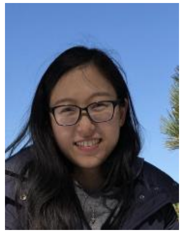
Efficient Neural Networks through Systematic Quantization

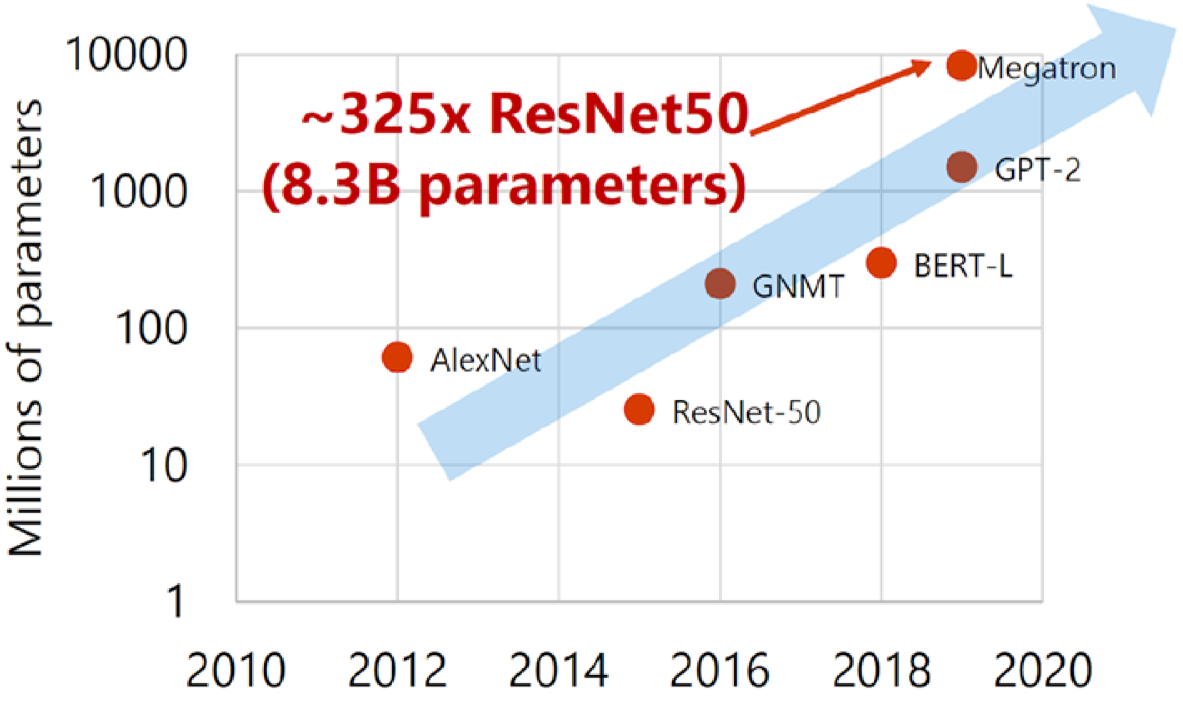
Zhen Dong, Zhewei Yao, Amir Gholami, Zhangcheng Zheng, Eric Tan, Daiyaan Arfeen,
Sheng Shen, Qijing Huang, Michael Mahoney, Kurt Keutzer

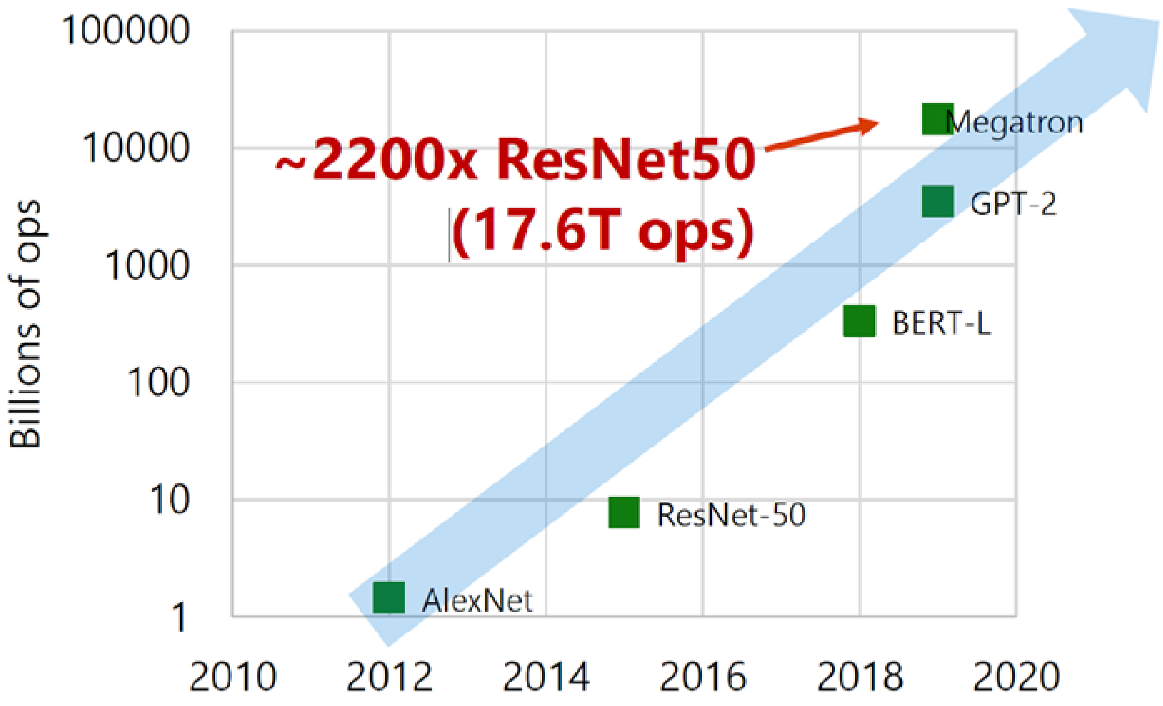
         

Outline

- Introduction
- Hessian-AWare Quantization
- Automated Mixed-Precision
- Hardware-Aware Deployment
- Conclusion

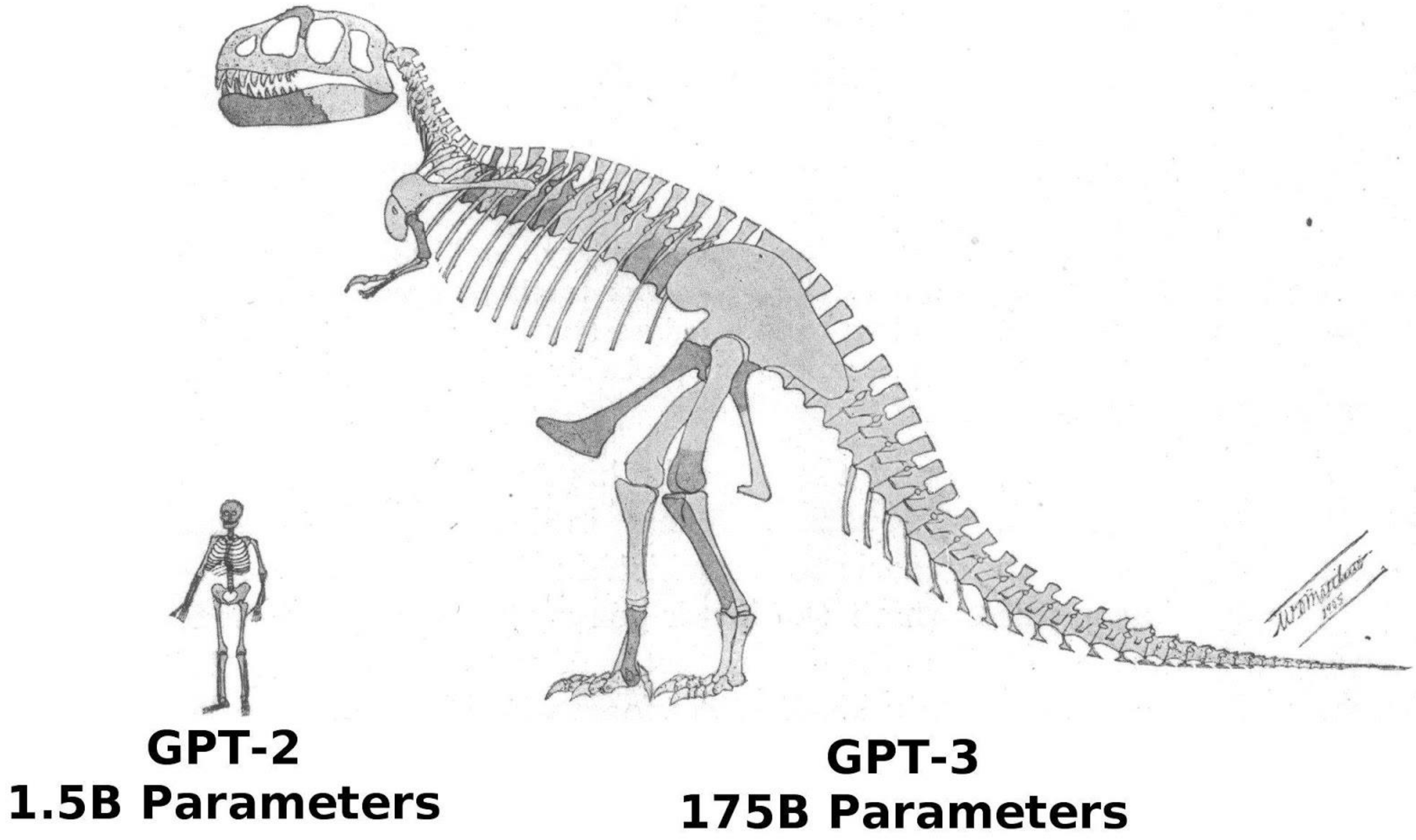
Model Size and Computation are Increasing





