ICML 2022

Qualcomm AI research

Overcoming Oscillations in Quantization-Aware Training




Markus Nagel* (Staff Engineer/Manager), Marios Fournarakis* (Senior Engineer), Yelysei Bondarenko, Tijmen Blankevoort

Qualcomm AI Research
Qualcomm Technologies Netherlands B.V.

*equal contribution

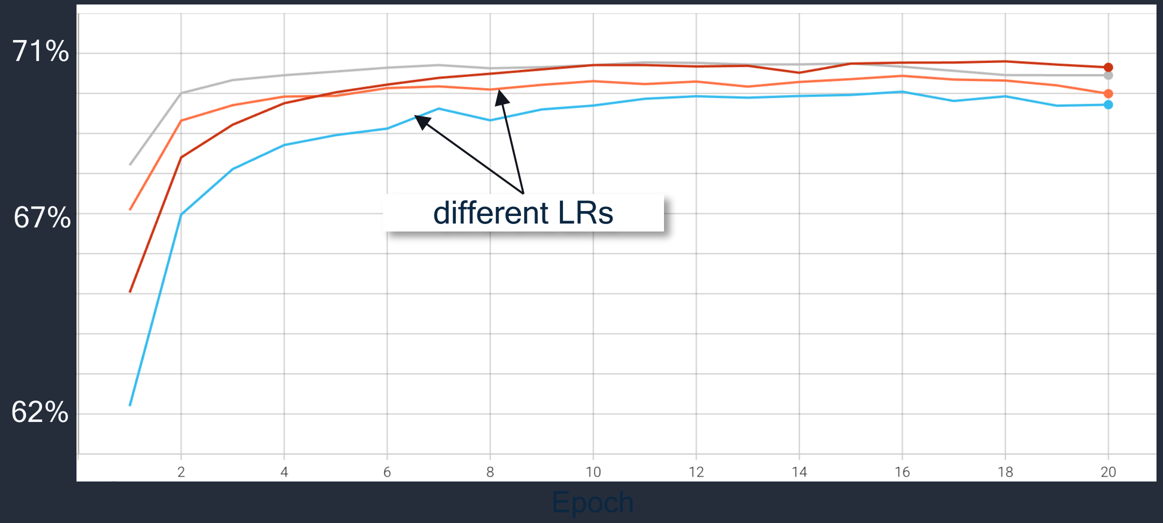
Motivation

- QAT for MobileNetV2 on ImageNet with 4-bit weights

Motivation

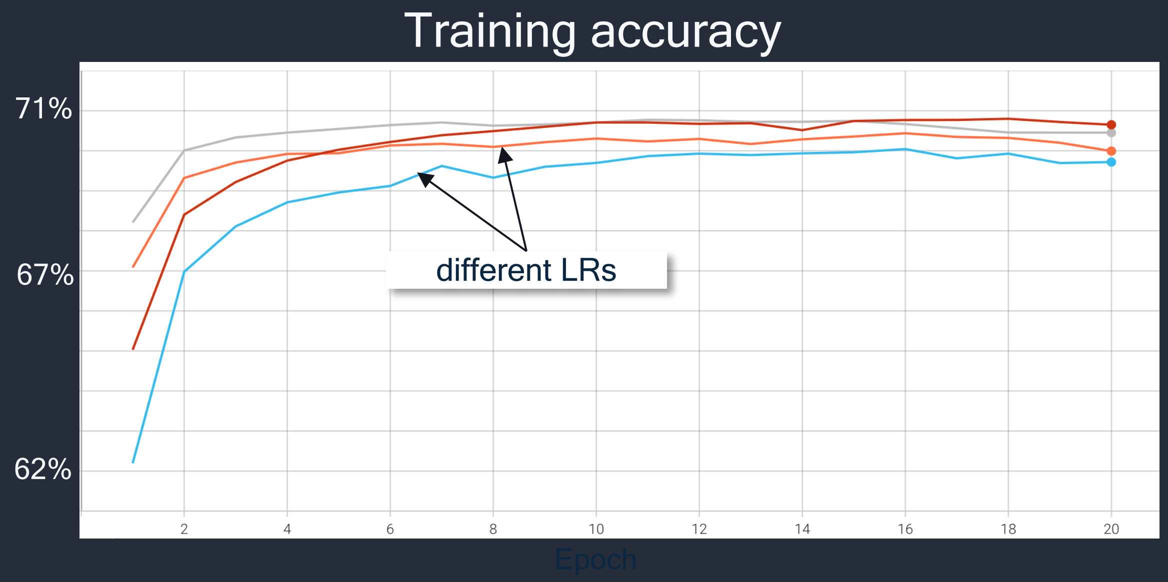
- QAT for MobileNetV2 on ImageNet with 4-bit weights

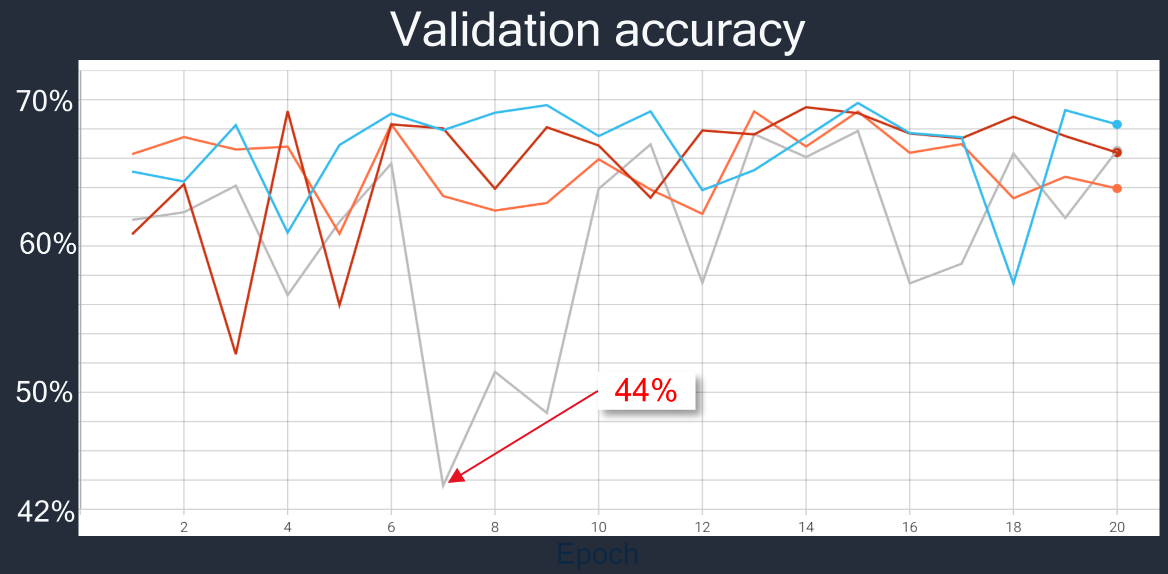
Training accuracy



Motivation

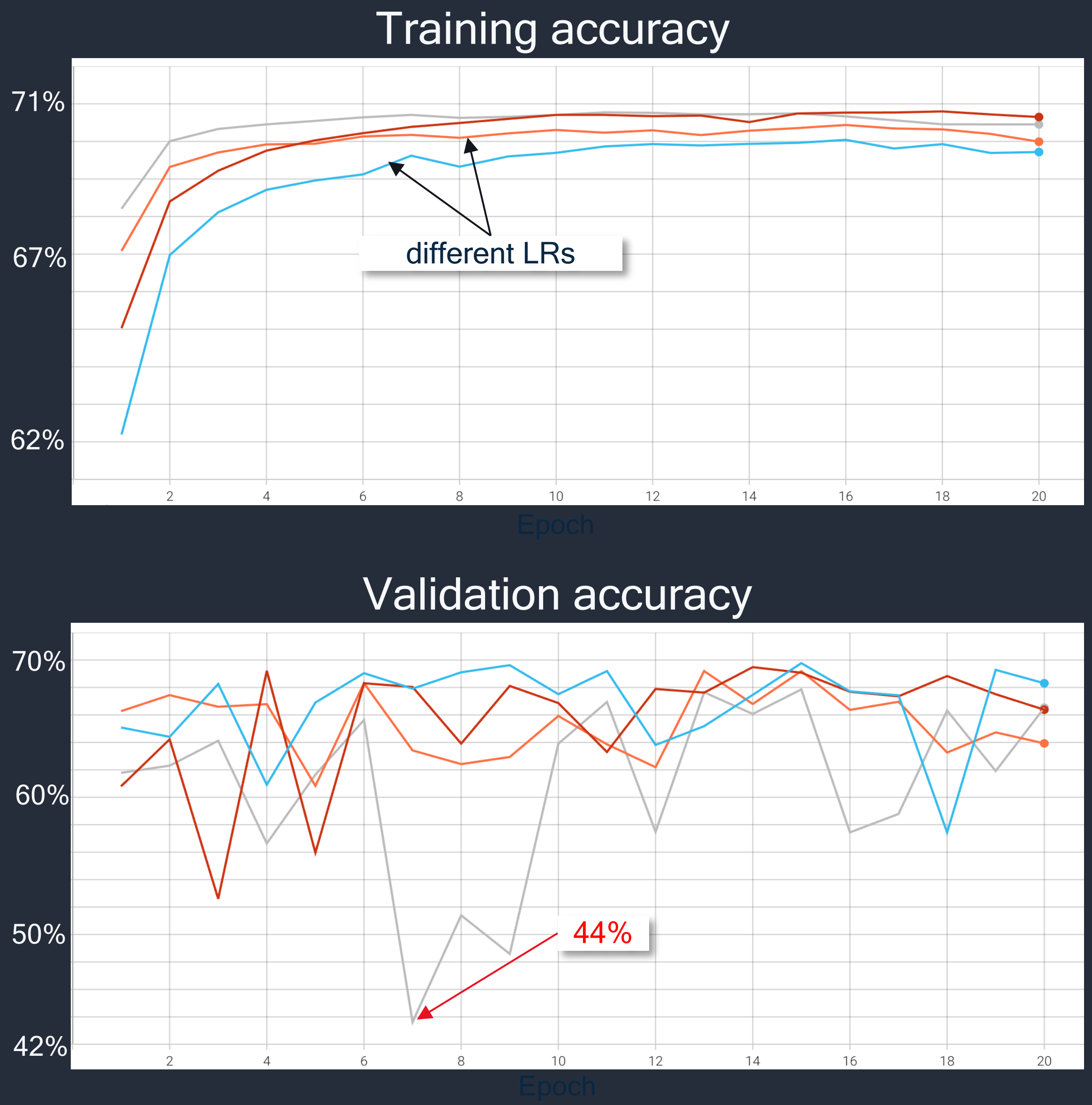
- QAT for MobileNetV2 on ImageNet with 4-bit weights
- Validation accuracy is unstable





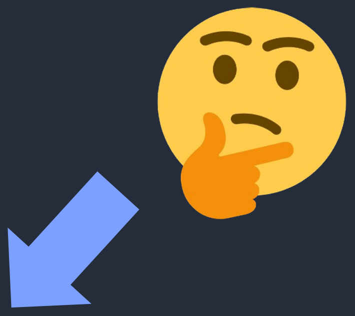
Motivation

- QAT for MobileNetV2 on ImageNet with 4-bit weights
- Validation accuracy is unstable



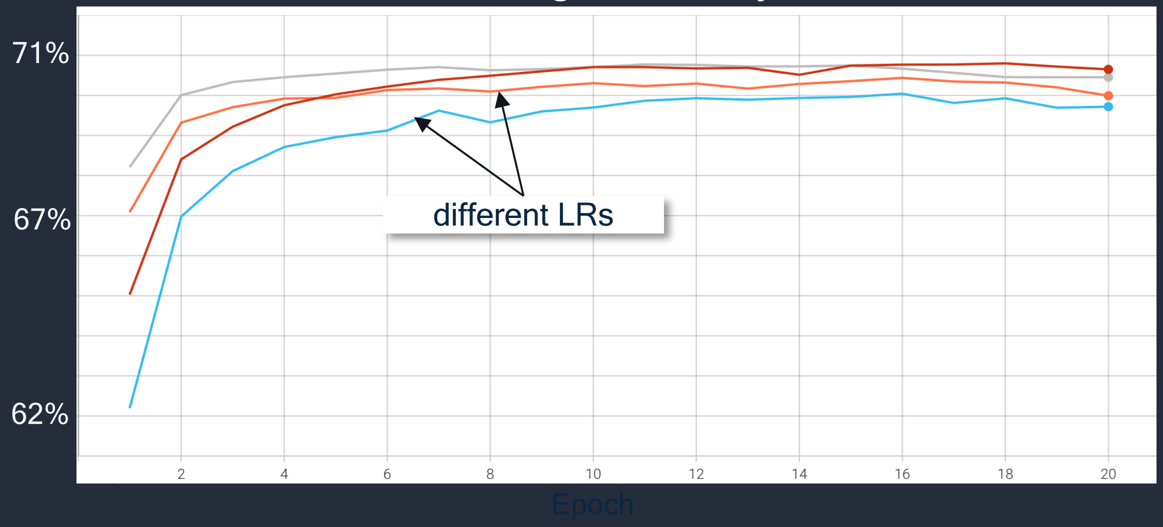
Motivation

- QAT for MobileNetV2 on ImageNet with 4-bit weights
- Validation accuracy is unstable

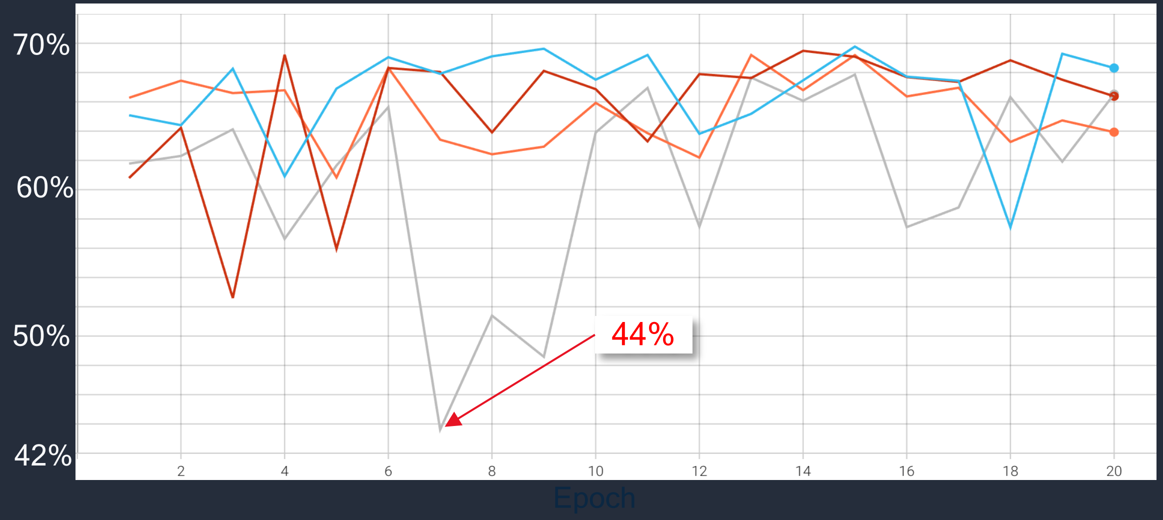


overfitting?

