软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(4):948−966 [doi: 10.13328/j.cnki.jos.005930]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于自回归预测模型的深度注意力强化学习方法*

梁星星, 冯旸赫, 黄金才, 王 琦, 马 扬, 刘 忠

(国防科技大学 系统工程学院,湖南 长沙 410072)

通讯作者: 冯旸赫, E-mail: fengyanghe@yeah.net

摘 要: 近年来,深度强化学习在各种决策、规划问题中展示了强大的智能性和良好的普适性,出现了诸如AlphaGo、OpenAI Five、Alpha Star 等成功案例.然而,传统深度强化学习对计算资源的重度依赖及低效的数据利用率严重限制了其在复杂现实任务中的应用.传统的基于模型的强化学习算法通过学习环境的潜在动态性,可充分利用样本信息,有效提升数据利用率,加快模型训练速度,但如何快速建立准确的环境模型是基于模型的强化学习面临的难题.结合基于模型和无模型两类强化学习的优势,提出了一种基于时序自回归预测模型的深度注意力强化学习方法.利用自编码模型压缩表示潜在状态空间,结合自回归模型建立环境预测模型,基于注意力机制结合预测模型估计每个决策状态的值函数,通过端到端的方式统一训练各算法模块,实现高效的训练.通过 CartPole-V0 等经典控制任务的实验结果表明,该模型能够高效地建立环境预测模型,并有效结合基于模型和无模型两类强化学习方法,实现样本的高效利用.最后,针对导弹突防智能规划问题进行了算法实证研究,应用结果表明,采用所提出的学习模型可在特定场景取得优于传统突防规划的效果.

关键词: 注意力机制;深度强化学习;actor-critic 算法;变分自动编码;混合密度网络-循环神经网络

中图法分类号: TP311

中文引用格式: 梁星星,冯旸赫,黄金才,王琦,马扬,刘忠.基于自回归预测模型的深度注意力强化学习方法.软件学报,2020,31(4):948−966. http://www.jos.org.cn/1000-9825/5930.htm

英文引用格式: Liang XX, Feng YH, Huang JC, Wang Q, Ma Y, Liu Z. Novel deep reinforcement learning algorithm based on attention-based value function and autoregressive environment model. Ruan Jian Xue Bao/Journal of Software, 2020,31(4): 948−966 (in Chinese). http://www.jos.org.cn/1000-9825/5930.htm

Novel Deep Reinforcement Learning Algorithm Based on Attention-based Value Function and Autoregressive Environment Model

LIANG Xing-Xing, FENG Yang-He, HUANG Jin-Cai, WANG Qi, MA Yang, LIU Zhong

(College of Systems Engineering, National University of Defense Technology, Changsha 410072, China)

Abstract: Recently, deep reinforcement learning (DRL) is believed to be promising in continuous decision-making and intelligent scheduling problems, and some examples such as AlphaGo, OpenAI Five, and Alpha Star have demonstrated the great generalization capability of the paradigm. However, the inefficient utility of collected experience dataset in DRL restricts the universal extension to more practical scenarios and complicated tasks. As the auxiliary, the model-based reinforcement learning can well capture the dynamics of environment and bring the reduction in experience sampling. This study aggregates the model-based and model-free reinforcement learning algorithms to formulate an end-to-end framework, where the autoregressive environment model is constructed, and attention layer is incorporated to forecast state value function. Experiments on classical CartPole-V0 and so on witness the effectiveness of proposed framework in simulating environment and advancing utility of dataset. Finally, penetration mission as the practical instantiation

* 基金项目: 国家自然科学基金(71701205)
Foundation item: National Natural Science Foundation of China (71701205)
本文由“非经典条件下的机器学习方法”专题特约编辑高新波教授、黎铭教授、李天瑞教授推荐.
收稿时间: 2019-05-31; 修改时间: 2019-07-29; 采用时间: 2019-09-20; jos 在线出版时间: 2020-01-10
CNKI 网络优先出版: 2020-01-14 10:49:02, http://kns.cnki.net/kcms/detail/11.2560.TP.20200114.1048.014.html

is successfully completed with the framework.

Key words: attention mechanism; deep reinforcement learning; actor-critic algorithm; variational auto-encoder (VAE); mixture density network-recurrent neural network (MDN-RNN)

深度强化学习(deep reinforcement learning,简称 DRL)在战略博弈[1,2]、无人机控制[3]、自主驾驶[4]和机器人合作[5]等领域取得了不错的成绩,是复杂调度与控制任务中颇具前景的一种学习范式,为通用人工智能的实现铺平了道路.DRL 应对环境和决策过程中不确定性的有效性,激发了将其应用于更多领域的研究热潮[6].Agent 与环境的持续交互能力是 DRL 取得成功的主要因素,这些交互能力缓解了环境的不确定性,揭示了环境中的动态性,使得 Agent 能够在延迟的奖励中学习驱动其获得良好表现的动作.

根据有无可用的环境模型,DRL 被分为两类:无模型 DRL(model-free DRL)和基于模型的 DRL(model-based DRL).

无模型的 DRL 仅仅利用环境奖赏,忽略了固有的、能够提高学习效率的潜在环境信息.虽然无模型 DRL 在实践中得到了广泛的应用,但其需要不断收集数以百万计的实例或经验来进行策略评估和改进,低效的数据利用率限制了其在复杂现实生活问题中的应用.

相反地,基于模型的 DRL 不严格依赖于环境交互,能够根据少量交互信息学习表征环境的潜在动力学,揭示任务的规律.利用获得的虚拟环境模型,Agent 不再通过与环境的直接交互来创造额外的经验,能够直接根据模型推导出最优策略.基于模型的算法在某些情况下往往保持更高的效率[7−9],但其修改能力弱,对环境精确建模的依赖性强,且在噪声环境中的适应性和灵活性较差.

在过去的几十年中,一些研究已经尝试将这两类方法结合起来,包括合成经验生成[10,11]和部分基于模型的反向传播[7,12,13],但在两种方法之间建立桥接的方式仍然有限.直觉表明,人类不需要太多的经验即可学习和适应复杂环境.虽然人类利用感官感知环境信息的能力有限,但可以概括复杂环境的知识,即人类能够从有限的感官信息中概念化事物,进而概括决策.神经网络模型的著作[14,15]认为,人类倾向于建立认知有限的世界模型,并依托该模型进行决策.人类大脑频繁地在头脑中使用先前自构建的物理模型[16,17],通过预测在某个状态下即时行动后的未来情景,迅速做出反应与决策并避免潜在的危险[18,19].Ha&amp 和 Schmidhuber[15]将上述观点付诸实践,建立了世界模型(world models,简称 WM),证明了 WM 能够以有限的现实经验建立,并显著改善策略学习的效率.这一框架通过学习虚拟环境模型,减少了在环境中收集转移以及相关消耗的繁重工作.在现有的一些军事平台仿真实验中,仿真往往需要耗费大量的时间进行模拟,进而为系统学习提供数据.此类系统的泛化能力与获取环境数据的人工和财务费用呈正相关.幸运的是,与虚拟环境的交互可以缓解这些消耗性的预测或控制.

本文对上述工作[15]进行了扩展,研究了无模型 DRL 算法和基于模型的控制相结合的方法,该方法探索了丰富的环境转移信息,指导了最优策略的搜索.本文利用神经网络进行了环境状态嵌入表示、自回归预测,并通过基于注意力机制的策略学习来改进 WM.

本文第 1 节总结一些相关的研究成果,并指出本文的研究意向.第 2 节对 WM 进行回顾,详细介绍本文提出的模型框架 VMAV-C,包括模型中的成分、训练过程和技术细节.第 3 节利用所提算法对经典的控制问题以及导弹智能突防任务进行研究与分析.最后得出结论,并介绍未来的研究工作.

1 相关工作

OpenAI Gym[20]提供了一系列虚拟环境开发和测试新的强化学习算法的任务与环境,对算法性能进行比较和验证.这些任务包括一些传统的控制问题,其中端到端(end-to-end)的任务更实际,且更具挑战性.端到端的任务要求 Agent 直接接收场景图像等作为原始输入,进而做出决策,包括 CartPole、MountainCar 等.图像固有的高维性给学习过程带来了很大的困难,激发了表征学习在强化学习中的应用[21].深度神经网络可以提取高维输入的紧凑表示特征,将复杂的实例编码为低维向量,能够训练一个处理复杂任务的强化学习模型.此外,深度学习具有良好的泛化能力,DQN[1]和 AlphaGo Zero[2]都得益于卷积神经网络对状态的表示,在 RL 的策略学习中实现

了最优的效果.

虽然强大的表示模型和日益增强的计算能力能够满足 DRL 解决复杂控制问题的基本要求,但从实际环境中访问数据仍然是 DRL 的瓶颈,算法对数据有着巨大的需求.与环境的交互对强化学习的成功应用起着决定性作用,为了达到理想的效果,需要消耗大量的人力、时间和金钱等资源来从环境中获得转移和奖赏.特别地,对于无模型强化学习算法,情况更为明显,数据利用效率较低,忽略了环境中的结构信息.这种困境受到了越来越多的关注,并激发了一些有趣的想法以解决这一问题.

在本文的研究中,学习环境模型是非常重要的,主要有两种模式来捕捉环境的特性和消除建模中的偏差.一种是学习以某种概率分布反映环境动态性的样本,并探索策略.早期的同步学习环境模型和策略的工作并不稳定,期望最大化方法(EM)[22]将参数从控制模型中分离出来,只需学习有限的控制参数就可以加快收敛速度.作为学习环境模型的突破,WM[15]可以自动揭示环境的动态性,并提到了从认知科学中获得的动机.Piergiovanni 等人[23]构建了深度神经网络,将状态编码和预测未来场景作为环境模型的模拟,并证明机器人可以通过与这种梦境的交互作用,学会在现实世界中行动的合理策略.考虑到在基于视觉的强化学习中处理图像观察的高度复杂性和成本[24],Nair 等人提出了一种将变分自动编码器(VAE)与非策略性目标条件强化学习相结合的图像目标强化学习算法,训练了一个循环状态空间模型以解决不确定环境下的规划问题,构造了一个称为深度规划网络的 Agent 可以学习控制策略[25].此外,原始图像很少用于环境建模,如世界模型[15]和 PA[26],大多采用自动编码器来获得低维状态,进一步提高了训练效率,减少了控制参数的规模.另一个范例是元学习,它寻求从不同环境中学习到的多个动态模型,并整合这些模型的特征来描述环境中的不确定性[22,27,28].

