

说明文档

代码功能概述:

项目使用深度强化学习算法（DDQN 和 MADDPG）实现了一个无人机辅助车辆任务处理系统。系统通过双层决策机制：外层使用 DDQN 算法决定部署的无人机数量，内层使用 MADDPG 算法优化无人机位置和任务分配。系统能够自动为行驶中的车辆提供计算和通信服务，包括任务处理、资源调度、位置优化等功能，同时考虑了通信延迟、能量消耗、任务处理效率等多个优化目标，实现了一个完整的智能交通服务系统。

该项目的关键组成如下：

- **主程序模块**

Main.py: 作为主程序入口，负责控制整个项目的流程，协调各个模块之间的交互，同时管理训练过程并记录训练结果。

- **实体类模块**

UAV.py: 定义无人机类，包含无人机的属性（如位置、电量等）和行为（如移动、处理任务等），还负责计算无人机的电量消耗和收益。

vehicle.py: 定义车辆用户类，包含车辆用户的属性（如位置、任务等）和行为（如生成任务、移动、决策任务卸载等）。

- **环境模块**

inter_layer_make_env.py: 实现内层决策过程，管理无人机和用户之间的交互，计算奖励和状态转换，为智能体提供训练环境。

- **工具模块**

myUtils.py: 集合了各种工具函数，提供辅助功能，如数据计算和转换，以及实现约束检查等。

myConfig.py: 存储系统的参数、常量和环境配置信息，可以设置迭代次数，车辆用户数量，每个用户产生的任务数等参数。

该项目的关键输入输出如下：

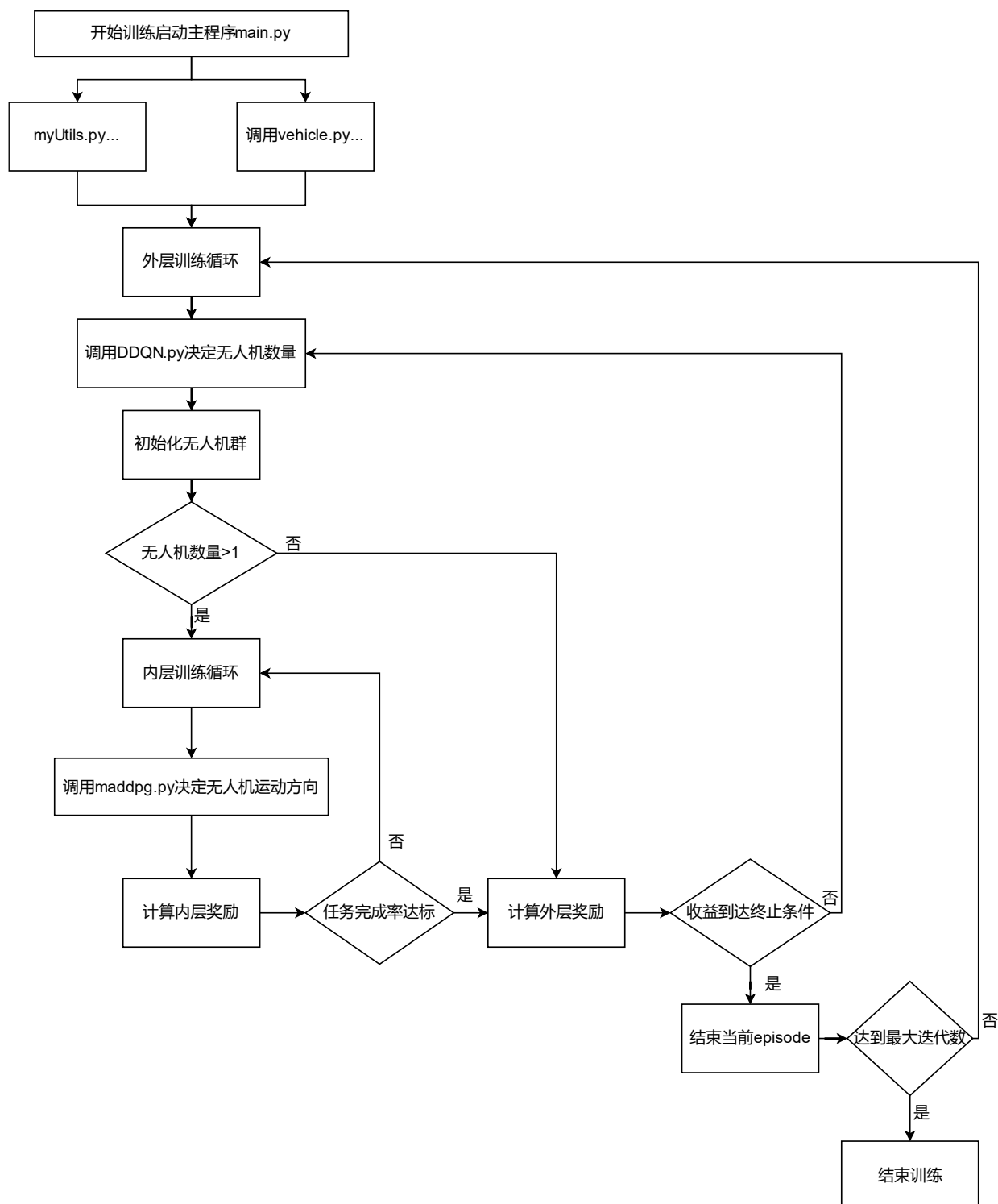
输入变量	变量含义	数值
Episode_num	训练迭代次数	3000
Users_num	车辆用户数量	100
User_tasks_num	每个用户的任务数	5
service_distance	无人机服务覆盖范围	400
d_uav_min	无人机最小间距	1000
uav_movable_range	无人机移动范围	10000

输出：

其决策结果是通过多智能体强化学习算法优化无人机的部署数量，任务分配和移动策略，以提高任务处理效率、最大化无人机的收益

其性能指标包括：任务完成的总收益，任务处理数量，任务处理时间

其训练流程如下所示:



代码运行说明：

1.环境配置参考

Python==3.8.20, Numpy==1.24.4, Pytorch==1.8.1

注意 Pytorch, Numpy, Python 的版本兼容问题

2.在终端或者命令行中启动训练过程

```
python main.py
```