VS-Quant: Per-vector Scaled Quantization for Accurate Low-Precision Neural Network Inference

Steve Dai, Rangharajan Venkatesan, Haoxing Ren, Brian Zimmer, William J. Dally, Brucek Khailany
NVIDIA Research

Overview

Motivations

- DNN models can be deployed for inference in lower precisions to maximize hardware performance and efficiency.
- Quantized inference accelerates compute-bound operations, conserve memory bandwidth for memory-bound operations, and reduce on-chip storage size.

Challenge

- Conventional per-channel scaled quantization results in severe accuracy loss, especially at low bitwidths and without quantization-aware training.
- The number of efficient design points with acceptable accuracy is limited.

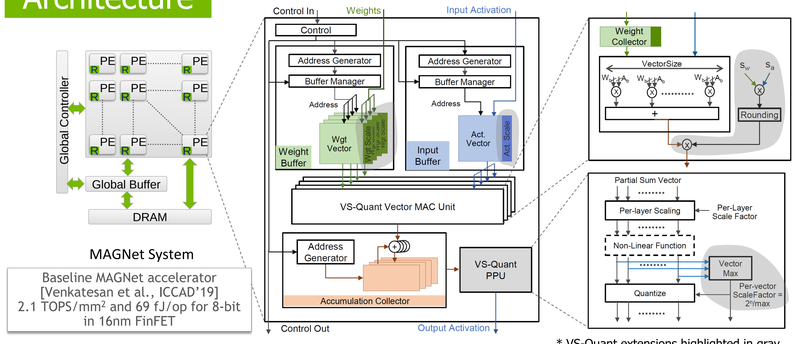
Opportunities

- To improve accuracy, we can leverage fine-grained scaling to mitigate accuracy loss typical in existing quantized models.
- To maximize hardware efficiency, we can co-design the quantization algorithm around the vector MAC unit ubiquitous in DNN hardware.

Our Proposal: VS-Quant

- Per-vector scaled quantization technique to mitigate accuracy loss.
- Two-level scaling algorithm to realize efficient per-vector scaled hardware.
- Explored tradeoffs between accuracy and hardware efficiency on a range of hardware implementations and DNN models.
- Achieves higher accuracy and/or hardware efficiency while enabling a rich design space

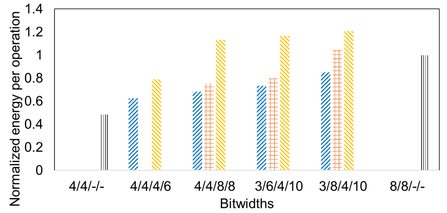
Architecture



MAGNet System

Baseline MAGNet accelerator [Venkatesan et al., ICCAD'19] 2.1 TOPS/mm² and 69 fJ/op for 8-bit in 16nm FinFET

* VS-Quant extensions highlighted in gray.



8-bit scale product
Full bitwidth scale product
12-bit scale product
Per-channel scaled

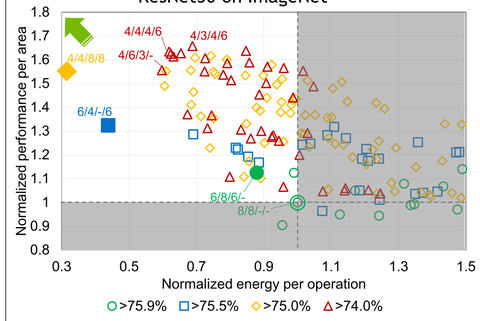
- With VS-Quant, we see energy overhead over corresponding per-channel scaled configuration due to additional multipliers and wider accumulation.
- When the scale factors product is rounded, the energy overheads of adding VS-Quant support can be substantially reduced.

Weight Width / Activation Width / Weight Scale Width / Activation Scale Width
"-" indicates per-channel scaling

Design Space Exploration

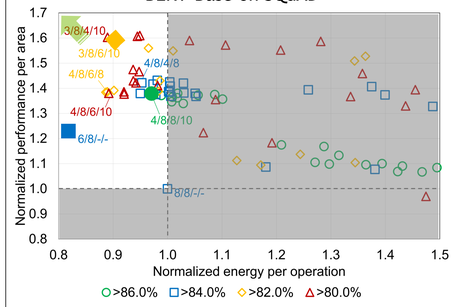
Weight Width / Activation Width / Weight Scale Width / Activation Scale Width
"-" indicates per-channel scaling
Amount of scale rounding varies among different design points.

