ECE 598NSG/498NSU
Deep Learning in Hardware
Fall 2020

Training DNNs in Fixed-Point

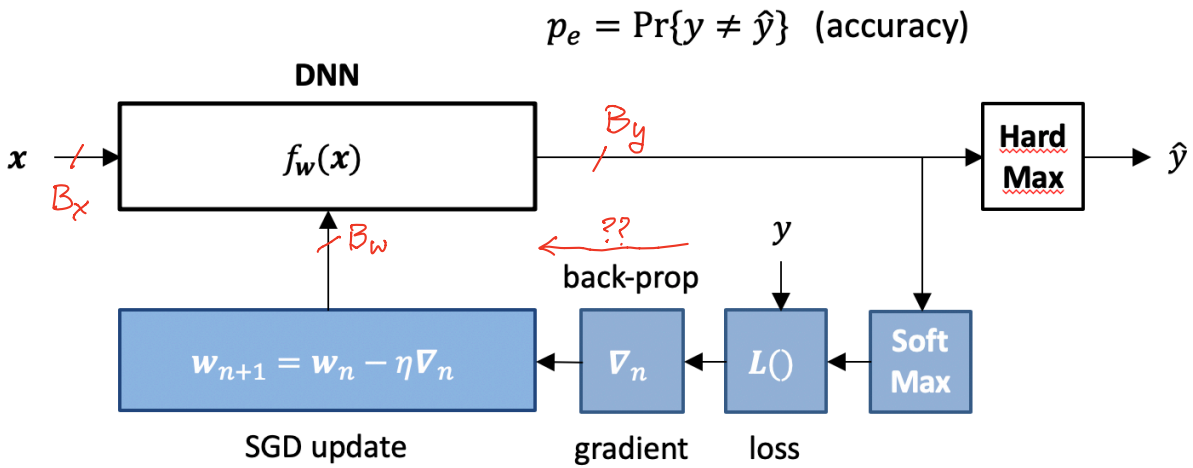
Naresh Shanbhag
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

http://shanbhag.ece.uiuc.edu

Today

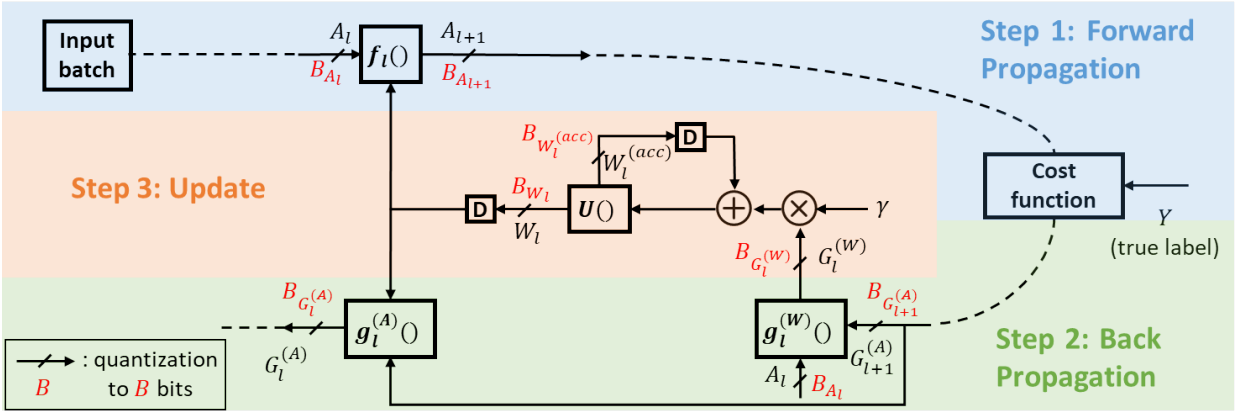
- Introduction
- Fixed-point LMS for a linear regressor
- Fixed-point back-prop for DNNs

Fixed-Point DNN Training



- Key question: How to assign precisions to minimize impact on p_e (error probability)?
- Previously: precision assignment for the inference (forward) path
- this lecture: precision assignment for the feedback path

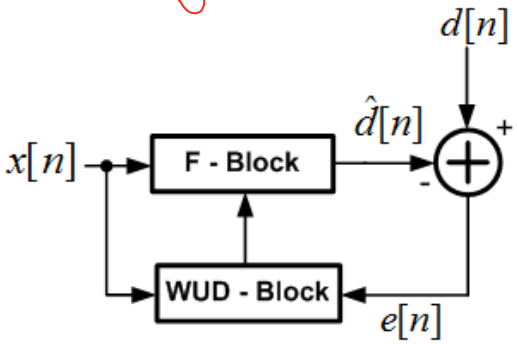
Challenges in Fixed-point Training



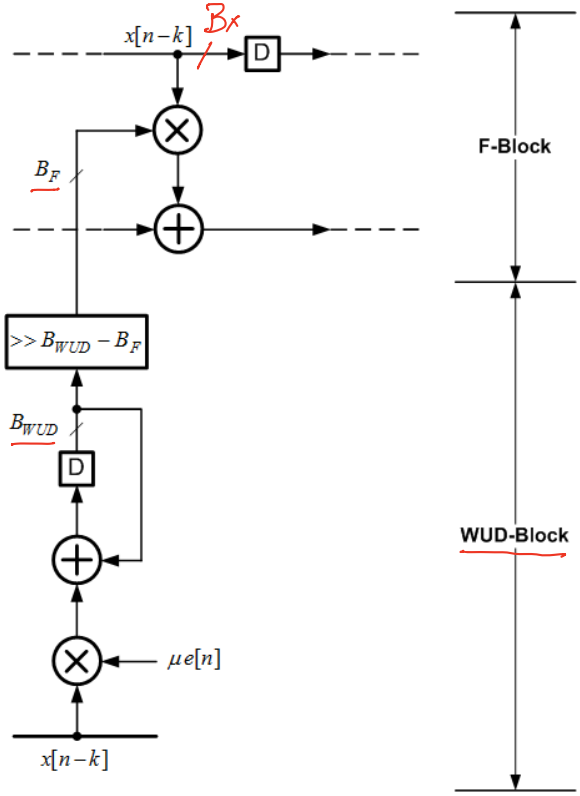
- multiple forward quantization noise sources
- unknown gradient dynamic range
- instability due to quantization noise bias in updates
- back-propagation of quantization noise in activation gradients
- risk of premature stoppage of convergence

