Production-Ready Learning-Augmented Data Management with Deep Reinforcement Learning*

Gabriel Campero Durand
Supervisor: Gunter Saake

University of Magdeburg, Universitätsplatz 2, 39106, Magdeburg, Germany
campero@ovgu.de

Abstract. Reports on the ability of deep reinforcement learning (DRL) models to master highly-challenging decision making, have spurred the interest of practitioners to use this approach for improving the performance of computer systems at tasks that have been traditionally solved with amendable human-engineered heuristics. However, when applying DRL to specific cases, such as data management applications (DRLDM), the impact of design choices (concerning problem framing, model characteristics, training process and application-specific aspects), paired with the challenges intrinsic to the learning task, are currently not entirely understood. The understanding of these aspects, through detailed studies, is essential for maturing the emerging area of DRLDM. In our work we seek to address these research issues by developing two case studies: RL-backed join order optimization with cardinality estimation (REJOICE) and RL-based physical data partitioning (GridFormation). We propose to investigate them in three production configurations: online learning, learning from demonstrations and learning from an experimental replica/simulation.

Keywords: Database Tuning· Deep Reinforcement Learning· Applied Machine Learning

1 Introduction and Motivation

Data management tools aim to facilitate improvements in the offerings of data-intensive applications. To accomplish this they provide general query and configuration interfaces, abstracting users from the complexities of system internals. In this way, across their many shapes (e.g. single-user in-memory libraries, multi-user databases or large-scale processing frameworks), data management tools fulfill an important role in modern organizations and society.

Two fundamental prerequisites for the efficiency of these tools is that: a) their chosen configuration should match well the workload that they process, and b)

* This work is partially funded by the DFG (grant no.: SA 465/50-1). Accepted for a short presentation in the ECML PKDD 2019 PhD Forum http://ecmlpkdd2019.org/submissions/phdforum/.

the mapping from declarative requests, to actual operations should be optimal. In many cases, the tasks required to fulfill these goals involve automated sequential decision making. Some examples include the ordering of predicate evaluation in a query, or the iterative selection of indexes to create. Given the difficulty of accurately modeling real-world performance impact factors, tools commonly address decision making with human-engineered heuristics. As other researchers have noted [10], this leaves room for improvement in their performance [3, 7], specially when considering that heuristics can result complex to maintain as systems evolve, and difficult to adapt (e.g. to changing workload characteristics). In recent years, deep reinforcement learning (DRL) has been established to be a proficient approach for highly-complex sequential decision making [12, 16, 1]. This has motivated a growing body of work proposing applications of DRL (as an alternative to traditional heuristics) for data management tasks in storage [15, 5] and query engines [13, 11, 6, 8], with promising experimental results. Taken together, these studies articulate a compelling research direction for evolving the capabilities of data management systems in new ways, by complementing time-tested solutions with learned models that should meet and exceed the standards set by the original solutions.

This initiative of applying DRL models in computer systems opens-up many design possibilities concerning: the problem framing, model characteristics, the training process and how the model is used by the end system. However, since the application of DRL in data management (DRLDM) is relatively new, the *impact of design choices* and the *challenges* intrinsic to the studied applications in this domain have not been fully mapped out. The study of these two aspects is essential for DRLDM tools to realize their potential for impacting real-world data applications in organizations.

In our research we study DRLDM tools, the impact of design choices and the identification of their key challenges. We take as case studies the development of two single agent DRL tools for relational databases, involving long-term planning: RL-backed join order optimization incorporating cardinality estimates (REJOICE), and physical data partitioning (GridFormation). These represent query and storage engine applications, respectively. Furthermore, we shape our work by considering specific configurations, as shown in Fig. 1. The proposed configurations represent three feasible alternatives to organize the learning process in production systems.

Research on DRLDM tools nowadays benefits from emerging initiatives providing standard extensible environments to evaluate reinforcement learning models applied to computer systems [10], coupled with the progress in DRL and deep learning frameworks for efficient model management. The aim to adopt and contribute to standards constitutes also a key choice shaping our work.

