ACCELERATING DEEP NEURAL NETWORKS FOR REAL-TIME DATA SELECTION FOR HIGH-RESOLUTION IMAGING PARTICLE DETECTORS

Georgia Karagiorgi\textsuperscript{1}, with Luca Carloni\textsuperscript{2}, Giuseppe Di Guglielmo\textsuperscript{2}, and Yeon-jae Jwa\textsuperscript{1}

\textsuperscript{1}Dept. of Physics, Columbia University
\textsuperscript{2}Computer Science Department, Columbia University
High-resolution imaging particle physics detectors

- E.g. Deep Underground Neutrino Experiment (DUNE)

[4 neutrino detector modules 1 mile underground]

[https://www.dunescience.org/]

Sanford Underground Research Facility, South Dakota

Fermi National Accelerator Laboratory, Illinois

800 miles/1300 km
What is DUNE “looking for”?

- Rare interactions of (otherwise) invisible particles:
  - Neutrinos from a beam produced at the Fermi US National Lab (~few hundred per year)
  - Neutrinos produced in cosmic ray air showers in the atmosphere (~few thousand per year)
What is DUNE “looking for”?

- Rare interactions of (otherwise) invisible particles:
  - Neutrinos from a beam produced at the Fermi US National Lab (~few hundred per year)
  - Neutrinos produced in cosmic ray air showers in the atmosphere (~few thousand per year)
  - Neutrinos produced in a (potential) nearby supernova burst (up to ~few thousand over 10 seconds, but ~once per century)
What is DUNE “looking for”?

- Rare interactions of (otherwise) invisible particles:
  - Neutrinos from a beam produced at the Fermi US National Lab (~few hundred per year)
  - Neutrinos produced in cosmic ray air showers in the atmosphere (~few thousand per year)
  - Neutrinos produced in a (potential) nearby supernova burst (up to ~few thousand over 10 seconds, but ~once per century)
  - Protons or neutrons inside the detector volume (liquid argon) spontaneously “decaying” in a way that violates fundamental symmetries of nature (~1 per year)

Rare processes, of fundamental importance in nature!
What would DUNE “see”?
What would DUNE “see”?

• For the most part:

Single frame from high-resolution video:
One of three 2D views from one of hundreds of cells in the detector

Color scale represents energy deposition (due to ionization) in the detector

“Static” is noise and small energy deposits from radiological impurities in the detector
What would DUNE “see”?  

- What **events of interest** would look like:

  - Easy to pick out from background!
  - On an event-by-event basis, difficult to differentiate between them
  - On average, events can be differentiated based on their **energy** (pixel intensity) and **topology** characteristics (spatial extent, shape, e.g. tracks vs. showers and multiplicity, connected vs. detached, …)
Not all events of interest are as easy to pick out!

• Special challenge: **neutrinos from supernova core collapse**
• Very low energy and small (in extent) topology, similar to radiological background activity in the detector

• Need $O(10^4)$ background suppression, while maintaining high efficiency to a frame containing a supernova neutrino interaction

See: yesterday’s talk by P. Nugent
Not all events of interest are as easy to pick out!

- Special challenge: **neutrinos from supernova core collapse**
- Very low energy and small (in extent) topology, similar to radiological background activity in the detector

- Need $O(10^4)$ background suppression, while maintaining high efficiency to a frame containing a supernova neutrino interaction

See: yesterday’s talk by P. Nugent
DUNE detector: working principle*

- particle-imaging detector
- stereoscopic “video capture” of activity within detector volume with sub-mm spatial resolution
- high-resolution “video” streams:
  - up to 4x150 cell volumes
  - 11.5 megapixel frames per 2.25ms
  - 12-bit resolution
  - a total of \(~40\) terabits/s
- continuous operation for more than a decade

See: Poster #8, Session #2, by J. I. Crespo-Anadon

*shown only for “single-phase” module technology; ~similar “dual-phase” module
DUNE detector: working principle*

- particle-imaging detector
- stereoscopic “video capture” of activity within detector volume with sub-mm spatial resolution
- high-resolution “video” streams:
  - up to 4x150 cell volumes
  - 11.5 megapixel frames per 2.25ms
  - 12-bit resolution
- a total of \(~40\) terabits/s
- continuous operation for more than a decade

