深度 Q 学习的二次主动采样方法

赵英男 [1] 刘 鹏 [1] 赵 巍 [1] 唐降龙 [1]

摘 要 实现深度 Q 学习的一种方式是深度 Q 网络 (Deep Q-networks, DQN). 经验回放方法利用经验池中的样本训练深度 Q 网络, 构造经验池需要智能体与环境进行大量交互, 这样会增加成本和风险. 一种减少智能体与环境交互次数的有效方式是高效利用样本. 样本所在序列的累积回报对深度 Q 网络训练有影响. 累积回报大的序列中的样本相对于累积回报小的序列中的样本更能加速深度 Q 网络的收敛速度, 并提升策略的质量. 本文提出深度 Q 学习的二次主动采样方法. 首先, 根据序列累积回报的分布构造优先级对经验池中的序列进行采样. 然后, 在已采样的序列中根据样本的 TD-error (Temporal-difference error) 分布构造优先级对样本采样. 随后用两次采样得到的样本训练深度 Q 网络. 该方法从序列累积回报和 TD-error 两个方面选择样本, 以达到加速深度 Q 网络收敛, 提升策略质量的目的. 在 Atari 平台上进行了验证. 实验结果表明, 用经过二次主动采样得到的样本训练深度 Q 网络取得了良好的效果.

关键词 优先经验回放, TD-error, 深度 Q 网络, 累积回报

引用格式 赵英男, 刘鹏, 赵巍, 唐降龙. 深度 Q 学习的二次主动采样方法. 自动化学报, 2019, **45**(10): 1870−1882

DOI 10.16383/j.aas.2018.c170635

Twice Sampling Method in Deep Q-network

ZHAO Ying-Nan[1] LIU Peng[1] ZHAO Wei[1] TANG Xiang-Long[1]

Abstract One way of implementing the deep Q-learning is the deep Q-networks (DQN). Experience replay is known to train deep Q-networks by reusing transitions from a replay memory. However, an agent needs to interact with the environment lots of times to construct the replay memory, which will increase the cost and risk. To reduce the times of interaction, one way is to use the transitions more efficiently. The cumulative reward of an episode where one transition is obtained from has an impact on the training of DQN. If a transition is obtained from the episode which can get a big cumulative reward, it can accelerate the convergence of DQN and improve the best policy compared with the transition which is obtained from a small cumulative reward′s episode. In this paper, we develop a framework for twice active sampling method in the deep Q-learning. First of all, we sample the episodes from the replay memory based on their cumulative reward. Then we sample the transitions from the selected episodes based on their temporal-difference error (TD-error). In the end, we train the DQN with these transitions. The method proposed in this paper not only accelerates the convergence of the deep Q-learning, but also leads to better policies because we replay transitions based on both TD-error and cumulative reward. By analyzing the results on Atari games, the experiments have shown that our method can achieve good results.

Key words Prioritized experience replay, temporal-difference error (TD-error), deep Q-networks (DQN), cumulative reward

Citation Zhao Ying-Nan, Liu Peng, Zhao Wei, Tang Xiang-Long. Twice sampling method in deep Q-network. *Acta Automatica Sinica*, 2019, **45**(10): 1870−1882

强化学习是解决序贯决策问题的一种途径[1]. 智能体在环境中做出动作, 得到环境的奖励, 根据奖励值调整自己的策略, 目的是最大化累积回报. 智能体通过这样不断与环境进行交互, 学习到最优策略. 从与环境的交互中学习是自然界有机生物体主要学习方式之一[2]. 智能体通过强化学习方式, 在极少先验知识条件下可习得最优策略.

强化学习主要包含两类算法[3−4], 一类是基于值函数估计的方法, 如 Q-learning[5] 和 SARSA[6] 等, 在智能体与环境的不断交互中得到最优值函数. 这类方法的好处是过程简单, 易于实现, 但是无法解决连续动作空间的问题. 另外一类是基于策略表示的方法, 也称为策略梯度法[7], 其中典型方法包括 REINFORCE 方法[7]、Actor critic 方法[8] 等, 其主要思想是将参数化策略与累积回报联系起来, 不断

收稿日期 2017-11-13 录用日期 2018-04-16
Manuscript received November 13, 2017; accepted April 16, 2018
国家自然科学基金 (61671175, 61672190) 资助
Supported by National Natural Science Foundation of China (61671175, 61672190)
本文责任编委 魏庆来
Recommended by Associate Editor WEI Qing-Lai
1. 哈尔滨工业大学计算机科学与技术学院模式识别与智能系统研究中心 哈尔滨 150001
1. Pattern Recognition and Intelligent System Research Center, School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001

优化得到最优策略. 这类方法可以解决连续动作空间的问题, 但是容易收敛到局部最优解中. 这两类算法是强化学习的经典算法, 仅能够解决一些小规模问题, 在面对现实世界中大规模, 复杂的问题时往往束手无策[3], 这限制了强化学习的发展.

强化学习与神经网络的首次结合可以追溯到 1994 年, Tesauro[9] 利用神经网络将近似值函数应用到了 Backgammon 游戏上, 并取得了很好的效果. 近年来, 深度神经网络与强化学习相结合, 形成了深度强化学习方法. 利用深度神经网络[10−11] 的泛化能力, 特征提取能力, 在许多领域取得了突破[12]. 在人机对抗领域, 结合深度强化学习和蒙特卡罗树搜索方法[13] 的围棋智能体 AlphaGo[14−16] 先后战胜了世界围棋冠军李世石、柯洁; 将动作值函数用深度神经网络的方式近似, 同时使用 Q-learning 更新规则来迭代动作值函数的方法称为深度 Q 学习. 实现深度 Q 学习的一种方式是深度 Q 网络. 使用深度 Q 网络[17−18] 训练的智能体在视频游戏上的表现达到或者超过了人类专家的水平; 在自动导航领域, 深度强化学习中的 A3C 方法结合辅助任务来帮助智能体更快实现在未知地图中的导航[19], 其中辅助任务包括深度预测和环形检测, 由循环神经网络[20] 实现; 在机器翻译领域, 文献 [21] 提出了一种对偶学习机制, 结合强化学习中的策略梯度法, 提升了翻译模型的性能; 在机器人控制领域, 文献 [22−23] 将深度学习与策略梯度结合提出了深度确定性策略梯度法, 解决了连续动作空间的控制问题.

目前仍存在一些因素制约深度强化学习的应用: 1) 智能体不具备不同任务之间的泛化能力[24], 在面对全新的任务时, 需要重新进行训练. 2) 模型稳定性和收敛性难以得到保证[25]. 3) 智能体为了能够适应环境往往需要与环境进行大量交互, 每次交互除了会增加时间和成本, 还会带来风险. 如何减少交互次数是本文要研究的问题.

减少智能体与环境的交互次数的一种思想是高效地利用已有的样本. 具体方式有: 1) 采用迁移学习的方式[26], 复用训练样本[27] 或策略[28], 将源任务中的样本或策略迁移到目标任务中, 这种方法的挑战在于难以衡量源任务和目标任务的相似性, 盲目迁移反而会降低效率. 2) 根据已经采样得到的样本对环境进行建模[29−30], 然后利用建立的模型产生样本, 减少智能体与环境交互次数. 这种方法对环境模型的准确性要求较高. 3) 在智能体训练时主动选择[31−33] 那些对训练有促进作用的样本, 典型方法有优先经验回放[31], 利用样本训练产生的误差 (Temporal difference error, TD-error) 作为优先级, 提高已采集样本的利用效率, 加快收敛.

哪些样本对训练有更大的促进作用, 如何从已采集的样本中选择这样的样本是本文研究的问题. 智能体执行一系列动作, 得到回报, 形成经验, 从经验中选择样本训练智能体的强化学习采样方法称作经验回放法[33]. 在强化学习中, 智能体从失败的或效果不好的经验中选择样本与从成功的或效果好的经验中选择样本都能得到优化的策略, 但是从成功的或效果好的经验中选择样本可以加速学习的过程. 本文在分析样本所在序列获得的累积回报对深度强化学习效果影响的基础上, 提出了二次主动采样方法. 首先, 计算经验池中序列的累积回报, 根据序列累积回报的分布对序列进行优先采样, 然后, 在被采样的序列中以 TD-error 的分布对样本进行优先采样. 两次采样在分别考量累积回报大的序列中的样本对学习的促进作用和 TD-error 大的样本对 Q 网络的收敛加速作用的同时, 在经验池中累积回报小的序列中的样本和 TD-error 值小的样本以较小的概率被采样, 从而保证了训练样本的多样性. 在 Atari 2600 视频游戏平台上进行验证, 实验结果表明, 本文方法加快了深度 Q 网络 (Deep Q-network, DQN) 的收敛速度, 提高了经验池中样本的利用效率, 也提升了最优策略的质量.

