

基于深度强化学习的改进移动边缘计算任务卸载算法研究

蒋守花, 舒 晖

(成都医学院 现代教育技术中心, 四川 成都 610500)

摘要: 大数据时代下移动终端用户规模不断扩大, 万物互联在给人们带来极大便利的同时, 也存在大量数据地理位置分散的问题, 给用户服务质量QoS带来了极大挑战。首先, 搭建一个基于移动边缘计算平台三层服务架构的任务卸载模型。其次, 结合MEC平台实际应用场景, 利用同策略经验回放和熵正则改进深度强化学习算法, 优化了MEC平台的任务卸载策略, 并设计了实验对3种传统算法和改进算法的能耗、时延、网络使用量进行对比分析。实验结果表明, 改进算法在降低能耗、时延和网络使用量方面具有更优越的性能。

关键词: 深度强化学习; 边缘计算任务卸载; 同策略经验回放; 熵正则

DOI: 10.11907/rjdk.232294

开放科学(资源服务)标识码(OSID):

中图分类号: TP393

文献标识码: A

文章编号: 1672-7800(2024)009-0150-07



Research on Improved Moving Edge Computing Task Unloading Algorithm Based on Deep Reinforcement Learning

JIANG Shouhua, SHU Hui

(Modern Education Technology Center, Chengdu Medical College, Chengdu 610500, China)

Abstract: In the era of big data, the scale of mobile terminal users continues to expand, and the Internet of everything brings great convenience to people. At the same time, there is also the problem of geographic dispersion of a large amount of data, which brings great challenges to the QoS of user service. In this paper, a task unloading model based on the three-layer service architecture of the mobile edge computing platform is first built. Combined with the actual application scenario of the MEC platform, the deep reinforcement learning algorithm is improved by using the same policy experience playback and entropy regularization, and the task unloading strategy of the MEC platform is optimized. Experiments are designed to compare and analyze the three indexes of energy consumption, delay and network usage of the three traditional algorithms and the improved algorithm, and verify that the improved algorithm has better performance in reducing energy consumption, delay and network usage.

Key Words: deep reinforcement learning; edge computing task offloading; same strategy experience replay; entropy regularity

0 引言

近年来,随着大数据、云计算、雾计算等新兴技术的不断发展,为智能技术的发展带来了契机,同时给计算模型提出了更高要求。大数据时代下移动终端用户规模不断

扩大,万物互联在给人们带来极大便利的同时,也存在大量数据地理位置分散的问题,给用户服务质量(Quality of Service, QoS)带来了极大挑战^[1]。虽然云计算为大数据处理提供了集中、高效的平台,但是网络带宽增加的速度远不及数据增长速度,而提高网络带宽又会增加更多硬件成本。同时,复杂的网络环境造成了对数据处理的延迟。针

收稿日期: 2023-12-21

扫描二维码阅读全文:



基金项目: 四川省高等学校人文社会科学重点研究基地·四川省教育信息化应用与发展研究中心项目(JYXX23-002); 成都医学院科研基金项目(CYSYB23-02)

作者简介: 蒋守花(1988-), 女, 硕士, CCF会员, 成都医学院现代教育技术中心工程师, 研究方向为信息系统建设与维护、智慧校园、边缘计算、人工智能、物联网; 舒晖(1972-), 女, 成都医学院现代教育技术中心高级工程师, 研究方向为现代教育技术。本文通讯作者: 舒晖。

对云计算模型亟需解决的网络延迟和处理数据实时性较差这两大瓶颈,研究者提出了基于移动边缘计算(Mobile Edge Computing, MEC)的物联网解决方案,通过将网络、存储、计算等资源融为一体的开放平台为移动用户提供服务^[2]。简单来说就是将网络、存储、计算等资源放在边缘侧为用户提供服务,由于服务是本地的,无须将请求发送至云端中心进行处理,既避免了数据在传输过程中因网络震荡、带宽限制等因素造成的网络延迟,又降低了传输过程中带来的数据泄露和被篡改的风险,提高了数据处理的实时性和用户的QoS^[3]。

虽然MEC在一定程度上提升了移动设备计算效率,也缓解了云计算中心的处理压力,但是边缘服务器单位时间内的承载能力和对数据的处理精度是有限的,当用户对数据处理精度要求很高时,边缘服务器往往不能达到用户对服务高效性、高准确性的需求。因此,设计一个将移动任务卸载到云服务器执行的方案是很有必要的。常见的任务卸载方案大致分为两类:一类是粗粒度任务卸载,另一类是细粒度任务卸载。粗粒度任务卸载顾名思义就是不对移动终端任务卸载进行分类,而是全部卸载到云服务器进行执行,其往往没有考虑到MEC对任务执行高效性和实时性的需求,同时对部署在边缘设备侧的边缘服务器资源利用率极低,造成了资源的严重浪费。细粒度任务卸载是指对移动终端的任务根据其所需算力、存储空间及能源消耗等特性进行分类,将这些分类作为依据把任务划分为内部具有联系的多个小任务,小任务可以卸载到边缘服务器进行执行,从而降低了任务处理的计算复杂度和本地设备的能源消耗,同时提高了边缘服务器的资源利用率^[4]。因此,本文设计一个基于细粒度任务卸载的算法方案,利用马尔可夫决策(Markov Decision Processes, MDP)方法架构基于MEC平台的任务卸载模型,判断哪些任务需要卸载,卸载部分还是全部,卸载到一个边缘服务器还是多个边缘服务器等问题,最终达到降低智能终端和各边缘节点的能耗、时延及网络使用量等资源的目的是,实现对MEC平台任务卸载策略的优化。

