Xilinx Machine Learning Strategies with Deephi Acquisition

Presented By

Nick Ni, Director of Product Marketing, AI and Edge Computing
Yi Shan, Sr. Director, AI Engineering & Former CTO of DeePhi
The Hottest Research: AI / Machine Learning

Nick’s ML Model
Nick’s ML Framework
AI/ML Monetization Is Here and Growing

copyright sources: Avigilon, Amazon GO, Daimler, SK Telecom
Challenges in Monetizing AI/ML

1080p Object Detection (SSD) @ 30 FPS = 43 TOPS

< 10W, < 50 ms latency, <$50
Who is Xilinx? Why Should I Care for ML?

1. Only HW/SW configurable device for fast changing networks

2. High performance / low power with custom internal memory hierarchy

3. Future proof to lower precisions

4. Low latency end-to-end

5. Scalable device family for different applications
## Integrated Xilinx-Deephi Roadmap

<table>
<thead>
<tr>
<th>Models</th>
<th>Edge/Embedded</th>
<th>Cloud/DC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software Stack</td>
<td>Deephi Pruning</td>
<td>Deephi Pruning</td>
</tr>
<tr>
<td>Deephi Quantizer</td>
<td>Deephi Quantizer</td>
<td></td>
</tr>
<tr>
<td>Deephi Compiler</td>
<td>SDSoc</td>
<td></td>
</tr>
<tr>
<td>Deephi Runtime</td>
<td>SDAccel</td>
<td></td>
</tr>
<tr>
<td>FPGA IP</td>
<td>Deephi DPU</td>
<td>Deephi LSTM</td>
</tr>
<tr>
<td>Xilinx U200, U250, U280</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 20+ pruned / customized / basic models

- Z7020 Board
- Z7020 SOM
- ZU2/3 SOM
- ZU2/3 Card
- ZU9 Card
- ZCU102
- ZCU104
- Ultra96
- xDNN

© Copyright 2018 Xilinx
Deephi as key part of Embedded Vision Development

Xilinx Announces the Acquisition of DeePhi Tech

Deal to Accelerate Data Center and Intelligent Edge Applications

BEIJING and SAN JOSE, CA., July 17, 2018 – Xilinx, Inc. (NASDAQ: XLNX), the leader in adaptive and intelligent computing, announced today that it has acquired DeePhi Tech, a Beijing-based privately held start-up with industry-leading capabilities in machine learning, specializing in deep compression, pruning, and system-level optimization for neural networks.
Long History, Close Collaboration, and Better Future

Collaboration with Xilinx University Program
Deep learning acceleration
Time series analysis
Stereo vision
……

Development of products on Xilinx FPGA platform since inception of DeePhi
Face recognition
Video analysis
Speech recognition acceleration
……

Co-Marketing and Co-Sales with Xilinx Team
Data Center
Automotive
Video surveillance
……

Now Part of Xilinx
Now Part of Xilinx

Provide DPU IP + software tools
AI performance level up significantly

Xilinx owns massive industry customers
Provide wide range of applications

Telecom & Data center  Automotive
Industry IoT  Consumers
Aerospace & Defense  Broadcast

FPGA No. 1
Market and tech leader
Annual revenue of 2.5 billion US dollars

Cloud computing  Intelligent driving  Embedded devices
Cooperation with IT giants  No. 2 semiconductor vendor for global ADAS
Industry, consumer, etc.

Typical application scenarios for AI
Pioneer in sparse-neural-network-based AI computing, explorer from theory to commercialization

First Paper in the World on Compressed and Sparse Neural Networks
“Learning both Weights and Connections for Efficient Neural Networks”, NIPS 2015
“Deep Compression”, ICLR 2016 Best Paper

First Paper in the World on Sparse Neural Network Accelerator
“EIE: Efficient Inference Engine on Compressed Deep Neural Network”, ISCA 2016

First Practical Case Using Sparse Neural Network Processor
Collaboration with Sogou Inc, partly revealed in:
“ESE: Efficient Speech Recognition Engine with Compressed LSTM on FPGA”, FPGA 2017 Best Paper

NIPS 2015: Top conference in neural information processing
FPGA 2016 & 2017: Top academic conference in FPGA
ICLR 2016: Top academic conference in machine learning
ISCA 2016: Top academic conference in computer architecture
Hot Chips 2016: Top academic conference in semiconductor
First prize of tech innovation China Computer Federation

Registering more than 100 invention patents both in China and US
Deep compression
Makes algorithm smaller and lighter

Deep Compression Tool can achieve significant compression on CNN and RNN
Algorithm can be compressed 7 times without losing accuracy under SSD object detection framework
Simple software development kit need only 50 lines of code to run ResNet-50 network
## Pruning Results

