商汤 x 智东西公开课

可部署量化感知训练算法研究

Deployable Quantization-aware Training

李雨杭
Yale & SenseTime
2021年9月9日星期四

Contents

目录

Contents

目录

Recap -- Quantization

模型压缩的目标：让模型变小变快还尽可能保证精度不降?

Quantization

需要硬件和优化
大概率变快
有精度损失风险
被主流硬件厂商支持
生产成本可高可低

Sparsity

Unstructured

需要硬件和优化
大概率变快
有精度损失风险
被少数硬件厂商支持
生产成本较高

Structured

不需硬件特殊设计
不一定变快
有精度损失风险
生产成本较高

NAS

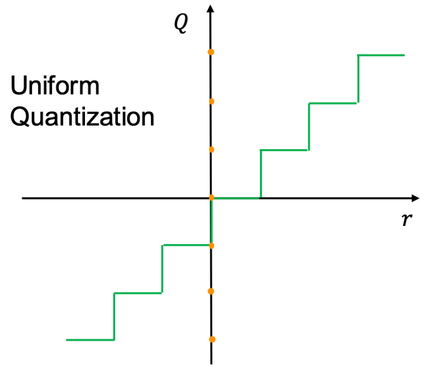
不需硬件特殊设计
大概率变快
有精度损失风险
生产成本较高

Distillation

与加速无关
主要用于提升精度

Recap -- Quantization

- **均匀量化**
  - **二值化**
    - xnor + popcount理论峰值比float32高
    - 引入额外的quantizer，可用SIMD方式加速
  - **线性量化(对称、非对称、Ristretto)**
    - arm/x86/nvGPU均支持高效的8-bit计算，TensorCore支持4bit计算
    - 引入额外的quantizer/de-quantizer，可用SIMD方式加速
- **非均匀量化**
  - **对数量化**
    - 可将乘法转变为加法，加法转变为索引
  - 其他



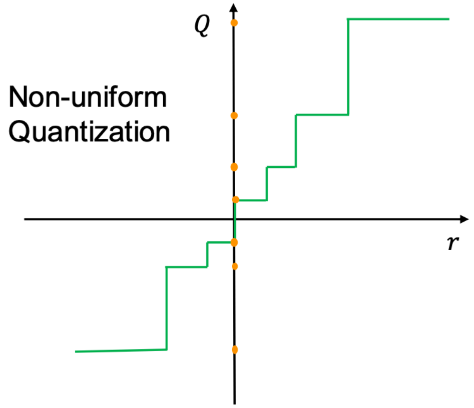


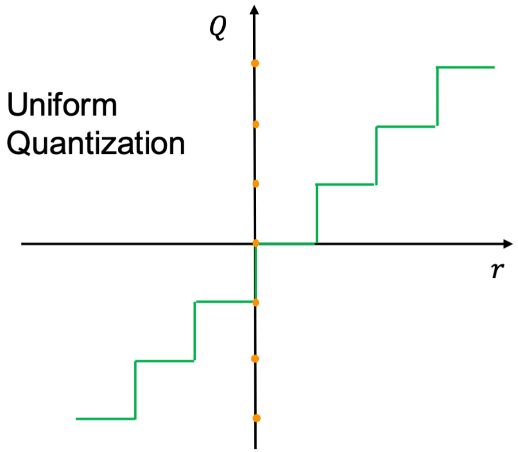
Figure taken from Gholami et al., 2021, A Survey of Quantization Methods for Efficient Neural Network Inference

Quantization Function

$$\bar{\boldsymbol{w}} = \text{clip}(\left\lfloor \frac{\boldsymbol{w}}{s} \right\rceil + z, N_{min}, N_{max}), \quad \hat{\boldsymbol{w}} = s \cdot (\bar{\boldsymbol{w}} - z)$$

Quantize：将全精度参数w映射到整数

De-Quantize：将整数映射到全精度范围



Quantization Algorithms

- DoReFa-Net
- Parameterized Activation Threshold (PACT)
- Quantization Interval Learning
- Learned Step Size Quantization

特点：　(1）都量化了网络权重和输入激活值，并且第一层和最后一层为8bit 或者 全精度。
(2）都采用了 per-tensor, symmetric 量化，对激活值采用unsigned 量化
(3）无法真正部署到硬件上
(4）采用了不同的训练配置，结果比较并不公平

DoReFa-Net

- Basic quantize operation:

$$\textbf{Forward: } r_o = \frac{1}{2^k - 1}\,\mathrm{round}((2^k - 1) r_i)$$

$$\textbf{Backward: } \frac{\partial c}{\partial r_i} = \frac{\partial c}{\partial r_o}.$$

- Weight quantization:

$$\textbf{Forward: } r_o = f_\omega^k(r_i) = 2\,\mathrm{quantize}_\mathrm{k}\left(\frac{\tanh(r_i)}{2\max(|\tanh(r_i)|)} + \frac{1}{2}\right) - 1.$$

$$\textbf{Backward: } \frac{\partial c}{\partial r_i} = \frac{\partial r_o}{\partial r_i}\frac{\partial c}{\partial r_o} \quad 4$$

- Activation quantization:

$$f_\alpha^k(r) = \mathrm{quantize}_\mathrm{k}(r).$$

Zhou et al., 2016, DoReFa-Net: Training Low Bitwidth Convolution Neural Network with Low Bitwidth Gradients.

