第 36 卷 第 1 期
2021 年 2 月

天津科技大学学报
Journal of Tianjin University of Science & Technology

Vol. 36 No. 1
Feb. 2021

DOI:10.13364/j.issn.1672-6510.20200044

一种基于条件生成对抗网络的模型化策略搜索方法

孔 乐，赵婷婷
（天津科技大学人工智能学院，天津 300457）

摘 要：模型化强化学习是深度强化学习领域中的一种有效学习模式，能够缓解强化学习在实际应用中样本利用率低的瓶颈问题. 然而，受环境复杂性及动态性影响，学习得到准确的状态转移环境模型极具挑战. 为此，本文提出一种基于条件生成对抗网络的复杂环境中有效的模型化策略搜索强化学习方法. 该方法首先利用条件生成对抗网络对环境中的状态转移函数学习，再利用经典的策略搜索方法进行策略学习. 通过实验验证，该方法能够准确地生成状态转移数据，为策略学习提供充足的学习样本，从而得到稳定、高性能的策略.

关键词：条件生成对抗网络；模型化强化学习；策略搜索；状态转移函数；环境模型
中图分类号：TP391 **文献标志码**：A **文章编号**：1672-6510(2021)01-0068-07

A Model-based Policy Search Method Based on Conditional Generative Adversarial Network

KONG Le，ZHAO Tingting
(College of Artificial Intelligence, Tianjin University of Science & Technology, Tianjin 300457, China)

Abstract: Model-based reinforcement learning (Mb-RL) is an effective method for deep reinforcement learning, as it can alleviate the bottleneck problem of data inefficiency, but it is extremely challenging to obtain an accurate state transition model with it in a high-dimensional environment due to the complexity and dynamics of the environment. Therefore, this paper proposes an effective policy search reinforcement learning method in complex environments. More specifically, conditional generative adversarial network is used first for state transition function learning, and then classical policy search method for policy learning. Extensive experimental results demonstrate that the proposed method can accurately generate state transition data, provide sufficient samples for policy learning, and thus generate stable and high-performance policy.

Key words: conditional generative adversarial network; model-based reinforcement learning; policy search; state transition function; environment model

深度强化学习(deep reinforcement learning, DRL)[1]是一种以试错机制与环境交互并最大化累积回报获得最优策略的机器学习范式. 为得到最优策略，DRL 要求智能体能够对周围环境有所认知、理解并根据任务要求做出符合环境情境的决策动作. 目前，DRL 已在智能对话系统[2]、无人驾驶车[3-4]、存储系统[5]、智能电网[6]、智能交通系统[7]、机器人系统[8]、航空航天系统[9]、游戏[10]及数字艺术智能系统[11]等领域取得突破性进展.

根据学习过程中环境模型是否可用，强化学习可分为模型化强化学习[12](model-based reinforcement learning, Mb-RL)和模型强化学习[12](model free reinforcement learning, Mf-RL). 环境模型即系统动力学模型，是对状态转移函数的描述. Mf-RL 方法中，环境模型是未知的，智能体必须与真实环境进行大量交互获得足够多的训练样本才能保证智能体的决策性

收稿日期：2020-03-29；**修回日期**：2020-06-25
基金项目：国家自然科学基金资助项目(61976156)；天津市教委计划科研项目(2017KJ034)
作者简介：孔 乐(1994—)，女，山东曲阜人，硕士研究生；**通信作者**：赵婷婷，副教授，tingting@tust.edu.cn

能. 因此，Mf-RL 方法样本利用率较低，如 RainbowDQN 算法至少需要 1 800 万帧的训练样本或大约 83 h 的训练时间才能学会玩游戏，而人类掌握游戏所用时间远远少于此算法[13]. 尽管 Mf-RL 方法在诸如游戏等虚拟决策任务中取得了良好的性能，但对于真实环境中的决策任务，收集充分的训练样本不仅需要大量时间与财力，样本收集过程还对系统硬件配置提出了较高要求，甚至存在损坏智能系统的风险. 另外，训练样本不足会导致智能体无法从少量训练样本中提取有用信息进行准确策略更新. 相比之下，Mb-RL 方法在对环境精准建模后，智能体无需与真实环境互动就可以进行策略学习，可直接与环境模型交互生成所需训练样本，从而在一定程度上缓解强化学习在实际应用中学习效率低、样本利用率低的问题.