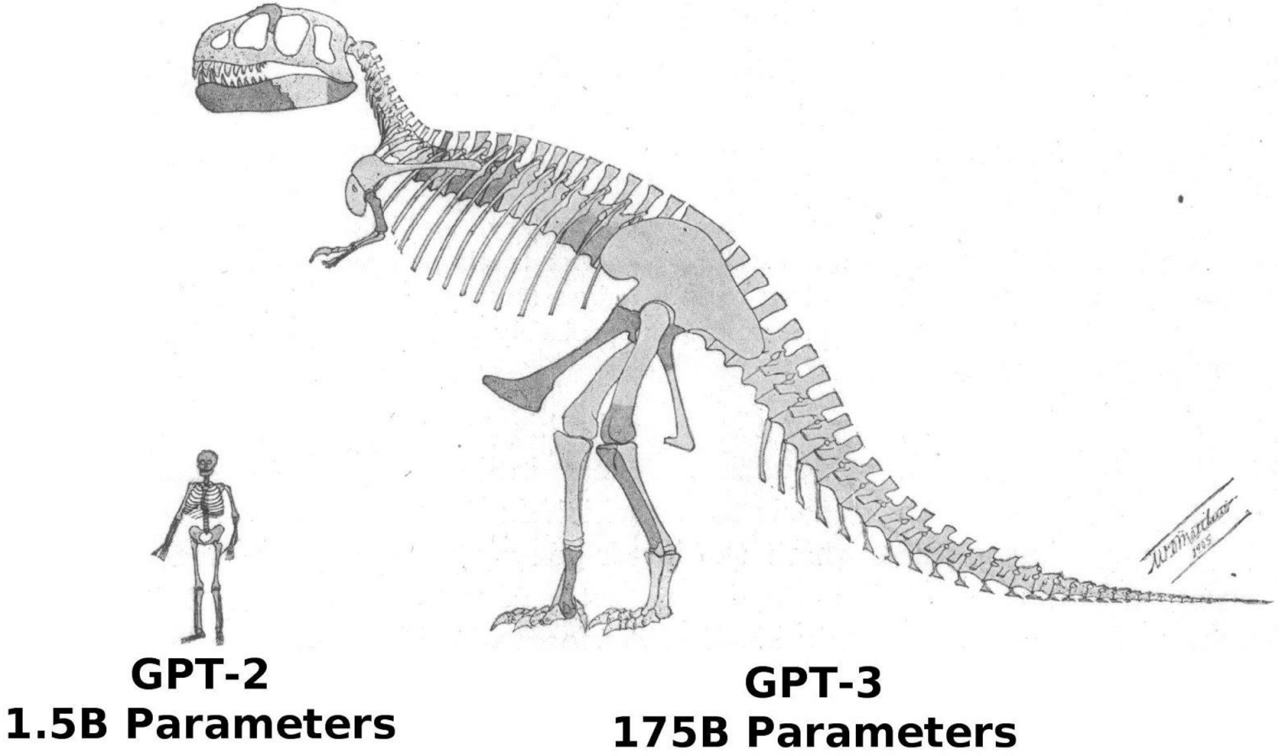
[1] Brainwave, Berkeley EE290 Hardware in Machine Learning, 2020.

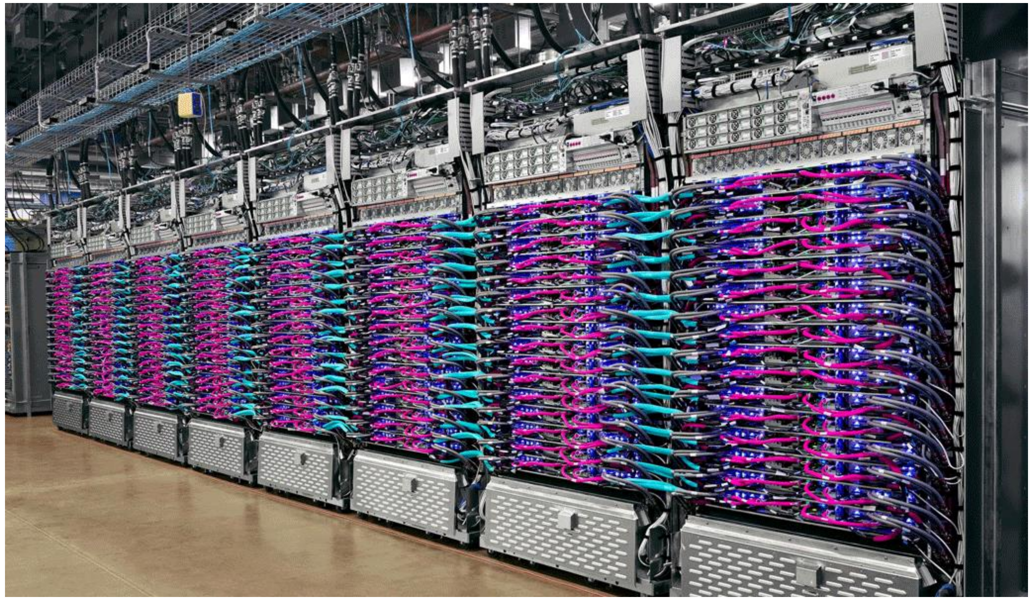
Model Size and Computation are Increasing



[1] Image from https://blog.exxactcorp.com/what-can-you-do-with-the-openai-gpt-3-language-model/.

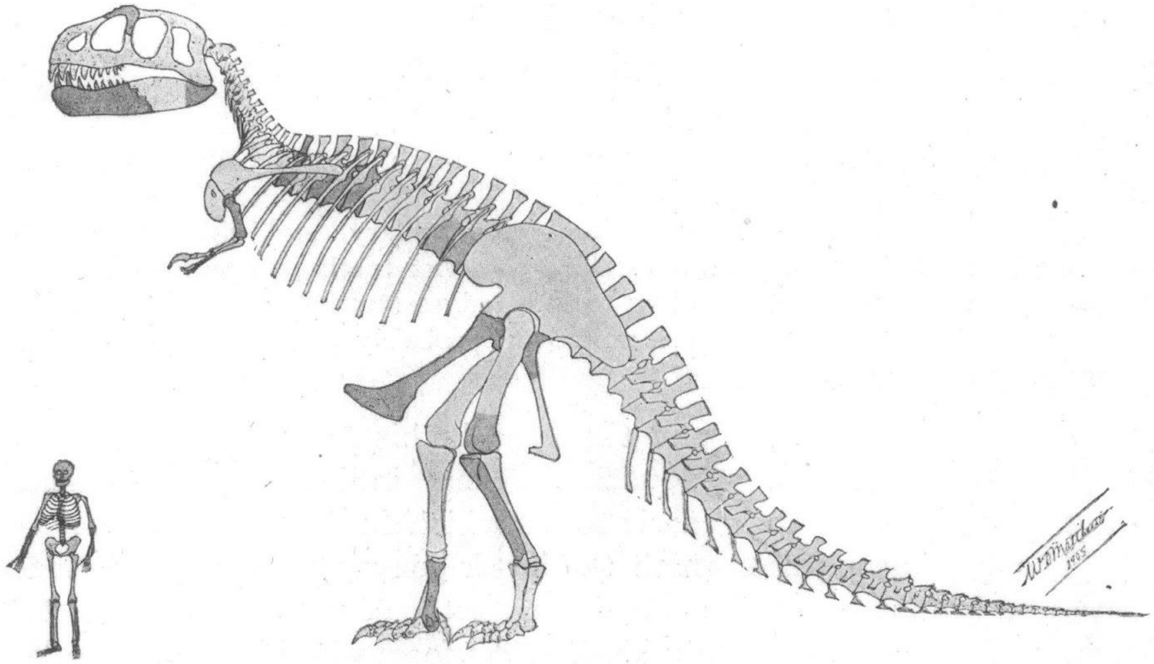
Model Size and Computation are Increasing





[1] Image from https://cloud.google.com/tpu.