### Classification Networks

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Pruning Result 1</th>
<th>Pruning Result 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Top-5</td>
<td>ΔTop5</td>
</tr>
<tr>
<td><strong>ResNet50 [7.7G]</strong></td>
<td>91.65%</td>
<td>91.23%</td>
<td>-0.42%</td>
</tr>
<tr>
<td><strong>Inception_v1 [3.2G]</strong></td>
<td>89.60%</td>
<td>89.02%</td>
<td>-0.58%</td>
</tr>
<tr>
<td><strong>Inception_v2 [4.0G]</strong></td>
<td>91.07%</td>
<td>90.37%</td>
<td>-0.70%</td>
</tr>
<tr>
<td><strong>SqueezeNet [778M]</strong></td>
<td>83.19%</td>
<td>82.46%</td>
<td>-0.73%</td>
</tr>
</tbody>
</table>

### Detection Networks

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Pruning Result 1</th>
<th>Pruning Result 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>mAP</td>
<td>ΔmAP</td>
</tr>
<tr>
<td><strong>DetectNet [17.5G]</strong></td>
<td>44.46</td>
<td>45.7</td>
<td>+1.24</td>
</tr>
<tr>
<td><strong>SSD+VGG [117G]</strong></td>
<td>61.5</td>
<td>62.0</td>
<td>+0.5</td>
</tr>
<tr>
<td>[A] <strong>SSD+VGG [173G]</strong></td>
<td>57.1</td>
<td>58.7</td>
<td>+1.6</td>
</tr>
<tr>
<td>[B] <strong>Yolov2 [198G]</strong></td>
<td>80.4</td>
<td>81.9</td>
<td>+1.5</td>
</tr>
</tbody>
</table>

### Segmentation Networks

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Pruning Result 1</th>
<th>Pruning Result 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>mIoU</td>
<td>ΔmIoU</td>
</tr>
<tr>
<td><strong>FPN [163G]</strong></td>
<td>65.69%</td>
<td>65.21%</td>
<td>-0.48%</td>
</tr>
</tbody>
</table>
Pruning Speedup Example – SSD

Pruning Speedup on Hardware (2xDPU-4096@Zu9)
SSD+VGG 4 classes detection @Deephi surveillance data

Operations(G) / mAP

<table>
<thead>
<tr>
<th>Pruning procedure</th>
<th>Operations(G)</th>
<th>mAP (%)</th>
<th>fps</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline</td>
<td>117</td>
<td>61.5</td>
<td>18</td>
</tr>
<tr>
<td>1</td>
<td>63.4</td>
<td>63.4</td>
<td>37</td>
</tr>
<tr>
<td>2</td>
<td>63.5</td>
<td>63.4</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>62.4</td>
<td>62</td>
<td>23</td>
</tr>
<tr>
<td>4</td>
<td>61.5</td>
<td>61.5</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>61.1</td>
<td>61</td>
<td>17</td>
</tr>
<tr>
<td>7</td>
<td>60.8</td>
<td>60.8</td>
<td>14.6</td>
</tr>
<tr>
<td>8</td>
<td>61</td>
<td>61</td>
<td>13.6</td>
</tr>
<tr>
<td>9</td>
<td>60.8</td>
<td>60.8</td>
<td>12.2</td>
</tr>
<tr>
<td>10</td>
<td>60.4</td>
<td>60.4</td>
<td>11.6</td>
</tr>
</tbody>
</table>

© Copyright 2018 Xilinx
Pruning Speedup Example – Yolo_v2

Pruning Speed up on Hardware (2xDPU@Zu9)
YoloV2 single class detection @ Customer’s data

Pruning procedure

Operations(G) / mAP

Baseline   1   2   3   4   5   6   7   8   9   10
173    11.6 14.4 18.4 21.2 26.8 28.4 32.4 37.2 41.6 43.6
Operations(G):
56.7 57.9 69 58.1 57.8 56.9 56.6 55.4 54.4

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

Baseline   1   2   3   4   5   6   7   8   9   10

Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)

2.8x
Operations(G) / mAP

mAP(%)
Compression perspective

- **Quantization**
  - **Low-bit and hybrid low-bit quantization**
    - Some simple hybrid low-bit experiments [Compared to 8bit results, without finetune]
      - 20% model size reduce, <1% accuracy drop
      - 10% model size reduce, <1% accuracy drop (hardware-friendly low-bit patterns)
  - **7nm FPGA with math engine**
    - Some fp32/fp16 resources -> Relax some restrictions for quantization -> Better performance
      - For low-bit quantization, non-uniform quantization with lookup tables is possible
      - Some layers can run without quantization
  - **AutoML for quantization**
    - Automated quantization for hybrid low-bit quantization

- **Pruning**
  - **AutoML for pruning**
    - Automated pruning by reinforcement learning

