Standard Deviation-Based Quantization for Deep Neural Networks

CMC Accelerating AI Workshop

Warren J. Gross

McGill University

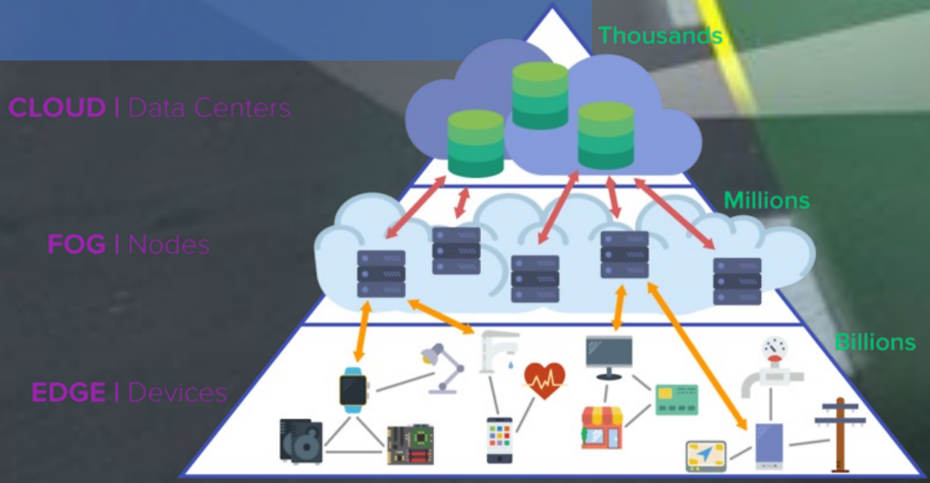
May 4, 2023

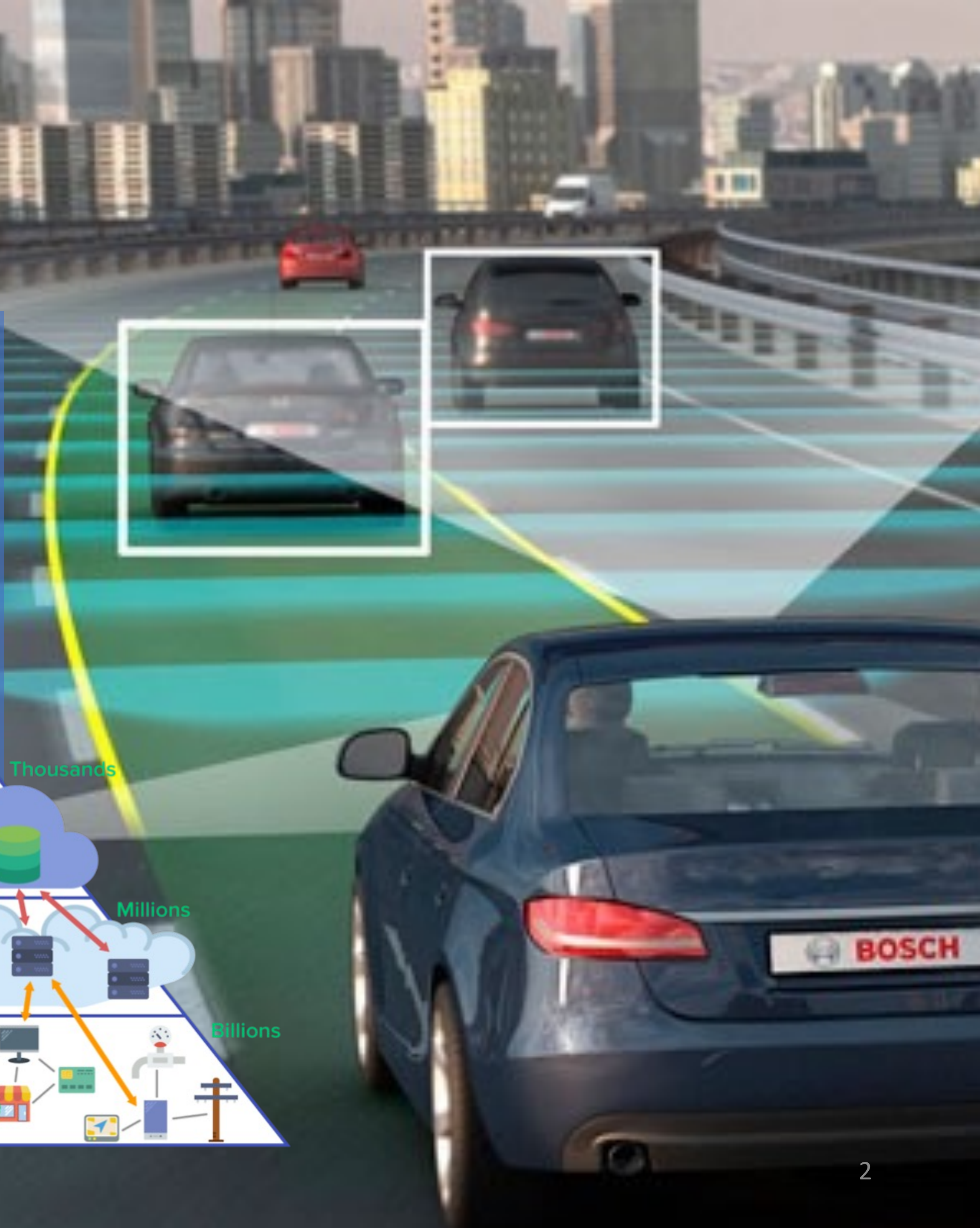
Edge Intelligence

Data is curated and processed entirely on-device

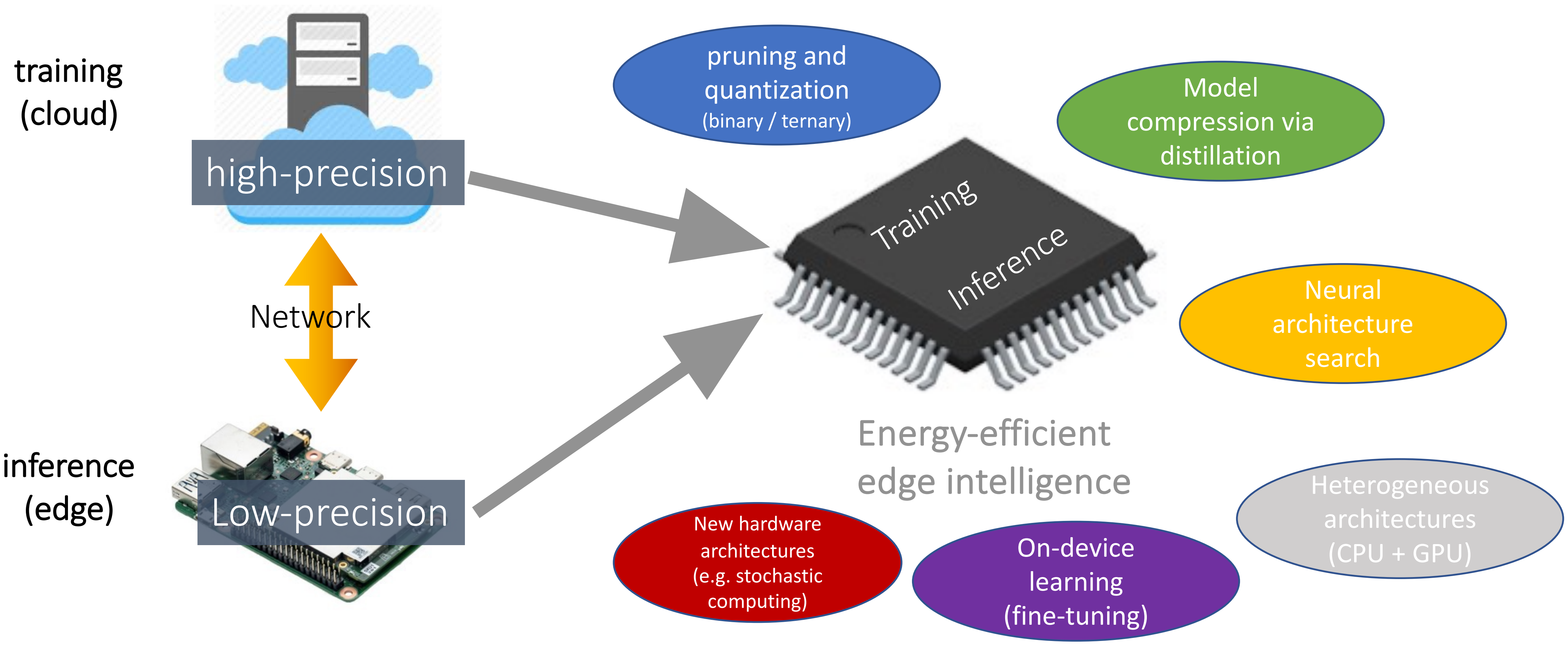
- *Cybersecurity and privacy are built in*
- *Quick adaptability to environment*

Processing the data can be very fast – avoid the latency and power required to communicate with the cloud

