基于深度强化学习的机械臂仿真投掷研究

摘要

本文提出一个基于 Sawyer 机器人的瓶子抛掷的 CoppeliaSim 仿真环境，并提出一个基于深度强化学习的机器人运动调整模型，给出几种情况下的抛掷仿真数据。

通过对赛题的分析，我们将其划分为三个子问题：

如何设计机器人？如何训练机器人使其抛掷瓶子至指定桌面并直立？不同情况下机器人的表现有什么差异？

在问题一中，我们使用 CoppeliaSim 仿真环境搭配 Bullet 物理引擎，置入 Sawyer 机器人、桌子，并将赛题中的瓶子抽象为参数可调的圆柱形刚体。通过 BØ-based remote API 使用 Python 控制机器人的运动。

在问题二中，我们基于 Pytorch 和深度强化学习平台天授，定义了一个全连接神经网络。使用蒙特卡罗方法生成每次仿真的初始状态，通过 DQN 强化学习算法，使用来自仿真环境的状态数据，训练前述神经网络，利用马尔可夫决策过程判断机器人行为的优劣并定义强化学习中的奖赏值。经过足够多次仿真与网络迭代，最终得到一个模型，可根据观测状态预测机器人下一步动作并判断抛掷的时机。

在问题三中，我们对抽象后的水瓶进行分析，考虑其体积、质量、角动量等参数。对其在被机器人抛掷出后的运动情况进行分析。同时考虑桌子与机器人的水平距离和高度差。将问题二中得到的模型加载为预训练模型，并基于修改过的参数进行训练，得到指定参数下的模型。

本文提出的方法基于仿真条件下的机器人运动，相较单纯的数学模型更加符合真实情况，在现实应用中具有更大的参考价值。同时基于深度强化学习的方法具有更强的鲁棒性，可以应对多样化的仿真环境，能有效解决赛题中包含的问题。同时，本文中的方法可以推广到有明确目标的机器人动作训练及优化任务。

一、问题重述

一、问题背景

随着科技的发展与进步，机械臂的应用场景十分广泛。目前在制造业、航天、服务业等多个领域已有众多应用场景与机械臂紧密相连。当前机械臂控制的主要方法是通过对机械臂运动学、动力学建模并求解。

相较于传统控制方法，近年来兴起的基于强化学习的控制方法越来越受到关注。传统控制方法需要通过复杂的建模分析、计算来求解多个运动学方程，并随着建模参数的增加，建模过程将越来越复杂；而基于强化学习的控制方法，只需设置好智能体（agent）、观测的状态（state）、智能体可执行的动作（action）、环境反馈给智能体的奖励（reward）等信息，结合强化学习算法，让智能体在场景环境中通过交互进行自我学习来完成任务，不需要复杂的建模过程，就能很好的完成任务[1]。

强化学习（RL）作为机器学习领域的研究热点，越来越多地受到关注。深度强化学习（DRL）是强化学习（RL）框架与深度学习（DL）算法的结合。它突破了之前强化算法在算法复杂度和扩展性上的限制，对于高维状态空间和高维动作空间的众多环境中也能产生很好的效果[1]。

CoppeliaSim 是一个具有集成开发环境的机器人模拟器，被广泛用于快速算法开发、工厂自动化模拟、快速原型制作和验证、机器人相关的教育、远程监控、安全双重检查等等。

二、需要解决的问题：

通过对赛题的分析，这是一个典型的设计问题。要求设计一款机器人，通过对机器人的调控，使之完成指定任务：机器人需要投掷一个瓶子，使得瓶子落到桌面指定位置，且满足正面朝上站立。

现要回答以下问题：

问题一、如何设计这样的场景，使得机器人能够进行投掷；

问题二、应该如何训练这样的机器人，使得它在投掷过程中能准确无误地投掷到指定位置且正面朝上；

问题三、如果改变参数，例如改变瓶子的大小，瓶中水的含量，机器人与桌子的远近等因素，对瓶子的投掷准确度产生的影响是什么，并绘制出瓶子运动的动态效果图。

二、问题分析

2.1 问题一的分析

问题一要求我们建立一个环境，使得机器人可以在这个环境下完成投掷过程。

机械臂是一种能模仿人手和臂的某些动作功能，用以按固定程序抓取、搬运物件或操作工具的自动操作装置。本题可以用机器臂代替机器人，对物体进行抛掷。搭建基于CoppeliaSim的仿真机器人环境场景，通过布局仿真实验里的机械臂，瓶子和桌面，进行仿真模拟现实机器人投掷。

2.2 问题二的分析

问题二要求建立一个模型，使得机器人能够准确无误得投掷到指定位置且正面朝上。

这里需要再建立一个强化学习训练场景，并连接强化学习场景与之前建立的仿真机器人环境场景。强化学习场景是指，机器人以“试错”的方式进行学习，通过与环境进行交互获得奖赏指导，目标是使机器人获得最大的奖赏。强化学习系统通过奖赏指导，得到行动评价，在行动评价的环境中获得信息，然后不断优化改进行动方案以适应环境。通过代码实现两个环境的对接。故有系统向机器人转递行为指导，控制机器人；仿真环境返回机械臂的交互信息，系统进行收集，并作出下一次行为指导的决策。

2.3 问题三的分析

问题三要求我们在投掷过程中对瓶子投掷准确度的影响因素进行分析，并且绘制出瓶子运动的动态效果图。

在实际投掷过程中，如果投掷的瓶子大小不一样，所受到的空气阻力则可能不同，因此用相同的模式，则不能完成指定的任务。同理，瓶中水的含量也将会直接影响投掷物的重量，机器人与桌子距离的远近，会直接影响我们投掷初速度的选取，同样会产生问题。对于此类种种问题，我们都可以通过在仿真过程中，对参数进行改变，在环境中得到数据，对所得数据进行分析得到。并在平台中加入Graph命令,可得到投掷过程的瓶子的三维运动轨迹图和瓶子在不同参量下的动态效果图。

三、模型假设与约定

1、假设瓶子是规则的、质量分布均匀的
2、假设瓶子、桌子、机器人均处于同一真空的惯性参考系中。
3、假设上述参考系的仅考虑刚体碰撞、均匀的重力作用、各刚体接触时均有合适的摩擦力。
4、假设机器人的运动为理想运动，即对于一个指定动作，只需要一次运动。
5、假设机械臂的后续动作不会影响瓶子的投掷轨迹

四、符号说明

符号名	含义
S	状态集
A	一组动作
P_{sa}	状态转移概率
R	回报函数
γ	折扣因子
mt	移动均值
Q	神经学习网络 Q 值
f	随机标量函数
g_t	随机标量函数梯度
vt	平方梯度
β_s	衰减率
$\hat{m}_t / \sqrt{\hat{v}_t}$	信噪比

五、模型的建立与求解

5.1 问题一的模型建立与求解

5.1.1 问题一的分析

问题一要求我们建立一个环境，使得机器人可以在这个环境下完成投掷过程。

我们希望建立一个有机器人，水瓶，有桌面的一个仿真环境。而机械臂是一种能模仿人手和臂某些动作功能的自动操作装置，因此本题用机器臂代替机器人，对物体进行抛掷。搭建基于 CoppeliaSim(旧版为 V-REP)的仿真机器人环境场景，通过布局仿真实验里的机械臂，瓶子和桌面，进行仿真模拟现实机器人投掷。

5.1.2 CoppeliaSim 机器人仿真环境

CoppeliaSim 是一个具有集成开发环境的机器人模拟器，它具有分布式控制体系结构的优点，每个对象/模型都可以通过嵌入式脚本、插件、ROS 或 BlueZcro 节点、远程 API 客户机或自定义解决方案进行单独控制。CoppeliaSim 通用性很强，支持多种编程语言和超过 400 种不同的应用编程接口函数，适合众多机器人应用。

CoppeliaSim 被广泛用于快速算法开发、工厂自动化模拟、快速原型制作和验证、机器人相关的教育、远程监控、安全双重检查等等，在众多行业有一定的认可度[1]。

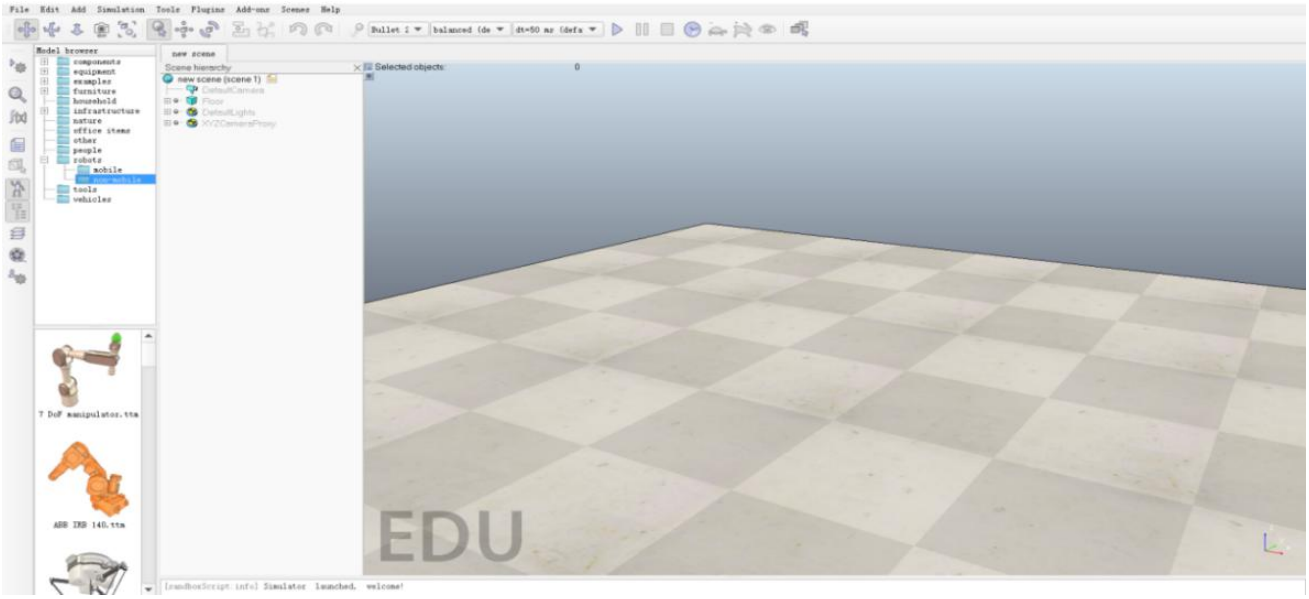

图 1 CoppeliaSim 初始环境

5.1.3 sawyer 机器臂

本模型采用的仿真机器臂为 Sawyer 机器臂， 基于德国工程能力和多年应用经验的结合，Sawyer 机器臂提供了安静的工作环境，同时因为拥有友好面孔，在员工中更受欢迎。基于具有 7 个自由度和一个 1260 mm 范围的机器人手臂的特点，Sawyer 可以用于没有空间供人类员工使用的地方。可能的应用领域包括对于人类危险或单调的任务，如数控机总成、电路板组装、金属加工、注塑、包装、装卸以及测试和检查。

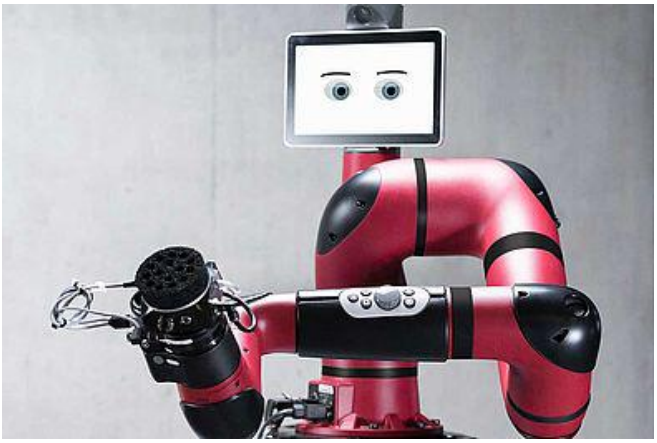

图 2 Sawyer 机械臂

在软件驱动硬件方面，易于应用。协作机器人制造商，提供了一个强大的易于使用的软件-Intera 软件，并实时提供生产指标的 cobot 解决方案。同时 Sawyer 在其手臂上配备了嵌入式 Cognex 视觉系统，使机器人定位系统 （RPS） 能够为机器人提供动态重新定位和轻松重新部署[2]。

5.1.4 Bullet（物理引擎）

Bullet 是一个开源的物理模拟计算引擎，世界三大物理模拟引擎之一，广泛应用于游戏开发和电影制作中。大部分的物理引擎都是通过 CPU 完成物理模拟计算，目前仅有 nVIDIA PhysX 可以调用 GPU 完成物理模拟计算且将其实用化。GPU 有着 CPU 无法比拟的并行计算和浮点计算能力，而复杂的物理模拟计算（例如流体模拟和柔性物体模拟）却十分依赖并行计算能力和浮点计算能力。在与 AMD 合作后，Bullet Physics 物理引擎可以透过 OpenGL 或者 DirectCompute，使用 GPU 完成物理模拟计算，这也是 AMD 开放物理计划的内容之一[3]。

5.1.5 仿真环境场景搭建

本模型设置的任务场景为一个投掷场景。包含的 3 个物理组件分别为：sawyer 机械臂（投掷装置）、模拟瓶子的圆柱体（被投掷物）和桌面（其中桌面上的一个指定位置为投掷目标）。将本场景的任务设置为：通过机械臂，将圆柱体投掷到桌面的指定位置。

首先按照任务要求，我们需要在场景中加载出仿真软件自带的 sawyer 机械臂、桌子。同时需要在 Coppeliasim 中新建一个紫色圆柱体的模型。我们将桌子放置在平面的一边，将 sawyer 机械臂放置在与桌面相对的另一边，同时保证机械臂的第二个关节与桌面相对，以保证最后投掷动作的顺利进行。在调节好圆柱体的尺寸，保证其在机械臂夹子的夹取大小范围内。将圆柱体与机械臂夹子下方的力传感器相连，使机械臂在正常运动过程中能够夹住圆柱体。

