https://doi.org/10.1051/jnwpu/20213951057

基于强化学习的机械臂自主视觉感知控制方法

胡春阳[1], 王恒[2], 史豪斌[2]

(1.湖北文理学院 计算机工程学院, 湖北 襄阳 441053; 2.西北工业大学 计算机学院, 陕西 西安 710129)

摘 要:传统机械臂控制方法按照人为预设固定轨迹来对其进行控制,完成特定的任务,依赖于精确的环境模型,并且控制过程缺乏一定的自适应性。为解决该问题,提出一种自主视觉感知与强化学习相结合的端到端机械臂智能控制方法。该方法中视觉感知使用 YOLO 算法,策略控制模块使用 DDPG 强化学习算法,使机械臂能够在复杂的环境中学习到自主控制策略,并且在训练过程使用了模仿学习与后视经验重播,加速了机械臂的学习过程。实验结果表明算法能够在更短的时间内收敛,并且在仿真环境中自主感知目标位置及整体策略控制都有着出色的表现。

关 键 词:机器视觉;强化学习;模仿学习;系统仿真;智能控制

中图分类号:TP391.4 **文献标志码**:A **文章编号**:1000-2758(2021)05-1057-07

传统的机械臂控制方法几乎都是按照人为预设轨迹来完成特定的任务目标。近些年来随着人工智能技术的发展,将人工智能技术应用在机械臂控制,实现复杂动态环境下的机械臂智能控制成为一个热门的研究方向。

智能控制的目标就是构建出一个能够自主学习适应新环境的系统,强化学习[1]凭借着其自身特点是实现这一目标的关键技术。深度神经网络和强化学习相结合组成的深度强化学习在游戏决策任务上已经取得了非常大的成功。Google DeepMind 团队最早提出深度神经网络与强化学习相结合的深度 Q 网络[1]在 Atari 游戏决策控制上获得了出色的表现,从此开启了深度强化学习的时代。其后又逐渐涌现了能够处理连续动作空间问题的深度确定性策略梯度算法(DDPG)[3]、近端策略优化算法(PPO)[4]、异步优势演员评论家算法(A3C)[5]等强化学习算法模型。

深度强化学习在游戏行为决策任务上表现得非常成功,并增强了强化学习的可解释性[6],很大一部分决定性因素在于游戏环境中奖励函数通常能够直接给出,并且能够直接用来优化。但是在机械臂控制环境中奖励设置往往是当智能体完成某一个任务目标时,环境给予一个正反馈,其他情况下没有反馈。由于智能体起初是随机地在环境中进行探索,绝大多数探索步骤没有奖励回馈,强化学习模型训练时很难收敛,并且当智能体所处的环境发生动态变化时会极大加剧这一状况。为了解决以上 2 个问题,Schaul 等提出了通用价值函数逼近器[8],算法将目标状态作为计算奖励的中间媒介,可以根据不同的目标对当前的状态进行估计,获得状态-目标值函数 $V(s,g|\theta)$,使得智能体学习到从任意状态 s 到达任意目标 g 的策略。受到统一函数逼近器算法的启发,Andrychowicz 等提出了后视经验重现算法(HER)[9],算法可以与任意的离线策略强化学习算法模型相结合,从失败中进行学习,通过不断采样新目标 g'来解决稀疏奖励问题,同时也能够使得模型最终在环境 E 中学习到从任意状态 s 到达任意目标 g 的策略。Hester 等提出了基于示范数据的深度 Q 网络算法模型(DQfD)[10],解决了复杂动态变化环境和稀疏奖励导致传统强化学习算法难以收敛的问

