# 增强型深度确定策略梯度算法

陈建平[1,2,3,4]，何超[1,2,3]，刘全[5]，吴宏杰[1,2,3,4]，胡伏原[1,2,3,4]，傅启明[1,2,3,4]

（1. 苏州科技大学电子与信息工程学院，江苏 苏州 215009；2. 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；
3. 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；
4. 苏州科技大学苏州市虚拟现实智能交互及应用技术重点实验室，江苏 苏州 215009；5. 苏州大学计算机科学与技术学院，江苏 苏州 215006）

**摘 要**：针对深度确定策略梯度算法收敛速率较慢的问题，提出了一种增强型深度确定策略梯度（E-DDPG）算法。该算法在深度确定策略梯度算法的基础上，重新构建两个新的样本池——多样性样本池和高误差样本池。在算法执行过程中，训练样本分别从多样性样本池和高误差样本池按比例选取，以兼顾样本多样性以及样本价值信息，提高样本的利用效率和算法的收敛性能。此外，进一步从理论上证明了利用自模拟度量方法对样本进行相似性度量的合理性，建立值函数与样本相似性之间的关系。将 E-DDPG 算法以及 DDPG 算法用于经典的 Pendulum 问题和 MountainCar 问题，实验结果表明，E-DDPG 具有更好的收敛稳定性，同时具有更快的收敛速率。

**关键词**：深度强化学习；样本排序；自模拟度量；时间差分误差

**中图分类号**：TP391

**文献标识码**：A

**doi:** 10.11959/j.issn.1000−436x.2018238

# Enhanced deep deterministic policy gradient algorithm

CHEN Jianping[1,2,3,4], HE Chao[1,2,3], LIU Quan[5], WU Hongjie[1,2,3,4], HU Fuyuan[1,2,3,4], FU Qiming[1,2,3,4]

1. Institute of Electronics and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China
2. Jiangsu Province Key Laboratory of Intelligent Building Energy Efficiency, Suzhou University of Science and Technology, Suzhou 215009, China
3. Suzhou Key Laboratory of Mobile Networking and Applied Technologies, Suzhou University of Science and Technology, Suzhou 215009, China
4. Virtual Reality Key Laboratory of Intelligent Interaction and Application Technology of Suzhou, Suzhou University of Science and Technology, Suzhou 215009, China
5. School of Computer Science and Technology, Soochow University, Suzhou 215006, China

**Abstract:** With the problem of slow convergence for deep deterministic policy gradient algorithm, an enhanced deep deterministic policy gradient algorithm was proposed. Based on the deep deterministic policy gradient algorithm, two sample pools were constructed, and the time difference error was introduced. The priority samples were added when the experience was played back. When the samples were trained, the samples were selected from two sample pools respectively. At the same time, the bisimulation metric was introduced to ensure the diversity of the selected samples and improve the convergence rate of the algorithm. The E-DDPG algorithm was used to pendulum problem. The experimental results show that the E-DDPG algorithm can effectively improve the convergence performance of the continuous action space problems and have better stability.

**Key words:** deep reinforcement learning, sample ranking, bisimulation metric, temporal difference error

---

**收稿日期**：2018−03−22；**修回日期**：2018−08−01

**通信作者**：傅启明，fqm_1@126.com

**基金项目**：国家自然科学基金资助项目（No.61502329, No.61772357, No.61750110519, No.61772355, No.61702055, No.61672371, No.61602334, No.61502323）；江苏省自然科学基金资助项目（No.BK20140283）； 江苏省重点研发计划基金资助项目（No.BE2017663）；江苏省高校自然科学研究基金资助项目（No.13KJB520020）；苏州市应用基础研究计划工业部分基金资助项目（No.SYG201422）

**Foundation Items:** The National Natural Science Foundation of China (No.61502329, No.61772357, No.61750110519, No.61772355, No.61702055, No.61672371, No.61602334, No.61502323), The Natural Science Foundation of Jiangsu Province (No.BK20140283), The Key Research and Development Program of Jiangsu Province (No.BE2017663), High School Natural Foundation of Jiangsu Province (No.13KJB520020), Suzhou Industrial Application of Basic Research Program Part (No.SYG201422)

## 1　引言

强化学习的基本思想是通过最大化智能体（agent）从环境中获得的累计奖赏值，以学习完成目标的最优策略[1]。依据策略表示方法和求解的不同，可以将强化学习方法分为 3 类：“评论家”算法，该算法利用值函数对策略进行评估，最终利用最优值函数求解最优策略；“行动者”算法，该算法利用类似启发式搜索的方法从策略空间中找出最优策略；“行动者—评论家”算法，行动者部分用于动作的选取，评论家部分用于评估动作的好坏，利用值函数信息指导策略的搜索[2]。然而对于上述任意一类算法，在学习过程中，都需要人工设定状态表示方法，而通过深度学习方法，可以实现状态特征的自动学习，以实现“端到端”的任务学习。目前，深度学习作为在机器学习领域的一个研究热点，已经在图像分析、语音识别、视频分类、自然语言处理等领域获得令人瞩目的成就。深度学习的基本思想是通过多层的网络结构和非线性变换，组合低层特征，形成抽象的、易于区分的高层表示，以发现数据的分布式特征表示[3]。深度学习模型通常由多层的非线性运算单元组合而成，将较低层的输出作为更高一层的输入，通过这种方式自动地从大量训练数据中学习抽象的特征表示[4-5]。

