一种基于生成对抗网络的强化学习算法

陈建平[1,2,3] 邹 锋[1,2,3] 刘 全[4] 吴宏杰[1,2,3] 胡伏原[1,2,3] 傅启明[1,2,3]

(苏州科技大学电子与信息工程学院 江苏 苏州 215009)[1]

(苏州科技大学江苏省建筑智慧节能重点实验室 江苏 苏州 215009)[2]

(苏州科技大学苏州市移动网络技术与应用重点实验室 江苏 苏州 215009)[3]

(苏州大学计算机科学与技术学院 江苏 苏州 215009)[4]

摘 要 针对强化学习方法在训练初期由于缺少经验样本所导致的学习速度慢的问题,提出了一种基于生成对抗网络的强化学习算法。在训练初期,该算法通过随机策略收集经验样本以构成真实样本池,并利用所收集的经验样本来训练生成对抗网络,然后利用生成对抗网络生成新的样本以构成虚拟样本池,再结合真实样本池和虚拟样本池来批量选择训练样本,以此来提高学习速度。同时,该算法引入了关系修正单元,结合深度神经网络,训练了真实样本池中样本的状态、动作与后续状态、奖赏之间的内部联系,结合相对熵优化生成对抗网络,提高生成样本的质量。最后,将所提出的算法与 DQN 算法应用于 OpenAI Gym 中的 CartPole 问题和 MountainCar 问题。实验结果表明,与 DQN 算法相比,所提算法可以有效地加快训练初期的学习速度,且收敛时间缩短了 15%。

关键词 强化学习,深度学习,经验样本,生成对抗网络

中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/jsjkx.180901655

Reinforcement Learning Algorithm Based on Generative Adversarial Networks

CHEN Jian-ping[1,2,3] ZOU Feng[1,2,3] LIU Quan[4] WU Hong-jie[1,2,3] HU Fu-yuan[1,2,3] FU Qi-ming[1,2,3]

(Institute of Electronics and Information Engineering, Suzhou University of Science and Technology, Suzhou, Jiangsu 215009, China)[1]

(Jiangsu Province Key Laboratory of Intelligent Building Energy Efficiency, Suzhou University of Science and Technology, Suzhou, Jiangsu 215009, China)[2]

(Suzhou Key Laboratory of Mobile Networking and Applied Technologies, Suzhou University of Science and Technology, Suzhou, Jiangsu 215009, China)[3]

(School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215009, China)[4]

Abstract With respect to the slow learning rate caused by the lack of experience samples at the early stage for most traditional reinforcement learning algorithms, this paper proposed a novel reinforcement learning algorithm based on the generative adversarial networks. At the early stage, the algorithm collects a small amount of experience samples to construct a real sample set by a stochastic policy, and utilizes the collected samples to train GAN. Then, this algorithm uses the GAN to generate samples to construct a virtual sample set. After that, by combining two sample set, this algorithm selects a batch of samples to train value function network, thus improving the learning rate to some extent. Moreover, combining a deep neural network, this algorithm introduces a new model namely rectified relationship unit to train the internal relationship between the state, action and the next state and reward, feedbacks the GAN with the relative entropy and improves the sample quality generated by GAN. Finally, this paper applied the proposed algorithm and DQN algorithm to the traditional CartPole and MountainCar problem on OpenAI Gym platform The experimental results show that the learning rate is accelerated effectively and the convergence time is cut down by 15% through the proposed method compared with DQN.

Keywords Reinforcement learning, Deep learning, Experience samples, Generative adversarial networks

---

到稿日期:2018-09-05 返修日期:2018-11-24 本文受国家自然科学基金项目(61502329,61772357,61750110519,61772355,61702055,61672371,61602334,61472267),江苏省自然科学基金项目(13KJB520020),江苏省重点研发计划项目(BE2017663),江苏省高校自然科学研究项目(13KJB520020),十三五省重点学科(20168765),航空基金(20151996016),苏州市应用基础研究计划工业部分(SYG201422)资助。

陈建平(1963-),男,教授,硕士生导师,主要研究方向为建筑节能、智能信息处理;**邹 锋**(1993-),男,硕士生,主要研究方向为强化学习、建筑节能;**刘 全**(1969-),男,教授,博士生导师,主要研究方向为强化学习、智能信息处理;**吴宏杰**(1977-),男,副教授,CCF 会员,主要研究方向为深度学习、模式识别、生物信息;**胡伏原**(1978-),男,教授,主要研究方向为图像处理、模式识别与机器学习;**傅启明**(1985-),男,博士,讲师,主要研究方向为强化学习、模式识别、建筑节能,E-mail:fqm_1@126.com(通信作者)。

1 引言

强化学习(Reinforcement Learning,RL)是一种从环境状态到行为映射的学习,并期望从环境中获得的累积奖赏最大化[1]。在学习的过程中,强化学习的 agent 选择一个动作(action),状态(state)会随之发生改变,环境对此给出一个立即奖赏(reward)作为激励信号,agent 的目标是从环境中得到最大的长期累计奖赏。根据强化学习算法在执行过程中是否需要完整的环境模型,可将强化学习算法分为基于模型的强化学习算法[2]和模型无关的强化学习算法,前者通常利用马尔科夫决策过程建模[3-4],然后利用经验样本求解最优策略,而后者则直接从经验样本中抽样求平均以逼近真实值函数,然后求解最优策略。这两类强化学习算法都需要 agent 通过不断地试错来发现最优策略,但是在训练初期,agent 并不清楚环境的工作方式,导致经验样本很少,当前值函数更新缓慢,agent 需要在与环境不断地交互中获得经验样本,因此学习速度较慢。同时,在状态数量很大的强化学习任务中,在每个时间步,值函数的计算都需要更大的计算量,训练前期经验样本的匮乏会导致无法从历史经验样本中提取更多的有用信息用来更新值函数,因此学习速度会降低,复杂度也会变得很高,并且维数灾难有时也会导致强化学习算法难以收敛。

