DOI: 10.7641/CTA.2015.40367

采用资格迹的神经网络学习控制算法

刘智斌[1†], 曾晓勤[2], 徐 彦[3], 禹继国[1]

(1. 曲阜师范大学 信息科学与工程学院, 山东 日照 276826; 2. 河海大学 计算机与信息学院, 江苏 南京 210098;
3. 南京农业大学 信息科技学院, 江苏 南京 210095)

摘要: 强化学习是解决自适应问题的重要方法, 被广泛地应用于连续状态下的学习控制, 然而存在效率不高和收敛速度较慢的问题. 在运用反向传播(back propagation, BP)神经网络基础上, 结合资格迹方法提出一种算法, 实现了强化学习过程的多步更新. 解决了输出层的局部梯度向隐层节点的反向传播问题, 从而实现了神经网络隐层权值的快速更新, 并提供一个算法描述. 提出了一种改进的残差法, 在神经网络的训练过程中将各层权值进行线性优化加权, 既获得了梯度下降法的学习速度又获得了残差梯度法的收敛性能, 将其应用于神经网络隐层的权值更新, 改善了值函数的收敛性能. 通过一个倒立摆平衡系统仿真实验, 对算法进行了验证和分析. 结果显示, 经过较短时间的学习, 本方法能成功地控制倒立摆, 显著提高了学习效率.

关键词: 强化学习; 神经网络; 资格迹; 倒立摆; 梯度下降

中图分类号: TP301 **文献标识码**: A

Learning to control by neural networks using eligibility traces

LIU Zhi-bin[1†], ZENG Xiao-qin[2], XU Yan[3], YU Ji-guo[1]

(1. School of Information Science and Engineering, Qufu Normal University, Rizhao Shandong 276826, China;
2. College of Computer and Information, Hohai University, Nanjing Jiangsu 210098, China;
3. College of Information Science and Technology, Nanjing Agricultural University, Nanjing Jiangsu 210095, China)

Abstract: Reinforcement learning is an important approach to solve the adaptive learning control problems in continuous state space. However, it is bedeviled by its low learning efficiency and low convergence rate. In order to eliminate those deficiencies, based on back propagation (BP) neural networks and eligibility traces, we propose a learning algorithm with a complete description to achieve the multi-step updates in the process of reinforced learning to realize the counter propagation of the local gradient from output layer nodes to hidden layer nodes; thus, rapidly adjusting the weights of hidden layers. During the training processes of neural networks, a modified residual method is employed to optimize the weights in each layer by linear combination, achieving the rapid learning rate of the direct gradient method as well as the desired convergence properties of the residual gradient method. Applying this method to update the weights of hidden layers in a neural network, we improve the convergence properties of value functions. A cart-pole system is adopted for testing the application results of the above mentioned algorithms. Simulation results show that all our algorithms can successfully achieve the control for the cart-pole balancing system and improve the learning efficiency significantly.

Key words: reinforcement learning; neural networks; eligibility traces; cart-pole system; gradient descent

1 引言(Introduction)

基于表格的强化学习[1–4]方法, 在未知环境中进行学习, 表现出了极好的自适应能力. 然而, 这种方法只能解决状态空间和行为空间较小的问题. 随着状态空间和行为空间规模的增大, 问题空间往往呈指数增加, “维数灾难”问题就显得尤为突出. 采用表格法解决大规模问题, 在离散空间中从状态到行为的映射需要精确对应, 这样往往占用大量的内存空间. 若将这一对应关系用连续函数代替, 用函数值代替表格中的值, 则能够取得较好的效果. 从状态空间到函数值的映射, 其建立方法分为线性参数拟合方法和非线性参数拟合方法[5]. 由于进行理论分析相对简单, 线性参数拟合方法常常应用于强化学习问题中. 而非线性参数方法, 在数据拟合方面也得到了广泛的应用. 非线性参数拟合方法比较典型的工具是神经网络. 神经网络具有较强的自适应能力和泛化性能[6], 将神经网络与强

收稿日期: 2014−04−27; 录用日期: 2015−04−10.

†通信作者. E-mail: lzbxian@163.com; Tel.: +86 633-3981026.

国家自然科学基金项目(61403205, 61373027, 60117089), 曲阜师范大学实验室开放基金项目(sk201415)资助.
Supported by National Natural Science Foundation of China (61403205, 61373027, 60117089) and Laboratory Open Foundation of Qufu Normal University (sk201415).

化学习相结合, 用神经网络代替表格, 能够取得较好的效果. 针对基于表格的强化学习, Sutton提出了瞬时差分TD(λ)方法[7], 为每个访问状态设立一个资格迹, 每执行一步更新, 这步更新也向后传递若干步, 使学习速度大大加快. 针对TD(λ)方法, Dayan等人[8]证明了它的收敛性. Sutton等[7,9]提出了在连续状态空间下的瞬时差分法, 并提出基于直接梯度法的资格迹方法.

将反向传播神经网络(back propagation neural networks, BPNN)运用于强化学习在国内外很多文献[10–13]都有过介绍, 但这些方法基本上采用单步权值更新. 在学习过程中引入资格迹, 能大大提高神经网络的训练效率, 但是这就使得神经网络的训练过程, 特别是神经网络隐层权值的更新将变得更加复杂. 另外, 强化学习采用“自举”方法训练神经网络, 这与常规的神经网络训练方法有所不同. 本文提出一种方法, 实现了BP神经网络应用于强化学习的训练过程, 运用资格迹, 把局部梯度从输出层传递到隐层, 实现了隐层权值的更新, 并提供了完整的学习算法描述.

