



基于权重值的竞争深度双 Q 网络算法

摘要

在深度强化学习中,深度 Q 网络算法存在严重高估动作值问题,使得智能体的表现不尽人意.尽管深度双 Q 网络和竞争网络结构可以部分缓解高估带来的影响,但引入双 Q 网络的同时,有时也会低估动作值.本文提出了一种基于权重值的竞争深度双 Q 网络算法 (Weighted Dueling Double Deep Q-Network, WD3QN),把改进的双估计器及竞争网络结构结合至深度 Q 网络中,将学习到的可能动作值进行加权产生最终动作值,有效减少估计误差.最后,将算法应用于 Open AI Gym 平台上的 CartPole 经典控制问题,仿真结果显示:与已有算法对比,本算法有更好的学习效果,收敛性和训练速度均有提升.

关键词

深度强化学习;深度双 Q 网络;竞争网络结构;权重值

中图分类号 TP273

文献标志码 A

收稿日期 2021-09-13

资助项目 国家自然科学基金(61873099,62073144);广东省自然科学基金(2020A1515010441);广州市科技计划(202002030158,202002030389);广东省青年创新人才项目(2018GkQNCX005)

作者简介

汪晨曦,男,硕士生,研究方向为深度强化学习、最优控制.auwcx@mail.scut.edu.cn

赵学艳(通信作者),女,副教授,硕士生导师,主要从事随机系统和非线性系统的稳定性与镇定,复杂系统的建模、分析和控制的研究.auxyzhao@scut.edu.cn

1 华南理工大学 自动化科学与工程学院,广州,510641

2 广东交通职业技术学院,机电工程学院,广州,510650

0 引言

强化学习 (Reinforcement Learning, RL) 是由心理学、神经科学以及控制科学等多学科融合而来的一类机器学习方法^[1-3],多用于解决序贯决策问题.目前强化学习可以分为基于模型的强化学习 (model-based RL) 和无模型强化学习 (model-free RL) 两大类.在无模型强化学习算法中,Q 学习 (Q-learning)^[4]是目前最流行的算法,它使用 Q 表格有效地构建动作状态对,可直接根据 Q 值进行动作选择.但在学习训练中,因为 Q 学习算法包括一个最大化的操作,直接导致对动作值的过于乐观估计,文献[5]证明了该过估计有上限,而文献[6]则证明在特定条件下,该过估计也可能存在下限.

随着人工智能 (Artificial Intelligence, AI)、深度学习 (Deep Learning, DL)^[7]等概念的愈发火热,神经网络已经成为一大研究热点.神经网络可将复杂逻辑的高维数据转换成可靠的低维表示,已在计算机视觉、自然语言处理、推荐系统等方面展示出巨大作用.将深度学习与强化学习结合起来,根据二者不同的特点,可以使得智能体在某些随机环境下获得更为良好的表现.

近年来,深度学习与强化学习相结合 (Deep Reinforcement Learning, DRL) 的智能体训练方法大放异彩.2013 年 Mnih 等^[8]提出的深度 Q 网络 (Deep Q-Network, DQN) 算法在 Atari 2600 部分游戏中大幅超越人类玩家水平,该算法可预估所有策略的价值,取其中最佳策略执行.

最初的 DQN 算法存在一些缺陷,例如目标值不稳定、样本利用不充分等.针对上述问题,2015 年 Mnih 等^[9]提出设立回放经验池和固定目标值以使智能体训练更加稳定:经验回放打破样本前后的关联性,在训练中随机采样,而固定目标值使得反向传播算法更加稳定.但由于估计偏差及噪声作用,该算法有时会高估动作值.

为了解决高估问题,Hasselt 等^[6]提出 DDQN (Double Deep Q-Network) 算法,采用双估计器,将动作选择与动作评估分离;文献[10]提出竞争网络结构,将 Q 网络有效分为价值函数与优势函数两部分,其中价值函数仅仅与状态有关,与动作无关.但 Zhang 等^[11]指出引入双估计器有时会导致低估动作值,提出 WDDQ (Weight Double Deep Q-learning) 算法,采用权重值的方法对 Q 值进行调节,仿真结果显示该算法可有效提升训练的稳定性.

本文针对 DQN 算法中的高估、DDQN 算法与竞争网络结构中的低估问题, 结合 WDDQ 算法的权重值方法, 提出基于权重值的竞争深度双 Q 网络算法 (Weighted Dueling Double Deep Q-Network, WD3QN), 将竞争网络结构与改进的双估计器结合, 对动作值有更精准的估计, 有效减少误差. 通过对 Open AI Gym 中的经典控制问题 CartPole^[12] 进行研究, 实验结果表明, WD3QN 算法与已有算法相比有更快的收敛速度和更好的稳定性.