本文的主要贡献有以下3个方面:

(1)搭建一个基于移动边缘计算平台三层服务架构的任务卸载模型。

(2)结合MEC平台实际应用场景,利用同策略经验回放(Replay the same Strategy Experience, RSSE)和熵正则(Entropy Regularity)改进传统的深度强化学习算法DQN(Deep Q-network),设计了算法流程并加以实现。

(3)设计实验对RSSE-DQN算法和其他3种算法在能耗、时延、网络使用量3个方面进行对比分析,证明了RSSE-DQN算法具有较优越的性能。

1 相关工作

随着5G技术的快速发展,基于移动边缘计算的任务卸载算法获得越来越多的研究者关注。文献[5]构建基于存储空间和计算资源限制的马尔可夫决策方法,提出基于改进深度强化学习的边缘计算服务卸载算法,并运用动态规划的思想提出了动作筛选算法,剔除了非法动作,优化了智能体动作空间大小,提升了算法训练效率。文献[6]提出基于长短期记忆网络和事后经验回放的改进的深度强化学习算法,相比传统的深度强化学习算法,改进算法在能耗、费用、负载均衡和延迟上都有很大程度的改善。文献[7]针对多用户多任务的移动边缘计算(MEC)系统,通过对多用户、多任务的MEC系统任务卸载调度和资源分配等进行综合建模,提出基于深度强化学习的多用户多任务的任务卸载调度和资源分配算法,利用贪心策略算法解决了任务卸载决策和卸载调度问题,改善了服务器频率分配问题。文献[8]针对在边缘计算环境中多目标任务调度策略优化效果差的问题,提出基于改进的竞争深度双Q网络的多目标任务调度算法,用深度双Q网络对目标中的最大动作进行分解,消除过高估计,采用即时奖励的经验样本方法,按照重要性程度高低对经验样本进行分类,通过算法筛选出重要性程度高的经验样本,提高算法对样本的训练速度和收敛效率。

从上述研究不难看出,基于深度学习和强化学习的任务卸载算法利用神经网络和策略学习方法,使算法具有自主学习和自主感知等特性,能够解决网络环境更复杂、维度更高的问题。同时,为提高算法在任务卸载具体应用场景中的自适应性,本文提出基于同策略经验回放和熵正则的改进深度Q网络算法(Improved DQN Algorithm Based on Same Strategy Empirical Playback and Entropy Regularization, RSS2E-DQN)。采用同策略经验回放对经验样本的价值网络进行训练,进而指导策略网络的更新,提高了算法收敛速度,降低了任务卸载时延;采用熵正则指导动态规划下的动作筛选,熵越小,优良动作越多、越集中,方便挑选出优良动作集合,优化动作空间大小,提高算法训练效率,进而优化边缘计算任务卸载策略。

2 相关概念

2.1 深度学习概念

深度学习是机器学习算法的分支之一,深度学习也称为深度结构化学学习,其深度表征的是神经网络层数,层数越多表示神经网络越复杂。传统的神经网络分为深度信念网络、卷积神经网络和循环神经网络(Recurrent Neural Networks, RNN)等,其工作模式是在连续状态下处理完每个任务的元素后,丢弃原有的网络隐藏状态^[9]。如果任务

内部存在依赖关系,上层网络中任务的某些状态丢失会影响结果的准确性。深度学习是一种现代变体,是由无限数量的有限层组成的,能够提高对网络中经验样本的训练效率。基于神经网络的深度学习模型已在许多应用场景中得到推广,例如自然语言处理、图像字幕生成、语音合成等领域。

2.2 强化学习概念

强化学习研究的数学基础和建模工具是马尔可夫决策过程(Markov Decision Processes, MDP),一个完整的MDP包含状态、动作、状态转移函数、奖励函数、折扣因子等元素^[10]。深度强化学习的主体是智能体(agent),与智能体进行交互的是环境(environment),智能体负责做出动作或决策,环境则为智能体提供做出动作或决策所依托的规则。状态(state)是对任意时刻环境的概括,其是智能体作出决策的依据。

强化学习的目标就是得到一个策略函数,其可以是确定的,也可以是随机的。策略中回报最大的称为最优策略。智能体根据策略函数对应的状态作出决策,决策依赖于环境当前的状态,与以前的状态无关。

