控制与决策
Control and Decision

# 基于径向基神经网络的多步Sarsa控制算法

司彦娜, 普杰信, 于晓升, 司鹏举, 孙力帆

引用本文:
司彦娜,普杰信,于晓升,司鹏举,孙力帆. 基于径向基神经网络的多步Sarsa控制算法[J]. 控制与决策, 2023, 38(4): 944−950.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.1728

---

# 基于径向基神经网络的多步Sarsa控制算法

司彦娜[1], 普杰信[1†], 于晓升[2], 司鹏举[1], 孙力帆[1]

(1. 河南科技大学 信息工程学院，河南 洛阳 471023；2. 东北大学 机器人科学与工程学院，沈阳 110169)

**摘 要:** 针对具有连续状态空间的无模型非线性系统,提出一种基于径向基(radial basis function, RBF)神经网络的多步强化学习控制算法. 首先,将神经网络引入强化学习系统,利用RBF神经网络的函数逼近功能近似表示状态-动作值函数,解决连续状态空间表达问题;然后,结合资格迹机制形成多步Sarsa算法,通过记录经历过的状态提高系统的学习效率;最后,采用温度参数衰减的方式改进softmax策略,优化动作的选择概率,达到平衡探索和利用关系的目的. MountainCar任务的仿真实验表明:所提出算法经过少量训练能够有效实现无模型情况下的连续非线性系统控制;与单步算法相比,该算法完成任务所用的平均收敛步数更少,效果更稳定,表明非线性值函数近似与多步算法结合在控制任务中同样可以具有良好的性能.

**关键词:** RBF神经网络；强化学习；Sarsa算法；连续空间；值函数近似；资格迹

**中图分类号:** TP181 **文献标志码:** A

**DOI:** 10.13195/j.kzyjc.2021.1728

**引用格式:** 司彦娜, 普杰信, 于晓升, 等. 基于径向基神经网络的多步Sarsa控制算法 [J]. 控制与决策, 2023, 38(4): 944-950.

# Multi-step Sarsa control algorithm based on RBF neural network

*SI Yan-na*[1], *PU Jie-xin*[1†], *YU Xiao-sheng*[2], *SI Peng-ju*[1], *SUN Li-fan*[1]

(1. College of Information Science and Engineering，Henan University of Science and Technology，Luoyang 471023，China；2. Faculty of Robot Science and Engineering，Northeastern University，Shenyang 110169，China)

**Abstract:** For a model-free nonlinear system with continuous state space, a multi-step reinforcement learning control algorithm based on the RBF neural network is proposed. Firstly, the neural network is introduced to a reinforcement learning system for approximating the state-action value function, which is a common solution to the problem of continuous state space expression in reinforcement learning. Then, combined with the eligibility trace mechanism，multi-step algorithm Sarsa($\lambda$) is formed to improve the learning efficiency of the system by recording the experienced states. Finally, the softmax strategy is improved by decayed temperature parameters, so as to optimize the selection probability of actions and balance the relationship between exploration and exploitation. The simulation results of the MountainCar task show that the proposed algorithm can effectively achieve the model-free control task of the continuous nonlinear system through fewer times of training. Compared with the single-step algorithm, the multi-step algorithm takes less average convergent steps to complete the task and perform more stable, which proves that the combination of nonlinear value function approximation and the multi-step algorithm has good performance in the control task.

**Keywords:** RBF neural network; reinforcement learning; Sarsa algorithm; continuous space; value function approximatio; eligibility trace

## 0 引 言

近年来,强化学习(reinforcement learning, RL)在人工智能领域取得了举世瞩目的成就,尤其在围棋和游戏方面,结合深度学习技术的智能体,经过训练优化在诸多场景中的成绩均已超越人类顶级选手[1-3]. 然而,强化学习的灵感来源于人和动物的学习行为,本质上是一种“trial and error”的学习方式,通过实践积累经验达到学习目的. 这种模式用于机器学习中,往往需要数以万计的试错训练才能达到理想的效果. 现实中的控制系统通常难以进行大量采样,并且试错学习可能对系统硬件带来损害. 因此,强化学习最出色的成果目前主要集中在游戏等虚拟场景

**收稿日期:** 2021-10-09；**录用日期:** 2021-12-30.

**基金项目:** 航空科学基金项目(20185142003)；国家国防基础研究计划项目(JCKY2018419C001)；河南省高等学校重点科研项目(20A120008)；河南省自然科学基金项目(202300410149).

**责任编委:** 张国山.

†通讯作者. E-mail: pjx@haust.edu.cn.

中,实际应用具有一定困难,尚处在发展初期.

