移动场景下基于端—边—云的深度学习模型训练加速研究

Research on Training Acceleration for Deep Learning Models Based on End-Edge-Cloud Collaboration in Mobile Scenarios

陈亦哲[1]，姜肇瑞[2]，冯传奋[1]（1. 山东师范大学，山东 济南 250014；2. 中国海洋大学，山东 青岛 266101）
Chen Yizhe[1]，Jiang Zhaorui[2]，Feng Chuanfen[1]（1.Shandong Normal University，Jinan 250014，China；2.Ocean University of China，Qingdao 266101，China）

摘　要：
针对移动场景下深度学习模型训练加速问题，首次提出高速移动场景下模型训练4种处理流程，设计端—边—云协同的深度学习模型训练加速方案。搭建模型训练时延优化模型，将训练加速问题转化为多目标受限条件下时延最优解问题。基于麻雀觅食与反捕食行为的启发式优化算法，选择最优模型训练卸载策略，为用户提供训练服务并给出最优卸载量。实验表明，端—边—云协同的训练加速方案可降低模型训练时延。

关键词：
模型训练加速；端—边—云协同；麻雀优化算法
doi：10.12045/j.issn.1007-3043.2025.12.014
文章编号：1007-3043（2025）12-0076-07
中图分类号：TN919
文献标识码：A
开放科学（资源服务）标识码（OSID）：

Abstract：
To address the issue of accelerating deep learning model training in mobile scenarios，it proposes four processing flows for model training in high-speed mobile scenarios，and designs a deep learning model training acceleration scheme based on end-edge-cloud collaboration. It presents a model training delay optimization model to transform the deep learning model training acceleration problem into a multi-objective constrained delay optimal solution problem. The scheme selects the optimal model training offloading strategy based on the heuristic optimization algorithm of sparrow's foraging behavior and anti-predation behavior，providing users with training services and determining the optimal offloading amount. Experimental results demonstrate that the end-edge-cloud collaboration training acceleration scheme can significantly reduce model training latency.

Keywords：
Model training acceleration；End-edge-cloud collaboration；Sparrow optimization algorithm

引用格式：陈亦哲，姜肇瑞，冯传奋. 移动场景下基于端—边—云的深度学习模型训练加速研究[J]. 邮电设计技术，2025（12）：76-82.

0 引言

深度学习是机器学习的重要分支，通过构建多层神经网络，使机器能够模拟人类处理多种类型数据的能力，完成认知层面的复杂任务[1-2]。目前，深度学习已成为人工智能领域的研究重点和主流方向，被广泛应用于计算机视觉、自然语言处理等多个领域。然而，随着模型规模、数据量的增长及任务要求的多元化，大规模神经网络模型的训练面临着新的挑战[3]。例如，AmoebaNet[4]、NASNet[5]、GPT-3[6]等模型因其参数量庞大，传统的集中式训练难以满足其实时性需求。

在图像识别领域，深度学习模型的训练方法主要有2种。一是将深度学习模型部署在云端数据中心，移动设备将收集的图像数据上传至云端，由云端完成模型的训练，实现图像识别。二是利用边缘计算技术，将用于图像识别的深度学习模型部署在多接入边缘计算（MEC）节点，由MEC节点负责模型的训练。

然而，第1种方法存在数据传输时延大的问题，大量图像数据经广域网传至远端云数据中心，使深度学习模型训练耗时多。第2种方法虽能减少图像数据从

收稿日期：2025-10-16

移动设备传至MEC节点的时延，但MEC节点计算与存储能力有限，训练高精度深度学习模型性能不佳。此外，因MEC节点覆盖范围有限，移动场景下模型训练需考虑任务迁移问题。因此，协同MEC节点及移动设备资源进行任务最优卸载处理成为关注焦点，目前主要有以下2种方法。

a）基于能耗敏感的任务最优卸载方案。该方案以降低能耗为目标，选择最优的MEC节点进行任务卸载[7-9]，如文献[7]提出基于Lyapunov优化的在线能耗优化算法，可满足动态任务下的能耗最优。

b）基于时延敏感的任务最优卸载方法。该方法旨在最小化用户任务处理的时延[10-12]。文献[10]提出启发式算法来求解混合整数线性规划（MILP）模型；而在文献[11]提出迭代算法，将问题分成3个子问题，以交替优化方法结合内凸逼近框架的方式解决该问题，从而降低时延。

此外，智能手机、无人机等移动端设备受其能量供给的限制，在进行深度学习模型训练时，能耗也是关键的考虑因素。为解决深度学习模型训练方法的问题，本文结合边缘计算技术与云计算技术，构建移动场景下端—边—云协同的深度学习模型训练加速系统架构。通过引入模型分割技术[13]，将端—边—云协同训练建模为在终端能耗、训练时延受限条件下，基于训练时延最小的任务最优卸载问题，实现训练加速。本文的主要贡献有以下3点。

a）首次分4类场景提出高速移动场景下深度学习模型训练的业务流程，在终端能耗、模型训练时延受限的情况下，综合考虑通信、计算等因素，联合优化任务卸载和任务迁移，设计基于时延最优的深度学习模型训练加速方案，并给出了最优的模型分割的切割点，实现模型训练加速。

b）设计了麻雀优化算法，解决了受限条件下最优解的问题。

c）对模型进行了实验，结果显示在资源受限情况下，本文设计的基于端—边—云协同模型训练时延优化方案相比端—云协同方案、端—边协同方案及端—边—云随机方案，模型训练时延分别降低45%、20%和50%以上。

