三峡大学
CHINA THREE GORGES UNIVERSITY

IEEE GLOBECOM 2022

Fused-Layer-based DNN Model Parallelism and Partial Computation Offloading

Authors : Mingze Li[1], Ning Wang[2], Huan Zhou[1], Yubin Duan[3] and Jie Wu[3]

1. China Three Gorges University, Yichang, China
2. Rowan University, Glassboro, USA
3. Temple University, Philadelphia, USA

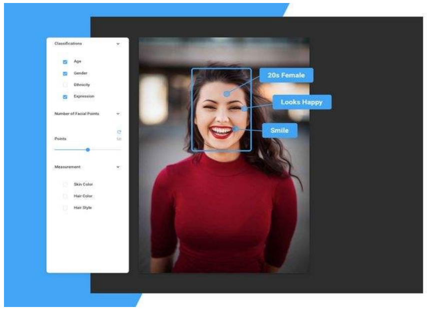
Contents

Contents

Next Part

Bottleneck of Deep learning

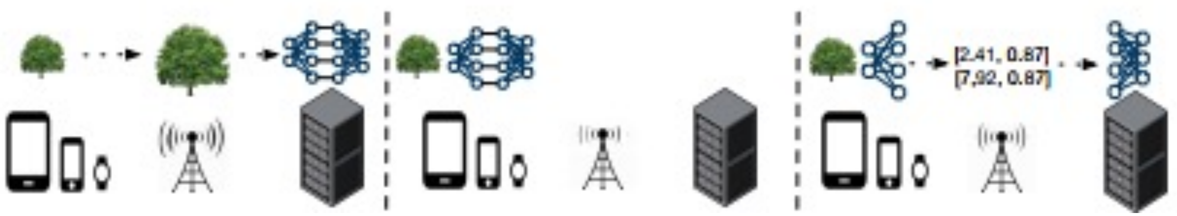
***Visions:** The flourishing IoT applications always have intense requirements for deep learning task, which needs real-time and high-precision results at the same time.*



Deep learning has shown success in complex tasks, including computer vision, natural language processing, auto pilot and many other tasks.

Device or Edge

- **Partial Computation Offloading**

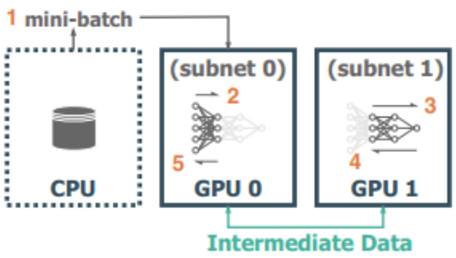


Fully Offloading **End device-only** **Partial Computation Offloading**

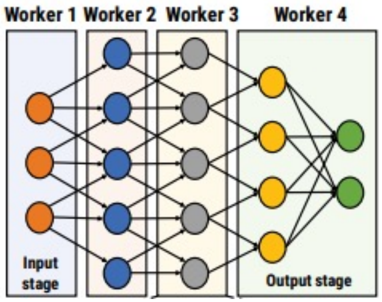
Some limitations of partial computation offloading：

- ***The dataset transmission between ES and end device-> Extra transmission cost***
- ***Cannot guarantee the latency-sensitive tasks***

Model Parallelism



Multi-GPU



Multi workers

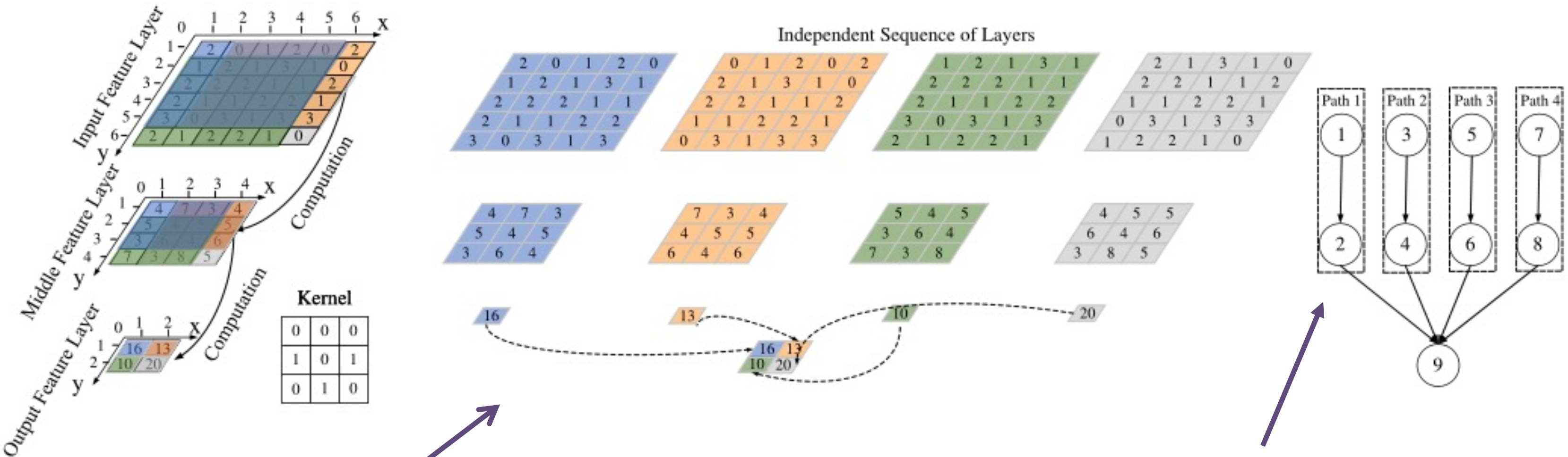
- **Model Parallelism**: DNN inference is extremely time-consuming, necessitating efficient multi-accelerator parallelization