随着研究工作的不断深入,利用强化学习方法解决传统控制领域模型未知、缺少先验知识或高度动态化的优化决策问题逐渐成为当前的研究重点之一[4-5]. 经典的强化学习算法建立在马尔科夫决策过程(Markov decision process, MDP)的基础之上,主要针对有限的离散空间、状态或状态-动作对的有关信息使用表格存储、更新. 当面对大规模任务时,例如围棋,状态数可达$10^{170}$个,表格型强化学习无法有效应对. 而机械手、无人机及移动机器人等,其状态和动作均属于连续空间,离散化处理也可能遭遇“维数灾难”,导致算法学习速度过慢甚至无法收敛.

针对此类情况,研究人员利用函数逼近的思想对价值函数进行参数化来表示代替表格存储,通过学习逼近器参数间接获得最优策略,并取得了较好的效果. 目前,常用的值函数近似方法主要包括线性多项式[6-8]和各种神经网络[9-12]. 其中,线性近似方法易于理解且具有可靠的收敛保证,非线性方法具有更强大的逼近能力,实用性更强[13]. 然而,利用非线性函数逼近的多步强化学习方法在诸多领域表现不佳,造成神经网络等非线性函数逼近在单步强化学习算法中更为普遍,而多步方法则与线性函数逼近结合更多的现象[14-15].

现有研究显示,多步更新在时序差分(temporal difference, TD)算法中有着重要作用,因为TD 算法的核心是自举(bootstrapping),即通过后一个状态的值函数计算当前状态的值函数,多步更新通过控制自举的起始状态可以影响更新目标的偏差与方差. 二者的最佳平衡介于单步更新(高偏差、低方差)与回合更新(无偏差、高方差)之间,即多步更新[16]. 因此,多步方法理论上比单步方法具有更好的性能. RBF神经网络结构简单、非线性逼近能力强,在模式识别[17]、数据挖掘[18]和控制工程[19]等研究领域应用广泛,效果显著. 利用RBF神经网络近似表示值函数时,可以看作一组基函数与一个参数向量的乘积,具有一定的线性逼近特征.

根据上述分析,本文提出一种基于径向基神经网络的多步强化学习算法,将RBF神经网络与Sarsa($\lambda$)算法相结合,用于解决具有连续状态空间的强化学习问题. 算法利用RBF网络近似表示状态-动作值函数,有效避免了表格型强化学习可能出现的“维数灾难”. 同时,引入资格迹(eligibility traces)机制,记录经历过的状态信息,加快系统的收敛速度. 然后,根据softmax选择策略,为值函数分配权重,优化动作的选择概率,平衡探索和利用的关系,进一步提高学习效率. 最后,通过Mountaincar任务验证所提出算法的正确性和有效性.

## 1 基于RBF神经网络的强化学习系统

强化学习不需要精确的系统模型和外部指导,仅通过不断地“试错”过程积累经验,学习最佳控制策略. 在典型的强化学习系统中,智能体是学习者和决策者,状态是对环境信息的描述;动作是智能体对环境作出的反应;奖励是环境对动作的评价. 学习过程中,智能体先观察环境,根据得到的状态信息选取合适的动作;环境接收到动作,做出相应的反馈,同时进入到新的状态;智能体获得来自环境的标量奖励,指导下一步动作.

智能体和环境在每个离散的时间步相互作用,在不断地循环交互中,根据环境的奖赏值适当调整动作,以获得最多奖赏为目标. 通常,回报值定义为每一步的奖励值随时间的加权累积和,即

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots . \tag{1}$$

其中: $R_t$为完成动作后得到的即时奖励, $\gamma \in [0,1]$表示折扣因子,用来平衡未来奖励相对于当前奖励的比重.

通过最大化回报值,可以获得最佳行动策略. 但是,同一状态可能包含多个不同动作,对应的奖励也有所不同,累积回报并非确定值. 为了更好地描述执行策略$\pi$时的长期价值,将其在状态$s$处的期望定义为状态值函数,有

$$\begin{aligned} &v_\pi(s) = \\ &E_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots | s_t = s] = \\ &E_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s\Big]. \end{aligned} \tag{2}$$

类似地,遵循策略$\pi$,在状态$s$处执行动作$a$的价值定义为状态-动作值函数,有

$$q_\pi(s,a) = E_\pi\Big[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s, a_t = a\Big]. \tag{3}$$

对于简单的控制任务,可以利用表格存储和计算值函数,表格的更新即为值函数的更新. 然而,在处理具有连续空间的强化学习问题时,表格型方法往往难以奏效,通常采用值函数近似方法,对价值函数进行参数化表示,参数的更新相当于值函数的更新. 此时,状态-动作值函数的近似表达式为

$$q(s,a) \approx \hat{q}(s,a;w). \tag{4}$$

其中: $w$为逼近器参数,每个参数值都相应地确定一个值函数.

