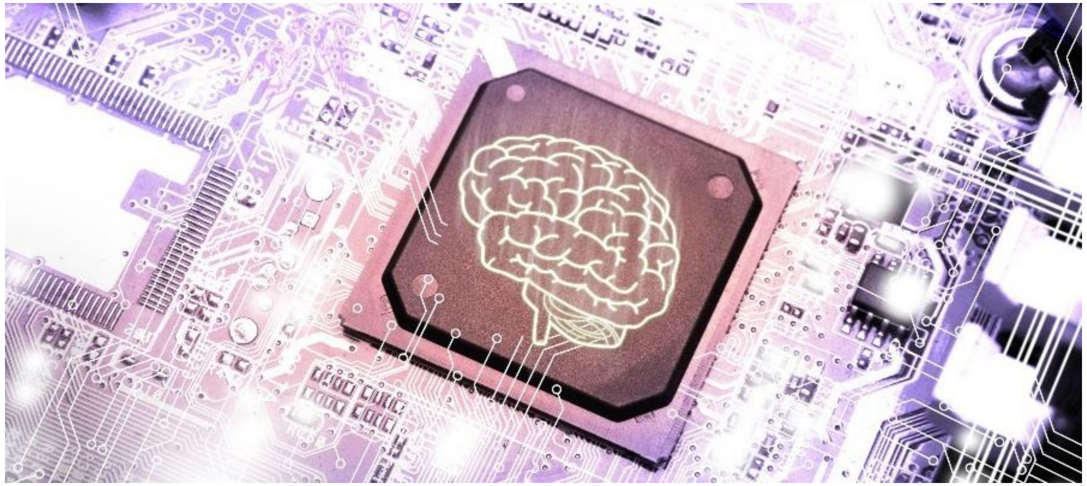


QUANTIZATION METHODOLOGIES

Vincent TEMPLIER
CEA LIST
vincent.templier@cea.fr

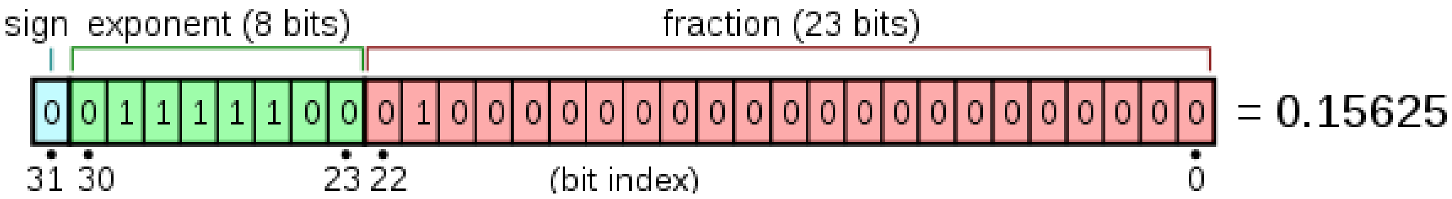
SUMMARY

1. General introduction
2. Post-Training Quantization (PTQ)
3. Quantization Aware-Training (QAT)
4. Experiments and validation
5. What’s next ?

GENERAL INTRODUCTION

- Default arithmetic in deep learning frameworks is 32-bit floating point (float32) or single precision

- Float32:
  - 1 bit for the sign
  - 8 bits for the exponent
  - 24 bits for the fraction
  - Max value: $3.4 * 10^{38}$



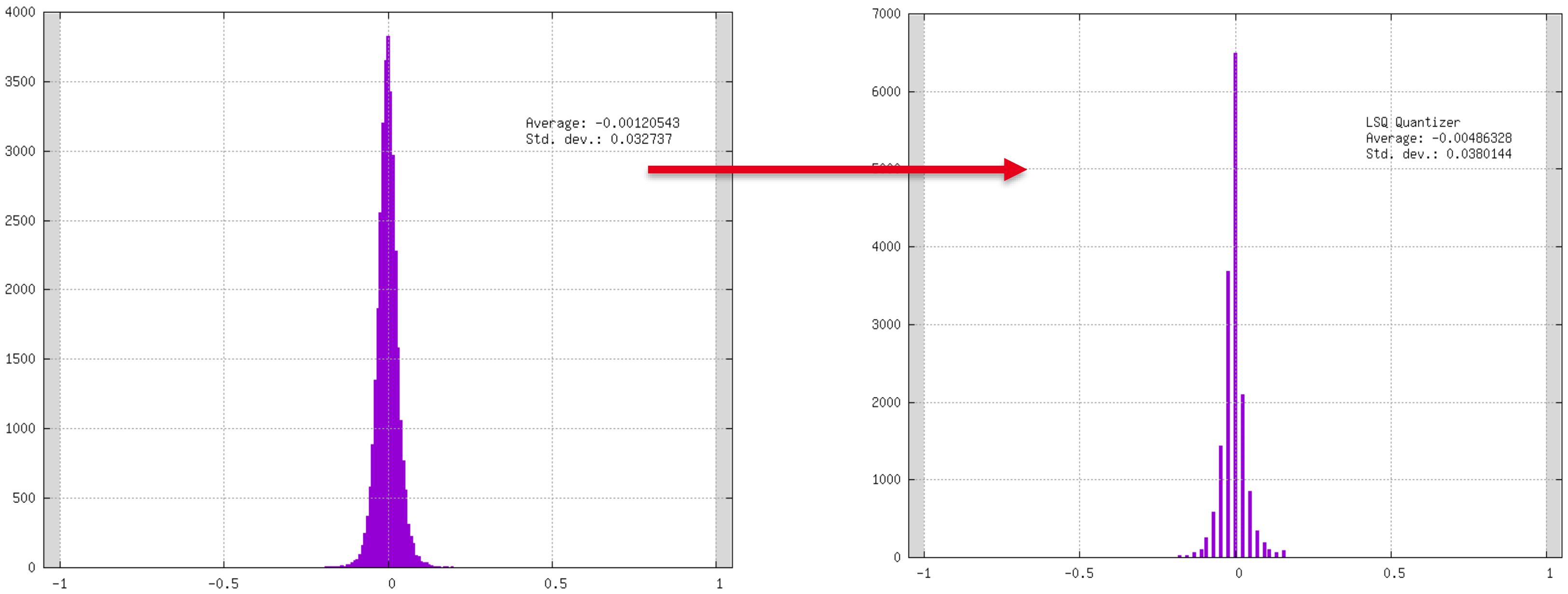
- Integer 8 (int8)
  - 8 bits (no fraction)
  - Max value: 256 values

- And even smaller formats

GENERAL INTRODUCTION

- In theory, quantization is simple

- Digital data reduction

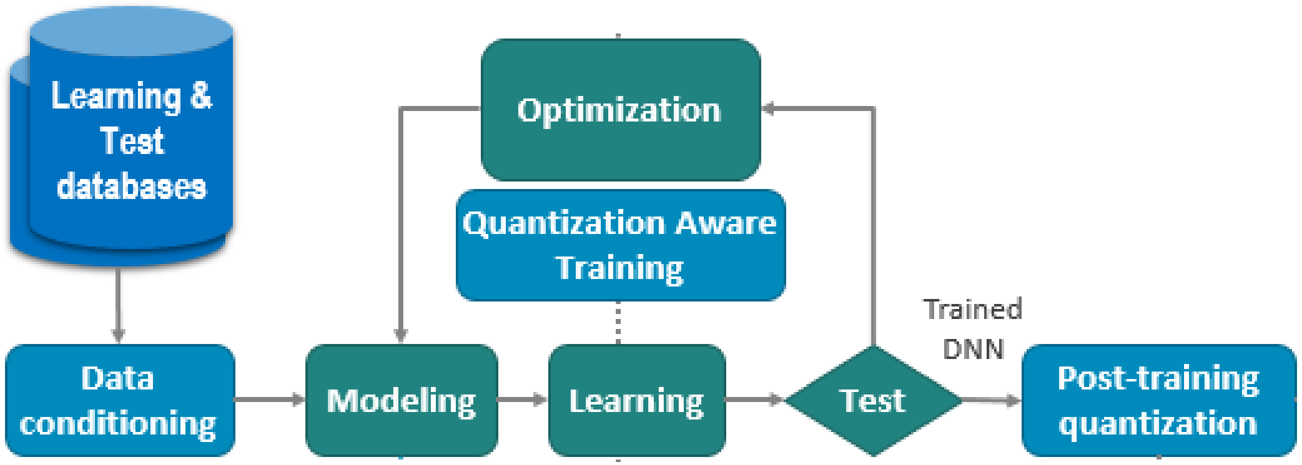


Quantization 4 bits

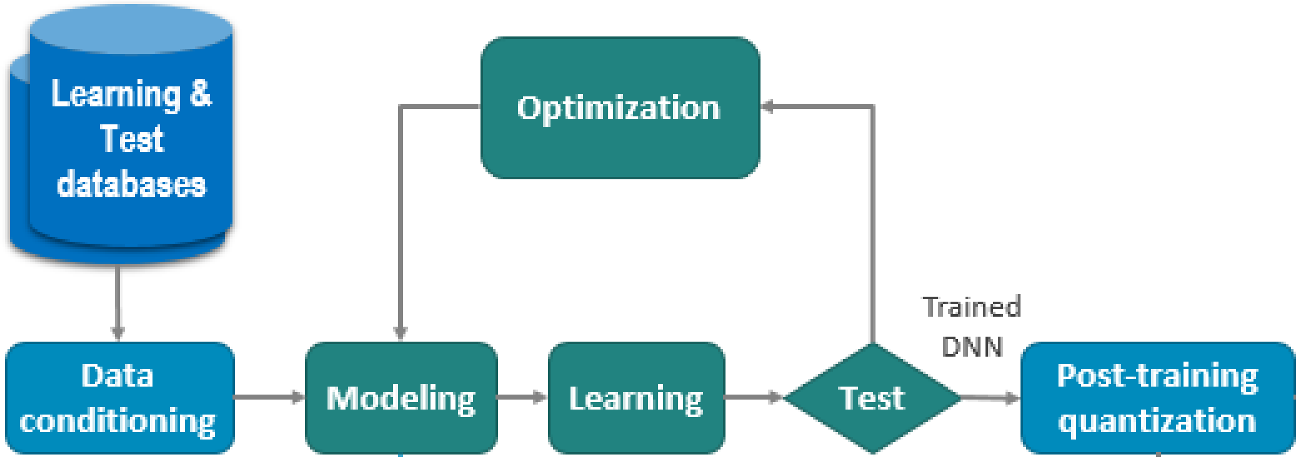
- Significant reduction in memory footprint, power consumption and gains in computational speed

GENERAL INTRODUCTION

- In practice, quantization is complicated
- 2 types of quantization



POST-TRAINING QUANTIZATION (PTQ)



POST-TRAINING QUANTIZATION (PTQ)

- **Post-training quantization algorithm in 3 steps**
  - Weights normalization
    - All weights are rescaled in the range [-1.0, 1.0]
    - Per layer normalization
    - Per layer and per output channel normalization
      ➔ finer grain, better usage of the quantized range for some output channels

  - Activations normalization
    - Activations at each layer are rescaled in the range [-1.0, 1.0] for signed outputs
      and [0.0, 1.0] for unsigned outputs
    - Find **optimal quantization threshold value** of the activation output of each layer
      ➔ using the validation dataset
    - Iterative process: need to take into account previous layers normalizing factors