针对强化学习算法在训练初期缺少经验样本从而导致学习速度慢的问题,Bradtke 等[5]于 1996 年提出了基于 V 值函数的最小二乘时间差分算法,该算法能够在经验样本较少的情况下,从经验样本中提取更多的有用信息来学习,通过提高经验样本的有效性来加快学习速度。但是该算法很难提取有用信息,效率比较低。因此,Hachiya 等[6]于 2009 年提出了自适应性抽样的离策略强化学习算法,该算法有效地利用了不同于当前优化策略的样本采样来更新值函数,可以有效控制逼近值函数的误差和方差,提高算法的稳定性。同时,Mahmood 等[7]于 2015 年提出了线性计算复杂度的离策略权值抽样算法,该算法在抽样的过程中会给经验样本加上权值,提高质量更好的样本被抽样到的概率,以此降低算法的复杂度,提高学习的效率。Chen 等[8]于 2018 年提出了良好选择重采样经验回放的深度强化学习算法,该算法构建了一个良好的经验回放机制,在不同的分层进行样本抽取,有效地加快了学习速度,同时避免了训练数据集的崩溃。但是,目前强化学习算法的改进大多是从经验样本的有效性和利用率出发的,并没有从生成经验样本的方面考虑,而生成对抗网络(Generative Adversarial Networks,GAN)可以用于生成经验样本。

GAN 作为一个生成式模型一经提出就引起了学术界的广泛关注,GAN 主要由生成器模型和判别器模型组成,两者通过对抗学习来训练,其基本思想是从训练库中获取很多经验样本,以学习这些经验样本的概率分布。目前,图像和视觉领域是对 GAN 研究和应用最广泛的一个领域,Ledig 等[9]于 2017 年提出了单图像超分辨率任务,即给定单张低分辨率图像,GAN 生成具有细节的高分辨率图像,避免了传统插值导致的图像模糊问题。曹志义等[10]于 2018 年提出了基于生成对抗网络的遮挡图像修复算法,该算法可直接生成并且填充可能的缺失元素,并在大量像素缺失的场景下复原出图像的本来面目。郑文博等[11]于 2018 年提出了基于贝叶斯生成对抗网络的背景消减算法,该算法利用生成对抗网络的特性,解决了图像光照渐变和突变、非静止背景以及鬼影的问题。此外,GAN 还被应用于语音和语言领域,Zhang 等[12]于 2016 年提出了利用对抗训练来生成文本的算法,该算法将生成对抗网络理论应用于文本任务,利用原始的句子和交换该句子中两个词的位置后得到的新句子进行判别训练,从而生成文本。Reed 等[13]于 2016 年提出利用 GAN 并根据文本描述来生成图像,生成器的输入为文本编码,输出为图像,实验结果证明了生成图像与文本描述之间具有较高的相关性。但是,在现有的研究中,GAN 与强化学习的结合还比较少。

本文针对强化学习算法在训练初期学习速度及收敛速度慢的问题,提出了一种基于生成对抗网络的强化学习算法(Reinforcement Learning Algorithm Based on Generative Adversarial Networks,GRL)。在训练初期,GRL 算法通过随机策略收集经验样本以构成真实样本池,并利用真实样本池中的经验样本来训练 GAN,然后利用 GAN 生成新的样本以构成虚拟样本池,最后结合真实样本池和虚拟样本池,选择训练样本提供给 agent 用于训练。同时引入关系修正单元(Rectified Relationship Unit,R-RU),结合深度神经网络训练真实样本池中样本的状态、动作与后续状态、奖赏之间的内部联系,然后利用相对熵来提高 GAN 生成样本的质量。针对收敛速度慢的问题,在 agent 不断学习的过程中,GAN 不断完善,并同时产生新的样本给 agent 用来训练,以提高收敛速度。最后将 GRL 算法应用于 CartPole 问题和 MountainCar 问题,实验结果表明,与强化学习算法相比,该算法不仅学习速度更快,收敛速度也更快。

2 相关理论

2.1 马尔科夫决策过程

马尔科夫决策过程可以用来对强化学习问题进行建模,其通常定义为一个四元组,$M=\langle S,A,R,P\rangle$。其中 S 是状态集合;A 是动作集合;R 是奖赏函数,$R(s,a)$是指在状态 s 时采取动作 a 所获得的回报值;P 是状态转移函数,$P(s,a,s')$表示在状态 s 下采用动作 a 转移到状态 s'的概率。

强化学习的最终目标是学习到能够获得最大化累积奖赏的策略,并可以利用该策略进行后续决策。策略根据其输出是一个动作还是一个动作选择的概率,通常可以分为确定策略和随机策略,其中确定策略 $\bar{h}:S\rightarrow A$ 表示在某一状态下选择某一动作,例如 $a=\bar{h}(s)$表示在状态 s 下选择动作 a;随机策略 $\tilde{h}:S\times A\rightarrow[0,1]$表示在某一状态下选择某一动作的概率,例如 $P(a|s)=\tilde{h}(s,a)$表示在状态 s 下选择动作 a 的概率。为了描述方便,后续将直接用 h 表示一个策略。假设在时刻 k,状态为 s_k,策略为 h,agent 根据当前状态 s_k 以及策略 h 选择动作 a_k,获得立即奖赏为 $R(s_k,a_k)$,状态转移至 s_{k+1}。在算法的学习过程中,不断重复该过程,agent 通过与环境的不断交互,最终获得最优策略,从而达到最大化累计奖赏的目的。

为了衡量策略 h 的优劣,在强化学习中引入状态值函数的概念,利用值函数评估策略,具体将值函数分为状态值函数

$V^h(s)$和动作值函数 $Q^h(s,a)$，其中 $V^h(s)$是在当前状态 s 下根据策略 h 所能获得的累计期望奖赏，$Q^h(s,a)$是在当前状态动作对(s,a)下根据策略 h 所能获得的累计期望奖赏。$V^h(s)$和 $Q^h(s,a)$可以认为是相应的 Bellman 公式的不动点解，表示为：

$$V^h(s)=\sum_{a\in A}h(s,a)[R(s,a)+\gamma\sum_{s'\in S}P(s,a,s')V^h(s')]$$

$$Q^h(s,a)=R(s,a)+\gamma\sum_{s'\in S}P(s,a,s')\sum_{a'\in A}h(s',a')Q^h(s',a')$$

