Resource-Efficient Knowledge Editing for Mobile LLMs

ZHENYAN LU[1], DONGQI CAI[1,2], CHEN PENG[1], ZEXI LI[3], SHANGGUANG WANG[1], NICHOLAS D. LANE[2], MENGWEI XU[1],

[1]Beijing University of Posts and Telecommunications, [2]University of Cambridge, [3]Zhejiang University

1 Introduction

Large language models (LLMs) can store knowledge in their parameters, but this knowledge is often outdated or requires personalization. Knowledge editing modifies relevant parameters to integrate new information without compromising generalization [1]. Although parameter-efficient, most existing methods rely on backpropagation, which is unsuitable for mobile devices due to memory limitations and hardware constraints. For example, editing a model of 3B parameters via backpropagation consumes over 8 GB of memory—exceeding the capacity of typical mobile devices. Moreover, mobile NPUs are not optimized for backpropagation and often lack support for critical operations such as `SELECT_OPS` [2].

In this work, we introduce zeroth-order optimization into the mobile knowledge editing process to eliminate the need for backpropagation [2]. This approach perturbs model weights with a random matrix of the same shape and obtains the estimated gradients by observing output changes, enabling memory-efficient editing on mobile hardware.

2 Early Results

Experimental Setup We prototype our editing system atop the open-source ROME framework [1]. We demonstrate editing on the Qwen 2.5 (3B parameters) model using a commercial mobile device with a Snapdragon 8 Gen 3 Elite chipset. Static quantization is applied using an `w8a16` format, calibrated with input data. An asymmetric quantization scheme clips the top 1% of outliers with a per-layer maximum threshold greater than 4. Fake quantization is used with a straight-through estimator, and both gradients and parameter deltas are maintained in FP32. We use a batch size of 7 and a sequence length of 16. Experiments are conducted on the ZsRE dataset.

Editing Performance The edit success rate reaches 86.88%. Portability is 46.11%, locality is 69.30%, and the fluency score is 611.02. These results indicate that zeroth-order editing achieves high fidelity and preserves fluency in most outputs. The relatively lower portability suggests room for improvement in generalizing edits beyond the immediate context. Post-quantization performance remains close to the full-precision baseline, showing minimal accuracy degradation from quantization.

Resource Efficiency Our editing system reduces peak memory usage to 6.2 GiB. In comparison, backpropagation on a smaller 1B model already exceeds 8 GB, making 3B-scale editing infeasible on edge devices. Single-step updates are 10× faster than traditional backpropagation. Although zeroth-order optimization may require more steps, NPUs are roughly 20× faster than CPUs, leading to an estimated 10× reduction in total energy consumption.

References

[1] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems* 35 (2022), 17359–17372.

[2] Mengwei Xu, Dongqi Cai*, Yaozong Wu, Xiang Li, and Shangguang Wang. 2024. {FwdLLM}: Efficient federated finetuning of large language models with perturbed inferences. In *2024 USENIX Annual Technical Conference (USENIX ATC 24)*. 579–596.

[1]https://github.com/kmeng01/rome

Author's Contact Information: dc912@cam.ac.uk.