Advanced Quantization Methods

Zaur Datkhuzhev,
Deep Learning research engineer at Huawei
March 2021

girafe ai

Outline

1. Recap on Quantization
2. Quantization granularity
3. Quantization techniques
    a. PACT
    b. LSQ
    c. LSQ+
4. Layer-wise and kernel-wise methods
    a. ReLeQ
    b. HAQ
    c. AutoQ

Recap on Quantization

girafe ai

01

Recap on Quantization

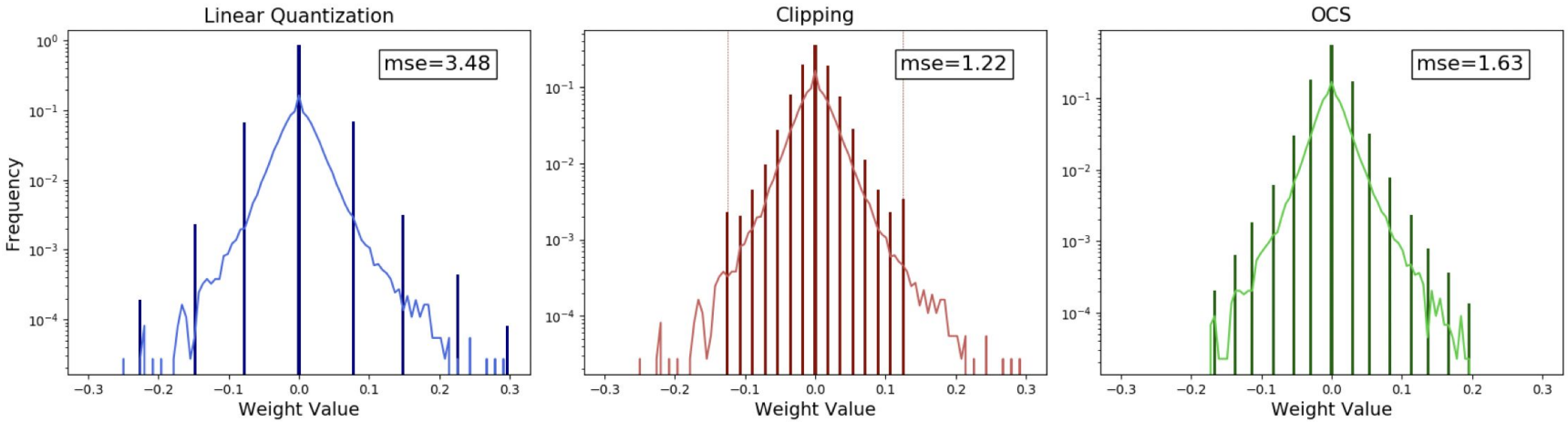


image source

Recap on Quantization

$$\bar{v} = \lfloor clip(v/s, -Q_N, Q_P) \rceil,$$

$$\hat{v} = \bar{v} \times s.$$

$$\frac{\partial \hat{v}}{\partial v} = \begin{cases} 1 & \text{if } -Q_N < v/s < Q_P \\ 0 & \text{otherwise,} \end{cases}$$

- v - data which we quantize,
- s - scale
- b - bit width
- Qn - 0(unsigned data), 2^(b-1) (signed data)
- Qp - 2^b - 1 (unsigned data), 2^(b-1) - 1 (signed data)

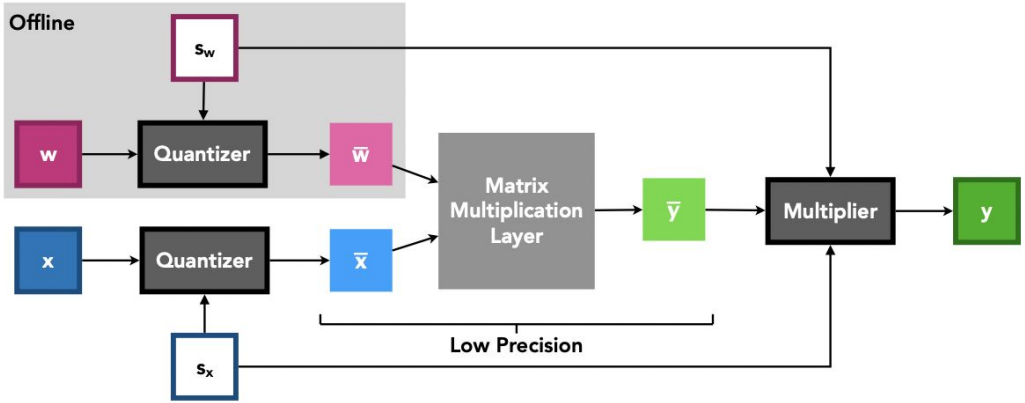


image source

Quantization granularity

girafe ai

02

Quantization granularity

Table 1: The search space size of network quantization. $QBN \in [0, 32]$, where 0 means the component is pruned. n_{layer} is the layer number of the network.

quantization granularity	search space size (weight × activation)
network-wise	33×33
layer-wise	$33^{n_{layer}} \times 33^{n_{layer}}$
kernel-wise	$33^{\sum_{i=1}^{n_{layer}} c_{outi}} \times 33^{n_{layer}}$

image source

Quantization techniques

girafe ai

03

PACT

Parameterized Clipping Activation for Quantized Neural Networks (code)

$$y = PACT(x) = 0.5(|x| - |x - \alpha| + \alpha) = \begin{cases} 0, & x \in (-\infty, 0) \\ x, & x \in [0, \alpha) \\ \alpha, & x \in [\alpha, +\infty) \end{cases}$$

$$y_q = round(y \cdot \frac{2^k - 1}{\alpha}) \cdot \frac{\alpha}{2^k - 1}$$

$$\frac{\partial y_q}{\partial \alpha} = \frac{\partial y_q}{\partial y} \frac{\partial y}{\partial \alpha} = \begin{cases} 0, & x \in (-\infty, \alpha) \\ 1, & x \in [\alpha, +\infty) \end{cases}$$

image source

PACT

Table 1: Comparison of top-1 accuracy between DoReFa and PACT. Weights are quantized with DoReFa scheme, whereas activations are quantized with our scheme. Note that CNNs with $4b$ quantization based on our scheme achieves full-precision accuracy for all the CNNs we explored.