收稿日期:2021-06-03

基金项目:湖北省科技厅重点研发项目(2020BBB092)和湖北省教育厅科学研究计划重点项目(D20192602)资助

作者简介:胡春阳(1975—),湖北文理学院副教授、博士,主要从事云计算、大数据和机器学习研究。

通信作者:史豪斌(1978—),西北工业大学教授,主要从事智能机器人、群机器人协同合作及机器人路径规划与导航研究。

e-mail: shihaobin@nwpu.edu.cn

题。Vecerik 等提出了基于示范数据的深度确定性策略梯度算法模型[11] 填充了 DQfD 不能处理连续动作空间的缺陷,在机械臂控制的仿真实验中有着不错的表现。但是这些算法依赖于精确的环境模型,不能对所处环境自适应感知,并且在大规模状态空间的训练过程中随机探索方案已经不太可行。

针对以上问题,本文采用 YOLO[12] 目标检测算法感知当前环境状态,将环境中目标感知模块与控制系统解耦,直接利用机械臂上方的摄像设备捕捉并计算获得拾取目标位置信息,接下来收集一系列仿真环境下人类操控机械臂的行为作为示范数据,在仿真环境中对机械臂控制强化学习算法模型进行监督学习预训练,即:模仿人类行为学习到部分控制策略[13-14],在此基础上结合 DDPG 与 HER 算法,对仿真环境中的机械臂进行控制,最终实现端到端的机器视觉-强化学习控制模型。

1 背 景

1.1 马尔可夫决策过程

强化学习是机器学习的一个重要分支,主要有基于模型和不基于模型 2 种强化学习类型,最常见的模型是马尔可夫决策过程, 一个马尔科夫决策过程由一个四元组组成, $\boldsymbol{M}=(s,A,p_{sa}^{s'},R)$, $s\in\boldsymbol{S}$ 表示状态空间, $a\in\boldsymbol{A}$ 表示动作空间, $p_{sa}^{s'}$ 为状态转移概率,表示在当前状态 s 下采取动作 a 转移到状态 s' 的概率。对于给定的任意目标 $g\in\boldsymbol{G}$,定义奖励函数 $r_g(s_i,a_i)$,当且仅当在状态 s_i 采取动作 a_i 到达目标 g 时环境会给予一个正反馈,在 t 时刻的状态回报定义为未来奖励的折扣和,即: $R_t=\sum_{i=1}^{T}\gamma^{i-t}r_g(s_i,a_i)$, γ 为折扣系数, $\gamma\in[0,1]$。智能体在环境 E 的探索行为定义为: $\pi:S\rightarrow P(A)$,强化学习的目的就是学习到策略 π 使得在初始状态的期望回报达到最大,策略 π 的具体定义如(1)所示

$$\pi=\arg\max_{\pi}E_{s,a\sim\pi}[R_1] \tag{1}$$

在强化学习算法中使用状态值函数 $V^{\pi}(s)$ 来表示在当前状态 s 按照策略 π 进行探索的期望回报,该值函数具体定义如(2)式所示

$$V^{\pi}(s)=E_{\pi}[R_t\mid s_t=s]=E_{\pi}\left[\sum_{k=1}^{\infty}\gamma^{k}r_{t+k}\mid s_t=s\right] \tag{2}$$

1.2 深度强化学习算法

深度强化学习算法即深度学习与强化学习相结合的产物,DQN[1] 是一个非常具有代表性的非基于模型的深度强化学算法,主要用来解决智能体在离散动作空间的决策问题。DQN 中定义了一个策略网络 Q^e 和一个目标网络 Q^T,策略网络用来估计状态行为值函数 Q^*,行为决策方式如(3)式所示

$$a=\arg\max_{a\in A}Q^{e}(s,a;\theta_e) \tag{3}$$

实际训练中通常使用 ε 贪心策略给予智能体一定的随机探索概率,使得智能体每次以概率 ε 随机选择动作,以 $(1-\varepsilon)$ 的概率策略 π_{Q^e} 选择确定动作 a。训练时在 t 时刻进行探索产生一条经验 $e=(s_t,a_t,r_t,s_{t+1})$ 存储在经验池中,为了使得训练样本间尽可能相互独立,网络训练采用随机策略从经验池中抽取一个批次数据进行训练。定义损失函数: $L=\mathrm{E}(Q(s_t,a_t\mid\theta_e)-y_t)^2$, y_t 的定义如(4)式所示

$$y_t=r_t+\gamma Q^{\mathrm{T}}(s_{t+1},a_{t+1};\theta_T) \tag{4}$$

式中

$$a_{t+1}=\arg\max_{a_{t+1}\in A}Q^{\mathrm{T}}(s_{t+1},a_{t+1};\theta_T) \tag{5}$$