模型化强化学习方法的基本思想是首先对环境动态建模，学习环境模型参数，当模型参数训练收敛得到稳定环境模型后，智能体便可直接与预测环境模型交互进行策略学习[14]. 整个过程中，仅在学习模型参数时需要一定训练样本，样本需求量相对较小. 然而，受环境噪声、系统动态性等因素影响，预测的环境模型通常难以准确描述真实环境，即学到的环境模型与真实环境间存在模型误差[15]. 使用存在模型误差的环境模型生成数据进行策略学习将会产生更大误差，最终导致任务失败. 为此，研究人员提出了一系列减小模型误差、提高环境模型准确性的方法，如 Dyna 模型化强化学习框架[16]、嵌入控制方法[17]，基于神经网络动力学和无模型微调的模型化深度强化学习方法(E2C)[18]、世界模型[19]等方法. 其中，Dyna 框架是 Mb-RL 中最经典的学习模式，学习控制的概率推理方法(probabilistic inference for learning control，PILCO)[20]和基于最小二乘条件密度估计方法的模型化策略搜索算法(Mb-PGPE-LSCDE)[21]是 Dyna 框架下经典的 Mb-RL 方法. PILCO 方法已广泛应用在机器人控制等领域，然而该方法将状态转移函数建模为高斯过程，且对回报函数也作了相应假设，这极大程度限制了它的实际应用；LSCDE 方法能够拟合任意形状的状态转移函数，但是当处理高维度状态空间问题时存在模型表达能力不足的缺陷.

近年，针对不同的应用场景，研究者提出了一系列基于 Mb-RL 的相关工作，如使用少量交互数据便可实现指定轨迹跟踪任务的基于神经网络动力学和无模型微调的模型化深度强化学习方法(MBMF)[18]，支持图像长期预测和复杂控制的嵌入控制方法(E2C)[17]，易于复现、可实现快速学习并迁移至真实环境的世界模型方法[22]，使用变分自编码(variational autoencoder，VAE)[23]方法捕捉状态转移函数的方法等. 上述相关工作在各自应用领域虽然已经取得较好成果，但是面向大规模复杂动态环境如何得到准确环境模型，仍是该领域亟待解决的问题.

生成对抗网络(generative adversarial networks，GAN)[24]是 Goodfellow 于 2014 年提出的生成模型，它在数据生成方面取得巨大进展，并已广泛应用于图像风格迁移[25]、视频预测[26]、自然语言处理[27]等领域. GAN 由生成器(generator，G)和判别器(discriminator，D)组成，生成器 G 旨在生成趋近真实数据分布的伪造数据，判别器 D 则旨在正确区分伪造数据和真实数据，二者在对抗中逐渐达到纳什均衡.

本文借助 GAN 在数据生成方面的优势，提出一种基于 GAN 的环境模型学习方法. 条件生成对抗网络(conditional generative adversarial networks，CGAN)对生成器 G 和判别器 D 分别作了限定，是 GAN 的变体之一，同样具备 GAN 的诸多优势[28]. 该方法是将 CGAN 与 Mb-RL 结合应用在学习状态转移模型上的首次尝试. 本文将 CGAN 与擅长处理连续动作空间的策略搜索方法结合，提出一种基于 CGAN 的模型化策略搜索方法. 与传统环境模型学习方法相比，该方法优势在于：传统概率生成模型需要马可夫链式的采样和推断，而 GAN 避免了此类计算复杂度高的过程，在一定程度上提高了生成模型在学习环境模型中的应用效率；GAN 的对抗训练机制可以逼近任意复杂的目标函数，使得在概率密度不可计算时，基于 GAN 的环境模型学习方法依然适用.

1　相关理论

1.1　问题模型

强化学习是指智能体在未知环境中，通过不断与环境交互，学习最优策略的学习范式. 智能体是具有决策能力的主体，通过状态感知、动作选择和接收反馈与环境互动. 通常，智能体与环境的交互过程可建模为马尔可夫决策过程(markov decision process，MDP)[29]，一个完整的 MDP 由状态、动作、状态转移函数、回报构成的五元组(S、A、P、P_0、R)表示，其中：S 表示状态空间，是所有状态的集合，s_t 为 t 时刻所处状态；A 表示动作空间，是所有动作的集合，a_t 为 t 时刻所选择的动作；P 表示状态转移概率，即环境模型，根据状态转移概率是否已知，强化学习方法分为

Mb-RL 和 Mf-RL；P_0 表示初始状态概率，是随机选择某一初始状态的可能性表示；R 表示智能体的累积回报，r_t 为 t 时刻的瞬时回报.