2 VMAV-C 模型

VMAV-C 模型对应于变分自动编码器(**v**ariational auto-encoder,简称 VAE)、混合密度网络-递归神经网络(mixture density network-recurrent neural network,简称 MDN-RNN)、基于注意力的值函数(attention-based value function,简称 AVF)和控制器模型的组合.

与 WM[15]中用于优化控制器模型的协变矩阵自适应进化策略不同,本文在控制器中使用了基于 PPO 的 actor-critic(AC)算法[31]以解决离散动作空间的任务以及连续动作空间的任务.根据值函数的精确估计可以加速策略学习这一直觉,本文在 critic 网络中考虑了注意力机制来估计状态值函数.

为方便阅读与理解,本节首先对 VMAV-C 的结构进行介绍,并在附录中给出具体的训练步骤;之后对该框架中 WM 框架下的 VM-C[15]中的一些基本组件(包括 VAE、MDN-RNN 和 Controller)进行简要介绍;然后重点介绍基于注意力的值函数,并提出与 critic 网络相结合的方法.

2.1 VMAV-C RL模型框架

VMAV-C 模型包含 VAE、MDN-RNN、Attention Value Function 以及 Controller 模型,其中,VAE 模型参数较多,且主要目的是对输入的观测进行压缩编码,因而可脱离原 VMAV-C 模型进行单独训练;MDN-RNN 模型采用 RNN 结构,网络训练缓慢,需要的参数与样本较多,与 AV-C 模型结合训练,将拖慢整体学习速度,因而可采用部分样本对 MDN-RNN 进行预先训练;Attention Value Function 依靠 MDN-RNN 的部分隐藏层信息,因而可将其与 MDN-RNN 剥离开来,构建参数较少的神经网络结构加速学习速度;Controller 是单独的模型,参数较少,可单独进行训练.

VMAV-C 强化学习训练架构的示意图如图 1 所示,包含:样本采集(步骤 0)、VAE 训练(步骤 1)、MAV 训练(步骤 2)、MAV-C 训练(步骤 3)和执行(步骤 4).不同模块中组件之间的依赖性也在图 1 的虚线框中进行了总结.这些步骤是按照训练和测试模型的顺序进行的,模块的详细训练说明在附录 2 中给出了描述.

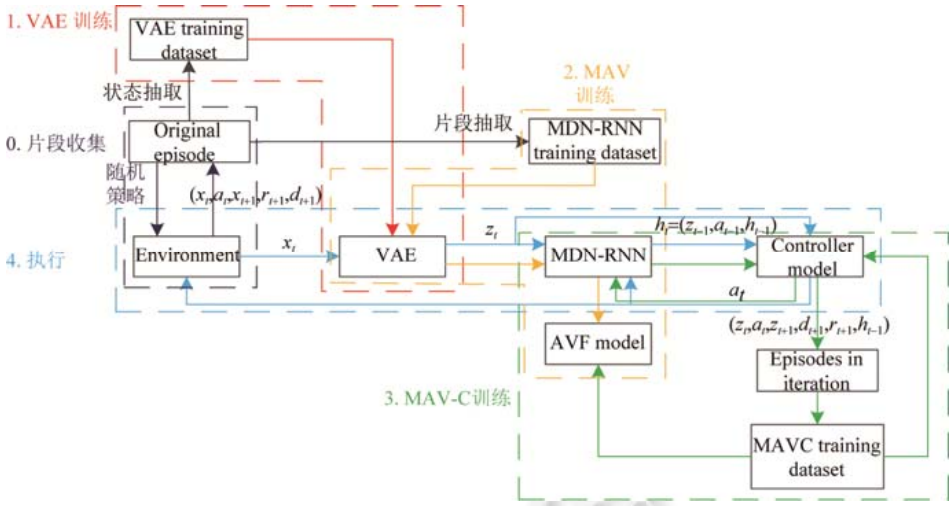


Fig.1 VMAV-C reinforcement learning training framework (Arrows suggest information flow in modules)

图 1 VMAV-C 强化学习训练架构(箭头示意为信息流)

2.2 VM-C模型框架

图 2 揭示了 VAE、MDN-RNN 和控制器模型之间的关系,并回答了 VMC 如何对环境做出动态反应的问题. 图 2 中的箭头线表示给定环境中的信息流和控制操作.

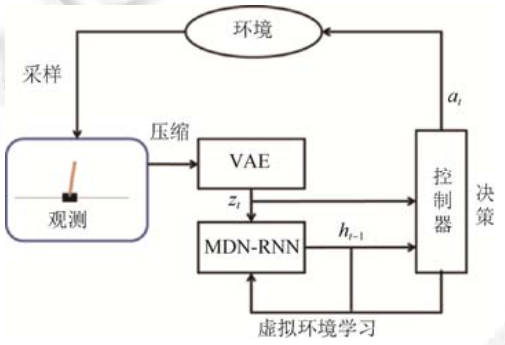


Fig.2 Framework of VM-C used in World Models

图 2 World Models 中的 VM-C 架构

2.2.1 VAE 模型

基于任一 d 维分布都可以由一个 d 维正态分布经过足够复杂的变换获得的思想,变分自编码器(variational auto-encoder,简称 VAE)[32]假定中间编码变量服从一个简单的高斯分布,例如 $\mathcal{N}(0,I)$ (I 为单位矩阵).

为了从模型中生成样本,VAE 从编码分布 $p_{model}(z)$ 中采样 z;然后通过可微的生成网络 $g(z)$;最后从分布 $p_{model}(x;g(z))=p_{model}(x|z)$ 中采样 x.在训练期间,编码器 $q(z|\boldsymbol{x})$ 用于获得 z,而 $p_{model}(\boldsymbol{x}|z)$ 则被视为解码器网络. VAE 通过最大化与数据点 $\boldsymbol{x}$ 相关联的变分下界 $\mathcal{L}(q)$ 对网络进行训练.

$$\mathcal{L}(q)=\mathbb{E}_{z\sim q(z|\boldsymbol{x})}\log p_{model}(\boldsymbol{x}|z)-D_{KL}(q(z|\boldsymbol{x})\|p_{model}(z)).$$

此外,还包含 MSEloss 损失函数 $\mathcal{L}(x,g(f(x)))$, 用于惩罚 $g(f(x))$与 x 的差异.

如图 3 所示,本文中的 VAE 输入是对环境的观测,即 CartPole-V0 等任务的场景图像,并将该观测压缩为一些低维向量作为状态的潜在表示;导弹智能突防实验中的输入则是红蓝双方的特征信息.图 3 中,编解码器是两个神经网络,均值向量和对数方差向量是某种状态的潜在表示.

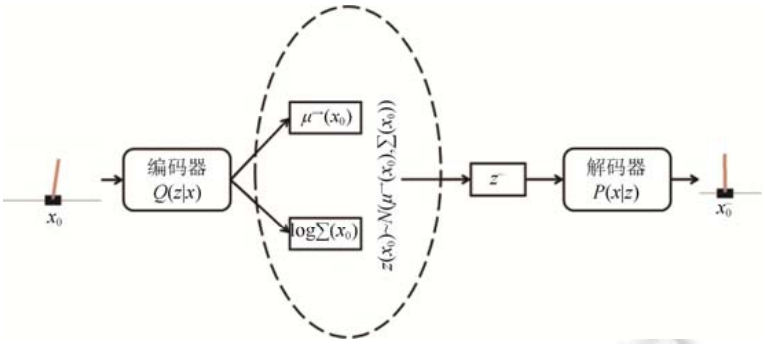


Fig.3 VAE in observation of CartPole-V0

图 3 VAE 实现 CartPole-V0 观测编码

2.2.2 MDN-RNN 模型

将混合密度模型与传统神经网络相结合,可以近似任意条件概率分布,特别是连续输入的概率分布,并在实际应用中解决了反演问题.同时,递归神经网络(RNN)在捕获序列数据集的依赖关系和感知序列趋势方面具有一定的效率.一些研究集中在这两种技术的结合上,并提出了 RNN 的一些变体,称为 MDN-RNN,用于处理现实生活中的问题[15,33],最近一项有趣的研究是将 MDN-RNN 应用于图纸中的草图生成[33].

在经典控制任务中,VAE 模型压缩编码了实验过程中的每帧图片,这些压缩编码在时间序列中,同样存在着原始图片信息内含的相关转移关系.MDN-RNN 模型的主要目的是预测当前状态下,agent 采取相关动作后,环境在下一时刻可能发生的状态,进而输出状态的压缩编码.在需要可视化的要求下,可以利用 VAE 的解码器对编码进行解码,获取图片信息,相应的示意图如图 3 所示.在复杂的真实环境中,环境的转移往往是不确定,因而,需要使用概率密度函数 $p(z)$替代确定预测 z 对未来进行估计.

在 RL 任务中,环境模型中的 RNN 通常被用来获得 $P(z_{t+1},r_{t+1}\mid a_t,z_t,h_t)$ 的状态转移函数,其中,a_t、z_t、h_t 分别表示当前时刻下的动作、状态表示、隐藏状态,z_{t+1} 表示对下一时刻状态表示的预测,r_{t+1} 表示下一时刻的奖赏.与传统的 seq2seq 任务不同,有结束状态 d_{t+1} 的 RL 环境还需要预测状态的结束标志,即标记该状态是否为结束标记,因而面向带结束状态的 RL 环境的 RNN 通常需要被建模为 $P(z_{t+1},d_{t+1},r_{t+1}\mid a_t,z_t,h_t)$, d_{t+1} 表示当前的状态 z_{t+1} 是否表示结束状态,r_{t+1} 表示获得的奖赏(有些任务中,奖赏是固定的,下文将省略表示奖赏).

