Dynamic Low-Rank Module Allocation

Miklos Hamar, Zsombor Fulop, Matey Krastev, Jesse Brouwers

2023/2024
Foundation Models

Supervisor:
Danilo de Goede

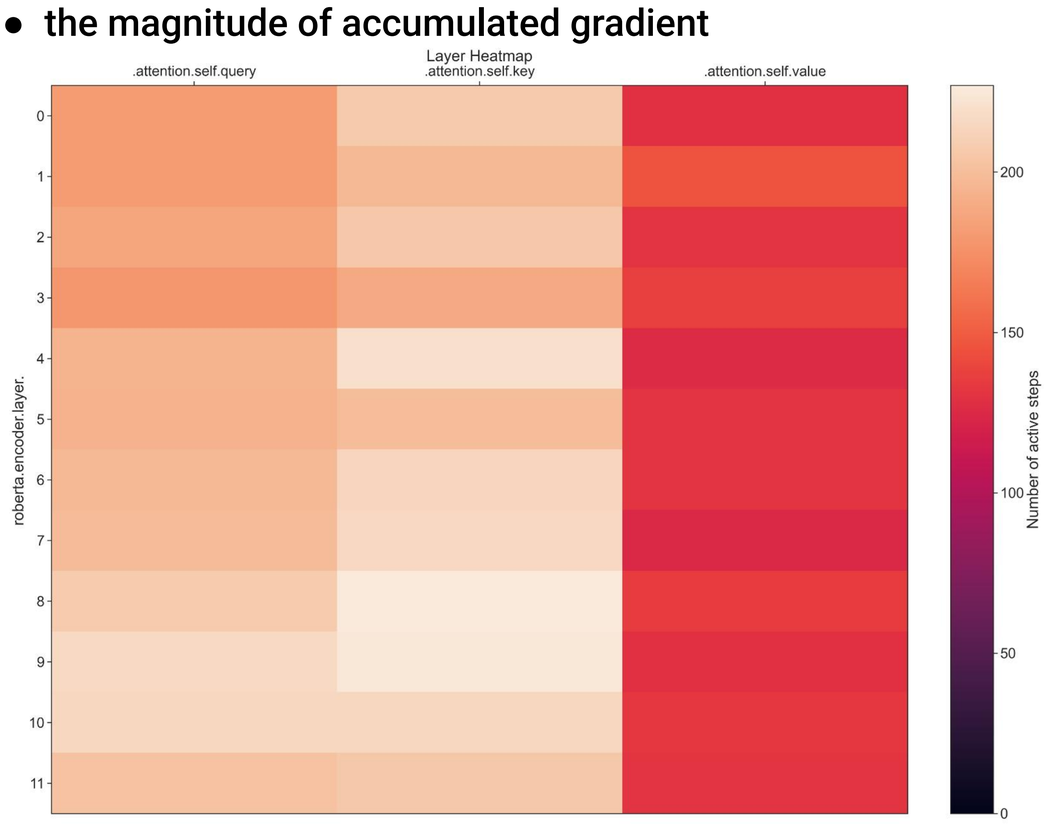
Low-Rank Adaptation (LoRA) [1] is a fine-tuning method which drastically reduces the number of trainable parameters, bringing efficient training and easy application of large pre-trained models. Since its inception, a plethora of enhancements have been proposed in order to bring down even further the costs of fine-tuning while preserving as much of the original performance as possible. But what if training dynamics are the key?

Module Importance

Empirical studies have suggested that certain modules may influence model performance more strongly than others. Targeting these specific modules can directly impact prediction quality.

