# 基于注意力的循环 PPO 算法及其应用

吕相霖[1,2],臧兆祥[1,2],李思博[1,2],王俊英[1,2]
(1. 三峡大学 水电工程智能视觉监测湖北省重点实验室,湖北 宜昌 443002;
2. 三峡大学 计算机与信息学院,湖北 宜昌 443002)

**摘 要**:针对深度强化学习算法在部分可观测环境中面临信息掌握不足、存在随机因素等问题,提出了一种融合注意力机制与循环神经网络的近端策略优化算法(ARPPO 算法)。该算法首先通过卷积网络层提取特征;其次采用注意力机制突出状态中重要的关键信息;再次通过 LSTM 网络提取数据的时域特性;最后基于 Actor-Critic 结构的 PPO 算法进行策略学习与训练提升。基于 Gym-Minigrid 环境设计了两项探索任务的消融与对比实验,实验结果表明 ARPPO 算法较已有的 A2C 算法、PPO 算法、RPPO 算法具有更快的收敛速度,且 ARPPO 算法在收敛之后具有很强的稳定性,并对存在随机因素的未知环境具备更强的适应力。

**关键词**:深度强化学习;部分可观测;注意力机制;LSTM 网络;近端策略优化算法
**中图分类号**:TP242.6 **文献标识码**:A **文章编号**:1673-629X(2024)01-0136-07
doi:10.3969/j.issn.1673-629X.2024.01.020

# Attention-based Recurrent PPO Algorithm and Its Application

LYU Xiang-lin[1,2], ZANG Zhao-xiang[1,2], LI Si-bo[1,2], WANG Jun-ying[1,2]
(1. Hubei Key Laboratory of Intelligent Visual Monitoring for Hydropower Engineering, Three Gorges University, Yichang 443002, China;
2. School of Computer and Information, Three Gorges University, Yichang 443002, China)

**Abstract**: A proximal policy optimization model based on attention mechanism and recurrent neural network (ARPPO) is proposed to address the problems faced by deep reinforcement learning algorithms in partially observable environments, such as insufficient information about the environment and randomness factors. The algorithm first processes the encoded information of environmental images through convolutional network layers; then highlights important key information in states using attention mechanism; then extracts temporal characteristics of data through LSTM network; finally improves policy learning and training based on PPO with Actor-Critic structure. Ablation and comparative experiments of two exploration tasks were designed based on the Gym-Minigrid environment. The experimental results show that ARPPO has faster training speed and stronger stability compared with A2C, PPO and RPPO, and has stronger adaptability to unknown environments with random factors.

**Key words**: deep reinforcement learning; partially observable; attention mechanism; LSTM network; proximal policy optimization algorithm

## 0 引 言

未知环境中的智能决策过程又称为部分可观测马尔可夫决策过程(POMDP),智能体通过掌握局部环境的观测信息进行问题分析与建模并智能化地做出后续决策。POMDP 问题符合现实中很多实际应用,并且现已被广泛用于军事兵力推演[1-2]、自动驾驶[3-4]、资源调度[5-6]、机器人控制[7-11]、游戏[12-13]等领域。

目前在 POMDP 下构建状态的方法主要有使用历史信息、信念状态和循环神经网络。王学宁等人[14]提出了基于记忆的强化学习算法 CPnSarsa($\lambda$),通过对状态进行重新定义,智能体结合历史信息来区分混淆状态。在部分可观测环境中,信念状态[15](belief,表示隐状态的分布)常被认为是具有马尔可夫性,根据这一特点,Egorov[16]使用 POMDP 任务的信念状态作为 DQN 输入对策略进行求解。Meng Lingheng 等[17]通过将记忆引入 TD3 算法,提出了基于长短时记忆的双

**收稿日期**:2023-04-01 **修回日期**:2023-08-03
**基金项目**:国家自然科学基金(61502274);湖北省自然科学基金(2015CFB336)
**作者简介**:吕相霖(1998-),男,硕士研究生,研究方向为强化学习;通讯作者:臧兆祥(1985-),男,博士,副教授,CCF 会员(40125M),研究方向为机器学习、决策智能、计算机游戏智能。