Fixed-point LMS

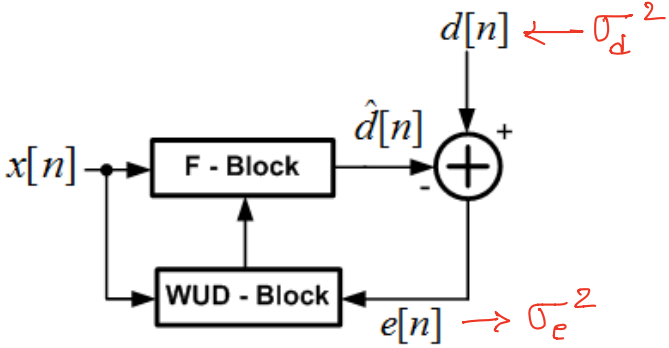
floating-pt. network

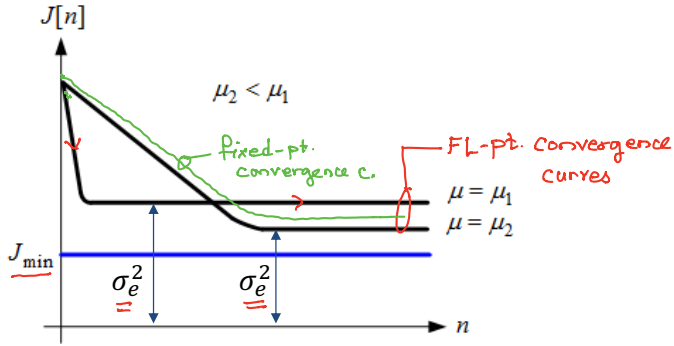


Fx



- input precision - B_X
- weight precision in **F**-block: B_F
- weight precision in the **WUD**-block: B_{WUD}
- also internal **F**-block precisions



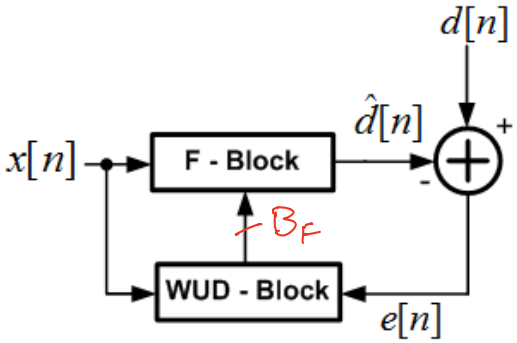


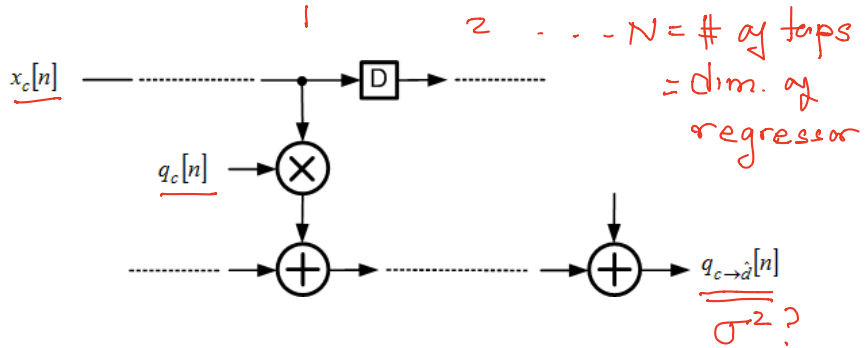
- SNR_{fl} : SNR of a floating point algorithm (given)

$$E\left[e^2[n]\right] = \sigma_e^2 = \frac{\sigma_d^2}{SNR_{fl}}$$

- SNR_{fx} : SNR of fixed point algorithm; want SNR_{fx} to be close to SNR_{fl}
- assume further that $x[n]$ is uncorrelated

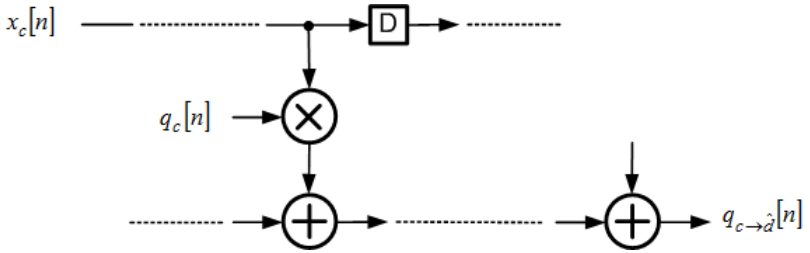
Finite-precision (forward) F-Block

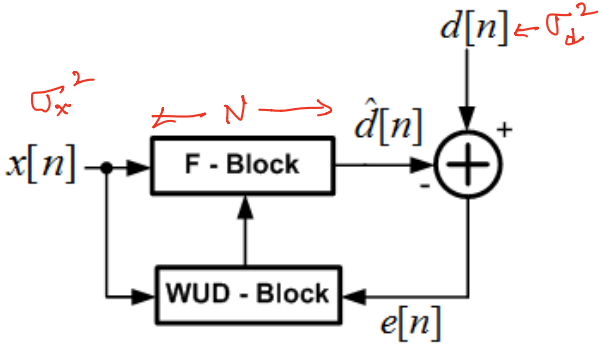




- focus on coefficient precision B_F only (ignore input quantization) $\frac{\Delta^2}{12}$
- coefficient as an RV: $c_q = c + q_c \rightarrow \sigma^2_{q_c} = \frac{2^{-2B_F}}{3}$ $(q_c \sim U[-2^{-B_F}, 2^{-B_F}])$ $-\frac{\Delta}{2}$ $\frac{\Delta}{2}$
- noise at output from coefficient quantization only:

$$\sigma^2_{q_c \rightarrow \hat{d}} = \sigma_x^2 N \left(\frac{2^{-2B_F}}{3}\right) \text{ (trade-off between } B_F, N \text{ and } \sigma_X^2)$$





- ensure $\sigma^2_{q_c\to\hat{d}} \ll \sigma_e^2$. Hence, for some $\alpha \ll 1$,

$$\sigma^2_{q_c\to\hat{d}} \leq \alpha\sigma_e^2$$

SNR_{fx} is close to SNR_{fl}

$\sigma^2_{q_c\to\hat{d}}$ → quantization noise seen @ the output.

$\alpha\sigma_e^2$ → MSE of FL

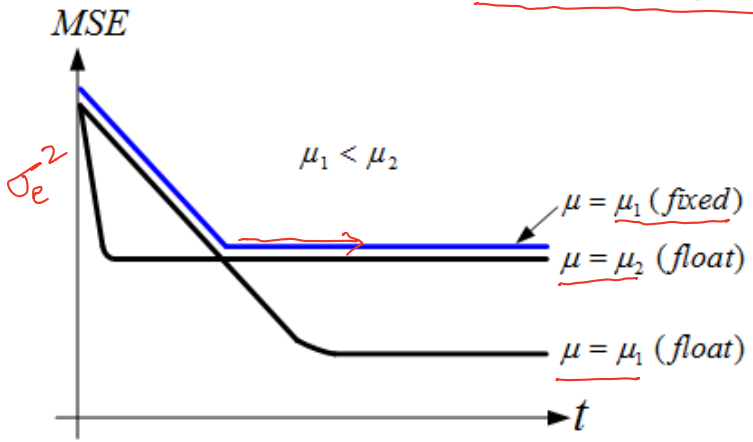
- thus, **F**-block precision B_F is given by

$$N\sigma_x^2\frac{2^{-2B_F}}{3} \leq \alpha\sigma_d^2 \quad (\checkmark SNR_{fx})$$

$$B_F \geq \frac{1}{2}\log_2\left(\frac{N\sigma_x^2}{3\alpha\sigma_d^2}\right) + \frac{SNR_{fl}(dB)}{6}$$

- note: B_F increases with N, σ_X^2 and SNR_{fl} and as α, σ_d^2 reduces

WUD-block Quantization (Stopping Criterion)



$$e[n] = d[n] - \boldsymbol{W}^T[n]\boldsymbol{X}[n]$$

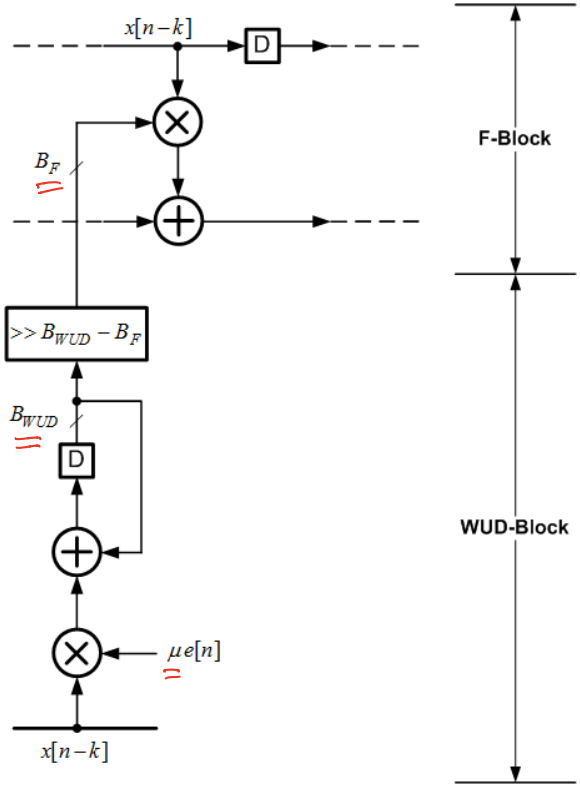
$$\boldsymbol{W}[n+1] = \boldsymbol{W}[n] + \mu e[n]\boldsymbol{X}[n]$$

$=0$

$\mathrm{Var}(\quad) < \frac{1}{2}$ LSB

accuracy!

- **WUD**-block requires more precision than the **F**-block
  - update term becomes small as μ and $e[n]$ reduce
- finite-precision filter stops adapting even if floating-point filter continues (stopping criterion)



- stopping criterion:

$$\mu^2 E\{e^2[n]\}\sigma_x^2 \geq 2^{-2B_{WUD}}$$

- **WUD**-block precision:

$$B_{WUD} \geq \frac{1}{2}\log_2\left(\frac{1}{\mu^2\sigma_d^2\sigma_x^2}\right) + \frac{SNR_{fl}(dB)}{6}$$

- precision increases as μ reduces

Example

- consider the design of an equalizer for VDSL. The equalizer output needs to achieve an SNR of 24 dB in order to meet a desired BER. The transmitted data are ±1s; equalizer length $N = 8$

$SNR_{fl} = 24$,

- determine precision of equalizer coefficients in order for the fixed-point algorithm to achieve output SNR within 0.1dB of the floating point algorithm

$SNR_{fx} \geq 24 - 0.1 = 23.9$ dB

B_F & B_{WID} ?