其中，γ是折扣因子。最优策略 h^* 是指能够获得最大化累计奖赏的策略，其对应的最优值函数 $V^*(s)$和 $Q^*(s,a)$表示为：

$$V^*(s)=\max_{a\in A}\{R(s,a)+\gamma\sum_{s'\in S}P(s,a,s')V^*(s')\}$$

$$Q^*(s,a)=R(s,a)+\gamma\sum_{s'\in S}P(s,a,s')\{\max_{a'\in A}Q^*(s',a')\}$$

值得注意的是，上述两个公式也被称作最优 Bellman 公式。

2.2　生成对抗网络

生成对抗网络[14]自 2014 年被提出以来，便获得了众多研究者的关注，这些研究者先后提出了多种针对该网络的模型：条件生成对抗网络（Conditional Generative Adversarial Nets，CGAN）、拉普拉斯金字塔对抗网络（Laplacian Pyramid of Adversarial Networks，LAPGAN）、深度卷积生成对抗网络（Deep Convolutional Generative Adversarial Networks，DCGAN）、回归神经对抗网络（Recurrent Adversarial Networks，GRAN）、Wasserstein 生成对抗网络（WGAN）等[15-16]。

GAN 启发自博弈论中的二人零和博弈，GAN 模型中的博弈双方分别是生成器模型（G）和判别器模型（D）。生成器模型 G 捕捉样本数据的分布，结合服从某一分布（均匀分布、高斯分布等）的噪声 z 生成类似真实训练数据的样本，目的是学习真实的数据分布。判别器模型 D 是一个将目标样本从训练数据以及生成数据中分开的二分类器，如果样本来自于真实的训练数据，那么判别器模型 D 输出大概率，否则输出小概率。为了取得博弈的胜利，生成器模型 G 和判别器模型 D 需要不断地进行优化，分别提高相应的生成能力和判别能力，最终达到纳什均衡。

GAN 的模型结构如图 1 所示。生成器模型 G 与判别器模型 D 利用可微分函数来表示，它们各自的输入分别为随机噪声 z 和真实数据 x。$G(z)$表示由生成器模型 G 生成的尽量服从真实数据分布 P_{data} 的样本。判别器模型 D 的目标是对数据来源进行判别：如果判别器模型 D 的判别输入来自于真实数据，则标注为 1，如果输入来自生成器模型 G，则标注为 0。在不断优化的过程中，生成器模型 G 的目标是使所生成的伪数据 $G(z)$在判别器模型 D 上的标注 $D(G(z))$与真实数据 x 在判别器模型 D 上的标注 $D(x)$一致。在学习过程中，两者间的相互对抗并且迭代优化的过程将不断提高生成器模型 G 的性能，同时当判别器模型 D 的判别能力提升到无法正确判断数据来源时，可以认为生成器模型 G 已经学习到真实数据的分布。

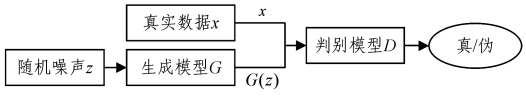


图 1　生成对抗网络

Fig. 1　Generative adversarial networks

GAN 是一个极小极大化问题，其目标函数可以描述如下：

$$\min_G\max_D V(D,G)=\mathrm{E}_{x\sim p_{d_{\text{data}}}(x)}[\log D(x)]+\mathrm{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$$

为了学习数据真实 x 的分布 p_{data}，预先定义一个输入噪音变量 $p_z(z)$，以及一个多层感知器的映射 $G(z;\theta_g)$。其中 G 是一个可微函数，参数 θ_g 代表多层感知器参数。同时，定义第二个多层感知器 $D(x;\theta_d)$，其输出为一个标量。$D(x)$表示 x 来自于真实数据而非生成数据的分布的概率，通过训练判别器模型 D，提高判别器判断样本是否来自生成器模型 G 的能力。此外，同时训练生成器模型 G，最小化 $\log(1-D(G(z)))$，以提高生成相似样本的能力。

在训练过程中：1）当生成器模型 G 固定时，判别器模型 D 的优化可以理解为，如果输入来自于真实数据，判别器模型 D 则通过优化网络参数使其输出概率为 1，反之调整网络结构使其输出概率为 0。2）当判别器模型 D 固定时，生成器模型 G 通过优化网络参数，使其输出样本尽可能与真实数据一致，并且使所生成的样本能够通过判别器模型 D 的判别。

在学习的初期，目标函数 $V(D,G)$无法提供足够的梯度给生成器模型 G 学习。当生成器模型 G 的生成能力较差时，其生成样本明显不同于真实样本，判别器模型 D 可以轻易判别输出样本的真假，此时 $\log(1-D(G(z)))$趋于 1。随着生成器模型 G 的生成能力的增强，判别器模型 D 的参数 $\log(1-D(G(z)))$会逐渐减小，$\log(D(G(z))$会逐渐增大，最终目标函数 $V(D,G)$会导致生成器模型 G 和判别器模型 D 处于同一个定点。

3　基于生成对抗网络的强化学习算法

强化算法通常需要通过一轮一轮地迭代计算值函数来获得最优策略，但是在训练初期，agent 由于缺少环境信息或者足够的经验样本，因此通常存在学习速度慢的情况。此外，在大状态空间问题中，算法的计算量将随着状态数量或者维度的增加而呈指数级增长。因此，针对训练初期强化学习算法效率低的问题，本文引入生成对抗网络，结合关系修正单元及相对熵，提出了一种基于生成对抗网络的强化学习算法，构建新的目标函数，以实现在训练初期通过生成样本，提高算法的学习速度和收敛速度。

3.1　算法原理

GAN 可以被看作是一个极小极大化的博弈游戏。判别器模型 D 试图最大化值函数，而生成器模型 G 试图最小化值函数，GAN 的具体公式如式(1)所示：

$$\min_G\max_D V(D,G)=\mathrm{E}_{x\sim p_{d_{\text{data}}}(x)}[\log D(x)]+\mathrm{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{1}$$

GAN 的目的是提高生成器模型 G 的生成能力，同时提高判别器模型 D 的判别能力，以达到纳什均衡。定义强化学习在学习过程中的真实经验样本集是以（状态 s，动作 a），（后续状态 s'，奖赏 r）成对出现的，上一时刻状态 s 对应相应的动作 a，(s,a)被称为状态动作对，下一时刻迁移至状态 s'，并获得立即奖赏 r，(s',r)被称为后续状态奖赏对。因此可以将真实经验样本集 $D_x=[s,a,s',r]$划分为两个部分：

$$D_x=[(s,a),(s',r)]=[x_1,x_2]$$

其中，x_1 表示状态动作对，x_2 表示后续状态奖赏对。由于后续状态 s' 与奖赏 r 基于上一时刻状态 s 与相对应的动作 a，因此 x_1 与 x_2 之间存在一定的联系，用互信息 I 来表示两者之间的关联：

$$\begin{aligned}I(x_2;x_1)&=H(x_2)-H(x_2|x_1)\\&=-\sum_{x_2\in X_2}p(x_2)\log_2(p(x_2))+\sum_{x_2\in X_2}\sum_{x_1\in X_1}p(x_2,x_1)\\&\quad\log_2p(x_2|x_1)\\&=\sum_{x_2\in X_2}\sum_{x_1\in X_1}p(x_2,x_1)\log_2\frac{p(x_2,x_1)}{p(x_2)p(x_1)}\end{aligned}$$