由于 $\mathrm{E}[\max(Q)]>\max\mathrm{E}[Q]$,上述估计 Q 值的方式会产生过估计,Hasselt 等在文献[7] 中重新对(5) 式的行为选择重新进行定义

$$a_{t+1}=\arg\max_{a_{t+1}\in A}Q^{e}(s_{t+1},a_{t+1};\theta_e) \tag{6}$$

在学习率为 α 时损失函数 L 的梯度以及参数 θ_e 的优化如下式

$$\nabla_{\theta}L=\mathrm{E}[Q(s_t,a_t\mid\theta_e)-y_t)\nabla_{\theta}Q(s_t,a_t;\theta)] \tag{7}$$

$$\theta\leftarrow\theta-a\nabla_{\theta}L \tag{8}$$

根据时序差分更新策略,经过 t 轮时间迭代后执行一次目标网络参数 θ_T 的更新: $\theta_T\leftarrow\theta_e$。

传统的深度强化学习算法在解决较为简单的决策任务上具有非常好的效果,但是处理复杂的控制任务时通常会存在稀疏奖励的问题,导致模型难以收敛。因此通常在处理复杂的决策控制问题时会结合 HER 算法来加速模型收敛速度。

2 算法流程

端到端的机械臂自主视觉感知控制算法主要由视觉感知算法模块和决策控制算法模块组成。

1) 视觉感知算法模块

视觉感知模块使用 YOLO-v5 算法,不同于其他

的两阶段系列目标检测算法,YOLO 将物体检测作为一个回归问题求解,算法将输入图像 M 划分成 $n \times n$ 的网格,每个网格负责识别目标中心落在其中的对象,经过一次神经网络 F 的计算推理,便能输出图象中所有物体的位置信息 O、类别信息 C 以及置信概率 P,$M \times F \rightarrow (O,C,P)$。这是一个典型的结构化机器学习算法,根据模型结构,相应的其损失函数也包括三部分:坐标误差 coordError、IOU 误差 iouError 以及分类误差 classError。损失函数 L 的定义如(9)式所示

$$L = \sum_{i=0}^{s^2} \text{coordError} + \text{iouError} + \text{classError} \quad (9)$$

在实验的单个目标识别中,分类误差 classError 表示目标和背景分类误差。视觉感知网络使用在 COCO 数据集上预训练的权重来初始化,在此基础上来训练我们标注的机械臂识别目标数据集。

通过目标识别能够确定当前目标相对于摄像机的具体位置信息 (x_1,y_1,C) 其中 C 为高度固定常量,即载物台相对于机械臂夹口初始状态的高度。接下来使用透视变换算法将其转换为目标相对于载物台具体的坐标信息(x_2,y_2,C)。透视变换是把一个图像投影到一个新的视平面过程,是一个非线性变换,包括:将一个二维坐标系转换为三维坐标系,然后将三维坐标系投影到新的二维坐标系。变换过程如(10)式所示。

