文章编号:0253-2778(2019)10-0812-08

基于渐进式神经网络的机器人控制策略迁移

隋洪建,尚伟伟,李 想,丛 爽

(中国科学技术大学自动化系,安徽合肥 230027)

摘要:在机器人领域,通过深度学习方法来解决复杂的控制任务非常具有吸引力,但是收集足够的机器人运行数据来训练深度学习模型是困难的.为此,提出一种基于渐进式神经网络(progressive neural network,PNN)的迁移算法,该算法基于深度确定性策略梯度(deep deterministic policy gradient,DDPG)框架,通过把模型池中的预训练模型与目标任务的控制模型有机地结合起来,从而完成从源任务到目标任务的控制策略的迁移.两个仿真实验的结果表明,该算法成功地把先前任务中学习到的控制策略迁移到了目标任务的控制模型中.相比于其他基准方法,该算法学习目标任务所需的时间大大减少.

关键词:机器人控制;迁移学习;深度强化学习;渐进式神经网络

中图分类号:TP242 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2019.10.006

引用格式:隋洪建,尚伟伟,李想,等. 基于渐进式神经网络的机器人控制策略迁移[J]. 中国科学技术大学学报,2019,49(10):812-819.

SUI Hongjian, SHANG Weiwei, LI Xiang, et al. Robot control policy transfer based on progressive neural network[J]. Journal of University of Science and Technology of China, 2019,49(10):812-819.

Robot control policy transfer based on progressive neural network

SUI Hongjian, SHANG Weiwei, LI Xiang, CONG Shuang

(*Department of Automation, University of Science and Technology of China, Hefei* 230027)

Abstract: In the field of robotic control, it is appealing to solve complicated control tasks through deep learning techniques. However, collecting enough robot operating data to train deep learning models is difficult. Thus, in this paper a transfer approach based on progressive neural network (PNN) and deep deterministic policy gradient (DDPG) is proposed. By linking the current task model and pretrained task models in the model pool with a novel structure, the control strategy in the pretrained task models is transferred to the current task model. Simulation experiments validate that, the proposed approach can successfully transfer control policies learned from the source task to the current task. And compared with other baselines, the proposed approach takes remarkably less time to achieve the same performance in all the experiments.

Key words: robot control; transfer learning; deep reinforcement learning; progressive neural network

收稿日期:2018-12-24;**修回日期**:2019-05-16

基金项目:国家自然科学基金(51675501)资助.

作者简介:隋洪建,男,1993年生,硕士生,研究方向:机器人迁移学习,E-mail:suihj@mail.ustc.edu.cn

通讯作者:尚伟伟,博士/副教授,E-mail:wwshang@ustc.edu.cn

0　引言

近年来，研究表明利用深度强化学习方法(deep reinforcement learning，DRL)可以训练出性能优异的端到端控制模型[1]. 端到端模型直接以高维感知信息如图片作为输入，并且控制效果往往可以达到或超过人类水平. DeepMind 的研究人员利用深度强化学习训练的智能体[2]，在 57 款雅达利 2600 系列游戏上的得分超过了人类选手[3]. 正如很多其他深度学习技术一样，这些深度强化学习算法的最终性能非常依赖于几百万甚至几千万条高质量训练数据. 在机器人控制领域，收集足够的机器人运行数据是耗时且代价昂贵的. 为了克服这一缺点，人们开始利用迁移学习来缓解数据不足的问题.

参数精调(fine-tuning)是深度学习中一种基本的模型迁移方法. 通过把一个已经训练好的模型(预训练模型)的最后一层或者最后几层的权重在另一个数据集上重新进行训练，使得模型可以迅速适应新的数据分布，该方法广泛应用于不同领域的图片分类，但是，在模型重新训练的过程中，由于很难马上发现先前任务特征与当前任务之间的关系，可能导致部分已学习到的源任务特征的丢失，这种现象也被称为神经网络的“灾难性遗忘”现象. 为了解决这一问题，Rabinowitz 等[4]提出了一种结构新颖的神经网络，这种网络被称为渐进式神经网络(progressive neural network，PNN). 渐进式神经网络将在源任务上训练好的预训练模型保存到模型池中并在目标任务的训练中保持它们的参数不变，训练时通过在模型间构建侧向连接来提取预训练模型中的源任务经验. 二者的区别在于，参数精调只在新模型初始化时利用了预训练模型的权重，在训练过程中这些权重可能被梯度更新所覆盖；而在渐进式神经网络中，预训练模型的权重在整个训练过程中都可以被重复利用；因此渐进式神经网络可以在进行模型间经验迁移的同时，避免出现源任务经验灾难性遗忘.