谷歌的 DeepMind 团队将深度学习和强化学习结合起来，提出深度强化学习方法，并将深度强化学习应用于围棋问题。2016 年，Alpha Go[6] 在人机围棋比赛中以 4:1 战胜围棋大师李世石，而新版的 Alpha Zero[7]可以不需要任何历史棋谱知识，不借助任何人类先验知识，仅利用深度强化学习进行自我对弈，最终能以 100:0 的战绩完胜 Alpha Go。目前，深度强化学习已经成为人工智能领域的研究热点。Mnih 等[8-9]将卷积神经网络与传统的 Q 学习[10]算法相结合，提出了深度 Q 网络（DQN, deep Q-network）模型。DQN 将未被处理过的像素点（原始图像）作为输入，通过样本池存储历史经验样本，同时利用经验回放打破样本间的联系，以避免网络参数的震荡。但是 DQN 只能解决离散的、低维的动作空间问题，将 DQN 应用到连续动作领域最简单的做法是将连续动作离散化，但是这会导致离散动作的数量随动作维度的增加而呈指数型增长，同时对连续动作进行简单的离散化会忽略动作域的结构，然而在很多情况下，动作域的结构对于问题的求解是非常重要的，因此，目前基于 DQN 算法提出了很多关于 DQN 的变体。Hasselt 等[11]在双重 Q 学习算法[12]的基础上提出了深度双重 Q 网络（DDQN, deep double Q-network）算法。Schaul 等[13]在 DDQN 的基础上提出了一种基于比例优先级采样的深度双 Q 网络（double deep Q-network with proportional prioritization）等。然而，这些改进的算法都不能够很好地解决连续动作空间问题。在连续动作空间中，策略梯度是常用的方法，它通过不断计算策略期望总奖赏关于策略参数的梯度来更新策略参数，最终收敛于最优策略[14]。因此，在解决深度强化学习问题时，可以采用深度神经网络表示策略，并利用策略梯度方法求解最优参数。此外，在求解深度强化学习问题时，基于策略梯度的算法能够直接优化策略的期望总奖赏，并以端对端的方式直接在策略空间中搜索最优策略。因此，与 DQN 及其改进算法相比，基于策略梯度的深度强化学习方法适用范围更广，策略优化的效果也更好。Lillicrap 等[15]将 DPG（deterministic policy gradient）算法[16]与 DQN 相结合，提出了 DDPG（deep deterministic policy gradient）算法。DDPG 可用于解决连续动作空间的强化学习问题。实验表明，DDPG 不但在一系列连续动作空间的任务中表现稳定，而且求得最优解所需要的时间步也远低于 DQN，但是 DDPG 需要大量的样本数据，且算法的收敛速度也有待提高。

本文在 DDPG 算法的基础上提出了增强型深度确定策略梯度（E-DDPG, enhanced deep deterministic policy gradient）算法。针对 DDPG 算法收敛速度慢的问题，E-DDPG 算法在原始样本池的基础上构建了两个样本池——高误差样本池和多样性样本池。高误差样本池将 TD（temporal-difference）error 作为启发式信息对样本进行排序，以提高误差较大的样本的选取概率。同时，多样性样本池利用自模拟度量方法度量样本间的距离，在原始样本池的基础上，选择低相似样本，以提高样本池中样本的多样性，提高算法的执行效率。在算法学习过程中，训练样本将分别从高误差样本池和多样性样本池按比例选取，以兼顾样本多样性以及样本价值信息，提高样本的利用效率和算法的收敛性能。实验结果表明，与 DDPG 算法相比，E-DDPG 算法具有更快的收敛速度以及更好的收敛稳定性。

## 2 相关理论

### 2.1 马尔可夫决策过程

强化学习问题通常可以建模成一个马尔可夫决策过程（MDP, Markov decision process）。一个 MDP 可以表示为一个五元组，$M =< S, A, P, r, \gamma >$，其中，$S$ 是状态空间；$A$ 是动作空间；$P$ 是状态转移函数；$P(s_{t+1} | s_t, a_t)$ 表示在状态 $s_t \in S$，采取动作 $a_t \in A$，转移到下一个状态 $s_{t+1} \in S$ 的概率；$R$ 是奖赏函数，$r(s_t, a_t)$ 表示在状态 $s_t$ 采取动作 $a_t$ 所获得的立即奖赏；$\gamma \in$（0，1）是折扣因子。对于一个 MDP 问题，在每一个时间步 $t$，agent 根据环境状态 $s_t$，采取一个动作 $a_t$ 并获取立即奖赏 $r_t$，然后获取到下一个状态 $s_{t+1}$。

强化学习算法的目的是求解一个最优策略，并利用这个策略进行决策。在强化学习中，$\pi$ 表示策略，$\pi(s,a)$ 表示在状态 $s$ 下选择动作 $a$ 的概率。如果策略 $\pi$ 是一个确定的策略，在任意状态 $s \in S$，$\pi(s)$ 表示在状态 $s$ 下所选择的动作 $a$。

强化学习中利用值函数来评估策略 $\pi$ 的好坏，具体可分为状态值函数 $V^\pi$ 和动作值函数 $Q^\pi$，其中，$V^\pi(s)$ 表示在状态 $s$ 由策略 $\pi$ 得到的累计期望奖赏，$Q^\pi(s)$ 表示状态动作对 $(s,a)$ 由策略 $\pi$ 得到的累计期望奖赏。通常采用动作值函数来评估策略 $\pi$ 的好坏，即

$$Q^\pi(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s' | s,a) \sum_{a' \in A} \pi(s',a') Q^\pi(s',a') \tag{1}$$

式(1)也被称作 Bellman 公式。

强化学习是为了获得最大化累计奖赏的最优策略 $\pi^*$，对应的 $Q^*(s,a)$ 可以表示为

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s' \in S} p(s' | s,a) \{ \max_{a' \in A} Q^*(s',a') \} \tag{2}$$

式(2)被称作最优 Bellman 公式。

### 2.2 深度确定策略梯度算法

DDPG 算法以行动者评论家算法为框架，同时结合 DPG 算法以及 DQN 算法。DPG 算法[16]利用近似函数 $\mu(s | \theta^\mu)$ 表示策略，其策略梯度可以表示为

$$\nabla_\theta J(\mu_\theta) = \int_S \rho^\mu(s) \nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s,a) |_{a=\mu_\theta(s)} \mathrm{d}s \tag{3}$$

在随机策略中，策略梯度取决于状态和动作，而在确定策略中，策略梯度仅取决于状态。因此，与随机策略梯度算法相比，确定策略梯度算法收敛需要的样本相对较少。

DDPG 算法在行动者部分引入确定策略梯度，利用式(3)更新策略网络参数，而在评论家部分利用式(4)对网络参数进行更新。但是，如果直接使用式(4)对评论家网络进行更新会导致网络震荡，因为在更新 $Q(s,a | \theta^Q)$ 的过程中，同时也会计算相应的目标值，即式(5)中的 $y_t$。

$$L(\theta^Q) = E_{s_t \sim \rho^\pi,\ a_t \sim \pi, r_t \sim E} [(Q(s_t, a_t | \theta^Q) - y_t)^2] \tag{4}$$