Why Inference at the Edge? Privacy

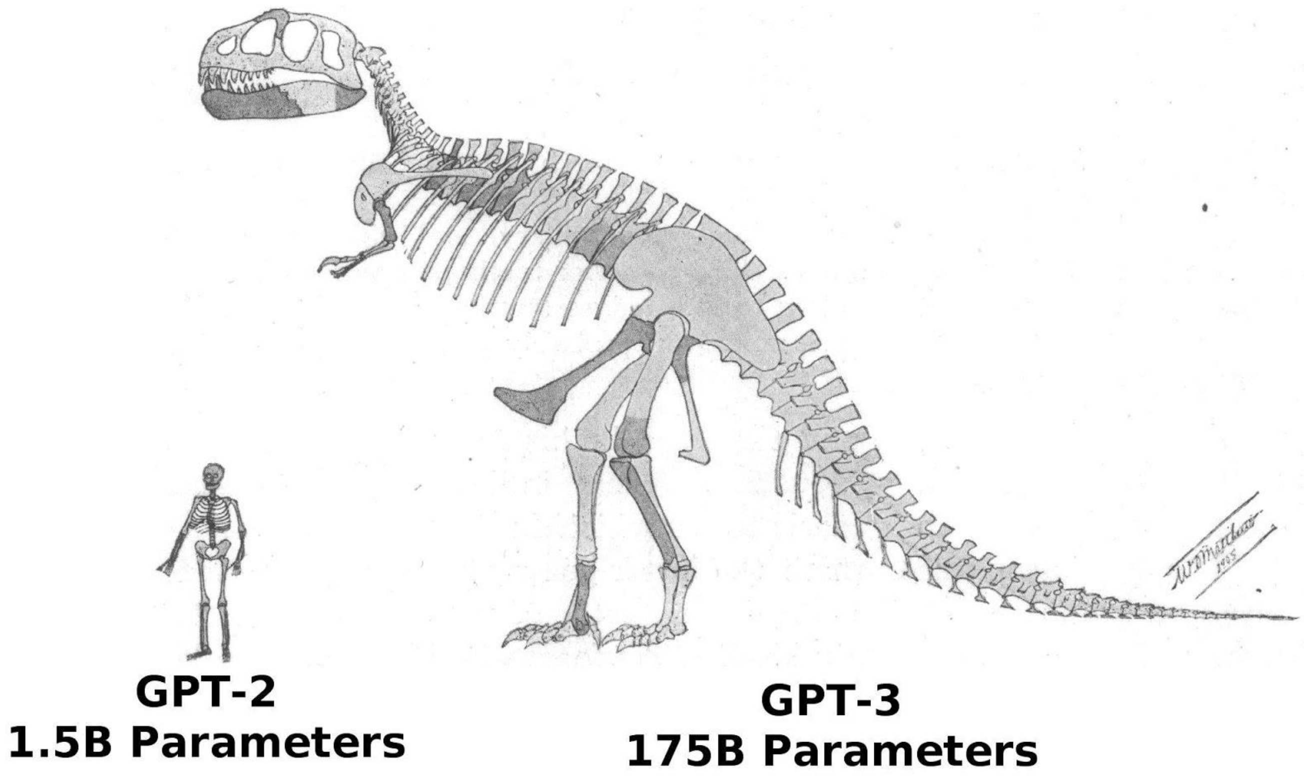


GPT-2
1.5B Parameters

GPT-3
175B Parameters

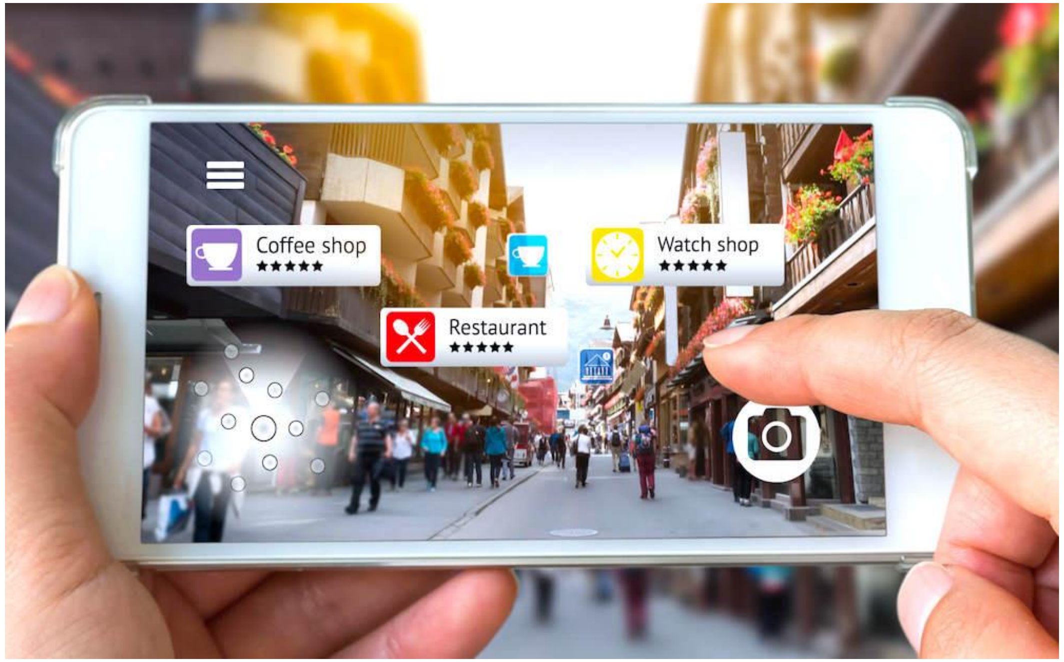
[1] Image from the Internet.

Why Inference at the Edge? Power



[1] Image from the Internet.

Why Inference at the Edge? Latency



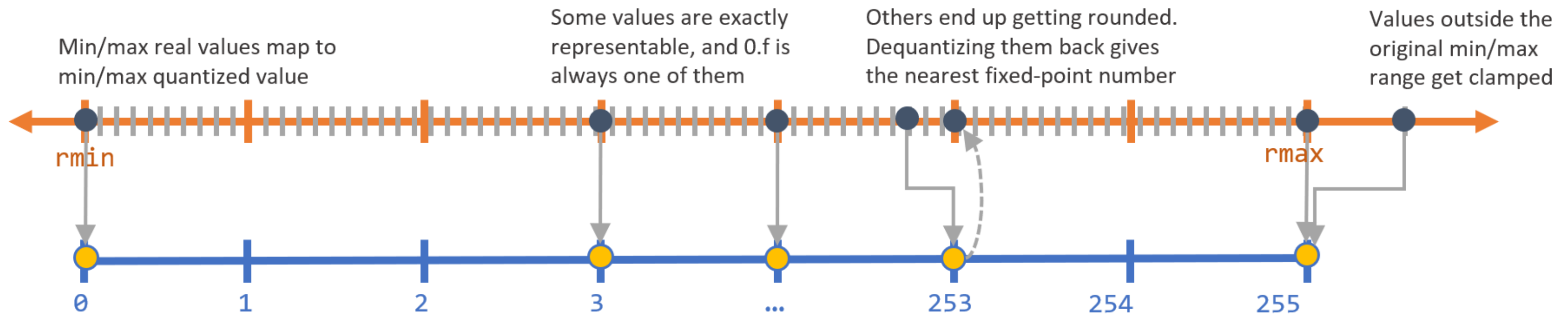
[1] Image from the Internet.

Quantization: Small Size and Fast Computation

- r: real value
- r_{max}, r_{min} : max/min of values
- B: Quantization Bit-width
- S (FP32): Scaling Factor
- z (int): Zero Point
- q: Fixed point quantized values

$$S = \frac{r_{max} - r_{min}}{2^\wedge B - 1}$$

$$r = S(q - z)$$

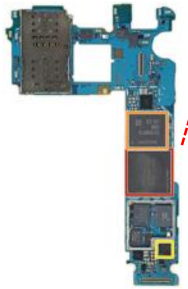


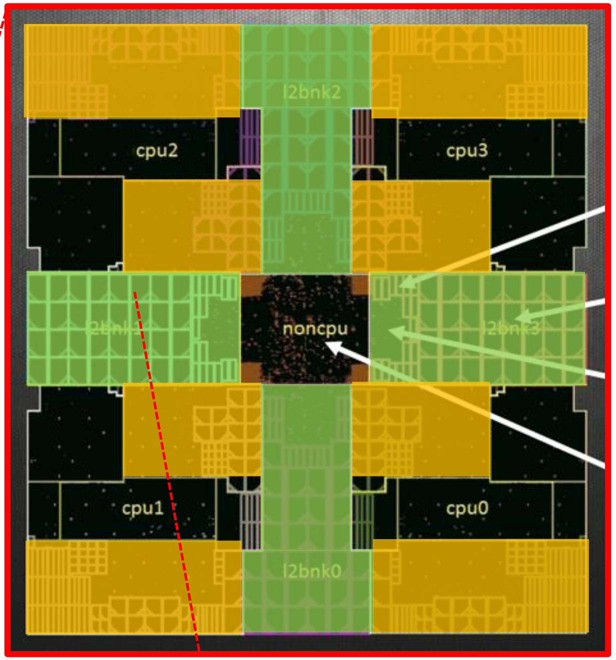
Uniform 8-bit Quantization

[1] Illustration from Sahni Manas.