在每个时间步长 t，智能体首先观察当前环境状态 s_t，并根据当前策略函数决策选择并采取动作 a_t，所采取动作一方面与环境交互，依据状态转移概率 $p(s_{t+1}|s_t,a_t)$ 实现状态转移，另一方面获得瞬时回报 r_t，该过程不断迭代 T 次直至最终状态，得到一条路径 $h^n:=\left[s_1^n,a_1^n,\cdots,s_T^n,a_T^n\right]$.

强化学习的目标是找到最优策略，从而最大化期望累积回报. 当得到一条路径后，便可计算该路径的累积回报

$$R(h):=\sum_{t=1}^{T}\gamma^{t-1}r(s_t,a_t,s_{t+1}) \tag{1}$$

其中 $0\leqslant\gamma<1$，决定回报的时间尺度.

累积回报的期望衡量策略好坏，累积回报期望为

$$J_\pi:=\int p(h)R(h)\mathrm{d}h \tag{2}$$

其中：$p(h)=p(s_1)\prod_{t=1}^{T}p(s_{t+1}|s_t,a_t)\pi(a_t|s_t)$ 为发生路径的概率密度函数. 强化学习的目标是找到最优策略 π^*，该策略可以最大化期望回报 J_π.

$$\pi^*:=\arg\max_\pi J_\pi \tag{3}$$

1.2　策略搜索方法

策略搜索方法是一种策略优化方法，该方法直接对策略进行学习，适用于解决具有连续动作空间的复杂决策任务[13]，本文将使用策略搜索方法进行策略学习.

策略搜索方法的学习目的是找到可最大化累积回报期望值 $J(\theta)$ 的参数 θ，即最优策略参数 θ^* 为

$$\theta^*:=\arg\max_\theta J(\theta) \tag{4}$$

其中 θ 是策略参数，累积回报期望 $J(\theta)$ 是策略参数 θ 的函数.

$$J(\theta):=\int p(h|\theta)R(h)\mathrm{d}h \tag{5}$$

式中：$p(h|\theta)=p(s_1)\prod_{t=1}^{T}p(s_{t+1}|s_t,a_t)\pi(a_t|s_t,\theta)$；

$$R(h):=\sum_{t=1}^{T}\gamma^{t-1}r(s_t,a_t,s_{t+1})$$

其中 $0\leqslant\gamma<1$，决定回报的时间尺度.

目前，最具代表性的策略搜索算法有 PEGASUS[13]、策略梯度方法[30-31]、自然策略梯度方法[32]等. 其中，策略梯度方法是寻找最优策略参数最简单、最常用的方法. 鉴于策略梯度方法中的近端策略优化方法（proximal policy optimization，PPO）的优越性能，本文使用 PPO 算法进行策略学习[33].

1.3　生成对抗网络

GAN 由生成器（generator，G）和判别器（discriminator，D）组成，如图 1 所示，其中：黑色框图为原始生成对抗网络网络结构，对生成器 G 和判别器 D 分别添加条件变量 y 后（红色虚线框图），网络结构为条件生成对抗网络示意图. 生成器 G 实现随机变量假样本数据 $G(z)$ 的映射，z 通常为服从高斯分布的随机噪声，生成器 G 的目的是使假样本数据 $G(z)$ 与真实数据 x 高度相似. 判别器 D 接收真实数据 x 或假样本数据 $G(z)$ 并输出概率值，该概率值表征输入数据是真实数据的几率. 若数据是真实数据，判别器 D 输出大概率；否则，判别器 D 输出小概率.

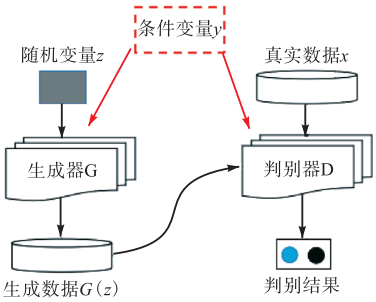


图 1　GAN 结构示意图

Fig. 1　Diagram of generative adversarial network

训练过程中，生成器 G 和判别器 D 不断交替更新模型参数，最终到达纳什均衡. 训练过程可表示为关于值函数 $V(D,G)$ 的极大化与极小化的博弈问题，其目标函数可表示为

$$\min_G\max_D V(D,G)=E_{x\sim P_{\mathrm{data}}(x)}[\log D(x)]+E_{z\sim P_z(z)}[\log(1-D(G(z)))] \tag{6}$$

式中：$V(D,G)$ 表示损失值；x 表示真实数据分布的采样；z 表示随机噪声变量.