1 组网架构及业务流程

深度学习模型（如卷积神经网络）由多个神经网络层（卷积层、池化层、全连接层等）叠加而成。因资源有限的设备难以直接训练复杂参数的神经网络，且不同神经网络层的计算资源需求和输出数据量差异显著，所以可根据设备计算能力和网络带宽分割整个深度学习模型。让设备仅处理从第1层开始的连续网络层计算任务，以提高资源利用效率。基于此，可将模型训练切分为3个部分，终端完成一部分任务，同时利用MEC节点和云节点资源优势，将剩余2个部分分别卸载给它们进行训练。

移动场景端—边—云组网模式下模型训练架构如图1所示，假定4组MEC服务器、1个云服器为移动用户提供服务，移动用户利用采集的图像训练深度学习模型进而实现图像识别。假定深度学习模型训练任务量为W，因本地计算能力有限，将训练任务量W进行切分，子任务将同时在本地终端设备、MEC或/和云服务器上进行训练处理。

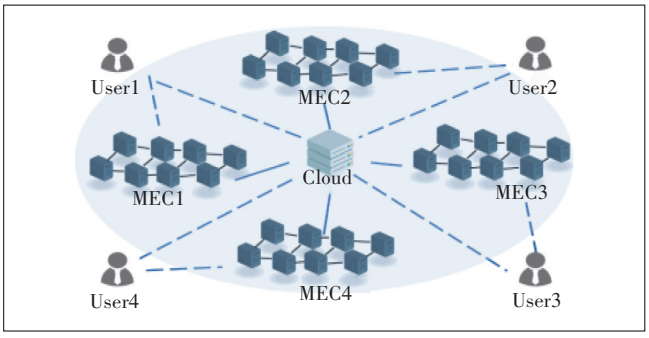

图1 端—边—云架构组网

在移动通信中，若移动用户与MEC连接时长未达到预设标准，即未完成任务就离开该MEC的覆盖范围，为保证服务连续和数据完整，需将未完成的任务数据从当前MEC节点（MEC_i，卸载MEC）迁移到下一个可用MEC服务器继续处理，此过程称为“迁移”[14]，执行迁移任务的MEC节点为MEC_j（迁移MEC）。根据迁移时间不同，端—边—云架构下深度学习模型训练处理流程分4种场景（见图2），场景1是模型任务未卸载完触发MEC切换，场景2是模型任务未训练完触发MEC切换，场景3是训练完成的任务模型未回传完触发MEC切换，场景4是模型任务训练中未触发MEC切换。其中，X_i是任务量W切分后计划卸载到MEC服务器的训练任务量，X_c是计划卸载到云服务器的训练任务量，因X_c无迁移现象，故按X_i迁移情况划分为4种场景。图2中所有①a、①b步骤在时间上并行。

