Automatic Network Adaptation for Ultra-Low Uniform-Precision Quantization

TinyML Research Symposium ’2023

Seongmin Park, Beomseok Kwon, Jieun Lim, Kyuyoung Sim, Tae-Ho Kim and Jungwook Choi

Presenter: Myungsu Chae, CEO, Nota AI
Summary

• Uniform-precision neural network quantization has gained popularity since it simplifies densely packed arithmetic unit for high computing capability.

• However, it ignores heterogeneous sensitivity to the impact of quantization errors across the layers, resulting in sub-optimal inference accuracy.

• This work proposes a novel neural architecture search called neural channel expansion that adjusts the network structure to alleviate accuracy degradation from ultra-low uniform-precision quantization.

• The proposed method selectively expands channels for the quantization sensitive layers while satisfying hardware constraints (e.g., FLOPs, PARAMs).

• We demonstrate that the proposed method can adapt several popular networks’ channels to achieve superior 2-bit quantization accuracy on CIFAR10 and ImageNet.

• In particular, we achieve the best-to-date Top-1/Top-5 accuracy for 2-bit ResNet50 with smaller FLOPs and the parameter size.
Motivation of the Research

Compensating Performance Degradation of Ultra Low-precision Quantization

**Original Model**
- **Quantization Sensitivity**
  - Low
  - High
  - Low

**Mixed Precision Quantization**
- **Precision**
  - 2 bit
  - 4 bit
  - 2 bit

**Uniform Precision Quantization**
- **Precision**
  - 2 bit
  - 2 bit
  - 2 bit

**Proposed Method**
- **Precision**
  - 2 bit
  - 2 bit
  - 2 bit

New Channel!!
Related Work

Neural-Net adaptation for accurate DNN quantization

Preliminary Research on Capacity Expansion\[1\]

<table>
<thead>
<tr>
<th>Width</th>
<th>Precision</th>
<th>Top-1 Acc. %</th>
<th>Compute cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1x wide</td>
<td>32b A, 32b W</td>
<td>73.59</td>
<td>1x</td>
</tr>
<tr>
<td></td>
<td>1b A, 1b W</td>
<td>60.54</td>
<td>0.03x</td>
</tr>
<tr>
<td>2x wide</td>
<td>4b A, 8b W</td>
<td>74.48</td>
<td>0.74x</td>
</tr>
<tr>
<td></td>
<td>4b A, 4b W</td>
<td>74.52</td>
<td>0.50x</td>
</tr>
<tr>
<td></td>
<td>4b A, 2b W</td>
<td>73.58</td>
<td>0.39x</td>
</tr>
<tr>
<td></td>
<td>2b A, 4b W</td>
<td>73.50</td>
<td>0.39x</td>
</tr>
<tr>
<td></td>
<td>2b A, 2b W</td>
<td>73.32</td>
<td>0.27x</td>
</tr>
<tr>
<td></td>
<td>1b A, 1b W</td>
<td>69.85</td>
<td>0.15x</td>
</tr>
<tr>
<td>3x wide</td>
<td>1b A, 1b W</td>
<td>72.38</td>
<td>0.30x</td>
</tr>
</tbody>
</table>

Channel Splitting to Reduce Dynamic Range of Weights\[2\]

However, above methods are not automatic and not plausible on the ultra-low precision.

---


Neural Channel Expansion

Algorithm 1: Neural Channel Expansion

Input:
- Split the training set into two dis-joint sets: \( D_{\text{weight}} \) and \( D_{\text{arch}} \) (\( n(D_{\text{weight}}) = n(D_{\text{arch}}) \))
- Search Parameter: \( [a_1, a_2, ... a_n] \in \mathbb{A}^L \), \( \mathbb{A} = \{A_1, A_2, ... A_L\} \)