Parameterized Clipping Activation

Weight quantization: Same as DoReFa-Net

Activation:

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

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

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

Quantization Interval Learning

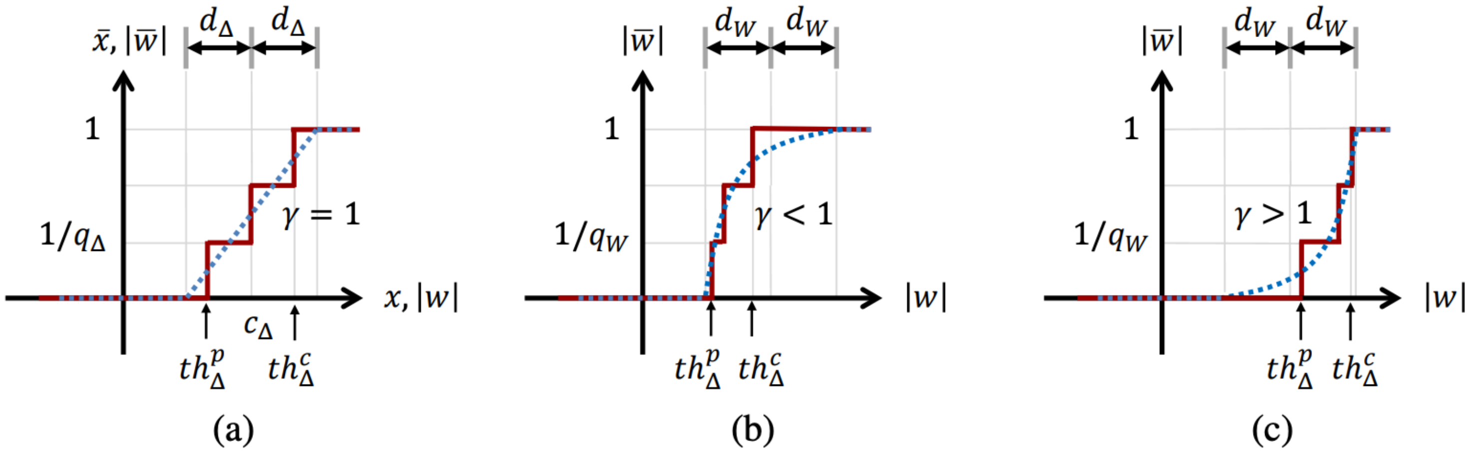
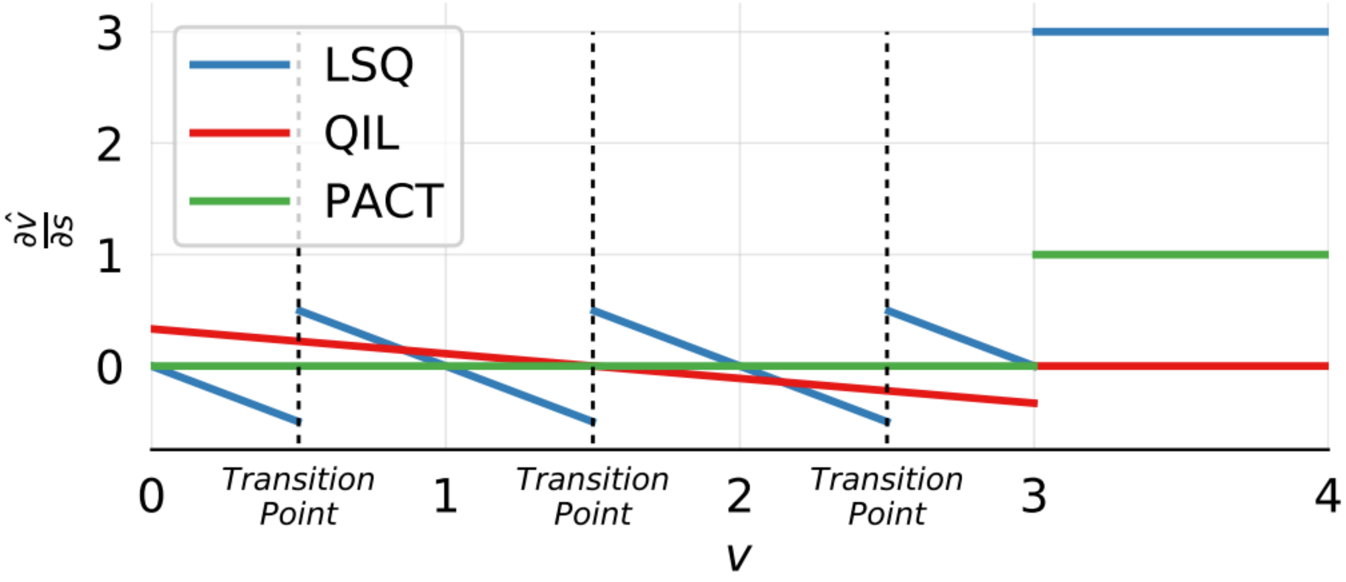


