Federated Deep Reinforcement Learning for Continuous Improving Intradependente Task Offloading in Mobile Edge Computing Network

Abstract

Mobile edge computing often suffers from the dynamic and unknown nature of the environment such as time-varying conditions, heterogeneous devices, and frequent communication requests, imposing significant challenges on improving system performance. To meet the rapidly growing demands of computation-intensive and time-sensitive applications, Reinforcement learning [1] has been proposed as an effective tool to establish low-latency and energy-efficient networks. RL enables network entities to interact with the environment and learn an optimal decision-making policy, usually modeled as a Markov decision process [2].

Introduction

Mobile Edge Computing is emerging as a promising paradigm to enhance the computational capacity of mobile devices by offloading tasks to nearby edge servers. This paradigm aims to reduce latency, energy consumption, and improve Quality of Experience (QoE) for end-users. However, one of the major challenges in MEC is the efficient decision-making process for computation offloading, considering the dynamic nature of the network, user demands, and limited resources. Traditional offloading strategies, which often rely on heuristic or single-agent models, fail to capture the complexity and stochastic nature of modern MEC systems.

Motivation

The rapid growth of mobile applications, such as augmented reality, real-time gaming, and high-definition video streaming, has placed significant demands on the computational resources of mobile devices. While the processing power of these devices has improved, there remain significant limitations in battery life, processing speed, and memory capacity. Mobile Edge Computing (MEC) has emerged as a solution, enabling computation offloading to nearby edge servers to alleviate the processing burden on mobile devices. However, the challenge lies in determining how and when to offload computational tasks efficiently, especially in a dynamic network environment with varying resources and user demands. The decision-making process becomes more complex with the presence of multiple devices competing for limited edge resources. Current solutions often employ heuristic or single-agent approaches, which are not robust in highly dynamic and multi-user MEC environments. Multi-Agent Deep Reinforcement Learning (DRL) offers a promising avenue for addressing this challenge. By allowing multiple agents (mobile devices and edge servers) to autonomously learn and adapt their offloading strategies, it becomes possible to optimize system-wide performance metrics such as energy consumption, latency, and resource utilization.

Problem Statement

- **Devise-edge task offloading.** Efficient task offloading is crucial to ensure seamless resource distribution in MEC. Device-edge task offloading enables devices to independently make decisions on offloading resource-intensive tasks to nearby edge servers, fostering efficient utilization of available resources.

- **Edge-edge task offloading.** Task offloading leverages edge-edge collaborations, where tasks initially received by a local edge server can be offloaded to neighboring servers with underutilized resources, ensuring better resource utilization. The task offloading decision-making process focuses on efficiently distributing tasks among edge servers. Offloading tasks between edge servers requires communication resources and may introduce additional transmission delay, which should be taken into account when designing offloading strategies.

- **Intra-edge resource allocation.** On edge servers, there may be several tasks competing for resources among offloaded tasks on the same server. Intra edge there is a resource competition among offloaded tasks on the same server. Intra-edge resource allocation aims to determine how resources should be allocated to each offloaded task.

Research Methodology

1. **Algorithm Design:** We will develop a Multi-Agent Deep Reinforcement Learning algorithm using techniques such as Proximal Policy Optimization (PPO) or Deep Q-Networks (DQN), with a focus on communication and collaboration between agents.

2. **Simulation Environment:** A simulated MEC environment will be developed using Python or a suitable simulation platform, where mobile devices can offload tasks to edge servers under different network conditions.

References

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, “Human-level control through deep reinforcement learning,” *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[2] M. L. Puterman, *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.