A Deep Reinforcement Learning Approach for Computation Offloading at the 5G Network Edge

C. Stan,[1] J.J. Vegas Olmos,[2] I. Tafur Monroy,[1] and S. Rommel[1]

[1] Dept. of Electrical Engineering, Eindhoven University of Technology, 5600MB Eindhoven, The Netherlands

[2] NVIDIA Corporation, Ofer Industrial Park, Yokneam, Israel

Multi-access edge computing (MEC) is envisioned as a key enabler for fifth generation (5G) networks, bringing new resources into the computing continuum spanning from user devices to the centralized cloud. The advantages introduced by MEC play an important role in achieving the targeted 5G key performance indicators (KPIs), but at the same time its introduction raises new questions in terms of how frequently data should be offloaded from the end user device to the edge for processing since the optical front-, mid- and backhaul are constantly employed in the data transfer and MEC capabilities are limited compared to cloud data centers. In this context, a deep reinforcement learning (DRL) computation offloading solution is proposed which aims to provide optimal binary offloading decisions (local or MEC task execution). The DRL agent is set to evaluate the trade-off between factors such as computation energy consumption, communication delay via time-varying wireless channels and optical connections between 5G base stations (gNodeBs) and multiple MEC nodes. In order to preserve user data privacy, we propose an alternative training method for the DRL algorithm, i.e., a federated learning framework, as opposed to the centralized method where training data is collected in a predefined storage location.

Introduction

Multi-access edge computing (MEC) [1] comes at a time when the requirements for quality of experience (QoE) are becoming more demanding with each generation of mobile networks. To cope with this, the advantages introduced by the traditional mobile cloud computing (MCC) are leveraged when deploying edge nodes into the network. This way, the computing, storage and networking resources are brought in the close proximity of the user equipment (UE), therefore the requests sent to the network receive a much faster response since they are processed at the edge. Moreover, emerging applications developed under the umbrella of 5G mobile networks benefit from low latency and enhanced computing power from MEC. Compared to MCC, MEC provides flexibility since the edge nodes are deployed in a distributed manner, but at the same time it also comes with limited computational power and storage capacity [2]. In this context, offloading every computational task to the edge node for processing is not feasible, therefore optimal decisions in terms of offloading to edge nodes have to be taken. There are multiple examples of applications that can benefit from computation offloading, augmented and virtual reality (AR/VR), face recognition and gaming being just a part of them. The decision of whether to offload or not tasks coming from such applications in a constantly changing network environment such as the 5G network, starting from the wireless domain up to the optical backhaul, can easily turn into a complex problem. In this case, artificial intelligence (AI) techniques such as reinforcement learning (RL) can

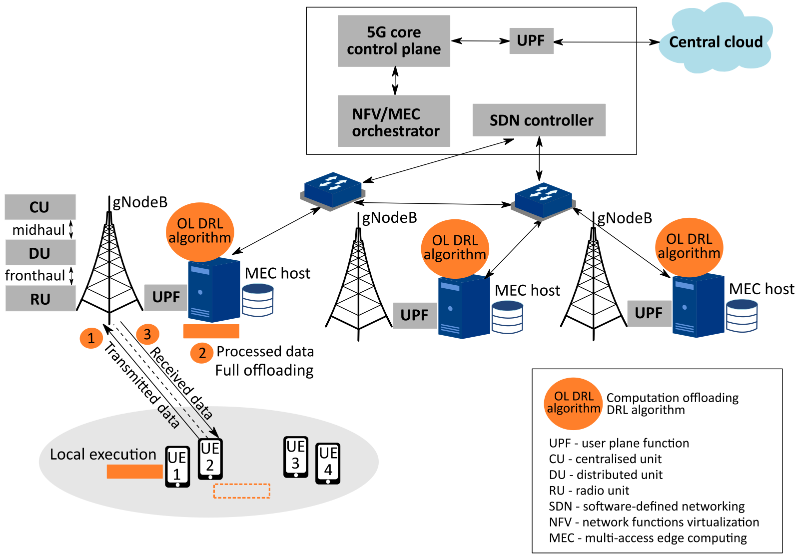

Fig. 1. 5G network architecture comprising MEC nodes with DRL agents and detailed offloading procedure (local or MEC execution).