- **Tools**
  - Unified compression tool supporting different frameworks
  - Fully tested tools, ease of use
  - Improved speed for pruning tool, supporting cluster
Core advantage | Instruction set and DPU architecture

DPU Aristotle CNN accelerator

- **GoogleNet-V3**: 23% (Aristotle), 14% (Iphone8plus), 18% (Kirin 970)
- **ResNet-50**: 24% (Aristotle), 13% (Iphone8plus), 18% (Kirin 970)
- **VGG16**: 40% (Aristotle), 52% (Iphone8plus), 51% (Kirin 970)

Very high hardware utilization

Source: Published results from Huawei

DPU/FPGA v.s. Sophon BM1680 (ASIC-Bitmain)

Under the same computing power performance, DeePhi’s FPGA lead Sophon significantly both in power consumption and hardware utilization.

ResNet-50
- **Sophon BM1680** (37.5%): 51% utilization
- **Aristotle on 7020 FPGA** (3%): 51% utilization

Power consumption (W) and Hardware utilization

Source: [https://sophon.ai/product/sc1.html](https://sophon.ai/product/sc1.html)

Note: *For ResNet-50, Sophon is 112GOPS with 2TOPS at peak, utilization is 5.5%. Aristotle is 117GOPS with 230GOPS at peak, utilization is 51%*
Current Ceiling of CNN Architecture

INT8 improvements are slowing down and approaching the ceiling.

Solutions

- Sparsity
- Low Precision

Sparsity architecture exploration

- On clouds, aiming at customers all over the world
- Already officially launched in AWS Marketplace and HUAWEI cloud (http://www.deephi.com/ddese.html)
- Now transplanting to Alibaba cloud

Features:

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low storage</td>
<td>Model compressed more than 10X with negligible loss of accuracy</td>
</tr>
<tr>
<td>Low latency</td>
<td>More than 2X speedup compared to GPU (P4)</td>
</tr>
<tr>
<td>Programmable</td>
<td>Reconfigurable for different requirements</td>
</tr>
</tbody>
</table>

For end-to-end speech recognition
Challenges of Sparse NN Accelerator

- The conflicts of the irregular pattern of mem access and the regular pattern of calculating.
- Difficult to take account of the sparsity of both activation and weights at the same time.
- Additional on-chip memory requirements for indexes.
Potentials of low precision

ISSCC, 2017

Low Precision Becomes Popular

ISSCC, 2018

- Scales performance
- Reduces hardware resources
- Less bandwidth/on-chip memory requirement
- Regular memory access pattern and calculating pattern

FPGA benefits a lot from low-precision.

### Energy Cost

<table>
<thead>
<tr>
<th>Operation</th>
<th>Energy (pJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1bit Fixed-point MAC</td>
<td>0.118</td>
</tr>
<tr>
<td>4bit Fixed-point MAC</td>
<td>0.517</td>
</tr>
<tr>
<td>8bit Fixed-point MAC</td>
<td>0.865</td>
</tr>
<tr>
<td>16bit Fixed-point MAC</td>
<td>1.64</td>
</tr>
</tbody>
</table>

*65nm process, 200Mhz, 1.2V, 25°C

### Model Size (ResNet-50)

<table>
<thead>
<tr>
<th>Precision</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b</td>
<td>3.2</td>
</tr>
<tr>
<td>8b</td>
<td>25.5</td>
</tr>
<tr>
<td>32b</td>
<td>102.5</td>
</tr>
</tbody>
</table>
Architecture perspective: Mixed Low-Precision

Fixed low-precision quantization already showed competitive results.

Next generation: Variable precision of activation/weights among layers

*accuracy drop less than 1%

Preliminary experiments on popular networks. (vgg-16, resNet-50, inception-v4)
Architecture perspective: Mixed Low-Precision CNN

- **Mixed Precision Support**
  - INT8/6/5/4/3/2

- **Flexible Between Throughput and Latency**
  - Switch between Throughput-Opt-Mode and Latency-Opt-Mode without RTL change

- **Enhanced Dataflow Techniques**
  - Make the balance among different layers. Do NOT require the model can be fully placed on chip, but load the data at the right time.
  - Physical-aware data flow design to meet higher frequency.
  - Supports high-resolution images at high utilization.
**Software perspective**

### Application

- **Continuous supporting customers for products and solutions**
  - Improving surveillance products and providing more ADAS/AD demonstration to customers

- **System-level optimization for applications**
  - Accelerating time-consuming operations by FPGA and optimizing memory access

### SDK

- **Providing complete SDK for surveillance customers**
  - Such as face and vehicle related SDK

- **Constructing ADAS/AD libraries for internal developers and customers**
  - Such as vehicle detection, segmentation etc.