其中，$H(x_2)$ 表示 x_2 的熵，用以衡量 x_2 的不确定度；$H(x_2|x_1)$ 表示在已知 x_1 的情况下，x_2 的不确定度。$I(x_2;x_1)$ 表示由 x_1 引起的 x_2 的不确定度减少的量。由于 x_1 与 x_2 相互关联，因此互信息 I 不可能为 0。故以此构建深度神经网络 R-RU(即为关系修正单元)，R-RU 的输入为 x_1，输出为 x_2，该关系修正单元用来训练经验样本集中 x_1 与 x_2 之间的内在联系。与经验样本集一致，GAN 生成的经验样本 $G_z=[s_z,a_z,s_z',r_z]$ 也可以划分为相应的两个部分：

$$G(z)=[(s_z,a_z),(s_z',r_z)]=[G_1(z),G_2(z)]$$

其中，$G_1(z)$ 表示生成的状态动作对，$G_2(z)$ 表示生成的后续状态奖赏对。为了提高所生成样本的质量，在生成的 $G(z)$ 的基础之上，所生成的 $G_2(z)$ 与 $G_1(z)$ 之间应该符合真实样本 $[x_1,x_2]$ 中的结构关系。因此，结合生成的样本 $G_1(z)$ 以及互信息 I，将 $G_1(z)$ 输入关系修正单元 R-RU，将 R-RU 的输出作为构建的后续状态奖赏对 $G_2(z)'$，目标是使得所生成的后续状态奖赏对 $G_2(z)$ 与构建的后续状态奖赏对 $G_2(z)'$ 之间具有较高的相似性。相对熵(KL 散度)用以表示 $G_2(z)$ 与 $G_2(z)'$ 之间的相似性，其公式如下：

$$\begin{aligned}D_{KL}(P\|Q)&=\sum_i p(i)\log\frac{1}{q(i)}-\sum_i p(i)\log\frac{1}{p(i)}\\&=\sum_i p(i)\log\frac{p(i)}{q(i)}\end{aligned}\tag{2}$$

其中，P 表示生成的后续状态奖赏对 $G_2(z)$，Q 表示构建的后续状态奖赏对 $G_2(z)'$。

定理 1 生成的后续状态奖赏对 $G_2(z)$ 与构建的后续状态奖赏对 $G_2(z)'$ 之间的相对熵 $D_{KL}(P\|Q)$ 大于或等于 0，当且仅当 $P=Q$ 时，$D_{KL}(P\|Q)=0$。

证明：由吉布斯不等式可知，若 $\sum_i^n p_i=\sum_i^n q_i=1$，且 $p_i,q_i\in(0,1]$，则有：

$$-\sum_i^n p_i\log p_i\leqslant-\sum_i^n p_i\log q_i$$

当且仅当 $p_i=q_i\ \forall i$ 时，等号成立。因此吉布斯不等式等价于：

$$0\leqslant\sum_i^n p_i\log p_i-\sum_i^n p_i\log q_i=\sum_i^n p_i\log\frac{p_i}{q_i}=D_{KL}(P\|Q)$$

当 $p_i=q_i$ 时，$D_{KL}(P\|Q)=\sum_i^n p_i\log 1=0$，则 $D_{KL}(P\|Q)\geqslant 0$。证毕。

当生成的后续状态奖赏对 $G_2(z)$ 与构建的后续状态奖赏对 $G_2(z)'$ 之间的相似性非常高时，两者的相对熵趋于 0，GAN 生成的样本质量较高，因此结合式(1)，值函数 $W(D,G)$ 可表示为：

$$\begin{aligned}W(D,G)&=\min_G\max_D V(D,G)+kD_{KL}(P\|Q)\\&=E_{x\sim p_{data}(x)}[\log D(x)]+E_{z\sim p_z(z)}[\log(1-\\&\quad D(G(z)))]+kD_{KL}(P\|Q)\end{aligned}\tag{3}$$

其中，k 是权重参数。同时，可以通过 L2 正则化方法优化 GAN 模型，进一步提高样本生成的质量。当 GAN 的性能不断优化时，$G_1(z)$ 和 $G_2(z)$ 会根据真实的经验样本集不断地更新，D_{KL} 也会趋向于 0，同时，$W(D,G)$ 也会取得全局最小值。

定理 2 当 $p_g=p_{data}$ 时，$V(D,G)$ 达到局部最优，且 $W(D,G)$ 取得全局最小值 $-\log 4$。

证明：对于固定的生成器模型 G，利用判别器模型 D 最大化目标函数 $V(D,G)$：

$$\begin{aligned}V(D,G)&=\int_x p_{data}(x)\log(D(x))\mathrm{d}x+\int_z p_z(z)\log(1-\\&\quad D(G(z)))\mathrm{d}z\\&=\int_x p_{data}(x)\log(D(x))+p_g(x)\log(1-D(x))\mathrm{d}x\end{aligned}$$

对于函数 $y=a\log(x)+b\log(1-x)$，其在 $\frac{a}{a+b}$ 处取得最大值，因此，最优判别器模型 D 如式(4)所示：

$$D_G^*(x)=\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\tag{4}$$

结合式(4)，GRL 算法的目标函数可以表示为：

$$\begin{aligned}W(D,G)&=\max_D V(D,G)+kD_{KL}(P\|Q)\\&=E_{x\sim p_{data}}[\log D_G^*(x)]+E_{z\sim p_z}[\log(1-D_G^*(G\\&\quad(z)))]+kD_{KL}(P\|Q)\\&=E_{x\sim p_{data}}[\log D_G^*(x)]+E_{x\sim p_g}[\log(1-\\&\quad D_G^*(x))]+kD_{KL}(P\|Q)\\&=E_{x\sim p_{data}}[\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}]+E_{x\sim p_g}[\log\\&\quad\frac{p_g(x)}{p_{data}(x)+p_g(x)}]+kD_{KL}(P\|Q)\end{aligned}\tag{5}$$