图2（a）展示了场景1中移动用户的行为及其与MEC节点的交互。用户首先上传了训练任务量B至

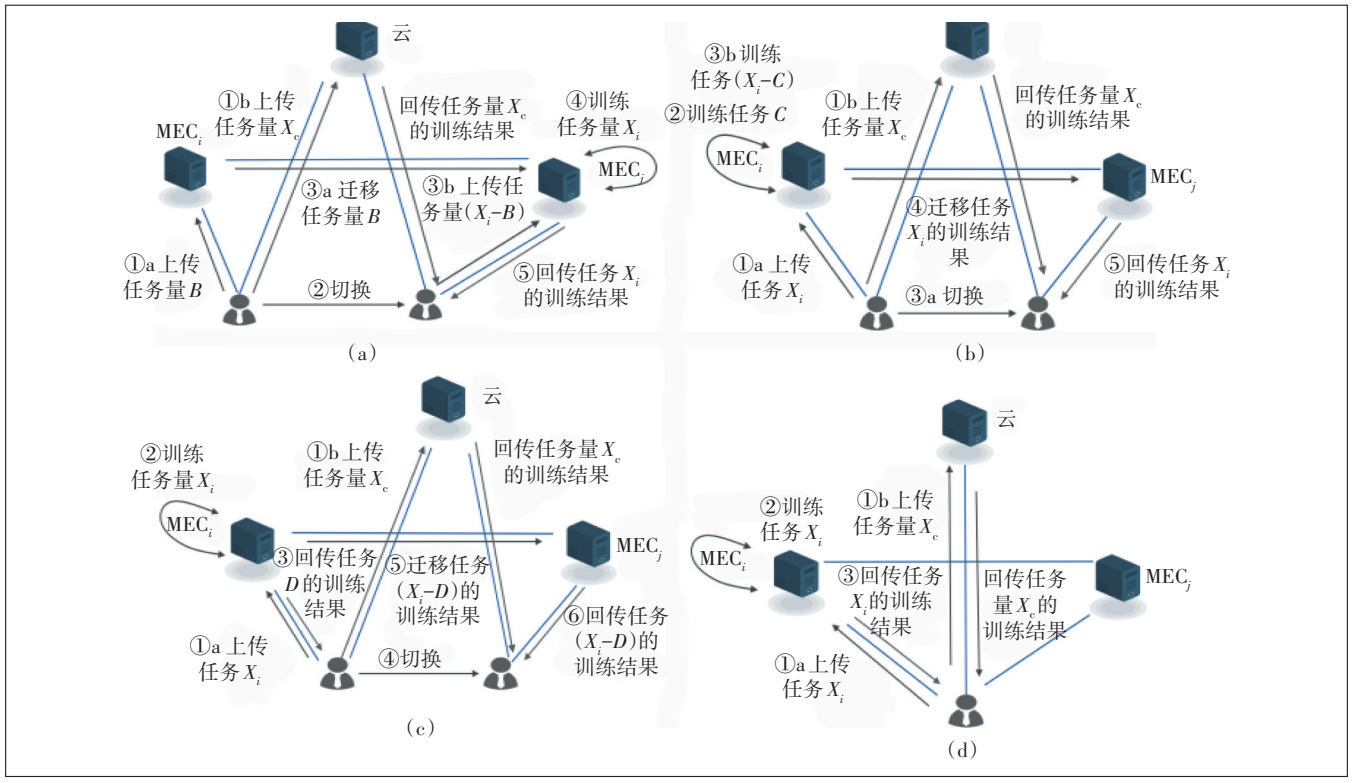

图2 端—边—云架构下深度学习模型训练处理流程

MEC_i。因用户的高速移动，其服务从 MEC_i 转移至 MEC_j。此时，用户需将剩余任务量 (X_i-B) 上传至新服务节点 MEC_j，同时 MEC_i 将已接收的任务量 B 迁移给 MEC_j 继续训练。最终，训练结果由 MEC_j 回传给用户。需强调的是，步骤③a和③b在此场景下是并行进行的。场景2如图2(b)所示，用户上传训练任务量 X_i 并在 MEC_i 完成 C 量的训练后，因移动至 MEC_j 服务区域，MEC_i 将剩余未完成的任务量 (X_i-C) 及其训练结果迁移至 MEC_j。此后，由 MEC_j 负责将训练结果回传给用户。同样，此场景下的步骤③a和③b是并行的。图2(c)描述了场景3的处理过程。在此场景中，用户完成训练任务量 X_i 的上传，并在 MEC_i 完成训练后收到 D 量的训练结果。但因用户移动至 MEC_j 服务区域，MEC_i 需将未回传的训练结果 (X_i-D) 迁移给 MEC_j。最终，由 MEC_j 负责将剩余训练结果回传给用户。图2(d)描绘了场景4的情况，即在整个训练任务上传、训练及结果回传过程中，未发生任何MEC节点间的切换。

2 基于“端—边—云”协同模型训练任务处理模型

为了更有效地降低深度学习模型训练任务的时间延迟，需考虑云节点、多接入边缘计算(MEC)节点以及移动用户终端间的通信和计算资源的协同作用。本章将依据训练任务处理流程，构建训练任务时延和能耗的模型，以优化训练效率。

2.1 系统模型

2.1.1 训练任务时延

2.1.1.1 训练任务通信时延

基于上述业务流程，考虑到MEC节点可能会发生迁移的情况，训练任务通信时延的计算过程如下。

移动用户与 MEC_i 之间任务传输所需的通信时延 $T_i^{\mathrm{com}}\left(x_i,T_i^{\mathrm{ct}}\right)$ 的公式及说明参见文献[15]。

$$T_i^{\mathrm{com}}\left(x_i,T_i^{\mathrm{ct}}\right)=\begin{cases}T_i^{\mathrm{ct}}, & 0<T_i^{\mathrm{ct}}<T_i^{\mathrm{rv}}\left(x_i\right)\\ T_i^{\mathrm{rv}}\left(x_i\right), & T_i^{\mathrm{rv}}\left(x_i\right)<T_i^{\mathrm{ct}}<\left[T_i^{\mathrm{rv}}\left(x_i\right)+T_i^{\mathrm{pc}}\left(x_i\right)\right]\\ T_i^{\mathrm{ct}}-T_i^{\mathrm{pc}}\left(x_i\right), & \left[T_i^{\mathrm{rv}}\left(x_i\right)+T_i^{\mathrm{pc}}\left(x_i\right)\right]<T_i^{\mathrm{ct}}<T_i^{\mathrm{total}}\left(x_i\right)\\ T_i^{\mathrm{rv}}\left(x_i\right)+T_i^{\mathrm{sd}}\left(x_i\right), & \text{其他}\end{cases}\tag{1}$$

其中，x_i 为深度学习模型切割后，用户计划卸载到 MEC_i 的训练任务量；T_i^{ct} 为移动用户与 MEC_i 的连接时

间；$T_i^{\mathrm{rv}}\left(x_i\right)$为训练任务量$x_i$卸载到$\mathrm{MEC}_i$的时间；$T_i^{\mathrm{pc}}\left(x_i\right)$为$\mathrm{MEC}_i$训练任务量$x_i$的时间；$T_i^{\mathrm{sd}}\left(x_i\right)$为回传训练结果的时间；$T_i^{\mathrm{total}}\left(x_i\right)$为$T_i^{\mathrm{rv}}\left(x_i\right)$、$T_i^{\mathrm{pc}}\left(x_i\right)$、$T_i^{\mathrm{sd}}\left(x_i\right)$之和。