渐进式神经网络最初被用于与深度 Q 学习相结合来验证其迁移效果. 由于深度 Q 学习是一种基于动作价值函数的方法，因此只能输出离散控制量. 机器人控制算法为了达到更高的精度，一般输出连续控制量. 为了在机器人控制中利用渐进式神经网络，本文提出了一种将渐进式神经网络与深度确定性策略梯度(deep deterministic policy gradient，DDPG)相结合的算法. 首先，在源任务上利用 DDPG 算法训练一个控制模型，将其作为一个预训练模型加入到模型池中；其次，初始化预训练模型与当前任务模型间的侧向连接，建立源任务控制策略到当前控制模型的传输通道；最后，在目标任务上对当前控制模型进行训练，使得当前模型可以提取源任务控制策略中的可迁移特征，从而减小学习当前任务所需的数据量. 实验表明，使用基于渐进式神经网络的迁移算法可以有效减少学习目标任务所需的时间，且在相同的训练时间内，控制模型最终取得的平均奖励值也更高.

1　相关工作

传统的数据处理方法处理高维数据比较困难，因为人工设计的特征工程往往只能提取有限的信息且对噪声鲁棒性差. 近年来，深度神经网络在大规模数据集的处理上表现出了巨大的优势[5-8]，往往能提取出优于人工设计的特征. 研究者们利用这一优势，将其应用于机器人复杂控制策略的学习. Finn 等[9]提出了一种基于深度神经网络的模型预测控制，首先收集 50000 次的机械臂推动盒子中物体的视频以及该过程中末端执行器的当前位姿和命令位姿；然后将这些数据输入一个构建好的深度神经网络并训练，直到其可以根据输入图片和命令位姿预测下一时刻的环境状态；最后设计了一个基于此预测模型的预测控制算法，根据要推动的目标物体生成一系列控制指令. 为了提高控制策略学习的自动化，研究者将深度神经网络和强化学习算法相结合[1,10,11]. 这些算法将强化学习中的动作价值函数或者行为策略通过神经网络进行近似，然后根据价值迭代或者策略迭代的思想设置一个损失函数，以此为依据更新神经网络的参数，最终智能体会学习到一个最优的动作价值函数或者最优的行为策略. 这些基于深度神经网络的控制算法的成功依赖于大规模数据，然而对于机器人来说收集数万条运行数据来进行训练是昂贵且费时的.

迁移学习是缓解深度学习算法应用中数据缺失问题的有效方法. 最简单且广泛使用的迁移学习算法是参数精调，在使用参数精调进行深度模型的迁移时，只需要重新训练模型的最后一层或者最后几层的权重. 参数精调可以被广泛使用，主要是由于图片、语音等数据中存在一些基本模式，如图形中的点、线以及基本的几何形状等，这些特征对于大多数

图片分类任务都是有效的,但机器人控制策略中显然不存在这样的基本模式.在复杂控制策略的参数精调迁移过程中,灾难性遗忘现象的出现使得经验迁移的效率大大下降.针对这一问题,Rabinowitz等[4]提出了渐进式神经网络,通过将预训练模型保存在模型池中来避免灾难性遗忘,同时在目标任务模型和预训练模型间的侧向连接构建可迁移特征的传输通道.实验表明,这种迁移结构可以取得比参数精调更好的迁移效果.Rabinowitz 等[4]将渐进式网络与深度 Q 学习相结合,然而这种算法只能应用于离散控制任务.我们设计了一种新的将渐进式网络与深度确定性策略梯度[12]相结合的方法,使得 Actor 网络可以输出连续控制量,从而极大地扩展了这种算法的应用范围.

2 深度确定性策略梯度

2.1 强化学习背景

强化学习是机器学习中的一种重要方法,强调智能体(agent)优化自身的策略从而获得最大的累积奖励.一般而言,存在一个智能体和智能体所属的环境 E,在每一个时间步 t,环境状态(state)记为 s_t,智能体从环境中接收的观察(observation)记为 o_t,其中通常包含一个即时奖励 r_t. o_t 一般为 s_t 的函数,即 $o_t=f(s_t)$,这取决于环境的可观测性,我们假设环境具有完全可观测性,即 $o_t=s_t$. 智能体允许从一个动作集合 A 中选取一个动作 a_t 并输出到环境 E 中执行.相应地,环境则变化到下一个状态 s_{t+1} 并给出下一状态的即时奖励 r_{t+1}. 一系列的状态和动作记录构成一个序列,即 $\{s_1,a_1,s_2,a_2\cdots s_t\}$,该序列被称为一个马尔科夫决策过程(Markov decision process,MDP).该序列的马尔科夫性体现在,序列中下一个时刻的状态 s_{t+1} 可以由前一个时刻的状态 s_t 以及所采取的动作 a_t 完全决定,而与状态历史 $s_{t-1},s_{t-2}\cdots$ 无关.

