Emergence of Locomotion Behaviours in Rich Environments（丰富环境中运动行为的出现）

作者：Nicolas Heess, Dhruva TB et al.
单位：DeepMind
论文发表期刊：Artificial Intelligence Computer and information sciences
论文发表时间：10 Jul 2017
论文查看网址：https://arxiv.org/abs/1707.02286

论文贡献：通过对训练环境进行设置，以课程的方式，让强化学习智能体在丰富的环境中由简单任务逐步学习，并伴随着智能体的性能加强来增加任务难度来持续学习。证明可以在有限的奖励信号的情况下，通过对环境的设计让智能体获得丰富而有效的行为，而不需要在奖励函数的设计上太过深入。

一. 写作动机

Why:

在深度强化学习领域，伴随着策略梯度（Policy gradient）算法的兴起，在连续状态空间连续动作空间的任务中取得显著的进步。但都需要针对任务特性来定义明确的奖励函数（Reward function）并利用奖励信号（Reward）对智能体的策略π_θ进行优化，引导智能体的动作行为能够按照预期行事。但是在具有复杂动作行为的任务环境中，通常奖励函数面对这些复杂动作，对智能体的引导是不显著的，即 智能体很难学会复杂动作行为。

What:

奖励函数的设计对于强化学习任务来说是非常重要的，奖励函数的稍微改动都会对智能体的动作行为产生影响。奖励函数产生的即时奖励可表示为 $r = R(s, a)$ 。因此针对具有复杂动作行为的任务环境，为了让智能体在环境中的得分能够收敛，会设计趋于谨慎的奖励函数。但同时谨慎的奖励函数就会**回避**掉强化学习的主要挑战：智能体直接从有限的奖励信号中学习、并引导策略，以期望智能体能够具有丰富而有效的动作行为。

How:

论文作者提出，在环境本身包含足够的丰富性和多样性的情况下，使用 **设置不同难度级别的环境（障碍物）来引导智能体从有限的环境中找到解决问题的方案** 的方法，就能利用简单的奖励函数来产生丰富而稳健的动作行为。这样的设置，既能够避免因为奖励函数和任务环境导致的过度拟合，又能让智能体找到在环境的各种挑战下都适用的特殊解决方案。

二. 背景介绍

1. Policy Gradient（策略梯度）

策略梯度算法是直接对策略π进行学习，在深度强化学习中使用深度神经网络来拟合策略函数（Policy function），因此对于随机策略$\pi_\theta(a|s)$的目标是希望 最大化神经网络参数θ 使得策略 π_θ 在动作行为上的期望回报最大化

$$J(\theta) = \mathbb{E}_{\rho_\theta(\tau)}\left[\sum_t \gamma^{t-1} r(s_t, a_t)\right]$$

其中τ 是智能体与动态环境交互得到的轨迹： $\tau = (s_0, a_0, s_1, a_1 \ldots)$

引导状态转移的主要因素有 智能体的策略π_θ 和 环境本身设置好的状态转移概率，为：

$$p(s_{t+1}|s_t, a_t)，\text{其中} a_t = \pi_\theta(s_t)$$

轨迹τ发生的概率是：

$$\rho_\theta(\tau) = p(s_0)\pi(a_0|s_0)p(s_1|s_0, a_0)\dots$$
$$= p(s_0)\prod_{t=1}^{T} p_\theta(a_t|s_t)p(s_{t+1}|s_t, a_t)$$

对于状态s_t的价值估计采用蒙特卡洛（Monte-Carlo）方法，使用带有 折扣因子γ 的 R_t 来表示 每个步骤的未来累积奖励：

$$R_t = \sum_{t'=t} \gamma^{t'-t} r(s_{t'}, a_{t'})$$
$$= r(s_{t'}, a_{t'}) + \gamma R_{t+1}$$

对 策略π_θ 的深度神经网络参数θ 进行梯度求导：

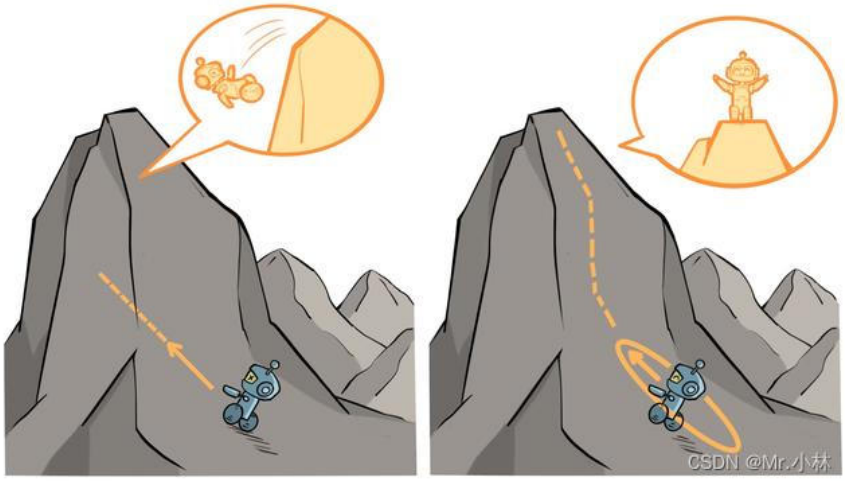
$$\nabla_\theta J = \mathbb{E}_\theta \left[\sum_t \nabla_\theta log\pi_\theta(a_t|s_t)(R_t - b_t)\right]$$

