Analytical Guarantees on Numerical Precision of Deep Neural Networks

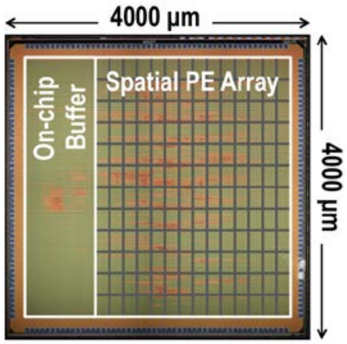
Charbel Sakr*, Yongjune Kim, Naresh Shanbhag*

Department of Electrical and Computer Engineering

University of Illinois at Urbana-Champaign

Machine Learning ASICs

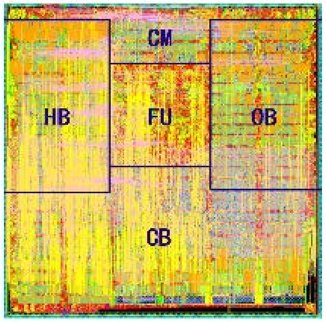
Eyeriss



[Sze’16, ISSCC]

AlexNet accelerator
16b fixed-point

PuDianNao



[Chen’15, ASPLOS]

ML accelerator
16b fixed-point

TPU

[Google’17, ISCA]

Tensorflow accelerator
8b fixed-point

How are they choosing these precisions?
Why is it working?
Can it be determined analytically?

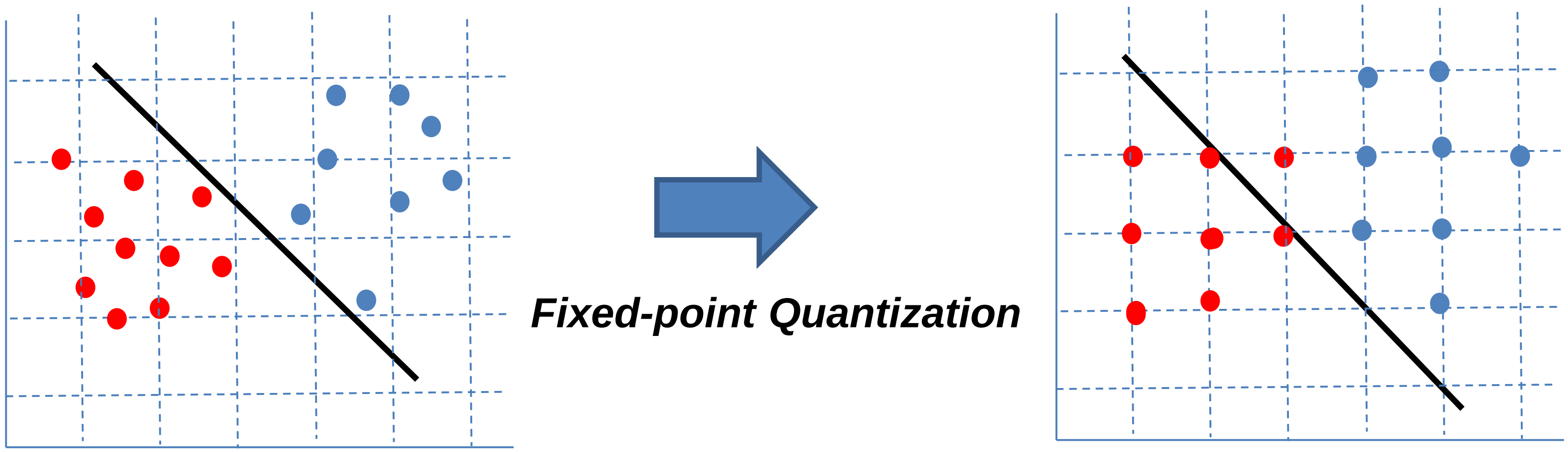
Current Approaches

- Stochastic Rounding during training [Gupta, ICML'15 – Hubara, NIPS'16]
  → Difficulty of training in a discrete space

