

无监督深度学习移动边缘计算卸载资源分配

张国平, 陈 雪, 徐洪波

(华中师范大学 物理科学与技术学院, 湖北 武汉 430079)

摘 要: 计算卸载作为移动边缘计算的核心技术之一, 有效缓解了移动设备处理计算密集型任务的压力, 为降低能量消耗和任务执行延迟提供了可能。计算卸载的资源分配是一个优化问题, 其计算量大, 复杂度高, 阻碍了其在实际应用中的应用。针对以上问题, 提出一种基于无监督学习的多用户移动边缘计算卸载框架, 通过将任务卸载到边缘服务器执行来降低移动设备能耗。在满足时延约束和发射功率约束的条件下, 通过联合优化移动边缘计算系统的卸载决策和资源分配, 使移动设备的能量消耗加权最小。仿真结果表明, 在计算复杂度较低的情况下, 经过训练的卸载网络能够有效解决移动设备上计算密集型任务的计算卸载和资源分配问题。

关键词: 移动边缘计算卸载; 无监督深度学习; 卸载决策; 资源分配

DOI: 10.13757/j.cnki.cn34-1328/n.2021.04.001

中图分类号: TP391

文献标志码: A

文章编号: 1007-4260(2021)04-0001-07

Resource Allocation Based on Unsupervised Deep Learning in Mobile Edge Computation Network

ZHANG Guoping, CHEN Xue, XU Hongbo

(College of Physical Science and Technology, Central China Normal University, Wuhan 430079, China)

Abstract: Mobile edge computation offloading (MECO), as one of the core technologies of the Mobile edge computation, is effective to reduce the energy consumption and task execution delay for tackling computation-intensive tasks on mobile device (MD). The resource allocation of mobile edge computation (MEC) is an optimization problem, however, the existing large amount of computation and high complexity may hinder its practical application. To reduce the computation, we propose a multiuser MEC framework based on unsupervised deep learning (DL) to reduce energy consumption by offloading tasks to edge servers. The offloading decision and resource allocation are jointly optimized to minimize energy consumption of MDs under latency constraint and transmit power constraint. Simulation results demonstrate that the trained offloading network is effective to solve the binary offloading and resource allocation problem for tackling computation-intensive tasks on mobile device with low computational complexity.

Key words: mobile edge computation offloading; unsupervised deep learning; offloading decision; resource allocation

随着5G通信技术发展、流量爆炸式增长和移动智能终端普及,移动用户对服务质量的要求也不断提高,尤其是时延和能耗问题。为了解决时延敏感型任务、计算密集型与移动设备资源有限之间的矛盾,欧洲电信标准协会ETSI提出移动边缘计算(MEC)技术^[1]。MEC技术是一种新型的无线网络接入模式,通过在移动用户附近的接入点(Access Points)部署高性能服务器,提高网络边缘的计算能力。移动设备通过无线网络将计算密集型任务卸载到边缘服务器上执行,能够有效降低任务执行时延和移动设备的能量消耗,从而提高用户的服务体验。

收稿日期:2021-09-01

基金项目:中央高校基本科研业务费项目资助(CCNU20TS008)

作者简介:张国平(1969—),男,湖北汉川人,博士,华中师范大学物理科学与技术学院教授,博士生导师,主要研究方向为信号处理、光电信息技术等。E-mail:gpzhang@mail.ccnu.edu.cn

近年来,工业界和学术界普遍认为合理的卸载决策和资源分配能够大大提升MEC系统的性能^[2]。但是,由于MEC中卸载决策和资源分配的联合优化问题多为非凸的NP-hard问题,传统数学方法解该类问题计算量大,复杂性高,这为MEC网络带来巨大的负担。随着深度学习(DL)的快速发展,其在自然语言处理、语音识别和计算机视觉等领域取得了巨大成功,研究表明DL也可以用来处理复杂的通信问题,例如信道预编码、功率控制、信道估计等^[3-5]。为了降低MEC联合优化问题的计算复杂度和任务执行时延,本文提出了基于无监督深度学习的计算卸载算法。通过联合优化卸载决策和通信资源分配,在满足时延约束的条件下,最小化移动设备的能量加权和。

