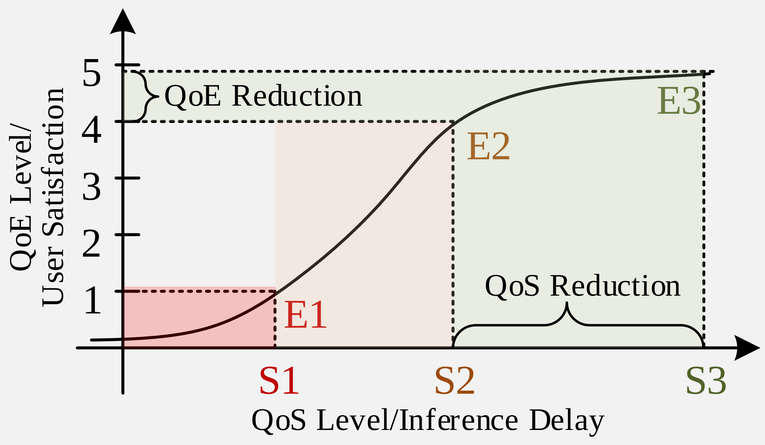
IEEE Hong Kong 6G Wireless Summit
11-12 September 2024
Hong Kong, China

QoE and QoS based High Efficiency Inference Accelerating Algorithm for Edge Intelligence

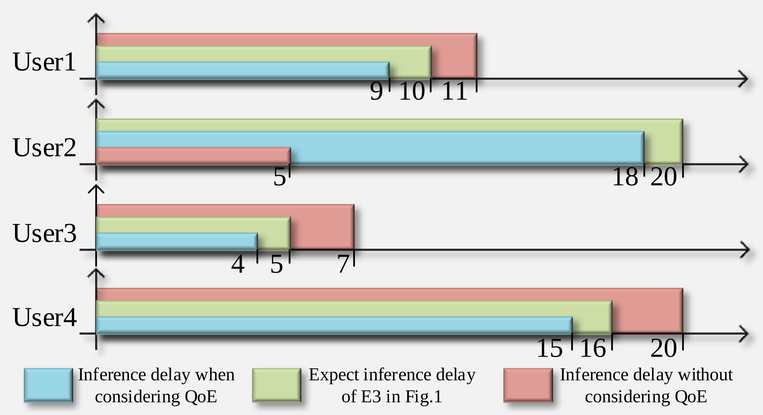
Xin Yuan, Ning Li, Quan Chen, Wenchao Xu, Jose Fernan Martinez, Song Guo

Abstract Even the artificial intelligence (AI) has been widely used and significantly changed our life, deploying the large AI models on resource limited edge devices directly is not appropriate. Thus, the model split inference is proposed to improve the performance of edge intelligence (EI), in which the AI model is divided into different sub-models and the resource-intensive sub-model is offloaded to edge server wirelessly for reducing resource requirements and inference latency. Unfortunately, with the sharp increasing of edge devices, the shortage of spectrum resource in edge network becomes seriously in recent years, which limits the performance improvement of EI. Refer to the NOMA-based edge computing (EC), integrating non-orthogonal multiple access (NOMA) technology with split inference in EI is attractive. However, the NOMA-based communication aspect and the influence of intermediate data transmission fail to be considered properly in model split inference of EI in previous works, and the sophistication in resource allocation caused by NOMA scheme makes it further complicated. Thus, the Effective Communication and Computing resource allocation algorithm is proposed in this paper for accelerating the split inference in NOMA-based EI, shorted as ECC. Specifically, the ECC takes the energy consumption and the inference latency into account to find the optimal model split strategy and resource allocation strategy (subchannel, transmission power, computing resource). Additionally, the properties of the proposed algorithms are investigated, including convergence, complexity, and approximation error. The experimental results demonstrate that the performance of ECC is much better than that of the previous studies.