其中，有

$$y_t = r(s_t, a_t) + \gamma Q(s_{t+1}, \mu(s_{t+1}) | \theta^Q) \tag{5}$$

针对这个问题，DDPG 采用 DQN 中的目标网络，但是并非直接复制权重参数到目标网络中，而是采用“soft”方式更新目标网络中的参数，即创建新的评论家和行动者网络（$Q'(s,a | \theta^{Q'})$ 和 $\mu'(s | \theta^{\mu'})$），来更新目标参数。目标策略网络和目标值网络中参数的更新规则为 $\theta^{\mu'} \leftarrow \alpha\theta + (1-\alpha)\theta^{\mu'}$，$\theta^{Q'} \leftarrow \alpha\theta^Q + (1-\alpha)\theta^{Q'} (\alpha << 1)$，该更新方法可以有效地限制目标值的更新速度，极大地提升了算法的稳定性。此外，通过引入 DQN 中使用的经验回放机制打破样本的关联性，来提高参数更新的有效性。

值得注意的是，确定策略梯度算法缺少对环境的探索，而 DDPG 算法通过引入随机噪声来完成策略探索。通过添加随机噪声 $N$，使动作的选择具有一定的随机性，以完成一定程度的策略探索，具体如式(6)所示。

$$\mu'(s_t) = \mu(s_t | \theta_t^\mu) + N \tag{6}$$

### 2.3 自模拟度量与状态之间的距离

为了度量 MDP 中状态的关系，自模拟关系被 Givan 等[17]引入 MDP 中。简而言之，如果两个状态满足自模拟关系，那么这两个状态就共享相同的最优值函数以及最优动作。

**定义 1** 自模拟（bisumulation）关系。若关系 $E \subseteq S \times S$ 是自模拟关系，则对于 $s', s'' \in S$，$s'Es''$ 满足以下性质。

①对于 $\forall a \in A, r(s',a) = r(s'',a)$

②对于 $\forall a \in A, \forall C \in \frac{S}{E}, \sum_{t \in C} P(t | s',a) = \sum_{t \in C} P(t | s'',a)$

其中，状态集合 $S$ 关于 $E$ 的等价集合用 $\frac{S}{E}$ 来表示。若两个状态（$s', s'' \in S$）满足自模拟关系，可记作 $s' \sim s''$。

从定义 1 可以得出，任意两个状态要么满足自模拟关系，要么不满足自模拟关系。这种度量方法过于苛刻，且限制其使用的范围。Ferns 等[18]提出了一种可用于衡量两个状态之间远近关系的自模拟度量方法（bisimulation metric）。

**定理 1**　$D$ 为定义在状态集 $S$ 上的度量集合，且度量 $d \in D$。对于 $\forall s', s'' \in S$，定义 $G: D \to D, G(d)(s', s'') = \max_{a \in A}(d_a(s', s'') + \gamma T_K(d)(P(s', a, \bullet), P(s'', a, \bullet)))$。其中，$d_a(s', s'') = |r(s', a) - r(s'', a)|$，$0 < \gamma < 1$，则 $G$ 存在一个最小不动点 $d_{\sim}$，$d_{\sim}$ 是一个自模拟度量，$d_{\sim}(s', s'')$ 是状态 $s'$ 和 $s''$ 之间的距离。

计算两个状态距离的算法如算法 1 所示。

**算法 1**　状态之间距离度量算法

**输入**　状态 $s_1$ 和状态 $s_2$

**输出**　$d_{\sim}(s_1, s_2)$

1) 初始化：$d(s_1, s_2) = 0, \gamma, \varsigma$

2) for $k = 1$ to $k \leqslant \left\lceil \dfrac{\ln \varsigma}{\ln \gamma} \right\rceil$ do

3)　　for $i = 1$ to $|A|$ do

4)　　$T_K(d)(P(s_1, a_i, \bullet), P(s_2, a_i, \bullet))$

5)　　end for

6)　　$d(s_1, s_2) = \max_{a \in A}\{d_a(s_1, s_2) + \gamma T_K(d)(P(s_1, a_i, \bullet), P(s_2, a_i, \bullet))\}$

7) end for

## 3　增强型深度确定策略梯度算法

### 3.1　样本池的构建

为了解决样本之间的关联性问题，Mnih 等[8]使用经验回放打破样本的关联性。本文将利用 3 个样本池——原始样本池 $R_0$、多样性样本池 $R_1$ 和高误差样本池 $R_2$，来存储学习所需要的样本。在学习过程中，E-DDPG 算法构建两套价值网络——行动家网络和评论家网络，行动家网络 $\mu(s|\theta^{\mu})$ 用来决定动作的选择，评论家网络 $Q(s, a|\theta^{Q})$ 用于评估状态动作对的好坏。通过行动家网络 $\mu(s|\theta^{\mu})$ 选择动作与环境进行交互，将所获取的样本 $(s_t, a_t, r_t, s_{t+1})$ 放入原始样本池 $R_0$。在学习过程中，计算动作值函数的 TD error $\varphi_t$（如式(7)所示），并根据 TD error 将高误差的样本放入高误差样本池 $R_2$ 中。

$$\varphi_i = r(s_i, a_i) + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) - Q(s_i, a_i|\theta^{Q}) \tag{7}$$

当 $\varphi_i$ 较大时，说明样本对动作值函数的变化影响较大，可以认为该样本具有更高的价值，因此，将该样本放入高误差样本池 $R_2$。当高误差样本池 $R_2$ 中样本足够时，选取训练样本不再单一地从原始样本池 $R_0$ 中选取，而是分别从原始样本池 $R_0$ 和高误差样本池 $R_2$ 中按一定比例选取。

同时，为了保证选取样本的多样性，引入自模拟度量方法。从原始样本池 $R_0$ 和高误差样本池 $R_2$ 中随机选取的样本，可能存在很多近似样本，甚至是重复样本，这会降低算法的执行效率。因此，考虑间隔 $N$ 个情节，利用算法 1 计算出原始样本池 $R_0$ 中样本之间的距离，将低相似性样本放入多样性样本池 $R_1$，以保证所选择样本的多样性。此后，算法 1 将分别从多样性样本池 $R_1$ 和高误差样本池 $R_2$ 按一定比例选取样本，进行学习，同时兼顾样本多样性以及高价值样本信息，进一步提高算法的执行效率。