2 Research Issues

1. *Application-side contribution:* The foremost concern about DRLDM solutions is to be able to properly evaluate the extent of their actual contri-

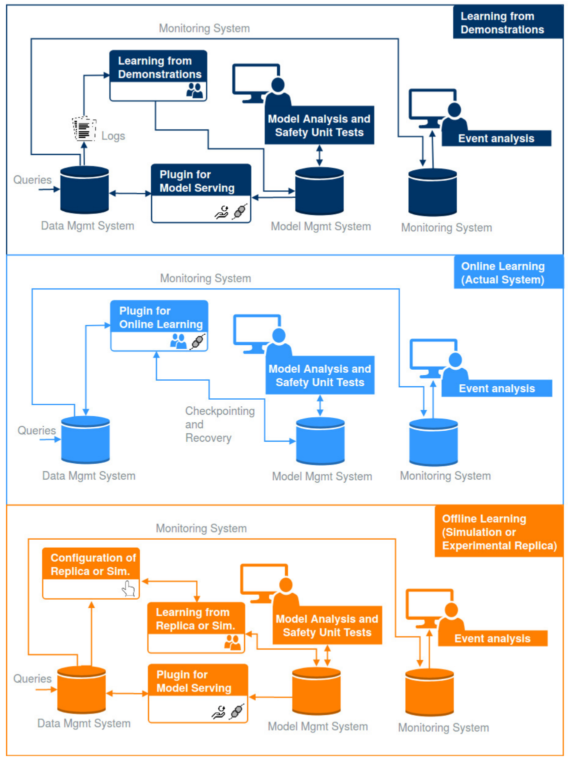

Fig. 1. Three learning configurations for applying reinforcement learning models in production data systems: learning from demonstrations, learning from the actual system and learning from an experimental replica/simulation.

bution to the overall system. In this regard, evaluation measures can be categorized according to focus areas: a) the performance at the learned task (effectiveness), b) the stability when facing safety (specification and robustness) challenges, c) the quality of the integration with the data system (e.g. contrasting the running time of models at inference w.r.t. that of baseline heuristics, or examining whether the model can be trained online without deteriorating the end system), d) the maintainability of the model (e.g. concerning its complexity and the ease for understanding data/behavior changes that should trigger model improvements), and finally e) the maintainability of the learning configuration (e.g. concerning the requirements from log collection, or from keeping simulated/replica environments).

Key Questions: What are the necessary metrics and application-specific benchmarks to assess DRLDM solutions in production settings? How can safety unit tests be defined for an application, and what measures of test coverage can be incorporated? What practices can improve the quality of the model integration with the data system, and the maintainability of the overall solution?

2. *Impact of problem framing:* There is usually a plethora of alternative observation spaces indicating different levels of information about the learning problem. In the context of DRLDM, time-varying settings could furthermore be considered, to account for rapid changes in task complexity or stochasticity in real-world rewards [10]). In order to converge to optimal designs, the role of such alternatives needs to be established.

   Key Questions: What are the trade-offs in the performance of agents, when changing the complexity of observation spaces? Can adaptive observation spaces be exploited for addressing reality-gap challenges? What measures can be given for a proper balance when decomposing actions into successive models, as a strategy to limit the size of the search space?

3. *Impact of model characteristics:* The state of the art in DRL already offers an abundance of possible model classes to employ, either individually or in creative combinations. Naturally, understanding the role of the model chosen for the application is a driving necessity for research in DRLDM. Beyond this, production-ready models require meticulous evaluations on the role of the neural network design, encompassing the hyper-parameters chosen. Within this context, departures from traditional DRL designs can reasonably require consideration for production-ready DRLDM, such as the use of neural Bayesian learning for efficient exploitation [14], or the adoption of graph-structured neural networks to cater for features naturally modeled as graphs [11], such as query plans.

   Key Questions: What experimentally-backed guidelines can be used to map a DRLDM problem to a model class? To what degree can specialized layers contribute to the overall performance of DRLDM models?

4. *Impact of training process:* Whether the goal is to excel at a highly-specific task configuration, or to generalize to a large space of possible instances, the training of DRLDM models can be configured in many ways. One useful approach for generalization is curriculum learning [2], where tasks are presented to agents in increasing levels of difficulty, and agents only move to successive tasks after mastering simpler ones.

   Key Questions: When should curriculum learning be employed in DRLDM tools, over ad-hoc or hand-crafted alternatives? To what extent can techniques to automate curriculum generation help the learning process? What trade-offs exist in selecting such techniques?

The contributions from addressing the outlined research issues should be: a) novel open-source, readily-applicable DRLDM tools, with b) a detailed identification on the impact of design choices in these tools, c) the documentation of their challenges, both general and pertaining to design choices; and finally d) guidelines/best practices for how novel DRLDM tools can be adopted, grounded on rigorous evaluations.

3 Research Plan

Approach We structure our work as two comprehensive case studies. For each case we investigate the pertinent research issues, as outlined in Sec. 2. We follow the pre-defined learning configurations presented in Fig. 1. In order to cover a wide variety of choices we employ four off-the-shelf DRL frameworks.

