计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(12):155−162 [doi: 10.15888/j.cnki.csa.008200]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于遗传交叉算子的深度 Q 网络样本扩充①

杨　彤[1], 秦　进[1], 谢仲涛[1], 袁琳琳[2]

[1](贵州大学 计算机科学与技术学院, 贵阳 550025)
[2](贵州开放大学 信息工程学院, 贵阳 550025)
通讯作者: 秦　进, E-mail: jqin1@gzu.edu.cn

摘　要: 区别于传统深度强化学习中通过从经验回放单元逐个选择的状态转移样本进行训练的方式, 针对采用整个序列轨迹作为训练样本的深度 Q 网络 (Deep Q Network, DQN), 提出基于遗传算法的交叉操作扩充序列样本的方法. 序列轨迹是由智能体与环境交互的试错决策过程中产生, 其中会存在相似的关键状态. 以两条序列轨迹中的相似状态作为交叉点, 能产生出当前未出现过的序列轨迹, 从而达到扩充序列样本数量、增大序列样本的多样性的目的, 进而增加智能体的探索能力, 提高样本效率. 与深度 Q 网络随机采样训练样本和采用序列样本向后更新的算法 (Episodic Backward Update, EBU) 进行对比, 所提出的方法在 Playing Atari 2600 视频游戏中能取得更高的奖赏值.

关键词: 深度强化学习; 经验回放; 样本效率; 遗传算法

引用格式: 杨彤,秦进,谢仲涛,袁琳琳.基于遗传交叉算子的深度 Q 网络样本扩充.计算机系统应用,2021,30(12):155–162. http://www.c-s-a.org.cn/1003-3254/8200.html

Samples Expanding of Deep Q Network Based on Genetic Crossover Operator

YANG Tong[1], QIN Jin[1], XIE Zhong-Tao[1], YUAN Lin-Lin[2]

[1](College of Computer Science & Technology, Guizhou University, Guiyang 550025, China)
[2](College of Information Engineering, Guizhou Open University, Guiyang 550025, China)

Abstract: Different from the traditional deep reinforcement learning method of training through transitions selected one by one from the experience replay, for the Deep Q Network (DQN) that uses the entire episode trajectory as the training sample, a method for expanding episode samples is proposed, which is based on genetic algorithm crossover operators. The episode trajectory is generated during the trial-and-error decision-making process of the interaction between the agent and the environment, in which similar key states will be encountered. With the similar state in the two episode trajectories as the intersection point, the episode trajectory that has not appeared till present can be generated to enlarge the number of episode samples and increase their diversity, thereby enhancing the agent's exploration ability and improving sample efficiency. Compared with DQN that randomly selects samples and uses the Episodic Backward Update (EBU) algorithm, the proposed method can achieve higher rewards in the Playing Atari 2600.

Key words: deep reinforcement learning; experience replay; sample efficiency; genetic algorithm

深度强化学习[1] 通过将深度学习[2] 与强化学习[3] 相结合, 在序贯决策问题上取得显著进展. 例如 Atari 2600 视频游戏[4]、棋盘游戏[5]、自动驾驶[6] 等领域. 深度学习的引入解决了传统强化学习在处理高维状态空间时所遇到的问题, 通过使用线性或非线性函数对强化学习中的值函数进行表示, 能直接从高维状态输入中学习到策略.

即便深度强化学习在很多复杂环境中取得突破,

① 基金项目: 国家自然科学基金 (61562009); 贵州省科技计划 (黔科合基础 [2019]1130 号)
Foundation item: National Natural Science Foundation of China(61562009); Science and Technology Plan of Guizhou Province ([2019]1130)
收稿时间: 2021-02-22; 修改时间: 2021-03-19; 采用时间: 2021-03-26

但目前仍有一些问题影响着深度强化学习的发展[7]. 其中样本效率问题是强化学习的挑战之一, 强化学习的学习过程是通过智能体与环境的相互作用, 探索未知的区域并获得数据样本, 再根据所获的数据样本进策略更新. 因而强化学习的学习过程中没有固定的数据集, 需要通过智能体与环境进行大量交互获取数据样本, 并依据样本不断改进策略, 最终才能适应环境以得到良好的策略. 这样会导致智能体需要消耗极大的时间成本用于获取数据样本, 特别是在现实环境中, 会承担很多风险与代价. 例如深度 Q 网络[1] 在 Arcade Learning Environment (ALE) 平台的游戏中需要大概两亿帧的状态转移样本 (大约 39 天的实时游戏时间) 来进行训练才能达到人类水平; 在自动驾驶等领域应用深度强化学习算法时, 不能冒险尝试多种情况, 使得获得样本的成本代价高; 在一些机器人控制研究中, 需要消耗大量的成本用于机器人与环境进行交互来获取数据等. 因此, 减少智能体与环境的交互次数显得尤为重要. 同时 如何对经验进行回放将会影响深度强化学习算法最终所训练出的策略的优劣, 选择不同的数据样本对学习最优策略产生的影响也将不同. 深度 Q 网络算法通过随机选择记忆回放单元中的样本进行训练, 这种方式能够打破样本之间的相关性, 为深度神 经网络提供独立同分布的样本, 但对每个样本进行同等对待的随机抽样方式显得并不高效.