Network	FullPrec	DoReFa				PACT			
		2b	3b	4b	5b	2b	3b	4b	5b
CIFAR10	0.916	0.882	0.899	0.905	0.904	0.897	0.911	0.913	0.917
SVHN	0.978	0.976	0.976	0.975	0.975	0.977	0.978	0.978	0.979
AlexNet	0.551	0.536	0.550	0.549	0.549	0.550	0.556	0.557	0.557
ResNet18	0.702	0.626	0.675	0.681	0.684	0.644	0.681	0.692	0.698
ResNet50	0.769	0.671	0.699	0.714	0.714	0.722	0.753	0.765	0.767

LSQ

Learned Step Size Quantization (code)

$$\frac{\partial \hat{v}}{\partial s} = \begin{cases} -v/s + \lfloor v/s \rceil & \text{if } -Q_N < v/s < Q_P \\ -Q_N & \text{if } v/s \leq -Q_N \\ Q_P & \text{if } v/s \geq Q_P \end{cases}$$

For this work, each layer of weights and each layer of activations has a distinct step size, represented as an fp32 value, initialized to $2\langle |v| \rangle / \sqrt{Q_P}$, computed on either the initial weights values or the first batch of activations, respectively.

image source

LSQ

Network	Method	Top-1 Accuracy @ Precision 2	3	4	8	Top-5 Accuracy @ Precision 2	3	4	8
ResNet-18		*Full precision: 70.5*				*Full precision: 89.6*			
	LSQ (Ours)	**67.6**	**70.2**	**71.1**	**71.1**	**87.6**	**89.4**	**90.0**	**90.1**
	QIL	65.7	69.2	70.1					
	FAQ			69.8	70.0			89.1	89.3
	LQ-Nets	64.9	68.2	69.3		85.9	87.9	88.8	
	PACT	64.4	68.1	69.2		85.6	88.2	89.0	
	NICE		67.7	69.8			87.9	89.21	
	Regularization	61.7		67.3	68.1	84.4		87.9	88.2
ResNet-34		*Full precision: 74.1*				*Full precision: 91.8*			
	LSQ (Ours)	**71.6**	**73.4**	**74.1**	**74.1**	**90.3**	**91.4**	**91.7**	**91.8**
	QIL	70.6	73.1	73.7					
	LQ-Nets	69.8	71.9			89.1	90.2		
	NICE		71.7	73.5			90.8	91.4	
	FAQ			73.3	73.7			91.3	91.6
ResNet-50		*Full precision: 76.9*				*Full precision: 93.4*			
	LSQ (Ours)	**73.7**	**75.8**	**76.7**	**76.8**	**91.5**	**92.7**	93.2	**93.4**
	PACT	72.2	75.3	76.5		90.5	92.6	93.2	
	NICE		75.1	76.5			92.3	**93.3**	
	FAQ			76.3	76.5			92.9	93.1
	LQ-Nets	71.5	74.2	75.1		90.3	91.6	92.4	
ResNet-101		*Full precision: 78.2*				*Full precision: 94.1*			
	LSQ (Ours)	**76.1**	**77.5**	**78.3**	**78.1**	**92.8**	**93.6**	**94.0**	**94.0**
ResNet-152		*Full precision: 78.9*				*Full precision: 94.3*			
	LSQ (Ours)	**76.9**	**78.2**	**78.5**	**78.5**	**93.2**	**93.9**	**94.1**	**94.2**
	FAQ			78.4	**78.5**			**94.1**	94.1
VGG-16bn		*Full precision: 73.4*				*Full precision: 91.5*			
	LSQ (Ours)	**71.4**	**73.4**	**74.0**	73.5	**90.4**	**91.5**	**92.0**	**91.6**
	FAQ			73.9	**73.7**			91.7	**91.6**
Squeeze Next-23-2x		*Full precision: 67.3*				*Full precision: 87.8*			
	LSQ (Ours)	**53.3**	**63.7**	**67.4**	**67.0**	**77.5**	**85.4**	**87.8**	**87.7**

image source

LSQ+

LSQ+: Improving low-bit quantization through learnable offsets and better initialization (code)

$$\bar{x} = \left\lfloor clamp\left(\frac{x-\beta}{s}, n, p\right)\right\rceil$$

$$\hat{x} = \bar{x} \times s + \beta$$

$$s_{init} = max(|\mu - 3 * \sigma|, |\mu + 3 * \sigma|)/2^{b-1}$$

where μ and σ are the mean (same as $\langle|w|\rangle$) and standard deviation of the weights in that layer.

