Nervana and Graphcore

Miheer Vaidya (@vmiheer)
The Ohio State University
Nervana and Graphcore

(Architectural advancements for AI)

Xiheer Vaidya (@vXiheer)
(The guy who asked questions to everyone! 😊)
Outline

Nervana
Graphcore
Similarities & differences
Outline
Nervana
Graphcore
Similarities & differences
Nervana

A video about architecture
Nervana

Built for Neural Networks

Efficiency

Memory

Compute
Nervana

Efficiency

Memory

Compute
Nervana
Efficiency
Memory
Banked HBM
Thought Bubble

“Cores are how compute happens parallelly. Banks is how memory accesses happens parallelly.”
Nervana

Efficiency

Memory

Banked HBM

Core local (shared?) memory

Is it same as GPUs?
Nervana
Efficiency
Memory
GPU arch
Nervana
Efficiency

Memory compared to GPUs:

Shared memory exchange across:
- Nodes
- Chips
⇒ Model Parallelism
Nervana
Efficiency
Memory
Compute
Nervana
Compute
Flexpoint
ieee floating point
Nervana

Flexpoint

(a) Weights

(b) Activations

(c) Weight updates

Base-2 logarithm of absolute value

Epoch 0
Epoch 164

Flexpoint: An Adaptive Numerical Format for Efficient Training of Deep Neural Networks Urs Köster et al.
Nervana

Compute

Flexpoint

Common exponent for a tensor

$\text{FlexN+M} \Rightarrow \text{N bit mantissa, M-bit exponent}$
Nervana
Compute
Flexpoint

Common exponent for a tensor
Claim: Flex16+5 $\sim$ Float32
Nervana
Compute
Flexpoint benefits
Half the memory
Half the bandwidth
Nervana
Compute

Flexpoint benefits

Low energy compute
Fixed point arithmetic
For tensor multiplication
For addition inside tensor
Nervana

Maximize Compute Utilization
Easy scale out
Less power
Nervana

Compute

So is this all blue skies and rainbows?
Nervana

“Tensors with a shared exponent must have a sufficiently narrow dynamic range such that mantissa bits alone can encode variability.”
Nervana

Flexpoint example

Flexpoint: An Adaptive Numerical Format for Efficient Training of Deep Neural Networks Urs Köster et al.
Nervana

Flexpoint

Exponent prediction and updates
Sliding window per tensor “f”

\[ \chi \leftarrow \alpha \left[ \max(f) + \beta \cdot std(f) + \gamma \chi \right] \]
Nervana

Flexpoint result

(a) ImageNet1k AlexNet

(b) CIFAR-10 ResNet
Nervana

Flexpoint

result
Outline

Nervana

Graphcore

Similarities & differences
Graphcore

https://www.graphcore.ai/nips2017_presentations
Graphcore’s “Colossus”
(in honour of Tommy Flowers)

- Designed ground-up for MI, both training and deployment.
- Large 16nm custom chip, cluster-able, 2 per PCIe card.
- >1000 truly independent processors per chip; all-to-all non-blocking exchange.
- All model state remains on chip; no directly-attached DRAM.
- Mixed-precision floating-point stochastic arithmetic.
- DNN performance well beyond Volta and TPU2; efficient without large batches.
- Unprecedented flexibility for non-DNN models; thrives on sparsity.
- Program in TensorFlow initially, other o/s frameworks follow; or Poplar™ for close-to-metal.
- Early access cards and appliances end-2017.

https://www.youtube.com/watch?v=Gh-Tff7DdzU
Graphcore
Compute
Memory
Graphcore
Compute
Memory
Graphcore

Compute

Computation on graphs
massive parallelism

Low precision, wide dynamic range arithmetic
mixed-precision float 16.32 (and smaller?)
Graphcore

Compute

uniformly distributed integers
vectors of approximately Gaussian distributed floats
Random zeroing of vector elements with specified probability
Stochastic rounding of f32 accumulations to f16
Graphcore
Compute
Memory
Graphcore

Memory

Static graph structure

  compiler can partition work, allocate memory, and schedule messages

  bulk-synchronized, address-less communication

  distributed memory
Graphcore

Memory

Serialise communication and compute

Keep memory local

Re-compute what you can’t remember
Graphcore

Memory

**Serialise communication and compute**

Keep memory local

Re-compute what you can’t remember
Computing in silicon is power-limited

Largest manufacturable die 8cm$^2$

200W (forced air)

