一种带自适应学习率的综合随机梯度下降Q-学习方法

金海东[1] 刘 全[1),2),3),4)] 陈冬火[1]

[1](苏州大学计算机科学与技术学院 江苏 苏州 215006)

[2](软件新技术与产业化协同创新中心 南京 210000)

[3](吉林大学符号计算与知识工程教育部重点实验室 长春 130012)

[4](苏州大学江苏省计算机信息处理技术重点实验室 江苏 苏州 215006)

摘 要 在线强化学习中,值函数的逼近通常采用随机梯度下降(Stochastic Gradient Descent,SGD)方法.在每个时间步,SGD 方法使用强化学习算法获取随机样本,计算损失函数的局部梯度,单次模型参数更新的计算量小,适合在线学习.但是,由于目标函数不同维度存在梯度差异,SGD 方法会产生优化震荡,导致迭代次数增多,收敛速度变慢甚至不能收敛.本文提出一种带自适应学习率的综合随机梯度下降方法(Adaptive Learning Rate on Integrated Stochastic Gradient Descent,ALRI-SGD),对 SGD 做了两方面改进:(1)在基于参数预测的基础上,利用历史随机梯度信息综合计算当前时间步的更新梯度;(2)根据不同维度的历史梯度信息,动态计算每个维度的学习率.在一定的数学约束条件下,证明了 ALRI-SGD 方法的收敛性.把 ALRI-SGD 方法与基于线性函数逼近的离策略 Q-学习算法结合,用于求解强化学习中经典的 Mountain Car 问题和平衡杆问题,并与基于 SGD 的 Q-学习算法进行实验比较.实验结果表明,ALRI-SGD 方法能动态匹配模型参数在不同维度上的梯度差异,并使学习率自动更新以适应不同维度的数据特征.ALRI-SGD 方法在收敛效率和收敛稳定性两个方面都有提升.

关键词 强化学习;综合随机梯度下降;自适应学习率;参数预测;Q-学习

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2019.02203

Adaptive Learning-Rate on Integrated Stochastic Gradient Decreasing Q-Learning

JIN Hai-Dong[1] LIU Quan[1),2),3),4)] CHEN Dong-Huo[1]

[1](*School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu* 215006)

[2](*Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing* 210000)

[3](*Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun* 130012)

[4](*Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, Jiangsu* 215006)

Abstract The basic idea of reinforcement learning is to learn the best strategy to reach the goal by maximizing the cumulative rewards that the agent receives from the environment. In the online reinforcement learning method based on value function approximation, SGD is used to update the weights of the value function, that is, to update the model parameters in the negative gradient direction to minimize the loss function. On each time-step a random sample is obtained according to the ε-greedy strategy to update the model parameters. Therefore, each update requires a small amount of computation and which is suitable for online learning. Due to the difference in gradient rate of the objective function in different dimensions, SGD may make the optimization goal converge to another extreme point. It is also difficult for SGD to choose a suitable learning rate.

收稿日期:2018-01-23;在线出版日期:2018-11-23.本课题得到国家自然科学基金(61772355,61702055,61502323,61502329)、江苏省高等学校自然科学研究重大项目(17KJA520004、18KJA520011)、吉林大学符号计算与知识工程教育部重点实验室资助项目(93K172014K04,93K172017K18)、苏州市应用基础研究计划工业部分(SYG201422)、苏州市重点产业技术创新-前瞻性应用研究项目(SYG201804)、江苏省高校省级重点实验室(苏州大学)(KJS1524)资助.**金海东**,博士研究生,主要研究方向为强化学习、深度学习和深度强化学习.E-mail:haidong@suda.edu.cn.**刘 全**(通信作者),博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为强化学习、深度强化学习和自动推理.E-mail:quanliu@suda.edu.cn.**陈冬火**,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为强化学习、软件形式化方法.

A learning rate that is too small leads to a slower convergence rate, and too large leads to an obstacle to convergence. If different dimensions of training data have different characteristics and value statistics space, different dimensions should adopt different learning rates. Thus, the drawback of the stochastic gradient descent method is that it sometimes brings about the optimization oscillation, which makes the number of iterations increases and the convergence rate slows down. In this paper, we proposed an Adaptive Learning Rate on Integrated Stochastic Gradient Descent-ALRI-SGD, which makes two improvements on the traditional SGD: Add the gradient at time $t-1$ to the current t time gradient, to update the parameters as an integrated gradient. Based on the prediction of model parameters, the historical gradient information is used to calculate the gradient of the current time-step. This improvement makes the oscillation reduced in the direction with a larger gradient, and the speed of approaching an extremum faster in the direction with a smaller gradient. In the same dimension, it makes the parameter update faster before it approaches the extremum. If the parameter exceeds the extremum and the oscillation occurs, it will actively reduce the update speed; the learning rate of each dimension is dynamically calculated according to the historical gradient sum of squares of the dimension. The cumulative square sum of the gradients gradually increases as the training progresses, so that the learning rate will gradually decrease. The ALRI-SGD based Q-learning algorithm was used to enhance the classical Mountain Car tasks and the Pole Balancing problem, which were compared with SGD-based Q-learning algorithm. The experimental results showed that ALRI-SGD can dynamically match the gradient differences of model parameters in different dimensions and can update the learning-rate automatically to adapt to the data features of different dimensions. The algorithm has obvious advantages in both convergence process and convergence efficiency. There are still some problems to be further studied and improved in this work, including: the convergence performance of the algorithm is sensitive to the historical gradient discount rate, its setting still needs further study; in the convergence proof of the ALRI-SGD method, the convex function is assumed to be its application brings theoretical restrictions.

Keywords reinforcement learning; ALRI-SGD; adaptive learning-rate; parameters prediction; Q-learning

1 引 言

强化学习(Reinforcement Learning,RL)的基本思想是通过最大化智能体(agent)从环境中获得累计奖赏,学习到达目标的最优策略.强化学习可以模仿人类从婴儿到成年的学习过程,这种学习方式更接近人类的学习过程,如感觉和直觉的学习[1].谷歌公司将具有感知能力的深度学习和具有决策能力的强化学习结合,形成了深度强化学习(Deep Reinforcement Learning,DRL).AlphaGo Zero[2]通过深度强化学习方式自我学习和完善,不再使用人类知识学习围棋,以 100∶0 击败了此前版本的 AlphaGo.强化学习目前已经广泛应用于仿真模拟[3]、优化与调度[4-5]、游戏博弈[6-7]等领域.

梯度下降(Gradient Descent,GD)是求解强化学习优化问题最常用的方法,是用于逼近最小偏差模型.梯度下降算法包括全量梯度下降、随机梯度下降和小批量随机梯度下降.其中随机梯度下降(Stochastic Gradient Descent,SGD)每次随机选择一个样本在线更新模型参数,单次学习的速度快,适合于在线强化学习.

由于目标函数在不同维度上梯度变化率的差异,SGD 可能使优化目标收敛到另外的极值点,这种震荡导致收敛过程复杂化,迭代次数增多,收敛速度变慢.Dauphin 等人[8]认为,这种现象是目标函数的鞍点造成的.另外,SGD 难以选择合适的学习率,学习率太小导致收敛速度变慢,太大会阻碍收敛[9].实际应用中一般采用逐步降低学习率的方式,对凸函数和非凸函数都能保证收敛到极值.但是人工安

排学习率需要根据训练数据集的特征提前定义列表或阈值[10-11]，否则不能反映数据集特性，因此不适合不同场景下强化学习模型参数的自适应优化. 另外，如果不同维度的训练数据有不同特点和取值统计空间，则不同维度应该采用不同的学习率. 这些存在的问题，都要求根据具体的应用场景，人工对 SGD 方法进行优化.

近年来，随着深度神经网络的发展，梯度下降作为监督学习的模型优化方法再次成为研究热点. Qian 等人[12]提出 Momentum 方法，在更新模型参数时，对当前梯度与上次梯度方向相同的维度进行梯度加强，与上次梯度方向不同的维度进行梯度减弱，以减少震荡的方式获得更快的收敛速度. NAG 方法[13]在计算参数的梯度时，在损失函数中减去梯度项，将其作为下一次参数的预估，使参数更新速率自适应梯度变化，显著提高了循环神经网络在一些任务上的性能. Adagrad 方法[14]针对不同维度自适应调整学习率，对低频出现的参数增加学习率，高频出现的参数减少学习率，适合处理稀疏数据. Dean 等人[15]发现，Adagrad 方法能提升 SGD 的鲁棒性. 谷歌公司使用 Adagrad 方法训练大规模多层卷积神经网络，能识别 YouTube 视频中的猫. Pennington 等人[16]使用 Adagrad 方法训练 GloVe 单词向量，低频词自适应使用更大的学习率. Adadelta 方法① 是 Adagrad 的扩展，仅计算有限时间区间内的梯度累积和，解决 Adagrad 学习率单调下降问题.