$$s_{init}, \beta_{init} = \arg\min_{s,\beta} ||\hat{x} - x||_F^2$$

image source

LSQ+

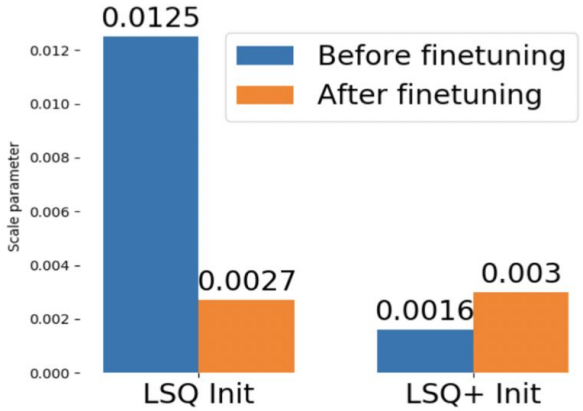


Figure 1. Figure shows the scale parameter of weight quantizer in **blocks.1.conv.0** layer of EfficientNet-B0 before and after finetuning with LSQ and LSQ+ initializations. For both experiments, we used configuration 4 for activation quantization. As shown, LSQ init of the scale is further from the converged value as compared to LSQ+. More on effects of initialization in Sec 4.3

image source

LSQ+

Table 1. Different possible parametrizations for LSQ+'s learnable asymmetric quantization scheme

Configuration	s	β	n	p
Config 1 : Unsigned + Symmetric (LSQ)	trainable	N/A	0	$2^b - 1$
Config 2 : Signed + Symmetric	trainable	N/A	-2^{b-1}	$2^{b-1} - 1$
Config 3 : Signed + Asymmetric	trainable	trainable	-2^{b-1}	$2^{b-1} - 1$
Config 4 : Unsigned + Asymmetric	trainable	trainable	0	$2^b - 1$

image source

LSQ+

Table 4. Comparison of all configurations of quantization with ResNet18 (FP accuracy: 70.1%)

Method	W2A2	W3A3	W4A4
PACT [5]	64.4%	68.1%	69.2%
DSQ [8]	65.2%	68.7%	69.6%
QIL [14]	65.7%	69.2%	70.1%
Config 1 : LSQ (Unsigned + Symmetric)	66.7%	**69.4%**	70.7%
Config 2 : Signed + Symmetric	64.7%	66.1%	69.2%
Config 3 : Signed + Asymmetric	66.7%	**69.4%**	70.7%
Config 4 : Unsigned + Asymmetric	**66.8%**	69.3%	**70.8%**

image source

Layer-wise and Kernel-wise methods

girafe ai

04

ReLeQ

ReLeQ: A Reinforcement Learning Approach for Deep Quantization of Neural Networks

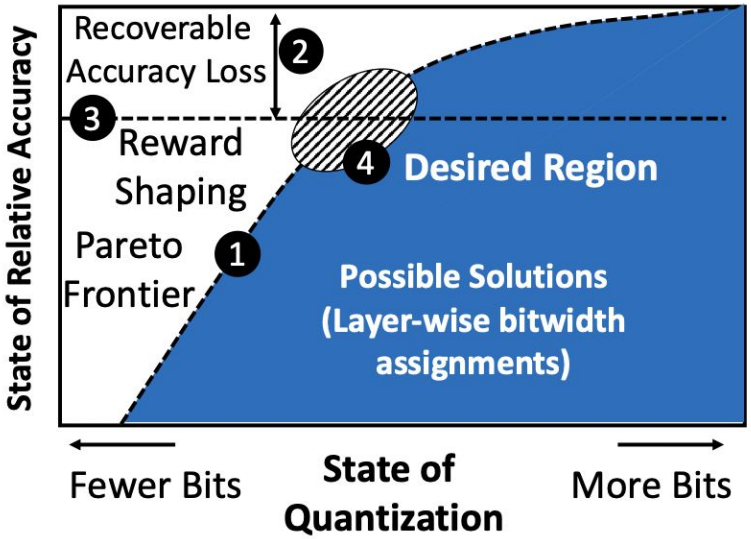


Figure 1. Optimum Automatic Quantization of a Neural Network

image source

ReLeQ

Table 1. Layer and network parameters for state embedding.

	Layer Specific	Network Specific
Static	Layer index Layer Dimensions Weight Statistics (standard deviation)	N/A
Dynamic	Quantization Level (Bitwidth)	State of Quantization State of Accuracy

image source

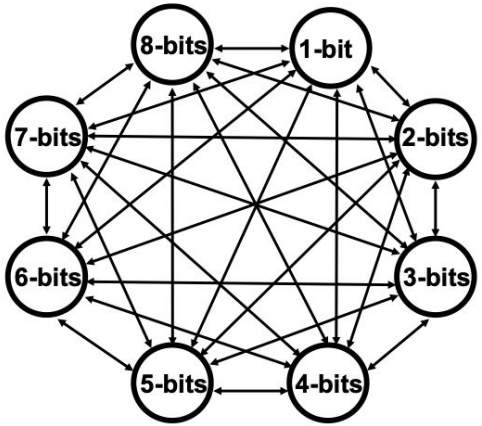
ReLeQ

$$State_{Quantization} = \frac{\sum_{l=0}^{L}[(n_l^w \times \frac{E_{MemoryAccess}}{E_{MAcc}} + n_l^{MAcc}) \times n_l^{bits}]}{\sum_{l=0}^{L}[n_l^w \times \frac{E_{MemoryAccess}}{E_{MAcc}} + n_l^{MAcc}] \times n_{max}^{bits}}$$

