神经网络可以分为三代: 第一代把 McCulloch-Pitts 神经元模型作为计算单元; 第二代为人工神经网络 (Artificial neural network, ANN), 它们的计算单元中带有激活函数; 脉冲神经网络 (Spiking neural network, SNN) 将脉冲神经元作为计算单元, 被称为第三代神经网络[5]. SNN 的学习方式与哺乳动物的学习方式非常类似[6]. 此外, SNN 能量效率高, 有报道证明 SNN 芯片比用现场可编程门阵列 (Field programmable gate array, FPGA) 实现的 ANN 能耗低两个数量级[7]. 因此, 基于 SNN 的强化学习算法更容易进行低功耗−硬件实现.

与 ANN 类似, SNN 的学习算法也分为监督学习算法和非监督学习算法. 非监督学习算法仅仅基于数据的特征, 这类算法对计算能力要求较低, 因为不需要数据集的多次迭代, 脉冲神经网络中典型的非监督学习算法是脉冲时间依赖可塑性 (Spike-timing dependent plasticity, STDP) 学习规则[8]. 而监督学习算法需要带有标签的数据集, 需要多次迭代运算, 主要有远程监督学习算法 (ReSuMe) 等[9].

目前许多训练 SNN 的学习算法都只能用于不含隐含层的网络, 且没有通用的方法[10]. 对于训练多层 SNN, 一种方式是先训练 ANN, 再将其转换为 SNN[11], 这种基于映射的学习方式会导致局部最优, 因为训练在 ANN 上进行, 而不是 SNN[12]. 也有人提出了利用突触延迟的监督学习算法, 并行调整隐含层和输出层权重[13]. 由于本文基于多层 SNN 实现强化学习算法, 因此设计有效的多层 SNN 的训练方法是一个必须要解决的问题.

基于传统半导体器件和集成技术实现的神经网络电路复杂度高、规模小、处理能力有限, 难以真正用于嵌入式智能体. 本文进一步引入新型纳米信息器件忆阻器, 探求强化学习算法的硬件加速新方案. 忆阻器是除电阻、电容、电感以外的第四种基本电路元件, 由 Chua[14] 于 1971 年基于电路完备性理论提出, 其定义忆阻器的电阻值为流经忆阻器的磁通量和电荷的比值 ($M = \mathrm{d}\phi/\mathrm{d}q$). 然而, 由于没有物理实物, 忆阻器一直没有引起太多的关注. 直到 2008 年, 美国惠普 (HP) 实验室制造出了基于二氧化钛的交叉存储阵列, 并声称交叉点处的存储单元即为预言的忆阻器[15], 立即引起了学术界和工业界的深厚兴趣. 之后, 研究者对忆阻器的模型、特性进行了广泛的研究[16−17]. 此外由于忆阻器具有记忆力和类似突触的可变导电性, 使其成为构建硬件神经网络关键部件 — 电子突触的理想器件. 近年来, Jo 等[18] 证明了 CMOS 神经元和忆阻突触构成的神经网络能够实现一些重要的突触行为, 如 STDP. 在此基础上, 研究者提出了多种用忆阻器实现 STDP 的方法, 例如 Panwar 等[19] 实现了对任意 STDP 波形的模拟. Serrano-Gotarredona 等[20] 仅用一个忆阻器实现并完成了对 STDP 的仿真.

本文提出并研究了基于多层 SNN 的强化学习算法, 并利用忆阻器设计了其硬件实现方案, 下文称之为忆阻脉冲强化学习 (Memristive spiking reinforcement learning, MSRL). 首先, 为了实现数据和脉冲之间的转换, 设计了用于数据−脉冲转换的脉冲神经元; 然后, 通过改进基本 STDP 学习规则, 将 SNN 与强化学习算法有效结合, 并设计相应的忆阻突触以期实现硬件加速. 此外, 为了进一步提高网络的学习效率, 构建了可动态调整的网络结构. 最后基于 brian2 框架[21] 完成了对 MSRL 的实验仿真. 结果显示, MSRL 控制的智能体可以以较低的计算资源消耗, 高效地完成强化学习任务.

本文结构如下: 第 1 节介绍了 Q 学习和 SNN 以及忆阻器的背景知识, 第 2 节给出 MSRL 算法的基础, 第 3 节详细地介绍了 MSRL 算法设计. 第 4 节给出仿真结果, 第 5 节总结全文.

1　背景知识

1.1　Q 学习

强化学习的理论基础是马尔科夫决策过程 (Markov decision process, MDP). MDP 可以表示为: $(\boldsymbol{S}, \boldsymbol{A}, P_a(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}), R_a(\boldsymbol{s}_t, \boldsymbol{s}_{t+1}))$, 其中 $\boldsymbol{S}$ 是状态集, $\boldsymbol{A}$ 是动作集, $P_a(\boldsymbol{s}_t, \boldsymbol{s}_{t+1})$ 表示若智能体在时间 t 时处于状态 $\boldsymbol{s}_t$, 采取动作 a 可以在时间 $t+1$ 时转换到 $\boldsymbol{s}_{t+1}$ 的概率; $R_a(\boldsymbol{s}_t, \boldsymbol{s}_{t+1})$ 表示通过动作 a, 状态 $\boldsymbol{s}_t$ 转换到 $\boldsymbol{s}_{t+1}$ 所带来的及时奖赏.

强化学习中的 Q 学习是一种经典的在线学习方

法. 在学习过程中, 智能体在每一个时间步 (step) 内尝试动作, 获得来自环境的奖赏, 从而更新 Q 值和优化行动策略 $\pi(\boldsymbol{s})$ (如图 1). 这个学习过程称为时间差分 (Temporal difference, TD) 学习[22].

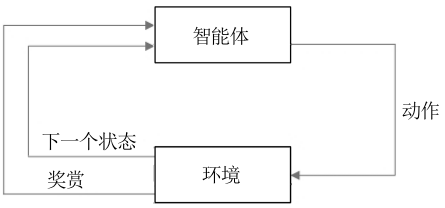


图 1 Q 学习过程

Fig. 1 The process of Q-learning

强化学习的目标是让智能体通过与环境的交互学到最优的行动策略 $\pi^*(\boldsymbol{s})$, 使累积奖赏即回报最大. 回报定义为

$$G_t = \sum_{i=0}^{\infty} \gamma^i r_{t+i} \tag{1}$$

其中, 折扣因子 $\gamma \in [0,1]$, 表示我们对未来奖赏的重视程度. $\gamma = 0$ 时智能体只关注当前奖赏值, $\gamma = 1$ 时未来奖赏与当前奖赏同样重要.

Q 学习算法中的 Q 值是智能体按照行动策略 $\pi(\boldsymbol{s})$ 执行动作后所得回报的期望, 定义为

$$Q_\pi(\boldsymbol{s}_t, a_t) = \mathrm{E}_\pi[G_\pi | \boldsymbol{S} = \boldsymbol{s}_t, A = a_t] \tag{2}$$

智能体通过 Q 值的更新优化行动策略 $\pi(\boldsymbol{s})$, 使其所得回报增大. Q 值更新公式为

$$Q(\boldsymbol{s}_t, a_t) \leftarrow Q(\boldsymbol{s}_t, a_t) + \alpha[r_t + \gamma \max_{a_{t+1}} Q(\boldsymbol{s}_{t+1}, a_{t+1}) - Q(\boldsymbol{s}_t, a_t)] \tag{3}$$

其中, $\max_{a_{t+1}} Q(\boldsymbol{s}_{t+1}, a_{t+1})$ 表示智能体在状态 $\boldsymbol{s}_{t+1}$ 下采取动作 a_{t+1} 后所得到的 Q 值中的最大值, 而 $\gamma \max_{a_{t+1}} Q(\boldsymbol{s}_{t+1}, a_{t+1}) - Q(\boldsymbol{s}_t, a_t)$ 便是所谓的 TD 误差, 用来衡量目标 Q 值 $\gamma \max_{a_{t+1}} Q(\boldsymbol{s}_{t+1}, a_{t+1})$ 和当前 Q 值 $Q(\boldsymbol{s}_t, a_t)$ 之间的差距, 学习率 $\alpha \in [0,1]$ 表示对过往经验的重视程度.