强化学习中智能体的目标是持续优化自身策略以获得最大的累积奖励.一般而言,累积奖励总是被回报 G_t (return)所代替.回报被定义为随时间指数衰减的奖励值之和,衰减因子 γ 用来平衡对长期奖励和即时奖励的重视程度.实际上,我们真正关心的是每种状态下不同动作的价值,因此定义在状态 s 下,遵循策略 π,不同动作的价值为

$$Q^{\pi}(s,a)=E_{\pi}\{G_t \mid s_t=s,a_t=a\}, \tag{1}$$

该函数叫作动作价值函数[13].我们称一个动作价值函数 $Q^*(s,a)$ 是最优动作价值函数,当且仅当 $Q^*(s,a)$ 在所有的策略中,在所有状态 s 和动作 a 下取得最大值.最优动作价值函数满足一个重要性质称为贝尔曼方程(Bellman equation)[13],即

$$Q^*(s,a)=E\{r_{t+1}+Q^*(s',a') \mid s_t=s,a_t=a\}. \tag{2}$$

利用贝尔曼方程,可以采用迭代求解的方法求得最优的动作价值函数,这一过程也称为价值迭代.

2.2 深度确定性策略梯度

深度确定性策略梯度是一种不基于环境模型的、适用于连续控制的强化学习算法.设智能体遵循行为策略 π_θ,将该策略用一个函数参数化表示为 $\mu(s\mid\theta^\mu)$,函数参数为 θ^μ,该函数也被称为 Actor.显然,$\mu(s\mid\theta^\mu)$ 是一个从环境状态 s 到智能体动作 a 的映射.我们构建一个策略目标函数 $J(\mu)$ 来衡量行为策略 π_θ 的好坏,$J(\mu)$ 定义为

$$J(\mu)=\int_S \rho^\mu(s)Q^\mu(s,\mu(s))\mathrm{d}s, \tag{3}$$

式中,$\rho^\mu(s)$ 是在行为策略 π_θ 下,环境状态 s 的稳态分布,$Q^\mu(s,\mu(s))$ 则是在同样策略下的动作价值函数.现有工作已经证明策略目标函数 $J(\mu)$ 关于行为策略参数 θ^μ 的梯度为[14]

$$\nabla_{\theta^\mu}J(\mu)=E_{s\sim\rho^\mu}\{\nabla_a Q^\mu(s,a)\mid_{a=\mu(s)}\nabla_{\theta^\mu}\mu(s\mid\theta^\mu)\}, \tag{4}$$

式中,$\nabla_{\theta^\mu}J(\mu)$ 也被称为策略梯度.Actor 函数使用一个神经网络近似,其梯度按式(4)计算.深度确定性策略梯度是一种 Actor-Critic 算法,这意味着它既有一个用于输出执行动作的 Actor,同时有另一个函数 Critic,用于估计当前策略的动作价值函数.使用一个神经网络来近似该 Critic 函数,并根据最优动作价值函数的贝尔曼方程,定义神经网络的损失函数为

$$L(\theta_i^Q)=E\{[r_{t+1}+\gamma Q(s',a'\mid\theta_i)-Q(s',a'\mid\theta_i)]^2 \mid s_t=s,a_t=a\} \tag{5}$$

式中,θ_i^Q 表示第 i 步迭代时 Critic 函数的参数,该损失函数可以通过随机梯度下降法进行优化.总体而言,Critic 函数通过价值迭代的方式进行自身参数的更新,而 Actor 函数根据 Critic 函数传导过来的梯度更新自身参数.

3 渐进式神经网络

如图 1 所示,渐进式神经网络以一系列在源任务上训练好的预训练模型作为初始.当需要迁移到目标任务上时,这些预训练模型被加入到模型池中

并且保持参数不变. 之后，一列新的神经网络被初始化，在预训练模型与新初始化的网络之间构造侧向连接. 设第 k 列网络的第 i 层神经元的输出用 h_i^k 表示，则建立了侧向连接后的 h_i^k 可以表示为

$$h_i^k = f(W_i^k h_{i-1}^k + \sum_{j<k} U_i^{j:k} h_{i-1}^j), \tag{6}$$

式中，W_i^k 是第 k 列网络第 i 层神经元的权重矩阵，$U_i^{j:k}$ 表示从第 j 列网络的第 $i-1$ 层到第 k 列网络第 i 层的侧向连接，$f(\cdot)$ 是第 i 层所采取的激活函数. 不难发现，$U_i^{j:k}$ 实际上是一个将特征向量从 n_{i-1}^j 维投影到 n_i^k 维的投影矩阵. 通过侧向连接，模型在源任务中学习到的特征通过投影变换转化为新任务模型的输入，由此实现任务间经验的迁移.