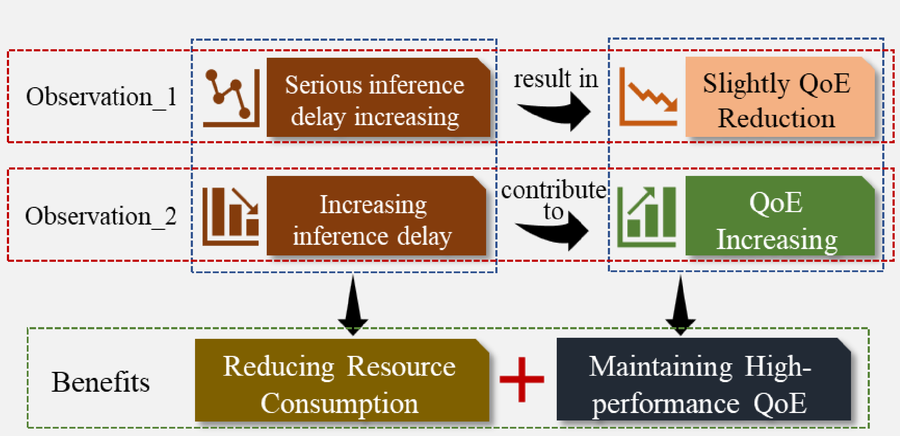
Motivation



The relationship of QoS and QoE



An example for the relationship of inference delay and QoE



The explanation and benefits of two observations

Network model and problem statement

Based on the network model and the proposed models (the ***split inference model, inference delay model and energy consumption model***), the problems that will be solved in this study (finding the minimum inference delay, maximum QoE and minimum resource consumption simultaneously) are described.

Optimal model split and resource allocation

The optimal solutions for Po are discussed. Since Po is difficult to solve, we propose an ***Li-GD algorithm*** for Po. Moreover, the properties of the proposed Li-GD algorithm are also investigated.

Properties of the Proposed Li-GD algorithm

The properties of the proposed Li-GD algorithm are investigated. The Li-GD algorithm is convergent, and the convergence time is $K = \frac{\|x^0 - x^*\|_2^2}{2\eta\epsilon}$, the complexity of the Li-GD is $O(X\bar{K}\mathcal{F}Mx^3\ln^2(x))$, the approximate error is smaller than $\frac{\varepsilon}{\rho_{min}(1-B_{max})\log_2\left(1+\frac{P_{min}}{\Delta^*+\frac{\alpha P_{max}}{2}}\right)}$. Additionally, it can reduce the complexity and convergence time compared with the traditional GD approach.

Algorithm 1: Loop iteration GD algorithm (Li-GD)

Input:

Objective function: $\mathbf{\Gamma} = \{\Gamma_1, \Gamma_2, \ldots, \Gamma_{s_j}, \ldots, \Gamma_{\mathcal{F}}\}$;

Gradient function: $\nabla = \{\nabla_{B_i} = \frac{\partial \Gamma_{s_j}}{\partial B_i}, \nabla_{P_i} = \frac{\partial \Gamma_{s_j}}{\partial P_i}, \nabla_{r_i} = \frac{\partial \Gamma_{s_j}}{\partial r_i}\}$;

Algorithm accuracy: ε;

Step size: λ;

Output:

The optimal solution $\mathbf{O}^* = \{\mathbf{B}^*, \mathbf{P}^*, \mathbf{r}^*\}$;

1. Let $\mathbf{B}^{j(0)} \in [0\ 1]$, $\mathbf{P}^{j(0)} \in [P_{min}\ P_{max}]$, and $\mathbf{r}^{j(0)} \in [r_{min}\ r_{max}]$, $\forall i \in [1\ U]$ and $\forall j \in [1\ \mathcal{F}]$;

 # Calculating the optimal strategy for the first layer #

2. If $j = 1$;
3. Let $k \leftarrow 0$, $\mathbf{B}^{j(k)} = \{B_1^{j(k)}, \ldots, B_{\mathcal{F}}^{j(k)}\}$, $\mathbf{P}^{j(k)} = \{P_1^{j(k)}, \ldots, P_{\mathcal{F}}^{j(k)}\}$ and $\mathbf{r}^{j(k)} = \{r_1^{j(k)}, \ldots, r_{\mathcal{F}}^{j(k)}\}$;
4. Calculating $\Gamma_{s_j}(\mathbf{B}^{j(k)}, \mathbf{P}^{j(k)}, \mathbf{r}^{j(k)})$;
5. Calculating the gradient $\mathbf{g_k} = g(\mathbf{B}^{j(k)}, \mathbf{P}^{j(k)}, \mathbf{r}^{j(k)})$;
6. If $\|g_k\| < \varepsilon$, then $\mathbf{B}^{j*} \leftarrow \mathbf{B}^{j(k)}$, $\mathbf{P}^{j*} \leftarrow \mathbf{P}^{j(k)}$ and $\mathbf{r}^{j*} \leftarrow \mathbf{r}^{j(k)}$;
7. Otherwise, let $\zeta_k = -g(\mathbf{B}^{j(k)}, \mathbf{P}^{j(k)}, \mathbf{r}^{j(k)})$, and let $\mathbf{B}^{j(k+1)} = \mathbf{B}^{j(k)} + \lambda\zeta_k$, $\mathbf{P}^{j(k+1)} = \mathbf{P}^{j(k)} + \lambda\zeta_k$, and $\mathbf{r}^{j(k+1)} = \mathbf{r}^{j(k)} + \lambda\zeta_k$;
8. Calculating $\Gamma_{s_j}(\mathbf{B}^{j(k+1)}, \mathbf{P}^{j(k+1)}, \mathbf{r}^{j(k+1)}) = \Gamma_{s_j}(\mathbf{B}^{j(k)} + \lambda\mathbf{p_k}, \mathbf{P}^{j(k)} + \lambda\zeta_k, \mathbf{r}^{j(k)} + \lambda\mathbf{p_k})$;
9. If $\|\Gamma_{s_j}(\mathbf{B}^{j(k+1)}, \mathbf{P}^{j(k+1)}, \mathbf{r}^{j(k+1)}) - \Gamma_{s_j}(\mathbf{B}^{j(k)}, \mathbf{P}^{j(k)}, \mathbf{r}^{j(k)})\| < \varepsilon$ or $\max\{\|\mathbf{B}^{j(k+1)} - \mathbf{B}^{j(k)}\|, \|\mathbf{P}^{j(k+1)} - \mathbf{P}^{j(k)}\|, \|\mathbf{r}^{j(k+1)} - \mathbf{r}^{j(k)}\|\} < \varepsilon$;
10. then $\mathbf{B}^{j*} \leftarrow \mathbf{B}^{j(k+1)}$, $\mathbf{P}^{j*} \leftarrow \mathbf{P}^{j(k+1)}$, and $\mathbf{r}^{j*} \leftarrow \mathbf{r}^{j(k+1)}$;
11. otherwise, $k = k + 1$;
12. end if

