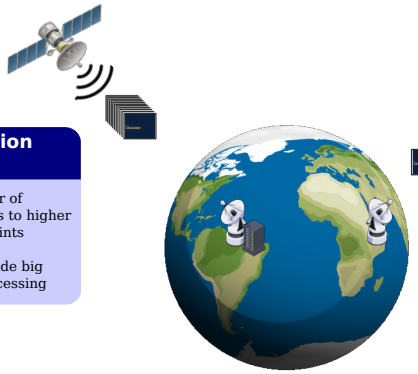
QUANTIZED NEURAL NETWORKS FOR ON-BOARD PROCESSING IN EARTH OBSERVATION BY SATELLITE

Cédric Gernigon[1], Silviu Filip[1], Olivier Sentieys[1], Clément Coggiola[2]

cedric.gernigon@inria.fr

[1] Univ Rennes, INRIA [2] CNES



Data transmission bottleneck

- The increase in the number of high-resolution sensors leads to higher downlink bandwidth constraints
- Deep learning (DL) has made big breakthroughs in image processing

Efficient data transmission

- On-board data processing reduces the amount of transmission
- Most DL algorithms are incompatible with embedded constraints

⟶ Need for efficient DL

Background

What is Quantization?
- Reduce the arithmetic bit-width (e.g., 32-bit floating point ⟶ 8-bit integer)

Why is it interesting?
- Lower power consumption
- Enables the use of FPGAs

Are there any disadvantages?
- Aggressive quantization leads to loss of accuracy

The impact of quantization on accurary

Post-Training Quantization (PTQ): quantize a pre-trained network ⟶ lower accuracy (-), data free (+), low computational cost (+)

Quantization aware training (QAT): train a network with quantized parameters ⟶ better accuracy (+), computationally expensive (-)

How to find a suitable bit-width?

Precision tuning during training

Learning the bit-width during training
- Find a uniform bit-width for both weights and activations

Hardware Loss:
$N_t \in \mathbb{R}_+^*$: bit-width, a new parameter to optimize

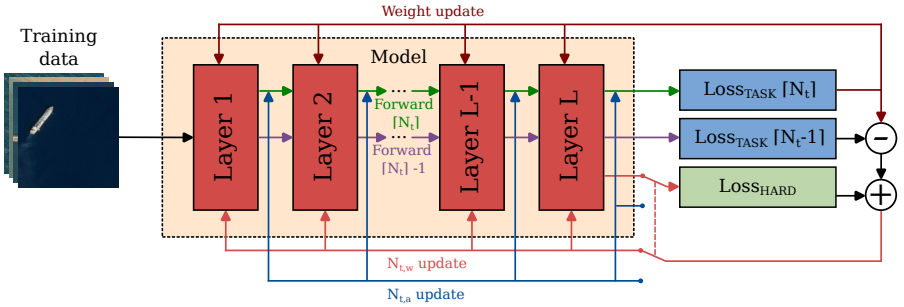
$$\mathrm{L_{Hard}} = \alpha \cdot \sum_{l=1}^{L-1} \left\lceil N_t^{(l)} \right\rceil$$

Total loss:

$$\mathrm{L_{Total}} = \lambda \cdot \mathrm{L_{Task}}\left\lceil N_t \right\rceil + (1-\lambda) \cdot \mathrm{L_{Hard}}\left\lceil N_t \right\rceil$$

Bit-width gradient approximation:

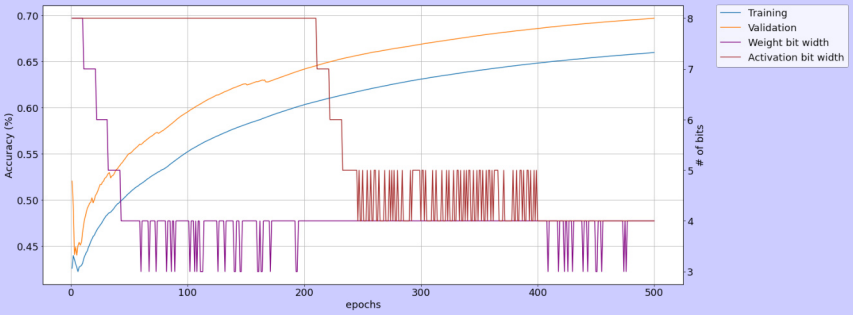
$$\lambda \cdot (\mathrm{L_{Task}}\left\lceil N_t \right\rceil - \mathrm{L_{Task}}\left\lceil N_t - 1 \right\rceil) + (1-\lambda) \cdot \frac{\partial \mathrm{L_{Hard}}\left\lceil N_t \right\rceil}{\partial \left\lceil N_t \right\rceil}$$



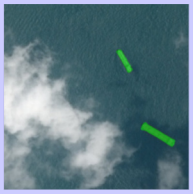
Training settings for image segmentation

Model: U-NET Dataset: Airbus Ship Kaggle Arithmetic: Integer

Hyper-parameters: $\lambda = 0.97$, $\alpha = 4$



Experimental Results

	Original image	Expected result
		
	FP32 model	**W4A4 model**
		
Accuracy	75.6%	70.1%
Size	288MB	36MB

Conclusion & future works

- Promising initial results for learning the bit-width of weights and activations

⟶ Extend the approach to mixed precision and other arithmetic
⟶ Model a more realistic hardware loss function
⟶ Develop an FPGA-based accelerator