基于视觉注意力机制的异步优势行动者-评论家算法

李 杰[1,2] 凌兴宏[1,2] 伏玉琛[1,2] 刘 全[1,2,3,4]

(苏州大学计算机科学与技术学院 江苏 苏州 215006)[1]

(苏州大学江苏省计算机信息处理技术重点实验室 江苏 苏州 215006)[2]

(吉林大学符号计算与知识工程教育部重点实验室 长春 130012)[3]

(软件新技术与产业化协同创新中心 南京 210000)[4]

摘 要 异步深度强化学习能够通过多线程技术极大地减少学习模型所需要的训练时间。然而作为异步深度强化学习的一种经典算法,异步优势行动者-评论家算法没有充分利用某些具有重要价值的区域信息,网络模型的学习效率不够理想。针对此问题,文中提出一种基于视觉注意力机制的异步优势行动者-评论家模型。该模型在传统异步优势行动者-评论家算法的基础上引入了视觉注意力机制,通过计算图像各区域点的视觉重要性值,利用回归、加权等操作得到注意力机制的上下文向量,从而使 Agent 将注意力集中于面积较小但更具丰富价值的图像区域,加快网络模型解码速度,更高效地学习近似最优策略。实验结果表明,与传统的异步优势行动者-评论家算法相比,该模型在基于视觉感知的决策任务上具有更好的性能表现。

关键词 异步深度强化学习,视觉注意力机制,行动者-评论家,异步优势行动者-评论家

中图法分类号 TP181 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2019.05.026

Asynchronous Advantage Actor-Critic Algorithm with Visual Attention Mechanism

LI Jie[1,2] LING Xing-hong[1,2] FU Yu-chen[1,2] LIU Quan[1,2,3,4]

(School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China)[1]

(Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, Jiangsu 215006, China)[2]

(Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China)[3]

(Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210000, China)[4]

Abstract Asynchronous deep reinforcement learning (ADRL) can greatly reduce the training time required for learning models by adopting the multiple threading techniques. However, as an exemplary algorithm of ADRL, asynchronous advantage actor-critic (A3C) algorithm fails to completely utilize some valuable regional information, leading to unsatisfactory performance for model training. Aiming at the above problem, this paper proposed an asynchronous advantage actor-critic model with visual attention mechanism (VAM-A3C). AM-A3C integrates visual attention mechanism with traditional asynchronous advantage actor-critic algorithms. By calculating the visual importance value of each area point in the whole image compared with the traditional Cofi algorithm, and obtaining the context vector of the attention mechanism via regression function and weighting function, Agent can focus on smaller but more valuable image areas to accelerate network model decoding and to learn the approximate optimal strategy more efficiently. Experimental results show the superior performance of VAM-A3C in some decision-making tasks based on visual perception compared with the traditional asynchronous deep reinforcement learning algorithm.

Keywords Asynchronous deep reinforcement learning, Visual attention mechanism, Actor-critic, Asynchronous advantage actor-critic

---

到稿日期:2018-05-10 返修日期:2018-08-11 本文受国家自然科学基金项目(61772355,61702055,61303108,61373094,61472262,61502323,61502329),江苏省高等学校自然科学研究重大项目(17KJA520004),吉林大学符号计算与知识工程教育部重点实验室资助项目(93K172014K04,93K172017K18),苏州市应用基础研究计划工业部分(SYG201422),苏州市民生科技项目(SS201736)资助。

李 杰(1994-),男,硕士生,主要研究方向为深度学习、深度强化学习;**凌兴宏**(1968-),男,博士,副教授,主要研究方向为机器学习、强化学习研究,E-mail:lingxinghong@suda.edu.cn(通信作者);**伏玉琛**(1968-),男,博士,教授,CCF 高级会员,主要研究方向为强化学习、人工智能;**刘 全**(1969-),男,博士,教授,博士生导师,CCF 高级会员,主要研究方向为机器学习、智能信息处理。

1 引言

深度强化学习(Deep Reinforcement Learning,DRL)结合了深度学习[1](Deep Learning,DL)和强化学习[2](Reinforcement Learning,RL),是目前人工智能领域的一个新的研究热点。DRL 方法首先利用深度神经网络对高维度数据进行特征表示,再通过强化学习算法在复杂状态任务中进行决策。深度学习可以从高维数据中抽取维度较低且高度可区分的特征,它侧重于对数据的认知和表征;而强化学习更侧重于学习解决问题的策略。因此,为了解决强化学习在高维状态空间任务中的数据预处理问题,Mnih 等[3-4]首次将深度学习的表征能力和强化学习的决策能力相结合,提出了一种深度 Q 网络模型(Deep Q-Network,DQN)。