本文的其余部分安排如下: 第 1 节介绍强化学习及其背景知识; 第 2 节给出 WD3QN 算法设计; 第 3 节给出实验结果与分析; 第 4 节是总结.

1 强化学习及背景知识

1.1 强化学习

在强化学习中, 智能体需与外界环境进行交互, 找到最优的序列决策, 使奖励函数最大化. 在一个离散的时间序列 $t = 0, 1, 2, 3, \dots$ 中, 对于每一个时刻 t , 智能体观察环境状态 $s_t \in S$, 根据当前状态 s_t 选择动作 $a_t \in A$, 获得奖励回报 $r_t \in R$ 而后进入下一个状态 $s_{t+1} \in S$. 马尔可夫决策框架下, 使用元组 $\langle S, A, P, R, \gamma \rangle$ 表示整个探索过程: S 是有限状态集合; A 是有限动作集合; 状态转移概率 $P(s_{t+1} | s_t, a_t) = P[S_{t+1} = s_{t+1} | S_t = s_t, A_t = a_t]$; $R(s, a) = E[R_t = r_t | S_t = s_t, A_t = a_t]$ 为相应的奖励值; $\gamma \in [0, 1]$, 为折扣因子.

智能体要学习策略 π , 得到每一个状态下的动作概率分布, 使其总和折扣回报值 $G_t = \sum_{i=0}^{\infty} \gamma^i R(s_i, \pi(s_i))$ 达到最大. 在策略 π 下, 动作价值函数是根据当前的状态 s , 选取动作 a 所获得的期望回报, 用 $Q^\pi(s, a) = E_\pi[G_t | S = s, A = a]$ 表示. 最优动作价值函数 $Q^*(s, a) = \max_\pi Q^\pi(s, a)$ 表示期望回报最大化.

若面对离散有限的动作及状态空间, 可以使用经典的值迭代算法——Q 学习算法. 通过学习出一个表格 (Q-table), 直接表示状态 s 下每个动作 a 的未来期望奖励, 通过 ϵ -greedy 算法采取相应的策略: 即以 ϵ 概率采取任意可能动作, 以 $1 - \epsilon$ 概率采取贪心策略, 可避免智能体陷入局部最优; 与此同时设置衰减的 ϵ 参数, 加快中后期智能体的训练速度. Q-learning 算法流程如下:

算法 1 Q-learning

初始化 Q, s ;

For episode = 1, \max_{episode} do

在当前状态 s 下, 通过 ϵ -greedy 算法, 基于 Q 表格选择动作 a ;

采取动作 a , 获取奖励值 r 及下一状态 s' ;

$a^* \leftarrow \operatorname{argmax}_a Q(s', a)$,

$\zeta \leftarrow r + \gamma Q(s', a^*) - Q(s, a)$;

$Q(s, a) \leftarrow Q(s, a) + \alpha \times \zeta, s \leftarrow s'$;

End For

当动作及状态空间很大时, 维数灾问题迎面而来, 使用表格存下所有的动作状态对显然并不现实. 可采用带参数 θ 的函数近似方法来逼近最优动作价值函数, 表达式如下:

$$Q(s, a; \theta) \approx Q^*(s, a). \quad (1)$$

1.2 DQN 算法

面对维数颇高的动作及状态空间, 经典的 Q 学习算法显得力不从心, 可将表格更新转变为函数近似问题, 使用函数值来代替 Q 表格值. 神经网络可将复杂逻辑的高维数据转换成可靠的低维表示, 有较好的特征提取能力, 与 Q 学习算法结合, 即深度 Q 网络.

在 DQN 算法中, 使用多层神经网络逼近动作价值函数. 为提高智能体 agent 训练时的稳定性, 引入两个重要机制: 经验回放与固定目标 Q 值. 在训练过程中, 当前状态 s 下选择动作 a , 获得奖励值 r 且进入下一状态 s' , 数据样本 (s, a, r, s') 存入经验池中, 网络参数 θ 通过随机梯度下降算法优化, 其中目标网络值:

$$y^{\text{DQN}} = r + \gamma \max_{a'} Q(s', a'; \theta^-), \quad (2)$$

损失函数如下:

$$l = (y^{\text{DQN}} - Q(s, a; \theta))^2. \quad (3)$$

为避免样本前后关联性对结果的影响, 每次随机抽取 m 个样本数据进行训练. 其中 θ^- 代表目标网络参数, θ 是当前在线网络参数, 二者的网络结构一致, 每 C 步进行赋值: $\theta^- \leftarrow \theta$, 经验回放与固定目标值可以提升算法的稳定性, 获得较好的实验结果, 算法流程如下:

算法 2 DQN 算法

1) 初始化 Q 网络 $Q(s, a; \theta)$ 参数, 随机初始化目标网络参数 θ^- ;

2) 初始化经验回放池 及外界环境;

3) 获取初始状态 s_0 , 根据 ϵ -greedy 算法选择动作 a_0 并记录 r_0 ;

4) For $i = 1, N$ do

- 5) 计算目标网络值: $y_i = r_i + \gamma \max_{a'} Q(s_{i+1}, a'; \theta^-)$;
- 6) 计算均方误差损失函数: $l = (y_i - Q(s_i, a_i; \theta))^2$;
- 7) 利用随机梯度下降算法更新网络参数;
- 8) 每过 C 步,把当前网络参数赋给目标网络: $\theta^- \leftarrow \theta$;
- 9) 将样本数据 (s_i, a_i, r_i, s_{i+1}) 存入经验池 ;
- 10) End For

在智能体训练中, DQN 算法使用固定 Q 作为目标值, 随机选取经验重放池 中小批量数据样本 (s, a, r, s') 进行梯度下降, 每 C 步更新目标网络参数.

2 基于权重值的竞争深度双 Q 网络算法

2.1 DDQN 算法

深度 Q 网络算法由于选择相应动作时对 Q 网络值取最大化操作, 导致对动作值存在高估问题. 深度双 Q 网络将动作的选择与评估分离, 使用在线网络选择动作, 而目标网络则对动作进行评估, 从而较好地降低过估计. 更新方式与 DQN 类似, 公式如下:

$$y^{\text{DDQN}} = r + \gamma Q(s', \arg \max_{a'} Q(s', a'; \theta); \theta'). \quad (4)$$

与经典深度 Q 网络算法相比, DDQN 算法没有额外增加网络, 目标网络与在线网络各司其职, 同样每 C 步对网络赋值: $\theta' \leftarrow \theta$. Hasselt 等^[6] 的实验结果显示, 相比于 DQN 算法, DDQN 能有效缓解高估问题, 智能体性能有较好提升.

2.2 Dueling 网络结构

深度双 Q 网络通过将动作的选择与评估操作分离, 有效降低了过高估计影响. 与此同时, Wang 等^[10] 通过优化神经网络结构从而达到优化算法的目的: 竞争网络结构将 Q 网络有效分为价值函数 $V(s; \theta, \alpha)$ (value function) 与优势函数 $A(s, a; \theta, \beta)$ (advantage function) 两部分, 其中价值函数仅与状态有关, 与动作无关, 如图 1^[10] 所示.

公式如下:

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \alpha) + A(s, a; \theta, \beta), \quad (5)$$

其中, α, β 分别为价值函数与优势函数独有的网络参数, 而 θ 则为公共网络参数.

在竞争网络结构中, 优势函数与价值函数作为子网络结构, 最终输出值由二者线性组合得到. 但在式(5)中, 无法直接辨识出价值函数与优势函数各自的作用. 为了提高函数可辨识度, 实际工程中所使

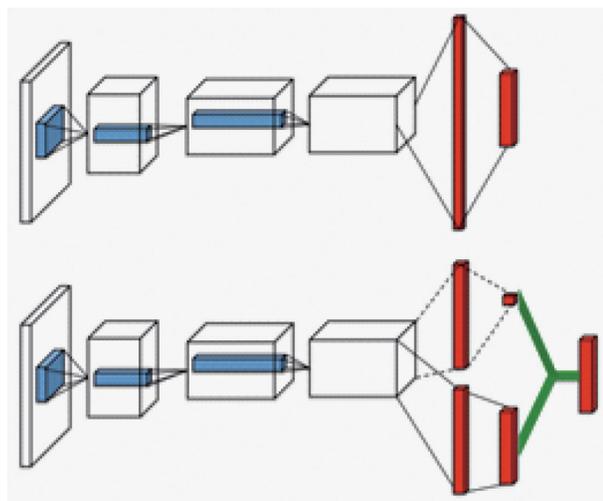


图 1 Dueling 网络结构^[10]

Fig. 1 Dueling network^[10]

用的方法如下:

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \alpha) + \left(A(s, a; \theta, \beta) - \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} A(s, a'; \theta, \beta) \right). \quad (6)$$

对优势函数 $A(s, a; \theta, \beta)$ 做中心化处理, 相比于 DQN, 仅对 Q 网络最终输出部分做些许调整, 结果证明, 竞争网络结构对降低过估计有显著作用, 提升智能体性能的同时亦优化了网络的稳定性.