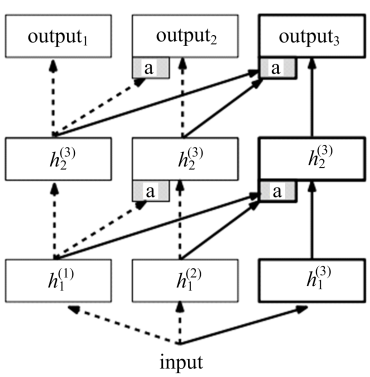


图 1　三列网络组成的渐进式神经网络

Fig. 1　Progressive neural network composed by 3 columns of network

限定侧向连接只能是线性投影变换无疑会限制可迁移特征的范围，为此构造一种新的非线性的侧向连接. 设前 $k-1$ 列网络的所有第 $i-1$ 层的输出构成向量 $h_{i-1}^{<k} = [h_{i-1}^1, h_{i-1}^2 \cdots h_{i-1}^{k-1}]$，其维度为 $n_{i-1}^{<k}$，我们以一个单隐层神经网络代替前面的线性侧向连接. 此时，第 k 列网络的第 i 层输出为

$$h_i^k = f(W_i^k h_{i-1}^k + U_i^k \sigma(V_i^k \alpha_{i-1}^{<k} h_{i-1}^{<k})) \tag{7}$$

式中，$\alpha_{i-1}^{<k}$ 是一个调整系数，用于调整进入单隐层网络的输入的大致范围，V_i^k 是一个投影矩阵，作用是将 $h_{i-1}^{<k}$ 从 $n_{i-1}^{<k}$ 维压缩到 n_{i-1}^k 维，而 $\sigma(\cdot)$ 是单隐层网络的激活函数.

如前所述，深度确定性策略梯度是一种 Actor-Critic 算法，即控制模型中既包含 Actor 网络又包含 Critic 网络. 显然，仅仅迁移 Actor 网络或者 Critic 网络中的源任务特征，都具有经验损失的风险. 因此，设计一种同时迁移 Actor 和 Critic 网络的方法，其结构如图 2 所示. 图中，“300，FC，relu”表示一个包含 300 个神经元、激活函数为 reLU 的网络层，a_dim 表示输出动作维度根据任务确定；虚线箭头表示所代表的权重在训练过程中保持不变. 在 Critic 网络中，动作 a 直到网络的第二层才被输入到动作价值网络中，由于其不包含从源任务学习到的特征，因此没有在输入动作 a 上建立侧向连接.

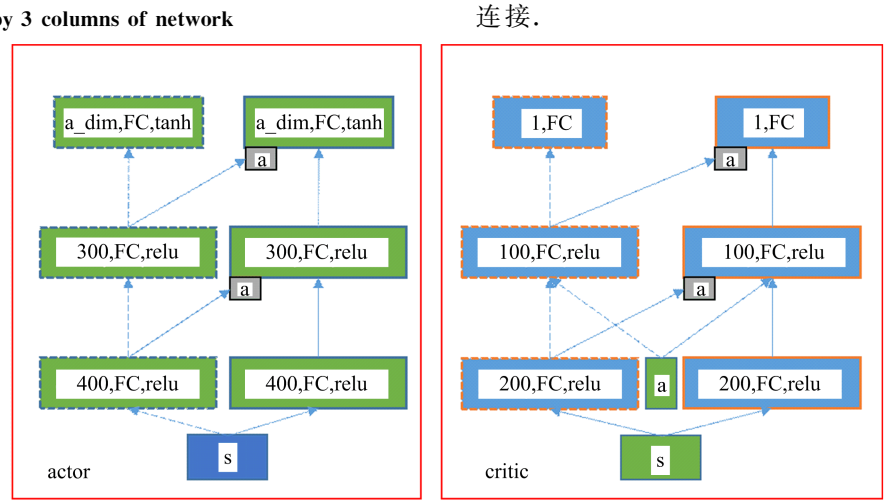


图 2　深度确定性策略梯度与渐进式神经网络结合示意

Fig. 2　The combination of DDPG and PNN

算法步骤可以简要概括为：

（Ⅰ）在源任务上训练一个控制模型，将训练好的模型其加入模型池中，在之后的训练中保持其参数不变.

（Ⅱ）当需要学习一个新任务（目标任务）时，在预训练模型和新建立模型之间建立侧向连接，之前任务中学习到的经验通过侧向连接迁移到新模型，并且迁移的强弱通过侧向连接自动调整.

（Ⅲ）在目标任务环境中对新模型进行训练，直至其可以完成目标任务.