1 相关工作

为了提高MEC系统的性能,学者们提出许多基于数学分析的传统算法来解决计算卸载问题。文献[4]提出了一种高效的一维搜索算法来处理功率约束的时延最小化问题,该算法根据应用缓冲区的缓冲状态、本地设备和远程传输的可用功率来获得最优的卸载决策,以及移动设备(Mobile Device, MD)和MEC服务器之间的信道状态信息(Channel State Information, CSI)。然而,该算法中MD需要MEC服务器的反馈才能做出卸载决策,增加了信令开销。文献[5]为了获得高效的卸载决策,设计了一种基于博弈论的分布式计算卸载方法,该算法需要MD和MEC进行多次通信迭代。类似的,文献[6]和[7]通过迭代更新完全卸载决策,解决卸载决策和资源联合分配问题。虽然这些算法能够有效解决计算卸载问题,但由于其较高的计算复杂度和较长的时延,并不适用于执行实时卸载任务。

随着人工智能的发展,机器学习在许多领域取得显著成功,而深度学习性能优于传统的机器学习,并被广泛用于通信领域。文献[8]采用深度强化学习方法处理无线MEC系统的资源分配问题。文献[9]提出了一种基于DL的智能节能部分卸载方案,根据设备剩余能量、应用能耗、信道条件、卸载数据量、计算能力等因素选择最优卸载方案,最小化损失函数和传输时延。文献[10]利用DL方法解决MEC卸载问题,通过优化移动环境状态变量最小化损失函数。但是,前面提到的深度学习方法都是有监督学习,很难获得合适的训练集,因此,本文设计了一种基于无监督学习的计算卸载方法,以提高MEC系统的性能。

2 多用户移动边缘计算卸载模型

2.1 网络模型

本文考虑了一个多用户移动边缘计算卸载系统,如图1所示,包含 K 个单天线移动设备和一个部署有高性能服务器的单天线基站。时间 T 被划分为 K 个时隙并分配给 K 个移动设备。在每个时隙里,设备选择将任务通过时分多址技术(TDMA)卸载到边缘服务器执行或者在本地执行。假设移动端任务可分割,即可以部分在本地执行,部分卸载到服务器上执行。为了使基站选择合适的移动设备进行任务卸载,可假设基站能获得所有设备的CSI和计算能力,而且在一个时隙内信道保持不变。

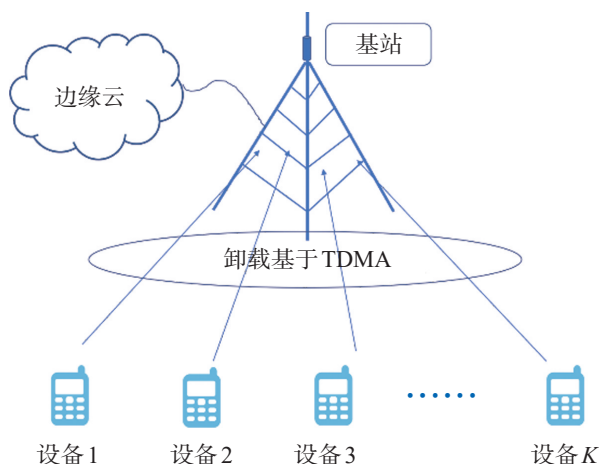


图1 多用户移动边缘计算卸载网络

2.2 本地执行模型

对于本地执行模型,根据 $P = \varepsilon f^3$ 计算本地设备执行任务的能量消耗^[11],其中, ε 由芯片结构决定, f 表示中央处理器(CPU)的计算速度,单位为转数/秒。用 B_k 、 C_k 、 f_k 分别表示设备 k 的任务大小、处理单位比特任务 CPU 的转数和计算速度,并假设每个设备的 C_k 、 f_k 保持不变。用 a_k 表示设备的卸载决策,假设任务可分割,则 $a_k \in [0, 1]$ 代表卸载任务的比率。设备 k 处理 B_k 任务 CPU 需要的转数为 $(1 - a_k) C_k B_k$, 所以设备 k 本地执行任务的时间可表示为