这里的 b_t 作为基线（Baseline）是代表在 状态s_t下 未来的平均累积期望奖励，通常是使用 价值函数（Value Function）进行估计。

$$b_t = V^\theta(s_t) = \mathbb{E}\left[R_t|s_t\right]$$

2. Proximal Policy Optimization（近端策略优化）

在使用策略梯度算法上，需要面临的问题有：
①来自同一条轨迹的累积奖励进行梯度更新，会带来高方差（Variance）的困扰；
②某些关键位置，容易因为策略产生差的动作导致掉入不好的状态、动作分布上，想要重新回到有需要走很长的路



面对上述问题，①是需要减少高方差的问题。②是需要选择一个合适的步长，使得每次更新得到的新策略所实现的回报值（回报 又称 累积奖励）单调不减。

信任区域策略优化（Trust region policy optimization，TRPO）算法中提出的 信赖域（Trust region）方法是更高级的步长更新方法，指在该区域内更新，策略所实现的回报值单调不减。

但是 TRPO 算法为避免 矩阵求逆计算 $\theta^{'} = \theta + \alpha F^{-1}\nabla_\theta J(\theta)$，是使用二阶优化，利用共轭梯度方法来求解，避免计算F矩阵的逆

TRPO的优化目标：

$$maximize_{\theta'}\mathbb{E}_{\tau\sim\rho_\theta(\tau)}\left[\sum_{t=0}^{\infty}\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}\gamma^t A_{\pi\theta}(s_t,a_t)\right]$$

$$D_{KL}\left(\pi_\theta\left(a_t|s_t\right)|\pi_{\theta'}\left(a_t|s_t\right)\right)\leq\epsilon$$

近端策略优化（Proximal policy optimization，PPO）算法基于TRPO算法做工程上的简化

$$maximize_{\theta'}\mathbb{E}_{\tau\sim\rho_\theta(\tau)}\left[\sum_{t=0}^{\infty}\frac{\pi_{\theta'}(a_t|s_t)}{\pi_\theta(a_t|s_t)}\gamma^t A_{\pi\theta}(s_t,a_t)\right]-\beta[D_{KL}\left(\pi_\theta\left(a_t|s_t\right)|\pi_{\theta'}\left(a_t|s_t\right)\right)-\epsilon]$$

将约束 D_{KL} 用拉格朗日乘子法放到优化目标里，引入超参数 β 进行调整。就能够实现**自适应**调整 KL惩罚因子，使得策略在 **信任域（Trust region）** 内更新。

PPO with Adaptive KL Penalty

Algorithm 1 Proximal Policy Optimization (adapted from [8])

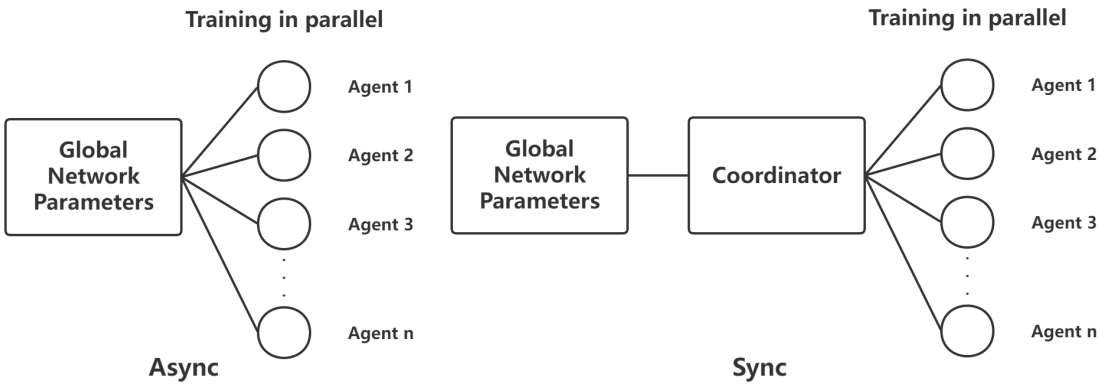
for $i \in \{1,\cdots,N\}$ **do**
  Run policy π_θ for T timesteps, collecting $\{s_t,a_t,r_t\}$
  Estimate advantages $\hat{A}_t=\sum_{t'>t}\gamma^{t'-t}r_{t'}-V_\phi(s_t)$
  $\pi_{\text{old}}\leftarrow\pi_\theta$
  for $j \in \{1,\cdots,M\}$ **do**
    $J_{PPO}(\theta)=\sum_{t=1}^{T}\frac{\pi_\theta(a_t|s_t)}{\pi_{old}(a_t|s_t)}\hat{A}_t-\lambda\text{KL}[\pi_{old}|\pi_\theta]$
    Update θ by a gradient method w.r.t. $J_{PPO}(\theta)$
  end for
  for $j \in \{1,\cdots,B\}$ **do**
    $L_{BL}(\phi)=-\sum_{t=1}^{T}(\sum_{t'>t}\gamma^{t'-t}r_{t'}-V_\phi(s_t))^2$
    Update ϕ by a gradient method w.r.t. $L_{BL}(\phi)$
  end for
  if $\text{KL}[\pi_{old}|\pi_\theta]>\beta_{high}\text{KL}_{target}$ **then**
    $\lambda\leftarrow\alpha\lambda$
  else if $\text{KL}[\pi_{old}|\pi_\theta]<\beta_{low}\text{KL}_{target}$ **then**
    $\lambda\leftarrow\lambda/\alpha$
  end if
end for