Training accuracy



Validation accuracy

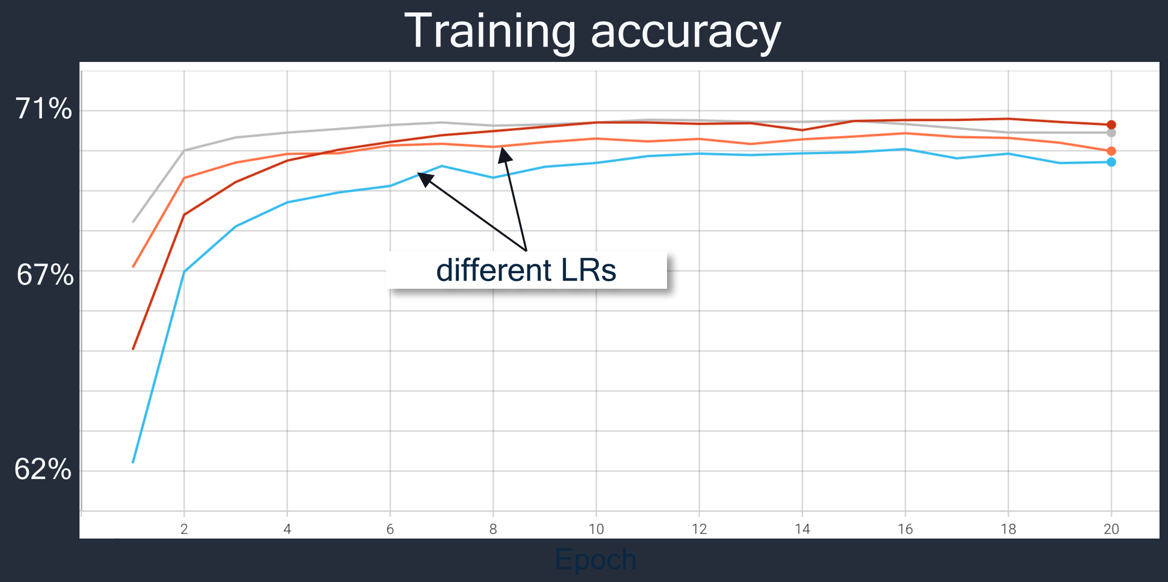


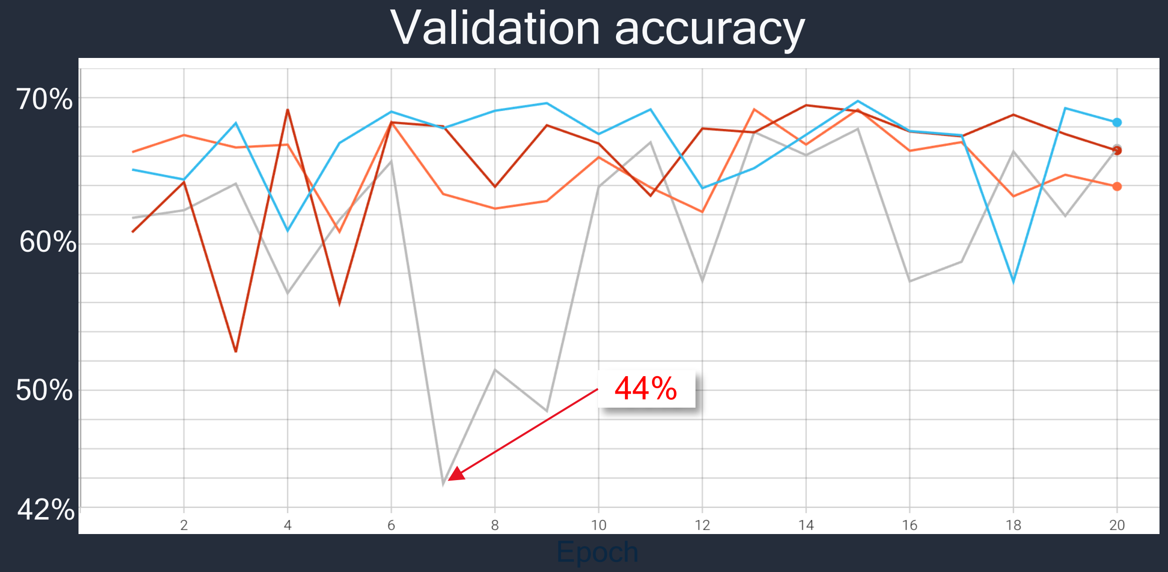
Motivation

- QAT for MobileNetV2 on ImageNet with 4-bit weights
- Validation accuracy is unstable

overfitting?

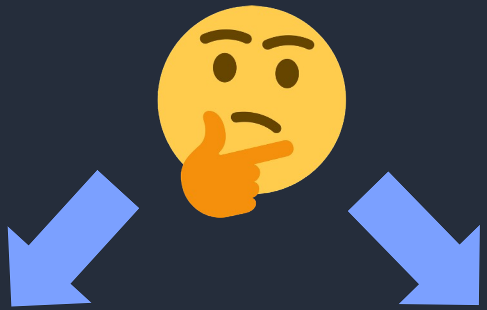
unlikely for ImageNet





Motivation

- QAT for MobileNetV2 on ImageNet with 4-bit weights
- Validation accuracy is unstable

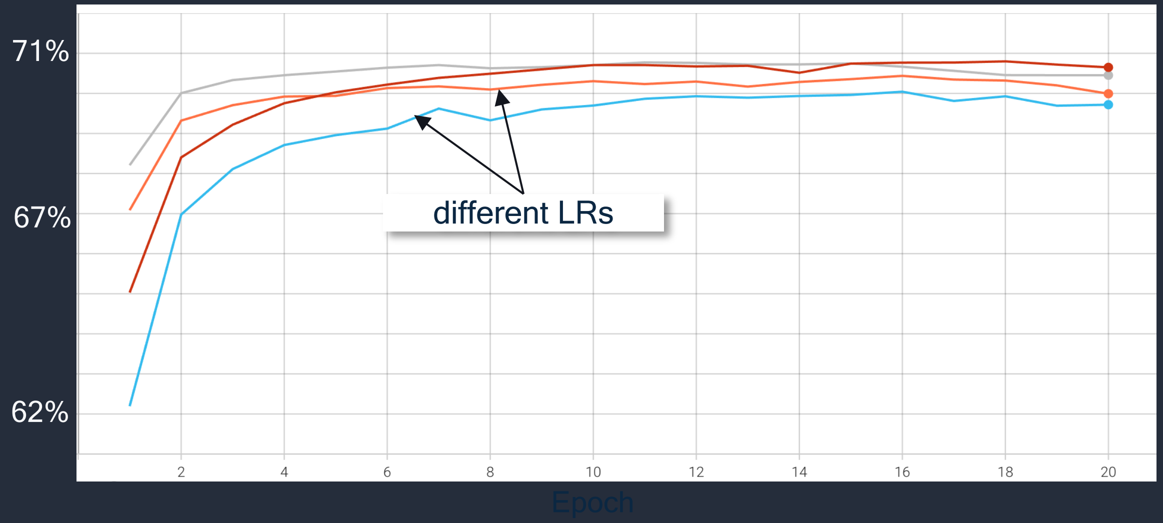


overfitting?

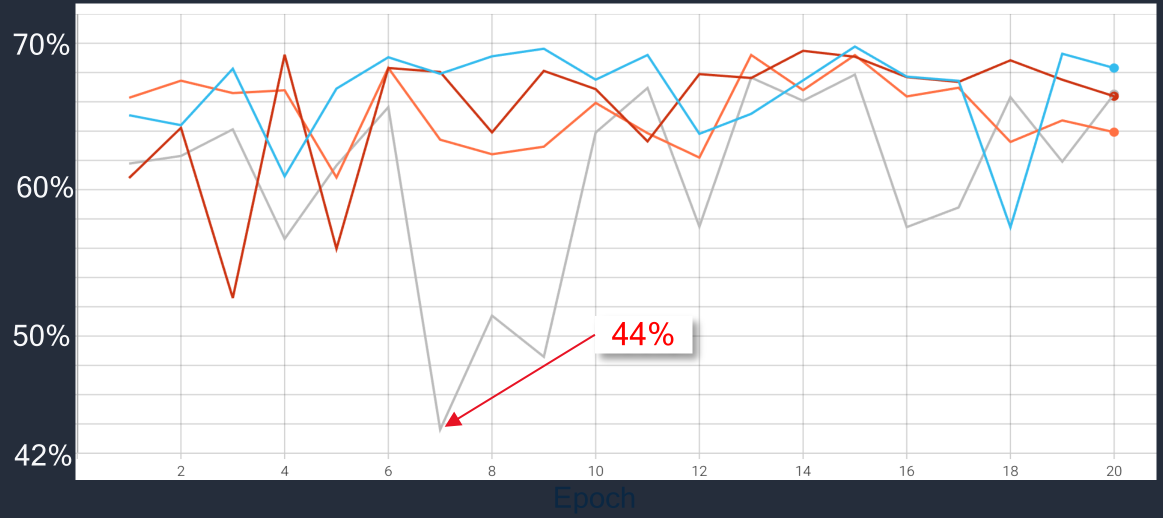
unlikely for ImageNet

other training/inference difference?

Training accuracy

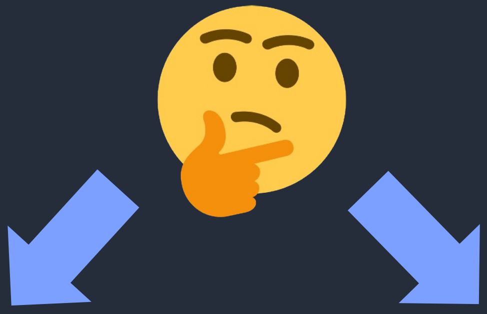


Validation accuracy



Motivation

- QAT for MobileNetV2 on ImageNet with 4-bit weights
- Validation accuracy is unstable



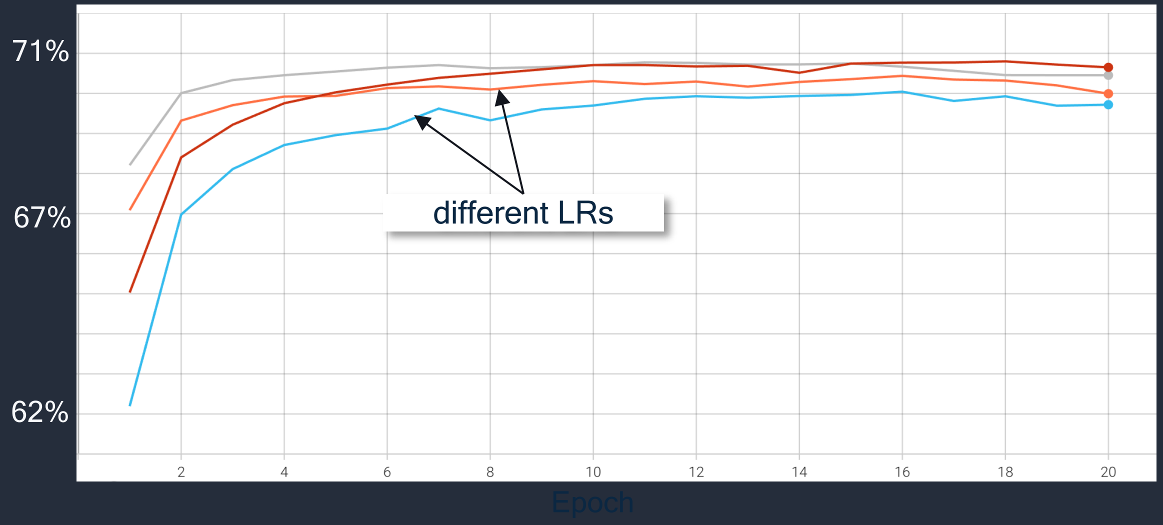
~~overfitting?~~

unlikely for ImageNet

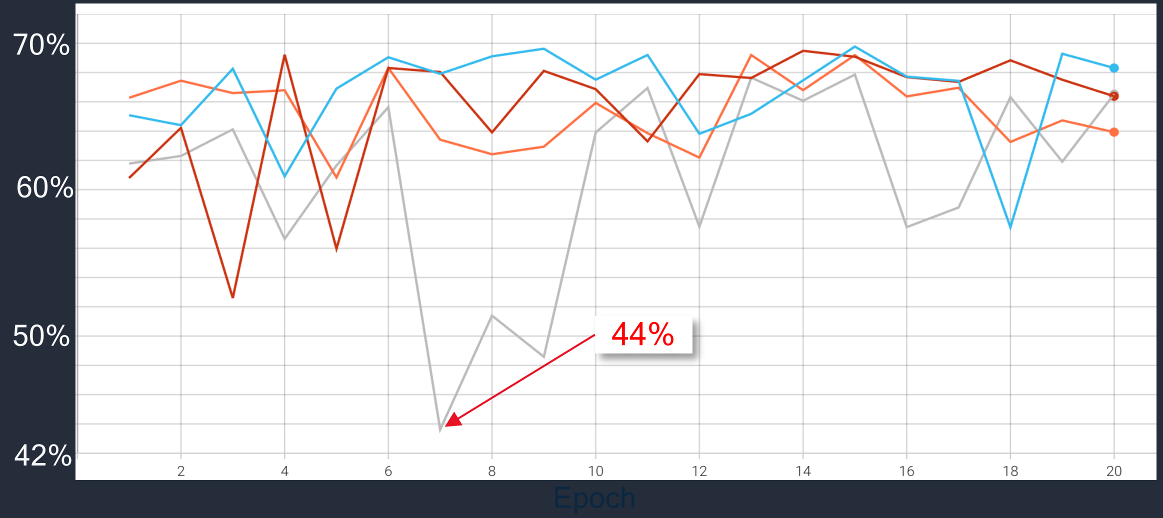
other training/inference difference?

Yes, batch-norm statistics!

Training accuracy



Validation accuracy



What happens with batch-norm?

- During training, batch statistics are used
- During inference, statistics are approximated with an exponential moving average (EMA) from training

What happens with batch-norm?

- During training, batch statistics are used
- During inference, statistics are approximated with an exponential moving average (EMA) from training
- EMA statistics are significantly different to true statistics for certain layers

Network	Layer	$\max D_{\mathrm{KL}}$	$\mathbb{E}[D_{\mathrm{KL}}]$
ResNet18	layer1.0.conv1	0.0059	0.0002
ResNet18	layer1.0.conv2	0.0130	0.0014
ResNet18	layer3.0.conv1	0.0006	0.0001
MobileNetV2	Conv3.0 (PW)	0.7858	0.0292
MobileNetV2	Conv3.1 (DW)	55.3782	1.25464
MobileNetV2	Conv3.2 (PW)	0.0065	0.0012
MobileNetV2	Conv10.0 (PW)	0.0037	0.0004
MobileNetV2	Conv10.1 (DW)	27.2618	0.2900
MobileNetV2	Conv10.2 (PW)	0.0267	0.0034

KL divergence between EMA and true sample statistics of the training dataset. $\max D_{KL}$: maximum per-channel, $\mathbb{E}[D_{KL}]$: average over channels

What happens with batch-norm?

