计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
[doi: 10.15888/j.cnki.csa.009755] [CSTR: 32024.14.csa.009755]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 TD3 的无人机计算卸载算法①

徐　飞, 赵前奔, 杨　雪

(西安工业大学 计算机科学与工程学院, 西安 710021)
通信作者: 赵前奔, E-mail: 503223597@qq.com

摘　要: 无人机 (unmanned aerial vehicle, UAV) 搭载边缘服务器构成移动边缘服务器, 可以在一些基站难以部署的场景下为用户设备 (user equipment, UE) 提供计算服务, 借助深度强化学习对智能体进行训练, 能够在连续复杂的状态空间中制定合理的卸载决策, 将用户产生的计算密集型任务部分卸载至边缘服务器处执行, 提高系统的续航和响应时间, 但目前的深度强化学习算法所使用的全连接神经网络无法较好地处理 UAV 辅助移动边缘计算 (mobile edge computing, MEC) 场景下的时间序列数据, 算法的智能体训练效率低, 决策性能差, 针对上述问题, 本文以最小化 UAV 辅助 MEC 系统总时延为目标, 提出了一种基于长短期记忆网络的双延迟深度确定性策略梯度算法 (twin delayed deep deterministic policy gradient algorithm based on long short term memory, LSTM-TD3), 利用 LSTM 改进 TD3 算法的 Actor-Critic 网络结构, 将网络划分成 3 部分: 包含 LSTM 的记忆提取单元, 当前特征提取单元, 以及感知整合单元; 并在改进了经验池中的样本数据, 定义了历史数据, 使记忆提取单元能够得到更好的训练效果. 仿真结果表明, 与 AC 算法、DQN 算法和 DDPG 算法相比, LSTM-TD3 算法在以系统最小总时延为目标对卸载策略进行优化时具有最好的性能.

关键词: 移动边缘计算; 无人机; 深度强化学习; 计算卸载

引用格式: 徐飞,赵前奔,杨雪.基于 TD3 的无人机计算卸载算法.计算机系统应用. http://www.c-s-a.org.cn/1003-3254/9755.html

UAV Computation Offloading Algorithm Based on TD3

XU Fei, ZHAO Qian-Ben, YANG Xue

(School of Computer Science and Engineering, Xi'an Technological University, Xi'an 710021, China)

Abstract: Unmanned aerial vehicle (UAV) is equipped with an edge server to constitute a mobile edge server. It can provide computing services for user equipment (UE) in some scenarios where base stations are difficult to deploy. With the help of deep reinforcement learning to train the intelligent body, it can formulate reasonable offloading decisions in a continuous and complex state space. It can also offload partial computing-intensive missions produced by users to edge servers for execution, thus improving the working and responding time of the system. However, at the moment, the fully connected neural networks used by the deep reinforcement learning algorithm are unable to handle the time-series data in the scenarios of UAV-assisted mobile edge computing (MEC). In addition, the training efficiency of the algorithm is low, and the decision-making performance is poor. To address the above problems, this study proposes a twin delayed deep deterministic policy gradient algorithm based on long short term memory (LSTM-TD3), using LSTM to improve the Actor-Critic network structure of the TD3 algorithm. In this way, the network is divided into three parts: the memory extraction unit containing LSTM, the current feature extraction unit, and the perceptual integration unit. Besides, the sample data in the experience pool are improved, and the historical data are defined, which provides the memory

① 基金项目: 咸阳市科技局重点研发计划 (2023ZDYF-NY-0019); 西安市碑林区科技计划 (GX2137)
收稿时间: 2024-07-25; 修改时间: 2024-08-20; 采用时间: 2024-08-29; csa 在线出版时间: 2024-11-15

extraction unit with a better training effect. Simulation results show that, compared with the AC algorithm, the DQN algorithm, and the DDPG algorithm, the LSTM-TD3 algorithm has the best performance when optimizing the offloading strategy with the minimum total delay of the system as the target.

Key words: mobile edge computing (MEC); unmanned aerial vehicle (UAV); deep reinforcement learning; computation offloading

随着移动互联网技术的更新迭代和用户需求的多元化发展, 越来越多的物联网设备被部署到网络中, 大量设备接入无线网络并且依靠在线资源处理各种任务. 物联网设备因其本身的算力有限, 在处理部分任务时需要借助云计算技术, 通过中央云服务器对任务进行处理, 然而, 将大量计算任务传输至中央云服务器上处理时, 往往会因为数据的长距离传输和任务拥堵导致设备的处理时延和能耗较高. 移动边缘计算的引入可用于解决上述问题, MEC 通过将边缘服务器部署在离终端用户更近的边缘端, 利用服务器算力资源为用户提供计算服务, 有效降低了任务处理延迟和能耗[1]. 但 MEC 服务器位置固定, 无法随时部署, 缺少解决突发性事务的能力, 无人机 (UAV) 凭借高机动性、易于部署和快速响应的特点, 通过搭载边缘服务器构成移动边缘服务器, 辅助 MEC 系统, 在一些地面基站难以部署的场景中, 为用户设备提供计算服务, 能够有效减少用户设备与云的频繁通信和任务上传所产生的时延和能耗[2].

移动边缘计算中的计算卸载技术, 是指用户设备将一部分或全部计算任务交给边缘服务器处理的技术, 以解决移动设备在资源存储、计算性能及能效等方面的不足. 在 UAV 辅助移动边缘计算场景下, 移动边缘服务器根据每个时隙时的用户和环境状态信息, 由系统给定的评价指标优化计算卸载决策, 从而为地面移动终端提供计算服务, 合理的计算卸载策略能够极大的提高移动边缘计算系统的性能, 是 MEC 的关键技术之一. 因此如何根据当前环境状态, 设计合适的算法优化计算卸载策略, 是当前移动边缘计算卸载领域的重点.

深度强化学习将深度学习和强化学习相结合, 利用深度神经网络拟合强化学习的价值函数和策略函数, 使智能体能够从连续复杂的状态空间中实时提取可用于训练的特征信息, 从而实现对卸载策略的动态优化调整, 因此当前无人机辅助移动边缘计算卸载领域的研究热点主要集中于利用深度强化学习算法对卸载策略进行优化. 本文考虑在单 UAV 对多地面终端场景下, 利用提出的 LSTM-TD3 算法对智能体进行训练, 以最小化移动边缘计算系统的总计算时延为目标优化计算卸载策略.

1 相关工作

在现有的文献中, 有许多关于 UAV 辅助移动 MEC 中的计算卸载方面的研究工作. 这些文献按照采样的算法可以分为基于传统算法的计算卸载研究和基于深度强化学习的计算卸载研究.

基于传统算法的计算卸载研究中, Wang 等人[3]考虑到 UAV 的移动轨迹可能对移动卸载产生负面影响, 在考虑移动轨迹和计算资源分配的情况下研究卸载问题, 他们以用户效用最大化为目标, 将非线性分式规划 (NFP) 和逐次凸逼近 (SCA) 相结合, 提出了一种近似最优算法. Liu 等人[4]构建了多 UAV 辅助的 MEC 系统, 考虑了计算效率和能量消耗, 提出了一个计算效率最大化问题. 他们对用户关联和 UAV 的轨迹调度进行联合优化. 由于问题的非凸性和变量间的耦合性, 提出了一种双环结构迭代优化算法来寻找最优解.

UAV 辅助通信系统中, 对于通信条件的时变性、任务卸载问题的非凸性, 传统算法并不能很好的解决, 同时, 广泛用于学习和优化无人机辅助 MEC 系统各种问题的机器学习算法需要提供足够的数据样本, 这对于决策问题是不现实的[5].