延迟深度确定性策略梯度算法(LSTM - TD3)。Matthw Hausknecht[18]等通过将长短期记忆与深度 Q 网络相结合,修改 DQN 以处理噪声观测特征。刘剑锋等人[19]在 DDQN 算法中引入对比预测编码(CPC)通过显式地对信念状态进行建模获取历史的地图编码信息进行训练。耿俊香等人[20]将注意力机制引入到多智能体 DDPG 算法的价值网络中,有选择地关注来自其他智能体的信息,使其在复杂的环境中成功实现智能体间合作、竞争等互动。刘国名等学者[21]尝试了将智能体与环境交互收集到的环境信息经过卷积神经网络处理后输入到 LSTM 神经网络,利用历史信息引导智能体的探索起到了很好的效果,但在收敛速度上仍存在着不足。在此基础上,该文提出了一种融合注意力机制与循环神经网络的深度强化学习算法(即 ARPPO 算法)进行 POMDP 的探索任务研究。实验结果表明 ARPPO 算法在存在动态改变的 POMDP 环境中有着更强的探索能力与适应性,且收敛速度较已有的 A2C,LSTM-PPO 等算法更快。

## 1　相关技术

### 1.1　LSTM 神经网络

循环神经网络(RNN)由于在当前时间片会将前一时间片的隐状态作为当前时间片的输入,故在时序数据的处理上表现优异。LSTM 神经网络是一种改进的 RNN,主要用于解决 RNN 存在的长期依赖问题。它通过引入 3 个门控结构和 1 个长期记忆单元控制信息的流通和损失,从而避免梯度消失和梯度爆炸问题,其结构如图 1 所示。

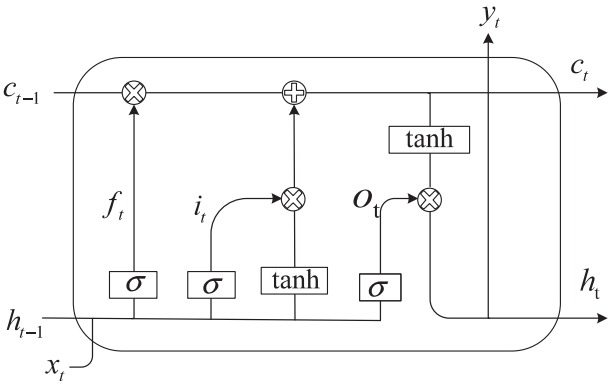



图 1　LSTM 网络结构

其中,$f$ 表示遗忘门,$i$ 表示输入门,$o$ 表示输出门,$c$ 表示记忆细胞状态。前一时间的隐状态 $h_{t-1}$ 与序列 $x_t$ 输入到网络中,$y_t$ 为网络最终的输出结果,同时更新隐状态和记忆细胞状态。其计算公式如式 1 ~ 式 5 所示。

$$f_t = \sigma(\boldsymbol{W}_f x_t + \boldsymbol{U}_f h_{t-1} + \boldsymbol{b}_f) \tag{1}$$

$$i_t = \sigma(\boldsymbol{W}_i x_t + \boldsymbol{U}_i h_{t-1} + \boldsymbol{b}_i) \tag{2}$$

$$o_t = \sigma(\boldsymbol{W}_o x_t + \boldsymbol{U}_o h_{t-1} + \boldsymbol{b}_o) \tag{3}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tanh(\boldsymbol{W}_c x_t + \boldsymbol{U}_c h_{t-1} + \boldsymbol{b}_c) \tag{4}$$

$$h_t = o_t \otimes \tanh(c_t) \tag{5}$$

其中,$\boldsymbol{W}_f$,$\boldsymbol{W}_i$,$\boldsymbol{W}_o$,$\boldsymbol{U}_f$,$\boldsymbol{U}_i$ 和 $\boldsymbol{U}_o$ 表示权重矩阵;$\boldsymbol{b}_f$,$\boldsymbol{b}_i$,$\boldsymbol{b}_o$ 和 $\boldsymbol{b}_c$ 为偏置向量;$\sigma$ 代表 Sigmoid 激活函数;$\otimes$ 表示哈达玛积;tanh 为双曲正切函数。

### 1.2　注意力机制

自注意力机制利用特征本身固有的信息进行注意交互。神经网络通过引入自注意力机制,解决了模型信息过载的问题,提高了网络的准确性和鲁棒性。自注意力机制的计算分为两个部分,第一部分是计算输入的序列信息中任意向量之间的注意力权重,第二部分是根据所得注意力权重计算输入序列的加权平均值,图 2 为自注意力机制原理。