Calculate B_F

$$\sigma_d^2 = \frac{(1)^2 + (-1)^2}{2} = 1$$

$$SNR_{fl}(dB) = 24$$

$$SNR_{fl} = \frac{\sigma_d^2}{\sigma_e^2}; \qquad SNR_{fx} = \frac{\sigma_d^2}{\sigma_e^2 + \sigma_{q_c \to d}^2} = \frac{\sigma_d^2}{\sigma_e^2} \frac{1}{(1+\alpha)};$$

$$SNR_{fx}(dB) = SNR_{fl}(dB) - 10\log_{10}(1+\alpha) = SNR_{fl}(dB) - 0.1$$

$$\alpha = 0.023$$

- Substitute: $N = 8; \sigma_X^2 = 1; \alpha = 0.023; \sigma_d^2 = 1; SNR_{fl}(dB) = 24;$ into

$$B_F \geq \frac{1}{2}\log_2\left(\frac{N\sigma_x^2}{3\alpha\sigma_d^2}\right) + \frac{SNR_{fl}(dB)}{6}$$

$$B_F \geq 7.42 = 8b$$

Calculate B_{WUD}

$$B_{WUD} \geq \frac{1}{2}\log_2\left(\frac{1}{\mu^2\sigma_d^2\sigma_x^2}\right) + \frac{SNR_{fl}(dB)}{6}$$

$$0 \leq \mu \leq \frac{1}{N\sigma_x^2} = 2^{-3}$$

$$gear-shift: \quad \mu = 2^{-4} \rightarrow 2^{-5} \rightarrow 2^{-6} \rightarrow 2^{-7} \rightarrow 2^{-8}$$

$$B_{WUD} \geq \frac{1}{2}\log_2\left(\frac{1}{2^{-16}}\right) + \frac{24}{6} = 12b$$

- Note: $B_{WUD} > B_F$ by roughly 2X

Summary

- need to minimize the precision requirements of machine learning kernels → improves latency and energy efficiency
- want to avoid trial-and-error approach
- precision of feedforward path dictated by output accuracy requirements
- precision of feedback (training) dictated by stopping criterion
- similar considerations expected in other SGD-based learning algorithms

Fixed-point DNN Training

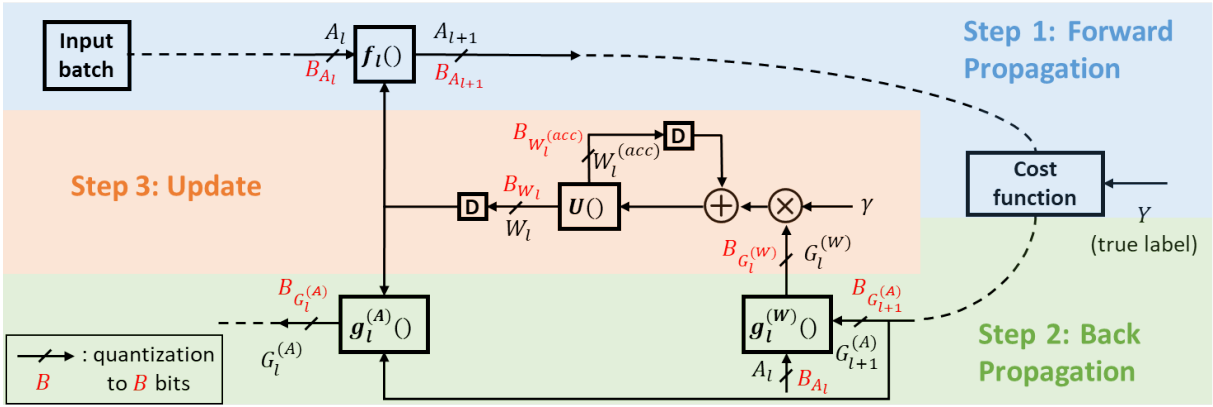
PER-TENSOR FIXED-POINT QUANTIZATION OF THE BACK-PROPAGATION ALGORITHM

Charbel Sakr and Naresh Shanbhag

University of Illinois at Urbana-Champaign

2019 International Conference on Learning Representations (ICLR)

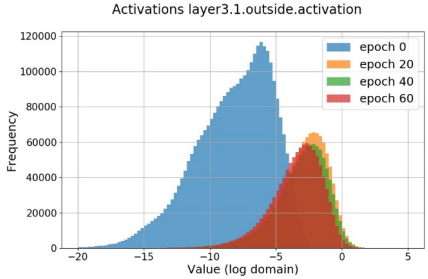
Problem Setup & Challenges

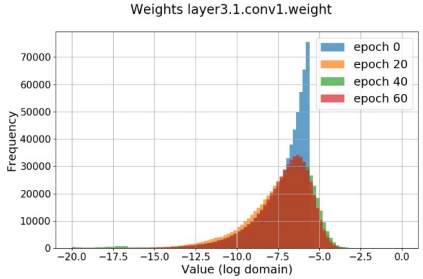


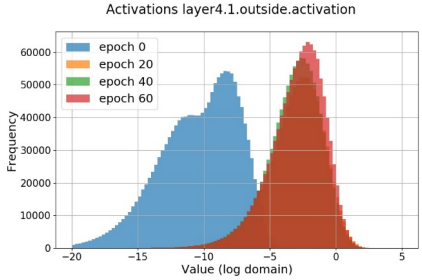
- multiple forward quantization noise sources
- unknown gradient dynamic range
- instability due to quantization noise bias in updates
- back-propagation of quantization noise in activation gradients
- risk of premature stoppage of convergence

