

基于自适应探索改进的深度增强学习算法

毛坚桓, 殷璐嘉

(中国电子科技集团公司 第二十八研究所, 江苏 南京 210007)

摘 要: 针对深度增强学习算法中探索开发策略的平衡问题, 提出一种基于 VDBE(Value-Difference Based Exploration)扩展的自适应探索改进算法. 该算法依赖于值函数差异提出一种基于状态的探索控制策略, 以达到在初始学习阶段不熟悉周围环境时 agent 采取积极探索策略, 而随着深入学习和周边环境的熟悉, agent 逐渐降低探索率的自适应探索/开发平衡的理想行为状态.

关键词: 深度增强学习; 探索/开发平衡; VDBE; 自适应控制

中图分类号: TP31

文献标识码: A

文章编号: 1000-7180(2016)06-0139-04

A Deep Reinforcement Learning Algorithm Based on Adaptive Exploration

MAO Jian-huan, YIN Lu-jia

(The 28th Institute, China Electronics Technology Group Corporation, Nanjing 210007, China)

Abstract: To find a balance between exploration and exploitation, this paper proposes a VDBE(Value-Difference Based Exploration) based algorithm. The algorithm proposes a state-based control strategy depends on the value difference. In order to achieve the ideal exploration/exploitation behavior state, agent takes positive actions to explore environments in the initial stage of learning when agent is unfamiliar with surrounding environment. As learning time goes on and agent is more familiar with surrounding, it gradually reduces the exploration rate.

Key words: deep reinforcement learning; exploration/exploitation balance; VDBE; adaptive control

1 引言

增强学习是通过 agent 和未知的周边环境交互以寻求获取最大化长期收益的一系列决策学习的模式^[1]. 深度增强学习算法基于增强学习产生, 成功应用于 7 个 Atari 2600 游戏的学习, 且在其中的 3 个游戏中达到了人类专家水平^[2], 这证明该算法已可应用于较复杂的机器人学习领域. 然而在学习时, agent 并不了解周边环境, 需要从历史经验中学习动作. 因此, 增强学习 agent 面对着一个基本的探索/开发折中问题^[3]. Agent 是应该在已积累的经验上开发, 执行当前最佳动作, 或是执行一个不同动作. 过多开发可能导致短期收益不能最大化, 过多探索不确定环境则可能会因选择动作非最优而影响长期最大收益. 平衡探

索和开发在同时看重学习时间和学习质量的增强学习中是一个重大的挑战. 本文通过对 VDBE(Value-Difference Based Exploration)^[4]方法的扩展应用, 提出一种基于自适应探索改进的深度增强学习算法, 以平衡 agent 学习过程中的探索和开发问题. 该算法基本思路是随着学习过程中回报的稳定而逐步减少探索概率, 以达到在初始学习阶段不熟悉周边环境时 agent 采取积极探索策略, 而随着学习深入熟悉周边环境后, agent 逐渐降低探索率, 达到自适应探索/开发平衡的理想行为状态.

2 自适应探索改进的深度增强学习算法

2.1 相关工作

Agent 学习过程中探索和开发的平衡需要一个

稳定收敛的控制策略. 为了平衡探索和开发问题, 目前已有许多学者提出多种不同理论和方法^[5-9]. 其中较为广泛应用的有 ϵ 贪婪策略^[8] 和 softmax^[9]. ϵ 贪婪策略根据人工调校设定的学习时长和当前的时长计数来计算探索概率 ϵ , 然后以 ϵ 的概率选择随机动作执行, 以 $1-\epsilon$ 的概率根据贪婪策略来选择下一步执行的动作. Softmax 则利用基于 Boltzmann 分布的值函数评价等级来决定动作选择概率. 两种方式各有优缺点^[9].

2.2 VDBE-Boltzmann 方法扩展

VDBE 方法基本思路是对 ϵ 贪婪策略进行扩展, 不依赖于人工调校而是依赖于值函数差异提供一种基于状态的探索控制策略. VDBE 理想行为状态是 agent 在学习初始阶段对周边环境不熟悉时, 积极采用探索策略, 该阶段值函数存在较大偏差. 随着 agent 对于周边环境的熟悉程度, 值函数偏差慢慢减少, 因此慢慢降低探索率.

