REMS: Resource-Efficient and Adaptive Model Selection in 5G NWDAF

Hyeonjae Jeong[1], Haneul Ko[2], and Sangheon Pack[1]
[1] School of Electrical Engineering, Korea University, Seoul, Korea.
[2] Department of Electronic Engineering, Kyung Hee University, Yongin-si, Korea.
Email: qeqe@korea.ac.kr, heko@khu.ac.kr, shpack@korea.ac.kr

***Abstract*—Deep neural networks (DNNs) are widely used to meet the needs of service consumers. However, most research on DNN focuses on the use of single-task learning models. In this approach, each task necessitates distinct inference resources, which results in the allocation of separate computing resources. This increases the possibility of failing to meet the deadline for task completion due to insufficient computing resources, especially as the task request rate increases. To tackle this challenge, we propose a resource-efficient and adaptive model selection (REMS) scheme that adaptively selects a multi-task learning model (MTL) and single-task learning (STL) model. We formulate the model selection problem as a Markov decision process (MDP) to minimize resource consumption while satisfying latency requirements. The optimal policy can be obtained by converting the MDP problem into Q-learning. The evaluation results demonstrate that REMS achieves a significant performance improvement compared to the existing scheme.**

***Index Terms*—Core network, multi-task learning, computing resource, Q-learning.**

I. Introduction

As the rapid expansion of 5G continues, there is a growing concern regarding control traffic overload in the core network due to the increased number of connected devices. Therefore, the core network requires evolution through the integration of AI-based control processes, using deep neural networks (DNN) to improve performance and automate network orchestration. However, the majority of research has primarily focused on the use of the conventional single-task learning (STL) model [1], [2]. Since the STL model can generate a single inference result, we need to conduct the inference of the STL model whenever the inference result is needed. Therefore, using the STL model in high task request rate scenarios can result in a substantial inference load. This load primarily occurs on the core network's primary AI function, the network data analytics function (NWDAF)

Efforts have also been made in research to alleviate the computational load through model transformation and the adoption of inference acceleration. Shao and Zhang [3] investigated a model transformation approach that balances computational costs and communication overhead to transmit intermediate features between the device and the edge server. They used incremental network pruning to achieve a significant compression ratio while minimizing performance degradation during the training process. Wang *et al.* [4] introduced inference acceleration methods that are input-dependent and resource-dependent inference mechanisms. In the input-dependent approach, a layer to skip is selected, while the resource-dependent method focuses on selecting the early-exit layer. However, methods such as model transformation or inference acceleration rely on fixed inference models and cannot dynamically select inference models based on resource conditions, making it challenging to adapt to fluctuations in task request rates.

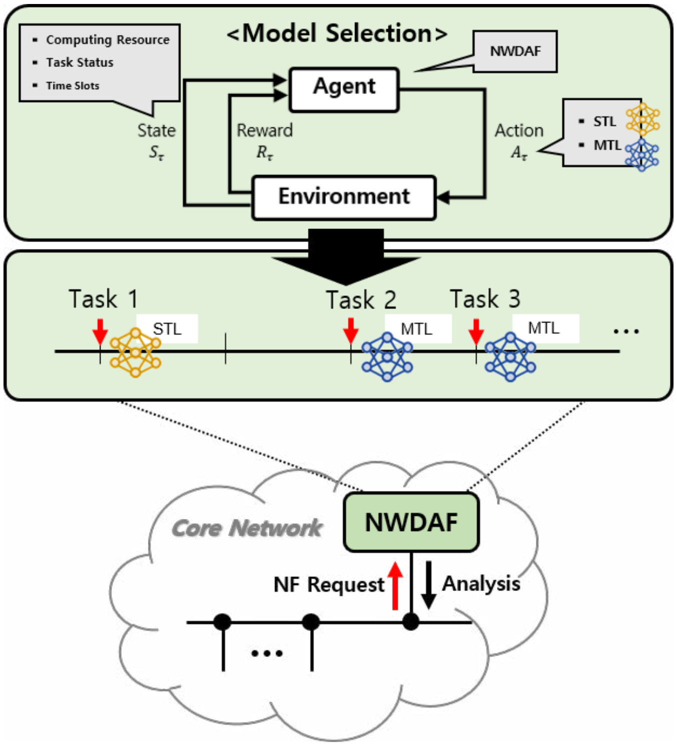

Fig. 1. System model.

