Memory-Efficient Fine-Tuning of Compressed Large Language Models via sub-4-bit Integer Quantization

Jeonghoon Kim[1,*], Jung Hyun Lee[1,*], Sungdong Kim[1,4], Joonsuk Park[1,2],

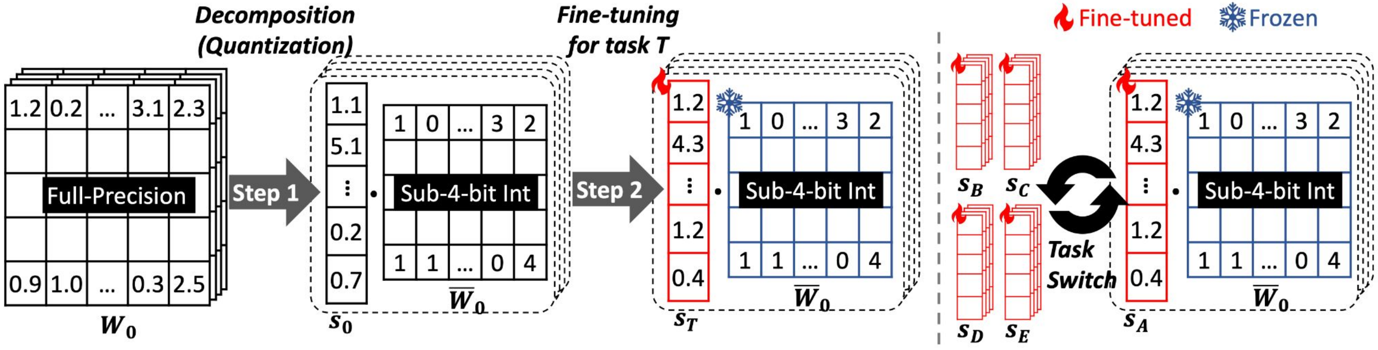
Kang Min Yoo[1,3], Se Jung Kwon[1], Dongsoo Lee[1]

[1]NAVER Cloud, [2]University of Richmond, [3]SNU AI Center, [4]KAIST AI

37th Conference on Neural Information Processing Systems (NeurIPS 2023)

NAVER Cloud

Method



Parameter-efficient Fine-tuning

Deployment

For pre-trained weights of a fully-connected layer $\boldsymbol{W}_0 \in \mathbb{R}^{n\times m}$,

$$\widehat{\boldsymbol{W}}_0 = \boldsymbol{s}_0 \cdot \overline{\boldsymbol{W}}_0 = \boldsymbol{s}_0 \cdot \Big(\mathrm{clamp}\Big(\Big\lfloor \frac{\boldsymbol{W}_0}{\boldsymbol{s}_0} \Big\rceil + \boldsymbol{z}_0, 0, 2^b - 1\Big) - \boldsymbol{z}_0\Big),$$

where $A \cdot B$, $\lfloor \cdot \rceil$, and $\mathrm{clamp}(\cdot, a, b)$ indicate the element-wise product of A and B, the rounding function, and the clamping function into the range $|a, b|$, respectively, while per-channel scales and zero-points (namely, $\boldsymbol{s}_0, \boldsymbol{z}_0 \in \mathbb{R}^{n\times 1}$) are initialized to minimize $\|\boldsymbol{W}_0 - \widehat{\boldsymbol{W}}_0\|_F^2$.