根据值函数评价的 Boltzmann 分布, 在每个学习时刻使用下列等式更新 $P_t(s)$, 可以得到接近理想行为状态的探索策略^[4].

$$f(s, a, \sigma) = \frac{1 - e^{-\frac{|Q_{t+1}(s, a) - Q_t(s, a)|}{\sigma}}}{1 + e^{-\frac{|Q_{t+1}(s, a) - Q_t(s, a)|}{\sigma}}} \\ = \frac{1 - e^{-\frac{|a, TD-Error|}{\sigma}}}{1 + e^{-\frac{|a, TD-Error|}{\sigma}}},$$

$$P_{t+1}(s) = \delta \cdot f(s_t, a_t, \sigma) + (1 - \delta) \cdot P_t(s).$$

式中, $P_t(s)$ 表示在 t 时刻 s 状态时的探索概率, 其中 σ 和 δ 都是正值常量, $\delta \in [0, 1)$, 一般 δ 取值为当前状态可选的动作集总数的倒数^[4]. 根据上面公式可知, 如果采用较小的 σ 就可以在值差异较小的时候也有较大的探索率.

考虑到实际环境中 agent 状态动作序列的海量, 无法直接使用该方式, 因此定义一个权重为 ω 的多层神经网络 P 来近似计算 t 时刻的 $P_t(s)$. 该网络输入状态 s_t , 输出 $P'_{t+1}(s)$. P 网络通过最小化每次迭代时的损失函数 $L_i(\omega_i)$ 来进行训练, $L_i(\omega_i) = E_{s, a \sim \rho(\cdot, \cdot)} [(P_i(s) - P'_i(s; \omega_i))^2]$, 其中

$$P_i(s) = \delta \cdot \frac{1 - e^{-\frac{|Q_i(s, a) - Q_{i-1}(s, a)|}{\sigma}}}{1 + e^{-\frac{|Q_i(s, a) - Q_{i-1}(s, a)|}{\sigma}}} + \\ (1 - \delta) \cdot P'_i(s; \omega_{i-1})$$

是第 i 次迭代的目标值, $\rho(s, a)$ 是序列 s 和动作 a 上的动作概率分布.

对损失函数 $L_i(\omega_i)$ 的权重 ω_i 进行随机梯度下降更新的公式为

$$\nabla_{\omega_i} L_i(\omega_i) =$$

$$E_{s, a \sim \rho(\cdot, \cdot)} [(P_{t+1}(s) - P'_{t+1}(s; \omega_i)) \nabla_{\omega_i} P'_{t+1}(s; \omega_i)].$$

2.3 基本算法

假设 agent 处于一个具有一定随机性的环境中, 每次该 agent 从合法的动作集合 $A = \{1, \dots, K\}$ 中选择一个动作 a_t 执行. Agent 执行该动作后, 环境监测器检测并修改环境内部状态, 计算相关回报, 同时将该回报值返回给 agent. agent 根据收到的回报调整学习过程, 进而再一次执行下一个动作. 由于正常情况下 agent 需要连续执行一系列动作和周边环境交互后才能有收益的反馈, 因此 agent 回报值可能依赖于前面一段历史动作的执行情况和观测结果.

假设 $X = \{x_1, x_2, \dots, x_n\}$ 为 agent 在环境中的状态集合, 因此考虑 agent 的状态和动作序列 $s_t = x_1, a_1, x_2, a_2, \dots, a_{t-1}, x_t$ 并且依赖于这些序列来学习下一步控制策略. Agent 在每个时刻的执行过程就是从 s_j 到 s_k 的转变, 可以认为这是一个标准的马尔科夫决策过程(MDP)^[10]. 所有序列都假设在有限的时间内终止, 这就导致大量有限的 MDP 产生. 所以可以用 s_t 作为时间点 t 时的状态表示, 将标准的 MDP 增强学习算法应用于该环境中.

Agent 的目标是选择最大化未来回报的动作并执行, 与环境产生交互. 假设未来回报以 γ 因子衰减,

并定义 t 时刻衰减的回报值为 $R_t = \sum_{i=t}^T \gamma^{i-t} r_i$. 定义优

化动作值函数 $Q^*(s, a)$ 作为在看见某个序列 s 后并执行某动作 a 的最大期望回报, $Q^*(s, a) = \max_{\pi} E[R_t | s_t = s, a_t = a, \pi]$, 其中 π 是策略到动作的映射. 在此引入一个多层神经网络来近似计算该值函数, 该网络以 Q 学习算法^[8] 的变体来进行训练.

在执行过程中, 由于周边环境的不确定性, agent 不可能总是在已有经验基础上选择能带来最大化收益的动作. 这里考虑采用 VDBE 扩展方法来实现 agent 探索 / 开发策略的平衡. Agent 每次执行后对 P 网络进行训练, 并采用随机梯度下降的方式更新权值. Agent 根据 $P'_{t+1}(s)$ 决定下一步是选择任意动作执行, 还是根据贪婪策略 $a = \max_a Q(s, a; \theta)$ 来确定下一步动作.

