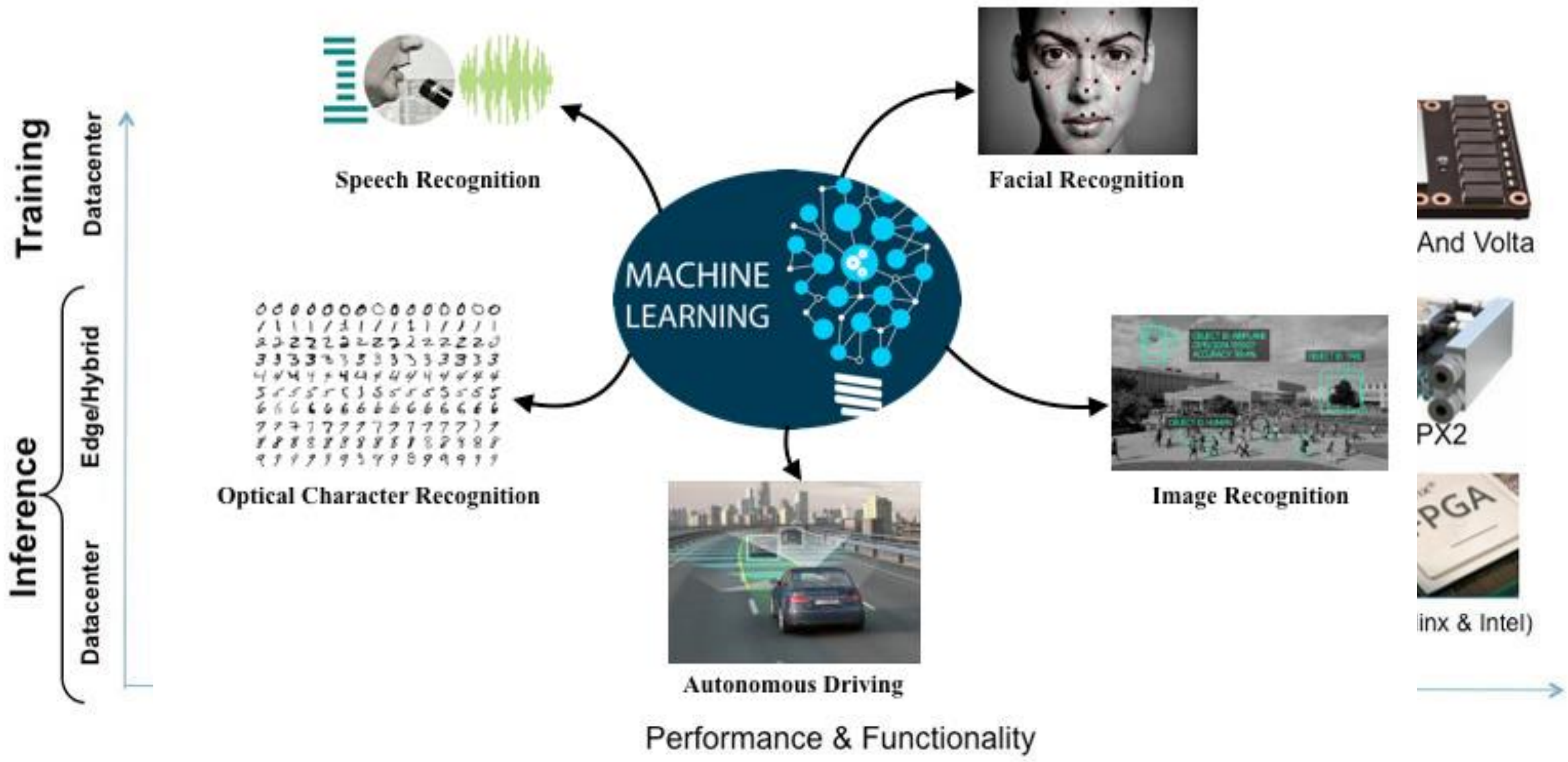
TaxoNN: A Light-Weight Accelerator for Deep Neural Network Training

Reza Hojabr[1], Kamyar Givaki[1*], Kossar Pourahmadi[1*], Parsa Nooralinejad[1*], Ahmad Khonsari[1,2], Dara Rahmati[2], M. Hassan Najafi[3]

[1]School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
[2]School of Computer Sciences, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran
[3]School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, LA, USA