### 3.2　行动者—评论家网络参数更新

为了保证评论家网络的稳定性，考虑在传统行动者—评论家方法估值网络的基础上，重新构建新的目标行动者—评论家网络，即 $Q'(s, a|\theta^{Q'})$ 和 $\mu'(s|\theta^{\mu'})$，以用于更新学习到的参数。目标策略网络和目标值网络中的参数由当前学习网络中的参数进行更新。

$$\theta^{\mu'} \leftarrow \alpha\theta + (1-\alpha)\theta^{\mu'}, \theta^{Q'} \leftarrow \alpha\theta^{Q} + (1-\alpha)\theta^{Q'} (\alpha << 1)$$

其中，$\alpha$ 不是一个定值，而是随训练情节增加而逐步增大的值。这在保证网络稳定性的同时，能够适当提升算法的训练速度。评论家网络的参数更新式是式(4)和式(5)，行动者网络的参数更新如式(8)所示。

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\theta^{Q})|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})|s_i \tag{8}$$

DDPG 算法通过引入噪声来进行动作探索，噪声信息具有很大的随机性，结合 $\mu(s_t|\theta_t^{\mu})$ 进行动作探索，即在任意状态 $s_t$ 下，选择动作 $a_t \sim N(\mu(s_t|\theta_t^{\mu}), \sigma_t^2)$，其中，$\sigma_t$ 表示策略探索的强度，可以使探索动作更加具有针对性。实验表明，与 DDPG 算法相比，本文所采用的动作探索方法可以使算法在收敛后具有更好的稳定性。

### 3.3　E-DDPG 算法

根据 3.1 节和 3.2 节的介绍，下面给出详细的 E-DDPG 算法的流程，如算法 2 所示。

**算法 2**　E-DDPG 算法

1) 初始化评论家网络 $Q(s,a|\theta^Q)$ 和行动者网络 $\mu(s|\theta^\mu)$ 及它们的权重 $\theta^Q$ 和 $\theta^\mu$；目标网络 $Q'$ 和 $\mu'$，且权重 $\theta^{Q'}\leftarrow\theta^Q$，$\theta'\leftarrow\theta$，原始样本池 $R_0$、多样性样本池 $R_1$ 和高误差样本池 $R_2$ 初始为空，自模拟度量间隔情节 $N$；时间步 $T$；阈值 $p$

2) for *episode* = 1 to $M$ do

3) if *episode* mod $N==0$

4) 利用自模拟度量方法，将低相似性样本放入多样性样本池 $R_1$

5) 获得初始观察状态 $S_1$

6) for $t$ = 1 to $T$ do

7) 根据当前的策略利用高斯分布选择动作 $a_t\sim N(\mu(s_t|\theta_t^\mu),\sigma_t^2)$

8) 执行动作 $a_t$，观察立即奖赏 $r_t$ 和新的状态 $s_{t+1}$

9) 将 $(s_t,a_t,r_t,s_{t+1})$ 存入原始样本池 $R_0$ 中

10) if $R_1\neq$ null

11) 随机地从多样性样本池 $R_1$ 和高误差样本池 $R_2$ 中各挑选一定数量的 $(s_i,a_i,r_i,s_{i+1})$ 样本

12) else if $R_2\neq$ null

13) 随机地从原始样本池 $R_0$ 和高误差样本池 $R_2$ 中各挑选一定数量的 $(s_i,a_i,r_i,s_{i+1})$ 样本

14) else

15) 随机地从原始样本池 $R_0$ 中挑选一定数量的 $(s_i,a_i,r_i,s_{i+1})$ 样本

16) 设置 $y_i=r(s_i,a_i)+\gamma Q'(s_{i+1},\mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$

17) if $y_i-Q(s_i,a_i|\theta^Q)>p$ then 将 $(s_i,a_i,r_i,s_{i+1})$ 放入高误差样本 $R_2$ 中

18) 通过最小化 $L=\frac{1}{N}\sum_i(y_i-Q(s_i,a_i|\theta^Q))^2$ 来更新 $\theta^Q$

19) 通过样本策略梯度 $\nabla_{\theta^\mu}J\approx\frac{1}{N}\sum_i\nabla_a Q(s,a|\theta^Q)|_{s=s_i,a=\mu(s_i)}\nabla_{\theta^\mu}\mu(s|\theta^\mu)|s_i$ 更新 $\theta^\mu$；

20) 更新目标网络中的参数 $\theta^{\mu'}\leftarrow\alpha\theta^\mu+(1-\alpha)\theta^{\mu'}$，$\theta^{Q'}\leftarrow\alpha\theta^Q+(1-\alpha)\theta^{Q'}$；

21) end if

22) end if

23) end if

24) end for

25) end if

26) end for

为获取高价值的样本，E-DDPG 算法的高误差样本池 $R_2$ 引入了 TD error $\varphi_t$，实验表明，该方法可以加速算法的收敛，但是其中 $\varphi_t$ 的阈值需要人工设置。

### 3.4 关于多样性样本池的分析

在算法实际运行过程中，因为环境模型是未知的，无法得到转移函数即 $p(s_{t+1}|s_t,a_t)$。针对该问题，本文采取数学统计的方法，通过统计样本池中状态 $s$ 转移到下一个状态 $s'$ 出现的次数，以近似地表示 $p(s_{t+1}|s_t,a_t)$。由大数定理可知，该次数与总的样本数的比值就是该状态 $s$ 到下一状态 $s'$ 的转移概率。为了提高算法的执行效率，在间隔一定情节之后，才会利用算法 1 计算样本间的距离，更新多样性样本池 $R1$，从而达到以尽量小的运算代价保证训练样本的多样性，提高算法的执行效率。

假设在任意状态下的动作空间一致 $s_1$ 和 $s_2$ 分别表示状态空间中的两个状态，状态 $s_1$ 和 $s_2$ 具有相同的动作空间，采用任意动作 $a$ 后，其后续状态分别为 $s_1'$ 和 $s_2'$，且 $Q^*$ 表示最优动作值函数。

**定理 2** 假设 $d_\sim(s_1,s_2)=m$，则 $|Q^*(s_1'|s_1,a)-Q^*(s_2'|s_2,a)|\leqslant m$。

**证明**

$$
\begin{aligned}
&|Q^*(s_1'|s_1,a)-Q^*(s_2'|s_2,a)|\\
&=|(r(s_1,a)+\gamma\sum_{s_1\in S}p_1(s_1'|s_1,a)V_1^*(s_1'))-\\
&(r(s_2,a)+\gamma\sum_{s_2\in S}p_2(s_2'|s_2,a)V_2^*(s_2'))|\\
&\leqslant\max_{a\in A}|r(s_1,a)+\gamma\sum_{s_1\in S}p_1(s_1'|s_1,a)V_1^*(s_1')-\\
&(r(s_2,a)+\gamma\sum_{s_2\in S}p_2(s_2'|s_2,a)V_2^*(s_2'))|\\
&=\max_{a\in A}|d_a(s_1,s_2)+\gamma T_k(d)(p_1(s_1,a,\bullet),p_2(s_2,a,\bullet))|\\
&=d_\sim(s_1,s_2)\\
&=m
\end{aligned}
$$