- During training, batch statistics are used
- During inference, statistics are approximated with an exponential moving average (EMA) from training
- EMA statistics are significantly different to true statistics for certain layers
- Depth-wise separable layers more prone to this discrepancy

Network	Layer	$\max D_{\mathrm{KL}}$	$\mathbb{E}\left[D_{\mathrm{KL}}\right]$
ResNet18	layer1.0.conv1	0.0059	0.0002
ResNet18	layer1.0.conv2	0.0130	0.0014
ResNet18	layer3.0.conv1	0.0006	0.0001
MobileNetV2	Conv3.0 (PW)	0.7858	0.0292
MobileNetV2	Conv3.1 (DW)	55.3782	1.25464
MobileNetV2	Conv3.2 (PW)	0.0065	0.0012
MobileNetV2	Conv10.0 (PW)	0.0037	0.0004
MobileNetV2	Conv10.1 (DW)	27.2618	0.2900
MobileNetV2	Conv10.2 (PW)	0.0267	0.0034

KL divergence between EMA and true sample statistics of the training dataset. $\max D_{KL}$: maximum per-channel, $\mathbb{E}[D_{KL}]$: average over channels

What happens with batch-norm?

- During training, batch statistics are used
- During inference, statistics are approximated with an exponential moving average (EMA) from training
- EMA statistics are significantly different to true statistics for certain layers
- Depth-wise separable layers more prone to this discrepancy
- Full convolutions are less affected by this

Network	Layer	$\max D_{\mathrm{KL}}$	$\mathbb{E}[D_{\mathrm{KL}}]$
ResNet18	layer1.0.conv1	0.0059	0.0002
ResNet18	layer1.0.conv2	0.0130	0.0014
ResNet18	layer3.0.conv1	0.0006	0.0001
MobileNetV2	Conv3.0 (PW)	0.7858	0.0292
MobileNetV2	Conv3.1 (DW)	55.3782	1.25464
MobileNetV2	Conv3.2 (PW)	0.0065	0.0012
MobileNetV2	Conv10.0 (PW)	0.0037	0.0004
MobileNetV2	Conv10.1 (DW)	27.2618	0.2900
MobileNetV2	Conv10.2 (PW)	0.0267	0.0034

KL divergence between EMA and true sample statistics of the training dataset. $\max D_{KL}$: maximum per-channel, $\mathbb{E}[D_{KL}]$: average over channels

Batch normalization re-estimation

- We can estimate correct BN statistics after QAT using training data:
  - Common practice in stochastic quantization[1, 2]

[1] Peters and Welling, Probabilistic binary neural networks. 2018.
[2] Louizos et al., Relaxed quantization for discretized neural networks. ICLR 2019.

Batch normalization re-estimation

- We can estimate correct BN statistics after QAT using training data:
  - Common practice in stochastic quantization[1, 2]

Network	Bits	pre-BN	post-BN
ResNet18	4	$70.15^{0.03}$	$70.20^{0.02}$
ResNet18	3	$69.63^{0.01}$	$69.70^{0.05}$
MobileNetV2	8	$71.79^{0.07}$	$71.89^{0.05}$
MobileNetV2	4	$68.99^{0.44}$	$71.01^{0.05}$
MobileNetV2	3	$64.97^{1.23}$	$69.50^{0.04}$

[1] Peters and Welling, Probabilistic binary neural networks. 2018.
[2] Louizos et al., Relaxed quantization for discretized neural networks. ICLR 2019.

Batch normalization re-estimation

- We can estimate correct BN statistics after QAT using training data:
  - Common practice in stochastic quantization[1, 2]
- BN re-estimation significantly improves validation accuracy for MobiletNetV2

Network	Bits	pre-BN	post-BN	
ResNet18	4	$70.15^{0.03}$	$70.20^{0.02}$	
ResNet18	3	$69.63^{0.01}$	$69.70^{0.05}$	
MobileNetV2	8	$71.79^{0.07}$	$71.89^{0.05}$	
MobileNetV2	4	$68.99^{0.44}$	$71.01^{0.05}$	+2.02
MobileNetV2	3	$64.97^{1.23}$	$69.50^{0.04}$	+4.53

[1] Peters and Welling, Probabilistic binary neural networks. 2018.
[2] Louizos et al., Relaxed quantization for discretized neural networks. ICLR 2019.

Batch normalization re-estimation

- We can estimate correct BN statistics after QAT using training data:
  - Common practice in stochastic quantization[1, 2]
- BN re-estimation significantly improves validation accuracy for MobiletNetV2
- BN re-estimation reduces variance between seeds
- Negligible effect on ResNet18

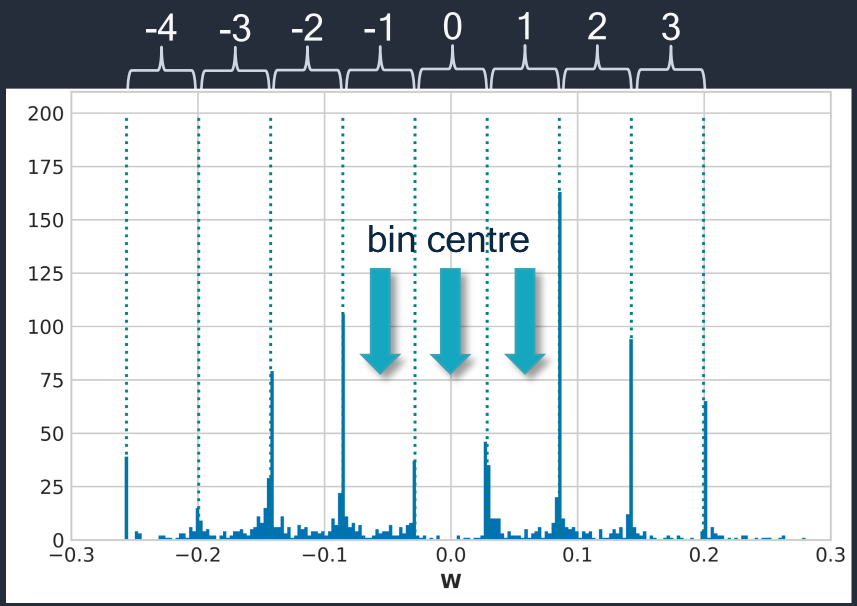
Network	Bits	pre-BN	post-BN	
ResNet18	4	$70.15^{0.03}$	$70.20^{0.02}$	
ResNet18	3	$69.63^{0.01}$	$69.70^{0.05}$	
MobileNetV2	8	$71.79^{0.07}$	$71.89^{0.05}$	
MobileNetV2	4	$68.99^{0.44}$	$71.01^{0.05}$	+2.02
MobileNetV2	3	$64.97^{1.23}$	$69.50^{0.04}$	+4.53

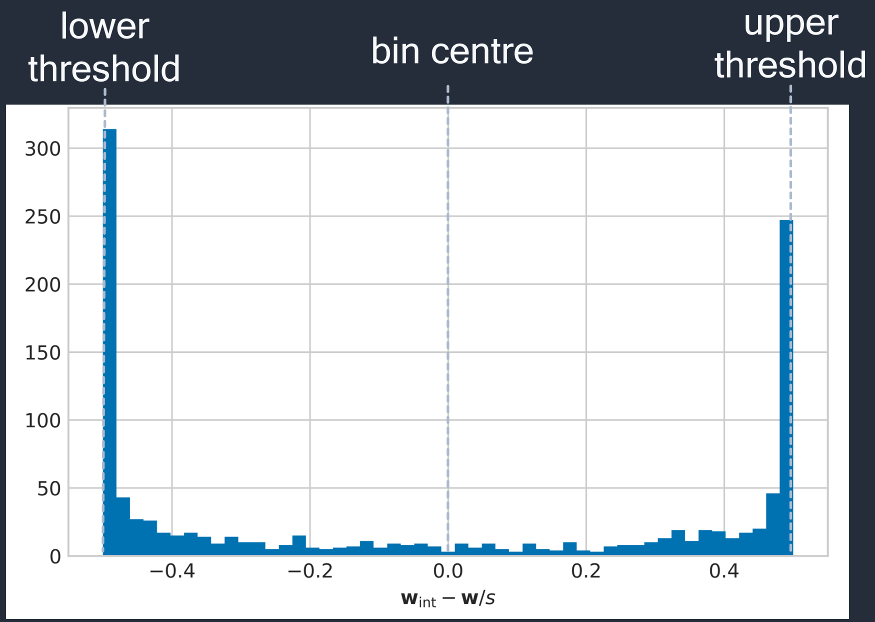
[1] Peters and Welling, Probabilistic binary neural networks. 2018.
[2] Louizos et al., Relaxed quantization for discretized neural networks. ICLR 2019.

Why are the BN statistics wrongly estimated for certain models?

Many weights are between two integer bins

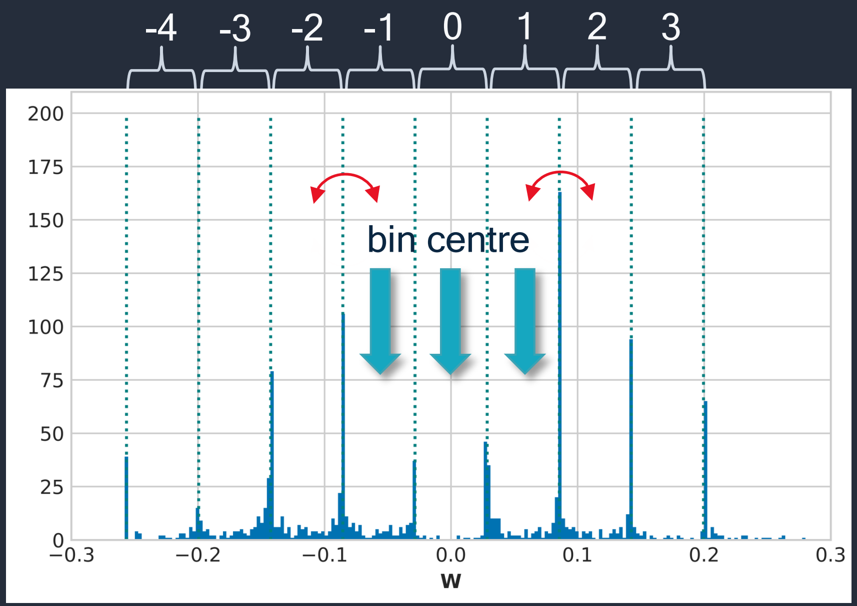
Latent weights histogram of depth-wise separable convolution from MobilnetNetV2