  - Quantization
    - Inputs, weights, biases and activations are quantized to the desired *nbbits* precision
    - Convert ranges from [-1.0, 1.0] and [0.0, 1.0] to $[-2^{nbbits-1}-1,\, 2^{nbbits-1}-1]$ and $[0,\, 2^{nbbits}-1]$ taking into account all dependencies

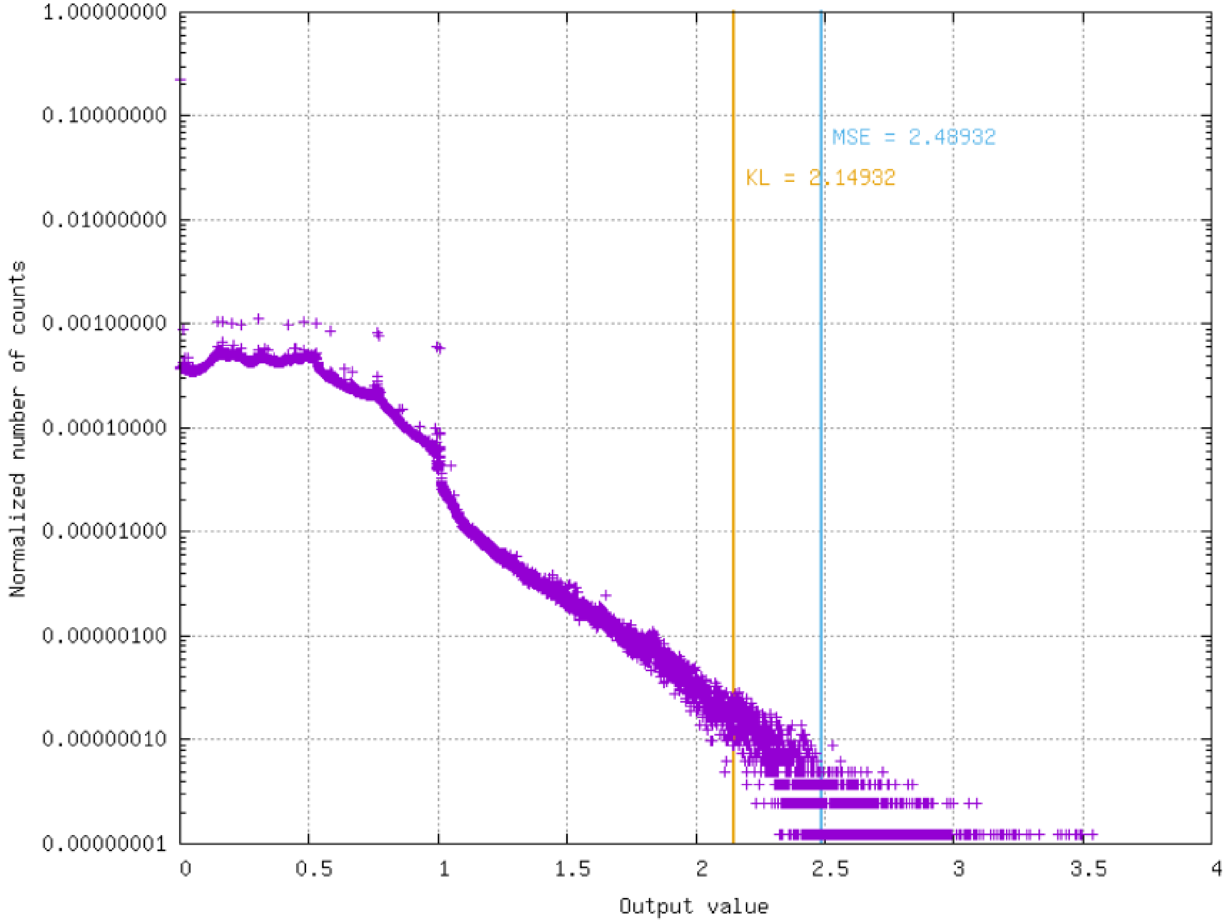
POST-TRAINING QUANTIZATION (PTQ)

- **Find optimal quantization threshold value of the activation output of each layer**
  - Compute histogram of activation values
  - Find threshold that minimizes distance between original distribution and clipped quantized distribution

    ➔ two distance algorithms can be used:

    - Mean Squared Error (MSE)
    - Kullback–Leibler divergence metric (KL-divergence)

Threshold value = activation scaling factor to be taken into account during quantization



POST-TRAINING QUANTIZATION (PTQ)

- **Additional optimization strategies**

  - Weights clipping *(optional)*
    same as activations: find optimal quantization threshold value

  - Activation scaling factor approximation
    - Fixed-point
      α ➔ $x2^{-p}$

    - Single-shift
      α ➔ 2^{x}

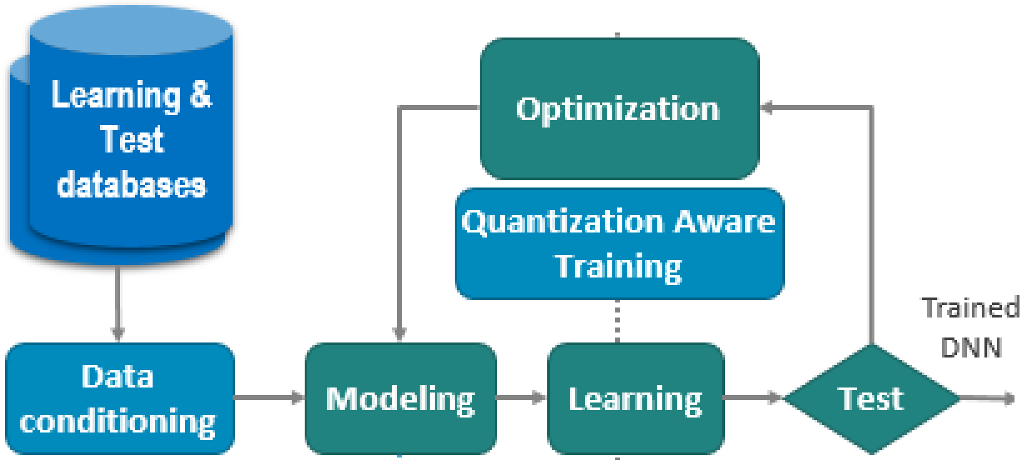
    - Double-shift
      α ➔ $2^{n} + 2^{m}$

POST-TRAINING QUANTIZATION (PTQ)

- Avantages:
  - No retraining needed
  - Quite fast to have a result

- Drawback:
  - Limited to 8-bit, very complicated to have good results with a lower precision

QUANTIZATION AWARE-TRAINING (QAT)

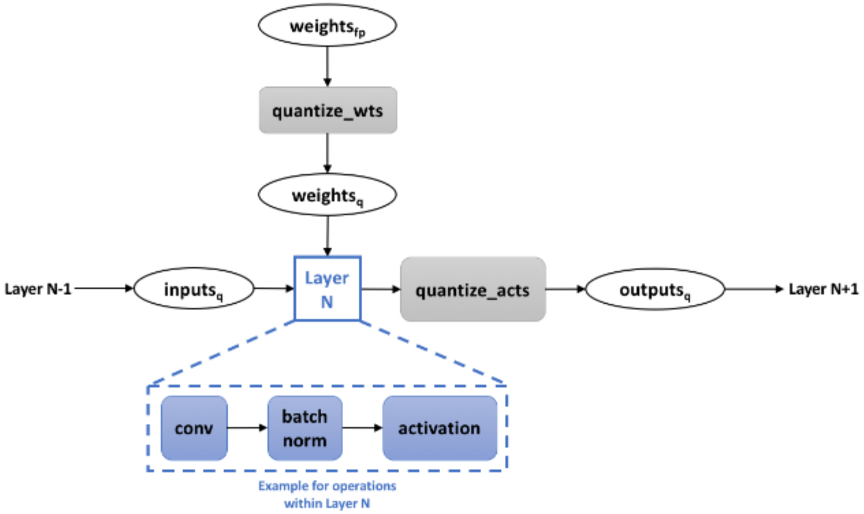


QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?

- Full precision copy of the weights maintained throughout the training process
  - Back-propagation computes the gradients of full precision weights to accumulate the small changes from the gradients
  - Forward propagation pass however simulates quantized inference
  - Weights quantized before being convolved with the input

- Activations are quantized following similar logic



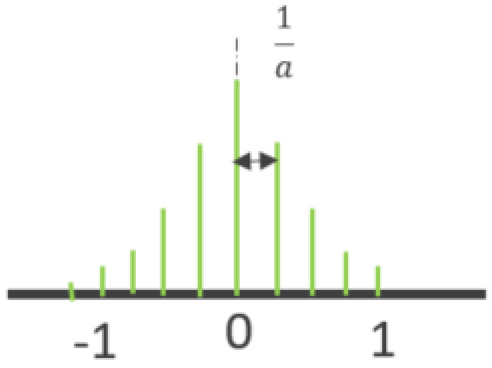
Quantization Aware Training Flow

QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?

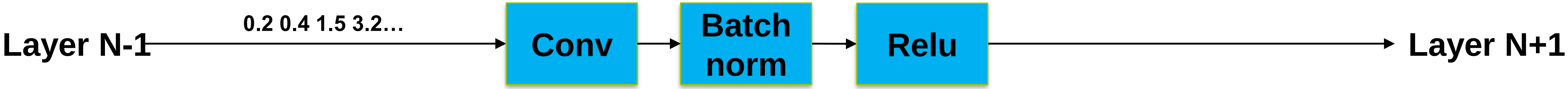
- Parameter « *a* » = distance between the quantization points

- Used to quantize the weights or/and activations



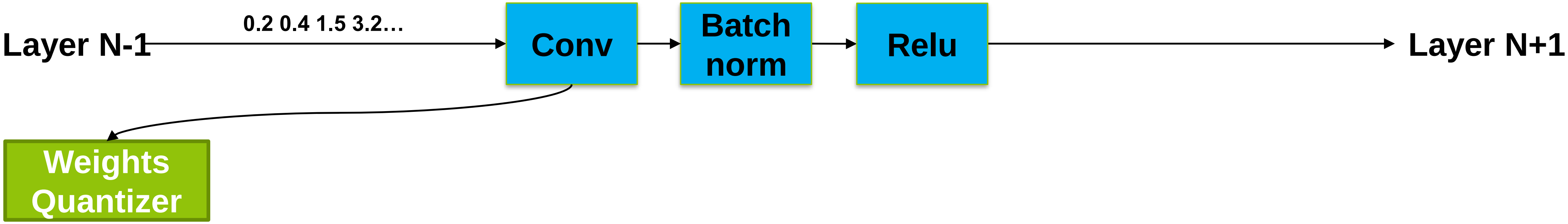
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



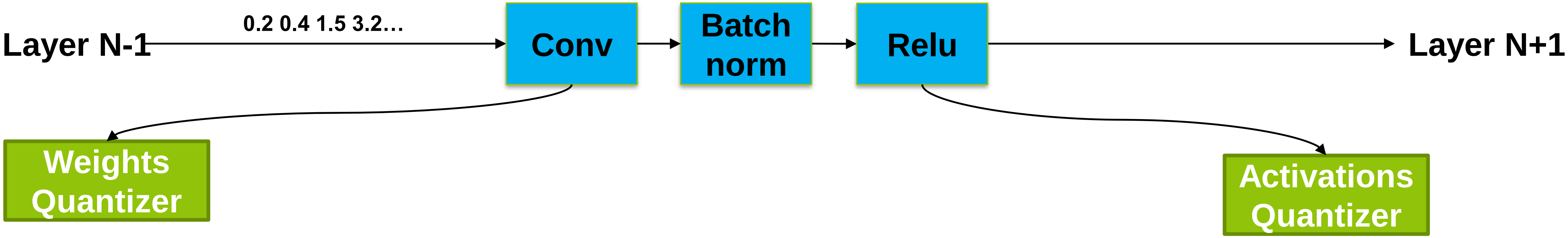
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