人工神经网络的非线性逼近优势明显,实际应用广泛. 其中,前馈类型的RBF神经网络,输入层到隐含层为非线性变换,而隐含层到输出层为线性变换,并且两个层间的变换参数可以分别进行学习,相较于其他类型的神经网络,其学习速度较快且可以避免局部极小问题[20]. 如图1所示,在结合RBF神经网络的强化学习系统中,将每一时刻观察到的状态信息作为神经网络的输入,输出为所有可能动作的值函数,即

$$y_n = \hat{q}(s, a_n, w). \tag{5}$$

其中:$y_n$为第$n$个输出层神经元,$a_n$为第$n$个可能执行的动作.

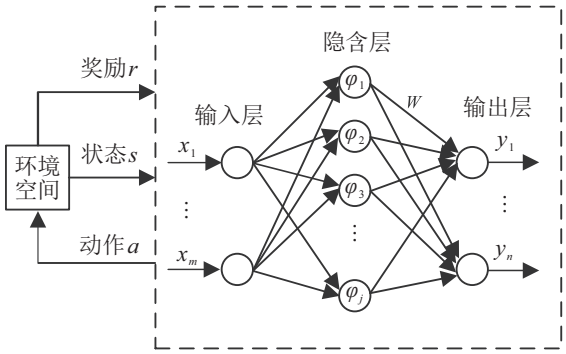


**图 1 基于RBF神经网络的强化学习系统**

通常,RBF神经网络采用高斯核(Gaussian kernel)作为径向基函数,隐含层的输出为

$$\varphi(x) = \exp\left(-\frac{\|x - c_j\|^2}{{\sigma_j}^2}\right). \tag{6}$$

其中:$x = [x_1, x_2, \ldots, x_m]$为网络的输入向量,$c_j$为第$j$个基函数的中心向量,$\sigma_j$为第$j$个基函数的宽度.

此时,神经网络的输出计算公式为

$$y_n = \sum_{1}^{j} \varphi_j(x) w_{jn}. \tag{7}$$

其中:$\varphi_j$为第$j$个隐含层神经元,$w_{jn}$为第$n$个输出层神经元的权重.

## 2 Sarsa($\lambda$)算法的神经网络学习过程

### 2.1 Sarsa($\lambda$)算法

智能体每次执行完动作后,值函数得到更新,系统状态也随之发生改变. 根据值函数更新规则的不同,强化学习算法也有所差别. Sarsa算法属于单步更新算法的一种,即每一时刻对当前价值函数的估算仅用到相邻下一时刻的值函数,其值函数更新的规则如下:

$$Q(s_{t+1}, a_{t+1}) \leftarrow$$
$$Q(s,a) + \alpha[R + \gamma Q(s_{t+1}, a_{t+1}) - Q(s,a)], \tag{8}$$

其中$\alpha \in [0,1]$为学习步长.

由于值函数更新和动作选择采用同一种策略,Sarsa可以看作是on-policy的TD学习算法[21]. 通常将减小TD误差作为网络学习的依据,代价函数定义为

$$E(t) = \frac{1}{2}(\varepsilon_{\mathrm{TD}}(t))^2. \tag{9}$$

其中$\varepsilon_{\mathrm{TD}}(t)$为TD误差,计算为

$$\varepsilon_{\mathrm{TD}}(t) = R_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t). \tag{10}$$

此时,网络权值的更新规则为

$$\Delta w(t) = \alpha \cdot \varepsilon_{\mathrm{TD}}(t) \frac{\partial Q(s_t, a_t)}{\partial w(t)} =$$
$$\alpha(R_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) \frac{\partial Q(s_t, a_t)}{\partial w(t)}. \tag{11}$$

强化学习面向的是序贯决策问题,智能体每个时刻的行为对最终的结果均有一定影响. 为了充分利用任务完成前的历史经验,加快系统学习速度,在Sarsa算法中引入资格迹机制,形成Sarsa($\lambda$)算法. 一般情况下,资格迹定义为

$$e(t) = \sum_{k=1}^{t} \lambda^{t-k} \frac{\partial Q(s_k, a_k)}{\partial w(k)}. \tag{12}$$

加入资格迹后,网络权值的更新规则变为

$$\Delta w(t) = \alpha \cdot \varepsilon_{\mathrm{TD}}(t) \cdot e(t) =$$
$$\alpha(R_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)) e(t). \tag{13}$$

资格迹记录了$t$时刻前的历史梯度累积和,在训练过程中,可以将一步误差向后传播至整个状态空间,只有经历过的状态才会对当前值函数更新产生影响,从而加快收敛速度.