因此，$\forall s_1,s_2\in S$，使 $|Q^*(s_1'|s_1,\mathrm{a})-Q^*(s_2'|s_2,a)|\leqslant m$ 成立。

证毕。

因此，利用自模拟度量方法计算样本间的距离，利用该距离确定样本间的相似性可以进一步反映样本在值函数空间中的相似性。从参数更新的角度而言，在算法学习过程中，高相似性的样本具有较低的价值，而低相似性的样本将提高算法的更新

效率，进而加快算法收敛速度。

### 3.5　关于高误差样本池的分析

在强化学习中，从历史样本池中进行均匀采样，可能会导致较多的更新集中在某一些低价值的样本上，如果将更新集中在某些特殊的样本上，则会使算法的更新更加高效。在均匀采样训练的过程中，会浪费大量时间和计算资源进行很多无用的更新，随着学习的不断进行，有用的更新区域不断增加，但是与将更新集中在高价值的样本上相比，学习的效率和效果差了很多。在连续状态空间中，这种非集中式搜索的效率将会非常低下。

本文以 TD error 作为启发式信息，将训练中高价值的样本挑选出来构建高误差样本池，在接下来的训练中，通过提高这些高价值样本的选取概率，进而更快地获得有用的更新区域。由于关于 TD error 的阈值是人为设置的，若仅仅从高误差样本池 $R_2$ 中选取训练样本，可能导致错过部分高价值样本，因此，算法同时也从多样性样本池中选择一定比例的样本。实验结果表明，该方法可以提高算法的收敛速度。

## 4　实验结果分析

### 4.1　Pendulum 问题

1)实验描述

为了验证算法的有效性，本文将 DDPG 算法和 E-DDPG 算法用于经典 Pendulum 问题。图 1 给出了 Pendulum 问题的示意。

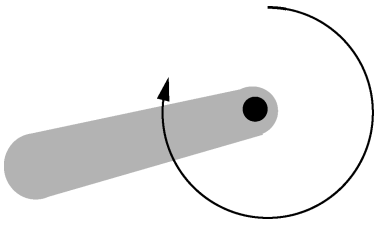


图 1　Pendulum 问题的示意

一个倒立的钟摆，摆杆绕中间转轴随机摆动。agent 的任务是学习到一个策略，使摆杆保持竖直。本文实验环境是 OpenAI gym，状态是三维的，其中，二维表示钟摆的位置，一维表示钟摆的速度。状态可以表示为

$$s=\{\cos(theta),\sin(theta),thetadot \mid theta\in[-1,1],\ thetadot\in[-8,8]\} \tag{9}$$

动作是一维的，表示对钟摆的作用力，取值范围为 [−2,2]。动作可以表示为

$$a=\{max\_torque \mid max\_torque\in[-2,2]\} \tag{10}$$

奖赏函数可以表示为

$$r=-(theta^2+0.1thetadot^2+0.01action^2),\ action\in[-2,2] \tag{11}$$

其中，$r$ 等于式(9)的计算值的概率是 0.1，等于 0 的概率是 0.9。

2) 实验设置

实验运行硬件环境为 Inter(R) Xeon(R) CPU E5-2660 处理器、NVIDIA GeForce GTX 1060 显卡、16 GB 内存；软件环境为 Windows 10 操作系统、python 3.5、TensorFlow_GPU-1.4.0。

神经网络参数的优化使用 Adam 优化器，其中，行动家网络的参数学习率是$10^{-4}$，评论家网络的参数学习率是$10^{-3}$，$L_2$ 权重缩减速率是$10^{-2}$，折扣因子是 0.99。目标网络的更新参数 $\alpha=0.001$，每隔 300 个情节更新参数 $\alpha$=1.1$\alpha$，神经网络的隐藏层使用修正非线性单元，行动家网络最后的输出层是 tanh 层，高斯分布的 $\sigma$=0.2，更新多样性样本池 $R_1$ 的情节间隔数 $N$=30，每个情节中的最大时间步数是 2 000（即当时间步达到 2 000 时，则情节结束）。

在该实验中，DDPG 算法收敛需要 8.1 h，未引入自模拟度量的 E-DDPG 算法收敛需要 5.2 h，而引入自模拟度量的 E-DDPG 算法收敛仅需要 2.4 h。

3) 实验分析

DDPG 算法、E-DDPG 算法应用于经典的 Pendulum 问题上的性能比较（在实验过程中，每个算法都独立执行 3 000 个情节）如图 2 所示，各种算法在不同情节下，目标任务达到终止状态时的总回报值（回报值是通过目标任务从开始状态达到终止状态时总的奖赏值）。其中，横坐标是情节数，纵坐标是算法执行 10 次的平均回报值。从图 2 可以看出，E-DDPG 算法在 300 个情节时基本收敛。DDPG 算法虽然在 400 个情节时取得较高的回报值，但是还在震荡并没有收敛，直到 1 200 个情节才收敛。因为 E-DDPG 算法引入了 TD error，加大了对具有更高价值的样本的选取概率，同时因为采用自模拟度量方法更新多样性样本池 $R_1$，使选取的训练样本多样性得到保证，从而进一步加快算法的收敛速度。此外，从图 2 还可以看出，两种算法在收敛后，E-DDPG 算法每个情节的回报值震荡的幅度比 DDPG 算法的震荡幅度更小，这充分说明

E-DDPG 算法的稳定性比 DDPG 算法更好。

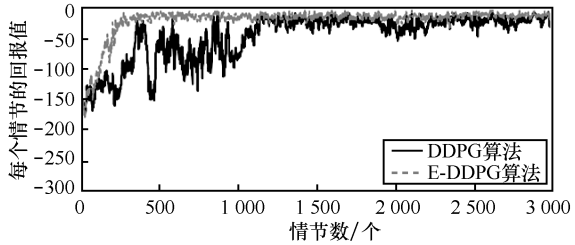



图 2 Pendulum 问题中两种算法的性能比较

引入自模拟度量 E-DDPG 算法、未引入自摸 E-DDPG 算法和 DDPG 算法进行的实验对比，结果如图 3 所示，其中，设置自模拟度量间隔的情节数 $N$=30。从图 3 可以看出，没有引入自模拟度量方法的 E-DDPG 算法在 700 个情节算法才收敛，引入自模拟度量方法的 E-DDPG 算法在 300 个情节算法就收敛，而 DDPG 算法在 1 200 个情节才收敛。因为自模拟度量方法使训练的样本具有更好的多样性，提高了训练的效率，从而加快了训练的速度。