除此之外, 在 Q 学习中选择动作的基本策略也即本文采取的策略是 $\epsilon-$greedy 策略, 该策略也是 Q 学习同其他机器学习所不同之处, 它反映了 Q 学习中智能体探索 (Exploration) 和利用 (Exploitation) 之间的权衡. 前者是指随机尝试动作, 以期获得更高的回报, 即 ϵ; 后者是执行根据历史经验学习到的可获得最大收益的动作, 即 greedy. 智能体以概率 ϵ 随机选择动作, 而以 $1-\epsilon$ 的概率选取最大价值所对应的动作.

基本 Q 学习的算法流程可描述为

算法 1. 基本 Q 学习算法

1 任意初始化动作价值 $Q_\pi(\boldsymbol{s}_0, a_0)$
2 **for** episode $= 1 : M$ **do**
3 初始化状态 $\boldsymbol{s}$, 概率 ϵ
4 **repeat**
5 以概率 ϵ 随机选择动作, 以概率 $1-\epsilon$ 选取最大价值所对应动作
6 执行动作 a_t, 获得奖赏 r_t, 观察到状态 $\boldsymbol{s}_{t+1}$
7 更新 Q 值: $Q(\boldsymbol{s}_t, a_t) \leftarrow Q(\boldsymbol{s}_t, a_t) + \alpha[r_t + \gamma \max_{a_{t+1}} Q(\boldsymbol{s}_{t+1}, a_{t+1}) - Q(\boldsymbol{s}_t, a_t)]$
8 更新状态: $\boldsymbol{s}_t \leftarrow \boldsymbol{s}_{t+1}$
9 **until** $\boldsymbol{s}$ is terminal
10 **end for**

1.2 脉冲神经网络

脉冲神经网络 (Spiking neural network, SNN) 起源于神经科学, 广泛用于构建类脑神经系统模型, 例如用于设计模拟大脑皮层中的信息传递和时间动态可观测过程[23]. 与 ANN 类似, SNN 也是由神经元和突触构成, 本文利用经典的 LIF (Leaky integrate-and-fire) 神经元模型和具有 STDP 学习规则的突触模型来构建 SNN.

在流经离子通道的电流作用下, 脉冲神经元 (Spiking neuron, SN) 的细胞膜将会产生动作电位 $u(t)$[24]. 当动作电位达到阈值后, 神经元将会发放脉冲, 这个过程可以描述为

$$u(t^{(f)}) = u_{\mathrm{th}} \tag{4}$$

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t}|_{t=t^{(f)}} > 0 \tag{5}$$

其中, $t^{(f)}$ 是神经元发放脉冲的时间, u_{th} 是阈值电压.

LIF 模型将神经元抽象为一个 RC 电路 (图 2).

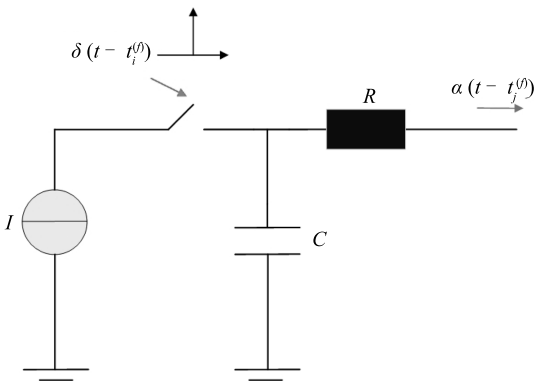


图 2 LIF 模型

Fig. 2 LIF model

图 2 中, $\delta(t-t_i^{(f)})$ 为来自突触前神经元 i 的脉冲信号, 而 $\alpha(t-t_j^{(f)})$ 为突触后神经元 j 的输出脉冲. 神经元收到输入电流后, 由于积分电路的作用, 动作电位会升高, 直到达到激活阈值, 发放脉冲, 这个过程称为积分点火. 在脉冲发放后, 由于漏电流的作用, 神经元的动作电位会立即恢复至静息电位, 这一过程是对真实生物神经元中的离子扩散效应的模拟[25]. LIF 模型的微分方程描述如下

$$I_1(t)=\frac{u(t)}{R_m(t)} \tag{6}$$

$$I(t)-I_1(t)=C_m\frac{\mathrm{d}u(t)}{\mathrm{d}t} \tag{7}$$

其中, C_m 为神经元膜电容, $I(t)$ 为外界输入电流, $I_1(t)$ 为漏电流, $R_m(t)$ 为神经元膜电阻.

在 LIF 模型中, 外部输入电流 $I(t)$ 通常为 $\delta(t-t_i^{(f)})$ 的加权和, 因此, 神经元 j 收到第 i 个神经元的输入电流可以表示为

$$I_j(t)=\sum_i\left\{w_{ij}\sum_f\delta(t-t_i^{(f)})\right\} \tag{8}$$

其中, w_{ij} 为神经元 i 和 j 之间的突触权重; $t_i^{(f)}$ 为突触前神经元 i 发出第 f 个脉冲的时间.

STDP 规则是 SNN 的基本学习规则之一, 具有良好的生物学基础. Hebb 等[26] 于 1949 年提出通过改变神经元相互之间的连接强度来完成神经系统学习过程的假设, 称为 Hebb 规则. Hebb 规则指出, 如果两个神经元同时发放脉冲, 则它们之间的突触权重会增加, 反之会减少. 这一假设描述了生物神经元突触可塑性的基本原理. 随后在海马趾上进行的研究发现了长时增强 (Long-term potentiation, LTP) 效应和长时抑制 (Long-term depression, LTD) 效应: 在一个时间窗口内, 如果突触后神经元发放脉冲晚于突触前神经元发放脉冲, 则会导致 LTP 效应, 而反之则会导致 LTD 效应. 前者称为 “突触前先于突触后” 事件 (“Pre before post” event), 后者称为 “突触后先于突触前” 事件 (“Post before pre” event). LTP 和 LTD 有力地支持了 Hebb 的假设.

LTP 和 LTD 效应是与脉冲发放时间高度相关的, 基于这两种效应和相关实验, Markram[27] 于 1997 年定义了 STDP 规则, 在 STDP 规则中权重的变化量是前后两个神经元激活的时间差的函数, 该函数称为学习窗函数 $\xi(\Delta t)$, STDP 学习窗函数 $\xi(\Delta t)$ 以及权重变化量 Δw_{ij} 如下所示

$$\xi(\Delta t)=\begin{cases}A^+\mathrm{e}^{-\frac{\Delta t}{\tau_{\mathrm{pre}}}}, & \Delta t\geq 0\\ A^-\mathrm{e}^{\frac{\Delta t}{\tau_{\mathrm{post}}}}, & \Delta t<0\end{cases} \tag{9}$$

$$\Delta w_{ij}=w_{ij}\xi(\Delta t) \tag{10}$$

式 (9) 中, $\Delta t=t_{\mathrm{post}}-t_{\mathrm{pre}}$ 为突触后神经元与突触前神经元发放脉冲时间差, 而 τ_{pre}, τ_{post} 分别为突触前后的时间常数, 权重增强的增益 $A^+>0$, 减弱的增益 $A^-<0$. $\Delta t\geq 0$ 对应 LTP 效应而 $\Delta t<0$ 对应 LTD 效应. STDP 学习规则可以看作是 Hebb 规则在时间上的改进版本, 因为它考虑了输入脉冲和输出脉冲调整突触权重时时间上的相关性, 换句话说, STDP 强调了脉冲之间的因果联系.

1.3　忆阻器模型

HP 实验室于 2008 年制造出了能够工作的物理忆阻器, 并提出了 HP 忆阻器模型 (图 3).