$$[x_2,y_2,C] = [x_1,y_1,C] \times \boldsymbol{T} \quad (10)$$

式中,$\boldsymbol{T}$ 为变换矩阵

$$\boldsymbol{T} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad (11)$$

给出 4 个对应像素的坐标点即可求出变换矩阵 $\boldsymbol{T}$。变换结果如图 1 所示。

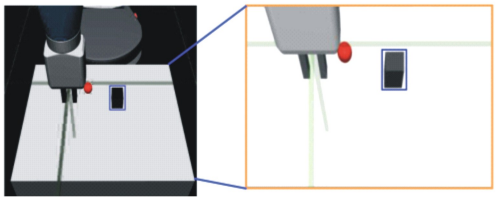


图 1　透视变换得到目标相对于载物台的准确 *XOY* 平面坐标信息

2) 决策控制算法模块

决策控制模块采用 DDPG 强化学习算法,为加速 DDPG 算法的收敛速度,采用模仿学习的方式首先从人类手动控制的经验数据中进行预学习训练,学习到部分初始的控制策略,接下来使用 DDPG 算法让机械臂自主的在环境中学习。DDPG 有 2 个网络结构:一个行为网络(Actor) $\pi_1:\boldsymbol{S} \rightarrow \boldsymbol{A}$ 和一个评估网络(Critic)$\pi_2:\boldsymbol{S} \times \boldsymbol{A} \rightarrow \boldsymbol{R}$,类似于 DQN,Actor 网络由决策估计$\mu(s \mid \theta^{\mu})$ 和决策期望$\mu(s \mid \theta^{\mu'})$ 组成,Critic 网络由估计网络 $Q(s,a \mid \theta^{Q})$ 和目标 $Q(s,a \mid \theta^{Q'})$ 组成。Critic 网络的工作就是评估在当前状态 s,Actor 网络所做出的决策 a 的好坏。对于任意当前输入状态 s_t,通过 Actor 网络选取行为 $a_t = \mu(s \mid \theta^{\mu}) + N_t$,其中 N_t 为随机噪声,执行此行为获得奖励 r_t,接着再使用 Critic 网络对当前状态 s_t 采取行为 a_t 进行打分评估,以此来不断优化 Actor 网络与 Critic 网络,完成整体 DDPG 算法的优化收敛。优化目标如(12)式所示

$$a = \underset{a=\mu(s \mid \theta^{\mu})}{\text{argmax}} Q(s,a \mid \theta^{Q}) \quad (12)$$

优化过程如图 2 所示。

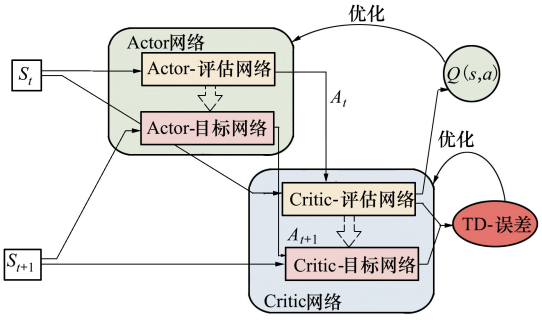


图 2　DDPG 算法整体优化过程

定义 DDPG 的 Critic 网络损失函数 L 如(13)式所示

$$L = E(Q(s_t,a_t \mid \theta^{Q}) - y_t)^2 \quad (13)$$

式中

$$y_t = r_t + \gamma Q'(s_{t+1},\mu'(s_{t+1} \mid \theta^{\mu'}) \mid \theta^{Q'} \quad (14)$$

Actor 网络使用梯度上升的方式优化 θ^{μ},θ^{μ} 梯度求解方式如(15)式所示。

$$\nabla_{\theta^{\mu}} \mid s_t = E(\nabla_a Q(s,a \mid \theta^{Q}) \mid_{s=s_t,a=\mu(s_t)}) = \nabla_a Q(s,a \mid \theta^{Q}) \mid_{s_t,a=\mu(s_t)} \nabla_{\theta^{\mu}} \mu(s_t \mid \theta^{\mu}) \mid_{s=s_t} \quad (15)$$