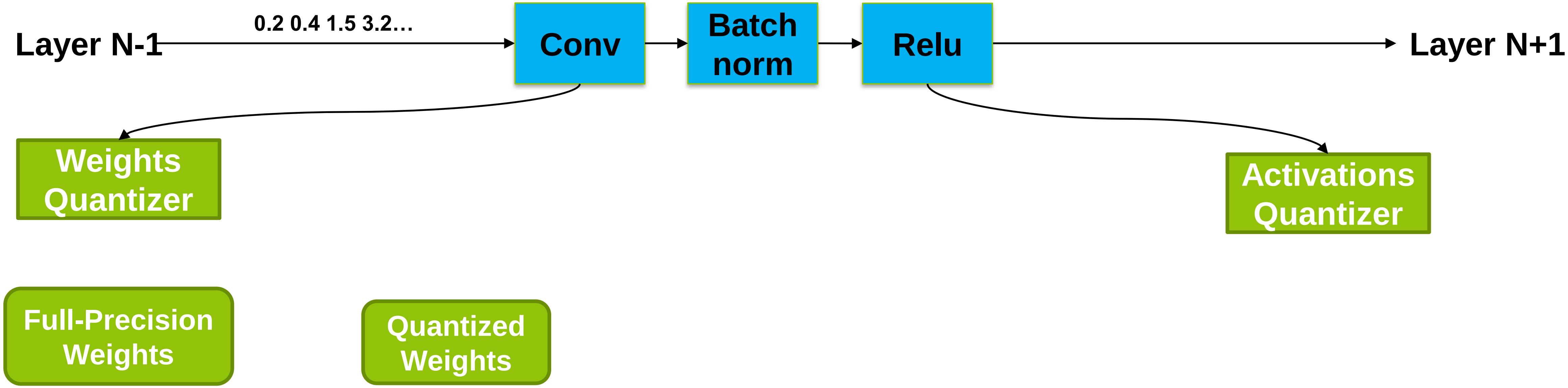
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



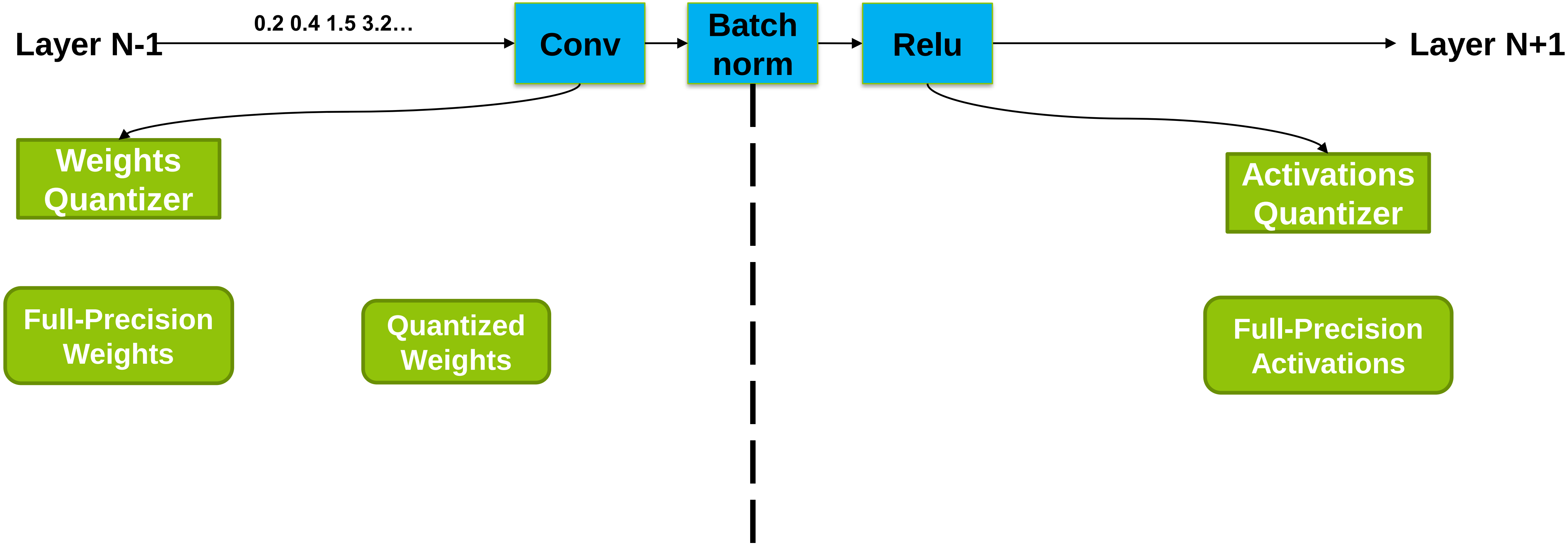
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



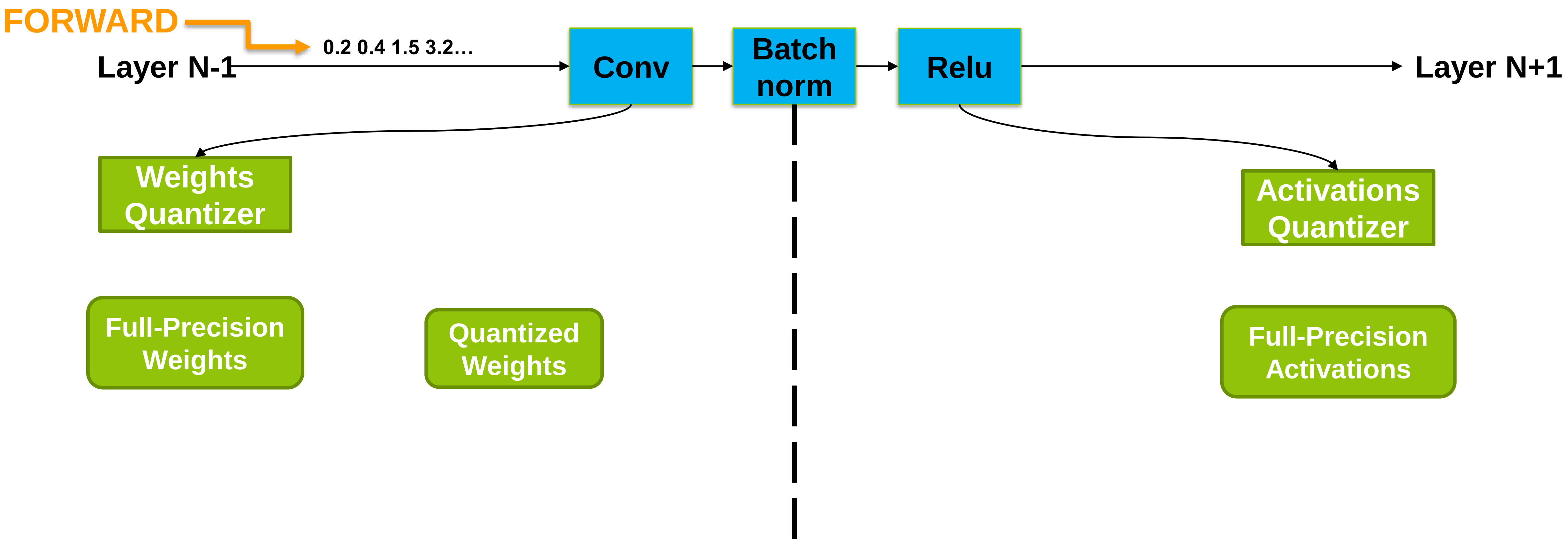
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



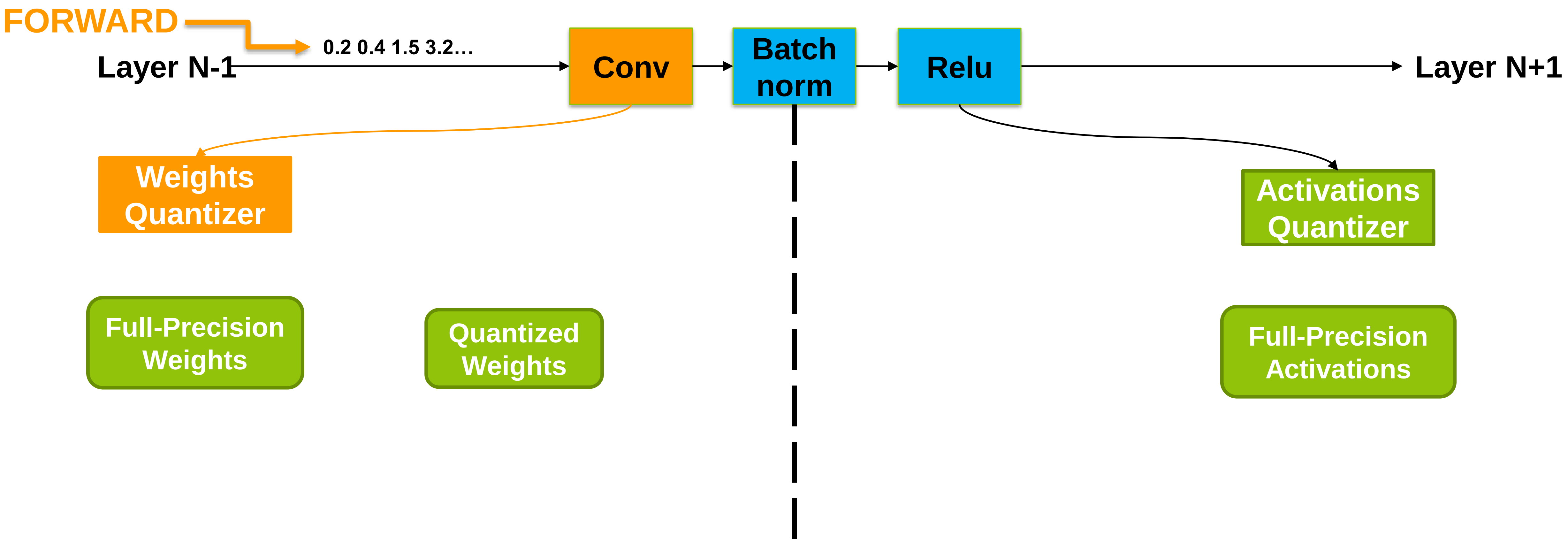
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



