Search by Target Distribution Learning for Continuous C

ıeng Zhang
ıghua University
.eng123@live.com

Yuanqi Li
IIIS, Tsinghua University
timezerolyq@gmail.com

Jian L
IIIS, Tsinghua U
lapordge@gm

【强化学习 100】TDL

张楚珩
清华大学 交叉信息院博士在读

终于迎来第 100 篇，在第 100 篇的时候讲一下自己的工作 TDL（AAAI-20 oral）。 TDL 是 target distribution learning 的简称。

原文传送门

Zhang Chuheng, Yuanqi Li, and Jian Li. "Policy Search by Target Distribution Learning for Continuous Control." arXiv preprint arXiv:1905.11041 (2019).

特色

传统的策略梯度方法常常会直接求目标函数关于神经网络参数的导数；在连续控制问题上，常常会学习一个随机策略，最常见的就是学习一个神经网络去输出高斯分布。我们设计了一个很简单的环境，在该环境中，最优策略是一个确定性策略。我们在实验中发现一些常见的算法（比如 PPO、A2C 等）在这样简单的环境中都不能很稳定地学习到最优策略。其主要的原因是当神经网络输出的分布的方差变小的时候，策略梯度会变得特别大，从而造成不稳定。

为了解决该稳定性问题，我们提出了目标分布学习（target distribution learning，TDL）方法用于强化学习中的连续控制优化问题。该方法通过迭代优化的方式来进行策略学习。在每一轮迭代中，该方法先根据按当前策略采样得到的样本来计算得到一个目标行动分布，再通过优化策略神经网络来逼近该分布。该方法在稳定性方面具有以下两方面的优势：其一是该方法能够有效地限定每一轮迭代中策略的变化，从而使得训练的过程变得更加稳定；其二是在每一轮迭代的优化步中，神经网络针对一个固定的目标来做优化，因此该步骤的优化效果受优化参数的影响更小，从而使得整个算法更加稳定。在 Mujoco 中的连续控制任务上，该算法的性能都接近或者超过了现有最优的算法。同时，相比于现有的一些方法，该算法在训练过程中的表现更为稳定，对于超参数也更为鲁棒。

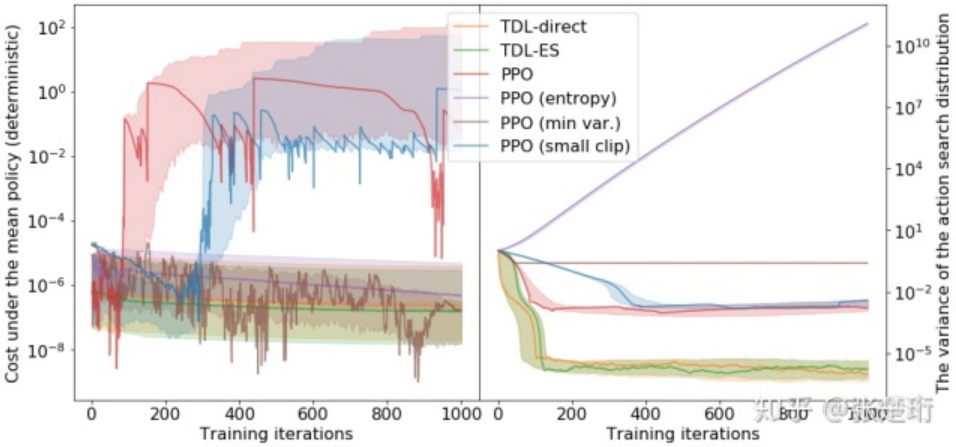
废话说了特别多，简言之，从个人的实验经验上面来看，我们提出的这个方法在训练过程中贼稳定，调参也很容易。实用性上来说，如果想要解决一个基于高斯分布的连续控制问题的话，非常推荐大家试用一下我们的方法。

过程

1. 现有算法的问题性问题

我们设计了一个很简单的环境：每回合只有一步的 RL，行动 a 是一个实数，每步造成一个 cost $c=a^2$，状态 s 也是一个随机的实数，实质上不起任何作用。这个环境里面的最优策略也很容易看出，就是每次都输出 $a=0$ 即可。就是这样的一个环境，PPO 表现都不是很好（见下图红线）。