鉴于 GAN 在数据生成方面的优势以及在强化学习领域取得的成功，本文拟用同样具有优秀数据生成能力的 GAN 分支之一的 CGAN 学习环境中的状态转移概率密度函数 $P_T(s_{t+1}|s_t,a_t)$. 其中，CGAN（如图 1 所示）额外在生成模型 G 和判别模型 D 中引入条件变量 y 对模型增加限定，用于指导数据生成过程. CGAN 的损失函数为

$$\min_G\max_D V(D,G)=E_{x\sim P_{\mathrm{data}}(x)}[\log D(x|y)]+E_{z\sim P_z(z)}[\log(1-D(G(z|y)))] \tag{7}$$

2　实现方法

2.1　算法执行步骤

Mb-RL 方法需要首先学习得到精准的状态转移模型，策略学习阶段便利用该模型生成所需样本，减少智能体与环境的交互次数.

本文所提的基于 CGAN 的模型化策略搜索方法，首先通过用 CGAN 学习序列数据 $\{s_0,a_0,s_1,a_1,\cdots\}$，得到状态转移函数的预测，即 $\hat{p}(s_{t+1}|s_t,a_t)$. 得到预测的状态转移函数后，当输入一个状态动作对 $[s_t,a_t]$ 时，无需等待真实环境反馈，可直接利用学到的状态转移函数 $\hat{p}(s_{t+1}|s_t,a_t)$ 预测下一状态 s_{t+1}. 本文所提出的 CGAN-MbRL 算法框架流程如下：

(1) 收集真实状态转移样本 $\{(s_t,a_t,s_{t+1})\}_{t=1}^{T}$.

(2) 利用 CGAN 对状态转移函数 $p_T(s_{t+1}|s_t,a_t)$ 进行建模，使用第 1 步搜集到的样本 $\{(s_t,a_t,s_{t+1})\}_{t=1}^{T}$ 进行模型的训练.

(3) 智能体与第 2 步得到的状态转移模型 $p_T(s_{t+1}|s_t,a_t)$ 交互，得到足够多的样本序列 $\{\tilde{h}_n\}_{n=1}^{\tilde{N}}$ 进行策略学习.

(4) 更新策略模型中的参数直至收敛，最终得到最优策略 π^*.

2.2　基于 CGAN 的环境模型学习方法

Mb-RL 方法中，当状态转移模型能够完全模拟真实环境时，智能体只需与学到的状态转移模型 $\hat{p}(s_{t+1}|s_t,a_t)$ 交互便可得到下一状态 $\hat{s}_{t+1}$，从而减少智能体与真实环境的交互. 因此，如何得到真实环境的状态转移函数是 Mb-RL 方法的关键. 本文使用 CGAN 捕捉真实环境的状态转移函数分布(图 2).

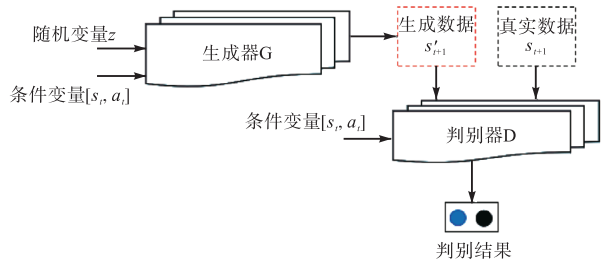


图 2　基于 CGAN 的环境模型学习方法

Fig. 2　Method of environment learning based on CGAN

状态转移函数中下一状态 s_{t+1} 受当前状态 s_t 和当前状态下采取动作 a_t 的限定，是一个条件概率密度模型，表示为 $P_T(s_{t+1}|s_t,a_t)$. 因此，本文将当前状态 s_t 和当前状态下采取动作 a_t 作为 CGAN 的条件变量 y 对生成器 G 和判别器 D 同时增加限定，指导下一状态 s_{t+1} 生成. 该条件变量 y 和随机变量 z 同时作为生成器 G 的输入，此时生成器 G 的输出是当前状态下 s_t 执行动作 a_t 到达的下一状态 s_{t+1}. 将该输出与真实样本数据连同条件变量 y 同时输入到判别器 D 中，可估计一个样本来自于训练数据的概率. 上述过程目标函数可表示为

$$\min_G \max_D V(D,G)=E_{s_{t+1}\sim P_{\text{data}}(s_{t+1})}[\log D(s_{t+1}|(s_t,a_t))]+E_{z\sim P_z(z)}[\log(1-D(G(z|(s_t,a_t))))] \tag{8}$$