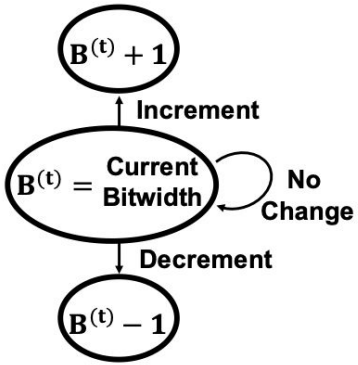
$$State_{Accuracy} = \frac{Acc_{Curr}}{Acc_{FullP}}$$

image source

ReLeQ



(a)



(b)

Reward Shaping:

$reward = 1.0 - (State_{Quantization})^a$

if $(State_{Acc} < th)$ **then**

$reward = -1.0$

else

$Acc_{discount} = max(State_{Acc}, th)^{(b/max(State_{Acc}, th))}$

$reward = reward \times Acc_{discount}$

end if

image source

ReLeQ

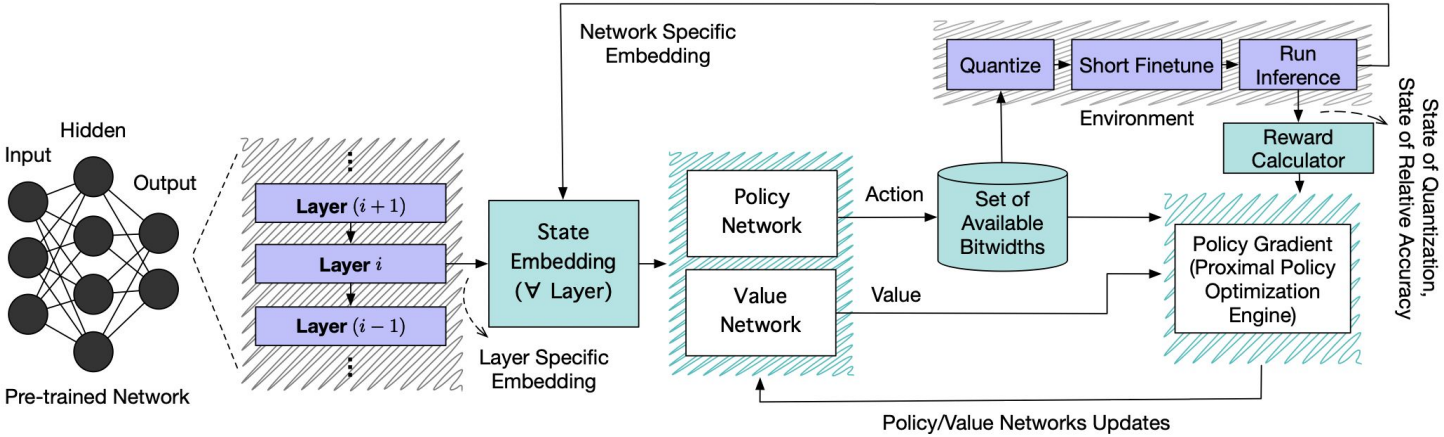


image source

ReLeQ

Table 2. Benchmark DNNs and their deep quantization with **ReLeQ**.

Network	Dataset	Quantization Bitwidths	Average Bitwidth	Acc Loss (%)
AlexNet	ImageNet	{8, 4, 4, 4, 4, 4, 4, 8}	5	**0.08**
SimpleNet	CIFAR10	{5, 5, 5, 5, 5}	5	**0.30**
LeNet	MNIST	{2, 2, 3, 2}	2.25	**0.00**
MobileNet	ImageNet	{8, 5, 6, 6, 4, 4, 7, 8, 4, 6, 8, 5, 5, 8, 6, 7, 7, 7, 6, 8, 6, 8, 8, 6, 7, 5, 5, 7, 8, 8}	6.43	**0.26**
ResNet-20	CIFAR10	{8, 2, 2, 3, 2, 2, 2, 3, 2, 3, 3, 3, 2, 2, 2, 3, 2, 2, 2, 2, 2, 8}	2.81	**0.12**
SVHN-10	SVHN	{8, 4, 4, 4, 4, 4, 4, 4, 4, 8}	4.80	**0.00**
VGG-11	CIFAR10	{8, 5, 8, 5, 6, 6, 6, 6, 8}	6.44	**0.17**
VGG-16	CIFAR10	{8, 8, 8, 6, 8, 6, 8, 6, 8, 6, 8, 6, 8, 6, 8, 8}	7.25	**0.10**

HAQ

HAQ: Hardware-Aware Automated Quantization with Mixed Precision (code)

	Inference latency on HW1	HW2	HW3
Best Q. policy for **HW1**	**16.29** ms	85.24 ms	117.44 ms
Best Q. policy for **HW2**	19.95 ms	**64.29** ms	108.64 ms
Best Q. policy for **HW3**	19.94 ms	66.15 ms	**99.68** ms

Table 1: Inference latency of MobileNet-V1 [12] on three hardware architectures under different quantization policies. The quantization policy that is optimized for one hardware is not optimal for the other. This suggests we need a **specialized** quantization solution for different hardware architectures. (HW1: BitFusion [25], HW2: BISMO [26] edge accelerator, HW3: BISMO cloud accelerator, batch = 16).

