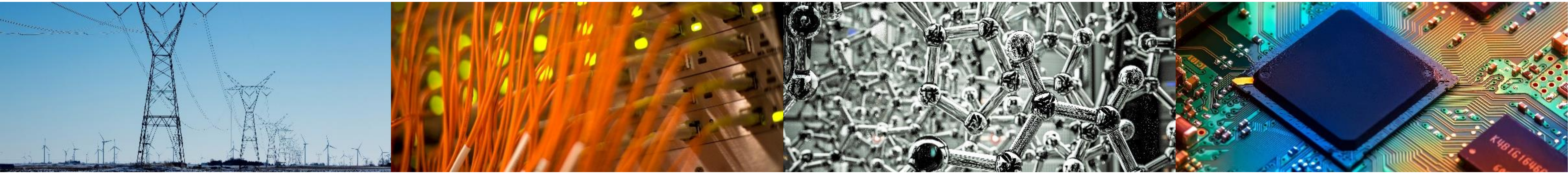
ASPDAC 2021: 26th Asia and South Pacific Design Automation Conference

Mixed Precision Quantization for ReRAM-based DNN Inference Accelerators

Sitao Huang[1], Aayush Ankit[2], Plinio Silveira[3], Rodrigo Antunes[3], Sai Rahul Chalamalasetti[4], Izzat El Hajj[5], Dong Eun Kim[2], Glaucimar Aguiar[3], Pedro Bruel[4,6], Sergey Serebryakov[4], Cong Xu[4], Can Li[4], Paolo Faraboschi[4], John Paul Strachan[4], Deming Chen[1], Kaushik Roy[2], Wen-mei Hwu[1], and Dejan Milojicic[4]

[1]University of Illinois at Urbana-Champaign, USA [2]Purdue University, USA [3]Hewlett Packard Enterprise, Brazil
[4]Hewlett Packard Enterprise, USA [5]American University of Beirut, Lebanon [6]University of São Paulo, Brazil

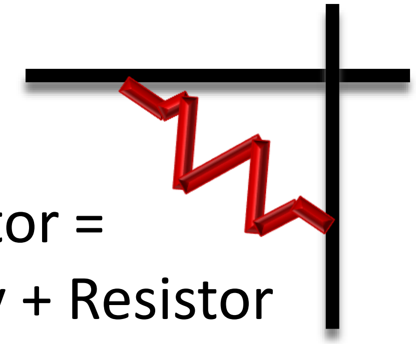
{shuang91,dchen,w-hwu}@illinois.edu, {aankit,kim2976,kaushik}@purdue.edu, izzat.elhajj@aub.edu.lb, {firstname.lastname}@hpe.com

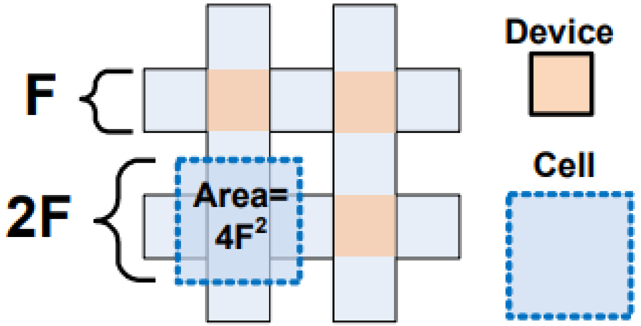


ILLINOIS
Electrical & Computer Engineering
GRAINGER COLLEGE OF ENGINEERING

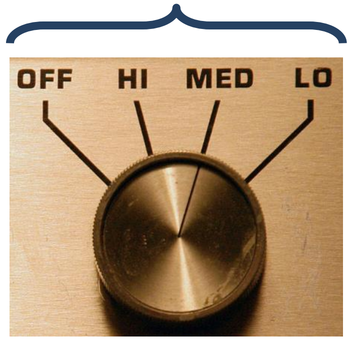
Background: Memristive Crossbars

Memristor =
Memory + Resistor





States



Memory aspects:
High Density + **On-Die Storage**

Array Density: 160MB/mm^2
- Mitigates off-chip data access

>100×

Compact cell structure

Cell area is $4F^2$ vs $120F^2$ for CMOS (SRAM).

30×

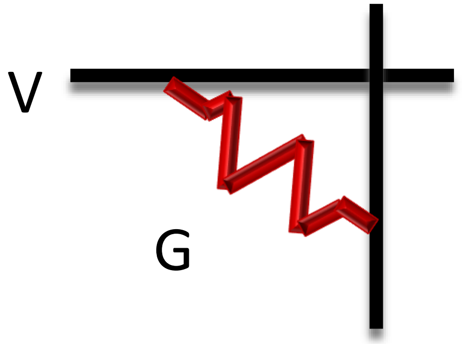
Tunable Resistance

2-6 bits per cell vs 1-bit for CMOS (SRAM).

6×

A. Ankit et al. PUMA: A Programmable Ultra-efficient Memristor-based Accelerator for Machine Learning Inference. ASPLOS 2019.