由于强化学习自身的特点，上述针对监督学习的 SGD 改进难以用于强化学习. 从机器学习的角度看，强化学习呈现出了一系列挑战. 强化学习在与环境交互的同时进行在线学习，因此需要一个效率很高的模型. 在监督学习算法中，假设数据样本是独立的，但是在强化学习中数据是高度序列化的. 与监督学习假设数据服从同一分布不同，按照 GPI 流程[1]，强化学习训练样本分布随着算法学习到的新行为而改变，需要优化的是非稳态目标函数. 此外，强化学习要从稀疏、有噪声和延迟的奖励信号中学习. 这些特点，给强化学习的值函数拟合模型以及基于 SGD 的优化算法带来很大挑战，一些基于监督学习的 SGD 改进方法难以用于强化学习，例如深度 Q-网络(Deep Q-Network，DQN)[4,5]采用的是传统 SGD 进行反向误差传播. 本文从基于线性值函数逼近的强化学习模型出发，针对 SGD 存在的问题，提出一种带自适应学习率的综合随机梯度下降 ALRI-SGD 方法，对 SGD 做了两方面改进：(1) 在基于参数预测的基础上，利用历史随机梯度信息综合计算当前时间步的参数更新梯度；(2) 根据不同维度的历史梯度平方和，在每个时间步动态计算不同维度的学习率. 本文在一定的数学约束条件下证明了 ALRI-SGD 方法能够收敛，并把 ALRI-SGD 方法用于线性函数逼近的离策略 Q-学习算法，求解强化学习中经典的 Mountain Car 和平衡杆问题，并与基于 SGD 的 Q-学习算法进行了实验比较. ALRI-SGD 能动态匹配目标函数在不同维度上的梯度差异，并使梯度学习率自动更新以适应不同维度的数据特征. 基于 ALRI-SGD 的 Q-学习算法在收敛效率和稳定性两个方面，都有显著提高.

2　背景知识

2.1　强化学习

强化学习通过与环境的交互，来实现学习目标. “学习者”称为 agent，agent 之外与之交互的一切要素都称为环境(environment). 目标是使 agent 在与环境交互过程中获得最大累积奖赏即回报[7]. 借助马尔可夫决策过程(Markov Decision Process，MDP)，可以形式化 RL 问题[17].

定义 1.　马尔科夫决策过程 MDP 为四元组 (S,A,ρ,f)，其中：

(1) S 是环境中所有状态的集合. $s_t\in S$ 表示 agent 在 t 时刻的状态；

(2) A 是 agent 在环境 S 中可执行动作的集合. $a_t\in A$ 表示 agent 在 t 时刻采取的动作；

(3) $\rho:S\times A\to R$ 是奖赏函数. $r_t=\rho(s_t,a_t)$ 表示 agent 在状态 s_t 执行动作 a_t 获得的立即奖赏，奖赏一般为标量；

(4) $f:S\times A\times S\to[0,1]$ 为状态转移概率函数. $f(s_t,a_t,s_{t+1})$ 表示 agent 在状态 s_t 执行动作 a_t 转移到下一状态 s_{t+1} 的概率.

在 RL 中，策略 $\pi:S\to A$ 是从状态空间到动作空间的映射，表示 agent 在状态 s_t 执行动作 a_t，并以概率 $f(s_t,a_t,s_{t+1})$ 转移到下一状态 s_{t+1}，同时接受来自环境的立即奖赏 r_t. 根据奖赏的累积方式，有几种不同类型的回报[18]，其中折扣回报是从 t 时刻开始到 T 时刻情节结束时带折扣率的累积回报，折扣回报定义为

① Zeiler M D. ADADELTA: An adaptive learning rate method. arXiv preprintarXiv:1212.5701, 2012

$$R_t \doteq \sum_{t'=t}^{T} \gamma^{t'-t} r'_t,$$

其中 $\gamma \in [0,1]$ 用于衡量未来奖赏对回报的影响，可以直观地解释为 agent 在考虑回报时的“远视”程度，或者解释为对未来奖赏不确定性的考虑程度[19].

策略 π 的状态-动作值函数 $Q^{\pi}(s,a)$ 是在当前状态 s 下执行动作 a，并遵循策略 π 直到情节结束，agent 获得的回报，表示为

$$Q^{\pi}(s,a) = E\{R_t \mid s_t = s, a_t = a, \pi\}.$$

对于所有的状态-动作对，如果策略 π^* 的期望回报大于或等于其他所有策略的期望回报，则称 π^* 为最优策略. 强化学习任务的目标是求解最优策略 π^* 以获得最大期望回报. 最优策略可能有多个，但都有相同的最优状态-动作值函数：

$$Q^*(s,a) = \max_{\pi} E\{R_t \mid s_t = s, a_t = a, \pi\}.$$

最优状态-动作值函数满足贝尔曼最优方程(Bellman optimality equation)，即

$$Q^*(s,a) = E_{s' \sim S}\{r + \gamma \max_{a'} Q^*(s',a') \mid s,a\}.$$

经典 RL 算法一般使用贝尔曼方程迭代求解 Q 值函数：

$$Q_{t+1}(s,a) = E_{s' \sim S}\{r + \gamma \max_{a'} Q_t(s_{t+1},a') \mid s,a\} \quad (1)$$

Q 值迭代算法从任意 Q_0 开始迭代，在第 l 轮的迭代中，利用式(1)更新 Q 值，式(1)也称为 Q 值迭代映射.

定义 2. 两轮迭代之间的 Q 值之差，称为时间差分(Temporal Difference，TD)：

$$\begin{aligned}\delta_t &\doteq Q_{t+1}(s,a) - Q_t(s,a) \\ &= r + \gamma \max_{a'} Q_t(s_{t+1},a') - Q_t(s,a) \end{aligned} \quad (2)$$

当 $i \to \infty$ 时，$Q_i \to Q^*$，即通过不断迭代使得状态-动作值函数 Q^* 最终收敛，并求得最优策略：

$$\pi^* \doteq \arg\max_{a \in A} Q^*(s,a).$$

2.2 线性函数逼近

大规模状态空间的强化学习任务中，无法精确存储每一个状态回报值，因此不能采用基于数据表的 Q 值迭代算法[19]. 当前的一些研究成果，如著名的 DQN，其值函数估计部分采用参数方程形式的近似值函数.

由于强化学习过程是在线的、动态的，训练集是 agent 与环境交互动时动态得到的，因此要求函数逼近方法必须能够从增量式的训练集中学习，另外，还要求能够解决目标函数不确定的问题. 在基于梯度下降的函数逼近方法中，由于线性函数逼近计算量较小，基函数形式简单，且易于分析算法的理论性质[19]，近年来，在强化学习中得到广泛应用.

假设 Q 值函数逼近器包含一个 n 维的参数向量 $\boldsymbol{\theta}$. 近似函数逼近器可以表示为近似映射 F，$F: R^n \to X$，其中 R^n 是 n 维的参数空间，X 是 Q 值函数空间，则每个参数向量 $\boldsymbol{\theta}$ 都对应一个近似的 Q 值函数空间：

$$\hat{Q}(s,a) = F(\boldsymbol{\theta}),$$

也可以等价写成关于状态-动作对的形式：

$$\hat{Q}(s,a) = F(\boldsymbol{\theta})(s,a),$$

其中 $F(\boldsymbol{\theta})(s,a)$ 表示近似 Q 值函数 $F(\boldsymbol{\theta})$ 对状态-动作对 (s,a) 的值评估. 因此，这种近似表示方法不需要存储每一个状态动作对的 Q 值，只需要存储一个 n 维的向量 $\boldsymbol{\theta}$. 如果状态-动作空间是离散的，一般 n 远小于 $|S| \cdot |A|$，因此近似值函数相对目标值函数，存在一定的近似误差.

线性 Q 值函数逼近器一般包含 n 个基函数(Basis Functions，BFs)，$\phi_1, \phi_2, \cdots, \phi_n: S \times A \to R$ 以及一个 n 维的参数向量 $\boldsymbol{\theta}$，基函数与参数向量之间满足线性关系. 线性条件下，在策略 π 下，状态-动作对 (s_t,a_t) 的近似 Q 值的计算如下：

$$Q_t^{\pi}(s_t,a_t) = E\left\{\sum_{n=0}^{\infty} \gamma^n r_{t+n+1} \mid s = s_t, a = a_t\right\} \quad (3)$$

$$\approx [F(\boldsymbol{\theta})](s_t,a_t) = \sum_{l=1}^{n} \phi_l(s_t,a_t)\boldsymbol{\theta} = \boldsymbol{\phi}^{\mathrm{T}}(s_t,a_t)\boldsymbol{\theta} \quad (4)$$

其中 $\boldsymbol{\phi}(s,a) = [\phi_1(s,a), \phi_2(s,a), \cdots, \phi_n(s,a)]^{\mathrm{T}}$ 是 n 维的基函数向量. 在一些文献中，基函数 $\boldsymbol{\phi}(s,a)$ 也称作状态-动作对 (s,a) 的特征向量[18].

线性函数逼近条件下，式(2)表示的 TD 误差 δ_t 为

$$\delta_t = r_{t+1} + \gamma \boldsymbol{\phi}^{\mathrm{T}}(s_{t+1},a')\theta_t - \boldsymbol{\phi}^{\mathrm{T}}(s_t,a_t)\theta_t \quad (5)$$

3 强化学习中的随机梯度方法

梯度下降法是利用负梯度方向来寻找每次迭代的搜索方向，使得迭代向目标函数增长最快的方向优化.