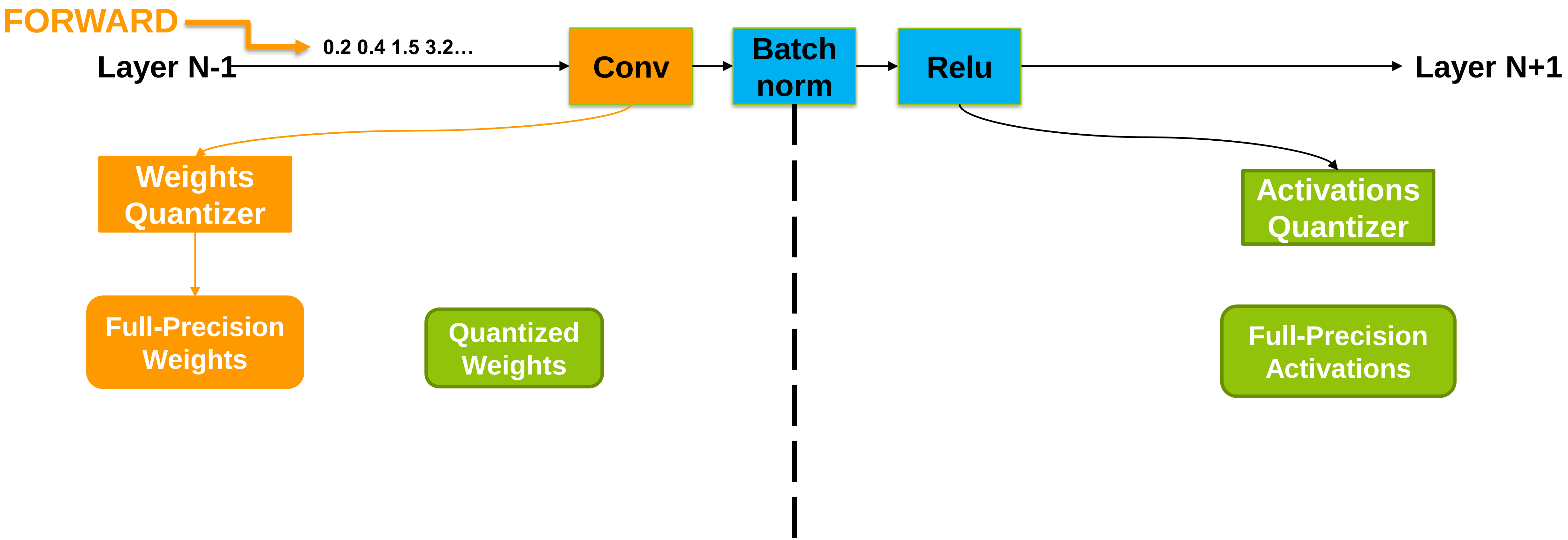
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



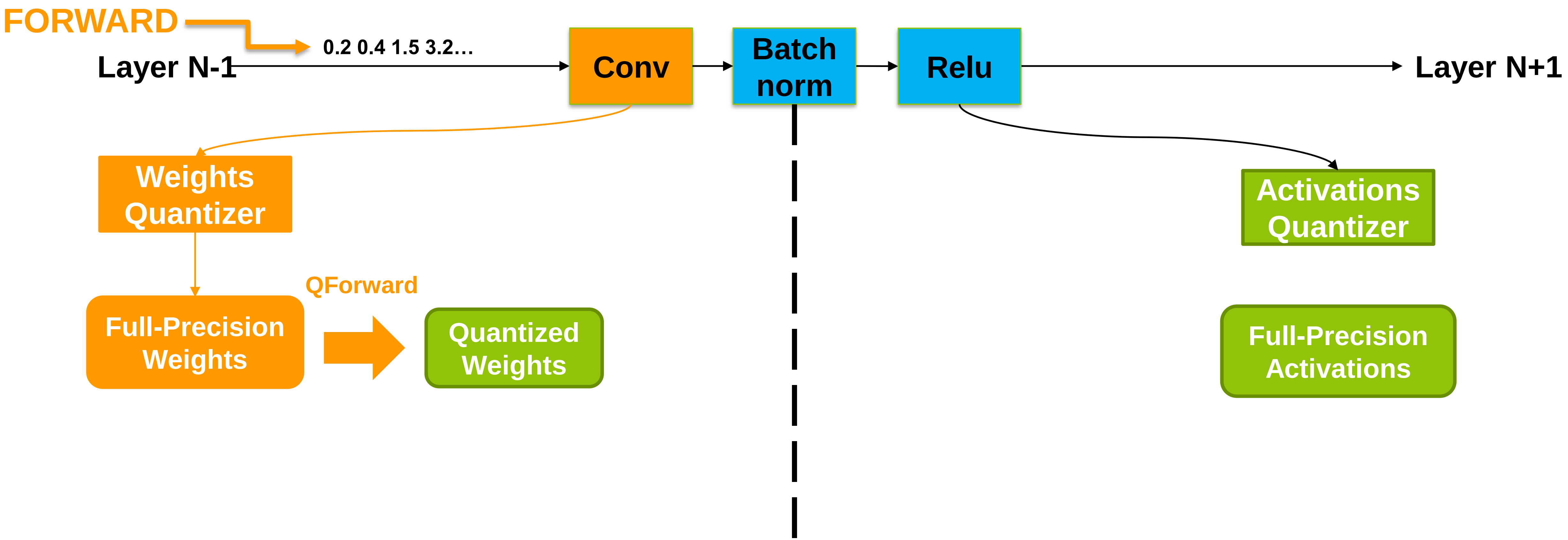
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



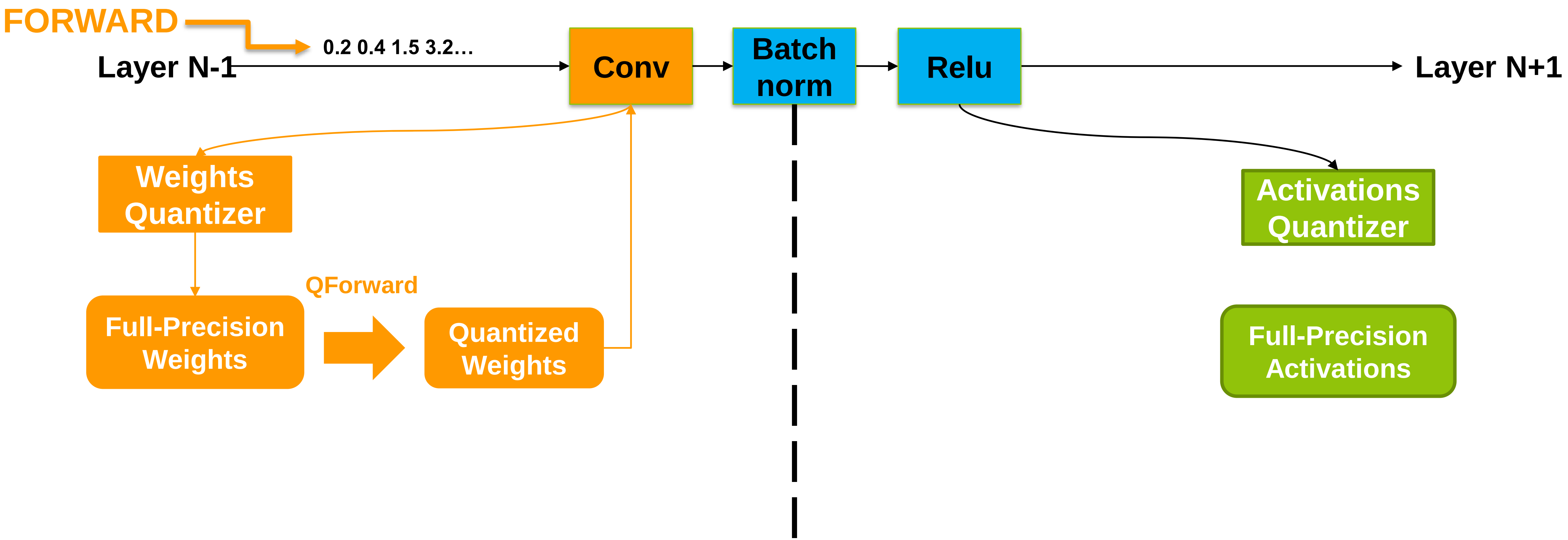
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



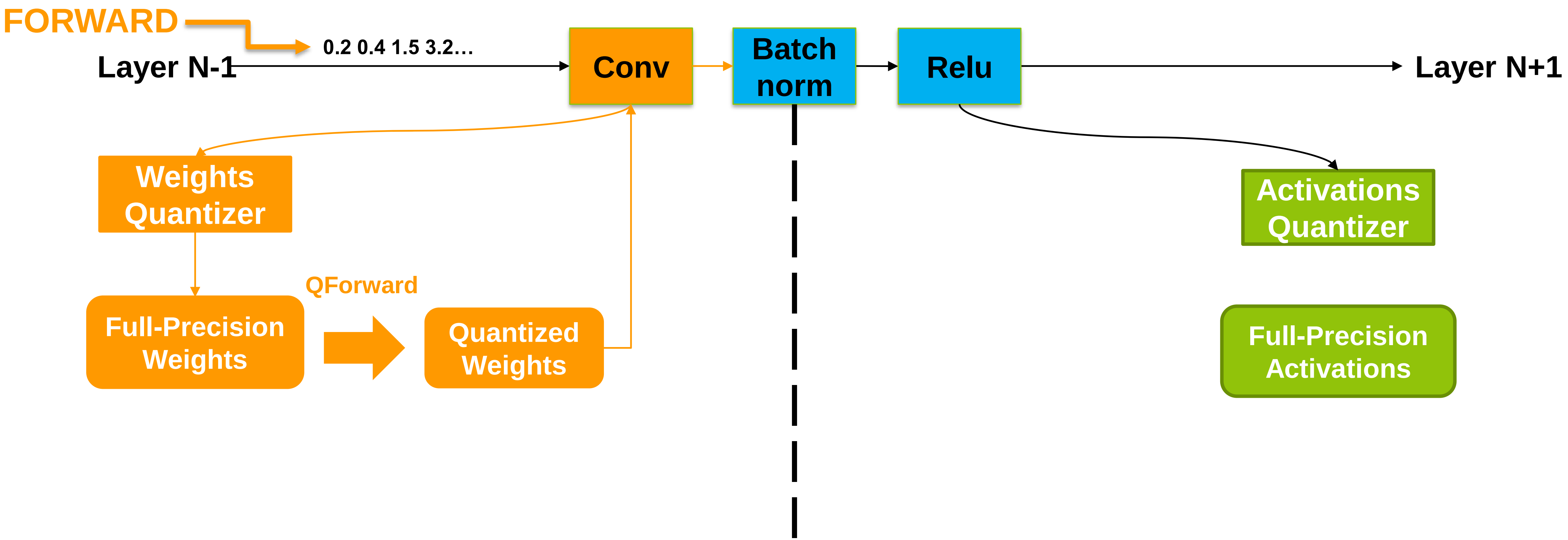
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