步骤一、加载好模型中的物理组件

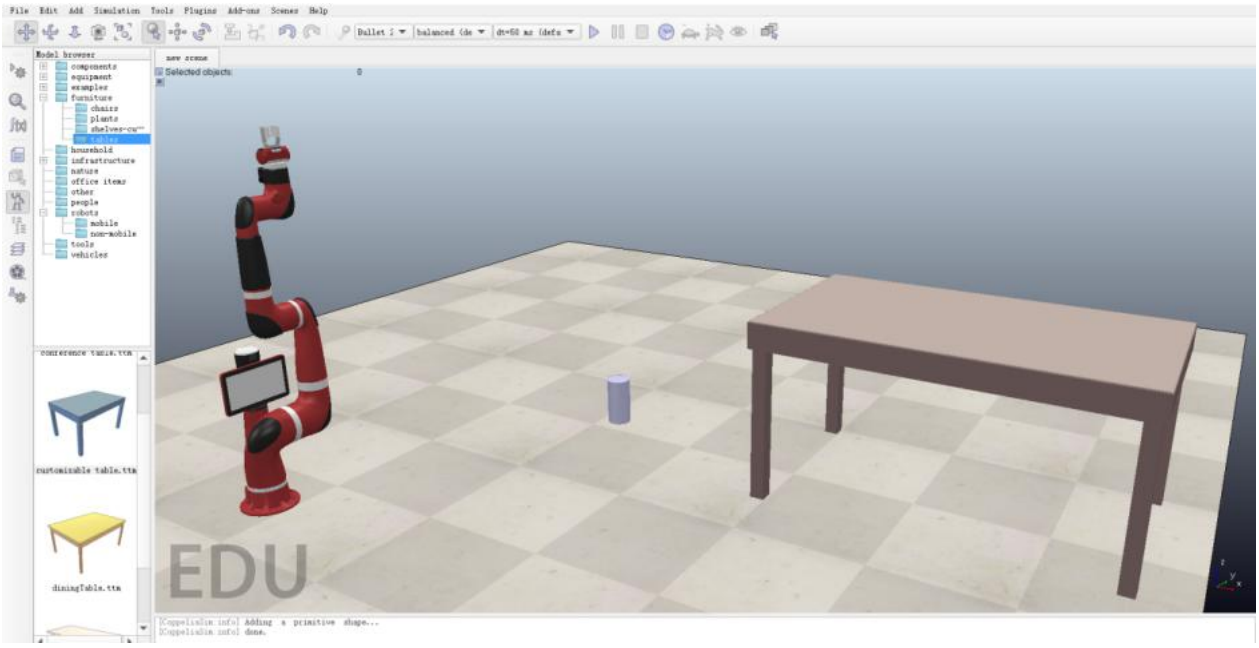

图 3 仿真环境下的物理组件

步骤二、布置场景中的组件位置

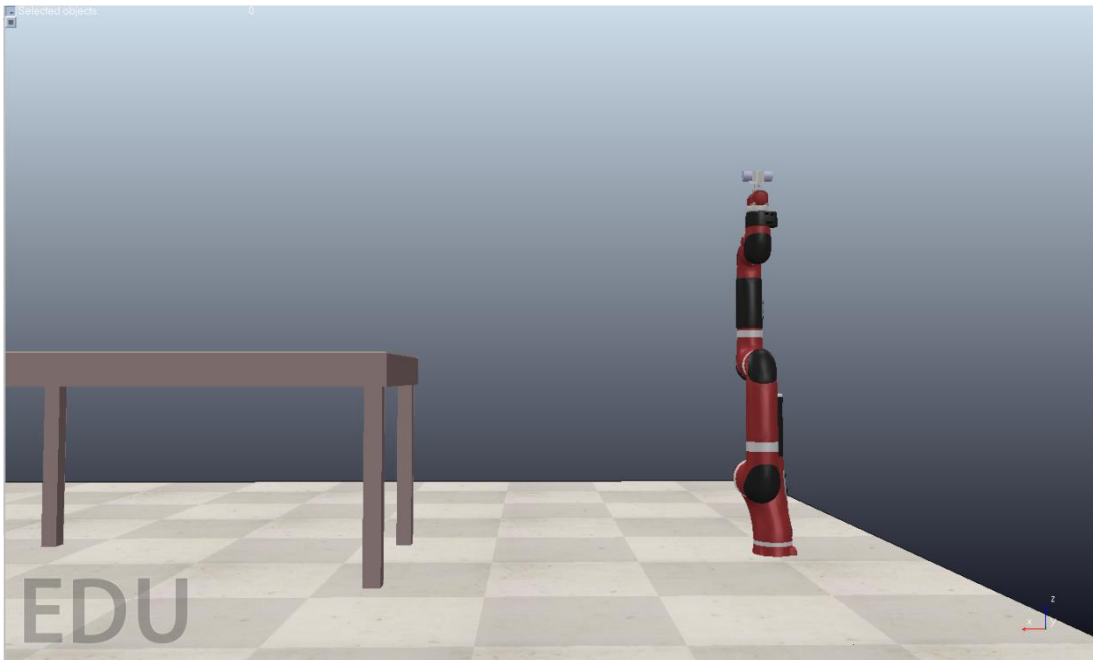

图 4 搭建好的仿真环境

可以观测到，机械臂与桌面位于同一水平面上；机械臂的夹子有将圆柱体控制住，且没有掉落；机械臂与桌面有一定的距离，是满足投掷的任务要求的。

为了更加全面的观测机械臂与桌面的分布，我们打开不同的视图，以确保准确性。

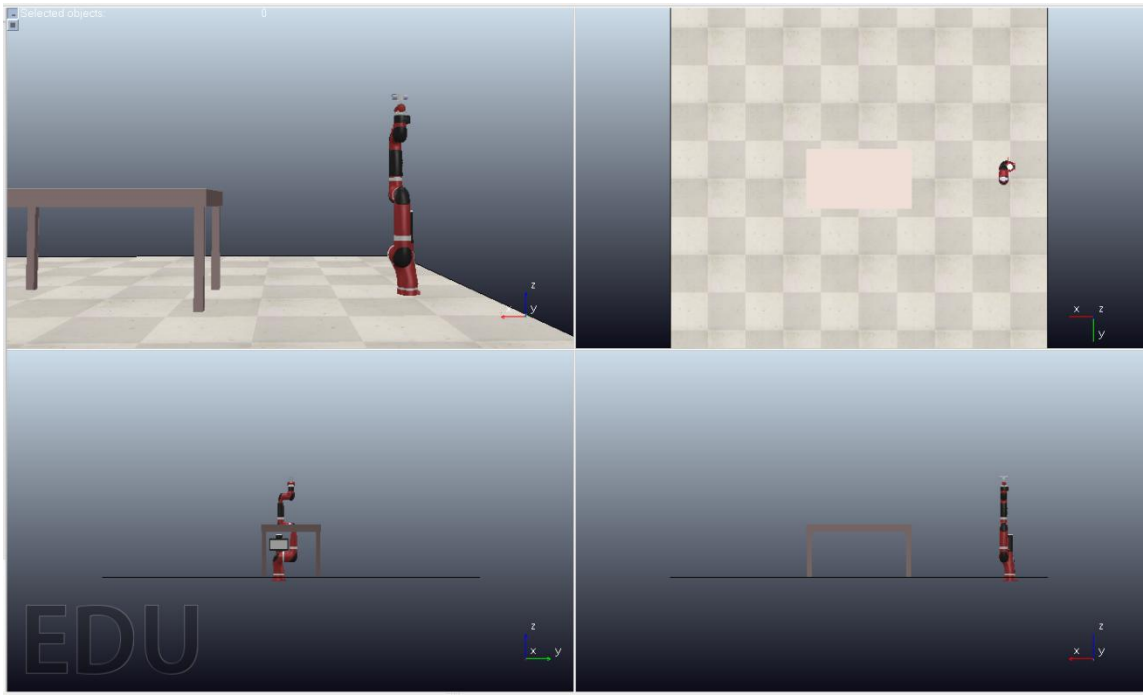

图 5 不同视图下的仿真环境

5.1.6 使用 python 控制仿真机器人

为了更加便捷地操作仿真机器人，同时使对机器人行为的深度强化学习成为可能，

我们需要使用 Python 来控制仿真机器人的行为。CoppeliaSim 提供 Legacy remote API 和 BØ-based remote API 两种远程控制 API。在经过比较测试后，我们选用 BØ-based remote API 作为连接 API。作为 Legacy remote API 的更新版本，BØ-based remote API 更加灵活、易用，且拓展性强。同时后者作为 CoppeliaSim 框架的组件之一，可以随程序启动而启动，不受当前场景（Scene）影响地保持连接，而前者只能在仿真开始后连接，极大的增加了仿真计时的不确定性和性能损耗。

为了方便后续问题的求解，我们使用 OpenAI 开发的基于 python 的强化学习环境框架 gym 建立了一个基于仿真环境的强化学习环境（env）。我们从源代码编译并在 python 工程中部署了 BØ-based remote API 的客户端。强化学习环境初始化时，连接 CoppeliaSim 的 BO Remote API server 得到一个 Client 句柄，我们通过该句柄获取 CoppeliaSim 中的各种状态，并实时操作机器人。通过对前述仿真环境的分析，我们认为只需要考虑以下环境参数：

x	y	z	α	β	C_{floor}	C_{desk}
瓶子 x 坐标	瓶子 y 坐标	瓶子 z 坐标	瓶子 a 轴角度	瓶子 b 轴角度	是否与地面碰撞	是否与桌子碰撞

表 1 环境参数

该强化学习环境包括一个观察指令，用于获得上述参数的实时数值。
通过对前述仿真环境的分析，我们认为只需要对机器人发出以下指令：

r_2	r_4	r_6	G
机器人转轴 2 角度	机器人转轴 4 角度	机器人转轴 6 角度	机器人抓手状态

表 2 发出的指令

该强化学习环境包括一个动作指令，用于传递上述参数给仿真环境中的机器人。
至此，我们可以在 python 环境中操控机器人并获得仿真环境中的数据[4]。

5.2 问题二的模型建立与求解

5.2.1 问题二的分析

问题二需要建立一个强化学习训练场景，并连接强化学习场景与之前建立的仿真机器人环境场景。强化学习场景是指，机器人以“试错”的方式进行学习，通过与环境进行交互获得奖赏指导，目标是使机器人获得最大的奖赏。

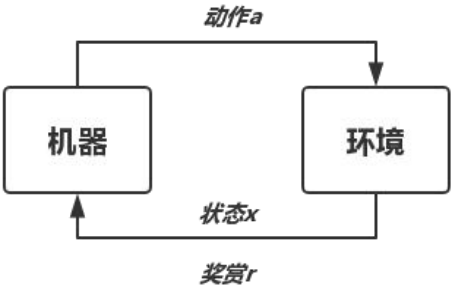

图 6 强化学习图示

强化学习系统通过奖赏指导，得到行动评价，在行动评价的环境中获得信息，然后不断优化改进行动方案以适应环境。本模型我们运用了马尔可夫决策过程、同策略蒙特卡罗方法、DQN 深度强化学习算法使机械臂进行强化学习，同时使用 Adam 算法进行优化。

系统向机器人转递行为指导，控制机器人；仿真环境返回机械臂的交互信息，系统进行收集，并作出下一次行为指导的决策。

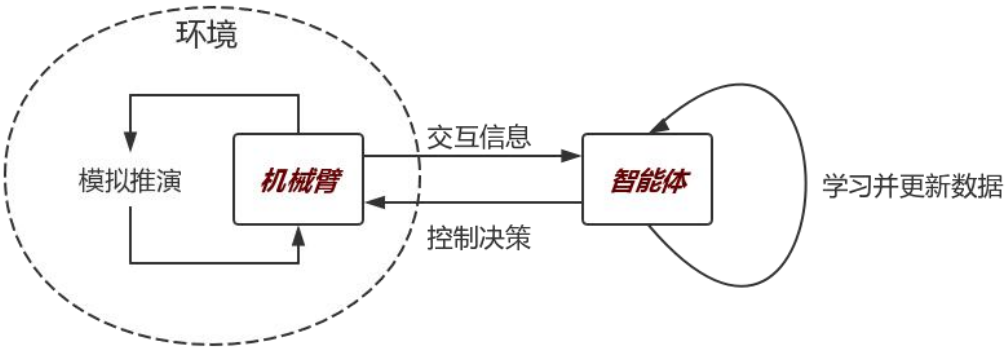

图 7 环境与智能体的关系

5.2.2 马尔可夫决策过程

为了让机器人能够正确投掷瓶子到指定位置且保持瓶子的正立，我们决定使用强化学习来训练 sawyer 机器人。运用马尔可夫决策过程（Markow Decision Process,简称 MDP）来描述强化学习的任务。

一个马尔可夫决策过程由一个元组构成 $M=<S,A,P_{sa},R,\gamma>$[5]

根据马尔可夫决策过程，结合本次题目，设置出以下学习模型：

其中 a_1=按照当前运动轨迹再投掷一次； a_2=调整机械臂运动轨迹再投掷；

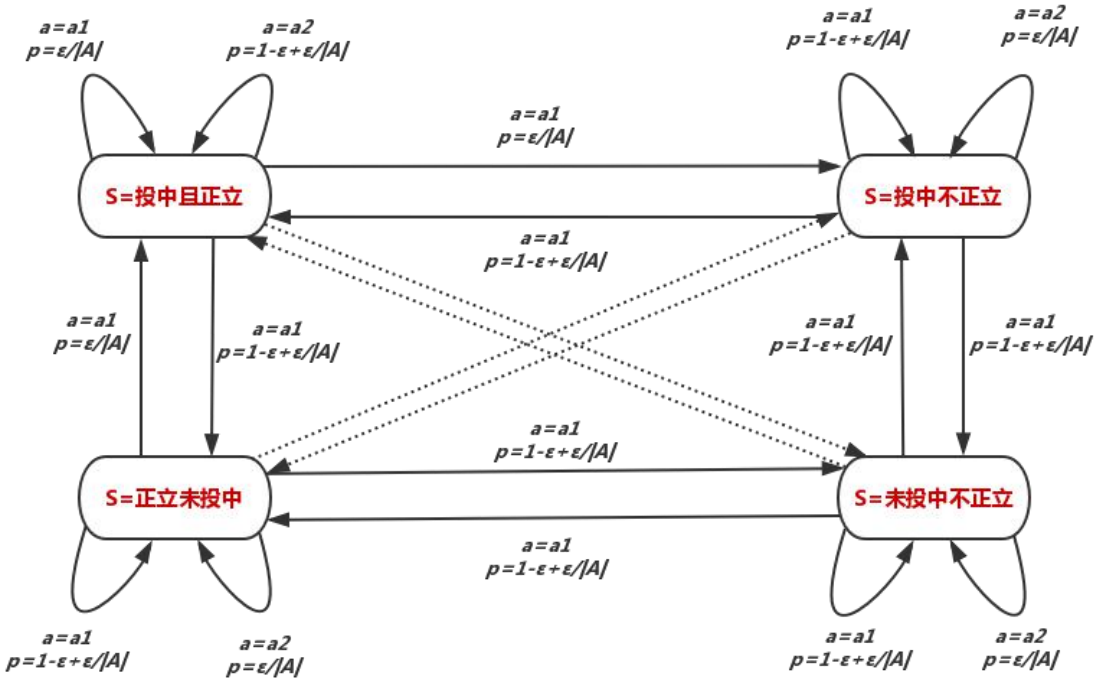

图 8 马尔可夫决策过程

补充：虚线部分

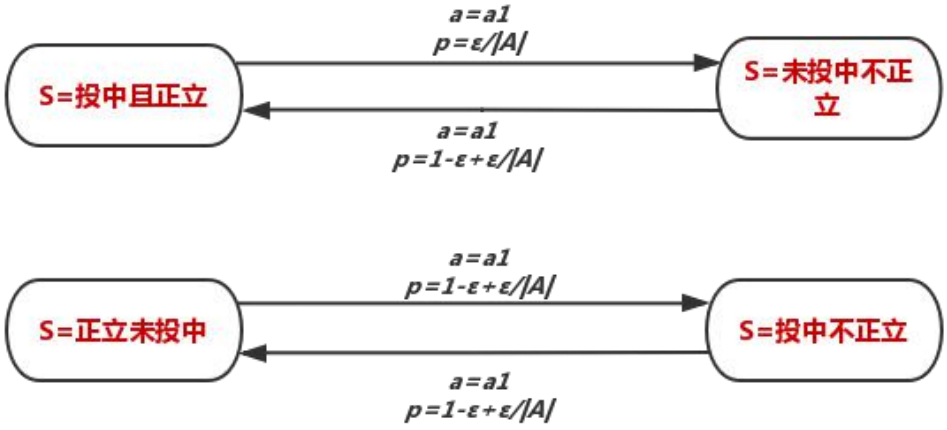

图 9 补马尔可夫决策过程

如图所示，在这里我们的任务有四种状态（投中且正立，投中不正立，正立未投中，未投中不正立）和两个动作（按照当前运动轨迹再投掷一次，调整机械臂运动轨迹再投掷）。对于所选取的概率，我们在 K 摇臂老虎机的基础上进行改进。采用将探索与利用进行折中，在原始的确定性策略π（我们将其称之为“原始策略”）的基础上使用ε-贪心算法，并记为：

$$\pi^{\varepsilon}=\begin{cases}\pi(x), & \text{以概率}1-\varepsilon\\ A\text{中以均匀概率选取的动作}, & \text{以概率}\varepsilon\end{cases}$$

在 ε -贪心策略 π^{ε} 中，当前最优动作被选中的概率是 $p=1-\varepsilon+\frac{\varepsilon}{|A|}$，其余每个非最优动作被选中的概率是 $p=\frac{\varepsilon}{|A|}$。[6]图中的箭头表示状态转移，箭头旁的 a,p 分别表示导致状态转移的动作以及转移概率。容易看出，最优策略在“投中”状态旁边选择动作“按照当前运动轨迹再投掷一次”，在其他状态旁选择动作“调整机械臂运动轨迹再投掷”。