小于预值，增大原先策略梯度优化目标权重

大于预值，减少原先策略梯度优化目标权重

基于分布式PPO的可拓展强化学习

本论文考虑到使用的环境包含足够的丰富性和多样性，因此在PPO算法的基础上，提出了分布式的版本**DPPO（Distributed PPO）**，并在**同步更新（Synchronize, Sync）**和**异步更新（Asynchronous, Async）**的方法上分别进行实验，发现在实践中使用**同步更新**的方法能够获得更好的结果。



同步更新方法使用 时间上的协同器（Coordinator），先等待各个线程完成各自的任务，再计算各个线程的梯度平均值，然后统一对参数进行更新。

在 *Asynchronous Methods for Deep Reinforcement Learning(Volodymyr Mnih et al., 2016)* 中对于异步更新方法设置了K步的时间窗口来进行回报值的计算。本论文也将来自K步的累积奖励和优势函数（Advantage function）相结合，并在K步之后从价值函数中自举（Bootstrap）。

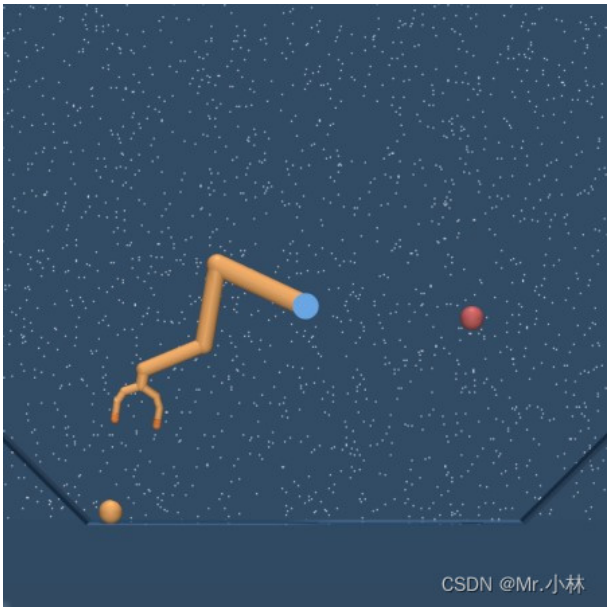
K步（K-steps）的优势函数为：$\hat{A}_t=\sum_{i=1}^{K}\gamma^{i-1}r_{t+i}+\gamma^{K-1}V_\phi(s_{t+K})-V_\phi(s_t)$

3. 分布式PPO的评估

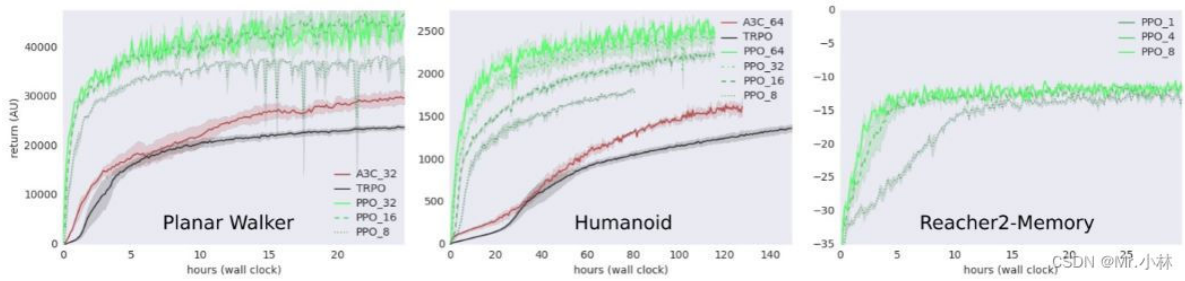
为了确定DPPO算法能够在有限的参数调整下进行稳健的策略优化，且有效的进行多线程扩展。通过选定线程数量与另外两种算法替代方案进行比较（TRPO 和 Continuous A3C），比较的基准任务环境有3种。其中两项是 **无障碍环境中的运动任务**，第三项是**需要记忆的平面目标到达任务**。任务的环境都是依赖于 Mujoco物理引擎 来搭建的。

双足机器人（Planar walker）： 具有9个自由度和6个扭矩驱动关节。 主要奖励获得是与其前进速度成正比，附加条款为惩罚控制和违反躯干高度和角度作为限制。 当双足机器人跌倒时，回合（Episode）提前终止。

