Deep Reinforcement Learning-Based Resource Allocation Strategy for Mobile Edge Computing

Authors: Feng Beipeng, Huang Yuzhe, Cao Yuhui, Guo Zhenzhen, Feng Beipeng

Date: 2023-01-17T00:00:00+00:00

Abstract

Cloud computing can address the problem of insufficient computing resources on mobile devices, but fails to meet the service requirements for low latency. Edge computing, as an extension of cloud computing technology, can provide users with low-latency and high-quality services by enhancing the computing capabilities of edge networks. In edge computing, it is necessary to deploy services on resource-constrained edge servers and allocate computing resources reasonably according to demands to improve the resource utilization of edge servers. To this end, this paper proposes a service resource allocation method based on deep reinforcement learning, which establishes a computing resource allocation function through dual mapping of the arctangent function and achieves dynamic adjustment of allocation proportions. Finally, simulation experiments are conducted based on real datasets, and the experimental results demonstrate that the proposed method can reasonably allocate computing resources while guaranteeing low latency. Cloud computing can address the problem of insufficient computing resources on mobile devices, but fails to meet the service requirements for low latency. Edge computing, as an extension of cloud computing technology, can provide users with low-latency and high-quality services by enhancing the computing capabilities of edge networks. In edge computing, it is necessary to deploy services on resource-constrained edge servers and allocate computing resources reasonably according to demands to improve the resource utilization of edge servers. To this end, this paper proposes a service resource allocation method based on deep reinforcement learning, which establishes a computing resource allocation function through dual mapping of the arctangent function and achieves dynamic adjustment of allocation proportions. Finally, simulation experiments are conducted based on real datasets, and the experimental results demonstrate that the proposed method can reasonably allocate computing resources while guaranteeing low latency.

Full Text

Strategy of Resource Allocation Based on Deep Reinforcement Learning in Mobile Edge Computing

Beipeng Feng, Yuze Huang*, Yuhui Cao, Zhenzhen Guo
School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing 400074, China

Abstract

Cloud computing can address the problem of insufficient computing resources on mobile devices but fails to meet stringent low-latency service requirements. As an extension of cloud computing technology, edge computing can provide users with low-latency, high-quality services by enhancing the computing capabilities of edge networks. In edge computing environments, services must be deployed on resource-constrained edge servers, and computing resources must be allocated reasonably according to demand to improve resource utilization. To this end, this paper proposes a service resource allocation method based on deep reinforcement learning that establishes a computing resource allocation function through dual arctangent function mapping and enables dynamic adjustment of allocation ratios. Simulation experiments based on real datasets demonstrate that the proposed method can allocate computing resources reasonably while guaranteeing low latency.

Keywords: Mobile edge computing; Deep reinforcement learning; Resource allocation
Classification: TP311.5

1 Introduction

We live in an era of ubiquitous connectivity where diverse mobile devices have joined the Internet via high-speed 5G networks, causing data on the network to grow exponentially and posing tremendous challenges to network load. Meanwhile, the development and deployment of latency-sensitive applications continue to impose new demands on hardware resources. In this context, mobile edge computing has emerged as a new computing paradigm that provides computational support to radio access networks near users, enabling the popularization of more applications. However, it should be noted that edge servers, as the primary processing units in edge computing, also face severe resource scarcity. Particularly due to their distributed nature, a single edge server cannot host multiple services simultaneously. Consequently, when user requests arrive at the edge network, they are typically processed by servers that have deployed the requested service. The amount of resources allocated by each server to its hosted services directly affects service response latency.

The optimization objectives of service resource allocation typically focus on re-

ducing energy consumption and average response latency. Previous research has proposed an energy-saving radio resource allocation offloading scheme for 5G heterogeneous networks that adapts to complex and dynamic network environments and demonstrates good performance in energy consumption based on a single-item data transmission energy model. However, this approach provides insufficient optimization for service latency, resulting in poor user experience. Other studies have focused on user experience by optimizing service response latency under resource constraints, but these algorithms suffer from high complexity and stringent application scenario requirements, making them unsuitable for dynamic mobile edge computing environments. Global optimization methods for cloud-edge-terminal three-tier architectures are often constrained by multiple factors, leading to suboptimal service response latency satisfaction. Among these approaches, deep reinforcement learning algorithms, with their human-like reasoning patterns where agents learn optimal policies through continuous trial-and-error interaction with the environment, offer an effective solution for resource allocation problems. These algorithms not only leverage their strength in solving global optimization problems but also adapt to different mobile edge computing environments.