对于奖赏值，我们对其进行了详细的划分，在后文的 5.2.6 中有完整的展示，在此处就不再赘述。

5.2.3 同策略蒙特卡罗方法

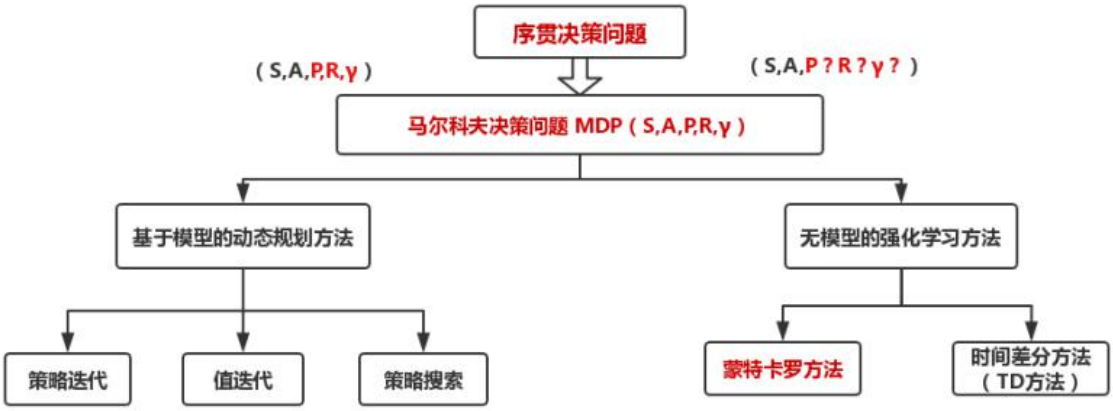

图 10 强化学习算法

如图所示，对于强化学习通过马尔可夫决策，往往会有两个分支：有模型学习和免模型学习（又被称为无模型学习）。在我们的强化学习任务中，状态和动作是经过合理推导与分析得到的，我们将其视为已知。但是对于概率和奖赏函数，我们并不知晓，因此我们决定使用无模型的强化学习方法。

在无模型的强化学习方法中有两种方法，我们在此选择使用蒙特卡罗方法。由于模型的部分变量未知，导致无法进行全概率展开，因此我们选择通过 sawyer 机器人在环境中执行的动作来观察转移的状态和得到的奖赏，经过多次的采样，我们把所取得的累计奖赏平均作为期望累计奖赏的近似，这种方法就是蒙特卡罗方法[6]。又由上文在马尔可夫决策图中展现，我们的策略只有一种，故我们所引用的是“同策略”蒙特卡罗强化学习算法，并用于生成单次仿真之前生成第一个动作。

5.2.4 DQN 深度强化学习算法

几年的深度强化学习算法的研究主要围绕 DQN 的相关研究和改进展开。DQN 将卷积神经网络与 Q 学习相结合，提出了深度 Q 网络（Deep Q-Network，DQN）模型，并引入

经验回放机制，使得计算机能够直接根据高维感知输入来学习控制策略[7]。

传统的 Q 学习算法，一般用 Q-Table 来存储状态-动作所对应的 Q 值，而如今的状态和动作都过于复杂，并且每种状态与动作对应的数据不同，则数据量会很大。如果仍然用表格存储的方法，会使得算法由于计算速度迭代缓慢，性能下降。将强化学习和深度神经网络结合，利用深度神经网络对 Q 值进行估计，代替 Q 学习中的 Q-Table 更新，由此这样可以解决连续高维的输入问题，即为 DQN 方法。

DQN 用估计函数的近似值来代替存储的 Q 值表，即：

$$Q(s,a)=f(s,a) \qquad (4.1)$$

如果是利用线性函数来估计，则如下式：

$$Q(s,a)=\omega_1 s+\omega_2 a+b \qquad (4.2)$$

公式（3.2）中，ω_1,ω_2,b 为需要训练调整的参数，这些参数用于将输入的状态与动作同 Q 值对应起来，使得满足 $Q(s,a)\approx f(s,a)$

DQN 算法的核心在于如何训练神经网络的函数 f，使得 $Q(s,a)\approx f(s,a)$ 即如何得到神经网络中的参数。

Q 学习的神经网络被称作 Q-Network，即 Q 网络。神经网络为监督学习，故需要大量被标签过的数据样本。在 DQN 算法中，标签是通过 Q 学习计算的 Q 值，加上输入状态与动作，成为 Q-Network 的样本，使用反向传播与梯度下降法来调整 Q-Network 中的参数[8]

故有 DQN 的公式为：

$$Q=(s,a)=Q(s,a)+\alpha[r+\gamma\max_{b\in A}Q'(s',b)-Q(s,a)] \qquad (4.3)$$

由此可得，DQN 的损失函数为

$$\mathrm{L}(\omega)=E[(Q_t-Q(s,a;\omega))^2] \qquad (4.4)$$

公式（3.4）中 Q_t 表示模型的目标 Q 值，为：

$$Q_t=r+\gamma\max_{b\in A}Q'(s',b;\omega) \qquad (4.5)$$

即有 DQN 算法模型图如下：

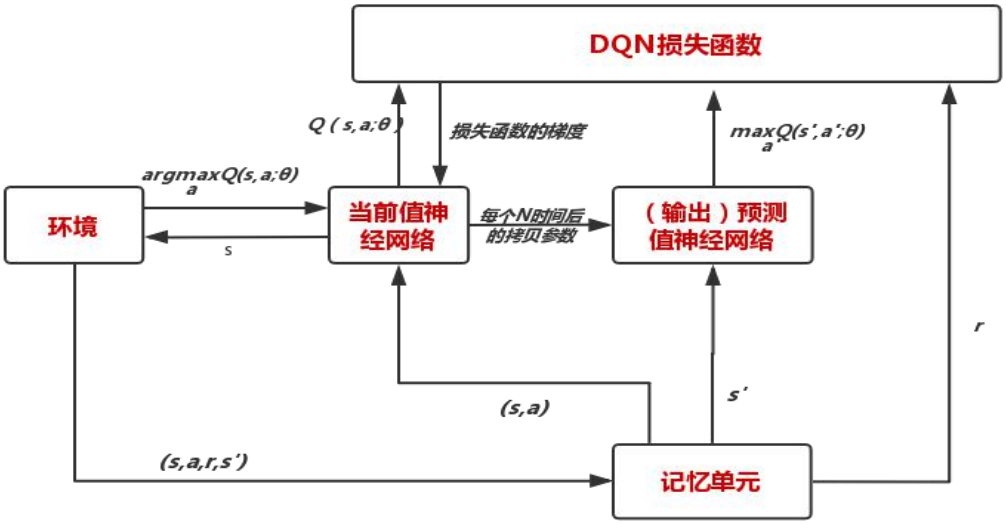

图 11 基于 DQN 的算法模型图

5.2.5 Adam 算法

在进行模型的优化时，我们使用 Adam 算法。我们在使用和学习 Adam 算法中援引了 OpenAI 的 Diederik Kingma 和多伦多大学的 Jimmy Ba 他们的 2015 年在 ICLR（深度学习的顶级会议）发表了一篇名为“Adam: A Method for Stochastic Optimization”的论文[9]。

它是一种基于低阶矩自适应估计的随机目标函数一阶梯度优化算法。Adam 的一些优点是，参数更新的大小对梯度的重新标度是不变的，其步长近似受步长超参数的约束，不需要平稳目标，使用稀疏梯度，自然地执行一种步长退火形式。

$f(\theta)$ 是一个嘈杂的目标函数：即关于参数 θ 可微的随机标量函数。目的是为了减少该函数的期望值，即对不同参数 θ，f 的期望值 $E[f(\theta)]$。其中 $f_1(\theta).....f_T(\theta)$ 表示在随后的时间步1.....T 上的随机函数值。

随机性可能来自随机子样本(小批量)的数据点评估，也可能来自固有的函数噪声。$g_t = \nabla_\theta f_t(\theta)$ 我们表示梯度，即在实践步骤 t 下 f_t 对 θ 的偏导数向量。

该算法更新梯度的指数移动均值 (mt) 和的平方梯度 (vt)，而参数 $\beta_1, \beta_2 \in [0,1)$ 控制这些移动均值的指数衰减率。移动均值本身是梯度的一阶矩(平均值)和二阶矩(无中心方差)的估计。然而，因为这些移动均值初始化为 $\vec{0}$，所以矩估计值会偏差向 0，特别是在初始的时间步中衰减率很小(即 β_s 接近 1)时会体现。

不过这可以很容易地抵消的初始偏差，因此我们可以得到偏差估计 $\hat{m}_t$ 和 $\hat{v}_t$。该算法

的效率可以通过改变计算顺序来提高，例如用以下几行替换算法中的最后三行：

$$\alpha_t = \alpha \cdot \sqrt{1-\beta_2^t} / \left(1-\beta_1^t\right) \tag{5.1}$$

$$\theta_t \leftarrow \theta_{t-1} - \alpha_t \cdot m_t / \left(\sqrt{v_t} + \hat{\varepsilon}\right) \tag{5.2}$$

除此之外，还要再介绍一下 Adam 算法的更新规则，Adam 算法的更新规则有一个重要的属性就是对它步长的选择十分谨慎。假设$\varepsilon = 0$，则在时间步t和参数空间所取的有效下降步长：

$$\Delta_t = \alpha \cdot \hat{m}_t / \sqrt{\hat{v}_t} \text{。} \tag{5.3}$$

有效步长有两个上界：在

$$\left|\Delta_t\right| \le \alpha \cdot \left(1-\beta_1\right) / \sqrt{1-\beta_2} \tag{5.4}$$

情况下有效步长上确界满足

$$\left(1-\beta_1\right) > \sqrt{1-\beta_2} \tag{5.5}$$

和在其他情况满足$\left|\Delta_t\right| \le \alpha$

第一种情况只发生在极其稀疏情况下才会发生:当梯度在除当前时间步长以外的所有时间步长都为零。对于较少稀疏的情况，有效步长会更小。当$\left(1-\beta_1\right) = \sqrt{1-\beta_2}$时，我们有$\left|\hat{m}_t / \hat{v}_t\right| < 1$

因此可以得到上确界：

$$\left|\Delta_t\right| \le \alpha \tag{5.6}$$

在更常见的情况下，由于：

$$\left|E[g] / \sqrt{E\left[g^2\right]}\right| \le 1 \tag{5.7}$$

我们会遇到这种情况：

$$\hat{m}_t / \sqrt{\hat{v}_t} \approx \pm 1 \tag{5.8}$$

每一个时间步的有效步长在参数空间中的量级近似受限于步长因子α，即$\left|\Delta_t\right| \lessapprox \alpha$

这可以理解为围绕当前参数值建立一个置信域，超过这个置信域，当前梯度估计就不能提供足够的信息。这可以令其相对简单地提前知道α正确的范围。

对于许多机器学习模型来说，我们知道最优解是在参数空间内的集合域上有极高的概率。例如我们可以在参数上有一个先验分布。因为α确定了参数空间内有效步长的量级（即上确界），我们常常可以推断出α的正确量级，而最优解也可以从θ_0开始通过一定量的迭代而达到。我们可以将$\hat{m}_t / \sqrt{\hat{v}_t}$称之为信噪比（SNR）。

信噪比越小，有效步长t越接近于零。这是一个理想的性质，因为较小的信噪比意味着对于的方向$\hat{m}_t$是否与真实梯度的方向相对应存在较大的不确定性。例如，SNR 值通常会趋近于 0，趋近于一个最优值，从而导致参数空间中更小的有效步长：即一种自动退火的形式。

有效步长Δ_t对梯度缩放来说是不变量：如果用因子c重缩放g，即相当于用因子c重缩放$\hat{m}_t$和用因子c^2重缩放$\hat{v}_t$。在计算信噪比时缩放因子会抵消得到下式：

$$\left(c\cdot\hat{m}_t\right)/\left(\sqrt{c^2\cdot\hat{v}_t}\right)=\hat{m}_t/\sqrt{\hat{v}_t} \qquad (5.9)[9]$$

5.2.6 模型求解

模型的实现使用基于 pytorch 的深度强化学习平台天授[10]，一个完整的模型包含环境（Env）、策略（Policy）、模型（Model）、采集器（Collector）、训练器（Trainer）等抽象模块。

此处的环境为第一问中定义的强化学习环境。

策略是强化学习算法的核心，将其形式化表示为

$$\pi_\theta:\left(o_t,h_t\right)\mapsto\left(o_t,h_{t+1},p_t\right) \qquad (6.1)$$

其中h_t是t时刻策略的隐藏层状态，通常用于循环神经网络（Recurrent Neural Network，RNN）的训练；p_t是t时刻策略输出的中间值，以备后续训练时使用。

此外不同策略在训练的时候所需要采样的数据模式不同，比如在计算 n 步回报的时候需要从数据缓冲区中采样出连续 n 帧的数据信息进行计算，因此策略需要有一个专门和数据缓冲区进行交互的接口。

策略中还包含模型（Model），包括表格模型、神经网络策略模型、环境模型等。模型可直接与策略进行交互，而不必和其他部分相互耦合。智能体除了需要做出决策，还需不断地学习来自我改进。

本模型中使用的策略为 DQN 强化学习算法。

定义一个神经网络：

	名称	类型	激活	可学习参数
1	fc_1	Linear	128	Weights 128*9 Bias 128*1
2	relu_1	ReLU	128	-
3	fc_2	Linear	128	Weights 128*128 Bias 128*1
4	relu_2	ReLU	128	-
5	fc_3	Linear	128	Weights 128*128 Bias 128*1
6	relu_3	ReLU	128	-
7	fc_4	Linear	4	Weights 4*128 Bias 4*1

表 3 定义的神经网络图

其中 Linear 为全连接层，定义如下：

$$y = xA^T + b \qquad (6.2)$$

ReLU 为线性整流单元，定义如下：

$$ReLU(x) = (x)^+ = max(0, x) \qquad (6.3)$$

常用的神经网络中的全连接层常通过设置大小不同的输入与输出来提取特征。对本神经网络而言，主要目的是储存特征，而不是提取特征，故除输入和输出层外的每个全连接层设置成相同的大小。

采集器定义了策略与环境交互的过程。让给定的策略和环境交互，并将交互过程中产生的数据存储进数据缓冲区中；采集器在多智能体强化学习的交互过程中，将不同的数据缓冲区和不同策略联系起来，即可进行交互与数据采样。

（天授还实现了异步采集器 AsyncCollector，它支持异步的环境采样（比如环境很慢或者 step 时间差异很大）。不过 AsyncCollector 的 collect 的语义和上面 Collector 有所不同，由于异步的特性，它只能保证**至少** n_step 或者 n_episode 地收集数据。）模型缺陷：使用同步采集器导致必须要等仿真环境响应后才能进行网络训练？

训练器负责最上层训练逻辑的控制，例如训练多少次之后进行策略和环境的交互。

现使用异策略学习训练器（Off-policy Trainer）。

强化学习的超参数设置如下：

网络参数的学习率 lr = 1e-3 #

参数最小调整的单位 eps_train, eps_test = 0.1, 0.05

num_epoch = 1000

每一个 epoch 执行的动作的数量 step_per_epoch = 100

每一组动作的数量 step_per_collect = 10

每一批观测值的大小 batch_size = 64

我们知道，在强化学习任务中，模型的训练依赖于与环境交互后获得的奖赏分数。因此，我们还需要定义一个奖赏函数 $Reward$，该函数由多个判断组成，抽象如下：

当一个动作完成时，得到本次动作完成时的时间与本轮仿真开始时间的时间差：

$$Reward_{time} = -\left(Time_{SimulationStart} - Time_{ActionStop}\right) \cdot \frac{(ms)}{1000}$$

