tinyML® Asia

Enabling Ultra-low Power Machine Learning at the Edge

November 29 & 30, 2022

www.tinyML.org
TILE-MPQ: Design Space Exploration of Tightly Integrated Layer-WisE Mixed-Precision Quantized Units for TinyML Inference

Xiaotian Zhao

2022/11/21
Outline

- Background
- Problem and Motivation
- Proposed Solution
- Evaluation and Results
- Method extensions
- Acknowledgment
- References
Outline

- Background
- Problem and Motivation
- Proposed Solution
- Evaluation and Results
- Method extensions
- Acknowledgment
- References
Deep convolutional neural networks have made breakthroughs in image processing, target detection, natural speech processing and other practical problems.
Layer-wise mixed-precision quantization (MPQ) has become prevailing for edge inference since it strikes a better balance between accuracy and efficiency compared to the uniform quantization scheme.
Outline

- Background
- Problem and Motivation
- Proposed Solution
- Evaluation and Results
- Method extensions
- Acknowledgment
- References
Problem and Motivation

- Please copy your related works here, honey!
Outline

- Background
- Problem and Motivation
- **Proposed Solution**
- Evaluation and Results
- Method extensions
- Acknowledgment
- References
Proposed Solution

MPQ VMM custom instructions encoding

<table>
<thead>
<tr>
<th>Code</th>
<th>vmm_4b</th>
<th>vmm_8b</th>
<th>vmm_16b</th>
<th>vmm.sd</th>
<th>vmm.ld</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1000000</td>
<td>0000000</td>
<td>1100000</td>
<td></td>
<td>imm[11:5]</td>
</tr>
<tr>
<td></td>
<td>rdh</td>
<td>rs1</td>
<td>funct3</td>
<td>rd1</td>
<td>rs2</td>
</tr>
<tr>
<td></td>
<td>1011011</td>
<td>1011011</td>
<td>1011011</td>
<td></td>
<td>1011011</td>
</tr>
</tbody>
</table>

Tightly integrated MPQ VMM AI Function unit (AFU) inside 5-stage RISC-V processor pipeline

Instruction Memory ➔ Register File ➔ Fetched Instruction ➔ Decode ➔ ALU ➔ LD/ST ➔ Writeback

MPQ VMM AFU

Generate MPQ VMM custom instructions in modified TVM

Hardware mapping

Feedback (Update Register file)

Int8 : 1x8

Int16 : 1x16

Int8 : 8x8 (addr[7:0])

Int16 : 4x4 (addr[3:0])

16x64 bit VMM AFU Memory

MPQ VMM Instructions
MPQ VMM load/store Instructions
Standard RISC-V instructions
Memory/Register File operations

Policy
Quantized Model
Layer 3
4 bit
Layer 4
8 bit
Layer 5
16 bit
Layer 6
4 bit
Proposed Solution

- Select 6-layer small-range DFS storm search
- No linear relationship between hardware overhead and model accuracy
- The effect varies when the number of quantization layers is the same
- Explore the balance of hardware consumption and model accuracy
- In-depth analysis of specific layers
Verify layer-wise property of the neural network experimentally

- the distribution of the predicted and the true values overlap highly

- complementing the preliminary experiments to validate the layer-wise property
Sample the sensitivity of accuracy and hardware cost

Propose a new metric denoted as $w$ to measure such impact

\[ \Delta BOPs = BOPs_8 - BOPs_4 \]

\[ \Delta \text{Accuracy} = \text{Accuracy}_8 - \text{Accuracy}_4 \]

\[ w = \Delta \text{BOPs} / \Delta \text{Accuracy} \]
① Apply $w$ to each layer of ResNet-18 and MobileNet-V2
② The taller bars indicate the layers that are most likely quantized in the priority queue
Proposed Solution

③ The increasing trend of each stage in line with the theoretical basis
④ We can directly make a coarse-grained quantization combination selection based on this result
Proposed Solution

- quantization selection will be performed layer by layer
- using a greedy algorithm within the hardware overhead constraint
- continuously selected and added to the quantization queue
- The final number of layers to be quantized under the constraint

Algorithm 1  \( w \)-based Greedy Design Space Search Algorithm

Input: \( \Delta \text{BOPs}, \Delta \text{Acc}, \text{MAX}_{\text{BOPs}}(\text{limit}) \)

