ShiftAddLLM: Accelerating Pretrained LLMs via Post-training Multiplication-less Reparameterization

Haoran You[1], Yipin Guo[1], Yichao Fu[1], Wei Zhou[1], Huihong Shi[1], Xiaofan Zhang[3], Souvik Kundu[2], Amir Yazdanbakhsh[4], Yingyan (Celine) Lin[1]

[1]Georgia Institute of Technology [2]Intel Labs [3]Google [4]Google DeepMind

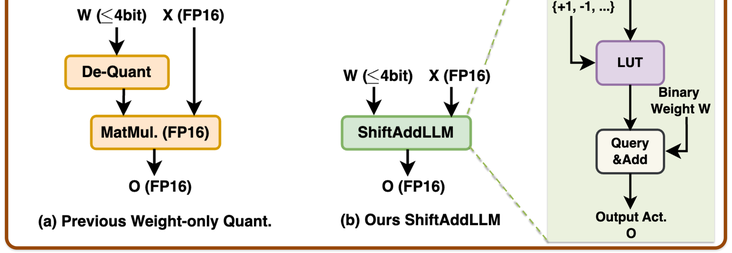
Background and Motivation

- **Powerful LLMs suffer from large deployment cost**
  - Extensive parameters and memory usages:
    - E.g., GPT-3 has 175B parameters and needs 350GB GPU memory
  - Dense multiplications:
    - E.g., GPT-3 performs 10^{15} FLOPs for a single forward pass
- **Previous solutions for efficient LLM deployment**
  - Solution 1: Post-training quantization:
    - Directly quantize the pretrained LLMs without fine-tuning
    - Still have many multiplications and need de-quantization
  - Solution 2: Shifts and adds reparameterization:
    - Up to 31× unit energy reduction and 26× unit area reduction
    - Previous methods need full parameter training to recover accuracy

*Can we reparameterize the pretrained LLMs with shifts and adds in a **"post-training"** manner?*

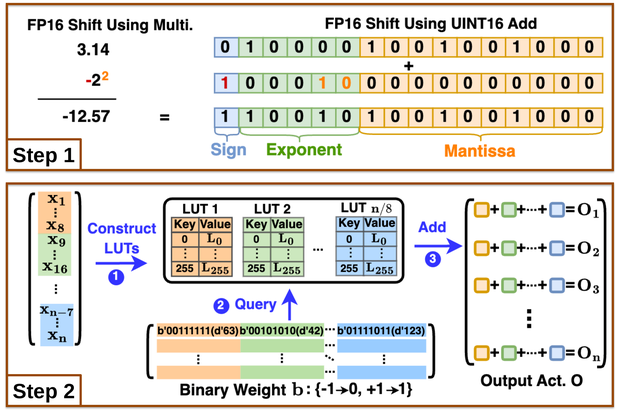
ShiftAddLLM: Reparameterization

- **ShiftAddLLM**
  - Do not need fine-tuning and de-quantization
  - Reparameterize multiplications with shifts and adds



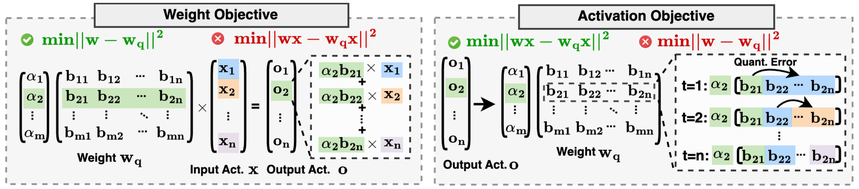
Reparameterization:

- **Step 1:** Quantize the scaling factor matrices to powers of 2 → perform FP16 shift for the input activations
- **Step 2:** Leverage LUTs to accelerate the multiplication between the shifted activations and the binary weight matrices

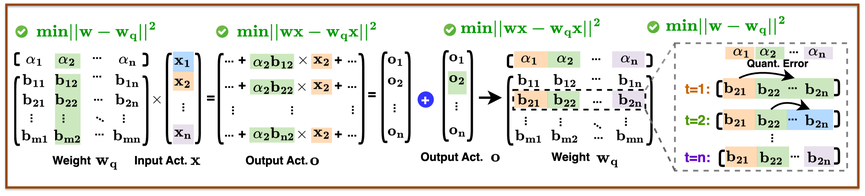


ShiftAddLLM: Multi-objective Optimization

- **Problem:** Accuracy drops after reparameterization
  - **One reason:** Mismatch between weight objectives and activation objectives

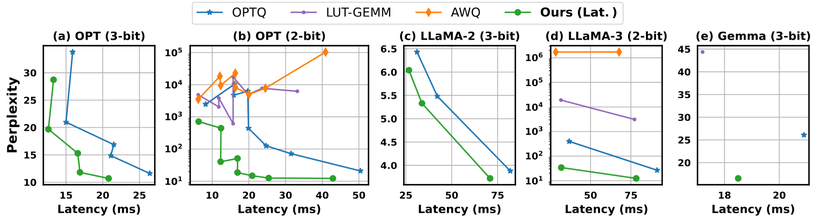


- **Solution:** Multi-objective optimization with column-wise scaling factors



Evaluation Results

- **Evaluation setup**
  - Tasks: One language modeling task (WikiText-2) and eight downstream tasks
  - Datasets: WikiText-2, ARC, BoolQ, Copa, PIQA, RTE, StoryCloze, and MMLU
  - Models: OPT, LLaMA-1/2/3, Gemma, Mistral, and Bloom
- **Benchmark baselines**
  - OPTQ [ICLR'22], LUT-GEMM [ICLR'24], QuIP [NeurIPS'23], AWQ [MLSys'24]
- **ShiftAddLLM over SOTA baselines**
  - Average perplexity reductions of 5.6 and 22.7 at comparable or even lower latency compared to the most competitive quantized LLMs at 3 and 2 bits
  - More than 80% memory and energy reductions over the original LLMs



Models	Methods	Bits	ARC_C	ARC_E	Copa	BoolQ	PIQA	Storycloze	RTE	MMLU	Mean
OPT-66B	Floating Point	16	37.20	71.25	86	69.82	78.67	77.47	60.65	25.89±0.37	63.37
	OPTQ	3	24.66	48.86	70	52.05	64.47	67.09	53.07	23.98±0.36	50.52
	LUT-GEMM	3	24.15	51.85	81	53.52	61.97	60.60	48.74	23.73±0.36	50.70
	Ours (Acc.)	3	**35.24**	**70.88**	**87**	**72.45**	**77.64**	**77.15**	**63.18**	**27.56±0.38**	**63.89**
LLaMA-2-70B	Floating Point	16	49.57	76.14	90	82.57	80.79	78.61	68.23	65.24±0.37	72.89
	OPTQ	3	45.82	76.34	90	81.74	79.71	77.34	67.51	60.14±0.36	72.33
	LUT-GEMM	3	47.70	76.42	89	80.31	80.20	77.78	68.59	-	-
	Ours (Acc.)	3	**48.38**	**77.06**	**93**	**84.25**	**80.47**	**78.49**	**75.09**	**62.33±0.38**	**74.88**

Acknowledge:
NSF RTML program and the CoCoSys, one of the seven centers in JUMP 2.0 sponsored by DARPA

Zoom for Online Q & A

SCAN ME