Quantization

King-Siong Si

Institute of Multimedia Knowledge Fusion and Engineering,
Xi'an JiaoTong University

sjsinx@stu.xjtu.edu.cn

May 8, 2024

What Is Quantization

- quantization is the process of constraining an input from a continuous or otherwise large set of values (such as the real numbers) to a discrete set (such as the integers). [5]

What Is Quantization

- quantization is the process of constraining an input from a continuous or otherwise large set of values (such as the real numbers) to a discrete set (such as the integers). [5]
- key msg: a continuous set $\mapsto$ a discret set

What Is Quantization

- quantization is the process of constraining an input from a continuous or otherwise large set of values (such as the real numbers) to a discrete set (such as the integers). [5]
- key msg: a continuous set $\mapsto$ a discret set
- what is quantization in DL?

What Is Quantization

- quantization is the process of constraining an input from a continuous or otherwise large set of values (such as the real numbers) to a discrete set (such as the integers). [5]
- key msg: a continuous set $\mapsto$ a discret set
- what is quantization in DL?
- float $\mapsto$ int?

What Is Quantization

- quantization is the process of constraining an input from a continuous or otherwise large set of values (such as the real numbers) to a discrete set (such as the integers). [5]
- key msg: a continuous set $\mapsto$ a discret set
- what is quantization in DL?
- float $\mapsto$ int?
- use less bits

What Is Quantization

- quantization is the process of constraining an input from a continuous or otherwise large set of values (such as the real numbers) to a discrete set (such as the integers). [5]
- key msg: a continuous set $\mapsto$ a discret set
- what is quantization in DL?
- float $\mapsto$ int?
- use less bits
- in general: FP32/FP16 $\mapsto$ INT8/INT4

Uniform Quantization

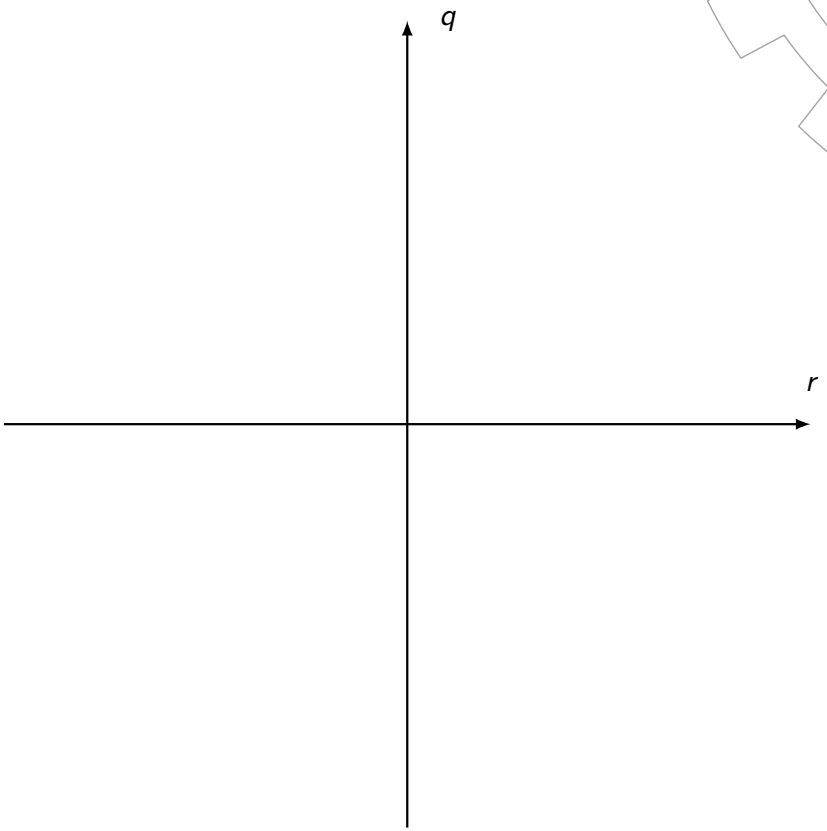
- uniform: the resulting quantized values are uniformly spaced

Uniform Quantization

- uniform: the resulting quantized values are uniformly spaced
- any function? a rounding function