同时，我们为了解决这个问题，我们也尝试了一下很简单的 trick，比如增加 entropy 项、设置最小 variance、设置小的 clip constant 等，效果都不好。



注意到，这是策略梯度类方法的一个比较常见的问题，其产生的原因是当逐步学习到确定性的最优策略时，策略会输出较小的方差，而方差较小时梯度会变得特别大。

$$\frac{\partial \hat{L}(\theta)}{\partial \theta} = \frac{1}{T}\sum_{t=1}^{T}\left[\hat{L}_t(\theta)\frac{\partial \log \mathcal{N}(a_t|\mu_\theta(s_t),\sigma)}{\partial \mu_\theta(s_t)}\frac{\partial \mu_\theta(s_t)}{\partial \theta}\right]. \tag{3}$$

$$\frac{\partial \log \mathcal{N}(a_t|\mu_\theta(s_t),\sigma)}{\partial \mu_\theta(s_t)} = \frac{a_t - \mu_\theta(s_t)}{\sigma^2}. \tag{4}$$

2. TDL 算法

我们提出的方法就是对这个训练过程『解耦合』，不要在样本上直接做策略梯度，而是先人为按照梯度公式计算出一个目标分布，然后在以有监督学习的方式来训练策略去拟合这个分布。

不同于以前的方法直接对如下目标做梯度：

$$L_{t,1}(\mu,\sigma) = \mathbb{E}_{a\sim\mathcal{N}(\mu,\sigma)}\left[A^{\pi^{\text{old}}}(s_t,a)\right] \tag{5}$$

我们提出，优化每一轮策略改进的概率，这样算法会更稳定：

$$L_{t,2}(\mu,\sigma) = \mathbb{E}_{a\sim\mathcal{N}(\mu,\sigma)}\left[\mathbb{I}\{A^{\pi^{\text{old}}}(s_t,a)>0\}\right] \tag{6}$$

