DNN Quantization: Theory to Practice

Dwith Chenna

MTS Product Engineer, AI Inference
AMD Inc.

Content

- Why Quantization?
- Quantization Schemes
- DNN Model Quantization
- Quantization Aware Training (QAT)
- Post Training Quantization (PTQ)
- Quantization Analysis
- Quantization: Best Practices

Why Quantization?

- Model compression techniques are crucial for edge computing, reducing deep learning model size for lower memory and processing needs
  - Knowledge Distillation
  - Pruning / Sparsity
  - **Quantization**
  - Network Architecture Search (NAS)

Quantization Scheme

- Quantization is the process of mapping real numbers, denoted as "r", to quantized integers, represented as "q"
  - Symmetric Quantization

$$q = round(r/S)$$

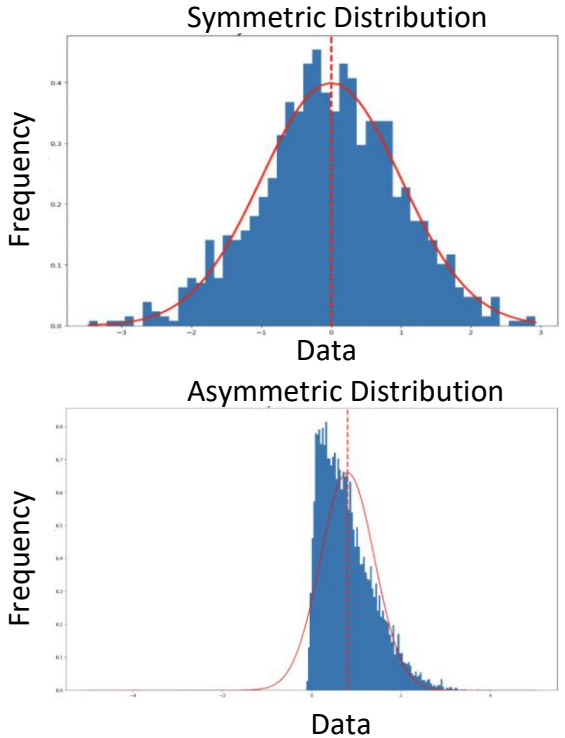
  - Asymmetric Quantization

$$q = round(r/S + Z)$$

where "S" is the scale and "Z" is the zero points

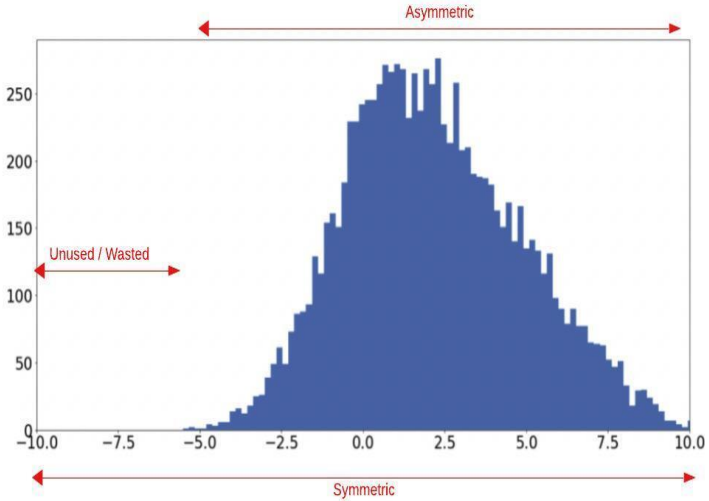
$$S = (r\_max - r\_min) / (q\_max - q\_min)$$

