XDN: Towards Efficient Inference of Residual Neural Networks on Cambricon Chips

— 2019 BenchCouncil International Artificial Intelligence System Challenges —

Guangli Li, Xueying Wang, Xiu Ma
Advisor: Prof. Xiaobing Feng

State Key Laboratory of Computer Architecture,
Institute of Computing Technology, Chinese Academy of Sciences
XiaoDianNao Team

Team Members:

Guangli Li is a Ph.D. student at ICT, CAS. His research interests include Programming Systems and Deep Learning.

Xueying Wang is a Ph.D. student at ICT, CAS. Her research interests include Parallel Programming and Deep Learning.

Xiu Ma is a Ph.D. student at Jilin University and a visiting student at ICT, CAS. Her research interests include Programming Language and Compiler.

Advisor:

Prof. Xiaobing Feng
Institute of Computing Technology, Chinese Academy of Sciences
Track 2
International AI System Challenge based on Cambricon Chip
18 Sept. ~ 14 Oct. (about 4 weeks)

Subject:
The implementation and optimization of convolutional neural network based image classification task on Cambricon AI Chips (MLUs).

Metrics:
Maximize the execution performance and minimize the prediction time (wall clock time) on provided test data.

Provided model: ResNet-50 (ACC: 84.39%)
Dataset: CIFAR-10

Provided Development Tools: Cambricon Caffe
Experiment Platform: BenchCouncil Testbed
Contents

I. Optimization Methodology
II. Implementation of the XDN Engine
III. Experimental Results
IV. Conclusion and Future Work
Methodology

- Aggressive Fusion Strategy
- Fusion-Guided Pruning Method
- Traditional Fusion Strategy
XDN (XiaoDianNao) —— An Efficient Optimization and Inference Engine

I. Pruning Optimizer and Trainer (aggressive fusion and pruning)
II. Fusion Optimizer (traditional fusion)
III. Auto-Tuner
IV. Data Preprocessor and Executor
V. Evaluator
Implementation of XDN

- Pruning and Fusion

---

Caffe Prototxt

Original Model

Pruning Optimizer

Cambricon Caffe (Trainer)

Fusion Optimizer

Caffe Model
Implementation of XDN

- Pruning and Fusion

Pruning Optimizer → Cambricon Caffe (Trainer) → Fusion Optimizer → Caffe Prototxt → Caffe Model → Optimized Model
Implementation of XDN

- Data Preprocessing

CIFAR-10 Images \[\rightarrow\] OpenCV \[\rightarrow\] Inference

Original Executer: On-line Data Preprocess
Implementation of XDN

- Data Preprocessing

Off-line Data Preprocess & Efficient Memory-mapped File
Implementation of XDN

- Auto-Tuning of Hyper-Parameters: A Grid Search Approach

Finding a best option of hyper-parameters
## Final Results

<table>
<thead>
<tr>
<th>Optimization</th>
<th>Total Time (ms)</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>2263</td>
<td>0.8439</td>
</tr>
<tr>
<td><strong>Filter Pruning</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPT-A-1</td>
<td>485</td>
<td>0.8439</td>
</tr>
<tr>
<td>OPT-B-1</td>
<td>449</td>
<td>0.8462</td>
</tr>
<tr>
<td>OPT-C-1</td>
<td>428</td>
<td>0.8191</td>
</tr>
<tr>
<td><strong>Pruning + Conv &amp; BN Fusion</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OPT-A-2</td>
<td>485</td>
<td>0.8441</td>
</tr>
<tr>
<td>OPT-B-2</td>
<td>304</td>
<td><strong>0.8455</strong></td>
</tr>
<tr>
<td>OPT-C-2</td>
<td>297</td>
<td>0.8203</td>
</tr>
</tbody>
</table>

### Performance of ResNet-50 with XDN Engine

Best Result: OPT-B-2, TIME=304ms, ACC=84.55%

1.44x speedup
Performance Analysis

<table>
<thead>
<tr>
<th>Optimizations of XDN</th>
<th>Total Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pruning</td>
<td>Fusion</td>
</tr>
<tr>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Performance of OPT-B-2

Each part contributes the final performance.
Artifact Description


- Detailed XDN Documentations (English and Chinese)
- The Codes of Reproducible Experiments

1. Code Structure

   XiaoDianNao
   ⊘ cifar10 ............................................ Example Test Dataset
   ⊘ input_list.txt .................................... Input Image List
   ⊘ label_list.txt .................................... Label List
   ⊘ model ............................................ Prototxts and Caffe models
     └... offline_models ................................ Offline Models
     └... clas_offline_XDN ................................ Source Code
       └ modules                                     
       └... clas_offline_XDN.cpp                   
       └ CMakeList.txt                             
   tools ............................................ Data Preprocess Tool
     └ input2binary.cpp                          
     └ preprocess.sh ................................ Preprocess Script
     └ genoff.py .................................... Offline Model Generation Script
     └ run_test.sh ................................... Test Script
     └ run_test_baseline.sh ............................ Test Script with Baseline
     └ evaluate.py ................................... Evaluation Script
Conclusion and Future Work

Highlight Contributions:

1. We proposed a optimization methodology
   • Characteristics of Cambricon Chips
   • Fusion-Guided Pruning Method

2. We implemented the Efficient XDN engine
   • Pruner, Trainer, Optimizer, Auto-Tuner, Executer…

3. We evaluated the performance
   • Achieves high speedup (7.44x) without accuracy loss

In the future, we plan to:
• test the method on large-scale datasets, such as ImageNet;
• extend the XDN engine to support more DNN models;
• test the method on other AI chips;
• …
XDN: Towards Efficient Inference of Residual Neural Networks on Cambricon Chips

Thank You

liguangli@ict.ac.cn