基于拟合器的强化学习方法在学习过程中更新其权值, 常用的方法有直接梯度法和残差梯度法[12]. 由于直接梯度法类似于监督学习中的最速下降法, 这种方法学习速度较快, 但是往往收敛性能不理想. 而残差梯度法能够保证较好的收敛性, 但是它的收敛速度比较缓慢. Baird[12]提出了一种残差法, 这种方法既能保证使用残差梯度法的收敛性, 又确保使用直接梯度法的收敛速度, 取得了良好的性能. 然而, Baird只给出了输出层权值更新的计算方法, 没有涉及隐层的情形. 本文对Baird的方法做了进一步推广, 提出一种改进的残差法, 不仅对神经网络输出层进行权值更新, 而且对隐层进行了优化权值更新, 保证了BP神经网络在强化学习过程中的良好性能.

2 基于神经网络的强化学习过程(Reinforcement learning process based on NN)

强化学习的学习过程是: 学习Agent在与环境的交互中, 不断获得评价性的反馈信息作为回报, 再将回报值做加权累加, Agent在行为选择过程中, 选择能够取得最大积累回报的行为作为其最优行为.

Agent在状态$s \in S$下的可执行行为记作$a \in A$, 它从行为集合A中选择使$Q^{\pi}(s,a)$最大的行为作为其最优行为, $Q^{\pi}(s,a)$的定义如下:

$$Q^{\pi}(s,a) = \mathrm{E}\{r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots | \\ s_t = s, a_t = s, \pi\}, \tag{1}$$

其中$\gamma \in (0,1)$.

在问题模型未知的情形下, Watins[14]提出了Q–学习算法, 表示为

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - \\ Q(s,a)). \tag{2}$$

Agent在每次迭代中更新$Q(s,a)$值, 在经过多次迭代后$Q(s,a)$值收敛, Watins[14]证明了它的收敛性.

在$Q(s,a)$值定义的基础上, V值定义如下:

$$V(s) = \max_{a \in A(S)} Q(s,a). \tag{3}$$

在状态s下, 求得当前最优策略为π^*, 定义如下:

$$\pi^*(s) = \arg\max_{a} Q(s,a). \tag{4}$$

采用BP神经网络作为强化学习值函数拟合器, 如图1所示, 右侧神经网络的输入端接收状态信息. 系统依据神经网络的输出值V和环境反馈的报酬信息r, 利用TD算法训练神经网络, Agent依据V值函数选取行为a.

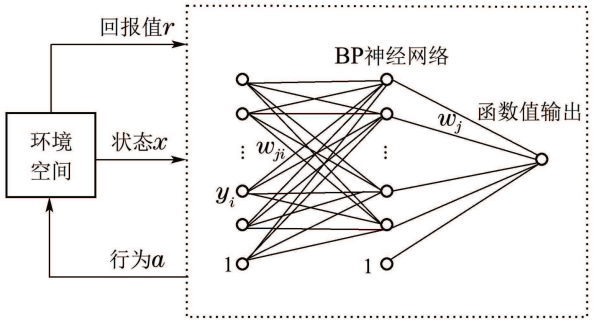

图 1 基于BP神经网络的强化学习模型

Fig. 1 Model of reinforcement learning based on BPNN

Agent从一个状态$\boldsymbol{x}_t$进入到另一状态$\boldsymbol{x}_{t+1}$, 获得报酬值r_t, 在状态$\boldsymbol{x}_t$下的函数值为$V(\boldsymbol{x}_t)$, $V(\boldsymbol{x}_t)$用拟合函数来表示. 对于输入状态$\boldsymbol{x}_t$, 它的目标输出值为$r_t + \gamma V(\boldsymbol{x}_{t+1})$. 在更新过程中相应拟合函数的权值按直接梯度法更新应为

$$\Delta w = \alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t)) \frac{\partial V(\boldsymbol{x}_t)}{\partial \boldsymbol{w}}, \tag{5}$$

其中$\boldsymbol{x} = [x_1 \ x_2 \ \cdots \ x_i \ \cdots \ x_m]^{\mathrm{T}}$为状态向量.

定义一个神经网络结构, 网络由输入层、隐含层和输出层3层组成, 其中: 输入层节点个数为$m+1$, 隐层节点的个数为$n+1$, 输出层节点个数为1. 其中: 向量$\boldsymbol{y} = [y_1 \ y_2 \ \cdots \ y_i \ \cdots \ x_m]^{\mathrm{T}}$为神经网络的输入向量; 状态向量$\boldsymbol{x}$中的分量依次赋值给神经网络输入向量$\boldsymbol{y}$中的对应分量; $y_i \leftarrow x_i$, 固定输入$y_0 \leftarrow 1$.

定义隐层节点到输出层节点的连接权值为

$$\boldsymbol{w}^2 = [w_0 \ w_1 \ w_2 \ \cdots \ w_n]. \tag{6}$$

输入层到隐层的连接权值为

$$W^1 = \begin{bmatrix} w_{10} & w_{11} & w_{12} & \cdots & w_{1m} \\ w_{20} & w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \vdots & & \vdots \\ w_{n0} & w_{n1} & w_{n2} & \cdots & w_{nm} \end{bmatrix}. \tag{7}$$

由神经元节点p连接到神经元节点q的突触权值的

修正值为

$$\Delta w_{\rm qp} = \alpha\delta_{\rm q} y_{\rm p}, \tag{8}$$

其中: $\delta_{\rm q}$为神经元的局部梯度, $y_{\rm p}$为输入值.

在一个3层神经网络中, 本问题的输出神经元只有一个, 其局部梯度为

$$\delta = (r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t))\varphi'(v), \tag{9}$$

其中:

$$v = \sum_{j=0}^{n} w_j y_j, \tag{10}$$

$\varphi(\cdot)$为输出节点的激活函数, $\varphi'(v)$为$\varphi(\cdot)$在v处的导数.

神经元j作为隐层节点, 其局部梯度为

$$\delta_j = \varphi_j'(v_j)\delta w_j, \tag{11}$$

其中

$$v_j = \sum_{i=0}^{m} w_{ji} y_i, \tag{12}$$

其中i为输入层节点索引.