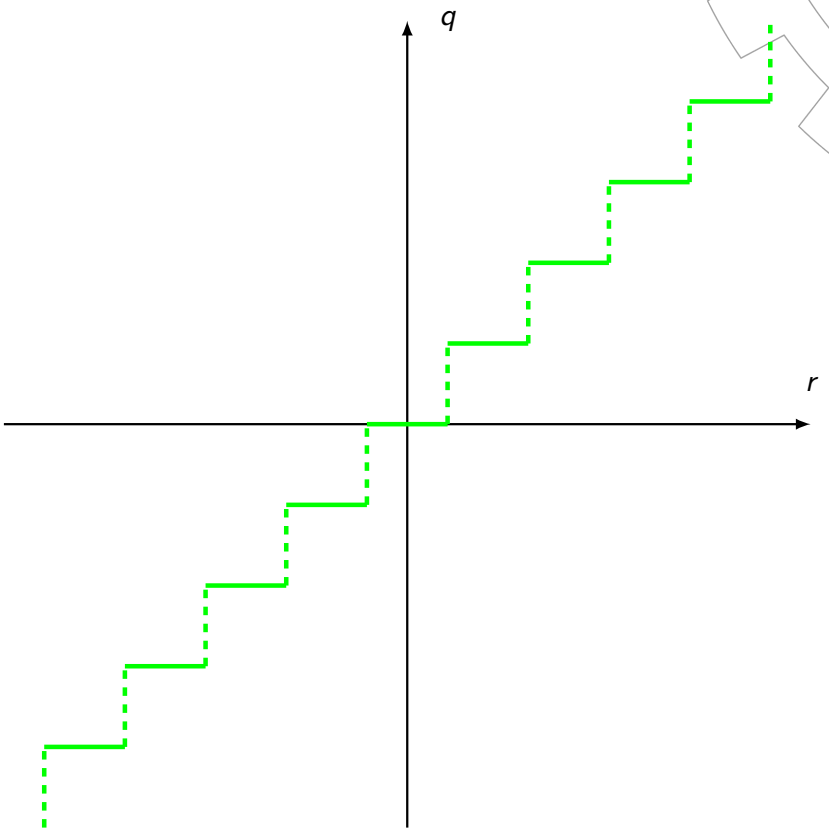
Uniform Quantization

- uniform: the resulting quantized values are uniformly spaced
- any function? a rounding function



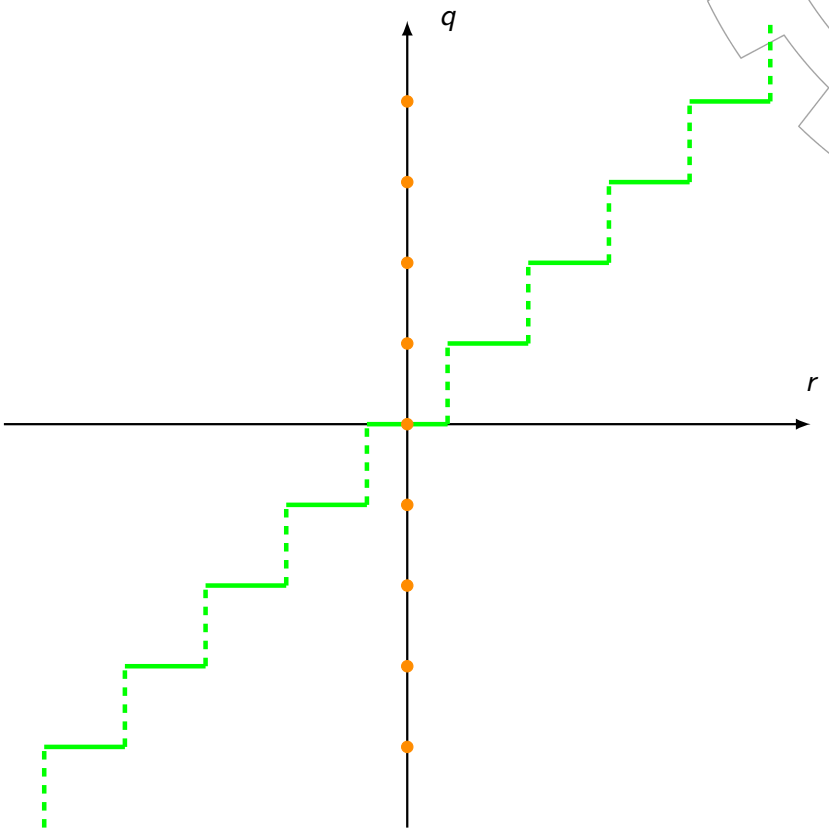
Uniform Quantization

- uniform: the resulting quantized values are uniformly spaced
- any function? a rounding function



