Introduction and Execution of Quantization based on PyTorch

2^{nd} Paper Study | 2022.07.20 Wed.

홍 세 현 Hong Sehyun

STRADVISION

Outline

1. Introduction
2. Dynamic Quantization
3. Static Quantization
4. Quantization Aware Training
5. Experiments
6. Reference and Q & A

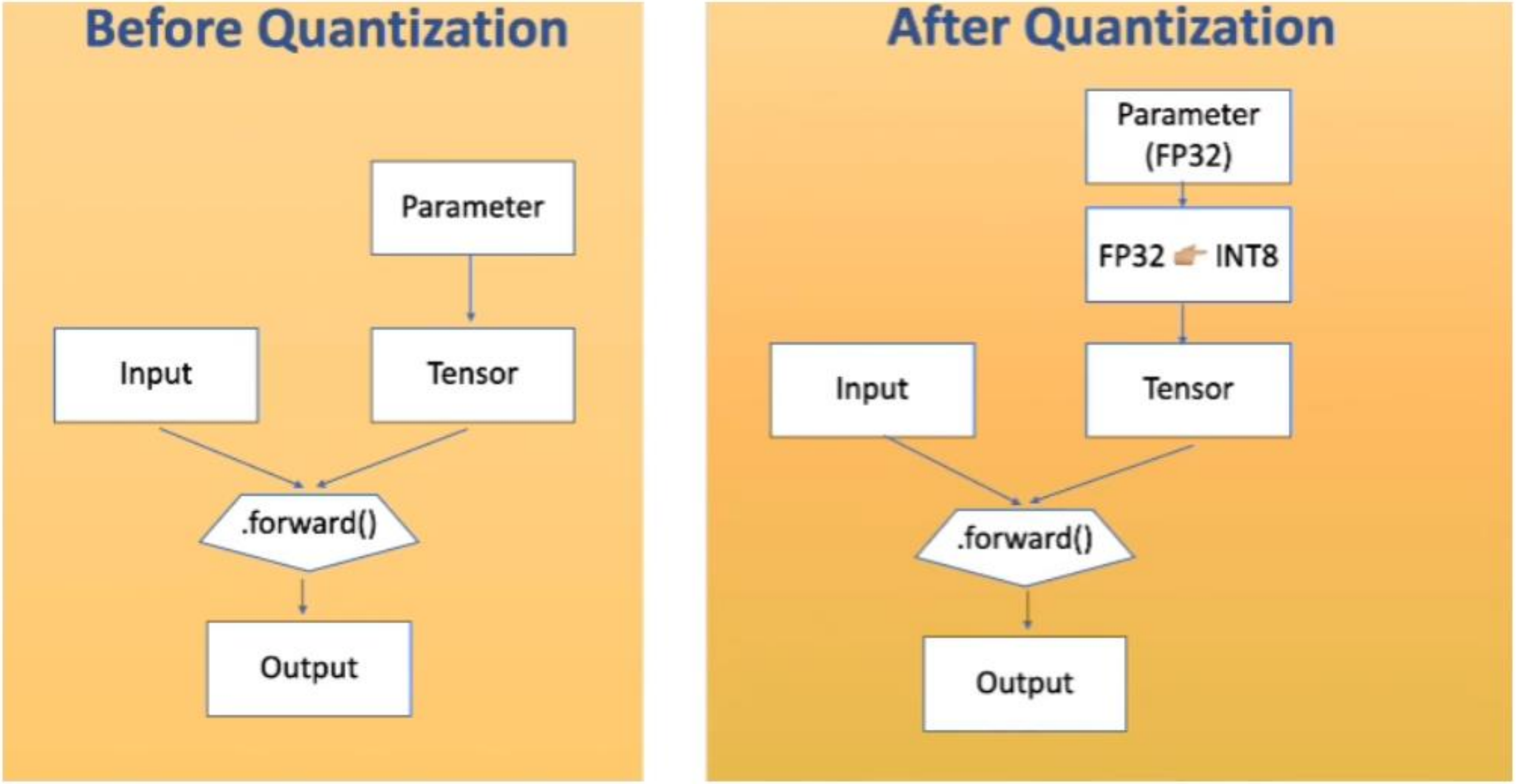
Introduction

Embedded Board Sample Image

Goal of Quantization

1. Reduce Model Size
2. Reduce Computation
3. Using Hardware more Efficiently and Economically (benefits for deployment)

Introduction



Result of Quantization

- Model Size Drop : ¼
- Inference Speed Up : 2~4 Times
- Almost no Performance Drop