在离散动作空间的任务中,本文对离散的动作进行编码后与环境编码进行结合并加入到预测模型中,即 $P(z_{t+1},d_{t+1}\mid f(a_t),z_t,h_t)$. MDN-RNN 的损失函数由下一状态的预测损失 L_s 以及结束标记的预测损失 L_p 所组成.

$$L_s=-\frac{1}{N}\sum_{i=1}^{N}\log\left(\sum_{j=1}^{M}\theta_{j,i}\mathcal{N}(x_i,y_i\mid\mu_{x,j,i},\mu_{y,j,i},\sigma_{x,j,i},\sigma_{y,j,i},\rho_{xy,j,i})\right),$$

$$L_p=-\frac{1}{N}\sum_{i=1}^{N}(\alpha d_{(t+1)i}\log q_i+(1-d_{(t+1)i})\log(1-q_i)),$$

其中, $\theta_k=\frac{\exp(\hat{\theta}_k)}{\sum_{j=1}^{M}\exp(\hat{\theta}_j)},k\in\{1,...,M\}$. 为了控制高斯分布采样的随机性,使用温度参数 τ 对权重、方差进行伸缩,τ 的取值通常在 0 和 1 之间(视具体任务而定,可松弛至>1): $\hat{\theta}_k\to\frac{\hat{\theta}_k}{\tau},\sigma^2\to\sigma^2\tau$.

损失函数 $Loss$ 是 L_s 和 L_p 的加权和:

$$Loss=\beta_1L_s+\beta_2L_p,$$

其中,$\{\beta_1,\beta_2\}$是损失项的权重.图 4 详细描述了 MDN-RNN 模型的结构,指出了动作、状态的潜在表示、RNN 序列中的隐藏信息和结束状态之间的依赖关系.图 4 中,每个 LSTM 网络包含 3 个 LSTM 单元.在结束标记的预测损失 L_p 中,考虑到一个时间样本中结束标记分布较小,为了提高预测的准确性,本文利用超参数调整结束位上的惩罚权重.

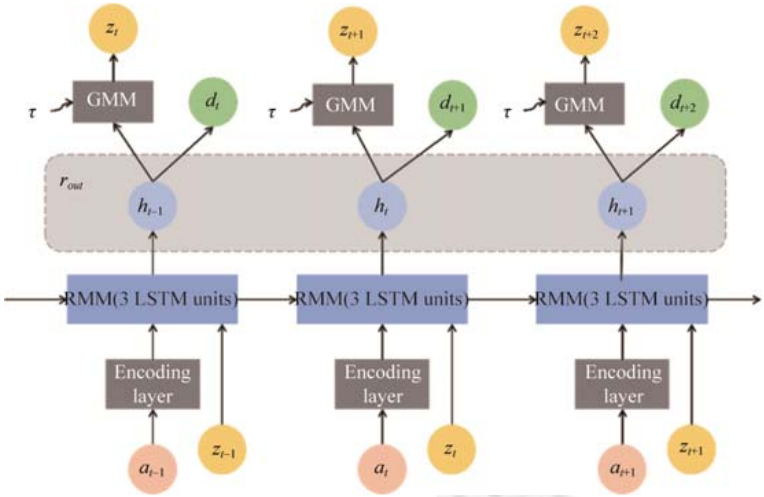


Fig.4 MDN-RNN

图 4 MDN-RNN 结构

2.2.3 控制器模型

控制器模型用来决策当前时刻以及状态下所采取的动作.在决策中,利用 MDN-RNN 的上一时刻隐藏状态信息以及当前时刻的状态信息共同预测当前时刻应采取的动作,即 $a_t \sim \pi(\boldsymbol{a} \mid z_t, h_t)$. 在真实环境中,环境编码的信息 z_t 来自于真实环境的观测压缩编码;在虚拟环境中,编码信息 z_t 来自于预测的采样信息 z_t.控制器模型结构图如图 5 所示.其中,h_t 来源于 MDN-RNN 中的隐藏信息,z_t 是当前状态的潜在表示,动作受这两种信息的制约.这里使用的是全连接(FC)网络.

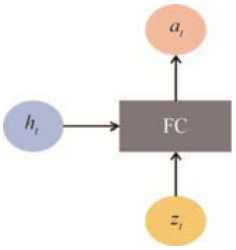


Fig.5 Controller model

图 5 控制器模型结构图

2.3 AVF模型

近年来,注意力机制越来越受到人们的关注,尤其是在序列学习领域.注意力机制是对历史序列中的隐藏信息赋予各种权重,然后对其进行聚合,形成预测时间步的上下文向量.与预测时间步相关的隐藏信息将受到更多的关注,并被赋予更多的权重,即给定某个 t 步序列的隐藏信息 $H=[h_1,\ldots,h_t]$,预测时间步的上下文向量 $v=\sum_{i=1}^{t}\alpha_i h_i$ 作为历史序列的嵌入信息.

在强化学习算法的训练过程中,本文将注意力机制引入状态值函数的估计中,历史隐藏信息将有助于准确地估计当前状态值.

在 AC 算法的 critic 网络中,图 6 所示的每一次隐藏信息都来自 MDN-RNN 的 r_{out},并利用前 n 个时间步的历史信息进行当前状态值估计.为了确保初始状态也能满足注意力结构,缺少的隐藏信息,如 $[h_{-2},h_{-1},h_0]$ 被初始化为 0(这里以图 6 中的情况为例).图 6 中,包含 MDN-RNN 隐藏信息的 4 个最近单元有助于状态值估计,注意层是计算这些信息的重要因素.

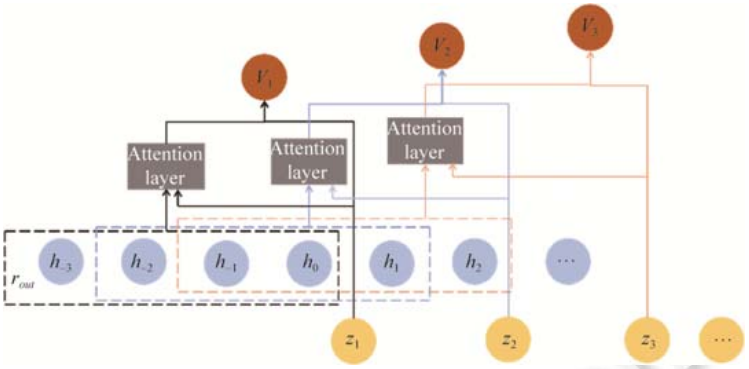


Fig.6 Attention-based value function representation

图 6 基于注意力的值函数(AVF)表示

因此,有注意力的上下文向量可以计算为

$$c_t=\sum_{i=1}^{n}\alpha_i h_{t-i},$$

$$\alpha_i=\frac{\exp(\tilde{\alpha}_i)}{\sum_{j=1}^{n}\exp(\tilde{\alpha}_j)},$$

$$\tilde{\alpha}_i=W[h_{t-i},z_t]+b,$$

其中,W、b 是注意力网络的参数,α_i 反映了以 i 为索引的历史信息在上下文向量 c_t、z_t 中的影响强度,z_t 作为预测时间步的输入.对于状态值函数估计,需要将从隐藏信息 $\{h_{t-1},h_{t-2},...,h_{t-n}\}$ 导出的上下文信息 c_t 和当前状态信息合并计算:$V(s_t)=W_v[z_t,c_t]+b_v$,其中,z_t 是时间步 t 中状态的潜在表示,c_t 是具有注意力机制的上下文向量,[.,.]是向量的串联,$\{W,b,W_v,b_v\}$是基于注意力机制的值神经网络中要学习的参数集.图 6 揭示了状态值的学习过程,这种结构也是本文实验中的具体设置.

3 实验和性能分析

上述小节描述了 VMAV-C 的实现过程,本节将在具体环境中对 VMAV-C 的算法性能进行测试与评估.本文选取了 CartPole-V0、MountainCar-V0 以及 Acrobot-V1 等 OpenAI Gym 中的经典控制任务和导弹智能突防等环境进行测试,分别考察了在真实环境和虚拟环境中算法的表现.采用 CartPole 等实验的原因是由于该任务较为简单,但其拥有复杂任务应有高维度、端到端、连续性交互等特点,能够对算法进行快速验证.针对当前强化学习方法在具体问题中难以落地的困境,我们结合自身项目需求,对 DRL 在军事问题中的应用进行了研究,利用本文的算法对我们自主开发的导弹智能突防环境中的任务进行了求解,验证 DRL 在实际系统中的可行性.

3.1 实验环境介绍

CartPole:在这一任务下,一辆推车与一根杆子连接在一起形成一个倒立摆,控制决策包括对倒立摆实施向左或向右的力等两个离散的动作.倒立摆初始化为直立式,延长直立时间是该任务的目标.当倒立摆从中心 2.4 单位的范围内移出,或者摆与垂直方向的夹角超过 15°时,任务结束.在杆保持直立的情况下,每一步返回+1 的奖赏信号.OpenAI GYM 中的 CartPole 包含 CartPole-V0 和 CartPole-V1 这两个任务,CartPole-V0 的最大步长为 200 步,CartPole-V1 为 500 步.本文所使用的任务为 CartPole-V0.

MountainCar-V0:在该任务中,汽车位于一条轨道上,位于两个“山脉”之间.目标是在右边开出;然而,汽车的发动机强度不足以在一次通过中攀登山峰.因此,成功的唯一途径是来回驾驶以增强动力.动作包含 3 个离散动作:向左、不动和向右.原本实验中的奖赏为单步−1,离开后奖赏为 0.本文为了加快收敛,将奖赏函数设置为距离底部越远,奖赏值越高,即 $r=abs(position+0.5)\times 0.1$,离开后奖赏值为 10,离开的标志为:if position-goal_

position>−0,done=Ture.学习的目标是使得小车用尽可能短的步数离开.