2.3 D3QN 算法

将深度双 Q 网络与竞争网络结构相结合, 形成了新的强化学习算法: 竞争深度双 Q 网络算法 (Dueling Double Deep Q-Network, D3QN). 与 DQN 相比, D3QN 有效缓解了最大化算子带来的过估计影响, 算法流程如下:

算法 3 D3QN 算法

- 1) 初始化 Q 网络 $Q(s, a; \theta, \alpha, \beta)$ 参数, 初始化目标网络参数 θ' , 将 Q 网络的参数值赋给目标网络: $\theta' \leftarrow \theta$, 初始化经验池 ;
- 2) For episode = 1, \max_{episode} do
- 3) 初始化外界环境, $r = 0$, 得到状态 s_0 ;
- 4) For $t = 1, \max_{\text{length}}$ do
- 5) 根据当前状态 s_t , 输入在线 Q 网络中, 根据 ϵ -greedy 算法选择当前状态下的动作 a_t ;
- 6) 获取下一状态 s_{t+1} 和即时奖励 r_t , 将样本 (s_t, a_t, r_t, s_{t+1}) 存入经验池 D ;
- 7) 从经验池 中随机抽样 (s_j, a_j, r_j, s_{j+1}) m 个进行更新;
- 8) 计算当前 Q 网络目标值: $y_j = r_j + \gamma Q(s_{j+1},$

$\operatorname{argmax}_a Q(s_{j+1}, a'; \theta, \alpha, \beta); \theta', \alpha, \beta);$

9) 计算均方误差损失函数: $l = (y_j - Q(s_j, a_j; \theta, \alpha, \beta))^2;$

10) 使用随机梯度下降算法更新优化网络参数;

11) 每经过 τ 步, 有 $\theta' \leftarrow \theta$, 赋值网络参数;

12) End For

13) End For

2.4 基于权重值的竞争深度双 Q 网络

深度双 Q 网络与竞争网络结构, 对过估计解决良好, 但双 Q 学习有时也会存在低估问题^[11]. 以上述算法为基础, 本文提出 WD3QN 算法, 将双估计器与竞争网络结构结合, Q 值基于权重进行调整, 综合算法性能.

同样将动作的选择以及评估分离, 佐以竞争网络结构, 使用 $Q(s', a'; \theta, \alpha, \beta)$ 与 $Q(s', a'; \theta', \alpha, \beta)$ 的加权值作目标网络值, 计算公式如下:

$$y^{\text{WD3QN}} = r + \gamma [\eta \times Q(s', a'; \theta, \alpha, \beta) + (1 - \eta) \times Q(s', a'; \theta', \alpha, \beta)], \quad (7)$$

式中的 η 为权值, 超参数 c 在实验中选择, 计算公式如下:

$$\eta = \delta / (c + \delta), \quad (8)$$

其中 δ 值计算如下:

$$\delta = |Q(s', a'; \theta', \alpha, \beta) - Q(s', a''; \theta', \alpha, \beta)|, \quad (9)$$

a', a'' 分别代表取当前网络值最大与最小动作:

$$a' = \operatorname{argmax}_a Q(s', a; \theta, \alpha, \beta), \quad (10)$$

$$a'' = \operatorname{argmin}_a Q(s', a; \theta, \alpha, \beta). \quad (11)$$

整体算法流程如下:

算法 4 WD3QN 算法

1) 随机初始化 Q 网络参数 θ 及目标网络参数 θ' ;

2) 初始化重放经验池, 初始化智能体环境;

3) For episode = 1, \max_{episode} do

4) 获取初始状态 s_0 ;

5) For $i = 1, T$ do

6) 将状态 s_i 输入在线 Q 网络中, 根据 ε -greedy 算法选择动作 a_i ;

7) 获取下一状态 s_{i+1} 和奖励 r_i , 将样本 (s_i, a_i, r_i, s_{i+1}) 存入经验池;

8) 从经验池中抽取 n 个样本 (s_k, a_k, r_k, s_{k+1}) 进行参数更新;

9) $a' = \operatorname{argmax}_a Q(s_{k+1}, a; \theta, \alpha, \beta),$

$a'' = \operatorname{argmin}_a Q(s_{k+1}, a; \theta, \alpha, \beta);$

10) $\delta = |Q(s_{k+1}, a'; \theta', \alpha, \beta) - Q(s_{k+1}, a''; \theta', \alpha, \beta)|, \eta = \delta / (c + \delta);$

11) 计算目标值: $y_k = r_k + \gamma [\eta \times Q(s_{k+1}, a'; \theta, \alpha, \beta) + (1 - \eta) \times Q(s_{k+1}, a'; \theta', \alpha, \beta)];$

12) 损失函数 $l = (y_k - Q(s_k, a_k; \theta, \alpha, \beta))^2$, 使用随机梯度下降算法优化网络参数;

13) 每经过 τ 步, 更新目标网络参数: $\theta' \leftarrow \theta;$

14) End For

15) End For

首先进行参数初始化, 智能体与环境交互并根据贪心策略选择相应动作, 将转移样本 (s, a, r, s') 存入经验重放池中. 在训练时, 随机选取小批量样本数据, 根据式(7)计算目标网络值, 使用随机梯度下降算法更新相应的网络参数, 每 τ 步对目标网络参数进行赋值.

