Caffe to Zynq: State-of-the-Art Machine Learning Inference Performance in Less Than 5 Watts

Vinod Kathail, Distinguished Engineer
May 24, 2017
Agenda

- Why Zynq SoCs for Deep Learning Inference
- Caffe to Zynq SoC in Seconds
- A Full System Example
Diverse Applications with Diverse Design Targets

Translate & AlphaGo:
Huge networks

Medical Diagnosis:
Small networks

Robotics:
Real-time

ADAS
High accuracy
Low latency

Hearing Aids:
Small network
Low latency

© Copyright 2017 Xilinx
Zynq Offers the Most Efficient Deep Learning Inference
# Zynq SoCs Offer Superior Throughput, Latency

<table>
<thead>
<tr>
<th>GoogLeNet @ batch = 1</th>
<th>Xilinx ZU9</th>
<th>Xilinx ZU5</th>
<th>eGPU*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images/s</td>
<td>370.0</td>
<td>155.0</td>
<td>70</td>
</tr>
<tr>
<td>Power (W)</td>
<td>7.0</td>
<td>4.5</td>
<td>7.9</td>
</tr>
<tr>
<td>Images/s/watt</td>
<td>53.0</td>
<td>34.5</td>
<td>8.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GoogLeNet @ batch = 1</th>
<th>Xilinx ZU9</th>
<th>Xilinx ZU5</th>
<th>eGPU*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images/s</td>
<td>370.0</td>
<td>155.0</td>
<td>70</td>
</tr>
<tr>
<td>Latency (ms)</td>
<td>2.7</td>
<td>6.4</td>
<td>14.2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GoogLeNet @ batch = 8</th>
<th>Xilinx ZU9</th>
<th>Xilinx ZU5</th>
<th>eGPU*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images/s</td>
<td>370.0</td>
<td>155.0</td>
<td>163</td>
</tr>
<tr>
<td>Latency (ms)</td>
<td>2.7</td>
<td>6.4</td>
<td>49.0</td>
</tr>
</tbody>
</table>

For large batch, CPU/GPU/DSPs latency increases significantly


© Copyright 2017 Xilinx
**The Divergence of Training and Inference**

**Training**: Process for machine to “learn” and optimize model from data

**Inference**: Using trained models to predict/estimate outcomes from new observations in efficient deployments

---

**Table:**

<table>
<thead>
<tr>
<th>Model</th>
<th>FP-32</th>
<th>FIXED-16 (INT16)</th>
<th>FIXED-8 (INT8)</th>
<th>Difference vs FP32</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>86.6%</td>
<td>86.6%</td>
<td>86.4%</td>
<td>(0.2%)</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>88.6%</td>
<td>88.5%</td>
<td>85.7%</td>
<td>(2.9%)</td>
</tr>
<tr>
<td>SqueezeNet</td>
<td>81.4%</td>
<td>81.4%</td>
<td>80.3%</td>
<td>(1.1%)</td>
</tr>
</tbody>
</table>

---

Inference now 8 bit and below for maximum efficiency

**Image Sources**:

© Copyright 2017 Xilinx
Inference Precisions Moving to Lower and Variable Precision

Convolution Layers

Fully Connected Layers

Citation: https://arxiv.org/pdf/1510.00149.pdf
Future Proof Architecture for Any Precisions

Limited to 32 bit operations
SIMD operations at INT8/16

New devices required to support change in precision efficiently

Reconfigurable to scale and optimize for different precisions
BNN: Unparalleled Performance

- Reducing precision from 8b to 1b shrinks LUT cost by 40x
- Potential to scale CNN performance to above 23TOPS (ZU9)

<table>
<thead>
<tr>
<th></th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
</tbody>
</table>

- Assuming 300 MHz with 90%/70% DSP/LUT utilizations
- Resource consumption assumption: 2.5 LUTs/op (INT1), 16 LUTs/op (INT4), 0.25 DSP/op (INT8)
BNN: Unparalleled Performance

- Reducing precision from 8b to 1b shrinks LUT cost by 40x
- Potential to scale CNN performance to above 23TOPS (ZU9)

<table>
<thead>
<tr>
<th>1b</th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
<th>1b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
<td>1b</td>
</tr>
</tbody>
</table>

- Assuming 300 MHz with 90%/70% DSP/LUT utilizations
- Resource consumption assumption: 2.5 LUTs/op (INT1), 16 LUTs/op (INT4), 0.25 DSP/op (INT8)

© Copyright 2017 Xilinx
BNN: Unparalleled Performance

Reducing precision from 8b to 1b shrinks LUT cost by 40x

Potential to scale CNN performance to above 23TOPS (ZU9)

- Assuming 300 MHz with 90%/70% DSP/LUT utilizations
- Resource consumption assumption: 2.5 LUTs/op (INT1), 16 LUTs/op (INT4), 0.25 DSP/op (INT8)
- 10W power assumption on ZU9

© Copyright 2017 Xilinx
Small degradation in accuracy but fast improving

Top-5 Error (ImageNet)

8bits to 1bit: What is the Challenge?
Low Latency Inference by Layer to Layer Dataflow On Chip

Nvidia Tegra X1 (GPU Regfile + L2)
- 6 Mb

Xilinx ZU7 (BRAM + URAM)
- 38 Mb

Up to 6x More On-chip Memory than SoCs and eGPUs

Frameworks

Libraries and Tools

Development Kits

OpenVX integration

Caffe

OpenCV

DNN

CNN

GoogLeNet

SSD

FCN ...