相比之下, DRL 可以从实际环境中动态采集并学习数据样本, 能够很好地解决这一决策问题. 文献[6]针对动态 ICV (intelligent connected vehicle) 环境中的顺序决策问题, 提出了一种新的基于分布式 DRL 的 P-D3QN 方法, 该方法使用优先级体验回放策略和对偶双深度 Q 网络 (D3QN) 算法来有效地解决最优任务卸载策略. 文献[7]针对无人机有限的计算资源和特定任务的数据依赖性对智能体训练效率的影响, 开发了一

种基于饱和训练 SAC 的无人机辅助依赖感知计算卸载算法 (STS-UDCO). STS-UDCO 学习 CO 策略的熵和值, 以有效地近似最优解, STS-UDCO 中提出的自适应饱和训练规则根据当前拟合状态动态控制临界点的更新频率, 以提高训练稳定性. 文献[8]以在最大可容忍延迟和计算限制的情况下最大限度地降低系统能耗为目标, 为了解决此类混合整数非线性规划 (MINLP) 问题, 提出了一种结合逐次凸近似 (SCA) 的深度强化学习 (DRL) 算法, 以寻求低复杂度的接近最优解. 具体而言, 通过 DRL 获得二进制服务分配和连续任务拆分, 而通过 SCA 依次联合优化轨迹规划和资源调度, 以加快收敛速度. 文献[9]提出了 DRL-UCTO 算法, 使用两个带权值的深度神经网络, 分别称为主网络和目标网络, 代替原 Q-learning 算法中的 Q-table, 解决了 Q-learning 算法无法在高维动作空间中使用的问题. 文献[10]建立了多目标马尔可夫决策过程 (MOMDP) 模型, 将强化学习的标量奖励扩展为向量奖励, 分别对应延迟和能耗, 同时将双深度 Q 网络 (double DQN) 和对偶深度网络 (dueling deep Q network) 相结合, 解决了 DQN 中过高估值的问题. 文献[11]以最小化系统总时延和能耗加权和为目标, 在 TD3 的基础上, 设计了一种用于将连续动作空间转换为离散动作空间的动作转换算法, 此外, 此外, 为了以非迭代的方式求解公式问题, 设计了一种有效的启发式算法. 文献[12]针对移动边缘计算中的多用户协同任务卸载场景, 利用双向长短期记忆网络 (bidirectional long short-term memory network, Bi-LSTM) 提取状态向量动态时序依赖关系的特征信息, 结合强化学习方法建立高维状态与动作之间的关系映射, 并设计了一种动态优先级协同采样算法, 用于提高多智能体的协同性.

从上述相关文献可以看出, 目前基于深度强化学习的计算卸载研究中, 仍存在一些可以优化的问题.

(1) 深度强化学习中的全连接神经网络无法较好处理无人机辅助移动边缘计算场景下的时间序列数据. 单无人机对多地面终端场景下, 移动边缘服务器在连续时隙中采集到的样本数据具有时间序列特性, 但目前的深度强化学习算法大多仅使用全连接神经网络来近似值函数, 无法较好地处理此类数据, 因此面对复杂场景下的无人机计算卸载问题, 传统的深度强化学习算法无法较好的对智能体进行训练以获取最优的计算卸载策略, 导致系统的总时延较高.

(2) 在智能体的训练过程中, 现有的深度强化学习算法对时间序列数据时训练效率低, 收敛性能差. 针对深度强化学习的全连接神经网络无法处理时间序列数据的问题, 已有部分文献进行了研究, 例如前文提到的文献[12]引入了 LSTM 网络来处理时间序列数据, 但问题在于, 目前大多数算法但未能提出较为合理的网络结构, 只是较为简单地将 LSTM 网络插入到原有的 Actor-Critic 网络中, 在面对单 UAV 对多地面终端场景下的复杂环境时, 高维的历史数据的引入将导致数据维度和数据量大幅增加, 简单地插入 LSTM 网络会导致较高的计算和训练成本, 降低算法的训练效率, 从而降低智能体在处理系统总计算时延方面的性能.

针对上述提到的现有文献中所存在的不足, 本文主要工作内容和创新点如下.

(1) 针对单 UAV 对多终端用户场景, 综合考虑环境阻碍、系统能量约束等条件, 构建双层系统模型, 包括通信模型、计算模型和约束模型, 以小化系统总任务处理时延为目标, 优化 UAV 的计算卸载策略.

(2) 针对上述计算卸载场景建立马尔可夫决策过程 (Markov decision process, MDP), 考虑到 TD3 的全连接网络结构无法较好的处理时间序列数据的问题, 提出了一种基于 LSTM 的双延迟深度确定性策略梯度算法 (LSTM-TD3) 求解上述马尔可夫决策过程, 该算法利用 LSTM 对 TD3 算法的 Actor-Critic 网络结构进行了改进, 将 Actor 和 Critic 网络划分成 3 部分, 分别为: 包含 LSTM 的记忆提取单元, 当前特征提取单元, 以及感知整合单元; 并在经验池中增加了历史数据, 使记忆提取单元能够得到更好的训练效果; 同时改进了经验池, 在经验池存储的样本数据中增加了历史数据, 使算法网络能够得到更好的训练效果.

(3) 将本文提出的算法与不同的深度强化学习算法进行仿真实验对比, 实验结果表明, 本文算法相较于目前所提出的 DRL 算法, 如 DDPG、AC、DQN, 在不同环境参数和通信条件下具有更好的性能表现.

2 系统框架

为了对比分析 LSTM-TD3 算法在单无人机对多终端场景下的性能, 本文搭建了一个 MEC 仿真环境, 该仿真环境包含智能体、地面用户、环境和经验池 4 个模块.

(1) 用户

即地面终端设备, 包含移动模型, 在每个时隙, 用户根据移动模型在一定范围内移动, 并和移动边缘服务器建立通信, 随机产生要卸载至移动边缘服务器上的计算密集型任务.

(2) 智能体

无人机搭载边缘服务器, 构成移动边缘服务器作为系统的智能体, 包含通信模型, 计算卸载模型和移动模型, 在每个时隙内, 移动边缘服务器通过移动模型在一定范围内移动, 通过通信模型与地面用户建立连接, 并通过计算模型计算当前卸载任务的总时延和能耗, 利用上述模型收集状态信息, 奖励值和策略信息, 作为训练智能体网络的数据样本.

(3) 环境

本文定义了一系列环境参数, 包括带宽, 信道增益, 噪声功率, 上行链路传输功率、UAV 质量、边缘服务器 CPU 转数等, 来模拟单无人机对多地面终端的计算卸载场景.

(4) 经验池

存储移动边缘服务器在连续时隙内采集到的样本数据, 用于后续对智能体神经网络的训练.

系统框架具体如图 1 所示, 在时隙 T 时刻, 智能体根据状态空间 S, 基于当前环境状态 $s(t)$, 根据 Actor 网络选取动作 $a(t)$, 对环境进行探索, 在探索和训练的过程中, 马尔可夫决策过程对智能体的状态、动作和奖励进行了定义和约束, 在此基础上, 智能体通过系统模型中定义的移动模型, 通信模型, 计算模型等模型, 对每个时隙内的能耗时延进行计算, 并及时更新马尔可夫决策过程所定义的状态、动作和奖励函数, 从而实现对智能体的训练.

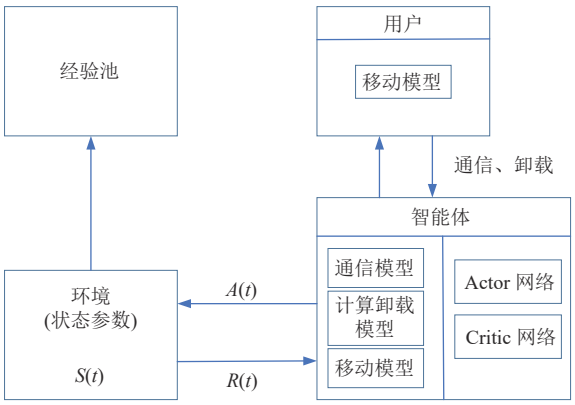

图 1 系统框架

根据系统框架, 为实现强化学习对智能体的训练过程, 本章第 3 节对系统模型和马尔可夫决策过程进行了定义, 并对提出的算法进行了详细介绍.