Introduction

Types of Quantification

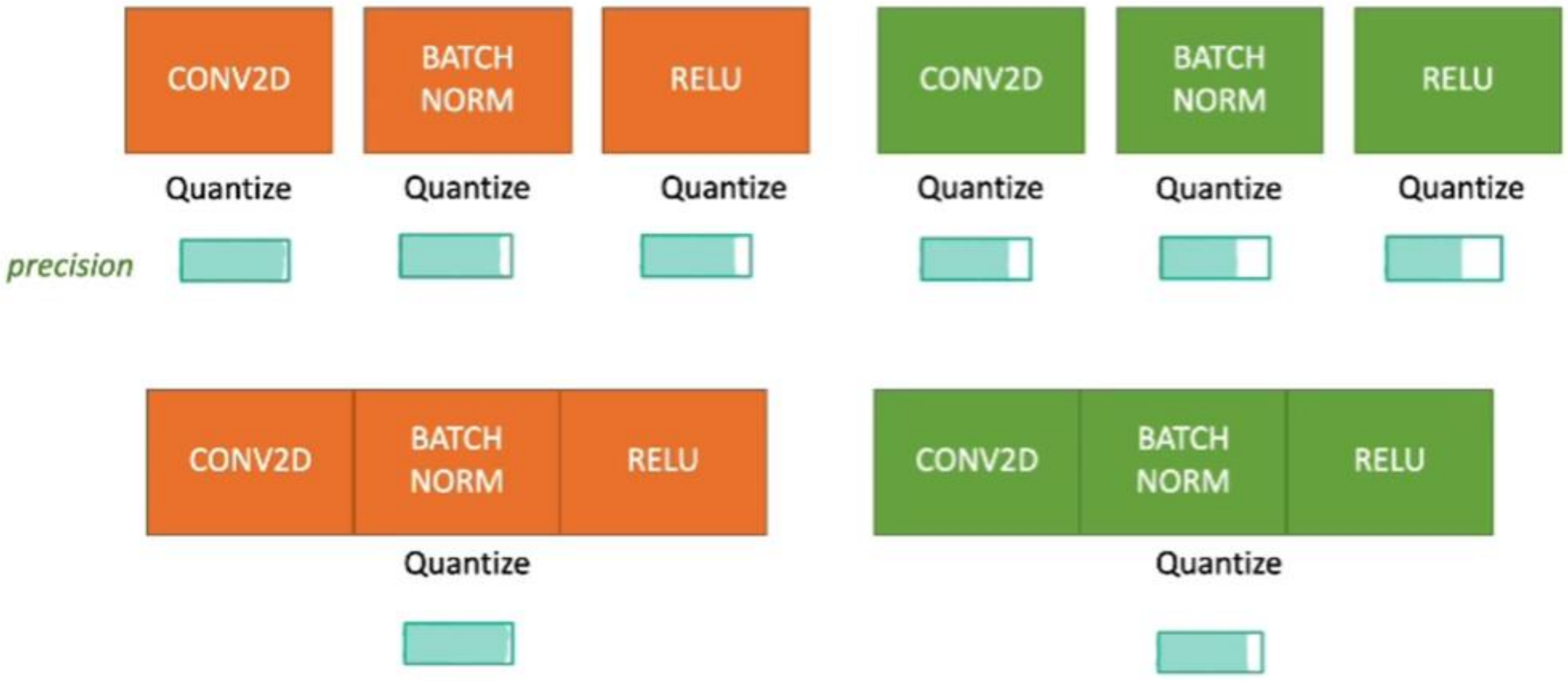
	Quantization	Dataset Requirements	Works Best For	Accuracy
Dynamic Quantization	weights only (both fp16 and int8)	None	LSTMs, MLPs, Transformers	good
Static Post Training Quantization	weights and activations (8 bit)	calibration	CNNs	good
Static Quantization-Aware Training	weights and activations (8 bit)	fine-tuning	CNNs	best