DQN 模型结合了 DL 中的卷积神经网络(Convolutional Neural Network,CNN)和 RL 中的 Q 学习算法[5],在 Atari 2600 游戏中表现出超过人类玩家的水平,但是容易出现过拟合的训练问题。Van Hasselt 等[6]提出了双重深度 Q 网络模型(Double Deep Q-Network,DDQN),DDQN 模型的关键点在于它使用了两套不同的网络参数来分别进行动作值计算和动作选择,有效缓解了 DQN 模型的过拟合问题。为了充分利用对模型训练更有价值的样本,Schaul 等[7]提出了一种基于优先级重放采样的深度强化学习算法,为经验缓冲池中的样本评判优先级标准,以非等概率方式优先采集缓冲池中的重要样本。然而,经验重放机制也存在一些固有问题,如学习模型需要大量的存储空间来存放训练样本,巨大的计算量对硬件要求很高,以及经验重放机制无法使用如 Sarsa 算法[8-9]的同策略强化学习方法。针对这些问题,Mnih 等[10]将异步方法与深度强化学习方法进行了有效结合,提出了异步深度强化学习(Asynchronous Deep Reinforcement Learning,ADRL),ADRL 替代了传统 DRL 算法的经验重放机制,节省了资源存储的开销,同时降低了模型训练的计算代价。传统的 DRL 算法要求智能体(Agent)在与环境的每一次交互中都更新网络参数,而 ADRL 则是在与环境多次交互后才计算损失,并利用梯度下降算法来更新参数。因此,ADRL 不需要如 GPU 等专门的计算设备来进行训练,它可以在多核 CPU 设备上利用多线程技术加速模型的学习,用更少的时间获得同样优秀的训练效果。作为 ADRL 的一种经典算法,异步优势行动者-评论家算法(Asynchronous Advantage Actor-Critic,A3C)不仅能够利用多线程技术加速训练,还能有效处理连续空间下的决策任务,极大地减少了动作选择的计算成本。

另一方面,基于注意力机制(Attention Mechanism,AM)的深度神经网络被广泛应用于机器翻译[11]、图像识别[12]等领域,训练模型可以通过 AM 自适应选择并提取文本或图像的重点区域,从而提升训练效果。

传统的 A3C 算法虽然利用多线程技术加速了学习模型的训练过程,但是没有将聚焦点集中在一些特定的区域,忽略了整幅图像的重要区域信息。因此,文中提出一种基于视觉注意力机制的异步优势行动者-评论家(Asynchronous Advantage Actor-Critic with Visual Attention Mechanism,VAM-A3C)模型。该模型在基于 CNN 的异步优势行动者-评论家模型中引入了视觉注意力机制[12](Visual Attention Mechanism,VAM),通过 VAM 将 Agent 聚焦于图片中具有重要信息的特定区域或像素位置,通过模型训练不断调整聚焦区域,最终以较少的训练数据和训练时间取得了更好的效果。实验表明,VAM-A3C 在一些 Atari 2600 游戏任务中能够提升传统 A3C 算法的性能。

2 背景知识

2.1 强化学习

强化学习,即激励学习,是一种从环境状态映射到动作的学习方法,激励 Agent 从奖赏反馈中进行自我学习[13]。Agent在与环境交互的过程中可以利用马尔科夫决策过程(Markov Decision Process,MDP)进行建模[2,13-14]。MDP 可以描述为 Agent 在时间步 t 下,根据策略 π 从当前状态 s_t 选择并执行动作 a_t,并以概率 $f(s_t,a_t)$ 转移到下一个状态 s_{t+1},同时获得来自环境反馈的奖赏 r_t,该过程直到环境给出终止条件才结束。Agent 的最终目标是最大化在状态 s_t 所获得的奖赏值,从而获取一个最优的策略 π^*。Agent 从 t 时刻到 T 时刻的累计奖赏 R_t 的计算式如下:

$$R_t=\sum_{t'=t}^{T}\gamma^{t'-t}r_{t'} \tag{1}$$

其中,$\gamma\in[0,1]$为折扣因子,用于权衡未来每个时间步的奖赏对累计奖赏的影响程度。

状态动作值函数是指 Agent 在当前状态 s_t 下执行动作 a_t 所获得的期望回报,计算式如式(2)所示:

$$Q^{\pi}(s,a)=E[R_t\mid s_t=s,a_t=a] \tag{2}$$

一般而言,强化学习方法包括基于模型的强化学习和模型无关的强化学习,模型无关的强化学习包括基于值函数的强化学习方法和基于策略梯度的强化学习方法。策略梯度强化学习方法通过在策略梯度方向上更新参数来改进策略[15-17],行动者-评论家算法[18-20](Actor-Critic,AC)就属于策略梯度强化学习方法。