1 相关工作

1.1 强化学习

强化学习解决的是序贯决策问题, 整个过程可以用马尔科夫决策过程 (Markov decision process, MDP) 来进行建模: 智能体在离散时间步 $t = 1, 2, 3, \cdots$ 内与环境进行交互, 在每个时间步 t, 智能体得到环境状态的表示 $S_t \in S$, 其中 S 是所有状态的集合, 基于当前状态表示 S_t 选择一个动作 $A_t \in A(S_t)$, $A(S_t)$ 是状态 S_t 可采取动作的集合. 一个时间步后, 智能体得到了一个标量奖励 $R_{t+1} \in \mathbf{R}$, 并且到达一个新的状态 S_{t+1}. 这个过程一直持续下去, 直到智能体转移到终止状态[1]. 智能体从初始状态转移到终止状态这一系列状态转移可以被描述成 $\langle S_1, A_1, S_2, R_1\rangle, \cdots, \langle S_{t-1}, A_{t-1}, S_t, R_{t-1}\rangle$ 这样一个序列, 这样一个序列在强化学习中被称为一个周期 (Episode). 从 t 时刻开始, 到 T 时刻终止, 智能体在一个周期中获得的累积回报为:

$$G_t = \sum_{t'=t}^{T} \gamma^{t'-t} R_{t'+1} \tag{1}$$

其中, $0 \le \gamma \le 1$, 称为折扣因子, 折扣因子的作用是对不同时间步获得的奖励赋予不同的权重, 使得距离当前时间步越远的奖励权重越小, 强化学习的任务是找到一个策略使得 G_t 的值最大. 定义动作值函

数为:

$$q(s,a)=\mathrm{E}[G_t|S_t=s,A_t=a,\pi] \tag{2}$$

式 (2) 表示智能体处于状态 s, 采用动作 a, 然后执行策略 π 可以获得的累积期望回报, 定义动作值函数更新公式为:

$$q(s,a)\leftarrow q(s,a)+\alpha[r+\gamma\max_{a'}q(s',a')-q(s,a)] \tag{3}$$

利用式 (3) 不断迭代, 最终 $q(s,a)$ 将收敛到 $q^*(s,a)$, 即最优动作值函数[6]. 通过迭代式 (3) 来得到最优动作值函数的方法被称为 Q-learning. 为了使其具有泛化能力, 扩大其应用范围, 将深度神经网络和 Q-learning 结合起来, 形成深度 Q 网络[18].

1.2 深度 Q 学习

深度 Q 网络 (Deep Q-networks, DQN) 是一种典型的深度 Q 学习方法[34]. DQN 的特点有: 1) DQN 充分发挥了深度神经网络强大的特征提取能力和强化学习适应未知环境的能力, 实现了一种端到端的训练方式, 同时仅仅需要极少的先验知识, 便能够在很多复杂任务中表现出色. 2) 利用经验回放和目标网络两个技术保证了在使用非线性函数逼近器的情况下, 动作值函数的收敛. 3) 使用相同的超参数, 网络结构和算法能够在多个不同的任务中表现出色, 算法具有较强通用性.

经验回放[28] 是 DQN 训练的一种典型方法. 经验回放方法打破了训练样本间的相关性, 保证了值函数的收敛. 该方法将智能体与环境交互产生的样本保存下来, 形成一个经验池, 每次训练时随机从经验池中选择若干个样本训练 Q-网络, 每个样本是 $\langle S_t,A_t,S_{t+1},R_t\rangle$ 四元组, 利用这个样本, 定义与该样本对应的第 i 次迭代的目标动作值函数为:

$$y_i=R_t+\gamma\max_{a'}q(S_{t+1},a',\boldsymbol{\theta}^-) \tag{4}$$

其中, $\boldsymbol{\theta}^-$ 代表目标值函数网络的参数. 此时当前神经网络计算出来的动作值函数为 $q(S_t,A_t;\boldsymbol{\theta})$. 定义损失函数 $L_i(\boldsymbol{\theta})=(y_i-q(S_t,A_t;\boldsymbol{\theta}))^2$, 利用梯度下降法不断更新权重 $\boldsymbol{\theta}$, 完成 DQN 的训练, 得到最优动作值函数.

文献 [35] 中提出了 Double DQN, 解决了 DQN 中对动作值函数过估计的问题; 文献 [36] 中提出对偶神经网络结构, 将对动作值函数的估计分解成状态值函数估计和优势值函数估计两个部分, 能够学习到更好的策略; 文献 [31] 针对等概率从经验池中抽取样本的操作提出了改进, 利用每个样本训练产生的 TD-error, 即目标值和估计值的误差, 来表示样本的重要性, 为经验池中样本赋予不同的优先级, 提升了 DQN 算法的表现.

1.3 主动采样问题

在深度强化学习算法训练过程中, 为了减少训练样本间的相关性, 常采用经验回放的方法, 将智能体产生的样本存储起来, 在训练时随机选择若干样本进行训练, 提高了算法的收敛性和稳定性. 经验池中样本数量庞大, 如何从经验池中选择样本来提高深度强化学习的收敛速度是主动采样问题.

文献 [31] 中提出了优先经验回放方法, 将每个样本训练产生的 TD-error 作为样本的优先级, 优先级越大, 样本被选择的概率就越大. 该方法可以更有效率地回放样本, 而且能够提升最优策略的表现. 但是该算法仅从加速深度神经网络收敛速度的角度出发, 考虑样本的 TD-error 对训练的影响, 忽略了样本所在序列的累积回报对强化学习的作用.

本文提出二次主动采样方法. 首先根据经验池中序列的累积回报分布, 以更大的概率选择累积回报大的序列. 然后, 在被选择的序列中根据 TD-error 分布选择样本来训练 Q 网络. 该方法从累积回报的作用和深度神经网络误差梯度两个方面加速 DQN 的学习过程. 强化学习的目的是获得最优策略使累积回报最大化. 从累积回报小的序列中选择样本会使智能体以更大的概率避免错误, 而从累积回报大的序列中选择样本会使智能体以更大的概率获得更大的累积回报, 这与强化学习的目的一致. 本文方法并没有完全放弃对累积回报小的序列中样本的采样, 只是这类样本被采样的概率小, 从而保证了样本的多样性. 在 Atari 2600 视频游戏上的实验表明, 本文提出的方法提高了经验池中样本的利用效率, 同时也改善了最优策略的质量.

2 二次主动采样方法

本文提出的方法建立在累积回报大的序列中的样本以更大的概率对 DQN 学习有促进作用的基础之上. 首先, 在经验池中按序列组织样本, 然后, 根据各序列累积回报的分布生成序列采样优先级, 依据优先级在经验池中对序列采样. 最后, 在已选择的序列中根据 TD-error 分布生成样本采样优先级, 在序列中选择样本, 训练 DQN.

2.1 序列累积回报对训练的影响

在优先经验回放方法中[21], 算法以样本训练产生的 TD-error 作为优先级, 并没有考虑到样本所在序列取得的累积回报. 事实上, 在样本选择时考虑样本所在序列的累积回报是十分必要的. 强化学习是包括人类在内的高级智能体的一种基于行为的学习方式. 智能体产生的每一个序列是为了完成某件事做出的一次尝试, 回报就是对一系列行为取得结果的评价. 在智能体再次面对相同或者相似的任务时,

首先会回忆以前的经历, 如果能够搜索到达到目的的经历, 那么只要按照相同的策略再执行一次即可. 如果没有搜索到达到目的的经历, 也会更倾向于学习 (尝试) 那些较为接近目的的经历. 这表明在成功的经验中, 智能体在更多状态下都采用了有效动作, 这样只需要经过少量的改进, 就能达到目的. 本文提出的方法也源于此思想[37], 成功的或者接近成功的序列, 其中含有有效动作的样本就越多, 就应该被更多地回放, 从而提高 DQN 的学习速度, 提升学习到的策略的质量.