- Trial-and-error approach [Sung, SiPS'14]
  → Exhaustive search is expensive

- SQNR based precision allocation [Lin, ICML'16]
  → Lack of precision/accuracy understanding

No theoretical guarantees on accuracy



Related Work from our Group

- Theoretical guarantees in LMS [Goel, TSP'98]
  → Precision requirements in linear estimators
    → Maintain tolerable output MSE
  → Precision requirements in LMS filters
    → Guarantee convergence in fixed-point

- Theoretical guarantees in hyperplane classifiers (SVM) [Sakr, ICASSP'17]
  → Precision requirements in classifier (forward mode)
    → Mimic geometry of floating-point classifiers
    → Guarantee worst case accuracy degradation
  → Precision requirements in trainer (SGD block)
    → Guarantee convergence in fixed-point

- Theoretical guarantees in deep learning [Sakr, ICML'17]
  → Precision requirements in classifier (feedforward neural network)
    → Guarantee worst case accuracy degradation
  → Precision requirements in trainer (SGD/Backprop block)
    → Unsolved today

Assumptions

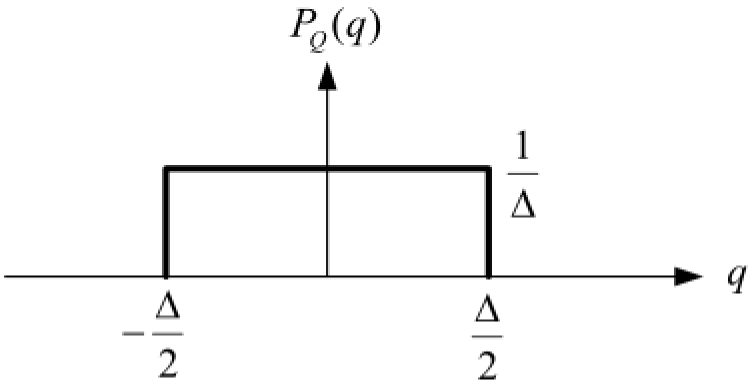
- Dynamic range before quantization is always equal to 2
  → Slight modification to activation function and weight update in backprop

- Quantization noise model
  → Additive noise:

$$x_q = x + q$$

$$q \sim U[-\tfrac{\Delta}{2}, \tfrac{\Delta}{2}]$$

$$\Delta = 2^{-(B-1)}$$



→ Uniformity + Independence
→ A useful result:

$$\sigma_q^2 = \frac{\Delta^2}{12} = \frac{2^{-2B}}{3}$$

Precision in Neural Networks

Classification

$$\hat{y} = \arg \max_{i=1,\ldots,M} z_i$$

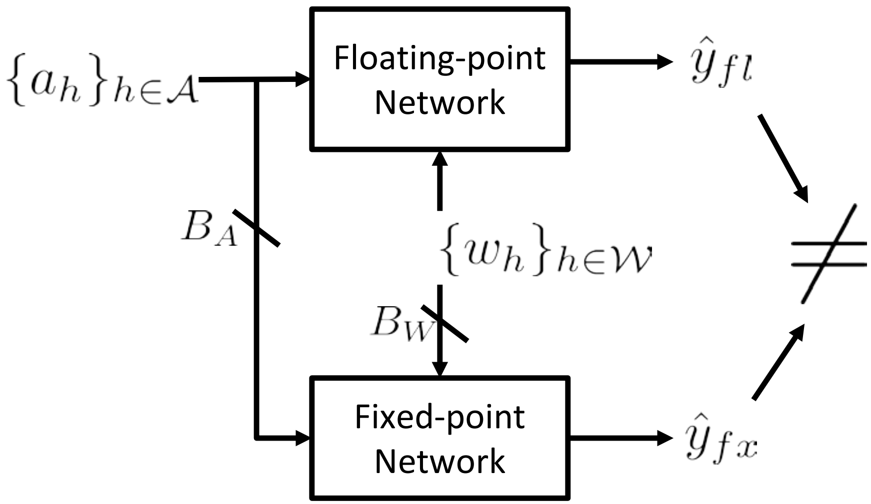
$$z_i = f\left(\{a_h\}_{h\in\mathcal{A}}, \{w_h\}_{h\in\mathcal{W}}\right)$$

