Dělen: Enabling Flexible and Adaptive Model-serving for Multi-tenant Edge AI

Qianlin Liang, Walid A. Hanafy, Noman Bashir, Ahmed Ali-Eldin, David Irwin, Prashant Shenoy

[05/11], 2023
AI-based IoT applications are becoming pervasive

• Many IoT applications require low-latency processing.

• Edge computing has emerged as the preferred architecture.

• Edge AI provides many benefits:
  ▪ Latency
  ▪ Bandwidth
  ▪ Privacy
Motivations: adaptive model serving at the edge

**Multi-tenant:** sharing *constrained* resources across workloads.

**Workload dynamics:** potential bursty workloads.

**Energy dynamics:** energy constraints due to limited energy availability.
Objectives

Design an edge model-serving system that is:

- **Multi-tenant**: share resources across multiple workloads.
- **Flexible**: satisfy a wide range of application SLOs.
- **Adaptive**: handle potential workload and energy dynamics.
- **Lightweight**: run on low-end devices.
Outline

Introduction

Dělen Design

• Dělen Evaluation

• Conclusion
Introduce to multi-exit DNNs

- Multi-exit DNNs[1] incorporate several exits points.
- Outputting at early exits will skip the execution of the rest of the network.
- Enable making trade-offs between accuracy, latency and energy.

Conditional Execution Framework

- A mechanism to provide applications with a configurable execution criteria.
- **Flexible** in supporting a wide range of exit criteria for application objectives.
- **Adaptive** in allowing applications to change their exit criteria at runtime.
Conditional Execution Framework

• Make adaptation by specifying and combining criteria.

• Flexible to implement a wide range of policies.
  ▪ Application-specific policies.
  ▪ Multi-tenant policies.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Operators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response time</td>
<td>&gt;</td>
</tr>
<tr>
<td>Confidence</td>
<td>==</td>
</tr>
<tr>
<td>Accuracy</td>
<td>&lt;</td>
</tr>
<tr>
<td>Energy</td>
<td>OR</td>
</tr>
<tr>
<td>FLOPs</td>
<td>AND</td>
</tr>
</tbody>
</table>

Dělen exit-selection criteria
Pareto Adaptation Policy

- **Idea**: Opportunistically choose early exits when confident.
- **Problem**: Choosing the right confidence threshold is crucial.
- **Method**: Optimize metrics with Pareto Frontier from workload profiles.
Multi-tenant Adaptation

- Enable support for multi-tenancy.
- Adapt to the change of shares and update the criteria accordingly.
- Multi-tenant policies
  - Cooperative
  - Non-cooperative
Outline

- Introduction
- Dělen Design

Dělen Evaluation

- Conclusion
Dělen Implementation

Hardware
• Nvidia Jetson Nano
• 18650 Li-Ion Battery

Software:
• TensorRT 7.1.3
• CUDA 10.2
• PyTorch 1.9

Workloads:
• Image classification
• Speech Recognition
Dělen’s Flexibility

**Objective**: optimize energy efficiency when meeting target accuracy.

**Policies**:
- Oracle: choose the first that is correct.
- PF: Pareto-Frontier
- pW: per-Workload, choose the first satisfied exit for all request.
- Full: the full model

**Key insight**: Dělen allows users *flexibly* specify high-level objectives and meeting them using different policies.
Key insight: Dělen is able to adapt to battery dynamics and prolong the battery life by up to 59%.
Dělen’s Adaptability – Workload

**Key insight:** The adaptability of Dělen allows applications to adapt to workload dynamics.
Dělen’s Multi-tenancy

**Key insight:** Dělen supports multi-tenancy by enabling flexible exit selection and runtime workload adaptation.
Conclusion

• Dělen is a *flexible*, *adaptive*, and *multi-tenant* model-serving system for supporting AI-based IoT applications on edge platforms.

• Dělen’s flexibility is demonstrated through the implementation of various adaptation policies using its API.

• Dělen’s adaptability was evaluated under different environmental dynamics and objectives when running single and multiple concurrent applications.
Questions?
Thank you!
COMPUTING FOR THE COMMON GOOD

Qianlin Liang
qliang@cs.umass.edu