Background: Matrix-Vector Multiplication Unit in DNN Accelerators



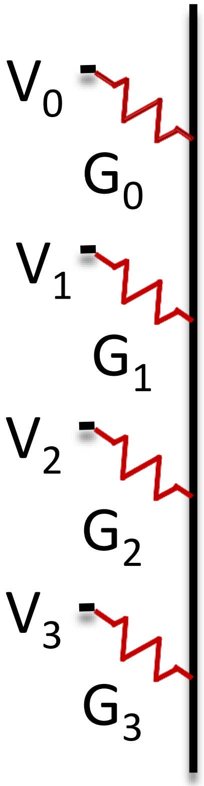
$$I = V * G$$

Compute Aspects:
Analog Multiplication

Analog MVM: 1.34pJ/op*
- Efficient compute

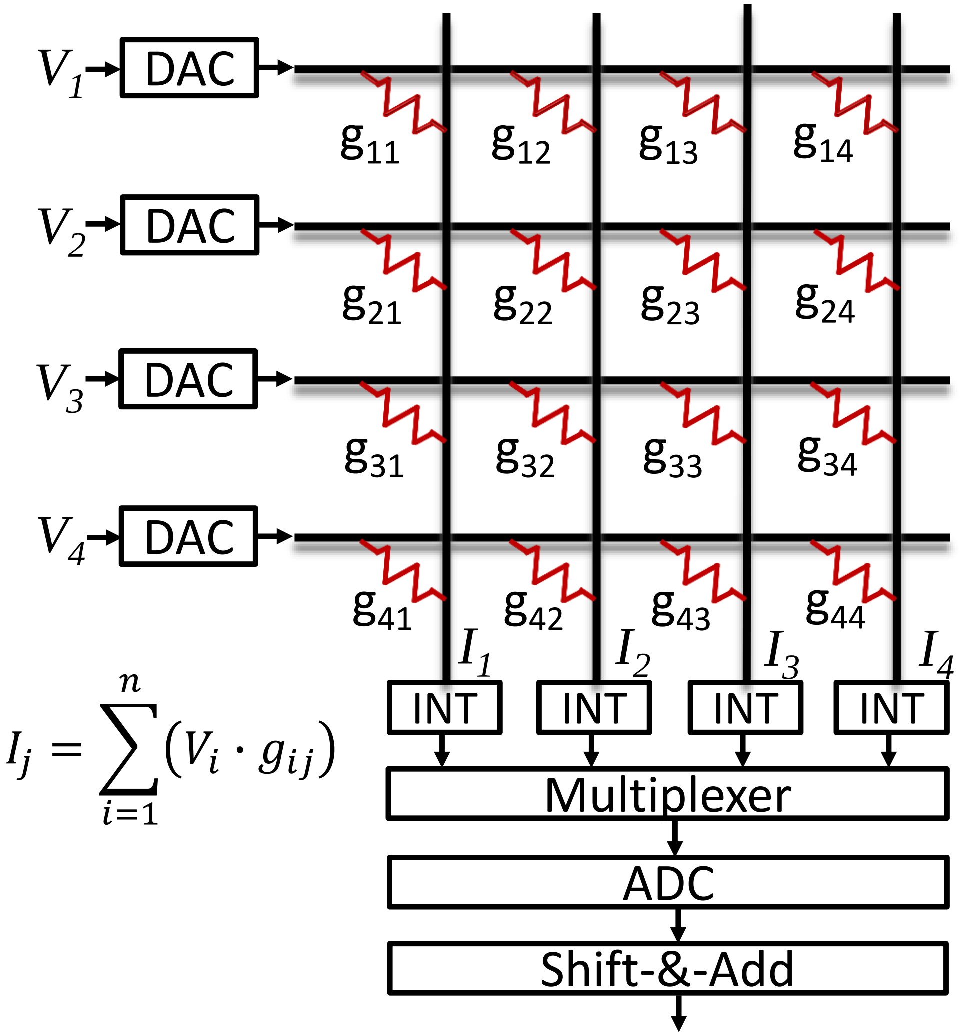
>4×

*32nm technology



$$I = I_0 + I_1 + I_2 + I_3 = \Sigma V_i G_i$$

Analog Dot Product

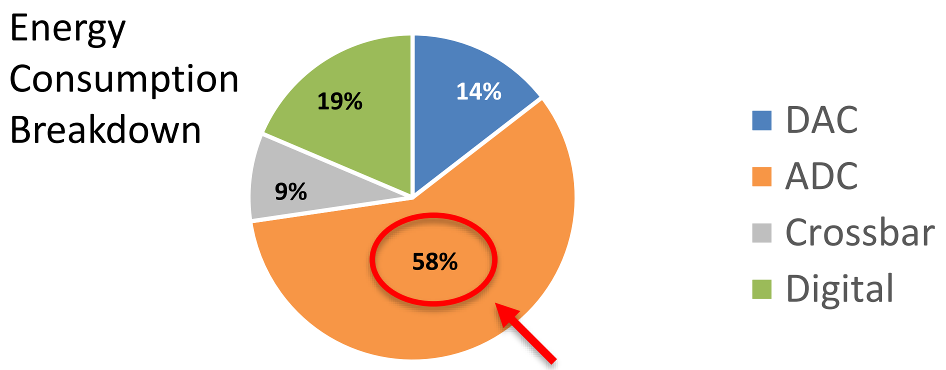


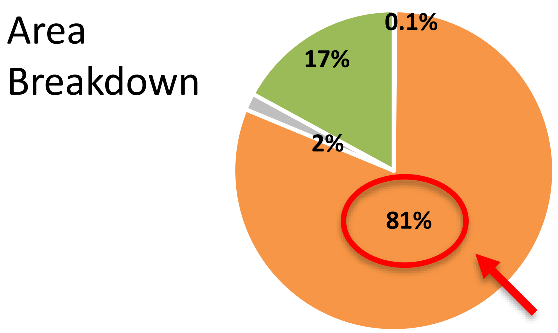
$$I_j = \sum_{i=1}^{n} (V_i \cdot g_{ij})$$

Analog Matrix-Vector Multiplication Unit (MVMU)

Motivation: Cost of Analog-to-Digital Convertors (ADC)