人型机器人（Humanoid）： 具有 28 个自由度和 21 个切割关节。 主要奖励获得是受到一个主要与其沿 x 轴速度成正比的奖励，以及每一步的恒定奖励，连同跌倒时回合终止，鼓励它不要跌倒。



记忆机械臂（Memory reacher）：一个简单的 2 DoF 机械臂被限制在平面上。为回合的前 10 步提供目标位置，在此期间不允许移动手臂。 当允许手臂移动时，目标已经消失，必须依靠 循环神经网络 RNN （Recurrent neural networks, RNN）记忆才能使手臂到达正确的目标位置。 该任务的奖励是末端执行器和目标位置之间的距离，选择该任务能够测试 DPPO算法 优化循环神经网络策略的能力。



在三个任务环境中的测试**结果**分析：
由上图可以看出，DPPO算法 能够在性能上与TRPO算法相似，并随着 线程 的拓展，可以明显 减少训练的收敛时间 和 获得更高的 回报值。且因为PPO算法是基于策略梯度算法的，可以直接使用循环神经网络（RNN）来拟合策略函数，在需要记忆的记忆机械臂任务环境中也能适用。

三. 方法：环境和模型

本论文的研究目标是使用基础、简单的奖励函数 能否 让智能体从不同的挑战和一系列的难度水平的学习中掌握复杂的运动技能。在第二节阐述的3种基准测试任务环境中验证可拓展DPPO算法的适用性后。通过一些环境上的设置来演示更复杂行为的出现。

1. 训练环境

为了让智能体面临各种各样的运动挑战，使用类似于平台游戏的物理模拟环境并利用 Mujoco物理引擎 实现。会程序化的生成大量具有各种障碍物的不同地形；并且每个回合生成的地形和障碍物实例都不同。

身体
考虑的智能体有三种不同的扭矩控制体，分别为 双足机器人（Planar walker）、人型机器人（Humanoid）、四足机器人（Quadruped）；其中 双足机器人和人型机器人已经在第二节解释过，这边主要针对 四足机器人 进行介绍。

四足机器人（Quadruped）：一个简单的三维四足动物身体，具有 12 个自由度和 8 个驱动关节。

奖励

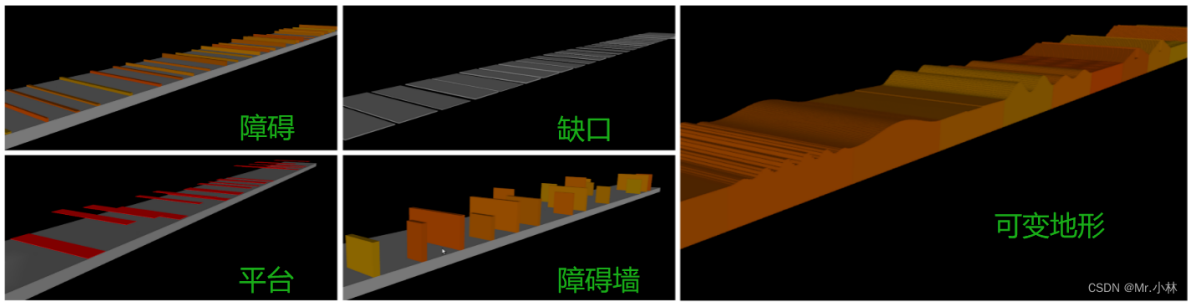
本次研究将保持所有任务环境的奖励函数简单且跨地形一致。 奖励包括一个与 X 轴上的速度成正比的主要部分，为鼓励智能体沿着轨道前进，奖励函数还加上一个小项惩罚扭矩。

对于双足机器人（Planar walker），奖励函数还包括与第二节中相同的姿势框约束。对于四足机器人（Quadruped）和人型机器人（Humanoid），奖励函数加上惩罚偏离轨道中心的行为，并且人型机器人在每个时间步长如果没跌倒还能获得额外奖励。

通过对比三种不同的 身体结构，直观能看出 它们的物理构造是不同的、有差异的。对于它们的身体间奖励函数的差异是在实验调整之前提出的，并没有特别的仔细调整。因此 奖励函数在不同身体结构的智能体上会略有不同，但没有对奖励函数进行 差异性巨大的改变 来 导致 引发单个身体的不同行为。

地形和障碍物

为了让 智能体从不同的挑战和一系列的难度水平中学习，在每个新的回合内 任务环境的地形和障碍物都会根据 智能体性能的统计数据 重新生成。考虑的地形和障碍物类型有：



障碍：需要跳过或爬过的高度和宽度可变的类似障碍物。
缺口：地面上必须跳过的缺口。
可变地形：具有不同特征的地形，例如斜坡、缺口、丘陵等。
障碍墙：形成需要绕行的障碍物的墙。
平台：悬停在地面上方的平台，可以跳上或蹲下。