3.1 随机梯度下降

随机梯度下降(SGD)的简单形式如下：

$$f_{t+1}(k) = f(k) - \alpha \frac{\partial}{\partial k} f_t(k),$$

其中 $f(k)$ 是目标函数，$\frac{\partial}{\partial k} f(k)$ 是 $f(k)$ 的梯度，α 是学习率. 通过 SGD 方式，可以逐步逼近目标函数 $f(k)$ 的极值.

在基于梯度的 Q-学习算法中，需要假设近似函数 $F(\boldsymbol{\theta})$关于参数 $\boldsymbol{\theta}$ 可导. 由于强化学习要优化的是非稳态目标函数，而函数逼近属于监督学习的技巧，需要静态训练样本集. 为了得到基于梯度的 Q-学习算法，在当前状态 s_t执行动作 a_t后，需要假设能够获得当前状态-动作对真实回报值 $Q^*(s_t,a_t)$，以及后续的状态 s_{k+1}及奖赏值 r_{k+1}. 因此，构造的强化学习训练样本为：$(s_t,a_t)\rightarrow Q^*(s_t,a_t)$.

定义 3.　训练样本集与拟合模型之间的均方误差 MSE 为

$$MSE\doteq\sum_{t=0}^{T}(Q^*(s_t,a_t)-[F(\theta_t)](s_t,a_t))^2,$$

其中 $Q^*(s_t,a_t)$是样本 (s_t,a_t)的真实值，$F(\theta_t)$是基于参数 θ_t的拟合模型.

强化学习算法的目标是使得当前 Q 值函数 $[F(\theta_t)](s_t,a_t)$与极值函数 $Q^*(s_t,a_t)$之间的均方差 MSE 最小. 构造损失函数 $J(\theta_t)$为

$$J(\theta_t)\doteq(Q^*(s_t,a_t)-[F(\theta_t)](s_t,a_t))^2,$$

损失函数相对参数 $\boldsymbol{\theta}$ 的梯度为

$$\frac{\partial}{\partial\theta_t}J(\theta_t)=\frac{\partial}{\partial\theta_t}(Q^*(s_t,a_t)-[F(\theta_t)](s_t,a_t))^2$$

$$=-2(Q^*(s_t,a_t)-[F(\theta_t)](s_t,a_t))\frac{\partial}{\partial\theta_t}[F(\theta_t)](s_t,a_t),$$

采用 SGD 方法进行参数更新：

$$\theta_{t+1}=\theta_t-\frac{1}{2}\alpha_t\frac{\partial}{\partial\theta_t}J(\theta_t)$$

$$=\theta_t+\alpha_t(Q^*(s_t,a_t)-[F(\theta_t)](s_t,a_t))\frac{\partial}{\partial\theta_t}[F(\theta_t)](s_t,a_t)\tag{6}$$

这里 α_t是 t 时刻的学习率.

$Q^*(s_t,a_t)$其实是未知的，可以通过式(1)的计算来近似替代. 通过这种替代，得到近似 Q-学习算法的更新式：

$$\theta_{t+1}=\theta_t+\alpha_t(r_{t+1}+\gamma\max_{a'}[F(\theta_t)](s_{t+1},a')-$$
$$[F(\theta_t)](s_t,a_t))\frac{\partial}{\partial\theta_t}[F(\theta_t)](s_t,a_t)\tag{7}$$

式(7)中，求解了 t 时刻 $Q^*(s_t,a_t)$的近似值 $\hat{Q}_t$.

$$\hat{Q}_t=r_{t+1}+\gamma\max_{a'}[F(\theta_t)](s_{t+1},a'),$$

因此，

$$Q^*(s_t,a_t)\approx\hat{Q}_t=r_{t+1}+\gamma\max_{a'}[F(\theta_t)](s_{t+1},a')\tag{8}$$

t 时刻 (s_t,a_t)的近似损失函数 $\hat{J}(\theta_t)$改写为

$$J[\theta_t](s_t,a_t)\approx\hat{J}[\theta_t](s_t,a_t)$$
$$=(r_{t+1}+\gamma\max_{a'}[F(\theta_t)](s_{t+1},a')-$$
$$[F(\theta_t)](s_t,a_t))^2\tag{9}$$

3.2　线性函数的随机梯度

线性条件下，式(8)表示的近似极值函数 $\hat{Q}_t$可以简化为

$$\hat{Q}_t=r_{t+1}+\gamma\max_{a'}(\boldsymbol{\phi}^{\mathrm{T}}(s_{t+1},a')\theta_t,$$

式(9)表示的 t 时刻 (s_t,a_t)近似损失函数简化为

$$J(\theta_t)=(r_{t+1}+\gamma\max_{a'}\boldsymbol{\phi}^{\mathrm{T}}(s_{t+1},a')\theta_t-\boldsymbol{\phi}^{\mathrm{T}}(s_t,a_t)\theta_t)^2\tag{10}$$

因此：

$$\frac{\partial}{\partial\theta_t}J(\theta_t)=-2(\hat{Q}_t-\boldsymbol{\phi}^{\mathrm{T}}(s_t,a_t)\theta_t)\boldsymbol{\phi}(s_t,a_t)$$
$$=-2(r_{t+1}+\gamma\max_{a'}\boldsymbol{\phi}^{\mathrm{T}}(s_{t+1},a')\theta_t-$$
$$\boldsymbol{\phi}^{\mathrm{T}}(s_t,a_t)\theta_t)\boldsymbol{\phi}(s_t,a_t).$$

把 SGD 用于式(4)表示的线性函数逼近器，式(7)的参数更新可以简化为

$$\theta_{t+1}=\theta_t+\alpha_t(r_{t+1}+\gamma\max_{a'}\boldsymbol{\phi}^{\mathrm{T}}(s_{t+1},a')\theta_t-$$
$$\boldsymbol{\phi}^{\mathrm{T}}(s_t,a_t)\theta_t)\boldsymbol{\phi}(s_t,a_t)\tag{11}$$

式(11)的含义是使用样本 $(s_t,a_t)\rightarrow\hat{Q}_t$对参数向量 $\boldsymbol{\theta}$ 采用随机梯度下降方法进行更新，可以形式化表示为

$$\theta_{t+1}=\theta_t-\alpha_t\cdot\frac{\partial}{\partial\theta_t}(\theta_t;(s_t,a_t);\hat{Q}_t)\tag{12}$$

其中 α_t是 t 时刻的参数学习率.

定理 1.　如果目标函数 $J(\theta_t)$是凸函数，并且 $\hat{Q}_t$是 $Q^*(s_t,a_t)$的无偏估计，随机梯度方法 $\theta_{t+1}=\theta_t-\alpha_t\cdot\frac{\partial}{\partial\theta_t}(\theta_t;(s_t,a_t);\hat{Q}_t)$以概率 1 收敛到 θ^*.

式(11)表示的 $TD(0)$算法，由于 $\hat{Q}_t$不是 $Q^*(s_t,a_t)$的无偏估计，随机梯度下降方法在这种情况下，不能在理论上证明可以收敛到局部最优. 尽管如此，这类自举算法在一些实际应用中能获得更快的收敛速度，文献[1]对此有详细的实验研究.

4　综合随机梯度下降

由于目标函数在不同维度上梯度变化率的差异，SGD 更新值的方差很大，频繁更新之后，SGD 可能使目标函数跳入另一个极值. 这种震荡使得收敛到特定极值的过程复杂化，导致迭代次数增多，收敛速度变慢甚至发散.

4.1　综合随机梯度

如果在某时刻 t，损失函数 $J(\theta_t)$在 i 方向的梯度远大于 j 方向的梯度，由于这种陡谷在极值中经常出现[8]，随机梯度在 i 方向由于梯度很大导致不断震荡，收敛过程不稳定，而在 j 方向由于梯度很小

导致更新缓慢，并且总体收敛速度由梯度较小的 j 方向决定. 一种改进方案是在当前 t 时刻梯度基础上加上 $t-1$ 时刻的梯度，作为 $\boldsymbol{\theta}$ 更新的综合梯度，记作 I. 在 i 方向上，在震荡阶段由于前后两次梯度方向相反，梯度是抵消的，而在 j 方向上，在逼近极值阶段由于前后两次梯度方向相同，梯度是累加的. 采用综合梯度的效果是，在梯度较大的 i 方向减小了震荡，而在梯度较小的 j 方向上逼近极值的速度会加快. 在同一个维度上，这种改进使得参数在接近极值之前更新速度不断加快，在超过极值而产生震荡时，会主动降低更新速度.

定义 4. 目标函数 $J(\boldsymbol{\theta})$ 在 t 时刻的综合梯度 I_t 为

$$I_0=0,\ I_t=\lambda\cdot I_{t-1}+\frac{1}{2}\frac{\partial}{\partial\theta_t}J(\theta_t),$$

其中 λ 是 $t-1$ 时刻综合梯度历史折扣率，$\frac{\partial}{\partial\theta_t}J(\theta_t)$ 是目标函数在 t 时刻的梯度.