- Energy consumption and latency of MVMU is typically dominated by ADC





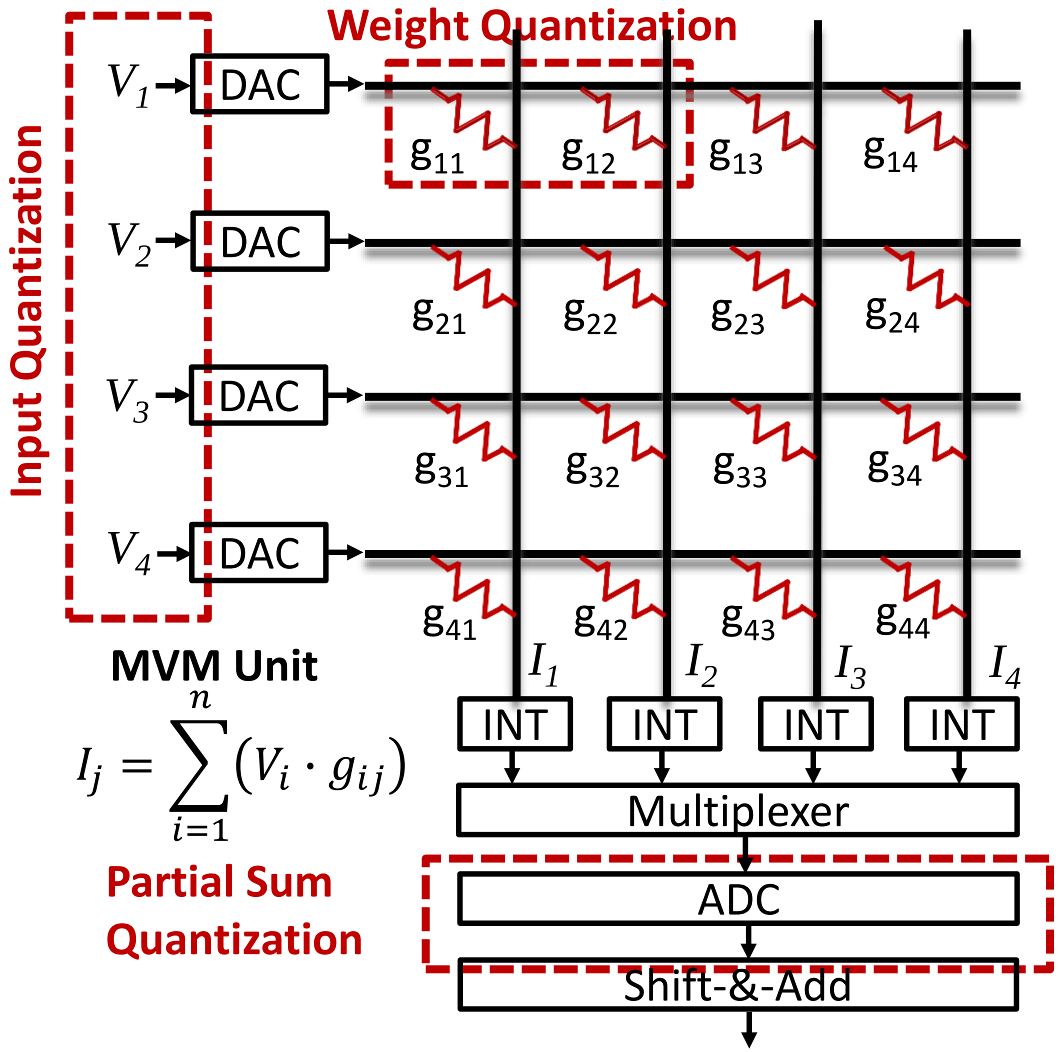
- Reducing ADC resolution can save MVMU energy and latency

ADC Resolution (bits)	Energy: LSTM (24 ReRAM tiles)		Energy: MLP (9 ReRAM tiles)		Latency: LSTM (24 ReRAM tiles)		Latency: MLP (9 ReRAM tiles)	
	Energy (µJ)	Reduction	Energy (µJ)	Reduction	Cycles	Reduction	Cycles	Reduction
8 (baseline)	65.1	-	18.7	-	48589	-	23923	-
6	45.1	30.7%	12.8	31.5%	39349	19.0%	18883	21.1%
4	30.3	**53.5%**	84.5	**54.7%**	30109	**38.0%**	13843	**42.1%**
2	20.6	68.3%	56.8	69.6%	20869	57.1%	8803	63.2%
1	17.8	72.7%	48.8	73.9%	16293	66.5%	6337	73.5%

Quantization in ReRAM Accelerators

- Quantization in ReRAM accelerators
  - **Weight quantization**: Use few bits to represent weights in crossbars
  - **Input quantization**: Quantize inputs (activations) in digital domain
  - **Partial sum (ADC) quantization**: ADC produces lower bitwidth digital outputs for better efficiency

- Benefits
  - Lower energy consumption
  - Lower processing latency
  - Higher area efficiency



Design Space Search Problem and Challenges

- Design Space Search problem
  - Given the *design space* $S = P_1 \times P_2 \times \cdots \times P_n$, and *cost function* $f(s)$,
  - Find the optimal $s^* = (p_1, p_2, \dots, p_n) \in S, \quad \text{s.t.} \quad s^* = \operatorname{argmin} f(s)$

Challenges

- The design space is enormous, even for small networks
  - Example: LeNet-5 design points $= 4^{16} = 4{,}294{,}967{,}296$ (only consider weights and input quantization)
  - Design points in VGG16:

$$(60 \times 60 \times 60 \times 9)^{16}$$

=41604486817032344297373778486959123571469861315936256000000000000000000000000000000000000000000000000