本论文针对上述地形和障碍物来设置 环境 对 智能体进行 训练学习复杂动作技能。环境的设置 可以选择的方法有两种。分别为：
① 单一类型障碍物的环境（每个回合内环境的障碍物都是相同的）。
② 混合类型障碍物的环境（每个回合内环境的障碍物不止一种）。
同时考虑 每个回合的障碍物数量相同 来使得回合的长度（步长）相同，以及 地形的难度随着智能体性能增加而逐渐增加障碍物数量 来使得回合的长度变长，这两种环境设置。

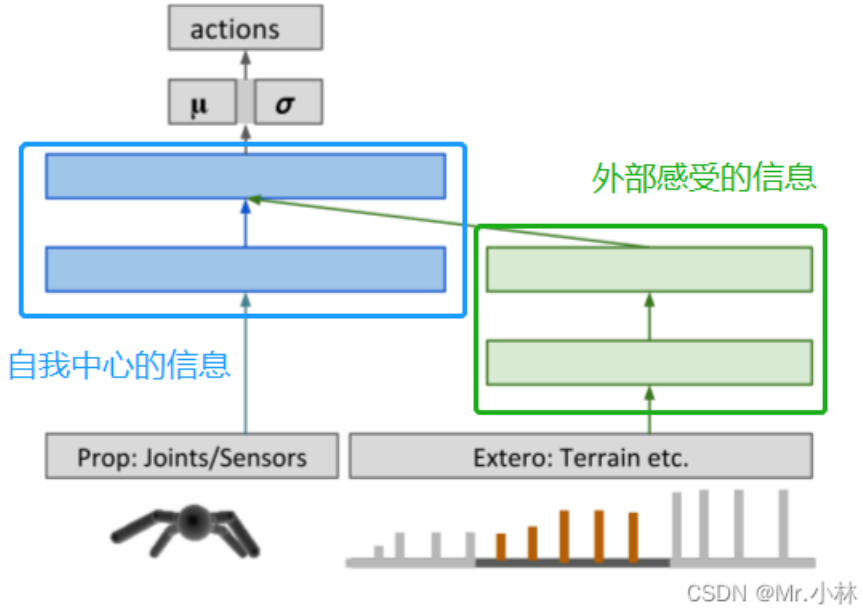
观测（Observations）
智能体的环境中会收到两组观测结果：
① 来自本体的特征：主要包含 关节角度和角速度，并且对于四足机器人、人型机器人还包含躯干的速度计、加速度计和陀螺仪的读数，提供以自我为中心的速度和加速度信息，以及连接到脚和腿的接触传感器。人型机器人的下肢关节也有扭矩传感器能够获取特征。
② 来自环境的外部特征：主要包括 相对于轨道中心的位置以及前方地形的轮廓。地形的信息通过沿 X 轴和 Y 轴与身体平移的采样点处获取的高度测量值数组的形式提供。双足机器人（Planar walker）因为被限制在X 轴、 Y 轴，所以导致不能左右移动而简化感知到的外部特征。

2. 策略函数参数化

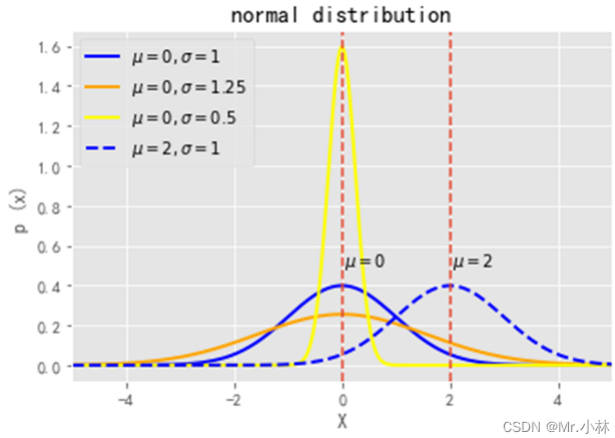
在构建参数化策略函数上，**目标**是深度神经网络能够对 来自本体的特征（基本的运动技能）与 来自环境的外部特征（地形感知和方向导航）进行关注点的分离。将策略构建为两个子网络，其中一个仅接收来自本体的特征，另一个仅来自环境的外部特征。

这种分离式神经网络架构 和 简单的全连接神经网络进行了比较，发现能够提高智能体策略的学习速度。

策略函数的神经网络架构为：



这边针对 μ （均值） 和 σ （方差）进行补充说明：
近端策略优化（Proximal policy optimization，PPO）算法是使用强化学习中的 随机性策略（Stochastic policy），前置假设认为动作概率分布符合正态分布（Normal distribution），输入 对于环境的观测，输出是所有动作的概率分布。



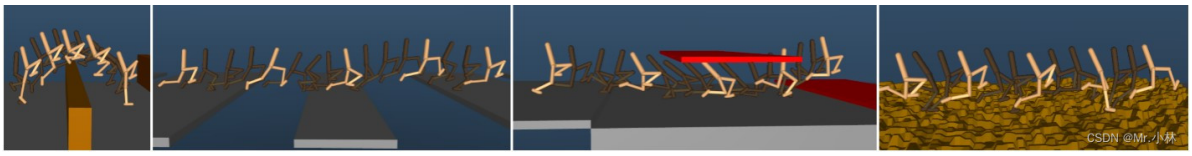
对于 策略函数 输出的 μ（均值） 和 σ（方差），通过 概率分布函数（Probability distribution function，PDF） 来构建 动作概率分布，并从中随机采样动作。