To address this issue, we consider using a multi-task learning model (MTL). MTL is a machine learning approach that simultaneously performs multiple related tasks. MTL model only requires a single inference step for generating multiple results, leading to reduced computational resource usage. However, excluding STL and relying solely on MTL would fail to meet varying latency requirements for individual tasks. This is because each task has diverse latency requirements, whereas MTL aims to accommodate all tasks simultaneously, making

979-8-3503-1327-7/23/$31.00 ©2023 IEEE

it challenging to fulfill the diverse demands of each individual task. Therefore, it is crucial to selectively use MTL and STL to delay resource exhaustion while ensuring the task's required latency. To address this issue, we propose a resource-efficient and adaptive model selection (REMS) scheme that reduces the inference burden. Our approach focuses on selecting the optimal inference model across varying task request rate scenarios to minimize resource usage while satisfying the required latency of each task.

II. Resource-Efficient and Adaptive Model Selection Scheme

As shown in Figure 1, REMS is initiated when NF's task request arrives at NWDAF. A request includes the required latency and various types of analytics, such as user mobility prediction and slice load prediction. NWDAF can leverage this information, along with real-time resource monitoring, to make decisions in selecting an inference model on dynamic task request rate scenarios. The inference model selection problem is formulated on the basis of the Markov Decision Process (MDP). MDP is used to solve model decision-making problems in situations where the outcome of an action is uncertain. Let NWDAF be an agent in this system that interacts with an environment. At each step, an action is selected from a set of possible actions, and the environment responds by transitioning to a new state based on the chosen action. Its goal is to find an optimal policy that maximizes a specific objective, such as long-term rewards. We formulate our environment as MDP where:

- State space S_τ = {available computing resource, task status, time slots of each task}
- Action space A_τ = {STL, MTL}
- Reward R_τ = (number of completed tasks)/(number of inference)

NWDAF receives rewards through its state and action as it seeks to find the optimal policy. This allows NWDAF to effectively adapt its decision-making process according to the current availability of resources. Also, the reward is obtained only when the requested latency is met. Therefore, NWDAF aims to maximize rewards by reducing the number of inferences performed while increasing the number of completed tasks, prioritizing MTL selection over STL. Consequently, MTL is chosen first to minimize resource usage while meeting the required latency of each task. As a result, it is possible to achieve efficient inference resource utilization even in situations with high task request rates by employing MTL. We assume that the selection of the inference model occurs periodically at each time slot τ. Since we need to decide which inference model to use for each task at every time slot and the environment seems to be complex, determining transition probabilities becomes challenging. To address this, we use Q-learning to find the optimal policy. Q-learning allows us to make decisions on model selection effectively, ensuring resource efficiency while meeting latency requirements.

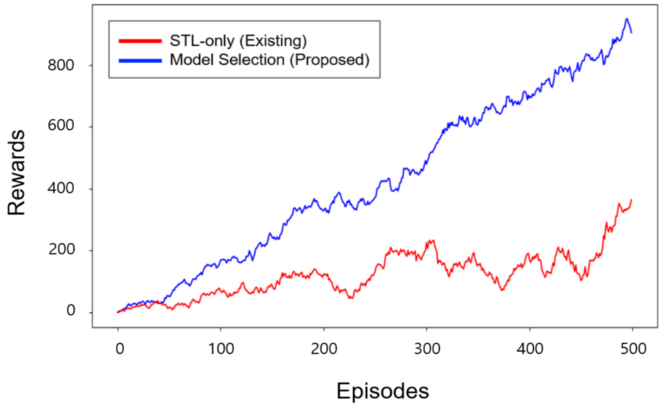

Fig. 2. Expected reward at task arrival rate = 1.