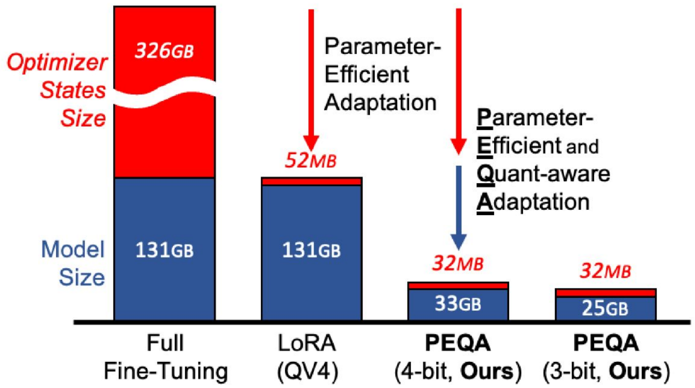
$$\widehat{\boldsymbol{W}} = (\boldsymbol{s}_0 + \Delta\boldsymbol{s}) \cdot \overline{\boldsymbol{W}}_0 = (\boldsymbol{s}_0 + \Delta\boldsymbol{s}) \cdot \Big(\mathrm{clamp}\Big(\Big\lfloor \frac{\boldsymbol{W}_0}{\boldsymbol{s}_0} \Big\rceil + \boldsymbol{z}_0, 0, 2^b - 1\Big) - \boldsymbol{z}_0\Big),$$

where $\Delta\boldsymbol{s} \in \mathbb{R}^{n\times 1}$ represents the gradient update of $\boldsymbol{s}_0$ obtained by adaptation to a downstream task. We dub Eq. 2 as Parameter-Efficient and Quantization-aware Adaptation (PEQA).

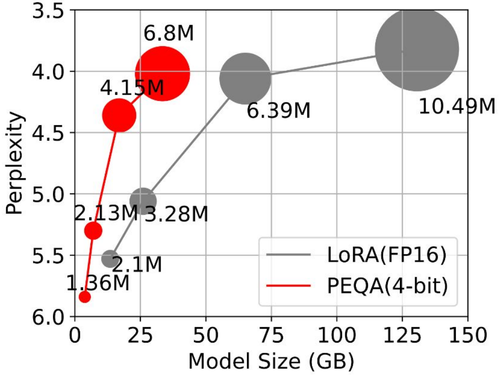
NAVER Cloud

Benefits

- Memory usage comparison of LLaMA 65B



- Perplexity over model size



- Comparison of PEQA with other methods using LLaMA 65B

Method	DRAM (Fine-Tuning)	DRAM (Deployment)	Inference Speed	Task-Switching
Full Fine-Tuning	457GB	131GB	Slow	Slow
PEFT	131GB	131GB	Slow	Fast
PEFT+PTQ	131GB	33GB	Fast	Slow
PTQ+PEFT	33GB	33GB	Slow	Fast
PEQA (Ours)	**33GB**	**33GB**	**Fast**	**Fast**

Experiments

Table 2: To empirically confirm the validity of PEQA's approach, we compare the perplexity (PPL) of fine-tuned LLMs through QAT, PEFT+PTQ, and PEQA on Wikitext2 [51] for GPT-Neo 2.7B, GPT-J 6B, LLaMA 7B, and LLaMA 13B. Weights are quantized into either 3-bit or 4-bit per channel, without a group size [28, 49]. LoRA configuration is set to QV4. The lower PPL, the better.

Method	W Bits	GPT-Neo 2.7B	GPT-J 6B	LLaMA 7B	LLaMA 13B
QAT	4	11.07	8.81	5.76	5.26
LoRA + OPTQ	4	12.09	8.91	7.13	5.31
PEQA (Ours)	4	11.38	8.84	5.84	5.30
QAT	3	12.37	9.60	6.14	5.59
LoRA + OPTQ	3	21.93	11.22	19.47	7.33
PEQA (Ours)	3	12.54	9.36	6.19	5.54

Experiments

Table 3: To show scalability of PEQA, the perplexity (PPL) on Wikitext2 and PennTreeBank (PTB) was compared with LoRA and PEQA. In this comparison, only the weights were quantized into 3-bit and 4-bit per-channel without group size. LoRA configuration is set to QV4. A lower PPL value indicates better performance.

