Enhancing Generalization through Task Vector Fusion in Deep Reinforcement Learning for Database Optimization

Taiyi Wang
Taiyi.Wang@cl.cam.ac.uk

Eiko Yoneki
eiko.yoneki@cl.cam.ac.uk

Recent advancements in the integration of Reinforcement Learning (RL) with database systems, notably in areas such as RL-enhanced query optimizer and index selection [2, 5], underscore the burgeoning interest in exploiting RL's inherent strengths in scheduling and solving complex combinatorial optimization problems. This interdisciplinary approach exploits RL's potential to enhance efficiency and performance by optimizing the complex spaces of database operations.

The migration of over 75% of databases to cloud environments by 2022 challenges cloud vendors to manage physical designs efficiently in SaaS scenarios [4]. This necessitates swift optimization strategies for dynamic workloads and performance maintenance. Traditional RL model training, with its static nature, faces difficulties adapting to the changing needs of such databases. Typically, RL models within database environments are calibrated for fixed workloads and settings, implying a necessity for retraining or adaptation as workloads evolve [3]. Given the dynamic nature of data and workloads—which are in a constant state of flux—this requirement poses significant practical hurdles. Retraining models to accommodate new or changing workloads incurs substantial computational and temporal costs, an obstacle that is particularly pronounced in database systems where real-time processing is paramount.

In response to this dilemma, we introduce an intuitive method that utilizes the fusion of task vectors from previously trained models, facilitating dynamic adaptation to changing workloads with minimal retraining effort. This approach offers a user-friendly and convenient solution for maintaining optimal database performance amidst evolving demands. Task vectors, in this context, represent the parameter sets or adjustments specific to individual tasks or workloads, encapsulating the essence of the model's learning relative to that task [1]. By fusing or performing arithmetic operations on these vectors, our methodology aims to harness the collective insights from existing models to forecast the optimal parameter adjustments for new, unseen workloads.

In advancing this novel approach, our methodology introduces a tailor-made algorithm for the fusion of task vectors, which are designed to capture differences among workloads. This algorithm identifies key features within these vectors for effective fusion and applies arithmetic operations to enhance model adaptability and prediction accuracy for new workloads. Central to our contribution is the development of an optimization framework that iteratively refines the fusion process, ensuring that task vectors optimally align with the dynamic characteristics of new downstream tasks. This framework is underpinned by a robust evaluation mechanism that assesses the generalization performance of the adapted models. In our paper, we detail how our system supports RL-based optimization tasks including learned index selection, query optimization, and index tuning. By reducing the need for frequent retraining and employing task vector fusion, we significantly improve the RL model's abilities to generalize across database environments. This approach marks a progression towards optimization solutions that are not only more efficient but also highly adaptive, catering to the dynamic needs of modern database systems.

References

[1] Ilharco G, Ribeiro M T, Wortsman M, et al. Editing models with task arithmetic[J]. arXiv preprint arXiv:2212.04089, 2022.

[2] Kossmann, Jan, Alexander Kastius, and Rainer Schlosser. "SWIRL: Selection of Workload-aware Indexes using Reinforcement Learning." EDBT. 2022.

[3] Kraska T, Alizadeh M, Beutel A, et al. Sagedb: A learned database system[J]. 2021.

[4] Ma, Lin, et al. "Query-based workload forecasting for self-driving database management systems." Proceedings of the 2018 International Conference on Management of Data. 2018

[5] Marcus R, Negi P, Mao H, et al. Bao: Learning to steer query optimizers[J]. arXiv preprint arXiv:2004.03814, 2020.