3　引入资格迹的直接梯度法(Direct gradient method using eligibility traces)

为了加快训练速度, 引入资格迹[1,7]方法. 资格迹方法能将一步误差更新向后传播若干步, 表现在神经网络上, 就是累积更新权值. 权值更新公式为

$$\Delta \boldsymbol{w}_t = \alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t)) \cdot \sum_{k=0}^{t} \lambda^{t-k} \frac{\partial V(\boldsymbol{x}_k)}{\partial \boldsymbol{w}}. \tag{13}$$

令$\boldsymbol{e}_t = \sum_{k=0}^{t} \lambda^{t-k} \frac{\partial V(\boldsymbol{x}_k)}{\partial \boldsymbol{w}}$, 通过迭代实现每一步的资格迹, 其公式推导如下:

$$\begin{aligned} \boldsymbol{e}_{t+1} &= \sum_{k=0}^{t+1} \lambda^{t+1-k} \frac{\partial V(\boldsymbol{x}_k)}{\partial \boldsymbol{w}} = \\ &\frac{\partial V(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{w}} + \sum_{k=0}^{t} \lambda^{t+1-k} \frac{\partial V(\boldsymbol{x}_k)}{\partial \boldsymbol{w}} = \\ &\frac{\partial V(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{w}} + \lambda \boldsymbol{e}_t. \end{aligned} \tag{14}$$

通过式(14)求得的每步资格迹与最后一步状态变换误差值的乘积, 就是神经网络的连接突触权值更新值.

隐层到输出层的任意连接突触更新Δw_j为

$$\Delta w_j = \alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t)) \cdot \sum_{k=0}^{t} \lambda^{t-k} \varphi_j'(v_j^k) y_j^k. \tag{15}$$

为了求得输入层到隐层的连接突触权值, 由式(15), 在时间步t, 获得误差值为$r_t+\gamma V(\boldsymbol{x}_{t+1})-V(\boldsymbol{x}_t)$, 传播到时间步$k$的误差值为

$$E^k = (r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t))\lambda^{t-k}. \tag{16}$$

在时间步k, 输出神经元的局部梯度为

$$\delta^k = (r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t))\lambda^{t-k}\varphi'^k(v^k). \tag{17}$$

对于神经元j作为隐层节点, 在时间步k, 其局部梯度为

$$\delta_j^k = \varphi'^k_j(v_j^k)\delta^k w_j^k. \tag{18}$$

到时间步k, 由神经元节点i连接到神经元节点j的突触权值的修正值为

$$\Delta w_{ji}^k = \alpha\delta_j^k y_i^k = \alpha\varphi'^k_j(v_j^k)\delta^k w_j^k y_i^k. \tag{19}$$

在时间步t, 引入资格迹后的由神经元节点i连接到神经元节点j的突触权值的修正值为

$$\begin{aligned} \Delta w_{ji} &= \sum_{k=0}^{t} \Delta w_{ji}^k = \\ &\alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t)) \cdot \\ &\sum_{k=0}^{t} \varphi'^k_j(v_j^k)\lambda^{t-k}\varphi'^k(v^k) w_j^k y_i^k. \end{aligned} \tag{20}$$

在以上的讨论中, 神经网络的隐层到输出层突触权值的更新依照梯度下降法进行调整, 神经网络输入层到输出层突触权值的更新依赖于输出层节点局部梯度到隐层节点局部梯度的反传.

4　残差梯度法(Residual gradient method)

采用BP神经网络拟合值函数, 若采取直接梯度法, 函数的收敛性能有时不能得到保证[12, 15], 而残差梯度法能保证在迭代的过程中函数达到收敛. 相比而言, 直接梯度法每一步更新需要的计算量相对较少, 但是它所利用的信息量也较少, 学习速度较快, 但有时不能保证收敛; 而残差梯度法能保证收敛, 但是它的收敛速度往往很慢.

Agent从一个状态$\boldsymbol{x}_t$转移到下一状态$\boldsymbol{x}_{t+1}$, 获得报酬值r_t, 在状态$\boldsymbol{x}_t$下的函数值为$V(\boldsymbol{x}_t)$, $V(\boldsymbol{x}_t)$用拟合函数来表示. 对于输入状态$\boldsymbol{x}_t$, 它的目标输出值为$r_t + \gamma V(\boldsymbol{x}_{t+1})$. 其误差信息$E$的计算公式为

$$E = \frac{1}{2}(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t))^2. \tag{21}$$

使误差E趋于最小, 采用梯度下降法, 求得每次迭代神经网络权值的变化量$\Delta\boldsymbol{w}$. 将$V(\boldsymbol{x}_t)$和$V(\boldsymbol{x}_{t+1})$都视为变化量, 由式(21)求得拟合函数的权值按残差梯度法更新为

$$\Delta \boldsymbol{w} = \alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t)) \cdot \left(\frac{\partial V(\boldsymbol{x}_t)}{\partial \boldsymbol{w}} - \gamma \frac{\partial V(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{w}}\right), \tag{22}$$

其中α为学习速度. 采用式(22)对神经网络进行权值迭代更新, 能保证值函数收敛[12].