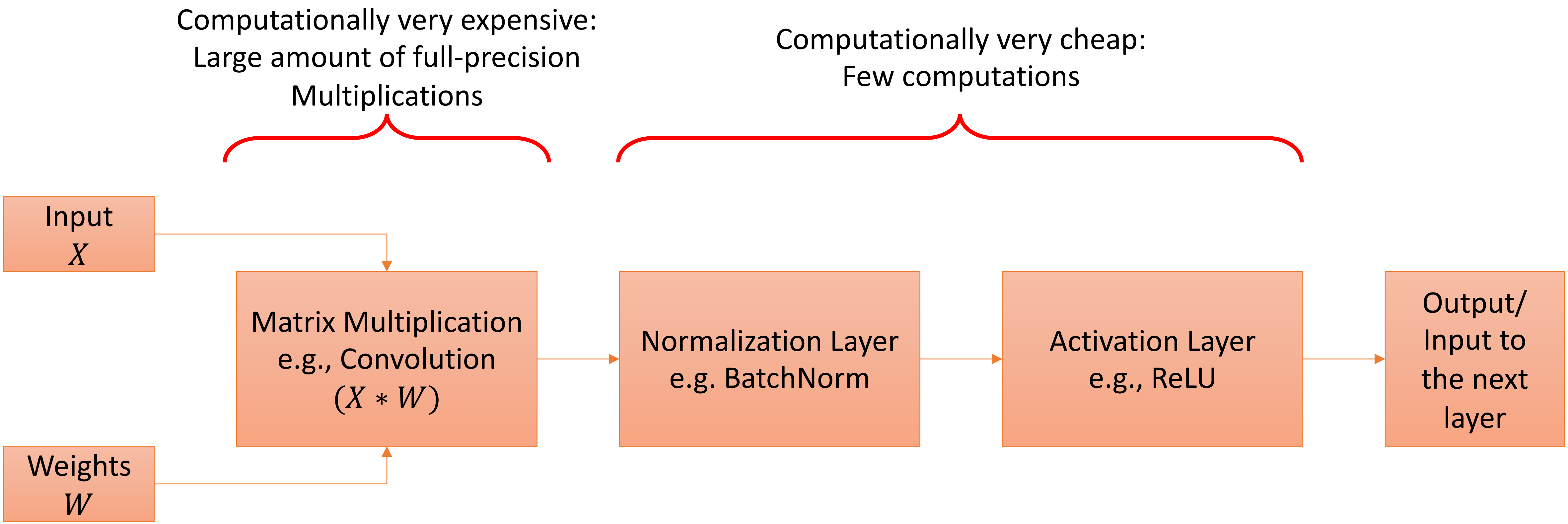




Our goal

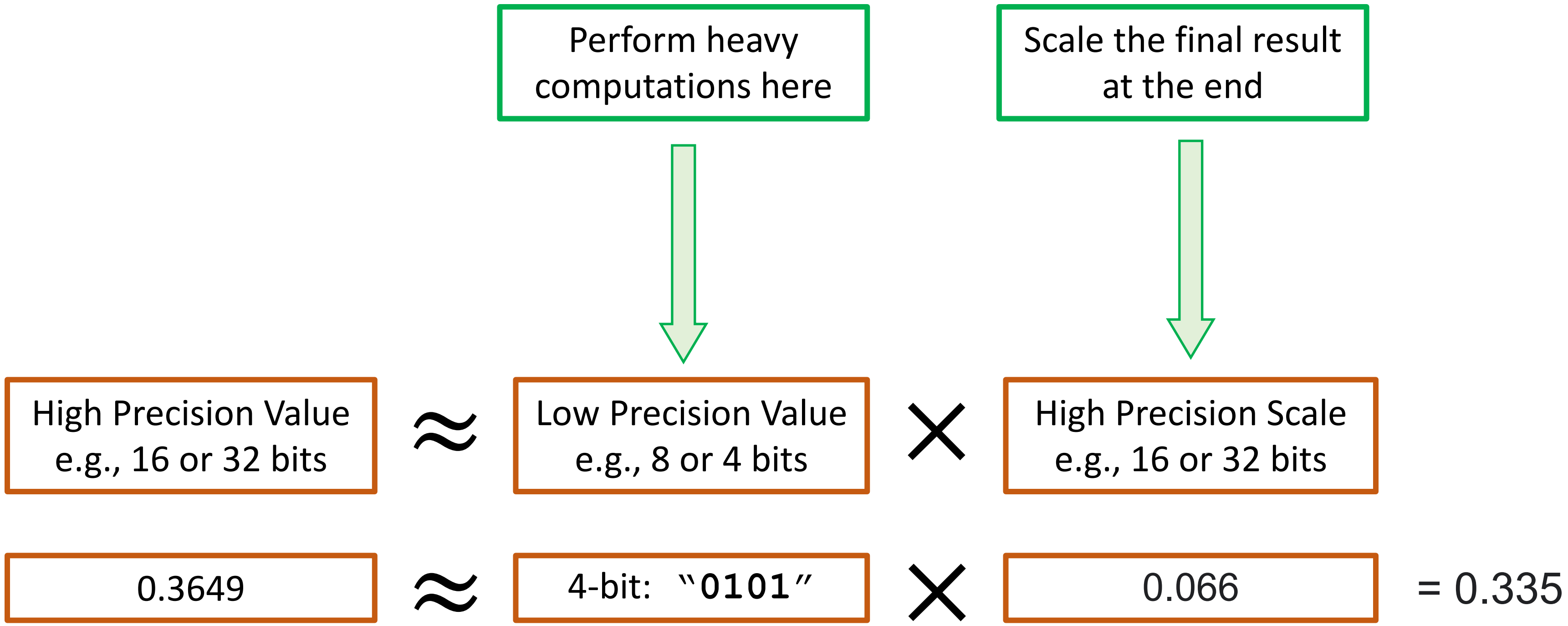


training
(cloud)
high-precision
Network
inference
(edge)
Low-precision
pruning and
quantization
(binary / ternary)
Model
compression via
distillation
Training
Inference
Neural
architecture
search
Energy-efficient
edge intelligence
Heterogeneous
architectures
(CPU + GPU)
New hardware
architectures
(e.g. stochastic
computing)
On-device
learning
(fine-tuning)