算法流程图如图 3 所示. 算法结合了渐进式神经网络的 DDPG 算法具有以下优势:①与参数精调的迁移方式相比,渐进式神经网络在模型池中存储了预训练模型的全部信息,因此可以在新模型的训练过程中重复利用预训练模型,避免出现灾难性遗忘问题. ②衡量源任务和目标任务的相似性一直是迁移学习的难点,渐进式网络通过侧向连接自动调整从源任务迁移经验的强弱,避免了人为定义任务的相似性. ③在神经网络的每层间都建立了侧向连接,使得低层次任务特征的迁移成为可能,提高了经验迁移的丰富性和灵活性. ④深度确定性策略梯度的 Actor 网络可以输出连续的控制量,使得算法可以应用于连续控制领域,增加了算法的应用范围.

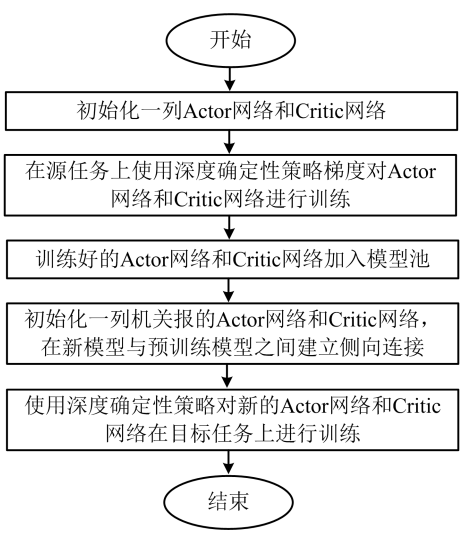


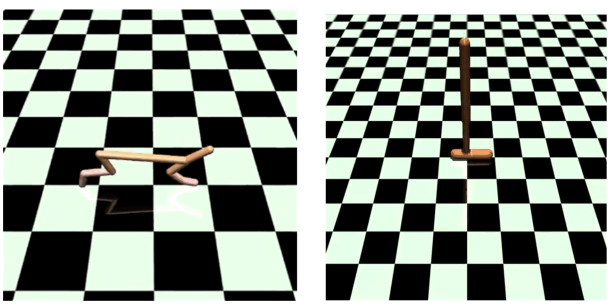
图 3 基于 PNN 的迁移算法的流程图

Fig. 3 Flowchart of the PNN-based transfer algorithm

4 实验

4.1 强化学习测试环境实验

OpenAI Gym 软件包是一款公认的用于测试强化学习算法性能的基准工具, HalfCheetah-v1 和 Hopper-v1 是其中所包含的两个模拟环境,其环境构造如图 4 所示. 图中,HalfCheetah-v1 环境的测试目标是在给定的时间内,使猎豹机器人(HalfCheetah)跑到尽可能远的距离,距离越远,获得的奖励越高;Hopper-v1 环境的测试目标是在保证跳跃机器人(Hopper)不倾倒的前提下,使机器人尽可能跳跃更远的距离,跳跃越远,获得的奖励越高. 在这两个模拟环境中,机器人的动力学特性都具有一定的复杂性,因此训练好的控制模型也并不简单. 要想将训练好的源任务模型中包含的经验迁移到目标任务,迁移算法必须经过精心的设计.



(a) HalfCheetah-v1环境 (b) Hopper-v1环境

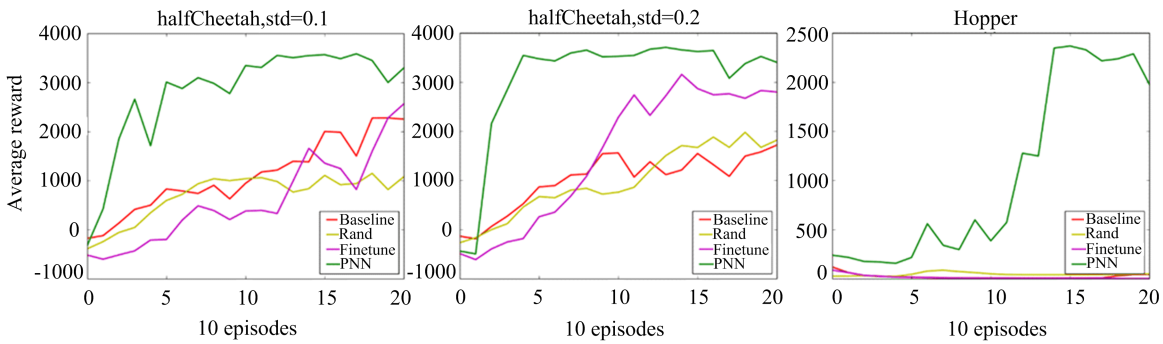
图 4 强化学习测试环境

Fig. 4 Reinforcement learning algorithm test environments

为了构造目标任务,对于 HalfCheetah-v1 中的猎豹机器人,给机器人每个关节的执行器添加均值为零,标准差为 0.1 或 0.2 的高斯噪声,以干扰机器人的运动;对于 Hopper-v1 中的跳跃机器人,将环境中的重力加速度从 $-9.8\ \mathrm{m/s^2}$ 增加至 $-12\ \mathrm{m/s^2}$,以增加机器人进行跳跃的难度. 在本实验中,机器人的各个关节接受连续的控制信号作为控制指令,说明我们的算法具有应用于连续控制任务的能力.