智能体与环境的交互过程大致可描述为:智能体观测到环境的状态 S ,作出决策 a ;环境在接收到决策 a 后,会根据决策中的指令作出相应变化;变化后环境会将新状态 S' 和得到的奖励反馈给智能体。这样往复循环,奖励积累逐渐增多,对强化学习模型中策略的训练效率越来越高,实现的推理任务也越来越精确。图1是智能体与环境交互的示意图。

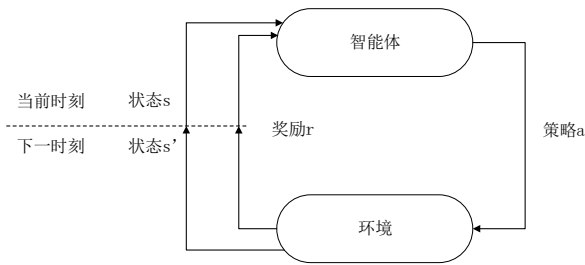


Fig. 1 Diagram of interaction between agent and environment

图1 智能体与环境交互示意图

2.3 深度强化学习概念

深度强化学习是将深度学习与强化学习结合后形成的一个机器学习的分支领域。强化学习中通过智能体和环境的交互获得奖励的累积,在该过程中实现策略的优化。但在此过程中,智能体的处理状态通常是非连续性的,在与深度学习结合后能弥补这一缺点。智能体能在相同条件下处理非结构化的连续性任务,使获得的奖励之间具有依赖性,提高了智能体的自我学习能力、自主感知能力和处理连续性任务的效率,并通过价值函数提升了策略的准确性。

常见的深度强化学习算法有深度Q网络算法(Deep Q-network, DQN)^[11]、深度确定性策略梯度算法(Deep De-

terministic Policy Gradient, DDPG)和SAC(Soft Actor Critic)算法。

3 移动边缘计算任务卸载模型

MEC平台主要由云计算中心层、边缘服务器层和移动终端设备层3部分组成,云计算中心层部署有大规模且性能良好的实体服务器,负责大规模的运算和推理;边缘服务器层部署有性能适中的边缘服务器,负责处理对实时性要求较高的任务;移动终端设备层主要由手机、智能手表及各类传感器等具有一定处理性能的智能终端设备组成。当某个来自移动终端设备的任务需要卸载时,切分算法会将该任务切分成许多小任务。这些小任务中的接口型任务须在本地执行,数据处理型任务因计算量较大,若设备本身处理不了,则会卸载到边缘服务器或云服务器进行处理。这些小任务内部之间可进行数据交互,且存在依赖关系。其可以相互结合执行,也可以独立执行,这是基于细粒度的任务卸载算法得以实现的前提,也是任务依赖模型得以成立的基础,如图2所示。

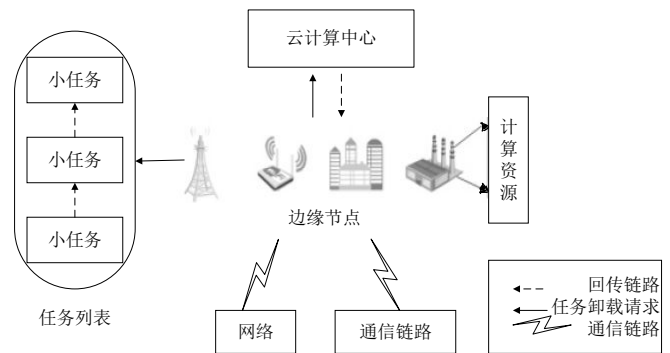


Fig. 2 MEC task unloading model

图2 MEC任务卸载模型

4 问题建模

本文将整个MEC任务卸载模型中的计算设备用 (DC, ES, MT) 表示,其中 DC 表示由一个物理服务器集群组成的拥有强大算力的云计算中心, ES 表示 m 个边缘服务器设备集 $\{es_1, es_2, \dots, es_m\}$, MT 表示 n 个移动终端设备集 $\{mt_1, mt_2, \dots, mt_n\}$,并假设每个终端设备在同一时刻只有一个应用发出请求。将终端发出的应用请求切分后的小任务用一个五元组 $\{N_{ij}^{up}, C_{ij}^{in}, M_{ij}, N_{ij}^{dw}, C_{ij}^{out}\}$ 表示,其中:① N_{ij}^{up} 表示第 i 个移动终端的第 j 个小任务接收数据匹配的上行带宽;② C_{ij}^{in} 表示同一任务输入的数据量,例如录入参数或输入文字;③ M_{ij} 表示部署该小任务所需的计算资源。大量研究表明,主机的功耗与CPU利用率息息相关。因此,作为本文的优化目标之一,采用该值表示部署该小任务产生的CPU耗能;④ N_{ij}^{dw} 表示该任务下载结果数

据匹配的下行带宽;⑤ C_{ij}^{out} 表示该任务计算产生的数据量。以下为本文设计的3种模型:

4.1 能耗模型

将第 i 台计算设备的总功率定义为:

$$P_i(u) = \begin{cases} K \times P_i^{\max} + (1 - K) \times P_i^{\max} \times u, & u > 0, \\ 0, & otherwise \end{cases} \quad (1)$$

其中, $P_i(u)$ 表示 i 台计算设备在负载最大时产生的总能耗, K 表示 i 台计算设备在无任务状态下产生的能耗百分比, u 为CPU利用率。计算设备的负载将随着时间不断变化,现假设 $u(t)$ 是CPU利用率的变化函数,则从 t_0 时刻开始,计算设备在单位时间 t 内的能耗定义为:

$$Energy_i(t_0) = \int_{t_0}^{t_0+t} P_i(u(t)) dt \quad (2)$$

则 $1+m+n$ 个计算设备在时间 t 内产生的总能耗定义为:

$$Energy_{sum}(t_0) = \sum_{i=1}^{1+m+n} Energy_i(t_0) \quad (3)$$

4.2 时延模型

将计算设备执行任务时间(本文忽略任务排队时间)定义为时延,则将所需时间定义为:

$$ET = \frac{1}{subnum} \sum_{i=1}^{subnum} (FT_{task_i} - ST_{task_i}) \quad (4)$$

其中, $subnum$ 表示设备应用共被切分为 $subnum$ 个小任务, ST_{task_i} 表示第 i 个小任务开始执行的时间, FT_{task_i} 表示第 i 个小任务执行结束的时间。

4.3 网络使用量

网络使用量表示单位时间内所有设备应用产生的数据传输总量。如果网络使用量过高,则会造成拥堵。网络使用量定义如下:

$$BW = \frac{1}{T_{unit}} \sum_{i=1}^{anum} (TL_i \times TS_i) \quad (5)$$

其中, T_{unit} 表示单位时间, $anum$ 表示边缘设备应用须处理任务的总数, TL_i 表示第 i 个设备应用处理任务产生的时延, TS_i 表示第 i 个设备应用处理任务产生的数据传输总量。

5 改进DQN算法RSS2E-DQN设计

5.1 MDP决策过程

5.1.1 状态空间

本文将移动边缘计算任务卸载问题建模为马尔可夫决策过程^[12]。假设边缘设备数据处理任务当前时间步 t 的状态为 s_t ,则切分后所有小任务的状态集合为 $S_t = \{N^{up}, C^{in}, M, N^{dow}, C^{out}, U_1, U_2, \dots, U_{2+m}\}$ 。 N^{up} 表示当前小任务接收数据匹配的上行带宽, C^{in} 表示同一任务输入的数据量, M 表示部署该小任务所需的CPU资源, N^{dow} 表示该任务下载结果数据匹配的下行带宽, C^{out} 表示该任务计算产

生的数据量, U_i 表示在时间步 t 第 i 个计算设备的CPU利用率。假设每个小任务都只能选择在本本地或边缘服务器上执行计算任务,则只用为其匹配1个云计算中心、1个本地设备和 m 个边缘服务器,总计 $2+m$ 个计算设备供其使用。

5.1.2 动作空间

$A_i = (0, 1, 0, \dots, 0)(0/1)_i^j$ 本文将动作空间定义为MEC平台中边缘节点在当前状态下可能会做出的动作集合。要将任务列表中的小任务卸载到相应的计算设备,规定动作空间与所需计算设备一一对应,用 a_i 表示第 i 个小任务是否卸载到第 j 个计算设备上,例如动作空间 a_i 表示第 i 个任务根据策略被卸载到第2个计算设备进行执行。因此,在包含 $2+m$ 个计算设备的边缘异构集群中,所有动作可以表示为动作空间 $A = (a_1, a_2, \dots, a_{2+m})$ 。定义了动作空间后,接下来定义本文的奖励函数。

5.1.3 奖励函数

本文将综合评估MEC平台所有设备的能耗、时延以及网络使用量,以评估任务卸载策略的优劣。由于异构的计算设备在算力、能耗等方面都具有差异,为了降低这些异构设备间性能差异导致的奖励值差异对实验造成的影响,用 $z-score$ 标准化方法对能耗、时延、网络使用量进行正规化处理^[13]。

当数据序列为 $\{x_1, x_2, \dots, x_{num}\}$ 时,得到以下算式:

$$\begin{cases} \bar{x} = \frac{1}{num} \sum_{i=1}^{num} x_i \\ \tau = \sqrt{\frac{1}{num-1} \sum_{i=1}^{num} (x_i - \bar{x})^2} \\ y_i = \frac{x_i - \bar{x}}{\tau} \end{cases} \quad (6)$$