当 $p_g=p_{data}$ 时，$D_G^*(x)=\frac{1}{2}$，因此，当 $D_G^*(x)=\frac{1}{2}$ 时，$V(D_G^*,G)=-\log 4$，将式(5)表示为：

$$\begin{aligned}W(D,G)&=E_{x\sim p_{data}}[\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}]+E_{x\sim p_g}[\log\\&\quad\frac{p_g(x)}{p_{data}(x)+p_g(x)}]+kD_{KL}(P\|Q)+V(D_G^*,\\&\quad G)-V(D_G^*,G)\\&=-\log 4+D_{KL}(p_{data}\|\frac{p_{data}+p_g}{2})+kD_{KL}(P\|\\&\quad Q)+D_{KL}(p_g\|\frac{p_{data}+p_g}{2})\\&=-\log 4+2JSD(p_{data}\|p_g)+kD_{KL}(P\|Q)\end{aligned}\tag{6}$$

当 $p_g=p_{data}$ 时，$P=G_2(z)=G_2(z)'=Q$，$JSD(p_{data}\|p_g)=0$，$D_{KL}(P\|Q)=0$，因此 $W(D,G)=-\log 4$。证毕。

当 $W(D,G)$ 取得全局最小值时，GAN 生成的经验样本与真实的经验样本具有较高的相似性，同时，生成的经验样本 $G(z)$ 内部生成的状态动作对 $G_1(z)$，与生成的后续状态奖赏

对 $G_2(z)$之间也具有较高的关联性。

GRL 算法的网络结构如图 2 所示，其中包含 3 个主要部分：1)GAN，用于生成经验样本；2)真实样本池 D_1 与虚拟样本池 D_2，D_1 用于收集 agent 与环境交互过程中获得的真实经验样本，D_2 用于收集 GAN 生成的样本，两者共同为 agent 提供训练样本；3)R-RU，表示所构建的深度神经网络，被称为关系修正单元，用来训练 D_1 中状态动作对$[s,a]$与后续状态奖赏对$[s',r]$之间的关系，指导生成器模型 G 提升性能。生成器模型 G 输入噪声 z，生成样本 $G_z=[s,a,s',r]$，判别器模型 D 判断生成样本的真假，并将判别信息反馈给生成器模型 G，优化生成器模型 G，以提高所生成样本的质量。此外，R-RU 分析 D_1 中$[s,a]$与$[s',r]$之间的内在联系，将生成器模型 G 生成的状态动作对$[s,a]$输入 R-RU，输出符合内在联系的后续状态奖赏对$[s'',r']$，将其与生成器模型 G 生成的后续状态奖赏对$[s',r]$做对比，进一步促进生成器模型 G 生成更加真实的样本。

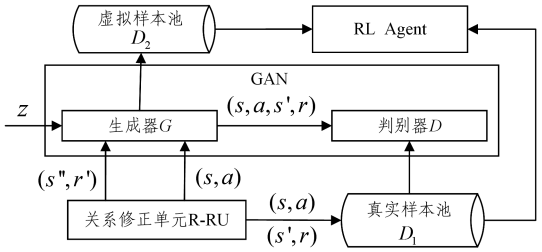


图 2　GRL 算法的网络结构

Fig. 2　Network structure of GRL algorithm

3.2　GRL 算法的流程及复杂度分析

GRL 算法的执行流程如图 3 所示，在训练初期，其收集经验样本，并将这些样本加入真实样本池 D_1，利用 D_1 中的真实经验样本训练 GAN，同时，R-RU 也会不断改进 GAN，从而提高 GAN 所生成的样本的质量；然后利用 GAN 生成新的经验样本，将其加入虚拟样本池 D_2 中，结合 D_1 和 D_2 选择样本数据，并将该数据提供给 agent 用于训练动作值函数网络，从而寻找最优策略。此外，在 agent 训练的同时，该算法还会继续生成新的经验样本，并将其加入到虚拟样本池 D_2 中，以加快 agent 的学习速度与收敛速度。GRL 算法的具体步骤如算法 1 所示。从复杂度方面考虑，GRL 算法的 GAN 与 R-RU 两个部分虽然提高了算法的空间复杂度，但是在算法的运行过程中，R-RU 会不断改善 GAN 生成样本的质量并将其加入 D_2，且 GAN，R-RU 与 agent 处于并行的位置，因此不会大幅增加算法的时间复杂度，但可以加快学习速度与收敛速度。

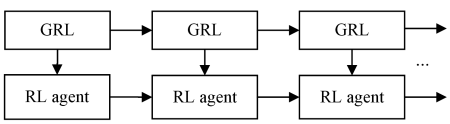


图 3　算法的执行流程

Fig. 3　Execution flow of algorithm

算法 1　基于生成对抗网络的强化学习算法

1. 初始化 GAN 和 R 的权值、θ、D_1 和 D_2 的长度 N，K 值
2. 令 $\theta^-=\theta$
3. k=0
4. For each episode
5. 　收集经验样本 $D_1(s,a,s',r)$
6. 　If k%K=0
7. 　　利用 $D_1(s,a,s',r)$训练 GAN 和 R-RU
8. 　　利用 R-RU 改进 GAN
9. 　end if
10. 　GAN 生成新的样本并加入到 D_2 中
11. 　从 D_1 和 D_2 中均匀随机地抽取 mini-match 个样本
12. 　For each sample
13. 　　$y=r+\gamma \max_{a'} Q(s',a';\theta^-)$
14. 　　$\Delta\theta=\Delta\theta+\alpha[y-Q(s,a;\theta)]\nabla Q(s,a;\theta)$
15. 　　$\theta=\theta+\Delta\theta$
16. 　　$\theta^-=\theta$
17. 　　k=k+1
18. 　End for
19. End for

