A Scalable Bayesian Inference Accelerator for Unsupervised Learning

Glenn Ko (Harvard University / Stochastic)

Yuji Chai¹, Marco Donato¹,², Paul N. Whatmough¹,³, Thierry Tambe¹, Rob A. Rutenbar⁴, Gu-Yeon Wei¹ and David Brooks¹

Harvard University¹, Tufts University², Arm Research³, University of Pittsburgh⁴
The exponential data growth problem

Humans are generating data at an exponentially increasing rate.

Machine learning allows us to extract useful information from data without much human intervention.
Did we solve the problem?

Up to 80% of time of a data scientist is spent on sourcing and cleaning data.

Deep learning requires large labeled datasets (via data annotation services).

“The future of AI will be about less data, not more”
- Jan 19’ Harvard Business Review

Ref: CIFAR-10
Humans need a lot less data than machines to generalize about and draw conclusions.

Babies can learn by crawling around and playing with toys or simply observing the behavior of adults.

Probabilistic Machine Learning

*Or also called Bayesian learning*

Probability is used to represent uncertainty about the relationship being learned. Our beliefs about the true relationship are expressed in a probability distribution.

For learning and prediction,

**Bayesian Inference:**

How one should update one’s beliefs upon observing data.

**Bayes’ theorem:**

\[
P(Hypothesis|Data) = \frac{P(Data|Hypothesis)P(Hypothesis)}{P(Data)}
\]
Known to be More Powerful for

1. Online learning from small chunks of scarcely or unlabeled data

2. Finding a distribution instead of a point estimate

3. Representing and computing with uncertainty

COVID-19 Predictions

Applications
- Biomedical
- Robotics
- Autonomous driving
- Finance

Problem
CPUs and GPUs are inefficient for Bayesian inference

Ref: Dehning et al. Science 2020, Kendell et al. NeurIPS 2017
PGMA: Probabilistic Graphical Models Accelerator

• First silicon accelerator for Bayesian inference

• Algorithm-hardware co-design for parallel MCMC inference

• To demonstrates efficient mobile implementation of Bayesian inference using unsupervised perceptual tasks
  - Stereo matching
  - Image restoration
  - Image segmentation
  - Sound source separation
SM5: A 16nm SoC for ML-Powered IoT Devices

- TSMC 16nm FFC
- 25mm² (5mm x 5mm) SoC
- PGMA die area: 2.3mm x 1.3mm
- Designed using CHIPKIT
- Short design cycle: RTL to tape-out in 3 months by 5 people (2 postdocs + 3 PhDs)

Harvard ML Research Platform + CHIPKIT

• SoC platform for architecture and systems research
• CHIPKIT: Agile research test chip design methodology

SM5 SoC Architecture

Embedded FPGA (eFPGA)
- Logic Tile
- Logic Tile
- DSP Tile
- DSP Tile

Arm Cortex-A53 CPU (A53)
- Arm A53 SIMD 64KB L1$
- Arm A53 SIMD 64KB L1$
- 2MB L2 Cache

On-Chip 2MB SRAM (2x Banks)

PGMA
- Sub-Graph Tile (SGT)
- Global Graph Cache (GGC) 300 KB
- Sub-Graph Tile (SGT)

ARM CoreLink NIC-400 128-bit

AHB 32-bit

UARTs GPIO Debug

Input Data Buffer (IDB)
- 20KB
- 320KB

Gibbs Sampler

Local Graph Cache (LGC)

Ref: Ko et al., VLSI 2020
Models, Inference and Applications

Model

Markov Random Field (MRF) - A generalization over Ising model
Various other probabilistic models (HMM, regression, etc.)

Inference

Gibbs sampling - A Markov Chain Monte Carlo (MCMC) algorithm derived from statistical physics

Application

computer vision, audio processing, combinatorial optimization, computational biology, recommender system, topic modeling, etc.
Example: Image restoration
Mapping to Markov Random Field (MRF)

(Pixel-labeling)

Damaged Image

Unsupervised Learning

Reconstructed image

Markov Random Field

Input pixels

Output labels
Gibbs sampling on MRF

while (< max Gibbs sampling iterations)
for each node in an image
sample()
Why is it hard to accelerate?