AC 算法中的行动者与评论家的结构分别是独立的,算法结构图如图 1 所示。AC 算法的描述如下:

1)行动者部分,即策略,Agent 在当前状态下根据策略 π 执行一个动作,环境迁移到下一时间步状态。

2)评论家部分,即值函数,Agent 使用时间差分(Temporal Difference,TD)误差项来评论当前状态所采取的动作的好坏。

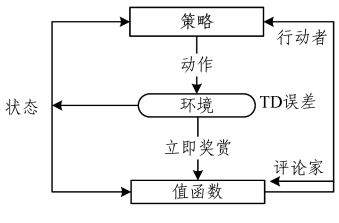

图 1 行动者-评论家(AC)算法结构示意图

Fig. 1 Structure diagram of actor-critic (AC) algorithm

由于 AC 算法是一种策略梯度算法,因此当动作空间连续时,Agent 不需要在无穷的动作空间中选择动作,而是通过对策略的直接更新使模型学习一个确定的随机策略。

2.2　优势行动者-评论家算法

AC 算法与深度神经网络相结合构成了行动者-评论家网络(Actor-Critic Network,ACN),ACN 使得传统 AC 算法能够在大规模状态空间的高维度任务中更加有效地训练。与传统 AC 算法类似,ACN 包括值网络部分和策略网络部分。值网络,即 $V(s;\theta_v)$,其中 θ_v 表示值网络参数;策略网络,即 $\pi(a_t|s_t;\theta)$,其中 θ 表示策略网络参数。ACN 在进行策略更新时,对于每一个状态动作对,都采取相同的权重,平等地利用每一个状态动作对。然而 Agent 在训练过程中,每一个状态动作对的重要性都是不一样的,有些状态动作对能够获得高回报值,而有些状态动作对的回报值则相对偏低。因此,为了充分利用这一有效信息,ACN 引入了一个优势函数,用于评价当前状态动作对的优势,称为优势行动者-评论家算法,表示为 $A(s_t,a_t;\theta,\theta_v)$。优势函数的计算式如式(3)所示:

$$A(s_t,a_t;\theta,\theta_v)=\sum_{i=0}^{n-1}\gamma^i r_{t+i}+\gamma^n V(s_{t+n};\theta_v)-V(s_t;\theta_v) \quad (3)$$

其中,$n=1$ 时,式(3)表示 1 步回报优势函数,$n=k$ 时式(3)表示 k 步回报优势函数。算法的值函数和策略函数的梯度计算如式(4)和式(5)所示:

$$\mathrm{d}\theta_v=\partial(R-V(s_t;\theta_v))^2/\partial\theta_v \quad (4)$$

$$\mathrm{d}\theta=\nabla_\theta\log\pi(a_t|s_t;\theta)A(s_t,a_t;\theta,\theta_v) \quad (5)$$

其中,R 表示 Agent 在状态 s_t 下根据策略 π 选择动作 a_t 所获得的回报值,值函数参数 θ_v 和策略函数参数 θ 通过随机梯度下降方法进行更新。

2.3　视觉注意力机制

注意力机制,又称聚焦机制,是当前深度学习的前沿热点之一,它可以帮助学习模型关注输入的不同部分,从而给出一系列的理解。Xu 等[12]提出的视觉注意力机制被应用于图像识别、图像描述等图像理解任务。下面具体分析 VAM 的计算过程。

1)计算 t 时刻图像各个区域的视觉重要性值:

$$e_{ti}=f_{att}(\boldsymbol{a}_{ti},h_{t-1}) \quad (6)$$

其中,a_t 表示 t 时刻网络模型编码器的输出向量集,$\boldsymbol{a}_{ti}$ 表示图像第 i 个区域位置的输出向量;f_{att} 表示视觉重要性值的计算函数;h_{t-1} 表示 $t-1$ 时刻的隐层状态值。