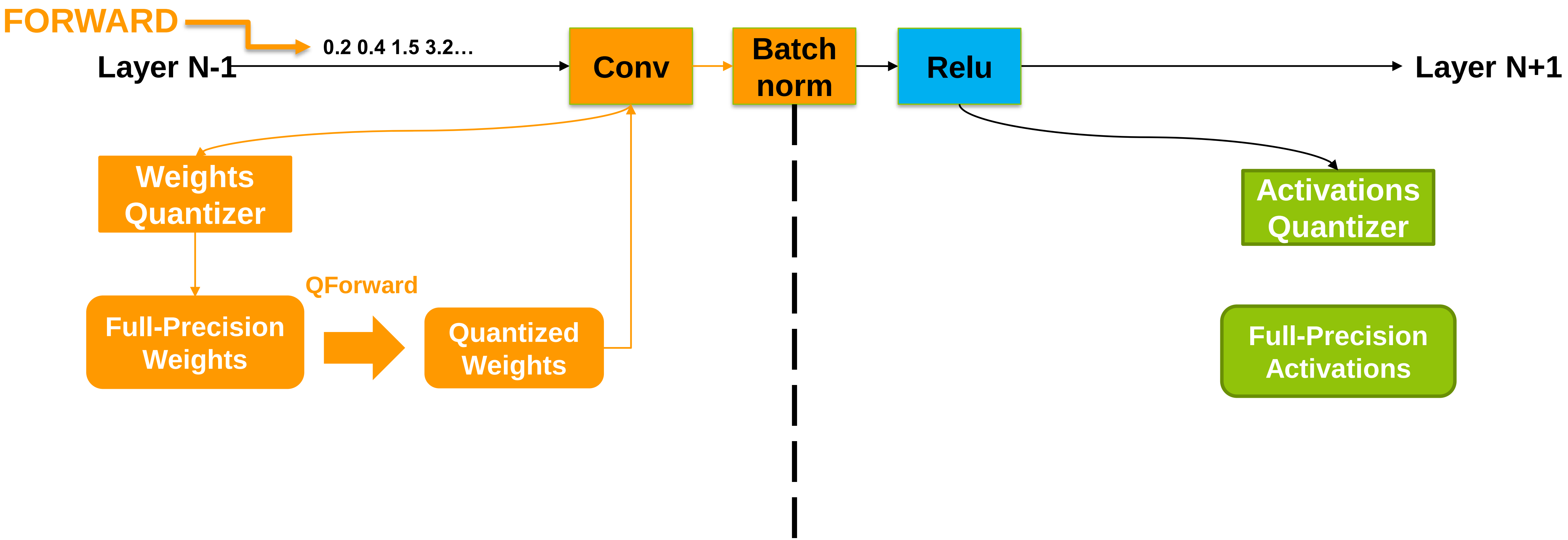
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



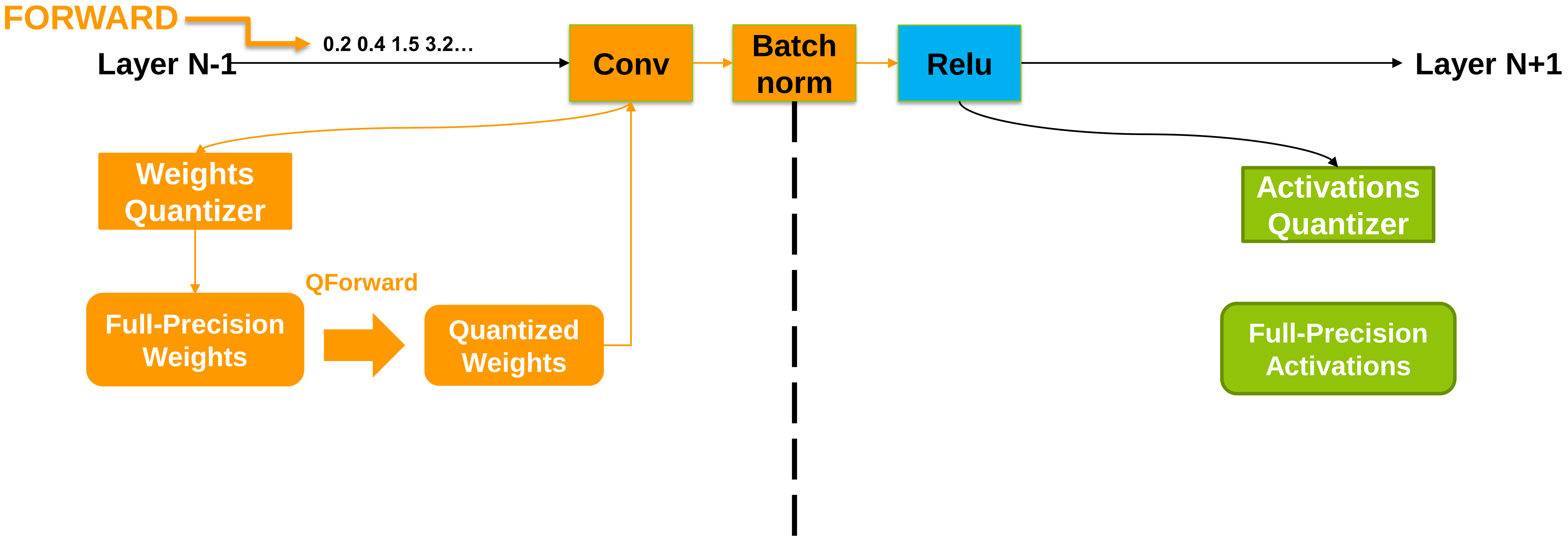
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



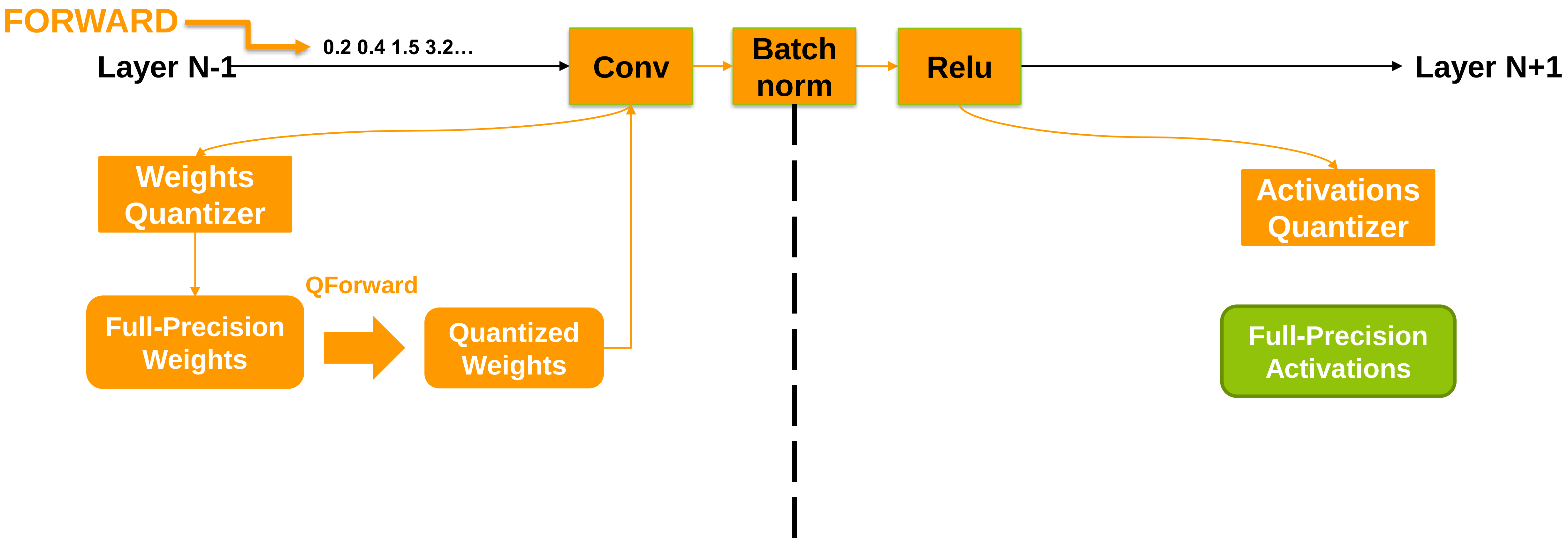
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



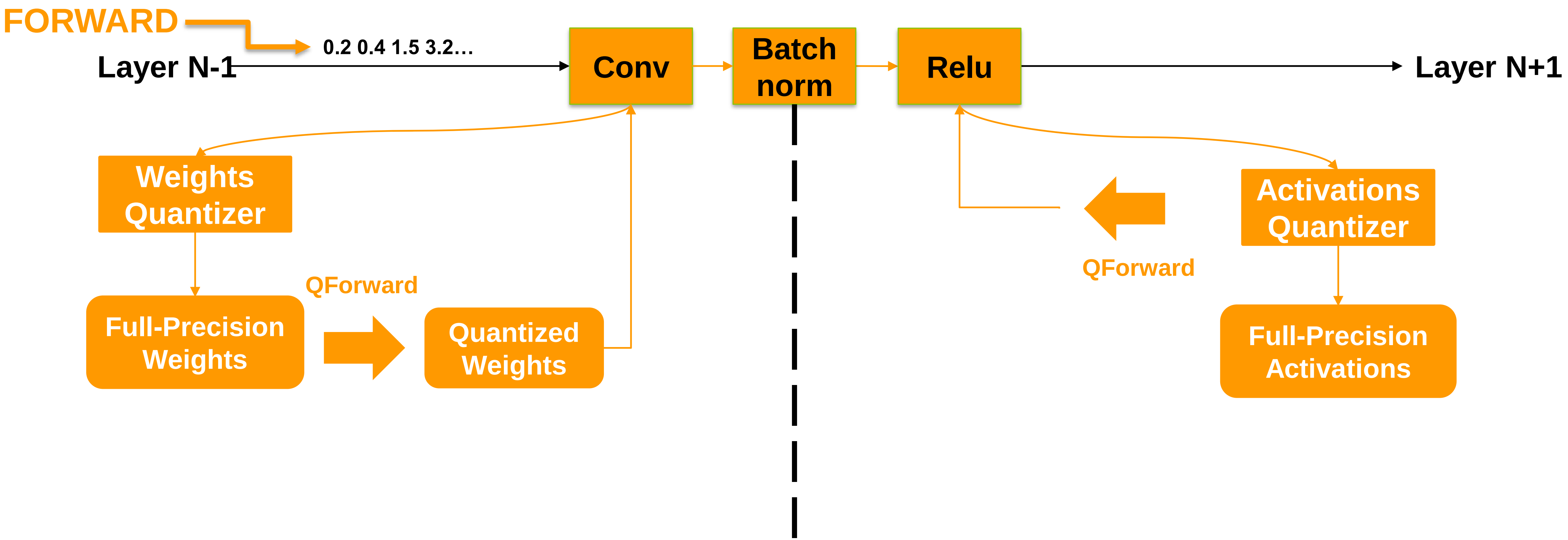
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