$$Z = round\ (q\_max - r\_max / S)$$



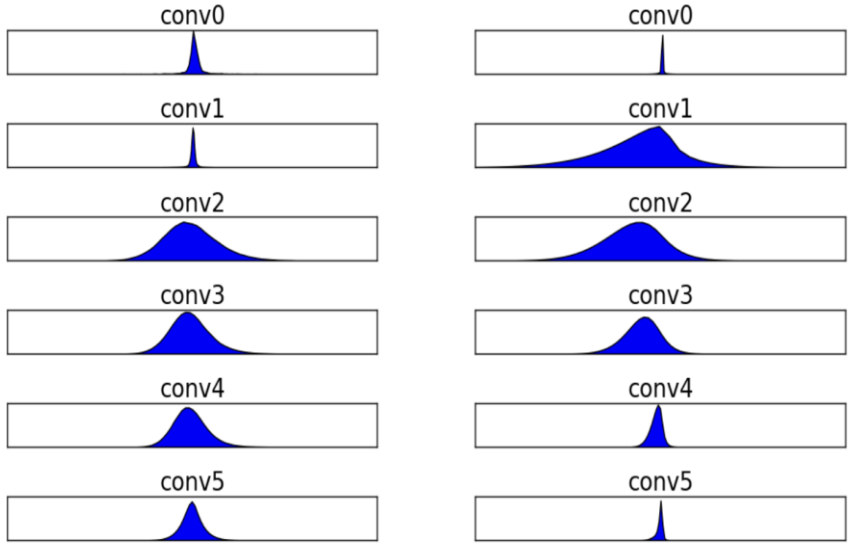
Quantization Scheme

- Symmetric vs asymmetric quantization
- Choice of quantization scheme depends on data distribution
- Make the best use of bit precision
- Avoid outliers in the data distribution



DNN Model Quantization

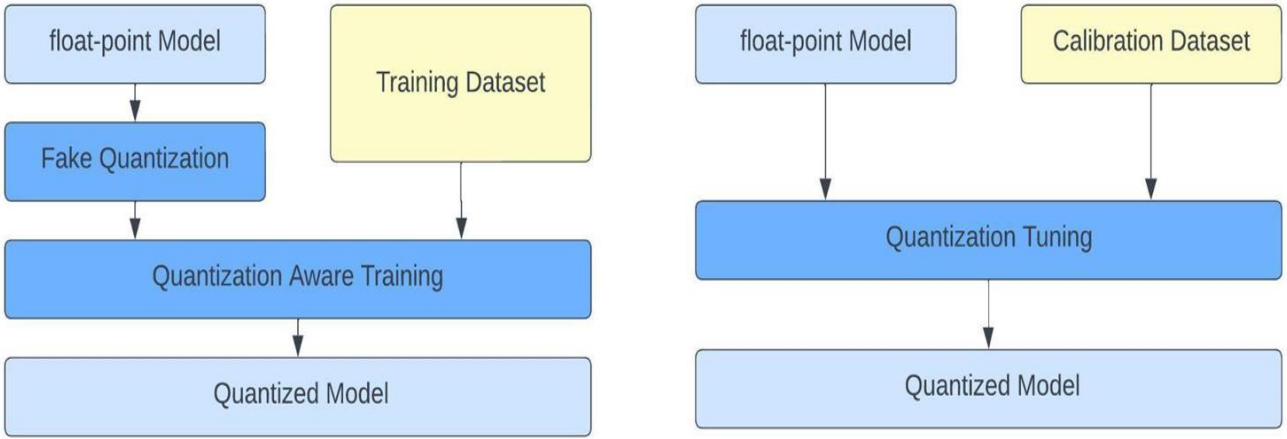
- Deep Neural Network (DNN) model
  - Weights: Symmetric per channel
  - Activation: Asymmetric per tensor



Histogram distribution of weights and activations [1]