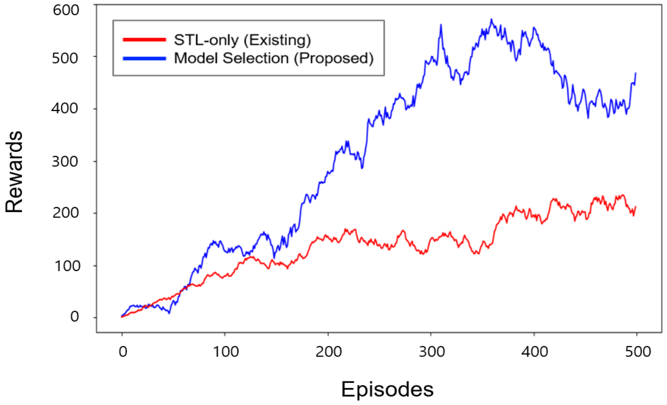

Fig. 3. Expected reward at task arrival rate = 2.

III. Evaluation Results

To demonstrate the effectiveness of REMS, we used the STL-only approach as a comparison scheme, where only STL models are used without any model selection process based on task request rates. We observe changes in the expected reward over 500 time slots for varying task arrival rates. In this simulation, our proposed scheme allows tasks to be jointly inferred using MTL models, with a range of 2 to a maximum of 4 tasks being inferred together. We assumed that both STL and MTL require the same computing resources for inference [5]. Figures 2 and 3 present the changes in average rewards over time for two different task arrival rates: 1 task per unit time and 2 tasks per unit time, respectively.

As shown in Figure 2 and Figure 3, the proposed scheme consistently outperforms the baseline approach, yielding higher rewards. REMS inherently leads to higher rewards as the inference count decreases, thus making MTL inevitably receive higher rewards compared to STL. Under high task arrival rates, as depicted in Figure 3, both schemes experience diminished rewards due to limited computing resources. The concept of attaining higher rewards implies less resource consumption, resulting in an increased number of tasks meeting latency requirements. In particular, in Figure 3, where the task request rate is relatively higher, our proposed scheme demonstrates significantly superior rewards compared to STL only. It is because our model selection scheme can achieve up to fourfold resource efficiency compared to STL-only when the same resources are allocated. Furthermore, since rewards are obtained when latency requirements are met, it can be concluded that our proposed scheme effectively achieves both the objectives of resource efficiency and latency requirement fulfillment.

IV. Conclusion

In this paper, we proposed the REMS for 5G NWDAF. REMS leverages MTL models to reduce the inference load while satisfying the latency requirement of tasks. Through an adaptive selection of both MTL and STL models, it is possible to reduce resource consumption in NWDAF. The evaluation results demonstrate that REMS consumes fewer inference resources than the compared scheme while achieving latency satisfaction. In our future work, we will search for applicable NWDAF scenarios and conduct more specific experiments.

Acknowledgment

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-01015, Development of Candidate Element Technology for Intelligent 6G Mobile Core Network).

References

[1] H. Yao, T. Mai, C. Jiang, L. Kuang, and S. Guo, "AI Routers & Network Mind: A Hybrid Machine Learning Paradigm for Packet Routing," *IEEE Computational Intelligence Magazine*, vol. 14, no. 4, pp. 21–30, November 2019.

[2] D. Bega *et al.*, "AI-Based Autonomous Control, Management, and Orchestration in 5G: From Standards to Algorithms," *IEEE Network*, vol. 34, no. 6, pp. 14–20, November/December 2020.

[3] J. Shao and J. Zhang "Communication-Computation Trade-off in Resource-Constrained Edge Inference," *IEEE Communications Magazine*, vol. 58, no. 78, pp. 20–26, December 2020.

[4] Y. Wang *et al.*, "Dual Dynamic Inference: Enabling More Efficient, Adaptive, and Controllable Deep Inference," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 4, pp. 623–633, May 2020.

[5] S. Vandenhende, *et al.*, "Multi-Task Learning for Dense Prediction Tasks: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3614–3633, July 2022.