经历 k 轮优化之后,使用软更新[3] 的策略优化目标网络中的参数,如(16)~(17)式所示。

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \tag{16}$$

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \tag{17}$$

在训练过程为了解决稀疏奖励问题,引入了 HER 算法,DDPG 网络输入的不仅仅是当前时刻的状态 s_t,还包括了要实现的目标 g_t,Actor 网络策略被重新定义为:$\pi: \boldsymbol{S} \times g \rightarrow \boldsymbol{A}$,当且仅当机械臂执行 Actor 网络输出的行为 a_t 到达的下一个状态 s_{t+1} 与 g_t 相等时环境会给予一个奖励。Critic 网络策略重新定义为:$\pi: \boldsymbol{S} \times \boldsymbol{A} \times g \rightarrow \boldsymbol{R}$。机械臂每历经一轮探索,HER 算法便会从历史经验池中进行一次目标采样,产生 m 条新的经验,并且按照(18)式的规则重新计算奖励奖励值 r,将其放到经验池中。最终使得智能体学到了从任意状态 s 到达任意目标 g 的策略,且解决了训练过程中稀疏奖励的问题。算法具体流程如算法 1 所示。

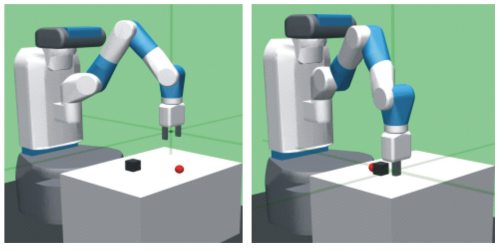
$$\text{succ} = \begin{cases} \text{true} & \text{if} \quad s_{t+1} = g \\ \text{false} & \text{otherwise} \end{cases} \tag{18}$$

算法 1 IL-DDPG-HER 算法

1. 初始化 DDPG 参数: $\theta, \theta', \mu, \mu'$;
   初始化 YOLO 网络参数 m;
   初始化迭代参数 n_1, n_2, n_3, n_4;
   初始化经验回放池 $\boldsymbol{R}$;
2. 创建 YOLO 目标定位训练数据集 $\boldsymbol{S}$;
   创建模仿学习示范数据集 $\boldsymbol{D}$;
// 训练 YOLO 目标定位神经网络
3. for episode = 1 to n_1 do
4. 随机从样本集 $\boldsymbol{S}$ 中抽取一个批次 b;
5. 训练 YOLO 网络参数 m;
6. end for;
// 模仿学习部分
7. for episode = 1 to n_2 do
8. 随机从样本 $\boldsymbol{D}$ 中抽取一个批次 b;
9. 监督学习训练 DDPG 网络参数 θ, μ;
10. end for;
11. 模仿学习训练完成得到初始策略 $\boldsymbol{A}$;
// 强化学习训练部分
12. for episode = 1 to n_3 do
13. for t = 1 to $T-1$ do
14. 摄像设备捕捉输入图像 i;
15. YOLO 网络定位目标所在图像位置;
16. 透视变化算法获取目标坐标信息 s_t;
17. 使用策略 $\boldsymbol{A}$ 获取行为 $a_t = \boldsymbol{A}(s_t \| g)$;
18. 执行 a_t 得到新的状态 s_{t+1},并获得奖励值 r_t;
19. 存储 $(s_t \| g, a_t, r_t, s_{t+1} \| g)$ 到 $\boldsymbol{R}$ 中;
20. HER 算法重新采样新目标,计算奖励值存储到 $\boldsymbol{R}$ 中;
21. end for;
22. for t = 0 to n_4 do
23. 从经验回放池 $\boldsymbol{R}$ 中随机采样一个批次 B;
24. 在 B 上对策略 $\boldsymbol{A}$ 进行优化;
25. end for;
26. end for;

3 实验设计与分析

为验证提出的端到端的识别-控制算法模型,采用 OpenAI Gym Robotics 的 FetchPickAndPlace-v1 机械臂三维空间控制实验仿真环境,如图 3 所示。实验首先通过人为控制机械臂完成相应的拾取-放置任务,收集人类操作经验来让机械臂预学习,接下来按照算法流程的奖励函数设计让其进行自主学习探索,最终学习到自主决策的能力。整个流程可解耦为自主视觉感知和强化学习控制两部分。