国内外研究者对以上问题开展了研究. 一些研究者认为缺乏有效的探索会导致所生成的数据样本不能为学习最优策略提供良好的帮助, 需要通过改变探索策略生成出更利于学习的数据样本, 这样能在一定程度上减少智能体与环境交互的次数. 通过引入随机性改变探索策略, Plappert 等人[8] 提出在参数空间加入适当的噪声; Fortunato 等人[9] 提出在参数空间与动作空间加入噪声相结合的方法. 不同于非盲目式的改变探索策略, Bellemare 等人[10] 将对状态的虚拟计数作为衡量不确定性的指标; Pathak 等人[11] 提出基于好奇心驱动的探索方法记录对状态的访问次数, 用于赋予内在奖励以鼓励智能体探索; 杨珉等人[12] 使用贝叶斯线性回归方法提高智能体的探索效率; 李超等人[13] 根据智能体对于状态空间的离散化程度改写值函数形式, 基于该值函数对环境进行合理探索. 当然, 若能够有效构建环境模型, 智能体将无需再与环境进行真实的交互, 直接从拟合的环境模型中获取数据样本. Tangkaratt 等人[14] 提出学习控制的概率推理方法使用高斯过程对状态转移模型进行建模, 使用最小二乘条件密度来学习状态转移模型; Ha 等人[15] 构建神经网络生成模型, 学习低维空间下的状态表示的世界模型. 这些方法都是通过从智能体与环境交互的角度出发来提升样本效率, 忽略了如何在本身已获得的数据样本上, 减少智能体与环境的交互次数. 因而, 在从与环境的交互中获取数据样本后, 如何高效利用样本也会影响到样本效率. Schaul 等人[16] 提出优先经验回放算法通过 TD-error 值赋予状态转移样本不同的重要性; 赵英男等人[17] 提出先以序列累计回报作为样本优先级再以 TD-error 分布构造优先级的二次采样的方式对样本进行选择; 朱斐等人[18] 通过奖赏、时间步和采样次数共同决定样本的优先级, 使智能体能更有效的选择动作; Lin 等人[19] 提出的序列记忆深度 Q 网络算法通过存储经验的奖赏值对采样到的状态转移样本进行区别对待等. 但以上的采样方式都是单个看待经验回放单元中的状态转移样本, 由于强化学习中的奖励具有稀疏性与延迟性, 对单个样本进行采样时很容易采样到不具有奖励的状态转移样本, 导致采样效率低. 同时, 强化学习作为一种通过与环境交互、从环境状态到动作映射的学习方式, 在学习过程中会产生无数条序列轨迹, 这些序列轨迹由多个样本所组成, 并且每个序列样本都会带有奖赏值. Lee 等人[20] 在 2019 年提出的序列向后更新 (Episodic Backward Update, EBU) 算法, 不同于以往对状态转移样本进行单个采样的方式, 通过采样整个序列轨迹由后向前更新对深度 Q 网络进行训练, 更快地传播奖赏值, 提高了采样效率.

遗传算法作为求解最优问题的常用算法, 能通过模拟生物在自然环境中的进化过程搜索全局最优解. 其中交叉操作能采用多种交叉方法以生成新的个体[21]. 当面临深度强化学习中需要解决减少智能体与环境的交互次数的问题时, 运用能产生新个体的交叉算子, 进而对数据样本数量进行扩充. 因此, 本文尝试利用遗传算法中的交叉算子, 作用于序列轨迹以产生新的序列样本, 从而扩充序列样本数量, 减小智能体与环境的交互次数, 使智能体能获得更优策略. 在 Playing Atari 2600 视频游戏验证本方法, 实验结果显示该方法能获得更高的平均奖赏值并提高样本的利用率.

1 研究基础

1.1 强化学习

在强化学习中, 智能体通过与环境进行交互以达

到最大化累计奖赏的目标, 整个学习过程使用马尔科夫决策过程进行建模: 在每个离散的时间步t, 智能体从与环境的交互过程中获得状态的表示s_t, 依据当前策略π, 得到应采取的动作a_t, 执行该动作得到即时奖励r_t并转移到下一状态s_{t+1}. 这个过程一直循环下去, 直到到达标志着智能体的任务结束时的终止状态, 意味着完成一个序列轨迹. 状态转移样本表示为(s_t,a_t,r_t,s_{t+1}), 智能体从初始状态到终止状态的序列轨迹由状态转移样本构成, 表示为: $(s_1,a_1,r_1,s_2),\cdots,(s_t,a_t,r_t,s_{t+1})$.