定义 5. 基于线性函数逼近的 $TD(0)$ 算法中，t 时刻综合梯度 I_t 为

$$I_t=\lambda\cdot I_{t-1}-(r_{t+1}+\gamma\max_{a'}\boldsymbol{\phi}^{\mathrm{T}}(s_{t+1},a')\theta_t-\boldsymbol{\phi}^{\mathrm{T}}(s_t,a_t)\theta_t)\boldsymbol{\phi}(s_t,a_t),$$

因此，基于综合梯度的 $\boldsymbol{\theta}$ 更新式为

$$\begin{aligned}\theta_{t+1}&=\theta_t-\alpha_t I_t\\&=\theta_t-\alpha_t(\lambda\cdot I_{t-1}-(r_{t+1}+\gamma\max_{a'}\boldsymbol{\phi}^{\mathrm{T}}(s_{t+1},a')\theta_t-\boldsymbol{\phi}^{\mathrm{T}}(s_t,a_t)\theta_t)\boldsymbol{\phi}(s_t,a_t)),\end{aligned}$$

其中 α_t 是 t 时刻的学习率. 参数更新过程中，算法需要保存 $t-1$ 时刻的综合梯度 I_{t-1}.

上述更新过程很好地适应了不同维度的梯度变化. 但是如果在达到极值之前沿着梯度方向盲目加快更新速度，很容易错过极值，实验效果并不令人满意. 计算梯度时，如果能够提前得到下一个时间步参数 θ_{t+1} 的值，并以 θ_{t+1} 计算损失函数的梯度，就可以预知梯度的变化，在到达极值之前减速从而避免错过极值. 由于实际上无法预知 θ_{t+1} 的值，可以使用 $\theta_t-\lambda I_{t-1}$ 作为 θ_{t+1} 的近似预计.

信息回放机制[20]通过存储并利用历史样本，来消除训练数据的关联性. 与之相比，综合随机梯度方法用于降低值函数逼近过程中的震荡，利用的是历史梯度信息，并且算法不需要增加额外的空间.

定义 6. 目标函数 $J(\boldsymbol{\theta})$ 在 t 时刻基于参数预测的综合梯度 I_t 为

$$I_0=0,\ I_t=\lambda I_{t-1}+\frac{1}{2}\frac{\partial}{\partial\theta_t}J(\theta_t-\lambda I_{t-1}) \quad (13)$$

其中 λ 是 $t-1$ 时刻综合梯度历史折扣率，$\frac{\partial}{\partial\theta_t}J(\theta_t)$ 是目标函数在 t 时刻的梯度.

为表示方便，设：

$$\delta_t=r_{t+1}+\gamma\max_{a'}\boldsymbol{\phi}^{\mathrm{T}}(s_{t+1},a')(\theta_t-\lambda I_{t-1})-\boldsymbol{\phi}^{\mathrm{T}}(s_t,a_t)(\theta_t-\lambda I_{t-1}),$$

对于线性函数逼近的 $TD(0)$ 算法，式(13)可简化表示为

$$I_t=\lambda I_{t-1}-\delta_t\boldsymbol{\phi}(s_t,a_t).$$

定义 7. 目标函数 $J(\boldsymbol{\theta})$ 在时间步 t 对第 i 维度 θ_i 基于参数预测的综合梯度 $I_{t,i}$ 为

$$I_{0,i}=0,\ I_{t,i}=\lambda I_{t-1,i}+\frac{1}{2}\frac{\partial}{\partial\theta_i}J(\theta_t-\lambda I_{t-1}) \quad (14)$$

因此，在线性函数逼近的 $TD(0)$ 算法中，时间步 t 对第 i 维度 θ_i 基于参数预测的综合梯度 $I_{t,i}$ 为

$$I_{t,i}=\lambda I_{t-1,i}-\delta_t\phi_i(s_t,a_t) \quad (15)$$

这种改进的综合梯度计算方法，使得参数更新速度可以自动适应不同维度的梯度变化.

t 时刻第 i 维度参数 $\theta_{t+1,i}$ 更新规则为

$$\theta_{t+1,i}=\theta_{t,i}-\alpha I_{t,i} \quad (16)$$

4.2 自适应学习率

在 t 时刻，式(16)对所有维度都采用了相同的学习率 α，并不适合具有不同数据特征的不同维度上的强化学习任务. 如果将学习率 α 按不同维度进行修正，t 时刻第 i 维度的学习率 $\alpha_{t,i}$ 调整为

$$\alpha_{t,i}\rightarrow\frac{\alpha}{\sqrt{G_{t,i}+\Im}},$$

其中 $\boldsymbol{G}$ 是一个向量，$G_{t,i}$ 表示从 0 时刻到 t 时刻 $J(\boldsymbol{\theta})$ 对于第 i 维参数 θ_i 梯度的累积平方和. $\Im$ 是常数平滑项，以避免分母为 0 的情况. 这里分母开平方是经验式，否则实验中容易导致不收敛等问题. 对于强化学习中的线性函数逼近器，t 时刻第 i 维度的 $G_{t,i}$ 为

$$G_{0,i}=0,\ G_{t,i}=G_{t-1,i}+\delta_t^2\phi_i(s_t,a_t)^2 \quad (17)$$

因此，时间步 t 在维度 i 上的学习率 $\alpha_{t,i}$ 为

$$\alpha_{t,i}=\alpha/\sqrt{G_{t-1,i}+\delta_t^2\phi_i(s_t,a_t)^2+\Im} \quad (18)$$

改进后的参数更新式为

$$\theta_{t+1,i}=\theta_{t,i}-\alpha_{t,i}I_{t,i},$$

累积的梯度平方和随着训练过程逐步增大，式(18)计算的学习率 $\alpha_{t,i}$ 会逐步减小.

定义 8. 在强化学习模型参数更新方法中，如果采用式(15)所示的基于参数预测的综合随机梯度，并采用式(18)所示的自适应学习率，称为带自适应学习率的综合随机梯度下降(ALRI-SGD)方法.

算法 1.　基于 ALRI-SGD 的 Q-学习算法.

输入：折扣因子 γ，探索方案 $\{\varepsilon_k\}_{k=0}^{\infty}$

基函数 $\phi_1,\phi_2,\cdots,\phi_n: S\times A\to R$

1. 初始化，如 $\theta_0=0,\alpha=0.1,\beta=0.9,G_0=0,I_0=0$
2. 获取初始状态 s_0
3. FOR 每个时间步 $t=0,1,2,\cdots$
4. $a_t=\begin{cases}a\in\arg\max\limits_{a'}Q_t(s_t,a'), & \text{以概率 } 1-\varepsilon_t\\ a\in A \text{ 中的均匀随机动作}, & \text{以概率 } \varepsilon_t\end{cases}$
5. 采取动作 a_t，观测下一状态 s_{t+1} 和奖赏 r_{t+1}
6. FOR θ_t 的每个维度 i：

① 式(15)计算 $I_{t,i}$

② 式(17)计算 $G_{t,i}$

③ 式(18)计算 $\alpha_{t,i}$

④ $\theta_{t+1,i}=\theta_{t,i}-\alpha_{t,i}I_{t,i}$

由于 ALRI-SGD 方法自动调节不同维度的学习率，基于 ALRI-SGD 的 Q-学习在特征稀疏的维度上采用相对较高的学习率，在其他特征的维度上采用相对较低的学习率. 这种基于预测的综合随机梯度更新方法，能避免过快的更新速度，并能提高算法的响应能力，同时参数更新速度自动适应不同维度的梯度变化. 存在的问题是，式(18)分母项积累了梯度平方和，导致随着训练过程学习率不断缩小甚至渐进为 0，有可能导致的问题还需要进一步研究.

5　理论分析

从理论上证明基于 ALRI-SGD 参数更新方法能够收敛，需要满足一定的数学约束条件.

假定 1.　式(8)中，假定 $\hat{Q}(s_t,a_t)$ 是值函数 $Q^*(s_t,a_t)$ 的无偏估计.

如蒙特卡洛方法是无偏估计，而 $TD(\lambda)$ 值函数估计方法是有偏估计. 虽然收敛性证明需要无偏估计的假定，但在一些实际应用中，基于 on-policy 或者 off-policy 的自举算法也能够与线性函数近似的梯度下降方法可靠结合，并收敛至一个解.

假定 2.　如果损失函数 $J(\boldsymbol{\theta})$ 满足如下条件，称其满足 Lipschitz 连续：

$$|J(\theta_1)-J(\theta_2)|\le L\|\theta_1-\theta_2\|,$$

其中 L 是 Lipschitz 常数，对于一个确定的损失函数，L 是定值. 假定 2 对损失函数的值相对参数的变化率进行了限制.