四. 结果

本论文将分布式PPO算法（Distributed PPO ）应用于各种类型的智能体、地形和障碍物。**目标**是确定当智能体在丰富的环境中接受训练时，简单的奖励函数是否会导致复杂运动技能的出现。 同时还对地形结构是否会影响学习成功和结果行为的鲁棒性感兴趣。

1. 双足机器人（Planar walker）的实验结果：

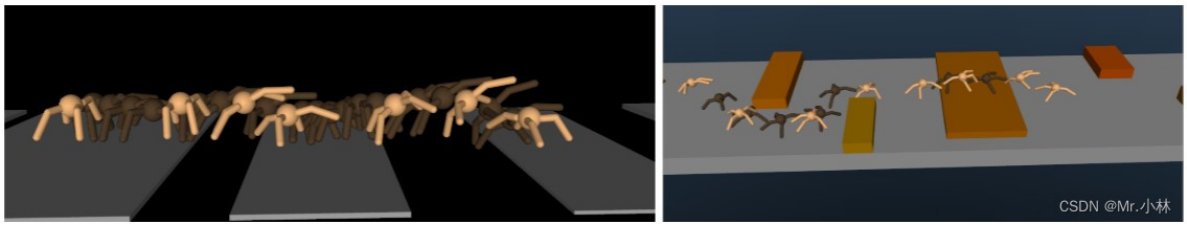
使用双足机器人分别在障碍、缺口、平台和可变地形上训练，在混合类型障碍物的环境中，智能体获得健壮的步态，学会了跳过障碍物和缝隙，学会走过平台或蹲伏在平台下。 所有这些行为都是自发出现的，没有特殊的奖励来诱导每个单独的行为。



上图表明 双足机器人 跳过障碍物、跳过缺口、蹲伏在平台下、穿越瓦砾场 的动作序列。并且在学习结束时，双足机器人能够跳过几乎和自己身体一样高的障碍。

2. 四足机器人（Quadruped）的实验结果：

四足机器人 相比于 双足机器人来说，身体结构是不灵活的。但是四足机器人也为控制问题增加了第三个维度。分别在可变地形、障碍墙和缺口和障碍地形的变体上训练，其中包含可以避开的障碍物，以及其他需要攀爬或跳跃的障碍物。 四足机器人也学会相当可靠地通过大多数障碍物。



上图表明 四足机器人 能够发现 向上或向前跳跃 是克服障碍和差距的合适策略，并且学会了在墙壁上穿行，适当地向左和向右转，并且是在只收到前进的奖励情况下学会的！四足机器人 能够根据 障碍地形的变化学会区分可以和/或必须越过的障碍物，以及必须绕过的障碍物。（在这种多变的地形上，对于四足机器人来说学习是非常困难的，因为 四足机器人 的腿与地形的变化相比比较短小）

3. 地形的性质影响智能体学习效能分析：

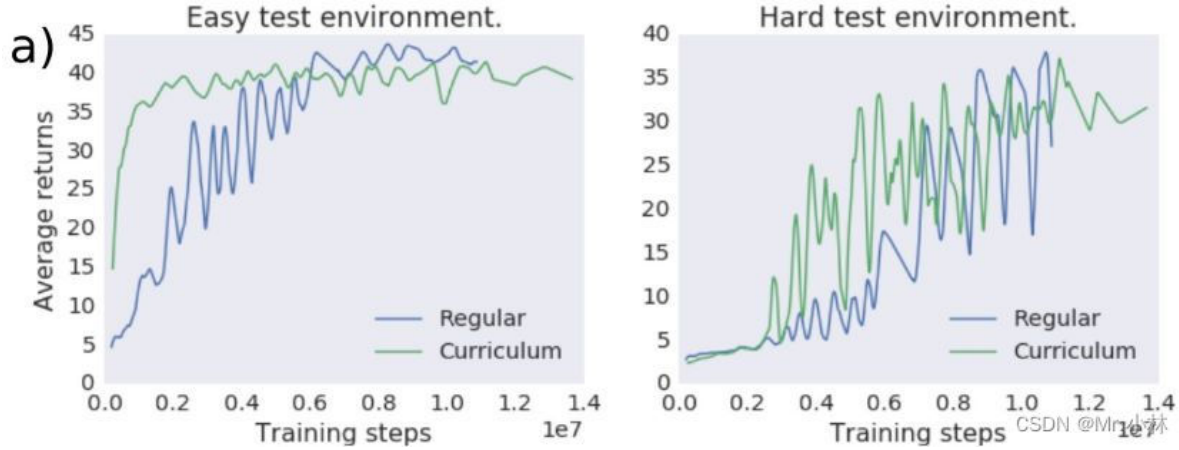
3.1 课程化的地形能够加快策略的学习

训练的任务环境的地形设置上，倘若一开始让智能体仅在 非常高的跨栏上 进行训练是不会有效的。且为了在设置中成功训练，会在地形上偶尔出现比较高的障碍，让双足机器人解决。