3 系统模型

本文提出的单无人机对多地面终端的移动边缘计算系统模型如图 2 所示. 系统包含搭载边缘服务器的单无人机, 以及多个地面终端用户. 无人机在执行计算卸载任务时, 悬停在固定高度的平面, 为地面用户提供服务, 用户可将计算密集型任务部分卸载至边缘服务器.

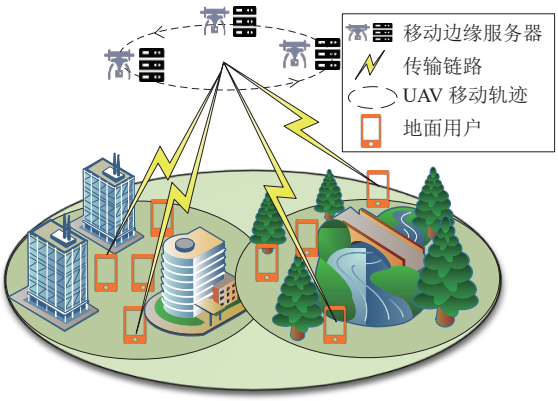

图 2 单 UAV 系统模型

单无人机系统灵活性高, 易于部署, 可在固定高度的一定范围内灵活移动, 及时为服务范围内的地面终端用户提高计算服务, 地面终端用户通过将计算密集型任务部分卸载至无人机搭载的边缘服务器中进行计算, 能够极大的提高自身的续航和系统的计算时延, 从而显著提高地面终端用户的任务响应时间.

3.1 通信模型

在整个 UAV 提供算力服务的过程中, 将通信周期 T 划分为 I 个时隙, 以向所有终端提供服务. 由于用户设备 (UE) 会在区域范围内以较低时速进行随机运动, 对于每个时隙, 无人机的运动分为飞行和悬停两部分, 当 UAV 在固定位置悬停时, 会与其中一个终端建立通信连接. 终端用户可选择将计算任务部分卸载至 UAV 搭载的边缘服务器上执行, 剩下的任务则在本地执行.

无人机在离地 H 的高度平面中执行飞行和计算卸载任务, 时隙 $i\in\{1, 2, \cdots, I\}$ 内, 无人机的起始坐标和终点坐标分别为 $q(i)=[x(i), y(i)]^{\mathrm{T}}\in R^{2\times1}$ 和 $q(i+1)=[x(i+1), y(i+1)]^{\mathrm{T}}\in R^{2\times1}$, $UEk\in\{1, 2, \cdots, K\}$ 的坐标为 $P_k(i)=[x_k(i),$

$y_k(i)]^{\mathrm{T}} \in R^{2\times 1}$.

无人机与第 k 个 UE 的信道增益可表示为:

$$g_k(i) = \alpha_0 d_k^{-2}(i) = \frac{\alpha_0}{\|q(i+1) - p_k(i)\|^2 + H^2} \tag{1}$$

其中, α_0 为参考距离 d=1 m 处的信道增益, $d_k(i)$ 为无人机与 UEk 之间的欧氏距离. 由于障碍物的阻塞, 根据香农公式, 无线传输速率为:

$$r_k(i) = B\log_2\left(1 + \frac{P_{\mathrm{up}} g_k(i)}{\sigma^2 + f_k(i) P_{\mathrm{NLOS}}}\right) \tag{2}$$

其中, B 为通信带宽, P_{up} 为上行链路 UE 的发射功率, σ^2 为噪声功率, P_{NLOS} 为传输损耗, $f_k(i)$ 为第 i 时段 UAV 与 UEk 之间是否存在阻塞 (0 为无阻塞, 1 为阻塞) 的指标.

3.2 用户移动模型

地面移动用户只能在长为 L, 宽为 W 的地面范围内移动. 地面移动用户 $k\in K$ 的水平坐标表示为 $p_k=[x_k, y_k]^{\mathrm{T}}$. 在下一个时隙, 地面用户移动到了新的位置 $p(i)$:

$$p(i+1) = [x_k(i+1), y_k(i+1)]^{\mathrm{T}} \tag{3}$$

$$\begin{cases} x_k(i+1) = x_k(i) + d_{\mathrm{ue},k}\cos\beta(i) \\ y_k(i+1) = y_k(i) + d_{\mathrm{ue},k}\sin\beta(i) \end{cases} \tag{4}$$

其中, $d_{\mathrm{ue},k}=t_{\mathrm{ue}}\cdot\theta_{\mathrm{ue}}$, $t_{\mathrm{ue}}\in[0,t_{\max}]$, 角度 $\theta_{\mathrm{ue}}\in[0, 2\pi]$, 地面用户的速度为固定值 v_{ue}.

3.3 计算模型

系统在每个时隙内采用部分卸载策略. $R_k\in[0, 1]$为任务卸载比, $1-R_k$ 为本地执行任务. 在时隙 i 时, 第 k 个 UE 的本地任务时延表示为:

$$t_{\mathrm{local},k}(i) = \frac{(1-R_k(i))D_k(i)s}{f_{\mathrm{ue}}} \tag{5}$$

其中, $D_k(i)$ 为第 k 个 UE 的任务大小, s 表示处理每个单位字节所需的 CPU 周期, f_{ue} 为 UE 算力. UAV 在每个时隙的起始位置为 $q(i+1)$:

$$q(i+1) = [x(i+1), y(i+1)]^{\mathrm{T}} \tag{6}$$

$$\begin{cases} x(i+1) = x(i) + v(i)t_{\mathrm{fly}}\cos\beta(i) \\ y(i+1) = y(i) + v(i)t_{\mathrm{fly}}\sin\beta(i) \end{cases} \tag{7}$$

在速度 $v(i)\in[0, v_{\max}]$和角度 $\beta(i)\in[0, 2\pi]$条件下. 当前此飞行所消耗的能量可以表示为:

$$E_{\mathrm{fly}}(i) = \phi\|v(i)\|^2 \tag{8}$$

其中, ϕ=0.5$M_{\mathrm{UAV}}t_{\mathrm{fly}}$, M 为无人机的载荷相关参数, t_{fly} 为 UAV 飞行时间. 在 MEC 系统中, 通常忽略服务器所提供的十分微小的计算结果[13]. 因此可忽略下行链路所带来的传输延迟, 系统的处理时延分为两部分. 首先是传输时延:

$$t_{\mathrm{tr},k}(i) = \frac{R_k(i)D_k(i)}{r_k(i)} \tag{9}$$

其次是服务器上的计算时延, 可以表示为:

$$t_{\mathrm{UAV},k}(i) = \frac{R_k(i)D_k(i)s}{f_{\mathrm{UAV}}} \tag{10}$$

其中, f_{UAV} 为服务器 CPU 的计算频率. 同时在时隙 i 内,也可将任务卸载给服务器的耗能分为两部分, 用于传输的能耗, 以及用于计算的能耗. 在 MEC 服务器上进行计算时, 其功耗可表示为:

$$P_{\mathrm{UAV},k}(i) = k f_{\mathrm{UAV}}^3 \tag{11}$$

则 MEC 服务器在时隙 i 的能耗为:

$$E_{\mathrm{UAV},k}(i) = P_{\mathrm{UAV},k}(i)t_{\mathrm{UAV},k}(i) = k f_{\mathrm{UAV}}^2 R_k(i)D_k(i)s \tag{12}$$