Gibbs sampling on MRF

1: Initialize $x^0$
2: for $t = 0$ to $T$ do
3: for $i = 0$ to $N$ do
4: $x_i^{(t+1)} \sim P(x_i | x_{north}^{(t)}, x_{south}^{(t)}, x_{west}^{(t)}, x_{east}^{(t)})$
5: end for
6: end for
7: return $x$

Sampling depends on the previous state and the dependency on previous loop iteration makes parallel programming hard.
Gibbs Sampler (GS)

- Supports up to 64 states (labels) per node
- 32b variable fixed point arithmetic
- Tightly coupled PRNG
- Iterative architecture for minimal footprint

Ref: Ko et al., VLSI 2020
Two-levels of parallelism

Chromatic Gibbs sampling:
Sample conditionally independent nodes concurrently

Asynchronous Gibbs sampling:
Sample different tiles in parallel as if they are separate images

Two-level Parallel Gibbs Sampling
while (< max Gibbs sampling iterations)
for each tile in an image
  while (< max tile sampling iterations)
    for each node in a tile
      sample()
PGMA: Probabilistic Graphical Models Accelerator

Sub-Graph Tile (SGT)

Input Data Buffer

Local Graph Cache

Global Graph Cache

640x480

Sub-Graph Tile (SGT)

Local Labels Cache

Input Data Buffer

Ref: Ko et al., VLSI 2020
Unsupervised perceptual tasks

Four example applications:
• Image restoration
• Stereo matching
• Image segmentation
• Sound source separation

Features:
• No labeled dataset
• Completely unsupervised
• Both training and inference on-the-fly

Ref: Ko et al., VLSI 2020
Multi-threaded Server-Class CPU

Machine: Intel(R) Xeon(R) CPU E5-2697A v4
Parallelism: Chromatic Gibbs sampling
Application: Stereo matching - 16 labels

< 6X speedup
Comparison with off-the-self embedded platforms

Nvidia Jetson TX2

Xilinx Zynq ZCU102

48x throughput improvement per Watt
(2108x vs Arm A57, single-thread)

247x throughput improvement per Watt

Parallelism: Chromatic Gibbs sampling

Ref: Ko et al., FPL, 2020. Ko et al. VLSI, 2020
SoC Results: vs. A53 and eFPGA

Achieves 1380x throughput improvements over Arm A53

Achieves 1965x throughput per Watt improvements over Arm A53

Ref: Ko et al., VLSI 2020
SoC Results: vs. A53 and eFPGA

Achieves 4162x throughput per Watt per mm² over Arm A53

PGMA achieves 1965x throughput per Watt improvements over Arm A53

Ref: Ko et al., VLSI 2020
Scaling within SGT

Can add GS’s for linear increase in throughput

Sub-Graph Tile (SGT)

Input Data Buffer

Local Graph Cache
Scaling with more SGT

Can add SGT’s for linear increase in throughput
Summary

• **PGMA - Probabilistic Graphical Models Accelerator**
  - First silicon Bayesian inference accelerator.
  - Can run various probabilistic models including MRF, HMM and more.
  - Solves various applications including computer vision, audio processing, recommender systems, topic modeling, combinatorial optimization, etc.

• **Scalable Bayesian inference accelerator architecture**
  - Algorithm-hardware co-design to enable parallelism in natively sequential algorithm.
  - Hierarchical architecture with two-levels of parallelism.
  - Energy-efficient mobile implementation for real-time unsupervised perceptual tasks.
  - *Stay tuned for server-class version for cloud applications.*

• **Rapid research SoC design and implementation using CHIPKIT**
  - Harvard’s open-source framework for chip design and testing.
Acknowledgments

• **Contributors:** Yuji Chai, Marco Donato, Paul N. Whatmough, Thierry Tambe, Rob A. Rutenbar, David Brooks and Gu-Yeon Wei

• Research sponsored by DARPA CRAFT and DSSoC programs, SRC JUMP ADA, Intel and Arm