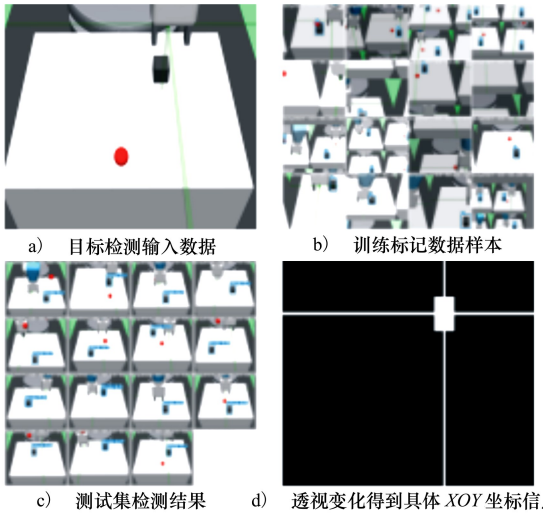


a) 机械臂处于左侧 b) 机械臂处于正位

图 3 机械臂三维空间控制仿真环境

3.1 目标识别与定位实验

仿真环境中摄像设备捕捉到载物台的图像数



a) 目标检测输入数据 b) 训练标记数据样本

c) 测试集检测结果 d) 透视变化得到具体 XOY 坐标信息

图 4 YOLO 目标检测输入数据

据,接下来借助 Roboflow 工具标注创建 YOLO 目标检测网络训练所需数据集,在仿真环境中训练了黑色块目标检测,经过对象识别与定位训练之后,算法模型便能够实现一般环境下黑块目标的识别与定位,最后通过透视变化算法即可获取物体在载物台上具体位置坐标信息,如图 4 所示。由此以来拾取对象位置信息不再依赖于仿真环境主动提供,直接由系统目标识别检测模块获得。实验在 YOLOv5-s 的预训练权重基础上对我们所识别定位的对象进行训练,在 100 个批次训练后 mAP 值、准确率、召回率上都能够达到较好的预期效果,如图 5 所示。