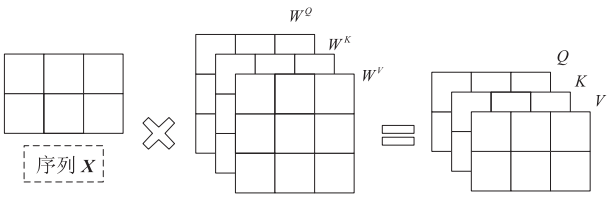



图 2　自注意力机制原理

其中,$\boldsymbol{X}$ 表示输入的序列数据,其详细计算公式如式 6 ~ 式 9 所示。

$$\boldsymbol{Q} = \boldsymbol{X}\boldsymbol{W}^Q \tag{6}$$

$$\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}^K \tag{7}$$

$$\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}^V \tag{8}$$

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{\dim}}\right)\boldsymbol{V} \tag{9}$$

其中,$\boldsymbol{Q}$,$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 分别表示查询矩阵、键矩阵和值矩阵,它们由输入的 $X$ 分别与对应的权重矩阵相乘所得,$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})$ 由 $\boldsymbol{Q}$ 与 $\boldsymbol{K}$ 矩阵的转秩相乘的结果除以 $\boldsymbol{Q}$,$\boldsymbol{K}$ 和 $\boldsymbol{V}$ 维数的平方根,然后乘以矩阵 $\boldsymbol{V}$ 所得。

多头注意力能够使模型在多个不同位置上关注到更多来自不同子空间的信息,最后将各空间所得信息进行拼接,能够更好地对重要信息增加权重,其计算公式为式 10 和式 11,$\boldsymbol{W}^O$ 表示计算头部注意力实例线性变换的矩阵。

$$\text{head}_i = \text{Attention}(\boldsymbol{Q}\boldsymbol{W}_i^Q,\boldsymbol{K}\boldsymbol{W}_i^K,\boldsymbol{V}\boldsymbol{W}_i^V) \tag{10}$$

$$\text{Multi}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \text{Concat}(\text{head}_i,\cdots,\text{head}_h)\boldsymbol{W}^O \tag{11}$$

### 1.3　近端策略优化算法

在深度强化学习领域中,通常将无模型的深度强化学习算法分为 Q 值函数方法和策略梯度算法[22]。近端策略优化算法(Proximal Policy Optimization,PPO)属于策略梯度算法,其原理是将策略参数化,通过参数化的线性函数或神经网络表示策略。

PPO 算法其中的一个核心是重要性采样,主要目的是用于评估新旧策略的差别有多大,重要性采样比很大或者很小就会限制新策略,不能让新策略和旧策略偏离太远,其公式如式 12 所示。

$$r(\theta)=\frac{\pi_{\theta}(a\mid s)}{\pi_{\theta_{old}}(a\mid s)} \tag{12}$$

另一个核心是梯度裁剪,PPO 算法的目标函数表达式为:

$$L^{\mathrm{CLIP}}(\theta)=E[\min(r(\theta)A,\mathrm{clip}(r(\theta)))] \tag{13}$$

$$A=Q(s,a)-V(s,a) \tag{14}$$

其中,$\theta$ 为策略参数,$A$ 为优势函数,$Q(s,a)$ 代表在状态 $s$ 下采取动作 $a$ 的累积奖励值,$V(s,a)$ 为状态估计值。clip 为裁减函数,梯度裁剪的作用则是使各动作的概率分布保持相近,基于上限 $1+\varepsilon$ 与下限 $1-\varepsilon$ 处进行截断操作,以此避免策略更新出现较大差异。PPO 算法的参数更新公式如下:

$$\theta\leftarrow\arg\max_{\theta}(E[L^{\mathrm{CLIP}}(\theta)]) \tag{15}$$

通过基于优势函数的 Actor-Critic 方法进行回报值估计,则会产生方差较小而偏差较大的问题。该文采取的 PPO 算法采用了泛化优势估计(GAE)权衡方差和偏差的问题,公式为:

$$A_t^{\mathrm{GAE}}=\sum_{l=0}^{T}(\gamma\lambda)^l\delta_{t+l} \tag{16}$$

$\lambda=0$ 时,advantage 的 GAE 表示退化成时序差分方法(one-step TD);$\lambda=1$ 时,advantage 的 GAE 表示退化成蒙特卡洛方法;$\lambda$ 在(0,1)区间时,表示在偏差和方差之间做出折衷。