Uniform Quantization

- uniform: the resulting quantized values are uniformly spaced
- any function? a rounding function



Uniform Quantization

- range dilemma: FLOAT $(-\infty, +\infty)$ while INT $[-128, 127]$

Uniform Quantization

- range dilemma: FLOAT $(-\infty, +\infty)$ while INT $[-128, 127]$

quantization function

$$q = \lfloor r/S \rceil - Z \tag{1}$$

Uniform Quantization

- range dilemma: FLOAT $(-\infty, +\infty)$ while INT $[-128, 127]$

quantization function

$$q = \lfloor r/S \rceil - Z \tag{1}$$

- where S, Z come from?

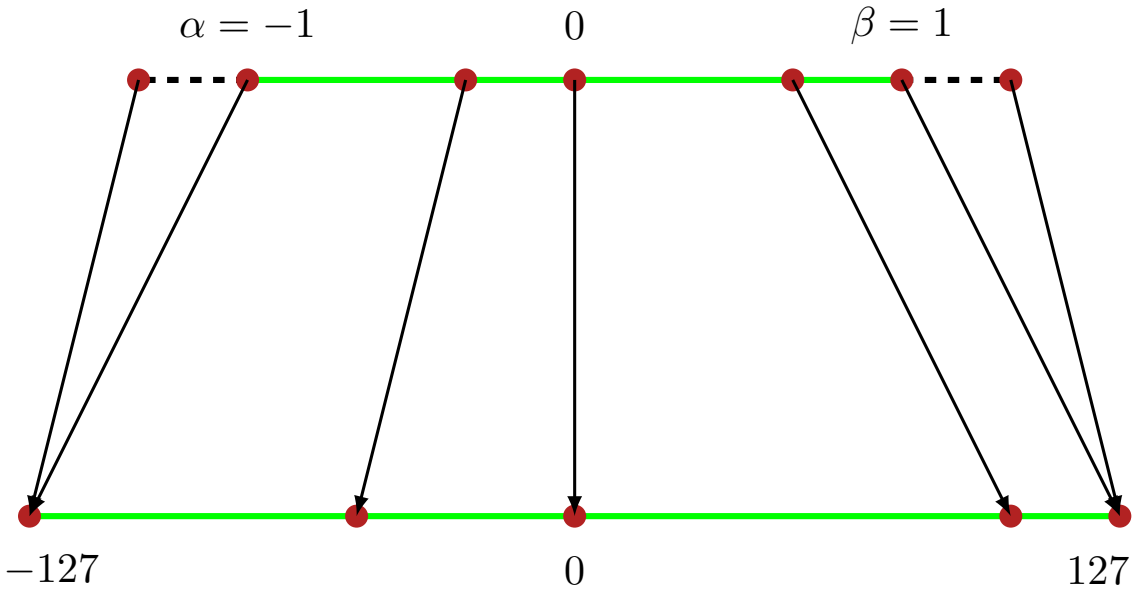
Uniform Quantization

- range dilemma: FLOAT $(-\infty, +\infty)$ while INT $[-128, 127]$

quantization function

$$q = \lfloor r/S \rceil - Z \tag{1}$$

- where S, Z come from?



Uniform Quantization

S formula

$$S = \frac{|\beta - \alpha|}{2^l - 1} \tag{2}$$

Uniform Quantization

S formula

$$S = \frac{|\beta - \alpha|}{2^l - 1} \tag{2}$$

- what is the function of Z?

Uniform Quantization

S formula

$$S = \frac{|\beta - \alpha|}{2^l - 1} \tag{2}$$

- what is the function of Z?
- symmetric or asymmetric

Uniform Quantization

S formula

$$S = \frac{|\beta - \alpha|}{2^l - 1} \tag{2}$$

- what is the function of Z?
- symmetric or asymmetric
- symmetric: $|\alpha| = |\beta|, Z = 0$
- asymmetric: $|\alpha| \neq |\beta|, Z \neq 0$

Rethink of Quantization

- quantize a model

Rethink of Quantization

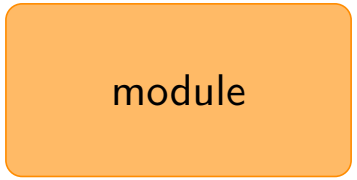
- quantize a model
- quantize a module in the model, e.g., a convolution layer