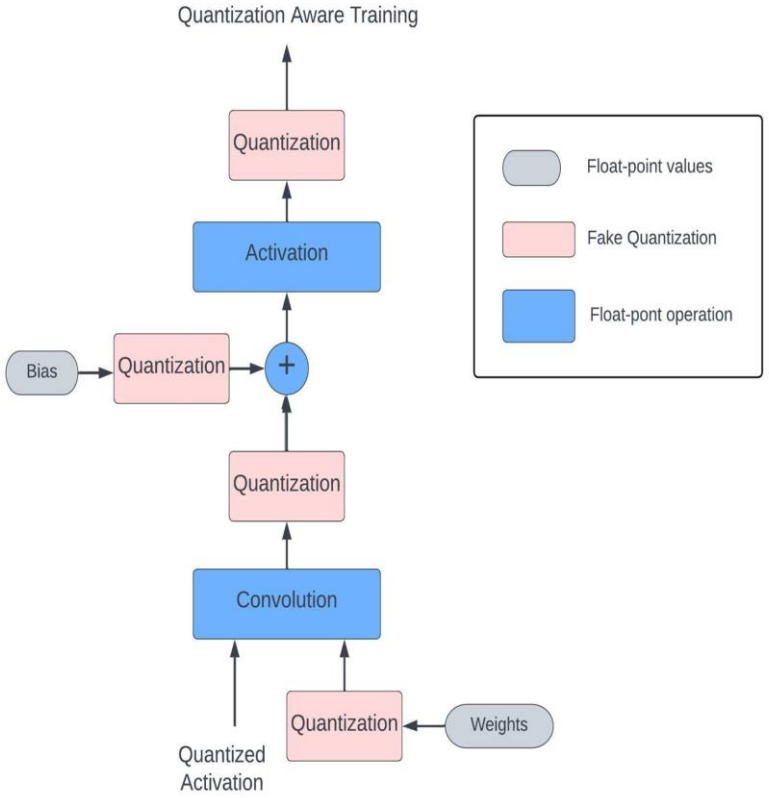
DNN Model Quantization

- DNN model quantization
  - Quantization Aware Training (QAT)
  - Post Training Quantization (PTQ)



Quantization Aware Training (QAT)

- Quantization Aware Training (QAT)
- Adds fake quantization nodes during training
- Pros:
  - Fine-tune trained float model
  - Improves quantized accuracy
- Cons:
  - Compute intensive process
  - Needs training dataset



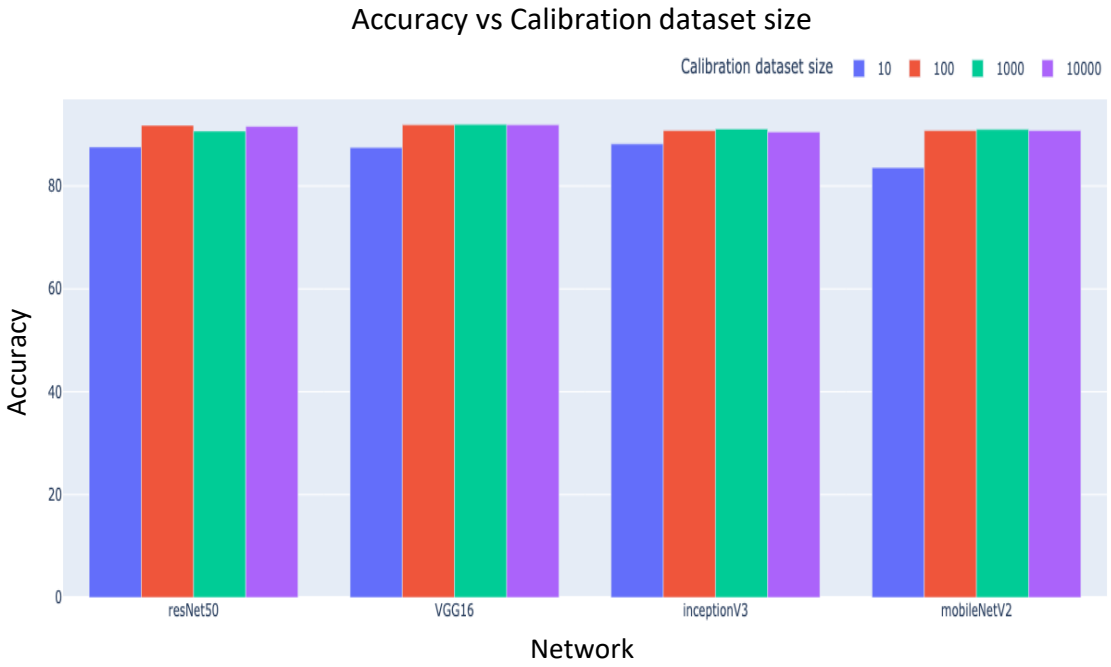
Post Training Quantization (PTQ)