ResNet50 on ImageNet

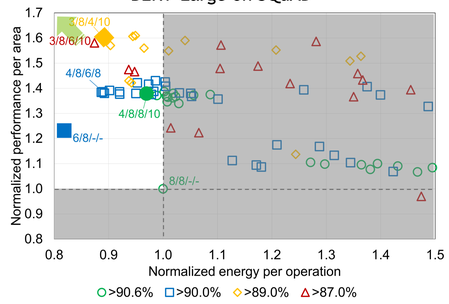


Design tools	
HLS Compiler	Mentor Graphics Catapult HLS
Verilog simulator	Synopsys VCS
Logic synthesis	Synopsys Design Compiler
Place-and-route	Synopsys ICC2
Power analysis	Synopsys PT-PX
Design space	
Vector size	16
Weight/activation precision	3-bit, 4-bit, 6-bit, 8-bit
Weight/activation scale precision	3-bit, 4-bit, 6-bit, 8-bit, 10-bit
Scaling granularity	Per-channel, per-vector

BERT-Base on SQuAD



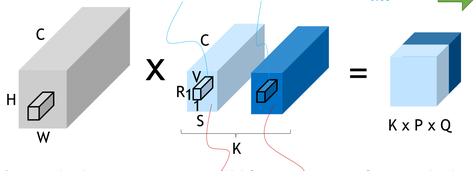
BERT-Large on SQuAD



- 4/4/8/8: 69% energy saving, 36% area saving, >75.0% accuracy for ResNet50 on ImageNet.
- 4/8/8/10: 28% area saving, 3% energy saving, near full-precision accuracy for BERT on SQuAD.
- Overall better accuracy, energy, and area tradeoffs for low-precision inference.

Per-vector Scaled Quantization

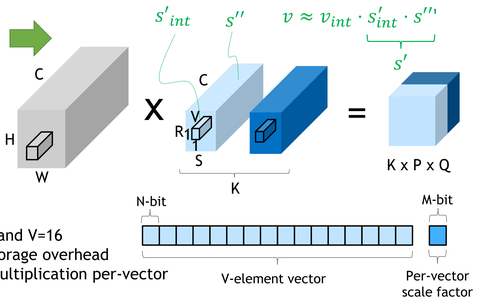
Per-vector scaling $s'_{0,r,s,0} \ldots s'_{K-1,r,s,0}$ $v \approx v_{int} \cdot s'$



Per-channel scaling $s_0^{wt} \ldots s_{K-1}^{wt}$ $v \approx v_{int} \cdot s$

Two-level quantization

1. $v \approx v_{int} \cdot s'$
2. $s' \approx s'_{int} \cdot s''$
3. $v \approx v_{int} \cdot s'_{int} \cdot s''$



Assume N=M=4 and V=16
1/16 = 6.25% storage overhead
12-bit x 8-bit multiplication per-vector

Post-training quantization accuracy for various per-vector scaled configurations
(Rows represent weight/activation bitwidths. Columns represent weight/activation scale bitwidths.)

Bitwidths	S=3/4	S=3/6	S=4/4	S=4/6	S=6/4	S=6/6	S=fp32	Per-channel
Wt=4 Act=4U	73.4	74.2	74.4	75.0	74.6	75.4	75.3	70.8
Wt=6 Act=4U	74.3	75.1	75.0	75.6	75.1	75.8	75.8	74.8
Wt=6 Act=6U	74.7	75.1	75.1	75.7	75.4	76.0	76.0	75.8
Wt=8 Act=8U	74.6	75.3	75.2	75.9	75.5	76.1	76.2	76.2
						FP Top1 Accuracy		76.2%

ResNet50 on ImageNet

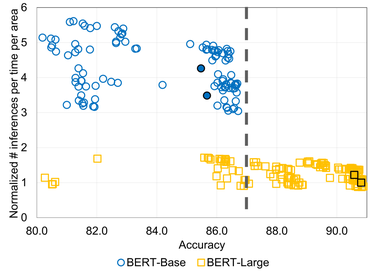
Bitwidths	S=4/8	S=4/10	S=6/8	S=6/10	S=fp16	S=fp32	Per-Channel
Wt=3 Act=8	81.6	81.8	82.6	82.8	82.9	82.9	12.1
Wt=4 Act=8	85.7	85.9	86.0	86.3	86.3	86.3	78.9
Wt=6 Act=8	85.9	86.2	86.3	86.7	86.6	86.6	85.5
Wt=8 Act=8	85.9	86.4	86.4	86.6	86.6	86.5	85.7
					FP Top1 Accuracy		86.9%

BERT-base on SQuAD

Bitwidths	S=4/8	S=4/10	S=6/8	S=6/10	S=fp16	S=fp32	Per-Channel
Wt=3 Act=8	88.7	89.0	89.4	89.6	89.6	89.5	11.6
Wt=4 Act=8	90.3	90.5	90.6	90.7	90.6	90.7	86.8
Wt=6 Act=8	90.5	90.5	90.7	90.8	90.8	90.8	90.6
Wt=8 Act=8	90.6	90.6	90.6	90.8	90.9	90.9	90.8
					FP Top1 Accuracy		90.9%

BERT-large on SQuAD

Model Size vs. Efficiency



- Use BERT-large if accuracy target is higher than best BERT-base accuracy
- Otherwise go with BERT-base for consistently better efficiency
- Configure size of the model based on the desired accuracy target to realize best hardware efficiency

Quantization-aware Training

- VS-Quant is not limited to post-training quantization.
- VS-Quant models can be finetuned with quantization-aware training to get even better accuracy or achieve lower precision.

Model	Bitwidths	VS-Quant (Accuracy @ epoch)	Per-channel (Accuracy @ epoch)
ResNet50	Wt=3, Act=3U	75.5% @20	72.0% @20
BERT-base	Wt=4, Act=4	86.2% @5	73.3% @20
BERT-large	Wt=3, Act=4	89.2% @2	21.6% @2