DeePar: A Hybrid Device-Edge-Cloud Execution Framework for Mobile Deep Learning Applications

Yutao Huang¹, Feng Wang², Fangxin Wang¹, Jiangchuan Liu¹

¹School of Computing Science, Simon Fraser University, Canada
²Department of Computer and Information Science, The University of Mississippi, USA

2019/04/29
DeePar: A Hybrid Device-Edge-Cloud Execution Framework for Mobile Deep Learning Applications

Outline

- Introduction
  - Today’s challenges for mobile deep learning applications
- DeePar: Layer-level Partitioning Optimization for DNNs
  - Enabling layer-level partitioning optimization for DNN inference
  - Scheduling tasks for optimized total delay
  - Experiment and simulation results
- Conclusion
Introduction

AI is changing our lives

- Face recognition
- Machine translation
- self-service supermarket
- Driverless car
Introduction

Rising in mobile deep learning applications

- Apple Siri
- Uber routing
- Face ID
- Migraine buddy
Introduction

- Models are getting larger:

  - 8 layers
    - ~16% error
    - AlexNet(2012)
  - 19 layers
    - ~7.5% error
    - VGG(2014)
  - 152 layers
    - ~3.5% error
Introduction

- Cloud: current solution for mobile ML apps

Amazon instances with GPU computing

<table>
<thead>
<tr>
<th>GPU Instances - Current Generation</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>p3.2xlarge</td>
<td>$3.06 per Hour</td>
<td>$0.27 per Hour</td>
</tr>
<tr>
<td>p3.8xlarge</td>
<td>$12.24 per Hour</td>
<td>$0.27 per Hour</td>
</tr>
<tr>
<td>p3.16xlarge</td>
<td>$24.48 per Hour</td>
<td>$0.27 per Hour</td>
</tr>
<tr>
<td>g3.4xlarge</td>
<td>$1.14 per Hour</td>
<td>$0.27 per Hour</td>
</tr>
<tr>
<td>g3s.xlarge</td>
<td>$0.75 per Hour</td>
<td>$0.188 per Hour</td>
</tr>
</tbody>
</table>
Introduction

- Disadvantages of cloud computing
  - Huge volume of Internet traffic
  - Limited network bandwidth
  - High latency response
  - Security issue
Introduction

Edge Computing:

The cloud
- Big data processing
- Data warehouses

The edge
- Real time data processing
- Local processing

Internet of things
- Smart devices
- Smart vehicles
- Connected systems
Introduction

- Advantages of edge computing

- Real-time or near real-time reaction
- Lower operating costs
- Reduced core network traffic
- Improved application performance
**Introduction**

- Edge-assisted learning

Diagram:

- Mobile device
- Network edge
- Remote cloud
Edge-assisted learning

For each task, which edge server should be arranged to offload data and computation?

How to allocate resource for each task?

What/which part shall be processed on the edge?

Mobile device  Network edge  Remote cloud
Outline

- Introduction
  - Today’s challenges for mobile deep learning applications

- DeePar: Layer-level Partitioning Optimization for DNNs
  - Enabling layer-level partitioning optimization for DNN inference
  - Scheduling tasks for optimized total delay
  - Experiment and simulation results

- Conclusion
Motivation

- AlexNet layer-level performance
Motivation

- AlexNet performance comparison

![AlexNet performance comparison chart](chart.png)
DeePar: A collaborative execution approach

- DeePar Framework for one single task
DeePar: A collaborative execution approach

- Online multi-task scheduling

Network resource indicator (from the device to the edge): $x_{ie}$

Edge computation resource indicator: $z_{ie}$

Network resource indicator (from the edge to the cloud): $y_{ie}$

Mobile device $i \in I$

Network edge $e \in E$

Remote cloud $C$
DeePar: A collaborative execution approach

Constraints

- Bandwidth constraint:
  \[ \sum_{i \in I} b_i^1 \times x_{ie} \leq B_e^1 \]

- Computation resource constraint:
  \[ \sum_{i \in I} z_{ie} \leq r_e \]

- Bandwidth constraint:
  \[ \sum_{i \in I} b_i^2 \times y_{ie} \leq B_e^2 \]
DeePar: A collaborative execution approach

Objective

Data transmission delay between device and edge server, and between edge server and cloud

Target: minimizing the total execution delay

\[
\min \sum_{i \in I} (t_i^1 + t_i^2 + t_i^3 + t_i^4 + t_i^5 + t_i^6)
\]

Computation delay on device, edge server and cloud

Final result transmission delay (from cloud to the device)
DeePar: A collaborative execution approach

- Delayed-start strategy
DeePar: A collaborative execution approach

- Single-task experiment

![Graph showing delay comparison between different execution environments](image_url)
DeePar: A collaborative execution approach

- Single-task experiment

VGG-16
DeePar: A collaborative execution approach

- Single-task experiment

![Graph showing delay comparison for different execution environments with LeNet]

- Computation delay
- Data transmission delay

Execution environment:
- Device-only
- Cloud-only
- Edge-only
- Device-edge-cloud
DeePar: A collaborative execution approach

- Multi-task simulation

![Graph showing delay vs number of IoT devices]

Time interval within 300s, 10 edge servers
DeePar: A collaborative execution approach

- Multi-task simulation

Time interval within 50s, 10 edge servers
Multi-task simulation

Time interval within 50s, 40 edge servers
Outline

- Introduction
  - Today’s challenges for mobile deep learning applications
- DeePar: Layer-level Partitioning Optimization for DNNs
  - Enabling layer-level partitioning optimization for DNN inference
  - Scheduling tasks for optimized total delay
  - Experiment and simulation results
- Conclusion
Conclusion

- We propose DeePar, a double-partition layer-level neural network partitioning optimization framework for edge inference tasks.

- We formulate a multi-task scheduling problem for DeePar and propose an online algorithm with a delayed-start strategy.

- Through experiments and simulations, DeePar can outperform device-only, edge-only and cloud-only execution with 20% - 80% delay reduction.