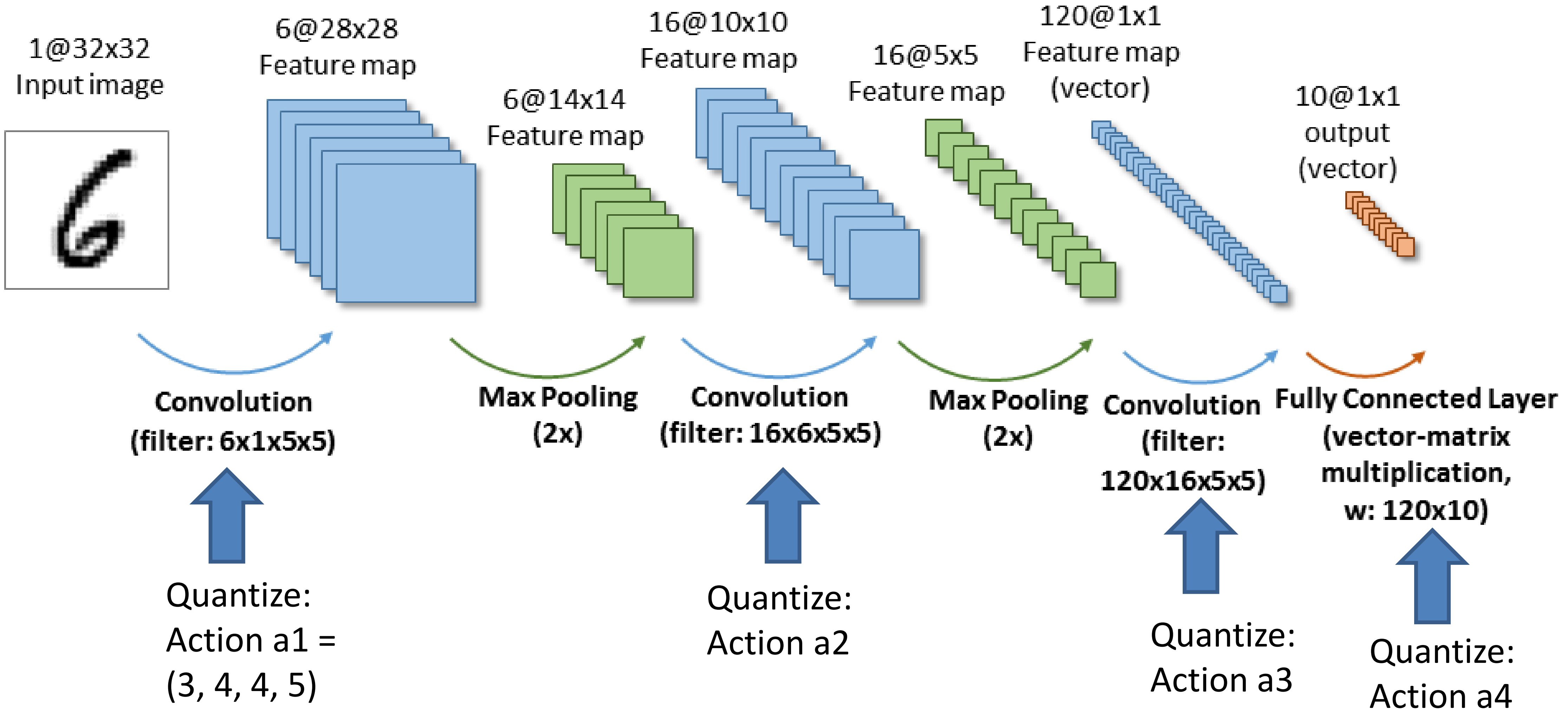
$\approx 4.16 \times 10^{100}$ > # of atoms in the universe (10^{82})

Parameters	Values
Weight Bitwidth	4, 8, 16, 32
Weight Bitwidth (fractional part)	1, 2, …, (weight bitwidth - 1)
Input Bitwitdth	4, 8, 16, 32
Input Bitwidth (fractional part)	1, 2, …, (input bitwidth - 1)
Accumulation Bitwidth	4, 8, 16, 32
Accumulation Bitwidth (fractional part)	1, 2, …, (accumulation bitwdith - 1)
ADC Precision	1, 2, 3, 4, 5, 6, 7, 8, 9

- Non-differentiable cost function (DNN accuracy/energy/latency/area)
- Cost function evaluation can be expensive (simulation)

DNN Quantization as a Reinforcement Learning (RL) Problem

Example: Quantizing LeNet-5



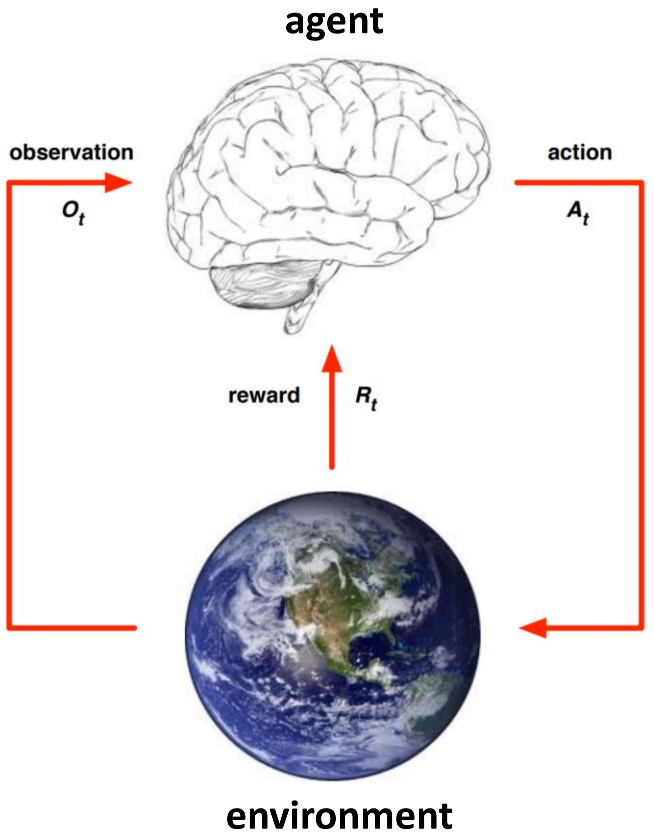
Action = (Layer1_activation_INT, Layer1_activation_FRAC, Layer1_weight_INT, Layer1_weight_FRAC)

Mixed Precision Quantization for ReRAM DNN Accelerators with RL

RL setting: *agent* interacts with *environment* and learns the best *policy* to take *actions* in certain *states* of the environment

- Markov Decision Process (MDP) model: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$
  - State space $\mathcal{S}$: all possible configurations of the DNN
  - Action space $\mathcal{A}$: all possible configurations for a layer in DNN
  - Transition function $\mathcal{P}$: quantize DNN layer by layer
  - Reward $\mathcal{R}$: function of inference accuracy, power, and latency
  - Discount factor γ: set to 1 (finite horizon)
- Policy π: a function that maps state space to action space
  - Tells the agent what action *a* to take given the current state *s:*

$$a = \pi(s)$$



Source: David Silver. Introduction to Reinforcement Learning.