在 CGAN 模型训练稳定后，可直接将训练稳定的生成器 G 作为环境预测模型，与智能体交互生成大量样本数据用于策略学习.

2.3　策略搜索方法

在 2.2 节的基础上，得到稳定高效的状态转移模型 $\hat{p}(s_{t+1}|s_t,a_t)$ 后，本文选择经典近端策略优化(proximal policy optimization，PPO)方法进行策略的学习. PPO 方法是策略梯度方法的改进，传统策略梯度方法存在参数更新慢，每次更新均需重新采样的问题，而 PPO 方法结构简单，一次采样可多次更新策略参数，样本利用率高，且能够自动调整参数空间步长达到策略空间均匀变化的目的，其期望累积回报为

$$L^{clip}(\rho)=\hat{E}_t[\min(r_t(\rho)\hat{A}_t,clip(r_t(\rho),1-\varepsilon,1+\varepsilon)\hat{A}_t)] \tag{9}$$

式中：ρ 为策略参数；$\hat{E}_t$ 为期望；ε 为常数，通常取 0.1 或 0.2；r_t 为新策略与旧策略的比值；$\hat{A}_t$ 为 t 时刻的优越性；$clip$ 项使得 r_t 不偏离 $[1-\varepsilon,1+\varepsilon]$ 所定义的区间.

综上，本方法将 CGAN 与 PPO 结合寻找最优策略，其中 CGAN 将状态动作空间模型化为状态转移模型 $\hat{p}(s'|s,a)$，随后利用学到的状态转移模型生成样本用于 PPO 的策略学习，从而得到最优策略 π^*.

3　环境模型测试实验

3.1　实验环境

玩具问题能够快速地验证算法有效性，先将原始复杂问题转化为简单问题，再进行求解. 本节将强化学习中的环境模型简化为四核高斯分布，探索本文所提的基于 CGAN 的环境模型学习方法在捕捉数据分布方面的能力.

3.2　实验设置

本实验将模拟实现基于 CGAN 的环境模型学习

方法的学习过程. 实验目的是使用基于 CGAN 的环境模型学习数据分布，其中 CGAN 中生成器 G 和判别器 D 的网络模型均为多层感知机. 实验中各变量设置如下：变量 y 代表 CGAN 中的条件变量，该条件变量对应强化学习中 t 时刻的状态 s_t 和动作 a_t，即 $[s_t,a_t]$；x 为真实数据，对应强化学习中的下一状态 s_{t+1}. 实验从 (0，1) 区间随机采样得到条件 y，经过二维转移函数映射得到真实数据 x.

$$P(x\mid y)=\prod_{k=1}^{4}N(x\mid\mu_k,\sum\nolimits_k)^{z_k} \tag{10}$$

其中 $\mu_1=[5,35]$，$\mu_2=[30,40]$，$\mu_3=[20,20]$，$\mu_4=[45,15]$，$\sum=[[30,0],[0,30]]$ 且

$$\begin{cases} z_1=1, 0\leqslant y<0.1 \\ z_2=1, 0.1\leqslant y<0.3 \\ z_3=1, 0.3\leqslant y<0.6 \\ z_4=1, 0.6\leqslant y<1 \end{cases} \tag{11}$$

最终得到的真实数据 x 的分布是四核高斯混合分布. 实验初期，将真实数据 x 归一化到 $[-1,1]$，基于 CGAN 的环境模型学习方法随机选取条件变量集 $[y_1,y_2,y_3,\cdots,y_n]$ 并通过公式 (10) 一一映射得到真实样本数据集 $[x_1,x_2,x_3,\cdots,x_n]$，使用 CGAN 对条件变量集和真实样本数据集建模学习，训练稳定收敛后 CGAN 中生成器 G 可直接生成与真实样本数据高度相似的数据分布.