See: Poster #8, Session #2, by J. I. Crespo-Anadon

*shown only for “single-phase” module technology; ~similar “dual-phase” module
Promise of imaging techniques for DUNE

- Raw data format ideally suited for image analysis
- **Convolutional Neural Networks (CNNs)** could be applied for image classification “on the fly”
  - Work with only one projection (2D): 4.3 megapixel
  - Down-sample and resize image to 0.36 megapixel
  - Classify via CNN as one of three cases: **background/supernova-like low energy activity/high-energy activity**
Promise of imaging techniques for DUNE

- Classification studies performed for DUNE simulated frames using CNN vgg16b:

<table>
<thead>
<tr>
<th>Background CNN score cut</th>
<th>Background frame selection efficiency</th>
<th>Background data rate</th>
<th>Supernova frame sel. efficiency</th>
<th>n-nbar frame sel. efficiency</th>
<th>p-decay frame sel. efficiency</th>
<th>atm. nu frame sel. efficiency</th>
<th>cosmic frame sel. efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;0.05</td>
<td>0.56% (99.44% rejection)</td>
<td>6.4 GB/s (201 PB/year)</td>
<td>89%</td>
<td>100%</td>
<td>99%</td>
<td>92%</td>
<td>92%</td>
</tr>
<tr>
<td>&lt;0.01</td>
<td>0.18% (99.82% rejection)</td>
<td>2.05 GB/s (65 PB/year)</td>
<td>86%</td>
<td>100%</td>
<td>99%</td>
<td>91%</td>
<td>92%</td>
</tr>
<tr>
<td>&lt;0.001</td>
<td>0.031% (99.969% rejection)</td>
<td>350 MB/s (11 PB/year)</td>
<td>77%</td>
<td>100%</td>
<td>98%</td>
<td>89%</td>
<td>90%</td>
</tr>
<tr>
<td>&lt;0.0002</td>
<td>0.011% (99.989% rejection)</td>
<td>125 MB/s (3.9 PB/year)</td>
<td>69%</td>
<td>100%</td>
<td>97%</td>
<td>87%</td>
<td>88%</td>
</tr>
</tbody>
</table>

- High selection efficiency across all topologies of interest

✅ CNN-based selection on unprocessed, raw data

- Further improvements possible by considering time-coincidence of activity over multiple (sequential) frames
Promise of imaging techniques for DUNE

- Deep Learning techniques already applied successfully in detectors sharing the same technology as DUNE
- E.g. MicroBooNE experiment (1/500\textsuperscript{th} size of DUNE) is pioneering such applications

See, e.g.:
Promise of imaging techniques for DUNE

- Deep Learning techniques already applied successfully in detectors sharing the same technology as DUNE
- E.g. MicroBooNE experiment (1/500\textsuperscript{th} size of DUNE) is pioneering such applications

DUNE readout and data acquisition system design

above ground in South Dakota

real-time or batch processing

batch processing

off-site permanent data storage and offline processing in Illinois, and international sites

100 Gbps

~few Tbps

40 Tbps

detector

1 mile underground in South Dakota

[DUNE Technical Design Report, in preparation.]
DUNE readout and data acquisition system design

above ground in South Dakota

1 mile underground in South Dakota

batch processing

~few Tbps

40 Tbps

detector

real-time or batch processing

Example: Real-time waveform processing (hit finding) in FPGA in the MicroBooNE readout.

See: Poster #8, Session #2, by J. I. Crespo-Anadon

[DUNE Technical Design Report, in preparation.]
DUNE readout and data acquisition system design

above ground in South Dakota

1 mile underground in South Dakota

40 Tbps

real-time or batch processing

~few Tbps

batch processing

100 Gbps

off-site permanent data storage and offline processing in Illinois, and international sites

• **Flexibility for potential implementation** of Deep Neural Networks for image-analysis-based data selection
• **Must keep within cost and performance constraints:** latency, power, cost envelope
Performance for batch processing for data selection with GPU implementation

• **GPU advantages**: High computational density, level of programmability, data-parallelism, flexibility

• **Investigated CNN-based selection performance** (latency) for DUNE simulated frames:
  On single GPU (NVIDIA GeForce GXT 1080 Ti)
  • vgg16b 26 ms/frame (compare to 2.25ms frame)
  • resnet14b 24 ms/frame