- The design of fine-grained partitioning of DNN models and parallel computing strategies in edge computing environment still lacks due attention

Contents

Next Part

FL Technique and Challenges



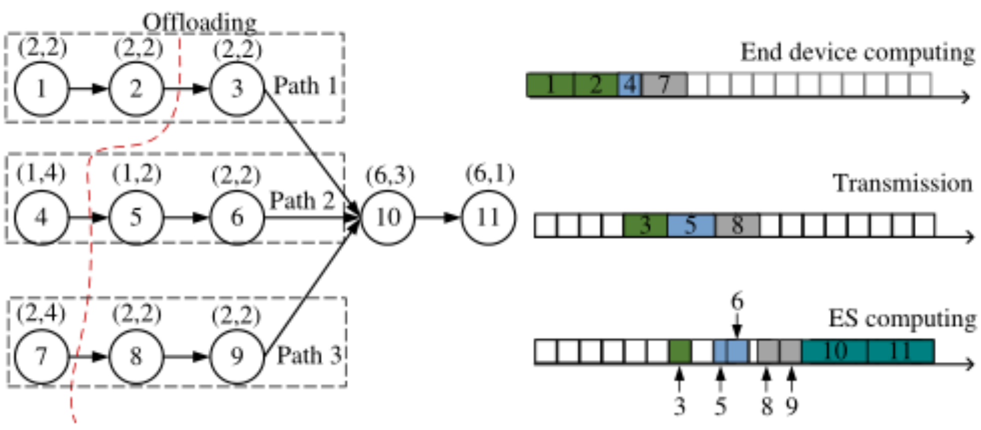
FL strategy: ***How*** **to develop the optimal strategy of FL path length, the number of paths, and the size of the fused layer?**

Path offloading strategy: ***How*** **to determine the offloading layer of each path?**

Path scheduling strategy: ***How*** **to determine the optimal path scheduling order?**

System Model

- *An end device and an ES will work collaboratively to finish DNN inference task.*
- *The computation dependency relationship of layers in a DNN as a DAG with V layers and P paths*



– c_v *the transmission data size of layer* v

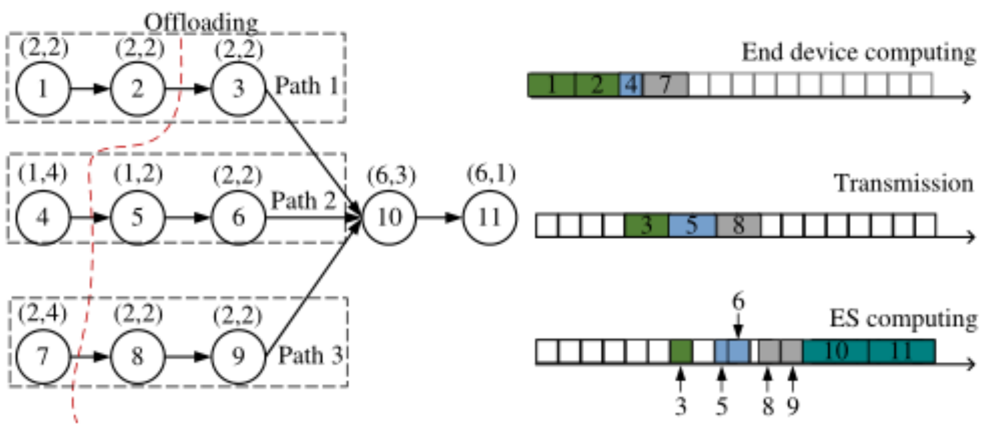
- d_v *the amount of computation of layer* v

- $e_{v'v}$ *the computation dependency relationship*

- h_v *the computation offloading strategy*

System Model

DNN Partial Computation Offloading Model



- *Computation time of the end device.*

$$t_v^{end} = \frac{d_v}{f_{end}}$$

- *Transmission time between the end device and the ES.*

$$t_v^{tr} = \frac{c_v}{R}$$

- *Computation time of the ES.*

$$t_v^{es} = \frac{d_v}{f_{es}}$$

Contents

Next Part

Problem Description

We aim to minimize the DNN inference time in partial computation offloading while considering DNN model parallelism optimization.

