QDyLoRA: Quantized Dynamic Low-Rank Adaptation for Efficient Large Language Model Tuning

Hossein Rajabzadeh[1][2] Mojtaba Valipour [1][2] Marzieh Tahaei [1] Hyock Ju Kwon [2]
Ali Ghodsi [2] Boxing Chen [1] Mehdi Rezagholizadeh [1]

[1]Huawei Noah's Ark Lab [2]University of Waterloo

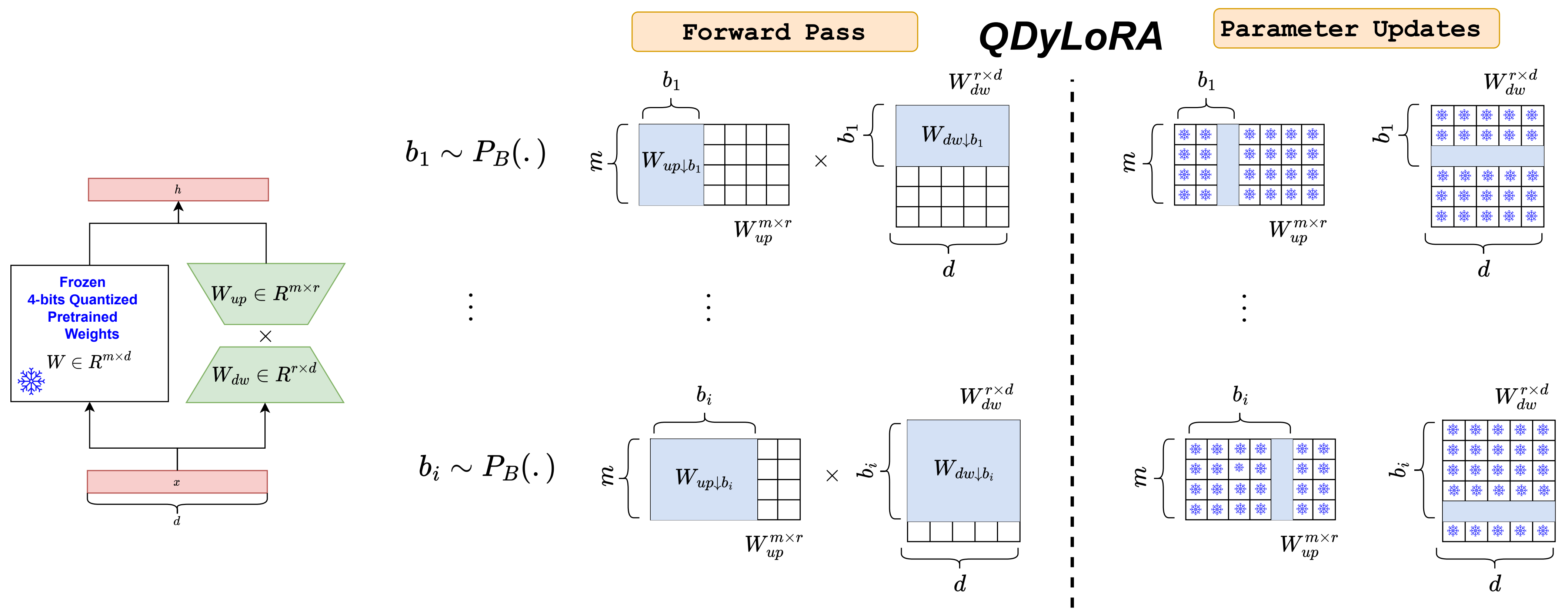

Figure 1. QDyLoRA: The workflow of QDyLoRA. The base model first get quantized into 4-bits precision and then DyLoRA approach starts finetuning LoRA modules for different randomly selected ranks.

Introduction

Finetuning large language models requires huge GPU memory, restricting the choice to acquire Leger language models. While the quantized version of the Low-Rank Adaptation technique, named QLoRA, significantly alleviates this issue, finding the efficient LoRA rank is still challenging. Moreover, QLoRA is trained on a pre-defined rank and, therefore, cannot be reconfigured for its lower ranks without requiring fine-tuning steps. This paper proposes QDyLoRA, combining the advantages of QLoRA with Dynamic LoRA to efficiently finetune LLMs on a set of pre-defined LoRA ranks. QDyLoRA enables fine-tuning Falcon-40b for ranks 1 to 64 on a single 32GiG V100-GPU through one round of fine-tuning.