Acrobot-V1:Acrobot 是一个双连杆摆,该机器人系统包括两个接头和两个连杆,其中,两个连杆之间的接头被致动,两个链接都可以自由摆动并且可以相互通过,即它们不会在具有相同的角度时会发生碰撞.最初,链接向下悬挂,目标是将下部链接的末端摆动到给定高度.动作是在两者之间的关节上施加+1、0 或−1 的扭矩.奖赏为固定奖赏−1.为了加快收敛,我们假定,若在 500 步以内没有到达指定高度,则结束,给予−10 的惩罚值,反之,则为+10.学习的目标是使得下部链接尽快到达指定高度.

导弹智能突防:导弹智能突防场景是一款面向军事应用的仿真环境,该环境依靠概念级模型以及部分半实物仿真模型构建而成,用于仿真当前红蓝双方的导弹对抗过程.该环境采用模拟时钟实时推进系统演化,可以通过控制仿真时钟速度而加快仿真速度.虽然该环境可以获得当前对抗的图片信息,但考虑到真实对抗过程中,决策的数据是经过多传感器综合处理后的目标特征.红方的导弹特征包含经度、纬度、高度、速度、偏向角、燃料、所处阶段等,蓝方拦截导弹的特征包含经度、纬度、高度、速度等.环境定义奖赏函数为燃料的消耗量以及预计落点相对目标的变化量: $r_{t+1}=-\Delta fuel+\alpha\Delta dis$, 其中,$\alpha$=$le$−3.此外,在仿真结束时刻,如果击中目标,则额外赋予+1 的奖赏值,反之,则为−1.仿真结束包含 3 种情况:红方导弹被拦截;红方导弹突防成功并击中目标;红方导弹突防成功未击中目标.

3.2 经典控制任务实验

CartPole-V0、MountainCar-V0 以及 Acrobot-V1 等 OpenAI Gym 中的经典控制任务有着相似的任务特征,本文在附录 4 中对实验的设置进行了描述,在本节中对实验的相关算法以及实验结果进行对比分析.在附录 6 中,对 MDN-RNN 所学习的环境模型进行了展示.

3.2.1 对比算法

为了更好地描述 VMAV-C 模型的性能,本文提出了两种基线算法,与 VMAV-C 方法进行了对比.

(1) 经过编码表示的输入的 PPO 算法(contractive PPO,简称 CP).该方法将从真实环境中获得的图片放入 VAE 模型中,获得当前状态的编码表示,并将此编码表示作为 agent 的决策输入.

(2) 带 MDN_RNN 模型输入的 PPO 算法(MDN-RNN PPO algorithm,简称 MRP).该方法以 Ha 等人[15]所述的方法作为基础,但在 controller 模型中采用 PPO 算法进行策略学习.

此外,在这些实验中还包括了完全在虚拟环境中训练 agent 进行决策的场景.本文将从实际环境中随机抽取一些初始状态进行编码作为虚拟环境的初始状态,运行 MDN-RNN,自动生成给定动作的未来状态、反馈奖赏和改进策略.此外,在实际环境中运用该学习策略,对累计奖励进行无折扣测试.此操作与步骤 4 不同,因为实际环境在策略改进中不提供连续的奖励信号,而只是在每个固定的时间段测试从虚拟环境中学习的策略.因此,采用 PPO 算法,作为 agent 的控制器通过与虚拟环境的交互直接学习策略.

3.2.2 结果与分析

实验结果分析

在经典控制问题中,采用了 CP 模型、MRP 模型和 VMAV-C 模型,3 个模型均使用 PPO 算法在真实的环境中进行了训练,模型差异见表 1.

Table 1 Comparison model in classic control environment

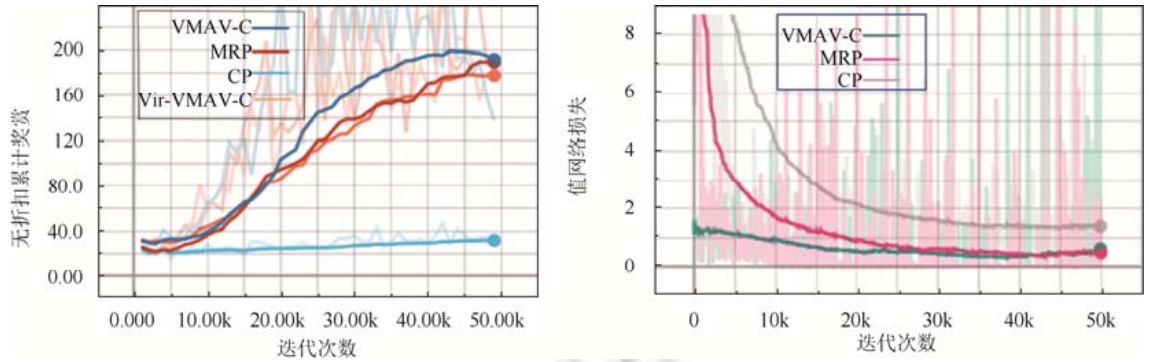
表 1 经典控制任务环境中的对比模型

模型	训练环境	输入	有无注意力机制
CP 模型	真实环境	最近的 4 帧图片	无
MRP 模型	真实环境	MDN-RNN 的隐层状态与当前时刻的观测编码	无
VMAV-C 模型	真实环境	MDN-RNN 的隐层状态与当前时刻的观测编码	有
Vir-VMAV-C 模型	虚拟环境	MDN-RNN 的隐层状态与生成的当前时刻的观测编码	有

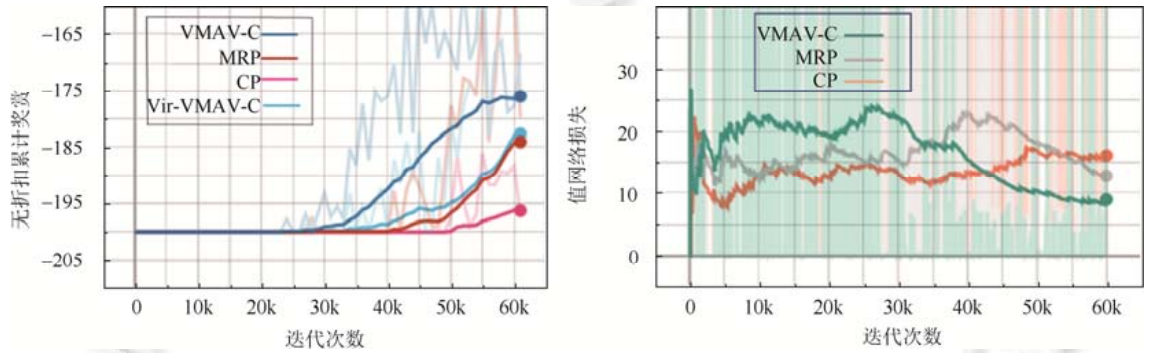
本文将分别分析上述 3 种模型无折扣累积奖赏和状态值网络的损失,结合表 1 与图 7 分析其背后的潜在原因.此外,还考察了 VMAV-C 在虚拟环境中的性能,在图 7 中,用 Vir-VMAV-C 进行表示,与上述 3 种模型一起进行

了比较.

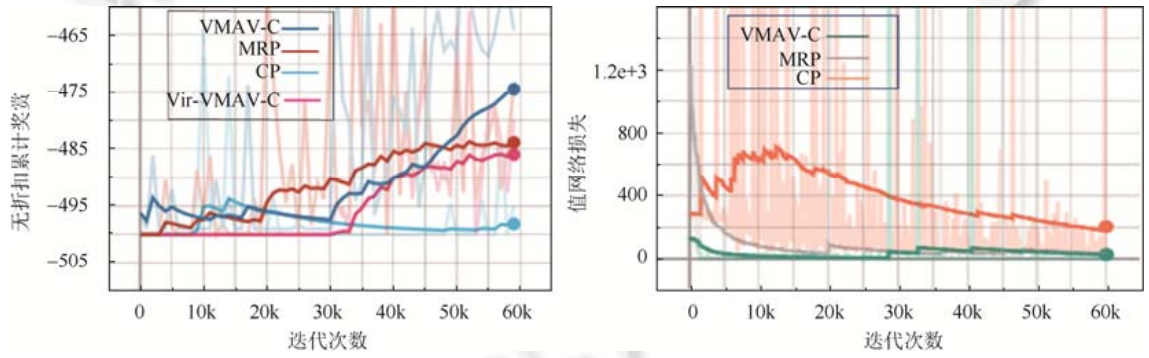
根据每个任务的特性,真实环境中训练的 3 个模型在训练中的迭代次数为 50 000 次及以上,每次迭代交互过程中最多有 32 个步骤的转换序列,剧烈波动的阴影曲线是结果中的真实累积值.为了更好地显示结果,使用 TensorboardX[34]将这些结果平滑为深色曲线.图 7 分别展示了 3 个任务下的实验效果.在图 7 所示的 Tensorboard 中,测试报酬和值网络损失的平滑率分别设置为 0.9 和 0.995.



(a) CartPole-V0



(b) MountainCar-V0



(c) Acrobot-V1

Fig.7 Cumulative rewards and value network losses under classic control environment

图 7 经典控制任务下的累计奖赏以及值函数损失表示

CP 模型

好的特征表示是深度强化学习成功的关键因素.在这一模型中,本文仅利用 VAE 模型对状态输入进行表示,采用最近的 4 个观测作为决策所需的状态,测试在这一条件下 agent 的学习效果.