由式(22)变形得

$$\Delta \boldsymbol{w} = \alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t))\frac{\partial V_t(\boldsymbol{x}_t)}{\partial \boldsymbol{w}} - \gamma\alpha(r_t+\gamma V(\boldsymbol{x}_{t+1})-V_t(\boldsymbol{x}_t))\frac{\partial V(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{w}}. \quad (23)$$

式(23)中: $\alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t))\dfrac{\partial V(\boldsymbol{x}_t)}{\partial \boldsymbol{w}}$ 项的求值跟第 2 节中的直接梯度法求法相同. $\gamma\alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t))\dfrac{\partial V(\boldsymbol{x}_{t+1})}{\partial \boldsymbol{w}}$项的求值跟第2节中的直接梯度法求法基本类似, 只是输入值为目标状态. 引入资格迹后, 求得相应的拟合函数的权值按残差梯度法更新为

$$\Delta \boldsymbol{w}_t = \alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t)) \cdot \sum_{k=0}^{t} \lambda^{t-k}(\frac{\partial V(\boldsymbol{x}_k)}{\partial \boldsymbol{w}} - \gamma\frac{\partial V(\boldsymbol{x}_{k+1})}{\partial \boldsymbol{w}}). \quad (24)$$

由式(24)变形得

$$\Delta \boldsymbol{w}_t = \alpha(r_t + \gamma V(\boldsymbol{x}_{t+1}) - V(\boldsymbol{x}_t)) \sum_{k=0}^{t} \lambda^{t-k}\frac{\partial V(\boldsymbol{x}_k)}{\partial \boldsymbol{w}} - \gamma\alpha(r_t + \gamma V(\boldsymbol{x}_t) - V(\boldsymbol{x}_t)) \sum_{k=0}^{t} \lambda^{t-k}\frac{\partial V(\boldsymbol{x}_{k+1})}{\partial \boldsymbol{w}}. \quad (25)$$

式(25)中, 等式右侧第1项的求值跟第3节中引入资格迹的直接梯度法求法相同. 等式右侧第2项的求值跟第3节中的直接梯度法求法基本类似, 只是输入值为目标状态. 基于资格迹的残差梯度法的算法描述只需在第3节算法的基础上进行适当修改即可.

5 改进的残差法(Modified residual method)

直接梯度法和残差梯度法各有优缺点. 采用直接梯度法会保证较快的学习效率, 但无法保证权值收敛. 通过验证, 在有的情况下直接梯度法会出现发散的情形. 文献[12]讨论了残差梯度法的收敛性, 论证了残差梯度法能保证在学习过程中权值收敛, 但学习速度较直接梯度法慢很多. 进而提出了一种残差法, 这种方法既能保证达到残差梯度法较好的收敛性, 又能获得直接梯度法的收敛速度, 取得了良好的总体性能.

本文在Baird的残差法的基础上, 提出一种改进的残差法, 将资格迹引入到权值更新, 同时将权值更新扩展到神经网络的隐层. 引入资格迹的直接梯度法, 当Agent执行一步状态转换, 其误差更新能够向后传播若干步. 利用第3节的方法, 将具有3层节点的BP神经网络的连接突触权值更新用一个$(m+2)n+1$维向量$\Delta \boldsymbol{w}_{\rm d}$表示为

$$\Delta \boldsymbol{w}_{\rm d} = [\Delta w_0\ \ \Delta w_1\ \ \cdots\ \ \Delta w_n\ \ \Delta w_{10}\ \ \Delta w_{20}\ \ \cdots\ \ \Delta w_{n0}\ \ \Delta w_{11}\ \ \cdots\ \ \Delta w_{ji}\ \ \cdots\ \ \Delta w_{nm}]_{\rm d}. \quad (26)$$

式(26)中的前$n+1$项是隐层到输出层的连接突触权值更新, 后$(m+1)n$项是输入层到隐层的连接突触权值更新.

采用第4节的方法, 用基于资格迹的残差梯度法来更新神经网络的连接突触权值, 将具有3层节点的BP神经网络的连接突触权值更新用一个$(m+2)n+1$维向量$\Delta \boldsymbol{w}_{\rm rg}$表示为

$$\Delta \boldsymbol{w}_{\rm rg} = [\Delta w_0\ \ \Delta w_1\ \ \cdots\ \ \Delta w_n\ \ \Delta w_{10}\ \ \Delta w_{20}\ \ \cdots\ \ \Delta w_{n0}\ \ \Delta w_{11}\ \ \cdots\ \ \Delta w_{ji}\ \ \cdots\ \ \Delta w_{nm}]_{\rm rg}. \quad (27)$$

1) 若$\Delta \boldsymbol{w}_{\rm d} \cdot \Delta \boldsymbol{w}_{\rm rg} > 0$, 则两向量之间的夹角为锐角, $\Delta W_{\rm d}$减小带来残差梯度更新量$\Delta \boldsymbol{w}_{\rm rg}$减小, 使拟合函数收敛.

2) 若$\Delta \boldsymbol{w}_{\rm d} \cdot \Delta \boldsymbol{w}_{\rm rg} < 0$, 则两向量之间的夹角为钝角, $\Delta \boldsymbol{w}_{\rm d}$减小带来残差梯度更新量$\Delta \boldsymbol{w}_{\rm rg}$增加, 使拟合函数发散.

为了避免发散, 又能够使神经网络的训练过程较为快速, 引入残差更新向量$\Delta \boldsymbol{w}_{\rm r}$, 其值为向量$\Delta \boldsymbol{w}_{\rm d}$和$\Delta \boldsymbol{w}_{\rm rg}$的加权平均值, 定义为

$$\Delta \boldsymbol{w}_{\rm r} = (1-\phi)\Delta \boldsymbol{w}_{\rm d} + \phi\Delta \boldsymbol{w}_{\rm rg}, \quad (28)$$

其中$\phi \in [0,1]$. ϕ的选取, 应使$\Delta \boldsymbol{w}_{\rm r}$与$\Delta \boldsymbol{w}_{\rm rg}$的夹角为锐角, 同时让$\Delta \boldsymbol{w}_{\rm r}$尽量与$\Delta \boldsymbol{w}_{\rm d}$离得近一些. 以下求使向量$\Delta \boldsymbol{w}_{\rm r}$与向量$\Delta \boldsymbol{w}_{\rm rg}$垂直的$\phi_\perp$值.

$$\Delta \boldsymbol{w}_{\rm r} \cdot \Delta \boldsymbol{w}_{\rm rg} = 0. \quad (29)$$