Mixed Precision Quantization for ReRAM DNN Accelerators with RL

RL setting: *agent* interacts with *environment* and learns the best *policy* to take *actions* in certain *states* of the environment

Reward $\mathcal{R}$: function of accuracy, power, and latency

- Cost of accuracy loss due to quantization:

  Cross-entropy loss of the ***quantized DNN*** and the ***original model***

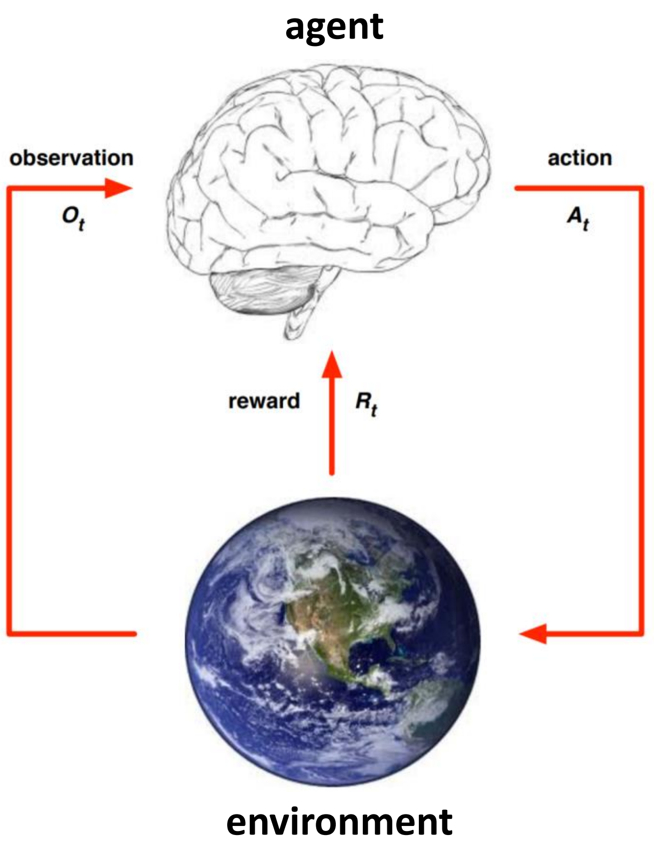
$$Cost_{\text{accuracy}} = Loss_{\text{quantization}} - Loss_{\text{original}}$$

- Cost of hardware (estimates of power and latency):

$$Cost_{\text{hardware}} = \sum_i \alpha^{B^i_{\text{ADC}}} \left(f^i_{\text{input}} \frac{B^i_{\text{input}}}{B_{\text{full}}} + f^i_{\text{weight}} \frac{B^i_{\text{weight}}}{B_{\text{full}}} \right)$$

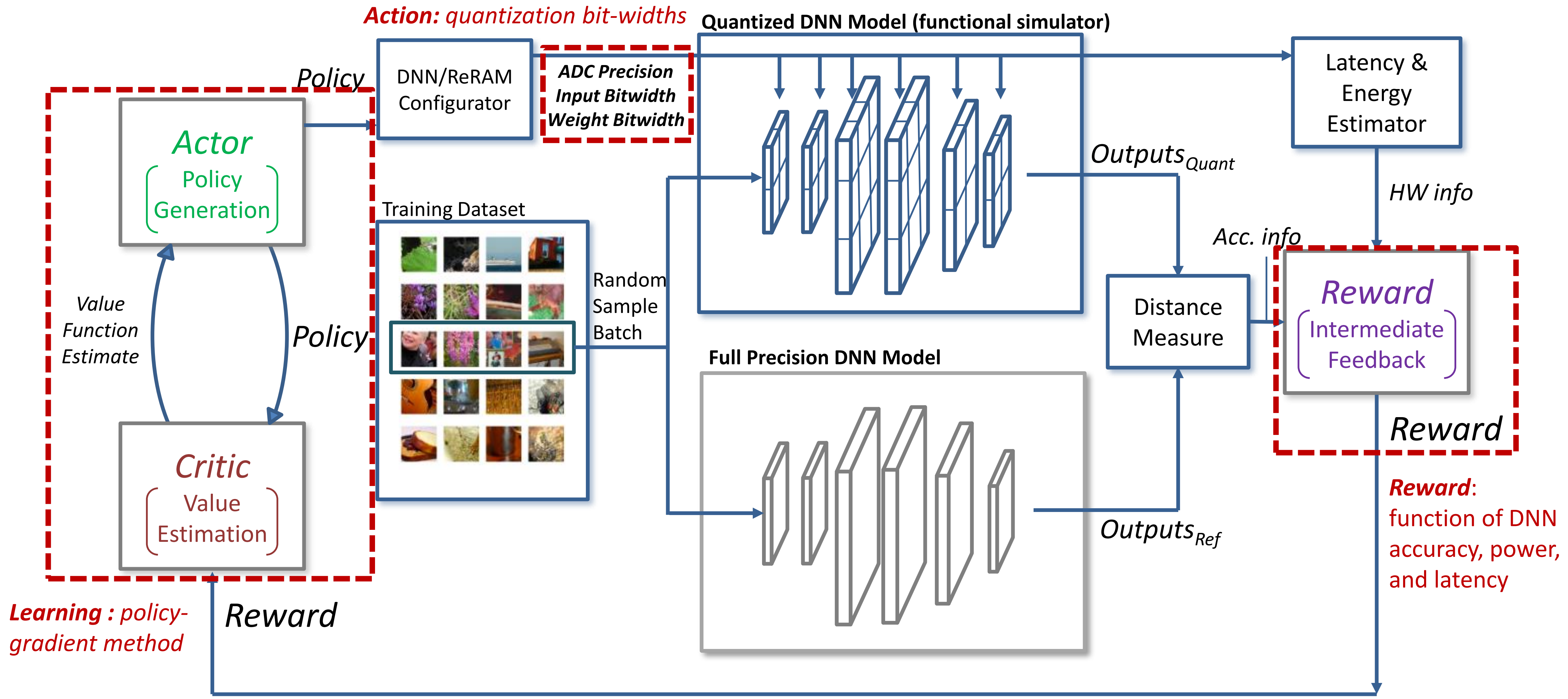
- Reward: $Reward = -T(Cost_{\text{accuracy}}) - Cost_{\text{hardware}}$

  where $T_t(x) = \infty \cdot \mathbb{1}_{x>t} + x$



Source: David Silver. Introduction to Reinforcement Learning.