在经典控制任务的 3 个实验中可以看到,与 MRP 模型相比,仅仅利用单纯的图片编码信息,随着值函数损失

的下降,agent 虽然也能够逐渐学习到较好的策略,但所需要的样本更多,学习速度慢.可以得出结论,MDN-RNN 模型不仅能够对过去的信息进行良好的总结,而且还包含这对未来信息的预测,获得的状态表示相比单纯的图片信息,更能够反映当前的实际状态.

MRP 模型

单纯的状态表示仅仅对当前的状态进行了较好的特征提取,但没有包含环境变化的趋势预测信息.MDN-RNN 模型从已有样本中对环境进行学习,利用当前的动作和过往的隐藏信息预测环境的未来状态,该隐藏信息不仅包含对过往信息的总结,同时也包含了对未来的预测.本文利用当前的状态以及对环境预测的信息得到的决策向量,作为 agent 的决策依据.

实验结果表明,结合编码信息和预测信息的 agent 能够快速对环境加以适应.相对于仅采用编码的训练方式,agent 所需训练样本更少,能够较快地达到较高的累积奖赏,而且值函数的损失相比 CP 模型下降得更快.但与 VMAV-C 模型相比,在 Critic 模型中没有包含注意力机制以及预训练,critic 函数的损失收敛较慢,使得算法性能落后于 VMAV-C 模型.该对比实验验证了 critic 函数对稳定 actor 函数具有重要的影响.

VMAV-C 模型

RNN 网络的隐藏信息虽然包含了对环境的预测信息,但忽略了信息的时间尺度,仅依赖上一时刻的预测信息作为输入,没有对历史信息加以足够的区分.带注意力机制的 world models 根据人类的思考方式,综合判断最近几次动作的行为,获得对当前状态值更好的估计.

带注意力机制的 world models 与原始的 world models 相比,具有更快的学习能力,达到同样的效果所需要的样本量更少.同时,对比两者的值函数损失值,结果表明,在注意力机制的影响下,agent 对状态值收敛更快,进而对策略具有更强的指导能力.

Vir-VMAV-C 模型

Vir-VMAV-C 模型不与真实环境进行交互,仅依靠 MDN-RNN 模型对环境进行向前推理.在具体实验设置中,本文从经典任务的初始状态中采样一帧图片,对其进行编码后传入 MDN-RNN 模型中进行虚拟交互.与 VMAV-C 模型的实验过程相同,每经过 1 000 次的训练后将所得 actor 模型在真实环境中进行 10 次测试后取其均值.

从图 7 可以看出,Vir-VMAV-C 模型的训练结果与 MRP 模型类似,但弱于在真实环境中进行训练的 VMAV-C 模型.考虑到经典任务环境相对简单,本文没有迭代式地对 MDN-RNN 模型进行训练.由于虚拟环境的自身特性,因而在该训练环境下无法达到真实环境中的训练效果.该实验验证了基于循环神经网络的 MDN-RNN 模型能够学习环境的动态性,且在该虚拟环境中进行一般性的强化学习训练能够达到与真实环境中训练相当的效果.训练虚拟环境所使用的交互样本少,而且达到同等效果所需时间仅为真实环境所需时间的 1/5(算法的软硬件环境描述见附录 5).

注意力权重分析

为了理解注意力机制在值函数估计中的作用,我们考察了 3 种任务下,在 1 次实验中 4 个潜在状态的注意力权重变化.如图 8 所示,在 CartPole-V0 任务中,我们发现,在实验初期,第 4 个潜在状态对值估计具有较高影响,4 个潜在状态按照距离当前时刻的远近,权重依次降低,而到了实验的后期,4 个状态对值估计趋于同等权重;在 MountainCar-V0 任务中,在运行之初,更偏重于第 4 个潜在状态,而越到后期越偏重于第 1 和第 4 状态,但更偏重于第 1 状态,我们认为这主要是由于该任务相邻的状态间难以分析山地车的相关速度信息,而较大的时间间隔能够获得相对速度信息,有助于做出更为准确的状态值估计;在 Acrobot-V1 任务中,在前期的摆动中,状态值估计更偏好于第 4 状态,在摆动的中期,则对 4 个状态给予同等的权重,到了后期,则更偏好于第 1 和第 4 状态,这样的分布变化是由该任务的特性所决定的.Acrobot-V1 的摆动过程也可以看作分 3 个阶段进行,首先是起始的加速度,将链接摆动到中间位置;其次,在中间位置时,需要调整下摆的位置,避免两杆的重合;在最后一个阶段,下摆要超过上杆并达到指定高度,因而需要对远近的状态进行比较.在图 8 所示的 TensorboardX 中,平滑率均设置为 0.97.

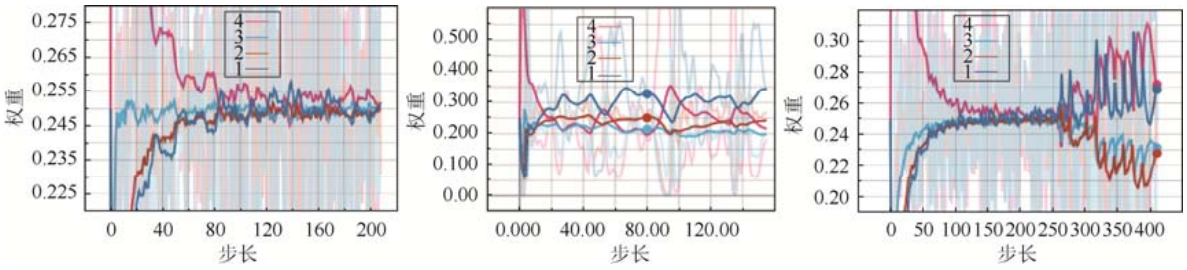


Fig.8 Attention weight change under classic control environment

图 8 经典任务下的注意力权重变化

3.3 导弹智能突防实验

CartPole 实验验证了虚拟环境中对 agent 训练的可行性,而且基于注意力机制的 critic 模型对 actor 模型的稳定具有重要影响.在智能突防的实验中,本文直接使用带注意力机制的 PPO 算法(critic 模型使用最近的 4 个 MDN-RNN 模型隐层状态)在多种数据获得的虚拟环境和真实环境中,分别对控制模型进行训练.在本文中,每经过 100 次训练,对控制器模型进行 15 次测试,获得当前模型的突防成功概率.对比算法的差异见表 2.

图 9 对不同训练条件下的获胜概率进行了表示,其中绿线表示利用内置的规则进行突防的获胜概率,灰线表示最大的获胜概率.VMAV-C 线表示的是利用 VMAV-C 方法和内置策略获得的 MDN-RNN 模型,在真实环境下进行训练的效果;Two iteration 线表示利用 VMAV-C 方法和预训练策略获得的 MDN-RNN 模型,在虚拟环境中进行训练的效果;Rule-based 线表示利用 VMAV-C 方法和内置策略获得的 MDN-RNN 模型,在虚拟环境中进行训练的效果;Random policy 线表示利用 VMAV-C 方法和随机策略获得的 MDN-RNN 模型,在虚拟环境中进行训练的效果.在图 9 所示的 Tensorboard 中,获胜概率的平滑率分别设置为 0.95.

Table 2 Comparison model in penetration mission

表 2 导弹智能突防实验的对比模型

模型	训练环境	MDN-RNN 数据来源
VMAV-C	真实环境	基于规则的内置策略产生的环境转移
Two iteration	虚拟环境	Random policy 训练结束后的预训练的 agent 产生的环境转移
Rule-based	虚拟环境	基于规则的内置策略产生的环境转移
Random policy	虚拟环境	基于随机策略产生的环境转移

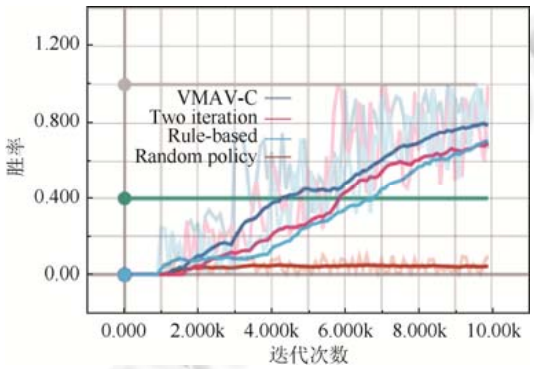


Fig.9 Win probability in the actual environment and virtual environment

图 9 智能突防场景胜率表示

从图 9 可以看到,VMAV-C 线训练效果最好,在经过 10 000 次的片段训练后,胜率可达 0.8;其次,Two iteration 线和 Rule-baesd 线效果相近,经过训练,胜率可达 0.7;而 Random policy 线训练效果最差,获胜概率始终在 0.1 以下.

经过分析我们发现,在利用随机策略采样得到的样本中,没有成功击中目标的片段,因而无法准确仿真出击中目标的实验片段,而利用该学习之后的策略再次从环境中采样,获得了一些成功的片段,所获得的样本能够训练出反映攻防对抗动态性的 MDN-RNN 模型,其达到的效果可以媲美基于内置策略获得的 MDN-RNN 模型.此外,尽管 VMAV-C 线获得的效果优于 Two iteration 线和 Rule-baesd 线,但其所花费的时间却远远大于后者.

随机因子的影响

为了分析 MDN-RNN 模型的随机性对虚拟环境的影响,本文在基于内置策略采样数据得到的 MDN-RNN 模型的基础上,考察了控制器模型在 3 个随机因子 τ={0.8,1.0,1.2}下的影响情况.

从图 10 可以看出,随着随机因子的提高,导弹的突防概率有所下降.当 τ=0.8 时,突防概率最高,突防概率可以达到 75%附近;当 τ=1.0 时,突防概率可以达到 70%左右;当 τ=1.2 时,突防概率则在 60%左右.为了分析这一原因,本文分别选取了各随机因子下的一段仿真片段进行解码还原.经过分析发现,随机因子越低,对导弹的对轨迹还原越精确;随机因子越高,导致导弹的飞行轨迹偏离实际运动模型,进而导致在虚拟环境中无法有效判别蓝方导弹是否有效拦截红方导弹,使得在真实环境中测试时效果较差.图 10 中,对于测试奖励,Tensorboard 中的平滑率分别设置为 0.9.