Includes data i/o and network inference time

• Speed sufficient for downstream implementation; but a factor of 10 speedup needed for upstream implementation (power constraints aside...)
  • Further optimization may be possible: e.g. image size: further down-sampling vs. region-of-interest

<table>
<thead>
<tr>
<th>Frame size</th>
<th>140x140</th>
<th>280x280</th>
<th>600x600</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured time (ms)</td>
<td>18.92</td>
<td>22.10</td>
<td>27.58</td>
</tr>
</tbody>
</table>
R&D for real-time processing for data selection with FPGA implementation

- **Advantages for upstream (FPGA) implementation**: reduction in overall data transfer to above ground, buffering needs, power dissipation
  - FPGA: power-aware platform for CNN acceleration, but resource-constrained
  - Concern: network size (resnet14b, vgg16b) and input image size are large

- **Exploring CNN acceleration** using a customizable and efficient hardware accelerator design for the various layers of CNN, utilizing High Level Synthesis-based design flow

- **Flexibility for optimization** (processing time, efficiency, resource utilization)
Design Flow for FPGA Accelerators

- Register Transfer Level (RTL) is a low level representation of digital circuits and is a *de facto* standard for designing hardware.
- High Level Synthesis (HLS) allows hardware designers to take advantage of benefits of working at a **higher level of abstraction**, while creating high-performance hardware.
  - HLS allows to efficiently and rapidly perform **Design Space Exploration (DSE)**.
High Level Synthesis

• HLS transforms a behavioral description into timed design

• This is done in three steps: scheduling, binding and technology mapping

```c
int func(int a, int b, int c, int d) {
    int z;
    z = (a + b) * c + d + a;
    return z;
}
```
Knobs of High Level Synthesis

- HLS allows to control fine-grain architectural implementation using pre-defined **knobs**
- Allow exploring concurrency in design, e.g.

```c
void sum(int a[4], int b[4], int c[4]) {
    for (int i = 0; i < 4; i++) {
        #pragma HLS UNROLL factor=1
        c[i] = a[i] + b[i];
    }
}
```

- Can explore implementations based on desired performance (latency) and cost (area, power)
Knobs of High Level Synthesis

- HLS allows to control fine-grain architectural implementation using pre-defined **knobs**
- Allow exploring concurrency in design, e.g.
  ```
  void sum(int a[4], int b[4], int c[4]) {
    for (int i = 0; i < 4; i++) {
      #pragma HLS UNROLL factor=2
      c[i] = a[i] + b[i];
    }
  }
  ```
- Can explore implementations based on desired performance (latency) and cost (area, power)
Knobs of High Level Synthesis

• HLS allows to control fine-grain architectural implementation using pre-defined **knobs**

• Allow exploring concurrency in design, e.g.

```c
void sum(int a[4], int b[4], int c[4]) {
    for (int i = 0; i < 4; i++) {
        #pragma HLS UNROLL factor=4
        c[i] = a[i] + b[i];
    }
}
```

• Can explore implementations based on desired performance (latency) and cost (area, power)
Knobs of High Level Synthesis

- All of these implementations are optimal in terms of cost (area) and performance (latency)

```c
void sum(int a[4], int b[4], int c[4]) {
    for (int i = 0; i < 4; i++) {
        c[i] = a[i] + b[i];
    }
}
```
Convolutional Layers

- Convolutional layers are the most computational intensive part in CNNs

\[
Y_{k,i,j} = \sum_{c=0}^{C-1} X_c \ast W_{k,c} + B_k = \left[ \sum_{c=0}^{C-1} \sum_{x=0}^{F-1} \sum_{y=0}^{F-1} X_{c,i+x-\frac{F}{2},j+y-\frac{F}{2}} \cdot W_{k,c,x,y} \right] + B_k
\]

Distribution of floating-point operations per stages in vgg16b
Balance of Computation and Communication

- For hardware accelerator, one should carefully design the algorithm to reuse data as much as possible, thus reducing expensive memory transfers from and to off-chip DRAM
Tailoring Private Local Memory

- Both inputs and weights are divided in chunks and the computation is done only with the on-chip copy of the data
Accelerator Structure Overview