Proposed Method: QDyLoRA

QDyLoRA - Training and Inference

```
Require: $r \in [r_{min}, r_{max}]$; $i$: the number of training iterations; $\alpha$: a scaling factor; $p_B$: probability distribution function for rank selection; $X \in \mathbb{R}^{d\times n}$ : all input features to LoRA; $W_0 \in \mathbb{R}^{m\times d}$ the original frozen pre-trained weight matrix, $W_{dw} \in \mathbb{R}^{r\times d}$; $W_{up} \in \mathbb{R}^{m\times r}$; $Q$: Quantizer; $\mathbb{L}_{\downarrow b}^{DY}$: objective function given truncated weights
    Initialization:
    $W_0^{NF4} = Q(W_0)$ // Quantize $W_0$ to NF4
    Iterations:
    while t < $i$ do:
        Forward:
        $b \sim p_B(.)$ // sample a specific rank, during test is given
        $W_{dw\downarrow b} = W_{dw}[:b,:]$ // truncate down-projection matrix
        $W_{up\downarrow b} = W_{up}[:,:b]$ // truncate up-projection matrix
        $W_0^{DDequant-NF4} = \frac{W_0^{NF4}}{c_2^{FP8}/c_1^{FP32}}$ // dequantized the chunks of the parameters that are needed
        $h = W_0^{DDequant-NF4} X^{BF16} + \frac{\alpha}{b} W_{up\downarrow b}^{BF16} W_{dw\downarrow b}^{BF16} X^{BF16}$ // calculate the LoRA output
        Backward:
        $W_{dw\downarrow b}^{BF16} \leftarrow W_{dw\downarrow b}^{BF16} - \eta \nabla_{W_{dw\downarrow b}^{BF16}} \mathcal{L}_{\downarrow b}^{\mathcal{DY}}$
        $W_{up\downarrow b}^{BF16} \leftarrow W_{up\downarrow b} - \eta \nabla_{W_{up\downarrow b}^{BF16}} \mathcal{L}_{\downarrow b}^{\mathcal{DY}}$
    end while
```

Contributions

The following is a summary of our contributions:

- **Introduction of QDyLoRA**: The paper introduces QDyLoRA, an efficient quantization approach for dynamic low-rank adaptation. QDyLoRA combines the benefits of QLoRA (Quantized Low-Rank Adaptation) with Dynamic LoRA to allow for efficient fine-tuning of Large Language Models (LLMs) on a set of pre-defined low-rank (LoRA) ranks.
- **Addressing GPU Memory Constraints**: Finetuning large language models often requires substantial GPU memory, limiting the choice of models that can be acquired. QDyLoRA addresses this issue by leveraging the quantized version of the Low-Rank Adaptation technique (QLoRA), significantly reducing memory demands during fine-tuning.
- **Efficient Rank Adaptation**: Unlike QLoRA, which is trained on a pre-defined rank and lacks reconfigurability for lower ranks without additional fine-tuning steps, QDyLoRA introduces dynamic low-rank adaptation. This approach efficiently fine-tunes models on pre-defined LoRA ranks, allowing for adaptive and flexible deployment during inference.
- **Practical Applicability**: QDyLoRA is applied to efficient fine-tuning of LLaMA-7b, LLaMA-13b, and Falcon-40b models across ranks, showcasing its practical applicability on a single 32GB V100 GPU. The determination of optimal rank through inference on the test set further emphasizes the efficiency of the proposed method.

Experiments

Table 1. A comparison between QLoRA and QDyLoRA on the MMLU benchmark, reporting 5-shot test results for LLMs of varying sizes. QDyLoRA is evaluated on ranks [1,2,4,8,16,32,64] and the best rank is reported in brackets.

Dataset	LLaMA-7b		LLaMA-13b		Falcon-40b	
	QLoRA	QDyLoRA	QLoRA	QDyLoRA	QLoRA	QDyLoRA
Alpaca	38.8 [64]	39.7 [16]	47.8 [64]	47.6 [8]	55.2 [64]	57.1 [4]
OASST1	36.6 [64]	36.8 [16]	46.4 [64]	47.2 [8]	56.3 [64]	56.7 [4]
Self-Instruct	36.4 [64]	37.2 [8]	33.3 [64]	41.6 [4]	51.8 [64]	51.1 [4]
FLAN-v2	44.5 [64]	45.9 [4]	51.4 [64]	52.1 [8]	58.3 [64]	60.2 [4]

Table 2. Comparing the performance of QLoRA and QDyLoRA across different evaluation ranks. Both models receives the same training settings. Maximum LoRA rank is set to 64. Falcon-40b is adopted as the base LLM. Exact matching and Bleu-score are used as evaluation measurements for GSM8k and Web-GLM, respectively.

Data	Method	Rank						
		1	2	4	8	16	32	64
Web-GLM	QLoRA	19.9	19.9	19.9	33.8	35.2	52.7	54.3
	QDyLoRA	43.3	**56.0**	54.9	53.3	53.3	50.5	50.2
GSM8k	QLoRA	8.9	8.91	8.9	15.1	20.5	22.6	28.1
	QDyLoRA	21.4	25.3	28.2	**30.6**	29.8	28.5	27.4

Conclusion

We employ the DyLoRA PEFT method in conjunction with the quantization scheme utilized in the QLoRA work, resulting in QDyLoRA. QDyLoRA unifies the advantages of these methods, paving the way towards finetuning larger models in the same GPU memory while resulting in multiple trained sub-models, each with different a LoRA rank.

References

[1] Valipour, Mojtaba, et al. "DyLoRA: Parameter-Efficient Tuning of Pretrained Models using Dynamic Search-Free Low-Rank Adaptation." Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics. 2023.

[2] Dettmers, Tim, et al. "Qlora: Efficient finetuning of quantized llms." Advances in Neural Information Processing Systems 36 (2024).

hossein.rajabzadeh@uwaterloo.ca