满足式(29)的向量$\Delta \boldsymbol{w}_{\rm r}$与向量$\Delta \boldsymbol{w}_{\rm rg}$垂直,

求解式(29), 得到$\phi_\perp$值为

$$\phi_\perp = \frac{\Delta \boldsymbol{w}_{\rm d} \cdot \Delta \boldsymbol{w}_{\rm rg}}{\Delta \boldsymbol{w}_{\rm d} \cdot \Delta \boldsymbol{w}_{\rm rg} - \Delta \boldsymbol{w}_{\rm rg} \cdot \Delta \boldsymbol{w}_{\rm rg}}. \quad (30)$$

ϕ的选取只需在$\phi_\perp$值上增加一个较小的正值μ, 使之略偏向向量$\Delta \boldsymbol{w}_{\rm rg}$一点.

$$\phi = \phi_\perp + \mu. \quad (31)$$

3) 若$\Delta \boldsymbol{w}_{\rm d} \cdot \Delta \boldsymbol{w}_{\rm rg} = 0$, 则两向量之间的夹角为直角, 这样有$\phi_\perp = 0$. ϕ的选取为

$$\phi = \phi_\perp + \mu = \mu. \quad (32)$$

残差梯度法能保证在迭代过程中权值收敛, 通过这种方法训练神经网络的各层权值, 其更新不会引起函数值发散, 是安全的; 而用直接梯度法训练神经网络的各层权值, 会有发散的情况. 改进的残差法将神经网络的各层权值都加以考虑, 使得权值更新向量$\Delta \boldsymbol{w}_{\rm r}$不会引起用残差梯度法得到的权值更新向量$\Delta \boldsymbol{w}_{\rm rg}$向其相反的方向变化, 从而保证收敛.

6 倒立摆仿真实验(Cart-pole balancing system simulation experiment)

倒立摆问题是一个典型的非线性控制问题[16–17],

以下采用未知环境下的倒立摆仿真实验作为一个算例来验证本文算法的有效性.

6.1 实验问题描述(Experiment description)

如图2所示, 一个小车可以在一个水平轨道上自由运动, 小车上安装了一个钢性的自由摆杆, 摆杆处在不稳定状态下. 小车在可控力F的作用下左右运动, 小车运动的轨道范围是$[-2.4, 2.4]$ m. 本问题是: 在力的作用下小车在导轨上运动, 学习系统力图让摆杆保持足够长时间的竖直状态而不倒掉. 当小车运动超出轨道范围$[-2.4, 2.4]$ m, 则本轮实验失败; 当小车的摆杆与垂直方向的夹角θ超过的某一数值也认定为实验失败. 将倒立摆的水平位移x、水平运动速度$\dot{x}$、夹角θ和θ对时间的导数$\dot{\theta}$作为神经网络的输入值. 当倒立摆在水平导轨上超出轨道范围$[-2.4, 2.4]$ m或θ夹角超出范围$[-12°, 12°]$都会得到奖惩值-1, 在其他状态范围, 得到的奖惩值为0.

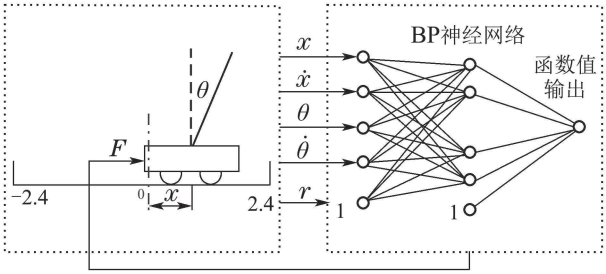

图 2 基于强化学习的倒立摆平衡控制模型

Fig. 2 Model of cart-pole balancing system based on reinforcement learning

倒立摆系统运动的参数方程描述为

$$\ddot{\theta} = (\cos\theta[\frac{-F - ml\dot{\theta}^2\sin\theta + \mu_{\rm c}\mathrm{sgn}\,\dot{x}}{m_{\rm c}+m}] + g\sin\theta - \frac{\mu_{\rm p}\dot{\theta}}{ml})/(l[\frac{4}{3} - \frac{m\cos^2\theta}{m_{\rm c}+m}]), \quad (33)$$

$$\ddot{x} = \frac{F + ml[\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta] - \mu_{\rm c}\mathrm{sgn}\,\dot{x}}{m_{\rm c}+m}, \quad (34)$$

其中: 重力加速度g $= -9.8\,\mathrm{m/s^2}$, 小车的重量$m_{\rm c} = 1.0$ kg, 摆杆重量$m = 0.1$ kg, 摆杆一半的长度$l = 0.5$ m, 小车在导轨上的摩擦系数$\mu_{\rm c} = 0.0005$, 摆杆与小车的摩擦系数$\mu_{\rm p} = 0.000002$. 对参数方程的更新采用欧拉方法计算, 时间步长设定为0.02 s, 这样可以很方便地求得小车的运动速度和位置以及摆杆的角速度和摆角度.

在仿真实验中按物理定律给出运动方程式, 但倒立摆学习系统事先并不知道其运动规律, 它的知识结构是在不断学习过程中逐步建立起来的. 在实验中, 设定参数为: 学习率$\alpha = 0.2$, 折扣因子$\gamma = 0.95$, 资格迹系数$\lambda = 0.8$, 探索行为选择概率$\varepsilon = 0.1$, 改进残差法参数$\mu = 0.1$. 神经网络采用4–16–1结构, 隐层节点采用sigmoid型激活函数, 输出层节点采用线性函数.

6.2 实验结果与分析(Experimental results and analysis)

为了验证算法的有效性, 将倒立摆控制仿真实验进行40次. 每次实验都初始化神经网络的权值参数, 每次实验包含若干轮的学习过程, 每一轮可能成功, 也可能失败. 每轮实验从一个有效的随机位置开始, 由力控制倒立摆的平衡, 若倒立摆在一轮学习过程中能保持10000步不倒掉, 就认为它学习到的知识能够成功地控制倒立摆. 若本轮控制实验失败或能保持成功步数达到10000步, 则重新开始新一轮的学习.