智能体在一个序列中状态s_t所获累计奖赏值定义为:

$$G_t=\sum_{k=0}^{T}\gamma^k R_{t+k+1} \tag{1}$$

其中, 折扣因子$\gamma\in(0,1]$用于调节不同状态对累计奖赏值的影响程度. 强化学习的目的是学习到使得G_t期望值最大的策略, 动作值函数定义为:

$$\begin{aligned}Q_\pi(s,a)&=E_\pi[G_t|S_t=s,A_t=a]\\&=\sum_{s'\in S,a'\in A}p(s'|s,a)[r+\gamma Q_\pi(s',a')]\end{aligned} \tag{2}$$

由于状态转移概率未知, 可通过 Q 学习更新值函数:

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

经过式 (3) 的不断迭代, 最终状态值函数$Q(s,a)$会收敛到最优值, 达到最大化累计序列奖赏值的目标.

1.2 深度强化学习与经验回放

在强化学习面对高维状态空间时, 深度 Q 网络[1]算法通过引入深度学习中的神经网络对 Q 学习算法中的 Q 值进行参数化, 经验回放机制为神经网络的训练提供数据样本. 经验回放单元 D 存储了智能体在每个时间步生成的状态转移样本(s_t,a_t,r_t,s_{t+1}), 此时, 状态动作值函数更新方式为:

$$Q(s,a;\theta)\leftarrow Q(s,a;\theta)+\alpha[r+\gamma\max_{a'\in A}Q(s',a';\theta')-Q(s,a;\theta)] \tag{4}$$

训练过程中, 智能体从经验回放单元中随机选择状态转移样本执行更新, 利用随机梯度下降方法最小化损失函数:

$$L(\theta)=E_{(s,a,r,s')\sim D}\left[(y-Q(s,a;\theta))^2\right] \tag{5}$$

其中,

$$y=\begin{cases}r+\gamma\max\limits_{a'\in A}Q(s',a';\theta'), & \text{如果}s'\text{是终态}\\ r, & \text{否则}\end{cases} \tag{6}$$

式 (6) 中, θ'为目标值函数网络的参数, 在一定的周期内将当前网络参数复制给目标值函数网络, 以保证学习过程的稳定性.

1.3 EBU 算法

EBU 算法的主要工作是解决深度强化学习中的样本效率问题. 主要有以下两方面:

(1) 深度 Q 网络通过均匀随机选择的方式对状态转移样本进行采样, 但在许多问题中, 智能体与环境交互所获的奖励往往具有稀疏性与延迟性, 该方式会使得抽样到具有奖励的状态转移样本的概率很低. (2) 在训练早期阶段, 所有 Q 值初始化为 0, 在使用式 (4) 对 Q 值进行更新时, 若下一状态的 Q 值也为 0, 会导致更新所获信息无效. 因此, 该算法通过采样一整个序列样本进行训练, 能保证至少一个带有奖励的状态转移样本被采样到, 同时, 在对 Q 值进行更新的时候, 从序列末尾往序列开始的方向进行更新, 使得更新过程能充分利用有效信息[11].

具体训练过程中, 采样一整个序列样本后, 通过序列由后向前更新的方式赋予式 (6) 中的 y 值. 由于直接使用此 y 值会造成深度 Q 网络的不稳定问题, 通过引入调节因子β降低影响, 更新方式为:

$$\tilde{Q}[A_{k+1},k]\leftarrow\beta y_{k+1}+(1-\beta)\tilde{Q}[A_{k+1},k] \tag{7}$$

其中, k为序列时间步, $\tilde{Q}$值为从目标值函数网络中获取的值. 当β=0时, 相当于一步更新的深度 Q 网络; 当β=1时, 容易产生过估计问题. 该算法分为固定β值和自适应β值两种, 本文仅针对样本问题, 选取固定β值的算法进行研究.