3.4 问题约束

基于上述模型, 本文提出了 UAV 辅助 MEC 系统的优化问题. 本文的目标是最小化所有终端的最大处理延迟, 为了更有效地利用有限的计算资源, 我们采用联合优化策略, 考虑了用户调度、无人机机动性以及系统内资源分配. 具体而言, 我们要解决的优化问题可以表示为:

$$\begin{cases} P1: \min \sum_{i=1}^{I}\sum_{k=1}^{K} \alpha_k(i) \max\left\{ \begin{matrix} t_{\mathrm{local},k}(i) \\ t_{\mathrm{ucv},k}(i) + t_{\mathrm{tr},k}(i) \end{matrix} \right\} \\ C1: \alpha_k(i) \in \{0,1\} \\ C2: \sum_{k=1}^{K} \alpha_k(i) \\ C3: R_k(i) = (0,1) \\ C4: q(i) \in \left\{ x(i), y(i) \mid x(i)\in[0,L], y(i)\in[0,W] \right\} \\ C5: p_n(i) \in \left\{ x_n(i), y_n(i) \mid x_n(i)\in[0,L], y_n(i)\in[0,W] \right\} \\ C6: f_k(i) = \{0,1\} \\ C7: \sum_{i=1}^{I} (E_{\mathrm{fk}}(i) + E_{\mathrm{uucv},n}(i) + E_{\mathrm{tr};n}(i)) \leqslant E_b \\ C8: \sum_{i=1}^{I}\sum_{k=1}^{K} \alpha_k(i) D_k(i) = D \end{cases} \tag{13}$$

其中, $C1$ 和 $C2$ 能够保证时隙 i 中, UAV 只为一个用户提供计算卸载服务. $C3$ 为任务卸载比范围约束. $C4$ 和

$C5$ 为 UAV 和 UE 的活动范围约束. $C6$ 表示时隙 i 内无人机与终端之间无线信道的阻塞. $C7$ 为 UAV 的电池容量约束. $C8$ 表示整个时段内的总计算任务量.

3.5 单无人机马尔科夫决策模型

本文将问题构建为 MDP 以应用 DRL 解决.

(1) 状态空间

状态空间包括无人机的状态、UE 的状态及其相应的环境参数, 可定义为:

$$s_i=\left(E_{\text{battery}}(i),q(i),\sum_{k=1}^{K}p_k(i),D_{\text{remain}}(i),\sum_{k=1}^{K}D_k(i),\sum_{k=1}^{K}f_k(i)\right) \tag{14}$$

$E_{\text{battery}}(i)$ 为 i 时隙内的电池容量, $q(i)$ 为无人机的位置信息, $p_k(i)$ 表示第 i 个时隙内第 n 个 UE 的位置, $D_{\text{remain}}(i)$ 为总任务余量, $D_k(i)$ 表示 UEk 在第 i 个时隙随机生成的任务大小, $f_k(i)$ 表示 UEk 的信号是否被障碍物阻塞.

(2) 动作空间

Agent 收集状态信息, 在第 i 个时隙中选择需要服务的 UEk、无人机的飞行角度、无人机的飞行速度、任务卸载率等动作. 动作 a_i 表示为:

$$a_i=\left(k(i),\beta(i),v(i),R_k(i)|i\in[1,K]\right) \tag{15}$$

值得注意的是, LSTM-TD3 中的 Actor 网络输出连续的动作. $k(i)\in[0, K]$代表 UAV 在当前时隙内选择的 UE 编号, 将其值离散化, 即 $k(i)=0$, 则 $k'=1$; 如果 $k(i)\neq 0$, 则 $k'=\lceil k(i)\rceil$, 其中, $\lceil\cdot\rceil$表示向上取整. $\beta(i)\in[0, 2\pi]$, $v(i)\in[0, v_{\max}]$, 表示无人机的动作可应用于连续的动作空间.

(3) 奖励函数

本文的目标是在式 (11) 的约束下, 通过最小化时延来最大化奖励函数:

$$r_i=r(s_i,a_i)=-\tau_{\text{delay}}(i) \tag{16}$$

在时隙 i 时的处理时延如下:

$$\tau_{\text{delay}}(i)=\sum_{k=1}^{K}\alpha_k(i)\max\left\{\begin{matrix}t_{\text{local},k}(i),\\ t_{\text{UAV},k}(i)+t_{\text{tr},k}(i)\end{matrix}\right\} \tag{17}$$

如果 $k=k'$, 则 $\alpha_k(i)=1$; 否则 $\alpha_k(i)=0$. 利用 LSTM-TD3 算法, 可以找到使 Q 值最大化的动作. 系统的长期平均报酬可以用 Bellman 方程表示为:

$$Q_\mu(s_i,a_i)=E_\mu[r(s_i,a_i)+\gamma Q_\mu(s_{i+1},\mu(s_{i+1}))] \tag{18}$$

4 LSTM-TD3 任务卸载算法

基于上述 MDP 的构建, 提出 LSTM-TD3 任务卸载算法, 利用 LSTM 对 TD3 的 Actor-Critic 网络进行改进.

4.1 LSTM 网络

复杂场景下的无人机辅助 MEC 系统所收集到的样本数据具有时间序列特性, 在连续时隙下, 历史状态和决策会对当前无人机的探索产生影响, 进而影响下一时隙无人机的状态和决策, 如果能从历史数据中获取经验, 智能体在当前状态下将会做出更好的决策, 然而传统 DRL 算法中的全连接神经网络结构不能较好的处理时间序列数据, 面对此类数据时训练效率低, 训练效果差, 因此需要对算法的网络结构进行改进优化.

长短期记忆网络 (LSTM) 利用门结构对信息进行了选择性输入, 增加了细胞状态, 直接传递和更新原始细胞状态信息, 从而更好地捕捉和处理时间序列数据中的长期依赖关系[14]. 因此, 面对无人机探索环境时收集到的一系列时间相关性数据, LSTM 能够通过提取并保留之前时隙中的历史数据特征信息, 使算法智能体具备记忆功能, 智能体的 Actor-Critic 网络能够从样本数据的学习训练中积累历史经验, 以保证在当前状态下做出更加合理的决策.

LSTM 的引入不仅能够提高算法对智能体的训练效率, 合理的决策也能降低系统的计算时延和能耗. 同时无人机在探索环境的过程中收集到的大量样本数据, 也能满足 LSTM 需要大量训练数据的特点, 因此可利用 LSTM 对 TD3 算法的网络结构进行改进, 从而解决上述提到的时间序列问题.

4.2 基于 LSTM 的改进的 TD3 算法

(1) 改进的经验池样本

算法对经验池中的样本数据进行了改进, 新增了历史数据, 改进的样本数据定义为 $(h_t^l, s_t, a_t, r_t, s_{t+1}, f_t)$, 其中, f_t 代表在执行动作 a_t 后是否到达终止状态, h_t^l为 t 时隙时, 长度为 l 的历史数据:

$$h_t^l=\left\{\begin{matrix}s_{t-l},a_{t-l},\cdots,s_{t-1},a_{t-1}, & \text{if } l,t\geqslant 1\\ s^0,a^0, & \text{otherwise}\end{matrix}\right. \tag{19}$$

其中, s^0, a^0 为零值虚拟向量, 其维度与正常的状态向量和动作向量保持一致. 当历史数据的长度 $l\geqslant 1$ 时, 历史数据h_t^l为 l 组 (s_t, a_t) 组成的数据对集合, 否则历史数据 h_t^l为零值虚拟向量.

(2) 改进的 LSTM-TD3 网络框架

本文提出的 LSTM-TD3 算法引入了 LSTM 网络对 Actor 和 Critic 网络框架进行了改进, 提出的记忆提取层可以对历史数据进行处理, 从而提取对 Actor 和 Critic 网络有益的历史特征信息.