平衡杆 (Cartpole) 问题是强化学习中的经典问题. 如图 1 所示, 平衡杆由两部分构成, 方块是一个小车, 上面连接着一个杆, 假设不存在摩擦力, 平衡杆的目的是通过施加给小车向左或者向右的力维持上面杆的平衡. 每次做出动作, 杆子如果没有倒下, 就会获得值为 1 的奖励. 平衡杆问题动作空间大小为 2, 状态空间是无限的, 适合用来验证 DQN 算法. 接下来在 Cartpole 中进行两个说明性实验来验证假设.

1) 累积回报与有效动作数量的关系

本文提出的方法基于一个假设: 获得累积回报越大的序列, 其中含有有效动作的样本数量越多. 为了验证这一假设, 首先训练出一个已经收敛的网络模型 M^*, 即输入状态, 可以输出最优动作, 作为有效动作. 然后利用 DQN 算法训练另外一个网络模型 M, 训练过程中将智能体产生的序列保存下来, 同时记录下每个序列得到的累积回报. 在某一时刻对序列进行采样, 按照累积回报从小到大的顺序对序列进行排序. 第一次从后 50 % 的序列中进行采样, 也就是从累积回报较大的序列中选择样本, 第二次从前 50 % 的序列中进行采样, 也就是从累积回报较小的序列中选择样本. 两次采样的样本数量相同, 均为 500 000. 遍历每一个样本 $\langle s, a, s', r\rangle$, 将状态 s 输入到 M^* 中, 得到有效动作 a^*, 判断样本中的动作 a 和有效动作 a^* 是否相同.

实验结果表明, 在累积回报大的序列中取得的 500 000 个样本中, 含有有效动作的样本的数量为 297 428 个, 在累积回报小的序列中取得的样本中, 含有有效动作的样本的数量仅为 208 286 个. 前者的数量比后者多出 42.8 %.

2) 累积回报对 DQN 训练的影响

另外, 结合优先经验回放方法[31], 对于平衡杆问题验证不同累积回报对 DQN 学习的影响. 程序采用 ϵ-贪心策略来控制智能体探索与利用的平衡问题, 探索因子设置为 0.1. 智能体与环境最大交互次数设置为 100 000, 序列采样数量设为 10. 若智能体产生的最近一百个序列累积回报的均值达到 199, 即为收敛状态, 程序停止.

对三种情况进行实验:

情况 A. 经典的优先经验回放方法[31], 没有考虑序列回报.

情况 B. 将经验池中的 N 个序列按照累积回报大小从大到小排序, 在前 10 % 序列中随机选取 8 个序列, 从剩余 90 % 序列中随机选取 2 个序列, 再利用优先经验回放方法对这 10 个序列的样本进行采样.

情况 C. 将经验池中的 N 个序列按照累积回报大小从大到小排序, 在后 10 % 序列中随机选取 8 个序列, 在剩余 90 % 序列中随机选取 2 个序列, 再利用优先经验回放方法对这 10 个序列的样本进行采样.

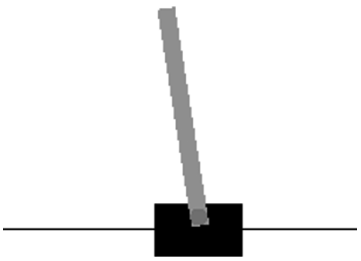


图 1　平衡杆环境示意图

Fig. 1　The diagram of cartpole

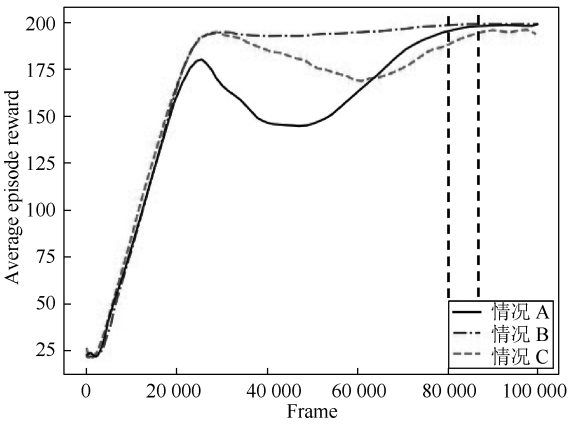


图 2　平衡杆在三种情况下的对比实验

Fig. 2　The cartpole comparison experiments in three cases

为了避免偶然性, 每种情况运行 10 次, 图 2 中展示的为取平均后的结果, 实验结果如图 2 所示. 图 2 的横坐标代表智能体与环境的交互次数, 纵坐标代表智能体产生的最近 100 个序列累积回报的均值. 下面对实验现象进行说明:

1) 从累积回报较小序列中样本学习的 DQN (情况 C) 没有达到最优策略.

2) 不考虑序列累积回报的 DQN (情况 A) 达到了最优策略, 但学习过程中波动较大.

3) 从累积回报较大序列中样本学习的 DQN (情

况 B) 最终达到最优策略, 而且学习过程中回报平稳, 学习过程中回报平稳带来的一个直接好处就是减少了训练过程中产生的风险.

4) 从累积回报较大的序列中选择样本的训练方法 (情况 B) 在 80 000 帧交互后使 DQN 达到收敛; 用优先经验回放方法 (情况 A) 选择样本在 89 000 帧交互后使 DQN 达到收敛.

这表明从累积回报大的序列中选择样本这种方法能够使 DQN 更快达到最优策略. 上述三种情况的对比实验表明, 累积回报大的序列中的样本对训练更有促进作用.

2.2 序列采样方法

从累积回报大的序列中选择样本, 会增加含有有效动作样本的数量, 进而加速 DQN 的收敛. 下面从 DQN 训练过程来解释含有有效动作的样本对 DQN 学习的促进作用. 经验池用 $E = \{l_1, l_2, \cdots\}$ 表示, 其中 $l_i = \{\langle S_1^i, A_1^i, S_2^i, R_1^i\rangle, \langle S_2^i, A_2^i, S_3^i, R_2^i\rangle, \cdots\}$ 为经验池中的第 i 个序列, $\langle S_j^i, A_j^i, S_{j+1}^i, R_j^i\rangle \in l_i$ 是序列 l_i 中的一个样本. 当智能体处于状态 S_t 时, 如果 E 中有两个序列 l_m, l_n 中的样本 $\langle S_t, A_t^m, S_{t+1}^m, R_t^m\rangle$ 和 $\langle S_t, A_t^n, S_{t+1}^n, R_t^n\rangle$ 均可以回放, 根据式 (3) 回放 $\langle S_t, A_t^m, S_{t+1}^m, R_t^m\rangle$ 使动作值函数收敛到 $q^*(S_t, A_t^m)$, 回放 $\langle S_t, A_t^n, S_{t+1}^n, R_t^n\rangle$ 使动作值函数收敛到 $q^*(S_t, A_t^n)$.

在 k 时刻, 智能体处于状态 S_k, 选择样本 $\langle S_k, A_k, S_t, R_k\rangle$ 进行回放, 即在 k 时刻, 采用动作 A_k 使智能体的状态由 S_k 转移到 S_t, 动作值函数更新为:

$$q(S_k, A_k) \leftarrow q(S_k, A_k) + \alpha[R_k + \gamma \max_{A'} q(S_t, A') - q(S_k, A_k)] \quad (5)$$

为了使 $q(S_k, A_k)$ 更快收敛, 就需要 $\max_{A'} q(S_t, A')$ 这项准确. 如果在 t 时刻的两个样本 $\langle S_t, A_t^m, S_{t+1}^m, R_t^m\rangle$ 和 $\langle S_t, A_t^n, S_{t+1}^n, R_t^n\rangle$ 对应的最优动作值函数满足 $q^*(S_t, A_t^m) > q^*(S_t, A_t^n)$. 则式 (5) 可以改写为 $q(S_k, A_k) \leftarrow q(S_k, A_k) + \alpha[R_{k+1} + \gamma q^*(S_t, A_t^m) - q(S_k, A_k)]$, 这表明如果回放序列 l_m 中的样本 $\langle S_t, A_t^m, S_{t+1}^m, R_t^m\rangle$ 会使 $q(S_k, A_k)$ 加速收敛.

DQN 训练的损失函数为 $L(\boldsymbol{\theta}) = (y - q(s, a; \boldsymbol{\theta}))^2$, 其中 $y = r + \gamma \max_{a'} q(s', a', \boldsymbol{\theta}^-)$, 其中 y 被称为目标值, 与监督学习中目标值是固定的不同, 在 DQN 中目标值是不断变化的, 而且有很多时候目标值不准确, 这就造成了 DQN 收敛速度慢, 训练过程不稳定等问题. 本文提出的二次主动采样方法, 首先从经验池中选择累积回报大的样本序列. 然后从样本序列中选择 TD-error 大的样本进行训练, 能增加参与训练的含有有效动作样本的数量. 两次采样中的第一次采样保证了 DQN 训练的损失函数中目标值的稳定性, 第二次采样加速了动作值函数迭代的收敛速度.