分别将式(2)、式(4)、式(5)代入式(6)中,可求出计算设备的能耗、产生时延、网络使用量的正规化值,进而将奖励函数表示为:

$$\begin{cases} F = \sigma \sum_{i=1}^{1+m+n} ZS_i^{eg} + \xi \sum_{i=1}^{1+m+n} ZS_i^{ct} + \varphi \sum_{i=1}^{1+m+n} ZS_i^{cw} \\ \sigma + \xi + \varphi = 1 \end{cases} \quad (7)$$

在式(7)中, ZS_i^{eg} 、 ZS_i^{ct} 、 ZS_i^{cw} 分别表示第 i 个设备能耗、时延、网络使用量的正规化值, σ 、 ξ 、 φ 表示能耗、时延、网络使用量所占权重,同时 $\sigma + \xi + \varphi = 1$ 。

5.2 DQN算法

传统以MDP建模的算法不能很好地优化任务卸载策略,主要因为状态转移概率难以预测。而基于深度强化学习的(Deep Reinforcement Learning, DRL)算法在不预先输入任何数据的前提下,可以依靠自主学习和自我感知能力在不断的学习、训练过程中,找到最佳动作对应的最优决策,因此成为解决这一问题的最优算法。最常见的DRL算法是DQN算法。

DQN算法的目标是得到动作执行过后的预计奖励,即Q值。Q值是一个估计值,不是确切的数值。DQN将边缘

节点的当前状态输入深度神经网络进行训练,并通过增加经验池和目标网络提升了算法的兼容性与稳定性。DQN使用需要训练的主网络与记录Q值的目标网络进行训练,并使用损失函数对模型参数进行训练。损失函数定义为目标网络Q的估计值与主网络输出的平方差:

$$\begin{cases} Loss = (Q_i(s, a) - Q(s, a))^2 \\ Q_i(s, a) = reward + \gamma Q'(s', \arg \max_{a'} Q'(s', a')) \end{cases} \quad (8)$$

$\arg \max_{a'} Q'(s', a')$ 其中, γ 为奖励折扣系数, $Q'(s', a')$ 为

目标网络下一步状态 s' 做出动作 a' 后得到的估计奖励值,而表示下一步状态 s' 中拥有最高Q值估计的动作 a' 。

5.3 同策略经验回放

传统的DQN算法虽然能找到最接近Q值的最佳策略,但是该算法只能在预测值能很好反映真实环境状态时才有较好效果。在真实的MEC环境中,由于设备和网络的多样性与复杂性,往往不能得到准确的预测值。因此,采取同策略经验回放,能很好地改善预测值不够准确这一问题。具体来说,将每一次卸载的每一步信息都记录下来,包括卸载的状态、动作、实时奖励和下一状态。整个卸载完成后,计算出每一步的实时奖励^[14]。将这些信息转化为四元数组 (S, A, R, S') 存储在回放数组(replay buffer)中,当任务卸载完成若干轮,收集到足够多当前策略下的经验后,从数组中抽取任意一个批处理四元数组对主网络进行训练,进而指导目标网络的更新^[15]。由于同策略算法的限制,目标网络只能更新一次,在目标网络更新后,要将回放数组清空,在新的卸载策略下收集新的训练经验。如此往复,找到移动边缘计算异构设备的最优任务卸载策略,降低边缘设备排队和任务推理的时间延迟,实现MEC平台的任务卸载策略优化。

5.4 基于熵正则的动作筛选

目标网络输出的是动作空间中动作的概率值,当某个状态下目标网络输出的概率值集中在某个动作时,则认定该动作为最佳动作的概率较大。改进算法中用熵(Entropy)来标定这一指标^[16]。熵值越小,说明优良动作出现的概率较为集中,此时筛选出最佳动作的概率较大;熵值越大,说明优良动作较为分散,不利于最佳动作的筛选。但是如果熵值过小,则智能体会安于现状,不去学习和感知更好的策略。所以为了让熵值不会太小,将熵作为正则项放入目标网络的函数中。目标网络输出的是维度为 $|A|$ 的向量,表示一个动作在动作空间中出现的概率^[17]。该概率用熵来定义:

$$H(s; \theta) = Entropy[\pi(\cdot | s; \theta)] = -\sum_{a \in A} \pi(a | s; \theta) \cdot \ln \pi(a | s; \theta) \quad (9)$$

其中,熵 $H(s; \theta)$ 只与智能体当前的状态 s 和网络参数 θ 相关。在大多数情况下,设定熵值较大,从而使智能体去探索更多动作,找到更好的策略。此时 $E_s[H(S; \theta)]$ 较