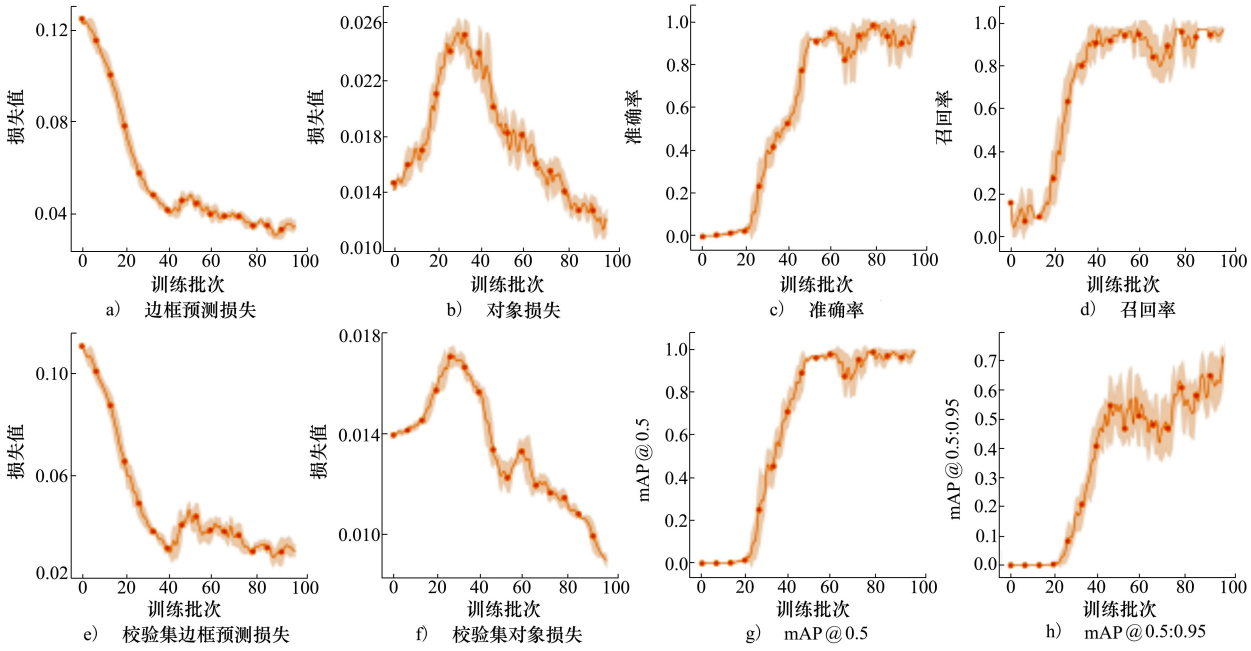


图 5　目标定位损失、目标识别损失、精确度、召回率、校验集目标定位损失、校验集目标识别损失,mAP 0.5 以及 mAP ∈ [0.5,0.95]

3.2　强化学习策略控制实验

实验首先通过人为的收集决策序列 $\tau_1,\tau_2,\cdots,\tau_m$,每个决策序列由状态集和动作集组成:$\tau_i=(s_1^i,a_1^i,s_2^i,a_2^i,\cdots,s_n^i,a_n^i)$,将所有的状态动作序列对抽取出来构造出新的初始经验集合

$$\boldsymbol{D}=\{(s_1,a_1),(s_2,a_2),(s_3,a_3),\cdots\}$$

继而将状态 s 作为输入特征,动作 a 作为输出的预测值,在连续状态空间的机械臂控制任务上当成一个回归问题来求解,使得模型在使用强化学习算法自主学习之前已经具备部分先验知识,以此来加速强化学习算法的收敛速度。最后让机械臂自主的开始在环境中探索,不断学习强化自身决策控制能力。实验对比分析了 IL-DDPG-HER 算法和 DDPG-HER 算法训练智能体执行任务的成功率,如图 6 所示。可以得到 IL-DDPG-HER 算法执行拾取-放置任务上的成功率收敛速度更快。

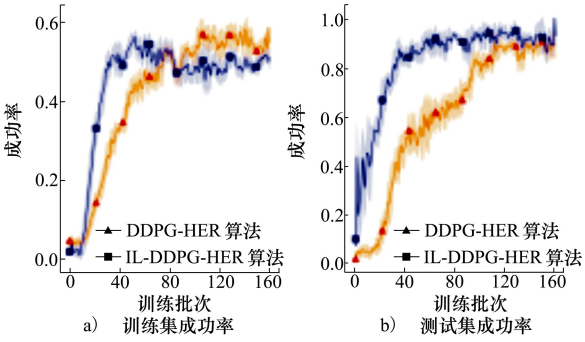


图 6　机械臂拾取-放置任务 IL-DDPG-HER 与 DDPG-HER 成功率实验对比分析

4　结　论

本文通过将计算机视觉技术与强化学习相结合,使得智能体具备自主感知真实环境的能力,这在机械臂拾取-放置任务有着非常重要的意义,尤其

是在适应动态变化环境上,机器视觉-强化学习的端到端控制模型让智能体感知环境的能力与智能决策能力解耦,在应对复杂变化的环境时,可直接对环境感知网络进行重新训练,而决策网络无需做任何改动。并且随着计算机视觉技术的成熟发展,视觉感知模型的训练已经不再是往日的时间"消耗战",往往能够在普通设备上稍作训练即可满足普通的目标定位任务。