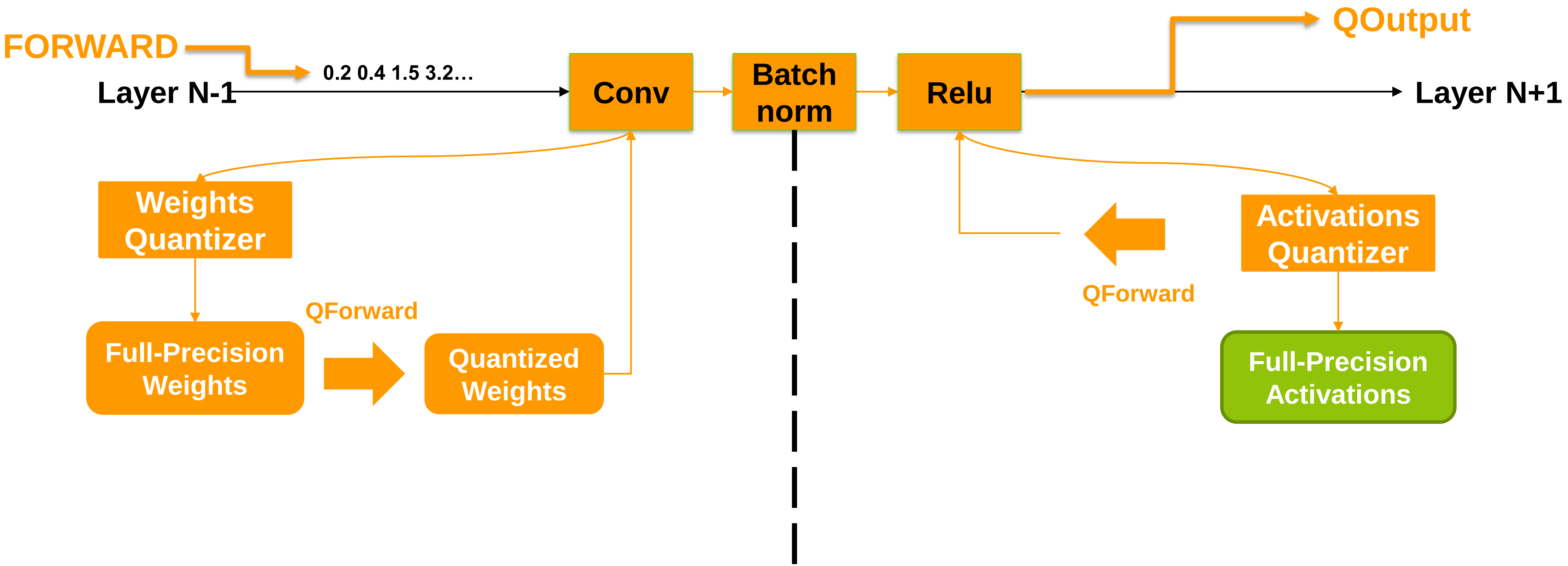
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



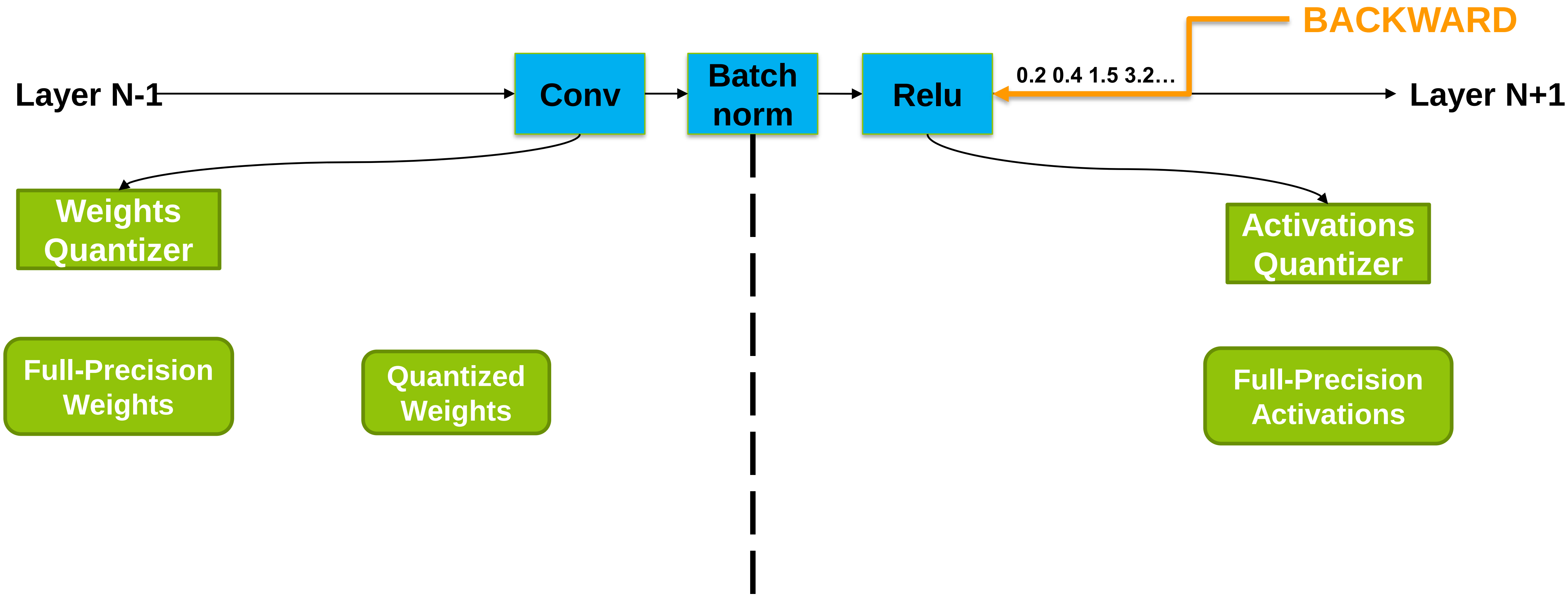
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



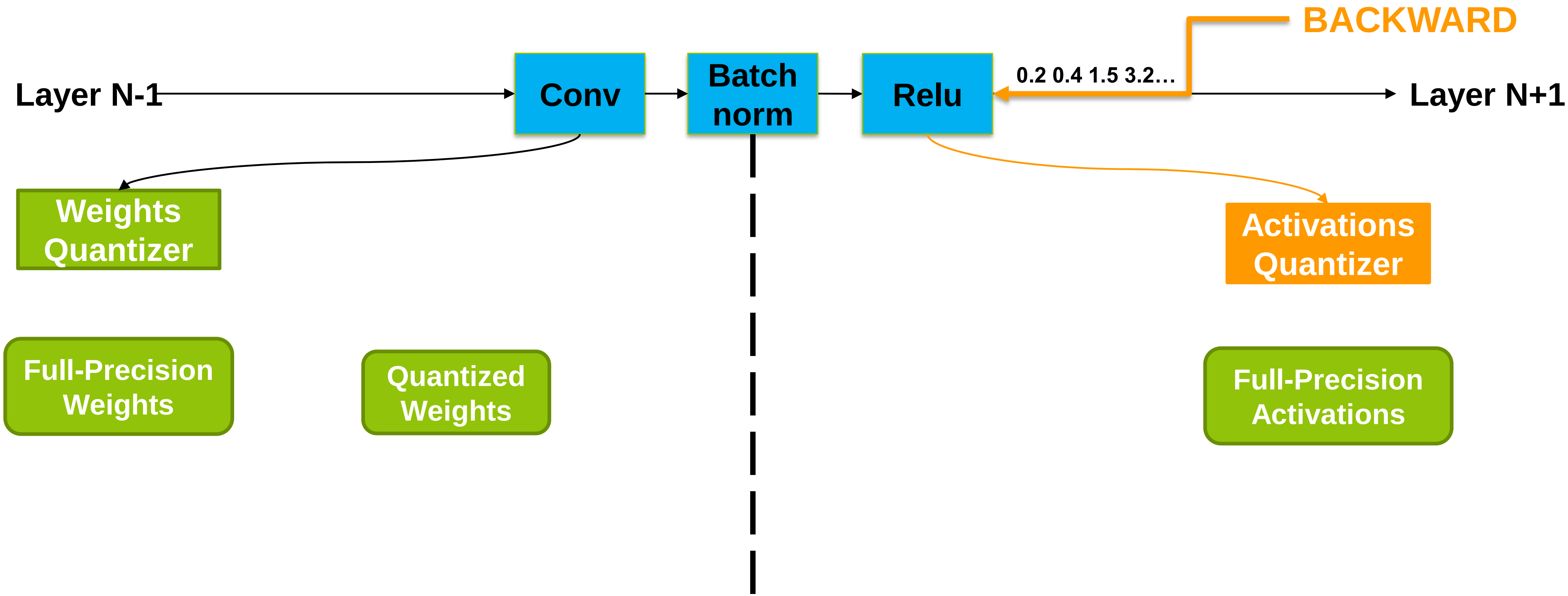
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



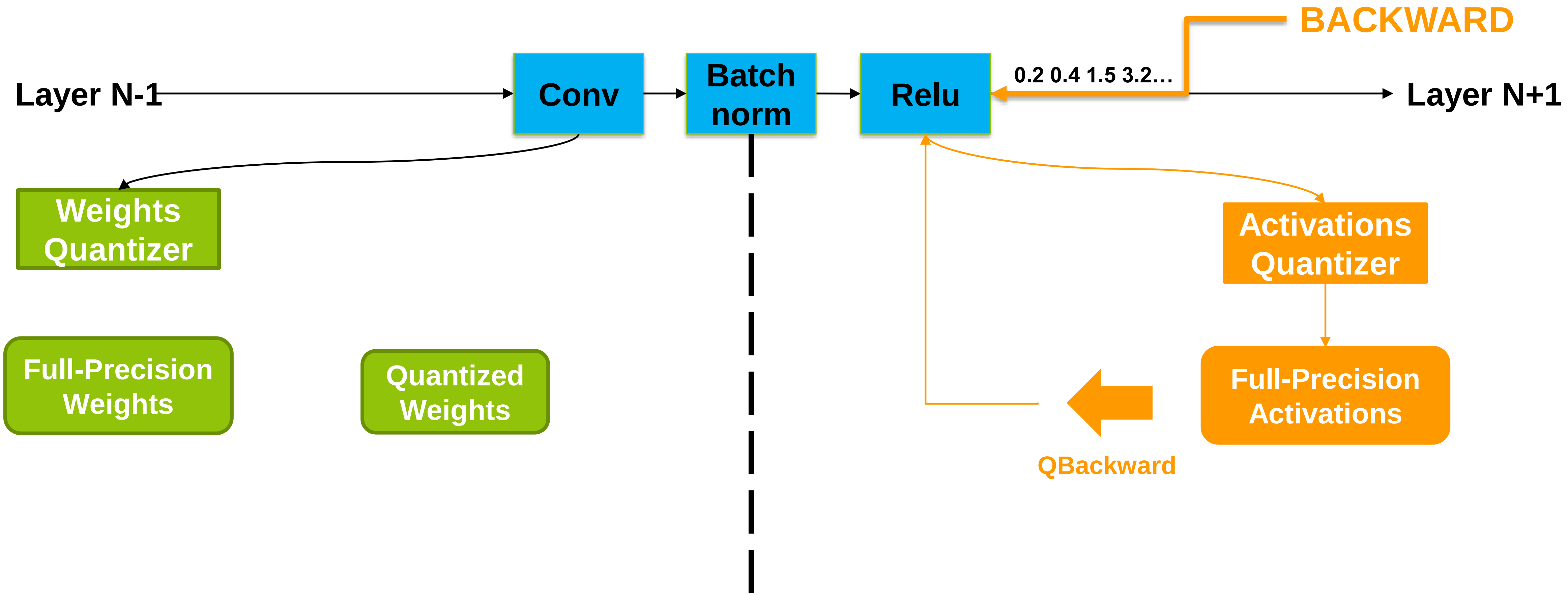
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