2)使用 Softmax 回归函数对各区域点的视觉重要性值进行归一化,得到每个区域点的相对视觉重要性:

$$\alpha_{ti}=\frac{\exp(e_{ti})}{\sum_{k=1}^{N}\exp(e_{tk})} \quad (7)$$

其中,N 表示图像的区域总数。

3)根据相对视觉重要性计算出基于注意力机制的上下文向量 $\boldsymbol{C}_t$:

$$\boldsymbol{C}_t=\sum_{k=1}^{N}\alpha_{ti}\boldsymbol{a}_{ti} \quad (8)$$

VAM 模块得到的上下文特征向量 $\boldsymbol{C}_t$ 代表了 a_t 中所有区域点关于视觉权重的一个线性加权。

3　基于视觉注意力机制的异步优势行动者-评论家模型

本节主要阐述 VAM-A3C 模型的具体框架以及处理数据的流程。如图 2 所示,基于视觉注意力机制的异步优势行动者-评论家模型主要由 CNNs,VAM 和 A3C 3 个模块组成。以 Atari 2600 游戏为实验对象,具体分析了 VAM-A3C 中各个模块的作用以及各模块之间的关联性。

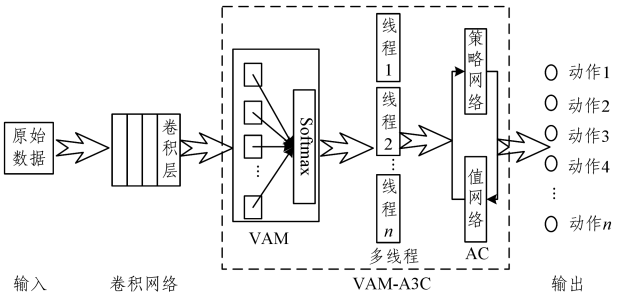

图 2　基于注意力机制的异步优势行动者-评论家(VAM-A3C)模型

Fig. 2　Asynchronous advantage actor-critic model with visual attention mechanism(VAM-A3C)

3.1　预处理

模型在训练 Atari 2600 游戏时,往往需要通过预处理操作来消除游戏过程中不重要的图像边缘信息,最大限度地减少数据的复杂性以及提高特征提取的准确性。本质上,预处理操作就是对游戏中的每一帧图像进行处理。在 Atari 游戏中,游戏画面的尺寸为 210×160,如果将原始数据直接输入到模型中,其所需的计算代价过大,因此模型需要预处理原始图像。首先,将原始的 RGB 图像转换成灰度图;其次,对 210×160 的灰度图进行降采样操作,得到 110×84 的缩略图;最后,剔除图像边界一些无价值的像素点,并裁剪成尺寸为 84×84 的图像。经过灰度转换、降采样和裁剪等预处理操作的游戏画面不仅不会导致价值特征的流失,还能减少模型计算和存储的代价。

3.2　CNNs

卷积神经网络可以对输入图像进行并行化处理,将高维数据编码成一系列低维、抽象的特征表示。VAM-A3C 模型以 4 层 CNN 作为编码器,其具体信息如表 1 所列。

表 1　CNNs 具体信息表

Table 1　Detail of CNNs

卷积层	输入尺寸	过滤器尺寸	步幅尺寸	过滤器数量	激活函数	输出尺寸
Conv1	84 * 84 * 1	3 * 3	2 * 2	64	Relu	42 * 42 * 64
Conv2	42 * 42 * 64	3 * 3	2 * 2	64	Relu	21 * 21 * 64
Conv3	21 * 21 * 64	3 * 3	2 * 2	128	Relu	11 * 11 * 128
Conv4	11 * 11 * 128	3 * 3	2 * 2	128	Relu	6 * 6 * 128

3.3　基于视觉注意力机制的 A3C 算法

在基于视觉感知的 DRL 任务中,Agent 需要在短时间内

完成对输入状态中关键特征的感知并依据特征作出相应的动作。传统的训练模型会生成用于关联编码器和解码器的上下文向量 $\boldsymbol{C}_t$，$\boldsymbol{C}_t$ 表示每个时间步输入数据各相关特征的动态信息，Agent 再依据 $\boldsymbol{C}_t$ 进行决策。然而，Agent 如果将所有的注意力放在整幅图像上，会延缓网络模型的解码速度，使得模型短时间内无法生成利于决策的有用信息。基于上述问题，本文首次将视觉注意力机制引入到 A3C 算法中，以卷积神经网络层输出的输出向量集作为 VAM 输入来重新计算新的上下文向量 $\boldsymbol{C}_t$，t 时刻该输出向量集合为：

$$a_t=\{\boldsymbol{a}_t^1,\boldsymbol{a}_t^2,\cdots,\boldsymbol{a}_t^N\}$$

其中，N 表示图像的区域数量，$\boldsymbol{a}_t^i$ 表示第 i 个区域的输出向量。

与传统的 VAM 不同，VAM-A3C 简化了视觉重要性值的计算，并未利用上一时间步的隐藏状态，而是直接通过当前时刻的输出向量集合来计算图像各区域点的视觉重要性值，其计算式如下：