- Post Training Quantization (PTQ)
- Analyze different quantization schemes
- Pros:
  - No model training
  - Limited calibration dataset
- Cons:
  - Degradation in accuracy

Network	Floating-point	Asymmetric per tensor	Asymmetric per channel
Mobilenet-v1 1 224	0.709	0.001	0.704
Mobilenet-v2 1 224	0.719	0.001	0.698
Nasnet-Mobile	0.74	0.722	0.74
Mobilenet-v2 1.4 224	0.749	0.004	0.74
Inception-v3	0.78	0.78	0.78
Resnet-v1 50	0.752	0.75	0.75
Resnet-v2 50	0.756	0.75	0.75
Resnet-v1 152	0.768	0.766	0.762
Resnet-v2 152	0.778	0.761	0.77

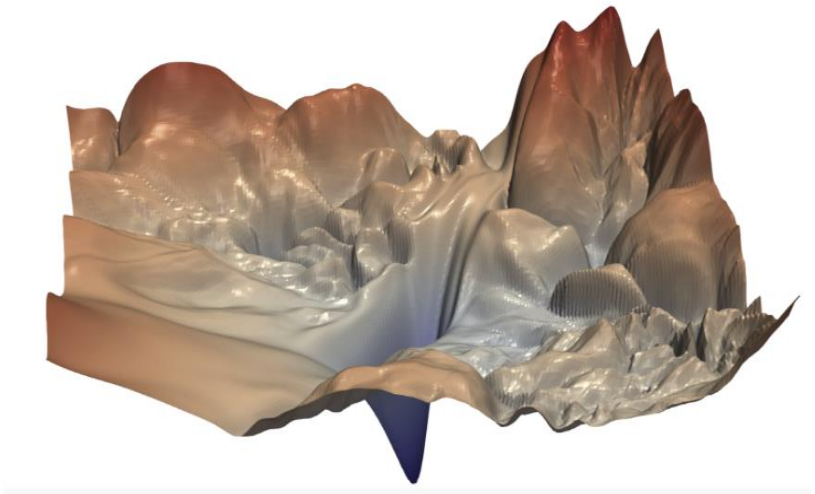
Calibration Dataset

- Calibration Dataset
  - Used to define quantization parameters
  - Representative dataset
  - Limited dataset ~100 to 1K images