(a) Critic 网络框架

Critic 网络可以看作是由记忆提取 Q^{me} 单元, 当前特征提取 Q^{cf} 单元和感知整合 Q^{pi} 单元所构成的复合函数, 结构如图 3 所示.

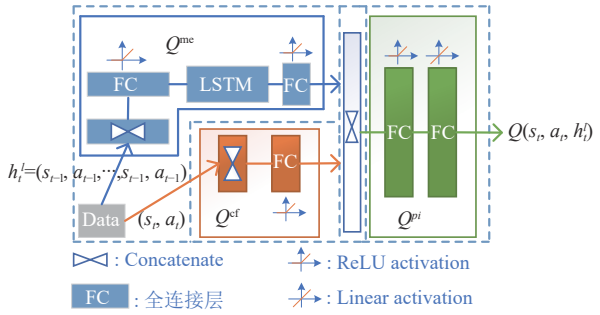

图 3 Critic 网络框架

$$Q(s_t,a_t,h_t^l)=Q^{\mathrm{me}}\cdot Q^{\mathrm{cf}}\cdot Q^{\mathrm{pi}}$$
$$=Q^{\mathrm{pi}}(Q^{\mathrm{me}}(h_t^l)\odot Q^{\mathrm{cf}}(s_t,a_t)) \quad (20)$$

其中, ⊙代表链接操作, Q^{me} 是记忆提取网络, 其以数据h_t^l为输入, 利用 LSTM 网络层提取历史数据中的特征信息.

Q^{cf} 是基于当前观测值的当前特征提取网络, 以当前状态 s_t 和动作 a_t 为输入, 提取当前状态和动作的特征信息.

(b) Actor 网络框架

Actor 网络同样也可以看作是由记忆提取 π^{me} 单元, 当前特征提 π^{cf} 单元和感知整合 π^{pi} 单元所构成的复合函数:

$$\pi(o_t,h_t^l)=\pi^{\mathrm{me}}\cdot\pi^{\mathrm{cf}}\cdot\pi^{\mathrm{pi}}=\pi^{\mathrm{pi}}(\pi^{\mathrm{me}}(h_t^l)\odot\pi^{\mathrm{cf}}(o_t)) \quad (21)$$

其中, π^{me} 是基于历史数据h_t^l的记忆提取网络, 其利用 LSTM 对输入的历史数据进行特征提取, 网络结构与 Critic 的 Q^{me} 单元相似.

π^{cf} 是基于当前观测值的当前特征提取网络, 以当前状态 s_t 和动作 a_t 为输入, 提取当前状态和动作的特征信息.

网络框架如图 4 所示, 对于 Actor 网络, 在进行数据处理时, 记忆提取单元 π^{me} 利用 LSTM 对历史数据h_t^l进行处理, 提取特征信息 $M_{\pi\mathrm{t}}$, 当前特征提取单元 π^{cf} 对当前状态 s_t 和动作 a_t 进行处理, 提取特征信息 $C_{\pi\mathrm{t}}$, 两个特征提取单元得到的特征信息 $M_{\pi\mathrm{t}}$ 和 $C_{\pi\mathrm{t}}$ 相链接作为感知整合单元 π^{pi} 的输入, 得到最终动作 $\pi(o_t, h_t^l)$; 对于 Critic 网络, 记忆提取单元 Q^{me} 利用 LSTM 对历史数据h_t^l进行处理, 提取特征信息 M_{Qt}, 当前特征提取单 Q^{cf} 对当前状态 s_t 和 Actor 网络得到的动作 $\pi(o_t, h_t^l)$ 进行处理, 提取特征信息 C_{Qt}, 两个特征提取单元得到的特征信息 M_{Qt} 和 C_{Qt} 相链接作为感知整合单元 Q^{pi} 的输入, 得到最终的 Q 值 $Q(s_t, \pi(o_t, h_t^l), h_t^l)$.

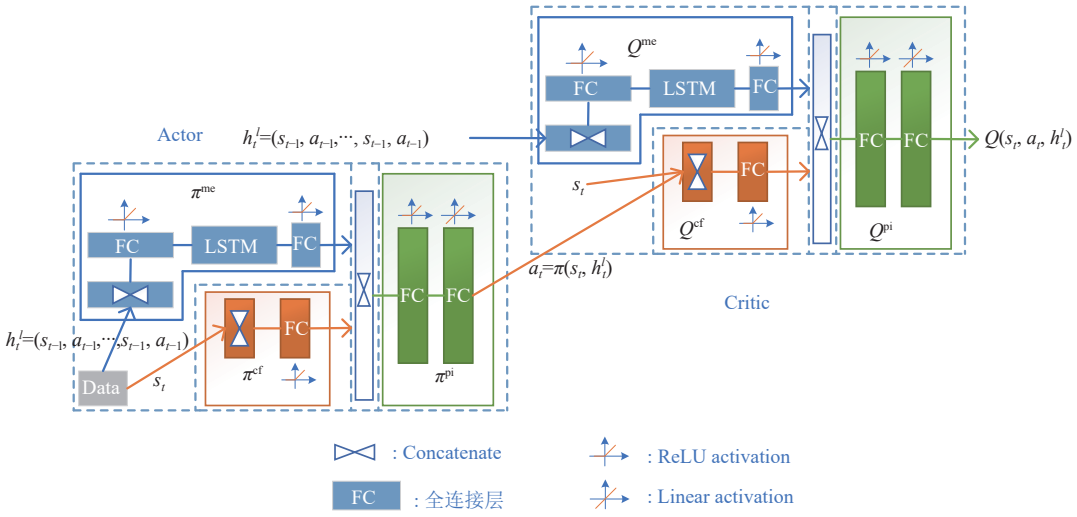

图 4 LSTM-TD3 算法的网络框架

(3) LSTM-TD3 算法框架

本文采用 TD3 算法对 Actor 和 Critic 网络框架进行训练优化, 算法包含一套 Actor 网络, 包含当前 Actor 网络 π 和目标 Actor 网络 π'; 两套 Critic 网络, 每套 Critic 网络包含一个当前 Critic 网络 Q_c 和目标 Critic 网络Q_c'.

(a) Critic 网络优化

对于 Critic 网络, 每套 Critic 网络 $Q_c\in\{1, 2\}$通过最小化当前网络 Q_c 和目标 Q 值 Q_{ta} 的均方误差来对网络参数进行更新优化:

$$\min_{\theta Qc} E_{\{(h_t^l,s_t,a_t,s_{t+1},f_t)_i\}_{i=1}^N}(Q_c-Q_{\text{ta}})^2 \tag{22}$$

目标 Q 值 Q_{ta} 由目标 Actor 网络 π'和目标 Critic 网络Q_c'共同共同决定, 且在双 Critic 网络中, 选取 Q 值较小的 Critic 网络进行目标 Q 值的计算, 有助于避免 Q 函数中的过高估计问题, 具体公式如下:

$$Q_{\text{ta}}=r_t+\gamma\times(1-f_t)\times\min_{c=1,2}Q_c'(s_{t+1},a',h_{t+1}^l) \tag{23}$$

其中, a'为 Actor 网络π_c'在状态 s_{t+1} 和历史数据h_{t+1}^l 下得到的目标动作:

$$a'=\pi'(s_{t+1},h_{t+1}^l)+\varepsilon,\varepsilon\sim clip(N(0,\sigma),-c,c) \tag{24}$$

其中, ε 为动作噪声, h_{t+1}^l是状态 s_{t+1} 之前的历史数据, 为 l 组 (s_t, a_t) 所组成的数据集合, 其定义为:

$$h_{t+1}^l=(h_t^l-(s_{t-1},a_{t-1}))\cup(s_t,a_t) \tag{25}$$

(b) Actor 网络优化

Actor 网络以最大化当前 Critic 网络的预测 Q 值来对目标对策略参数进行优化, 其中网络 Q 可以是两套 Critic 当前网络中的任意一个:

$$\max_{\theta\pi} E_{\{(h_t^l,s_t)_i\}_{i=1}^N}Q(s_t,\pi(s_t,h_t^l),h_t^l) \tag{26}$$

本文利用 LSTM 网络来改进 TD3 算法的 Actor 和 Critic 神经网络, LSTM 可以更好地处理连续时间场景下的时间序列特性数据, 从而使 Agent 能够更好地进行训练和决策.