尽管累积回报大的样本对 DQN 训练有更大的促进作用, 但是样本多样性也是保证 DQN 训练收敛的一个重要方面. 本文根据序列累积回报的分布构造对经验池中序列的采样概率. 设经验池为 $E = \{l_1, l_2, \cdots, l_m\}$, 其中 $l_i = \{\langle S_1^i, A_1^i, S_2^i, R_1^i\rangle, \langle S_2^i, A_2^i, S_3^i, R_2^i\rangle, \cdots\}$ 为经验池中的第 i 个序列, $\langle S_j^i, A_j^i, S_{j+1}^i, R_j^i\rangle \in l_i$ 是序列 l_i 中的一个样本. 序列 l_i 中的样本数量用 $|l_i|$ 表示. 第 i 个序列取得的累积回报为 $G_i = \sum_{t=0}^{T} \gamma^t R_{t+1}^i$, 令 $p_i = G_i + \epsilon$ 为第 i 个序列的优先级, 其中 ϵ 是一个极小的正数, 目的是保证所有序列的优先级均大于 0. l_i 被采样的概率为:

$$P(i) = \frac{p_i^\alpha}{\sum_{k=1}^{m} p_k^\alpha} \quad (6)$$

其中, α 决定了优先级所占比例, 当 $\alpha = 0$ 时, 对序列的采样退化为均匀采样. 用 $P(i), i \in \{1, 2, \cdots, m\}$ 对经验池 E 中的序列采样, 使累积回报大的序列以更大的概率被选择, 同时累积回报小的序列也有机会被选择. 在保证累积回报大的序列中样本以更大的概率参加 DQN 训练的同时, 也保证了样本的多样性.

2.3 从序列中选择样本

经过对经验池 E 中的序列 l_i 以 $P(i)$ 的概率采样后, 被选择的序列用 $\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}_n$ 表示. 这些序列中的样本构成了一个小型经验池 $\tilde{E} = \{\tilde{l}_1, \tilde{l}_2, \cdots, \tilde{l}_n\}$. 在经验池 $\tilde{E}$ 中样本的总数为 $N = \sum_{k=1}^{n} |\tilde{l}_k|$. 其中样本 $\tilde{e}_u = \langle \tilde{S}_t, \tilde{A}_t, \tilde{S}_{t+1}, \tilde{R}_t\rangle, u \in \{1, 2, \cdots, N\}$ 在 DQN 训练中的 TD-error 为:

$$\tilde{\delta}_u = \tilde{R}_t + \gamma \max_{a'} q(\tilde{S}_{t+1}, a') - q(\tilde{S}_t, \tilde{A}_t) \quad (7)$$

样本 $\tilde{e}_u$ 的优先级为 $\tilde{p}_u = |\tilde{\delta}_u| + \epsilon$. 对经验池 $\tilde{E}$ 中样本 $\tilde{e}_u$ 的采样概率为:

$$\tilde{P}(u) = \frac{\tilde{p}_u^\alpha}{\sum_{k=1}^{N} \tilde{p}_k^\alpha} \quad (8)$$

以 $\tilde{P}(u)$ 概率对经验池 $\tilde{E}$ 中的样本进行采样来训练 DQN, 加速 DQN 的收敛速度.

本文提出的二次主动采样方法, 首先以序列的形式组织经验池 E. 根据序列累积回报的分布对经

验池 E 中的序列进行采样, 形成一个小型的经验池 $\tilde{E}$, 然后在经验池 $\tilde{E}$ 中根据样本的 TD-error 分布进行第二次采样, 得到用于 DQN 训练的样本. 第一次采样以累积回报作为序列选择的准则, 使得到的经验池 $\tilde{E}$ 中的样本中以更大的概率来自于累积回报大的序列. 在第二次采样中, 以 DQN 训练的 TD-error 为依据, 使能够加速 DQN 收敛的样本以更大的概率参加 DQN 训练. 本文提出的二次主动采样方法从样本所在序列的性质 (累积回报) 和样本在 DQN 迭代中加速作用 (TD-error) 两方面出发来选择样本. 用本文方法选择的样本既可以获得最优策略, 也保证了 DQN 的收敛速度.

本文提出的二次主动采样方法首先以累积回报为标准从样本池中选择序列, 再以 TD-error 为标准从样本序列中选择样本. 两次采样的顺序是不能交换的. 如果忽略样本序列导致的累积回报, 先从样本池中选择 TD-error 值大的样本来更新动作值函数, 这样的样本所在的样本序列未必使累积回报值得到优化. Q-learning 的目标是在某个状态下得到一个动作使累积回报最大化, 本文提出的二次采样方法的基本思想是在累积回报大的样本序列中选择 TD-error 大的样本会使 Q-learning 以更快的收敛速度获得最优策略.

2.4 算法总结

算法 1. 二次主动采样算法

输入. 训练频率 K, 采样序列数量 N, 奖励折扣因子 γ, 样本批量大小 k, 指数 α, 学习率 η, 训练终止时间步 T, 算法初始经验池容量 $size$, 重要性采样权重 β, 目标 Q 网络更新频率 L

输出. Q 网络参数

初始化. 经验池 E 为空, 初始化 Q 网络和目标 Q 网络参数为 $\boldsymbol{\theta}$ 和 $\boldsymbol{\theta}^-$, 优先级 $\tilde{p}_1 = 1$, 单个序列存储经验池 h

选择动作 $A_0 \sim \pi_{\boldsymbol{\theta}}(S_0)$ ($\epsilon-$greedy)
for $t = 1$ to T do:
  观察到状态 S_t, 获得奖励 R_{t-1}, 是否到达终止状态标记 done
  将样本 $(S_{t-1}, A_{t-1}, S_t, R_{t-1})$ 存储到经验池 h 中, 并赋予当前经验池中最大的优先级 $\tilde{p}_t = \max_{i<t}\tilde{p}_i$
  if done:
    将 h 保存到 E 中, 清空 h
  if $t \,\%\, K == 0$ and $t > size$:
    计算得到经验池 E 中序列总数 M
    for $j = 1$ to M do:
      序列 l_j 累积回报 $G_j = \sum_{t=0}^{T}\gamma^t R_{t+1}$
      序列 l_j 优先级 $p_j = G_j + \epsilon$
    end for
    初始化临时经验池 $\tilde{E}$
    for $= 1$ to N do:
      采样序列 $l \sim P(l) = p_l^\alpha / \sum_i p_i^\alpha$
      将序列 l 中样本放入经验池 $\tilde{E}$ 中
    end for
    计算得到经验池 $\tilde{E}$ 含有样本数量 $\tilde{N}$
    for $j = 1$ to k do:
      采样样本 $e \sim \tilde{P}(e) = \tilde{p}_e^\alpha / \sum_i \tilde{p}_i^\alpha$
      计算样本重要性系数 $\omega_e = (\tilde{N}\tilde{p}_e)^{-\beta}/\max_i\omega_i$
      计算 TD-error: $\delta_e = R_e + \gamma q(S_e^{'}, \text{argmax}_{a'} q(S_e^{'}, a', \boldsymbol{\theta}), \boldsymbol{\theta}^-) - q(S_e, A_e, \boldsymbol{\theta})$
      更新样本优先级: $\tilde{p}_e \leftarrow |\delta_e| + \epsilon$
      累积梯度 $\Delta \leftarrow \Delta + \omega_e\delta_e\nabla_{\boldsymbol{\theta}} q(S_e, A_e, \boldsymbol{\theta})$
    end for
    更新网络权重 $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta\Delta, \Delta = 0$
  if $t \,\%\, L == 0$:
    $\boldsymbol{\theta}^- = \boldsymbol{\theta}$
  选择下一个动作 $A_t \sim \pi_{\boldsymbol{\theta}}(S_t)(\epsilon-\text{greedy})$
end for

为了方便对序列的采样, 算法在第 3～7 步将整个序列保存到经验池中. 在第 8 步中开始训练, 训练前判断当前时间步是否大于 $size$, 这里的 $size$ 指的是经验池初始化容量, 目的是在经验池中存储一些样本, 方便接下来的训练. 第 9～27 步为二次采样过程. 其中, 第 9～19 步为算法的第一部分. 首先计算每个序列的累积回报, 再计算优先级, 最后得到序列被采样的概率. 第 15～19 步根据优先级采样 N 个序列构成小经验池 $\tilde{E}$. 第 19～27 步为样本采样和训练过程. 第 23 步计算 TD-error 利用的是 Double DQN[18] 算法, 这种计算方式可以减少 DQN 对值函数的过估计. 第 28 步是更新目标网络参数过程, 是一种保证 DQN 收敛的一种方法, 将目标值网络和当前值网络分开, 使得目标值网络的权重在一段时间内保持不变, 从而使得目标值也会在一段时间内保持不变, 使得训练过程更加稳定.

