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

Authors: Mingze Li\textsuperscript{1}, Ning Wang\textsuperscript{2}, Huan Zhou\textsuperscript{1}, Yubin Duan\textsuperscript{3} and Jie Wu\textsuperscript{3}

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

1 Introduction

2 System Model

3 Problem Formulation & Solving

4 Performance Evaluation

5 Conclusion
Contents

1 Introduction

2 System Model

3 Problem Formulation & Solving

4 Performance Evaluation

5 Conclusion
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

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

- **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

1. Introduction

2. System Model

3. Problem Formulation & Solving

4. Performance Evaluation

5. Conclusion
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.

\[ t_{v\text{end}} = \frac{d_v}{f_{\text{end}}} \]

• Transmission time between the end device and the ES.

\[ t_{v\text{tr}} = \frac{c_v}{R} \]

• Computation time of the ES.

\[ t_{v\text{es}} = \frac{d_v}{f_{\text{es}}} \]
Contents

Next Part

1 Introduction

2 System Model

3 Problem Formulation & Solving

4 Performance Evaluation

5 Conclusion
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 $p$

  \[
  T_p(v) = \begin{cases} 
  \max T_p(v') + t_v^{end}, & \{h_{v'}, h_v\} = \{0, 0\}, \\
  \max T_p(v') + t_v^{tr} + t_v^{es}, & \{h_{v'}, h_v\} = \{0, 1\}, \\
  \max T_p(v') + t_v^{es}, & \{h_{v'}, h_v\} = \{1, 1\},
  \end{cases}
  \]

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

  \[
  T(v) = \max \ T(v') + t_v^{es}
  \]
Problem Formulation

The optimization problem can be formulated as:

\[(P) \quad \min_{F,S,\emptyset} T = \max T(v)\]

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

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

\( C1 \) & \( C2 \): 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
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.
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
Contents

Next Part

1 Introduction
2 System Model
3 Problem Formulation & Solving
4 Performance Evaluation
5 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

Mingze Li

Email: limingze927@163.com