作为一个负项，本项用于防止网络为避免触发其他扣分项而不进行动作。

$$Reward_{distance} = d(Position_{Goal}, Position_{BottleInitial}) - d(Position_{BottleInitial}, Position_{BottleCurrent})$$

该项计算水瓶目标位置对初始位置的距离与水瓶初始位置对当前位置的距离之差，用于限制网络的输出值过大。

$$Reward_{Gripper} = \begin{cases} 25, & 抛掷出去 \\ 0, & 未抛掷 \end{cases}$$

该项用于促使网络做出抛出的预测。

$$Reward_{Goal} = \begin{cases} 0 \\ 50, d(Position_{Goal}, Position_{BottleCurrent}) = 0 \\ 100, d(Position_{Goal}, Position_{BottleCurrent}) = 0 \cap \alpha = 0 \cap \beta = 0 \end{cases}$$

该项用于完成目标的奖赏判断。

$$Reward_{Action} = \left(\prod_{i=2,4,6} r_i\right)^{\frac{1}{3}}$$

该项计算每次动作中机器人每个节点运动的幅度的几何平均值，用于促使网络尝试输出更大的动作，与$Reward_{distance}$项相互拮抗。

定义一个函数 $Collision{:}X \to \boldsymbol{B}$：：

$$Collision(Entity_1, Entity_2)$$

用于判断仿真环境中的指定两个物体是否发生接触或碰撞。

$$Reward_{Collision} = -50 \cdot Collision(floor, bottle) - 50 \cdot Collision(sawyer, bottle) + 50 \cdot Collision(desk, bottle)$$

该项作为 $Reward$ 函数中的加权规则项，用于整体调节网络的训练方向，即避免瓶子与地面和机器人自身碰撞，尽可能使其与桌面接触。

综上，

$$Reward = Reward_{time} + Reward_{distance} + Reward_{Gripper} + Reward_{Goal} + Reward_{Action} + Reward_{Collision}$$

为本任务的奖赏函数。

定义了上述模组和参数后，在 pycharm 连接 Coppeliasim 下训练。每一次完整的训练大约耗时 8 小时。最后我们可以得到一个神经网络，对输入的观察状态给出其认为最佳的动作指令，并判断何时应该执行抛掷动作。

5.3 问题三的模型建立与求解

通过对本问题的分析，我们列出以下可控的环境变量：

符号	含义	缺省值
$\mathrm{Distance_x}$	桌面中心 x 坐标与机器人中心 x 坐标差值	$2.0000e+00$
$\mathrm{Distance_z}$	桌面中心 z 坐标与机器人底面 z 坐标差值	$7.0000e-01$
$r_2 \quad r_4 \quad r_6$	机器人转轴 2，4，6 的初始角度	$-9.000e+01 \quad +0.000e+00 \quad +0.000e+00$
mass	瓶子的质量	$9.079e-02$
radius height	瓶子的半径、高度	$3.4000e-02 \quad 1.0000e-01$
B	瓶子的初始角度	$0.0000e+00$
X Y Z	瓶子的重心坐标	$-1.919e+00 \quad -1.281e-07 \quad +1.326e+00$
$I=\begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{xy} & I_{zz} \end{bmatrix}$	瓶子的惯性张量	$\begin{bmatrix} +1.445e-04 & +1.445e-04 & +4.118e-10 \\ +3.006e-10 & +1.008e-03 & -1.703e-16 \\ +4.118e-10 & -1.981e-16 & +1.008e-03 \end{bmatrix}$

表 4 可控的环境变量

由于我们所使用的 CoppeliaSim 仿真环境只能对刚体进行仿真，故我们通过修改 $mass$、X、Y、Z、$\boldsymbol{I}$ 等参数来模拟瓶子中填充不同体积、密度的液体对投掷过程的影响。

使用缺省值训练的神经网络参数见附录一，示意图如下：

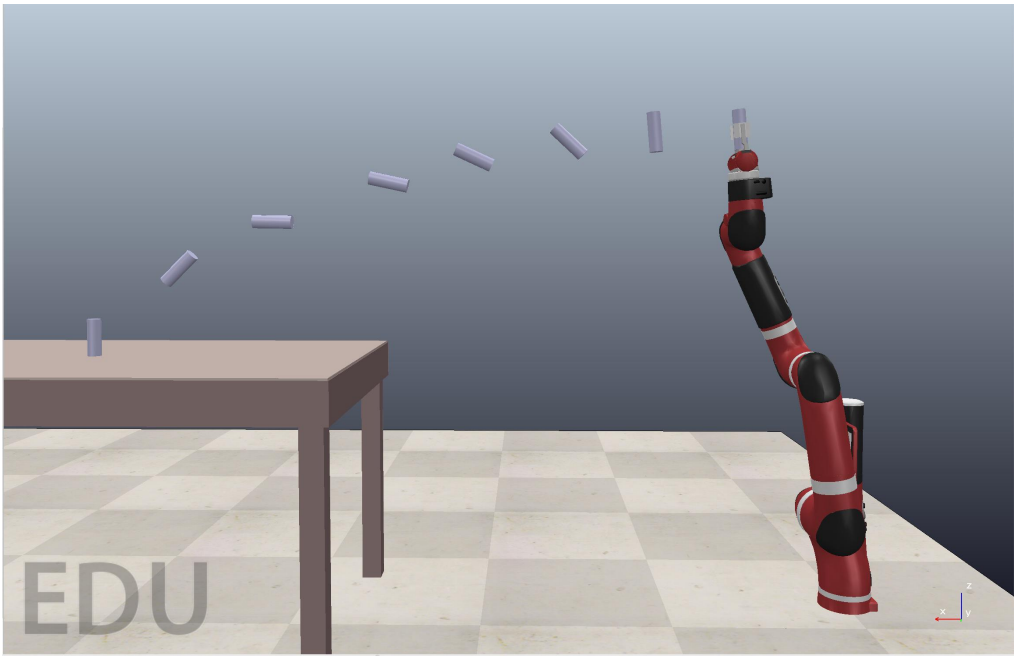

图 10 缺省值下的抛掷示意图

调整瓶子的初始角度B至 $9.0000e+01$，其他参数不变，在前述神经网络参数的基础上继续训练，得到的神经网络参数见附录二，示意图如下：

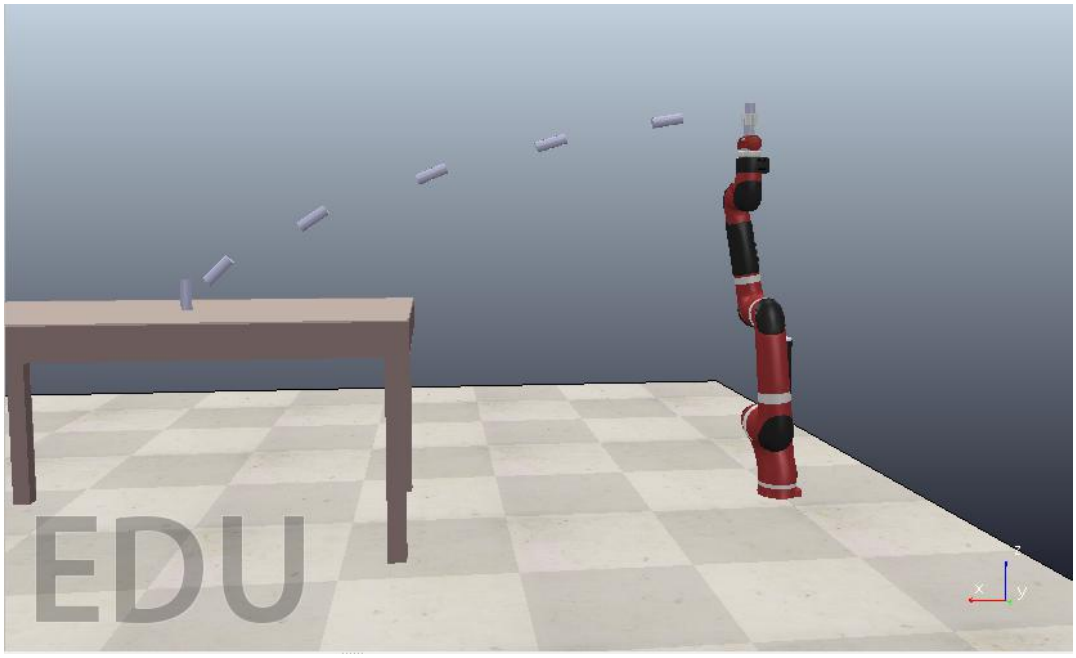

图 11 调整初始角度后的抛掷示意图

调整瓶子的质量 $mass$ 至 $2.500e-01$，初始角度B至 $9.0000e+01$，其他参数不变，在前述神经网络参数的基础上继续训练，得到的神经网络参数见附录三。

示意图如上：

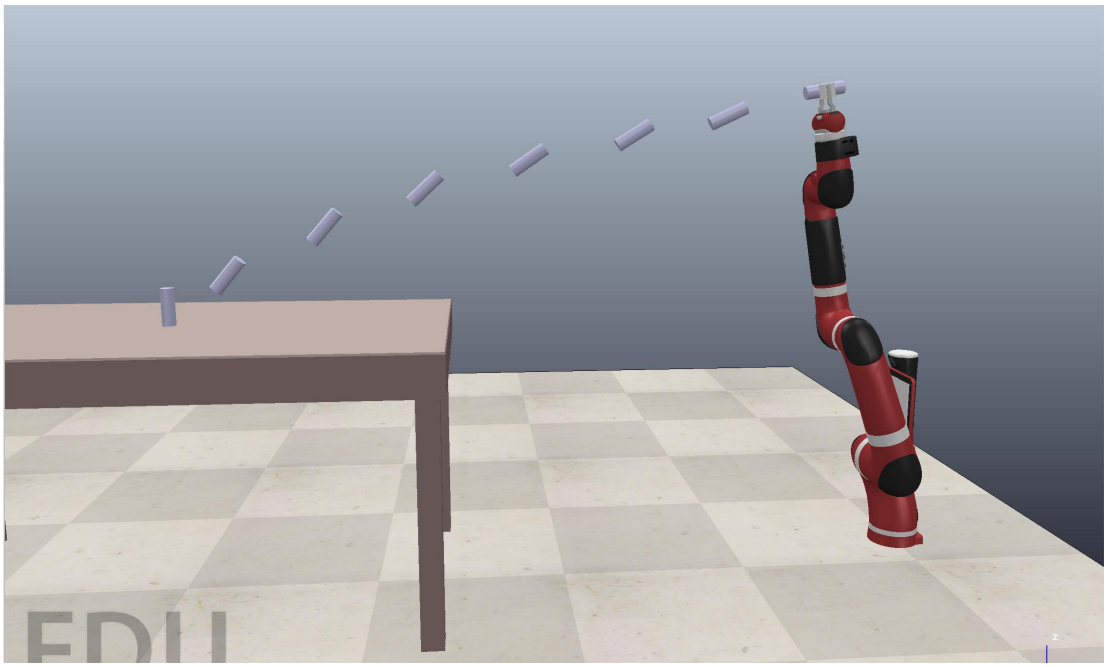

图 12 调整瓶子质量后的抛掷示意图

六、模型评价

6.1 模型的优点

1.我们使用了 CoppeliaSim 平台进行机器人投掷仿真，结果可靠、易观。除此之外，CoppeliaSim 平台支持六种编程语言进行编写，使我们的模型可以得到更加广泛的利用。

2.在机器人方面我们选择了 Sawyer 机械臂，它经历过时间的检验具有精度高，训练快等优点。

3.我们使用强化学习中的算法来训练模型，使模型能够更加准确，迅速完成目标。

6.2 模型的缺点

由于 bullet 物理引擎（Coppeliasim 平台内置的物理引擎）的限制，我们对水瓶参数修改不够充分。

七、模型改进

7.1 模型的改进

该模型只使用了 Sawyer 这一种机械臂，可以使用更多其它机械臂来使模型更具普适性。

7.2 模型的推广

该模型是在强化学习的算法基础上建立的，因此具有超强的学习能力。通过编写不同的代码，可以让它学习做到更多更难的任务，满足工厂和实验室不同的要求，具有现实意义。

八、参考文献

[1]余天洋. 基于强化学习的机械臂投掷问题研究[D].南昌大学,2020.

[2]https://www.rethinkrobotics.com/sawyer/, Sawyer BLACK Edition

[3]https://baike.baidu.com/item/Bullet/8198766?fr=aladdin, Bullet

[4]John Schulman.OpenAI Gym. https://arxiv.org/abs/1606.01540

[5]http://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

[6]周志华.机器学习[M].北京：清华大学出版社，2016 P384.

[7]赵星宇,丁世飞.深度强化学习研究综述[J].计算机科学,2018.

[8]吴子秋. M2M 通信中基于深度强化学习的中继选择算法[D].南京邮电大学,2020.

[9]Diederik P. Kingma,Jimmy Lei Ba.ADAM:A METHOD FOR STOCHASTIC OPTIMIZATION , URL：Published as a conference paper at ICLR 2015.