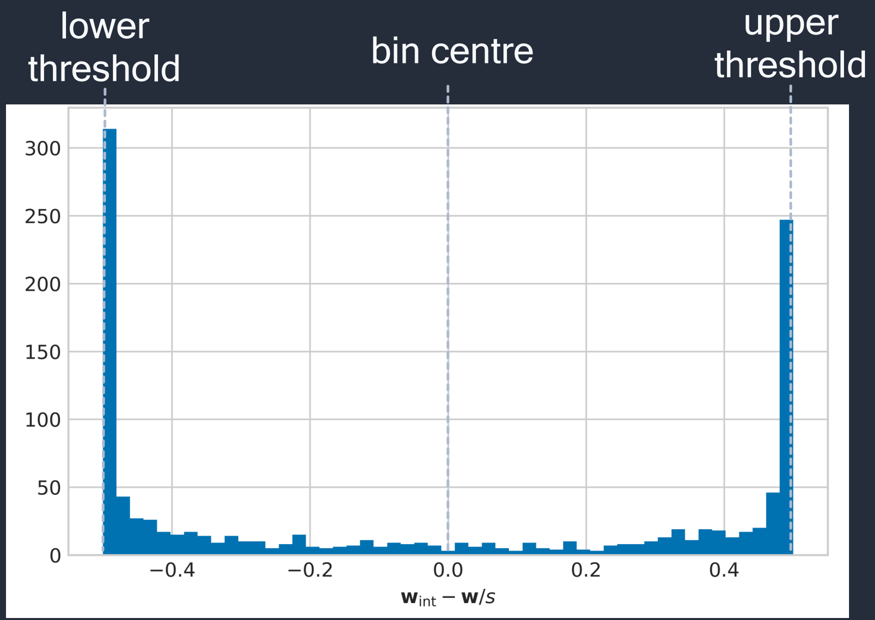




Many weights are between two integer bins

Latent weights histogram of depth-wise separable convolution from MobilnetNetV2

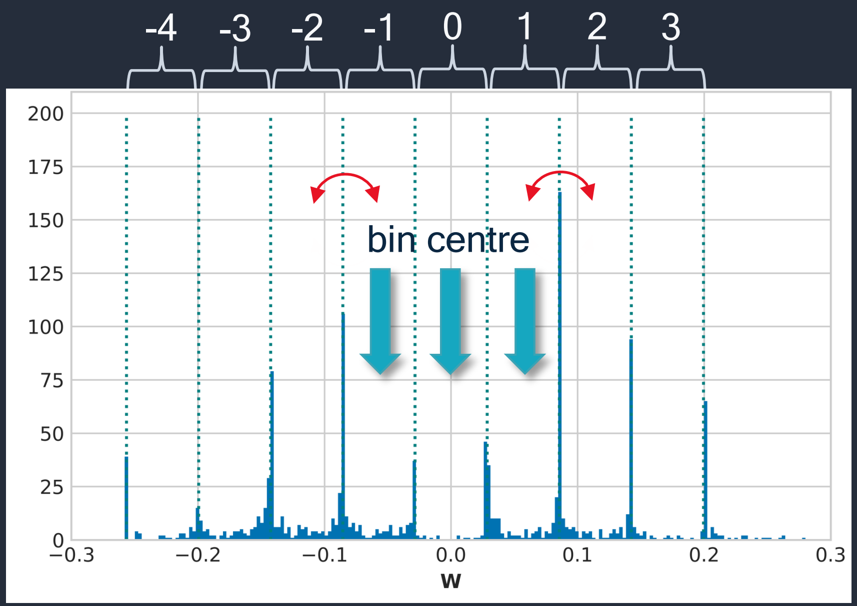


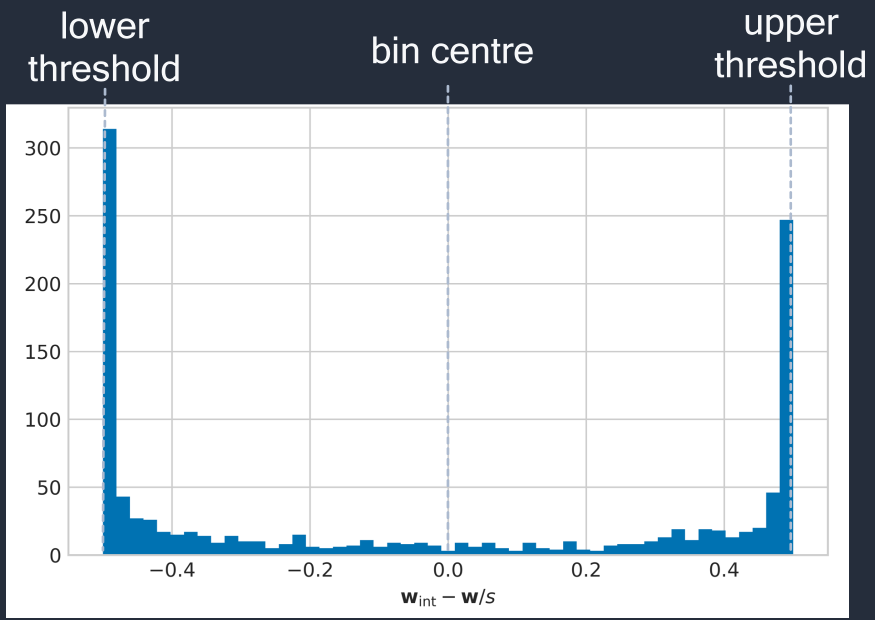


- Many weights at threshold → high chance weights change integer assignment

Many weights are between two integer bins

Latent weights histogram of depth-wise separable convolution from MobilnetNetV2

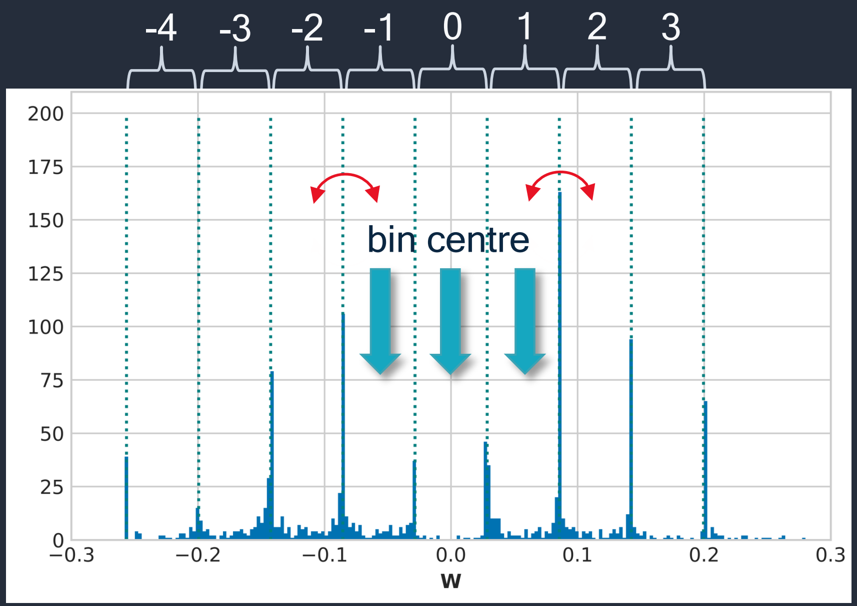


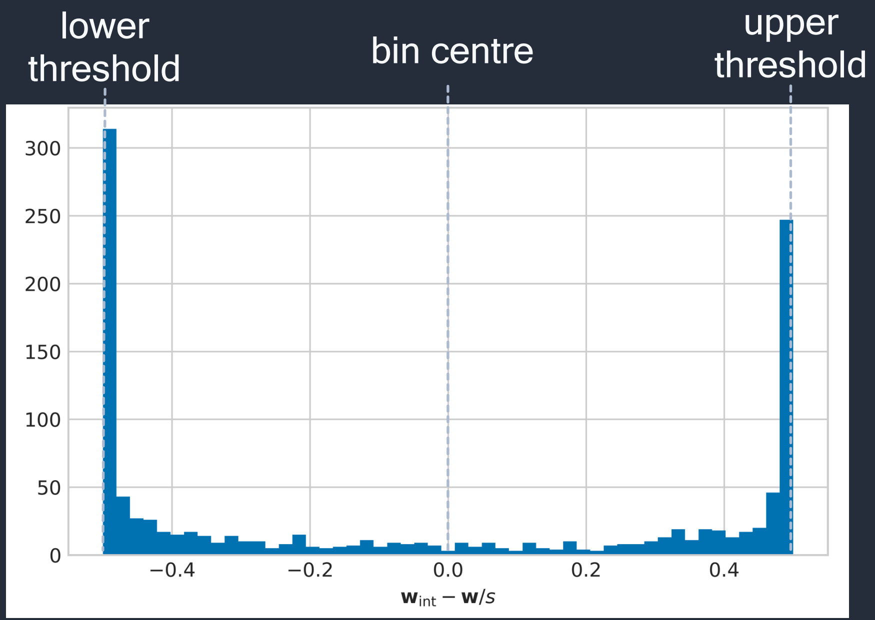


- Many weights at threshold → high chance weights change integer assignment
- Possible large change in output distribution (BN statistics):

Many weights are between two integer bins

Latent weights histogram of depth-wise separable convolution from MobilnetNetV2

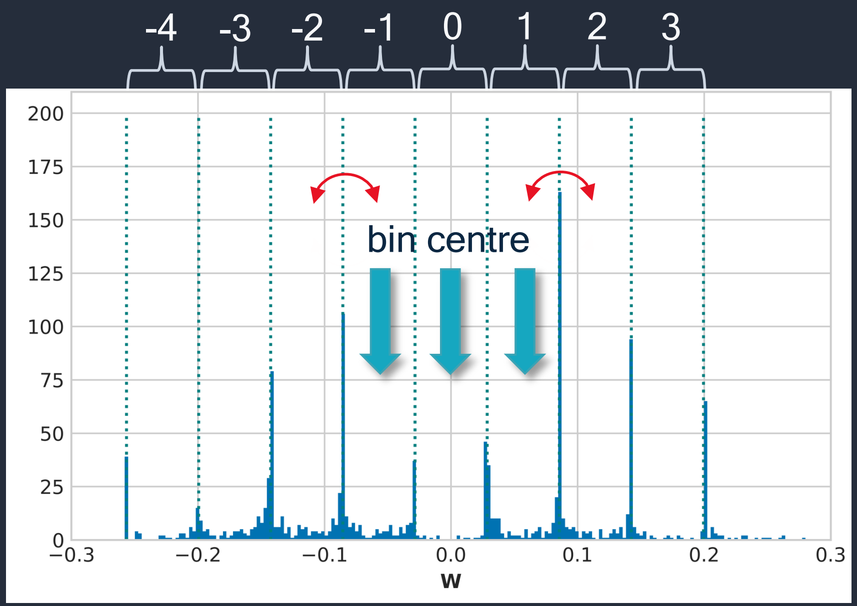


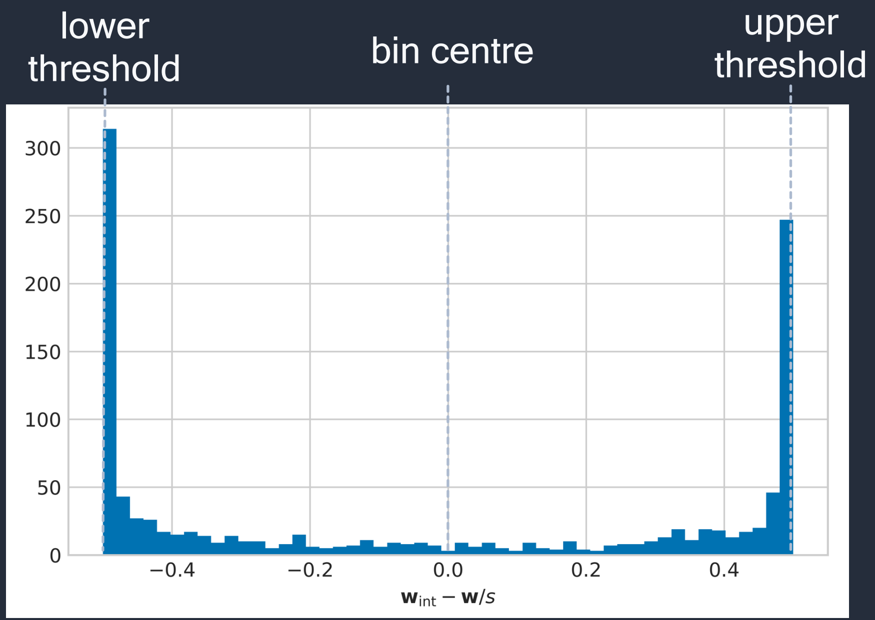


- Many weights at threshold → high chance weights change integer assignment
- Possible large change in output distribution (BN statistics):
  - Low bit quantization → larger change in quantized value $(\sim 1/b^2)$

Many weights are between two integer bins

Latent weights histogram of depth-wise separable convolution from MobilnetNetV2





- Many weights at threshold → high chance weights change integer assignment
- Possible large change in output distribution (BN statistics):
  - Low bit quantization → larger change in quantized value ($\sim 1/b^2$)
  - Layers with fewer weights (e.g. DS convs) → larger contribution of individual weights

What causes the latent weights to be close to the threshold?

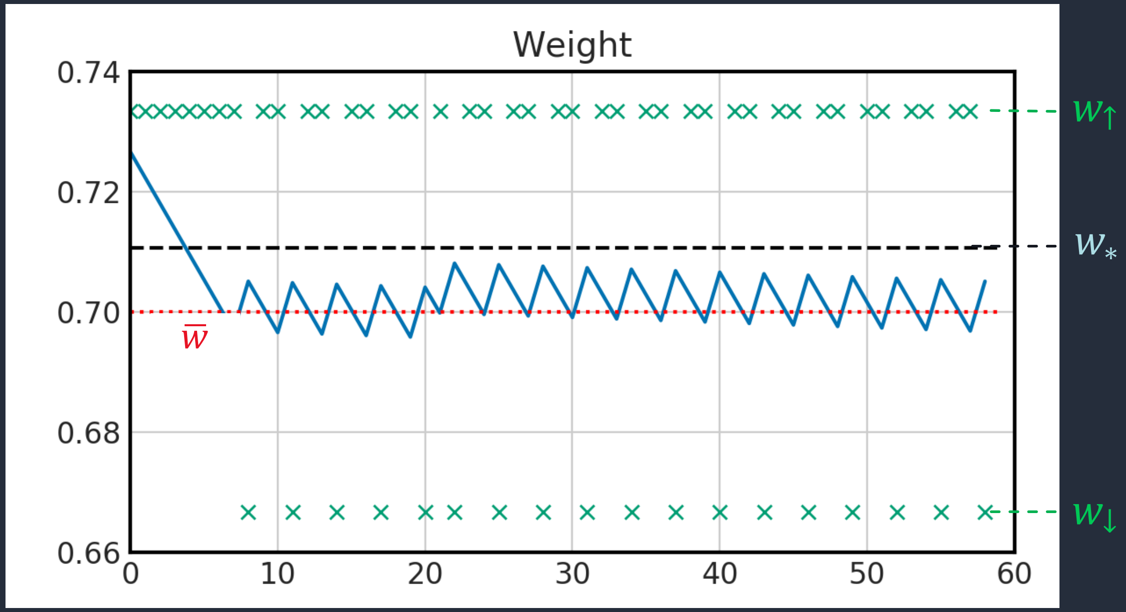
Oscillating weights in QAT

- Example regression problem:
  - Latent weight: w
  - Quantized weight: $q(w) = \mathrm{s}_w \cdot \mathrm{round}(w/s_w)$
  - Objective: $\min_w \mathcal{L}(w) = (w_* - q(w))^2$

[3] Bengio et al., Estimating or propagating gradients through stochastic neurons for conditional computation. 2013.

Oscillating weights in QAT

- Example regression problem:
  - Latent weight: w
  - Quantized weight: $q(w) = s_w \cdot \text{round}(w/s_w)$
  - Objective: $\min\limits_{w} \mathcal{L}(w) = (w_* - q(w))^2$



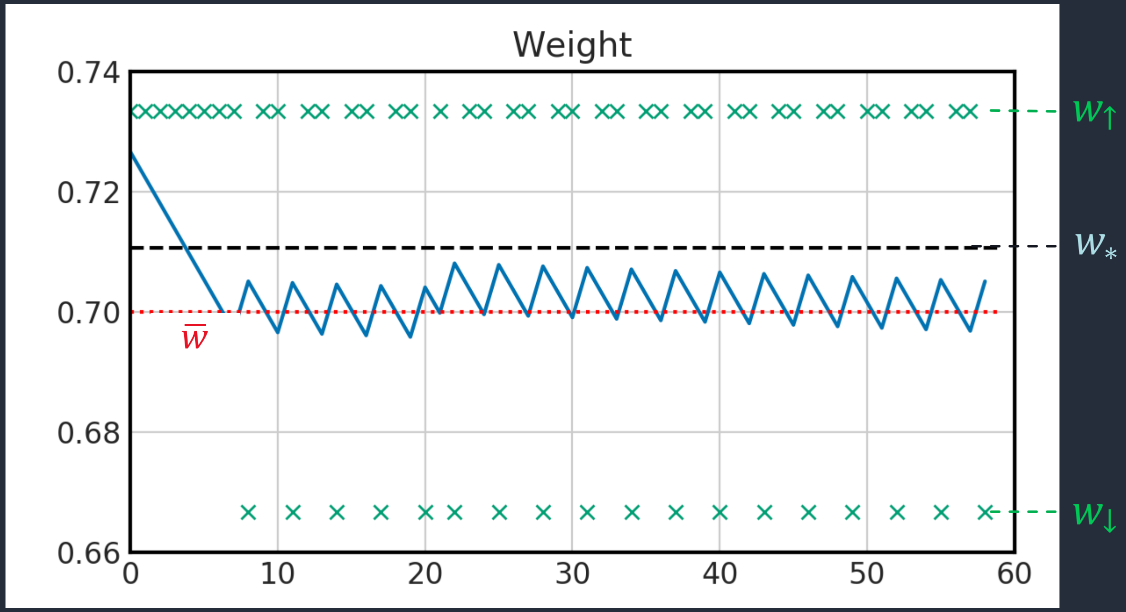
[3] Bengio et al., Estimating or propagating gradients through stochastic neurons for conditional computation. 2013.