### 2.2 改进的Softmax动作选择策略

在强化学习过程中,智能体每个时刻根据环境的反馈调整动作,目标是在任务结束时获得最多的累积奖赏. 对于动作选择,如何平衡探索与利用(exploration and exploitation)的关系是强化学习中一个非常重要的问题,也是决定智能体能否学习到最优策略的关键之一.

目前,使用最多的是$\varepsilon$-greedy策略,即每次以$\varepsilon$的概率随机探索动作,以$(1-\varepsilon)$的概率利用当前平均奖励最高的动作. 该策略中和了随机策略(random strategy)和贪婪策略(greedy strategy)的特点,前者每次选择随机动作,始终处于探索状态;后者每次选择最高奖励对应的动作,只考虑当前经验. 两种策略都很难学习到最优解决方案,甚至无解.

虽然,$\varepsilon$-greedy策略令所有动作都有被选择的可能,但每个动作的概率是相同的. 实际上,期望奖励高的动作应该具有更高的选择概率,尤其当动作数量较

多时. 为了尽可能优化动作的选择概率,帮助更快地学习到好的策略,根据Boltzmann分布,可以为每个动作分配概率权重,即softmax选择策略

$$P(a_i)=\frac{\exp(Q(s,a_i)/\tau)}{\sum_{i=1}^{k}\exp(Q(s,a_i)/\tau)}. \tag{14}$$

其中: $a_i$为第$i$个可能的动作, $\tau>0$为温度控制参数.

由于$\tau$值越大越倾向于随机选择动作,反之则倾向于选择平均奖赏最高的动作,让$\tau$随时间衰减更符合实际情况. 一方面可以避免繁琐的参数试凑,另一方面可以有效平衡探索和利用(exploration-exploitation)的关系. 同时,为了避免$\tau$值过大或过小带来的计算问题,将softmax函数改写成对数形式为

$$P(a_i)=\exp\left(\frac{Q(s,a_i)}{{\tau_0}^n}-\ln\sum_{i=1}^{k}\exp\left(\frac{Q(s,a_i)}{{\tau_0}^n}\right)\right). \tag{15}$$

其中: $\tau_0$为初始温度参数, $n$为时间步.

### 2.3　算法步骤描述

根据以上分析,基于RBF神经网络的Sarsa($\lambda$)算法步骤如下.

step 1: 初始化学习步长$\alpha$,折扣因子$\gamma$,资格迹参数$\lambda$,温度参数$\tau$及权重$W$和资格迹$e$;

step 2: 观察当前状态$s\leftarrow(x,v)$;

step 3: 计算网络输出$Y=\varphi(s)W$;

step 4: 根据改进的softmax策略选择动作$a$;

step 5: 获得奖励$r$,观察新状态$s'\leftarrow(x,v)$;

step 6: 由式(7)计算资格迹$e(t)$;

step 7: 计算新的网络输出$Y'=\varphi(s')W$;

step 8: 根据改进的softmax策略选择动作$a'$;

step 9: 由式(8)更新权重$W$;

step 10: 循环step 2 ~ step 9,直到满足结束条件.

## 3　仿真实验

### 3.1　MountainCar任务

为了验证所提出算法在具有连续状态空间的非线性系统中的有效性,选择MountainCar任务进行仿真实验. 如图2所示,任务描述了一辆处在山谷中的小车,旗子的位置代表山峰的最高处,学习目标是让小车用最少的尝试步数到达旗子所在位置. 但是,小车动力不足无法一次到达山峰,需要不停地向左、向右移动来积累动量帮助完成任务.

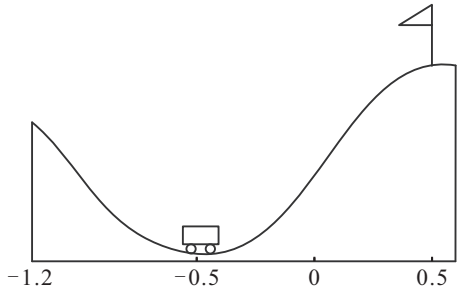


图 2　MountainCar任务示意图

在强化学习研究中,该任务作为典型的连续状态空间问题被广泛用于验证算法的学习性能. 系统的状态包括水平方向上的位移$x$和速度$v$,属于二维连续变量;动作包括向左、向右和静止,属于离散变量. 系统的动力学特征可以表示为

$$\begin{cases}\dot{x}=v,\\ \dot{v}=0.001u-g\cos(3x).\end{cases} \tag{16}$$

其中: $u$为作用在小车上的控制力, $g$为重力相关的参数. 状态空间约束满足

$$\{(x,v)\in R^2|-1.2\leqslant x\leqslant 0.6,-0.07\leqslant v\leqslant 0.07\}. \tag{17}$$

在缺少精确模型的条件下,传统控制方法难以完成该任务. 而强化学习算法仅需观察到的状态信息及简单的奖励,即可令小车学习以最少的尝试步数从任意初始位置到达目标点,有效完成任务.