Rethink of Quantization

- quantize a model
- quantize a module in the model, e.g., a convolution layer
- regard a convolution layer as a matrix

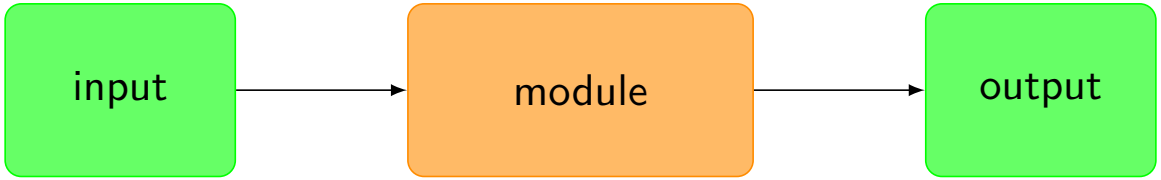
Rethink of Quantization

- quantize a model
- quantize a module in the model, e.g., a convolution layer
- regard a convolution layer as a matrix



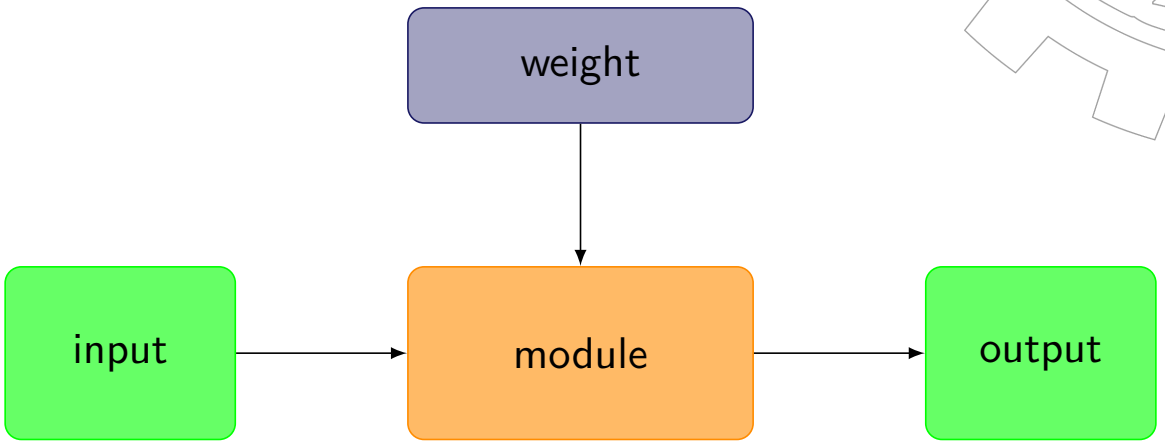
Rethink of Quantization

- quantize a model
- quantize a module in the model, e.g., a convolution layer
- regard a convolution layer as a matrix



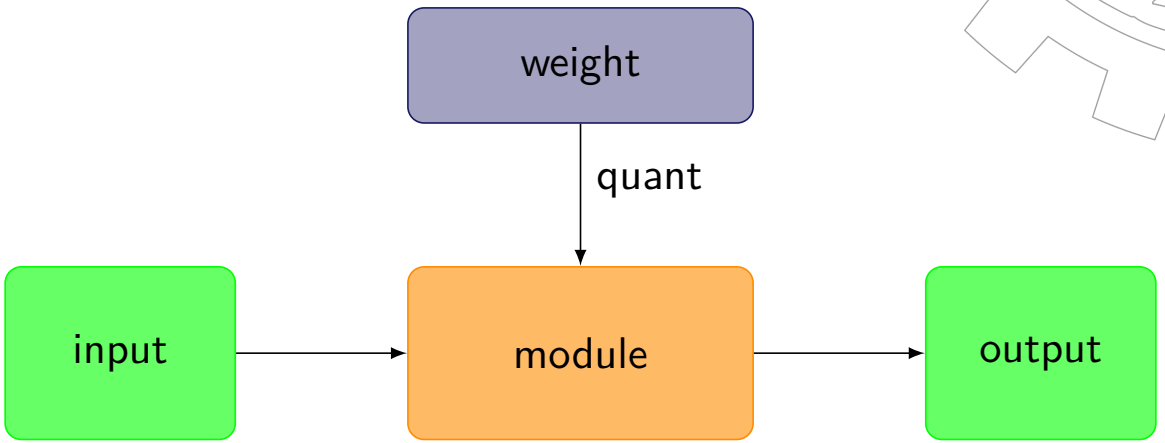
Rethink of Quantization

- quantize a model
- quantize a module in the model, e.g., a convolution layer
- regard a convolution layer as a matrix