大,修改目标函数为:

$$\max_{\theta} J(\theta) + \lambda \cdot E_s[H(S; \theta)] \quad (10)$$

其中,参数 λ 需要手动调整,最大化过程的求解采用策略梯度求解方法。策略梯度求解过程不是本文的研究重点,此处不赘述。

5.5 算法流程及实现

RSS2E-DQN算法基于同策略经验回放和熵正则思想对传统的DQN算法进行改进,算法流程如图3所示。

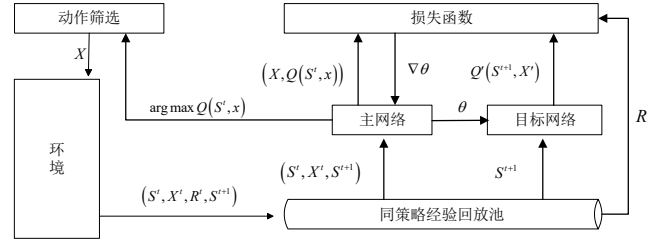


Fig. 3 RSS2E-DQN algorithm flow

图3 RSS2E-DQN算法流程

改进算法在网络结构方面与传统DQN算法相同,主网络和目标网络具有相同结构。主网络对新动作 x_k 进行训练,目标网络则输出对应的预计奖励值。首先,算法根据学习策略随机选取动作或者某状态下Q值最高的动作输入环境中,运用式(10)筛选出最优动作 x_k 放入最优动作列表 X' 中,并将状态、动作、奖励、下一状态这些信息转化为的四元组 (S', X', R', S^{t+1}) 放入同策略经验回放池^[18] 中进行保存,这些四元组都是当前状态下动作与环境交互产生的结果数据。然后,算法在经验池中随机选择一个批四元组对主网络进行训练,目标网络则输出动作列表的Q值 $Q'(S^{t+1}, X')$ 。接下来,利用损失函数计算每个 x_k 的损失值,根据损失值对目标网络参数进行更新。由于同策略经验回放的限制,每回只能对目标网络参数更新一次。参数更新后,经验池清空回放数组,主网络重新训练并收集经验。如此往复,找到最优策略。

依照本文提出的RSS2E-DQN算法,通过定期更新目标网络的参数 θ ,再由主网络收集新的经验数组放入经验回放池中,经过主网络的反复训练,优化目标网络对于边缘节点设备应用采取的动作集合 X' 。优秀动作筛选完毕后,经过智能体与环境的交互,可达到优化移动边缘计算任务卸载策略的目的。

6 实验设计与结果分析

6.1 实验参数设置

本文针对MEC平台任务卸载策略会影响任务执行效率这一问题,构建一个包含1个云数据中心、70台边缘服务器、若干智能移动设备的大规模异构设备集群,采用Edgecloudsim^[19]对MEC平台中的任务卸载算法进行模拟仿真实验,对比各算法在降低能耗、时延、网络使用量三方

面的表现来反映卸载策略的优劣。其中, 包括基于移动设备策略的 MC (Mobile Computing) 算法^[20]、基于边缘服务器策略的 EC (Edge Computing) 算法^[21]、基于 DRL 的 DQN 算法^[22]和本文提出的 RSS2E-DQN 算法。设定每个终端设备在同一时间只能发出一个应用卸载请求, 并参照标准性能评估组织 (Standard Performance Evaluation Corporation, SPEC) 设定的标准, 相关参数参考文献[6]进行设置。实验关键参数详情如表 1 所示。

Table 1 Details of key parameters of simulation experiment

表 1 仿真实验关键参数详情

符号	值	定义
C	1	云数据中心
E	70	边缘服务器
M	若干	移动智能设备
N	50	边缘设备发出任务
λ	0.9	折扣系数
W	10Gb/s	带宽
L	200	算法最大循环轮数
n	512	神经网络深度
A	150	随机抽取经验样本
p	(0, 100)	卸载功率范围
X	300	动作空间
N_0	-180	环境高斯白噪声功率密度 (dBm/Hz)
F	10	边缘服务器总频率 (Hz)
P	4	算法数量
α	0.005	学习率

6.2 实验结果分析

由于实验结果与移动应用在不同时间段的资源使用率相关, 本文采用 Google Cluster Trace 数据集模拟各模块利用率随时间推移所发生的变化, 并且在验证各策略的兼容性和高效性之前, 先从 Google 数据集中选取 1 200 个应用对各神经网络进行训练, 随后选取其他数据对训练后的网络模型进行检验, 以保障实验结果的严谨性、算法产生策略的高效性和可用性。