$$\tau_{k,l} = \frac{(1 - a_k) C_k B_k}{f_k}, \quad (1)$$

设备 k 本地执行任务的能量消耗可表示为

$$E_{l,k} = P \tau_{k,l} = \varepsilon (1 - a_k) B_k C_k f_k^2. \quad (2)$$

2.3 卸载模型

计算卸载包括3部分:1)移动端上行任务传输(采用TDMA技术无线接入边缘服务器);2)边缘服务器执行卸载任务;3)服务器返回执行结果。受带宽和发射功率的限制,任务执行和结果返回的时延远小于任务上传,因此,本文只考虑任务上传过程的资源分配。设备 k 上传任务时的传输速率可表示为

$$R_k = W \log \left(1 + \frac{p_k g_k^2}{N_0} \right), \quad (3)$$

其中, p_k 、 g_k 表示设备 k 的发射功率和信道增益, N_0 表示复杂高斯白噪声的方差, W 表示带宽。设备 k 卸载任务需要的时间可表示为

$$\tau'_k \triangleq \frac{a_k B_k}{R_k} = \frac{a_k B_k}{W \log \left(1 + \frac{p_k g_k^2}{N_0} \right)}. \quad (4)$$

假设设备分配的发送时隙为 τ_k , 为保证任务能完全上传,需满足条件: $\tau'_k \leq \tau_k$ 。设备 k 卸载任务的能耗可表示为

$$e_{\text{off},k} = p_k \tau_k. \quad (5)$$

2.4 问题描述

为了降低移动设备能量消耗,本文在满足时延和功率约束的条件下,通过优化卸载决策、发射时间和发射功率,最小化移动设备的能耗和,相应的优化问题表示为

$$\begin{aligned} \text{P1: } \min_{a_k, p_k, \tau_k} & \sum_{k=1}^K \beta_k [p_k \tau_k + \varepsilon (1 - a_k) B_k C_k f_k^2] \\ \text{s.t. } & \begin{cases} \text{C1: } \sum_{k=1}^K \tau_k \leq T, \\ \text{C2: } \frac{(1 - a_k) C_k B_k}{f_k} \leq T, \forall k, \\ \text{C3: } 0 \leq p_k \leq p_{\max}, \forall k, \\ \text{C4: } 0 \leq a_k \leq 1, \forall k, \\ \text{C5: } \tau'_k \leq \tau_k, \forall k, \end{cases} \end{aligned} \quad (6)$$

其中, β_k 为正定的权重因子,用来保证用户的公平性。由于传统数学分析方法求解该问题存在计算量大、复杂度高等不足,因此本文使用无监督深度学习方法进行求解。

3 基于无监督深度学习的卸载算法

本文采用无监督深度神经网络求解优化问题 P1, 其中信道增益作为网络输入, 卸载策略、时间分配和功率分配作为网络输出。首先介绍基于全连接层网络的深度学习框架, 接着将问题 P1 转化为无约束的优化问题, 最后采用无监督学习方法进行求解。

3.1 参数化深度学习模型

本文采用的全连接层神经网络包括一个输入层、两个隐藏层和一个输出层,如图2所示,其中 K 个设备的信道增益 \mathbf{g} 作为网络输入层,卸载策略 \mathbf{a} 、时间分配 \mathbf{t} 和功率分配 \mathbf{p} 作为输出层。 $L_i, i=1,2$ 表示第 i 个隐藏层节点的个数,第 i 个隐藏层的输出表示为

$$\mathbf{x}_i = \text{ReLU}(\text{BN}(\mathbf{W}_i \mathbf{x}_{i-1} + \mathbf{b}_i)), \quad (7)$$

其中, $\mathbf{x}_i, \mathbf{x}_{i-1}$ 分别表示当前层和前一层的输出向量, \mathbf{W}_i 为第 i 层节点的权重矩阵, \mathbf{b}_i 为偏置矩阵,BN(batch normalization)表示批量归一化,ReLU表示线性修正函数 $\max(x, 0)$ 。该网络选择sigmoid函数作为激活层函数,其形式为