8cm$^2$

0.1mOhm

800mV
250A

https://www.youtube.com/watch?v=Gh-Tff7DdzU
Graphcore Memory

- Largest manufacturable die: ~825mm²
- 16nm 1Pflop: 16.32 < 700mm²
  1000W
- 8cm² die: @ 200W

https://www.graphcore.ai/nips2017_presentations
Graphcore

Memory

Used exclusively for other applications (graphics, HPC)

- 10% ram
- 55% dark
- 35% fpu

used for ML

GPU plan:
- 75% ram
- 25% fpu

IPU plan:
- 100% of memory

https://www.graphcore.ai/nips2017_presentations
Graphcore Memory

Concurrent compute and communication

Serialized compute and communication

https://www.graphcore.ai/nips2017_presentations
Graphcore

Memory

Compute Phase
stateful codelets execute on local memory state

Exchange Phase
memory-to-memory data movement, no compute, no concurrency hazards

https://www.graphcore.ai/nips2017_presentations
Graphcore
Memory
Serialise communication and compute
Keep memory local
Re-compute what you can’t remember
Graphcore
Memory
Serialise communication and compute
Keep memory local
Re-compute what you can’t remember
Graphcore Memory

https://www.graphcore.ai/nips2017_presentations
Graphcore
Results
8 IPUs = 54 V100 = 128 P100
(resnet @16,000 images/sec)

https://www.graphcore.ai/nips2017_presentations
Graphcore

Results

DEEPVOICE: SAMPLE RATE

Sample Rate

16KHz

Does not meet real-time performance

Voice quality - Fair
Mean Opinion Score = 3.35
layers=20 | residual channels = 84 | skip channels = 128

Voice quality - Good
Mean Opinion Score = 3.91
layers=40 | residual channels = 84 | skip channels = 256

CPU
GPU
IPU

Intel Xeon
E5-2680 v3 Haswell (2.6GHz)

Nvidia
GeForce GTX Titan X Maxwell

Graphcore C2
IPU accelerator PCIe card

Source: Baidu SVM & Graphcore
https://arxiv.org/abs/1702.07625

https://www.graphcore.ai/nips2017_presentations
Graphcore Results

Mean Opinion Score = 3.35 | layers=20 | residual channels = 64 | skip channels = 128

- IPU: 183X
- GPU: unable to meet minimum sample rate
- CPU: low

Audio channels supported per device:
- Graphcore C2 IPU accelerator PCIe card
- Nvidia GeForce GTX Titan X Maxwell
- Intel Xeon E5-2680 v3 Haswell (2.6GHz)

https://www.graphcore.ai/nips2017_presentations
Outline

Nervana
Graphcore
Similarities & differences
Similarities & differences

<table>
<thead>
<tr>
<th>Nervana</th>
<th>Graphcore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Per core memory</td>
<td>Per core memory</td>
</tr>
<tr>
<td>Specialized for application</td>
<td>Specialized for application</td>
</tr>
<tr>
<td>Low precision compute</td>
<td>Low precision compute</td>
</tr>
<tr>
<td>Effective on/off chip communication</td>
<td>Effective on/off chip communication</td>
</tr>
</tbody>
</table>
### Similarities & differences

<table>
<thead>
<tr>
<th>Nervana</th>
<th>Graphcore</th>
</tr>
</thead>
<tbody>
<tr>
<td>NN only</td>
<td>Generic ml</td>
</tr>
<tr>
<td>HBM</td>
<td>No central memory</td>
</tr>
<tr>
<td>Flexpoint</td>
<td>Fp16 + fp32</td>
</tr>
</tbody>
</table>
Thank you!
Graphcore:
https://www.youtube.com/watch?v=Gh-Tff7DdzU
https://www.youtube.com/watch?v=bwDAWh_GuA0
Graphcore

Specialization between interference and training not needed

Not just for NN (Think ahead 20 years)

Unlike HPC it’s high performance low precision

Static graphs (knowledge doesn’t suddenly change)
Graphcore

What shall we bet is fundamental to MI workloads?

- Huge parallelism
- Sparsity (layers, embedding dimension mismatch)
- Low precision (approximate inference on stochastically-learned models)
- Model parameter re-use (convolution, recurrence)
- Static graph structure (compiler can partition, allocate, and schedule)
- Linear projections
- Sources of entropy
- More numeric range than precision (floating-point)