- Highly configurable accelerator
## Preliminary Results

<table>
<thead>
<tr>
<th></th>
<th>MFLOP</th>
<th>CPU</th>
<th>Accelerator</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time</td>
<td>GFLOPS</td>
<td>Time</td>
<td>GFLOPS</td>
<td>Speedup</td>
<td></td>
</tr>
<tr>
<td>conv1_1</td>
<td>86.7</td>
<td>2.17</td>
<td>0.04</td>
<td>0.21</td>
<td>0.41</td>
<td>10.31</td>
</tr>
<tr>
<td>conv1_2</td>
<td>3699.4</td>
<td>51.05</td>
<td>0.07</td>
<td>3.66</td>
<td>1.01</td>
<td>13.95</td>
</tr>
<tr>
<td>conv2_1</td>
<td>1849.7</td>
<td>25.24</td>
<td>0.07</td>
<td>1.82</td>
<td>1.02</td>
<td>13.87</td>
</tr>
<tr>
<td>conv2_2</td>
<td>3699.4</td>
<td>51.27</td>
<td>0.07</td>
<td>3.46</td>
<td>1.07</td>
<td>14.82</td>
</tr>
<tr>
<td>conv3_1</td>
<td>1849.7</td>
<td>24.84</td>
<td>0.07</td>
<td>1.72</td>
<td>1.08</td>
<td>14.44</td>
</tr>
<tr>
<td>conv3_2</td>
<td>3699.4</td>
<td>50.85</td>
<td>0.07</td>
<td>3.37</td>
<td>1.10</td>
<td>15.09</td>
</tr>
<tr>
<td>conv3_3</td>
<td>3699.4</td>
<td>51.24</td>
<td>0.07</td>
<td>3.37</td>
<td>1.10</td>
<td>15.20</td>
</tr>
<tr>
<td>conv4_1</td>
<td>1849.7</td>
<td>25.23</td>
<td>0.07</td>
<td>1.68</td>
<td>1.10</td>
<td>15.02</td>
</tr>
<tr>
<td>conv4_2</td>
<td>3699.4</td>
<td>50.68</td>
<td>0.07</td>
<td>3.34</td>
<td>1.11</td>
<td>15.17</td>
</tr>
<tr>
<td>conv4_3</td>
<td>3699.4</td>
<td>50.68</td>
<td>0.07</td>
<td>3.34</td>
<td>1.11</td>
<td>15.17</td>
</tr>
<tr>
<td>conv5_1</td>
<td>924.8</td>
<td>12.46</td>
<td>0.07</td>
<td>0.84</td>
<td>1.10</td>
<td>14.83</td>
</tr>
<tr>
<td>conv5_2</td>
<td>924.8</td>
<td>12.46</td>
<td>0.07</td>
<td>0.84</td>
<td>1.10</td>
<td>14.83</td>
</tr>
<tr>
<td>conv5_3</td>
<td>924.8</td>
<td>12.46</td>
<td>0.07</td>
<td>0.84</td>
<td>1.10</td>
<td>14.83</td>
</tr>
</tbody>
</table>

### Performance Metrics

<table>
<thead>
<tr>
<th></th>
<th>Time (s)</th>
<th>Power (W)</th>
<th>PET (Img/s/W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARM A53</td>
<td>420</td>
<td>3.2</td>
<td>0.001</td>
</tr>
<tr>
<td>Xilinx XCZU9EG FPGA</td>
<td>28</td>
<td>0.8</td>
<td>0.045</td>
</tr>
</tbody>
</table>

15x average speedup
45x more power efficient
w.r.t software implementation on ARM Cortex A53
Summary

• There is an increasing need for real-time processing of high-resolution images from particle detectors

• DUNE is a prime application for image processing using DNNs, and calls for optimizing DNN implementation on power-efficient platforms

• Serves as an ideal case for collaboration between physics and computer science

  • Demonstrated applicability of DNN-based selection
  • In the process of optimizing implementation on power-efficient platform
  • Future plans: demonstration of real-time processing meeting performance and cost requirements
Acknowledgements

• Simone Rossi for initial accelerator studies
• Ashley Koo for ResNet studies
• Yuyang Zhou for initial CNN data selection studies
• Jeremy Hewes for initial CNN physics studies
• Kazu Terao for valuable feedback

Funding support though:
• National Science Foundation
• Columbia Research Initiatives in Science and Engineering (RISE)
• Columbia University Provost’s Office