## 2 融合注意力与 LSTM 的 ARPPO 模型

如图 3 所示,融合注意力机制与 LSTM 网络的近端策略优化算法主要分为 4 个模块,即卷积网络模块、注意力模块、长短时记忆网络模块和 PPO 算法模块。

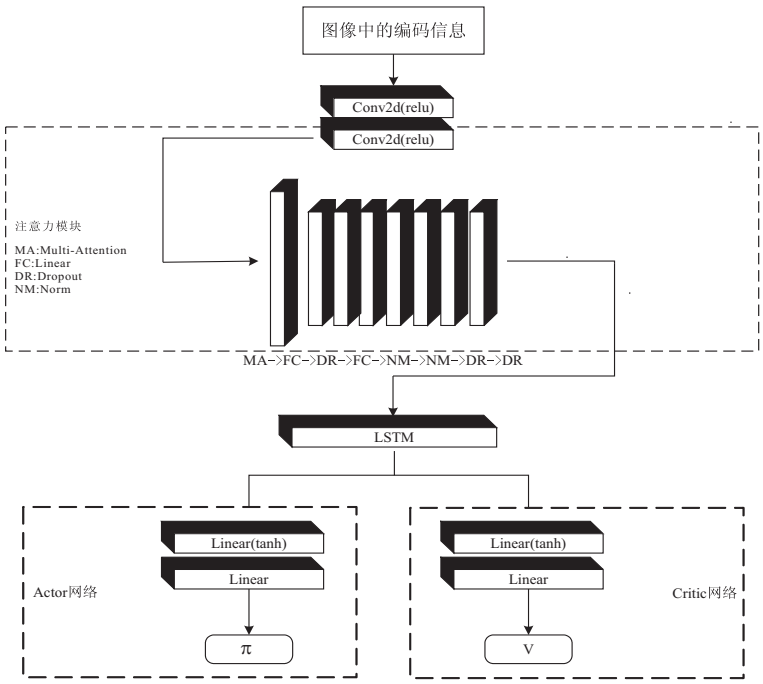



图 3 ARPPO 模型

具体步骤如下:

(1)对智能体与环境交互获取的图像编码信息进行卷积处理后提取特征。

(2)将提取的特征输入到注意力网络,捕捉信息的关联性,一定程度上实现多变量解耦或部分解耦。

(3)将注意力网络输出的数据信息,引入 LSTM 网络提取数据的时域特性。

(4)分别输入到强化学习的 Actor-Critic 框架中进行策略提升与训练。

卷积网络模块对图像编码信息进行特征提取,考虑到计算复杂度与过拟合问题,设计了两层卷积网络提取数据的深层多维信息。第一层卷积网络输入通道数为 3,输出通道数为 32,卷积核大小为 4,步长为 1。第二层卷积网络输入通道数为 32,输出通道数为 64,卷积核大小为 4。

注意力编码模块由多头注意力网络、全连接层、dropout 层和 batch-norm 层组成。多头注意力网络中采用多头数为 8。第一层全连接网络使用 64 个输入通道和2 048个输出通道。第二层全连接网络使用 2 048个输入通道和 64 个输出通道。卷积输出的信息进入注意力网络层进行权重叠加,并使用全连接层进行数据调整。两层 norm 使用的 eps 值为 $10^{-5}$。并且模型使用了 dropout 层防止出现过拟合现象。

PPO 算法基于 Actor-Critic 框架,其中 Actor 网络

通过输入处理后的特征信息获取当前各项动作选取的概率数组,Critic 网络对当前所处状态进行评价与估量,返回一个状态评估值。Actor 网络中的第一层全连接层的输入通道数为 64,输出通道数为 64。第二层全连接层输入通道为 64,输出通道为 7。Critic 网络中的第一层全连接层的输入通道数为 64,输出通道数为 64。第二层全连接层输入通道为 64,输出通道为 1。

## 3　实验设计

### 3.1　实验环境

为验证所提出的 ARPPO 算法基于部分可观测环境的训练效果与学习情况,采用 Gym-Minigrid[23] 网格环境。该环境中智能体在导航时仅能获取其朝向方向 7×7 大小的图像编码信息,且无法感知墙壁后方信息。该文基于 Minigrid 已有的环境做出改动,设计了 Empty-16×16-v1 和 FourRooms-v1 两种不同难度的地图环境,旨在验证算法对于动态变化环境的性能与表现。