由此，我们提出了几种 TDL 的算法，算法框图如下，具体的公式可以参考原文。

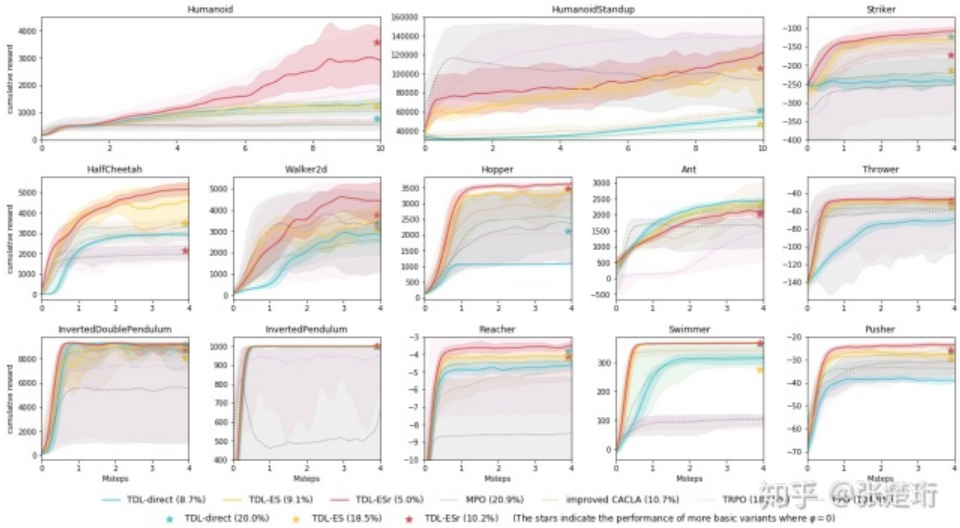
Algorithm 1 Target learning

Number of timesteps in one iteration T, minibatch size M, number of epochs E
Initialize the action distribution of the policy $\mathcal{N}(\mu_\theta(\cdot),\sigma_\theta(\cdot)=\sigma^{1/(\varphi+1)}\tilde{\sigma}_\theta(\cdot)^{\varphi/(\varphi+1)})$
Initialize the critic network $V_\phi(\cdot)$
for $i=0,1,2,\cdots$ **do**
  Interact with the environment and obtain T on-policy transitions $\{(s_t,a_t,r_t,s_{t+1})\}$
  Calculate the Monte Carlo return for each transition $R_t=\sum_{t'=t}^{T}\gamma^{t'-t}r_{t'}$
  Calculate the advantage function estimate $\hat{A}_t$ for each transition (by GAE)
  Calculate the target standard deviation $\hat{\sigma}_t$ following (8)
  if TDL-direct **then** calculate the target means $\hat{\mu}_t$ following (11)
  if TDL-ESr **then** revise the sampled actions following (15)
  if TDL-ES **or** TDL-ESr **then** calculate the target means $\hat{\mu}_t$ following (12)
  for $j=1:ET/M$ **do**
    Sample a minibatch that contains M transitions
    Update the policy network to minimize $\frac{1}{M}\sum_{t=1}^{M}(\hat{\mu}_t-\mu_\theta(s_t))^2$ and $\frac{1}{M}\sum_{t=1}^{M}(\hat{\sigma}_t-\tilde{\sigma}_\theta(s_t))^2$ on the minibatch
    Update the critic network to minimize $\frac{1}{M}\sum_{t=1}^{M}(R_t-V_\phi(s_t))^2$ on the minibatch
  Update σ to $\hat{\sigma}$ defined in (10) and $\sigma_\theta(\cdot)$ following (9)

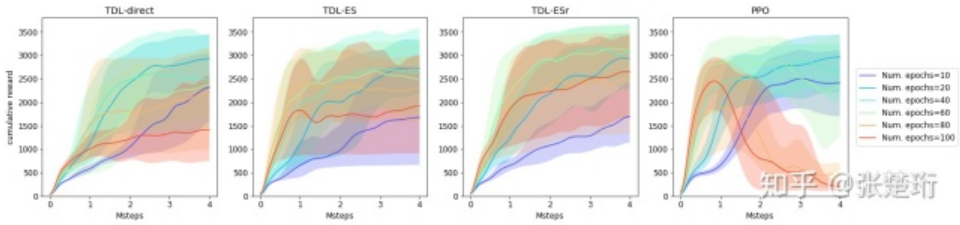
3. 实验

在实验中，我们发现了 TDL 算法几方面的优势。

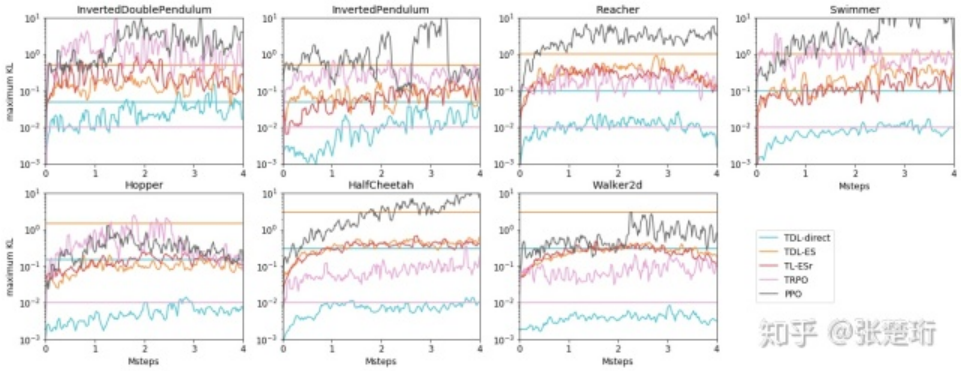
第一，该算法在性能和样本效率上基本上达到 state-of-the-art 效果。



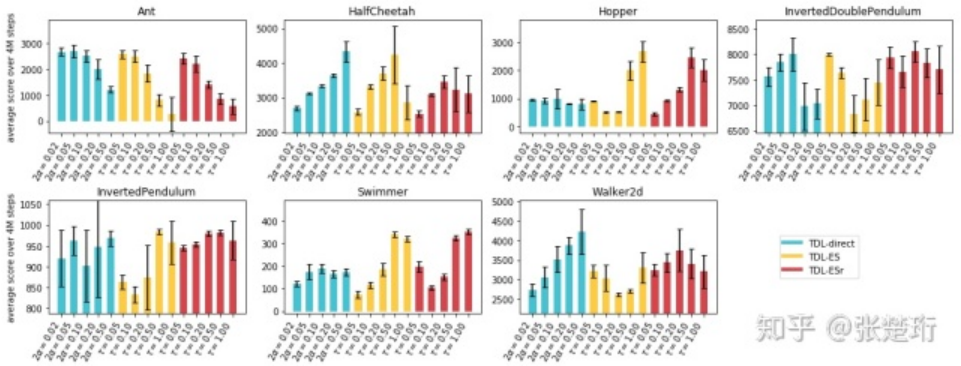
第二，回忆到在 PPO 等 on-policy 的算法里面，为了提高样本的使用效率，我们每采集一波样本，都会反复把它们送到神经网络里面训练多轮。这样的做法虽然提高了样本使用效率，但是如果让神经网络在同一波数据上面学习多轮，策略可能会『走远』了，从而造成性能的降低。在我们的算法中，由于对于每一波数据都计算出来了固定的目标分布，反复在这一波数据上训练策略网络去拟合目标分布并不会导致策略『走远』，因此我们的算法可以安全地增大同一波数据上面训练的轮数，从而最大化样本使用效率。