Quantization: Low Power Consumption

Galaxy S7



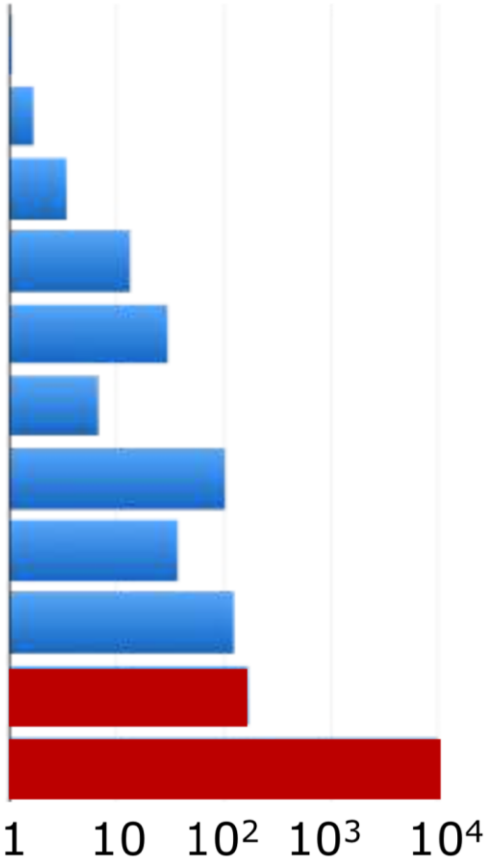


L1 Cache/TLB

L2 Cache

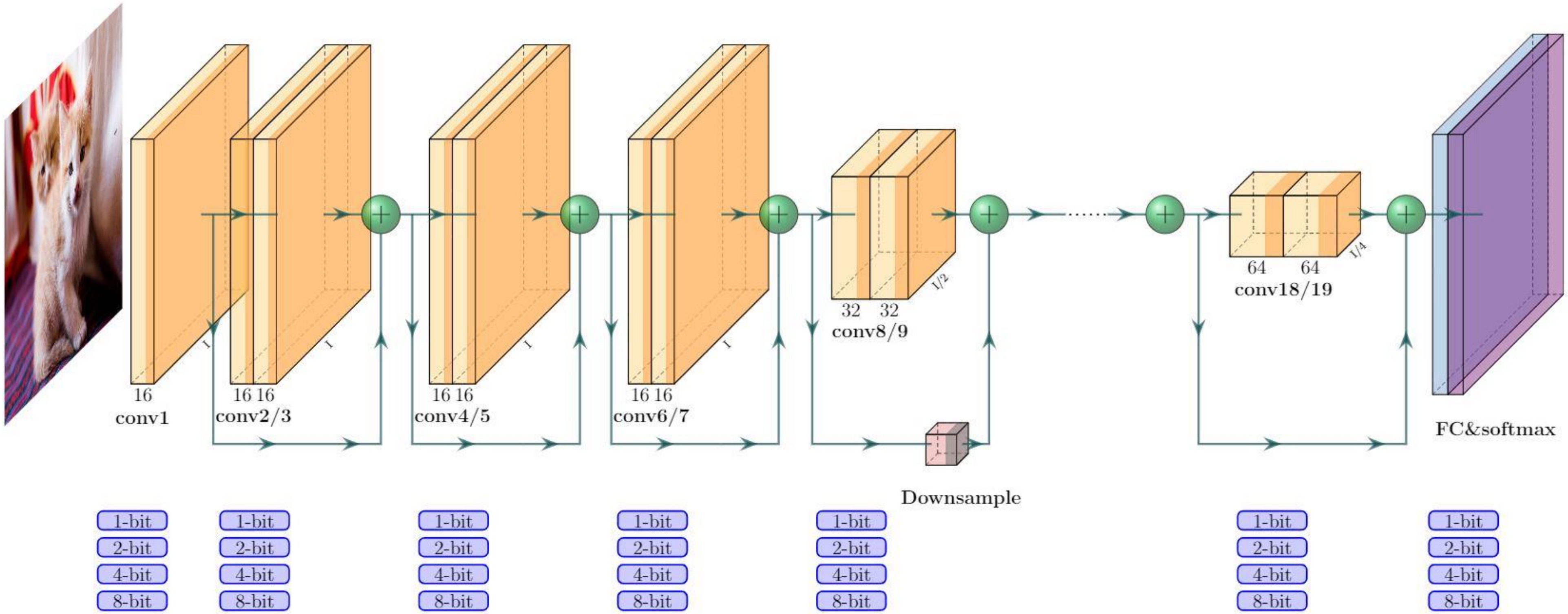
Operation:	Energy (pJ)
8b Add	0.03
16b Add	0.05
32b Add	0.1
16b FP Add	0.4
32b FP Add	0.9
8b Multiply	0.2
32b Multiply	3.1
16b FP Multiply	1.1
32b FP Multiply	3.7
32b SRAM Read (8KB)	5
32b DRAM Read	640

Relative Energy Cost



[**Horowitz**, *ISSCC* 2014]

Mixed-Precision: Exponential Search Space

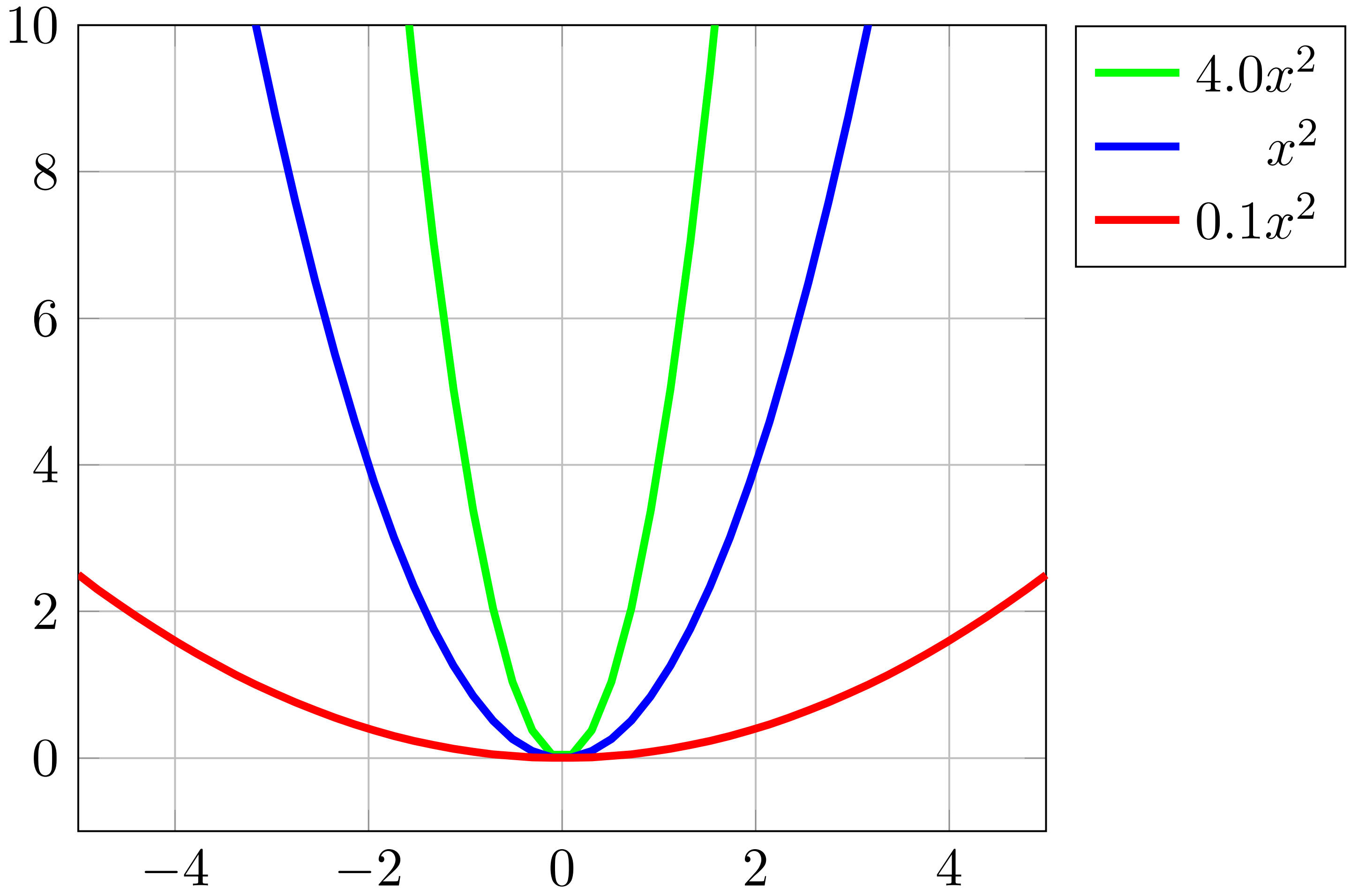


Which mixed-precision setting works better?