3 实验分析

3.1 实验平台与参数设置

算法验证环境: Open AI Gym^[13], 深度学习框架为 PyTorch 1.8.1, Python 版本 3.7, 以 Gym 中经典控制问题 CartPole 为实验对象, 采用 DDQN 算法, WD-DQN 算法^[14] 及 D3QN 算法作为 baseline 进行对比, 其中经验重放池大小为 200 000, minibatch 为 32, 学习率设置 0.000 5. WD3QN 算法中: 以系统状态元组作为输入, 第 1 个全连接层为 state_dim×512, 而后分别过优势函数与价值函数层(均为 512×512), 优势函数输出层为 512×action_dim, 价值函数输出层为 512×1, 经线性组合得 $Q(s, a; \theta, \alpha, \beta)$. D3QN 算法与上述基本相同, 无基于权重值的双估计器结构; WD-DQN 算法则无竞争网络结构. 训练时选择随机梯度下降算法, 贪心策略中初始值为 0.1, 更新法则如下: $\varepsilon = \max(0.01, \varepsilon - 10^{-6}), \gamma = 0.99$.

3.2 实验结果分析

与传统的监督学习不同, 深度强化学习使用自身产生的数据作为训练集, 对算法达到稳定快慢以及稳定的持续时间长短进行评估.

首先研究 WD3QN 算法中超参数 c 的取值影响, 分别取值 1, 10, 100 进行训练, 图中横坐标 episode 为训练次数, 纵坐标为每次训练的总和回报值, 对比结果图 2 所示.

若将以上过程视为训练状态, 对智能体每 50 个 episode 进行评估, 结果如图 3 所示.

由图 3 可以看出, 算法的收敛性与稳定性在 $c =$

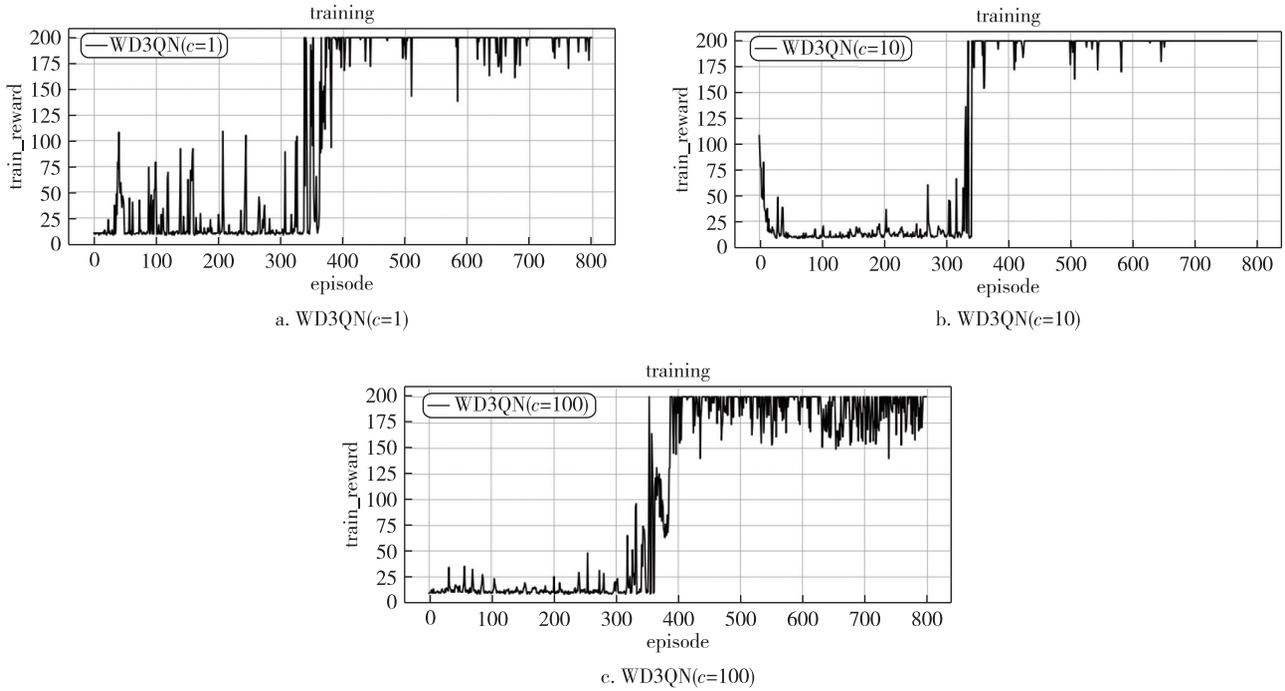


图2 超参数 $c = (1, 10, 100)$ 训练结果

Fig. 2 Training results with hyperparameter c equals 1 (a), 10 (b), and 100(c)