Figure 2. A quantizer as a combination of a transformer and a discretizer with various γ where (a) $\gamma = 1$, (b) $\gamma < 1$, and (c) $\gamma > 1$. The blue dotted lines indicate the transformers, and the red solid lines are their corresponding quantizers. The th_Δ^p and the th_Δ^c represent the pruning and clipping thresholds, respectively.

$$\hat{w} = \begin{cases} 0 & |w| < c_W - d_W \\ \mathrm{sign}(w) & |w| > c_W + d_W \\ (\alpha_W |w| + \beta_W)^\gamma \cdot \mathrm{sign}(w) & otherwise, \end{cases}$$

$$\hat{x} = \begin{cases} 0 & x < c_X - d_X \\ 1 & x > c_X + d_X \\ \alpha_X x + \beta_X & otherwise, \end{cases}$$

Learned Step Size Quantization



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

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

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

商汤 sensetime

Contents

目录

可部署的量化模型与学术论文量化模型的区别

- 量化函数配置不一致
  - 在学术论文中，普遍的方法是选择 per-tensor，symmetric量化函数，然而在实际硬件中存在多种配置，每一种硬件会定义自己的量化函数配置。
  - 一些量化算法可能在per-channel时表现并不好。
  - 学术论文通常考虑激活值量化函数是无符号的，然而真实硬件没有定义量化值的符号
- 激活值量化插入点不一致：
  - 在学术论文中，对激活值一般只量化卷积层的输入；然而在真实硬件中，会对elemental-wise add等操作进行量化。
- 未考虑吧BN层折叠
  - 在学术论文未考虑BN层的折叠，然而在真实硬件中所有的BN应该先被折叠进卷积层再发生量化。

量化函数

$$\bar{\boldsymbol{w}} = \mathrm{clip}(\left\lfloor \frac{\boldsymbol{w}}{s} \right\rceil + z, N_{min}, N_{max}), \quad \hat{\boldsymbol{w}} = s \cdot (\bar{\boldsymbol{w}} - z)$$

- Symmetric量化： 零点z始终为0，而asymmetric量化中z可以为其他整数
- Per-tensor量化：w矩阵中使用一个步长和零点，perchannel会对每个w_i分配一个步长和零点
- s的格式：通常情况下s是FP32，但是部分硬件会使用2次幂的步长，即s=2^n
- 无符号量化：学术集为了处理ReLU激活值全部非负的情况，将Nmin设置为0，Nmax设置为2^b-1，这样就不会浪费一般的整数范围。然而真实硬件通常不会这样定义，只存在对称和非对称的设置。

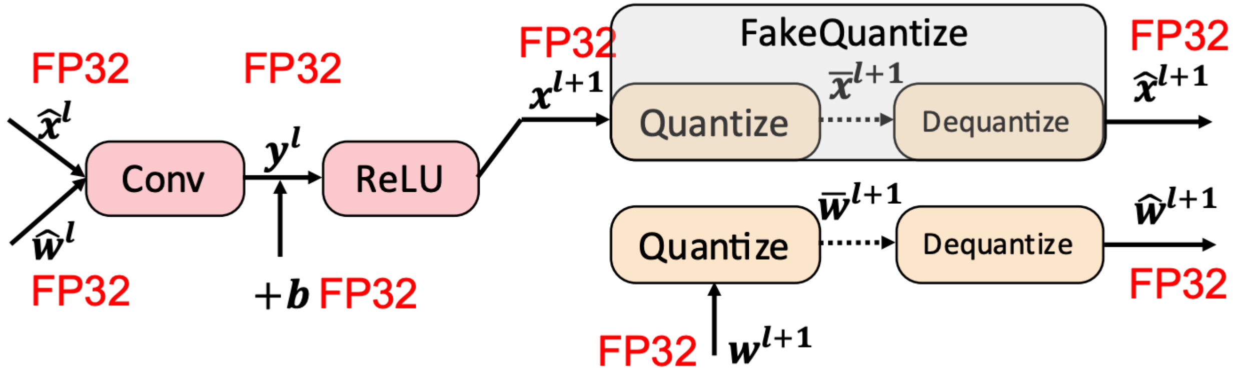
量化函数

$$\bar{w} = \mathrm{clip}\left(\left\lfloor \frac{w}{s} \right\rceil + z, N_{min}, N_{max}\right), \quad \hat{w} = s \cdot (\bar{w} - z)$$

