Cell Division: Weight Bit-Width Reduction Technique for Convolutional Neural Network Hardware Accelerators

Hanmin Park, Kiyoung Choi

Neural Processing Research Center
Design Automation Laboratory
Seoul National University
1. Motivation
   • Mismatch b/w two research communities: CNN inference bit-width reduction CNN inference HW accelerator design

2. Elaboration
   • Cell division technique applied to: Fully connected layer Convolutional layer

3. Discussion
   • How to suppress the number of new neurons
   • Applicability of cell division technique to recent researches

4. Conclusion
Motivation

• Data type for CNN inference HW accelerators
  ◦ Fixed-point format than floating-point format

• Design parameters of fixed-point format
  ◦ ulp (unit in the last place): Once decided, it is implicitly assumed throughout the computation.
  ◦ BW (bit-width): Largely affects chip-area, power, etc.
    ▪ In trade-off relation w/ CNN accuracy.
    ▪ Reduction efforts in two research communities.
Motivation

CNN inference bit-width reduction

• Inter-network BW opt
  (P. Judd et al., arXiv 2015)
  ◦ AlexNet on ImageNet:
    ▪ 10-bit weights
  ◦ GooLeNet on ImageNet:
    ▪ 9-bit weights

• Intra-network BW opt
  (D. Lin et al., ICML 2016)
  ◦ AlexNet-like CNN on ImageNet (5 conv layers):
    ▪ $\beta, \beta - 5, \beta - 4, \beta - 5$, and $\beta - 4$ bit weights

CNN inference HW accelerator design

• 16-bit weights:
  ◦ DaDianNao
  ◦ Eyeriss
  ◦ Stripes etc.

• 8-bit weights:
  ◦ TPU v1

Very (too) Pessimistic!
Motivation

CNN inference bit-width reduction

- Inter-network BW opt  
  (P. Judd et al., arXiv 2015)
  - AlexNet on ImageNet:
    - 10-bit weights
  - GoLeNet on ImageNet:
    - 9-bit weights
- Intra-network BW opt  
  (D. Lin et al., ICML 2016)
  - AlexNet-like CNN on ImageNet (5 conv layers):
    - $\beta$, $\beta - 5$, $\beta - 4$, $\beta - 5$, and $\beta - 4$ bit weights

CNN inference HW accelerator design

- 16-bit weights:
  - DaDianNao
  - Eyeriss
  - Stripes

We want to:
- Alleviate the pessimism.
- Make CNNs executable.

Very (too) Pessimistic!
1. Motivation
   • Mismatch b/w two research communities:
     CNN inference bit-width reduction
     CNN inference HW accelerator design

2. Elaboration
   • Cell division technique applied to:
     Fully connected layer
     Convolutional layer

3. Discussion
   • How to suppress the number of new neurons
   • Applicability of cell division technique to recent researches

4. Conclusion
Main Idea

• Start w/ a fixed-point quantized CNN:
  
  A quantization result (w/ 0.3 %p test accuracy drop):
  • \((\text{BW, ulp}) = (7, 2^{-6})\)
  • Range = \([-1, 1]\)

  Let’s assume we only have a HW accelerator that assumes 6-bit weights.
  \(\rightarrow\) Not executable w/o CNN accuracy drop.

  \(\text{float Wgt distribution of LeNet-5.conv1.}\)

• Cell division technique:
  
  \[a_{in} \cdot w = a_{in} \cdot \sum_i w'_i = \sum_i a_{in} \cdot w'_i\]
  
  where \(w \in [-1,1)\) and \(w' \in [-0.5, 0.5)\).

  • We target no specific HW support for this technique.
Cell Division for Fully Connected Layer

4-bit weights $\in [-8, 7]$

$$\begin{bmatrix}
0010 & 1100 \\
1110 & 1010 \\
0001 & 0011
\end{bmatrix}$$

$$\begin{bmatrix}
2 & -4 \\
-2 & -6 \\
1 & 3
\end{bmatrix}$$

$$1010 \cdot a_{in} - b$$

3-bit weights $\in [-4, 3]$

$$\begin{bmatrix}
010 & 100 \\
110 & 100 \\
001 & 011 \\
000 & 110
\end{bmatrix}$$

$$\begin{bmatrix}
2 & -4 \\
-2 & -4 \\
1 & 3 \\
0 & -2
\end{bmatrix}$$

$$(100 + 110) \cdot a_{in} - b = (100 \cdot a_{in} - b) + (110 \cdot a_{in} - 0)$$
**Cell Division for Fully Connected Layer**

- Part of the act-dup layer is the identity matrix.
- No HW modification req (w/ performance overhead).
- Biases of the neurons in the act-dup layer are all zero.
4-bit weights $\in [-8, 7]$  

3-bit weights $\in [-4, 3]$

Input feature map's channel direction.