Quantization Analysis

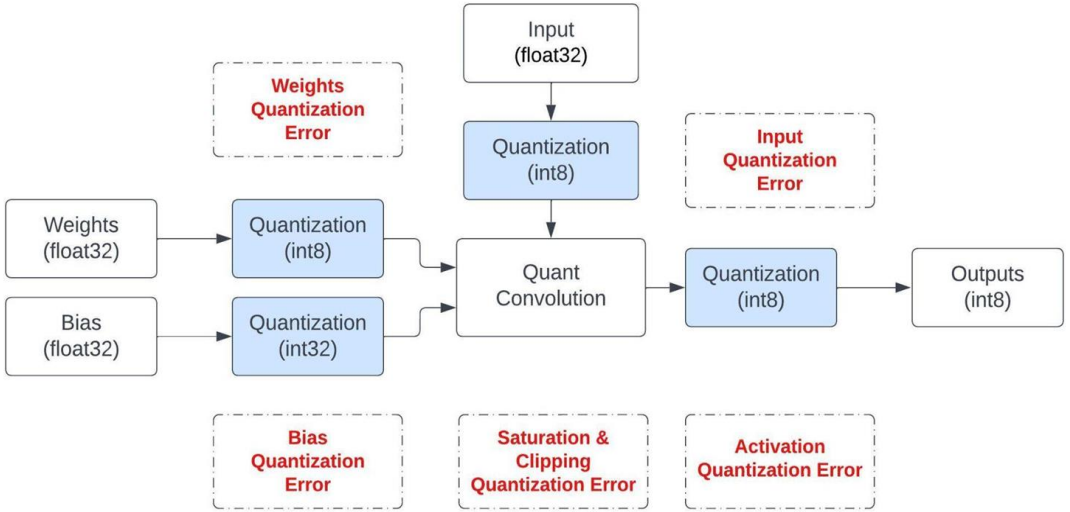
- Quantization introduces noise in the weights and activation
- Can lead to significant degradation in model accuracy
- Quantization analysis:
  - Quantization error
  - Visualization
  - Min/max tuning
  - Layer-wise analysis
  - Mixed precision
  - Weight equalization



Loss surface of ResNet-56 by Hao Li et al. [4]

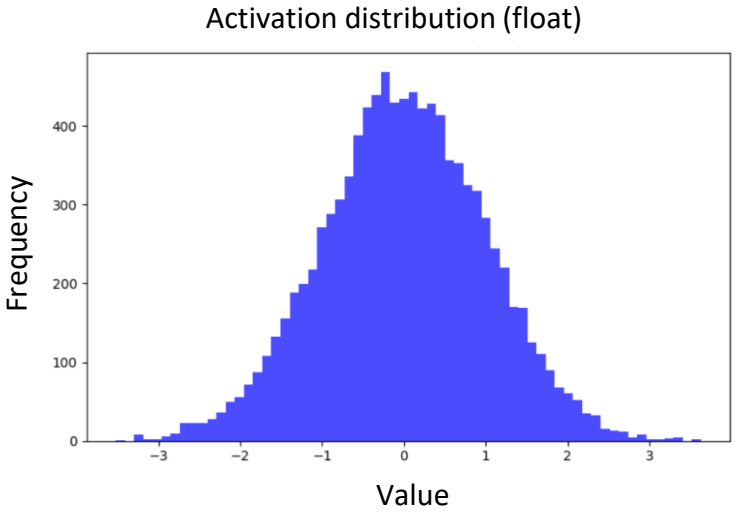
Quantization Error

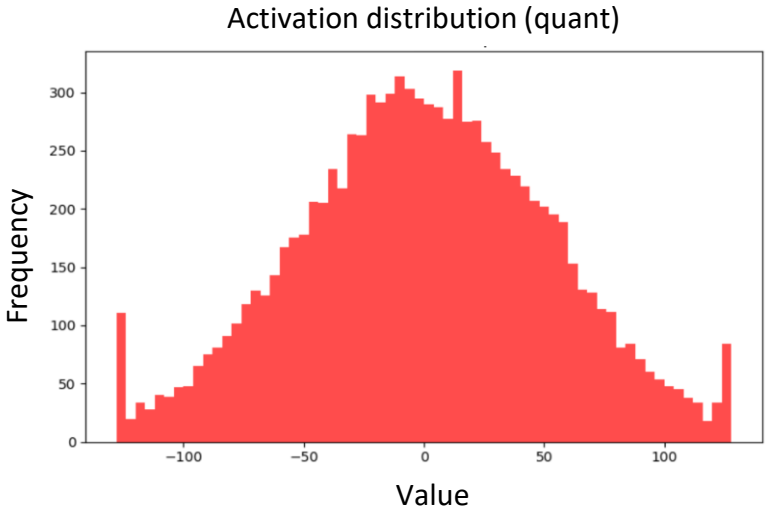
- Quantization error sources in convolution operation
  - Weight quantization error
  - Activation quantization error
  - Saturation and clipping
  - Bias quantization error