Outline

- Introduction
- Hessian-AWare Quantization
- Automated Mixed-Precision
- Hardware-Aware Deployment
- Conclusion

Second Order Sensitivity Analysis

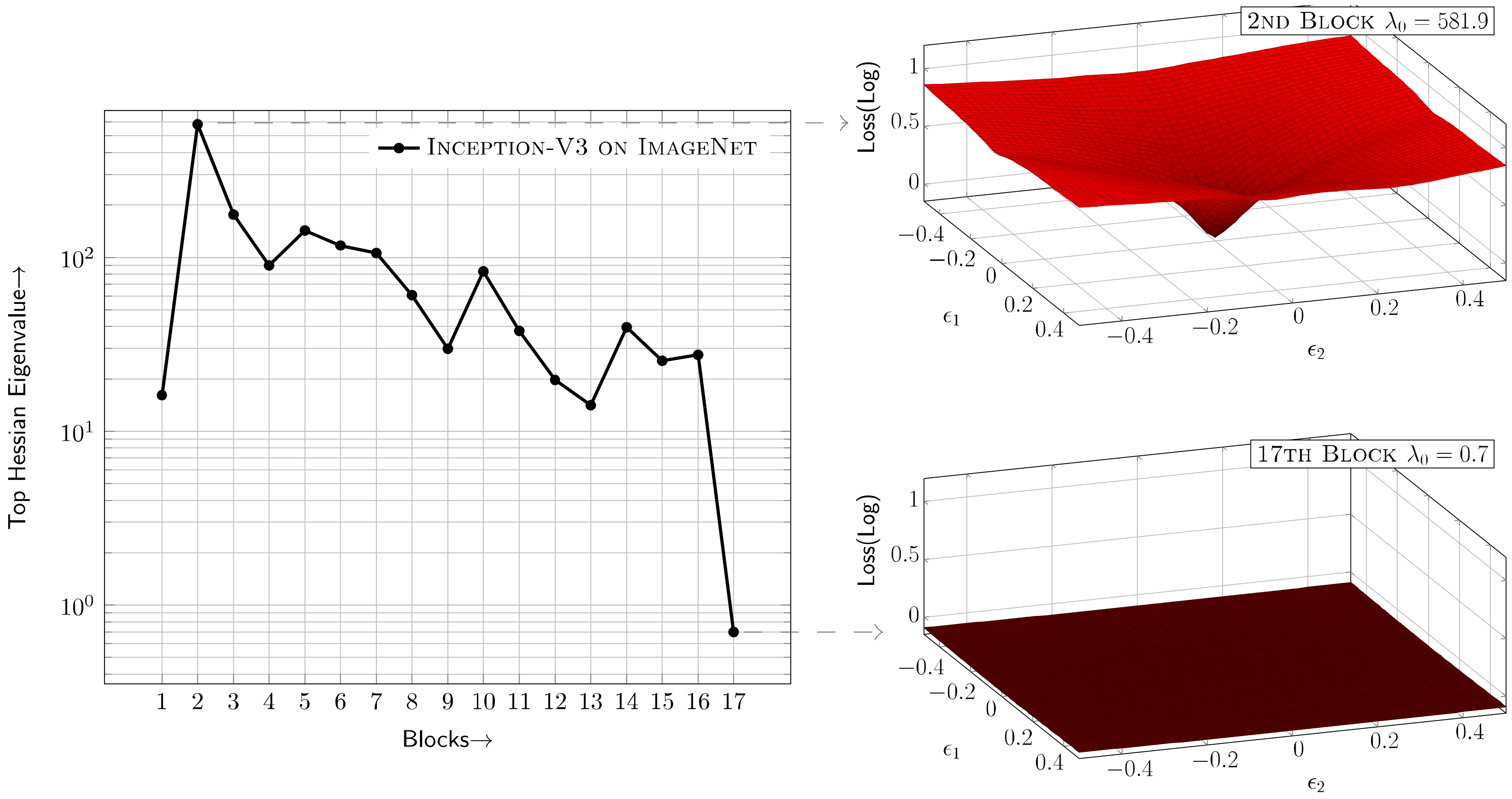


At the origin, the first derivative of $y = 4x^2$, $y = x^2$, $y = 0.1\,x^2$ is all the same: 0

The **second derivative** give more information: 8 , 2, and 0.2 respectively

HAWQ: Hessian-AWare Quantization

Only quantize layers that have **small top eigenvalue** to **ultra-low precision**



[1] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV'19

HAWQ: ResNet50 on ImageNet

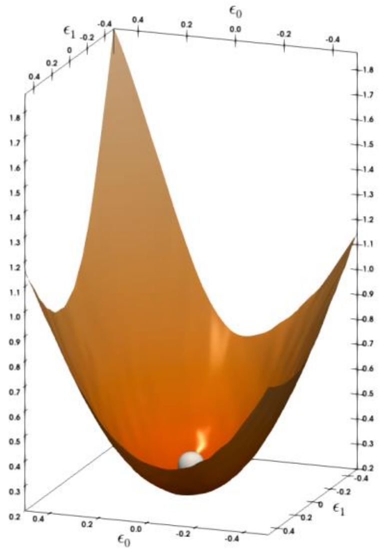
Method	w-bits	a-bits	Top-1	W-Comp	Size(MB)
Baseline	32	32	77.39	1.00×	97.8
Dorefa [43]	2	2	67.10	16.00×	6.11
Dorefa [43]	3	3	69.90	10.67×	9.17
PACT [2]	2	2	72.20	16.00×	6.11
PACT [2]	3	3	75.30	10.67×	9.17
LQ-Nets [40]	3	3	74.20	10.67×	9.17
Deep Comp. [8]	3	MP	75.10	10.41×	9.36
HAQ [35]	MP	MP	75.30	10.57×	9.22
HAWQ	2 MP	4 MP	**75.48**	12.28×	**7.96**

Go to page 8

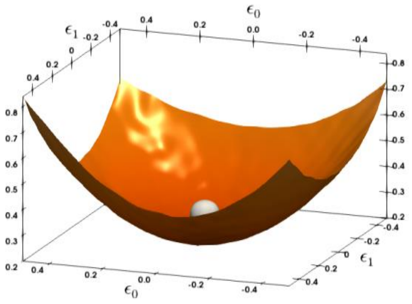
[1] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV'19

Hessian Aware Quantization for BERT-Base on MNLI

Method	w-bits	e-bits	Acc m	Acc mm	Size	Size w/o-e
Baseline	32	32	84.00	84.40	415.4	324.5
Q-BERT	8	8	83.91	83.83	103.9	81.2
DirectQ	4	8	76.69	77.00	63.4	40.6
Q-BERT	4	8	**83.89**	**84.17**	63.4	40.6
DirectQ	3	8	70.27	70.89	53.2	30.5
Q-BERT	3	8	**83.41**	**83.83**	53.2	30.5
Q-BERT$_{MP}$	2/4 $_{MP}$	8	**83.51**	**83.55**	53.2	30.5
DirectQ	2	8	53.29	53.32	43.1	20.4
Q-BERT	2	8	**76.56**	**77.02**	43.1	20.4
Q-BERT$_{MP}$	2/3 $_{MP}$	8	**81.75**	**82.29**	**46.1**	**23.4**



4th Layer



10th Layer

[1] Shen Sheng, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael Mahoney, Kurt Keutzer, Q-BERT: Hessian-based Quantization for BERT, AAAI 2020.

Outline

- Introduction
- Hessian-AWare Quantization
- Automated Mixed-Precision
- Hardware-Aware Deployment
- Conclusion

Automated Mixed-Precision

We prove Hessian Trace is a better sensitivity metric than the Top-1 Eigenvalue.

Hessian Trace can be used to quantify second-order perturbation Ω.