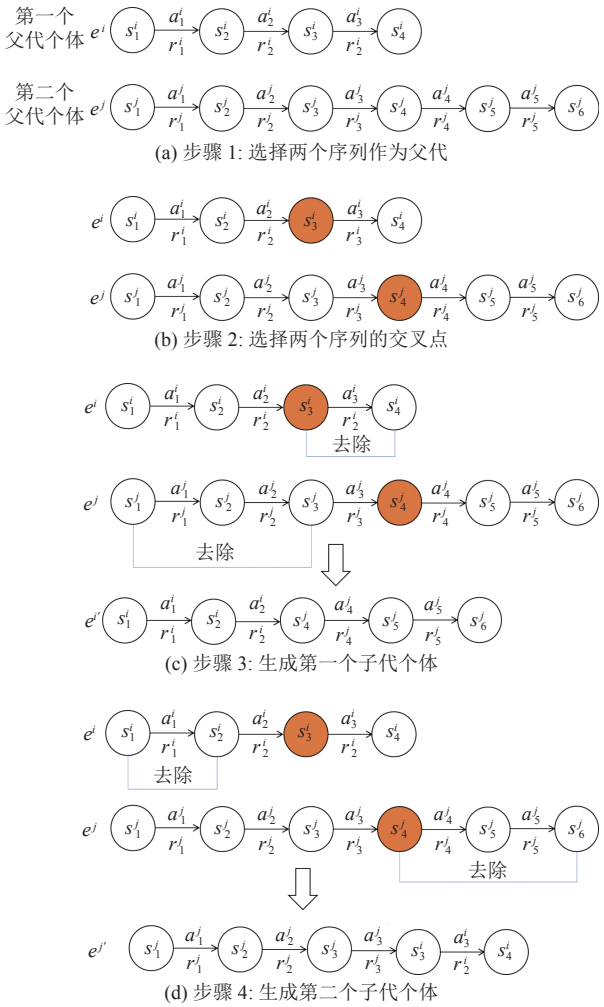
2 基于遗传交叉算子增强深度 Q 网络

2.1 利用交叉算子扩充序列样本数量

强化学习作为基于行为的学习方式, 不同于依据正例与反例来告知应采取何种行为的监督与非监督学习, 需要通过试错的决策过程来生成无数条序列轨迹, 以用来发现最优策略. 为了减少该决策过程所需消耗的成本与代价, 可采用遗传交叉算子对序列轨迹进行扩充. 强化学习试错的学习方式会使得序列轨迹中存在相似状态, 选择这些相似状态作为交叉点, 能产生出目前未出现过的序列轨迹, 以此用来扩充序列轨迹数量, 提高对当前样本的利用率.

本文将智能体与环境交互产生的序列样本作为染色

体, 序列样本中的状态作为基因. 每次从经验回放单元中选择两条染色体作为父代, 选取相似基因作为交叉点, 采用遗传算法中的单点交叉算子, 交换父代基因生成两个子代. 具体而言, 可如图 1 所示, 经验回放单元可表示为$D=\{e^1,e^2,\cdots\}$, $e^i=\{(s_1^i,a_1^i,r_1^i,s_2^i),(s_2^i,a_2^i,r_2^i,s_3^i),\cdots,(s_{T-1}^i,a_{T-1}^i,r_{T-1}^i,s_T^i)\}$为经验回放中的第 i 个序列, 在 D 中选择两个序列e^i、e^j, 判定s_3^i与s_4^j作为交叉点, 依据单点交叉生成新的$e^{i'}$与$e^{j'}$并存入 D'中.



(a) 步骤 1: 选择两个序列作为父代

(b) 步骤 2: 选择两个序列的交叉点

(c) 步骤 3: 生成第一个子代个体

(d) 步骤 4: 生成第二个子代个体

图 1 序列样本的交叉操作步骤

同时, 由于经验回放单元中所存储的序列是由智能体不断与环境交互试错生成, 因此每条序列的终态会存在大量相似性, 在选择序列进行交叉操作时, 我们所选取的应是以不同状态作为终态的序列轨迹, 以便利用遗传算子的探索能力, 增加新生成的序列的多样性. 如图 2 所示, 在这个迷宫环境中, 绿色与蓝色轨迹是智能体与环境交互生成, 且以不同的状态作为结尾, 黄色五角星为交叉状态所在的位置, 红色与橙色为交叉后生成的序列轨迹. 若选择以相同状态作为结尾的轨迹, 虽然也能产生新的序列, 但并没有选择以不同状态作为结束的轨迹交叉而产生的序列轨迹的多样性高.

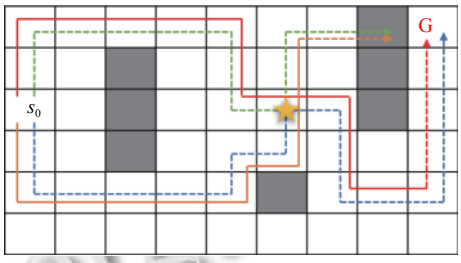


图 2 迷宫环境中序列轨迹交叉图

此外, 由式 (1) 计算每条序列的累计奖赏, 其可作为当前策略的评价指标, 累计奖赏值越大说明当前策略越优. 并且, 序列累计奖赏值越大的序列所包含的有效动作样本越多[15], 意味着使用这些状态转移样本会更有利于学习最优策略. 因此, 对于利用交叉算子所生成的新序列, 我们仅保留序列累计奖赏变高的序列. 通过使用交叉算子生成的序列降低了数据样本的生成成本, 增加了经验回放单元中的序列数量.

2.2 判定状态相似的方法

当状态空间巨大甚至无穷时, 一个状态出现在两个不同的序列轨迹中的概率非常低. 交叉点不必是两个不同序列中的完全相同的状态, 而是两个相似度高的状态. 是因为强化学习采用函数近似之后, 本质上就是对有限样本上的经验进行泛化, 把所访问过的状态的值推广到新的近似状态上.

深度 Q 网络中采用直接将图像作为输入, 通过深度神经网络提取出高层特征, 实现端到端的训练方式. 基于神经网络提取的高层特征度量两个状态的相似度一方面降低了维度, 另一方面也抓住了所求解问题的状态内在相似的内在本质. 这里, 用于特征提取的深度神经网络结构如图 3 所示.