以上强化学习测试环境中进行迁移实验,实验过程遵循第 3 小节中提出的算法步骤. 我们对比了基于 PNN 的迁移算法和如下 3 种方法:①不使用任何迁移算法,在目标任务上重新进行训练,该方法被称为 Baseline;②使用预训练模型的参数初始化新控制模型除了最后一层外其他几层的参数,也就是参数精调,该方法被称为 Finetune;③将目标任务模型与一列普通的神经网络通过侧向连接相结合,该神经网络没有在源任务上经过训练,其参数全部是随机值,该方法被称为 Rand. 本文提出的基于渐进式神经网络的迁移方法在图 5 中被称为 PNN. 以 Actor 网络的迁移为例,四种方法的模型结构分别如图 5 所示,其中深(红)色表示参数来自预训练模型,浅(灰)色表示参数是随机初始化的. 虚线方框表示训练过程中参数不变,无虚线方框则表示训练时参数可变. 记录训练过程中控制模型获得的奖励值并绘制奖励曲线,对每种方法取 4 次实验的平均值绘制平均奖励曲线,如图 6 所示.

由图 6 可知,在 3 组迁移实验中,本文算法的收敛速度快于其他几种方法,且最终取得的平均奖励值也比其他几种方法更高. 如表 1 所示,定量地比较了 4 种算法在目标任务上最终取得的平均奖励值.



(a) HalfCheetah-v1环境　(b)HalfCheetah-v1环境　(c) Hopper-v1环境

图 5　实验中所采用的四种方法的模型结构

Fig. 5　Model structures of the four approaches in the experiment

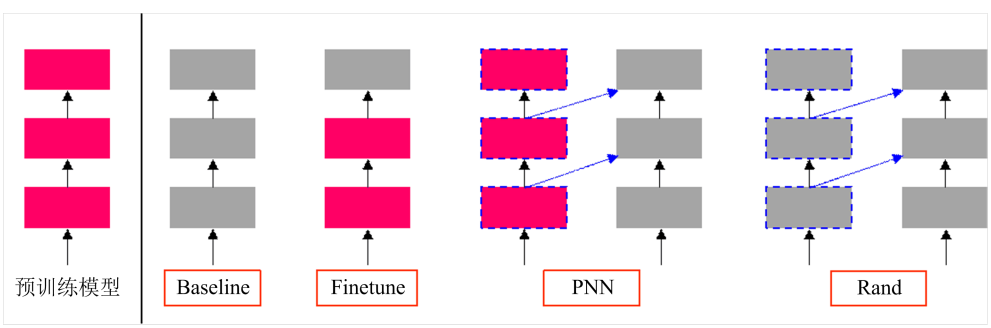


图 6　强化学习测试环境中算法的平均奖励值变化情况

Fig. 6　Reward curve of the 4 approaches in the RL test environments

表 2 展示了 3 组实验中每种方法获得超过 1000 的平均奖励值时所需的最少回合数，可以看到基于 PNN 的迁移方法是所有方法中收敛最快的. 值得注意的是，在表 1 和表 2 中，Finetune 方法有时并不是次优的，在某些指标要上弱于 Baseline 方法和 Rand 方法. 这主要是由于要迁移的源任务控制策略比较复杂，导致训练过程中出现了灾难性遗忘的问题，即在目标任务上进行训练时梯度更新覆盖了源任务模型的一部分有效参数. 在基于 PNN 的迁移方法中，预训练模型的参数在训练过程中是不可变的，因此可以有效防止灾难性遗忘现象的出现. 同时，由于预训练模型中的相应网络层中包含了从源任务提取的有效特征，这些特征在新模型的训练中经过测量连接中的线性投影矩阵被投影到新模型的对应网络层作为输入. 并且投影矩阵中的权重是可训练的，因此迁移的源任务特征可以针对目标任务自动进行调整，以使得 Actor 网络所获得的奖励值最大. 由于利用了源任务的经验，因此我们的算法可以更快地学习目标任务. 以上实验结果表明，基于 PNN 的迁移算法成功地将源任务中的经验迁移到了目标任务的学习中，大大提升了对目标任务的学习效率.