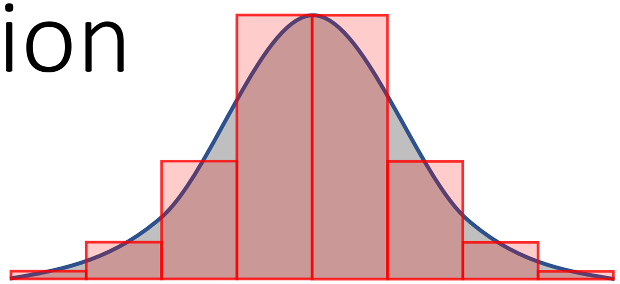
- Quantize X, W with fewer bits
- Reduce the cost of computations by using integer multipliers

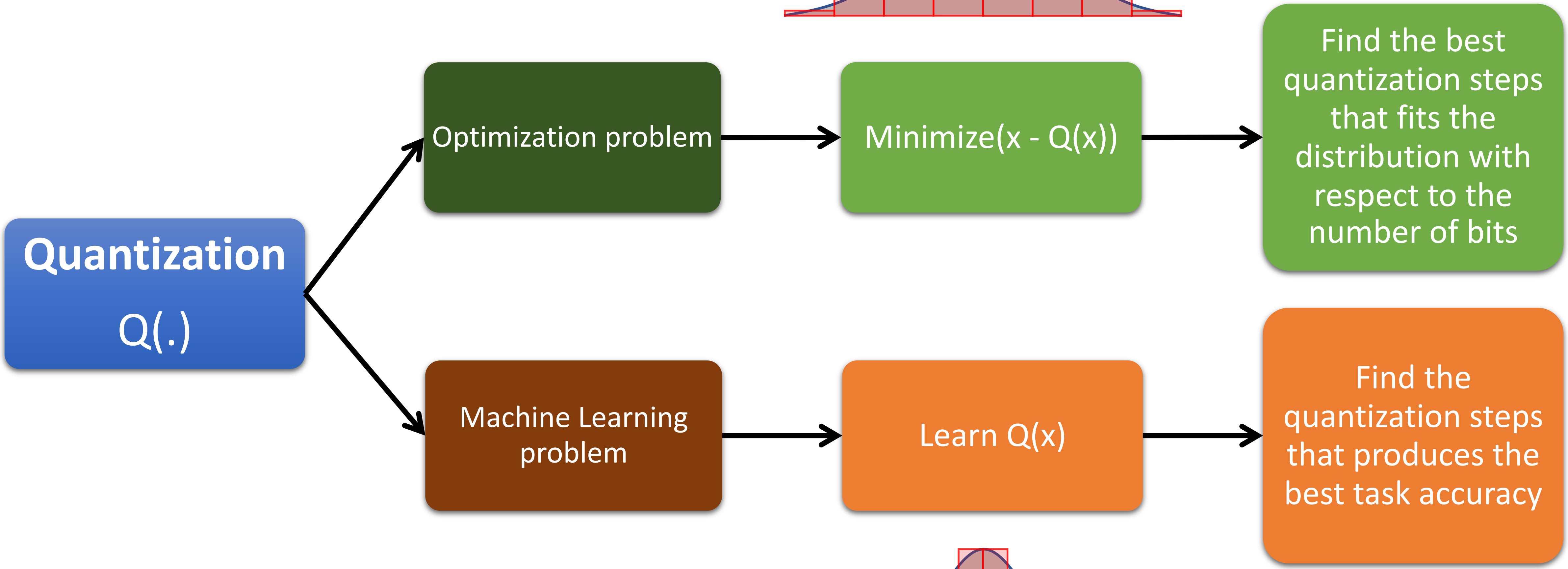
What is Quantization

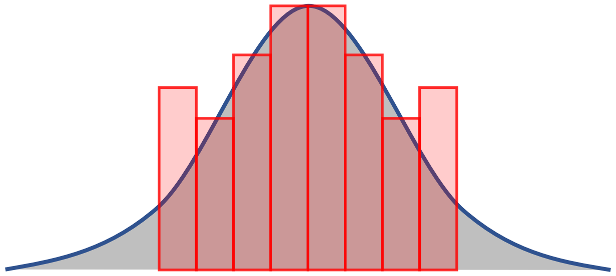


- Layer-by-layer quantization (same number of bits, but different scaling values)
- Each layer has a potentially unique quantization (scale). Rescale to higher precision in the the normalization layer.