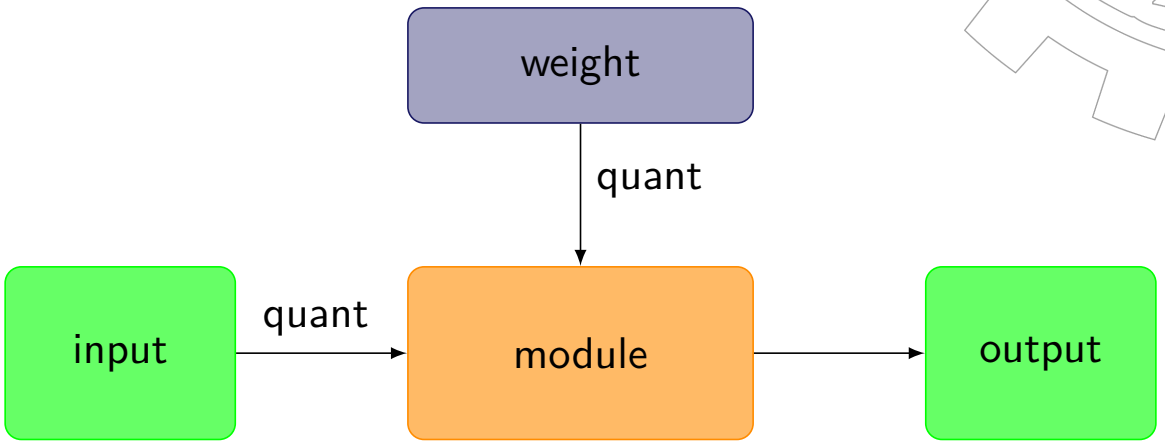
Rethink of Quantization

- quantize a model
- quantize a module in the model, e.g., a convolution layer
- regard a convolution layer as a matrix



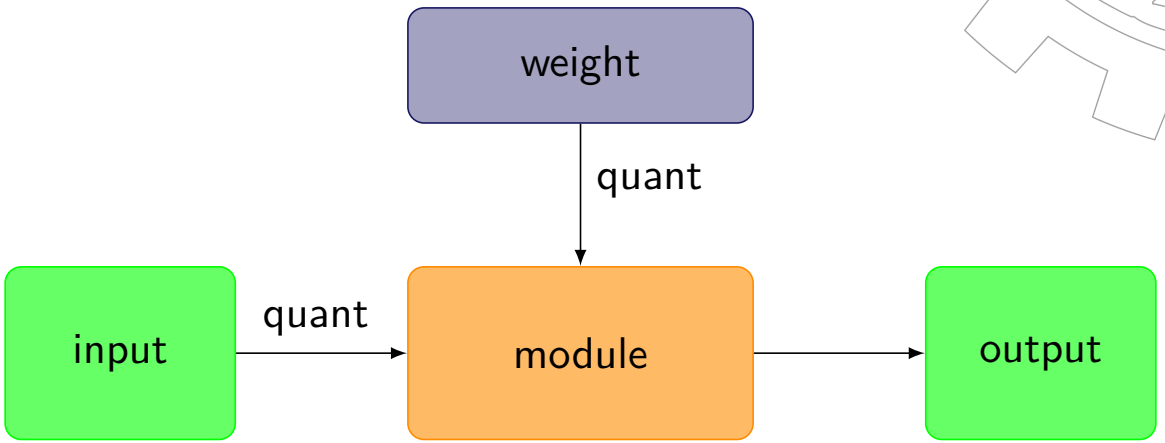
Rethink of Quantization

- quantize a model
- quantize a module in the model, e.g., a convolution layer
- regard a convolution layer as a matrix