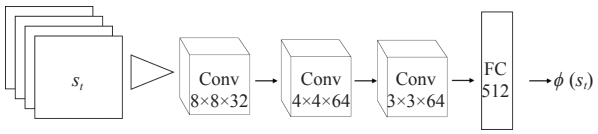


图 3 对状态抽取特征网络图

对序列轨迹中的状态进行高层特征抽取后, 比较两状态特征之间的距离, 距离越大说明相似度越低, 距

离越小则相似度越高. 依据式 (8) 计算特征之间的距离:

$$d\left(s_t^i, s_t^j\right) = \sqrt{\left(\phi\left(s_t^i\right) - \phi\left(s_t^j\right)\right)^2} \tag{8}$$

其中, $\phi(s_t)$为通过深度神经网络对状态进行特征抽取后的结果.

2.3 从经验回放单元中选择样本

强化学习作为利用试错的方法解决序贯决策问题的学习方式, 每条序列轨迹都是智能体为了达到最终的学习目标所做出的尝试, 经验回放单元中存储着智能体与环境实时交互所产生的序列轨迹. 与序列向后更新算法类似, 我们通过选择整个序列的采样技术, 使得深度 Q 网络在训练时采取从整个序列由后往前的状态转移样本进行训练.

通过使用对原始经验回放单元中的序列轨迹进行交叉操作后, 将交叉生成的新的序列轨迹存储在经验回放单元 D'之中, 得到了新的经验回放单元 D'. D'作为原始经验回放单元 D 的补充, 在采样过程中, 优先采样 D 中的序列, 再以一定的概率采样新的经验回放单元 D'中的序列, 具体为随机产生 [0, 1] 之间的随机数σ, 如果小于设置的参数μ, 则从经验回放单元 D 中选择序列样本, 反之, 从 D'中选择.

2.4 算法描述

本文提出的利用遗传交叉算子 (Genetic Crossover Operator, GCO) 扩充序列样本的方法是建立在序列向后更新 (EBU) 算法的基础上, 为智能体在选择序列样本进行训练时, 扩大序列样本的选择面, 增大序列样本多样性, 有利于获得更优质的策略. 具体算法如算法 1.

算法 1. GCO-EBU 算法

1) 初始化: 记忆回放单元 D 的容量为 N, 生成的记忆回放单元 D'容量为 N', 折扣因子 γ, 学习率 η, 抽样样本批量大小 B, 训练最大时间步 M, 目标 Q 网络更新频率 F;
2) for e=1 to I do:
3) for t=1 to T_e do:
4) 基于当前状态s_t, 以ε的概率随机选择动作a_t, 否则选择动作$a_t=\arg\max Q(s,a;\theta)$;
5) 执行动作a_t并观察到下一状态s_{t+1}, 获得即时奖励r_t, 以及是否达到终态;
6) 获得状态转移样本(s_t,a_t,r_t,s_{t+1}), 存储到经验回放单元 D 中;
7) 利用式 (8) 从 D 或 D'中采样序列 e', 采样到的序列长度为 L;
8) 初始化临时目标 Q 表$\tilde{Q}=\tilde{Q}(s',\cdot;\theta')$;
9) 初始化目标向量 y, $y_L=R_L$;
10) for $k=T_{e'}-1$ to 1 do:
11) $\tilde{Q}[A_{k+1},k]\leftarrow\beta y_{k+1}+(1-\beta)\tilde{Q}[A_{k+1},k]$
12) $y_k\leftarrow R_k+\gamma\max_{a\in A}Q[a,k]$
13) end for
14) 计算损失函数$L(\theta)=(y-Q(s,a;\theta))^2$,并由梯度下降更新参数$\theta$;
15) if $t\%F$ ==0:
16) $\theta'=\theta$
17) end for
18) 从经验回放中选择以不同状态作为终态结束的序列e^i,e^j
$e^i=<s_0^i,a_0^i,r_0^i,s_1^i,a_1^i,r_1^i,s_2^i,\cdots,s_{T_i}^i>$
$e^j=<s_0^j,a_0^j,r_0^j,s_1^j,a_1^j,r_1^j,s_2^j,\cdots,s_{T_j}^j>$
19) 选取序列 e^i, e^j 中$d(s_m^i,s_n^j)$最小的两个状态作为交叉点
20) 应用交叉算子生成新的序列样本
$e^{j'}=<s_0^j,a_0^j,r_0^j,s_1^i,\cdots,s_m^i,a_m^i,\cdots,s_{T_i}^i>$
$e^{i'}=<s_0^i,a_0^i,r_0^i,s_1^j,\cdots,s_n^j,a_n^j,\cdots,s_{T_j}^j>$
21) 将$e^{i'}$、$e^{j'}$存入D'
22) end for

该算法中, 4)–6) 步为强化学习使用 ε-greedy 策略作为行为策略产生样本, 并将样本保存到经验回放单元中; 7)–16) 步是采样序列样本并进行训练的过程, 其中 7)–12) 步为 EBU 算法过程, 通过从后向前倒序回放序列中的状态转移样本进行训练, 第 7) 步为控制采样到本文所提出方法所生成的序列样本的比例; 18)~20) 步是通过交叉操作产生新的序列的过程, 从经验回放单元中随机选择到两条序列, 比较两条序列的终态的高级特征是否相似, 选取终态差异较大的序列进行交叉, 其中第 19) 步为依据状态的高级特征判断出两条序列的交叉点, 依据交叉点最终完成交叉, 生成出新的序列样本.