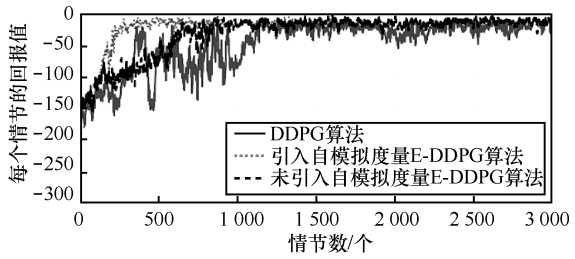



图 3 Pendulum 问题中 E-DDPG 算法是否引入自模拟度量方法与 DDPG 算法的实验对比

图 2 中 TD error 的阈值 $p$ =0.4，训练的小 batch 的样本总数是 64，其中，随机样本每次训练选取 32 个，高 TD error 的样本每次训练选取 32 个。以上参数均为手工设置，本文针对这两点分别设计实验，验证算法的收敛性与 TD Error 的阈值 $p$ 设置和样本选取方式之间的关联性。

从图 4 可以看出，虽然 TD Error 的阈值 $p$ 不同，但是 E-DDPG 算法收敛速度依然比 DDPG 算法快。当选取的 TD error 的阈值 $p$ =0.3 时，E-DDPG 算法在 150 个情节获得较好的回报，但是并没有稳定，之后还会震荡，当选取 TD error 的阈值 $p$ =0.5 时，E-DDPG 算法在 220 个情节就可以获得较好的回报值，虽然此时获得较好回报所需要的情节数大于 TD error 阈值 $p$ =0.3，但是在获得较好回报后基本稳定。TD error 的阈值 $p$ =0.5 时的性能比 TD error 的阈值 $p$ =0.3 的性能更好。但是 TD error 的阈值 $p$ 也不是越高越好，当选取 TD error 的阈值 $p$ =0.6 时，E-DDPG 算法在 380 个情节收敛，算法在收敛后的稳定性也变差。这是由于阈值 $p$ 过高，会导致高价值的样本数量过少，每次从高误差样本池 $R_2$ 中选出来的训练样本过度重复，减慢算法的收敛速度。

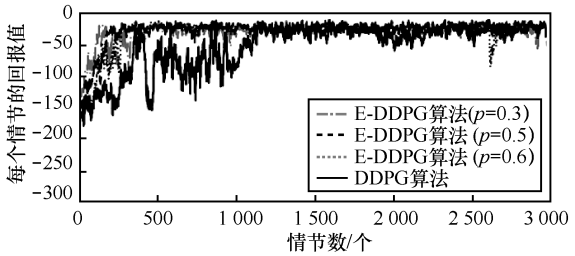



图 4 Pendulum 问题中 E-DDPG 算法不同 TD Error 和 DDPG 算法的实验对比

从图 5 可以看出，不同的样本选取也会对算法的收敛性和稳定性产生影响，但是收敛效果比 DDPG 算法的收敛性好。在图 5 中设置 TD error 的阈值 $p$ =0.4，训练的小 batch 的样本总数是 64，其中，多样性样本池 $R_1$ 每次训练样本选取 48 个，高误差样本池 $R_2$ 每次训练样本选取 16 个，即 4 816 选取方式；训练的小 batch 的样本总数是 64，其中，多样性样本池 $R_1$ 每次训练样本选取 16 个，高误差样本池 $R_2$ 每次训练样本选取 48 个，即 1 648 选取方式，然后分别独立进行实验。对比图 4 中两种样本的选取比例，样本选取方式为 4 816，算法前期有较好的回报值，但是算法不断震荡，直到 480 个情节才真正收敛；样本选取方式为 1 648，算法在 850 个情节才收敛。实验结果表明，多样性样本池 $R_1$ 选取 48 个样本的训练效果比选取 16 个样本的训练效果更好一点。这是由于高误差样本池 $R_2$ 中的样本源于已经训练样本中 TD error 大的样本，因为多样性样本池 $R_1$ 中每次训练样本所占的比例较小，会导致高误差样本池 $R_2$ 中的样本数量增加缓慢，样本较少。此外，高误差样本池 $R_2$ 中样本有较大的选取比例，因此同一样本被选取进行训练次数太多，从而导致新的样本被选取数量太少，造成收敛性能变差。

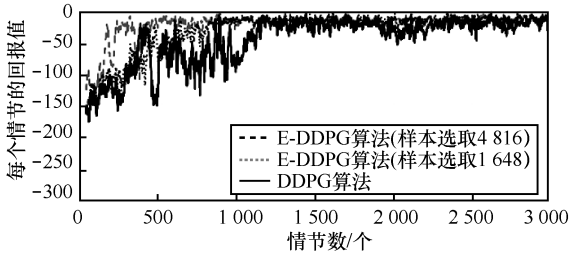



图 5 Pendulum 问题中 E-DDPG 算法不同样本选取比例和 DDPG 算法的实验对比

### 4.2　MountainCar 问题

1) 实验描述

为了验证算法的有效性，本文将 DDPG 算法和 E-DDPG 算法用于经典的 MountainCar 问题。图 6 给出了 MountainCar 问题的示意。

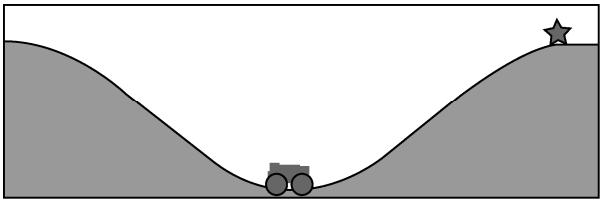


图 6　MountainCar 问题的示意

曲面表示一个带有坡度的路面，小车处在坡底，由于动力不足，小车无法直接加速冲上坡顶，因此必须通过前后加速借助惯性到达坡顶，即图 6 中右侧“星”形标记的位置。本文实验的环境是 OpenAI gym，状态是二维的，其中，一维表示位置，另一维表示速度，状态可以表示为

$$s=\{position,velocity\,|\,position\in[-1.2,0.6],\ velocity\in[-0.07,0.07]\} \tag{12}$$