Visualization

- Visualization of the weights/activations
- Nature of the distribution
  - Multimodal distribution
  - Long tails in data distribution





Min/Max Tuning

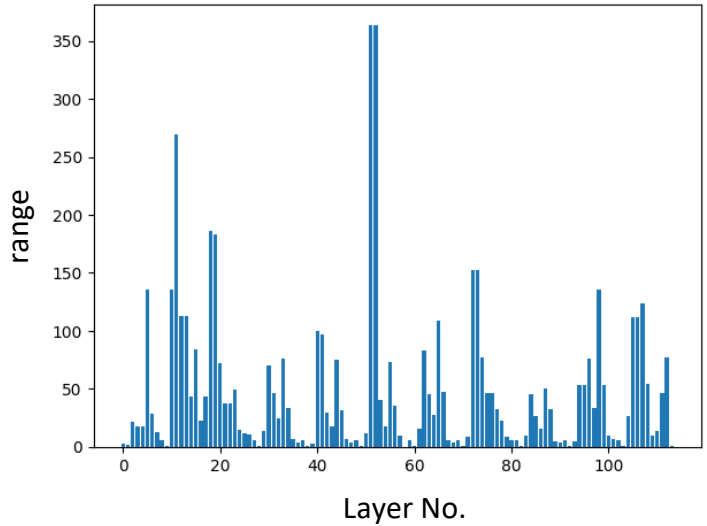
- Min/max tuning is used to eliminate outliers in weights and activations
  - Min/max: absolute min/max values
  - Percentile: histogram-based percentile to select quantization range
  - Entropy: minimize distribution entropy using KL divergence
  - MSE: Mean Square Error

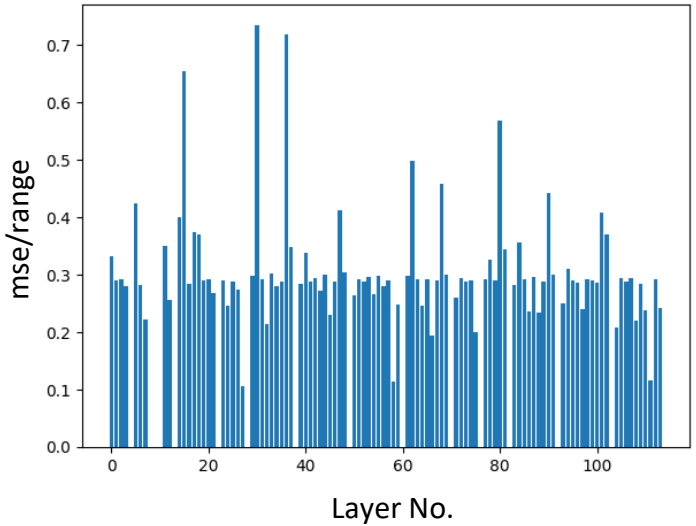
Model	Float (FP32)	Max	Percentile	Entropy	MSE
ResNet50	0.846	0.833	0.838	**0.840**	0.839
EfficietNetB0	0.831	0.831	0.832	**0.832**	0.832
MobileNetV3Small	0.816	0.531	0.582	**0.744**	0.577

Accuracy results for different min/max tuning methods on CIFAR100 dataset [5]

Layerwise Error

- Large quantization errors can be attributed to only a few problematic layers
- Identify the layers use visualization or min/max tuning techniques





