

香港中文大學 The Chinese University of Hong Kong

CENG5030 Part 2-4: CNN Inaccurate Speedup-2 —- Quantization

Bei Yu

(Latest update: March 25, 2019)

Spring 2019

つへへ

These slides contain/adapt materials developed by

- \triangleright Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. ICML*, pp. 1737–1746
- \triangleright Ritchie Zhao et al. (2017). "Accelerating binarized convolutional neural networks with software-programmable FPGAs". In: *Proc. FPGA*, pp. 15–24
- \triangleright Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542

State of the art recognition methods

- Very Expensive
	- Memory'
	- Computation
	- Power'

[Fixed-Point Representation](#page-9-0)

[Binary/Ternary Network](#page-24-0)

[Reading List](#page-59-0)

Overview

[Fixed-Point Representation](#page-9-0)

[Binary/Ternary Network](#page-24-0)

[Reading List](#page-59-0)

Fixed-Point v.s. Floating-Point

Fixed-Point v.s. Floating-Point

Fixed-Point v.s. Floating-Point

Fixed-Point Arithmetic

Number representation

7

1

H

 QQQ

¹Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-12-0)*, p[p.](#page-14-0) [17](#page-12-0)[3](#page-13-0)[7](#page-23-0)[–1](#page-24-0)[7](#page-8-0)[4](#page-9-0)[6.](#page-23-0)

Fixed-Point Arithmetic

1

Fixed-Point Arithmetic: Rounding Modes

Round-to-nearest

9

1

¹Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-14-0)*, p[p.](#page-16-0) [17](#page-12-0)[3](#page-13-0)[7](#page-23-0)[–1](#page-24-0)[7](#page-8-0)[4](#page-9-0)[6.](#page-23-0) E 299

Fixed-Point Arithmetic: Rounding Modes

Round-to-nearest Stochastic rounding $|x|$ $|x|+\epsilon$ $|x|+2\epsilon$ $|x|$ $-\epsilon$ $Round(x, \langle IL, FL \rangle) =$ $\begin{cases} \lfloor x \rfloor & \text{w.p. } 1 - \frac{x - \lfloor x \rfloor}{\epsilon} \\ \lvert x \rvert + \epsilon & \text{w.p. } \frac{x - \lfloor x \rfloor}{\epsilon} \end{cases}$ Non-zero probability of rounding to

> Unbiased rounding scheme: expected rounding error is zero

either $|x|$ or $|x| + \epsilon$

10

1

¹Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-15-0)*, p[p.](#page-17-0) [17](#page-12-0)[3](#page-13-0)[7](#page-23-0)[–1](#page-24-0)[7](#page-8-0)[4](#page-9-0)[6.](#page-23-0)

MNIST: *Fully-connected DNNs*

11

1Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-16-0)*, p[p.](#page-18-0) [17](#page-12-0)[3](#page-13-0)[7](#page-8-0)+17[4](#page-9-0)[6.](#page-23-0)

1

ali
Na

MNIST: *Fully-connected DNNs*

- For small fractional lengths $(FL < 12)$, a large majority of weight updates are rounded to zero when using the round-to-nearest scheme.
	- **EXECONVERGERICE Slows down**
- For $FL < 12$, there is a noticeable degradation in the classification accuracy

1

H

つへへ

¹Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-17-0)*, p[p.](#page-19-0) [17](#page-12-0)[3](#page-13-0)[7](#page-23-0)[–1](#page-24-0)[7](#page-8-0)[4](#page-9-0)[6.](#page-23-0)

MNIST: *Fully-connected DNNs*

- Stochastic rounding preserves gradient information (statistically)
	- No degradation in convergence properties
- **Fig.** Test error nearly equal to that obtained using 32-bit floats

13

1

ari
Ali

 QQ

¹Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-18-0)*, p[p.](#page-20-0) [17](#page-12-0)[3](#page-13-0)[7](#page-23-0)[–1](#page-24-0)[7](#page-8-0)[4](#page-9-0)[6.](#page-23-0)