[10]https://github.com/thu-ml/tianshou

附录

附录一

神经网络的部分参数如下：

```
    fc1_weights ([[-0.2364,  0.0742,  0.0592,  ..., -0.1690, -0.0706,
-0.2684],
           [-0.2318,    0.4050,  -0.0686,    ...,  -0.0649,  -0.0194,
0.0645],
           [  0.2150,   0.0058,   0.2059,   ...,   0.1852,   0.1954,
0.4631],
           ...,
           [  0.2336,  -0.2635,  -0.0197,   ...,   0.0201,   0.1318,
-0.1780],
           [  0.3776,   0.2322,   0.0192,   ...,   0.2940,  -0.0961,
0.0851],
           [-0.0021,  -0.3864,   0.2662,   ...,   0.6351,  -0.0382,
-0.0591]])
    fc2_weights ([[-0.0590, -0.1602,  0.1077,  ...,  0.2020, -0.0206,
-0.2104],
           [-0.0268,  -0.0525,   0.0597,   ...,   0.0740,   0.0231,
-0.0647],
           [-0.0709,  -0.1733,   0.2503,   ...,   0.1049,   0.0335,
-0.3692],
           ...,
           [-0.0200,   0.0751,   0.0199,   ...,   0.0720,  -0.0719,
-0.0308],
           [-0.0800,  -0.0210,   0.1643,   ...,   0.0613,   0.1007,
-0.2700],
           [  0.0508,  -0.1683,   0.1525,   ...,   0.1387,  -0.0350,
-0.2223]])
    fc3_weights ([[-0.0149, -0.0153,  0.0273,  ...,  0.0826, -0.0073,
0.1119],
           [  0.0320,   0.0112,   0.0806,   ...,  -0.0492,   0.0212,
0.0362],
           [  0.0527,  -0.0208,  -0.0899,   ...,  -0.0149,   0.0512,
-0.0203],
           ...,
           [-0.0029,  -0.0805,  -0.0380,   ...,   0.0235,   0.0980,
0.0696],
           [  0.0087,  -0.0405,  -0.0382,   ...,   0.0110,  -0.0509,
0.0318],
           [-0.0570,   0.0767,  -0.0491,   ...,  -0.0858,   0.0397,
```

```
0.0418]])
    fc4_weights ([[ 9.3059e-02, -4.6459e-02, -4.5485e-02,  2.3002e-01,
9.1647e-03,
            -6.8580e-02,   -7.0369e-02,     2.7714e-02,   -7.5141e-02,
-9.4338e-02,
            -1.1786e-01,    8.0293e-02,    3.5112e-02,    1.8736e-02,
9.3458e-02,
            -7.0348e-02,    1.7724e-01,   -9.6269e-04,    1.4932e-01,
-1.1682e-01,
             2.7166e-01,    8.8732e-02,   -1.1795e-01,   -4.4923e-02,
-5.9886e-02,
            -8.4444e-02,   -5.9348e-02,    1.9414e-02,    2.1302e-01,
-4.0925e-02,
             2.2628e-02,    4.5065e-02,   -4.6951e-02,   -9.3996e-02,
2.8974e-02,
             1.8106e-01,    8.2809e-02,   -9.6371e-02,   -1.7508e-02,
-2.2037e-02,
             1.9192e-01,   -8.4705e-02,   -5.2212e-02,   -7.3371e-02,
9.0866e-03,
             3.1419e-01,    2.7163e-02,    1.5922e-01,    2.5938e-01,
1.0962e-01,
             1.3306e-01,    1.2632e-01,    1.0863e-02,    5.1567e-03,
4.2730e-02,
             3.9103e-02,   -4.4040e-02,   -6.1862e-02,   -1.0002e-01,
-3.0268e-03,
            -8.2161e-02,   -7.1072e-02,   -8.4511e-02,   -3.2795e-02,
8.1840e-03,
             1.0091e-01,    2.6392e-01,   -4.8300e-02,   -5.6241e-03,
2.0445e-01,
            -2.1977e-02,   -8.7008e-02,    2.8184e-02,    1.8234e-02,
4.6348e-02,
             2.4983e-01,    1.1229e-01,    2.1672e-03,   -5.4920e-02,
-1.1492e-01,
             1.3772e-01,   -9.8008e-02,    3.2824e-03,    1.9112e-01,
1.4347e-01,
             2.8757e-03,    2.3092e-02,   -4.8185e-02,   -2.9848e-02,
-3.4586e-02,
            -3.6921e-02,   -8.8386e-02,    1.6393e-01,    1.2336e-01,
1.3570e-02,
            -3.1001e-02,   -5.9725e-02,   -1.8864e-02,    9.9997e-02,
-1.0183e-01,
            -1.0843e-01,    7.4204e-02,    1.5545e-01,    1.4902e-01,
-1.4746e-02,
```

```
           -5.0776e-02,    3.2533e-02,    1.4987e-01,    1.4239e-01,
-7.5512e-02,
           -1.6311e-02,    4.2465e-02,    6.7888e-02,    5.0868e-02,
2.4591e-02,
            4.6930e-02,    1.0705e-02,    6.5745e-02,    3.7003e-02,
-8.1330e-02,
           -3.8358e-02,   -4.3659e-02,    1.4403e-01,   -1.7188e-02,
-5.8590e-02,
           -1.0906e-01, -6.0016e-02,  1.3812e-01],
          [  2.1351e-01,   -2.9692e-02,   -2.8072e-02,    2.3616e-01,
6.2935e-02,
           -1.0201e-01,   -5.8975e-02,   -6.5667e-02,    6.2647e-03,
-7.5987e-02,
           -1.2410e-02,    1.8706e-01,    9.9402e-03,    4.7884e-02,
1.6687e-01,
            1.5269e-02,    7.8676e-02,   -3.4452e-03,    6.2852e-02,
4.5059e-02,
            2.3282e-01,    3.6337e-02,   -4.0299e-03,   -5.1043e-02,
-8.7927e-02,
           -8.0860e-03,   -7.7375e-02,    9.7222e-02,    2.3949e-01,
-8.9227e-02,
           -1.8858e-02,   -3.0869e-02,    5.0326e-02,   -1.0973e-01,
1.6836e-02,
            9.3119e-02,    8.9729e-02,   -3.1635e-02,    1.0969e-01,
4.4766e-03,
            1.6329e-01,   -1.0308e-01,   -3.5251e-02,   -9.0666e-02,
8.8521e-04,
            2.8185e-01,   -1.1440e-01,    1.9010e-01,    1.0339e-01,
8.3809e-02,
            1.1606e-01,    3.9359e-02,   -6.9893e-02,   -3.1760e-03,
-3.3082e-02,
            6.8030e-03,    2.1081e-02,   -7.7603e-02,   -9.7612e-02,
-1.1334e-01,
           -7.1277e-02,   -6.7007e-02,   -4.1515e-02,    1.1465e-01,
-2.4519e-02,
            5.2277e-02,    2.3262e-01,   -8.2867e-02,   -3.4361e-02,
1.5025e-01,
           -2.2628e-02,   -1.1417e-01,    3.8133e-02,    1.7298e-01,
-1.8323e-02,
            8.1759e-02,    1.6859e-01,   -4.7037e-02,   -9.4171e-02,
-9.3210e-02,
            1.2836e-01,   -5.1050e-02,   -9.2243e-02,    1.4436e-01,
9.1141e-02,
```

```
            4.9961e-02,   -2.4815e-03,     3.5760e-02,   -3.1024e-02,
-7.9133e-02,
           -5.6875e-02,   -4.2015e-02,     1.6671e-01,    8.4657e-02,
7.5265e-02,
            4.5437e-03,    1.5454e-02,   -6.2665e-02,    1.5656e-01,
1.2895e-02,
           -7.9122e-02,    1.9148e-01,    4.1419e-02,    1.6208e-01,
3.4109e-02,
           -1.3921e-02,   -2.8380e-02,    2.9389e-01,    1.4226e-01,
-9.5250e-02,
           -6.6341e-02,   -7.6147e-02,   -4.8886e-02,   -8.7450e-02,
-7.1646e-02,
            1.3426e-01,    1.0925e-02,    1.1497e-01,   -3.9744e-02,
-6.5903e-02,
           -1.2217e-01,   -1.2182e-02,    1.7569e-01,   -1.0244e-01,
6.6390e-02,
           -3.2420e-02, -1.3473e-02,  1.8935e-01],
          [  1.1901e-01,   -9.4128e-02,   -9.0807e-03,    1.9821e-01,
3.4366e-02,
           -5.8035e-02,   -4.7631e-02,   -6.7450e-02,    3.3279e-03,
2.4574e-02,
           -2.2935e-02,    1.2923e-01,   -9.6639e-02,   -2.3124e-02,
2.4760e-01,
           -1.0472e-01,    9.3477e-02,   -1.0558e-01,    1.5956e-01,
-4.5313e-02,
            1.7169e-01,    1.1716e-01,    3.4271e-02,    1.7804e-03,
-5.9712e-02,
           -7.9103e-02,   -2.7370e-02,    9.1895e-02,    1.6140e-01,
-2.2893e-02,
           -8.1692e-02,   -5.5912e-02,   -1.0532e-01,   -2.1346e-02,
-2.6514e-02,
            1.6042e-01,    1.9666e-01,   -5.7544e-02,    9.1547e-03,
-1.0193e-01,
            2.3011e-01,    3.7957e-02,   -1.0616e-01,   -7.7428e-02,
-1.1047e-01,
            1.9697e-01,    4.7905e-02,    1.3690e-01,    1.0588e-01,
1.0077e-01,
            5.0391e-02,    7.3920e-02,   -1.0856e-01,   -1.8498e-02,
-2.0659e-02,
            6.1168e-02,   -3.7170e-02,   -9.9050e-02,   -1.2097e-02,
-4.7778e-02,
           -4.7533e-02,    3.7878e-02,    3.7904e-02,   -5.2295e-02,
-2.9312e-04,
```

```
            6.5374e-02,     1.3798e-01,   -3.2407e-02,   -1.1203e-02,
1.4207e-01,
           -1.9923e-02,   -6.4454e-02,   -8.1968e-02,     6.9696e-02,
-1.9118e-02,
            1.1696e-01,     7.4625e-02,     2.9748e-02,   -9.7051e-02,
-6.5513e-02,
            5.1472e-02,   -1.5276e-02,   -7.9857e-02,     2.5570e-01,
7.9117e-02,
           -1.0929e-02,   -1.8155e-02,     2.6166e-02,     1.6404e-02,
-9.3830e-03,
           -9.7319e-02,   -3.2940e-02,     2.5169e-01,     1.6444e-01,
-1.9505e-02,
            3.5774e-02,   -6.1391e-03,   -7.8880e-02,     7.6563e-02,
-1.1790e-01,
            2.4984e-02,     1.5851e-01,     6.9867e-02,     1.6784e-01,
-1.0566e-01,
           -8.8914e-02,   -6.9090e-02,     2.3813e-01,     1.0169e-01,
-4.4809e-02,
           -1.4446e-03,   -3.1632e-02,   -5.8005e-02,     4.6330e-02,
-9.6746e-02,
            1.3121e-01,   -3.2689e-02,     1.2710e-01,   -9.2143e-02,
-1.0140e-01,
           -2.7316e-02,     4.7102e-02,     1.8345e-01,     5.9334e-02,
6.0789e-02,
           -9.7960e-02, -8.5420e-02,  2.4426e-01],
          [  2.3934e-01,  -1.0027e-01,     2.6131e-02,     1.2116e-01,
2.3617e-02,
            1.5558e-02,   -2.3417e-02,     4.4109e-02,   -4.3742e-02,
-1.9448e-03,
           -1.0476e-01,     1.9050e-01,   -1.0359e-01,   -3.9786e-02,
2.0595e-01,
           -2.0723e-02,     1.4334e-01,   -3.4990e-02,     1.1204e-01,
-7.2540e-02,
            2.4304e-01,     1.1399e-01,   -1.0107e-01,   -6.5935e-02,
-5.1727e-02,
           -8.0159e-02,     8.4884e-03,     8.1498e-02,     2.4450e-01,
-7.0126e-02,
           -4.5290e-02,     2.6813e-03,   -1.0453e-01,   -1.1489e-02,
4.8308e-02,
            1.6508e-01,     1.0637e-01,   -1.0411e-01,     9.4414e-02,
-1.0381e-01,
            2.3112e-01,   -7.3444e-02,   -1.0814e-01,     7.4677e-04,
-9.2275e-02,
```

```
            1.8359e-01,  -9.5607e-02,    1.6560e-01,    2.4703e-01,
8.1384e-02,
            1.7748e-01,    1.4881e-01,  -7.1624e-02,  -1.0105e-01,
1.0607e-02,
            2.9770e-02,  -8.0165e-02,    4.3573e-02,    7.7593e-02,
-5.1108e-02,
            8.0262e-03,  -9.6141e-02,  -7.2257e-02,    7.4982e-03,
-1.9697e-02,
            1.2221e-01,    1.9409e-01,    5.5436e-02,  -8.4651e-02,
7.5080e-02,
           -7.5192e-02,    5.0470e-02,    1.2099e-02,    1.3652e-01,
-1.0023e-01,
            1.1632e-01,    1.5940e-01,    9.0192e-02,    4.5977e-02,
-7.7311e-02,
            1.1369e-01,  -1.6700e-02,  -9.3977e-03,    1.1254e-01,
1.1178e-01,
            4.5045e-02,    7.5515e-03,  -1.2000e-02,    2.9771e-02,
-5.0652e-02,
           -6.2686e-02,  -7.0860e-02,    2.1577e-01,    1.3921e-01,
1.3382e-01,
           -3.5713e-02,  -5.9044e-02,  -1.0272e-01,    1.5142e-01,
-1.3827e-02,
           -4.0302e-02,    1.0962e-01,    1.1689e-01,    2.4095e-01,
-6.4607e-03,
           -8.7346e-02,  -8.0652e-02,    3.1888e-01,    1.8717e-01,
-6.4523e-02,
           -1.0341e-01,    5.0393e-02,    2.4890e-02,  -7.4718e-02,
-1.2242e-02,
            6.7534e-02,  -6.0284e-02,    1.0794e-01,  -1.0466e-01,
-1.0684e-01,
           -2.9888e-02,    5.3365e-02,    1.6527e-01,  -4.9402e-02,
7.8446e-02,
           -4.1689e-02, -1.7489e-03,  1.4214e-01]])
   fc1_bias ([ 0.1480,  0.2340, -0.0142, -0.2174,  0.1083,  0.1398,
0.1037,  0.0622,
           0.0136,   0.2442,   0.3421, -0.0193, -0.2460, -0.2265,
-0.1624, -0.0194,
          -0.1305,   0.3749,   0.0442, -0.2703,   0.1191,   0.1238,
-0.1373, -0.0172,
          -0.1194, -0.1399,   0.2527,   0.0268,   0.2745,   0.0677,
0.0149,  0.0128,
           0.4145, -0.2106,   0.0631, -0.0355,   0.2034, -0.0739,
0.2714, -0.0536,
```

```
           0.2215,   0.1872,   0.3504,   0.2802,   0.2992, -0.0935,
-0.0290,  0.2048,
           0.3080, -0.1608,   0.3172, -0.1100, -0.0176,   0.3103,
-0.0452, -0.2231,
           0.2737,   0.2242, -0.2505,   0.2219,   0.1169, -0.1858,
0.2615, -0.2794,
           0.2471,   0.2710,   0.0048,   0.3914,   0.0654,   0.2976,
0.3629,  0.1935,
          -0.1372,   0.0883,   0.0637, -0.0701,   0.0793, -0.0811,
-0.2138,  0.0309,
          -0.1390,   0.0015,   0.2860,   0.2903,   0.3576,   0.1104,
0.0482, -0.3239,
           0.4177, -0.1372, -0.1032,   0.2243,   0.2718, -0.2817,
0.3255,  0.1806,
           0.2527,   0.0848, -0.2264, -0.0324, -0.0056, -0.0326,
-0.1121,  0.2835,
          -0.2215, -0.1425, -0.1829,   0.2114,   0.2717, -0.2716,
0.3047, -0.2661,
          -0.2556, -0.0471, -0.0447, -0.1500, -0.2100, -0.3726, -0.1429,
0.0331,
          -0.1687,   0.1198,   0.3258, -0.1013,   0.0101,   0.4393,
0.3428, -0.2741])
   fc2_bias ([ 0.0811,  0.0266,  0.1179,  0.1269,  0.1121,  0.0727,
-0.0346, -0.0286,
          -0.0969, -0.1013,   0.0141,   0.0489,   0.0705, -0.0539,
0.1405, -0.0648,
           0.1087, -0.0747,   0.1725,   0.1161,   0.0490,   0.0304,
0.0688, -0.0978,
           0.0269,   0.0403,   0.0983,   0.0489,   0.0070, -0.0469,
0.1215, -0.0758,
           0.0638,   0.0161,   0.0398,   0.1278,   0.0479,   0.1621,
0.0391,  0.0024,
           0.0092,   0.1082, -0.0137,   0.0069,   0.0974, -0.0165,
0.0672,  0.1451,
           0.0221,   0.0075,   0.0300,   0.0461,   0.0844,   0.0018,
0.0675,  0.0651,
          -0.0150,   0.1667,   0.0325,   0.0830,   0.0517, -0.0937,
-0.0208,  0.0110,
           0.0021,   0.1274,   0.0767,   0.0601,   0.0996,   0.1048,
0.0062, -0.0374,
           0.0059,   0.0368, -0.0472,   0.1050,   0.1291,   0.0962,
0.0773,  0.0439,
           0.0755, -0.0042, -0.0378,   0.0591, -0.0750, -0.0379,
```

```
0.0479,  0.0958,
           0.0214,   0.0431,   0.0417,   0.0159,   0.0746,  -0.0498,
0.0319,  0.0231,
           0.0390,   0.0260,   0.0380,   0.0144,   0.0869,  -0.0327,
0.0066,  0.0587,
          -0.0098,  -0.0063,   0.0124,   0.1112,   0.0421,   0.1431,
-0.0472,  0.0952,
          -0.0024,  -0.0329,  -0.0420,   0.0915,  -0.0142,   0.0652,
0.1162,  0.0088,
          -0.0209,   0.0101,  -0.1258,   0.0253,  -0.0592,   0.0744,
-0.0071,  0.0922])
    fc3_bias ([ 0.0122,  0.1110, -0.0292,  0.1649,  0.0017,  0.1069,
0.0778, -0.0163,
           0.0456,  -0.0126,   0.0006,   0.1459,   0.0220,  -0.0229,
0.1128,  0.0695,
           0.1393,   0.0037,  -0.0183,   0.1348,   0.1536,  -0.0027,
0.0381,  0.0957,
           0.0454,  -0.0192,   0.0083,   0.0049,   0.1580,   0.1065,
0.0012, -0.0018,
           0.0412,   0.0017,  -0.0768,   0.1614,   0.0031,   0.1102,
-0.0311,  0.0283,
           0.1600,   0.1026,   0.0543,   0.1113,   0.1186,   0.1965,
0.1216,  0.0289,
           0.1197,  -0.0050,   0.0547,   0.0645,   0.1099,   0.0592,
0.0645, -0.0008,
           0.0503,   0.0277,  -0.0365,   0.0874,   0.1279,   0.1380,
0.1324, -0.0137,
          -0.0799,   0.0154,   0.1688,   0.0115,   0.0446,   0.0961,
0.1190,  0.0358,
          -0.0239,  -0.0005,   0.1203,   0.0573,   0.0092,  -0.1177,
-0.0072,  0.0132,
           0.0342,   0.1386,   0.0786,   0.1142,  -0.0053,  -0.1010,
-0.0950, -0.0607,
          -0.0947,  -0.0277,   0.0697,  -0.0134,   0.1190,   0.0179,
-0.0551, -0.0750,
           0.0428,   0.1157,   0.0425,   0.0998,   0.0881,   0.0459,
0.0464,  0.1063,
          -0.0449,   0.0223,  -0.0525,   0.2437,   0.0373,   0.1289,
0.0896,  0.0649,
          -0.0007,  -0.0261,   0.1254,   0.0477,   0.0498,  -0.0293,
0.0254,  0.0816,
           0.1140,  -0.0458,   0.0554,   0.0202,   0.0329,  -0.0125,
0.0766, -0.0036])
```

fc4_bias ([-0.0503, -0.1164, 0.0018, 0.0240])