1. For Warm-up Epoch do
2. Sample batch data \( D_w \) from \( D_{\text{weight}} \) and network from \( \mathbb{A} \sim U(0,1) \)
3. Calculate \( \text{Loss}_{\text{weight}} \) on \( D_w \) to update network weights
4. End for
5. For Search Epoch do
6. Sample batch data \( D_w \) from \( D_{\text{weight}} \) and network from \( \text{Softmax}(\mathbb{A}) \)
7. Calculate \( \text{Loss}_{\text{weight}} \) on \( D_w \) to update network weights
8. Sample batch data \( D_a \) from \( D_{\text{arch}} \) and network from \( \text{Softmax}(\mathbb{A}) \)
9. Calculate \( \text{Loss}_{\text{arch}} \) on \( D_a \) to update \( \mathbb{A} \)
10. For layer do
11. \( j \leftarrow \#a_i \)
12. If \( \text{Softmax}(a_j) \geq T \) do
13. Expand search space \( (\delta_{j+1}) \)
14. \( a_{j+1} \leftarrow a_j \) \# copy search parameter
15. End if
16. End for
17. End for
18. Derive the searched network from \( \mathbb{A} \)
19. Randomly initialize the searched network and optimize it on the training set

Warm-up training
Weights update
Architecture Params update

If the architecture param of max channel exceeds the threshold, expand the search space.
Analysis on the Impact of Channel Expansion to Quantization

Quantization affects to the dynamic range of the weights

- Quantization applied to a given network substantially increases the dynamic range of activation, hindering successful DNN quantization
- Unlike straightforward capacity to all layers as described in WRPN, we selectively expand layers so we can keep lower PARAMs while preserving accuracy.
Analysis on Search Space of Neural Channel Expansion

The flexible search space of NCE can lead efficient search compared to TAS

• When we sample the network from the search space randomly, architecture searched from NCE and TAS, then the accuracy of the networks are compared. NCE shows the higher accuracy, it means that search space and the method is superior to others.
Analysis on Channel Selection Preference of NCE

Gradient shows the preference on the larger channel numbers

- Architecture parameters associated to large channel get negative gradient at the early stage of training, but the demand is decreased as training goes.
Experimental Results

### Benchmark with other SOTA models

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-1 Acc(%)</th>
<th>Top-5 Acc(%)</th>
<th>FLOPs</th>
<th>PARAMs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Full precision</strong></td>
<td>70.56</td>
<td>89.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSQ</td>
<td>67.6</td>
<td>87.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QIL</td>
<td>65.7</td>
<td>-</td>
<td>1.814G</td>
<td>11.69M</td>
</tr>
<tr>
<td>PACT</td>
<td>64.4</td>
<td>85.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EdMIPS</td>
<td>65.9</td>
<td>86.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/o NCE(Ours)</td>
<td>64.08</td>
<td>86.47</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w/ NCE(Ours)</td>
<td><strong>66.18</strong></td>
<td><strong>86.75</strong></td>
<td><strong>1.897G</strong></td>
<td><strong>10.94M</strong></td>
</tr>
</tbody>
</table>

### Impact of Threshold

<table>
<thead>
<tr>
<th>Threshold $T$</th>
<th>0.30</th>
<th>0.25</th>
<th>0.20</th>
<th>0.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>91.60%</td>
<td>91.44%</td>
<td>91.40%</td>
<td>90.77%</td>
</tr>
</tbody>
</table>

### On the various precision

<table>
<thead>
<tr>
<th>ResNet50-CIFAR10</th>
<th>Accuracy</th>
<th>FLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original structure (w/o NCE)</td>
<td>W32A32</td>
<td>92.88%</td>
</tr>
<tr>
<td>W3A3</td>
<td>92.45%</td>
<td></td>
</tr>
<tr>
<td>W4A4</td>
<td>92.69%</td>
<td></td>
</tr>
<tr>
<td>NCE</td>
<td>W3A3</td>
<td>92.66%</td>
</tr>
<tr>
<td>W4A4</td>
<td>92.75%</td>
<td></td>
</tr>
</tbody>
</table>
Analysis on Comparison of Network Structures

NCE balances the reduction and expansion of the ResNet50
Conclusion

• In this work, we propose a novel approach that explores the neural network structure to achieve robust inference accuracy while using the simple uniform-precision arithmetic operations.

• Our novel differentiable neural architecture search, called neural channel expansion, employs the search space that can shrink and expand the channels.

• More sensitive layers can be equipped with more channels while the overall resource requirements (e.g., FLOPs and PARAMs) are maintained.

• We demonstrate that the proposed method can achieve superior performance in ultra-low uniform-precision quantization for CIFAR10 and ImageNet networks.
Copyright Notice

This presentation in this publication was presented at the tinyML® Research Symposium (March 27, 2023). 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