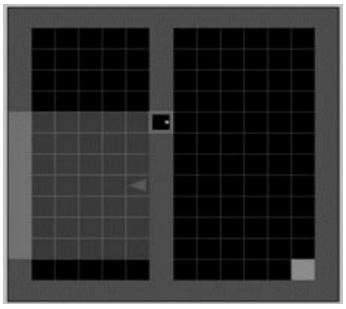


图 4　Empty-16×16-v1

图 4 为改进的环境 Empty-16×16-v1,智能体在障碍物左上侧位置上随机初始化朝向,智能体仅有的视野范围内学会在相应位置保持正确朝向并行进,且需要在受中间围墙的视野影响下学会找到围墙中间区域出现的门并且学会开门动作,获取围墙另一侧的环境信息,最终找到右下方的目标点。并且每一回合产生的门位置是随机变化的。图 5 为改进的环境 FourRooms-v1,智能体同样位于左上角位置朝向随机,智能体需要在仅有的视野范围内离开左上方的房间并且找到右下角的目标点,不同的是该环境存在更多的动态变化因素,每一回合地图中的四堵墙的缺着口是变化的,这为智能体探索目标点带来了相应的困难,该环境旨在测验算法应对动态环境的可适用性。

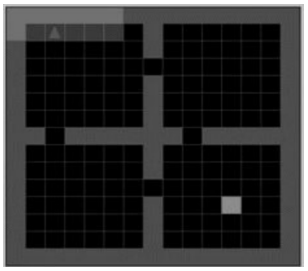


图 5　FourRooms-v1

### 3.2　奖励设计

奖励是对每回合智能体与环境交互产生的回报。该文设计了一种随步数变化而变化的奖励函数,旨在引导智能体在一个 episode 内以更少的完成步数获取更高的奖励值,避免出现局部收敛使得智能体停止探索任务的情况。具体如式 17 所示。

$$\text{reward}=\begin{cases}1-0.9*\dfrac{a_{\text{step}}}{a_{\text{maxstep}}}, & \text{if } a_{\text{step}}\leqslant a_{\text{maxstep}}\\ 0, & \text{if } a_{\text{step}}>a_{\text{maxstep}}\end{cases}\tag{17}$$

### 3.3　训练过程与结果分析

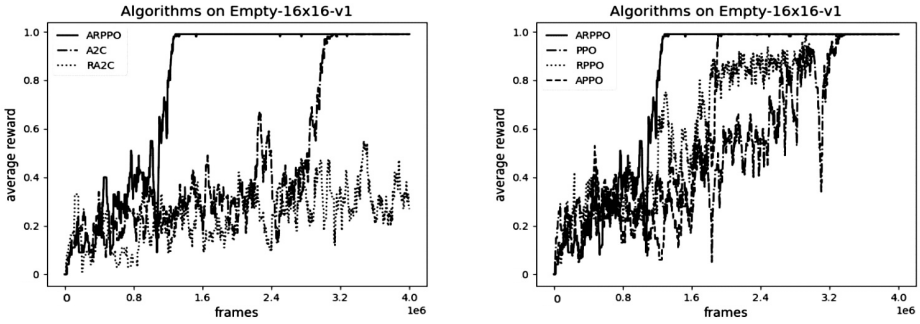
实验采用 Ubuntu18.04,Python 版本为 3.9,基于 torch1.13 搭建的深度强化学习框架。实验设备为含有两张显存大小为 8G 的 GTX 1080 显卡的服务器。为验证所提出的 ARPPO 算法的性能表现,设计了 ARPPO 算法的消融实验,证明并非仅因 LSTM 网络或注意力机制使得算法效果提升。同时也选择了 A2C 算法与 RA2C 算法(A2C-LSTM 算法)进行对比实验,由于 ARA2C 算法在实验过程中表现效果很差,通过实验测试在两张地图上均不能收敛,故不作为该项实验的比较算法。除特定的注意力编码器和循环神经网络参数以外,所有算法都共用相同的参数:迭代次数为 4,训练批大小为 256,学习率为 0.001,折扣率为 0.99,采用 optimizer 优化器。循环神经网络的 recurrence 参数设置为 4。

基于 Actor-Critic 框架的网络层采用 tanh 激活函数,其余使用了激活函数的网络层均为 relu 激活函数。三种算法结构均使用相同的帧编码器,表 1 总结了六种算法结构的异同之处。