image source

HAQ

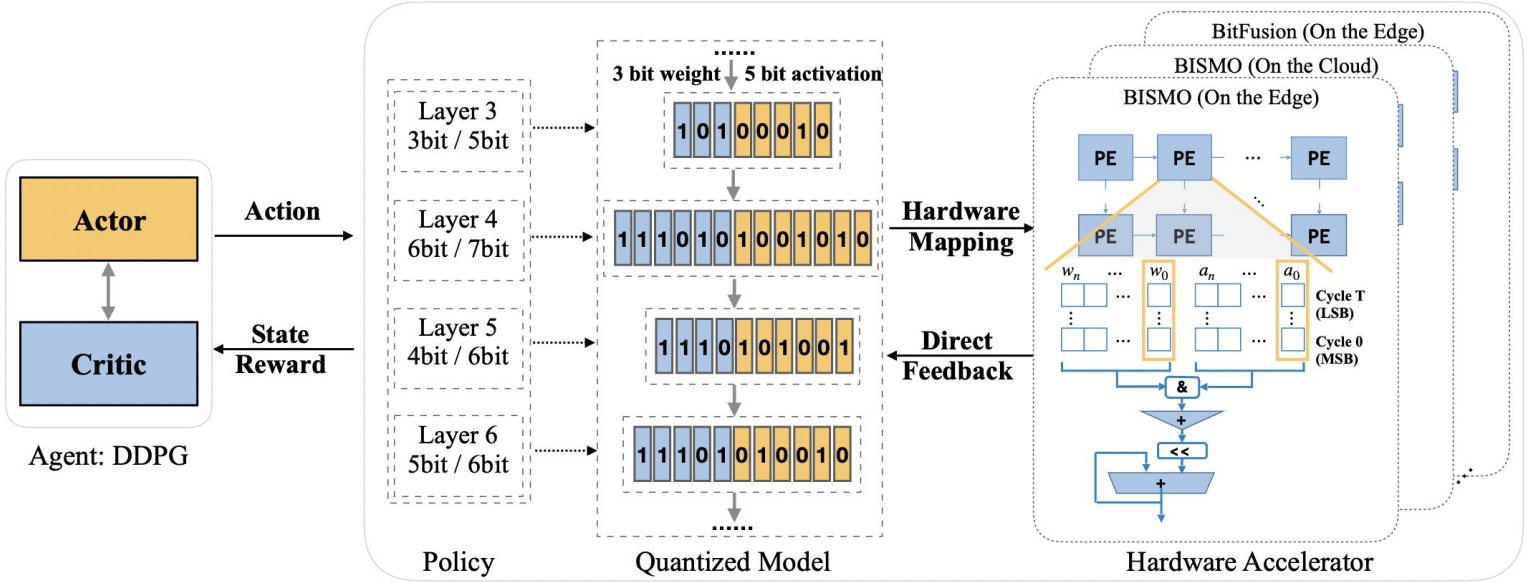


image source

HAQ

$$O_k = (k, c_{\text{in}}, c_{\text{out}}, s_{\text{kernel}}, s_{\text{stride}}, s_{\text{feat}}, n_{\text{params}}, i_{\text{dw}}, i_{\text{w/a}}, a_{k-1}), \tag{1}$$

$$b_k = \text{round}(b_{\text{min}} - 0.5 + a_k \times (b_{\text{max}} - b_{\text{min}} + 1)),$$

$$\mathcal{R} = \lambda \times (\text{acc}_{\text{quant}} - \text{acc}_{\text{origin}}),$$

image source

HAQ

		Edge Accelerator						Cloud Accelerator					
		MobileNet-V1			MobileNet-V2			MobileNet-V1			MobileNet-V2		
	Bitwidths	Acc.-1	Acc.-5	Latency	Acc.-1	Acc.-5	Latency	Acc.-1	Acc.-5	Latency	Acc.-1	Acc.-5	Latency
PACT [2]	4 bits	62.44	84.19	45.45 ms	61.39	83.72	52.15 ms	62.44	84.19	57.49 ms	61.39	83.72	74.46 ms
Ours	*flexible*	**67.40**	**87.90**	45.51 ms	**66.99**	**87.33**	52.12 ms	**65.33**	**86.60**	57.40 ms	**67.01**	**87.46**	73.97 ms
PACT [2]	5 bits	67.00	87.65	57.75 ms	68.84	88.58	66.94 ms	67.00	87.65	77.52 ms	68.84	88.58	99.43 ms
Ours	*flexible*	**70.58**	**89.77**	57.70 ms	**70.90**	**89.91**	66.92 ms	**69.97**	**89.37**	77.49 ms	**69.45**	**88.94**	99.07 ms
PACT [2]	6 bits	70.46	89.59	70.67 ms	71.25	90.00	82.49 ms	70.46	89.59	99.86 ms	71.25	90.00	127.07 ms
Ours	*flexible*	**71.20**	**90.19**	70.35 ms	**71.89**	**90.36**	82.34 ms	**71.20**	**90.08**	99.66 ms	**71.85**	**90.24**	127.03 ms
Original	8 bits	70.82	89.85	96.20 ms	71.81	90.25	115.84 ms	70.82	89.85	151.09 ms	71.81	90.25	189.82 ms