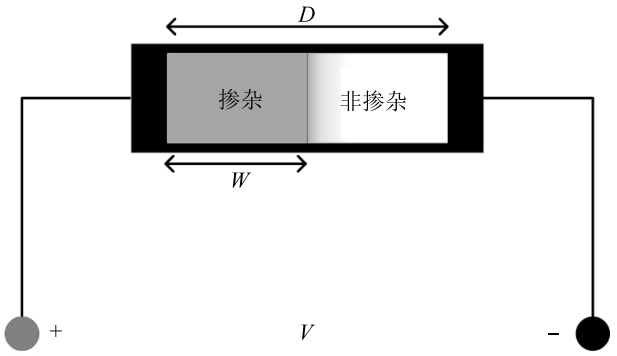


图 3　HP 忆阻器模型示意图

Fig. 3　HP memristor

图 3 中, D 是二氧化钛薄膜的厚度, 亦为忆阻器的全长, W 是掺杂层的宽度, 会在电场作用下改变, 并与流过忆阻器的电荷数有关. 当掺杂宽度 W 增大, 忆阻值减小, 反之忆阻值增大. 忆阻器的总电阻值可表示为

$$R_{\mathrm{mem}}(x)=R_{\mathrm{on}}x+R_{\mathrm{off}}(1-x) \tag{11}$$

$$x=\frac{W}{D}\in(0,1) \tag{12}$$

其中, R_{on} 和 R_{off} 分别为掺杂区和非掺杂区的长度达到全长时的电阻, 也称为极值电阻. 由于在时间 t 时, 掺杂区的宽度取决于通过忆阻器的电荷量, 而电流为电荷的导数, 因此, 内部状态变量 x 的变化可以表示为电流的函数

$$\frac{\mathrm{d}x}{\mathrm{d}t}=\frac{U_D}{D}=\frac{\mu E}{D}=\frac{\mu R_{\mathrm{on}}i(t)}{D^2}f(x) \tag{13}$$

其中, U_D 是掺杂区和非掺杂区之间边界移动的速度, μ 是平均离子漂移率, E 是掺杂区的电场强度, $i(t)$ 为流经忆阻器的电流, $f(x)$ 为窗函数, 已存在多

种多样的函数表达形式, 通常用于模拟离子漂移的非线性, 限制器件边缘特性等. 本文的主要目的并非提出新的忆阻器模型, 而是利用合适的模型实现忆阻突触, 后文详述, 这里不做过多讨论.

2 基于忆阻 SNN 的强化学习

忆阻脉冲神经网络强化学习 (MSRL) 算法的目标在于减小 TD 误差的绝对值, 使回报最大. 训练 SNN 所需样本来自对过往经验的回放, 这些经验存放在记忆池中. 经验回放减少了需要学习的经验数目, 学习效率高于智能体直接与环境交互学习的方式[28]. 由此, 设计 MSRL 算法的首先任务是设计学习效率较高的 SNN 并使之与 Q 学习结合.

2.1 忆阻 SNN

MSRL 算法的设计是基于一个三层的 SNN, 如图 4 所示. 图中省略号表示神经元的数量随着任务的不同而变化. 网络中输入神经元将状态值转换为状态脉冲 $\delta_S(t)$, 其数量等于状态的维数. 输出神经元的输出为 Q 值脉冲 $\delta_Q(t)$, 其数量等于动作数. 这样的结构意味着每个输入神经元对应每个状态维度, 每个输出神经元对应每个动作. 相邻层神经元之间用忆阻器连接, 忆阻器可工作在三种状态: a) 权重不可更改状态; b) 权重调节状态; c) 断开状态.

适当调节隐含层节点数量是有必要的, 如果隐含层节点数过少, 网络的学习能力和信息处理能力不足. 反之, 如果节点过多可能会增加网络结构的复杂性, 减慢运行速度. 具体的隐含层神经元数量对网络性能的影响将在第 4 节讨论.

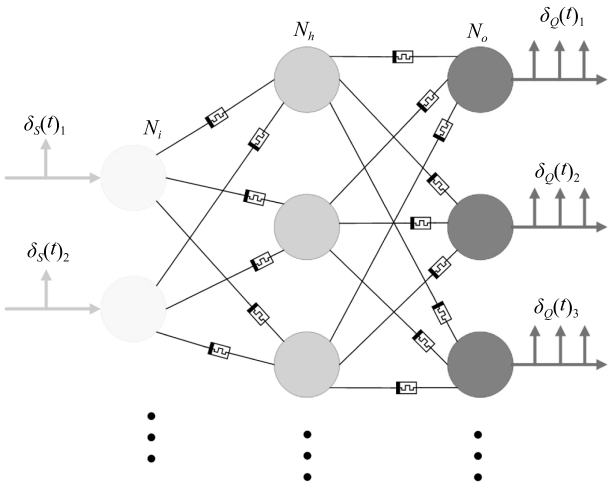


图 4 脉冲神经网络结构

Fig. 4 The structure of SNN

2.2 数据 – 脉冲转换

考虑到脉冲神经元接受、处理和传递的信息是脉冲信号, 因此有必要设计数据与脉冲之间的转换关系. 在本文中, 模拟数据转换为脉冲时间序列的过程为编码, 其逆过程为解码. 一个时间窗口 T 为 10 ms.

1) 输入层神经元

生物学研究表明, 在生物视觉神经网络中, 神经元对信息的编码与首次发放脉冲的时间有关, 发放时间越提前说明输入脉冲与输出脉冲之间的相关性越强[29]. 由此引入一维编码方式[30]: 状态值 $s \in [s_{\min}, s_{\max}]$, 编码后首次发放时间 $t(s) \in [0, T]$, 则编码规则为

$$t(s) = \frac{T(s - s_{\min})}{s_{\max} - s_{\min}} \tag{14}$$

这种编码方式使输入神经元在一个 T 内只发放一个脉冲. 基于式 (14), 并结合式 (8) 得到隐含层输入电流 $I_h(t)$ 为

$$I_h(t) = \sum_i w_{ih} t(s)_i \tag{15}$$

其中, w_{ih} 为输入层与隐含层神经元之间的突触权重. 输入神经元用于将状态值转换为单个的状态脉冲, 没有解码过程.

2) 隐含层神经元

研究以下情形: 两个 LIF 神经元 i, j 由一个突触连接. 突触前神经元 i 为输入神经元而突触后神经元 j 为输出神经元, 它们的初始电压均为 0, 神经元 i 在 t_0 时间电压达到阈值而发放脉冲, 根据式 (8), 脉冲将通过突触转换为输入至神经元 j 的电流, 如果输入电流能使突触后电位达到阈值, 则突触后神经元 j 将发放脉冲. 通过神经元不应期的设置, 在一个时间窗口的时间内, 神经元 j 只会发放一个脉冲, 如图 5 (a) 所示.

对于隐含层神经元, 设置激发态时其只发放一个脉冲, 解码时将其脉冲发放时间 t_h 直接作为输出数据, 从而可得输出层输入电流 $I_o(t)$

$$I_o(t) = \sum_h w_{ho} t_h \tag{16}$$

其中, w_{ho} 是隐含层与输出层之间的突触权重, 编码时则根据发放时间还原脉冲即可.