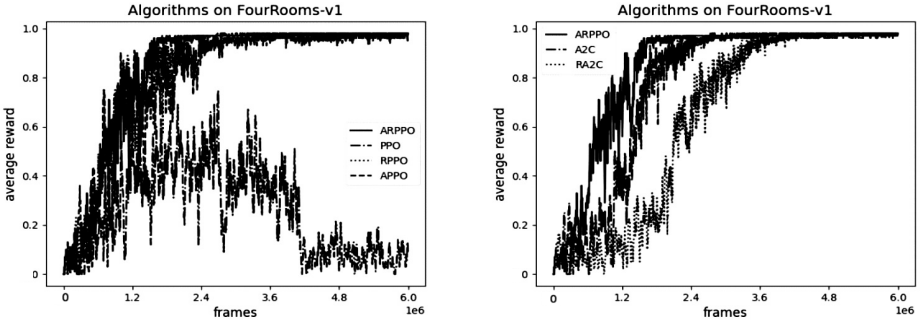
表 1　各算法结构设置

| 结构 | PPO | RPPO | APPO | ARPPO | A2C | RA2C |
|---|---|---|---|---|---|---|
| 帧编码器(CNN) | √ | √ | √ | √ | √ | √ |
| 注意力编码器(Attention) | × | × | √ | √ | × | × |
| 历史编码器(LSTM) | × | √ | × | √ | × | √ |
| 全连接层(FC) | √ | √ | √ | √ | √ | √ |

在 Empty-16×16-v1 和 FourRooms-v1 环境下对六种算法进行了训练效果的测试,采用多进程的训练方式加快深度强化学习算法的收敛。在每个进程中生成随机种子不同的训练环境,智能体每与环境交互 128 次后将数据信息存入经验池,然后随机从经验池中抽取 batch-size 大小的数据信息进行参数更新,采用各进程的平均策略损失值与平均价值损失值作为目标函数的 loss 值项进行反向传播与参数更新,最终平均奖励值体现总体的训练效果。六种算法在 Empty-16×16-v1 环境下的训练奖励值变化如图 6(a)所示,横坐标的 frames 表示智能体与环境交互的总步数。由于环境中随机波动因素较小,门的位置仅在围墙中间部分波动,智能体在五种算法情况下都能成功找到最终目标。其中,ARPPO 能够以较快的速度达到最高奖励值并完全收敛,得益于该算法采用了注意力机制,获取到了更多的重要关键信息,忽略了一些无关紧要的编码信息,并且 LSTM 网络对历史信息编码,能够对更多的信息进行充分利用,做出更佳的判断与决策。值得注意的是,面对存在小部分随机因素的环境,仅融合循环神经网络或注意力机制模块的 PPO 算法不能很好地对随机变化的因素进行判断与决策。而且 A2C 算法对于探索该类非固定场景具有良好的表现,这是由于 A2C 算法不存在重要性采样,策略更新变化幅度大,对于动态变化因素适应力比 PPO 算法更强。然而参考历史数据信息进行训练的 LSTM-A2C 算法表现效果并不理想,某一地图场景训练所得的策略参数很难适用于其他不同场景,训练效果甚至比不上仅用卷积网络处理特征信息的 A2C 算法。




(a)Empty-16×16-v1




(b)FourRooms-v1

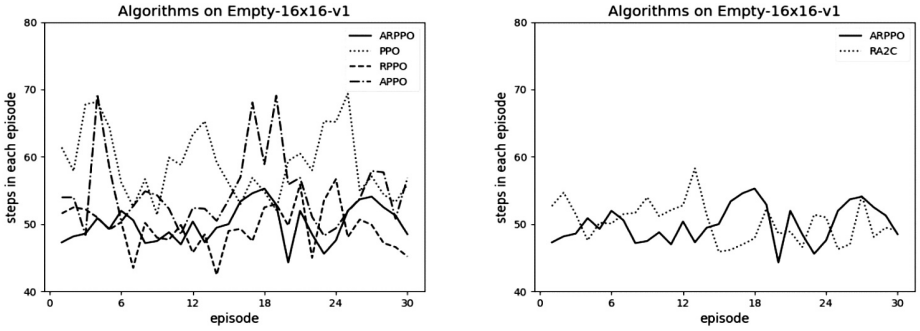
图 6 不同算法的性能对比

具有更多动态变化因素的 FourRooms-v1 环境,每一回合地图中间的四面墙会随机产生一个位置缺口,智能体要学会在每一回合中找到墙壁缺口并最终找到右下角的目标点。六种算法在该环境的训练表现如图 6(b)所示。表 2 体现了表现各算法在两张地图上的详细收敛情况。数据表明 ARPPO 算法综合收敛速度比表现较好的 APPO 算法与 RPPO 算法分别提高了 37.96% 与 37.65%,且从图 5 与图 6 的每回合的步数使用情况来看,ARPPO 算法在收敛之后的稳定性也不错。综上表明,ARPPO 算法明显比 RPPO 算法收敛更快且收敛之后比 APPO 算法更具有稳定性,这是由于 LSTM 网络为样本数据建立时序依赖关系,而引入注意力机制则强化了长距离中重要且关键的样本数据之间的依赖关系,解决了随着时间跨度增加,前阶段所采集的样本数据对后续的策略选择与价值估计的影响呈指数衰减这一现象。

