Centered Symmetric Quantization for Hardware-Efficient Low-bit Neural Networks

Faaiz Asim*, Jaewoo Park*, Azat Azamat, Jongeun Lee

Introduction

Problem:
- Conventional Linear Quantization (CLQ) results in unequal number of positive and negative quantization levels (e.g., 2-bit signed CLQ has {−2, −1, 0, 1} quantization levels)
- A perfectly symmetric linear quantizer cannot be realized using standard multipliers without large overhead

Contributions:
- We propose Centered Symmetric Quantization (CSQ)
- We propose efficient methods to realize CSQ on hardware:
  1. Using standard multipliers with small overhead
  2. Using Binarized Neural Network (BNN) hardware with no overhead

CSQ Definition

We define Centered Symmetric Quantization (CSQ) as a quantizer that has:
1. Uniform step-size between quantization levels
2. Perfectly symmetric quantization levels

CSQ is formulated as:

\[ z_{CSQ} = \frac{z}{2^{n-1}} \]

\[ z'_{CSQ} = \text{clip} \left( z_{CSQ}, 0, Q \right) \]

\[ z'_{CLQ} = z_{CLQ} - 0.5 \]

CSQ Hardware Realization

Method 1: On standard multipliers using affine quantization

Zero-point for CSQ and CLQ in context of affine quantization:

\[ x_{CSQ} = 2^{n-1}, \quad x_{CLQ} = 2^{n-1} - 0.5 \]

\[ \Delta x_{CSQ} = \Delta x_{CLQ} - 0.5 \]

Realizing CSQ as affine quantizer:

\[ z_{CSQ} = \left( \left( z - 0.5 \right) \times x_{CSQ} \right) / x_{CSQ} \]

\[ = 6 \Delta x_{CSQ} - 0.5 \times 1_{CSQ} \]

- Significantly smaller overhead compared to affine quantizer
- Flexible method generalizable to any hardware
- Results in small overhead

Method 2: Binarized BNN hardware-based method

An n-bit CSQ number can be represented in binary using +1 and −1, instead of 0 and 1. (See 2-bit CSQ binary encoding example)

For signed weights and unsigned activations, CSQ-CLQ multiplication can be computed using AND-popcount:

\[ v_{CSQ} 	imes x_{CLQ} = 2 \cdot \text{popcount} \left( \text{AND} \left( v_{CSQ}, x_{CLQ} \right) \right) - \text{popcount} \left( x_{CLQ} \right) \]

- No overhead
- Configurable precision due to bit-wise operations
- Cannot be realized using standard multipliers

Analyzing CSQ vs CLQ

Learned per-layer zero-point distribution for ResNet-20 compared to CSQ vs. CLQ
- Zero-point distribution is closer to CSQ than CLQ especially at 2-bit precision
- Zero-point distribution tends to move from CSQ towards CLQ as the precision increases

ImageNet Results

FPGA Results

Conclusion

- CSQ for weight quantization can provide significant performance improvement compared to CLQ at extremely low precision (≤3-bits)
- At higher precisions (≥4-bits) the performance improvement using CSQ diminishes.
- For standard multipliers (e.g., GPU, CPU), our affine quantization-based hardware realization method allows CSQ realization with a very small overhead.
- Our bitwise BNN hardware-based method allows CSQ realization without any overhead.