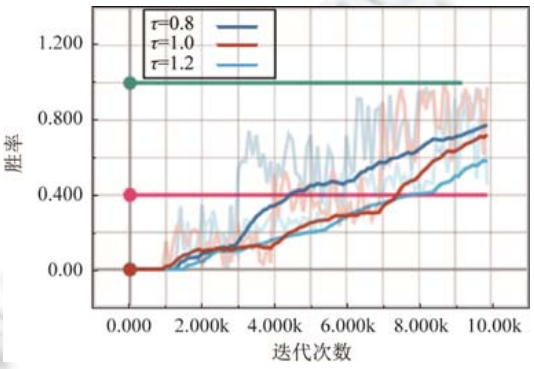


Fig.10 Win probability in various τ values of GMMs

图 10 不同随机因子对突防概率的影响

4 结 论

深度强化学习的进步为智能决策的发展提供了新的契机.无模型的强化学习通过与环境的交互可以获得容量相当的模型,然而其需要大量的样本,数据利用率低;基于模型的强化学习方法,通过规划能够快速得到决策模型,但其泛化能力较弱,对环境模型要求高.本文对世界模型的工作进行了修改,改进了策略学习过程,包括将注意力机制纳入状态值估计函数,利用基于 PPO 的 AC 算法优化离散动作空间的任务的策略学习,并利用高斯采样动作对导弹智能突防场景的突防策略进行了学习.实验结果证明了这些改进的有效性,加快了策略的学习速度,并进一步证明:结合 VAE 和 MDN-RNN 的有限经验可以建立对任务有益的虚拟环境模型,在虚拟环境中的训练大幅提高了 agent 的数据效率.在实际仿真系统中,VMAV-C 的性能优于以前的工作,大幅提高了数据的利用效率,且在虚拟环境中训练的 agent 也能够学习有效的策略.

在未来,我们将探索建立更多环境模型的方法,在更加复杂的任务中应用该模型.此外,也会更多地关注多 agent 系统,以提高仿真性能和效率.

References:

[1] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G. Human-level control through deep reinforcement learning. Nature, 2015,518(7540):529.

[2] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A. Mastering the game of Go without human knowledge. Nature, 2017,550(7676):354.

[3] Conde R, Llata JR, Torre-Ferrero C. Time-varying formation controllers for unmanned aerial vehicles using deep reinforcement learning. arXiv Preprint arXiv:1706.01384, 2017.

[4] Shalev-Shwartz S, Shammah S, Shashua A. Safe, multi-agent, reinforcement learning for autonomous driving. arXiv Preprint arXiv: 1610.03295, 2016.

[5] Su PH, Gasic M, Mrksic N, Rojas-Barahona L, Ultes S, Vandyke D, Wen TH, Young S. On-line active reward learning for policy optimisation in spoken dialogue systems. arXiv Preprint arXiv:1605.07669, 2016.

[6] Wang Q, Zhao X, Huang J, *et al*. Addressing complexities of machine learning in big data: Principles, trends and challenges from systematical perspectives. 2017. [doi: 10.20944/preprints201710.0076.v1]

[7] Pong V, Gu S, Dalal M, Levine S. Temporal difference models: Model-free deep rl for model-based control. arXiv Preprint arXiv: 1802.09081, 2018.

[8] Nagabandi A, Kahn G, Fearing RS, Levine S. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. 2017. https://arxiv.org/pdf/1708.02596.pdf

[9] Kamthe S, Deisenroth MP. Data-efficient reinforcement learning with probabilistic model predictive control. arXiv Preprint arXiv: 1706.06491, 2017.

[10] Sutton RS. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In: Proc. of the Machine Learning. Elsevier, 1990. 216–224.

[11] Kumar A, Biswas A, Sanyal S. eCommerceGAN: A generative adversarial network for e-commerce. arXiv Preprint arXiv:1801. 03244, 2018.

[12] Heess N, Wayne G, Silver D, Lillicrap T, Erez T, Tassa Y. Learning continuous control policies by stochastic value gradients. 2015. https://arxiv.org/pdf/1510.09142.pdf

[13] Chebotar Y, Hausman K, Zhang M, Sukhatme G, Schaal S, Levine S. Combining model-based and model-free updates for trajectory-centric reinforcement learning. arXiv Preprint arXiv:1703.03078, 2017.

[14] Forrester JW. Counterintuitive behavior of social systems. Technological Forecasting and Social Change, 1971,3:1–22.

[15] Ha D, Schmidhuber J. World Models. 2018. https://arxiv.org/pdf/1803.10122.pdf

[16] Chang L, Tsao DY. The code for facial identity in the primate brain. Cell, 2017,169(6):1013–1028.

[17] Nortmann N, Rekauzke S, Onat S, König P, Jancke D. Primary visual cortex represents the difference between past and present. Cerebral Cortex, 2013,25(6):1427–1440.

[18] Leinweber M, Ward DR, Sobczak JM, Attinger A, Keller GB. A sensorimotor circuit in mouse cortex for visual flow predictions. Neuron, 2017,95(6):1420–1432.

[19] Mobbs D, Hagan CC, Dalgleish T, Silston B, Prévost C. The ecology of human fear: Survival optimization and the nervous system. Frontiers in Neuroscience, 2015,9:55.

[20] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W. Openai GYM. arXiv Preprint arXiv:1606. 01540, 2016.

[21] Sun R, Silver D, Tesauro G, Huang GB. Introduction to the special issue on deep reinforcement learning: An editorial. 2018. [doi: 10.1016/j.neunet.2018.08.001]

[22] Kurutach T, Clavera I, Duan Y, Tamar A, Abbeel P. Model-ensemble trust-region policy optimization. arXiv Preprint arXiv:1802. 10592, 2018.

[23] Piergiovanni A, Wu A, Ryoo MS. Learning real-world robot policies by dreaming. arXiv Preprint arXiv:1805.07813, 2018.

[24] Nair AV, Pong V, Dalal M, Bahl S, Lin S, Levine S. Visual reinforcement learning with imagined goals. 2018. http://export.arxiv. org/abs/1807.04742

[25] Hafner D, Lillicrap T, Fischer I, Villegas R, Ha D, Lee H, Davidson J. Learning latent dynamics for planning from pixels. arXiv Preprint arXiv:1811.04551, 2018.

[26] Cuccu G, Togelius J, Cudre-Mauroux P. Playing atari with six neurons. arXiv Preprint arXiv:1806.01363, 2018.

[27] Clavera I, Rothfuss J, Schulman J, Fujita Y, Asfour T, Abbeel P. Model-based reinforcement learning via meta-policy optimization. arXiv Preprint arXiv:1809.05214, 2018.

[28] Rajeswaran A, Ghotra S, Ravindran B, Levine S. Epopt: Learning robust neural network policies using model ensembles. arXiv Preprint arXiv:1610.01283, 2016.

[29] Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. Trust region policy optimization. 2015. https://arxiv.org/abs/1502.05477

[30] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv Preprint arXiv:1707.06347, 2017.

[31] Konda VR, Tsitsiklis JN. Actor-critic algorithms. In: Advances in Neural Information Processing Systems. 2000. 1008−1014.

[32] Kingma DP, Welling M. Auto-encoding variational Bayes. arXiv Preprint arXiv:1312.6114, 2013.

[33] Ha D, Eck D. A neural representation of sketch drawings. arXiv Preprint arXiv:1704.03477, 2017.

[34] Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M. Tensorflow: A system for large-scale machine learning. 2016. https://arxiv.org/abs/1605.08695

附录 1 深度强化学习背景知识

强化学习(reinforcement learning,简称 RL)是机器学习的一个子领域,学习如何将场景(环境状态)映射到动作,以获取能够反映任务目标的最大的数值型奖赏信号,即在某种环境状态下,决策选择何种动作去改变环境,使得获得的收益最大(策略,状态到动作的映射).现有强化学习方法利用马尔可夫决策过程(Markov decision process,简称 MDP)从理论方面对 RL 问题进行基础建模.MDP 由一个五元组$\langle S,A,R,T,\gamma\rangle$定义,其中,$S$ 表示由有限状态集合组成的环境;A 表示可采取的一组有限动作集;状态转移函数 $T:S\times A\to\Delta(S)$表示将某一状态-动作对映射到可能的后继状态的概率分布,$\Delta(S)$表示状态全集的概率分布,对于状态 $s,s'\in S$ 以及 $a\in A$,函数 T 确定了采取动作 a 后,环境由状态 s 转移到状态 s'的概率;奖赏函数 $R(s,a,s')$定义了状态转移获得的立即奖赏;γ是折扣因子,代表长期奖赏与立即奖赏之间的权衡.与基本的强化学习方法相比,DRL 将深度神经网络作为函数近似和策略梯度的回归函数.虽然使用深度神经网络解决强化学习问题缺乏较好的理论保证,但深度神经网络的强大表现力使得 DRL 的结果远超预期.

近端策略优化

在非凸优化的情况下,梯度可以用数值方法或抽样方法计算,但很难确定适当的迭代学习率,需要随时间变化以确保更好的性能.早期的强化学习研究在使用基于梯度的优化技术时也遇到了这样的困境,为了规避瓶颈,Schulman 等人[29]提出了一种处理随机策略的信任域策略优化(trust region policy optimization,简称 TRPO)算法,该算法在目标函数中考虑了旧策略和更新策略之间的 Kullback-Leibler(KL)发散,并能对每个状态点的 KL 发散进行有界处理.该方法跳出了对学习率的修正,使策略改进过程更加稳定,理论证明,该方法单调地增加了累积奖赏.考虑到 TRPO 中二阶 Hessian 矩阵计算的复杂性,Schulman 等人[30]进一步发展了一阶导数近端策略优化(proximal policy optimization,简称 PPO)算法.