Oscillating weights in QAT

- Example regression problem:
  - Latent weight: w
  - Quantized weight: $q(w) = s_w \cdot \mathrm{round}(w/s_w)$
  - Objective: $\min_w \mathcal{L}(w) = (w_* - q(w))^2$

- Rounding is approximated by STE[3]:

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial q(w)} = \begin{cases} w_* - w_\uparrow, & \text{if } w \geq \overline{w} \\ w_* - w_\downarrow, & \text{if } w < \overline{w} \end{cases}$$



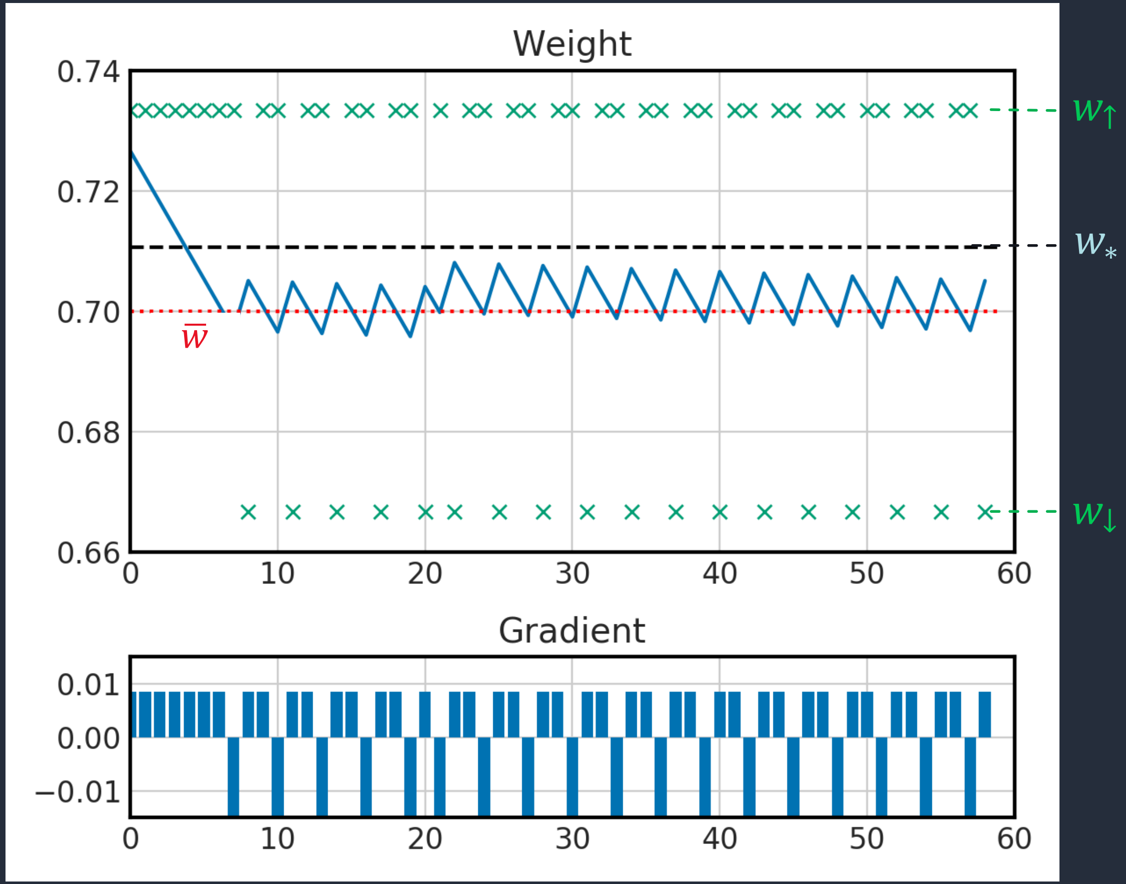
[3] Bengio et al., Estimating or propagating gradients through stochastic neurons for conditional computation. 2013.

Oscillating weights in QAT

- Example regression problem:
  - Latent weight: w
  - Quantized weight: $q(w) = s_w \cdot \text{round}(w/s_w)$
  - Objective: $\min_w \mathcal{L}(w) = (w_* - q(w))^2$

- Rounding is approximated by STE[3]:

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}}{\partial q(w)} = \begin{cases} w_* - w_\uparrow, & \text{if } w \geq \overline{w} \\ w_* - w_\downarrow, & \text{if } w < \overline{w} \end{cases}$$

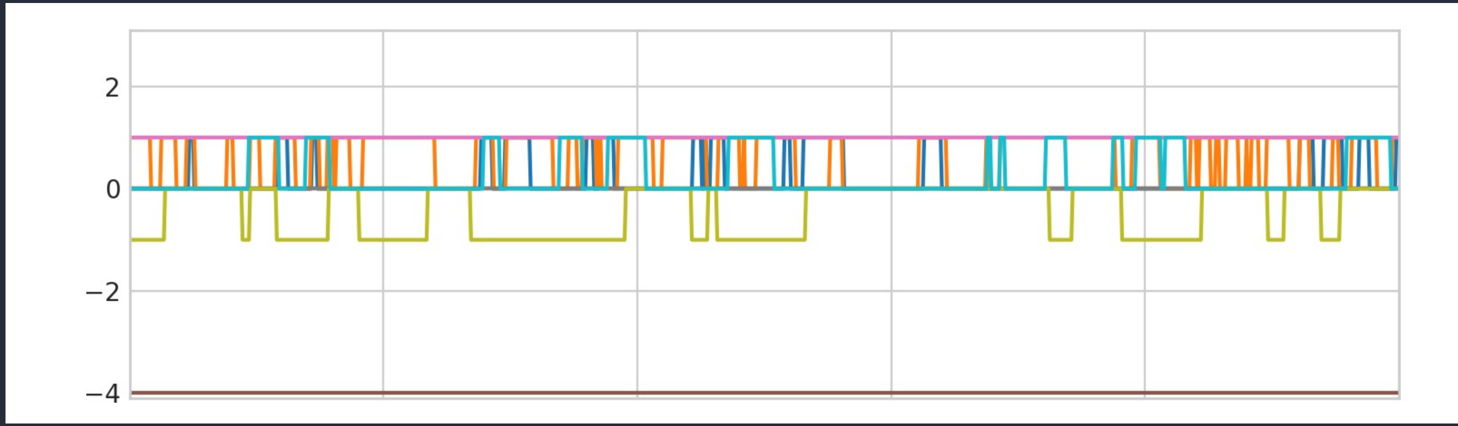


[3] Bengio et al., Estimating or propagating gradients through stochastic neurons for conditional computation. 2013.

Oscillations in practice

Example of MobileNetV2 training (last 1000 iterations)

Quantized weights, $q(w)$

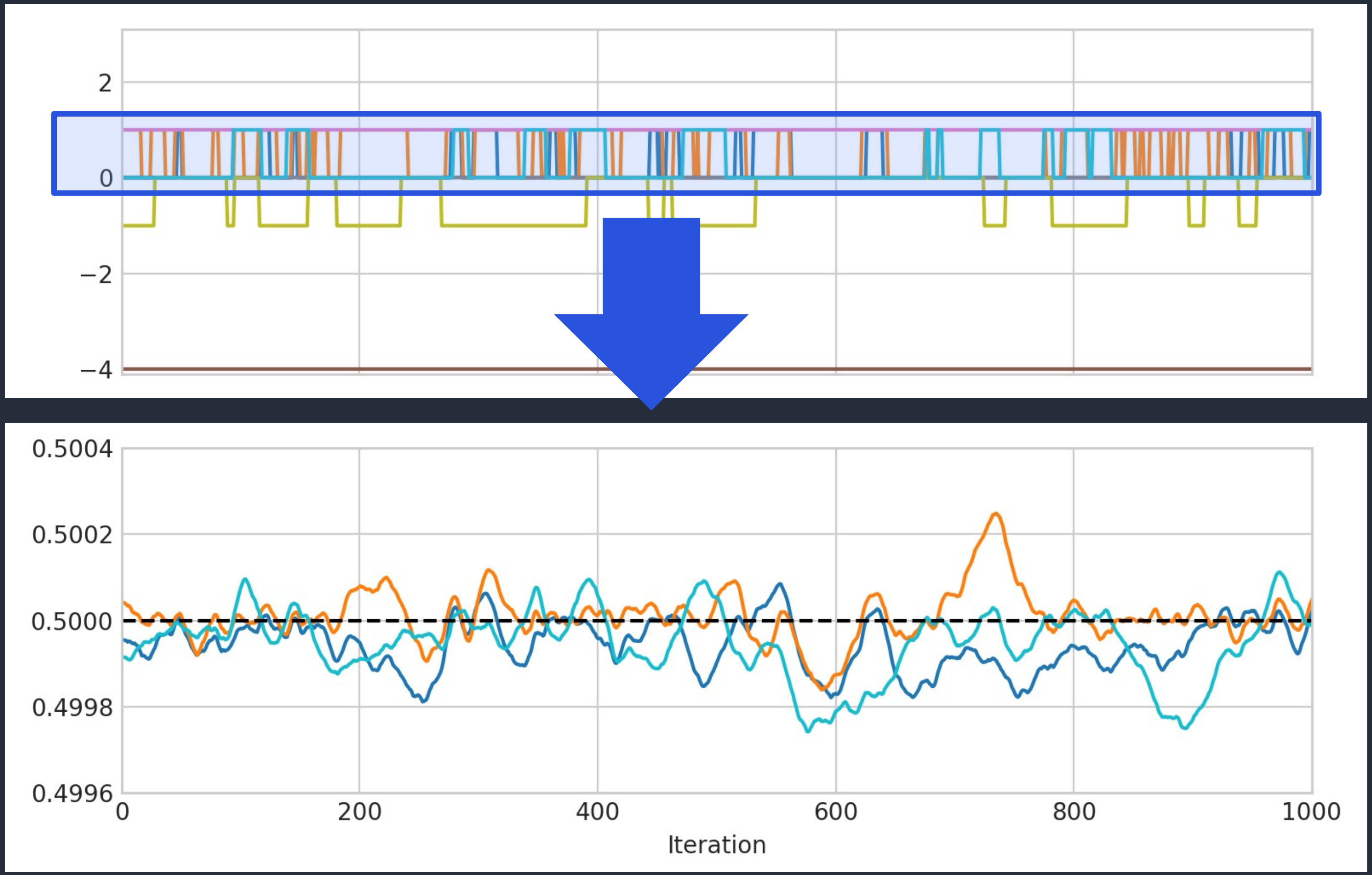


Oscillations in practice

Example of MobileNetV2 training (last 1000 iterations)

Quantized weights, $q(w)$

Latent weights, w



Do oscillations harm more than BN statistics?

Do oscillations harm more than BN statistics?

- Weights are either stationary or oscillating

Do oscillations harm more than BN statistics?

- Weights are either stationary or oscillating
- Experiment:
  - Stochastically sample all oscillating weights after QAT

Do oscillations harm more than BN statistics?

- Weights are either stationary or oscillating
- Experiment:
  - Stochastically sample all oscillating weights after QAT

Method	Train Loss	Val. Acc. (%)
Baseline	1.3566	69.50
SR (mean + std)	$1.3547^{0.0053}$	$69.58^{0.09}$
SR (best)	1.3391	69.85
AdaRound	1.3070	70.12
Freezing	-	70.33

MobileNetV2 with 3-bit quantized weights

Do oscillations harm more than BN statistics?

- Weights are either stationary or oscillating
- Experiment:
  - Stochastically sample all oscillating weights after QAT
  - Average sample is on par with baseline

Method	Train Loss	Val. Acc. (%)
Baseline	1.3566	69.50
SR (mean + std)	$1.3547^{0.0053}$	$69.58^{0.09}$
SR (best)	1.3391	69.85
AdaRound	1.3070	70.12
Freezing	-	70.33

MobileNetV2 with 3-bit quantized weights

Do oscillations harm more than BN statistics?

- Weights are either stationary or oscillating
- Experiment:
  - Stochastically sample all oscillating weights after QAT
  - Average sample is on par with baseline
  - Best sampled quantized weights have lower train loss than baseline

Method	Train Loss	Val. Acc. (%)
Baseline	1.3566	69.50
SR (mean + std)	$1.3547^{0.0053}$	$69.58^{0.09}$
SR (best)	1.3391	69.85
AdaRound	1.3070	70.12
Freezing	-	70.33

MobileNetV2 with 3-bit quantized weights

Do oscillations harm more than BN statistics?

- Weights are either stationary or oscillating
- Experiment:
  - Stochastically sample all oscillating weights after QAT
  - Average sample is on par with baseline
  - Best sampled quantized weights have lower train loss than baseline
  - Binary optimization (AdaRound) of oscillating weights further improves

Method	Train Loss	Val. Acc. (%)
Baseline	1.3566	69.50
SR (mean + std)	$1.3547^{0.0053}$	$69.58^{0.09}$
SR (best)	1.3391	69.85
AdaRound	1.3070	70.12
Freezing	-	70.33

MobileNetV2 with 3-bit quantized weights

Do oscillations harm more than BN statistics?

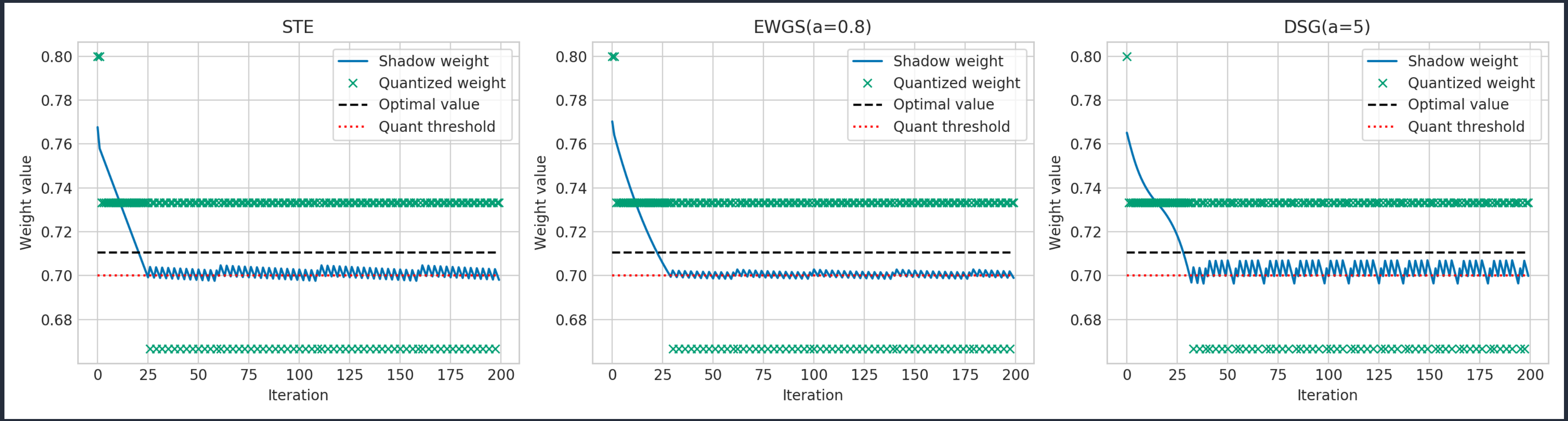
- Weights are either stationary or oscillating
- Experiment:
  - Stochastically sample all oscillating weights after QAT
  - Average sample is on par with baseline
  - Best sampled quantized weights have lower train loss than baseline
  - Binary optimization (AdaRound) of oscillating weights further improves
- Oscillating weights prevent the network from converging to best local minimum!