$\exists$ 2 filters, originally.

A new filter is added.

$$0101 \cdot a_{in} - b$$

$$= (011 \cdot a_{in} - b) + (010 \cdot a_{in} - 0)$$
Part of the chn-fusing fltrs is the identify filters.

No HW modification req (w/ performance overhead).

Biases of the neurons in the chn-fusing fltrs are all zero.
Experimental Results

The Best Fixed-Point Quantization for Each CNN
LE: LSB’s Exp ($\log_2$ ulp)
BW: Bit-Width

<table>
<thead>
<tr>
<th>CNN</th>
<th>LE</th>
<th>BW</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet-300-100</td>
<td>-4, -4, -3</td>
<td>4, 4, 4</td>
</tr>
<tr>
<td>LeNet-5</td>
<td>-6, -4, -5, -5</td>
<td>7, 4, 3, 5</td>
</tr>
<tr>
<td>AlexNet</td>
<td>-8, -9, -9, -10, -9, -9, -9</td>
<td>8, 9, 9, 10, 9, 6, 7, 7</td>
</tr>
<tr>
<td>VGG-16</td>
<td>-7, -6, -8, -8, -8, -8, -9, -9, -9, -9, -9, -8, -8</td>
<td>8, 6, 8, 8, 9, 9, 8, 8, 8, 9, 8, 9, 5, 5, 5</td>
</tr>
</tbody>
</table>

Weight storage requirements according to cell-division’s target bit-widths (normalized to those of 16-bit fixed-point quantized CNNs).
1. Motivation
   • Mismatch b/w two research communities:
     CNN inference bit-width reduction
     CNN inference HW accelerator design

2. Elaboration
   • Cell division technique applied to:
     Fully connected layer
     Convolutional layer

3. Discussion
   • How to suppress the number of new neurons
   • Applicability of cell division technique to recent researches

4. Conclusion
Contents NOT in the Paper

FAQs
How to Suppress the Number of New Neurons

Weight Distribution

- Weight distribution of LeNet-5.conv1 (LE, BW) = (-6, 7)

- But this weight distribution characteristic per se is not enough!

The second reasoning is about my technique for further reducing # new neurons.

Only small portion of weights get cell-divisioned (9 out of 500 wgts).
How to Suppress the Number of New Neurons

One Cell Div w/ Multiple Synapse Divs

Cell Division Technique
Applicability to Recent Researches

• “Why are you referring to ancient (2015, 2016) quantization schemes?”
  ◦ “AlexNet on ImageNet is successful w/ only 3-bit wgts.”

• We wanted our approach to be as generic as possible.
  ◦ Basic
    ▪ Float training first → Fixed-point quantization, next.
  ◦ Advanced
    ▪ Fixed-point quantization during training.
    ▪ Mixed usage of float & fixed-point (C. Leng et al., AAAI 2018).

• TMI: We were very strict about DNN accuracy drop due to fixed-point quantization in the paper.
  ◦ 0.1 % training accuracy & 0.3 % test accuracy drop.
Applicability to Recent Researches

• Weight quantization levels (3 bits per weight):
  ◦ {-2, +2} → {-2, -1, 0, +1, +2}
  ◦ {-4, +4} → {-4, -2, -1, 0, +1, +2, +4}

• W/ layer-wise floating-point scaling factors

• Mathematical formulation as a mixed integer programs (MIP) enables:

<table>
<thead>
<tr>
<th>Accuracy</th>
<th>{-2, +2}</th>
<th>{-4, +4}</th>
<th>Full Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-1</td>
<td>0.592</td>
<td>0.600</td>
<td>0.600</td>
</tr>
<tr>
<td>AlexNet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top-5</td>
<td>0.818</td>
<td>0.822</td>
<td>0.824</td>
</tr>
</tbody>
</table>

• Note that our technique can be applied here.
  ◦ {-4, +4} → {-2, +2} w/ more accuracy.
  ◦ Or no shift operations required at all.

Uses shift operations instead of multiplications.
1. Motivation
   • Mismatch b/w two research communities:
     CNN inference bit-width reduction
     CNN inference HW accelerator design

2. Elaboration
   • Cell division technique applied to:
     Fully connected layer
     Convolutional layer

3. Discussion
   • How to suppress # new neurons
   • Applicability of cell division technique to recent researches

4. Conclusion
Conclusion

• We proposed the **cell division technique**, which:
  ◦ Can reduce the fixed-point bit-width of CNN weights w/o any accuracy change.

• We also proposed the **activation duplication layer & channel fusing filters** for legacy CNN inference HW accelerators.

• The cell division technique enables:
  ◦ **Alleviating the pessimism** behind the weight bit-width selection when designing CNN inference HW accelerators.
  ◦ **Making CNNs executable** on CNN inference HW accelerator which assumes narrower weight bit-width.
THANK YOU