移动用户与MEC_j之间任务传输所需的通信时延$T_j^{\mathrm{com}}\left(x_i,T_i^{\mathrm{ct}}\right)$的公式及说明参见文献[15]。

$$T_j^{\mathrm{cm}}\left(x_i,T_i^{\mathrm{ct}}\right)=\begin{cases}\dfrac{x_i-V_{Ri}^{\mathrm{u}}T_i^{\mathrm{ct}}}{V_{Rj}^{\mathrm{u}}}+\dfrac{\rho x_i}{V_{Rj}^{\mathrm{d}}},\ 0<T_i^{\mathrm{ct}}<T_i^{\mathrm{rv}}\left(x_i\right)\\ \dfrac{\rho x_i}{V_{Rj}^{\mathrm{d}}},\ T_i^{\mathrm{rv}}\left(x_i\right)<T_i^{\mathrm{ct}}<\left[T_i^{\mathrm{rv}}\left(x_i\right)+T_i^{\mathrm{pc}}\left(x_i\right)\right]\\ \dfrac{\rho x_i-V_{Ri}^{\mathrm{d}}\left[T_i^{\mathrm{ct}}-T_i^{\mathrm{rv}}\left(x_i\right)-T_i^{\mathrm{pc}}\left(x_i\right)\right]}{V_{Rj}^{\mathrm{d}}},\\ \left[T_i^{\mathrm{rv}}\left(x_i\right)+T_i^{\mathrm{pc}}\left(x_i\right)\right]<T_i^{\mathrm{ct}}<T_i^{\mathrm{total}}\left(x_i\right)\\ 0,\ \text{其他}\end{cases}\tag{2}$$

其中，V_{Ri}^{u}、V_{Ri}^{d}分别为用户与MEC_i的上下行速率；V_{Rj}^{u}、V_{Rj}^{d}分别为用户与MEC_j的上下行速率；ρx_i为x_i训练结果。

移动用户与云节点之间任务传输所需通信时延如式(3)所示。

$$T_c^{\mathrm{com}}\left(x_c\right)=\frac{x_c}{V_{Rc}^{\mathrm{u}}}+\frac{\rho x_c}{V_{Rc}^{\mathrm{d}}}\tag{3}$$

其中，V_{Rc}^{u}为移动用户与云节点的上行链路传输速率；V_{Rc}^{d}为移动用户与云节点的下行链路传输速率；x_c为深度学习模型切割后，移动用户计划卸载到云节点的训练任务量；ρx_c为模型在云节点训练结果大小。

2.1.1.2 任务训练时延

MEC_i训练时延如式(4)所示。

$$T_i^{\mathrm{comp}}\left(x_i,T_i^{\mathrm{ct}}\right)=\begin{cases}0,\ 0<T_i^{\mathrm{ct}}<T_i^{\mathrm{recv}}\left(x_i\right)\\ T_i^{\mathrm{proc}}\left(x_i\right),\ \text{其他}\end{cases}\tag{4}$$

其中，$T_i^{\mathrm{recv}}\left(x_i\right)$为训练任务量$x_i$卸载到$\mathrm{MEC}_i$的传输时间，$T_i^{\mathrm{recv}}\left(x_i\right)=\dfrac{x_i}{V_{Ri}^{\mathrm{u}}}$，$V_{\mathrm{Ri}}^{\mathrm{u}}$为$\mathrm{MEC}_i$上行链路传输速率；$T_i^{\mathrm{proc}}\left(x_i\right)$为训练任务量$x_i$在$\mathrm{MEC}_i$的训练时间，$T_i^{\mathrm{proc}}\left(x_i\right)=\dfrac{x_i}{V_{Ci}}$，$V_{Ci}$为$\mathrm{MEC}_i$的训练速率。

MEC_j训练时延如式(5)所示。

$$T_j^{\mathrm{comp}}\left(x_i,T_i^{\mathrm{ct}}\right)=\begin{cases}\dfrac{x_i}{V_{Cj}},\ 0<T_i^{\mathrm{ct}}<T_i^{\mathrm{recv}}\left(x_i\right)\\ 0,\ \text{其他}\end{cases}\tag{5}$$

其中，V_{Cj}为MEC_j的训练速率。

移动用户终端训练时延如式(6)所示。

$$T_l^{\mathrm{comp}}\left(x_i,x_c\right)=\frac{\left(W-x_i-x_c\right)}{V_{\mathrm{Cl}}}\tag{6}$$

其中，W为训练任务总量，V_{Cl}为移动用户终端训练速率。

云节点训练时延如式(7)所示。

$$T_c^{\mathrm{comp}}\left(x_c\right)=\frac{x_c}{V_{Cc}}\tag{7}$$

其中，V_{Cc}为云节点训练速率。

2.1.1.3 训练任务迁移时延

由于移动用户与MEC_i的连接时间T_i^{ct}是随机的，当连接时间不足以完成训练任务传输或者任务训练时都会发生任务迁移，那么迁移时延可表示为式(8)。

$$T_{ij}^{\mathrm{mig}}\left(x_i,T_i^{\mathrm{ct}}\right)=T_{\mathrm{HO}}+\begin{cases}\dfrac{V_{Ri}^{\mathrm{u}}T_i^{\mathrm{ct}}}{R_{i,j}},\ 0<T_i^{\mathrm{ct}}<T_i^{\mathrm{recv}}\left(x_i\right)\\ \dfrac{\rho x_i}{R_{i,j}},\ T_i^{\mathrm{recv}}\left(x_i\right)<T_i^{\mathrm{ct}}<\left[T_i^{\mathrm{recv}}\left(x_i\right)+T_i^{\mathrm{proc}}\left(x_i\right)\right]\\ \dfrac{\rho x_i-V_{Ri}^{d}\left[T_i^{\mathrm{ct}}-T_i^{\mathrm{recv}}\left(x_i\right)-T_i^{\mathrm{proc}}\left(x_i\right)\right]}{R_{i,j}},\\ \left[T_i^{\mathrm{recv}}\left(x_i\right)+T_i^{\mathrm{proc}}\left(x_i\right)\right]<T_i^{\mathrm{ct}}<T_i^{\mathrm{total}}\left(x_i\right)\\ 0,\ \text{其他}\end{cases}\tag{8}$$