假定 3.　如果 $J(\boldsymbol{\theta})$ 的梯度函数满足值为 β 的 Lipschitz 连续，称 $J(\boldsymbol{\theta})$ 为 β 平滑：

$$\left\|\frac{\partial}{\partial\theta}J(\theta_1)-\frac{\partial}{\partial\theta}J(\theta_2)\right\|^2\le\beta\|\theta_1-\theta_2\|^2,$$

其中 $\|\boldsymbol{\theta}\|^2=\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\theta}$. 假定 3 对损失函数梯度的变化进行了限制：损失函数梯度之差的模长，不超过 β 倍模型参数之差的模长.

定理 2.　满足 β 平滑的损失函数有如下性质：

$$\left|J(\theta_1)-J(\theta_2)-\frac{\partial}{\partial\theta}J(\theta_2)^{\mathrm{T}}(\theta_1-\theta_2)\right|\le\frac{1}{2}\beta\|\theta_1-\theta_2\|^2 \tag{19}$$

证明.　构造插值函数 $g(t)=J(\theta_2+t(\theta_1-\theta_2))$，其关于 t 的导数为

$$g'(t)=\frac{\partial}{\partial\theta}J(\theta_2+t(\theta_1-\theta_2))^{\mathrm{T}}(\theta_1-\theta_2),$$

把函数值之差转化为积分：

$$J(\theta_1)-J(\theta_2)=g(1)-g(0)=\int_0^1 g'(t)\mathrm{d}t$$
$$=\int_0^1\frac{\partial}{\partial\theta}J(\theta_2+t(\theta_1-\theta_2))^{\mathrm{T}}(\theta_1-\theta_2)\mathrm{d}t,$$

代入式(19)左侧：

$$\left|J(\theta_1)-J(\theta_2)-\frac{\partial}{\partial\theta}J(\theta_2)^{\mathrm{T}}(\theta_1-\theta_2)\right|$$
$$=\left|\int_0^1\frac{\partial}{\partial\theta}J(\theta_2+t(\theta_1-\theta_2))^{\mathrm{T}}(\theta_1-\theta_2)\mathrm{d}t-\frac{\partial}{\partial\theta}J(\theta_2)^{\mathrm{T}}(\theta_1-\theta_2)\right|$$
$$=\left|\int_0^1\frac{\partial}{\partial\theta}J(\theta_2+t(\theta_1-\theta_2))^{\mathrm{T}}(\theta_1-\theta_2)\mathrm{d}t-\int_0^1\frac{\partial}{\partial\theta}J(\theta_2)^{\mathrm{T}}(\theta_1-\theta_2)\mathrm{d}t\right|$$
$$=\left|\int_0^1\left(\frac{\partial}{\partial\theta}J(\theta_2+t(\theta_1-\theta_2))^{\mathrm{T}}(\theta_1-\theta_2)-\frac{\partial}{\partial\theta}J(\theta_2)^{\mathrm{T}}(\theta_1-\theta_2)\right)\mathrm{d}t\right|$$
$$\le\int_0^1\left|\frac{\partial}{\partial\theta}J(\theta_2+t(\theta_1-\theta_2))^{\mathrm{T}}(\theta_1-\theta_2)-\frac{\partial}{\partial\theta}J(\theta_2)^{\mathrm{T}}(\theta_1-\theta_2)\right|\mathrm{d}t$$
$$\le\int_0^1\left|\left[\frac{\partial}{\partial\theta}J(\theta_2+t(\theta_1-\theta_2))-\frac{\partial}{\partial\theta}J(\theta_2)\right]^{\mathrm{T}}(\theta_1-\theta_2)\right|\mathrm{d}t$$
$$\le\int_0^1\sqrt{\left\|\frac{\partial}{\partial\theta}J(\theta_2+t(\theta_1-\theta_2))-\frac{\partial}{\partial\theta}J(\theta_2)\right\|^2\|\theta_1-\theta_2\|^2}\mathrm{d}t,$$

利用假定 3：

$$\le\int_0^1\sqrt{\beta\|t(\theta_1-\theta_2)\|^2\cdot\|\theta_1-\theta_2\|^2}\mathrm{d}t$$
$$=\beta\|\theta_1-\theta_2\|^2\int_0^1 t\mathrm{d}t=\frac{1}{2}\beta\|\theta_1-\theta_2\|^2.$$　　证毕.

定理 3.　如果 $J(\boldsymbol{\theta})$ 是满足 β 平滑的凸函数，则：

$$J(\theta_1)-J(\theta_2)\le\frac{\partial}{\partial\theta}J(\theta_1)^{\mathrm{T}}(\theta_1-\theta_2)-\frac{1}{2\beta}\left\|\frac{\partial}{\partial\theta}J(\theta_1)-\frac{\partial}{\partial\theta}J(\theta_2)\right\|^2 \tag{20}$$

证明.　设

$$x=\theta_2-\frac{1}{\beta}\left(\frac{\partial}{\partial\theta}J(\theta_2)-\frac{\partial}{\partial\theta}J(\theta_1)\right) \tag{21}$$

把式(20)的左边分解为

$$J(\theta_1)-J(\theta_2)=J(\theta_1)-J(x)+J(x)-J(\theta_2).$$

根据凸函数的假定：

$$J(\theta_1)-J(x)\leq\frac{\partial}{\partial\theta}J(\theta_1)^{\mathrm{T}}(\theta_1-x),$$

则：

$$J(\theta_1)-J(x)\leq\frac{\partial}{\partial\theta}J(\theta_1)^{\mathrm{T}}(\theta_1-\theta_2)+\frac{\partial}{\partial\theta}J(\theta_1)^{\mathrm{T}}(\theta_2-x) \tag{22}$$

由定理 3 得到：

$$J(x)-J(\theta_2)\leq\frac{\partial}{\partial\theta}J(\theta_2)^{\mathrm{T}}(x-\theta_2)+\frac{\beta}{2}\|x-\theta_2\|^2,$$

$$J(x)-J(\theta_2)\leq-\frac{\partial}{\partial\theta}J(\theta_2)^{\mathrm{T}}(\theta_2-x)+\frac{\beta}{2}\|x-\theta_2\|^2 \tag{23}$$

式(22)和式(23)的左、右边对应项相加，合并 θ_2-x 项：

$$J(\theta_1)-J(\theta_2)\leq\frac{\partial}{\partial\theta}J(\theta_1)^{\mathrm{T}}(\theta_1-\theta_2)+\left(\frac{\partial}{\partial\theta}J(\theta_1)-\frac{\partial}{\partial\theta}J(\theta_2)\right)^{\mathrm{T}}(\theta_2-x)+\frac{\beta}{2}\|x-\theta_2\|^2 \tag{24}$$

由式(21)得：

$$\theta_2-x=\frac{1}{\beta}\left(\frac{\partial}{\partial\theta}J(\theta_2)-\frac{\partial}{\partial\theta}J(\theta_1)\right),$$

代入式(24)：

$$J(\theta_1)-J(\theta_2)\leq\frac{\partial}{\partial\theta}J(\theta_1)^{\mathrm{T}}(\theta_1-\theta_2)+\frac{1}{\beta}\left(\frac{\partial}{\partial\theta}J(\theta_1)-\frac{\partial}{\partial\theta}J(\theta_2)\right)^{\mathrm{T}}\left(\frac{\partial}{\partial\theta}J(\theta_2)-\frac{\partial}{\partial\theta}J(\theta_1)\right)+\frac{\beta}{2}\cdot\frac{1}{\beta^2}\left\|\frac{\partial}{\partial\theta}J(\theta_1)-\frac{\partial}{\partial\theta}J(\theta_2)\right\|^2,$$

因此，

$$J(\theta_1)-J(\theta_2)\leq\frac{\partial}{\partial\theta}J(\theta_1)^{\mathrm{T}}(\theta_1-\theta_2)-\frac{1}{2\beta}\left\|\frac{\partial}{\partial\theta}J(\theta_1)-\frac{\partial}{\partial\theta}J(\theta_2)\right\|^2.$$

证毕.

定理 4. 满足假定 1、假定 2 和假定 3 的条件下，收敛条件 $\|\theta_{t+1}-\theta^*\|^2\leq\|\theta_t-\theta^*\|^2$ 成立.

其中 θ^* 是损失函数 $J(\boldsymbol{\theta})$ 负梯度方向的极值点.

在满足假定 1 的条件下，ALRI-SGD 方法在接近最终解产生震荡之前，由于维度 i 上不同时刻的梯度方向相同，也即 $\frac{\partial}{\partial\theta}J(\theta_t)$ 的符号相同，根据式(17)计算的 ∇_t 必定沿着梯度方向接近最终解.

下面进一步证明满足假定 2 和假定 3 条件下，ALRI-SGD 方法在逼近最终解的震荡阶段，能够收敛.

