neuFlow: A Dataflow Architecture for Vision

Clément Farabet, Yann LeCun

joint work with:
Yann LeCun, Laurent Najman, Marco Scoffier, Srinivas Turaga
Eugenio Culurciello, Berin Martini, Polina Akselrod, Darko Jelaca,
A Runtime Reconfigurable Dataflow Architecture

grid of passive processing tiles (PTs) [x20 on a Virtex6 LX240T]

multi-port memory controller (DMA) [x12 on a V6 LX240T]

RISC CPU, to reconfigure tiles and data paths, at runtime

global network-on-chip to allow fast reconfiguration

neuFlow: Architecture
neuFlow: Processing Tile (PT) Structure

term-by-term streaming operators (MUL, DIV, ADD, SUB, MAX)

configurable bank of FIFOs, for stream buffering, up to 10kB per PT

configurable router, to stream data in and out of the tile, to neighbors or DMA ports

full 1/2D parallel convolver with 100 MAC units

configurable piece-wise linear or quadratic mapper

[Virtex6 LX240T]
luaFlow: A Dataflow Compiler

a home-grown compiler that compiles ConvNets and the likes to sequences of grid reconfigurations (e.g. neuFlow bytecode)
luaFlow: A Dataflow Compiler

high-level (functional) description

```python
net = nn.Sequential()
net:add(nn.SpatialConvolution(3,6,9,9))
net:add(nn.Tanh())
net:add(nn.SpatialSubSampling(6,4,4))
net:add(nn.SpatialConvolution(6,12,9,9))
net:add(nn.SpatialLinear(12,6))
```

(Torch5 code)
luaFlow: A Dataflow Compiler

2/5

infer a flow-graph model from the user description
luAFlow: A Dataflow Compiler

3/5

divide the graph into subgraphs that fit on the grid
for each subgraph, generate the routes and configs for each PT and DMA port

once configured, data streams ripple through the grid, the grid is “passive”
5/5

global optimization: instruction reordering

configuration cycles

data streaming cycles
luaFlow: Supported Operations

Coding: Q8.8 (16bit, fixed-point)

- 1D convolution
- 2D convolution
- local pooling/subsampling/histogramming (max, average, weighted)
- term-by-term div/add/sub/mul/muladd
- point-wise non-linear mapping
- local contrast normalization
- temporal difference
- ...


## Profiling

<table>
<thead>
<tr>
<th></th>
<th>Intel 2Core</th>
<th>neuFlow Virtex4</th>
<th>neuFlow Virtex 6</th>
<th>nVidia GT335m</th>
<th>neuFlow IBM 45nm</th>
<th>nVidia GTX480</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak GOP/sec</td>
<td>10?</td>
<td>40</td>
<td>160</td>
<td>182</td>
<td>1280</td>
<td>1350</td>
</tr>
<tr>
<td>Actual GOP/sec</td>
<td>1.1</td>
<td>37</td>
<td>147</td>
<td>54</td>
<td>1164</td>
<td>294</td>
</tr>
<tr>
<td>FPS</td>
<td>1.4</td>
<td>46</td>
<td>182</td>
<td>67</td>
<td>1456</td>
<td>374</td>
</tr>
<tr>
<td>Power (W)</td>
<td>30</td>
<td>10</td>
<td>10</td>
<td>30</td>
<td>5</td>
<td>220</td>
</tr>
<tr>
<td>Embed? (GOP/s/W)</td>
<td>0.03667</td>
<td>3.7</td>
<td>14.7</td>
<td>1.8</td>
<td>232.8</td>
<td>1.33636</td>
</tr>
</tbody>
</table>

* computing a 16x10x10 filter bank over a 4x500x500 input image
## Resources

<table>
<thead>
<tr>
<th></th>
<th>neuFlow Virtex4</th>
<th>neuFlow Virtex 6</th>
<th>neuFlow IBM 45nm 3x3mm</th>
<th>neuFlow IBM 45nm 6x6mm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak GOP/sec</td>
<td>40</td>
<td>160</td>
<td>320</td>
<td>1280</td>
</tr>
<tr>
<td>Sys+DDR Frequency MHz</td>
<td>200</td>
<td>200</td>
<td>400</td>
<td>400</td>
</tr>
<tr>
<td>DDR Bdwdth GB/s (pins)</td>
<td>0.8 (16)</td>
<td>3 (64)</td>
<td>6 (64)</td>
<td>24 (256)</td>
</tr>
<tr>
<td>MACs #avail (#used)</td>
<td>192 (109)</td>
<td>680 (436)</td>
<td>436 (all)</td>
<td>1744 (all)</td>
</tr>
<tr>
<td>Tiles #avail</td>
<td>4</td>
<td>20</td>
<td>20</td>
<td>80</td>
</tr>
</tbody>
</table>
Application: Scene Parsing

dense labeling of natural images
multiscale ConvNet, trained end-to-end to optimize a dual term energy: a segmentation loss and a pixelwise classification loss
Application: Scene Parsing
Application: Scene Parsing

Live Demo.
thank you

www.neuflow.org