### 3.2　实验结果

实验过程中,根据现有经验选择固定学习步长$\alpha=0.01$,折扣因子$\gamma=0.95$;统一设置每次实验回合数(episodes)为300,最大实验步数(steps)为2 000. 小车到达目标点或达到最大实验步数则一个episode结束.

**实验1**　进行消融实验,采用RBF神经网络近似值函数,分别验证引入资格迹机制和改进softmax策略的作用. 其他条件一致,对比Sarsa算法和Sarsa($\lambda$)算法完成任务所需要的步数,每种算法随机实验5次,对步数取平均值,结果如图3所示.

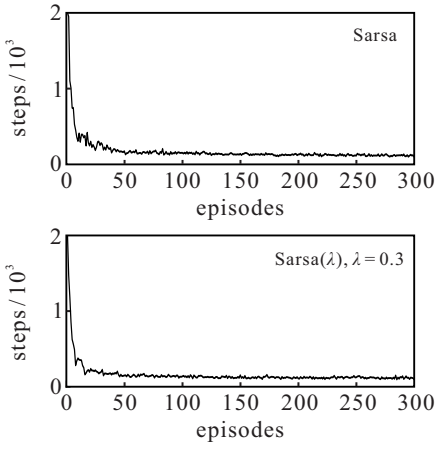


图 3　引入资格迹机制的学习效果对比

由图3可见,RBF神经网络与单步Sarsa算法结合学习效果良好,在大约50个episodes后,步数逐渐

稳定在130左右. 与多步算法Sarsa($\lambda$)相结合,学习效果受到资格迹参数$\lambda$取值的影响. 当$\lambda = 0.3$时,获得与单步算法相近的学习效果,在约40个episodes后步数稳定在120左右,收敛时间相对提前.

为了更清晰地对比学习效果,对单步Sarsa和$\lambda = 0.3$时的多步Sarsa两种算法的原始数据(steps < 500之后的episodes)进行统计分析,表1为部分统计结果. 数据显示,Sarsa($\lambda$)算法比单步Sarsa算法具有更少的平均收敛步数,且方差更小,表明多步算法的学习步数更加稳定,整体控制效果相对更好.

**表 1 不同算法的原始学习数据**

| 算法 | max-step | min-step | average-step | 方差$\sigma$ |
|---|---|---|---|---|
| Sarsa | 453 | 97 | 152 | 55.24 |
| Sarsa($\lambda$) | 488 | 90 | 136 | 49.98 |

固定资格迹参数$\lambda = 0.2$,对比softmax动作选择策略在固定温度参数和衰减温度参数下完成任务需要的步数. 每种情况随机实验5次,取平均步数,结果如图4所示.

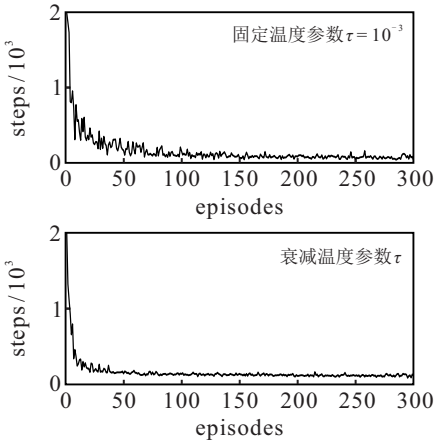


**图 4 改进softmax策略的学习效果对比**

由图4可以看出,温度参数随时间衰减的softmax动作选择策略明显比温度参数固定的学习效果更好. 参数衰减策略下,小车在大约50个episodes的学习之后基本稳定在120步左右. 而在参数固定的情况下,学习效果受参数取值大小影响,当$\tau = 10^{-3}$时学习效果相对较好,大概在100个episodes之后趋于稳定. 值得注意的是,$\tau = 10^{-3}$的稳定步数整体略低于温度参数衰减的策略,但整个学习过程中,步数波动相对较大,尤其在前100个episodes表现明显.