证明. t 时刻的参数解 θ_t 到最终解 θ^* 的距离为

$$\begin{aligned}\|\theta_{t+1}-\theta^*\|^2&=\|\theta_t-\alpha\cdot I_t-\theta^*\|^2\\&=\|\theta_t-\theta^*\|^2-2\alpha I_t^{\mathrm{T}}(\theta_t-\theta^*)+\alpha^2\|I_t\|^2\end{aligned} \tag{25}$$

根据定理 4，考察在 θ^* 点的线性拟合：

$$J(\theta_t)-J(\theta^*)\leq\left[\frac{\partial}{\partial\theta}J(\theta_t)\right]^{\mathrm{T}}(\theta_t-\theta^*)-\frac{1}{2\beta}\left\|\frac{\partial}{\partial\theta}J(\theta_t)-\frac{\partial}{\partial\theta}J(\theta^*)\right\|^2 \tag{26}$$

由于 θ^* 为最终解，因此：

$$J(\theta_t)-J(\theta^*)\geq0,\text{并且 }J(\theta^*)=0,$$

再由式(26)得：

$$\left[\frac{\partial}{\partial\theta}J(\theta_t)\right]^{\mathrm{T}}(\theta_t-\theta^*)-\frac{1}{2\beta}\left\|\frac{\partial}{\partial\theta}J(\theta_t)\right\|^2\geq0$$

$$-\left[\frac{\partial}{\partial\theta}J(\theta_t)\right]^{\mathrm{T}}(\theta_t-\theta^*)\leq-\frac{1}{2\beta}\left\|\frac{\partial}{\partial\theta}J(\theta_t)\right\|^2 \tag{27}$$

式(27)代入式(25)右侧中间项：

$$\|\theta_{t+1}-\theta^*\|^2\leq\|\theta_t-\theta^*\|^2-\frac{\alpha}{\beta}\left\|\frac{\partial}{\partial\theta}J(\theta_t)\right\|^2+\alpha^2\|I_t\|^2 \tag{28}$$

在参数更新的震荡阶段，由于不同时刻的梯度方向不同，即 $\frac{\partial}{\partial\theta}J(\theta_{t-1})$ 与 $\frac{\partial}{\partial\theta}J(\theta_t)$ 异号，根据式(17)，必定存在 $\|I_t\|^2\leq\left\|\frac{\partial}{\partial\theta}J(\theta_t)\right\|^2$. 因此：

$$\|\theta_{t+1}-\theta^*\|^2\leq\|\theta_t-\theta^*\|^2-\frac{\alpha}{\beta}\|I_t\|^2+\alpha^2\|I_t\|^2,$$

$$\|\theta_{t+1}-\theta^*\|^2\leq\|\theta_t-\theta^*\|^2-\alpha\left(\frac{1}{\beta}-\alpha\right)\|I_t\|^2 \tag{29}$$

因此，只要 $\alpha\left(\frac{1}{\beta}-\alpha\right)\|I_t\|^2\geq0$，即 $\alpha\leq1/\beta$，就可以保证 $\|\theta_{t+1}-\theta^*\|^2\leq\|\theta_t-\theta^*\|^2$，收敛条件成立. 利用式(18)计算 $\alpha_{t,i}$ 时，常数 α 设置为一个特定值，即能满足 $\alpha_{t,i}\leq1/\beta$ 条件，该特定值由梯度累计平方和以及 β 确定. 证毕.

6 实验及结果分析

6.1 Mountain Car 实验

本实验用于求解 Mountain Car 问题，有两个实验目的：(1) 分析 ALGI-SGD 方法在历史梯度折扣率 λ 取不同值时的收敛特性；(2) 比较 ALRI-SGD 方法与 SGD 方法的收敛性能.

Mountain Car 实例是把强化学习方法用于连续控制的经典例子. 实例如图 1 所示，由于小车的重力大于动力，无法通过直接加速到达右侧坡顶，需要借助左侧的小坡，加足油门获得足够的惯性后，才能把小车带到右侧山坡上的顶点. 状态空间是连续的，

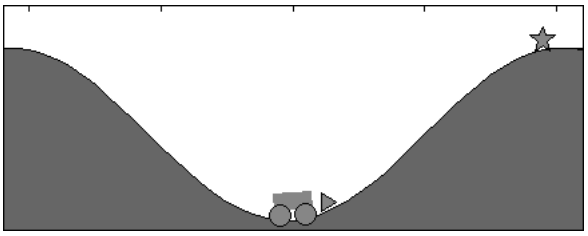

图 1　Mountain Car 实例

由位置 x 和速度 v 两个分量表示. 小车到达右侧坡顶获得+1 的立即奖赏，其他情况的立即奖赏为 0. 状态转换如下：

$$x_{t+1}=bound[x_t+v_{t+1}],$$

$$v_{t+1}=bound[v_t+0.001a_t+-0.0025\cos(3x_t)].$$

边界操作被强制在 $-1.2\leq x_{t+1}\leq 0.5$ 以及 $-0.07\leq v_{t+1}\leq 0.07$. 当 x_{t+1} 到达左边界时，v_{t+1} 被重置为 0. 到达右边界时，情节结束.

状态空间采用 Tiling 编码，共使用 10 个 9×9 的 Tiling，每个 Tiling 有 8 个 tile. 共有 3 个动作，因此状态-动作对空间被离散为 30 个 Tiling.

实验 1.　收敛性分析.

ALGI-SGD 方法中，λ 是历史梯度折扣率，表示历史梯度信息对当前综合梯度以及参数预测的影响程度. 实验 1 展示了 λ 设置为不同值时，Mountain Car 实验的收敛过程.

图 2 展示了 λ 设置为不同值时，到达目标的时间步数随着学习情节数的收敛过程. $\lambda=0.1$ 时，前期能够快速学习，并且收敛之后非常稳定. $\lambda=0.9$ 时，前期难以到达目标，部分情节的时间步数超过 10 万，并且后期持续震荡. $\lambda=0.05$ 时，收敛速度和收敛稳定性比不上 $\lambda=0.1$. 从实验效果来看，$\lambda=0.1$ 收敛过程和收敛结果是最好的.

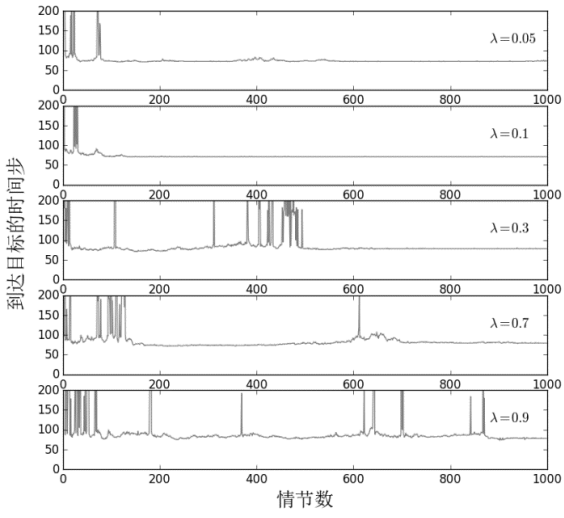

图 2　参数 λ 对时间步的影响

本实验的状态-动作空间采用 Tiling 编码，总共有 2430 个特征，对应有 2430 个需要学习的参数. 为了观察参数更新过程，从这些参数中随机挑选了第 1610 个参数分量，观察 λ 设置为不同数值时，该参数的更新过程，如图 3 所示. 总体上看，$\lambda=0.1$ 时该参数收敛效果最好，虽然收敛速度慢于 $\lambda=0.05$，但是只在第 162～164 情节时有一个较小的反向震荡，并且震荡的范围和幅度都小于 $\lambda=0.05$. $\lambda=0.9$ 时，参数更新的后期持续强烈震荡，不能收敛. 图 3 参数分量[1610]的收敛过程与图 7 时间步收敛过程基本吻合，显示了 λ 取不同值时，对算法收敛过程影响的一致性.

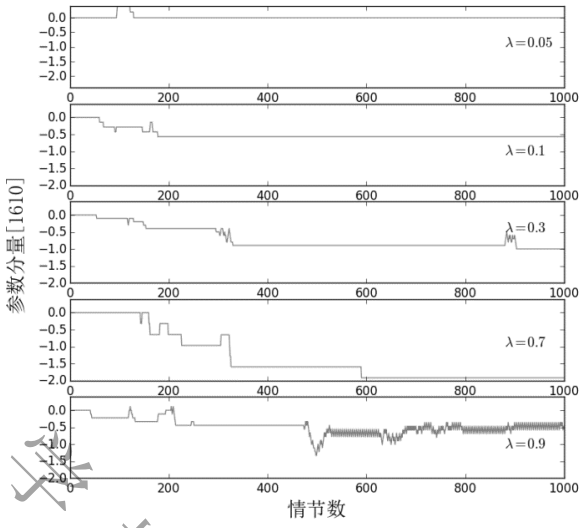

图 3　参数分量[1610]的更新过程

为了观察整体参数随 λ 不同取值的收敛过程，图 4 展示了在不同 λ 取值下，所有参数分量的平方和与学习情节数的关系. 由于不同参数在震荡过程

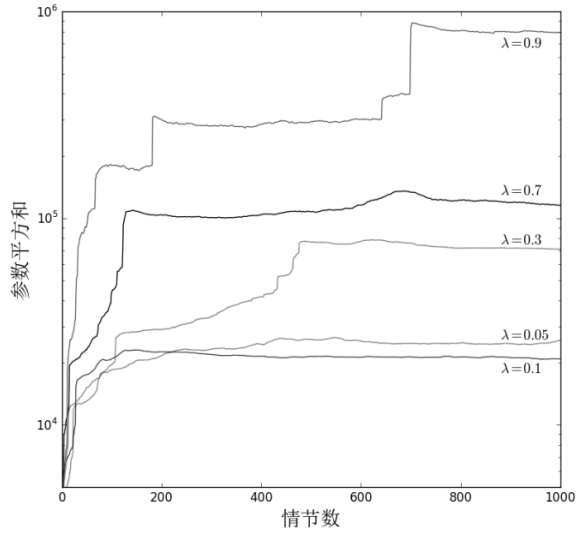

图 4　参数平方和的收敛过程

中存在相互抵消,图 4 的展示方法并不十分科学,但能大体上反映参数整体随 λ 取值的收敛过程.图中可以看出,$\lambda=0.1$ 时参数平方和最早收敛并且收敛过程平稳.$\lambda=0.9$ 时,1000 个情节之后参数平方和仍在继续上升(图中没有显示),说明参数更新过程持续震荡而不能收敛.