be employed to take the optimal offloading decisions based on the always varying 5G environment and weight the trade-off between factors that influence offloading such as computational delay and energy consumption, to name but a few. Moreover, to deal with the computational complexity coming from the dynamic network states, deep reinforcement learning (DRL) can be adopted to approximate values or policies for RL problems with a large state space [3]. In respect to computation offloading decisions based on DRL, various research has been carried out for different types of networks. In [4] a two-time scale DRL approach was designed to minimize task execution time and computation, communication and storage usage in 5G ultra dense networks (UDN), also employing federated learning (FL) training. [5] uses DRL for computation offloading formulated as a binary optimization problem in a vehicular edge-cloud computing network to minimize the total energy and time cost of the system. DRL-based binary offloading decisions were also applied in [6] in an edge assisted Power Internet of Things (PIoT) network, aiming to maximize the average utility function, modeled as a weighted sum of reduction in latency and energy consumption. In [7], DRL-based online offloading was proposed to learn the binary offloading decisions from experience and deal with resource allocation, given the time-varying wireless channels.

In the following sections, we offer insights about computation offloading (factors that influence offloading, offloading types), propose a DRL framework where training is performed with centralized and distributed methods, and finally conclude the paper.

Deep reinforcement learning for computation offloading

When it comes to computation offloading, several challenges have to be addressed in order to offer a high QoE to the users. One challenge is the offloading decision itself which aims to determine whether the offloading will bring benefit to the users or not by evaluating the computation and communication costs. Since 5G networks are the main focus of the paper, user mobility is considered another challenge in offloading since service continuity has to be ensured while offloading [2]. The main focus of the paper is the offloading decision challenge where different offloading factors are weighted in order to make the optimal decision. Therefore, the main factors that can influence an offloading decision are: the communication cost, both in time-varying wireless channels and optical front-, mid- and backhaul and the energy consumption required for both communication and computation. Given these factors, one can define the offloading cost as follows:

$$O = w_1(T^{comm,DL} + T^{comm,UL}) + w_2T^{exec} + w_3(E^{comm,DL} + E^{comm,UL}) + w_4E^{exec} \quad (1)$$

where $T^{comm,DL}$ and $T^{comm,UL}$ represent the time required for uplink and downlink communication in the wireless and optical domains, and T^{exec} is the task execution time. $E^{comm,DL}$, $E^{comm,UL}$ and E^{exec} refer to the energy consumed to communicate on downlink, uplink and during task execution, respectively. $w_{1,2,3,4}$ represent the weights with values in [0,1], which are directly linked to the importance of each component specified in Eq. (1). In common offloading scenarios, the uplink channels are employed more often compared to the downlink channels and the reason is that the data volume transmitted by the UEs to gNodeB is higher compared to the response sent my MEC after processing. Moving on to offloading types, there are three common offloading decision that can be configured into the network: first is local execution – the computation is performed locally at the UE since the evaluation of the offloading cost did not show otherwise, second is full offloading at MEC – all the computation tasks as performed at MEC, and third is partial offloading – in this case, a fraction of the offloading tasks are performed at MEC, while the rest is executed locally on the UE. In Fig. 1., two such types of offloading decisions are represented in a 5G network architecture. First, for UE 1 the best decision is to proceed with local execution, while for UE 2, the optimal decision is to offload to MEC through 5G gNodeB. Steps 1 and 3 show the transmission on uplink and downlink, while step 2 is the actual processing.

Known for its ability to learn by interacting with the environment, DRL [3] is proposed for optimal decision-making in an offloading scenario. In order to do so, the first step is to model the DRL as a Markov Decision Process (MDP) with state, action and space. The state space can be represented by the user data rates in both uplink and downlink which are highly influenced by the large-scale fading (pathloss, shadowing) and small-scale fading (statistically modelled as Rice or Rayleigh distributions). Other components of the state space can be: the data size of the task, the number of CPU cycles required for processing or a delay constraint for low-latency applications. The action space is represented as the offloading decision which can be seen as a binary decision: 0 – local execution and 1 – offload to MEC. The reward function aims to minimize the offloading cost as exemplified in Eq. (1).