第三，回忆到，由于在策略学习的过程中，样本的分布是在变化的，因此基于策略梯度的算法需要某种程度上 conservative（即，每一轮策略变化不要太大），才能有较好的性能保证。由于使用了目标分布，因此每一轮策略变化是非常可控的，我们能够准确控制到希望一轮策略更新，其新旧策略的 KL divergence 相差有多大。相比之下，PPO、TRPO 就只能通过经验上的超参数来控制。



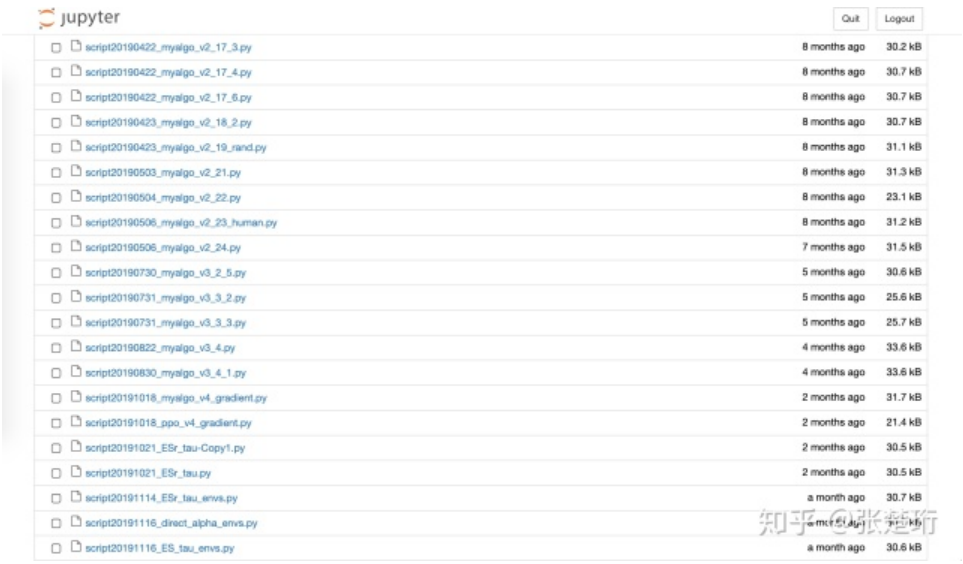
第四，我们的算法对于超参数也不是很敏感，即，使用起来不用费大力气调参。



这一篇 blog 拖了真的是太久了，以至于我今天真的打算写一下最近读的一些理论工作了，发现编号连不上才又返回来把它写完。

作为博士的第一篇工作，这个工作的历程真是艰辛（但是充实和开心）的博士生涯的一个缩影。大概在 2018 年十一之后提出来了最初的方案，由于十分没有科研经验，相关的文献也了解的很少，凭着自己所阅读的若干个 benchmark 算法和动手做的一些简单实验，就提出了若干个十分粗糙的项目方案。跟导师讨论之后，导师（凭着多年的经验和 insight，基于可行性，我猜）选了这个基于 PPO 的改进方案。在做的过程中，一边做一边想一边改，改到 2019 年过年的时候基本上能跑出来一些实验效果了。然后就叫了学弟加入帮忙调参（学弟真的超级靠谱！马上也会有我们合作的第二篇 paper 了！）。春节回来之后就一边继续改进，一边准备投 NeurIPS，改到 NeurIPS 的版本时，