Therefore, to address the challenge of limited network resources in mobile edge computing while satisfying user experience requirements, this paper proposes a deep reinforcement learning-based computing resource allocation method. This approach obtains feedback after resource allocation and dynamically adjusts the next allocation ratio through dual arctangent function mapping to optimize the average service response latency.

2.1 Edge Server and Service Model

As shown in [Figure 1: see original paper], the mobile edge computing scenario consists of an edge network formed by edge servers that provide stable computing services to end users, with cloud servers acting as resource scheduling centers that distribute DQN (Deep Q-Learning) trained resource allocation decisions to each edge server.

The set of edge servers can be represented by a quadruple parameter: $\langle MATH_0 \rangle$, where the four variables respectively represent the computing resources of edge server x, whether service y is deployed, the proportion of computing resources allocated to service y by edge server x at time slot t, and the number of service requests y received at time slot t. The variable $\langle MATH_1 \rangle$ is either 0 or 1, where 1 indicates that edge server x has deployed service y and 0 otherwise.

Service y is represented by a binary parameter: $\langle MATH_2 \rangle$, where $\langle MATH_3 \rangle$ and $\langle MATH_4 \rangle$ denote the data forwarding volume and computing resources required for processing service y requests, respectively. Since edge server computing resources are limited, the computing resource allocation proportion for any edge server x cannot exceed its total capacity: $\langle MATH_5 \rangle$.

2.2 Service Processing Delay Model

Since the data volume of service processing results is typically small, the delays involved in service processing mainly include request forwarding delay and edge server computing delay. When a terminal user sends a service request y to edge server x, if the edge server has deployed service y, it processes the request and returns the result. This process includes data transmission delay between the terminal and edge server and the edge server' s computing delay, which can be expressed as: $\langle MATH_6 \rangle$, where $\langle MATH_7 \rangle$ represents the number of service requests y received by edge server x at time slot t, and $\langle MATH_8 \rangle$ is the data transmission rate.

When edge server x receives a service request y from a terminal user, if the receiving server has not deployed the service, it must forward the request to other edge servers that have deployed the service. Therefore, the number of service requests y that need to be forwarded at time slot t can be expressed as: $\langle MATH_9 \rangle$. The data transmission delay caused by service requests can then be expressed as: $\langle MATH_{10} \rangle$. Consequently, the total delay for processing service request y is: $\langle MATH_{11} \rangle$.

Our objective is to reduce the average service response latency through computing resource allocation, which can be formulated as: $\langle MATH_{12} \rangle$, where equation (9) constrains the amount of allocatable computing resources. To solve this optimization problem, we employ a deep reinforcement learning algorithm for training.

2.3 Deep Reinforcement Learning-Based Service Computing Resource Allocation Algorithm

Deep reinforcement learning modeling is typically based on Markov Decision Processes. In this scenario, the key elements include environment, state, action, and reward. Therefore, we first construct the state space S, action space A, and reward function R.

The action space primarily consists of the computing resource allocation proportions that each edge server assigns to its corresponding services. In our algorithm, the amount of computing resources allocated by edge server x to service y is not determined at once but is obtained through iterative small-scale allocations. Notably, the algorithm dynamically adjusts the magnitude of the next allocation based on feedback after each allocation. Therefore, at time slot t, the action space for edge server x can be expressed as: $\langle MATH_{13} \rangle$.

We aim to dynamically adjust each allocation proportion based on the quality of the previous action. If the previous action significantly reduced service response latency, we increase the computing resources allocated to that service in the next step. This is implemented as follows: $\langle MATH_{14} \rangle$ represents the average service response latency in state S_t, and $\langle MATH_{15} \rangle$ represents the average service response latency after taking action A_t in state S_t to obtain

state S_{t+1}. Equation (12) represents the difference in average response latency between time slot t and $t+1$. A larger $\langle MATH_{16}\rangle$ indicates that action A_t achieves greater latency reduction, necessitating an amplification of the next allocation proportion. Conversely, if $\langle MATH_{17}\rangle$ is small, the next allocation proportion should be reduced. We implement this through equation (13), where $\langle MATH_{18}\rangle$ denotes the initial computing resource allocation proportion. This equation, through dual arctangent function mapping, constrains the next allocation proportion within $\langle MATH_{19}\rangle$.