4　实验结果分析

为了验证算法的有效性及收敛性能，将所提出的 GRL 算法、DQN 算法[17-18] 应用于 CartPole 问题和 MountainCar 问题。

4.1　CartPole 实验

4.1.1　实验描述

如图 4 所示，在 CartPole 环境中有一辆小车处于无阻力的轨道上，车上绑着一个连接不牢固的杆，通过在小车上施加一个向左或者向右的力来调整小车的运动方向与速度，以使杆不倒下来。如果杆从垂直角度倒下一定角度，或者小车到达轨道的末端即算失败。每次失败后，都要将杆放置在垂直位置。该任务可以看作一个情节式任务，通过不断尝试保持杆的平衡。

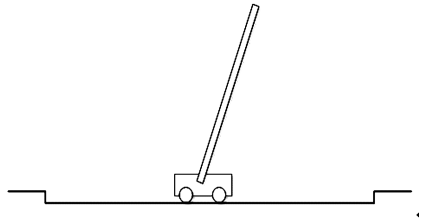


图 4　平衡杆

Fig. 4　CartPole

4.1.2　实验设置

本实验环境基于 OpenAI Gym，在实验初期，GRL 算法通过 ε-greedy 随机策略收集部分经验样本数据$[s,a,s',r]$，并将其加入真实样本池，其 GAN 部分以此生成新的样本，并将新样本加入虚拟样本池，两个样本池共同提供经验样本用于训练，探索因子 $\varepsilon=0.1$。

在每个情节中，若杆在每个时间步都没有失败，奖赏值为 +1，在其他情况下，奖赏值为 −100。每个情节的最大平均回报为 500，即当连续 10 个情节的平均回报大于 500 时，实验结束。在实验过程中，虚拟样本池与真实样本池的长度均为 5000，每次随机从两个样本池中抽取 32 个样本来训练深度神经网络以逼近动作值函数。折扣因子 $\gamma=0.99$，学习率 $\alpha=0.001$。

将 GRL 算法、DQN 算法在相同的实验环境下独立重复 10 次实验，取实验结果的平均值来比较两种算法的性能。

4.1.3 实验分析

图 5 主要用于比较 GRL 算法、DQN 算法在 CartPole 问题上的性能(在实验过程中，当实验进行到第 10 个情节时，开始生成样本，且每个算法都被独立执行 10 次)。横坐标是情节数，纵坐标是算法在不同情节下的累计奖赏(其中累计奖赏值是独立执行算法 10 次的平均值)。从图 5 中可以看出，GRL 算法的学习速度要比 DQN 算法快，GRL 算法在 100 个情节左右时已经收敛到最优，而 DQN 算法在 140 个情节左右收敛，收敛时间分别为 134 s 与 165 s。这主要是因为 GRL 算法在训练初期，结合了真实样本池收集的经验样本，GAN 以此生成了新的样本并将其加入虚拟样本池，两个样本池同时为 agent 提供训练样本，agent 通过训练深度神经网络来逼近动作值函数，从而指导 agent 在每个状态选择更准确的动作，使得获得的累积奖赏更大，因此学习速度和收敛速度较快。在训练初期，GAN 生成的样本质量较差，获得的累积奖赏较少，但是，随着情节数的增加，生成的样本质量变好，学习速度也会加快。同时，在训练中期，深度神经网络参数的微小变化可能会引起策略选择的巨大改变，因此累积奖赏波动较大，但并不影响 GRL 算法最后的收敛性。DQN 算法由于在训练初期缺少足够的经验样本，容易导致动作值函数更新缓慢，因此学习、收敛速度较慢。综上所示，GRL 算法能够加快学习速度，提高收敛性。

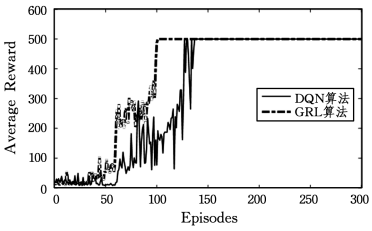
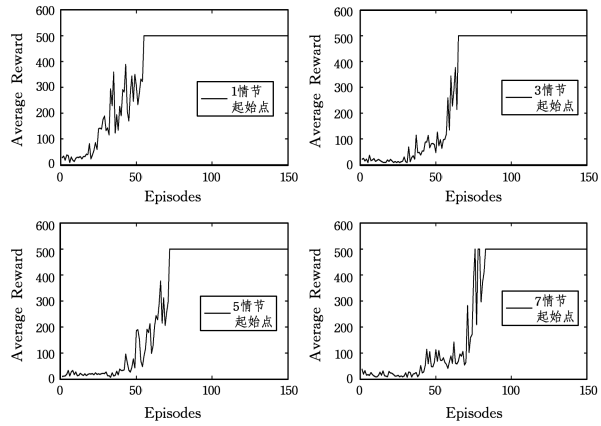


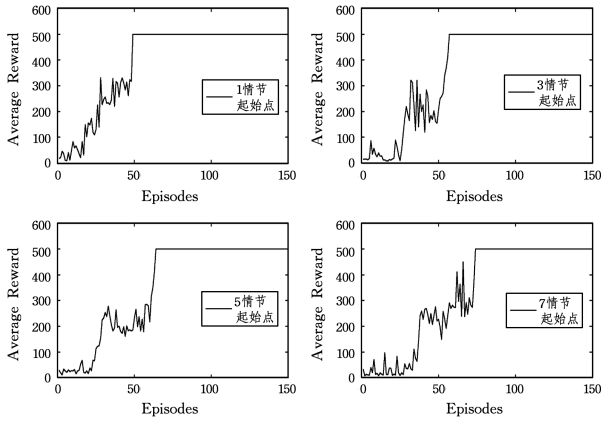
图 5 GRL 算法与 DQN 算法的性能比较

Fig. 5 Performance comparison of GRL algorithm and DQN algorithm

图 6 是在不同的情节起始点及不同的学习率下 GRL 算法的性能比较图。其中，图 6(a)和图 6(b)分别是在学习率为 0.001 和 0.01 的情况下，当实验中的情节个数达到 1，3，5，7 时，开始生成样本的 GRL 算法的性能图。图中横坐标是情节个数，纵坐标是算法执行 10 次每个情节到达终止状态后的平均奖赏值。从图 6(a)可以看出，在 4 个不同的情节起始点下，GRL 算法都具有较好的收敛性，分别在第 57，63，73，85 个情节左右收敛，收敛时间分别为 46 s，51 s，57 s 与 69 s。算法在 1 个情节起始点时的收敛速度相比于在 3 个与 5 个情节起始点时，分别提升了 10% 与 20%。在 1 个情节起始点时，算法在训练初期的学习速度较快，获得的平均奖赏波动较大，在其他情节起始点时，算法在训练初期的学习速度慢，获得的累计奖赏较少。这是因为在训练过程中，在虚拟样本池中加入 GAN 生成的样本的时刻越早，agent 就能获得越多的经验样本来训练神经网络，以逼近动作值函数，从而寻找最优策略，所以 1 个情节起始点时，算法在训练初期的学习速度较快。同时，虚拟样本池中的经验样本也会导致训练初期累计奖赏波动较大，但是并不影响算法的收敛性。因此，GRL 算法能在保证收敛的情况下，有较快的学习速度和收敛速度，并且其稳定性在后续情节中也较好。