Output: layers need quantization

1. struct layer { layer_name, BOPs, Acc, w, quant_bit }
2. for each layer do
3. \( w_i = \Delta \text{BOPs}_i / \Delta \text{Acc}_i \)
4. end for
5. sort(\( w_i \))
6. for each layer do
7. while BOPs \( \geq \) MAX_{BOPs} do
8. \( \text{BOPs} = \text{BOPs} - \Delta \text{BOPs}_i \)
9. quant_bit = 4
10. print(layer_name)
11. end while
12. end for
13. return layers being quantized (Optimal MPQ policy)
### Evaluation and Results

<table>
<thead>
<tr>
<th></th>
<th>Level</th>
<th>BOPS (GB)</th>
<th>Acc-1 (%)</th>
<th>Acc-5 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline [11]</td>
<td>FP32</td>
<td>1858</td>
<td>71.47</td>
<td>-</td>
</tr>
<tr>
<td>Greedy (Proposed)</td>
<td>high</td>
<td>90</td>
<td>70.146</td>
<td>89.384</td>
</tr>
<tr>
<td>NLP (Proposed)</td>
<td>high</td>
<td>90</td>
<td>70.146</td>
<td>89.384</td>
</tr>
<tr>
<td>HAWQ-V3 [11]</td>
<td>mid</td>
<td>72</td>
<td>64.022</td>
<td>84.724</td>
</tr>
<tr>
<td>Greedy (Proposed)</td>
<td>mid</td>
<td>71</td>
<td>67.740</td>
<td>87.892</td>
</tr>
<tr>
<td>NLP (Proposed)</td>
<td>mid</td>
<td>72</td>
<td>68.178</td>
<td>88.090</td>
</tr>
<tr>
<td>Greedy (Proposed)</td>
<td>low</td>
<td>53</td>
<td>59.156</td>
<td>81.276</td>
</tr>
</tbody>
</table>

- **Accuracy of the proposed algorithms on ResNet18.** Our solution achieves 3%-11% higher inference accuracy with similar hardware cost compared to the state of the art MPQ strategies.

- **Accuracy of the proposed algorithms on MobileNetV2.**
Results of the proposed NLP-based algorithm on ResNet-18. The proposed NLP-based method can effectively improve the accuracy on the ResNet-18 model compared to the fewer bit widths method.

Results with QAT on ResNet-18. Our solution achieves highest accuracy among the mainstream quantization algorithms on ResNet-18.
Outline

- Background
- Problem and Motivation
- Proposed Solution
- Evaluation and Results
- Method extensions
- Acknowledgment
- References
Compare the QAT and PTQ

- QAT (Quantization-aware Training) needs training for model
- PTQ (Post-training Quantization) ONLY needs inference

We define a new indicator $P$
We select some models, calculate the $p$ and plot the figure.

We can find some outliers.

So we analyzed these special models.

### Table 1: Comparison of Different Quantized Layers

<table>
<thead>
<tr>
<th>model 41</th>
<th>model 25</th>
<th>model 4</th>
<th>model 21</th>
<th>model 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>stage1.unit2.quant_convbn2</td>
<td>stage1.unit2.quant_act1</td>
<td>stage1.unit2.quant_act1</td>
<td>stage1.unit2.quant_act1</td>
<td>stage1.unit2.quant_act1</td>
</tr>
<tr>
<td>stage3.unit1.quant_convbn1</td>
<td>stage3.unit1.quant_convbn1</td>
<td>stage3.unit1.quant_act</td>
<td>stage3.unit1.quant_act</td>
<td>stage3.unit1.quant_convbn1</td>
</tr>
<tr>
<td>stage4.unit1.quant_act</td>
<td>stage4.unit1.quant_act</td>
<td>stage3.unit1.quant_convbn1</td>
<td>stage4.unit1.quant_convbn1</td>
<td>stage4.unit1.quant_convbn1</td>
</tr>
</tbody>
</table>
Conclusion
Thank you!

Contact Info

- xinfei.guo@sjtu.edu.cn
- +86-21-3420-6045
  Ext. 4191
- www.xinfeiguo.com
- Office 419, Longbin Building, Shanghai Jiao Tong University
- Lab 212-11B, Longbin Building, Shanghai Jiao Tong University

https://sites.ji.sjtu.edu.cn/icas/
Copyright Notice

This presentation in this publication was presented at tinyML® Asia 2022. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyML.org