除深度 Q 学习以外, 还有许多强化学习方法如 SARSA 算法[38]、Actor critic 算法[39] 都可以采用经验回放方式来提升样本利用率, 加快收敛速度. 经验回放就是将样本保存在经验池中, 训练时从经验池中选择样本. 本文提出的二次主动采样方法也适用于其他涉及到经验回放的强化学习的样本选择过程.

2.5 时间复杂度分析

本文提出的算法主要分成两个部分, 首先是序列采样, 主要思想是序列优先级越大被采样概率越

大, 采用的主要方法如下: 对每个序列的优先级进行加和记为 sum, 序列优先级最小值为 ϵ, 在 ϵ 到 sum 之间随机选一个数 random. 接下来遍历整个序列集合, 统计遍历到的序列的优先级之和, 如果大于 random, 则选择这个序列. 假设经验池中共有 M 个序列, 采样的序列数量为 N 个. 该方法需要遍历所有序列, 时间复杂度为 $\mathrm{O}(MN)$.

算法第二部分是对小经验池进行优先经验回放, 假设采样序列数量为 N, 平均每个序列含有 K 个样本, 因此小经验池的容量为 KN. 这里采用与第一部分相同的采样方法, 假设采样样本数量为 m. 可知时间复杂度为 $\mathrm{O}(mKN)$.

因此本文提出的算法总体时间复杂度为 $\mathrm{O}(MN) + \mathrm{O}(mKN))$. 由于经验池容量固定, 因此本文提出的方法运行时间在 DQN 训练过程中基本保持不变.

3 实验结果与分析

3.1 实验平台描述

本文在 openAI Gym (https://gym.openai.com) 上的 Atari 游戏平台进行实验. Atari 游戏平台[40] 有两个特点: 1) 仅仅以图像作为输入, 与人类玩游戏的方式相同. 2) 游戏种类多样, 包括射击类、动作类、策略类等, 适合验证算法的通用性和泛化能力. Atari 平台上的部分游戏界面如图 3 所示. Atari 平台是当前深度强化学习的主流测试平台.

实验硬件平台是 Dell-T630 工作站, 处理器为两个 Intel(R) Xeon(R) CPU E5-2650 v4, 显卡为 3 块 GTX-1080 TI 和 1 块 GTX-1070, 工作站内存 64 GB, 操作系统为 Ubuntu 16.04.

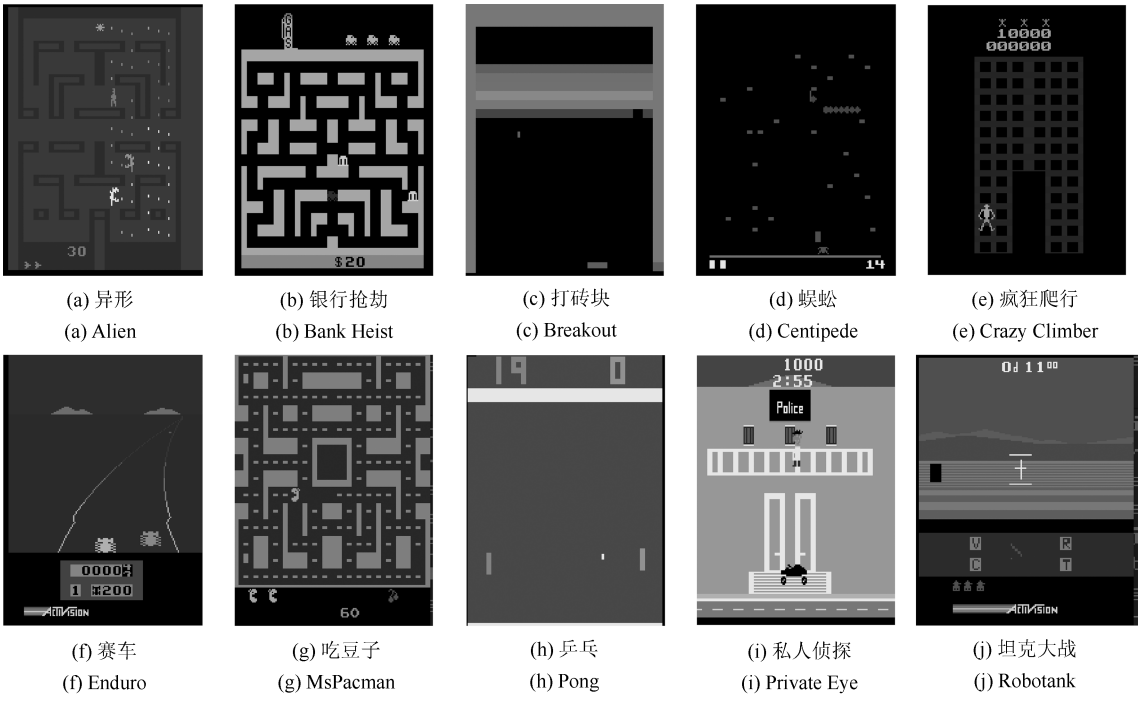
3.2 实验参数设置

本文提出的方法中有一个超参数需要设置, 即序列采样的数量, 序列采样数量过少, 会使样本失去多样性, DQN 训练发散; 采样过多则会使得被采样的序列构成小型经验池变大, 导致第二次采样时间变长, 算法运行时间变长. 为了选择适当的序列采样数量, 在平衡杆上用 6 个不同的序列采样数量进行实验. 每个实验运行 10 次. 算法的平均运行时间和平均收敛步数列于表 1.

表 1 平衡杆问题在不同采样序列数量下的平均运行时间和平均收敛步数

Table 1 Average convergent step numbers and consuming time using different sampling episode numbers in cartpole

采样序列数量	平均运行时间 (s)	平均收敛步数
8	455.7025	79 251.0
16	491.0820	71 950.0
24	498.1949	69 188.8
32	527.1340	68 543.8
40	541.2012	63 389.2
48	567.1340	64 344.3



(a) 异形 (a) Alien (b) 银行抢劫 (b) Bank Heist (c) 打砖块 (c) Breakout (d) 蜈蚣 (d) Centipede (e) 疯狂爬行 (e) Crazy Climber

(f) 赛车 (f) Enduro (g) 吃豆子 (g) MsPacman (h) 乒乓 (h) Pong (i) 私人侦探 (i) Private Eye (j) 坦克大战 (j) Robotank

图 3 Atari 游戏截图

Fig. 3 Screenshots of some games

由表 1 可以看出, 随着采样序列数量的增加, 算法运行时间逐渐变长, 平均收敛步数也逐渐减少, 但是采样序列数量大于 40 时, DQN 的平均收敛步数增加, 原因是当采样序列的数量过大时, 则会使得算法逐渐退化成为优先经验回放方法, 进而造成性能下降. 因此将采样序列数量设置为 40.

DQN 每次迭代前, 从已被采样的序列中选择样本的数量为 32. 折扣因子 γ 设置为 0.99. 探索因子设置为随着时间步数增加而分段递减形式, 如图 4 所示. 在训练开始阶段, 探索因子 ϵ 取较大值, 智能体以更大概率探索, 在训练后期, ϵ 取较小的值. 智能体以更大的概率利用已获得的策略.