**实验2** 进行对比实验,比较本文算法与一般BP神经网络值函数近似算法及DQN算法[1]的控制效果. 其中,DQN算法是深度强化学习的代表性算法之一,其利用卷积神经网络近似动作值函数,并采用经验回放(experience replay)对样本进行存储、采样,打破了数据之间的强关联性,同时设置单独的网络更新目标提高训练的稳定性. 图5为3种算法的学习结果对比曲线.

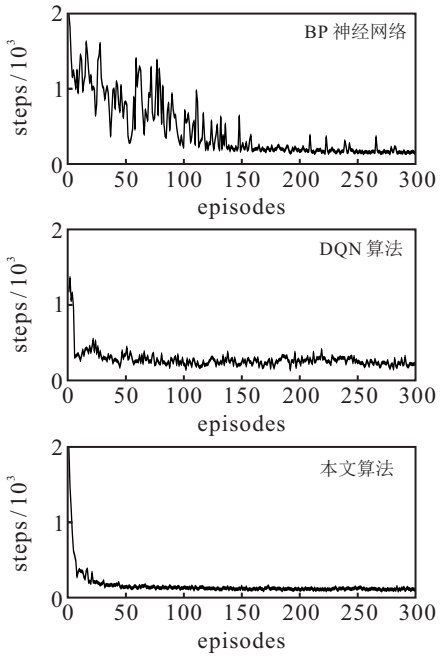


**图 5 不同算法的学习步数比较**

由图5可以看出,所提出算法步数下降最快,尤其在前50个episodes,迅速达到稳定状态,并且收敛之后完成步数一直保持平稳,控制效果相对最好. 基于普通BP神经网络的近似方法,在学习初期步数波动幅度最大,150个episodes之后逐渐开始收敛,偶尔有较大的步数出现,但稳定步数大于本文算法. DQN算法整体趋势与本文算法接近,前期步数迅速下降,之后在小范围内波动,难以维持较好的控制效果. 这是因为DQN算法使用了深层网络,训练难度增加,网络的稳定性缺乏保证.

## 4 参数影响

资格迹参数代表从当前状态向前利用并更新状态信息的程度. 当$\lambda = 0$时,多步Sarsa($\lambda$)算法退化成单步Sarsa算法;当$\lambda = 1$时,Sarsa($\lambda$)算法相当于回合更新的蒙特卡洛算法. 对$\lambda$在$[0, 1]$之间取不同值进行实验,每个取值随机测试5次,求每个episode的平均步数,部分结果如图6所示.

由图6可以看出: 当$\lambda = 0.9$时,任务完成步数的波动很大,算法无法收敛;随着取值不断减小,整体学习效果越来越好. 对不同$\lambda$值的所有episodes再次求平均完成步数,结果如图7所示.

当$0 < \lambda < 0.5$时,多步算法的学习效果优于单步算法,其中在$\lambda = 0.3$时,平均步数为155,达到所有取值中的最小. 当$\lambda > 0.5$时,随着取值的逐渐增大,

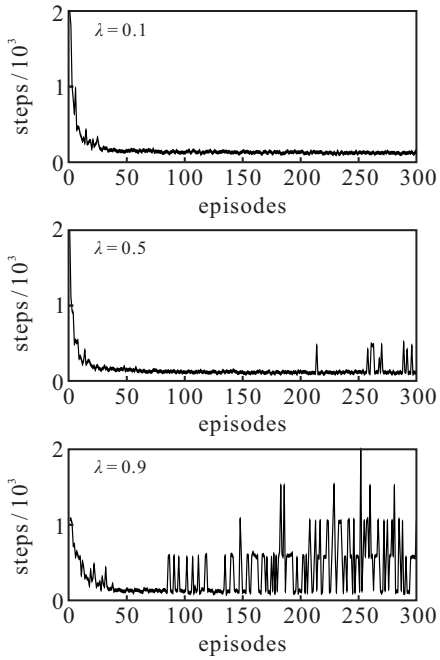


图 6　不同λ值的任务完成步数

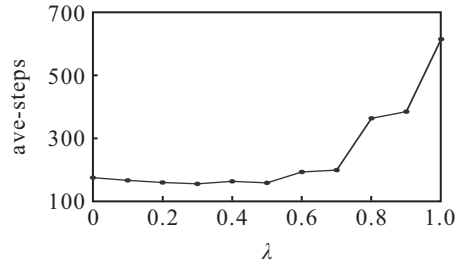


图 7　不同λ值的平均任务完成步数

Sarsa(λ) 算法的平均步数迅速上升,学习效果变差. 其原因在于λ的值越大越接近于回合更新,其方差增大,直接表现为学习步数振动较大.