动作是一维的，表示小车的加速度，取值范围为[−1,1]。动作可以表示为

$$a=\{action\,|\,action\in[-1,1]\} \tag{13}$$

在情节开始时，给定小车一个随机的位置和速度，然后进行交互学习。当小车到达目标位置（图 6 中的“星”形位置）或当前执行的时间步超过 1 000 时，情节结束，并开始一个新的情节。当小车到达目标位置时，立即奖赏是 100；其他情况下，小车的立即奖赏满足

$$r=\{-(0.1action^2)\} \tag{14}$$

2)实验设置

实验运行硬件环境为 Inter(R) Xeon(R) CPU E5-2660 处理器、NVIDIA GeForce GTX 1060 显卡、16 GB 内存；软件环境为 Windows 10 操作系统、python 3.5、TensorFlow_GPU-1.4.0。

本实验神经网络参数的优化使用 Adam 优化器，其中，行动家网络的参数学习率是$10^{-4}$，评论家网络的参数学习率是$10^{-3}$，$L_2$ 权重缩减速率是$10^{-2}$，折扣因子是 0.99。目标网络的更新参数$\alpha$=0. 001，每隔 300 个情节更新参数$\alpha$=1.1$\alpha$，神经网络的隐藏层使用修正非线性单元，行动家网络最后的输出层是 tanh 层，高斯分布的$\sigma$=0.2，更新多样性样本池的情节间隔数 $N$=30，每个情节中的最大时间步数是 1 000（即当时间步达到 1 000 时，则情节结束）。

在本实验中，DDPG 算法收敛需要 7.5 h，未引入自模拟度量的 E-DDPG 算法收敛需要 4.7 h，而引入自模拟度量的 E-DDPG 算法收敛仅需要 1.6 h。

3) 实验分析

DDPG 算法、E-DDPG 算法应用于经典的 MountainCar 问题上的性能比较（在实验过程中，每个算法都独立执行 2 000 个情节）如图 7 所示，各个算法在不同情节下，目标任务达到终止状态时总的回报值（回报值是通过目标任务从开始状态达到终止状态时总的奖赏值）。其中，横坐标是情节数，纵坐标是算法执行 10 次的平均回报值。从图 7 可以看出，E-DDPG 算法在 120 个情节基本收敛。DDPG 算法虽然在 220 个情节时取得较高的回报值，但是还在震荡并没有收敛，直到 780 个情节才收敛。

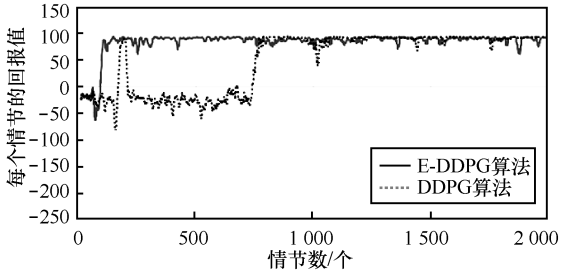



图 7　MountainCar 问题中两种算法的性能比较

E-DDPG 算法是否引入自模拟度量方法进行的实验对比如图 8 所示，其中，设置自模拟度量间隔的情节数 $N$=30。从图 8 可以看出，没有引入自模拟度量方法的 E-DDPG 算法在 470 个情节算法才收敛，引入自模拟度量方法的 E-DDPG 算法在 120 个情节算法就收敛了，而 DDPG 算法在 780 个情节才收敛。这是因为自模拟度量方法使训练的样本具有更好的多样性，提高了训练的效率，从而加快了训练的速度。实验表明，自模拟度量方法能够加快算法的收敛速度。

图 7 中 TD error 的阈值 $p$=0.2，训练的小 batch 的样本总数是 64，其中，随机样本每次训练选取 32 个，高 TD error 的样本每次训练选取 32 个。

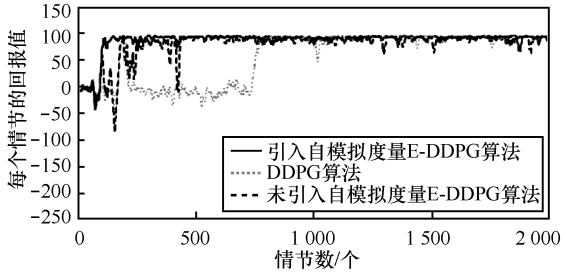



图 8　MountainCar 问题中 E-DDPG 算法是否引入自模拟度量方法的实验比较

从图 9 可以看出，在保证样本选取方式相同的情况下，TD error 的阈值 $p$ 选取的不同，会使算法的收敛性和稳定性不同。虽然 TD error 的阈值 $p$ 不同，但是 E-DDPG 算法收敛速度依然比 DDPG 算法快。当选取的 TD error 的阈值 $p$ =0.3 时，算法在 260 个情节收敛；当选取 TD error 的阈值 $p$ =0.1 时，算法在 150 个情节就可以收敛，TD error 的阈值 $p$ =0.1 时的性能比 TD error 的阈值 $p$ =0.3 的性能更好。TD error 的阈值 $p$ 也不是越低越好，当选取 TD error 的阈值 $p$ =0.09 时，算法在 180 个情节收敛，但是与 TD error 的阈值 $p$ =0.1 相比收敛速度变慢。

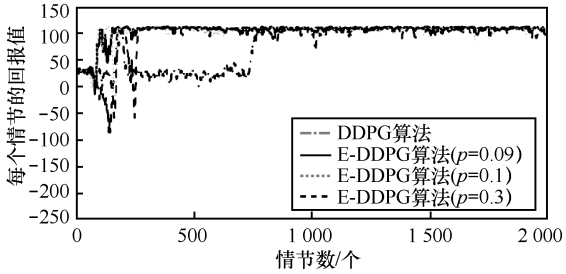



图 9　MountainCar 问题中 E-DDPG 算法不同 TD Error 和 DDPG 算法的实验比较

从图 10 可以看出，不同的样本选取也会对算法的收敛性和稳定性产生影响，但是收敛效果比 DDPG 算法的收敛性好。在图 10 中设置 TD error 的阈值 $p$ =0.2，对样本采用 4 816 和 1 648 选取方式，分别独立进行实验。对比图 10 中两种样本的选取比例，样本选取方式为 4 816，E-DDPG 算法前期有较好的回报值，但是算法不断震荡，直到 260 个情节才真正收敛；样本选取方式为 1 648，E-DDPG 算法在 400 个情节才收敛。实验结果表明，多样性样本池选取 48 个样本的训练效果比选取 16 个样本的训练效果更好一点。