Method	Train Loss	Val. Acc. (%)
Baseline	1.3566	69.50
SR (mean + std)	$1.3547^{0.0053}$	$69.58^{0.09}$
SR (best)	1.3391	69.85
AdaRound	1.3070	70.12
Freezing	-	70.33

MobileNetV2 with 3-bit quantized weights

Additional insights on oscillations

- Learning rate only effects the amplitude, but not the frequency of oscillations
- Oscillations also affect alternatives to STE, e.g. EWGS[4], DSQ[5] etc
- Check our paper for more theoretical insights

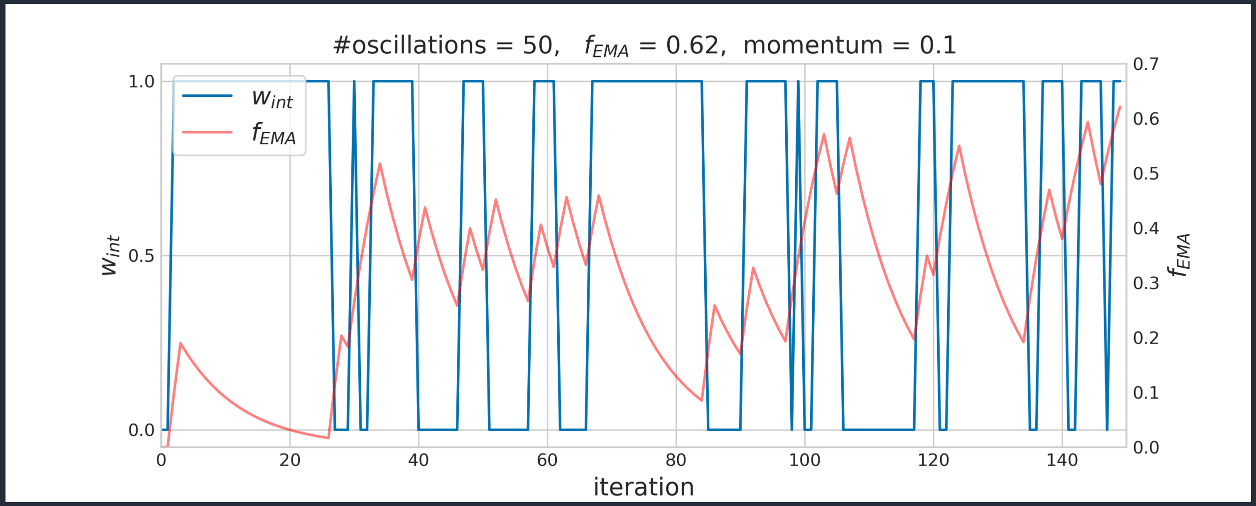


[4] J. Lee, D. Kim, B. H. Network quantization with element- wise gradient scaling. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
[5] Gong, R., Liu, X., Jiang, S., Li, T., Hu, P., Lin, J., Yu, F., and Yan, J. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. International Conference on Computer Vision (ICCV), 2019.

Overcoming oscillations

Tracking oscillations

- Oscillation occurs if integer value changes and its direction opposite to its previous one
- We track oscillations using an EMA of oscillations



Iterative weight freezing

Iterative weight freezing

- We define an oscillation threshold: f_{th}

Algorithm 1 QAT with iterative weight freezing

1: Init: $f^0 \leftarrow \mathbf{0}, b \leftarrow \mathbf{0}, \Delta^\tau \leftarrow \mathbf{0}, \mathbf{w}^0_{\mathrm{EMA(int)}} \leftarrow \mathbf{w}^0_{\mathrm{int}}$

Iterative weight freezing

- We define an oscillation threshold: f_{th}
- For step t in training iterations and for each weight:

Algorithm 1 QAT with iterative weight freezing

1: Init: $f^0 \leftarrow \mathbf{0}, b \leftarrow \mathbf{0}, \Delta^\tau \leftarrow \mathbf{0}, \mathbf{w}^0_{\text{EMA(int)}} \leftarrow \mathbf{w}^0_{\text{int}}$
2: **for** t = 1,. . . , T **do**

Iterative weight freezing

- We define an oscillation threshold: f_{th}
- For step t in training iterations and for each weight:
  1. Calculate the EMA oscillation frequency

Algorithm 1 QAT with iterative weight freezing

Init: $f^0 \leftarrow \mathbf{0}, b \leftarrow \mathbf{0}, \Delta^\tau \leftarrow \mathbf{0}, \mathbf{w}^0_{\text{EMA(int)}} \leftarrow \mathbf{w}^0_{\text{int}}$
for t = 1,..., T **do**
  Calculate gradient $g^t = \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$
  Optimizer update for weights $\mathbf{w}^t[\neg b]$ using g^t
  $\mathbf{w}^t_{\text{int}} \leftarrow \text{clip}\left(\left\lfloor \frac{\mathbf{w}^t}{s} \right\rceil, n, p\right)$
  $\Delta^t_{\text{int}} \leftarrow \mathbf{w}^t_{\text{int}} - \mathbf{w}^{t-1}_{\text{int}}$
  $o^t \leftarrow (\text{sign}(\Delta^t_{\text{int}}) \neq \text{sign}(\Delta^\tau_{\text{int}})) \odot (\Delta^t_{\text{int}} \neq 0)$
  $f^t \leftarrow m \cdot o^t + (1-m) \cdot f^{t-1}$

calculate f^t

Iterative weight freezing

- We define an oscillation threshold: f_{th}
- For step t in training iterations and for each weight:
  1. Calculate the EMA oscillation frequency
  2. If frequency exceeds the threshold ($f^t > f_{\text{th}}$), we freeze that weight

Algorithm 1 QAT with iterative weight freezing

Init: $f^0 \leftarrow \mathbf{0}, b \leftarrow \mathbf{0}, \Delta^\tau \leftarrow \mathbf{0}, \mathbf{w}^0_{\text{EMA(int)}} \leftarrow \mathbf{w}^0_{\text{int}}$
for t = 1,..., T **do**
  Calculate gradient $g^t = \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$
  Optimizer update for weights $\mathbf{w}^t[\neg b]$ using g^t
  $\mathbf{w}^t_{\text{int}} \leftarrow \text{clip}\left(\left\lfloor \frac{\mathbf{w}^t}{s} \right\rceil, n, p\right)$ calculate f^t
  $\Delta^t_{\text{int}} \leftarrow \mathbf{w}^t_{\text{int}} - \mathbf{w}^{t-1}_{\text{int}}$
  $o^t \leftarrow (\text{sign}(\Delta^t_{\text{int}}) \neq \text{sign}(\Delta^\tau_{\text{int}})) \odot (\Delta^t_{\text{int}} \neq 0)$
  $f^t \leftarrow m \cdot o^t + (1 - m) \cdot f^{t-1}$
  for i = 1,..., N **do**
    if $f^t_i > f_{\text{th}}$ **then**

Iterative weight freezing

- We define an oscillation threshold: f_{th}
- For step t in training iterations and for each weight:
  1. Calculate the EMA oscillation frequency
  2. If frequency exceeds the threshold ($f^t > f_{\text{th}}$), we freeze that weight
  3. Assign optimal value to frozen weights

Algorithm 1 QAT with iterative weight freezing

Init: $f^0 \leftarrow \mathbf{0}, b \leftarrow \mathbf{0}, \Delta^\tau \leftarrow \mathbf{0}, \mathbf{w}^0_{\text{EMA(int)}} \leftarrow \mathbf{w}^0_{\text{int}}$
for t = 1,..., T **do**
  Calculate gradient $g^t = \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$
  Optimizer update for weights $\mathbf{w}^t[\neg b]$ using g^t
  $\mathbf{w}^t_{\text{int}} \leftarrow \text{clip}\left(\left\lfloor \frac{\mathbf{w}^t}{s} \right\rceil, n, p\right)$

calculate f^t

  $\Delta^t_{\text{int}} \leftarrow \mathbf{w}^t_{\text{int}} - \mathbf{w}^{t-1}_{\text{int}}$
  $o^t \leftarrow (\text{sign}(\Delta^t_{\text{int}}) \neq \text{sign}(\Delta^\tau_{\text{int}})) \odot (\Delta^t_{\text{int}} \neq 0)$
  $f^t \leftarrow m \cdot o^t + (1 - m) \cdot f^{t-1}$
  for i = 1,..., N **do**
    if $f^t_i > f_{\text{th}}$ **then**
      $b_i \leftarrow$ True
      $\mathbf{w}^t_i \leftarrow s \cdot \left\lfloor \mathbf{w}^{t-1}_{\text{EMA(int)}_i} \right\rceil$
    end if

Iterative weight freezing

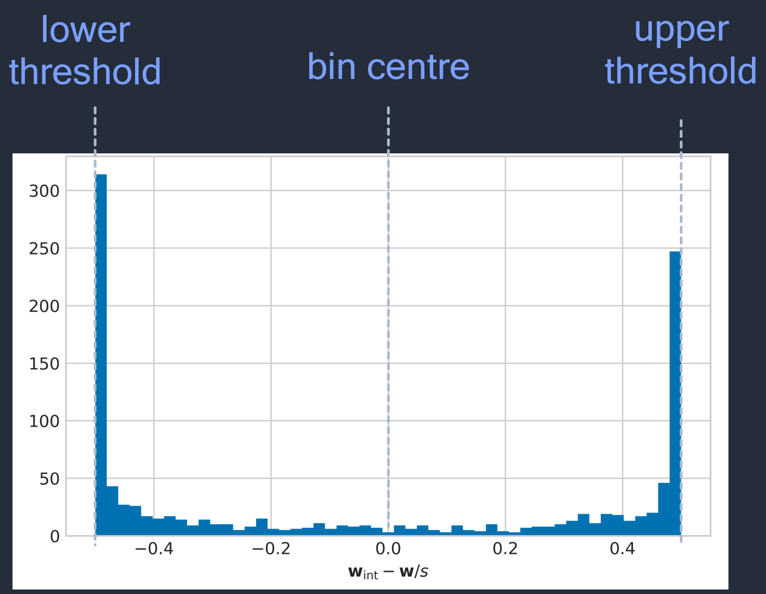
- We define an oscillation threshold: f_{th}
- For step t in training iterations and for each weight:
  1. Calculate the EMA oscillation frequency
  2. If frequency exceeds the threshold ($f^t > f_{\text{th}}$), we freeze that weight
  3. Assign optimal value to frozen weights
  4. Update EMA of integer weight, $\mathbf{w}^t_{\text{EMA(int)}}$

Algorithm 1 QAT with iterative weight freezing

Init: $f^0 \leftarrow \mathbf{0}, b \leftarrow \mathbf{0}, \Delta^\tau \leftarrow \mathbf{0}, \mathbf{w}^0_{\text{EMA(int)}} \leftarrow \mathbf{w}^0_{\text{int}}$
for t = 1, . . . , T **do**
  Calculate gradient $g^t = \frac{\partial \mathcal{L}}{\partial \mathbf{w}}$
  Optimizer update for weights $\mathbf{w}^t[\neg b]$ using g^t
  $\mathbf{w}^t_{\text{int}} \leftarrow \text{clip}\left(\left\lfloor \frac{\mathbf{w}^t}{s} \right\rceil, n, p\right)$
calculate f^t
  $\Delta^t_{\text{int}} \leftarrow \mathbf{w}^t_{\text{int}} - \mathbf{w}^{t-1}_{\text{int}}$
  $o^t \leftarrow (\text{sign}(\Delta^t_{\text{int}}) \neq \text{sign}(\Delta^\tau_{\text{int}})) \odot (\Delta^t_{\text{int}} \neq 0)$
  $f^t \leftarrow m \cdot o^t + (1-m) \cdot f^{t-1}$
  for i = 1, . . . , N **do**
    if $f^t_i > f_{\text{th}}$ **then**
      $b_i \leftarrow$ True
      $\mathbf{w}^t_i \leftarrow s \cdot \left\lfloor \mathbf{w}^{t-1}_{\text{EMA(int)}_i} \right\rceil$
    end if
  end for
calculate EMA of int weights
  $\mathbf{w}^t_{\text{EMA(int)}} \leftarrow m \cdot \mathbf{w}^{t-1} + (1-m) \cdot \mathbf{w}^{t-1}_{\text{EMA(int)}}$
  $\Delta^\tau_{\text{int}}[o^t] \leftarrow \Delta^t_{\text{int}}[o^t]$
end for

Oscillation dampening

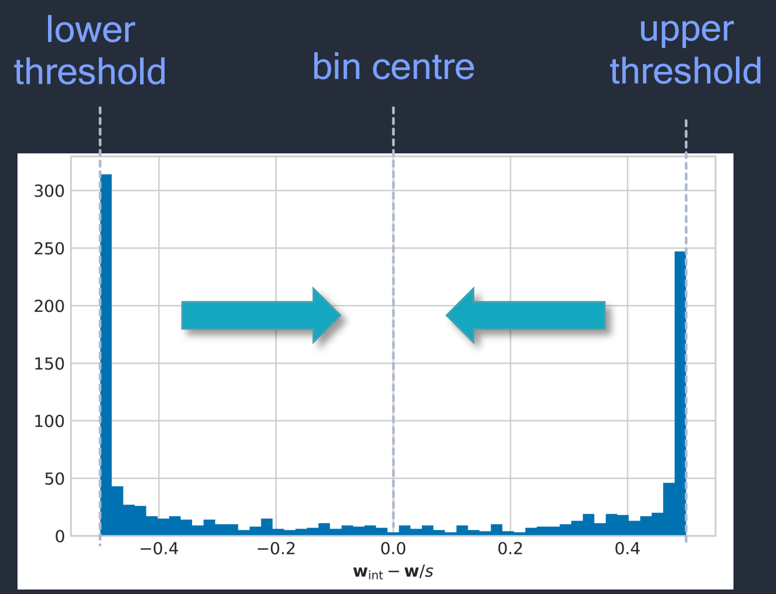
- Oscillating weights are always close to the quantization bin edge



Oscillation dampening

- Oscillating weights are always close to the quantization bin edge
- We regularize weights to force them closer to the centre of then bin

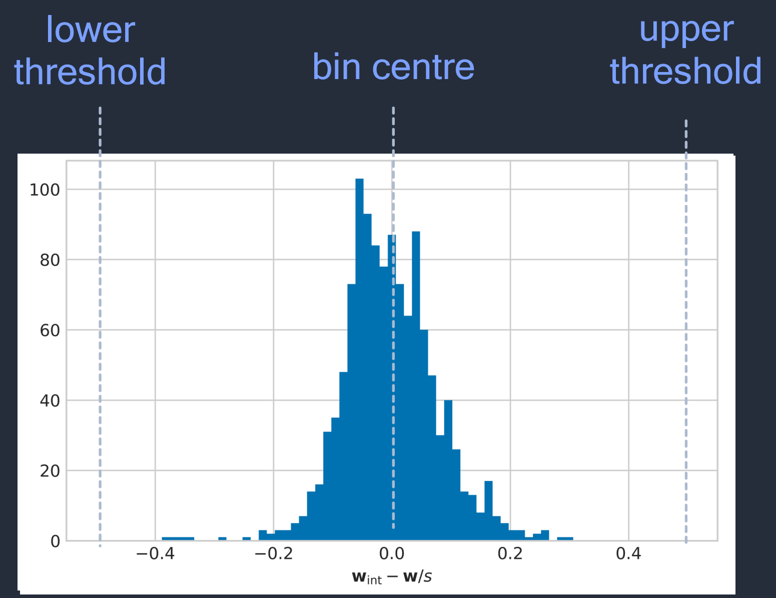
$$L_{\text{dampen}} = \frac{1}{2} \| q(\mathbf{w}) \; - \text{clip}(\mathbf{w}, q_{\text{min}}, q_{\text{max}}) \|_2^F$$