Softmax动作选择策略中的温度参数$\tau$直接影响探索与利用 (exploration and exploitation) 的平衡关系. 当$\tau$值较大时,鼓励智能体探索潜在动作,反之则鼓励智能体利用已知经验. 极限情况 $\tau = 0$相当于 $\arg\max$函数,每次选择奖励值最高的动作,即贪婪策略,缺少对整个状态空间的探索;当$\tau = +\infty$时,智能体总是随机选择一个动作,无法利用好的经验,表现不够智能. 对$\tau$在$[10^{-5}, 10^{2}]$中取不同值测试,并对每个取值随机实验5次取平均步数,部分结果如图8所示. 整体来看,随着$\tau$值的减小,平均任务完成步数随之减少,学习效果逐渐变好. 当$\tau = 10$时,完成步数多在1 000步以上;当$\tau = 0.1$时,完成步数减少到500以内的episodes占大部分,但步数仍存在较大波动. 继续缩小100倍,$\tau = 10^{-3}$学习效果相对最好,完成步数在50个episodes之后能够长期保持稳定.

对不同$\tau$值求所有episodes的任务完成平均步数,结果如图9所示. 其中当$\tau < 0.1$时,平均步数变化相对平缓,并且在$\tau = 10^{-3}$时平均步数最小,约150步. 当$\tau > 0.1$时,由于智能体更倾向于随机选择动作,平均步数急剧增加.

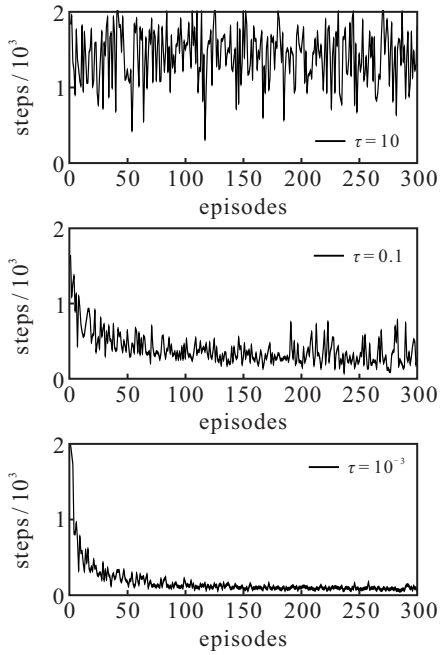


图 8　不同$\tau$值的任务完成步数

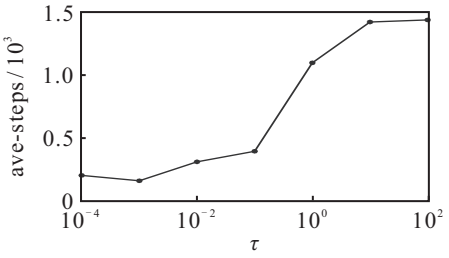


图 9　不同$\tau$值的平均任务完成步数

## 5　结　论

本文提出了一种基于神经网络的多步强化学习算法,将RBF神经网络与Sarsa(λ)算法结合,实现了无模型情况下对连续非线性系统的有效控制. 算法通过引入资格迹机制利用历史信息,提高了学习效率,同时采用概率优化动作选择策略,达到了平衡探索和利用关系的目的,最终获得了比单步算法更好的学习效果. MountainCar任务的实验结果验证了该算法的正确性和有效性,同时也表明了非线性值函数近似在多步强化学习算法中同样可以得到良好的控制性能. 之后将继续改进,进一步提高算法的鲁棒性和泛化性,将其应用到其他控制系统.

**参考文献**(References)

[1] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[2] Silver D, Huang A, Maddison C J, et al. Mastering the game of go with deep neural networks and tree search[J].

Nature, 2016, 529(7587): 484-489.

[3] Silver D, Schrittwieser J, Simonyan K, et al. Mastering the game of go without human knowledge[J]. Nature, 2017, 550(7676): 354-359.

[4] Buoniu L, de Bruin T, Toli D, et al. Reinforcement learning for control: Performance, stability, and deep approximators[J]. Annual Reviews in Control, 2018, 46: 8-28.

[5] Kiumarsi B, Vamvoudakis K G, Modares H, et al. Optimal and autonomous control using reinforcement learning: A survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2018, 29(6): 2042-2062.

[6] Sun T, Shen H, Chen T Y, et al. Adaptive temporal difference learning with linear function approximation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 9645(99): 1-16.

[7] Joseph A G, Bhatnagar S. An online prediction algorithm for reinforcement learning with linear function approximation using cross entropy method[J]. Machine Learning, 2018, 107(8/9/10): 1385-1429.