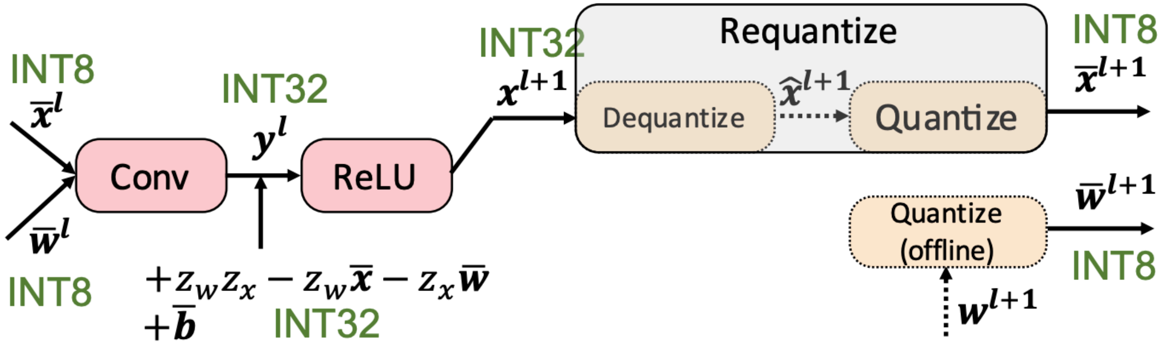
Table 2: Comparison of (1) the different hardware we selected and (2) the different QAT algorithms. *Infer. Lib.* is the inference library; *FBN* means whether fold BN. ✱ means undeployable originally, but can be deployable when certain requirements are satisfied.

Infer. Lib.	Provider	HW Type	Hardware	s Form.	Granularity	Symmetry	Graph	FBN
TensorRT [22]	NVIDIA	GPU	Tesla T4/P4	FP32	Per-channel	Symmetric	2	✓
ACL [23]	HUAWEI	ASIC	Ascend310	FP32	Per-channel	Asymmetric	1	✓
TVM [25]	OctoML	CPU	ARM	POT	Per-tensor	Symmetric	3	✓
SNPE [24]	Qualcomm	DSP	Snapdragon	FP32	Per-tensor	Asymmetric	3	✓
FBGEMM [26]	Facebook	CPU	X86	FP32	Per-channel	Asymmetric	3	✓

量化节点



(a) Fake quantization for simulated-training.



(b) Real quantization in deployments.

量化节点

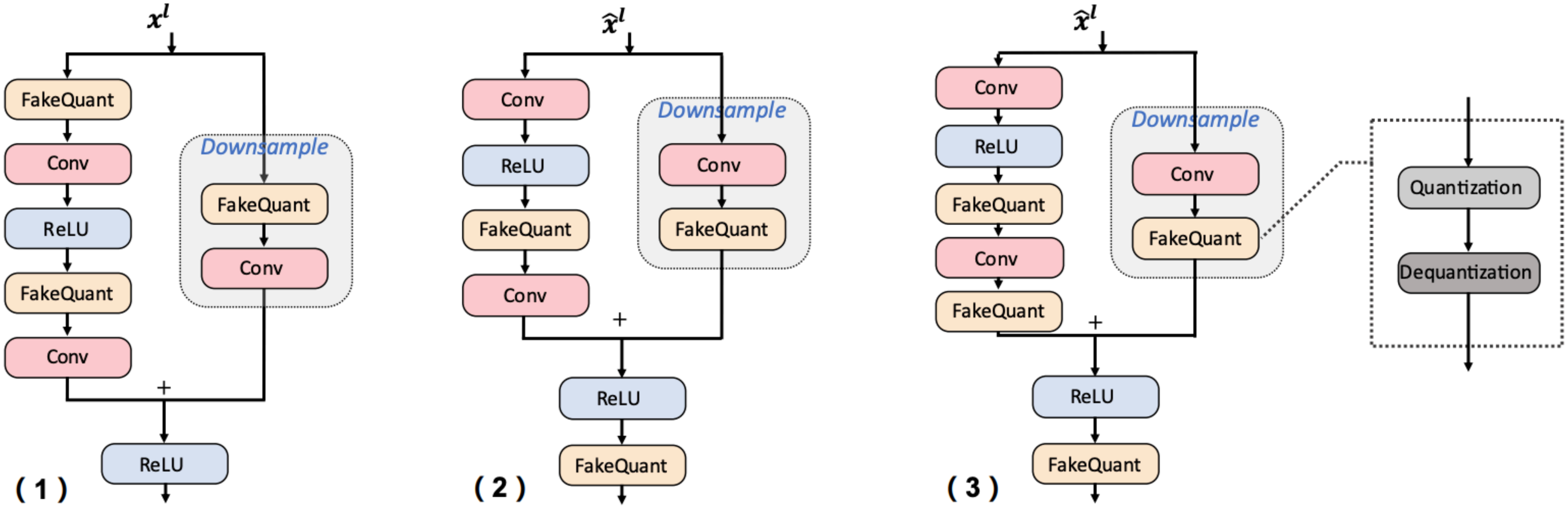


Figure 4: Comparison of different quantization implementations for a basic block in the ResNet [17].