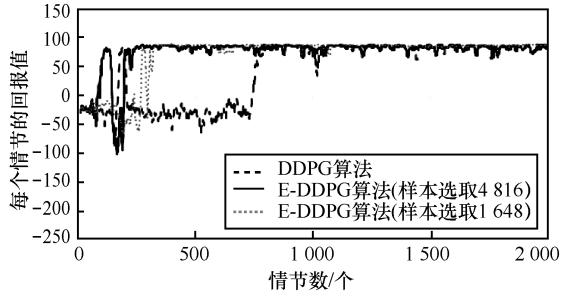



图 10　MountainCar 问题中 E-DDPG 算法不同样本选取比例和 DDPG 算法的实验比较

## 5　结束语

本文针对 DDPG 算法在大规模状态动作空间中存在收敛速度较慢的问题，提出了 E-DDPG 算法。该算法在深度确定策略梯度算法的基础上，重新构建两个新的样本池——多样性样本池和高误差样本池。其中，多样性样本池主要利用自模拟度量方法对原始样本池中的样本相似性进行度量，选择低相似性样本，并在学习过程中持续更新；高误差样本池主要通过计算时间差分误差对所选择的训练样本进行排序，选择具有高价值的高误差样本，以提高后续参数更新的有效性。将算法应用到 Pendulum 问题，从算法性能角度与 DDPG 算法进行比较。实验结果表明，E-DDPG 算法比 DDPG 算法收敛速度更快，同时算法的稳定性也更好。针对 TD error 阈值和多样性样本池与高误差样本池训练样本比例等参数的人工设置不同，对算法性能的影响分别进行了实验。实验结果表明，虽然 TD error 阈值选取和样本选取比例不同会导致 E-DDPG 算法性能不一样，但是与 DDPG 算法相比还是有较好的效果。

本文主要以 Pendulum 问题和 MountainCar 问题作为实验平台验证算法性能，从实验结果可以看出，算法具有较好的收敛性和稳定性。但是 E-DDPG 算法中 TD error 的选取和样本比例的选取都是人工设置的，且不同的设置参数会对算法收敛性和稳定性产生不同的影响。因此，接下来的工作是进一步分析如何设置 TD error 和样本选取比例，让算法可以获得最好的收敛性和稳定性，使算法具有更强的通用性。

## 参考文献：

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

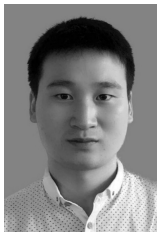
[2] 朱斐, 刘全, 傅启明, 等. 一种用于连续动作空间的最小二乘行动者-评论家方法[J]. 计算机研究与发展, 2014, 51(3): 548-558.
ZHU F, LIU Q, FU Q M. A least square actor-critic approach for continuous action space[J]. Journal of Computer Research and Development, 2014, 51(3): 548-558.

[3] 孙志军, 薛磊, 许阳明, 等. 深度学习研究综述[J]. 计算机应用研究, 2012, 29(8): 2806-2810.

SUN Z J, XUE L, XU Y M, et al. Overview of deep learning[J]. Application Research of Computers, 2012, 29(8): 2806-2810.
[4] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.
[5] HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets[J]. Neural Computation, 2006, 18(7): 1527-1554.
[6] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of Go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.
[7] SILVER D, SCHRITTWIESER J, SIMONYAN K, et al. Mastering the game of go without human knowledge[J]. Nature, 2017, 550(7676): 354-359.
[8] MNIH V, KAVUKCUOFLU K, SILVER D, et al. Playing atari with deep reinforcement learning[C]//Workshops at the 26th Neural Information Processing Systems. 2013.
[9] MNIH V, KAVUKCUOFLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.
[10] WATKINS C J C H. Learning from delayed rewards[J]. Robotics and Autonomous Systems, 1989, 15(4): 233-235.
[11] VAN H V, GUEZ A, SILVER D. Deep reinforcement learning with double q-learning[C]//The AAAI Conference on Artificial Intelligence. 2016.
[12] HASSELT H V. Double Q-learning[C]//The Advances in Neural Information Processing Systems. 2010.
[13] SCHAUL T, QUAN J, ANTONOGLOU I, et al. Prioritized experience replay[C]//The 4th International Conference on Learning Representations. 2016: 322-355.
[14] SUTTON R S, MCALLESTER D, SINGH S, et al. Policy gradient methods for reinforcement learning with function approximation[J]. Advances in Neural Information Processing Systems, 2000, 12: 1057-1063.
[15] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[C]//The 4th International Conference on Learning Representations. 2015.
[16] SILVER D, LEVER G, HEESS N, et al. Deterministic policy gradient algorithms[C]//The International Conference on Machine Learning. 2014.
[17] GIVAN R, DEAN T, GREIG M. Equivalence notions and model minimization in Markov decision processes[J]. Artificial Intelligence, 2003, 147(1-2): 163-223.
[18] FERNS N, PANANGADEN P, PRECUP D. Metrics for finite markov decision processes[C]//The 20th Conference on Uncertainty in Artificial Intelligence. 2004.

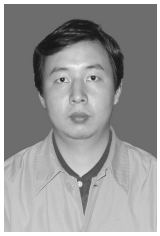
[作者简介]

**陈建平**（1963−），男，江苏南京人，博士，苏州科技大学教授，主要研究方向为大数据分析与应用、建筑节能、智能信息处理。



**何超**（1993−），男，江苏徐州人，苏州科技大学硕士生，主要研究方向为强化学习、深度学习、建筑节能。

**刘全**（1969−），男，内蒙古牙克石人，博士，苏州大学教授、博士生导师，主要研究方向为智能信息处理、自动推理与机器学习。



**吴宏杰**（1977−），男，江苏苏州人，博士，苏州科技大学副教授，主要研究方向为深度学习、模式识别、生物信息。

**胡伏原**（1978−），男，湖南岳阳人，博士，苏州科技大学教授，主要研究方向为模式识别与机器学习。

**傅启明**（1985−），男，江苏淮安人，博士，苏州科技大学讲师，主要研究方向为强化学习、深度学习及建筑节能。