 # Calculating the optimal strategy of the rest layers #

13. When $1 < j \leq \mathcal{F}$;

 # Loop iteration #

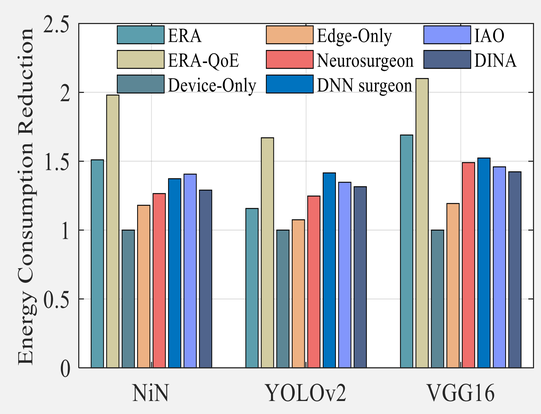
14. Let $\mathbf{B}^{j+1(0)} = \mathbf{B}^{j*}$, $\mathbf{P}^{j+1(0)} = \mathbf{P}^{j*}$, and $\mathbf{r}^{j+1(0)} = \mathbf{r}^{j*}$, $\forall i \in [1\ U]$ and $\forall j \in [1\ \mathcal{F}]$;
15. repeating step 3 to Step 11;
16. $j = j + 1$;

 # Finding the optimal strategy #

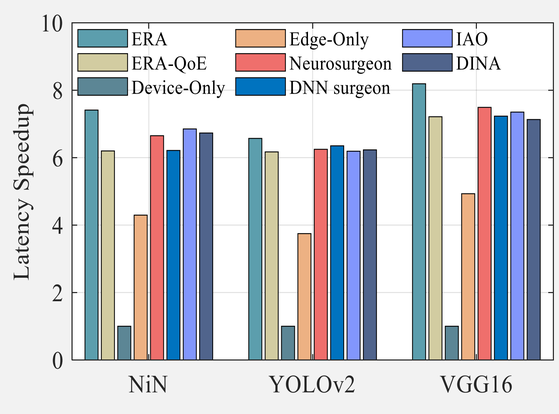
17. Calculating $\mathbf{\Gamma} = \{\Gamma_1(\mathbf{B}^{1*}, \mathbf{P}^{1*}, \mathbf{r}^{1*}), \ldots, \Gamma_{\mathcal{F}}(\mathbf{B}^{\mathcal{F}*}, \mathbf{P}^{\mathcal{F}*}, \mathbf{r}^{\mathcal{F}*})\}$;
18. $(\mathbf{s}, \mathbf{B}, \mathbf{P}, \mathbf{r}) \leftarrow \arg\min_{s^*, B^*, P^*, r^*} \mathbf{\Gamma}$;
19. If $B > 0.5 \rightarrow B = 1$;
20. otherwise $B = 0$.

Performance evaluation

We compare the performance of ECC-NOMA (ECC approach with NOMA channel) and ECC-OMA (ECC approach with OMA channel, shorted as ECC in the following sections) with that of the Device-Only, Edge-Only, Neurosurgeon, and DNN surgeon. We use the Device-Only method as the baseline, i.e., the performance is normalized to the Device-Only method.



Energy consumption reduction for different DNN models



Latency speedup for different DNN models

ACKNOWLEDGMENT

This work was supported in part by the Research Grants Council of Hong Kong under the Areas of Excellence scheme grant AoE/E-601/22-R.