BN层折叠

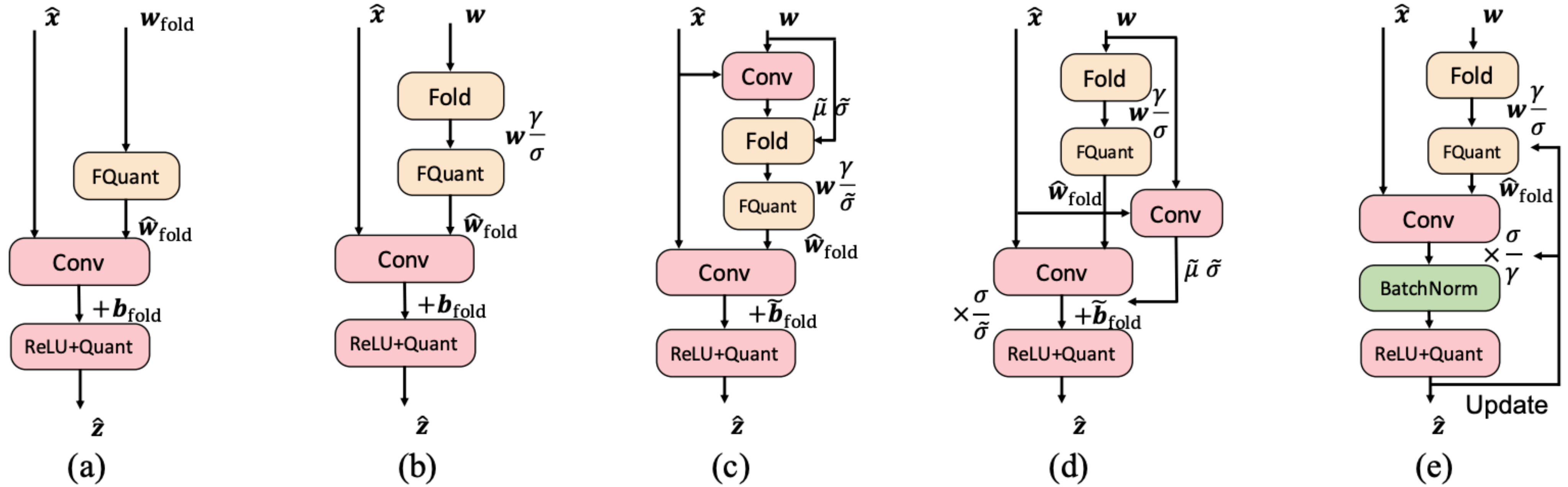


Figure 3: Comparison of different Batch Normalization folding technologies. (a) Removing BN layers and directly update w_{fold}. (b) BN folding without any statistics update. (c) BN folding with two convolutions. (d) folding with running statistics and also requires two convolutions. (e) folding running statistics with an explicitly BN layer in training. *Graph (bcde) can be transformed to (a) during inference.* FQuant=FakeQuantize.

Contents

目录

可复现的量化模型

- 已有的学术论文无法复现精度，主要因为：
  - 部分算法代码没开源
  - 使用的训练配置不一样，具体有：
    - 使用全精度预训练模型初始化
    - 使用不同的LR scheduler
    - 使用不同的batchsize，训练时长
    - 使用不同的weight decay，etc
  - 随着软件库的发展，也有进步。比如现在使用pytorch训练resnet-18很容易超过70%精度，高于论文的69.8。

可复现的量化模型

- MQBench采用统一的配置，统一的预训练模型初始化，消除了训练偏差。

Table 3: Training hyper-parameters. *Batch Size* is the batch size per GPU. * means 0 weight decay for BN parameters.

Model	LR	$L2$ Reg.	Batch Size	# GPU
ResNet-18	0.004	10^{-4}	64	8
ResNet-50	0.004	10^{-4}	16	16
EffNet&MbV2	0.01	$10^{-5}*$	32	16
RegNet	0.004	4×10^{-5}	32	16

Contents

目录

如何获得实质的推理性能提升?

必要条件2: 软件推理库优化

有了指令和架构支持还不够，还需要有对应的软件实现

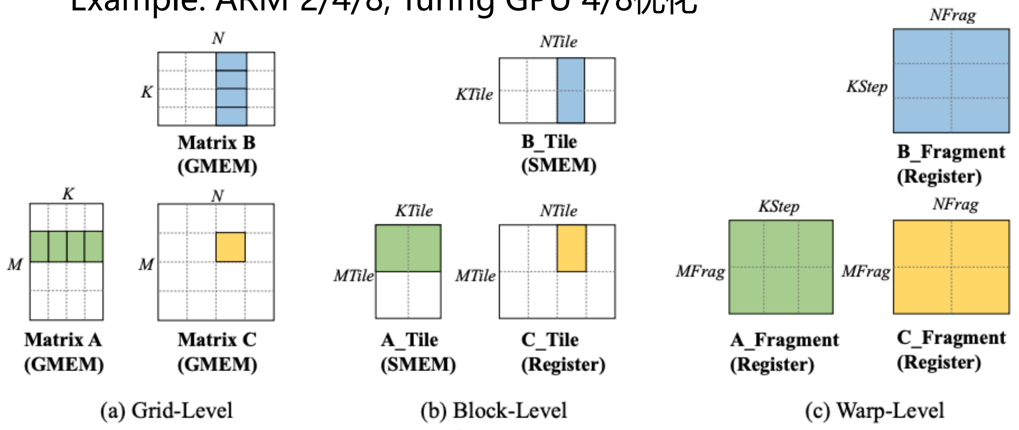
NV Turing架构有int4指令支持，就能跑了吗？ ✕

一些可用的开源/闭源推理库举例：

https://developer.nvidia.com/zh-cn/tensorrt
https://github.com/alibaba/MNN
https://github.com/Tencent/ncnn
https://github.com/OAID/Tengine