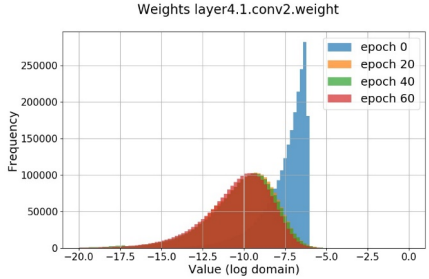
Unknown Spatio-temporally Varying Weight/Act. Distributions

- layer and time-dependent distributions
- precision assignment should be done so that:
  1. disparity in distributions is accounted for
  2. accuracy of model upon convergence is optimized

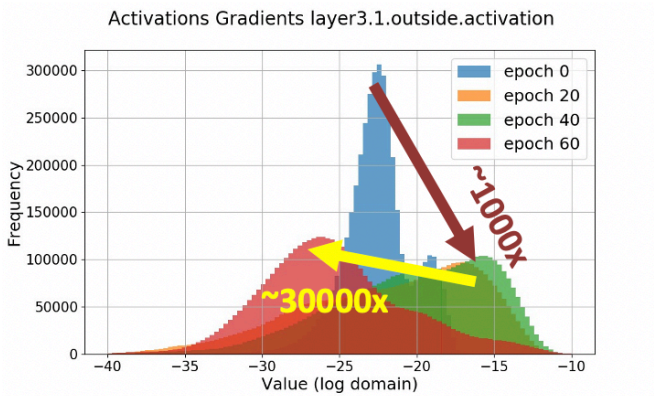


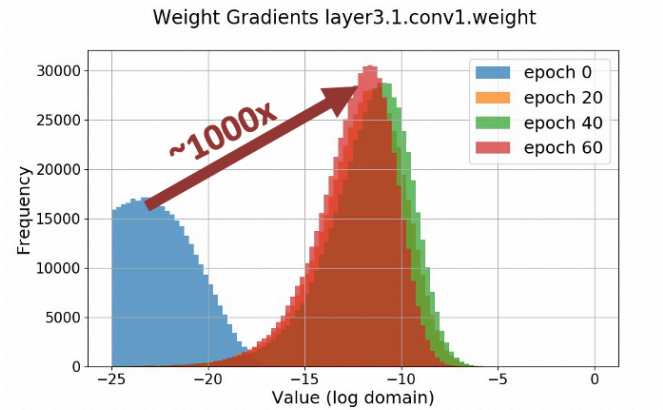






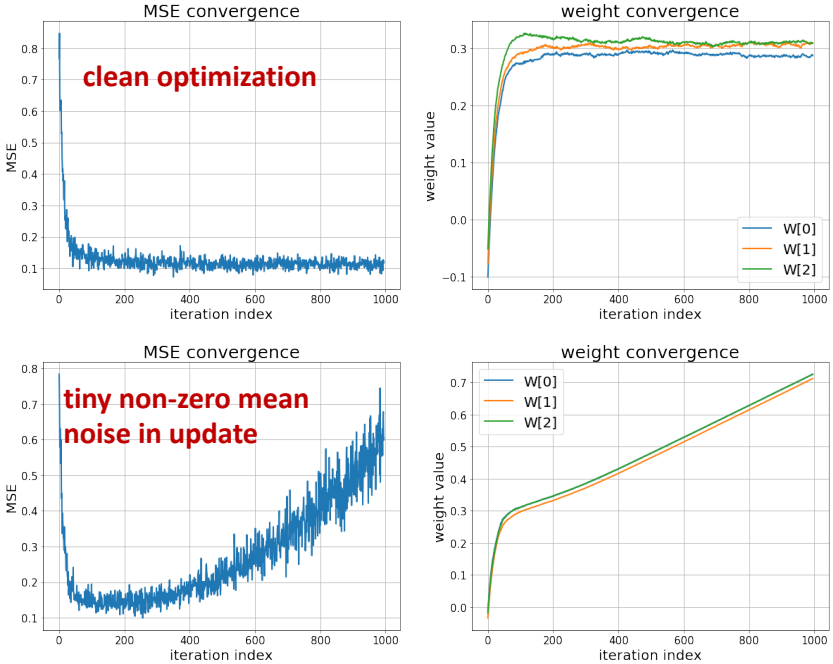
Gradients: Unknown and Time-varying Dynamic Range





- activation gradients: start small -> become larger -> end up very small
  - range fluctuation: up to ~30000x
- weight gradients: end up ~1000x larger than start

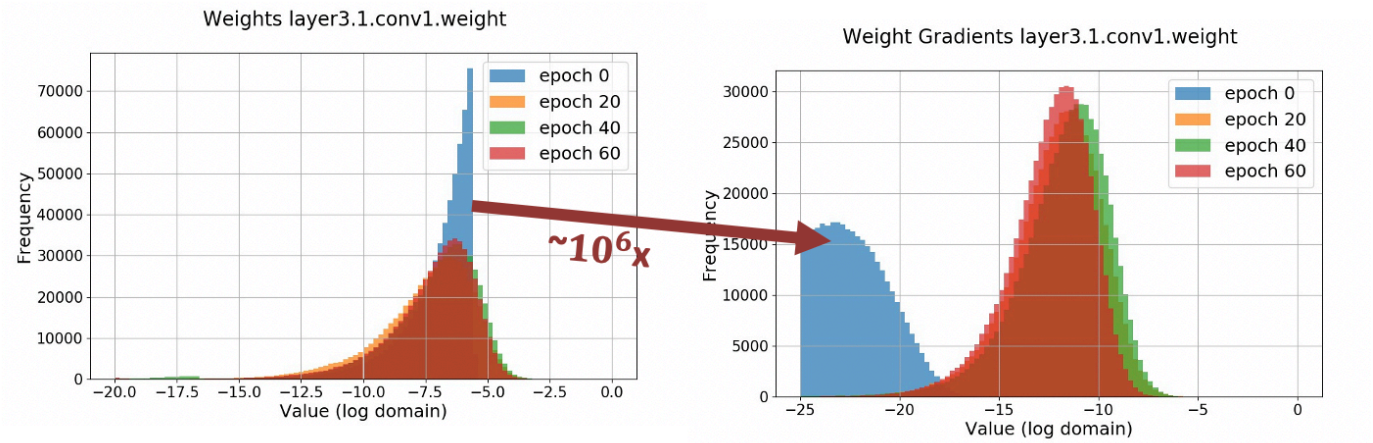
Bias in SGD Loop Leads to Instability



$$\boldsymbol{W}[n+1] = \boldsymbol{W}[n] + \mu e[n]\boldsymbol{X}[n] + \eta \qquad \eta \sim U[0,0.001]$$