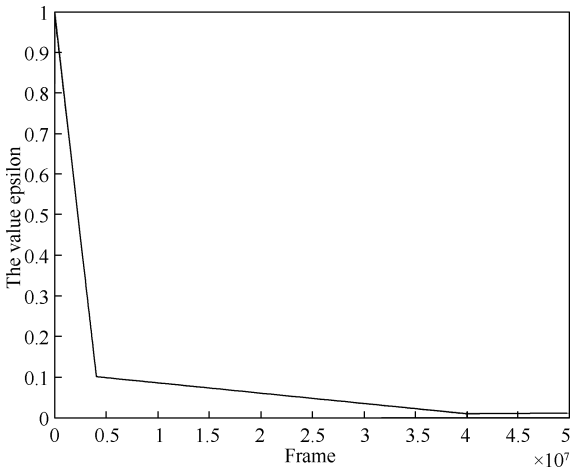


图 4　ϵ 值变化曲线

Fig. 4　The value of ϵ

本文提出的二次主动采样算法, 首先按照累积回报大小对序列进行采样构成一个较小的经验池, 再按照优先经验回放的方法在其中进行采样. 这种方法与优先经验回放的不同之处在于, 优先经验回放方法更加频繁地回放那些在整个经验池中 TD-error 值较大的样本, 而本文提出的方法更加关注那些所处序列获得的累积回报较大同时 TD-error 值也较大的样本, 这些样本是对训练更有促进作用的样本. 在进行对比实验之前, 首先选择蜈蚣 (Centipede) 这个游戏, 对其经验池中样本进行了分析, 训练曲线图如图 5 所示, 横坐标为交互次数, 纵坐标为平均累积回报. 分析时考虑样本所处序列的回报和样本 TD-error 两个方面. 将训练过程中某一时刻的经验池保存下来, 记录下每个样本的 TD-error 值和所处序列累积回报值, 将 TD-error 从大到小排序, 经验池中样本总数为 10^6, 取出 TD-error 值较大的前 3×10^5 个样本, 并得到其所处序列获得的累积回报, 这里分别记录时间步为 4×10^7 和 1.6×10^8 两个时刻的经验池, 如图 5 中黑色虚线所示. 将样本的累积回报分别绘制成分布图如图 6 (a) 和 (b).

图 6 中横坐标为样本所处序列获得的累积回报, 纵坐标为样本数量. 可以发现, 多数样本其所在序列累积回报较小, 同时也存在一部分样本所处序列累积回报较大, 即处于图中右侧部分的样本, 这部分的样本是对 DQN 训练更有促进作用的样本, 即具有 TD-error 值较大, 同时所处序列累积回报较大两种性质的样本. 实验结果表明具有这两种性质的样本在游戏训练不同时刻都是存在的. 打砖块 (Breakout) 和坦克大战 (Robotank) 也有相似的现象, 得出的样本分布情况如图 6 (c) 和 (d) 所示.

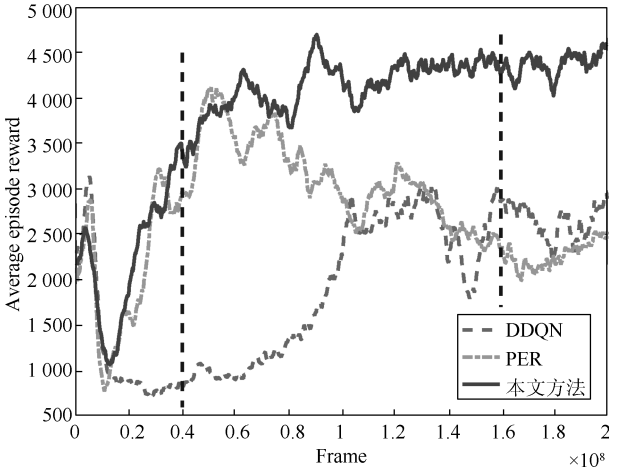


图 5　蜈蚣游戏训练曲线

Fig. 5　The training curve of centipede

本文提出方法首先从样本池中选择累积回报大的样本序列, 再从这些序列中选择 TD-error 大的样本对 DQN 训练以提高其收敛速度. 这个过程包含两次选择, 一次是对样本序列的选择, 另一次是在已选择的样本序列中对样本进行选择. 如果交换这两次选择的顺序, 先从样本池中选择 TD-error 大的样本, 再根据已选择样本所在序列的累积回报的大小对已选择样本进行二次筛选, 则不能达到提高 DQN 收敛速度的效果. 用平衡杆实验来验证样本采样顺序对 DQN 收敛性的影响. 对两种选择顺序分别进行 10 次实验, 每次实验最大交互步数为 100 000. 实验结果列于表 2. 实验结果表明, 本文提出方法的样本选择顺序取得了更好的收敛效果.

表 2　样本选择顺序对比实验 (平衡杆)

Table 2　The comparison experiments in different sampling order (cartpole)

样本选择顺序	10 次实验中的未收敛次数	平均收敛步数
累积回报-TD-error	1	61 024.0
TD-error-累积回报	5	63 010.0

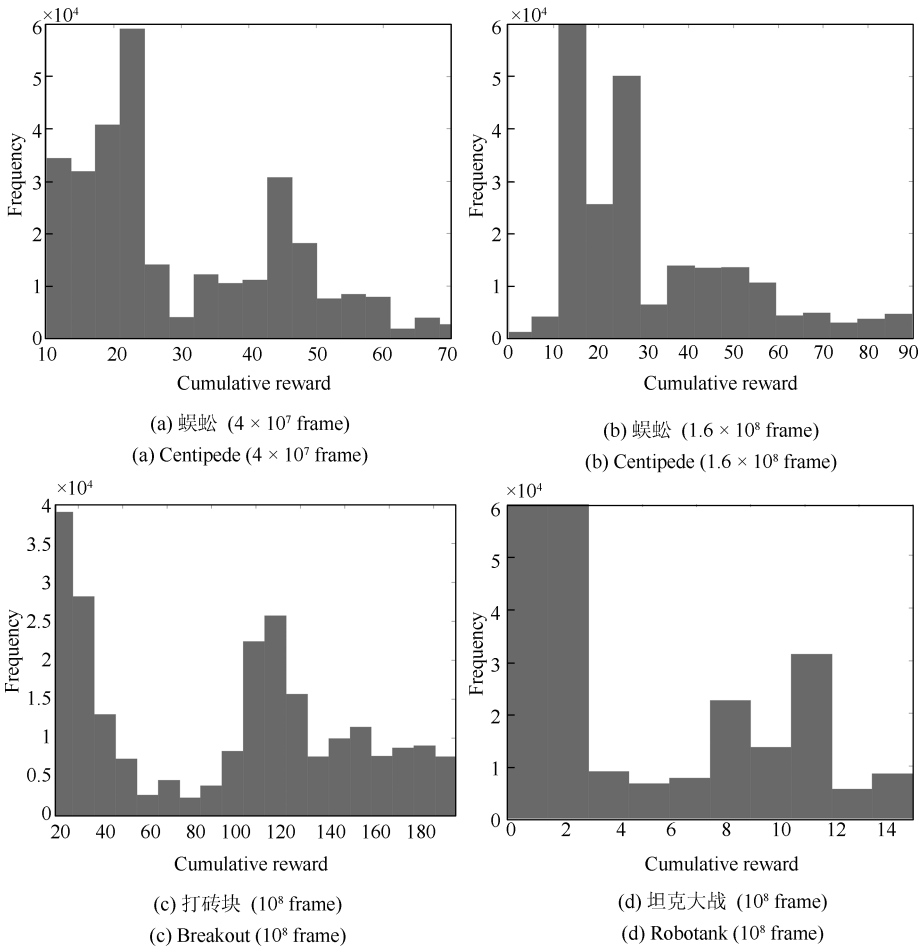


图 6 经验池样本分布图

Fig. 6 The distribution map of samples

3.3 对比实验

本文选择 Double DQN (DDQN)[35] 和优先经验回放 (Prioritized experience replay, PER)[31] 作为对比算法, 其中 DDQN 主要思想是在经验池中随机选择样本, 进行训练, PER 主要思想是将样本训练产生的 TD-error 作为优先级, 然后在经验池中有选择地采样训练, 本文方法则是在 PER 基础上增加了对序列采样的过程, 提升了样本的利用效率. 算法评价标准包括: 1) 算法收敛速度. 2) 算法得到最优策略的质量. 实验训练曲线如图 7 所示. 图 7 横坐标为交互次数, 纵坐标为平均累积回报. 由图 7 可以看出, 本文提出的方法与优先经验回放和 Double DQN 算法相比, 在多数游戏中能够使用更少的交互次数达到相同的平均回报, 在收敛速度上要优于这两个算法. 智能体学习到的策略可以通过模型参数得到, 训练过程中, 每 5×10^5 时间步保存一次模型参数, 找到训练曲线中最高点将那一时刻保存的模型参数作为最优策略, 然后利用文献 [35] 提出的评价方法, 也称为无动作评价 (No-op action) 来评价不同算法得到的最优策略的优劣. 具体做法在 1 ∼ 31 之间随机选一个数字 f, 在序列的前 f 帧不执行动作, 保证每个序列有不同的起始状态, 将探索因子 ϵ 设置成 0.05, 然后按照要测试的策略执行 100 个序列, 取序列平均得分, 用该得分评价策略的优劣. 由于每个游戏都有自己不同的分数计算方式, 为了使算法能够在不同游戏间进行比较, 利用式 (9) 对分数进行归一化:

$$score_{\text{normalized}} = \frac{score_{\text{agent}} - score_{\text{random}}}{|score_{\text{human}} - score_{\text{random}}|} \quad (9)$$

其中, $score_{\text{random}}$ 代表智能体随机采用动作得到的分数, $score_{\text{human}}$ 代表人类专家的得分, $score_{\text{agent}}$ 代表采用本文算法学习到的策略得到的分数. 人类专家和随机智能体的得分数据来自于文献 [7], DDQN 算法的得分来自于文献 [35], PER 算法的得分来自于文献 [31], 其中有些数据与实际测试不同, 以实际测试为准. 对归一化之后的分数进行统计得到表 3.