可行路径1: 手工优化

Example: ARM 2/4/8, Turing GPU 4/8优化



Extremely Low-bit Convolution Optimization for Quantized Neural Network on Modern Computer Architectures, ICPP20

可行路径2: 编译优化

Example: ViT cuda int8优化 （单位：ms）
Intel(R) Xeon(R) Gold 6246 CPU @ 3.30GHz， Tesla T4

模型	TRT7.1 FP32	TRT7.1 INT8	Ours INT8	Speed Up
patch16	12.13	11.85	6.92	1.71
patch32	4.54	4.61	3.18	1.45

已经被TVM社区merge：https://github.com/apache/tvm/pull/7814

可行路径3: 等待硬件公司的推理库. 最常见

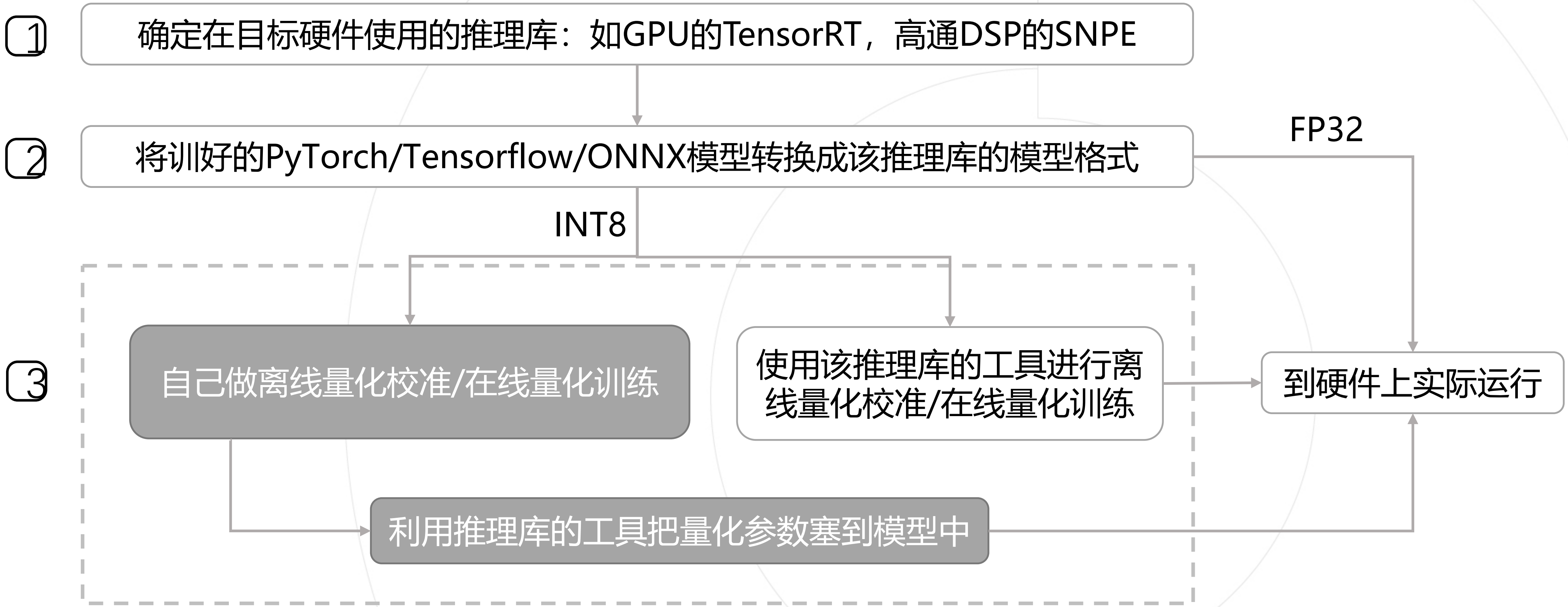
如何获得实质的推理性能提升?

量化硬件及相关推理库概览

硬件	公司	推理库	比特数	量化方案
GPU	NVIDIA	TensorRT	8	线性对称per channel
			FP16	IEEE 754
		NART-QUANT	4/8	线性对称per layer/channel
3559/3519/3516	Hisilicon	NNIE	8/16	对数量化
Ceva DSP	Ceva	-	8/16	线性非对称per layer/channel
Hexagon DSP	Qualcomm	SNPE	8	线性非对称per layer
Adreno 5/6 serial	Qualcomm	OCL	FP16	IEEE 754 without Subnormal
ARM	ARM	NART-QUANT	2-8	线性非对称per layer
WUQI	WUQI tech.	WUQI sdk	8/16	Ristretto
SigmaStar	SigmaStar Technology	SigmaStar sdk	8/16	线性weight对称(per channel)，activation非对称
Ascend 310	HUAWEI	ACL	8	线性非对称per channel
			FP16	IEEE 754
Ambarella	Ambarella	CVFlow	8/16	Ristretto
			FP16	IEEE 754

如何生产一个硬件能跑的量化模型?

硬件能跑需要做哪些工作?