同时, LSTM-TD3 算法采取了软更新的策略对目标网络参数进行更新, 更新方式如下:

$$\theta^{Q'}\leftarrow\theta^{Q}+(1-\tau)\theta^{Q'} \tag{27}$$

$$\theta^{\pi'}\leftarrow\theta^{\pi}+(1-\tau)\theta^{\pi'} \tag{28}$$

具体流程如算法 1.

算法 1. LSTM-TD3 算法

输入: 迭代数 K, 样本长度 N, 学习率 α_c 和 α_a, 折扣因子 γ.
输出: 无人机飞行角度 $\beta(i)$, 任务卸载率 $R_k(i)$, 服务的 UEk.

1) 初始化: Critic 网络参数θ^{Q1} 和θ^{Q1}, Actor 网络参数θ^{π}, 目标网络参数$\theta^{Q'1}\leftarrow\theta^{Q1}$, $\theta^{Q'2}\leftarrow\theta^{Q2}$ 和$\theta^{\pi'}\leftarrow\theta^{\pi}$, 环境状态 s_1=*env.reset*(), 初始化历史数据$h_1^l\leftarrow0$ 和经验池 D
2) **for** t=1 **to** T **do**
/* 探索阶段 */
3) 当前 Actor 通过当前状态s_t选取动作 $a_t=\pi_\theta^{\pi}(s_t, h_t^l+\varepsilon, \varepsilon\sim N(0, \sigma))$
4) 根据动作a_t, 通过通信模型、移动模型和计算模型, 得到新的状态, 由式 (15) 获得奖励值 R_t
5) s_{t+1}, R_t, d_t=*env.step*(a_t)
6) (s_t, a_t, R_t, s_{t+1}, h_t^l, d_t) 存入经验池 D
7) **if** d **then**
8) 重置环境状态 s_{t+1}=*env.reset*()
9) 重置历史数据$h_t^l+1\leftarrow0$
10) **else**
/* 更新历史数据 */
11) $h_{t+1}^l=(h_t^l(s_{t-1}, a_{t-1}))\cup(s_t, a_t)$
12) **end**
/* 训练阶段 */
13) 从经验池中随机抽取 N 条训练样本 $\{(s_t,a_t,R_t,s_{t+1},h_t^l,d_t)_i\}_{i=1}^N$ from D
14) 由式 (22) 优化 Critic 网络参数
15) 由式 (26) 优化 Acotor 网络参数
16) 由式 (27) 式 (28) 更新目标网络参数

5 仿真实验和结果分析

5.1 仿真实验说明

根据本文构建的 MEC 仿真环境对无人机辅助 MEC 计算卸载进行仿真, 仿真环境为 Python 3.7 和 TensorFlow 2.0, 实验场景基于 gym 仿真平台实现, 利用 gym搭建仿真的无人机环境, 模拟无人机辅助智能机器人进行任务卸载的应用.

(1) 参数设置

在无人机辅助的 MEC 系统中, 考虑一个 2D 正方形区域, 大小为 $W\times W$=100×100 m^2, 无人机的飞行高度固定在 h=100 m. 无人机的水平最大覆盖半径为 5 m. 无人机的总质量定义为 M_{UAV}=9.65 kg. 整个时间段定义为 T=400 s, 且被均匀划分为 I=40 个时隙. 无人机的最大时速定义为 $V_{\max}$=13 m/s, 其中在每时隙无人机的飞行时间为 t_{fly}=1 s. 传输带宽被设置为 B=5 MHz. 噪声功率设置为 σ^2=−100 dB. 我们设定用户的传输功率为 P_{up}=0.1 W, 无人机的电池容量为 b=500 KJ, 无人机处理每单位字节周期 s=[500, 1000] cycles/bit. 具体参数表如表 1.

对于 UAV 携带的边缘服务器的参数按照 Jetson Xavier NX 进行设置, 具体参数如表 2.

(2) 仿真实验设计

本文仿真实验设计的无人机计算卸载过程分为两个阶段: 探索阶段和训练阶段.

表 1 仿真实验数据参数

参数	说明	数值
M	UE数量	4
M_{UAV}	UAV质量	9.65 kg
I	时隙数	40
B_u	带宽	5 MHz
P_{NLOS}	传输损耗	−100 dB
E_b	UAV总能量	500 KJ
f_{ue}	UE算力	0.6 GHz
f_{uav}	UAV算力	1.6 GHz
T	运行周期	400 s
$v_{\max}$	UAV最大飞行速度	13 m/s
α_0	信道增益	−50 dB
σ^2	接收噪声	−100 dB
P_{up}	UE传输功率	0.1 W
s	CPU转数	[500, 1000] cycle/bit

表 2 移动边缘服务器参数

参数	数值
型号	Jetson Xavier NX
重量	0.3 kg
CPU	NVIDIA ARMv8.2 64 bit @1.6 GHz
显存	8 GB DDR4, NVIDIA Volta GPU @ 1100 MHz
存储容量	128 GB

(a) 探索阶段

1) 每个时隙, 地面终端用户会根据移动模型, 由式 (3)、式 (4), 在一定的范围内进行移动, 并随机产生计算密集型任务.

2) 每个时隙, 无人机都会根据地面终端用户的状态与任务量、自身的状态和约束、仿真环境的环境参数与 gym 仿真的环境 env 进行交互, 得到当前的动作空间和下一时隙的状态, 作为训练样本存入经验池:

(b) 训练阶段

当经验池达到最大存储容量后, 无人机根据经验池中的样本数据, 对智能体进行训练, 更新网络参数.

5.2 对比算法分析及评价指标

(1) 对比算法

1) DQN: DQN 算法是一种将 Q_learning 通过神经网络近似值函数的一种方法[15].

2) TD3: 基础的 TD3 算法[16].

3) DDPG 算法: DDPG 算法是一种无模型 (model-free) 的强化学习算法, 其可以在连续动作空间中学习策略. 该算法由策略函数和 Q 值函数组成. 策略函数作为 Actor 产生动作. Q 值函数作为 Critic 去评估 Actor 的表现并且指导其后续的动作[17].

4) Actor-Critic 算法: Actor-Critic 算法是一种强化学习中的策略梯度 (policy gradient) 算法. Actor 主要负责制定动作策略, 它根据当前状态选择一个动作, 并接收环境的奖励信号. 而 Critic 则负责评估该行为策略的好坏, 它基于环境状态和实际奖励信号对该行为策略进行评估, 并给出一个状态值函数或者优势函数的估计结果, 用于指导 Actor 的更新[18,19].

(2) 评价指标

不同条件下的系统总时延大小:

$$\tau_{\mathrm{delay}}(i)=\sum_{k=1}^{K}\alpha_k(i)\max\left\{\begin{matrix}t_{\mathrm{local},k}(i),\\ t_{\mathrm{UAV},k}(i)+t_{\mathrm{tr},k}(i)\end{matrix}\right\} \tag{29}$$

5.3 仿真结果分析

图 5 对 AC 网络的学习率进行了比较分析, 从图 5 可以看出, 在合理的学习率下, 随着训练回合数的增加, 系统总奖励会越来越大, 并最终收敛. 这是因为在训练过程中, Actor 和 Critic 网络会调整自己的网络参数, 以获得最优策略. 随着训练步数增加, LSTM-TD3 算法在不同学习率下的系统的总时延逐渐降低, 当学习率为 0.1 时, 系统总时延反而难于收敛, 相反, 当学习率过低时 (取值为 0.000 1), 算法收敛速度较慢, 大约在训练 400 步后达到收敛. 因此结合这一实际情况, 本文在剩余的仿真实验中将 Actor 和 Critic 学习率设置为 0.001.