Method	W Bits	GPT-Neo 2.7B	GPT-J 6B	LLaMA 7B	LLaMA 13B	LLaMA 30B	LLaMA 65B
				Wikitext2			
LoRA	16	10.63	8.50	5.53	5.06	4.06	3.82
LoRA+OPTQ	4	12.09	8.91	7.13	5.31	4.39	4.10
PEQA (Ours)	4	**11.38**	**8.84**	**5.84**	**5.30**	**4.36**	**4.02**
LoRA+OPTQ	3	21.93	11.22	19.47	7.33	5.94	5.32
PEQA (Ours)	3	**12.54**	**9.36**	**6.19**	**5.54**	**4.58**	**4.27**
				PTB			
LoRA	16	15.92	12.92	9.14	8.52	7.21	7.11
LoRA+OPTQ	4	18.83	13.46	11.22	8.83	**7.55**	7.46
PEQA (Ours)	4	**16.55**	**13.30**	**9.69**	**8.64**	7.68	**7.36**

	Method	GPT-Neo 2.7B	GPT-J 6B	LLaMA 7B	LLaMA 13B	LLaMA 30B	LLaMA 65B
# of Learnable Param. (M)	LoRA (QV4)	1.31	1.84	2.10	3.28	6.39	10.49
	LoRA (QKVO16)	5.24	7.34	8.39	13.11	25.56	41.94
	PEQA (Ours)	0.74	1.03	1.36	2.13	4.15	6.80
Model Size (GB)	LoRA (QV4)	5.30	12.10	13.48	26.03	65.06	130.57
	PEQA (Ours, 4-bit)	1.53	3.65	3.77	7.01	16.92	33.45
	PEQA (Ours, 3-bit)	1.21	2.94	2.96	5.42	12.90	25.35

NAVER Cloud

Experiments

Table 7: Massive Multitask Language Understanding (MMLU) benchmark performance of PEQA-tuned LLaMAs using Alpaca datasets. Five-shot accuracy is reported for the MMLU. Quantization precision of PEQA is set to 4-bit. When we quantize LLaMA [6] into 4-bit precision using the RTN method, no group size is applied. For LLaMA2 [7], a group size of 256 is used with the RTN method. Note that RTN stands for round-to-nearest in the table.

	# Params	Model Size	Humanities	STEM	Social Sciences	Other	Average
LLaMA [6]	7B	13.5GB	32.6	29.6	38.0	37.9	34.4
	13B	26.1GB	42.8	36.1	53.3	53.2	46.1
	30B	65.1GB	54.6	46.5	66.1	63.4	57.4
+ RTN (w/o group size)	7B	**3.8GB**	28.4	25.6	26.9	31.8	28.3
	13B	**7.0GB**	30.5	27.2	35.5	38.8	32.8
	30B	**16.9GB**	39.6	34.0	46.1	49.7	42.1
+ PEQA	7B	**3.8GB**	35.7	30.9	38.2	40.0	35.8
	13B	**7.0GB**	42.8	37.7	53.6	49.0	45.0
	30B	**16.9GB**	51.1	44.1	62.4	60.7	54.3
LLaMA2 [7]	7B	13.5GB	43.3	37.0	51.8	52.4	45.9
	13B	26.0GB	54.4	44.2	63.4	60.8	55.7
	70B	138.0GB	65.2	57.9	80.3	74.7	69.1
+ RTN (g256)	7B	**3.8GB**	39.5	35.5	49.3	49.9	43.2
	13B	**7.0GB**	50.2	42.6	61.3	59.7	53.2
	70B	**35.3GB**	63.7	55.9	78.4	71.6	67.0
+ PEQA	7B	**3.8GB**	52.0	38.4	54.1	52.0	48.1
	13B	**7.0GB**	60.5	45.0	63.3	57.0	55.3
	70B	**35.3GB**	73.9	55.3	77.8	68.2	67.5

NAVER Cloud

Conclusion

- Interested in fine-tuning a Quantized LLM?
- Seeking to enhance the accuracy of your Quantized LLM?
- Looking to fine-tune your LLM with memory efficiency?

Look no further—use PEQA for your scholarly endeavors!