$$vam(\boldsymbol{a}_t^i)=Linear(Tanh(Linear(\boldsymbol{a}_t^i))) \tag{9}$$

其中，$Linear$ 是一种线性函数，$Tanh$ 是一种非线性变换。

其次，利用视觉重要性值计算各区域点的视觉权重：

$$\alpha_t^i=\exp(vam(\boldsymbol{a}_t^i))/\sum_{k=1}^{N}\exp(vam(\boldsymbol{a}_t^k)) \tag{10}$$

最后，根据图像的输出向量集和各区域的视觉权重计算 t 时刻 Encoder 模块的上下文向量：

$$\boldsymbol{C}_t=\sum_{i=1}^{N}\alpha_t^i\boldsymbol{a}_t^i \tag{11}$$

通过 VAM 重新计算新的上下文向量 $\boldsymbol{C}_t$，使得 Agent 每次都可以自适应地将注意力聚焦在面积较小但具有重要价值的图像区域中，加快网络模型的训练速度。新的上下文特征向量 $\boldsymbol{C}_t$ 是 a_t 中所有区域点的视觉权重与 a_t 的线性加权，以此作为策略网络层和值网络层的输入状态。

策略网络是一个全连接层，其神经元数量与所处理的游戏的动作空间大小相同，该网络根据策略 $\pi(a_t|s_t;\theta')$ 来选择 t 时刻的最优动作；值网络是一个仅包含一个神经元的全连接层。策略网络的输出是每一个动作对应的动作值，经过 Softmax 回归操作后，选择最优的游戏动作；值网络的输出则是状态值，用于计算优势。策略网络和值网络的梯度更新如下所示：

$$d\theta\leftarrow d\theta+\nabla_{\theta'}\log\pi(a_i|s_i;\theta')(R-V(s_i;\theta_v')) \tag{12}$$

$$d\theta_v\leftarrow d\theta_v+\partial(R-V(s_i;\theta_v'))^2/\partial\theta_v' \tag{13}$$

VAM-A3C 模型利用多线程技术加速训练，每一个线程都有各自的网络模型和网络参数，每个线程的网络参数都是从共享网络模型中获取的。线程并不更新自己的网络参数，而是更新共享网络中的参数。VAM-A3C 模型的算法过程如算法 1 所示。

算法 1 基于视觉注意力机制的异步优势行动者评论家(VAM-A3C)算法

Assume global shared parameter vectors θ and θ_v and global shared counter $T=0$

Assume thread-specific parameter vectors θ' and θ_v'

Initialize thread step counter $t\leftarrow1$

repeat

Reset gradients $d\theta\leftarrow0$ and $d\theta_v\leftarrow0$

Synchronize thread-specific parameters $\theta'=\theta$ and $\theta_v'=\theta_v$

$t_{start}=t$

Get input vectors a_t

Accumulate visual importance $vam(a_t)=Linear(Tanh(Linear(a_t)))$

Use softmax for visual importance $\alpha_t^i=\exp(vam(\mathbf{a}_t^i))/\sum_{k=1}^{N}\exp(vam(\mathbf{a}_t^k))$

Accumulate context vector $\mathbf{C}_t=\sum_{i=1}^{N}\alpha_t^i\mathbf{a}_t^i$

Get state $s_t\leftarrow\mathbf{C}_t$

repeat

Perform a_t according to policy $\pi(a_t|s_t;\theta')$

Receive reward r_t and new state s_{t+1}

$t\leftarrow t+1$ and $T\leftarrow T+1$

until terminal s_t or $t-t_{start}==t_{max}$

$$R=\begin{cases}0, & \text{for terminal } s_t\\ V(s_t,\theta_v'), & \text{for non-terminal } s_t\text{//Bootstrap from last state}\end{cases}$$

for $i\in\{t-1,\cdots,t_{start}\}$ do

$R\leftarrow r_i+\gamma R$

Accumulate gradients wrt θ': $d\theta\leftarrow d\theta+\nabla_{\theta'}\log\pi(a_i|s_i;\theta')(R-V(s_i;\theta_v'))$