Two approaches to quantization

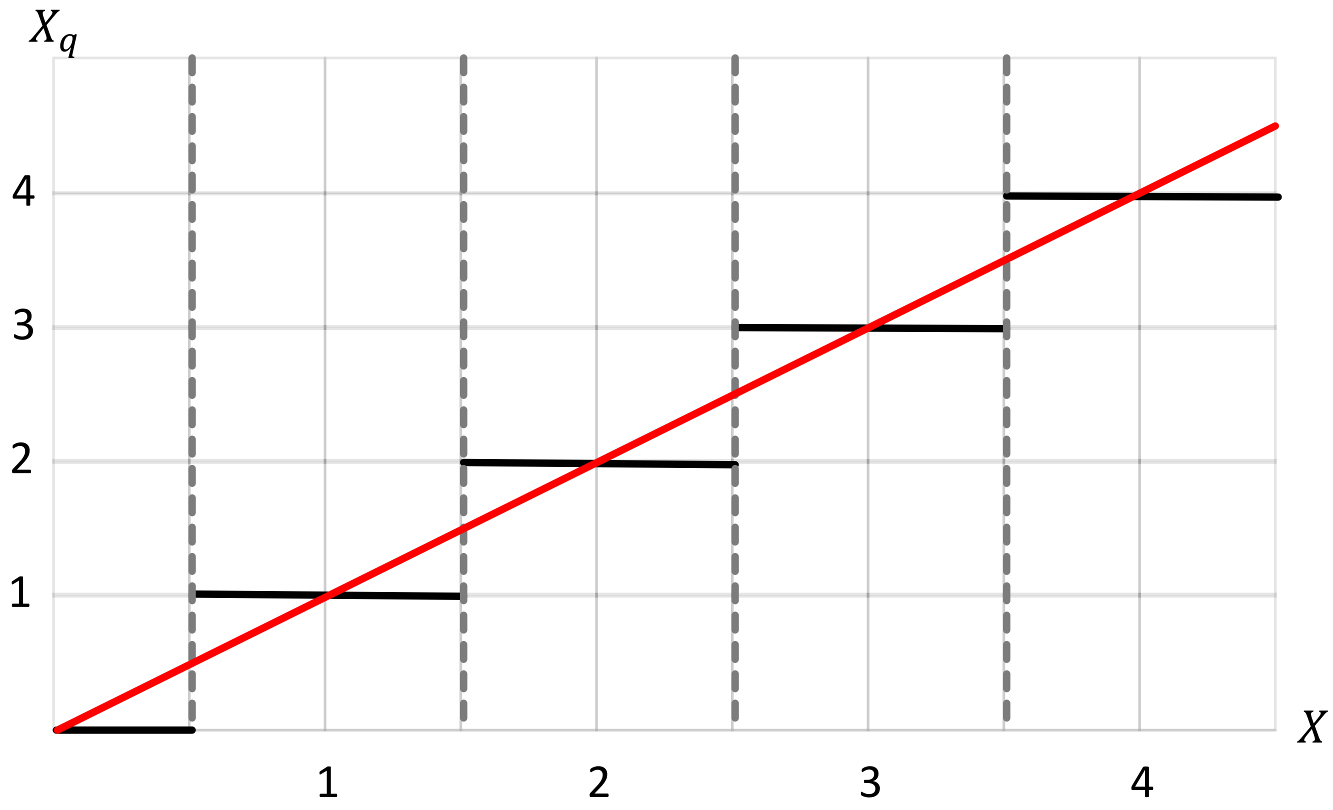






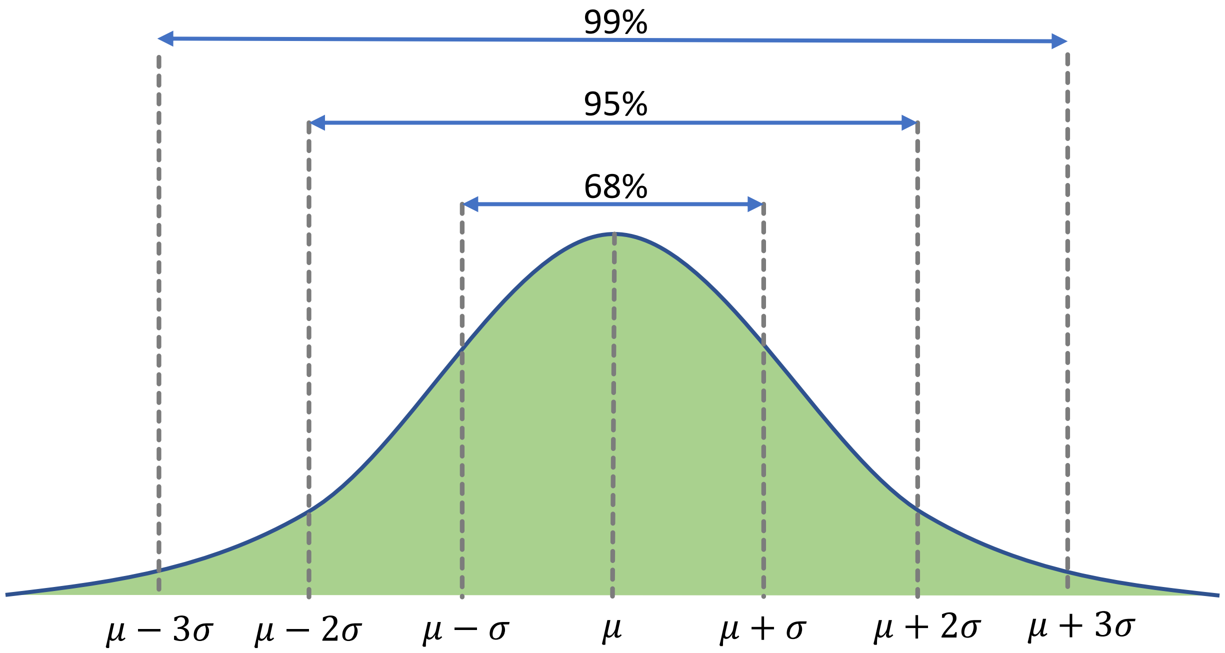
Uniform (Linear) quantization

- All steps (quantization intervals) are equal

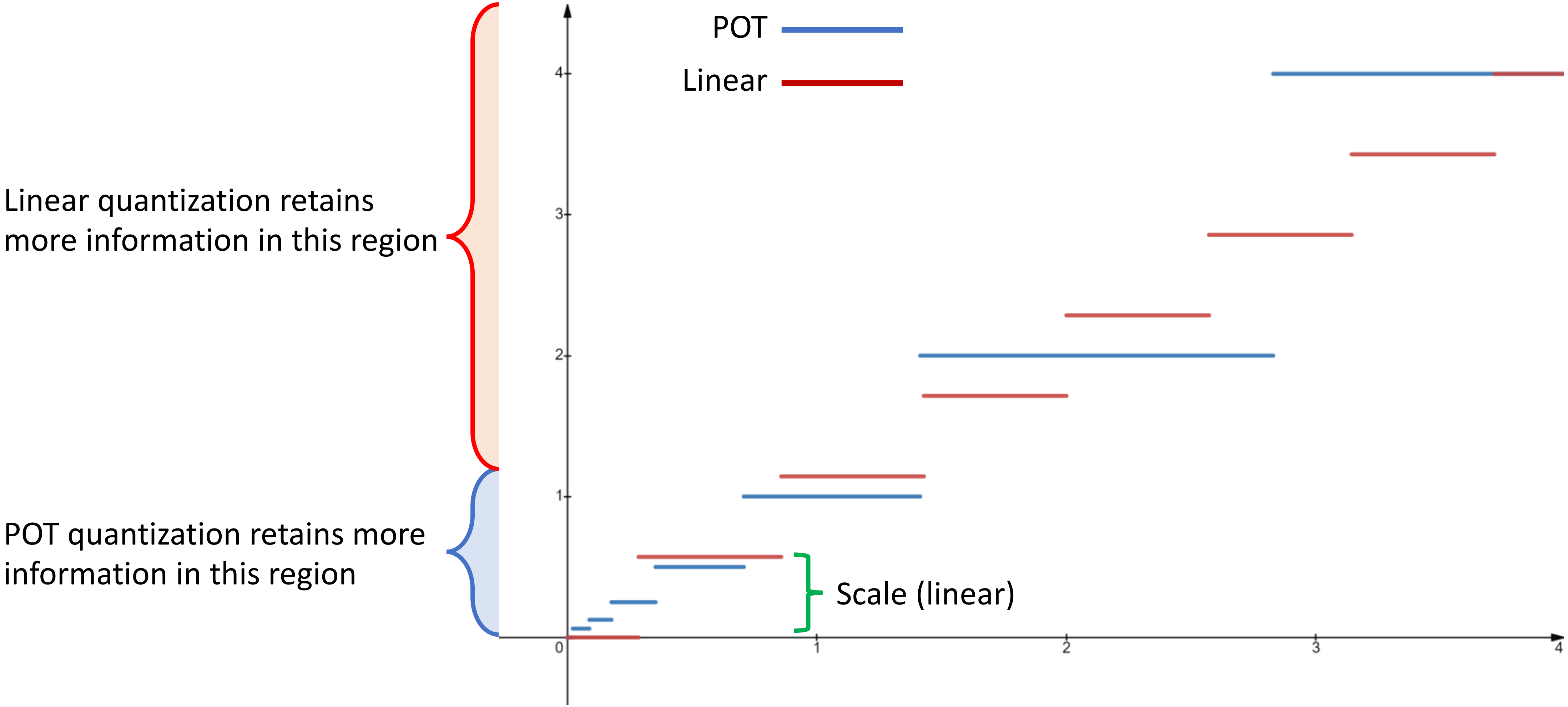


Weights and Activation Distribution

- L2-norm gradient descent (weight decay factor) -> Gaussian-like distribution of weights