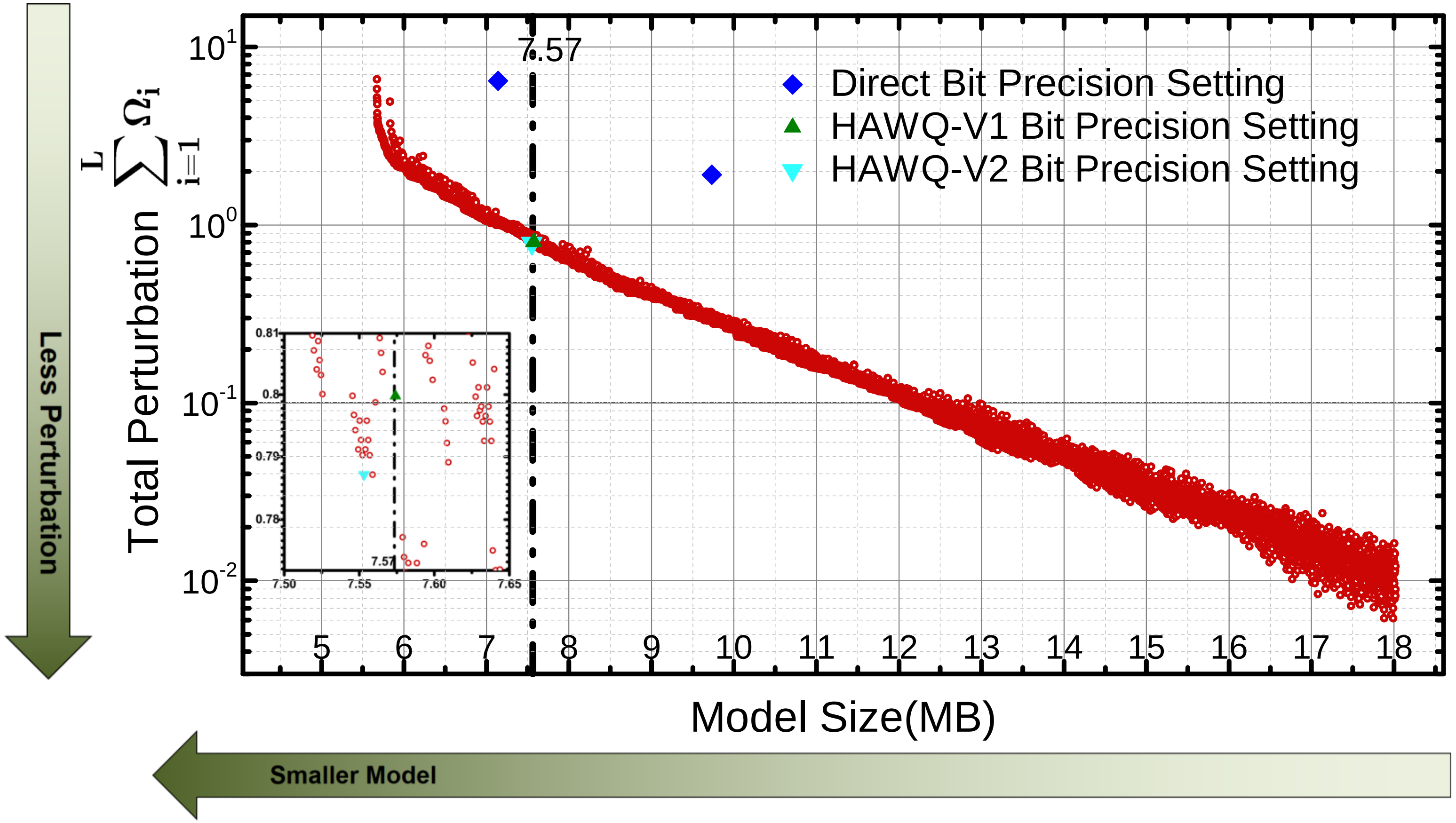
Mixed-precision quantization becomes an Integer Linear Programming (ILP) problem:

$$\Omega = \sum_{i=1}^{L} \Omega_i = \sum_{i=1}^{L} \overline{Tr}(H_i) \cdot \|Q(W_i) - W_i\|_2^2,$$

$$\text{Objective:} \quad \min_{\{b_i\}_{i=1}^{L}} \sum_{i=1}^{L} \Omega_i^{(b_i)},$$

$$\text{Subject to:} \quad \sum_{i=1}^{L} M_i^{(b_i)} \leq \text{ Model Size Limit},$$

Automated Mixed-Precision



HAWQ-V2: ResNet50 on ImageNet

Precisions for all layers are 100% automatically selected.

Method	w-bits	a-bits	Top-1	W-Comp	Size(MB)
Baseline	32	32	77.39	1.00×	97.8
Dorefa [28]	2	2	67.10	16.00×	6.11
Dorefa [28]	3	3	69.90	10.67×	9.17
PACT [6]	2	2	72.20	16.00×	6.11
PACT [6]	3	3	75.30	10.67×	9.17
LQ-Nets [26]	3	3	74.20	10.67×	9.17
Deep Comp. [10]	3	MP	75.10	10.41×	9.36
HAQ [23]	MP	MP	75.30	10.57×	9.22
HAWQ [7]	2 MP	4 MP	75.48	12.28×	7.96
HAWQ-V2	2 MP	4 MP	**75.92**	12.24×	7.99

[1] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV 2019.
[2] Z. Dong, Z. Yao, D. Arfeen, A. Gholami, M. Mahoney, K. Keutzer, HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks, NeurIPS 2020.

HAWQ-V2: SqueezeNext on ImageNet

Precisions for all layers are 100% automatically selected.

Method	w-bits	a-bits	Top-1	W-Comp	Size(MB)
Baseline	32	32	69.38	1.00×	10.1
Direct [7]	3 MP	8	65.39	9.04×	1.12
HAWQ [7]	3 MP	8	68.02	9.26×	1.09
HAWQ-V2	3 MP	8	**68.68**	9.40×	1.07

[1] Z. Dong, Z. Yao, A. Gholami, M. Mahoney, K. Keutzer, HAWQ: Hessian Aware Quantization of Neural Networks With Mixed Precision, ICCV 2019.
[2] Z. Dong, Z. Yao, D. Arfeen, A. Gholami, M. Mahoney, K. Keutzer, HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks, NeurIPS 2020.

Outline

- Introduction
- Hessian-AWare Quantization
- Automated Mixed-Precision
- Hardware-Aware Deployment
- Conclusion

TVM

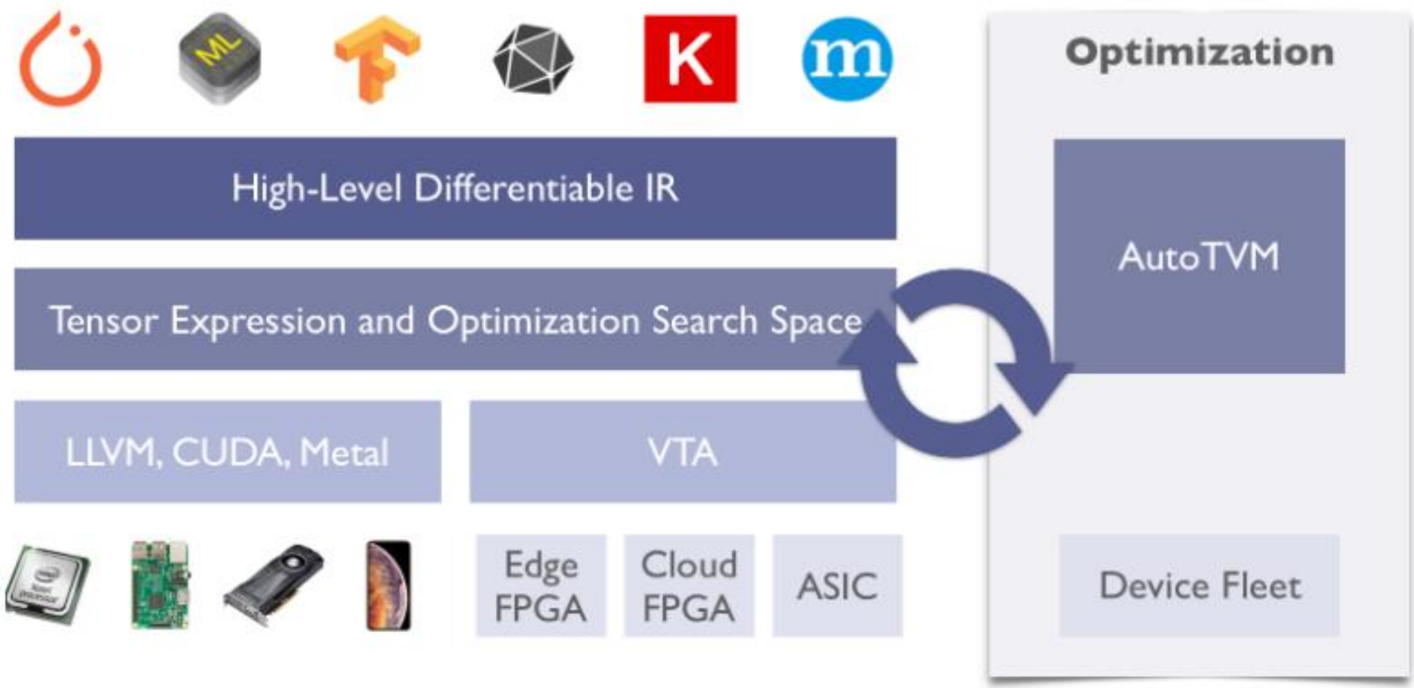
- A compiler stack for CPU, GPU and accelerators