Accumulate gradients wrt θ_v': $d\theta_v\leftarrow d\theta_v+\partial(R-V(s_i;\theta_v'))^2/\partial\theta_v'$

end for

Perform asynchronous update of θ using $d\theta$ and of θ_v using $d\theta_v$

until $T>T_{max}$

4 实验

本节首先介绍了实验所使用的训练环境和训练过程中所使用的参数设置，然后在 5 种 Atari 2600 游戏中评估了 DQN，FF-A3C 和 VAM-A3C 模型的训练时间和实验效果。其中，FF-A3C 表示仅仅使用 4 层 CNN 的前馈神经网络的 A3C 模型，VAM-A3C 是本文提出的基于视觉注意力机制的 A3C 模型。3 种模型在训练时使用了相同的参数集合。

4.1 实验环境

本文使用 Intel Core i7-6800k CPU 作为硬件环境，以 OpenAI Gym 开源平台中的 Atari 2600 游戏作为实验对象。OpenAI Gym 是一个开源的工具包，用于开发和对比强化学习算法，其涉及了策略、体育竞技和桌游等类型的游戏。

Mnih 等[10]的研究表明，在 Atari 2600 大部分游戏中，FF-A3C 在训练时间和效果上都明显优于基于经验重放机制的 DRL 算法，如 DQN，DDQN 等，因此本文没有将基于经验重放机制的 DRL 算法作为比较训练时间的实验对象，仅仅对比了传统的 FF-A3C 和 VAM-A3C 算法。另一方面，通过 DQN，FF-A3C 和 VAM-A3C 3 种模型对 Gravitar 等 5 种 Atari 2600 游戏进行了性能评估。

4.2 实验参数

为了有效对比 DQN，FF-A3C 和 VAM-A3C 模型的性

能,3 种实验模型均采用相同的实验参数。模型均对原始数据进行了相同的预处理,均采用了 4 层 CNN 作为编码器且 CNN 网络参数均相同。

FF-A3C 和 VAM-A3C 模型的异步更新方式如下:实验均使用 12 个线程来加速模型训练;采用 1000 个训练周期,每 10000 个情节作为一个训练周期,每个情节的临界步数设置为 8000;每 20 帧或情节结束时更新一次共享网络模型的参数;学习率 η 为 0.001,一阶矩估计衰减率 β_1 为 0.9,二阶矩估计衰减率 β_2 为 0.99,超参数 ε 为 0.001,折扣因子 γ 为 0.99。

4.3　实验评估与结果分析

强化学习中,评估实验性能的指标是从情节开始到结束所获得的累计奖赏,即回报值;深度学习中,一般分阶段来训练网络。深度强化学习算法结合了以上两种评估方式,为每一个训练阶段计算出平均情节奖赏,并以此作为评估实验效果的指标。

实验阶段采用了 1000 个训练周期,每 10000 个情节作为一个训练周期,每个情节的临界步数设置为 8000;评估阶段选取了 1000 个情节,计算并比较这 1000 个情节的平均奖赏、最大奖赏以及 100 个情节的平均步数。

本文比较了 FF-A3C 和 VAM-A3C 模型在训练 Agent 玩 Gravitar,Breakout,StarGunner,Seaquest,Gopher 5 种游戏时的每步训练时间,在 Intel Core i7-6800k CPU 上两种模型的训练时间如表 2 所列。表中数据展示了在这 5 种游戏训练过程中 VAM-A3C 模型平均每步的训练时间,该时间与 FF-A3C 模型平均每步的训练时间相差不大。

表 2　两种模型的每步训练时间

Table 2　Training time for each step of two models

(单位:s)

游戏 \ 模型	FF-A3C	VAM-A3C
Gravitar	0.0018	0.0022
Breakout	0.0012	0.0017
StarGunner	0.0016	**0.0011**
Seaquest	0.0029	**0.0015**
Gopher	0.0018	**0.0017**

其中,VAM-A3C 在 StarGunner 游戏上比 FF-A3C 的每步训练时间少 0.0005 s,在 Seaquest 游戏上少 0.0014 s,在 Gopher 游戏上少 0.0001 s。其原因在于:虽然 VAM-A3C 模型需要一定的时间来处理视觉注意力机制,但是注意力机制帮助 Agent 将注意力集中在图像中的关键区域上,避免了 Agent 花费过多时间在价值低的区域,使得 VAM-A3C 的训练时间与传统 A3C 算法相差不大,甚至在 Seaquest 等游戏上的训练时间更少。

此外,本文描述了 FF-A3C 和 VAM-A3C 模型对上述 5 种 Atari 2600 游戏的训练过程,训练结果如图 3 所示。每张游戏训练图的横坐标表示训练周期,纵坐标表示每个周期中情节的平均奖赏数。