$$\text{Sig}(x) = \frac{1}{1 + \exp(-x)}. \quad (8)$$

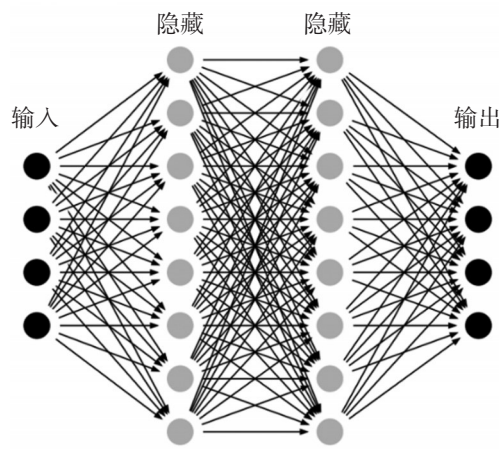


图2 全连接层神经网络结构图

因此,全连接层神经网络的输出可以用参数化模型表示为

$$\mathbf{x}_o = \mathbf{x}_i = \varphi(\mathbf{g}, \boldsymbol{\theta}), \quad (9)$$

其中, \mathbf{g} 表示输入的信道增益, $\boldsymbol{\theta}$ 表示网络参数 $\{\mathbf{W}, \mathbf{b}\}$ 。

3.2 基于深度学习的问题描述及算法

由于标准的深度学习问题为无约束问题,处理深度学习问题的方法并不能直接用于处理带约束的MECO问题。常用的消除约束的方法包括:拼接自定义激活层作为输出层和在损失函数上增加额外的约束项以惩罚约束违反。本文也采用以上两种方法将优化问题P1转化为无约束深度学习问题。将优化变量参数转化为网络参数的函数:

$$\begin{aligned} \mathbf{a} &= \varphi_a(\mathbf{g}, \boldsymbol{\theta}), \\ \mathbf{t} &= \varphi_t(\mathbf{g}, \boldsymbol{\theta}), \\ \mathbf{p} &= \varphi_p(\mathbf{g}, \boldsymbol{\theta}), \end{aligned} \quad (10)$$

将优化变量参数化后,优化问题可以转化为以下形式:

$$\begin{aligned} \text{P2: } \min_{\boldsymbol{\theta}} \quad & \sum_{k=1}^K \beta_k [\varphi_p^k(\mathbf{g}, \boldsymbol{\theta}) \varphi_t^k(\mathbf{g}, \boldsymbol{\theta}) \boxplus \varepsilon (1 - \varphi_a^k(\mathbf{g}, \boldsymbol{\theta})) B_k C_k f_k^2] \\ \text{s.t. } \quad & \begin{cases} \text{C1: } \sum_{k=1}^K \varphi_t^k(\mathbf{g}, \boldsymbol{\theta}) \leq T, \\ \text{C2: } \frac{(1 - \varphi_a^k(\mathbf{g}, \boldsymbol{\theta})) C_k B_k}{f_k} \leq T, \quad \forall k, \\ \text{C3: } 0 \leq \varphi_p^k(\mathbf{g}, \boldsymbol{\theta}) \leq p_{\max}, \quad \forall k, \\ \text{C4: } 0 \leq \varphi_a^k(\mathbf{g}, \boldsymbol{\theta}) \leq 1, \quad \forall k, \\ \text{C5: } \tau_k \leq \varphi_t^k(\mathbf{g}, \boldsymbol{\theta}), \quad \forall k, \end{cases} \end{aligned} \quad (11)$$

其中, $\varphi_a^k(\mathbf{g}, \boldsymbol{\theta})$ 、 $\varphi_p^k(\mathbf{g}, \boldsymbol{\theta})$ 、 $\varphi_t^k(\mathbf{g}, \boldsymbol{\theta})$ 分别表示设备 k 的优化变量 \mathbf{a} 、 \mathbf{p} 、 \mathbf{t} 的参数化函数。为了处理问题 P2 中的约束, 在网络末端拼接不同的自定义激活层。而为了满足时间分配约束 C1, 采用归一化函数作为网络的激活层, 其定义如下,

$$\mathbf{t} = T \min(1, \|\varphi_t(\mathbf{g}, \boldsymbol{\theta})\|) \frac{\varphi_t(\mathbf{g}, \boldsymbol{\theta})}{\|\varphi_t(\mathbf{g}, \boldsymbol{\theta})\|} \quad (12)$$