由于 Agent 在连续动作的执行过程中,数据会存在一定的相关性,并且在 s 状态下 a 的选取不一定满足静态概率分布. 因此算法采用了一种基于历史随机样本的经验回放机制^[11] 来缓解该问题,该机制利用大量历史行为的数据来平滑训练分布. 算法记录一定长度的历史行为数据,在 agent 每次训练时,以 minibatch 方式从历史记录中随机抽取 n 个记录,计算每个记录的回报并进行训练,以平滑训练数据.

2.4 算法实现

基于自适应探索改进的深度增强学习算法如下所示.

(1) 初始化存放经验回放元组的内存 M .

(2) 以任意权重初始化动作值函数 Q 和 P .

(3) 对 $k = 1$ 到 N 个片段进行如下循环操作:

初始化序列 $s_1 = \{x_1\}$, 所有 $p_1(x) = 1$, 预处理状态序列 $\varphi_1 = \varphi(s_1)$;

对 $t = 1$ 到 T , 执行:

$$p_k(s_t) = P'(s_t; \omega)$$

根据 p_k 选择一个任意动作 a_t , 或者选择 a_t

$$= \max_a Q^*(\varphi(s_t), a; \theta);$$

执行动作 a_t , 并且观察回报 r_t ;

使 $s_{t+1} = s_t, a_t, x_{t+1}$, 预处理状态序列 φ_{t+1}

$$= \varphi(s_{t+1});$$

存储 $(\varphi_t, a_t, r_t, \varphi_{t+1})$ 到内存 M 中;

使用 minibatch 方式从内存 M 中随机抽取历史

元组 $(\varphi_j, a_j, r_j, \varphi_{j+1})$;

$y_j = \begin{cases} r_j, \varphi_{j+1} \text{ 为序列结束操作} \\ r_j + \gamma \max_{a'} Q(\varphi_{j+1}, a'; \theta), \varphi_{j+1} \text{ 不是序列结束操作} \end{cases}$

更新 Q 网络参数

TD-Error = $|y_j - Q(\varphi_j, a_j; \theta)|$

$$P_{k+1}(s_t) = \delta \cdot \frac{1 - e^{-\frac{|y_j - Q(\varphi_j, a_j; \theta)|}{\sigma}}}{1 + e^{-\frac{|y_j - Q(\varphi_j, a_j; \theta)|}{\sigma}}} + (1 - \delta) \cdot P'(s_t; \omega)$$

对 P 网络根据 $(P_{k+1}(s_t) - P'_{k+1}(s_t; \omega))^2$ 执行随机梯度下降, 更新权值 ω

t 时刻执行结束

k 片段执行结束

3 实验及结果分析

为了说明自适应探索改进深度增强学习算法的有效性, 但又不失一般性. 本文在斯坦福大学 Andrej Karpathy 博士根据文献^[2] 构建的算法进行了实验. 实验基于一个四边形的封闭环境, 里面存在

墙, 一定量随机分布的红苹果和绿苹果三种物体, 碰到墙无法前行, 吃到红苹果可以得到正回报, 吃到绿苹果则得到负回报. Agent 具有一个最大可视距离, 且并不知晓这三种物体及周边环境, 它只感知到 $[0, 1]$ 之间的一个状态向量 (向量维度为物体种类个数, $[0, 1]$ 表示根据 agent 最大可视距离规范化后的 agent 到各类最近物体之间的距离). agent 可以采取直行、左右转等 5 个动作. 在 agent 根据当前探索概率选择一个动作并执行后, 周边环境对 agent 新状态做出更新, 计算当前回报, 并以标量形式返回给 agent. Agent 在收到该回报后, 通过算法进行自适应调整学习, 并思考下一步动作. 为了让 agent 具有充分的学习机会, 在每次 agent 行动后都会根据一定的条件补充一定量的随机分布红苹果或绿苹果. 同时探索率最低门限设为 5%, 让 agent 在充分学习后还能保持一定的探索能力.

为了保持较大的探索率, 取 VDBE 方法参数为 $\sigma = 0.3, \delta$ 取 $1/|A| = 0.2$. 通过在上述实验环境下进行的探索策略改进前后的测试, 获得图 1~4 所示的结果.

