LoftQ: LoRA-Fine-Tuning-Aware Quantization for Large Language Models

Yixiao Li*, Yifan Yu*, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, Tuo Zhao

Finetune Generative AI on Your PC

Stable Diffusion · MISTRAL AI_ · Phi · LLaMA

- Focus on specific domain.
- Protect your personal and private data.
- Transfer to your own writing or painting style.
- ...

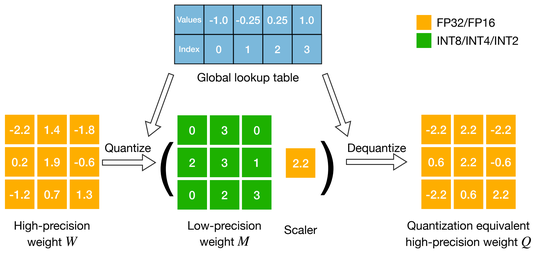
Challenge: Impossible for one RTX 4080 (16GB)

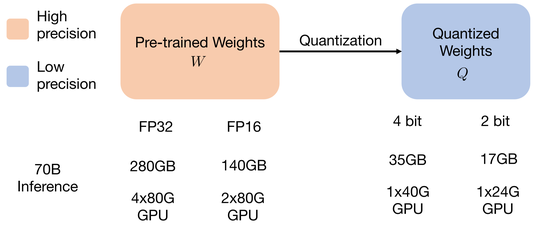
Full-model or LoRA fine-tuning fails.

	Pretrained Weight W	Pretrained Weight W ❄, $B=0$, $A\sim\mathcal{N}(0,\sigma)$ LoRA (Hu et al., 2021)
Memory of the model	7B x 16bit = 14GB	7B x 16bit = 14GB
Memory of grad and optim states	7B x 32bit x 3 = 84GB	5% x 7B x 32bit x 3 = 4.2GB
GPU Footprint	108GB	>27GB

Memory-Efficient Adaption

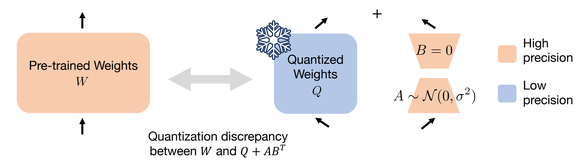
Quantization reduces backbone memory in LoRA.





Memory-Efficient Adaption (Cont'd)

Quantization brings discrepancy.



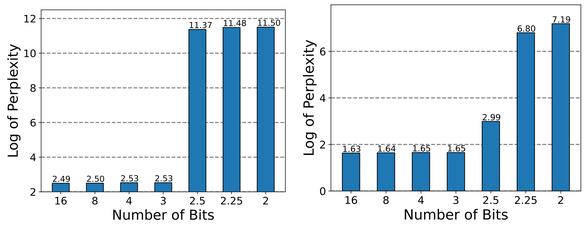


Figure 1. Perplexity of LLaMA-2-7b on Wikitext dataset. Left: test at QLoRA init; Right: test after QLoRA finetuning.

Our Method: LoftQ

LoRA-Aware Quantization: Find a quantized weight Q and low-rank adapters A, B simultaneously, such that

$$\min_{Q,A,B} \|W - Q - AB^\top\|_{\mathrm{F}}^2.$$

- Q is often 4 bit or 2 bit. $A, B \in \mathbb{R}^{d\times r}, r \ll d$.
- Better finetuning results stems from small initialization gaps.

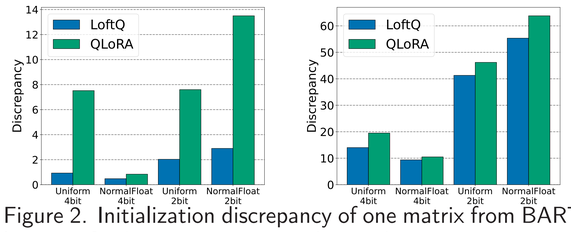


Figure 2. Initialization discrepancy of one matrix from BART-large. Left: Spectrum norm; Right: Frobenius norm.

Alternating Optimization

Input: T: # of iterations; $q(\cdot)$: quantization function.
$A_0 \leftarrow 0, B_0 \leftarrow 0$.
for t = 1 to T **do**
  Quantization: $Q_t \leftarrow q(W - A_{t-1}B_{t-1}^\top)$.
  Low-rank approx: $A_t, B_t \leftarrow \texttt{Truncated-SVD}(W - Q_t)$.
end for
Output: Q_T, A_T, B_T.

- Use Q_T, A_T, B_T as the initialization for LoRA fine-tuning.
- No limit to quantization methods.
- Alternating optimization reinforces Q and A, B.
- Without calibration data.
- Apply it to different weights in parallel.