Mixed Precision Quantization for ReRAM DNN Accelerators with RL



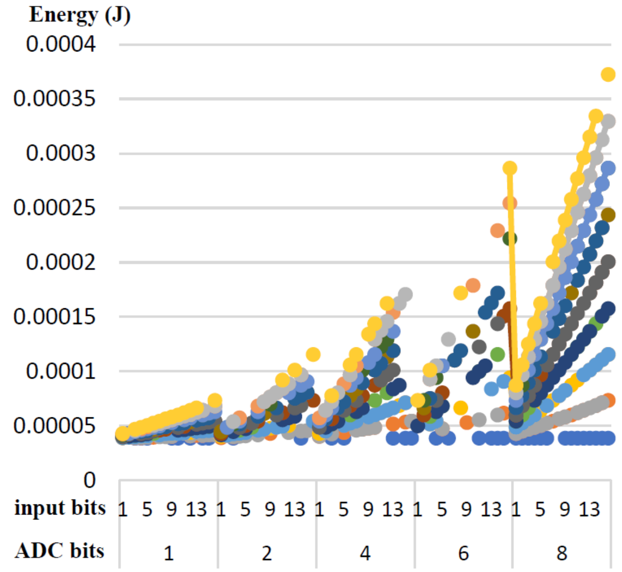
Learning: actor-critic based method

Evaluation

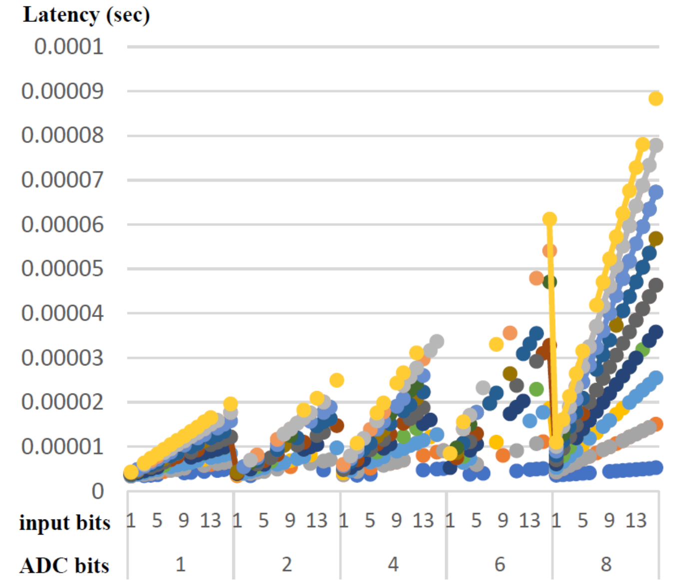
- Benefits of Quantization for DNN Layer
  - Profile certain layers of DNN with *all* possible quantization configurations
  - Study how do quantization schemes change the energy and latency of a DNN layer running on ReRAM DNN accelerators
  - No search flow involved

- Mixed Precision Quantization Search Flow
  - Enable RL based search flow
  - Study the quality of the search results
  - Quantify the benefits of quantization for complete DNN running on ReRAM accelerators

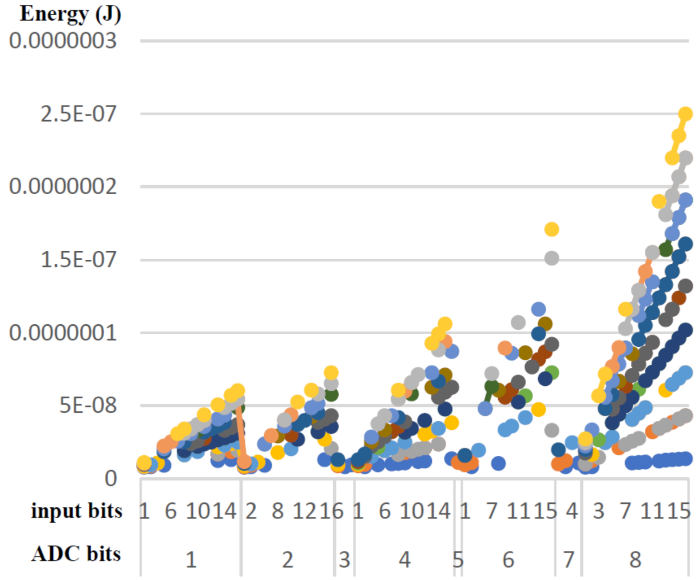
Evaluation: Benefits of Quantization for DNN Layer