- Batch normalization -> Gaussian-like distribution of activations
- Huge portion of weights and activation values reside near zero



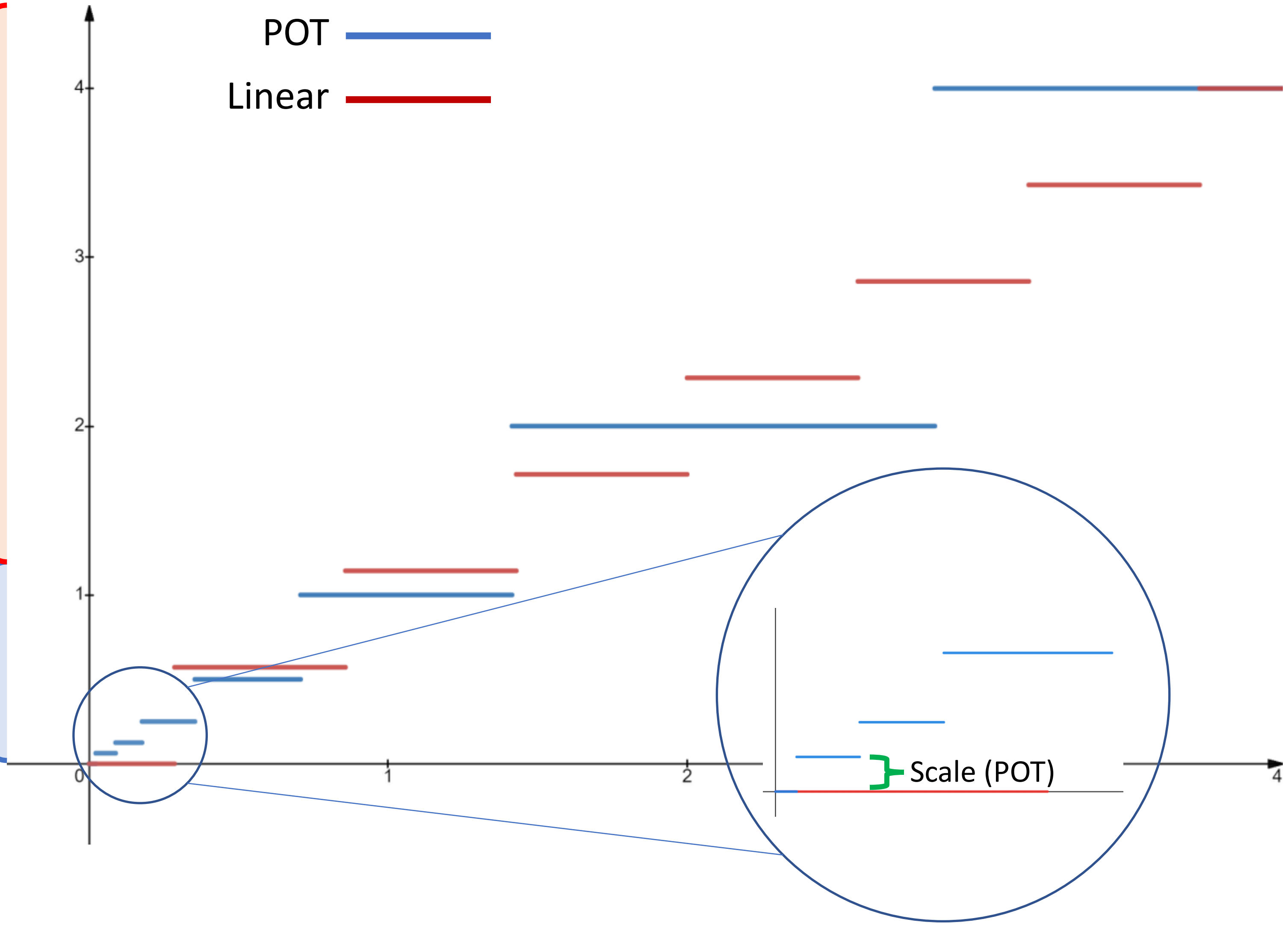
Non-Uniform/Power of Two (POT) Quantization



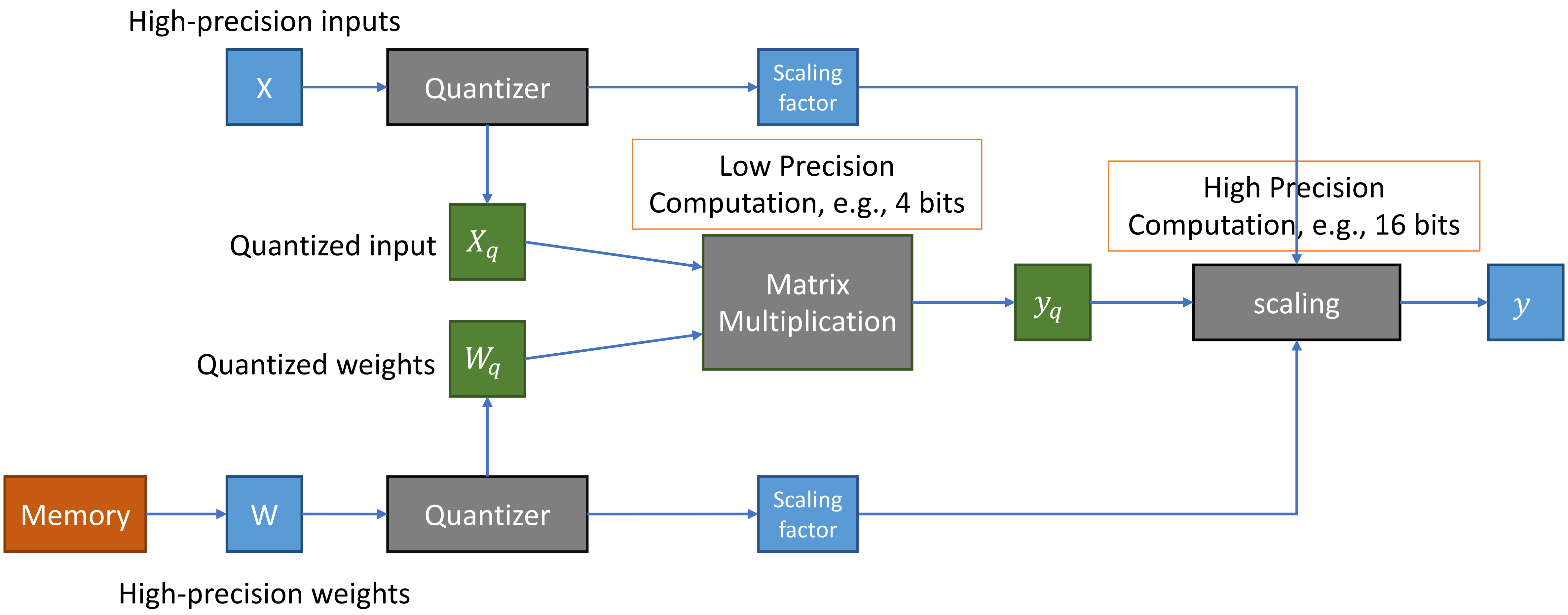
Non-Uniform/Power of Two (POT) Quantization