- *Let $T_p(v)$ denote the task completion time of layer v on path p*

$$\mathrm{T}_p(v) = \begin{cases} \max \mathrm{T}_p(v') + t_v^{end}, & \{h_{v'}, h_v\} = \{0, 0\}, \\ \max \mathrm{T}_p(v') + t_v^{tr} + t_v^{es}, & \{h_{v'}, h_v\} = \{0, 1\}, \\ \max \mathrm{T}_p(v') + t_v^{es}, & \{h_{v'}, h_v\} = \{1, 1\}, \end{cases}$$

- *$T(v)$ is the task completion time of layer v after FL paths fused on ES*

$$\mathrm{T}(v) = \max\ \mathrm{T}(v') + t_v^{es}$$

Problem Formulation

The optimization problem can be formulated as:

$$(\mathbf{P}) \quad \min_{\mathbb{F},\mathbb{S},\mathbb{O}} \mathbb{T} = \max T(v)$$

$$s.t. \quad C1: \quad T_p(v') \leq T_p(v)$$

$$C2: \quad T(v') \leq T(v)$$

C1 & C2: The computation dependency. The computation layer can only be computed if all of its predecessors have been computed.

Challenges:

- ✓ **Low time complexity method**
- ✓ **Belongs to HP-hard**

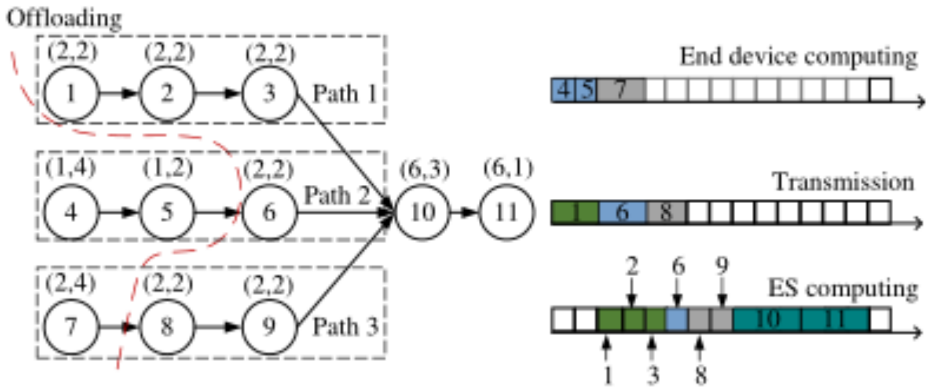
Problem Solving

Algorithm 1 Minimizing Waiting

Input: Neural network layers l and their parameters.

Output: The minimum completion time $\mathbb{T}_{MW}^{best}$; The best solution $\mathbb{U}_{MW}^{best}$, τ_{MW}^{best}, $\mathbb{O}_{MW}^{best}$, $\mathbb{S}_{MW}^{best}$.

Initialize $\mathbb{T}_{MW}^{best}$ = NULL
for FL path length $\tau = 1 : l$ **do**
  for The number of FL paths $P = 1 : S_\tau^L \times S_\tau^W$ **do**
    The intercepted fused layer's size is obtained as S_τ divided into P layers of the same size.
    for each FL path **do**
      The first scheduling path and the offloaded layer are determined as:
        $p, o_p \leftarrow \min\left(\mathrm{T}_p(v-1) + t_v^{tr}\right)$
      The scheduling path is recorded as s_1 and the offloaded layer o_p is recorded in $\mathbb{O}$.
    end for
    for The p-th scheduling path $p = 2 : P$ **do**
      The p-th scheduling path and the offloaded layer are determined as:
        $p, o_p \leftarrow \min\left(\left|\mathrm{T}_{p-1}(v) - t_v^{es} - \mathrm{T}_p(v')\right|\right)$
      The scheduling path is recorded as s_p and the offloaded layer o_p is recorded in $\mathbb{O}$.
    end for
    According to the $\mathbb{S}$, $\mathbb{O}$, $\mathbb{U}$, the DNN inference time $\mathbb{T}_{MW}$ is obtained.
    If $\mathbb{T}_{MW}^{best}$ = NULL
      Update $\mathbb{T}_{MW}^{best} \leftarrow \mathbb{T}_{MW}$, $\tau_{MW}^{best} \leftarrow \tau$,
        $\mathbb{S}_{MW}^{best} \leftarrow \mathbb{S}$, $\mathbb{U}_{MW}^{best} \leftarrow \mathbb{U}$.
    End If
    If $\mathbb{T}_{MW} \leq \mathbb{T}_{MW}^{best}$
      Update $\mathbb{T}_{MW}^{best} \leftarrow \mathbb{T}_{MW}$, $\tau_{MW}^{best} \leftarrow \tau$,
        $\mathbb{S}_{MW}^{best} \leftarrow \mathbb{S}$, $\mathbb{U}_{MW}^{best} \leftarrow \mathbb{U}$, $\mathbb{O}_{MW}^{best} \leftarrow \mathbb{O}$.
    End If
  end for
end for