附录二

神经网络的部分参数如下：

fc1_weights ([[0.1713, -0.1039, -0.2577, ..., -0.0824, 0.0719,
-0.2845],
[-0.2640, 0.2745, -0.0006, ..., -0.1140, -0.0802,
-0.0187],
[0.2312, -0.0232, 0.2010, ..., 0.3168, -0.1216,
0.2441],
...,
[-0.0480, 0.2822, -0.0680, ..., 0.3202, 0.3039,
0.2823],
[0.1435, -0.1451, 0.3138, ..., 0.1219, -0.0337,
0.2216],
[0.0177, -0.0045, 0.3106, ..., 0.1246, 0.1667,
-0.3066]])

fc2_weights ([[-0.0353, 0.0485, 0.0194, ..., -0.0674, 0.0585,
0.0782],
[-0.0114, 0.0166, 0.0027, ..., -0.0538, 0.0443,
-0.0547],
[0.0279, 0.0103, -0.0471, ..., 0.0689, 0.0882,
-0.0560],
...,
[-0.0702, 0.0144, -0.0525, ..., 0.0711, -0.0481,
0.0599],
[0.0529, -0.0821, 0.0403, ..., 0.0575, -0.0237,
-0.0642],
[0.0389, -0.0480, 0.0246, ..., -0.0003, 0.0870,
0.0621]])

fc3_weights ([[-0.0751, -0.0879, -0.0151, ..., -0.0828, -0.0667,
-0.0538],
[0.0009, -0.0139, -0.0586, ..., 0.0488, 0.0381,
-0.0661],
[0.0175, 0.0789, -0.0820, ..., 0.0610, -0.0171,
-0.0529],
...,
[0.0211, -0.0617, 0.0169, ..., 0.0220, -0.0784,
-0.0491],
[-0.0686, 0.0710, -0.0326, ..., -0.0714, -0.0037,
0.0487],
[0.0638, -0.0624, 0.0708, ..., 0.0833, 0.0392,
-0.0203]])

```
    fc4_weights ([[-5.8174e-02, -8.0977e-02,  3.9299e-02,  2.4994e-02,
-4.4581e-02,
            -6.0245e-02,    1.4903e-03,  -4.2084e-02,    8.1265e-02,
1.6197e-02,
            -1.8695e-02,    4.2060e-02,  -2.1849e-02,  -7.4715e-02,
-3.8552e-02,
            -2.7149e-02,    3.9247e-02,    7.5486e-02,  -2.3316e-02,
7.9082e-02,
            -3.2416e-02,    3.8460e-03,    3.2618e-02,    3.7323e-02,
-1.3667e-02,
            -5.5876e-02,  -6.8210e-02,    5.6212e-02,  -5.1256e-02,
-2.6275e-02,
             7.4099e-03,    8.7737e-02,  -3.4859e-02,    2.6133e-02,
5.5960e-02,
            -7.4558e-02,    3.0359e-02,  -5.4714e-02,    8.0497e-02,
-8.6662e-02,
            -4.2512e-02,   -7.6012e-02,   -7.9146e-03,   -2.5418e-02,
-6.5955e-02,
            -4.0427e-02,  -3.5067e-02,  -8.1673e-02,    6.8398e-03,
8.4158e-02,
             7.2387e-02,    5.6957e-02,  -6.8238e-02,    6.8875e-02,
-6.3669e-02,
            -7.9450e-02,    3.5192e-03,    2.6937e-03,  -5.8324e-02,
4.1695e-02,
            -8.1918e-02,  -1.5956e-02,    7.0665e-02,    2.3651e-02,
-8.0694e-02,
            -2.4581e-02,  -2.4339e-02,    5.1436e-02,  -4.7092e-02,
8.5195e-03,
             2.4411e-02,  -4.9286e-02,    6.2106e-02,  -4.0458e-02,
-4.2052e-02,
             5.1423e-02,  -1.6184e-02,  -4.8322e-02,    6.1119e-02,
4.0796e-02,
            -4.5001e-02,  -6.7169e-02,    5.1067e-02,  -6.5339e-02,
7.4244e-02,
             2.3961e-02,   -6.7072e-03,   -4.0034e-02,   -5.5036e-02,
6.3628e-02,
             4.8005e-02,  -1.9380e-02,  -8.2355e-02,    1.3736e-02,
-7.4460e-02,
             7.0613e-02,  -5.0222e-02,    2.1691e-02,    6.4198e-02,
5.7705e-02,
            -5.1466e-02,    4.9098e-02,  -4.8421e-02,  -7.2822e-02,
6.1266e-02,
            -2.8933e-02,   -7.3800e-02,   -4.8485e-02,   -6.1404e-02,
```

```
7.8811e-02,
            -6.8142e-02,   -4.4195e-02,   -7.0720e-02,     6.4230e-02,
1.0093e-02,
             5.8062e-02,    9.8847e-03,    4.3260e-03,   -2.7482e-02,
4.7731e-03,
             4.4363e-02,   -6.0413e-02,   -5.8212e-02,   -3.5901e-02,
6.9106e-02,
            -4.0862e-02, -8.5339e-02,  6.6371e-02],
           [-7.3136e-02,   -8.2525e-02,   -3.0244e-02,   -3.1460e-02,
-2.2976e-02,
            -3.6606e-02,   -7.0879e-02,    1.7152e-02,   -2.7551e-02,
-6.9438e-02,
             2.6002e-02,   -7.3229e-03,    3.8474e-02,    2.8169e-02,
-2.9086e-02,
             5.2870e-02,    2.7158e-03,   -5.4839e-02,    5.7884e-02,
-7.6709e-02,
             7.5749e-02,   -4.7651e-02,   -6.8437e-02,    6.7878e-02,
-6.0067e-02,
            -5.6606e-02,    2.4289e-02,    4.7968e-02,   -2.1478e-02,
8.6176e-02,
             3.0593e-02,   -6.2151e-02,   -2.0306e-02,   -7.5809e-02,
-3.1632e-02,
            -6.2073e-02,   -7.8655e-02,    3.0976e-02,    2.0649e-02,
6.8346e-03,
            -3.0166e-02,    1.3640e-02,   -2.8695e-02,    8.1006e-02,
2.0785e-02,
            -2.5505e-02,   -3.7952e-02,    1.7706e-02,    8.7361e-02,
2.5017e-02,
            -1.5930e-03,    2.5087e-02,    4.1548e-02,    6.3298e-02,
2.4572e-03,
             3.5170e-02,   -3.2152e-02,    7.9047e-02,    4.6264e-02,
-8.0211e-02,
            -7.9975e-03,   -5.9178e-02,   -6.6321e-02,   -7.8126e-02,
6.5668e-02,
            -5.6784e-02,   -1.2018e-02,    1.8063e-02,    5.8883e-02,
-4.0990e-02,
             2.3009e-02,   -6.8735e-02,    2.0835e-02,    7.5736e-02,
2.1614e-05,
            -1.5268e-02,    4.9575e-02,   -2.1445e-02,    8.1830e-02,
-1.6237e-03,
             8.8086e-02,    2.9504e-02,   -6.5520e-02,    1.1662e-02,
-6.1474e-02,
            -4.1073e-02,   -8.7386e-02,   -8.1918e-02,    8.2413e-02,
```

```
-4.6422e-02,
             3.4487e-02,     2.2706e-02,   -2.1494e-02,     1.8916e-02,
2.1369e-02,
            -2.9265e-02,     7.3294e-02,   -2.3484e-02,     8.2218e-02,
-5.8776e-02,
             5.2330e-02,     1.2190e-02,   -6.6440e-02,   -6.6038e-02,
-8.3968e-02,
             4.8613e-02,     1.9650e-02,   -5.0879e-02,     2.0348e-02,
8.4926e-02,
             5.2823e-02,   -3.4725e-02,   -3.3006e-02,     2.7594e-02,
2.0409e-02,
            -2.7430e-03,     7.7070e-02,    8.4623e-02,    2.5266e-02,
-8.2369e-02,
            -3.3895e-02,    -4.6728e-02,   -2.0878e-02,   -6.7512e-02,
6.6180e-02,
             4.8883e-02,  4.4454e-02,  6.1101e-02],
           [  7.0818e-02,  -2.6246e-02,    7.4153e-02,    1.5817e-02,
-1.4961e-02,
             2.2093e-02,     5.0268e-02,    3.0949e-02,  -4.3270e-02,
-1.0619e-02,
            -2.8543e-02,   -7.3387e-02,    2.3591e-02,    4.6256e-02,
-8.7385e-02,
            -8.2651e-02,     4.6559e-02,   -7.9741e-02,   -6.6325e-02,
-7.4284e-02,
            -6.9310e-02,     2.3651e-02,   -5.0319e-02,    7.2774e-02,
4.4622e-02,
            -5.0999e-02,   -8.0276e-02,     6.6114e-02,   -1.1274e-02,
-8.3586e-02,
            -6.8432e-02,     8.2504e-02,   -7.6115e-02,   -6.6594e-02,
3.7433e-02,
             3.0792e-02,     2.7831e-02,   -3.3352e-02,   -4.0737e-02,
-5.0470e-02,
             5.8450e-02,   -1.9219e-02,   -6.4426e-02,     1.7391e-02,
8.2687e-02,
            -3.6919e-02,   -5.9678e-02,   -3.3910e-02,     1.1744e-02,
4.1525e-02,
             4.7489e-02,   -3.5247e-02,   -3.2518e-02,   -7.4773e-02,
-2.8919e-02,
             6.8512e-02,   -7.0756e-02,    6.1083e-02,    5.4786e-02,
-5.3232e-02,
            -1.0975e-02,   -7.0035e-02,     5.0730e-02,   -7.5112e-02,
8.4132e-02,
            -2.5385e-02,   -8.8146e-02,     7.7398e-02,   -4.1488e-02,
```

```
4.0095e-02,
            -4.5764e-02,    7.7291e-02,  -5.0021e-02,    3.2726e-02,
-7.8277e-02,
             5.8155e-02,    2.4880e-02,   5.9138e-02,    5.0200e-02,
7.1803e-02,
             2.9719e-02,   -5.6220e-02,  -1.5186e-02,   -8.5197e-02,
1.1172e-03,
            -1.7580e-03,    6.6123e-03,   3.5297e-02,  -4.7656e-02,
-6.8791e-02,
            -6.3442e-02,   -5.9051e-03,  -6.0617e-02,   -6.7431e-02,
8.3938e-03,
             7.7379e-02,   -2.9354e-02,   6.2852e-02,    3.8435e-02,
-1.5470e-02,
             4.2759e-02,   -8.5293e-02,  -7.1656e-02,    4.3200e-02,
-7.3510e-02,
             2.0044e-02,   -2.2321e-02,   3.4940e-02,    8.7219e-02,
7.7503e-02,
            -5.4557e-02,   -2.3215e-02,  -2.8879e-02,    3.2538e-02,
-3.3381e-02,
            -7.9830e-03,   -8.3499e-02,  -5.0984e-02,   -6.2701e-02,
4.8884e-02,
             6.9804e-02,   -2.2731e-02,  -4.1077e-02,   -6.9938e-02,
3.9429e-02,
             8.1110e-02,  5.1469e-02,  1.6924e-02],
           [  8.2848e-02,  -4.7601e-02,  -1.9854e-02,    6.4818e-02,
-5.2408e-02,
            -7.7654e-02,    8.1968e-02,  -8.1166e-02,   -8.9635e-03,
-4.8490e-02,
            -7.7951e-02,    3.6801e-02,  -6.1851e-02,   -1.8512e-02,
2.9123e-02,
             6.8807e-02,    2.8921e-02,   3.7661e-02,  -4.9916e-02,
3.2204e-02,
             2.5809e-03,   -5.8519e-02,  -6.5956e-02,    7.6752e-02,
1.7071e-02,
            -3.3162e-02,   -1.1243e-02,   7.5480e-03,    1.2399e-02,
7.4106e-02,
            -5.5516e-02,   -2.4071e-02,  -1.5830e-02,   -8.5437e-02,
-7.8929e-02,
             8.5253e-02,   -4.6976e-02,  -5.7163e-02,    8.2313e-02,
-4.2794e-02,
             8.8314e-02,    3.7785e-02,  -3.6079e-02,    2.3914e-02,
-6.7997e-02,
            -8.1177e-02,    3.1382e-02,   8.6998e-03,  -2.4730e-02,
```

```
-4.3615e-02,
             7.1695e-02,     5.5292e-03,   -8.5867e-02,     7.7514e-02,
5.8106e-02,
            -6.5357e-02,   -3.0188e-02,   -7.3610e-02,     4.8650e-02,
-4.5115e-02,
             2.6390e-02,    4.9796e-02,    6.6228e-02,    1.3710e-02,
2.0284e-02,
             2.9390e-03,    6.0815e-03,    2.5919e-02,   -3.9113e-02,
8.5868e-02,
            -6.8934e-02,   -1.9805e-02,   -7.3774e-02,     2.6106e-02,
8.8133e-02,
            -4.8235e-02,   -3.9911e-02,    4.7038e-02,    1.8387e-02,
-7.2787e-02,
            -1.2128e-03,    1.7659e-02,    6.8389e-02,   -2.5327e-02,
-7.9452e-02,
             4.7927e-02,    4.8301e-03,    4.8930e-02,   -8.3396e-02,
-4.2425e-02,
             3.4128e-03,    7.6602e-02,   -5.8841e-02,   -2.2292e-02,
5.1113e-02,
            -7.1612e-02,    2.7142e-02,   -8.1527e-02,    8.3503e-02,
3.6154e-02,
            -3.6273e-02,   -3.2576e-02,   -2.5349e-02,   -8.9532e-03,
9.7264e-03,
             6.8434e-02,   -2.6290e-02,   -1.1433e-02,   -1.3732e-02,
-7.9207e-02,
            -6.9766e-03,    4.1258e-02,   -6.9923e-02,    1.9943e-02,
5.2400e-02,
             7.3216e-02,   -5.5680e-02,    2.3752e-02,    5.8721e-02,
-6.1622e-02,
             4.3693e-02,   -2.4381e-04,   -1.4439e-02,   -6.6351e-02,
3.2335e-02,
             5.4499e-02, -6.2266e-02, -8.0407e-04]])
   fc1_bias ([-0.2990,  0.1760,  0.3324, -0.0684,  0.1691, -0.1476,
-0.0042, -0.3195,
           -0.3050,  -0.3249,  -0.2637,  -0.1292,  -0.2437,  -0.2975,
0.2259, -0.2427,
            0.3018,  -0.1621,  -0.1431,   0.0200,   0.1443,  -0.2412,
0.2487, -0.3246,
            0.2328,   0.3138,   0.2147,  -0.1336,   0.2007,  -0.0471,
0.2489, -0.1696,
           -0.1136,   0.0670,  -0.2179,   0.2009,   0.3014,  -0.0793,
-0.0307, -0.2136,
           -0.1371,  -0.1097,   0.2409,   0.1192,  -0.2525,  -0.1200,
```

```
0.2727, -0.2597,
          -0.2095,  -0.1579,  -0.2221,  -0.3262,   0.2678,  -0.2607,
-0.2946,  0.2471,
          -0.3010,   0.2397,   0.1172,  -0.2491,   0.0671,   0.2226,
-0.1890,  0.2516,
           0.2420,  -0.2748,   0.2016,   0.0110,   0.1868,   0.1349,
-0.0999,  0.0383,
          -0.1434,  -0.1082,   0.1614,  -0.0826,   0.0446,  -0.2456,
0.0612, -0.2868,
          -0.1654,  -0.1306,   0.1982,  -0.1068,   0.1390,  -0.1672,
0.3175, -0.2244,
           0.2431,   0.2937,   0.0979,  -0.2803,   0.1390,  -0.2818,
-0.0217,  0.2726,
          -0.3024,   0.2125,   0.2517,   0.1915,  -0.0736,  -0.2052,
-0.1474,  0.0116,
           0.0912,   0.0051,   0.1525,   0.2283,   0.2625,   0.2928,
-0.1367, -0.1599,
           0.1587,  -0.3122,   0.1291,  -0.1290,  -0.1576,   0.1393,
-0.2013, -0.2580,
          -0.3139,  -0.2020,  -0.1167,   0.1340,   0.1269,  -0.0058,
-0.2918, -0.2479])
   fc2_bias ([-0.0592, -0.0073, -0.0601, -0.0673,  0.0271,  0.0883,
-0.0821, -0.0148,
           -0.0110,  -0.0776,   0.0647,  -0.0267,   0.0834,  -0.0109,
-0.0533, -0.0708,
           -0.0208,   0.0467,  -0.0372,   0.0150,  -0.0169,   0.0789,
0.0240,  0.0576,
           -0.0648,   0.0028,  -0.0220,  -0.0701,  -0.0722,   0.0103,
-0.0268, -0.0549,
            0.0551,   0.0579,   0.0257,   0.0331,  -0.0353,  -0.0646,
0.0427,  0.0845,
            0.0337,  -0.0625,   0.0464,  -0.0871,   0.0088,  -0.0632,
-0.0322,  0.0244,
            0.0050,  -0.0505,  -0.0753,   0.0708,  -0.0232,  -0.0501,
-0.0683, -0.0114,
            0.0698,  -0.0811,   0.0058,   0.0605,  -0.0037,  -0.0637,
-0.0572,  0.0522,
           -0.0804,   0.0588,   0.0243,   0.0223,  -0.0245,   0.0247,
-0.0186, -0.0697,
            0.0572,   0.0836,  -0.0242,  -0.0830,   0.0210,  -0.0198,
0.0805, -0.0318,
            0.0617,  -0.0186,  -0.0660,  -0.0192,   0.0734,  -0.0070,
0.0151, -0.0015,
```

```
          -0.0425,  -0.0835,   0.0049,   0.0880,  -0.0640,   0.0267,
-0.0775,  0.0033,
           0.0165,   0.0427,   0.0565,  -0.0067,  -0.0458,   0.0170,
0.0728, -0.0559,
          -0.0660,   0.0061,   0.0208,  -0.0684,   0.0564,   0.0600,
0.0140, -0.0655,
           0.0381,   0.0588,   0.0594,   0.0478,   0.0575,  -0.0072,
0.0729,  0.0414,
           0.0495,   0.0828,  -0.0113,  -0.0806,   0.0298,  -0.0022,
0.0332,  0.0212])
   fc3_bias ([ 4.2408e-02,   7.5830e-02,   6.8950e-02,   2.1540e-02,
5.5833e-02,
           4.8692e-03,    3.3868e-02,    3.0484e-02,  -1.7070e-02,
3.7822e-02,
           3.0021e-02,    7.7647e-02,  -4.3418e-02,    3.2161e-02,
-2.2234e-02,
          -1.9787e-02,  -2.7237e-03,  -4.3806e-03,    3.5433e-02,
1.7427e-05,
           3.1298e-02,   -2.1070e-02,   -3.5486e-04,   -7.4799e-02,
1.9312e-02,
          -6.9865e-02,    7.0002e-02,    4.5469e-02,    6.6501e-02,
8.5939e-02,
          -9.7107e-03,  -4.4504e-03,    1.9915e-02,    6.6032e-02,
8.6161e-02,
           8.8438e-03,    5.0384e-02,  -6.1096e-02,  -6.9091e-02,
7.7917e-03,
          -4.4845e-02,  -6.1582e-02,  -4.1469e-02,    6.4179e-02,
-5.4732e-02,
          -4.3637e-02,    4.8910e-03,    3.4428e-02,  -1.8957e-02,
2.9300e-02,
          -5.6802e-02,  -7.2899e-02,    4.5947e-02,    8.3106e-02,
5.4316e-02,
           7.4613e-02,   -2.6765e-02,   -2.6713e-02,   -2.9444e-02,
3.5044e-02,
          -1.4919e-02,  -2.1322e-02,    7.6034e-02,  -4.9590e-02,
2.4137e-02,
          -4.6246e-03,    3.1189e-02,    3.0720e-02,    8.6546e-02,
3.1366e-02,
           9.5602e-03,   -8.4288e-02,   -7.3388e-02,   -3.2247e-02,
-3.1284e-02,
          -4.0539e-03,  -8.0195e-02,    7.8492e-02,  -7.5917e-02,
-7.2831e-02,
          -4.3539e-03,    1.5448e-02,  -3.9295e-02,  -7.9944e-03,
```