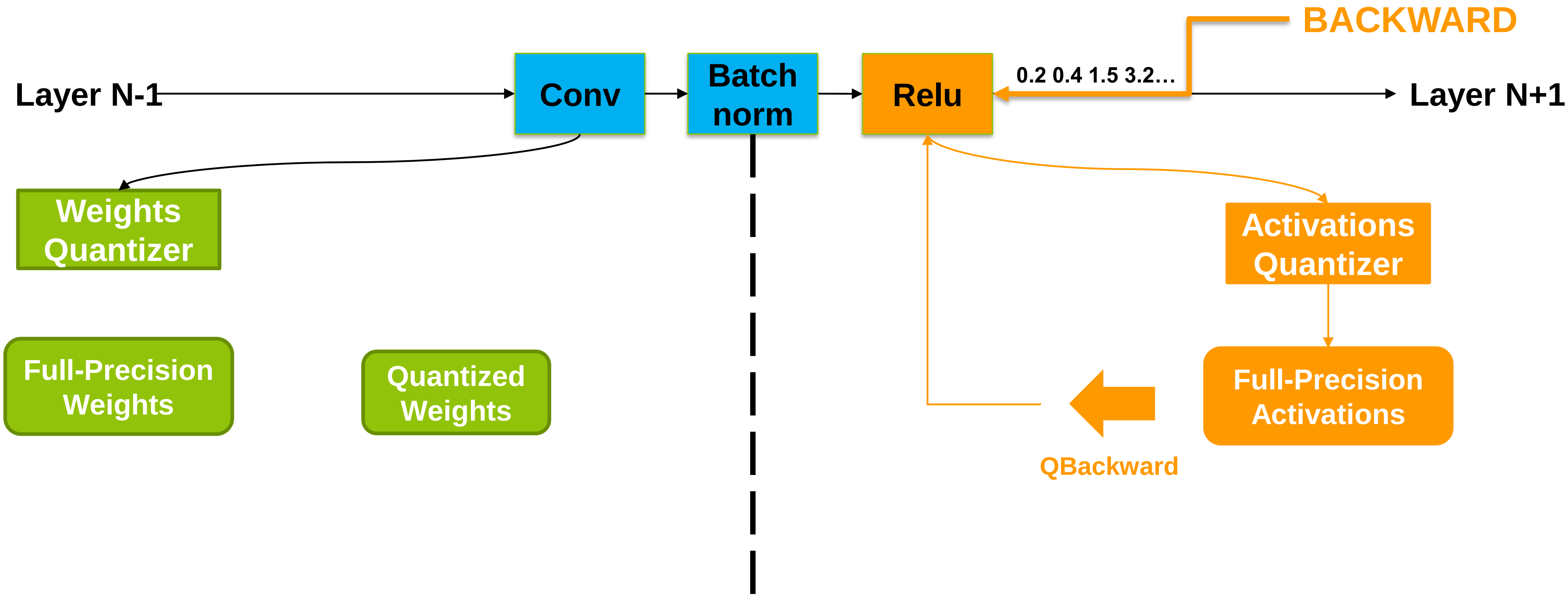
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



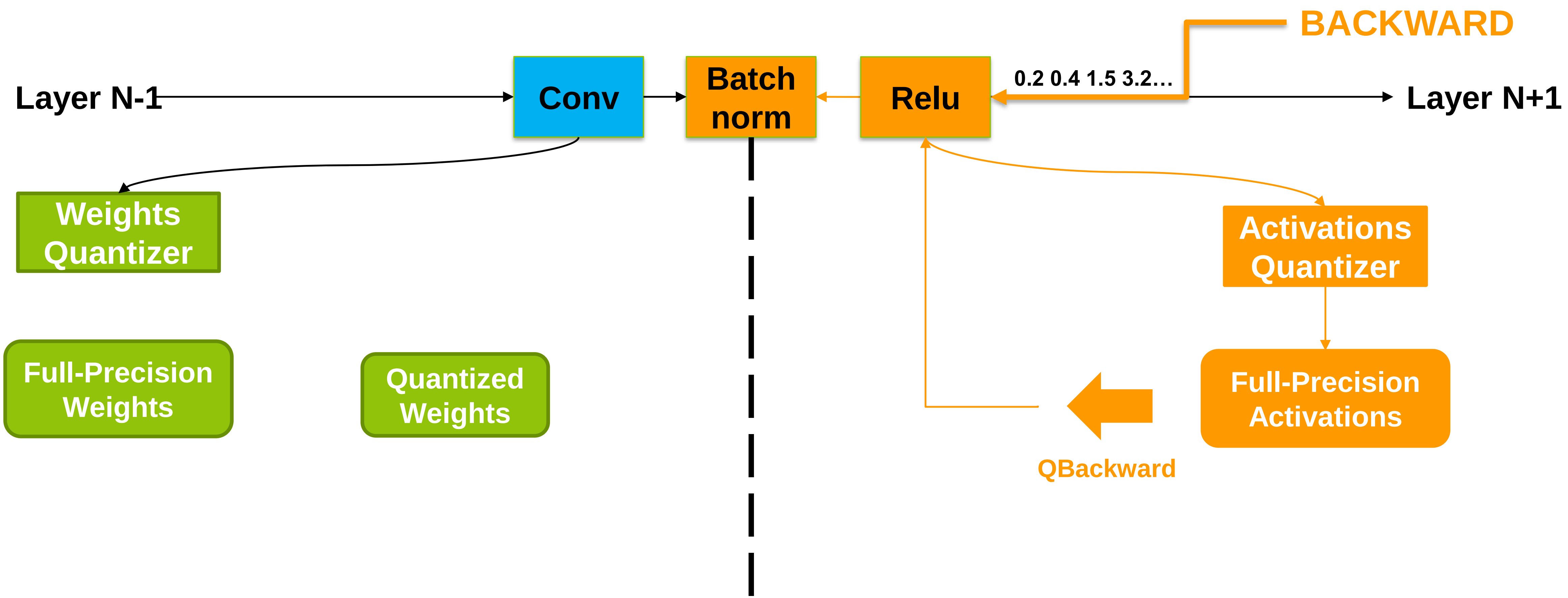
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



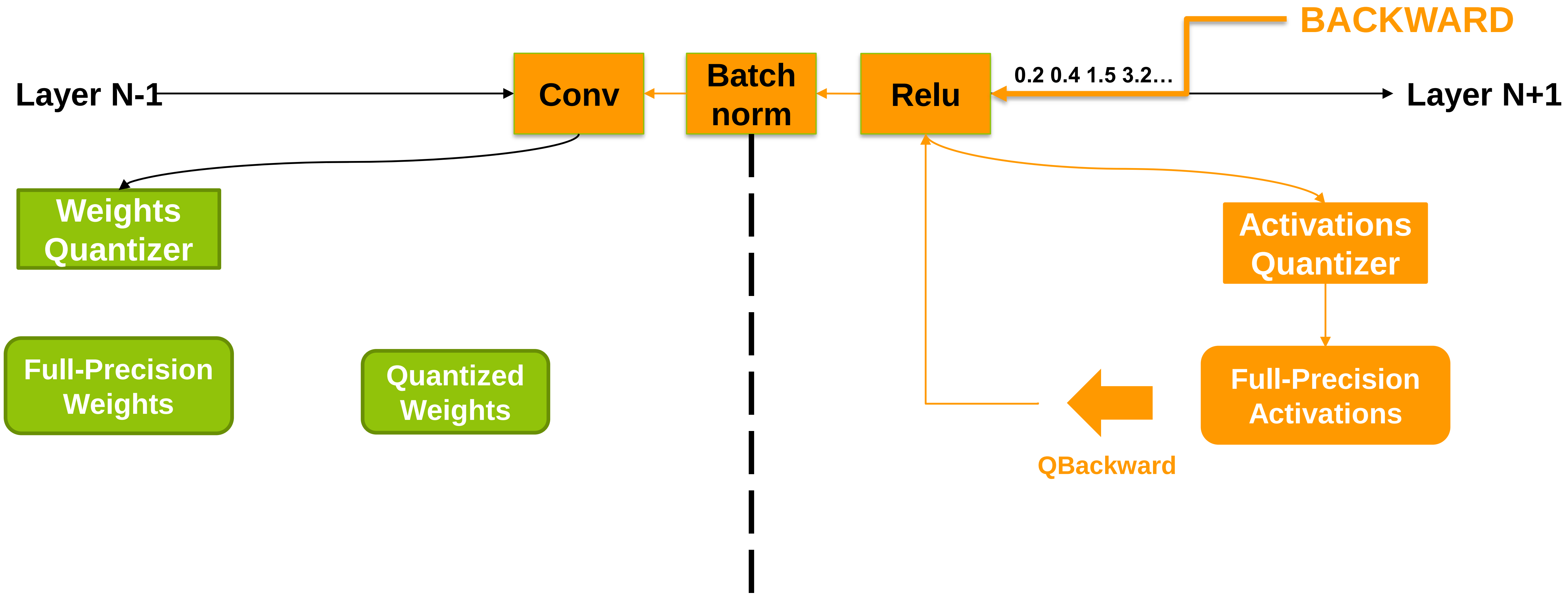
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



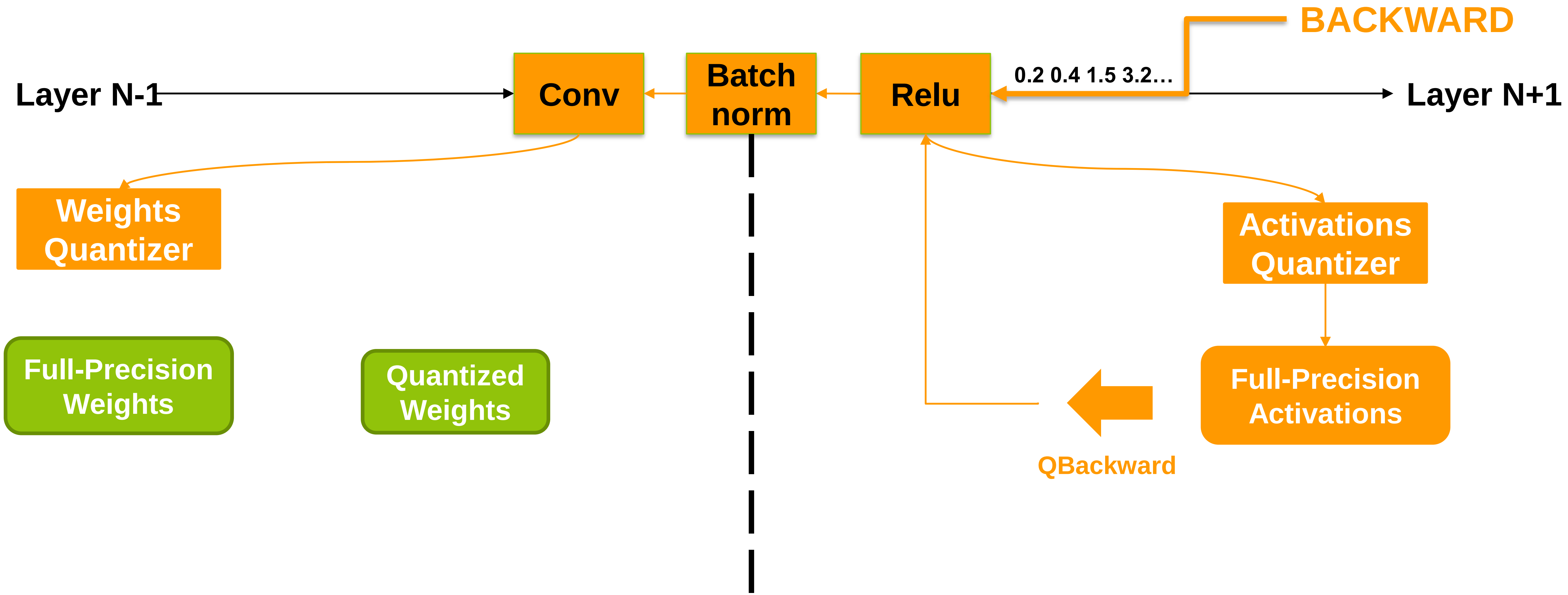
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