- same LMS example as in last time's demo
- adding a tiny non-zero mean noise source in updates unstabilizes the algorithm
- quantization, if not done properly, can turn into such noise source

Huge Dynamic Range Mismatch between Weights and Gradients



- gradients need to be added to weights during updates
- gradients are **extremely small** compared to weights
- gradients multiplied by the learning rate → tinier updates

Criteria-based Solution

Criterion 1: equalization of quantization noise gains

$$B_{W_l} = rnd\left(\log_2\left(\sqrt{\frac{E_{W_l \to p_m}}{E^{(\min)}}}\right)\right) + B^{(\min)} \qquad B_{A_l} = rnd\left(\log_2\left(\sqrt{\frac{E_{A_l \to p_m}}{E^{(\min)}}}\right)\right) + B^{(\min)}$$

Criterion 2: proper gradient clipping

$$r_{G_l^{(W)}} \geq 2\sigma^{(\max)}_{G_l^{(W)}} \qquad r_{G_{l+1}^{(A)}} \geq 4\sigma^{(\max)}_{G_{l+1}^{(A)}}$$

Criterion 3: quantization bias elimination

$$\Delta_{G_l^{(W)}} < \frac{\sigma^{(\min)}_{G_l^{(W)}}}{4}$$

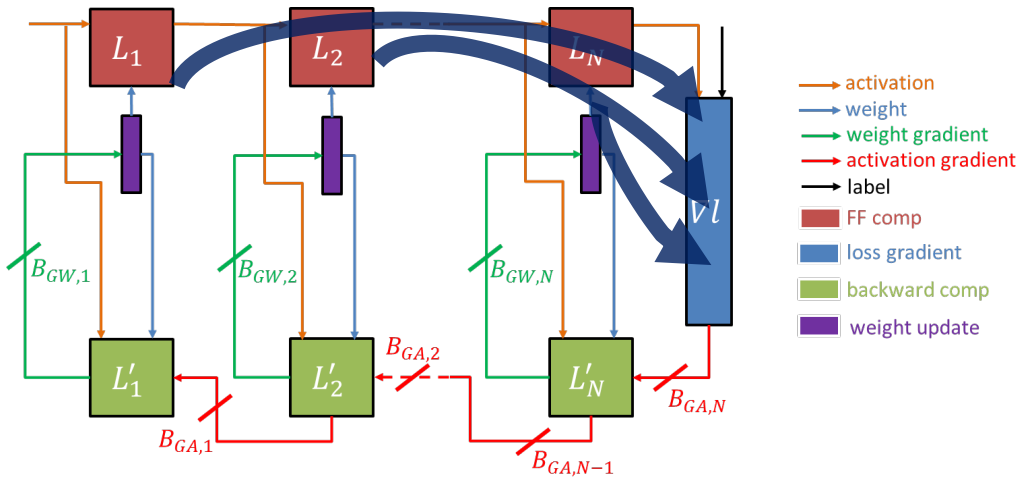
Criterion 4: back-propagated noise bound

$$\Delta_{G_{l+1}^{(A)}} < \frac{\Delta_{G_l^{(W)}}}{\sqrt{\lambda^{(\max)}_{G_{l+1}^{(A)} \to G_l^{(W)}}}} \left(\frac{\left|G_l^{(W)}\right|}{\left|G_{l+1}^{(A)}\right|}\right)^{1/4}$$

Criterion 5: accumulator stopping condition

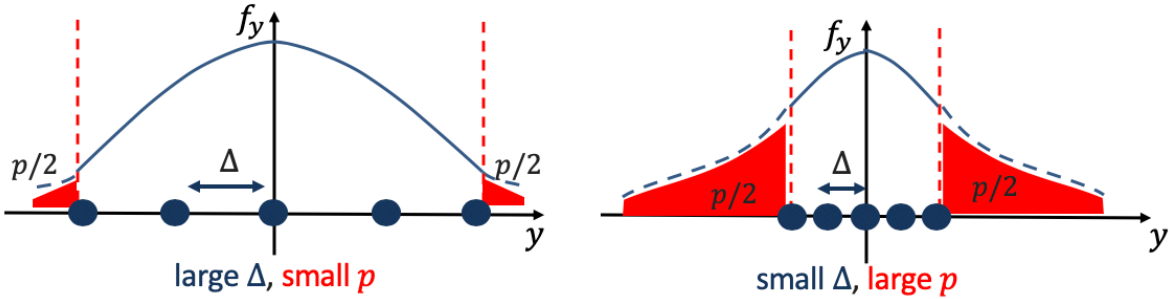
$$r_{W_l^{(acc)}} \geq 2^{-B_{W_l}} \qquad \Delta_{W_l^{(acc)}} < \gamma^{(\min)} \Delta_{G_l^{(W)}}$$