表1给出了一个统计表, 记录了40次仿真实验中, 每次实验系统能成功控制倒立摆所经历的学习轮数. 在这40次实验中, 采用本文的算法, 学习系统都能有效地学习并成功地控制倒立摆. 其中: 最多学习轮数为18, 最少学习轮数为8, 平均学习轮数为12.05.

表 1 每次实验能成功控制倒立摆的学习轮数

Table 1 Numbers of episodes till the inverted pendulum can be controlled successfully in each experiment

编号	轮数	编号	轮数	编号	轮数	编号	轮数
1	12	11	10	21	16	31	12
2	14	12	11	22	12	32	9
3	9	13	12	23	10	33	11
4	12	14	18	24	9	34	10
5	18	15	13	25	16	35	15
6	8	16	12	26	12	36	17
7	8	17	9	27	17	37	10
8	13	18	9	28	9	38	9
9	17	19	10	29	10	39	10
10	14	20	13	30	13	40	14

本文从实验中抽取第11次实验, 对其实验过程进行观察, 发现按照本文的方法在经历了前9轮的失败后, 从第10轮开始, 系统能成功地实现倒立摆控制. 前10轮的学习步数分别为: $7, 10, 10, 36, 18, 74, 64, 706, 2411, 10000$. 仿真实验的学习过程曲线见图3.

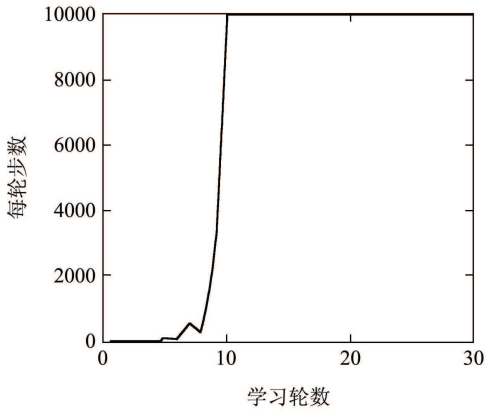

图 3 算法仿真性能

Fig. 3 Performance of the algorithms in simulation experiment

将本文方法结果与其他方法结果做一个对比. Barto等[18]提出了自适应启发评价(adaptive heuristic critic, AHC)方法, 将4维参数作为输入, 采用两个单层神经网络分别作为联想搜索单元(associative search element, ASE)和自适应评价单元(adaptive critic element, ACE), 实现控制倒立摆, 其参数设置跟本文相同. 这种方法将连续状态离散化, 没有导入先验知识, 在实现中较为复杂. Anderson等[19]在AHC方法基础上, 提出方法并实现了连续状态的控制. Berenji等[20]提出了一种基于广义近似推理的智能控制(generalized approximate reasoning based intelligent control, GARIC)方法, 采用模糊逻辑的方法, 实现了基于泛化规则智能控制结构的强化学习系统来控制倒立摆平衡. Lin等[21]提出了一种强化模糊自适应学习控制网络(reinforcement fuzzy adaptive learning control network, RFALCON)方法来解决倒立摆问题, 他们植入了模糊先验知识, 通过调节Critic网络和Action网络进行动态的参数学习. Moriarty等[22]研究了基于表格的Q学习算法实现倒立摆平衡问题, 同时提出了一个基于符号的、自适应进化神经网络的SANE(scanner access now easy)算法. 蒋国飞等[13]采用基于Q学习算法和BP神经网络来研究倒立摆控制问题, 实现了倒立摆的无模型控制, 这种方法没有运用资格迹技术. Lagoudakis等[23]利用最小二乘策略迭代(least-squares policy iteration, LSPI)算法, 采用基于基函数逼近和最小策略迭代法对倒立摆问题进行了研究. Bhatnagar等[24]实现了PG算法, 他们采用了自然梯度法和函数拟合的思想进行时域差分学习, 在线训练值函数的参数. Martín等[25]提出一种基于加权k近邻的强化学习方法kNN–TD, 将当前状态最临近的k个状态的Q值进行加权拟合, 求得当前Q值, 这样较好地对Q值进行了泛化. 为提高学习效率, 他们进而提出了基于资格迹的kNN–TD(λ)算法. Lee等[26]提出一种接受域加权的执行器–评价器(receptive field weighted actor-critic, RFWAC)算法, 采用了增量构建的径向基网络来构成, 以接受域加权回归作为其理论基础. 接受域用来构建局部模型, 其形状和规模可以进行自适应控制. Vien等[27]提出一种通过评价强化手动训练智能体(training an agent manually via evaluative reinforcement, TAMER)算法, 这种方法植入训练者早期的训练知识, 再进行强化学习. 采用的学习框架易于实现, 这种方法较好地运用于倒立摆的训练上. 各种方法的性能比较如表2所示.

为了进一步分析本文算法的性能, 图4–6分别给出了系统学习到第50轮时小车位置、摆杆角度以及外界对小车控制力随时间变化的曲线图. 图4–5设定测试时间为300 s, 行为次数为30000步. 从曲线图中看出, 小车的位置和角速度都在规定范围之内, 可见本算法取得了较好的学习和控制效果. 图6只给出的测试时间为50 s, 行为次数在2500步内, 外界对倒立摆系统进行控制的时间–作用力曲线图.