实验仿真结果如图 4—图 6 所示。

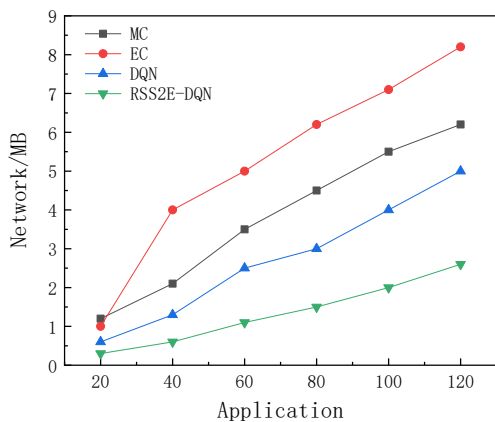


Fig. 4 Network usage comparison of each algorithm in simulation experiment

图 4 各算法在仿真实验中网络使用量对比

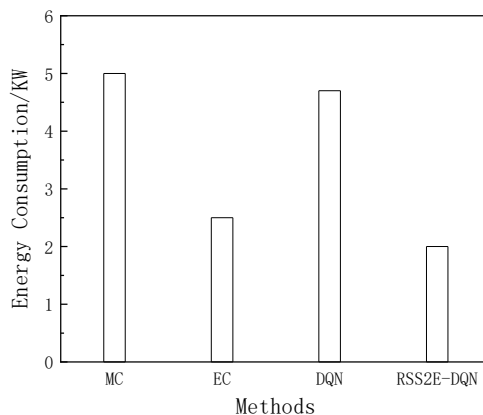


Fig. 5 Energy consumption comparison of each algorithm in simulation experiment

图 5 各算法在仿真实验中能耗对比

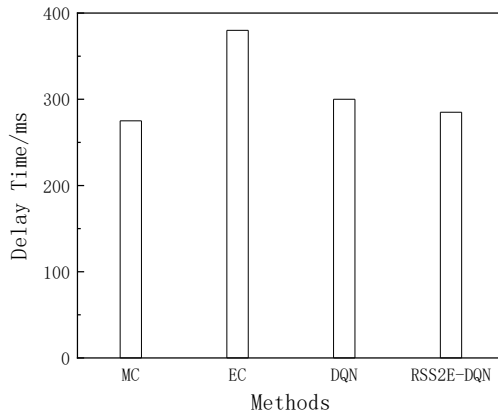


Fig. 6 Time delay comparison of each algorithm in simulation experiment

图 6 各算法在仿真实验中时延对比

从图 4 可以看出, 随着设备应用的不断增多, EC 算法的网络使用量最高, 这是因为其算法策略是将所有任务都卸载到边缘服务器进行处理, 在提高了边缘服务器利用率的同时, 也增加了网络带宽的负荷, 会在终端用户集中的区域造成网络拥堵^[23]。从图 5 可以看出, 基于移动设备策略的 MC 算法随着设备应用的增加, 能耗是最高的, 这是因为其算法策略是所有任务都在本地执行, 虽然节约了任务上传下载的带宽, 但是增加了移动智能设备的能耗, 在单位时间内, 电池的消耗会更快。从图 6 可以看出, EC 算法和 DQN 算法随着设备应用的数量不断增加, 其处理任务产生的时延较长。这是因为 EC 算法没有对任务进行分类即全部进行了卸载, 虽然边缘服务器能快速地处理某些任务, 但代价是更高的时延^[24-25]。综上所述, 本文提出的基于同策略经验回放和熵正则思想的 RSS2E-DQN 算法在能耗、时延、网络使用量这 3 个决定奖励大小的指标中表现最好。因为其是基于细粒度任务卸载的, 能将任务卸载到

合适的计算设备,且拥有可以收集经验样本的同策略经验回放池,可以反复训练经验样本,提高样本的自我学习和环境感知能力,并能够感知异构设备间的性能差异,通过基于熵正则的动作筛选后,实现移动边缘计算平台中移动设备间的任务卸载策略优化。

7 结语

本文搭建了一个基于移动边缘计算平台三层服务架构的任务卸载模型,结合MEC平台的实际应用场景,利用同策略经验回放和熵正则改进深度强化学习算法,优化了MEC平台的任务卸载策略,并设计了实验对3种传统算法和改进算法的能耗、时延、网络使用量3个指标进行对比分析,验证了改进算法在降低能耗、时延和网络使用量方面具有更优越的性能。但是,本文仅对任务卸载算法和策略进行了着重研究,其实任务调度在整个架构中也同样重要。未来可将任务调度和任务卸载算法与策略研究相结合,优化架构中的资源调度和任务卸载方案。

参考文献:

- [1] LI Y K, ZHANG T, CHEN J L. Width-twin networks for edge computing applications [J]. *Acta Automatica Sinica*, 2020, 46(10):2060-2071.
李毅楷,张通,陈俊龙.面向边缘计算应用的宽度孪生网络[J].*自动化学报*,2020,46(10):2060-2071.
- [2] YU G M, LI F P. Media innovation in 5G era: research on application paradigm of edge computing [J]. *Journal of Shanxi University: Philosophy and Social Sciences Edition*, 2020, 43(1):65-69.
喻国明,李凤萍.5G时代的传媒创新:边缘计算的应用范式研究[J].*山西大学学报:哲学社会科学版*,2020,43(1):65-69.
- [3] ZHANG Z, LI F, LIN C, et al. Choosing appropriate AI-enabled edge devices, not the costly ones [C]//2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS), 2021: 201-208.
- [4] LIU L Q, CHANG Z, GUO X J, et al. Multi-objective optimization for computation offloading in fog computing [J]. *IEEE Internet of Things Journal*, 2017, 5(1):283-294.
- [5] CAO T F, LIU Y L, WANG X Y. Edge computing service offloading algorithm based on improved deep reinforcement learning [J]. *Computer Applications*, 2023, 43(5):1543-1550.
曹腾飞,刘廷亮,王晓英.基于改进深度强化学习的边缘计算服务卸载算法[J].*计算机应用*,2023,43(5):1543-1550.
- [6] LU H F, GU C H, LUO F, et al. Research on task offloading of mobile edge computing based on deep reinforcement learning [J]. *Journal of Computer Research and Development*, 2020, 57(7):1539-1554.
卢海峰,顾春华,罗飞,等.基于深度强化学习的移动边缘计算任务卸载研究[J].*计算机研究与发展*,2020,57(7):1539-1554.
- [7] KUANG Z F, CHEN Q L, LI L F, et al. Multi-user edge computing task offload scheduling and resource allocation algorithm based on deep reinforcement learning [J]. *Journal of Computer Science*, 2022, 45(4): 812-824.
邝祝芳,陈清林,李林峰,等.基于深度强化学习的多用户边缘计算任务卸载调度与资源分配算法[J].*计算机学报*,2022,45(4):812-824.
- [8] SHENG Y, ZHU Z W, ZHU C Y, et al. Research on multi-objective edge task scheduling based on deep reinforcement learning [J]. *Electronic Measurement Technology*, 2023, 46(8):74-81.
盛煜,朱正伟,朱晨阳,等.基于深度强化学习的多目标边缘任务调度研究[J].*电子测量技术*,2023,46(8):74-81.
- [9] JAISWAL A, BABU A R, ZADEH M Z, et al. A survey on contrastive self-supervised learning [J]. *Technologies*, 2020, 9(1):2.
- [10] GRAVES A, MOHAMED A R, HINTON G. Speech recognition with deep recurrent neural networks [C]//2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013: 6645-6649.
- [11] BAI Z, ZHANG Z, ZHU Y, et al. Pipe switch: fast pipelined context switching for deep learning applications [C]//Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation, 2020:499-514.
- [12] ALFAKIH T, HASSAN M M, GUMAEI A, et al. Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA [J]. *IEEE Access*, 2020(8):54074-54084.
- [13] SADIKI A, BENTAHAR J, DSSOULI R, et al. Deep reinforcement learning for the computation offloading in MIMO-based edge computing [J]. *Ad Hoc Networks*, 2023, 141:103080.
- [14] LI M S, GAO J, ZHAO L, et al. Deep reinforcement learning for collaborative edge computing in vehicular networks [J]. *IEEE Transactions on Cognitive Communications and Networking*, 2020, 6(4):1122-1135.
- [15] LI D J, XU S Y, LI P Y. Deep reinforcement learning-empowered resource allocation for mobile edge computing in cellular V2X networks [J]. *Sensors*, 2021, 21(2):372.
- [16] XIAO Z. Vehicular task offloading via heat-aware MEC cooperation using game-theoretic method [J]. *IEEE Internet of Things Journal*, 2020, 7(3):2038-2052.
- [17] SHENG M, DAI Y, LIU J, et al. Delay-aware computation offloading in NOMA MEC under differentiated uploading delay [J]. *IEEE Transactions on Wireless Communications*, 2020, 19(4):2813-2826.
- [18] ZHAN W, LUO C, MIN G, et al. Mobility-aware multi-user offloading optimization for mobile edge computing [J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(3):3341-3356.
- [19] HUY H V, HO T M, LE L B. Mobility-aware computation offloading in MEC-based vehicular wireless networks [J]. *IEEE Communications Letters*, 2020, 24(2): 466-469.
- [20] JOSILO S, DAN G. Joint management of wireless and computing resources for computation offloading in mobile edge clouds [J]. *IEEE Transactions on Cloud Computing*, 2021, 9(4):1507-1520.
- [21] YUAN H, ZHOU M. Profit-maximized collaborative computation offloading and resource allocation in distributed cloud and edge computing systems [J]. *IEEE Transactions on Automation Science and Engineering*, 2021, 18(3):1277-1287.
- [22] ZHAO T, ZHOU S, SONG L, et al. Energy-optimal and delay-bounded computation offloading in mobile edge computing with heterogeneous clouds [J]. *China Communications*, 2020, 17(5):191-210.
- [23] KUANG Z, LI L, GAO J, et al. Partial offloading scheduling and power allocation for mobile edge computing systems [J]. *IEEE Internet of Things Journal*, 2019, 6(4):6774-6785.
- [24] YANG S M, SHAN Z, DING Y, et al. Survey of research on deep reinforcement learning [J]. *Computer Engineering*, 2021, 47(12):19-29.
杨思明,单征,丁煜,等.深度强化学习研究综述[J].*计算机工程*, 2021,47(12):19-29.
- [25] ABADI A, AGARWAL A, BARHAM P, et al. Tensor flow: large-scale machine learning on heterogeneous distributed systems [DB/OL]. <https://arxiv.org/pdf/1603.04467>.