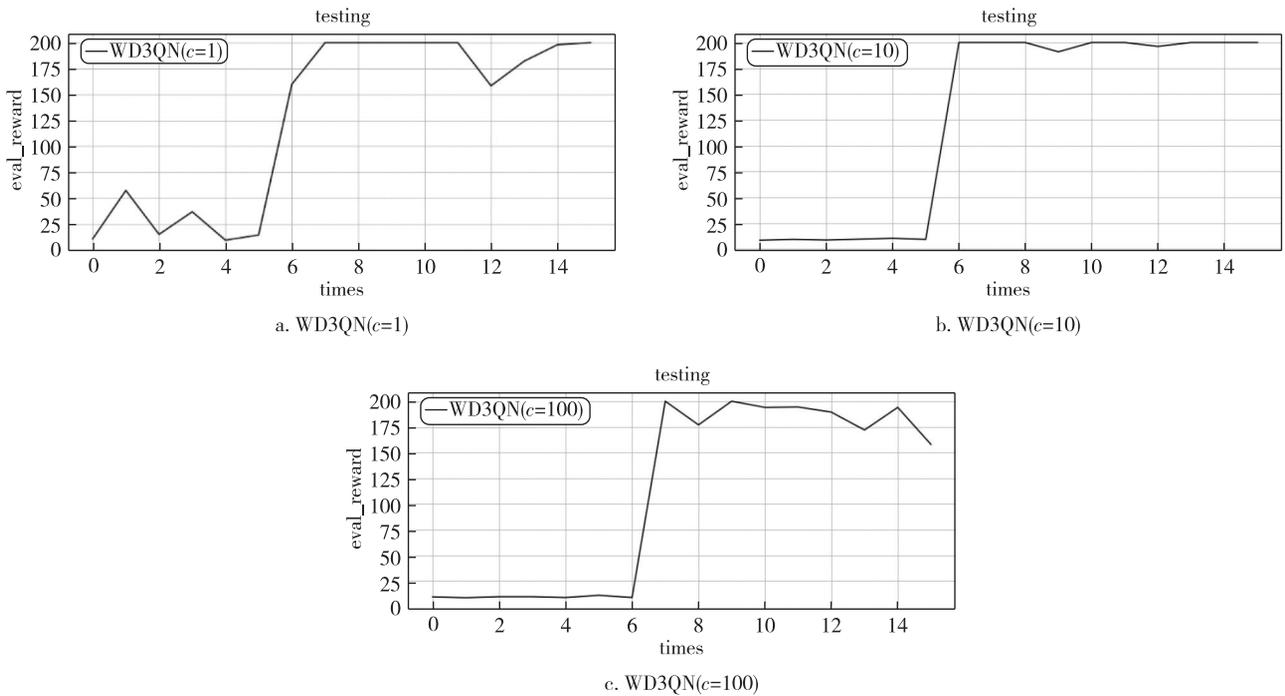


图3 超参数 $c = (1, 10, 100)$ 测试结果

Fig. 3 Testing results with hyperparameter c equals 1 (a), 10 (b), and 100(c)

10 时优于 $c = 1$ 和 100. 粗略设置超参数 c 为常数其实并不准确, 在后续研究中, 或可以考虑将其设置为自适应参数. 下面对比实验中, 默认算法超参数 $c = 10$.

图 4 为不同算法 (DDQN, D3QN, WDDQN,

WD3QN) 的训练以及评估效果.

同样对智能体每 50 个 episode 进行相应评估, 结果如图 5 所示.