3.3　实验分析

为了分析本文所提算法在环境数据生成方面的能力，将本文所提出的基于 CGAN 的环境模型学习方法与基于条件变分自编码器 (conditional variational autoencoder，CVAE)[23] 的环境模型学习方法及相关工作 MBMF 算法[18] 所提出的使用神经网络模型 (neural network，NN) 学习环境模型的方法进行对比实验.

本实验拟用基于 CGAN 的环境模型学习方法在玩具问题中对指定形式的环境进行建模学习，实验使用 3 000 个条件变量 y 以及对应的真实样本数据 x 对 CGAN 训练迭代 5 000 次. 测试阶段，将在 (0，1) 区间选取 500 个随机数 $[y_1,y_2,y_3,\cdots,y_{500}]$ 作为条件变量 y 进行预测.

探索使用不同学习方法的学习过程. 图 3 表示使用不同方法对环境进行学习的过程中得到的生成数据与对应真实数据间的误差. 模型训练过程中，每迭代 400 次对模型进行一次测试，计算测试结果 $[\hat{x}_1,\hat{x}_2,\hat{x}_3,\cdots,\hat{x}_{500}]$ 与真实样本数据 $[x_1,x_2,x_3,\cdots,x_{500}]$ 的误差. 由图 3 可知，使用 CGAN 方法的训练初期，生成器 G 与判别器 D 在对抗中学习并不断优化自身，大约 2 000 次迭代模型就可收敛到 0.075，其学习收敛速度最快. 此外，使用 CGAN 方法学习环境模型的性能优于使用 CVAE 和 NN 的方法，利用其得到真实数据与生成数据间的平均距离和方差明显小于对比方法，且其性能也较稳定.

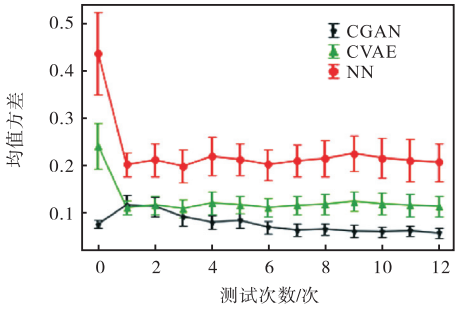


图 3　学习过程中生成数据与真实数据间的误差

Fig. 3　Errors in generative data and real data during the learning process

图 4 表示使用上述 3 种方法预测的状态转移数据与真实数据间的均方差 (mean squared error) 对比结果. 由图 4 可知，使用基于 CGAN 生成数据的准确度明显优于 CVAE 和 NN 方法得到的数据.

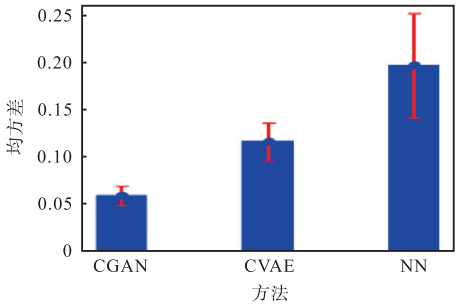
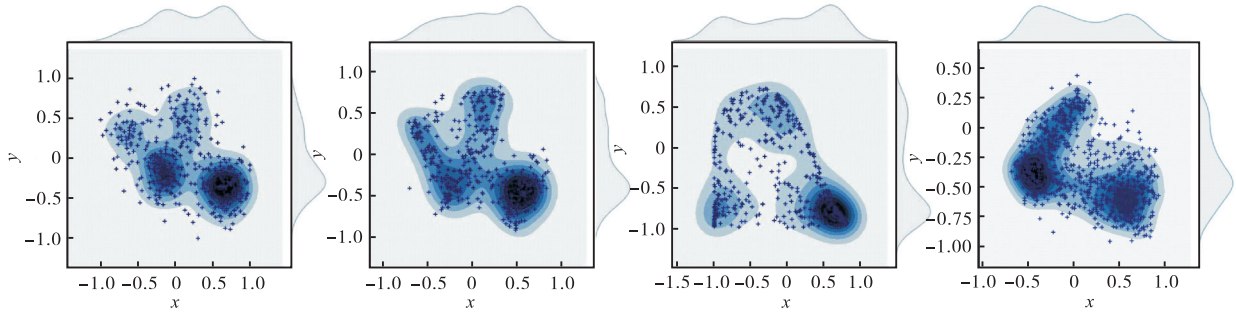


