Pooling Acceleration in the DaVinci Architecture Using Im2col and Col2im Instructions

Caio S. Rohwedder,
João L. P. de Carvalho, José Nelson Amaral, Guido Araújo, Giancarlo Colmenares, Kai-Ting Amy Wang
Overview

- AI Accelerators
  - High demand for CNNs
  - DaVinci
    - Matrix Multiplier
    - Im2col & Col2im
Overview

- AI Accelerators
  - High demand for CNNs
  - DaVinci
    - Matrix Multiplier
    - Im2col & Col2im

- Proposition:
  - Im2col/Col2im Based Pooling
    Using Specialized DaVinci Instructions
Background

- Convolution & Im2col
- Col2im & Backward Operators
- Pooling Operators
Convolution - Input Image

Input: In

Ih

Kh

Kw

IW
Convolution - Input Kernel
Im2col: Convolution $\rightarrow$ GEMM
Col2im & Backward Operators

- Propagates Im2col
  - Backward
Col2im & Backward Operators

- Propagates Im2col

Input: In

OutIn

Col2im

$O_{h} \times O_{w}$

$C \times K_{h} \times K_{w}$
Overlapping Im2col & Col2im

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td></td>
</tr>
</tbody>
</table>
Overlapping Im2col & Col2im

- Im2col: Duplicates

![Im2col Diagram]

![Col2im Diagram]
Overlapping Im2col & Col2im

- **Im2col**: Duplicates
- **Col2im**: Merges
Pooling Operators

- Maxpool
  - Max Value
Pooling Operators

- Maxpool
  - Max Value
The DaVinci Architecture

- AI Core
- Fractal Memory Layout
- Im2col & Col2im Instruction
DaVinci - AI Core

- Cube Unit
  - Matrix Multiplication
- Vector Unit
  - Pooling
- Scalar Unit
DaVinci - AI Core

- Scratch-Pad Memories
DaVinci - AI Core

- Scratch-Pad Memories
- SCU: Storage Conversion Unit
- Global Memory
DaVinci - Fractal Memory Layout

NCHW → NC₁HWC₀

- C₁ = ⌈C/C₀⌉
- C₀ = 16
  - (Float16)
Im2col Instruction

- Loads fractals
  - \( \text{NC}_1 \text{HWC}_0 \)
Im2col Instruction

- Loads fractals
  - $NC_1 HWC_0$
Im2col Instruction

- Loads fractals
  - $NC_1 HWC_0$
Col2im Instruction

- Inputs are fractals
  - $\text{NC}_1 \text{HWC}_0$
Col2im Instruction

- Inputs are fractals
  - $NC_1 HWC_0$
Col2im Instruction

- Inputs are fractals
  - \( NC_1 HWC_0 \)
Col2im Instruction

- Inputs are fractals
  - $\text{NC}_1 \text{HWC}_0$
Col2im Instruction

- Inputs are fractals
  - NC₁HWC₀
Im2col & Col2im Based Pooling
Forward Pooling

Input UB → Maxpool UB → Mask UB

Input GM → Load → Maxpool UB → Store → Maxpool GM

Mask UB → Store → Mask GM
Forward Pooling

Input UB

Maxpool UB

Mask UB

Input GM

Maxpool GM

Mask GM
Forward Pooling

Input UB → Maxpool UB → Mask UB

Load → Maxpool GM → Mask GM

Diagram showing the flow of data through different units and buffers.
Forward Pooling
Forward Pooling

- Input UB
  - Load
  - Maxpool
  - Maxpool UB
  - Store
  - Maxpool GM

- Mask UB
  - Store
  - Mask GM

13 19
Forward Pooling + Mask

Input UB → Maxpool → Maxpool UB → Mask UB → Mask GM

Input GM → Load → Maxpool UB → Store → Maxpool GM

Mask UB → Store → Mask GM

Diagram showing various buffer and unit connections, including LOC, Cube Unit, Vector Unit, Scalar Unit, L0B Buffer, L0B Buffer, Unified Buffer, HBM, DDR, L2 Buffer, and Global Memory.
Forward Pooling + Mask
Im2col-Based Forward Pooling (+ Mask)
Backward Pooling

Mask UB
  ↓ Load
  ↓ Mask GM
Gradient UB
  ↓ Load
  ↓ Gradient GM
Output_Mask UB
  Merge
  ↓ Output UB
  ↓ Store
  ↓ Output GM
Backward Pooling
Backward Pooling

Mask UB

Load

Mask GM

Gradient UB

Load

Gradient GM

Output Mask UB

Merge

Output UB

Store

Output GM
Backward Pooling

![Diagram of Backward Pooling process]
Backward Pooling

Mask UB
Load
Mask GM

Gradient UB
Load
Gradient GM

Output_Mask UB

Output UB
Store
Output GM

Merge

Matrix:
0 0 0 0 0
0 0 0 Y 0
0 0 X 0 0

Diagram showing various units and buffers, including LOC Buffer, Cube Unit, Vector Unit, Scalar Unit, LOB Buffer, Unified Buffer, and AI Core.
Backward Pooling

- Mask UB
  - Load
    - Mask GM
- Gradient UB
  - Load
    - Gradient GM
- Output_Mask UB
- Merge
- Output UB
  - Store
    - Output GM

Matrix:

```
0 0 0 0 0
0 0 0 Y 0
0 0 X 0 0
```
Col2im-Based Backward Pooling
Avgpool
Avgpool

Diagram showing the flow of data from Mask UB, Gradient UB, and Output_Mask UB to Output UB, with operations such as Load and Store.
Experimental Evaluation

- InceptionV3 Comparison
- Stride Tests
## Maxpool Inputs in CNNs

<table>
<thead>
<tr>
<th>CNN</th>
<th>Input 1</th>
<th>Input 2</th>
<th>Input 3</th>
<th>Input 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>InceptionV3</td>
<td>147,147,64</td>
<td>71,71,192</td>
<td>35,35,288</td>
<td>17,17,768</td>
</tr>
<tr>
<td>Xception</td>
<td>147,147,128</td>
<td>74,74,192</td>
<td>37,37,728</td>
<td>19,19,1024</td>
</tr>
<tr>
<td>Resnet50</td>
<td>112,112,64</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VGG16</td>
<td>224,224,64</td>
<td>112,112,64</td>
<td>56,56,256</td>
<td>28,28,512</td>
</tr>
</tbody>
</table>
InceptionV3 - Maxpool Forward

- Maxpool
- Maxpool with Im2col

Cycle Count

Input Shape

(1, 48, 17, 17, 16)
(1, 18, 35, 35, 16)
(1, 12, 71, 71, 16)

Maxpool + Mask
Maxpool + Mask with Im2col

Cycle Count

Input Shape

(1, 48, 17, 17, 16)
(1, 18, 35, 35, 16)
(1, 12, 71, 71, 16)
InceptionV3 - Maxpool Backward

![Graph showing cycle count for different input shapes.](image_url)
Stride Tests - Speedup

Stride = (2,2)

Stride = (3,3)
Stride Tests - Slowdown

Stride = (1,1)

- Maxpool
- Maxpool with Im2col
- Maxpool with expansion

Cycle Count

Input Height and Width
Stride Tests - XY Split

Stride = (2,2)
Conclusion

- DaVinci’s Im2col & Col2im Instructions
- Im2col/Col2im Based Pooling in DaVinci
- Experimental Evaluation
  - Speedups up to 5.8x
  - Improvements for all but (1,1) stride
  - Superior to the X-Y split
Thank you!

Caio Salvador Rohwedder
c157754@dac.unicamp.br