每个游戏的具体得分如表 4 所示, 每个游戏的

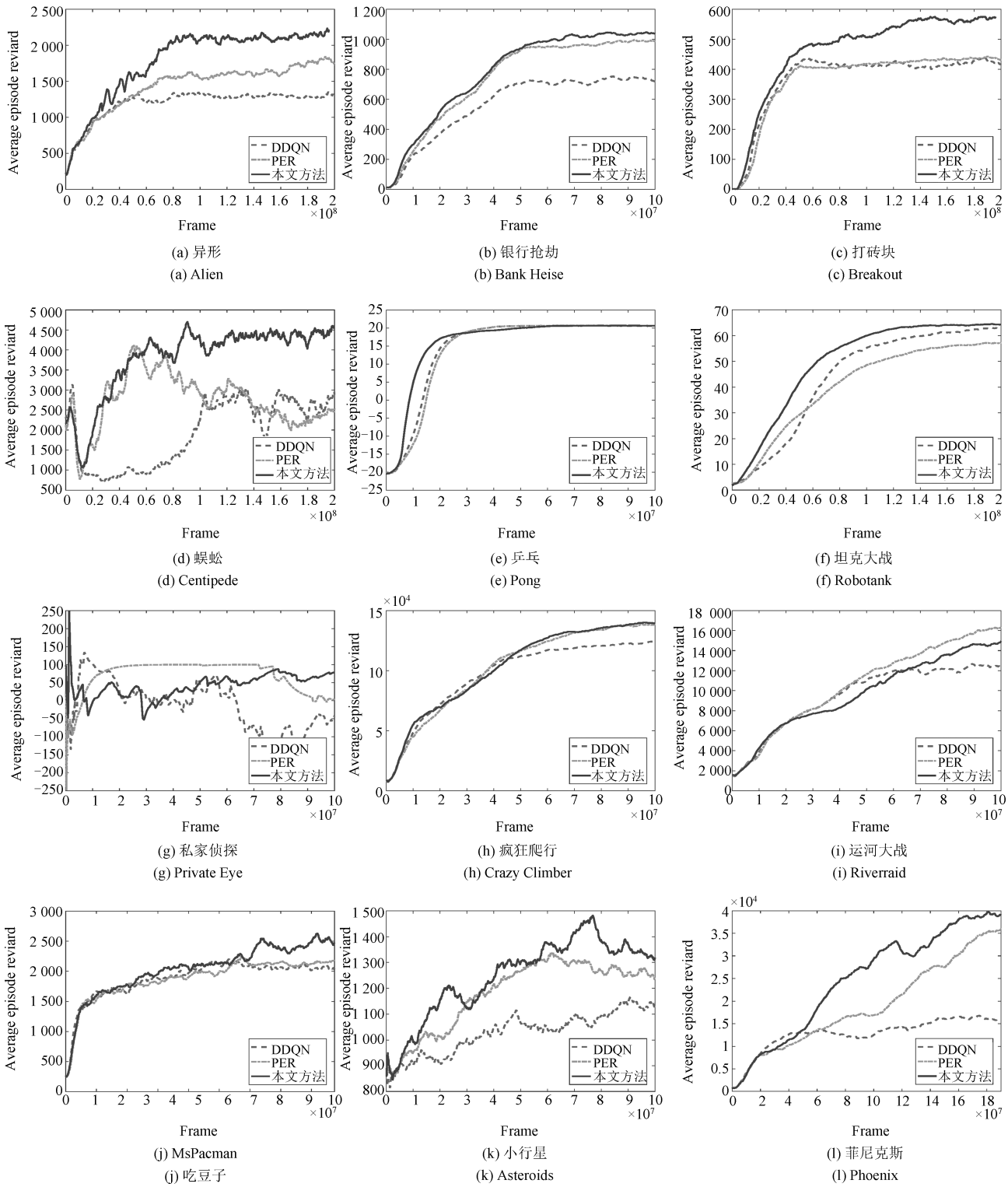


图 7　Atari 游戏训练曲线

Fig. 7　The training curves of Atari games

归一化得分如表 5 所示.

本文提出的方法在打砖块 (Breakout), 行星大战 (Asteroids)、蜈蚣 (Centipede)、坦克大战 (Robotank) 等游戏中表现出色, 能够取得更高的分数. 原因在于在这类游戏中, 智能体目标较为明确而且单一, 即为消灭敌人或者击打砖块, 这样获得累积回报较大的序列相对于获得累积回报较小的序列会有指导意义. 在运河大战 (Riverraid) 游戏中, 智

能体可以从不同途径获得奖励, 包括消灭敌人 (如图 8 (a)), 获得补给 (如图 8 (b)), 和消灭补给 (如图 8 (c)), 这使得获得不同累积回报的序列之间没有明显的优劣关系, 累积回报作为序列采样的依据这一特点没有表现出来. 所以本文方法在运河大战游戏中表现不佳.

表 3 全部游戏的规约得分总体统计表

Table 3 Summary of normalized score on all games

	DDQN[24]	PER[20]	本文方法
平均数	221.14 %	300.16 %	**357.27** %

表 4 全部游戏实际得分统计结果

Table 4 Scores on 12 Atari games with no-ops evaluation

游戏名称	随机智能体	人类专家	DDQN	PER	本文方法
Alien	227.80	6 875.40	2 907.30	3 310.80	**3 692.30**
Asteroids	719.10	13 156.70	930.60	1 699.30	**1 927.30**
Bank Heist	14.20	734.40	728.30	1 126.80	**1 248.20**
Breakout	1.70	31.80	403.00	381.50	**533.01**
Centipede	2 090.90	11 963.20	4 139.40	5 175.40	**5 691.30**
Crazy Climber	10 780.50	35 410.50	101 874.00	183 137.00	**185 513.70**
MsPacman	307.30	15 693.40	3 210.00	4 751.2	**5 313.90**
Phoenix	761	7 242.6	12 252.5	32 808.3	**39 427.4**
Pong	−20.70	9.30	21.00	20.70	**21.00**
Private Eye	24.90	69 571.30	129.70	200.00	**265.00**
Riverraid	1 338.50	13 513.30	12 015.30	**20 494.00**	14 231.70
Robotank	2.20	11.90	62.70	58.60	**66.70**

表 5 全部游戏规约得分统计结果

Table 5 Normalized scores on 12 Atari games

游戏名称	DDQN	PER	本文方法
Alien	40.31 %	46.38 %	**52.12 %**
Asteroids	1.70 %	7.8 %	**9.71 %**
Bank Heist	99.15 %	154.48 %	**171.34 %**
Breakout	1 333.22 %	1 261.79 %	**1 765.12 %**
Centipede	20.75 %	31.24 %	**36.47 %**
Crazy Climber	369.85 %	699.78 %	**709.43 %**
MsPacman	18.87 %	28.88 %	**32.54 %**
Phoenix	177.30 %	494.40 %	**596.59 %**
Pong	139.00 %	138.00 %	**139.00 %**
Private Eye	0.15 %	0.25 %	**0.35 %**
Riverraid	87.7 %	**157.34 %**	105.90 %
Robotank	623.71 %	581.44 %	**664.95 %**

(a) 消灭敌人 (a) Destroy the enemy

(b) 获得补给 (b) Get the recharge

(c) 消灭补给 (c) Destroy the recharge

图 8 Riverraid 游戏截图

Fig. 8 Screenshots of Riverraid

4 结论

强化学习的目的是得到策略使累积回报最大化. 累积回报大的序列中的样本和累积回报小的序列中的样本都可以用于深度 Q 网络的训练, 但是累积回报大的序列中的样本对深度 Q 网络的收敛和策略提升有更大的促进作用. 本文提出的深度 Q 学习的二次主动采样方法从序列累积回报分布和样本 TD-error 分布两个方面构造序列采样优先级和样本采样优先级. 用本文方法获得的样本训练深度 Q 网络, 在 Atari 平台上进行了验证, 实验结果表明, 本文方法加速了网络的收敛速度, 提高了经验池中样本的利用效率, 也提升了策略的质量.