FPGA prototyping: GEMM with stochastic rounding

Top-level controller and memory hierarchy designed to maximize data reuse

Wavefront systolic array for computing matrix product *AB.* Arrows indicate dataflow

21

September

¹ 290

¹Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-19-0)*, p[p.](#page-21-0) [17](#page-12-0)[3](#page-13-0)[7](#page-23-0)[–1](#page-24-0)[7](#page-8-0)[4](#page-9-0)[6.](#page-23-0)

Maximizing data reuse

Inner Loop:

Cycle through columns of Matrix B (*M/n* iterations)

Outer Loop:

Cycle through rows of Matrix A (*K/p.n* iterations)

Re-use factor for Matrix A: *M* times Re-use factor for Matrix B: *p.n* times

n : dimension of the systolic array *p* : parameter chosen based on available BRAM resources

22

ali
Na

¹Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-20-0)*, p[p.](#page-22-0) [17](#page-12-0)[3](#page-13-0)[7](#page-23-0)[–1](#page-24-0)[7](#page-8-0)[4](#page-9-0)[6.](#page-23-0)

Stochastic rounding

23

¹Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-21-0)*, p[p.](#page-23-0) [17](#page-12-0)[3](#page-13-0)[7](#page-23-0)[–1](#page-24-0)[7](#page-8-0)[4](#page-9-0)[6.](#page-23-0) 290 E

Stochastic rounding

1

 299

¹Suyog Gupta et al. (2015). "Deep learning with limited numerical precision". In: *Proc. IC[ML](#page-22-0)*, p[p.](#page-24-0) [17](#page-12-0)[3](#page-13-0)[7](#page-23-0)[–1](#page-24-0)[7](#page-8-0)[4](#page-9-0)[6.](#page-23-0)

Overview

[Fixed-Point Representation](#page-9-0)

[Binary/Ternary Network](#page-24-0)

[Reading List](#page-59-0)

Binarized Neural Networks (BNN)

CNN

Key Differences

- 1. Inputs are binarized (−1 or +1)
- 2. Weights are binarized (-1 or +1)
- 3. Results are binarized after **batch normalization**

BNN

BNN CIFAR-10 Architecture [2]

- ► 6 conv layers, 3 dense layers, 3 max pooling layers
- \blacktriangleright All conv filters are 3x3
- \blacktriangleright First conv laver takes in floating-point input
- ▶ **13.4 Mbits total model size** (after hardware optimizations)

[7](#page-24-0) [2] M. Courbariaux et al. **Binarized Neural Networks: Training Deep Neural Networks with Weights and Activatio[ns C](#page-25-0)o[nst](#page-27-0)[rai](#page-24-0)[ne](#page-25-0)[d](#page-28-0) [to](#page-29-0) [+1](#page-23-0) or -1**. *arXiv:1602.02830*, Feb 2016.

Advantages of BNN

1. Floating point ops replaced with binary logic ops

- Encode {+1,−1} as {0,1} à multiplies become XORs
- Conv/dense layers do dot products \rightarrow XOR and popcount
- Operations can map to LUT fabric as opposed to DSPs

2. Binarized weights may reduce total model size

– Fewer bits per weight may be offset by having more weights

BNN vs CNN Parameter Efficiency

* Assuming each float param can be quantized to 8-bit fixed-point

Comparison:

- Conservative assumption: ResNet can use 8-bit weights
- BNN is based on VGG (less advanced architecture)
- BNN seems to hold promise!

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. K ロ) K (@) K (할) K (할) () 할 . 298

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542.