Output Quantization

$$z_i + q_{z_i} = f\left(\{a_h + q_{a_h}\}_{h\in\mathcal{A}}, \{w_h + q_{w_h}\}_{h\in\mathcal{W}}\right)$$

$$q_{z_i} = \sum_{h\in\mathcal{A}} q_{a_h} \frac{\partial z_i}{\partial a_h} + \sum_{h\in\mathcal{W}} q_{w_h} \frac{\partial z_i}{\partial w_h}$$

Mismatch Probability



$$p_m = \Pr\{\hat{Y}_{fx} \neq \hat{Y}_{fl}\}$$

p_m: “Mismatch Probability”

Second Order Bound on p_m

$$p_m \leq \Delta_A^2 E_A + \Delta_W^2 E_W$$

- Input/Weight precision trade-off:
  → Bound is a sum of two terms
  → Optimal precision allocation by balancing the sum

$$B_A - B_W = round\left(\log_2 \sqrt{\frac{E_A}{E_W}}\right)$$

- Data dependence (compute once and reuse)
  → Derivatives obtained in last step of backprop
  → Only one forward-backward pass needed

$$E_A = \mathbb{E}\left[\sum_{\substack{i=1 \\ i \neq \hat{Y}_{fl}}}^{M} \frac{\sum_{h \in \mathcal{A}} \left|\frac{\partial (Z_i - Z_{\hat{Y}_{fl}})}{\partial A_h}\right|^2}{24|Z_i - Z_{\hat{Y}_{fl}}|^2}\right] \qquad E_W = \mathbb{E}\left[\sum_{\substack{i=1 \\ i \neq \hat{Y}_{fl}}}^{M} \frac{\sum_{h \in \mathcal{W}} \left|\frac{\partial (Z_i - Z_{\hat{Y}_{fl}})}{\partial w_h}\right|^2}{24|Z_i - Z_{\hat{Y}_{fl}}|^2}\right]$$

- Increasing in quantization noise variance
  → Decreases exponentially with precision – not surprising

$$\Delta_A = 2^{-(B_A - 1)}$$
$$\Delta_W = 2^{-(B_W - 1)}$$

Proof Sketch

- For one input, when do we have a mismatch?
  → If FL network predicts label "j"
  → But FX network predicts label "i" where $i \neq j$
    → This happens with some probability computed as follows:

$$\Pr\left(z_i + q_{z_i} > z_j + q_{z_j}\right)$$ (Output re-ordering due to quantization)

$$= \Pr\left(q_{z_i} - q_{z_j} > z_j - z_i\right) = \frac{1}{2}\Pr\left(\left|q_{z_i} - q_{z_j}\right| > \left|z_j - z_i\right|\right)$$ (Symmetry of quantization noise)

→ But we already know

$$q_{z_i} - q_{z_j} = \sum_{h \in \mathcal{A}} q_{a_h} \frac{\partial(z_i - z_j)}{\partial a_h} + \sum_{h \in \mathcal{W}} q_{w_h} \frac{\partial(z_i - z_j)}{\partial w_h}$$

→ Whose variance is

$$\frac{\Delta_A^2}{12} \sum_{h \in \mathcal{A}} \left|\frac{\partial(z_i - z_j)}{\partial a_h}\right|^2 + \frac{\Delta_W^2}{12} \sum_{h \in \mathcal{W}} \left|\frac{\partial(z_i - z_j)}{\partial w_h}\right|^2$$