由图 4 及图 5 可以看出, 在智能体训练与测试

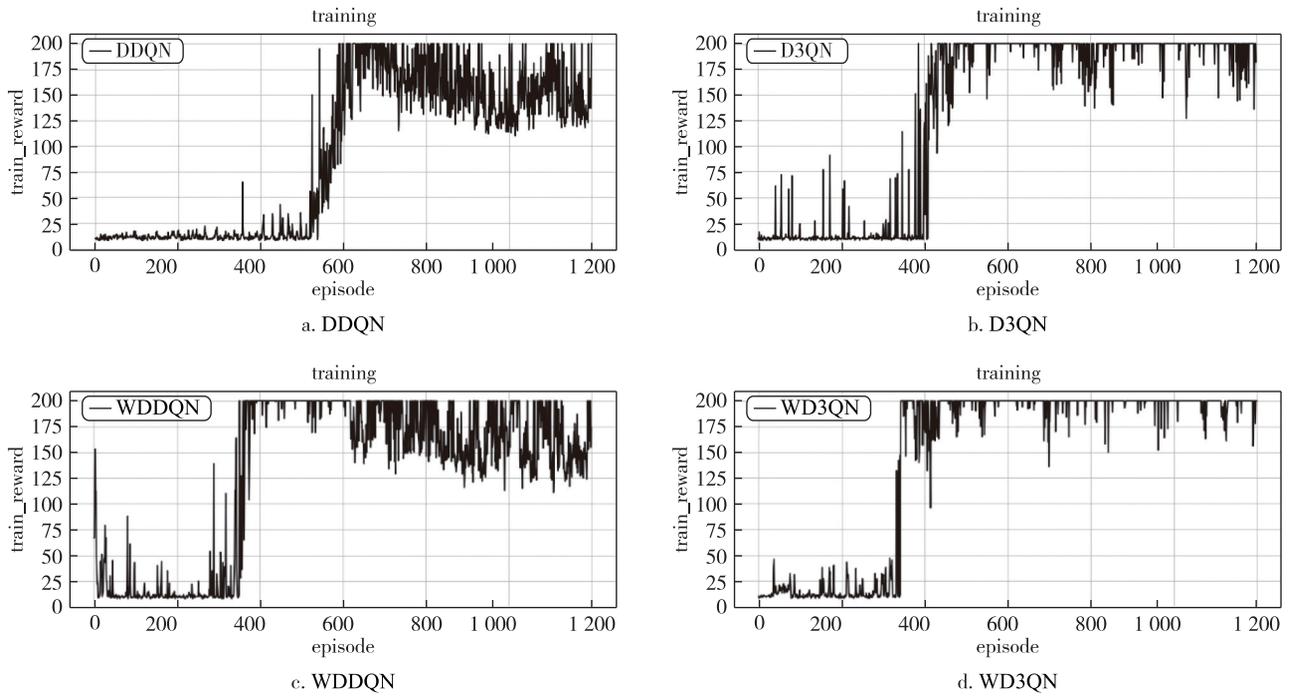


图4 不同算法训练结果

Fig. 4 Training results of different algorithms

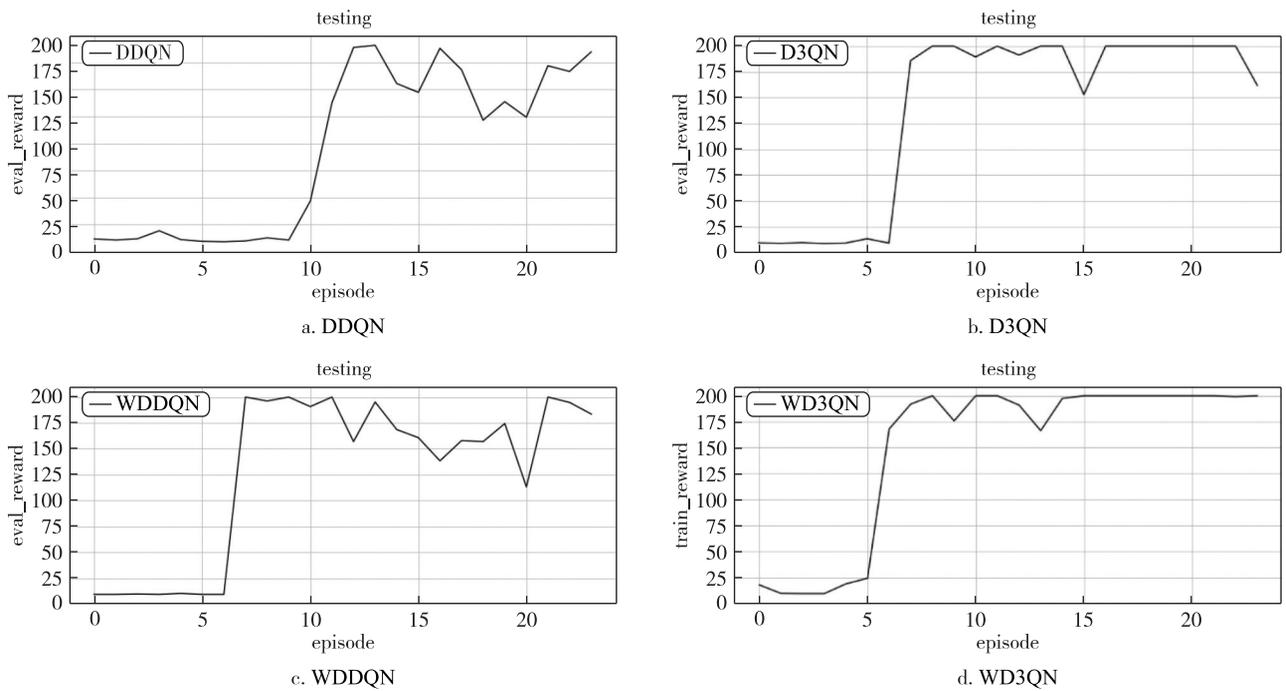


图5 不同算法测试结果

Fig. 5 Testing results of different algorithms

中,WD3QN 算法的收敛性与稳定性均明显优于其他三种算法,得益于竞争网络结构与深度双 Q 网络,缓解了对动作值高估的影响.与此同时,基于权重值的双估计器结构在训练后期 (episode>600) 减轻对动

作值的低估问题,对目标值的估计更加精确.

4 总结

本文提出一种基于权重值的竞争深度双 Q 网络

算法,将深度双 Q 网络与竞争网络结构结合,引入带权重的双估计器,对目标网络值有更精准的估计,从而有更优的策略选择.通过实验仿真对比,证明该算法的收敛性与稳定性均有效提升.下一步的研究内容即对权重比例 c 进行探讨,将其设置为自适应超参数;与此同时也可尝试加入循环神经网络结构、图神经网络模型等.

参考文献

References

- [1] Bellman R. Dynamic programming [J]. Science, 1966, 153 (3731): 34-37
- [2] Sutton R S, Barto A G. Introduction to reinforcement learning [M]. Cambridge: MIT Press, 1998
- [3] Bertsekas D P. Dynamic programming and optimal control; volume I [M]. Nashua, NH, USA: Athena Scientific, 1995
- [4] Watkins C J C H, Dayan P. Q-learning [J]. Machine Learning, 1992, 8(3/4): 279-292
- [5] Thrun S, Schwartz A. Issues in using function approximation for reinforcement learning [C] // Proceedings of the Fourth Connectionist Models Summer School, 1993: 255-263
- [6] Hasselt H V, Guez A, Silver D. Deep reinforcement learning with double Q-learning [C] // Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16), 2016: 2094-2100
- [7] LeCun Y, Bengio Y, Hinton G. Deep learning [J]. Nature, 2015, 521(7553): 436-444
- [8] Mnih V, Kavukcuoglu K, Silver D, et al. Playing atari with deep reinforcement learning [J]. arXiv e-print, 2013, arXiv: 1312.5602
- [9] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540): 529-533
- [10] Wang Z Y, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning [C] // Proceedings of the 33rd International Conference on International Conference on Machine Learning, 2016, 48: 1995-2003