为了证明 在环境的地形上 设置让障碍的难度由浅到深难度逐渐增加，实验设置通过两种不同类型的障碍地形进行对比：

- 第一种是 拥有随机交错的高低障碍的 地形。
- 第二种是 障碍的难度由最小和最大高度决定，随着智能体的学习来通过课程的方式增加障碍难度的地形。

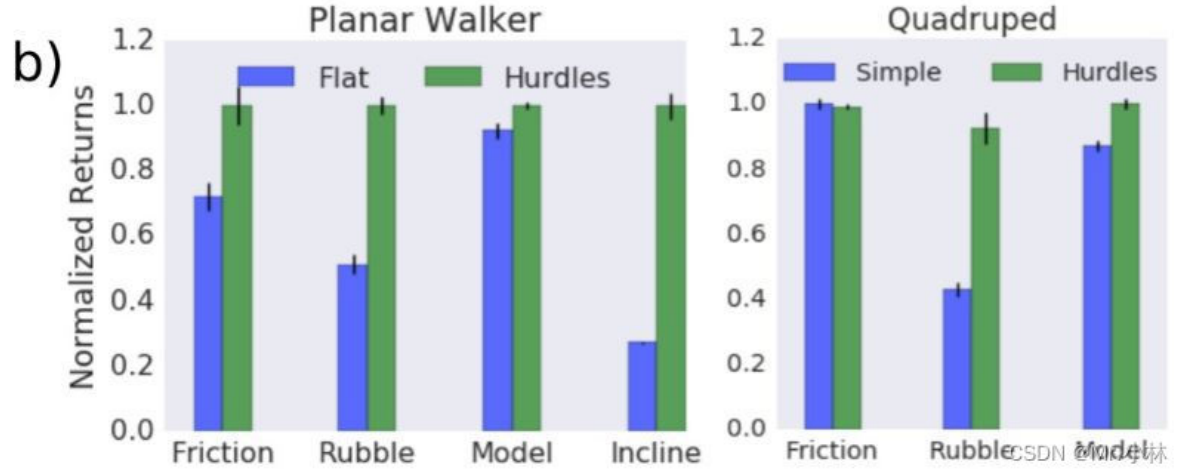
通过在两个测试地形上学习，期间评估策略来衡量学习进度，一个简单的障碍较浅，一个困难的障碍较高。



常规（Regular，蓝色线）环境 包含任意交叉的高障碍和低障碍。
课程（Curriculum，绿色线）环境 为难度逐渐增大的障碍。
上图表明，双足机器人（Planar walker）在难度逐渐增加的地形上训练的策略比在固定地形上训练的策略改进得更快。

3.2 障碍的地形训练的策略具有显著优势

评估在平坦地形（Flat）和 障碍地形（Hurdles）上训练对于智能体策略稳定性提高的优势。



使用两种智能体，双足机器人（Planar walker）和 四足机器人（Quadruped）分布在平坦地形和障碍地形上评估训练得到的策略的稳定性和鲁棒性。
Friction：未观察到表面的摩擦变化
Rubble：未观察到的杂乱地区
Model： 未观察到身体模型的变化
Incline：未观察到的倾斜/下降 地面
上图表明，在某些情况下，**在障碍地形上训练得到的智能体策略具有显著优势**，同时障碍地形也能增加智能体策略在未观察到地形某部分信息情况下的鲁棒性。

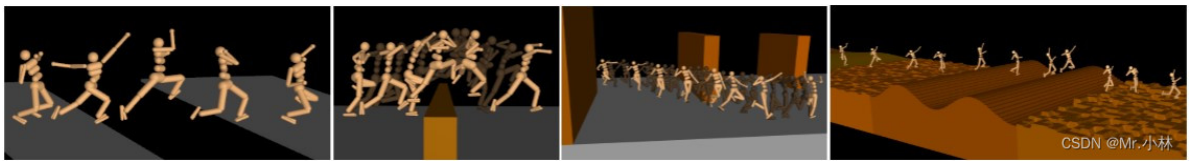
4. 人型机器人（Humanoid）的实验结果：

人型机器人 相比于 双足机器人、四足机器人 来说，在身体构造方面复杂的多。环境上 同样使用相同的混合地形 进行训练。

关于人型机器人的**奖励函数**的设计，选择沿 X 轴的速度成正比。
关于人型机器人的**终止条件**的设计，本论文试验了两种替代终止条件：

- 第一种为 当人型机器人的头脚之间的最小距离低于 0.9m 时终止回合。
- 第二种为 当人型机器人的头部和地面之间的最小距离低于 1.1m 时，终止回合。

在关于 人型机器人 的环境中，因为 机器人的关节较多，具有相对较大的自由度，在训练过程中智能体容易利用任务规范中的冗余和/或陷入局部最优，结果导致 机器人出现有趣但视觉上不令人满意的步态。



上图表明 在本论文的设置下，人型机器人在环境的每个地形都能够具有良好的表现，无论是在性能方面还是在步态视觉方面。

五. 相关工作

在强化学习领域，基于物理仿真任务环境的智能体是一个长期活跃的领域，也已经产生了大量相关的工作和令人印象深刻结果。但是基本上所有方法都依赖于对问题专业领域的重要先验知识和对动作数据的捕捉。