未来的视觉感知研究可以加入双目甚至多目摄像头,或者是其它的深度感知传感器,来完成 3D 空间任意位置的目标感知,结合已经训练好的强化学习模型,最终让智能体完成更加复杂的控制决策任务。

参考文献:

[1] LUMELSKY V, STEPANOV A. Dynamic path planning for a mobile automaton with limited information on the environment[J]. IEEE Trans on Automatic Control, 1986, 31(11): 1058-1063

[2] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[J]. Computer Science, 2013, 12(19): 5602

[3] LILLICRAP T P, HUNT J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J/OL]. (2019-07-05)[2021-10-22]. https://arxiv.org/pdf/1509.02971v6.pdf

[4] SCHULMAN J, WOLSKI F, DHARIWAL P, et al. Proximal policy optimization algorithms[J/OL]. (2017-08-28)[2021-10-22]. https://arxiv.org/pdf/1707.06347.pdf

[5] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning[C]//International Conference on Machine Learning, 2016: 1928-1937

[6] LI J, SHI H, HWANG K S. An explainable ensemble feedforward method with Gaussian convolutional filter[J]. Knowledge Based Systems, 2021(225): 107103

[7] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double q-learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2016

[8] SCHAUL T, HORGAN D, GREGOR K, et al. Universal value function approximators[C]//International Conference on Machine learning, 2015: 1312-1320

[9] ANDRYCHOWICZ M, WOLSKI F, RAY A, et al. Hindsight experience replay[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017: 5055-5065

[10] HESTER T, VECERIK M, PIETQUIN O, et al. Deep q-learning from demonstrations[C]//Thirty-Second AAAI Conference on Artificial Intelligence, 2018

[11] VECERÍK M, HESTER T, SCHOLZ J, et al. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards[J/OL]. (2018-10-08)[2021-10-22]. https://arxiv.org/pdf/1707.08817v1.pdf

[12] BOCHKOVSKIY A, WANG C Y, LIAO H. YOLOv4: optimal speed and accuracy of object detection[J/OL]. (2020-04-23)[2021-10-22]. https://arxiv.org/pdf/2004.10934v1.pdf

[13] Wu Y H, Lin S D. A low-cost ethics shaping approach for designing reinforcement learning agents[C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2018

[14] CHRISTIANO P F, LEIKE J, BROWN T B, et al. Deep reinforcement learning from human preferences[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, 2017

Robotic arm reinforcement learning control method based on autonomous visual perception

HU Chunyang[1], WANG Heng[2], SHI Haobin[2]

(1.School of Computer, Hubei University of Arts and Science, Xiangyang 441053, China;
2.School of Computer, Northwestern Polytechnical University, Xi'an 710129, China)

Abstract: The traditional robotic arm control methods are often based on artificially preset fixed trajectories to control them to complete specific tasks, which rely on accurate environmental models, and the control process lacks the ability of self-adaptability. Aiming at the above problems, we proposed an end-to-end robotic arm intelligent control method based on the combination of machine vision and reinforcement learning. The visual perception uses the YOLO algorithm, and the strategy control module uses the DDPG reinforcement learning algorithm, which enables the robotic arm to learn autonomous control strategies in a complex environment. Otherwise, we used imitation learning and hindsight experience replay algorithm during the training process, which accelerated the learning process of the robotic arm. The experimental results show that the algorithm can converge in a shorter time, and it has excellent performance in autonomously perceiving the target position and overall strategy control in the simulation environment.

Keywords: machine vision; reinforcement learning; imitation learning; system simulation; intelligent control

引用格式:*胡春阳, 王恒, 史豪斌. 基于强化学习的机械臂自主视觉感知控制方法[J]. 西北工业大学学报,2021,39(5):1057-1063*

HU Chunyang, WANG Heng, SHI Haobin. Robotic arm reinforcement learning control method based on autonomous visual perception[J]. Journal of Northwestern Polytechnical University, 2021, 39(5): 1057-1063 (in Chinese)