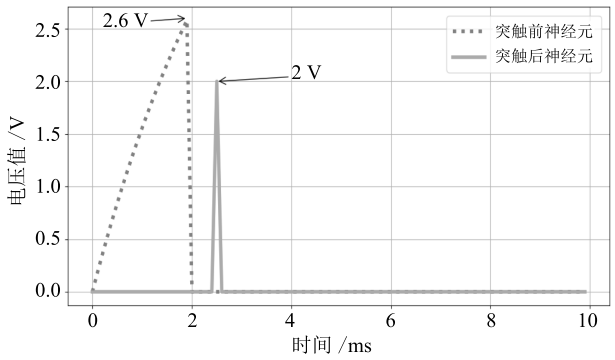
3) 输出层神经元

由于首次发放时间越提前说明输入输出相关性越强, 则可以认为在一个时间窗口内, 输出层中最早发放脉冲的神经元为动作价值最大的动作, 这意味着首次发放时间和动作价值呈反相关关系, 解码时

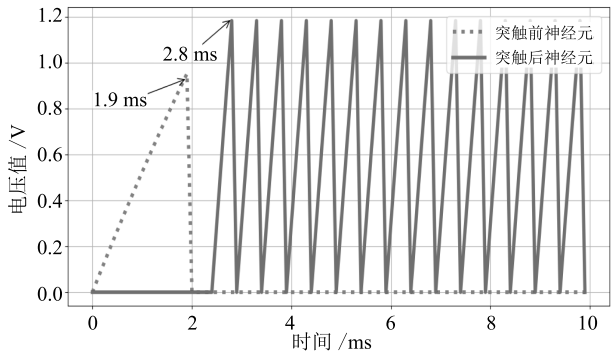
直接将首次发放时间作为输出数据则需要修改 Q 值更新公式. 为了减少算法设计的复杂度, 输出层神经元发放脉冲的形式设置为连续发放. 如图 5 (b) 所示, 进而计算其平均发放率 (Mean firing rate)[24]

$$v = \frac{n_{\rm sp}}{T} \tag{17}$$

其中, $n_{\rm sp}$ 是输出层神经元在一个时间窗口 T 内发放脉冲的数量. 事实上, 平均发放率和首次发放时间是等效的, 一个神经元的平均发放率越高, 由于脉冲时间间隔均等, 则说明它的首次发放时间就越提前[24]. 因而如果设置输出层神经元总是在一个时间窗口内, 连续发放时间间隔相同的脉冲, 那么可以直接将 $n_{\rm sp}$ 作为输出动作价值. 进一步, 近似认为输出脉冲时间将 T 均等分, 所以输出脉冲序列的发放时间为等差数列, 在已知数列项数即脉冲数量 $n_{\rm sp}$ 的情况下可还原脉冲序列.



(a) 输出为单脉冲
(a) Output: one spike



(b) 输出为连续脉冲
(b) Output: continuous spikes

图 5　脉冲神经元响应

Fig. 5　The response of spiking neurons

2.3　改进 STDP 与忆阻突触设计

神经科学领域的主要研究问题之一是对生物学习过程的解释. 例如, STDP 学习规则的提出是基于对单个生物突触的实验, 但对于 STDP 规则如何在脉冲神经网络中实现权重调整并没有统一且详尽的描述[31]. 为了实现 STDP 规则对脉冲神经网络的权重调整, 进而应用于强化学习中, 需要对基本 STDP 规则加以改进. 其思路在于引入第三方信号 (可以是奖赏信号或 TD 误差信号), 作为突触权重的调节信号[31].

以奖赏信号为调节信号的 STDP 规则称作 Reward STDP, 例如文献 [32] 提出如下权重调节规则

$$\Delta w_{ij} = \frac{T_e \xi(\Delta t)}{T_e + t_{\rm re} - t_t} S_{\rm rp} \tag{18}$$

方案中奖赏为一个时间函数 $S_{\rm rp}$, $t_{\rm re}$ 是奖赏出现的时间而 t_t 是智能体执行动作的时间. T_e 是每次迭代持续的时间. Reward STDP 实现了在虚拟环境中对觅食行为这一生物问题的建模. 但是, 这种方案不适用于强化学习任务, 因为在强化学习任务中, 执行动作的事件和奖赏之间可能达到上千步的延迟, 导致学习效率非常低.

以 TD 误差信号作为调节信号的 STDP 规则称作 TD STDP 规则, 为了方便讨论, 将 TD 误差重写

$$TD = r_t + \gamma \max_{a_{t+1}} Q(\boldsymbol{s}_{t+1}, a_{t+1}) - Q(\boldsymbol{s}_t, a_t) \tag{19}$$

利用式 (19), 文献 [33] 提出如下的权重调节方案

$$\frac{{\rm d}w_{ij}(t)}{{\rm d}t} = \eta TD \frac{\rho(stdp_{ij}(t))}{w_{ij}(t)} \tag{20}$$

其中, $\rho(stdp_{ij}(t))$ 为突触前发放脉冲与突触后发放脉冲的概率之差, $\eta \in [0, 1]$ 为学习率. 此改进方案的立足点在于, TD 误差反映了目标值和实际输出值的偏离程度. 如果 TD 误差为正, 说明目标值优于实际值, 当前的突触权重应该加强, 反之应该减弱, 但是, 这种权重调节方案并不能直接应用于 MSRL 算法, 原因在于, 此方案限制每个神经元仅发放一个脉冲, 而 MSRL 中输出层神经元发放的是连续脉冲. 另外, 直接将 TD 误差作为权重调节系数不能最小化误差, 需要定义损失函数.

我们在式 (20) 基础上提出改进的 STDP 规则. 首先, 将 $\xi(\Delta t)$ 简化如下

$$\xi(\Delta t) = \begin{cases} A^+, & \Delta t \geq 0 \\ A^-, & \Delta t < 0 \end{cases} \tag{21}$$

式 (21) 不考虑输入和输出脉冲的时间差, 只考虑输入和输出脉冲之间的相关性. 进一步, 按照文献 [34], 定义损失函数如下

$$L_i(\theta_i) = \mathrm{E}[(y_i - Q(\boldsymbol{s}_t, a_t; \theta_i))^2] \tag{22}$$

其中, $y_i = \mathrm{E}[r_t + \gamma \max_{a_{t+1}} Q(\boldsymbol{s}_{t+1}, a_{t+1}; \theta_{i-1})]$ 为第 i 次迭代的目标 Q 值, θ 为网络参数. 改进 STDP 的目标在于使平方 TD 误差的期望 (即式 (22)) 最小. 最后, 改进的 STDP 规则表示为

$$\frac{\mathrm{d}w_{ij}(t)}{\mathrm{d}t} = \eta L_i(\theta_i) \frac{\xi(\Delta t)}{w_{ij}(t)} \tag{23}$$

在此基础上, 本文还设计了对应的基于忆阻器的人工突触, 以期进一步实现所提出算法的硬件加速. 定义非线性窗函数如下

$$f(v_{\mathrm{MR}}) = \begin{cases} v_{\mathrm{MR}}, & |v_{\mathrm{MR}}| > v_{\mathrm{th}} \\ 0, & |v_{\mathrm{MR}}| \le v_{\mathrm{th}} \end{cases} \tag{24}$$

其中, v_{MR} 为忆阻器两端电压, v_{th} 为忆阻器的阈值电压, 调整忆阻器两端电压大小可使忆阻器处于权重调节或不可更改两个状态.

进一步, 设置权重调节状态时 v_{MR} 为

$$v_{\mathrm{MR}}(\Delta t) = \begin{cases} A^+, & \Delta t \ge 0 \\ A^-, & \Delta t < 0 \end{cases} \tag{25}$$

而突触权重的更新如下

$$\frac{\mathrm{d}w_{ij}(t)}{\mathrm{d}t} = \eta L_i(\theta_i) \frac{f(v_{\mathrm{MR}}(\Delta t))}{w_{ij}(t)} \tag{26}$$

即可实现改进后的 STDP 学习规则.

3 算法流程

在第 2 节基础上, 给出 MSRL (算法 2) 的具体实现流程. 如下所示:

1) 数据收集

强化学习任务开始时, 没有足够的样本用于训练 SNN, 需要通过智能体与环境的交互以获取样本. 此时使权重服从均值和方差均为 0.1 的正态分布, 并通过正则化提高权重收敛速率, 即

$$w_{ij} \leftarrow \frac{w_{ij}}{\sqrt{n}} \tag{27}$$

其中, n 为输入神经元的数量. 另外, 为了消除脉冲之间的相关性, 每个神经元注入了微量的噪声[33]. 每一个时间步 (step) 内, 神经网络的运行时间为两个时间窗口 T. 我们设置输入层和隐含层只在第一个 T 内发放脉冲, 一个 T 的时间过后, 输出层再发放脉冲. 一旦神经网络运行完成, 便得到了输出脉冲数量 Q, 隐含层输出脉冲时间 t_h, 根据 ϵ-greedy 策略, 智能体有 $1-\epsilon$ 的概率选择 Q 最多的神经元所对应的动作, 而以 ϵ 的概率随机选择动作. ϵ-greedy 中 ϵ 的值会随着迭代次数的增加而递减, 以确保随着任务的进行智能体逐渐依赖于策略 $\pi(\boldsymbol{s})$ 而不是无目的的选取动作.