2

COMPANY

-
²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary 4 ロ ▶ 4 @ ▶ 4 할 ▶ 4 할 ▶ 2 할 | + 9 Q Q + 2

2 Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In:

2 Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional $8/9$ are binary W , where the elements of Binary-weights two variations of binary CNN: ²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: in figure 2 (second row) by X1 and X2. Due to overlaps by X1 and X2. Due to overlaps betw[ee](#page-24-0)[n](#page-0-0) [s](#page-23-0)[u](#page-24-0)[b](#page-58-0)[te](#page-59-0)nsions, comput-Estimating binary weights: Without loss of generality we assume W, [B](#page-30-0) a[re](#page-32-0) [v](#page-28-0)[e](#page-29-0)[ct](#page-58-0)[ors](#page-23-0) *Proc. ECCV*, pp. 525–542. in Rn, where n \mathbb{R} where n \mathbb{R} we solve the W \mathbb{R} , we solve the W \mathbb{R}

 $t₁$, μ is a constraint a con

 $\overline{\mathbf{c}}$

Quantization Error

and XNOR-Networks, where elements of binary tensors. α , where it is a contribution for α , α , β

following optimization:

 $W^B = \text{sign}(W)$ $W^B = \text{sign}(W)$ $\mathbf{v} = \text{sign}(\mathbf{v})$

2

 173.2 ^J(B, ↵)= ^k^W ↵Bk² ²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. K ロ ▶ K @ ▶ K ミ X K 동 X [E | YO Q Q

Optimal Scaling Factor can be solved by a solve Ontimal Scaling Factor Upunial Ocaling ractor indicates ^a convolution without any multiplication. Since the weight values where, are binary, we can implement the convolution with a convolution with a convolution with a convolution with a c
Subtraction with a convolution with a convolution with a convolution with a convolution with a convolutions. T

Training Binary-Weights-Networkssteps; forward pass, backward pass and param[ete](#page-32-0)r[s](#page-34-0) [up](#page-28-0)[d](#page-29-0)[a](#page-58-0)[t](#page-59-0)[e.](#page-23-0)[T](#page-58-0)[o](#page-59-0) [t](#page-23-0)[ra](#page-24-0)[i](#page-58-0)[n](#page-59-0) [a](#page-0-0) [CN](#page-60-0)N with bi-2
2 Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: nary weights (in convolutional layers), we only binarize the weights during the forward t **R 2 W respectively. here** in the *width* here is the *width* here is the *width* $\frac{1}{2}$ and $\frac{1}{2}$ a *Proc. ECCV*, pp. 525–542.

 $\overline{\mathbf{c}}$

 $\mathcal{L}_{\mathcal{B}}$

 $\mu_{\rm out}$ to train a CNN with binary filtors 2 $\frac{1}{2}$ and $\frac{1}{2}$ and $\frac{1}{2}$ and weight filter which $\frac{1}{2}$ matrix. How to train a CNN with binary filters? $\frac{1}{2}$ and we first filter which $\frac{1}{2}$ matrix.

 $\overline{\mathbb{F}_q}$ 2 Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classi[f](#page-28-0)icationusing binary c[o](#page-59-0)nvolutional neu[r](#page-23-0)al networks". In: ﷺ
2Mohamma[d](#page-24-0) Rast[e](#page-59-0)gar[i](#page-24-0) et al. (2016). "XNOR-NET: Image[n](#page-58-0)et classification usin[g](#page-59-0) binary convolutional *Proc. ECCV*, pp. 525–542.

scaling factors and specifying the right order for layers in a block of CNN with binary input.

Training Binary Weight Networks

Naive Solution:

1. Train a network with real value parameters 2. Binarize the weight filters

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. $\mathbf{A} \cap \mathbf{B} \rightarrow \mathbf{A} \oplus \mathbf{B} \rightarrow \mathbf{A} \oplus \mathbf{B} \rightarrow \mathbf{B} \oplus \mathbf{B} \oplus \mathbf{B}$ 2

 QQ

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: **The Contract of the Contract o** *Proc. ECCV*, pp. 525–542. イロトメ 倒 トメ 君 トメ 君 トー 君 298