Rethink of Quantization

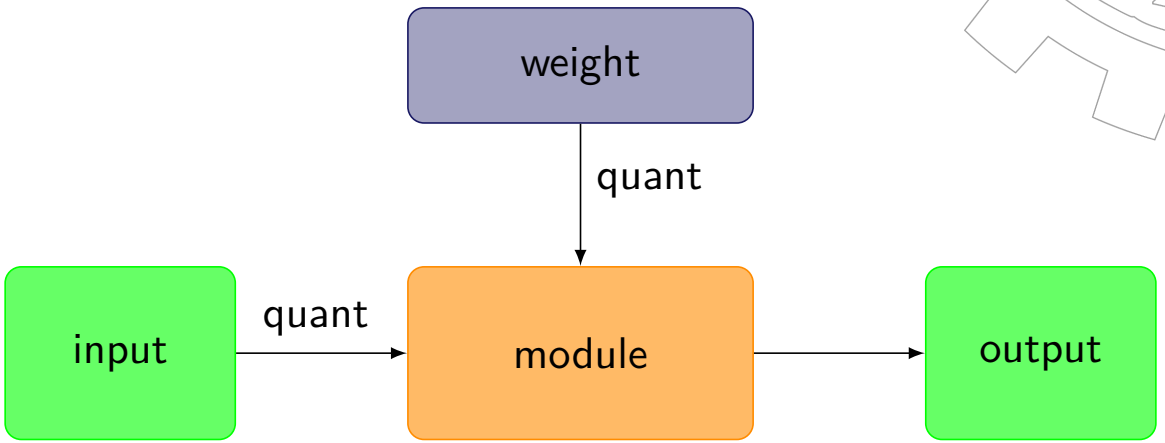
- quantize a model
- quantize a module in the model, e.g., a convolution layer
- regard a convolution layer as a matrix



- $\tilde{r} = (q + Z) \times S$

Rethink of Quantization

- quantize a model
- quantize a module in the model, e.g., a convolution layer
- regard a convolution layer as a matrix



- $\tilde{r} = (q + Z) \times S$
- fake quantization, Q/DQ

Rethink of Quantization

- integer-arithmetic-only quantization[4]

Rethink of Quantization

- integer-arithmetic-only quantization[4]
- $r_\alpha^{(i,j)} = S_\alpha(q_\alpha^{(i,j)} + Z_\alpha)$

Rethink of Quantization

- integer-arithmetic-only quantization[4]
- $r_\alpha^{(i,j)} = S_\alpha(q_\alpha^{(i,j)} + Z_\alpha)$

matrix multiplication

$$S_o(q_o^{(i,j)} + Z_o) = \sum_{k=1}^{n} S_i(q_i^{(i,k)} + Z_i) S_w(q_w^{(k,j)} + Z_w) \tag{3}$$

$$\Rightarrow q_o^{(i,j)} = Z_o + S_i S_w / S_o \sum_{k=1}^{n} (q_i^{(i,k)} + Z_i)(q_w^{(k,j)} + Z_w) \tag{4}$$

Rethink of Quantization

- integer-arithmetic-only quantization[4]
- $r_\alpha^{(i,j)} = S_\alpha(q_\alpha^{(i,j)} + Z_\alpha)$

matrix multiplication

$$S_o(q_o^{(i,j)} + Z_o) = \sum_{k=1}^{n} S_i(q_i^{(i,k)} + Z_i)S_w(q_w^{(k,j)} + Z_w) \tag{3}$$

$$\Rightarrow q_o^{(i,j)} = Z_o + S_i S_w / S_o \sum_{k=1}^{n} (q_i^{(i,k)} + Z_i)(q_w^{(k,j)} + Z_w) \tag{4}$$

- let $M := \frac{S_i S_w}{S_o}$

Rethink of Quantization

- integer-arithmetic-only quantization[4]
- $r_\alpha^{(i,j)} = S_\alpha(q_\alpha^{(i,j)} + Z_\alpha)$

matrix multiplication

$$S_o(q_o^{(i,j)} + Z_o) = \sum_{k=1}^{n} S_i(q_i^{(i,k)} + Z_i)S_w(q_w^{(k,j)} + Z_w) \tag{3}$$

$$\Rightarrow q_o^{(i,j)} = Z_o + S_i S_w / S_o \sum_{k=1}^{n} (q_i^{(i,k)} + Z_i)(q_w^{(k,j)} + Z_w) \tag{4}$$

- let $M := \frac{S_i S_w}{S_o}$
- let $M = 2^{-B} M_0 \Leftrightarrow M_0 = 2^B M$[4],

How to Get Parameters

- what should we know? α and β

How to Get Parameters

- what should we know? α and β
- if we know the distribution,

How to Get Parameters

- what should we know? α and β
- if we know the distribution,
  - min-max

How to Get Parameters

- what should we know? α and β
- if we know the distribution,
  - min-max
  - entropy

How to Get Parameters

- what should we know? α and β
- if we know the distribution,
  - min-max
  - entropy
  - percentile

How to Get Parameters

- what should we know? α and β
- if we know the distribution,
  - min-max
  - entropy
  - percentile
- anyway, we can know what parameters are if we know the data

How to Get Parameters

- what should we know? α and β
- if we know the distribution,
  - min-max
  - entropy
  - percentile
- anyway, we can know what parameters are if we know the data
- activation of a module?