We propose module importance scores based on:

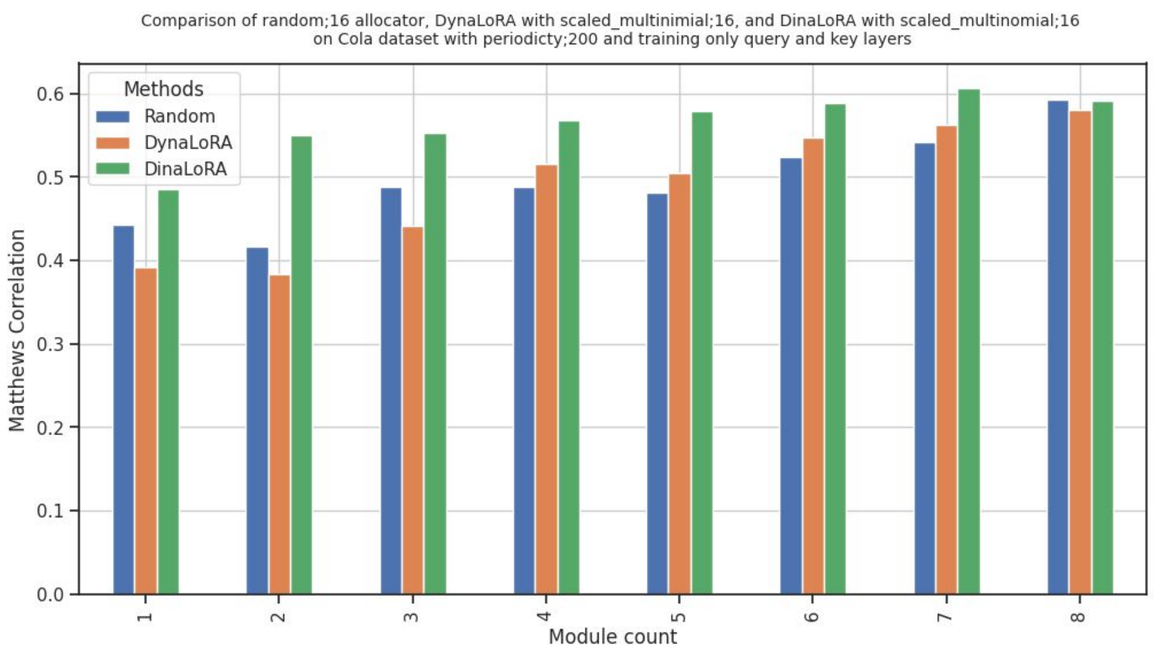
- the magnitude of forward activations [2]
- the magnitude of accumulated gradient



Allocation Schedules

- Re-allocate only once after a specified N steps
- Re-allocate periodically, every N steps

We find that periodic schedules at low intervals yield highest performance but take longer to train.