Main Results

Encoder-Only: DeBERTaV3-base on NLU

Method	Quantization	MNLI m / mm	QNLI Acc	RTE Acc	SST-2 Acc	SQuAD F1	ANLI Acc
Fine-Tune	-	90.5/90.6	94.0	82.0	95.3	92.8	59.8
LoRA	-	90.4/90.5	94.6	85.1	95.1	93.1	60.2
QLoRA	2-bit NF2	75.4/75.6	82.4	55.9	86.5	71.2	N/A
LoftQ	2-bit NF2	**84.7/85.1**	**86.6**	**61.4**	**90.2**	**88.6**	**47.1**
QLoRA	2-bit Uniform	76.5/76.3	83.8	56.7	86.6	77.6	-
LoftQ	2-bit Uniform	**87.3/87.1**	**90.6**	**61.1**	**94.0**	**91.2**	-

"N/A": model does not converge. LoRA rank is 16.

Encoder-Decoder: BART-large on Summarization

Method	Quantization	XSum ROUGE-1/2/L	CNN/DailyMail ROUGE-1/2/L
Lead 3	-	16.30/1.60/11.95	40.42/17.62/36.67
Fine-Tune	-	45.14/22.27/37.25	44.16/21.28/40.90
LoRA	-	43.95/20.72/35.68	45.03/21.84/42.15
QLoRA	4-bit NF4	43.29/20.05/35.15	43.42/20.62/40.44
LoftQ	4-bit NF4	**44.51/21.14/36.18**	**43.96/21.06/40.96**
QLoRA	4-bit Uniform	42.45/19.36/34.38	43.00/20.19/40.02
LoftQ	4-bit Uniform	**44.29/20.90/36.00**	**43.87/20.99/40.92**

LoRA rank is 16.

Decoder-Only: LLaMA-2 on NLG

Method	Quantization	LLaMA-2-7b WikiText-2 PPL↓	LLaMA-2-7b GSM8K Acc↑	LLaMA-2-13b WikiText-2 PPL↓	LLaMA-2-13b GSM8K Acc↑
LoRA	-	5.08	36.9	5.12	43.1
LoRA(w/Reg)	-	-	34.4	-	45.3
QLoRA	4-bit NF	5.70	**35.1**	5.22	39.9
LoftQ	4-bit NF	**5.24**	35.0	**5.16**	**45.0**
QLoRA	3-bit NF	5.73	32.1	5.22	40.7
LoftQ	3-bit NF	**5.63**	**32.9**	**5.13**	**44.4**
QLoRA	2-bit NF	N/A	N/A	N/A	N/A
LoftQ	2-bit NF	**7.85**	**20.9**	**7.69**	**25.4**

N/A: model does not converge. LoRA rank is 64. NF: NormalFloat.

Phi-2 and LLAMA-3 on GSM8K

Model	Bits	Method	GSM8K
Phi-2 (2.7B)	16	Full FT	66.8 ± 1.2
Phi-2 (2.7B)	16	LoRA	64.8 ± 0.5
Phi-2 (2.7B)	4	QLoRA	60.2 ± 0.6
Phi-2 (2.7B)	4	LoftQ	**64.1 ± 0.7**
LLAMA-3-8B	16	Full FT	70.4 ± 0.7
LLAMA-3-8B	16	LoRA	69.3 ± 1.5
LLAMA-3-8B	4	QLoRA	67.4 ± 1.0
LLAMA-3-8B	4	LoftQ	**68.0 ± 0.6**

Quantization method is NF4. LoRA rank is 64.

How to Use LoftQ

Code Example: An Easy Replacement of QLoRA

```
backbone = AutoModelForCausalLM.from_pretrained(
    args.model_name_or_path,
    quantization_config=BitsAndBytesConfig(
        load_in_4bit=True,
        bnb_4bit_quant_type="nf4",
        ...
    ),
)
model = PeftModel.from_pretrained(
    backbone,
    args.model_name_or_path,
    subfolder="loftq_init",
    is_trainable=True,
)
```

Initialize the quantized backbone with LoftQ

Initialize LoRA adapter with LoftQ

Off-the-Shelf Models

- Llama-2 (7B, 13B, 70B)
- CodeLlama(7B, 13B, 70B) Phi-2
- Mistral-7B
- Llama-3 (8B, 70B)
- Llama-3-Instruct (8B, 70B)
- Phi-3 (mini, ...)

References

[1] Dettmers, T., Pagnoni, A., Holtzman, A. and Zettlemoyer, L. (2023). Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314.
[2] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

Resources

Arxiv · Hugginface