3 实验分析

3.1 实验设置

本文采用 ALE 作为实验验证平台. ALE 中包含了大量各种类别的 Playing Atari 2600 游戏接口, 为研究深度强化学习提供了丰富内容, 许多改进的深度强化学习算法都在该平台验证. 采用 Playing Atari 2600 游戏中的 9 个控制问题验证本文方法的有效性, 控制问题分别为 Breakout、Space_invaders、Ms_pacman、Atlantis、Phoenix、Video_pinball、Assault、Asteriods.

将本文方法与 NATURE-DQN[1]、EBU[20] 算法进行对比, 其中 NATURE-DQN 的训练过程是从经验回放单元中随机选择样本, 本文则是在利用序列样本训练的 EBU 算法基础上应用所提出的方法. 具体参数

设置为: 由于 ALE 所提供的游戏画面大小统一为 210×210, 为了方便训练, 将图像灰度化处理并将大小裁剪为 84×84, 并将 4 帧图像作为一个状态看待; 由于每个 Atari 游戏奖励设置不同, 实验过程中对每个游戏所获的奖励进行统一化处理, 即将奖励归一化在[−1,1] 之间; 在训练过程中, 经验回放单元初始化是没有任何样本存在, 两个经验回放单元大小设置为 1000000, 在智能体与环境交互超过 50 000 步时再从经验回放单元中选择样本进行训练, 每次选择样本的个数为 32 的倍数, 序列样本的大小不足 32 的倍数时, 在往前选择状态转移样本进行填充; 使用 ε-greedy 策略作为智能体的行为策略, ε大小随着步数从 1 递减到 0.1; 采用 RMSProp 方式更新网络参数, 动量设置为 0.95; 折扣因子γ设置为 0.99; 学习率设置为 0.000 25; 选择经验回放单元 D'的参数μ设置为 0.9; 网络结构与 N3ATURE-DQN 相同, 使用三层卷积网络与两层全连接网络构成, 第 1 层卷积具有 32 个大小为 8×8 的滤波器, 步长为 4, 第 2 层卷积具有 64 个大小为 4×4 的滤波器, 步长为 2, 第 3 层卷积具有 64 个大小为 3×3的滤波器, 步长为 1, 选取的激活函数为 ReLU, 第 1 层全连接具有 512 个神经元, 第 2 层全连接神经元个数为当前游戏所对应的动作空间大小; 目标值网络与该网络结构相同, 每隔 10 000 步将当前网络参数赋给目标值网络. 在对序列样本进行交叉操作时, 使用三层卷积层与一层全连接层对状态进行高层特征抽取, 判断两个状态是否相似. 本文采用的 EBU 算法为文献 [20] 中的算法 2, 超参数β为 0.5.

3.2 实验结果与分析

在实验中, 3 种方法均采用相同的实验参数, 图 4 为实验结果曲线图, 横坐标为 epoch, 每个 epoch 包含 62 500 步, 即智能体与环境交互 62 500 次, 纵坐标为各方法中智能体结束一个 epoch 后测试 30 次序列的平均累计奖赏值, 奖赏值越高说明智能体所学习到的策略越优.

通过实验结果可以看出, 本文提出的方法 GCO-EBU 与 NATURE-DQN 和 EBU 算法相比, 在多数控制问题上能使用更少的与环境交互的次数获得更高的奖赏值. 具体而言, 在图 4(a)–图 4(d) 的 Assault、Space_invader、Altantis、Phoenix 控制问题上 GCO-EBU 算法能在训练初期就能观察到明显效果, 说明相对于两个对比算法, 我们对序列样本数量的扩充为训练提供了帮助, 使得智能体获得了更高的平均奖赏值. 在图 4(e) Asteroids 问题中, GCO-EBU 在训练初期效果低于对比算法, 但在训练后期所获奖赏值远高于两种方法,由于该问题在训练初期序列轨迹终态相似度高, 利用交叉所生成的轨迹并未给序列样本好的扩充, 而在训练后期终态相似度大, 能生成出多样性大的序列轨迹. 在图 4(f)–图 4(g) Breakout、Video_pinball 问题上, 所获奖赏值的曲线比较相似, 稳定性低于其他问题, 这和两个问题的游戏规则有关. Breakout 问题上 GCO-EBU 优于 NATURE-DQN 算法, 但和 EBU 算法相比却没有更大的改进, 这由于 Breakout 问题在游戏初期所获的奖赏值是非常稀疏的, 导致在运用本文方法时, 所生成的序列样本的累计奖赏值发生改变的情况过少, 因此未能达到很好扩充序列样本的目的. 由 Video_pinball 问题的实验结果图也可看出, GCO-EBU 与 EBU 相比的效果也并不明显, 但整体而言, 本文所提出的方法还是能帮助智能体在相同时间内获得更高的奖赏值. 在图 4(h) Ms_pacman 问题上, GCO-EBU 与 EBU 的实验效果不如 NATURE-DQN, 但本文的改进是建立在 EBU 算法上, 在训练后期本文方法在相同的与智能体交互次数中所取得的奖赏值高于 EBU.