Mixed Precision

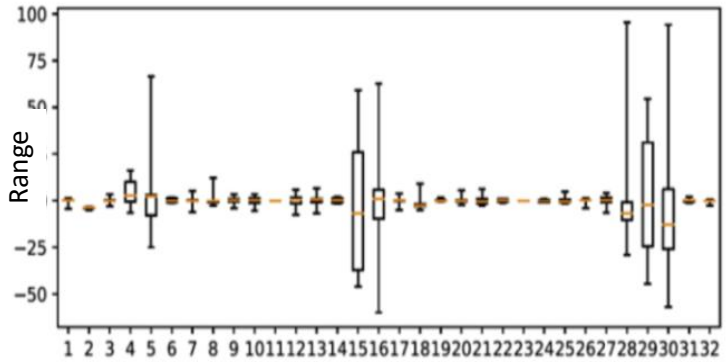
- Use different 8-bit/16-bit integers or FP8/FP16 for quantization
- Switch high quantization error layers to higher bit precision
- Reduce quantization overheads for light weight operations by running in float

Model	FP32 Accuracy	FP16 Quantization	INT8 Quantization	INT16 Activation	Mixed (FP32 + INT8) precision
ResNet50	0.8026	0.8028	0.8022	0.8021	0.8048
EfficietNetB2	0.8599	0.8593	0.8083	0.8578	0.8597
MobileNetV3Small	0.8365	0.8368	0.4526	0.7979	0.8347

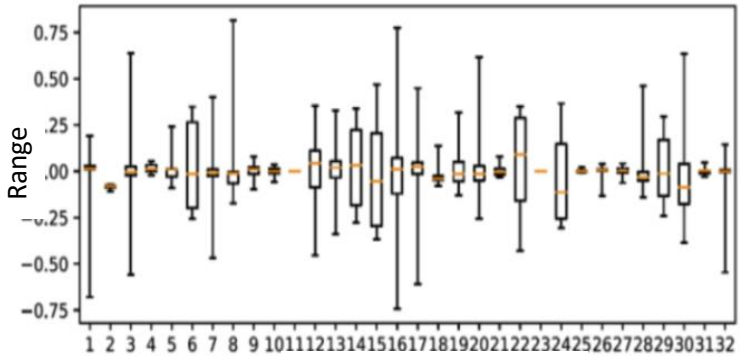
Evaluation of mixed precision accuracy on CIFAR10 dataset [5]

Quantization Analysis: Weight Equalization

- Reduce the variance of weight distribution across channels
- Adjust the scale factor across layers
- Enables use of simpler quantization schemes like per tensor instead of per channel



Pre-equalization box chart



Post-equalization box chart

Quantization: Best Practices

- ***Model selection***
  - Large models are more tolerant of quantization error.
  - NAS for efficient architecture for quantization

- ***Model quantization***
  - Post Training Quantization (PTQ) is favored for its efficiency
  - Quantization Aware Training (QAT) is resource-intensive but effective

- ***Calibration dataset***
  - Statistical data from around ~100-1K samples for quantization parameters

- ***Quantization tools***
  - Available tools for support of different quantization schemes
  - Limited quantization analysis capabilities

Quantization: Best Practices

- ***Quantization Scheme***
  - ***Weights:*** Symmetric-per-channel quantization
  - ***Activations:*** Asymmetric-per-tensor quantization

- ***Quantization Evaluation***
  - Evaluate model quantized accuracy across different quantization schemes

- ***Quantization Analysis***
  - Identify potentially problematic layers through layer-wise analysis
  - Degradation in accuracy could potentially be recovered through techniques like mixed precision, min/max tuning and weight equalization.

References

[1] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. arXiv preprint arXiv:1806.08342, 2018.

[2] From Theory to Practice: Quantizing Convolutional Neural Networks for Practical Deployment [Link]

[3] Quantization of Convolutional Neural Networks: Model Quantization [Link]

[4] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In Advances in Neural Information Processing Systems, pages 6389–6399, 2018.

[5] Quantization of Convolutional Neural Networks: Quantization Analysis [Link]

Thank you!