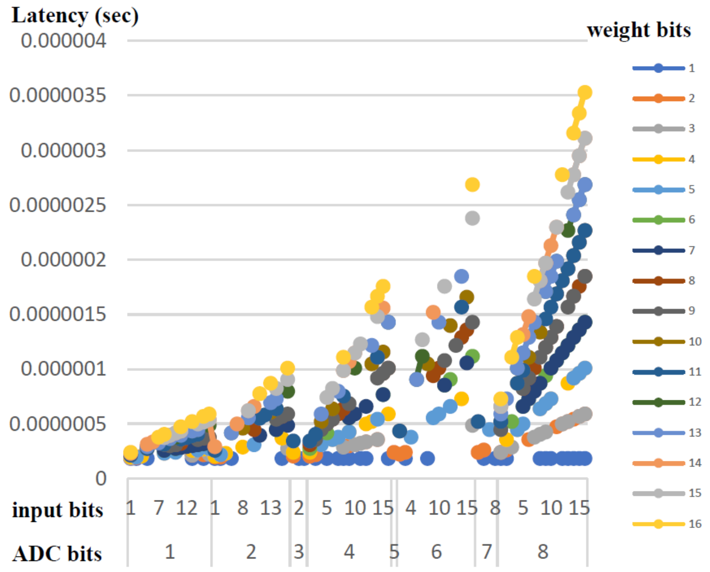
(a) LeNet conv3 energy



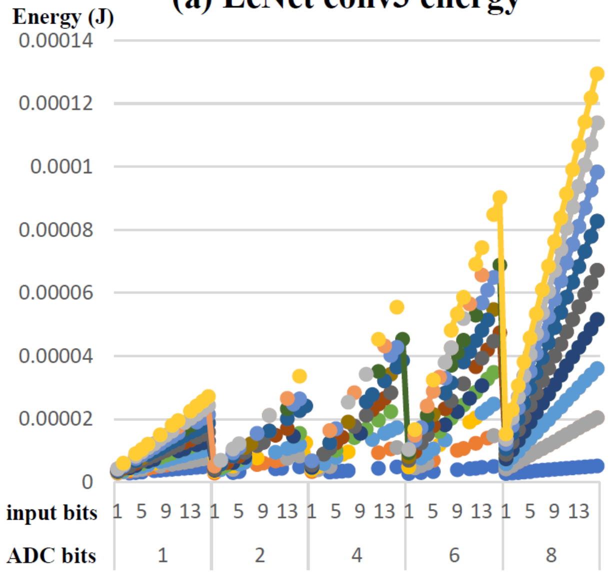
(b) LeNet conv3 latency



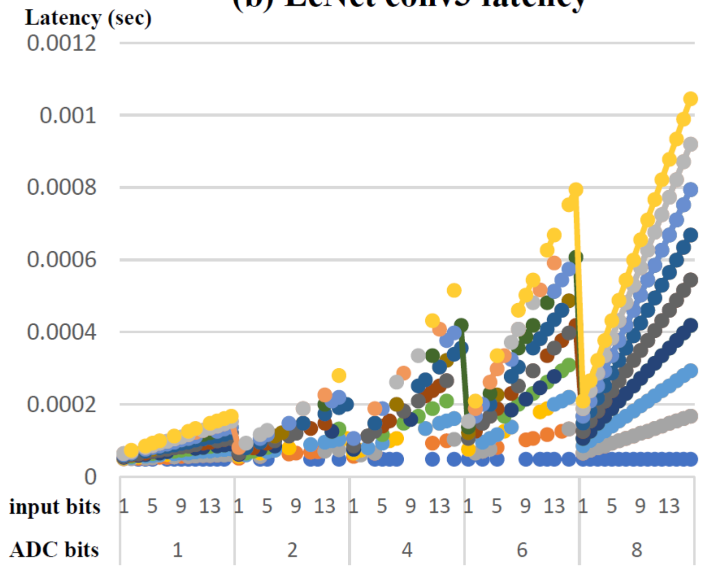
(c) LeNet FC energy



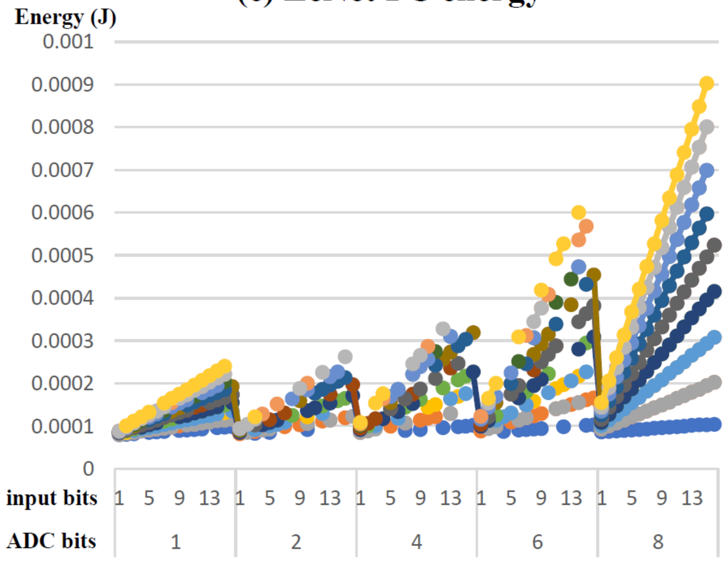
(d) LeNet FC latency



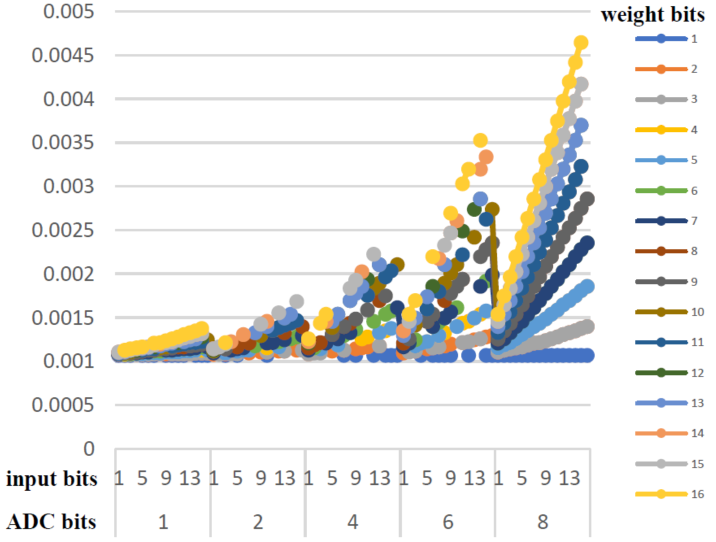
(e) VGG16 conv1 energy



(f) VGG16 conv1 latency



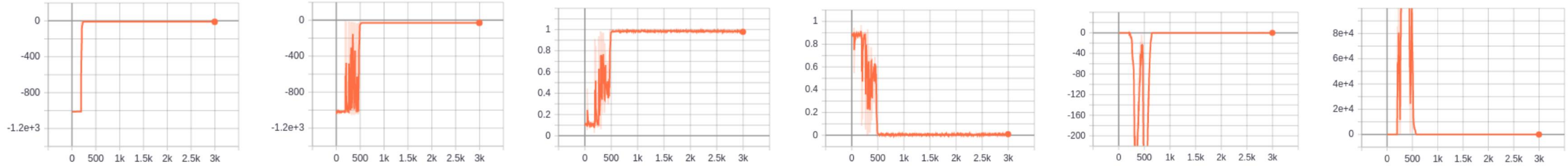
(g) VGG16 conv5 energy



(h) VGG16 conv5 latency

Evaluation: Mixed Precision Quantization Search Flow

- Intermediate values in search



(a) best reward (b) last reward (c) accuracy (d) accuracy diff (e) policy loss (f) value loss

- Search results

Quantization	Energy (μJ)	Latency (ms)	Accuracy
$Q_{baseline}$	850.99 (1.00x)	2.95 (1.00×)	97.27% (-0.00%)
Q_A	175.61 (**4.84×**)	0.76 (**3.89×**)	96.09% (**-1.18%**)
Q_B	229.82 (3.70×)	0.85 (3.48×)	96.29% (-0.98%)
Q_C	468.48 (1.82×)	1.69 (1.74×)	97.07% (-0.20%)

Q_{base} :(16,16,8),(16,16,8),(16,16,8),(16,16,8)

Q_A: (4, 16, 7), (4, 8, 8), (4, 8, 7), (4, 16, 8)

Q_B: (16, 8, 8), (4, 8, 8), (8, 8, 8), (4, 8, 8)

Q_C: (16,16,6), (16,8,8), (4,8,7), (4,16,7)

Conclusion

- We proposed a quantization scheme for ReRAM-based DNN inference accelerators that jointly targets weights, inputs, and partial sums, with a functional simulator that models the quantization scheme
- We proposed an automated mixed precision quantization flow powered by deep reinforcement learning that searches for the best quantization configuration for DNN inference on ReRAM-based accelerators
- Evaluation results show that our quantization scheme and search flow effectively improves the efficiency of ReRAM-based DNN accelerators
- Future work
  - Enable retraining to improve quantization and search results
  - Extend the flow to larger networks and other types of DNNs
  - Leverages faster and more accurate performance models to guide the search