How to Get Parameters

- what should we know? α and β
- if we know the distribution,
  - min-max
  - entropy
  - percentile
- anyway, we can know what parameters are if we know the data
- activation of a module?
- we do need DATA!

How to Get Parameters

- what should we know? α and β
- if we know the distribution,
  - min-max
  - entropy
  - percentile
- anyway, we can know what parameters are if we know the data
- activation of a module?
- we do need DATA!
- PTQ (Post-Training Quantization)[1]

How to Get Parameters

- what should we know? α and β
- if we know the distribution,
  - min-max
  - entropy
  - percentile
- anyway, we can know what parameters are if we know the data
- activation of a module?
- we do need DATA!
- PTQ (Post-Training Quantization)[1]
- calibration data (mainly use a part of training data)

How to Get Parameters

- QAT (Quantization-Aware Training)[1]

How to Get Parameters

- QAT (Quantization-Aware Training)[1]
- after PTQ, fine tune the model[2]

How to Get Parameters

- QAT (Quantization-Aware Training)[1]
- after PTQ, fine tune the model[2]
- NOT mutually exclusive

How to Get Parameters

- QAT (Quantization-Aware Training)[1]
- after PTQ, fine tune the model[2]
- NOT mutually exclusive

It is some time known as "quantization aware training". We don't use the name because it doesn't reflect the underneath assumption. If anything, it makes training being "unaware" of quantization because of the STE approximation. — pytorch-quantization's documentation

Pipeline

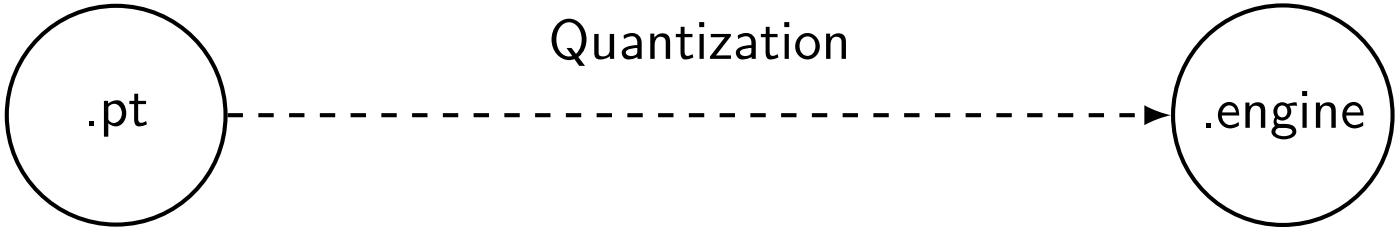
- NVIDIA Inc. is astonishing

Pipeline

- NVIDIA Inc. is astonishing
- NVIDIA invented a new concept

Pipeline

- NVIDIA Inc. is astonishing
- NVIDIA invented a new concept
- explicit/implicit quantization[3]