2) 网络训练

根据文献 [35], 突触权重变化会逆行而快速的传播到突触前神经元树突的突触上, 但并不会向前传播到下一级突触上, 这表明类似反向传播算法的机制可以在脉冲神经网络中存在并发挥作用. 因此提出如图 6 所示的训练方式. 图中画出的忆阻器表示此时忆阻器处于权重调节状态, 未画出的忆阻器则处于断开状态. 一次训练包含多个样本, 每一个样本使神经网络运行三个时间窗口 T. 训练时, 首先断开所有忆阻器. 之后使目标动作对应的输出神经元与隐含层之间的忆阻器导通, 这类似于监督学习中利用标签进行训练. 令隐含层神经元发放对应的隐含层脉冲 $\delta_h(t)$, 运行一个时间窗口后, 在第二个时间窗口内令输出神经元发放目标脉冲 $\delta_{y_j}(t)$ (图 6 (a)). 网络运行完两个时间窗口后, 断开隐含层与输出层之间的忆阻器, 使输入层和隐含层之间的忆阻器导通 (图 6 (b)), 令输入神经元发放状态脉冲 $\delta_{\boldsymbol{S}}(t)$, 同时令隐含层神经元再次发放隐含层脉冲 $\delta_h(t)$. 如此循环往复, 直到一次训练完成.

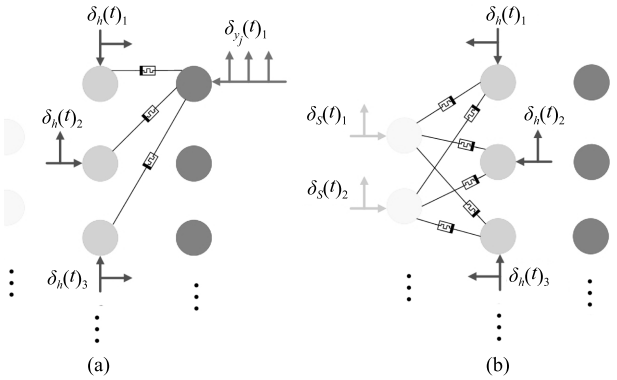


图 6 忆阻脉冲神经网络的训练过程

Fig. 6 The training process of memristive spiking neural network

3) 网络测试

测试时忆阻突触的权重将完全由训练结果决定, 通过设置忆阻器电压, 可以使其工作在权重不可更改状态. 神经网络的运行步骤同训练前.

具体的 MSRL 算法描述如下:

算法 2. 忆阻脉冲神经网络强化学习 (MSRL) 算法

1 初始化容量为 N 的记忆池 $\boldsymbol{D}$

2 初始化观测值 o, 样本容量 b

3 初始化权重

for episode $= 1 : M$ **do**
  初始化状态 $\boldsymbol{s}_0$
  repeat
    运行神经网络, 得到输出层输出脉冲数量 Q_t, 隐含层脉冲发放时间 t_h
    以概率 ϵ 随机选择动作 a_t, 以 $1-\epsilon$ 执行 $a_t = \arg\max Q_t$
    执行动作 a_t, 得到奖赏值 r_t 和下一个状态 $\boldsymbol{s}_{t+1}$
    存储 $\boldsymbol{e}_t = (\boldsymbol{s}_t, a_t, r_t, \boldsymbol{s}_{t+1}, t_h, Q_t)$ 于记忆池 $\boldsymbol{D}$
    if 迭代步数大于 o, **then**
      从 $\boldsymbol{D}$ 中随机抽取 b 个元组 $(\boldsymbol{s}_j, a_j, r_j, \boldsymbol{s}_{j+1}, t_{hj}, Q_j)$ 作为训练样本
      将 $\boldsymbol{s}_{j+1}$ 输入到神经网络中, 得到 Q_{j+1}
      $y_j =$

$$\begin{cases} r_j, & \text{如果任务在 } step_{j+1} \text{ 终止} \\ r_j + \gamma \max\limits_{a_{j+1}} Q_{j+1}, & \text{否则} \end{cases}$$

      目标脉冲 $\delta_{y_j}(t)$ 数量 $n_{y_j} = \text{ceil}(y_j)$
      对每一个动作, 分别按式 (22) 求出其平方 TD 误差的期望
      运用改进 STDP 算法训练神经网络
    $\boldsymbol{s}_t \leftarrow \boldsymbol{s}_{t+1}$
  until $\boldsymbol{s}$ is terminal
end for

4　实验与分析

4.1　实验设置

1) CartPole-v0

如图 7 所示, 一辆小车上用铰链装有一只平衡杆, 平衡杆可以自由移动. 在力 F 的作用下, 小车在离散时间区间内向左或向右移动, 从而改变小车自身的位置和杆的角度. 这个模型的状态有 4 个维度: a) 小车在轨道上的位置 x; b) 平衡杆与垂直方向的夹角 θ; c) 小车速度 v; d) 平衡杆角速度 ω.

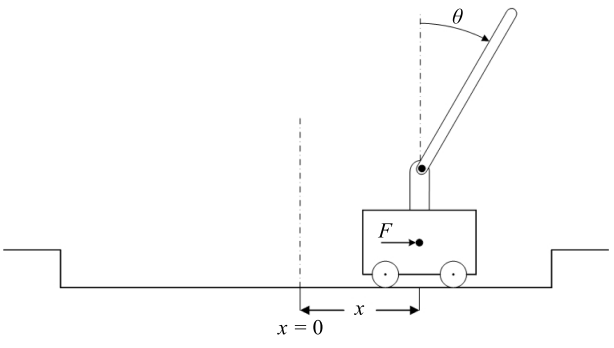


图 7　CartPole-v0 示意图

Fig. 7　CartPole-v0

游戏中每移动一个时间步 (step), 智能体都会通过观察获得下一个状态的值, 并且会获得值为 1 的奖赏. 游戏终止的条件为: a) 平衡杆的角度的绝对值大于 12°; b) 小车的位置与 $x = 0$ 的位置的距离超出 ±2.4 的范围; c) 在一次迭代 (episode) 中 step 数超过 200. 满足条件 c) 则认为游戏成功. 由于摆杆角度和车位移的绝对值较小的情况下游戏容易成功, 因而定义每一步的游戏得分为

$$S_c = \frac{1}{100}\left(\frac{1}{|x|} + \frac{1}{|\theta|}\right) \tag{28}$$

每次游戏得分通过此次游戏总分除以此次游戏迭代步数得到. MSRL 参数设置如下: 对 $\epsilon-$greedy, 设置 $\epsilon = 0.1$, 学习率 η 设置为 0.1, 记忆池容量为 10 000, 折扣因子 γ 为 0.9. 算法运行 500 次迭代.

2) MountainCar-v0

如图 8 所示, 一辆小车被置于两座山峰之间, 小车的初始位置 $x_0 \in (-0.6, -0.4)$, 山谷处的位置为 -0.5. 任务目标是开到右边小旗处. 但是, 车的动力不足以一次爬上山顶, 因此, 小车需要来回移动以获取足够的速度到达目标处. 智能体的状态由两个维度组成: a) 小车轨道位置 $x \in (-1.2, 0.6)$; b) 小车的速度 $y \in (-0.07, 0.07)$.