同时, 为满足功率分配约束 C3, 且 sigmoid 函数能够将变量映射到 $(0, 1)$ 之间, 将网络的激活层定义为

$$\mathbf{p} = p_{\max} \varphi_p(\mathbf{g}, \boldsymbol{\theta}) \quad (13)$$

计算卸载决策需要满足约束 C2 和 C4, 假设 a_k 表示本地任务卸载比率, 则必须满足 $a_k \geq s_k^+$ 且 $s_k = 1 - \frac{f_k T}{B_k C_k}$, $s_k^+ = \max\{s, 0\}$, 因此, 激活函数可定义为

$$\mathbf{a} = \varphi_a(\mathbf{g}, \boldsymbol{\theta}) (1 - s_k^+) + s_k^+ \quad (14)$$

通过以上自定义激活层, 优化问题 P2 可转化为

$$\begin{aligned} \text{P3: } \min_{\boldsymbol{\theta}} \sum_{k=1}^K \beta_k [\varphi_p^k(\mathbf{g}, \boldsymbol{\theta}) \varphi_t^k(\mathbf{g}, \boldsymbol{\theta}) + \varepsilon (1 - \varphi_a^k(\mathbf{g}, \boldsymbol{\theta})) B_k C_k f_k^2] \\ \text{s.t. C5: } \tau_k - \varphi_t^k(\mathbf{g}, \boldsymbol{\theta}) \leq 0, \forall k, \end{aligned} \quad (15)$$

通过在神经网络末端拼接不同的自定义激活层, 优化问题 P3 只需要满足约束 C5。

为了满足约束 C5, 在神经网络的损失函数中添加惩罚项, 惩罚约束违反。由于惩罚项只考虑约束违反的情况, 在约束满足的情况下可以忽略 $\tau_k - \varphi_t^k(\mathbf{g}, \boldsymbol{\theta})$ 的值, 因此定义如下函数:

$$H(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \quad (16)$$

约束 C5 等价于 $H(\tau_k - \varphi_t^k(\mathbf{g}, \boldsymbol{\theta})) = 0, \forall k$, 即表示约束 C5 不满足时的惩罚。引入惩罚项后, 优化问题 P3 可转化为

$$\begin{aligned} \text{P4: } \min_{\boldsymbol{\theta}, \lambda_B} \text{loss}(\boldsymbol{\theta}, \lambda_B) = \\ E \left\{ \sum_{k=1}^K \beta_k [\varphi_p^k(\mathbf{g}, \boldsymbol{\theta}) \varphi_t^k(\mathbf{g}, \boldsymbol{\theta}) + \varepsilon (1 - \varphi_a^k(\mathbf{g}, \boldsymbol{\theta})) B_k C_k f_k^2] + \lambda_B \sum_{k=1}^K H(\tau_k - \varphi_t^k(\mathbf{g}, \boldsymbol{\theta})) \right\} \end{aligned} \quad (17)$$

其中, λ_B 为超参数, 用于平衡损失函数和惩罚项。

综上所述, 通过引入自定义激活层和惩罚项, 成功将原优化问题转化为无约束优化问题。

优化问题 P4 为标准的深度学习问题, 通过随机梯度下降法, 最小化损失函数 $\text{loss}(\boldsymbol{\theta}, \lambda_B)$, 无监督深度学习网络实现从信道样本到卸载决策和资源分配策略的映射, 具体算法如算法 1 所示。与传统的有监督学习相比, 无监督学习不需要有标签的训练集, 而有标签的数据集多是通过传统数学分析方法获得, 计算量大, 复杂度高。在输入信道样本后, 经过训练的神经网络, 直接输出最优分配策略, 有效缩短时延, 并降低复杂度。

算法 1 基于无监督深度学习计算卸载算法

- 1) 初始化网络参数 $\boldsymbol{\theta}$, 超参数 λ_B , 步长 α , 总迭代次数 T ;
- 2) 生成 20 000 个信道数据样本并且设定 minibatch-size = 1 000;
- 3) for $i = 1:T$ do;
- 4) 更新全连接层神经网络参数 $\boldsymbol{\theta}$ 通过最小化 $\text{loss}(\boldsymbol{\theta}, \lambda_B)$;
- 5) 更新惩罚项参数 λ_B , 通过次梯度下降法更新 $\lambda_B^{(i+1)} \leftarrow \lambda_B^{(i)} + \alpha \nabla_{\lambda_B} \text{loss}(\boldsymbol{\theta}, \lambda_B)$;
- 6) end for