-2.9277e-02,
            6.0977e-02,   -2.2041e-02,   -3.8314e-02,   8.4641e-02,
7.4338e-02,
           -1.1297e-04,   8.2940e-02,   -6.0967e-02,   -6.0804e-02,
-7.8904e-02,
           -4.9917e-02,   -4.3714e-03,   -1.9855e-03,   7.1746e-02,
-8.4238e-02,
            2.1622e-02,   2.7671e-02,   8.3575e-02,   -6.4186e-02,
-2.7827e-02,
            8.5915e-02,   -4.5582e-02,   2.1307e-02,   -4.9832e-02,
3.8489e-02,
            8.2230e-02,   8.6293e-02,   3.4783e-03,   7.5713e-02,
1.9149e-02,
            1.8189e-02,   4.6096e-02,   -5.1339e-02,   -5.2326e-02,
-3.1277e-03,
            3.2993e-03,   -8.4581e-02,   -3.6118e-02,   7.2603e-02,
8.7725e-02,
           -5.4814e-03, 6.8790e-02, -5.2900e-02])
    fc4_bias ([-0.0019, 0.0679, 0.0592, -0.0742])

    附录三
    神经网络的部分参数如下：
    fc1_weights ([[-0.2375, 0.0725, 0.0579, ..., -0.1677, -0.0709,
-0.2701],
           [-0.2302,   0.4054, -0.0674,   ..., -0.0638, -0.0212,
0.0632],
           [0.2159,   0.0075,   0.2077,   ...,   0.1846,   0.1938,
0.4613],
           ...,
           [0.2358, -0.2608, -0.0176,   ...,   0.0200,   0.1300,
-0.1774],
           [0.3776,   0.2322,   0.0190,   ...,   0.2942, -0.0961,
0.0848],
           [-0.0029, -0.3874,   0.2657,   ...,   0.6340, -0.0349,
-0.0579]])
    fc2_weights ([[-0.0589, -0.1601, 0.1068, ..., 0.2009, -0.0204,
-0.2095],
           [-0.0230, -0.0487,   0.0632,   ...,   0.0778,   0.0231,
-0.0647],
           [-0.0708, -0.1731,   0.2491,   ...,   0.1030,   0.0337,
-0.3684],
           ...,
           [-0.0200,   0.0751,   0.0199,   ...,   0.0720, -0.0719,

```
-0.0308],
          [-0.0800,  -0.0209,    0.1627,    ...,    0.0586,    0.1010,
-0.2697],
          [  0.0509,  -0.1682,    0.1520,    ...,    0.1379,  -0.0350,
-0.2218]])
   fc3_weights ([[-0.0141, -0.0191,  0.0278,  ...,  0.0826, -0.0018,
0.1129],
          [  0.0317,    0.0147,    0.0804,    ...,  -0.0492,    0.0197,
0.0359],
          [  0.0527,  -0.0208,  -0.0897,    ...,  -0.0149,    0.0512,
-0.0203],
          ...,
          [-0.0031,  -0.0770,  -0.0382,    ...,    0.0235,    0.0965,
0.0693],
          [  0.0085,  -0.0369,  -0.0383,    ...,    0.0110,  -0.0521,
0.0316],
          [-0.0601,    0.0767,  -0.0492,    ...,  -0.0858,    0.0358,
0.0394]])
   fc4_weights ([[ 0.0847, -0.0460, -0.0456,  0.2236,  0.0092, -0.0680,
-0.0699,  0.0282,
            -0.0746,  -0.0937,  -0.1172,   0.0707,   0.0357,   0.0187,
0.0877, -0.0698,
             0.1679,  -0.0004,   0.1439,  -0.1163,   0.2618,   0.0786,
-0.1174, -0.0442,
            -0.0594,  -0.0839,  -0.0588,   0.0194,   0.2040,  -0.0404,
0.0232,  0.0451,
            -0.0464,  -0.0934,   0.0290,   0.1756,   0.0722,  -0.0959,
-0.0277, -0.0214,
             0.1828,  -0.0841,  -0.0517,  -0.0728,   0.0096,   0.3031,
0.0277,  0.1592,
             0.2499,   0.1028,   0.1238,   0.1213,   0.0114,   0.0057,
0.0427,  0.0391,
            -0.0436,  -0.0612,  -0.0995,  -0.0025,  -0.0816,  -0.0706,
-0.0840, -0.0328,
             0.0080,   0.0908,   0.2569,  -0.0477,  -0.0049,   0.1944,
-0.0215, -0.0865,
             0.0282,   0.0078,   0.0468,   0.2399,   0.1031,   0.0018,
-0.0544, -0.1144,
             0.1377,  -0.0975,   0.0038,   0.1813,   0.1435,   0.0029,
0.0231, -0.0482,
            -0.0299,  -0.0340,  -0.0363,  -0.0879,   0.1541,   0.1138,
0.0039, -0.0310,
            -0.0592,  -0.0183,   0.0902,  -0.1013,  -0.1079,   0.0679,
```

```
0.1453,  0.1384,
           -0.0143,  -0.0503,   0.0325,   0.1394,   0.1329,  -0.0749,
-0.0158,  0.0425,
            0.0679,   0.0513,   0.0252,   0.0377,   0.0111,   0.0550,
0.0375, -0.0808,
           -0.0378,  -0.0437,   0.1440,  -0.0168,  -0.0586,  -0.1085,
-0.0595,  0.1287],
          [ 0.2052,  -0.0293,  -0.0282,   0.2297,   0.0629,  -0.1014,
-0.0585, -0.0652,
            0.0068,  -0.0753,  -0.0117,   0.1775,   0.0106,   0.0479,
0.1611,  0.0158,
            0.0694,  -0.0029,   0.0574,   0.0456,   0.2230,   0.0263,
-0.0035, -0.0504,
           -0.0874,  -0.0075,  -0.0768,   0.0972,   0.2305,  -0.0887,
-0.0183, -0.0309,
            0.0508,  -0.1091,   0.0168,   0.0876,   0.0792,  -0.0311,
0.0995,  0.0051,
            0.1542,  -0.1025,  -0.0348,  -0.0900,   0.0014,   0.2708,
-0.1139,  0.1901,
            0.0939,   0.0770,   0.1068,   0.0344,  -0.0694,  -0.0027,
-0.0331,  0.0068,
            0.0215,  -0.0770,  -0.0971,  -0.1128,  -0.0707,  -0.0665,
-0.0410,  0.1147,
           -0.0247,   0.0422,   0.2256,  -0.0823,  -0.0337,   0.1402,
-0.0221, -0.1136,
            0.0381,   0.1626,  -0.0179,   0.0718,   0.1595,  -0.0474,
-0.0936, -0.0926,
            0.1284,  -0.0505,  -0.0917,   0.1345,   0.0911,   0.0500,
-0.0025,  0.0358,
           -0.0310,  -0.0786,  -0.0563,  -0.0415,   0.1569,   0.0751,
0.0657,  0.0045,
            0.0159,  -0.0621,   0.1468,   0.0135,  -0.0785,   0.1852,
0.0313,  0.1515,
            0.0346,  -0.0134,  -0.0284,   0.2834,   0.1328,  -0.0947,
-0.0659, -0.0761,
           -0.0489,  -0.0870,  -0.0711,   0.1251,   0.0113,   0.1042,
-0.0393, -0.0654,
           -0.1216,  -0.0122,   0.1757,  -0.1020,   0.0664,  -0.0319,
-0.0129,  0.1800],
          [ 0.1106,  -0.0937,  -0.0092,   0.1917,   0.0344,  -0.0574,
-0.0471, -0.0670,
            0.0039,   0.0252,  -0.0223,   0.1197,  -0.0960,  -0.0231,
0.2418, -0.1042,
```

```
            0.0842,  -0.1050,   0.1541,  -0.0448,   0.1618,   0.1070,
0.0348,  0.0025,
           -0.0592,  -0.0785,  -0.0268,   0.0919,   0.1524,  -0.0223,
-0.0811, -0.0559,
           -0.1048,  -0.0208,  -0.0265,   0.1549,   0.1861,  -0.0571,
-0.0011, -0.1013,
            0.2210,   0.0385,  -0.1057,  -0.0768,  -0.1099,   0.1859,
0.0484,  0.1369,
            0.0963,   0.0939,   0.0411,   0.0689,  -0.1080,  -0.0180,
-0.0207,  0.0612,
           -0.0367,  -0.0984,  -0.0116,  -0.0473,  -0.0470,   0.0384,
0.0384, -0.0523,
           -0.0005,   0.0552,   0.1308,  -0.0318,  -0.0105,   0.1320,
-0.0194, -0.0639,
           -0.0820,   0.0592,  -0.0187,   0.1070,   0.0654,   0.0294,
-0.0965, -0.0650,
            0.0515,  -0.0148,  -0.0793,   0.2458,   0.0791,  -0.0109,
-0.0182,  0.0262,
            0.0164,  -0.0088,  -0.0967,  -0.0324,   0.2419,   0.1548,
-0.0292,  0.0358,
           -0.0056,  -0.0783,   0.0668,  -0.1173,   0.0256,   0.1521,
0.0597,  0.1572,
           -0.1052,  -0.0884,  -0.0691,   0.2276,   0.0922,  -0.0442,
-0.0010, -0.0316,
           -0.0580,   0.0468,  -0.0962,   0.1220,  -0.0323,   0.1163,
-0.0917, -0.1009,
           -0.0268,   0.0471,   0.1835,   0.0597,   0.0608,  -0.0974,
-0.0849,  0.2348],
          [ 0.2311,  -0.0998,   0.0260,   0.1148,   0.0236,   0.0162,
-0.0229,  0.0446,
           -0.0432,  -0.0013,  -0.1041,   0.1811,  -0.1030,  -0.0398,
0.2003, -0.0202,
            0.1342,  -0.0344,   0.1067,  -0.0720,   0.2333,   0.1040,
-0.1006, -0.0652,
           -0.0512,  -0.0796,   0.0091,   0.0815,   0.2357,  -0.0696,
-0.0447,  0.0027,
           -0.1040,  -0.0109,   0.0483,   0.1597,   0.0959,  -0.1036,
0.0844, -0.1032,
            0.2222,  -0.0729,  -0.1077,   0.0014,  -0.0917,   0.1727,
-0.0951,  0.1656,
            0.2376,   0.0747,   0.1684,   0.1440,  -0.0711,  -0.1005,
0.0106,  0.0298,
           -0.0797,   0.0442,   0.0781,  -0.0506,   0.0086,  -0.0956,
```

```
-0.0717,  0.0075,
           -0.0199,   0.1122,   0.1873,   0.0560,  -0.0840,   0.0652,
-0.0747,  0.0510,
            0.0121,   0.1262,  -0.0998,   0.1065,   0.1504,   0.0899,
0.0465, -0.0767,
            0.1137,  -0.0162,  -0.0088,   0.1028,   0.1118,   0.0450,
0.0076, -0.0120,
            0.0298,  -0.0501,  -0.0621,  -0.0703,   0.2061,   0.1298,
0.1243, -0.0357,
           -0.0585,  -0.1021,   0.1418,  -0.0133,  -0.0397,   0.1034,
0.1069,  0.2304,
           -0.0060,  -0.0869,  -0.0807,   0.3085,   0.1778,  -0.0639,
-0.1029,  0.0504,
            0.0249,  -0.0743,  -0.0116,   0.0585,  -0.0599,   0.0973,
-0.1042, -0.1063,
           -0.0293,   0.0534,   0.1653,  -0.0490,   0.0784,  -0.0411,
-0.0012,  0.1329]])
   fc1_bias ([ 1.4865e-01,   2.3202e-01,  -1.5668e-02,  -2.1498e-01,
1.0928e-01,
            1.3978e-01,    1.0371e-01,    5.9281e-02,    1.3394e-02,
2.4238e-01,
            3.4031e-01,   -2.1917e-02,   -2.4592e-01,   -2.2660e-01,
-1.6366e-01,
           -2.1865e-02,   -1.2870e-01,    3.7480e-01,    4.3707e-02,
-2.7053e-01,
            1.1674e-01,    1.2304e-01,   -1.3878e-01,   -1.8324e-02,
-1.2039e-01,
           -1.4152e-01,    2.5258e-01,    2.5432e-02,    2.7277e-01,
6.7678e-02,
            1.8150e-02,    1.2695e-02,    4.1218e-01,   -2.1317e-01,
6.3010e-02,
           -3.6125e-02,    2.0295e-01,   -7.5485e-02,    2.6876e-01,
-5.6261e-02,
            2.2030e-01,    1.8898e-01,    3.4795e-01,    2.8029e-01,
2.9850e-01,
           -9.3465e-02,   -2.8789e-02,    2.0428e-01,    3.0839e-01,
-1.6224e-01,
            3.1784e-01,   -1.0994e-01,   -2.0137e-02,    3.0977e-01,
-4.1523e-02,
           -2.2454e-01,    2.7377e-01,    2.2415e-01,   -2.5075e-01,
2.2089e-01,
            1.1766e-01,   -1.8574e-01,    2.6026e-01,   -2.7863e-01,
2.5092e-01,
```

```
           2.6924e-01,    2.6906e-03,    3.9066e-01,    6.3898e-02,
2.9719e-01,
           3.6197e-01,    1.9283e-01,  -1.3950e-01,    8.8387e-02,
6.5684e-02,
          -7.1172e-02,    7.8566e-02,  -8.4742e-02,  -2.1288e-01,
3.0926e-02,
          -1.3781e-01,  -4.2882e-04,    2.8749e-01,    2.8790e-01,
3.5703e-01,
           1.1140e-01,    4.7048e-02,  -3.2374e-01,    4.1768e-01,
-1.3514e-01,
          -1.0455e-01,    2.2338e-01,    2.6802e-01,  -2.8148e-01,
3.2300e-01,
           1.7824e-01,    2.5092e-01,    8.2733e-02,  -2.2681e-01,
-3.0281e-02,
          -7.7575e-03,  -3.3156e-02,  -1.1249e-01,    2.8257e-01,
-2.2317e-01,
          -1.4306e-01,  -1.8469e-01,    2.1157e-01,    2.6935e-01,
-2.7055e-01,
           3.0542e-01,  -2.6936e-01,  -2.5703e-01,  -4.8577e-02,
-4.4210e-02,
          -1.5235e-01,  -2.1151e-01,  -3.6809e-01,  -1.4408e-01,
3.1501e-02,
          -1.6972e-01,    1.2242e-01,    3.2363e-01,  -1.0339e-01,
9.8791e-03,
           4.3717e-01,  3.4291e-01, -2.7288e-01])
   fc2_bias ([ 0.0803,  0.0303,  0.1165,  0.1254,  0.1110,  0.0713,
-0.0354, -0.0293,
          -0.0968, -0.1013,  0.0118,  0.0488,  0.0692, -0.0539,
0.1382, -0.0590,
           0.1073, -0.0747,  0.1697,  0.1144,  0.0464,  0.0291,
0.0670, -0.0978,
           0.0269,  0.0381,  0.0970,  0.0471,  0.0058, -0.0508,
0.1198, -0.0758,
           0.0615,  0.0144,  0.0382,  0.1261,  0.0464,  0.1597,
0.0367,  0.0024,
           0.0080,  0.1061, -0.0137,  0.0069,  0.0962, -0.0181,
0.0662,  0.1427,
           0.0206,  0.0064,  0.0260,  0.0437,  0.0819,  0.0022,
0.0666,  0.0626,
          -0.0170,  0.1635,  0.0310,  0.0813,  0.0500, -0.0937,
-0.0226,  0.0092,
           0.0011,  0.1238,  0.0753,  0.0595,  0.0978,  0.1038,
0.0055, -0.0407,
```

```
           0.0059,   0.0368,  -0.0545,   0.1025,   0.1277,   0.0943,
0.0755,  0.0419,
           0.0744,  -0.0082,  -0.0394,   0.0582,  -0.0750,  -0.0378,
0.0467,  0.0930,
           0.0202,   0.0418,   0.0395,   0.0143,   0.0702,  -0.0500,
0.0319,  0.0231,
           0.0390,   0.0249,   0.0380,   0.0129,   0.0847,  -0.0327,
0.0057,  0.0606,
          -0.0119,  -0.0063,   0.0124,   0.1092,   0.0406,   0.1405,
-0.0472,  0.0935,
          -0.0102,  -0.0381,  -0.0420,   0.0887,  -0.0164,   0.0629,
0.1150,  0.0088,
          -0.0160,   0.0042,  -0.1263,   0.0241,  -0.0592,   0.0744,
-0.0091,  0.0916])
   fc3_bias ([ 0.0154,  0.1075, -0.0291,  0.1598,  0.0017,  0.1034,
0.0744, -0.0181,
           0.0428,  -0.0157,  -0.0030,   0.1354,   0.0187,  -0.0229,
0.1080,  0.0664,
           0.1282,   0.0009,  -0.0232,   0.1315,   0.1465,  -0.0174,
0.0347,  0.0924,
           0.0418,  -0.0227,   0.0054,   0.0049,   0.1487,   0.1030,
-0.0017, -0.0018,
           0.0378,  -0.0019,  -0.0768,   0.1561,  -0.0090,   0.1066,
-0.0430,  0.0246,
           0.1511,   0.0992,   0.0505,   0.1076,   0.1152,   0.1858,
0.1188,  0.0289,
           0.1120,  -0.0158,   0.0460,   0.0583,   0.1064,   0.0564,
0.0645, -0.0008,
           0.0474,   0.0243,  -0.0395,   0.0840,   0.1246,   0.1347,
0.1294, -0.0137,
          -0.0779,   0.0023,   0.1630,   0.0090,   0.0416,   0.0871,
0.1159,  0.0325,
          -0.0239,  -0.0147,   0.1178,   0.0465,  -0.0009,  -0.1177,
-0.0104,  0.0092,
           0.0342,   0.1354,   0.0755,   0.1054,  -0.0053,  -0.1010,
-0.0950, -0.0607,
          -0.0947,  -0.0309,   0.0661,  -0.0172,   0.1067,   0.0066,
-0.0694, -0.0750,
           0.0402,   0.1120,   0.0295,   0.0964,   0.0847,   0.0390,
0.0355,  0.0950,
          -0.0472,   0.0188,  -0.0525,   0.2334,   0.0292,   0.1252,
0.0863,  0.0649,
          -0.0007,  -0.0284,   0.1224,   0.0350,   0.0478,  -0.0431,
```

```
0.0220,  0.0776,
            0.1105,  -0.0458,    0.0554,    0.0177,    0.0329,  -0.0163,
0.0735, -0.0134])
    fc4_bias ([-0.0489, -0.1149,  0.0033,  0.0257])
```