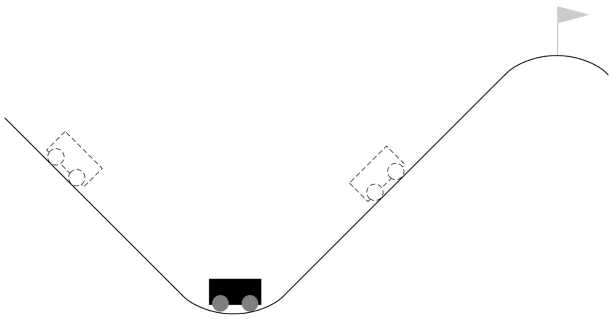


图 8　MountainCar-v0 示意图

Fig. 8　MountainCar-v0

每一个 step 中, 小车有三个动作可供选择: 向右、停止、向左. 小车移动一步后会获得观察值和值为 -1 的奖赏. 根据小车与终点的距离, 定义每步游戏得分 S_m 为

$$S_m = \frac{1}{0.6 - x} \tag{29}$$

每次游戏得分计算方式与 CartPole-v0 相同. 另外, 设定当一次迭代中步数超过 300 游戏也会自动结束. MSRL 参数设置如下: 对 $\epsilon-$ greedy, 同样设置 $\epsilon = 0.1$, 学习率 η 设置为 0.1, 记忆池容量为 5 000, 折扣因子 γ 为 0.9. 算法运行 100 次迭代.

3) 隐含层神经元数量

为了确定 SNN 隐含层神经元的数量, 我们在其他实验参数相同的情况下分别独立运行了隐含层神

经元数量不同的 MSRL 算法, 并比较它们的 TD 方差, 结果见表 1. 表 1 的列展示了隐含层神经元数量不同的情况下 TD 方差的大小, 在其他参数相同的条件下进行实验. TD 方差小表明学习效率更高. CartPole-v0 的输入神经元为 4 个, MountainCar-v0 为 2 个.

表 1 不同隐含层神经元数量 TD 方差对比
Table 1 The comparison of TD variance for different hidden neurons

任务	CartPole-v0	MountainCar-v0
Hidden = 1	27.14	5.17
Hidden = 2	24.52	5.03
Hidden = 4	21.2	4.96
Hidden = 6	19.45	4.87
Hidden = 10	17.26	4.79
Hidden = 12	14.04	4.65

从表 1 中可以发现, 隐含层神经元数量较少, TD 方差较大, 但数量过多并没有显著提高学习效率, 反而可能会增加网络复杂度, 减慢运行速率. 因此我们设置 CartPole-v0 隐含层神经元数量为 6, MountainCar-v0 隐含层神经元数量为 4, 作为折中的一种优化选择.

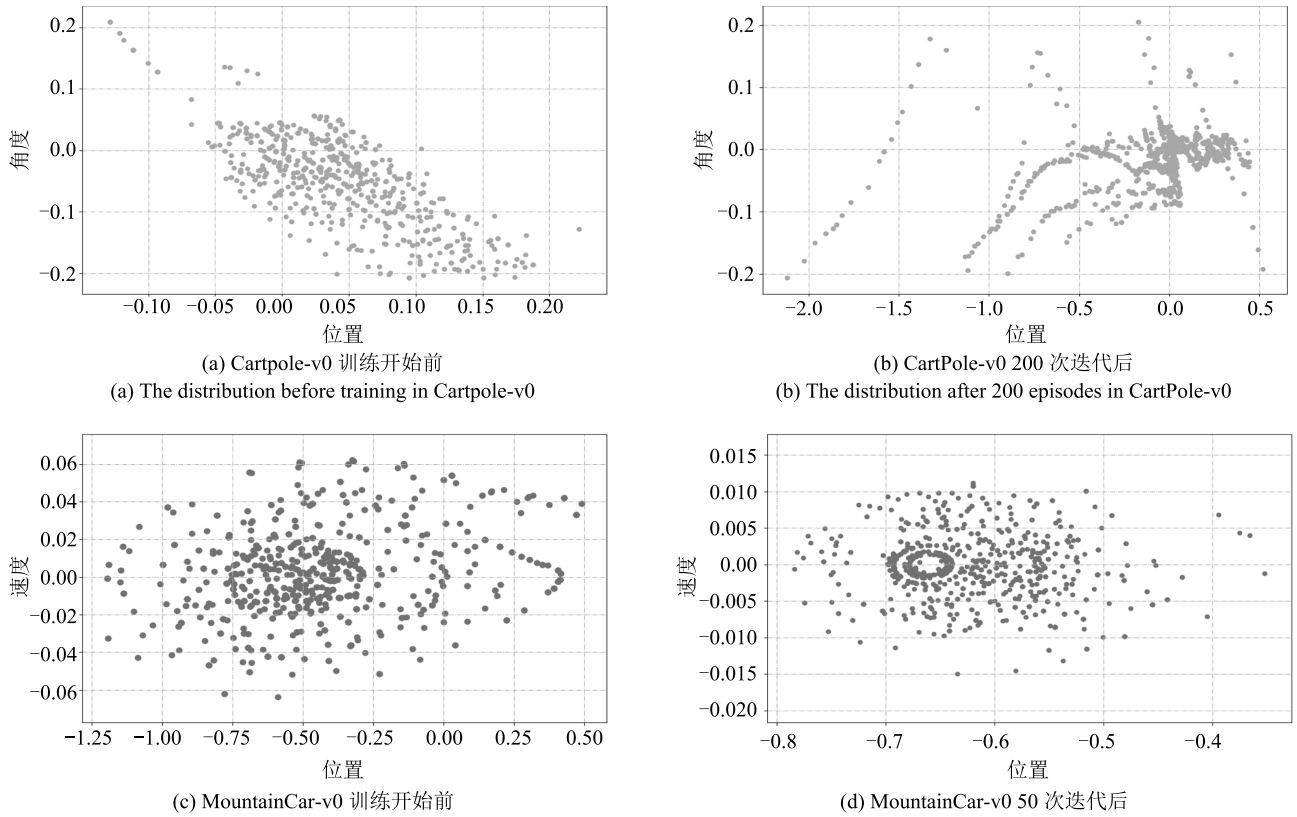
4.2 实验结果与分析

1) MSRL 有效性验证

在实验过程中智能体的状态反映了学习效果. Cartpole-v0 游戏中, 平衡杆的角度和小车的位移越小越好, 这样游戏才可能成功. 而 MountainCar-v0 游戏中, 小车在速率足够大的情况下才能爬上右侧山坡, 到达目标. 我们分别在训练开始前和训练开始后随机抽取相同数量的样本以观察样本的数值分布, 结果如图 9 所示. 可以看出, 在 CartPole-v0 中, 当完成了 200 次游戏后, 平衡杆的角度和小车的位置集中于原点附近. 而在 MountainCar-v0 中, 完成了 50 次游戏后, 坐标值的变化显示小车学会了利用左侧山坡获得反向势能, 并且速率大于训练之前.