© Copyright 2017 Xilinx
xFdnn: Direct Deep Learning Inference from Caffe

1. Import .prototxt and trained weights
   - Caffe Prototxt
   - Filter Weights
   - Target Data

2. Call prototxt runtime API in your application
   - Zynq Ultrascale+ MPSoC
   - C/C++ Application
   - Pre-optimized CNN Engine
   - Programmable Logic
   - Conv1
   - Pool1
   - FC

3. Cross-compile for Cortex-A53 and run on a board

Compiles only ARM software code in minutes. No hardware compilation.

© Copyright 2017 Xilinx
Caffe Prototxt to Zynq

```
name: "CaffeNet"
layer {
  name: "data"
type: "Input"
top: "data"
input_param { shape: {
dim: 10 dim: 3 dim: 227
dim: 227 }}
}
layer {
  name: "conv1"
type: "Convolutions"
bottom: "data"
top: "conv1"
convolution_param {
  num_output: 96
  kernel_size: 11
  stride: 4
}
}
layer {
  name: "relu1"
type: "ReLU"
bottom: "conv1"
top: "conv1"
}
```

Network Scheduler (ARM)

PS
- Convolutions
- Pooling
- De-convolutions
- Fully-connected

© Copyright 2017 Xilinx
32 Bit Training to 8 Bit Inference

Approach 1: Quick evaluation

- Caffe Weights
  - Weights from default Caffe

- Float --> Fixed convertor
  - Using the script provided

- Reduced Precision Caffe weights
  - The lower precision weights are now ready

Approach 2: Use Ristertto Caffe to retrain and fine-tune the weights for better accuracy

<table>
<thead>
<tr>
<th>Network</th>
<th>Default Caffe</th>
<th>Approach# 1</th>
<th>Approach# 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoogleNet</td>
<td>67.35</td>
<td>65.33</td>
<td>66.15</td>
</tr>
<tr>
<td>Alexnet</td>
<td>55.66</td>
<td>53.55</td>
<td>54.01</td>
</tr>
</tbody>
</table>
# Deep Learning Design Examples

<table>
<thead>
<tr>
<th>Model</th>
<th>May 2017</th>
<th>Roadmap</th>
</tr>
</thead>
<tbody>
<tr>
<td>GoogLeNet @ batch = 1 3.2 Gops/img</td>
<td>Images/s: 114</td>
<td>370</td>
</tr>
<tr>
<td></td>
<td>Power (W): 6.0</td>
<td>7.0</td>
</tr>
<tr>
<td></td>
<td>Images/s/watt: 19.0</td>
<td>52.9</td>
</tr>
<tr>
<td>SSD @ batch = 1 62.4 Gops/img</td>
<td>Images/s: 6.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power (W): 6.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Images/s/watt: 1.1</td>
<td></td>
</tr>
<tr>
<td>FCN-AlexNet @ batch = 1 42.0 Gops/img</td>
<td>Images/s: 7.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Power (W): 6.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Images/s/watt: 1.2</td>
<td></td>
</tr>
</tbody>
</table>
Deep Learning IP Export Flow

- Export DNN IP and ARM scheduler to integrate into real system
- Compile-time configuration of DNN IP (e.g. DSP, BRAM, buffer size ...)

```c
main()
{
    imread(A);
    imread(B);
    roi_crop(A, img)
    xFdnn <DSP, BRAM, BUF,...>({img, prototxt, weights, out})
    imshow(out);
}
```
Building a Full Embedded Vision System

System Optimizing Compiler

- Optimized Accelerators & Data Motion Network
- Scheduling of Pre-Optimized Neural Network Layers

Machine Learning

- Caffe
- DNN
- CNN
- GoogLeNet
- SSD
- FCN...

.reVIsION

Optimized Accelerators & Data Motion Network

Scheduling of Pre-Optimized Neural Network Layers

© Copyright 2017 Xilinx
Building a Full Embedded Vision System

C/C++/OpenCL Creation

Profiling to Identify Bottlenecks

System Optimizing Compiler

Optimized Accelerators & Data Motion Network

Scheduling of Pre-Optimized Neural Network Layers

Computer Vision

Machine Learning

OpenVX integration

Caffe

OpenVX

DNN

CNN

GoogLeNet

SSD

FCN...

.prototxt

& Trained Weights

Optimized Accelerators

& Data Motion Network

Scheduling of Pre-Optimized Neural Network Layers

© Copyright 2017 Xilinx
Putting It All Together: CV and CNN with Multiple Sensors

USB-3 to Stereo Vision

SD Card File Read to CNN

Frame Buffer to Optical Flow

Frame Buffer to Frame Buffer

Video Mixer to HDMI TX

MIPI CSI

ISP

VPSS Scaler

Frame Buffer

Optical Flow

Frame Buffer

HDMI TX 3840x2160p60 RGB

USB-3 2x720p60 YUYV 4:2:2

SD Card File Read 500x500x10 RGB

Frame Buffer 1280x720p60 YUYV 4:2:2
Summary

• Zynq SoCs offer superior performance and lower latency compared to other SoC offerings

• reVISION stack provides seamless inference of custom deep learning networks from Caffe to Zynq SoCs

• Visit Xilinx.com/reVISION for more information
Empowering Product Creators to Harness Embedded Vision

The Embedded Vision Alliance (www.Embedded-Vision.com) is a partnership of 60+ leading embedded vision technology and services suppliers.

Mission: Inspire and empower product creators to incorporate visual intelligence into their products.

The Alliance provides low-cost, high-quality technical educational resources for product developers.


The Alliance enables vision technology providers to grow their businesses through leads, ecosystem partnerships, and insights.

For membership, email us: membership@Embedded-Vision.com.
For more information and resources visit www.xilinx.com/reVISION