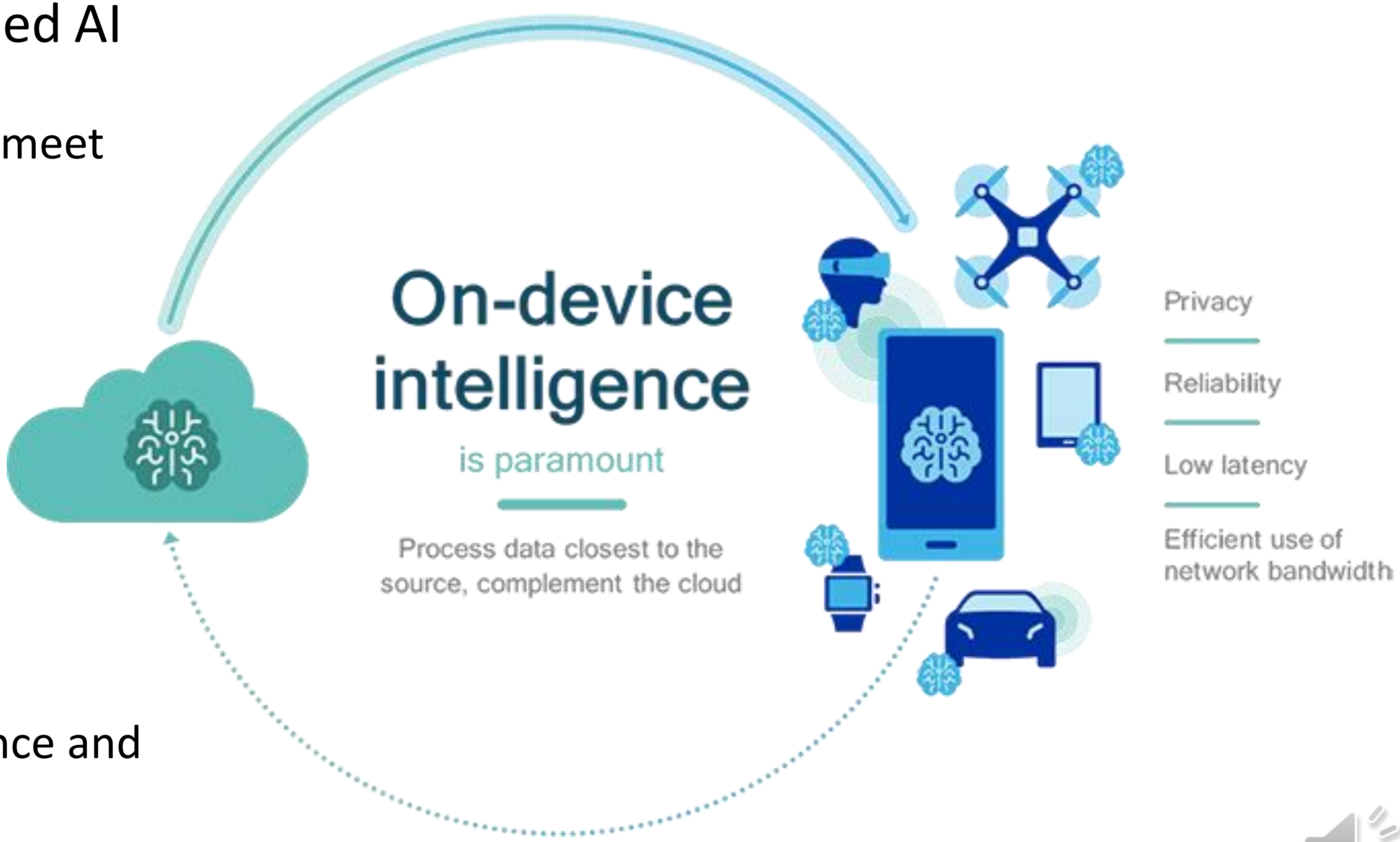
2020 IEEE International Symposium on Circuits and Systems
Virtual, October 10-21, 2020

Deep Learning Accelerators



Edge AI – The Next Wave of AI

- The need for embedded AI accelerators
  - GPU and TPU cannot meet the requirements:
    - Power consumption
    - Price $
- On-device training
  - Efficient use of BW
  - Privacy
  - Reliability
  - Low latency
- TaxoNN
  - Supports both inference and training