表 1　四种算法最终获得的平均奖励

Tab. 1　The final average reward accomplished by each approach

	HalfCheetah (std=0.1)	HalfCheetah (std=0.2)	Hopper
Baseline	2253.8	1716.7	43.4
Finetune	2563.6	2797.6	1.43
Rand	1078.2	1817.5	48.5
PNN	3295.7	3401.2	1979.0

表 2　四种算法的收敛速度

Tab. 2　The converge speed of each approach

	HalfCheetah (std=0.1)	HalfCheetah (std=0.2)	Hopper
Baseline	110	70	/
Finetune	140	80	/
Rand	80	120	/
PNN	30	30	120

实验中所采用的 4 种模型的可训练参数数量、计算量和单次前向计算时间如表 3 所示. 表 3 中，模型的计算量由进行一次前向运算所需的浮点操作次

数表示，单位为 MFLOPS. 模型的单次前向计算时间是在一台内存容量 8GB，CPU 型号为 Intel(R) Core(TM) i7-6500U 的电脑上测量得到的，分别进行了两组 Batch Size 大小分别为 64 和 128 的实验. 由表 3 可知，基于 PNN 的迁移方法模型的参数数量、计算量和计算时间大概是 Baseline 方法的两倍，这是由于基于 PNN 的迁移算法中包含了一列和当前模型结构相同的预训练模型，因此网络的规模和计算量相比原始模型会增加约一倍. Finetune 方法的可训练参数数量远小于其他方法，这是因为参数精调中只会对源任务模型的最后一层进行训练. 虽然基于 PNN 的迁移方法比 Finetune 方法可训练参数数量和计算量都更大，但该方法可以更有效地进行跨任务经验迁移，由此产生的对目标任务学习的提升作用要超过其计算量增大带来的弊端，因此具有实际应用的价值.

表 3 四种模型的参数数量、计算量和计算时间

Table 3 Number of parameters, FLOPS and computation time of the four models

	参数数量 (10^3)	计算量 (MFLOPS)	计算时间 1 (10^{-3} s)	计算时间 2 (10^{-3} s)
Baseline	294.6	1.152	1.250	1.856
Finetune	1.406	1.152	1.268	1.871
Rand	576.0	2.257	2.603	4.235
PNN	576.6	2.257	2.615	4.191

4.2 仿真环境中的 Baxter 机器人实验

我们选择仿真环境中的 Baxter 机器人控制模块的迁移实验. 实验所采用的 Baxter 研究型机器人如图 7 所示. 实验开始时，机器人的右臂处于一个随机初始化的位姿，控制模型需要控制机器人右臂在工作空间内移动，直到其右臂末端进入目标区域(图中立方体表示)内，并在其中停留超过 50 个时间步. 仿真环境的状态通过一个低维的状态向量表示，其元素包括机器人右臂 7 个关节的角度、角速度、右臂末端点离目标区域中心在 x，y，z 三个方向的距离以及一个表示右臂末端点是否在目标区域内部的布尔值. 实验时，机器人处于速度控制模式下，故行为策略网络的输出是机器人右臂 7 个关节的速度指令值. 为构造目标任务，我们将立方体区域的大小从源任务的 0.2 m × 0.2 m × 0.2 m 减小到目标任务的 0.1 m × 0.1 m × 0.1 m，其他条件保持不变.

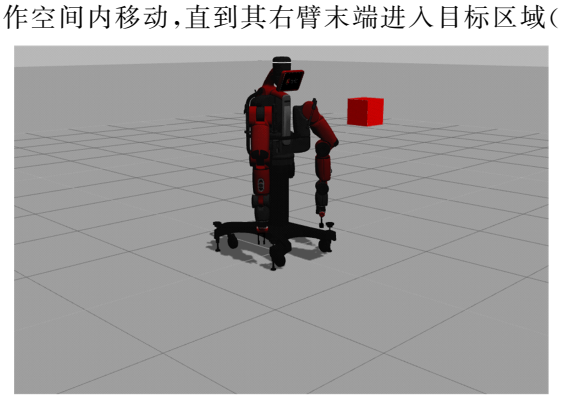


图 7 Gazebo 中的 Baxter 机器人

Fig. 7 Baxter in Gazebo

在目标任务上进行训练的实验结果如图 8 所示，出于时间成本考虑，只比较了本文算法与 Baseline 方法. 由图 8 可以看到，与前面的实验结果一致，本文算法比 Baseline 方法对目标任务的学习更快，说明在源任务和目标任务间进行的经验迁移是成功的，并大大提高了目标任务的学习效率.

