Hybrid FP8 ONNX

FP8 MatMul and Elementwise Operation

Acts/Wts cross data type support (ZM)			
Activations	Weights	Output Size	Input Z multiple
I8	I8	I8	16
U8	U8	U8	16
FP16	FP16	FP16	16
FP16	I8	FP16	16
FP16	BF8	FP16	16
FP16	HF8	FP16	16
BF8	FP16	FP16	16
HF8	FP16	FP16	16
BF16	BF16	BF16	16
BF16	BF8	BF16	16
BF16	HF8	BF16	16
BF8	BF16	BF16	16
HF8	BF16	BF16	16
BF8	BF8	BF8	16
BF8	HF8	BF8/HF8	16
HF8	HF8	HF8	16
HF8	BF8	HF8/BF8	16

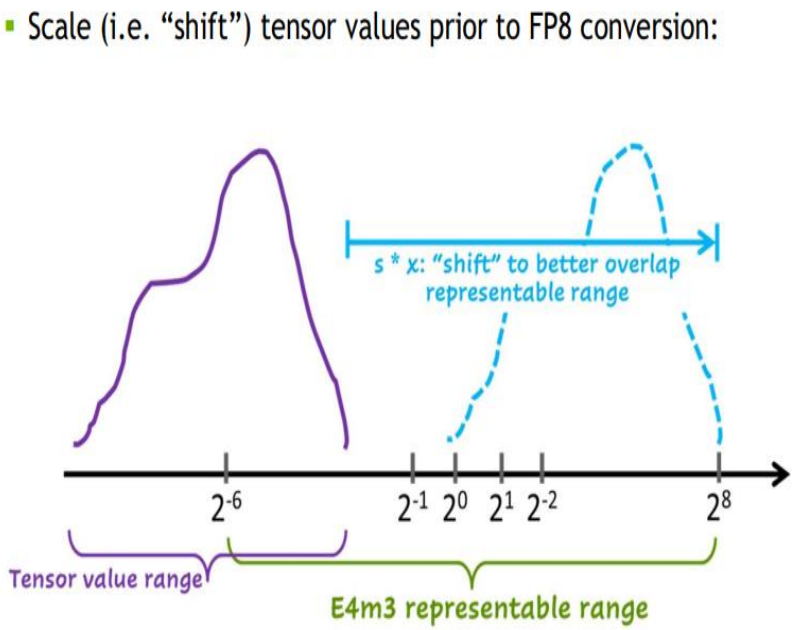
* HF8 –> E4M3 and BF8 –> E5M2

Acts/Wts data type support (Elt-wise)			
Activations	Weights	Output Size	Input Z multiple
I8	I8	I8	16
U8	U8	U8	16
BF8	BF8	FP16/BF16	16
BF8	HF8	FP16/BF16	16
HF8	HF8	FP16/BF16	16
HF8	BF8	FP16/BF16	16
FP16	BF8	FP16	16
FP16	HF8	FP16	16
BF8	FP16	FP16	16
HF8	FP16	FP16	16
FP16	FP16	FP16	16
BF16	BF8	BF16	16
BF16	HF8	BF16	16
BF8	BF16	BF16	16
HF8	BF16	BF16	16
BF16	BF16	BF16	16

For all floating point matmul operations (including hybrid floating point formats), the accumulation is done at FP32 precision.

FP8 Scale factor

- HF8(E4M3) has a more limited dynamic range that BF8 (E5M3) and was found to be less effective on large NLP models [FP8_Whitepaper_v3.1]. However, the range issue was addressed by applying per layer scaling to the weight and activation tensors using methodologies that are already applied for INT8 quantization.
- In the “FP8 Formats for Deep Learning” paper from Intel, ARM and Nvidia, it states that “leaving per-tensor scaling to software implementation enables more flexibility” and that the “scaling factor can take on any real value (typically represented in higher precision)”.



FP8 Scale Factor (contd.)

- For operations such as the convolution or the elementwise multiplication of two tensors**, the scaling factors can be combined and applied as part of the post processing engine after the value is computed.**.
- For operations such as the elementwise addition or the elementwise subtraction of two tensors**, the per tensor scale factors must be applied to each tensor prior to the MatMul performing the compute operation.**
- During an FP8 elementwise addition or elementwise subtraction operation, the accelerator **converts the FP8 tensor data to either FP16 or BF16, and supply both the 16-bit scaling factor and the 16-bit converted FP8 value to the MAC.**

Rules for Eltwise Add/Sub

Rules for Elementwise Addition/Subtraction of Two FP8 Tensors

- The tensors can both be the same format (BF8/BF8 or HF8/HF8) or one tensor can be HF8 while the other tensor is BF8.
- The scale factor for both tensors must be the same format – either BF16 or FP16.

Rules for Elementwise Addition/Subtraction of an FP8 Tensor and a BF16/FP16 Tensor

- One tensor can be an 8-bit FP format (BF8/HF8) and one tensor can be a 16-bit FP format (BF16/FP16).
- The scale factor for the 8-bit FP tensor must be in the same format, either BF16 or FP16, as the format of the 16-bit FP tensor.
- No scale factor is configured for the 16-bit FP tensor.

ONNX Suggestion: To avoid complexity, dequantize the activations to the correct FP16/BF16 format before eltwise ops.

Backup