Criterion 1: Equalization of Noise Gains



$$B_{W_l} = rnd\left(\log_2\left(\sqrt{\frac{E_{W_l \to p_m}}{E^{(\min)}}}\right)\right) + B^{(\min)} \qquad B_{A_l} = rnd\left(\log_2\left(\sqrt{\frac{E_{A_l \to p_m}}{E^{(\min)}}}\right)\right) + B^{(\min)}$$

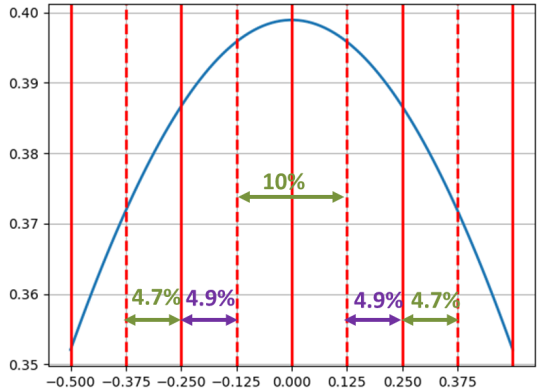
Criterion 2: Proper Gradient Clipping



- need to properly clip y to maximize its SQNR for a fixed number of bits
- need to estimate $y_{\max}$ which is such that $\Pr(|y| > y_{\max})$ is small
- if $y_{\max}$ is overestimated → larger quantization step → more noise
  $SQNR_y = 6B_y + 4.78 - PAR_y$ where $PAR_y = 20\log_{10}\frac{y_{\max}}{\sigma_y}$

$$r_{G_l^{(W)}} \geq 2\sigma_{G_l^{(W)}}^{(\max)} \qquad r_{G_{l+1}^{(A)}} \geq 4\sigma_{G_{l+1}^{(A)}}^{(\max)}$$

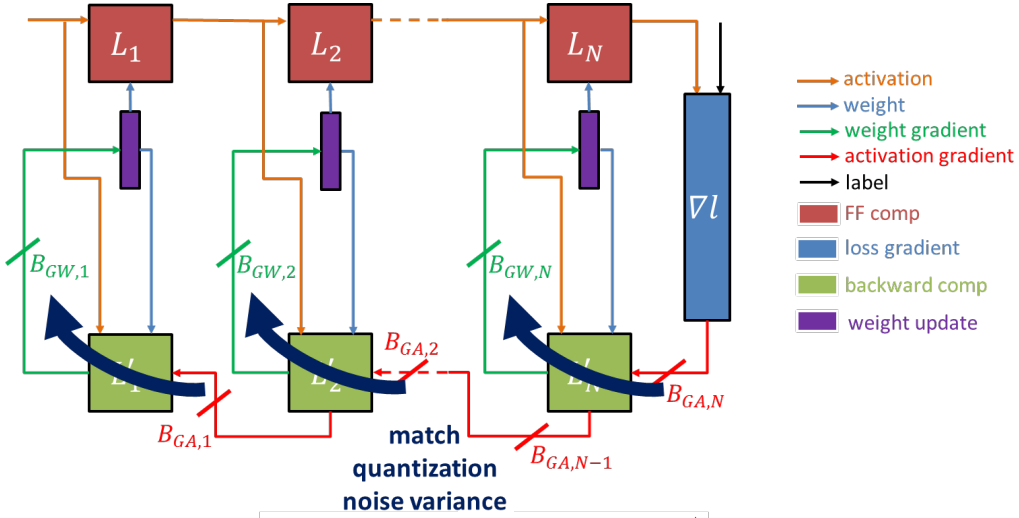
Criterion 3: Quantization Bias Elimination



- when $\text{LSB} \leq \frac{\sigma}{4}$, 4.9% of updates are smaller than the quantization level, and 4.7% are greater → very small quantization bias ($\leq 1\%$)

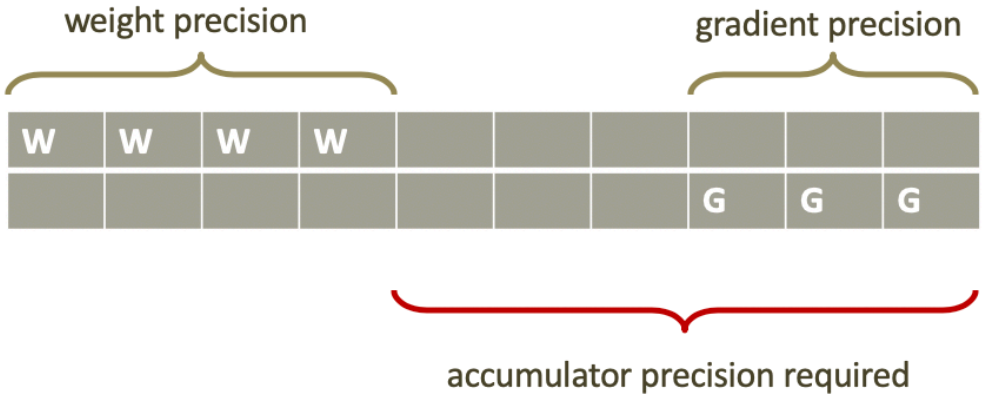
$$\Delta_{G_l^{(W)}} < \frac{\sigma_{G_l^{(W)}}^{(\min)}}{4}$$

Criterion 4: Back-propagated Noise Bound



$$\Delta_{G_{l+1}^{(A)}} < \frac{\Delta_{G_l^{(W)}}}{\sqrt{\lambda^{(\max)}_{G_{l+1}^{(A)} \to G_l^{(W)}}}} \left(\frac{\left| G_l^{(W)} \right|}{\left| G_{l+1}^{(A)} \right|} \right)^{1/4}$$