已经和最初提出的方案完全不一样了，甚至连 motivation 都全部变了个说法。最后 NeurIPS 还是被拒了，感觉还是跟自己个人经验太少有关，写文章的时候连 Related Work section 都没有。提交前几天被老板催着写 related work，但是由于我实在了解太少，憋了几天都没写出了，只在 introduction 的部分勉强概述了一下。最后审稿结果果然主要以没有对比前人工作为由给拒了。接下来，有一段时间真是心累，然后再补实验，改 paper 投了 AAAI。AAAI rebuttal 的时候生怕被拒，强力跪舔审稿人，最后给了个 oral。



从最开始的试验，到最后文章里面用到的试验，写了有超过一百个版本 code。

接下来，我个人将会更多地在强化学习理论方面探索，欢迎大家一块来交流！

AAAI Oral Presentation Slides

Policy Search by Target Distribution Learning for Continuous Control

Chuheng Zhang **Yuanqi Li** **Jian Li**

Institute for Interdisciplinary Information Science (IIIS), Tsinghua University

Continuous Control

- Continuous control problem: action is continuous
- Continuous control has many real applications:

Robotic Control

Finance
Order Book Execution

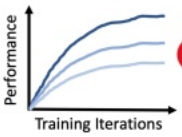
Healthcare
Dynamic medical treatment

- Value-based methods: hard to convert the value to policy
- Policy-based methods: learn the policy directly

Robustness in RL

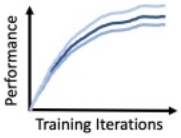
- Robustness is important for real applications

Not stable during the training
- May cause damage in real systems



High variance across different runs / different hyperparameters
- Performance is not assured in a single run

Our objective



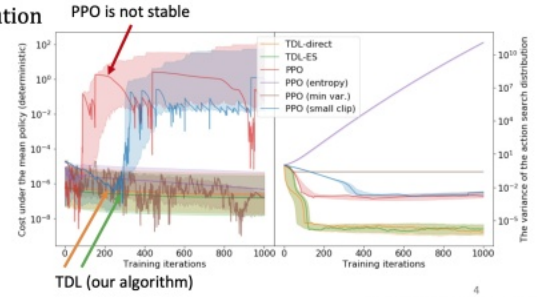
Instability in PPO

- Experiment: PPO [Schulman et al 2017] is unstable even in a simple environment
- Environment: cost: $c = -a^2$; state: $s \sim U[0,1]$; optimal policy: $a = 0$
- Policy: output action distribution

The std. of the gaussian distribution used in the policy $\mathcal{N}(\mu, \sigma)$

$$\left|\frac{\partial L(\theta)}{\partial \theta}\right| \propto \frac{1}{\sigma}$$

Average gradient



Target Distribution Learning (TDL)

- Our solution based on CPI: decouple the policy gradient backward process

For each iteration:
1. Collect samples following the current policy
2. Set targets of policy based on the samples
3. Update the policy network towards the targets

- Target distribution on each state sample s_t: what the policy should output on s_t?
- We set the targets relying on the evolutionary strategy rather than the gradient
  - Our objective: maximize the probability of policy improvement

$$L_{t,2}(\mu, \sigma) = \mathbb{P}\left[A^{\pi^{old}}(s_t, a) > 0 | a \sim \mathcal{N}(\mu, \sigma)\right]$$

  - Policy gradient objective: maximize the policy improvement

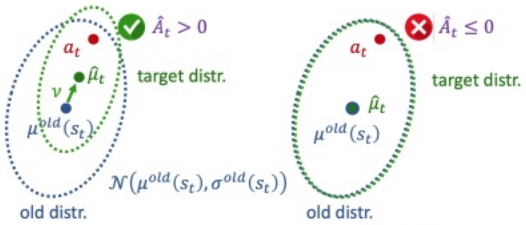
$$L_{t,1}(\mu, \sigma) = \mathbb{E}\left[A^{\pi^{old}}(s_t, a) | a \sim \mathcal{N}(\mu, \sigma)\right]$$

清华大学 交叉信息研究院

Target Distribution Learning (TDL)

- Target distribution for each sample in TDL-ES: a Gaussian distribution $\mathcal{N}(\hat{\mu}_t, \hat{\sigma}_t)$