实验 2. 收敛性比较.

将基于 SGD 的 Q-学习算法和基于 ALRI-SGD 的 Q-学习算法分别用于 Mountain Car 实例.设置奖赏折扣率 $\gamma=1.0$,探索概率 $\varepsilon=0.1$,ALRI-SGD 方法的 $\lambda=0.1$.

图 5、图 6 展示了两个算法在每个情节包含的时间步,横坐标表示算法执行的情节数,纵坐标表示每个情节到达目标需要的时间步.由于算法的探索概率带来的随机性,值函数收敛之后行为策略仍然还有随机性,表现为每个情节的时间步不能完全稳定收敛.本实验以每个情节上下波动幅度百分比小于探索概率作为收敛标准.图 6 所示的 ALRI-SGD Q-学习算法在第 403 个情节时达到收敛标准,小车平均 92 个时间步到达右侧坡顶.图 5 所示的 SGD Q-学习算法在第 403 个情节没有达到收敛标准.不仅收敛速度 ALRI-SGD 方法优于 SGD 方法,ALRI-SGD 方法在收敛过程中表现得更为稳定.

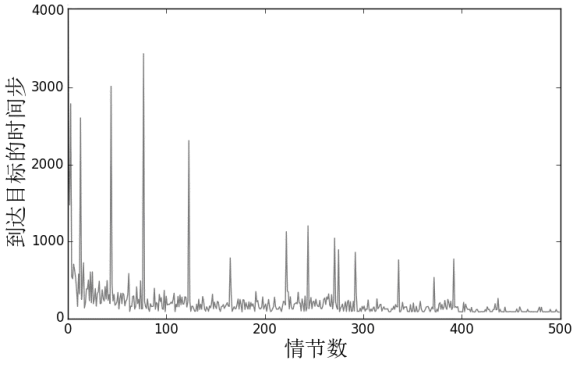

图 5 SGD Q-学习算法时间步收敛过程

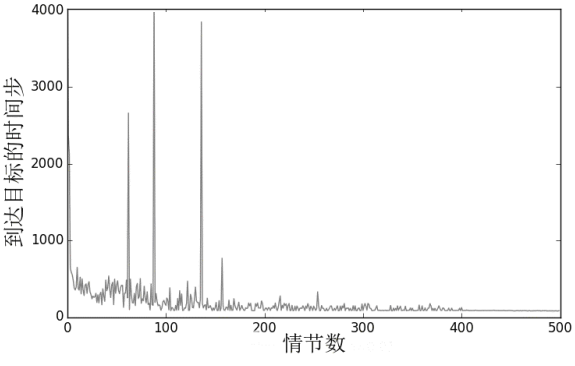

图 6 ALRI-SGD Q-学习算法时间步收敛过程

实验中比较 SGD Q-学习算法与 ALRI-SGD Q-学习算法的最优值函数更新情况.第 500 个情节时,两个算法的值函数都没有收敛,但 ALRI-SGD 方法的更新进度优于 SGD 方法.如 ALRI-SGD 方法部分状态的值函数已经超出 -25 接近 -30,但 SGD 方法所有状态的值函数都不超过 -25.Mountain Car 实验中,被频繁访问状态的回报值,比未探索状态的回报值更小.

ALRI-SGD 是对 SGD 方法的改进.上述实验表明,在 Mountain Car 实验中,基于 ALRI-SGD 的 Q-学习算法在收敛稳定性和收敛速度两个方面,都优于 SGD Q-学习算法.

通过实验 1 和实验 2 的一系列观察,ALGI-SGD 方法在历史梯度折扣率 $\lambda=0.1$ 附近时,有较好的实验效果,收敛速度与稳定性优于 SGD 方法,也优于 λ 的其他取值.而 λ 取值较大如大于 0.7 时,ALGI-SGD 方法的收敛性能不如 SGD 方法.

6.2 平衡杆实验

平衡杆如图 7 所示.水平方向上有一辆可以左右移动质量 $M_c=1\,\mathrm{kg}$ 的小车,小车上连接一根质量 $m_p=0.1\,\mathrm{kg}$ 长度 $l=1\,\mathrm{m}$ 的杆子.当杆子与竖直方向的角度满足$[-\pi/4,\pi/4]$时,认为杆子是平衡的.在没有外力的作用下,杆子无法持续保持平衡.利用 MDP 对该问题建模,目标是学习一个策略,即学习如何对小车施加水平向左或向右的作用力 F,使杆子保持平衡.施加力 F 的时间间隔 $\Delta t=0.1\,\mathrm{s}$,范围为$[-50,50]\mathrm{N}$,正力的方向向右,反之向左.

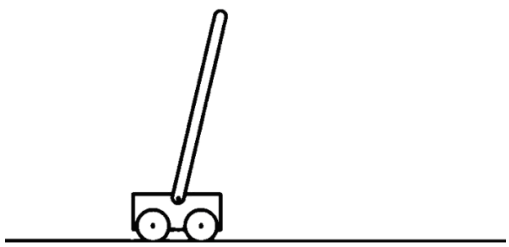

图 7 平衡杆示意图

平衡杆的状态是两维向量$(\boldsymbol{\theta},\dot{\boldsymbol{\theta}})$,$\boldsymbol{\theta}$ 表示杆子与竖直方向的夹角,$\dot{\boldsymbol{\theta}}$ 表示杆子的角速度.角加速度$\ddot{\boldsymbol{\theta}}$的计算表示为

$$\ddot{\boldsymbol{\theta}}=\frac{g\sin\boldsymbol{\theta}+\cos\theta\left(\dfrac{-F-m_p l\dot{\boldsymbol{\theta}}\dot{\boldsymbol{\theta}}\sin\boldsymbol{\theta}}{m_p+M_c}\right)}{l\left(\dfrac{4}{3}-\dfrac{m_p\cos^2\boldsymbol{\theta}}{m_p+M_c}\right)},$$

其中 $g=9.811\,\mathrm{m/s^2}$ 为重力加速度.状态转移函数:

$$\begin{cases}\theta_{t+1}=bound[\theta_t+\dot{\theta}_t\Delta t]\\ \dot{\theta}_{t+1}=bound[\dot{\theta}_t+\ddot{\theta}_{t+1}\Delta t]\end{cases}.$$

当$\dot{\boldsymbol{\theta}}>2$时，限制$\dot{\boldsymbol{\theta}}=2$；$\dot{\boldsymbol{\theta}}<-2$时，限制$\dot{\boldsymbol{\theta}}=-2$. 当$|\boldsymbol{\theta}|>\pi/4$时，认为杆子失去平衡并倒下，该情节结束. 奖赏函数：$|\boldsymbol{\theta}|>\pi/4$时奖赏为$-1$，否则为 0. 情节结束条件：$|\boldsymbol{\theta}|>\pi/4$或杆子平衡达到 3000 个时间步. 如果稳定在 3000 个时间步数没有倒下，认为平衡杆在该情节学习成功. 如果连续 2000 个情节不能学习成功，则认为该设置条件下无法学习成功，即认为需要∞个情节数才能学习成功.

实验中设置了 3 个离散动作：-1，0，1，分别表示小车受到-50 N、0 N、50 N 的作用力，其中负数表示向左，正数表示向右. 平衡杆的状态采用 tile 编码，实验中每个动作的状态空间设置 15 个 Tiling，每个 Tiling 有 10 个 tile. 因此，3 个动作总共有 45 个 Tiling，每个状态有$15\times10\times10=1500$个 0、1 特征. 由于奖赏函数的设置鼓励了探索，所以探索概率设置为 0.

为验证随机性对实验干扰的影响，实验分为两组：第 1 组没有随机外力的干扰；第 2 组小车在任意时刻都会受到水平方向$[-10,10]$N 范围内随机外力的干扰.

实验 1.　没有随机外力干扰的平衡杆.