References

1 Sutton R S, Barto A. *Reinforcement Learning: An Introduction 2nd(Draft)*. MIT Press, 2017.

2 Glimcher P W. Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. *Proceedings of the National Academy of Sciences*, 2011, **108**(Supplement 3): 15647−15654

3 Kober J, Bagnell J A, Peters J. Reinforcement learning in robotics: a survey. *The International Journal of Robotics Research*, 2013, **32**(11): 1238−1274

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

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

6 Sutton R S. Learning to predict by the methods of temporal differences. *Machine learning*, 1988, **3**(1): 9−44

7 Sutton R S, McAllester D A, Singh S P, et al. Policy gradient methods for reinforcement learning with function approximation. In: Advances in neural information processing systems. Denver, United States: MIT Press, 2000. 1057−1063

8 Konda V R, Tsitsiklis J N. Actor-critic algorithms. In: Advances in neural information processing systems. Denver, United States: MIT Press, 2000. 1008−1014

9 Tesauro G. Td-gammon: A self-teaching backgammon program. *Applications of Neural Networks*, 1995. 267−285

10 LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, **521**(7553): 436−444

11 Goodfellow I, Bengio Y, Courville A. *Deep learning*. MIT press, 2016

12 Guo Xiao-Xiao, Li Cheng, Mei Qiao-Zhu. Deep Learning Applied to Games. *Acta Automatica Sinica*, 2016, **42**(5): 676−684
(郭潇逍, 李程, 梅俏竹. 深度学习在游戏中的应用. 自动化学报, 2016, **42**(5): 676−684)

13 Browne C B, Powley E, Whitehouse D, et al. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 2012, **4**(1): 1−43

14 Silver D, Huang A, Maddison C J, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016, **529**(7587): 484−489

15 Tian Yuan-Dong. A simple analysis of AlphaGo. *Acta Automatica Sinica*, 2016, **42**(5): 671−675
(田渊栋. 阿法狗围棋系统的简要分析. 自动化学报, 2016, **42**(5): 671−675)

16 Chen Xing-Guo, Yu Yang. Reinforcement Learning and Its Application to the Game of Go. *Acta Automatica Sinica*, 2016, **42**(5): 685−695
(陈兴国, 俞扬. 强化学习及其在电脑围棋中的应用. 自动化学报, 2016, **42**(5): 685−695)

17 Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *Nature*, 2015, **518**(7540): 529−533

18 Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning. *arXiv preprint, arXiv*, 2013. 1312.5602

19 Mirowski P, Pascanu R, Viola F, et al. Learning to navigate in complex environments. *arXiv preprint, arXiv*, 2016. 1611.03673

20 Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model. *Interspeech*, 2010, 2: 3

21 He D, Xia Y, Qin T, et al. Dual learning for machine translation. In: Advances in Neural Information Processing Systems. Barcelona, Spain: Mit Press, 2016. 820−828

22 Silver D, Lever G, Heess N, et al. Deterministic policy gradient algorithms. In: Proceedings of the 31st international conference on machine learning (ICML−14). Beijing, China: ACM, 2014. 387−395

23 Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. *arXiv preprint, arXiv*, 2015. 1509.02971

24 Li Y. Deep reinforcement learning: An overview. *arXiv preprint, arXiv*, 2017. 1701.07274

25 Baird L. Residual algorithms: Reinforcement learning with function approximation. In: Proceedings of the 12th international conference on machine learning. Tahoe City, United States: ACM, 1995. 30−37

26 Taylor M E, Stone P. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 2009, **10**(Jul): 1633−1685

27 Yin H, Pan S J. Knowledge Transfer for Deep Reinforcement Learning with Hierarchical Experience Replay. In: Proceedings of the 31st AAAI Conf on Artificial Intelligence. Menlo Park, United States: AAAI, 2017. 1640−1646

28 Glatt R, Costa A H R. Policy Reuse in Deep Reinforcement Learning. In: Proceedings of the 31st AAAI Conf on Artificial Intelligence. Menlo Park, United States: AAAI, 2017. 4929−4930

29 Sutton R S. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 1991, **2**(4): 160−163

30 Deisenroth M, Rasmussen C E. PILCO: a model-based and data-efficient approach to policy search. In: Proceedings of the 28th International Conference on machine learning (ICML−11). Bellevue, United States: ACM, 2011. 465−472

31 Schaul T, Quan J, Antonoglou I, et al. Prioritized experience replay. *arXiv preprint, arXiv*, 2015, 1511.05952

32 Zhai J, Liu Q, Zhang Z, et al. Deep Q-learning with prioritized sampling. In: International Conference on Neural Information Processing. Kyoto, Japan: Springer, 2016. 13−22

33 Lin L H. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine learning*, 1992, **8**(3/4): 69−97

34 Morton J. Deep Reinforcement Learning [Online], available: http://web.stanford.edu/class/aa228/drl.pdf, April 18, 2018.

35 Van Hasselt H, Guez A, Silver D. Deep Reinforcement Learning with Double Q-Learning. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence. Menlo Park, United States: AAAI, 2016. 2094−2100

36 Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv*, 2015. 1511.06581

37 Dolan R J, Dayan P. Goals and habits in the brain. *Neuron*, 2013, **80**(2): 312−325

38 Zhao D, Wang H, Shao K, et al. Deep reinforcement learning with experience replay based on sarsa. *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on. IEEE*, 2016. 1−6

39 Wang Z, Bapst V, Heess N, et al. Sample efficient actor-critic with experience replay. *arXiv preprint, arXiv*, 2016. 1611.01224

40 Bellemare M G, Naddaf Y, Veness J, et al. The Arcade Learning Environment: An evaluation platform for general agents. *J. Artif. Intell. Res.(JAIR)*, 2013, 47: 253−279

赵英男 哈尔滨工业大学计算机科学与技术学院博士研究生. 2017 年获得哈尔滨工业大学计算机科学与技术硕士学位. 主要研究方向为强化学习, 机器学习.
E-mail: ynzhao_rl@163.com
(**ZHAO Ying-Nan** Ph. D. candidate at the School of Computer Science and Technology, Harbin Institute of Technology. He received his master degree in computer science and technology from Harbin Institute of Technology in 2017. His research interest covers reinforcement learning and machine learning.)

刘 鹏 哈尔滨工业大学计算机科学与技术学院副教授. 2007 年获得哈尔滨工业大学微电子与固体电子学博士学位. 主要研究方向为图像处理, 视频分析, 模式识别, 超大规模集成电路设计.
E-mail: pengliu@hit.edu.cn
(**LIU Peng** Associate professor at the School of Computer Science and Technology, Harbin Institute of Technology. He received his Ph. D. degree in microelectronics and solid state electronics from Harbin Institute of Technology in 2007. His research interest covers image processing, video analysis, pattern recognition, and design of very large scale integrated (VLSI) circuit.)

赵 巍 哈尔滨工业大学计算机科学与技术学院副教授. 曾获黑龙江省科技进步一等奖. 主要研究方向为模式识别, 机器学习, 计算机视觉. 本文通信作者.
E-mail: zhaowei@hit.edu.cn
(**ZHAO Wei** Associate professor at the School of Computer Science and Technology, Harbin Institute of Technology. She won a First Prize of Heilongjiang Province Science and Technology Progress. Her research interest covers pattern recognition, machine learning, and computer vision. Corresponding author of this paper.)

唐降龙 哈尔滨工业大学计算机科学与技术学院教授. 1995 年获得哈尔滨工业大学计算机应用技术博士学位. 主要研究方向为模式识别, 图像处理, 机器学习.
E-mail: tangxl@hit.edu.cn
(**TANG Xiang-Long** Professor at the School of Computer Science and Technology, Harbin Institute of Technology. He received his Ph. D. degree of computer application technology from Harbin Institute of Technology in 1995. His research interest covers pattern recognition, image processing, and machine learning.)