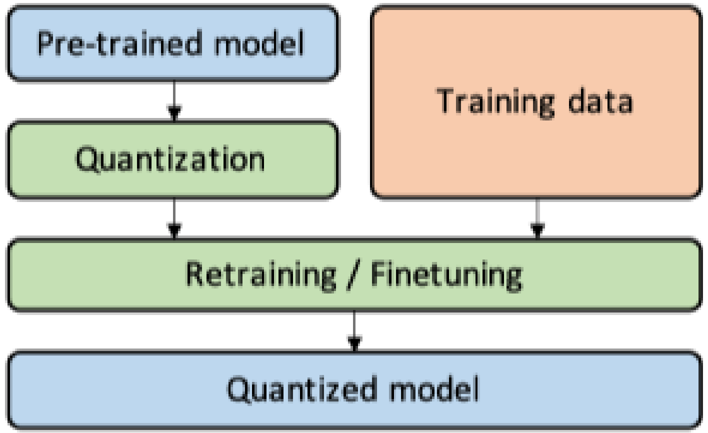


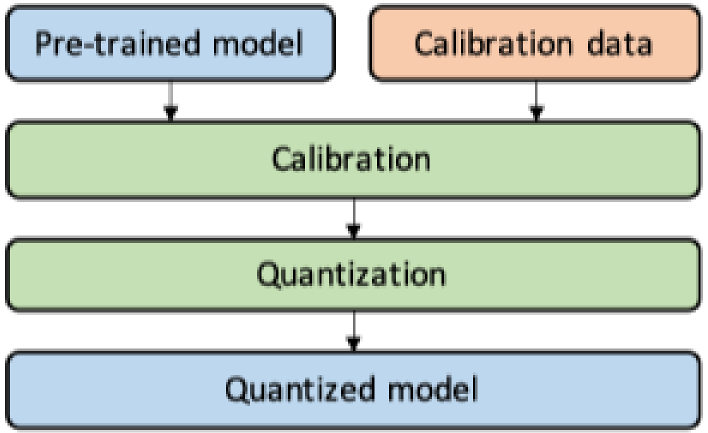
如何挽救精度损失?

手段1: 离线量化（后训练量化） — Post Training Quantization (PTQ)

手段2: 量化感知训练 — Quantization Aware Training (QAT)

	算法位置	所需数据量	是否需要训练	时间	能力上限	上手复杂度
PTQ	贴近部署	不需/少量校准数据	☒	几分钟	可以解决8-bit的大部分问题，部分hard case会掉精度	低，几乎一个命令行
QAT	贴近训练	大规模训练数据	☑	几小时or几天	用于解决8-bit的hard case和更低bit的模型	高，需要侵入训练代码





Gholami et al., 2021, A Survey of Quantization Methods for Efficient Neural Network Inference

Contents
目录

基本的离线量化方法

解决的主要问题：给定一个Tensor，如何确定其截断范围

考虑该Tensor本身的量化误差：

- Min Max：直接取最大最小
- Histogram：做直方图统计，分bin，从两边开始向中间收缩，确定最合适的bin作为截断范围
- Percentile：使用分位点作为截断值
- Mean squared error：使用最小均方误差，优化得到截断值
- KL Divergence：使用KL散度，优化得到截断值
- Cross Entropy：使用交叉熵，优化得到截断值，适用于最后一层的logits

硬件和推理库离线量化工具的能力

- 我们所接触过的18种深度学习硬件，全都支持基本的离线量化校准
  - 这是因为离线量化是最快而且可以完全独立于训练的方式，最容易支持，使用方式也最容易被人接受

- 有的甚至把好的离线量化方案作为推销硬件的卖点

——“这里面的算法是我们经过很多调试和验证总结出来的，绝对能保证精度！”

- 然然，由于硬件厂商离应用和业务较远，大部分验证都是在学术模型和数据上做的，因此结论会存在一定的偏差。掉点还是经常发生！ 

离线量化难点

- 校准数据有限
  - 夜间、白天
  - RGB/近红外数据domain
  - …
- 异常数据分布
  - weight分布异常（特别大的weight）
- 优化空间有限
  - 之前只能调整量化参数
  - 现在可以微调weight
- 优化粒度选择
  - Layer by layer?
  - End to end?

离线量化能追平量化感知训练吗?

Contents

目录

MQBench Results: Academic Setting

Table 4: Academic setting benchmark for 4-bit quantization-aware training, result in bracket is the reported accuracy in original paper. "NC" denotes not converged.

Model	LSQ [15]	APoT [27]	QIL [18]	DSQ [28]	PACT [30]	DoReFa [31]
ResNet-18	70.7 (71.1)	70.5 (70.7)	**70.8** (70.1)	70.0 (69.6)	70.5 (69.2)	70.7 (68.1)
ResNet-50	**77.4** (76.7)	77.1 (76.6)	77.2 (N/A)	76.4 (N/A)	76.3 (76.5)	76.4 (71.4)
MobileNetV2	70.6 (66.3)	68.6 (N/A)	70.3 (N/A)	69.6 (64.8)	**70.7** (61.4)	NC (N/A)
EfficientNet-Lite0	72.6 (N/A)	70.0 (N/A)	72.7 (N/A)	72.6 (N/A)	**73.0** (N/A)	NC (N/A)
RegNetX-600MF	72.7 (N/A)	**73.0** (N/A)	71.6 (N/A)	71.7 (N/A)	72.2 (N/A)	72.9 (N/A)
Avg. Arch.	**72.8**	71.9	72.5	72.1	72.5	44.0

MQBench Results： BN Folding

商汤 sensetime

Table 5: Comparison of the accuracy on a 4-bit quantized ResNet-18 and MobileNetV2, using LSQ [15] and PACT [30], given different folding strategies ("-1" denotes normal BN training without folding, others are folding strategies introduced in Sec. 3.2.); "NC" denotes Not Converged; "*" denotes asynchronous statistics.