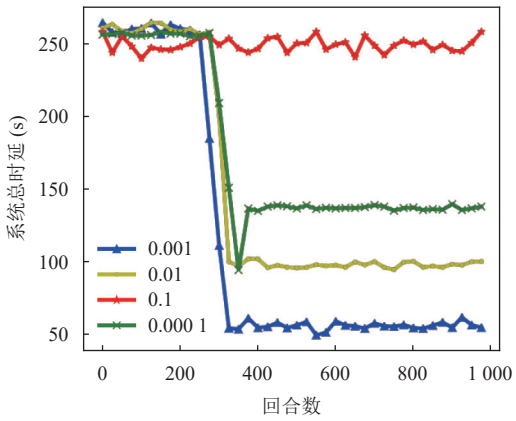

图 5 不同学习率下的 LSTM-TD3 算法性能

图 6 对算法的折扣因子进行了比较分析, 折扣因子同样会对算法的训练效果产生较大影响, 折扣因子反映了移动边缘服务器在进行决策时对未来情况的考量, 当折扣因子为 0 时, 代表智能体只关注当前决策所得到的及时奖励, 折扣因子越大, 代表智能体越注重长远的考虑, 未来奖励的影响越大, 由图 6 可知, 算法在折扣因子为 0.9 时, 性能最好, 因此本文在剩余的仿真实验中将算法的折扣因子设为 0.9.

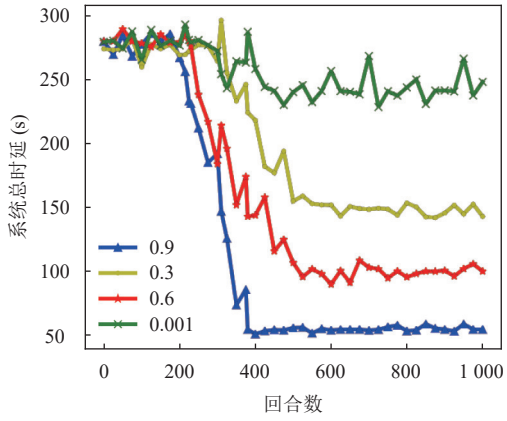

图 6　不同折扣因子下的 LSTM-TD3 算法性能

图 7 对比分析了不同长度的历史数据对算法性能产生的影响, 分别将历史数据的长度设置为 0, 1, 5 和 10, 算法中的历史数据用于对算法网络中的记忆提取单元进行训练, 同时也会对当前决策产生影响, 历史数据越长, 代表网络能从历史决策中提取并保留更多信息, 但同时也会增加网络的训练成本和决策的执行成本, 由对比分析可得到, 历史数据的长度设置为 5 时, 算法的性能最好.

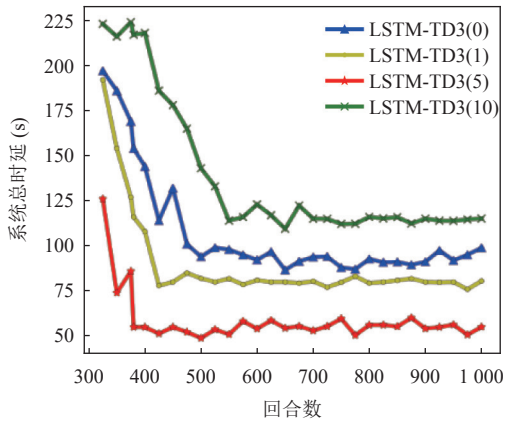

图 7　不同长度的历史数据下 LSTM-TD3 算法性能

为验证 LSTM-TD3 算法的性能, 本文将设计的 LSTM-TD3 算法与另外 4 种算法 (AC、DQN、DDPG、TD3) 在相同的实验场景下进行对比分析. 图 8 表明在计算任务量 50 MB 的条件下, 系统总时延在不同算法下, 随着迭代次数增加的表现情况. 由图可知, Actor-Critic 算法难以收敛, 是因为 Actor-Critic 算法网络结构中的两套网络是同步更新的, 因此会出现难以收敛的情况, DQN 有着同样的情况. 另外 3 种算法中, 由于 DDPG, TD3 和 LSTM-TD3 具有双网络结构, 因此能有效规避这个问题. 从结果看, LSTM-TD3 的收敛性更好, 该算法下系统的时延性能更好.

本文将计算任务卸载时延作为算法的评价指标, 对包括本文算法在内的 5 种算法性能进行了对比分析. 图 9、图 10 为不同任务量和终端数的条件下各算法的总计算时延. 由图可知, 其他条件相同的情况下, LSTM-TD3 算法的时延性能优于其他 3 种算法. AC 算法和 DQN 算法由于算法输出动作的取值范围差异较大, 因此随着终端数和任务大小的增加, 波动较大, DDPG, TD3 和 LSTM-TD3 算法波动较小, 且逐渐趋于平稳, 由图可知, 本文提出的 LSTM-TD3 算法的性能最好.

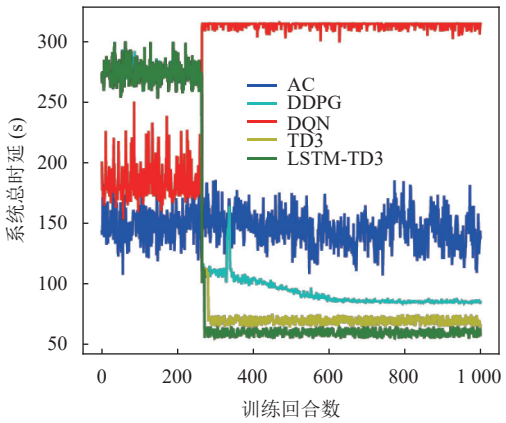

图 8　不同算法优化系统总时延的性能对比

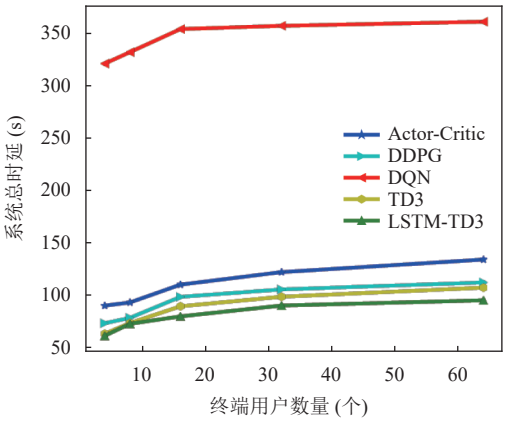

图 9　不同终端数量下各算法总时延

实验同时研究了终端用户的算力对计算卸载时延的影响, 图 11 测试了不同终端算力对算法时延性能的影响. 可以发现, 对于本文提出的 LSTM-TD3 算法, 终端算力越高, 其对应的任务处理时延越小. 另一方面, 从图 12 中能看出, 在不同终端算力条件下, LSTM-TD3 的性能最好.

通过上述各个实验的仿真对比分析, 可知本文提出的算法相较于其他 4 种常见的 DRL 算法, 具有更好

的收敛和鲁棒性. 因此, 本文提出的 LSTM-TD3 算法在对系统的计算卸载策略进行优化后, 相较于其他算法会有更小的最大总处理时延, 性能更好.