Linear quantization retains more information in this region

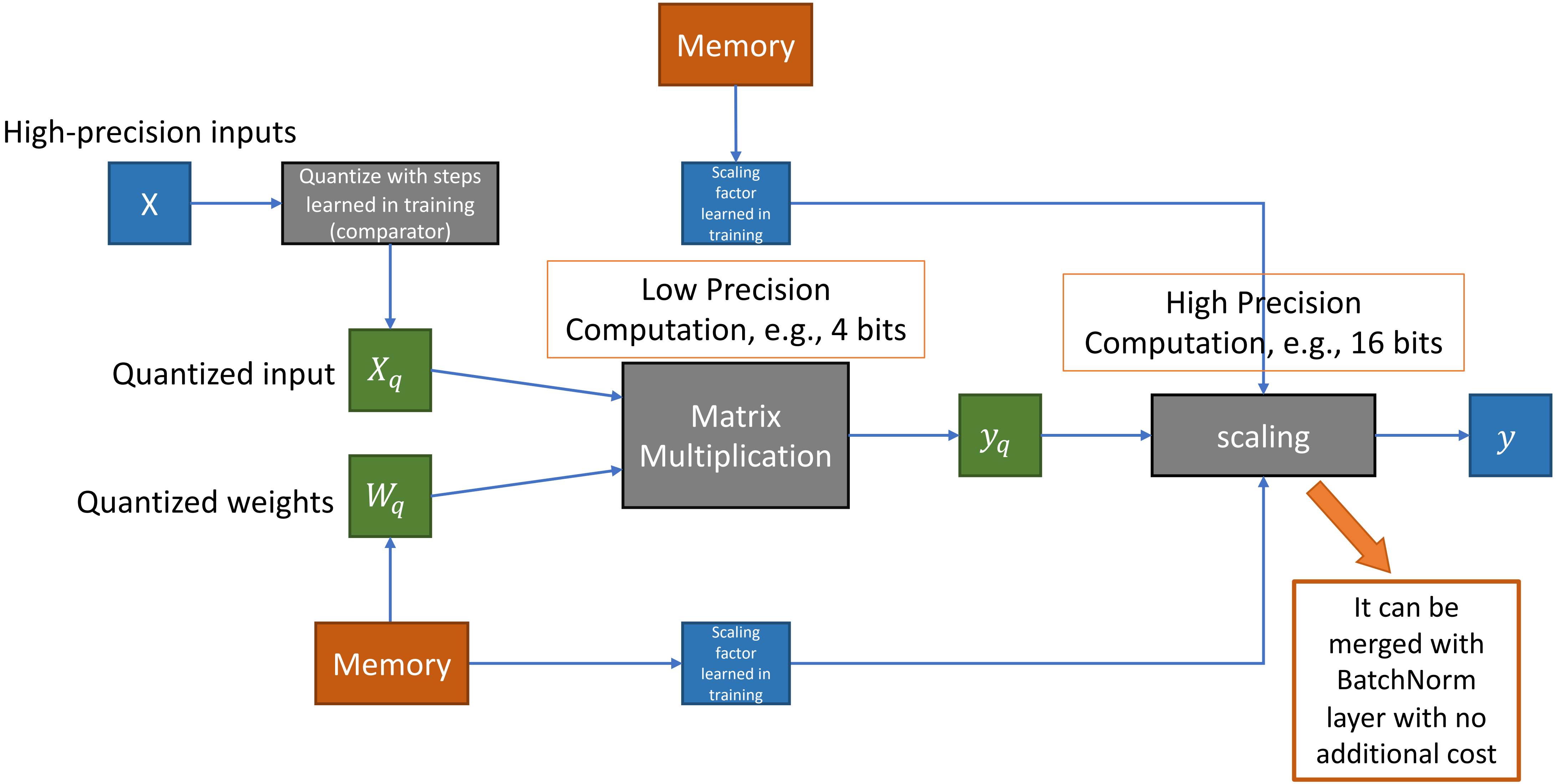
POT quantization retains more information in this region



Computations in Quantized Network During Training Phase



Computations in Quantized Network During Inference



Quantization (Training Phase)

- Normalize data so that $\text{abs}(x) \in [0,1]$, with minimum value 0 and maximum value 1
- We can normalize x by first *clipping* and dividing $\text{abs}(x)$ by a parameter (α) :

$$\frac{\text{clip}(\text{abs}(x), 0, \alpha)}{\alpha}$$

- Find α by solving the optimization problem --> $\min_{\alpha}(Q(x) - x)^2$ where $Q(x)$ is the quantizer function

or

- Find α by gradient descent and backpropagation --> $\min_{\alpha,w} L(\hat{y}, y)^2$ where $\hat{y}$ is the network output and y is the ground truth

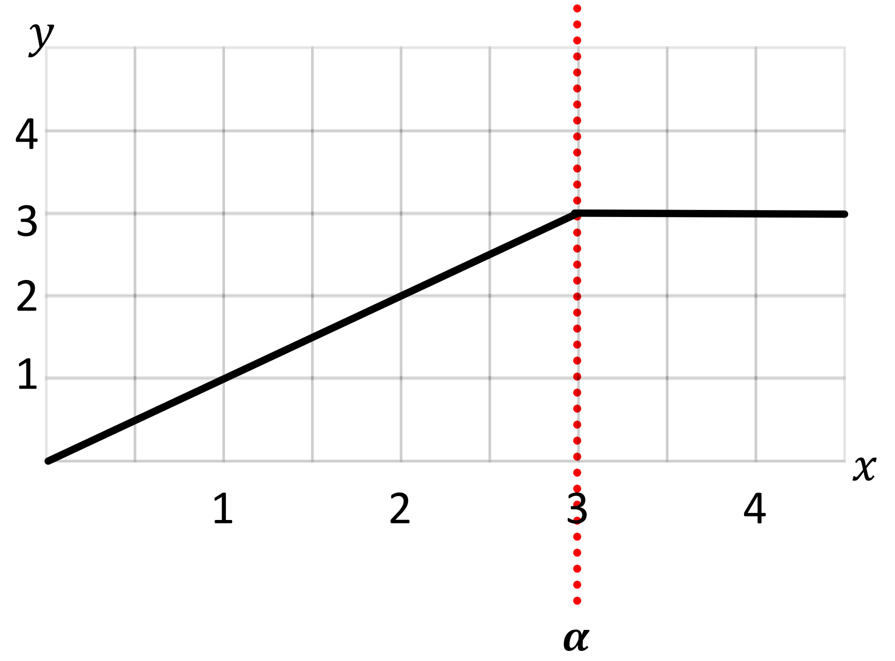
Parameterized Clipping Threshold (PACT[1])