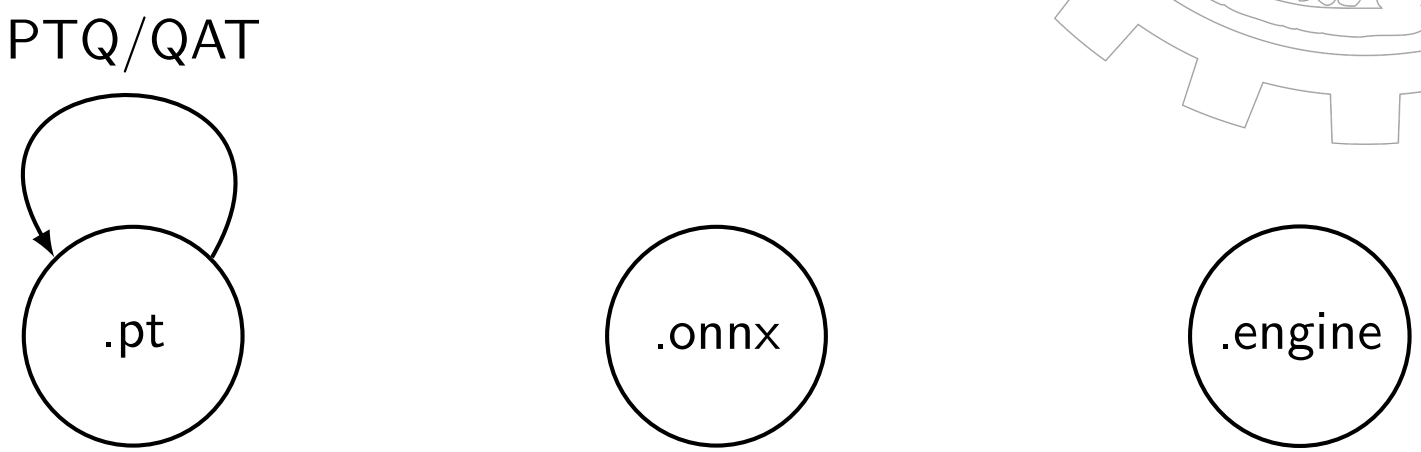


Pipeline

- NVIDIA Inc. is astonishing
- NVIDIA invented a new concept
- explicit/implicit quantization[3]

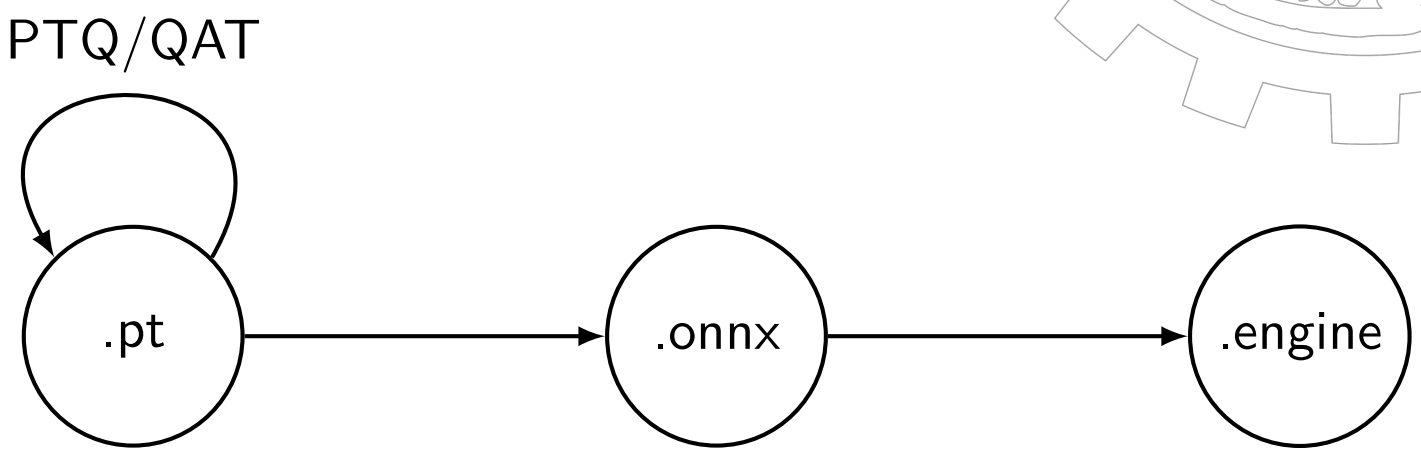
Pipeline

- NVIDIA Inc. is astonishing
- NVIDIA invented a new concept
- explicit/implicit quantization[3]



Pipeline

- NVIDIA Inc. is astonishing
- NVIDIA invented a new concept
- explicit/implicit quantization[3]



[1] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer.
A survey of quantization methods for efficient neural network inference.
In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022.

[2] NVIDIA Inc.
pytorch-quantization's documentation, 2021.
[Online; accessed 7-May-2024].

[3] NVIDIA Inc.
Nvidia tensorrt documentation, 2024.
[Online; accessed 7-May-2024].

[4] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko.
Quantization and training of neural networks for efficient integer-arithmetic-only inference.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.

[5] Wikipedia contributors.
Quantization — Wikipedia, the free encyclopedia, 2023.
[Online; accessed 7-May-2024].

Thanks!