附录 n

robot/__init__.py

```
    from gym import register



register(

    id='robot-v0',

    entry_point='robot.robot:Robot',

    reward_threshold=100.0,

)
```

robot/robot.py

```
    import gym
import b0RemoteApi
import math
import numpy as np
from gym import spaces
from gym.utils import seeding
from torch.utils.tensorboard import SummaryWriter
import os


lr = 1



def goal_distance(goal_a, goal_b):
    assert goal_a.shape == goal_b.shape
    return np.linalg.norm(goal_a - goal_b, axis=-1)
```

```
class Robot(gym.GoalEnv):
    """
    创建一个环境（environment），用于与仿真程序交互，判断是否完成目标
    """
    metadata = {'render.modes': 'ansi'}

    def __init__(self):
        self.tensorboard: SummaryWriter
        self.seed()  # 定义随机数生成的种子
        self.distance_threshold = 0.5  # 投掷后落入的范围，即设(1，1)为目标点，半径为0.5范围内都算投中
        self.default_state = (-90, 0, 0, 0, -1.9189, 0, 1.3255, 0, 0)  # 初始状态
        self.low = np.array([-3.819, -3.051, -2.985, 0, -2.5000, -2.5000, +0.8000, -3.140, -3.140], dtype=np.float16)
        self.high = np.array([+2.281, +3.052, +2.984, 1, +2.5000, +2.5000, +0.0000, +3.139, +3.139],
                             dtype=np.float16)  # 设定观测参数的上下界
```

```
        self.goal = self._sample_goal()
        self.observation_space = spaces.Dict(dict(
            desired_goal=spaces.Box(self.low, self.high, dtype=np.float16),
            achieved_goal=spaces.Box(self.low, self.high, dtype=np.float16),
            observation=spaces.Box(self.low, self.high, dtype=np.float16),
        ))
        self.action_space = spaces.Box(np.array([-3.819, -3.051, -2.985, 0],
dtype=np.float16),
                                       np.array([+2.281, +3.052, +2.984, 1],
dtype=np.float16), dtype='float32')

        self.client = bORemoteApi.RemoteApiClient('bORemoteApi_V-REP',
'bORemoteApi', 60, timeout=60)
        _, self.baseHandle = self.client.simxGetObjectHandle('Sawyer',
self.client.simxServiceCall())
        self.sawyerHandle = {}
        for j in range(2, 7, 2):
            _, self.sawyerHandle[j] = self.client.simxGetObjectHandle('Sawyer_joint' +
str(j),
```

```
self.client.simxServiceCall())

        _, self.gripperHandle =

self.client.simxGetObjectHandle('BaxterGripper_closeJoint',



self.client.simxServiceCall())

        _, self.bottleHandle = self.client.simxGetObjectHandle('Cylinder',

self.client.simxServiceCall())

        _, self.floorHandle = self.client.simxGetObjectHandle('Floor',

self.client.simxServiceCall())

        _, self.sensorHandle =

self.client.simxGetObjectHandle('BaxterGripper_attachPoint',



self.client.simxServiceCall())

        _, self.deskHandle = self.client.simxGetObjectHandle('customizableTable',

self.client.simxServiceCall())

        self.client.simxSetJointTargetVelocity(self.gripperHandle, -5,

self.client.simxDefaultPublisher())

        self.start_time =

self.client.simxGetServerTimeInMs(self.client.simxServiceCall())[1]

        self.time = self.client.simxGetServerTimeInMs(self.client.simxServiceCall())[1]
```

```
        self.obs = self._get_obs()  # 沟通仿真软件的部分


    def step(self, action):
        """
        执行每一步动作
        动作指令包含：
        【机械臂 2 角度，机械臂 4 角度，机械臂 6 角度，投掷指令】
        """
        action = np.clip(action, self.action_space.low, self.action_space.high)
        try:
            time = self.time - self.start_time
            self.tensorboard.add_scalars('Action/Sawyer',
                                         {'joint2': action[0], 'joint4': action[1], 'joint6':
action[2]}, time)
            self.tensorboard.add_scalar('Action/Gripper', action[3], time)
        except Exception:
            pass
        self.client.simxStartSimulation(self.client.simxDefaultPublisher())
        self._set_action(action)
        # [jpos_sawyer2, jpos_sawyer4, jpos_sawyer6, jpos_gripper]


        self.obs = self._get_obs()
```

```
        info = {'is_success': self._is_success(self.obs['achieved_goal'], self.goal), 'gripper':
action[3] >= 0.5,
                'action': action}
        reward = self.compute_reward(self.obs['achieved_goal'], self.goal, info)
        if action[3] >= 0.5 or \
                self.client.simxCheckCollision(self.floorHandle, self.bottleHandle,
self.client.simxServiceCall())[1]:
            done = True
            self.time =
self.client.simxGetServerTimeInMs(self.client.simxServiceCall())[1]
            self.client.simxSleep(2)
        else:
            done = False
        # 判断本次仿真是否结束，条件一：水杯抛掷出去了，条件二：水杯落地（即与地面发生碰
撞）

        self.client.simxPauseSimulation(self.client.simxDefaultPublisher())
        return self.obs, reward, done, info

    def compute_reward(self, achieved_goal, desired_goal, info):
        """
```

```
        计算奖赏，影响因素有：
        每次仿真的时间（-）
        与目标点的距离（--）
        抛出水瓶（+）
        水瓶直立（++）
        与目标点重合（++）
        机械臂运动幅度（+）
        水瓶落地（--）
        """
        reward = -(self.client.simxGetServerTimeInMs(self.client.simxServiceCall())[1] -
self.time) / 1000
        distance = goal_distance(self.goal[:3], np.array([-1.9189, 0, 1.3255]))
        reward += distance - goal_distance(self.goal[:3], achieved_goal[:3])

        if info['gripper']:
            reward += 25
            if achieved_goal[3] == 0 and achieved_goal[4] == 0:
                reward += 50
                if distance == 0:
                    reward += 50
        reward += math.sqrt(info['action'][0] ** 2 + info['action'][1] ** 2 +
```

RL.py

```
    import os
import gym
import tianshou as ts
from tianshou.data import Batch
import torch
import torch.nn as nn
from torch.utils.tensorboard import SummaryWriter
import numpy as np
import robot


task = 'robot-v0'
lr = 1e-3  # 网络参数的学习率
gamma = 0.9
n_step = 4
eps_train, eps_test = 0.1, 0.05
num_epoch = 1000
step_per_epoch = 100  # 每一个 epoch 执行动作的次数
step_per_collect = 10  # 每 10 个动作做为一组，这一组不一定是一个完整的运动轨迹
target_freq = 320
batch_size = 64
```

```
buffer_size = 20000



class Net(nn.Module):  # 定义一个全连接神经网络
    def __init__(self, state_shape, action_shape):
        super().__init__()
        self.model = nn.Sequential(*[
            nn.Linear(int(np.prod(state_shape)), 128),
            nn.ReLU(inplace=True),
            nn.Linear(128, 128), nn.ReLU(inplace=True),
            nn.Linear(128, 128), nn.ReLU(inplace=True),
            nn.Linear(128, int(np.prod(action_shape)))
        ])

    def forward(self, s, state=None, info={}):
        o = s.observation
        # s.achieved_goal, s.desired_goal are also available
        if not isinstance(o, torch.Tensor):
            o = torch.tensor(o, dtype=torch.float)
        batch = o.shape[0]
        logits = self.model(o.view(batch, -1))
```