近端策略优化方法与 TRPO 方法一样,定义了 surrogate 目标:

$$\max L^{CPI}(\theta)=\max\hat{\mathbb{E}}_t[r_t(\theta)\hat{A}_t],\quad r_t(\theta)=\frac{\pi_\theta(a_t\mid s_t)}{\pi_{\theta_{old}}(a_t\mid s_t)},$$

其中,π代表策略,$\pi_{\theta_{old}}$代表上一时刻的策略,$\hat{A}_t$估计了动作 a_t在状态下 s_t的优势函数.

在 PPO 中,对上述代理目标进行了裁剪:

$$L^{CLIP}(\theta)=\hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t,\mathrm{clip}(r_t(\theta),1-\varepsilon,1+\varepsilon)\hat{A}_t],$$

$$\mathrm{clip}(x,x_{\mathrm{MIN}},x_{\mathrm{MAX}})=\begin{cases}x, & \text{if } x_{\mathrm{MIN}}\leqslant x\leqslant x_{\mathrm{MAX}}\\ x_{\mathrm{MIN}}, & \text{if } x<x_{\mathrm{MIN}}\\ x_{\mathrm{MAX}}, & \text{if } x_{\mathrm{MAX}}<x\end{cases}.$$

该目标 $L^{CLIP}(\theta)$实现了一种与随机梯度下降兼容的信赖域修正方法,并通过消除 KL 损失来简化算法以及减小适应性修正的需求.

附录 2 训练步骤

预训练 VMAV-C 模型

VAE、MDN-RNN、AVF 和 Controller 模型的目的是同时学习环境中状态的表示和动态转移,但是网络结构的庞大参数和复杂性使得 VMAV-C 的训练变得困难和耗时,因此,同步预训练 VMAV 是本文实验中的必要步骤.为了实现这一目标,首先需要与实际环境进行一系列的互动,利用随机策略获得多个完整的训练片段,如步骤 0 中的集合 $\{episode=\{(x_t,a_t,x_{t+1},r_{t+1},d_{t+1})\}\}$.环境的屏幕截图用作 VAE 的训练数据集,在相对简单的任务中初始采样包含了环境的动态信息,尤其是状态表示和有关环境转移的信息.

在图 6 所示的步骤 1 中,将作为 VAE 输入的整个状态数据集随机分为两部分:75%用于训练过程,其余部分用于测试重构性能.在此过程中,通过对测试数据集重构误差的监测,有效地探索了图像的环境潜在空间.一旦完成 VAE 的训练过程,用低维向量编码采集到的图像作为 MDN-RNN 模型的输入.

在将采样片段应用于 MDN-RNN 之前,本文首先按照时间顺序将这些片段合并成一个长序列,然后分割成固定长度的小序列作为数据集学习 MDN-RNN.经过几次迭代后,AVF 将加入到预训练过程中.步骤 2 得到了一个隐藏在 MDN-RNN 中的虚拟环境.

算法 1. 预训练 VMAV-C 模型.

输入:利用随机参数初始化 VAE、MDN-RNN、AVF;

输出:训练好的 VAE、MDN-RNN 以及预训练的 AVF.

(1) 利用随机策略通过环境交互 N 次,存储所有的动作、观测、奖赏以及结束标记.
$\{episode=\{(x_t,a_t,x_{t+1},r_{t+1},d_{t+1})\}\}$ 到内部存储 D 中

(2) 收集所有的观测 $\{x_t\}$ 训练 VAE 模型

While VAE 未收敛 **do**:

采样观测的 mini-batch

$$loss_{vae}=\frac{1}{N}\sum_{i=1}^{N}\left[(VAE(x_i)-x_i)^2+\frac{1}{2}\sum_{j=1}^{k}(\mu_{x_i}^{(j)2}+\sigma_{x_i}^{(j)2}-\ln\sigma_{x_i}^{(j)2}-k)\right]$$

后向传播更新 VAE //默认优化器是 RMSProp

(3) 收集 MDN-RNN 的训练数据集

For episode in storage D:

将片段转换成固定长度为 L 的序列

For each time step:

规范化采集样本为 $(z_t=VAE_{Enco}(x_t),a_t,z_{t+1}=VAE_{Enco}(x_{t+1}),r_{t+1},d_{t+1})$

将这些 mini-sequence 存储至内存 $M_{MDN\text{-}RNN}$

(4) 训练 MDN-RNN

While MDN-RNN 未收敛 **Do**:

从 $M_{MDN-RNN}$ 中采样 batch

计算损失函数 $L_{total}=\beta_1\times L_s+\beta_2\times L_p$

后向传播更新 MDN-RNN //默认优化器是 Adam

(5) 训练 AVF

While AVF 未收敛 **Do**:

从 $M_{MDN-RNN}$ 中采样 mini-batch

生成 r_{out},并按照图 5 所示规范化数据集

使用 n-step 返回:

$$\tilde{V}=\begin{cases}\sum_{t=0}^{T-1}\lambda^t r+AVF(z_T,h_T), & \text{if } d_T=0\\ \sum_{t=0}^{T-1}\lambda^t r+0, & \text{if } d_T=1\end{cases}$$

损失函数 $loss_{AVF}=-\mathbb{E}(\tilde{V}-AVF(z,h))^2$

后向传播更新 AVF //默认优化器为 Adam

训练 MAV-C 模型

步骤 2 学习了一个基于 MDN-RNN 的虚拟环境模型,从理论上揭示了状态和奖赏信号的转移.在步骤 3 中,通过与虚拟环境的交互训练 AVF 和控制器,并利用 PPO 算法对控制器模型进行优化.在步骤 4 中,使用步骤 3 中的 VAE、MDN-RNN 和训练有素的控制器在实际环境中进行决策.此外,除了在策略学习中使用虚拟环境信息外,步骤 4 也是利用 MDN-RNN 在真实环境中训练控制器模型的过程.算法 2 给出了离散环境下的控制模型训练过程.

算法 2. 基于 PPO 的 MAV-C 模型训练.

输入:训练好的 VAE、MDN-RNN 以及预训练好的 AVF 模型.

(1) 初始化环境,采样获得初始状态

(2) For i=0,1,...,K:

驱动 agent 与虚拟环境 MDN-RNN 进行交互,收集训练中 RNN 的隐藏信息 h,本征向量 z,动作 a 和奖赏 r

利用 PPO 算法优化控制器模型: $L^{CLP}(\theta)=\mathbb{ET}[\min(r_t(\theta)\tilde{A}_t,\text{clip}(r_t(\theta),1-\varepsilon,1+\varepsilon)\tilde{A}_t]$

AVF 损失: $\min_w E\left(\sum_{t=1}^{T}\lambda^{t-1}r_t+AVF(h_T,z_T,w)-AVF(h,z,w)\right)^2$

End For

附录 3 实验设置

经典控制任务

经典控制任务中环境图片的大小初始为 400×600.我们发现 CartPole-V0 环境中的大部分区域都是空白的,因而,我们将倒立摆的位置作为图片中心,将原图裁剪为 160×320 大小(如果倒立摆的位置靠近边缘,则截取边缘的 320 个像素点),之后进一步将图片压缩为 40×80 大小.对于 MountainCar-V0 以及 Acrobot-V1,为了保持图片原本的比例信息,我们将其压缩为 80×120 大小.3 种任务的 VAE 网络架构相似,在图 11 中进行了展示.其中,在实验中,潜变量服从 32 维多元正态分布.3 种任务的采集数据设置见表 3,其中,MountainCar-V0 和 Acrobot-V1 的 kernel_size 和 stride 设计由下方粗体公式表示.

在获得任务的本征空间的基础上,我们对上述获得的数据中的图片状态进行压缩,对 MDN-RNN 模型进行训练.在训练过程中,我们将训练样本拼接在一起构成训练集,并按照 32-step 的长度对其逐位进行切割;将剩余的测试样本作为测试集.这样操作避免了 done 结束位仅出现在最后一个时间片中的问题,done 可以出现在每个时间序列中的任一位置,克服了 MDN_RNN 模型利用该缺陷获得不良预测模型.然而,这样的操作却带来了起始状态的隐藏层信息不够准确,起始状态的隐藏层信息应来自于初始化的隐藏层信息,而非上一结束时间片传递而来的隐藏状态信息.本文利用 LSTM cell,对每一个 batch 中当前时间片的后续隐藏信息是否初始化进行判断.采用算法 1 中的步骤 4 对 MDN-RNN 进行预训练,各个环境中的参数设置见表 4.