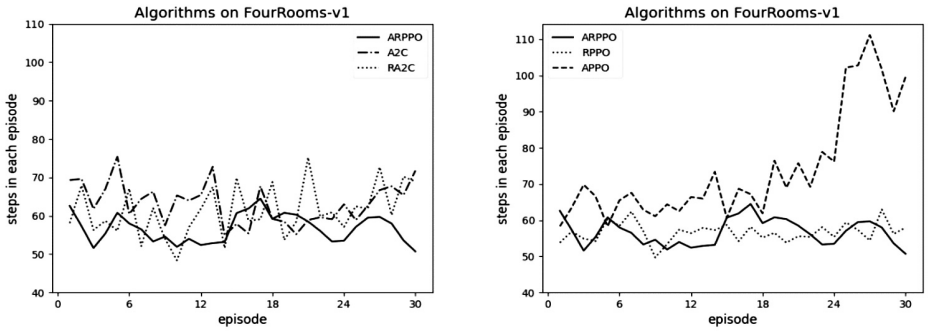
表 2 各算法收敛所用的环境交互次数($\times 10^6$)

| 算法 | Empty-16×16-v1 | FourRooms-v1 |
|---|---|---|
| PPO | 3.27 | — |
| APPO | 1.90 | 3.03 |
| RPPO | 2.91 | 2.15 |
| ARPPO | 1.26 | 1.75 |
| A2C | 3.01 | 2.64 |
| RA2C | — | 3.91 |

为进一步验证算法收敛后的稳定性,选取了最后 30 个 episode 的训练情况作为参考对象,从具体步数来探究算法收敛后的稳定性。各类算法在 Empty-16×16-v1 与 FourRooms-v1 的训练情况如图 7 所示。




(a)Empty-16×16-v1




(b)FourRooms-v1

图 7　最后 30 个回合的算法步数变化情况。

由图 7 中发现,PPO 算法与 APPO 算法收敛后步数变化幅度较大,对于动态随机因素的适应性稍弱,环境发生改变时,并不能选取最优的探索路径。RPPO 算法与 ARPPO 算法收敛后的稳定性很强,对于动态改变的环境仍具有较好的适应能力。

由于环境动态改变的随机性,各回合离目标点的距离不确定,故仅平均步数并不能客观地体现出各算法的稳定性,所以还选取了 30 个 episode 的步数标准差作为评估对象。表 3 中数据前项为平均步数,后项为标准差。综合数据体现出 ARPPO 算法与 RPPO 算法的稳定性最优,每回合都能采取更优的探索路径完成探索任务。但在两种算法稳定性相当的情况下,ARPPO 算法的收敛速度比 RPPO 算法的更快。

表 3　各算法最后 30 个 episode 的平均步数与标准差

| 算法 | Empty-16×16-v1 | FourRooms-v1 |
|---|---|---|
| PPO | 58.98±5.02 | — |
| APPO | 54.99±5.50 | 73.93±14.84 |
| RPPO | 49.43±3.26 | 56.65±2.67 |
| ARPPO | 50.03±2.76 | 56.73±3.66 |
| A2C | 50.32±2.91 | 63.39±5.33 |
| RA2C | — | 61.22±6.56 |

## 4　结束语

针对部分可观测环境因缺乏全局信息导致探索困难这一问题,提出了一种基于注意力机制和循环神经网络的深度强化学习算法,即 ARPPO 算法。该算法引入注意力机制和 LSTM 网络虽然在计算量和复杂度上有一定的增加,但网络模型结构设计简单,仅设计了一层多注意力模型提高智能体的信息提取能力,相比复杂的注意力模型而言,计算量与复杂度增加相对较小,并且结合注意力与 LSTM 网络增强了智能体的长时记忆能力,使其能够在动态随机性强的环境保持长时记忆,在环境中获取重要且关键的信息,从而能够快速地学习到有效的探索策略,使得算法达到收敛效果,最终完成探索任务。基于 Minigrid 设计了两项部分可观测环境的探索任务验证 ARPPO 算法的效果,实验结果表明 ARPPO 算法在收敛速度方面优于已有的 RPPO,A2C 等算法,同时兼顾了稳定性,具有较强的泛化能力。该文为解决部分可观测环境的探索问题提供了一种有效的方法,也为未来的研究提出了一些可能的方向,比如在更为复杂和具有更多动态变化因素的环境中测试 ARPPO 算法,并尝试使用多层注意力模块或 Bi-LSTM 网络来进一步提升其性能。

**参考文献:**

[1]　任　航,贺筱媛,陶九阳. 联合战役兵棋 AI 体系框架设计及关键技术分析[J]. 火力与指挥控制,2023,48(1):121-129.