Model	ResNet-18							MobileNetV2						
Folding Strategy	-1	0	1	2	3	4	4*	-1	0	1	2	3	4	4*
LSQ	70.7	69.8	70.1	70.2	70.3	**70.4**	70.1	70.6	69.5	69.9	70.0	**70.1**	**70.1**	64.8
PACT	70.5	NC	NC	NC	NC	**67.8**	65.5	70.7	NC	NC	NC	NC	**60.8**	NC

MQBench Results: Hardware Setting

Table 7: 4-bit Quantization-Aware Training benchmark on the ImageNet dataset, given different algorithms, hardware inference libraries, and architectures. "NC" means not converged. Red and Green numbers denotes the decrease and increase of the hardware deployable quantization.

Model	Method	Paper Acc.	Academic	TensorRT	ACL	TVM	SNPE	FBGEMM	Avg. HW
ResNet-18 FP: 71.0	LSQ [15]	71.1 / 70.7[1]	**70.7**	69.3(1.4)	70.2(0.5)	67.7(3.0)	**69.7(1.0)**	**69.8**(0.9)	69.3±0.87
	DSQ [28]	69.6	70.0	66.9(3.1)	69.7(0.3)	67.1(2.9)	68.9(1.1)	68.9(1.1)	68.3±1.10
	PACT [30]	69.2	70.5	69.1(1.4)	**70.4(0.1)**	57.5(13.0)	69.3(1.2)	69.7(**0.8**)	67.2±4.87
	DoReFa [31]	68.1[2]	**70.7**	**69.6(1.1)**	**70.4**(0.3)	**68.2(2.5)**	68.9(1.8)	69.7(1.0)	**69.4±0.75**
ResNet-50 FP: 77.0	LSQ [15]	76.7	**77.4**	**76.3**(1.1)	**76.5**(0.9)	**75.9(1.5)**	**76.2**(1.2)	76.4(1.0)	**76.3±0.21**
	DSQ [28]	N/A	76.4	74.8(1.6)	76.2(0.2)	74.4(2.0)	75.9(**0.5**)	76.0(0.4)	75.5±0.72
	PACT [30]	76.5	76.3	**76.3(0.0)**	76.1(0.2)	NC	NC	**76.6(0.3)**	45.8±37.4
	DoReFa [31]	71.4[2]	76.4	76.2(0.2)	76.3(**0.1**)	NC	NC	75.9(0.5)	45.7±37.3
MobileNetV2 FP: 72.6	LSQ [15]	66.3[3]	70.6	66.1(4.5)	68.1(2.5)	**64.5(6.1)**	**66.3(4.3)**	65.5(5.1)	**66.1±1.18**
	DSQ [28]	64.8	69.6	48.4(21.2)	68.3(1.3)	29.4(39.8)	41.3(28.3)	50.7(18.9)	47.6±12.7
	PACT [30]	61.4[4]	**70.7**	**66.5(4.2)**	**70.3(0.4)**	48.1(22.6)	60.3(10.4)	**66.5(4.2)**	62.3±7.8
	DoReFa [31]	N/A	NC	NC	NC	NC	NC	NC	0±0
EfficientNet-Lite0 FP: 75.3	LSQ [15]	N/A	72.6	67.0(5.6)	65.5(7.1)	**65.0(7.6)**	**68.6(4.0)**	66.9(6.7)	**66.6±1.27**
	DSQ [28]	N/A	72.6	35.1(37.5)	69.6(3.0)	NC	7.5(65.1)	45.9(26.7)	31.6±25.5
	PACT [30]	N/A	**73.0**	**68.2(4.8)**	**72.6(0.4)**	45.9(27.1)	56.5(16.5)	**69.0(4.0)**	62.4±9.88
	DoReFa [31]	N/A	NC	NC	NC	NC	NC	NC	0±0
RegNetX-600MF FP: 73.7	LSQ [15]	N/A	72.7	**72.5(0.2)**	72.8(0.1)	**70.0(2.7)**	**72.5(0.2)**	**72.5**(0.2)	**72.1±1.04**
	DSQ [28]	N/A	71.7	68.6(2.1)	71.4(0.3)	64.5(7.2)	70.0(1.7)	70.0(1.7)	68.9±2.37
	PACT [30]	N/A	72.2	72.0(**0.2**)	**73.3(1.1)**	NC	NC	**72.5(0.3)**	43.6±35.5
	DoReFa [31]	N/A	**72.9**	72.4(0.5)	73.2(0.3)	NC	NC	72.2(0.7)	43.6±35.6

[1,2,3,4] Accuracy reported in [30, 42, 43, 44], respectively.

Thanks for listening!

Q&A Time