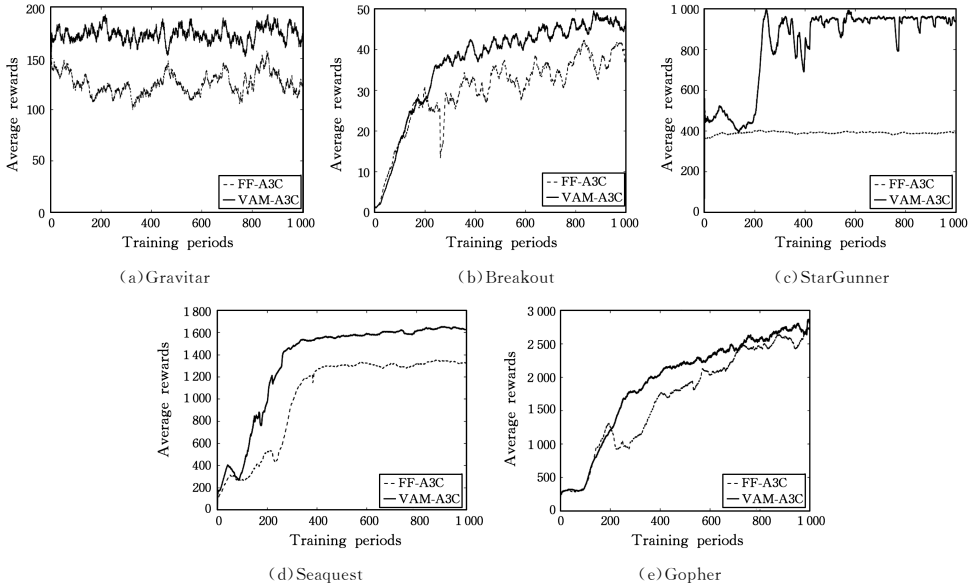

图 3　FF-A3C 和 VAM-A3C 在 5 种游戏上的训练过程

Fig. 3　Training process of FF-A3C and VAM-A3C on five games

由图 3 可知,FF-A3C 和 VAM-A3C 模型在训练 Gravitar,Breakout,StarGunner,Seaquest 和 Gopher 5 种游戏时,利用视觉注意力机制的 A3C 模型比传统的 A3C 模型的训练表现更好,这充分说明了 VAM 的引入可以提升 Agent 的学习能力,可以更好地完成了一些基于视觉感知的决策任务。

本文通过已训练完成的 DQN,FF-A3C 和 VAM-A3C 模型评估并比较了 Gym 平台上 Gravitar,Breakout,StarGunner,Seaquest,Gopher 5 种游戏的实验结果,如表 3 所列。表 3 数据表明:相对于传统 DRL 算法和传统 A3C 模型,VAM-A3C 模型具有更好的实验性能。

评估表 3 结果可知,与传统的 DQN 和 FF-A3C 相比,训练完成后的 VAM-A3C 模型在指导 Agent 玩 Atari 2600 5 种游戏时均取得了一定幅度的提升;同时,从最大奖赏一列中可以看出,训练完成后的 VAM-A3C 模型在 5 种游戏中的最优表现都优于传统 A3C 模型;从平均步数/情节一列中可以看出,VAM-A3C 模型在指导 Agent 玩 Atari 2600 游戏时,

Agent 在 Gravitar,Breakout 和 Seaquest 3 种游戏中每 100 个情节的平均步数均大于另外两种模型。

表 3 3 种模型在 5 种游戏上的实验评估结果

Table 3 Experimental evaluation results of three models on five games

游戏	Agent	平均奖赏	最大奖赏	平均步数/情节
Gravitar	DQN	62.50	272.67	725.00
	FF-A3C	126.33	883.33	1004.63
	VAM-A3C	**174.21**	**1033.33**	**1105.71**
Breakout	DQN	35.08	106.42	1309.75
	FF-A3C	37.12	105.83	1305.01
	VAM-A3C	**47.07**	**120.83**	**1503.76**
StarGunner	DQN	385.08	952.25	1741.67
	FF-A3C	391.50	750.00	1984.61
	VAM-A3C	**949.08**	**1258.33**	1805.66
Seaquest	DQN	1119.74	1602.07	2190.33
	FF-A3C	1325.98	1727.87	2524.83
	VAM-A3C	**1624.32**	**1838.77**	**2816.88**
Gopher	DQN	2720.62	4932.31	2190.36
	FF-A3C	2685.32	6274.08	2308.13
	VAM-A3C	**2806.38**	**6495.78**	2215.45