- Autotuning framework

Need to add:

1. Mixed-precision support

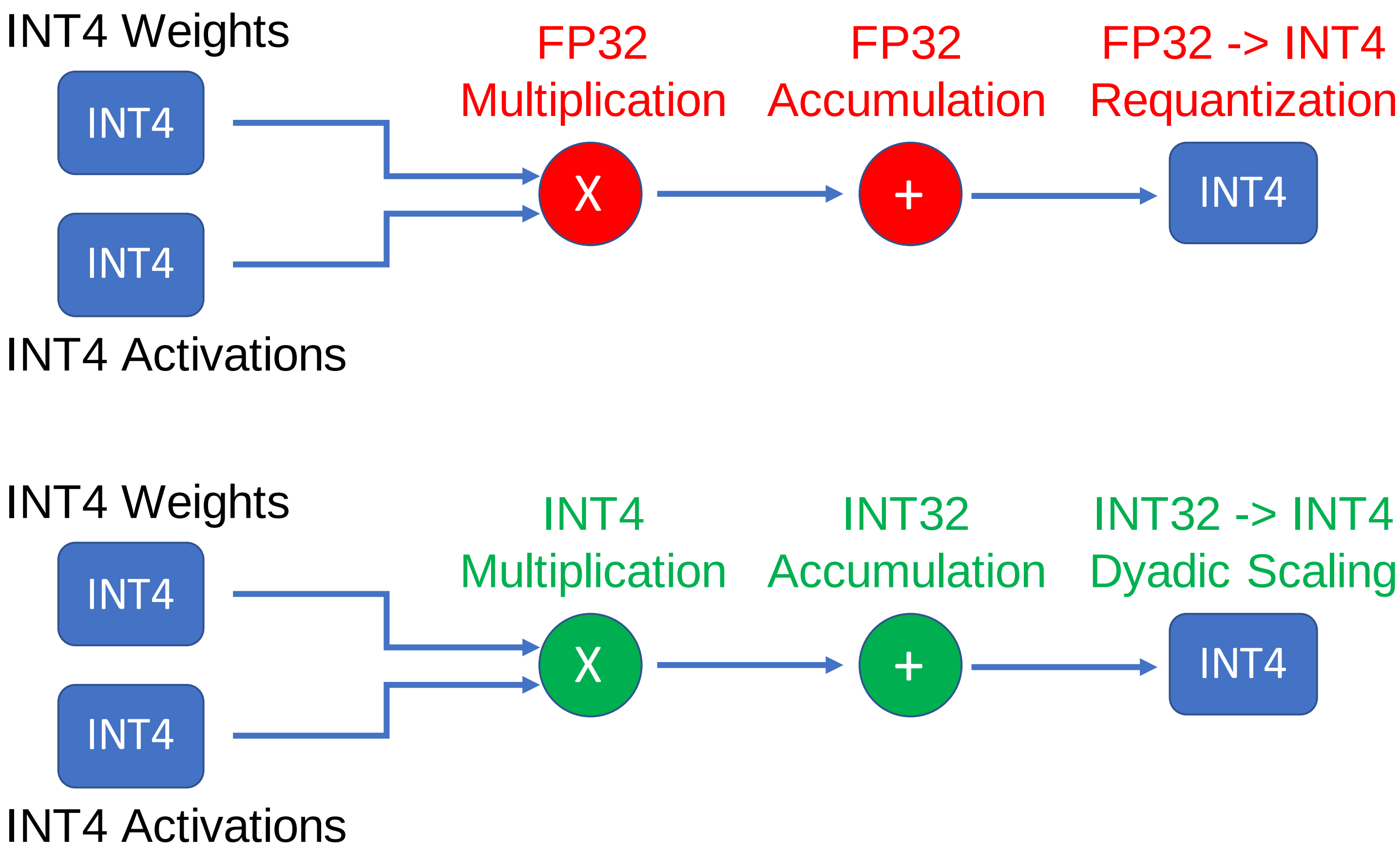
2. Low bit operations support



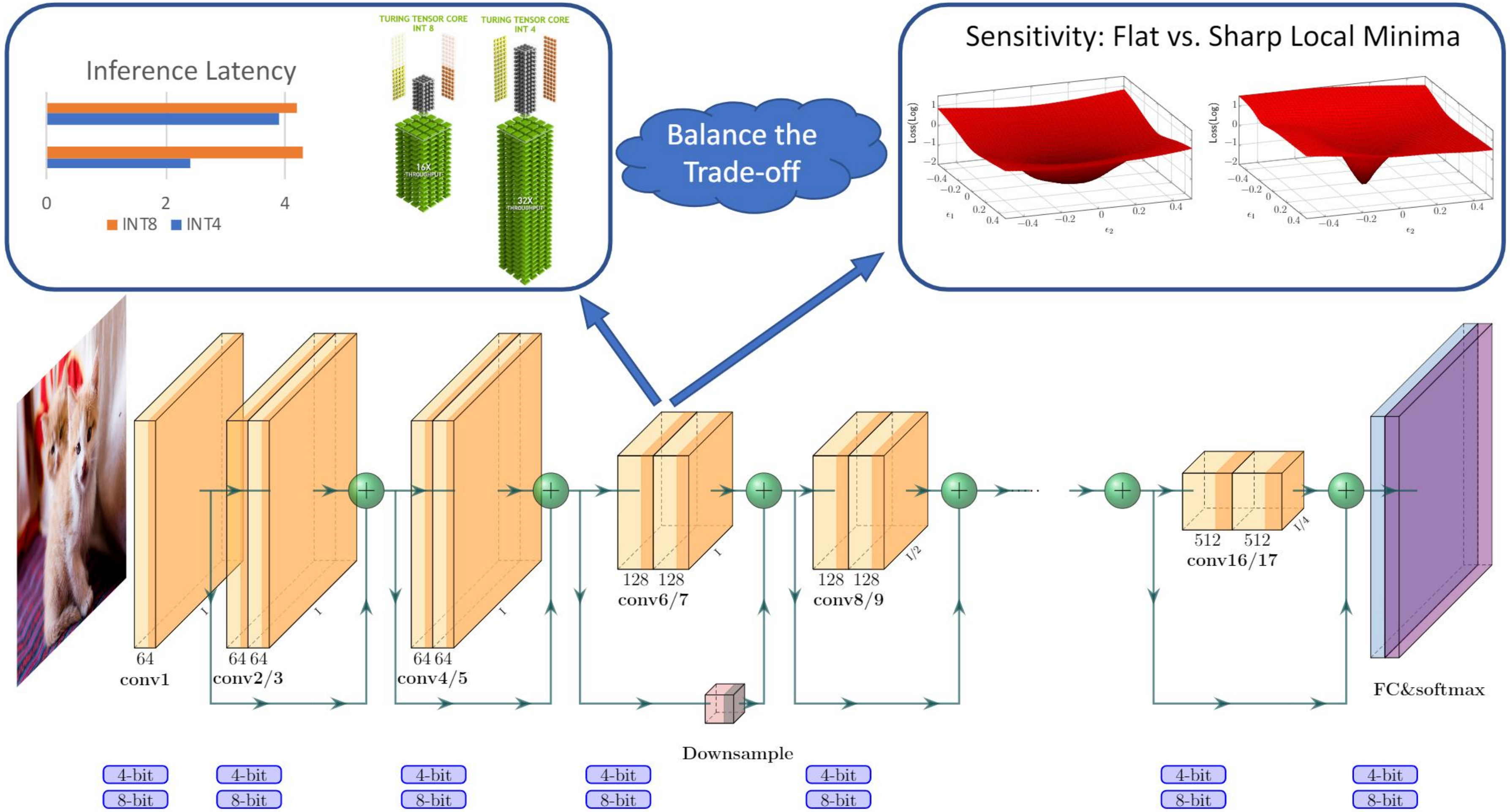
About Apache (incubating) TVM. (n.d.). Retrieved from https://tvm.apache.org/about

[1] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, Carlos Guestrin, and Arvind Krishnamurthy. TVM: An Automated End-to-End Optimizing Compiler for Deep Learning. In 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), pages 578–594, 2018.