HAQ

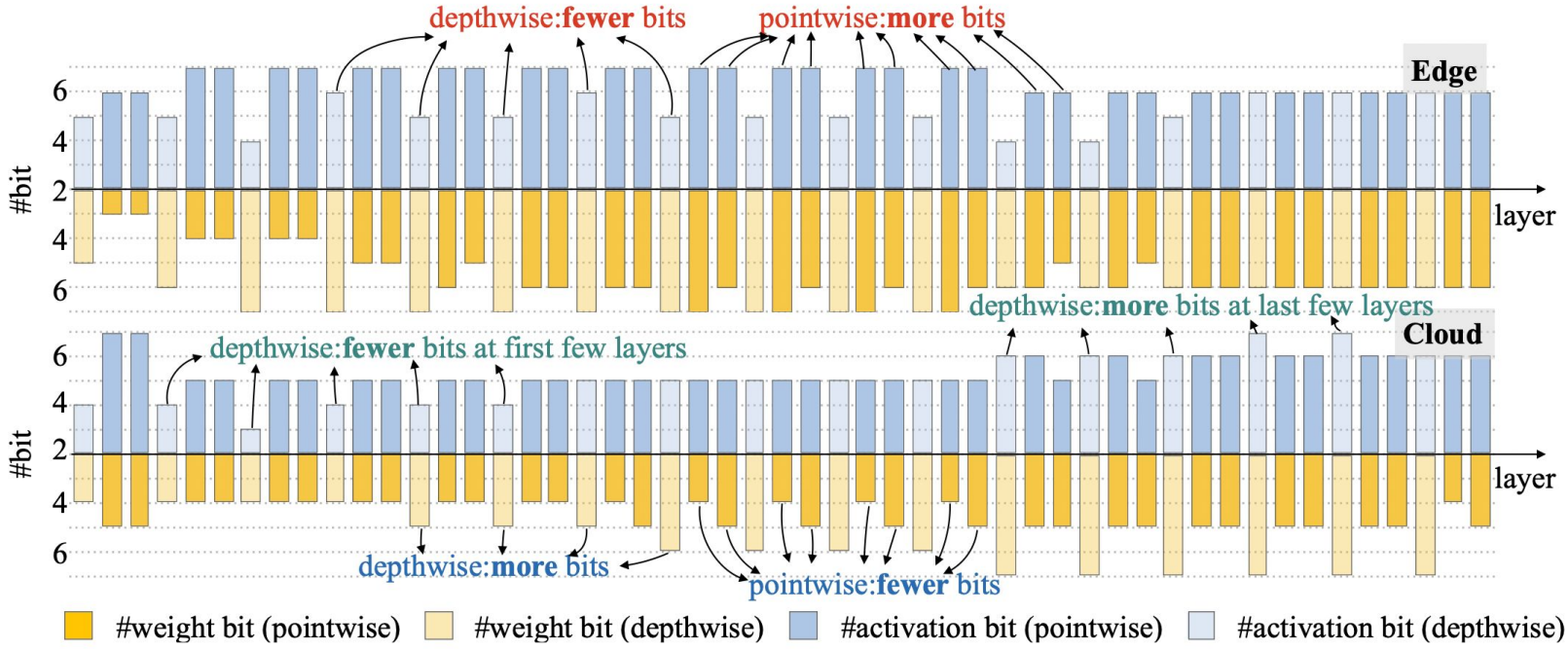


image source

AutoQ

AUTOQ: AUTOMATED KERNEL-WISE NEURAL NETWORK QUANTIZATION

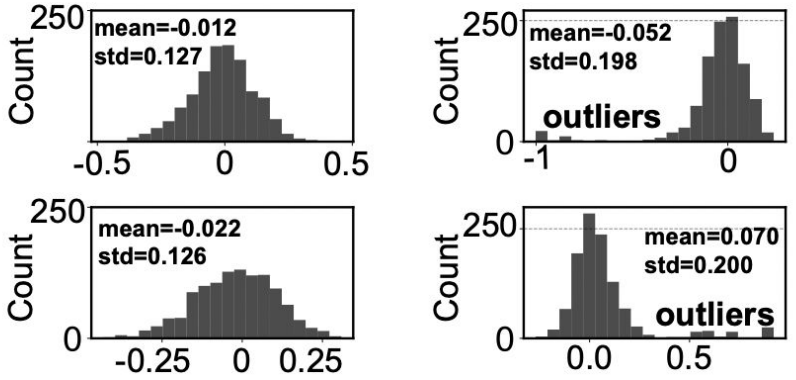


Figure 1: The weight distribution of kernels.

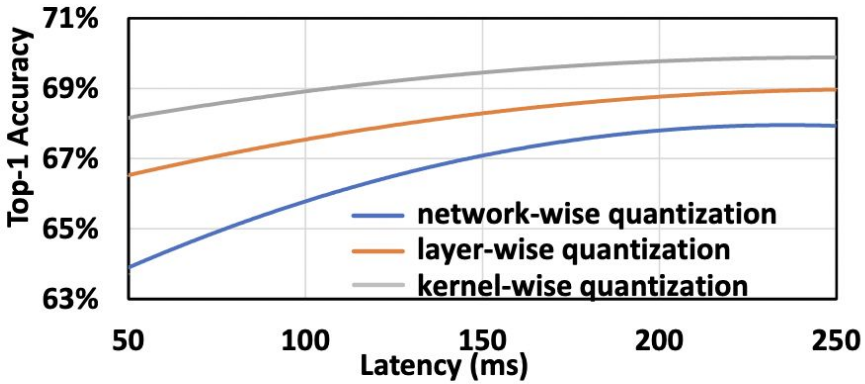


Figure 2: Inference accuracy and latency.