Regarding the reward function design, since the reward value correlates with the value of the executed action, we assign a negative reward when the action fails to reduce the average service response latency and a positive reward otherwise. The reward function can be expressed as: $\langle MATH_{20}\rangle$, where α is a constant.

Since our objective is to minimize the average service response latency, which translates to maximizing the reward value in the algorithm, each step' s action must be valuable. Therefore, the action value function for time step t can be expressed as the sum of future rewards: $\langle MATH_{21}\rangle$, where γ is the discount factor and $\langle MATH_{22}\rangle$ represents the expected value for the next T time steps. This formulation transforms the problem of minimizing average service response latency into maximizing the action value: $\langle MATH_{23}\rangle$.

Furthermore, to ensure that the average service response latency converges to the optimal value as training progresses, the loss function is set as: $\langle MATH_{24}\rangle$, where $\langle MATH_{25}\rangle$ represents the state obtained after taking action a in state s, D denotes the experience replay buffer storing training metadata after each action, and θ represents the neural network weights updated via gradient descent: $\langle MATH_{26}\rangle$, where η is the learning rate. After a certain number of steps, the target network parameters $\langle MATH_{27}\rangle$ are updated.

3.1 Experimental Parameters and Results Analysis

The simulation parameters are listed in . To validate the effectiveness of our algorithm, we compare the DQN algorithm against three baseline methods: a resource allocation method based on service request quantity, a resource allocation method based on greedy strategy, and a random resource allocation method.

[Figure 2: see original paper] illustrates the convergence comparison of different algorithms. As shown, as the number of DQN algorithm iterations increases, the training performance improves progressively, eventually converging to the optimal value before 600 iterations. The average service response latency decreases from 1.6 at the beginning of training to less than 1. Notably, the convergence value of the DQN algorithm is superior to other baseline algorithms, demonstrating that deep reinforcement learning achieves more efficient resource utilization.

This paper investigates the computing resource allocation problem in mobile edge computing environments and proposes a deep reinforcement learning-based

resource allocation method. The approach designs the computing resource allocation magnitude for each action step in deep reinforcement learning to optimize the average service response latency. Experiments demonstrate that the method can fully utilize computing resources and achieve lower service response latency.

References

[1] Zhang J, Chen B, Zhao Y, et al. Data security and privacy-preserving in edge computing paradigm: Survey and open issues[J]. IEEE access, 2018, 6: 18209-18237.
[2] 黄永明, 郑冲, 张征明等. 大规模无线通信网络移动边缘计算和缓存研究 [J]. 通信学报, 2021, 42(4): 44-61.
[3] Morabito I G. The internet of things: A survey[J]. Computer Networks, 2010.
[4] Pham Q. V., FangF, Ha V. N., et al. A Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art. IEEE Access, 2020,8:116974-117017.
[5] 梁广俊, 王群, 辛建芳, 李梦, 许威. 移动边缘计算资源分配综述 [J]. 信息安全学报,2021,6(03):227-256.
[6] Zhang K, Mao Y M, Leng S P, et al. Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks[J].IEEE Access, 2016, 4: 5896-5907.
[7] You C S, Zeng Y, Zhang R, et al. Asynchronous Mobile Edge Computation Offloading: Energy-Efficient Resource Management[J]. IEEE Transactions on Wireless Communications, 2018,17(11): 7590-7605.
[8] J. Huang, A. Zhou and S. Wang, "Price-Aware Service Deployment in Hierarchical Mobile-Edge Computing[J].in IEEE Internet of Things Journal, 2022,pp. 11533-11541.

*Corresponding author: Yuze Huang (E-mail: huangyz@cqjtu.edu.cn)

Author Contributions

Yuze Huang: Conceived the research idea on mobile edge computing resource allocation and defined the research problem.
Beipeng Feng: Designed the resource allocation function with dual arctangent mapping in the action space based on deep reinforcement learning algorithms and conducted experiments.
Yuhui Cao and Zhenzhen Guo: Provided real datasets and analyzed experimental results.
Beipeng Feng: Drafted the manuscript.
Yuze Huang: Revised the final version of the paper.

Note: Figure translations are in progress. See original paper for figures.