Oscillation dampening

- Oscillating weights are always close to the quantization bin edge
- We regularize weights to force them closer to the centre of then bin

$$L_{\text{dampen}} = \frac{1}{2} \| q(\mathbf{w}) \ - \text{clip}(\mathbf{w}, q_{\min}, q_{\max}) \|_2^F$$

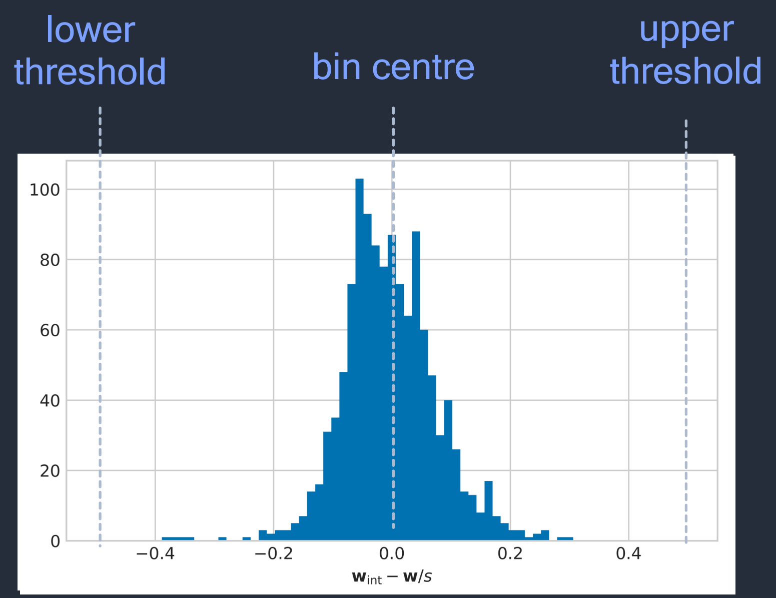


Oscillation dampening

- Oscillating weights are always close to the quantization bin edge
- We regularize weights to force them closer to the centre of then bin

$$L_{\text{dampen}} = \frac{1}{2} \| q(\mathbf{w}) - \text{clip}(\mathbf{w}, q_{\min}, q_{\max}) \|_2^F$$

- Final training objective: $L = L_{\text{task}} + \lambda L_{\text{dampen}}$



Experiments

Ablation study I: dampening strength

MobiletNetV2

Regulatization	pre-BN	post-BN	Osc.(%)
Baseline	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$\lambda = 10^{-4}$	$65.97^{1.52}$	$69.65^{0.08}$	2.18
$\lambda = 10^{-3}$	$66.99^{1.41}$	$\mathbf{69.96}^{0.12}$	0.21
$\lambda = 10^{-2}$	$\mathbf{68.04}^{1.04}$	$68.57^{0.07}$	0.01
$\lambda = \cos(0, 10^{-4})$	$64.47^{1.59}$	$69.61^{0.07}$	2.64
$\lambda = \cos(0, 10^{-3})$	$68.79^{1.31}$	$\mathbf{70.37}^{0.06}$	1.63
$\lambda = \cos(0, 10^{-2})$	$\mathbf{70.18}^{0.18}$	$70.26^{0.08}$	1.11

Ablation study I: dampening strength

✓Strong dampening reduces oscillations

MobiletNetV2

Regulatization	pre-BN	post-BN	Osc.(%)
Baseline	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$\lambda = 10^{-4}$	$65.97^{1.52}$	$69.65^{0.08}$	2.18
$\lambda = 10^{-3}$	$66.99^{1.41}$	$\mathbf{69.96}^{0.12}$	0.21
$\lambda = 10^{-2}$	$\mathbf{68.04}^{1.04}$	$68.57^{0.07}$	0.01
$\lambda = \cos(0, 10^{-4})$	$64.47^{1.59}$	$69.61^{0.07}$	2.64
$\lambda = \cos(0, 10^{-3})$	$68.79^{1.31}$	$\mathbf{70.37}^{0.06}$	1.63
$\lambda = \cos(0, 10^{-2})$	$\mathbf{70.18}^{0.18}$	$70.26^{0.08}$	1.11

Ablation study I: dampening strength

- ✓Strong dampening reduces oscillations
- ✓Strong dampening closes pre & post-BN re-estimation accuracy gap

MobiletNetV2

Regulatization		pre-BN	post-BN	Osc.(%)
Baseline	δ= 4.5	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$\lambda = 10^{-4}$	δ= 3.7	$65.97^{1.52}$	$69.65^{0.08}$	2.18
$\lambda = 10^{-3}$	δ= 3.0	$66.99^{1.41}$	**69.96**$^{0.12}$	0.21
$\lambda = 10^{-2}$	δ= 0.5	**68.04**$^{1.04}$	$68.57^{0.07}$	0.01
$\lambda = \cos(0, 10^{-4})$		$64.47^{1.59}$	$69.61^{0.07}$	2.64
$\lambda = \cos(0, 10^{-3})$		$68.79^{1.31}$	**70.37**$^{0.06}$	1.63
$\lambda = \cos(0, 10^{-2})$		**70.18**$^{0.18}$	$70.26^{0.08}$	1.11

Ablation study I: dampening strength

- ✓ Strong dampening reduces oscillations
- ✓ Strong dampening closes pre & post-BN re-estimation accuracy gap
- x Strong dampening leads to lower final accuracy

MobiletNetV2

Regulatization		pre-BN	post-BN	Osc.(%)
Baseline	δ= 4.5	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$\lambda = 10^{-4}$	δ= 3.7	$65.97^{1.52}$	$69.65^{0.08}$	2.18
$\lambda = 10^{-3}$	δ= 3.0	$66.99^{1.41}$	$\mathbf{69.96}^{0.12}$	0.21
$\lambda = 10^{-2}$	δ= 0.5	$\mathbf{68.04}^{1.04}$	$68.57^{0.07}$	0.01
$\lambda = \cos(0, 10^{-4})$		$64.47^{1.59}$	$69.61^{0.07}$	2.64
$\lambda = \cos(0, 10^{-3})$		$68.79^{1.31}$	$\mathbf{70.37}^{0.06}$	1.63
$\lambda = \cos(0, 10^{-2})$		$\mathbf{70.18}^{0.18}$	$70.26^{0.08}$	1.11

Ablation study I: dampening strength

✓Strong dampening reduces oscillations

✓Strong dampening closes pre & post-BN re-estimation accuracy gap

x Strong dampening leads to lower final accuracy

MobiletNetV2

Regulatization		pre-BN	post-BN	Osc.(%)
Baseline	δ= 4.5	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$\lambda = 10^{-4}$	δ= 3.7	$65.97^{1.52}$	$69.65^{0.08}$	2.18
$\lambda = 10^{-3}$	δ= 3.0	$66.99^{1.41}$	$\mathbf{69.96}^{0.12}$	0.21
$\lambda = 10^{-2}$	δ= 0.5	$\mathbf{68.04}^{1.04}$	$68.57^{0.07}$	0.01
$\lambda = \cos(0, 10^{-4})$		$64.47^{1.59}$	$69.61^{0.07}$	2.64
$\lambda = \cos(0, 10^{-3})$		$68.79^{1.31}$	$\mathbf{70.37}^{0.06}$	1.63
$\lambda = \cos(0, 10^{-2})$		$\mathbf{70.18}^{0.18}$	$70.26^{0.08}$	1.11

Ablation study I: dampening strength

- ✓Strong dampening reduces oscillations
- ✓Strong dampening closes pre & post-BN re-estimation accuracy gap
- x Strong dampening leads to lower final accuracy
- Hypothesis: weights cannot move freely early during training

MobiletNetV2

Regulatization		pre-BN	post-BN	Osc.(%)
Baseline	δ= 4.5	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$\lambda = 10^{-4}$	δ= 3.7	$65.97^{1.52}$	$69.65^{0.08}$	2.18
$\lambda = 10^{-3}$	δ= 3.0	$66.99^{1.41}$	$\mathbf{69.96}^{0.12}$	0.21
$\lambda = 10^{-2}$	δ= 0.5	$\mathbf{68.04}^{1.04}$	$68.57^{0.07}$	0.01
$\lambda = \cos(0, 10^{-4})$		$64.47^{1.59}$	$69.61^{0.07}$	2.64
$\lambda = \cos(0, 10^{-3})$		$68.79^{1.31}$	$\mathbf{70.37}^{0.06}$	1.63
$\lambda = \cos(0, 10^{-2})$		$\mathbf{70.18}^{0.18}$	$70.26^{0.08}$	1.11

Ablation study I: dampening strength

✓Strong dampening reduces oscillations

✓Strong dampening closes pre & post-BN re-estimation accuracy gap

x Strong dampening leads to lower final accuracy

- Hypothesis: weights cannot move freely early during training
- Solution: gradually increase (anneal) λ during training

MobiletNetV2

Regulatization		pre-BN	post-BN	Osc.(%)
Baseline	δ= 4.5	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$\lambda = 10^{-4}$	δ= 3.7	$65.97^{1.52}$	$69.65^{0.08}$	2.18
$\lambda = 10^{-3}$	δ= 3.0	$66.99^{1.41}$	**69.96**$^{0.12}$	0.21
$\lambda = 10^{-2}$	δ= 0.5	**68.04**$^{1.04}$	$68.57^{0.07}$	0.01
$\lambda = \cos(0, 10^{-4})$		$64.47^{1.59}$	$69.61^{0.07}$	2.64
$\lambda = \cos(0, 10^{-3})$		$68.79^{1.31}$	**70.37**$^{0.06}$	1.63
$\lambda = \cos(0, 10^{-2})$		**70.18**$^{0.18}$	$70.26^{0.08}$	1.11

Ablation study I: dampening strength

✓Strong dampening reduces oscillations

✓Strong dampening closes pre & post-BN re-estimation accuracy gap

x Strong dampening leads to lower final accuracy

- Hypothesis: weights cannot move freely early during training
- Solution: gradually increase (anneal) λ during training

✓Annealing increases both pre & post-BN re-estimation accuracy

MobiletNetV2

Regulatization		pre-BN	post-BN	Osc.(%)
Baseline	δ= 4.5	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$\lambda = 10^{-4}$	δ= 3.7	$65.97^{1.52}$	$69.65^{0.08}$	2.18
$\lambda = 10^{-3}$	δ= 3.0	$66.99^{1.41}$	**69.96**$^{0.12}$	0.21
$\lambda = 10^{-2}$	δ= 0.5	**68.04**$^{1.04}$	$68.57^{0.07}$	0.01
$\lambda = \cos(0, 10^{-4})$		$64.47^{1.59}$	$69.61^{0.07}$	2.64
$\lambda = \cos(0, 10^{-3})$		$68.79^{1.31}$	**70.37**$^{0.06}$	1.63
$\lambda = \cos(0, 10^{-2})$		**70.18**$^{0.18}$	$70.26^{0.08}$	1.11

Ablation study II: Freezing threshold

MobiletNetV2

Method	pre-BN	post-BN	Osc.(%)
Baseline	$64.97^{1.23}$	$69.50^{0.04}$	4.93
f_{th}= 0.02	$68.13^{2.14}$	$69.96^{0.04}$	2.93
f_{th}= 0.015	$\mathbf{69.79}^{0.07}$	$\mathbf{70.13}^{0.05}$	1.23
f_{th}= 0.01	$69.12^{0.53}$	$69.18^{0.47}$	0.06
f_{th}= cos(0.04,0.015)	$69.51^{0.15}$	$69.96^{0.03}$	2.33
f_{th}= cos(0.04,0.01)	$\mathbf{69.97}^{0.06}$	$\mathbf{70.33}^{0.07}$	0.04

Ablation study II: Freezing threshold

✓Low frequency threshold very effective at reducing oscillations

MobiletNetV2

Method	pre-BN	post-BN	Osc.(%)
Baseline	$64.97^{1.23}$	$69.50^{0.04}$	4.93
f_{th}= 0.02	$68.13^{2.14}$	$69.96^{0.04}$	2.93
f_{th}= 0.015	$\mathbf{69.79}^{0.07}$	$\mathbf{70.13}^{0.05}$	1.23
f_{th}= 0.01	$69.12^{0.53}$	$69.18^{0.47}$	0.06
f_{th}= cos(0.04,0.015)	$69.51^{0.15}$	$69.96^{0.03}$	2.33
f_{th}= cos(0.04,0.01)	$\mathbf{69.97}^{0.06}$	$\mathbf{70.33}^{0.07}$	0.04

Ablation study II: Freezing threshold

- ✓Low frequency threshold very effective at reducing oscillations
- ✓Low frequency threshold closes pre & post-BN re-estimation accuracy gap

MobiletNetV2

Method		pre-BN	post-BN	Osc.(%)
Baseline	$\delta = 4.5$	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$f_{th} = 0.02$	$\delta = 1.8$	$68.13^{2.14}$	$69.96^{0.04}$	2.93
$f_{th} = 0.015$	$\delta = 0.4$	**69.79**$^{0.07}$	**70.13**$^{0.05}$	1.23
$f_{th} = 0.01$	$\delta = 0.5$	$69.12^{0.53}$	$69.18^{0.47}$	0.06
$f_{th} = \cos(0.04, 0.015)$		$69.51^{0.15}$	$69.96^{0.03}$	2.33
$f_{th} = \cos(0.04, 0.01)$		**69.97**$^{0.06}$	**70.33**$^{0.07}$	0.04

Ablation study II: Freezing threshold

✓Low frequency threshold very effective at reducing oscillations

✓Low frequency threshold closes pre & post-BN re-estimation accuracy gap

x Low frequency threshold leads to lower final accuracy

MobiletNetV2

Method		pre-BN	post-BN	Osc.(%)
Baseline	δ= 4.5	$64.97^{1.23}$	$69.50^{0.04}$	4.93
f_{th}= 0.02	δ= 1.8	$68.13^{2.14}$	$69.96^{0.04}$	2.93
f_{th}= 0.015	δ= 0.4	$\mathbf{69.79}^{0.07}$	$\mathbf{70.13}^{0.05}$	1.23
f_{th}= 0.01	δ= 0.5	$69.12^{0.53}$	$69.18^{0.47}$	0.06
f_{th}= cos(0.04,0.015)		$69.51^{0.15}$	$69.96^{0.03}$	2.33
f_{th}= cos(0.04,0.01)		$\mathbf{69.97}^{0.06}$	$\mathbf{70.33}^{0.07}$	0.04