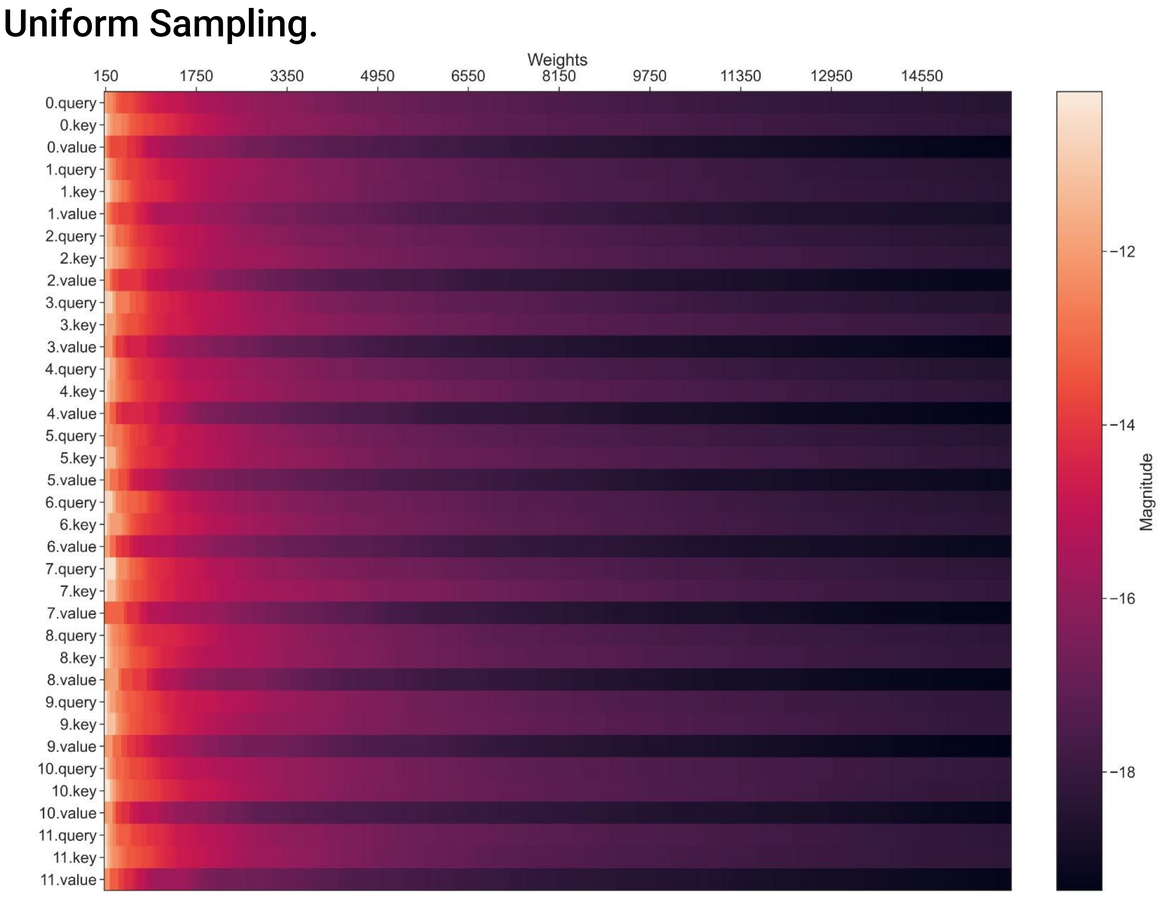


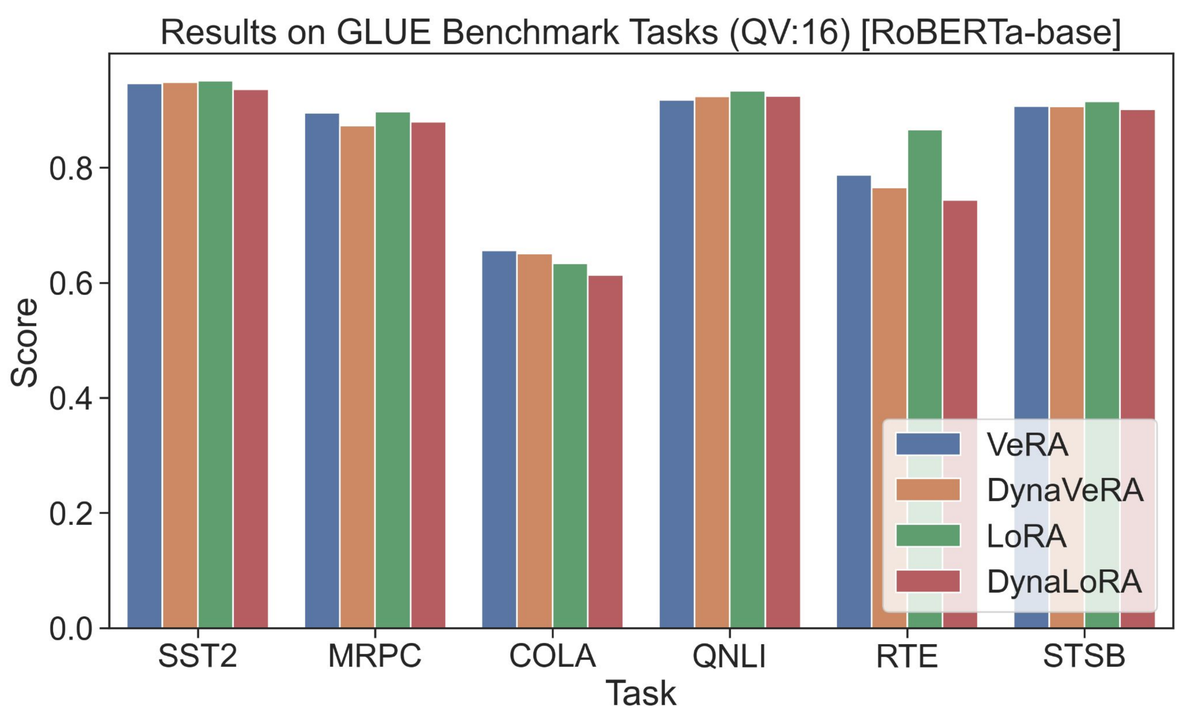
Cost Savings

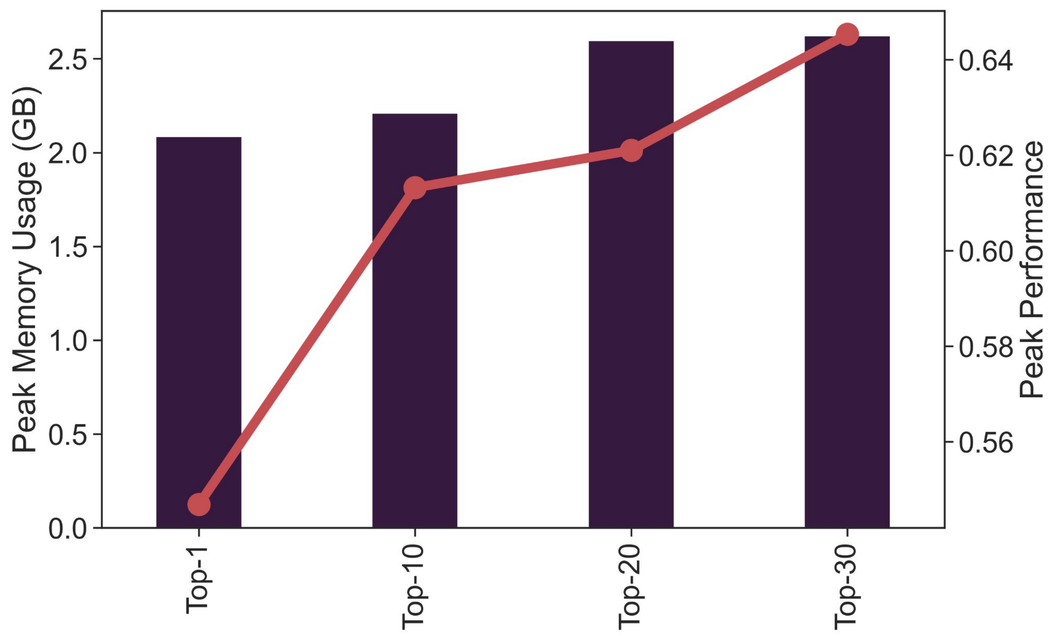
Dynamic allocation can effectively reduce the number of trainable parameters by 50% without major impact on performance, up to 21% memory savings.

Allocation Strategies

- Top-K. Select only the top-k modules with the highest importance scores.
- Threshold (Top-P). Select the top modules with cumulative sum greater than some threshold T.
- Multinomial Sampling. Use module importance scores as sampling weights.
- Discounted Sampling. More frequently chosen modules are discounted to encourage exploration-exploitation during training.
- Uniform Sampling.







Cost Savings

Dynamic allocation can effectively reduce the number of trainable parameters by 50% without major impact on performance, up to 21% memory savings.

SOURCE CODE

References
[1] Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." *arXiv preprint arXiv:2106.09685* (2021).
[2] Zhou, Hongyun, et al. "LoRA-drop: Efficient LoRA Parameter Pruning based on Output Evaluation." *arXiv preprint arXiv:2402.07721* (2024).

Limitations

- Scope of experiments
- Considered benchmarks

Future work

- Extend to larger and more diverse architectures such as ViT's.
- Explore compatibility with rank-pruning methods.