### Embedded

- **Providing system for evaluation and product boards**
  - From ZU2 to ZU11

- **Developing more IO drivers**
  - Such as USB 3.0, MIPI etc.

- **Researching other system related with our products**

---

**Software team will provide full stack solutions for AI applications**

- **Applications**
  - Surveillance
  - ADAS/AD
  - Data center

- **User-space libraries**
  - Application-specific AI library
  - Acceleration library
  - Communication library

- **IO drivers**
  - OS
  - DPU runtime

- **DPU architecture**
  - DPU kernel module

- **FPGA**
DNNDK perspective

Solid Toolchain Stack for XILINX ACAP

- Most efficiency solution for ML on XILINX next generation computing platform
- Most easy-to-use & productive toolchain for ML algorithms deployment
System perspective: schedule ADAS tasks in single FPGA

> **Multi-task Models**
  
  >> **Training:**
  - Knowledge sharing
  - Reduce computation cost
  
  >> **Pruning:**
  - Balance different objective functions

> **Sensor Fusion**
  
  >> Sensor alignment & Data Fusion

> **Task scheduling**
  
  >> Resource constrained scheduling: Serialization & Parallelization
  
  >> Task scheduling and memory management framework with low context-switching cost
  
  >> Support new operations with runtime variable parameter by software and hardware co-design
System perspective: Video Surveillance in single FPGA

- Platform: ZU4EV
- DPU: B2304_EU
- Peak perf.: 921Gops (400Mhz)
- Power: 7.7W (XPE)

Single Chip Solution

This solution needs to further enhance ISP functionality.
Attend presentations and workshops for more details

<table>
<thead>
<tr>
<th>DAY 1</th>
<th>DAY 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Crystal</strong></td>
<td><strong>Regency 1</strong></td>
</tr>
<tr>
<td><strong>Piedmont</strong></td>
<td><strong>Sacramento</strong></td>
</tr>
<tr>
<td>9am-10am</td>
<td>9am-10am</td>
</tr>
<tr>
<td>10am-11am</td>
<td>10am-11am</td>
</tr>
<tr>
<td>11am-12pm</td>
<td>11am-12pm</td>
</tr>
<tr>
<td>12pm-1pm</td>
<td>12pm-1pm</td>
</tr>
<tr>
<td>1pm-2pm</td>
<td>1pm-2pm</td>
</tr>
<tr>
<td>2pm-3pm</td>
<td>2pm-3pm</td>
</tr>
<tr>
<td>3pm-4pm</td>
<td>3pm-4pm</td>
</tr>
<tr>
<td>4pm-5pm</td>
<td>4pm-5pm</td>
</tr>
<tr>
<td><strong>Presentation</strong></td>
<td><strong>Lab</strong></td>
</tr>
<tr>
<td>(own laptop required)</td>
<td><strong>ML Expert panel</strong></td>
</tr>
<tr>
<td><strong>Interactive</strong></td>
<td><strong>Building ML vision systems with SDSoC</strong></td>
</tr>
<tr>
<td><strong>Machine learning</strong></td>
<td><strong>Machine learning for embedded systems</strong></td>
</tr>
<tr>
<td><strong>with DeePhi</strong></td>
<td><strong>Ford ADAS</strong></td>
</tr>
<tr>
<td><strong>Building ML vision systems</strong></td>
<td><strong>Machine learning</strong></td>
</tr>
<tr>
<td><strong>with SDSoC</strong></td>
<td><strong>for embedded</strong></td>
</tr>
<tr>
<td><strong>Machine learning</strong></td>
<td><strong>Machine Learning</strong></td>
</tr>
<tr>
<td><strong>with SDSoC for EV</strong></td>
<td><strong>with SDSoC for EV</strong></td>
</tr>
<tr>
<td><strong>Machine learning for</strong></td>
<td><strong>Building ML vision systems</strong></td>
</tr>
<tr>
<td><strong>embedded systems</strong></td>
<td><strong>with SDSoC</strong></td>
</tr>
</tbody>
</table>

© Copyright 2018 Xilinx
Architecture perspective: Mixed Low-Precision CNN

> Mixed Precision Support
  >> INT8/6/5/4/3/2

> Flexible Between Throughput and Latency
  >> Switch between Throughput-Opt-Mode and Latency-Opt-Mode without RTL change

> Enhanced Dataflow Techniques
  >> Make the balance among different layers. Do NOT require the model can be fully placed on chip, but load the data at the right time.
  >> Physical-aware data flow design to meet higher frequency.
  >> Supports high-resolution images at high utilization.

> Performance Target (googlenet_v1)
  >> 3103 FPS (INT8)
  >> 5320 FPS (INT8/4/2 mixed)
  >> 12412 FPS (INT2 only)

> Release Plan
  >> First version: 2019Q1