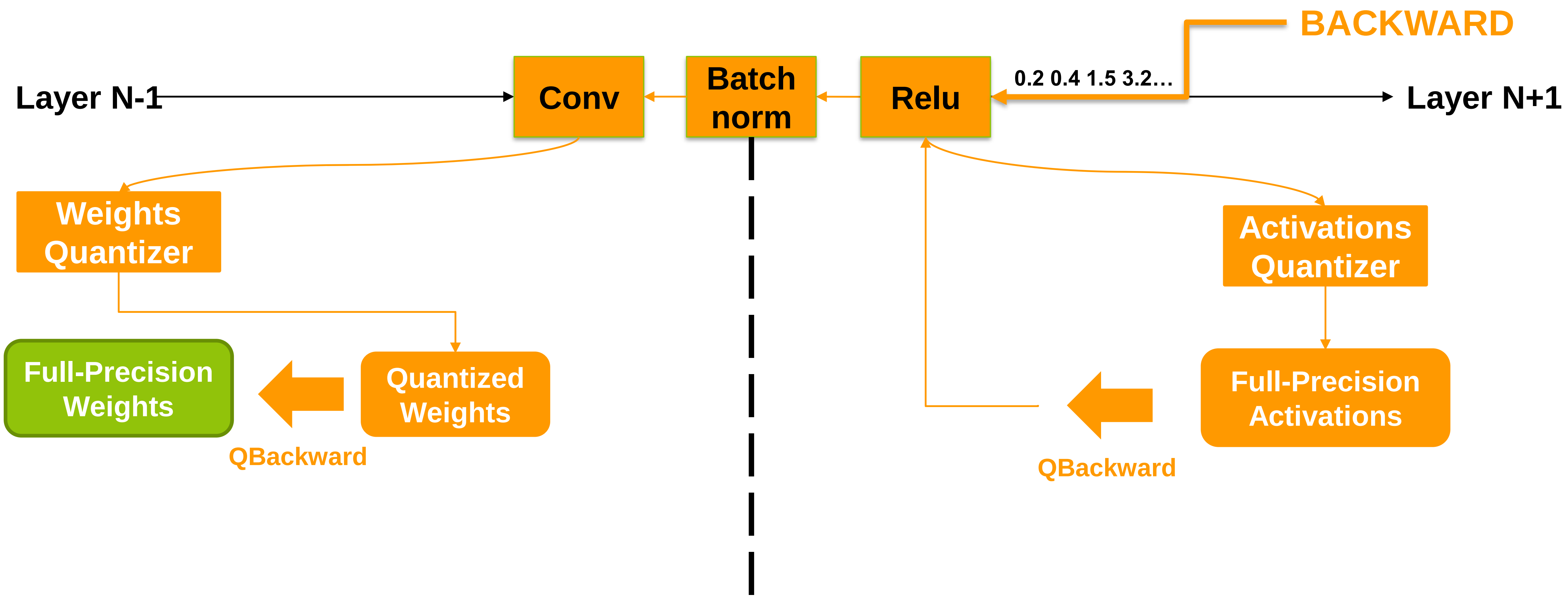
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



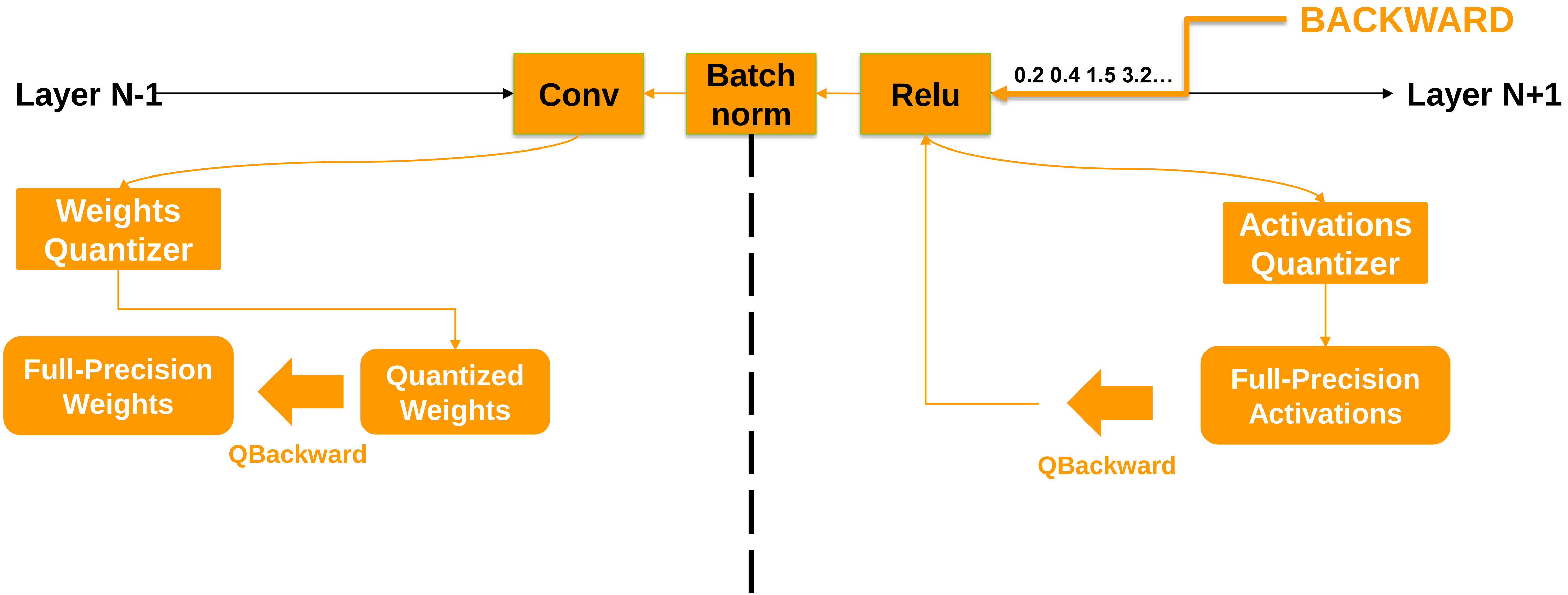
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



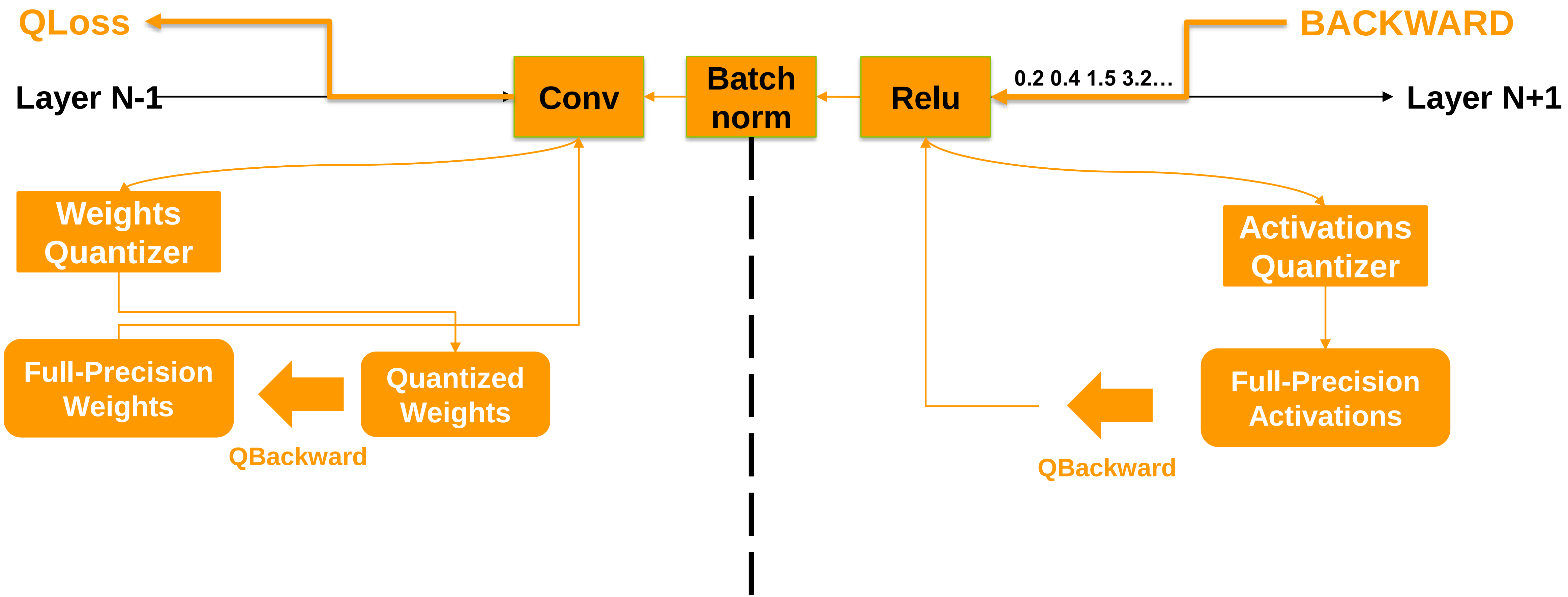
QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



QUANTIZATION AWARE-TRAINING (QAT)

- Principle: Training the model in a way that considers the quantization
- How does it work ?



QUANTIZATION AWARE-TRAINING (QAT)

- Avantages:
  - Training can be adapted to the precision (number of bits) required
  - 8-bit models often give better results than the full precision ones
  - 4-bit models now have the same accuracy than the full precision ones
  - Possibility to quantize lower than 4 bits with a small accuracy loss

- Drawback:
  - Need a retrain of the model with many epochs (90-150)
  - Quite slow to have a result

RESULTS AND VALIDATION

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**

Source: https://arxiv.org/pdf/1902.08153.pdf

RESULTS AND VALIDATION

Results obtained with the `SAT` method (~150 epochs) under the integer only mode :

MobileNet-v1 - `SAT` ImageNet Performances - Integer ONLY					
	Quantization Range (bits)				
Top-1 Precision	Weights	Activations	Parameters	Memory	Alpha
`72.60 %`	8	8	4 209 088	4.2 MB	1.0
`71.50 %`	4	8	4 209 088	2.6 MB	1.0
`65.00 %`	2	8	4 209 088	1.8 MB	1.0
`60.15 %`	1	8	4 209 088	1.4 MB	1.0
`70.90 %`	4	4	4 209 088	2.6 MB	1.0
`64.60 %`	3	3	4 209 088	2.2 MB	1.0
`57.00 %`	2	2	4 209 088	1.8 MB	1.0
`69.00 %`	8	8	3 156 816	2.6 MB	0.75
`69.00 %`	4	8	3 156 816	1.6 MB	0.75

Source: https://cea-list.github.io/N2D2-docs/quant/qat.html

WHAT’S NEXT ?

- Pruning and quantization

- Mixed quantization
  - some studies have shown that the last layers of a NN can have a very low precision while the first ones should remain in higher precision (8-bit)

- Aggressive quantization (binary networks)