4 实验结果

本小节通过数值仿真验证了无监督深度学习算法的有效性。考虑多用户 MEC 系统, 其由一个部署有高性能服务器的单天线基站和 K 个单天线移动设备构成。假设无线信道为大规模瑞利衰落信道, 大规模衰落平均功率损耗为 10^{-6} , 其他仿真参数如表 1 所示。

表1 仿真参数设置

变量名	参数涵义	设定值
W	总带宽	10 MHz
K	移动设备数量	4 ~ 20
P_{\max}	设备最大发射功率	23 dBm
N_0	复杂高斯白噪声方差	10^{-9} W
B_k	设备 k 任务大小	(100 ~ 300) kB
C_k	计算每比特数据CPU转数	(200 ~ 400) cycles/bit
T	总时间槽	0.5 s
f_k	本地设备 k 计算能力	(1 ~ 2) GHz
β_k	等公平权重因子	1
ε	芯片参数	10^{-28}

通过对比本文提出的无监督深度学习计算卸载机制和最小计算卸载机制,以说明无监督深度学习计算卸载算法的有效性。最小卸载机制(MCOS):该方法优先考虑本地执行,如果本地计算能力不足,将剩余任务以最大发射功率传输到服务器上执行,计算 K 个设备的能耗加权和。无监督计算卸载机制(UDLCOS):通过无监督神经网络优化卸载决策、功率分配、时间分配,在得到最优解的情况下,计算 K 个设备的能耗加权和。

图3为MCOS和UDLCOS两种方案 K 个设备的能耗加权和。可见,随着设备数从4增加到20、总时延为0.5 s时,两种方案的能耗加权和都增加,且MCOS的能耗加权和明显高于UDLCOS,说明在延迟约束下,合理的卸载决策和资源分配能够有效降低设备的能耗,证明了UDLCOS是有效的。图4展示了总时延从0.3 s增加到1.6 s、总设备数为14时,UDLCOS和MCOS两方案对设备能量消耗权重和的影响。可见,随着总时延的增加,MCOS的能耗加权和基本保持不变,而UDLCOS的能耗加权和先随着总时延的增加下降,随后保持不变。对于UDLCOS而言,由于优先考虑本地计算,分配的时间总是能满足本地任务的卸载需求,所以总时延增加对能耗加权和几乎没有影响;而随着总时延的增加,UDLCOS下设备能够将更多的任务卸载到服务器上执行,当总时延进一步增加时,能耗加权和基本保持不变,说明时延已经达到饱和,即满足所有任务卸载的需求。因此,合理的卸载决策和资源分配能够有效降低本地设备的能耗加权和,证明了UDLCOS的有效性。