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. K ロ ▶ K @ ▶ K 할 > K 할 > → 할 → 9 Q @

2

PARTIES

 2990

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. メロトメ 御 トメ 差 トメ 差 トー 差し

Binary Weight Network

Train for binary weights:

- 1. Randomly initialize W
- 2. For iter $= 1$ to N
- 3. Load a random input image X

4.
$$
W^B = sign(W)
$$

$$
5. \qquad \alpha = \frac{\|W\|_{\ell 1}}{n}
$$

- 6. Forward pass with α , $\mathbf{W}^{\mathbf{B}}$
- 7. Compute loss function C

8.
$$
\frac{\partial \mathbf{C}}{\partial \mathbf{W}} =
$$
 Backward pass with α , $\mathbf{W}^{\mathbf{B}}$

9. Update W
$$
(W = W - \frac{\partial C}{\partial W})
$$

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. KONKARD KENKEN E

Binary Weight Network W

Train for binary weights:

- 1. Randomly initialize W
- 2. For $iter = 1$ to N
- 3. Load a random input image X

4.
$$
W^B = sign(W)
$$

$$
5. \qquad \alpha = \frac{\|W\|_{\ell 1}}{n}
$$

- 6. Forward pass with α , $\mathbf{W}^{\mathbf{B}}$
- 7. Compute loss function C

8.
$$
\frac{\partial \mathbf{C}}{\partial \mathbf{W}} =
$$
 Backward pass with α , $\mathbf{W}^{\mathbf{B}}$

9. Update W
$$
(W = W - \frac{\partial C}{\partial W})
$$

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ - B

2

 QQ

Binary Weight Network R R W

Train for binary weights:

- 1. Randomly initialize W
- 2. For $iter = 1$ to N
- 3. Load a random input image X

4.
$$
W^B = sign(W)
$$

$$
5. \qquad \alpha = \frac{\|W\|_{\ell 1}}{n}
$$

- 6. Forward pass with $\alpha, \mathbf{W}^{\mathbf{B}}$
- 7. Compute loss function C

8.
$$
\frac{\partial \mathbf{C}}{\partial \mathbf{W}} =
$$
 Backward pass with α , $\mathbf{W}^{\mathbf{B}}$

9. Update W
$$
(W = W - \frac{\partial C}{\partial W})
$$

2

 QQ

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. K ロ ▶ K 優 ▶ K 결 ▶ K 결 ▶ │ 결

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. イロト 不優 トス 差 トス 差 ト - B QQ

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$

2

 QQ

2

 QQ

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. KONKARD KENKEN E

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: **The Contract of the Contract o** *Proc. ECCV*, pp. 525–542. イロト イ団 トイモト イモト 一毛 2990

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542.

2

COMPANY

Binary Input and Binary Weight (XNOR- $\bm{\mathsf{Net}})$ a sub-tensor centered at the location is (across width and height). This procedure is procedure is procedure i nary Input and Binary Weight (XNOR- $\,$ Binary imput and Binary Weight (ANONbetween input I and weight filter W mainly using binary operations: indicates a convolution with $\mathcal{C}(\mathcal{C})$ and we include $\mathcal{C}(\mathcal{C})$ where, D and at the location in the location is a factor of the location in the location in the location is procedure in the location in the location in the location is a factor of the location in the location in the loca Binary-Input and Binary-V and \mathbf{N} (denoted by \mathbf{N}), we can approximate the convolution of convolution \mathbf{N} where Net out from the optimization and the optimal solutions can be achieved from equations can be achieved from equation 2 as α $\frac{1}{2}$ 1 International Control Control
1 International Control Contro $3...$ $6...$

^R ² ^W respectively. *hei[g](#page-23-0)ht* and *width* ^h ⇥w⇥^c . We propose in ^h , ^h in ^w ^w , where 2
2 Mohammad Rastegar[i](#page-29-0) et al. (2016). "XNOR-NET: Imagenet classifica[ti](#page-0-0)on using [b](#page-58-0)inary convolutional neural networks". In: possible sub-tensors in I with same size as W. Two of these sub-tensors are illustrated $\frac{8}{9}$ *Proc. ECCV*, pp. 525–542.