Ablation study II: Freezing threshold

✓Low frequency threshold very effective at reducing oscillations

✓Low frequency threshold closes pre & post-BN re-estimation accuracy gap

x Low frequency threshold leads to lower final accuracy

- Solution: gradually reduce (anneal) f_{th} during training

MobiletNetV2

Method		pre-BN	post-BN	Osc.(%)
Baseline	δ= 4.5	$64.97^{1.23}$	$69.50^{0.04}$	4.93
f_{th}= 0.02	δ= 1.8	$68.13^{2.14}$	$69.96^{0.04}$	2.93
f_{th}= 0.015	δ= 0.4	$\mathbf{69.79}^{0.07}$	$\mathbf{70.13}^{0.05}$	1.23
f_{th}= 0.01	δ= 0.5	$69.12^{0.53}$	$69.18^{0.47}$	0.06
f_{th}= cos(0.04,0.015)		$69.51^{0.15}$	$69.96^{0.03}$	2.33
f_{th}= cos(0.04,0.01)		$\mathbf{69.97}^{0.06}$	$\mathbf{70.33}^{0.07}$	0.04

Ablation study II: Freezing threshold

- ✓ Low frequency threshold very effective at reducing oscillations
- ✓ Low frequency threshold closes pre & post-BN re-estimation accuracy gap
- x Low frequency threshold leads to lower final accuracy
- Solution: gradually reduce (anneal) f_{th} during training
- ✓ Annealing increases both pre & post-BN re-estimation accuracy

MobiletNetV2

Method		pre-BN	post-BN	Osc.(%)
Baseline	$\delta = 4.5$	$64.97^{1.23}$	$69.50^{0.04}$	4.93
$f_{\text{th}} = 0.02$	$\delta = 1.8$	$68.13^{2.14}$	$69.96^{0.04}$	2.93
$f_{\text{th}} = 0.015$	$\delta = 0.4$	$\mathbf{69.79}^{0.07}$	$\mathbf{70.13}^{0.05}$	1.23
$f_{\text{th}} = 0.01$	$\delta = 0.5$	$69.12^{0.53}$	$69.18^{0.47}$	0.06
$f_{\text{th}} = \cos(0.04, 0.015)$		$69.51^{0.15}$	$69.96^{0.03}$	2.33
$f_{\text{th}} = \cos(0.04, 0.01)$		$\mathbf{69.97}^{0.06}$	$\mathbf{70.33}^{0.07}$	0.04

MobileNetV2 - comparison to literature

- We train with LSQ[6] and BN re-estimation
- We achieve SOTA for W4A4 and W3A3
- Dampening and freezing preform on par
- Freezing faster during training than dampening ~30%

MobiletNetV2

Method	W/A	Val. Acc. (%)
Full-precision	32/32	71.7
LSQ* (Esser et al., 2020)	4/4	69.5 (-2.3)
PACT (Choi et al., 2018)	4/4	61.4 (-10.3)
DSQ (Gong et al., 2019)	4/4	64.8 (-6.9)
EWGS (J. Lee, 2021)	4/4	70.3 (-1.6)
LSQ + BR (Han et al., 2021)	4/4	70.4 (-1.4)
LSQ + Dampen (ours)	4/4	**70.5** (-1.2)
LSQ + Freeze (ours)	4/4	**70.6** (-1.1)
LSQ* (Esser et al., 2020)	3/3	65.3 (-6.5)
LSQ + BR (Han et al., 2021)	3/3	67.4 (-4.4)
LSQ + Dampen (ours)	3/3	**67.8** (-3.9)
LSQ + Freeze (ours)	3/3	**67.6** (-4.1)

[6] Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. In International Conference on Learning Representations (ICLR), 2020

MobileNetV2 - comparison to literature

- We train with LSQ[6] and BN re-estimation
- We achieve SOTA for W4A4 and W3A3
- Dampening and freezing preform on par
- Freezing faster during training than dampening ~30%

MobiletNetV2

Method	W/A	Val. Acc. (%)
Full-precision	32/32	71.7
LSQ* (Esser et al., 2020)	4/4	69.5 (-2.3)
PACT (Choi et al., 2018)	4/4	61.4 (-10.3)
DSQ (Gong et al., 2019)	4/4	64.8 (-6.9)
EWGS (J. Lee, 2021)	4/4	70.3 (-1.6)
LSQ + BR (Han et al., 2021)	4/4	70.4 (-1.4)
LSQ + Dampen (ours)	4/4	**70.5** (-1.2)
LSQ + Freeze (ours)	4/4	**70.6** (-1.1)
LSQ* (Esser et al., 2020)	3/3	65.3 (-6.5)
LSQ + BR (Han et al., 2021)	3/3	67.4 (-4.4)
LSQ + Dampen (ours)	3/3	**67.8** (-3.9)
LSQ + Freeze (ours)	3/3	**67.6** (-4.1)

[6] Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. In International Conference on Learning Representations (ICLR), 2020

MobileNetV2 - comparison to literature

- We train with LSQ[6] and BN re-estimation
- We achieve SOTA for W4A4 and W3A3
- Dampening and freezing preform on par
- Freezing faster during training than dampening ~30%

MobiletNetV2

Method	W/A	Val. Acc. (%)
Full-precision	32/32	71.7
LSQ* (Esser et al., 2020)	4/4	69.5 (-2.3)
PACT (Choi et al., 2018)	4/4	61.4 (-10.3)
DSQ (Gong et al., 2019)	4/4	64.8 (-6.9)
EWGS (J. Lee, 2021)	4/4	70.3 (-1.6)
LSQ + BR (Han et al., 2021)	4/4	70.4 (-1.4)
LSQ + Dampen (ours)	4/4	**70.5** (-1.2)
LSQ + Freeze (ours)	4/4	**70.6** (-1.1)
LSQ* (Esser et al., 2020)	3/3	65.3 (-6.5)
LSQ + BR (Han et al., 2021)	3/3	67.4 (-4.4)
LSQ + Dampen (ours)	3/3	**67.8** (-3.9)
LSQ + Freeze (ours)	3/3	**67.6** (-4.1)

[6] Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. In International Conference on Learning Representations (ICLR), 2020

MobileNetV2 - comparison to literature

- We train with LSQ[6] and BN re-estimation
- We achieve SOTA for W4A4 and W3A3
- Dampening and freezing preform on par
- Freezing faster during training than dampening ~30%

MobiletNetV2

Method	W/A	Val. Acc. (%)
Full-precision	32/32	71.7
LSQ* (Esser et al., 2020)	4/4	69.5 (-2.3)
PACT (Choi et al., 2018)	4/4	61.4 (-10.3)
DSQ (Gong et al., 2019)	4/4	64.8 (-6.9)
EWGS (J. Lee, 2021)	4/4	70.3 (-1.6)
LSQ + BR (Han et al., 2021)	4/4	70.4 (-1.4)
LSQ + Dampen (ours)	4/4	**70.5** (-1.2)
LSQ + Freeze (ours)	4/4	**70.6** (-1.1)
LSQ* (Esser et al., 2020)	3/3	65.3 (-6.5)
LSQ + BR (Han et al., 2021)	3/3	67.4 (-4.4)
LSQ + Dampen (ours)	3/3	**67.8** (-3.9)
LSQ + Freeze (ours)	3/3	**67.6** (-4.1)

[6] Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. In International Conference on Learning Representations (ICLR), 2020

MobileNetV2 - comparison to literature

- We train with LSQ[6] and BN re-estimation
- We achieve SOTA for W4A4 and W3A3
- Dampening and freezing preform on par
- Freezing faster during training than dampening ~30%

MobiletNetV2

Method	W/A	Val. Acc. (%)
Full-precision	32/32	71.7
LSQ* (Esser et al., 2020)	4/4	69.5 (-2.3)
PACT (Choi et al., 2018)	4/4	61.4 (-10.3)
DSQ (Gong et al., 2019)	4/4	64.8 (-6.9)
EWGS (J. Lee, 2021)	4/4	70.3 (-1.6)
LSQ + BR (Han et al., 2021)	4/4	70.4 (-1.4)
LSQ + Dampen (ours)	4/4	**70.5** (-1.2)
LSQ + Freeze (ours)	4/4	**70.6** (-1.1)
LSQ* (Esser et al., 2020)	3/3	65.3 (-6.5)
LSQ + BR (Han et al., 2021)	3/3	67.4 (-4.4)
LSQ + Dampen (ours)	3/3	**67.8** (-3.9)
LSQ + Freeze (ours)	3/3	**67.6** (-4.1)

[6] Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. In International Conference on Learning Representations (ICLR), 2020

MobileNetV2 – comparison to literature

- We train with LSQ[6] and BN re-estimation
- We achieve SOTA for W4A4 and W3A3
- Dampening and freezing preform on par
- Freezing faster during training than dampening ~30%

MobiletNetV2

Method	W/A	Val. Acc. (%)
Full-precision	32/32	71.7
LSQ* (Esser et al., 2020)	4/4	69.5 (-2.3)
PACT (Choi et al., 2018)	4/4	61.4 (-10.3)
DSQ (Gong et al., 2019)	4/4	64.8 (-6.9)
EWGS (J. Lee, 2021)	4/4	70.3 (-1.6)
LSQ + BR (Han et al., 2021)	4/4	70.4 (-1.4)
LSQ + Dampen (ours)	4/4	**70.5** (-1.2)
LSQ + Freeze (ours)	4/4	**70.6** (-1.1)
LSQ* (Esser et al., 2020)	3/3	65.3 (-6.5)
LSQ + BR (Han et al., 2021)	3/3	67.4 (-4.4)
LSQ + Dampen (ours)	3/3	**67.8** (-3.9)
LSQ + Freeze (ours)	3/3	**67.6** (-4.1)

[6] Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., and Modha, D. S. Learned step size quantization. In International Conference on Learning Representations (ICLR), 2020

MobileNetV3-Small & EfficientNet-lite

MobiletNetV3

Method	W/A	Val. Acc. (%)
Full-precision	32/32	65.1
LSQ* (Esser et al., 2020)	4/4	61.0
LSQ + BR (Han et al., 2021)	4/4	61.5
LSQ + Dampen (ours)	4/4	**63.7**
LSQ + Freeze (ours)	4/4	**63.6**
LSQ* (Esser et al., 2020)	3/3	52.0
LSQ + BR (Han et al., 2021)	3/3	56.0
LSQ + Dampen (ours)	3/3	**59.0**
LSQ + Freeze (ours)	3/3	**58.9**

EfficientNet-Lite

Method	W/A	Val. Acc. (%)
Full-precision	32/32	75.4
LSQ* (Esser et al., 2020)	4/4	72.3
LSQ + Dampen (ours)	4/4	**73.5**
LSQ + Freeze (ours)	4/4	**73.5**
LSQ* (Esser et al., 2020)	3/3	69.7
LSQ + Dampen (ours)	3/3	**71.1**
LSQ + Freeze (ours)	3/3	**71.0**

MobileNetV3-Small & EfficientNet-lite

MobiletNetV3

Method	W/A	Val. Acc. (%)	
Full-precision	32/32	65.1	
LSQ* (Esser et al., 2020)	4/4	61.0	
LSQ + BR (Han et al., 2021)	4/4	61.5	
LSQ + Dampen (ours)	4/4	**63.7**	+2.7
LSQ + Freeze (ours)	4/4	**63.6**	+2.6
LSQ* (Esser et al., 2020)	3/3	52.0	
LSQ + BR (Han et al., 2021)	3/3	56.0	
LSQ + Dampen (ours)	3/3	**59.0**	
LSQ + Freeze (ours)	3/3	**58.9**	

EfficientNet-Lite

Method	W/A	Val. Acc. (%)
Full-precision	32/32	75.4
LSQ* (Esser et al., 2020)	4/4	72.3
LSQ + Dampen (ours)	4/4	**73.5**
LSQ + Freeze (ours)	4/4	**73.5**
LSQ* (Esser et al., 2020)	3/3	69.7
LSQ + Dampen (ours)	3/3	**71.1**
LSQ + Freeze (ours)	3/3	**71.0**

MobileNetV3-Small & EfficientNet-lite

MobiletNetV3

Method	W/A	Val. Acc. (%)	
Full-precision	32/32	65.1	
LSQ* (Esser et al., 2020)	4/4	61.0	
LSQ + BR (Han et al., 2021)	4/4	61.5	
LSQ + Dampen (ours)	4/4	**63.7**	+2.7
LSQ + Freeze (ours)	4/4	**63.6**	+2.6
LSQ* (Esser et al., 2020)	3/3	52.0	
LSQ + BR (Han et al., 2021)	3/3	56.0	
LSQ + Dampen (ours)	3/3	**59.0**	+7.0
LSQ + Freeze (ours)	3/3	**58.9**	+6.9

EfficientNet-Lite

Method	W/A	Val. Acc. (%)
Full-precision	32/32	75.4
LSQ* (Esser et al., 2020)	4/4	72.3
LSQ + Dampen (ours)	4/4	**73.5**
LSQ + Freeze (ours)	4/4	**73.5**
LSQ* (Esser et al., 2020)	3/3	69.7
LSQ + Dampen (ours)	3/3	**71.1**
LSQ + Freeze (ours)	3/3	**71.0**

MobileNetV3-Small & EfficientNet-lite

MobiletNetV3

Method	W/A	Val. Acc. (%)	
Full-precision	32/32	65.1	
LSQ* (Esser et al., 2020)	4/4	61.0	
LSQ + BR (Han et al., 2021)	4/4	61.5	
LSQ + Dampen (ours)	4/4	**63.7**	+2.7
LSQ + Freeze (ours)	4/4	**63.6**	+2.6
LSQ* (Esser et al., 2020)	3/3	52.0	
LSQ + BR (Han et al., 2021)	3/3	56.0	
LSQ + Dampen (ours)	3/3	**59.0**	+7.0
LSQ + Freeze (ours)	3/3	**58.9**	+6.9

EfficientNet-Lite

Method	W/A	Val. Acc. (%)	
Full-precision	32/32	75.4	
LSQ* (Esser et al., 2020)	4/4	72.3	
LSQ + Dampen (ours)	4/4	**73.5**	+1.2
LSQ + Freeze (ours)	4/4	**73.5**	+1.2
LSQ* (Esser et al., 2020)	3/3	69.7	
LSQ + Dampen (ours)	3/3	**71.1**	+1.4
LSQ + Freeze (ours)	3/3	**71.0**	+1.3

Conclusion

- Oscillating weights are an inherent problem of QAT:
  - They corrupt BN statistics
  - They prevent model convergence
- We propose two methods for tackling the source of oscillations:
  - Oscillations dampening
  - Iterative weight freezing
- We achieve SOTA for low-bit quantization of efficient models

paper

code

Thank you

Follow us on:

For more information, visit us at:

qualcomm.com & qualcomm.com/blog