The Basic Idea of Minimizing Waiting Method once the current path has completed its transmission, the next path should finish its computation and start to transmit without waiting.

Contents

Next Part

Performance Evaluation

For performance comparison, we introduce the following two benchmark methods:

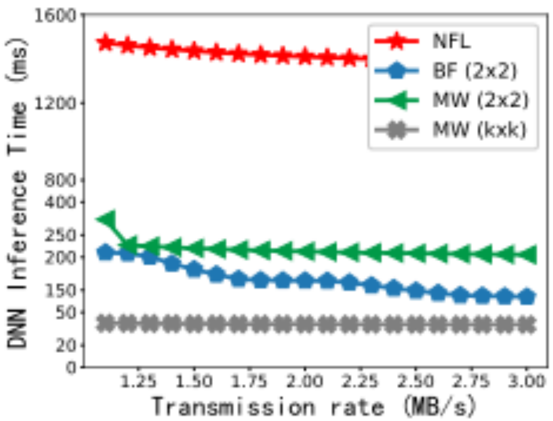
- ***No Fused-Layer (NFL):*** *Partial computation offloading without the FL technique is used in this algorithm.*

- ***Brute Force (BF):*** *The FL technique is used in this algorithm, and the optimal solution is obtained by traversing all feasible solutions.*

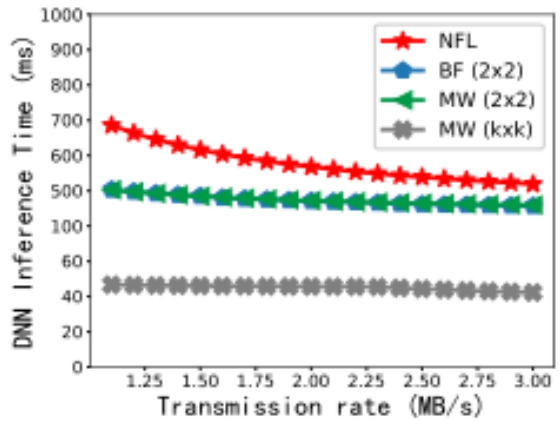
Performance Evaluation

Compare the performance : The DNN inference time with only changing transmission speed.

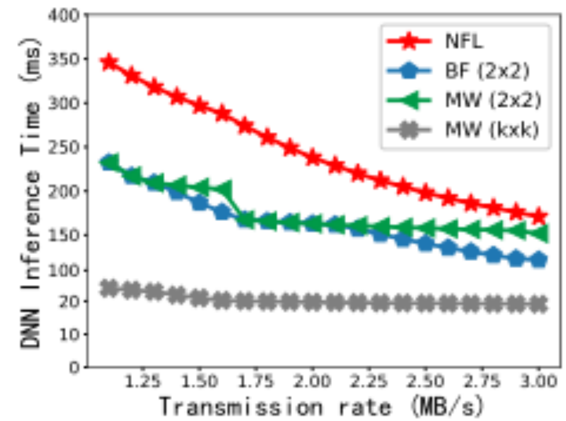
✓ **(a) AlexNet. (b) SqueezeNet. (c) YOLOv2**



(a) AlexNet



(b) SqueezeNet



(c) YOLOv2

Contents

Next Part

Conclusion

- In this paper, we presented a new solution for DNN parallelism and partial computation offloading in MEC.
- We proposed a DNN partitioning model based on the FL technique and the corresponding computation model when the DNN is transformed into a DAG.
- Subsequently, we proposed the MW method to solve the problem. Specifically, we design the MW algorithm to determine the FL strategy, path scheduling strategy, and path offloading strategy.
- Finally, we validated the effectiveness and superiority of the method through extensive simulation experiments, and the simulation results showed that our proposed method can reduce the DNN inference time by an average of 18.39 times compared with NFL.

Q&A

Q & A

Mingze Li

Email :limingze927@163.com