→ Applying Chebyshev + LTP yields the result

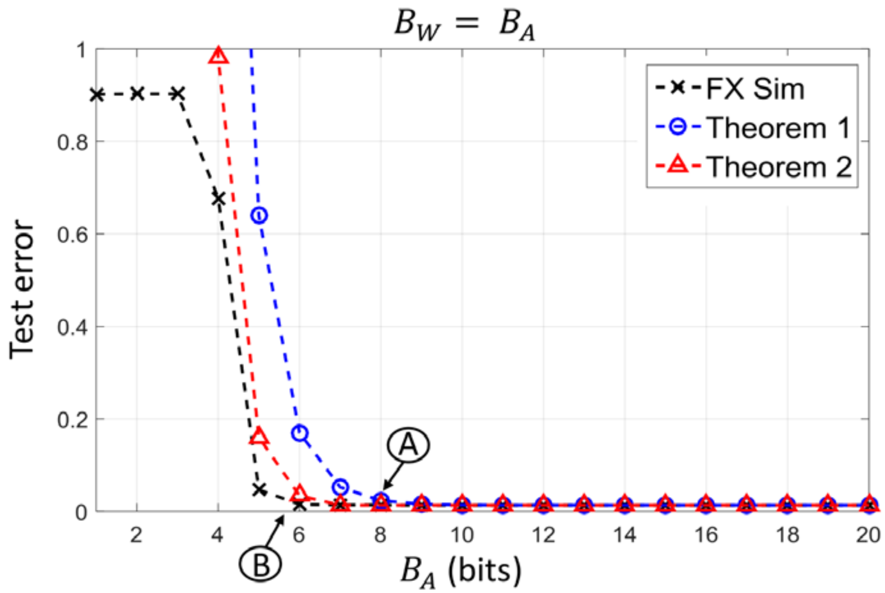
Tighter Bound on p_m

$$p_m \leq \mathbb{E}\left[\sum_{\substack{i=1 \\ i \neq \hat{Y}_{fl}}}^{M} e^{-S^{(i,\hat{Y}_{fl})}} P_1^{(i,\hat{Y}_{fl})} P_2^{(i,\hat{Y}_{fl})}\right]$$

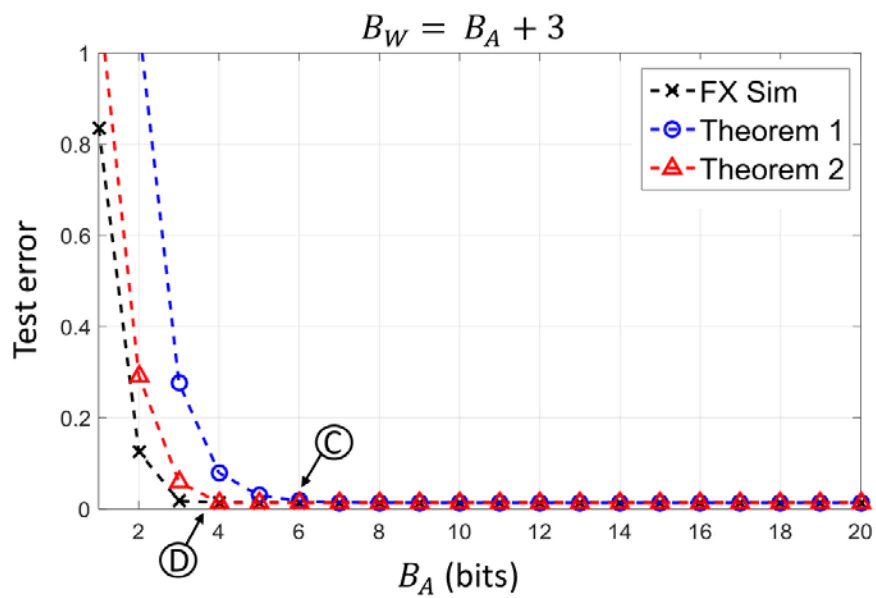
- Mismatch probability decreases *double exponentially* with precision
  → Theoretically stronger than Theorem 1
  → Unfortunately, less practical