[8] 刘全, 章鹏, 钟珊, 等. 连续空间中的一种动作加权行动者评论家算法[J]. 计算机学报, 2017, 40(6): 1252-1264.
(Liu Q, Zhang P, Zhong S, et al. An improved actor-critic algorithm in continuous spaces with action weighting[J]. Chinese Journal of Computers, 2017, 40(6): 1252-1264.)

[9] 张耀中, 胡小方, 周跃, 等. 基于多层忆阻脉冲神经网络的强化学习及应用[J]. 自动化学报, 2019, 45(8): 1536-1547.
(Zhang Y Z, Hu X F, Zhou Y, et al. A novel reinforcement learning algorithm based on multilayer memristive spiking neural network with applications[J]. Acta Automatica Sinica, 2019, 45(8): 1536-1547.)

[10] 唐昊, 杨羊, 戴飞, 等. 基于RBF-Q学习的多品种CSPS系统前视距离控制[J]. 控制与决策, 2019, 34(7): 1456-1462.
(Tang H, Yang Y, Dai F, et al. Look-ahead control of multi-type products CSPS system based on RBF-Q learning[J]. Control and Decision, 2019, 34(7): 1456-1462.)

[11] Zeng J J, Ju R S, Qin L, et al. Navigation in unknown dynamic environments based on deep reinforcement learning[J]. Sensors, 2019, 19(18): 3837.

[12] Zhang H, Li S Y, Zheng Y. Q-learning-based model predictive control for nonlinear continuous-time systems[J]. Industrial & Engineering Chemistry Research, 2020, 59(40): 17987-17999.

[13] Tang D F, Chen L, Tian Z F, et al. Modified value-function-approximation for synchronous policy iteration with single-critic configuration for nonlinear optimal control[J]. International Journal of Control, 2021, 94(5): 1321-1333.

[14] 傅启明, 刘全, 王辉, 等. 一种基于线性函数逼近的离策略$Q(\lambda)$算法[J]. 计算机学报, 2014, 37(3): 677-686.
(Fu Q M, Liu Q, Wang H, et al. A novel off policy $Q(\lambda)$ algorithm based on linear function approximation[J]. Chinese Journal of Computers, 2014, 37(3): 677-686.)

[15] van Seijen H. Effective multi-step temporal-difference learning for non-linear function approximation[J/OL]. 2016, arXiv: 1608.05151.

[16] 何斌, 刘全, 张琳琳, 等. 一种加速时间差分算法收敛的方法[J]. 自动化学报, 2021, 47(7): 1679-1688.
(He B, Liu Q, Zhang L L, et al. A method of accelerating the convergence of temporal difference learning[J]. Acta Automatica Sinica, 2021, 47(7): 1679-1688.)

[17] Abpeykar S, Ghatee M. An ensemble of RBF neural networks in decision tree structure with knowledge transferring to accelerate multi-classification[J]. Neural Computing and Applications, 2019, 31(11): 7131-7151.

[18] Robnik-Sikonja M. Data generators for learning systems based on RBF networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(5): 926-938.

[19] Yang H J, Liu J K. An adaptive RBF neural network control method for a class of nonlinear systems[J]. IEEE/CAA Journal of Automatica Sinica, 2018, 5(2): 457-462.

[20] 张伟, 黄卫民. 基于SAPSO算法的RBF神经网络设计[J]. 控制与决策, 2021, 36(9): 2305-2312.
(Zhang W, Huang W M. Design of RBF neural network based on SAPSO algorithm[J]. Control and Decision, 2021, 36(9): 2305-2312.)

[21] Singh S, Jaakkola T, Littman M L, et al. Convergence results for single-step on-policy reinforcement-learning algorithms[J]. Machine Learning, 2000, 38(3): 287-308.

## 作者简介

司彦娜(1990–), 女, 博士生, 从事强化学习、智能控制的研究, E-mail: siyanna2722@163.com;

普杰信(1959–), 男, 教授, 博士生导师, 从事模式识别、智能控制等研究, E-mail: pjx@haust.edu.cn;

于晓升(1984–), 男, 讲师, 博士, 从事无线传感器网络、人工智能等研究, E-mail: yuxiaosheng@mail.neu.edu.cn;

司鹏举(1987–), 男, 讲师, 博士, 从事车联网、无线传感器网络等研究, E-mail: sipengju@haust.edu.cn;

孙力帆(1982–), 男, 副教授, 博士, 从事信息融合、目标跟踪等研究, E-mail: lifan.sun@gmail.com.

(责任编辑: 郑晓蕾)