Goal: Learn the quantization steps

Parameterized Clipping Function: $y(x) = \begin{cases} 0, & x \in (-\infty, 0) \\ x, & x \in [0, \alpha) \\ \alpha, & x \in [\alpha, +\infty) \end{cases}$

Backpropagation:

$$\frac{\partial L}{\partial \alpha} = \begin{cases} \frac{\partial L}{\partial x_q}, & x > \alpha \\ 0, & otherwise \end{cases}$$



- Problem: Single parameter alpha for the entire layer – all the gradients coming to the layer are summed –> large value for updating alpha –> alpha grows quickly to a large value or vanishes
- Condition for good convergence: ratio of average gradient magnitude ≈ average parameter magnitude[2]
- We must scale the gradient for $\boldsymbol{\alpha}$
    - Divide it by #features, #weights or a constant value

[1] J. Choi et al., Pact: Parameterized clipping activation for quantized neural networks, arXiv preprint arXiv:1805.06085 (2018).
[2] Y. You et al., Large batch training of convolutional networks, arXiv preprint arXiv:1708.03888 (2017).

Our Approach

Intuition:

- Give a meaning to alpha: how many STD from mean to keep
- Linking the two methods: give the optimizer knowledge of the distribution

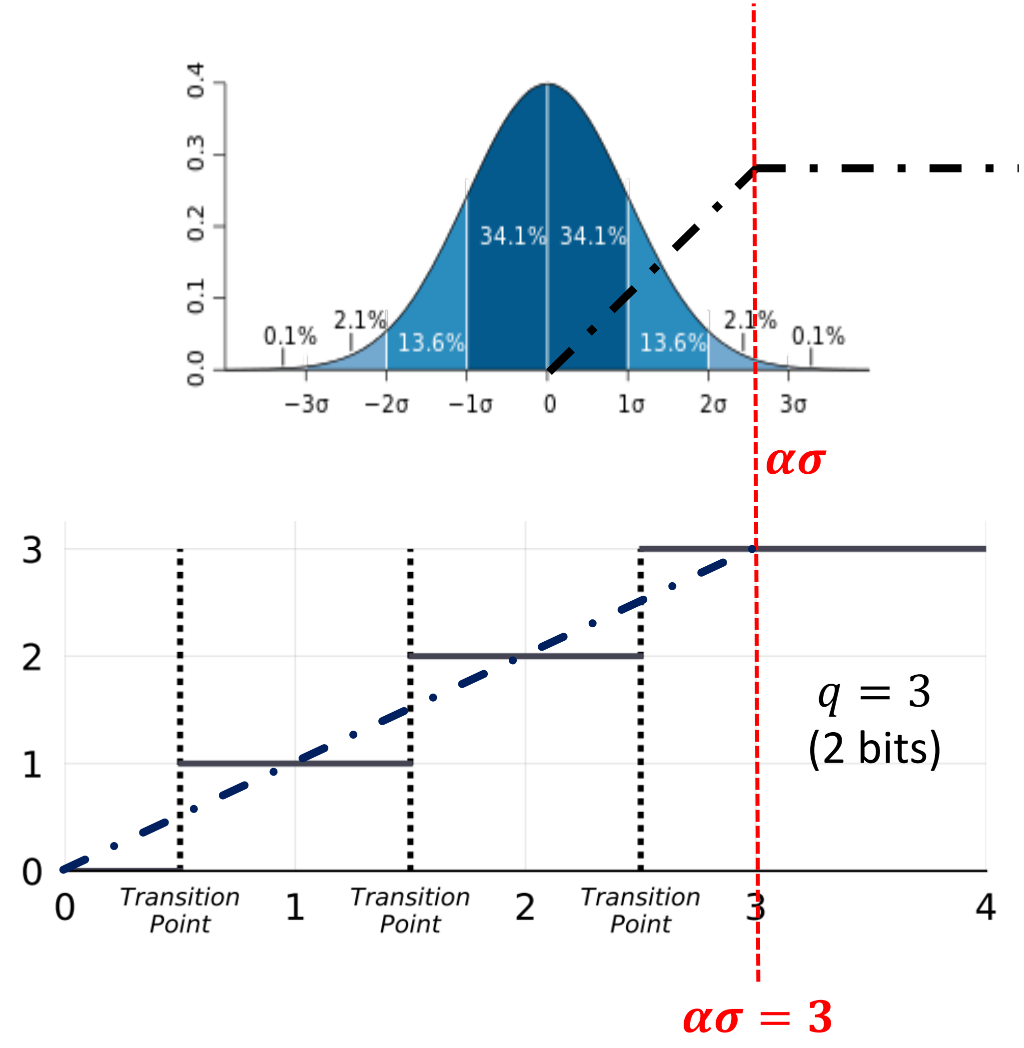
$$y(x) = \begin{cases} 0, & x \in (-\infty, 0) \\ x, & x \in [0, \alpha\sigma) \\ \alpha\sigma, & x \in [\alpha\sigma, +\infty) \end{cases}$$

$$\frac{\partial L}{\partial \alpha} = \begin{cases} \frac{\partial L}{\partial y} \times \frac{\partial Q(x)}{\partial \alpha} = \frac{\partial L}{\partial y} \times \boldsymbol{\sigma}, & abs(x) > \boldsymbol{\alpha\sigma} \\ 0, & otherwise \end{cases}$$

$$\frac{\partial L}{\partial \alpha} = s \frac{\partial L}{\partial y} \sigma + \lambda \alpha$$

s: constant gradient scaling factor

λ: decay factor



A. Ardakani, et al. "Standard Deviation-Based Quantization for Deep Neural Networks." *arXiv preprint arXiv:2202.12422* (2022)

Results on CIFAR10 – Linear Quantization ResNet-20 and Small-VGG Architectures

ResNet-20 – FP (32-bit float) Accuracy = 91.8%

Quantization method		Accuracy @ precision (A and W)			
Activations	Weights	5	4	3	2
DSQ	DSQ	–	–	–	90.11
LQ-Net	LQ-Net	–	–	91.6	90.2
PACT	DoReFa	91.7	91.3	91.1	89.7
Ours	DoReFa	92.03	92.00	91.65	90.32
Ours	Ours	**92.27**	**92.28**	**92.23**	**90.77**