其中，$R_{i,j}$为MEC_i与MEC_j之间的传输速率；T_{HO}为移动用户切换时延；$T_i^{\mathrm{total}}\left(x_i\right)$为$\mathrm{MEC}_i$训练任务量$x_i$所需时间，$T_i^{\mathrm{total}}\left(x_i\right)=T_i^{\mathrm{recv}}\left(x_i\right)+T_i^{\mathrm{proc}}\left(x_i\right)+T_i^{\mathrm{send}}\left(x_i\right)$；$T_i^{\mathrm{send}}\left(x_i\right)$为训练结果$\rho x_i$从$\mathrm{MEC}_i$传输到移动用户的时间，$T_i^{\mathrm{send}}\left(x_i\right)=\dfrac{\rho x_i}{V_{Ri}^{\mathrm{d}}}$；$\rho x_i$为模型训练结果大小；$V_{Ri}^{\mathrm{d}}$为$\mathrm{MEC}_i$下行链路传输速率。

训练任务时延由上述通信时延、训练实验和迁移时延计算获得，具体如式(9)所示。

$$T\left(x_i,T_i^{\mathrm{ct}},x_c\right)=\begin{cases}T1\left(x_i,T_i^{\mathrm{ct}},x_c\right),\ 0<T_i^{\mathrm{ct}}<T_i^{\mathrm{recv}}\left(x_i\right)\\ T2\left(x_i,T_i^{\mathrm{ct}},x_c\right),\ T_i^{\mathrm{recv}}\left(x_i\right)<T_i^{\mathrm{ct}}<\left[T_i^{\mathrm{recv}}\left(x_i\right)+T_i^{\mathrm{proc}}\left(x_i\right)\right]\\ T3\left(x_i,T_i^{\mathrm{ct}},x_c\right),\ \text{其他}\end{cases}\tag{9}$$

其中，$T1$为$0<T_i^{\mathrm{ct}}<T_i^{\mathrm{recv}}\left(x_i\right)$时的训练任务时延，$T2$为$T_i^{\mathrm{recv}}\left(x_i\right)<T_i^{\mathrm{ct}}<\left[T_i^{\mathrm{recv}}\left(x_i\right)+T_i^{\mathrm{proc}}\left(x_i\right)\right]$时的训练任务

时延，$T3$为$T_i^{\mathrm{ct}} > \left[T_i^{\mathrm{recv}}\left(x_i\right) + T_i^{\mathrm{proc}}\left(x_i\right)\right]$时的训练任务时延。

具体地：

$$T1\left(x_i,T_i^{\mathrm{ct}},x_c\right) = \max\left\{\left[T_i^{\mathrm{com}}\left(x_i,T_i^{\mathrm{ct}}\right) + T_{\mathrm{HO}} + \max\left[\frac{V_{Ri}^{\mathrm{u}}T_i^{\mathrm{ct}}}{R_{i,j}},\frac{x_i - V_{Ri}^{\mathrm{u}}T_i^{\mathrm{ct}}}{V_{Rj}^{\mathrm{u}}}\right] + T_i^{\mathrm{comp}}\left(x_i,T_i^{\mathrm{ct}}\right) + T_j^{\mathrm{comp}}\left(x_i,T_i^{\mathrm{ct}}\right) + \frac{\rho x_i}{V_{Rj}^{\mathrm{d}}}\right],T_l^{\mathrm{comp}}\left(x_i,x_c\right),\left[T_c^{\mathrm{com}}\left(x_c\right) + T_c^{\mathrm{comp}}\left(x_c\right)\right]\right\} \quad (10)$$

其中，V_{Rj}^{u}为用户与MEC_j的上行链路传输速率；V_{Rj}^{d}为用户与MEC_j的下行链路传输速率。

$$T2\left(x_i,T_i^{\mathrm{ct}},x_c\right) = \max\left\{\left[T_i^{\mathrm{com}}\left(x_i,T_i^{\mathrm{ct}}\right) + T_j^{\mathrm{com}}\left(x_i,T_i^{\mathrm{ct}}\right) + T_{ij}^{\mathrm{mig}}\left(x_i,T_i^{\mathrm{ct}}\right) + T_i^{\mathrm{ct}} - T_i^{\mathrm{recv}}\left(x_i\right) + T_j^{\mathrm{comp}}\left(x_i,T_i^{\mathrm{ct}}\right)\right],T_l^{\mathrm{comp}}\left(x_i,x_c\right),\left[T_c^{\mathrm{com}}\left(x_c\right) + T_c^{\mathrm{comp}}\left(x_c\right)\right]\right\} \quad (11)$$

$$T3\left(x_i,T_i^{\mathrm{ct}},x_c\right) = \max\left\{\left[T_i^{\mathrm{com}}\left(x_i,T_i^{\mathrm{ct}}\right) + T_j^{\mathrm{com}}\left(x_i,T_i^{\mathrm{ct}}\right) + T_i^{\mathrm{comp}}\left(x_i,T_i^{\mathrm{ct}}\right) + T_j^{\mathrm{comp}}\left(x_i,T_i^{\mathrm{ct}}\right) + T_{ij}^{\mathrm{mig}}\left(x_i,T_i^{\mathrm{ct}}\right)\right],T_l^{\mathrm{comp}}\left(x_i,x_c\right),\left[T_c^{\mathrm{com}}\left(x_c\right) + T_c^{\mathrm{comp}}\left(x_c\right)\right]\right\} \quad (12)$$