image source

AutoQ

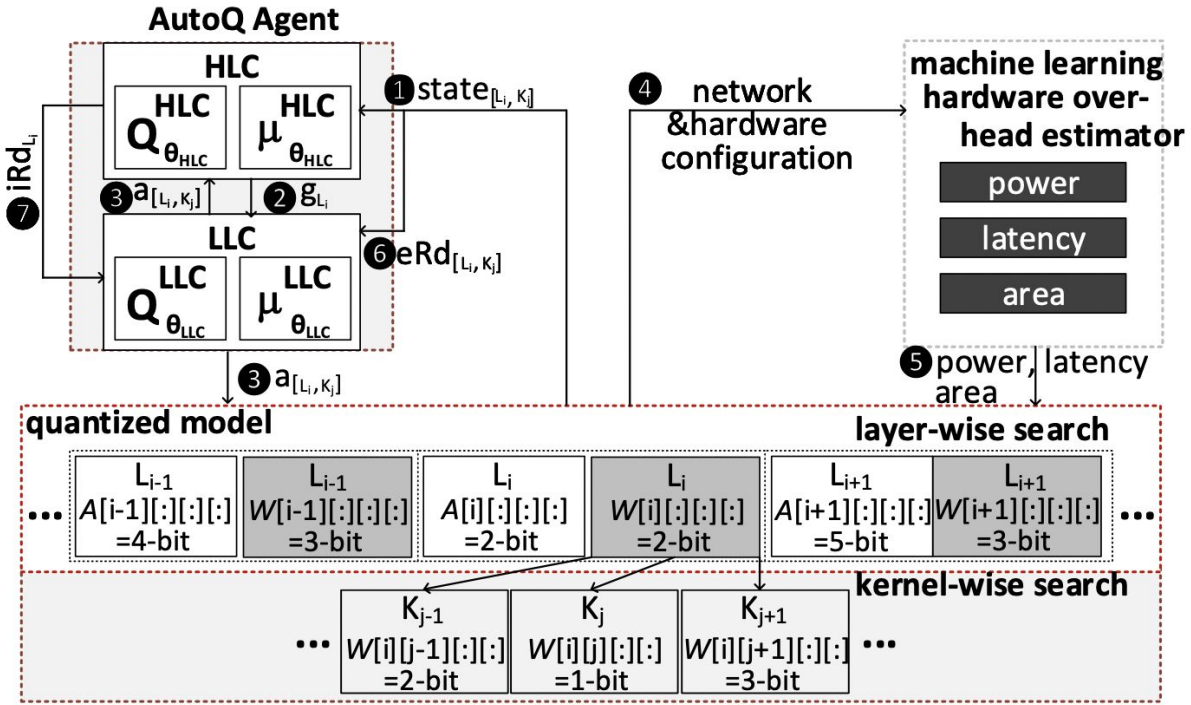


image source

AutoQ

Goal and Action Space. The HLC produces the average QBN for all weight kernels of each layer or the QBN for each activation layer as a goal, while the LLC generates a QBN for each weight kernel in a layer as an action. The HLC goal g_{L_i} for the L_i layer uses a *continuous* space and can be any real value between 1 and $goal_{max}$, where $goal_{max}$ is the maximum average QBN for a layer and we set it to 8. If the L_i layer is an activation layer, we round the real-valued g_{L_i} to the discrete value of $roundup(1 + g_{L_i} \cdot (goal_{max} - 1))$. Although the LLC action is an integer between 0 and $action_{max}$, it still uses a continuous space to capture the relative order, i.e., 2-bit is more aggressive than 3-bit, where $action_{max}$ is the maximum QBN for a kernel and we set it to 8. For the K_j kernel of the L_i layer, the LLC generates the continuous action $ra_{[L_i,K_j]}$ that is in the range of $[0, 1]$, and round it up to the discrete value $a_{[L_i,K_j]} = roundup(ra_{[L_i,K_j]} \cdot action_{max})$.

image source

AutoQ

$$state_{[L_i,K_j]} = (L_i, K_j, c_{in}, c_{out}, s_{kernel}, s_{stride}, s_{feature}, b_{dw}, b_{w/a}, g_{L_{i-1}}, a_{[L_i,K_{j-1}]})$$

$$eRd_{[L_i,K_j]}(NC, HC) = log(\frac{accuracy(NC)^{\psi_{acc}}}{lat(NC, HC)^{\psi_l} \cdot en(NC, HC)^{\psi_e} \cdot area(NC, HC)^{\psi_a}})$$

$$iRd_{L_i} = (1-\zeta) \cdot (-||g_{L_i} \cdot c_{out} - \sum_{j=0}^{c_{out}-1} a_{L_i,K_j}||_2) + \zeta \cdot \sum_{j=0}^{c_{out}-1} eRd_{L_i,K_j}$$

$$(Q^{LLC}_{\theta_{LLC}}(state_{[L_i,K_j]}, g_{L_i}, a_{[L_i,K_j]}) - iRd_{L_i} - \gamma_{iRd} \cdot Q^{LLC}_{\theta_{LLC}}(state_{[L_i,K_{j+1}]}, g_{L_i}, \mu^{LLC}_{\phi_{LLC}}(state_{[L_i,K_{j+1}]}, g_{L_i})))^2$$

exponentially. The HLC converts a series of high-level transition tuples

$$(s_{[L_i,K_0:K_{c_{out}-1}]}, g_{L_i}, a_{[L_i,K_0:K_{c_{out}-1}]}, eRd_{[L_i,K_0:K_{c_{out}-1}]}, s_{[L_{i+1},K_0]})$$

to state-goal-reward transitions

$$(s_{[L_i,K_0]}, g_{L_i}, \sum eRd_{[L_i,K_0:K_{c_{out}-1}]}, s_{[L_{i+1},K_0]})$$