Main Contributions

Split the SGD algorithm into smaller computation units

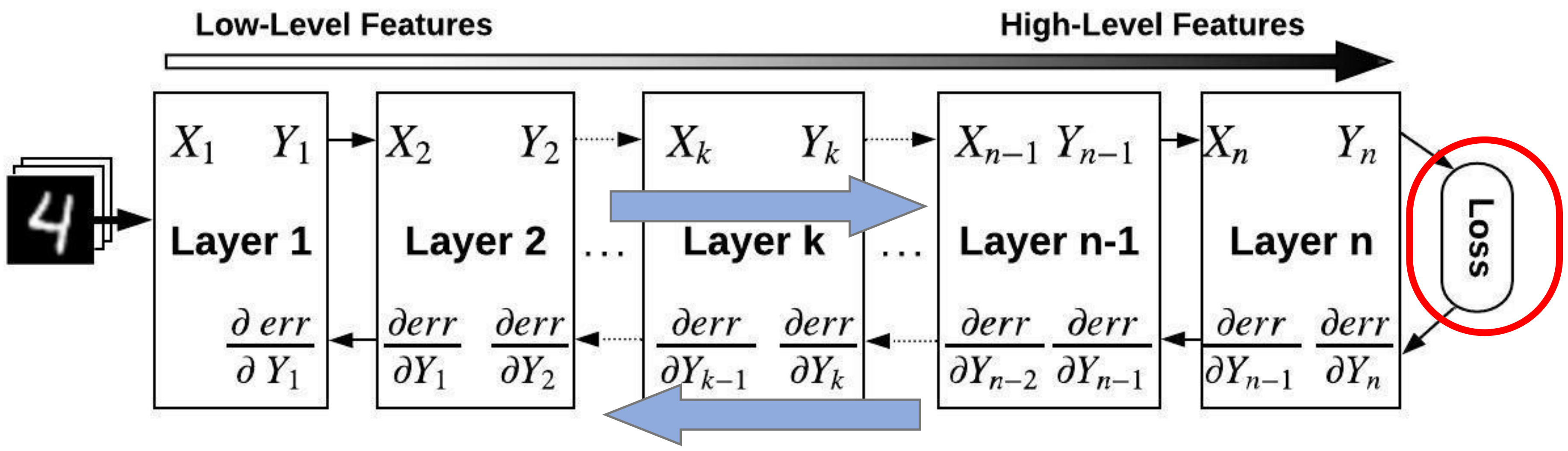
TaxoNN

Adding the training ability to the baseline inference only Processing Elements

Low-bitwidth operations for different layers

Stochastic Gradient Descent

$$W_i = W_i - \alpha \frac{\partial error}{\partial W_i}$$



Simplifying SGD Algorithm

Y_i is the output of i^{th} layer

$$W_i = W_i - \alpha \frac{\partial error}{\partial W_i}$$

Chain rule

$$\frac{\partial error}{\partial W_i} = \frac{\partial error}{\partial Y_{i+1}} \times \frac{\partial Y_{i+1}}{\partial Y_i} \times \frac{\partial Y_i}{\partial W_i}$$

$$\frac{\partial Y_{i+1}}{\partial Y_i} = \frac{\partial f_{i+1}(W_{i+1}Y_i)}{\partial Y_i} = W_{i+1}\, f'_{i+1}(W_{i+1}Y_i)$$

$$\frac{\partial Y_i}{\partial W_i} = \frac{\partial f_i(W_iY_{i-1})}{\partial W_i} = Y_{i-1}\, f'_i(W_iY_{i-1})$$

$$\frac{\partial error}{\partial W_i} = \underbrace{\frac{\partial error}{\partial Y_{i+1}} \times f'_{i+1}}_{G_{i+1}} \times W_{i+1} \times f'_i \times \underbrace{Y_{i-1}}_{X_i}$$