M: Number of Classes
S: Signal to quantization noise ratio
$P1$ & $P2$: Correction factors

Simulations – MNIST



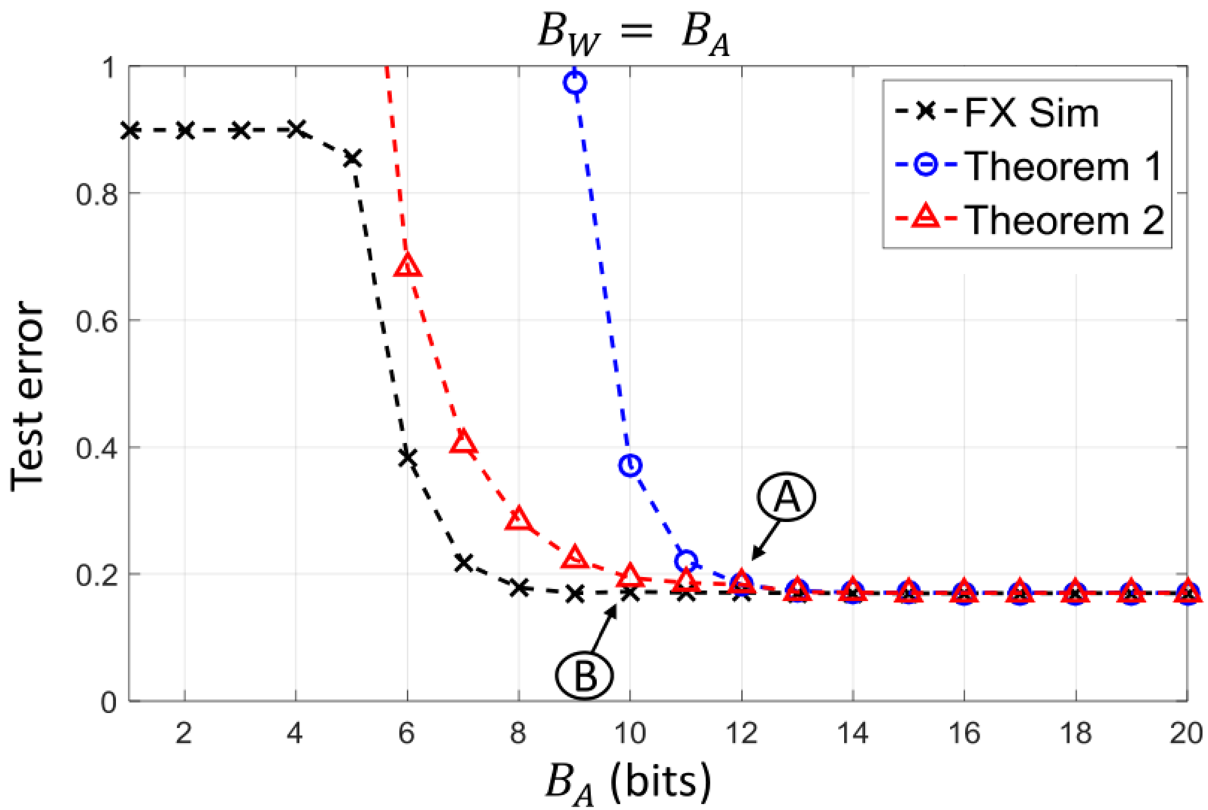
- Upper bounds are valid
- Activation precision reduced by 3 bits with no accuracy degradation



A: $B_W = B_A; p_m \leq 1\%$ (Theorem 1)
B: $B_W = B_A; p_m \leq 1\%$ (Theorem 2)
C: $B_W = B_A + 3; p_m \leq 1\%$ (Theorem 1)
D: $B_W = B_A + 3; p_m \leq 1\%$ (Theorem 2)