image source

AutoQ

The DRL scheme comparison

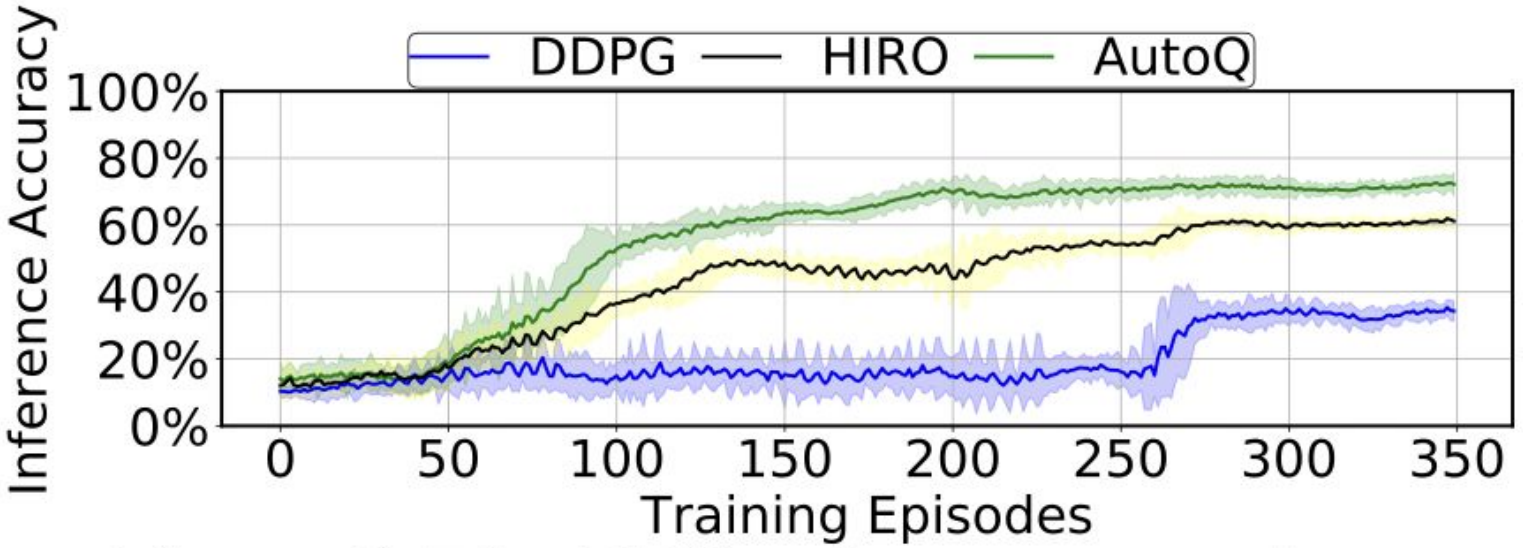


image source

AutoQ

Table 3: Network Quantization by AutoQ (A-QBN: the average QBN of activations; W-QBN: the average QBN of weights; LAT: inference latency).

model	scheme	resource-constrained					accuracy-guaranteed				
		top-1 err (%)	top-5 err(%)	A-QBN (bit)	W-QBN (bit)	LAT (ms)	top-1 err (%)	top-5 err(%)	A-QBN (bit)	W-QBN (bit)	LAT (ms)
ResNet-18	network-wise	32.7	12.32	4	4	296.8	32.7	12.32	4	4	296.8
	layer-wise	31.8	11.92	3.32	4.63	290.9	32.5	11.90	3.37	3.65	189.6
	kernel-wise	**30.22**	**11.62**	4.12	3.32	286.3	32.6	11.82	**3.02**	**2.19**	125.3
	original	30.10	11.62	16	16	1163	30.10	11.62	16	16	1163
ResNet-50	network-wise	27.57	9.02	4	4	616.3	27.57	9.02	4	4	616.3
	layer-wise	26.79	8.32	4.23	3.51	612.3	27.49	9.15	4.02	3.12	486.4
	kernel-wise	**25.53**	**7.92**	3.93	4.02	610.3	27.53	9.12	**3.07**	**2.21**	327.3
	original	25.20	7.82	16	16	2357	25.20	7.82	16	16	2357
SqueezeNetV1	network-wise	45.67	23.12	4	4	43.1	45.67	23.12	4	4	43.1
	layer-wise	44.89	21.14	3.56	4.27	42.1	45.63	23.04	3.95	3.28	25.5
	kernel-wise	**43.51**	**20.89**	4.05	3.76	41.6	45.34	23.02	**3.29**	**2.32**	12.5
	original	43.10	20.5	16	16	127.3	43.10	20.5	16	16	127.3
MobileNetV2	network-wise	31.75	11.67	4	4	37.4	31.35	11.67	4	4	37.4
	layer-wise	30.98	10.57	3.57	4.22	36.9	31.34	10.57	3.92	3.21	23.9
	kernel-wise	**29.20**	**9.67**	4.14	3.67	36.1	31.32	11.32	**3.13**	**2.26**	10.2
	original	28.90	9.37	16	16	123.6	28.90	9.37	16	16	123.6

image source

Summary

1. Recap on Quantization
2. Quantization granularity
3. Quantization techniques
    a. PACT
    b. LSQ
    c. LSQ+
4. Layer-wise and kernel-wise methods
    a. ReLeQ
    b. HAQ
    c. AutoQ

Thanks for attention!

Questions? Additions? Welcome!

---

girafe
ai