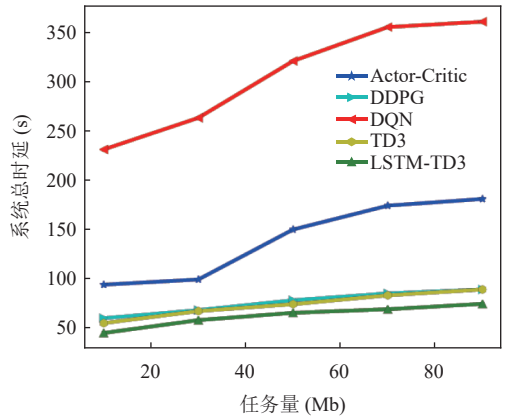

图 10 不同任务量下各算法总时延

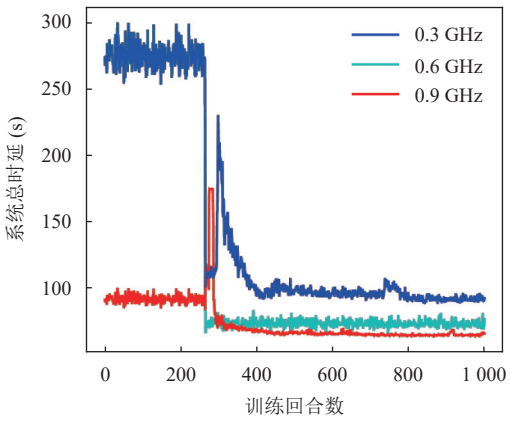

图 11 不同终端算力下 LSTM-TD3 的总时延

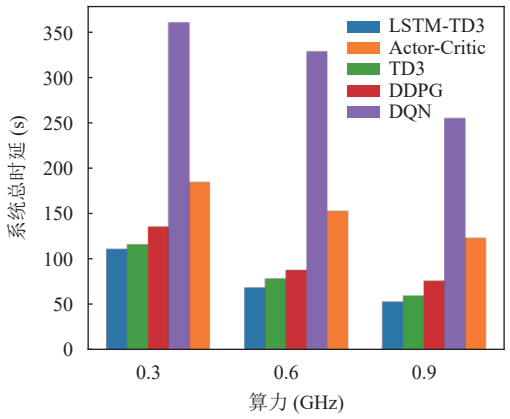

图 12 不同终端算力下各算法的总时延

6 结语

本文研究了面向多地面终端用户的单无人机计算任务卸载问题. 提出了基于长短期记忆网络的双延时确定策略梯度算法 (LSTM-TD3), 利用 LSTM 对 Actor-Critic 网络结构进行了优化, 使智能体能够从历史数据提取特征信息, 能够好的处理时间序列数据. 对本文算法进行了对比仿真时延以验证算法的可行性. 对多个评价指标进行了实验, 与另外 4 种基线算法进行比较分析, 验证了 LSTM-TD3 算法在系统时延方面的优越性. 未来的研究将着重于研究多 UAV 等复杂环境下的移动边缘计算任务卸载问题.

参考文献

1 Wang XJ, Li JM, Ning ZL, *et al*. Wireless powered mobile edge computing networks: A survey. ACM Computing Surveys, 2023, 55(13s): 263.

2 Ning ZL, Hu H, Wang XJ, *et al*. Mobile edge computing and machine learning in the internet of unmanned aerial vehicles: A survey. ACM Computing Surveys, 2023, 56(1): 13.

3 Wang Y, Ru ZY, Wang KZ, *et al*. Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing. IEEE Transactions on Cybernetics, 2020, 50(9): 3984–3997. [doi: 10.1109/TCYB.2019.2935466]

4 Liu BH, Liu CX, Peng MG. Computation offloading and resource allocation in unmanned aerial vehicle networks. IEEE Transactions on Vehicular Technology, 2023, 72(4): 4981–4995. [doi: 10.1109/TVT.2022.3222907]

5 Glanois C, Weng P, Zimmer M, *et al*. A survey on interpretable reinforcement learning. Machine Learning, 2024, 113(8): 5847–5890. [doi: 10.1007/s10994-024-06543-w]

6 Li CL, Jiang K, Zhang Y, *et al*. Deep reinforcement learning-based mining task offloading scheme for intelligent connected vehicles in UAV-aided MEC. ACM Transactions on Design Automation of Electronic Systems, 2024, 29(3): 54.

7 Zhang LX, Tan RT, Zhang YF, *et al*. UAV-assisted dependency-aware computation offloading in device-edge-cloud collaborative computing based on improved Actor-Critic DRL. Journal of Systems Architecture, 2024, 154: 103215. [doi: 10.1016/j.sysarc.2024.103215]

8 Gao A, Zhang S, Zhang Q, *et al*. Task offloading and energy optimization in hybrid UAV-assisted mobile edge computing systems. IEEE Transactions on Vehicular Technology, 2024, 73(8): 12052–12066. [doi: 10.1109/TVT.2024.3380003]

9 Shi ML, Zhang XQ, Chen J, *et al*. UAV cluster-assisted task offloading for emergent disaster scenarios. Applied Sciences, 2023, 13(8): 4724. [doi: 10.3390/app13084724]

10 Liu X, Chai ZY, Li YL, *et al*. Multi-objective deep reinforcement learning for computation offloading in UAV-assisted multi-access edge computing. Information Sciences, 2023, 642: 119154. [doi: 10.1016/j.ins.2023.119154]

11 Zhou TQ, Xu M, Qin D, *et al*. Computing offloading based on TD3 algorithm in cache-assisted vehicular NOMA-MEC networks. Sensors, 2023, 23(22): 9064 [doi: 10.3390/s23229064]

12 张茜, 苏冬冬, 张聪, 等. 面向 MEC 多智能体协同任务卸载的深度强化学习算法. 电讯技术, 1–9. [doi: 10.20079/j.issn.1001-893x.240107001]

13 Jain V, Kumar B. QoS-aware task offloading in fog environment using multi-agent deep reinforcement learning. Journal of Network and Systems Management, 2023, 31(1): 7. [doi: 10.1007/s10922-022-09696-y]

14 Seid AM, Boateng GO, Mareri B, *et al*. Multi-agent DRL for task offloading and resource allocation in multi-UAV enabled IoT edge network. IEEE Transactions on Network and Service Management, 2021, 18(4): 4531–4547. [doi: 10.1109/TNSM.2021.3096673]

15 Wang YP, Fang WW, Ding Y, *et al*. Computation offloading optimization for UAV-assisted mobile edge computing: A deep deterministic policy gradient approach. Wireless Networks, 2021, 27(4): 2991–3006. [doi: 10.1007/s11276-021-02632-z]

16 Yan JJ, Zhao XH, Li Z. Deep-reinforcement-learning-based computation offloading in UAV-assisted vehicular edge computing networks. IEEE Internet of Things Journal, 2024, 11(11): 19882–19897. [doi: 10.1109/JIOT.2024.3370553]

17 Yan M, Xiong R, Wang Y, *et al*. Edge computing task offloading optimization for a UAV-assisted Internet of vehicles via deep reinforcement learning . IEEE Transactions on Vehicular Technology, 2023, 73(4): 5647–5658. [doi: 10.1109/TVT.2023.3331363]

18 Zhou TQ, Xu M, Qin D, *et al*. Computing offloading based on TD3 algorithm in cache-assisted vehicular NOMA-MEC networks. Sensors, 2023, In: Zhang SX, Liao HJ, Zhou ZY, *eds*. Federated deep Actor-Critic-based task offloading in air-ground electricity IoT. Proceedings of the 2021 IEEE Global Communications Conference. Madrid: IEEE, 2021. 1–6.

19 Yang J, Yuan QF, Chen SW, *et al*. Cooperative task offloading for mobile edge computing based on multi-agent deep reinforcement learning. IEEE Transactions on Network and Service Management, 2023, 20(30): 3205–3219.

(校对责编: 王欣欣)