2.1.2 训练任务终端能耗

移动用户终端训练任务能耗为：

$$E_i^{\mathrm{local}}\left(x_i,x_c\right) = \left(W - x_i - x_c\right)P_{\mathrm{Cl}} \quad (13)$$

其中，P_{Cl}为移动用户终端进行模型训练的单bit能耗。

训练任务传输能耗为：

$$E_{i,j}^{\mathrm{tran}}\left(x_i,T_i^{\mathrm{ct}},x_c\right) = P_{\mathrm{Rl}}\left(x_i + x_c + \rho x_i + \rho x_c\right) \quad (14)$$

其中，P_{Rl}为移动用户终端发射/接收单bit能耗。

根据移动用户终端训练能耗和训练任务传输能耗得到终端侧的能耗为：

$$E_{\mathrm{l}}\left(x_i,T_i^{\mathrm{ct}},x_c\right) = E_i^{\mathrm{local}}\left(x_i,x_c\right) + E_{i,j}^{\mathrm{tran}}\left(x_i,T_i^{\mathrm{ct}},x_c\right) \quad (15)$$

2.2 目标函数

为了最小化模型训练任务时延，并且同时考虑模型训练任务时延、终端能耗限制，提出的优化问题可以表示为：

$$\min_{x_i,x_c} T\left(x_i,T_i^{\mathrm{ct}},x_c\right) = \begin{cases} T1\left(x_i,T_i^{\mathrm{ct}},x_c\right), & 0 < T_i^{\mathrm{ct}} < T_i^{\mathrm{rv}}\left(x_i\right) \\ T2\left(x_i,T_i^{\mathrm{ct}},x_c\right), & T_i^{\mathrm{rv}}\left(x_i\right) < T_i^{\mathrm{ct}} < \left[T_i^{\mathrm{rv}}\left(x_i\right) + T_i^{\mathrm{pc}}\left(x_i\right)\right] \\ T3\left(x_i,T_i^{\mathrm{ct}},x_c\right), & \text{其他} \end{cases} \quad (16)$$

s.t.

$$D_m \geqslant \begin{cases} T1\left(x_i,T_i^{\mathrm{ct}},x_c\right), & 0 < T_i^{\mathrm{ct}} < T_i^{\mathrm{rv}}\left(x_i\right) \\ T2\left(x_i,T_i^{\mathrm{ct}},x_c\right), & T_i^{\mathrm{rv}}\left(x_i\right) < T_i^{\mathrm{ct}} < \left[T_i^{\mathrm{rv}}\left(x_i\right) + T_i^{\mathrm{pc}}\left(x_i\right)\right] \\ T3\left(x_i,T_i^{\mathrm{ct}},x_c\right), & \text{其他} \end{cases} \quad (17)$$

$$E_l\left(x_i,T_i^{\mathrm{ct}},x_c\right) < E_{\mathrm{UE}}^{\mathrm{th}} \quad (18)$$

$$0 < x_i < W \quad (19)$$

$$0 < x_c < W \quad (20)$$

$$0 < \left(x_c + x_i\right) < W \quad (21)$$

其中，D_m为模型训练任务完成期限，$E_{\mathrm{UE}}^{\mathrm{th}}$为移动终端能量阈值。

3 基于麻雀的觅食行为和反捕食行为的启发式优化算法

麻雀优化算法（Sparrow Search Algorithm，SSA）由东华大学薛建凯于2020年提出，是模仿麻雀觅食和反捕食行为衍生的新型群体智能优化算法。在SSA体系中，麻雀族群分为探索者、追随者和警戒者3类。探索者能量储备高，负责搜寻食物丰富区域并为追随者提供指引；追随者紧随探索者觅食；警戒者监控环境，发现威胁则鸣叫报警，警报超阈值时，探索者引导追随者转移。该算法逐步优化当前解以搜索最优解，在探索局部最优解上表现出色。在边缘计算领域，局部搜索对卸载决策意义重大，该算法思路简洁、易掌握，应用前景广阔，可快速解决卸载策略问题，在解空间较小问题上搜索高效。鉴于边缘计算资源有限，高效算法对卸载决策至关重要，所以，麻雀搜索算法适用于求解最优任务卸载策略，具体算法如图3所示。

4 仿真与分析

经过上述模型与算法的运算，本文对深度学习模型在图1场景下的训练性能进行仿真实验。为简化分析，设区域1中MEC1、MEC2、MEC3承担用户服务任务，区域2中MEC4、MEC5、MEC6执行相应服务。MEC1通信与计算性能最佳但能耗最高；MEC2次之，MEC3相对较低。区域2的MEC4、MEC5、MEC6的性能分别与区域1的MEC1、MEC2、MEC3一致。其中，MEC1/4的通信速率为15 Mbit/s，计算速率为13.4 Mbit/s；MEC2/5通信速率为13 Mbit/s，计算速率为11.6 Mbit/s；MEC3/6通信速率为12.5 Mbit/s，计算速率为11.2 Mbit/s。在此次仿真中，设定任务完成的截止时