Dyadic Quantization with Integer Arithmetic



HAWQ-V3: Hardware-Aware Deployment



HAWQ-V3: Hardware-Aware Deployment

We find the best bit precision configuration such that:

- Minimally perturbs the model
- Meets application specific requirements:
  - Model size constraint
  - Total bit operations for inference
  - Inference Latency

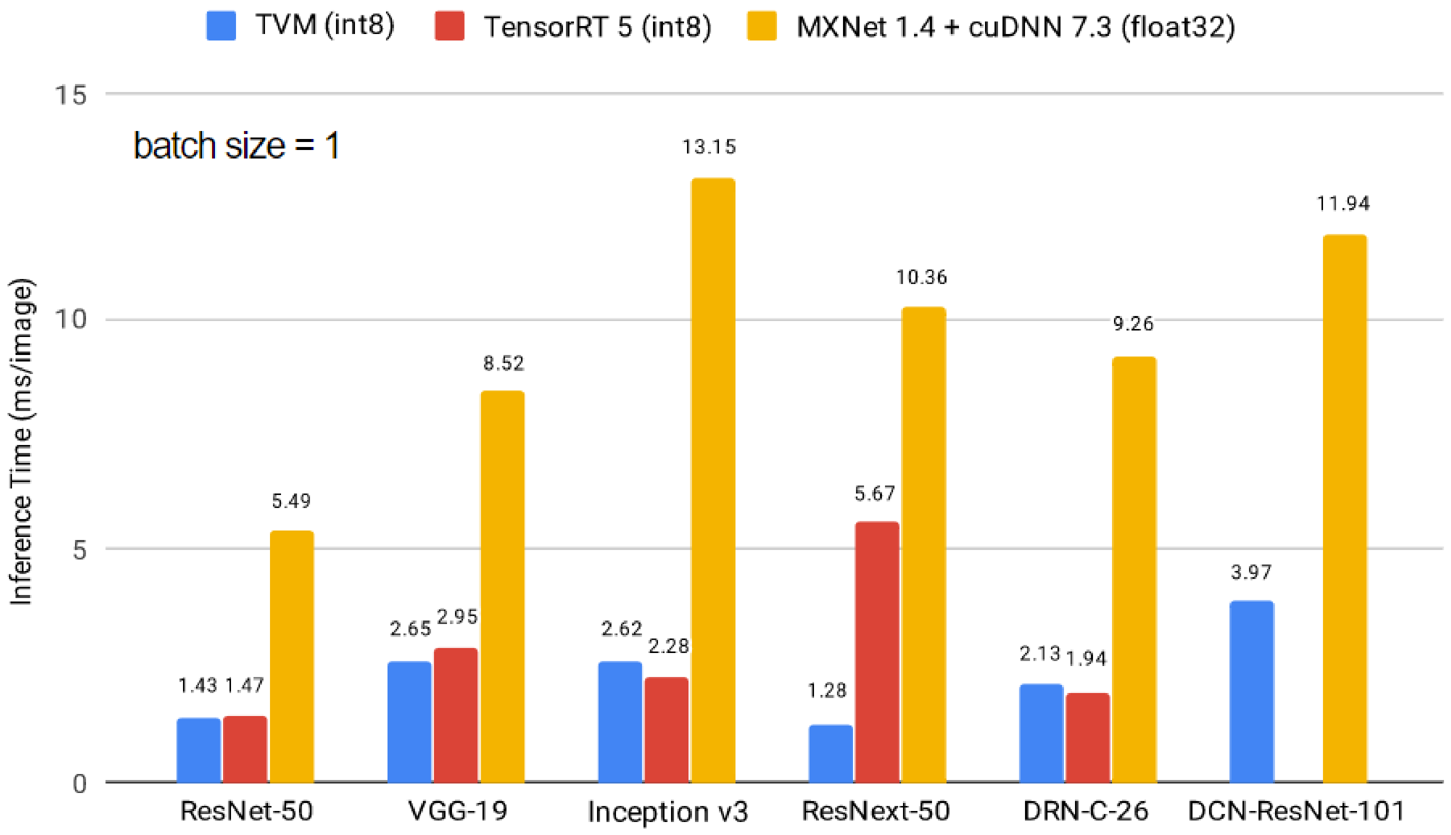
$$\text{Objective: } \min_{\{b_i\}_{i=1}^{L}} \sum_{i=1}^{L} \Omega_i^{(b_i)},$$

$$\text{Subject to: } \sum_{i=1}^{L} M_i^{(b_i)} \leq \text{ Model Size Limit},$$

$$\sum_{i=1}^{L} G_i^{(b_i)} \leq \text{ Bops Limit},$$

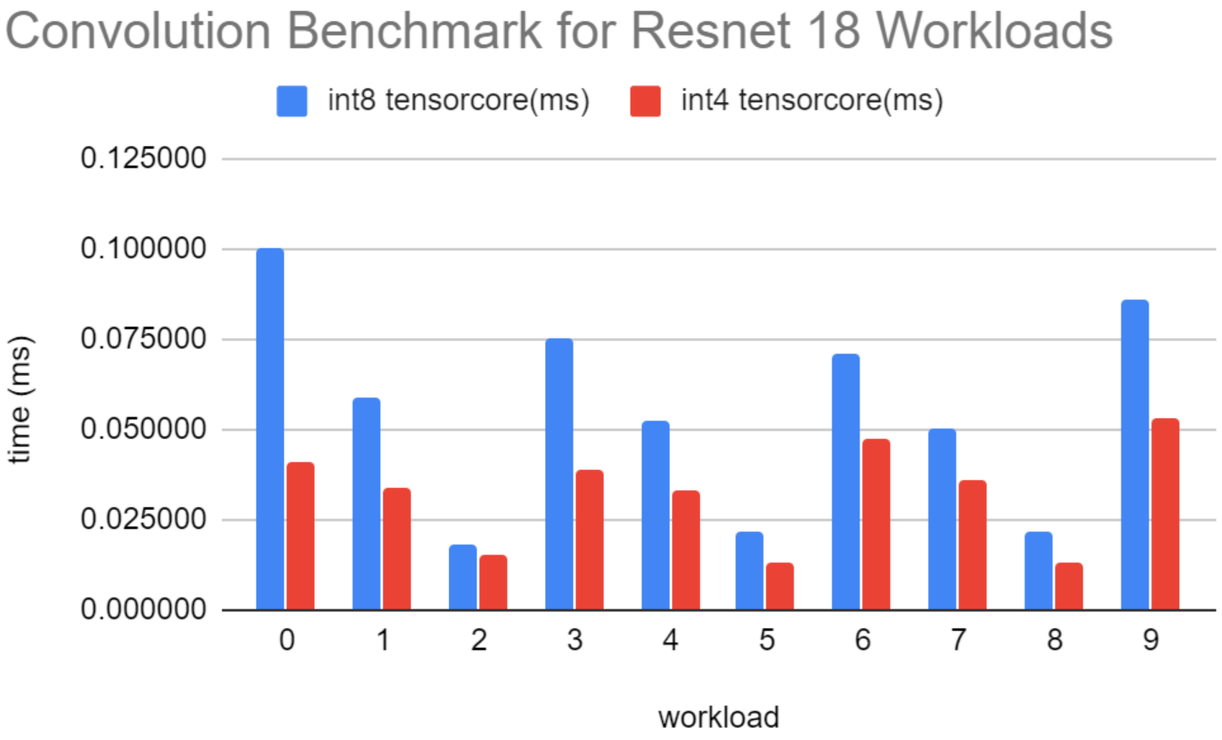
$$\sum_{i=1}^{L} Q_i^{(b_i)} \leq \text{ Latency Limit}.$$

Uniform 8-bit Performance



[1] Image from https://tvm.apache.org/2019/04/29/opt-cuda-quantized

Int4 Performance of ResNet18 on ImageNet



Resnet 18	Int8 (ms)	Int4 (ms)	Speed-up
Batch=1	0.85	0.62	1.37x
Batch=8	4.55	3.02	1.51x
Batch=16	8.84	5.91	1.50x

***A workload is a convolutional function with certain shape*

Open-Source Projects

- QTorch: coming soon (github keyword: HAWQ)
  HAWQ-V3 + TVM,
  Easy to use, such as torchvision,
  Support Various Networks: ResNets, Inceptions, MobileNets, EfficientNets, and so on,
  Easy deployment and Fast inference,
  High accuracy mixed-precision models (19MB ResNet50, 77% Acc on ImageNet).

- PyHessian: https://github.com/amirgholami/PyHessian

- ZeroQ: https://github.com/amirgholami/ZeroQ

Outline

- Introduction
- Hessian-AWare Quantization
- Automated Mixed-Precision
- Hardware-Aware Deployment
- Conclusion

Conclusion

- We use Hessian information to help conduct mixed-precision quantization.
- We develop an automated method to generate good mixed-precision settings.
- Our methods generalize well for different models on classification, object detection and NLP tasks.
- We develop TVM implementation for our low bit mixed-precision models.
- We show hardware-aware deployment where we jointly consider model size and latency.

Thank you for listening!