- [11] Zhang Z Z, Pan Z Y, Kochenderfer M J. Weighted double Q-learning [C] // Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017: 3455-3461
- [12] Geva S, Sitte J. A cartpole experiment benchmark for trainable controllers [J]. IEEE Control Systems Magazine, 1993, 13(5): 40-51
- [13] Brockman G, Cheung V, Pettersson L, et al. OpenAI Gym [J]. arXiv e-print, 2016, arXiv: 1606.01540
- [14] 吴金金, 刘全, 陈松, 等. 一种权重平均值的深度双 Q 网络方法 [J]. 计算机研究与发展, 2020, 57(3): 576-589
- WU Jinjin, LIU Quan, CHEN Song, et al. Averaged weighted double deep Q-network [J]. Journal of Computer Research and Development, 2020, 57(3): 576-589

Weighted dueling double deep Q-network

WANG Chenxi¹ ZHAO Xueyan¹ GUO Xin²

¹ School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641

² School of Mechanical and Electrical Engineering, Guangdong Communication Polytechnic, Guangzhou 510650

Abstract In deep reinforcement learning, the deep Q-network algorithm seriously overestimates the action value, which degrades the performance of agents. The double deep Q-network and dueling network structure can partially alleviate the impact of overestimation, sometimes the former one even underestimate the action value. Here, a Weighted Dueling Double Deep Q-Network (WD3QN) algorithm is proposed, in which the improved double estimators and dueling network structure are combined into the deep Q-network, and the learned possible action values are weighted to produce the final action value, which can effectively reduce the estimation error. Finally, the algorithm is applied to the classical CartPole control problem on the open AI Gym platform. The simulation results show that compared with other existing algorithms, the proposed algorithm has better learning effect, convergence and training speed.

Key words deep reinforcement learning; double deep Q-network; dueling network structure; weighted value