1. **REJOICE:** We select for our first study the case of join order optimization. This is one of the most studied database problems. As such, it has well-established, challenging benchmarks. This problem has already been studied with DRL [13, 11, 6]. To be formulated as a Markov Decision Process (MDP), the agent takes as input one query and each action consists of determining a pair of tables to join, until there are no more joins left. We distinguish our design from prior work, by seeking to combine DRL-supported selectivity estimation with the optimization itself. Rewards are given by the actual query running time, but can also be defined on cost models.
2. **GridFormation:** For our second study we pick physical data partitioning. This task consists of decomposing a relation in either vertical, horizontal, or hybrid partitions (at increasing levels of complexity, from the perspective of the optimization problem), in order to improve either I/O, memory or network usage while processing an expected workload. In contrast to join order optimization, this task does not count with dedicated challenging benchmarks. A further complication in studying storage engine tasks, is that these often rely on accurate forecasts of the expected workload (i.e., at least query patterns). Gathering such forecasts in an automated manner constitutes a sub-task that demands separate consideration [9]. To formulate partitioning as an MDP, we have proposed the following actions: the replication of a partition, the deletion of a replicated partition, splitting one partition into two, or merging two partitions [4]. Rewards can be given by cost models or by the actual workload running time.

4 Conclusion

In this progress paper we provide a sketch of our early research initiative to further the understanding on how to build tools for deep reinforcement learning in data management (DRLDM). We establish as two guiding research issues in the field: the need to study rigorously design impact factors and to identify the challenges pertinent to each task. We further outline the basic categories of aspects that need to be studied for appraising in full the impact of design choices: application-contribution, problem framing, model characteristics and training design. We propose three learning settings for DRLDM tools in production scenarios, and we introduce briefly the two case studies that we are considering in our work.

References

1. Bello, I., Pham, H., Le, Q.V., Norouzi, M., Bengio, S.: Neural combinatorial optimization with reinforcement learning. arXiv preprint arXiv:1611.09940 (2016)
2. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning. pp. 41–48. ACM (2009)
3. Borovica, R., Alagiannis, I., Ailamaki, A.: Automated physical designers: what you see is (not) what you get. In: Proceedings of the Fifth International Workshop on Testing Database Systems. p. 9. ACM (2012)
4. Durand, G.C., Pinnecke, M., Piriyev, R., Mohsen, M., Broneske, D., Saake, G., Sekeran, M.S., Rodriguez, F., Balami, L.: Gridformation: Towards self-driven online data partitioning using reinforcement learning. In: Proceedings of the First International Workshop on Exploiting Artificial Intelligence Techniques for Data Management. p. 1. ACM (2018)
5. Hilprecht, B., Binnig, C., Roehm, U.: Learning a partitioning advisor with deep reinforcement learning. arXiv preprint arXiv:1904.01279 (2019)
6. Krishnan, S., Yang, Z., Goldberg, K., Hellerstein, J., Stoica, I.: Learning to optimize join queries with deep reinforcement learning. arXiv preprint arXiv:1808.03196 (2018)
7. Leis, V., Gubichev, A., Mirchev, A., Boncz, P., Kemper, A., Neumann, T.: How good are query optimizers, really? Proceedings of the VLDB Endowment **9**(3), 204–215 (2015)
8. Liang, X., Elmore, A.J., Krishnan, S.: Opportunistic view materialization with deep reinforcement learning. arXiv preprint arXiv:1903.01363 (2019)
9. Ma, L., Van Aken, D., Hefny, A., Mezerhane, G., Pavlo, A., Gordon, G.J.: Query-based workload forecasting for self-driving database management systems. In: Proceedings of the 2018 International Conference on Management of Data. pp. 631–645. ACM (2018)
10. Mao, H., Narayan, A., Negi, P., Wang, H., Yang, J., Wang, H., Khani, M., He, S., Addanki, R., Marcus, R., et al.: Park: An open platform for learning augmented computer systems (2019)
11. Marcus, R., Negi, P., Mao, H., Zhang, C., Alizadeh, M., Kraska, T., Papaemmanouil, O., Tatbul, N.: Neo: A learned query optimizer. arXiv preprint arXiv:1904.03711 (2019)
12. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
13. Ortiz, J., Balazinska, M., Gehrke, J., Keerthi, S.S.: Learning state representations for query optimization with deep reinforcement learning. arXiv preprint arXiv:1803.08604 (2018)
14. Riquelme, C., Tucker, G., Snoek, J.: Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. arXiv preprint arXiv:1802.09127 (2018)
15. Sharma, A., Schuhknecht, F.M., Dittrich, J.: The case for automatic database administration using deep reinforcement learning. arXiv preprint arXiv:1801.05643 (2018)
16. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature **529**(7587), 484 (2016)