Introduction

Table of Quantization Type Selection Guide

Model Type	Preferred scheme	Why
LSTM/RNN	Dynamic Quantization	Throughput dominated by compute/memory bandwidth for weights
BERT/Transformer	Dynamic Quantization	Throughput dominated by compute/memory bandwidth for weights
CNN	Static Quantization	Throughput limited by memory bandwidth for activations
CNN	Quantization Aware Training	In the case where accuracy can't be achieved with static quantization

Quantization Techniques[1] – Model Fusion



Introduction

Quantization Techniques[2] – Formula Definition

FP32

-3.57	4.67	-3.97
-1.74	2.34	-1.76
-4.75	-0.06	3.07

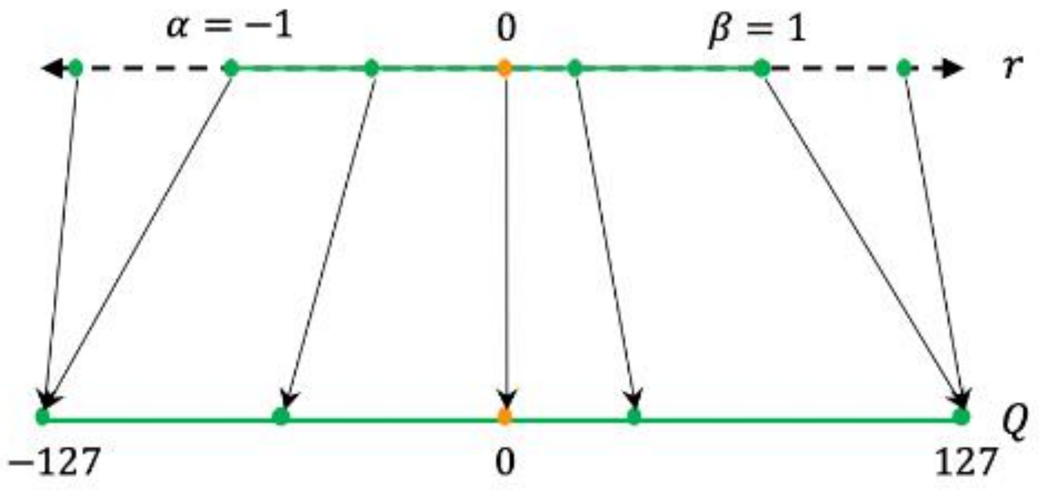
Quantization →

INT8

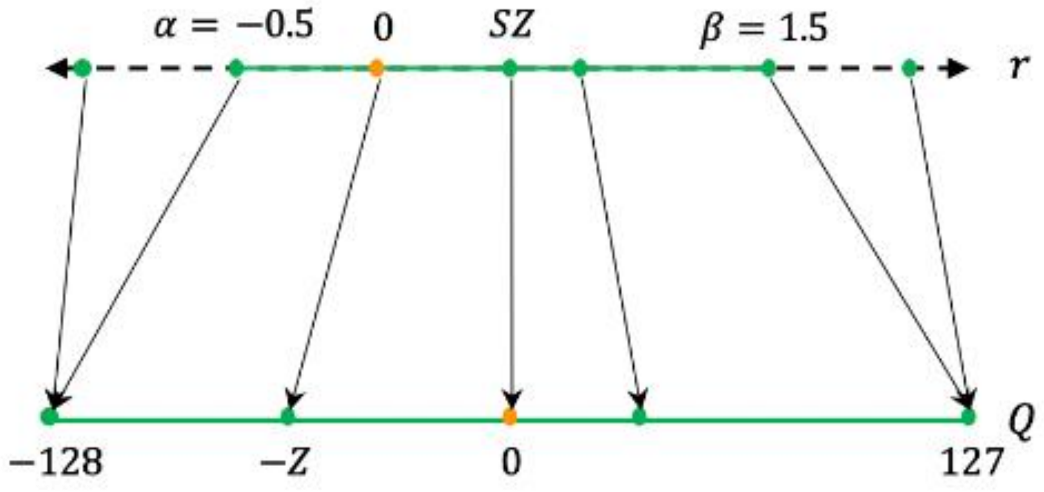
33	255	22
82	192	81
1	127	212

Introduction

Quantization Techniques[2] – Formula Definition



Symmetric Quantization / Affine Quantization Mapping



Asymmetric Quantization / Scale Quantization Mapping

0.1 (Float Type) $\rightarrow$ $round\left(127 \times \frac{1}{10}\right) = \mathbf{13}$

Quantization Techniques[3] – DeQuantization

FP32

-3.57	4.67	-3.97
-1.74	2.34	-1.76
-4.75	-0.06	3.07

← De-quantization

INT8

33	255	22
82	192	81
1	127	212

Dynamic and Static Quantization

Dynamic Quantization

Dynamic Clipping Range

Static Quantization
(Post Training Quantization, PTQ)