Neural network considered is a MLP with architecture: 784 – 512 – 512 – 512 – 10

Simulations – CIFAR-10



A: $B_W = B_A; p_m \leq 1\%$ (Theorem 1)
B: $B_W = B_A; p_m \leq 1\%$ (Theorem 2)

Neural network considered is a Convnet with architecture: 64C5 – 64C1 – 64C1 – MP2 – 64C5 – 64C1 – 64C1 – MP2 – 64C5 – 64FC – 64FC – 64FC – 10

Comparison with related works

- Simplified but meaningful model of complexity
  → **Computational cost**
    → Total number of FAs used assuming folded MACs with bit growth allowed
    → Number of MACs is equal to the number of dot products computed
    → Number of FAs per MAC:

$$DB_A B_W + (D-1)(B_A + B_W + \lceil \log_2(D) \rceil - 1)$$

  → **Representational cost**
    → Total number of bits needed to represent weights and activations
    → High level measure of area and communications cost (data movement)

$$|\mathcal{A}|\, B_A + |\mathcal{W}|\, B_W$$

- Other works considered
  → Stochastic quantization (SQ) [Gupta'15, ICML]
    → 784 – 1000 – 1000 – 10 (MNIST)
    → 64C5 – MP2 – 64C5 – MP2 – 64FC – 10 (CIFAR10)

  → BinaryNet (BN) [Hubara'16, NIPS]
    → 784 – 2048 – 2048 – 2048 – 10 (MNIST) & VGG (CIFAR10)

Results – MNIST

Precision Assignment	Test error (%)	Computational Cost (10^6 FAs)	Representational Cost (10^6 bits)
Floating-point	1.36	N/A	N/A
$(8, 8)$	1.41	82.9	7.5
$(6, 6)$	1.54	53.1	**5.63**
$(6, 9)$	**1.35**	72.7	8.43
$(4, 7)$	1.43	**44.7**	6.54
SQ $(16, 16)$ (Gupta et al., 2015)	1.4	533	28
BN $(1, 1)$ (Hubara et al., 2016b)	1.4	117	10

- No loss in accuracy

- ~2 × gains in computational and representational costs over BinaryNets!
  → Key finding: complexity scales *quadratically* with height (#neurons/layer)
    → BN (2048/512) is 4 times higher → 16 times more complex
    → Using up to 9 bits → still about 2 times less complex than BN

Results – CIFAR-10

Precision Assignment	Test error (%)	Computational Cost (10^6 FAs)	Representational Cost (10^6 bits)
Floating-point	17.02	N/A	N/A
$(12, 12)$	17.08	3626	5.09
$(10, 10)$	17.23	**2749**	**4.24**
SQ $(16, 16)$ (Gupta et al., 2015)	25.4	4203	4.54
BN $(1, 1)$ (Hubara et al., 2016b)	**10.15**	3608	6.48

- Clear win in accuracy and complexity over SQ
- Less complexity than BN but worse accuracy

Conclusion & Future Work

- Theoretical bounds on accuracy degradation in fixed-point
  → Theorem 1
    → Based on second order statistics
    → Introduces interesting trade-off between activation and weight precisions
  → Theorem 2
    → Tighter, based on Chernoff bound
    → Establishes double exponential quantization tolerance

- Complexity vs. accuracy comparison
  → Computational and representational costs
    → Meaningful metrics of complexity measure
  → Quadratic scaling of complexity with network height is key
    → Big binary network not better than small low precision w/ same accuracy

- Future work
  → Theoretical guarantees in different setups
    → Layer wise granular precision analysis
  → Precision minimization and model reduction
    → Precision guided pruning

Thank you!

Acknowledgment:
This work was supported in part by Systems on Nanoscale Information fabriCs (SONIC), one of the six SRC STARnet Centers, sponsored by MARCO and DARPA.