DNN-Dataflow-Hardware Co-Design for Enabling Pervasive General-Purpose AI

Tushar Krishna

Assistant Professor
School of ECE
Georgia Institute of Technology
tushar@ece.gatech.edu

CRNCH Summit
November 2, 2018
The Dream!
Deep Learning Applications

“AI is the new electricity” - Andrew Ng

Object Detection

Image Segmentation

Medical Imaging

Speech Recognition

Text to Speech

Recommendations

Games
What is a Deep Neural Network?

Each synapse has a weight for neuron activation
Deep Learning Landscape

Training → Inference
Deep Learning Landscape

Training → Inference

**Convolutional Layers** (Feature Extraction)

Conv. Layer → Conv. Layer → ... → Pool. Layer → FC Layer

Summarize features

Intermediate features

“Klaus Advanced Computing Building”
Deep Learning Landscape

Training → Inference

Labelled Datasets

ML Practitioner

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

“Klaus Advanced Computing Building”
Computation Platforms

Training → Inference

HPC cluster

Accelerators

ARM Trillum
NVDLA
Apple Neural Engine
CambriconX
ShiDianNao
Eyeriss

GT CRNCH Summit 2018
Tushar Krishna | Georgia Institute of Technology
November 2, 2018
Challenges in Designing and Deploying AI

- DNN Architecture
- Mapping (Dataflow)
- Accelerator Microarchitecture

Training ➔ Inference

Energy ➔ Runtime
Outline of Talk

Training → Inference

DNN Architecture

Mapping (Dataflow)

Accelerator Microarchitecture

Genesys

MAESTRO

MAERI
Deep Learning Landscape

What if

Training → Inference

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?

Deep Learning not viable for continuous learning

HPC Cluster

ML Practitioner

Labelled Datasets

Deep Learning not viable for continuous learning

“Klaus Advanced Computing Building”

DNN Model (Topology + Weights)

Pool. Layer → FC Layer

What if?
Template for Continuous Learning

Robust

??

Learning Agent

Topology

Weights

Continuous

Inference Agent

Action

Reward

Environment

Learn how to improve at one task

Learn multiple tasks

Accumulated Rewards

GT CRNCH Summit 2018

Tushar Krishna | Georgia Institute of Technology

November 2, 2018
Neuro-Evolutionary (NE) Algorithm

Neural Network (NN) expressed as a graph

**Gene**: Vertex or Edge in the graph

**Genome**: Collection of all genes (i.e., a NN)

Neuro-Evolutionary (NE) Algorithm

Neural Network (NN) expressed as a graph

**Gene**: Vertex or Edge in the graph

**Genome**: Collection of all genes (i.e., a NN)

Properties of NE algorithms

**Algorithmic**
- Robustness
- No Training

Change fitness function

**Systems**
- Massive Parallelism
  - Genomes within Population
  - Genes within a Genome
- No backprop
  - No gradient calculations or storage

**Low Memory Footprint**
- Only store genomes in current generation

**Simple HW-friendly Ops**
- MACs in Inference Crossover and Mutation in Evolution

**NE is viable for continuous learning**
**GeneSys SoC**

Ananda Samajdar, Parth Mannan, Kartikay Garg, and Tushar Krishna  
**GeneSys: Enabling Continuous Learning through Neural Network Evolution in Hardware**  
MICRO 2018
GeneSys ASIC: Runtime and Energy

Inference per generation

Evolution per generation

Runtime (Log Scale)

Energy (Log Scale)

More efficient

100-10000X
Outline of Talk

Training  →  Inference

DNN Architecture

Mapping (Dataflow)

Accelerator Microarchitecture

GeneSys

MAESTRO

MAERI
Why do we need DNN accelerators?

**Millions of Parameters (i.e., weights)**
- Billions of computations
  - Need lots of parallel compute
  - Need to reduce energy

<table>
<thead>
<tr>
<th>DNN Topology</th>
<th>Number of Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet (2012)</td>
<td>3.98M</td>
</tr>
<tr>
<td>VGGnet-16 (2014)</td>
<td>28.25M</td>
</tr>
<tr>
<td>GoogleNet (2015)</td>
<td>6.77M</td>
</tr>
<tr>
<td>Resnet-20 (2016)</td>
<td>0.27M</td>
</tr>
<tr>
<td>Resnet-110 (2016)</td>
<td>1.7M</td>
</tr>
</tbody>
</table>

**Heavy data movement**
- Data Movement Energy Cost
  - DRAM to ALU: 500×
  - Buffer to ALU: 10×
  - PE to ALU: 3×
  - RF to ALU: 1×
  - ALU: 1× (Reference)

This makes CPUs inefficient
This makes GPUs inefficient
Spatial (or Dataflow) Accelerators

- **Millions of Parameters** (i.e., weights)
  - Billions of computations
    - Spread computations across hundreds of ALUs
    - Heavy data movement
      - Reuse data within the array via direct communication

Examples: MIT Eyeriss, Google TPU, ...
High-Dimensional Compute $\rightarrow$ HW

**Convolutional Layer**

- **7D Computation Space**
  - $R \times S \times X \times Y \times C \times K \times N$

- **4D Operand / Result Spaces**
  - **Weights** - $R \times S \times C \times K$
  - **Inputs** - $X \times Y \times C \times N$
  - **Outputs** - $X' \times Y' \times K \times N$