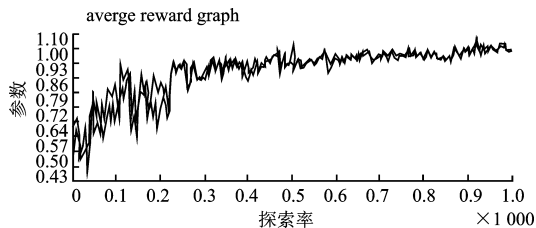


图 1 原平均收益图

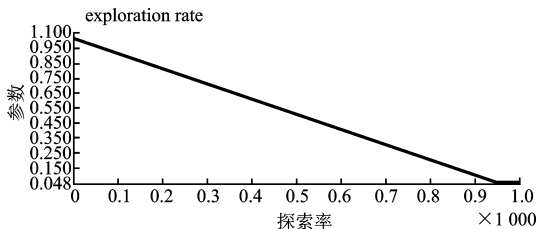


图 2 原探索率变化图

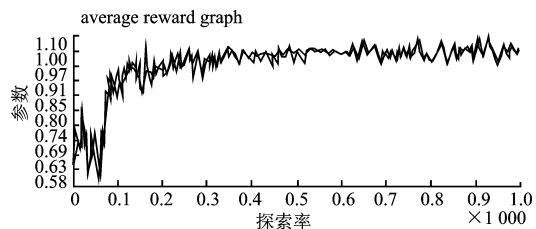


图 3 改进后平均收益图

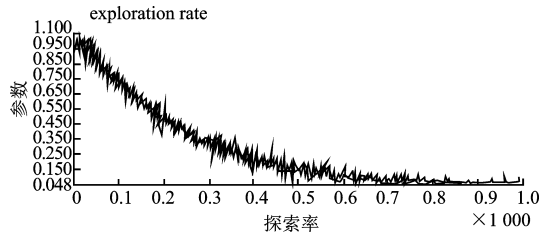


图 4 改进后探索率变化图

根据结果可知,在改进探索策略后,和原学习过程相比,使用更少的学习时长便获得了更好的学习结果. agent 在原学习过程中大概花费 19 万步学习时长(1 000 = 200 000 时长),而改进后大概只花费了 16 万步学习时长就完成了学习过程. 由于改进算法采用值差异方式自动调整学习过程和探索策略,不需要针对不同模型进行调校,所以可以用更贴合未知环境的探索/开发策略控制整个学习过程,并且获取更高的平均收益.

4 结束语

本文分析了深度增强学习过程中 agent 探索/开发折中问题,并提出了基于自适应探索改进的深度增强学习算法. 经实验证明,该算法可在模型未知、没有先验经验的条件下,为 agent 学习过程中存在的探索/开发平衡问题提供一种自适应的解决策略,并取得了较好的结果.

参考文献:

[1] ToddHester, Manuel Lopes, Peter Stone. Learning exploration strategies in model-based reinforcement learning[C]// The Twelfth International Conference on Autonomous Agents and Multiagent Systems (AA-MAS), St Paul, Minnesota:IEEE,2013.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Playing Atari with Deep Reinforcement Learning [R]. arXiv preprint arXiv:2013.

[3] Bertsekas D P. Dynamic Programming: Deterministic and Stochastic Models [M]. Englewood Cliffs, NJ: Prentice-Hall,1987.

[4] Michel Tokic. Adaptive ϵ -greedy exploration in reinforcement learning based on value differences [C] // Advances in Artificial Intelligence 33 rd Annual German Conference on AI, Karlsruhe, Germany, IEEE, 2010:21-24.

[5] Thrun S B. Efficient exploration in reinforcement learning [R]. Technical Report CMU-CS-92-102, Carnegie Mellon University, Pittsburgh, PA, USA;1992.

[6] Brafman R I, Tennenholtz M. R-MAX a general polynomial time algorithm for near-optimal reinforcement learning[R]. Journal of Machine Learning Research, 2002,21(3): 213-231.

[7] Ishii S, Yoshida W, Yoshimoto J. Control of exploitation-exploration metaparameter in reinforcement learning[J]. Neural Networks,2002,15(4-6):665-687.

[8] Watkins C. Learning from delayed rewards[D]. Cambridge: University of Cambridge, 1989.

[9] Sutton RS, Barto AG. Reinforcement Learning: An Introduction[M]. MIT Press, Cambridge,MA;1998.

[10] Kaelbling L P, Littman M L, Moore A W. Reinforcement Learning: a survey[J]. Journal of Artificial Intelligence Research, 1996,22(4):237-285.

[11] Long-Ji Lin. Reinforcement learning for robots using neural networks[R]. Technical report, DTIC Document, 1993.

作者简介:

毛坚桓 男,(1977-),硕士,工程师. 研究方向为机器学习、指控系统自动化. E-mail:mao_jh@163. com.

殷璐嘉 男,(1982-),本科,工程师. 研究方向为指控系统自动化.