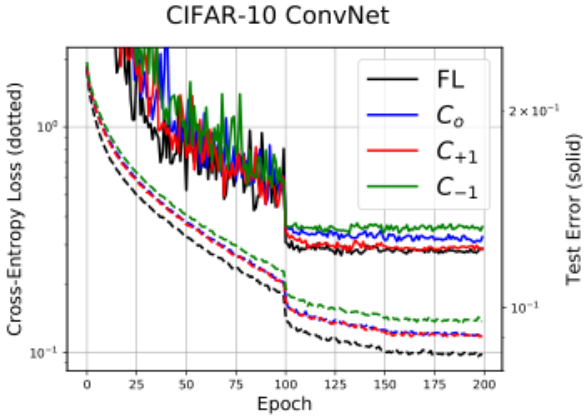
Criterion 5: Accumulator Stopping Criterion

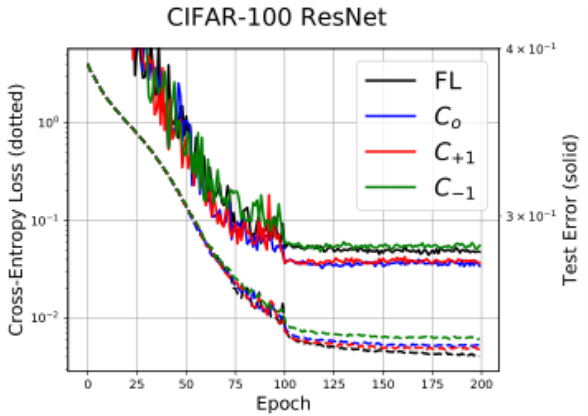


$$r_{W_l^{(acc)}} \geq 2^{-B_{W_l}}$$

$$\Delta_{W_l^{(acc)}} < \gamma^{(\min)} \Delta_{G_l^{(W)}}$$

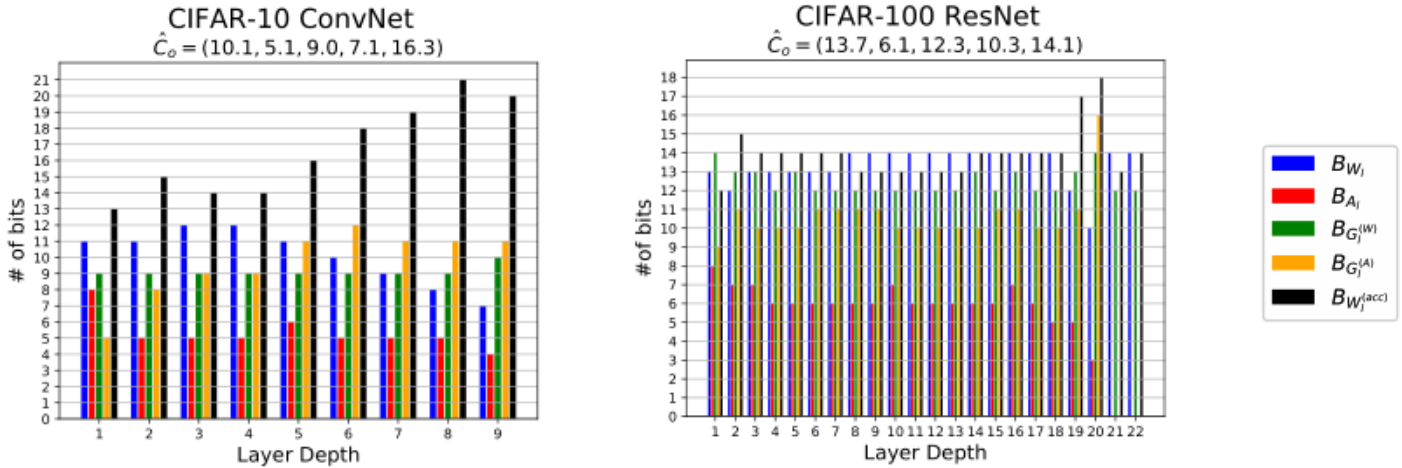
FX Training Converges with Close-to-Minimal Precisions





- FX training was believed to be impossible due to dynamic range issues [Koester et al. – NIPS'2017]
- proposed FX training is able to match FL training accuracy
- precision assignment found to be nearly minimal

Per-layer Precision Trends Vary



- weight precision decreases from network input to output
- precisions of activation gradients and weight accumulators increase
- ResNets have uniform precision requirements per tensor type

Comparison w.r.t. Hyper-precision Reduction Techniques

	$\mathcal{C}_W$ (10^6b)	$\mathcal{C}_A$ (10^6b)	$\mathcal{C}_M$ (10^9FA)	$\mathcal{C}_C$ (10^6b)	Test Error	$\mathcal{C}_W$ (10^6b)	$\mathcal{C}_A$ (10^6b)	$\mathcal{C}_M$ (10^9FA)	$\mathcal{C}_C$ (10^6b)	Test Error
	CIFAR-10 ConvNet					**CIFAR-100 ResNet**				
FL	148	9.3	94.4	49	12.02%	1789	97	4319	597	28.06%
FX (C_o)	**56.5**	**1.7**	11.9	14	12.58%	**750**	**25**	776	216	**27.43%**
BN	100	4.7	**2.8**	49	18.50%	1211	50	**128**	597	29.35%
SQ	78.8	**1.7**	11.9	14	**11.32%**	1081	**25**	776	216	28.03%
TG	102	9.3	94.4	**3.1**	12.49%	1230	97	4319	**37.3**	30.62%

- feedforward binarization (BN) and gradient ternarization (TG) fail to match FL accuracy for same topology
- stochastic quantization (SQ) provides marginal returns
- BN, TG, SQ do not address the fundamental problem of realizing true FX training

Course Web Page

https://courses.grainger.illinois.edu/ece598nsg/fa2020/
https://courses.grainger.illinois.edu/ece498nsu/fa2020/

http://shanbhag.ece.uiuc.edu