2) 算法对比

为进一步说明 MSRL 的特点, 我们将深度 Q 网络 (Deep Q network, DQN) 和离散状态 Q-learning



(a) Cartpole-v0 训练开始前
(a) The distribution before training in Cartpole-v0

(b) CartPole-v0 200 次迭代后
(b) The distribution after 200 episodes in CartPole-v0

(c) MountainCar-v0 训练开始前
(c) The distribution before training in MountainCar-v0

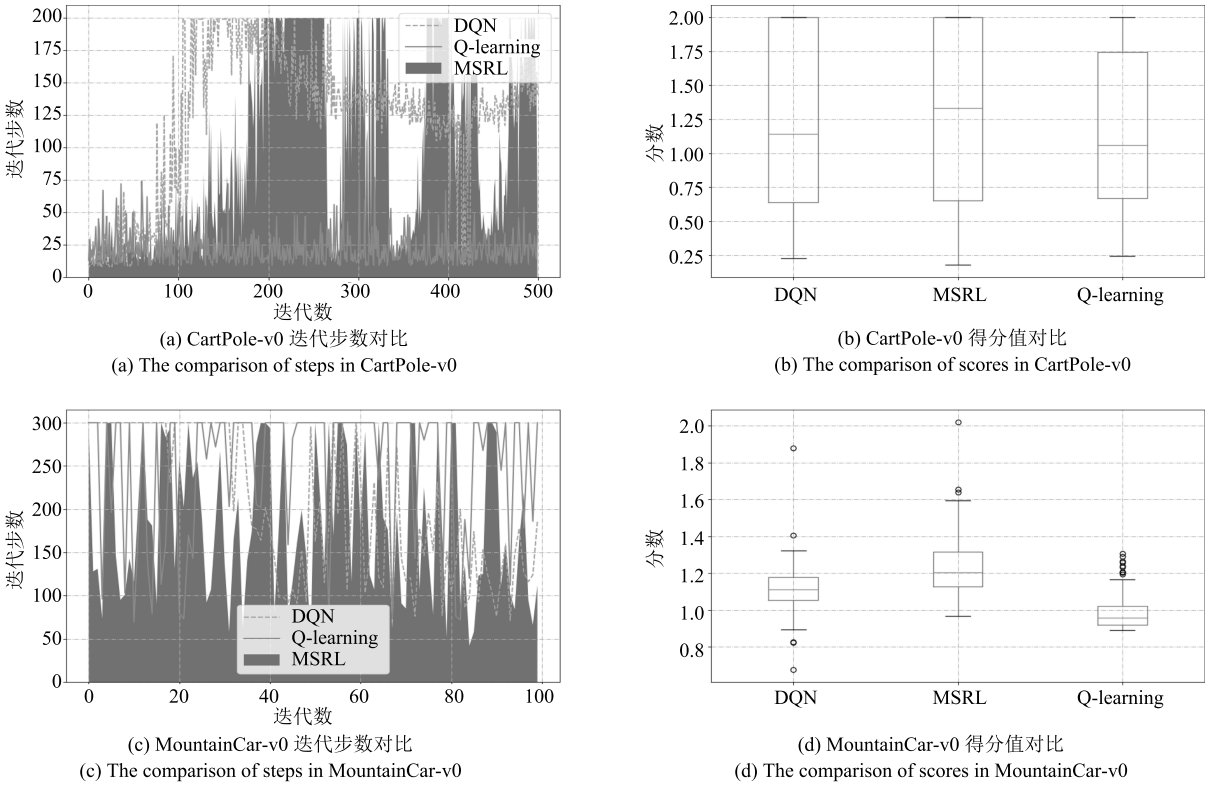
(d) MountainCar-v0 50 次迭代后
(d) The distribution after 50 episodes in MountainCar-v0

图 9 MSRL 训练前后样本状态分布对比
Fig. 9 The comparison of sample states distribution before and after training of MSRL

作为比较的对象. 三者折扣因子和学习率均相同, DQN 同样采用三层全连接前向网络结构, 隐含层神经元数量为 10, 且其记忆池容量与 MSRL 相同. 三个算法在同一台计算机上分别独立运行. 对比结果如图 10 和表 2 所示.

根据游戏环境的设置, 在 CartPole-v0 游戏中每次游戏的迭代步数越高越好, 而 MountainCar-v0 则相反. 图 10 (a) 和 10 (b) 的结果显示, 在 CartPole-v0 游戏中, MSRL 算法所控制的倒立摆系统游戏成功率和得分高于另外两种算法. 尽管 DQN 先于 MSRL 算法完成游戏目标, 但其收敛性较差. 图 10 (c) 和 10 (d) 的结果显示, 在 MountainCar-v0 游戏中, MSRL 算法所控制的小车容易以较少的步数达到目标处, 且最少步数小于另外两种算法, 同时游戏得分为三者中的最高值. 从两个游戏的结果可以看出, 离散状态之后的 Q-learning 算法难以达成目标. 我们将结果列在表 2 里以更清楚对比结果.

表 2 中, 平均迭代步数为实验中的累积步数除以迭代数, 而平均分数为累积分数除以累积步数. 在 CartPole-v0 游戏中, MSRL 算法总平均迭代步数相比于 DQN 和离散 Q-learning 明显增加, 而在 MountainCar-v0 游戏中, MSRL 算法总平均迭代步数相比于 DQN 和离散 Q-learning 明显减少. 两个游戏中得分最高者均为 MSRL. 此外, 我们还在



(a) CartPole-v0 迭代步数对比
(a) The comparison of steps in CartPole-v0

(b) CartPole-v0 得分值对比
(b) The comparison of scores in CartPole-v0

(c) MountainCar-v0 迭代步数对比
(c) The comparison of steps in MountainCar-v0

(d) MountainCar-v0 得分值对比
(d) The comparison of scores in MountainCar-v0

图 10　比较结果 (A)

Fig. 10　The results of comparison (A)

表 2　比较结果 (B)

Table 2　The results of comparison (B)

评价指标	平均迭代步数	平均分数	平均 CPU 利用率 (%)	运行时间 (s)
MSRL (CartPole-v0)	98.93	1.28	12.0	3 528.38
DQN (CartPole-v0)	61.79	1.22	23.5	1 119.52
Q-learning (CartPole-v0)	11.83	1.14	0.3	105.60
MSRL (MountainCar-v0)	183.87	1.23	11.8	1 358.14
DQN (MountainCar-v0)	204.32	1.12	22.9	359.21
Q-learning (MountainCar-v0)	250.26	0.98	0.2	32.68

游戏执行的每一步中记录 CPU 利用率, 最后用累积 CPU 利用率除以累积步数以计算平均 CPU 利用率. 结果显示, 尽管 Q-learning 能以较短的运行时间和较低的 CPU 利用率完成目标, 但是其计算效果不如神经网络式强化学习. 而 MSRL 算法 CPU 利用率低于 DQN, 但运行时间长于 DQN. 根据文献 [36], 采用不同的模拟策略影响脉冲神经网络的运行时间. 而本文利用新型信息器件忆阻器的高密度、非易失性等优势, 融合优化的网络结构和改进的学习算法, 有望以实现 MSRL 的硬件加速, 同时减少对计算资源的依赖.

5 结论

尽管传统的神经网络与强化学习算法的结合提高了智能体的学习能力, 但这些算法对计算能力依赖性较强, 同时网络复杂度高, 不适合硬件实现. 为了进一步达到硬件加速, 促进嵌入式智能体在实际环境中独立执行任务, 本文设计了基于多层忆阻脉冲神经网络的强化学习 (MSRL) 算法. 首先解决了数据与脉冲之间的转换问题; 在前人工作基础上, 改进了 STDP 学习规则, 使 SNN 能够与强化学习有机结合, 同时也设计了相应的忆阻突触; 进一步, 设计了结构可动态调整的多层忆阻脉冲神经网络, 这种网络具有较高的学习效率和适应能力. 实验结果表明, MSRL 与传统的强化学习算法相比能更高效地完成学习任务, 同时更节省计算资源. 在未来的工作中, 我们将研究深度 SNN 与更复杂的强化学习算法例如 Actor-Critic 算法的结合, 并进一步改进学习算法以增强算法稳定性.