Small-VGG - FP Accuracy = 93.6%

Method	All layers	Accuracy @ precision (A/W)	
		2/1	2/2
LQ-Net	No	93.40	93.50
HWGQ	No	92.51	NA
LLSQ	No	NA	93.31
Ours	No	**93.88**	**94.36**
RQST	Yes	NA	90.92
LLSQ	Yes	NA	93.12
Ours	Yes	NA	**93.90**

DSQ: R. Gong et al. (ICCV 2019), **LQ-Net**: D. Zhang et al. (ECCV 2018), **PACT**: J. Choi et al. (arXiv preprint 2018)
HWGQ: Z. Cai et al. (CVPR 2017), **LLSQ**: X. Zhao et al. (ICLR 2020), **RQST**: C. Louizos et al. (arXiv preprint 2018)
DoReFa: S. Zhou et al. (arXiv preprint 2016)

Results on ImageNet - Linear Quantization

ResNet-34

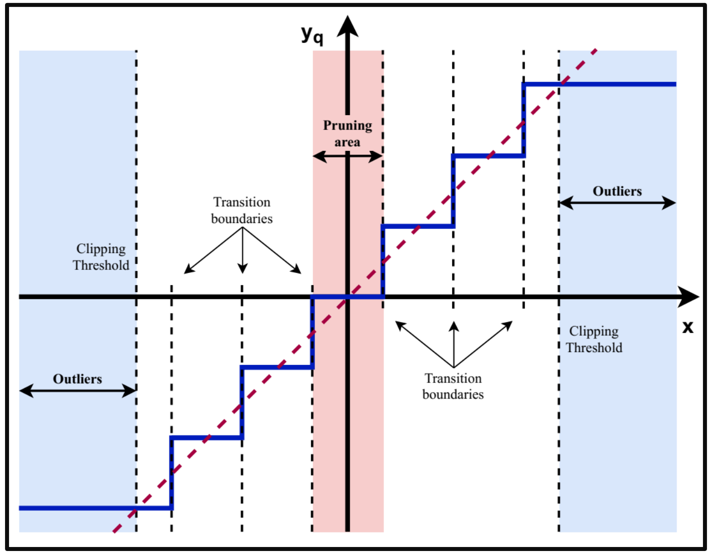
Method	Top-1 Accuracy @Precision		
	FP	4	3
Ours	73.3	**73.5 (+0.2)**	**73.2 (-0.1)**
LSQ*	74.1	74.1 (0)	73.4 (-0.7)
QIL	73.7	73.7 (0)	73.1 (-0.6)

AlexNet – FP Accuracy = 61.8

Method	Top-1 accuracy @ precision (A and W)		
	4	3	2
Ours	**62.5**	**62.2**	**59.2**
QIL	62	61.3	58.1
LQ-Net	–	–	57.4
TSQ	–	–	58

*LSQ method uses Pre-Activation variant of ResNet which has a higher performance than the original architecture.

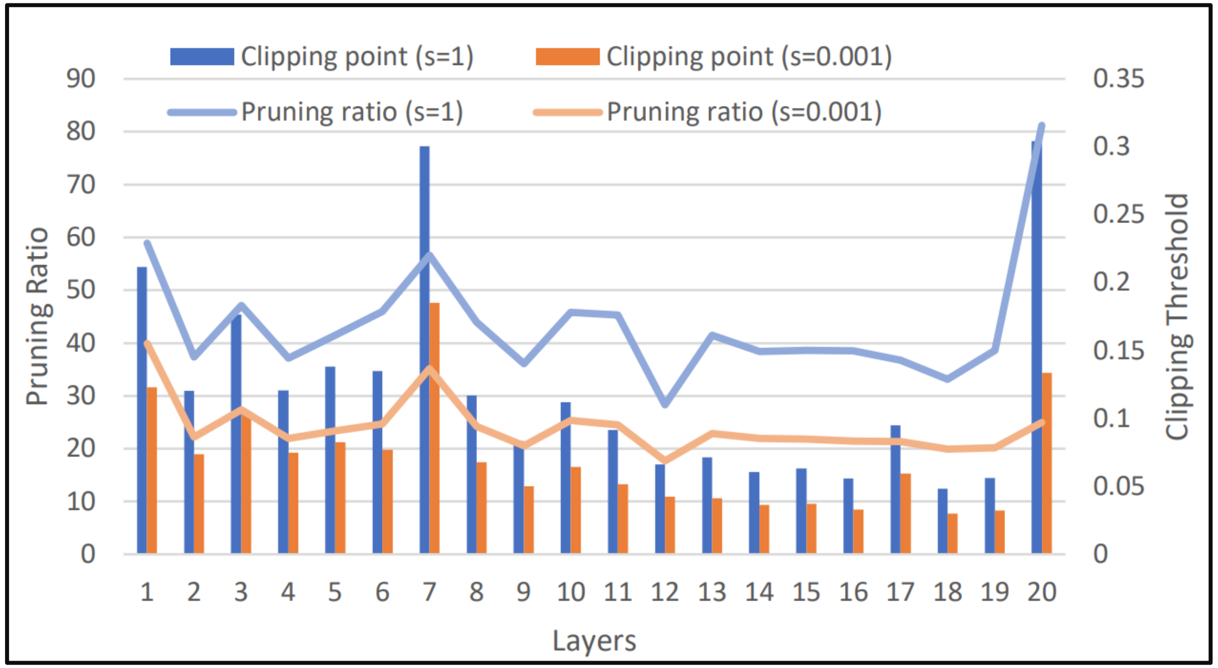
Pruning VS. Clipping Point



- ➢ Pruning ratio can be adjusted by varying gradient scale (S)
- ➢ We can achieve **18% more pruning ratio** at the cost of **1.12% accuracy loss**

3-bit Quantized ResNet-18

Gradient scale	1	0.1	0.01	0.001
Top-1 Acc. (%)	68.92	69.59	69.70	70.04
Pruning ratio (%)	40.21	29.74	24.97	21.57



Log_2 (Power of Two) Quantization

Compared to S^3: 4x less memory during training
10x faster convergence

Results on ImageNet

Model	Method	Width	Top-1/Top-5 Acc. (%)
ResNet-18	FP	32	69.76/89.08
	DeepShift	5	69.56/89.17
	INQ	3	68.08/88.36
	S^3	3	69.82/89.23
	Ours	3	**70.23/89.33**
ResNet-50	FP	32	76.13/92.86
	INQ	5	74.81/92.45
	DeepShift	5	76.33/93.05
	S^3	3	75.75/92.80
	Ours	3	**76.37/93.08**

[Deepshift] M. Elhoushi et al., "Deepshift: Towards multiplication-less neural networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021
[INQ] A. Zhou et al., "Incremental network quantization: Towards lossless cnns with low-precision weights." *arXiv preprint*, 2017
[S^3] X. Li et al., "S^3 : Sign-sparse-shift reparametrization for effective training of low-bit shift networks." Advances in Neural Information Processing Systems, 34, 2021

Conclusion

- Proposed new quantization method that takes advantage of the knowledge of weights and activation distributions (stddev)
- The proposed method outperforms prior art in various image classification tasks
- The non-uniform base-2 logarithmic quantization method converges 10x faster than the SOTA
- Flexibility to trade-off accuracy and network size by controlling the pruning ratio
- Future work: apply this method to transformers