结束语 异步深度强化学习算法可以利用多核 CPU 进行并行计算。虽然传统的基于前馈网络的异步深度强化学习能够利用多线程技术加速模型的训练,但是传统模型通常会忽略图像中某些具有重要价值的区域信息。为了能够利用并行化计算加强对重要区域信息的处理,本文提出了一种基于视觉注意力机制的异步优势行动者-评论家(VAM-A3C)算法,通过在前馈网络中引入视觉注意力机制将注意力集中在更小但更有价值的图像区域。本文选取 5 种 Atari 2600 游戏验证了 VAM-A3C 的训练时间与传统 A3C 的训练时间相差不大,同时又利用这 5 种 Atari 2600 游戏验证了 VAM-A3C 的实验性能比传统 A3C 更好,表现了 VAM-A3C 模型的优异性。

然而,基于视觉注意力机制的模型在 Atari 2600 平台的某些战略性游戏上表现得并不是很好,原因是其网络结构无法有效记忆不同时间尺度状态之间的依赖信息。因此下一步的研究重点是考虑将注意力机制引入到基于 LSTM,GRU 等循环神经网络的异步深度强化学习算法中,指导 Agent 更有效、快速地学会玩一些其他战略性的游戏。

参 考 文 献

[1] YU K,JIA L,CHEN Y Q,et al. Deep learning:yesterday,today,and tomorrow[J]. Journal of computer Research and Development,2013,50(9):1799-1804.(in Chinese)
余凯,贾磊,陈雨强,等. 深度学习的昨天、今天和明天[J]. 计算机研究与发展,2013,50(9):1799-1804.

[2] SUTTON R S,BARTO A G. Reinforcement learning:An introduction[M]. Cambridge:MIT Press,1998.

[3] MNIH V,KAVUKCUOGLU K,SILVER D,et al. Playing atari with deep reinforcement learning[C]// Proceedings of Workshops at the 26th Neural Information Processing Systems 2013. Lake Tahoe,USA,2013:201-220.

[4] MNIH V,KAVUKCUOGLU K,SILVER D,et al. Human-level control through deep reinforcement learning[J]. Nature,2015,518(7540):529-533.

[5] WATKINS C J C H. Learning from Delayed Rewards[J]. Robotics & Autonomous Systems,1989,15(4):233-235.

[6] VAN HASSELT H,GUEZ A,SILVER D. Deep Reinforcement Learning with Double Q-Learning[C]// Association for the Advance of Artificial Intelligence. 2016:2094-2100.

[7] SCHAUL T,QUAN J,ANTONOGLOU I,et al. Prioritized experience replay[C]// Proceedings of the 4th International Conference on Learning Representations. San Juan,Puerto Rico,2016:322-355.

[8] RUMMERY G A,NIRANJAN M. On-line Q-learning using connectionist systems[D]. Cambridge:University of Cambridge,1994.

[9] SUTTON R S. Generalization in reinforcement learning:successful examples using sparse coarse coding[C]// International Conference on Neural Information Processing Systems. MIT Press,1995:1038-1044.

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

[11] BAHDANAU D,CHO K,BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. arXiv:1409.0473,2014.

[12] XU K,BA J,KIROS R,et al. Show,attend and tell:Neural image caption generation with visual attention[C]// International Conference on Machine Learning. 2015:2048-2057.

[13] BUSONIU L,BABUSKA R,DE SCHUTTER B,et al. Reinforcement learning and dynamic programming using function approximators[M]. CRC Press,2010.

[14] WIERING M,OTTERLO M V. Reinforcement Learning:State-of-the-Art[M]. Springer Publishing Company,Incorporated,2012.

[15] SUTTON R S,MCALLESTER D A,SINGH S P,et al. Policy gradient methods for reinforcement learning with function approximation[C]// Advances in neural information processing systems. 2000:1057-1063.

[16] KAKADE S. A natural policy gradient[C]// International Conference on Neural Information Processing Systems:Natural and Synthetic. MIT Press,2001:1531-1538.

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

[18] KONDA V R,TSITSIKLIS J N. Actor-critic algorithms [C]// Advances in Neural Information Processing Systems. 2000:1008-1014.

[19] BHATNAGAR S,GHAVAMZADEH M,LEE M,et al. Incremental natural actor-critic algorithms[C]// Advances in Neural Information Processing Systems. 2008:105-112.

[20] KONDA V R,TSITSIKLIS J N. Actor-critic algorithms[C]// Advances in Neural Information Processing Systems. 2000:1008-1014.