References

1 Gao Yang, Chen Shi-Fu, Lu Xin. Research on reinforcement learning: a review. *Acta Automatica Sinica*, 2004, **30**(1): 86−100
(高阳, 陈世富, 陆鑫. 强化学习研究综述. 自动化学报, 2004, **30**(1): 86−100)

2 Tang Hao, Wan Hai-Feng, Han Jiang-Hong, Zhou Lei. Coordinated look-ahead control of multiple CSPS system by multi-agent reinforcement learning. *Acta Automatica Sinica*, 2010, **36**(2): 289−296
(唐昊, 万海峰, 韩江洪, 周雷. 基于多 Agent 强化学习的多站点 CSPS 系统的协作 Look-ahead 控制. 自动化学报, 2010, **36**(2): 289−296)

3 Qin Rui, Zeng Shuai, Li Juan-Juan, Yuan Yong. Parallel enterprises resource planning based on deep reinforcement learning. *Acta Automatica Sinica*, 2017, **43**(9): 1588−1596
(秦蕊, 曾帅, 李娟娟, 袁勇. 基于深度强化学习的平行企业资源计划. 自动化学报, 2017, **43**(9): 1588−1596)

4 Watkins C J C H, Dayan P. Q-learning. *Machine Learning*, 1992, **8**(3−4): 279−292

5 Maass W. Networks of spiking neurons: the third generation of neural network models. *Neural Networks*, 1997, **10**(9): 1659−1671

6 Florian R V. Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity. *Neural Computation*, 2007, **19**(6): 1468−1502

7 Cao Y Q, Chen Y, Khosla D. Spiking deep convolutional neural networks for energy-efficient object recognition. *International Journal of Computer Vision*, 2015, **113**(1): 54−66

8 Ghosh-Dastidar S, Adeli H. Spiking neural networks. *International Journal of Neural Systems*, 2009, **19**(4): 295−308

9 Ponulak F. Analysis of the ReSuMe learning process for spiking neural networks. *International Journal of Applied Mathematics and Computer Science*, 2008, **18**(2): 117−127

10 Mostafa H. Supervised learning based on temporal coding in spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, **29**(7): 3227−3235

11 de Kamps M, van der Velde F. From artificial neural networks to spiking neurons and back again. *Neural Networks*, 2001, **14**(6−7): 941−953

12 Zheng N, Mazumder P. Learning in memristor crossbar-based spiking neural networks through modulation of weight dependent spike-timing-dependent plasticity. *IEEE Transactions on Nanotechnology*, 2018, **17**(3): 520−532

13 Taherkhani A, Belatreche A, Li Y H, Maguire L P. A supervised learning algorithm for learning precise timing of multiple spikes in multilayer spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2018, **29**(11): 5394−5407

14 Chua L O. Memristor — the missing circuit element. *IEEE Transactions on Circuit Theory*, 1971, **18**(5): 507−519

15 Strukov D B, Snider G S, Stewart D R, Williams R S. The missing memristor found. *Nature*, 2008, **453**(7191): 80−83

16 Kvatinsky S, Friedman E G, Kolodny A, Weiser U C. TEAM: threshold adaptive memristor model. *IEEE Transactions on Circuits and Systems I — Regular Papers*, 2013, **60**(1): 211−221

17 Hu X F, Feng G, Liu L, Duan S K. Composite characteristics of memristor series and parallel circuits. *International Journal of Bifurcation and Chaos*, 2015, **25**(8): 1530019

18 Jo S H, Chang T, Ebong I, Bhadviya B B, Mazumder P, Lu W. Nanoscale memristor device as synapse in neuromorphic systems. *Nano letters*, 2010, **10**(4): 1297−1301

19 Panwar N, Rajendran B, Ganguly U. Arbitrary spike time dependent plasticity (STDP) in memristor by analog waveform engineering. *IEEE Electron Device Letters*, 2017, **38**(6): 740−743

20 Serrano-Gotarredona T, Masquelier T, Prodromakis T, Indiveri G, Linares-Barranco B. STDP and STDP variations with memristors for spiking neuromorphic learning systems. *Frontiers in Neuroscience*, 2013, **7**(2), DOI: 10.3389/fnins.2013.00002

21 Goodman D F M, Brette R. The brian simulator. *Frontiers in Neuroscience*, 2009, **3**(2): 192−197

22 Sutton R S, Barto A G. *Reinforcement Learning: An Introduction*. London: The MIT Press, 1999. 185−187

23 Ferré P, Mamalet F, Thorpe S J. Unsupervised feature learning with winner-takes-all based STDP. *Frontiers in Computational Neuroscience*, 2018, **12**(24), DOI: 10.3389/fncom.2018.00024

24 Gerstner W, Kistler W M. *Spiking Neuron Models*. New York: Cambridge University Press, 2002.

25 Hasselmo M E. Methods in neuronal modeling: from ions to networks. *Science*, 1998, **282**(5391): 1055−1055

26 Hebb D O, Martinez J L, Glickman S E. The organization of behavior: a neuropsychological theory. *Contemporary Psychology*, 1994, **39**(11): 1018−1020

27 Markram H, Lubke J, Frotscher M, Sakmann B. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 1997, **275**(5297): 213−215

28 Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. In: Proceedings of the 4th International Conference on Learning Representations. Puerto Rico, San Juan: Cornell University Library, 2016.

29 Gollisch T, Meister M. Rapid neural coding in the retina with relative spike latencies. *Science*, 2008, **319**(5866): 1108−1111

30 Kostal L, Lansky P, Rospars J P. Neuronal coding and spiking randomness. *European Journal of Neuroscience*, 2007, **26**(10): 2693−2701

31 Legenstein R, Pecevski D, Maass W. A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. *Plos Computational Biology*, 2008, **4**(10): e1000180

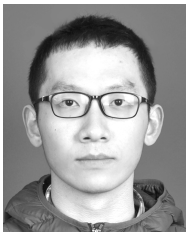
32 Skorheim S, Lonjers P, Bazhenov M. A spiking network model of decision making employing rewarded STDP. *Plos One*, 2014, **9**(3), DOI: 10.1371/journal.pone.0090821

33 Zheng N, Mazumder P. Hardware-friendly actor-critic reinforcement learning through modulation of spike-timing-dependent plasticity. *IEEE Transactions on Computers*, 2017, **66**(2): 299−311

34 Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing atari with deep reinforcement learning. In: Proceedings of the 26th Conference and Workshop on Neural Information Processing Systems, Nevada, USA: Cornell University Library. 2013.

35 Li C Y, Lu J T, Wu C P, Duan S M, Poo M M. Bidirectional modification of presynaptic neuronal excitability accompanying spike timing-dependent synaptic plasticity, *Neuron*, 2004, **41**(2): 257−268

36 Brette R, Rudolph M, Carnevale T, Hines M, Beeman D, Bower J M, et al. Simulation of networks of spiking neurons: a review of tools and strategies. *Journal of Computational Neuroscience*, 2007, **23**(3): 349−398



张耀中 西南大学计算机与信息科学学院本科生. 主要研究方向为强化学习, 脉冲神经网络理论与应用.
E-mail: zhangyaozhong9@126.com
(**ZHANG Yao-Zhong** Undergraduate at the College of Computer and Information Science, Southwest University. His research interest covers reinforcement learning, theories and applications of spiking neural networks.)

胡小方 西南大学人工智能学院副教授. 2015 年获得中国香港城市大学机械与生物医学工程系博士学位. 主要研究方向为忆阻器件与系统应用, 神经网络算法, 模型与硬件实现, 强化学习, 图像处理. 本文通信作者.
E-mail: huxf@swu.edu.cn
(**HU Xiao-Fang** Associate professor at the College of Artificial Intelligence, Southwest University. She received her Ph. D. degree from City University of Hong Kong, China in 2015. Her research interest covers memristive devices and system applications, neural network algorithm, model and hardware implementation, reinforcement learning, image processing. Corresponding author of this paper.)

周 跃 西南大学电子信息工程学院研究助理. 2012 年获得南京大学工程管理学院硕士学位. 主要研究方向为机器学习, 深度学习, 信息安全, 忆阻器件与系统. E-mail: zhouyuenju@163.com
(**ZHOU Yue** Research assistant at the College of Electronic and Information Engineering, Southwest University. He received his master degree from Nanjing University in 2012. His research interest covers machine learning, deep learning, information security, memristor devices and systems.)

段书凯 西南大学人工智能学院教授. 2006 年获得重庆大学计算机科学学院博士学位. 主要研究方向为纳米信息器件与系统, 神经形态计算系统, 非线性电路与系统, 机器学习.
E-mail: duansk@swu.edu.cn
(**DUAN Shu-Kai** Professor at the College of Artificial Intelligence, Southwest University. He received his Ph. D. degree from Chongqing University in 2006. His research interest covers nano-information devices and systems, nonlinear circuits and systems, and machine learning.)