FPGA-based Accelerator for Long Short-Term Memory Recurrent Neural Networks

Yijin Guan\textsuperscript{1}, Zhihang Yuan\textsuperscript{1}, Guangyu Sun\textsuperscript{1,3}, Jason Cong\textsuperscript{2,3,1}

\textsuperscript{1}Center for Energy-Efficient Computing and Applications, Peking University, China
\textsuperscript{2}Computer Science Department, University of California, Los Angeles, USA
\textsuperscript{3}PKU/UCLA Joint Research Institute in Science and Engineering
Deep Learning

- Scenarios
- Applications
**Recurrent Neural Network**

Feed-forward NN

RNN
Recurrent Neural Network

RNN unfolds into a DNN over time
$$i_t = \sigma (W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$
Long Short-Term Memory

\[ f_t = \sigma (W_{xf} x_t + W_{hf} h_{t-1} + b_f) \]
\[ \tilde{c}_t = \tanh (W_{xc} x_t + W_{hc} h_{t-1} + b_c) \]
Long Short-Term Memory

$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$
Long Short-Term Memory

\[ o_t = \sigma (W_{xo} x_t + W_{ho} h_{t-1} + b_o) \]
Long Short-Term Memory

\[ x_t \quad h_{t-1} \]

\[ \begin{align*}
    \text{Input Gate} & \quad \text{Forget Gate} \\
    W_{xi} & \quad W_{xf} & \quad W_{xo} \\
    \text{Cell Gate} & \quad \text{Output Gate} \\
    W_{xc} & \quad W_{wh} & \\
    \tilde{c}_t & \quad \tanh(\tilde{c}_t) \\
    i_t & \quad f_t & \quad o_t \\
    c_t & \quad h_t
\end{align*} \]

\[ h_t = o_t \odot \tanh(c_t) \]
Why FPGA

CPU

GPU

ASIC

FPGA
Design Challenges and Optimizations

Off-chip Memory

FPGA Chip

Computation Engine

Data Buffers

12
Design Challenges and Optimizations

- Computation Resources & Performance
  - Loop Unroll
  - Deep Pipeline
Design Challenges and Optimizations

- On-chip Memory Resources
  - Loop Tiling
  - Eclectic Data Partition
Design Challenges and Optimizations

- Bandwidth
  - Ping-pong Buffers
  - Reshaping Data Layout

FPGA Chip

Computation Engine

Data Buffers

Off-chip Memory
System Design

- Vivado High-level Synthesis (v2015.4)
- Vivado Design Suite (v2015.4)
Accelerator Design

Output Group 0

Output Group 1

LSTM Functional Logic

Cell Buffer

Input Group 0

Input Group 1

i

f

c

o
Experimental Results

<table>
<thead>
<tr>
<th>Device</th>
<th>Model</th>
<th>Freq.</th>
<th>Development Env.</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Xeon E5-2430</td>
<td>2.20 GHz</td>
<td>gcc -O3 &amp; OpenMP</td>
</tr>
<tr>
<td>FPGA</td>
<td>Xilinx Virtex7-485t</td>
<td>150 MHz</td>
<td>Vivado Design Suite</td>
</tr>
</tbody>
</table>
# Experimental Results

<table>
<thead>
<tr>
<th>Imp.</th>
<th>Model</th>
<th>Freq.</th>
<th>Data Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous Imp.</td>
<td>Xilinx Zynq7020</td>
<td>142MHz</td>
<td>Fixed-16</td>
</tr>
<tr>
<td>Our Imp.</td>
<td>Xilinx Virtex7-485t</td>
<td>150 MHz</td>
<td>Float-32</td>
</tr>
</tbody>
</table>

![Speedup Graph]

- Previous Imp. A: ~15.5x
- Previous Imp. B: 2x
- Our Imp. A: ~47%
- Our Imp. B: ~15x

The table above summarizes the implementation details and performance metrics for the Xilinx Zynq7020 and Xilinx Virtex7 devices. The speedup values indicate significant improvements in performance for the new implementation compared to previous approaches.
Future Work

- Data quantization
  - Low-precision fixed-point numbers

- Model compression
  - Connection Pruning
  - Matrix compression (e.g. SVD)

- General architecture
  - Support for all LSTM variants
Conclusions

- An accelerator for LSTM-RNN

- Optimizations for computation & communication at architecture level

- On-board implementation with high-performance computation engines & data dispatcher

- Outperforms CPU- & other FPGA- implementations
Thank You

Q & A