图 4　不同预测方法状态转移数据与真实数据的均方差对比

Fig. 4　Mean squared errors in state transition data predicted by different methods and real data

真实数据分布、使用基于 CGAN 的环境模型方法所得分布以及使用 CVAE、NN 捕捉得到分布的对比结果如图 5 所示，每幅结果图的中间部分表示在条件变量限定下的数据联合分布，上侧和右侧分别表示数据在 x 轴和 y 轴的边缘概率分布. 在模型训练稳定后输入为同一批随机条件变量 $[y_1,y_2,y_3,\cdots,y_{500}]$ 进行对比验证. 图 5(a) 为真实数据分布；图 5(b) 为同样条件变量下使用基于 CGAN 的环境模型方法在 CGAN 模型训练稳定收敛后，仅使用其中的生成器

G 捕捉得到的数据预测分布；图 5(c)为使用 CVAE 方法进行模型训练收敛后，在同样条件变量下捕捉得到的数据预测分布；图 5(d)为使用 NN 方法在模型训练收敛后的数据预测分布. 从图 5 可以看出，在相同条件变量的限定下，基于 CGAN 的环境模型学习方法相比使用 CVAE、NN 方法捕捉数据分布的方法，具有较好的表现性能，不仅可以生成与真实数据分布高度相似的样本，高效地捕捉联合分布，在捕捉边缘概率分布方面也可得到较好结果. 使用 CVAE 方法虽然可以学习到边缘分布的大体分布是双峰的，但并不能很好地捕捉到数据联合分布，最终捕捉到的结果为三核高斯分布，且每个高斯核的数据相对集中. 以上结果是由于 CVAE 中使用的变分方法引入了决定性偏置，优化的是对数似然下界而不是似然度本身，导致了变分自编码器生成的实例比条件生成对抗网络生成的更模糊，进而会导致概率较小的数据很难捕捉到.



(a) 真实数据分布　(b) CGAN 生成数据分布　(c) CVAE 生成数据分布　(d) NN 生成数据分布

图 5　不同方法的生成数据对比

Fig. 5　Comparison of data generated by different methods

综上，本文所提的基于 CGAN 的环境模型学习方法可以用来学习强化学习中的环境模型，并能够取得较好结果，且能较快收敛.

4　结　语

本文将深度强化学习在实际应用中面临的瓶颈问题作为研究背景，对已有模型化强化学习进行详细研究，在条件生成对抗网络的基础上，提出一种基于条件生成对抗网络的模型化策略搜索强化学习方法. 该方法首先利用条件生成对抗网络对环境中的状态转移函数进行学习，再利用经典策略学习方法寻找最优策略. 通过实验验证了该方法能够很好地捕捉到状态转移函数的数据分布，为策略学习提供充足的学习样本.

参考文献：

[1] Sutton R S, Barto A G. Reinforcement Learning: An Introduction[M]. Cambridge: MIT Press, 1998.

[2] Lipton Z C, Li X, Gao J, et al. BBQ-Networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems[EB/OL]. [2020-03-29]. https://arxiv. org/pdf/1711.05715v1.pdf.

[3] Lee D, Choi M, Bang H. Model-free linear quadratic tracking control for unmanned helicopters using reinforcement learning[C]//IEEE. The 5th International Conference on Automation, Robotics and Applications. New York: IEEE, 2011: 6144849.

[4] Konidaris G, Barto A. Autonomous shaping: Knowledge transfer in reinforcement learning[C]//ACM. International Conference on Machine Learning. New York: ACM, 2006: 489–496.

[5] Wu C, Yoshinaga T, Ji Y, et al. A reinforcement learning-based data storage scheme for vehicular ad hoc networks[J]. IEEE Transactions on Vehicular Technology, 2017, 66(7): 6336–6348.

[6] Kim B G, Yu Z, Van D S M, et al. Dynamic pricing for smart grid with reinforcement learning[C]// IEEE. 2014 IEEE Conference on Computer Communications Workshops. New York: IEEE, 2014: 6849306.

[7] 刘智勇，马凤伟. 城市交通信号的在线强化学习控制[C]//第二十六届中国控制会议论文集. 北京：万方数据电子出版社，2007.

[8] Miljkovic Z, Mitic M, Lazarevic M, et al. Neural network reinforcement learning for visual control of robot manipulators[J]. Expert Systems with Application, 2013, 40(5): 1721–1736.