(a)$\alpha=0.001$



(b)$\alpha=0.01$

图 6 不同情节起始点的 GRL 算法的性能比较

Fig. 6 Performance comparison of GRL algorithm with different episode starting points

与图 6(a)相比，图 6(b)将学习率设为 0.01，以比较不同情节起始点下，GRL 算法的性能。从图 6(b)可以看出，相比于图 6(a)，4 个不同的情节起始点下，GRL 算法在训练过程中的平均奖赏波动较大。算法在 1 个情节起始点时的收敛性能明显高于在 3 个、5 个和 7 个情节起始点时的收敛性能，并在第 50 个情节左右进入收敛状态，并且该算法在整个执行过程中较稳定；然而在 3 个、5 个和 7 个情节起始点时，GRL 算法分别在第 57 个、第 65 个和第 73 个情节左右才趋于收敛，4 个不同的情节起始点时，算法的收敛时间为 41 s，46 s，52 s 与 59 s。这是因为在生成样本的起始点相同的情况下，提高学习率会加快动作值函数的更新速度，指导 agent 在每个状态选择更为准确的动作，学习速度也更快，但学习率的增大也会导致参数的更新幅度增大，进而影响动作选择的概率，因此训练初期平均奖赏的波动较大。此外，将图 6(b)与图 6(a)相比可以发现，学习率的增大会减弱样本数量对于前期平均奖赏波动的影响，并且过大的学习率可能会导致算法无法收敛。综

上所述，在不同的情节起始点下，且在振荡合理的范围内，GRL 算法的学习率越大，学习速度越快且收敛性能良好。

4.2 MountainCar 实验

4.2.1 实验描述

如图 7 所示，在 MountainCar 环境中有一辆小车处于带有坡度的路面的谷底，由于动力不足，小车无法直接加速冲上坡顶，即图 7 中最右侧的由五角形标记的位置，小车需要经过前后加速并借助惯性才能到达坡顶。其中，状态由位置和速度两个维度组成，用 p 和 v 表示，则状态可以表示为 $s=[p, v]$。在任意时刻，小车都有 3 个可选动作，即向右加速、向左加速和不加速，分别用 $+1$，-1，0 表示，即动作 $a=\{+1, -1, 0\}$。

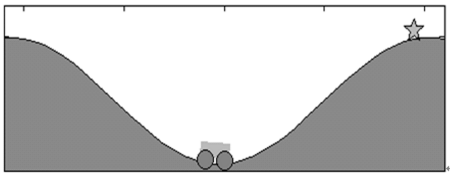


图 7　山地车

Fig. 7　MountainCar

4.2.2 实验设置

本实验基于 OpenAI Gym。在实验初期，GRL 算法会通过 ε-greedy 随机策略收集部分经验样本数据$[s, a, s', r]$加入真实样本池，GAN 以此生成新的样本经验并将其加入虚拟样本池，两个样本池共同为 agent 提供样本用来训练，探索因子 $\varepsilon=0.1$。

在每个情节中，若小车冲上坡顶，或者超过 1000 个时间步还未冲上坡顶，则实验结束。在实验过程中，真实样本池与虚拟样本池的长度均为 5000，每次随机从两个样本池中抽取 32 个样本来训练深度神经网络以逼近动作值函数。折扣因子 $\gamma=0.99$，学习率 $\alpha=0.001$。

将 GRL 算法、DQN 算法在相同的实验环境下独立重复 10 次实验，取实验结果的平均值来进行各算法的性能比较。

4.2.3 实验分析

图 8 主要用于比较 GRL 算法与 DQN 算法在 MountainCar 问题上的性能（在实验过程中，当实验进行到第 5 个情节时，开始生成样本，且每个算法都被独立执行 10 次）。横坐标是情节数，纵坐标是算法执行 10 次的平均步数。从图 8 中可以看出，GRL 算法和 DQN 算法最终都能收敛到 185 步左右，但是 GRL 算法在第 60 个情节左右时就已经收敛，而 DQN 算法在第 68 个情节左右才趋于收敛，收敛时间分别为 78 s 与 89 s。GRL 算法在训练过程中的学习速度较快，且平均步数的波动性比 DQN 算法小。这主要是因为 GRL 算法在训练初期结合了真实样本池中的经验样本，其 GAN 部分以此生成了新的样本，并将其加入了虚拟样本池，两个样本池共同提供经验样本给 agent 来训练深度神经网络，从而逼近动作值函数，所以学习速度较快。DQN 算法在训练初期由于缺少经验样本，需要通过不断地试错来获得经验样本，然后更新动作值函数，因此学习速度相对较慢。综上所述，相比于 DQN 算法，GRL 算法提高了学习速度以及收敛速度。

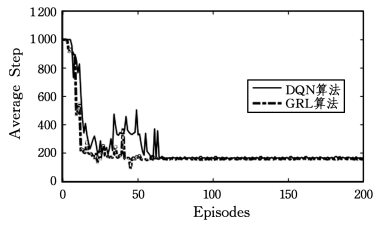


图 8　GRL 算法与 DQN 算法的性能比较

Fig. 8　Performance of GRL algorithm and DQN algorithm

图 9 是在学习率为 0.001 的情况下，不同的情节起始点时，GRL 算法的性能比较图，横坐标是情节个数，纵坐标是算法执行 10 次每个情节到达终止状态后的平均步数。在图 9 中，当实验达到 1 个、3 个、5 个和 7 个情节时，开始生成样本，并将生成的样本加入到虚拟样本池中。从图 9 可以看出，在训练初期，4 个不同的情节起始点下，GRL 算法最终都能收敛，其中 1 个情节起始点下，GRL 算法收敛得最快，在第 40 个情节左右就已收敛，而 3 个、5 个和 7 个情节起始点下，GRL 算法分别在第 47 个、第 52 个和第 57 个情节处收敛，并且在收敛过程中，平均时间步的波动性小；4 个不同的情节起始点下，算法的收敛时间分别为 52s，61s，68s 与 75s。这是因为在训练过程中，在虚拟样本池中加入 GAN 生成的样本的时刻越早，agent 就能获得越多的经验样本来训练神经网络，从而逼近动作值函数，以此寻找最优策略，而 DQN 算法在训练初期缺少足够的经验样本，需要通过不断的试错来获得经验样本。因此，结合图 8，相比于 DQN 算法，GRL 算法具有较快的学习速度及收敛速率，性能较好。