2

 $\overline{\mathbf{c}}$

Binary Input and Binary Weight (XNOR- $\bm{\mathsf{Net}})$ a sub-tensor centered at the location is $\frac{1}{2}$ (across width and height). This procedure is procedure is procedure in Binary imput and Binary Weight (ANON- $\begin{bmatrix} 1 \end{bmatrix}$ indicates a convolution with $\mathcal{C}(\mathcal{C})$ and we include $\mathcal{C}(\mathcal{C})$ where, D and at the location in the location is a factor of the location in the location in the location is procedure in the location in the location in the location is a factor of the location in the location in the loca Binary Input and Binary Weight (XNORand \mathbf{N} (denoted by \mathbf{N}), we can approximate the convolution of convolution \mathbf{N} N et) is an n-dimensional vector where N out from the optimization and the optimal solutions can be achieved from equation 2 as $\frac{1}{2}$ 1 International Control Control
1 International Control Contro $\overline{}$ is a lk B is a set of positive real scalars, such that A a set of α a set of α a set of binary tensors and α and

^R ² ^W respectively. *hei[g](#page-23-0)ht* and *width* ^h ⇥w⇥^c . We propose in ^h , ^h in ^w ^w , where 2
2Mohammad Rastegar[i](#page-29-0) et al. (2016). "XNOR-NET: Imagenet classifica[ti](#page-0-0)on using [b](#page-58-0)inary convolutional neural networks". In: ing f and f possible sub-tensors leads to a large number of r redundant computations. are binary tensors. W and I , where elements of both \mathbb{R} , where elements of both XNOR-Networks tensors and I , where \mathbb{R} *Proc. ECCV*, pp. 525–542.

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. イロトメ 御 トメ 君 トメ 君 トー 君 2990

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. K ロ ▶ K @ ▶ K 경 ▶ K 경 ▶ ○ 경

2

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: **The Contract of the Contract o** *Proc. ECCV*, pp. 525–542. イロト イ団 トイモト イモト 一毛 2990

Network Structure in XNOR-Networks

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. イロト イ団ト イモト イモト 一重 299

Network Structure in XNOR-Networks

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. K ロ ▶ K 優 ▶ K 결 ▶ K 결 ▶ │ 결 QQ

Network Structure in XNOR-Networks

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. K ロ ▶ K @ ▶ K 경 ▶ K 경 ▶ ○ 경 QQ

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. イロト イ部 トイモト イモト 一毛 2990

 2990

-
²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. イロト イ部 トイモト イモト 一毛

²Mohammad Rastegari et al. (2016). "XNOR-NET: Imagenet classification using binary convolutional neural networks". In: *Proc. ECCV*, pp. 525–542. イロト イ団 トイモト イモト 一毛 2990

2

PARTIES

Overview

[Fixed-Point Representation](#page-9-0)

[Binary/Ternary Network](#page-24-0)

[Reading List](#page-59-0)

Further Reading List

Fixed-Point Representation:

- ▶ Darryl Lin, Sachin Talathi, and Sreekanth Annapureddy (2016). "Fixed point quantization of deep convolutional networks". In: *Proc. ICML*, pp. 2849–2858
- ▶ Soroosh Khoram and Jing Li (2018). "Adaptive quantization of neural networks". In: *Proc. ICLR*

Binary/Ternary Network:

 \blacktriangleright Hyeonuk Kim et al. (2017). "A Kernel Decomposition Architecture for Binary-weight Convolutional Neural Networks". In: *Proc. DAC*, 60:1–60:6

■ Chenzhuo Zhu et al. (2017). "Trained ternary quantization". In: *Proc. ICLR*

9 / 9