表 2 实现倒立摆控制的各种算法性能比较

Table 2 Performance comparison of various algorithms to control the cart-pole system

学习控制方法	连续状态	先验知识	平均学习轮数
AHC	否	否	457
Anderson	是	否	8000
GARIC	是	是	300
RFALCON	是	是	15
基于表格的Q学习	否	否	3103
SANE	是	否	535
蒋国飞	是	否	1000
LSPI	是	否	275
PG	是	否	245
kNN–TD(λ)	是	否	29
RFWAC	是	否	153
TAMER	是	是	150
本文方法	是	否	12

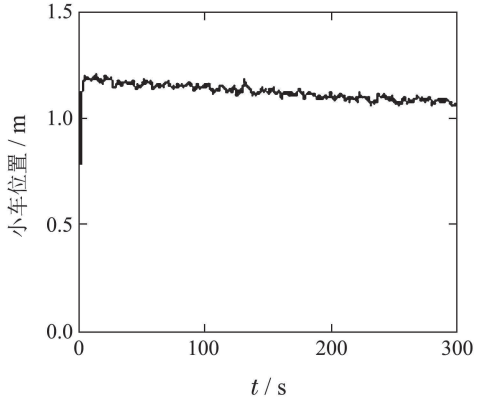

图 4 仿真实验中小车位置随时间变化图

Fig. 4 Time-position curve of the cart in simulation experiment

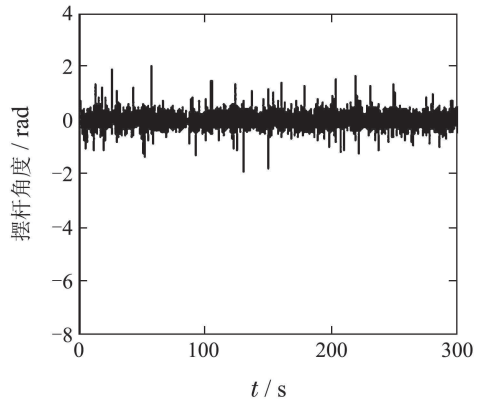

图 5 仿真实验中摆杆角度随时间变化图

Fig. 5 Time-angle curve of the pole in simulation experiment

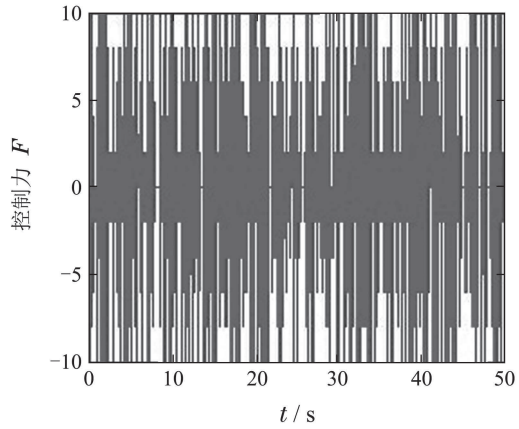

图 6 仿真实验中控制力随时间变化图

Fig. 6 Time-force curve in simulation experiment

在表2中GARIC方法充分利用了先验知识进行强化学习, 性能有了较大的提高, 使学习轮数提高到300. RFALCON方法同样引入了先验知识, 使学习轮数提高到15. 本文实验结果没有植入先验知识, 获得了较好的学习性能. 植入部分先验知识, 重做以上实验. 先验知识描述如下:

If $\theta > 0$ and $\dot{\theta} > 0$, then $F > 0$;

If $\theta < 0$ and $\dot{\theta} < 0$, then $F < 0$.

同样进行40次实验, 每次实验学习系统都能有效地学习并成功地控制倒立摆. 表3给出了一个统计表, 记录了植入上述知识后, 每次实验系统能成功控制倒立摆所经历的学习轮数. 其中: 最多学习轮数为14, 最少学习轮数为5, 平均学习轮数为7.93. 可见, 植入先验知识能大大提高强化学习的效率.

表 3 植入先验知识后每次实验能成功控制倒立摆的学习轮数

Table 3 Numbers of episodes till the inverted pendulum can be controlled successfully in each experiment after the priori knowledge is implanted

编号	轮数	编号	轮数	编号	轮数	编号	轮数
1	6	11	13	21	8	31	5
2	10	12	6	22	5	32	11
3	7	13	8	23	6	33	10
4	5	14	6	24	8	34	7
5	6	15	10	25	7	35	9
6	12	16	5	26	14	36	8
7	8	17	13	27	12	37	7
8	5	18	7	28	9	38	6
9	9	19	6	29	6	39	7
10	11	20	8	30	6	40	5

7 结论(Conclusions)

传统的强化学习方法学习效率较低, 为了解决这一问题, 人们往往采用各种方法对其进行改进. 其中利用函数拟合曲面能逐步逼近强化学习的函数值. 神经网络具有较好的泛化能力和良好的学习性能, 作为一种典型的非线性参数拟合器, BP神经网络在数据拟合方面得到了广泛的应用. 本文的研究内容是基于BP神经网络的强化学习技术研究, 结合资格迹方法能大大提高强化学习的效率, 这成为强化学习研究的重要方法之一, 但采用资格迹方法对神经网络的隐层权值更新带来了较高的复杂性. 本文结合资格迹技术, 不但实现了BP神经网络输出层权值的更新, 而且提出一种方法实现了神经网络隐层权值的更新, 提高了学习效率. 针对采用直接梯度法进行权值更新的方法其收敛性能较差的问题, 提出了一种改进的残差法, 将输出层权值和隐层权值更新量放在一个一维向量空间中, 将残差法和直接梯度法结合来进行权值更新, 保证了BP神经网络在学习过程中的良好性能. 本文针对倒立摆控制问题进行算法验证, 获得了较理想的效果. 本算法具有较强的通用性, 可以应用到多个理论和应用领域.

参考文献(References):

[1] SUTTON R S, BARTO A G. *Reinforcement Learning: An Introduction* [M]. Cambridge Massachusetts: MIT Press, 1998.

[2] LIU C, XU X, HU D. Multiobjective reinforcement learning: a comprehensive overview [J]. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 2013, 99(4): 1 – 13.

[3] WIERING M, OTTERLO M V. *Reinforcement Learning State of the Art* [M]. Berlin: Springer-Verlag, 2012, 10(3): 325 – 331.