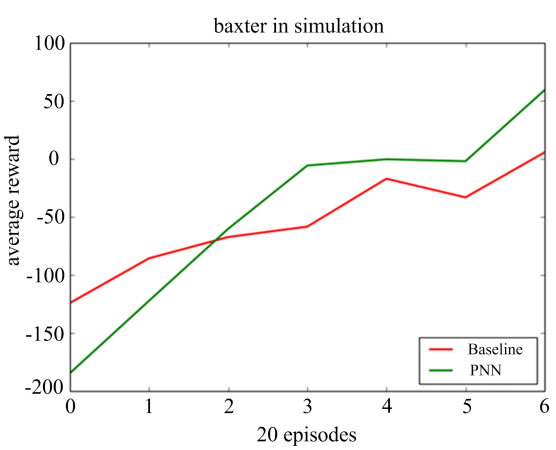


图 8 Baxter 控制实验中两种算法的平均奖励值随回合变化情况

Fig. 8 Reward curve of the 2 approaches in the Baxter control experiment

5 结论

为了有效利用从源任务中学习到的控制策略，克服深度学习技术应用于机器人控制领域效率低下的问题，本文将渐进式神经网络与深度确定性策略梯度算法相结合，提出了一种新的跨任务经验迁移算法，该算法可以显著减少在目标任务上训练时所需的时间. 我们在强化学习测试环境中的猎豹机器人和跳跃机器人的控制任务中评估了所提出的算法，实验结果表明算法成功地将从源任务中学习到的经验迁移到了对目标任务的学习中，提高了对目标任务的学习效率. 此外，在仿真环境中的 Baxter

机器人上进行的控制模型迁移实验表明,本文提出的迁移算法具有迁移实际机器人控制策略的潜力.

参考文献(References)

[1] LEVINE S, FINN C, DARRELL T, et al. End-to-end training of deep visuomotor policies[J]. The Journal of Machine Learning Research, 2016, 17(1): 1334-1373.

[2] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing Atari with deep reinforcement learning[J]. Computer Science, 2013,arXiv:1312.5602.

[3] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529.

[4] RABINOWITZ N C, DESJARDINS G, RUSU A A, et al. Progressive neural networks: U. S. Patent Application 15/396,319[P]. 2017-11-23.

[5] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks [C]//Advances in neural information processing systems. 2012,25(2): 1097-1105.

[6] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [J]. Computer Science, 2014, arXiv preprint arXiv:1409.1556.

[7] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions [C]//Proceedings of the IEEE conference on computer vision and pattern recognition. Boston, USA: IEEE, 2015: 1-9.

[8] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, USA: IEEE, 2016: 770-778.

[9] FINN C, LEVINE S. Deep visual foresight for planning robot motion [C]//IEEE International Conference on Robotics and Automation. Ningbo, China: IEEE, 2017: 2786-2793.

[10] YAHYA A, LI A, KALAKRISHNAN M, et al. Collective robot reinforcement learning with distributed asynchronous guided policy search [C]//IEEE/RSJ International Conference on Intelligent Robots and Systems . Vancouver, Canada:IEEE, 2017: 79-86.

[11] MNIH V, BADIA A P, MIRZA M, et al. Asynchronous methods for deep reinforcement learning [C]//International conference on machine learning. New York, USA: IEEE, 2016: 1928-1937.

[12] LILLICRAP T P , HUNT J J , PRITZEL A , et al. Continuous control with deep reinforcement learning [J]. Computer Science, 2015, 8(6):A187.

[13] SUTTON R S, BARTO A G. Reinforcement Learning: An Introduction[M]. MIT press, 2018.

[14] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms [C]// International Conference on Machine Learning. Beijing, China: IEEE, 2014: 387-395.

(上接第 804 页)

[12] ZOU Q, ZHANG H, WEN C K, et al. Concise derivation for generalized approximate message passing using expectation propagation [J]. IEEE Signal Processing Letters, 2018, 25(12): 1835-1839.

[13] MINKA T P. Expectation propagation for approximate Bayesian inference[C]//Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence. Pittsburgh: Morgan Kaufmann Publishers Inc., 2001: 362-369.

[14] RASMUSSEN CE, WILLIAMS K I. Gaussian Process for Machine Learning[M]. The MIT Press, 2006.

[15] VILA J, SCHNITER P, RANGAN S, et al. Adaptive damping and mean removal for the generalized approximate message passing algorithm [C]//2015 IEEE International Conference on Acoustics, Speech and Signal Processing. Brisbane, Australia: IEEE, 2015: 2021-2025.

[16] CALTAGIRONE F, ZDEBOROVÁ L, KRZAKALA F. On convergence of approximate message passing [C]//2014 IEEE International Symposium on Information Theory. Honolulu, USA: IEEE, 2014: 1812-1816.

[17] SCHNITER P, RANGAN S. Compressive phase retrieval via generalized approximate message passing [J]. IEEE Transactions on Signal Processing, 2015, 63(4): 1043-1055.

[18] BEYME S, LEUNG C. Efficient computation of DFT of Zadoff-Chu sequences[J]. Electronics letters, 2009, 45(9): 461-463.