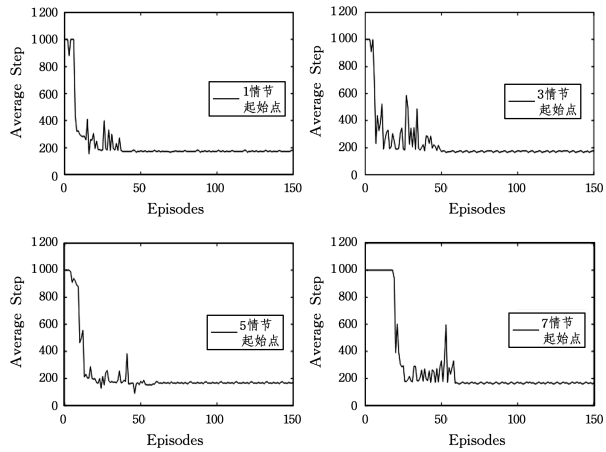


图 9　不同情节起始点的 GRL 算法的性能比较

Fig. 9　Performance comparison of GRL algorithm with different episode starting points

结束语　本文主要针对强化学习算法在训练初期缺少足够的经验样本，导致学习速度慢的问题，提出了一种基于生成对抗网络的强化学习算法。该算法利用 GAN 生成经验样本，同时引入关系修正单元，反向修正 GAN 网络参数，以提高 GAN 生成的经验样本的质量。实验基于 OpenAIGym 平台，从算法性能的角度进行了比较，实验结果表明，基于生成对抗网络的强化学习算法加快了训练初期的学习速度和收敛速度。

本文主要利用 OpenAI Gym 实验平台对算法做了相关分析，从结果可以看出，所提算法在训练初期具有较快的学习速度与收敛速度。CartPole 和 MountainCar 是两种小规模的连续状态空间问题，接下来我们考虑将该算法应用于大规模的连续空间的实际问题，以进一步检验算法的性能。同时，如何进一步改善 GAN 的性能来生成更加真实的经验样本也是下一步工作的研究方向。

参 考 文 献

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

[2] PUTERMAN M. Markov decision process [J]. Statistica Neerlandica, 1985, 39(2): 219-233.

[3] WU Y, SHEN T. Policy Iteration algorithm for optimal control of stochastic logical dynamical systems [J]. IEEE Transactions on Neural Networks & Learning Systems, 2017, 28(99): 1-6.

[4] WEI Q, LIU D, LIN H. Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear systems [J]. IEEE Transactions on Cybernetics, 2016, 46(3): 840-853.

[5] BRADTKE S J, BARTO A G. Linear least-squares algorithms for temporal difference learning [J]. Machine Learning, 1996, 22(1/2/3): 33-57.

[6] HACHIYA H, AKIYAMA T, SUGIAYMA M, et al. Adaptive importance sampling for value function approximation in off-policy reinforcement learning [J]. Neural Networks, 2009, 22(10): 1399-1410.

[7] MAHMOOD A R, SUTTON R S. Off-policy learning based on weighted importance sampling with linear computational complexity[C]// Proceedings of the 31st International Conference on Uncertainty in Artificial Intelligence. Amsterdam: AUAI, 2015: 552-561.

[8] CHEN X L, CAO L, LI C X, et al. Deep reinforcement learning via good choice resampling experience replay memory [J]. Control and Decision, 2018, 33(4): 129-134.

[9] LEDIG C, THEIS L, HUSZÁR F, et al. Photo-realistic single image super-resolution using a generative adversarial network [C]// Proceedings of the 30th IEEE Conference on ComputerVision and Pattern Recognition. Hawaii: IEEE, 2017: 105-114.

[10] CAO Z Y, NIU S Z, ZHANG J W. Masked image inpainting algorithm based on generative adversarial networks [J]. Journal of Beijing University of Posts and Telecom, 2018, 41(3): 81-86. (in Chinese)
曹志义，牛少彰，张继威. 基于生成对抗网络的遮挡图像修复算法[J]. 北京邮电大学学报，2018，41(3)：81-86.

[11] ZHENG W B, WANG K F, WANG F Y. Background subtraction algorithm with bayesian generative adversarial networks [J]. Acta Automatica Sinica, 2018, 44(5): 878-890. (in Chinese)
郑文博，王坤峰，王飞跃. 基于贝叶斯生成对抗网络的背景消减算法[J]. 自动化学报，2018，44(5)：878-890.

[12] ZHANG Y Z, GAN Z, CARIN L. Generating text via adversarial training[C]// Proceedings of the 30th Conference on Neural Information Processing Systems. Barcelona: MIT Press, 2016: 1543-1551.

[13] REED S, AKATA Z, YAN X C, et al. Generative adversarial text to image synthesis[C]// Proceedings of the 33rd International Conference on Machine Learning. New York: ACM, 2016: 1060-1069.

[14] WANG K F, GOU C, DUAN Y J, et al. Generative adversarial networks: the state of the art and beyond[J]. Acta Automatica Sinica, 2017, 43(3): 321-332. (in Chinese)
王坤峰，苟超，段艳杰，等. 生成式对抗网络 GAN 的研究进展与展望[J]. 自动化学报，2017，43(3)：321-332.

[15] ARJVSKY M, CHINTALA S, BOTTOU L. Wasserstein generative adversarial networks[C]// Proceedings of the 34th International Conference on Machine Learning. Sydney: ACM, 2017: 214-223.

[16] MIRZA M, OSINDERO S. Conditional generative adversarial nets [J]. Computer Science, 2014, 8(13): 2672-2680.

[17] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

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