

A decorative network diagram consisting of a series of interconnected nodes and lines, forming a complex web-like structure, is positioned in the upper left corner of the slide.

Operators & quantized op

**Adding bit parameter to QuantizeLinear and
DequantizeLinear operators for improved
hardware compatibility**

More and more hardware support low bit quantization

But Onnx only supports INT8 quantization for the QuantizeLinear and DequantizeLinear operators

Risc V - low
power ML

UNTETHER AI




Ternary to INT8


INT4 to INT8




Binary & INT8

- Datakalab -

Already an MR that discusses what we want to do

 Open



add dtype #4363
bzhang3 wants to merge 9 commits into `onnx:main` from `bzhang3:low-bit-proto` 

**daquexian** commented on Aug 9, 2022 • edited  Member  ...

The ops could be Conv, Pool or any other ops.

@bzhang3 @AlexandreEichenberger I think the support for low-bit quantization is of importance. However, can we update QDQ operators only and keep other operators unchanged? For example, after adding a `bit_width` attribute to QDQ, a 4-bit conv can be represented by

```
input (fp32) -> Q (bit_width=4) -> DQ (bit_width=4) ---> Conv -> output (fp32)
                                     /
weight (4 bit integer) -> DQ (bit_width=4) ---
```

  3

The Advantages



Simple backward compatibility as the default would be int8



Can help in making ONNX a better framework for the tiny edges (MCU, FPGA, RISC V) that have harder requirements



Not a massive MR, it won't reduce the size of the model but will help in the usability of the model



Will help in improving the experience of quantization on ONNX compare to TFLITE



Simple MR to add

- Datakalab -



MERCI!

Do you have any questions?

lf@datakalab.com