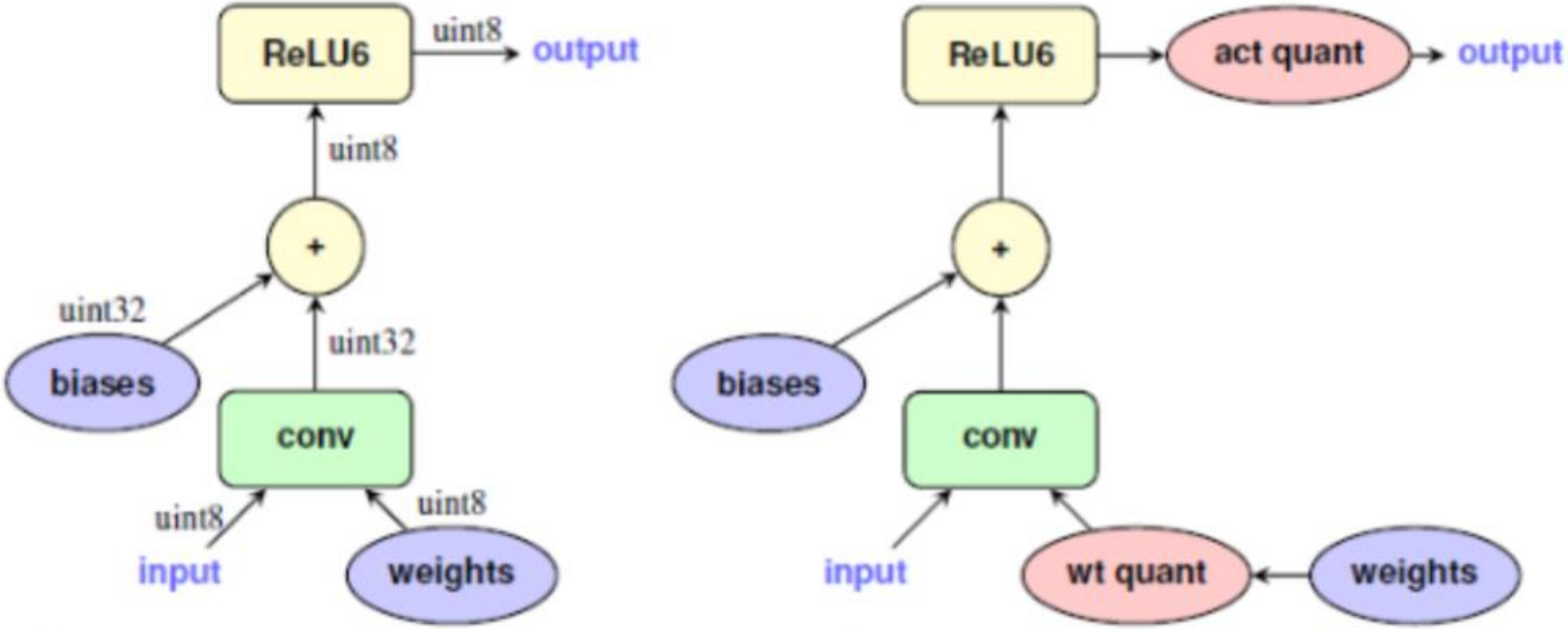
Static Clipping Range

Quantization Modes	Features	Inference Latency	Inference Accuracy Loss
Dynamic Quantization	Dynamic, Real Time Calculating Data Range	Faster	**Smallest**
Static Quantization(=PTQ)	Calibration & Limit Data Range	**Fastest**	Smaller
Quantization Aware Training	-	**Fastest**	**Smallest**

Quantization Aware Training

	PTQ(Post Training Quantization)	QAT(Quantization Aware Training)
Definition	Floating Point 모델로 학습을 한 뒤 결과 Weight값들에 대하여 Quantization하는 방식으로 학습을 완전히 끝내 놓고 Quantization error를 최소화하는 방식	학습 진행 시점에 Inference 시 Quantization 적용에 의한 영향을 미리 시뮬레이션을 하는 방식으로 최적의 Weight를 구하는 것과 동시에 Quantization을 하는 방식
Advantage	파라미터 Size 큰 대형 모델에 대해서는 정확도 하락의 폭이 작음	Quantization 이후 모델의 **정확도 감소 폭을 최소화**할 수 있음
Disadvantage	파라미터 Size가 작은 소형 모델에 대해서는 정확도 하락의 폭이 큼	모델 학습 이후 추가 학습이 필요

Quantization Aware Training



(a) Integer-arithmetic-only inference

(b) Training with simulated quantization

Quantization Aware Training

```
# original model
# all tensors and computations are in floating point
previous_layer_fp32 -- linear_fp32 -- activation_fp32 -- next_layer_fp32
                 /
linear_weight_fp32

# dynamically quantized model
# linear and LSTM weights are in int8
previous_layer_fp32 -- linear_int8_w_fp32_inp -- activation_fp32 -- next_layer_fp32
                     /
   linear_weight_int8
```

Dynamic Quantization

```
# statically quantized model
# weights and activations are in int8
previous_layer_int8 -- linear_with_activation_int8 -- next_layer_int8
                   /
  linear_weight_int8
```

Static Quantization

```
# model with fake_quants for modeling quantization numerics during training
previous_layer_fp32 -- fq -- linear_fp32 -- activation_fp32 -- fq -- next_layer_fp32
                           /
   linear_weight_fp32 -- fq

# quantized model
# weights and activations are in int8
previous_layer_int8 -- linear_with_activation_int8 -- next_layer_int8
                     /
   linear_weight_int8
```

Quantization Aware Training

Experiments and Results

Learned Step Size Quantization

Steven K. Esser *, Jeffrey L. McKinstry, Deepika Bablani, Rathinakumar Appuswamy, Dharmendra S. Modha