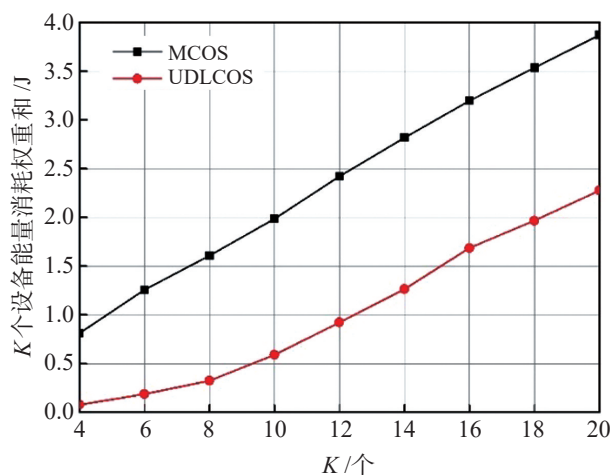


图3 UDLCOS和MCOS下
能量消耗权重和与设备数 K 的关系

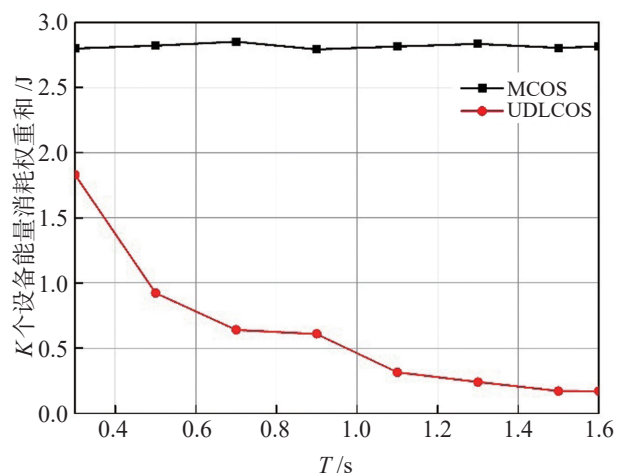


图4 UDLCOS和MCOS下
 K 为14时能量消耗权重和与时延的关系

5 结束语

综上所述,为了缓解移动设备存储能耗有限的问题,本文提出了一种基于无监督深度学习的计算卸

载方案,该方案在规定时延约束下可以有效降低移动端设备能量消耗。本文考虑多用户移动边缘计算卸载系统,其包含一个部署有高性能服务器的单天线基站和 K 个单天线移动设备,在规定时延和功率约束下,通过优化卸载决策、时间分配和功率分配,最小化设备的能耗加权。通过将该约束优化问题转化为无约束优化问题,采用无监督深度学习方法进行求解,经过训练的神经网络能够在低复杂度的情况下,获得最优的卸载决策和资源分配方案,有效地降低了移动边缘计算设备的总能耗。仿真结果亦证明了所提出的卸载方案是有效的。在未来工作中,我们将研究基于联邦学习的移动边缘计算方案。

参考文献:

- [1] 谢人超, 廉晓飞, 贾庆民, 等. 移动边缘计算卸载技术综述[J]. 通信学报, 2018, 39(11): 138-155.
- [2] MUNOZ O, PASCUAL-ISERTE A, VIDAL J. Optimization of radio and computational resources for energy efficiency in latency-constrained application offloading[J]. IEEE Transactions on Vehicular Technology, 2015, 64(10): 4738-4755.
- [3] NACHMANI E, MARCIANO E, LUGOSCH L, et al. Deep learning methods for improved decoding of linear codes[J]. IEEE Journal of Selected Topics in Signal Processing, 2018, 12(1): 119-131.
- [4] TRAN T, POMPILI D. Joint task offloading and resource allocation for multi-server mobile-edge computing networks[J]. IEEE Transactions on Vehicular Technology, 2019, 68(1): 856-868.
- [5] CHEN X, JIAO L, LI W, et al. Efficient multi-user computation offloading for mobile-edge cloud computing[J]. IEEE/ACM Transactions on Networking, 2016, 24(5): 2795-2808.
- [6] BI S, ZHANG Y. Computation rate maximization for wireless powered mobile-edge computing with binary computation offloading[J]. IEEE Transactions on Wireless Communications, 2018, 17(6): 4177-4190.
- [7] WANG J, FENG D, ZHANG S, et al. Joint computation offloading and resource allocation for mec-enabled iot systems with imperfect CSI[J]. IEEE Internet of Things Journal, 2021, 8(5): 3462-3475.
- [8] LI J, GAO H, LV T, et al. Deep reinforcement learning based computation offloading and resource allocation for MEC[C]. Proceedings of the 2018 IEEE Wireless Communications and Networking Conference (WCNC), 2018, 1-6.
- [9] ALI Z, JIAO L, BAKER T, et al. A deep learning approach for energy efficient computational offloading in mobile edge computing[J]. IEEE Access, 2019, 7: 149623-149633.
- [10] GONG Y, LV C, CAO S, et al. Deep learning-based computation offloading with energy and performance optimization[J]. EURASIP Journal on Wireless Communications and Networking, 2020 (1): 69.
- [11] WANG Y, SHENG M, WANG X, et al. Energy-optimal partial computation offloading using dynamic voltage scaling[C]. Proceedings of the IEEE ICC 2015-Workshop on Cloud-Processing in Heterogeneous Mobile Communication Networks (IWCPM), 2015, 2695-2700.