Table 3 Data collecting setting in classic control environment

表 3 经典控制任务下采集数据设置

环境名称	数据量	采集信息	备注
CartPole-V0	2 000episode:1 500 个训练,500 个测试	$\langle state, action, next\_state, done(bool)\rangle$	每个 episode 记录了从开始到结束
MountainCar-V0	20 000step:16 000 步训练,4 000 步	$\langle state, action, next\_state, done(bool), dis\rangle$	一个 episode 太长,因而使用 step 统计,并额外统计当前与终点之间的距离 dis,可以推导出奖赏
Acrobot-V1	30 000step:25 000 步训练,5 000 步	$\langle state, action, next\_state, done(bool)\rangle$	一个 episode 太长,因而使用 step 统计

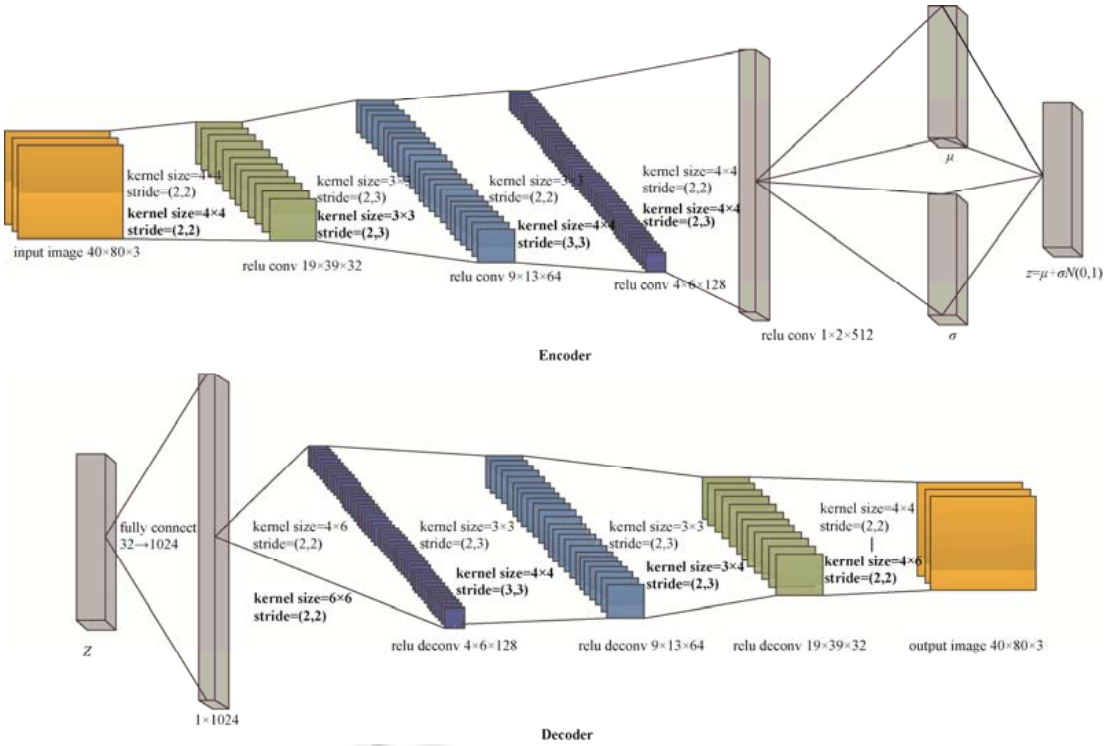


Fig.11 The network structure of VAE in classic control environment

图 11 经典控制任务中的 VAE 结构

Table 4 MDN-RNN parameter setting in classic control environment

表 4 经典控制任务中 MDN-RNN 的参数设置

环境名称	β_1	β_1	α	lr	Batch_size	Optimer	τ	λ	PPO 中的ε
CartPole-V0	1	1	1	4.77e-5	256	Adam	1	0.95	0.1
MountainCar-V0	1	1	-	1e-5	128	Adam	1	0.99	0.1
Acrobot-V1	1	1	100	1e-5	128	Adam	1	0.99	0.1

导弹智能突防实验

在导弹智能突防任务中,我们构建了红方的一枚导弹突破蓝方的两枚导弹,并命中目标的智能突防场景.在这一场景中,我们利用 VMAV-C 方法对红方导弹的突防策略进行学习,即根据当前的红蓝双方的信息,决策导弹的变轨矢量以及变轨时间;蓝方的拦截策略由内置的规则进行控制.

为了获得初始的样本,我们利用随机策略、内置策略以及预训练策略(根据第 1 次虚拟训练获得的策略)对红方的导弹进行控制,分别进行了 1 000 次的仿真实验,从导弹进入可规划段开始采集数据,每隔 2s 对导弹进行 1 次规划,将当前的红蓝双方导弹特征作为一次观测,存入训练数据集合 $\{episode=\{(x_t,a_t,x_{t+1},r_{t+1},d_{t+1})\}\}$ 中,其中,x_t 包含红蓝双方的特征.智能突防场景中的 VAE 结构图如图 12 所示,将原始的红蓝双方的特征信息经过两层全连接层压缩为一个 8 维向量编码,经过采样后,利用两层的全连接层将采样编码解码为同维度的特征.在实验中,潜变量服从 8 维多元正态分布.

与 CartPole 的固定奖赏相同,该任务下的奖赏是可变的,因而需要利用 MDN-RNN 的隐藏状态预测下一时刻的奖赏和结束位表示.智能突防场景中 MDN-RNN 的结束位预测包含 3 种情况:仿真未结束;击中目标;未击中目标.此外,在 VAE 获得的潜在状态空间的基础上,对训练样本中的数据进行压缩编码.在训练过程中,我们将前 800 个 episode 拼接在一起构成训练集,并按照 32-step 的长度对其逐位进行切割;将剩余的 200 个 episode 作为测试集.采用算法 1 中的步骤 4 对 MDN-RNN 进行预训练,其中,参数设置为β_1=1,β_2=2.5,batch 大小设置为 128,优化器选择为 Adam,学习率设置为 l*e*–5,随机性控制参数τ=1.在神经网络设计中,我们将动作的 8 维嵌入信息和

状态的编码表示作为输入,经过 3 层的 LSTM,获得隐藏层信息,根据这一隐藏信息,分别输出 5 个高斯分布的 mean 和 log-sigma 以及它们的权重,同时输出对 d_{t+1} 的预测.在基于注意力的值函数中,我们采用当前 4 个隐藏信息来获得注意向量.本文利用 PPO 算法对控制器模型进行训练,利用高斯分布在连续动作空间内采样动作.

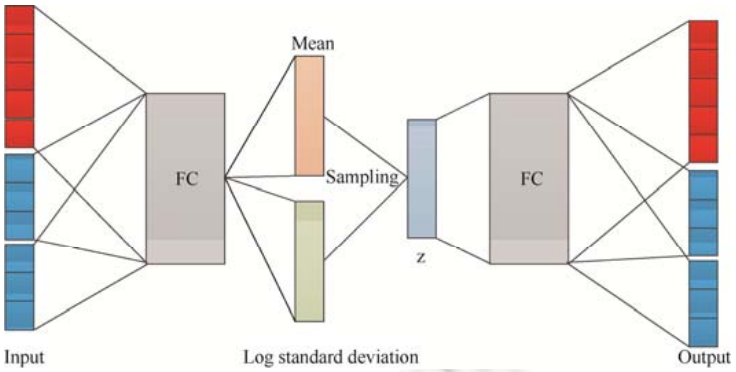


Fig.12 The network structure of VAE in task of intelligence penetration

图 12 智能突防场景中的 VAE 结构

附录 4 VMC 决策过程

```
obs=env.reset()
h=rnn.initial_state() //初始化 RNN 的隐藏状态
done=Flase
cumulative_reward=0 //初始化累计奖赏
While not done:
z=vae.encode(obs) //编码环境观测,获得状态的潜在表示
a=controller(z,h) //输入观测的潜在表示和 RNN 的隐藏状态
next_obs, reward,done,_=env.step(a) //在环境中执行动作和获得响应
cumulative_reward +=reward
h=rnn.forward(a,z,h) //计算下一时刻的 RNN 隐藏状态
obs=next_obs
return cumulative_reward
```

附录 5 软硬件环境

软件:使用的神经网络框架为 pytorch=0.4.1,torchvision=0.2.1,数据可视化软件为 tensorflow=1.13.1,tensorboardX=1.4,数据处理工具 numpy=1.14.6,强化学习环境为 openAI gym=0.10.5,mujoco=1.50.1.56.

硬件:本文所使用的微机环境为:一块华硕 1080TI 显卡,一块 8 核 Inter I7 7820X CPU,主板为华硕 X299,内存为 16G,硬盘为 256G 的固态硬盘.

附录 6 MDN-RNN 虚拟环境场景展示

本文从经典控制任务中采样初始状态作为 MDN-RNN 环境的第 1 帧状态,在此基础上,每两步进行一次还原,进行 32 步仿真,获得的场景图如图 13～图 15 所示.

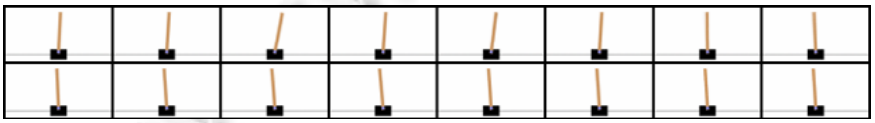


Fig.13 Virtual simulation in CartPole-v0

图 13 CartPole-v0 环境中的虚拟仿真

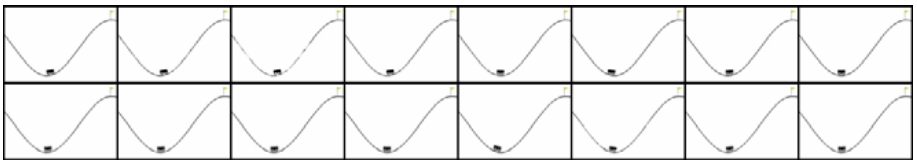


Fig.14 Virtual simulation in MountainCar-v0
图 14 MountainCar-v0 环境中的虚拟仿真

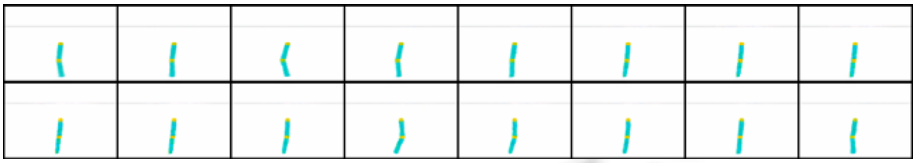


Fig.15 Virtual simulation in Acrobot-V1
图 15 Acrobot-V1 环境中的虚拟仿真

梁星星(1992－),男,山西永济人,硕士,主要研究领域为多 agent 智能规划,多 agent 深度强化学习.

王琦(1992－),男,硕士,主要研究领域为不确定性可控的强化学习,贝叶斯统计学习.

冯旸赫(1985－),男,博士,副教授,主要研究领域为因果发现与推理,主动学习,强化学习.

马扬(1993－),男,硕士,主要研究领域为网络嵌入,链路预测,图神经网络.

黄金才(1973－),男,博士,教授,博士生导师,主要研究领域为智能调度与控制.

刘忠(1968－),男,博士,教授,博士生导师,主要研究领域为多智能体系统.