IBM Research
San Jose, California, USA

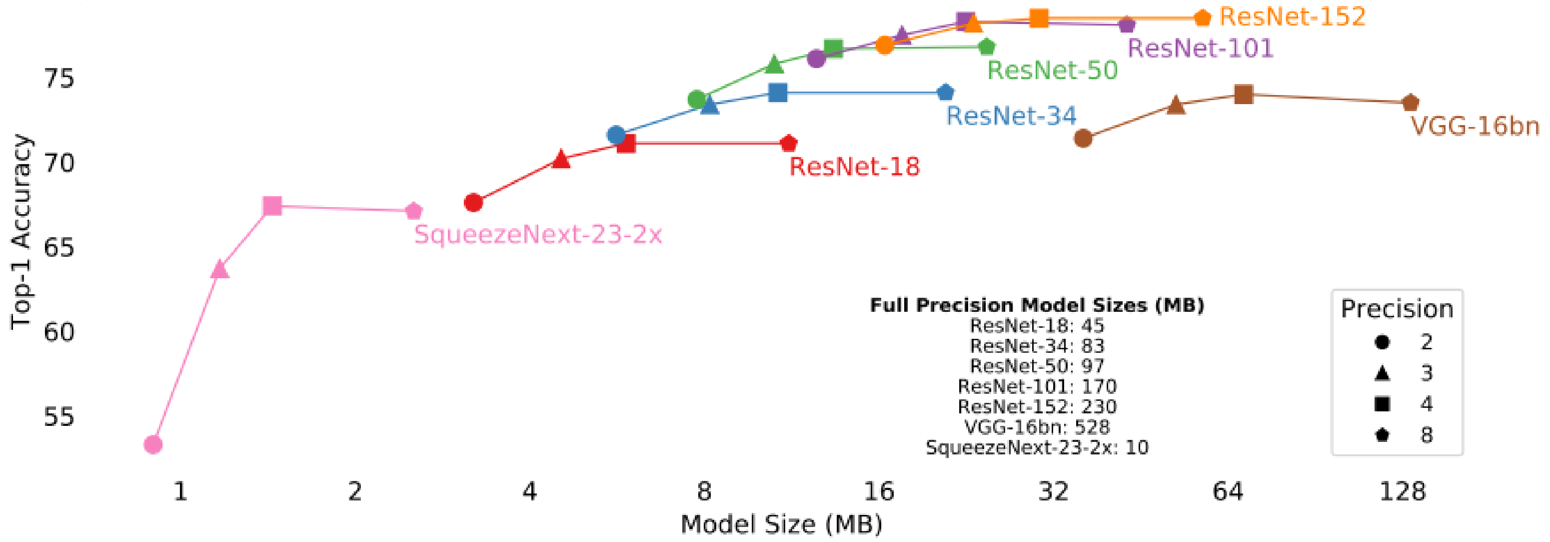


Figure 3: Accuracy vs. model size for the networks considered here show some 2-bit networks provide the highest accuracy at a given model size. Full precision model sizes are inset for reference.

References

PyTorch   Get Started   Ecosystem   Mobile   Blog   Tutorials   Docs   Resources   GitHub

Quantization leverages 8bit integer (int8) instructions to reduce the model size and run the inference faster (reduced latency) and can be the difference between a model achieving quality of service goals or even fitting into the resources available on a mobile device. Even when resources aren't quite so constrained it may enable you to deploy a larger and more accurate model. Quantization is available in PyTorch starting in version 1.3 and with the release of PyTorch 1.4 we published quantized models for ResNet, ResNext, MobileNetV2, GoogleNet, InceptionV3 and ShuffleNetV2 in the PyTorch torchvision 0.5 library.

References

[Paper]

Wu, H., Judd, P., Zhang, X., Isaev, M., & Micikevicius, P. (2020). Integer quantization for deep learning inference: Principles and empirical evaluation. *arXiv preprint arXiv:2004.09602*.

Esser, S. K., McKinstry, J. L., Bablani, D., Appuswamy, R., & Modha, D. S. (2019). Learned step size quantization. *arXiv preprint arXiv:1902.08153*.

[Documentation]

Quantization Recipe — PyTorch Tutorials 1.12.0+cu102 documentation
Quantization — PyTorch 1.12 documentation
Introduction to Quantization on PyTorch | PyTorch
양자화 인식 훈련 | TensorFlow Model Optimization
양자화 인식 훈련 종합 가이드 | TensorFlow Model Optimization
Keras 예제의 양자화 인식 훈련 | TensorFlow Model Optimization

[Note & Memo]

Distiller 모델 압축 기법 (3) : Quantization 양자화 – dankernel : Deep Tech Blog (sciomagelab.com)
딥러닝 Quantization(양자화) 정리 (velog.io)
Quantization (velog.io)

Q & A

Thank you ☺