[9] Valasek J, D oebbler J, Tandale M D, et al. Improved adaptive-reinforcement learning control for morphing unmanned air vehicles[J]. IEEE Transactions on Systems

Man & Cybernetics Part B Cybernetics, 2008, 38(4): 1014–1020.

[10] 唐振韬, 邵坤, 赵冬斌, 等. 深度强化学习进展: 从 AlphaGo 到 AlphaGo Zero[J]. 控制理论与应用, 2017, 34(12): 1529–1546.

[11] Xie N, Hachiya H, Sugiyama M. Artist agent: A reinforcement learning approach to automatic stroke generation in oriental ink painting[J]. IEICE Transactions on Information & Systems, 2013(5): 1134–1144.

[12] 周志华. 机器学习[M]. 北京: 清华大学出版社, 2016.

[13] Ng M, Jordan M. Pegasus: A policy search method for large Mdps and Pomdps[EB/OL]. [2020–03–29] https://arxiv.org/abs/1301.3878.

[14] 万里鹏, 兰旭光, 张翰博, 等. 深度强化学习理论及其应用综述[J]. 模式识别与人工智能, 2019, 32(1): 67–81.

[15] Hessel M, Modayil J, Van Hasselt H, et al. Rain-bow: Combining improvements in deep reinforce-ment learning[EB/OL]. [2020–03–29]. https://arxiv.org/abs/1710.02298.

[16] Sutton R S. Dyna, an integrated architecture for learning, planning, and reacting[J]. ACM Sigart Bulletin, 1991, 2(4): 160–163.

[17] Watter M, Springenberg J, Boedecker J, et al. Embed to control: A locally linear latent dynamics model for control from raw images[J]. Advances in Neural Information Processing Systems, 2015, 2: 2746–2754.

[18] Nagabandi A, Kahn G, Fearing R S, et al. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning[C]//IEEE. 2018 IEEE International Conference on Robotics and Automation (ICRA). New York: IEEE, 2018: 7559–7566.

[19] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. Computer Science, 2015, 8(6): A187.

[20] Li J, Monroe W, Shi T, et al. Adversarial learning for neural dialogue generation[EB/OL]. [2020–03–29]. https://arxiv.org/abs/1701.06547.

[21] Tangkaratt V, Mori S, Zhao T, et al. Model-based policy gradients with parameter-based exploration by least-squares conditional density estimation[J]. Neural Networks, 2014, 57: 128–140.

[22] Ha D, Schmidhuber J. World models[EB/OL]. [2020–03–29]. https://arxiv.org/abs/1803.10122.

[23] Doersch C. Tutorial on variational autoencoders [EB/OL]. [2020–03–29]. https://arxiv.org/abs/1606.05908.

[24] Goodfellow I. NIPS 2016 tutorial: Generative adversarial networks[EB/OL]. [2020–03–29]. https://arxiv.org/abs/1701.00160.

[25] Isola P, Zhu J, Zhou Ti, et al. Image-to-image translation with conditional adversarial networks[EB/OL]. [2020–03–29]. https://arxiv.org/abs/1611.07004.

[26] Vondrick C, Pirsiavash H, Torralba A. Generating videos with scene dynamics[EB/OL]. [2020–03–29]. https://arxiv.org/abs/1609.02612.

[27] Yu L, Zhang W, Wang J, et al. Seqgan: Sequence generative adversarial nets with policy gradient[EB/OL]. [2020–03–29]. https://arxiv.org/abs/1609.05473.

[28] Schulman J, Wolski F, Dhariwal P, et al. Proximal policy optimization algorithms[EB/OL].[2020–03–29]. https://arxiv. org/abs/1707.06347.

[29] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. Computer Science, 2015, 8(6): A187.

[30] Sehnke F, Osendorfer C, Ruckstiess T, et al. Parameter-exploring policy gradients[J]. Neural Networks, 2010, 23(4): 551–559.

[31] Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning, 1992, 8(3/4): 229–256.

[32] Kakade S. A natural policy gradient[EB/OL]. [2020–03–29].http://papers.nips.cc/paper/2073-a-natural-policy-gradient. pdf.

[33] Mirza M, Osindero S. Conditional generative adversarial nets[EB/OL]. [2020–03–29]. https://arxiv.org/abs/1411.1784.

责任编辑: 郎婧