Algorithm：麻雀优化算法

Input：
G：最大迭代次数
N：种群数目
Ub：最大任务卸载量
Lb：最小任务卸载量
S：安全阀值
DS：发现者数量
FL：追随者数量
CT：警戒者数量
Output：
Xi_{best}：边节点最优任务卸载量
Xc_{best}：云节点最优任务卸载量
F_{best}：训练模型最优时延
1：调用式(19)~(21)初始化种群位置，根据式(17)和式(18)在时延、能耗限制条件下调用式(9)对应卸载方案的模型训练时延
2：while (t<G)
3：S=rand(1)
4：for i=1：DS
5：　发现者迭代
6：endfor
7：for i=(DS+1)：(DS+FL)
8：　追随者迭代
9：endfor
10：for i=(DS+FL+1)：N
11：　警戒者迭代
12：endfor
13：调用公式(19)~(21)检查迭代位置是否在搜索边界内
14：根据迭代后的任务卸载方案，根据式(17)和式(18)在时延、能耗限制条件下调用式(9)对应卸载方案的模型训练时延
15：更新种群中的最差和最优任务卸载方案
16：根据适应度对种群排序，选择新的发现者、追随者和警戒者
17：t=t+1
18：endwhile
19：return Xi_{best}，Xc_{best}，F_{best}

图3　麻雀优化算法

间为180 s，用户终端的能耗阈值为1 000 J，连接时间为15 s，其他参数参考文献[16]及[17]进行设置。

模型训练时延与训练任务量的关系如图4所示，各方案目标均为实现最低训练时延。具体而言，端—云协同方案利用云节点和用户端资源协同优化训练任务；端—边协同方案利用边节点和用户端资源协同优化；本文提出的端—边—云协同方案，利用云、边节点和用户端资源协同优化；端—边—云随机方案则随机分配任务量给云节点、MEC和用户端进行优化。

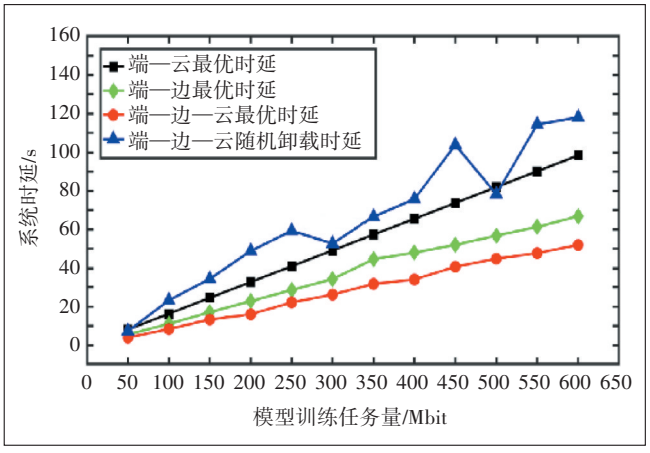

图4　模型训练时延随模型训练任务量变化情况

由图4可知，端—云、端—边、端—边—云协同方案的训练时延随任务量增加而增加，端—边—云随机方案因随机卸载处理，该规律不明显。在相同任务量下，本文提出的端—边—云协同方案平均时延最低，端—边协同方案次之，端—云协同方案再次之，端—边—云随机方案最高，体现了端—边—云协同的优势。该方案相比端—云、端—边协同及端—边—云随机方案，训练时延分别降低45.93%、23.10%和57.01%。

模型训练时延随连接时间变化情况如图5所示，模型训练任务量为350 Mbit。从图5可以看出，在相同的连接时间下，本文提出的端—边—云协同方案的模型训练平均时延最低，端—边协同方案次之，然后是端—云协同方案，端—边—云随机方案最高。与端—云协同方案、端—边协同方案及端—边—云随机方案相比，本文提出的端—边—云协同模型训练时延优化方案分别降低了48.00%、25.24%和54.25%。

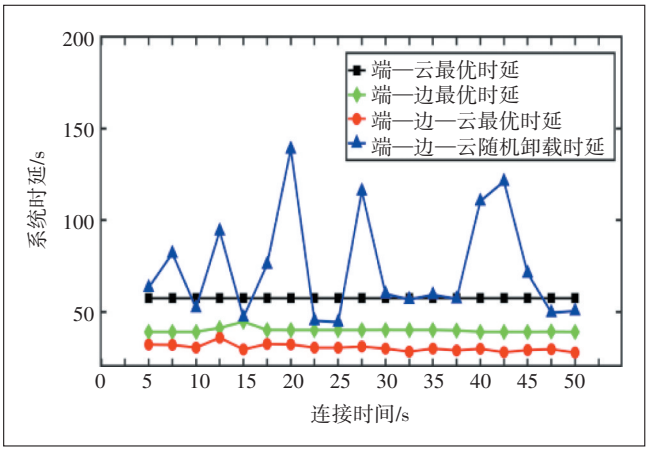

图5　模型训练时延随模型连接时间变化情况

模型训练时延对比如图6所示。待处理的模型训练任务量在[50，350](Mbit)内随机取值，连接时间在[10，15](s)内随机取值。随着用户数的变化，本文提出的端—边—云协同方案的模型训练平均时延最低，端—边协同方案次之，然后是端—云协同方案，端—边—云随机方案最高。本文提出的端—边—云协同的模型训练时延优化方案相比端—云协同方案、端—