[2]　程　恺,陈　刚,余晓晗,等. 知识牵引与数据驱动的兵棋

AI 设计及关键技术[J]. 系统工程与电子技术,2021,43(10):2911-2917.
[3] 高振海,闫相同,高 菲. 基于逆向强化学习的纵向自动驾驶决策方法[J]. 汽车工程,2022,44(7):969-975.
[4] 田 康,于 镝,李 擎,等. 基于改进 TD3 的自动驾驶车道保持决策方法[J]. 北京交通大学学报,2022,46(5):84-94.
[5] 马新新,管昕洁,白光伟,等. 边缘计算场景下基于强化学习的应用最优部署[J]. 计算机工程与设计,2021,42(1):15-23.
[6] 张 翔,吴 华,陈 游,等. 基于 POMDP 的主动雷达制导弹干扰措施优化方法[J]. 空军工程大学学报:自然科学版,2018,19(5):90-96.
[7] SATHEESHBABU S,UPPALAPATI N K,CHOWDHARY G,et al. Open loop position of soft continuum arm using deep reinforcement learning[C]//2019 international conference on robotics and automation (ICRA). Montreal:IEEE,2019:5133-5139.
[8] 何富君,王晓争,刘 凯. 基于 LSTM 与非对称网络的改进 DDPG 算法研究[J]. 计算机应用研究,2022,39(1):183-187.
[9] 李少波,刘意杨. 基于改进深度强化学习的动态移动机器人协同计算卸载[J]. 计算机应用研究,2022,39(7):2087-2090.
[10] FENG C,ZHANG Y W,HUANG C,et al. Deep reinforcement learning method for gait control of bipedal robots[J]. Computer Integrated Manufacturing Systems,2021(8):2341-2349.
[11] WONG C C,CHIEN S Y,FENG H M,et al. Motion planning for dual-arm robot based on soft actor-critic[J]. IEEE Access,2021,9:26871-26885.
[12] YE D,LIU Z,SUN M,et al. Mastering complex control in MOBA games with deep reinforcement learning[C]//Proceeding of the AAAI conference on artificial intelligence. [s. l.]:AAAI,2020:6672-6679.
[13] VINYALS O,BABUSCHKIN I,CZARNECKI W M,et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning[J]. Nature,2019,575(7782):350-354.
[14] 王学宁,贺汉根,徐 昕. 求解部分可观测马氏决策过程的强化学习算法[J]. 控制与决策,2004,19(11):1263-1266.
[15] DOSHI-VELEZ F,PFAU D,WOOD F,et al. Bayesian nonparametric methods for partial observable reinforcement learning[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2013,37(2):394-407.
[16] EGOROV M. Deep reinforcement learning with POMDPs[D]. Stanford:Stanford University,2015.
[17] MENG L,GORBET R,KULIĆ D. Memory-based deep reinforcement learning for POMDPs[C]//IEEE/RSJ international conference on intelligent robots and systems. Piscataway:IEEE,2021:5619-5626.
[18] HAUSKNECHT M,STONE P. Deep recurrent Q-learning for partially observable MDPs[C]//2015 association for the advancement of artificial intelligence fall symposium series. Palo Alto:AAAI,2015:1-8.
[19] 刘剑锋,普杰信,孙力帆. 融合对比预测编码的深度双 Q 网络[J]. 计算机工程与应用,2023,59(6):162-170.
[20] 耿俊香,姜 静,魏胜楠,等. CIDDPG 的多智能体通信优化方法研究[J]. 沈阳理工大学学报,2021,40(4):29-34.
[21] 刘国名,李彩虹,李永迪,等. 基于改进 PPO 算法的机器人局部路径规划[J]. 计算机工程,2023,49(2):119-126.
[22] 申翔翔. 深度强化学习在实时策略游戏中的应用研究[D]. 北京:北京交通大学,2019.
[23] CHEVALIER-BOISVERT M,WILLEMS L,PAL S. Minimalistic grid world environment for OpenAI gym[EB/OL]. 2018. https://github.com/maximecb/gym-minigrid.