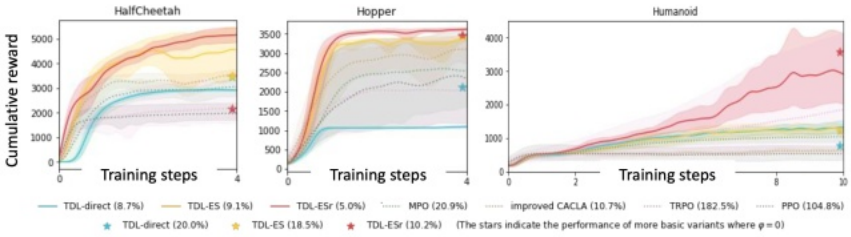
  Old distribution $\mathcal{N}(\mu^{old}(s_t), \sigma^{old}(s_t))$ → action sample a_t → estimate advantage $\hat{A}_t$ → propose $\mathcal{N}(\hat{\mu}_t, \hat{\sigma}_t)$



- Prior knowledge incorporated in the target setting: TDL-ESr

Experiment Results

- TDL achieves comparable asymptotic performance and sample efficiency, while being more robust on Mujoco tasks



MPO [Abdolmaleki et al 2018]

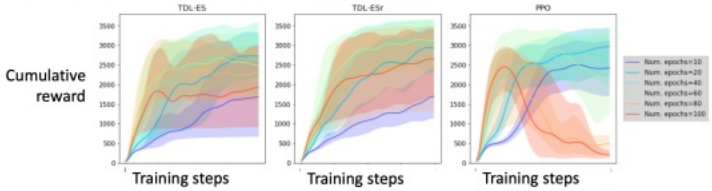
TRPO [Schulman et al 2015]

CACLA [Van Hasselt and Wiering 2007]

PPO [Schulman et al 2017]

Experiment Results

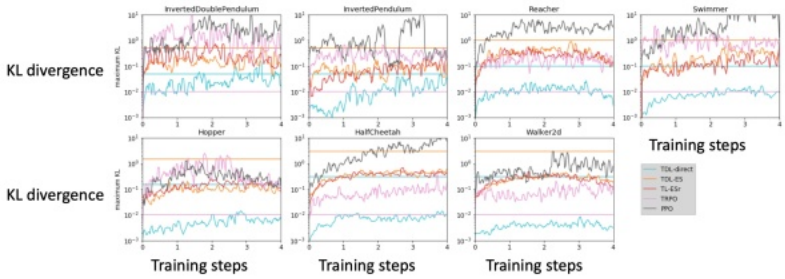
- TDL can safely learn more epochs on the same samples, which improves sample efficiency for on-policy algorithms.



For each iteration:
1. Sample the rollouts
2. Set targets of policy based on the samples
3. Update the policy network towards the targets

Experiment Results

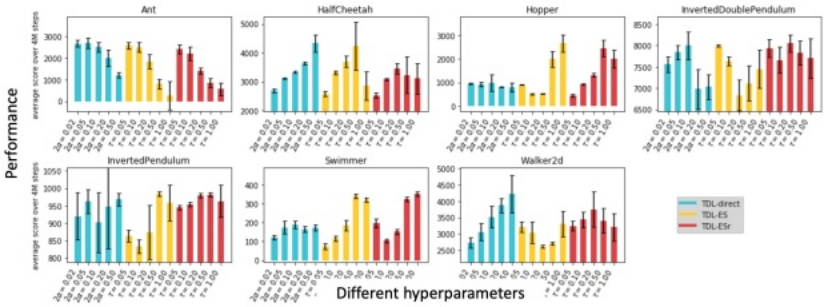
- In TDL, the policy change for each iteration is smaller, yet not reducing sample efficiency.



清华大学 交叉信息研究院

Experiment Results

- TDL is robust to different hyperparameters.



清华大学 交叉信息研究院

Conclusion

- We proposed target distribution learning (TDL), a policy iteration algorithm that decouples the policy gradient process for robust continuous control, which
  - achieves state-of-the-art performance on Mujoco tasks
  - is robust during the training and across different runs
  - is robust to increase sample reuse for the maximum sample efficiency
  - updates more conservatively while being efficient
  - is robust across different hyperparameter settings

Thank you for watching!

- My E-mail: zhangchuheng123@live.com

Source Code

Full Paper Online

编辑于 2020-02-07

强化学习 (Reinforcement Learning) 学术会议 机器学习

文章被以下专栏收录

强化学习前沿
读呀读paper

进入专栏