$$G_i = G_{i+1} \times W_{i+1} \times f'_i$$

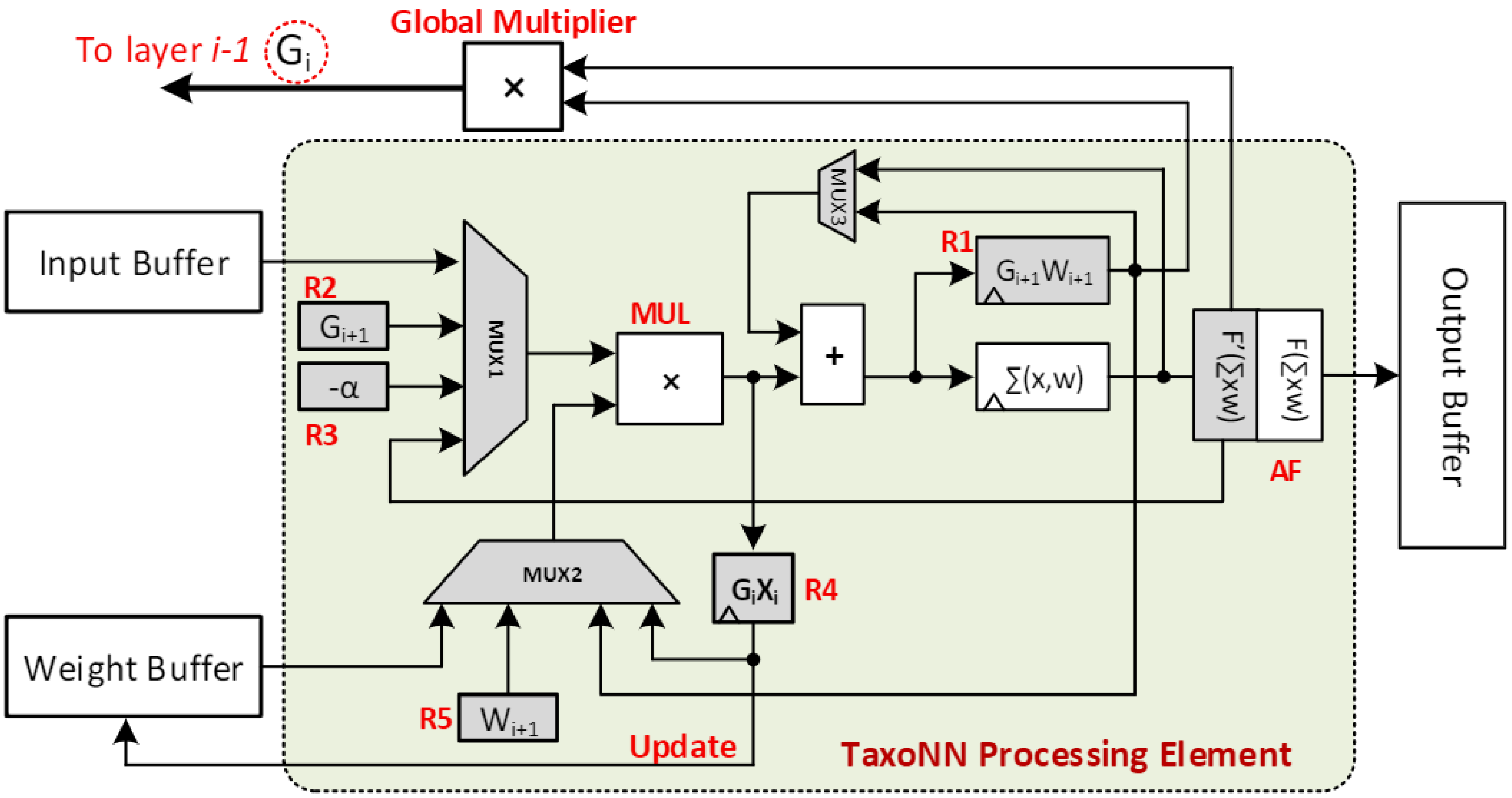
$$\frac{\partial error}{\partial W_i} = G_i \times X_i$$

$f_i()$ is the activation function of the i^{th} layer

$f'()$ is the derivation of the activation function

TaxoNN Architecture

$$W_i = W_i - \alpha \frac{\partial error}{\partial W_i} \qquad \frac{\partial error}{\partial W_i} = G_i \times X_i$$