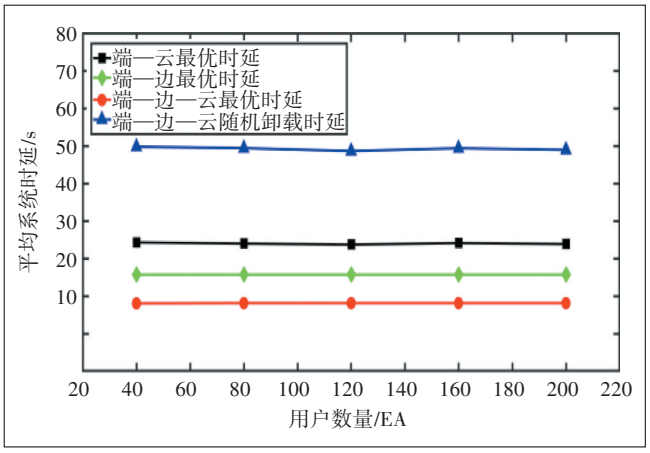

图6　多用户场景下模型训练时延对比

边协同方案及端—边—云随机方案分别降低了66.47%、48.83%和80.97%。

5　结束语

为解决移动场景下深度学习模型训练加速问题，本文先将高速移动场景下模型训练处理流程分为四大类别。考虑到模型训练时延和终端能耗受限，综合考量通信、计算和能耗等因素，构建高速移动场景优化模型以降低处理时延。为提升训练效率，设计端—边—云协同加速方案。此外，针对受限条件下的最优解问题，提出基于麻雀觅食与反捕食行为的启发式优化算法并给出最优卸载策略。仿真结果显示，相同条件下，本文提出的端—边—云协同模型训练时延优化方案，相比端—云协同、端—边协同及端—边—云随机方案，模型训练平均时延分别降低45%、20%和50%以上。

参考文献：

[1] LECUN Y, BENGIO Y, HINTON G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.

[2] SHARIFANI K, AMINNI M. Machine learning and deep learning: a review of methods and applications[J]. World Information Technology and Engineering Journal, 2023, 10(7): 3897-3904.

[3] 朱泓睿，元国军，姚成吉，等. 分布式深度学习训练网络综述[J]. 计算机研究与发展，2021，58(1)：98-115.

[4] REAL E, AGGARWAL A, HUANG Y P, et al. Regularized evolution for image classifier architecture search[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1): 4780-4789.

[5] ZOPH B, VASUDEVAN V, SHLENS J, et al. Learning transferable architectures for scalable image recognition [C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018: 8697-8710.

[6] BROWN T B, MANN B, RYDER N, et al. Language models are few-shot learners[C]//Proceedings of the 34th International Conference on Neural Information Processing System. Vancouver, BC: Curran Associates Inc., 2020: 1877-1901.

[7] TONG Z, CAI J H, MEI J, et al. Dynamic energy-saving offloading strategy guided by lyapunov optimization for IoT devices[J]. IEEE Internet of Things Journal, 2022, 9(20): 19903-19915.

[8] LI Y, ZHANG X, LEI B, et al. Computation rate maximization for wireless-powered edge computing with multi-user cooperation [J]. IEEE Open Journal of the Communications Society, 2024, 5: 965-981.

[9] HUYNH L N T, HOSSAIN M D, PHAM Q V, et al. A block-structured optimization approach for data sensing and computing in vehicle-assisted edge computing networks[J]. IEEE Sensors Journal, 2024, 24(1): 952-961.

[10] WANG X, JI Y F, ZHANG J W, et al. Joint optimization of latency and deployment cost over TDM-PON based MEC-enabled cloud radio access networks[J]. IEEE Access, 2020, 8: 681-696.

[11] VAN HUYNH D, NGUYEN V D, KHOSRAVIRAD S R, et al. uRLLC edge networks with joint optimal user association, task offloading and resource allocation: a digital twin approach[J]. IEEE Transactions on Communications, 2022, 70(11): 7669-7682.

[12] LAI X Z, JIANG H L, BHUNIA S, et al. Reducing latency in MEC networks with short-packet communications[J]. IEEE Transactions on Vehicular Technology, 2024, 73(2): 3000-3004.

[13] 黄煜. 移动边缘计算中基于CNN模型分割的计算适配和负载均衡研究[D]. 北京：北京邮电大学，2023.

[14] MACH P, BECVAR Z. Mobile edge computing: a survey on architecture and computation offloading[J]. IEEE Communications Surveys & Tutorials, 2017, 19(3): 1628-1656.

[15] FENG C F, SUN J D. Mobility-aware task offloading scheme based on optimal system benefit in multi-access edge computing[C]//Signal and Information Processing, Networking and Computers. Singapore: Springer, 2023: 1025-1033.

[16] CHEN M H, LIANG B, DONG M. Joint offloading and resource allocation for computation and communication in mobile cloud with computing access point[C]//IEEE INFOCOM 2017 - IEEE Conference on Computer Communications. Atlanta: IEEE, 2017: 1-9.

[17] PLACHY J, BECVAR Z, STRINATI E C, et al. Dynamic allocation of computing and communication resources in multi-access edge computing for Mobile users[J]. IEEE Transactions on Network and Service Management, 2021, 18(2): 2089-2106.

作者简介：

陈亦哲，山东师范大学研究生在读，主要研究方向为移动通信规划与优化、人工智能等；姜肇瑞，中国海洋大学本科在读，主要研究方向为深度学习、计算生物、机器人学等；冯传奋，毕业于北京邮电大学，教授级高级工程师，博士，主要研究方向为移动通信网络规划、设计优化及“端—边—云”协同系统优化。