表 1 是 3 种算法在 8 个控制问题上的平均奖赏值的对比, 整体而言, 说明依据遗传算子来扩大序列样本数量, 增加序列样本多样性的方法, 能使得在智能体与环境的相同交互次数中, 探索到更多的序列, 提高样本的利用率, 从而获得更高的奖赏值, 得到更优质的策略.

4 结论与展望

本文所提出的利用遗传算法中的交叉算子作用于序列样本的方法, 能增大深度强化学习算法的序列样本数量, 减少智能体与环境的交互次数, 提升样本效率, 使得智能体获得更高的平均奖赏值, 最终导出更优质策略. 通过 ALE 平台上的 8 个 Playing Atari 游戏验证了该方法的有效性. 目前仅在基于值函数的强化学习算法上验证本方法, 今后的研究工作是将该方法应用于基于策略的强化学习算法, 通过扩充序列样本的方式, 增大策略空间, 更好地服务于策略梯度算法, 使得策略能跳出局部最优的困境.

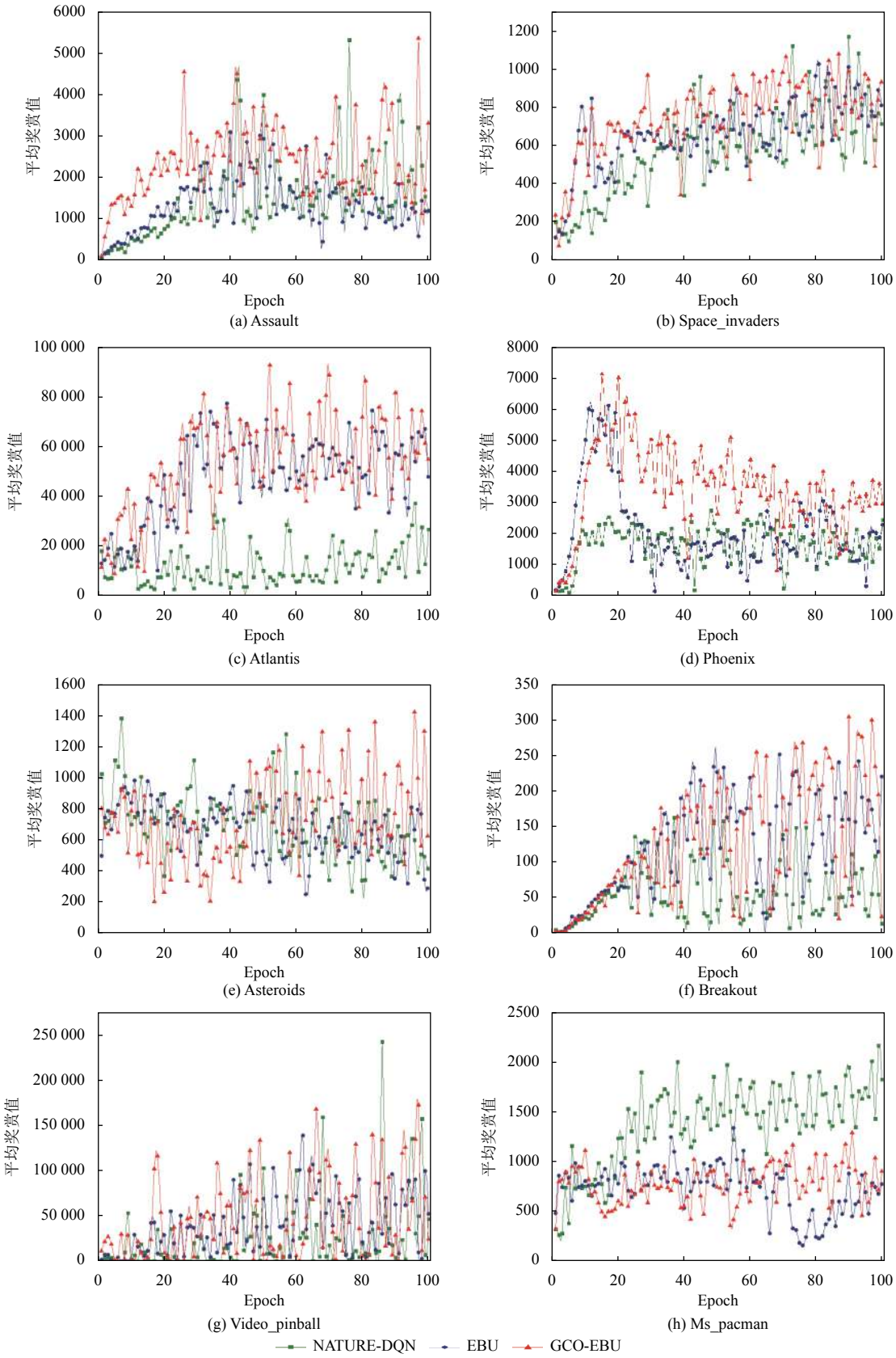


图 4 NATURE-DQN、EBU 与 GCO-EBU 在 Atari 游戏中的平均奖赏值

表 1 NATURE-DQN、EBU 与 GCO-EBU 在 Atari 游戏中的平均奖赏值得分

游戏名称	Breakout	Space_invaders	Ms_pacman	Atlantis	Phoenix	Video_pinball	Assault	Asteriods
NATURE-DQN	85.62	566.92	**1389.38**	12 500.93	1687.04	24 869.62	1511.24	702.92
EBU	122.73	663.94	735.84	47 954.27	2119.47	40 833.62	1374.97	675.67
GCO-EBU	**127.64**	**741.29**	778.46	**53 437.63**	**3481.16**	**46 809.75**	**2383.35**	**740.16**

参考文献

1 Mnih V, Kavukcuoglu K, Silver D, *et al*. Human-level control through deep reinforcement learning. Nature, 2015, 518(7540): 529–533. [doi: 10.1038/nature14236]

2 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521(7553): 436–444. [doi: 10.1038/nature14539]

3 Sutton RS, Barto AG. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 2018.

4 Mnih V, Badia AP, Mirza M, *et al*. Asynchronous methods for deep reinforcement learning. Proceedings of the 33rd International Conference on International Conference on Machine Learning-Volume 48. New York: ACM, 2016. 1928–1937.

5 Silver D, Huang A, Maddison CJ, *et al*. Mastering the game of go with deep neural networks and tree search. Nature, 2016, 529(7587): 484–489. [doi: 10.1038/nature16961]

6 Aradi S. Survey of deep reinforcement learning for motion planning of autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems. 2020. [doi: 10.1109/TITS.2020.3024655]

7 李茹杨, 彭慧民, 李仁刚, 等. 强化学习算法与应用综述. 计算机系统应用, 2020, 29(12): 13–25. [doi: 10.15888/j.cnki.csa.007701]

8 Plappert M, Houthooft R, Dhariwal P, *et al*. Parameter space noise for exploration. Proceedings of the 6th International Conference on Learning Representations. Vancouver: OpenReview.net, 2018.