---

**Accelerator HW (ASIC or FPGA or HPC System)**

**Data Reuse**

- **Temporal**
  - DRAM $\rightarrow$ Buf $\rightarrow$ RF $\rightarrow$ *

- **Spatial**

- **Spatio-Temporal**

---

** Millions of non-trivial mappings **

**Energy Benefits = f(Dimension Sizes, Hardware Resources, Dataflow)**

**How do we explore all possible dataflows?**
MAESTRO: Analytical Cost/Benefit Model

Buffer Analysis
- Size Requirement
- Access Count (Energy)

NoC Analysis
- BW Requirement
- NoC Activity Count

Runtime Analysis
- Roofline Throughput
- Expected Runtime

MAESTRO:

Abstract HW Model
Data Reuse Analysis
Communication Analysis
Computation Analysis

DNN Layer Sizes

HW Resources

Mapping (Dataflow)

Describing Dataflows in MAESTRO

for(int x = 0; x < 20; x++)
for(int s = 0; s < 10; s++)
Output[x] += Weight[s] * Input[x+s]

Parameters: (Mapping size, Offset)

MAESTRO: Temporal_Map(1, 1) X -> Spatial_Map(2, 2) S

Time step = 0

* Temporal_Map: to map the same loop variable set to each PE
* Spatial_Map: to map different loop variable sets to each PE
Input: MAESTRO DSL

1 | //Hardware Resource Description
2 | L1Size 64
3 | L2 Size 1024
4 | NoCBW 64
5 | Multicast True
6 | NumPEs 256
7 | ...
8 | //DNN Layer Description
9 | Layer CONV VGG16_C1
10 | K=64;C=3;R=3;S=3;Y=224;X=224
11 | endLayer
12 |
13 | //Mapping (Dataflow) Description
14 | Temporal_Map (1,1) K
15 | -> Temporal_Map (1,1) C
16 | -> Temporal_Map (3,1) Y
17 | -> Tile (3) Y
18 | -> Spatial_Map (1,1) X
19 | -> Unroll R
20 | -> Unroll S

Temporally/Spatially maps iteration variables to each PE
No single dataflow is good for every layer
Outline of Talk

DNN Architecture

Mapping (Dataflow)

Accelerator Microarchitecture

Training

Inference

GeneSys

MAESTRO

MAERI
Myriad Dataflows in DNN Accelerators

- **DNN Topologies**
  - Layer size / shape
  - Layer types: Convolution / Pool / FC / LSTM
  - New sub-structure: e.g., Inception in Googlenet

- **Irregular Networks**
  - Weight Pruning during Training
  - Generated by GeneSys

- **Compiler/Mapper Optimization (i.e., MAESTRO)**
  - Loop reordering
  - Loop tiling size
  - Cross-layer mapping
The current trend for supporting multiple dataflows

- New Dataflow → New Accelerator
  - Data reuse: FlexFlow (2017), Eyeriss (2016), …
  - Cross-layer: Fused CNN (2016)
  - Sparse CNN: SCNN (2017), EIE(2016), …
  - LSTM: ESE (2017), …

Can we have one architectural solution that can handle arbitrary dataflows and provides ~100% utilization?
What is the computation in a DNN?

Compute weighted sum

Independent multiplication

Accumulation of partial products

Our Key insight: Each dataflow translates into neurons of different sizes
The MAERI Abstraction

Multiplier Pool

Adder Pool

Virtual Neuron (VN): Temporary grouping of compute units for an output

How to enable flexible grouping? Reconfigurable Interconnects!
Traffic Patterns in DNN Accelerators*

PB: Prefetch buffer (Global buffer)
NoC: Network-on-Chip (Interconnection network)
PE: Processing element (Compute units)

* Distribution
  e.g. input and weight distribution to PEs

* Reduction
  e.g. partial sum and output reduction

* Local Forwarding
  e.g. input/weight/partial-sum forwarding

* H. Kwon et al., Rethinking NoCs for Spatial DNN Accelerators, NOCS 2017
The MAERI Implementation

Dataflow (from CPU)

Legend
- Simple Switch
- Multiplier Switch
- Adder Switch
- Lookup Table

Micro-Switches

Distribution Network
- Spatial Reuse via Multicasts
- High Bandwidth via fat links

Linear Local Network
- Forwarding of weights
- Spatio-Temporal Reuse

Reduction Network
- High Bandwidth via fat links
- Provably Non-blocking Reductions via forwarding links

Download RTL: http://synergy.ece.gatech.edu/tools/maeri
Example Mapping - Dense CNN

Our Key insight: Each dataflow translates into neurons of different sizes

\[ \sum (W_i X_i) \]

Weights/Inputs

Partial Outputs

VN0

VN1

VN2

inputs

weights

output
Example Mapping - Sparse DNN

Our Key insight: Each dataflow translates into neurons of different sizes.
Our Key insight: Each dataflow translates into neurons of different sizes

\[ \sum (W_i X_i) \]
Performance with Dense Workload

• Total Latency (Runtime) for Convolution

* Normalized to ideal case (100% utilization, Infinite bandwidth)

MAERI reduces CNN runtime upto 65%, 42% in avg.

MAERI reduces LSTM runtime upto 63%, 57% in avg.

LSTMs from Yonghui We, et. al., "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.", Arxiv Preprint, 2016
Takeaways

AI will be pervasive

Thank you!

GW-SW Co-Design of NE Algorithms shows promise for continuous learning at the edge

MAESTRO
Analytical Model for CONV Dataflow Analysis

MAERI
DNN Accelerator with Configurable Interconnects can map Irregular Dataflows