Timing and Pipelining

$$W_i = W_i - \alpha \frac{\partial error}{\partial W_i}$$

$$\frac{\partial error}{\partial W_i} = G_i \times X_i$$

G_i: *a vec of size* N_i

N_i: *#of neuron in layer i*

N_i = # of neurons in layer *i*

Layer n-2

$\frac{\partial error}{\partial W_{n-2}} = G_{n-2} \times X_{n-2}$

$G_{n-2} = G_{n-1} \times W_{n-1} \times f'_{n-2}$

$N_{n-1} + N_{n-2}$ cycles

• • •

Layer n-1

$\frac{\partial error}{\partial W_{n-1}} = G_{n-1} \times X_{n-1}$

$G_{n-1} = G_n \times W_n \times f'_{n-1}$

$N_n + N_{n-1}$ cycles

Layer *n* and LossFunc

$\frac{\partial error}{\partial W_n} = G_n \times X_n$

$G_n = \frac{\partial error}{\partial Y_n} \times f'_n$

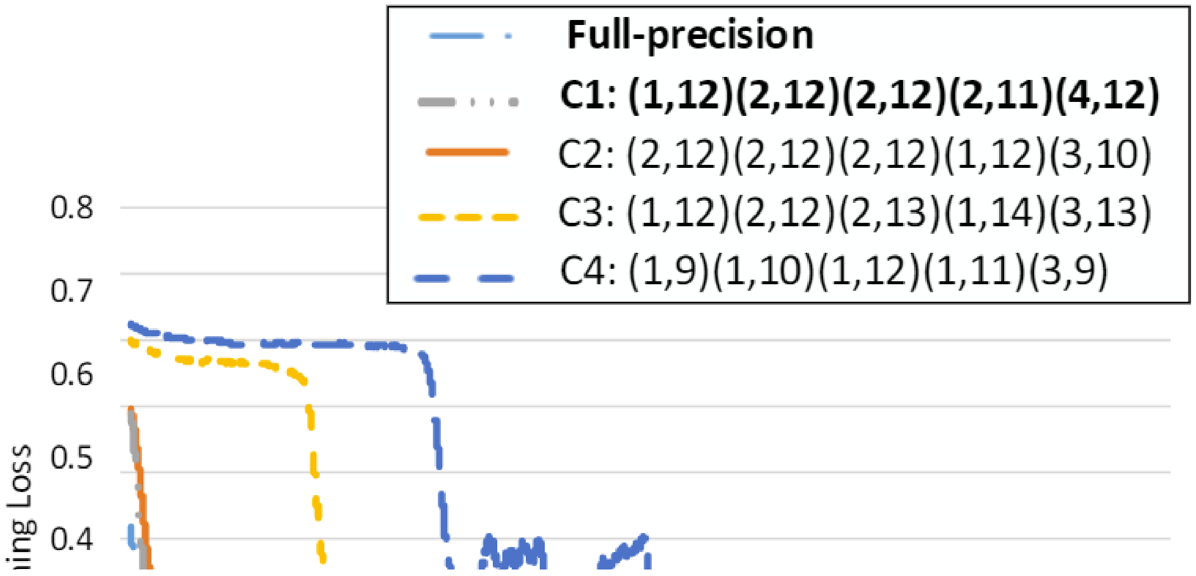
$2 \times N_n$ cycles

Back-Propagation →

$N_n + \sum N_i$ cycles

Performance Evaluation

- Network
  - LeNet
- Dataset
  - MNIST
  - CIFAR10
  - SVHN
- Developed a cycle-accurate simulator
- Full-precision results are extracted from TensorFlow



Dataset	Precision per Layer (I,F)	TaxoNN Accuracy	Full-precision Accuracy
MNIST	(2,12)(2,12)(2,12)(1,12)(3,10)	99.1	99.4
CIFAR10	(2,10)(2,11)(1,10)(1,13)(2,13)	84.1	85.4
SVHN	(1,12)(2,12)(2,12)(2,11)(4,12)	94.7	96.0

Hardware Cost

- Implemented TaxoNN in RTL Verilog
- Synthesized using DC with 45nm technology

Bitwidth	21	20	19	18	17	16	15	14	13	Average Area Overhead
Eyeriss	14.3	13.1	11.8	11.1	10.6	10.1	9.7	9.0	8.1	
TaxoNN	15.5	14.3	12.9	12.1	11.7	11.2	10.6	9.9	9.0	
Overhead	8.3%	9.2%	9.1%	8.6%	10.0%	10.8%	8.8%	9.8%	10.5%	9.5%

Bitwidth	21	20	19	18	17	16	15	14	13	Average Power Overhead
Eyeriss	4.54	4.48	4.42	4.31	4.22	4.10	3.98	3.88	3.75	
TaxoNN	4.84	4.78	4.70	4.65	4.49	4.31	4.15	4.13	4.04	
Overhead	6.5%	6.7%	6.2%	7.9%	6.5%	5.2%	4.3%	6.5%	7.7%	6.4%

Dataset	Power Reduction	Area Reduction
MNIST	2.1×	1.7×
CIFAR10	2.3×	1.8×
SVHN	1.9×	1.5×

TaxoNN Summary

- TaxoNN: a light-wight accelerator for DNN training
- A novel method to unroll and parallelize the SGD algorithm
- A fine-grained and optimized datapath to perform the SGD
- Evaluated TaxoNN using low-bitwidth operations for each layer
- 1.65x area and 2.1x power saving at the cost of 0.97% misclassification rate compared to full-precision

Reza Hojabr

r.hojabr@ut.ac.ir