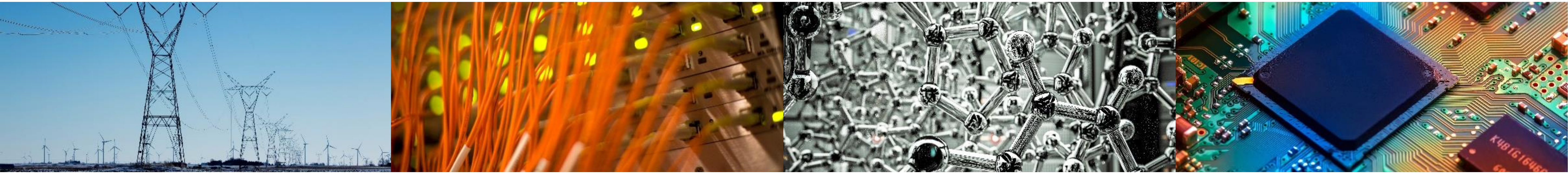
ASPDAC 2021: 26th Asia and South Pacific Design Automation Conference

Mixed Precision Quantization for ReRAM-based DNN Inference Accelerators

Sitao Huang[1], Aayush Ankit[2], Plinio Silveira[3], Rodrigo Antunes[3], Sai Rahul Chalamalasetti[4], Izzat El Hajj[5], Dong Eun Kim[2], Glaucimar Aguiar[3], Pedro Bruel[4,6], Sergey Serebryakov[4], Cong Xu[4], Can Li[4], Paolo Faraboschi[4], John Paul Strachan[4], Deming Chen[1], Kaushik Roy[2], Wen-mei Hwu[1], and Dejan Milojicic[4]

[1]University of Illinois at Urbana-Champaign, USA
[2]Purdue University, USA
[3]Hewlett Packard Enterprise, Brazil
[4]Hewlett Packard Enterprise, USA
[5]American University of Beirut, Lebanon
[6]University of São Paulo, Brazil

{shuang91,dchen,w-hwu}@illinois.edu, {aankit,kim2976,kaushik}@purdue.edu, izzat.elhajj@aub.edu.lb, {firstname.lastname}@hpe.com



ILLINOIS
Electrical & Computer Engineering
GRAINGER COLLEGE OF ENGINEERING

ASIA SOUTH PACIFIC DAC DESIGN AUTOMATION CONFERENCE *ASPDAC 2021: 26th Asia and South Pacific Design Automation Conference*

Thank You!

Mixed Precision Quantization for ReRAM-based DNN Inference Accelerators

Sitao Huang[1], Aayush Ankit[2], Plinio Silveira[3], Rodrigo Antunes[3], Sai Rahul Chalamalasetti[4], Izzat El Hajj[5], Dong Eun Kim[2], Glaucimar Aguiar[3], Pedro Bruel[4,6], Sergey Serebryakov[4], Cong Xu[4], Can Li[4], Paolo Faraboschi[4], John Paul Strachan[4], Deming Chen[1], Kaushik Roy[2], Wen-mei Hwu[1], and Dejan Milojicic[4]

[1]University of Illinois at Urbana-Champaign, USA [2]Purdue University, USA [3]Hewlett Packard Enterprise, Brazil
[4]Hewlett Packard Enterprise, USA [5]American University of Beirut, Lebanon [6]University of São Paulo, Brazil

{shuang91,dchen,w-hwu}@illinois.edu, {aankit,kim2976,kaushik}@purdue.edu, izzat.elhajj@aub.edu.lb, {firstname.lastname}@hpe.com