当 ALRI-SGD 方法历史梯度折扣率$\lambda=0$，即不引入 ALRI-SGD 方法情况下，实验中学习率$\alpha\in[0.01,0.05]$时能够收敛. 如$\alpha=0.01$时，第 16 个情节就能学习成功. 当$\alpha=0.06$时，由于学习率过高，用于近似表示状态-动作的近似值函数持续震荡不能收敛. 表现在平衡杆上，第 240 个情节之后，每个情节稳定在 6 个时间步数就会倒下. 此时引入综合随机梯度，表 1 是$\alpha=0.06$时，历史梯度折扣率λ在不同取值时，需要学习的情节数.

由于没有随机外力干扰，表 1 中各项实验结果，每次都能稳定重复出现，因此表 1 中每项数据都是 1 次实验的结果. 与没有采用 ALRI-SGD 方法学习成功需要 37 个情节相比，当$\lambda\in[0.97,0.997]$时，ALRI-SGD 方法只需要 25 个情节就能学习成功，明显加快了近似值函数的收敛速度. 在此范围之外的λ值会导致Q值溢出不能收敛. 上述一系列实验中，$\lambda\in[0.97,0.997]$时学习成功需要的情节数都是 25，是巧合还是有理论上的必然性，需要继续研究.

表 1　没有随机外力干扰的平衡杆

λ	情节数
0	**37**
0.9	∞
0.95	∞
0.97	25
0.98	25
0.99	25
0.993	25
0.997	25
0.998	∞
0.999	∞

实验 2.　有随机外力干扰的平衡杆.

任意时刻小车会受水平方向$[-10,10]$N 范围内随机外力的干扰. 当学习率$\alpha=0.01$，没有引入 ALRI-SGD 即$\lambda=0$时，平均需要 77.36 个情节学习成功(对比第 1 组实验 36 个情节). 表 2 是当$\alpha=0.01$时引入 ALRI-SGD 方法，历史梯度折扣率在不同取值时的收敛情况.

表 2　有随机外力干扰的平衡杆

λ	平均情节数
0	**77.36**
0.001	66.25
0.01	70.43
0.1	74.72
0.2	66.95
0.3	57.10
0.4	65.44
0.5	**47.28**
0.6	54.59
0.7	71.64
0.8	58.81
0.9	∞
0.95	∞

由于随机外力作用，每次实验学习成功需要的情节数有随机性，表 2 中每项实验结果都是 100 次实验结果的平均值.

$\lambda\geq0.9$时，学习不成功，但是Q值也没有溢出. $\lambda\in[0.01,0.8]$时，学习成功需要的平均情节数，相比$\lambda=0$需要的情节数，均有所降低. $\lambda=0.5$时达到最低，平均需要 47.28 个情节就能学习成功. 表 2 看出，学习成功需要的情节数，与λ取值没有严格的对应关系，这或许是受到随机性的影响. 总体上看，λ以 0.5 附近为中心，往两端学习成功需要的平均情节数有逐渐增大的趋势.

6.3　实验讨论

上述实验可以看出，ALRI-SGD 方法中一个核心问题是历史梯度折扣率λ的大小对收敛过程的影

响.设某时刻的随机梯度为 g,算法经过 n 个时间步迭代后,g 的累加量为

$$(1+\lambda+\lambda^2+\cdots+\lambda^{n-1})g=\left(\frac{1-\lambda^n}{1-\lambda}\right)g,$$

如果希望经过 n 个时间步迭代后,梯度 g 的累加量是 g 的 m 倍,则应设置 $\lambda\approx(m-1)/m$.实际应用中,需要设置梯度累加量的大小,仍然需要具体问题具体分析.

7 结束语

基于值函数逼近的在线强化学习方法中,使用SGD方法更新值函数的权重,即在负梯度方向上更新模型参数,最小化损失函数.针对SGD方法在大规模状态空间强化学习任务中收敛慢甚至发散问题,本文提出ALRI-SGD方法,改进了随机梯度下降方式并进行了学习率优化设计.据此设计的算法能动态匹配目标函数的参数在不同维度上的梯度差异,并使学习率自动更新以适应不同维度的数据特征,较好地降低了优化震荡,提高了算法收敛效率和收敛稳定性.已有的理论分析和实验显示,ALRI-SGD方法是一个有价值的研究方向.

本文的工作仍存在一些问题有待进一步研究和完善,包括:实验中算法收敛性能对历史梯度折扣率 λ 敏感,λ 的设置尚需要深入研究;在ALRI-SGD方法的收敛性证明中,凸函数的假定给其应用带来理论上限制.

参 考 文 献

[1] Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Machine Learning, 1992, 8(3-4): 225-227

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

[3] Fu Qi-Ming, Liu Quan, Wang Hui, et al. A novel off policy $Q(\lambda)$ algorithm based on linear function approximation. Chinese Journal of Computers, 2014, 37(3): 677-686(in Chinese)
(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. 计算机学报, 2014, 37(3): 677-686)

[4] Wei Ying-Zi, Zhao Ming-Yang. A reinforcement learning-based approach to dynamic job-shop scheduling. Acta Automatica Sinica, 2005, 31(5): 765-771(in Chinese)
(魏英姿, 赵明扬. 一种基于强化学习的作业车间动态调度方法.自动化学报, 2005, 31(5): 765-771)

[5] Ipek E, Mutlu O, Martnez J F, et al. Self-optimizing memory controllers: A reinforcement learning approach. ACM SIGARCH Computer Architecture News, 2008, 36(3): 39-50

[6] Tesauro G. TD-Gammon, a self-teaching backgammon program, achieves master-level play. Neural Computation, 1994, 6(2): 215-219

[7] Kocsis L. Bandit based Monte-Carlo planning//Proceedings of the 17th European Conference on Machine Learning. Berlin, Germany, 2006: 282-293

[8] Dauphin Y N, Pascanu R, Gulcehre C, et al. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization//Proceedings of the International Conference on Neural Information Processing Systems. Kuching, Malaysia, 2014: 2933-2941

[9] Robbins H, Monro S. A Stochastic Approximation Method. Herbert Robbins Selected Papers. New York, USA: Springer, 1985: 102-109

[10] Kirkpatrick S, Gelatt C D, Vecchi M P. Optimization by simulated annealing. Science, 1983, 220(4598): 671-680

[11] Fabian V. On asymptotic normality in stochastic approximation. Annals of Mathematical Statistics, 1968, 39(4): 1327-1332

[12] Qian N. On the momentum term in gradient descent learning algorithms. Neural Networks, 1999, 12(1): 145-151

[13] Nesterov Y. A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$. Soviet Mathematics Doklady, 1983, 269: 543-547

[14] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research, 2011, 12(7): 257-269

[15] Dean J, Corrado G S, Monga R, et al. Large scale distributed deep networks//Proceedings of the International Conference on Neural Information Processing Systems. Doha, Qatar, 2012: 1223-1231

[16] Pennington J, Socher R, Manning C. Glove: Global vectors for word representation//Proceedings of the Conference on Empirical Methods in Natural Language Processing. Doha, Qatar, 2014: 1532-1543

[17] Puterman M L. Markov decision processes: Discrete stochastic dynamic programming. Technometrics, 2009, 37(3): 353-353

[18] Bertsekas D P, Tsitsiklis J N, Volgenant A. Neuro-dynamic programming. Encyclopedia of Optimization, 1996, 27(6): 1687-1692

[19] Busoniu L, Babuska R, Schutter B D, et al. Reinforcement Learning and Dynamic Programming Using Function Approximators. Florida, USA: CRC Press, 2010

[20] Lin Long-Ji. Self-improvement based on reinforcement learning, planning and teaching//Birnbaum L A, Collins G C eds. Machine Learning Proceedings. San Francisco, USA: Morgan Kaufmann, 1991: 323-327

JIN Hai-Dong, Ph. D. candidate. His research interests include beep reinforcement learning, reinforcement learning and deep learning.

LIU Quan, postdoctoral, professor, Ph. D. supervisor. His main research interests include beep reinforcement learning, automated reasoning and reinforcement learning.

CHEN Dong-Huo, Ph. D., lecturer. His research interests include software formal methods, reinforcement learning.

Background

Reinforcement Learning(RL) aims at finding an optimal policy, mapping from states to actions, via maximizing the returns. In face of large scale state-action space problem, online reinforcement learning generally uses the function approximation to represent the value function, and uses the stochastic gradient descent method to update the parameters of the approximation function step by step. Due to the difference in gradient rate of the objective function in different dimensions, SGD may make the optimization goal converge to another extreme point. The ALRI-SGD method proposed in this paper improves the SGD method in two aspects: based on the prediction of the parameters, the historical gradient information is used to calculate the update gradient of the current time step. Based on the historical gradient information of different dimensions, the learning rate of each dimension is dynamically calculated. These improvements reduce the optimization of oscillation, and the convergent efficiency and convergence stability of the algorithm are significantly improved.

This paper is supported by the National Natural Science Foundation of China (61772355, 61702055, 61502323, 61502329), the Jiangsu Province Natural Science Research University Major Projects (17KJA520004), the Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, the Jilin University (93K172014K04, 93K172017K18), the Suzhou Industrial application of basic research program part (SYG201422), the Jiangsu College Natural Science Research Key Program(17KJA520004), and the Suzhou Key Industries Technological Innovation-Prospective Applied Research Project (SYG201804), the Program of the Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) (KJS1524).