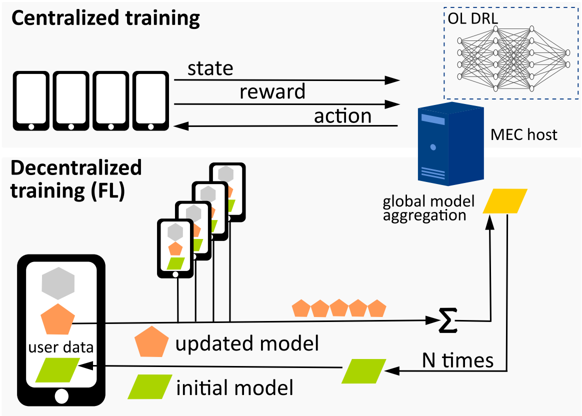

Fig. 2. Centralized and decentralized (FL) training methods for the offloading DRL.

In addition, apart from the traditional method of training an AI algorithm, i.e., the centralized method where the algorithm is running on a server where the training data is collected in a predefined storage location (Fig. 2. top), the DRL can also be trained in a decentralized manner using the FL framework. As depicted at the bottom of Fig. 2., the DRL model is sent from MEC to the selected UEs for on-device training and model update. The responses coming from the end devices are aggregated into a global model and after a specified number of rounds for model update, the final model is obtained. The benefit of such training is user privacy preservation since data necessary for DRL training coming from the UEs does not have to be sent and stored in a central database. In the FL framework, the update coming from the UEs contains much less information compared to the case where the entire raw data has to be centrally collected [8]. Such type of training also comes with drawbacks since the data coming from model update is transmitted through the wireless and optical domains, as it happens with offloading, therefore open questions still remain on how the two techniques interact with each other, how are the resources split in the network or what should be the priority queue between them.

Conclusions and future work

In this work, computation offloading together with DRL and FL are introduced as methods to improve QoE and shift the computationally intensive tasks at the 5G network edge. First, benefits and challenges of computation offloading were introduced, and second, a DRL algorithm with two training methods is proposed which aims to make optimal offloading decisions in the always changing and expanding network environment. As future work, the proposed offloading DRL is planned to be implemented and evaluated, as well as compared to the current benchmark solutions.

Acknowledgements

This work was supported by the EC H2020 MSCA ITN-ETN IoTalentum (grant no. 953442) and ECSEL JU BRAINE (grant no. 876967).

References

[1] ETSI GS MEC 003, “Multi-access Edge Computing (MEC); Framework and Reference Architecture,” March 2022, v3.1.1.

[2] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," in IEEE Commun. Surv. Tutor., vol. 19, no. 3, pp. 1628-1656, thirdquarter 2017, doi: 10.1109/COMST.2017.2682318.

[3] X. Wang et al., "Deep Reinforcement Learning: A Survey," in IEEE Trans Neural Netw. Learn Syst., 2022, doi: 10.1109/TNNLS.2022.3207346.

[4] S. Yu et al., ”When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multitimescale Resource Management for Multiaccess Edge Computing in 5G Ultradense Network,” in IEEE Internet Things J., vol. 8, no. 4, pp. 2238- 2251, 15 Feb.15, 2021, doi: 10.1109/JIOT.2020.3026589.

[5] M. Khayyat et al., ”Advanced Deep Learning-Based Computational Offloading for Multilevel Vehicular Edge-Cloud Computing Networks,” in IEEE Access, vol. 8, pp. 137052-137062, 2020, doi: 10.1109/ACCESS. 2020.3011705.

[6] J. Hu et al., ”Deep Reinforcement Learning for Task Offloading in Edge Computing Assisted Power IoT,” in IEEE Access, vol. 9, pp. 93892-93901, 2021, doi: 10.1109/ACCESS.2021.3092381.

[7] L. Huang et al., "Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks," in IEEE Trans. Mob. Comput., vol. 19, no. 11, pp. 2581-2593, 1 Nov. 2020, doi: 10.1109/TMC.2019.2928811.

[8] K. Bonawitz et al., “Towards Federated Learning at Scale: System Design,” 2019. [Online]. Available: https://arxiv.org/abs/1902.01046.