9 Fortunato M, Azar MG, Piot B, *et al*. Noisy networks for exploration. Proceedings of the 6th International Conference on Learning Representations. Vancouver: OpenReview.net, 2018.

10 Bellemare MG, Srinivasan S, Ostrovski G, *et al*. Unifying count-based exploration and intrinsic motivation. Proceedings of the 30th International Conference on Neural Information Processing Systems. Barcelona: ACM, 2016. 1479–1487.

11 Pathak D, Agrawal P, Efros AA, *et al*. Curiosity-driven exploration by self-supervised prediction. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Honolulu: IEEE, 2017. 488–489.

12 杨珉, 汪洁. 解决深度探索问题的贝叶斯深度强化学习算法. 计算机科学与探索, 2020, 14(2): 307–316. [doi: 10.3778/j.issn.1673-9418.1901020]

13 李超, 门昌骞, 王文剑. PAC 最优的 RMAX-KNN 探索算法. 计算机科学与探索, 2020, 14(3): 513–526. [doi: 10.3778/j.issn.1673-9418.1905023]

14 Tangkaratt V, Mori S, Zhao TT, *et al*. Model-based policy gradients with parameter-based exploration by least-squares conditional density estimation. Neural Networks, 2014, 57: 128–140. [doi: 10.1016/j.neunet.2014.06.006]

15 Ha D, Schmidhuber J. World models. arXiv: 1803.10122v4, 2018.

16 Schaul T, Quan J, Antonoglou I, *et al*. Prioritized experience replay. Proceedings of the 4th International Conference on Learning Representations 2016. San Juan, 2016.

17 赵英男, 刘鹏, 赵巍, 等. 深度 Q 学习的二次主动采样方法. 自动化学报, 2019, 45(10): 1870–1882.

18 朱斐, 吴文, 刘全, 等. 一种最大置信上界经验采样的深度 Q 网络方法. 计算机研究与发展, 2018, 55(8): 1694–1705. [doi: 10.7544/issn1000-1239.2018.20180148]

19 Lin ZC, Zhao TQ, Yang GW, *et al*. Episodic memory deep Q-networks. Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm, 2018. 2433–2439.

20 Lee SY, Sungik C, Chung SY. Sample-efficient deep reinforcement learning via episodic backward update. Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Vancouver, 2019. 2110–2119.

21 Mirjalili S. Genetic algorithm. Mirjalili S. Evolutionary Algorithms and Neural Networks. Cham: Springer, 2019. 43–55.