课程的概念，在机器学习领域的文献中已存在很久。但是在本论文的设计中，能够使用简单的 奖励函数和课程式训练在具有挑战性的环境中产生适应性动作，且同时仅对 策略和动作行为施加有限的影响，是一个新颖的研究方向。

六. 讨论和结论

本论文 探讨了在丰富的环境中训练智能体能够在何种程度上会导致出现不通过奖励函数能够实现直接激励复杂动作行为的问题。在控制问题中常见的设置是通过仔细设计奖励函数来实现特定的解决方案。

本论文 则相反，特意使用简单的、通用的奖励函数，在丰富的环境下训练智能体。一系列的实验表明，在不同的地形上进行训练可以促进智能体的动作行为的出现多样化发展，例如学会 跳跃、蹲伏和转弯等动作，但以此相对来说，针对特定动作来设计合理的奖励函数来引导智能体是不容易的。

因此通过设置 环境 实现 **课程式方法** 来让智能体学习，能够让智能体学到的动作策略更具有稳定性和鲁棒性。选择一个看似更复杂的环境实际上可能会让智能体的学习变得更容易。

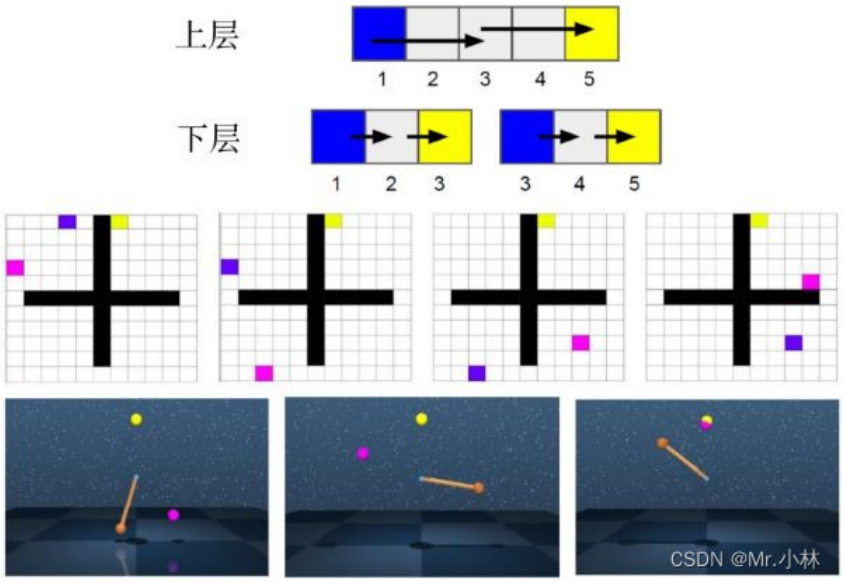
七. 贡献

在原先的强化学习任务中，为了让智能体在与动态环境的交互过程中更好的学习到优质的策略，通常的做法是设计能够正确评价智能体动作的奖励函数来引导智能体的行为。也基于此，奖励函数的设计需要对 环境有大量的先验知识。

但本论文则 提出对 动态环境 进行 **课程式方法** 的设计，来让智能体在简单的奖励函数下，逐渐根据策略性能的提高来学会更丰富的动作行为，这样能把 对于奖励函数设计上的注意力 转移到 对于动态环境课程式方法的设计上，当 面临某些任务的 奖励函数不好设计时，可以 参考本论文，尝试能否从 动态环境的课程式方法设计上来 解决 智能体面对复杂动作策略的学习问题。

八. 下一步的工作

针对于复杂的任务环境，本人觉得除了本论文提出的 对环境使用 **课程式方法** 引导智能体的学习之外，可以尝试使用 **分层强化学习** 的概念进行解决。



分层强化学习是指：“将一个复杂的强化学习问题分解成多个小的、简单的子问题，每个子问题都可以单独用马尔可夫决策过程来建模。这样，可以将智能体的策略分为高层次策略和低层次策略，高层次策略根据当前状态决定如何执行低层次策略。这样，智能体就可以解决一些非常复杂的任务”。------《蘑菇书EasyRL》

面对 本论文的 混合地形的设计，能否 针对不同地形进行 进行问题分解，来挖掘 不同地形的共性，来透过 低层次策略 与 高层次策略 共同对混合地形问题进行拆分，让智能体也能够学会稳定的、鲁棒的策略。

参考文献

1. Volodymyr Mnih, J. et al. 2016. Asynchronous Methods for Deep Reinforcement Learning.
2. 推荐中的序列化建模：Session-based neural recommendation
3. 论文十问-快速理解论文主旨的框架
4. RLChina 2022 强化学习暑期课
5. 蘑菇书EasyRL
6. GitModel统计分析
7. 动手学强化学习

作者：林俊杰

研究单位：台湾成功大学制造资讯与系统研究所

研究方向：强化学习、深度学习

联系邮箱：554317150@qq.com