[4] 戴朝晖, 袁娇红, 吴敏, 等. 基于概率模型的动态分层强化学习 [J]. 控制理论与应用, 2011, 28(11): 1595 – 1600.
(DAI Zhaohui, YUAN Jiaohong, WU Min, et al. Dynamic hierarchical reinforcement learning based on probability model [J]. *Control Theory & Applications*, 2011, 28(11): 1595 – 1600.)

[5] LUCIAN B, ROBERT B, BART D S. *Reinforcement Learning and Dynamic Programming Using Function Approximators* [M]. New York: CRC Press, 2010.

[6] DRIES S VAN DEN, WIERING M A. Neural-fitted TD-leaf learning for playing othello with structured neural networks [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2012, 23(11): 1701 – 1713.

[7] SUTTON R S. Learning to predict by the methods of temporal differences [J]. *Machine Learning*, 1988, 3(1): 9 – 44.

[8] DAYAN P, SEJNOWSKI T J. TD(λ) converges with probability 1 [J]. *Machine Learning*, 1994, 14(1): 295 – 301.

[9] 刘智斌, 曾晓勤. 基于路径引导知识启发的强化学习方法 [J]. 四川大学学报(工程科学版), 2012, 44(5): 136 – 142.
(LIU Zhibin, ZENG Xiaoqin. A method of heuristic reinforcement learning based on acquired path guiding knowledge [J]. *Journal of Sichuan University (Engineering Science Edition)*, 2012, 44(5): 136 – 142.)

[10] MIROLLI M, SANTUCCI V G, BALDASSARRE G. Phasic dopamine as a prediction error of intrinsic and extrinsic reinforcements driving both action acquisition and reward maximization: a simulated robotic study [J]. *Neural Networks*, 2013, 39(3): 40 – 51.

[11] BHASIN S, KAMALAPURKAR R, JOHNSON M, et al. A novel actor-critic-identifier architecture for approximate optimal control of uncertain nonlinear systems [J]. *Automatica*, 2013, 49(1): 82 – 92.

[12] BAIRD L C. Residual algorithms: reinforcement learning with function approximation [C] //*Proceedings of the 12th International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc, 1995: 30 – 37.

[13] 蒋国飞, 吴沧浦. 基于Q学习算法和BP神经网络的倒立摆控制 [J]. 自动化学报, 1998, 24(5): 662 – 666.
(JIANG Guofei, WU Cangpu. Learning to control an inverted pendulum using Q–learning and neural networks [J]. *Acta Automatica Sinica*, 1998, 24(5): 662 – 666.)

[14] WATINS C J. *Learning from delayed rewards* [D]. Cambridge: Cambridge University, 1989.

[15] KOHL N, MIIKKULAINEN R. An integrated neuroevolutionary approach to reactive control and high-level strategy [J]. *IEEE Transactions on Evolutionary Computation*, 2012, 16(4): 472 – 488.

[16] HINOJOSA W M, NEFTI H. System control with generalized probabilistic fuzzy-reinforcement learning [J]. *IEEE Transactions on Fuzzy System*, 2011, 19(1): 51 – 64.

[17] WANG X, CHENG Y, YI J. A fuzzy actor-critic reinforcement learning network [J]. *Information Sciences*, 2007, 177(18): 3764 – 3781.

[18] BARTO A G, SUTTON R S. Neuronlike adaptive elements that can solve difficult learning control problem [J]. *IEEE Transactions on Systems, Man, and Cybernetics*, 1983, 13(5): 835 – 846.

[19] ANDERSON C W. Learning to control an inverted pendulum using neural networks [J]. *IEEE Control System Megazine*, 1989, 9(3): 31 – 37.

[20] BERENJI H R, KHEDHAR P. Learning and turning fuzzy logic controllers through reinforcements [J]. *IEEE Transactions on Neural Networks*, 1992, 3(5): 724 – 740.

[21] LIN C J, LIN C T. Reinforcement learning for an ART-based fuzzy adaptive learning control network [J]. *IEEE Transactions on Neural Networks*, 1996, 7(3): 709 – 731.

[22] MORIARTY D E, MIKKULAINEN R. Efficient reinforcement learning through symbiotic evolution [J]. *Machine Learning*, 1996, 22(1/3): 11 – 32.

[23] LAGOUDAKIS M G, PARR R. Least-squares policy iteration [J]. *Journal of Machine Learning Research*, 2003, 4(12): 1107 – 1149.

[24] BHATNAGAR S, SUTTON R, GHAVAMZADEH M, et al. Natural actor-critic algorithms [J]. *Automatica*, 2009, 45(11): 2471 – 2482.

[25] MARTÍN H J A, LOPE J D, MARAVALL D. Robust high performance reinforcement learning through weighted $k-$nearest neighbors [J]. *Neurocomputing*, 2011, 74(8): 1251 – 1259.

[26] LEE D, LEE J. Incremental receptive field weighted actor-critic [J]. *IEEE Transactions on Industrial Informatics*, 2013, 9(1): 62 – 71.

[27] VIEN N A, ERTEL W, CHUNG T C. Learning via human feedback in continuous state and action spaces [J]. *Applied Intelligence*, 2013, 39(2): 267 – 278.

作者简介:

刘智斌 (1968–), 男, 博士, 副教授, 目前研究方向为机器学习、智能计算等, E-mail: lzbxian@163.com;

曾晓勤 (1957–), 男, 博士, 教授, 博士生导师, 目前研究方向为人工神经网络、图文法等, E-mail: xzeng@hhu.edu.cn;

徐 彦 (1979–), 男, 博士, 讲师, 目前研究方向为人工神经网络、模式识别等, E-mail: xuyannn@njau.edu.cn;

禹继国 (1972–), 男, 博士, 教授, 目前研究方向为分布式计算、图论等, E-mail: jiguoyu@sina.com.